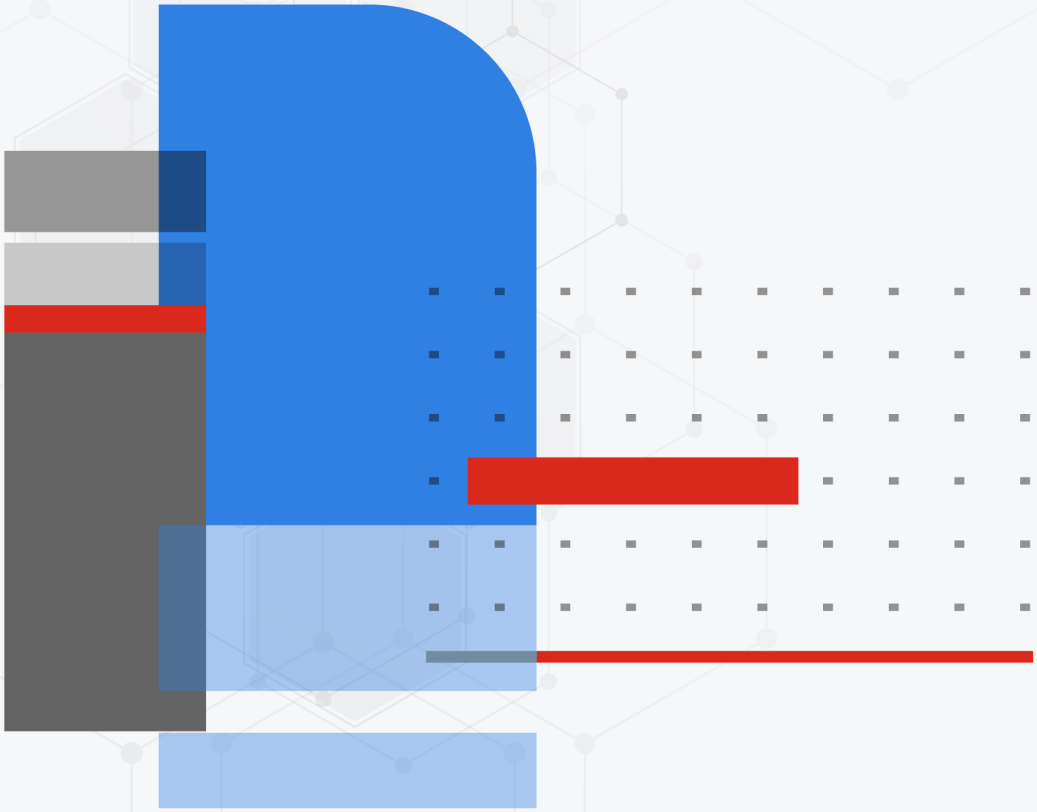
Dataset Reference

FortiAnalyzer 7.4.10



FORTINET DOCUMENT LIBRARY
https://docs.fortinet.com

FORTINET VIDEO LIBRARY
https://video.fortinet.com

FORTINET BLOG
https://blog.fortinet.com

CUSTOMER SERVICE & SUPPORT
https://support.fortinet.com

FORTINET TRAINING & CERTIFICATION PROGRAM
https://www.fortinet.com/training-certification

FORTINET TRAINING INSTITUTE
https://training.fortinet.com

FORTIGUARD LABS
https://www.fortiguard.com

END USER LICENSE AGREEMENT
https://www.fortinet.com/doc/legal/EULA.pdf

FEEDBACK
Email: techdoc@fortinet.com

January 27, 2026
FortiAnalyzer 7.4.10 Dataset Reference
05-7410-901329-20260127

TABLE OF CONTENTS

Change Log **4**
Introduction **5**
Understanding datasets and macros 5
Dataset Reference List **6**
Macro Reference List **531**

Change Log

Date	Change Description
2026-01-27	Initial release.

Introduction

This document provides information about the various types of FortiAnalyzer datasets.

Understanding datasets and macros

FortiAnalyzer datasets are collections of log messages from monitored devices.

Charts in FortiAnalyzer are generated based on the datasets. To create a chart, you can use the predefined datasets, or you can create your own custom datasets by querying the log messages in the SQL database on the FortiAnalyzer unit. Both predefined and custom datasets can be cloned, but only custom datasets can be deleted. You can also view the SQL query for a dataset, and test the query against specific devices or log arrays.

You can create custom reports that contain macros that are created based on predefined and custom datasets. Macros are used to dynamically display the device log data as text in a report. They can be embedded within a text field of a paragraph in a report layout in XML format. Macros display a single value, such as a user name, highest session count, or highest bandwidth, and so on.

For more information about how to create datasets, charts, and macros, see the FortiAnalyzer *Administration Guide*.

Dataset Reference List

The following tables list the datasets included with FortiAnalyzer. The tables contain the name, SQL query syntax, and log category for each dataset.

Dataset Name	Description	Log Category
Traffic-Bandwidth-Summary-Day-Of-Month	Traffic bandwidth timeline	traffic

```
select
  $flex_timescale(timestamp) as hodex,
  sum(traffic_out) as traffic_out,
  sum(traffic_in) as traffic_in
from
  ###(select timestamp, sum(bandwidth) as bandwidth, sum(traffic_out) as traffic_out, sum(traffic_
in) as traffic_in, sum(sessions) as sessions from ###base(/*tag:rpt_base_t_bndwdth_sess*/select
$flex_timestamp as timestamp, dvid, srcip, dstip, epid, euid, appcat, apprisk, coalesce(nullifna
(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service, sum(CASE WHEN
(logflag&1>0) THEN 1 ELSE 0 END) as sessions, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum
(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in from $log-traffic where $filter and (logflag&
(1|32)>0) group by timestamp, dvid, srcip, dstip, epid, euid, appcat, apprisk, user_src, service
/*SkipSTART*/order by bandwidth desc, sessions desc/*SkipEND*/)base### base_query group by
timestamp order by bandwidth desc, sessions desc)### t where $filter-drilldown group by hodex
having sum(traffic_out+traffic_in)>0 order by hodex
```

Dataset Name	Description	Log Category
Session-Summary-Day-Of-Month	Number of session timeline	traffic

```
select
  $flex_timescale(timestamp) as hodex,
  sum(sessions) as sessions
from
  ###(select timestamp, sum(bandwidth) as bandwidth, sum(traffic_out) as traffic_out, sum(traffic_
in) as traffic_in, sum(sessions) as sessions from ###base(/*tag:rpt_base_t_bndwdth_sess*/select
$flex_timestamp as timestamp, dvid, srcip, dstip, epid, euid, appcat, apprisk, coalesce(nullifna
(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service, sum(CASE WHEN
(logflag&1>0) THEN 1 ELSE 0 END) as sessions, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum
(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in from $log-traffic where $filter and (logflag&
(1|32)>0) group by timestamp, dvid, srcip, dstip, epid, euid, appcat, apprisk, user_src, service
/*SkipSTART*/order by bandwidth desc, sessions desc/*SkipEND*/)base### base_query group by
timestamp order by bandwidth desc, sessions desc)### t where $filter-drilldown group by hodex
order by hodex
```

Dataset Name	Description	Log Category
Top-Users-By-Bandwidth	Bandwidth application top users by bandwidth usage	traffic

```
select
  user_src,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out,
  sum(sessions) as sessions
from
  ###(select appid, app, appcat, apprisk, user_src, hostname, dstip, sum(traffic_in) as traffic_
in, sum(traffic_out) as traffic_out, sum(bandwidth) as bandwidth, sum(sessions) as sessions from
###base(/*tag:rpt_base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid,
euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service,
appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions
from $log-traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by
timestamp, dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat, apprisk,
hostname order by bandwidth desc, sessions desc)base### t group by appid, app, appcat, apprisk,
user_src, hostname, dstip /*SkipSTART*/order by bandwidth desc, sessions desc/*SkipEND*/)### t
group by user_src having sum(bandwidth)>0 order by bandwidth desc
```

Dataset Name	Description	Log Category
Top-App-By-Bandwidth	Top applications by bandwidth usage	traffic

```
select
  app_group_name(app) as app_group,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out,
  sum(sessions) as sessions
from
  ###(select appid, app, appcat, apprisk, user_src, hostname, dstip, sum(traffic_in) as traffic_
in, sum(traffic_out) as traffic_out, sum(bandwidth) as bandwidth, sum(sessions) as sessions from
###base(/*tag:rpt_base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid,
euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service,
appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions
from $log-traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by
timestamp, dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat, apprisk,
hostname order by bandwidth desc, sessions desc)base### t group by appid, app, appcat, apprisk,
user_src, hostname, dstip /*SkipSTART*/order by bandwidth desc, sessions desc/*SkipEND*/)### t
group by app_group having sum(bandwidth)>0 order by bandwidth desc
```

Dataset Name	Description	Log Category
Top-User-Source-By-Sessions	Top user source by session count	traffic

```
select
  user_src,
  sum(sessions) as sessions
from
  ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, count
(*) as sessions from $log where $filter and (logflag&1>0) group by user_src order by sessions
desc)### t group by user_src order by sessions desc
```

Dataset Name	Description	Log Category
Top-App-By-Sessions	Top applications by session count	traffic

```
select
  app_group,
  sum(sessions) as sessions
from
  ###(select app_group_name(app) as app_group, appcat, service, sum(coalesce(sentdelta, sentbyte,
0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as
traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from $log
where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by app_group, appcat,
service order by bandwidth desc)### t group by app_group order by sessions desc
```

Dataset Name	Description	Log Category
Top-Destination-Addresses-By-Sessions	Top destinations by session count	traffic

```
select
  coalesce(
    nullifna(
      root_domain(hostname)
    ),
    ipstr(dstip)
  ) as domain,
  count(*) as sessions
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
group by
  domain
```

```
order by
  sessions desc
```

Dataset Name	Description	Log Category
Top-Destination-Addresses-By-Bandwidth	Top destinations by bandwidth usage	traffic

```
select
  coalesce(
    nullifna(
      root_domain(hostname)
    ),
    ipstr(dstip)
  ) as domain,
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  ) as bandwidth,
  sum(
    coalesce(rcvdbyte, 0)
  ) as traffic_in,
  sum(
    coalesce(sentbyte, 0)
  ) as traffic_out
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
  and coalesce(
    nullifna(
      root_domain(hostname)
    ),
    ipstr(`dstip`)
  ) is not null
group by
  domain
having
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  )> 0
order by
  bandwidth desc
```

Dataset Name	Description	Log Category
DHCP-Summary-By-Port	Event top dhcp summary	event

```
drop
  table if exists rpt_tmptbl_1;
drop
  table if exists rpt_tmptbl_2;
drop
  table if exists rpt_tmptbl_3; create temporary table rpt_tmptbl_1 as
select
  devintf,
  mac
from
  ###(select concat(interface, '.', devid) as devintf, mac, count(*) as total_num from $log where
$last3day_period $filter and logid_to_int(logid) = 26001 and dhcp_msg = 'Ack' group by devintf,
mac order by total_num desc)### t group by devintf, mac; create temporary table rpt_tmptbl_2 as
select devintf, mac from ###(select concat(interface, '.', devid) as devintf, mac, count(*) as
total_num from $log where $filter and logid_to_int(logid) = 26001 and dhcp_msg = 'Ack' group by
devintf, mac order by total_num desc)### t group by devintf, mac; create temporary table rpt_
tmptbl_3 as select distinct on (1) devintf, cast(used*100.0/total as decimal(18,2)) as percent_of_
allocated_ip from ###(select distinct on (devintf) concat(interface, '.', devid) as devintf, used,
total, itime from $log where $filter and logid_to_int(logid)=26003 and total>0 /*SkipSTART*/order
by devintf, itime desc/*SkipEND*/)### t order by devintf, itime desc; select t1.devintf as
interface, percent_of_allocated_ip, new_cli_count from rpt_tmptbl_3 t1 inner join (select devintf,
count(mac) as new_cli_count from rpt_tmptbl_2 where not exists (select 1 from rpt_tmptbl_1 where
rpt_tmptbl_2.mac=rpt_tmptbl_1.mac) group by devintf) t2 on t1.devintf=t2.devintf order by
interface, percent_of_allocated_ip desc
```

Dataset Name	Description	Log Category
Top-Wifi-Client-By-Bandwidth	Traffic top WiFi client by bandwidth usage	traffic

```
select
  user_src,
  srcssid,
  devtype_new,
  hostname_mac,
  sum(bandwidth) as bandwidth
from
  (
    select
      user_src,
      srcssid,
      get_devtype(srcswversion, osname, devtype) as devtype_new,
      hostname_mac,
      sum(bandwidth) as bandwidth
    from
      ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
ap, srcintf, srcssid, srcssid as ssid, srcmac, srcmac as stamac, coalesce(nullifna(`srcname`),
`srcmac`) as hostname_mac, max(srcswversion) as srcswversion, max(osname) as osname, max
(osversion) as osversion, max(devtype) as devtype, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte,
0)) as bandwidth, count(*) as subtotal from $log-traffic where $filter and (logflag&1>0) and
(srcssid is not null or dstssid is not null) group by user_src, ap, srcintf, srcssid, srcmac,
hostname_mac /*SkipSTART*/order by bandwidth desc, subtotal desc/*SkipEND*/)### t group by user_
```

```
src, srcssid, devtype_new, hostname_mac having sum(bandwidth)>0 union all select user_src, ssid as
srcssid, null as devtype_new, stamac as hostname_mac, sum(bandwidth) as bandwidth from ###(select
$flex_timestamp as timestamp, stamac, stamac as srcmac, ap, ssid, ssid as srcssid, user_src, sum
(coalesce(sentdelta, 0)) as sentdelta, sum(coalesce(rcvddelta, 0)) as rcvddelta, sum(coalesce
(sentdelta, 0)+coalesce(rcvddelta, 0)) as bandwidth from (select itime, stamac, ap, ssid, coalesce
(`user`, ipstr(`srcip`)) as user_src, sentbyte-lag(coalesce(sentbyte, 0)) over (partition by
stamac order by itime) as sentdelta, rcvdbyte-lag(coalesce(rcvdbyte, 0)) over (partition by stamac
order by itime) as rcvddelta from $log-event where $filter and subtype='wireless' and stamac is
not null and ssid is not null and action in ('sta-wl-bridge-traffic-stats', 'reassoc-req', 'assoc-
req')) as t group by timestamp, stamac, ap, ssid, user_src /*SkipSTART*/order by bandwidth
desc/*SkipEND*/)### t where user_src is not null group by user_src, ssid, devtype_new, stamac
having sum(bandwidth)>0) t group by user_src, srcssid, devtype_new, hostname_mac order by
bandwidth desc
```

Dataset Name	Description	Log Category
Traffic-History-By-Active-User	Traffic history by active user	traffic

```
select
  $flex_timescale(timestamp) as hodex,
  count(
    distinct(user_src)
  ) as total_user
from
  ###(select timestamp, user_src, sum(sessions) as sessions from ###base(/*tag:rpt_base_t_bndwdth_
sess*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid, euid, appcat, apprisk,
coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service, sum(CASE
WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum
(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in from $log-traffic where $filter and (logflag&
(1|32)>0) group by timestamp, dvid, srcip, dstip, epid, euid, appcat, apprisk, user_src, service
/*SkipSTART*/order by bandwidth desc, sessions desc/*SkipEND*/)base### base_query group by
timestamp, user_src order by sessions desc)### t where $filter-drilldown group by hodex order by
hodex
```

Dataset Name	Description	Log Category
Top-Allowed-Websites-By-Requests	UTM top allowed web sites by request	traffic

```
select
  hostname,
  catdesc,
  count(*) as requests
from
  $log
where
  $filter
  and (
```

```
    logflag&1>0
  )
  and utmevent in (
    & #039;webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') and
hostname is not null and (utmaction not in ('block', 'blocked') or action!='deny') group by
hostname, catdesc order by requests desc
```

Dataset Name	Description	Log Category
Top-50-Websites-By-Bandwidth	Webfilter top allowed web sites by bandwidth usage	traffic

```
select
  domain,
  string_agg(
    distinct catdesc,
    & #039;, ') as agg_catdesc, sum(bandwidth) as bandwidth, sum(traffic_in) as traffic_in, sum
(traffic_out) as traffic_out from ###(select coalesce(nullifna(hostname), ipstr(`dstip`)) as
domain, catdesc, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce
(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out from $log where $filter
and (logflag&1>0) and utmaction!='blocked' and (countweb>0 or ((logver is null or
logver<502000000) and (hostname is not null or utmevent in ('webfilter', 'banned-word', 'web-
content', 'command-block', 'script-filter')))) group by domain, catdesc having sum(coalesce
(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 /*SkipSTART*/order by bandwidth desc/*SkipEND*/)### t group
by domain, catdesc order by bandwidth desc
```

Dataset Name	Description	Log Category
Top-Blocked-Websites	UTM top blocked web sites by request	traffic

```
select
  hostname,
  count(*) as requests
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
  and utmevent in (
    & #039;webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') and
hostname is not null and (utmaction in ('block', 'blocked') or action='deny') group by hostname
order by requests desc
```

Dataset Name	Description	Log Category
Top-Web-Users-By-Request	UTM top web users by request	traffic

```
select
  user_src,
  devtype_new,
  srcname,
  sum(requests) as requests
from
  ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, get_
devtype(srcswversion, osname, devtype) as devtype_new, srcname, action, utmaction, sum(coalesce
(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentbyte, 0)) as traffic_out, count(*) as requests from $log where $filter and
(logflag&1>0) and utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block',
'script-filter') group by user_src, devtype_new, srcname, action, utmaction order by requests
desc)### t group by user_src, devtype_new, srcname order by requests desc
```

Dataset Name	Description	Log Category
Top-Allowed-WebSites-By-Bandwidth	UTM top allowed websites by bandwidth usage	traffic

```
select
  appid,
  hostname,
  catdesc,
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  ) as bandwidth,
  sum(
    coalesce(rcvdbyte, 0)
  ) as traffic_in,
  sum(
    coalesce(sentbyte, 0)
  ) as traffic_out
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
  and utmevent in (
    & #039;webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') and
hostname is not null group by appid, hostname, catdesc having sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0))>0 order by bandwidth desc
```

Dataset Name	Description	Log Category
Top-Blocked-Web-Users	UTM top blocked web users	traffic

```
select
  user_src,
  devtype_new,
  srcname,
  sum(requests) as requests
from
  ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, get_
devtype(srcswversion, osname, devtype) as devtype_new, srcname, action, utmaction, sum(coalesce
(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentbyte, 0)) as traffic_out, count(*) as requests from $log where $filter and
(logflag&1>0) and utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block',
'script-filter') group by user_src, devtype_new, srcname, action, utmaction order by requests
desc)### t where (utmaction in ('block', 'blocked') or action='deny') group by user_src, devtype_
new, srcname order by requests desc
```

Dataset Name	Description	Log Category
Top-20-Web-Users-By-Bandwidth	Webfilter top web users by bandwidth usage	traffic

```
select
  coalesce(
    f_user,
    euname,
    ipstr(`srcip`)
  ) as user_src,
  coalesce(
    epname,
    ipstr(`srcip`)
  ) as ep_src,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out
from
  (
    select
      dvid,
      f_user,
      srcip,
      ep_id,
      eu_id,
      sum(bandwidth) as bandwidth,
      sum(traffic_in) as traffic_in,
      sum(traffic_out) as traffic_out
    from
      ###(select dvid, coalesce(nullifna(`user`), nullifna(`unauthuser`)) as f_user, srcip, (case
when epid<1024 then null else epid end) as ep_id, (case when euid<1024 then null else euid end) as
eu_id, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte, 0))
as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out from $log where $filter and (logflag&1>0)
and (countweb>0 or ((logver is null or logver<502000000) and (hostname is not null or utmevent in
('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter')))) group by dvid, f_
user, srcip, ep_id, eu_id having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0
```

```
/*SkipSTART*/order by bandwidth desc/*SkipEND*/)### t group by dvid, f_user, srcip, ep_id, eu_id
order by bandwidth desc) t1 left join (select epid, euid, srcmac as epmac, dvid from $ADOM_EPEU_
DEVMAP dm inner join devtable dt ON dm.devid=dt.devid and dm.vd=dt.vd) t2 on t1.ep_id=t2.epid and
t1.eu_id=t2.euid and t1.dvid=t2.dvid left join $ADOM_ENDPOINT t3 on t1.ep_id=t3.epid and
t2.epmac=t3.mac and left(t3.epdevtype, 6)!='Router' left join $ADOM_ENDUSER t4 on t1.eu_id=t4.euid
group by user_src, ep_src order by bandwidth desc
```

Dataset Name	Description	Log Category
Top-Web-Users-By-Bandwidth	UTM top web users by bandwidth usage	traffic

```
select
  user_src,
  devtype_new,
  srcname,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out
from
  ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, get_
devtype(srcswversion, osname, devtype) as devtype_new, srcname, action, utmaction, sum(coalesce
(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentbyte, 0)) as traffic_out, count(*) as requests from $log where $filter and
(logflag&1>0) and utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block',
'script-filter') group by user_src, devtype_new, srcname, action, utmaction order by requests
desc)### t group by user_src, devtype_new, srcname having sum(bandwidth)>0 order by bandwidth desc
```

Dataset Name	Description	Log Category
Top-Video-Streaming-Websites-By-Bandwidth	UTM top video streaming websites by bandwidth usage	traffic

```
select
  appid,
  hostname,
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  ) as bandwidth,
  sum(
    coalesce(rcvdbyte, 0)
  ) as traffic_in,
  sum(
    coalesce(sentbyte, 0)
  ) as traffic_out
from
  $log
where
  $filter
  and (
```

```
    logflag&1>0
  )
  and catdesc in (
    & #039;Streaming Media and Download') group by appid, hostname having sum(coalesce(sentbyte,
0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc
```

Dataset Name	Description	Log Category
Top-Email-Senders-By-Count	Default top email senders by count	traffic

```
select
  user_src,
  sum(requests) as requests
from
  ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
service, count(*) as requests from $log where $filter and (logflag&1>0) group by user_src, service
order by requests desc)### t where service in ('smtp', 'SMTP', 'tcp/25', 'tcp/587', 'smtps',
'SMTPS', 'tcp/465') group by user_src order by requests desc
```

Dataset Name	Description	Log Category
Top-Email-Receivers-By-Count	Default email top receivers by count	traffic

```
select
  user_src,
  sum(requests) as requests
from
  ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
service, count(*) as requests from $log where $filter and (logflag&1>0) group by user_src, service
order by requests desc)### t where service in ('pop3', 'POP3', 'tcp/110', 'imap', 'IMAP',
'tcp/143', 'imaps', 'IMAPS', 'tcp/993', 'pop3s', 'POP3S', 'tcp/995') group by user_src order by
requests desc
```

Dataset Name	Description	Log Category
Top-Email-Senders-By-Bandwidth	Default email top senders by bandwidth usage	traffic

```
select
  coalesce(
    nullifna(`user`),
    nullifna(`unauthuser`),
    ipstr(`srcip`)
  ) as user_src,
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  ) as bandwidth
from
  $log
```

```
where
  $filter
  and (
    logflag&1>0
  )
  and service in (
    & #039;smtp', 'SMTP', 'tcp/25', 'tcp/587', 'smtps', 'SMTPS', 'tcp/465') group by user_src
having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc
```

Dataset Name	Description	Log Category
Top-Email-Receivers-By-Bandwidth	Default email top receivers by bandwidth usage	traffic

```
select
  coalesce(
    nullifna(`user`),
    nullifna(`unauthuser`),
    ipstr(`srcip`)
  ) as user_src,
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  ) as bandwidth
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
  and service in (
    & #039;pop3', 'POP3', 'tcp/110', 'imap', 'IMAP', 'tcp/143', 'imaps', 'IMAPS', 'tcp/993',
'pop3s', 'POP3S', 'tcp/995') group by user_src having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte,
0))>0 order by bandwidth desc
```

Dataset Name	Description	Log Category
Top-Malware-By-Name	UTM top virus	virus

```
select
  virus,
  max(virusid_s) as virusid,
  (
    case when virus like & #039;Riskware%' then 'Spyware' when virus like 'Adware%' then 'Adware'
else 'Virus' end) as malware_type, sum(totalnum) as totalnum from ###(select virus, virusid_to_str
(virusid, eventtype) as virusid_s, count(*) as totalnum from $log where $filter and nullifna
(virus) is not null group by virus, virusid_s /*SkipSTART*/order by totalnum desc/*SkipEND*/)### t
group by virus, malware_type order by totalnum desc
```

Dataset Name	Description	Log Category
Top-Virus-By-Name	UTM top virus	virus

```
select
  virus,
  max(virusid_s) as virusid,
  (
    case when virus like & #039;Riskware%' then 'Spyware' when virus like 'Adware%' then 'Adware'
else 'Virus' end) as malware_type, sum(totalnum) as totalnum from ###(select virus, virusid_to_str
(virusid, eventtype) as virusid_s, count(*) as totalnum from $log where $filter and nullifna
(virus) is not null group by virus, virusid_s /*SkipSTART*/order by totalnum desc/*SkipEND*/)### t
group by virus, malware_type order by totalnum desc
```

Dataset Name	Description	Log Category
Top-Virus-Victim	UTM top virus user	virus

```
select
  user_src,
  sum(totalnum) as totalnum
from
  ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, eventtype, logver, virus,
count(*) as totalnum from $log where $filter group by user_src, eventtype, logver, virus
/*SkipSTART*/order by totalnum desc/*SkipEND*/)### t where nullifna(virus) is not null group by
user_src order by totalnum desc
```

Dataset Name	Description	Log Category
Top-Attack-Source	UTM top attack source	attack

```
select
  user_src,
  sum(totalnum) as totalnum
from
  ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, eventtype, logver, count(*)
as totalnum from $log where $filter group by user_src, eventtype, logver /*SkipSTART*/order by
totalnum desc/*SkipEND*/)### t group by user_src order by totalnum desc
```

Dataset Name	Description	Log Category
Top-Attack-Victim	UTM top attack dest	attack

```
select
  victim,
  count(*) as totalnum
from
```

```
  (
    select
      (
        CASE WHEN direction =& #039;incoming' THEN srcip ELSE dstip END) as victim from $log where
$filter) t where victim is not null group by victim order by totalnum desc
```

Dataset Name	Description	Log Category
Top-Static-IPSEC-Tunnels-By-Bandwidth	Top static IPsec tunnels by bandwidth usage	event

```
select
  vpn_name,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out
from
  ###(select devid, vd, remip, vpn_trim(vpntunnel) as vpn_name, tunnelid, tunnelip, coalesce(sum
(case when sentdelta>0 then sentdelta else 0 end), 0) AS traffic_out, coalesce(sum(case when
rcvddelta>0 then rcvddelta else 0 end), 0) AS traffic_in, coalesce(sum(case when rcvddelta>0 and
sentdelta>0 then rcvddelta+sentdelta when rcvddelta>0 then rcvddelta when sentdelta>0 then
sentdelta else 0 end), 0) AS bandwidth from (select devid, vd, remip, tunnelid,
tunnelip,vpntunnel, tunneltype, action, sentbyte-lag(coalesce(sentbyte, 0)) OVER (PARTITION BY
dvid, tunnelid ORDER BY eventtime) AS sentdelta, rcvdbyte-lag(coalesce(rcvdbyte, 0)) OVER
(PARTITION BY dvid, tunnelid ORDER BY eventtime) AS rcvddelta from $log where $filter and
subtype='vpn' and action in ('tunnel-up','tunnel-stats', 'tunnel-down', 'ssl-login-fail', 'ipsec-
login-fail') and tunnelid is not null) t where tunneltype like 'ipsec%' and nullifna(vpntunnel) is
not null and tunnelid!=0 group by devid, vd, remip, vpn_name, tunnelid, tunnelip order by
bandwidth desc)### t where (tunnelip is null or tunnelip='0.0.0.0') group by vpn_name having sum
(bandwidth)>0 order by bandwidth desc
```

Dataset Name	Description	Log Category
Top-SSL-VPN-Tunnel-Users-By-Bandwidth	Top SSL VPN tunnel users by bandwidth usage	event

```
select
  user_src,
  remip as remote_ip,
  from_dtime(
    min(s_time)
  ) as start_time,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out
from
  ###(select devid, vd, remip, coalesce(nullifna(`user`), ipstr(`remip`)) as user_src, tunnelid,
tunneltype, max(coalesce(duration,0)) as max_duration, min(coalesce(duration,0)) as min_duration,
min(coalesce(dtime, 0)) as s_time, max(coalesce(dtime, 0)) as e_time, coalesce(sum(case when
```

```
sentdelta>0 then sentdelta else 0 end), 0) AS traffic_out, coalesce(sum(case when rcvddelta>0 then
rcvddelta else 0 end), 0) AS traffic_in, coalesce(sum(case when rcvddelta>0 and sentdelta>0 then
rcvddelta+sentdelta when rcvddelta>0 then rcvddelta when sentdelta>0 then sentdelta else 0 end),
0) AS bandwidth from (select dtime, devid, vd, remip, `user`, duration, tunnelid, tunneltype,
sentbyte-lag(coalesce(sentbyte, 0)) OVER (PARTITION BY dvid, tunnelid ORDER BY eventtime) AS
sentdelta, rcvdbyte-lag(coalesce(rcvdbyte, 0)) OVER (PARTITION BY dvid, tunnelid ORDER BY
eventtime) AS rcvddelta from $log where $filter and subtype='vpn' and action in ('tunnel-
up','tunnel-stats', 'tunnel-down', 'ssl-login-fail', 'ipsec-login-fail') and tunnelid is not null
and tunnelid!=0) t where tunneltype like 'ssl%' and coalesce(nullifna(`user`), ipstr(`remip`)) is
not null group by devid, vd, user_src, remip, tunnelid, tunneltype order by bandwidth desc)### t
where tunneltype='ssl-tunnel' group by user_src, remote_ip having sum(bandwidth)>0 order by
bandwidth desc
```

Dataset Name	Description	Log Category
Top-Dial-Up-IPSEC-Tunnels-By-Bandwidth	Top dial up IPsec tunnels by bandwidth usage	event

```
select
  vpn_name,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out
from
  ###(select devid, vd, remip, vpn_trim(vpntunnel) as vpn_name, tunnelid, tunnelip, coalesce(sum
(case when sentdelta>0 then sentdelta else 0 end), 0) AS traffic_out, coalesce(sum(case when
rcvddelta>0 then rcvddelta else 0 end), 0) AS traffic_in, coalesce(sum(case when rcvddelta>0 and
sentdelta>0 then rcvddelta+sentdelta when rcvddelta>0 then rcvddelta when sentdelta>0 then
sentdelta else 0 end), 0) AS bandwidth from (select devid, vd, remip, tunnelid,
tunnelip,vpntunnel, tunneltype, action, sentbyte-lag(coalesce(sentbyte, 0)) OVER (PARTITION BY
dvid, tunnelid ORDER BY eventtime) AS sentdelta, rcvdbyte-lag(coalesce(rcvdbyte, 0)) OVER
(PARTITION BY dvid, tunnelid ORDER BY eventtime) AS rcvddelta from $log where $filter and
subtype='vpn' and action in ('tunnel-up','tunnel-stats', 'tunnel-down', 'ssl-login-fail', 'ipsec-
login-fail') and tunnelid is not null) t where tunneltype like 'ipsec%' and nullifna(vpntunnel) is
not null and tunnelid!=0 group by devid, vd, remip, vpn_name, tunnelid, tunnelip order by
bandwidth desc)### t where not (tunnelip is null or tunnelip='0.0.0.0') group by vpn_name having
sum(bandwidth)>0 order by bandwidth desc
```

Dataset Name	Description	Log Category
Top-Dial-Up-IPSEC-Users-By-Bandwidth	Top dial up IPsec users by bandwidth usage	event

```
select
  coalesce(
    xauthuser_agg,
    user_agg,
    ipstr(`remip`)
  ) as user_src,
```

```
  remip,
  from_dtime(
    min(s_time)
  ) as start_time,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out
from
  (
    select
      devid,
      vd,
      string_agg(
        distinct xauthuser_agg,
        & #039; ') as xauthuser_agg, string_agg(distinct user_agg, ' ') as user_agg, remip,
tunnelid, min(s_time) as s_time, max(e_time) as e_time, sum(bandwidth) as bandwidth, sum(traffic_
in) as traffic_in, sum(traffic_out) as traffic_out from ###(select devid, vd, remip, nullifna
(`xauthuser`) as xauthuser_agg, nullifna(`user`) as user_agg, tunnelid, max(coalesce(duration,0))
as max_duration, min(coalesce(duration,0)) as min_duration, min(coalesce(dtime, 0)) as s_time, max
(coalesce(dtime, 0)) as e_time, coalesce(sum(case when sentdelta>0 then sentdelta else 0 end), 0)
AS traffic_out, coalesce(sum(case when rcvddelta>0 then rcvddelta else 0 end), 0) AS traffic_in,
coalesce(sum(case when rcvddelta>0 and sentdelta>0 then rcvddelta+sentdelta when rcvddelta>0 then
rcvddelta when sentdelta>0 then sentdelta else 0 end), 0) AS bandwidth from (select dtime, devid,
vd, remip, `xauthuser`, `user`, duration, tunnelid, tunnelip, tunneltype, sentbyte-lag(coalesce
(sentbyte, 0)) OVER (PARTITION BY dvid, tunnelid ORDER BY eventtime) AS sentdelta, rcvdbyte-lag
(coalesce(rcvdbyte, 0)) OVER (PARTITION BY dvid, tunnelid ORDER BY eventtime) AS rcvddelta from
$log where $filter and subtype='vpn' and action in ('tunnel-up','tunnel-stats', 'tunnel-down',
'ssl-login-fail', 'ipsec-login-fail') and tunnelid is not null) t where tunneltype like 'ipsec%'
and not (tunnelip is null or tunnelip='0.0.0.0') and tunnelid!=0 group by devid, vd, remip,
xauthuser_agg, user_agg, tunnelid order by bandwidth desc)### t group by devid, vd, remip,
tunnelid) tt where bandwidth>0 group by user_src, remip order by bandwidth desc
```

Dataset Name	Description	Log Category
Top-Dial-Up-IPSEC-Users-By-Duration	Top dial up IPsec users by duration	event

```
select
  coalesce(
    xauthuser_agg,
    user_agg,
    ipstr(`remip`)
  ) as user_src,
  from_dtime(
    min(s_time)
  ) as start_time,
  sum(duration) as duration,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out
from
```

```
  (
    select
      devid,
      vd,
      remip,
      string_agg(
        distinct xauthuser_agg,
        & #039; ') as xauthuser_agg, string_agg(distinct user_agg, ' ') as user_agg, tunnelid, min
(s_time) as s_time, max(e_time) as e_time, (case when min(s_time)=max(e_time) then max(max_
duration) else max(max_duration)-min(min_duration) end) as duration, sum(bandwidth) as bandwidth,
sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_out from ###(select devid, vd, remip,
nullifna(`xauthuser`) as xauthuser_agg, nullifna(`user`) as user_agg, tunnelid, max(coalesce
(duration,0)) as max_duration, min(coalesce(duration,0)) as min_duration, min(coalesce(dtime, 0))
as s_time, max(coalesce(dtime, 0)) as e_time, coalesce(sum(case when sentdelta>0 then sentdelta
else 0 end), 0) AS traffic_out, coalesce(sum(case when rcvddelta>0 then rcvddelta else 0 end), 0)
AS traffic_in, coalesce(sum(case when rcvddelta>0 and sentdelta>0 then rcvddelta+sentdelta when
rcvddelta>0 then rcvddelta when sentdelta>0 then sentdelta else 0 end), 0) AS bandwidth from
(select dtime, devid, vd, remip, `xauthuser`, `user`, duration, tunnelid, tunnelip, tunneltype,
sentbyte-lag(coalesce(sentbyte, 0)) OVER (PARTITION BY dvid, tunnelid ORDER BY eventtime) AS
sentdelta, rcvdbyte-lag(coalesce(rcvdbyte, 0)) OVER (PARTITION BY dvid, tunnelid ORDER BY
eventtime) AS rcvddelta from $log where $filter and subtype='vpn' and action in ('tunnel-
up','tunnel-stats', 'tunnel-down', 'ssl-login-fail', 'ipsec-login-fail') and tunnelid is not null)
t where tunneltype like 'ipsec%' and not (tunnelip is null or tunnelip='0.0.0.0') and tunnelid!=0
group by devid, vd, remip, xauthuser_agg, user_agg, tunnelid order by bandwidth desc)### t group
by devid, vd, remip, tunnelid) tt where bandwidth>0 group by user_src order by duration desc
```

Dataset Name	Description	Log Category
Top-SSL-VPN-Web-Mode-Users-By-Bandwidth	Top SSL VPN web mode users by bandwidth usage	event

```
select
  user_src,
  remip as remote_ip,
  from_dtime(
    min(s_time)
  ) as start_time,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out
from
  ###(select devid, vd, remip, coalesce(nullifna(`user`), ipstr(`remip`)) as user_src, tunnelid,
tunneltype, max(coalesce(duration,0)) as max_duration, min(coalesce(duration,0)) as min_duration,
min(coalesce(dtime, 0)) as s_time, max(coalesce(dtime, 0)) as e_time, coalesce(sum(case when
sentdelta>0 then sentdelta else 0 end), 0) AS traffic_out, coalesce(sum(case when rcvddelta>0 then
rcvddelta else 0 end), 0) AS traffic_in, coalesce(sum(case when rcvddelta>0 and sentdelta>0 then
rcvddelta+sentdelta when rcvddelta>0 then rcvddelta when sentdelta>0 then sentdelta else 0 end),
0) AS bandwidth from (select dtime, devid, vd, remip, `user`, duration, tunnelid, tunneltype,
sentbyte-lag(coalesce(sentbyte, 0)) OVER (PARTITION BY dvid, tunnelid ORDER BY eventtime) AS
sentdelta, rcvdbyte-lag(coalesce(rcvdbyte, 0)) OVER (PARTITION BY dvid, tunnelid ORDER BY
eventtime) AS rcvddelta from $log where $filter and subtype='vpn' and action in ('tunnel-
```

```
up','tunnel-stats', 'tunnel-down', 'ssl-login-fail', 'ipsec-login-fail') and tunnelid is not null
and tunnelid!=0) t where tunneltype like 'ssl%' and coalesce(nullifna(`user`), ipstr(`remip`)) is
not null group by devid, vd, user_src, remip, tunnelid, tunneltype order by bandwidth desc)### t
group by user_src, remote_ip having sum(bandwidth)>0 order by bandwidth desc
```

Dataset Name	Description	Log Category
Top-SSL-VPN-Web-Mode-Users-By-Duration	Top SSL VPN web mode users by duration	event

```
select
  user_src,
  remip as remote_ip,
  from_dtime(
    min(s_time)
  ) as start_time,
  sum(duration) as duration
from
  (
    select
      devid,
      vd,
      user_src,
      remip,
      tunnelid,
      min(s_time) as s_time,
      (
        case when min(s_time)= max(e_time) then max(max_duration) else max(max_duration)- min(min_
duration) end
      ) as duration
    from
      ###(select devid, vd, remip, coalesce(nullifna(`user`), ipstr(`remip`)) as user_src,
tunnelid, tunneltype, max(coalesce(duration,0)) as max_duration, min(coalesce(duration,0)) as min_
duration, min(coalesce(dtime, 0)) as s_time, max(coalesce(dtime, 0)) as e_time, coalesce(sum(case
when sentdelta>0 then sentdelta else 0 end), 0) AS traffic_out, coalesce(sum(case when rcvddelta>0
then rcvddelta else 0 end), 0) AS traffic_in, coalesce(sum(case when rcvddelta>0 and sentdelta>0
then rcvddelta+sentdelta when rcvddelta>0 then rcvddelta when sentdelta>0 then sentdelta else 0
end), 0) AS bandwidth from (select dtime, devid, vd, remip, `user`, duration, tunnelid,
tunneltype, sentbyte-lag(coalesce(sentbyte, 0)) OVER (PARTITION BY dvid, tunnelid ORDER BY
eventtime) AS sentdelta, rcvdbyte-lag(coalesce(rcvdbyte, 0)) OVER (PARTITION BY dvid, tunnelid
ORDER BY eventtime) AS rcvddelta from $log where $filter and subtype='vpn' and action in ('tunnel-
up','tunnel-stats', 'tunnel-down', 'ssl-login-fail', 'ipsec-login-fail') and tunnelid is not null
and tunnelid!=0) t where tunneltype like 'ssl%' and coalesce(nullifna(`user`), ipstr(`remip`)) is
not null group by devid, vd, user_src, remip, tunnelid, tunneltype order by bandwidth desc)### t
where tunneltype='ssl-web' group by devid, vd, user_src, remip, tunnelid) tt group by user_src,
remote_ip order by duration desc
```

Dataset Name	Description	Log Category
Top-SSL-VPN-Users-By-Duration	Top SSL VPN users by duration	event

```
select
  user_src,
  tunneltype,
  sum(duration) as duration,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out
from
  (
    select
      devid,
      vd,
      remip,
      user_src,
      tunneltype,
      tunnelid,
      (
        case when min(s_time)= max(e_time) then max(max_duration) else max(max_duration)- min(min_
duration) end
      ) as duration,
      sum(traffic_in) as traffic_in,
      sum(traffic_out) as traffic_out,
      sum(bandwidth) as bandwidth
    from
      ###(select devid, vd, remip, coalesce(nullifna(`user`), ipstr(`remip`)) as user_src,
tunnelid, tunneltype, max(coalesce(duration,0)) as max_duration, min(coalesce(duration,0)) as min_
duration, min(coalesce(dtime, 0)) as s_time, max(coalesce(dtime, 0)) as e_time, coalesce(sum(case
when sentdelta>0 then sentdelta else 0 end), 0) AS traffic_out, coalesce(sum(case when rcvddelta>0
then rcvddelta else 0 end), 0) AS traffic_in, coalesce(sum(case when rcvddelta>0 and sentdelta>0
then rcvddelta+sentdelta when rcvddelta>0 then rcvddelta when sentdelta>0 then sentdelta else 0
end), 0) AS bandwidth from (select dtime, devid, vd, remip, `user`, duration, tunnelid,
tunneltype, sentbyte-lag(coalesce(sentbyte, 0)) OVER (PARTITION BY dvid, tunnelid ORDER BY
eventtime) AS sentdelta, rcvdbyte-lag(coalesce(rcvdbyte, 0)) OVER (PARTITION BY dvid, tunnelid
ORDER BY eventtime) AS rcvddelta from $log where $filter and subtype='vpn' and action in ('tunnel-
up','tunnel-stats', 'tunnel-down', 'ssl-login-fail', 'ipsec-login-fail') and tunnelid is not null
and tunnelid!=0) t where tunneltype like 'ssl%' and coalesce(nullifna(`user`), ipstr(`remip`)) is
not null group by devid, vd, user_src, remip, tunnelid, tunneltype order by bandwidth desc)### t
group by devid, vd, remip, user_src, tunnelid, tunneltype) tt where bandwidth>0 group by user_src,
tunneltype order by duration desc
```

Dataset Name	Description	Log Category
vpn-User-Login-history	VPN user login history	event

```
select
  $flex_timescale(timestamp) as hodex,
  sum(tunnelup) as total_num
from
  (
    select
      timestamp,
```

```
      devid,
      vd,
      remip,
      tunnelid,
      max(tunnelup) as tunnelup,
      max(traffic_in) as traffic_in,
      max(traffic_out) as traffic_out
    from
      ###(select $flex_timestamp as timestamp, devid, vd, remip, tunnelid, max((case when
action='tunnel-up' then 1 else 0 end)) as tunnelup, max(coalesce(sentbyte, 0)) as traffic_out, max
(coalesce(rcvdbyte, 0)) as traffic_in from $log where $filter and subtype='vpn' and (tunneltype
like 'ipsec%' or tunneltype like 'ssl%') and action in ('tunnel-up', 'tunnel-stats', 'tunnel-
down') and tunnelid is not null group by timestamp, devid, vd, remip, tunnelid /*SkipSTART*/order
by timestamp desc/*SkipEND*/)### t group by timestamp, devid, vd, remip, tunnelid having max
(traffic_in)+max(traffic_out)>0) t group by hodex order by total_num desc
```

Dataset Name	Description	Log Category
vpn-Failed-Login-Atempts	VPN failed logins	event

```
select
  f_user,
  tunneltype,
  sum(total_num) as total_num
from
  ###(select coalesce(nullifna(`xauthuser`), `user`) as f_user, tunneltype, count(*) as total_num
from $log where $filter and subtype='vpn' and (tunneltype like 'ipsec%' or tunneltype like 'ssl%')
and action in ('ssl-login-fail', 'ipsec-login-fail') and coalesce(nullifna(`xauthuser`), nullifna
(`user`)) is not null group by f_user, tunneltype order by total_num desc)### t group by f_user,
tunneltype order by total_num desc
```

Dataset Name	Description	Log Category
vpn-Traffic-Usage-Trend-VPN-Summary	VPN traffic usage trend	event

```
select
  hodex,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out,
  sum(ssl_traffic_bandwidth) as ssl_bandwidth,
  sum(ipsec_traffic_bandwidth) as ipsec_bandwidth
from
  (
    select
      $flex_timescale(timestamp) as hodex,
      devid,
      vd,
      remip,
```

```
      tunnelid,
      sum(traffic_in) as traffic_in,
      sum(traffic_out) as traffic_out,
      (
        case when t_type like & #039;ssl%' then sum(bandwidth) else 0 end) as ssl_traffic_
bandwidth, (case when t_type like 'ipsec%' then sum(bandwidth)  else 0 end) as ipsec_traffic_
bandwidth, min(s_time) as s_time, max(e_time) as e_time from ###(select $flex_timestamp as
timestamp, devid, vd, remip, tunnelid, (case when tunneltype like 'ipsec%' then 'ipsec' else
tunneltype end) as t_type, (case when action='tunnel-up' then 1 else 0 end) as tunnelup, coalesce
(sum(case when sentdelta>0 then sentdelta else 0 end), 0) AS traffic_out, coalesce(sum(case when
rcvddelta>0 then rcvddelta else 0 end), 0) AS traffic_in, coalesce(sum(case when rcvddelta>0 and
sentdelta>0 then rcvddelta+sentdelta when rcvddelta>0 then rcvddelta when sentdelta>0 then
sentdelta else 0 end), 0) AS bandwidth, min(coalesce(dtime, 0)) as s_time, max(coalesce(dtime, 0))
as e_time, coalesce(nullifna(`xauthuser`), nullifna(`user`), ipstr(`remip`)) as f_user,
tunneltype, action, count(*) as total_num  from (select itime, dtime, devid, vd, remip,
`xauthuser`, `user`, duration, tunnelid, tunnelip, tunneltype, action, sentbyte-lag(coalesce
(sentbyte, 0)) OVER (PARTITION BY dvid, tunnelid ORDER BY eventtime) AS sentdelta, rcvdbyte-lag
(coalesce(rcvdbyte, 0)) OVER (PARTITION BY dvid, tunnelid ORDER BY eventtime) AS rcvddelta from
$log where $filter and subtype='vpn' and tunnelid is not null) t where (tunneltype like 'ipsec%'
or tunneltype like 'ssl%') and action in ('tunnel-up','tunnel-stats', 'tunnel-down', 'ssl-login-
fail', 'ipsec-login-fail') group by timestamp, devid, vd, remip, t_type, tunnelid, action, f_user,
tunneltype /*SkipSTART*/order by bandwidth desc/*SkipEND*/)### t where action in ('tunnel-
up','tunnel-stats', 'tunnel-down') and tunnelid is not null and tunnelid!=0 group by hodex, devid,
t_type, vd, remip, tunnelid) tt group by hodex order by hodex
```

Dataset Name	Description	Log Category
Top-S2S-IPSEC-Tunnels-By-Bandwidth-and-Availability	Top S2S IPsec tunnels by bandwidth usage and avail	event

```
select
  vpntunnel,
  tunneltype,
  sum(traffic_out) as traffic_out,
  sum(traffic_in) as traffic_in,
  sum(bandwidth) as bandwidth,
  sum(uptime) as uptime
from
  (
    select
      vpntunnel,
      tunneltype,
      tunnelid,
      devid,
      vd,
      sum(sent_end - sent_beg) as traffic_out,
      sum(rcvd_end - rcvd_beg) as traffic_in,
      sum(
        sent_end - sent_beg + rcvd_end - rcvd_beg
      ) as bandwidth,
      sum(duration_end - duration_beg) as uptime
```

```
    from
      ###(select tunnelid, tunneltype, vpntunnel, devid, vd, min(coalesce(sentbyte, 0)) as sent_
beg, max(coalesce(sentbyte, 0)) as sent_end, min(coalesce(rcvdbyte, 0)) as rcvd_beg, max(coalesce
(rcvdbyte, 0)) as rcvd_end, min(coalesce(duration, 0)) as duration_beg, max(coalesce(duration, 0))
as duration_end from $log where $filter and subtype='vpn' and action='tunnel-stats' and tunneltype
like 'ipsec%' and (tunnelip is null or tunnelip='0.0.0.0') and nullifna(`user`) is null and
tunnelid is not null and tunnelid!=0 group by tunnelid, tunneltype, vpntunnel, devid, vd
/*SkipSTART*/order by tunnelid/*SkipEND*/)### t group by vpntunnel, tunneltype, tunnelid, devid,
vd order by bandwidth desc) t where bandwidth>0 group by vpntunnel, tunneltype order by bandwidth
desc
```

Dataset Name	Description	Log Category
Top-Dialup-IPSEC-By-Bandwidth-and-Availability	Top dialup IPsec users by bandwidth usage and avail	event

```
select
  user_src,
  remip,
  sum(traffic_out) as traffic_out,
  sum(traffic_in) as traffic_in,
  sum(bandwidth) as bandwidth,
  sum(uptime) as uptime
from
  (
    select
      user_src,
      remip,
      tunnelid,
      devid,
      vd,
      sum(sent_end - sent_beg) as traffic_out,
      sum(rcvd_end - rcvd_beg) as traffic_in,
      sum(
        sent_end - sent_beg + rcvd_end - rcvd_beg
      ) as bandwidth,
      sum(duration_end - duration_beg) as uptime
    from
      ###(select tunnelid, coalesce(nullifna(`xauthuser`), nullifna(`user`), ipstr(`remip`)) as
user_src, remip, devid, vd, min(coalesce(sentbyte, 0)) as sent_beg, max(coalesce(sentbyte, 0)) as
sent_end, min(coalesce(rcvdbyte, 0)) as rcvd_beg, max(coalesce(rcvdbyte, 0)) as rcvd_end, min
(coalesce(duration, 0)) as duration_beg, max(coalesce(duration, 0)) as duration_end from $log
where $filter and subtype='vpn' and action='tunnel-stats' and tunneltype like 'ipsec%' and not
(tunnelip is null or tunnelip='0.0.0.0') and tunnelid is not null and tunnelid!=0 group by
tunnelid, user_src, remip, devid, vd /*SkipSTART*/order by tunnelid/*SkipEND*/)### t group by
user_src, remip, tunnelid, devid, vd order by bandwidth desc) t where bandwidth>0 group by user_
src, remip order by bandwidth desc
```

Dataset Name	Description	Log Category
Top-SSL-Tunnel-Mode-By-Bandwidth-and-Availability	Top SSL tunnel users by bandwidth usage and avail	event

```
select
  user_src,
  remote_ip,
  sum(traffic_out) as traffic_out,
  sum(traffic_in) as traffic_in,
  sum(bandwidth) as bandwidth,
  sum(uptime) as uptime
from
  (
    select
      user_src,
      remip as remote_ip,
      tunnelid,
      devid,
      vd,
      sum(sent_end - sent_beg) as traffic_out,
      sum(rcvd_end - rcvd_beg) as traffic_in,
      sum(
        sent_end - sent_beg + rcvd_end - rcvd_beg
      ) as bandwidth,
      sum(duration_end - duration_beg) as uptime
    from
      ###(select tunnelid, tunneltype, coalesce(nullifna(`user`), ipstr(`remip`)) as user_src,
remip, devid, vd, min(coalesce(sentbyte, 0)) as sent_beg, max(coalesce(sentbyte, 0)) as sent_end,
min(coalesce(rcvdbyte, 0)) as rcvd_beg, max(coalesce(rcvdbyte, 0)) as rcvd_end, min(coalesce
(duration, 0)) as duration_beg, max(coalesce(duration, 0)) as duration_end from $log where $filter
and subtype='vpn' and action='tunnel-stats' and coalesce(nullifna(`user`), ipstr(`remip`)) is not
null and tunnelid is not null group by tunnelid, tunneltype, user_src, remip, devid, vd
/*SkipSTART*/order by tunnelid/*SkipEND*/)### t where tunneltype in ('ssl-tunnel', 'ssl') group by
user_src, remote_ip, tunnelid, devid, vd order by bandwidth desc) t where bandwidth>0 group by
user_src, remote_ip order by bandwidth desc
```

Dataset Name	Description	Log Category
Top-SSL-Web-Mode-By-Bandwidth-and-Availability	Top SSL web users by bandwidth usage and avail	event

```
select
  user_src,
  remote_ip,
  sum(traffic_out) as traffic_out,
  sum(traffic_in) as traffic_in,
  sum(bandwidth) as bandwidth,
  sum(uptime) as uptime
from
```

```
  (
    select
      user_src,
      remip as remote_ip,
      tunnelid,
      devid,
      vd,
      sum(sent_end - sent_beg) as traffic_out,
      sum(rcvd_end - rcvd_beg) as traffic_in,
      sum(
        sent_end - sent_beg + rcvd_end - rcvd_beg
      ) as bandwidth,
      sum(duration_end - duration_beg) as uptime
    from
      ###(select tunnelid, tunneltype, coalesce(nullifna(`user`), ipstr(`remip`)) as user_src,
remip, devid, vd, min(coalesce(sentbyte, 0)) as sent_beg, max(coalesce(sentbyte, 0)) as sent_end,
min(coalesce(rcvdbyte, 0)) as rcvd_beg, max(coalesce(rcvdbyte, 0)) as rcvd_end, min(coalesce
(duration, 0)) as duration_beg, max(coalesce(duration, 0)) as duration_end from $log where $filter
and subtype='vpn' and action='tunnel-stats' and coalesce(nullifna(`user`), ipstr(`remip`)) is not
null and tunnelid is not null group by tunnelid, tunneltype, user_src, remip, devid, vd
/*SkipSTART*/order by tunnelid/*SkipEND*/)### t where tunneltype='ssl-web' group by user_src,
remote_ip, tunnelid, devid, vd having sum(sent_end-sent_beg+rcvd_end-rcvd_beg)>0 order by
bandwidth desc) t where bandwidth>0 group by user_src, remote_ip order by bandwidth desc
```

Dataset Name	Description	Log Category
Admin-Login-Summary	Event admin login summary	event

```
select
  f_user,
  ui,
  sum(login) as total_num,
  sum(login_duration) as total_duration,
  sum(config_change) as total_change
from
  (
    select
      `user` as f_user,
      ui,
      (
        case when logid_to_int(logid)= 32001 then 1 else 0 end
      ) as login,
      (
        case when logid_to_int(logid)= 32003 then duration else 0 end
      ) as login_duration,
      (
        case when logid_to_int(logid)= 32003
        and state is not null then 1 else 0 end
      ) as config_change
    from
      $log
```

```
    where
      $filter
      and nullifna(`user`) is not null
      and logid_to_int(logid) in (32001, 32003)
  ) t
group by
  f_user,
  ui
having
  sum(login)+ sum(config_change)> 0
order by
  total_num desc
```

Dataset Name	Description	Log Category
Admin-Login-Summary-By-Date	Event admin login summary by date	event

```
select
  $flex_timescale(timestamp) as dom,
  sum(total_num) as total_num,
  sum(total_change) as total_change
from
  ###(select timestamp, sum(login) as total_num, sum(config_change) as total_change from (select
$flex_timestamp as timestamp, (case when logid_to_int(logid)=32001 then 1 else 0 end) as login,
(case when logid_to_int(logid)=32003 and state is not null then 1 else 0 end) as config_change
from $log where $filter and logid_to_int(logid) in (32001, 32003)) t group by timestamp having sum
(login)+sum(config_change)>0 /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t group by dom
order by dom
```

Dataset Name	Description	Log Category
Admin-Failed-Login-Summary	Event admin failed login summary	event

```
select
  `user` as f_user,
  ui,
  count(status) as total_failed
from
  $log
where
  $filter
  and nullifna(`user`) is not null
  and logid_to_int(logid) = 32002
group by
  ui,
  f_user
order by
  total_failed desc
```

Dataset Name	Description	Log Category
System-Summary-By-Severity	Event system summary by severity	event

```
select
  severity_tmp as severity,
  sum(count) as total_num
from
  ###(select coalesce(nullifna(logdesc), msg) as msg_desc, (case when level in ('critical',
'alert', 'emergency') then 'Critical' when level='error' then 'High' when level='warning' then
'Medium' when level='notice' then 'Low' else 'Info' end) as severity_tmp, count(*) as count from
$log where $filter and subtype='system' group by msg_desc, severity_tmp /*SkipSTART*/order by
count desc/*SkipEND*/)### t group by severity order by total_num desc
```

Dataset Name	Description	Log Category
System-Summary-By-Date	Event system summary by date	event

```
select
  $flex_timescale(timestamp) as dom,
  sum(critical) as critical,
  sum(high) as high,
  sum(medium) as medium
from
  ###(select $flex_timestamp as timestamp, sum(case when level in ('critical', 'alert',
'emergency') then 1 else 0 end) as critical, sum(case when level = 'error' then 1 else 0 end) as
high, sum(case when level = 'warning' then 1 else 0 end) as medium from $log where $filter and
subtype='system' group by timestamp /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t group by
dom order by dom
```

Dataset Name	Description	Log Category
Important-System-Summary-By-Date	Event system summary by date	event

```
select
  $flex_timescale(timestamp) as dom,
  sum(critical) as critical,
  sum(high) as high,
  sum(medium) as medium
from
  ###(select $flex_timestamp as timestamp, sum(case when level in ('critical', 'alert',
'emergency') then 1 else 0 end) as critical, sum(case when level = 'error' then 1 else 0 end) as
high, sum(case when level = 'warning' then 1 else 0 end) as medium from $log where $filter and
subtype='system' group by timestamp /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t group by
dom order by dom
```

Dataset Name	Description	Log Category
System-Critical-Severity-Events	Event system critical severity events	event

```
select
  msg_desc as msg,
  severity_tmp as severity,
  sum(count) as counts
from
  ###(select coalesce(nullifna(logdesc), msg) as msg_desc, (case when level in ('critical',
'alert', 'emergency') then 'Critical' when level='error' then 'High' when level='warning' then
'Medium' when level='notice' then 'Low' else 'Info' end) as severity_tmp, count(*) as count from
$log where $filter and subtype='system' group by msg_desc, severity_tmp /*SkipSTART*/order by
count desc/*SkipEND*/)### t where severity_tmp='Critical' group by msg, severity_tmp order by
counts desc
```

Dataset Name	Description	Log Category
System-High-Severity-Events	Event system high severity events	event

```
select
  msg_desc as msg,
  severity_tmp as severity,
  sum(count) as counts
from
  ###(select coalesce(nullifna(logdesc), msg) as msg_desc, (case when level in ('critical',
'alert', 'emergency') then 'Critical' when level='error' then 'High' when level='warning' then
'Medium' when level='notice' then 'Low' else 'Info' end) as severity_tmp, count(*) as count from
$log where $filter and subtype='system' group by msg_desc, severity_tmp /*SkipSTART*/order by
count desc/*SkipEND*/)### t where severity_tmp='High' group by msg, severity_tmp order by counts
desc
```

Dataset Name	Description	Log Category
System-Medium-Severity-Events	Event system medium severity events	event

```
select
  msg_desc as msg,
  severity_tmp as severity,
  sum(count) as counts
from
  ###(select coalesce(nullifna(logdesc), msg) as msg_desc, (case when level in ('critical',
'alert', 'emergency') then 'Critical' when level='error' then 'High' when level='warning' then
'Medium' when level='notice' then 'Low' else 'Info' end) as severity_tmp, count(*) as count from
$log where $filter and subtype='system' group by msg_desc, severity_tmp /*SkipSTART*/order by
count desc/*SkipEND*/)### t where severity_tmp='Medium' group by msg, severity_tmp order by counts
desc
```

Dataset Name	Description	Log Category
utm-drilldown-Top-Traffic-Summary	UTM drilldown traffic summary	traffic

```
select
  srcip,
  srcname
from
  ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
srcip, srcname, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from $log where
$filter and (logflag&1>0) group by user_src, srcip, srcname order by bandwidth desc)### t where
$filter-drilldown group by srcip, srcname
```

Dataset Name	Description	Log Category
utm-drilldown-Top-User-Destination	UTM drilldown top user destination	traffic

```
select
  appid,
  app,
  dstip,
  sum(sessions) as sessions,
  sum(bandwidth) as bandwidth
from
  ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
appid, app, dstip, count(*) as sessions, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as
bandwidth from $log where $filter and (logflag&1>0) and dstip is not null and nullifna(app) is not
null group by user_src, appid, app, dstip having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte,
0))>0 order by bandwidth desc)### t where $filter-drilldown group by appid, app, dstip order by
bandwidth desc
```

Dataset Name	Description	Log Category
utm-drilldown-Email-Senders-Summary	UTM drilldown email senders summary	traffic

```
select
  sum(requests) as requests,
  sum(bandwidth) as bandwidth
from
  ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
sender, count(*) as requests, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from
$log where $filter and (logflag&1>0) and service in ('smtp', 'SMTP', 'tcp/25', 'tcp/587', 'smtps',
'SMTPS', 'tcp/465') group by user_src, sender order by requests desc)### t where $filter-drilldown
```

Dataset Name	Description	Log Category
utm-drilldown-Email-Receivers-Summary	UTM drilldown email receivers summary	traffic

```
select
  sum(requests) as requests,
  sum(bandwidth) as bandwidth
from
  ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
recipient, count(*) as requests, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth
from $log where $filter and (logflag&1>0) and recipient is not null and service in ('pop3',
'POP3', 'tcp/110', 'imap', 'IMAP', 'tcp/143', 'imaps', 'IMAPS', 'tcp/993', 'pop3s', 'POP3S',
'tcp/995') group by user_src, recipient order by requests desc)### t where $filter-drilldown
```

Dataset Name	Description	Log Category
utm-drilldown-Top-Email-Recipients-By-Bandwidth	UTM drilldown top email recipients	traffic

```
select
  recipient,
  sum(bandwidth) as bandwidth
from
  ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
recipient, count(*) as requests, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth
from $log where $filter and (logflag&1>0) and recipient is not null and service in ('pop3',
'POP3', 'tcp/110', 'imap', 'IMAP', 'tcp/143', 'imaps', 'IMAPS', 'tcp/993', 'pop3s', 'POP3S',
'tcp/995') group by user_src, recipient order by requests desc)### t where $filter-drilldown group
by recipient having sum(bandwidth)>0 order by bandwidth desc
```

Dataset Name	Description	Log Category
utm-drilldown-Top-Email-Senders-By-Bandwidth	UTM drilldown top email senders	traffic

```
select
  sender,
  sum(bandwidth) as bandwidth
from
  ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
sender, count(*) as requests, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from
$log where $filter and (logflag&1>0) and service in ('smtp', 'SMTP', 'tcp/25', 'tcp/587', 'smtps',
'SMTPS', 'tcp/465') group by user_src, sender order by requests desc)### t where $filter-drilldown
and sender is not null group by sender having sum(bandwidth)>0 order by bandwidth desc
```

Dataset Name	Description	Log Category
utm-drilldown-Top-Allowed-Websites-By-Bandwidth	UTM drilldown top allowed web sites by bandwidth	traffic

```
select
  appid,
  hostname,
  sum(bandwidth) as bandwidth
from
  ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
appid, hostname, (case when utmaction in ('block', 'blocked') then 1 else 0 end) as blocked, sum
(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from $log-traffic where $filter and
(logflag&1>0) and (countweb>0 or ((logver is null or logver<502000000) and (hostname is not null
or utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter'))))
and hostname is not null group by user_src, appid, hostname, blocked order by bandwidth desc)### t
where $filter-drilldown and blocked=0 group by appid, hostname order by bandwidth desc
```

Dataset Name	Description	Log Category
utm-drilldown-Top-Blocked-Websites-By-Request	UTM drilldown top blocked web sites by request	webfilter

```
select
  appid,
  hostname,
  sum(requests) as requests
from
  ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, 0 as appid, hostname, (case
when action='blocked' then 1 else 0 end) as blocked, count(*) as requests from $log where $filter
and hostname is not null group by user_src, appid, hostname, blocked order by requests desc)### t
where $filter-drilldown and blocked=1 group by appid, hostname order by requests desc
```

Dataset Name	Description	Log Category
utm-drilldown-Top-Virus-By-Name	UTM drilldown top virus	virus

```
select
  virus,
  sum(totalnum) as totalnum
from
  ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, virus, count(*) as totalnum
from $log where $filter and nullifna(virus) is not null group by user_src, virus order by totalnum
desc)### t where $filter-drilldown group by virus order by totalnum desc
```

Dataset Name	Description	Log Category
utm-drilldown-Top-Attacks	UTM drilldown top attacks by name	attack

```
select
  attack,
  sum(attack_count) as attack_count
from
  ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, attack, count(*) as attack_
count from $log where $filter and nullifna(attack) is not null group by user_src, attack order by
attack_count desc)### t where $filter-drilldown group by attack order by attack_count desc
```

Dataset Name	Description	Log Category
utm-drilldown-Top-Vulnerability	UTM drilldown top vulnerability by name	netscan

```
select
  vuln,
  sum(totalnum) as totalnum
from
  ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, vuln, count(*) as totalnum
from $log where $filter and action='vuln-detection' and vuln is not null group by user_src, vuln
order by totalnum desc)### t where $filter-drilldown group by vuln order by totalnum desc
```

Dataset Name	Description	Log Category
utm-drilldown-Top-App-By-Bandwidth	UTM drilldown top applications by bandwidth usage	traffic

```
select
  appid,
  app,
  sum(bandwidth) as bandwidth
from
  ###(select user_src, appid, app, appcat, apprisk, sum(bandwidth) as bandwidth, sum(sessions) as
sessions from ###base(/*tag:rpt_base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip,
dstip, epid, euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
service, appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_
in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte,
0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END)
as sessions from $log-traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null
group by timestamp, dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat,
apprisk, hostname order by bandwidth desc, sessions desc)base### t group by user_src, appid, app,
appcat, apprisk /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t where
$filter-drilldown group by appid, app having sum(bandwidth)>0 order by bandwidth desc
```

Dataset Name	Description	Log Category
utm-drilldown-Top-App-By-Sessions	UTM drilldown top applications by session count	traffic

```
select
  appid,
  app,
  sum(sessions) as sessions
from
  ###(select user_src, appid, app, appcat, apprisk, sum(bandwidth) as bandwidth, sum(sessions) as
sessions from ###base(/*tag:rpt_base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip,
dstip, epid, euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
service, appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_
in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte,
0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END)
as sessions from $log-traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null
group by timestamp, dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat,
apprisk, hostname order by bandwidth desc, sessions desc)base### t group by user_src, appid, app,
appcat, apprisk /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t where
$filter-drilldown group by appid, app order by sessions desc
```

Dataset Name	Description	Log Category
Top5-Users-By-Bandwidth	UTM drilldown top users by bandwidth usage	traffic

```
select
  coalesce(
    nullifna(`user`),
    nullifna(`unauthuser`),
    ipstr(`srcip`)
  ) as dldn_user,
  count(*) as session,
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  ) as bandwidth,
  sum(
    coalesce(sentbyte, 0)
  ) as traffic_out,
  sum(
    coalesce(rcvdbyte, 0)
  ) as traffic_in
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
group by
```

```
  dldn_user
having
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  )> 0
order by
  bandwidth desc
```

Dataset Name	Description	Log Category
bandwidth-app-Top-App-By-Bandwidth-Sessions	Top applications by bandwidth usage	traffic

```
select
  app_group_name(app) as app_group,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out,
  sum(sessions) as sessions
from
  ###(select appid, app, appcat, apprisk, user_src, hostname, dstip, sum(traffic_in) as traffic_
in, sum(traffic_out) as traffic_out, sum(bandwidth) as bandwidth, sum(sessions) as sessions from
###base(/*tag:rpt_base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid,
euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service,
appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions
from $log-traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by
timestamp, dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat, apprisk,
hostname order by bandwidth desc, sessions desc)base### t group by appid, app, appcat, apprisk,
user_src, hostname, dstip /*SkipSTART*/order by bandwidth desc, sessions desc/*SkipEND*/)### t
group by app_group having sum(bandwidth)>0 order by bandwidth desc
```

Dataset Name	Description	Log Category
bandwidth-app-Category-By-Bandwidth	Application Risk Application Usage by Category	traffic

```
select
  appcat,
  sum(bandwidth) as bandwidth
from
  ###(select appid, app, appcat, apprisk, user_src, hostname, dstip, sum(traffic_in) as traffic_
in, sum(traffic_out) as traffic_out, sum(bandwidth) as bandwidth, sum(sessions) as sessions from
###base(/*tag:rpt_base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid,
euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service,
appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions
```

```
from $log-traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by
timestamp, dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat, apprisk,
hostname order by bandwidth desc, sessions desc)base### t group by appid, app, appcat, apprisk,
user_src, hostname, dstip /*SkipSTART*/order by bandwidth desc, sessions desc/*SkipEND*/)### t
where $filter-drilldown and nullifna(appcat) is not null group by appcat having sum(bandwidth)>0
order by bandwidth desc
```

Dataset Name	Description	Log Category
bandwidth-app-Top-Users-By-Bandwidth-Sessions	Bandwidth application top users by bandwidth usage	traffic

```
select
  user_src,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out,
  sum(sessions) as sessions
from
  ###(select appid, app, appcat, apprisk, user_src, hostname, dstip, sum(traffic_in) as traffic_
in, sum(traffic_out) as traffic_out, sum(bandwidth) as bandwidth, sum(sessions) as sessions from
###base(/*tag:rpt_base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid,
euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service,
appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions
from $log-traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by
timestamp, dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat, apprisk,
hostname order by bandwidth desc, sessions desc)base### t group by appid, app, appcat, apprisk,
user_src, hostname, dstip /*SkipSTART*/order by bandwidth desc, sessions desc/*SkipEND*/)### t
group by user_src having sum(bandwidth)>0 order by bandwidth desc
```

Dataset Name	Description	Log Category
bandwidth-app-Traffic-By-Active-User-Number	Bandwidth application traffic by active user number	traffic

```
select
  $flex_timescale(timestamp) as hodex,
  count(distinct user_src) as total_user
from
  ###(select $flex_timestamp as timestamp, coalesce(nullifna(`user`), nullifna(`unauthuser`),
ipstr(`srcip`)) as user_src, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) AS sessions from $log
where $filter and (logflag&(1|32)>0) group by timestamp, user_src order by sessions desc)### t
group by hodex order by hodex
```

Dataset Name	Description	Log Category
bandwidth-app-Top-Dest-By-Bandwidth-Sessions	Bandwidth application top dest by bandwidth usage sessions	traffic

```
select
  coalesce(
    nullifna(
      root_domain(hostname)
    ),
    ipstr(`dstip`)
  ) as dst,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out,
  sum(bandwidth) as bandwidth,
  sum(sessions) as sessions
from
  ###(select appid, app, appcat, apprisk, user_src, hostname, dstip, sum(traffic_in) as traffic_
in, sum(traffic_out) as traffic_out, sum(bandwidth) as bandwidth, sum(sessions) as sessions from
###base(/*tag:rpt_base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid,
euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service,
appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions
from $log-traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by
timestamp, dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat, apprisk,
hostname order by bandwidth desc, sessions desc)base### t group by appid, app, appcat, apprisk,
user_src, hostname, dstip /*SkipSTART*/order by bandwidth desc, sessions desc/*SkipEND*/)### t
group by dst order by bandwidth desc
```

Dataset Name	Description	Log Category
bandwidth-app-Top-Policies-By-Bandwidth-Sessions	Top policies by bandwidth and sessions	traffic

```
select
  coalesce(
    pol.name,
    cast(policyid as text)
  ) as polid,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out,
  sum(sessions) as sessions
from
  ###(select policyid, poluuid, sum(coalesce(rcvddelta, rcvdbyte, 0) + coalesce(sentdelta,
sentbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce
(sentdelta, sentbyte, 0)) as traffic_out, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as
sessions from $log where $filter and (logflag&(1|32)>0) group by policyid, poluuid order by
```

```
bandwidth desc)### t1 left join $ADOMTBL_PLHD_POLINFO pol on t1.poluuid=pol.uuid group by polid
order by bandwidth desc
```

Dataset Name	Description	Log Category
bandwidth-app-Traffic-Statistics	Bandwidth application traffic statistics	traffic

```
drop
  table if exists rpt_tmptbl_1; create temporary table rpt_tmptbl_1(
    total_sessions varchar(255),
    total_bandwidth varchar(255),
    ave_session varchar(255),
    ave_bandwidth varchar(255),
    active_date varchar(255),
    total_users varchar(255),
    total_app varchar(255),
    total_dest varchar(255)
  ); insert into rpt_tmptbl_1 (
    total_sessions, total_bandwidth,
    ave_session, ave_bandwidth
  )
select
  format_numeric_no_decimal(
    sum(sessions)
  ) as total_sessions,
  bandwidth_unit(
    sum(bandwidth)
  ) as total_bandwidth,
  format_numeric_no_decimal(
    cast(
      sum(sessions)/ $days_num as decimal(18, 0)
    )
  ) as ave_session,
  bandwidth_unit(
    cast(
      sum(bandwidth)/ $days_num as decimal(18, 0)
    )
  ) as ave_bandwidth
from
  ###(select timestamp, sum(bandwidth) as bandwidth, sum(traffic_out) as traffic_out, sum(traffic_
in) as traffic_in, sum(sessions) as sessions from ###base(/*tag:rpt_base_t_top_app*/select $flex_
timestamp as timestamp, dvid, srcip, dstip, epid, euid, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, service, appid, app, appcat, apprisk, hostname, sum
(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as
traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth,
sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions from $log-traffic where $filter and
(logflag&(1|32)>0) and nullifna(app) is not null group by timestamp, dvid, srcip, dstip, epid,
euid, user_src, service, appid, app, appcat, apprisk, hostname order by bandwidth desc, sessions
desc)base### base_query group by timestamp order by bandwidth desc, sessions desc)### t; update
rpt_tmptbl_1 set active_date=t1.dom from (select dom, sum(sessions) as sessions from ###(select
$DAY_OF_MONTH as dom, count(*) as sessions from $log where $filter and (logflag&(1|32)>0) group by
```

```
dom order by sessions desc)### t group by dom order by sessions desc limit 1) as t1; update rpt_
tmptbl_1 set total_users=t2.totalnum from (select format_numeric_no_decimal(count(distinct(user_
src))) as totalnum from ###(select appid, app, appcat, apprisk, user_src, hostname, dstip, sum
(traffic_in) as traffic_in, sum(traffic_out) as traffic_out, sum(bandwidth) as bandwidth, sum
(sessions) as sessions from ###base(/*tag:rpt_base_t_top_app*/select $flex_timestamp as timestamp,
dvid, srcip, dstip, epid, euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`))
as user_src, service, appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0))
as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta,
sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1
ELSE 0 END) as sessions from $log-traffic where $filter and (logflag&(1|32)>0) and nullifna(app)
is not null group by timestamp, dvid, srcip, dstip, epid, euid, user_src, service, appid, app,
appcat, apprisk, hostname order by bandwidth desc, sessions desc)base### t group by appid, app,
appcat, apprisk, user_src, hostname, dstip /*SkipSTART*/order by bandwidth desc, sessions
desc/*SkipEND*/)### t) as t2; update rpt_tmptbl_1 set total_app=t3.totalnum from (select format_
numeric_no_decimal(count(distinct(app_group_name(app)))) as totalnum from ###(select appid, app,
appcat, apprisk, user_src, hostname, dstip, sum(traffic_in) as traffic_in, sum(traffic_out) as
traffic_out, sum(bandwidth) as bandwidth, sum(sessions) as sessions from ###base(/*tag:rpt_base_t_
top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid, euid, coalesce(nullifna
(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service, appid, app, appcat,
apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta,
sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte,
0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions from $log-traffic
where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by timestamp, dvid,
srcip, dstip, epid, euid, user_src, service, appid, app, appcat, apprisk, hostname order by
bandwidth desc, sessions desc)base### t group by appid, app, appcat, apprisk, user_src, hostname,
dstip /*SkipSTART*/order by bandwidth desc, sessions desc/*SkipEND*/)### t) as t3; update rpt_
tmptbl_1 set total_dest=t4.totalnum from (select format_numeric_no_decimal(count(distinct(dstip)))
as totalnum from ###(select appid, app, appcat, apprisk, user_src, hostname, dstip, sum(traffic_
in) as traffic_in, sum(traffic_out) as traffic_out, sum(bandwidth) as bandwidth, sum(sessions) as
sessions from ###base(/*tag:rpt_base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip,
dstip, epid, euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
service, appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_
in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte,
0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END)
as sessions from $log-traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null
group by timestamp, dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat,
apprisk, hostname order by bandwidth desc, sessions desc)base### t group by appid, app, appcat,
apprisk, user_src, hostname, dstip /*SkipSTART*/order by bandwidth desc, sessions
desc/*SkipEND*/)### t ) as t4; select 'Total Sessions' as summary, total_sessions as stats from
rpt_tmptbl_1 union all select 'Total Bytes Transferred' as summary, total_bandwidth as stats from
rpt_tmptbl_1 union all select 'Most Active Date By Sessions' as summary, active_date as stats from
rpt_tmptbl_1 union all select 'Total Users' as summary, total_users as stats from rpt_tmptbl_1
union all select 'Total Applications' as summary, total_app as stats from rpt_tmptbl_1 union all
select 'Total Destinations' as summary, total_dest as stats from rpt_tmptbl_1 union all select
'Average Sessions Per Day' as summary, ave_session as stats from rpt_tmptbl_1 union all select
'Average Bytes Per Day' as summary, ave_bandwidth as stats from rpt_tmptbl_1
```

Dataset Name	Description	Log Category
bandwidth-app-Bandwidth-Usage-Summary	Application Traffic Usage Timeline	traffic

```
select
  $flex_timescale(timestamp) as hodex,
  sum(traffic_out) as traffic_out,
  sum(traffic_in) as traffic_in
from
  ###(select timestamp, sum(bandwidth) as bandwidth, sum(traffic_out) as traffic_out, sum(traffic_
in) as traffic_in, sum(sessions) as sessions from ###base(/*tag:rpt_base_t_top_app*/select $flex_
timestamp as timestamp, dvid, srcip, dstip, epid, euid, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, service, appid, app, appcat, apprisk, hostname, sum
(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as
traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth,
sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions from $log-traffic where $filter and
(logflag&(1|32)>0) and nullifna(app) is not null group by timestamp, dvid, srcip, dstip, epid,
euid, user_src, service, appid, app, appcat, apprisk, hostname order by bandwidth desc, sessions
desc)base### base_query group by timestamp order by bandwidth desc, sessions desc)### t where
$filter-drilldown group by hodex having sum(bandwidth)>0 order by hodex
```

Dataset Name	Description	Log Category
bandwidth-app-Sessions-Summary	Number of session timeline	traffic

```
select
  $flex_timescale(timestamp) as hodex,
  sum(sessions) as sessions
from
  ###(select timestamp, sum(bandwidth) as bandwidth, sum(traffic_out) as traffic_out, sum(traffic_
in) as traffic_in, sum(sessions) as sessions from ###base(/*tag:rpt_base_t_bndwdth_sess*/select
$flex_timestamp as timestamp, dvid, srcip, dstip, epid, euid, appcat, apprisk, coalesce(nullifna
(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service, sum(CASE WHEN
(logflag&1>0) THEN 1 ELSE 0 END) as sessions, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum
(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in from $log-traffic where $filter and (logflag&
(1|32)>0) group by timestamp, dvid, srcip, dstip, epid, euid, appcat, apprisk, user_src, service
/*SkipSTART*/order by bandwidth desc, sessions desc/*SkipEND*/)base### base_query group by
timestamp order by bandwidth desc, sessions desc)### t where $filter-drilldown group by hodex
order by hodex
```

Dataset Name	Description	Log Category
bandwidth-app-Top-App-Bandwidth-Usage	Top Application by Bandwidth	traffic

```
select
  app,
  appcat,
  count(distinct user_src) as num_user,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out,
  sum(bandwidth) as bandwidth,
```

```
  sum(sessions) as sessions
from
  ###(select appid, app, appcat, apprisk, user_src, hostname, dstip, sum(traffic_in) as traffic_
in, sum(traffic_out) as traffic_out, sum(bandwidth) as bandwidth, sum(sessions) as sessions from
###base(/*tag:rpt_base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid,
euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service,
appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions
from $log-traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by
timestamp, dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat, apprisk,
hostname order by bandwidth desc, sessions desc)base### t group by appid, app, appcat, apprisk,
user_src, hostname, dstip /*SkipSTART*/order by bandwidth desc, sessions desc/*SkipEND*/)### t
where $filter-drilldown group by app, appcat having sum(bandwidth) > 0 order by bandwidth desc,
sessions desc
```

Dataset Name	Description	Log Category
bandwidth-app-Top-App-Category-By-Bandwidth	Application Risk Application Usage by App and Category	traffic

```
select
  appcat,
  app,
  sum(bandwidth) as bandwidth
from
  ###(select app, appcat, user_src, sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_
out, sum(bandwidth) as bandwidth, sum(sessions) as sessions from ###base(/*tag:rpt_base_t_top_
app*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid, euid, coalesce(nullifna
(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service, appid, app, appcat,
apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta,
sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte,
0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions from $log-traffic
where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by timestamp, dvid,
srcip, dstip, epid, euid, user_src, service, appid, app, appcat, apprisk, hostname order by
bandwidth desc, sessions desc)base### t group by app, appcat, user_src /*SkipSTART*/order by
bandwidth desc, sessions desc/*SkipEND*/)### t group by appcat, app order by bandwidth desc
```

Dataset Name	Description	Log Category
bandwidth-app-Active-User-Count-Timeline	Bandwidth application traffic by active user number	traffic

```
select
  $flex_timescale(timestamp) as hodex,
  count(distinct user_src) as total_user
from
  ###(select $flex_timestamp as timestamp, coalesce(nullifna(`user`), nullifna(`unauthuser`),
ipstr(`srcip`)) as user_src, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) AS sessions from $log
```

```
where $filter and (logflag&(1|32)>0) group by timestamp, user_src order by sessions desc)### t
group by hodex order by hodex
```

Dataset Name	Description	Log Category
bandwidth-app-Top-Dest-By-Bandwidth	Bandwidth application top dest by bandwidth usage sessions	traffic

```
select
  coalesce(
    nullifna(
      root_domain(hostname)
    ),
    ipstr(`dstip`)
  ) as dst,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out,
  sum(bandwidth) as bandwidth,
  sum(sessions) as sessions
from
  ###(select appid, app, appcat, apprisk, user_src, hostname, dstip, sum(traffic_in) as traffic_
in, sum(traffic_out) as traffic_out, sum(bandwidth) as bandwidth, sum(sessions) as sessions from
###base(/*tag:rpt_base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid,
euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service,
appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions
from $log-traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by
timestamp, dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat, apprisk,
hostname order by bandwidth desc, sessions desc)base### t group by appid, app, appcat, apprisk,
user_src, hostname, dstip /*SkipSTART*/order by bandwidth desc, sessions desc/*SkipEND*/)### t
group by dst order by bandwidth desc
```

Dataset Name	Description	Log Category
bandwidth-app-Top-Dest-By-Session	Bandwidth application top dest by bandwidth usage sessions	traffic

```
select
  coalesce(
    nullifna(
      root_domain(hostname)
    ),
    ipstr(`dstip`)
  ) as dst,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out,
  sum(bandwidth) as bandwidth,
  sum(sessions) as sessions
```

```
from
  ###(select appid, app, appcat, apprisk, user_src, hostname, dstip, sum(traffic_in) as traffic_
in, sum(traffic_out) as traffic_out, sum(bandwidth) as bandwidth, sum(sessions) as sessions from
###base(/*tag:rpt_base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid,
euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service,
appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions
from $log-traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by
timestamp, dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat, apprisk,
hostname order by bandwidth desc, sessions desc)base### t group by appid, app, appcat, apprisk,
user_src, hostname, dstip /*SkipSTART*/order by bandwidth desc, sessions desc/*SkipEND*/)### t
group by dst order by bandwidth desc
```

Dataset Name	Description	Log Category
bandwidth-app-Top-Bandwidth-Users	Bandwidth application top users by bandwidth usage	traffic

```
select
  user_src,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out,
  sum(sessions) as sessions
from
  ###(select appid, app, appcat, apprisk, user_src, hostname, dstip, sum(traffic_in) as traffic_
in, sum(traffic_out) as traffic_out, sum(bandwidth) as bandwidth, sum(sessions) as sessions from
###base(/*tag:rpt_base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid,
euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service,
appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions
from $log-traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by
timestamp, dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat, apprisk,
hostname order by bandwidth desc, sessions desc)base### t group by appid, app, appcat, apprisk,
user_src, hostname, dstip /*SkipSTART*/order by bandwidth desc, sessions desc/*SkipEND*/)### t
group by user_src having sum(bandwidth)>0 order by bandwidth desc
```

Dataset Name	Description	Log Category
bandwidth-app-Top-Session-Users	Bandwidth application top users by bandwidth usage	traffic

```
select
  user_src,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out,
  sum(sessions) as sessions
```

```
from
  ###(select appid, app, appcat, apprisk, user_src, hostname, dstip, sum(traffic_in) as traffic_
in, sum(traffic_out) as traffic_out, sum(bandwidth) as bandwidth, sum(sessions) as sessions from
###base(/*tag:rpt_base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid,
euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service,
appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions
from $log-traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by
timestamp, dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat, apprisk,
hostname order by bandwidth desc, sessions desc)base### t group by appid, app, appcat, apprisk,
user_src, hostname, dstip /*SkipSTART*/order by bandwidth desc, sessions desc/*SkipEND*/)### t
group by user_src having sum(bandwidth)>0 order by bandwidth desc
```

Dataset Name	Description	Log Category
Score-Summary-For-All-Users-Devices	Reputation score summary for all users devices	traffic

```
select
  $flex_timescale(timestamp) as hodex,
  sum(scores) as scores
from
  ###(select $flex_timestamp as timestamp, sum(crscore%65536) as scores, count(*) as totalnum from
$log where $filter and (logflag&1>0) and crscore is not null group by timestamp having sum
(crscore%65536)>0 order by timestamp desc)### t group by hodex order by hodex
```

Dataset Name	Description	Log Category
Number-Of-Incidents-For-All-Users-Devices	Reputation number of incidents for all users devices	traffic

```
select
  $flex_timescale(timestamp) as hodex,
  sum(scores) as scores,
  sum(totalnum) as totalnum
from
  ###(select $flex_timestamp as timestamp, sum(crscore%65536) as scores, count(*) as totalnum from
$log where $filter and (logflag&1>0) and crscore is not null group by timestamp having sum
(crscore%65536)>0 order by timestamp desc)### t group by hodex order by hodex
```

Dataset Name	Description	Log Category
Top-Users-By-Reputation-Scores	Reputation top users by scores	traffic

```
select
  coalesce(
    nullifna(`user`),
```

```
    nullifna(`unauthuser`),
    ipstr(`srcip`)
  ) as user_src,
  sum(crscore % 65536) as scores
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
  and crscore is not null
group by
  user_src
having
  sum(crscore % 65536)> 0
order by
  scores desc
```

Dataset Name	Description	Log Category
Top-Devices-By-Reputation-Scores	Reputation top devices by scores	traffic

```
select
  max(
    get_devtype(srcswversion, osname, devtype)
  ) as devtype_new,
  coalesce(
    nullifna(`srcname`),
    nullifna(`srcmac`),
    ipstr(`srcip`)
  ) as dev_src,
  sum(crscore % 65536) as scores
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
  and crscore is not null
group by
  dev_src
having
  sum(crscore % 65536)> 0
order by
  scores desc
```

Dataset Name	Description	Log Category
Top-Users-With-Increased-Scores	Reputation top users with increased scores	traffic

```
drop
  table if exists rpt_tmptbl_1;
drop
  table if exists rpt_tmptbl_2; create temporary table rpt_tmptbl_1 as
select
  f_user,
  sum(sum_rp_score) as sum_rp_score
from
  ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as f_user, sum
(crscore%65536) as sum_rp_score from $log where $pre_period $filter and (logflag&1>0) and crscore
is not null group by f_user having sum(crscore%65536)>0 order by sum_rp_score desc)### t group by
f_user; create temporary table rpt_tmptbl_2 as select f_user, sum(sum_rp_score) as sum_rp_score
from ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as f_user, sum
(crscore%65536) as sum_rp_score from $log where $filter and (logflag&1>0) and crscore is not null
group by f_user having sum(crscore%65536)>0 order by sum_rp_score desc)### t group by f_user;
select t1.f_user, sum(t1.sum_rp_score) as t1_sum_score, sum(t2.sum_rp_score) as t2_sum_score, (sum
(t2.sum_rp_score)-sum(t1.sum_rp_score)) as delta from rpt_tmptbl_1 as t1 inner join rpt_tmptbl_2
as t2 on t1.f_user=t2.f_user where t2.sum_rp_score > t1.sum_rp_score group by t1.f_user order by
delta desc
```

Dataset Name	Description	Log Category
Top-Devices-With-Increased-Scores	Reputation top devices with increased scores	traffic

```
drop
  table if exists rpt_tmptbl_1;
drop
  table if exists rpt_tmptbl_2; create temporary table rpt_tmptbl_1 as
select
  f_device,
  devtype_new,
  sum(sum_rp_score) as sum_rp_score
from
  ###(select coalesce(nullifna(`srcname`),nullifna(`srcmac`), ipstr(`srcip`)) as f_device, get_
devtype(srcswversion, osname, devtype) as devtype_new, sum(crscore%65536) as sum_rp_score from
$log where $pre_period $filter and (logflag&1>0) and crscore is not null group by f_device,
devtype_new having sum(crscore%65536)>0 order by sum_rp_score desc)### t group by f_device,
devtype_new; create temporary table rpt_tmptbl_2 as select f_device, devtype_new, sum(sum_rp_
score) as sum_rp_score from ###(select coalesce(nullifna(`srcname`),nullifna(`srcmac`), ipstr
(`srcip`)) as f_device, get_devtype(srcswversion, osname, devtype) as devtype_new, sum
(crscore%65536) as sum_rp_score from $log where $filter and (logflag&1>0) and crscore is not null
group by f_device, devtype_new having sum(crscore%65536)>0 order by sum_rp_score desc)### t group
by f_device, devtype_new; select t1.f_device, t1.devtype_new , sum(t1.sum_rp_score) as t1_sum_
score, sum(t2.sum_rp_score) as t2_sum_score, (sum(t2.sum_rp_score)-sum(t1.sum_rp_score)) as delta
from rpt_tmptbl_1 as t1 inner join rpt_tmptbl_2 as t2 on t1.f_device=t2.f_device and t1.devtype_
```

```
new=t2.devtype_new where t2.sum_rp_score > t1.sum_rp_score group by t1.f_device, t1.devtype_new
order by delta desc
```

Dataset Name	Description	Log Category
Attacks-By-Severity	Threat attacks by severity	attack

```
select
  (
    case when severity =& #039;critical' then 'Critical' when severity='high' then 'High' when
severity='medium' then 'Medium' when severity='low' then 'Low' when severity='info' then 'Info'
end) as severity, count(*) as totalnum from $log where $filter group by severity order by totalnum
desc
```

Dataset Name	Description	Log Category
Top-Attacks-Detected	Threat top attacks detected	attack

```
select
  attack,
  attackid,
  cve,
  severity,
  sum(attack_count) as attack_count
from
  ###(select attack, attackid, t1.severity, cve, (case when t1.severity = 'critical' then 1 when
t1.severity = 'high' then 2 when t1.severity = 'medium'  then 3 when t1.severity = 'low' then 4
else 5 end) as severity_level, count(*) as attack_count from $log t1 left join (select name, cve,
vuln_type from ips_mdata) t2 on t1.attack=t2.name where $filter and nullifna(attack) is not null
group by attack, attackid, t1.severity, severity_level, cve /*SkipSTART*/order by severity_level,
attack_count desc/*SkipEND*/)### t group by attack, attackid, severity, severity_level, cve order
by severity_level, attack_count desc
```

Dataset Name	Description	Log Category
Top-Attacks-Blocked	Threat top attacks blocked	attack

```
select
  attack,
  count(*) as attack_count
from
  $log
where
  $filter
  and nullifna(attack) is not null
  and action not in (
    & #039;detected', 'pass_session') group  by attack order by attack_count desc
```

Dataset Name	Description	Log Category
Top-Virus-Source	Threat top virus source	virus

```
select
  source,
  hostname,
  sum(totalnum) as totalnum
from
  ###(select source, ipstr(`victim`) as hostname, sum(totalnum) as totalnum from ( select (CASE
WHEN direction='incoming' THEN dstip ELSE srcip END) as source, (CASE WHEN direction='incoming'
THEN srcip ELSE dstip END) as victim, count(*) as totalnum from $log where $filter and nullifna
(virus) is not null group by source, victim ) t group by source, hostname /*SkipSTART*/order by
totalnum desc/*SkipEND*/)### t group by source, hostname order by totalnum desc
```

Dataset Name	Description	Log Category
Intrusion-in-Last-7-Days	Threat intrusion timeline	attack

```
select
  $flex_timescale(timestamp) as hodex,
  sum(totalnum) as totalnum
from
  ###(select $flex_timestamp as timestamp, count(*) as totalnum from $log where $filter group by
timestamp /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t group by hodex order by hodex
```

Dataset Name	Description	Log Category
Virus-Time-Line	Threat virus timeline	virus

```
select
  $flex_datetime(timestamp) as hodex,
  sum(totalnum) as totalnum
from
  ###(select $flex_timestamp as timestamp, count(*) as totalnum from $log where $filter and
nullifna(virus) is not null group by timestamp /*SkipSTART*/order by timestamp desc/*SkipEND*/)###
t group by hodex order by hodex
```

Dataset Name	Description	Log Category
Top-Spyware-Victims	Threat top spyware victims	virus

```
select
  user_src,
  sum(totalnum) as totalnum
from
  ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, virus, count(*) as totalnum
```

```
from $log where $filter group by user_src, virus /*SkipSTART*/order by totalnum
desc/*SkipEND*/)### t where  virus like 'Riskware%' group by user_src order by totalnum desc
```

Dataset Name	Description	Log Category
Top-Spyware-by-Name	Threat top spyware by name	virus

```
select
  virus,
  max(virusid_s) as virusid,
  sum(totalnum) as totalnum
from
  ###(select filename, analyticscksum, service, fsaverdict, dtype, coalesce(nullifna(`user`),
ipstr(`srcip`)) as user_src, virus, virusid_to_str(virusid, eventtype) as virusid_s, count(*) as
totalnum from $log where $filter group by filename, analyticscksum, service, fsaverdict, dtype,
user_src, virus, virusid_s /*SkipSTART*/order by totalnum desc/*SkipEND*/)### t where virus like
'Riskware%' group by virus order by totalnum desc
```

Dataset Name	Description	Log Category
Top-Spyware-Source	Threat top spyware source	traffic

```
select
  srcip,
  hostname,
  sum(totalnum) as totalnum
from
  ###(select srcip, hostname, virus, count(*) as totalnum from $log where $filter and
(logflag&1>0) group by srcip, hostname, virus order by totalnum desc)### t where virus like
'Riskware%' group by srcip, hostname order by totalnum desc
```

Dataset Name	Description	Log Category
Spyware-Time-Line	Threat spyware timeline	virus

```
select
  $flex_timescale(timestamp) as hodex,
  sum(totalnum) as totalnum
from
  ###(select $flex_timestamp as timestamp, virus, count(*) as totalnum from $log where $filter
group by timestamp, virus /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where virus like
'Riskware%' group by hodex order by hodex
```

Dataset Name	Description	Log Category
Top-Adware-Victims	Threat top adware victims	virus

```
select
  user_src,
  sum(totalnum) as totalnum
from
  ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, virus, count(*) as totalnum
from $log where $filter group by user_src, virus /*SkipSTART*/order by totalnum
desc/*SkipEND*/)### t where virus like 'Adware%' group by user_src order by totalnum desc
```

Dataset Name	Description	Log Category
Top-Adware-by-Name	Threat top adware by name	virus

```
select
  virus,
  max(virusid_s) as virusid,
  sum(totalnum) as totalnum
from
  ###(select filename, analyticscksum, service, fsaverdict, dtype, coalesce(nullifna(`user`),
ipstr(`srcip`)) as user_src, virus, virusid_to_str(virusid, eventtype) as virusid_s, count(*) as
totalnum from $log where $filter group by filename, analyticscksum, service, fsaverdict, dtype,
user_src, virus, virusid_s /*SkipSTART*/order by totalnum desc/*SkipEND*/)### t where virus like
'Adware%' group by virus order by totalnum desc
```

Dataset Name	Description	Log Category
Top-Adware-Source	Threat top adware source	traffic

```
select
  srcip,
  hostname,
  sum(totalnum) as totalnum
from
  ###(select srcip, hostname, virus, count(*) as totalnum from $log where $filter and
(logflag&1>0) group by srcip, hostname, virus order by totalnum desc)### t where virus like
'Adware%' group by srcip, hostname order by totalnum desc
```

Dataset Name	Description	Log Category
Adware-Time-Line	Threat adware timeline	virus

```
select
  $flex_timescale(timestamp) as hodex,
  sum(totalnum) as totalnum
from
  ###(select $flex_timestamp as timestamp, virus, count(*) as totalnum from $log where $filter
group by timestamp, virus /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where virus like
'Adware%' group by hodex order by hodex
```

Dataset Name	Description	Log Category
Intrusions-Timeline-By-Severity	Threat intrusions timeline by severity	attack

```
select
  $flex_timescale(timestamp) as timescale,
  sum(critical) as critical,
  sum(high) as high,
  sum(medium) as medium,
  sum(low) as low,
  sum(info) as info
from
  ###(select $flex_timestamp as timestamp, sum(case when severity = 'critical' then 1 else 0 end)
as critical, sum(case when severity = 'high' then 1 else 0 end) as high, sum(case when severity =
'medium' then 1 else 0 end) as medium, sum(case when severity in ('notice', 'low') then 1 else 0
end) as low, sum(case when severity = 'info' or severity = 'debug' then 1 else 0 end) as info from
$log where $filter group by timestamp /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t group
by timescale order by timescale
```

Dataset Name	Description	Log Category
Important-Intrusions-Timeline-By-Severity	Threat intrusions timeline by severity	attack

```
select
  $flex_timescale(timestamp) as timescale,
  sum(critical) as critical,
  sum(high) as high,
  sum(medium) as medium,
  sum(low) as low,
  sum(info) as info
from
  ###(select $flex_timestamp as timestamp, sum(case when severity = 'critical' then 1 else 0 end)
as critical, sum(case when severity = 'high' then 1 else 0 end) as high, sum(case when severity =
'medium' then 1 else 0 end) as medium, sum(case when severity in ('notice', 'low') then 1 else 0
end) as low, sum(case when severity = 'info' or severity = 'debug' then 1 else 0 end) as info from
$log where $filter group by timestamp /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t group
by timescale order by timescale
```

Dataset Name	Description	Log Category
Top-Intrusions-By-Types	Threat top intrusions by types	attack

```
select
  vuln_type,
  count(*) as totalnum
from
  $log t1
```

```
  left join (
    select
      name,
      cve,
      vuln_type
    from
      ips_mdata
  ) t2 on t1.attack = t2.name
where
  $filter
  and vuln_type is not null
group by
  vuln_type
order by
  totalnum desc
```

Dataset Name	Description	Log Category
Critical-Severity-Intrusions	Threat critical severity intrusions	attack

```
select
  attack,
  attackid,
  cve,
  vuln_type,
  count(*) as totalnum
from
  $log t1
  left join (
    select
      name,
      cve,
      vuln_type
    from
      ips_mdata
  ) t2 on t1.attack = t2.name
where
  $filter
  and t1.severity = & #039;critical' and nullifna(attack) is not null group by attack, attackid,
cve, vuln_type order by totalnum desc
```

Dataset Name	Description	Log Category
High-Severity-Intrusions	Threat high severity intrusions	attack

```
select
  attack,
  attackid,
  vuln_type,
```

```
  cve,
  count(*) as totalnum
from
  $log t1
  left join (
    select
      name,
      cve,
      vuln_type
    from
      ips_mdata
  ) t2 on t1.attack = t2.name
where
  $filter
  and t1.severity =& #039;high' and nullifna(attack) is not null group by attack, attackid, vuln_
type, cve order by totalnum desc
```

Dataset Name	Description	Log Category
Medium-Severity-Intrusions	Threat medium severity intrusions	attack

```
select
  attack,
  vuln_type,
  cve,
  count(*) as totalnum
from
  $log t1
  left join (
    select
      name,
      cve,
      vuln_type
    from
      ips_mdata
  ) t2 on t1.attack = t2.name
where
  $filter
  and t1.severity =& #039;medium' and nullifna(attack) is not null group by attack, vuln_type, cve
order by totalnum desc
```

Dataset Name	Description	Log Category
Top-Intrusion-Victims	Threat top intrusion victims	attack

```
select
  victim,
  sum(cri_num) as critical,
  sum(high_num) as high,
```

```
  sum(med_num) as medium,
  sum(cri_num + high_num + med_num) as totalnum
from
  ###(select (CASE WHEN direction='incoming' THEN srcip ELSE dstip END) as victim, sum((case when
severity='critical' then 1 else 0 end)) as cri_num, sum(case when severity='high' then 1 else 0
end) as high_num, sum(case when severity='medium' then 1 else 0 end) as med_num from $log where
$filter and severity in ('critical', 'high', 'medium') group by victim order by cri_num desc,
high_num desc, med_num desc)### t group by victim order by totalnum desc
```

Dataset Name	Description	Log Category
Top-Intrusion-Sources	Threat top intrusion sources	attack

```
select
  source,
  sum(cri_num) as critical,
  sum(high_num) as high,
  sum(med_num) as medium,
  sum(cri_num + high_num + med_num) as totalnum
from
  ###(select (CASE WHEN direction='incoming' THEN dstip ELSE srcip END) as source, sum(case when
severity='critical' then 1 else 0 end) as cri_num, sum(case when severity='high' then 1 else 0
end) as high_num, sum(case when severity='medium' then 1 else 0 end) as med_num from $log where
$filter and severity in ('critical', 'high', 'medium') group by source order by cri_num desc,
high_num desc, med_num desc)### t group by source order by totalnum desc
```

Dataset Name	Description	Log Category
Top-Blocked-Intrusions	Threat top blocked intrusions	attack

```
select
  attack,
  attackid,
  (
    case when severity =& #039;critical' then 'Critical' when severity='high' then 'High' when
severity='medium' then 'Medium' when severity='low' then 'Low' when severity='info' then 'Info'
end) as severity_name, sum(totalnum) as totalnum, vuln_type, (case when severity='critical' then 0
when severity='high' then 1 when severity='medium' then 2 when severity='low' then 3 when
severity='info' then 4 else 5 end) as severity_number from ###(select attack, attackid,
t1.severity, count(*) as totalnum, vuln_type, action from $log t1 left join (select name, cve,
vuln_type from ips_mdata) t2 on t1.attack=t2.name where $filter and nullifna(attack) is not null
group by attack, attackid, t1.severity, vuln_type, action order by totalnum desc)### t where
action not in ('detected', 'pass_session') group by attack, attackid, severity, vuln_type order by
severity_number, totalnum desc
```

Dataset Name	Description	Log Category
Top-Monitored-Intrusions	Threat top monitored intrusions	attack

```
select
  attack,
  attackid,
  (
    case when severity =& #039;critical' then 'Critical' when severity='high' then 'High' when
severity='medium' then 'Medium' when severity='low' then 'Low' when severity='info' then 'Info'
end) as severity_name, sum(totalnum) as totalnum, vuln_type, (case when severity='critical' then 0
when severity='high' then 1 when severity='medium' then 2 when severity='low' then 3 when
severity='info' then 4 else 5 end) as severity_number from ###(select attack, attackid,
t1.severity, count(*) as totalnum, vuln_type, action from $log t1 left join (select name, cve,
vuln_type from ips_mdata) t2 on t1.attack=t2.name where $filter and nullifna(attack) is not null
group by attack, attackid, t1.severity, vuln_type, action order by totalnum desc)### t where
action in ('detected', 'pass_session') group by attack, attackid, severity, vuln_type order by
severity_number, totalnum desc
```

Dataset Name	Description	Log Category
Attacks-Over-HTTP-HTTPs	Threat attacks over HTTP HTTPs	attack

```
select
  attack,
  attackid,
  (
    case when severity =& #039;critical' then 'Critical' when  severity='high' then 'High' when
severity='medium' then 'Medium' when severity='low' then 'Low' when severity='info' then 'Info'
end) as severity, count(*) as totalnum, (case when severity='critical' then 0 when severity='high'
then 1 when severity='medium' then 2 when severity='low' then 3 when severity='info' then 4 else 5
end) as severity_number from $log where $filter and severity in ('critical', 'high', 'medium') and
upper(service) in ('HTTP', 'HTTPS') group by attack, attackid, severity, severity_number order by
severity_number, totalnum desc
```

Dataset Name	Description	Log Category
default-AP-Detection-Summary-by-Status-OffWire	Default access point detection summary by status off-wire	event

```
select
  (
    case apstatus when 1 then & #039;rogue' when 2 then 'accepted' when 3 then 'suppressed' else
'others' end) as ap_full_status, count(*) as totalnum from (select apstatus, bssid, ssid from ###
(select apstatus, bssid, ssid, onwire, count(*) as subtotal from $log where $filter and apstatus
is not null and apstatus!=0 and bssid is not null and logid_to_int(logid) in (43527, 43521, 43525,
43563, 43564, 43565, 43566, 43569, 43570, 43571, 43582, 43583, 43584, 43585) group by apstatus,
bssid, ssid, onwire order by subtotal desc)### t where onwire='no' group by apstatus, bssid, ssid)
t group by ap_full_status order by totalnum desc
```

Dataset Name	Description	Log Category
default-AP-Detection-Summary-by-Status-OffWire_table	Default access point detection summary by status off-wire	event

```
select
  (
    case apstatus when 1 then & #039;rogue' when 2 then 'accepted' when 3 then 'suppressed' else
'others' end) as ap_full_status, count(*) as totalnum from (select apstatus, bssid, ssid from ###
(select apstatus, bssid, ssid, onwire, count(*) as subtotal from $log where $filter and apstatus
is not null and apstatus!=0 and bssid is not null and logid_to_int(logid) in (43527, 43521, 43525,
43563, 43564, 43565, 43566, 43569, 43570, 43571, 43582, 43583, 43584, 43585) group by apstatus,
bssid, ssid, onwire order by subtotal desc)### t where onwire='no' group by apstatus, bssid, ssid)
t group by ap_full_status order by totalnum desc
```

Dataset Name	Description	Log Category
default-AP-Detection-Summary-by-Status-OnWire	Default access point detection summary by status on-wire	event

```
select
  (
    case apstatus when 1 then & #039;rogue' when 2 then 'accepted' when 3 then 'suppressed' else
'others' end) as ap_full_status, count(*) as totalnum from (select apstatus, bssid, ssid from ###
(select apstatus, bssid, ssid, onwire, count(*) as subtotal from $log where $filter and apstatus
is not null and apstatus!=0 and bssid is not null and logid_to_int(logid) in (43527, 43521, 43525,
43563, 43564, 43565, 43566, 43569, 43570, 43571, 43582, 43583, 43584, 43585) group by apstatus,
bssid, ssid, onwire order by subtotal desc)### t where onwire='yes' group by apstatus, bssid,
ssid) t group by ap_full_status order by totalnum desc
```

Dataset Name	Description	Log Category
default-AP-Detection-Summary-by-Status-OnWire_table	Default access point detection summary by status on-wire	event

```
select
  (
    case apstatus when 1 then & #039;rogue' when 2 then 'accepted' when 3 then 'suppressed' else
'others' end) as ap_full_status, count(*) as totalnum from (select apstatus, bssid, ssid from ###
(select apstatus, bssid, ssid, onwire, count(*) as subtotal from $log where $filter and apstatus
is not null and apstatus!=0 and bssid is not null and logid_to_int(logid) in (43527, 43521, 43525,
43563, 43564, 43565, 43566, 43569, 43570, 43571, 43582, 43583, 43584, 43585) group by apstatus,
bssid, ssid, onwire order by subtotal desc)### t where onwire='yes' group by apstatus, bssid,
ssid) t group by ap_full_status order by totalnum desc
```

Dataset Name	Description	Log Category
default-Managed-AP-Summary	Default managed access point summary	event

```
select
  (
    case when (
      action like & #039;%join%' and logid_to_int(logid) in (43522, 43551)) then 'Authorized' else
'Unauthorized' end) as ap_status, count(*) as totalnum from $log where $filter and logid_to_int
(logid) in (43522, 43551) group by ap_status order by totalnum desc
```

Dataset Name	Description	Log Category
default-Managed-AP-Summary_table	Default managed access point summary	event

```
select
  (
    case when (
      action like & #039;%join%' and logid_to_int(logid) in (43522, 43551)) then 'Authorized' else
'Unauthorized' end) as ap_status, count(*) as totalnum from $log where $filter and logid_to_int
(logid) in (43522, 43551) group by ap_status order by totalnum desc
```

Dataset Name	Description	Log Category
default-Unclassified-AP-Summary	Default unclassified access point summary	event

```
select
  (
    case onwire when & #039;no' then 'off-wire' when 'yes' then 'on-wire' else 'others' end) as
ap_status, count(*) as totalnum from ###(select onwire, ssid, bssid, count(*) as subtotal from
$log where $filter and apstatus=0 and bssid is not null and logid_to_int(logid) in (43521, 43525,
43527, 43563, 43564, 43565, 43566, 43569, 43570, 43571, 43582, 43583, 43584, 43585) group by
onwire, ssid, bssid order by subtotal desc)### t group by ap_status order by totalnum desc
```

Dataset Name	Description	Log Category
default-Unclassified-AP-Summary_table	Default unclassified access point summary	event

```
select
  (
    case onwire when & #039;no' then 'off-wire' when 'yes' then 'on-wire' else 'others' end) as
ap_status, count(*) as totalnum from ###(select onwire, ssid, bssid, count(*) as subtotal from
$log where $filter and apstatus=0 and bssid is not null and logid_to_int(logid) in (43521, 43525,
43527, 43563, 43564, 43565, 43566, 43569, 43570, 43571, 43582, 43583, 43584, 43585) group by
onwire, ssid, bssid order by subtotal desc)### t group by ap_status order by totalnum desc
```

Dataset Name	Description	Log Category
default-selected-AP-Details-OffWire	Default selected access point details off-wire	event

```
select
  (
    case apstatus when 0 then & #039;unclassified' when 1 then 'rogue' when 2 then 'accepted' when
3 then 'suppressed' else 'others' end) as ap_full_status, devid, vd, ssid, bssid, manuf, rssi,
channel, radioband, from_dtime(min(first_seen)) as first_seen, from_dtime(max(last_seen)) as last_
seen, detectionmethod, timestamp, onwire as on_wire from ###(select apstatus, devid, vd, ssid,
bssid, manuf, rssi, channel, radioband, min(dtime) as first_seen, max(dtime) as last_seen,
detectionmethod, $flex_timestamp as timestamp, onwire from $log where $filter and apstatus is not
null and bssid is not null and logid_to_int(logid) in (43521, 43563, 43564, 43565, 43566, 43569,
43570, 43571) group by apstatus, devid, vd, ssid, bssid, manuf, rssi, channel, radioband,
detectionmethod, timestamp, onwire order by timestamp desc)### t where onwire='no' group by ap_
full_status, devid, vd, ssid, bssid, manuf, rssi, channel, radioband, detectionmethod, timestamp,
onwire, apstatus order by timestamp desc
```

Dataset Name	Description	Log Category
default-selected-AP-Details-OnWire	Default selected access point details on-wire	event

```
select
  (
    case apstatus when 0 then & #039;unclassified' when 1 then 'rogue' when 2 then 'accepted' when
3 then 'suppressed' else 'others' end) as ap_full_status, devid, vd, ssid, bssid, manuf, rssi,
channel, radioband, from_dtime(min(first_seen)) as first_seen, from_dtime(max(last_seen)) as last_
seen, detectionmethod, timestamp, onwire as on_wire from ###(select apstatus, devid, vd, ssid,
bssid, manuf, rssi, channel, radioband, min(dtime) as first_seen, max(dtime) as last_seen,
detectionmethod, $flex_timestamp as timestamp, onwire from $log where $filter and apstatus is not
null and bssid is not null and logid_to_int(logid) in (43521, 43563, 43564, 43565, 43566, 43569,
43570, 43571) group by apstatus, devid, vd, ssid, bssid, manuf, rssi, channel, radioband,
detectionmethod, timestamp, onwire order by timestamp desc)### t where onwire='yes' group by ap_
full_status, devid, vd, ssid, bssid, manuf, rssi, channel, radioband, detectionmethod, timestamp,
onwire, apstatus order by timestamp desc
```

Dataset Name	Description	Log Category
event-Wireless-Client-Details	Event wireless client details	event

```
drop
  table if exists rpt_tmptbl_1; create temporary table rpt_tmptbl_1 as
select
  ip,
  lmac,
  sn,
  ssid,
```

```
  channel,
  radioband,
  min(first) as first,
  max(last) as last
from
  ###(select srcip as ip, lower(stamac) as lmac, sn, ssid, channel, radioband, min(dtime) as
first, max(dtime) as last from $log-event where $filter and srcip is not null and stamac is not
null and sn is not null and ssid is not null group by srcip, lmac, sn, ssid, channel, radioband
order by srcip)### t group by ip, lmac, sn, ssid, channel, radioband; select user_src, ip, lmac,
sn, ssid, channel, radioband, from_dtime(first) as first_seen, from_dtime(last) as last_seen, cast
(volume as decimal(18,2)) as bandwidth from (select * from rpt_tmptbl_1 inner join (select user_
src, srcip, sum(volume) as volume from ###(select coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, srcip, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte,
0)) as volume from $log-traffic where $filter-time and (logflag&1>0) and srcip is not null group
by user_src, srcip having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by volume
desc)### t group by user_src, srcip order by user_src, srcip) t on rpt_tmptbl_1.ip = t.srcip) t
order by volume desc
```

Dataset Name	Description	Log Category
event-Wireless-Accepted-Offwire	Event wireless accepted off-wire	event

```
select
  & #039;accepted' as ap_full_status, devid, vd, ssid, bssid, manuf, channel, radioband, from_
dtime(max(last_seen)) as last_seen, detectionmethod, snclosest, 'no' as on_wire from ###(select
devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest, onwire, logid,
apstatus, max(dtime) as last_seen from $log where $filter and bssid is not null and logid_to_int
(logid) in (43521, 43525, 43563, 43564, 43565, 43566, 43569, 43570, 43571) group by devid, vd,
ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest, onwire, logid, apstatus order
by last_seen desc)### t where apstatus=2 and onwire='no' group by devid, vd, ssid, bssid, manuf,
channel, radioband, detectionmethod, snclosest order by last_seen desc
```

Dataset Name	Description	Log Category
event-Wireless-Accepted-Onwire	Event wireless accepted on-wire	event

```
select
  & #039;accepted' as ap_full_status, devid, vd, ssid, bssid, manuf, channel, radioband, from_
dtime(max(last_seen)) as last_seen, detectionmethod, snclosest, 'yes' as on_wire from ###(select
devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest, onwire, apstatus,
signal, max(dtime) as last_seen from $log where $filter and bssid is not null and logid_to_int
(logid) in (43521, 43525, 43563, 43564, 43565, 43566, 43569, 43570, 43571) group by devid, vd,
ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest, onwire, apstatus, signal order
by last_seen desc)### t where apstatus=2 and onwire='yes' group by devid, vd, ssid, bssid, manuf,
channel, radioband, detectionmethod, snclosest order by last_seen desc
```

Dataset Name	Description	Log Category
event-Wireless-Rogue-Offwire	Event wireless rogue off-wire	event

```
select
  & #039;rogue' as ap_full_status, devid, vd, ssid, bssid, manuf, channel, radioband, from_dtime
(max(last_seen)) as last_seen, detectionmethod, snclosest, 'no' as on_wire from ###(select devid,
vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest, onwire, logid, apstatus,
max(dtime) as last_seen from $log where $filter and bssid is not null and logid_to_int(logid) in
(43521, 43525, 43563, 43564, 43565, 43566, 43569, 43570, 43571) group by devid, vd, ssid, bssid,
manuf, channel, radioband, detectionmethod, snclosest, onwire, logid, apstatus order by last_seen
desc)### t where apstatus=1 and onwire='no' group by devid, vd, ssid, bssid, manuf, channel,
radioband, detectionmethod, snclosest order by last_seen desc
```

Dataset Name	Description	Log Category
event-Wireless-Rogue-Onwire	Event wireless rogue on-wire	event

```
select
  & #039;rogue' as ap_full_status, devid, vd, ssid, bssid, manuf, channel, radioband, from_dtime
(max(last_seen)) as last_seen, detectionmethod, snclosest, 'yes' as on_wire from ###(select devid,
vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest, onwire, apstatus, signal,
max(dtime) as last_seen from $log where $filter and bssid is not null and logid_to_int(logid) in
(43521, 43525, 43563, 43564, 43565, 43566, 43569, 43570, 43571) group by devid, vd, ssid, bssid,
manuf, channel, radioband, detectionmethod, snclosest, onwire, apstatus, signal order by last_seen
desc)### t where apstatus=1 and onwire='yes' group by devid, vd, ssid, bssid, manuf, channel,
radioband, detectionmethod, snclosest order by last_seen desc
```

Dataset Name	Description	Log Category
event-Wireless-Suppressed-Offwire	Event wireless suppressed off-wire	event

```
select
  & #039;suppressed' as ap_full_status, devid, vd, ssid, bssid, manuf, channel, radioband, from_
dtime(max(last_seen)) as last_seen, detectionmethod, snclosest, 'no' as on_wire from ###(select
devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest, onwire, logid,
apstatus, max(dtime) as last_seen from $log where $filter and bssid is not null and logid_to_int
(logid) in (43521, 43525, 43563, 43564, 43565, 43566, 43569, 43570, 43571) group by devid, vd,
ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest, onwire, logid, apstatus order
by last_seen desc)### t where apstatus=3 and onwire='no' group by devid, vd, ssid, bssid, manuf,
channel, radioband, detectionmethod, snclosest order by last_seen desc
```

Dataset Name	Description	Log Category
event-Wireless-Suppressed-Onwire	Event wireless suppressed on-wire	event

```
select
  & #039;suppressed' as ap_full_status, devid, vd, ssid, bssid, manuf, channel, radioband, from_
dtime(max(last_seen)) as last_seen, detectionmethod, snclosest, 'yes' as on_wire from ###(select
devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest, onwire, apstatus,
signal, max(dtime) as last_seen from $log where $filter and bssid is not null and logid_to_int
(logid) in (43521, 43525, 43563, 43564, 43565, 43566, 43569, 43570, 43571) group by devid, vd,
```

```
ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest, onwire, apstatus, signal order
by last_seen desc)### t where apstatus=3 and onwire='yes' group by devid, vd, ssid, bssid, manuf,
channel, radioband, detectionmethod, snclosest order by last_seen desc
```

Dataset Name	Description	Log Category
event-Wireless-Unclassified-Offwire	Event wireless unclassified off-wire	event

```
select
  & #039;unclassified' as ap_full_status, devid, vd, ssid, bssid, manuf, channel, radioband, from_
dtime(max(last_seen)) as last_seen, detectionmethod, snclosest, 'no' as on_wire from ###(select
devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest, onwire, logid,
apstatus, max(dtime) as last_seen from $log where $filter and bssid is not null and logid_to_int
(logid) in (43521, 43525, 43563, 43564, 43565, 43566, 43569, 43570, 43571) group by devid, vd,
ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest, onwire, logid, apstatus order
by last_seen desc)### t where apstatus=0 and onwire='no' group by devid, vd, ssid, bssid, manuf,
channel, radioband, detectionmethod, snclosest order by last_seen desc
```

Dataset Name	Description	Log Category
event-Wireless-Unclassified-Onwire	Event wireless unclassified on-wire	event

```
select
  & #039;unclassified' as ap_full_status, devid, vd, ssid, bssid, manuf, channel, radioband, from_
dtime(max(last_seen)) as last_seen, detectionmethod, snclosest, 'yes' as on_wire from ###(select
devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest, onwire, apstatus,
signal, max(dtime) as last_seen from $log where $filter and bssid is not null and logid_to_int
(logid) in (43521, 43525, 43563, 43564, 43565, 43566, 43569, 43570, 43571) group by devid, vd,
ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest, onwire, apstatus, signal order
by last_seen desc)### t where apstatus=0 and onwire='yes' group by devid, vd, ssid, bssid, manuf,
channel, radioband, detectionmethod, snclosest order by last_seen desc
```

Dataset Name	Description	Log Category
default-Top-IPSEC-Vpn-Dial-Up-User-By-Bandwidth	Default top IPsec VPN dial up user by bandwidth usage	event

```
select
  coalesce(
    xauthuser_agg,
    user_agg,
    ipstr(`remip`)
  ) as user_src,
  from_dtime(
    min(s_time)
  ) as start_time,
  sum(bandwidth) as bandwidth,
```

```
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out
from
  (
    select
      devid,
      vd,
      string_agg(
        distinct xauthuser_agg,
        & #039; ') as xauthuser_agg, string_agg(distinct user_agg, ' ') as user_agg, remip,
tunnelid, min(s_time) as s_time, max(e_time) as e_time, sum(bandwidth) as bandwidth, sum(traffic_
in) as traffic_in, sum(traffic_out) as traffic_out from ###(select devid, vd, remip, nullifna
(`xauthuser`) as xauthuser_agg, nullifna(`user`) as user_agg, tunnelid, max(coalesce(duration,0))
as max_duration, min(coalesce(duration,0)) as min_duration, min(coalesce(dtime, 0)) as s_time, max
(coalesce(dtime, 0)) as e_time, coalesce(sum(case when sentdelta>0 then sentdelta else 0 end), 0)
AS traffic_out, coalesce(sum(case when rcvddelta>0 then rcvddelta else 0 end), 0) AS traffic_in,
coalesce(sum(case when rcvddelta>0 and sentdelta>0 then rcvddelta+sentdelta when rcvddelta>0 then
rcvddelta when sentdelta>0 then sentdelta else 0 end), 0) AS bandwidth from (select dtime, devid,
vd, remip, `xauthuser`, `user`, duration, tunnelid, tunnelip, tunneltype, sentbyte-lag(coalesce
(sentbyte, 0)) OVER (PARTITION BY dvid, tunnelid ORDER BY eventtime) AS sentdelta, rcvdbyte-lag
(coalesce(rcvdbyte, 0)) OVER (PARTITION BY dvid, tunnelid ORDER BY eventtime) AS rcvddelta from
$log where $filter and subtype='vpn' and action in ('tunnel-up','tunnel-stats', 'tunnel-down',
'ssl-login-fail', 'ipsec-login-fail') and tunnelid is not null) t where tunneltype like 'ipsec%'
and not (tunnelip is null or tunnelip='0.0.0.0') and tunnelid!=0 group by devid, vd, remip,
xauthuser_agg, user_agg, tunnelid order by bandwidth desc)### t group by devid, vd, remip,
tunnelid) tt group by user_src having sum(bandwidth)>0 order by bandwidth desc
```

Dataset Name	Description	Log Category
default-Top-Sources-Of-SSL-VPN-Tunnels-By-Bandwidth	Default top sources of SSL VPN tunnels by bandwidth usage	event

```
select
  remip as remote_ip,
  sum(bandwidth) as bandwidth
from
  ###(select $flex_timestamp as timestamp, devid, vd, remip, tunnelid, (case when tunneltype like
'ipsec%' then 'ipsec' else tunneltype end) as t_type, (case when action='tunnel-up' then 1 else 0
end) as tunnelup, coalesce(sum(case when sentdelta>0 then sentdelta else 0 end), 0) AS traffic_
out, coalesce(sum(case when rcvddelta>0 then rcvddelta else 0 end), 0) AS traffic_in, coalesce(sum
(case when rcvddelta>0 and sentdelta>0 then rcvddelta+sentdelta when rcvddelta>0 then rcvddelta
when sentdelta>0 then sentdelta else 0 end), 0) AS bandwidth, min(coalesce(dtime, 0)) as s_time,
max(coalesce(dtime, 0)) as e_time, coalesce(nullifna(`xauthuser`), nullifna(`user`), ipstr
(`remip`)) as f_user, tunneltype, action, count(*) as total_num  from (select itime, dtime, devid,
vd, remip, `xauthuser`, `user`, duration, tunnelid, tunnelip, tunneltype, action, sentbyte-lag
(coalesce(sentbyte, 0)) OVER (PARTITION BY dvid, tunnelid ORDER BY eventtime) AS sentdelta,
rcvdbyte-lag(coalesce(rcvdbyte, 0)) OVER (PARTITION BY dvid, tunnelid ORDER BY eventtime) AS
rcvddelta from $log where $filter and subtype='vpn' and tunnelid is not null) t where (tunneltype
like 'ipsec%' or tunneltype like 'ssl%') and action in ('tunnel-up','tunnel-stats', 'tunnel-down',
'ssl-login-fail', 'ipsec-login-fail') group by timestamp, devid, vd, remip, t_type, tunnelid,
action, f_user, tunneltype /*SkipSTART*/order by bandwidth desc/*SkipEND*/)### t where t_type like
```

```
'ssl%' and action in ('tunnel-up','tunnel-stats', 'tunnel-down') and tunnelid is not null and
tunnelid!=0 group by remote_ip having sum(traffic_in+traffic_out)>0 order by bandwidth desc
```

Dataset Name	Description	Log Category
vpn-Login-Connection-Count-by-Type	VPN authenticated logins	event

```
select
  coalesce(
    xauthuser_agg,
    user_agg,
    ipstr(`remip`)
  ) as f_user,
  t_type as tunneltype,
  from_dtime(
    min(s_time)
  ) as start_time,
  count(distinct tunnelid) as total_num,
  sum(duration) as duration
from
  (
    select
      string_agg(
        distinct xauthuser_agg,
        & #039; ') as xauthuser_agg, string_agg(distinct user_agg, ' ') as user_agg, t_type,
devid, vd, remip, tunnelid, min(s_time) as s_time, max(e_time) as e_time, (case when min(s_
time)=max(e_time) then NULL else max(max_duration)-min(min_duration) end) as duration, sum
(bandwidth) as bandwidth, sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_out, count
(distinct tunnelid) as total_num from ###(select devid, vd, remip, nullifna(`xauthuser`) as
xauthuser_agg, nullifna(`user`) as user_agg, (case when tunneltype like 'ipsec%' then 'ipsec' else
tunneltype end) as t_type, tunnelid, tunnelip, min(coalesce(dtime, 0)) as s_time, max(coalesce
(dtime, 0)) as e_time, max(coalesce(duration,0)) as max_duration, min(coalesce(duration,0)) as
min_duration, coalesce(sum(case when sentdelta>0 then sentdelta else 0 end), 0) AS traffic_out,
coalesce(sum(case when rcvddelta>0 then rcvddelta else 0 end), 0) AS traffic_in, coalesce(sum(case
when rcvddelta>0 and sentdelta>0 then rcvddelta+sentdelta when rcvddelta>0 then rcvddelta when
sentdelta>0 then sentdelta else 0 end), 0) AS bandwidth, sum((case when action='tunnel-up' then 1
else 0 end)) as tunnelup from (select dtime, devid, vd, remip, `xauthuser`, `user`, duration,
tunnelid, tunnelip, tunneltype, action, sentbyte-lag(coalesce(sentbyte, 0)) OVER (PARTITION BY
dvid, tunnelid ORDER BY eventtime) AS sentdelta, rcvdbyte-lag(coalesce(rcvdbyte, 0)) OVER
(PARTITION BY dvid, tunnelid ORDER BY eventtime) AS rcvddelta from $log where $filter and
subtype='vpn' and action in ('tunnel-up','tunnel-stats', 'tunnel-down', 'ssl-login-fail', 'ipsec-
login-fail') and tunnelid is not null) t where (tunneltype like 'ipsec%' or tunneltype like
'ssl%') and tunnelid!=0 group by xauthuser_agg, user_agg, devid, vd, remip, t_type, tunnelid,
tunnelip order by bandwidth desc)### t group by t_type, devid, vd, remip, tunnelid) tt where
bandwidth>0 group by f_user, tunneltype order by total_num desc
```

Dataset Name	Description	Log Category
vpn-Login-User-Count-by-Type	VPN Login User Count by VPN Type	event

```
select
  type_agg,
  count(distinct f_user) as num_user
from
  (
    select
      coalesce(
        xauthuser_agg,
        user_agg,
        ipstr(`remip`)
      ) as f_user,
      string_agg(
        distinct t_type,
        & #039; ') as type_agg from (select string_agg(distinct xauthuser_agg, ' ') as xauthuser_
agg, string_agg(distinct user_agg, ' ') as user_agg, t_type, devid, vd, remip, tunnelid, sum
(bandwidth) as bandwidth from ###(select devid, vd, remip, nullifna(`xauthuser`) as xauthuser_agg,
nullifna(`user`) as user_agg, (case when tunneltype like 'ipsec%' then 'ipsec' else tunneltype
end) as t_type, tunnelid, tunnelip, min(coalesce(dtime, 0)) as s_time, max(coalesce(dtime, 0)) as
e_time, max(coalesce(duration,0)) as max_duration, min(coalesce(duration,0)) as min_duration,
coalesce(sum(case when sentdelta>0 then sentdelta else 0 end), 0) AS traffic_out, coalesce(sum
(case when rcvddelta>0 then rcvddelta else 0 end), 0) AS traffic_in, coalesce(sum(case when
rcvddelta>0 and sentdelta>0 then rcvddelta+sentdelta when rcvddelta>0 then rcvddelta when
sentdelta>0 then sentdelta else 0 end), 0) AS bandwidth, sum((case when action='tunnel-up' then 1
else 0 end)) as tunnelup from (select dtime, devid, vd, remip, `xauthuser`, `user`, duration,
tunnelid, tunnelip, tunneltype, action, sentbyte-lag(coalesce(sentbyte, 0)) OVER (PARTITION BY
dvid, tunnelid ORDER BY eventtime) AS sentdelta, rcvdbyte-lag(coalesce(rcvdbyte, 0)) OVER
(PARTITION BY dvid, tunnelid ORDER BY eventtime) AS rcvddelta from $log where $filter and
subtype='vpn' and action in ('tunnel-up','tunnel-stats', 'tunnel-down', 'ssl-login-fail', 'ipsec-
login-fail') and tunnelid is not null) t where (tunneltype like 'ipsec%' or tunneltype like
'ssl%') and tunnelid!=0 group by xauthuser_agg, user_agg, devid, vd, remip, t_type, tunnelid,
tunnelip order by bandwidth desc)### t group by t_type, devid, vd, remip, tunnelid) tt where
bandwidth>0 group by f_user) ttt group by type_agg order by num_user desc
```

Dataset Name	Description	Log Category
vpn-Login-Total-Bandwidth-by-Type	VPN Login Total Bandwidth by VPN Type	event

```
select
  t_type,
  sum(bandwidth) as total_bandwidth
from
  ###(select devid, vd, remip, nullifna(`xauthuser`) as xauthuser_agg, nullifna(`user`) as user_
agg, (case when tunneltype like 'ipsec%' then 'ipsec' else tunneltype end) as t_type, tunnelid,
tunnelip, min(coalesce(dtime, 0)) as s_time, max(coalesce(dtime, 0)) as e_time, max(coalesce
(duration,0)) as max_duration, min(coalesce(duration,0)) as min_duration, coalesce(sum(case when
sentdelta>0 then sentdelta else 0 end), 0) AS traffic_out, coalesce(sum(case when rcvddelta>0 then
rcvddelta else 0 end), 0) AS traffic_in, coalesce(sum(case when rcvddelta>0 and sentdelta>0 then
rcvddelta+sentdelta when rcvddelta>0 then rcvddelta when sentdelta>0 then sentdelta else 0 end),
0) AS bandwidth, sum((case when action='tunnel-up' then 1 else 0 end)) as tunnelup from (select
dtime, devid, vd, remip, `xauthuser`, `user`, duration, tunnelid, tunnelip, tunneltype, action,
```

```
sentbyte-lag(coalesce(sentbyte, 0)) OVER (PARTITION BY dvid, tunnelid ORDER BY eventtime) AS
sentdelta, rcvdbyte-lag(coalesce(rcvdbyte, 0)) OVER (PARTITION BY dvid, tunnelid ORDER BY
eventtime) AS rcvddelta from $log where $filter and subtype='vpn' and action in ('tunnel-
up','tunnel-stats', 'tunnel-down', 'ssl-login-fail', 'ipsec-login-fail') and tunnelid is not null)
t where (tunneltype like 'ipsec%' or tunneltype like 'ssl%') and tunnelid!=0 group by xauthuser_
agg, user_agg, devid, vd, remip, t_type, tunnelid, tunnelip order by bandwidth desc)### t where
bandwidth>0 group by t_type order by total_bandwidth desc
```

Dataset Name	Description	Log Category
vpn-Login-Attempt-by-Type	VPN Login Attempts by VPN Type	event

```
select
  type,
  sum(total_num) as total_num
from
  (
    select
      (
        case when action in (
          & #039;ssl-login-fail', 'ipsec-login-fail') then 'Failed' else 'Success' end) as type,
(case when action in ('ssl-login-fail', 'ipsec-login-fail') then sum(total_num) else count
(distinct tunnelid) end) as total_num from ###(select coalesce(nullifna(`xauthuser`), nullifna
(`user`), ipstr(`remip`)) as f_user, tunneltype, tunnelid, action, count(*) as total_num from $log
where $filter and subtype='vpn' and (tunneltype like 'ipsec%' or tunneltype like 'ssl%') and
action in ('ssl-login-fail', 'ipsec-login-fail', 'tunnel-up') group by f_user, tunneltype,
tunnelid, action order by total_num desc)### t group by action) t group by type having sum(total_
num)>0 order by total_num desc
```

Dataset Name	Description	Log Category
vpn-Traffic-Usage-Trend	VPN traffic usage trend	event

```
select
  hodex,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out,
  sum(ssl_traffic_bandwidth) as ssl_bandwidth,
  sum(ipsec_traffic_bandwidth) as ipsec_bandwidth
from
  (
    select
      $flex_timescale(timestamp) as hodex,
      devid,
      vd,
      remip,
      tunnelid,
      sum(traffic_in) as traffic_in,
      sum(traffic_out) as traffic_out,
```

```
      (
        case when t_type like & #039;ssl%' then sum(bandwidth) else 0 end) as ssl_traffic_
bandwidth, (case when t_type like 'ipsec%' then sum(bandwidth)  else 0 end) as ipsec_traffic_
bandwidth, min(s_time) as s_time, max(e_time) as e_time from ###(select $flex_timestamp as
timestamp, devid, vd, remip, tunnelid, (case when tunneltype like 'ipsec%' then 'ipsec' else
tunneltype end) as t_type, (case when action='tunnel-up' then 1 else 0 end) as tunnelup, coalesce
(sum(case when sentdelta>0 then sentdelta else 0 end), 0) AS traffic_out, coalesce(sum(case when
rcvddelta>0 then rcvddelta else 0 end), 0) AS traffic_in, coalesce(sum(case when rcvddelta>0 and
sentdelta>0 then rcvddelta+sentdelta when rcvddelta>0 then rcvddelta when sentdelta>0 then
sentdelta else 0 end), 0) AS bandwidth, min(coalesce(dtime, 0)) as s_time, max(coalesce(dtime, 0))
as e_time, coalesce(nullifna(`xauthuser`), nullifna(`user`), ipstr(`remip`)) as f_user,
tunneltype, action, count(*) as total_num  from (select itime, dtime, devid, vd, remip,
`xauthuser`, `user`, duration, tunnelid, tunnelip, tunneltype, action, sentbyte-lag(coalesce
(sentbyte, 0)) OVER (PARTITION BY dvid, tunnelid ORDER BY eventtime) AS sentdelta, rcvdbyte-lag
(coalesce(rcvdbyte, 0)) OVER (PARTITION BY dvid, tunnelid ORDER BY eventtime) AS rcvddelta from
$log where $filter and subtype='vpn' and tunnelid is not null) t where (tunneltype like 'ipsec%'
or tunneltype like 'ssl%') and action in ('tunnel-up','tunnel-stats', 'tunnel-down', 'ssl-login-
fail', 'ipsec-login-fail') group by timestamp, devid, vd, remip, t_type, tunnelid, action, f_user,
tunneltype /*SkipSTART*/order by bandwidth desc/*SkipEND*/)### t where action in ('tunnel-
up','tunnel-stats', 'tunnel-down') and tunnelid is not null and tunnelid!=0 group by hodex, devid,
t_type, vd, remip, tunnelid) tt group by hodex order by hodex
```

Dataset Name	Description	Log Category
vpn-Authenticated-Logins	VPN authenticated logins	event

```
select
  coalesce(
    xauthuser_agg,
    user_agg,
    ipstr(`remip`)
  ) as f_user,
  t_type as tunneltype,
  from_dtime(
    min(s_time)
  ) as start_time,
  count(distinct tunnelid) as total_num,
  sum(duration) as duration
from
  (
    select
      string_agg(
        distinct xauthuser_agg,
        & #039; ') as xauthuser_agg, string_agg(distinct user_agg, ' ') as user_agg, t_type,
devid, vd, remip, tunnelid, min(s_time) as s_time, max(e_time) as e_time, (case when min(s_
time)=max(e_time) then NULL else max(max_duration)-min(min_duration) end) as duration, sum
(bandwidth) as bandwidth, sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_out, count
(distinct tunnelid) as total_num from ###(select devid, vd, remip, nullifna(`xauthuser`) as
xauthuser_agg, nullifna(`user`) as user_agg, (case when tunneltype like 'ipsec%' then 'ipsec' else
tunneltype end) as t_type, tunnelid, tunnelip, min(coalesce(dtime, 0)) as s_time, max(coalesce
(dtime, 0)) as e_time, max(coalesce(duration,0)) as max_duration, min(coalesce(duration,0)) as
```

```
min_duration, coalesce(sum(case when sentdelta>0 then sentdelta else 0 end), 0) AS traffic_out,
coalesce(sum(case when rcvddelta>0 then rcvddelta else 0 end), 0) AS traffic_in, coalesce(sum(case
when rcvddelta>0 and sentdelta>0 then rcvddelta+sentdelta when rcvddelta>0 then rcvddelta when
sentdelta>0 then sentdelta else 0 end), 0) AS bandwidth, sum((case when action='tunnel-up' then 1
else 0 end)) as tunnelup from (select dtime, devid, vd, remip, `xauthuser`, `user`, duration,
tunnelid, tunnelip, tunneltype, action, sentbyte-lag(coalesce(sentbyte, 0)) OVER (PARTITION BY
dvid, tunnelid ORDER BY eventtime) AS sentdelta, rcvdbyte-lag(coalesce(rcvdbyte, 0)) OVER
(PARTITION BY dvid, tunnelid ORDER BY eventtime) AS rcvddelta from $log where $filter and
subtype='vpn' and action in ('tunnel-up','tunnel-stats', 'tunnel-down', 'ssl-login-fail', 'ipsec-
login-fail') and tunnelid is not null) t where (tunneltype like 'ipsec%' or tunneltype like
'ssl%') and tunnelid!=0 group by xauthuser_agg, user_agg, devid, vd, remip, t_type, tunnelid,
tunnelip order by bandwidth desc)### t group by t_type, devid, vd, remip, tunnelid) tt where
bandwidth>0 group by f_user, tunneltype order by total_num desc
```

Dataset Name	Description	Log Category
vpn-Failed-Login-Attempt-by-User	VPN failed logins	event

```
select
  f_user,
  tunneltype,
  sum(total_num) as total_num
from
  ###(select coalesce(nullifna(`xauthuser`), `user`) as f_user, tunneltype, count(*) as total_num
from $log where $filter and subtype='vpn' and (tunneltype like 'ipsec%' or tunneltype like 'ssl%')
and action in ('ssl-login-fail', 'ipsec-login-fail') and coalesce(nullifna(`xauthuser`), nullifna
(`user`)) is not null group by f_user, tunneltype order by total_num desc)### t group by f_user,
tunneltype order by total_num desc
```

Dataset Name	Description	Log Category
vpn-Failed-Login-Timeline	VPN Failed Login Timeline	event

```
select
  $flex_timescale(timestamp) as hodex,
  sum(total_num) as total_num
from
  ###(select $flex_timestamp as timestamp, devid, vd, remip, tunnelid, (case when tunneltype like
'ipsec%' then 'ipsec' else tunneltype end) as t_type, (case when action='tunnel-up' then 1 else 0
end) as tunnelup, coalesce(sum(case when sentdelta>0 then sentdelta else 0 end), 0) AS traffic_
out, coalesce(sum(case when rcvddelta>0 then rcvddelta else 0 end), 0) AS traffic_in, coalesce(sum
(case when rcvddelta>0 and sentdelta>0 then rcvddelta+sentdelta when rcvddelta>0 then rcvddelta
when sentdelta>0 then sentdelta else 0 end), 0) AS bandwidth, min(coalesce(dtime, 0)) as s_time,
max(coalesce(dtime, 0)) as e_time, coalesce(nullifna(`xauthuser`), nullifna(`user`), ipstr
(`remip`)) as f_user, tunneltype, action, count(*) as total_num  from (select itime, dtime, devid,
vd, remip, `xauthuser`, `user`, duration, tunnelid, tunnelip, tunneltype, action, sentbyte-lag
(coalesce(sentbyte, 0)) OVER (PARTITION BY dvid, tunnelid ORDER BY eventtime) AS sentdelta,
rcvdbyte-lag(coalesce(rcvdbyte, 0)) OVER (PARTITION BY dvid, tunnelid ORDER BY eventtime) AS
rcvddelta from $log where $filter and subtype='vpn' and tunnelid is not null) t where (tunneltype
```

```
like 'ipsec%' or tunneltype like 'ssl%') and action in ('tunnel-up','tunnel-stats', 'tunnel-down',
'ssl-login-fail', 'ipsec-login-fail') group by timestamp, devid, vd, remip, t_type, tunnelid,
action, f_user, tunneltype /*SkipSTART*/order by bandwidth desc/*SkipEND*/)### t where action in
('ssl-login-fail', 'ipsec-login-fail') and f_user is not null group by hodex order by total_num
desc
```

Dataset Name	Description	Log Category
vpn-Top-Dial-Up-VPN-Users-By-Duration	Top dial up VPN users by duration	event

```
select
  coalesce(
    xauthuser_agg,
    user_agg,
    ipstr(`remip`)
  ) as user_src,
  t_type as tunneltype,
  from_dtime(
    min(s_time)
  ) as start_time,
  sum(duration) as duration,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out
from
  (
    select
      devid,
      vd,
      remip,
      string_agg(
        distinct xauthuser_agg,
        & #039; ') as xauthuser_agg, string_agg(distinct user_agg, ' ') as user_agg, t_type,
tunnelid, min(s_time) as s_time, max(e_time) as e_time, (case when min(s_time)=max(e_time) then
max(max_duration) else max(max_duration)-min(min_duration) end) as duration, sum(bandwidth) as
bandwidth, sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_out from ###(select devid,
vd, remip, nullifna(`xauthuser`) as xauthuser_agg, nullifna(`user`) as user_agg, (case when
tunneltype like 'ipsec%' then 'ipsec' else tunneltype end) as t_type, tunnelid, tunnelip, min
(coalesce(dtime, 0)) as s_time, max(coalesce(dtime, 0)) as e_time, max(coalesce(duration,0)) as
max_duration, min(coalesce(duration,0)) as min_duration, coalesce(sum(case when sentdelta>0 then
sentdelta else 0 end), 0) AS traffic_out, coalesce(sum(case when rcvddelta>0 then rcvddelta else 0
end), 0) AS traffic_in, coalesce(sum(case when rcvddelta>0 and sentdelta>0 then
rcvddelta+sentdelta when rcvddelta>0 then rcvddelta when sentdelta>0 then sentdelta else 0 end),
0) AS bandwidth, sum((case when action='tunnel-up' then 1 else 0 end)) as tunnelup from (select
dtime, devid, vd, remip, `xauthuser`, `user`, duration, tunnelid, tunnelip, tunneltype, action,
sentbyte-lag(coalesce(sentbyte, 0)) OVER (PARTITION BY dvid, tunnelid ORDER BY eventtime) AS
sentdelta, rcvdbyte-lag(coalesce(rcvdbyte, 0)) OVER (PARTITION BY dvid, tunnelid ORDER BY
eventtime) AS rcvddelta from $log where $filter and subtype='vpn' and action in ('tunnel-
up','tunnel-stats', 'tunnel-down', 'ssl-login-fail', 'ipsec-login-fail') and tunnelid is not null)
t where (tunneltype like 'ipsec%' or tunneltype like 'ssl%') and tunnelid!=0 group by xauthuser_
```

```
agg, user_agg, devid, vd, remip, t_type, tunnelid, tunnelip order by bandwidth desc)### t where
(t_type like 'ssl%' or (t_type like 'ipsec%' and not (tunnelip is null or tunnelip='0.0.0.0')))
group by devid, vd, remip, t_type, tunnelid) tt where bandwidth>0 group by user_src, tunneltype
order by duration desc
```

Dataset Name	Description	Log Category
vpn-Top-SSL-VPN-Tunnel-Duration-By-Users	Top SSL VPN Tunnel Duration by Users	event

```
select
  user_src,
  sum(duration) as duration,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out
from
  (
    select
      devid,
      vd,
      remip,
      user_src,
      tunnelid,
      (
        case when min(s_time)= max(e_time) then max(max_duration) else max(max_duration)- min(min_
duration) end
      ) as duration,
      sum(traffic_in) as traffic_in,
      sum(traffic_out) as traffic_out,
      sum(bandwidth) as bandwidth
    from
      ###(select devid, vd, remip, coalesce(nullifna(`user`), ipstr(`remip`)) as user_src,
tunnelid, tunneltype, max(coalesce(duration,0)) as max_duration, min(coalesce(duration,0)) as min_
duration, min(coalesce(dtime, 0)) as s_time, max(coalesce(dtime, 0)) as e_time, coalesce(sum(case
when sentdelta>0 then sentdelta else 0 end), 0) AS traffic_out, coalesce(sum(case when rcvddelta>0
then rcvddelta else 0 end), 0) AS traffic_in, coalesce(sum(case when rcvddelta>0 and sentdelta>0
then rcvddelta+sentdelta when rcvddelta>0 then rcvddelta when sentdelta>0 then sentdelta else 0
end), 0) AS bandwidth from (select dtime, devid, vd, remip, `user`, duration, tunnelid,
tunneltype, sentbyte-lag(coalesce(sentbyte, 0)) OVER (PARTITION BY dvid, tunnelid ORDER BY
eventtime) AS sentdelta, rcvdbyte-lag(coalesce(rcvdbyte, 0)) OVER (PARTITION BY dvid, tunnelid
ORDER BY eventtime) AS rcvddelta from $log where $filter and subtype='vpn' and action in ('tunnel-
up','tunnel-stats', 'tunnel-down', 'ssl-login-fail', 'ipsec-login-fail') and tunnelid is not null
and tunnelid!=0) t where tunneltype like 'ssl%' and coalesce(nullifna(`user`), ipstr(`remip`)) is
not null group by devid, vd, user_src, remip, tunnelid, tunneltype order by bandwidth desc)### t
where tunneltype='ssl-tunnel' group by devid, vd, remip, user_src, tunnelid) tt where bandwidth>0
group by user_src order by duration desc
```

Dataset Name	Description	Log Category
vpn-Top-SSL-VPN-Tunnel-Users-By-Traffic-Directions	Top SSL VPN Tunnel Users by Traffic Directions	event

```
select
  user_src,
  unnest(traffic_direction) as direction,
  unnest(traffic) as traffic
from
  (
    select
      user_src,
      sum(bandwidth) as bandwidth,
      array[ & #039;Received', 'Sent'] as traffic_direction, array[sum(traffic_in), sum(traffic_
out)] as traffic from ###(select devid, vd, remip, coalesce(nullifna(`user`), ipstr(`remip`)) as
user_src, tunnelid, tunneltype, max(coalesce(duration,0)) as max_duration, min(coalesce
(duration,0)) as min_duration, min(coalesce(dtime, 0)) as s_time, max(coalesce(dtime, 0)) as e_
time, coalesce(sum(case when sentdelta>0 then sentdelta else 0 end), 0) AS traffic_out, coalesce
(sum(case when rcvddelta>0 then rcvddelta else 0 end), 0) AS traffic_in, coalesce(sum(case when
rcvddelta>0 and sentdelta>0 then rcvddelta+sentdelta when rcvddelta>0 then rcvddelta when
sentdelta>0 then sentdelta else 0 end), 0) AS bandwidth from (select dtime, devid, vd, remip,
`user`, duration, tunnelid, tunneltype, sentbyte-lag(coalesce(sentbyte, 0)) OVER (PARTITION BY
dvid, tunnelid ORDER BY eventtime) AS sentdelta, rcvdbyte-lag(coalesce(rcvdbyte, 0)) OVER
(PARTITION BY dvid, tunnelid ORDER BY eventtime) AS rcvddelta from $log where $filter and
subtype='vpn' and action in ('tunnel-up','tunnel-stats', 'tunnel-down', 'ssl-login-fail', 'ipsec-
login-fail') and tunnelid is not null and tunnelid!=0) t where tunneltype like 'ssl%' and coalesce
(nullifna(`user`), ipstr(`remip`)) is not null group by devid, vd, user_src, remip, tunnelid,
tunneltype order by bandwidth desc)### t where tunneltype='ssl-tunnel' and bandwidth>0 group by
user_src) ttt order by bandwidth desc
```

Dataset Name	Description	Log Category
vpn-Top-SSL-VPN-Web-Mode-Users-By-Duration	Top SSL VPN web mode users by duration	event

```
select
  user_src,
  remip as remote_ip,
  from_dtime(
    min(s_time)
  ) as start_time,
  sum(duration) as duration
from
  (
    select
      devid,
      vd,
      user_src,
      remip,
```

```
      tunnelid,
      min(s_time) as s_time,
      (
        case when min(s_time)= max(e_time) then max(max_duration) else max(max_duration)- min(min_
duration) end
      ) as duration
    from
      ###(select devid, vd, remip, coalesce(nullifna(`user`), ipstr(`remip`)) as user_src,
tunnelid, tunneltype, max(coalesce(duration,0)) as max_duration, min(coalesce(duration,0)) as min_
duration, min(coalesce(dtime, 0)) as s_time, max(coalesce(dtime, 0)) as e_time, coalesce(sum(case
when sentdelta>0 then sentdelta else 0 end), 0) AS traffic_out, coalesce(sum(case when rcvddelta>0
then rcvddelta else 0 end), 0) AS traffic_in, coalesce(sum(case when rcvddelta>0 and sentdelta>0
then rcvddelta+sentdelta when rcvddelta>0 then rcvddelta when sentdelta>0 then sentdelta else 0
end), 0) AS bandwidth from (select dtime, devid, vd, remip, `user`, duration, tunnelid,
tunneltype, sentbyte-lag(coalesce(sentbyte, 0)) OVER (PARTITION BY dvid, tunnelid ORDER BY
eventtime) AS sentdelta, rcvdbyte-lag(coalesce(rcvdbyte, 0)) OVER (PARTITION BY dvid, tunnelid
ORDER BY eventtime) AS rcvddelta from $log where $filter and subtype='vpn' and action in ('tunnel-
up','tunnel-stats', 'tunnel-down', 'ssl-login-fail', 'ipsec-login-fail') and tunnelid is not null
and tunnelid!=0) t where tunneltype like 'ssl%' and coalesce(nullifna(`user`), ipstr(`remip`)) is
not null group by devid, vd, user_src, remip, tunnelid, tunneltype order by bandwidth desc)### t
where tunneltype='ssl-web' group by devid, vd, user_src, remip, tunnelid) tt group by user_src,
remote_ip order by duration desc
```

Dataset Name	Description	Log Category
vpn-Top-SSL-VPN-Web-Mode-Users-By-Traffic-Directions	Top SSL VPN Web Mode Users by Traffic Directions	event

```
select
  user_src,
  unnest(traffic_direction) as direction,
  unnest(traffic) as traffic
from
  (
    select
      user_src,
      sum(bandwidth) as bandwidth,
      array[ & #039;Received', 'Sent'] as traffic_direction, array[sum(traffic_in), sum(traffic_
out)] as traffic from ###(select devid, vd, remip, coalesce(nullifna(`user`), ipstr(`remip`)) as
user_src, tunnelid, tunneltype, max(coalesce(duration,0)) as max_duration, min(coalesce
(duration,0)) as min_duration, min(coalesce(dtime, 0)) as s_time, max(coalesce(dtime, 0)) as e_
time, coalesce(sum(case when sentdelta>0 then sentdelta else 0 end), 0) AS traffic_out, coalesce
(sum(case when rcvddelta>0 then rcvddelta else 0 end), 0) AS traffic_in, coalesce(sum(case when
rcvddelta>0 and sentdelta>0 then rcvddelta+sentdelta when rcvddelta>0 then rcvddelta when
sentdelta>0 then sentdelta else 0 end), 0) AS bandwidth from (select dtime, devid, vd, remip,
`user`, duration, tunnelid, tunneltype, sentbyte-lag(coalesce(sentbyte, 0)) OVER (PARTITION BY
dvid, tunnelid ORDER BY eventtime) AS sentdelta, rcvdbyte-lag(coalesce(rcvdbyte, 0)) OVER
(PARTITION BY dvid, tunnelid ORDER BY eventtime) AS rcvddelta from $log where $filter and
subtype='vpn' and action in ('tunnel-up','tunnel-stats', 'tunnel-down', 'ssl-login-fail', 'ipsec-
login-fail') and tunnelid is not null and tunnelid!=0) t where tunneltype like 'ssl%' and coalesce
(nullifna(`user`), ipstr(`remip`)) is not null group by devid, vd, user_src, remip, tunnelid,
```

```
tunneltype order by bandwidth desc)### t where tunneltype='ssl-web' and bandwidth>0 group by user_
src) ttt order by bandwidth desc
```

Dataset Name	Description	Log Category
vpn-Top-IPsec-Vpn-Dial-Up-User-By-Bandwidth	Default top IPsec VPN dial up user by bandwidth usage	event

```
select
  coalesce(
    xauthuser_agg,
    user_agg,
    ipstr(`remip`)
  ) as user_src,
  from_dtime(
    min(s_time)
  ) as start_time,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out
from
  (
    select
      devid,
      vd,
      string_agg(
        distinct xauthuser_agg,
        & #039; ') as xauthuser_agg, string_agg(distinct user_agg, ' ') as user_agg, remip,
tunnelid, min(s_time) as s_time, max(e_time) as e_time, sum(bandwidth) as bandwidth, sum(traffic_
in) as traffic_in, sum(traffic_out) as traffic_out from ###(select devid, vd, remip, nullifna
(`xauthuser`) as xauthuser_agg, nullifna(`user`) as user_agg, tunnelid, max(coalesce(duration,0))
as max_duration, min(coalesce(duration,0)) as min_duration, min(coalesce(dtime, 0)) as s_time, max
(coalesce(dtime, 0)) as e_time, coalesce(sum(case when sentdelta>0 then sentdelta else 0 end), 0)
AS traffic_out, coalesce(sum(case when rcvddelta>0 then rcvddelta else 0 end), 0) AS traffic_in,
coalesce(sum(case when rcvddelta>0 and sentdelta>0 then rcvddelta+sentdelta when rcvddelta>0 then
rcvddelta when sentdelta>0 then sentdelta else 0 end), 0) AS bandwidth from (select dtime, devid,
vd, remip, `xauthuser`, `user`, duration, tunnelid, tunnelip, tunneltype, sentbyte-lag(coalesce
(sentbyte, 0)) OVER (PARTITION BY dvid, tunnelid ORDER BY eventtime) AS sentdelta, rcvdbyte-lag
(coalesce(rcvdbyte, 0)) OVER (PARTITION BY dvid, tunnelid ORDER BY eventtime) AS rcvddelta from
$log where $filter and subtype='vpn' and action in ('tunnel-up','tunnel-stats', 'tunnel-down',
'ssl-login-fail', 'ipsec-login-fail') and tunnelid is not null) t where tunneltype like 'ipsec%'
and not (tunnelip is null or tunnelip='0.0.0.0') and tunnelid!=0 group by devid, vd, remip,
xauthuser_agg, user_agg, tunnelid order by bandwidth desc)### t group by devid, vd, remip,
tunnelid) tt group by user_src having sum(bandwidth)>0 order by bandwidth desc
```

Dataset Name	Description	Log Category
vpn-Top-Static-IPsec-Tunnels-By-Traffic-Directions	Top Static IPsec Tunnels by Traffic Directions	event

```
select
  vpn_name,
  unnest(traffic_direction) as direction,
  unnest(traffic) as traffic
from
  (
    select
      vpn_name,
      sum(bandwidth) as bandwidth,
      array[ & #039;Received', 'Sent'] as traffic_direction, array[sum(traffic_in), sum(traffic_
out)] as traffic from ###(select devid, vd, remip, vpn_trim(vpntunnel) as vpn_name, tunnelid,
tunnelip, coalesce(sum(case when sentdelta>0 then sentdelta else 0 end), 0) AS traffic_out,
coalesce(sum(case when rcvddelta>0 then rcvddelta else 0 end), 0) AS traffic_in, coalesce(sum(case
when rcvddelta>0 and sentdelta>0 then rcvddelta+sentdelta when rcvddelta>0 then rcvddelta when
sentdelta>0 then sentdelta else 0 end), 0) AS bandwidth from (select devid, vd, remip, tunnelid,
tunnelip,vpntunnel, tunneltype, action, sentbyte-lag(coalesce(sentbyte, 0)) OVER (PARTITION BY
dvid, tunnelid ORDER BY eventtime) AS sentdelta, rcvdbyte-lag(coalesce(rcvdbyte, 0)) OVER
(PARTITION BY dvid, tunnelid ORDER BY eventtime) AS rcvddelta from $log where $filter and
subtype='vpn' and action in ('tunnel-up','tunnel-stats', 'tunnel-down', 'ssl-login-fail', 'ipsec-
login-fail') and tunnelid is not null) t where tunneltype like 'ipsec%' and nullifna(vpntunnel) is
not null and tunnelid!=0 group by devid, vd, remip, vpn_name, tunnelid, tunnelip order by
bandwidth desc)### t where (tunnelip is null or tunnelip='0.0.0.0') group by vpn_name having sum
(traffic_in+traffic_out)>0) ttt order by bandwidth desc
```

Dataset Name	Description	Log Category
vpn-Top-Dial-Up-IPsec-Users-By-Duration	Top dial up IPsec users by duration	event

```
select
  coalesce(
    xauthuser_agg,
    user_agg,
    ipstr(`remip`)
  ) as user_src,
  from_dtime(
    min(s_time)
  ) as start_time,
  sum(duration) as duration,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out
from
  (
    select
      devid,
      vd,
      remip,
      string_agg(
        distinct xauthuser_agg,
        & #039; ') as xauthuser_agg, string_agg(distinct user_agg, ' ') as user_agg, tunnelid, min
```

```
(s_time) as s_time, max(e_time) as e_time, (case when min(s_time)=max(e_time) then max(max_
duration) else max(max_duration)-min(min_duration) end) as duration, sum(bandwidth) as bandwidth,
sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_out from ###(select devid, vd, remip,
nullifna(`xauthuser`) as xauthuser_agg, nullifna(`user`) as user_agg, tunnelid, max(coalesce
(duration,0)) as max_duration, min(coalesce(duration,0)) as min_duration, min(coalesce(dtime, 0))
as s_time, max(coalesce(dtime, 0)) as e_time, coalesce(sum(case when sentdelta>0 then sentdelta
else 0 end), 0) AS traffic_out, coalesce(sum(case when rcvddelta>0 then rcvddelta else 0 end), 0)
AS traffic_in, coalesce(sum(case when rcvddelta>0 and sentdelta>0 then rcvddelta+sentdelta when
rcvddelta>0 then rcvddelta when sentdelta>0 then sentdelta else 0 end), 0) AS bandwidth from
(select dtime, devid, vd, remip, `xauthuser`, `user`, duration, tunnelid, tunnelip, tunneltype,
sentbyte-lag(coalesce(sentbyte, 0)) OVER (PARTITION BY dvid, tunnelid ORDER BY eventtime) AS
sentdelta, rcvdbyte-lag(coalesce(rcvdbyte, 0)) OVER (PARTITION BY dvid, tunnelid ORDER BY
eventtime) AS rcvddelta from $log where $filter and subtype='vpn' and action in ('tunnel-
up','tunnel-stats', 'tunnel-down', 'ssl-login-fail', 'ipsec-login-fail') and tunnelid is not null)
t where tunneltype like 'ipsec%' and not (tunnelip is null or tunnelip='0.0.0.0') and tunnelid!=0
group by devid, vd, remip, xauthuser_agg, user_agg, tunnelid order by bandwidth desc)### t group
by devid, vd, remip, tunnelid) tt where bandwidth>0 group by user_src order by duration desc
```

Dataset Name	Description	Log Category
vpn-Top-Dial-Up-IPsec-Tunnels-By-Traffic-Directions	Top Dial Up IPsec Tunnels by Traffic Directions	event

```
select
  vpn_name,
  unnest(traffic_direction) as direction,
  unnest(traffic) as traffic
from
  (
    select
      vpn_name,
      sum(bandwidth) as bandwidth,
      array[ & #039;Received', 'Sent'] as traffic_direction, array[sum(traffic_in), sum(traffic_
out)] as traffic from ###(select devid, vd, remip, vpn_trim(vpntunnel) as vpn_name, tunnelid,
tunnelip, coalesce(sum(case when sentdelta>0 then sentdelta else 0 end), 0) AS traffic_out,
coalesce(sum(case when rcvddelta>0 then rcvddelta else 0 end), 0) AS traffic_in, coalesce(sum(case
when rcvddelta>0 and sentdelta>0 then rcvddelta+sentdelta when rcvddelta>0 then rcvddelta when
sentdelta>0 then sentdelta else 0 end), 0) AS bandwidth from (select devid, vd, remip, tunnelid,
tunnelip,vpntunnel, tunneltype, action, sentbyte-lag(coalesce(sentbyte, 0)) OVER (PARTITION BY
dvid, tunnelid ORDER BY eventtime) AS sentdelta, rcvdbyte-lag(coalesce(rcvdbyte, 0)) OVER
(PARTITION BY dvid, tunnelid ORDER BY eventtime) AS rcvddelta from $log where $filter and
subtype='vpn' and action in ('tunnel-up','tunnel-stats', 'tunnel-down', 'ssl-login-fail', 'ipsec-
login-fail') and tunnelid is not null) t where tunneltype like 'ipsec%' and nullifna(vpntunnel) is
not null and tunnelid!=0 group by devid, vd, remip, vpn_name, tunnelid, tunnelip order by
bandwidth desc)### t where not (tunnelip is null or tunnelip='0.0.0.0') group by vpn_name having
sum(bandwidth)>0) ttt order by bandwidth desc
```

Dataset Name	Description	Log Category
webfilter-Web-Activity-Summary-By-Requests	Webfilter web activity summary by requests	webfilter

```
select
  $flex_timescale(timestamp) as hodex,
  sum(allowed_request) as allowed_request,
  sum(blocked_request) as blocked_request
from
  ###(select $flex_timestamp as timestamp, sum(case when action!='blocked' then 1 else 0 end) as
allowed_request, sum(case when action='blocked' then 1 else 0 end) as blocked_request from $log
where $filter group by timestamp /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t group by
hodex order by hodex
```

Dataset Name	Description	Log Category
traffic-Browsing-Time-Summary	Traffic browsing time summary	traffic

```
select
  $flex_timescale(timestamp) as hodex,
  cast(
    ebtr_value(
      ebtr_agg_flat(browsetime),
      null,
      $timespan
    )/ 60.0 as decimal(18, 2)
  ) as browsetime
from
  ###(select $flex_timestamp as timestamp, ebtr_agg_flat($browse_time) as browsetime from $log
where $filter and (logflag&1>0) and $browse_time is not null group by timestamp /*SkipSTART*/order
by timestamp desc/*SkipEND*/)### t group by hodex order by hodex
```

Dataset Name	Description	Log Category
webfilter-Top-Web-Users-By-Blocked-Requests	Webfilter top web users by blocked requests	webfilter

```
select
  coalesce(
    f_user,
    euname,
    ipstr(`srcip`)
  ) as user_src,
  coalesce(
    epname,
    ipstr(`srcip`)
  ) as ep_src,
```

```
  sum(requests) as requests
from
  (
    select
      dvid,
      f_user,
      srcip,
      ep_id,
      eu_id,
      sum(requests) as requests
    from
      ###(select dvid, coalesce(nullifna(`user`), nullifna(`unauthuser`)) as f_user, srcip, (case
when epid<1024 then null else epid end) as ep_id, (case when euid<1024 then null else euid end) as
eu_id, action, count(*) as requests from $log where $filter and coalesce(nullifna(`user`),
nullifna(`unauthuser`), ipstr(`srcip`)) is not null group by dvid, f_user, srcip, ep_id, eu_id,
action /*SkipSTART*/order by requests desc/*SkipEND*/)### t where action='blocked' group by dvid,
f_user, srcip, ep_id, eu_id order by requests desc) t1 left join (select epid, euid, srcmac as
epmac, dvid from $ADOM_EPEU_DEVMAP dm inner join devtable dt ON dm.devid=dt.devid and dm.vd=dt.vd)
t2 on t1.ep_id=t2.epid and t1.eu_id=t2.euid and t1.dvid=t2.dvid left join $ADOM_ENDPOINT t3 on
t1.ep_id=t3.epid and t2.epmac=t3.mac and left(t3.epdevtype, 6)!='Router' left join $ADOM_ENDUSER
t4 on t1.eu_id=t4.euid group by user_src, ep_src order by requests desc
```

Dataset Name	Description	Log Category
webfilter-Top-Web-Users-By-Allowed-Requests	Webfilter top web users by allowed requests	webfilter

```
select
  coalesce(
    f_user,
    euname,
    ipstr(`srcip`)
  ) as user_src,
  coalesce(
    epname,
    ipstr(`srcip`)
  ) as ep_src,
  sum(requests) as requests
from
  (
    select
      dvid,
      f_user,
      srcip,
      ep_id,
      eu_id,
      sum(requests) as requests
    from
      ###(select dvid, coalesce(nullifna(`user`), nullifna(`unauthuser`)) as f_user, srcip, (case
when epid<1024 then null else epid end) as ep_id, (case when euid<1024 then null else euid end) as
eu_id, action, count(*) as requests from $log where $filter and coalesce(nullifna(`user`),
```

```
nullifna(`unauthuser`), ipstr(`srcip`)) is not null group by dvid, f_user, srcip, ep_id, eu_id,
action /*SkipSTART*/order by requests desc/*SkipEND*/)### t where action!='blocked' group by dvid,
f_user, srcip, ep_id, eu_id order by requests desc) t1 left join (select epid, euid, srcmac as
epmac, dvid from $ADOM_EPEU_DEVMAP dm inner join devtable dt ON dm.devid=dt.devid and dm.vd=dt.vd)
t2 on t1.ep_id=t2.epid and t1.eu_id=t2.euid and t1.dvid=t2.dvid left join $ADOM_ENDPOINT t3 on
t1.ep_id=t3.epid and t2.epmac=t3.mac and left(t3.epdevtype, 6)!='Router' left join $ADOM_ENDUSER
t4 on t1.eu_id=t4.euid group by user_src, ep_src order by requests desc
```

Dataset Name	Description	Log Category
traffic-Top-Web-Users-By-Browsing-Time	Traffic top web users by browsing time	traffic

```
select
  user_src,
  ebtr_value(
    ebtr_agg_flat(browsetime),
    null,
    $timespan
  ) as browsetime,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out
from
  ###(select user_src, ebtr_agg_flat(browsetime) as browsetime, sum(bandwidth) as bandwidth, sum
(traffic_in) as traffic_in, sum(traffic_out) as traffic_out from (select coalesce(nullifna
(`user`), ipstr(`srcip`)) as user_src, ebtr_agg_flat($browse_time) as browsetime, sum(coalesce
(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentbyte, 0)) as traffic_out from $log where $filter and $browse_time is not null group
by user_src) t group by user_src /*SkipSTART*/order by ebtr_value(ebtr_agg_flat(browsetime), null,
null) desc/*SkipEND*/)### t group by user_src order by browsetime desc
```

Dataset Name	Description	Log Category
webfilter-Top-Blocked-Web-Sites-By-Requests	Webfilter top blocked web sites by requests	webfilter

```
select
  domain,
  catdesc,
  sum(requests) as requests
from
  ###(select hostname as domain, catdesc, action, count(*) as requests from $log where $filter and
hostname is not null and catdesc is not null group by domain, catdesc, action /*SkipSTART*/order
by requests desc/*SkipEND*/)### t where action='blocked' group by domain, catdesc order by
requests desc
```

Dataset Name	Description	Log Category
webfilter-Top-Allowed-Web-Sites-By-Requests	Webfilter top allowed web sites by requests	webfilter

```
select
  domain,
  string_agg(
    distinct catdesc,
    & #039;, ') as agg_catdesc, sum(requests) as requests from ###(select hostname as domain,
catdesc, action, count(*) as requests from $log where $filter and hostname is not null and catdesc
is not null group by domain, catdesc, action /*SkipSTART*/order by requests desc/*SkipEND*/)### t
where action!='blocked' group by domain order by requests desc
```

Dataset Name	Description	Log Category
webfilter-Top-Video-Streaming-Websites-By-Bandwidth	Webfilter top video streaming websites by bandwidth usage	webfilter

```
select
  domain,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out
from
  ###(select coalesce(nullifna(root_domain(hostname)), 'other') as domain, sum(coalesce(sentbyte,
0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce
(sentbyte, 0)) as traffic_out from $log-traffic where $filter and (logflag&1>0) and (countweb>0 or
((logver is null or logver<502000000) and (hostname is not null or utmevent in ('webfilter',
'banned-word', 'web-content', 'command-block', 'script-filter')))) and catdesc in ('Streaming
Media and Download') group by domain having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0
/*SkipSTART*/order by bandwidth desc/*SkipEND*/)### t group by domain order by bandwidth desc
```

Dataset Name	Description	Log Category
webfilter-Top-Blocked-Web-Categories	Webfilter top blocked web categories	webfilter

```
select
  catdesc,
  sum(requests) as requests
from
  ###(select catdesc, action, count(*) as requests from $log-webfilter where $filter and catdesc
is not null group by catdesc, action /*SkipSTART*/order by requests desc/*SkipEND*/)### t where
action='blocked' group by catdesc order by requests desc
```

Dataset Name	Description	Log Category
webfilter-Top-Allowed-Web-Categories	Webfilter top allowed web categories	webfilter

```
select
  catdesc,
  sum(requests) as requests
from
  ###(select catdesc, action, count(*) as requests from $log-webfilter where $filter and catdesc
is not null group by catdesc, action /*SkipSTART*/order by requests desc/*SkipEND*/)### t where
action!='blocked' group by catdesc order by requests desc
```

Dataset Name	Description	Log Category
traffic-Top-50-Sites-By-Browsing-Time	Traffic top sites by browsing time	traffic

```
select
  hostname,
  string_agg(
    distinct catdesc,
    & #039;, ') as agg_catdesc, ebtr_value(ebtr_agg_flat(browsetime), null, $timespan) as
browsetime, sum(bandwidth) as bandwidth, sum(traffic_in) as traffic_in, sum(traffic_out) as
traffic_out from ###(select hostname, catdesc, ebtr_agg_flat(browsetime) as browsetime, sum
(bandwidth) as bandwidth, sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_out from
(select hostname, catdesc, ebtr_agg_flat($browse_time) as browsetime, sum(coalesce(sentbyte,
0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce
(sentbyte, 0)) as traffic_out from $log where $filter and (logflag&1>0) and hostname is not null
and $browse_time is not null group by hostname, catdesc) t group by hostname, catdesc
/*SkipSTART*/order by ebtr_value(ebtr_agg_flat(browsetime), null, null) desc/*SkipEND*/)### t
group by hostname order by browsetime desc
```

Dataset Name	Description	Log Category
traffic-Top-10-Categories-By-Browsing-Time	Traffic top category by browsing time	traffic

```
select
  catdesc,
  ebtr_value(
    ebtr_agg_flat(browsetime),
    null,
    $timespan
  ) as browsetime,
  sum(bandwidth) as bandwidth
from
  ###(select catdesc, ebtr_agg_flat(browsetime) as browsetime, sum(bandwidth) as bandwidth from
```

```
(select catdesc, ebtr_agg_flat($browse_time) as browsetime, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth from $log where $filter and (logflag&1>0) and catdesc is not null and
$browse_time is not null group by catdesc) t group by catdesc /*SkipSTART*/order by ebtr_value
(ebtr_agg_flat(browsetime), null, null) desc/*SkipEND*/)### t group by catdesc order by browsetime
desc
```

Dataset Name	Description	Log Category
traffic-Top-Destination-Countries-By-Browsing-Time	Traffic top destination countries by browsing time	traffic

```
select
  dstcountry,
  ebtr_value(
    ebtr_agg_flat(browsetime),
    null,
    $timespan
  ) as browsetime,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out
from
  ###(select dstcountry, ebtr_agg_flat(browsetime) as browsetime, sum(bandwidth) as bandwidth, sum
(traffic_in) as traffic_in, sum(traffic_out) as traffic_out from (select dstcountry, ebtr_agg_flat
($browse_time) as browsetime, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum
(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out from $log where
$filter and (logflag&1>0) and $browse_time is not null group by dstcountry) t group by dstcountry
/*SkipSTART*/order by ebtr_value(ebtr_agg_flat(browsetime), null, null) desc/*SkipEND*/)### t
group by dstcountry order by browsetime desc
```

Dataset Name	Description	Log Category
webfilter-Top-Search-Phrases	Webfilter top search phrases	webfilter

```
select
  keyword,
  count(*) as requests
from
  $log
where
  $filter
  and keyword is not null
group by
  keyword
order by
  requests desc
```

Dataset Name	Description	Log Category
Top-10-Users-Browsing-Time	Estimated browsing time	traffic

```
select
  coalesce(
    f_user,
    euname,
    ipstr(`srcip`)
  ) as user_src,
  coalesce(
    epname,
    ipstr(`srcip`)
  ) as ep_src,
  ebtr_value(
    ebtr_agg_flat(browsetime),
    null,
    $timespan
  ) as browsetime
from
  (
    select
      dvid,
      f_user,
      srcip,
      ep_id,
      eu_id,
      ebtr_agg_flat(browsetime) as browsetime
    from
      ###(select dvid, f_user, srcip, ep_id, eu_id, ebtr_agg_flat(browsetime) as browsetime from
(select dvid, coalesce(nullifna(`user`), nullifna(`unauthuser`)) as f_user, srcip, (case when
epid<1024 then null else epid end) as ep_id, (case when euid<1024 then null else euid end) as eu_
id, ebtr_agg_flat($browse_time) as browsetime from $log where $filter and (logflag&1>0) and
$browse_time is not null group by dvid, f_user, srcip, ep_id, eu_id) t group by dvid, f_user,
srcip, ep_id, eu_id order by ebtr_value(ebtr_agg_flat(browsetime), null, null) desc)### t group by
dvid, f_user, srcip, ep_id, eu_id order by ebtr_value(ebtr_agg_flat(browsetime), null, null) desc)
t1 left join (select epid, euid, srcmac as epmac, dvid from $ADOM_EPEU_DEVMAP dm inner join
devtable dt ON dm.devid=dt.devid and dm.vd=dt.vd) t2 on t1.ep_id=t2.epid and t1.eu_id=t2.euid and
t1.dvid=t2.dvid left join $ADOM_ENDPOINT t3 on t1.ep_id=t3.epid and t2.epmac=t3.mac and left
(t3.epdevtype, 6)!='Router' left join $ADOM_ENDUSER t4 on t1.eu_id=t4.euid group by user_src, ep_
src order by browsetime desc
```

Dataset Name	Description	Log Category
Estimated-Browsing-Time	Estimated browsing time	traffic

```
select
  coalesce(
    f_user,
    euname,
```

```
    ipstr(`srcip`)
  ) as user_src,
  coalesce(
    epname,
    ipstr(`srcip`)
  ) as ep_src,
  ebtr_value(
    ebtr_agg_flat(browsetime),
    null,
    $timespan
  ) as browsetime
from
  (
    select
      dvid,
      f_user,
      srcip,
      ep_id,
      eu_id,
      ebtr_agg_flat(browsetime) as browsetime
    from
      ###(select dvid, f_user, srcip, ep_id, eu_id, ebtr_agg_flat(browsetime) as browsetime from
(select dvid, coalesce(nullifna(`user`), nullifna(`unauthuser`)) as f_user, srcip, (case when
epid<1024 then null else epid end) as ep_id, (case when euid<1024 then null else euid end) as eu_
id, ebtr_agg_flat($browse_time) as browsetime from $log where $filter and (logflag&1>0) and
$browse_time is not null group by dvid, f_user, srcip, ep_id, eu_id) t group by dvid, f_user,
srcip, ep_id, eu_id order by ebtr_value(ebtr_agg_flat(browsetime), null, null) desc)### t group by
dvid, f_user, srcip, ep_id, eu_id order by ebtr_value(ebtr_agg_flat(browsetime), null, null) desc)
t1 left join (select epid, euid, srcmac as epmac, dvid from $ADOM_EPEU_DEVMAP dm inner join
devtable dt ON dm.devid=dt.devid and dm.vd=dt.vd) t2 on t1.ep_id=t2.epid and t1.eu_id=t2.euid and
t1.dvid=t2.dvid left join $ADOM_ENDPOINT t3 on t1.ep_id=t3.epid and t2.epmac=t3.mac and left
(t3.epdevtype, 6)!='Router' left join $ADOM_ENDUSER t4 on t1.eu_id=t4.euid group by user_src, ep_
src order by browsetime desc
```

Dataset Name	Description	Log Category
wifi-Top-AP-By-Bandwidth	Top access point by bandwidth usage	traffic

```
select
  ap_srcintf,
  sum(bandwidth) as bandwidth
from
  (
    select
      coalesce(ap, srcintf) as ap_srcintf,
      sum(bandwidth) as bandwidth
    from
      ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
ap, srcintf, srcssid, srcssid as ssid, srcmac, srcmac as stamac, coalesce(nullifna(`srcname`),
`srcmac`) as hostname_mac, max(srcswversion) as srcswversion, max(osname) as osname, max
(osversion) as osversion, max(devtype) as devtype, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte,
```

```
0)) as bandwidth, count(*) as subtotal from $log-traffic where $filter and (logflag&1>0) and
(srcssid is not null or dstssid is not null) group by user_src, ap, srcintf, srcssid, srcmac,
hostname_mac /*SkipSTART*/order by bandwidth desc, subtotal desc/*SkipEND*/)### t group by ap_
srcintf having sum(bandwidth)>0 union all select ap as ap_srcintf, sum(bandwidth) as bandwidth
from ###(select $flex_timestamp as timestamp, stamac, stamac as srcmac, ap, ssid, ssid as srcssid,
user_src, sum(coalesce(sentdelta, 0)) as sentdelta, sum(coalesce(rcvddelta, 0)) as rcvddelta, sum
(coalesce(sentdelta, 0)+coalesce(rcvddelta, 0)) as bandwidth from (select itime, stamac, ap, ssid,
coalesce(`user`, ipstr(`srcip`)) as user_src, sentbyte-lag(coalesce(sentbyte, 0)) over (partition
by stamac order by itime) as sentdelta, rcvdbyte-lag(coalesce(rcvdbyte, 0)) over (partition by
stamac order by itime) as rcvddelta from $log-event where $filter and subtype='wireless' and
stamac is not null and ssid is not null and action in ('sta-wl-bridge-traffic-stats', 'reassoc-
req', 'assoc-req')) as t group by timestamp, stamac, ap, ssid, user_src /*SkipSTART*/order by
bandwidth desc/*SkipEND*/)### t group by ap having sum(bandwidth)>0) t group by ap_srcintf order
by bandwidth desc
```

Dataset Name	Description	Log Category
wifi-Top-AP-By-Client	Top access point by client	traffic

```
select
  ap_srcintf as srcintf,
  count(distinct srcmac) as totalnum
from
  (
    select
      coalesce(ap, srcintf) as ap_srcintf,
      srcmac
    from
      ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
ap, srcintf, srcssid, srcssid as ssid, srcmac, srcmac as stamac, coalesce(nullifna(`srcname`),
`srcmac`) as hostname_mac, max(srcswversion) as srcswversion, max(osname) as osname, max
(osversion) as osversion, max(devtype) as devtype, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte,
0)) as bandwidth, count(*) as subtotal from $log-traffic where $filter and (logflag&1>0) and
(srcssid is not null or dstssid is not null) group by user_src, ap, srcintf, srcssid, srcmac,
hostname_mac /*SkipSTART*/order by bandwidth desc, subtotal desc/*SkipEND*/)### t where srcmac is
not null group by ap_srcintf, srcmac union all (select ap as ap_srcintf, stamac as srcmac from ###
(select $flex_timestamp as timestamp, stamac, stamac as srcmac, ap, ssid, ssid as srcssid, user_
src, sum(coalesce(sentdelta, 0)) as sentdelta, sum(coalesce(rcvddelta, 0)) as rcvddelta, sum
(coalesce(sentdelta, 0)+coalesce(rcvddelta, 0)) as bandwidth from (select itime, stamac, ap, ssid,
coalesce(`user`, ipstr(`srcip`)) as user_src, sentbyte-lag(coalesce(sentbyte, 0)) over (partition
by stamac order by itime) as sentdelta, rcvdbyte-lag(coalesce(rcvdbyte, 0)) over (partition by
stamac order by itime) as rcvddelta from $log-event where $filter and subtype='wireless' and
stamac is not null and ssid is not null and action in ('sta-wl-bridge-traffic-stats', 'reassoc-
req', 'assoc-req')) as t group by timestamp, stamac, ap, ssid, user_src /*SkipSTART*/order by
bandwidth desc/*SkipEND*/)### t where stamac is not null group by ap, stamac)) t group by srcintf
order by totalnum desc
```

Dataset Name	Description	Log Category
wifi-Top-SSID-By-Bandwidth	Top SSIDs by bandwidth usage	traffic

```
select
  srcssid,
  sum(bandwidth) as bandwidth
from
  (
    select
      srcssid,
      sum(bandwidth) as bandwidth
    from
      ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
ap, srcintf, srcssid, srcssid as ssid, srcmac, srcmac as stamac, coalesce(nullifna(`srcname`),
`srcmac`) as hostname_mac, max(srcswversion) as srcswversion, max(osname) as osname, max
(osversion) as osversion, max(devtype) as devtype, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte,
0)) as bandwidth, count(*) as subtotal from $log-traffic where $filter and (logflag&1>0) and
(srcssid is not null or dstssid is not null) group by user_src, ap, srcintf, srcssid, srcmac,
hostname_mac /*SkipSTART*/order by bandwidth desc, subtotal desc/*SkipEND*/)### t where srcssid is
not null group by srcssid having sum(bandwidth)>0 union all select ssid as srcssid, sum(bandwidth)
as bandwidth from ###(select $flex_timestamp as timestamp, stamac, stamac as srcmac, ap, ssid,
ssid as srcssid, user_src, sum(coalesce(sentdelta, 0)) as sentdelta, sum(coalesce(rcvddelta, 0))
as rcvddelta, sum(coalesce(sentdelta, 0)+coalesce(rcvddelta, 0)) as bandwidth from (select itime,
stamac, ap, ssid, coalesce(`user`, ipstr(`srcip`)) as user_src, sentbyte-lag(coalesce(sentbyte,
0)) over (partition by stamac order by itime) as sentdelta, rcvdbyte-lag(coalesce(rcvdbyte, 0))
over (partition by stamac order by itime) as rcvddelta from $log-event where $filter and
subtype='wireless' and stamac is not null and ssid is not null and action in ('sta-wl-bridge-
traffic-stats', 'reassoc-req', 'assoc-req')) as t group by timestamp, stamac, ap, ssid, user_src
/*SkipSTART*/order by bandwidth desc/*SkipEND*/)### t group by ssid having sum(bandwidth)>0) t
group by srcssid order by bandwidth desc
```

Dataset Name	Description	Log Category
wifi-Top-SSID-By-Client	Top SSIDs by client	traffic

```
select
  srcssid,
  count(distinct srcmac) as totalnum
from
  (
    select
      srcssid,
      srcmac
    from
      ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
ap, srcintf, srcssid, srcssid as ssid, srcmac, srcmac as stamac, coalesce(nullifna(`srcname`),
`srcmac`) as hostname_mac, max(srcswversion) as srcswversion, max(osname) as osname, max
(osversion) as osversion, max(devtype) as devtype, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte,
0)) as bandwidth, count(*) as subtotal from $log-traffic where $filter and (logflag&1>0) and
(srcssid is not null or dstssid is not null) group by user_src, ap, srcintf, srcssid, srcmac,
hostname_mac /*SkipSTART*/order by bandwidth desc, subtotal desc/*SkipEND*/)### t where srcmac is
not null group by srcssid, srcmac union all select ssid as srcssid, stamac as srcmac from ###
(select $flex_timestamp as timestamp, stamac, stamac as srcmac, ap, ssid, ssid as srcssid, user_
src, sum(coalesce(sentdelta, 0)) as sentdelta, sum(coalesce(rcvddelta, 0)) as rcvddelta, sum
```

```
(coalesce(sentdelta, 0)+coalesce(rcvddelta, 0)) as bandwidth from (select itime, stamac, ap, ssid, coalesce(`user`, ipstr(`srcip`)) as user_src, sentbyte-lag(coalesce(sentbyte, 0)) over (partition by stamac order by itime) as sentdelta, rcvdbyte-lag(coalesce(rcvdbyte, 0)) over (partition by stamac order by itime) as rcvddelta from $log-event where $filter and subtype='wireless' and stamac is not null and ssid is not null and action in ('sta-wl-bridge-traffic-stats', 'reassoc-req', 'assoc-req')) as t group by timestamp, stamac, ap, ssid, user_src /*SkipSTART*/order by bandwidth desc/*SkipEND*/)### t where stamac is not null group by ssid, stamac) t where srcssid is not null group by srcssid order by totalnum desc
```

Dataset Name	Description	Log Category
wifi-Top-App-By-Bandwidth	Top WiFi applications by bandwidth usage	traffic

```
select
  appid,
  app,
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  ) as bandwidth
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
  and (
    srcssid is not null
    or dstssid is not null
  )
  and nullifna(app) is not null
group by
  appid,
  app
having
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  )> 0
order by
  bandwidth desc
```

Dataset Name	Description	Log Category
wifi-Top-Client-By-Bandwidth	Top WiFi client by bandwidth usage	traffic

```
select
  client,
  sum(bandwidth) as bandwidth
from
```

```
  (
    select
      (
        coalesce(
          hostname_mac,
          & #039;unknown') || ' (' || get_devtype(srcswversion, osname, devtype) || ', ' ||
coalesce(osname, '') || (case when srcswversion is null then '' else ' ' || srcswversion end) ||
')') as client, sum(bandwidth) as bandwidth from ###(select coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, ap, srcintf, srcssid, srcssid as ssid, srcmac, srcmac
as stamac, coalesce(nullifna(`srcname`), `srcmac`) as hostname_mac, max(srcswversion) as
srcswversion, max(osname) as osname, max(osversion) as osversion, max(devtype) as devtype, sum
(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, count(*) as subtotal from $log-traffic
where $filter and (logflag&1>0) and (srcssid is not null or dstssid is not null) group by user_
src, ap, srcintf, srcssid, srcmac, hostname_mac /*SkipSTART*/order by bandwidth desc, subtotal
desc/*SkipEND*/)### t group by client having sum(bandwidth)>0 union all select (coalesce(stamac,
'unknown')) as client, sum(bandwidth) as bandwidth from ###(select $flex_timestamp as timestamp,
stamac, stamac as srcmac, ap, ssid, ssid as srcssid, user_src, sum(coalesce(sentdelta, 0)) as
sentdelta, sum(coalesce(rcvddelta, 0)) as rcvddelta, sum(coalesce(sentdelta, 0)+coalesce
(rcvddelta, 0)) as bandwidth from (select itime, stamac, ap, ssid, coalesce(`user`, ipstr
(`srcip`)) as user_src, sentbyte-lag(coalesce(sentbyte, 0)) over (partition by stamac order by
itime) as sentdelta, rcvdbyte-lag(coalesce(rcvdbyte, 0)) over (partition by stamac order by itime)
as rcvddelta from $log-event where $filter and subtype='wireless' and stamac is not null and ssid
is not null and action in ('sta-wl-bridge-traffic-stats', 'reassoc-req', 'assoc-req')) as t group
by timestamp, stamac, ap, ssid, user_src /*SkipSTART*/order by bandwidth desc/*SkipEND*/)### t
group by client having sum(bandwidth) > 0) t where client is not null group by client order by
bandwidth desc
```

Dataset Name	Description	Log Category
wifi-Top-OS-By-Bandwidth	Top WiFi os by bandwidth usage	traffic

```
select
  (
    coalesce(
      osname,
      & #039;unknown') || ' ' || coalesce(srcswversion, '')) as os, sum(bandwidth) as bandwidth
from ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
ap, srcintf, srcssid, srcssid as ssid, srcmac, srcmac as stamac, coalesce(nullifna(`srcname`),
`srcmac`) as hostname_mac, max(srcswversion) as srcswversion, max(osname) as osname, max
(osversion) as osversion, max(devtype) as devtype, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte,
0)) as bandwidth, count(*) as subtotal from $log-traffic where $filter and (logflag&1>0) and
(srcssid is not null or dstssid is not null) group by user_src, ap, srcintf, srcssid, srcmac,
hostname_mac /*SkipSTART*/order by bandwidth desc, subtotal desc/*SkipEND*/)### t group by os
having sum(bandwidth)>0 order by bandwidth desc
```

Dataset Name	Description	Log Category
wifi-Top-OS-By-WiFi-Client	Top WiFi os by WiFi client	traffic

```
select
  (
    coalesce(
      osname,
      & #039;unknown') || ' ' || coalesce(osversion, '')) as os, count(distinct srcmac) as
totalnum from ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as
user_src, ap, srcintf, srcssid, srcssid as ssid, srcmac, srcmac as stamac, coalesce(nullifna
(`srcname`), `srcmac`) as hostname_mac, max(srcswversion) as srcswversion, max(osname) as osname,
max(osversion) as osversion, max(devtype) as devtype, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte,
0)) as bandwidth, count(*) as subtotal from $log-traffic where $filter and (logflag&1>0) and
(srcssid is not null or dstssid is not null) group by user_src, ap, srcintf, srcssid, srcmac,
hostname_mac /*SkipSTART*/order by bandwidth desc, subtotal desc/*SkipEND*/)### t where srcmac is
not null group by os order by totalnum desc
```

Dataset Name	Description	Log Category
wifi-Top-Device-By-Bandwidth	Top WiFi device by bandwidth usage	traffic

```
select
  get_devtype(srcswversion, osname, devtype) as devtype_new,
  sum(bandwidth) as bandwidth
from
  ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, ap,
srcintf, srcssid, srcssid as ssid, srcmac, srcmac as stamac, coalesce(nullifna(`srcname`),
`srcmac`) as hostname_mac, max(srcswversion) as srcswversion, max(osname) as osname, max
(osversion) as osversion, max(devtype) as devtype, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte,
0)) as bandwidth, count(*) as subtotal from $log-traffic where $filter and (logflag&1>0) and
(srcssid is not null or dstssid is not null) group by user_src, ap, srcintf, srcssid, srcmac,
hostname_mac /*SkipSTART*/order by bandwidth desc, subtotal desc/*SkipEND*/)### t where devtype is
not null group by devtype_new having sum(bandwidth)>0 order by bandwidth desc
```

Dataset Name	Description	Log Category
wifi-Top-Device-By-Client	Top WiFi device by client	traffic

```
select
  devtype_new,
  count(distinct srcmac) as totalnum
from
  (
    select
      get_devtype(srcswversion, osname, devtype) as devtype_new,
      srcmac
    from
      ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
ap, srcintf, srcssid, srcssid as ssid, srcmac, srcmac as stamac, coalesce(nullifna(`srcname`),
`srcmac`) as hostname_mac, max(srcswversion) as srcswversion, max(osname) as osname, max
(osversion) as osversion, max(devtype) as devtype, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte,
0)) as bandwidth, count(*) as subtotal from $log-traffic where $filter and (logflag&1>0) and
```

```
(srcssid is not null or dstssid is not null) group by user_src, ap, srcintf, srcssid, srcmac,
hostname_mac /*SkipSTART*/order by bandwidth desc, subtotal desc/*SkipEND*/)### t where srcmac is
not null) t where devtype_new is not null group by devtype_new order by totalnum desc
```

Dataset Name	Description	Log Category
wifi-Overall-Traffic	WiFi overall traffic	traffic

```
select
  sum(bandwidth) as bandwidth
from
  (
    select
      sum(bandwidth) as bandwidth
    from
      ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
ap, srcintf, srcssid, srcssid as ssid, srcmac, srcmac as stamac, coalesce(nullifna(`srcname`),
`srcmac`) as hostname_mac, max(srcswversion) as srcswversion, max(osname) as osname, max
(osversion) as osversion, max(devtype) as devtype, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte,
0)) as bandwidth, count(*) as subtotal from $log-traffic where $filter and (logflag&1>0) and
(srcssid is not null or dstssid is not null) group by user_src, ap, srcintf, srcssid, srcmac,
hostname_mac /*SkipSTART*/order by bandwidth desc, subtotal desc/*SkipEND*/)### t group by srcssid
union all select sum(bandwidth) as bandwidth from ###(select $flex_timestamp as timestamp, stamac,
stamac as srcmac, ap, ssid, ssid as srcssid, user_src, sum(coalesce(sentdelta, 0)) as sentdelta,
sum(coalesce(rcvddelta, 0)) as rcvddelta, sum(coalesce(sentdelta, 0)+coalesce(rcvddelta, 0)) as
bandwidth from (select itime, stamac, ap, ssid, coalesce(`user`, ipstr(`srcip`)) as user_src,
sentbyte-lag(coalesce(sentbyte, 0)) over (partition by stamac order by itime) as sentdelta,
rcvdbyte-lag(coalesce(rcvdbyte, 0)) over (partition by stamac order by itime) as rcvddelta from
$log-event where $filter and subtype='wireless' and stamac is not null and ssid is not null and
action in ('sta-wl-bridge-traffic-stats', 'reassoc-req', 'assoc-req')) as t group by timestamp,
stamac, ap, ssid, user_src /*SkipSTART*/order by bandwidth desc/*SkipEND*/)### t) t
```

Dataset Name	Description	Log Category
wifi-Num-Distinct-Client	WiFi num distinct client	traffic

```
select
  count(distinct srcmac) as totalnum
from
  (
    select
      srcmac
    from
      ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
ap, srcintf, srcssid, srcssid as ssid, srcmac, srcmac as stamac, coalesce(nullifna(`srcname`),
`srcmac`) as hostname_mac, max(srcswversion) as srcswversion, max(osname) as osname, max
(osversion) as osversion, max(devtype) as devtype, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte,
0)) as bandwidth, count(*) as subtotal from $log-traffic where $filter and (logflag&1>0) and
(srcssid is not null or dstssid is not null) group by user_src, ap, srcintf, srcssid, srcmac,
```

```
hostname_mac /*SkipSTART*/order by bandwidth desc, subtotal desc/*SkipEND*/)### t where srcmac is
not null group by srcmac union all select stamac as srcmac from ###(select $flex_timestamp as
timestamp, stamac, stamac as srcmac, ap, ssid, ssid as srcssid, user_src, sum(coalesce(sentdelta,
0)) as sentdelta, sum(coalesce(rcvddelta, 0)) as rcvddelta, sum(coalesce(sentdelta, 0)+coalesce
(rcvddelta, 0)) as bandwidth from (select itime, stamac, ap, ssid, coalesce(`user`, ipstr
(`srcip`)) as user_src, sentbyte-lag(coalesce(sentbyte, 0)) over (partition by stamac order by
itime) as sentdelta, rcvdbyte-lag(coalesce(rcvdbyte, 0)) over (partition by stamac order by itime)
as rcvddelta from $log-event where $filter and subtype='wireless' and stamac is not null and ssid
is not null and action in ('sta-wl-bridge-traffic-stats', 'reassoc-req', 'assoc-req')) as t group
by timestamp, stamac, ap, ssid, user_src /*SkipSTART*/order by bandwidth desc/*SkipEND*/)### t
where stamac is not null group by stamac) t
```

Dataset Name	Description	Log Category
Top30-Subnets-by-Bandwidth-and-Sessions	Top subnets by application bandwidth	traffic

```
select
  ip_subnet(`srcip`) as subnet,
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  ) as bandwidth,
  sum(
    coalesce(rcvdbyte, 0)
  ) as traffic_in,
  sum(
    coalesce(sentbyte, 0)
  ) as traffic_out,
  count(*) as sessions
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
group by
  subnet
having
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  )> 0
order by
  bandwidth desc
```

Dataset Name	Description	Log Category
Top30-Subnets-by-Application-Bandwidth	Top applications by bandwidth	traffic

```
select
  ip_subnet(`srcip`) as subnet,
  app_group_name(app) as app_group,
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  ) as bandwidth
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
  and nullifna(app) is not null
group by
  subnet,
  app_group
having
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  )> 0
order by
  bandwidth desc
```

Dataset Name	Description	Log Category
Top30-Subnets-by-Application-Sessions	Top applications by sessions	traffic

```
select
  ip_subnet(`srcip`) as subnet,
  app_group_name(app) as app_group,
  count(*) as sessions
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
  and nullifna(app) is not null
group by
  subnet,
  app_group
order by
  sessions desc
```

Dataset Name	Description	Log Category
Top30-Subnets-by-Website-Bandwidth	Top websites and web category by bandwidth	traffic

```
select
  subnet,
  website,
  sum(bandwidth) as bandwidth
from
  ###(select ip_subnet(`srcip`) as subnet, hostname as website, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth from $log-traffic where $filter and hostname is not null and
(logflag&1>0) and (countweb>0 or ((logver is null or logver<502000000) and (hostname is not null
or utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter'))))
group by subnet, website order by bandwidth desc)### t group by subnet, website order by bandwidth
desc
```

Dataset Name	Description	Log Category
Top30-Subnets-by-Website-Hits	Top websites and web category by sessions	webfilter

```
select
  subnet,
  website,
  sum(hits) as hits
from
  ###(select ip_subnet(`srcip`) as subnet, hostname as website, count(*) as hits from $log where
$filter and hostname is not null group by subnet, website order by hits desc)### t group by
subnet, website order by hits desc
```

Dataset Name	Description	Log Category
Top30-Subnets-with-Top10-User-by-Bandwidth	Top users by bandwidth	traffic

```
select
  ip_subnet(`srcip`) as subnet,
  coalesce(
    nullifna(`user`),
    nullifna(`unauthuser`),
    ipstr(`srcip`)
  ) as user_src,
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  ) as bandwidth
from
  $log
where
```

```
  $filter
  and (
    logflag&1>0
  )
  and srcip is not null
group by
  subnet,
  user_src
having
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  )> 0
order by
  bandwidth desc
```

Dataset Name	Description	Log Category
Top30-Subnets-with-Top10-User-by-Sessions	Top users by sessions	traffic

```
select
  ip_subnet(`srcip`) as subnet,
  coalesce(
    nullifna(`user`),
    nullifna(`unauthuser`),
    ipstr(`srcip`)
  ) as user_src,
  count(*) as sessions
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
group by
  subnet,
  user_src
order by
  sessions desc
```

Dataset Name	Description	Log Category
app-Top-20-Category-and-Applications-by-Bandwidth	Top category and applications by bandwidth usage	traffic

```
select
  appcat,
  app,
```

```
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  ) as bandwidth
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
group by
  appcat,
  app
having
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  )> 0
order by
  bandwidth desc
```

Dataset Name	Description	Log Category
app-Top-20-Category-and-Applications-by-Session	Top category and applications by session	traffic

```
select
  appcat,
  app,
  count(*) as sessions
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
group by
  appcat,
  app
order by
  sessions desc
```

Dataset Name	Description	Log Category
app-Top-500-Allowed-Applications-by-Bandwidth	Top allowed applications by bandwidth usage	traffic

```
select
  from_itime($flex_timestamp) as timestamp,
```

```
  coalesce(
    nullifna(`user`),
    nullifna(`unauthuser`),
    ipstr(`srcip`)
  ) as user_src,
  appcat,
  app,
  coalesce(
    root_domain(hostname),
    ipstr(dstip)
  ) as destination,
  sum(
    coalesce(`sentbyte`, 0)+ coalesce(`rcvdbyte`, 0)
  ) as bandwidth
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
  and action in (
    & #039;accept', 'close', 'timeout') group by timestamp, user_src, appcat, app, destination
order by bandwidth desc
```

Dataset Name	Description	Log Category
app-Top-500-Blocked-Applications-by-Session	Top blocked applications by session	traffic

```
select
  coalesce(
    nullifna(`user`),
    nullifna(`unauthuser`),
    ipstr(`srcip`)
  ) as user_src,
  appcat,
  app,
  count(*) as sessions
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
  and action in (
    & #039;deny', 'blocked', 'reset', 'dropped') group by user_src, appcat, app order by sessions
desc
```

Dataset Name	Description	Log Category
web-Detailed-Website-Browsing-Log	Web detailed website browsing log	traffic

```
select
  from_dtime(dtime) as timestamp,
  catdesc,
  hostname as website,
  status,
  sum(bandwidth) as bandwidth
from
  ###(select dtime, catdesc, hostname, cast(utmaction as text) as status, coalesce(sentbyte,
0)+coalesce(rcvdbyte, 0) as bandwidth from $log-traffic where $filter and hostname is not null and
(logflag&1>0) and (countweb>0 or ((logver is null or logver<502000000) and (hostname is not null
or utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter'))))
order by dtime desc)### t group by timestamp, catdesc, website, status order by timestamp desc
```

Dataset Name	Description	Log Category
web-Hourly-Category-and-Website-Hits-Action	Web hourly category and website hits action	webfilter

```
select
  hod,
  website,
  sum(hits) as hits
from
  ###(select $hour_of_day as hod, (hostname || ' (' || coalesce(`catdesc`, 'Unknown') || ')') as
website , count(*) as hits from $log where $filter and hostname is not null group by hod, website
order by hod, hits desc)### t group by hod, website order by hod, hits desc
```

Dataset Name	Description	Log Category
web-Top-20-Category-and-Websites-by-Bandwidth	Web top category and websites by bandwidth usage	traffic

```
select
  website,
  catdesc,
  sum(bandwidth) as bandwidth
from
  ###(select hostname as website, catdesc, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as
bandwidth from $log-traffic where $filter and hostname is not null and (logflag&1>0) and
(countweb>0 or ((logver is null or logver<502000000) and (hostname is not null or utmevent in
('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter')))) group by website,
catdesc order by bandwidth desc)### t group by website, catdesc order by bandwidth desc
```

Dataset Name	Description	Log Category
web-Top-20-Category-and-Websites-by-Session	Web top category and websites by session	webfilter

```
select
  website,
  catdesc,
  sum(sessions) as hits
from
  ###(select hostname as website, catdesc, count(*) as sessions from $log where $filter and
hostname is not null group by hostname, catdesc order by sessions desc)### t group by website,
catdesc order by hits desc
```

Dataset Name	Description	Log Category
web-Top-500-Website-Sessions-by-Bandwidth	Web top website sessions by bandwidth usage	traffic

```
select
  from_dtime(timestamp) as timestamp,
  user_src,
  website,
  catdesc,
  cast(
    sum(dura)/ 60 as decimal(18, 2)
  ) as dura,
  sum(bandwidth) as bandwidth
from
  ###(select $flex_timestamp(dtime) as timestamp, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, hostname as website, catdesc, sum(coalesce(duration,
0)) as dura, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from $log where $filter
and hostname is not null and (logflag&1>0) and action in ('accept','close','timeout') group by
timestamp, user_src, website, catdesc having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0
order by bandwidth desc)### t group by timestamp, user_src, website, catdesc order by bandwidth
desc
```

Dataset Name	Description	Log Category
web-Top-500-User-Visted-Websites-by-Bandwidth	Web top user visted websites by bandwidth usage	traffic

```
select
  website,
  catdesc,
  sum(bandwidth) as bandwidth
from
  ###(select hostname as website, catdesc, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as
```

```
bandwidth from $log-traffic where $filter and hostname is not null and (logflag&1>0) and
(countweb>0 or ((logver is null or logver<502000000) and (hostname is not null or utmevent in
('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter')))) group by
hostname, catdesc having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth
desc)### t group by website, catdesc order by bandwidth desc
```

Dataset Name	Description	Log Category
web-Top-500-User-Visted-Websites-by-Session	Web top user visted websites by session	webfilter

```
select
  website,
  catdesc,
  sum(sessions) as sessions
from
  ###(select hostname as website, catdesc, count(*) as sessions from $log where $filter and
hostname is not null group by hostname, catdesc order by sessions desc)### t where catdesc is not
null group by website, catdesc order by sessions desc
```

Dataset Name	Description	Log Category
fct-Installed-Feature-Summary	Installed Feature Summary	fct-event

```
select
  subtype,
  count(distinct fctuid) as totalnum
from
  ###(select uid as fctuid, regexp_replace(os, '\\(build.*', '') as os_short, fctver, subtype,
fgtserial, max(case when msg like 'Compliance rules%applied' then 1 else 0 end) as compliance_flag
from $log where $filter and subtype != 'admin' group by uid, os_short, fctver, subtype, fgtserial
order by compliance_flag desc)### t where subtype is not null group by subtype order by totalnum
desc
```

Dataset Name	Description	Log Category
fct-Device-by-Operating-System	Device by OS	fct-event

```
select
  os_short as os,
  count(distinct fctuid) as totalnum
from
  ###(select uid as fctuid, regexp_replace(os, '\\(build.*', '') as os_short, fctver, subtype,
fgtserial, max(case when msg like 'Compliance rules%applied' then 1 else 0 end) as compliance_flag
from $log where $filter and subtype != 'admin' group by uid, os_short, fctver, subtype, fgtserial
order by compliance_flag desc)### t where os_short is not null group by os order by totalnum desc
```

Dataset Name	Description	Log Category
fct-Installed-FortiClient-Version	FortiClient Version	fct-event

```
select
  fctver as fctver_short,
  count(distinct fctuid) as totalnum
from
  ###(select uid as fctuid, regexp_replace(os, '\\(build.*', '') as os_short, fctver, subtype,
fgtserial, max(case when msg like 'Compliance rules%applied' then 1 else 0 end) as compliance_flag
from $log where $filter and subtype != 'admin' group by uid, os_short, fctver, subtype, fgtserial
order by compliance_flag desc)### t where fctver is not null group by fctver order by totalnum
desc
```

Dataset Name	Description	Log Category
fct-Endpoint-Profile-Deployment	Endpoint Profile Deployment	fct-event

```
select
  profile,
  count(distinct fctuid) as totalnum
from
  ###(select uid as fctuid, coalesce(nullifna(usingpolicy), 'No Profile') as profile, count(*) as
total_num from $log where $filter group by uid, profile order by total_num desc)### t group by
profile order by totalnum desc
```

Dataset Name	Description	Log Category
fct-Client-Summary	Client Summary	fct-event

```
select
  hostname,
  deviceip,
  os_short as os,
  profile,
  fctver,
  from_itime(
    max(itime)
  ) as last_seen
from
  ###(select hostname, deviceip, regexp_replace(os, '\\(build.*', '') as os_short, nullifna
(usingpolicy) as profile, fctver, max(itime) as itime from $log where $filter and os is not null
group by hostname, deviceip, os_short, profile, fctver order by itime desc)### t group by
hostname, deviceip, os, profile, fctver order by last_seen desc
```

Dataset Name	Description	Log Category
fct-Total-Threats-Found	Total Threats Found	fct-traffic

```
select
  utmevent_s as utmevent,
  count(distinct threat) as totalnum
from
  ###(select coalesce(nullifna(lower(utmevent)), 'unknown') as utmevent_s, threat, count(*) as
total_num from $log where $filter and threat is not null and utmaction='blocked' group by
utmevent_s, threat order by total_num desc)### t group by utmevent order by totalnum desc
```

Dataset Name	Description	Log Category
fct-Top10-AV-Threats-Detected	Top AV Threats Detected	fct-traffic

```
select
  threat,
  sum(totalnum) as totalnum
from
  (
    (
      select
        threat,
        sum(totalnum) as totalnum
      from
        ###(select threat, count(*) as totalnum from $log-fct-traffic where $filter and threat is
not null and lower(utmevent)='antivirus' group by threat order by totalnum desc)### t group by
threat) union all (select threat, sum(totalnum) as totalnum from ###(select virus as threat, count
(*) as totalnum from $log-fct-event where $filter and virus is not null group by threat order by
totalnum desc)### t group by threat)) t group by threat order by totalnum desc
```

Dataset Name	Description	Log Category
fct-Top10-Infected-Devices-with-Botnet	Top Infected Devices with Botnet	fct-traffic

```
select
  hostname,
  count(*) as totalnum
from
  $log
where
  $filter
  and hostname is not null
  and lower(utmevent) in (
    & #039;webfilter', 'appfirewall') and lower(threat) like '%botnet%' group by hostname order by
totalnum desc
```

Dataset Name	Description	Log Category
fct-Top10-Infected-Devices-with-Virus-Malware	Top Infected Devices with Virus Malware	fct-traffic

```
select
  hostname,
  sum(totalnum) as totalnum
from
  (
    (
      select
        hostname,
        sum(totalnum) as totalnum
      from
        ###(select hostname, count(*) as totalnum from $log-fct-traffic where $filter and hostname
is not null and lower(utmevent) in ('antivirus', 'antimalware') group by hostname order by
totalnum desc)### t group by hostname) union all (select hostname, sum(totalnum) as totalnum from
###(select hostname, count(*) as totalnum from $log-fct-event where $filter and hostname is not
null and virus is not null group by hostname order by totalnum desc)### t group by hostname)) t
group by hostname order by totalnum desc
```

Dataset Name	Description	Log Category
fct-All-Antivirus-Antimalware-Detections	All Antivirus and Antimalware Detections	fct-traffic

```
select
  threat,
  hostname,
  hostuser,
  utmaction,
  from_dtime(
    max(dtime)
  ) as last_seen
from
  (
    (
      select
        threat,
        hostname,
        hostuser,
        utmaction,
        max(dtime) as dtime
      from
        ###(select threat, hostname, coalesce(nullifna(`user`), 'Unknown') as hostuser, utmaction,
max(dtime) as dtime from $log-fct-traffic where $filter and lower(utmevent) in ('antivirus',
'antimalware') group by threat, hostname, hostuser, utmaction order by threat)### t group by
threat, hostname, hostuser, utmaction) union all (select threat, hostname, hostuser, utmaction,
max(dtime) as dtime from ###(select virus as threat, hostname, coalesce(nullifna(`user`),
'Unknown') as hostuser, action as utmaction, max(dtime) as dtime from $log-fct-event where $filter
and (logflag is null or logflag&8=0) and virus is not null group by threat, hostname, hostuser,
utmaction order by threat)### t group by threat, hostname, hostuser, utmaction)) t group by
threat, hostname, hostuser, utmaction order by threat
```

Dataset Name	Description	Log Category
fct-Web-Filter-Violations	Web Filter Violations	fct-traffic

```
select
  hostuser,
  hostname,
  string_agg(
    distinct remotename,
    & #039;,') as remotename, utmaction, sum(total) as totalnum, from_dtime(max(dtime)) as last_
seen from ###(select remotename, hostname, coalesce(nullifna(`user`), 'Unknown') as hostuser,
utmaction, count(*) as total, max(dtime) as dtime from $log where $filter and lower
(utmevent)='webfilter' and utmaction='blocked' group by remotename, hostname, hostuser, utmaction
order by total desc)### t group by hostuser, hostname, utmaction order by totalnum desc
```

Dataset Name	Description	Log Category
fct-Application-Firewall	Application Firewall	fct-traffic

```
select
  threat,
  hostname,
  hostuser,
  utmaction,
  from_dtime(
    max(dtime)
  ) as last_seen
from
  ###(select threat, hostname, coalesce(nullifna(`user`), 'Unknown') as hostuser, utmaction, max
(dtime) as dtime from $log where $filter and lower(utmevent)='appfirewall' and utmaction='blocked'
group by threat, hostname, hostuser, utmaction order by dtime desc)### t1 left join app_mdata t2
on t1.threat=t2.name group by threat, risk, hostname, hostuser, utmaction order by risk desc
```

Dataset Name	Description	Log Category
fct-Errors-and-Alerts	Errors and Alerts	fct-event

```
select
  msg,
  hostname,
  hostuser,
  from_dtime(
    max(dtime)
  ) as last_seen
from
  ###(select msg, hostname, coalesce(nullifna(`user`), 'Unknown') as hostuser, max(dtime) as dtime
from $log where $filter and level in ('error', 'alert') group by msg, hostname, hostuser order by
dtime desc)### t group by msg, hostname, hostuser order by last_seen desc
```

Dataset Name	Description	Log Category
fct-Threats-by-Top-Devices	Threats by Top Devices	fct-traffic

```
select
  hostname,
  count(*) as totalnum
from
  $log
where
  $filter
  and hostname is not null
  and utmevent is not null
  and utmaction =& #039;blocked' group by hostname order by totalnum desc
```

Dataset Name	Description	Log Category
fct-vuln-Device-Vulnerabilities	Vulnerabilities Detected by User/Device	fct-netscan

```
select
  vulnseverity,
  (
    CASE vulnseverity WHEN & #039;Critical' THEN 5 WHEN 'High' THEN 4 WHEN 'Medium' THEN 3 WHEN
'Info' THEN 2 WHEN 'Low' THEN 1 ELSE 0 END) as severity_number, count(distinct vulnname) as vuln_
num from ###(select vulnseverity, devid, vulnname, count(*) as total_num from $log where $filter
and nullifna(vulnseverity) is not null and nullifna(vulnname) is not null group by vulnseverity,
vulnname, devid order by total_num desc)### t group by vulnseverity order by severity_number desc
```

Dataset Name	Description	Log Category
fct-vuln-Category-Type-Vulnerabilities	Vulnerabilities Detected by Category Type	fct-netscan

```
select
  vulncat,
  count(distinct vulnname) as totalnum
from
  ###(select vulncat, vulnname, count(*) as total_num from $log where $filter and nullifna
(vulncat) is not null and nullifna(vulnname) is not null group by vulncat, vulnname order by
total_num desc)### t group by vulncat order by totalnum desc
```

Dataset Name	Description	Log Category
fct-vuln-Vulnerabilities-by-OS	Forticlient Vulnerabilities by OS	fct-netscan

```
select
  os,
```

```
  count(distinct vulnname) as totalnum
from
  ###(select os, vulnname, count(*) as total_num from $log where $filter and nullifna(os) is not
null and nullifna(vulnname) is not null group by os, vulnname order by total_num desc)### t group
by os order by totalnum desc
```

Dataset Name	Description	Log Category
fct-vuln-Vulnerabilities-by-Risk-Level	Number Vulnerability by Device and Risk Level	fct-netscan

```
select
  vulnseverity,
  (
    case when vulnseverity =& #039;Critical' then 5 when vulnseverity='High' then 4 when
vulnseverity='Medium' then 3 when vulnseverity='Low' then 2 when vulnseverity='Info' then 1 else 0
end) as severity_number, count(distinct vulnname) as vuln_num, count(distinct devid) as dev_num
from ###(select vulnseverity, devid, vulnname, count(*) as total_num from $log where $filter and
nullifna(vulnseverity) is not null and nullifna(vulnname) is not null group by vulnseverity,
vulnname, devid order by total_num desc)### t where nullifna(devid) is not null group by
vulnseverity order by dev_num desc, severity_number desc
```

Dataset Name	Description	Log Category
fct-vuln-Device-by-Risk-Level	Number Vulnerability by Device and Risk Level	fct-netscan

```
select
  vulnseverity,
  (
    case when vulnseverity =& #039;Critical' then 5 when vulnseverity='High' then 4 when
vulnseverity='Medium' then 3 when vulnseverity='Low' then 2 when vulnseverity='Info' then 1 else 0
end) as severity_number, count(distinct vulnname) as vuln_num, count(distinct devid) as dev_num
from ###(select vulnseverity, devid, vulnname, count(*) as total_num from $log where $filter and
nullifna(vulnseverity) is not null and nullifna(vulnname) is not null group by vulnseverity,
vulnname, devid order by total_num desc)### t where nullifna(devid) is not null group by
vulnseverity order by dev_num desc, severity_number desc
```

Dataset Name	Description	Log Category
fct-vuln-Vulnerability-Trend	Vulnerability Trend	fct-netscan

```
select
  $flex_timescale(timestamp) as hodex,
  count(distinct vulnname) as total_num
from
  ###(select $flex_timestamp as timestamp, vulnname from $log where $filter and nullifna(vulnname)
is not null group by timestamp, vulnname order by timestamp desc)### t group by hodex order by
hodex
```

Dataset Name	Description	Log Category
fct-vuln-Details-by-Risk-Level-Device	Vulnerability Details for Each Risk Level by Device	fct-netscan

```
select
  hostname,
  os,
  vulnseverity,
  count(distinct vulnname) as vuln_num,
  count(distinct products) as products,
  count(distinct cve_id) as cve_count
from
  ###(select hostname, os, vulnname, vulnseverity, vulnid, count(*) as total_num from $log where
$filter and vulnname is not null and vulnseverity is not null and hostname is not null group by
hostname, os, vulnname, vulnseverity, vulnid order by total_num desc)### t1 left join fct_mdata t2
on t1.vulnid=t2.vid::int group by hostname, os, vulnseverity order by vuln_num desc, hostname
```

Dataset Name	Description	Log Category
fct-vuln-Details-by-Device-User	Vulnerability Details by Device User	fct-netscan

```
select
  hostname,
  (
    & #039;<div>' || vulnname || '</div>') as vulnname, vulnseverity, vulncat, string_agg(distinct
products, ',') as products, string_agg(distinct cve_id, ',') as cve_list, ('<a href=' || String_
agg(DISTINCT vendor_link, ',') || '>Remediation Info</a>') as vendor_link from ###(select
hostname, vulnname, vulnseverity, vulncat, vulnid, count(*) as total_num from $log where $filter
and vulnname is not null and hostname is not null group by hostname, vulnname, vulnseverity,
vulncat, vulnid order by total_num desc)### t1 inner join fct_mdata t2 on t1.vulnid=t2.vid::int
group by hostname, vulnname, vulnseverity, vulncat order by hostname
```

Dataset Name	Description	Log Category
fct-vuln-Remediation-by-Device	Remediate The Vulnerability Found on Device	fct-netscan

```
select
  hostname,
  (
    & #039;<div>' || vulnname || '</div>') as vulnname, vulnseverity, string_agg(distinct vendor_
link, ',') as vendor_link from ###(select hostname, vulnname, vulnseverity, vulnid, count(*) as
total_num from $log where $filter and vulnname is not null and hostname is not null group by
hostname, vulnname, vulnseverity, vulnid order by total_num desc)### t1 inner join fct_mdata t2 on
t1.vulnid=t2.vid::int group by hostname, vulnname, vulnseverity order by vulnseverity, hostname
```

Dataset Name	Description	Log Category
fct-vuln-Remediation-by-Vulnerability	Remediation by Vulnerability	fct-netscan

```
select
  (
    & #039;<b>' || vulnname || '</b><br/><br/>' || 'Description<br/><div style=word-break:normal>'
|| description || '</div><br/><br/>' || 'Affected Products<br/>' || products || '<br/><br/>' ||
'Impact<br/>' || impact || '<br/><br/>' || 'Recommended Actions<br/>' || vendor_link ||
'<br/><br/><br/>') as remediation from ###(select devid, vulnname, vulnseverity, (case
vulnseverity when 'low' then 1 when 'info' then 2 when 'medium' then 3 when 'high' then 4 when
'critical' then 5 else 0 end) as severity_level, vulnid from $log where $filter and vulnname is
not null group by devid, vulnname, vulnseverity, severity_level, vulnid order by severity_
level)### t1 inner join fct_mdata t2 on t1.vulnid=t2.vid::int group by remediation order by
remediation
```

Dataset Name	Description	Log Category
fct-vuln-Top-30-Targeted-High-Risk-Vulnerabilities	Top 30 Targeted High Risk Vulnerabilities	fct-netscan

```
select
  t3.cve_id,
  score,
  string_agg(
    distinct products,
    & #039;,') as products, string_agg(distinct ('<a href=' || vendor_link || '>Mitigation
Information</a>'), '<br>') as vendor_link from ###(select vulnid from $log where $filter group by
vulnid order by vulnid)### t1 inner join fct_mdata t2 on t2.vid=t1.vulnid::text inner join fct_
cve_score t3 on strpos(t2.cve_id, t3.cve_id) > 0 group by t3.cve_id, score order by score desc,
t3.cve_id
```

Dataset Name	Description	Log Category
fct-Endpoints-by-FortiGate	Endpoints by FortiGate	fct-event

```
select
  fgtserial,
  count(distinct fctuid) as totalnum
from
  ###(select uid as fctuid, regexp_replace(os, '\\(build.*', '') as os_short, fctver, subtype,
fgtserial, max(case when msg like 'Compliance rules%applied' then 1 else 0 end) as compliance_flag
from $log where $filter and subtype != 'admin' group by uid, os_short, fctver, subtype, fgtserial
order by compliance_flag desc)### t where fgtserial is not null group by fgtserial order by
totalnum desc
```

Dataset Name	Description	Log Category
fct-Top-Malware-Detections	Top Infected Devices with Malware	fct-traffic

```
select
  hostname,
  fctuid,
  sum(totalnum) as totalnum
from
  (
    (
      select
        hostname,
        fctuid,
        sum(totalnum) as totalnum
      from
        ###(select threat, hostname, coalesce(nullifna(`user`), 'Unknown') as hostuser, utmaction,
max(dtime) as dtime, uid as fctuid, count(*) as totalnum from $log-fct-traffic where $filter and
lower(utmevent) in ('antivirus', 'antimalware') group by threat, hostname, hostuser, utmaction,
uid order by threat)### t group by hostname, fctuid) union all (select hostname, fctuid, sum
(totalnum) as totalnum from ###(select virus as threat, hostname, coalesce(nullifna(`user`),
'Unknown') as hostuser, action as utmaction, max(dtime) as dtime, uid as fctuid, count(*) as
totalnum from $log-fct-event where $filter and (logflag is null or logflag&8=0) and virus is not
null group by threat, hostname, hostuser, utmaction, uid order by threat)### t group by hostname,
fctuid)) t group by hostname, fctuid order by totalnum desc
```

Dataset Name	Description	Log Category
fct-Top10-Malware-Detections	Top 10 Infected Devices with Malware	fct-traffic

```
select
  threat,
  hostname,
  hostuser,
  utmaction,
  fctuid,
  sum(totalnum) as totalnum
from
  (
    (
      select
        threat,
        hostname,
        hostuser,
        utmaction,
        fctuid,
        sum(totalnum) as totalnum
      from
        ###(select threat, hostname, coalesce(nullifna(`user`), 'Unknown') as hostuser, utmaction,
max(dtime) as dtime, uid as fctuid, count(*) as totalnum from $log-fct-traffic where $filter and
```

```
lower(utmevent) in ('antivirus', 'antimalware') group by threat, hostname, hostuser, utmaction,
uid order by threat)### t group by threat, hostname, hostuser, utmaction, fctuid) union all
(select threat, hostname, hostuser, utmaction, fctuid, sum(totalnum) as totalnum from ###(select
virus as threat, hostname, coalesce(nullifna(`user`), 'Unknown') as hostuser, action as utmaction,
max(dtime) as dtime, uid as fctuid, count(*) as totalnum from $log-fct-event where $filter and
(logflag is null or logflag&8=0) and virus is not null group by threat, hostname, hostuser,
utmaction, uid order by threat)### t group by threat, hostname, hostuser, utmaction, fctuid)) t
where utmaction != 'pass' group by threat, hostname, hostuser, utmaction, fctuid order by totalnum
desc
```

Dataset Name	Description	Log Category
fct-Devices-with-Botnet	Infected Devices with Botnet	fct-traffic

```
select
  threat,
  hostname,
  coalesce(
    nullifna(`user`),
    & #039;Unknown') as hostuser, utmaction, uid as fctuid, count(*) as totalnum from $log where
$filter and hostname is not null and lower(utmevent) in ('webfilter', 'appfirewall') and lower
(threat) like '%botnet%' group by threat, hostname, hostuser, utmaction, fctuid order by totalnum
desc
```

Dataset Name	Description	Log Category
fct-vuln-Vulnerability-by-Hostname	Vulnerability Details for Each Risk Level by Device	fct-netscan

```
select
  hostname,
  os,
  vulnseverity,
  count(distinct vulnname) as vuln_num,
  count(distinct products) as products,
  count(distinct cve_id) as cve_count
from
  ###(select hostname, os, vulnname, vulnseverity, vulnid, count(*) as total_num from $log where
$filter and vulnname is not null and vulnseverity is not null and hostname is not null group by
hostname, os, vulnname, vulnseverity, vulnid order by total_num desc)### t1 left join fct_mdata t2
on t1.vulnid=t2.vid::int group by hostname, os, vulnseverity order by vuln_num desc, hostname
```

Dataset Name	Description	Log Category
fct-Users-With-Web-Violations	Web Filter Violations	fct-traffic

```
select
  hostuser,
  hostname,
```

```
  string_agg(
    distinct remotename,
    & #039;,') as remotename, utmaction, sum(total) as totalnum, from_dtime(max(dtime)) as last_
seen from ###(select remotename, hostname, coalesce(nullifna(`user`), 'Unknown') as hostuser,
utmaction, count(*) as total, max(dtime) as dtime from $log where $filter and lower
(utmevent)='webfilter' and utmaction='blocked' group by remotename, hostname, hostuser, utmaction
order by total desc)### t group by hostuser, hostname, utmaction order by totalnum desc
```

Dataset Name	Description	Log Category
fct-Compliance-by-FortiGate	FortiClinet Compliance by FortiGate Enforcing	fct-event

```
select
  fgtserial,
  count(distinct fctuid) as totalnum
from
  (
    select
      fgtserial,
      fctuid,
      max(compliance_flag) as compliance_flag
    from
      ###(select uid as fctuid, regexp_replace(os, '\\(build.*', '') as os_short, fctver, subtype,
fgtserial, max(case when msg like 'Compliance rules%applied' then 1 else 0 end) as compliance_flag
from $log where $filter and subtype != 'admin' group by uid, os_short, fctver, subtype, fgtserial
order by compliance_flag desc)### tt group by fgtserial, fctuid) t where compliance_flag = 1 group
by fgtserial order by totalnum desc
```

Dataset Name	Description	Log Category
fct-Compliance-Status	Number of FortiClinets by Compliance Status	fct-event

```
select
  (
    case compliance_flag when 1 then & #039;Compliant' else 'Non-Compliant' end) as compliance,
count(distinct fctuid) as totalnum from (select fctuid, max(compliance_flag) as compliance_flag
from ###(select uid as fctuid, regexp_replace(os, '\\(build.*', '') as os_short, fctver, subtype,
fgtserial, max(case when msg like 'Compliance rules%applied' then 1 else 0 end) as compliance_flag
from $log where $filter and subtype != 'admin' group by uid, os_short, fctver, subtype, fgtserial
order by compliance_flag desc)### tt group by fctuid) t group by compliance order by totalnum desc
```

Dataset Name	Description	Log Category
fct-Non-Compliant-Endpoints	Non-compliant Endpoints	fct-event

```
select
  t1.fgtserial,
  t3.srcintf,
```

```
  t2.epname as hostname,
  t2.mac,
  & #039;Non-Compliant' as status from (select fgtserial, fctuid, max(compliance_flag) as
compliance_flag from ###(select uid as fctuid, regexp_replace(os, '\\(build.*', '') as os_short,
fctver, subtype, fgtserial, max(case when msg like 'Compliance rules%applied' then 1 else 0 end)
as compliance_flag from $log where $filter and subtype != 'admin' group by uid, os_short, fctver,
subtype, fgtserial order by compliance_flag desc)### tt group by fgtserial, fctuid) t1 left join
$ADOM_ENDPOINT t2 on t1.fctuid = t2.fctuid left join $ADOM_EPEU_DEVMAP t3 on t2.epid = t3.epid
where compliance_flag = 0 group by t1.fctuid, t1.fgtserial, t3.srcintf, t2.epname, t2.mac
```

Dataset Name	Description	Log Category
fct-Traffic-Web-Hits	Web Traffic Trend	fct-traffic

```
select
  $flex_timescale(timestamp) as hodex,
  sum(requests) as requests
from
  ###(select $flex_timestamp as timestamp, count(*) as requests from $log where $filter and lower
(utmevent)='webfilter' group by timestamp order by timestamp desc)### t group by hodex order by
hodex
```

Dataset Name	Description	Log Category
fct-Traffic-Top-Allowed-Web-Cat	Top Visited Web Categories	fct-traffic

```
select
  category,
  sum(requests) as requests
from
  ###(select fct_webcat(threat) as category, remotename as website, direction, utmaction, count(*)
as requests from $log where $filter and threat is not null and lower(utmevent)='webfilter' group
by category, website, direction, utmaction order by requests desc)### t where direction='outbound'
and utmaction='passthrough' group by category order by requests desc
```

Dataset Name	Description	Log Category
fct-Traffic-Top-Allowed-Website	Top Visited Websites	fct-traffic

```
select
  website,
  string_agg(
    distinct category,
    & #039;, ') as agg_category, sum(requests) as requests from ###(select fct_webcat(threat) as
category, remotename as website, direction, utmaction, count(*) as requests from $log where
$filter and threat is not null and lower(utmevent)='webfilter' group by category, website,
direction, utmaction order by requests desc)### t where direction='outbound' and
utmaction='passthrough' and website is not null group by website order by requests desc
```

Dataset Name	Description	Log Category
fct-Traffic-Top-Category-By-Website-Session	Top Web Categories by Website Session	fct-traffic

```
select
  category,
  website,
  sum(requests) as requests
from
  ###(select fct_webcat(threat) as category, remotename as website, direction, utmaction, count(*)
as requests from $log where $filter and threat is not null and lower(utmevent)='webfilter' group
by category, website, direction, utmaction order by requests desc)### t where nullifna(category)
is not null group by category, website order by requests desc
```

Dataset Name	Description	Log Category
fct-Traffic-Top-Web-Users-By-Website	Top Web Users by Website	fct-traffic

```
select
  coalesce(
    nullifna(`user`),
    ipstr(`srcip`)
  ) as user_src,
  remotename as website,
  count(*) as requests
from
  $log
where
  $filter
  and direction =& #039;outbound' and remotename is not null and utmaction='passthrough' and lower
(utmevent)='webfilter' group by user_src, website order by requests desc
```

Dataset Name	Description	Log Category
os-Detect-OS-Count	Detected operation system count	traffic

```
select
  (
    coalesce(
      osname,
      & #039;Unknown')) as os, count(*) as totalnum from $log where $filter and (logflag&1>0)
group by os order by totalnum desc
```

Dataset Name	Description	Log Category
drilldown-Top-App-By-Sessions-Table	Drilldown top applications by session count	traffic

```
select
  appid,
  app,
  sum(sessions) as sessions
from
  ###(select appid, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as
user_src, dstip, srcintf, dstintf, policyid, count(*) as sessions, sum(coalesce(sentbyte,
0)+coalesce(rcvdbyte, 0)) as bandwidth from $log where $filter-exclude-var and (logflag&1>0) group
by appid, app, user_src, dstip, srcintf, dstintf, policyid order by sessions desc)### t where
$filter-drilldown and nullifna(app) is not null group by appid, app order by sessions desc
```

Dataset Name	Description	Log Category
drilldown-Top-App-By-Sessions-Bar	Drilldown top applications by session count	traffic

```
select
  appid,
  app,
  sum(sessions) as sessions
from
  ###(select appid, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as
user_src, dstip, srcintf, dstintf, policyid, count(*) as sessions, sum(coalesce(sentbyte,
0)+coalesce(rcvdbyte, 0)) as bandwidth from $log where $filter-exclude-var and (logflag&1>0) group
by appid, app, user_src, dstip, srcintf, dstintf, policyid order by sessions desc)### t where
$filter-drilldown and nullifna(app) is not null group by appid, app order by sessions desc
```

Dataset Name	Description	Log Category
drilldown-Top-App-By-Bandwidth-Table	Drilldown top applications by bandwidth usage	traffic

```
select
  appid,
  app,
  sum(bandwidth) as bandwidth
from
  ###(select appid, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as
user_src, dstip, srcintf, dstintf, policyid, count(*) as sessions, sum(coalesce(sentbyte,
0)+coalesce(rcvdbyte, 0)) as bandwidth from $log where $filter-exclude-var and (logflag&1>0) group
by appid, app, user_src, dstip, srcintf, dstintf, policyid order by sessions desc)### t where
$filter-drilldown and nullifna(app) is not null group by appid, app having sum(bandwidth)>0 order
by bandwidth desc
```

Dataset Name	Description	Log Category
drilldown-Top-App-By-Bandwidth-Bar	Drilldown top applications by bandwidth usage	traffic

```
select
  appid,
  app,
  sum(bandwidth) as bandwidth
from
  ###(select appid, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as
user_src, dstip, srcintf, dstintf, policyid, count(*) as sessions, sum(coalesce(sentbyte,
0)+coalesce(rcvdbyte, 0)) as bandwidth from $log where $filter-exclude-var and (logflag&1>0) group
by appid, app, user_src, dstip, srcintf, dstintf, policyid order by sessions desc)### t where
$filter-drilldown and nullifna(app) is not null group by appid, app having sum(bandwidth)>0 order
by bandwidth desc
```

Dataset Name	Description	Log Category
drilldown-Top-Destination-By-Sessions-Table	Drilldown top destination by session count	traffic

```
select
  dstip,
  sum(sessions) as sessions
from
  ###(select appid, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as
user_src, dstip, srcintf, dstintf, policyid, count(*) as sessions, sum(coalesce(sentbyte,
0)+coalesce(rcvdbyte, 0)) as bandwidth from $log where $filter-exclude-var and (logflag&1>0) group
by appid, app, user_src, dstip, srcintf, dstintf, policyid order by sessions desc)### t where
$filter-drilldown and dstip is not null group by dstip order by sessions desc
```

Dataset Name	Description	Log Category
drilldown-Top-Destination-By-Bandwidth-Table	Drilldown top destination by bandwidth usage	traffic

```
select
  dstip,
  sum(bandwidth) as bandwidth
from
  ###(select appid, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as
user_src, dstip, srcintf, dstintf, policyid, count(*) as sessions, sum(coalesce(sentbyte,
0)+coalesce(rcvdbyte, 0)) as bandwidth from $log where $filter-exclude-var and (logflag&1>0) group
by appid, app, user_src, dstip, srcintf, dstintf, policyid order by sessions desc)### t where
$filter-drilldown and dstip is not null group by dstip having sum(bandwidth)>0 order by bandwidth
desc
```

Dataset Name	Description	Log Category
drilldown-Top-User-By-Sessions-Table	Drilldown top user by session count	traffic

```
select
  user_src,
  sum(sessions) as sessions
from
  ###(select appid, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as
user_src, dstip, srcintf, dstintf, policyid, count(*) as sessions, sum(coalesce(sentbyte,
0)+coalesce(rcvdbyte, 0)) as bandwidth from $log where $filter-exclude-var and (logflag&1>0) group
by appid, app, user_src, dstip, srcintf, dstintf, policyid order by sessions desc)### t where
$filter-drilldown and user_src is not null group by user_src order by sessions desc
```

Dataset Name	Description	Log Category
drilldown-Top-User-By-Sessions-Bar	Drilldown top user by session count	traffic

```
select
  user_src,
  sum(sessions) as sessions
from
  ###(select appid, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as
user_src, dstip, srcintf, dstintf, policyid, count(*) as sessions, sum(coalesce(sentbyte,
0)+coalesce(rcvdbyte, 0)) as bandwidth from $log where $filter-exclude-var and (logflag&1>0) group
by appid, app, user_src, dstip, srcintf, dstintf, policyid order by sessions desc)### t where
$filter-drilldown and user_src is not null group by user_src order by sessions desc
```

Dataset Name	Description	Log Category
drilldown-Top-User-By-Bandwidth-Table	Drilldown top user by bandwidth usage	traffic

```
select
  user_src,
  sum(bandwidth) as bandwidth
from
  ###(select appid, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as
user_src, dstip, srcintf, dstintf, policyid, count(*) as sessions, sum(coalesce(sentbyte,
0)+coalesce(rcvdbyte, 0)) as bandwidth from $log where $filter-exclude-var and (logflag&1>0) group
by appid, app, user_src, dstip, srcintf, dstintf, policyid order by sessions desc)### t where
$filter-drilldown and user_src is not null group by user_src having sum(bandwidth)>0 order by
bandwidth desc
```

Dataset Name	Description	Log Category
drilldown-Top-User-By-Bandwidth-Bar	Drilldown top user by bandwidth usage	traffic

```
select
  user_src,
  sum(bandwidth) as bandwidth
from
  ###(select appid, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as
user_src, dstip, srcintf, dstintf, policyid, count(*) as sessions, sum(coalesce(sentbyte,
0)+coalesce(rcvdbyte, 0)) as bandwidth from $log where $filter-exclude-var and (logflag&1>0) group
by appid, app, user_src, dstip, srcintf, dstintf, policyid order by sessions desc)### t where
$filter-drilldown and user_src is not null group by user_src having sum(bandwidth)>0 order by
bandwidth desc
```

Dataset Name	Description	Log Category
drilldown-Top-Web-User-By-Visit-Table	Drilldown top web user by visit	traffic

```
select
  user_src,
  sum(requests) as visits
from
  (
    (
      select
        user_src,
        hostname,
        sum(requests) as requests
      from
        ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
hostname, count(*) as requests from $log-traffic where $filter-exclude-var and (logflag&1>0) and
utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') and
hostname is not null group by user_src, hostname order by requests desc)### t group by user_src,
hostname) union all (select user_src, hostname, sum(requests) as requests from ###(select coalesce
(nullifna(`user`), ipstr(`srcip`)) as user_src, hostname, count(*) as requests from $log-webfilter
where $filter-exclude-var and hostname is not null group by user_src, hostname order by requests
desc)### t group by user_src, hostname)) t where user_src is not null group by user_src order by
visits desc
```

Dataset Name	Description	Log Category
drilldown-Top-Web-User-By-Visit-Bar	Drilldown top web user by visit	traffic

```
select
  user_src,
  sum(requests) as visits
from
  (
    (
      select
        user_src,
        hostname,
        sum(requests) as requests
      from
        ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
hostname, count(*) as requests from $log-traffic where $filter-exclude-var and (logflag&1>0) and
utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') and
hostname is not null group by user_src, hostname order by requests desc)### t group by user_src,
hostname) union all (select user_src, hostname, sum(requests) as requests from ###(select coalesce
(nullifna(`user`), ipstr(`srcip`)) as user_src, hostname, count(*) as requests from $log-webfilter
where $filter-exclude-var and hostname is not null group by user_src, hostname order by requests
desc)### t group by user_src, hostname)) t where user_src is not null group by user_src order by
visits desc
```

Dataset Name	Description	Log Category
drilldown-Top-Website-By-Request-Table	Drilldown top website by request	traffic

```
select
  hostname,
  sum(requests) as visits
from
  (
    (
      select
        user_src,
        hostname,
        sum(requests) as requests
      from
        ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
hostname, count(*) as requests from $log-traffic where $filter-exclude-var and (logflag&1>0) and
utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') and
hostname is not null group by user_src, hostname order by requests desc)### t group by user_src,
hostname) union all (select user_src, hostname, sum(requests) as requests from ###(select coalesce
(nullifna(`user`), ipstr(`srcip`)) as user_src, hostname, count(*) as requests from $log-webfilter
where $filter-exclude-var and hostname is not null group by user_src, hostname order by requests
desc)### t group by user_src, hostname)) t where hostname is not null group by hostname order by
visits desc
```

Dataset Name	Description	Log Category
drilldown-Top-Website-By-Request-Bar	Drilldown top website by request	traffic

```
select
  hostname,
  sum(requests) as visits
from
  (
    (
      select
        user_src,
        hostname,
        sum(requests) as requests
      from
        ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
hostname, count(*) as requests from $log-traffic where $filter-exclude-var and (logflag&1>0) and
utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') and
hostname is not null group by user_src, hostname order by requests desc)### t group by user_src,
hostname) union all (select user_src, hostname, sum(requests) as requests from ###(select coalesce
(nullifna(`user`), ipstr(`srcip`)) as user_src, hostname, count(*) as requests from $log-webfilter
where $filter-exclude-var and hostname is not null group by user_src, hostname order by requests
desc)### t group by user_src, hostname)) t where hostname is not null group by hostname order by
visits desc
```

Dataset Name	Description	Log Category
drilldown-Top-Email-Sender-By-Volume	Drilldown top email sender by volume	traffic

```
select
  sender,
  sum(bandwidth) as volume
from
  (
    (
      select
        sender,
        recipient,
        sum(requests) as requests,
        sum(bandwidth) as bandwidth
      from
        ###(select sender, recipient, count(*) as requests, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth from $log-traffic where $filter-exclude-var and (logflag&1>0) and
service in ('smtp', 'SMTP', 'tcp/25', 'tcp/587', 'smtps', 'SMTPS', 'tcp/465') and utmevent in
('general-email-log', 'spamfilter') group by sender, recipient order by requests desc)### t group
by sender, recipient) union all (select sender, recipient, sum(requests) as requests, sum
(bandwidth) as bandwidth from ###(select `from` as sender, `to` as recipient, count(*) as
requests, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from $log-emailfilter
```

```
where $filter-exclude-var and service in ('smtp', 'SMTP', 'tcp/25', 'tcp/587', 'smtps', 'SMTPS',
'tcp/465') and eventtype is null group by `from`, `to` order by requests desc)### t group by
sender, recipient)) t where sender is not null group by sender having sum(bandwidth)>0 order by
volume desc
```

Dataset Name	Description	Log Category
drilldown-Top-Email-Send-Recipient-By-Volume	Drilldown top email send recipient by volume	traffic

```
select
  recipient,
  sum(bandwidth) as volume
from
  (
    (
      select
        sender,
        recipient,
        sum(requests) as requests,
        sum(bandwidth) as bandwidth
      from
        ###(select sender, recipient, count(*) as requests, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth from $log-traffic where $filter-exclude-var and (logflag&1>0) and
service in ('smtp', 'SMTP', 'tcp/25', 'tcp/587', 'smtps', 'SMTPS', 'tcp/465') and utmevent in
('general-email-log', 'spamfilter') group by sender, recipient order by requests desc)### t group
by sender, recipient) union all (select sender, recipient, sum(requests) as requests, sum
(bandwidth) as bandwidth from ###(select `from` as sender, `to` as recipient, count(*) as
requests, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from $log-emailfilter
where $filter-exclude-var and service in ('smtp', 'SMTP', 'tcp/25', 'tcp/587', 'smtps', 'SMTPS',
'tcp/465') and eventtype is null group by `from`, `to` order by requests desc)### t group by
sender, recipient)) t where recipient is not null group by recipient having sum(bandwidth)>0 order
by volume desc
```

Dataset Name	Description	Log Category
drilldown-Top-Email-Sender-By-Count	Drilldown top email sender by count	traffic

```
select
  sender,
  sum(requests) as requests
from
  (
    (
      select
        sender,
        recipient,
        sum(requests) as requests,
```

```
        sum(bandwidth) as bandwidth
      from
        ###(select sender, recipient, count(*) as requests, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth from $log-traffic where $filter-exclude-var and (logflag&1>0) and
service in ('smtp', 'SMTP', 'tcp/25', 'tcp/587', 'smtps', 'SMTPS', 'tcp/465') and utmevent in
('general-email-log', 'spamfilter') group by sender, recipient order by requests desc)### t group
by sender, recipient) union all (select sender, recipient, sum(requests) as requests, sum
(bandwidth) as bandwidth from ###(select `from` as sender, `to` as recipient, count(*) as
requests, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from $log-emailfilter
where $filter-exclude-var and service in ('smtp', 'SMTP', 'tcp/25', 'tcp/587', 'smtps', 'SMTPS',
'tcp/465') and eventtype is null group by `from`, `to` order by requests desc)### t group by
sender, recipient)) t where sender is not null group by sender order by requests desc
```

Dataset Name	Description	Log Category
drilldown-Top-Email-Send-Recipient-By-Count	Drilldown top email send recipient by count	traffic

```
select
  recipient,
  sum(requests) as requests
from
  (
    (
      select
        sender,
        recipient,
        sum(requests) as requests,
        sum(bandwidth) as bandwidth
      from
        ###(select sender, recipient, count(*) as requests, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth from $log-traffic where $filter-exclude-var and (logflag&1>0) and
service in ('smtp', 'SMTP', 'tcp/25', 'tcp/587', 'smtps', 'SMTPS', 'tcp/465') and utmevent in
('general-email-log', 'spamfilter') group by sender, recipient order by requests desc)### t group
by sender, recipient) union all (select sender, recipient, sum(requests) as requests, sum
(bandwidth) as bandwidth from ###(select `from` as sender, `to` as recipient, count(*) as
requests, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from $log-emailfilter
where $filter-exclude-var and service in ('smtp', 'SMTP', 'tcp/25', 'tcp/587', 'smtps', 'SMTPS',
'tcp/465') and eventtype is null group by `from`, `to` order by requests desc)### t group by
sender, recipient)) t where recipient is not null group by recipient order by requests desc
```

Dataset Name	Description	Log Category
drilldown-Top-Email-Recipient-By-Volume	Drilldown top email receiver by volume	traffic

```
select
  recipient,
  sum(bandwidth) as volume
```

```
from
  (
    (
      select
        recipient,
        sender,
        sum(requests) as requests,
        sum(bandwidth) as bandwidth
      from
        ###(select recipient, sender, count(*) as requests, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth from $log where $filter-exclude-var and (logflag&1>0) and service in
('pop3', 'POP3', 'tcp/110', 'imap', 'IMAP', 'tcp/143', 'imaps', 'IMAPS', 'tcp/993', 'pop3s',
'POP3S', 'tcp/995') and utmevent in ('general-email-log', 'spamfilter') group by recipient, sender
order by requests desc)### t group by sender, recipient) union all (select recipient, sender, sum
(requests) as requests, sum(bandwidth) as bandwidth  from ###(select `to` as recipient, `from` as
sender, count(*) as requests, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from
$log-emailfilter where $filter-exclude-var and service in ('pop3', 'POP3', 'tcp/110', 'imap',
'IMAP', 'tcp/143', 'imaps', 'IMAPS', 'tcp/993', 'pop3s', 'POP3S', 'tcp/995') and eventtype is null
group by `to`, `from` order by requests desc)### t group by sender, recipient)) t where recipient
is not null group by recipient having sum(bandwidth)>0 order by volume desc
```

Dataset Name	Description	Log Category
drilldown-Top-Email-Receive-Sender-By-Volume	Drilldown top email receive sender by volume	traffic

```
select
  sender,
  sum(bandwidth) as volume
from
  (
    (
      select
        recipient,
        sender,
        sum(requests) as requests,
        sum(bandwidth) as bandwidth
      from
        ###(select recipient, sender, count(*) as requests, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth from $log where $filter-exclude-var and (logflag&1>0) and service in
('pop3', 'POP3', 'tcp/110', 'imap', 'IMAP', 'tcp/143', 'imaps', 'IMAPS', 'tcp/993', 'pop3s',
'POP3S', 'tcp/995') and utmevent in ('general-email-log', 'spamfilter') group by recipient, sender
order by requests desc)### t group by sender, recipient) union all (select recipient, sender, sum
(requests) as requests, sum(bandwidth) as bandwidth  from ###(select `to` as recipient, `from` as
sender, count(*) as requests, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from
$log-emailfilter where $filter-exclude-var and service in ('pop3', 'POP3', 'tcp/110', 'imap',
'IMAP', 'tcp/143', 'imaps', 'IMAPS', 'tcp/993', 'pop3s', 'POP3S', 'tcp/995') and eventtype is null
group by `to`, `from` order by requests desc)### t group by sender, recipient)) t where sender is
not null group by sender having sum(bandwidth)>0 order by volume desc
```

Dataset Name	Description	Log Category
drilldown-Top-Email-Recipient-By-Count	Drilldown top email receiver by count	traffic

```
select
  recipient,
  sum(requests) as requests
from
  (
    (
      select
        recipient,
        sender,
        sum(requests) as requests,
        sum(bandwidth) as bandwidth
      from
        ###(select recipient, sender, count(*) as requests, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth from $log where $filter-exclude-var and (logflag&1>0) and service in
('pop3', 'POP3', 'tcp/110', 'imap', 'IMAP', 'tcp/143', 'imaps', 'IMAPS', 'tcp/993', 'pop3s',
'POP3S', 'tcp/995') and utmevent in ('general-email-log', 'spamfilter') group by recipient, sender
order by requests desc)### t group by sender, recipient) union all (select recipient, sender, sum
(requests) as requests, sum(bandwidth) as bandwidth  from ###(select `to` as recipient, `from` as
sender, count(*) as requests, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from
$log-emailfilter where $filter-exclude-var and service in ('pop3', 'POP3', 'tcp/110', 'imap',
'IMAP', 'tcp/143', 'imaps', 'IMAPS', 'tcp/993', 'pop3s', 'POP3S', 'tcp/995') and eventtype is null
group by `to`, `from` order by requests desc)### t group by sender, recipient)) t where recipient
is not null group by recipient order by requests desc
```

Dataset Name	Description	Log Category
drilldown-Top-Email-Receive-Sender-By-Count	Drilldown top email receive sender by count	traffic

```
select
  sender,
  sum(requests) as requests
from
  (
    (
      select
        recipient,
        sender,
        sum(requests) as requests,
        sum(bandwidth) as bandwidth
      from
        ###(select recipient, sender, count(*) as requests, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth from $log where $filter-exclude-var and (logflag&1>0) and service in
('pop3', 'POP3', 'tcp/110', 'imap', 'IMAP', 'tcp/143', 'imaps', 'IMAPS', 'tcp/993', 'pop3s',
'POP3S', 'tcp/995') and utmevent in ('general-email-log', 'spamfilter') group by recipient, sender
```

```
order by requests desc)### t group by sender, recipient) union all (select recipient, sender, sum
(requests) as requests, sum(bandwidth) as bandwidth  from ###(select `to` as recipient, `from` as
sender, count(*) as requests, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from
$log-emailfilter where $filter-exclude-var and service in ('pop3', 'POP3', 'tcp/110', 'imap',
'IMAP', 'tcp/143', 'imaps', 'IMAPS', 'tcp/993', 'pop3s', 'POP3S', 'tcp/995') and eventtype is null
group by `to`, `from` order by requests desc)### t group by sender, recipient)) t where sender is
not null group by sender order by requests desc
```

Dataset Name	Description	Log Category
drilldown-Top-Attack-Destination	Drilldown top attack dest	attack

```
select
  victim,
  sum(totalnum) as totalnum
from
  ###(select (CASE WHEN direction='incoming' THEN dstip ELSE srcip END) as source, (CASE WHEN
direction='incoming' THEN srcip ELSE dstip END) as victim, count(*) as totalnum from $log where
$filter-exclude-var group by source, victim order by totalnum desc)### t where $filter-drilldown
and victim is not null group by victim order by totalnum desc
```

Dataset Name	Description	Log Category
drilldown-Top-Attack-Source	Drilldown top attack source	attack

```
select
  source,
  sum(totalnum) as totalnum
from
  ###(select (CASE WHEN direction='incoming' THEN dstip ELSE srcip END) as source, (CASE WHEN
direction='incoming' THEN srcip ELSE dstip END) as victim, count(*) as totalnum from $log where
$filter-exclude-var group by source, victim order by totalnum desc)### t where $filter-drilldown
and source is not null group by source order by totalnum desc
```

Dataset Name	Description	Log Category
drilldown-Top-Attack-List	Drilldown top attack list	attack

```
select
  from_itime(itime) as timestamp,
  attack,
  source,
  victim
from
  ###(select itime, attack, (CASE WHEN direction='incoming' THEN dstip ELSE srcip END) as source,
(CASE WHEN direction='incoming' THEN srcip ELSE dstip END) as victim from $log where $filter-
exclude-var order by itime desc)### t where $filter-drilldown order by timestamp desc
```

Dataset Name	Description	Log Category
drilldown-Top-Virus	UTM top virus	virus

```
select
  virus,
  max(virusid_s) as virusid,
  (
    case when virus like & #039;Riskware%' then 'Spyware' when virus like 'Adware%' then 'Adware'
else 'Virus' end) as malware_type, sum(totalnum) as totalnum from ###(select virus, virusid_to_str
(virusid, eventtype) as virusid_s, count(*) as totalnum from $log where $filter and nullifna
(virus) is not null group by virus, virusid_s /*SkipSTART*/order by totalnum desc/*SkipEND*/)### t
group by virus, malware_type order by totalnum desc
```

Dataset Name	Description	Log Category
drilldown-Virus-Detail	Drilldown virus detail	virus

```
select
  from_itime(itime) as timestamp,
  virus,
  user_src,
  victim,
  hostname,
  recipient
from
  ###(select itime, virus, coalesce(nullifna(`user`), ipstr((CASE WHEN direction='incoming' THEN
dstip ELSE srcip END))) as user_src, (CASE WHEN direction='incoming' THEN srcip ELSE dstip END) as
victim, cast(' ' as char) as hostname, cast(' ' as char) as recipient from $log where $filter and
nullifna(virus) is not null order by itime desc)### t where $filter-drilldown order by timestamp
desc
```

Dataset Name	Description	Log Category
user-drilldown-Top-Blocked-Web-Sites-By-Requests	User drilldown top blocked web sites by requests	webfilter

```
select
  hostname,
  sum(requests) as requests
from
  ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as usersrc, euid, hostname, catdesc,
action, count(*) as requests from $log where $filter group by usersrc, euid, hostname, catdesc,
action order by requests desc)### t where $filter-drilldown and action='blocked' and hostname is
not null group by hostname order by requests desc
```

Dataset Name	Description	Log Category
user-drilldown-Top-Allowed-Web-Sites-By-Requests	User drilldown top allowed web sites by requests	webfilter

```
select
  hostname,
  sum(requests) as requests
from
  ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as usersrc, euid, hostname, catdesc,
action, count(*) as requests from $log where $filter group by usersrc, euid, hostname, catdesc,
action order by requests desc)### t where $filter-drilldown and action!='blocked' and hostname is
not null group by hostname order by requests desc
```

Dataset Name	Description	Log Category
user-drilldown-Top-Blocked-Web-Categories	User drilldown top blocked web categories	webfilter

```
select
  catdesc,
  sum(requests) as requests
from
  ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, catdesc, action, count(*) as
requests from $log where $filter and catdesc is not null group by user_src, catdesc, action order
by requests desc)### t where $filter-drilldown and action='blocked' group by catdesc order by
requests desc
```

Dataset Name	Description	Log Category
user-drilldown-Top-Allowed-Web-Categories	User drilldown top allowed web categories	webfilter

```
select
  catdesc,
  sum(requests) as requests
from
  ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, catdesc, action, count(*) as
requests from $log where $filter and catdesc is not null group by user_src, catdesc, action order
by requests desc)### t where $filter-drilldown and action!='blocked' group by catdesc order by
requests desc
```

Dataset Name	Description	Log Category
user-drilldown-Top-Attacks	User drilldown top attacks by name	attack

```
select
  attack,
  sum(attack_count) as attack_count
from
  ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, attack, (case when severity
in ('critical', 'high') then 1 else 0 end) as high_severity, count(*) as attack_count from $log
where $filter and nullifna(attack) is not null group by user_src, attack, high_severity order by
attack_count desc)### t where $filter-drilldown group by attack order by attack_count desc
```

Dataset Name	Description	Log Category
user-drilldown-Top-Attacks-High-Severity	User drilldown top attacks high severity	attack

```
select
  attack,
  sum(attack_count) as attack_count
from
  ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, attack, (case when severity
in ('critical', 'high') then 1 else 0 end) as high_severity, count(*) as attack_count from $log
where $filter and nullifna(attack) is not null group by user_src, attack, high_severity order by
attack_count desc)### t where $filter-drilldown and high_severity=1 group by attack order by
attack_count desc
```

Dataset Name	Description	Log Category
user-drilldown-Top-Virus-By-Name	User drilldown top virus	virus

```
select
  virus,
  max(virusid_s) as virusid,
  sum(totalnum) as totalnum
from
  ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, virus, virusid_to_str
(virusid, eventtype) as virusid_s, count(*) as totalnum from $log where $filter and nullifna
(virus) is not null group by user_src, virus, virusid_s order by totalnum desc)### t where
$filter-drilldown group by virus order by totalnum desc
```

Dataset Name	Description	Log Category
user-drilldown-Top-Virus-Receivers-Over-Email	User drilldown top virus receivers over email	virus

```
select
  receiver,
  sum(totalnum) as totalnum
from
  ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, `to` as receiver, count(*) as
```

```
totalnum from $log where $filter and eventtype='infected' and (service in ('smtp', 'SMTP',
'tcp/25', 'tcp/587', 'smtps', 'SMTPS', 'tcp/465') or service in ('pop3', 'POP3', 'tcp/110',
'imap', 'IMAP', 'tcp/143', 'imaps', 'IMAPS', 'tcp/993', 'pop3s', 'POP3S', 'tcp/995')) and nullifna
(virus) is not null group by user_src, receiver order by totalnum desc)### t where $filter-
drilldown group by receiver order by totalnum desc
```

Dataset Name	Description	Log Category
user-drilldown-Count-Spam-Activity-by-Hour-of-Day	User drilldown count spam activity by hour of day	emailfilter

```
select
  $hour_of_day(timestamp) as hourstamp,
  sum(totalnum) as totalnum
from
  ###(select $flex_timestamp as timestamp, coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src,
`from` as mf_sender, `to` as mf_receiver, action, eventtype, count(*) as totalnum from $log where
$filter group by timestamp, user_src, mf_sender, mf_receiver, action, eventtype /*SkipSTART*/order
by timestamp desc/*SkipEND*/)### t where $filter-drilldown and mf_receiver is not null and action
in ('detected', 'blocked') group by hourstamp order by hourstamp
```

Dataset Name	Description	Log Category
user-drilldown-Top-Spam-Sources	User drilldown top spam sources	emailfilter

```
select
  mf_sender,
  sum(totalnum) as totalnum
from
  ###(select $flex_timestamp as timestamp, coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src,
`from` as mf_sender, `to` as mf_receiver, action, eventtype, count(*) as totalnum from $log where
$filter group by timestamp, user_src, mf_sender, mf_receiver, action, eventtype /*SkipSTART*/order
by timestamp desc/*SkipEND*/)### t where $filter-drilldown and mf_sender is not null and action in
('detected', 'blocked') group by mf_sender order by totalnum desc
```

Dataset Name	Description	Log Category
event-Usage-CPU	Event usage CPU	event

```
select
  $hour_of_day(timestamp) as hourstamp,
  cast(
    sum(total_cpu)/ sum(count) as decimal(6, 2)
  ) as cpu_avg_usage
from
  ###(select $flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_trate,
sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min(itime) as
first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) as
```

```
mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum(coalesce
(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max(coalesce(trate, 0)+coalesce(erate,
0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalsession, max
(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part(bandwidth, '/', 1), '0')
as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as recv,
max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_part
(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max
(coalesce(setuprate, 0)) as cps_peak, count(*) as count from $log where $filter and
subtype='system' and action='perf-stats' group by timestamp, devid, slot order by total_mem
desc)### t group by hourstamp order by hourstamp
```

Dataset Name	Description	Log Category
event-Usage-Memory	Event usage memory	event

```
select
  $hour_of_day(timestamp) as hourstamp,
  cast(
    sum(total_mem)/ sum(count) as decimal(6, 2)
  ) as mem_avg_usage
from
  ###(select $flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_trate,
sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min(itime) as
first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) as
mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum(coalesce
(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max(coalesce(trate, 0)+coalesce(erate,
0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalsession, max
(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part(bandwidth, '/', 1), '0')
as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as recv,
max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_part
(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max
(coalesce(setuprate, 0)) as cps_peak, count(*) as count from $log where $filter and
subtype='system' and action='perf-stats' group by timestamp, devid, slot order by total_mem
desc)### t group by hourstamp order by hourstamp
```

Dataset Name	Description	Log Category
event-Usage-Sessions	Event usage sessions	event

```
select
  $hour_of_day(timestamp) as hourstamp,
  cast(
    sum(totalsession)/ sum(count) as decimal(10, 2)
  ) as sess_avg_usage
from
  ###(select $flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_trate,
sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min(itime) as
first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) as
mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum(coalesce
```

```
(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max(coalesce(trate, 0)+coalesce(erate,
0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalsession, max
(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part(bandwidth, '/', 1), '0')
as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as recv,
max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_part
(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max
(coalesce(setuprate, 0)) as cps_peak, count(*) as count from $log where $filter and
subtype='system' and action='perf-stats' group by timestamp, devid, slot order by total_mem
desc)### t group by hourstamp order by hourstamp
```

Dataset Name	Description	Log Category
event-Usage-CPU-Sessions	Event usage CPU sessions	event

```
select
  $hour_of_day(timestamp) as hourstamp,
  cast(
    sum(totalsession)/ sum(count) as decimal(10, 2)
  ) as sess_avg_usage,
  cast(
    sum(total_cpu)/ sum(count) as decimal(6, 2)
  ) as cpu_avg_usage
from
  ###(select $flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_trate,
sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min(itime) as
first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) as
mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum(coalesce
(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max(coalesce(trate, 0)+coalesce(erate,
0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalsession, max
(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part(bandwidth, '/', 1), '0')
as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as recv,
max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_part
(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max
(coalesce(setuprate, 0)) as cps_peak, count(*) as count from $log where $filter and
subtype='system' and action='perf-stats' group by timestamp, devid, slot order by total_mem
desc)### t group by hourstamp order by hourstamp
```

Dataset Name	Description	Log Category
App-Risk-Top-Users-By-Bandwidth	Top users by bandwidth usage	traffic

```
select
  coalesce(
    nullifna(`user`),
    nullifna(`unauthuser`),
    ipstr(`srcip`)
  ) as user_src,
  srcip,
  sum(
```

```
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  ) as bandwidth,
  sum(
    coalesce(rcvdbyte, 0)
  ) as traffic_in,
  sum(
    coalesce(sentbyte, 0)
  ) as traffic_out
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
  and srcip is not null
group by
  user_src,
  srcip
having
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  )> 0
order by
  bandwidth desc
```

Dataset Name	Description	Log Category
App-Risk-Top-User-Source-By-Sessions	Application risk top user source by session count	traffic

```
select
  srcip,
  coalesce(
    nullifna(`user`),
    nullifna(`unauthuser`),
    ipstr(`srcip`)
  ) as user_src,
  count(*) as sessions
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
  and srcip is not null
group by
  srcip,
  user_src
```

```
order by
  sessions desc
```

Dataset Name	Description	Log Category
App-Risk-Top-Users-By-Reputation-Scores-Bar	Application risk reputation top users by scores	traffic

```
select
  coalesce(
    nullifna(`user`),
    nullifna(`unauthuser`),
    ipstr(`srcip`)
  ) as user_src,
  sum(crscore % 65536) as scores
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
  and crscore is not null
group by
  user_src
having
  sum(crscore % 65536)> 0
order by
  scores desc
```

Dataset Name	Description	Log Category
App-Risk-Top-Devices-By-Reputation-Scores	Application risk reputation top devices by scores	traffic

```
select
  max(
    get_devtype(srcswversion, osname, devtype)
  ) as devtype_new,
  coalesce(
    nullifna(`srcname`),
    nullifna(`srcmac`),
    ipstr(`srcip`)
  ) as dev_src,
  sum(crscore % 65536) as scores
from
  $log
where
  $filter
```

```
  and (
    logflag&1>0
  )
  and crscore is not null
group by
  dev_src
having
  sum(crscore % 65536)> 0
order by
  scores desc
```

Dataset Name	Description	Log Category
App-Risk-Application-Usage-By-Category-With-Pie	Application Risk Application Usage by Category	traffic

```
select
  appcat,
  sum(bandwidth) as bandwidth
from
  ###(select appid, app, appcat, apprisk, user_src, hostname, dstip, sum(traffic_in) as traffic_
in, sum(traffic_out) as traffic_out, sum(bandwidth) as bandwidth, sum(sessions) as sessions from
###base(/*tag:rpt_base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid,
euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service,
appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions
from $log-traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by
timestamp, dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat, apprisk,
hostname order by bandwidth desc, sessions desc)base### t group by appid, app, appcat, apprisk,
user_src, hostname, dstip /*SkipSTART*/order by bandwidth desc, sessions desc/*SkipEND*/)### t
where $filter-drilldown and nullifna(appcat) is not null group by appcat having sum(bandwidth)>0
order by bandwidth desc
```

Dataset Name	Description	Log Category
App-Risk-App-Usage-by-Category	Application Risk Application Usage by Category	traffic

```
select
  appcat,
  sum(bandwidth) as bandwidth
from
  ###(select appid, app, appcat, apprisk, user_src, hostname, dstip, sum(traffic_in) as traffic_
in, sum(traffic_out) as traffic_out, sum(bandwidth) as bandwidth, sum(sessions) as sessions from
###base(/*tag:rpt_base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid,
euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service,
appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions
```

```
from $log-traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by
timestamp, dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat, apprisk,
hostname order by bandwidth desc, sessions desc)base### t group by appid, app, appcat, apprisk,
user_src, hostname, dstip /*SkipSTART*/order by bandwidth desc, sessions desc/*SkipEND*/)### t
where $filter-drilldown and nullifna(appcat) is not null group by appcat having sum(bandwidth)>0
order by bandwidth desc
```

Dataset Name	Description	Log Category
Top-20-Categories-By-Bandwidth	Webfilter categories by bandwidth usage	traffic

```
select
  catdesc,
  sum(bandwidth) as bandwidth
from
  ###(select catdesc, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from $log
where $filter and (logflag&1>0) and (countweb>0 or ((logver is null or logver<502000000) and
(hostname is not null or utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block',
'script-filter')))) and catdesc is not null group by catdesc /*SkipSTART*/order by bandwidth
desc/*SkipEND*/)### t group by catdesc order by bandwidth desc
```

Dataset Name	Description	Log Category
App-Risk-Key-Applications-Crossing-The-Network	Application risk application activity	traffic

```
select
  app_group,
  appcat,
  sum(bandwidth) as bandwidth,
  sum(sessions) as num_session
from
  ###(select app_group_name(app) as app_group, appcat, service, sum(coalesce(sentdelta, sentbyte,
0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as
traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from $log
where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by app_group, appcat,
service order by bandwidth desc)### t group by app_group, appcat order by bandwidth desc
```

Dataset Name	Description	Log Category
App-Risk-Applications-Running-Over-HTTP	Application risk applications running over HTTP	traffic

```
select
  app_group,
  service,
  sum(sessions) as sessions,
  sum(bandwidth) as bandwidth
```

```
from
  ###(select app_group_name(app) as app_group, appcat, service, sum(coalesce(sentdelta, sentbyte,
0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as
traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from $log
where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by app_group, appcat,
service order by bandwidth desc)### t where service in ('tcp/80', 'tcp/443', 'HTTP', 'HTTPS',
'http', 'https') group by app_group, service having sum(bandwidth)>0 order by bandwidth desc
```

Dataset Name	Description	Log Category
App-Risk-Top-Web-Sites-Visited-By-Network-Users-Pie-Cha	Application risk web browsing summary category	traffic

```
select
  catdesc,
  sum(num_sess) as num_sess,
  sum(bandwidth) as bandwidth
from
  ###(select catdesc, count(*) as num_sess, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as
bandwidth from $log-traffic where $filter and (logflag&1>0) and (countweb>0 or ((logver is null or
logver<502000000) and (hostname is not null or utmevent in ('webfilter', 'banned-word', 'web-
content', 'command-block', 'script-filter')))) and catdesc is not null group by catdesc order by
num_sess desc)### t group by catdesc order by num_sess desc
```

Dataset Name	Description	Log Category
App-Risk-Top-Web-Sites-Visited-By-Network-Users	Application risk web browsing summary category	traffic

```
select
  catdesc,
  sum(num_sess) as num_sess,
  sum(bandwidth) as bandwidth
from
  ###(select catdesc, count(*) as num_sess, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as
bandwidth from $log-traffic where $filter and (logflag&1>0) and (countweb>0 or ((logver is null or
logver<502000000) and (hostname is not null or utmevent in ('webfilter', 'banned-word', 'web-
content', 'command-block', 'script-filter')))) and catdesc is not null group by catdesc order by
num_sess desc)### t group by catdesc order by num_sess desc
```

Dataset Name	Description	Log Category
App-Risk-Web-Browsing-Hostname-Category	Application risk web browsing activity hostname category	webfilter

```
select
  catdesc,
  domain,
```

```
  sum(visits) as visits
from
  ###(select coalesce(nullifna(hostname), ipstr(`dstip`)) as domain, catdesc, count(*) as visits
from $log where $filter and catdesc is not null group by domain, catdesc order by visits desc)###
t group by catdesc, domain order by visits desc
```

Dataset Name	Description	Log Category
Top-Destination-Countries-By-Browsing-Time	Traffic top destination countries by browsing time	traffic

```
select
  dstcountry,
  ebtr_value(
    ebtr_agg_flat(browsetime),
    null,
    $timespan
  ) as browsetime,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out
from
  ###(select dstcountry, ebtr_agg_flat(browsetime) as browsetime, sum(bandwidth) as bandwidth, sum
(traffic_in) as traffic_in, sum(traffic_out) as traffic_out from (select dstcountry, ebtr_agg_flat
($browse_time) as browsetime, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum
(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out from $log where
$filter and (logflag&1>0) and $browse_time is not null group by dstcountry) t group by dstcountry
/*SkipSTART*/order by ebtr_value(ebtr_agg_flat(browsetime), null, null) desc/*SkipEND*/)### t
group by dstcountry order by browsetime desc
```

Dataset Name	Description	Log Category
App-Risk-Traffic-Top-Hostnames-By-Browsing-Time	Traffic top domains by browsing time	traffic

```
select
  hostname,
  ebtr_value(
    ebtr_agg_flat(browsetime),
    null,
    $timespan
  ) as browsetime,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out
from
  ###(select hostname, ebtr_agg_flat(browsetime) as browsetime, sum(bandwidth) as bandwidth, sum
(traffic_in) as traffic_in, sum(traffic_out) as traffic_out from (select hostname, ebtr_agg_flat
($browse_time) as browsetime, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum
```

```
(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out from $log where
$filter and (logflag&1>0) and hostname is not null and $browse_time is not null group by hostname)
t group by hostname /*SkipSTART*/order by ebtr_value(ebtr_agg_flat(browsetime), null, null)
desc/*SkipEND*/)### t group by hostname order by browsetime desc
```

Dataset Name	Description	Log Category
App-Risk-Top-Threat-Vectors-Crossing-The-Network	Application risk top threat vectors	attack

```
select
  severity,
  sum(totalnum) as totalnum
from
  ###(select attack, severity, ref, count(*) as totalnum from $log where $filter and nullifna
(attack) is not null group by attack, severity, ref order by totalnum desc)### t group by severity
order by totalnum desc
```

Dataset Name	Description	Log Category
App-Risk-Top-Critical-Threat-Vectors-Crossing-The-Network	Application risk top critical threat vectors	attack

```
select
  attack,
  severity,
  ref,
  sum(totalnum) as totalnum
from
  ###(select attack, severity, ref, count(*) as totalnum from $log where $filter and nullifna
(attack) is not null group by attack, severity, ref order by totalnum desc)### t where
severity='critical' group by attack, severity, ref order by totalnum desc
```

Dataset Name	Description	Log Category
App-Risk-Top-High-Threat-Vectors-Crossing-The-Network	Application risk top high threat vectors	attack

```
select
  attack,
  severity,
  ref,
  sum(totalnum) as totalnum
from
  ###(select attack, severity, ref, count(*) as totalnum from $log where $filter and nullifna
(attack) is not null group by attack, severity, ref order by totalnum desc)### t where
severity='high' group by attack, severity, ref order by totalnum desc
```

Dataset Name	Description	Log Category
App-Risk-Top-Medium-Threat-Vectors-Crossing-The-Network	Application risk top medium threat vectors	attack

```
select
  attack,
  severity,
  ref,
  sum(totalnum) as totalnum
from
  ###(select attack, severity, ref, count(*) as totalnum from $log where $filter and nullifna
(attack) is not null group by attack, severity, ref order by totalnum desc)### t where
severity='medium' group by attack, severity, ref order by totalnum desc
```

Dataset Name	Description	Log Category
App-Risk-Top-Low-Threat-Vectors-Crossing-The-Network	Application risk top low threat vectors	attack

```
select
  attack,
  severity,
  ref,
  sum(totalnum) as totalnum
from
  ###(select attack, severity, ref, count(*) as totalnum from $log where $filter and nullifna
(attack) is not null group by attack, severity, ref order by totalnum desc)### t where
severity='low' group by attack, severity, ref order by totalnum desc
```

Dataset Name	Description	Log Category
App-Risk-Top-Info-Threat-Vectors-Crossing-The-Network	Application risk top info threat vectors	attack

```
select
  attack,
  severity,
  ref,
  sum(totalnum) as totalnum
from
  ###(select attack, severity, ref, count(*) as totalnum from $log where $filter and nullifna
(attack) is not null group by attack, severity, ref order by totalnum desc)### t where
severity='info' group by attack, severity, ref order by totalnum desc
```

Dataset Name	Description	Log Category
App-Risk-Top-Virus-By-Name	UTM top virus	virus

```
select
  virus,
  max(virusid_s) as virusid,
  (
    case when virus like & #039;Riskware%' then 'Spyware' when virus like 'Adware%' then 'Adware'
else 'Virus' end) as malware_type, sum(totalnum) as totalnum from ###(select virus, virusid_to_str
(virusid, eventtype) as virusid_s, count(*) as totalnum from $log where $filter and nullifna
(virus) is not null group by virus, virusid_s /*SkipSTART*/order by totalnum desc/*SkipEND*/)### t
group by virus, malware_type order by totalnum desc
```

Dataset Name	Description	Log Category
App-Risk-Top-Virus-Victim	UTM top virus user	virus

```
select
  user_src,
  sum(totalnum) as totalnum
from
  ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, eventtype, logver, virus,
count(*) as totalnum from $log where $filter group by user_src, eventtype, logver, virus
/*SkipSTART*/order by totalnum desc/*SkipEND*/)### t where nullifna(virus) is not null group by
user_src order by totalnum desc
```

Dataset Name	Description	Log Category
App-Risk-Data-Loss-Prevention-Type-Events	Application risk DLP UTM event	dlp

```
select
  subtype : :text as utmsubtype,
  count(*) as number
from
  ###(select itime, hostname,`from` as sender, `to` as receiver, profile, action, service,
subtype, srcip, dstip, severity, filename, direction, filesize, (case when severity='critical'
then 'Critical Data Exfiltration' else (case when coalesce(nullifna(`user`), ipstr(`srcip`)) is
not null then 'User Associated Data Loss' else NULL end) end) as data_loss from $log where $filter
/*SkipSTART*/order by itime desc/*SkipEND*/)### t where $filter-drilldown and subtype is not null
group by subtype order by number desc
```

Dataset Name	Description	Log Category
App-Risk-Vulnerability-Discovered	Application risk vulnerability discovered	netscan

```
select
  vuln,
  vulnref as ref,
  vulncat,
  severity,
  count(*) as totalnum
from
  $log
where
  $filter
  and vuln is not null
group by
  vuln,
  vulnref,
  vulncat,
  severity
order by
  totalnum desc
```

Dataset Name	Description	Log Category
App-Risk-Malware-Discovered	Application risk virus discovered	virus

```
select
  dom,
  sum(totalnum) as totalnum
from
  ###(select $DAY_OF_MONTH as dom, count(*) as totalnum from $log where $filter and nullifna
(virus) is not null group by dom order by totalnum desc)### t group by dom order by totalnum desc
```

Dataset Name	Description	Log Category
App-Risk-Breakdown-Of-Risk-Applications	Application risk breakdown of risk applications	traffic

```
select
  unnest(
    string_to_array(
      behavior,
      & #039;,')) as d_behavior, count(*) as number from $log t1 inner join app_mdata t2 on
t1.appid=t2.id where $filter and (logflag&1>0) group by d_behavior order by number desc
```

Dataset Name	Description	Log Category
App-Risk-Number-Of-Applications-By-Risk-Behavior	Application risk number of applications by risk behavior	traffic

```
select
  risk as d_risk,
  unnest(
    string_to_array(
      behavior,
      & #039;,')) as f_behavior, count(*) as number from $log t1 inner join app_mdata t2 on
t1.appid=t2.id where $filter and (logflag&1>0) group by risk, f_behavior order by risk desc,
number desc
```

Dataset Name	Description	Log Category
App-Risk-High-Risk-Application	Application risk high risk application	traffic

```
select
  risk as d_risk,
  behavior as d_behavior,
  t2.id,
  t2.name,
  t2.app_cat,
  t2.technology,
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  ) as bandwidth,
  count(*) as sessions
from
  $log t1
  inner join app_mdata t2 on t1.appid = t2.id
where
  $filter
  and (
    logflag&1>0
  )
  and behavior is not null
group by
  t2.id
order by
  risk desc,
  sessions desc
```

Dataset Name	Description	Log Category
Apprisk-Ctrl-Breakdown-Of-High-Risk-Application	Severe and high risk applications	traffic

```
select
  appcat,
  count(distinct app) as total_num
from
  ###(select appid, app, appcat, apprisk, sum(bandwidth) as bandwidth, sum(sessions) as sessions
```

```
from ###base(/*tag:rpt_base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip,
epid, euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
service, appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_
in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte,
0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END)
as sessions from $log-traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null
group by timestamp, dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat,
apprisk, hostname order by bandwidth desc, sessions desc)base### t group by appid, app, appcat,
apprisk /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t where $filter-
drilldown and nullifna(appcat) is not null and apprisk in ('critical', 'high') group by appcat
order by total_num desc
```

Dataset Name	Description	Log Category
Apprisk-Ctrl-High-Risk-Application-Behavioral	Application Behavioral Characteristics	traffic

```
select
  behavior,
  round(
    sum(total_num)* 100 / sum(
      sum(total_num)
    ) over (),
    2
  ) as percentage
from
  (
    (
      select
        timestamp,
        behavior,
        sum(total_num) as total_num
      from
        ###(select timestamp, (case when lower(appcat)='botnet' then 'malicious' when lower
(appcat)='remote.access' then 'tunneling' when lower(appcat) in ('storage.backup', 'video/audio')
then 'bandwidth-consuming' when lower(appcat)='p2p' then 'peer-to-peer' when lower(appcat)='proxy'
then 'proxy' end) as behavior, sum(sessions) as total_num from ###base(/*tag:rpt_base_t_bndwdth_
sess*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid, euid, appcat, apprisk,
coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service, sum(CASE
WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum
(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in from $log-traffic where $filter and (logflag&
(1|32)>0) group by timestamp, dvid, srcip, dstip, epid, euid, appcat, apprisk, user_src, service
/*SkipSTART*/order by bandwidth desc, sessions desc/*SkipEND*/)base### t where lower(appcat) in
('botnet', 'remote.access', 'storage.backup', 'video/audio', 'p2p', 'proxy') and apprisk in
('critical', 'high') group by timestamp, behavior order by total_num desc)### t group by
timestamp, behavior) union all (select timestamp, behavior, sum(total_num) as total_num from ###
(select $flex_timestamp as timestamp, 'malicious' as behavior,  count(*) as total_num from $log-
attack where $filter and (logflag&16>0) and severity in ('critical', 'high') group by timestamp,
behavior order by total_num desc)### t group by timestamp, behavior)) t group by behavior order by
percentage desc
```

Dataset Name	Description	Log Category
Apprisk-Ctrl-Key-Application-Crossing-The-Network	Key Application Crossing The Network	traffic

```
select
  risk as d_risk,
  count(distinct user_src) as users,
  id,
  name,
  app_cat,
  technology,
  sum(bandwidth) as bandwidth,
  sum(sessions) as sessions
from
  ###(select app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, count(*) as sessions from $log
where $filter and (logflag&1>0) group by app, user_src order by bandwidth desc)### t1 inner join
app_mdata t2 on t1.app=t2.name group by id, app, app_cat, technology, risk order by bandwidth desc
```

Dataset Name	Description	Log Category
Apprisk-Ctrl-Risk-Application-Usage-By-Category-With-Pie	Application Risk Application Usage by Category	traffic

```
select
  appcat,
  sum(bandwidth) as bandwidth
from
  ###(select appid, app, appcat, apprisk, user_src, hostname, dstip, sum(traffic_in) as traffic_
in, sum(traffic_out) as traffic_out, sum(bandwidth) as bandwidth, sum(sessions) as sessions from
###base(/*tag:rpt_base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid,
euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service,
appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions
from $log-traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by
timestamp, dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat, apprisk,
hostname order by bandwidth desc, sessions desc)base### t group by appid, app, appcat, apprisk,
user_src, hostname, dstip /*SkipSTART*/order by bandwidth desc, sessions desc/*SkipEND*/)### t
where $filter-drilldown and nullifna(appcat) is not null group by appcat having sum(bandwidth)>0
order by bandwidth desc
```

Dataset Name	Description	Log Category
Apprisk-Ctrl-Risk-Application-Usage-By-Category-Pie	Application Risk Application Usage by Category	traffic

```
select
  appcat,
  sum(bandwidth) as bandwidth
from
  ###(select appid, app, appcat, apprisk, user_src, hostname, dstip, sum(traffic_in) as traffic_
in, sum(traffic_out) as traffic_out, sum(bandwidth) as bandwidth, sum(sessions) as sessions from
###base(/*tag:rpt_base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid,
euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service,
appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions
from $log-traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by
timestamp, dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat, apprisk,
hostname order by bandwidth desc, sessions desc)base### t group by appid, app, appcat, apprisk,
user_src, hostname, dstip /*SkipSTART*/order by bandwidth desc, sessions desc/*SkipEND*/)### t
where $filter-drilldown and nullifna(appcat) is not null group by appcat having sum(bandwidth)>0
order by bandwidth desc
```

Dataset Name	Description	Log Category
App-Usage-Timeline	Application Category with Most Average Bandwidth Used	traffic

```
select
  $flex_timestamp(timestamp) as hodex,
  sum(bandwidth) as bandwidth
from
  ###(select timestamp, app, appcat, user_src, hostname, sum(traffic_in) as traffic_in, sum
(traffic_out) as traffic_out, sum(bandwidth) as bandwidth, sum(sessions) as sessions from ###base
(/*tag:rpt_base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid, euid,
coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service, appid,
app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce
(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta,
rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions from $log-
traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by timestamp,
dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat, apprisk, hostname order by
bandwidth desc, sessions desc)base### t group by timestamp, app, appcat, user_src, hostname
/*SkipSTART*/order by bandwidth desc, sessions desc/*SkipEND*/)### t group by hodex order by hodex
```

Dataset Name	Description	Log Category
Apprisk-Ctrl-Category-Breakdown-By-Bandwidth	Category breakdown of all applications, sorted by bandwidth	traffic

```
select
  appcat,
  count(distinct app) as app_num,
  count(distinct user_src) as user_num,
  sum(bandwidth) as bandwidth,
```

```
  sum(sessions) as num_session
from
  ###(select app, appcat, user_src, sum(bandwidth) as bandwidth, sum(sessions) as sessions from
###base(/*tag:rpt_base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid,
euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service,
appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions
from $log-traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by
timestamp, dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat, apprisk,
hostname order by bandwidth desc, sessions desc)base### t where nullifna(appcat) is not null and
appcat not in ('Not.Scanned', 'unscanned', 'unknown') group by app, appcat, user_src order by
bandwidth desc)### t where $filter-drilldown group by appcat order by bandwidth desc
```

Dataset Name	Description	Log Category
Apprisk-Ctrl-Common-Virus-Botnet-Spyware	Common virus disvocered, the botnet communictions and the spyware/adware	traffic

```
select
  virus_s as virus,
  (
    case when lower(appcat)=& #039;botnet' then 'Botnet C&C' else (case when virus_s like
'Riskware%' then 'Spyware' when virus_s like 'Adware%' then 'Adware' else 'Virus' end) end) as
malware_type, appid, app, count(distinct dstip) as victims, count(distinct srcip) as source, sum
(total_num) as total_num from ((select virus_s, appcat, appid, app, dstip, srcip, sum(total_num)
as total_num from ###(select app as virus_s, appcat, appid, app, dstip, srcip, count(*) as total_
num from $log-traffic where $filter and (logflag&1>0) and lower(appcat)='botnet' group by virus_s,
appcat, appid, dstip, srcip, app order by total_num desc)### t group by virus_s, appcat, appid,
dstip, srcip, app) union all (select virus_s, appcat, appid, app, dstip, srcip, sum(total_num) as
total_num from ###(select unnest(string_to_array(virus, ',')) as virus_s, appcat, appid, app,
dstip, srcip, count(*) as total_num from $log-traffic where $filter and (logflag&1>0) and virus is
not null group by virus_s, appcat, appid, dstip, srcip, app order by total_num desc)### t group by
virus_s, appcat, appid, dstip, srcip, app) union all (select virus_s, appcat, appid, app, dstip,
srcip, sum(total_num) as total_num from ###(select attack as virus_s, 'botnet' as appcat, 0 as
appid, attack as app, dstip, srcip, count(*) as total_num from $log-attack where $filter and
(logflag&16>0) group by virus_s, appcat, appid, dstip, srcip, app order by total_num desc)### t
group by virus_s, appcat, appid, dstip, srcip, app)) t group by virus, appid, app, malware_type
order by total_num desc
```

Dataset Name	Description	Log Category
Apprisk-Ctrl-Zero-Day-Detected-On-Network	Zero-day malware detected on the network	traffic

```
select
  virus_s,
  appid,
  app,
```

```
  count(distinct dstip) as victims,
  count(distinct srcip) as source,
  sum(total_num) as total_num
from
  ###(select unnest(string_to_array(virus, ',')) as virus_s, appid, app, dstip, srcip, count(*) as
total_num from $log where $filter and (logflag&1>0) and virus like '%PossibleThreat.SB%' group by
virus_s, dstip, srcip, appid, app order by total_num desc)### t where virus_s like
'%PossibleThreat.SB%' group by virus_s, appid, app  order by total_num desc
```

Dataset Name	Description	Log Category
Apprisk-Ctrl-Files-Analyzed-By-FortiCloud-Sandbox	Files analyzed by FortiCloud Sandbox	virus

```
select
  $DAY_OF_MONTH as dom,
  count(*) as total_num
from
  $log
where
  $filter
  and nullifna(filename) is not null
  and logid_to_int(logid)= 9233
group by
  dom
order by
  dom
```

Dataset Name	Description	Log Category
Apprisk-Ctrl-File-Transferred-By-Application	File transferred by applications on the network	app-ctrl

```
select
  appid,
  app,
  filename,
  cloudaction,
  max(filesize) as filesize
from
  $log
where
  $filter
  and filesize is not null
  and clouduser is not null
  and filename is not null
group by
  cloudaction,
  appid,
```

```
  app,
  filename
order by
  filesize desc
```

Dataset Name	Description	Log Category
appctrl-Top-Blocked-SCCP-Callers	Appctrl top blocked SCCP callers	app-ctrl

```
select
  caller,
  sum(totalnum) as totalnum
from
  ###(select srcname as caller, app, count(*) as totalnum from $log where $filter and srcname is
not null and lower(appcat)='voip' and action='block' group by caller, app order by totalnum
desc)### t where app='sccp' group by caller order by totalnum desc
```

Dataset Name	Description	Log Category
appctrl-Top-Blocked-SIP-Callers	Appctrl top blocked SIP callers	app-ctrl

```
select
  caller,
  sum(totalnum) as totalnum
from
  ###(select srcname as caller, app, count(*) as totalnum from $log where $filter and srcname is
not null and lower(appcat)='voip' and action='block' group by caller, app order by totalnum
desc)### t where app='sip' group by caller order by totalnum desc
```

Dataset Name	Description	Log Category
360-degree-security-Application-Visiblity-and-Control-Summary	Application Visibolity and Control Summary	app-ctrl

```
select
  appcat,
  count(distinct app) as total_num
from
  ###(select appcat, app, count(*) as total_num from $log where $filter and app is not null and
appcat is not null group by appcat, app order by total_num desc)### t group by appcat order by
total_num desc
```

Dataset Name	Description	Log Category
360-degree-security-Threats-Detection-and-Prevention-Summary	Threat Prevention	app-ctrl

```
select
  threat_name,
  count(distinct threats) as total_num
from
  (
    (
      select
        threat_name,
        threats,
        sum(total_num) as total_num
      from
        ###(select cast('Malware & Botnet C&C' as char(32)) as threat_name, app as threats, count
(*) as total_num from $log-app-ctrl where $filter and lower(appcat)='botnet' group by app order by
total_num desc)### t group by threat_name, threats) union all (select threat_name, threats, sum
(total_num) as total_num from ###(select cast('Malware & Botnet C&C' as char(32)) as threat_name,
virus as threats, count(*) as total_num from $log-virus where $filter and nullifna(virus) is not
null group by virus order by total_num desc)### t group by threat_name, threats) union all (select
threat_name, threats, sum(total_num) as total_num from ###(select cast('Malicious & Phishing
Sites' as char(32)) as threat_name, hostname as threats, count(*) as total_num from $log-webfilter
where $filter and cat in (26, 61) group by hostname order by total_num desc)### t group by threat_
name, threats) union all (select threat_name, threats, sum(total_num) as total_num from ###(select
cast('Critical & High Intrusion Attacks' as char(32)) as threat_name, attack as threats, count(*)
as total_num from $log-attack where $filter and severity in ('critical', 'high') group by attack
order by total_num desc)### t group by threat_name, threats)) t group by threat_name order by
total_num desc
```

Dataset Name	Description	Log Category
360-degree-security-Data-Exfiltration-Detection-and-Prevention-Summary	Data Exfiltration Summary	dlp

```
select
  data_loss,
  count(*) as total_num
from
  ###(select itime, hostname,`from` as sender, `to` as receiver, profile, action, service,
subtype, srcip, dstip, severity, filename, direction, filesize, (case when severity='critical'
then 'Critical Data Exfiltration' else (case when coalesce(nullifna(`user`), ipstr(`srcip`)) is
not null then 'User Associated Data Loss' else NULL end) end) as data_loss from $log where $filter
/*SkipSTART*/order by itime desc/*SkipEND*/)### t where $filter-drilldown and data_loss is not
null group by data_loss order by total_num desc
```

Dataset Name	Description	Log Category
360-degree-security-Endpoint-Protection-Summary	Endpoint Protection	fct-traffic

```
select
  blocked_event,
  count(*) as total_num
from
  (
    select
      (
        case utmevent when & #039;antivirus' then 'Malware Deteced and Blocked' when 'appfirewall'
then 'Risk Application Blocked' when 'webfilter' then (case when coalesce(nullifna(`user`),  ipstr
(`srcip`)) is not null then 'Web Sites Violation Blocked' else 'Non User Initiated Web Visits'
end) else NULL end) as blocked_event from $log where $filter and utmaction in ('blocked',
'quarantined')) t where blocked_event is not null group by blocked_event order by total_num desc
```

Dataset Name	Description	Log Category
security-Top20-High-Risk-Application-In-Use	High risk application in use	traffic

```
select
  d_risk,
  count(distinct f_user) as users,
  name,
  app_cat,
  technology,
  sum(bandwidth) as bandwidth,
  sum(sessions) as sessions
from
  ###(select risk as d_risk, coalesce(nullifna(t1.`user`), nullifna(t1.`unauthuser`), ipstr
(t1.`srcip`)) as f_user, t2.name, t2.app_cat, t2.technology, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth,  count(*) as sessions from $log t1 inner join app_mdata t2 on
t1.appid=t2.id where $filter and risk>='4' and (logflag&1>0) group by f_user, t2.name, t2.app_cat,
t2.technology, risk order by bandwidth desc, sessions desc)### t group by d_risk, name, app_cat,
technology order by d_risk desc, sessions desc
```

Dataset Name	Description	Log Category
security-High-Risk-Application-By-Category	High risk application by category	traffic

```
select
  app_cat,
  count(distinct app) as total_num
from
  ###(select app_cat, app, count(*) as total_num from $log t1 inner join app_mdata t2 on
t1.appid=t2.id where $filter and risk>='4' and (logflag&1>0) group by app_cat, app order by total_
num desc)### t group by app_cat order by total_num desc
```

Dataset Name	Description	Log Category
security-Top10-Application-Categories-By-Bandwidth	Application Risk Application Usage by Category	traffic

```
select
  appcat,
  sum(bandwidth) as bandwidth
from
  ###(select appid, app, appcat, apprisk, user_src, hostname, dstip, sum(traffic_in) as traffic_
in, sum(traffic_out) as traffic_out, sum(bandwidth) as bandwidth, sum(sessions) as sessions from
###base(/*tag:rpt_base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid,
euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service,
appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions
from $log-traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by
timestamp, dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat, apprisk,
hostname order by bandwidth desc, sessions desc)base### t group by appid, app, appcat, apprisk,
user_src, hostname, dstip /*SkipSTART*/order by bandwidth desc, sessions desc/*SkipEND*/)### t
where $filter-drilldown and nullifna(appcat) is not null group by appcat having sum(bandwidth)>0
order by bandwidth desc
```

Dataset Name	Description	Log Category
Security-Category-Breakdown-By-Bandwidth	Category breakdown of all applications, sorted by bandwidth	traffic

```
select
  appcat,
  count(distinct app) as app_num,
  count(distinct user_src) as user_num,
  sum(bandwidth) as bandwidth,
  sum(sessions) as num_session
from
  ###(select app, appcat, user_src, sum(bandwidth) as bandwidth, sum(sessions) as sessions from
###base(/*tag:rpt_base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid,
euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service,
appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions
from $log-traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by
timestamp, dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat, apprisk,
hostname order by bandwidth desc, sessions desc)base### t where nullifna(appcat) is not null and
appcat not in ('Not.Scanned', 'unscanned', 'unknown') group by app, appcat, user_src order by
bandwidth desc)### t where $filter-drilldown group by appcat order by bandwidth desc
```

Dataset Name	Description	Log Category
security-Top25-Web-Applications-By-Bandwidth	Top Web Applications by Bandwidth	traffic

```
select
  risk as d_risk,
  t2.name,
  t2.app_cat,
  t2.technology,
  count(distinct f_user) as users,
  sum(bandwidth) as bandwidth,
  sum(num_session) as sessions
from
  ###(select appid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as f_user,
sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, count(*) as num_session from $log
where $filter and (logflag&1>0) and nullifna(app) is not null and service in ('tcp/80', 'tcp/443',
'HTTP', 'HTTPS', 'http', 'https') group by appid, f_user order by bandwidth desc)### t1 inner join
app_mdata t2 on t1.appid=t2.id group by d_risk, t2.name, t2.app_cat, t2.technology order by d_risk
desc, bandwidth desc
```

Dataset Name	Description	Log Category
Security-Top25-Web-Categories-Visited	Top 25 Web Categories Visited	traffic

```
select
  catdesc,
  count(distinct f_user) as user_num,
  sum(sessions) as sessions,
  sum(bandwidth) as bandwidth
from
  ###(select catdesc, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as f_
user, count(*) as sessions, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from
$log-traffic where $filter and catdesc is not null and (logflag&1>0) and (countweb>0 or ((logver
is null or logver<502000000) and (hostname is not null or utmevent in ('webfilter', 'banned-word',
'web-content', 'command-block', 'script-filter')))) group by f_user, catdesc order by sessions
desc)### t group by catdesc order by sessions desc
```

Dataset Name	Description	Log Category
security-Top25-Malware-Virus-Botnet-Spyware	Malware: viruses, Bots, Spyware/Adware	traffic

```
select
  virus_s as virus,
  (
    case when lower(appcat)=& #039;botnet' then 'Botnet C&C' else (case when virus_s like
```

```
'Riskware%' then 'Spyware' when virus_s like 'Adware%' then 'Adware' else 'Virus' end) end) as
malware_type, count(distinct dstip) as victims, count(distinct srcip) as source, sum(total_num) as
total_num from ((select virus_s, appcat, dstip, srcip, sum(total_num) as total_num  from ###
(select app as virus_s, appcat, dstip, srcip, count(*) as total_num from $log-traffic where
$filter and (logflag&1>0) and lower(appcat)='botnet' group by virus_s, appcat, dstip, srcip order
by total_num desc)### t group by virus_s, appcat, dstip, srcip) union all (select virus_s, appcat,
dstip, srcip, sum(total_num) as total_num  from ###(select unnest(string_to_array(virus, ',')) as
virus_s, appcat, dstip, srcip, count(*) as total_num from $log-traffic where $filter and
(logflag&1>0) and virus is not null group by virus_s, appcat, dstip, srcip order by total_num
desc)### t group by virus_s, appcat, dstip, srcip) union all (select virus_s, appcat, dstip,
srcip, sum(total_num) as total_num  from ###(select attack as virus_s, 'null' as appcat, dstip,
srcip, count(*) as total_num from $log-attack where $filter and (logflag&16>0) group by virus_s,
appcat, dstip, srcip order by total_num desc)### t group by virus_s, appcat, dstip, srcip)) t
group by virus, malware_type order by total_num desc
```

Dataset Name	Description	Log Category
security-Top10-Malware-Virus-Spyware	Malware: viruses, Spyware/Adware	virus

```
select
  virus,
  max(virusid_s) as virusid,
  malware_type,
  count(distinct victim) as victims,
  count(distinct source) as source,
  sum(total_num) as total_num
from
  ###(select virus, virusid_to_str(virusid, eventtype) as virusid_s, (CASE WHEN
direction='incoming' THEN dstip ELSE srcip END) as source, (CASE WHEN direction='incoming' THEN
srcip ELSE dstip END) as victim, (case when virus like 'Riskware%' then 'Spyware' when virus like
'Adware%' then 'Adware' else 'Virus' end)  as malware_type, count(*) as total_num from $log where
$filter and nullifna(virus) is not null group by virus, virusid_s, source, victim order by total_
num desc)### t group by virus, malware_type order by total_num desc
```

Dataset Name	Description	Log Category
security-Top10-Malware-Botnet	Malware: Botnet	appctrl

```
select
  app,
  appid,
  malware_type,
  count(distinct victim) as victims,
  count(distinct source) as source,
  sum(total_num) as total_num
from
  (
    (
```

```
      select
        app,
        appid,
        malware_type,
        source,
        victim,
        sum(total_num) as total_num
      from
        ###(select app, appid, cast('Botnet C&C' as char(32)) as malware_type,(CASE WHEN
direction='incoming' THEN dstip ELSE srcip END) as source, (CASE WHEN direction='incoming' THEN
srcip ELSE dstip END) as victim, count(*) as total_num from $log-app-ctrl where $filter and lower
(appcat)='botnet' and nullifna(app) is not null group by app, appid, malware_type, source, victim
order by total_num desc)### t group by app, appid, malware_type, source, victim) union all (select
app, appid, malware_type, source, victim, sum(total_num) as total_num from ###(select attack as
app, 0 as appid, cast('Botnet C&C' as char(32)) as malware_type, (CASE WHEN direction='incoming'
THEN dstip ELSE srcip END) as source, (CASE WHEN direction='incoming' THEN srcip ELSE dstip END)
as victim, count(*) as total_num from $log-attack where $filter and (logflag&16>0) group by
attack, appid, malware_type, source, victim order by total_num desc)### t group by app, appid,
malware_type, source, victim)) t group by app, appid, malware_type order by total_num desc
```

Dataset Name	Description	Log Category
security-Top10-Victims-of-Malware	Victims of Malware	virus

```
select
  coalesce(
    nullifna(`user`),
    nullifna(`unauthuser`),
    ipstr(`srcip`)
  ) as user_src,
  virus as malware,
  count(*) as total_num
from
  $log
where
  $filter
  and virus is not null
group by
  user_src,
  malware
order by
  total_num desc
```

Dataset Name	Description	Log Category
security-Top10-Victims-of-Phishing-Site	Victims of Phishing Site	webfilter

```
select
  coalesce(
    nullifna(`user`),
    nullifna(`unauthuser`),
    ipstr(`srcip`)
  ) as user_src,
  url as phishing_site,
  count(*) as total_num
from
  $log
where
  $filter
  and cat in (26, 61)
group by
  user_src,
  phishing_site
order by
  total_num desc
```

Dataset Name	Description	Log Category
security-Top25-Malicious-Phishing-Sites	Malicious Phishing Site	webfilter

```
select
  phishing_site,
  count(distinct dstip) as victims,
  count(distinct srcip) as source,
  sum(total) as total_num
from
  ###(select url as phishing_site, dstip, srcip, count(*) as total from $log where $filter and cat
in (26, 61) group by phishing_site, dstip, srcip order by total desc)### t group by phishing_site
order by total_num desc
```

Dataset Name	Description	Log Category
security-Application-Vulnerability	Application vulnerabilities discovered	attack

```
select
  attack,
  attackid,
  vuln_type,
  cve,
  severity_number,
  count(
    distinct (
      CASE WHEN direction =& #039;incoming' THEN srcip ELSE dstip END)) as victims, count(distinct
(CASE WHEN direction='incoming' THEN dstip ELSE srcip END)) as sources, sum(totalnum) as totalnum
from ###(select attack, attackid, (case when severity='critical' then 5 when severity='high' then
```

```
4 when severity='medium' then 3 when severity='low' then 2 when severity='info' then 1 else 0 end)
as severity_number, direction, dstip, srcip, count(*) as totalnum from $log where $filter and
nullifna(attack) is not null and severity is not null group by attack, attackid, severity,
direction, dstip, srcip order by totalnum desc)### t1 left join (select name, cve, vuln_type from
ips_mdata) t2 on t1.attack=t2.name group by attack, attackid, vuln_type, severity_number, cve
order by severity_number desc, totalnum desc
```

Dataset Name	Description	Log Category
security-Files-Analyzed-By-FortiCloud-Sandbox	Files analyzed by FortiCloud Sandbox	virus

```
select
  $day_of_week as dow,
  count(*) as total_num
from
  $log
where
  $filter
  and nullifna(filename) is not null
  and logid_to_int(logid)= 9233
group by
  dow
order by
  dow
```

Dataset Name	Description	Log Category
Security-Zero-Day-Detected-On-Network	Zero-day malware detected on the network	traffic

```
select
  virus_s,
  app,
  count(distinct dstip) as victims,
  count(distinct srcip) as source,
  sum(total_num) as total_num
from
  ###(select unnest(string_to_array(virus, ',')) as virus_s, app, dstip, srcip, count(*) as total_
num from $log where $filter and (logflag&1>0) and virus like '%PossibleThreat.SB%' group by virus_
s, dstip, srcip, app order by total_num desc)### t group by virus_s, app order by total_num desc
```

Dataset Name	Description	Log Category
security-Data-Loss-Incidents-By-Severity	Data loss incidents summary by severity	dlp

```
select
  initcap(severity : :text) as s_severity,
  count(*) as total_num
from
  ###(select itime, hostname,`from` as sender, `to` as receiver, profile, action, service,
subtype, srcip, dstip, severity, filename, direction, filesize, (case when severity='critical'
then 'Critical Data Exfiltration' else (case when coalesce(nullifna(`user`), ipstr(`srcip`)) is
not null then 'User Associated Data Loss' else NULL end) end) as data_loss from $log where $filter
/*SkipSTART*/order by itime desc/*SkipEND*/)### t where $filter-drilldown and severity is not null
group by s_severity order by total_num desc
```

Dataset Name	Description	Log Category
security-Data-Loss-Files-By-Service	Data Lass Files By Service	dlp

```
select
  filename,
  (
    case direction when & #039;incoming' then 'Download' when 'outgoing' then 'Upload' end) as
action, max(filesize) as filesize, service from  ###(select itime, hostname,`from` as sender, `to`
as receiver, profile, action, service, subtype, srcip, dstip, severity, filename, direction,
filesize, (case when severity='critical' then 'Critical Data Exfiltration' else (case when
coalesce(nullifna(`user`), ipstr(`srcip`)) is not null then 'User Associated Data Loss' else NULL
end) end) as data_loss from $log where $filter /*SkipSTART*/order by itime desc/*SkipEND*/)### t
where $filter-drilldown and filesize is not null group by filename, direction, service order by
filesize desc
```

Dataset Name	Description	Log Category
security-Endpoint-Security-Events-Summary	Endpoint Security Events summary	fct-traffic

```
select
  (
    case utmevent when & #039;antivirus' then 'Malware incidents' when 'webfilter' then
'Malicious/phishing websites' when 'appfirewall' then 'Risk applications' when 'dlp' then 'Data
loss incidents' when 'netscan' then 'Vulnerability detected' else 'Others' end) as events, count
(*) as total_num from $log where $filter and utmevent is not null group by events order by total_
num desc
```

Dataset Name	Description	Log Category
security-Top-Endpoing-Running-High-Risk-Application	Endpoints Running High Risk Application	fct-traffic

```
select
  coalesce(
```

```
    nullifna(`user`),
    ipstr(`srcip`),
    & #039;Unknown') as f_user, coalesce(nullifna(hostname), 'Unknown') as host_name, threat as
app, t2.app_cat as appcat, risk as d_risk from $log t1 inner join app_mdata t2 on
t1.threat=t2.name where $filter and utmevent='appfirewall' and risk>='4' group by f_user, host_
name, t1.threat, t2.app_cat, t2.risk order by risk desc
```

Dataset Name	Description	Log Category
security-Top-Endpoints-Infected-With-Malware	Endpoints Infected With Malware	fct-event

```
select
  coalesce(
    nullifna(`user`),
    ipstr(`deviceip`),
    & #039;Unknown') as f_user, coalesce(nullifna(hostname), 'Unknown') as host_name, virus, file,
count(*) as total_num from $log where $filter and subtype='av' and virus is not null group by f_
user, host_name, virus, file order by total_num desc
```

Dataset Name	Description	Log Category
security-Top-Endpoints-With-Web-Violateions	Endpoints With Web Violations	fct-traffic

```
select
  f_user,
  host_name,
  remotename,
  sum(total_num) as total_num
from
  ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as f_user, coalesce(nullifna(hostname),
'Unknown') as host_name, remotename, count(*) as total_num from $log where $filter and
utmevent='webfilter' and remotename is not null and utmaction='blocked' group by f_user, host_
name, remotename order by total_num desc)### t group by f_user, host_name, remotename order by
total_num desc
```

Dataset Name	Description	Log Category
security-Top-Endpoints-With-Data-Loss-Incidents	Endpoints With Data Loss Incidents	fct-event

```
select
  f_user,
  host_name,
  msg,
  sum(total_num) as total_num
from
```

```
  ###(select coalesce(nullifna(`user`), ipstr(`deviceip`), 'Unknown') as f_user, coalesce(nullifna
(hostname), 'Unknown') as host_name, msg, count(*) as total_num from $log where $filter and
subtype='dlp' group by f_user, host_name, msg order by total_num desc)### t group by f_user, host_
name, msg order by total_num desc
```

Dataset Name	Description	Log Category
content-Count-Total-SCCP-Call-Registrations-by-Hour-of-Day	Content count total SCCP call registrations by hour of day	content

```
select
  hourstamp,
  count(totalnum) as totalnum
from
  ###(select $hour_of_day as hourstamp, proto, kind, status, sum(duration) as sccp_usage, count(*)
as totalnum from $log-content where $filter group by hourstamp, proto, kind, status order by
totalnum desc)### t where proto='sccp' and kind='register' group by hourstamp order by hourstamp
```

Dataset Name	Description	Log Category
content-Count-Total-SCCP-Calls-Duration-by-Hour-of-Day	Content count total SCCP calls duration by hour of day	content

```
select
  hourstamp,
  sum(sccp_usage) as sccp_usage
from
  ###(select $hour_of_day as hourstamp, proto, kind, status, sum(duration) as sccp_usage, count(*)
as totalnum from $log-content where $filter group by hourstamp, proto, kind, status order by
totalnum desc)### t where proto='sccp' and kind='call-info' and status='end' group by hourstamp
order by hourstamp
```

Dataset Name	Description	Log Category
content-Count-Total-SCCP-Calls-per-Status	Content count total SCCP calls per status	content

```
select
  status,
  count(totalnum) as totalnum
from
  ###(select $hour_of_day as hourstamp, proto, kind, status, sum(duration) as sccp_usage, count(*)
as totalnum from $log-content where $filter group by hourstamp, proto, kind, status order by
totalnum desc)### t where proto='sccp' and kind='call-info' group by status order by totalnum desc
```

Dataset Name	Description	Log Category
content-Count-Total-SIP-Call-Registrations-by-Hour-of-Day	Content count total SIP call registrations by hour of day	content

```
select
  hourstamp,
  count(totalnum) as totalnum
from
  ###(select $hour_of_day as hourstamp, proto, kind, status, sum(duration) as sccp_usage, count(*)
as totalnum from $log-content where $filter group by hourstamp, proto, kind, status order by
totalnum desc)### t where proto='sip' and kind='register' group by hourstamp order by hourstamp
```

Dataset Name	Description	Log Category
content-Count-Total-SIP-Calls-per-Status	Content count total SIP calls per status	content

```
select
  status,
  count(totalnum) as totalnum
from
  ###(select $hour_of_day as hourstamp, proto, kind, status, sum(duration) as sccp_usage, count(*)
as totalnum from $log-content where $filter group by hourstamp, proto, kind, status order by
totalnum desc)### t where proto='sip' and kind='call' group by status order by totalnum desc
```

Dataset Name	Description	Log Category
content-Dist-Total-SIP-Calls-by-Duration	Content dist total SIP calls by duration	content

```
select
  (
    case when duration<60 then & #039;LESS_ONE_MIN' when duration < 600 then 'LESS_TEN_MIN' when
duration < 3600 then 'LESS_ONE_HOUR' when duration >= 3600 then 'MORE_ONE_HOUR' else 'unknown'
end) as f_duration, count(*) as totalnum from $log where $filter and proto='sip' and kind='call'
and status='end' group by f_duration order by totalnum desc
```

Dataset Name	Description	Log Category
Botnet-Activity-By-Sources	Botnet activity by sources	traffic

```
select
  app,
  user_src,
  sum(events) as events
```

```
from
  (
    (
      select
        app,
        user_src,
        sum(totalnum) as events
      from
        ###(select app, appcat, apprisk, srcip, dstip, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, count(*) as totalnum from $log-traffic where $filter
and (logflag&1>0) and appcat='Botnet' and nullifna(app) is not null group by app, appcat, apprisk,
srcip, dstip, user_src order by totalnum desc)### t group by app, user_src order by events desc)
union all (select attack, user_src, sum(totalnum) as events from ###(select attack, coalesce
(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, $flex_timestamp as
timestamp, hostname, severity, crlevel, eventtype, service,  (CASE WHEN direction='incoming' THEN
srcip ELSE dstip END) as victim, (CASE WHEN direction='incoming' THEN dstip ELSE srcip END) as
source, count(*) as totalnum from $log-attack where $filter and (logflag&16>0) group by attack,
user_src, timestamp, hostname, severity, crlevel, eventtype, service, victim, source order by
timestamp desc)### t group by attack, user_src order by events desc)) t group by app, user_src
order by events desc
```

Dataset Name	Description	Log Category
Botnet-Infected-Hosts	Botnet infected hosts	traffic

```
select
  user_src,
  devtype_new,
  host_mac,
  sum(events) as events
from
  (
    (
      select
        user_src,
        devtype_new,
        host_mac,
        sum(events) as events
      from
        ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
get_devtype(srcswversion, osname, devtype) as devtype_new, coalesce(srcname, srcmac) as host_mac,
count(*) as events from $log-traffic where $filter and (logflag&1>0) and appcat='Botnet' group by
user_src, devtype_new, host_mac order by events desc)### t group by user_src, devtype_new, host_
mac) union all (select user_src, devtype_new, host_mac, sum(events) as events from ###(select
coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, 'Unknown' as
devtype_new, hostname as host_mac, count(*) as events from $log-attack where $filter and
(logflag&16>0) group by user_src, devtype_new, host_mac order by events desc)### t group by user_
src, devtype_new, host_mac)) t group by user_src, devtype_new, host_mac order by events desc
```

Dataset Name	Description	Log Category
Detected-Botnet	Detected botnet	traffic

```
select
  app,
  sum(events) as events
from
  (
    (
      select
        app,
        sum(totalnum) as events
      from
        ###(select app, appcat, apprisk, srcip, dstip, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, count(*) as totalnum from $log-traffic where $filter
and (logflag&1>0) and appcat='Botnet' and nullifna(app) is not null group by app, appcat, apprisk,
srcip, dstip, user_src order by totalnum desc)### t group by app order by events desc) union all
(select attack, sum(totalnum) as events from ###(select attack, coalesce(nullifna(`user`),
nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, $flex_timestamp as timestamp, hostname,
severity, crlevel, eventtype, service,  (CASE WHEN direction='incoming' THEN srcip ELSE dstip END)
as victim, (CASE WHEN direction='incoming' THEN dstip ELSE srcip END) as source, count(*) as
totalnum from $log-attack where $filter and (logflag&16>0) group by attack, user_src, timestamp,
hostname, severity, crlevel, eventtype, service, victim, source order by timestamp desc)### t
group by attack order by events desc)) t group by app order by events desc
```

Dataset Name	Description	Log Category
Botnet-Sources	Botnet sources	traffic

```
select
  dstip,
  domain,
  sum(events) as events
from
  (
    (
      select
        dstip,
        domain,
        sum(events) as events
      from
        ###(select dstip, root_domain(hostname) as domain, count(*) as events from $log-traffic
where $filter and (logflag&1>0) and appcat='Botnet' and dstip is not null group by dstip, domain
order by events desc)### t group by dstip, domain) union all (select victim as dstip, root_domain
(hostname) as domain, sum(totalnum) as events from ###(select attack, coalesce(nullifna(`user`),
nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, $flex_timestamp as timestamp, hostname,
severity, crlevel, eventtype, service,  (CASE WHEN direction='incoming' THEN srcip ELSE dstip END)
as victim, (CASE WHEN direction='incoming' THEN dstip ELSE srcip END) as source, count(*) as
totalnum from $log-attack where $filter and (logflag&16>0) group by attack, user_src, timestamp,
```

```
hostname, severity, crlevel, eventtype, service, victim, source order by timestamp desc)### t
group by dstip, domain)) t group by dstip, domain order by events desc
```

Dataset Name	Description	Log Category
Botnet-Victims	Botnet victims	traffic

```
select
  user_src,
  sum(events) as events
from
  (
    (
      select
        user_src,
        sum(totalnum) as events
      from
        ###(select app, appcat, apprisk, srcip, dstip, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, count(*) as totalnum from $log-traffic where $filter
and (logflag&1>0) and appcat='Botnet' and nullifna(app) is not null group by app, appcat, apprisk,
srcip, dstip, user_src order by totalnum desc)### t group by user_src) union all (select user_src,
sum(totalnum) as events from ###(select attack, coalesce(nullifna(`user`), nullifna(`unauthuser`),
ipstr(`srcip`)) as user_src, $flex_timestamp as timestamp, hostname, severity, crlevel, eventtype,
service,  (CASE WHEN direction='incoming' THEN srcip ELSE dstip END) as victim, (CASE WHEN
direction='incoming' THEN dstip ELSE srcip END) as source, count(*) as totalnum from $log-attack
where $filter and (logflag&16>0) group by attack, user_src, timestamp, hostname, severity,
crlevel, eventtype, service, victim, source order by timestamp desc)### t group by user_src)) t
group by user_src order by events desc
```

Dataset Name	Description	Log Category
Botnet-Timeline	Botnet timeline	traffic

```
select
  $flex_datetime(timestamp) as hodex,
  sum(events) as events
from
  (
    (
      select
        timestamp,
        sum(events) as events
      from
        ###(select $flex_timestamp as timestamp, count(*) as events from $log-traffic where
$filter and (logflag&1>0) and appcat='Botnet' group by timestamp order by timestamp desc)### t
group by timestamp) union all (select timestamp, sum(events) as events from ###(select $flex_
timestamp as timestamp, count(*) as events from $log-dns where $filter and (botnetdomain is not
null or botnetip is not null) group by timestamp order by timestamp)### t group by timestamp)
union all (select timestamp, sum(events) as events from ###(select $flex_timestamp as timestamp,
```

```
count(*) as events from $log-attack where $filter and (logflag&16>0) group by timestamp order by
timestamp)### t group by timestamp)) t group by hodex order by hodex
```

Dataset Name	Description	Log Category
Application-Session-History	Application session history	traffic

```
select
  $flex_timescale(timestamp) as hodex,
  sum(counter) as counter
from
  ###(select $flex_timestamp as timestamp, count(*) as counter from $log where $filter and
(logflag&1>0) group by timestamp order by timestamp desc)### t group by hodex order by hodex
```

Dataset Name	Description	Log Category
Application-Usage-List	Detailed application usage	traffic

```
select
  appid,
  app,
  appcat,
  (
    case when (
      utmaction in (
        & #039;block', 'blocked') or action='deny') then 'Blocked' else 'Allowed' end) as
custaction, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth,count(*) as num_session
from $log where $filter and (logflag&1>0) and nullifna(app) is not null and policyid != 0 group by
appid,app, appcat,custaction order by bandwidth desc
```

Dataset Name	Description	Log Category
PCI-DSS-Compliance-Summary	PCI DSS Compliance Summary	event

```
select
  status,
  num_reason as requirements,
  cast(
    num_reason * 100.0 /(
      sum(num_reason) over()
    ) as decimal(18, 2)
  ) as percent
from
  (
    select
      (
        case when fail_count>0 then & #039;Non-Compliant' else 'Compliant' end) as status, count
(distinct reason) as num_reason from (select ftnt_pci_id, (sum(fail_count) over (partition by
```

```
ftnt_pci_id)) as fail_count, reason from ###(select ftnt_pci_id, (case when result='fail' then 1
else 0 end) as fail_count, reason, count(*) as total_num from $log t1 inner join pci_dss_mdata t2
on t1.reason=t2.ftnt_id where $filter and subtype='compliance-check' group by ftnt_pci_id, result,
reason order by total_num desc)### t) t group by status) t order by status
```

Dataset Name	Description	Log Category
PCI-DSS-Non-Compliant-Requirements-By-Severity	PCI DSS Non-Compliant Requirements by Severity	event

```
with query as (
  select
    *
  from
    (
      select
        ftnt_pci_id,
        severity,
        (
          sum(fail_count) over (partition by ftnt_pci_id)
        ) as fail_count,
        reason
      from
        ###(select ftnt_pci_id, t2.severity, (case when result='fail' then 1 else 0 end) as fail_
count, reason from $log t1 inner join pci_dss_mdata t2 on t1.reason=t2.ftnt_id where $filter and
subtype='compliance-check' group by ftnt_pci_id, t2.severity, result, reason order by fail_count
desc)### t) t where fail_count>0) select t.severity, count(distinct t.reason) as requirements from
(select distinct on (1) reason, severity from query order by reason, (case lower(severity) when
'high' then 4 when 'critical' then 3 when 'medium' then 2 when 'low' then 1 else 0 end) desc) t
group by t.severity order by requirements desc
```

Dataset Name	Description	Log Category
PCI-DSS-Compliant-Requirements-By-Severity	PCI DSS Compliant Requirements by Severity	event

```
with query as (
  select
    *
  from
    (
      select
        ftnt_pci_id,
        severity,
        (
          sum(fail_count) over (partition by ftnt_pci_id)
        ) as fail_count,
        reason
      from
```

```
        ###(select ftnt_pci_id, t2.severity, (case when result='fail' then 1 else 0 end) as fail_
count, reason from $log t1 inner join pci_dss_mdata t2 on t1.reason=t2.ftnt_id where $filter and
subtype='compliance-check' group by ftnt_pci_id, t2.severity, result, reason order by fail_count
desc)### t) t where fail_count=0) select t.severity, count(distinct t.reason) as requirements from
(select distinct on (1) reason, severity from query order by reason, (case lower(severity) when
'high' then 4 when 'critical' then 3 when 'medium' then 2 when 'low' then 1 else 0 end) desc) t
group by t.severity order by requirements desc
```

Dataset Name	Description	Log Category
PCI-DSS-Fortinet-Security-Best-Practice-Summary	PCI DSS Fortinet Security Best Practice Summary	event

```
select
  status,
  num_reason as practices,
  cast(
    num_reason * 100.0 /(
      sum(num_reason) over()
    ) as decimal(18, 2)
  ) as percent
from
  (
    select
      (
        case when result =& #039;fail' then 'Failed' else 'Passed' end) as status, count(distinct
reason) as num_reason from ###(select result, reason, count(*) as total_num from $log where
$filter and subtype='compliance-check' and result in ('fail','pass') group by result, reason order
by total_num desc)### t group by status) t order by status desc
```

Dataset Name	Description	Log Category
PCI-DSS-Failed-Fortinet-Security-Best-Practices-By-Severity	PCI DSS Failed Fortinet Security Best Practices by Severity	event

```
select
  status,
  num_reason as practices,
  cast(
    num_reason * 100.0 /(
      sum(num_reason) over()
    ) as decimal(18, 2)
  ) as percent
from
  (
    select
      initcap(status) as status,
      count(distinct reason) as num_reason
    from
```

```
      ###(select status, reason, result, count(*) as total_num from $log where $filter and
subtype='compliance-check' group by status, reason, result order by total_num desc)### t where
result='fail' group by status) t order by status
```

Dataset Name	Description	Log Category
PCI-DSS-Passed-Fortinet-Security-Best-Practices-By-Severity	PCI DSS Passed Fortinet Security Best Practices by Severity	event

```
select
  status,
  num_reason as practices,
  cast(
    num_reason * 100.0 /(
      sum(num_reason) over()
    ) as decimal(18, 2)
  ) as percent
from
  (
    select
      initcap(status) as status,
      count(distinct reason) as num_reason
    from
      ###(select status, reason, result, count(*) as total_num from $log where $filter and
subtype='compliance-check' group by status, reason, result order by total_num desc)### t where
result='pass' group by status) t order by status
```

Dataset Name	Description	Log Category
PCI-DSS-Requirements-Compliance-Details	PCI DSS Requirements Compliance Details	event

```
select
  ftnt_pci_id,
  left(
    string_agg(
      distinct ftnt_id,
      & #039;,'), 120) as practice, (case when sum(fail_count)>0 then 'Non-Compliant' else
'Compliant' end) as compliance, pci_requirement from ###(select ftnt_pci_id, ftnt_id, (case when
result='fail' then 1 else 0 end) as fail_count, pci_requirement, count(*) as total_num from $log
t1 inner join pci_dss_mdata t2 on t1.reason=t2.ftnt_id where $filter and subtype='compliance-
check' group by ftnt_pci_id, ftnt_id, result, pci_requirement order by total_num desc)### t group
by ftnt_pci_id, pci_requirement order by ftnt_pci_id
```

Dataset Name	Description	Log Category
PCI-DSS-Fortinet-Security-Best-Practice-Details	PCI DSS Fortinet Security Best Practice Details	event

```
select
  reason as ftnt_id,
  msg,
  initcap(status) as status,
  module
from
  $log
where
  $filter
  and subtype =& #039;compliance-check' group by reason, status, module, msg order by ftnt_id
```

Dataset Name	Description	Log Category
DLP-Email-Activity-Details	Email DLP Violations Summary	dlp

```
select
  from_itime(itime) as timestamp,
  sender,
  receiver,
  regexp_replace(
    filename,
    & #039;.*/', '') as filename, filesize, profile, action, direction from ###(select itime,
hostname,`from` as sender, `to` as receiver, profile, action, service, subtype, srcip, dstip,
severity, filename, direction, filesize, (case when severity='critical' then 'Critical Data
Exfiltration' else (case when coalesce(nullifna(`user`), ipstr(`srcip`)) is not null then 'User
Associated Data Loss' else NULL end) end) as data_loss from $log where $filter /*SkipSTART*/order
by itime desc/*SkipEND*/)### t where $filter-drilldown and (service in ('smtp', 'SMTP', 'tcp/25',
'tcp/587', 'smtps', 'SMTPS', 'tcp/465') or service in ('pop3', 'POP3', 'tcp/110', 'imap', 'IMAP',
'tcp/143', 'imaps', 'IMAPS', 'tcp/993', 'pop3s', 'POP3S', 'tcp/995')) order by timestamp desc
```

Dataset Name	Description	Log Category
Email-DLP-Chart	Email DLP Activity Summary	dlp

```
select
  profile,
  count(*) as total_num
from
  ###(select itime, hostname,`from` as sender, `to` as receiver, profile, action, service,
subtype, srcip, dstip, severity, filename, direction, filesize, (case when severity='critical'
then 'Critical Data Exfiltration' else (case when coalesce(nullifna(`user`), ipstr(`srcip`)) is
not null then 'User Associated Data Loss' else NULL end) end) as data_loss from $log where $filter
/*SkipSTART*/order by itime desc/*SkipEND*/)### t where $filter-drilldown and (service in ('smtp',
'SMTP', 'tcp/25', 'tcp/587', 'smtps', 'SMTPS', 'tcp/465') or service in ('pop3', 'POP3',
'tcp/110', 'imap', 'IMAP', 'tcp/143', 'imaps', 'IMAPS', 'tcp/993', 'pop3s', 'POP3S', 'tcp/995'))
group by profile order by total_num desc
```

Dataset Name	Description	Log Category
DLP-Web-Activity-Details	Web DLP Violations Summary	dlp

```
select
  from_itime(itime) as timestamp,
  srcip,
  dstip,
  hostname,
  profile,
  filename,
  filesize,
  action,
  direction
from
  ###(select itime, hostname,`from` as sender, `to` as receiver, profile, action, service,
subtype, srcip, dstip, severity, filename, direction, filesize, (case when severity='critical'
then 'Critical Data Exfiltration' else (case when coalesce(nullifna(`user`), ipstr(`srcip`)) is
not null then 'User Associated Data Loss' else NULL end) end) as data_loss from $log where $filter
/*SkipSTART*/order by itime desc/*SkipEND*/)### t where $filter-drilldown and lower(service) in
('http', 'https') order by timestamp desc
```

Dataset Name	Description	Log Category
Web-DLP-Chart	Web DLP Activity Summary	dlp

```
select
  profile,
  count(*) as total_num
from
  ###(select itime, hostname,`from` as sender, `to` as receiver, profile, action, service,
subtype, srcip, dstip, severity, filename, direction, filesize, (case when severity='critical'
then 'Critical Data Exfiltration' else (case when coalesce(nullifna(`user`), ipstr(`srcip`)) is
not null then 'User Associated Data Loss' else NULL end) end) as data_loss from $log where $filter
/*SkipSTART*/order by itime desc/*SkipEND*/)### t where $filter-drilldown and lower(service) in
('http', 'https') group by profile order by total_num desc
```

Dataset Name	Description	Log Category
DLP-FTP-Activity-Details	Web DLP Violations Summary	dlp

```
select
  from_itime(itime) as timestamp,
  srcip,
  dstip,
  filename,
  profile,
  filesize,
```

```
  action,
  direction
from
  ###(select itime, hostname,`from` as sender, `to` as receiver, profile, action, service,
subtype, srcip, dstip, severity, filename, direction, filesize, (case when severity='critical'
then 'Critical Data Exfiltration' else (case when coalesce(nullifna(`user`), ipstr(`srcip`)) is
not null then 'User Associated Data Loss' else NULL end) end) as data_loss from $log where $filter
/*SkipSTART*/order by itime desc/*SkipEND*/)### t where $filter-drilldown and lower(service) in
('ftp', 'ftps') order by timestamp desc
```

Dataset Name	Description	Log Category
FTP-DLP-Chart	FTP DLP Activity Summary	dlp

```
select
  profile,
  count(*) as total_num
from
  ###(select itime, hostname,`from` as sender, `to` as receiver, profile, action, service,
subtype, srcip, dstip, severity, filename, direction, filesize, (case when severity='critical'
then 'Critical Data Exfiltration' else (case when coalesce(nullifna(`user`), ipstr(`srcip`)) is
not null then 'User Associated Data Loss' else NULL end) end) as data_loss from $log where $filter
/*SkipSTART*/order by itime desc/*SkipEND*/)### t where $filter-drilldown and lower(service) in
('ftp', 'ftps') group by profile order by total_num desc
```

Dataset Name	Description	Log Category
top-users-by-browsetime	Top Users by website browsetime	traffic

```
select
  user_src,
  domain,
  ebtr_value(
    ebtr_agg_flat(browsetime),
    null,
    $timespan
  ) as browsetime
from
  ###(select user_src, domain, ebtr_agg_flat(browsetime) as browsetime from (select coalesce
(nullifna(`user`), ipstr(`srcip`)) as user_src, coalesce(nullifna(hostname), ipstr(`dstip`)) as
domain, ebtr_agg_flat($browse_time) as browsetime from $log where $filter and $browse_time is not
null group by user_src, domain) t group by user_src, domain order by ebtr_value(ebtr_agg_flat
(browsetime), null, null) desc)### t group by user_src, domain order by browsetime desc
```

Dataset Name	Description	Log Category
wifi-usage-by-hour-authenticated	Wifi Usage by Hour - Authenticated	event

```
select
  hod,
  count(distinct stamac) as totalnum
from
  ###(select $HOUR_OF_DAY as hod, stamac, count(*) as total_num from $log where $filter and
subtype='wireless' and action='client-authentication' group by hod, stamac order by total_num
desc)### t group by hod order by hod
```

Dataset Name	Description	Log Category
wifi-usage-authenticated-timeline	Wifi Usage Timeline - Authenticated	event

```
select
  $flex_timescale(timestamp) as hodex,
  count(distinct stamac) as totalnum
from
  ###(select $flex_timestamp as timestamp, stamac from $log where $filter and subtype='wireless'
and action='client-authentication' group by timestamp, stamac order by timestamp desc)### t group
by hodex order by hodex
```

Dataset Name	Description	Log Category
app-top-user-by-bandwidth	Top 10 Applications Bandwidth by User Drilldown	traffic

```
select
  app,
  coalesce(
    nullifna(`user`),
    nullifna(`unauthuser`),
    ipstr(`srcip`)
  ) as user_src,
  sum(
    coalesce(`sentbyte`, 0)+ coalesce(`rcvdbyte`, 0)
  ) as bandwidth
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
  and nullifna(app) is not null
group by
  app,
  user_src
order by
  bandwidth desc
```

Dataset Name	Description	Log Category
app-top-user-by-session	Top 10 Application Sessions by User Drilldown	traffic

```
select
  app,
  coalesce(
    nullifna(`user`),
    nullifna(`unauthuser`),
    ipstr(`srcip`)
  ) as user_src,
  count(*) as sessions
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
  and nullifna(app) is not null
group by
  app,
  user_src
order by
  sessions desc
```

Dataset Name	Description	Log Category
traffic-Interface-Bandwidth-Usage	Interface Bandwidth Usage	traffic

```
with qry as (
  select
    dom as dom_s,
    devid as devid_s,
    vd as vd_s,
    srcintf,
    dstintf,
    total_sent,
    total_rcvd
  from
    ###(select $DAY_OF_MONTH as dom, devid, vd, srcintf, dstintf, sum(coalesce(sentbyte, 0)) as
total_sent, sum(coalesce(rcvdbyte, 0)) as total_rcvd, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte,
0)) as total from $log where $filter and (logflag&1>0) and nullifna(srcintf) is not null and
nullifna(dstintf) is not null group by dom, devid, vd, srcintf, dstintf having sum(coalesce
(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by total desc)### t) select dom, unnest(array
['download', 'upload']) as type, unnest(array[sum(download), sum(upload)]) as bandwidth from
(select coalesce(t1.dom_s, t2.dom_s) as dom, coalesce(t1.devid_s, t2.devid_s) as devid, coalesce
(t1.vd_s, t2.vd_s) as vd, coalesce(t1.srcintf, t2.dstintf) as intf, sum(coalesce(t1.total_sent,
0)+coalesce(t2.total_rcvd, 0)) as download, sum(coalesce(t2.total_sent, 0)+coalesce(t1.total_rcvd,
```

```
0)) as upload from qry t1 full join qry t2 on t1.dom_s=t2.dom_s and t1.srcintf=t2.dstintf group by
dom, devid, vd, intf) t where $filter-drilldown group by dom order by dom
```

Dataset Name	Description	Log Category
CTAP-Threat-Detected-Timeline	Threat Detected Timeline	app-ctrl

```
select
  $flex_timestamp(timestamp) as hodex,
  type,
  sum(totalnum) as totalnum
from
  (
    (
      select
        timestamp,
        & #039;IPS Attacks' as type, sum(total_num) as totalnum from ###(select $flex_timestamp as
timestamp, attack, (case when (logflag&16>0) then 1 else 0 end) as botnet_flag, (CASE WHEN
direction='incoming' THEN dstip ELSE srcip END) as source, (CASE WHEN direction='incoming' THEN
srcip ELSE dstip END) as victim, service, count(*) as total_num from $log-attack where $filter and
nullifna(attack) is not null group by timestamp, attack, botnet_flag, source, victim, service
order by total_num desc)### t group by timestamp, type order by totalnum desc) union all (select
timestamp, 'Malware/Botnets' as type, count(distinct malware) as totalnum from ((select timestamp,
app as malware from ###(select $flex_timestamp as timestamp, app, appcat, appid, apprisk, (CASE
WHEN direction='incoming' THEN dstip ELSE srcip END) as source, (CASE WHEN direction='incoming'
THEN srcip ELSE dstip END) as victim, count(*) as total_num from $log-app-ctrl where $filter and
nullifna(app) is not null group by timestamp, app, appcat, appid, apprisk, source, victim order by
total_num desc)### t where lower(appcat)='botnet') union all (select timestamp, virus as malware
from ###(select $flex_timestamp as timestamp, virus, (CASE WHEN direction='incoming' THEN dstip
ELSE srcip END) as source, (CASE WHEN direction='incoming' THEN srcip ELSE dstip END) as victim,
service, count(*) as total_num from $log-virus where $filter and nullifna(virus) is not null group
by timestamp, virus, source, victim, service order by total_num desc)### t) union all (select
timestamp, attack as malware from ###(select $flex_timestamp as timestamp, attack, (case when
(logflag&16>0) then 1 else 0 end) as botnet_flag, (CASE WHEN direction='incoming' THEN dstip ELSE
srcip END) as source, (CASE WHEN direction='incoming' THEN srcip ELSE dstip END) as victim,
service, count(*) as total_num from $log-attack where $filter and nullifna(attack) is not null
group by timestamp, attack, botnet_flag, source, victim, service order by total_num desc)### t
where botnet_flag>0)) t group by timestamp, type order by totalnum desc) union all (select
timestamp, 'High-Risk Applications' as type, count(distinct app) as totalnum from ###(select
$flex_timestamp as timestamp, app, appcat, appid, apprisk, (CASE WHEN direction='incoming' THEN
dstip ELSE srcip END) as source, (CASE WHEN direction='incoming' THEN srcip ELSE dstip END) as
victim, count(*) as total_num from $log-app-ctrl where $filter and nullifna(app) is not null group
by timestamp, app, appcat, appid, apprisk, source, victim order by total_num desc)### t where
apprisk in ('critical', 'high') group by timestamp, type order by totalnum desc) union all (select
timestamp, 'Malicious Websites' as type, count(distinct hostname) as totalnum from ###(select
$flex_timestamp as timestamp, hostname, count(*) as total_num from $log-webfilter where $filter
and hostname is not null and catdesc='Malicious Websites' group by timestamp, hostname order by
total_num desc)### t group by timestamp, type order by totalnum desc)) t group by hodex, type
order by hodex
```

Dataset Name	Description	Log Category
ctap-SB-Files-Needing-Inspection-vs-Others	Files Needing Inspection vs Others	virus

```
select
  (
    case when suffix in (
      &
#039;bat','cmd','exe','jar','msi','vbs','7z','zip','gzip','lzw','tar','rar','cab','doc','docx','xl
s','xlsx','ppt','pptx','pdf','swf','lnk','js') then 'Higher Risk File Types' else 'Excluded Files'
end) as files, sum(total_num) as total_num from ###(select filename, file_name_ext(filename) as
suffix, fsaverdict, (CASE WHEN direction='incoming' THEN dstip ELSE srcip END) as source, (CASE
WHEN direction='incoming' THEN srcip ELSE dstip END) as victim, service, count(*) as total_num
from $log where $filter and dtype='fortisandbox' and nullifna(filename) is not null group by
filename, suffix, fsaverdict, source, victim, service order by total_num desc)### t group by files
order by total_num desc
```

Dataset Name	Description	Log Category
ctap-SB-Files-Needing-Inspection-vs-Others-Donut	Files Needing Inspection vs Others	virus

```
select
  (
    case when suffix in (
      &
#039;bat','cmd','exe','jar','msi','vbs','7z','zip','gzip','lzw','tar','rar','cab','doc','docx','xl
s','xlsx','ppt','pptx','pdf','swf','lnk','js') then 'Higher Risk File Types' else 'Excluded Files'
end) as files, sum(total_num) as total_num from ###(select filename, file_name_ext(filename) as
suffix, fsaverdict, (CASE WHEN direction='incoming' THEN dstip ELSE srcip END) as source, (CASE
WHEN direction='incoming' THEN srcip ELSE dstip END) as victim, service, count(*) as total_num
from $log where $filter and dtype='fortisandbox' and nullifna(filename) is not null group by
filename, suffix, fsaverdict, source, victim, service order by total_num desc)### t group by files
order by total_num desc
```

Dataset Name	Description	Log Category
ctap-SB-Breakdown-of-File-Types	Breakdown of File Types	virus

```
select
  (
    case when suffix in (
      & #039;exe','msi','upx','vbs','bat','cmd','dll','ps1','jar') then 'Executable Files' when
suffix in ('pdf') then 'Adobe PDF' when suffix in ('swf') then 'Adobe Flash' when suffix in
('doc','docx','rtf','dotx','docm','dotm','dot') then 'Microsoft Word' when suffix in
('xls','xlsx','xltx','xlsm','xlsb','xlam','xlt') then 'Microsoft Excel' when suffix in
('ppsx','ppt','pptx','potx','sldx','pptm','ppsm','potm','ppam','sldm','pps','pot') then 'Microsoft
```

```
PowerPoint' when suffix in ('msg') then 'Microsoft Outlook' when suffix in
('htm','js','url','lnk') then 'Web Files' when suffix in
('cab','tgz','z','7z','tar','lzh','kgb','rar','zip','gz','xz','bz2') then 'Archive Files' when
suffix in ('apk') then 'Android Files' else 'Others' end) as filetype, sum(total_num) as total_num
from ###(select filename, file_name_ext(filename) as suffix, fsaverdict, (CASE WHEN
direction='incoming' THEN dstip ELSE srcip END) as source, (CASE WHEN direction='incoming' THEN
srcip ELSE dstip END) as victim, service, count(*) as total_num from $log where $filter and
dtype='fortisandbox' and nullifna(filename) is not null group by filename, suffix, fsaverdict,
source, victim, service order by total_num desc)### t group by filetype order by total_num desc
```

Dataset Name	Description	Log Category
ctap-SB-Top-Sandbox-Malicious-Exes		virus

```
select
  (
    case fsaverdict when & #039;malicious' then 5 when 'high risk' then 4 when 'medium risk' then
3 when 'low risk' then 2 else 1 end) as risk, filename, service, count(*) as total_num from ###
(select filename, file_name_ext(filename) as suffix, fsaverdict, (CASE WHEN direction='incoming'
THEN dstip ELSE srcip END) as source, (CASE WHEN direction='incoming' THEN srcip ELSE dstip END)
as victim, service, count(*) as total_num from $log where $filter and dtype='fortisandbox' and
nullifna(filename) is not null group by filename, suffix, fsaverdict, source, victim, service
order by total_num desc)### t where suffix='exe' and fsaverdict not in ('clean','submission
failed') group by filename, risk, service order by risk desc, total_num desc, filename
```

Dataset Name	Description	Log Category
ctap-SB-Sources-of-Sandbox-Discovered-Malware	Sources of Sandbox Discovered Malware	virus

```
select
  source,
  sum(total_num) as total_num
from
  ###(select filename, file_name_ext(filename) as suffix, fsaverdict, (CASE WHEN
direction='incoming' THEN dstip ELSE srcip END) as source, (CASE WHEN direction='incoming' THEN
srcip ELSE dstip END) as victim, service, count(*) as total_num from $log where $filter and
dtype='fortisandbox' and nullifna(filename) is not null group by filename, suffix, fsaverdict,
source, victim, service order by total_num desc)### t where fsaverdict not in ('clean','submission
failed') group by source order by total_num desc
```

Dataset Name	Description	Log Category
ctap-SB-Sources-of-Sandbox-Discovered-Malware-Bubble	Sources of Sandbox Discovered Malware	virus

```
select
  source,
  sum(total_num) as total_num
from
  ###(select filename, file_name_ext(filename) as suffix, fsaverdict, (CASE WHEN
direction='incoming' THEN dstip ELSE srcip END) as source, (CASE WHEN direction='incoming' THEN
srcip ELSE dstip END) as victim, service, count(*) as total_num from $log where $filter and
dtype='fortisandbox' and nullifna(filename) is not null group by filename, suffix, fsaverdict,
source, victim, service order by total_num desc)### t where fsaverdict not in ('clean','submission
failed') group by source order by total_num desc
```

Dataset Name	Description	Log Category
Total-Recommended-Actions-by-Count	Total Recommended Actions Detected	traffic

```
select
  action,
  sum(totalnum) as totalnum
from
  (
    (
      select
        & #039;Application Vulnerbility Atacks' as action, count(distinct attack) as totalnum from
###(select $flex_timestamp as timestamp, attack, (case when (logflag&16>0) then 1 else 0 end) as
botnet_flag, (CASE WHEN direction='incoming' THEN dstip ELSE srcip END) as source, (CASE WHEN
direction='incoming' THEN srcip ELSE dstip END) as victim, service, count(*) as total_num from
$log-attack where $filter and nullifna(attack) is not null group by timestamp, attack, botnet_
flag, source, victim, service order by total_num desc)### t) union all (select 'Malware Detected'
as action, count(distinct virus) as totalnum from ###(select $flex_timestamp as timestamp, virus,
(CASE WHEN direction='incoming' THEN dstip ELSE srcip END) as source, (CASE WHEN
direction='incoming' THEN srcip ELSE dstip END) as victim, service, count(*) as total_num from
$log-virus where $filter and nullifna(virus) is not null group by timestamp, virus, source,
victim, service order by total_num desc)### t) union all (select 'Botnet Infections' as action,
count(distinct app) as totalnum from ((select distinct app from ###(select app, appcat, apprisk,
srcip, dstip, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
count(*) as totalnum from $log-traffic where $filter and (logflag&1>0) and appcat='Botnet' and
nullifna(app) is not null group by app, appcat, apprisk, srcip, dstip, user_src order by totalnum
desc)### t where apprisk in ('critical', 'high') group by app) union all (select distinct attack
as app from ###(select attack, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`))
as user_src, $flex_timestamp as timestamp, hostname, severity, crlevel, eventtype, service,  (CASE
WHEN direction='incoming' THEN srcip ELSE dstip END) as victim, (CASE WHEN direction='incoming'
THEN dstip ELSE srcip END) as source, count(*) as totalnum from $log-attack where $filter and
(logflag&16>0) group by attack, user_src, timestamp, hostname, severity, crlevel, eventtype,
service, victim, source order by timestamp desc)### t group by attack)) t) union all (select
'Malicious Website' as action, count(distinct hostname) as totalnum from ###(select $flex_
timestamp as timestamp, hostname, count(*) as total_num from $log-webfilter where $filter and
hostname is not null and catdesc='Malicious Websites' group by timestamp, hostname order by total_
num desc)### t) union all (select 'Phishing Websites' as action, count(distinct hostname) as
totalnum from ###(select hostname, count(*) as total_num from $log-webfilter where $filter and
hostname is not null and catdesc='Phishing' group by hostname order by total_num desc)### t) union
```

```
all (select 'Proxy Applications' as action, count(distinct app) as totalnum from ###(select $flex_
timestamp as timestamp, app, appcat, appid, apprisk, (CASE WHEN direction='incoming' THEN dstip
ELSE srcip END) as source, (CASE WHEN direction='incoming' THEN srcip ELSE dstip END) as victim,
count(*) as total_num from $log-app-ctrl where $filter and nullifna(app) is not null group by
timestamp, app, appcat, appid, apprisk, source, victim order by total_num desc)### t where lower
(appcat)='proxy') union all (select 'Remote Access Applications' as action, count(distinct app) as
totalnum from ###(select $flex_timestamp as timestamp, app, appcat, appid, apprisk, (CASE WHEN
direction='incoming' THEN dstip ELSE srcip END) as source, (CASE WHEN direction='incoming' THEN
srcip ELSE dstip END) as victim, count(*) as total_num from $log-app-ctrl where $filter and
nullifna(app) is not null group by timestamp, app, appcat, appid, apprisk, source, victim order by
total_num desc)### t where lower(appcat)='remote.access') union all (select 'P2P and Filesharing
Applications' as action, count(distinct app) as totalnum from ###(select timestamp, app, appcat,
user_src, hostname, sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_out, sum(bandwidth)
as bandwidth, sum(sessions) as sessions from ###base(/*tag:rpt_base_t_top_app*/select $flex_
timestamp as timestamp, dvid, srcip, dstip, epid, euid, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, service, appid, app, appcat, apprisk, hostname, sum
(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as
traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth,
sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions from $log-traffic where $filter and
(logflag&(1|32)>0) and nullifna(app) is not null group by timestamp, dvid, srcip, dstip, epid,
euid, user_src, service, appid, app, appcat, apprisk, hostname order by bandwidth desc, sessions
desc)base### t group by timestamp, app, appcat, user_src, hostname /*SkipSTART*/order by bandwidth
desc, sessions desc/*SkipEND*/)### t where appcat in ('P2P', 'Storage.Backup', 'File.Sharing'))) t
group by action
```

Dataset Name	Description	Log Category
ctap-apprisk-ctrl-High-Risk-Application	Application risk high risk application	traffic

```
select
  risk as d_risk,
  count(distinct user_src) as users,
  id,
  name,
  app_cat,
  technology,
  sum(bandwidth) as bandwidth,
  sum(sessions) as sessions
from
  ###(select app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
action, utmaction, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, count(*) as
sessions from $log where $filter and (logflag&1>0) group by app, user_src, action, utmaction order
by bandwidth desc)### t1 inner join app_mdata t2 on t1.app=t2.name where risk>='4' group by id,
name, app_cat, technology, risk order by d_risk desc, sessions desc
```

Dataset Name	Description	Log Category
ctap-apprisk-ctrl-Application-Vulnerability	Application vulnerabilities discovered	attack

```
select
  attack,
  attackid,
  vuln_type,
  cve,
  severity_number,
  count(
    distinct (
      CASE WHEN direction =& #039;incoming' THEN srcip ELSE dstip END)) as victims, count(distinct
(CASE WHEN direction='incoming' THEN dstip ELSE srcip END)) as sources, sum(totalnum) as totalnum
from ###(select attack, attackid, (case when severity='critical' then 5 when severity='high' then
4 when severity='medium' then 3 when severity='low' then 2 when severity='info' then 1 else 0 end)
as severity_number, direction, dstip, srcip, count(*) as totalnum from $log where $filter and
nullifna(attack) is not null and severity is not null group by attack, attackid, severity,
direction, dstip, srcip order by totalnum desc)### t1 left join (select name, cve, vuln_type from
ips_mdata) t2 on t1.attack=t2.name group by attack, attackid, vuln_type, severity_number, cve
order by severity_number desc, totalnum desc
```

Dataset Name	Description	Log Category
ctap-apprisk-ctrl-Top-Common-Virus-Botnet-Spyware	Common Virus Botnet Spyware	app-ctrl

```
select
  malware as virus,
  (
    case when lower(appcat)=& #039;botnet' then 'Botnet C&C' else (case when malware like
'Riskware%' then 'Spyware' when malware like 'Adware%' then 'Adware' else 'Virus' end) end) as
malware_type, appid, app, count(distinct victim) as victims, count(distinct source) as source, sum
(total_num) as total_num from ((select app as malware, appcat, appid, app, source, victim, sum
(total_num) as total_num from ###(select $flex_timestamp as timestamp, app, appcat, appid,
apprisk, (CASE WHEN direction='incoming' THEN dstip ELSE srcip END) as source, (CASE WHEN
direction='incoming' THEN srcip ELSE dstip END) as victim, count(*) as total_num from $log-app-
ctrl where $filter and nullifna(app) is not null group by timestamp, app, appcat, appid, apprisk,
source, victim order by total_num desc)### t where lower(appcat)='botnet' group by malware,
appcat, appid, app, victim, source, app order by total_num desc) union all (select virus as
malware, 'null' as appcat, 0 as appid, service as app, source, victim, sum(total_num) as total_num
from ###(select $flex_timestamp as timestamp, virus, (CASE WHEN direction='incoming' THEN dstip
ELSE srcip END) as source, (CASE WHEN direction='incoming' THEN srcip ELSE dstip END) as victim,
service, count(*) as total_num from $log-virus where $filter and nullifna(virus) is not null group
by timestamp, virus, source, victim, service order by total_num desc)### t group by malware,
appcat, app, appid, victim, source order by total_num desc) union all (select attack as malware,
'null' as appcat, 0 as appid, service as app, source, victim, sum(total_num) as total_num from ###
(select $flex_timestamp as timestamp, attack, (case when (logflag&16>0) then 1 else 0 end) as
botnet_flag, (CASE WHEN direction='incoming' THEN dstip ELSE srcip END) as source, (CASE WHEN
direction='incoming' THEN srcip ELSE dstip END) as victim, service, count(*) as total_num from
$log-attack where $filter and nullifna(attack) is not null group by timestamp, attack, botnet_
flag, source, victim, service order by total_num desc)### t where botnet_flag>0 group by malware,
appcat, app, appid, victim, source order by total_num desc)) t group by malware, malware_type,
app, appid order by total_num desc
```

Dataset Name	Description	Log Category
CTAP-Malware-Botnet-Spyware-Timeline	Common Virus Botnet Spyware	app-ctrl

```
select
  $flex_timestamp(timestamp) as hodex,
  (
    case when lower(appcat)=& #039;botnet' then 'Botnet' else (case when malware like 'Riskware%'
or malware like 'Adware%' then 'Spyware/Adware' else 'Malware' end) end) as malware_type, sum
(total_num) as total_num from ((select timestamp, appcat, app as malware, sum(total_num) as total_
num from ###(select $flex_timestamp as timestamp, app, appcat, appid, apprisk, (CASE WHEN
direction='incoming' THEN dstip ELSE srcip END) as source, (CASE WHEN direction='incoming' THEN
srcip ELSE dstip END) as victim, count(*) as total_num from $log-app-ctrl where $filter and
nullifna(app) is not null group by timestamp, app, appcat, appid, apprisk, source, victim order by
total_num desc)### t where lower(appcat)='botnet' group by timestamp, appcat, malware order by
total_num desc) union all (select timestamp, 'null' as appcat, virus as malware, sum(total_num) as
total_num from ###(select $flex_timestamp as timestamp, virus, (CASE WHEN direction='incoming'
THEN dstip ELSE srcip END) as source, (CASE WHEN direction='incoming' THEN srcip ELSE dstip END)
as victim, service, count(*) as total_num from $log-virus where $filter and nullifna(virus) is not
null group by timestamp, virus, source, victim, service order by total_num desc)### t group by
timestamp, appcat, malware order by total_num desc) union all (select timestamp, 'null' as appcat,
attack as malware, sum(total_num) as total_num from ###(select $flex_timestamp as timestamp,
attack, (case when (logflag&16>0) then 1 else 0 end) as botnet_flag, (CASE WHEN
direction='incoming' THEN dstip ELSE srcip END) as source, (CASE WHEN direction='incoming' THEN
srcip ELSE dstip END) as victim, service, count(*) as total_num from $log-attack where $filter and
nullifna(attack) is not null group by timestamp, attack, botnet_flag, source, victim, service
order by total_num desc)### t where botnet_flag>0 group by timestamp, appcat, malware order by
total_num desc)) t group by hodex, malware_type order by hodex
```

Dataset Name	Description	Log Category
ctap-App-Risk-Reputation-Top-Devices-By-Scores	Reputation Top Devices By-Scores	traffic

```
select
  coalesce(
    nullifna(`srcname`),
    ipstr(`srcip`),
    nullifna(`srcmac`)
  ) as dev_src,
  sum(crscore % 65536) as scores
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
  and crscore is not null
```

```
group by
  dev_src
having
  sum(crscore % 65536)> 0
order by
  scores desc
```

Dataset Name	Description	Log Category
ctap-App-Risk-Reputation-Top-Devices-By-Scores-Bubble	Reputation Top Devices By-Scores	traffic

```
select
  coalesce(
    nullifna(`srcname`),
    ipstr(`srcip`),
    nullifna(`srcmac`)
  ) as dev_src,
  sum(crscore % 65536) as scores
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
  and crscore is not null
group by
  dev_src
having
  sum(crscore % 65536)> 0
order by
  scores desc
```

Dataset Name	Description	Log Category
ctap-HTTP-SSL-Traffic-Ratio	HTTP SSL Traffic Ratio	traffic

```
select
  (
    case when service in (
      & #039;tcp/80', 'HTTP', 'http') then 'HTTP' else  'HTTPS' end) as service, sum(coalesce
(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from $log where $filter and (logflag&1>0) and
nullifna(app) is not null and service in ('tcp/80', 'tcp/443', 'HTTP', 'HTTPS', 'http', 'https')
group by service having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc
```

Dataset Name	Description	Log Category
ctap-HTTP-SSL-Traffic-Ratio-Donut	HTTP SSL Traffic Ratio	traffic

```
select
  (
    case when service in (
      & #039;tcp/80', 'HTTP', 'http') then 'HTTP' else  'HTTPS' end) as service, sum(coalesce
(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from $log where $filter and (logflag&1>0) and
nullifna(app) is not null and service in ('tcp/80', 'tcp/443', 'HTTP', 'HTTPS', 'http', 'https')
group by service having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc
```

Dataset Name	Description	Log Category
ctap-Top-Source-Countries	Top Source Countries	traffic

```
select
  srccountry,
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  ) as bandwidth
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
  and nullifna(srccountry) is not null
  and srccountry <> & #039;Reserved' group by srccountry having sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0))>0 order by bandwidth desc, srccountry
```

Dataset Name	Description	Log Category
ctap-Top-Source-Countries-Bubble	Top Source Countries	traffic

```
select
  srccountry,
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  ) as bandwidth
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
  and nullifna(srccountry) is not null
  and srccountry <> & #039;Reserved' group by srccountry having sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0))>0 order by bandwidth desc, srccountry
```

Dataset Name	Description	Log Category
ctap-SaaS-Apps	CTAP SaaS Apps	traffic

```
select
  app_group,
  sum(bandwidth) as bandwidth
from
  ###(select app_group_name(app) as app_group, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as
bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out,
count(*) as sessions from $log where $filter and (logflag&1>0) and nullifna(app) is not null group
by app_group having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc)###
t1 inner join app_mdata t2 on lower(t1.app_group)=lower(t2.name) where behavior like '%Cloud%'
group by app_group order by bandwidth desc
```

Dataset Name	Description	Log Category
ctap-SaaS-Apps-Donut	CTAP SaaS Apps	traffic

```
select
  app_group,
  sum(bandwidth) as bandwidth
from
  ###(select app_group_name(app) as app_group, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as
bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out,
count(*) as sessions from $log where $filter and (logflag&1>0) and nullifna(app) is not null group
by app_group having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc)###
t1 inner join app_mdata t2 on lower(t1.app_group)=lower(t2.name) where behavior like '%Cloud%'
group by app_group order by bandwidth desc
```

Dataset Name	Description	Log Category
ctap-IaaS-Apps	CTAP IaaS Apps	traffic

```
select
  app_group,
  sum(bandwidth) as bandwidth
from
  ###(select app_group_name(app) as app_group, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as
bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out,
count(*) as sessions from $log where $filter and (logflag&1>0) and nullifna(app) is not null group
by app_group having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc)###
t1 inner join app_mdata t2 on lower(t1.app_group)=lower(t2.name) where app_cat='Cloud.IT' group by
app_group order by bandwidth desc
```

Dataset Name	Description	Log Category
ctap-IaaS-Apps-Donut	CTAP IaaS Apps	traffic

```
select
  app_group,
  sum(bandwidth) as bandwidth
from
  ###(select app_group_name(app) as app_group, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as
bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out,
count(*) as sessions from $log where $filter and (logflag&1>0) and nullifna(app) is not null group
by app_group having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc)###
t1 inner join app_mdata t2 on lower(t1.app_group)=lower(t2.name) where app_cat='Cloud.IT' group by
app_group order by bandwidth desc
```

Dataset Name	Description	Log Category
ctap-RAS-Apps	CTAP RAS Apps	traffic

```
select
  name as app_group,
  sum(bandwidth) as bandwidth
from
  ###(select app_group_name(app) as app_group, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as
bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out,
count(*) as sessions from $log where $filter and (logflag&1>0) and nullifna(app) is not null group
by app_group having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc)###
t1 inner join app_mdata t2 on lower(t1.app_group)=lower(t2.name) where app_cat='Remote.Access'
group by name order by bandwidth desc
```

Dataset Name	Description	Log Category
ctap-RAS-Apps-Donut	CTAP RAS Apps	traffic

```
select
  name as app_group,
  sum(bandwidth) as bandwidth
from
  ###(select app_group_name(app) as app_group, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as
bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out,
count(*) as sessions from $log where $filter and (logflag&1>0) and nullifna(app) is not null group
by app_group having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc)###
t1 inner join app_mdata t2 on lower(t1.app_group)=lower(t2.name) where app_cat='Remote.Access'
group by name order by bandwidth desc
```

Dataset Name	Description	Log Category
ctap-Proxy-Apps	CTAP Proxy Apps	traffic

```
select
  name as app_group,
  sum(bandwidth) as bandwidth
```

```
from
  ###(select app_group_name(app) as app_group, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as
bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out,
count(*) as sessions from $log where $filter and (logflag&1>0) and nullifna(app) is not null group
by app_group having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc)###
t1 inner join app_mdata t2 on lower(t1.app_group)=lower(t2.name) where app_cat='Proxy' group by
name order by bandwidth desc
```

Dataset Name	Description	Log Category
ctap-Proxy-Apps-Donut	CTAP Proxy Apps	traffic

```
select
  name as app_group,
  sum(bandwidth) as bandwidth
from
  ###(select app_group_name(app) as app_group, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as
bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out,
count(*) as sessions from $log where $filter and (logflag&1>0) and nullifna(app) is not null group
by app_group having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc)###
t1 inner join app_mdata t2 on lower(t1.app_group)=lower(t2.name) where app_cat='Proxy' group by
name order by bandwidth desc
```

Dataset Name	Description	Log Category
ctap-Top-SocialMedia-App-By-Bandwidth	Top SocialMedia Applications by Bandwidth Usage	traffic

```
select
  app_group,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out,
  sum(sessions) as sessions
from
  ###(select app_group_name(app) as app_group, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as
bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out,
count(*) as sessions from $log where $filter and (logflag&1>0) and nullifna(app) is not null group
by app_group having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc)###
t1 inner join app_mdata t2 on lower(t1.app_group)=lower(t2.name) where app_cat='Social.Media'
group by app_group order by bandwidth desc
```

Dataset Name	Description	Log Category
ctap-Top-SocialMedia-App-By-Bandwidth-Bubble	Top SocialMedia Applications by Bandwidth Usage	traffic

```
select
  app_group,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out,
  sum(sessions) as sessions
from
  ###(select app_group_name(app) as app_group, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as
bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out,
count(*) as sessions from $log where $filter and (logflag&1>0) and nullifna(app) is not null group
by app_group having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc)###
t1 inner join app_mdata t2 on lower(t1.app_group)=lower(t2.name) where app_cat='Social.Media'
group by app_group order by bandwidth desc
```

Dataset Name	Description	Log Category
ctap-Top-Streaming-App-By-Bandwidth	Top Streaming applications by bandwidth usage	traffic

```
select
  app_group,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out,
  sum(sessions) as sessions
from
  ###(select app_group_name(app) as app_group, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as
bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out,
count(*) as sessions from $log where $filter and (logflag&1>0) and nullifna(app) is not null group
by app_group having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc)###
t1 inner join app_mdata t2 on lower(t1.app_group)=lower(t2.name) where app_cat='Video/Audio' group
by app_group order by bandwidth desc
```

Dataset Name	Description	Log Category
ctap-Top-Streaming-App-By-Bandwidth-Bubble	Top Streaming applications by bandwidth usage	traffic

```
select
  app_group,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out,
  sum(sessions) as sessions
from
  ###(select app_group_name(app) as app_group, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as
bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out,
count(*) as sessions from $log where $filter and (logflag&1>0) and nullifna(app) is not null group
by app_group having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc)###
```

```
t1 inner join app_mdata t2 on lower(t1.app_group)=lower(t2.name) where app_cat='Video/Audio' group
by app_group order by bandwidth desc
```

Dataset Name	Description	Log Category
ctap-Top-Game-App-By-Bandwidth	Top Game applications by bandwidth usage	traffic

```
select
  app_group,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out,
  sum(sessions) as sessions
from
  ###(select app_group_name(app) as app_group, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as
bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out,
count(*) as sessions from $log where $filter and (logflag&1>0) and nullifna(app) is not null group
by app_group having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc)###
t1 inner join app_mdata t2 on lower(t1.app_group)=lower(t2.name) where app_cat='Game' group by
app_group order by bandwidth desc
```

Dataset Name	Description	Log Category
ctap-Top-Game-App-By-Bandwidth-Bubble	Top Game applications by bandwidth usage	traffic

```
select
  app_group,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out,
  sum(sessions) as sessions
from
  ###(select app_group_name(app) as app_group, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as
bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out,
count(*) as sessions from $log where $filter and (logflag&1>0) and nullifna(app) is not null group
by app_group having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc)###
t1 inner join app_mdata t2 on lower(t1.app_group)=lower(t2.name) where app_cat='Game' group by
app_group order by bandwidth desc
```

Dataset Name	Description	Log Category
ctap-Top-P2P-App-By-Bandwidth	Top P2P applications by bandwidth usage	traffic

```
select
  app_group,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
```

```
  sum(traffic_out) as traffic_out,
  sum(sessions) as sessions
from
  ###(select app_group_name(app) as app_group, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as
bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out,
count(*) as sessions from $log where $filter and (logflag&1>0) and nullifna(app) is not null group
by app_group having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc)###
t1 inner join app_mdata t2 on lower(t1.app_group)=lower(t2.name) where app_cat='P2P' group by app_
group order by bandwidth desc
```

Dataset Name	Description	Log Category
ctap-Top-P2P-App-By-Bandwidth-Bubble	Top P2P applications by bandwidth usage	traffic

```
select
  app_group,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out,
  sum(sessions) as sessions
from
  ###(select app_group_name(app) as app_group, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as
bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out,
count(*) as sessions from $log where $filter and (logflag&1>0) and nullifna(app) is not null group
by app_group having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc)###
t1 inner join app_mdata t2 on lower(t1.app_group)=lower(t2.name) where app_cat='P2P' group by app_
group order by bandwidth desc
```

Dataset Name	Description	Log Category
ctap-apprisk-ctrl-Top-Web-Categories-Visited	Top 25 Web Categories Visited	traffic

```
select
  catdesc,
  count(distinct f_user) as user_num,
  sum(sessions) as sessions,
  sum(bandwidth) as bandwidth
from
  ###(select catdesc, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as f_
user, count(*) as sessions, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from
$log-traffic where $filter and catdesc is not null and (logflag&1>0) and (countweb>0 or ((logver
is null or logver<502000000) and (hostname is not null or utmevent in ('webfilter', 'banned-word',
'web-content', 'command-block', 'script-filter')))) group by f_user, catdesc order by sessions
desc)### t group by catdesc order by sessions desc
```

Dataset Name	Description	Log Category
apprisk-ctrl-Top-Web-Categories-Visited-by-Bandwidth	Top 25 Web Categories Visited	traffic

```
select
  catdesc,
  sum(bandwidth) as bandwidth
from
  ###(select catdesc, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as f_
user, count(*) as sessions, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from
$log-traffic where $filter and catdesc is not null and (logflag&1>0) and (countweb>0 or ((logver
is null or logver<502000000) and (hostname is not null or utmevent in ('webfilter', 'banned-word',
'web-content', 'command-block', 'script-filter')))) group by f_user, catdesc order by sessions
desc)### t group by catdesc order by bandwidth desc
```

Dataset Name	Description	Log Category
apprisk-ctrl-Top-Web-Categories-Visited	Top 25 Web Categories Visited	traffic

```
select
  catdesc,
  count(distinct f_user) as user_num,
  sum(sessions) as sessions,
  sum(bandwidth) as bandwidth
from
  ###(select catdesc, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as f_
user, count(*) as sessions, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from
$log-traffic where $filter and catdesc is not null and (logflag&1>0) and (countweb>0 or ((logver
is null or logver<502000000) and (hostname is not null or utmevent in ('webfilter', 'banned-word',
'web-content', 'command-block', 'script-filter')))) group by f_user, catdesc order by sessions
desc)### t group by catdesc order by sessions desc
```

Dataset Name	Description	Log Category
ctap-App-Risk-Applications-Running-Over-HTTP	Application risk applications running over HTTP	traffic

```
select
  app_group,
  service,
  sum(sessions) as sessions,
  sum(bandwidth) as bandwidth
from
  ###(select app_group_name(app) as app_group, appcat, service, sum(coalesce(sentdelta, sentbyte,
0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as
traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from $log
```

```
where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by app_group, appcat,
service order by bandwidth desc)### t where service in ('tcp/80', 'tcp/443', 'HTTP', 'HTTPS',
'http', 'https') group by app_group, service having sum(bandwidth)>0 order by bandwidth desc
```

Dataset Name	Description	Log Category
ctap-App-Risk-Web-Browsing-Activity-Hostname-Category	Application risk web browsing activity hostname category	webfilter

```
select
  catdesc,
  domain,
  sum(visits) as visits
from
  ###(select coalesce(nullifna(hostname), ipstr(`dstip`)) as domain, catdesc, count(*) as visits
from $log where $filter and catdesc is not null group by domain, catdesc order by visits desc)###
t group by catdesc, domain order by visits desc
```

Dataset Name	Description	Log Category
ctap-Top-Web-Domain-and-Category-by-Visits	Application risk web browsing activity hostname category	webfilter

```
select
  catdesc,
  domain,
  sum(visits) as visits
from
  ###(select coalesce(nullifna(hostname), ipstr(`dstip`)) as domain, catdesc, count(*) as visits
from $log where $filter and catdesc is not null group by domain, catdesc order by visits desc)###
t group by catdesc, domain order by visits desc
```

Dataset Name	Description	Log Category
ctap-Top-Sites-By-Browsing-Time	Traffic top sites by browsing time	traffic

```
select
  hostname,
  string_agg(
    distinct catdesc,
    & #039;, ') as agg_catdesc, ebtr_value(ebtr_agg_flat(browsetime), null, $timespan) as
browsetime, sum(bandwidth) as bandwidth, sum(traffic_in) as traffic_in, sum(traffic_out) as
traffic_out from ###(select hostname, catdesc, ebtr_agg_flat(browsetime) as browsetime, sum
(bandwidth) as bandwidth, sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_out from
(select hostname, catdesc, ebtr_agg_flat($browse_time) as browsetime, sum(coalesce(sentbyte,
0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce
(sentbyte, 0)) as traffic_out from $log where $filter and (logflag&1>0) and hostname is not null
and $browse_time is not null group by hostname, catdesc) t group by hostname, catdesc
```

```
/*SkipSTART*/order by ebtr_value(ebtr_agg_flat(browsetime), null, null) desc/*SkipEND*/)### t
group by hostname order by browsetime desc
```

Dataset Name	Description	Log Category
ctap-Top-Sites-and-Category-by-Browsing-Time	Traffic Top Sites and Category by Browsing Time	traffic

```
select
  catdesc,
  hostname,
  ebtr_value(
    ebtr_agg_flat(browsetime),
    null,
    $timespan
  ) as browsetime
from
  ###(select hostname, catdesc, ebtr_agg_flat(browsetime) as browsetime, sum(bandwidth) as
bandwidth, sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_out from (select hostname,
catdesc, ebtr_agg_flat($browse_time) as browsetime, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte,
0)) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_
out from $log where $filter and (logflag&1>0) and hostname is not null and $browse_time is not
null group by hostname, catdesc) t group by hostname, catdesc /*SkipSTART*/order by ebtr_value
(ebtr_agg_flat(browsetime), null, null) desc/*SkipEND*/)### t group by catdesc, hostname order by
browsetime desc
```

Dataset Name	Description	Log Category
ctap-Average-Bandwidth-Hour	Average Bandwidth Hour	traffic

```
select
  hourstamp,
  sum(bandwidth)/ count(distinct daystamp) as bandwidth
from
  ###(select to_char(from_dtime(dtime), 'HH24:00') as hourstamp, to_char(from_dtime(dtime), 'DD
Mon') as daystamp, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from $log where
$filter and (logflag&1>0) group by hourstamp, daystamp having sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0))>0 order by hourstamp)### t group by hourstamp order by hourstamp
```

Dataset Name	Description	Log Category
ctap-Top-Bandwidth-Hosts	Top Bandwidth Hosts	traffic

```
select
  hostname,
  sum(bandwidth) as bandwidth
from
  ###(select timestamp, app, appcat, user_src, hostname, sum(traffic_in) as traffic_in, sum
```

```
(traffic_out) as traffic_out, sum(bandwidth) as bandwidth, sum(sessions) as sessions from ###base
(/*tag:rpt_base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid, euid,
coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service, appid,
app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce
(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta,
rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions from $log-
traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by timestamp,
dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat, apprisk, hostname order by
bandwidth desc, sessions desc)base### t group by timestamp, app, appcat, user_src, hostname
/*SkipSTART*/order by bandwidth desc, sessions desc/*SkipEND*/)### t where hostname is not null
group by hostname order by bandwidth desc
```

Dataset Name	Description	Log Category
ctap-Top-Bandwidth-Hosts-Bubble	Top Bandwidth Hosts	traffic

```
select
  hostname,
  sum(bandwidth) as bandwidth
from
  ###(select timestamp, app, appcat, user_src, hostname, sum(traffic_in) as traffic_in, sum
(traffic_out) as traffic_out, sum(bandwidth) as bandwidth, sum(sessions) as sessions from ###base
(/*tag:rpt_base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid, euid,
coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service, appid,
app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce
(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta,
rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions from $log-
traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by timestamp,
dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat, apprisk, hostname order by
bandwidth desc, sessions desc)base### t group by timestamp, app, appcat, user_src, hostname
/*SkipSTART*/order by bandwidth desc, sessions desc/*SkipEND*/)### t where hostname is not null
group by hostname order by bandwidth desc
```

Dataset Name	Description	Log Category
saas-Application-Discovered	All Applications Discovered on the Network	traffic

```
select
  (
    case is_saas when 1 then & #039;SaaS Apps' else 'Other Apps' end) as app_type, count(distinct
app_s) as total_num from ###(select app_s, (case when saas_s>=10 then 1 else 0 end) as is_saas
from (select unnest(apps) as app_s, unnest(saasinfo) as saas_s from $log where $filter and apps is
not null) t group by app_s, is_saas order by is_saas desc)### t group by is_saas order by is_saas
```

Dataset Name	Description	Log Category
saas-SaaS-Application-by-Category	Number of SaaS Applications by Category	traffic

```
select
  (
    case saas_cat when 0 then & #039;Sanctioned' else 'Unsanctioned' end) as saas_cat_str, count
(distinct app_s) as num_saas_app from ###(select app_s, saas_s%10 as saas_cat, sum
(sentbyte+rcvdbyte) as bandwidth, count(*) as total_app from (select unnest(apps) as app_s, unnest
(saasinfo) as saas_s, coalesce(sentbyte, 0) as sentbyte, coalesce(rcvdbyte, 0) as rcvdbyte from
$log where $filter and apps is not null) t where saas_s>=10 group by app_s, saas_cat order by
bandwidth desc)### t where saas_cat in (0, 1) group by saas_cat order by saas_cat
```

Dataset Name	Description	Log Category
saas-SaaS-Application-by-Bandwidth	Number of SaaS Applications by Bandwidth	traffic

```
select
  (
    case saas_cat when 0 then & #039;Sanctioned' else 'Tolerated' end) as saas_cat_str, sum
(bandwidth) as bandwidth from ###(select app_s, saas_s%10 as saas_cat, sum(sentbyte+rcvdbyte) as
bandwidth, count(*) as total_app from (select unnest(apps) as app_s, unnest(saasinfo) as saas_s,
coalesce(sentbyte, 0) as sentbyte, coalesce(rcvdbyte, 0) as rcvdbyte from $log where $filter and
apps is not null) t where saas_s>=10 group by app_s, saas_cat order by bandwidth desc)### t where
saas_cat in (0, 2) group by saas_cat order by saas_cat
```

Dataset Name	Description	Log Category
saas-SaaS-Application-by-Session	Number of SaaS Applications by Session	traffic

```
select
  (
    case saas_cat when 0 then & #039;Sanctioned' else 'Tolerated' end) as saas_cat_str, sum(total_
app) as total_app from ###(select app_s, saas_s%10 as saas_cat, sum(sentbyte+rcvdbyte) as
bandwidth, count(*) as total_app from (select unnest(apps) as app_s, unnest(saasinfo) as saas_s,
coalesce(sentbyte, 0) as sentbyte, coalesce(rcvdbyte, 0) as rcvdbyte from $log where $filter and
apps is not null) t where saas_s>=10 group by app_s, saas_cat order by bandwidth desc)### t where
saas_cat in (0, 2) group by saas_cat order by saas_cat
```

Dataset Name	Description	Log Category
saas-SaaS-App-Users-vs-Others	Number of Users of SaaS Apps vs Others	traffic

```
select
  (
    case is_saas when 0 then & #039;Other Apps' else 'SaaS Apps' end) as app_type, count(distinct
saasuser) as total_user from ###(select saasuser, saas_s/10 as is_saas, count(*) as total_num from
(select coalesce(nullifna(`user`), nullifna(`clouduser`), nullifna(`unauthuser`), srcname, ipstr
(`srcip`)) as saasuser, unnest(saasinfo) as saas_s from $log where $filter and apps is not null) t
group by saasuser, is_saas order by total_num desc)### t group by app_type
```

Dataset Name	Description	Log Category
saas-SaaS-App-Users	Number of Users of SaaS Apps	traffic

```
select
  (
    case saas_cat when 0 then & #039;Sanctioned' when 1 then 'Unsanctioned' else 'Others' end) as
app_type, count(distinct saasuser) as total_user from ###(select saasuser, saas_s%10 as saas_cat,
count(*) as total_num from (select coalesce(nullifna(`user`), nullifna(`clouduser`), nullifna
(`unauthuser`), srcname, ipstr(`srcip`)) as saasuser, unnest(saasinfo) as saas_s from $log where
$filter and apps is not null) t where saas_s>=10 group by saasuser, saas_cat order by total_num
desc)### t group by saas_cat order by saas_cat
```

Dataset Name	Description	Log Category
saas-Top-SaaS-User-by-Bandwidth-Session	Top SaaS Users by Bandwidth and Session	traffic

```
select
  saasuser,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out,
  sum(sessions) as sessions,
  sum(session_block) as session_block,
  (
    sum(sessions)- sum(session_block)
  ) as session_pass,
  count(distinct app_s) as total_app
from
  ###(select saasuser, app_s, sum(sentbyte+rcvdbyte) as bandwidth, sum(rcvdbyte) as traffic_in,
sum(sentbyte) as traffic_out, count(*) as sessions, sum(is_blocked) as session_block from (select
coalesce(nullifna(`user`), nullifna(`clouduser`), nullifna(`unauthuser`), srcname, ipstr(`srcip`))
as saasuser, unnest(apps) as app_s, unnest(saasinfo) as saas_s, coalesce(sentbyte, 0) as sentbyte,
coalesce(rcvdbyte, 0) as rcvdbyte, (CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END) as is_blocked from
$log where $filter and apps is not null) t where saas_s>=10 group by saasuser, app_s order by
bandwidth desc)### t group by saasuser order by bandwidth desc
```

Dataset Name	Description	Log Category
saas-Top-Category-by-SaaS-Application-Usage	Top Categories by SaaS Application Usage	traffic

```
select
  app_cat,
  (
    case saas_cat when 0 then & #039;Sanctioned' else 'Unsactioned' end) as saas_cat_str, count
(distinct app_s) as total_app from ###(select app_s, saas_s%10 as saas_cat, count(*) as total_num
```

```
from (select unnest(apps) as app_s, unnest(saasinfo) as saas_s from $log where $filter and apps is
not null) t where saas_s>=10 group by app_s, saas_cat order by total_num desc)### t1 inner join
app_mdata t2 on t1.app_s=t2.name where saas_cat in (0, 1) group by app_cat, saas_cat order by
total_app desc
```

Dataset Name	Description	Log Category
saas-Top-SaaS-Category-by-Number-of-User	Top SaaS Categories by Number of Users	traffic

```
select
  app_cat,
  (
    case saas_cat when 0 then & #039;Sanctioned' else 'Unsactioned' end) as saas_cat_str, count
(distinct saasuser) as total_user from ###(select app_s, saas_s%10 as saas_cat, saasuser from
(select unnest(apps) as app_s, unnest(saasinfo) as saas_s, coalesce(nullifna(`user`), nullifna
(`clouduser`), nullifna(`unauthuser`), srcname, ipstr(`srcip`)) as saasuser from $log where
$filter and apps is not null) t where saas_s>=10 group by app_s, saas_cat, saasuser order by saas_
cat desc)### t1 inner join app_mdata t2 on t1.app_s=t2.name where saas_cat in (0, 1) group by app_
cat, saas_cat order by total_user desc
```

Dataset Name	Description	Log Category
saas-Top-User-by-Number-of-SaaS-Application	Top Users by Number of SaaS Applications	traffic

```
select
  saasuser,
  (
    case saas_cat when 0 then & #039;Sanctioned' else 'Unsactioned' end) as saas_cat_str, count
(distinct app_s) as total_app from ###(select app_s, saas_s%10 as saas_cat, saasuser from (select
unnest(apps) as app_s, unnest(saasinfo) as saas_s, coalesce(nullifna(`user`), nullifna
(`clouduser`), nullifna(`unauthuser`), srcname, ipstr(`srcip`)) as saasuser from $log where
$filter and apps is not null) t where saas_s>=10 group by app_s, saas_cat, saasuser order by saas_
cat desc)### t where saas_cat in (0, 1) group by saasuser, saas_cat order by total_app desc
```

Dataset Name	Description	Log Category
saas-Top-SaaS-Application-by-Bandwidth-Session	Top SaaS Applications by Sessions and Bandwidth	traffic

```
select
  t2.id as app_id,
  app_s,
  app_cat,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out,
```

```
  sum(sessions) as sessions,
  sum(session_block) as session_block,
  (
    sum(sessions)- sum(session_block)
  ) as session_pass
from
  ###(select app_s, sum(sentbyte+rcvdbyte) as bandwidth, sum(rcvdbyte) as traffic_in, sum
(sentbyte) as traffic_out, count(*) as sessions, sum(is_blocked) as session_block from (select
unnest(apps) as app_s, unnest(saasinfo) as saas_s, coalesce(sentbyte, 0) as sentbyte, coalesce
(rcvdbyte, 0) as rcvdbyte, (CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END) as is_blocked from $log
where $filter and apps is not null) t where saas_s>=10 group by app_s order by bandwidth desc,
traffic_in desc, traffic_out desc, sessions desc, session_block desc)### t1 inner join app_mdata
t2 on t1.app_s=t2.name group by app_id, app_s, app_cat order by bandwidth desc
```

Dataset Name	Description	Log Category
saas-Top-Tolerated-SaaS-Application-by-Bandwidth	Top Tolerated SaaS Applications by Bandwidth	traffic

```
select
  app_s,
  sum(sentbyte + rcvdbyte) as bandwidth
from
  (
    select
      unnest(apps) as app_s,
      unnest(saasinfo) as saas_s,
      coalesce(sentbyte, 0) as sentbyte,
      coalesce(rcvdbyte, 0) as rcvdbyte
    from
      $log
    where
      $filter
      and apps is not null
  ) t
where
  saas_s = 12
group by
  app_s
order by
  bandwidth desc
```

Dataset Name	Description	Log Category
saas-drilldown-Top-Tolerated-SaaS-Application	Top Tolerated SaaS Applications	traffic

```
select
  app_s,
```

```
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out,
  sum(sessions) as sessions,
  sum(session_block) as session_block,
  (
    sum(sessions)- sum(session_block)
  ) as session_pass
from
  ###(select saasuser, app_s, sum(sentbyte+rcvdbyte) as bandwidth, sum(rcvdbyte) as traffic_in,
sum(sentbyte) as traffic_out, count(*) as sessions, sum(is_blocked) as session_block from (select
coalesce(nullifna(`user`), nullifna(`clouduser`), nullifna(`unauthuser`), srcname, ipstr(`srcip`))
as saasuser, unnest(apps) as app_s, unnest(saasinfo) as saas_s, coalesce(sentbyte, 0) as sentbyte,
coalesce(rcvdbyte, 0) as rcvdbyte, (CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END) as is_blocked from
$log where $filter and apps is not null) t where saas_s=12 group by saasuser, app_s order by
bandwidth desc)### t where $filter-drilldown group by app_s order by bandwidth desc
```

Dataset Name	Description	Log Category
saas-Top-User-by-Tolerated-SaaS-Application-Drilldown	Top Users by Tolerated SaaS Applications	traffic

```
select
  saasuser,
  count(distinct app_s) as total_app
from
  ###(select saasuser, app_s, sum(sentbyte+rcvdbyte) as bandwidth, sum(rcvdbyte) as traffic_in,
sum(sentbyte) as traffic_out, count(*) as sessions, sum(is_blocked) as session_block from (select
coalesce(nullifna(`user`), nullifna(`clouduser`), nullifna(`unauthuser`), srcname, ipstr(`srcip`))
as saasuser, unnest(apps) as app_s, unnest(saasinfo) as saas_s, coalesce(sentbyte, 0) as sentbyte,
coalesce(rcvdbyte, 0) as rcvdbyte, (CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END) as is_blocked from
$log where $filter and apps is not null) t where saas_s=12 group by saasuser, app_s order by
bandwidth desc)### t group by saasuser order by total_app desc
```

Dataset Name	Description	Log Category
saas-drilldown-Top-File-Sharing-SaaS-Application-Detail	Top File Sharing SaaS Applications Detail	traffic

```
select
  saasuser,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out,
  sum(sessions) as sessions,
  sum(session_block) as session_block,
  (
    sum(sessions)- sum(session_block)
  ) as session_pass
```

```
from
  ###(select app_group_name(app_s) as app_group, saasuser, sum(sentbyte+rcvdbyte) as bandwidth,
sum(rcvdbyte) as traffic_in, sum(sentbyte) as traffic_out, count(*) as sessions, sum(is_blocked)
as session_block from (select coalesce(nullifna(`user`), nullifna(`clouduser`), nullifna
(`unauthuser`), srcname, ipstr(`srcip`)) as saasuser, unnest(apps) as app_s, unnest(saasinfo) as
saas_s, coalesce(sentbyte, 0) as sentbyte, coalesce(rcvdbyte, 0) as rcvdbyte, (CASE WHEN
(logflag&2>0) THEN 1 ELSE 0 END) as is_blocked from $log where $filter and apps is not null) t
where saas_s>=10 group by app_group, saasuser order by bandwidth desc)### t where $filter-
drilldown group by saasuser order by sessions desc
```

Dataset Name	Description	Log Category
saas-Top-File-Sharing-SaaS-Application	Top File Sharing Applications	traffic

```
select
  t2.id as appid,
  (
    case t2.risk when & #039;5' then 'Critical' when '4' then 'High' when '3' then 'Medium' when
'2' then 'Info' else 'Low' end) as risk, app_group, bandwidth, traffic_in, traffic_out, sessions,
session_block, session_pass, total_user from (select app_group, count(distinct saasuser) as total_
user, sum(bandwidth) as bandwidth, sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_out,
sum(sessions) as sessions, sum(session_block) as session_block, (sum(sessions)-sum(session_block))
as session_pass from ###(select app_group_name(app_s) as app_group, saasuser, sum
(sentbyte+rcvdbyte) as bandwidth, sum(rcvdbyte) as traffic_in, sum(sentbyte) as traffic_out, count
(*) as sessions, sum(is_blocked) as session_block from (select coalesce(nullifna(`user`), nullifna
(`clouduser`), nullifna(`unauthuser`), srcname, ipstr(`srcip`)) as saasuser, unnest(apps) as app_
s, unnest(saasinfo) as saas_s, coalesce(sentbyte, 0) as sentbyte, coalesce(rcvdbyte, 0) as
rcvdbyte, (CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END) as is_blocked from $log where $filter and
apps is not null) t where saas_s>=10 group by app_group, saasuser order by bandwidth desc)### t
group by app_group) t1 inner join app_mdata t2 on lower(t1.app_group)=lower(t2.name) where t2.app_
cat='Storage.Backup' order by total_user desc, bandwidth desc
```

Dataset Name	Description	Log Category
saas-Top-File-Sharing-SaaS-Application-Drilldown	Top File Sharing Applications	traffic

```
select
  t2.id as appid,
  (
    case t2.risk when & #039;5' then 'Critical' when '4' then 'High' when '3' then 'Medium' when
'2' then 'Info' else 'Low' end) as risk, app_group, bandwidth, traffic_in, traffic_out, sessions,
session_block, session_pass, total_user from (select app_group, count(distinct saasuser) as total_
user, sum(bandwidth) as bandwidth, sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_out,
sum(sessions) as sessions, sum(session_block) as session_block, (sum(sessions)-sum(session_block))
as session_pass from ###(select app_group_name(app_s) as app_group, saasuser, sum
(sentbyte+rcvdbyte) as bandwidth, sum(rcvdbyte) as traffic_in, sum(sentbyte) as traffic_out, count
(*) as sessions, sum(is_blocked) as session_block from (select coalesce(nullifna(`user`), nullifna
```

```
(`clouduser`), nullifna(`unauthuser`), srcname, ipstr(`srcip`)) as saasuser, unnest(apps) as app_
s, unnest(saasinfo) as saas_s, coalesce(sentbyte, 0) as sentbyte, coalesce(rcvdbyte, 0) as
rcvdbyte, (CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END) as is_blocked from $log where $filter and
apps is not null) t where saas_s>=10 group by app_group, saasuser order by bandwidth desc)### t
group by app_group) t1 inner join app_mdata t2 on lower(t1.app_group)=lower(t2.name) where t2.app_
cat='Storage.Backup' order by total_user desc, bandwidth desc
```

Dataset Name	Description	Log Category
aware-Device-By-Location	Device by Location	traffic

```
select
  & #039;All'::text as country, count(distinct devid) as device_count from ###(select devid from
$log where $filter group by devid order by devid)### t
```

Dataset Name	Description	Log Category
aware-Network-Endpoint-Devices	Endpoint Devices on Network	

```
select
  category,
  total_num
from
  (
    select
      & #039;Seen Devices' as category, 1 as idx, count(distinct epname) as total_num from (select
epname, map_dev.devid, map_dev.vd, max(lastseen) as itime from $ADOM_ENDPOINT t inner join $ADOM_
EPEU_DEVMAP map_dev on t.epid=map_dev.epid where $filter-drilldown and epname is not null group by
epname, map_dev.devid, map_dev.vd) t where $filter and $filter-drilldown union all select 'New
Devices' as category, 2 as idx, count(distinct epname) as total_num from (select epname, map_
dev.devid, map_dev.vd, min(firstseen) as itime from $ADOM_ENDPOINT  t inner join $ADOM_EPEU_DEVMAP
map_dev on t.epid=map_dev.epid where epname is not null group by epname, map_dev.devid, map_
dev.vd) t where $filter and $filter-drilldown union all select 'Unseen Devices' as category, 3 as
idx, count(distinct t1.epname) as total_num from $ADOM_ENDPOINT t1 where not exists (select 1 from
(select epname, map_dev.devid, map_dev.vd, max(lastseen) as itime from $ADOM_ENDPOINT t inner join
$ADOM_EPEU_DEVMAP map_dev on t.epid=map_dev.epid where epname is not null group by epname, map_
dev.devid, map_dev.vd) t2 where $filter and $filter-drilldown and t1.epname=t2.epname)) t order by
idx
```

Dataset Name	Description	Log Category
aware-New-Endpoint-Devices	New Endpoint Devices	

```
drop
  table if exists devmap_tmp; create temporary table devmap_tmp as (
    select
      epid,
      max(euid) as max_euid
```

```
    from
      $ADOM_EPEU_DEVMAP tx
    where
      $filter - drilldown
      and euid >= 1024
    group by
      epid
  );
select
  timestamp,
  epname as hostname,
  max(osname) as osname,
  max(devtype) as devtype,
  max(srcip) as srcip,
  string_agg(
    distinct epname,
    & #039;,') as user_agg from (select from_itime(itime) as timestamp, osname, epname, epdevtype
as devtype, epip as srcip, epid from (select max(osname) as osname, max(epname) as epname, max
(epdevtype) as epdevtype, max(epip) as epip, t.epid, map_dev.devid, map_dev.vd, min(firstseen) as
itime from $ADOM_ENDPOINT t inner join $ADOM_EPEU_DEVMAP map_dev on t.epid=map_dev.epid where
epname is not null group by epname, t.epid, map_dev.devid, map_dev.vd) t where $filter and
$filter-drilldown) t1 inner join devmap_tmp on devmap_tmp.epid=t1.epid inner join $ADOM_ENDUSER as
teu on devmap_tmp.max_euid=teu.euid group by timestamp, hostname order by timestamp desc
```

Dataset Name	Description	Log Category
aware-New-Endpoint-Devices-Trend	New Endpoint Devices Trend	

```
select
  $flex_timescale(itime) as hodex,
  count(distinct epname) as total_num
from
  (
    select
      epname,
      map_dev.devid,
      map_dev.vd,
      min(firstseen) as itime
    from
      $ADOM_ENDPOINT t
      inner join $ADOM_EPEU_DEVMAP map_dev on t.epid = map_dev.epid
    where
      $filter - drilldown
      and epname is not null
    group by
      epname,
      map_dev.devid,
      map_dev.vd
  ) t
where
```

```
  $filter
  and $filter - drilldown
group by
  hodex
order by
  hodex
```

Dataset Name	Description	Log Category
aware-Top-Endpoint-Operating-Systems	Top Endpoint Operating Systems	fct-traffic

```
select
  os1 as os,
  count(distinct hostname) as total_num
from
  ###(select split_part(os, ',', 1) as os1, hostname, count(*) as total_num from $log where
$filter and nullifna(os) is not null group by os1, hostname order by total_num desc)### t group by
os order by total_num desc
```

Dataset Name	Description	Log Category
aware-Top-Endpoint-Applications-Windows	Top Endpoint Applications Windows	fct-traffic

```
select
  srcname1 as srcname,
  count(distinct hostname) as total_num
from
  ###(select split_part(srcname, '.', 1) as srcname1, hostname, count(*) as total_num from $log
where $filter and nullifna(srcname) is not null and lower(os) like '%windows%' group by srcname,
hostname order by total_num desc)### t group by srcname order by total_num desc
```

Dataset Name	Description	Log Category
aware-Top-Endpoint-Applications-Mac	Top Endpoint Applications Mac	fct-traffic

```
select
  srcname1 as srcname,
  count(distinct hostname) as total_num
from
  ###(select split_part(srcname, '.', 1) as srcname1, hostname, count(*) as total_num from $log
where $filter and nullifna(srcname) is not null and lower(os) like '%mac os%' group by srcname,
hostname order by total_num desc)### t group by srcname order by total_num desc
```

Dataset Name	Description	Log Category
aware-Top-SaaS-Application-by-Number-of-Users	Top SaaS Applications by Number of Users	traffic

```
select
  app_group,
  count(distinct saasuser) as total_user
from
  ###(select app_group_name(app_s) as app_group, saasuser, count(*) as total_num from (select
unnest(apps) as app_s, unnest(saasinfo) as saas_s, coalesce(nullifna(`user`), nullifna
(`clouduser`), nullifna(`unauthuser`), srcname, ipstr(`srcip`)) as saasuser from $log where
$filter and (logflag&1>0) and apps is not null) t where saas_s>=10 group by app_group, saasuser
order by total_num desc)### t group by app_group order by total_user desc
```

Dataset Name	Description	Log Category
aware-Summary-Of-Changes	Summary of Changes	event

```
select
  regexp_replace(
    msg,
    & #039;[^ ]*$','') as msg_trim, count(*) as total_num from $log where $filter and logid_to_int
(logid)=44547 group by msg_trim order by total_num desc
```

Dataset Name	Description	Log Category
aware-Change-Details	Change Details	event

```
select
  $calendar_time as timestamp,
  `user`,
  ui,
  msg
from
  $log
where
  $filter
  and logid_to_int(logid)= 44547
group by
  timestamp,
  `user`,
  ui,
  msg
order by
  timestamp desc
```

Dataset Name	Description	Log Category
aware-Vulnerabilities-By-Severity	Vulnerabilities by Security	fct-netscan

```
select
  vulnseverity,
  count(distinct vulnname) as vuln_num
from
  ###(select vulnseverity, vulnname, count(*) as total_num from $log where $filter and nullifna
(vulnname) is not null and nullifna(vulnseverity) is not null group by vulnseverity, vulnname
order by total_num desc)### t group by vulnseverity order by vuln_num desc
```

Dataset Name	Description	Log Category
aware-Vulnerabilities-Trend	Vulnerabilities Trend	fct-netscan

```
select
  $flex_timescale(timestamp) as timescale,
  sum(critical) as critical,
  sum(high) as high,
  sum(medium) as medium,
  sum(low) as low
from
  ###(select $flex_timestamp as timestamp, sum(case when lower(vulnseverity) = 'critical' then 1
else 0 end) as critical, sum(case when lower(vulnseverity) = 'high' then 1 else 0 end) as high,
sum(case when lower(vulnseverity) = 'medium' then 1 else 0 end) as medium, sum(case when lower
(vulnseverity) = 'notice' then 1 else 0 end) as Low from $log where $filter group by timestamp
order by timestamp desc)### t group by timescale order by timescale
```

Dataset Name	Description	Log Category
aware-Top-Critical-Vulnerabilities	Top Critical Vulnerabilities	fct-netscan

```
select
  vulnname,
  vulnseverity,
  vulncat,
  count(distinct hostname) as total_num
from
  ###(select hostname, vulnname, vulnseverity, vulncat, count(*) as total_num from $log where
$filter and nullifna(vulnname) is not null and vulnseverity='Critical' group by hostname,
vulnname, vulnseverity, vulncat order by total_num desc)### t group by vulnname, vulnseverity,
vulncat order by total_num desc
```

Dataset Name	Description	Log Category
aware-Top-Vulnerabilities-Last-Period	Top Vulnerabilities Last Period	fct-netscan

```
select
  vulnname,
  vulnseverity,
  sev_num,
  vulncat,
  count(distinct hostname) as total_num
from
  ###(select hostname, vulnname, vulnseverity, (CASE vulnseverity WHEN 'Critical' THEN 5 WHEN
'High' THEN 4 WHEN 'Medium' THEN 3 WHEN 'Info' THEN 2 WHEN 'Low' THEN 1 ELSE 0 END) as sev_num,
vulncat, count(*) as total_num from $log where $pre_period $filter and nullifna(vulnname) is not
null group by hostname, vulnname, vulnseverity, vulncat order by sev_num desc, total_num desc)###
t group by vulnname, vulnseverity, sev_num, vulncat order by sev_num desc, total_num desc
```

Dataset Name	Description	Log Category
aware-Top-New-Vulnerabilities	Top New Vulnerabilities	fct-netscan

```
drop
  table if exists rpt_tmptbl_1;
drop
  table if exists rpt_tmptbl_2; create temporary table rpt_tmptbl_1 as
select
  vulnid,
  vulnname,
  vulnseverity,
  vulncat,
  hostname
from
  ###(select vulnid, vulnname, vulnseverity, vulncat, hostname, count(*) as total_num  from $log
where $pre_period $filter and nullifna(vulnname) is not null group by vulnid, vulnname,
vulnseverity, vulncat, hostname order by total_num desc)### t group by vulnid, vulnname,
vulnseverity, vulncat, hostname; create temporary table rpt_tmptbl_2 as select vulnid, vulnname,
vulnseverity, vulncat, hostname from ###(select vulnid, vulnname, vulnseverity, vulncat, hostname,
count(*) as total_num from $log where $filter and nullifna(vulnname) is not null group by vulnid,
vulnname, vulnseverity, vulncat, hostname order by total_num desc)### t group by vulnid, vulnname,
vulnseverity, vulncat, hostname; select vulnname, (case when vulnseverity='Critical' then 5 when
vulnseverity='High' then 4 when vulnseverity='Medium' then 3 when vulnseverity='Low' then 2 when
vulnseverity='Info' then 1 else 0 end) as sev, vulnseverity, vulncat, count(distinct hostname) as
host_num, cve_id from rpt_tmptbl_2 t1 left join fct_mdata t2 on t1.vulnid=t2.vid::int where not
exists (select 1 from rpt_tmptbl_1 where t1.vulnid=rpt_tmptbl_1.vulnid) group by vulnname, sev,
vulnseverity, vulncat, cve_id order by sev desc, host_num desc
```

Dataset Name	Description	Log Category
aware-Top-User-With-Critical-Vulnerabilities	Top Users with Critical Vulnerabilities	fct-netscan

```
select
  hostname,
```

```
  `user` as user_src,
  vulnname,
  vulncat,
  count(*) as total_num
from
  $log
where
  $filter
  and nullifna(`user`) is not null
  and vulnseverity =& #039;Critical' group by hostname, user_src, vulnname, vulncat order by
total_num desc
```

Dataset Name	Description	Log Category
aware-Ingress-Data-Flow-By-Zone	Ingress Data Flow By Zone	traffic

```
select
  app,
  tag,
  sum(rcvdbyte) as rcvdbyte
from
  ###(select dvid, app, dstintf, sum(coalesce(rcvdbyte, 0)) as rcvdbyte from $log where $filter
group by dvid, app, dstintf having sum(coalesce(rcvdbyte, 0)) > 0 order by rcvdbyte desc)### tt1
inner join (select dvid, intfname, unnest(tags) as tag from $ADOM_INTF_INFO ti) tt2 on
tt1.dvid=tt2.dvid and tt1.dstintf=tt2.intfname group by app, tag order by rcvdbyte desc
```

Dataset Name	Description	Log Category
aware-Egress-Data-Flow-By-Zone	Egress Data Flow By Zone	traffic

```
select
  app,
  tag,
  sum(sentbyte) as sentbyte
from
  ###(select dvid, app, srcintf, sum(coalesce(sentbyte, 0)) as sentbyte from $log where $filter
group by dvid, app, srcintf having sum(coalesce(sentbyte, 0)) > 0 order by sentbyte desc)### tt1
inner join (select dvid, intfname, unnest(tags) as tag from $ADOM_INTF_INFO ti) tt2 on
tt1.dvid=tt2.dvid and tt1.srcintf=tt2.intfname group by app, tag order by sentbyte desc
```

Dataset Name	Description	Log Category
aware-Top-Device-Attack-Targets	Top Device Attack Targets	fct-netscan

```
select
  hostname,
  count(*) as total_num
from
```

```
  $log
where
  $filter
  and nullifna(hostname) is not null
  and nullifna(vulnname) is not null
group by
  hostname
order by
  total_num desc
```

Dataset Name	Description	Log Category
aware-Top-Attack-Targets	Top Attack Targets	fct-netscan

```
select
  hostname,
  srcip,
  os,
  vuln_num,
  (
    CASE sevid WHEN 5 THEN & #039;Critical' WHEN 4 THEN 'High' WHEN 3 THEN 'Medium' WHEN '2' THEN
'Info' ELSE 'Low' END) as vulnseverity, sevid as severity_num, left(cve_agg, 512) as cve_agg from
(select hostname, max(srcip) as srcip, string_agg(distinct os1, '/') as os, count(distinct
vulnname) as vuln_num, max((CASE vulnseverity WHEN 'Critical' THEN 5 WHEN 'High' THEN 4 WHEN
'Medium' THEN 3 WHEN 'Info' THEN 2 WHEN 'Low' THEN 1 ELSE 0 END)) as sevid, string_agg(distinct
cve_id, ',') as cve_agg from ###(select hostname, max(deviceip) as srcip, split_part(os, ',', 1)
as os1, vulnname, vulnseverity, vulnid, count(*) as total_num from $log where $filter and nullifna
(vulnname) is not null and nullifna(vulnseverity) is not null group by hostname, os1, vulnname,
vulnseverity, vulnid order by total_num desc)### t1 left join fct_mdata t2 on
t1.vulnid=t2.vid::int group by hostname) t order by severity_num desc, vuln_num desc
```

Dataset Name	Description	Log Category
aware-Threats-By-Severity	Threats by Severity	attack

```
select
  initcap(sev) as severity,
  sum(total_num) as total_num
from
  (
    (
      select
        sev,
        sum(total_num) as total_num
      from
        ###(select crlevel::text as sev, count(*) as total_num from $log-virus where $filter and
nullifna(virus) is not null  and crlevel is not null group by sev order by total_num desc)### t
group by sev) union all (select sev, sum(total_num) as total_num from ###(select severity::text as
sev, count(*) as total_num from $log-attack where $filter and nullifna(attack) is not null and
```

```
severity is not null group by sev order by total_num desc)### t group by sev) union all (select
sev, sum(total_num) as total_num from ###(select apprisk::text as sev, count(*) as total_num from
$log-app-ctrl where $filter and lower(appcat)='botnet' and apprisk is not null group by sev order
by total_num desc)### t group by sev)) t group by severity order by total_num desc
```

Dataset Name	Description	Log Category
aware-Threats-Type-By-Severity	Threats Type by Severity	virus

```
select
  threat_type,
  sum(critical) as critical,
  sum(high) as high,
  sum(medium) as medium,
  sum(low) as low
from
  (
    (
      select
        threat_type,
        sum(critical) as critical,
        sum(high) as high,
        sum(medium) as medium,
        sum(low) as low
      from
        ###(select (case when eventtype='botnet' then 'Botnets' else 'Malware' end) as threat_
type, sum(case when crlevel = 'critical' then 1 else 0 end) as critical, sum(case when crlevel =
'high' then 1 else 0 end) as high, sum(case when crlevel = 'medium' then 1 else 0 end) as medium,
sum(case when crlevel = 'low' then 1 else 0 end) as low from $log-virus where $filter and nullifna
(virus) is not null group by threat_type order by critical desc, high desc, medium desc, low
desc)### t group by threat_type ) union all (select threat_type, sum(critical) as critical, sum
(high) as high, sum(medium) as medium, sum(low) as low from ###(select 'Intrusions' as threat_
type, sum(case when severity = 'critical' then 1 else 0 end) as critical, sum(case when severity =
'high' then 1 else 0 end) as high, sum(case when severity = 'medium' then 1 else 0 end) as medium,
sum(case when severity = 'low' then 1 else 0 end) as low from $log-attack where $filter and
nullifna(attack) is not null group by threat_type order by critical desc, high desc, medium desc,
low desc)### t group by threat_type ) union all (select threat_type, sum(critical) as critical,
sum(high) as high, sum(medium) as medium, sum(low) as low from ###(select 'Botnets' as threat_
type, sum(case when apprisk = 'critical' then 1 else 0 end) as critical, sum(case when apprisk =
'high' then 1 else 0 end) as high, sum(case when apprisk = 'medium' then 1 else 0 end) as medium,
sum(case when apprisk = 'low' then 1 else 0 end) as low from $log-app-ctrl where $filter and lower
(appcat)='botnet' group by threat_type order by critical desc, high desc, medium desc, low
desc)### t group by threat_type )) t group by threat_type
```

Dataset Name	Description	Log Category
aware-Threats-By-Day	Threats by Day	virus

```
select
  daystamp,
  sum(total_num) as total_num
from
  (
    (
      select
        daystamp,
        sum(total_num) as total_num
      from
        ###(select $day_of_week as daystamp, count(*) as total_num from $log-virus where $filter
and nullifna(virus) is not null group by daystamp order by total_num desc)### t group by daystamp)
union all (select daystamp, sum(total_num) as total_num from ###(select $day_of_week as daystamp,
count(*) as total_num from $log-attack where $filter and nullifna(attack) is not null group by
daystamp order by total_num desc)### t group by daystamp) union all (select daystamp, sum(total_
num) as total_num from ###(select $day_of_week as daystamp, count(*) as total_num from $log-app-
ctrl where $filter and lower(appcat)='botnet' group by daystamp order by total_num desc)### t
group by daystamp)) t group by daystamp order by daystamp
```

Dataset Name	Description	Log Category
aware-Threats-By-Day-Radar	Threats by Day	virus

```
select
  daystamp,
  sum(total_num) as total_num
from
  (
    (
      select
        daystamp,
        sum(total_num) as total_num
      from
        ###(select $day_of_week as daystamp, count(*) as total_num from $log-virus where $filter
and nullifna(virus) is not null group by daystamp order by total_num desc)### t group by daystamp)
union all (select daystamp, sum(total_num) as total_num from ###(select $day_of_week as daystamp,
count(*) as total_num from $log-attack where $filter and nullifna(attack) is not null group by
daystamp order by total_num desc)### t group by daystamp) union all (select daystamp, sum(total_
num) as total_num from ###(select $day_of_week as daystamp, count(*) as total_num from $log-app-
ctrl where $filter and lower(appcat)='botnet' group by daystamp order by total_num desc)### t
group by daystamp)) t group by daystamp order by daystamp
```

Dataset Name	Description	Log Category
aware-Count-Of-Malware-Events	Count of Malware Events	virus

```
select
  virus,
  count(*) as total_num
```

```
from
  $log
where
  $filter
  and nullifna(virus) is not null
group by
  virus
order by
  total_num desc
```

Dataset Name	Description	Log Category
aware-Top-Malware-By-Count	Top Malware by Count	app-ctrl

```
select
  virus,
  malware_type,
  risk_level,
  count(distinct victim) as victim,
  count(distinct source) as source,
  sum(total_num) as total_num
from
  (
    (
      select
        virus,
        malware_type,
        risk_level,
        source,
        victim,
        sum(total_num) as total_num
      from
        ###(select app as virus, 'Botnet C&C' as malware_type, apprisk::text as risk_level, (CASE
WHEN direction='incoming' THEN dstip ELSE srcip END) as source, (CASE WHEN direction='incoming'
THEN srcip ELSE dstip END) as victim, count(*) as total_num from $log-app-ctrl where $filter and
lower(appcat)='botnet' and apprisk is not null group by app, malware_type, apprisk, victim, source
order by total_num desc)### t group by virus, malware_type, risk_level, source, victim) union all
(select virus, malware_type, risk_level, source, victim, sum(total_num) as total_num from ###
(select virus, (case when eventtype='botnet' then 'Botnet C&C' else 'Virus' end) as malware_type,
crlevel::text as risk_level, (CASE WHEN direction='incoming' THEN dstip ELSE srcip END) as source,
(CASE WHEN direction='incoming' THEN srcip ELSE dstip END) as victim, count(*) as total_num from
$log-virus where $filter and nullifna(virus) is not null and crlevel is not null group by virus,
malware_type, crlevel, victim, source order by total_num desc)### t group by virus, malware_type,
risk_level, source, victim) union all (select virus, malware_type, risk_level, source, victim, sum
(total_num) as total_num from ###(select attack as virus, (case when eventtype='botnet' then
'Botnet C&C' else 'Virus' end) as malware_type, crlevel::text as risk_level, (CASE WHEN
direction='incoming' THEN dstip ELSE srcip END) as source, (CASE WHEN direction='incoming' THEN
srcip ELSE dstip END) as victim, count(*) as total_num from $log-attack where $filter and
(logflag&16>0) and crlevel is not null group by virus, malware_type, crlevel, victim, source order
by total_num desc)### t group by virus, malware_type, risk_level, source, victim)) t group by
virus, malware_type, risk_level order by total_num desc
```

Dataset Name	Description	Log Category
aware-Top-Failed-Login-Attempts	Top Failed Login Attempts	event

```
select
  `user` as f_user,
  ui,
  dstip,
  count(status) as total_failed
from
  $log
where
  $filter
  and nullifna(`user`) is not null
  and logid_to_int(logid) = 32002
group by
  ui,
  f_user,
  dstip
order by
  total_failed desc
```

Dataset Name	Description	Log Category
aware-Top-Failed-Authentication-Attempts	VPN failed logins	event

```
select
  f_user,
  tunneltype,
  sum(total_num) as total_num
from
  ###(select coalesce(nullifna(`xauthuser`), `user`) as f_user, tunneltype, count(*) as total_num
from $log where $filter and subtype='vpn' and (tunneltype like 'ipsec%' or tunneltype like 'ssl%')
and action in ('ssl-login-fail', 'ipsec-login-fail') and coalesce(nullifna(`xauthuser`), nullifna
(`user`)) is not null group by f_user, tunneltype order by total_num desc)### t group by f_user,
tunneltype order by total_num desc
```

Dataset Name	Description	Log Category
aware-Top-Denied-Connections	Top Denied Connections	traffic

```
select
  coalesce(
    nullifna(`user`),
    ipstr(`srcip`)
  ) as user_src,
  service || & #039;(' || ipstr(srcip) || ')' as interface, dstip, count(*) as total_num from $log
```

```
where $filter and (logflag&1>0) and action = 'deny' group by user_src, interface, dstip order by
total_num desc
```

Dataset Name	Description	Log Category
aware-Failed-Compliance-Checked-By-Device	Failed Compliance Checked by Device	event

```
select
  devid,
  & #039;Failed' as results, count(distinct reason) as total_num from ###(select devid, reason,
count(*) as total_num from $log where $filter and subtype='compliance-check' and result='fail'
group by devid, reason order by total_num desc)### t group by devid, results order by total_num
desc
```

Dataset Name	Description	Log Category
aware-Ioc-Blacklist-Summary	IOC Blacklist Summary	app-ctrl

```
drop
  table if exists tmp_ep_eu_map; create temporary table tmp_ep_eu_map as (
    select
      epid,
      euid
    from
      $ADOM_EPEU_DEVMAP tx
    where
      euid >= 1024
  );
select
  coalesce(
    nullif(
      epname,
      & #039;unknown'), nullifna(ipstr(`srcip`)), 'Unknown') as epname, user_agg, sevid, (CASE
sevid WHEN 5 THEN 'Critical' WHEN 4 THEN 'High' WHEN 3 THEN 'Medium' WHEN '2' THEN 'Info' ELSE
'Low' END) as severity, threats, bl_count as total_bl from (select th1.epid, srcip, sevid, bl_
count, threats from (select epid, srcip, max(verdict)+1 as sevid, sum(bl_count) as bl_count from
((select epid, srcip, day_st as itime, bl_count, verdict, unnest(dvid) as dvid_s from $ADOMTBL_
PLHD_IOC_VERDICT where bl_count>0) union all (select epid, srcip, day_st as itime, bl_count,
verdict, unnest(dvid) as dvid_s from $ADOMTBL_PLHD_INTERIM_IOC_VERDICT where bl_count>0)) tvdt
inner join devtable_ext td on td.dvid = tvdt.dvid_s where $filter and $filter-drilldown and $dev_
filter group by epid, srcip) th1 inner join (select epid, string_agg(name, ',') as threats from
(select * from (select epid, thid from ((select epid, thid, itime, unnest(dvid) as dvid_s from
(select epid, unnest(threatid) as thid, day_st as itime, dvid from $ADOMTBL_PLHD_IOC_VERDICT where
bl_count>0) ta1) union all (select epid, thid, itime, unnest(dvid) as dvid_s from (select epid,
unnest(threatid) as thid, day_st as itime, dvid from $ADOMTBL_PLHD_INTERIM_IOC_VERDICT where  bl_
count>0) ta2)) t inner join devtable_ext td on td.dvid = t.dvid_s where $filter and $filter-
drilldown and $dev_filter group by epid, thid) thr inner join td_threat_name_mdata tm on
tm.id=thr.thid) t group by epid) th2 on th1.epid=th2.epid) t1 left join (select epid, string_agg
```

```
(distinct euname, ',') as user_agg from tmp_ep_eu_map tpu inner join $ADOM_ENDUSER as teu on
tpu.euid=teu.euid group by epid) t2 on t2.epid=t1.epid inner join $ADOM_ENDPOINT as tep on
tep.epid=t1.epid order by total_bl desc, sevid desc
```

Dataset Name	Description	Log Category
aware-ioc-Potential-Breach-By-Day	IOC Potential Breach by Day	app-ctrl

```
select
  number,
  day_st as itime
from
  (
    select
      count(epid) as number,
      to_char(
        from_itime(itime),
        & #039;Day') as day_st from (select epid, day_st as itime, unnest(dvid) as dvid_s from
$ADOMTBL_PLHD_INTERIM_IOC_VERDICT where $filter-drilldown and cs_count>0 union all (select epid,
day_st as itime, unnest(dvid) as dvid_s from $ADOMTBL_PLHD_IOC_VERDICT where $filter-drilldown and
cs_count>0)) t inner join devtable_ext td on td.dvid = t.dvid_s where $filter and $filter-
drilldown group by day_st) tt order by itime
```

Dataset Name	Description	Log Category
aware-ioc-Potential-Breach-By-Day-Bar	IOC Potential Breach by Day	app-ctrl

```
select
  number,
  day_st as itime
from
  (
    select
      count(epid) as number,
      to_char(
        from_itime(itime),
        & #039;Day') as day_st from (select epid, day_st as itime, unnest(dvid) as dvid_s from
$ADOMTBL_PLHD_INTERIM_IOC_VERDICT where $filter-drilldown and cs_count>0 union all (select epid,
day_st as itime, unnest(dvid) as dvid_s from $ADOMTBL_PLHD_IOC_VERDICT where $filter-drilldown and
cs_count>0)) t inner join devtable_ext td on td.dvid = t.dvid_s where $filter and $filter-
drilldown group by day_st) tt order by itime
```

Dataset Name	Description	Log Category
aware-ioc-Suspicion-Summary	IOC Suspicion Summary	app-ctrl

```
select
  coalesce(
    nullif(
      epname,
      & #039;unknown'), nullifna(ipstr(`srcip`)), 'Unknown') as epname, cs_count as total_cs, cs_
score as max_cs, verdict as max_verdict, threats from (select th1.epid, srcip, itime, cs_count,
verdict, cs_score, threats from (select epid, srcip, min(itime) as itime, sum(cs_count) as cs_
count, max(verdict) as verdict, max(cs_score) as cs_score from ((select epid, srcip, day_st as
itime, cs_count, verdict, cs_score, unnest(dvid) as dvid_s from $ADOMTBL_PLHD_IOC_VERDICT where
$filter-drilldown and (bl_count is null or bl_count=0) and cs_count>0) union all (select epid,
srcip, day_st as itime, cs_count, verdict, cs_score, unnest(dvid) as dvid_s from $ADOMTBL_PLHD_
INTERIM_IOC_VERDICT where $filter-drilldown and (bl_count is null or bl_count=0) and cs_count>0))
tvdt inner join devtable_ext td on td.dvid = tvdt.dvid_s where $filter and $filter-drilldown group
by epid, srcip) th1 inner join (select epid, string_agg(name, ',') as threats from (select * from
(select epid, thid from ((select epid, thid, itime, unnest(dvid) as dvid_s from (select epid,
unnest(threatid) as thid, day_st as itime, dvid from $ADOMTBL_PLHD_IOC_VERDICT where (bl_count is
null or bl_count=0) and cs_count>0) ta1) union all (select epid, thid, itime, unnest(dvid) as
dvid_s from (select epid, unnest(threatid) as thid, day_st as itime, dvid from $ADOMTBL_PLHD_
INTERIM_IOC_VERDICT where (bl_count is null or bl_count=0) and cs_count>0) ta2)) tt1 inner join
devtable_ext td on td.dvid = tt1.dvid_s where $filter and $filter-drilldown group by epid, thid)
thr inner join td_threat_name_mdata tm on tm.id=thr.thid) tt2 group by epid) th2 on
th1.epid=th2.epid) t inner join $ADOM_ENDPOINT as tep on tep.epid=t.epid order by max_verdict
desc, max_cs desc, total_cs desc
```

Dataset Name	Description	Log Category
aware-Botnet-IP	Top Source IP Affected by Botnet	virus

```
select
  f_user,
  source,
  string_agg(
    distinct `virus`,
    & #039;,') as virus_agg, count(distinct ipstr(`victim`)) as dstip_cnt, max(action) as action,
sum(total_num) as total_num, min(from_itime(first_seen)) as first_seen, max(from_itime(last_seen))
as last_seen from ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`)) as f_user, virus,
(CASE WHEN direction='incoming' THEN dstip ELSE srcip END) as source, (CASE WHEN
direction='incoming' THEN srcip ELSE dstip END) as victim, max(action) as action, count(*) as
total_num, min(itime) as first_seen, max(itime) as last_seen from $log where $filter and logid in
('0202009248', '0202009249') and virus is not null group by f_user, virus, source, victim order by
total_num desc)### t group by source, f_user order by total_num desc
```

Dataset Name	Description	Log Category
aware-Botnet-Domain	New Botnet Domains	dns

```
select
  botnet,
  count(distinct `qname`) as qname_cnt,
```

```
  count(
    distinct ipstr(`dstip`)
  ) as dnssvr_cnt,
  sum(total_num) as total_num,
  min(
    from_itime(first_seen)
  ) as first_seen,
  max(
    from_itime(last_seen)
  ) as last_seen
from
  ###(select coalesce(`botnetdomain`, ipstr(`botnetip`)) as botnet, qname, dstip, count(*) as
total_num, min(nanosec_to_sec(eventtime)) as first_seen, max(nanosec_to_sec(eventtime)) as last_
seen from $log where $filter and logid in ('1501054601', '1501054600') group by botnet, qname,
dstip order by total_num desc)### t group by botnet order by first_seen desc
```

Dataset Name	Description	Log Category
aware-High-Risk-URL-Category	Category of High Risk URLs	webfilter

```
select
  catdesc,
  string_agg(
    distinct hostname,
    & #039;,') as hostname_agg, max(action) as action, sum(total_num) as total_num, min(from_itime
(first_seen)) as first_seen, max(from_itime(last_seen)) as last_seen from ###(select catdesc,
hostname, max(action) as action, count(*) as total_num, min(itime) as first_seen, max(itime) as
last_seen from $log where $filter and cat in (26, 61, 86, 88, 90, 91, 93) group by catdesc,
hostname order by total_num desc)### t group by catdesc order by total_num desc
```

Dataset Name	Description	Log Category
aware-Malicious-Files	Type of Malicious Files from AV and Sandbox	virus

```
select
  virus,
  left(url_agg, 1000) as url_agg,
  left(filename_agg, 1000) as filename_agg,
  quarskip,
  action,
  from_sandbox,
  total_num,
  first_seen,
  last_seen
from
  (
    select
      virus,
      string_agg(
```

```
        distinct url,
        & #039;<br/>') as url_agg, string_agg(distinct filename, '<br/>') as filename_agg, max
(quarskip) as quarskip, max(action) as action, max(from_sandbox) as from_sandbox, sum(total_num)
as total_num, min(from_itime(first_seen)) as first_seen, max(from_itime(last_seen)) as last_seen
from ###(select virus, url, filename, max(quarskip) as quarskip, max(action) as action, (case when
logid in ('0211009234', '0211009235') then 1 else 0 end) as from_sandbox, count(*) as total_num,
min(itime) as first_seen, max(itime) as last_seen from $log where $filter and virus is not null
and logid in ('0211009234', '0201009235', '0211008192', '0211008193', '0211008194', '0211008195')
group by virus, url, filename, from_sandbox order by total_num desc)### t group by virus) t order
by total_num desc
```

Dataset Name	Description	Log Category
newthing-New-Users	New users	fct-traffic

```
drop
  table if exists rpt_tmptbl_1;
drop
  table if exists rpt_tmptbl_2; create temporary table rpt_tmptbl_1 as
select
  f_user,
  min(start_time) as start_time
from
  ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as f_user, min(dtime) as start_time from
$log where $pre_period $filter group by f_user order by start_time desc)### t group by f_user;
create temporary table rpt_tmptbl_2 as select f_user, min(start_time) as start_time from ###
(select coalesce(nullifna(`user`), ipstr(`srcip`)) as f_user, min(dtime) as start_time from $log
where $filter group by f_user order by start_time desc)### t group by f_user; select f_user, from_
dtime(min(start_time)) as start_time from rpt_tmptbl_2 where f_user is not null and not exists
(select 1 from rpt_tmptbl_1 where rpt_tmptbl_2.f_user=rpt_tmptbl_1.f_user) group by f_user order
by start_time desc
```

Dataset Name	Description	Log Category
newthing-New-Devices	New devices	fct-traffic

```
drop
  table if exists rpt_tmptbl_1;
drop
  table if exists rpt_tmptbl_2; create temporary table rpt_tmptbl_1 as
select
  hostname,
  os,
  srcip,
  fctver
from
  ###(select hostname, os, srcip, fctver from $log where $pre_period $filter and hostname is not
null group by hostname, os, srcip, fctver order by hostname)### t group by hostname, os, srcip,
fctver; create temporary table rpt_tmptbl_2 as select hostname, os, srcip, fctver from ###(select
```

```
hostname, os, srcip, fctver from $log where $filter and hostname is not null group by hostname,
os, srcip, fctver order by hostname)### t group by hostname, os, srcip, fctver; select hostname,
max(fctos_to_devtype(os)) as devtype, string_agg(distinct os, '/') as os_agg, string_agg(distinct
ipstr(srcip), '/') as srcip_agg, string_agg(distinct fctver, '/') as fctver_agg from rpt_tmptbl_2
where not exists (select 1 from rpt_tmptbl_1 where rpt_tmptbl_2.hostname=rpt_tmptbl_1.hostname)
group by hostname order by hostname
```

Dataset Name	Description	Log Category
newthing-New-Software-Installed	New software installed	fct-traffic

```
drop
  table if exists rpt_tmptbl_1;
drop
  table if exists rpt_tmptbl_2; create temporary table rpt_tmptbl_1 as
select
  srcproduct,
  hostname
from
  ###(select srcproduct, hostname from $log where $pre_period $filter and nullifna(srcproduct) is
not null group by srcproduct, hostname order by srcproduct)### t group by srcproduct, hostname;
create temporary table rpt_tmptbl_2 as select srcproduct, hostname from ###(select srcproduct,
hostname from $log where $filter and nullifna(srcproduct) is not null group by srcproduct,
hostname order by srcproduct)### t group by srcproduct, hostname; select srcproduct, string_agg
(distinct hostname, ',') as host_agg from rpt_tmptbl_2 where not exists (select 1 from rpt_tmptbl_
1 where rpt_tmptbl_2.srcproduct=rpt_tmptbl_1.srcproduct) group by srcproduct order by srcproduct
```

Dataset Name	Description	Log Category
newthing-New-Security-Threats	New security threats	virus

```
drop
  table if exists rpt_tmptbl_1;
drop
  table if exists rpt_tmptbl_2; create temporary table rpt_tmptbl_1 as
select
  threat_name,
  cat_id,
  source
from
  (
    (
      select
        threat_name,
        cat_id,
        source,
        sum(total_num) as total_num
      from
        ###(select app as threat_name, 1 as cat_id, (CASE WHEN direction='incoming' THEN dstip
```

```
ELSE srcip END) as source, count(*) as total_num from $log-app-ctrl where $pre_period $filter and
nullifna(app) is not null and lower(appcat)='botnet' group by threat_name, cat_id, source order by
total_num desc)### t group by threat_name, cat_id, source) union all (select threat_name, cat_id,
source, sum(total_num) as total_num from ###(select virus as threat_name, 2 as cat_id, (CASE WHEN
direction='incoming' THEN dstip ELSE srcip END) as source, count(*) as total_num from $log-virus
where $pre_period $filter and nullifna(virus) is not null group by threat_name, cat_id, source
order by total_num desc)### t group by threat_name, cat_id, source) union all (select threat_name,
cat_id, source, sum(total_num) as total_num from ###(select attack as threat_name, 3 as cat_id,
(CASE WHEN direction='incoming' THEN dstip ELSE srcip END) as source, count(*) as total_num from
$log-attack where $pre_period $filter and nullifna(attack) is not null group by threat_name, cat_
id, source order by total_num desc)### t group by threat_name, cat_id, source)) t; create
temporary table rpt_tmptbl_2 as select daystamp, threat_name, cat_id, source from ((select
daystamp, threat_name, cat_id, source from ###(select $DAY_OF_MONTH as daystamp, app as threat_
name, 1 as cat_id, (CASE WHEN direction='incoming' THEN dstip ELSE srcip END) as source from $log-
app-ctrl where $filter and nullifna(app) is not null and lower(appcat)='botnet' group by daystamp,
threat_name, cat_id, source order by daystamp)### t group by daystamp, threat_name, cat_id,
source) union all (select daystamp, threat_name, cat_id, source from ###(select $DAY_OF_MONTH as
daystamp, virus as threat_name, 2 as cat_id, (CASE WHEN direction='incoming' THEN dstip ELSE srcip
END) as source from $log-virus where $filter and nullifna(virus) is not null group by daystamp,
threat_name, cat_id, source order by daystamp)### t group by daystamp, threat_name, cat_id,
source) union all (select daystamp, threat_name, cat_id, source from ###(select $DAY_OF_MONTH as
daystamp, attack as threat_name, 3 as cat_id, (CASE WHEN direction='incoming' THEN dstip ELSE
srcip END) as source from $log-attack where $filter and nullifna(attack) is not null group by
daystamp, threat_name, cat_id, source order by daystamp)### t group by daystamp, threat_name, cat_
id, source)) t; select threat_name, (case cat_id when 1 then 'Botnet' when 2 then 'Malware' when 3
then 'Attack' end) as threat_cat, count(distinct source) as host_num, string_agg(distinct cve,
',') as cve_agg from rpt_tmptbl_2 left join ips_mdata t2 on rpt_tmptbl_2.threat_name=t2.name where
not exists (select 1 from rpt_tmptbl_1 where rpt_tmptbl_2.threat_name=rpt_tmptbl_1.threat_name)
group by threat_name, threat_cat order by host_num desc
```

Dataset Name	Description	Log Category
newthing-dns-Botnet-Domain-IP	New Queried Botnet C&C Domains and IPs	dns

```
drop
  table if exists rpt_tmptbl_1;
drop
  table if exists rpt_tmptbl_2; create temporary table rpt_tmptbl_1 as
select
  domain,
  malware_type,
  action_s as action,
  srcip,
  sevid
from
  ###(select coalesce(botnetdomain, ipstr(botnetip)) as domain, cast('Botnet C&C' as char(32)) as
malware_type, (case when action='block' then 'Blocked' when action='redirect' then 'Redirected'
else 'Passed' end) as action_s, srcip, (CASE WHEN level IN ('critical', 'alert', 'emergency') THEN
5 WHEN level='error' THEN 4 WHEN level='warning' THEN 3 WHEN level='notice' THEN 2 ELSE 1 END) as
sevid, coalesce(botnetdomain, ipstr(botnetip)) as sources_s, count(*) as total_num from $log where
$pre_period $filter and (botnetdomain is not null or botnetip is not null) group by domain,
```

```
action_s, srcip, sevid order by sevid desc)### t group by domain, malware_type, action, srcip,
sevid; create temporary table rpt_tmptbl_2 as select domain, malware_type, action_s as action,
srcip, sevid from ###(select coalesce(botnetdomain, ipstr(botnetip)) as domain, cast('Botnet C&C'
as char(32)) as malware_type, (case when action='block' then 'Blocked' when action='redirect' then
'Redirected' else 'Passed' end) as action_s, srcip, (CASE WHEN level IN ('critical', 'alert',
'emergency') THEN 5 WHEN level='error' THEN 4 WHEN level='warning' THEN 3 WHEN level='notice' THEN
2 ELSE 1 END) as sevid, coalesce(botnetdomain, ipstr(botnetip)) as sources_s, count(*) as total_
num from $log where $filter and (botnetdomain is not null or botnetip is not null) group by
domain, action_s, srcip, sevid order by sevid desc)### t group by domain, malware_type, action,
srcip, sevid; select domain, srcip, sevid, (CASE sevid WHEN 5 THEN 'Critical' WHEN 4 THEN 'High'
WHEN 3 THEN 'Medium' WHEN '2' THEN 'Info' ELSE 'Low' END) as severity from rpt_tmptbl_2 where
(domain is not null and not exists (select 1 from rpt_tmptbl_1 where rpt_tmptbl_2.domain=rpt_
tmptbl_1.domain)) or (srcip is not null and not exists (select 1 from rpt_tmptbl_1 where rpt_
tmptbl_2.srcip=rpt_tmptbl_1.srcip)) group by domain, srcip, sevid order by sevid desc, domain
```

Dataset Name	Description	Log Category
newthing-New-Security-Threats-Timeline	New security threats timeline	virus

```
drop
  table if exists rpt_tmptbl_1;
drop
  table if exists rpt_tmptbl_2; create temporary table rpt_tmptbl_1 as
select
  threat_name,
  cat_id,
  source
from
  (
    (
      select
        threat_name,
        cat_id,
        source,
        sum(total_num) as total_num
      from
        ###(select app as threat_name, 1 as cat_id, (CASE WHEN direction='incoming' THEN dstip
ELSE srcip END) as source, count(*) as total_num from $log-app-ctrl where $pre_period $filter and
nullifna(app) is not null and lower(appcat)='botnet' group by threat_name, cat_id, source order by
total_num desc)### t group by threat_name, cat_id, source) union all (select threat_name, cat_id,
source, sum(total_num) as total_num from ###(select virus as threat_name, 2 as cat_id, (CASE WHEN
direction='incoming' THEN dstip ELSE srcip END) as source, count(*) as total_num from $log-virus
where $pre_period $filter and nullifna(virus) is not null group by threat_name, cat_id, source
order by total_num desc)### t group by threat_name, cat_id, source) union all (select threat_name,
cat_id, source, sum(total_num) as total_num from ###(select attack as threat_name, 3 as cat_id,
(CASE WHEN direction='incoming' THEN dstip ELSE srcip END) as source, count(*) as total_num from
$log-attack where $pre_period $filter and nullifna(attack) is not null group by threat_name, cat_
id, source order by total_num desc)### t group by threat_name, cat_id, source)) t; create
temporary table rpt_tmptbl_2 as select timestamp, threat_name, cat_id, source from ((select
timestamp, threat_name, cat_id, source from ###(select $flex_timestamp as timestamp, app as
```

```
threat_name, 1 as cat_id, (CASE WHEN direction='incoming' THEN dstip ELSE srcip END) as source
from $log-app-ctrl where $filter and nullifna(app) is not null and lower(appcat)='botnet' group by
timestamp, threat_name, cat_id, source order by timestamp)### t group by timestamp, threat_name,
cat_id, source) union all (select timestamp, threat_name, cat_id, source from ###(select $flex_
timestamp as timestamp, virus as threat_name, 2 as cat_id, (CASE WHEN direction='incoming' THEN
dstip ELSE srcip END) as source from $log-virus where $filter and nullifna(virus) is not null
group by timestamp, threat_name, cat_id, source order by timestamp)### t group by timestamp,
threat_name, cat_id, source) union all (select timestamp, threat_name, cat_id, source from ###
(select $flex_timestamp as timestamp, attack as threat_name, 3 as cat_id, (CASE WHEN
direction='incoming' THEN dstip ELSE srcip END) as source from $log-attack where $filter and
nullifna(attack) is not null group by timestamp, threat_name, cat_id, source order by
timestamp)### t group by timestamp, threat_name, cat_id, source)) t; select $flex_datetime
(timestamp) as timescale, count(distinct source) as host_num, (case cat_id when 1 then 'Botnet'
when 2 then 'Malware' when 3 then 'Attack' end) as threat_cat from rpt_tmptbl_2 where not exists
(select 1 from rpt_tmptbl_1 where rpt_tmptbl_2.threat_name=rpt_tmptbl_1.threat_name) group by
timescale, cat_id order by timescale, cat_id
```

Dataset Name	Description	Log Category
newthing-New-Vulnerability	New vulnerabilities	fct-netscan

```
drop
  table if exists rpt_tmptbl_1;
drop
  table if exists rpt_tmptbl_2; create temporary table rpt_tmptbl_1 as
select
  vulnid,
  vulnname,
  vulnseverity,
  vulncat,
  hostname
from
  ###(select vulnid, vulnname, vulnseverity, vulncat, hostname, count(*) as total_num from $log
where $pre_period $filter and nullifna(vulnname) is not null group by vulnid, vulnname,
vulnseverity, vulncat, hostname order by total_num desc)### t group by vulnid, vulnname,
vulnseverity, vulncat, hostname; create temporary table rpt_tmptbl_2 as select vulnid, vulnname,
vulnseverity, vulncat, hostname from ###(select vulnid, vulnname, vulnseverity, vulncat, hostname,
count(*) as total_num from $log where $filter and nullifna(vulnname) is not null group by vulnid,
vulnname, vulnseverity, vulncat, hostname order by total_num desc)### t group by vulnid, vulnname,
vulnseverity, vulncat, hostname; select vulnname, (case when vulnseverity='Critical' then 5 when
vulnseverity='High' then 4 when vulnseverity='Medium' then 3 when vulnseverity='Low' then 2 when
vulnseverity='Info' then 1 else 0 end) as sev, vulnseverity, vulncat, count(distinct hostname) as
host_num, cve_id from rpt_tmptbl_2 t1 left join fct_mdata t2 on t1.vulnid=t2.vid::int where not
exists (select 1 from rpt_tmptbl_1 where t1.vulnid=rpt_tmptbl_1.vulnid) group by vulnname, sev,
vulnseverity, vulncat, cve_id order by sev desc, host_num desc
```

Dataset Name	Description	Log Category
newthing-New-Vulnerability-Graph	New vulnerabilities (Graph)	fct-netscan

```
drop
  table if exists rpt_tmptbl_1;
drop
  table if exists rpt_tmptbl_2; create temporary table rpt_tmptbl_1 as
select
  vulnid,
  vulnname,
  vulnseverity,
  vulncat,
  hostname
from
  ###(select vulnid, vulnname, vulnseverity, vulncat, hostname, count(*) as total_num from $log
where $pre_period $filter and nullifna(vulnname) is not null group by vulnid, vulnname,
vulnseverity, vulncat, hostname order by total_num desc)### t group by vulnid, vulnname,
vulnseverity, vulncat, hostname; create temporary table rpt_tmptbl_2 as select vulnid, vulnname,
vulnseverity, vulncat, hostname from ###(select vulnid, vulnname, vulnseverity, vulncat, hostname,
count(*) as total_num from $log where $filter and nullifna(vulnname) is not null group by vulnid,
vulnname, vulnseverity, vulncat, hostname order by total_num desc)### t group by vulnid, vulnname,
vulnseverity, vulncat, hostname; select vulnseverity, count (distinct vulnid) as vuln_num from
rpt_tmptbl_2 where not exists (select 1 from rpt_tmptbl_1 where rpt_tmptbl_2.vulnid=rpt_tmptbl_
1.vulnid) group by vulnseverity order by (case when vulnseverity='Critical' then 5 when
vulnseverity='High' then 4 when vulnseverity='Medium' then 3 when vulnseverity='Low' then 2 when
vulnseverity='Info' then 1 else 0 end) desc
```

Dataset Name	Description	Log Category
newthing-System-Alerts	System Alerts	local-event

```
select
  from_itime(itime) as timestamp,
  msg
from
  $log
where
  $filter
  and msg is not null
  and level =& #039;critical' order by timestamp desc
```

Dataset Name	Description	Log Category
newthing-Configuration-Changes	Configuration Changes	event

```
select
  `user` as f_user,
  devid,
  from_dtime(dtime) as time_s,
  ui,
  msg
from
```

```
  $log
where
  $filter
  and cfgtid>0
order by
  time_s desc
```

Dataset Name	Description	Log Category
newthing-FortiGate-Upgrades	FortiGate Upgrades	event

```
select
  devid,
  from_dtime(dtime) as time_s,
  info[1] as intf,
  info[2] as prev_ver,
  info[3] as new_ver
from
  (
    select
      devid,
      dtime,
      regexp_matches(
        msg,
        & #039;from ([^ ]+) \\(([^ ]+) -> ([^)]+)\\)') as info from $log where $filter and
action='restore-image') t order by time_s desc
```

Dataset Name	Description	Log Category
newthing-User-Upgrades	User Upgrades	fct-event

```
drop
  table if exists rpt_tmptbl_1;
drop
  table if exists rpt_tmptbl_2; create temporary table rpt_tmptbl_1 as
select
  fgtserial,
  hostname,
  deviceip,
  os,
  dtime
from
  ###(select distinct on (fgtserial, hostname) fgtserial, hostname, deviceip, os, dtime from $log
where $pre_period $filter and hostname is not null order by fgtserial, hostname, dtime desc)### t;
create temporary table rpt_tmptbl_2 as select fgtserial, hostname, deviceip, os, dtime from ###
(select distinct on (fgtserial, hostname) fgtserial, hostname, deviceip, os, dtime from $log where
$filter and hostname is not null order by fgtserial, hostname, dtime desc)### t; select distinct
on (1, 2) t2.fgtserial as devid, t2.hostname, t2.deviceip, t1.os as prev_os, t2.os as cur_os,
from_dtime(t1.dtime) as time_s from rpt_tmptbl_2 t2 inner join rpt_tmptbl_1 t1 on
```

```
t2.fgtserial=t1.fgtserial and t2.hostname=t1.hostname and t2.os!=t1.os order by devid,
t2.hostname, t1.dtime desc
```

Dataset Name	Description	Log Category
GTP-List-of-APN-Used	List of APNs Used	gtp

```
select
  apn,
  from_dtime(
    min(first_seen)
  ) as first_seen,
  from_dtime(
    max(last_seen)
  ) as last_seen
from
  ###(select apn, min(dtime) as first_seen, max(dtime) as last_seen from $log where $filter and
nullifna(apn) is not null group by apn order by last_seen desc)### t group by apn order by last_
seen desc, first_seen
```

Dataset Name	Description	Log Category
GTP-Top-APN-by-Bytes	Top APNs by Bytes	gtp

```
select
  apn,
  sum(
    coalesce(`u-bytes`, 0)
  ) as total_bytes
from
  $log
where
  $filter
  and nullifna(apn) is not null
  and status =& #039;traffic-count' group by apn having sum(coalesce(`u-bytes`, 0))>0 order by
total_bytes desc
```

Dataset Name	Description	Log Category
GTP-Top-APN-by-Duration	Top APNs by Duration	gtp

```
select
  apn,
  sum(
    coalesce(duration, 0)
  ) as total_dura
from
  $log
```

```
where
  $filter
  and nullifna(apn) is not null
  and status =& #039;traffic-count' group by apn having sum(coalesce(duration, 0)) >0 order by
total_dura desc
```

Dataset Name	Description	Log Category
GTP-Top-APN-by-Packets	Top APNs by Number of Packets	gtp

```
select
  apn,
  sum(
    coalesce(`u-pkts`, 0)
  ) as total_num
from
  $log
where
  $filter
  and nullifna(apn) is not null
  and status =& #039;traffic-count' group by apn having sum(coalesce(`u-pkts`, 0))>0 order by
total_num desc
```

Dataset Name	Description	Log Category
Top10-dns-Botnet-Domain-IP	Top Queried Botnet C&C Domains and IPs	dns

```
select
  domain,
  malware_type,
  action,
  count(distinct srcip) as victims,
  count(distinct sources_s) as sources,
  sum(total_num) as total_num
from
  ###(select $flex_timestamp as timestamp, coalesce(botnetdomain, ipstr(botnetip)) as domain,
qname, cast('Botnet C&C' as char(32)) as malware_type, (case when action='block' then 'Blocked'
when action='redirect' then 'Redirected' else 'Passed' end) as action, srcip, coalesce(nullifna
(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, (CASE WHEN level IN ('critical',
'alert', 'emergency') THEN 5 WHEN level='error' THEN 4 WHEN level='warning' THEN 3 WHEN
level='notice' THEN 2 ELSE 1 END) as sevid, coalesce(botnetdomain, ipstr(botnetip)) as sources_s,
count(*) as total_num from $log-dns where $filter and (botnetdomain is not null or botnetip is not
null) group by timestamp, domain, qname, action, srcip, user_src, sevid order by sevid desc)### t
group by domain, malware_type, action order by total_num desc
```

Dataset Name	Description	Log Category
dns-Botnet-Usage	Top Queried Botnet C&C Domains and IPs	dns

```
select
  domain,
  malware_type,
  action,
  count(distinct srcip) as victims,
  count(distinct sources_s) as sources,
  sum(total_num) as total_num
from
  ###(select $flex_timestamp as timestamp, coalesce(botnetdomain, ipstr(botnetip)) as domain,
qname, cast('Botnet C&C' as char(32)) as malware_type, (case when action='block' then 'Blocked'
when action='redirect' then 'Redirected' else 'Passed' end) as action, srcip, coalesce(nullifna
(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, (CASE WHEN level IN ('critical',
'alert', 'emergency') THEN 5 WHEN level='error' THEN 4 WHEN level='warning' THEN 3 WHEN
level='notice' THEN 2 ELSE 1 END) as sevid, coalesce(botnetdomain, ipstr(botnetip)) as sources_s,
count(*) as total_num from $log-dns where $filter and (botnetdomain is not null or botnetip is not
null) group by timestamp, domain, qname, action, srcip, user_src, sevid order by sevid desc)### t
group by domain, malware_type, action order by total_num desc
```

Dataset Name	Description	Log Category
Dns-Detected-Botnet	Top Queried Botnet C&C Domains and IPs	dns

```
select
  domain,
  malware_type,
  action,
  count(distinct srcip) as victims,
  count(distinct sources_s) as sources,
  sum(total_num) as total_num
from
  ###(select $flex_timestamp as timestamp, coalesce(botnetdomain, ipstr(botnetip)) as domain,
qname, cast('Botnet C&C' as char(32)) as malware_type, (case when action='block' then 'Blocked'
when action='redirect' then 'Redirected' else 'Passed' end) as action, srcip, coalesce(nullifna
(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, (CASE WHEN level IN ('critical',
'alert', 'emergency') THEN 5 WHEN level='error' THEN 4 WHEN level='warning' THEN 3 WHEN
level='notice' THEN 2 ELSE 1 END) as sevid, coalesce(botnetdomain, ipstr(botnetip)) as sources_s,
count(*) as total_num from $log-dns where $filter and (botnetdomain is not null or botnetip is not
null) group by timestamp, domain, qname, action, srcip, user_src, sevid order by sevid desc)### t
group by domain, malware_type, action order by total_num desc
```

Dataset Name	Description	Log Category
dns-Botnet-Domain-IP	Queried Botnet C&C Domains and IPs	dns

```
select
  domain,
  srcip,
  sevid,
  (
```

```
    CASE sevid WHEN 5 THEN & #039;Critical' WHEN 4 THEN 'High' WHEN 3 THEN 'Medium' WHEN '2' THEN
'Info' ELSE 'Low' END) as severity from ###(select $flex_timestamp as timestamp, coalesce
(botnetdomain, ipstr(botnetip)) as domain, qname, cast('Botnet C&C' as char(32)) as malware_type,
(case when action='block' then 'Blocked' when action='redirect' then 'Redirected' else 'Passed'
end) as action, srcip, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_
src, (CASE WHEN level IN ('critical', 'alert', 'emergency') THEN 5 WHEN level='error' THEN 4 WHEN
level='warning' THEN 3 WHEN level='notice' THEN 2 ELSE 1 END) as sevid, coalesce(botnetdomain,
ipstr(botnetip)) as sources_s, count(*) as total_num from $log-dns where $filter and (botnetdomain
is not null or botnetip is not null) group by timestamp, domain, qname, action, srcip, user_src,
sevid order by sevid desc)### t group by domain, srcip, sevid order by sevid desc, domain
```

Dataset Name	Description	Log Category
dns-High-Risk-Source	High Risk Sources	dns

```
select
  srcip,
  sum(total_num) as total_num,
  sum(
    case when sevid = 5 then total_num else 0 end
  ) as num_cri,
  sum(
    case when sevid = 4 then total_num else 0 end
  ) as num_hig,
  sum(
    case when sevid = 3 then total_num else 0 end
  ) as num_med
from
  ###(select srcip, (CASE WHEN level IN ('critical', 'alert', 'emergency') THEN 5 WHEN
level='error' THEN 4 WHEN level='warning' THEN 3 WHEN level='notice' THEN 2 ELSE 1 END) as sevid,
count(*) as total_num from $log where $filter and srcip is not null group by srcip, sevid order by
total_num desc)### t where sevid>=3 group by srcip having sum(total_num)>0 order by total_num desc
```

Dataset Name	Description	Log Category
dns-DNS-Request-Over-Time	DNS Request Over Time	dns

```
select
  $flex_timescale(timestamp) as timescale,
  sum(
    case when sevid = 5 then total_num else 0 end
  ) as num_cri,
  sum(
    case when sevid = 4 then total_num else 0 end
  ) as num_hig,
  sum(
    case when sevid = 3 then total_num else 0 end
  ) as num_med,
  sum(
```

```
    case when sevid = 2 then total_num else 0 end
  ) as num_inf,
  sum(
    case when sevid = 1 then total_num else 0 end
  ) as num_low
from
  ###(select $flex_timestamp as timestamp, (CASE WHEN level IN ('critical', 'alert', 'emergency')
THEN 5 WHEN level='error' THEN 4 WHEN level='warning' THEN 3 WHEN level='notice' THEN 2 ELSE 1
END) as sevid, count(*) as total_num from $log where $filter group by timestamp, sevid order by
total_num desc)### t group by timescale order by timescale
```

Dataset Name	Description	Log Category
dns-Top-Queried-Domain	Top Queried Domain	dns

```
select
  qname,
  count(*) as total_num
from
  $log
where
  $filter
  and qname is not null
group by
  qname
order by
  total_num desc
```

Dataset Name	Description	Log Category
dns-Top-Domain-Lookup-Failure-Bar	Top Domain Lookup Failures	dns

```
select
  qname,
  srcip,
  count(*) as total_num
from
  $log
where
  $filter
  and qname is not null
  and (
    action =& #039;block' or logid_to_int(logid)=54200) group by qname, srcip order by total_num
desc
```

Dataset Name	Description	Log Category
dns-Top-Domain-Lookup-Failure-Table	Top Domain Lookup Failures	dns

```
select
  qname,
  srcip,
  count(*) as total_num
from
  $log
where
  $filter
  and qname is not null
  and (
    action =& #039;block' or logid_to_int(logid)=54200) group by qname, srcip order by total_num
desc
```

Dataset Name	Description	Log Category
dns-Query-Timeout	Query Timeout	dns

```
select
  srcip,
  qname,
  count(*) as total_num
from
  $log
where
  $filter
  and srcip is not null
  and logid_to_int(logid)= 54200
group by
  qname,
  srcip
order by
  total_num desc
```

Dataset Name	Description	Log Category
dns-Blocked-Query	Blocked Queries	dns

```
select
  srcip,
  msg,
  count(*) as total_num
from
  $log
```

```
where
  $filter
  and srcip is not null
  and action =& #039;block' group by srcip, msg order by total_num desc
```

Dataset Name	Description	Log Category
perf-stat-cpu-usage-drilldown	Fortigate resource detail timeline	event

```
select
  hodex,
  cast(
    sum(cpu_ave)/ count(*) as decimal(6, 0)
  ) as cpu_ave,
  cast(
    sum(mem_ave)/ count(*) as decimal(6, 0)
  ) as mem_ave,
  cast(
    sum(disk_ave)/ count(*) as decimal(6, 0)
  ) as disk_ave,
  cast(
    sum(log_rate)/ count(*) as decimal(10, 2)
  ) as log_rate,
  cast(
    sum(sessions)/ count(*) as decimal(10, 0)
  ) as sessions,
  cast(
    sum(sent_kbps)/ count(*) as decimal(10, 0)
  ) as sent_kbps,
  cast(
    sum(recv_kbps)/ count(*) as decimal(10, 0)
  ) as recv_kbps,
  cast(
    sum(transmit_kbps)/ count(*) as decimal(10, 0)
  ) as transmit_kbps,
  max(mem_peak) as mem_peak,
  max(disk_peak) as disk_peak,
  max(cpu_peak) as cpu_peak,
  max(lograte_peak) as lograte_peak,
  max(session_peak) as session_peak,
  max(transmit_kbps_peak) as transmit_kbps_peak,
  cast(
    sum(cps_ave)/ count(*) as decimal(10, 0)
  ) as cps_ave,
  max(cps_peak) as cps_peak
from
  (
    select
      hodex,
      devid,
      get_fgt_role(devid, slot) as role,
```

```
    cast(
      sum(cpu_ave)/ count(*) as decimal(6, 0)
    ) as cpu_ave,
    cast(
      sum(mem_ave)/ count(*) as decimal(6, 0)
    ) as mem_ave,
    cast(
      sum(disk_ave)/ count(*) as decimal(6, 0)
    ) as disk_ave,
    cast(
      sum(log_rate) as decimal(10, 2)
    ) as log_rate,
    cast(
      sum(sessions) as decimal(10, 0)
    ) as sessions,
    cast(
      sum(sent_kbps) as decimal(10, 0)
    ) as sent_kbps,
    cast(
      sum(recv_kbps) as decimal(10, 0)
    ) as recv_kbps,
    cast(
      sum(transmit_kbps) as decimal(10, 0)
    ) as transmit_kbps,
    max(mem_peak) as mem_peak,
    max(disk_peak) as disk_peak,
    max(cpu_peak) as cpu_peak,
    cast(
      max(lograte_peak) as decimal(10, 2)
    ) as lograte_peak,
    max(session_peak) as session_peak,
    max(transmit_kbps_peak) as transmit_kbps_peak,
    cast(
      sum(cps_ave) as decimal(10, 0)
    ) as cps_ave,
    sum(cps_peak) as cps_peak
  from
    (
      select
        $flex_timescale(timestamp) as hodex,
        devid,
        slot,
        sum(total_cpu)/ sum(count) cpu_ave,
        sum(total_mem)/ sum(count) as mem_ave,
        sum(total_disk)/ sum(count) as disk_ave,
        sum(
          total_trate + total_erate + total_orate
        )/ 100.00 / sum(count) as log_rate,
        sum(totalsession)/ sum(count) as sessions,
        sum(sent)/ sum(count) as sent_kbps,
        sum(recv)/ sum(count) as recv_kbps,
        sum(sent + recv)/ sum(count) as transmit_kbps,
```

```
          max(mem_peak) as mem_peak,
          max(disk_peak) as disk_peak,
          max(cpu_peak) as cpu_peak,
          max(lograte_peak)/ 100.00 as lograte_peak,
          max(session_peak) as session_peak,
          max(transmit_peak) as transmit_kbps_peak,
          sum(cps)/ sum(count) as cps_ave,
          max(cps_peak) as cps_peak
        from
          ###(select $flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_
trate, sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min(itime)
as first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0))
as mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum
(coalesce(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max(coalesce(trate,
0)+coalesce(erate, 0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as
totalsession, max(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part
(bandwidth, '/', 1), '0') as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2),
'0') as integer)) as recv, max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast
(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce
(setuprate, 0)) as cps, max(coalesce(setuprate, 0)) as cps_peak, count(*) as count from $log where
$filter and subtype='system' and action='perf-stats' group by timestamp, devid, slot order by
total_mem desc)### t where $filter-drilldown group by hodex, devid, slot) t group by hodex, devid,
role) t group by hodex order by hodex
```

Dataset Name	Description	Log Category
perf-stat-mem-usage-drilldown	Fortigate resource detail timeline	event

```
select
  hodex,
  cast(
    sum(cpu_ave)/ count(*) as decimal(6, 0)
  ) as cpu_ave,
  cast(
    sum(mem_ave)/ count(*) as decimal(6, 0)
  ) as mem_ave,
  cast(
    sum(disk_ave)/ count(*) as decimal(6, 0)
  ) as disk_ave,
  cast(
    sum(log_rate)/ count(*) as decimal(10, 2)
  ) as log_rate,
  cast(
    sum(sessions)/ count(*) as decimal(10, 0)
  ) as sessions,
  cast(
    sum(sent_kbps)/ count(*) as decimal(10, 0)
  ) as sent_kbps,
  cast(
    sum(recv_kbps)/ count(*) as decimal(10, 0)
  ) as recv_kbps,
```

```
  cast(
    sum(transmit_kbps)/ count(*) as decimal(10, 0)
  ) as transmit_kbps,
  max(mem_peak) as mem_peak,
  max(disk_peak) as disk_peak,
  max(cpu_peak) as cpu_peak,
  max(lograte_peak) as lograte_peak,
  max(session_peak) as session_peak,
  max(transmit_kbps_peak) as transmit_kbps_peak,
  cast(
    sum(cps_ave)/ count(*) as decimal(10, 0)
  ) as cps_ave,
  max(cps_peak) as cps_peak
from
  (
    select
      hodex,
      devid,
      get_fgt_role(devid, slot) as role,
      cast(
        sum(cpu_ave)/ count(*) as decimal(6, 0)
      ) as cpu_ave,
      cast(
        sum(mem_ave)/ count(*) as decimal(6, 0)
      ) as mem_ave,
      cast(
        sum(disk_ave)/ count(*) as decimal(6, 0)
      ) as disk_ave,
      cast(
        sum(log_rate) as decimal(10, 2)
      ) as log_rate,
      cast(
        sum(sessions) as decimal(10, 0)
      ) as sessions,
      cast(
        sum(sent_kbps) as decimal(10, 0)
      ) as sent_kbps,
      cast(
        sum(recv_kbps) as decimal(10, 0)
      ) as recv_kbps,
      cast(
        sum(transmit_kbps) as decimal(10, 0)
      ) as transmit_kbps,
      max(mem_peak) as mem_peak,
      max(disk_peak) as disk_peak,
      max(cpu_peak) as cpu_peak,
      cast(
        max(lograte_peak) as decimal(10, 2)
      ) as lograte_peak,
      max(session_peak) as session_peak,
      max(transmit_kbps_peak) as transmit_kbps_peak,
      cast(
```

```
        sum(cps_ave) as decimal(10, 0)
      ) as cps_ave,
      sum(cps_peak) as cps_peak
    from
      (
        select
          $flex_timescale(timestamp) as hodex,
          devid,
          slot,
          sum(total_cpu)/ sum(count) cpu_ave,
          sum(total_mem)/ sum(count) as mem_ave,
          sum(total_disk)/ sum(count) as disk_ave,
          sum(
            total_trate + total_erate + total_orate
          )/ 100.00 / sum(count) as log_rate,
          sum(totalsession)/ sum(count) as sessions,
          sum(sent)/ sum(count) as sent_kbps,
          sum(recv)/ sum(count) as recv_kbps,
          sum(sent + recv)/ sum(count) as transmit_kbps,
          max(mem_peak) as mem_peak,
          max(disk_peak) as disk_peak,
          max(cpu_peak) as cpu_peak,
          max(lograte_peak)/ 100.00 as lograte_peak,
          max(session_peak) as session_peak,
          max(transmit_peak) as transmit_kbps_peak,
          sum(cps)/ sum(count) as cps_ave,
          max(cps_peak) as cps_peak
        from
          ###(select $flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_
trate, sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min(itime)
as first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0))
as mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum
(coalesce(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max(coalesce(trate,
0)+coalesce(erate, 0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as
totalsession, max(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part
(bandwidth, '/', 1), '0') as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2),
'0') as integer)) as recv, max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast
(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce
(setuprate, 0)) as cps, max(coalesce(setuprate, 0)) as cps_peak, count(*) as count from $log where
$filter and subtype='system' and action='perf-stats' group by timestamp, devid, slot order by
total_mem desc)### t where $filter-drilldown group by hodex, devid, slot) t group by hodex, devid,
role) t group by hodex order by hodex
```

Dataset Name	Description	Log Category
perf-stat-disk-usage-drilldown	Fortigate resource detail timeline	event

```
select
  hodex,
  cast(
    sum(cpu_ave)/ count(*) as decimal(6, 0)
```

```
  ) as cpu_ave,
  cast(
    sum(mem_ave)/ count(*) as decimal(6, 0)
  ) as mem_ave,
  cast(
    sum(disk_ave)/ count(*) as decimal(6, 0)
  ) as disk_ave,
  cast(
    sum(log_rate)/ count(*) as decimal(10, 2)
  ) as log_rate,
  cast(
    sum(sessions)/ count(*) as decimal(10, 0)
  ) as sessions,
  cast(
    sum(sent_kbps)/ count(*) as decimal(10, 0)
  ) as sent_kbps,
  cast(
    sum(recv_kbps)/ count(*) as decimal(10, 0)
  ) as recv_kbps,
  cast(
    sum(transmit_kbps)/ count(*) as decimal(10, 0)
  ) as transmit_kbps,
  max(mem_peak) as mem_peak,
  max(disk_peak) as disk_peak,
  max(cpu_peak) as cpu_peak,
  max(lograte_peak) as lograte_peak,
  max(session_peak) as session_peak,
  max(transmit_kbps_peak) as transmit_kbps_peak,
  cast(
    sum(cps_ave)/ count(*) as decimal(10, 0)
  ) as cps_ave,
  max(cps_peak) as cps_peak
from
  (
    select
      hodex,
      devid,
      get_fgt_role(devid, slot) as role,
      cast(
        sum(cpu_ave)/ count(*) as decimal(6, 0)
      ) as cpu_ave,
      cast(
        sum(mem_ave)/ count(*) as decimal(6, 0)
      ) as mem_ave,
      cast(
        sum(disk_ave)/ count(*) as decimal(6, 0)
      ) as disk_ave,
      cast(
        sum(log_rate) as decimal(10, 2)
      ) as log_rate,
      cast(
        sum(sessions) as decimal(10, 0)
```

```
      ) as sessions,
      cast(
        sum(sent_kbps) as decimal(10, 0)
      ) as sent_kbps,
      cast(
        sum(recv_kbps) as decimal(10, 0)
      ) as recv_kbps,
      cast(
        sum(transmit_kbps) as decimal(10, 0)
      ) as transmit_kbps,
      max(mem_peak) as mem_peak,
      max(disk_peak) as disk_peak,
      max(cpu_peak) as cpu_peak,
      cast(
        max(lograte_peak) as decimal(10, 2)
      ) as lograte_peak,
      max(session_peak) as session_peak,
      max(transmit_kbps_peak) as transmit_kbps_peak,
      cast(
        sum(cps_ave) as decimal(10, 0)
      ) as cps_ave,
      sum(cps_peak) as cps_peak
    from
      (
        select
          $flex_timescale(timestamp) as hodex,
          devid,
          slot,
          sum(total_cpu)/ sum(count) cpu_ave,
          sum(total_mem)/ sum(count) as mem_ave,
          sum(total_disk)/ sum(count) as disk_ave,
          sum(
            total_trate + total_erate + total_orate
          )/ 100.00 / sum(count) as log_rate,
          sum(totalsession)/ sum(count) as sessions,
          sum(sent)/ sum(count) as sent_kbps,
          sum(recv)/ sum(count) as recv_kbps,
          sum(sent + recv)/ sum(count) as transmit_kbps,
          max(mem_peak) as mem_peak,
          max(disk_peak) as disk_peak,
          max(cpu_peak) as cpu_peak,
          max(lograte_peak)/ 100.00 as lograte_peak,
          max(session_peak) as session_peak,
          max(transmit_peak) as transmit_kbps_peak,
          sum(cps)/ sum(count) as cps_ave,
          max(cps_peak) as cps_peak
        from
          ###(select $flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_
trate, sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min(itime)
as first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0))
as mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum
(coalesce(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max(coalesce(trate,
```

```
0)+coalesce(erate, 0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as
totalsession, max(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part
(bandwidth, '/', 1), '0') as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2),
'0') as integer)) as recv, max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast
(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce
(setuprate, 0)) as cps, max(coalesce(setuprate, 0)) as cps_peak, count(*) as count from $log where
$filter and subtype='system' and action='perf-stats' group by timestamp, devid, slot order by
total_mem desc)### t where $filter-drilldown group by hodex, devid, slot) t group by hodex, devid,
role) t group by hodex order by hodex
```

Dataset Name	Description	Log Category
perf-stat-sessions-drilldown	Fortigate resource detail timeline	event

```
select
  hodex,
  cast(
    sum(cpu_ave)/ count(*) as decimal(6, 0)
  ) as cpu_ave,
  cast(
    sum(mem_ave)/ count(*) as decimal(6, 0)
  ) as mem_ave,
  cast(
    sum(disk_ave)/ count(*) as decimal(6, 0)
  ) as disk_ave,
  cast(
    sum(log_rate)/ count(*) as decimal(10, 2)
  ) as log_rate,
  cast(
    sum(sessions)/ count(*) as decimal(10, 0)
  ) as sessions,
  cast(
    sum(sent_kbps)/ count(*) as decimal(10, 0)
  ) as sent_kbps,
  cast(
    sum(recv_kbps)/ count(*) as decimal(10, 0)
  ) as recv_kbps,
  cast(
    sum(transmit_kbps)/ count(*) as decimal(10, 0)
  ) as transmit_kbps,
  max(mem_peak) as mem_peak,
  max(disk_peak) as disk_peak,
  max(cpu_peak) as cpu_peak,
  max(lograte_peak) as lograte_peak,
  max(session_peak) as session_peak,
  max(transmit_kbps_peak) as transmit_kbps_peak,
  cast(
    sum(cps_ave)/ count(*) as decimal(10, 0)
  ) as cps_ave,
  max(cps_peak) as cps_peak
from
```

```
(
  select
    hodex,
    devid,
    get_fgt_role(devid, slot) as role,
    cast(
      sum(cpu_ave)/ count(*) as decimal(6, 0)
    ) as cpu_ave,
    cast(
      sum(mem_ave)/ count(*) as decimal(6, 0)
    ) as mem_ave,
    cast(
      sum(disk_ave)/ count(*) as decimal(6, 0)
    ) as disk_ave,
    cast(
      sum(log_rate) as decimal(10, 2)
    ) as log_rate,
    cast(
      sum(sessions) as decimal(10, 0)
    ) as sessions,
    cast(
      sum(sent_kbps) as decimal(10, 0)
    ) as sent_kbps,
    cast(
      sum(recv_kbps) as decimal(10, 0)
    ) as recv_kbps,
    cast(
      sum(transmit_kbps) as decimal(10, 0)
    ) as transmit_kbps,
    max(mem_peak) as mem_peak,
    max(disk_peak) as disk_peak,
    max(cpu_peak) as cpu_peak,
    cast(
      max(lograte_peak) as decimal(10, 2)
    ) as lograte_peak,
    max(session_peak) as session_peak,
    max(transmit_kbps_peak) as transmit_kbps_peak,
    cast(
      sum(cps_ave) as decimal(10, 0)
    ) as cps_ave,
    sum(cps_peak) as cps_peak
  from
    (
      select
        $flex_timescale(timestamp) as hodex,
        devid,
        slot,
        sum(total_cpu)/ sum(count) cpu_ave,
        sum(total_mem)/ sum(count) as mem_ave,
        sum(total_disk)/ sum(count) as disk_ave,
        sum(
          total_trate + total_erate + total_orate
```

```
          )/ 100.00 / sum(count) as log_rate,
          sum(totalsession)/ sum(count) as sessions,
          sum(sent)/ sum(count) as sent_kbps,
          sum(recv)/ sum(count) as recv_kbps,
          sum(sent + recv)/ sum(count) as transmit_kbps,
          max(mem_peak) as mem_peak,
          max(disk_peak) as disk_peak,
          max(cpu_peak) as cpu_peak,
          max(lograte_peak)/ 100.00 as lograte_peak,
          max(session_peak) as session_peak,
          max(transmit_peak) as transmit_kbps_peak,
          sum(cps)/ sum(count) as cps_ave,
          max(cps_peak) as cps_peak
        from
          ###(select $flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_
trate, sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min(itime)
as first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0))
as mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum
(coalesce(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max(coalesce(trate,
0)+coalesce(erate, 0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as
totalsession, max(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part
(bandwidth, '/', 1), '0') as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2),
'0') as integer)) as recv, max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast
(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce
(setuprate, 0)) as cps, max(coalesce(setuprate, 0)) as cps_peak, count(*) as count from $log where
$filter and subtype='system' and action='perf-stats' group by timestamp, devid, slot order by
total_mem desc)### t where $filter-drilldown group by hodex, devid, slot) t group by hodex, devid,
role) t group by hodex order by hodex
```

Dataset Name	Description	Log Category
perf-stat-lograte-drilldown	Fortigate resource detail timeline	event

```
select
  hodex,
  cast(
    sum(cpu_ave)/ count(*) as decimal(6, 0)
  ) as cpu_ave,
  cast(
    sum(mem_ave)/ count(*) as decimal(6, 0)
  ) as mem_ave,
  cast(
    sum(disk_ave)/ count(*) as decimal(6, 0)
  ) as disk_ave,
  cast(
    sum(log_rate)/ count(*) as decimal(10, 2)
  ) as log_rate,
  cast(
    sum(sessions)/ count(*) as decimal(10, 0)
  ) as sessions,
  cast(
```

```
    sum(sent_kbps)/ count(*) as decimal(10, 0)
  ) as sent_kbps,
  cast(
    sum(recv_kbps)/ count(*) as decimal(10, 0)
  ) as recv_kbps,
  cast(
    sum(transmit_kbps)/ count(*) as decimal(10, 0)
  ) as transmit_kbps,
  max(mem_peak) as mem_peak,
  max(disk_peak) as disk_peak,
  max(cpu_peak) as cpu_peak,
  max(lograte_peak) as lograte_peak,
  max(session_peak) as session_peak,
  max(transmit_kbps_peak) as transmit_kbps_peak,
  cast(
    sum(cps_ave)/ count(*) as decimal(10, 0)
  ) as cps_ave,
  max(cps_peak) as cps_peak
from
  (
    select
      hodex,
      devid,
      get_fgt_role(devid, slot) as role,
      cast(
        sum(cpu_ave)/ count(*) as decimal(6, 0)
      ) as cpu_ave,
      cast(
        sum(mem_ave)/ count(*) as decimal(6, 0)
      ) as mem_ave,
      cast(
        sum(disk_ave)/ count(*) as decimal(6, 0)
      ) as disk_ave,
      cast(
        sum(log_rate) as decimal(10, 2)
      ) as log_rate,
      cast(
        sum(sessions) as decimal(10, 0)
      ) as sessions,
      cast(
        sum(sent_kbps) as decimal(10, 0)
      ) as sent_kbps,
      cast(
        sum(recv_kbps) as decimal(10, 0)
      ) as recv_kbps,
      cast(
        sum(transmit_kbps) as decimal(10, 0)
      ) as transmit_kbps,
      max(mem_peak) as mem_peak,
      max(disk_peak) as disk_peak,
      max(cpu_peak) as cpu_peak,
      cast(
```

```
        max(lograte_peak) as decimal(10, 2)
      ) as lograte_peak,
      max(session_peak) as session_peak,
      max(transmit_kbps_peak) as transmit_kbps_peak,
      cast(
        sum(cps_ave) as decimal(10, 0)
      ) as cps_ave,
      sum(cps_peak) as cps_peak
    from
      (
        select
          $flex_timescale(timestamp) as hodex,
          devid,
          slot,
          sum(total_cpu)/ sum(count) cpu_ave,
          sum(total_mem)/ sum(count) as mem_ave,
          sum(total_disk)/ sum(count) as disk_ave,
          sum(
            total_trate + total_erate + total_orate
          )/ 100.00 / sum(count) as log_rate,
          sum(totalsession)/ sum(count) as sessions,
          sum(sent)/ sum(count) as sent_kbps,
          sum(recv)/ sum(count) as recv_kbps,
          sum(sent + recv)/ sum(count) as transmit_kbps,
          max(mem_peak) as mem_peak,
          max(disk_peak) as disk_peak,
          max(cpu_peak) as cpu_peak,
          max(lograte_peak)/ 100.00 as lograte_peak,
          max(session_peak) as session_peak,
          max(transmit_peak) as transmit_kbps_peak,
          sum(cps)/ sum(count) as cps_ave,
          max(cps_peak) as cps_peak
        from
          ###(select $flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_
trate, sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min(itime)
as first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0))
as mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum
(coalesce(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max(coalesce(trate,
0)+coalesce(erate, 0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as
totalsession, max(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part
(bandwidth, '/', 1), '0') as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2),
'0') as integer)) as recv, max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast
(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce
(setuprate, 0)) as cps, max(coalesce(setuprate, 0)) as cps_peak, count(*) as count from $log where
$filter and subtype='system' and action='perf-stats' group by timestamp, devid, slot order by
total_mem desc)### t where $filter-drilldown group by hodex, devid, slot) t group by hodex, devid,
role) t group by hodex order by hodex
```

Dataset Name	Description	Log Category
perf-stat-connections-drilldown	Fortigate resource detail timeline	event

```
select
  hodex,
  cast(
    sum(cpu_ave)/ count(*) as decimal(6, 0)
  ) as cpu_ave,
  cast(
    sum(mem_ave)/ count(*) as decimal(6, 0)
  ) as mem_ave,
  cast(
    sum(disk_ave)/ count(*) as decimal(6, 0)
  ) as disk_ave,
  cast(
    sum(log_rate)/ count(*) as decimal(10, 2)
  ) as log_rate,
  cast(
    sum(sessions)/ count(*) as decimal(10, 0)
  ) as sessions,
  cast(
    sum(sent_kbps)/ count(*) as decimal(10, 0)
  ) as sent_kbps,
  cast(
    sum(recv_kbps)/ count(*) as decimal(10, 0)
  ) as recv_kbps,
  cast(
    sum(transmit_kbps)/ count(*) as decimal(10, 0)
  ) as transmit_kbps,
  max(mem_peak) as mem_peak,
  max(disk_peak) as disk_peak,
  max(cpu_peak) as cpu_peak,
  max(lograte_peak) as lograte_peak,
  max(session_peak) as session_peak,
  max(transmit_kbps_peak) as transmit_kbps_peak,
  cast(
    sum(cps_ave)/ count(*) as decimal(10, 0)
  ) as cps_ave,
  max(cps_peak) as cps_peak
from
  (
    select
      hodex,
      devid,
      get_fgt_role(devid, slot) as role,
      cast(
        sum(cpu_ave)/ count(*) as decimal(6, 0)
      ) as cpu_ave,
      cast(
        sum(mem_ave)/ count(*) as decimal(6, 0)
      ) as mem_ave,
      cast(
        sum(disk_ave)/ count(*) as decimal(6, 0)
      ) as disk_ave,
      cast(
```

```
        sum(log_rate) as decimal(10, 2)
      ) as log_rate,
      cast(
        sum(sessions) as decimal(10, 0)
      ) as sessions,
      cast(
        sum(sent_kbps) as decimal(10, 0)
      ) as sent_kbps,
      cast(
        sum(recv_kbps) as decimal(10, 0)
      ) as recv_kbps,
      cast(
        sum(transmit_kbps) as decimal(10, 0)
      ) as transmit_kbps,
      max(mem_peak) as mem_peak,
      max(disk_peak) as disk_peak,
      max(cpu_peak) as cpu_peak,
      cast(
        max(lograte_peak) as decimal(10, 2)
      ) as lograte_peak,
      max(session_peak) as session_peak,
      max(transmit_kbps_peak) as transmit_kbps_peak,
      cast(
        sum(cps_ave) as decimal(10, 0)
      ) as cps_ave,
      sum(cps_peak) as cps_peak
    from
      (
        select
          $flex_timescale(timestamp) as hodex,
          devid,
          slot,
          sum(total_cpu)/ sum(count) cpu_ave,
          sum(total_mem)/ sum(count) as mem_ave,
          sum(total_disk)/ sum(count) as disk_ave,
          sum(
            total_trate + total_erate + total_orate
          )/ 100.00 / sum(count) as log_rate,
          sum(totalsession)/ sum(count) as sessions,
          sum(sent)/ sum(count) as sent_kbps,
          sum(recv)/ sum(count) as recv_kbps,
          sum(sent + recv)/ sum(count) as transmit_kbps,
          max(mem_peak) as mem_peak,
          max(disk_peak) as disk_peak,
          max(cpu_peak) as cpu_peak,
          max(lograte_peak)/ 100.00 as lograte_peak,
          max(session_peak) as session_peak,
          max(transmit_peak) as transmit_kbps_peak,
          sum(cps)/ sum(count) as cps_ave,
          max(cps_peak) as cps_peak
        from
          ###(select $flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_
```

```
trate, sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min(itime)
as first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0))
as mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum
(coalesce(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max(coalesce(trate,
0)+coalesce(erate, 0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as
totalsession, max(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part
(bandwidth, '/', 1), '0') as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2),
'0') as integer)) as recv, max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast
(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce
(setuprate, 0)) as cps, max(coalesce(setuprate, 0)) as cps_peak, count(*) as count from $log where
$filter and subtype='system' and action='perf-stats' group by timestamp, devid, slot order by
total_mem desc)### t where $filter-drilldown group by hodex, devid, slot) t group by hodex, devid,
role) t group by hodex order by hodex
```

Dataset Name	Description	Log Category
perf-stat-bandwidth-drilldown	Fortigate resource detail timeline	event

```
select
  hodex,
  cast(
    sum(cpu_ave)/ count(*) as decimal(6, 0)
  ) as cpu_ave,
  cast(
    sum(mem_ave)/ count(*) as decimal(6, 0)
  ) as mem_ave,
  cast(
    sum(disk_ave)/ count(*) as decimal(6, 0)
  ) as disk_ave,
  cast(
    sum(log_rate)/ count(*) as decimal(10, 2)
  ) as log_rate,
  cast(
    sum(sessions)/ count(*) as decimal(10, 0)
  ) as sessions,
  cast(
    sum(sent_kbps)/ count(*) as decimal(10, 0)
  ) as sent_kbps,
  cast(
    sum(recv_kbps)/ count(*) as decimal(10, 0)
  ) as recv_kbps,
  cast(
    sum(transmit_kbps)/ count(*) as decimal(10, 0)
  ) as transmit_kbps,
  max(mem_peak) as mem_peak,
  max(disk_peak) as disk_peak,
  max(cpu_peak) as cpu_peak,
  max(lograte_peak) as lograte_peak,
  max(session_peak) as session_peak,
  max(transmit_kbps_peak) as transmit_kbps_peak,
  cast(
```

```
    sum(cps_ave)/ count(*) as decimal(10, 0)
  ) as cps_ave,
  max(cps_peak) as cps_peak
from
  (
    select
      hodex,
      devid,
      get_fgt_role(devid, slot) as role,
      cast(
        sum(cpu_ave)/ count(*) as decimal(6, 0)
      ) as cpu_ave,
      cast(
        sum(mem_ave)/ count(*) as decimal(6, 0)
      ) as mem_ave,
      cast(
        sum(disk_ave)/ count(*) as decimal(6, 0)
      ) as disk_ave,
      cast(
        sum(log_rate) as decimal(10, 2)
      ) as log_rate,
      cast(
        sum(sessions) as decimal(10, 0)
      ) as sessions,
      cast(
        sum(sent_kbps) as decimal(10, 0)
      ) as sent_kbps,
      cast(
        sum(recv_kbps) as decimal(10, 0)
      ) as recv_kbps,
      cast(
        sum(transmit_kbps) as decimal(10, 0)
      ) as transmit_kbps,
      max(mem_peak) as mem_peak,
      max(disk_peak) as disk_peak,
      max(cpu_peak) as cpu_peak,
      cast(
        max(lograte_peak) as decimal(10, 2)
      ) as lograte_peak,
      max(session_peak) as session_peak,
      max(transmit_kbps_peak) as transmit_kbps_peak,
      cast(
        sum(cps_ave) as decimal(10, 0)
      ) as cps_ave,
      sum(cps_peak) as cps_peak
    from
      (
        select
          $flex_timescale(timestamp) as hodex,
          devid,
          slot,
          sum(total_cpu)/ sum(count) cpu_ave,
```

```
          sum(total_mem)/ sum(count) as mem_ave,
          sum(total_disk)/ sum(count) as disk_ave,
          sum(
            total_trate + total_erate + total_orate
          )/ 100.00 / sum(count) as log_rate,
          sum(totalsession)/ sum(count) as sessions,
          sum(sent)/ sum(count) as sent_kbps,
          sum(recv)/ sum(count) as recv_kbps,
          sum(sent + recv)/ sum(count) as transmit_kbps,
          max(mem_peak) as mem_peak,
          max(disk_peak) as disk_peak,
          max(cpu_peak) as cpu_peak,
          max(lograte_peak)/ 100.00 as lograte_peak,
          max(session_peak) as session_peak,
          max(transmit_peak) as transmit_kbps_peak,
          sum(cps)/ sum(count) as cps_ave,
          max(cps_peak) as cps_peak
        from
          ###(select $flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_
trate, sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min(itime)
as first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0))
as mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum
(coalesce(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max(coalesce(trate,
0)+coalesce(erate, 0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as
totalsession, max(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part
(bandwidth, '/', 1), '0') as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2),
'0') as integer)) as recv, max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast
(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce
(setuprate, 0)) as cps, max(coalesce(setuprate, 0)) as cps_peak, count(*) as count from $log where
$filter and subtype='system' and action='perf-stats' group by timestamp, devid, slot order by
total_mem desc)### t where $filter-drilldown group by hodex, devid, slot) t group by hodex, devid,
role) t group by hodex order by hodex
```

Dataset Name	Description	Log Category
perf-stat-usage-summary-average	Fortigate resource summary view	event

```
select
  devid,
  get_fgt_role(devid, slot) as role,
  cast(
    sum(cpu_ave)/ count(*) as decimal(6, 0)
  ) as cpu_ave,
  cast(
    sum(mem_ave)/ count(*) as decimal(6, 0)
  ) as mem_ave,
  cast(
    sum(disk_ave)/ count(*) as decimal(6, 0)
  ) as disk_ave,
  cast(
    sum(log_rate) as decimal(10, 2)
```

```
  ) as log_rate,
  cast(
    sum(sessions) as decimal(10, 0)
  ) as sessions,
  cast(
    sum(sent_kbps) as decimal(10, 0)
  ) as sent_kbps,
  cast(
    sum(recv_kbps) as decimal(10, 0)
  ) as recv_kbps,
  cast(
    sum(transmit_kbps) as decimal(10, 0)
  ) as transmit_kbps,
  max(mem_peak) as mem_peak,
  max(disk_peak) as disk_peak,
  max(cpu_peak) as cpu_peak,
  cast(
    max(lograte_peak) as decimal(10, 2)
  ) as lograte_peak,
  max(session_peak) as session_peak,
  max(transmit_kbps_peak) as transmit_kbps_peak
from
  (
    select
      devid,
      slot,
      sum(total_cpu)/ sum(count) as cpu_ave,
      sum(total_mem)/ sum(count) as mem_ave,
      sum(total_disk)/ sum(count) as disk_ave,
      sum(
        total_trate + total_erate + total_orate
      )/ 100.00 / sum(count) as log_rate,
      sum(totalsession)/ sum(count) as sessions,
      sum(sent)/ sum(count) as sent_kbps,
      sum(recv)/ sum(count) as recv_kbps,
      sum(sent + recv)/ sum(count) as transmit_kbps,
      max(mem_peak) as mem_peak,
      max(disk_peak) as disk_peak,
      max(cpu_peak) as cpu_peak,
      max(lograte_peak)/ 100.00 as lograte_peak,
      max(session_peak) as session_peak,
      max(transmit_peak) as transmit_kbps_peak
    from
      ###(select $flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_
trate, sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min(itime)
as first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0))
as mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum
(coalesce(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max(coalesce(trate,
0)+coalesce(erate, 0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as
totalsession, max(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part
(bandwidth, '/', 1), '0') as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2),
'0') as integer)) as recv, max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast
```

```
(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce
(setuprate, 0)) as cps, max(coalesce(setuprate, 0)) as cps_peak, count(*) as count from $log where
$filter and subtype='system' and action='perf-stats' group by timestamp, devid, slot order by
total_mem desc)### t group by devid, slot) t group by devid, role order by devid, role
```

Dataset Name	Description	Log Category
perf-stat-usage-summary-peak	Fortigate resource summary view	event

```
select
  devid,
  get_fgt_role(devid, slot) as role,
  cast(
    sum(cpu_ave)/ count(*) as decimal(6, 0)
  ) as cpu_ave,
  cast(
    sum(mem_ave)/ count(*) as decimal(6, 0)
  ) as mem_ave,
  cast(
    sum(disk_ave)/ count(*) as decimal(6, 0)
  ) as disk_ave,
  cast(
    sum(log_rate) as decimal(10, 2)
  ) as log_rate,
  cast(
    sum(sessions) as decimal(10, 0)
  ) as sessions,
  cast(
    sum(sent_kbps) as decimal(10, 0)
  ) as sent_kbps,
  cast(
    sum(recv_kbps) as decimal(10, 0)
  ) as recv_kbps,
  cast(
    sum(transmit_kbps) as decimal(10, 0)
  ) as transmit_kbps,
  max(mem_peak) as mem_peak,
  max(disk_peak) as disk_peak,
  max(cpu_peak) as cpu_peak,
  cast(
    max(lograte_peak) as decimal(10, 2)
  ) as lograte_peak,
  max(session_peak) as session_peak,
  max(transmit_kbps_peak) as transmit_kbps_peak
from
  (
    select
      devid,
      slot,
      sum(total_cpu)/ sum(count) as cpu_ave,
      sum(total_mem)/ sum(count) as mem_ave,
```

```
      sum(total_disk)/ sum(count) as disk_ave,
      sum(
        total_trate + total_erate + total_orate
      )/ 100.00 / sum(count) as log_rate,
      sum(totalsession)/ sum(count) as sessions,
      sum(sent)/ sum(count) as sent_kbps,
      sum(recv)/ sum(count) as recv_kbps,
      sum(sent + recv)/ sum(count) as transmit_kbps,
      max(mem_peak) as mem_peak,
      max(disk_peak) as disk_peak,
      max(cpu_peak) as cpu_peak,
      max(lograte_peak)/ 100.00 as lograte_peak,
      max(session_peak) as session_peak,
      max(transmit_peak) as transmit_kbps_peak
    from
      ###(select $flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_
trate, sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min(itime)
as first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0))
as mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum
(coalesce(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max(coalesce(trate,
0)+coalesce(erate, 0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as
totalsession, max(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part
(bandwidth, '/', 1), '0') as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2),
'0') as integer)) as recv, max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast
(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce
(setuprate, 0)) as cps, max(coalesce(setuprate, 0)) as cps_peak, count(*) as count from $log where
$filter and subtype='system' and action='perf-stats' group by timestamp, devid, slot order by
total_mem desc)### t group by devid, slot) t group by devid, role order by devid, role
```

Dataset Name	Description	Log Category
perf-stat-usage-details-drilldown-master	Fortigate resource summary view	event

```
select
  devid,
  get_fgt_role(devid, slot) as role,
  cast(
    sum(cpu_ave)/ count(*) as decimal(6, 0)
  ) as cpu_ave,
  cast(
    sum(mem_ave)/ count(*) as decimal(6, 0)
  ) as mem_ave,
  cast(
    sum(disk_ave)/ count(*) as decimal(6, 0)
  ) as disk_ave,
  cast(
    sum(log_rate) as decimal(10, 2)
  ) as log_rate,
  cast(
    sum(sessions) as decimal(10, 0)
```

```
  ) as sessions,
  cast(
    sum(sent_kbps) as decimal(10, 0)
  ) as sent_kbps,
  cast(
    sum(recv_kbps) as decimal(10, 0)
  ) as recv_kbps,
  cast(
    sum(transmit_kbps) as decimal(10, 0)
  ) as transmit_kbps,
  max(mem_peak) as mem_peak,
  max(disk_peak) as disk_peak,
  max(cpu_peak) as cpu_peak,
  cast(
    max(lograte_peak) as decimal(10, 2)
  ) as lograte_peak,
  max(session_peak) as session_peak,
  max(transmit_kbps_peak) as transmit_kbps_peak
from
  (
    select
      devid,
      slot,
      sum(total_cpu)/ sum(count) as cpu_ave,
      sum(total_mem)/ sum(count) as mem_ave,
      sum(total_disk)/ sum(count) as disk_ave,
      sum(
        total_trate + total_erate + total_orate
      )/ 100.00 / sum(count) as log_rate,
      sum(totalsession)/ sum(count) as sessions,
      sum(sent)/ sum(count) as sent_kbps,
      sum(recv)/ sum(count) as recv_kbps,
      sum(sent + recv)/ sum(count) as transmit_kbps,
      max(mem_peak) as mem_peak,
      max(disk_peak) as disk_peak,
      max(cpu_peak) as cpu_peak,
      max(lograte_peak)/ 100.00 as lograte_peak,
      max(session_peak) as session_peak,
      max(transmit_peak) as transmit_kbps_peak
    from
      ###(select $flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_
trate, sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min(itime)
as first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0))
as mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum
(coalesce(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max(coalesce(trate,
0)+coalesce(erate, 0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as
totalsession, max(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part
(bandwidth, '/', 1), '0') as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2),
'0') as integer)) as recv, max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast
(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce
(setuprate, 0)) as cps, max(coalesce(setuprate, 0)) as cps_peak, count(*) as count from $log where
$filter and subtype='system' and action='perf-stats' group by timestamp, devid, slot order by
```

```
total_mem desc)### t group by devid, slot) t group by devid, role order by devid, role
```

Dataset Name	Description	Log Category
incident-Incident-Count-by-Status	Incident status distribution	

```
select
  status,
  sum(cnt) as cnt
from

  /*fabricStart*/
  (
    select
      status,
      count(*) as cnt
    from
      $incident
    where
      $filter - drilldown
    group by
      status
    order by
      status
  )
  /*fabricEnd*/
  t
group by
  status
order by
  status
```

Dataset Name	Description	Log Category
incident-Incident-Count-by-Status-Donut	Incident status distribution	

```
select
  status,
  sum(cnt) as cnt
from

  /*fabricStart*/
  (
    select
      status,
      count(*) as cnt
    from
      $incident
```

```
    where
      $filter - drilldown
    group by
      status
    order by
      status
  )
  /*fabricEnd*/
  t
group by
  status
order by
  status
```

Dataset Name	Description	Log Category
incident-Open-Incident-Count-Timeline	Incident count by status over time	

```
select
  hodex,
  max(num_sta_draft) as num_sta_draft,
  max(num_sta_analysis) as num_sta_analysis,
  max(num_sta_response) as num_sta_response,
  max(num_sta_closed) as num_sta_closed,
  max(num_sta_cancelled) as num_sta_cancelled
from

  /*fabricStart*/
  (
    select
      $flex_timescale(agg_time) as hodex,
      max(num_sta_draft) as num_sta_draft,
      max(num_sta_analysis) as num_sta_analysis,
      max(num_sta_response) as num_sta_response,
      max(num_sta_closed) as num_sta_closed,
      max(num_sta_cancelled) as num_sta_cancelled
    from
      $incident_history
    where
      $filter - drilldown
      and $cust_time_filter(agg_time)
    group by
      hodex
    order by
      hodex
  )
  /*fabricEnd*/
  t
group by
  hodex
```

```
order by
  hodex
```

Dataset Name	Description	Log Category
incident-Closed-Incident-Count-Timeline	Incident count by status over time	

```
select
  hodex,
  max(num_sta_draft) as num_sta_draft,
  max(num_sta_analysis) as num_sta_analysis,
  max(num_sta_response) as num_sta_response,
  max(num_sta_closed) as num_sta_closed,
  max(num_sta_cancelled) as num_sta_cancelled
from

  /*fabricStart*/
  (
    select
      $flex_timescale(agg_time) as hodex,
      max(num_sta_draft) as num_sta_draft,
      max(num_sta_analysis) as num_sta_analysis,
      max(num_sta_response) as num_sta_response,
      max(num_sta_closed) as num_sta_closed,
      max(num_sta_cancelled) as num_sta_cancelled
    from
      $incident_history
    where
      $filter - drilldown
      and $cust_time_filter(agg_time)
    group by
      hodex
    order by
      hodex
  )
  /*fabricEnd*/
  t
group by
  hodex
order by
  hodex
```

Dataset Name	Description	Log Category
Top-10-Interested-Apps-by-Bandwidth	Top Interested Applications by Bandwidth Usage	traffic

```
select
  app,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out,
  sum(sessions) as sessions
from
  ###(select timestamp, user_src, appid, app, appcat, sum(bandwidth) as bandwidth, sum(traffic_in)
as traffic_in, sum(traffic_out) as traffic_out, sum(sessions) as sessions from ###base(/*tag:rpt_
base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid, euid, coalesce
(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service, appid, app,
appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce
(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta,
rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions from $log-
traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by timestamp,
dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat, apprisk, hostname order by
bandwidth desc, sessions desc)base### t where appcat in ('P2P', 'Storage.Backup', 'File.Sharing',
'Video/Audio') group by timestamp, user_src, appid, app, appcat /*SkipSTART*/order by bandwidth
desc/*SkipEND*/)### t group by app having sum(bandwidth)>0 order by bandwidth desc
```

Dataset Name	Description	Log Category
Top-Interested-App-Users-by-Bandwidth	Top Interested Application Users by Bandwidth	traffic

```
select
  user_src,
  sum(bandwidth) as bandwidth
from
  ###(select timestamp, user_src, appid, app, appcat, sum(bandwidth) as bandwidth, sum(traffic_in)
as traffic_in, sum(traffic_out) as traffic_out, sum(sessions) as sessions from ###base(/*tag:rpt_
base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid, euid, coalesce
(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service, appid, app,
appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce
(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta,
rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions from $log-
traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by timestamp,
dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat, apprisk, hostname order by
bandwidth desc, sessions desc)base### t where appcat in ('P2P', 'Storage.Backup', 'File.Sharing',
'Video/Audio') group by timestamp, user_src, appid, app, appcat /*SkipSTART*/order by bandwidth
desc/*SkipEND*/)### t group by user_src having sum(bandwidth)>0 order by bandwidth desc
```

Dataset Name	Description	Log Category
Top-10-Interested-Applications-by-Number-of-Users	Top Applications by number of users	traffic

```
select
  app,
```

```
  count(distinct user_src) as number
from
  ###(select timestamp, user_src, appid, app, appcat, sum(bandwidth) as bandwidth, sum(traffic_in)
as traffic_in, sum(traffic_out) as traffic_out, sum(sessions) as sessions from ###base(/*tag:rpt_
base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid, euid, coalesce
(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service, appid, app,
appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce
(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta,
rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions from $log-
traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by timestamp,
dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat, apprisk, hostname order by
bandwidth desc, sessions desc)base### t where appcat in ('P2P', 'Storage.Backup', 'File.Sharing',
'Video/Audio') group by timestamp, user_src, appid, app, appcat /*SkipSTART*/order by bandwidth
desc/*SkipEND*/)### t group by app order by number desc
```

Dataset Name	Description	Log Category
Top-10-User-by-Session	Top user by session count	traffic

```
select
  user_src,
  sum(sessions) as sessions
from
  ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, count
(*) as sessions from $log where $filter and (logflag&1>0) group by user_src order by sessions
desc)### t group by user_src order by sessions desc
```

Dataset Name	Description	Log Category
Top-10-Interested-Apps-by-Session	Top Interested Applications by Bandwidth Usage	traffic

```
select
  app,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out,
  sum(sessions) as sessions
from
  ###(select timestamp, user_src, appid, app, appcat, sum(bandwidth) as bandwidth, sum(traffic_in)
as traffic_in, sum(traffic_out) as traffic_out, sum(sessions) as sessions from ###base(/*tag:rpt_
base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid, euid, coalesce
(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service, appid, app,
appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce
(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta,
rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions from $log-
traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by timestamp,
dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat, apprisk, hostname order by
bandwidth desc, sessions desc)base### t where appcat in ('P2P', 'Storage.Backup', 'File.Sharing',
```

```
'Video/Audio') group by timestamp, user_src, appid, app, appcat /*SkipSTART*/order by bandwidth
desc/*SkipEND*/)### t group by app having sum(bandwidth)>0 order by bandwidth desc
```

Dataset Name	Description	Log Category
Interested-Applications-by-Risk-Level	Interested Applications by Risk Level	traffic

```
select
  app,
  min(id) as id,
  appcat,
  max(risk) as d_risk,
  (
    case when max(risk)=& #039;5' then 'Critical' when max(risk)='4' then 'High' when max
(risk)='3' then 'Medium' when max(risk)='2' then 'Low' else 'Info' end) as risk_level, sum
(sessions) as sessions, sum(traffic_out) as sent, sum(traffic_in) as received, sum(bandwidth) as
bandwidth from ###(select timestamp, user_src, appid, app, appcat, sum(bandwidth) as bandwidth,
sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_out, sum(sessions) as sessions from
###base(/*tag:rpt_base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid,
euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service,
appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions
from $log-traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by
timestamp, dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat, apprisk,
hostname order by bandwidth desc, sessions desc)base### t where appcat in ('P2P',
'Storage.Backup', 'File.Sharing', 'Video/Audio') group by timestamp, user_src, appid, app, appcat
/*SkipSTART*/order by bandwidth desc/*SkipEND*/)### t1 inner join app_mdata t2 on lower
(t1.app)=lower(t2.name) group by app, appcat order by d_risk desc, bandwidth desc
```

Dataset Name	Description	Log Category
Top-App-Category-by-Bandwidth	Total number of bandwidth consuming applications	traffic

```
select
  appcat,
  sum(bandwidth) as bandwidth
from
  ###(select timestamp, app, appcat, user_src, hostname, sum(traffic_in) as traffic_in, sum
(traffic_out) as traffic_out, sum(bandwidth) as bandwidth, sum(sessions) as sessions from ###base
(/*tag:rpt_base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid, euid,
coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service, appid,
app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce
(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta,
rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions from $log-
traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by timestamp,
dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat, apprisk, hostname order by
bandwidth desc, sessions desc)base### t group by timestamp, app, appcat, user_src, hostname
```

```
/*SkipSTART*/order by bandwidth desc, sessions desc/*SkipEND*/)### t group by appcat order by
bandwidth desc
```

Dataset Name	Description	Log Category
Top-Interested-Apps-by-Number-of-Users	Top Applications by number of users	traffic

```
select
  app,
  count(distinct user_src) as number
from
  ###(select timestamp, user_src, appid, app, appcat, sum(bandwidth) as bandwidth, sum(traffic_in)
as traffic_in, sum(traffic_out) as traffic_out, sum(sessions) as sessions from ###base(/*tag:rpt_
base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid, euid, coalesce
(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service, appid, app,
appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce
(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta,
rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions from $log-
traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by timestamp,
dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat, apprisk, hostname order by
bandwidth desc, sessions desc)base### t where appcat in ('P2P', 'Storage.Backup', 'File.Sharing',
'Video/Audio') group by timestamp, user_src, appid, app, appcat /*SkipSTART*/order by bandwidth
desc/*SkipEND*/)### t group by app order by number desc
```

Dataset Name	Description	Log Category
Top-Interested-App-Users-By-Bandwidth-Timeline	Top Interested Application Users by Bandwidth Timeline	traffic

```
select
  hodex,
  t1.user_src,
  t1.bandwidth
from
  (
    select
      $flex_timescale(timestamp) as hodex,
      user_src,
      sum(bandwidth) as bandwidth
    from
      ###(select timestamp, user_src, appid, app, appcat, sum(bandwidth) as bandwidth, sum
(traffic_in) as traffic_in, sum(traffic_out) as traffic_out, sum(sessions) as sessions from
###base(/*tag:rpt_base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid,
euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service,
appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions
from $log-traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by
```

```
timestamp, dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat, apprisk,
hostname order by bandwidth desc, sessions desc)base### t where appcat in ('P2P',
'Storage.Backup', 'File.Sharing', 'Video/Audio') group by timestamp, user_src, appid, app, appcat
/*SkipSTART*/order by bandwidth desc/*SkipEND*/)### t group by hodex, user_src having sum
(bandwidth)>0 order by hodex) t1 inner join (select user_src, sum(bandwidth) as bandwidth from ###
(select timestamp, user_src, appid, app, appcat, sum(bandwidth) as bandwidth, sum(traffic_in) as
traffic_in, sum(traffic_out) as traffic_out, sum(sessions) as sessions from ###base(/*tag:rpt_
base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid, euid, coalesce
(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service, appid, app,
appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce
(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta,
rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions from $log-
traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by timestamp,
dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat, apprisk, hostname order by
bandwidth desc, sessions desc)base### t where appcat in ('P2P', 'Storage.Backup', 'File.Sharing',
'Video/Audio') group by timestamp, user_src, appid, app, appcat /*SkipSTART*/order by bandwidth
desc/*SkipEND*/)### t group by user_src order by bandwidth desc limit $ddown-top) t2 on t1.user_
src=t2.user_src order by hodex
```

Dataset Name	Description	Log Category
soc-Event-vs-Incident-Today-Trend	Events vs Incidents Today Trend	

```
select
  item,
  num_cur,
  num_pre,
  num_diff
from

  /*fabricStart*/
  (
    select
      & #039;Events' as item, num_cur, num_pre, (num_cur-num_pre) as num_diff from (select (select
count(*) from $event t1 left join devtable_ext t2 on t1.dvid=t2.dvid where $filter-drilldown and
$cust_time_filter(alerttime,TODAY)) as num_cur, (select count(*) from $event t1 left join
devtable_ext t2 on t1.dvid=t2.dvid where $filter-drilldown and $cust_time_filter
(alerttime,YESTERDAY)) as num_pre) t union all select 'Incidents' as item, num_cur, num_pre, (num_
cur-num_pre) as num_diff from (select (select count(*) from $incident where $cust_time_filter
(createtime,TODAY)) as num_cur, (select count(*) from $incident where $cust_time_filter
(createtime,YESTERDAY)) as num_pre) t)/*fabricEnd*/ t order by item
```

Dataset Name	Description	Log Category
soc-Event-vs-Incident-History-Trend	Events vs Incidents History Trend	

```
select
  item,
  num_cur,
  num_pre,
  num_diff
from

  /*fabricStart*/
  (
    select
      & #039;Events' as item, num_cur, num_pre, (num_cur-num_pre) as num_diff from (select (select
count(*) from $event t1 left join devtable_ext t2 on t1.dvid=t2.dvid where $filter-drilldown and
$cust_time_filter(alerttime)) as num_cur, (select count(*) from $event t1 left join devtable_ext
t2 on t1.dvid=t2.dvid where $filter-drilldown and $cust_time_filter(alerttime,LAST_N_PERIOD,1)) as
num_pre) t union all select 'Incidents' as item, num_cur, num_pre, (num_cur-num_pre) as num_diff
from (select (select count(*) from $incident where $cust_time_filter(createtime)) as num_cur,
(select count(*) from $incident where $cust_time_filter(createtime,LAST_N_PERIOD,1)) as num_pre)
t)/*fabricEnd*/ t order by item
```

Dataset Name	Description	Log Category
soc-Event-vs-Incident-Trend	Events vs Incidents Trend	

```
select
  t1.item,
  t1.num_cur as num_today,
  t1.num_pre as num_yesterday,
  t1.num_diff as num_diff1,
  t2.num_cur as num_this_period,
  t2.num_pre as num_last_period,
  t2.num_diff as num_diff2
from

  /*fabricStart*/
  (
    select
      & #039;Events' as item, num_cur, num_pre, (num_cur-num_pre) as num_diff from (select (select
count(*) from $event t1 left join devtable_ext t2 on t1.dvid=t2.dvid where $filter-drilldown and
$cust_time_filter(alerttime,TODAY)) as num_cur, (select count(*) from $event t1 left join
devtable_ext t2 on t1.dvid=t2.dvid where $filter-drilldown and $cust_time_filter
(alerttime,YESTERDAY)) as num_pre) t union all select 'Incidents' as item, num_cur, num_pre, (num_
cur-num_pre) as num_diff from (select (select count(*) from $incident where $cust_time_filter
(createtime,TODAY)) as num_cur, (select count(*) from $incident where $cust_time_filter
(createtime,YESTERDAY)) as num_pre) t)/*fabricEnd*/ t1 full join /*fabricStart*/(select 'Events'
as item, num_cur, num_pre, (num_cur-num_pre) as num_diff from (select (select count(*) from $event
t1 left join devtable_ext t2 on t1.dvid=t2.dvid where $filter-drilldown and $cust_time_filter
(alerttime)) as num_cur, (select count(*) from $event t1 left join devtable_ext t2 on
t1.dvid=t2.dvid where $filter-drilldown and $cust_time_filter(alerttime,LAST_N_PERIOD,1)) as num_
pre) t union all select 'Incidents' as item, num_cur, num_pre, (num_cur-num_pre) as num_diff from
(select (select count(*) from $incident where $cust_time_filter(createtime)) as num_cur, (select
```

```
count(*) from $incident where $cust_time_filter(createtime,LAST_N_PERIOD,1)) as num_pre)
t)/*fabricEnd*/ t2 on t1.item=t2.item order by t1.item
```

Dataset Name	Description	Log Category
soc-Total-Event-by-Severity-History	Total Events by Severity History	

```
select
  dom,
  (
    CASE severity WHEN 0 THEN & #039;Critical' WHEN 1 THEN 'High' WHEN 2 THEN 'Medium' WHEN 3 THEN
'Low' ELSE NULL END) as sev, sum(num_events) as num_events from /*fabricStart*/(select dom, unnest
(agg_sev) as severity, unnest(agg_num) as num_events from (select $DAY_OF_MONTH(agg_time) as dom,
array[0, 1, 2, 3] as agg_sev, array[max(num_sev_critical), max(num_sev_high), max(num_sev_medium),
max(num_sev_low)] as agg_num from $event_history where $filter-drilldown and $cust_time_filter
(agg_time) group by dom order by dom) t)/*fabricEnd*/ t group by dom, severity order by dom,
severity
```

Dataset Name	Description	Log Category
soc-Total-Event-by-Severity-Category	Total Events Count by Severity and Category	

```
select
  sev,
  triggername,
  sum(num_events) as num_events
from

  /*fabricStart*/
  (
    select
      (
        CASE severity WHEN 0 THEN & #039;Critical' WHEN 1 THEN 'High' WHEN 2 THEN 'Medium' WHEN 3
THEN 'Low' ELSE NULL END) as sev, triggername, count(*) as num_events from $event t1 left join
devtable_ext t2 on t1.dvid=t2.dvid where $cust_time_filter(alerttime) and $filter-drilldown group
by severity, triggername order by severity desc, triggername)/*fabricEnd*/ t group by sev,
triggername order by sev desc, triggername
```

Dataset Name	Description	Log Category
soc-Total-Incident-by-Severity	Total Incidents by Severity	

```
select
  severity,
  count(num_inc) as num_inc
from
```

```
  /*fabricStart*/
  (
    select
      severity,
      count(*) as num_inc
    from
      $incident
    where
      $filter - drilldown
    group by
      severity
    order by
      severity
  )
  /*fabricEnd*/
  t
group by
  severity
order by
  severity
```

Dataset Name	Description	Log Category
soc-Total-Event-vs-Incident-History	Total Events vs Incidents History	

```
select
  hodex,
  max(num_event_total) as num_event_total,
  max(num_inc_total) as num_inc_total,
  max(num_event_high) as num_event_high
from

  /*fabricStart*/
  (
    select
      coalesce(t1.hodex, t2.hodex) as hodex,
      coalesce(num_event_total, 0) as num_event_total,
      coalesce(num_inc_total, 0) as num_inc_total,
      coalesce(num_event_high, 0) as num_event_high
    from
      (
        select
          $flex_timescale(agg_time) as hodex,
          max(num_total) as num_event_total,
          max(num_sev_critical + num_sev_high) as num_event_high
        from
          $event_history
        where
          $cust_time_filter(agg_time)
```

```
        group by
          hodex
        order by
          hodex
      ) t1 full
      join (
        select
          $flex_timescale(agg_time) as hodex,
          max(
            num_sev_high + num_sev_medium + num_sev_low
          ) as num_inc_total
        from
          $incident_history
        where
          $cust_time_filter(agg_time)
        group by
          hodex
        order by
          hodex
      ) t2 on t1.hodex = t2.hodex
    order by
      hodex
  )
  /*fabricEnd*/
  t
group by
  hodex
order by
  hodex
```

Dataset Name	Description	Log Category
soc-Incident-by-Severity	Incidents by Severity	

```
select
  severity,
  sum(incnum) as incnum
from

  /*fabricStart*/
  (
    select
      severity,
      count(*) as incnum
    from
      $incident
    where
      $cust_time_filter(createtime)
    group by
      severity
    order by
```

```
      incnum desc
  )
  /*fabricEnd*/
  t
group by
  severity
order by
  incnum desc
```

Dataset Name	Description	Log Category
soc-Incident-by-Status	Incidents by Status	

```
select
  status,
  sum(incnum) as incnum
from

  /*fabricStart*/
  (
    select
      status,
      count(*) as incnum
    from
      $incident
    where
      $cust_time_filter(createtime)
    group by
      status
    order by
      incnum desc
  )
  /*fabricEnd*/
  t
group by
  status
order by
  incnum desc
```

Dataset Name	Description	Log Category
soc-Incident-by-Category-Unresolved	Unresolved Incidents by Category	

```
select
  category,
  count(incnum) as incnum
from
```

```
  /*fabricStart*/
  (
    select
      inc_cat_encode(category) as category,
      count(*) as incnum
    from
      $incident
    where
      $cust_time_filter(createtime)
      and status not in (
        & #039;closed', 'cancelled') group by category order by incnum desc)/*fabricEnd*/ t group
by category order by incnum desc
```

Dataset Name	Description	Log Category
soc-Incident-by-Severity-Unresolved	Unresolved Incidents by Severity	

```
select
  severity,
  sum(incnum) as incnum
from

  /*fabricStart*/
  (
    select
      severity,
      count(*) as incnum
    from
      $incident
    where
      $cust_time_filter(createtime)
      and status not in (
        & #039;closed', 'cancelled') group by severity order by incnum desc)/*fabricEnd*/ t group
by severity order by incnum desc
```

Dataset Name	Description	Log Category
soc-Incident-Timeline-by-Category	Incidents Timeline by Category	

```
select
  hodex,
  max(num_cat1) as num_cat1,
  max(num_cat2) as num_cat2,
  max(num_cat3) as num_cat3,
  max(num_cat4) as num_cat4,
  max(num_cat5) as num_cat5,
  max(num_cat6) as num_cat6
from
```

```
  /*fabricStart*/
  (
    select
      $flex_timescale(agg_time) as hodex,
      max(num_cat_cat1) as num_cat1,
      max(num_cat_cat2) as num_cat2,
      max(num_cat_cat3) as num_cat3,
      max(num_cat_cat4) as num_cat4,
      max(num_cat_cat5) as num_cat5,
      max(num_cat_cat6) as num_cat6
    from
      $incident_history
    where
      $cust_time_filter(agg_time)
    group by
      hodex
    order by
      hodex
  )
  /*fabricEnd*/
  t
group by
  hodex
order by
  hodex
```

Dataset Name	Description	Log Category
soc-Incident-List-Unresolved	List of Unresolved Incidents	

```
select
  incnum,
  timestamp,
  severity,
  status,
  endpoint,
  description
from

  /*fabricStart*/
  (
    select
      incid_to_str(incid) as incnum,
      from_itime(createtime) as timestamp,
      severity,
      status,
      endpoint,
      description
    from
      $incident
```

```
    where
      $cust_time_filter(createtime)
      and status not in (
        & #039;closed', 'cancelled') order by severity desc)/*fabricEnd*/ t order by severity desc
```

Dataset Name	Description	Log Category
fex-RSRQ-timeline	FortiExtender RSRQ timeline	event

```
select
  $flex_timescale(timestamp) as hodex,
  cast(
    sum(rsrq_sum)/ sum(count) as decimal(18, 2)
  ) || & #039;dB' as rsrq from ###(select $flex_timestamp(dtime) as timestamp, sum(to_number(rsrq,
'999999.99')) as rsrq_sum, sum(to_number(sinr, '999999.99')) as sinr_sum, count(*) as count from
$log where $filter and logid='0111046409' group by timestamp order by timestamp desc)### t group
by hodex order by hodex desc
```

Dataset Name	Description	Log Category
fex-SINR-timeline	FortiExtender SINR timeline	event

```
select
  $flex_timescale(timestamp) as hodex,
  cast(
    sum(sinr_sum)/ sum(count) as decimal(18, 0)
  ) || & #039;dB' as sinr from ###(select $flex_timestamp(dtime) as timestamp, sum(to_number(rsrq,
'999999.99')) as rsrq_sum, sum(to_number(sinr, '999999.99')) as sinr_sum, count(*) as count from
$log where $filter and logid='0111046409' group by timestamp order by timestamp desc)### t group
by hodex order by hodex desc
```

Dataset Name	Description	Log Category
fgt-device-monitoring-inventory	FortiGate Device Monitoring Inventory	event

```
select
  devname,
  id_devid,
  ip,
  platform,
  os,
  total_num
from

  /*fabricStart*/
  (
    select
      devname,
```

```
      (
        & #039; ' || devid) as id_devid, ip, platform, os, '1' as total_num from $func-fgt-
inventory as t1 where exists (select 1 from devtable_ext t2 where $dev_filter and
t2.devid=t1.devid) order by devname)/*fabricEnd*/ t
```

Dataset Name	Description	Log Category
fgt-inventory-hardware	FortiGate Monitoring Inventory Hardware	event

```
select
  platform,
  sum(total_num) as total_num
from

  /*fabricStart*/
  (
    select
      platform,
      count(*) as total_num
    from
      $func - fgt - inventory as t1
    where
      exists (
        select
          1
        from
          devtable_ext t2
        where
          $dev_filter
          and t2.devid = t1.devid
      )
    group by
      platform
    order by
      total_num desc
  )
  /*fabricEnd*/
  t
group by
  platform
order by
  total_num desc
```

Dataset Name	Description	Log Category
fgt-inventory-software	FortiGate Monitoring Inventory Software	event

```
select
  sf_name,
```

```
  firmware,
  sum(total_num) as total_num
from

  /*fabricStart*/
  (
    select
      & #039;FortiOS' as sf_name, (platform || ' ' || os) as firmware, count(*) as total_num from
$func-fgt-inventory as t1 where exists (select 1 from devtable_ext t2 where $dev_filter and
t2.devid=t1.devid) group by platform, os order by total_num desc)/*fabricEnd*/ t group by sf_name,
firmware order by total_num desc
```

Dataset Name	Description	Log Category
cup-utilization-timeline-for-each-device	FortiGate cpu utilization timeline	event

```
select
  $flex_timescale(timestamp) as hodex,
  devid,
  cast(
    sum(total_cpu)/ sum(count) as decimal(6, 0)
  ) as cpu_ave,
  cast(
    sum(total_mem)/ sum(count) as decimal(6, 0)
  ) as mem_ave,
  cast(
    sum(total_disk)/ sum(count) as decimal(6, 0)
  ) as disk_ave,
  cast(
    sum(sent)/ sum(count) as decimal(10, 0)
  ) as sent_kbps,
  cast(
    sum(recv)/ sum(count) as decimal(10, 0)
  ) as recv_kbps
from
  ###(select $flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_trate,
sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min(itime) as
first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) as
mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum(coalesce
(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max(coalesce(trate, 0)+coalesce(erate,
0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalsession, max
(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part(bandwidth, '/', 1), '0')
as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as recv,
max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_part
(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max
(coalesce(setuprate, 0)) as cps_peak, count(*) as count from $log where $filter and
subtype='system' and action='perf-stats' group by timestamp, devid, slot order by total_mem
desc)### t where $filter-drilldown group by hodex, devid order by hodex
```

Dataset Name	Description	Log Category
status-timeline-by-device-cpu-utilization	FortiGate cpu summary view	event

```
select
  devid,
  cast(
    sum(total_cpu)/ sum(count) as decimal(6, 0)
  ) as cpu_ave,
  max(cpu_peak) as cpu_peak
from
  ###(select $flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_trate,
sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min(itime) as
first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) as
mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum(coalesce
(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max(coalesce(trate, 0)+coalesce(erate,
0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalsession, max
(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part(bandwidth, '/', 1), '0')
as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as recv,
max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_part
(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max
(coalesce(setuprate, 0)) as cps_peak, count(*) as count from $log where $filter and
subtype='system' and action='perf-stats' group by timestamp, devid, slot order by total_mem
desc)### t group by devid order by cpu_peak desc
```

Dataset Name	Description	Log Category
event-cpu-utilization-dev	FortiGate cpu summary view	event

```
select
  devid,
  cast(
    sum(total_cpu)/ sum(count) as decimal(6, 0)
  ) as cpu_ave,
  max(cpu_peak) as cpu_peak
from
  ###(select $flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_trate,
sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min(itime) as
first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) as
mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum(coalesce
(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max(coalesce(trate, 0)+coalesce(erate,
0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalsession, max
(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part(bandwidth, '/', 1), '0')
as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as recv,
max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_part
(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max
(coalesce(setuprate, 0)) as cps_peak, count(*) as count from $log where $filter and
subtype='system' and action='perf-stats' group by timestamp, devid, slot order by total_mem
desc)### t group by devid order by cpu_peak desc
```

Dataset Name	Description	Log Category
memory-utilization-timeline-for-each-device	FortiGate cpu utilization timeline	event

```
select
  $flex_timescale(timestamp) as hodex,
  devid,
  cast(
    sum(total_cpu)/ sum(count) as decimal(6, 0)
  ) as cpu_ave,
  cast(
    sum(total_mem)/ sum(count) as decimal(6, 0)
  ) as mem_ave,
  cast(
    sum(total_disk)/ sum(count) as decimal(6, 0)
  ) as disk_ave,
  cast(
    sum(sent)/ sum(count) as decimal(10, 0)
  ) as sent_kbps,
  cast(
    sum(recv)/ sum(count) as decimal(10, 0)
  ) as recv_kbps
from
  ###(select $flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_trate,
sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min(itime) as
first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) as
mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum(coalesce
(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max(coalesce(trate, 0)+coalesce(erate,
0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalsession, max
(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part(bandwidth, '/', 1), '0')
as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as recv,
max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_part
(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max
(coalesce(setuprate, 0)) as cps_peak, count(*) as count from $log where $filter and
subtype='system' and action='perf-stats' group by timestamp, devid, slot order by total_mem
desc)### t where $filter-drilldown group by hodex, devid order by hodex
```

Dataset Name	Description	Log Category
status-timeline-by-device-mem-utilization	FortiGate memory summary view	event

```
select
  devid,
  cast(
    sum(total_mem)/ sum(count) as decimal(6, 0)
  ) as mem_ave,
  max(mem_peak) as mem_peak
from
```

```
  ###(select $flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_trate,
sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min(itime) as
first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) as
mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum(coalesce
(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max(coalesce(trate, 0)+coalesce(erate,
0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalsession, max
(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part(bandwidth, '/', 1), '0')
as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as recv,
max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_part
(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max
(coalesce(setuprate, 0)) as cps_peak, count(*) as count from $log where $filter and
subtype='system' and action='perf-stats' group by timestamp, devid, slot order by total_mem
desc)### t group by devid order by mem_peak desc
```

Dataset Name	Description	Log Category
event-mem-utilization-dev	FortiGate memory summary view	event

```
select
  devid,
  cast(
    sum(total_mem)/ sum(count) as decimal(6, 0)
  ) as mem_ave,
  max(mem_peak) as mem_peak
from
  ###(select $flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_trate,
sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min(itime) as
first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) as
mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum(coalesce
(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max(coalesce(trate, 0)+coalesce(erate,
0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalsession, max
(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part(bandwidth, '/', 1), '0')
as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as recv,
max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_part
(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max
(coalesce(setuprate, 0)) as cps_peak, count(*) as count from $log where $filter and
subtype='system' and action='perf-stats' group by timestamp, devid, slot order by total_mem
desc)### t group by devid order by mem_peak desc
```

Dataset Name	Description	Log Category
disk-utilization-timeline-for-each-device	FortiGate cpu utilization timeline	event

```
select
  $flex_timescale(timestamp) as hodex,
  devid,
  cast(
    sum(total_cpu)/ sum(count) as decimal(6, 0)
```

```
  ) as cpu_ave,
  cast(
    sum(total_mem)/ sum(count) as decimal(6, 0)
  ) as mem_ave,
  cast(
    sum(total_disk)/ sum(count) as decimal(6, 0)
  ) as disk_ave,
  cast(
    sum(sent)/ sum(count) as decimal(10, 0)
  ) as sent_kbps,
  cast(
    sum(recv)/ sum(count) as decimal(10, 0)
  ) as recv_kbps
from
  ###(select $flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_trate,
sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min(itime) as
first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) as
mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum(coalesce
(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max(coalesce(trate, 0)+coalesce(erate,
0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalsession, max
(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part(bandwidth, '/', 1), '0')
as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as recv,
max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_part
(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max
(coalesce(setuprate, 0)) as cps_peak, count(*) as count from $log where $filter and
subtype='system' and action='perf-stats' group by timestamp, devid, slot order by total_mem
desc)### t where $filter-drilldown group by hodex, devid order by hodex
```

Dataset Name	Description	Log Category
status-timeline-by-device-disk-utilization	FortiGate disk summary view	event

```
select
  devid,
  cast(
    sum(total_disk)/ sum(count) as decimal(6, 0)
  ) as disk_ave,
  max(disk_peak) as disk_peak
from
  ###(select $flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_trate,
sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min(itime) as
first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) as
mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum(coalesce
(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max(coalesce(trate, 0)+coalesce(erate,
0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalsession, max
(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part(bandwidth, '/', 1), '0')
as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as recv,
max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_part
(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max
(coalesce(setuprate, 0)) as cps_peak, count(*) as count from $log where $filter and
```

```
subtype='system' and action='perf-stats' group by timestamp, devid, slot order by total_mem
desc)### t group by devid order by disk_peak desc
```

Dataset Name	Description	Log Category
event-disk-utilization-dev	FortiGate disk summary view	event

```
select
  devid,
  cast(
    sum(total_disk)/ sum(count) as decimal(6, 0)
  ) as disk_ave,
  max(disk_peak) as disk_peak
from
  ###(select $flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_trate,
sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min(itime) as
first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) as
mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum(coalesce
(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max(coalesce(trate, 0)+coalesce(erate,
0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalsession, max
(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part(bandwidth, '/', 1), '0')
as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as recv,
max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_part
(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max
(coalesce(setuprate, 0)) as cps_peak, count(*) as count from $log where $filter and
subtype='system' and action='perf-stats' group by timestamp, devid, slot order by total_mem
desc)### t group by devid order by disk_peak desc
```

Dataset Name	Description	Log Category
event-total-session-summary	FortiGate Total Sessions	event

```
select
  devid,
  max(session_peak) as max_session,
  cast(
    sum(totalsession)/ sum(count) as decimal(10, 0)
  ) as sessions,
  max(cps_peak) as cps_peak,
  cast(
    sum(cps)/ sum(count) as decimal(10, 0)
  ) as cps_ave
from
  ###(select $flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_trate,
sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min(itime) as
first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) as
mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum(coalesce
(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max(coalesce(trate, 0)+coalesce(erate,
0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalsession, max
```

```
(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part(bandwidth, '/', 1), '0')
as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as recv,
max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_part
(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max
(coalesce(setuprate, 0)) as cps_peak, count(*) as count from $log where $filter and
subtype='system' and action='perf-stats' group by timestamp, devid, slot order by total_mem
desc)### t group by devid order by max_session desc
```

Dataset Name	Description	Log Category
event-session-rate-summary	FortiGate Session Rate	event

```
select
  devid,
  max(cps_peak) as max_rate
from
  ###(select $flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_trate,
sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min(itime) as
first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) as
mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum(coalesce
(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max(coalesce(trate, 0)+coalesce(erate,
0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalsession, max
(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part(bandwidth, '/', 1), '0')
as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as recv,
max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_part
(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max
(coalesce(setuprate, 0)) as cps_peak, count(*) as count from $log where $filter and
subtype='system' and action='perf-stats' group by timestamp, devid, slot order by total_mem
desc)### t group by devid order by max_rate desc
```

Dataset Name	Description	Log Category
event-session-summary-dev	FortiGate Total Sessions	event

```
select
  devid,
  max(session_peak) as max_session,
  cast(
    sum(totalsession)/ sum(count) as decimal(10, 0)
  ) as sessions,
  max(cps_peak) as cps_peak,
  cast(
    sum(cps)/ sum(count) as decimal(10, 0)
  ) as cps_ave
from
  ###(select $flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_trate,
sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min(itime) as
first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) as
mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum(coalesce
```

```
(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max(coalesce(trate, 0)+coalesce(erate,
0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalsession, max
(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part(bandwidth, '/', 1), '0')
as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as recv,
max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_part
(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max
(coalesce(setuprate, 0)) as cps_peak, count(*) as count from $log where $filter and
subtype='system' and action='perf-stats' group by timestamp, devid, slot order by total_mem
desc)### t group by devid order by max_session desc
```

Dataset Name	Description	Log Category
fgt-intf-down-timeline-for-each-device	FortiGate Interface Down Timeline	event

```
select
  $flex_timescale(timestamp) as hodex,
  devid,
  sum(total_num) as total_num
from
  ###(select $flex_timestamp as timestamp, devid, status, count(*) as total_num from $log where
$filter and logid_to_int(logid)=20099 and status='DOWN' group by timestamp, devid, status order by
total_num desc)### t where $filter-drilldown group by hodex, devid order by hodex
```

Dataset Name	Description	Log Category
fgt-intf-down-timeline-by-device	FortiGate Interface Down by Device	event

```
select
  devid,
  status,
  sum(total_num) as total_num
from
  ###(select $flex_timestamp as timestamp, devid, status, count(*) as total_num from $log where
$filter and logid_to_int(logid)=20099 and status='DOWN' group by timestamp, devid, status order by
total_num desc)### t group by devid, status order by total_num desc
```

Dataset Name	Description	Log Category
fgt-intf-down-dev-donut	FortiGate Interface Down by Device	event

```
select
  devid,
  status,
  sum(total_num) as total_num
from
  ###(select $flex_timestamp as timestamp, devid, status, count(*) as total_num from $log where
```

```
$filter and logid_to_int(logid)=20099 and status='DOWN' group by timestamp, devid, status order by
total_num desc)### t group by devid, status order by total_num desc
```

Dataset Name	Description	Log Category
fgt-intf-down-dev-tbl	FortiGate Interface Down by Device	event

```
select
  devid,
  status,
  sum(total_num) as total_num
from
  ###(select $flex_timestamp as timestamp, devid, status, count(*) as total_num from $log where
$filter and logid_to_int(logid)=20099 and status='DOWN' group by timestamp, devid, status order by
total_num desc)### t group by devid, status order by total_num desc
```

Dataset Name	Description	Log Category
intf-sent-timeline-for-each-device	FortiGate cpu utilization timeline	event

```
select
  $flex_timescale(timestamp) as hodex,
  devid,
  cast(
    sum(total_cpu)/ sum(count) as decimal(6, 0)
  ) as cpu_ave,
  cast(
    sum(total_mem)/ sum(count) as decimal(6, 0)
  ) as mem_ave,
  cast(
    sum(total_disk)/ sum(count) as decimal(6, 0)
  ) as disk_ave,
  cast(
    sum(sent)/ sum(count) as decimal(10, 0)
  ) as sent_kbps,
  cast(
    sum(recv)/ sum(count) as decimal(10, 0)
  ) as recv_kbps
from
  ###(select $flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_trate,
sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min(itime) as
first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) as
mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum(coalesce
(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max(coalesce(trate, 0)+coalesce(erate,
0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalsession, max
(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part(bandwidth, '/', 1), '0')
as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as recv,
max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_part
(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max
```

```
(coalesce(setuprate, 0)) as cps_peak, count(*) as count from $log where $filter and
subtype='system' and action='perf-stats' group by timestamp, devid, slot order by total_mem
desc)### t where $filter-drilldown group by hodex, devid order by hodex
```

Dataset Name	Description	Log Category
status-timeline-by-device-intf-sent	FortiGate interface summary view	event

```
select
  devid,
  cast(
    sum(sent)/ sum(count) as decimal(10, 0)
  ) as sent_kbps,
  cast(
    sum(recv)/ sum(count) as decimal(10, 0)
  ) as recv_kbps,
  cast(
    sum(sent + recv)/ sum(count) as decimal(10, 0)
  ) as transmit_kbps,
  max(transmit_peak) as transmit_kbps_peak
from
  ###(select $flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_trate,
sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min(itime) as
first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) as
mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum(coalesce
(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max(coalesce(trate, 0)+coalesce(erate,
0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalsession, max
(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part(bandwidth, '/', 1), '0')
as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as recv,
max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_part
(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max
(coalesce(setuprate, 0)) as cps_peak, count(*) as count from $log where $filter and
subtype='system' and action='perf-stats' group by timestamp, devid, slot order by total_mem
desc)### t group by devid order by transmit_kbps_peak desc
```

Dataset Name	Description	Log Category
intf-recv-timeline-for-each-device	FortiGate cpu utilization timeline	event

```
select
  $flex_timescale(timestamp) as hodex,
  devid,
  cast(
    sum(total_cpu)/ sum(count) as decimal(6, 0)
  ) as cpu_ave,
  cast(
    sum(total_mem)/ sum(count) as decimal(6, 0)
  ) as mem_ave,
  cast(
```

```
    sum(total_disk)/ sum(count) as decimal(6, 0)
  ) as disk_ave,
  cast(
    sum(sent)/ sum(count) as decimal(10, 0)
  ) as sent_kbps,
  cast(
    sum(recv)/ sum(count) as decimal(10, 0)
  ) as recv_kbps
from
  ###(select $flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_trate,
sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min(itime) as
first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) as
mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum(coalesce
(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max(coalesce(trate, 0)+coalesce(erate,
0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalsession, max
(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part(bandwidth, '/', 1), '0')
as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as recv,
max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_part
(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max
(coalesce(setuprate, 0)) as cps_peak, count(*) as count from $log where $filter and
subtype='system' and action='perf-stats' group by timestamp, devid, slot order by total_mem
desc)### t where $filter-drilldown group by hodex, devid order by hodex
```

Dataset Name	Description	Log Category
status-timeline-by-device-intf-recv	FortiGate interface summary view	event

```
select
  devid,
  cast(
    sum(sent)/ sum(count) as decimal(10, 0)
  ) as sent_kbps,
  cast(
    sum(recv)/ sum(count) as decimal(10, 0)
  ) as recv_kbps,
  cast(
    sum(sent + recv)/ sum(count) as decimal(10, 0)
  ) as transmit_kbps,
  max(transmit_peak) as transmit_kbps_peak
from
  ###(select $flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_trate,
sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min(itime) as
first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) as
mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum(coalesce
(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max(coalesce(trate, 0)+coalesce(erate,
0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalsession, max
(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part(bandwidth, '/', 1), '0')
as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as recv,
max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_part
(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max
(coalesce(setuprate, 0)) as cps_peak, count(*) as count from $log where $filter and
```

```
subtype='system' and action='perf-stats' group by timestamp, devid, slot order by total_mem
desc)### t group by devid order by transmit_kbps_peak desc
```

Dataset Name	Description	Log Category
event-intf-summary-dev	FortiGate interface summary view	event

```
select
  devid,
  cast(
    sum(sent)/ sum(count) as decimal(10, 0)
  ) as sent_kbps,
  cast(
    sum(recv)/ sum(count) as decimal(10, 0)
  ) as recv_kbps,
  cast(
    sum(sent + recv)/ sum(count) as decimal(10, 0)
  ) as transmit_kbps,
  max(transmit_peak) as transmit_kbps_peak
from
  ###(select $flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_trate,
sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min(itime) as
first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) as
mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum(coalesce
(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max(coalesce(trate, 0)+coalesce(erate,
0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalsession, max
(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part(bandwidth, '/', 1), '0')
as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as recv,
max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_part
(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max
(coalesce(setuprate, 0)) as cps_peak, count(*) as count from $log where $filter and
subtype='system' and action='perf-stats' group by timestamp, devid, slot order by total_mem
desc)### t group by devid order by transmit_kbps_peak desc
```

Dataset Name	Description	Log Category
fgt-intf-stats-timeline-util-in-each	FortiGate Interface Statistics Timeline	event

```
select
  $flex_timescale(tmstamp) as hodex,
  (
    devname || & #039;:' || intfname) as dev_intf, cast(sum(bps_out)/sum(interval)/1000 as decimal
(10, 0)) as kbps_out_avg, cast(sum(bps_in)/sum(interval)/1000 as decimal(10, 0)) as kbps_in_avg,
cast(sum(util_out)/sum(interval)/100 as decimal(10, 2)) as util_out_avg, cast(sum(util_in)/sum
(interval)/100 as decimal(10, 2)) as util_in_avg from /*fabricStart*/(select $flex_timestamp
(timestamp) as tmstamp, devname, intfname, sum(interval) as interval, sum(sentbps*interval) as
bps_out, sum(rcvdbps*interval) as bps_in, sum(sentutil*interval) as util_out, sum
(rcvdutil*interval) as util_in from $ADOM_INTF_STATS t1 left join devtable_ext t2 on t1.dvid =
```

```
t2.dvid where $filter-drilldown and $cust_time_filter(timestamp) group by tmstamp, devname,
intfname)/*fabricEnd*/ t group by hodex, dev_intf order by hodex
```

Dataset Name	Description	Log Category
fgt-intf-stats-timeline-util-in	FortiGate Interface Received Utilization	event

```
select
  (
    devname || & #039;:' || intfname) as dev_intf, cast(sum(bps_out)/sum(interval)/1000 as decimal
(10, 0)) as kbps_out_avg, cast(sum(bps_in)/sum(interval)/1000 as decimal(10, 0)) as kbps_in_avg,
cast(sum(util_out)/sum(interval)/100 as decimal(10, 2)) as util_out_avg, cast(sum(util_in)/sum
(interval)/100 as decimal(10, 2)) as util_in_avg from /*fabricStart*/(select $flex_timestamp
(timestamp) as tmstamp, devname, intfname, sum(interval) as interval, sum(sentbps*interval) as
bps_out, sum(rcvdbps*interval) as bps_in, sum(sentutil*interval) as util_out, sum
(rcvdutil*interval) as util_in from $ADOM_INTF_STATS t1 left join devtable_ext t2 on t1.dvid =
t2.dvid where $dev_filter and $cust_time_filter(timestamp) group by tmstamp, devname,
intfname)/*fabricEnd*/ t group by dev_intf order by util_in_avg desc, kbps_in_avg desc, kbps_out_
avg desc
```

Dataset Name	Description	Log Category
fgt-intf-stats-timeline-util-out-each	FortiGate Interface Statistics Timeline	event

```
select
  $flex_timescale(tmstamp) as hodex,
  (
    devname || & #039;:' || intfname) as dev_intf, cast(sum(bps_out)/sum(interval)/1000 as decimal
(10, 0)) as kbps_out_avg, cast(sum(bps_in)/sum(interval)/1000 as decimal(10, 0)) as kbps_in_avg,
cast(sum(util_out)/sum(interval)/100 as decimal(10, 2)) as util_out_avg, cast(sum(util_in)/sum
(interval)/100 as decimal(10, 2)) as util_in_avg from /*fabricStart*/(select $flex_timestamp
(timestamp) as tmstamp, devname, intfname, sum(interval) as interval, sum(sentbps*interval) as
bps_out, sum(rcvdbps*interval) as bps_in, sum(sentutil*interval) as util_out, sum
(rcvdutil*interval) as util_in from $ADOM_INTF_STATS t1 left join devtable_ext t2 on t1.dvid =
t2.dvid where $filter-drilldown and $cust_time_filter(timestamp) group by tmstamp, devname,
intfname)/*fabricEnd*/ t group by hodex, dev_intf order by hodex
```

Dataset Name	Description	Log Category
fgt-intf-stats-timeline-util-out	FortiGate Interface Sent Utilization	event

```
select
  (
    devname || & #039;:' || intfname) as dev_intf, cast(sum(bps_out)/sum(interval)/1000 as decimal
(10, 0)) as kbps_out_avg, cast(sum(bps_in)/sum(interval)/1000 as decimal(10, 0)) as kbps_in_avg,
cast(sum(util_out)/sum(interval)/100 as decimal(10, 2)) as util_out_avg, cast(sum(util_in)/sum
(interval)/100 as decimal(10, 2)) as util_in_avg from /*fabricStart*/(select $flex_timestamp
(timestamp) as tmstamp, devname, intfname, sum(interval) as interval, sum(sentbps*interval) as
```

```
bps_out, sum(rcvdbps*interval) as bps_in, sum(sentutil*interval) as util_out, sum
(rcvdutil*interval) as util_in from $ADOM_INTF_STATS t1 left join devtable_ext t2 on t1.dvid =
t2.dvid where $dev_filter and $cust_time_filter(timestamp) group by tmstamp, devname,
intfname)/*fabricEnd*/ t group by dev_intf order by util_out_avg desc, kbps_out_avg desc, kbps_in_
avg desc
```

Dataset Name	Description	Log Category
fgt-intf-stats-timeline-bit-rate-in-each	FortiGate Interface Statistics Timeline	event

```
select
  $flex_timescale(tmstamp) as hodex,
  (
    devname || & #039;:' || intfname) as dev_intf, cast(sum(bps_out)/sum(interval)/1000 as decimal
(10, 0)) as kbps_out_avg, cast(sum(bps_in)/sum(interval)/1000 as decimal(10, 0)) as kbps_in_avg,
cast(sum(util_out)/sum(interval)/100 as decimal(10, 2)) as util_out_avg, cast(sum(util_in)/sum
(interval)/100 as decimal(10, 2)) as util_in_avg from /*fabricStart*/(select $flex_timestamp
(timestamp) as tmstamp, devname, intfname, sum(interval) as interval, sum(sentbps*interval) as
bps_out, sum(rcvdbps*interval) as bps_in, sum(sentutil*interval) as util_out, sum
(rcvdutil*interval) as util_in from $ADOM_INTF_STATS t1 left join devtable_ext t2 on t1.dvid =
t2.dvid where $filter-drilldown and $cust_time_filter(timestamp) group by tmstamp, devname,
intfname)/*fabricEnd*/ t group by hodex, dev_intf order by hodex
```

Dataset Name	Description	Log Category
fgt-intf-stats-timeline-bit-rate-in	FortiGate Interface Received Bit Rate	event

```
select
  (
    devname || & #039;:' || intfname) as dev_intf, cast(sum(bps_out)/sum(interval)/1000 as decimal
(10, 0)) as kbps_out_avg, cast(sum(bps_in)/sum(interval)/1000 as decimal(10, 0)) as kbps_in_avg,
cast(sum(util_out)/sum(interval)/100 as decimal(10, 2)) as util_out_avg, cast(sum(util_in)/sum
(interval)/100 as decimal(10, 2)) as util_in_avg from /*fabricStart*/(select $flex_timestamp
(timestamp) as tmstamp, devname, intfname, sum(interval) as interval, sum(sentbps*interval) as
bps_out, sum(rcvdbps*interval) as bps_in, sum(sentutil*interval) as util_out, sum
(rcvdutil*interval) as util_in from $ADOM_INTF_STATS t1 left join devtable_ext t2 on t1.dvid =
t2.dvid where $dev_filter and $cust_time_filter(timestamp) group by tmstamp, devname,
intfname)/*fabricEnd*/ t group by dev_intf order by kbps_in_avg desc
```

Dataset Name	Description	Log Category
fgt-intf-stats-timeline-bit-rate-out-each	FortiGate Interface Statistics Timeline	event

```
select
  $flex_timescale(tmstamp) as hodex,
  (
```

```
    devname || & #039;:' || intfname) as dev_intf, cast(sum(bps_out)/sum(interval)/1000 as decimal
(10, 0)) as kbps_out_avg, cast(sum(bps_in)/sum(interval)/1000 as decimal(10, 0)) as kbps_in_avg,
cast(sum(util_out)/sum(interval)/100 as decimal(10, 2)) as util_out_avg, cast(sum(util_in)/sum
(interval)/100 as decimal(10, 2)) as util_in_avg from /*fabricStart*/(select $flex_timestamp
(timestamp) as tmstamp, devname, intfname, sum(interval) as interval, sum(sentbps*interval) as
bps_out, sum(rcvdbps*interval) as bps_in, sum(sentutil*interval) as util_out, sum
(rcvdutil*interval) as util_in from $ADOM_INTF_STATS t1 left join devtable_ext t2 on t1.dvid =
t2.dvid where $filter-drilldown and $cust_time_filter(timestamp) group by tmstamp, devname,
intfname)/*fabricEnd*/ t group by hodex, dev_intf order by hodex
```

Dataset Name	Description	Log Category
fgt-intf-stats-timeline-bit-rate-out	FortiGate Interface Sent Bit Rate	event

```
select
  (
    devname || & #039;:' || intfname) as dev_intf, cast(sum(bps_out)/sum(interval)/1000 as decimal
(10, 0)) as kbps_out_avg, cast(sum(bps_in)/sum(interval)/1000 as decimal(10, 0)) as kbps_in_avg,
cast(sum(util_out)/sum(interval)/100 as decimal(10, 2)) as util_out_avg, cast(sum(util_in)/sum
(interval)/100 as decimal(10, 2)) as util_in_avg from /*fabricStart*/(select $flex_timestamp
(timestamp) as tmstamp, devname, intfname, sum(interval) as interval, sum(sentbps*interval) as
bps_out, sum(rcvdbps*interval) as bps_in, sum(sentutil*interval) as util_out, sum
(rcvdutil*interval) as util_in from $ADOM_INTF_STATS t1 left join devtable_ext t2 on t1.dvid =
t2.dvid where $dev_filter and $cust_time_filter(timestamp) group by tmstamp, devname,
intfname)/*fabricEnd*/ t group by dev_intf order by kbps_out_avg desc
```

Dataset Name	Description	Log Category
fgt-intf-stats-summary-view	FortiGate Interface Received Utilization	event

```
select
  (
    devname || & #039;:' || intfname) as dev_intf, cast(sum(bps_out)/sum(interval)/1000 as decimal
(10, 0)) as kbps_out_avg, cast(sum(bps_in)/sum(interval)/1000 as decimal(10, 0)) as kbps_in_avg,
cast(sum(util_out)/sum(interval)/100 as decimal(10, 2)) as util_out_avg, cast(sum(util_in)/sum
(interval)/100 as decimal(10, 2)) as util_in_avg from /*fabricStart*/(select $flex_timestamp
(timestamp) as tmstamp, devname, intfname, sum(interval) as interval, sum(sentbps*interval) as
bps_out, sum(rcvdbps*interval) as bps_in, sum(sentutil*interval) as util_out, sum
(rcvdutil*interval) as util_in from $ADOM_INTF_STATS t1 left join devtable_ext t2 on t1.dvid =
t2.dvid where $dev_filter and $cust_time_filter(timestamp) group by tmstamp, devname,
intfname)/*fabricEnd*/ t group by dev_intf order by util_in_avg desc, kbps_in_avg desc, kbps_out_
avg desc
```

Dataset Name	Description	Log Category
fgt-ha-failure-timeline	FortiGate HA Failure Timeline	event

```
select
  $flex_timescale(timestamp) as hodex,
  count(*) as total_num
from
  ###(select $flex_timestamp as timestamp, dtime, devid, coalesce(nullifna(logdesc), msg) as msg_
desc from $log where $filter and subtype='ha' and logid_to_int(logid) in (35011, 35012, 35013,
37892, 37893, 37897, 37898, 37901, 37902, 37907, 37908) order by dtime desc)### t group by hodex
order by hodex
```

Dataset Name	Description	Log Category
fgt-ha-failure-summary	FortiGate HA Failure Summary	event

```
select
  from_dtime(dtime) as time_s,
  devid,
  msg_desc
from
  ###(select $flex_timestamp as timestamp, dtime, devid, coalesce(nullifna(logdesc), msg) as msg_
desc from $log where $filter and subtype='ha' and logid_to_int(logid) in (35011, 35012, 35013,
37892, 37893, 37897, 37898, 37901, 37902, 37907, 37908) order by dtime desc)### t order by time_s
desc
```

Dataset Name	Description	Log Category
fgt-env-faults-power	FortiGate Power Supply Faults	event

```
select
  time_s,
  devid,
  msg_desc
from
  ###(select from_dtime(dtime) as time_s, devid, coalesce(nullifna(logdesc), msg) as msg_desc,
logid_to_int(logid) as logid from $log where $filter and logid_to_int(logid) in (22105, 22107,
22108, 22109) order by time_s desc)### t where logid in (22105, 22107) order by time_s desc
```

Dataset Name	Description	Log Category
fgt-env-faults-fan	FortiGate Fan Faults	event

```
select
  time_s,
  devid,
  msg_desc
from
  ###(select from_dtime(dtime) as time_s, devid, coalesce(nullifna(logdesc), msg) as msg_desc,
logid_to_int(logid) as logid from $log where $filter and logid_to_int(logid) in (22105, 22107,
22108, 22109) order by time_s desc)### t where logid=22108 order by time_s desc
```

Dataset Name	Description	Log Category
fgt-env-faults-temperature	FortiGate Temperatre Too High	event

```
select
  time_s,
  devid,
  msg_desc
from
  ###(select from_dtime(dtime) as time_s, devid, coalesce(nullifna(logdesc), msg) as msg_desc,
logid_to_int(logid) as logid from $log where $filter and logid_to_int(logid) in (22105, 22107,
22108, 22109) order by time_s desc)### t where logid=22109 order by time_s desc
```

Dataset Name	Description	Log Category
Behaviour-Banned-Application	Bullying Chat Search and Message Logging by Platforms	app-ctrl

```
select
  app,
  count(*) as requests
from
  ###(select filename, app, itime, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr
(`srcip`)) as user_src, `group`, `srcip` from $log where $filter and ($bully_keywords) and (lower
(app) in ('facebook_post', 'facebook_chat', 'twitter_post', 'youtube_video.access', 'gmail_chat',
'gmail_send.message', 'linkedin_post', 'vimeo_video.access', 'google.search_search.phrase',
'bing.search_search.phrase')) order by itime desc)### t group by app order by requests desc
```

Dataset Name	Description	Log Category
Behaviour-Banned-User	Bullying Chat Search and Message Logging by Users	app-ctrl

```
select
  user_src,
  count(*) as requests
from
  ###(select filename, app, itime, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr
(`srcip`)) as user_src, `group`, `srcip` from $log where $filter and ($bully_keywords) and (lower
(app) in ('facebook_post', 'facebook_chat', 'twitter_post', 'youtube_video.access', 'gmail_chat',
'gmail_send.message', 'linkedin_post', 'vimeo_video.access', 'google.search_search.phrase',
'bing.search_search.phrase')) order by itime desc)### t group by user_src order by requests desc
```

Dataset Name	Description	Log Category
Behaviour-Banned-User-Drilldown	Users Bullying Chat Search and Message Logging	app-ctrl

```
select
  user_src,
  filename,
  min(id) as id,
  string_agg(
    distinct app,
    & #039; ') as app_agg, string_agg(distinct from_itime(itime)::text, ' ') as itime_agg, string_
agg(distinct `group`, ' ') as group_agg, string_agg(distinct ipstr(`srcip`), ' ') as srcip_agg,
count(*) as requests from ###(select filename, app, itime, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, `group`, `srcip` from $log where $filter and ($bully_
keywords) and (lower(app) in ('facebook_post', 'facebook_chat', 'twitter_post', 'youtube_
video.access', 'gmail_chat', 'gmail_send.message', 'linkedin_post', 'vimeo_video.access',
'google.search_search.phrase', 'bing.search_search.phrase')) order by itime desc)### t left join
app_mdata t2 on lower(t.app)=lower(t2.name) group by user_src, filename order by requests desc
```

Dataset Name	Description	Log Category
Behaviour-Banned-User-Drilldown-per-App	Users Bullying Chat Search and Message Logging	app-ctrl

```
select
  user_src,
  filename,
  min(id) as id,
  string_agg(
    distinct app,
    & #039; ') as app_agg, string_agg(distinct from_itime(itime)::text, ' ') as itime_agg, string_
agg(distinct `group`, ' ') as group_agg, string_agg(distinct ipstr(`srcip`), ' ') as srcip_agg,
count(*) as requests from ###(select filename, app, itime, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, `group`, `srcip` from $log where $filter and ($bully_
keywords) and (lower(app) in ('facebook_post', 'facebook_chat', 'twitter_post', 'youtube_
video.access', 'gmail_chat', 'gmail_send.message', 'linkedin_post', 'vimeo_video.access',
'google.search_search.phrase', 'bing.search_search.phrase')) order by itime desc)### t left join
app_mdata t2 on lower(t.app)=lower(t2.name) group by user_src, filename order by requests desc
```

Dataset Name	Description	Log Category
behaviour-banned	Bullying Chat Search and Message Logging	app-ctrl

```
select
  filename,
  min(id) as id,
  string_agg(
    distinct app,
    & #039; ') as app_agg, string_agg(distinct from_itime(itime)::text, ' ') as itime_agg, string_
agg(distinct user_src, ', ') as user_agg, string_agg(distinct `group`, ' ') as group_agg, string_
agg(distinct ipstr(`srcip`), ' ') as srcip_agg, count(*) as requests from ###(select filename,
app, itime, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
`group`, `srcip` from $log where $filter and ($bully_keywords) and (lower(app) in ('facebook_
```

```
post', 'facebook_chat', 'twitter_post', 'youtube_video.access', 'gmail_chat', 'gmail_
send.message', 'linkedin_post', 'vimeo_video.access', 'google.search_search.phrase', 'bing.search_
search.phrase')) order by itime desc)### t left join app_mdata t2 on lower(t.app)=lower(t2.name)
group by filename order by requests desc
```

Dataset Name	Description	Log Category
Self-Harm-Behaviour-Banned-User-Pie	Self-Harm Chat Search and Message Logging	app-ctrl

```
select
  filename,
  string_agg(
    distinct app,
    & #039; ') as app_agg, string_agg(distinct user_src, ' ') as user_agg, string_agg(distinct
`group`, ' ') as group_agg, string_agg(distinct ipstr(`srcip`), ' ') as srcip_agg, count(*) as
requests from ###(select $flex_timestamp as timestamp, filename, app, coalesce(nullifna(`user`),
nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, `group`, `srcip`, count(*) as total_num from
$log where $filter and ($banned_keywords) and (lower(app) in ('facebook_post', 'facebook_chat',
'twitter_post', 'youtube_video.access', 'gmail_chat', 'gmail_send.message', 'linkedin_post',
'vimeo_video.access', 'google.search_search.phrase', 'bing.search_search.phrase')) group by
timestamp, filename, app, user_src, `group`, `srcip` /*SkipSTART*/order by total_num desc,
timestamp desc/*SkipEND*/)### t group by filename order by requests desc
```

Dataset Name	Description	Log Category
Self-Harm-Behaviour-Banned-Application-Pie	Self-Harm Chat Search and Message Logging	app-ctrl

```
select
  filename,
  string_agg(
    distinct app,
    & #039; ') as app_agg, string_agg(distinct user_src, ' ') as user_agg, string_agg(distinct
`group`, ' ') as group_agg, string_agg(distinct ipstr(`srcip`), ' ') as srcip_agg, count(*) as
requests from ###(select $flex_timestamp as timestamp, filename, app, coalesce(nullifna(`user`),
nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, `group`, `srcip`, count(*) as total_num from
$log where $filter and ($banned_keywords) and (lower(app) in ('facebook_post', 'facebook_chat',
'twitter_post', 'youtube_video.access', 'gmail_chat', 'gmail_send.message', 'linkedin_post',
'vimeo_video.access', 'google.search_search.phrase', 'bing.search_search.phrase')) group by
timestamp, filename, app, user_src, `group`, `srcip` /*SkipSTART*/order by total_num desc,
timestamp desc/*SkipEND*/)### t group by filename order by requests desc
```

Dataset Name	Description	Log Category
Self-Harm-Behaviour-Banned-User-Bar	Self-Harm Chat Search and Message Logging	app-ctrl

```
select
  filename,
  string_agg(
    distinct app,
    & #039; ') as app_agg, string_agg(distinct user_src, ' ') as user_agg, string_agg(distinct
`group`, ' ') as group_agg, string_agg(distinct ipstr(`srcip`), ' ') as srcip_agg, count(*) as
requests from ###(select $flex_timestamp as timestamp, filename, app, coalesce(nullifna(`user`),
nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, `group`, `srcip`, count(*) as total_num from
$log where $filter and ($banned_keywords) and (lower(app) in ('facebook_post', 'facebook_chat',
'twitter_post', 'youtube_video.access', 'gmail_chat', 'gmail_send.message', 'linkedin_post',
'vimeo_video.access', 'google.search_search.phrase', 'bing.search_search.phrase')) group by
timestamp, filename, app, user_src, `group`, `srcip` /*SkipSTART*/order by total_num desc,
timestamp desc/*SkipEND*/)### t group by filename order by requests desc
```

Dataset Name	Description	Log Category
Self-Harm-Behaviour-Banned-User-Drilldown	Self-Harm Chat Search and Message Logging	app-ctrl

```
select
  filename,
  string_agg(
    distinct app,
    & #039; ') as app_agg, string_agg(distinct user_src, ' ') as user_agg, string_agg(distinct
`group`, ' ') as group_agg, string_agg(distinct ipstr(`srcip`), ' ') as srcip_agg, count(*) as
requests from ###(select $flex_timestamp as timestamp, filename, app, coalesce(nullifna(`user`),
nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, `group`, `srcip`, count(*) as total_num from
$log where $filter and ($banned_keywords) and (lower(app) in ('facebook_post', 'facebook_chat',
'twitter_post', 'youtube_video.access', 'gmail_chat', 'gmail_send.message', 'linkedin_post',
'vimeo_video.access', 'google.search_search.phrase', 'bing.search_search.phrase')) group by
timestamp, filename, app, user_src, `group`, `srcip` /*SkipSTART*/order by total_num desc,
timestamp desc/*SkipEND*/)### t group by filename order by requests desc
```

Dataset Name	Description	Log Category
Self-Harm-behaviour-banned	Self-Harm Chat Search and Message Logging	app-ctrl

```
select
  filename,
  string_agg(
    distinct app,
    & #039; ') as app_agg, string_agg(distinct user_src, ' ') as user_agg, string_agg(distinct
`group`, ' ') as group_agg, string_agg(distinct ipstr(`srcip`), ' ') as srcip_agg, count(*) as
requests from ###(select $flex_timestamp as timestamp, filename, app, coalesce(nullifna(`user`),
nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, `group`, `srcip`, count(*) as total_num from
$log where $filter and ($banned_keywords) and (lower(app) in ('facebook_post', 'facebook_chat',
'twitter_post', 'youtube_video.access', 'gmail_chat', 'gmail_send.message', 'linkedin_post',
'vimeo_video.access', 'google.search_search.phrase', 'bing.search_search.phrase')) group by
```

```
timestamp, filename, app, user_src, `group`, `srcip` /*SkipSTART*/order by total_num desc,
timestamp desc/*SkipEND*/)### t group by filename order by requests desc
```

Dataset Name	Description	Log Category
self-harm-Risky-Terms-By-App	Self-Harm Chat Search and Message Logging by Platforms	app-ctrl

```
select
  app,
  count(*) as requests
from
  ###(select $flex_timestamp as timestamp, filename, app, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, `group`, `srcip`, count(*) as total_num from $log
where $filter and ($banned_keywords) and (lower(app) in ('facebook_post', 'facebook_chat',
'twitter_post', 'youtube_video.access', 'gmail_chat', 'gmail_send.message', 'linkedin_post',
'vimeo_video.access', 'google.search_search.phrase', 'bing.search_search.phrase')) group by
timestamp, filename, app, user_src, `group`, `srcip` /*SkipSTART*/order by total_num desc,
timestamp desc/*SkipEND*/)### t group by app order by requests desc
```

Dataset Name	Description	Log Category
self-harm-Risky-Terms-Timeline	Self-Harm Chat Search and Message Logging Timeline	app-ctrl

```
select
  $flex_timescale(timestamp) as hodex,
  count(*) as requests
from
  ###(select $flex_timestamp as timestamp, filename, app, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, `group`, `srcip`, count(*) as total_num from $log
where $filter and ($banned_keywords) and (lower(app) in ('facebook_post', 'facebook_chat',
'twitter_post', 'youtube_video.access', 'gmail_chat', 'gmail_send.message', 'linkedin_post',
'vimeo_video.access', 'google.search_search.phrase', 'bing.search_search.phrase')) group by
timestamp, filename, app, user_src, `group`, `srcip` /*SkipSTART*/order by total_num desc,
timestamp desc/*SkipEND*/)### t group by hodex order by requests desc
```

Dataset Name	Description	Log Category
self-harm-Risky-Term-User-Drilldown	Self-Harm Chat Search and Message Logging by Users	app-ctrl

```
select
  user_src,
  filename,
  count(*) as requests
from
  ###(select $flex_timestamp as timestamp, filename, app, coalesce(nullifna(`user`), nullifna
```

```
(`unauthuser`), ipstr(`srcip`)) as user_src, `group`, `srcip`, count(*) as total_num from $log
where $filter and ($banned_keywords) and (lower(app) in ('facebook_post', 'facebook_chat',
'twitter_post', 'youtube_video.access', 'gmail_chat', 'gmail_send.message', 'linkedin_post',
'vimeo_video.access', 'google.search_search.phrase', 'bing.search_search.phrase')) group by
timestamp, filename, app, user_src, `group`, `srcip` /*SkipSTART*/order by total_num desc,
timestamp desc/*SkipEND*/)### t group by user_src, filename order by requests desc
```

Dataset Name	Description	Log Category
Browsing-Time-per-Social-Media	Browsing Time vs. Domain	traffic

```
select
  domain,
  ebtr_value(
    ebtr_agg_flat(browsetime),
    null,
    $timespan
  ) as browsetime
from
  ###(select domain, f_user, srcip, ebtr_agg_flat(browsetime) as browsetime, sum(bandwidth) as
bandwidth from (select app_group_name(app) as app_group, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as f_user, srcip, coalesce(nullifna(root_domain(hostname)), ipstr
(dstip), NULL) as domain, ebtr_agg_flat($browse_time) as browsetime, sum(coalesce(sentbyte,
0)+coalesce(rcvdbyte, 0)) as bandwidth from $log where $filter and (logflag&1>0) group by app_
group, f_user, hostname, domain, srcip, dstip) t1 inner join app_mdata t2 on lower(t1.app_
group)=lower(t2.name) where app_cat='Social.Media' group by domain, f_user, srcip order by
browsetime, bandwidth desc)### t where browsetime is not null group by domain order by browsetime
desc
```

Dataset Name	Description	Log Category
Social-Networking-Bar-Graph	Social Networking Browsing Time	traffic

```
select
  f_user,
  sum(bandwidth) as bandwidth
from
  ###(select domain, f_user, srcip, ebtr_agg_flat(browsetime) as browsetime, sum(bandwidth) as
bandwidth from (select app_group_name(app) as app_group, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as f_user, srcip, coalesce(nullifna(root_domain(hostname)), ipstr
(dstip), NULL) as domain, ebtr_agg_flat($browse_time) as browsetime, sum(coalesce(sentbyte,
0)+coalesce(rcvdbyte, 0)) as bandwidth from $log where $filter and (logflag&1>0) group by app_
group, f_user, hostname, domain, srcip, dstip) t1 inner join app_mdata t2 on lower(t1.app_
group)=lower(t2.name) where app_cat='Social.Media' group by domain, f_user, srcip order by
browsetime, bandwidth desc)### t where bandwidth>0 group by f_user order by bandwidth desc
```

Dataset Name	Description	Log Category
Top-Social-Networking-Durations-Sources-Drilldown	Top Social Networking Durations from Sources Drilldown	traffic

```
select
  f_user,
  ebtr_value(
    ebtr_agg_flat(browsetime),
    null,
    $timespan
  ) as browsetime
from
  ###(select domain, f_user, srcip, ebtr_agg_flat(browsetime) as browsetime, sum(bandwidth) as
bandwidth from (select app_group_name(app) as app_group, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as f_user, srcip, coalesce(nullifna(root_domain(hostname)), ipstr
(dstip), NULL) as domain, ebtr_agg_flat($browse_time) as browsetime, sum(coalesce(sentbyte,
0)+coalesce(rcvdbyte, 0)) as bandwidth from $log where $filter and (logflag&1>0) group by app_
group, f_user, hostname, domain, srcip, dstip) t1 inner join app_mdata t2 on lower(t1.app_
group)=lower(t2.name) where app_cat='Social.Media' group by domain, f_user, srcip order by
browsetime, bandwidth desc)### t where $filter-drilldown and browsetime is not null group by f_
user order by browsetime desc
```

Dataset Name	Description	Log Category
Top-Social-Networking-Durations-Domains-Drilldown	Browsing Time vs. Domain	traffic

```
select
  domain,
  ebtr_value(
    ebtr_agg_flat(browsetime),
    null,
    $timespan
  ) as browsetime
from
  ###(select domain, f_user, srcip, ebtr_agg_flat(browsetime) as browsetime, sum(bandwidth) as
bandwidth from (select app_group_name(app) as app_group, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as f_user, srcip, coalesce(nullifna(root_domain(hostname)), ipstr
(dstip), NULL) as domain, ebtr_agg_flat($browse_time) as browsetime, sum(coalesce(sentbyte,
0)+coalesce(rcvdbyte, 0)) as bandwidth from $log where $filter and (logflag&1>0) group by app_
group, f_user, hostname, domain, srcip, dstip) t1 inner join app_mdata t2 on lower(t1.app_
group)=lower(t2.name) where app_cat='Social.Media' group by domain, f_user, srcip order by
browsetime, bandwidth desc)### t where browsetime is not null group by domain order by browsetime
desc
```

Dataset Name	Description	Log Category
Facebook-Posts	Facebook Posts	app-ctrl

```
select
  i_time,
  f_user,
  srcip,
  filename
from
  ###(select from_itime(itime) as i_time, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr
(`srcip`)) as f_user, srcip, filename, app from $log where $filter and filename is not null order
by i_time desc)### t where lower(app)=lower('Facebook_Post') order by i_time desc
```

Dataset Name	Description	Log Category
Facebook-Chats	Facebook Chats	app-ctrl

```
select
  filename,
  string_agg(
    distinct from_itime(itime): :text,
    & #039; ') as itime_agg, string_agg(distinct user_src, ' ') as user_agg, string_agg(distinct
`group`, ' ') as group_agg, string_agg(distinct ipstr(srcip), ' ') as srcip_agg, count(*) as
requests from ###(select filename, itime, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr
(`srcip`)) as user_src, `group`, srcip, app from $log where $filter and filename is not null order
by itime desc)### t where lower(app)=lower('Facebook_Chat') group by filename order by requests
desc
```

Dataset Name	Description	Log Category
Twitter-Posts	Twitter Posts	app-ctrl

```
select
  i_time,
  f_user,
  srcip,
  filename
from
  ###(select from_itime(itime) as i_time, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr
(`srcip`)) as f_user, srcip, filename, app from $log where $filter and filename is not null order
by i_time desc)### t where lower(app)=lower('Twitter_Post') order by i_time desc
```

Dataset Name	Description	Log Category
LinkedIn-Posts-and-Comments	LinkedIn Posts and Comments	app-ctrl

```
select
  filename,
  string_agg(
    distinct from_itime(itime): :text,
    & #039; ') as itime_agg, string_agg(distinct user_src, ' ') as user_agg, string_agg(distinct
```

```
`group`, ' ') as group_agg, string_agg(distinct ipstr(srcip), ' ') as srcip_agg, count(*) as
requests from ###(select filename, itime, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr
(`srcip`)) as user_src, `group`, srcip, app from $log where $filter and filename is not null order
by itime desc)### t where lower(app)=lower('LinkedIn_Post') group by filename order by requests
desc
```

Dataset Name	Description	Log Category
sdwan-fw-Device-Interface-Quality_Bibandwidth-drilldown	SD-WAN Device-Interface Statistic	event

```
select
  devid,
  sum(bibandwidth)/ sum(count) as bibandwidth
from
  ###(select $flex_timestamp as timestamp, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck as sla_rule, sum(link_status) as link_status, sum(failed_latency) as
failed_latency, sum(failed_jitter) as failed_jitter,  sum(failed_packetloss) as failed_packetloss,
sum(latency) as latency, max(latency) as latency_max, min(latency) as latency_min, sum(jitter) as
jitter, max(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max
(packetloss) as packetloss_max, min(packetloss) as packetloss_min, sum(inbandwidth) as
inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as
count, sum(CASE WHEN link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_status) as sdwan_
status, sum(speedtest_cnt) as speedtest_cnt, sum(upbandwidth) as upbandwidth, sum(downbandwidth)
as downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE NULL END) AS latency, (CASE
WHEN link_status=1 THEN jitter ELSE NULL END) AS jitter, (CASE WHEN link_status=1 THEN packetloss
ELSE NULL END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END)
AS failed_packetloss, (CASE WHEN sla_failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed_
jitter, (CASE WHEN sla_failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed_latency, (CASE
WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status,  (CASE WHEN link_status=1 THEN
inbandwidth ELSE 0 END) AS inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth ELSE 0 END) AS
outbandwidth,  (CASE WHEN link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth, speedtest_
cnt, upbandwidth, downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, trim
(trailing 'ms' from latency)::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%'
THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS
sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num
(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE
WHEN logid_to_int(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num
(upbandwidthmeasured) as upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth
from $log where $filter and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is
not null) t ) t group by timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where $filter-drilldown and
bibandwidth is not null group by devid having sum(count)>0 order by bibandwidth desc
```

Dataset Name	Description	Log Category
sdwan-Device-Interface-Latency-Line	SD-WAN Device-Interface Latency Timeline	event

```
select
  $flex_timescale(timestamp) as hodex,
  t1.interface,
  min(latency) as latency
from
  (
    select
      timestamp,
      devid,
      interface,
      (
        case when sum(count_linkup)> 0 then sum(latency)/ sum(count_linkup) else NULL end
      ) as latency
    from
      ###(select $flex_timestamp as timestamp, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck as sla_rule, sum(link_status) as link_status, sum(failed_latency) as
failed_latency, sum(failed_jitter) as failed_jitter,  sum(failed_packetloss) as failed_packetloss,
sum(latency) as latency, max(latency) as latency_max, min(latency) as latency_min, sum(jitter) as
jitter, max(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max
(packetloss) as packetloss_max, min(packetloss) as packetloss_min, sum(inbandwidth) as
inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as
count, sum(CASE WHEN link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_status) as sdwan_
status, sum(speedtest_cnt) as speedtest_cnt, sum(upbandwidth) as upbandwidth, sum(downbandwidth)
as downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE NULL END) AS latency, (CASE
WHEN link_status=1 THEN jitter ELSE NULL END) AS jitter, (CASE WHEN link_status=1 THEN packetloss
ELSE NULL END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END)
AS failed_packetloss, (CASE WHEN sla_failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed_
jitter, (CASE WHEN sla_failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed_latency, (CASE
WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status,  (CASE WHEN link_status=1 THEN
inbandwidth ELSE 0 END) AS inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth ELSE 0 END) AS
outbandwidth,  (CASE WHEN link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth, speedtest_
cnt, upbandwidth, downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, trim
(trailing 'ms' from latency)::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%'
THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS
sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num
(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE
WHEN logid_to_int(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num
(upbandwidthmeasured) as upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth
from $log where $filter and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is
not null) t ) t group by timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t group by timestamp, devid,
interface having sum(count)>0) t1 inner join (select interface, count(*) as num_intf from ###
(select $flex_timestamp as timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
```

```
healthcheck as sla_rule, sum(link_status) as link_status, sum(failed_latency) as failed_latency,
sum(failed_jitter) as failed_jitter,  sum(failed_packetloss) as failed_packetloss, sum(latency) as
latency, max(latency) as latency_max, min(latency) as latency_min, sum(jitter) as jitter, max
(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max(packetloss)
as packetloss_max, min(packetloss) as packetloss_min, sum(inbandwidth) as inbandwidth, sum
(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN
link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_status) as sdwan_status, sum
(speedtest_cnt) as speedtest_cnt, sum(upbandwidth) as upbandwidth, sum(downbandwidth) as
downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE NULL END) AS latency, (CASE
WHEN link_status=1 THEN jitter ELSE NULL END) AS jitter, (CASE WHEN link_status=1 THEN packetloss
ELSE NULL END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END)
AS failed_packetloss, (CASE WHEN sla_failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed_
jitter, (CASE WHEN sla_failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed_latency, (CASE
WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status,  (CASE WHEN link_status=1 THEN
inbandwidth ELSE 0 END) AS inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth ELSE 0 END) AS
outbandwidth,  (CASE WHEN link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth, speedtest_
cnt, upbandwidth, downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, trim
(trailing 'ms' from latency)::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%'
THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS
sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num
(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE
WHEN logid_to_int(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num
(upbandwidthmeasured) as upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth
from $log where $filter and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is
not null) t ) t group by timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where $filter-drilldown and
interface is not null group by interface order by num_intf desc limit $ddown-top)t2 on
t1.interface=t2.interface group by hodex, t1.interface order by hodex
```

Dataset Name	Description	Log Category
sdwan-Device-Interface-Jitter-Line	SD-WAN Device-Interface Jitter Timeline	event

```
select
  $flex_timescale(timestamp) as hodex,
  t1.interface,
  min(jitter) as jitter
from
  (
    select
      timestamp,
      devid,
      interface,
      (
        case when sum(count_linkup)> 0 then sum(jitter)/ sum(count_linkup) else NULL end
      ) as jitter
    from
      ###(select $flex_timestamp as timestamp, csf, devname, dvid, devid, vd, interface,
```

```
speedtestserver, healthcheck as sla_rule, sum(link_status) as link_status, sum(failed_latency) as
failed_latency, sum(failed_jitter) as failed_jitter,  sum(failed_packetloss) as failed_packetloss,
sum(latency) as latency, max(latency) as latency_max, min(latency) as latency_min, sum(jitter) as
jitter, max(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max
(packetloss) as packetloss_max, min(packetloss) as packetloss_min, sum(inbandwidth) as
inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as
count, sum(CASE WHEN link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_status) as sdwan_
status, sum(speedtest_cnt) as speedtest_cnt, sum(upbandwidth) as upbandwidth, sum(downbandwidth)
as downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE NULL END) AS latency, (CASE
WHEN link_status=1 THEN jitter ELSE NULL END) AS jitter, (CASE WHEN link_status=1 THEN packetloss
ELSE NULL END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END)
AS failed_packetloss, (CASE WHEN sla_failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed_
jitter, (CASE WHEN sla_failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed_latency, (CASE
WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status,  (CASE WHEN link_status=1 THEN
inbandwidth ELSE 0 END) AS inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth ELSE 0 END) AS
outbandwidth,  (CASE WHEN link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth, speedtest_
cnt, upbandwidth, downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, trim
(trailing 'ms' from latency)::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%'
THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS
sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num
(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE
WHEN logid_to_int(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num
(upbandwidthmeasured) as upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth
from $log where $filter and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is
not null) t ) t group by timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t group by timestamp, devid,
interface having sum(count)>0) t1 inner join (select interface, count(*) as num_intf from ###
(select $flex_timestamp as timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck as sla_rule, sum(link_status) as link_status, sum(failed_latency) as failed_latency,
sum(failed_jitter) as failed_jitter,  sum(failed_packetloss) as failed_packetloss, sum(latency) as
latency, max(latency) as latency_max, min(latency) as latency_min, sum(jitter) as jitter, max
(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max(packetloss)
as packetloss_max, min(packetloss) as packetloss_min, sum(inbandwidth) as inbandwidth, sum
(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN
link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_status) as sdwan_status, sum
(speedtest_cnt) as speedtest_cnt, sum(upbandwidth) as upbandwidth, sum(downbandwidth) as
downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE NULL END) AS latency, (CASE
WHEN link_status=1 THEN jitter ELSE NULL END) AS jitter, (CASE WHEN link_status=1 THEN packetloss
ELSE NULL END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END)
AS failed_packetloss, (CASE WHEN sla_failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed_
jitter, (CASE WHEN sla_failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed_latency, (CASE
WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status,  (CASE WHEN link_status=1 THEN
inbandwidth ELSE 0 END) AS inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth ELSE 0 END) AS
outbandwidth,  (CASE WHEN link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth, speedtest_
cnt, upbandwidth, downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, trim
(trailing 'ms' from latency)::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%'
```

```
THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS
sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num
(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE
WHEN logid_to_int(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num
(upbandwidthmeasured) as upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth
from $log where $filter and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is
not null) t ) t group by timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where $filter-drilldown and
interface is not null group by interface order by num_intf desc limit $ddown-top)t2 on
t1.interface=t2.interface group by hodex, t1.interface order by hodex
```

Dataset Name	Description	Log Category
sdwan-Device-Interface-Packetloss-Line	SD-WAN Device-Interface Packetloss Timeline	event

```
select
  $flex_timescale(timestamp) as hodex,
  t1.interface,
  min(packetloss) as packetloss
from
  (
    select
      timestamp,
      devid,
      interface,
      (
        case when sum(count_linkup)> 0 then sum(packetloss)/ sum(count_linkup) else NULL end
      ) as packetloss
    from
      ###(select $flex_timestamp as timestamp, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck as sla_rule, sum(link_status) as link_status, sum(failed_latency) as
failed_latency, sum(failed_jitter) as failed_jitter,  sum(failed_packetloss) as failed_packetloss,
sum(latency) as latency, max(latency) as latency_max, min(latency) as latency_min, sum(jitter) as
jitter, max(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max
(packetloss) as packetloss_max, min(packetloss) as packetloss_min, sum(inbandwidth) as
inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as
count, sum(CASE WHEN link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_status) as sdwan_
status, sum(speedtest_cnt) as speedtest_cnt, sum(upbandwidth) as upbandwidth, sum(downbandwidth)
as downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE NULL END) AS latency, (CASE
WHEN link_status=1 THEN jitter ELSE NULL END) AS jitter, (CASE WHEN link_status=1 THEN packetloss
ELSE NULL END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END)
AS failed_packetloss, (CASE WHEN sla_failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed_
jitter, (CASE WHEN sla_failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed_latency, (CASE
WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status,  (CASE WHEN link_status=1 THEN
inbandwidth ELSE 0 END) AS inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth ELSE 0 END) AS
outbandwidth,  (CASE WHEN link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth, speedtest_
cnt, upbandwidth, downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, trim
(trailing 'ms' from latency)::float as latency, jitter::float as jitter, trim(trailing '%' from
```

```
packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%'
THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS
sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num
(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE
WHEN logid_to_int(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num
(upbandwidthmeasured) as upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth
from $log where $filter and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is
not null) t ) t group by timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t group by timestamp, devid,
interface having sum(count)>0) t1 inner join (select interface, count(*) as num_intf from ###
(select $flex_timestamp as timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck as sla_rule, sum(link_status) as link_status, sum(failed_latency) as failed_latency,
sum(failed_jitter) as failed_jitter,  sum(failed_packetloss) as failed_packetloss, sum(latency) as
latency, max(latency) as latency_max, min(latency) as latency_min, sum(jitter) as jitter, max
(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max(packetloss)
as packetloss_max, min(packetloss) as packetloss_min, sum(inbandwidth) as inbandwidth, sum
(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN
link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_status) as sdwan_status, sum
(speedtest_cnt) as speedtest_cnt, sum(upbandwidth) as upbandwidth, sum(downbandwidth) as
downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE NULL END) AS latency, (CASE
WHEN link_status=1 THEN jitter ELSE NULL END) AS jitter, (CASE WHEN link_status=1 THEN packetloss
ELSE NULL END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END)
AS failed_packetloss, (CASE WHEN sla_failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed_
jitter, (CASE WHEN sla_failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed_latency, (CASE
WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status,  (CASE WHEN link_status=1 THEN
inbandwidth ELSE 0 END) AS inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth ELSE 0 END) AS
outbandwidth,  (CASE WHEN link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth, speedtest_
cnt, upbandwidth, downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, trim
(trailing 'ms' from latency)::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%'
THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS
sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num
(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE
WHEN logid_to_int(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num
(upbandwidthmeasured) as upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth
from $log where $filter and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is
not null) t ) t group by timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where $filter-drilldown and
interface is not null group by interface order by num_intf desc limit $ddown-top)t2 on
t1.interface=t2.interface group by hodex, t1.interface order by hodex
```

Dataset Name	Description	Log Category
sdwan-Device-Latency-Line	SD-WAN Device Latency Timeline	event

```
select
  $flex_timescale(timestamp) as hodex,
  devid,
  min(latency) as latency
```

```
from
  (
    select
      timestamp,
      devid,
      interface,
      (
        case when sum(count_linkup)> 0 then sum(latency)/ sum(count_linkup) else NULL end
      ) as latency
    from
      ###(select $flex_timestamp as timestamp, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck as sla_rule, sum(link_status) as link_status, sum(failed_latency) as
failed_latency, sum(failed_jitter) as failed_jitter,  sum(failed_packetloss) as failed_packetloss,
sum(latency) as latency, max(latency) as latency_max, min(latency) as latency_min, sum(jitter) as
jitter, max(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max
(packetloss) as packetloss_max, min(packetloss) as packetloss_min, sum(inbandwidth) as
inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as
count, sum(CASE WHEN link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_status) as sdwan_
status, sum(speedtest_cnt) as speedtest_cnt, sum(upbandwidth) as upbandwidth, sum(downbandwidth)
as downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE NULL END) AS latency, (CASE
WHEN link_status=1 THEN jitter ELSE NULL END) AS jitter, (CASE WHEN link_status=1 THEN packetloss
ELSE NULL END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END)
AS failed_packetloss, (CASE WHEN sla_failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed_
jitter, (CASE WHEN sla_failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed_latency, (CASE
WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status,  (CASE WHEN link_status=1 THEN
inbandwidth ELSE 0 END) AS inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth ELSE 0 END) AS
outbandwidth,  (CASE WHEN link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth, speedtest_
cnt, upbandwidth, downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, trim
(trailing 'ms' from latency)::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%'
THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS
sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num
(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE
WHEN logid_to_int(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num
(upbandwidthmeasured) as upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth
from $log where $filter and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is
not null) t ) t group by timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where $filter-drilldown and
latency is not null group by timestamp, devid, interface having sum(count)>0) t1 group by hodex,
devid order by hodex
```

Dataset Name	Description	Log Category
sdwan-Device-Jitter-Line	SD-WAN Device Jitter Timeline	event

```
select
  $flex_timescale(timestamp) as hodex,
  devid,
  min(jitter) as jitter
```

```
from
  (
    select
      timestamp,
      devid,
      interface,
      (
        case when sum(count_linkup)> 0 then sum(jitter)/ sum(count_linkup) else NULL end
      ) as jitter
    from
      ###(select $flex_timestamp as timestamp, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck as sla_rule, sum(link_status) as link_status, sum(failed_latency) as
failed_latency, sum(failed_jitter) as failed_jitter,  sum(failed_packetloss) as failed_packetloss,
sum(latency) as latency, max(latency) as latency_max, min(latency) as latency_min, sum(jitter) as
jitter, max(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max
(packetloss) as packetloss_max, min(packetloss) as packetloss_min, sum(inbandwidth) as
inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as
count, sum(CASE WHEN link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_status) as sdwan_
status, sum(speedtest_cnt) as speedtest_cnt, sum(upbandwidth) as upbandwidth, sum(downbandwidth)
as downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE NULL END) AS latency, (CASE
WHEN link_status=1 THEN jitter ELSE NULL END) AS jitter, (CASE WHEN link_status=1 THEN packetloss
ELSE NULL END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END)
AS failed_packetloss, (CASE WHEN sla_failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed_
jitter, (CASE WHEN sla_failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed_latency, (CASE
WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status,  (CASE WHEN link_status=1 THEN
inbandwidth ELSE 0 END) AS inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth ELSE 0 END) AS
outbandwidth,  (CASE WHEN link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth, speedtest_
cnt, upbandwidth, downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, trim
(trailing 'ms' from latency)::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%'
THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS
sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num
(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE
WHEN logid_to_int(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num
(upbandwidthmeasured) as upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth
from $log where $filter and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is
not null) t ) t group by timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where $filter-drilldown and
jitter is not null group by timestamp, devid, interface having sum(count)>0) t1 group by hodex,
devid order by hodex
```

Dataset Name	Description	Log Category
sdwan-Device-Packetloss-Line	SD-WAN Device Packet Loss Timeline	event

```
select
  $flex_timescale(timestamp) as hodex,
  devid,
  min(packetloss) as packetloss
```

```
from
  (
    select
      timestamp,
      devid,
      interface,
      (
        case when sum(count_linkup)> 0 then sum(packetloss)/ sum(count_linkup) else NULL end
      ) as packetloss
    from
      ###(select $flex_timestamp as timestamp, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck as sla_rule, sum(link_status) as link_status, sum(failed_latency) as
failed_latency, sum(failed_jitter) as failed_jitter,  sum(failed_packetloss) as failed_packetloss,
sum(latency) as latency, max(latency) as latency_max, min(latency) as latency_min, sum(jitter) as
jitter, max(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max
(packetloss) as packetloss_max, min(packetloss) as packetloss_min, sum(inbandwidth) as
inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as
count, sum(CASE WHEN link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_status) as sdwan_
status, sum(speedtest_cnt) as speedtest_cnt, sum(upbandwidth) as upbandwidth, sum(downbandwidth)
as downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE NULL END) AS latency, (CASE
WHEN link_status=1 THEN jitter ELSE NULL END) AS jitter, (CASE WHEN link_status=1 THEN packetloss
ELSE NULL END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END)
AS failed_packetloss, (CASE WHEN sla_failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed_
jitter, (CASE WHEN sla_failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed_latency, (CASE
WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status,  (CASE WHEN link_status=1 THEN
inbandwidth ELSE 0 END) AS inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth ELSE 0 END) AS
outbandwidth,  (CASE WHEN link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth, speedtest_
cnt, upbandwidth, downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, trim
(trailing 'ms' from latency)::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%'
THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS
sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num
(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE
WHEN logid_to_int(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num
(upbandwidthmeasured) as upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth
from $log where $filter and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is
not null) t ) t group by timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where $filter-drilldown and
packetloss is not null group by timestamp, devid, interface having sum(count)>0) t1 group by
hodex, devid order by hodex
```

Dataset Name	Description	Log Category
sdwan-Device-Interface-Summary-by-Bibandwidth	SD-WAN Device Interface Summary by Bibandwidth	event

```
select
  tb1.devid,
  interface,
```

```
  sum(bibandwidth)/ sum(count) as bibandwidth,
  cast(
    min(latency_min) as decimal(18, 2)
  ) as latency_min,
  cast(
    (
      case when sum(count_linkup)> 0 then sum(latency)/ sum(count_linkup) else NULL end
    ) as decimal(18, 2)
  ) as latency_avg,
  cast(
    max(latency_max) as decimal(18, 2)
  ) as latency_max,
  cast(
    min(jitter_min) as decimal(18, 2)
  ) as jitter_min,
  cast(
    (
      case when sum(count_linkup)> 0 then sum(jitter)/ sum(count_linkup) else NULL end
    ) as decimal(18, 2)
  ) as jitter_avg,
  cast(
    max(jitter_max) as decimal(18, 2)
  ) as jitter_max,
  cast(
    min(packetloss_min) as decimal(18, 2)
  ) as packetloss_min,
  cast(
    (
      case when sum(count_linkup)> 0 then sum(packetloss)/ sum(count_linkup) else NULL end
    ) as decimal(18, 2)
  ) as packetloss_avg,
  cast(
    max(packetloss_max) as decimal(18, 2)
  ) as packetloss_max
from
  ###(select $flex_timestamp as timestamp, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck as sla_rule, sum(link_status) as link_status, sum(failed_latency) as
failed_latency, sum(failed_jitter) as failed_jitter,  sum(failed_packetloss) as failed_packetloss,
sum(latency) as latency, max(latency) as latency_max, min(latency) as latency_min, sum(jitter) as
jitter, max(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max
(packetloss) as packetloss_max, min(packetloss) as packetloss_min, sum(inbandwidth) as
inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as
count, sum(CASE WHEN link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_status) as sdwan_
status, sum(speedtest_cnt) as speedtest_cnt, sum(upbandwidth) as upbandwidth, sum(downbandwidth)
as downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE NULL END) AS latency, (CASE
WHEN link_status=1 THEN jitter ELSE NULL END) AS jitter, (CASE WHEN link_status=1 THEN packetloss
ELSE NULL END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END)
AS failed_packetloss, (CASE WHEN sla_failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed_
jitter, (CASE WHEN sla_failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed_latency, (CASE
WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status,  (CASE WHEN link_status=1 THEN
inbandwidth ELSE 0 END) AS inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth ELSE 0 END) AS
```

```
outbandwidth,  (CASE WHEN link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth, speedtest_
cnt, upbandwidth, downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, trim
(trailing 'ms' from latency)::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%'
THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS
sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num
(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE
WHEN logid_to_int(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num
(upbandwidthmeasured) as upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth
from $log where $filter and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is
not null) t ) t group by timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### tb1 inner join (select dvid,
intfname from $ADOM_SDWAN_INTF_INFO where $dev_filter) tb2 on tb1.dvid=tb2.dvid and
tb1.interface=tb2.intfname where $filter-drilldown and interface is not null group by tb1.devid,
interface having sum(count)>0 and sum(count_linkup)>0 order by devid, interface
```

Dataset Name	Description	Log Category
sdwan-Top-App-By-Bandwidth	Top SD-WAN application by bandwidth	traffic

```
select
  app_group,
  sum(bandwidth) as bandwidth,
  sum(sessions) as sessions
from
  ###(select $flex_timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf,
srcintfrole, dstintfrole, appid, appcat, app_group_name(app) as app_group, coalesce
(vwlservice,vwlname) as rulename, service, coalesce(nullifna(`srcname`),ipstr(`srcip`),nullifna
(`srcmac`)) as dev_src,  sum(crscore%65536) as crscore, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from $log-traffic where
$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf,
devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_
src, dev_src order by bandwidth desc)### t where $filter-drilldown group by app_group order by
bandwidth desc
```

Dataset Name	Description	Log Category
sdwan-Top-App-By-Bandwidth-Sankey	Top SD-WAN application by bandwidth usage	traffic

```
select
  & #039;SD-WAN Utilization' as summary, app_group, devid, dstintf as interface, sum(bandwidth) as
bandwidth  from ###(select $flex_timestamp as timestamp, csf, devid, vd, srccountry, dstintf,
srcintf, srcintfrole, dstintfrole, appid, appcat, app_group_name(app) as app_group, coalesce
(vwlservice,vwlname) as rulename, service, coalesce(nullifna(`srcname`),ipstr(`srcip`),nullifna
(`srcmac`)) as dev_src,  sum(crscore%65536) as crscore, coalesce(nullifna(`user`), nullifna
```

```
(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from $log-traffic where
$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf,
devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_
src, dev_src order by bandwidth desc)### t where $filter-drilldown group by app_group, devid,
interface order by bandwidth desc
```

Dataset Name	Description	Log Category
sdwan-Device-Interface-bandwidth-Drilldown	SD-WAN Device Statistic by Bibandwidth	event

```
select
  devid,
  sum(bibandwidth)/ sum(count) as bibandwidth
from
  ###(select $flex_timestamp as timestamp, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck as sla_rule, sum(link_status) as link_status, sum(failed_latency) as
failed_latency, sum(failed_jitter) as failed_jitter,  sum(failed_packetloss) as failed_packetloss,
sum(latency) as latency, max(latency) as latency_max, min(latency) as latency_min, sum(jitter) as
jitter, max(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max
(packetloss) as packetloss_max, min(packetloss) as packetloss_min, sum(inbandwidth) as
inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as
count, sum(CASE WHEN link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_status) as sdwan_
status, sum(speedtest_cnt) as speedtest_cnt, sum(upbandwidth) as upbandwidth, sum(downbandwidth)
as downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE NULL END) AS latency, (CASE
WHEN link_status=1 THEN jitter ELSE NULL END) AS jitter, (CASE WHEN link_status=1 THEN packetloss
ELSE NULL END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END)
AS failed_packetloss, (CASE WHEN sla_failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed_
jitter, (CASE WHEN sla_failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed_latency, (CASE
WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status,  (CASE WHEN link_status=1 THEN
inbandwidth ELSE 0 END) AS inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth ELSE 0 END) AS
outbandwidth,  (CASE WHEN link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth, speedtest_
cnt, upbandwidth, downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, trim
(trailing 'ms' from latency)::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%'
THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS
sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num
(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE
WHEN logid_to_int(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num
(upbandwidthmeasured) as upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth
from $log where $filter and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is
not null) t ) t group by timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where $filter-drilldown and
bibandwidth is not null group by devid having sum(count)>0 order by bibandwidth desc
```

Dataset Name	Description	Log Category
sdwan-Device-Rules-Donut-Bandwidth	Top SD-WAN Links bandwidth	traffic

```
select
  coalesce(
    rulename,
    & #039;Implicit') as rulename, sum(bandwidth) as bandwidth from ###(select $flex_timestamp as
timestamp, csf, devid, vd, srccountry, dstintf, srcintf, srcintfrole, dstintfrole, appid, appcat,
app_group_name(app) as app_group, coalesce(vwlservice,vwlname) as rulename, service, coalesce
(nullifna(`srcname`),ipstr(`srcip`),nullifna(`srcmac`)) as dev_src,  sum(crscore%65536) as
crscore, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, sum
(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce
(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out,
count(*) as sessions from $log-traffic where $filter and vwlid IS NOT NULL and (logflag&(1|32)>0)
group by timestamp, srccountry, dstintf, csf, devid, vd, srcintf, srcintfrole, dstintfrole, appid,
appcat, app_group, rulename, service, user_src, dev_src order by bandwidth desc)### t where
$filter-drilldown group by rulename order by bandwidth desc limit 10
```

Dataset Name	Description	Log Category
sdwan-device-interface-bandwidth	Top SD-WAN Links bandwidth	traffic

```
select
  tb1.intfname as interface,
  sum(rcvdbyte + sentbyte) as bandwidth
from
  $ADOM_INTF_STATS tb1
  inner join (
    select
      dvid,
      intfname
    from
      $ADOM_SDWAN_INTF_INFO
    where
      $filter - drilldown
      and $dev_filter
  ) tb2 on tb1.dvid = tb2.dvid
  and tb1.intfname = tb2.intfname
where
  $cust_time_filter(timestamp)
group by
  interface
order by
  bandwidth desc
```

Dataset Name	Description	Log Category
sdwan-Top-Application-Session-Bandwidth	Top SD-WAN application by bandwidth	traffic

```
select
  app_group,
  sum(bandwidth) as bandwidth,
  sum(sessions) as sessions
from
  ###(select $flex_timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf,
srcintfrole, dstintfrole, appid, appcat, app_group_name(app) as app_group, coalesce
(vwlservice,vwlname) as rulename, service, coalesce(nullifna(`srcname`),ipstr(`srcip`),nullifna
(`srcmac`)) as dev_src,  sum(crscore%65536) as crscore, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from $log-traffic where
$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf,
devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_
src, dev_src order by bandwidth desc)### t where $filter-drilldown group by app_group order by
bandwidth desc
```

Dataset Name	Description	Log Category
sdwan-Top-Users-By-Bandwidth-Bar	SD-WAN Top users by bandwidth usage	traffic

```
select
  user_src,
  sum(bandwidth) as bandwidth
from
  ###(select $flex_timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf,
srcintfrole, dstintfrole, appid, appcat, app_group_name(app) as app_group, coalesce
(vwlservice,vwlname) as rulename, service, coalesce(nullifna(`srcname`),ipstr(`srcip`),nullifna
(`srcmac`)) as dev_src,  sum(crscore%65536) as crscore, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from $log-traffic where
$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf,
devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_
src, dev_src order by bandwidth desc)### t where $filter-drilldown group by user_src order by
bandwidth desc
```

Dataset Name	Description	Log Category
sdwan-top-user-app-Drilldown	SD-WAN Top users and Application by bandwidth	traffic

```
select
  user_src,
  app_group,
  sum(bandwidth) as bandwidth
from
  ###(select $flex_timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf,
srcintfrole, dstintfrole, appid, appcat, app_group_name(app) as app_group, coalesce
(vwlservice,vwlname) as rulename, service, coalesce(nullifna(`srcname`),ipstr(`srcip`),nullifna
(`srcmac`)) as dev_src,  sum(crscore%65536) as crscore, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from $log-traffic where
$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf,
devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_
src, dev_src order by bandwidth desc)### t where $filter-drilldown group by user_src, app_group
order by bandwidth desc
```

Dataset Name	Description	Log Category
sdwan-Device-Intfe-traffic-out-bandwidth-Line	SD-WAN Device-Interface traffic sent bandwidth Timeline	traffic

```
select
  $flex_timescale(timestamp) as hodex,
  t1.dstintf as interface,
  sum(traffic_out) as bandwidth
from
  ###(select $flex_timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf,
srcintfrole, dstintfrole, appid, appcat, app_group_name(app) as app_group, coalesce
(vwlservice,vwlname) as rulename, service, coalesce(nullifna(`srcname`),ipstr(`srcip`),nullifna
(`srcmac`)) as dev_src,  sum(crscore%65536) as crscore, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from $log-traffic where
$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf,
devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_
src, dev_src order by bandwidth desc)### t1 inner join (select dstintf, count(*) as num_intf from
###(select $flex_timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf,
srcintfrole, dstintfrole, appid, appcat, app_group_name(app) as app_group, coalesce
(vwlservice,vwlname) as rulename, service, coalesce(nullifna(`srcname`),ipstr(`srcip`),nullifna
(`srcmac`)) as dev_src,  sum(crscore%65536) as crscore, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from $log-traffic where
$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf,
devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_
src, dev_src order by bandwidth desc)### t where $filter-drilldown group by dstintf order by num_
intf desc limit $ddown-top)t2 on t1.dstintf=t2.dstintf group by hodex, t1.dstintf order by hodex
```

Dataset Name	Description	Log Category
sdwan-Device-Intfe-traffic-in-bandwidth-Line	SD-WAN Device-Interface traffic received bandwidth Timeline	traffic

```
select
  $flex_timescale(timestamp) as hodex,
  t1.srcintf as interface,
  sum(traffic_in) as bandwidth
from
  ###(select $flex_timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf,
srcintfrole, dstintfrole, appid, appcat, app_group_name(app) as app_group, coalesce
(vwlservice,vwlname) as rulename, service, coalesce(nullifna(`srcname`),ipstr(`srcip`),nullifna
(`srcmac`)) as dev_src,  sum(crscore%65536) as crscore, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from $log-traffic where
$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf,
devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_
src, dev_src order by bandwidth desc)### t1 inner join (select srcintf, count(*) as num_intf from
###(select $flex_timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf,
srcintfrole, dstintfrole, appid, appcat, app_group_name(app) as app_group, coalesce
(vwlservice,vwlname) as rulename, service, coalesce(nullifna(`srcname`),ipstr(`srcip`),nullifna
(`srcmac`)) as dev_src,  sum(crscore%65536) as crscore, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from $log-traffic where
$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf,
devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_
src, dev_src order by bandwidth desc)### t where $filter-drilldown and srcintf is not null and
srcintfrole ='wan' group by srcintf order by num_intf desc limit $ddown-top)t2 on
t1.srcintf=t2.srcintf group by hodex, t1.srcintf order by hodex
```

Dataset Name	Description	Log Category
sdwan-Device-Intfe-traffic-bandwidth-Line	SD-WAN Device-Interface traffic bandwidth Timeline	traffic

```
select
  $flex_timescale(timestamp) as hodex,
  tb1.intfname as interface,
  sum(sentbyte) as traffic_out,
  sum(rcvdbyte) as traffic_in,
  sum(rcvdbyte + sentbyte) as bandwidth
from
  $ADOM_INTF_STATS tb1
  inner join (
    select
      dvid,
      intfname
```

```
    from
      $ADOM_SDWAN_INTF_INFO
    where
      $filter - drilldown
      and $dev_filter
  ) tb2 on tb1.dvid = tb2.dvid
  and tb1.intfname = tb2.intfname
where
  $cust_time_filter(timestamp)
group by
  hodex,
  tb1.intfname
order by
  hodex
```

Dataset Name	Description	Log Category
sdwan-Device-SLA-Interface-bandwidth-Drilldown	SD-WAN Device Statistic by Bibandwidth	event

```
select
  devid,
  sum(bibandwidth)/ sum(count) as bibandwidth
from
  ###(select $flex_timestamp as timestamp, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck as sla_rule, sum(link_status) as link_status, sum(failed_latency) as
failed_latency, sum(failed_jitter) as failed_jitter,  sum(failed_packetloss) as failed_packetloss,
sum(latency) as latency, max(latency) as latency_max, min(latency) as latency_min, sum(jitter) as
jitter, max(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max
(packetloss) as packetloss_max, min(packetloss) as packetloss_min, sum(inbandwidth) as
inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as
count, sum(CASE WHEN link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_status) as sdwan_
status, sum(speedtest_cnt) as speedtest_cnt, sum(upbandwidth) as upbandwidth, sum(downbandwidth)
as downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE NULL END) AS latency, (CASE
WHEN link_status=1 THEN jitter ELSE NULL END) AS jitter, (CASE WHEN link_status=1 THEN packetloss
ELSE NULL END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END)
AS failed_packetloss, (CASE WHEN sla_failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed_
jitter, (CASE WHEN sla_failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed_latency, (CASE
WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status,  (CASE WHEN link_status=1 THEN
inbandwidth ELSE 0 END) AS inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth ELSE 0 END) AS
outbandwidth,  (CASE WHEN link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth, speedtest_
cnt, upbandwidth, downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, trim
(trailing 'ms' from latency)::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%'
THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS
sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num
(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE
WHEN logid_to_int(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num
(upbandwidthmeasured) as upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth
```

```
from $log where $filter and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is
not null) t ) t group by timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where $filter-drilldown and
bibandwidth is not null group by devid having sum(count)>0 order by bibandwidth desc
```

Dataset Name	Description	Log Category
sdwan-Device-SLA-Rule-Latency-Line	SD-WAN Device-SLA-Rule Latency Line	event

```
select
  $flex_timescale(timestamp) as hodex,
  t1.intf_sla,
  (
    case when sum(count_linkup)> 0 then sum(latency)/ sum(count_linkup) else NULL end
  ) as latency
from
  (
    select
      timestamp,
      interface || & #039;:' || sla_rule as intf_sla, sum(latency) as latency, sum(count_linkup)
as count_linkup from ###(select $flex_timestamp as timestamp, csf, devname, dvid, devid, vd,
interface, speedtestserver, healthcheck as sla_rule, sum(link_status) as link_status, sum(failed_
latency) as failed_latency, sum(failed_jitter) as failed_jitter,  sum(failed_packetloss) as
failed_packetloss, sum(latency) as latency, max(latency) as latency_max, min(latency) as latency_
min, sum(jitter) as jitter, max(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss)
as packetloss, max(packetloss) as packetloss_max, min(packetloss) as packetloss_min, sum
(inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth,
count(*) as count, sum(CASE WHEN link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_
status) as sdwan_status, sum(speedtest_cnt) as speedtest_cnt, sum(upbandwidth) as upbandwidth, sum
(downbandwidth) as downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE NULL END) AS
latency, (CASE WHEN link_status=1 THEN jitter ELSE NULL END) AS jitter, (CASE WHEN link_status=1
THEN packetloss ELSE NULL END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN
1 ELSE 0 END) AS failed_packetloss, (CASE WHEN sla_failed=1 AND metric='jitter' THEN 1 ELSE 0 END)
AS failed_jitter, (CASE WHEN sla_failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed_
latency, (CASE WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status,  (CASE WHEN link_
status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth
ELSE 0 END) AS outbandwidth,  (CASE WHEN link_status=1 THEN bibandwidth ELSE 0 END) AS
bibandwidth, speedtest_cnt, upbandwidth, downbandwidth from (select itime, csf, devname, dvid,
devid, vd, interface, speedtestserver, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS
link_status, trim(trailing 'ms' from latency)::float as latency, jitter::float as jitter, trim
(trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE
'%SLA failed%' THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1
ELSE 0 END) AS sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_
num(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE
WHEN logid_to_int(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num
(upbandwidthmeasured) as upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth
from $log where $filter and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is
not null) t ) t group by timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where latency is not null group
```

```
by timestamp, intf_sla having sum(count)>0) t1 inner join (select interface || ':' || sla_rule as
intf_sla, count(*) as num_intf from ###(select $flex_timestamp as timestamp, csf, devname, dvid,
devid, vd, interface, speedtestserver, healthcheck as sla_rule, sum(link_status) as link_status,
sum(failed_latency) as failed_latency, sum(failed_jitter) as failed_jitter,  sum(failed_
packetloss) as failed_packetloss, sum(latency) as latency, max(latency) as latency_max, min
(latency) as latency_min, sum(jitter) as jitter, max(jitter) as jitter_max, min(jitter) as jitter_
min, sum(packetloss) as packetloss, max(packetloss) as packetloss_max, min(packetloss) as
packetloss_min, sum(inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum
(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN link_status=1 THEN 1 ELSE 0 END) AS
count_linkup, min(sdwan_status) as sdwan_status, sum(speedtest_cnt) as speedtest_cnt, sum
(upbandwidth) as upbandwidth, sum(downbandwidth) as downbandwidth from (select itime, csf,
devname, dvid, devid, vd, interface, speedtestserver, healthcheck, link_status, (CASE WHEN link_
status=1 THEN latency ELSE NULL END) AS latency, (CASE WHEN link_status=1 THEN jitter ELSE NULL
END) AS jitter, (CASE WHEN link_status=1 THEN packetloss ELSE NULL END) AS packetloss, (CASE WHEN
sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed_packetloss, (CASE WHEN sla_
failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed_jitter, (CASE WHEN sla_failed=1 AND
metric='latency' THEN 1 ELSE 0 END) AS failed_latency, (CASE WHEN sla_failed=1 THEN 3 ELSE sdwan_
status END) AS sdwan_status,  (CASE WHEN link_status=1 THEN inbandwidth ELSE 0 END) AS
inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth,  (CASE WHEN
link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth, speedtest_cnt, upbandwidth,
downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, trim(trailing 'ms' from
latency)::float as latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as
packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS
sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan_status,
convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num(outbandwidthused) as
outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE WHEN logid_to_int
(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num(upbandwidthmeasured) as
upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth from $log where $filter
and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is not null) t ) t group by
timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver, healthcheck
/*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where $filter-drilldown and sla_rule is not
null group by intf_sla order by num_intf desc limit $ddown-top)t2 on t1.intf_sla=t2.intf_sla group
by hodex, t1.intf_sla order by hodex
```

Dataset Name	Description	Log Category
sdwan-Device-SLA-Rule-Jitter-Line	SD-WAN Device-SLA-Rule Jitter Line	event

```
select
  $flex_timescale(timestamp) as hodex,
  t1.intf_sla,
  (
    case when sum(count_linkup)> 0 then sum(jitter)/ sum(count_linkup) else NULL end
  ) as jitter
from
  (
    select
      timestamp,
      interface || & #039;:' || sla_rule as intf_sla, sum(jitter) as jitter, sum(count_linkup) as
count_linkup from ###(select $flex_timestamp as timestamp, csf, devname, dvid, devid, vd,
```

```
interface, speedtestserver, healthcheck as sla_rule, sum(link_status) as link_status, sum(failed_
latency) as failed_latency, sum(failed_jitter) as failed_jitter,  sum(failed_packetloss) as
failed_packetloss, sum(latency) as latency, max(latency) as latency_max, min(latency) as latency_
min, sum(jitter) as jitter, max(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss)
as packetloss, max(packetloss) as packetloss_max, min(packetloss) as packetloss_min, sum
(inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth,
count(*) as count, sum(CASE WHEN link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_
status) as sdwan_status, sum(speedtest_cnt) as speedtest_cnt, sum(upbandwidth) as upbandwidth, sum
(downbandwidth) as downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE NULL END) AS
latency, (CASE WHEN link_status=1 THEN jitter ELSE NULL END) AS jitter, (CASE WHEN link_status=1
THEN packetloss ELSE NULL END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN
1 ELSE 0 END) AS failed_packetloss, (CASE WHEN sla_failed=1 AND metric='jitter' THEN 1 ELSE 0 END)
AS failed_jitter, (CASE WHEN sla_failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed_
latency, (CASE WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status,  (CASE WHEN link_
status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth
ELSE 0 END) AS outbandwidth,  (CASE WHEN link_status=1 THEN bibandwidth ELSE 0 END) AS
bibandwidth, speedtest_cnt, upbandwidth, downbandwidth from (select itime, csf, devname, dvid,
devid, vd, interface, speedtestserver, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS
link_status, trim(trailing 'ms' from latency)::float as latency, jitter::float as jitter, trim
(trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE
'%SLA failed%' THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1
ELSE 0 END) AS sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_
num(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE
WHEN logid_to_int(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num
(upbandwidthmeasured) as upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth
from $log where $filter and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is
not null) t ) t group by timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where jitter is not null group
by timestamp, intf_sla having sum(count)>0) t1 inner join (select interface || ':' || sla_rule as
intf_sla, count(*) as num_intf from ###(select $flex_timestamp as timestamp, csf, devname, dvid,
devid, vd, interface, speedtestserver, healthcheck as sla_rule, sum(link_status) as link_status,
sum(failed_latency) as failed_latency, sum(failed_jitter) as failed_jitter,  sum(failed_
packetloss) as failed_packetloss, sum(latency) as latency, max(latency) as latency_max, min
(latency) as latency_min, sum(jitter) as jitter, max(jitter) as jitter_max, min(jitter) as jitter_
min, sum(packetloss) as packetloss, max(packetloss) as packetloss_max, min(packetloss) as
packetloss_min, sum(inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum
(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN link_status=1 THEN 1 ELSE 0 END) AS
count_linkup, min(sdwan_status) as sdwan_status, sum(speedtest_cnt) as speedtest_cnt, sum
(upbandwidth) as upbandwidth, sum(downbandwidth) as downbandwidth from (select itime, csf,
devname, dvid, devid, vd, interface, speedtestserver, healthcheck, link_status, (CASE WHEN link_
status=1 THEN latency ELSE NULL END) AS latency, (CASE WHEN link_status=1 THEN jitter ELSE NULL
END) AS jitter, (CASE WHEN link_status=1 THEN packetloss ELSE NULL END) AS packetloss, (CASE WHEN
sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed_packetloss, (CASE WHEN sla_
failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed_jitter, (CASE WHEN sla_failed=1 AND
metric='latency' THEN 1 ELSE 0 END) AS failed_latency, (CASE WHEN sla_failed=1 THEN 3 ELSE sdwan_
status END) AS sdwan_status,  (CASE WHEN link_status=1 THEN inbandwidth ELSE 0 END) AS
inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth,  (CASE WHEN
link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth, speedtest_cnt, upbandwidth,
downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, trim(trailing 'ms' from
latency)::float as latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as
```

```
packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS
sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan_status,
convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num(outbandwidthused) as
outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE WHEN logid_to_int
(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num(upbandwidthmeasured) as
upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth from $log where $filter
and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is not null) t ) t group by
timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver, healthcheck
/*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where $filter-drilldown and sla_rule is not
null group by intf_sla order by num_intf desc limit $ddown-top)t2 on t1.intf_sla=t2.intf_sla group
by hodex, t1.intf_sla order by hodex
```

Dataset Name	Description	Log Category
sdwan-Device-SLA-Rule-Packetloss-Line	SD-WAN Device-SLA-Rule Packetloss Line	event

```
select
  $flex_timescale(timestamp) as hodex,
  t1.intf_sla,
  (
    case when sum(count_linkup)> 0 then sum(packetloss)/ sum(count_linkup) else NULL end
  ) as packetloss
from
  (
    select
      timestamp,
      interface || & #039;:' || sla_rule as intf_sla, sum(packetloss) as packetloss, sum(count_
linkup) as count_linkup from ###(select $flex_timestamp as timestamp, csf, devname, dvid, devid,
vd, interface, speedtestserver, healthcheck as sla_rule, sum(link_status) as link_status, sum
(failed_latency) as failed_latency, sum(failed_jitter) as failed_jitter,  sum(failed_packetloss)
as failed_packetloss, sum(latency) as latency, max(latency) as latency_max, min(latency) as
latency_min, sum(jitter) as jitter, max(jitter) as jitter_max, min(jitter) as jitter_min, sum
(packetloss) as packetloss, max(packetloss) as packetloss_max, min(packetloss) as packetloss_min,
sum(inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as
bibandwidth, count(*) as count, sum(CASE WHEN link_status=1 THEN 1 ELSE 0 END) AS count_linkup,
min(sdwan_status) as sdwan_status, sum(speedtest_cnt) as speedtest_cnt, sum(upbandwidth) as
upbandwidth, sum(downbandwidth) as downbandwidth from (select itime, csf, devname, dvid, devid,
vd, interface, speedtestserver, healthcheck, link_status, (CASE WHEN link_status=1 THEN latency
ELSE NULL END) AS latency, (CASE WHEN link_status=1 THEN jitter ELSE NULL END) AS jitter, (CASE
WHEN link_status=1 THEN packetloss ELSE NULL END) AS packetloss, (CASE WHEN sla_failed=1 AND
metric='packetloss' THEN 1 ELSE 0 END) AS failed_packetloss, (CASE WHEN sla_failed=1 AND
metric='jitter' THEN 1 ELSE 0 END) AS failed_jitter, (CASE WHEN sla_failed=1 AND metric='latency'
THEN 1 ELSE 0 END) AS failed_latency, (CASE WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS
sdwan_status,  (CASE WHEN link_status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth,  (CASE WHEN
link_status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth,  (CASE WHEN link_status=1 THEN
bibandwidth ELSE 0 END) AS bibandwidth, speedtest_cnt, upbandwidth, downbandwidth from (select
itime, csf, devname, dvid, devid, vd, interface, speedtestserver, healthcheck, (CASE WHEN
status='down' THEN 0 ELSE 1 END) AS link_status, trim(trailing 'ms' from latency)::float as
latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE
WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla_failed, metric,
```

```
(CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan_status, convert_unit_to_num
(inbandwidthused) as inbandwidth, convert_unit_to_num(outbandwidthused) as outbandwidth, convert_
unit_to_num(bibandwidthused) as bibandwidth, (CASE WHEN logid_to_int(logid)=22938 THEN 1 ELSE 0
END) AS speedtest_cnt, convert_unit_to_num(upbandwidthmeasured) as upbandwidth, convert_unit_to_
num(downbandwidthmeasured) as downbandwidth from $log where $filter and logid_to_int(logid) in
(22925, 22933, 22936, 22938) and interface is not null) t ) t group by timestamp, csf, devname,
dvid, devid, vd, interface, speedtestserver, healthcheck /*SkipSTART*/order by timestamp
desc/*SkipEND*/)### t where packetloss is not null group by timestamp, intf_sla having sum
(count)>0) t1 inner join (select interface || ':' || sla_rule as intf_sla, count(*) as num_intf
from ###(select $flex_timestamp as timestamp, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck as sla_rule, sum(link_status) as link_status, sum(failed_latency) as
failed_latency, sum(failed_jitter) as failed_jitter,  sum(failed_packetloss) as failed_packetloss,
sum(latency) as latency, max(latency) as latency_max, min(latency) as latency_min, sum(jitter) as
jitter, max(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max
(packetloss) as packetloss_max, min(packetloss) as packetloss_min, sum(inbandwidth) as
inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as
count, sum(CASE WHEN link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_status) as sdwan_
status, sum(speedtest_cnt) as speedtest_cnt, sum(upbandwidth) as upbandwidth, sum(downbandwidth)
as downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE NULL END) AS latency, (CASE
WHEN link_status=1 THEN jitter ELSE NULL END) AS jitter, (CASE WHEN link_status=1 THEN packetloss
ELSE NULL END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END)
AS failed_packetloss, (CASE WHEN sla_failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed_
jitter, (CASE WHEN sla_failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed_latency, (CASE
WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status,  (CASE WHEN link_status=1 THEN
inbandwidth ELSE 0 END) AS inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth ELSE 0 END) AS
outbandwidth,  (CASE WHEN link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth, speedtest_
cnt, upbandwidth, downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, trim
(trailing 'ms' from latency)::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%'
THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS
sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num
(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE
WHEN logid_to_int(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num
(upbandwidthmeasured) as upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth
from $log where $filter and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is
not null) t ) t group by timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where $filter-drilldown and sla_
rule is not null group by intf_sla order by num_intf desc limit $ddown-top)t2 on t1.intf_
sla=t2.intf_sla group by hodex, t1.intf_sla order by hodex
```

Dataset Name	Description	Log Category
sdwan-device-sla-intf-latency-pass-percent	SD-WAN Device Latency Pass Percentage by SLA rules and Interface	event

```
select
  sla_rule,
  interface,
  cast(
```

```
    100 *(
      1 - sum(failed_latency)/ sum(count_linkup)
    ) as decimal(18, 2)
  ) as latency
from
  ###(select $flex_timestamp as timestamp, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck as sla_rule, sum(link_status) as link_status, sum(failed_latency) as
failed_latency, sum(failed_jitter) as failed_jitter,  sum(failed_packetloss) as failed_packetloss,
sum(latency) as latency, max(latency) as latency_max, min(latency) as latency_min, sum(jitter) as
jitter, max(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max
(packetloss) as packetloss_max, min(packetloss) as packetloss_min, sum(inbandwidth) as
inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as
count, sum(CASE WHEN link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_status) as sdwan_
status, sum(speedtest_cnt) as speedtest_cnt, sum(upbandwidth) as upbandwidth, sum(downbandwidth)
as downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE NULL END) AS latency, (CASE
WHEN link_status=1 THEN jitter ELSE NULL END) AS jitter, (CASE WHEN link_status=1 THEN packetloss
ELSE NULL END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END)
AS failed_packetloss, (CASE WHEN sla_failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed_
jitter, (CASE WHEN sla_failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed_latency, (CASE
WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status,  (CASE WHEN link_status=1 THEN
inbandwidth ELSE 0 END) AS inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth ELSE 0 END) AS
outbandwidth,  (CASE WHEN link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth, speedtest_
cnt, upbandwidth, downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, trim
(trailing 'ms' from latency)::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%'
THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS
sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num
(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE
WHEN logid_to_int(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num
(upbandwidthmeasured) as upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth
from $log where $filter and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is
not null) t ) t group by timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where $filter-drilldown and sla_
rule is not null group by sla_rule, interface having sum(count_linkup)>0 order by latency desc
```

Dataset Name	Description	Log Category
sdwan-device-sla-intf-jitter-pass-percent	SD-WAN Device Jitter Pass Percentage by SLA rules and Interface	event

```
select
  sla_rule,
  interface,
  cast(
    100 *(
      1 - sum(failed_jitter)/ sum(count_linkup)
    ) as decimal(18, 2)
  ) as jitter
from
```

```
  ###(select $flex_timestamp as timestamp, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck as sla_rule, sum(link_status) as link_status, sum(failed_latency) as
failed_latency, sum(failed_jitter) as failed_jitter,  sum(failed_packetloss) as failed_packetloss,
sum(latency) as latency, max(latency) as latency_max, min(latency) as latency_min, sum(jitter) as
jitter, max(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max
(packetloss) as packetloss_max, min(packetloss) as packetloss_min, sum(inbandwidth) as
inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as
count, sum(CASE WHEN link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_status) as sdwan_
status, sum(speedtest_cnt) as speedtest_cnt, sum(upbandwidth) as upbandwidth, sum(downbandwidth)
as downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE NULL END) AS latency, (CASE
WHEN link_status=1 THEN jitter ELSE NULL END) AS jitter, (CASE WHEN link_status=1 THEN packetloss
ELSE NULL END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END)
AS failed_packetloss, (CASE WHEN sla_failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed_
jitter, (CASE WHEN sla_failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed_latency, (CASE
WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status,  (CASE WHEN link_status=1 THEN
inbandwidth ELSE 0 END) AS inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth ELSE 0 END) AS
outbandwidth,  (CASE WHEN link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth, speedtest_
cnt, upbandwidth, downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, trim
(trailing 'ms' from latency)::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%'
THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS
sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num
(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE
WHEN logid_to_int(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num
(upbandwidthmeasured) as upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth
from $log where $filter and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is
not null) t ) t group by timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where $filter-drilldown and sla_
rule is not null group by sla_rule, interface having sum(count_linkup)>0 order by jitter desc
```

Dataset Name	Description	Log Category
sdwan-device-sla-intf-packetloss-pass-percent	SD-WAN Device Packet Loss Pass Percentage by SLA rules and Interface	event

```
select
  sla_rule,
  interface,
  cast(
    100 *(
      1 - sum(failed_packetloss)/ sum(count_linkup)
    ) as decimal(18, 2)
  ) as packetloss
from
  ###(select $flex_timestamp as timestamp, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck as sla_rule, sum(link_status) as link_status, sum(failed_latency) as
failed_latency, sum(failed_jitter) as failed_jitter,  sum(failed_packetloss) as failed_packetloss,
sum(latency) as latency, max(latency) as latency_max, min(latency) as latency_min, sum(jitter) as
jitter, max(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max
```

```
(packetloss) as packetloss_max, min(packetloss) as packetloss_min, sum(inbandwidth) as
inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as
count, sum(CASE WHEN link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_status) as sdwan_
status, sum(speedtest_cnt) as speedtest_cnt, sum(upbandwidth) as upbandwidth, sum(downbandwidth)
as downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE NULL END) AS latency, (CASE
WHEN link_status=1 THEN jitter ELSE NULL END) AS jitter, (CASE WHEN link_status=1 THEN packetloss
ELSE NULL END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END)
AS failed_packetloss, (CASE WHEN sla_failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed_
jitter, (CASE WHEN sla_failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed_latency, (CASE
WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status,  (CASE WHEN link_status=1 THEN
inbandwidth ELSE 0 END) AS inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth ELSE 0 END) AS
outbandwidth,  (CASE WHEN link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth, speedtest_
cnt, upbandwidth, downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, trim
(trailing 'ms' from latency)::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%'
THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS
sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num
(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE
WHEN logid_to_int(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num
(upbandwidthmeasured) as upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth
from $log where $filter and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is
not null) t ) t group by timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where $filter-drilldown and sla_
rule is not null group by sla_rule, interface having sum(count_linkup)>0 order by packetloss desc
```

Dataset Name	Description	Log Category
sdwan-Device-Intf-List-by-Availability	SD-WAN Device Interface List by Availability	event

```
select
  devname || & #039;:' || interface as dev_intf, sum(count_linkup)/sum(count) as available from
###(select $flex_timestamp as timestamp, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck as sla_rule, sum(link_status) as link_status, sum(failed_latency) as
failed_latency, sum(failed_jitter) as failed_jitter,  sum(failed_packetloss) as failed_packetloss,
sum(latency) as latency, max(latency) as latency_max, min(latency) as latency_min, sum(jitter) as
jitter, max(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max
(packetloss) as packetloss_max, min(packetloss) as packetloss_min, sum(inbandwidth) as
inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as
count, sum(CASE WHEN link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_status) as sdwan_
status, sum(speedtest_cnt) as speedtest_cnt, sum(upbandwidth) as upbandwidth, sum(downbandwidth)
as downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE NULL END) AS latency, (CASE
WHEN link_status=1 THEN jitter ELSE NULL END) AS jitter, (CASE WHEN link_status=1 THEN packetloss
ELSE NULL END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END)
AS failed_packetloss, (CASE WHEN sla_failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed_
jitter, (CASE WHEN sla_failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed_latency, (CASE
WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status,  (CASE WHEN link_status=1 THEN
inbandwidth ELSE 0 END) AS inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth ELSE 0 END) AS
```

```
outbandwidth,  (CASE WHEN link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth, speedtest_
cnt, upbandwidth, downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, trim
(trailing 'ms' from latency)::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%'
THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS
sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num
(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE
WHEN logid_to_int(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num
(upbandwidthmeasured) as upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth
from $log where $filter and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is
not null) t ) t group by timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where $filter-drilldown group by
dev_intf having sum(count)>0 order by dev_intf
```

Dataset Name	Description	Log Category
sdwan-Device-Intf-Updown-Timeline	SD-WAN Device Interface Updown Time Line	event

```
select
  $fv_line_timescale(timestamp) as hodex,
  devname || & #039;:' || interface as dev_intf, cast(100*sum(count_linkup)/sum(count) as decimal
(10,2)) as sdwan_status from ###(select $flex_timestamp as timestamp, csf, devname, dvid, devid,
vd, interface, speedtestserver, healthcheck as sla_rule, sum(link_status) as link_status, sum
(failed_latency) as failed_latency, sum(failed_jitter) as failed_jitter,  sum(failed_packetloss)
as failed_packetloss, sum(latency) as latency, max(latency) as latency_max, min(latency) as
latency_min, sum(jitter) as jitter, max(jitter) as jitter_max, min(jitter) as jitter_min, sum
(packetloss) as packetloss, max(packetloss) as packetloss_max, min(packetloss) as packetloss_min,
sum(inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as
bibandwidth, count(*) as count, sum(CASE WHEN link_status=1 THEN 1 ELSE 0 END) AS count_linkup,
min(sdwan_status) as sdwan_status, sum(speedtest_cnt) as speedtest_cnt, sum(upbandwidth) as
upbandwidth, sum(downbandwidth) as downbandwidth from (select itime, csf, devname, dvid, devid,
vd, interface, speedtestserver, healthcheck, link_status, (CASE WHEN link_status=1 THEN latency
ELSE NULL END) AS latency, (CASE WHEN link_status=1 THEN jitter ELSE NULL END) AS jitter, (CASE
WHEN link_status=1 THEN packetloss ELSE NULL END) AS packetloss, (CASE WHEN sla_failed=1 AND
metric='packetloss' THEN 1 ELSE 0 END) AS failed_packetloss, (CASE WHEN sla_failed=1 AND
metric='jitter' THEN 1 ELSE 0 END) AS failed_jitter, (CASE WHEN sla_failed=1 AND metric='latency'
THEN 1 ELSE 0 END) AS failed_latency, (CASE WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS
sdwan_status,  (CASE WHEN link_status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth,  (CASE WHEN
link_status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth,  (CASE WHEN link_status=1 THEN
bibandwidth ELSE 0 END) AS bibandwidth, speedtest_cnt, upbandwidth, downbandwidth from (select
itime, csf, devname, dvid, devid, vd, interface, speedtestserver, healthcheck, (CASE WHEN
status='down' THEN 0 ELSE 1 END) AS link_status, trim(trailing 'ms' from latency)::float as
latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE
WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla_failed, metric,
(CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan_status, convert_unit_to_num
(inbandwidthused) as inbandwidth, convert_unit_to_num(outbandwidthused) as outbandwidth, convert_
unit_to_num(bibandwidthused) as bibandwidth, (CASE WHEN logid_to_int(logid)=22938 THEN 1 ELSE 0
END) AS speedtest_cnt, convert_unit_to_num(upbandwidthmeasured) as upbandwidth, convert_unit_to_
num(downbandwidthmeasured) as downbandwidth from $log where $filter and logid_to_int(logid) in
```

```
(22925, 22933, 22936, 22938) and interface is not null) t ) t group by timestamp, csf, devname,
dvid, devid, vd, interface, speedtestserver, healthcheck /*SkipSTART*/order by timestamp
desc/*SkipEND*/)### t group by hodex, dev_intf order by hodex
```

Dataset Name	Description	Log Category
sdwan-Device-Availability-status	SD-WAN Device Statistic by Bibandwidth	event

```
select
  devid,
  sum(bibandwidth)/ sum(count) as bibandwidth
from
  ###(select $flex_timestamp as timestamp, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck as sla_rule, sum(link_status) as link_status, sum(failed_latency) as
failed_latency, sum(failed_jitter) as failed_jitter,  sum(failed_packetloss) as failed_packetloss,
sum(latency) as latency, max(latency) as latency_max, min(latency) as latency_min, sum(jitter) as
jitter, max(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max
(packetloss) as packetloss_max, min(packetloss) as packetloss_min, sum(inbandwidth) as
inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as
count, sum(CASE WHEN link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_status) as sdwan_
status, sum(speedtest_cnt) as speedtest_cnt, sum(upbandwidth) as upbandwidth, sum(downbandwidth)
as downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE NULL END) AS latency, (CASE
WHEN link_status=1 THEN jitter ELSE NULL END) AS jitter, (CASE WHEN link_status=1 THEN packetloss
ELSE NULL END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END)
AS failed_packetloss, (CASE WHEN sla_failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed_
jitter, (CASE WHEN sla_failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed_latency, (CASE
WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status,  (CASE WHEN link_status=1 THEN
inbandwidth ELSE 0 END) AS inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth ELSE 0 END) AS
outbandwidth,  (CASE WHEN link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth, speedtest_
cnt, upbandwidth, downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, trim
(trailing 'ms' from latency)::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%'
THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS
sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num
(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE
WHEN logid_to_int(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num
(upbandwidthmeasured) as upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth
from $log where $filter and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is
not null) t ) t group by timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where $filter-drilldown and
bibandwidth is not null group by devid having sum(count)>0 order by bibandwidth desc
```

Dataset Name	Description	Log Category
sdwan-device-intf-availability-percentage-bar	SD-WAN Device Interface Availability Percentage	event

```
(
  select
    & #039;SD-WAN' as interface, cast(sum(availcnt)*100.0/sum(count) as decimal(18,2)) as
available from (select timestamp, devid, first_value(count) OVER (PARTITION BY timestamp, devid
ORDER BY link_status/count desc, count desc) as count, first_value(link_status) OVER (PARTITION BY
timestamp, devid ORDER BY link_status/count desc, count desc) as availcnt from (select timestamp,
devid, interface, sum(link_status) as link_status, sum(count) as count from ###(select $flex_
timestamp as timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver, healthcheck as
sla_rule, sum(link_status) as link_status, sum(failed_latency) as failed_latency, sum(failed_
jitter) as failed_jitter,  sum(failed_packetloss) as failed_packetloss, sum(latency) as latency,
max(latency) as latency_max, min(latency) as latency_min, sum(jitter) as jitter, max(jitter) as
jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max(packetloss) as
packetloss_max, min(packetloss) as packetloss_min, sum(inbandwidth) as inbandwidth, sum
(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN
link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_status) as sdwan_status, sum
(speedtest_cnt) as speedtest_cnt, sum(upbandwidth) as upbandwidth, sum(downbandwidth) as
downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE NULL END) AS latency, (CASE
WHEN link_status=1 THEN jitter ELSE NULL END) AS jitter, (CASE WHEN link_status=1 THEN packetloss
ELSE NULL END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END)
AS failed_packetloss, (CASE WHEN sla_failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed_
jitter, (CASE WHEN sla_failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed_latency, (CASE
WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status,  (CASE WHEN link_status=1 THEN
inbandwidth ELSE 0 END) AS inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth ELSE 0 END) AS
outbandwidth,  (CASE WHEN link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth, speedtest_
cnt, upbandwidth, downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, trim
(trailing 'ms' from latency)::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%'
THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS
sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num
(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE
WHEN logid_to_int(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num
(upbandwidthmeasured) as upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth
from $log where $filter and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is
not null) t ) t group by timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where $filter-drilldown and
count>0 group by timestamp, devid, interface)t) t group by interface) union all (select interface,
cast(sum(link_status)*100.0/sum(count) as decimal(18,2)) as available from ###(select $flex_
timestamp as timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver, healthcheck as
sla_rule, sum(link_status) as link_status, sum(failed_latency) as failed_latency, sum(failed_
jitter) as failed_jitter,  sum(failed_packetloss) as failed_packetloss, sum(latency) as latency,
max(latency) as latency_max, min(latency) as latency_min, sum(jitter) as jitter, max(jitter) as
jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max(packetloss) as
packetloss_max, min(packetloss) as packetloss_min, sum(inbandwidth) as inbandwidth, sum
(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN
link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_status) as sdwan_status, sum
(speedtest_cnt) as speedtest_cnt, sum(upbandwidth) as upbandwidth, sum(downbandwidth) as
downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE NULL END) AS latency, (CASE
WHEN link_status=1 THEN jitter ELSE NULL END) AS jitter, (CASE WHEN link_status=1 THEN packetloss
ELSE NULL END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END)
```

```
AS failed_packetloss, (CASE WHEN sla_failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed_
jitter, (CASE WHEN sla_failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed_latency, (CASE
WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status,  (CASE WHEN link_status=1 THEN
inbandwidth ELSE 0 END) AS inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth ELSE 0 END) AS
outbandwidth,  (CASE WHEN link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth, speedtest_
cnt, upbandwidth, downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, trim
(trailing 'ms' from latency)::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%'
THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS
sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num
(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE
WHEN logid_to_int(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num
(upbandwidthmeasured) as upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth
from $log where $filter and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is
not null) t ) t group by timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where $filter-drilldown group by
interface order by interface)
```

Dataset Name	Description	Log Category
sdwan-device-intf-availability-percentage-donut	SD-WAN Device Interface Availability Percentage Donut	event

```
select
  interface,
  unnest(avail) as avail,
  unnest(val) as val
from
  (
    select
      interface,
      array[ & #039;Available', 'Unavailable'] as avail, array[available, 100-available] as val
from ((select 'SD-WAN' as interface, cast(sum(availcnt)*100.0/sum(count) as decimal(18,2)) as
available from (select timestamp, devid, first_value(count) OVER (PARTITION BY timestamp, devid
ORDER BY link_status/count desc, count desc) as count, first_value(link_status) OVER (PARTITION BY
timestamp, devid ORDER BY link_status/count desc, count desc) as availcnt from (select timestamp,
devid, interface, sum(link_status) as link_status, sum(count) as count from ###(select $flex_
timestamp as timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver, healthcheck as
sla_rule, sum(link_status) as link_status, sum(failed_latency) as failed_latency, sum(failed_
jitter) as failed_jitter,  sum(failed_packetloss) as failed_packetloss, sum(latency) as latency,
max(latency) as latency_max, min(latency) as latency_min, sum(jitter) as jitter, max(jitter) as
jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max(packetloss) as
packetloss_max, min(packetloss) as packetloss_min, sum(inbandwidth) as inbandwidth, sum
(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN
link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_status) as sdwan_status, sum
(speedtest_cnt) as speedtest_cnt, sum(upbandwidth) as upbandwidth, sum(downbandwidth) as
downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE NULL END) AS latency, (CASE
WHEN link_status=1 THEN jitter ELSE NULL END) AS jitter, (CASE WHEN link_status=1 THEN packetloss
ELSE NULL END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END)
```

```
AS failed_packetloss, (CASE WHEN sla_failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed_
jitter, (CASE WHEN sla_failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed_latency, (CASE
WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status,  (CASE WHEN link_status=1 THEN
inbandwidth ELSE 0 END) AS inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth ELSE 0 END) AS
outbandwidth,  (CASE WHEN link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth, speedtest_
cnt, upbandwidth, downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, trim
(trailing 'ms' from latency)::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%'
THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS
sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num
(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE
WHEN logid_to_int(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num
(upbandwidthmeasured) as upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth
from $log where $filter and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is
not null) t ) t group by timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where $filter-drilldown and
count>0 group by timestamp, devid, interface)t) t group by interface) union all (select interface,
cast(sum(link_status)*100.0/sum(count) as decimal(18,2)) as available from ###(select $flex_
timestamp as timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver, healthcheck as
sla_rule, sum(link_status) as link_status, sum(failed_latency) as failed_latency, sum(failed_
jitter) as failed_jitter,  sum(failed_packetloss) as failed_packetloss, sum(latency) as latency,
max(latency) as latency_max, min(latency) as latency_min, sum(jitter) as jitter, max(jitter) as
jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max(packetloss) as
packetloss_max, min(packetloss) as packetloss_min, sum(inbandwidth) as inbandwidth, sum
(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN
link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_status) as sdwan_status, sum
(speedtest_cnt) as speedtest_cnt, sum(upbandwidth) as upbandwidth, sum(downbandwidth) as
downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE NULL END) AS latency, (CASE
WHEN link_status=1 THEN jitter ELSE NULL END) AS jitter, (CASE WHEN link_status=1 THEN packetloss
ELSE NULL END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END)
AS failed_packetloss, (CASE WHEN sla_failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed_
jitter, (CASE WHEN sla_failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed_latency, (CASE
WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status,  (CASE WHEN link_status=1 THEN
inbandwidth ELSE 0 END) AS inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth ELSE 0 END) AS
outbandwidth,  (CASE WHEN link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth, speedtest_
cnt, upbandwidth, downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, trim
(trailing 'ms' from latency)::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%'
THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS
sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num
(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE
WHEN logid_to_int(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num
(upbandwidthmeasured) as upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth
from $log where $filter and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is
not null) t ) t group by timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where $filter-drilldown group by
interface order by interface)) t) t
```

Dataset Name	Description	Log Category
sdwan-Device-Application-sdwan-Rules-and-Ports-drilldown	SD-WAN Device Statistic by Bibandwidth	event

```
select
  devid,
  sum(bibandwidth)/ sum(count) as bibandwidth
from
  ###(select $flex_timestamp as timestamp, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck as sla_rule, sum(link_status) as link_status, sum(failed_latency) as
failed_latency, sum(failed_jitter) as failed_jitter,  sum(failed_packetloss) as failed_packetloss,
sum(latency) as latency, max(latency) as latency_max, min(latency) as latency_min, sum(jitter) as
jitter, max(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max
(packetloss) as packetloss_max, min(packetloss) as packetloss_min, sum(inbandwidth) as
inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as
count, sum(CASE WHEN link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_status) as sdwan_
status, sum(speedtest_cnt) as speedtest_cnt, sum(upbandwidth) as upbandwidth, sum(downbandwidth)
as downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE NULL END) AS latency, (CASE
WHEN link_status=1 THEN jitter ELSE NULL END) AS jitter, (CASE WHEN link_status=1 THEN packetloss
ELSE NULL END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END)
AS failed_packetloss, (CASE WHEN sla_failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed_
jitter, (CASE WHEN sla_failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed_latency, (CASE
WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status,  (CASE WHEN link_status=1 THEN
inbandwidth ELSE 0 END) AS inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth ELSE 0 END) AS
outbandwidth,  (CASE WHEN link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth, speedtest_
cnt, upbandwidth, downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, trim
(trailing 'ms' from latency)::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%'
THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS
sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num
(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE
WHEN logid_to_int(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num
(upbandwidthmeasured) as upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth
from $log where $filter and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is
not null) t ) t group by timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where $filter-drilldown and
bibandwidth is not null group by devid having sum(count)>0 order by bibandwidth desc
```

Dataset Name	Description	Log Category
sdwan-Device-Interface-Application-Traffic-Sankey	Top SD-WAN application by bandwidth sankey	traffic

```
select
  & #039;SD-WAN Rules' as summary, 'Rule:' || coalesce(rulename, 'Implicit') as rule_name, app_
group, devid, dstintf as interface, sum(bandwidth) as bandwidth from ###(select $flex_timestamp as
timestamp, csf, devid, vd, srccountry, dstintf, srcintf, srcintfrole, dstintfrole, appid, appcat,
```

```
app_group_name(app) as app_group, coalesce(vwlservice,vwlname) as rulename, service, coalesce
(nullifna(`srcname`),ipstr(`srcip`),nullifna(`srcmac`)) as dev_src,  sum(crscore%65536) as
crscore, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, sum
(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce
(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out,
count(*) as sessions from $log-traffic where $filter and vwlid IS NOT NULL and (logflag&(1|32)>0)
group by timestamp, srccountry, dstintf, csf, devid, vd, srcintf, srcintfrole, dstintfrole, appid,
appcat, app_group, rulename, service, user_src, dev_src order by bandwidth desc)### t where
$filter-drilldown group by rule_name, app_group, devid, interface order by bandwidth desc
```

Dataset Name	Description	Log Category
sdwan-fw-Device-Interface-test3	SD-WAN Device-Interface Statistic	event

```
select
  devid,
  sum(bibandwidth)/ sum(count) as bibandwidth
from
  ###(select $flex_timestamp as timestamp, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck as sla_rule, sum(link_status) as link_status, sum(failed_latency) as
failed_latency, sum(failed_jitter) as failed_jitter,  sum(failed_packetloss) as failed_packetloss,
sum(latency) as latency, max(latency) as latency_max, min(latency) as latency_min, sum(jitter) as
jitter, max(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max
(packetloss) as packetloss_max, min(packetloss) as packetloss_min, sum(inbandwidth) as
inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as
count, sum(CASE WHEN link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_status) as sdwan_
status, sum(speedtest_cnt) as speedtest_cnt, sum(upbandwidth) as upbandwidth, sum(downbandwidth)
as downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE NULL END) AS latency, (CASE
WHEN link_status=1 THEN jitter ELSE NULL END) AS jitter, (CASE WHEN link_status=1 THEN packetloss
ELSE NULL END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END)
AS failed_packetloss, (CASE WHEN sla_failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed_
jitter, (CASE WHEN sla_failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed_latency, (CASE
WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status,  (CASE WHEN link_status=1 THEN
inbandwidth ELSE 0 END) AS inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth ELSE 0 END) AS
outbandwidth,  (CASE WHEN link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth, speedtest_
cnt, upbandwidth, downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, trim
(trailing 'ms' from latency)::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%'
THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS
sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num
(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE
WHEN logid_to_int(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num
(upbandwidthmeasured) as upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth
from $log where $filter and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is
not null) t ) t group by timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where $filter-drilldown and
bibandwidth is not null group by devid having sum(count)>0 order by bibandwidth desc
```

Dataset Name	Description	Log Category
sdwan-CTAP-Total-Bandwidth-Internal-And-External2	CTAP SD-WAN Internal and External Bandwidth	traffic

```
select
  dstintf as interface,
  coalesce(
    sum(bandwidth),
    0
  ) as bandwidth
from
  ###(select $flex_timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf,
srcintfrole, dstintfrole, appid, appcat, app_group_name(app) as app_group, coalesce
(vwlservice,vwlname) as rulename, service, coalesce(nullifna(`srcname`),ipstr(`srcip`),nullifna
(`srcmac`)) as dev_src,  sum(crscore%65536) as crscore, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from $log-traffic where
$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf,
devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_
src, dev_src order by bandwidth desc)### t where $filter-drilldown group by interface
```

Dataset Name	Description	Log Category
sdwan-Device-Intf-Avail-Percentage-Timeline	SD-WAN Device Interface Availability Percentage Timeline	event

```
select
  hodex,
  interface,
  available
from
  (
    (
      select
        $flex_datetime(timestamp) as hodex,
        & #039;SD-WAN' as interface, cast(sum(availcnt)*100.0/sum(count) as decimal(18,2)) as
available from (select timestamp, devid, first_value(count) OVER (PARTITION BY timestamp, devid
ORDER BY link_status/count desc, count desc) as count, first_value(link_status) OVER (PARTITION BY
timestamp, devid ORDER BY link_status/count desc, count desc) as availcnt from (select timestamp,
devid, interface, sum(link_status) as link_status, sum(count) as count from ###(select $flex_
timestamp as timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver, healthcheck as
sla_rule, sum(link_status) as link_status, sum(failed_latency) as failed_latency, sum(failed_
jitter) as failed_jitter,  sum(failed_packetloss) as failed_packetloss, sum(latency) as latency,
max(latency) as latency_max, min(latency) as latency_min, sum(jitter) as jitter, max(jitter) as
jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max(packetloss) as
packetloss_max, min(packetloss) as packetloss_min, sum(inbandwidth) as inbandwidth, sum
(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN
link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_status) as sdwan_status, sum
```

```
(speedtest_cnt) as speedtest_cnt, sum(upbandwidth) as upbandwidth, sum(downbandwidth) as
downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE NULL END) AS latency, (CASE
WHEN link_status=1 THEN jitter ELSE NULL END) AS jitter, (CASE WHEN link_status=1 THEN packetloss
ELSE NULL END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END)
AS failed_packetloss, (CASE WHEN sla_failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed_
jitter, (CASE WHEN sla_failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed_latency, (CASE
WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status,  (CASE WHEN link_status=1 THEN
inbandwidth ELSE 0 END) AS inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth ELSE 0 END) AS
outbandwidth,  (CASE WHEN link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth, speedtest_
cnt, upbandwidth, downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, trim
(trailing 'ms' from latency)::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%'
THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS
sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num
(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE
WHEN logid_to_int(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num
(upbandwidthmeasured) as upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth
from $log where $filter and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is
not null) t ) t group by timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where $filter-drilldown and
count>0 group by timestamp, devid, interface)t) t group by hodex order by hodex) union all (select
$flex_datetime(timestamp) as hodex, interface, cast(sum(link_status)*100.0/sum(count) as decimal
(18,2)) as available from ###(select $flex_timestamp as timestamp, csf, devname, dvid, devid, vd,
interface, speedtestserver, healthcheck as sla_rule, sum(link_status) as link_status, sum(failed_
latency) as failed_latency, sum(failed_jitter) as failed_jitter,  sum(failed_packetloss) as
failed_packetloss, sum(latency) as latency, max(latency) as latency_max, min(latency) as latency_
min, sum(jitter) as jitter, max(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss)
as packetloss, max(packetloss) as packetloss_max, min(packetloss) as packetloss_min, sum
(inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth,
count(*) as count, sum(CASE WHEN link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_
status) as sdwan_status, sum(speedtest_cnt) as speedtest_cnt, sum(upbandwidth) as upbandwidth, sum
(downbandwidth) as downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE NULL END) AS
latency, (CASE WHEN link_status=1 THEN jitter ELSE NULL END) AS jitter, (CASE WHEN link_status=1
THEN packetloss ELSE NULL END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN
1 ELSE 0 END) AS failed_packetloss, (CASE WHEN sla_failed=1 AND metric='jitter' THEN 1 ELSE 0 END)
AS failed_jitter, (CASE WHEN sla_failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed_
latency, (CASE WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status,  (CASE WHEN link_
status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth
ELSE 0 END) AS outbandwidth,  (CASE WHEN link_status=1 THEN bibandwidth ELSE 0 END) AS
bibandwidth, speedtest_cnt, upbandwidth, downbandwidth from (select itime, csf, devname, dvid,
devid, vd, interface, speedtestserver, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS
link_status, trim(trailing 'ms' from latency)::float as latency, jitter::float as jitter, trim
(trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE
'%SLA failed%' THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1
ELSE 0 END) AS sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_
num(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE
WHEN logid_to_int(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num
(upbandwidthmeasured) as upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth
from $log where $filter and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is
```

```
not null) t ) t group by timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where $filter-drilldown group by
hodex, interface order by hodex)) t order by hodex
```

Dataset Name	Description	Log Category
sdwan-Device-Intf-Inbandwidth-Timeline	SD-WAN Device-Interface Bandwidth Timeline	event

```
select
  time,
  interface,
  (bps_in)/(interval) as inbandwidth,
  (bps_out)/(interval) as outbandwidth,
  (bps_in + bps_out)/(interval) as bandwidth
from
  (
    select
      $flex_timescale(timestamp) as time,
      tb1.intfname as interface,
      sum(sentbps * interval) as bps_out,
      sum(rcvdbps * interval) as bps_in,
      sum(interval) as interval
    from
      $ADOM_INTF_STATS tb1
      inner join (
        select
          dvid,
          intfname
        from
          $ADOM_SDWAN_INTF_INFO
        where
          $filter - drilldown
          and $dev_filter
      ) tb2 on tb1.dvid = tb2.dvid
      and tb1.intfname = tb2.intfname
    where
      $cust_time_filter(timestamp)
    group by
      time,
      tb1.intfname
    order by
      time
  ) t
order by
  time
```

Dataset Name	Description	Log Category
sdwan-Device-Intf-Outbandwidth-Timeline	SD-WAN Device-Interface Bandwidth Timeline	event

```
select
  time,
  interface,
  (bps_in)/(interval) as inbandwidth,
  (bps_out)/(interval) as outbandwidth,
  (bps_in + bps_out)/(interval) as bandwidth
from
  (
    select
      $flex_timescale(timestamp) as time,
      tb1.intfname as interface,
      sum(sentbps * interval) as bps_out,
      sum(rcvdbps * interval) as bps_in,
      sum(interval) as interval
    from
      $ADOM_INTF_STATS tb1
      inner join (
        select
          dvid,
          intfname
        from
          $ADOM_SDWAN_INTF_INFO
        where
          $filter - drilldown
          and $dev_filter
      ) tb2 on tb1.dvid = tb2.dvid
      and tb1.intfname = tb2.intfname
    where
      $cust_time_filter(timestamp)
    group by
      time,
      tb1.intfname
    order by
      time
  ) t
order by
  time
```

Dataset Name	Description	Log Category
sdwan-Speed-Test-Upstream-Bandwidth-Line	SD-WAN Speed Test Upstream Bandwidth Timeline	event

```
select
  $flex_timescale(timestamp) as time,
```

```
  t1.interface,
  cast(
    sum(upbandwidth)/ sum(speedtest_cnt) as decimal(18, 2)
  ) as upbandwidth
from
  (
    select
      timestamp,
      devid,
      interface,
      sum(speedtest_cnt) as speedtest_cnt,
      sum(upbandwidth) as upbandwidth
    from
      ###(select $flex_timestamp as timestamp, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck as sla_rule, sum(link_status) as link_status, sum(failed_latency) as
failed_latency, sum(failed_jitter) as failed_jitter,  sum(failed_packetloss) as failed_packetloss,
sum(latency) as latency, max(latency) as latency_max, min(latency) as latency_min, sum(jitter) as
jitter, max(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max
(packetloss) as packetloss_max, min(packetloss) as packetloss_min, sum(inbandwidth) as
inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as
count, sum(CASE WHEN link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_status) as sdwan_
status, sum(speedtest_cnt) as speedtest_cnt, sum(upbandwidth) as upbandwidth, sum(downbandwidth)
as downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE NULL END) AS latency, (CASE
WHEN link_status=1 THEN jitter ELSE NULL END) AS jitter, (CASE WHEN link_status=1 THEN packetloss
ELSE NULL END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END)
AS failed_packetloss, (CASE WHEN sla_failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed_
jitter, (CASE WHEN sla_failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed_latency, (CASE
WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status,  (CASE WHEN link_status=1 THEN
inbandwidth ELSE 0 END) AS inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth ELSE 0 END) AS
outbandwidth,  (CASE WHEN link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth, speedtest_
cnt, upbandwidth, downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, trim
(trailing 'ms' from latency)::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%'
THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS
sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num
(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE
WHEN logid_to_int(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num
(upbandwidthmeasured) as upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth
from $log where $filter and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is
not null) t ) t group by timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where $filter-drilldown and
speedtest_cnt>0 group by timestamp, devid, interface) t1 inner join (select devid, interface, sum
(upbandwidth+downbandwidth)/sum(speedtest_cnt) as bandwitdth from ###(select $flex_timestamp as
timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver, healthcheck as sla_rule, sum
(link_status) as link_status, sum(failed_latency) as failed_latency, sum(failed_jitter) as failed_
jitter,  sum(failed_packetloss) as failed_packetloss, sum(latency) as latency, max(latency) as
latency_max, min(latency) as latency_min, sum(jitter) as jitter, max(jitter) as jitter_max, min
(jitter) as jitter_min, sum(packetloss) as packetloss, max(packetloss) as packetloss_max, min
(packetloss) as packetloss_min, sum(inbandwidth) as inbandwidth, sum(outbandwidth) as
outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN link_status=1 THEN
```

```
1 ELSE 0 END) AS count_linkup, min(sdwan_status) as sdwan_status, sum(speedtest_cnt) as speedtest_
cnt, sum(upbandwidth) as upbandwidth, sum(downbandwidth) as downbandwidth from (select itime, csf,
devname, dvid, devid, vd, interface, speedtestserver, healthcheck, link_status, (CASE WHEN link_
status=1 THEN latency ELSE NULL END) AS latency, (CASE WHEN link_status=1 THEN jitter ELSE NULL
END) AS jitter, (CASE WHEN link_status=1 THEN packetloss ELSE NULL END) AS packetloss, (CASE WHEN
sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed_packetloss, (CASE WHEN sla_
failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed_jitter, (CASE WHEN sla_failed=1 AND
metric='latency' THEN 1 ELSE 0 END) AS failed_latency, (CASE WHEN sla_failed=1 THEN 3 ELSE sdwan_
status END) AS sdwan_status,  (CASE WHEN link_status=1 THEN inbandwidth ELSE 0 END) AS
inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth,  (CASE WHEN
link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth, speedtest_cnt, upbandwidth,
downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, trim(trailing 'ms' from
latency)::float as latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as
packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS
sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan_status,
convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num(outbandwidthused) as
outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE WHEN logid_to_int
(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num(upbandwidthmeasured) as
upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth from $log where $filter
and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is not null) t ) t group by
timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver, healthcheck
/*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where $filter-drilldown and speedtest_cnt>0
group by devid, interface order by bandwitdth desc limit $ddown-top)t2 on
t1.interface=t2.interface and t1.devid=t2.devid group by time, t1.interface order by time
```

Dataset Name	Description	Log Category
sdwan-Speed-Test-Downstream-Bandwidth-Line	SD-WAN Speed Test Downstream Bandwidth Timeline	event

```
select
  $flex_timescale(timestamp) as time,
  t1.interface,
  cast(
    sum(downbandwidth)/ sum(speedtest_cnt) as decimal(18, 2)
  ) as downbandwidth
from
  (
    select
      timestamp,
      devid,
      interface,
      sum(speedtest_cnt) as speedtest_cnt,
      sum(downbandwidth) as downbandwidth
    from
      ###(select $flex_timestamp as timestamp, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck as sla_rule, sum(link_status) as link_status, sum(failed_latency) as
failed_latency, sum(failed_jitter) as failed_jitter,  sum(failed_packetloss) as failed_packetloss,
sum(latency) as latency, max(latency) as latency_max, min(latency) as latency_min, sum(jitter) as
jitter, max(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max
```

```
(packetloss) as packetloss_max, min(packetloss) as packetloss_min, sum(inbandwidth) as
inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as
count, sum(CASE WHEN link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_status) as sdwan_
status, sum(speedtest_cnt) as speedtest_cnt, sum(upbandwidth) as upbandwidth, sum(downbandwidth)
as downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE NULL END) AS latency, (CASE
WHEN link_status=1 THEN jitter ELSE NULL END) AS jitter, (CASE WHEN link_status=1 THEN packetloss
ELSE NULL END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END)
AS failed_packetloss, (CASE WHEN sla_failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed_
jitter, (CASE WHEN sla_failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed_latency, (CASE
WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status,  (CASE WHEN link_status=1 THEN
inbandwidth ELSE 0 END) AS inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth ELSE 0 END) AS
outbandwidth,  (CASE WHEN link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth, speedtest_
cnt, upbandwidth, downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, trim
(trailing 'ms' from latency)::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%'
THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS
sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num
(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE
WHEN logid_to_int(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num
(upbandwidthmeasured) as upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth
from $log where $filter and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is
not null) t ) t group by timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where $filter-drilldown and
speedtest_cnt>0 group by timestamp, devid, interface) t1 inner join (select devid, interface, sum
(downbandwidth+downbandwidth)/sum(speedtest_cnt) as bandwitdth from ###(select $flex_timestamp as
timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver, healthcheck as sla_rule, sum
(link_status) as link_status, sum(failed_latency) as failed_latency, sum(failed_jitter) as failed_
jitter,  sum(failed_packetloss) as failed_packetloss, sum(latency) as latency, max(latency) as
latency_max, min(latency) as latency_min, sum(jitter) as jitter, max(jitter) as jitter_max, min
(jitter) as jitter_min, sum(packetloss) as packetloss, max(packetloss) as packetloss_max, min
(packetloss) as packetloss_min, sum(inbandwidth) as inbandwidth, sum(outbandwidth) as
outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN link_status=1 THEN
1 ELSE 0 END) AS count_linkup, min(sdwan_status) as sdwan_status, sum(speedtest_cnt) as speedtest_
cnt, sum(upbandwidth) as upbandwidth, sum(downbandwidth) as downbandwidth from (select itime, csf,
devname, dvid, devid, vd, interface, speedtestserver, healthcheck, link_status, (CASE WHEN link_
status=1 THEN latency ELSE NULL END) AS latency, (CASE WHEN link_status=1 THEN jitter ELSE NULL
END) AS jitter, (CASE WHEN link_status=1 THEN packetloss ELSE NULL END) AS packetloss, (CASE WHEN
sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed_packetloss, (CASE WHEN sla_
failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed_jitter, (CASE WHEN sla_failed=1 AND
metric='latency' THEN 1 ELSE 0 END) AS failed_latency, (CASE WHEN sla_failed=1 THEN 3 ELSE sdwan_
status END) AS sdwan_status,  (CASE WHEN link_status=1 THEN inbandwidth ELSE 0 END) AS
inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth,  (CASE WHEN
link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth, speedtest_cnt, upbandwidth,
downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, trim(trailing 'ms' from
latency)::float as latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as
packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS
sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan_status,
convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num(outbandwidthused) as
outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE WHEN logid_to_int
```

```
(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num(upbandwidthmeasured) as
upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth from $log where $filter
and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is not null) t ) t group by
timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver, healthcheck
/*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where $filter-drilldown and speedtest_cnt>0
group by devid, interface order by bandwitdth desc limit $ddown-top)t2 on
t1.interface=t2.interface and t1.devid=t2.devid group by time, t1.interface order by time
```

Dataset Name	Description	Log Category
sdwan-Speed-Test-Dev-Bandwidth	SD-WAN Speed Test Device Bandwidth	event

```
select
  devid,
  unnest(up_down_stream) as up_down_stream,
  unnest(bandwidth) as bandwidth
from
  (
    select
      devid,
      array[ & #039;Upstream','Downstream'] as up_down_stream, array[cast(sum(upbandwidth) as
decimal(18,2)), cast(sum(downbandwidth) as decimal(18,2))] as bandwidth from (select devid,
interface, sum(upbandwidth)/sum(speedtest_cnt) as upbandwidth, sum(downbandwidth)/sum(speedtest_
cnt) as downbandwidth, sum(downbandwidth+upbandwidth)/sum(speedtest_cnt) as bandwidth from ###
(select $flex_timestamp as timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck as sla_rule, sum(link_status) as link_status, sum(failed_latency) as failed_latency,
sum(failed_jitter) as failed_jitter,  sum(failed_packetloss) as failed_packetloss, sum(latency) as
latency, max(latency) as latency_max, min(latency) as latency_min, sum(jitter) as jitter, max
(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max(packetloss)
as packetloss_max, min(packetloss) as packetloss_min, sum(inbandwidth) as inbandwidth, sum
(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN
link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_status) as sdwan_status, sum
(speedtest_cnt) as speedtest_cnt, sum(upbandwidth) as upbandwidth, sum(downbandwidth) as
downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE NULL END) AS latency, (CASE
WHEN link_status=1 THEN jitter ELSE NULL END) AS jitter, (CASE WHEN link_status=1 THEN packetloss
ELSE NULL END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END)
AS failed_packetloss, (CASE WHEN sla_failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed_
jitter, (CASE WHEN sla_failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed_latency, (CASE
WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status,  (CASE WHEN link_status=1 THEN
inbandwidth ELSE 0 END) AS inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth ELSE 0 END) AS
outbandwidth,  (CASE WHEN link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth, speedtest_
cnt, upbandwidth, downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, trim
(trailing 'ms' from latency)::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%'
THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS
sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num
(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE
WHEN logid_to_int(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num
(upbandwidthmeasured) as upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth
```

```
from $log where $filter and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is
not null) t ) t group by timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where $filter-drilldown and
speedtest_cnt>0 group by devid, interface) t1 group by devid) t order  by bandwidth desc
```

Dataset Name	Description	Log Category
sdwan-Speed-Test-Summary	SD-WAN Speed Test Summary	event

```
select
  devname,
  devid,
  interface,
  speedtestserver,
  cast(
    sum(downbandwidth)/ sum(speedtest_cnt)/ 1000 as decimal(18, 2)
  ) as downbandwidth,
  cast(
    sum(upbandwidth)/ sum(speedtest_cnt)/ 1000 as decimal(18, 2)
  ) as upbandwidth,
  cast(
    sum(downbandwidth + upbandwidth)/ sum(speedtest_cnt) as decimal(18, 2)
  ) as bandwidth
from
  ###(select $flex_timestamp as timestamp, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck as sla_rule, sum(link_status) as link_status, sum(failed_latency) as
failed_latency, sum(failed_jitter) as failed_jitter,  sum(failed_packetloss) as failed_packetloss,
sum(latency) as latency, max(latency) as latency_max, min(latency) as latency_min, sum(jitter) as
jitter, max(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max
(packetloss) as packetloss_max, min(packetloss) as packetloss_min, sum(inbandwidth) as
inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as
count, sum(CASE WHEN link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_status) as sdwan_
status, sum(speedtest_cnt) as speedtest_cnt, sum(upbandwidth) as upbandwidth, sum(downbandwidth)
as downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE NULL END) AS latency, (CASE
WHEN link_status=1 THEN jitter ELSE NULL END) AS jitter, (CASE WHEN link_status=1 THEN packetloss
ELSE NULL END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END)
AS failed_packetloss, (CASE WHEN sla_failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed_
jitter, (CASE WHEN sla_failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed_latency, (CASE
WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status,  (CASE WHEN link_status=1 THEN
inbandwidth ELSE 0 END) AS inbandwidth,  (CASE WHEN link_status=1 THEN outbandwidth ELSE 0 END) AS
outbandwidth,  (CASE WHEN link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth, speedtest_
cnt, upbandwidth, downbandwidth from (select itime, csf, devname, dvid, devid, vd, interface,
speedtestserver, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, trim
(trailing 'ms' from latency)::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%'
THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS
sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num
(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth, (CASE
WHEN logid_to_int(logid)=22938 THEN 1 ELSE 0 END) AS speedtest_cnt, convert_unit_to_num
(upbandwidthmeasured) as upbandwidth, convert_unit_to_num(downbandwidthmeasured) as downbandwidth
```

```
from $log where $filter and logid_to_int(logid) in (22925, 22933, 22936, 22938) and interface is
not null) t ) t group by timestamp, csf, devname, dvid, devid, vd, interface, speedtestserver,
healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where $filter-drilldown and
speedtest_cnt>0 group by devname, devid, interface, speedtestserver order by bandwidth desc
```

Dataset Name	Description	Log Category
Top-Web-Sites-by-Bandwidth	Top web sites by bandwidth usage	webfilter

```
select
  domain,
  sum(bandwidth) as bandwidth
from
  ###(select coalesce(nullifna(hostname), ipstr(`dstip`)) as domain, sum(coalesce(sentbyte,
0)+coalesce(rcvdbyte, 0)) as bandwidth from $log-traffic where $filter and (logflag&1>0) and
(countweb>0 or ((logver is null or logver<502000000) and (hostname is not null or utmevent in
('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter')))) group by domain
having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc)### t group by
domain order by bandwidth desc
```

Dataset Name	Description	Log Category
Top-App-Category-by-Session	Application risk application usage by category	traffic

```
select
  appcat,
  sum(sessions) as total_num
from
  ###(select appid, app, appcat, apprisk, sum(bandwidth) as bandwidth, sum(sessions) as sessions
from ###base(/*tag:rpt_base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip,
epid, euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
service, appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_
in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte,
0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END)
as sessions from $log-traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null
group by timestamp, dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat,
apprisk, hostname order by bandwidth desc, sessions desc)base### t group by appid, app, appcat,
apprisk /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t where $filter-
drilldown group by appcat order by total_num desc
```

Dataset Name	Description	Log Category
Top-Region-Name-by-Traffic	Traffic top destination countries by browsing time	traffic

```
select
  dstcountry,
  sum(bandwidth) as bandwidth
from
```

```
  ###(select dstcountry, ebtr_agg_flat(browsetime) as browsetime, sum(bandwidth) as bandwidth, sum
(traffic_in) as traffic_in, sum(traffic_out) as traffic_out from (select dstcountry, ebtr_agg_flat
($browse_time) as browsetime, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum
(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out from $log where
$filter and (logflag&1>0) and $browse_time is not null group by dstcountry) t group by dstcountry
/*SkipSTART*/order by ebtr_value(ebtr_agg_flat(browsetime), null, null) desc/*SkipEND*/)### t
where $filter-drilldown group by dstcountry order by bandwidth desc
```

Dataset Name	Description	Log Category
Top-App-By-Bandwidth-Chart	Top applications by bandwidth usage	traffic

```
select
  app_group_name(app) as app_group,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out,
  sum(sessions) as sessions
from
  ###(select appid, app, appcat, apprisk, user_src, hostname, dstip, sum(traffic_in) as traffic_
in, sum(traffic_out) as traffic_out, sum(bandwidth) as bandwidth, sum(sessions) as sessions from
###base(/*tag:rpt_base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid,
euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service,
appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions
from $log-traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by
timestamp, dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat, apprisk,
hostname order by bandwidth desc, sessions desc)base### t group by appid, app, appcat, apprisk,
user_src, hostname, dstip /*SkipSTART*/order by bandwidth desc, sessions desc/*SkipEND*/)### t
group by app_group having sum(bandwidth)>0 order by bandwidth desc
```

Dataset Name	Description	Log Category
Top-Protocols-By-Traffic	Top applications by bandwidth usage	traffic

```
select
  service,
  sum(bandwidth) as bandwidth
from
  ###(select service, sum(bandwidth) as bandwidth from ###base(/*tag:rpt_base_t_bndwdth_
sess*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid, euid, appcat, apprisk,
coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service, sum(CASE
WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum
(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in from $log-traffic where $filter and (logflag&
(1|32)>0) group by timestamp, dvid, srcip, dstip, epid, euid, appcat, apprisk, user_src, service
/*SkipSTART*/order by bandwidth desc, sessions desc/*SkipEND*/)base### base_query group by service
order by bandwidth desc)### t where $filter-drilldown group by service order by bandwidth desc
```

Dataset Name	Description	Log Category
Top-Web-Sites-by-Sessions	Top web sites by session count	webfilter

```
select
  domain,
  sum(sessions) as sessions
from
  ###(select  coalesce(nullifna(hostname), ipstr(`dstip`)) as domain, count(*) as sessions from $log where $filter group by domain order by sessions desc)### t group by domain order by sessions desc
```

Dataset Name	Description	Log Category
Top-Attacks-by-Count	Threat attacks by severity	attack

```
select
  attack,
  sum(attack_count) as totalnum
from
  ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, attack, (case when severity in ('critical', 'high') then 1 else 0 end) as high_severity, count(*) as attack_count from $log where $filter and nullifna(attack) is not null group by user_src, attack, high_severity order by attack_count desc)### t where $filter-drilldown and attack is not null group by attack order by totalnum desc
```

Dataset Name	Description	Log Category
Top-Spams-by-Count	User drilldown top spam sources	emailfilter

```
select
  user_src,
  sum(totalnum) as totalnum
from
  ###(select $flex_timestamp as timestamp, coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, `from` as mf_sender, `to` as mf_receiver, action, eventtype, count(*) as totalnum from $log where $filter group by timestamp, user_src, mf_sender, mf_receiver, action, eventtype /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where $filter-drilldown and mf_sender is not null group by user_src order by totalnum desc
```

Dataset Name	Description	Log Category
utm-Top-Virus-Count	UTM top virus	virus

```
select
  virus,
  max(virusid_s) as virusid,
```

```
  (
    case when virus like & #039;Riskware%' then 'Spyware' when virus like 'Adware%' then 'Adware'
else 'Virus' end) as malware_type, sum(totalnum) as totalnum from ###(select virus, virusid_to_str
(virusid, eventtype) as virusid_s, count(*) as totalnum from $log where $filter and nullifna
(virus) is not null group by virus, virusid_s /*SkipSTART*/order by totalnum desc/*SkipEND*/)### t
group by virus, malware_type order by totalnum desc
```

Dataset Name	Description	Log Category
security-Antivirus-Inspections	Antivirus Inspections	virus

```
select
  action,
  sum(totalnum) as totalnum
from
  ###(select $flex_timestamp as timestamp, coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src,
`from` as mf_sender, `to` as mf_receiver, action, eventtype, count(*) as totalnum from $log where
$filter group by timestamp, user_src, mf_sender, mf_receiver, action, eventtype /*SkipSTART*/order
by timestamp desc/*SkipEND*/)### t where $filter-drilldown and action is not null group by action
order by totalnum desc
```

Dataset Name	Description	Log Category
Top-DLP-by-Count	Email DLP Activity Summary	dlp

```
select
  profile,
  count(*) as total_num
from
  ###(select itime, hostname,`from` as sender, `to` as receiver, profile, action, service,
subtype, srcip, dstip, severity, filename, direction, filesize, (case when severity='critical'
then 'Critical Data Exfiltration' else (case when coalesce(nullifna(`user`), ipstr(`srcip`)) is
not null then 'User Associated Data Loss' else NULL end) end) as data_loss from $log where $filter
/*SkipSTART*/order by itime desc/*SkipEND*/)### t where $filter-drilldown and profile is not null
group by profile order by total_num desc
```

Dataset Name	Description	Log Category
wifi-Top-AP-By-Client	Top access point by client	traffic

```
select
  ap_srcintf as srcintf,
  count(distinct srcmac) as totalnum
from
  (
    select
      coalesce(ap, srcintf) as ap_srcintf,
      srcmac
```

```
    from
      ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
ap, srcintf, srcssid, srcssid as ssid, srcmac, srcmac as stamac, coalesce(nullifna(`srcname`),
`srcmac`) as hostname_mac, max(srcswversion) as srcswversion, max(osname) as osname, max
(osversion) as osversion, max(devtype) as devtype, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte,
0)) as bandwidth, count(*) as subtotal from $log-traffic where $filter and (logflag&1>0) and
(srcssid is not null or dstssid is not null) group by user_src, ap, srcintf, srcssid, srcmac,
hostname_mac /*SkipSTART*/order by bandwidth desc, subtotal desc/*SkipEND*/)### t where srcmac is
not null group by ap_srcintf, srcmac union all (select ap as ap_srcintf, stamac as srcmac from ###
(select $flex_timestamp as timestamp, stamac, stamac as srcmac, ap, ssid, ssid as srcssid, user_
src, sum(coalesce(sentdelta, 0)) as sentdelta, sum(coalesce(rcvddelta, 0)) as rcvddelta, sum
(coalesce(sentdelta, 0)+coalesce(rcvddelta, 0)) as bandwidth from (select itime, stamac, ap, ssid,
coalesce(`user`, ipstr(`srcip`)) as user_src, sentbyte-lag(coalesce(sentbyte, 0)) over (partition
by stamac order by itime) as sentdelta, rcvdbyte-lag(coalesce(rcvdbyte, 0)) over (partition by
stamac order by itime) as rcvddelta from $log-event where $filter and subtype='wireless' and
stamac is not null and ssid is not null and action in ('sta-wl-bridge-traffic-stats', 'reassoc-
req', 'assoc-req')) as t group by timestamp, stamac, ap, ssid, user_src /*SkipSTART*/order by
bandwidth desc/*SkipEND*/)### t where stamac is not null group by ap, stamac)) t group by srcintf
order by totalnum desc
```

Dataset Name	Description	Log Category
wifi-Top-AP-By-Bandwidth	Top access point by bandwidth usage	traffic

```
select
  ap_srcintf,
  sum(bandwidth) as bandwidth
from
  (
    select
      coalesce(ap, srcintf) as ap_srcintf,
      sum(bandwidth) as bandwidth
    from
      ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
ap, srcintf, srcssid, srcssid as ssid, srcmac, srcmac as stamac, coalesce(nullifna(`srcname`),
`srcmac`) as hostname_mac, max(srcswversion) as srcswversion, max(osname) as osname, max
(osversion) as osversion, max(devtype) as devtype, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte,
0)) as bandwidth, count(*) as subtotal from $log-traffic where $filter and (logflag&1>0) and
(srcssid is not null or dstssid is not null) group by user_src, ap, srcintf, srcssid, srcmac,
hostname_mac /*SkipSTART*/order by bandwidth desc, subtotal desc/*SkipEND*/)### t group by ap_
srcintf having sum(bandwidth)>0 union all select ap as ap_srcintf, sum(bandwidth) as bandwidth
from ###(select $flex_timestamp as timestamp, stamac, stamac as srcmac, ap, ssid, ssid as srcssid,
user_src, sum(coalesce(sentdelta, 0)) as sentdelta, sum(coalesce(rcvddelta, 0)) as rcvddelta, sum
(coalesce(sentdelta, 0)+coalesce(rcvddelta, 0)) as bandwidth from (select itime, stamac, ap, ssid,
coalesce(`user`, ipstr(`srcip`)) as user_src, sentbyte-lag(coalesce(sentbyte, 0)) over (partition
by stamac order by itime) as sentdelta, rcvdbyte-lag(coalesce(rcvdbyte, 0)) over (partition by
stamac order by itime) as rcvddelta from $log-event where $filter and subtype='wireless' and
stamac is not null and ssid is not null and action in ('sta-wl-bridge-traffic-stats', 'reassoc-
req', 'assoc-req')) as t group by timestamp, stamac, ap, ssid, user_src /*SkipSTART*/order by
bandwidth desc/*SkipEND*/)### t group by ap having sum(bandwidth)>0) t group by ap_srcintf order
by bandwidth desc
```

Dataset Name	Description	Log Category
wifi-Top-SSID-By-Bandwidth	Top SSIDs by bandwidth usage	traffic

```
select
  srcssid,
  sum(bandwidth) as bandwidth
from
  (
    select
      srcssid,
      sum(bandwidth) as bandwidth
    from
      ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
ap, srcintf, srcssid, srcssid as ssid, srcmac, srcmac as stamac, coalesce(nullifna(`srcname`),
`srcmac`) as hostname_mac, max(srcswversion) as srcswversion, max(osname) as osname, max
(osversion) as osversion, max(devtype) as devtype, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte,
0)) as bandwidth, count(*) as subtotal from $log-traffic where $filter and (logflag&1>0) and
(srcssid is not null or dstssid is not null) group by user_src, ap, srcintf, srcssid, srcmac,
hostname_mac /*SkipSTART*/order by bandwidth desc, subtotal desc/*SkipEND*/)### t where srcssid is
not null group by srcssid having sum(bandwidth)>0 union all select ssid as srcssid, sum(bandwidth)
as bandwidth from ###(select $flex_timestamp as timestamp, stamac, stamac as srcmac, ap, ssid,
ssid as srcssid, user_src, sum(coalesce(sentdelta, 0)) as sentdelta, sum(coalesce(rcvddelta, 0))
as rcvddelta, sum(coalesce(sentdelta, 0)+coalesce(rcvddelta, 0)) as bandwidth from (select itime,
stamac, ap, ssid, coalesce(`user`, ipstr(`srcip`)) as user_src, sentbyte-lag(coalesce(sentbyte,
0)) over (partition by stamac order by itime) as sentdelta, rcvdbyte-lag(coalesce(rcvdbyte, 0))
over (partition by stamac order by itime) as rcvddelta from $log-event where $filter and
subtype='wireless' and stamac is not null and ssid is not null and action in ('sta-wl-bridge-
traffic-stats', 'reassoc-req', 'assoc-req')) as t group by timestamp, stamac, ap, ssid, user_src
/*SkipSTART*/order by bandwidth desc/*SkipEND*/)### t group by ssid having sum(bandwidth)>0) t
group by srcssid order by bandwidth desc
```

Dataset Name	Description	Log Category
sdwan-CTAP-Total-Bandwidth-Internal-And-External	CTAP SD-WAN Internal and External Bandwidth	traffic

```
select
  dstintf as interface,
  coalesce(
    sum(bandwidth),
    0
  ) as bandwidth
from
  ###(select $flex_timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf,
srcintfrole, dstintfrole, appid, appcat, app_group_name(app) as app_group, coalesce
(vwlservice,vwlname) as rulename, service, coalesce(nullifna(`srcname`),ipstr(`srcip`),nullifna
(`srcmac`)) as dev_src,  sum(crscore%65536) as crscore, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
```

```
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from $log-traffic where
$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf,
devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_
src, dev_src order by bandwidth desc)### t where $filter-drilldown group by interface
```

Dataset Name	Description	Log Category
sdwan-CTAP-Total-Bandwidth-External-Business-nonBusiness-Network	CTAP SD-WAN Bandwidth of External Business and nonBusiness	traffic

```
select
  (
    case when appcat not in (
      & #039;Network.Service',
'Mobile','Social.Media','Proxy','Video\/Audio','Game','P2P','unknown') then 'Business' when appcat
in ('Mobile','Social.Media','Proxy','Video\/Audio','Game','P2P','unknown') then 'nonBusiness'when
appcat in ('Network.Service') then 'Network Service' end) as app_cat, coalesce(sum(bandwidth), 0)
as bandwidth from ###(select $flex_timestamp as timestamp, csf, devid, vd, srccountry, dstintf,
srcintf, srcintfrole, dstintfrole, appid, appcat, app_group_name(app) as app_group, coalesce
(vwlservice,vwlname) as rulename, service, coalesce(nullifna(`srcname`),ipstr(`srcip`),nullifna
(`srcmac`)) as dev_src,  sum(crscore%65536) as crscore, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from $log-traffic where
$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf,
devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_
src, dev_src order by bandwidth desc)### t where $filter-drilldown group by app_cat order by
bandwidth desc
```

Dataset Name	Description	Log Category
sdwan-CTAP-Top-Appcat-Appgroup-By-Bandwidth-Sankey	CTAP SD-WAN Top SD-WAN application by bandwidth usage	traffic

```
select
  & #039;External' as summary, appcat, app_group, sum(bandwidth) as bandwidth from ###(select
$flex_timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf, srcintfrole,
dstintfrole, appid, appcat, app_group_name(app) as app_group, coalesce(vwlservice,vwlname) as
rulename, service, coalesce(nullifna(`srcname`),ipstr(`srcip`),nullifna(`srcmac`)) as dev_src,
sum(crscore%65536) as crscore, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`))
as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth,
sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as
traffic_out, count(*) as sessions from $log-traffic where $filter and vwlid IS NOT NULL and
(logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf, devid, vd, srcintf, srcintfrole,
dstintfrole, appid, appcat, app_group, rulename, service, user_src, dev_src order by bandwidth
desc)### t where $filter-drilldown and bandwidth>0 group by appcat, app_group order by bandwidth
desc
```

Dataset Name	Description	Log Category
sdwan-CTAP-Business-Apps-Bandwidth	CTAP SD-WAN Business Application with Bandwidth	traffic

```
select
  app_group,
  sum(bandwidth) as bandwidth
from
  ###(select $flex_timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf,
srcintfrole, dstintfrole, appid, appcat, app_group_name(app) as app_group, coalesce
(vwlservice,vwlname) as rulename, service, coalesce(nullifna(`srcname`),ipstr(`srcip`),nullifna
(`srcmac`)) as dev_src,  sum(crscore%65536) as crscore, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from $log-traffic where
$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf,
devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_
src, dev_src order by bandwidth desc)### t1 inner join app_mdata t2 on lower(t1.app_group)=lower
(t2.name) where $filter-drilldown and appcat not in ('Network.Service',
'Mobile','Social.Media','Proxy','Video\/Audio','Game','P2P','unknown') group by app_group order by
bandwidth desc, app_group
```

Dataset Name	Description	Log Category
sdwan-CTAP-Cloud-IT-Apps-Bandwidth	CTAP SD-WAN Cloud IT Application Bandwidth	traffic

```
select
  app_group,
  sum(bandwidth) as bandwidth
from
  ###(select $flex_timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf,
srcintfrole, dstintfrole, appid, appcat, app_group_name(app) as app_group, coalesce
(vwlservice,vwlname) as rulename, service, coalesce(nullifna(`srcname`),ipstr(`srcip`),nullifna
(`srcmac`)) as dev_src,  sum(crscore%65536) as crscore, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from $log-traffic where
$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf,
devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_
src, dev_src order by bandwidth desc)### t where $filter-drilldown and appcat='Cloud.IT' and
bandwidth>0 group by app_group order by bandwidth desc
```

Dataset Name	Description	Log Category
sdwan-CTAP-Storage-Backup-Apps-Bandwidth	CTAP SD-WAN Storage Backup Application Bandwidth	traffic

```
select
  app_group,
  sum(bandwidth) as bandwidth
from
  ###(select $flex_timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf,
srcintfrole, dstintfrole, appid, appcat, app_group_name(app) as app_group, coalesce
(vwlservice,vwlname) as rulename, service, coalesce(nullifna(`srcname`),ipstr(`srcip`),nullifna
(`srcmac`)) as dev_src,  sum(crscore%65536) as crscore, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from $log-traffic where
$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf,
devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_
src, dev_src order by bandwidth desc)### t where $filter-drilldown and appcat='Storage.Backup' and
bandwidth>0 group by app_group order by bandwidth desc
```

Dataset Name	Description	Log Category
sdwan-CTAP-Collaboration-Apps-Bandwidth	CTAP SD-WAN Collaboration Application Bandwidth	traffic

```
select
  app_group,
  sum(bandwidth) as bandwidth
from
  ###(select $flex_timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf,
srcintfrole, dstintfrole, appid, appcat, app_group_name(app) as app_group, coalesce
(vwlservice,vwlname) as rulename, service, coalesce(nullifna(`srcname`),ipstr(`srcip`),nullifna
(`srcmac`)) as dev_src,  sum(crscore%65536) as crscore, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from $log-traffic where
$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf,
devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_
src, dev_src order by bandwidth desc)### t where $filter-drilldown and appcat='Collaboration' and
bandwidth>0 group by app_group order by bandwidth desc
```

Dataset Name	Description	Log Category
sdwan-CTAP-Top-Streaming-App-By-Bandwidth	CTAP SD-WAN Top Streaming Application by Bandwidth	traffic

```
select
  app_group,
  sum(bandwidth) as bandwidth
from
  ###(select $flex_timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf,
srcintfrole, dstintfrole, appid, appcat, app_group_name(app) as app_group, coalesce
(vwlservice,vwlname) as rulename, service, coalesce(nullifna(`srcname`),ipstr(`srcip`),nullifna
```

```
(`srcmac`)) as dev_src,  sum(crscore%65536) as crscore, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from $log-traffic where
$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf,
devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_
src, dev_src order by bandwidth desc)### t where $filter-drilldown and appcat='Video\/Audio' and
bandwidth>0 group by app_group order by bandwidth desc
```

Dataset Name	Description	Log Category
sdwan-CTAP-Top-SocialMedia-App-By-Bandwidth	CTAP SD-WAN Top SocialMedia Application by Bandwidth	traffic

```
select
  app_group,
  sum(bandwidth) as bandwidth
from
  ###(select $flex_timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf,
srcintfrole, dstintfrole, appid, appcat, app_group_name(app) as app_group, coalesce
(vwlservice,vwlname) as rulename, service, coalesce(nullifna(`srcname`),ipstr(`srcip`),nullifna
(`srcmac`)) as dev_src,  sum(crscore%65536) as crscore, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from $log-traffic where
$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf,
devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_
src, dev_src order by bandwidth desc)### t where $filter-drilldown and appcat='Social.Media' and
bandwidth>0 group by app_group order by bandwidth desc
```

Dataset Name	Description	Log Category
sdwan-CTAP-App-Risk-Reputation-Top-Devices-By-Scores	Reputation Top Devices By-Scores	traffic

```
select
  coalesce(
    nullifna(`srcname`),
    ipstr(`srcip`),
    nullifna(`srcmac`)
  ) as dev_src,
  sum(crscore % 65536) as scores
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
```

```
  and crscore is not null
group by
  dev_src
having
  sum(crscore % 65536)> 0
order by
  scores desc
```

Dataset Name	Description	Log Category
sdwan-CTAP-SB-Top-Sandbox-Files	CTAP SD-WAN Sandbox Top Sandbox Files	virus

```
select
  filename,
  analyticscksum,
  service,
  sum(totalnum) as total_num,
  (
    case fsaverdict when & #039;malicious' then 'Malicious'  when 'high risk' then 'High' when
'medium risk' then 'Medium'  when 'low risk' then 'Low' else 'Other' end) as risk, (case
fsaverdict when 'malicious' then 5 when 'high risk' then 4 when 'medium risk' then 3 when 'low
risk' then 2 else 1 end) as risk_level from ###(select filename, analyticscksum, service,
fsaverdict, dtype, coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, virus, virusid_to_str
(virusid, eventtype) as virusid_s, count(*) as totalnum from $log where $filter group by filename,
analyticscksum, service, fsaverdict, dtype, user_src, virus, virusid_s /*SkipSTART*/order by
totalnum desc/*SkipEND*/)### t where $filter-drilldown and filename is not null and
dtype='fortisandbox' and fsaverdict not in ('clean', 'submission failed') group by filename,
analyticscksum, risk_level, risk, service  order by risk_level desc, total_num desc, service,
filename
```

Dataset Name	Description	Log Category
sdwan-CTAP-SB-Total-Number-of-Malicious-Suspicious-Files	CTAP SD-WAN Sandbox Malicious Suspicious Files Number	virus

```
select
  (
    case fsaverdict when & #039;malicious' then 'Malicious'  when 'high risk' then 'High' when
'medium risk' then 'Medium'  when 'low risk' then 'Low' else 'Other' end) as risk,  sum(totalnum)
as total_num  from ###(select filename, analyticscksum, service, fsaverdict, dtype, coalesce
(nullifna(`user`), ipstr(`srcip`)) as user_src, virus, virusid_to_str(virusid, eventtype) as
virusid_s, count(*) as totalnum from $log where $filter group by filename, analyticscksum,
service, fsaverdict, dtype, user_src, virus, virusid_s /*SkipSTART*/order by totalnum
desc/*SkipEND*/)### t where $filter-drilldown and dtype='fortisandbox' and fsaverdict not in
('clean','submission failed') group by risk order by total_num desc
```

Dataset Name	Description	Log Category
sdwan-CTAP-Top-Source-Countries	CTAP SD-WAN Top Source Countries	traffic

```
select
  srccountry,
  sum(bandwidth) as bandwidth
from
  ###(select $flex_timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf,
srcintfrole, dstintfrole, appid, appcat, app_group_name(app) as app_group, coalesce
(vwlservice,vwlname) as rulename, service, coalesce(nullifna(`srcname`),ipstr(`srcip`),nullifna
(`srcmac`)) as dev_src,  sum(crscore%65536) as crscore, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from $log-traffic where
$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf,
devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_
src, dev_src order by bandwidth desc)### t where $filter-drilldown and nullifna(srccountry) is not
null and srccountry <> 'Reserved' and bandwidth>0 group by srccountry order by bandwidth desc,
srccountry
```

Dataset Name	Description	Log Category
sdwan-CTAP-Average-Bandwidth-Day-Hour	CTAP SD-WAN Average Bandwidth by Day of Week and Hour	traffic

```
select
  hourstamp,
  daystamp,
  round(
    sum(bandwidth) / count(*)
  ) as bandwidth
from
  (
    select
      $hour_of_day(timestamp) as hourstamp,
      $HOUR_OF_DAY(timestamp) as hour_stamp,
      $day_of_week(timestamp) as daystamp,
      sum(bandwidth) as bandwidth
    from
      ###(select $flex_timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf,
srcintfrole, dstintfrole, appid, appcat, app_group_name(app) as app_group, coalesce
(vwlservice,vwlname) as rulename, service, coalesce(nullifna(`srcname`),ipstr(`srcip`),nullifna
(`srcmac`)) as dev_src,  sum(crscore%65536) as crscore, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from $log-traffic where
$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf,
devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_
```

```
src, dev_src order by bandwidth desc)### t where $filter-drilldown group by hourstamp, hour_stamp,
daystamp) t group by hourstamp, daystamp order by hourstamp
```

Dataset Name	Description	Log Category
sdwan-CTAP-Average-Log-Rate-By-Hour	CTAP SD-WAN Average Log Rate by Hour	event

```
select
  $hour_of_day(timestamp) as hourstamp,
  cast(
    (
      sum(
        total_trate + total_erate + total_orate
      )
    )/ sum(count)/ 100.0 as decimal(10, 2)
  ) as log_rate
from
  ###(select $flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_trate,
sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min(itime) as
first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) as
mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum(coalesce
(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max(coalesce(trate, 0)+coalesce(erate,
0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalsession, max
(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part(bandwidth, '/', 1), '0')
as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as recv,
max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_part
(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max
(coalesce(setuprate, 0)) as cps_peak, count(*) as count from $log where $filter and
subtype='system' and action='perf-stats' group by timestamp, devid, slot order by total_mem
desc)### t where $filter-drilldown group by hourstamp order by hourstamp
```

Dataset Name	Description	Log Category
sdwan-CTAP-CPU-Usage-Per-Hour	Event usage CPU	event

```
select
  $hour_of_day(timestamp) as hourstamp,
  cast(
    sum(total_cpu)/ sum(count) as decimal(6, 2)
  ) as cpu_avg_usage
from
  ###(select $flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_trate,
sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min(itime) as
first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) as
mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum(coalesce
(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max(coalesce(trate, 0)+coalesce(erate,
0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalsession, max
(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part(bandwidth, '/', 1), '0')
```

```
as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as recv,
max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_part
(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max
(coalesce(setuprate, 0)) as cps_peak, count(*) as count from $log where $filter and
subtype='system' and action='perf-stats' group by timestamp, devid, slot order by total_mem
desc)### t group by hourstamp order by hourstamp
```

Dataset Name	Description	Log Category
sdwan-CTAP-Memory-Usage-Per-Hour	Event usage memory	event

```
select
  $hour_of_day(timestamp) as hourstamp,
  cast(
    sum(total_mem)/ sum(count) as decimal(6, 2)
  ) as mem_avg_usage
from
  ###(select $flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_trate,
sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min(itime) as
first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) as
mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum(coalesce
(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max(coalesce(trate, 0)+coalesce(erate,
0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalsession, max
(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part(bandwidth, '/', 1), '0')
as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as recv,
max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_part
(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max
(coalesce(setuprate, 0)) as cps_peak, count(*) as count from $log where $filter and
subtype='system' and action='perf-stats' group by timestamp, devid, slot order by total_mem
desc)### t group by hourstamp order by hourstamp
```

Dataset Name	Description	Log Category
sdwan-Top-Destination-Addresses-By-Bandwidth-Bar	SD-WAN Top Destinations by Bandwidth Usage	traffic

```
select
  user_src as domain,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out
from
  ###(select $flex_timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf,
srcintfrole, dstintfrole, appid, appcat, app_group_name(app) as app_group, coalesce
(vwlservice,vwlname) as rulename, service, coalesce(nullifna(`srcname`),ipstr(`srcip`),nullifna
(`srcmac`)) as dev_src,  sum(crscore%65536) as crscore, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
```

```
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from $log-traffic where
$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf,
devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_
src, dev_src order by bandwidth desc)### t group by domain having sum(bandwidth)>0 order by
bandwidth desc
```

Dataset Name	Description	Log Category
intf-Timeline-Sampling	Interface Utilization Timeline by Data Sampling	event

```
with base_qry as (
  select
    tm,
    rcvdbps,
    ntile(100) over (
      order by
        rcvdbps
    ) as percentile
  from
    (
      select
        tm,
        sum(rcvdbps) as rcvdbps
      from

        /*fabricStart*/
        (
          select
            devname,
            devid,
            intfname,
            (timestamp / 300 * 300) as tm,
            sum(rcvdbyte) as rcvdbyte,
            sum(sentbyte) as sentbyte,
            sum(rcvdbyte)* 8 / sum(interval) as rcvdbps,
            sum(sentbyte)* 8 / sum(interval) as sentbps
          from
            $intf_billing tb
          where
            $dev_filter
            and $cust_time_filter(timestamp)
            and $filter - drilldown
          group by
            devname,
            devid,
            intfname,
            tm
        )
        /*fabricEnd*/
        tmp
      group by
```

```
        tm
    ) t
),
ref_qry as (
  select
    cast(
      max(rcvdbps)/ 1000000 as decimal(18, 2)
    ) as ref_val
  from
    base_qry
  where
    percentile = 95
)
select
  from_itime(timestamp) as tmstamp,
  cast(
    rcvdbps / 1000000 as decimal(18, 2)
  ) as rcvdbps,
  ref_val
from
  ref_qry,
  (
    select
      tm as timestamp,
      rcvdbps,
      rank() over(
        partition by (tm / 3600)
        order by
          tm
      ) as r
    from
      base_qry
  ) t
where
  r = 1
order by
  tmstamp
```

Dataset Name	Description	Log Category
intf-Util-Histogram	Interface Utilization Value Distribution	event

```
select
  cast(
    (
      (
        max(max_value) over ()
      )* seq / 100
    ) as decimal(16, 0)
  ) as value,
  cnt
```

```
from
  (
    select
      generate_series(0, 100, 2) as seq
  ) t1
  left join (
    select
      perc,
      max_value,
      count(*) as cnt
    from
      (
        select
          WIDTH_BUCKET(
            rcvdbps,
            0,
            (
              max(rcvdbps) over ()
            ) + 1,
            50
          )* 2 as perc,
          max(rcvdbps) over () as max_value
        from
          (
            select
              tm,
              sum(rcvdbps) as rcvdbps
            from

              /*fabricStart*/
              (
                select
                  devname,
                  devid,
                  intfname,
                  (timestamp / 300 * 300) as tm,
                  sum(rcvdbyte) as rcvdbyte,
                  sum(sentbyte) as sentbyte,
                  sum(rcvdbyte)* 8 / sum(interval) as rcvdbps,
                  sum(sentbyte)* 8 / sum(interval) as sentbps
                from
                  $intf_billing tb
                where
                  $dev_filter
                  and $cust_time_filter(timestamp)
                  and $filter - drilldown
                group by
                  devname,
                  devid,
                  intfname,
                  tm
              )
```

```
              /*fabricEnd*/
              tmp
            group by
              tm
          ) t
      ) t_bucket
    group by
      perc,
      max_value
  ) t2 on t1.seq = t2.perc
order by
  seq
```

Dataset Name	Description	Log Category
intf-Sorted-Line	Interface Utilization Line Sorted by bps	event

```
with base_qry as (
  select
    rcvdbps,
    ntile(100) over (
      order by
        rcvdbps
    ) as percentile
  from
    (
      select
        tm,
        sum(rcvdbps) as rcvdbps
      from

        /*fabricStart*/
        (
          select
            devname,
            devid,
            intfname,
            (timestamp / 300 * 300) as tm,
            sum(rcvdbyte) as rcvdbyte,
            sum(sentbyte) as sentbyte,
            sum(rcvdbyte)* 8 / sum(interval) as rcvdbps,
            sum(sentbyte)* 8 / sum(interval) as sentbps
          from
            $intf_billing tb
          where
            $dev_filter
            and $cust_time_filter(timestamp)
            and $filter - drilldown
          group by
            devname,
            devid,
```

```
            intfname,
            tm
        )
        /*fabricEnd*/
        tmp
      group by
        tm
    ) t
),
ref_qry as (
  select
    cast(
      max(rcvdbps)/ 1000000 as decimal(18, 2)
    ) as ref_val
  from
    base_qry
  where
    percentile = 95
)
select
  n_perc,
  cast(
    rcvdbps / 1000000 as decimal(18, 2)
  ) as rcvdbps,
  ref_val
from
  (
    select
      seq as n_perc,
      rcvdbps
    from
      (
        select
          generate_series(0, 100, 1) as seq
      ) t1
      left join (
        select
          max(rcvdbps) as rcvdbps,
          percentile
        from
          base_qry
        group by
          percentile
      ) t2 on t1.seq = t2.percentile
  ) t,
  ref_qry
order by
  n_perc
```

Dataset Name	Description	Log Category
intf-Data-Analysis-Table	Interface Utilization Data Analysis	event

```
with base_qry as (
  select
    rcvdbps,
    rcvdbyte,
    ntile(100) over (
      order by
        rcvdbps
    ) as percentile
  from
    (
      select
        tm,
        sum(rcvdbps) as rcvdbps,
        sum(rcvdbyte) as rcvdbyte
      from

        /*fabricStart*/
        (
          select
            devname,
            devid,
            intfname,
            (timestamp / 300 * 300) as tm,
            sum(rcvdbyte) as rcvdbyte,
            sum(sentbyte) as sentbyte,
            sum(rcvdbyte)* 8 / sum(interval) as rcvdbps,
            sum(sentbyte)* 8 / sum(interval) as sentbps
          from
            $intf_billing tb
          where
            $dev_filter
            and $cust_time_filter(timestamp)
            and $filter - drilldown
          group by
            devname,
            devid,
            intfname,
            tm
        )
        /*fabricEnd*/
        tmp
      group by
        tm
    ) t
)
select
  min_mbps,
  low_ref_mbps,
  mean_mbps,
  ref_mbps,
  peak_mbps,
  actual_gb,
```

```
  total
from
  (
    select
      cast(
        min(rcvdbps)/ 1000000 as decimal(18, 2)
      ) as min_mbps,
      cast(
        avg(rcvdbps)/ 1000000 as decimal(18, 2)
      ) as mean_mbps,
      cast(
        max(rcvdbps)/ 1000000 as decimal(18, 2)
      ) as peak_mbps,
      cast(
        (
          select
            max(rcvdbps)
          from
            base_qry
          where
            percentile = 5
        )/ 1000000 as decimal(18, 2)
      ) as low_ref_mbps,
      cast(
        (
          select
            max(rcvdbps)
          from
            base_qry
          where
            percentile = 95
        )/ 1000000 as decimal(18, 2)
      ) as ref_mbps,
      cast(
        sum(rcvdbyte)/(1024 * 1024 * 1024) as decimal(18, 2)
      ) as actual_gb,
      count(*) as total
    from
      base_qry
  ) t
```

Dataset Name	Description	Log Category
intf-Device-Summary	Interface Utilization Device Summary	event

```
select
  devname,
  intfname,
  cast(
    sum(rcvdbyte)/(1024 * 1024 * 1024) as decimal(18, 2)
  ) as rcvd_gb
```

```
from

  /*fabricStart*/
  (
    select
      devname,
      devid,
      intfname,
      (timestamp / 300 * 300) as tm,
      sum(rcvdbyte) as rcvdbyte,
      sum(sentbyte) as sentbyte,
      sum(rcvdbyte)* 8 / sum(interval) as rcvdbps,
      sum(sentbyte)* 8 / sum(interval) as sentbps
    from
      $intf_billing tb
    where
      $dev_filter
      and $cust_time_filter(timestamp)
      and $filter - drilldown
    group by
      devname,
      devid,
      intfname,
      tm
  )
  /*fabricEnd*/
  t
group by
  devname,
  intfname
order by
  devname,
  rcvd_gb desc,
  intfname
```

Dataset Name	Description	Log Category
intf-Device-Sent-Summary	Interface Utilization Device Summary	event

```
select
  devname,
  intfname,
  cast(
    sum(sentbyte)/(1024 * 1024 * 1024) as decimal(18, 2)
  ) as sent_gb
from

  /*fabricStart*/
  (
    select
      devname,
```

```
      devid,
      intfname,
      (timestamp / 300 * 300) as tm,
      sum(rcvdbyte) as rcvdbyte,
      sum(sentbyte) as sentbyte,
      sum(rcvdbyte)* 8 / sum(interval) as rcvdbps,
      sum(sentbyte)* 8 / sum(interval) as sentbps
    from
      $intf_billing tb
    where
      $dev_filter
      and $cust_time_filter(timestamp)
      and $filter - drilldown
    group by
      devname,
      devid,
      intfname,
      tm
  )
  /*fabricEnd*/
  t
group by
  devname,
  intfname
order by
  devname,
  sent_gb desc,
  intfname
```

Dataset Name	Description	Log Category
intf-Device-Rcvd-Sent-Summary	Interface Utilization Device Received and Sent Data Summary	event

```
select
  devname,
  intfname,
  rcvd_gb,
  sent_gb,
  (rcvd_gb + sent_gb) as total_gb
from
  (
    select
      devname,
      intfname,
      cast(
        sum(rcvdbyte)/(1024 * 1024 * 1024) as decimal(18, 2)
      ) as rcvd_gb,
      cast(
        sum(sentbyte)/(1024 * 1024 * 1024) as decimal(18, 2)
      ) as sent_gb
```

```
    from

      /*fabricStart*/
      (
        select
          devname,
          devid,
          intfname,
          (timestamp / 300 * 300) as tm,
          sum(rcvdbyte) as rcvdbyte,
          sum(sentbyte) as sentbyte,
          sum(rcvdbyte)* 8 / sum(interval) as rcvdbps,
          sum(sentbyte)* 8 / sum(interval) as sentbps
        from
          $intf_billing tb
        where
          $dev_filter
          and $cust_time_filter(timestamp)
          and $filter - drilldown
        group by
          devname,
          devid,
          intfname,
          tm
      )
      /*fabricEnd*/
      t
    group by
      devname,
      intfname
  ) t
order by
  total_gb desc
```

Dataset Name	Description	Log Category
intf-Util-Device-List	Interface Utilization Device List	event

```
select
  devname,
  devid,
  (rcvd_gb + sent_gb) as total_gb
from
  (
    select
      devname,
      devid,
      cast(
        sum(rcvdbyte)/(1024 * 1024 * 1024) as decimal(18, 2)
      ) as rcvd_gb,
      cast(
```

```
        sum(sentbyte)/(1024 * 1024 * 1024) as decimal(18, 2)
      ) as sent_gb
    from

      /*fabricStart*/
      (
        select
          devname,
          devid,
          intfname,
          (timestamp / 300 * 300) as tm,
          sum(rcvdbyte) as rcvdbyte,
          sum(sentbyte) as sentbyte,
          sum(rcvdbyte)* 8 / sum(interval) as rcvdbps,
          sum(sentbyte)* 8 / sum(interval) as sentbps
        from
          $intf_billing tb
        where
          $dev_filter
          and $cust_time_filter(timestamp)
          and $filter - drilldown
        group by
          devname,
          devid,
          intfname,
          tm
      )
      /*fabricEnd*/
      t
    group by
      devname,
      devid
  ) t
where
  (rcvd_gb + sent_gb)> 0
order by
  total_gb desc
```

Dataset Name	Description	Log Category
intf-Util-Interface-List	Interface Utilization Interface List	event

```
select
  devname,
  devid,
  intfname,
  (rcvd_gb + sent_gb) as total_gb
from
  (
    select
      devname,
```

```
      devid,
      intfname,
      cast(
        sum(rcvdbyte)/(1024 * 1024 * 1024) as decimal(18, 2)
      ) as rcvd_gb,
      cast(
        sum(sentbyte)/(1024 * 1024 * 1024) as decimal(18, 2)
      ) as sent_gb
    from

      /*fabricStart*/
      (
        select
          devname,
          devid,
          intfname,
          (timestamp / 300 * 300) as tm,
          sum(rcvdbyte) as rcvdbyte,
          sum(sentbyte) as sentbyte,
          sum(rcvdbyte)* 8 / sum(interval) as rcvdbps,
          sum(sentbyte)* 8 / sum(interval) as sentbps
        from
          $intf_billing tb
        where
          $dev_filter
          and $cust_time_filter(timestamp)
          and $filter - drilldown
        group by
          devname,
          devid,
          intfname,
          tm
      )
      /*fabricEnd*/
      t
    group by
      devname,
      devid,
      intfname
  ) t
where
  (rcvd_gb + sent_gb)> 0
order by
  total_gb desc
```

Dataset Name	Description	Log Category
intf-Timeline-Rcvd-Sampling-drilldown	Interface Utilization Timeline by Received Data Sampling Drilldown	event

```
with base_qry as (
  select
    devid,
    intfname,
    tm,
    rcvdbps,
    ntile(100) over (
      partition by (devid, intfname)
      order by
        rcvdbps
    ) as percentile
  from

    /*fabricStart*/
    (
      select
        devname,
        devid,
        intfname,
        (timestamp / 300 * 300) as tm,
        sum(rcvdbyte) as rcvdbyte,
        sum(sentbyte) as sentbyte,
        sum(rcvdbyte)* 8 / sum(interval) as rcvdbps,
        sum(sentbyte)* 8 / sum(interval) as sentbps
      from
        $intf_billing tb
      where
        $dev_filter
        and $cust_time_filter(timestamp)
        and $filter - drilldown
      group by
        devname,
        devid,
        intfname,
        tm
    )
    /*fabricEnd*/
    tmp
),
ref_qry as (
  select
    devid,
    intfname,
    cast(
      max(rcvdbps)/ 1000000 as decimal(18, 2)
    ) as ref_val
  from
    base_qry
  where
    percentile = 95
  group by
    devid,
```

```
    intfname
)
select
  t.devid as devid,
  t.intfname as intfname,
  timestamp / 3600 * 3600 as tmstamp,
  cast(
    rcvdbps / 1000000 as decimal(18, 2)
  ) as rcvdbps,
  ref_val
from
  (
    select
      devid,
      intfname,
      tm as timestamp,
      rcvdbps,
      rank() over (
        partition by (tm / 3600, devid, intfname)
        order by
          tm
      ) as r
    from
      base_qry
  ) t
  left join ref_qry t1 on t.devid = t1.devid
  and t.intfname = t1.intfname
where
  r = 1
order by
  tmstamp
```

Dataset Name	Description	Log Category
intf-Rcvd-Data-Analysis-Table-drilldown	Interface Utilization Received Data Analysis Drilldown	event

```
with base_qry as (
  select
    devid,
    intfname,
    rcvdbps,
    rcvdbyte,
    ntile(100) over (
      partition by (devid, intfname)
      order by
        rcvdbps
    ) as percentile
  from

    /*fabricStart*/
```

```
    (
      select
        devname,
        devid,
        intfname,
        (timestamp / 300 * 300) as tm,
        sum(rcvdbyte) as rcvdbyte,
        sum(sentbyte) as sentbyte,
        sum(rcvdbyte)* 8 / sum(interval) as rcvdbps,
        sum(sentbyte)* 8 / sum(interval) as sentbps
      from
        $intf_billing tb
      where
        $dev_filter
        and $cust_time_filter(timestamp)
        and $filter - drilldown
      group by
        devname,
        devid,
        intfname,
        tm
    )
    /*fabricEnd*/
    tmp
)
select
  devid,
  intfname,
  min_mbps,
  low_ref_mbps,
  mean_mbps,
  ref_mbps,
  peak_mbps,
  actual_gb,
  total
from
  (
    select
      devid,
      intfname,
      cast(
        min(rcvdbps)/ 1000000 as decimal(18, 2)
      ) as min_mbps,
      cast(
        avg(rcvdbps)/ 1000000 as decimal(18, 2)
      ) as mean_mbps,
      cast(
        max(rcvdbps)/ 1000000 as decimal(18, 2)
      ) as peak_mbps,
      cast(
        (
          select
```

```
            max(rcvdbps)
          from
            base_qry
          where
            percentile = 5
        )/ 1000000 as decimal(18, 2)
      ) as low_ref_mbps,
      cast(
        (
          select
            max(rcvdbps)
          from
            base_qry
          where
            percentile = 95
        )/ 1000000 as decimal(18, 2)
      ) as ref_mbps,
      cast(
        sum(rcvdbyte)/(1024 * 1024 * 1024) as decimal(18, 2)
      ) as actual_gb,
      count(*) as total
    from
      base_qry
    group by
      devid,
      intfname
  ) t
```

Dataset Name	Description	Log Category
intf-Util-Rcvd-Histogram-drilldown	Interface Utilization Received Value Distribution Drilldown	event

```
select
  devid,
  intfname,
  cast(
    (
      (
        max(max_value) over ()
      )* seq / 100
    ) as decimal(16, 0)
  ) as value,
  cnt
from
  (
    select
      generate_series(0, 100, 2) as seq
  ) t1
  left join (
    select
```

```
  devid,
  intfname,
  perc,
  max_value,
  count(*) as cnt
from
  (
    select
      devid,
      intfname,
      WIDTH_BUCKET(
        rcvdbps,
        0,
        (
          max(rcvdbps) over (
            partition by (devid, intfname)
          )
        ) + 1,
        50
      )* 2 as perc,
      max(rcvdbps) over (
        partition by (devid, intfname)
      ) as max_value
    from

      /*fabricStart*/
      (
        select
          devname,
          devid,
          intfname,
          (timestamp / 300 * 300) as tm,
          sum(rcvdbyte) as rcvdbyte,
          sum(sentbyte) as sentbyte,
          sum(rcvdbyte)* 8 / sum(interval) as rcvdbps,
          sum(sentbyte)* 8 / sum(interval) as sentbps
        from
          $intf_billing tb
        where
          $dev_filter
          and $cust_time_filter(timestamp)
          and $filter - drilldown
        group by
          devname,
          devid,
          intfname,
          tm
      )
      /*fabricEnd*/
      tmp
  ) t_bucket
group by
```

```
      devid,
      intfname,
      perc,
      max_value
  ) t2 on t1.seq = t2.perc
order by
  seq
```

Dataset Name	Description	Log Category
intf-Rcvd-Sorted-Line-drilldown	Interface Utilization Line Sorted by Received bps Drilldown	event

```
with base_qry as (
  select
    devid,
    intfname,
    rcvdbps,
    ntile(100) over (
      partition by (devid, intfname)
      order by
        rcvdbps
    ) as percentile
  from

    /*fabricStart*/
    (
      select
        devname,
        devid,
        intfname,
        (timestamp / 300 * 300) as tm,
        sum(rcvdbyte) as rcvdbyte,
        sum(sentbyte) as sentbyte,
        sum(rcvdbyte)* 8 / sum(interval) as rcvdbps,
        sum(sentbyte)* 8 / sum(interval) as sentbps
      from
        $intf_billing tb
      where
        $dev_filter
        and $cust_time_filter(timestamp)
        and $filter - drilldown
      group by
        devname,
        devid,
        intfname,
        tm
    )
    /*fabricEnd*/
    tmp
),
```

```
ref_qry as (
  select
    devid,
    intfname,
    cast(
      max(rcvdbps)/ 1000000 as decimal(18, 2)
    ) as ref_val
  from
    base_qry
  where
    percentile = 95
  group by
    devid,
    intfname
)
select
  t.devid,
  t.intfname,
  n_perc,
  cast(
    rcvdbps / 1000000 as decimal(18, 2)
  ) as rcvdbps,
  ref_val
from
  (
    select
      devid,
      intfname,
      seq as n_perc,
      rcvdbps
    from
      (
        select
          generate_series(0, 100, 1) as seq
      ) t1
      left join (
        select
          devid,
          intfname,
          max(rcvdbps) as rcvdbps,
          percentile
        from
          base_qry
        group by
          devid,
          intfname,
          percentile
      ) t2 on t1.seq = t2.percentile
  ) t
  left join ref_qry t1 on t.devid = t1.devid
  and t.intfname = t1.intfname
where
```

```
  n_perc>0
order by
  n_perc
```

Dataset Name	Description	Log Category
intf-Timeline-Sent-Sampling-drilldown	Interface Utilization Timeline by Data Sampling Drilldown	event

```
with base_qry as (
  select
    devid,
    intfname,
    tm,
    sentbps,
    ntile(100) over (
      partition by (devid, intfname)
      order by
        sentbps
    ) as percentile
  from

    /*fabricStart*/
    (
      select
        devname,
        devid,
        intfname,
        (timestamp / 300 * 300) as tm,
        sum(rcvdbyte) as rcvdbyte,
        sum(sentbyte) as sentbyte,
        sum(rcvdbyte)* 8 / sum(interval) as rcvdbps,
        sum(sentbyte)* 8 / sum(interval) as sentbps
      from
        $intf_billing tb
      where
        $dev_filter
        and $cust_time_filter(timestamp)
        and $filter - drilldown
      group by
        devname,
        devid,
        intfname,
        tm
    )
    /*fabricEnd*/
    tmp
),
ref_qry as (
  select
    devid,
```

```
    intfname,
    cast(
      max(sentbps)/ 1000000 as decimal(18, 2)
    ) as ref_val
  from
    base_qry
  where
    percentile = 95
  group by
    devid,
    intfname
)
select
  t.devid as devid,
  t.intfname as intfname,
  timestamp / 3600 * 3600 as tmstamp,
  cast(
    sentbps / 1000000 as decimal(18, 2)
  ) as sentbps,
  ref_val
from
  (
    select
      devid,
      intfname,
      tm as timestamp,
      sentbps,
      rank() over (
        partition by (tm / 3600, devid, intfname)
        order by
          tm
      ) as r
    from
      base_qry
  ) t
  left join ref_qry t1 on t.devid = t1.devid
  and t.intfname = t1.intfname
where
  r = 1
order by
  tmstamp
```

Dataset Name	Description	Log Category
intf-Sent-Data-Analysis-Table-drilldown	Interface Utilization Data Analysis Drilldown	event

```
with base_qry as (
  select
    devid,
    intfname,
```

```
    sentbps,
    sentbyte,
    ntile(100) over (
      partition by (devid, intfname)
      order by
        sentbps
    ) as percentile
  from

    /*fabricStart*/
    (
      select
        devname,
        devid,
        intfname,
        (timestamp / 300 * 300) as tm,
        sum(rcvdbyte) as rcvdbyte,
        sum(sentbyte) as sentbyte,
        sum(rcvdbyte)* 8 / sum(interval) as rcvdbps,
        sum(sentbyte)* 8 / sum(interval) as sentbps
      from
        $intf_billing tb
      where
        $dev_filter
        and $cust_time_filter(timestamp)
        and $filter - drilldown
      group by
        devname,
        devid,
        intfname,
        tm
    )
    /*fabricEnd*/
    tmp
)
select
  devid,
  intfname,
  min_mbps,
  low_ref_mbps,
  mean_mbps,
  ref_mbps,
  peak_mbps,
  actual_gb,
  total
from
  (
    select
      devid,
      intfname,
      cast(
        min(sentbps)/ 1000000 as decimal(18, 2)
```

```
      ) as min_mbps,
      cast(
        avg(sentbps)/ 1000000 as decimal(18, 2)
      ) as mean_mbps,
      cast(
        max(sentbps)/ 1000000 as decimal(18, 2)
      ) as peak_mbps,
      cast(
        (
          select
            max(sentbps)
          from
            base_qry
          where
            percentile = 5
        )/ 1000000 as decimal(18, 2)
      ) as low_ref_mbps,
      cast(
        (
          select
            max(sentbps)
          from
            base_qry
          where
            percentile = 95
        )/ 1000000 as decimal(18, 2)
      ) as ref_mbps,
      cast(
        sum(sentbyte)/(1024 * 1024 * 1024) as decimal(18, 2)
      ) as actual_gb,
      count(*) as total
    from
      base_qry
    group by
      devid,
      intfname
  ) t
```

Dataset Name	Description	Log Category
intf-Util-Sent-Histogram-drilldown	Interface Utilization Value Distribution Drilldown	event

```
select
  devid,
  intfname,
  cast(
    (
      (
        max(max_value) over ()
      )* seq / 100
    ) as decimal(16, 2)
```

```
  ) as value,
  cnt
from
  (
    select
      generate_series(0, 100, 2) as seq
  ) t1
  left join (
    select
      devid,
      intfname,
      perc,
      max_value,
      count(*) as cnt
    from
      (
        select
          devid,
          intfname,
          WIDTH_BUCKET(
            sentbps,
            0,
            (
              max(sentbps) over (
                partition by (devid, intfname)
              )
            ) + 1,
            50
          )* 2 as perc,
          max(sentbps) over (
            partition by (devid, intfname)
          ) as max_value
        from

          /*fabricStart*/
          (
            select
              devname,
              devid,
              intfname,
              (timestamp / 300 * 300) as tm,
              sum(rcvdbyte) as rcvdbyte,
              sum(sentbyte) as sentbyte,
              sum(rcvdbyte)* 8 / sum(interval) as rcvdbps,
              sum(sentbyte)* 8 / sum(interval) as sentbps
            from
              $intf_billing tb
            where
              $dev_filter
              and $cust_time_filter(timestamp)
              and $filter - drilldown
            group by
```

```
              devname,
              devid,
              intfname,
              tm
          )
          /*fabricEnd*/
          tmp
      ) t_bucket
    group by
      devid,
      intfname,
      perc,
      max_value
  ) t2 on t1.seq = t2.perc
order by
  seq
```

Dataset Name	Description	Log Category
intf-Sent-Sorted-Line-drilldown	Interface Utilization Line Sorted by bps Drilldown	event

```
with base_qry as (
  select
    devid,
    intfname,
    sentbps,
    ntile(100) over (
      partition by (devid, intfname)
      order by
        sentbps
    ) as percentile
  from

    /*fabricStart*/
    (
      select
        devname,
        devid,
        intfname,
        (timestamp / 300 * 300) as tm,
        sum(rcvdbyte) as rcvdbyte,
        sum(sentbyte) as sentbyte,
        sum(rcvdbyte)* 8 / sum(interval) as rcvdbps,
        sum(sentbyte)* 8 / sum(interval) as sentbps
      from
        $intf_billing tb
      where
        $dev_filter
        and $cust_time_filter(timestamp)
        and $filter - drilldown
      group by
```

```
        devname,
        devid,
        intfname,
        tm
    )
    /*fabricEnd*/
    tmp
),
ref_qry as (
  select
    devid,
    intfname,
    cast(
      max(sentbps)/ 1000000 as decimal(18, 2)
    ) as ref_val
  from
    base_qry
  where
    percentile = 95
  group by
    devid,
    intfname
)
select
  t.devid,
  t.intfname,
  n_perc,
  cast(
    sentbps / 1000000 as decimal(18, 2)
  ) as sentbps,
  ref_val
from
  (
    select
      devid,
      intfname,
      seq as n_perc,
      sentbps
    from
      (
        select
          generate_series(0, 100, 1) as seq
      ) t1
      left join (
        select
          devid,
          intfname,
          max(sentbps) as sentbps,
          percentile
        from
          base_qry
        group by
```

```
          devid,
          intfname,
          percentile
      ) t2 on t1.seq = t2.percentile
  ) t
  left join ref_qry t1 on t.devid = t1.devid
  and t.intfname = t1.intfname
where
  n_perc>0
order by
  n_perc
```

Dataset Name	Description	Log Category
daily-Summary-Traffic-Bandwidth-Line	Daily Summary - Traffic Bandwidth Line	traffic

```
select
  $fv_line_timescale(timescale) as time,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out,
  sum(session_block) as session_block,
  (
    sum(sessions)- sum(session_block)
  ) as session_pass
from
  (
    (
      select
        timescale,
        sum(traffic_in) as traffic_in,
        sum(traffic_out) as traffic_out,
        sum(session_block) as session_block,
        sum(sessions) as sessions
      from
        t
      group by
        timescale
    )
    union all
      (
        select
          timescale,
          sum(traffic_in) as traffic_in,
          sum(traffic_out) as traffic_out,
          sum(session_block) as session_block,
          sum(sessions) as sessions
        from
          t
        group by
          timescale
```

```
      )
  ) t
group by
  time
order by
  time
```

Dataset Name	Description	Log Category
daily-Summary-Top-User	Daily Summary - Top User by Bandwidth	traffic

```
select
  coalesce(
    nullifna(f_user),
    ipstr(srcip),
    & #039;Unknown') as f_user, srcip, sum(bandwidth) as bandwidth FROM  t group by f_user, srcip
order by bandwidth desc
```

Dataset Name	Description	Log Category
daily-Summary-Top-Domain	Daily Summary - Top Domain by Bandwidth	traffic

```
select
  domain,
  sum(bandwidth) as bandwidth
from
  t
where
  domain is not null
group by
  domain
order by
  bandwidth desc
```

Dataset Name	Description	Log Category
daily-Summary-Top-Appcat-Bandwidth	Daily Summary - Top Application Category by Bandwidth	traffic

```
select
  appcat,
  sum(bandwidth) as bandwidth
from
  (
    select
      t1.*,
      t2.app_cat as appcat
    from
```

```
      t1
      left join app_mdata t2 on t1.app_group = t2.name
  ) t
where
  $filter - drilldown
  and appcat is not null
group by
  appcat
order by
  bandwidth desc
```

Dataset Name	Description	Log Category
daily-Summary-Top-App	Daily Summary - Top Application	traffic

```
select
  app_group,
  max(appcat) as appcat,
  (
    case max(d_risk) when 1 then & #039;Low' when 2 then 'Elevated' when 3 then 'Medium' when 4
then 'High' when 5 then 'Critical' else NULL end) as risk, sum(bandwidth) as bandwidth, sum
(traffic_in) as traffic_in, sum(traffic_out) as traffic_out, sum(session_block) as session_block,
(sum(sessions)-sum(session_block)) as session_pass, sum(sessions) as sessions from (select t1.*,
(case when (d_flags & 1) = 1 then 'Not.Scanned' when t2.app_cat is null then 'Unknown' else
t2.app_cat end) as appcat, (case when t2.risk is null then 0 else t2.risk::int end) as d_risk from
 t1 left join app_mdata t2 on t1.app_group=t2.name) t where $filter-drilldown group by app_group
order by max(d_risk) desc, sessions desc, bandwidth desc
```

Dataset Name	Description	Log Category
daily-Summary-Top-Threats	Daily Summary - Top Threats	traffic

```
select
  threat_s as threat,
  threattype_s as threattype,
  sum(threatweight) as threatweight,
  sum(threat_block) as threat_block,
  (
    sum(threatweight)- sum(threat_block)
  ) as threat_pass,
  sum(incidents) as incidents,
  sum(incident_block) as incident_block,
  (
    sum(incidents)- sum(incident_block)
  ) as incident_pass
from
  (
    (
      select
```

```
        threat_s,
        threattype_s,
        sum(threatweight) as threatweight,
        sum(threat_block) as threat_block,
        sum(incidents) as incidents,
        sum(incident_block) as incident_block
      from
        t
      group by
        threat_s,
        threattype_s
    )
    union all
      (
        select
          threat_s,
          threattype_s,
          sum(threatweight) as threatweight,
          sum(threat_block) as threat_block,
          sum(incidents) as incidents,
          sum(incident_block) as incident_block
        from
          t
        group by
          threat_s,
          threattype_s
      )
  ) t
group by
  threat,
  threattype
order by
  threatweight desc
```

Dataset Name	Description	Log Category
daily-Summary-Top-Compromised-Hosts	Daily Summary - Top Compromised Hosts	traffic

```
select
  epid,
  devid,
  vd,
  (
    case when email <>& #039;' then inet '0.0.0.0' else srcip end) as srcip, ip_reversed, devtype,
fctuid, euid, bmp_logtype as logtype, unauthuser, srcmac, osname, osversion, f_user,(case when
epid>1024 then epname when email<>'' then '' else ipstr(srcip) end) as epname,threat_num, bl_
count, cs_score, cs_count, verdict, rescan, (case verdict when 1 then 'Low Suspicion' when 2 then
'Medium Suspicion' when 3 then 'High Suspicion' when 4 then 'Infected' else 'N/A' end) as verdict_
s,ack_time, ack_note, last_bl as last_detected_time from (/*NOLOG_SUBQRY_BEG*/SELECT epid, email,
itime, bl_count, cs_score, cs_count, threat_num, bmp_logtype, last_bl, verdict, rescan, srcip, ip_
```

```
reversed, epname, srcmac, osname, osversion, devtype, fctuid, euid, unauthuser, f_user, ack_note,
ack_time, devid, vd, csf, devname FROM (SELECT tvdt.epid, tvdt.email, itime, tvdt.bl_count,
tvdt.cs_score, tvdt.cs_count, tvdt.threat_num, tvdt.bmp_logtype, tvdt.last_bl, tvdt.verdict,
tvdt.ip_reversed, tvdt.rescan, (CASE WHEN tvdt.epid>1024 THEN tep.epip ELSE tvdt.srcip END) as
srcip, tep.epname, tep.mac as srcmac, tep.osname, tep.osversion,tep.epdevtype as devtype,
teu.fctuid, teu.euid, teu.unauthuser, (case when teu.euid>1024 then teu.euname when email<>'' then
email when ipstr(tvdt.srcip)<>'0.0.0.0' then ipstr(tvdt.srcip) else NULL end) as f_user, tack.ack_
note, (case when (tvdt.ack_time_max=0 or tvdt.ack_time_min=0) then NULL else tvdt.ack_time_max
end) as ack_time,tdev.devid, tdev.vd, tdev.csf, tdev.devname, tdev.devgrps FROM (SELECT epid,
srcip, email, min(day_st) as itime, array_length(intarr_agg(threatid), 1) as threat_num, intarr_
agg(dvid) as dvid, sum(bl_count) as bl_count, max(cs_score) as cs_score, sum(cs_count) as cs_
count, max(last_bl) as last_bl, max(ack_time) as ack_time_max, min(ack_time) as ack_time_min, bit_
or(bmp_logtype) as bmp_logtype, max(verdict) as verdict, max(ip_reversed) as ip_reversed, max
(rescan) as rescan FROM (SELECT epid, (coalesce(srcip, '0.0.0.0'::inet)) as srcip, (coalesce(ioc_
email, ''::text)) as email, day_st, ack_time, threatid, dvid,bl_count, cs_score, cs_count, last_
bl, bmp_logtype, verdict, (case when ioc_flags&2>0 then 1 else 0 end) as ip_reversed, (case when
ioc_flags&1>0 then 1 else 0 end) as rescan FROM $ADOMTBL_PLHD_IOC_VERDICT /*verdict table*/WHERE
day_st>=$start_time and day_st<=$end_time /*time filter*/) tvdt_int GROUP BY epid, srcip, email)
tvdt INNER JOIN /*end points*/  $ADOM_ENDPOINT as tep ON tvdt.epid=tep.epid LEFT JOIN /*end user*/
(select epid, euname, fctuid, euid, unauthuser from (select epid, eu.euid, euname, fctuid, euname
as unauthuser, row_number() over (partition by epid order by ((case when fctuid is null then 0
else 1 end), lastactive) desc) nth from $ADOM_ENDUSER eu /*end user*/, $ADOM_EPEU_DEVMAP as map
/*epeu dev_map*/ where eu.euid=map.euid and eu.euid>1024) eum where nth=1) teu on
tvdt.epid=teu.epid LEFT JOIN /*ack table*/(SELECT epid, srcip, ack_time, ack_note FROM (SELECT
epid, srcip, ack_time, ack_note, row_number() over (PARTITION BY epid, srcip order by ack_time
desc) as ackrank FROM ioc_ack WHERE adomoid=$adom_oid) rankqry WHERE ackrank=1) tack ON
tvdt.epid=tack.epid and (tack.srcip is null or tvdt.srcip=tack.srcip) LEFT JOIN /*devtable */
devtable_ext tdev ON tdev.dvid = tvdt.dvid[1] WHERE tvdt.dvid && (SELECT array_agg(dvid) from
/*devtable */ devtable_ext WHERE $filter-drilldown)) tioc /*NOLOG_SUBQRY_END*/ ) t order by
threat_num desc
```

Dataset Name	Description	Log Category
daily-Summary-Incidents-by-Severity	Incidents by Severity	

```
select
  severity,
  sum(incnum) as incnum
from

  /*fabricStart*/
  (
    select
      severity,
      count(*) as incnum
    from
      $incident
    where
      $cust_time_filter(createtime)
    group by
```

```
      severity
    order by
      incnum desc
  )
  /*fabricEnd*/
  t
group by
  severity
order by
  incnum desc
```

Dataset Name	Description	Log Category
ueba-Asset-Count-by-Detecttype	Asset Count by Detection Type	

```
select
  (
    case detecttype when & #039;by_ip' then 'IP' when 'by_mac' then 'MAC' end) as detecttype,
count(distinct epid) as count from $ADOM_ENDPOINT t1 where epid>1024 and $filter-drilldown and
lastseen>=$start_time and firstseen<$end_time and detecttype in ('by_ip', 'by_mac') group by
detecttype order by count desc
```

Dataset Name	Description	Log Category
ueba-Asset-Identification	Asset Count by Identification	

```
with qualified_ep as (
  select
    t2.epid,
    t2.euid
  from
    $ADOM_ENDPOINT t1
    inner join $ADOM_EPEU_DEVMAP t2 on t1.epid = t2.epid
  where
    $filter - drilldown
    and lastseen >= $start_time
    and firstseen<$end_time
    and t2.epid>1024
),
identified_ep as (
  select
    distinct epid
  from
    qualified_ep t1
    inner join $ADOM_ENDUSER t2 on t1.euid = t2.euid
  where
    t1.euid is not null
    and t1.euid>1024
    and euname !=& #039;(none)' and euname is not null) (select 'Identified' as type, count
```

```
(distinct epid) as count from identified_ep) union all (select 'Unidentified' as type, count
(distinct epid) as count from qualified_ep where epid not in (select * from identified_ep))
```

Dataset Name	Description	Log Category
ueba-Asset-Count-by-HWOS	Asset Count by Hardware OS	

```
select
  osname,
  count(distinct t2.epid) as count
from
  $ADOM_ENDPOINT t1
  inner join $ADOM_EPEU_DEVMAP t2 on t1.epid = t2.epid
where
  $filter - drilldown
  and lastseen >= $start_time
  and firstseen<$end_time
  and osname is not null
  and t2.epid>1024
group by
  osname
order by
  count desc
```

Dataset Name	Description	Log Category
ueba-Asset-Count-by-Device-and-Detecttype	Asset Count by Source and Detection Type	

```
select
  devname,
  (
    case detecttype when & #039;by_ip' then 'IP' when 'by_mac' then 'MAC' end) as detecttype,
count(distinct t1.epid) as count from $ADOM_ENDPOINT t1 inner join $ADOM_EPEU_DEVMAP t2 on
t1.epid=t2.epid inner join devtable_ext t3 on t2.devid=t3.devid where t1.epid>1024 and $filter-
drilldown and t1.lastseen>=$start_time and firstseen<$end_time and devname is not null and
detecttype in ('by_ip', 'by_mac') group by devname, detecttype order by count desc
```

Dataset Name	Description	Log Category
ueba-User-Count-by-Usergroup	User Count by User Group	

```
select
  coalesce(
    eugroup,
    & #039;Unknown') as eugroup, count(distinct t1.euid) as count from $ADOM_ENDUSER t1 inner join
$ADOM_EPEU_DEVMAP t2 ON t1.euid=t2.euid where $filter-drilldown and t1.euid>1024 and
t1.lastseen>=$start_time and firstseen<$end_time group by eugroup order by count desc
```

Dataset Name	Description	Log Category
ueba-Asset-User-Count-by-Device	Asset and User Count by Device	

```
select
  devname,
  cnt_for,
  sum(count) as count
from
  (
    (
      select
        devname,
        & #039;Endpoint' as cnt_for, count(distinct t2.epid) as count from $ADOM_ENDPOINT t1 inner
join $ADOM_EPEU_DEVMAP t2 on t1.epid=t2.epid inner join devtable_ext t3 on t2.devid=t3.devid where
$filter-drilldown and t1.lastseen>=$start_time and t1.firstseen<$end_time and t2.epid>1024 group
by devname order by count desc) union all (select devname, 'User' as cnt_for, count(distinct
t1.euid) as count from $ADOM_ENDUSER t1 inner join $ADOM_EPEU_DEVMAP t2 ON t1.euid=t2.euid inner
join devtable_ext t3 on t2.devid=t3.devid where $filter-drilldown and t1.lastseen>=$start_time and
t1.firstseen<$end_time and euname != '(none)' and epid>1024 and t1.euid>1024  group by devname
order by count desc)) t group by devname, cnt_for order by count desc
```

Dataset Name	Description	Log Category
ueba-Asset-User-Count-by-Device-Interface-and-Detectiontype	Asset and User Count by Source Device Interface and Detection Method	

```
select
  devname,
  srcintf,
  sum(mac_cnt) as mac_cnt,
  sum(ip_cnt) as ip_cnt,
  sum(ep_count) as ep_count,
  sum(eu_count) as eu_count
from
  (
    (
      select
        devname,
        srcintf,
        sum(
          case when detecttype =& #039;by_mac' then count else 0 end) as mac_cnt, sum(case when
detecttype='by_ip' then count else 0 end) as ip_cnt, sum(count) as ep_count, 0 as eu_count from
(select devname, srcintf, detecttype, count(distinct t1.epid) as count from $ADOM_ENDPOINT t1
inner join $ADOM_EPEU_DEVMAP t2 on t1.epid=t2.epid inner join devtable_ext t3 on t2.devid=t3.devid
where t1.epid>1024 and $filter-drilldown and t1.lastseen>=$start_time and firstseen<$end_time and
devname is not null and srcintf is not null and detecttype in ('by_ip', 'by_mac') group by
devname,srcintf, detecttype order by count desc) t1 group by devname,srcintf order by ep_count
desc) union all (SELECT devname, srcintf, 0 as mac_cnt, 0 as ip_cnt, 0 as ep_count, count(DISTINCT
```

```
euid) as eu_count from (select euid, euname, t3.epid, eugroup, srcintf, devname, devid from
(select t1.euid, euname, epid, eugroup, srcintf, devname, t2.devid from  $ADOM_ENDUSER t1 inner
join $ADOM_EPEU_DEVMAP t2 ON t1.euid=t2.euid inner join devtable_ext t3 on t2.devid=t3.devid where
t1.lastseen>=$start_time and t1.firstseen<$end_time and srcintf is not null ) t3 LEFT JOIN $ADOM_
ENDPOINT t4 ON t3.epid = t4.epid) t5 where euname != '(none)' and epid>1024 and euid>1024 and
$filter-drilldown group by devname, srcintf order by eu_count desc)) t group by devname, srcintf
order by devname, sum(eu_count)+ sum(ep_count) desc
```

Dataset Name	Description	Log Category
ueba-Asset-User-Discovery-by-Time	Asset and User Count by Discovery Time	

```
select
  $flex_timescale(firstseen) as time,
  count(distinct epid) as ep_count,
  count(distinct euid) as eu_count
from
  (
    (
      select
        firstseen,
        t1.epid,
        null as euid
      from
        $ADOM_ENDPOINT t1
        inner join $ADOM_EPEU_DEVMAP t2 on t1.epid = t2.epid
      where
        $filter - drilldown
        and t1.firstseen >= $start_time
        and t1.firstseen<$end_time
        and t1.epid>1024
    )
    union all
      (
        select
          firstseen,
          null as epid,
          t1.euid
        from
          $ADOM_ENDUSER t1
          inner join $ADOM_EPEU_DEVMAP t2 ON t1.euid = t2.euid
        where
          t1.euid>1024
          and $filter - drilldown
          and firstseen >= $start_time
          and firstseen<$end_time
      )
  ) t
group by
  time
```

```
order by
  time
```

Dataset Name	Description	Log Category
dns-Security-Domain-Count-by-Threat-Level	Domain Count by Threat level	dns

```
select
  threat_level,
  total_num
from
  (
    select
      (
        case when tdtype in (
          & #039;infected-domain', 'infected-ip', 'infected-url') then 'critical' when is_botnet
or catdesc in ('Malicious Websites', 'Phishing', 'Spam URLs') then 'high' when catdesc in ('Newly
Observed Domain', 'Newly Registered Domain', 'Proxy Avoidance','Unrated') or catdesc LIKE
'%Dynamic DNS%' then 'medium' end) as threat_level, sum(total_num) as total_num from  ###(select
dvid, qname, coalesce(nullifna(`user`), nullifna(`unauthuser`)) as f_user, dstip, srcip, catdesc,
level, tdtype,  (botnetdomain is not null or botnetip is not null) as is_botnet, min(nanosec_to_
sec(eventtime)) as first_seen, max(nanosec_to_sec(eventtime)) as last_seen, count(*) as total_num
from $log-dns where $filter group by dvid, qname, f_user, dstip, srcip, catdesc, level, tdtype,
is_botnet order by total_num desc)### t group by threat_level order by total_num desc) t where
threat_level is not null order by total_num desc
```

Dataset Name	Description	Log Category
dns-Top-Queried-Domain-Bar	Top Queried Domain	dns

```
select
  qname,
  count(*) as total_num
from
  $log
where
  $filter
  and qname is not null
group by
  qname
order by
  total_num desc
```

Dataset Name	Description	Log Category
dns-Security-Top-Visited-Domain-Categories	Top Visited Domain Categories	dns

```
select
  catdesc,
  sum(total_num) as total_num
from
  ###(select dvid, qname, coalesce(nullifna(`user`), nullifna(`unauthuser`)) as f_user, dstip,
srcip, catdesc, level, tdtype,  (botnetdomain is not null or botnetip is not null) as is_botnet,
min(nanosec_to_sec(eventtime)) as first_seen, max(nanosec_to_sec(eventtime)) as last_seen, count
(*) as total_num from $log-dns where $filter group by dvid, qname, f_user, dstip, srcip, catdesc,
level, tdtype, is_botnet order by total_num desc)### t where catdesc is not null group by catdesc
order by total_num desc
```

Dataset Name	Description	Log Category
dns-Security-Top-Visited-High-Risk-Domain-Categories	Top Visited High Risk Domain Categories	dns

```
select
  catdesc,
  sum(total_num) as total_num
from
  ###(select dvid, qname, coalesce(nullifna(`user`), nullifna(`unauthuser`)) as f_user, dstip,
srcip, catdesc, level, tdtype,  (botnetdomain is not null or botnetip is not null) as is_botnet,
min(nanosec_to_sec(eventtime)) as first_seen, max(nanosec_to_sec(eventtime)) as last_seen, count
(*) as total_num from $log-dns where $filter group by dvid, qname, f_user, dstip, srcip, catdesc,
level, tdtype, is_botnet order by total_num desc)### t where level>='warning' and catdesc is not
null group by catdesc order by total_num desc
```

Dataset Name	Description	Log Category
dns-Security-Top-Domain-with-Botnet-CC-Detected	Top Domain with Botnet C&C Detected	dns

```
select
  qname,
  sum(total_num) as total_num
from
  ###(select $flex_timestamp as timestamp, coalesce(botnetdomain, ipstr(botnetip)) as domain,
qname, cast('Botnet C&C' as char(32)) as malware_type, (case when action='block' then 'Blocked'
when action='redirect' then 'Redirected' else 'Passed' end) as action, srcip, coalesce(nullifna
(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, (CASE WHEN level IN ('critical',
'alert', 'emergency') THEN 5 WHEN level='error' THEN 4 WHEN level='warning' THEN 3 WHEN
level='notice' THEN 2 ELSE 1 END) as sevid, coalesce(botnetdomain, ipstr(botnetip)) as sources_s,
count(*) as total_num from $log-dns where $filter and (botnetdomain is not null or botnetip is not
null) group by timestamp, domain, qname, action, srcip, user_src, sevid order by sevid desc)### t
where qname is not null group by qname order by total_num desc
```

Dataset Name	Description	Log Category
dns-Security-FortiGate-with-Top-Domain-Visited-by_Source-IP	FortiGate with Top Domain Visited by Source IP	dns

```
select
  devname,
  srcip,
  qname,
  category,
  total_num
from
  (
    select
      devname,
      srcip,
      qname,
      category,
      total_num,
      row_number() over (
        partition by devname,
        srcip,
        qname
        order by
          total_num desc,
          qname
      ) as rank
    from
      (
        select
          devname,
          srcip,
          qname,
          max(catdesc) as category,
          sum(total_num) as total_num
        from
          ###(select dvid, qname, coalesce(nullifna(`user`), nullifna(`unauthuser`)) as f_user,
dstip, srcip, catdesc, level, tdtype,  (botnetdomain is not null or botnetip is not null) as is_
botnet, min(nanosec_to_sec(eventtime)) as first_seen, max(nanosec_to_sec(eventtime)) as last_seen,
count(*) as total_num from $log-dns where $filter group by dvid, qname, f_user, dstip, srcip,
catdesc, level, tdtype, is_botnet order by total_num desc)### t1 inner join devtable_ext t2 on
t1.dvid=t2.dvid where qname is not null and srcip is not null group by devname, srcip, qname order
by total_num desc) t) t where rank=1 order by devname, srcip, qname
```

Dataset Name	Description	Log Category
dns-Security-Top-Domain-Lookup-Failure-by-Count	Top Domain Lookup Failures by Count	dns

```
select
  qname,
  count(*) as total_num
from
  $log - dns
where
  $filter
  and qname is not null
  and (
    action =& #039;block' or logid_to_int(logid)=54200) group by qname order by total_num desc
```

Dataset Name	Description	Log Category
dns-Security-Top-Source-IP-by-Destination-Count	Top Source IP by Destination Count	dns

```
select
  srcip,
  count(distinct dstip) as total_num
from
  ###(select dvid, qname, coalesce(nullifna(`user`), nullifna(`unauthuser`)) as f_user, dstip,
srcip, catdesc, level, tdtype,  (botnetdomain is not null or botnetip is not null) as is_botnet,
min(nanosec_to_sec(eventtime)) as first_seen, max(nanosec_to_sec(eventtime)) as last_seen, count
(*) as total_num from $log-dns where $filter group by dvid, qname, f_user, dstip, srcip, catdesc,
level, tdtype, is_botnet order by total_num desc)### t where srcip is not null and dstip is not
null group by srcip order by total_num desc
```

Dataset Name	Description	Log Category
dns-Security-Top-Destination-IP-by-Source-Count	Top Destination IP by Source Count	dns

```
select
  dstip,
  count(distinct srcip) as total_num
from
  ###(select dvid, qname, coalesce(nullifna(`user`), nullifna(`unauthuser`)) as f_user, dstip,
srcip, catdesc, level, tdtype,  (botnetdomain is not null or botnetip is not null) as is_botnet,
min(nanosec_to_sec(eventtime)) as first_seen, max(nanosec_to_sec(eventtime)) as last_seen, count
(*) as total_num from $log-dns where $filter group by dvid, qname, f_user, dstip, srcip, catdesc,
level, tdtype, is_botnet order by total_num desc)### t where srcip is not null and dstip is not
null  group by dstip order by total_num desc
```

Dataset Name	Description	Log Category
dns-Security-Severity-by-High-Risk-Source-IPs-Count	Severity by High Risk Source IPs Count	dns

```
select
  (
    CASE sevid WHEN 5 THEN & #039;Critical' WHEN 4 THEN 'High' WHEN 3 THEN 'Medium' WHEN '2' THEN
'Info' ELSE 'Low' END) as severity, count(distinct srcip) as total_num from (select srcip, (CASE
WHEN level IN ('critical', 'alert', 'emergency') THEN 5 WHEN level='error' THEN 4 WHEN
level='warning' THEN 3 WHEN level='notice' THEN 2 ELSE 1 END) as sevid, count(*) as total_num from
###(select dvid, qname, coalesce(nullifna(`user`), nullifna(`unauthuser`)) as f_user, dstip,
srcip, catdesc, level, tdtype,  (botnetdomain is not null or botnetip is not null) as is_botnet,
min(nanosec_to_sec(eventtime)) as first_seen, max(nanosec_to_sec(eventtime)) as last_seen, count
(*) as total_num from $log-dns where $filter group by dvid, qname, f_user, dstip, srcip, catdesc,
level, tdtype, is_botnet order by total_num desc)### t where level>='warning' and srcip is not
null group by srcip, sevid order by total_num desc) t group by severity having sum(total_num)>0
order by total_num desc
```

Dataset Name	Description	Log Category
dns-Security-Top-DNS-High-Risk-Source-IP	Top DNS High Risk Source IP	dns

```
select
  srcip,
  sum(
    case when sevid = 5 then total_num else 0 end
  ) as num_cri,
  sum(
    case when sevid = 4 then total_num else 0 end
  ) as num_hig,
  sum(
    case when sevid = 3 then total_num else 0 end
  ) as num_med,
  sum(total_num) as total_num
from
  (
    select
      srcip,
      (
        CASE WHEN level IN (
          & #039;critical', 'alert', 'emergency') THEN 5 WHEN level='error' THEN 4 WHEN
level='warning' THEN 3 WHEN level='notice' THEN 2 ELSE 1 END) as sevid, count(*) as total_num from
###(select dvid, qname, coalesce(nullifna(`user`), nullifna(`unauthuser`)) as f_user, dstip,
srcip, catdesc, level, tdtype,  (botnetdomain is not null or botnetip is not null) as is_botnet,
min(nanosec_to_sec(eventtime)) as first_seen, max(nanosec_to_sec(eventtime)) as last_seen, count
(*) as total_num from $log-dns where $filter group by dvid, qname, f_user, dstip, srcip, catdesc,
level, tdtype, is_botnet order by total_num desc)### t where level>='warning' and srcip is not
null group by srcip, sevid order by total_num desc) t group by srcip having sum(total_num)>0 order
by total_num desc
```

Dataset Name	Description	Log Category
dns-Security-Top-Infected-Domain-by-Count	Top Infected Domain by Count	dns

```
select
  qname,
  count(distinct srcip) as total_num
from
  ###(select dvid, qname, coalesce(nullifna(`user`), nullifna(`unauthuser`)) as f_user, dstip,
srcip, catdesc, level, tdtype,  (botnetdomain is not null or botnetip is not null) as is_botnet,
min(nanosec_to_sec(eventtime)) as first_seen, max(nanosec_to_sec(eventtime)) as last_seen, count
(*) as total_num from $log-dns where $filter group by dvid, qname, f_user, dstip, srcip, catdesc,
level, tdtype, is_botnet order by total_num desc)### t where qname is not null and
tdtype='infected-domain' group by qname order by total_num desc
```

Dataset Name	Description	Log Category
dns-Security-Top-Blocked-Domains-by-Reason	Top Blocked Domains by Reason	dns

```
select
  qname,
  msg,
  count(*) as total_num
from
  $log
where
  $filter
  and qname is not null
  and msg LIKE & #039;Domain was blocked%' group by qname, msg order by total_num desc
```

Dataset Name	Description	Log Category
dns-Security-Top-Users-by-Infected-Domain-Visits	Top Users by Infected Domain Visits	dns

```
select
  coalesce(
    f_user,
    ipstr(`srcip`)
  ) as user_src,
  count(distinct qname) as total_num
from
  ###(select dvid, qname, coalesce(nullifna(`user`), nullifna(`unauthuser`)) as f_user, dstip,
srcip, catdesc, level, tdtype,  (botnetdomain is not null or botnetip is not null) as is_botnet,
min(nanosec_to_sec(eventtime)) as first_seen, max(nanosec_to_sec(eventtime)) as last_seen, count
(*) as total_num from $log-dns where $filter group by dvid, qname, f_user, dstip, srcip, catdesc,
```

```
level, tdtype, is_botnet order by total_num desc)### t where qname is not null and
tdtype='infected-domain' and (f_user is not null or srcip is not null) group by user_src order by
total_num desc
```

Dataset Name	Description	Log Category
dns-Security-Top-Users-and-Infected-Domain-by-Visit-Count	Top Users and Infected Domain by Visit Count	dns

```
select
  coalesce(
    f_user,
    ipstr(`srcip`)
  ) as user_src,
  qname,
  sum(total_num) as total_num
from
  ###(select dvid, qname, coalesce(nullifna(`user`), nullifna(`unauthuser`)) as f_user, dstip,
srcip, catdesc, level, tdtype,  (botnetdomain is not null or botnetip is not null) as is_botnet,
min(nanosec_to_sec(eventtime)) as first_seen, max(nanosec_to_sec(eventtime)) as last_seen, count
(*) as total_num from $log-dns where $filter group by dvid, qname, f_user, dstip, srcip, catdesc,
level, tdtype, is_botnet order by total_num desc)### t where qname is not null and (f_user is not
null or srcip is not null) and tdtype='infected-domain' group by user_src, qname order by total_
num desc
```

Dataset Name	Description	Log Category
dns-Security-Top-Users-by-Visited-Domain-Category-Count	Top Users by Visited Domain Category Count	dns

```
select
  coalesce(
    f_user,
    ipstr(`srcip`)
  ) as user_src,
  count(distinct catdesc) as total_num
from
  ###(select dvid, qname, coalesce(nullifna(`user`), nullifna(`unauthuser`)) as f_user, dstip,
srcip, catdesc, level, tdtype,  (botnetdomain is not null or botnetip is not null) as is_botnet,
min(nanosec_to_sec(eventtime)) as first_seen, max(nanosec_to_sec(eventtime)) as last_seen, count
(*) as total_num from $log-dns where $filter group by dvid, qname, f_user, dstip, srcip, catdesc,
level, tdtype, is_botnet order by total_num desc)### t where catdesc is not null and (f_user is
not null or srcip is not null) group by user_src order by total_num desc
```

Dataset Name	Description	Log Category
dns-Security-Top-Users-and-Visited-Domain-Category-by-Count	Top Users and Visited Domain Category by Count	dns

```
select
  coalesce(
    f_user,
    ipstr(`srcip`)
  ) as user_src,
  catdesc,
  srcip,
  sum(total_num) as total_num
from
  ###(select dvid, qname, coalesce(nullifna(`user`), nullifna(`unauthuser`)) as f_user, dstip,
srcip, catdesc, level, tdtype,  (botnetdomain is not null or botnetip is not null) as is_botnet,
min(nanosec_to_sec(eventtime)) as first_seen, max(nanosec_to_sec(eventtime)) as last_seen, count
(*) as total_num from $log-dns where $filter group by dvid, qname, f_user, dstip, srcip, catdesc,
level, tdtype, is_botnet order by total_num desc)### t where catdesc is not null and (f_user is
not null or srcip is not null) group by user_src, catdesc, srcip order by total_num desc
```

Dataset Name	Description	Log Category
dns-Security-Top-Newly-Detected-Domain-by-Count	Top Newly Detected Domain by Count	dns

```
select
  qname,
  sum(total_num) as total_num
from
  ###(select dvid, qname, coalesce(nullifna(`user`), nullifna(`unauthuser`)) as f_user, dstip,
srcip, catdesc, level, tdtype,  (botnetdomain is not null or botnetip is not null) as is_botnet,
min(nanosec_to_sec(eventtime)) as first_seen, max(nanosec_to_sec(eventtime)) as last_seen, count
(*) as total_num from $log-dns where $filter group by dvid, qname, f_user, dstip, srcip, catdesc,
level, tdtype, is_botnet order by total_num desc)### t where last_seen>=$start_time and first_
seen<$end_time and tdtype is not null and qname is not null group by qname order by total_num desc
```

Dataset Name	Description	Log Category
dns-Security-Top-Newly-Detected-Domain-and-Source-IP-with-First-Seen-and-Last-Seen	Top Newly Detected Domain and Source IP with First Seen and Last Seen	dns

```
select
  qname,
  srcip,
  from_itime(
    min(first_seen)
  ) as first_seen,
  from_itime(
    max(last_seen)
  ) as last_seen,
  sum(total_num) as total_num
from
```

```
  ###(select dvid, qname, coalesce(nullifna(`user`), nullifna(`unauthuser`)) as f_user, dstip,
srcip, catdesc, level, tdtype,  (botnetdomain is not null or botnetip is not null) as is_botnet,
min(nanosec_to_sec(eventtime)) as first_seen, max(nanosec_to_sec(eventtime)) as last_seen, count
(*) as total_num from $log-dns where $filter group by dvid, qname, f_user, dstip, srcip, catdesc,
level, tdtype, is_botnet order by total_num desc)### t where last_seen>=$start_time and first_
seen<$end_time and tdtype is not null and qname is not null group by qname, srcip order by total_
num desc
```

Dataset Name	Description	Log Category
web-Usage-Top-User-Category-By-Count	Top Web User and Category by Count	traffic

```
select
  coalesce(
    firstname || & #039; ' || lastname, euname, usersrc) as user_src, catdesc, requests, sum
(requests) over (partition by usersrc) as total_num from ###(select $flex_timestamp as timestamp,
coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as usersrc, euid, catdesc,
hostname as website, ebtr_agg_flat($browse_time) as browsetime, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0))
as traffic_out, count(*) as requests from $log-traffic where $filter and (logflag&1>0) and
(countweb>0 or ((logver is null or logver<502000000) and (hostname is not null or utmevent in
('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter')))) group by
timestamp, usersrc, euid, catdesc,website /*SkipSTART*/order by bandwidth desc/*SkipEND*/)### t1
left join $ADOM_ENDUSER t3 on t1.euid=t3.euid where usersrc is not null and catdesc<>'Unknown'
order by total_num desc, user_src
```

Dataset Name	Description	Log Category
web-Usage-Top-User-Category-by-Browsing-Time	Web Usage Top User and Category by Browsing Time	traffic

```
select
  coalesce(
    firstname || & #039; ' || lastname, euname, usersrc) as user_src, catdesc, ebtr_value(ebtr_
agg_flat(browsetime), null, $timespan) as browsetime from ###(select $flex_timestamp as timestamp,
coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as usersrc, euid, catdesc,
hostname as website, ebtr_agg_flat($browse_time) as browsetime, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0))
as traffic_out, count(*) as requests from $log-traffic where $filter and (logflag&1>0) and
(countweb>0 or ((logver is null or logver<502000000) and (hostname is not null or utmevent in
('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter')))) group by
timestamp, usersrc, euid, catdesc,website /*SkipSTART*/order by bandwidth desc/*SkipEND*/)### t1
left join $ADOM_ENDUSER t3 on t1.euid=t3.euid where usersrc is not null group by user_src, catdesc
order by browsetime desc, user_src, catdesc
```

Dataset Name	Description	Log Category
web-Usage-Count-By-Allowed-Blocked	Web Usage Allowed and Blocked Count	webfilter

```
select
  unnest(type) as allow_block,
  unnest(request_cnt) as totoal_num
from
  (
    select
      array[ & #039;Allowed', 'Blocked'] as type, array[sum(case when action!='blocked' then
requests end), sum(case when action='blocked' then requests end)] as request_cnt from ###(select
$flex_timestamp as timestamp, coalesce(nullifna(`user`), ipstr(`srcip`)) as usersrc, euid, action,
count(*) as requests from $log-webfilter where $filter and coalesce(nullifna(`user`), ipstr
(`srcip`)) is not null group by timestamp, usersrc, euid, action /*SkipSTART*/order by requests
desc, timestamp desc/*SkipEND*/)### t) t
```

Dataset Name	Description	Log Category
web-Usage-Top-Web-Users-By-Allowed-Requests	Web Usage Top Web Users by Allowed Requests	webfilter

```
select
  coalesce(
    firstname || & #039; ' || lastname, euname, usersrc) as user_src, sum(requests) as requests
from ###(select $flex_timestamp as timestamp, coalesce(nullifna(`user`), ipstr(`srcip`)) as
usersrc, euid, action, count(*) as requests from $log-webfilter where $filter and coalesce
(nullifna(`user`), ipstr(`srcip`)) is not null group by timestamp, usersrc, euid, action
/*SkipSTART*/order by requests desc, timestamp desc/*SkipEND*/)### t1 left join $ADOM_ENDUSER t3
on t1.euid=t3.euid where action!='blocked' group by user_src order by requests desc
```

Dataset Name	Description	Log Category
web-Usage-Top-Web-Users-By-Blocked-Requests	Web Usage Top Web Users by Blocked Requests	webfilter

```
select
  coalesce(
    firstname || & #039; ' || lastname, euname, usersrc) as user_src, sum(requests) as requests
from ###(select $flex_timestamp as timestamp, coalesce(nullifna(`user`), ipstr(`srcip`)) as
usersrc, euid, action, count(*) as requests from $log-webfilter where $filter and coalesce
(nullifna(`user`), ipstr(`srcip`)) is not null group by timestamp, usersrc, euid, action
/*SkipSTART*/order by requests desc, timestamp desc/*SkipEND*/)### t1 left join $ADOM_ENDUSER t3
on t1.euid=t3.euid where action='blocked' group by user_src order by requests desc
```

Dataset Name	Description	Log Category
web-Usage-Request-Summary-Timeline	Webfilter web activity summary by requests	webfilter

```
select
  $flex_timescale(timestamp) as hodex,
  sum(allowed_request) as allowed_request,
  sum(blocked_request) as blocked_request
from
  ###(select $flex_timestamp as timestamp, sum(case when action!='blocked' then 1 else 0 end) as
allowed_request, sum(case when action='blocked' then 1 else 0 end) as blocked_request from $log
where $filter group by timestamp /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t group by
hodex order by hodex
```

Dataset Name	Description	Log Category
web-Usage-Bandwidth-Timeline	Web Usage Bandwidth Timeline	traffic

```
select
  $flex_timescale(timestamp) as hodex,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out
from
  ###(select $flex_timestamp as timestamp, coalesce(nullifna(`user`), nullifna(`unauthuser`),
ipstr(`srcip`)) as usersrc, euid, catdesc, hostname as website, ebtr_agg_flat($browse_time) as
browsetime, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte,
0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out, count(*) as requests from $log-
traffic where $filter and (logflag&1>0) and (countweb>0 or ((logver is null or logver<502000000)
and (hostname is not null or utmevent in ('webfilter', 'banned-word', 'web-content', 'command-
block', 'script-filter')))) group by timestamp, usersrc, euid, catdesc,website /*SkipSTART*/order
by bandwidth desc/*SkipEND*/)### t group by hodex order by hodex
```

Dataset Name	Description	Log Category
web-Usage-Top-Web-Users-By-Requests	Web Usage Top Web Users by Requests	webfilter

```
select
  coalesce(
    firstname || & #039; ' || lastname, euname, usersrc) as user_src, sum(requests) as requests
from ###(select $flex_timestamp as timestamp, coalesce(nullifna(`user`), ipstr(`srcip`)) as
usersrc, euid, action, count(*) as requests from $log-webfilter where $filter and coalesce
(nullifna(`user`), ipstr(`srcip`)) is not null group by timestamp, usersrc, euid, action
/*SkipSTART*/order by requests desc, timestamp desc/*SkipEND*/)### t1 left join $ADOM_ENDUSER t3
on t1.euid=t3.euid where usersrc is not null group by user_src order by requests desc
```

Dataset Name	Description	Log Category
web-Usage-Top-Web-Users-By-Requests-Timeline	Web Usage top Web Users by Requests Timeline	webfilter

```
with time_users as (
  select
    $flex_timescale(timestamp) as hodex,
    coalesce(
      firstname || & #039; ' || lastname, euname, usersrc) as user_src, sum(requests) as requests
from (select timestamp, usersrc, euid, requests from ###(select $flex_timestamp as timestamp,
coalesce(nullifna(`user`), ipstr(`srcip`)) as usersrc, euid, action, count(*) as requests from
$log-webfilter where $filter and coalesce(nullifna(`user`), ipstr(`srcip`)) is not null group by
timestamp, usersrc, euid, action /*SkipSTART*/order by requests desc, timestamp
desc/*SkipEND*/)### t where usersrc is not null) t1 left join $ADOM_ENDUSER t3 on t1.euid=t3.euid
group by hodex, user_src order by hodex), top_users as (select user_src, sum(requests) as requests
from time_users group by user_src order by requests desc limit $ddown-top) select hodex, user_src,
requests from time_users t where exists (select 1 from top_users where user_src=t.user_src) order
by hodex
```

Dataset Name	Description	Log Category
web-Usage-Top-Category-Sites-By-Session	Web top user visted websites by session	webfilter

```
select
  website,
  catdesc,
  sum(sessions) as sessions
from
  ###(select hostname as website, catdesc, count(*) as sessions from $log where $filter and
hostname is not null group by hostname, catdesc order by sessions desc)### t where catdesc is not
null group by website, catdesc order by sessions desc
```

Dataset Name	Description	Log Category
web-Usage-Top-User-Browsing-Time	Web Usage Top User Browsing Time	traffic

```
select
  user_src,
  sum(browsetime) as browsetime
from
  (
    select
      coalesce(
        firstname || & #039; ' || lastname, euname, usersrc) as user_src, catdesc, ebtr_value
(ebtr_agg_flat(browsetime), null, $timespan) as browsetime from ###(select $flex_timestamp as
```

```
timestamp, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as usersrc, euid,
catdesc, hostname as website, ebtr_agg_flat($browse_time) as browsetime, sum(coalesce(sentbyte,
0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce
(sentbyte, 0)) as traffic_out, count(*) as requests from $log-traffic where $filter and
(logflag&1>0) and (countweb>0 or ((logver is null or logver<502000000) and (hostname is not null
or utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter'))))
group by timestamp, usersrc, euid, catdesc,website /*SkipSTART*/order by bandwidth
desc/*SkipEND*/)### t1 left join $ADOM_ENDUSER t3 on t1.euid=t3.euid where usersrc is not null
group by user_src, catdesc order by browsetime desc) t group by user_src order by browsetime desc,
user_src
```

Dataset Name	Description	Log Category
web-Usage-Top-Category-By-Website-Browsetime	Top Category By Website Browsetime	traffic

```
select
  catdesc,
  ebtr_value(
    ebtr_agg_flat(browsetime),
    null,
    $timespan
  ) as browsetime
from
  ###(select $flex_timestamp as timestamp, coalesce(nullifna(`user`), nullifna(`unauthuser`),
ipstr(`srcip`)) as usersrc, euid, catdesc, hostname as website, ebtr_agg_flat($browse_time) as
browsetime, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte,
0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out, count(*) as requests from $log-
traffic where $filter and (logflag&1>0) and (countweb>0 or ((logver is null or logver<502000000)
and (hostname is not null or utmevent in ('webfilter', 'banned-word', 'web-content', 'command-
block', 'script-filter')))) group by timestamp, usersrc, euid, catdesc,website /*SkipSTART*/order
by bandwidth desc/*SkipEND*/)### t where catdesc!='Unrated' and browsetime is not null group by
catdesc order by browsetime desc
```

Dataset Name	Description	Log Category
web-Usage-Top-Sites-By-Browsing-Time	Web Usage Top Websites by Browsing Time	traffic

```
select
  website,
  max(catdesc) as catdesc,
  ebtr_value(
    ebtr_agg_flat(browsetime),
    null,
    $timespan
  ) as browsetime,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
```

```
  sum(traffic_out) as traffic_out
from
  ###(select $flex_timestamp as timestamp, coalesce(nullifna(`user`), nullifna(`unauthuser`),
ipstr(`srcip`)) as usersrc, euid, catdesc, hostname as website, ebtr_agg_flat($browse_time) as
browsetime, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte,
0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out, count(*) as requests from $log-
traffic where $filter and (logflag&1>0) and (countweb>0 or ((logver is null or logver<502000000)
and (hostname is not null or utmevent in ('webfilter', 'banned-word', 'web-content', 'command-
block', 'script-filter')))) group by timestamp, usersrc, euid, catdesc,website /*SkipSTART*/order
by bandwidth desc/*SkipEND*/)### t where website is not null and catdesc is not null group by
website order by browsetime desc
```

Dataset Name	Description	Log Category
web-Usage-Top-User-By-Bandwidth	Web Usage Top User By Bandwidth	traffic

```
select
  coalesce(
    firstname || & #039; ' || lastname, euname, usersrc) as user_src, sum(bandwidth) as bandwidth
from ###(select $flex_timestamp as timestamp, coalesce(nullifna(`user`), nullifna(`unauthuser`),
ipstr(`srcip`)) as usersrc, euid, catdesc, hostname as website, ebtr_agg_flat($browse_time) as
browsetime, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte,
0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out, count(*) as requests from $log-
traffic where $filter and (logflag&1>0) and (countweb>0 or ((logver is null or logver<502000000)
and (hostname is not null or utmevent in ('webfilter', 'banned-word', 'web-content', 'command-
block', 'script-filter')))) group by timestamp, usersrc, euid, catdesc,website /*SkipSTART*/order
by bandwidth desc/*SkipEND*/)### t1 left join $ADOM_ENDUSER t3 on t1.euid=t3.euid where
bandwidth>0 group by user_src order by bandwidth desc
```

Dataset Name	Description	Log Category
web-Usage-Top-User-By-Bandwidth-Timeline	Web Usage Top User By Bandwidth Timeline	traffic

```
with time_users as (
  select
    $flex_timescale(timestamp) as hodex,
    coalesce(
      firstname || & #039; ' || lastname, euname, usersrc) as user_src, sum(bandwidth) as
bandwidth from (select timestamp, usersrc, euid, bandwidth from ###(select $flex_timestamp as
timestamp, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as usersrc, euid,
catdesc, hostname as website, ebtr_agg_flat($browse_time) as browsetime, sum(coalesce(sentbyte,
0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce
(sentbyte, 0)) as traffic_out, count(*) as requests from $log-traffic where $filter and
(logflag&1>0) and (countweb>0 or ((logver is null or logver<502000000) and (hostname is not null
or utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter'))))
group by timestamp, usersrc, euid, catdesc,website /*SkipSTART*/order by bandwidth
desc/*SkipEND*/)### t where usersrc is not null) t1 left join $ADOM_ENDUSER t3 on t1.euid=t3.euid
```

```
group by hodex, user_src order by bandwidth desc), top_users as (select user_src, sum(bandwidth)
as bandwidth from time_users where bandwidth>0 group by user_src order by bandwidth desc limit
$ddown-top) select hodex, user_src, bandwidth from time_users t where exists (select 1 from top_
users where user_src=t.user_src) order by hodex
```

Dataset Name	Description	Log Category
web-Usage-Top-Category-Website-By-Bandwidth	Web Usage Top Web Category and Websites by Bandwidth	traffic

```
select
  catdesc,
  website,
  bandwidth,
  sum(bandwidth) over (partition by catdesc) as sub_bandwidth
from
  (
    select
      website,
      catdesc,
      sum(bandwidth) as bandwidth
    from
      ###(select $flex_timestamp as timestamp, coalesce(nullifna(`user`), nullifna(`unauthuser`),
ipstr(`srcip`)) as usersrc, euid, catdesc, hostname as website, ebtr_agg_flat($browse_time) as
browsetime, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte,
0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out, count(*) as requests from $log-
traffic where $filter and (logflag&1>0) and (countweb>0 or ((logver is null or logver<502000000)
and (hostname is not null or utmevent in ('webfilter', 'banned-word', 'web-content', 'command-
block', 'script-filter')))) group by timestamp, usersrc, euid, catdesc,website /*SkipSTART*/order
by bandwidth desc/*SkipEND*/)### t where  website is not null and catdesc is not null group by
website, catdesc order by bandwidth desc) t order by sub_bandwidth desc, catdesc
```

Dataset Name	Description	Log Category
web-Usage-Top-Blocked-User-Category-By-Request	Web Usage Top Blocked Web User and Category by Request	webfilter

```
select
  user_src,
  catdesc,
  requests,
  sum(requests) over (partition by user_src) as total_num
from
  (
    select
      coalesce(
        firstname || & #039; ' || lastname, euname, usersrc) as user_src, catdesc, sum(requests)
as requests from  ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as usersrc, euid,
hostname, catdesc, action, count(*) as requests from $log where $filter group by usersrc, euid,
```

```
hostname, catdesc, action order by requests desc)### t1 left join $ADOM_ENDUSER t3 on
t1.euid=t3.euid where usersrc is not null and catdesc<>'Unknown' and action='blocked' group by
user_src, catdesc order by requests desc) t order by total_num desc, user_src
```

Dataset Name	Description	Log Category
web-Usage-Top-Web-Users-By-Blocked-Requests-Timeline	Web Usage Top Web Users Timeline by Blocked Requests	webfilter

```
with time_users as (
  select
    $flex_timescale(timestamp) as hodex,
    coalesce(
      firstname || & #039; ' || lastname, euname, usersrc) as user_src, sum(requests) as requests
from (select timestamp, usersrc, euid, requests from ###(select $flex_timestamp as timestamp,
coalesce(nullifna(`user`), ipstr(`srcip`)) as usersrc, euid, action, count(*) as requests from
$log-webfilter where $filter and coalesce(nullifna(`user`), ipstr(`srcip`)) is not null group by
timestamp, usersrc, euid, action /*SkipSTART*/order by requests desc, timestamp
desc/*SkipEND*/)### t where usersrc is not null and action='blocked') t1 left join $ADOM_ENDUSER
t3 on t1.euid=t3.euid group by hodex, user_src order by hodex), top_users as (select user_src, sum
(requests) as requests from time_users group by user_src order by requests desc limit $ddown-top)
select hodex, user_src, requests from time_users t where exists (select 1 from top_users where
user_src=t.user_src) order by hodex
```

Dataset Name	Description	Log Category
web-Usage-Top-Blocked-Web-Categories-by-Request	Web Usage Top Blocked Web Categories by Request	webfilter

```
select
  catdesc,
  hostname,
  sum(requests) as requests
from
  ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as usersrc, euid, hostname, catdesc,
action, count(*) as requests from $log where $filter group by usersrc, euid, hostname, catdesc,
action order by requests desc)### t1 where catdesc is not null and hostname is not null and
action='blocked' group by catdesc, hostname order by requests desc
```

Dataset Name	Description	Log Category
web-Usage-Browsing-Time-Summary-Timeline	Traffic browsing time summary	traffic

```
select
  $flex_timescale(timestamp) as hodex,
  cast(
    ebtr_value(
```

```
      ebtr_agg_flat(browsetime),
      null,
      $timespan
    )/ 60.0 as decimal(18, 2)
  ) as browsetime
from
  ###(select $flex_timestamp as timestamp, ebtr_agg_flat($browse_time) as browsetime from $log
where $filter and (logflag&1>0) and $browse_time is not null group by timestamp /*SkipSTART*/order
by timestamp desc/*SkipEND*/)### t group by hodex order by hodex
```

Dataset Name	Description	Log Category
360-security-Rating-Asset-Endpoint-HWOS-Count	Asset Endpoint Count by OS	

```
select
  osname,
  count(distinct t2.epid) as count
from
  $ADOM_ENDPOINT t1
  inner join $ADOM_EPEU_DEVMAP t2 on t1.epid = t2.epid
where
  exists (
    select
      1
    from
      devtable_ext t3
    where
      $dev_filter
      and t3.devid = t2.devid
  )
  and lastseen >= $start_time
  and firstseen<$end_time
  and osname is not null
  and t2.epid>1024
group by
  osname
order by
  count desc
```

Dataset Name	Description	Log Category
360-security-daily-Summary-Traffic-Session-Line	Daily Summary - Traffic Bandwidth Line	traffic

```
select
  $fv_line_timescale(timescale) as time,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out,
```

```
  sum(session_block) as session_block,
  (
    sum(sessions)- sum(session_block)
  ) as session_pass
from
  (
    (
      select
        timescale,
        sum(traffic_in) as traffic_in,
        sum(traffic_out) as traffic_out,
        sum(session_block) as session_block,
        sum(sessions) as sessions
      from
        t
      group by
        timescale
    )
    union all
      (
        select
          timescale,
          sum(traffic_in) as traffic_in,
          sum(traffic_out) as traffic_out,
          sum(session_block) as session_block,
          sum(sessions) as sessions
        from
          t
        group by
          timescale
      )
  ) t
group by
  time
order by
  time
```

Dataset Name	Description	Log Category
360-security-wifi-WiFi-Client-Number-Timeline	WiFi client Number Timeline	event

```
select
  $flex_timescale(timestamp) as hodex,
  count(
    distinct (
      case when radioband =& #039;5G' then stamac else NULL end)) as g5, count(distinct (case when
radioband='2G' then stamac else NULL end)) as g2 from ###(select $flex_timestamp as timestamp,
stamac, radioband from $log  where $filter and subtype='wireless' group by timestamp, stamac,
radioband /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t group by hodex order by hodex
```

Dataset Name	Description	Log Category
360-security-ueba-Asset-Count-by-HWOS-Donut	Asset Count by Hardware OS	

```
select
  osname,
  count(distinct t2.epid) as count
from
  $ADOM_ENDPOINT t1
  inner join $ADOM_EPEU_DEVMAP t2 on t1.epid = t2.epid
where
  $filter - drilldown
  and lastseen >= $start_time
  and firstseen<$end_time
  and osname is not null
  and t2.epid>1024
group by
  osname
order by
  count desc
```

Dataset Name	Description	Log Category
360-security-Rating-Posture-Stats-Status-Count	Posture Security Rating Statistic Status Count	

```
select
  unnest(name) as stats,
  unnest(val) as value
from
  (
    select
      array[ & #039;Passed','Failed','Exempt','Unmet'] as name, array[(sum
(passedchkcnt::int)/count(*)), sum((failedchkcnt-unmetchkcnt)::int)/count(*), sum((data-
>'statistics'->'numExemptChecks')::int)/count(*), sum(unmetchkcnt::int)/count(*)] as val from
$ADOMTBL_PLHD_AUDIT_HST t inner join devtable_ext td on td.dvid = t.dvid where $filter-drilldown
and $cust_time_filter(itime) and reporttype='PostureReport') t
```

Dataset Name	Description	Log Category
360-security-Rating-Coverage-Stats-Status-Count	Fabric Coverage Security Rating Statistic Status Count	

```
select
  unnest(name) as stats,
  unnest(val) as value
from
```

```
  (
    select
      array[ & #039;Passed','Failed','Exempt'] as name, array[(sum(passedchkcnt::int)/count(*)),
sum(failedchkcnt::int)/count(*), sum((data->'statistics'->'numExemptChecks')::int)/count(*)] as
val from $ADOMTBL_PLHD_AUDIT_HST t inner join devtable_ext td on td.dvid = t.dvid where $filter-
drilldown and $cust_time_filter(itime) and reporttype='CoverageReport') t
```

Dataset Name	Description	Log Category
360-security-Rating-Optimize-Stats-Status-Count	Optimization Security Rating Statistic Status Count	

```
select
  unnest(name) as stats,
  unnest(val) as value
from
  (
    select
      array[ & #039;Passed','Failed','Exempt'] as name, array[(sum(passedchkcnt::int)/count(*)),
sum(failedchkcnt::int)/count(*), sum((data->'statistics'->'numExemptChecks')::int)/count(*)] as
val from $ADOMTBL_PLHD_AUDIT_HST t inner join devtable_ext td on td.dvid = t.dvid where $filter-
drilldown and $cust_time_filter(itime) and reporttype='OptimizationReport') t
```

Dataset Name	Description	Log Category
360-security-Rating-Asset-Count-by-HWVendor	Asset Count by Hardware Vendor	

```
select
  (
    case when hwvendor =& #039;Fortinet' then hwvendor else 'Other identified device' end) as
vendor, sum(total_num) as total_num from (select osname, hwvendor, srcintf, count(distinct
t1.epid) as total_num from $ADOM_ENDPOINT t1 inner join $ADOM_EPEU_DEVMAP t2 on t1.epid=t2.epid
where exists (select 1 from devtable_ext t3 where $dev_filter and t3.devid=t2.devid) and
lastseen>=$start_time and firstseen<$end_time and hwvendor is not null and osname is not null and
t2.srcintf is not null and t2.epid>1024 group by osname, hwvendor, srcintf order by total_num
desc) t group by vendor order by vendor
```

Dataset Name	Description	Log Category
360-security-Rating-Asset-Count-by-HWOS-List	Asset Count by Hardware OS List	

```
select
  osname,
  sum(total_num) as total_num
from
  (
```

```
    select
      osname,
      hwvendor,
      srcintf,
      count(distinct t1.epid) as total_num
    from
      $ADOM_ENDPOINT t1
      inner join $ADOM_EPEU_DEVMAP t2 on t1.epid = t2.epid
    where
      exists (
        select
          1
        from
          devtable_ext t3
        where
          $dev_filter
          and t3.devid = t2.devid
      )
      and lastseen >= $start_time
      and firstseen<$end_time
      and hwvendor is not null
      and osname is not null
      and t2.srcintf is not null
      and t2.epid>1024
    group by
      osname,
      hwvendor,
      srcintf
    order by
      total_num desc
  ) t
group by
  osname
order by
  total_num desc
```

Dataset Name	Description	Log Category
360-security-Rating-Asset-Count-by-Interface	Asset Count by Interface	

```
select
  srcintf,
  sum(total_num) as count
from
  (
    select
      osname,
      hwvendor,
      srcintf,
      count(distinct t1.epid) as total_num
```

```
    from
      $ADOM_ENDPOINT t1
      inner join $ADOM_EPEU_DEVMAP t2 on t1.epid = t2.epid
    where
      exists (
        select
          1
        from
          devtable_ext t3
        where
          $dev_filter
          and t3.devid = t2.devid
      )
      and lastseen >= $start_time
      and firstseen<$end_time
      and hwvendor is not null
      and osname is not null
      and t2.srcintf is not null
      and t2.epid>1024
    group by
      osname,
      hwvendor,
      srcintf
    order by
      total_num desc
  ) t
group by
  srcintf
order by
  count desc
```

Dataset Name	Description	Log Category
360-security-Rating-Asset-List-From-Fortinet	Asset List from Fortinet	traffic

```
select
  coalesce(
    epname,
    ipstr(`srcip`)
  ) as ep_name,
  coalesce(
    ipstr(epip) || & #039; ' || mac::text, ipstr(`srcip`)) as addr, osname, hwfamily, hwversion,
coalesce(osname, max(epdevtype)) as devtype, sum(sessions) as sessions from (select dvid, epid,
srcip, sum(sessions) as sessions from ###(select dvid, $flex_timestamp as timestamp, epid, srcip,
policyname, policyid, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as
bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte,
0)) as traffic_out, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block, sum(CASE
WHEN (logflag&1>0) THEN 1 ELSE 0 END) AS sessions from $log-traffic where $filter and (logflag&
(1|32)>0) group by dvid, timestamp, epid, srcip, policyname, policyid order by bandwidth desc)###
t where epid>1024 group by dvid, epid, srcip) t1 inner join (select epid, srcmac as epmac, dvid
```

```
from $ADOM_EPEU_DEVMAP dm inner join devtable dt ON dm.devid=dt.devid and dm.vd=dt.vd) t2 on
t1.epid=t2.epid and t1.dvid=t2.dvid left join $ADOM_ENDPOINT t3 on t1.epid=t3.epid and
t2.epmac=t3.mac where hwvendor='Fortinet' group by ep_name, addr, osname, hwfamily, hwversion
order by sessions desc
```

Dataset Name	Description	Log Category
360-security-Rating-Asset-List-From-Other-Identified-Device	Asset List from Other Identified Device	traffic

```
select
  coalesce(
    epname,
    ipstr(`srcip`)
  ) as ep_name,
  coalesce(
    ipstr(epip) || & #039; ' || mac::text, ipstr(`srcip`)) as addr, osname, hwfamily, hwversion,
coalesce(osname, max(epdevtype)) as devtype, sum(sessions) as sessions from (select dvid, epid,
srcip, sum(sessions) as sessions from ###(select dvid, $flex_timestamp as timestamp, epid, srcip,
policyname, policyid, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as
bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte,
0)) as traffic_out, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block, sum(CASE
WHEN (logflag&1>0) THEN 1 ELSE 0 END) AS sessions from $log-traffic where $filter and (logflag&
(1|32)>0) group by dvid, timestamp, epid, srcip, policyname, policyid order by bandwidth desc)###
t where epid>1024 group by dvid, epid, srcip) t1 inner join (select epid, srcmac as epmac, dvid
from $ADOM_EPEU_DEVMAP dm inner join devtable dt ON dm.devid=dt.devid and dm.vd=dt.vd) t2 on
t1.epid=t2.epid and t1.dvid=t2.dvid left join $ADOM_ENDPOINT t3 on t1.epid=t3.epid and
t2.epmac=t3.mac where hwvendor<>'Fortinet' group by ep_name, addr, osname, hwfamily, hwversion
order by sessions desc
```

Dataset Name	Description	Log Category
360-security-wifi-AP-WaitingAuth-Online-Offline-Count	WiFi AP count by Waiting Auth Online and Offline Status	event

```
select
  *
from
  (
    select
      unnest(status) as ap_status,
      unnest(num) as totalnum
    from
      (
        select
          array[ & #039;Online', 'Offline'] as status, array[sum(case when onwire!='no' or onwire
is null then 1 end), sum(case when onwire='no' then 1 end)] as num from ###(select apstatus,
bssid, ssid, onwire, count(*) as subtotal from $log where $filter and apstatus is not null and
apstatus!=0 and bssid is not null and logid_to_int(logid) in (43527, 43521, 43525, 43563, 43564,
```

```
43565, 43566, 43569, 43570, 43571, 43582, 43583, 43584, 43585) group by apstatus, bssid, ssid,
onwire order by subtotal desc)### t)t union all (select ap_status, totalnum from ###(select (case
when not (action like '%join%') then 'Waiting for Authentication' end) as ap_status, count(*) as
totalnum from $log where $filter and logid_to_int(logid) in (43522, 43551) group by ap_status
order by totalnum desc)### t)) t where ap_status is not null and totalnum>0
```

Dataset Name	Description	Log Category
360-security-wifi-Top-AP-By-Client	WiFi Top Access Point by Client	event

```
select
  ap,
  count(distinct lmac) as totalnum
from
  ###(select ap, stamac as lmac, ssid, action, max(dtime) as last from $log-event where $filter
and ssid is not null group by ap, lmac, ssid, action order by last desc)### t group by ap order by
totalnum desc
```

Dataset Name	Description	Log Category
360-security-wifi-Signal-By-Client	WiFi Signal by Client	event

```
select
  sig_status,
  count(distinct lmac) as totalnum
from
  ###(select ap, stamac as lmac, ssid, action, (case when signal>=-65 then 'Good (>=-65dBm)' when
signal<-75 then 'Poor (<-75dBm)' end) as sig_status, max(dtime) as last from $log-event where
$filter and ssid is not null group by ap, lmac, ssid, action, sig_status order by last desc)### t
where sig_status is not null group by sig_status order by totalnum desc
```

Dataset Name	Description	Log Category
360-security-wifi-Auth-Failure-Event	WiFi Authentication Failure Event	event

```
select
  ssid,
  from_dtime(last) as last
from
  ###(select ap, stamac as lmac, ssid, action, max(dtime) as last from $log-event where $filter
and ssid is not null group by ap, lmac, ssid, action order by last desc)### t where action like
'%auth-failure' order by last desc
```

Dataset Name	Description	Log Category
360-security-Top-Policy-Bandwidth-Timeline	Top Policy Bandwidth Timeline	traffic

```
select
  timestamp,
  policy,
  bandwidth,
  sum(bandwidth) over (partition by policy) as total_bandwidth
from
  (
    select
      timestamp,
      t1.policy,
      t1.bandwidth
    from
      (
        select
          $fv_line_timescale(timestamp) as timestamp,
          coalesce(policyname, policyid : :text) as policy,
          sum(bandwidth) as bandwidth
        FROM
          ###(select dvid, $flex_timestamp as timestamp, epid, srcip, policyname, policyid, sum
(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce
(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum
((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block, sum(CASE WHEN (logflag&1>0) THEN 1
ELSE 0 END) AS sessions from $log-traffic where $filter and (logflag&(1|32)>0) group by dvid,
timestamp, epid, srcip, policyname, policyid order by bandwidth desc)### t group by timestamp,
policy order by timestamp) t1 inner join (select coalesce(policyname, policyid::text) as policy,
sum(bandwidth) as bandwidth FROM ###(select dvid, $flex_timestamp as timestamp, epid, srcip,
policyname, policyid, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as
bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte,
0)) as traffic_out, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block, sum(CASE
WHEN (logflag&1>0) THEN 1 ELSE 0 END) AS sessions from $log-traffic where $filter and (logflag&
(1|32)>0) group by dvid, timestamp, epid, srcip, policyname, policyid order by bandwidth desc)###
t where coalesce(policyname, policyid::text) is not null and bandwidth>0  group by policy order by
bandwidth desc limit $ddown-top) t2 on t1.policy=t2.policy order by timestamp) t order by
timestamp, total_bandwidth desc
```

Dataset Name	Description	Log Category
360-security-Policy-by-Bandwidth	Top Policy by Bandwidth	traffic

```
select
  policy,
  sum(bandwidth) as bandwidth
FROM
  ###(select coalesce(policyname, policyid::text) as policy, max(policytype) as policytype,
srcintf, dstintf, max(devname) as devname, max(vd) as vd, sum(coalesce(sentdelta, sentbyte,
```

```
0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END)
AS sessions, from_dtime(max(dtime)) as time_stamp from $log-traffic where $filter and (logflag&
(1|32)>0) and coalesce(policyname, policyid::text) is not null group by policy, srcintf, dstintf
order by bandwidth desc)### t where bandwidth>0 group by policy order by bandwidth desc
```

Dataset Name	Description	Log Category
360-security-Policy-by-Session	Top Policy by Session	traffic

```
select
  coalesce(policyname, policyid : :text) as policy,
  sum(sessions) as sessions
FROM
  ###(select dvid, $flex_timestamp as timestamp, epid, srcip, policyname, policyid, sum(coalesce
(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta,
rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum((CASE WHEN
(logflag&2>0) THEN 1 ELSE 0 END)) AS session_block, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END)
AS sessions from $log-traffic where $filter and (logflag&(1|32)>0) group by dvid, timestamp, epid,
srcip, policyname, policyid order by bandwidth desc)### t where policyid is not null group by
policy order by sessions desc
```

Dataset Name	Description	Log Category
360-security-Policy-Details	Top Policy with Details by Bandwidth	traffic

```
select
  policy,
  max(policytype) as policytype,
  string_agg(
    distinct srcintf,
    & #039;,') as srcintf, string_agg(distinct dstintf, ',') as dstintf, max(devname) as devname,
max(vd) as vd, sum(bandwidth) as bandwidth, sum(sessions) as sessions, max(time_stamp) as time_
stamp from ###(select coalesce(policyname, policyid::text) as policy, max(policytype) as
policytype, srcintf, dstintf, max(devname) as devname, max(vd) as vd, sum(coalesce(sentdelta,
sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1
ELSE 0 END) AS sessions, from_dtime(max(dtime)) as time_stamp from $log-traffic where $filter and
(logflag&(1|32)>0) and coalesce(policyname, policyid::text) is not null group by policy, srcintf,
dstintf order by bandwidth desc)### t where bandwidth>0 group by policy order by bandwidth desc
```

Dataset Name	Description	Log Category
360-security-Top-Source-Session-Timeline	Top Source Session Timeline	traffic

```
select
  $fv_line_timescale(timestamp) as timestamp,
  sum(session_block) as session_block,
  (
```

```
    sum(sessions)- sum(session_block)
  ) as session_pass
FROM
  ###(select dvid, $flex_timestamp as timestamp, epid, srcip, policyname, policyid, sum(coalesce
(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta,
rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum((CASE WHEN
(logflag&2>0) THEN 1 ELSE 0 END)) AS session_block, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END)
AS sessions from $log-traffic where $filter and (logflag&(1|32)>0) group by dvid, timestamp, epid,
srcip, policyname, policyid order by bandwidth desc)### t group by timestamp order by timestamp
```

Dataset Name	Description	Log Category
360-security-Top-Source-Details	Top Source with Details by Bandwidth	traffic

```
select
  f_user,
  string_agg(
    distinct srcintf,
    & #039;,') as srcintf, string_agg(distinct dev_src, ',') as dev_src, sum(threatwgt) as
threatweight, sum(threat_block) as threat_block, (sum(threatwgt)-sum(threat_block)) as threat_
pass, sum(bandwidth) as bandwidth, sum(sessions) as sessions from ###(select coalesce(nullifna
(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as f_user, srcintf, max(coalesce(srcname,
srcmac)) AS dev_src, sum(threatwgt) as threatwgt, sum(CASE WHEN (logflag&2>0) THEN threatwgt ELSE
0 END) AS threat_block, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as
bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) AS sessions from (select `user`,
unauthuser, srcip, srcintf, srcname, srcmac, threatweight_sum(threatwgts, threatcnts) as
threatwgt, sentdelta, sentbyte, rcvddelta, rcvdbyte, logflag  from $log-traffic where $filter and
(logflag&(1|32)>0)) t group by f_user, srcintf order by bandwidth desc)### t where f_user is not
null group by f_user order by bandwidth desc
```

Dataset Name	Description	Log Category
360-security-Top-Destination-Bandwidth-Timeline	Top Destination Bandwidth Timeline	traffic

```
select
  $fv_line_timescale(timestamp) as timestamp,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out
from
  ###(select dvid, $flex_timestamp as timestamp, epid, srcip, policyname, policyid, sum(coalesce
(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta,
rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum((CASE WHEN
(logflag&2>0) THEN 1 ELSE 0 END)) AS session_block, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END)
AS sessions from $log-traffic where $filter and (logflag&(1|32)>0) group by dvid, timestamp, epid,
srcip, policyname, policyid order by bandwidth desc)### t group by timestamp order by timestamp
```

Dataset Name	Description	Log Category
360-security-Top-Destination-Details	Top Destination with Details by Bandwidth	traffic

```
select
  dstip,
  count(distinct app_group) as app_num,
  sum(sessions) as sessions,
  sum(bandwidth) as bandwidth
from
  ###(select dstip, app_group_name(app) as app_group, sum(coalesce(sentdelta, sentbyte,
0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END)
AS sessions from $log-traffic where $filter and (logflag&(1|32)>0) group by dstip, app_group order
by bandwidth desc)### t1 where dstip is not null group by dstip order by bandwidth desc
```

Dataset Name	Description	Log Category
360-security-High-Risk-Application-By-Category	High risk application by category	traffic

```
select
  app_cat,
  count(distinct app) as total_num
from
  ###(select app_cat, app, count(*) as total_num from $log t1 inner join app_mdata t2 on
t1.appid=t2.id where $filter and risk>='4' and (logflag&1>0) group by app_cat, app order by total_
num desc)### t group by app_cat order by total_num desc
```

Dataset Name	Description	Log Category
360-security-Apprisk-Ctrl-High-Risk-Application-Behavioral	Application Behavioral Characteristics	traffic

```
select
  behavior,
  round(
    sum(total_num)* 100 / sum(
      sum(total_num)
    ) over (),
    2
  ) as percentage
from
  (
    (
      select
        timestamp,
        behavior,
```

```
        sum(total_num) as total_num
      from
        ###(select timestamp, (case when lower(appcat)='botnet' then 'malicious' when lower
(appcat)='remote.access' then 'tunneling' when lower(appcat) in ('storage.backup', 'video/audio')
then 'bandwidth-consuming' when lower(appcat)='p2p' then 'peer-to-peer' when lower(appcat)='proxy'
then 'proxy' end) as behavior, sum(sessions) as total_num from ###base(/*tag:rpt_base_t_bndwdth_
sess*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid, euid, appcat, apprisk,
coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service, sum(CASE
WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum
(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in from $log-traffic where $filter and (logflag&
(1|32)>0) group by timestamp, dvid, srcip, dstip, epid, euid, appcat, apprisk, user_src, service
/*SkipSTART*/order by bandwidth desc, sessions desc/*SkipEND*/)base### t where lower(appcat) in
('botnet', 'remote.access', 'storage.backup', 'video/audio', 'p2p', 'proxy') and apprisk in
('critical', 'high') group by timestamp, behavior order by total_num desc)### t group by
timestamp, behavior) union all (select timestamp, behavior, sum(total_num) as total_num from ###
(select $flex_timestamp as timestamp, 'malicious' as behavior,  count(*) as total_num from $log-
attack where $filter and (logflag&16>0) and severity in ('critical', 'high') group by timestamp,
behavior order by total_num desc)### t group by timestamp, behavior)) t group by behavior order by
percentage desc
```

Dataset Name	Description	Log Category
360-security-Top10-App-Category-Group-By-Bandwidth	Category breakdown of all applications, sorted by bandwidth	traffic

```
select
  appcat,
  count(distinct app) as app_num,
  count(distinct user_src) as user_num,
  sum(bandwidth) as bandwidth,
  sum(sessions) as num_session
from
  ###(select app, appcat, user_src, sum(bandwidth) as bandwidth, sum(sessions) as sessions from
###base(/*tag:rpt_base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid,
euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service,
appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions
from $log-traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by
timestamp, dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat, apprisk,
hostname order by bandwidth desc, sessions desc)base### t where nullifna(appcat) is not null and
appcat not in ('Not.Scanned', 'unscanned', 'unknown') group by app, appcat, user_src order by
bandwidth desc)### t where $filter-drilldown group by appcat order by bandwidth desc
```

Dataset Name	Description	Log Category
360-security-Applications-By-Bandwidth	Top Web Applications by Bandwidth	traffic

```
select
  risk as d_risk,
  t2.name,
  t2.app_cat,
  t2.technology,
  count(distinct f_user) as users,
  sum(bandwidth) as bandwidth,
  sum(num_session) as sessions
from
  ###(select appid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as f_user,
sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, count(*) as num_session from $log
where $filter and (logflag&1>0) and nullifna(app) is not null and service in ('tcp/80', 'tcp/443',
'HTTP', 'HTTPS', 'http', 'https') group by appid, f_user order by bandwidth desc)### t1 inner join
app_mdata t2 on t1.appid=t2.id group by d_risk, t2.name, t2.app_cat, t2.technology order by d_risk
desc, bandwidth desc
```

Dataset Name	Description	Log Category
360-security-Top-Web-Categories-Visited	Top Web Category and User by Count	traffic

```
select
  catdesc,
  coalesce(
    firstname || & #039; ' || lastname, euname, usersrc) as user_src, requests, sum(requests) over
(partition by catdesc) as total_num from ###(select $flex_timestamp as timestamp, coalesce
(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as usersrc, euid, catdesc, hostname as
website, ebtr_agg_flat($browse_time) as browsetime, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte,
0)) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_
out, count(*) as requests from $log-traffic where $filter and (logflag&1>0) and (countweb>0 or
((logver is null or logver<502000000) and (hostname is not null or utmevent in ('webfilter',
'banned-word', 'web-content', 'command-block', 'script-filter')))) group by timestamp, usersrc,
euid, catdesc,website /*SkipSTART*/order by bandwidth desc/*SkipEND*/)### t1 left join $ADOM_
ENDUSER t3 on t1.euid=t3.euid where usersrc is not null and catdesc<>'Unknown' order by total_num
desc, catdesc
```

Dataset Name	Description	Log Category
360-security-Top5-Malware-Virus-Botnet-Spyware	Top Virus Botnet Spyware Adware and Phishing Websites	traffic

```
select
  malware_type,
  virus_s,
  total_num,
  sum(total_num) over (partition by malware_type) as type_total_num
from
  (
    (
```

```
      select
        (
          case when lower(appcat)=& #039;botnet' then 'Botnet C&C' else (case when virus_s like
'Riskware%' then 'Spyware' when virus_s like 'Adware%' then 'Adware' else 'Virus' end) end) as
malware_type, virus_s, sum(total_num) as total_num from ((select virus_s, appcat, hostname, sum
(total_num) as total_num from ###(select app as virus_s, appcat, hostname, count(*) as total_num
from $log-traffic where $filter and (logflag&1>0) and lower(appcat)='botnet' group by virus_s,
appcat, hostname order by total_num desc)### t group by virus_s, appcat, hostname) union all
(select virus_s, appcat, hostname, sum(total_num) as total_num from ###(select unnest(string_to_
array(virus, ',')) as virus_s, appcat, hostname, count(*) as total_num from $log-traffic where
$filter and (logflag&1>0) and virus is not null group by virus_s, appcat, hostname order by total_
num desc)### t group by virus_s, appcat, hostname) union all (select virus_s, appcat, hostname,
sum(total_num) as total_num from ###(select attack as virus_s, 'botnet' as appcat, hostname, count
(*) as total_num from $log-attack where $filter and (logflag&16>0) group by virus_s, appcat,
hostname order by total_num desc)### t group by virus_s, appcat, hostname)) t where virus_s is not
null group by malware_type, virus_s) union all (select malware_type, virus_s, sum(total_num) as
total_num from ###(select 'Phishing' as malware_type, hostname as virus_s, count(*) as total_num
from $log-webfilter where $filter and hostname is not null and catdesc='Phishing' group by
malware_type, virus_s)### t group by malware_type, virus_s)) t order by type_total_num desc,
virus_s
```

Dataset Name	Description	Log Category
360-security-Top5-Victims-of-Malware	Victims of Malware	virus

```
select
  coalesce(
    nullifna(`user`),
    nullifna(`unauthuser`),
    ipstr(`srcip`)
  ) as user_src,
  virus as malware,
  count(*) as total_num
from
  $log
where
  $filter
  and virus is not null
group by
  user_src,
  malware
order by
  total_num desc
```

Dataset Name	Description	Log Category
360-security-Top5-Victims-of-Phishing-Site	Victims of Phishing Site	webfilter

```
select
  coalesce(
    nullifna(`user`),
    nullifna(`unauthuser`),
    ipstr(`srcip`)
  ) as user_src,
  url as phishing_site,
  count(*) as total_num
from
  $log
where
  $filter
  and cat in (26, 61)
group by
  user_src,
  phishing_site
order by
  total_num desc
```

Dataset Name	Description	Log Category
360-security-Top5-Malicious-Phishing-Sites	Victims of Phishing Site by Count	webfilter

```
select
  phishing_site,
  user_src,
  total_num,
  sum(total_num) over (partition by phishing_site) as user_total_num
from
  ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
hostname as phishing_site, count(*) as total_num from $log where $filter and lower(service) in
('http', 'https') and hostname is not null and cat in (26, 61) group by user_src, phishing_site
order by total_num desc)### t order by user_total_num desc, user_src
```

Dataset Name	Description	Log Category
360-security-Application-Vulnerability	Application vulnerabilities discovered	attack

```
select
  attack,
  attackid,
  vuln_type,
  cve,
  severity_number,
  count(
    distinct (
      CASE WHEN direction =& #039;incoming' THEN srcip ELSE dstip END)) as victims, count(distinct
```

```
(CASE WHEN direction='incoming' THEN dstip ELSE srcip END)) as sources, sum(totalnum) as totalnum
from ###(select attack, attackid, (case when severity='critical' then 5 when severity='high' then
4 when severity='medium' then 3 when severity='low' then 2 when severity='info' then 1 else 0 end)
as severity_number, direction, dstip, srcip, count(*) as totalnum from $log where $filter and
nullifna(attack) is not null and severity is not null group by attack, attackid, severity,
direction, dstip, srcip order by totalnum desc)### t1 left join (select name, cve, vuln_type from
ips_mdata) t2 on t1.attack=t2.name group by attack, attackid, vuln_type, severity_number, cve
order by severity_number desc, totalnum desc
```

Dataset Name	Description	Log Category
360-security-Files-Analyzed-By-FortiCloud-Sandbox	Files analyzed by FortiCloud Sandbox	virus

```
select
  $day_of_week as dow,
  count(*) as total_num
from
  $log
where
  $filter
  and nullifna(filename) is not null
  and logid_to_int(logid)= 9233
group by
  dow
order by
  dow
```

Dataset Name	Description	Log Category
360-security-Apprisk-Ctrl-Malicious-Files-Detected-By-FortiCloud-Sandbox	Files detected by FortiCloud Sandbox	virus

```
select
  filename,
  analyticscksum,
  count(distinct victim) as victims,
  count(distinct source) as source
from
  ###(select filename, analyticscksum,(CASE WHEN direction='incoming' THEN dstip ELSE srcip END)
as source, (CASE WHEN direction='incoming' THEN srcip ELSE dstip END) as victim, count(*) as
totalnum from $log where $filter and filename is not null and logid_to_int(logid)=9233 and
analyticscksum is not null group by filename, analyticscksum, source, victim order by totalnum
desc)### t group by filename, analyticscksum order by victims desc, source desc
```

Dataset Name	Description	Log Category
360-security-Data-Loss-Incidents-By-Severity	Data loss incidents summary by severity	dlp

```
select
  initcap(severity : :text) as s_severity,
  count(*) as total_num
from
  ###(select itime, hostname,`from` as sender, `to` as receiver, profile, action, service,
subtype, srcip, dstip, severity, filename, direction, filesize, (case when severity='critical'
then 'Critical Data Exfiltration' else (case when coalesce(nullifna(`user`), ipstr(`srcip`)) is
not null then 'User Associated Data Loss' else NULL end) end) as data_loss from $log where $filter
/*SkipSTART*/order by itime desc/*SkipEND*/)### t where $filter-drilldown and severity is not null
group by s_severity order by total_num desc
```

Dataset Name	Description	Log Category
360-security-Data-Loss-Files-By-Service	Data Lass Files By Service	dlp

```
select
  filename,
  (
    case direction when & #039;incoming' then 'Download' when 'outgoing' then 'Upload' end) as
action, max(filesize) as filesize, service from  ###(select itime, hostname,`from` as sender, `to`
as receiver, profile, action, service, subtype, srcip, dstip, severity, filename, direction,
filesize, (case when severity='critical' then 'Critical Data Exfiltration' else (case when
coalesce(nullifna(`user`), ipstr(`srcip`)) is not null then 'User Associated Data Loss' else NULL
end) end) as data_loss from $log where $filter /*SkipSTART*/order by itime desc/*SkipEND*/)### t
where $filter-drilldown and filesize is not null group by filename, direction, service order by
filesize desc
```

Dataset Name	Description	Log Category
360-security-Endpoint-Security-Events-Summary	Endpoint Security Events summary	fct-traffic

```
select
  (
    case utmevent when & #039;antivirus' then 'Malware incidents' when 'webfilter' then
'Malicious/phishing websites' when 'appfirewall' then 'Risk applications' when 'dlp' then 'Data
loss incidents' when 'netscan' then 'Vulnerability detected' else 'Others' end) as events, count
(*) as total_num from $log where $filter and utmevent is not null group by events order by total_
num desc
```

Dataset Name	Description	Log Category
360-security-Top-Endpoing-Running-High-Risk-Application	Endpoints Running High Risk Application	fct-traffic

```
select
  coalesce(
    nullifna(`user`),
    ipstr(`srcip`),
    & #039;Unknown') as f_user, coalesce(nullifna(hostname), 'Unknown') as host_name, threat as
app, t2.app_cat as appcat, risk as d_risk from $log t1 inner join app_mdata t2 on
t1.threat=t2.name where $filter and utmevent='appfirewall' and risk>='4' group by f_user, host_
name, t1.threat, t2.app_cat, t2.risk order by risk desc
```

Dataset Name	Description	Log Category
soc-Total-Event-by-Severity	Total Events by Severity	

```
select
  sev,
  sum(num_events) as num_events
from

  /*fabricStart*/
  (
    select
      (
        CASE severity WHEN 0 THEN & #039;Critical' WHEN 1 THEN 'High' WHEN 2 THEN 'Medium' WHEN 3
THEN 'Low' ELSE NULL END) as sev, count(*) as num_events from $event t1 left join devtable_ext t2
on t1.dvid=t2.dvid where $cust_time_filter(alerttime) and $filter-drilldown group by severity
order by severity desc)/*fabricEnd*/ t group by sev order by sev desc
```

Dataset Name	Description	Log Category
soc-summary-Total-Event-by-Severity-Category	Total Events Count by Severity and Category	

```
select
  sev,
  triggername,
  sum(num_events) as num_events
from

  /*fabricStart*/
  (
    select
      (
        CASE severity WHEN 0 THEN & #039;Critical' WHEN 1 THEN 'High' WHEN 2 THEN 'Medium' WHEN 3
```

```
THEN 'Low' ELSE NULL END) as sev, triggername, count(*) as num_events from $event t1 left join
devtable_ext t2 on t1.dvid=t2.dvid where $cust_time_filter(alerttime) and $filter-drilldown group
by severity, triggername order by severity desc, triggername)/*fabricEnd*/ t group by sev,
triggername order by sev desc, triggername
```

Dataset Name	Description	Log Category
soc-summary-Affected-Endpoint-by-HWOS	Affected Endpoint Count by OS	

```
select
  osname,
  sum(count) as count
from

  /*fabricStart*/
  (
    select
      (
        case when osname is null then & #039;N/A' else osname end) as osname, count(distinct
(endpoint)) as count from $incident t1 inner join $ADOM_ENDPOINT t2 on t1.epid=t2.epid where
$cust_time_filter(createtime) and t2.epid>1024 group by osname order by count desc)/*fabricEnd*/ t
group by osname order by count desc
```

Dataset Name	Description	Log Category
soc-summary-Incident-by-Category	Incident Count by Category	

```
select
  cat,
  sum(num_cat) as num_cat
from

  /*fabricStart*/
  (
    select
      inc_cat_encode(category) as cat,
      count(*) as num_cat
    from
      $incident
    where
      $cust_time_filter(createtime)
    group by
      cat
    order by
      num_cat desc
  )
  /*fabricEnd*/
  t
```

```
group by
  cat
order by
  num_cat desc
```

Dataset Name	Description	Log Category
soc-summary-Incident-by-Status	Incidents by Status	

```
select
  status,
  sum(incnum) as incnum
from

  /*fabricStart*/
  (
    select
      status,
      count(*) as incnum
    from
      $incident
    where
      $cust_time_filter(createtime)
    group by
      status
    order by
      incnum desc
  )
  /*fabricEnd*/
  t
group by
  status
order by
  incnum desc
```

Dataset Name	Description	Log Category
soc-Incident-List	List of Incidents	

```
select
  incnum,
  timestamp,
  category,
  severity,
  status,
  endpoint
from

  /*fabricStart*/
```

```
  (
    select
      incid_to_str(incid) as incnum,
      from_itime(createtime) as timestamp,
      inc_cat_encode(category) as category,
      severity,
      status,
      endpoint
    from
      $incident
    where
      $cust_time_filter(createtime)
    order by
      createtime desc
  )
  /*fabricEnd*/
  t
order by
  timestamp desc
```

Dataset Name	Description	Log Category
Apprisk-Ctrl-Severe-High-Risk-Application	Severe and high risk applications	traffic

```
select
  appcat,
  count(distinct app) as total_num
from
  ###(select appid, app, appcat, apprisk, sum(bandwidth) as bandwidth, sum(sessions) as sessions
from ###base(/*tag:rpt_base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip,
epid, euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
service, appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_
in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte,
0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END)
as sessions from $log-traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null
group by timestamp, dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat,
apprisk, hostname order by bandwidth desc, sessions desc)base### t group by appid, app, appcat,
apprisk /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t where $filter-
drilldown and nullifna(appcat) is not null and apprisk in ('critical', 'high') group by appcat
order by total_num desc
```

Dataset Name	Description	Log Category
Apprisk-Ctrl-Threats-Prevention	Threat Prevention	app-ctrl

```
select
  threat_name,
  count(distinct threats) as total_num
```

```
from
  (
    (
      select
        threat_name,
        threats,
        sum(total_num) as total_num
      from
        ###(select cast('Malware & Botnet C&C' as char(32)) as threat_name, app as threats, count
(*) as total_num from $log-app-ctrl where $filter and lower(appcat)='botnet' group by app order by
total_num desc)### t group by threat_name, threats) union all (select threat_name, threats, sum
(total_num) as total_num from ###(select cast('Malware & Botnet C&C' as char(32)) as threat_name,
virus as threats, count(*) as total_num from $log-virus where $filter and nullifna(virus) is not
null group by virus order by total_num desc)### t group by threat_name, threats) union all (select
threat_name, threats, sum(total_num) as total_num from ###(select cast('Malicious & Phishing
Sites' as char(32)) as threat_name, hostname as threats, count(*) as total_num from $log-webfilter
where $filter and cat in (26, 61) group by hostname order by total_num desc)### t group by threat_
name, threats) union all (select threat_name, threats, sum(total_num) as total_num from ###(select
cast('Critical & High Intrusion Attacks' as char(32)) as threat_name, attack as threats, count(*)
as total_num from $log-attack where $filter and severity in ('critical', 'high') group by attack
order by total_num desc)### t group by threat_name, threats)) t group by threat_name order by
total_num desc
```

Dataset Name	Description	Log Category
Apprisk-Ctrl-Top-High-Risk-Application	Application risk high risk application	traffic

```
select
  risk as d_risk,
  count(distinct user_src) as users,
  id,
  name,
  app_cat,
  technology,
  sum(bandwidth) as bandwidth,
  sum(sessions) as sessions
from
  ###(select app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
action, utmaction, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, count(*) as
sessions from $log where $filter and (logflag&1>0) group by app, user_src, action, utmaction order
by bandwidth desc)### t1 inner join app_mdata t2 on t1.app=t2.name where risk>='4' group by id,
name, app_cat, technology, risk order by d_risk desc, sessions desc
```

Dataset Name	Description	Log Category
Apprisk-Ctrl-High-Risk-Application-Behavioral-Pie-Chart	Application Behavioral Characteristics	traffic

```
select
  behavior,
  round(
    sum(total_num)* 100 / sum(
      sum(total_num)
    ) over (),
    2
  ) as percentage
from
  (
    (
      select
        timestamp,
        behavior,
        sum(total_num) as total_num
      from
        ###(select timestamp, (case when lower(appcat)='botnet' then 'malicious' when lower
(appcat)='remote.access' then 'tunneling' when lower(appcat) in ('storage.backup', 'video/audio')
then 'bandwidth-consuming' when lower(appcat)='p2p' then 'peer-to-peer' when lower(appcat)='proxy'
then 'proxy' end) as behavior, sum(sessions) as total_num from ###base(/*tag:rpt_base_t_bndwdth_
sess*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid, euid, appcat, apprisk,
coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service, sum(CASE
WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum
(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in from $log-traffic where $filter and (logflag&
(1|32)>0) group by timestamp, dvid, srcip, dstip, epid, euid, appcat, apprisk, user_src, service
/*SkipSTART*/order by bandwidth desc, sessions desc/*SkipEND*/)base### t where lower(appcat) in
('botnet', 'remote.access', 'storage.backup', 'video/audio', 'p2p', 'proxy') and apprisk in
('critical', 'high') group by timestamp, behavior order by total_num desc)### t group by
timestamp, behavior) union all (select timestamp, behavior, sum(total_num) as total_num from ###
(select $flex_timestamp as timestamp, 'malicious' as behavior,  count(*) as total_num from $log-
attack where $filter and (logflag&16>0) and severity in ('critical', 'high') group by timestamp,
behavior order by total_num desc)### t group by timestamp, behavior)) t group by behavior order by
percentage desc
```

Dataset Name	Description	Log Category
Apprisk-Ctrl-High-Risk-Apps-Behavioral-Timeline	Application Behavioral Timeline	traffic

```
select
  $flex_timescale(timestamp) as hodex,
  behavior,
  sum(total_num) as total_num
from
  (
    (
      select
        timestamp,
        behavior,
        sum(total_num) as total_num
```

```
      from
        ###(select timestamp, (case when lower(appcat)='botnet' then 'malicious' when lower
(appcat)='remote.access' then 'tunneling' when lower(appcat) in ('storage.backup', 'video/audio')
then 'bandwidth-consuming' when lower(appcat)='p2p' then 'peer-to-peer' when lower(appcat)='proxy'
then 'proxy' end) as behavior, sum(sessions) as total_num from ###base(/*tag:rpt_base_t_bndwdth_
sess*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid, euid, appcat, apprisk,
coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service, sum(CASE
WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum
(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in from $log-traffic where $filter and (logflag&
(1|32)>0) group by timestamp, dvid, srcip, dstip, epid, euid, appcat, apprisk, user_src, service
/*SkipSTART*/order by bandwidth desc, sessions desc/*SkipEND*/)base### t where lower(appcat) in
('botnet', 'remote.access', 'storage.backup', 'video/audio', 'p2p', 'proxy') and apprisk in
('critical', 'high') group by timestamp, behavior order by total_num desc)### t group by
timestamp, behavior) union all (select timestamp, behavior, sum(total_num) as total_num from ###
(select $flex_timestamp as timestamp, 'malicious' as behavior,  count(*) as total_num from $log-
attack where $filter and (logflag&16>0) and severity in ('critical', 'high') group by timestamp,
behavior order by total_num desc)### t group by timestamp, behavior)) t group by hodex, behavior
order by total_num desc
```

Dataset Name	Description	Log Category
Apprisk-Ctrl-Top-High-Risk-Application-By-Bandwidth	High Risk Applications by Bandwidth	traffic

```
select
  risk as d_risk,
  count(distinct user_src) as users,
  id,
  name,
  app_cat,
  technology,
  sum(bandwidth) as bandwidth,
  sum(sessions) as sessions
from
  ###(select app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
action, utmaction, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, count(*) as
sessions from $log where $filter and (logflag&1>0) group by app, user_src, action, utmaction order
by bandwidth desc)### t1 inner join app_mdata t2 on t1.app=t2.name where risk>='4' group by id,
name, app_cat, technology, risk order by d_risk desc, bandwidth desc
```

Dataset Name	Description	Log Category
Apprisk-Ctrl-Top-Web-Applications	Top 25 Web Applications by Bandwidth	traffic

```
select
  risk as d_risk,
  id,
  name,
```

```
  technology,
  count(distinct user_src) as user_num,
  sum(bandwidth) as bandwidth,
  sum(num_session) as num_session
from
  ###(select appid, user_src, sum(bandwidth) as bandwidth, sum(sessions) as num_session from
###base(/*tag:rpt_base_t_top_app*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid,
euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service,
appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum
(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions
from $log-traffic where $filter and (logflag&(1|32)>0) and nullifna(app) is not null group by
timestamp, dvid, srcip, dstip, epid, euid, user_src, service, appid, app, appcat, apprisk,
hostname order by bandwidth desc, sessions desc)base### t where nullifna(app) is not null and
service in ('tcp/80', 'tcp/443', 'HTTP', 'HTTPS', 'http', 'https') group by appid, user_src order
by bandwidth desc)### t1 inner join app_mdata t2 on t1.appid=t2.id group by d_risk, id, name,
technology order by bandwidth desc
```

Dataset Name	Description	Log Category
Apprisk-Ctrl-Top-Visited-Web-Categories	Top 25 Web Categories Visited	traffic

```
select
  catdesc,
  count(distinct f_user) as user_num,
  sum(sessions) as sessions,
  sum(bandwidth) as bandwidth
from
  ###(select catdesc, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as f_
user, count(*) as sessions, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from
$log-traffic where $filter and catdesc is not null and (logflag&1>0) and (countweb>0 or ((logver
is null or logver<502000000) and (hostname is not null or utmevent in ('webfilter', 'banned-word',
'web-content', 'command-block', 'script-filter')))) group by f_user, catdesc order by sessions
desc)### t group by catdesc order by sessions desc
```

Dataset Name	Description	Log Category
Apprisk-Ctrl-Top-Application-Vulnerability	Application vulnerabilities discovered	attack

```
select
  attack,
  attackid,
  vuln_type,
  cve,
  severity_number,
  count(
    distinct (
```

```
      CASE WHEN direction =& #039;incoming' THEN srcip ELSE dstip END)) as victims, count(distinct
(CASE WHEN direction='incoming' THEN dstip ELSE srcip END)) as sources, sum(totalnum) as totalnum
from ###(select attack, attackid, (case when severity='critical' then 5 when severity='high' then
4 when severity='medium' then 3 when severity='low' then 2 when severity='info' then 1 else 0 end)
as severity_number, direction, dstip, srcip, count(*) as totalnum from $log where $filter and
nullifna(attack) is not null and severity is not null group by attack, attackid, severity,
direction, dstip, srcip order by totalnum desc)### t1 left join (select name, cve, vuln_type from
ips_mdata) t2 on t1.attack=t2.name group by attack, attackid, vuln_type, severity_number, cve
order by severity_number desc, totalnum desc
```

Dataset Name	Description	Log Category
Apprisk-Ctrl-Files-FortiCloud-Sandbox-Analyzed	Files FortiCloud Sandbox Analyzed	virus

```
select
  $fv_line_timescale(timestamp) as dom,
  sum(total_num) as total_num
from
  ###(select $flex_timestamp as timestamp, count(*) as total_num from $log where $filter and
nullifna(filename) is not null and logid_to_int(logid)=9233 group by timestamp order by total_num
desc)### t group by dom order by dom
```

Dataset Name	Description	Log Category
Apprisk-Ctrl-Malicious-Files-Detected-By-FortiCloud-Sandbox	Files detected by FortiCloud Sandbox	virus

```
select
  filename,
  analyticscksum,
  count(distinct victim) as victims,
  count(distinct source) as source
from
  ###(select filename, analyticscksum,(CASE WHEN direction='incoming' THEN dstip ELSE srcip END)
as source, (CASE WHEN direction='incoming' THEN srcip ELSE dstip END) as victim, count(*) as
totalnum from $log where $filter and filename is not null and logid_to_int(logid)=9233 and
analyticscksum is not null group by filename, analyticscksum, source, victim order by totalnum
desc)### t group by filename, analyticscksum order by victims desc, source desc
```

Dataset Name	Description	Log Category
Apprisk-Ctrl-High-Risk-Category-App-by-Bandwidth	High Risk Applications and Categories by Bandwidth	traffic

```
select
  app_cat,
  name,
```

```
  bandwidth,
  sum(bandwidth) over (partition by app_cat) as sub_bandwidth
from
  (
    select
      app_cat,
      name,
      sum(bandwidth) as bandwidth
    from
      ###(select app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_
src, action, utmaction, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, count(*) as
sessions from $log where $filter and (logflag&1>0) group by app, user_src, action, utmaction order
by bandwidth desc)### t1 inner join app_mdata t2 on t1.app=t2.name where risk>='4' group by app_
cat, name order by bandwidth desc) t order by sub_bandwidth desc, app_cat
```

Dataset Name	Description	Log Category
Apprisk-Ctrl-Malware-Virus-Botnet-Spyware-by-Count	Malware: Viruses, Bots, Spyware/Adware by Count	traffic

```
select
  malware_type,
  virus,
  totalnum,
  sum(totalnum) over (partition by malware_type) as sub_totalnum
from
  (
    select
      (
        case when lower(appcat)=& #039;botnet' then 'Botnet C&C' else (case when virus_s like
'Riskware%' then 'Spyware' when virus_s like 'Adware%' then 'Adware' else 'Virus' end) end) as
malware_type, virus_s as virus, sum(total_num) as totalnum from ((select virus_s, appcat, dstip,
srcip, sum(total_num) as total_num from ###(select app as virus_s, appcat, dstip, srcip, count(*)
as total_num from $log-traffic where $filter and (logflag&1>0) and lower(appcat)='botnet' group by
virus_s, appcat, dstip, srcip order by total_num desc)### t group by virus_s, appcat, dstip,
srcip) union all (select virus_s, appcat, dstip, srcip, sum(total_num) as total_num from ###
(select unnest(string_to_array(virus, ',')) as virus_s, appcat, dstip, srcip, count(*) as total_
num from $log-traffic where $filter and (logflag&1>0) and virus is not null group by virus_s,
appcat, dstip, srcip order by total_num desc)### t group by virus_s, appcat, dstip, srcip) union
all (select virus_s, appcat, dstip, srcip, sum(total_num) as total_num from ###(select attack as
virus_s, 'null' as appcat, dstip, srcip, count(*) as total_num from $log-attack where $filter and
(logflag&16>0) group by virus_s, appcat, dstip, srcip order by total_num desc)### t group by
virus_s, appcat, dstip, srcip)) t group by malware_type, virus order by totalnum desc ) t order by
sub_totalnum desc, malware_type
```

Dataset Name	Description	Log Category
security-Rating-Audit-Entry-Fail-List	Security Rating Audit Entry Fail List	

```
select
  audit_entry,
  max(compliance) as compliance,
  sum(count) as count,
  rtrim(
    to_char(
      sum(score),
      & #039;FM99999999D999'), '.') as score from /*fabricStart*/(select audit_entry, failed as
count, compliance, score from (select audit_entry, max(compliance) as compliance, sum(case when
result='passed' then 1 else 0 end) as passed, sum(case when result='exempt' then 1 else 0 end) as
exempt, sum(case when result='failed' then 1 else 0 end) as failed, sum(case when
result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score from (select audit_
entry, devid, result, sum(score) as score, '(' || string_agg(distinct compliance, ',') || ')' as
compliance from (select audit_entry, (instance->>'score')::float as score, instance->>'deviceID'
as devid, instance->>'device' as devtype, (case when instance->'domain'->>'type' in ('adom-
global', 'global') then 'Global' when instance->'domain'->'id' is not null then instance-
>'domain'->>'id' else 'Device' end) as scope, instance->'recommendation' #>> '{}' as
recommendation, instance->>'result' as result, (case instance->>'result' when 'passed' then 'none'
when 'exempt' then 'low' else instance->>'severity' end) as severity, '$security-standard ' ||
compliance as compliance from (select td.*, reporttype, audit_entry, compliance, instance from
(select dvid, reporttype, json_build_object('name', audit_entry, 'desc', audit_result-
>>'description')::text as audit_entry, json_array_elements_text(audit_result->'$security-
standard') as compliance, json_array_elements(audit_result->'instances') as instance from (select
dvid, reporttype, (json_each((data->'results')::json)).key as audit_entry, (json_each((data-
>'results')::json)).value as audit_result from (select distinct on(dvid, reporttype) dvid,
reporttype, data from $ADOMTBL_PLHD_AUDIT_HST t where $cust_time_filter(itime) order by dvid,
reporttype, itime desc) t) t where audit_result->'$security-standard' is not NULL) t inner join
devtable_ext td on td.dvid = t.dvid) t where $filter-drilldown) t group by audit_entry, devid,
result) t group by audit_entry) t where failed>0 order by count desc, audit_entry)/*fabricEnd*/ t
group by audit_entry order by count desc, audit_entry
```

Dataset Name	Description	Log Category
security-Rating-Audit-Entry-Unmet-List	Security Rating Audit Entry Unmet List	

```
select
  audit_entry,
  unmet as count,
  compliance,
  rtrim(
    to_char(
      score,
      & #039;FM99999999D999'), '.') as score from /*fabricStart*/(select audit_entry, max
(compliance) as compliance, sum(case when result='passed' then 1 else 0 end) as passed, sum(case
when result='exempt' then 1 else 0 end) as exempt, sum(case when result='failed' then 1 else 0
end) as failed, sum(case when result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score)
as score from (select audit_entry, devid, result, sum(score) as score, '(' || string_agg(distinct
compliance, ',') || ')' as compliance from (select audit_entry, (instance->>'score')::float as
score, instance->>'deviceID' as devid, instance->>'device' as devtype, (case when instance-
>'domain'->>'type' in ('adom-global', 'global') then 'Global' when instance->'domain'->'id' is not
```

```
null then instance->'domain'->>'id' else 'Device' end) as scope, instance->'recommendation' #>> '
{}' as recommendation, instance->>'result' as result, (case instance->>'result' when 'passed' then
'none' when 'exempt' then 'low' else instance->>'severity' end) as severity, '$security-standard '
|| compliance as compliance from (select td.*, reporttype, audit_entry, compliance, instance from
(select dvid, reporttype, json_build_object('name', audit_entry, 'desc', audit_result-
>>'description')::text as audit_entry, json_array_elements_text(audit_result->'$security-
standard') as compliance, json_array_elements(audit_result->'instances') as instance from (select
dvid, reporttype, (json_each((data->'results')::json)).key as audit_entry, (json_each((data-
>'results')::json)).value as audit_result from (select distinct on(dvid, reporttype) dvid,
reporttype, data from $ADOMTBL_PLHD_AUDIT_HST t where $cust_time_filter(itime) order by dvid,
reporttype, itime desc) t) t where audit_result->'$security-standard' is not NULL) t inner join
devtable_ext td on td.dvid = t.dvid) t where $filter-drilldown) t group by audit_entry, devid,
result) t group by audit_entry)/*fabricEnd*/ t where failed=0 and unmet>0 order by count desc,
audit_entry
```

Dataset Name	Description	Log Category
security-Rating-Audit-Entry-Pass-List	Security Rating Audit Entry Pass List	

```
select
  audit_entry,
  passed as count,
  compliance,
  rtrim(
    to_char(
      score,
      & #039;FM99999999D999'), '.') as score from /*fabricStart*/(select audit_entry, max
(compliance) as compliance, sum(case when result='passed' then 1 else 0 end) as passed, sum(case
when result='exempt' then 1 else 0 end) as exempt, sum(case when result='failed' then 1 else 0
end) as failed, sum(case when result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score)
as score from (select audit_entry, devid, result, sum(score) as score, '(' || string_agg(distinct
compliance, ',') || ')' as compliance from (select audit_entry, (instance->>'score')::float as
score, instance->>'deviceID' as devid, instance->>'device' as devtype, (case when instance-
>'domain'->>'type' in ('adom-global', 'global') then 'Global' when instance->'domain'->'id' is not
null then instance->'domain'->>'id' else 'Device' end) as scope, instance->'recommendation' #>> '
{}' as recommendation, instance->>'result' as result, (case instance->>'result' when 'passed' then
'none' when 'exempt' then 'low' else instance->>'severity' end) as severity, '$security-standard '
|| compliance as compliance from (select td.*, reporttype, audit_entry, compliance, instance from
(select dvid, reporttype, json_build_object('name', audit_entry, 'desc', audit_result-
>>'description')::text as audit_entry, json_array_elements_text(audit_result->'$security-
standard') as compliance, json_array_elements(audit_result->'instances') as instance from (select
dvid, reporttype, (json_each((data->'results')::json)).key as audit_entry, (json_each((data-
>'results')::json)).value as audit_result from (select distinct on(dvid, reporttype) dvid,
reporttype, data from $ADOMTBL_PLHD_AUDIT_HST t where $cust_time_filter(itime) order by dvid,
reporttype, itime desc) t) t where audit_result->'$security-standard' is not NULL) t inner join
devtable_ext td on td.dvid = t.dvid) t where $filter-drilldown) t group by audit_entry, devid,
result) t group by audit_entry)/*fabricEnd*/ t where failed=0 and unmet=0 and exempt=0 and
passed>0 order by score desc, audit_entry
```

Dataset Name	Description	Log Category
security-Rating-Audit-Entry-Exempt-List	Security Rating Audit Entry Exempt List	

```
select
  audit_entry,
  exempt as count,
  compliance,
  rtrim(
    to_char(
      score,
      & #039;FM99999999D999'), '.') as score from /*fabricStart*/(select audit_entry, max
(compliance) as compliance, sum(case when result='passed' then 1 else 0 end) as passed, sum(case
when result='exempt' then 1 else 0 end) as exempt, sum(case when result='failed' then 1 else 0
end) as failed, sum(case when result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score)
as score from (select audit_entry, devid, result, sum(score) as score, '(' || string_agg(distinct
compliance, ',') || ')' as compliance from (select audit_entry, (instance->>'score')::float as
score, instance->>'deviceID' as devid, instance->>'device' as devtype, (case when instance-
>'domain'->>'type' in ('adom-global', 'global') then 'Global' when instance->'domain'->'id' is not
null then instance->'domain'->>'id' else 'Device' end) as scope, instance->'recommendation' #>> '
{}' as recommendation, instance->>'result' as result, (case instance->>'result' when 'passed' then
'none' when 'exempt' then 'low' else instance->>'severity' end) as severity, '$security-standard '
|| compliance as compliance from (select td.*, reporttype, audit_entry, compliance, instance from
(select dvid, reporttype, json_build_object('name', audit_entry, 'desc', audit_result-
>>'description')::text as audit_entry, json_array_elements_text(audit_result->'$security-
standard') as compliance, json_array_elements(audit_result->'instances') as instance from (select
dvid, reporttype, (json_each((data->'results')::json)).key as audit_entry, (json_each((data-
>'results')::json)).value as audit_result from (select distinct on(dvid, reporttype) dvid,
reporttype, data from $ADOMTBL_PLHD_AUDIT_HST t where $cust_time_filter(itime) order by dvid,
reporttype, itime desc) t) t where audit_result->'$security-standard' is not NULL) t inner join
devtable_ext td on td.dvid = t.dvid) t where $filter-drilldown) t group by audit_entry, devid,
result) t group by audit_entry)/*fabricEnd*/ t where failed=0 and unmet=0 and exempt>0 order by
score desc, audit_entry
```

Dataset Name	Description	Log Category
security-Rating-Stats-Status-Details	Security Rating Statistics	

```
select
  audit_entry,
  devtype,
  devid,
  scope,
  severity,
  rtrim(
    to_char(
      sum(score),
      & #039;FM99999999D999'), '.') AS score, result, string_agg(distinct (CASE WHEN compliance =
```

```
'None' THEN NULL ELSE compliance END), ',') AS compliance from /*fabricStart*/(select audit_entry,
(instance->>'score')::float as score, instance->>'deviceID' as devid, instance->>'device' as
devtype, (case when instance->'domain'->>'type' in ('adom-global', 'global') then 'Global' when
instance->'domain'->'id' is not null then instance->'domain'->>'id' else 'Device' end) as scope,
instance->'recommendation' #>> '{}' as recommendation, instance->>'result' as result, (case
instance->>'result' when 'passed' then 'none' when 'exempt' then 'low' else instance->>'severity'
end) as severity, '$security-standard ' || compliance as compliance from (select audit_entry,
instance, json_array_elements_text(compliances) as compliance from (select td.*, reporttype,
audit_entry, compliances, instance from (select dvid, reporttype, json_build_object('name', audit_
entry, 'desc', audit_result->>'description')::text as audit_entry, (case when json_array_length
(audit_result->'$security-standard') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-Stats-Recommendation	Security Rating Statistics Recommendation	

```
select
  audit_entry,
  devtype,
  devid,
  scope,
  severity,
  rtrim(
    to_char(
      score,
      & #039;FM99999999D999'), '.') as score, result, max(recommendation) as recommendation from
/*fabricStart*/(select audit_entry, (instance->>'score')::float as score, instance->>'deviceID' as
devid, instance->>'device' as devtype, (case when instance->'domain'->>'type' in ('adom-global',
'global') then 'Global' when instance->'domain'->'id' is not null then instance->'domain'->>'id'
else 'Device' end) as scope, instance->'recommendation' #>> '{}' as recommendation, instance-
>>'result' as result, (case instance->>'result' when 'passed' then 'none' when 'exempt' then 'low'
else instance->>'severity' end) as severity, '$security-standard ' || compliance as compliance
from (select audit_entry, instance, json_array_elements_text(compliances) as compliance from
(select td.*, reporttype, audit_entry, compliances, instance from (select dvid, reporttype, json_
build_object('name', audit_entry, 'desc', audit_result->>'description')::text as audit_entry,
(case when json_array_length(audit_result->'$security-standard') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-Stats-Status-Count	Security Rating Statistic Status Count	

```
select
  unnest(name) as stats,
  unnest(val) as value
from
  (
    select
      array[ & #039;Failed', 'Unmet', 'Passed','Exempt'] as name, array[ count(distinct (case when
```

```
failed>0 then audit_entry end)), count(distinct (case when failed=0 and unmet>0 then audit_entry
end)), count(distinct (case when failed=0 and unmet=0 and exempt=0 and passed>0 then audit_entry
end)), count(distinct (case when failed=0 and unmet=0 and exempt>0 then audit_entry end))] as val
from /*fabricStart*/(select audit_entry, max(compliance) as compliance, sum(case when
result='passed' then 1 else 0 end) as passed, sum(case when result='exempt' then 1 else 0 end) as
exempt, sum(case when result='failed' then 1 else 0 end) as failed, sum(case when
result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score from (select audit_
entry, devid, result, sum(score) as score, '(' || string_agg(distinct compliance, ',') || ')' as
compliance from (select audit_entry, (instance->>'score')::float as score, instance->>'deviceID'
as devid, instance->>'device' as devtype, (case when instance->'domain'->>'type' in ('adom-
global', 'global') then 'Global' when instance->'domain'->'id' is not null then instance-
>'domain'->>'id' else 'Device' end) as scope, instance->'recommendation' #>> '{}' as
recommendation, instance->>'result' as result, (case instance->>'result' when 'passed' then 'none'
when 'exempt' then 'low' else instance->>'severity' end) as severity, '$security-standard ' ||
compliance as compliance from (select td.*, reporttype, audit_entry, compliance, instance from
(select dvid, reporttype, json_build_object('name', audit_entry, 'desc', audit_result-
>>'description')::text as audit_entry, json_array_elements_text(audit_result->'$security-
standard') as compliance, json_array_elements(audit_result->'instances') as instance from (select
dvid, reporttype, (json_each((data->'results')::json)).key as audit_entry, (json_each((data-
>'results')::json)).value as audit_result from (select distinct on(dvid, reporttype) dvid,
reporttype, data from $ADOMTBL_PLHD_AUDIT_HST t where $cust_time_filter(itime) order by dvid,
reporttype, itime desc) t) t where audit_result->'$security-standard' is not NULL) t inner join
devtable_ext td on td.dvid = t.dvid) t where $filter-drilldown) t group by audit_entry, devid,
result) t group by audit_entry)/*fabricEnd*/ t) t
```

Dataset Name	Description	Log Category
security-Rating-CIS-Control-Result-Count	Security Rating CIS Control Result by Count	

```
select
  unnest(name) as stats,
  unnest(val) as value
from
  (
    select
      array[ & #039;Failed', 'Passed', 'Exempt', 'Unmet'] as name, array[ count(distinct (case
when failed>0 then cis_sub_control_id||devid||fsbp_id end)), count(distinct (case when failed=0
and unmet=0 and exempt=0 and passed>0 then cis_sub_control_id||devid||fsbp_id end)), count
(distinct (case when failed=0 and unmet=0 and exempt>0 then cis_sub_control_id||devid||fsbp_id
end)), count(distinct (case when failed=0 and unmet>0 then cis_sub_control_id||devid||fsbp_id
end)) ] as val from /*fabricStart*/(select devid, devtype, scope, result, severity, compliance,
cis, cis_sub, cis_sub_control_id, asset_type, title, fsbp_id, name, description, recommendation,
sum(case when result='passed' then 1 else 0 end) as passed, sum(case when result='exempt' then 1
else 0 end) as exempt, sum(case when result='failed' then 1 else 0 end) as failed, sum(case when
result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score from (select audit_
entry, (instance->>'score')::float as score, instance->>'deviceID' as devid, instance->>'device'
as devtype, (case when instance->'domain'->>'type' in ('adom-global', 'global') then 'Global' when
instance->'domain'->'id' is not null then instance->'domain'->>'id' else 'Device' end) as scope,
instance->'recommendation' #>> '{}' as recommendation, instance->>'result' as result, (case
instance->>'result' when 'passed' then 'none' when 'exempt' then 'low' else instance->>'severity'
```

```
end) as severity, compliance, cis, cis_sub, cis_sub_control_id, asset_type, title, fsbp_id, name,
description from (select cis::int as cis, cis_sub, cis_sub_control_id, t1.asset_type, t1.title,
fsbp_id, t2.name, t2.description from (select split_part(t1.id, '.', 1) as cis,  substr(t1.id,
strpos(t1.id, '.')+1) as cis_sub, t1.id as cis_sub_control_id, t2.asset_type, t2.name as title,
fsbp_id from security_cis_fsbp_map t1 inner join security_cis_mdata t2 on t1.id= t2.id) t1 inner
join security_cis_mdata t2 on t1.cis = t2.id order by cis, cis_sub, fsbp_id) t1 left join (select
td.*, reporttype, audit_entry, json_array_elements_text(compliances) as compliance, instance from
(select dvid, reporttype, json_build_object('name', audit_entry, 'desc', audit_result-
>>'description')::text as audit_entry, (case when json_array_length(audit_result->'FSBP') = 0 then
'[\
```

Dataset Name	Description	Log Category
security-Rating-CIS-Control-Compliance-Results	Security Rating CIS Control Compliance Results	

```
select
  & #039;CIS Controls '||unnest(name) as stats, unnest(val) as value from (select array['Failed',
'Passed'] as name, array[ count(distinct (case when failed>0 or (failed=0 and passed=0) then cis_
sub_control_id end)), count(distinct (case when failed = 0 and passed>0 then cis_sub_control_id
end)) ] as val from (select cis_sub_control_id, count(distinct (case when failed>0 or unmet>0 or
exempt>0 then devid end)) as failed, count(distinct (case when failed=0 and unmet=0 and exempt=0
and passed>0 then devid end)) as passed from (select cis_sub_control_id, devid, sum(failed) as
failed, sum(passed) as passed, sum(unmet) as unmet, sum(exempt) as exempt from /*fabricStart*/
(select devid, devtype, scope, result, severity, compliance, cis, cis_sub, cis_sub_control_id,
asset_type, title, fsbp_id, name, description, recommendation, sum(case when result='passed' then
1 else 0 end) as passed, sum(case when result='exempt' then 1 else 0 end) as exempt, sum(case when
result='failed' then 1 else 0 end) as failed, sum(case when result='dependenciesNotMet' then 1
else 0 end) as unmet, sum(score) as score from (select audit_entry, (instance->>'score')::float as
score, instance->>'deviceID' as devid, instance->>'device' as devtype, (case when instance-
>'domain'->>'type' in ('adom-global', 'global') then 'Global' when instance->'domain'->'id' is not
null then instance->'domain'->>'id' else 'Device' end) as scope, instance->'recommendation' #>> '
{}' as recommendation, instance->>'result' as result, (case instance->>'result' when 'passed' then
'none' when 'exempt' then 'low' else instance->>'severity' end) as severity, compliance, cis, cis_
sub, cis_sub_control_id, asset_type, title, fsbp_id, name, description from (select cis::int as
cis, cis_sub, cis_sub_control_id, t1.asset_type, t1.title, fsbp_id, t2.name, t2.description from
(select split_part(t1.id, '.', 1) as cis,  substr(t1.id, strpos(t1.id, '.')+1) as cis_sub, t1.id
as cis_sub_control_id, t2.asset_type, t2.name as title, fsbp_id from security_cis_fsbp_map t1
inner join security_cis_mdata t2 on t1.id= t2.id) t1 inner join security_cis_mdata t2 on t1.cis =
t2.id order by cis, cis_sub, fsbp_id) t1 left join (select td.*, reporttype, audit_entry, json_
array_elements_text(compliances) as compliance, instance from (select dvid, reporttype, json_
build_object('name', audit_entry, 'desc', audit_result->>'description')::text as audit_entry,
(case when json_array_length(audit_result->'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-CIS-Control-Overview	Security Rating CIS Control Overview	

```
select
  cis,
  (
    case when cis_sub is not null then & #039;CIS Control '||cis||'.'||cis_sub else 'CIS Control
'||cis end) as cis_control, title, passed, failed from (select cis, cis_sub, title, count(distinct
(case when failed=0 and unmet=0 and exempt=0 and passed>0 then devid end)) as passed, count
(distinct (case when failed>0 or unmet>0 or exempt>0 then devid end)) as failed from ((select cis,
cis_sub, devid, title, sum(failed) as failed, sum(passed) as passed, sum(unmet) as unmet, sum
(exempt) as exempt from /*fabricStart*/(select devid, devtype, scope, result, severity,
compliance, cis, cis_sub, cis_sub_control_id, asset_type, title, fsbp_id, name, description,
recommendation, sum(case when result='passed' then 1 else 0 end) as passed, sum(case when
result='exempt' then 1 else 0 end) as exempt, sum(case when result='failed' then 1 else 0 end) as
failed, sum(case when result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score
from (select audit_entry, (instance->>'score')::float as score, instance->>'deviceID' as devid,
instance->>'device' as devtype, (case when instance->'domain'->>'type' in ('adom-global',
'global') then 'Global' when instance->'domain'->'id' is not null then instance->'domain'->>'id'
else 'Device' end) as scope, instance->'recommendation' #>> '{}' as recommendation, instance-
>>'result' as result, (case instance->>'result' when 'passed' then 'none' when 'exempt' then 'low'
else instance->>'severity' end) as severity, compliance, cis, cis_sub, cis_sub_control_id, asset_
type, title, fsbp_id, name, description from (select cis::int as cis, cis_sub, cis_sub_control_id,
t1.asset_type, t1.title, fsbp_id, t2.name, t2.description from (select split_part(t1.id, '.', 1)
as cis,  substr(t1.id, strpos(t1.id, '.')+1) as cis_sub, t1.id as cis_sub_control_id, t2.asset_
type, t2.name as title, fsbp_id from security_cis_fsbp_map t1 inner join security_cis_mdata t2 on
t1.id= t2.id) t1 inner join security_cis_mdata t2 on t1.cis = t2.id order by cis, cis_sub, fsbp_
id) t1 left join (select td.*, reporttype, audit_entry, json_array_elements_text(compliances) as
compliance, instance from (select dvid, reporttype, json_build_object('name', audit_entry, 'desc',
audit_result->>'description')::text as audit_entry, (case when json_array_length(audit_result-
>'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-CIS-Control-Result-List	Security Rating CIS Control Result List	

```
select
  cis,
  name,
  description,
  count(
    distinct (
      case when failed>0 then cis_sub || devid || fsbp_id end
    )
  ) as total_failed,
  count(
    distinct (
      case when failed = 0
      and unmet>0 then cis_sub || devid || fsbp_id end
    )
  ) as total_unmet,
  count(
    distinct (
```

```
      case when failed = 0
      and unmet = 0
      and exempt = 0
      and passed>0 then cis_sub || devid || fsbp_id end
    )
  ) as total_passed,
  count(
    distinct (
      case when failed = 0
      and unmet = 0
      and exempt>0
      and passed = 0 then cis_sub || devid || fsbp_id end
    )
  ) as total_exempt
from

  /*fabricStart*/
  (
    select
      devid,
      devtype,
      scope,
      result,
      severity,
      compliance,
      cis,
      cis_sub,
      cis_sub_control_id,
      asset_type,
      title,
      fsbp_id,
      name,
      description,
      recommendation,
      sum(
        case when result =& #039;passed' then 1 else 0 end) as passed, sum(case when
result='exempt' then 1 else 0 end) as exempt, sum(case when result='failed' then 1 else 0 end) as
failed, sum(case when result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score
from (select audit_entry, (instance->>'score')::float as score, instance->>'deviceID' as devid,
instance->>'device' as devtype, (case when instance->'domain'->>'type' in ('adom-global',
'global') then 'Global' when instance->'domain'->'id' is not null then instance->'domain'->>'id'
else 'Device' end) as scope, instance->'recommendation' #>> '{}' as recommendation, instance-
>>'result' as result, (case instance->>'result' when 'passed' then 'none' when 'exempt' then 'low'
else instance->>'severity' end) as severity, compliance, cis, cis_sub, cis_sub_control_id, asset_
type, title, fsbp_id, name, description from (select cis::int as cis, cis_sub, cis_sub_control_id,
t1.asset_type, t1.title, fsbp_id, t2.name, t2.description from (select split_part(t1.id, '.', 1)
as cis,  substr(t1.id, strpos(t1.id, '.')+1) as cis_sub, t1.id as cis_sub_control_id, t2.asset_
type, t2.name as title, fsbp_id from security_cis_fsbp_map t1 inner join security_cis_mdata t2 on
t1.id= t2.id) t1 inner join security_cis_mdata t2 on t1.cis = t2.id order by cis, cis_sub, fsbp_
id) t1 left join (select td.*, reporttype, audit_entry, json_array_elements_text(compliances) as
compliance, instance from (select dvid, reporttype, json_build_object('name', audit_entry, 'desc',
audit_result->>'description')::text as audit_entry, (case when json_array_length(audit_result-
```

```
>'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-CIS-Sub-Control-Result-List	Security Rating CIS Sub Control Result List	

```
select
  cis_sub_control_id,
  cis,
  cis_sub,
  asset_type,
  title,
  count(distinct devid || fsbp_id) as total_num
from

  /*fabricStart*/
  (
    select
      devid,
      devtype,
      scope,
      result,
      severity,
      compliance,
      cis,
      cis_sub,
      cis_sub_control_id,
      asset_type,
      title,
      fsbp_id,
      name,
      description,
      recommendation,
      sum(
        case when result =& #039;passed' then 1 else 0 end) as passed, sum(case when
result='exempt' then 1 else 0 end) as exempt, sum(case when result='failed' then 1 else 0 end) as
failed, sum(case when result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score
from (select audit_entry, (instance->>'score')::float as score, instance->>'deviceID' as devid,
instance->>'device' as devtype, (case when instance->'domain'->>'type' in ('adom-global',
'global') then 'Global' when instance->'domain'->'id' is not null then instance->'domain'->>'id'
else 'Device' end) as scope, instance->'recommendation' #>> '{}' as recommendation, instance-
>>'result' as result, (case instance->>'result' when 'passed' then 'none' when 'exempt' then 'low'
else instance->>'severity' end) as severity, compliance, cis, cis_sub, cis_sub_control_id, asset_
type, title, fsbp_id, name, description from (select cis::int as cis, cis_sub, cis_sub_control_id,
t1.asset_type, t1.title, fsbp_id, t2.name, t2.description from (select split_part(t1.id, '.', 1)
as cis,  substr(t1.id, strpos(t1.id, '.')+1) as cis_sub, t1.id as cis_sub_control_id, t2.asset_
type, t2.name as title, fsbp_id from security_cis_fsbp_map t1 inner join security_cis_mdata t2 on
t1.id= t2.id) t1 inner join security_cis_mdata t2 on t1.cis = t2.id order by cis, cis_sub, fsbp_
id) t1 left join (select td.*, reporttype, audit_entry, json_array_elements_text(compliances) as
compliance, instance from (select dvid, reporttype, json_build_object('name', audit_entry, 'desc',
```

```
audit_result->>'description')::text as audit_entry, (case when json_array_length(audit_result-
>'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-CIS-Sub-Control-Failed-Result-List	Security Rating CIS Sub Control Fail List	

```
select
  title,
  count(distinct devid || fsbp_id) as total_num,
  string_agg(
    distinct compliance,
    & #039;,') as compliance, rtrim(to_char(sum(score), 'FM99999999D999'), '.') as score from
/*fabricStart*/(select devid, devtype, scope, result, severity, compliance, cis, cis_sub, cis_sub_
control_id, asset_type, title, fsbp_id, name, description, recommendation, sum(case when
result='passed' then 1 else 0 end) as passed, sum(case when result='exempt' then 1 else 0 end) as
exempt, sum(case when result='failed' then 1 else 0 end) as failed, sum(case when
result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score from (select audit_
entry, (instance->>'score')::float as score, instance->>'deviceID' as devid, instance->>'device'
as devtype, (case when instance->'domain'->>'type' in ('adom-global', 'global') then 'Global' when
instance->'domain'->'id' is not null then instance->'domain'->>'id' else 'Device' end) as scope,
instance->'recommendation' #>> '{}' as recommendation, instance->>'result' as result, (case
instance->>'result' when 'passed' then 'none' when 'exempt' then 'low' else instance->>'severity'
end) as severity, compliance, cis, cis_sub, cis_sub_control_id, asset_type, title, fsbp_id, name,
description from (select cis::int as cis, cis_sub, cis_sub_control_id, t1.asset_type, t1.title,
fsbp_id, t2.name, t2.description from (select split_part(t1.id, '.', 1) as cis,  substr(t1.id,
strpos(t1.id, '.')+1) as cis_sub, t1.id as cis_sub_control_id, t2.asset_type, t2.name as title,
fsbp_id from security_cis_fsbp_map t1 inner join security_cis_mdata t2 on t1.id= t2.id) t1 inner
join security_cis_mdata t2 on t1.cis = t2.id order by cis, cis_sub, fsbp_id) t1 left join (select
td.*, reporttype, audit_entry, json_array_elements_text(compliances) as compliance, instance from
(select dvid, reporttype, json_build_object('name', audit_entry, 'desc', audit_result-
>>'description')::text as audit_entry, (case when json_array_length(audit_result->'FSBP') = 0 then
'[\
```

Dataset Name	Description	Log Category
security-Rating-CIS-Sub-Control-Failed-Stats-Recommendation	Security Rating CIS Sub Control Failed Statistics Recommendation	

```
select
  cis,
  cis_sub,
  devtype,
  devid,
  compliance,
  severity,
  rtrim(
    to_char(
```

```
      sum(score),
      & #039;FM99999999D999'), '.') as score, result, recommendation from /*fabricStart*/(select
devid, devtype, scope, result, severity, compliance, cis, cis_sub, cis_sub_control_id, asset_type,
title, fsbp_id, name, description, recommendation, sum(case when result='passed' then 1 else 0
end) as passed, sum(case when result='exempt' then 1 else 0 end) as exempt, sum(case when
result='failed' then 1 else 0 end) as failed, sum(case when result='dependenciesNotMet' then 1
else 0 end) as unmet, sum(score) as score from (select audit_entry, (instance->>'score')::float as
score, instance->>'deviceID' as devid, instance->>'device' as devtype, (case when instance-
>'domain'->>'type' in ('adom-global', 'global') then 'Global' when instance->'domain'->'id' is not
null then instance->'domain'->>'id' else 'Device' end) as scope, instance->'recommendation' #>> '
{}' as recommendation, instance->>'result' as result, (case instance->>'result' when 'passed' then
'none' when 'exempt' then 'low' else instance->>'severity' end) as severity, compliance, cis, cis_
sub, cis_sub_control_id, asset_type, title, fsbp_id, name, description from (select cis::int as
cis, cis_sub, cis_sub_control_id, t1.asset_type, t1.title, fsbp_id, t2.name, t2.description from
(select split_part(t1.id, '.', 1) as cis,  substr(t1.id, strpos(t1.id, '.')+1) as cis_sub, t1.id
as cis_sub_control_id, t2.asset_type, t2.name as title, fsbp_id from security_cis_fsbp_map t1
inner join security_cis_mdata t2 on t1.id= t2.id) t1 inner join security_cis_mdata t2 on t1.cis =
t2.id order by cis, cis_sub, fsbp_id) t1 left join (select td.*, reporttype, audit_entry, json_
array_elements_text(compliances) as compliance, instance from (select dvid, reporttype, json_
build_object('name', audit_entry, 'desc', audit_result->>'description')::text as audit_entry,
(case when json_array_length(audit_result->'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-CIS-Sub-Control-Unmet-Result-List	Security Rating CIS Sub Control Unmet List	

```
select
  title,
  count(distinct devid || fsbp_id) as total_num,
  string_agg(
    distinct compliance,
    & #039;,') as compliance, rtrim(to_char(sum(score), 'FM99999999D999'), '.') as score from
/*fabricStart*/(select devid, devtype, scope, result, severity, compliance, cis, cis_sub, cis_sub_
control_id, asset_type, title, fsbp_id, name, description, recommendation, sum(case when
result='passed' then 1 else 0 end) as passed, sum(case when result='exempt' then 1 else 0 end) as
exempt, sum(case when result='failed' then 1 else 0 end) as failed, sum(case when
result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score from (select audit_
entry, (instance->>'score')::float as score, instance->>'deviceID' as devid, instance->>'device'
as devtype, (case when instance->'domain'->>'type' in ('adom-global', 'global') then 'Global' when
instance->'domain'->'id' is not null then instance->'domain'->>'id' else 'Device' end) as scope,
instance->'recommendation' #>> '{}' as recommendation, instance->>'result' as result, (case
instance->>'result' when 'passed' then 'none' when 'exempt' then 'low' else instance->>'severity'
end) as severity, compliance, cis, cis_sub, cis_sub_control_id, asset_type, title, fsbp_id, name,
description from (select cis::int as cis, cis_sub, cis_sub_control_id, t1.asset_type, t1.title,
fsbp_id, t2.name, t2.description from (select split_part(t1.id, '.', 1) as cis,  substr(t1.id,
strpos(t1.id, '.')+1) as cis_sub, t1.id as cis_sub_control_id, t2.asset_type, t2.name as title,
fsbp_id from security_cis_fsbp_map t1 inner join security_cis_mdata t2 on t1.id= t2.id) t1 inner
join security_cis_mdata t2 on t1.cis = t2.id order by cis, cis_sub, fsbp_id) t1 left join (select
td.*, reporttype, audit_entry, json_array_elements_text(compliances) as compliance, instance from
(select dvid, reporttype, json_build_object('name', audit_entry, 'desc', audit_result-
```

```
>>'description')::text as audit_entry, (case when json_array_length(audit_result->'FSBP') = 0 then
'[\
```

Dataset Name	Description	Log Category
security-Rating-CIS-Sub-Control-Unmet-Stats-Recommendation	Security Rating CIS Sub Control Unmet Statistics Recommendation	

```
select
  cis,
  cis_sub,
  devtype,
  devid,
  compliance,
  severity,
  rtrim(
    to_char(
      sum(score),
      & #039;FM99999999D999'), '.') as score, result, recommendation from /*fabricStart*/(select
devid, devtype, scope, result, severity, compliance, cis, cis_sub, cis_sub_control_id, asset_type,
title, fsbp_id, name, description, recommendation, sum(case when result='passed' then 1 else 0
end) as passed, sum(case when result='exempt' then 1 else 0 end) as exempt, sum(case when
result='failed' then 1 else 0 end) as failed, sum(case when result='dependenciesNotMet' then 1
else 0 end) as unmet, sum(score) as score from (select audit_entry, (instance->>'score')::float as
score, instance->>'deviceID' as devid, instance->>'device' as devtype, (case when instance-
>'domain'->>'type' in ('adom-global', 'global') then 'Global' when instance->'domain'->'id' is not
null then instance->'domain'->>'id' else 'Device' end) as scope, instance->'recommendation' #>> '
{}' as recommendation, instance->>'result' as result, (case instance->>'result' when 'passed' then
'none' when 'exempt' then 'low' else instance->>'severity' end) as severity, compliance, cis, cis_
sub, cis_sub_control_id, asset_type, title, fsbp_id, name, description from (select cis::int as
cis, cis_sub, cis_sub_control_id, t1.asset_type, t1.title, fsbp_id, t2.name, t2.description from
(select split_part(t1.id, '.', 1) as cis,  substr(t1.id, strpos(t1.id, '.')+1) as cis_sub, t1.id
as cis_sub_control_id, t2.asset_type, t2.name as title, fsbp_id from security_cis_fsbp_map t1
inner join security_cis_mdata t2 on t1.id= t2.id) t1 inner join security_cis_mdata t2 on t1.cis =
t2.id order by cis, cis_sub, fsbp_id) t1 left join (select td.*, reporttype, audit_entry, json_
array_elements_text(compliances) as compliance, instance from (select dvid, reporttype, json_
build_object('name', audit_entry, 'desc', audit_result->>'description')::text as audit_entry,
(case when json_array_length(audit_result->'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-CIS-Sub-Control-Passed-Result-List	Security Rating CIS Sub Control Passed List	

```
select
  title,
  count(distinct devid || fsbp_id) as total_num,
  string_agg(
    distinct compliance,
```

```
    & #039;,') as compliance, rtrim(to_char(sum(score), 'FM99999999D999'), '.') as score from
/*fabricStart*/(select devid, devtype, scope, result, severity, compliance, cis, cis_sub, cis_sub_
control_id, asset_type, title, fsbp_id, name, description, recommendation, sum(case when
result='passed' then 1 else 0 end) as passed, sum(case when result='exempt' then 1 else 0 end) as
exempt, sum(case when result='failed' then 1 else 0 end) as failed, sum(case when
result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score from (select audit_
entry, (instance->>'score')::float as score, instance->>'deviceID' as devid, instance->>'device'
as devtype, (case when instance->'domain'->>'type' in ('adom-global', 'global') then 'Global' when
instance->'domain'->'id' is not null then instance->'domain'->>'id' else 'Device' end) as scope,
instance->'recommendation' #>> '{}' as recommendation, instance->>'result' as result, (case
instance->>'result' when 'passed' then 'none' when 'exempt' then 'low' else instance->>'severity'
end) as severity, compliance, cis, cis_sub, cis_sub_control_id, asset_type, title, fsbp_id, name,
description from (select cis::int as cis, cis_sub, cis_sub_control_id, t1.asset_type, t1.title,
fsbp_id, t2.name, t2.description from (select split_part(t1.id, '.', 1) as cis,  substr(t1.id,
strpos(t1.id, '.')+1) as cis_sub, t1.id as cis_sub_control_id, t2.asset_type, t2.name as title,
fsbp_id from security_cis_fsbp_map t1 inner join security_cis_mdata t2 on t1.id= t2.id) t1 inner
join security_cis_mdata t2 on t1.cis = t2.id order by cis, cis_sub, fsbp_id) t1 left join (select
td.*, reporttype, audit_entry, json_array_elements_text(compliances) as compliance, instance from
(select dvid, reporttype, json_build_object('name', audit_entry, 'desc', audit_result-
>>'description')::text as audit_entry, (case when json_array_length(audit_result->'FSBP') = 0 then
'[\
```

Dataset Name	Description	Log Category
security-Rating-CIS-Sub-Control-Passed-Stats-Recommendation	Security Rating CIS Sub Control Passed Statistics Recommendation	

```
select
  cis,
  cis_sub,
  devtype,
  devid,
  compliance,
  severity,
  rtrim(
    to_char(
      sum(score),
      & #039;FM99999999D999'), '.') as score, result, recommendation from /*fabricStart*/(select
devid, devtype, scope, result, severity, compliance, cis, cis_sub, cis_sub_control_id, asset_type,
title, fsbp_id, name, description, recommendation, sum(case when result='passed' then 1 else 0
end) as passed, sum(case when result='exempt' then 1 else 0 end) as exempt, sum(case when
result='failed' then 1 else 0 end) as failed, sum(case when result='dependenciesNotMet' then 1
else 0 end) as unmet, sum(score) as score from (select audit_entry, (instance->>'score')::float as
score, instance->>'deviceID' as devid, instance->>'device' as devtype, (case when instance-
>'domain'->>'type' in ('adom-global', 'global') then 'Global' when instance->'domain'->'id' is not
null then instance->'domain'->>'id' else 'Device' end) as scope, instance->'recommendation' #>> '
{}' as recommendation, instance->>'result' as result, (case instance->>'result' when 'passed' then
'none' when 'exempt' then 'low' else instance->>'severity' end) as severity, compliance, cis, cis_
sub, cis_sub_control_id, asset_type, title, fsbp_id, name, description from (select cis::int as
cis, cis_sub, cis_sub_control_id, t1.asset_type, t1.title, fsbp_id, t2.name, t2.description from
(select split_part(t1.id, '.', 1) as cis,  substr(t1.id, strpos(t1.id, '.')+1) as cis_sub, t1.id
```

```
as cis_sub_control_id, t2.asset_type, t2.name as title, fsbp_id from security_cis_fsbp_map t1
inner join security_cis_mdata t2 on t1.id= t2.id) t1 inner join security_cis_mdata t2 on t1.cis =
t2.id order by cis, cis_sub, fsbp_id) t1 left join (select td.*, reporttype, audit_entry, json_
array_elements_text(compliances) as compliance, instance from (select dvid, reporttype, json_
build_object('name', audit_entry, 'desc', audit_result->>'description')::text as audit_entry,
(case when json_array_length(audit_result->'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-CIS-Sub-Control-Exempt-Result-List	Security Rating CIS Sub Control Exempt List	

```
select
  title,
  count(distinct devid || fsbp_id) as total_num,
  string_agg(
    distinct compliance,
    & #039;,') as compliance, rtrim(to_char(sum(score), 'FM99999999D999'), '.') as score from
/*fabricStart*/(select devid, devtype, scope, result, severity, compliance, cis, cis_sub, cis_sub_
control_id, asset_type, title, fsbp_id, name, description, recommendation, sum(case when
result='passed' then 1 else 0 end) as passed, sum(case when result='exempt' then 1 else 0 end) as
exempt, sum(case when result='failed' then 1 else 0 end) as failed, sum(case when
result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score from (select audit_
entry, (instance->>'score')::float as score, instance->>'deviceID' as devid, instance->>'device'
as devtype, (case when instance->'domain'->>'type' in ('adom-global', 'global') then 'Global' when
instance->'domain'->'id' is not null then instance->'domain'->>'id' else 'Device' end) as scope,
instance->'recommendation' #>> '{}' as recommendation, instance->>'result' as result, (case
instance->>'result' when 'passed' then 'none' when 'exempt' then 'low' else instance->>'severity'
end) as severity, compliance, cis, cis_sub, cis_sub_control_id, asset_type, title, fsbp_id, name,
description from (select cis::int as cis, cis_sub, cis_sub_control_id, t1.asset_type, t1.title,
fsbp_id, t2.name, t2.description from (select split_part(t1.id, '.', 1) as cis,  substr(t1.id,
strpos(t1.id, '.')+1) as cis_sub, t1.id as cis_sub_control_id, t2.asset_type, t2.name as title,
fsbp_id from security_cis_fsbp_map t1 inner join security_cis_mdata t2 on t1.id= t2.id) t1 inner
join security_cis_mdata t2 on t1.cis = t2.id order by cis, cis_sub, fsbp_id) t1 left join (select
td.*, reporttype, audit_entry, json_array_elements_text(compliances) as compliance, instance from
(select dvid, reporttype, json_build_object('name', audit_entry, 'desc', audit_result-
>>'description')::text as audit_entry, (case when json_array_length(audit_result->'FSBP') = 0 then
'[\
```

Dataset Name	Description	Log Category
security-Rating-CIS-Sub-Control-Exempt-Stats-Recommendation	Security Rating CIS Sub Control Exempt Statistics Recommendation	

```
select
  cis,
  cis_sub,
  devtype,
  devid,
```

```
  compliance,
  severity,
  rtrim(
    to_char(
      sum(score),
      & #039;FM99999999D999'), '.') as score, result, recommendation from /*fabricStart*/(select
devid, devtype, scope, result, severity, compliance, cis, cis_sub, cis_sub_control_id, asset_type,
title, fsbp_id, name, description, recommendation, sum(case when result='passed' then 1 else 0
end) as passed, sum(case when result='exempt' then 1 else 0 end) as exempt, sum(case when
result='failed' then 1 else 0 end) as failed, sum(case when result='dependenciesNotMet' then 1
else 0 end) as unmet, sum(score) as score from (select audit_entry, (instance->>'score')::float as
score, instance->>'deviceID' as devid, instance->>'device' as devtype, (case when instance-
>'domain'->>'type' in ('adom-global', 'global') then 'Global' when instance->'domain'->'id' is not
null then instance->'domain'->>'id' else 'Device' end) as scope, instance->'recommendation' #>> '
{}' as recommendation, instance->>'result' as result, (case instance->>'result' when 'passed' then
'none' when 'exempt' then 'low' else instance->>'severity' end) as severity, compliance, cis, cis_
sub, cis_sub_control_id, asset_type, title, fsbp_id, name, description from (select cis::int as
cis, cis_sub, cis_sub_control_id, t1.asset_type, t1.title, fsbp_id, t2.name, t2.description from
(select split_part(t1.id, '.', 1) as cis,  substr(t1.id, strpos(t1.id, '.')+1) as cis_sub, t1.id
as cis_sub_control_id, t2.asset_type, t2.name as title, fsbp_id from security_cis_fsbp_map t1
inner join security_cis_mdata t2 on t1.id= t2.id) t1 inner join security_cis_mdata t2 on t1.cis =
t2.id order by cis, cis_sub, fsbp_id) t1 left join (select td.*, reporttype, audit_entry, json_
array_elements_text(compliances) as compliance, instance from (select dvid, reporttype, json_
build_object('name', audit_entry, 'desc', audit_result->>'description')::text as audit_entry,
(case when json_array_length(audit_result->'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-PCI-Control-Result-Count	Security Rating PCI Control Result by Count	

```
select
  unnest(name) as stats,
  unnest(val) as value
from
  (
    select
      array[ & #039;Failed', 'Passed', 'Exempt', 'Unmet'] as name, array[ count(distinct (case
when failed>0 then pci_sub_control_id||devid||fsbp_id end)), count(distinct (case when failed=0
and unmet=0 and exempt=0 and passed>0 then pci_sub_control_id||devid||fsbp_id end)), count
(distinct (case when failed=0 and unmet=0 and exempt>0 then pci_sub_control_id||devid||fsbp_id
end)), count(distinct (case when failed=0 and unmet>0 then pci_sub_control_id||devid||fsbp_id
end)) ] as val from /*fabricStart*/(select devid, devtype, scope, result, severity, compliance,
pci, pci_sub, pci_sub_control_id, null as asset_type, title, fsbp_id, name, description,
recommendation, sum(case when result='passed' then 1 else 0 end) as passed, sum(case when
result='exempt' then 1 else 0 end) as exempt, sum(case when result='failed' then 1 else 0 end) as
failed, sum(case when result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score
from (select audit_entry, (instance->>'score')::float as score, instance->>'deviceID' as devid,
instance->>'device' as devtype, (case when instance->'domain'->>'type' in ('adom-global',
'global') then 'Global' when instance->'domain'->'id' is not null then instance->'domain'->>'id'
else 'Device' end) as scope, instance->'recommendation' #>> '{}' as recommendation, instance-
```

```
>>'result' as result, (case instance->>'result' when 'passed' then 'none' when 'exempt' then 'low'
else instance->>'severity' end) as severity, compliance, pci, pci_sub, pci_sub_control_id, title,
fsbp_id, name, description from (select pci::int as pci, pci_sub, pci_sub_control_id, t1.title,
fsbp_id, t2.name, null as description from (select split_part(t1.id, '.', 1) as pci,  substr
(t1.id, strpos(t1.id, '.')+1) as pci_sub, t1.id as pci_sub_control_id, t2.name as title, fsbp_id
from security_pci_dss_v321_fsbp_map t1 inner join security_pci_dss_v321_mdata t2 on t1.id= t2.id)
t1 inner join security_pci_dss_v321_mdata t2 on t1.pci = t2.id) t1 left join (select td.*,
reporttype, audit_entry, json_array_elements_text(compliances) as compliance, instance from
(select dvid, reporttype, json_build_object('name', audit_entry, 'desc', audit_result-
>>'description')::text as audit_entry, (case when json_array_length(audit_result->'FSBP') = 0 then
'[\
```

Dataset Name	Description	Log Category
security-Rating-PCI-Control-Compliance-Results	Security Rating PCI Control Compliance Results	

```
select
  & #039;PCI DSS 3.2.1 Checks '||unnest(name) as stats, unnest(val) as value from (select array
['Failed', 'Passed'] as name, array[ count(distinct (case when failed>0 or (failed=0 and passed=0)
then pci_sub_control_id end)), count(distinct (case when failed = 0 and passed>0 then pci_sub_
control_id end)) ] as val from (select pci_sub_control_id, count(distinct (case when failed>0 or
unmet>0 or exempt>0 then devid end)) as failed, count(distinct (case when failed=0 and unmet=0 and
exempt=0 and passed>0 then devid end)) as passed from (select pci_sub_control_id, devid, sum
(failed) as failed, sum(passed) as passed, sum(unmet) as unmet, sum(exempt) as exempt from
/*fabricStart*/(select devid, devtype, scope, result, severity, compliance, pci, pci_sub, pci_sub_
control_id, null as asset_type, title, fsbp_id, name, description, recommendation, sum(case when
result='passed' then 1 else 0 end) as passed, sum(case when result='exempt' then 1 else 0 end) as
exempt, sum(case when result='failed' then 1 else 0 end) as failed, sum(case when
result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score from (select audit_
entry, (instance->>'score')::float as score, instance->>'deviceID' as devid, instance->>'device'
as devtype, (case when instance->'domain'->>'type' in ('adom-global', 'global') then 'Global' when
instance->'domain'->'id' is not null then instance->'domain'->>'id' else 'Device' end) as scope,
instance->'recommendation' #>> '{}' as recommendation, instance->>'result' as result, (case
instance->>'result' when 'passed' then 'none' when 'exempt' then 'low' else instance->>'severity'
end) as severity, compliance, pci, pci_sub, pci_sub_control_id, title, fsbp_id, name, description
from (select pci::int as pci, pci_sub, pci_sub_control_id, t1.title, fsbp_id, t2.name, null as
description from (select split_part(t1.id, '.', 1) as pci,  substr(t1.id, strpos(t1.id, '.')+1) as
pci_sub, t1.id as pci_sub_control_id, t2.name as title, fsbp_id from security_pci_dss_v321_fsbp_
map t1 inner join security_pci_dss_v321_mdata t2 on t1.id= t2.id) t1 inner join security_pci_dss_
v321_mdata t2 on t1.pci = t2.id) t1 left join (select td.*, reporttype, audit_entry, json_array_
elements_text(compliances) as compliance, instance from (select dvid, reporttype, json_build_
object('name', audit_entry, 'desc', audit_result->>'description')::text as audit_entry, (case when
json_array_length(audit_result->'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-PCI-Control-Overview	Security Rating PCI Control Overview	

```
select
  pci,
  (
    case when pci_sub is not null then pci ||& #039;.'||pci_sub else pci::text end) as pci_
control, title, passed, failed from (select pci, pci_sub, title, count(distinct (case when
failed=0 and unmet=0 and exempt=0 and passed>0 then devid end)) as passed, count(distinct (case
when failed>0 or unmet>0 or exempt>0 then devid end)) as failed from ((select pci, pci_sub, devid,
title, sum(failed) as failed, sum(passed) as passed, sum(unmet) as unmet, sum(exempt) as exempt
from /*fabricStart*/(select devid, devtype, scope, result, severity, compliance, pci, pci_sub,
pci_sub_control_id, null as asset_type, title, fsbp_id, name, description, recommendation, sum
(case when result='passed' then 1 else 0 end) as passed, sum(case when result='exempt' then 1 else
0 end) as exempt, sum(case when result='failed' then 1 else 0 end) as failed, sum(case when
result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score from (select audit_
entry, (instance->>'score')::float as score, instance->>'deviceID' as devid, instance->>'device'
as devtype, (case when instance->'domain'->>'type' in ('adom-global', 'global') then 'Global' when
instance->'domain'->'id' is not null then instance->'domain'->>'id' else 'Device' end) as scope,
instance->'recommendation' #>> '{}' as recommendation, instance->>'result' as result, (case
instance->>'result' when 'passed' then 'none' when 'exempt' then 'low' else instance->>'severity'
end) as severity, compliance, pci, pci_sub, pci_sub_control_id, title, fsbp_id, name, description
from (select pci::int as pci, pci_sub, pci_sub_control_id, t1.title, fsbp_id, t2.name, null as
description from (select split_part(t1.id, '.', 1) as pci,  substr(t1.id, strpos(t1.id, '.')+1) as
pci_sub, t1.id as pci_sub_control_id, t2.name as title, fsbp_id from security_pci_dss_v321_fsbp_
map t1 inner join security_pci_dss_v321_mdata t2 on t1.id= t2.id) t1 inner join security_pci_dss_
v321_mdata t2 on t1.pci = t2.id) t1 left join (select td.*, reporttype, audit_entry, json_array_
elements_text(compliances) as compliance, instance from (select dvid, reporttype, json_build_
object('name', audit_entry, 'desc', audit_result->>'description')::text as audit_entry, (case when
json_array_length(audit_result->'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-PCI-Control-Result-List	Security Rating PCI Control Result List	

```
select
  pci,
  name,
  description,
  count(
    distinct (
      case when failed>0 then pci_sub || devid || fsbp_id end
    )
  ) as total_failed,
  count(
    distinct (
      case when failed = 0
      and unmet>0 then pci_sub || devid || fsbp_id end
    )
  ) as total_unmet,
  count(
    distinct (
      case when failed = 0
```

```
      and unmet = 0
      and exempt = 0
      and passed>0 then pci_sub || devid || fsbp_id end
    )
  ) as total_passed,
  count(
    distinct (
      case when failed = 0
      and unmet = 0
      and exempt>0
      and passed = 0 then pci_sub || devid || fsbp_id end
    )
  ) as total_exempt
from

  /*fabricStart*/
  (
    select
      devid,
      devtype,
      scope,
      result,
      severity,
      compliance,
      pci,
      pci_sub,
      pci_sub_control_id,
      null as asset_type,
      title,
      fsbp_id,
      name,
      description,
      recommendation,
      sum(
        case when result =& #039;passed' then 1 else 0 end) as passed, sum(case when
result='exempt' then 1 else 0 end) as exempt, sum(case when result='failed' then 1 else 0 end) as
failed, sum(case when result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score
from (select audit_entry, (instance->>'score')::float as score, instance->>'deviceID' as devid,
instance->>'device' as devtype, (case when instance->'domain'->>'type' in ('adom-global',
'global') then 'Global' when instance->'domain'->'id' is not null then instance->'domain'->>'id'
else 'Device' end) as scope, instance->'recommendation' #>> '{}' as recommendation, instance-
>>'result' as result, (case instance->>'result' when 'passed' then 'none' when 'exempt' then 'low'
else instance->>'severity' end) as severity, compliance, pci, pci_sub, pci_sub_control_id, title,
fsbp_id, name, description from (select pci::int as pci, pci_sub, pci_sub_control_id, t1.title,
fsbp_id, t2.name, null as description from (select split_part(t1.id, '.', 1) as pci,  substr
(t1.id, strpos(t1.id, '.')+1) as pci_sub, t1.id as pci_sub_control_id, t2.name as title, fsbp_id
from security_pci_dss_v321_fsbp_map t1 inner join security_pci_dss_v321_mdata t2 on t1.id= t2.id)
t1 inner join security_pci_dss_v321_mdata t2 on t1.pci = t2.id) t1 left join (select td.*,
reporttype, audit_entry, json_array_elements_text(compliances) as compliance, instance from
(select dvid, reporttype, json_build_object('name', audit_entry, 'desc', audit_result-
>>'description')::text as audit_entry, (case when json_array_length(audit_result->'FSBP') = 0 then
'[\
```

Dataset Name	Description	Log Category
security-Rating-PCI-Sub-Control-Result-List	Security Rating PCI Sub Control Result List	

```
select
  pci_sub_control_id,
  pci,
  pci_sub,
  asset_type,
  title,
  count(distinct devid || fsbp_id) as total_num
from

  /*fabricStart*/
  (
    select
      devid,
      devtype,
      scope,
      result,
      severity,
      compliance,
      pci,
      pci_sub,
      pci_sub_control_id,
      null as asset_type,
      title,
      fsbp_id,
      name,
      description,
      recommendation,
      sum(
        case when result =& #039;passed' then 1 else 0 end) as passed, sum(case when
result='exempt' then 1 else 0 end) as exempt, sum(case when result='failed' then 1 else 0 end) as
failed, sum(case when result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score
from (select audit_entry, (instance->>'score')::float as score, instance->>'deviceID' as devid,
instance->>'device' as devtype, (case when instance->'domain'->>'type' in ('adom-global',
'global') then 'Global' when instance->'domain'->'id' is not null then instance->'domain'->>'id'
else 'Device' end) as scope, instance->'recommendation' #>> '{}' as recommendation, instance-
>>'result' as result, (case instance->>'result' when 'passed' then 'none' when 'exempt' then 'low'
else instance->>'severity' end) as severity, compliance, pci, pci_sub, pci_sub_control_id, title,
fsbp_id, name, description from (select pci::int as pci, pci_sub, pci_sub_control_id, t1.title,
fsbp_id, t2.name, null as description from (select split_part(t1.id, '.', 1) as pci,  substr
(t1.id, strpos(t1.id, '.')+1) as pci_sub, t1.id as pci_sub_control_id, t2.name as title, fsbp_id
from security_pci_dss_v321_fsbp_map t1 inner join security_pci_dss_v321_mdata t2 on t1.id= t2.id)
t1 inner join security_pci_dss_v321_mdata t2 on t1.pci = t2.id) t1 left join (select td.*,
reporttype, audit_entry, json_array_elements_text(compliances) as compliance, instance from
(select dvid, reporttype, json_build_object('name', audit_entry, 'desc', audit_result-
>>'description')::text as audit_entry, (case when json_array_length(audit_result->'FSBP') = 0 then
'[\
```

Dataset Name	Description	Log Category
security-Rating-PCI-Sub-Control-Failed-Result-List	Security Rating PCI Sub Control Fail List	

```
select
  title,
  count(distinct devid || fsbp_id) as total_num,
  string_agg(
    distinct compliance,
    & #039;,') as compliance, rtrim(to_char(sum(score), 'FM99999999D999'), '.') as score from
/*fabricStart*/(select devid, devtype, scope, result, severity, compliance, pci, pci_sub, pci_sub_
control_id, null as asset_type, title, fsbp_id, name, description, recommendation, sum(case when
result='passed' then 1 else 0 end) as passed, sum(case when result='exempt' then 1 else 0 end) as
exempt, sum(case when result='failed' then 1 else 0 end) as failed, sum(case when
result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score from (select audit_
entry, (instance->>'score')::float as score, instance->>'deviceID' as devid, instance->>'device'
as devtype, (case when instance->'domain'->>'type' in ('adom-global', 'global') then 'Global' when
instance->'domain'->'id' is not null then instance->'domain'->>'id' else 'Device' end) as scope,
instance->'recommendation' #>> '{}' as recommendation, instance->>'result' as result, (case
instance->>'result' when 'passed' then 'none' when 'exempt' then 'low' else instance->>'severity'
end) as severity, compliance, pci, pci_sub, pci_sub_control_id, title, fsbp_id, name, description
from (select pci::int as pci, pci_sub, pci_sub_control_id, t1.title, fsbp_id, t2.name, null as
description from (select split_part(t1.id, '.', 1) as pci,  substr(t1.id, strpos(t1.id, '.')+1) as
pci_sub, t1.id as pci_sub_control_id, t2.name as title, fsbp_id from security_pci_dss_v321_fsbp_
map t1 inner join security_pci_dss_v321_mdata t2 on t1.id= t2.id) t1 inner join security_pci_dss_
v321_mdata t2 on t1.pci = t2.id) t1 left join (select td.*, reporttype, audit_entry, json_array_
elements_text(compliances) as compliance, instance from (select dvid, reporttype, json_build_
object('name', audit_entry, 'desc', audit_result->>'description')::text as audit_entry, (case when
json_array_length(audit_result->'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-PCI-Sub-Control-Failed-Stats-Recommendation	Security Rating PCI Sub Control Failed Statistics Recommendation	

```
select
  pci,
  pci_sub,
  devtype,
  devid,
  compliance,
  severity,
  rtrim(
    to_char(
      sum(score),
      & #039;FM99999999D999'), '.') as score, result, recommendation from /*fabricStart*/(select
devid, devtype, scope, result, severity, compliance, pci, pci_sub, pci_sub_control_id, null as
asset_type, title, fsbp_id, name, description, recommendation, sum(case when result='passed' then
1 else 0 end) as passed, sum(case when result='exempt' then 1 else 0 end) as exempt, sum(case when
```

```
result='failed' then 1 else 0 end) as failed, sum(case when result='dependenciesNotMet' then 1
else 0 end) as unmet, sum(score) as score from (select audit_entry, (instance->>'score')::float as
score, instance->>'deviceID' as devid, instance->>'device' as devtype, (case when instance-
>'domain'->>'type' in ('adom-global', 'global') then 'Global' when instance->'domain'->'id' is not
null then instance->'domain'->>'id' else 'Device' end) as scope, instance->'recommendation' #>> '
{}' as recommendation, instance->>'result' as result, (case instance->>'result' when 'passed' then
'none' when 'exempt' then 'low' else instance->>'severity' end) as severity, compliance, pci, pci_
sub, pci_sub_control_id, title, fsbp_id, name, description from (select pci::int as pci, pci_sub,
pci_sub_control_id, t1.title, fsbp_id, t2.name, null as description from (select split_part(t1.id,
'.', 1) as pci,  substr(t1.id, strpos(t1.id, '.')+1) as pci_sub, t1.id as pci_sub_control_id,
t2.name as title, fsbp_id from security_pci_dss_v321_fsbp_map t1 inner join security_pci_dss_v321_
mdata t2 on t1.id= t2.id) t1 inner join security_pci_dss_v321_mdata t2 on t1.pci = t2.id) t1 left
join (select td.*, reporttype, audit_entry, json_array_elements_text(compliances) as compliance,
instance from (select dvid, reporttype, json_build_object('name', audit_entry, 'desc', audit_
result->>'description')::text as audit_entry, (case when json_array_length(audit_result->'FSBP') =
0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-PCI-Sub-Control-Unmet-Result-List	Security Rating PCI Sub Control Unmet List	

```
select
  title,
  count(distinct devid || fsbp_id) as total_num,
  string_agg(
    distinct compliance,
    & #039;,') as compliance, rtrim(to_char(sum(score), 'FM99999999D999'), '.') as score from
/*fabricStart*/(select devid, devtype, scope, result, severity, compliance, pci, pci_sub, pci_sub_
control_id, null as asset_type, title, fsbp_id, name, description, recommendation, sum(case when
result='passed' then 1 else 0 end) as passed, sum(case when result='exempt' then 1 else 0 end) as
exempt, sum(case when result='failed' then 1 else 0 end) as failed, sum(case when
result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score from (select audit_
entry, (instance->>'score')::float as score, instance->>'deviceID' as devid, instance->>'device'
as devtype, (case when instance->'domain'->>'type' in ('adom-global', 'global') then 'Global' when
instance->'domain'->'id' is not null then instance->'domain'->>'id' else 'Device' end) as scope,
instance->'recommendation' #>> '{}' as recommendation, instance->>'result' as result, (case
instance->>'result' when 'passed' then 'none' when 'exempt' then 'low' else instance->>'severity'
end) as severity, compliance, pci, pci_sub, pci_sub_control_id, title, fsbp_id, name, description
from (select pci::int as pci, pci_sub, pci_sub_control_id, t1.title, fsbp_id, t2.name, null as
description from (select split_part(t1.id, '.', 1) as pci,  substr(t1.id, strpos(t1.id, '.')+1) as
pci_sub, t1.id as pci_sub_control_id, t2.name as title, fsbp_id from security_pci_dss_v321_fsbp_
map t1 inner join security_pci_dss_v321_mdata t2 on t1.id= t2.id) t1 inner join security_pci_dss_
v321_mdata t2 on t1.pci = t2.id) t1 left join (select td.*, reporttype, audit_entry, json_array_
elements_text(compliances) as compliance, instance from (select dvid, reporttype, json_build_
object('name', audit_entry, 'desc', audit_result->>'description')::text as audit_entry, (case when
json_array_length(audit_result->'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-PCI-Sub-Control-Unmet-Stats-Recommendation	Security Rating PCI Sub Control Unmet Statistics Recommendation	

```
select
  pci,
  pci_sub,
  devtype,
  devid,
  compliance,
  severity,
  rtrim(
    to_char(
      sum(score),
      & #039;FM99999999D999'), '.') as score, result, recommendation from /*fabricStart*/(select
devid, devtype, scope, result, severity, compliance, pci, pci_sub, pci_sub_control_id, null as
asset_type, title, fsbp_id, name, description, recommendation, sum(case when result='passed' then
1 else 0 end) as passed, sum(case when result='exempt' then 1 else 0 end) as exempt, sum(case when
result='failed' then 1 else 0 end) as failed, sum(case when result='dependenciesNotMet' then 1
else 0 end) as unmet, sum(score) as score from (select audit_entry, (instance->>'score')::float as
score, instance->>'deviceID' as devid, instance->>'device' as devtype, (case when instance-
>'domain'->>'type' in ('adom-global', 'global') then 'Global' when instance->'domain'->'id' is not
null then instance->'domain'->>'id' else 'Device' end) as scope, instance->'recommendation' #>> '
{}' as recommendation, instance->>'result' as result, (case instance->>'result' when 'passed' then
'none' when 'exempt' then 'low' else instance->>'severity' end) as severity, compliance, pci, pci_
sub, pci_sub_control_id, title, fsbp_id, name, description from (select pci::int as pci, pci_sub,
pci_sub_control_id, t1.title, fsbp_id, t2.name, null as description from (select split_part(t1.id,
'.', 1) as pci,  substr(t1.id, strpos(t1.id, '.')+1) as pci_sub, t1.id as pci_sub_control_id,
t2.name as title, fsbp_id from security_pci_dss_v321_fsbp_map t1 inner join security_pci_dss_v321_
mdata t2 on t1.id= t2.id) t1 inner join security_pci_dss_v321_mdata t2 on t1.pci = t2.id) t1 left
join (select td.*, reporttype, audit_entry, json_array_elements_text(compliances) as compliance,
instance from (select dvid, reporttype, json_build_object('name', audit_entry, 'desc', audit_
result->>'description')::text as audit_entry, (case when json_array_length(audit_result->'FSBP') =
0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-PCI-Sub-Control-Passed-Result-List	Security Rating PCI Sub Control Passed List	

```
select
  title,
  count(distinct devid || fsbp_id) as total_num,
  string_agg(
    distinct compliance,
    & #039;,') as compliance, rtrim(to_char(sum(score), 'FM99999999D999'), '.') as score from
/*fabricStart*/(select devid, devtype, scope, result, severity, compliance, pci, pci_sub, pci_sub_
control_id, null as asset_type, title, fsbp_id, name, description, recommendation, sum(case when
result='passed' then 1 else 0 end) as passed, sum(case when result='exempt' then 1 else 0 end) as
```

```
exempt, sum(case when result='failed' then 1 else 0 end) as failed, sum(case when
result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score from (select audit_
entry, (instance->>'score')::float as score, instance->>'deviceID' as devid, instance->>'device'
as devtype, (case when instance->'domain'->>'type' in ('adom-global', 'global') then 'Global' when
instance->'domain'->'id' is not null then instance->'domain'->>'id' else 'Device' end) as scope,
instance->'recommendation' #>> '{}' as recommendation, instance->>'result' as result, (case
instance->>'result' when 'passed' then 'none' when 'exempt' then 'low' else instance->>'severity'
end) as severity, compliance, pci, pci_sub, pci_sub_control_id, title, fsbp_id, name, description
from (select pci::int as pci, pci_sub, pci_sub_control_id, t1.title, fsbp_id, t2.name, null as
description from (select split_part(t1.id, '.', 1) as pci,  substr(t1.id, strpos(t1.id, '.')+1) as
pci_sub, t1.id as pci_sub_control_id, t2.name as title, fsbp_id from security_pci_dss_v321_fsbp_
map t1 inner join security_pci_dss_v321_mdata t2 on t1.id= t2.id) t1 inner join security_pci_dss_
v321_mdata t2 on t1.pci = t2.id) t1 left join (select td.*, reporttype, audit_entry, json_array_
elements_text(compliances) as compliance, instance from (select dvid, reporttype, json_build_
object('name', audit_entry, 'desc', audit_result->>'description')::text as audit_entry, (case when
json_array_length(audit_result->'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-PCI-Sub-Control-Passed-Stats-Recommendation	Security Rating PCI Sub Control Passed Statistics Recommendation	

```
select
  pci,
  pci_sub,
  devtype,
  devid,
  compliance,
  severity,
  rtrim(
    to_char(
      sum(score),
      & #039;FM99999999D999'), '.') as score, result, recommendation from /*fabricStart*/(select
devid, devtype, scope, result, severity, compliance, pci, pci_sub, pci_sub_control_id, null as
asset_type, title, fsbp_id, name, description, recommendation, sum(case when result='passed' then
1 else 0 end) as passed, sum(case when result='exempt' then 1 else 0 end) as exempt, sum(case when
result='failed' then 1 else 0 end) as failed, sum(case when result='dependenciesNotMet' then 1
else 0 end) as unmet, sum(score) as score from (select audit_entry, (instance->>'score')::float as
score, instance->>'deviceID' as devid, instance->>'device' as devtype, (case when instance-
>'domain'->>'type' in ('adom-global', 'global') then 'Global' when instance->'domain'->'id' is not
null then instance->'domain'->>'id' else 'Device' end) as scope, instance->'recommendation' #>> '
{}' as recommendation, instance->>'result' as result, (case instance->>'result' when 'passed' then
'none' when 'exempt' then 'low' else instance->>'severity' end) as severity, compliance, pci, pci_
sub, pci_sub_control_id, title, fsbp_id, name, description from (select pci::int as pci, pci_sub,
pci_sub_control_id, t1.title, fsbp_id, t2.name, null as description from (select split_part(t1.id,
'.', 1) as pci,  substr(t1.id, strpos(t1.id, '.')+1) as pci_sub, t1.id as pci_sub_control_id,
t2.name as title, fsbp_id from security_pci_dss_v321_fsbp_map t1 inner join security_pci_dss_v321_
mdata t2 on t1.id= t2.id) t1 inner join security_pci_dss_v321_mdata t2 on t1.pci = t2.id) t1 left
join (select td.*, reporttype, audit_entry, json_array_elements_text(compliances) as compliance,
instance from (select dvid, reporttype, json_build_object('name', audit_entry, 'desc', audit_
```

```
result->>'description')::text as audit_entry, (case when json_array_length(audit_result->'FSBP') =
0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-PCI-Sub-Control-Exempt-Result-List	Security Rating PCI Sub Control Exempt List	

```
select
  title,
  count(distinct devid || fsbp_id) as total_num,
  string_agg(
    distinct compliance,
    & #039;,') as compliance, rtrim(to_char(sum(score), 'FM99999999D999'), '.') as score from
/*fabricStart*/(select devid, devtype, scope, result, severity, compliance, pci, pci_sub, pci_sub_
control_id, null as asset_type, title, fsbp_id, name, description, recommendation, sum(case when
result='passed' then 1 else 0 end) as passed, sum(case when result='exempt' then 1 else 0 end) as
exempt, sum(case when result='failed' then 1 else 0 end) as failed, sum(case when
result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score from (select audit_
entry, (instance->>'score')::float as score, instance->>'deviceID' as devid, instance->>'device'
as devtype, (case when instance->'domain'->>'type' in ('adom-global', 'global') then 'Global' when
instance->'domain'->'id' is not null then instance->'domain'->>'id' else 'Device' end) as scope,
instance->'recommendation' #>> '{}' as recommendation, instance->>'result' as result, (case
instance->>'result' when 'passed' then 'none' when 'exempt' then 'low' else instance->>'severity'
end) as severity, compliance, pci, pci_sub, pci_sub_control_id, title, fsbp_id, name, description
from (select pci::int as pci, pci_sub, pci_sub_control_id, t1.title, fsbp_id, t2.name, null as
description from (select split_part(t1.id, '.', 1) as pci,  substr(t1.id, strpos(t1.id, '.')+1) as
pci_sub, t1.id as pci_sub_control_id, t2.name as title, fsbp_id from security_pci_dss_v321_fsbp_
map t1 inner join security_pci_dss_v321_mdata t2 on t1.id= t2.id) t1 inner join security_pci_dss_
v321_mdata t2 on t1.pci = t2.id) t1 left join (select td.*, reporttype, audit_entry, json_array_
elements_text(compliances) as compliance, instance from (select dvid, reporttype, json_build_
object('name', audit_entry, 'desc', audit_result->>'description')::text as audit_entry, (case when
json_array_length(audit_result->'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-PCI-Sub-Control-Exempt-Stats-Recommendation	Security Rating PCI Sub Control Exempt Statistics Recommendation	

```
select
  pci,
  pci_sub,
  devtype,
  devid,
  compliance,
  severity,
  rtrim(
    to_char(
      sum(score),
```

```
      & #039;FM99999999D999'), '.') as score, result, recommendation from /*fabricStart*/(select
devid, devtype, scope, result, severity, compliance, pci, pci_sub, pci_sub_control_id, null as
asset_type, title, fsbp_id, name, description, recommendation, sum(case when result='passed' then
1 else 0 end) as passed, sum(case when result='exempt' then 1 else 0 end) as exempt, sum(case when
result='failed' then 1 else 0 end) as failed, sum(case when result='dependenciesNotMet' then 1
else 0 end) as unmet, sum(score) as score from (select audit_entry, (instance->>'score')::float as
score, instance->>'deviceID' as devid, instance->>'device' as devtype, (case when instance-
>'domain'->>'type' in ('adom-global', 'global') then 'Global' when instance->'domain'->'id' is not
null then instance->'domain'->>'id' else 'Device' end) as scope, instance->'recommendation' #>> '
{}' as recommendation, instance->>'result' as result, (case instance->>'result' when 'passed' then
'none' when 'exempt' then 'low' else instance->>'severity' end) as severity, compliance, pci, pci_
sub, pci_sub_control_id, title, fsbp_id, name, description from (select pci::int as pci, pci_sub,
pci_sub_control_id, t1.title, fsbp_id, t2.name, null as description from (select split_part(t1.id,
'.', 1) as pci,  substr(t1.id, strpos(t1.id, '.')+1) as pci_sub, t1.id as pci_sub_control_id,
t2.name as title, fsbp_id from security_pci_dss_v321_fsbp_map t1 inner join security_pci_dss_v321_
mdata t2 on t1.id= t2.id) t1 inner join security_pci_dss_v321_mdata t2 on t1.pci = t2.id) t1 left
join (select td.*, reporttype, audit_entry, json_array_elements_text(compliances) as compliance,
instance from (select dvid, reporttype, json_build_object('name', audit_entry, 'desc', audit_
result->>'description')::text as audit_entry, (case when json_array_length(audit_result->'FSBP') =
0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-ISO-Control-Result-Count	Security Rating ISO Control Result by Count	

```
select
  unnest(name) as stats,
  unnest(val) as value
from
  (
    select
      array[ & #039;Failed', 'Passed', 'Exempt', 'Unmet'] as name, array[ count(distinct (case
when failed>0 then iso_sub_control_id||devid||fsbp_id end)), count(distinct (case when failed=0
and unmet=0 and exempt=0 and passed>0 then iso_sub_control_id||devid||fsbp_id end)), count
(distinct (case when failed=0 and unmet=0 and exempt>0 then iso_sub_control_id||devid||fsbp_id
end)), count(distinct (case when failed=0 and unmet>0 then iso_sub_control_id||devid||fsbp_id
end)) ] as val from /*fabricStart*/(select devid, devtype, scope, result, severity, compliance,
iso, iso_sub, iso_sub_control_id, null as asset_type, title, fsbp_id, name, description,
recommendation, sum(case when result='passed' then 1 else 0 end) as passed, sum(case when
result='exempt' then 1 else 0 end) as exempt, sum(case when result='failed' then 1 else 0 end) as
failed, sum(case when result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score
from (select audit_entry, (instance->>'score')::float as score, instance->>'deviceID' as devid,
instance->>'device' as devtype, (case when instance->'domain'->>'type' in ('adom-global',
'global') then 'Global' when instance->'domain'->'id' is not null then instance->'domain'->>'id'
else 'Device' end) as scope, instance->'recommendation' #>> '{}' as recommendation, instance-
>>'result' as result, (case instance->>'result' when 'passed' then 'none' when 'exempt' then 'low'
else instance->>'severity' end) as severity, compliance, iso, iso_sub, iso_sub_control_id, title,
fsbp_id, name, description from (select iso::int as iso, iso_sub, iso_sub_control_id, t1.title,
fsbp_id, t2.name, t2.title as description from (select split_part(t1.id, '.', 1) as iso,  substr
(t1.id, strpos(t1.id, '.')+1) as iso_sub, t1.id as iso_sub_control_id, t2.name as title, fsbp_id
```

```
from security_iso_27001_fsbp_map t1 inner join security_iso_27001_mdata t2 on t1.id= t2.id) t1
inner join security_iso_27001_mdata t2 on t1.iso = t2.id) t1 left join (select td.*, reporttype,
audit_entry, json_array_elements_text(compliances) as compliance, instance from (select dvid,
reporttype, json_build_object('name', audit_entry, 'desc', audit_result->>'description')::text as
audit_entry, (case when json_array_length(audit_result->'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-ISO-Control-Compliance-Results	Security Rating ISO Control Compliance Results	

```
select
  & #039;ISO Compliance Checks '||unnest(name) as stats, unnest(val) as value from (select array
['Failed', 'Passed'] as name, array[ count(distinct (case when failed>0 or (failed=0 and passed=0)
then iso_sub_control_id end)), count(distinct (case when failed = 0 and passed>0 then iso_sub_
control_id end)) ] as val from (select iso_sub_control_id, count(distinct (case when failed>0 or
unmet>0 or exempt>0 then devid end)) as failed, count(distinct (case when failed=0 and unmet=0 and
exempt=0 and passed>0 then devid end)) as passed from (select iso_sub_control_id, devid, sum
(failed) as failed, sum(passed) as passed, sum(unmet) as unmet, sum(exempt) as exempt from
/*fabricStart*/(select devid, devtype, scope, result, severity, compliance, iso, iso_sub, iso_sub_
control_id, null as asset_type, title, fsbp_id, name, description, recommendation, sum(case when
result='passed' then 1 else 0 end) as passed, sum(case when result='exempt' then 1 else 0 end) as
exempt, sum(case when result='failed' then 1 else 0 end) as failed, sum(case when
result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score from (select audit_
entry, (instance->>'score')::float as score, instance->>'deviceID' as devid, instance->>'device'
as devtype, (case when instance->'domain'->>'type' in ('adom-global', 'global') then 'Global' when
instance->'domain'->'id' is not null then instance->'domain'->>'id' else 'Device' end) as scope,
instance->'recommendation' #>> '{}' as recommendation, instance->>'result' as result, (case
instance->>'result' when 'passed' then 'none' when 'exempt' then 'low' else instance->>'severity'
end) as severity, compliance, iso, iso_sub, iso_sub_control_id, title, fsbp_id, name, description
from (select iso::int as iso, iso_sub, iso_sub_control_id, t1.title, fsbp_id, t2.name, t2.title as
description from (select split_part(t1.id, '.', 1) as iso,  substr(t1.id, strpos(t1.id, '.')+1) as
iso_sub, t1.id as iso_sub_control_id, t2.name as title, fsbp_id from security_iso_27001_fsbp_map
t1 inner join security_iso_27001_mdata t2 on t1.id= t2.id) t1 inner join security_iso_27001_mdata
t2 on t1.iso = t2.id) t1 left join (select td.*, reporttype, audit_entry, json_array_elements_text
(compliances) as compliance, instance from (select dvid, reporttype, json_build_object('name',
audit_entry, 'desc', audit_result->>'description')::text as audit_entry, (case when json_array_
length(audit_result->'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-ISO-Control-Overview	Security Rating ISO Control Overview	

```
select
  (
    case when iso_sub is not null then iso ||& #039;.'||iso_sub else iso::text end) as iso_
control, title, passed, failed from (select iso, iso_sub, title, count(distinct (case when
failed=0 and unmet=0 and exempt=0 and passed>0 then devid end)) as passed, count(distinct (case
```

```
when failed>0 or unmet>0 or exempt>0 then devid end)) as failed from ((select iso, iso_sub, devid,
title, sum(failed) as failed, sum(passed) as passed, sum(unmet) as unmet, sum(exempt) as exempt
from /*fabricStart*/(select devid, devtype, scope, result, severity, compliance, iso, iso_sub,
iso_sub_control_id, null as asset_type, title, fsbp_id, name, description, recommendation, sum
(case when result='passed' then 1 else 0 end) as passed, sum(case when result='exempt' then 1 else
0 end) as exempt, sum(case when result='failed' then 1 else 0 end) as failed, sum(case when
result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score from (select audit_
entry, (instance->>'score')::float as score, instance->>'deviceID' as devid, instance->>'device'
as devtype, (case when instance->'domain'->>'type' in ('adom-global', 'global') then 'Global' when
instance->'domain'->'id' is not null then instance->'domain'->>'id' else 'Device' end) as scope,
instance->'recommendation' #>> '{}' as recommendation, instance->>'result' as result, (case
instance->>'result' when 'passed' then 'none' when 'exempt' then 'low' else instance->>'severity'
end) as severity, compliance, iso, iso_sub, iso_sub_control_id, title, fsbp_id, name, description
from (select iso::int as iso, iso_sub, iso_sub_control_id, t1.title, fsbp_id, t2.name, t2.title as
description from (select split_part(t1.id, '.', 1) as iso,  substr(t1.id, strpos(t1.id, '.')+1) as
iso_sub, t1.id as iso_sub_control_id, t2.name as title, fsbp_id from security_iso_27001_fsbp_map
t1 inner join security_iso_27001_mdata t2 on t1.id= t2.id) t1 inner join security_iso_27001_mdata
t2 on t1.iso = t2.id) t1 left join (select td.*, reporttype, audit_entry, json_array_elements_text
(compliances) as compliance, instance from (select dvid, reporttype, json_build_object('name',
audit_entry, 'desc', audit_result->>'description')::text as audit_entry, (case when json_array_
length(audit_result->'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-ISO-Control-Result-List	Security Rating ISO Control Result List	

```
select
  iso,
  name,
  description,
  count(
    distinct (
      case when failed>0 then iso_sub || devid || fsbp_id end
    )
  ) as total_failed,
  count(
    distinct (
      case when failed = 0
      and unmet>0 then iso_sub || devid || fsbp_id end
    )
  ) as total_unmet,
  count(
    distinct (
      case when failed = 0
      and unmet = 0
      and exempt = 0
      and passed>0 then iso_sub || devid || fsbp_id end
    )
  ) as total_passed,
  count(
```

```
    distinct (
      case when failed = 0
      and unmet = 0
      and exempt>0
      and passed = 0 then iso_sub || devid || fsbp_id end
    )
  ) as total_exempt
from

  /*fabricStart*/
  (
    select
      devid,
      devtype,
      scope,
      result,
      severity,
      compliance,
      iso,
      iso_sub,
      iso_sub_control_id,
      null as asset_type,
      title,
      fsbp_id,
      name,
      description,
      recommendation,
      sum(
        case when result =& #039;passed' then 1 else 0 end) as passed, sum(case when
result='exempt' then 1 else 0 end) as exempt, sum(case when result='failed' then 1 else 0 end) as
failed, sum(case when result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score
from (select audit_entry, (instance->>'score')::float as score, instance->>'deviceID' as devid,
instance->>'device' as devtype, (case when instance->'domain'->>'type' in ('adom-global',
'global') then 'Global' when instance->'domain'->'id' is not null then instance->'domain'->>'id'
else 'Device' end) as scope, instance->'recommendation' #>> '{}' as recommendation, instance-
>>'result' as result, (case instance->>'result' when 'passed' then 'none' when 'exempt' then 'low'
else instance->>'severity' end) as severity, compliance, iso, iso_sub, iso_sub_control_id, title,
fsbp_id, name, description from (select iso::int as iso, iso_sub, iso_sub_control_id, t1.title,
fsbp_id, t2.name, t2.title as description from (select split_part(t1.id, '.', 1) as iso,  substr
(t1.id, strpos(t1.id, '.')+1) as iso_sub, t1.id as iso_sub_control_id, t2.name as title, fsbp_id
from security_iso_27001_fsbp_map t1 inner join security_iso_27001_mdata t2 on t1.id= t2.id) t1
inner join security_iso_27001_mdata t2 on t1.iso = t2.id) t1 left join (select td.*, reporttype,
audit_entry, json_array_elements_text(compliances) as compliance, instance from (select dvid,
reporttype, json_build_object('name', audit_entry, 'desc', audit_result->>'description')::text as
audit_entry, (case when json_array_length(audit_result->'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-ISO-Sub-Control-Result-List	Security Rating ISO Sub Control Result List	

```
select
  iso_sub_control_id,
  iso,
  iso_sub,
  asset_type,
  title,
  count(distinct devid || fsbp_id) as total_num
from

  /*fabricStart*/
  (
    select
      devid,
      devtype,
      scope,
      result,
      severity,
      compliance,
      iso,
      iso_sub,
      iso_sub_control_id,
      null as asset_type,
      title,
      fsbp_id,
      name,
      description,
      recommendation,
      sum(
        case when result =& #039;passed' then 1 else 0 end) as passed, sum(case when
result='exempt' then 1 else 0 end) as exempt, sum(case when result='failed' then 1 else 0 end) as
failed, sum(case when result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score
from (select audit_entry, (instance->>'score')::float as score, instance->>'deviceID' as devid,
instance->>'device' as devtype, (case when instance->'domain'->>'type' in ('adom-global',
'global') then 'Global' when instance->'domain'->'id' is not null then instance->'domain'->>'id'
else 'Device' end) as scope, instance->'recommendation' #>> '{}' as recommendation, instance-
>>'result' as result, (case instance->>'result' when 'passed' then 'none' when 'exempt' then 'low'
else instance->>'severity' end) as severity, compliance, iso, iso_sub, iso_sub_control_id, title,
fsbp_id, name, description from (select iso::int as iso, iso_sub, iso_sub_control_id, t1.title,
fsbp_id, t2.name, t2.title as description from (select split_part(t1.id, '.', 1) as iso,  substr
(t1.id, strpos(t1.id, '.')+1) as iso_sub, t1.id as iso_sub_control_id, t2.name as title, fsbp_id
from security_iso_27001_fsbp_map t1 inner join security_iso_27001_mdata t2 on t1.id= t2.id) t1
inner join security_iso_27001_mdata t2 on t1.iso = t2.id) t1 left join (select td.*, reporttype,
audit_entry, json_array_elements_text(compliances) as compliance, instance from (select dvid,
reporttype, json_build_object('name', audit_entry, 'desc', audit_result->>'description')::text as
audit_entry, (case when json_array_length(audit_result->'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-ISO-Sub-Control-Failed-Result-List	Security Rating ISO Sub Control Fail List	

```
select
  title,
  count(distinct devid || fsbp_id) as total_num,
  string_agg(
    distinct compliance,
    & #039;,') as compliance, rtrim(to_char(sum(score), 'FM99999999D999'), '.') as score from
/*fabricStart*/(select devid, devtype, scope, result, severity, compliance, iso, iso_sub, iso_sub_
control_id, null as asset_type, title, fsbp_id, name, description, recommendation, sum(case when
result='passed' then 1 else 0 end) as passed, sum(case when result='exempt' then 1 else 0 end) as
exempt, sum(case when result='failed' then 1 else 0 end) as failed, sum(case when
result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score from (select audit_
entry, (instance->>'score')::float as score, instance->>'deviceID' as devid, instance->>'device'
as devtype, (case when instance->'domain'->>'type' in ('adom-global', 'global') then 'Global' when
instance->'domain'->'id' is not null then instance->'domain'->>'id' else 'Device' end) as scope,
instance->'recommendation' #>> '{}' as recommendation, instance->>'result' as result, (case
instance->>'result' when 'passed' then 'none' when 'exempt' then 'low' else instance->>'severity'
end) as severity, compliance, iso, iso_sub, iso_sub_control_id, title, fsbp_id, name, description
from (select iso::int as iso, iso_sub, iso_sub_control_id, t1.title, fsbp_id, t2.name, t2.title as
description from (select split_part(t1.id, '.', 1) as iso,  substr(t1.id, strpos(t1.id, '.')+1) as
iso_sub, t1.id as iso_sub_control_id, t2.name as title, fsbp_id from security_iso_27001_fsbp_map
t1 inner join security_iso_27001_mdata t2 on t1.id= t2.id) t1 inner join security_iso_27001_mdata
t2 on t1.iso = t2.id) t1 left join (select td.*, reporttype, audit_entry, json_array_elements_text
(compliances) as compliance, instance from (select dvid, reporttype, json_build_object('name',
audit_entry, 'desc', audit_result->>'description')::text as audit_entry, (case when json_array_
length(audit_result->'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-ISO-Sub-Control-Failed-Stats-Recommendation	Security Rating ISO Sub Control Failed Statistics Recommendation	

```
select
  iso,
  iso_sub,
  devtype,
  devid,
  compliance,
  severity,
  rtrim(
    to_char(
      sum(score),
      & #039;FM99999999D999'), '.') as score, result, recommendation from /*fabricStart*/(select
devid, devtype, scope, result, severity, compliance, iso, iso_sub, iso_sub_control_id, null as
asset_type, title, fsbp_id, name, description, recommendation, sum(case when result='passed' then
1 else 0 end) as passed, sum(case when result='exempt' then 1 else 0 end) as exempt, sum(case when
result='failed' then 1 else 0 end) as failed, sum(case when result='dependenciesNotMet' then 1
else 0 end) as unmet, sum(score) as score from (select audit_entry, (instance->>'score')::float as
score, instance->>'deviceID' as devid, instance->>'device' as devtype, (case when instance-
>'domain'->>'type' in ('adom-global', 'global') then 'Global' when instance->'domain'->'id' is not
null then instance->'domain'->>'id' else 'Device' end) as scope, instance->'recommendation' #>> '
{}' as recommendation, instance->>'result' as result, (case instance->>'result' when 'passed' then
```

```
'none' when 'exempt' then 'low' else instance->>'severity' end) as severity, compliance, iso, iso_
sub, iso_sub_control_id, title, fsbp_id, name, description from (select iso::int as iso, iso_sub,
iso_sub_control_id, t1.title, fsbp_id, t2.name, t2.title as description from (select split_part
(t1.id, '.', 1) as iso,  substr(t1.id, strpos(t1.id, '.')+1) as iso_sub, t1.id as iso_sub_control_
id, t2.name as title, fsbp_id from security_iso_27001_fsbp_map t1 inner join security_iso_27001_
mdata t2 on t1.id= t2.id) t1 inner join security_iso_27001_mdata t2 on t1.iso = t2.id) t1 left
join (select td.*, reporttype, audit_entry, json_array_elements_text(compliances) as compliance,
instance from (select dvid, reporttype, json_build_object('name', audit_entry, 'desc', audit_
result->>'description')::text as audit_entry, (case when json_array_length(audit_result->'FSBP') =
0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-ISO-Sub-Control-Unmet-Result-List	Security Rating ISO Sub Control Unmet List	

```
select
  title,
  count(distinct devid || fsbp_id) as total_num,
  string_agg(
    distinct compliance,
    & #039;,') as compliance, rtrim(to_char(sum(score), 'FM99999999D999'), '.') as score from
/*fabricStart*/(select devid, devtype, scope, result, severity, compliance, iso, iso_sub, iso_sub_
control_id, null as asset_type, title, fsbp_id, name, description, recommendation, sum(case when
result='passed' then 1 else 0 end) as passed, sum(case when result='exempt' then 1 else 0 end) as
exempt, sum(case when result='failed' then 1 else 0 end) as failed, sum(case when
result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score from (select audit_
entry, (instance->>'score')::float as score, instance->>'deviceID' as devid, instance->>'device'
as devtype, (case when instance->'domain'->>'type' in ('adom-global', 'global') then 'Global' when
instance->'domain'->'id' is not null then instance->'domain'->>'id' else 'Device' end) as scope,
instance->'recommendation' #>> '{}' as recommendation, instance->>'result' as result, (case
instance->>'result' when 'passed' then 'none' when 'exempt' then 'low' else instance->>'severity'
end) as severity, compliance, iso, iso_sub, iso_sub_control_id, title, fsbp_id, name, description
from (select iso::int as iso, iso_sub, iso_sub_control_id, t1.title, fsbp_id, t2.name, t2.title as
description from (select split_part(t1.id, '.', 1) as iso,  substr(t1.id, strpos(t1.id, '.')+1) as
iso_sub, t1.id as iso_sub_control_id, t2.name as title, fsbp_id from security_iso_27001_fsbp_map
t1 inner join security_iso_27001_mdata t2 on t1.id= t2.id) t1 inner join security_iso_27001_mdata
t2 on t1.iso = t2.id) t1 left join (select td.*, reporttype, audit_entry, json_array_elements_text
(compliances) as compliance, instance from (select dvid, reporttype, json_build_object('name',
audit_entry, 'desc', audit_result->>'description')::text as audit_entry, (case when json_array_
length(audit_result->'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-ISO-Sub-Control-Unmet-Stats-Recommendation	Security Rating ISO Sub Control Unmet Statistics Recommendation	

```
select
  iso,
```

```
  iso_sub,
  devtype,
  devid,
  compliance,
  severity,
  rtrim(
    to_char(
      sum(score),
      & #039;FM99999999D999'), '.') as score, result, recommendation from /*fabricStart*/(select
devid, devtype, scope, result, severity, compliance, iso, iso_sub, iso_sub_control_id, null as
asset_type, title, fsbp_id, name, description, recommendation, sum(case when result='passed' then
1 else 0 end) as passed, sum(case when result='exempt' then 1 else 0 end) as exempt, sum(case when
result='failed' then 1 else 0 end) as failed, sum(case when result='dependenciesNotMet' then 1
else 0 end) as unmet, sum(score) as score from (select audit_entry, (instance->>'score')::float as
score, instance->>'deviceID' as devid, instance->>'device' as devtype, (case when instance-
>'domain'->>'type' in ('adom-global', 'global') then 'Global' when instance->'domain'->'id' is not
null then instance->'domain'->>'id' else 'Device' end) as scope, instance->'recommendation' #>> '
{}' as recommendation, instance->>'result' as result, (case instance->>'result' when 'passed' then
'none' when 'exempt' then 'low' else instance->>'severity' end) as severity, compliance, iso, iso_
sub, iso_sub_control_id, title, fsbp_id, name, description from (select iso::int as iso, iso_sub,
iso_sub_control_id, t1.title, fsbp_id, t2.name, t2.title as description from (select split_part
(t1.id, '.', 1) as iso,  substr(t1.id, strpos(t1.id, '.')+1) as iso_sub, t1.id as iso_sub_control_
id, t2.name as title, fsbp_id from security_iso_27001_fsbp_map t1 inner join security_iso_27001_
mdata t2 on t1.id= t2.id) t1 inner join security_iso_27001_mdata t2 on t1.iso = t2.id) t1 left
join (select td.*, reporttype, audit_entry, json_array_elements_text(compliances) as compliance,
instance from (select dvid, reporttype, json_build_object('name', audit_entry, 'desc', audit_
result->>'description')::text as audit_entry, (case when json_array_length(audit_result->'FSBP') =
0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-ISO-Sub-Control-Passed-Result-List	Security Rating ISO Sub Control Passed List	

```
select
  title,
  count(distinct devid || fsbp_id) as total_num,
  string_agg(
    distinct compliance,
    & #039;,') as compliance, rtrim(to_char(sum(score), 'FM99999999D999'), '.') as score from
/*fabricStart*/(select devid, devtype, scope, result, severity, compliance, iso, iso_sub, iso_sub_
control_id, null as asset_type, title, fsbp_id, name, description, recommendation, sum(case when
result='passed' then 1 else 0 end) as passed, sum(case when result='exempt' then 1 else 0 end) as
exempt, sum(case when result='failed' then 1 else 0 end) as failed, sum(case when
result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score from (select audit_
entry, (instance->>'score')::float as score, instance->>'deviceID' as devid, instance->>'device'
as devtype, (case when instance->'domain'->>'type' in ('adom-global', 'global') then 'Global' when
instance->'domain'->'id' is not null then instance->'domain'->>'id' else 'Device' end) as scope,
instance->'recommendation' #>> '{}' as recommendation, instance->>'result' as result, (case
instance->>'result' when 'passed' then 'none' when 'exempt' then 'low' else instance->>'severity'
end) as severity, compliance, iso, iso_sub, iso_sub_control_id, title, fsbp_id, name, description
```

```
from (select iso::int as iso, iso_sub, iso_sub_control_id, t1.title, fsbp_id, t2.name, t2.title as
description from (select split_part(t1.id, '.', 1) as iso,  substr(t1.id, strpos(t1.id, '.')+1) as
iso_sub, t1.id as iso_sub_control_id, t2.name as title, fsbp_id from security_iso_27001_fsbp_map
t1 inner join security_iso_27001_mdata t2 on t1.id= t2.id) t1 inner join security_iso_27001_mdata
t2 on t1.iso = t2.id) t1 left join (select td.*, reporttype, audit_entry, json_array_elements_text
(compliances) as compliance, instance from (select dvid, reporttype, json_build_object('name',
audit_entry, 'desc', audit_result->>'description')::text as audit_entry, (case when json_array_
length(audit_result->'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-ISO-Sub-Control-Passed-Stats-Recommendation	Security Rating ISO Sub Control Passed Statistics Recommendation	

```
select
  iso,
  iso_sub,
  devtype,
  devid,
  compliance,
  severity,
  rtrim(
    to_char(
      sum(score),
      & #039;FM99999999D999'), '.') as score, result, recommendation from /*fabricStart*/(select
devid, devtype, scope, result, severity, compliance, iso, iso_sub, iso_sub_control_id, null as
asset_type, title, fsbp_id, name, description, recommendation, sum(case when result='passed' then
1 else 0 end) as passed, sum(case when result='exempt' then 1 else 0 end) as exempt, sum(case when
result='failed' then 1 else 0 end) as failed, sum(case when result='dependenciesNotMet' then 1
else 0 end) as unmet, sum(score) as score from (select audit_entry, (instance->>'score')::float as
score, instance->>'deviceID' as devid, instance->>'device' as devtype, (case when instance-
>'domain'->>'type' in ('adom-global', 'global') then 'Global' when instance->'domain'->'id' is not
null then instance->'domain'->>'id' else 'Device' end) as scope, instance->'recommendation' #>> '
{}' as recommendation, instance->>'result' as result, (case instance->>'result' when 'passed' then
'none' when 'exempt' then 'low' else instance->>'severity' end) as severity, compliance, iso, iso_
sub, iso_sub_control_id, title, fsbp_id, name, description from (select iso::int as iso, iso_sub,
iso_sub_control_id, t1.title, fsbp_id, t2.name, t2.title as description from (select split_part
(t1.id, '.', 1) as iso,  substr(t1.id, strpos(t1.id, '.')+1) as iso_sub, t1.id as iso_sub_control_
id, t2.name as title, fsbp_id from security_iso_27001_fsbp_map t1 inner join security_iso_27001_
mdata t2 on t1.id= t2.id) t1 inner join security_iso_27001_mdata t2 on t1.iso = t2.id) t1 left
join (select td.*, reporttype, audit_entry, json_array_elements_text(compliances) as compliance,
instance from (select dvid, reporttype, json_build_object('name', audit_entry, 'desc', audit_
result->>'description')::text as audit_entry, (case when json_array_length(audit_result->'FSBP') =
0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-ISO-Sub-Control-Exempt-Result-List	Security Rating ISO Sub Control Exempt List	

```
select
  title,
  count(distinct devid || fsbp_id) as total_num,
  string_agg(
    distinct compliance,
    & #039;,') as compliance, rtrim(to_char(sum(score), 'FM99999999D999'), '.') as score from
/*fabricStart*/(select devid, devtype, scope, result, severity, compliance, iso, iso_sub, iso_sub_
control_id, null as asset_type, title, fsbp_id, name, description, recommendation, sum(case when
result='passed' then 1 else 0 end) as passed, sum(case when result='exempt' then 1 else 0 end) as
exempt, sum(case when result='failed' then 1 else 0 end) as failed, sum(case when
result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score from (select audit_
entry, (instance->>'score')::float as score, instance->>'deviceID' as devid, instance->>'device'
as devtype, (case when instance->'domain'->>'type' in ('adom-global', 'global') then 'Global' when
instance->'domain'->'id' is not null then instance->'domain'->>'id' else 'Device' end) as scope,
instance->'recommendation' #>> '{}' as recommendation, instance->>'result' as result, (case
instance->>'result' when 'passed' then 'none' when 'exempt' then 'low' else instance->>'severity'
end) as severity, compliance, iso, iso_sub, iso_sub_control_id, title, fsbp_id, name, description
from (select iso::int as iso, iso_sub, iso_sub_control_id, t1.title, fsbp_id, t2.name, t2.title as
description from (select split_part(t1.id, '.', 1) as iso,  substr(t1.id, strpos(t1.id, '.')+1) as
iso_sub, t1.id as iso_sub_control_id, t2.name as title, fsbp_id from security_iso_27001_fsbp_map
t1 inner join security_iso_27001_mdata t2 on t1.id= t2.id) t1 inner join security_iso_27001_mdata
t2 on t1.iso = t2.id) t1 left join (select td.*, reporttype, audit_entry, json_array_elements_text
(compliances) as compliance, instance from (select dvid, reporttype, json_build_object('name',
audit_entry, 'desc', audit_result->>'description')::text as audit_entry, (case when json_array_
length(audit_result->'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-ISO-Sub-Control-Exempt-Stats-Recommendation	Security Rating ISO Sub Control Exempt Statistics Recommendation	

```
select
  iso,
  iso_sub,
  devtype,
  devid,
  compliance,
  severity,
  rtrim(
    to_char(
      sum(score),
      & #039;FM99999999D999'), '.') as score, result, recommendation from /*fabricStart*/(select
devid, devtype, scope, result, severity, compliance, iso, iso_sub, iso_sub_control_id, null as
asset_type, title, fsbp_id, name, description, recommendation, sum(case when result='passed' then
1 else 0 end) as passed, sum(case when result='exempt' then 1 else 0 end) as exempt, sum(case when
result='failed' then 1 else 0 end) as failed, sum(case when result='dependenciesNotMet' then 1
else 0 end) as unmet, sum(score) as score from (select audit_entry, (instance->>'score')::float as
score, instance->>'deviceID' as devid, instance->>'device' as devtype, (case when instance-
>'domain'->>'type' in ('adom-global', 'global') then 'Global' when instance->'domain'->'id' is not
null then instance->'domain'->>'id' else 'Device' end) as scope, instance->'recommendation' #>> '
{}' as recommendation, instance->>'result' as result, (case instance->>'result' when 'passed' then
```

```
'none' when 'exempt' then 'low' else instance->>'severity' end) as severity, compliance, iso, iso_
sub, iso_sub_control_id, title, fsbp_id, name, description from (select iso::int as iso, iso_sub,
iso_sub_control_id, t1.title, fsbp_id, t2.name, t2.title as description from (select split_part
(t1.id, '.', 1) as iso,  substr(t1.id, strpos(t1.id, '.')+1) as iso_sub, t1.id as iso_sub_control_
id, t2.name as title, fsbp_id from security_iso_27001_fsbp_map t1 inner join security_iso_27001_
mdata t2 on t1.id= t2.id) t1 inner join security_iso_27001_mdata t2 on t1.iso = t2.id) t1 left
join (select td.*, reporttype, audit_entry, json_array_elements_text(compliances) as compliance,
instance from (select dvid, reporttype, json_build_object('name', audit_entry, 'desc', audit_
result->>'description')::text as audit_entry, (case when json_array_length(audit_result->'FSBP') =
0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-HIPAA-Control-Result-Count	Security Rating HIPAA Compliance Result by Count	

```
select
  unnest(name) as stats,
  unnest(val) as value
from
  (
    select
      array[ & #039;Failed', 'Passed', 'Exempt', 'Unmet'] as name, array[ count(distinct (case
when failed>0 then hipaa_sub_control_id||devid||fsbp_id end)), count(distinct (case when failed=0
and unmet=0 and exempt=0 and passed>0 then hipaa_sub_control_id||devid||fsbp_id end)), count
(distinct (case when failed=0 and unmet=0 and exempt>0 then hipaa_sub_control_id||devid||fsbp_id
end)), count(distinct (case when failed=0 and unmet>0 then hipaa_sub_control_id||devid||fsbp_id
end)) ] as val from /*fabricStart*/(select devid, devtype, scope, result, severity, compliance,
hipaa, hipaa_sub, hipaa_sub_control_id, null as asset_type, title, fsbp_id, name, description,
recommendation, sum(case when result='passed' then 1 else 0 end) as passed, sum(case when
result='exempt' then 1 else 0 end) as exempt, sum(case when result='failed' then 1 else 0 end) as
failed, sum(case when result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score
from (select audit_entry, (instance->>'score')::float as score, instance->>'deviceID' as devid,
instance->>'device' as devtype, (case when instance->'domain'->>'type' in ('adom-global',
'global') then 'Global' when instance->'domain'->'id' is not null then instance->'domain'->>'id'
else 'Device' end) as scope, instance->'recommendation' #>> '{}' as recommendation, instance-
>>'result' as result, (case instance->>'result' when 'passed' then 'none' when 'exempt' then 'low'
else instance->>'severity' end) as severity, compliance, hipaa, hipaa_sub, hipaa_sub_control_id,
title, fsbp_id, name, description from (select hipaa as hipaa, hipaa_sub, hipaa_sub_control_id,
t1.title, fsbp_id, t2.name, null as description from (select split_part(t1.id, '(', 1) as hipaa,
substr(t1.id, strpos(t1.id, '(')) as hipaa_sub, t1.id as hipaa_sub_control_id, t2.name as title,
fsbp_id from security_hipaa_fsbp_map t1 inner join security_hipaa_mdata t2 on t1.id= t2.id) t1
inner join security_hipaa_mdata t2 on t1.hipaa = t2.id) t1 left join (select td.*, reporttype,
audit_entry, json_array_elements_text(compliances) as compliance, instance from (select dvid,
reporttype, json_build_object('name', audit_entry, 'desc', audit_result->>'description')::text as
audit_entry, (case when json_array_length(audit_result->'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-HIPAA-Control-Compliance-Results	Security Rating HIPAA Compliance Results	

```
select
  & #039;HIPAA Checks '||unnest(name) as stats, unnest(val) as value from (select array['Failed',
'Passed'] as name, array[ count(distinct (case when failed>0 or (failed=0 and passed=0) then
hipaa_sub_control_id end)), count(distinct (case when failed = 0 and passed>0 then hipaa_sub_
control_id end)) ] as val from (select hipaa_sub_control_id, count(distinct (case when failed>0 or
unmet>0 or exempt>0 then devid end)) as failed, count(distinct (case when failed=0 and unmet=0 and
exempt=0 and passed>0 then devid end)) as passed from (select hipaa_sub_control_id, devid, sum
(failed) as failed, sum(passed) as passed, sum(unmet) as unmet, sum(exempt) as exempt from
/*fabricStart*/(select devid, devtype, scope, result, severity, compliance, hipaa, hipaa_sub,
hipaa_sub_control_id, null as asset_type, title, fsbp_id, name, description, recommendation, sum
(case when result='passed' then 1 else 0 end) as passed, sum(case when result='exempt' then 1 else
0 end) as exempt, sum(case when result='failed' then 1 else 0 end) as failed, sum(case when
result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score from (select audit_
entry, (instance->>'score')::float as score, instance->>'deviceID' as devid, instance->>'device'
as devtype, (case when instance->'domain'->>'type' in ('adom-global', 'global') then 'Global' when
instance->'domain'->'id' is not null then instance->'domain'->>'id' else 'Device' end) as scope,
instance->'recommendation' #>> '{}' as recommendation, instance->>'result' as result, (case
instance->>'result' when 'passed' then 'none' when 'exempt' then 'low' else instance->>'severity'
end) as severity, compliance, hipaa, hipaa_sub, hipaa_sub_control_id, title, fsbp_id, name,
description from (select hipaa as hipaa, hipaa_sub, hipaa_sub_control_id, t1.title, fsbp_id,
t2.name, null as description from (select split_part(t1.id, '(', 1) as hipaa,  substr(t1.id,
strpos(t1.id, '(')) as hipaa_sub, t1.id as hipaa_sub_control_id, t2.name as title, fsbp_id from
security_hipaa_fsbp_map t1 inner join security_hipaa_mdata t2 on t1.id= t2.id) t1 inner join
security_hipaa_mdata t2 on t1.hipaa = t2.id) t1 left join (select td.*, reporttype, audit_entry,
json_array_elements_text(compliances) as compliance, instance from (select dvid, reporttype, json_
build_object('name', audit_entry, 'desc', audit_result->>'description')::text as audit_entry,
(case when json_array_length(audit_result->'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-HIPAA-Control-Overview	Security Rating HIPAA Compliance Overview	

```
select
  hipaa,
  (
    case when hipaa_sub is not null then hipaa || hipaa_sub else hipaa end
  ) as hipaa_control,
  title,
  passed,
  failed
from
  (
    select
      hipaa,
```

```
      hipaa_sub,
      title,
      count(
        distinct (
          case when failed = 0
          and unmet = 0
          and exempt = 0
          and passed>0 then devid end
        )
      ) as passed,
      count(
        distinct (
          case when failed>0
          or unmet>0
          or exempt>0 then devid end
        )
      ) as failed
    from
      (
        (
          select
            hipaa,
            hipaa_sub,
            devid,
            title,
            sum(failed) as failed,
            sum(passed) as passed,
            sum(unmet) as unmet,
            sum(exempt) as exempt
          from

            /*fabricStart*/
            (
              select
                devid,
                devtype,
                scope,
                result,
                severity,
                compliance,
                hipaa,
                hipaa_sub,
                hipaa_sub_control_id,
                null as asset_type,
                title,
                fsbp_id,
                name,
                description,
                recommendation,
                sum(
                  case when result =& #039;passed' then 1 else 0 end) as passed, sum(case when
result='exempt' then 1 else 0 end) as exempt, sum(case when result='failed' then 1 else 0 end) as
```

```
failed, sum(case when result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score
from (select audit_entry, (instance->>'score')::float as score, instance->>'deviceID' as devid,
instance->>'device' as devtype, (case when instance->'domain'->>'type' in ('adom-global',
'global') then 'Global' when instance->'domain'->'id' is not null then instance->'domain'->>'id'
else 'Device' end) as scope, instance->'recommendation' #>> '{}' as recommendation, instance-
>>'result' as result, (case instance->>'result' when 'passed' then 'none' when 'exempt' then 'low'
else instance->>'severity' end) as severity, compliance, hipaa, hipaa_sub, hipaa_sub_control_id,
title, fsbp_id, name, description from (select hipaa as hipaa, hipaa_sub, hipaa_sub_control_id,
t1.title, fsbp_id, t2.name, null as description from (select split_part(t1.id, '(', 1) as hipaa,
substr(t1.id, strpos(t1.id, '(')) as hipaa_sub, t1.id as hipaa_sub_control_id, t2.name as title,
fsbp_id from security_hipaa_fsbp_map t1 inner join security_hipaa_mdata t2 on t1.id= t2.id) t1
inner join security_hipaa_mdata t2 on t1.hipaa = t2.id) t1 left join (select td.*, reporttype,
audit_entry, json_array_elements_text(compliances) as compliance, instance from (select dvid,
reporttype, json_build_object('name', audit_entry, 'desc', audit_result->>'description')::text as
audit_entry, (case when json_array_length(audit_result->'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-HIPAA-Control-Result-List	Security Rating HIPAA Compliance Result List	

```
select
  hipaa,
  name,
  description,
  count(
    distinct (
      case when failed>0 then hipaa_sub || devid || fsbp_id end
    )
  ) as total_failed,
  count(
    distinct (
      case when failed = 0
      and unmet>0 then hipaa_sub || devid || fsbp_id end
    )
  ) as total_unmet,
  count(
    distinct (
      case when failed = 0
      and unmet = 0
      and exempt = 0
      and passed>0 then hipaa_sub || devid || fsbp_id end
    )
  ) as total_passed,
  count(
    distinct (
      case when failed = 0
      and unmet = 0
      and exempt>0
      and passed = 0 then hipaa_sub || devid || fsbp_id end
    )
```

```
  ) as total_exempt
from

  /*fabricStart*/
  (
    select
      devid,
      devtype,
      scope,
      result,
      severity,
      compliance,
      hipaa,
      hipaa_sub,
      hipaa_sub_control_id,
      null as asset_type,
      title,
      fsbp_id,
      name,
      description,
      recommendation,
      sum(
        case when result =& #039;passed' then 1 else 0 end) as passed, sum(case when
result='exempt' then 1 else 0 end) as exempt, sum(case when result='failed' then 1 else 0 end) as
failed, sum(case when result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score
from (select audit_entry, (instance->>'score')::float as score, instance->>'deviceID' as devid,
instance->>'device' as devtype, (case when instance->'domain'->>'type' in ('adom-global',
'global') then 'Global' when instance->'domain'->'id' is not null then instance->'domain'->>'id'
else 'Device' end) as scope, instance->'recommendation' #>> '{}' as recommendation, instance-
>>'result' as result, (case instance->>'result' when 'passed' then 'none' when 'exempt' then 'low'
else instance->>'severity' end) as severity, compliance, hipaa, hipaa_sub, hipaa_sub_control_id,
title, fsbp_id, name, description from (select hipaa as hipaa, hipaa_sub, hipaa_sub_control_id,
t1.title, fsbp_id, t2.name, null as description from (select split_part(t1.id, '(', 1) as hipaa,
substr(t1.id, strpos(t1.id, '(')) as hipaa_sub, t1.id as hipaa_sub_control_id, t2.name as title,
fsbp_id from security_hipaa_fsbp_map t1 inner join security_hipaa_mdata t2 on t1.id= t2.id) t1
inner join security_hipaa_mdata t2 on t1.hipaa = t2.id) t1 left join (select td.*, reporttype,
audit_entry, json_array_elements_text(compliances) as compliance, instance from (select dvid,
reporttype, json_build_object('name', audit_entry, 'desc', audit_result->>'description')::text as
audit_entry, (case when json_array_length(audit_result->'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-HIPAA-Sub-Control-Result-List	Security Rating HIPAA Sub Compliance Result List	

```
select
  hipaa_sub_control_id,
  hipaa,
  hipaa_sub,
  asset_type,
  title,
```

```
  count(distinct devid || fsbp_id) as total_num
from

  /*fabricStart*/
  (
    select
      devid,
      devtype,
      scope,
      result,
      severity,
      compliance,
      hipaa,
      hipaa_sub,
      hipaa_sub_control_id,
      null as asset_type,
      title,
      fsbp_id,
      name,
      description,
      recommendation,
      sum(
        case when result =& #039;passed' then 1 else 0 end) as passed, sum(case when
result='exempt' then 1 else 0 end) as exempt, sum(case when result='failed' then 1 else 0 end) as
failed, sum(case when result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score
from (select audit_entry, (instance->>'score')::float as score, instance->>'deviceID' as devid,
instance->>'device' as devtype, (case when instance->'domain'->>'type' in ('adom-global',
'global') then 'Global' when instance->'domain'->'id' is not null then instance->'domain'->>'id'
else 'Device' end) as scope, instance->'recommendation' #>> '{}' as recommendation, instance-
>>'result' as result, (case instance->>'result' when 'passed' then 'none' when 'exempt' then 'low'
else instance->>'severity' end) as severity, compliance, hipaa, hipaa_sub, hipaa_sub_control_id,
title, fsbp_id, name, description from (select hipaa as hipaa, hipaa_sub, hipaa_sub_control_id,
t1.title, fsbp_id, t2.name, null as description from (select split_part(t1.id, '(', 1) as hipaa,
substr(t1.id, strpos(t1.id, '(')) as hipaa_sub, t1.id as hipaa_sub_control_id, t2.name as title,
fsbp_id from security_hipaa_fsbp_map t1 inner join security_hipaa_mdata t2 on t1.id= t2.id) t1
inner join security_hipaa_mdata t2 on t1.hipaa = t2.id) t1 left join (select td.*, reporttype,
audit_entry, json_array_elements_text(compliances) as compliance, instance from (select dvid,
reporttype, json_build_object('name', audit_entry, 'desc', audit_result->>'description')::text as
audit_entry, (case when json_array_length(audit_result->'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-HIPAA-Sub-Control-Failed-Result-List	Security Rating HIPAA Sub Compliance Fail List	

```
select
  title,
  count(distinct devid || fsbp_id) as total_num,
  string_agg(
    distinct compliance,
    & #039;,') as compliance, rtrim(to_char(sum(score), 'FM99999999D999'), '.') as score from
```

```
/*fabricStart*/(select devid, devtype, scope, result, severity, compliance, hipaa, hipaa_sub,
hipaa_sub_control_id, null as asset_type, title, fsbp_id, name, description, recommendation, sum
(case when result='passed' then 1 else 0 end) as passed, sum(case when result='exempt' then 1 else
0 end) as exempt, sum(case when result='failed' then 1 else 0 end) as failed, sum(case when
result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score from (select audit_
entry, (instance->>'score')::float as score, instance->>'deviceID' as devid, instance->>'device'
as devtype, (case when instance->'domain'->>'type' in ('adom-global', 'global') then 'Global' when
instance->'domain'->'id' is not null then instance->'domain'->>'id' else 'Device' end) as scope,
instance->'recommendation' #>> '{}' as recommendation, instance->>'result' as result, (case
instance->>'result' when 'passed' then 'none' when 'exempt' then 'low' else instance->>'severity'
end) as severity, compliance, hipaa, hipaa_sub, hipaa_sub_control_id, title, fsbp_id, name,
description from (select hipaa as hipaa, hipaa_sub, hipaa_sub_control_id, t1.title, fsbp_id,
t2.name, null as description from (select split_part(t1.id, '(', 1) as hipaa,  substr(t1.id,
strpos(t1.id, '(')) as hipaa_sub, t1.id as hipaa_sub_control_id, t2.name as title, fsbp_id from
security_hipaa_fsbp_map t1 inner join security_hipaa_mdata t2 on t1.id= t2.id) t1 inner join
security_hipaa_mdata t2 on t1.hipaa = t2.id) t1 left join (select td.*, reporttype, audit_entry,
json_array_elements_text(compliances) as compliance, instance from (select dvid, reporttype, json_
build_object('name', audit_entry, 'desc', audit_result->>'description')::text as audit_entry,
(case when json_array_length(audit_result->'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-HIPAA-Sub-Control-Failed-Stats-Recommendation	Security Rating HIPAA Sub Compliance Failed Statistics Recommendation	

```
select
  hipaa,
  hipaa_sub,
  devtype,
  devid,
  compliance,
  severity,
  rtrim(
    to_char(
      sum(score),
      & #039;FM99999999D999'), '.') as score, result, recommendation from /*fabricStart*/(select
devid, devtype, scope, result, severity, compliance, hipaa, hipaa_sub, hipaa_sub_control_id, null
as asset_type, title, fsbp_id, name, description, recommendation, sum(case when result='passed'
then 1 else 0 end) as passed, sum(case when result='exempt' then 1 else 0 end) as exempt, sum(case
when result='failed' then 1 else 0 end) as failed, sum(case when result='dependenciesNotMet' then
1 else 0 end) as unmet, sum(score) as score from (select audit_entry, (instance->>'score')::float
as score, instance->>'deviceID' as devid, instance->>'device' as devtype, (case when instance-
>'domain'->>'type' in ('adom-global', 'global') then 'Global' when instance->'domain'->'id' is not
null then instance->'domain'->>'id' else 'Device' end) as scope, instance->'recommendation' #>> '
{}' as recommendation, instance->>'result' as result, (case instance->>'result' when 'passed' then
'none' when 'exempt' then 'low' else instance->>'severity' end) as severity, compliance, hipaa,
hipaa_sub, hipaa_sub_control_id, title, fsbp_id, name, description from (select hipaa as hipaa,
hipaa_sub, hipaa_sub_control_id, t1.title, fsbp_id, t2.name, null as description from (select
split_part(t1.id, '(', 1) as hipaa,  substr(t1.id, strpos(t1.id, '(')) as hipaa_sub, t1.id as
hipaa_sub_control_id, t2.name as title, fsbp_id from security_hipaa_fsbp_map t1 inner join
```

```
security_hipaa_mdata t2 on t1.id= t2.id) t1 inner join security_hipaa_mdata t2 on t1.hipaa =
t2.id) t1 left join (select td.*, reporttype, audit_entry, json_array_elements_text(compliances)
as compliance, instance from (select dvid, reporttype, json_build_object('name', audit_entry,
'desc', audit_result->>'description')::text as audit_entry, (case when json_array_length(audit_
result->'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-HIPAA-Sub-Control-Unmet-Result-List	Security Rating HIPAA Sub Compliance Unmet List	

```
select
  title,
  count(distinct devid || fsbp_id) as total_num,
  string_agg(
    distinct compliance,
    & #039;,') as compliance, rtrim(to_char(sum(score), 'FM99999999D999'), '.') as score from
/*fabricStart*/(select devid, devtype, scope, result, severity, compliance, hipaa, hipaa_sub,
hipaa_sub_control_id, null as asset_type, title, fsbp_id, name, description, recommendation, sum
(case when result='passed' then 1 else 0 end) as passed, sum(case when result='exempt' then 1 else
0 end) as exempt, sum(case when result='failed' then 1 else 0 end) as failed, sum(case when
result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score from (select audit_
entry, (instance->>'score')::float as score, instance->>'deviceID' as devid, instance->>'device'
as devtype, (case when instance->'domain'->>'type' in ('adom-global', 'global') then 'Global' when
instance->'domain'->'id' is not null then instance->'domain'->>'id' else 'Device' end) as scope,
instance->'recommendation' #>> '{}' as recommendation, instance->>'result' as result, (case
instance->>'result' when 'passed' then 'none' when 'exempt' then 'low' else instance->>'severity'
end) as severity, compliance, hipaa, hipaa_sub, hipaa_sub_control_id, title, fsbp_id, name,
description from (select hipaa as hipaa, hipaa_sub, hipaa_sub_control_id, t1.title, fsbp_id,
t2.name, null as description from (select split_part(t1.id, '(', 1) as hipaa,  substr(t1.id,
strpos(t1.id, '(')) as hipaa_sub, t1.id as hipaa_sub_control_id, t2.name as title, fsbp_id from
security_hipaa_fsbp_map t1 inner join security_hipaa_mdata t2 on t1.id= t2.id) t1 inner join
security_hipaa_mdata t2 on t1.hipaa = t2.id) t1 left join (select td.*, reporttype, audit_entry,
json_array_elements_text(compliances) as compliance, instance from (select dvid, reporttype, json_
build_object('name', audit_entry, 'desc', audit_result->>'description')::text as audit_entry,
(case when json_array_length(audit_result->'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-HIPAA-Sub-Control-Unmet-Stats-Recommendation	Security Rating HIPAA Sub Compliance Unmet Statistics Recommendation	

```
select
  hipaa,
  hipaa_sub,
  devtype,
  devid,
  compliance,
```

```
  severity,
  rtrim(
    to_char(
      sum(score),
      & #039;FM99999999D999'), '.') as score, result, recommendation from /*fabricStart*/(select
devid, devtype, scope, result, severity, compliance, hipaa, hipaa_sub, hipaa_sub_control_id, null
as asset_type, title, fsbp_id, name, description, recommendation, sum(case when result='passed'
then 1 else 0 end) as passed, sum(case when result='exempt' then 1 else 0 end) as exempt, sum(case
when result='failed' then 1 else 0 end) as failed, sum(case when result='dependenciesNotMet' then
1 else 0 end) as unmet, sum(score) as score from (select audit_entry, (instance->>'score')::float
as score, instance->>'deviceID' as devid, instance->>'device' as devtype, (case when instance-
>'domain'->>'type' in ('adom-global', 'global') then 'Global' when instance->'domain'->'id' is not
null then instance->'domain'->>'id' else 'Device' end) as scope, instance->'recommendation' #>> '
{}' as recommendation, instance->>'result' as result, (case instance->>'result' when 'passed' then
'none' when 'exempt' then 'low' else instance->>'severity' end) as severity, compliance, hipaa,
hipaa_sub, hipaa_sub_control_id, title, fsbp_id, name, description from (select hipaa as hipaa,
hipaa_sub, hipaa_sub_control_id, t1.title, fsbp_id, t2.name, null as description from (select
split_part(t1.id, '(', 1) as hipaa,  substr(t1.id, strpos(t1.id, '(')) as hipaa_sub, t1.id as
hipaa_sub_control_id, t2.name as title, fsbp_id from security_hipaa_fsbp_map t1 inner join
security_hipaa_mdata t2 on t1.id= t2.id) t1 inner join security_hipaa_mdata t2 on t1.hipaa =
t2.id) t1 left join (select td.*, reporttype, audit_entry, json_array_elements_text(compliances)
as compliance, instance from (select dvid, reporttype, json_build_object('name', audit_entry,
'desc', audit_result->>'description')::text as audit_entry, (case when json_array_length(audit_
result->'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-HIPAA-Sub-Control-Passed-Result-List	Security Rating HIPAA Sub Compliance Passed List	

```
select
  title,
  count(distinct devid || fsbp_id) as total_num,
  string_agg(
    distinct compliance,
    & #039;,') as compliance, rtrim(to_char(sum(score), 'FM99999999D999'), '.') as score from
/*fabricStart*/(select devid, devtype, scope, result, severity, compliance, hipaa, hipaa_sub,
hipaa_sub_control_id, null as asset_type, title, fsbp_id, name, description, recommendation, sum
(case when result='passed' then 1 else 0 end) as passed, sum(case when result='exempt' then 1 else
0 end) as exempt, sum(case when result='failed' then 1 else 0 end) as failed, sum(case when
result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score from (select audit_
entry, (instance->>'score')::float as score, instance->>'deviceID' as devid, instance->>'device'
as devtype, (case when instance->'domain'->>'type' in ('adom-global', 'global') then 'Global' when
instance->'domain'->'id' is not null then instance->'domain'->>'id' else 'Device' end) as scope,
instance->'recommendation' #>> '{}' as recommendation, instance->>'result' as result, (case
instance->>'result' when 'passed' then 'none' when 'exempt' then 'low' else instance->>'severity'
end) as severity, compliance, hipaa, hipaa_sub, hipaa_sub_control_id, title, fsbp_id, name,
description from (select hipaa as hipaa, hipaa_sub, hipaa_sub_control_id, t1.title, fsbp_id,
t2.name, null as description from (select split_part(t1.id, '(', 1) as hipaa,  substr(t1.id,
strpos(t1.id, '(')) as hipaa_sub, t1.id as hipaa_sub_control_id, t2.name as title, fsbp_id from
security_hipaa_fsbp_map t1 inner join security_hipaa_mdata t2 on t1.id= t2.id) t1 inner join
```

```
security_hipaa_mdata t2 on t1.hipaa = t2.id) t1 left join (select td.*, reporttype, audit_entry,
json_array_elements_text(compliances) as compliance, instance from (select dvid, reporttype, json_
build_object('name', audit_entry, 'desc', audit_result->>'description')::text as audit_entry,
(case when json_array_length(audit_result->'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-HIPAA-Sub-Control-Passed-Stats-Recommendation	Security Rating HIPAA Sub Compliance Passed Statistics Recommendation	

```
select
  hipaa,
  hipaa_sub,
  devtype,
  devid,
  compliance,
  severity,
  rtrim(
    to_char(
      sum(score),
      & #039;FM99999999D999'), '.') as score, result, recommendation from /*fabricStart*/(select
devid, devtype, scope, result, severity, compliance, hipaa, hipaa_sub, hipaa_sub_control_id, null
as asset_type, title, fsbp_id, name, description, recommendation, sum(case when result='passed'
then 1 else 0 end) as passed, sum(case when result='exempt' then 1 else 0 end) as exempt, sum(case
when result='failed' then 1 else 0 end) as failed, sum(case when result='dependenciesNotMet' then
1 else 0 end) as unmet, sum(score) as score from (select audit_entry, (instance->>'score')::float
as score, instance->>'deviceID' as devid, instance->>'device' as devtype, (case when instance-
>'domain'->>'type' in ('adom-global', 'global') then 'Global' when instance->'domain'->'id' is not
null then instance->'domain'->>'id' else 'Device' end) as scope, instance->'recommendation' #>> '
{}' as recommendation, instance->>'result' as result, (case instance->>'result' when 'passed' then
'none' when 'exempt' then 'low' else instance->>'severity' end) as severity, compliance, hipaa,
hipaa_sub, hipaa_sub_control_id, title, fsbp_id, name, description from (select hipaa as hipaa,
hipaa_sub, hipaa_sub_control_id, t1.title, fsbp_id, t2.name, null as description from (select
split_part(t1.id, '(', 1) as hipaa,  substr(t1.id, strpos(t1.id, '(')) as hipaa_sub, t1.id as
hipaa_sub_control_id, t2.name as title, fsbp_id from security_hipaa_fsbp_map t1 inner join
security_hipaa_mdata t2 on t1.id= t2.id) t1 inner join security_hipaa_mdata t2 on t1.hipaa =
t2.id) t1 left join (select td.*, reporttype, audit_entry, json_array_elements_text(compliances)
as compliance, instance from (select dvid, reporttype, json_build_object('name', audit_entry,
'desc', audit_result->>'description')::text as audit_entry, (case when json_array_length(audit_
result->'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-HIPAA-Sub-Control-Exempt-Result-List	Security Rating HIPAA Sub Compliance Exempt List	

```
select
  title,
```

```
  count(distinct devid || fsbp_id) as total_num,
  string_agg(
    distinct compliance,
    & #039;,') as compliance, rtrim(to_char(sum(score), 'FM99999999D999'), '.') as score from
/*fabricStart*/(select devid, devtype, scope, result, severity, compliance, hipaa, hipaa_sub,
hipaa_sub_control_id, null as asset_type, title, fsbp_id, name, description, recommendation, sum
(case when result='passed' then 1 else 0 end) as passed, sum(case when result='exempt' then 1 else
0 end) as exempt, sum(case when result='failed' then 1 else 0 end) as failed, sum(case when
result='dependenciesNotMet' then 1 else 0 end) as unmet, sum(score) as score from (select audit_
entry, (instance->>'score')::float as score, instance->>'deviceID' as devid, instance->>'device'
as devtype, (case when instance->'domain'->>'type' in ('adom-global', 'global') then 'Global' when
instance->'domain'->'id' is not null then instance->'domain'->>'id' else 'Device' end) as scope,
instance->'recommendation' #>> '{}' as recommendation, instance->>'result' as result, (case
instance->>'result' when 'passed' then 'none' when 'exempt' then 'low' else instance->>'severity'
end) as severity, compliance, hipaa, hipaa_sub, hipaa_sub_control_id, title, fsbp_id, name,
description from (select hipaa as hipaa, hipaa_sub, hipaa_sub_control_id, t1.title, fsbp_id,
t2.name, null as description from (select split_part(t1.id, '(', 1) as hipaa,  substr(t1.id,
strpos(t1.id, '(')) as hipaa_sub, t1.id as hipaa_sub_control_id, t2.name as title, fsbp_id from
security_hipaa_fsbp_map t1 inner join security_hipaa_mdata t2 on t1.id= t2.id) t1 inner join
security_hipaa_mdata t2 on t1.hipaa = t2.id) t1 left join (select td.*, reporttype, audit_entry,
json_array_elements_text(compliances) as compliance, instance from (select dvid, reporttype, json_
build_object('name', audit_entry, 'desc', audit_result->>'description')::text as audit_entry,
(case when json_array_length(audit_result->'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
security-Rating-HIPAA-Sub-Control-Exempt-Stats-Recommendation	Security Rating HIPAA Sub Compliance Exempt Statistics Recommendation	

```
select
  hipaa,
  hipaa_sub,
  devtype,
  devid,
  compliance,
  severity,
  rtrim(
    to_char(
      sum(score),
      & #039;FM99999999D999'), '.') as score, result, recommendation from /*fabricStart*/(select
devid, devtype, scope, result, severity, compliance, hipaa, hipaa_sub, hipaa_sub_control_id, null
as asset_type, title, fsbp_id, name, description, recommendation, sum(case when result='passed'
then 1 else 0 end) as passed, sum(case when result='exempt' then 1 else 0 end) as exempt, sum(case
when result='failed' then 1 else 0 end) as failed, sum(case when result='dependenciesNotMet' then
1 else 0 end) as unmet, sum(score) as score from (select audit_entry, (instance->>'score')::float
as score, instance->>'deviceID' as devid, instance->>'device' as devtype, (case when instance-
>'domain'->>'type' in ('adom-global', 'global') then 'Global' when instance->'domain'->'id' is not
null then instance->'domain'->>'id' else 'Device' end) as scope, instance->'recommendation' #>> '
{}' as recommendation, instance->>'result' as result, (case instance->>'result' when 'passed' then
'none' when 'exempt' then 'low' else instance->>'severity' end) as severity, compliance, hipaa,
```

```
hipaa_sub, hipaa_sub_control_id, title, fsbp_id, name, description from (select hipaa as hipaa,
hipaa_sub, hipaa_sub_control_id, t1.title, fsbp_id, t2.name, null as description from (select
split_part(t1.id, '(', 1) as hipaa,  substr(t1.id, strpos(t1.id, '(')) as hipaa_sub, t1.id as
hipaa_sub_control_id, t2.name as title, fsbp_id from security_hipaa_fsbp_map t1 inner join
security_hipaa_mdata t2 on t1.id= t2.id) t1 inner join security_hipaa_mdata t2 on t1.hipaa =
t2.id) t1 left join (select td.*, reporttype, audit_entry, json_array_elements_text(compliances)
as compliance, instance from (select dvid, reporttype, json_build_object('name', audit_entry,
'desc', audit_result->>'description')::text as audit_entry, (case when json_array_length(audit_
result->'FSBP') = 0 then '[\
```

Dataset Name	Description	Log Category
shadowit-Total-Managed-vs-Unmanaged-Apps	Total Managed vs Unmanaged Cloud Apps	app-ctrl

```
select
  (
    case when action =& #039;pass' then 'Managed' else 'Unmanaged' end) as type, count(distinct
app) as total_num from ###(select $flex_timestamp as timestamp, app, siappid, filename, profile,
service, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
dstcountry, action, sum(case when cloudaction='upload' then coalesce(filesize, 0) else 0 end) as
upload_size, sum(case when cloudaction='download' or (cloudaction='others' and app like '%Video%')
then coalesce(filesize, 0) else 0 end) as download_size, sum(filesize) as filesize, count(*) as
sessions from $log-app-ctrl where $filter and siappid is not null and action in ('pass', 'block',
'reset') group by timestamp, app, siappid, filename, profile, service, action, user_src,
dstcountry order by sessions desc)### t1 inner join shadowit_application t2 on t1.siappid=t2.appid
group by type
```

Dataset Name	Description	Log Category
shadowit-Total-Managed-vs-Unmanaged-Users	Total Managed vs Unmanaged Cloud App Users	app-ctrl

```
select
  (
    case when action =& #039;pass' then 'Managed' else 'Unmanaged' end) as type, count(distinct
user_src) as total_num from ###(select $flex_timestamp as timestamp, app, siappid, filename,
profile, service, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
dstcountry, action, sum(case when cloudaction='upload' then coalesce(filesize, 0) else 0 end) as
upload_size, sum(case when cloudaction='download' or (cloudaction='others' and app like '%Video%')
then coalesce(filesize, 0) else 0 end) as download_size, sum(filesize) as filesize, count(*) as
sessions from $log-app-ctrl where $filter and siappid is not null and action in ('pass', 'block',
'reset') group by timestamp, app, siappid, filename, profile, service, action, user_src,
dstcountry order by sessions desc)### t1 inner join shadowit_application t2 on t1.siappid=t2.appid
group by type
```

Dataset Name	Description	Log Category
shadowit-Total-Data-Volume	Total Shadow IT Cloud App Data Volume	app-ctrl

```
select
  direction,
  volume
from
  (
    select
      unnest(traffic_direction) as direction,
      unnest(traffic_volume) as volume
    from
      (
        select
          array[ & #039;Download', 'Upload'] as traffic_direction, array[sum(download_size), sum
(upload_size)] as traffic_volume from ###(select $flex_timestamp as timestamp, app, siappid,
filename, profile, service, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as
user_src, dstcountry, action, sum(case when cloudaction='upload' then coalesce(filesize, 0) else 0
end) as upload_size, sum(case when cloudaction='download' or (cloudaction='others' and app like
'%Video%') then coalesce(filesize, 0) else 0 end) as download_size, sum(filesize) as filesize,
count(*) as sessions from $log-app-ctrl where $filter and siappid is not null and action in
('pass', 'block', 'reset') group by timestamp, app, siappid, filename, profile, service, action,
user_src, dstcountry order by sessions desc)### t1 inner join shadowit_application t2 on
t1.siappid=t2.appid) t) t where volume > 0
```

Dataset Name	Description	Log Category
dlp-Total-Allow-vs-Block-Actions	Total DLP Allow vs Block Actions	dlp

```
select
  action_type as type,
  sum(sessions) as sessions
from
  ###(select $flex_timestamp as timestamp, hostname, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry, (case when action = 'block' then 'Block'
else 'Allow' end) as action_type, severity, coalesce(sensitivity, 'Unclassified') as sen, service,
profile, (case when direction='incoming' then 'Download' else 'Upload' end) as traffic_direction,
filename, sum(filesize) as filesize, count(*) as sessions from $log-dlp where $filter and hostname
is not null and action in ('log-only', 'exempt', 'block') group by timestamp, hostname, user_src,
dstcountry, action_type, severity, sen, service, profile, traffic_direction, filename order by
sessions desc)### t group by type
```

Dataset Name	Description	Log Category
shadowit-Total-Appctrl-vs-Inline-CASB-Upload-Size	Total App Control vs Inline CASB Upload Size	app-ctrl

```
select
  type,
  upload_size
from
  (
```

```
    (
      select
        & #039;App Control' as type, sum(upload_size) as upload_size from ###(select $flex_
timestamp as timestamp, app, siappid, filename, profile, service, coalesce(nullifna(`user`),
nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry, action, sum(case when
cloudaction='upload' then coalesce(filesize, 0) else 0 end) as upload_size, sum(case when
cloudaction='download' or (cloudaction='others' and app like '%Video%') then coalesce(filesize, 0)
else 0 end) as download_size, sum(filesize) as filesize, count(*) as sessions from $log-app-ctrl
where $filter and siappid is not null and action in ('pass', 'block', 'reset') group by timestamp,
app, siappid, filename, profile, service, action, user_src, dstcountry order by sessions desc)###
t1 inner join shadowit_application t2 on t1.siappid=t2.appid group by type) union all (select
'Inline CASB' as type, sum(upload_size) as upload_size from ###(select saasname, srcip, dstip,
coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, sum((CASE WHEN
(logflag&2>0) THEN 1 ELSE 0 END)) as session_block, count(*) as sessions, sum(case when
accessctrl='upload' THEN coalesce(sentbyte, 0) ELSE 0 END) AS upload_size from $log-traffic where
$filter and (logflag&1>0) and saasname is not null group by saasname, srcip, dstip, user_src order
by sessions desc)### t group by type)) t where upload_size > 0
```

Dataset Name	Description	Log Category
shadowit-App-Actions-by-Session	Cloud App Actions by Session	app-ctrl

```
select
  (
    case when action =& #039;block' then 'Block' when action='reset' then 'Reset' else 'Allow'
end) as action, sum(sessions) as sessions from ###(select $flex_timestamp as timestamp, app,
siappid, filename, profile, service, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr
(`srcip`)) as user_src, dstcountry, action, sum(case when cloudaction='upload' then coalesce
(filesize, 0) else 0 end) as upload_size, sum(case when cloudaction='download' or
(cloudaction='others' and app like '%Video%') then coalesce(filesize, 0) else 0 end) as download_
size, sum(filesize) as filesize, count(*) as sessions from $log-app-ctrl where $filter and siappid
is not null and action in ('pass', 'block', 'reset') group by timestamp, app, siappid, filename,
profile, service, action, user_src, dstcountry order by sessions desc)### t1 inner join shadowit_
application t2 on t1.siappid=t2.appid group by action order by sessions desc
```

Dataset Name	Description	Log Category
shadowit-App-Categories-by-Session	Cloud App Categories by Session	app-ctrl

```
select
  attributes ->& #039;Information'->>'Category' as category, sum(sessions) as sessions from ###
(select $flex_timestamp as timestamp, app, siappid, filename, profile, service, coalesce(nullifna
(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry, action, sum(case when
cloudaction='upload' then coalesce(filesize, 0) else 0 end) as upload_size, sum(case when
cloudaction='download' or (cloudaction='others' and app like '%Video%') then coalesce(filesize, 0)
else 0 end) as download_size, sum(filesize) as filesize, count(*) as sessions from $log-app-ctrl
where $filter and siappid is not null and action in ('pass', 'block', 'reset') group by timestamp,
app, siappid, filename, profile, service, action, user_src, dstcountry order by sessions desc)###
```

```
t1 inner join shadowit_application t2 on t1.siappid=t2.appid group by category order by sessions
desc
```

Dataset Name	Description	Log Category
shadowit-App-Risk-Levels-by-Session	Cloud App Risk Levels by Session	app-ctrl

```
select
  (
    case when riskscore between 1
    and 15 then & #039;Low' when riskscore between 16 and 30 then 'Guarded' when riskscore between
31 and 50 then 'Elevated' when riskscore between 51 and 70 then 'High' when riskscore between 71
and 100 then 'Severe' else 'N/A' end) as risk_level, sum(sessions) as sessions from ###(select
$flex_timestamp as timestamp, app, siappid, filename, profile, service, coalesce(nullifna(`user`),
nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry, action, sum(case when
cloudaction='upload' then coalesce(filesize, 0) else 0 end) as upload_size, sum(case when
cloudaction='download' or (cloudaction='others' and app like '%Video%') then coalesce(filesize, 0)
else 0 end) as download_size, sum(filesize) as filesize, count(*) as sessions from $log-app-ctrl
where $filter and siappid is not null and action in ('pass', 'block', 'reset') group by timestamp,
app, siappid, filename, profile, service, action, user_src, dstcountry order by sessions desc)###
t1 inner join shadowit_application t2 on t1.siappid=t2.appid group by risk_level order by risk_
level desc
```

Dataset Name	Description	Log Category
shadowit-Top-Managed-Cloud-App-Users-by-Requests	Top Managed Cloud App Users by Requests	app-ctrl

```
select
  user_src,
  sum(sessions) as sessions
from
  ###(select $flex_timestamp as timestamp, app, siappid, filename, profile, service, coalesce
(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry, action, sum
(case when cloudaction='upload' then coalesce(filesize, 0) else 0 end) as upload_size, sum(case
when cloudaction='download' or (cloudaction='others' and app like '%Video%') then coalesce
(filesize, 0) else 0 end) as download_size, sum(filesize) as filesize, count(*) as sessions from
$log-app-ctrl where $filter and siappid is not null and action in ('pass', 'block', 'reset') group
by timestamp, app, siappid, filename, profile, service, action, user_src, dstcountry order by
sessions desc)### t1 inner join shadowit_application t2 on t1.siappid=t2.appid where action =
'pass' group by user_src order by sessions desc
```

Dataset Name	Description	Log Category
shadowit-Top-Unmanaged-Cloud-App-Users-by-Requests	Top Unmanaged Cloud App Users by Requests	app-ctrl

```
select
  user_src,
  sum(sessions) as sessions
from
  ###(select $flex_timestamp as timestamp, app, siappid, filename, profile, service, coalesce
(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry, action, sum
(case when cloudaction='upload' then coalesce(filesize, 0) else 0 end) as upload_size, sum(case
when cloudaction='download' or (cloudaction='others' and app like '%Video%') then coalesce
(filesize, 0) else 0 end) as download_size, sum(filesize) as filesize, count(*) as sessions from
$log-app-ctrl where $filter and siappid is not null and action in ('pass', 'block', 'reset') group
by timestamp, app, siappid, filename, profile, service, action, user_src, dstcountry order by
sessions desc)### t1 inner join shadowit_application t2 on t1.siappid=t2.appid where action !=
'pass' group by user_src order by sessions desc
```

Dataset Name	Description	Log Category
shadowit-Top-Cloud-App-Users-by-Risk-Score	Top Cloud App Users by Risk Score	app-ctrl

```
select
  user_src,
  sum(riskscore * sessions) as riskscore
from
  ###(select $flex_timestamp as timestamp, app, siappid, filename, profile, service, coalesce
(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry, action, sum
(case when cloudaction='upload' then coalesce(filesize, 0) else 0 end) as upload_size, sum(case
when cloudaction='download' or (cloudaction='others' and app like '%Video%') then coalesce
(filesize, 0) else 0 end) as download_size, sum(filesize) as filesize, count(*) as sessions from
$log-app-ctrl where $filter and siappid is not null and action in ('pass', 'block', 'reset') group
by timestamp, app, siappid, filename, profile, service, action, user_src, dstcountry order by
sessions desc)### t1 inner join shadowit_application t2 on t1.siappid=t2.appid group by user_src
order by riskscore desc
```

Dataset Name	Description	Log Category
shadowit-Malware-Types-by-Occurrences	Cloud App Malware Types by Occurrences	traffic

```
select
  malware_type,
  sum(sessions) as sessions
from
  (
    select
      (
        case when virus like & #039;Riskware%' then 'Spyware' when virus like 'Adware%' then
'Adware' else 'Virus' end) as malware_type, app, sum(sessions) as sessions from ###(select attack,
virus, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, (CASE
WHEN (logflag&128>0) THEN srcip ELSE dstip END) as victim, (CASE WHEN (logflag&128>0) THEN dstip
```

```
ELSE srcip END) as source, (CASE WHEN utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2 WHEN
utmaction='reset' THEN 3 ELSE 0 END) as action_flag, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0
END)) as session_block, count(*) as sessions from $log where $filter and (logflag&1>0) and
nullifna(app) is not null and (attack is not null or virus is not null) group by attack, virus,
app, user_src, victim, source, action_flag order by sessions desc)### t where virus is not null
group by malware_type, app order by sessions desc) t1 inner join shadowit_application t2 on
t1.app=t2.appname group by malware_type order by sessions desc
```

Dataset Name	Description	Log Category
shadowit-Malware-Actions-by-Occurrences	Cloud App Malware Actions by Occurrences	traffic

```
select
  (
    case when action_flag = 1 then & #039;Block' when action_flag=2 then 'Allow' else 'Reset' end)
as action, sum(sessions) as sessions from (select action_flag, app, sum(sessions) as sessions from
###(select attack, virus, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`))
as user_src, (CASE WHEN (logflag&128>0) THEN srcip ELSE dstip END) as victim, (CASE WHEN
(logflag&128>0) THEN dstip ELSE srcip END) as source, (CASE WHEN utmaction='block' THEN 1 WHEN
utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3 ELSE 0 END) as action_flag, sum((CASE WHEN
(logflag&2>0) THEN 1 ELSE 0 END)) as session_block, count(*) as sessions from $log where $filter
and (logflag&1>0) and nullifna(app) is not null and (attack is not null or virus is not null)
group by attack, virus, app, user_src, victim, source, action_flag order by sessions desc)### t
where virus is not null group by action_flag, app order by sessions desc) t1 inner join shadowit_
application t2 on t1.app=t2.appname where action_flag>0 group by action
```

Dataset Name	Description	Log Category
shadowit-Top-Malwares-by-Occurrences	Top Cloud App Malwares by Occurrences	traffic

```
select
  virus,
  (
    case when virus like & #039;Riskware%' then 'Spyware' when virus like 'Adware%' then 'Adware'
else 'Virus' end) as malware_type, sum(session_block) as session_block, sum(sessions)-sum(session_
block) as session_pass, sum(sessions) as sessions from (select virus, app, sum(session_block) as
session_block, sum(sessions) as sessions from ###(select attack, virus, app, coalesce(nullifna
(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, (CASE WHEN (logflag&128>0) THEN
srcip ELSE dstip END) as victim, (CASE WHEN (logflag&128>0) THEN dstip ELSE srcip END) as source,
(CASE WHEN utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3
ELSE 0 END) as action_flag, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) as session_block,
count(*) as sessions from $log where $filter and (logflag&1>0) and nullifna(app) is not null and
(attack is not null or virus is not null) group by attack, virus, app, user_src, victim, source,
action_flag order by sessions desc)### t where virus is not null group by virus, app order by
sessions desc) t1 inner join shadowit_application t2 on t1.app=t2.appname group by virus, malware_
type order by sessions desc
```

Dataset Name	Description	Log Category
shadowit-Top-Malware-Victims-by-Occurrences	Top Cloud App Malware Victims by Occurrences	traffic

```
select
  user_src,
  sum(sessions) as sessions
from
  (
    select
      user_src,
      app,
      sum(sessions) as sessions
    from
      ###(select attack, virus, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr
(`srcip`)) as user_src, (CASE WHEN (logflag&128>0) THEN srcip ELSE dstip END) as victim, (CASE
WHEN (logflag&128>0) THEN dstip ELSE srcip END) as source, (CASE WHEN utmaction='block' THEN 1
WHEN utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3 ELSE 0 END) as action_flag, sum((CASE
WHEN (logflag&2>0) THEN 1 ELSE 0 END)) as session_block, count(*) as sessions from $log where
$filter and (logflag&1>0) and nullifna(app) is not null and (attack is not null or virus is not
null) group by attack, virus, app, user_src, victim, source, action_flag order by sessions
desc)### t where virus is not null group by user_src, app order by sessions desc) t1 inner join
shadowit_application t2 on t1.app=t2.appname group by user_src order by sessions desc
```

Dataset Name	Description	Log Category
shadowit-Top-Managed-Apps-by-Risk	Top Managed Cloud Apps by Risk	app-ctrl

```
select
  (
    case when riskscore between 1
    and 15 then & #039;info' when riskscore between 16 and 30 then 'low' when riskscore between 31
and 50 then 'medium' when riskscore between 51 and 70 then 'high' when riskscore between 71 and
100 then 'citical' else 'info' end) as risk_level, riskscore, appname, attributes->'Information'-
>>'Category' as category, count(distinct user_src) as num_users, replace(right(left(attributes-
>>'Compliance', -1), -1), '``', '') as compliance from ###(select $flex_timestamp as timestamp,
app, siappid, filename, profile, service, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr
(`srcip`)) as user_src, dstcountry, action, sum(case when cloudaction='upload' then coalesce
(filesize, 0) else 0 end) as upload_size, sum(case when cloudaction='download' or
(cloudaction='others' and app like '%Video%') then coalesce(filesize, 0) else 0 end) as download_
size, sum(filesize) as filesize, count(*) as sessions from $log-app-ctrl where $filter and siappid
is not null and action in ('pass', 'block', 'reset') group by timestamp, app, siappid, filename,
profile, service, action, user_src, dstcountry order by sessions desc)### t1 inner join shadowit_
application t2 on t1.siappid=t2.appid where action='pass' group by risk_level, riskscore, appname,
category, compliance order by riskscore desc
```

Dataset Name	Description	Log Category
shadowit-Top-Unmanaged-Apps-by-Risk	Top Unmanaged Cloud Apps by Risk	app-ctrl

```
select
  (
    case when riskscore between 1
    and 15 then & #039;info' when riskscore between 16 and 30 then 'low' when riskscore between 31
and 50 then 'medium' when riskscore between 51 and 70 then 'high' when riskscore between 71 and
100 then 'citical' else 'info' end) as risk_level, riskscore, appname, attributes->'Information'-
>>'Category' as category, count(distinct user_src) as num_users, replace(right(left(attributes-
>>'Compliance', -1), -1), ``', '') as compliance from ###(select $flex_timestamp as timestamp,
app, siappid, filename, profile, service, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr
(`srcip`)) as user_src, dstcountry, action, sum(case when cloudaction='upload' then coalesce
(filesize, 0) else 0 end) as upload_size, sum(case when cloudaction='download' or
(cloudaction='others' and app like '%Video%') then coalesce(filesize, 0) else 0 end) as download_
size, sum(filesize) as filesize, count(*) as sessions from $log-app-ctrl where $filter and siappid
is not null and action in ('pass', 'block', 'reset') group by timestamp, app, siappid, filename,
profile, service, action, user_src, dstcountry order by sessions desc)### t1 inner join shadowit_
application t2 on t1.siappid=t2.appid where action!='pass' group by risk_level, riskscore,
appname, category, compliance order by riskscore desc
```

Dataset Name	Description	Log Category
shadowit-App-Vulnerability-Risk-Levels-by-Occurrences	Cloud App Vulnerability Risk Levels by Occurrences	traffic

```
select
  (
    case when riskscore between 1
    and 15 then & #039;info' when riskscore between 16 and 30 then 'low' when riskscore between 31
and 50 then 'medium' when riskscore between 51 and 70 then 'high' when riskscore between 71 and
100 then 'citical' else 'info' end) as severity, sum(sessions) as sessions from (select attack,
app, sum(sessions) as sessions from ###(select attack, virus, app, coalesce(nullifna(`user`),
nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, (CASE WHEN (logflag&128>0) THEN srcip ELSE
dstip END) as victim, (CASE WHEN (logflag&128>0) THEN dstip ELSE srcip END) as source, (CASE WHEN
utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3 ELSE 0 END)
as action_flag, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) as session_block, count(*) as
sessions from $log where $filter and (logflag&1>0) and nullifna(app) is not null and (attack is
not null or virus is not null) group by attack, virus, app, user_src, victim, source, action_flag
order by sessions desc)### t where attack is not null group by attack, app order by sessions desc)
t1 inner join shadowit_application t2 on t1.app=t2.appname left join (select name, id, cve, vuln_
type from ips_mdata) t3 on t1.attack=t3.name group by severity order by severity desc
```

Dataset Name	Description	Log Category
shadowit-App-Vulnerability-Types-by-Occurrences	Cloud App Vulnerability Types by Occurrences	traffic

```
select
  vuln_type,
  sum(sessions) as sessions
from
  (
    select
      attack,
      app,
      sum(sessions) as sessions
    from
      ###(select attack, virus, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr
(`srcip`)) as user_src, (CASE WHEN (logflag&128>0) THEN srcip ELSE dstip END) as victim, (CASE
WHEN (logflag&128>0) THEN dstip ELSE srcip END) as source, (CASE WHEN utmaction='block' THEN 1
WHEN utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3 ELSE 0 END) as action_flag, sum((CASE
WHEN (logflag&2>0) THEN 1 ELSE 0 END)) as session_block, count(*) as sessions from $log where
$filter and (logflag&1>0) and nullifna(app) is not null and (attack is not null or virus is not
null) group by attack, virus, app, user_src, victim, source, action_flag order by sessions
desc)### t where attack is not null group by attack, app order by sessions desc) t1 inner join
shadowit_application t2 on t1.app=t2.appname left join (select name, id, cve, vuln_type from ips_
mdata) t3 on t1.attack=t3.name where vuln_type is not null group by vuln_type order by sessions
desc
```

Dataset Name	Description	Log Category
shadowit-App-Vulnerability-Actions-by-Occurrences	Cloud App Vulnerability Actions by Occurrences	traffic

```
select
  (
    case when action_flag = 1 then & #039;Block' when action_flag=2 then 'Allow' else 'Reset' end)
as action, sum(sessions) as sessions from (select attack, app, action_flag, sum(sessions) as
sessions from ###(select attack, virus, app, coalesce(nullifna(`user`), nullifna(`unauthuser`),
ipstr(`srcip`)) as user_src, (CASE WHEN (logflag&128>0) THEN srcip ELSE dstip END) as victim,
(CASE WHEN (logflag&128>0) THEN dstip ELSE srcip END) as source, (CASE WHEN utmaction='block' THEN
1 WHEN utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3 ELSE 0 END) as action_flag, sum
((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) as session_block, count(*) as sessions from $log
where $filter and (logflag&1>0) and nullifna(app) is not null and (attack is not null or virus is
not null) group by attack, virus, app, user_src, victim, source, action_flag order by sessions
desc)### t where attack is not null group by attack, app, action_flag order by sessions desc) t1
inner join shadowit_application t2 on t1.app=t2.appname left join (select name, id, cve, vuln_type
from ips_mdata) t3 on t1.attack=t3.name where action_flag>0 group by action order by sessions desc
```

Dataset Name	Description	Log Category
shadowit-Top-App-Vulnerabilities-by-Severity	Top App Vulnerabilities by Severity	traffic

```
select
  attack,
```

```
  id as attackid,
  vuln_type,
  cve,
  (
    case when riskscore between 1
    and 15 then & #039;info' when riskscore between 16 and 30 then 'low' when riskscore between 31
and 50 then 'medium' when riskscore between 51 and 70 then 'high' when riskscore between 71 and
100 then 'citical' else 'info' end) as severity, count(distinct victim) as victims, count(distinct
source) as sources, sum(session_block) as session_block, sum(sessions)-sum(session_block) as
session_pass, sum(sessions) as sessions from (select attack, app, victim, source, sum(session_
block) as session_block, sum(sessions) as sessions from ###(select attack, virus, app, coalesce
(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, (CASE WHEN (logflag&128>0)
THEN srcip ELSE dstip END) as victim, (CASE WHEN (logflag&128>0) THEN dstip ELSE srcip END) as
source, (CASE WHEN utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2 WHEN utmaction='reset'
THEN 3 ELSE 0 END) as action_flag, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) as session_
block, count(*) as sessions from $log where $filter and (logflag&1>0) and nullifna(app) is not
null and (attack is not null or virus is not null) group by attack, virus, app, user_src, victim,
source, action_flag order by sessions desc)### t where attack is not null group by attack, app,
victim, source order by sessions desc) t1 inner join shadowit_application t2 on t1.app=t2.appname
left join (select name, id, cve, vuln_type from ips_mdata) t3 on t1.attack=t3.name group by
attack, attackid, vuln_type, severity, cve order by severity desc, sessions desc
```

Dataset Name	Description	Log Category
shadowit-Top-Managed-Apps-by-Upload-Size-Timeline	Top Managed Cloud Apps by Upload Size Timeline	app-ctrl

```
select
  hodex,
  t1.appname,
  t1.upload_size
from
  (
    select
      $flex_timestamp(timestamp) as hodex,
      appname,
      sum(upload_size) as upload_size
    from
      ###(select $flex_timestamp as timestamp, app, siappid, filename, profile, service, coalesce
(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry, action, sum
(case when cloudaction='upload' then coalesce(filesize, 0) else 0 end) as upload_size, sum(case
when cloudaction='download' or (cloudaction='others' and app like '%Video%') then coalesce
(filesize, 0) else 0 end) as download_size, sum(filesize) as filesize, count(*) as sessions from
$log-app-ctrl where $filter and siappid is not null and action in ('pass', 'block', 'reset') group
by timestamp, app, siappid, filename, profile, service, action, user_src, dstcountry order by
sessions desc)### t1 inner join shadowit_application t2 on t1.siappid=t2.appid where action =
'pass' group by hodex, appname having sum(upload_size)>0 order by hodex) t1 inner join (select
appname, sum(upload_size) as upload_size from ###(select $flex_timestamp as timestamp, app,
siappid, filename, profile, service, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr
(`srcip`)) as user_src, dstcountry, action, sum(case when cloudaction='upload' then coalesce
(filesize, 0) else 0 end) as upload_size, sum(case when cloudaction='download' or
```

```
(cloudaction='others' and app like '%Video%') then coalesce(filesize, 0) else 0 end) as download_
size, sum(filesize) as filesize, count(*) as sessions from $log-app-ctrl where $filter and siappid
is not null and action in ('pass', 'block', 'reset') group by timestamp, app, siappid, filename,
profile, service, action, user_src, dstcountry order by sessions desc)### t1 inner join shadowit_
application t2 on t1.siappid=t2.appid where action = 'pass' group by appname order by upload_size
desc limit $ddown-top) t2 on t1.appname=t2.appname order by hodex
```

Dataset Name	Description	Log Category
shadowit-Top-Managed-Apps-by-Upload-Size	Top Managed Cloud Apps by Upload Size	app-ctrl

```
select
  (
    case when riskscore between 1
    and 15 then & #039;info' when riskscore between 16 and 30 then 'low' when riskscore between 31
and 50 then 'medium' when riskscore between 51 and 70 then 'high' when riskscore between 71 and
100 then 'citical' else 'info' end) as risk_level, riskscore, appname, attributes->'Information'-
>>'Category' as category, count(distinct user_src) as num_users, replace(right(left(attributes-
>>'Compliance', -1), -1), '`', '') as compliance, sum(upload_size) as upload_size from ###(select
$flex_timestamp as timestamp, app, siappid, filename, profile, service, coalesce(nullifna(`user`),
nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry, action, sum(case when
cloudaction='upload' then coalesce(filesize, 0) else 0 end) as upload_size, sum(case when
cloudaction='download' or (cloudaction='others' and app like '%Video%') then coalesce(filesize, 0)
else 0 end) as download_size, sum(filesize) as filesize, count(*) as sessions from $log-app-ctrl
where $filter and siappid is not null and action in ('pass', 'block', 'reset') group by timestamp,
app, siappid, filename, profile, service, action, user_src, dstcountry order by sessions desc)###
t1 inner join shadowit_application t2 on t1.siappid=t2.appid where action = 'pass' group by risk_
level, riskscore, appname, category, compliance having sum(upload_size)>0 order by upload_size
desc
```

Dataset Name	Description	Log Category
shadowit-Top-Managed-Apps-by-User-Num-Timeline	Top Managed Cloud Apps by Total Users Timeline	app-ctrl

```
select
  hodex,
  t1.appname,
  t1.num_users
from
  (
    select
      $flex_timestamp(timestamp) as hodex,
      appname,
      count(distinct user_src) as num_users
    from
      ###(select $flex_timestamp as timestamp, app, siappid, filename, profile, service, coalesce
(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry, action, sum
```

```
(case when cloudaction='upload' then coalesce(filesize, 0) else 0 end) as upload_size, sum(case
when cloudaction='download' or (cloudaction='others' and app like '%Video%') then coalesce
(filesize, 0) else 0 end) as download_size, sum(filesize) as filesize, count(*) as sessions from
$log-app-ctrl where $filter and siappid is not null and action in ('pass', 'block', 'reset') group
by timestamp, app, siappid, filename, profile, service, action, user_src, dstcountry order by
sessions desc)### t1 inner join shadowit_application t2 on t1.siappid=t2.appid where action =
'pass' group by hodex, appname order by hodex) t1 inner join (select appname, count(distinct user_
src) as num_users from ###(select $flex_timestamp as timestamp, app, siappid, filename, profile,
service, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
dstcountry, action, sum(case when cloudaction='upload' then coalesce(filesize, 0) else 0 end) as
upload_size, sum(case when cloudaction='download' or (cloudaction='others' and app like '%Video%')
then coalesce(filesize, 0) else 0 end) as download_size, sum(filesize) as filesize, count(*) as
sessions from $log-app-ctrl where $filter and siappid is not null and action in ('pass', 'block',
'reset') group by timestamp, app, siappid, filename, profile, service, action, user_src,
dstcountry order by sessions desc)### t1 inner join shadowit_application t2 on t1.siappid=t2.appid
where action = 'pass' group by appname order by num_users desc limit $ddown-top) t2 on
t1.appname=t2.appname order by hodex
```

Dataset Name	Description	Log Category
shadowit-Top-Managed-Apps-by-User-Num	Top Managed Cloud Apps by Total Users	app-ctrl

```
select
  (
    case when riskscore between 1
    and 15 then & #039;info' when riskscore between 16 and 30 then 'low' when riskscore between 31
and 50 then 'medium' when riskscore between 51 and 70 then 'high' when riskscore between 71 and
100 then 'citical' else 'info' end) as risk_level, riskscore, appname, attributes->'Information'-
>>'Category' as category, count(distinct user_src) as num_users, replace(right(left(attributes-
>>'Compliance', -1), -1), '``', '') as compliance from ###(select $flex_timestamp as timestamp,
app, siappid, filename, profile, service, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr
(`srcip`)) as user_src, dstcountry, action, sum(case when cloudaction='upload' then coalesce
(filesize, 0) else 0 end) as upload_size, sum(case when cloudaction='download' or
(cloudaction='others' and app like '%Video%') then coalesce(filesize, 0) else 0 end) as download_
size, sum(filesize) as filesize, count(*) as sessions from $log-app-ctrl where $filter and siappid
is not null and action in ('pass', 'block', 'reset') group by timestamp, app, siappid, filename,
profile, service, action, user_src, dstcountry order by sessions desc)### t1 inner join shadowit_
application t2 on t1.siappid=t2.appid where action = 'pass' group by risk_level, riskscore,
appname, category, compliance order by num_users desc
```

Dataset Name	Description	Log Category
dlp-Total-DLP-Events-by-Severity	Total DLP Events by Severity	dlp

```
select
  severity,
  sum(sessions) as sessions
from
```

```
  (
    (
      select
        severity,
        sum(sessions) as sessions
      from
        ###(select $flex_timestamp as timestamp, hostname, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry, (case when action = 'block' then 'Block'
else 'Allow' end) as action_type, severity, coalesce(sensitivity, 'Unclassified') as sen, service,
profile, (case when direction='incoming' then 'Download' else 'Upload' end) as traffic_direction,
filename, sum(filesize) as filesize, count(*) as sessions from $log-dlp where $filter and hostname
is not null and action in ('log-only', 'exempt', 'block') group by timestamp, hostname, user_src,
dstcountry, action_type, severity, sen, service, profile, traffic_direction, filename order by
sessions desc)### t group by severity) union all (select (case when riskscore between 1 and 15
then 'info' when riskscore between 16 and 30 then 'low' when riskscore between 31 and 50 then
'medium' when riskscore between 51 and 70 then 'high' when riskscore between 71 and 100 then
'citical' else 'info' end)::fgt_enum_severity as severity, sum(sessions) as sessions from ###
(select $flex_timestamp as timestamp, app, siappid, filename, profile, service, coalesce(nullifna
(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry, action, sum(case when
cloudaction='upload' then coalesce(filesize, 0) else 0 end) as upload_size, sum(case when
cloudaction='download' or (cloudaction='others' and app like '%Video%') then coalesce(filesize, 0)
else 0 end) as download_size, sum(filesize) as filesize, count(*) as sessions from $log-app-ctrl
where $filter and siappid is not null and action in ('pass', 'block', 'reset') group by timestamp,
app, siappid, filename, profile, service, action, user_src, dstcountry order by sessions desc)###
t1 inner join shadowit_application t2 on t1.siappid=t2.appid where filename is not null group by
severity)) t group by severity order by severity desc
```

Dataset Name	Description	Log Category
dlp-Total-DLP-Events-by-Sensitivity	Total DLP Events by Sensitivity	dlp

```
select
  sensitivity,
  sum(sessions) as sessions
from
  (
    (
      select
        sen as sensitivity,
        sum(sessions) as sessions
      from
        ###(select $flex_timestamp as timestamp, hostname, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry, (case when action = 'block' then 'Block'
else 'Allow' end) as action_type, severity, coalesce(sensitivity, 'Unclassified') as sen, service,
profile, (case when direction='incoming' then 'Download' else 'Upload' end) as traffic_direction,
filename, sum(filesize) as filesize, count(*) as sessions from $log-dlp where $filter and hostname
is not null and action in ('log-only', 'exempt', 'block') group by timestamp, hostname, user_src,
dstcountry, action_type, severity, sen, service, profile, traffic_direction, filename order by
sessions desc)### t group by sensitivity) union all (select 'Unclassified' as sensitivity, sum
(sessions) as sessions from ###(select $flex_timestamp as timestamp, app, siappid, filename,
```

```
profile, service, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
dstcountry, action, sum(case when cloudaction='upload' then coalesce(filesize, 0) else 0 end) as
upload_size, sum(case when cloudaction='download' or (cloudaction='others' and app like '%Video%')
then coalesce(filesize, 0) else 0 end) as download_size, sum(filesize) as filesize, count(*) as
sessions from $log-app-ctrl where $filter and siappid is not null and action in ('pass', 'block',
'reset') group by timestamp, app, siappid, filename, profile, service, action, user_src,
dstcountry order by sessions desc)### t1 inner join shadowit_application t2 on t1.siappid=t2.appid
where filename is not null)) t group by sensitivity order by sessions desc
```

Dataset Name	Description	Log Category
dlp-Total-DLP-Events-by-Action	Total DLP Events by Action	dlp

```
select
  action_type,
  sum(sessions) as sessions
from
  (
    (
      select
        action_type,
        sum(sessions) as sessions
      from
        ###(select $flex_timestamp as timestamp, hostname, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry, (case when action = 'block' then 'Block'
else 'Allow' end) as action_type, severity, coalesce(sensitivity, 'Unclassified') as sen, service,
profile, (case when direction='incoming' then 'Download' else 'Upload' end) as traffic_direction,
filename, sum(filesize) as filesize, count(*) as sessions from $log-dlp where $filter and hostname
is not null and action in ('log-only', 'exempt', 'block') group by timestamp, hostname, user_src,
dstcountry, action_type, severity, sen, service, profile, traffic_direction, filename order by
sessions desc)### t group by action_type) union all (select (case when action='pass' then 'Allow'
else 'Block' end) as action_type, sum(sessions) as sessions from ###(select $flex_timestamp as
timestamp, app, siappid, filename, profile, service, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry, action, sum(case when
cloudaction='upload' then coalesce(filesize, 0) else 0 end) as upload_size, sum(case when
cloudaction='download' or (cloudaction='others' and app like '%Video%') then coalesce(filesize, 0)
else 0 end) as download_size, sum(filesize) as filesize, count(*) as sessions from $log-app-ctrl
where $filter and siappid is not null and action in ('pass', 'block', 'reset') group by timestamp,
app, siappid, filename, profile, service, action, user_src, dstcountry order by sessions desc)###
t1 inner join shadowit_application t2 on t1.siappid=t2.appid where filename is not null group by
action_type)) t group by action_type
```

Dataset Name	Description	Log Category
dlp-Top-DLP-Events-by-Severity	Top DLP Events by Severity	dlp

```
select
  severity,
  hostname,
```

```
  user_src,
  filename,
  sum(bytes) as bytes,
  sensitivity,
  service,
  profile,
  sum(session_pass) as session_pass,
  sum(session_block) as session_block,
  sum(sessions) as sessions
from
  (
    (
      select
        severity,
        hostname,
        user_src,
        filename,
        sum(filesize) as bytes,
        sen as sensitivity,
        service,
        profile,
        sum(
          case when action_type = & #039;Allow' then sessions else 0 end) as session_pass, sum
(case when action_type = 'Block' then sessions else 0 end) as session_block, sum(sessions) as
sessions from ###(select $flex_timestamp as timestamp, hostname, coalesce(nullifna(`user`),
nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry, (case when action = 'block' then
'Block' else 'Allow' end) as action_type, severity, coalesce(sensitivity, 'Unclassified') as sen,
service, profile, (case when direction='incoming' then 'Download' else 'Upload' end) as traffic_
direction, filename, sum(filesize) as filesize, count(*) as sessions from $log-dlp where $filter
and hostname is not null and action in ('log-only', 'exempt', 'block') group by timestamp,
hostname, user_src, dstcountry, action_type, severity, sen, service, profile, traffic_direction,
filename order by sessions desc)### t group by severity, hostname, user_src, filename,
sensitivity, service, profile) union all (select (case when riskscore between 1 and 15 then 'info'
when riskscore between 16 and 30 then 'low' when riskscore between 31 and 50 then 'medium' when
riskscore between 51 and 70 then 'high' when riskscore between 71 and 100 then 'citical' else
'info' end)::fgt_enum_severity as severity, app as hostname, user_src, filename, sum(filesize) as
bytes, 'Unclassified' as sensitivity, service, null as profile, sum((case when action='pass' then
sessions else 0 end)) as session_pass, sum((case when action!='pass' then sessions else 0 end)) as
session_block, sum(sessions) as sessions from ###(select $flex_timestamp as timestamp, app,
siappid, filename, profile, service, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr
(`srcip`)) as user_src, dstcountry, action, sum(case when cloudaction='upload' then coalesce
(filesize, 0) else 0 end) as upload_size, sum(case when cloudaction='download' or
(cloudaction='others' and app like '%Video%') then coalesce(filesize, 0) else 0 end) as download_
size, sum(filesize) as filesize, count(*) as sessions from $log-app-ctrl where $filter and siappid
is not null and action in ('pass', 'block', 'reset') group by timestamp, app, siappid, filename,
profile, service, action, user_src, dstcountry order by sessions desc)### t1 inner join shadowit_
application t2 on t1.siappid=t2.appid where filename is not null group by severity, hostname,
user_src, filename, sensitivity, service)) t group by severity, hostname, user_src, filename,
sensitivity, service, profile order by severity desc
```

Dataset Name	Description	Log Category
shadowit-Top-Inline-CASB-Apps-by-Upload-Size	Top Inline CASB Apps by Upload Size	traffic

```
select
  saasname,
  srcip,
  dstip,
  user_src,
  sum(session_block) as session_block,
  sum(sessions)- sum(session_block) as session_pass,
  sum(sessions) as sessions,
  sum(upload_size) as upload_size
from
  ###(select saasname, srcip, dstip, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr
(`srcip`)) as user_src, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) as session_block, count
(*) as sessions, sum(case when accessctrl='upload' THEN coalesce(sentbyte, 0) ELSE 0 END) AS
upload_size from $log-traffic where $filter and (logflag&1>0) and saasname is not null group by
saasname, srcip, dstip, user_src order by sessions desc)### t group by saasname, srcip, dstip,
user_src order by sessions desc
```

Dataset Name	Description	Log Category
OT-Asset-OS-by-Count	OT Zone OS by Asset Count	traffic

```
select
  (
    case when osname is null then & #039;Unknown OS' else osname end) as osname, count(distinct
t1.ep_id) as total_num from ###(select $flex_timestamp as timestamp, dvid, epid as ep_id, srcip,
dstip, osname, proto, dstport, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr
(`srcip`)) as user_src, (CASE WHEN utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2 WHEN
utmaction='reset' THEN 3 ELSE 0 END) as action_flag, sum(crscore%65536) as scores, sum(coalesce
(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0
END)) AS session_block, count(*) as sessions from $log where $filter and (logflag&1>0) and
epid>1024 group by timestamp, dvid, ep_id, srcip, dstip, osname, proto, dstport, app, user_src,
action_flag order by bandwidth desc)### t1 inner join (select epid, min(purduelevel::float/10) as
purduelevel from $ADOM_ENDPOINT ep where purduelevel in (20, 30, 35) group by epid) t2 on t1.ep_
id=t2.epid inner join app_mdata t3 on t1.app=t3.name group by osname order by total_num desc
```

Dataset Name	Description	Log Category
IT-Asset-OS-by-Count	IT Zone OS by Asset Count	traffic

```
select
  (
    case when osname is null then & #039;Unknown OS' else osname end) as osname, count(distinct
t1.ep_id) as total_num from ###(select $flex_timestamp as timestamp, dvid, epid as ep_id, srcip,
```

```
dstip, osname, proto, dstport, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr
(`srcip`)) as user_src, (CASE WHEN utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2 WHEN
utmaction='reset' THEN 3 ELSE 0 END) as action_flag, sum(crscore%65536) as scores, sum(coalesce
(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0
END)) AS session_block, count(*) as sessions from $log where $filter and (logflag&1>0) and
epid>1024 group by timestamp, dvid, ep_id, srcip, dstip, osname, proto, dstport, app, user_src,
action_flag order by bandwidth desc)### t1 inner join (select epid, min(purduelevel::float/10) as
purduelevel from $ADOM_ENDPOINT ep where purduelevel in (40, 50) group by epid) t2 on t1.ep_
id=t2.epid inner join app_mdata t3 on t1.app=t3.name group by osname order by total_num desc
```

Dataset Name	Description	Log Category
OT-Application-Vulnerabilities-by-Risk-Level	OT Zone Application Vulnerabilities by Risk Level	attack

```
select
  severity,
  (
    case when severity =& #039;critical' then 5 when severity='high' then 4 when severity='medium'
then 3 when severity='low' then 2 when severity='info' then 1 else 0 end) as severity_number,
count(distinct t1.attack) as totalnum from ###(select $flex_timestamp as timestamp, dvid, (CASE
WHEN direction='incoming' THEN epid ELSE dstepid END) as ep_id, attack, attackid, severity,
direction, dstip, srcip, srccountry, (case when action in ('clear_session', 'pass_session') then 1
when action in ('reset', 'reset_client', 'reset_server') then 2 else 3 end) as action_flag, count
(*) as totalnum from $log-attack where $filter and nullifna(attack) is not null and severity is
not null and (epid>1024 or dstepid>1024) group by timestamp, dvid, ep_id, attack, attackid,
severity, direction, dstip, srcip, srccountry, action_flag order by totalnum desc)### t1 inner
join (select epid, min(purduelevel::float/10) as purduelevel from $ADOM_ENDPOINT ep where
purduelevel in (20, 30, 35) group by epid) t2 on t1.ep_id=t2.epid left join (select name, cve,
vuln_type from ips_mdata) t3 on t1.attack=t3.name group by severity, severity_number order by
severity_number desc
```

Dataset Name	Description	Log Category
OT-Application-Vulnerabilities-by-Type	OT Zone Application Vulnerabilities by Type	attack

```
select
  vuln_type,
  count(distinct t1.attack) as totalnum
from
  ###(select $flex_timestamp as timestamp, dvid, (CASE WHEN direction='incoming' THEN epid ELSE
dstepid END) as ep_id, attack, attackid, severity, direction, dstip, srcip, srccountry, (case when
action in ('clear_session', 'pass_session') then 1 when action in ('reset', 'reset_client',
'reset_server') then 2 else 3 end) as action_flag, count(*) as totalnum from $log-attack where
$filter and nullifna(attack) is not null and severity is not null and (epid>1024 or dstepid>1024)
group by timestamp, dvid, ep_id, attack, attackid, severity, direction, dstip, srcip, srccountry,
action_flag order by totalnum desc)### t1 inner join (select epid, min(purduelevel::float/10) as
purduelevel from $ADOM_ENDPOINT ep where purduelevel in (20, 30, 35) group by epid) t2 on t1.ep_
```

```
id=t2.epid left join (select name, cve, vuln_type from ips_mdata) t3 on t1.attack=t3.name where
vuln_type is not null group by vuln_type order by totalnum desc
```

Dataset Name	Description	Log Category
OT-Application-Vulnerabilities-by-Action	OT Zone Application Vulnerabilities by Action	attack

```
select
  (
    case when action_flag = 1 then & #039;Allow' when action_flag=2 then 'Reset' else 'Block' end)
as action, count(distinct t1.attack) as totalnum from ###(select $flex_timestamp as timestamp,
dvid, (CASE WHEN direction='incoming' THEN epid ELSE dstepid END) as ep_id, attack, attackid,
severity, direction, dstip, srcip, srccountry, (case when action in ('clear_session', 'pass_
session') then 1 when action in ('reset', 'reset_client', 'reset_server') then 2 else 3 end) as
action_flag, count(*) as totalnum from $log-attack where $filter and nullifna(attack) is not null
and severity is not null and (epid>1024 or dstepid>1024) group by timestamp, dvid, ep_id, attack,
attackid, severity, direction, dstip, srcip, srccountry, action_flag order by totalnum desc)### t1
inner join (select epid, min(purduelevel::float/10) as purduelevel from $ADOM_ENDPOINT ep where
purduelevel in (20, 30, 35) group by epid) t2 on t1.ep_id=t2.epid left join (select name, cve,
vuln_type from ips_mdata) t3 on t1.attack=t3.name group by action order by totalnum desc
```

Dataset Name	Description	Log Category
OT-Top-Application-Vulnerability-by-Risk-Level	Top OT Zone Application Vulnerabilities by Risk Level	attack

```
select
  attack,
  attackid,
  vuln_type,
  cve,
  (
    case when severity =& #039;critical' then 5 when severity='high' then 4 when severity='medium'
then 3 when severity='low' then 2 when severity='info' then 1 else 0 end) as severity_number,
count(distinct (CASE WHEN direction='incoming' THEN srcip ELSE dstip END)) as victims, count
(distinct (CASE WHEN direction='incoming' THEN dstip ELSE srcip END)) as sources, sum(case when
action_flag=3 then totalnum else 0 end) as total_block, sum(case when action_flag!=3 then totalnum
else 0 end) as total_allow, sum(totalnum) as totalnum from ###(select $flex_timestamp as
timestamp, dvid, (CASE WHEN direction='incoming' THEN epid ELSE dstepid END) as ep_id, attack,
attackid, severity, direction, dstip, srcip, srccountry, (case when action in ('clear_session',
'pass_session') then 1 when action in ('reset', 'reset_client', 'reset_server') then 2 else 3 end)
as action_flag, count(*) as totalnum from $log-attack where $filter and nullifna(attack) is not
null and severity is not null and (epid>1024 or dstepid>1024) group by timestamp, dvid, ep_id,
attack, attackid, severity, direction, dstip, srcip, srccountry, action_flag order by totalnum
desc)### t1 inner join (select epid, min(purduelevel::float/10) as purduelevel from $ADOM_ENDPOINT
ep where purduelevel in (20, 30, 35) group by epid) t2 on t1.ep_id=t2.epid left join (select name,
cve, vuln_type from ips_mdata) t3 on t1.attack=t3.name group by attack, attackid, vuln_type,
severity_number, cve order by severity_number desc, totalnum desc
```

Dataset Name	Description	Log Category
OT-High-Risk-Apps-by-Risk-Level	OT Zone High Risk Applications by Risk Level	traffic

```
select
  & #039;Risk' || risk as severity, risk as d_risk, count(distinct app) as sessions from ###
(select $flex_timestamp as timestamp, dvid, epid as ep_id, srcip, dstip, osname, proto, dstport,
app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, (CASE WHEN
utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3 ELSE 0 END)
as action_flag, sum(crscore%65536) as scores, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as
bandwidth, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block, count(*) as sessions
from $log where $filter and (logflag&1>0) and epid>1024 group by timestamp, dvid, ep_id, srcip,
dstip, osname, proto, dstport, app, user_src, action_flag order by bandwidth desc)### t1 inner
join (select epid, min(purduelevel::float/10) as purduelevel from $ADOM_ENDPOINT ep where
purduelevel in (20, 30, 35) group by epid) t2 on t1.ep_id=t2.epid inner join app_mdata t3 on
t1.app=t3.name where risk>='2' group by severity, d_risk order by sessions desc
```

Dataset Name	Description	Log Category
OT-High-Risk-Apps-by-Category	OT Zone High Risk Applications by Category	traffic

```
select
  (
    case when proto = 6 then & #039;TCP' when proto=17 then 'UDP' else 'N/A' end) as protocol,
count(distinct app) as sessions from ###(select $flex_timestamp as timestamp, dvid, epid as ep_id,
srcip, dstip, osname, proto, dstport, app, coalesce(nullifna(`user`), nullifna(`unauthuser`),
ipstr(`srcip`)) as user_src, (CASE WHEN utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2
WHEN utmaction='reset' THEN 3 ELSE 0 END) as action_flag, sum(crscore%65536) as scores, sum
(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum((CASE WHEN (logflag&2>0) THEN 1
ELSE 0 END)) AS session_block, count(*) as sessions from $log where $filter and (logflag&1>0) and
epid>1024 group by timestamp, dvid, ep_id, srcip, dstip, osname, proto, dstport, app, user_src,
action_flag order by bandwidth desc)### t1 inner join (select epid, min(purduelevel::float/10) as
purduelevel from $ADOM_ENDPOINT ep where purduelevel in (20, 30, 35) group by epid) t2 on t1.ep_
id=t2.epid inner join app_mdata t3 on t1.app=t3.name where risk>='2' group by protocol order by
sessions desc
```

Dataset Name	Description	Log Category
OT-High-Risk-Apps-by-Action	OT Zone High Risk Applications by Action	traffic

```
select
  (
    case when action_flag = 1 then & #039;Block' when action_flag=2 then 'Allow' else 'Reset' end)
as action, count(distinct app) as sessions from ###(select $flex_timestamp as timestamp, dvid,
epid as ep_id, srcip, dstip, osname, proto, dstport, app, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, (CASE WHEN utmaction='block' THEN 1 WHEN
utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3 ELSE 0 END) as action_flag, sum
(crscore%65536) as scores, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum
```

```
((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block, count(*) as sessions from $log
where $filter and (logflag&1>0) and epid>1024 group by timestamp, dvid, ep_id, srcip, dstip,
osname, proto, dstport, app, user_src, action_flag order by bandwidth desc)### t1 inner join
(select epid, min(purduelevel::float/10) as purduelevel from $ADOM_ENDPOINT ep where purduelevel
in (20, 30, 35) group by epid) t2 on t1.ep_id=t2.epid inner join app_mdata t3 on t1.app=t3.name
where risk>='2' and action_flag>0 group by action order by sessions desc
```

Dataset Name	Description	Log Category
OT-Top-High-Risk-Apps-by-Risk	Top OT Zone High Risk Applications by Risk Level	traffic

```
select
  risk as d_risk,
  name,
  max(
    (
      case when proto = 6 then & #039;TCP' || dstport when proto=17 then 'UDP' || dstport else
'N/A' end)) as port, max(srcip) as asset, sum(bandwidth) as bandwidth, sum(session_block) as
session_block, sum(sessions)-sum(session_block) as session_pass, sum(sessions) as sessions from
###(select $flex_timestamp as timestamp, dvid, epid as ep_id, srcip, dstip, osname, proto,
dstport, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
(CASE WHEN utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3
ELSE 0 END) as action_flag, sum(crscore%65536) as scores, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block,
count(*) as sessions from $log where $filter and (logflag&1>0) and epid>1024 group by timestamp,
dvid, ep_id, srcip, dstip, osname, proto, dstport, app, user_src, action_flag order by bandwidth
desc)### t1 inner join (select epid, min(purduelevel::float/10) as purduelevel from $ADOM_ENDPOINT
ep where purduelevel in (20, 30, 35) group by epid) t2 on t1.ep_id=t2.epid inner join app_mdata t3
on t1.app=t3.name where risk>='2' group by d_risk, name order by d_risk desc, bandwidth desc
```

Dataset Name	Description	Log Category
OT-Top-Asset-by-Threat-Score	Top OT Zone Assets by Threat Score	traffic

```
select
  srcip,
  & #039;Level ' || min(purduelevel) as purduelevel, sum(scores) as scores from ###(select $flex_
timestamp as timestamp, dvid, epid as ep_id, srcip, dstip, osname, proto, dstport, app, coalesce
(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, (CASE WHEN
utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3 ELSE 0 END)
as action_flag, sum(crscore%65536) as scores, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as
bandwidth, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block, count(*) as sessions
from $log where $filter and (logflag&1>0) and epid>1024 group by timestamp, dvid, ep_id, srcip,
dstip, osname, proto, dstport, app, user_src, action_flag order by bandwidth desc)### t1 inner
join (select epid, min(purduelevel::float/10) as purduelevel from $ADOM_ENDPOINT ep where
purduelevel in (20, 30, 35) group by epid) t2 on t1.ep_id=t2.epid inner join app_mdata t3 on
t1.app=t3.name group by srcip having sum(scores)>0 order by scores desc
```

Dataset Name	Description	Log Category
OT-Top-High-Risk-Apps-By-Bandwidth-Timeline	Top OT Applications by Bandwidth Timeline	traffic

```
select
  hodex,
  t1.app,
  t1.bandwidth
from
  (
    select
      $flex_timescale(timestamp) as hodex,
      app,
      sum(bandwidth) as bandwidth
    from
      ###(select $flex_timestamp as timestamp, dvid, epid as ep_id, srcip, dstip, osname, proto,
dstport, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
(CASE WHEN utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3
ELSE 0 END) as action_flag, sum(crscore%65536) as scores, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block,
count(*) as sessions from $log where $filter and (logflag&1>0) and epid>1024 group by timestamp,
dvid, ep_id, srcip, dstip, osname, proto, dstport, app, user_src, action_flag order by bandwidth
desc)### t1 inner join (select epid, min(purduelevel::float/10) as purduelevel from $ADOM_ENDPOINT
ep where purduelevel in (20, 30, 35) group by epid) t2 on t1.ep_id=t2.epid inner join app_mdata t3
on t1.app=t3.name where risk>='3' group by hodex, app having sum(bandwidth)>0 order by hodex) t1
inner join (select app, sum(bandwidth) as bandwidth from ###(select $flex_timestamp as timestamp,
dvid, epid as ep_id, srcip, dstip, osname, proto, dstport, app, coalesce(nullifna(`user`),
nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, (CASE WHEN utmaction='block' THEN 1 WHEN
utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3 ELSE 0 END) as action_flag, sum
(crscore%65536) as scores, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum
((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block, count(*) as sessions from $log
where $filter and (logflag&1>0) and epid>1024 group by timestamp, dvid, ep_id, srcip, dstip,
osname, proto, dstport, app, user_src, action_flag order by bandwidth desc)### t1 inner join
(select epid, min(purduelevel::float/10) as purduelevel from $ADOM_ENDPOINT ep where purduelevel
in (20, 30, 35) group by epid) t2 on t1.ep_id=t2.epid inner join app_mdata t3 on t1.app=t3.name
where risk>='3' group by app order by bandwidth desc limit $ddown-top) t2 on t1.app=t2.app order
by hodex
```

Dataset Name	Description	Log Category
OT-Top-High-Risk-Apps-by-Bandwidth	Top OT Zone High Risk Applications by Bandwidth	traffic

```
select
  risk as d_risk,
  name,
  sum(bandwidth) as bandwidth
from
  ###(select $flex_timestamp as timestamp, dvid, epid as ep_id, srcip, dstip, osname, proto,
```

```
dstport, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
(CASE WHEN utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3
ELSE 0 END) as action_flag, sum(crscore%65536) as scores, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block,
count(*) as sessions from $log where $filter and (logflag&1>0) and epid>1024 group by timestamp,
dvid, ep_id, srcip, dstip, osname, proto, dstport, app, user_src, action_flag order by bandwidth
desc)### t1 inner join (select epid, min(purduelevel::float/10) as purduelevel from $ADOM_ENDPOINT
ep where purduelevel in (20, 30, 35) group by epid) t2 on t1.ep_id=t2.epid inner join app_mdata t3
on t1.app=t3.name where risk>='3' group by d_risk, name, app_cat, technology order by bandwidth
desc, d_risk desc
```

Dataset Name	Description	Log Category
OT-Top-High-Risk-Apps-By-Sessions-Timeline	Top OT Applications by Sessions Timeline	traffic

```
select
  hodex,
  t1.app,
  t1.sessions
from
  (
    select
      $flex_timescale(timestamp) as hodex,
      app,
      sum(sessions) as sessions
    from
      ###(select $flex_timestamp as timestamp, dvid, epid as ep_id, srcip, dstip, osname, proto,
dstport, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
(CASE WHEN utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3
ELSE 0 END) as action_flag, sum(crscore%65536) as scores, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block,
count(*) as sessions from $log where $filter and (logflag&1>0) and epid>1024 group by timestamp,
dvid, ep_id, srcip, dstip, osname, proto, dstport, app, user_src, action_flag order by bandwidth
desc)### t1 inner join (select epid, min(purduelevel::float/10) as purduelevel from $ADOM_ENDPOINT
ep where purduelevel in (20, 30, 35) group by epid) t2 on t1.ep_id=t2.epid inner join app_mdata t3
on t1.app=t3.name where risk>='3' group by hodex, app having sum(sessions)>0 order by hodex) t1
inner join (select app, sum(sessions) as sessions from ###(select $flex_timestamp as timestamp,
dvid, epid as ep_id, srcip, dstip, osname, proto, dstport, app, coalesce(nullifna(`user`),
nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, (CASE WHEN utmaction='block' THEN 1 WHEN
utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3 ELSE 0 END) as action_flag, sum
(crscore%65536) as scores, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum
((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block, count(*) as sessions from $log
where $filter and (logflag&1>0) and epid>1024 group by timestamp, dvid, ep_id, srcip, dstip,
osname, proto, dstport, app, user_src, action_flag order by bandwidth desc)### t1 inner join
(select epid, min(purduelevel::float/10) as purduelevel from $ADOM_ENDPOINT ep where purduelevel
in (20, 30, 35) group by epid) t2 on t1.ep_id=t2.epid inner join app_mdata t3 on t1.app=t3.name
where risk>='3' group by app order by sessions desc limit $ddown-top) t2 on t1.app=t2.app order by
hodex
```

Dataset Name	Description	Log Category
OT-Top-High-Risk-Apps-by-Sessions	Top OT Zone High Risk Applications by Sessions	traffic

```
select
  risk as d_risk,
  name,
  sum(sessions) as sessions
from
  ###(select $flex_timestamp as timestamp, dvid, epid as ep_id, srcip, dstip, osname, proto,
dstport, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
(CASE WHEN utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3
ELSE 0 END) as action_flag, sum(crscore%65536) as scores, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block,
count(*) as sessions from $log where $filter and (logflag&1>0) and epid>1024 group by timestamp,
dvid, ep_id, srcip, dstip, osname, proto, dstport, app, user_src, action_flag order by bandwidth
desc)### t1 inner join (select epid, min(purduelevel::float/10) as purduelevel from $ADOM_ENDPOINT
ep where purduelevel in (20, 30, 35) group by epid) t2 on t1.ep_id=t2.epid inner join app_mdata t3
on t1.app=t3.name where risk>='3' group by d_risk, name, app_cat, technology order by sessions
desc, d_risk desc
```

Dataset Name	Description	Log Category
OT-Traffic-Flow-by-Bandwidth	Top OT Zone Traffic Flow by Bandwidth	traffic

```
select
  srcip,
  dstip,
  sum(bandwidth) as bandwidth
from
  ###(select $flex_timestamp as timestamp, dvid, epid as ep_id, srcip, dstip, osname, proto,
dstport, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
(CASE WHEN utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3
ELSE 0 END) as action_flag, sum(crscore%65536) as scores, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block,
count(*) as sessions from $log where $filter and (logflag&1>0) and epid>1024 group by timestamp,
dvid, ep_id, srcip, dstip, osname, proto, dstport, app, user_src, action_flag order by bandwidth
desc)### t1 inner join (select epid, min(purduelevel::float/10) as purduelevel from $ADOM_ENDPOINT
ep where purduelevel in (20, 30, 35) group by epid) t2 on t1.ep_id=t2.epid inner join app_mdata t3
on t1.app=t3.name where risk>='3' group by srcip, dstip order by bandwidth desc
```

Dataset Name	Description	Log Category
OT-Top-Asset-by-Last-External-Connection	Top OT Zone Assets by Last External Connection	traffic

```
select
  srcip,
  min(purduelevel) as purduelevel,
  sum(sessions) as sessions,
  max(last_app) as last_app,
  from_dtime(
    max(timestamp)
  ) as last_seen
from
  (
    select
      srcip,
      timestamp,
      purduelevel,
      sessions,
      first_value(app) over (
        PARTITION by srcip
        order by
          timestamp desc RANGE BETWEEN UNBOUNDED PRECEDING
          AND UNBOUNDED FOLLOWING
      ) as last_app
    from
      ###(select $flex_timestamp as timestamp, dvid, epid as ep_id, srcip, dstip, osname, proto,
dstport, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
(CASE WHEN utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3
ELSE 0 END) as action_flag, sum(crscore%65536) as scores, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block,
count(*) as sessions from $log where $filter and (logflag&1>0) and epid>1024 group by timestamp,
dvid, ep_id, srcip, dstip, osname, proto, dstport, app, user_src, action_flag order by bandwidth
desc)### t1 inner join (select epid, min(purduelevel::float/10) as purduelevel from $ADOM_ENDPOINT
ep where purduelevel in (20, 30, 35) group by epid) t2 on t1.ep_id=t2.epid inner join app_mdata t3
on t1.app=t3.name where risk>='3') t group by srcip order by last_seen desc, sessions desc
```

Dataset Name	Description	Log Category
OT-Top-Asset-by-Bandwidth	Top OT Zone Assets by Bandwidth	traffic

```
select
  srcip,
  min(purduelevel) as purduelevel,
  sum(bandwidth) as bandwidth,
  max(last_app) as last_app,
  sum(sessions) as sessions,
  max(dstip) as dstip,
  from_dtime(
    max(timestamp)
  ) as last_seen
from
  (
    select
      srcip,
```

```
      dstip,
      timestamp,
      purduelevel,
      sessions,
      bandwidth,
      first_value(app) over (
        PARTITION by srcip
        order by
          timestamp desc RANGE BETWEEN UNBOUNDED PRECEDING
          AND UNBOUNDED FOLLOWING
      ) as last_app
    from
      ###(select $flex_timestamp as timestamp, dvid, epid as ep_id, srcip, dstip, osname, proto,
dstport, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
(CASE WHEN utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3
ELSE 0 END) as action_flag, sum(crscore%65536) as scores, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block,
count(*) as sessions from $log where $filter and (logflag&1>0) and epid>1024 group by timestamp,
dvid, ep_id, srcip, dstip, osname, proto, dstport, app, user_src, action_flag order by bandwidth
desc)### t1 inner join (select epid, min(purduelevel::float/10) as purduelevel from $ADOM_ENDPOINT
ep where purduelevel in (20, 30, 35) group by epid) t2 on t1.ep_id=t2.epid inner join app_mdata t3
on t1.app=t3.name where risk>='3') t group by srcip order by last_seen desc, bandwidth desc
```

Dataset Name	Description	Log Category
IT-Application-Vulnerabilities-by-Risk-Level	IT Zone Application Vulnerabilities by Risk Level	attack

```
select
  severity,
  (
    case when severity =& #039;critical' then 5 when severity='high' then 4 when severity='medium'
then 3 when severity='low' then 2 when severity='info' then 1 else 0 end) as severity_number,
count(distinct t1.attack) as totalnum from ###(select $flex_timestamp as timestamp, dvid, (CASE
WHEN direction='incoming' THEN epid ELSE dstepid END) as ep_id, attack, attackid, severity,
direction, dstip, srcip, srccountry, (case when action in ('clear_session', 'pass_session') then 1
when action in ('reset', 'reset_client', 'reset_server') then 2 else 3 end) as action_flag, count
(*) as totalnum from $log-attack where $filter and nullifna(attack) is not null and severity is
not null and (epid>1024 or dstepid>1024) group by timestamp, dvid, ep_id, attack, attackid,
severity, direction, dstip, srcip, srccountry, action_flag order by totalnum desc)### t1 inner
join (select epid, min(purduelevel::float/10) as purduelevel from $ADOM_ENDPOINT ep where
purduelevel in (40, 50) group by epid) t2 on t1.ep_id=t2.epid left join (select name, cve, vuln_
type from ips_mdata) t3 on t1.attack=t3.name group by severity, severity_number order by severity_
number desc
```

Dataset Name	Description	Log Category
IT-Application-Vulnerabilities-by-Type	IT Zone Application Vulnerabilities by Type	attack

```
select
  vuln_type,
  count(distinct t1.attack) as totalnum
from
  ###(select $flex_timestamp as timestamp, dvid, (CASE WHEN direction='incoming' THEN epid ELSE
dstepid END) as ep_id, attack, attackid, severity, direction, dstip, srcip, srccountry, (case when
action in ('clear_session', 'pass_session') then 1 when action in ('reset', 'reset_client',
'reset_server') then 2 else 3 end) as action_flag, count(*) as totalnum from $log-attack where
$filter and nullifna(attack) is not null and severity is not null and (epid>1024 or dstepid>1024)
group by timestamp, dvid, ep_id, attack, attackid, severity, direction, dstip, srcip, srccountry,
action_flag order by totalnum desc)### t1 inner join (select epid, min(purduelevel::float/10) as
purduelevel from $ADOM_ENDPOINT ep where purduelevel in (40, 50) group by epid) t2 on t1.ep_
id=t2.epid left join (select name, cve, vuln_type from ips_mdata) t3 on t1.attack=t3.name where
vuln_type is not null group by vuln_type order by totalnum desc
```

Dataset Name	Description	Log Category
IT-Application-Vulnerabilities-by-Action	IT Zone Application Vulnerabilities by Action	attack

```
select
  (
    case when action_flag = 1 then & #039;Allow' when action_flag=2 then 'Reset' else 'Block' end)
as action, count(distinct t1.attack) as totalnum from ###(select $flex_timestamp as timestamp,
dvid, (CASE WHEN direction='incoming' THEN epid ELSE dstepid END) as ep_id, attack, attackid,
severity, direction, dstip, srcip, srccountry, (case when action in ('clear_session', 'pass_
session') then 1 when action in ('reset', 'reset_client', 'reset_server') then 2 else 3 end) as
action_flag, count(*) as totalnum from $log-attack where $filter and nullifna(attack) is not null
and severity is not null and (epid>1024 or dstepid>1024) group by timestamp, dvid, ep_id, attack,
attackid, severity, direction, dstip, srcip, srccountry, action_flag order by totalnum desc)### t1
inner join (select epid, min(purduelevel::float/10) as purduelevel from $ADOM_ENDPOINT ep where
purduelevel in (40, 50) group by epid) t2 on t1.ep_id=t2.epid left join (select name, cve, vuln_
type from ips_mdata) t3 on t1.attack=t3.name group by action order by totalnum desc
```

Dataset Name	Description	Log Category
IT-Top-Application-Vulnerability-by-Risk-Level	Top IT Zone Application Vulnerabilities by Risk Level	attack

```
select
  attack,
  attackid,
  vuln_type,
  cve,
  (
    case when severity =& #039;critical' then 5 when severity='high' then 4 when severity='medium'
then 3 when severity='low' then 2 when severity='info' then 1 else 0 end) as severity_number,
count(distinct (CASE WHEN direction='incoming' THEN srcip ELSE dstip END)) as victims, count
(distinct (CASE WHEN direction='incoming' THEN dstip ELSE srcip END)) as sources, sum(case when
```

```
action_flag=3 then totalnum else 0 end) as total_block, sum(case when action_flag!=3 then totalnum
else 0 end) as total_allow, sum(totalnum) as totalnum from ###(select $flex_timestamp as
timestamp, dvid, (CASE WHEN direction='incoming' THEN epid ELSE dstepid END) as ep_id, attack,
attackid, severity, direction, dstip, srcip, srccountry, (case when action in ('clear_session',
'pass_session') then 1 when action in ('reset', 'reset_client', 'reset_server') then 2 else 3 end)
as action_flag, count(*) as totalnum from $log-attack where $filter and nullifna(attack) is not
null and severity is not null and (epid>1024 or dstepid>1024) group by timestamp, dvid, ep_id,
attack, attackid, severity, direction, dstip, srcip, srccountry, action_flag order by totalnum
desc)### t1 inner join (select epid, min(purduelevel::float/10) as purduelevel from $ADOM_ENDPOINT
ep where purduelevel in (40, 50) group by epid) t2 on t1.ep_id=t2.epid left join (select name,
cve, vuln_type from ips_mdata) t3 on t1.attack=t3.name group by attack, attackid, vuln_type,
severity_number, cve order by severity_number desc, totalnum desc
```

Dataset Name	Description	Log Category
IT-High-Risk-Apps-by-Risk-Level	IT Zone High Risk Applications by Risk Level	traffic

```
select
  & #039;Risk' || risk as severity, risk as d_risk, count(distinct app) as sessions from ###
(select $flex_timestamp as timestamp, dvid, epid as ep_id, srcip, dstip, osname, proto, dstport,
app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, (CASE WHEN
utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3 ELSE 0 END)
as action_flag, sum(crscore%65536) as scores, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as
bandwidth, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block, count(*) as sessions
from $log where $filter and (logflag&1>0) and epid>1024 group by timestamp, dvid, ep_id, srcip,
dstip, osname, proto, dstport, app, user_src, action_flag order by bandwidth desc)### t1 inner
join (select epid, min(purduelevel::float/10) as purduelevel from $ADOM_ENDPOINT ep where
purduelevel in (40, 50) group by epid) t2 on t1.ep_id=t2.epid inner join app_mdata t3 on
t1.app=t3.name where risk>='3' group by severity, d_risk order by sessions desc
```

Dataset Name	Description	Log Category
IT-High-Risk-Apps-by-Category	IT Zone High Risk Applications by Category	traffic

```
select
  app_cat,
  count(distinct app) as sessions
from
  ###(select $flex_timestamp as timestamp, dvid, epid as ep_id, srcip, dstip, osname, proto,
dstport, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
(CASE WHEN utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3
ELSE 0 END) as action_flag, sum(crscore%65536) as scores, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block,
count(*) as sessions from $log where $filter and (logflag&1>0) and epid>1024 group by timestamp,
dvid, ep_id, srcip, dstip, osname, proto, dstport, app, user_src, action_flag order by bandwidth
desc)### t1 inner join (select epid, min(purduelevel::float/10) as purduelevel from $ADOM_ENDPOINT
ep where purduelevel in (40, 50) group by epid) t2 on t1.ep_id=t2.epid inner join app_mdata t3 on
t1.app=t3.name where risk>='3' group by app_cat order by sessions desc
```

Dataset Name	Description	Log Category
IT-High-Risk-Apps-by-Action	IT Zone High Risk Applications by Action	traffic

```
select
  (
    case when action_flag = 1 then & #039;Block' when action_flag=2 then 'Allow' else 'Reset' end)
as action, count(distinct app) as sessions from ###(select $flex_timestamp as timestamp, dvid,
epid as ep_id, srcip, dstip, osname, proto, dstport, app, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, (CASE WHEN utmaction='block' THEN 1 WHEN
utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3 ELSE 0 END) as action_flag, sum
(crscore%65536) as scores, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum
((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block, count(*) as sessions from $log
where $filter and (logflag&1>0) and epid>1024 group by timestamp, dvid, ep_id, srcip, dstip,
osname, proto, dstport, app, user_src, action_flag order by bandwidth desc)### t1 inner join
(select epid, min(purduelevel::float/10) as purduelevel from $ADOM_ENDPOINT ep where purduelevel
in (40, 50) group by epid) t2 on t1.ep_id=t2.epid inner join app_mdata t3 on t1.app=t3.name where
risk>='3' and action_flag>0 group by action order by sessions desc
```

Dataset Name	Description	Log Category
IT-Top-High-Risk-Apps-by-Risk	Top IT Zone High Risk Applications by Risk Level	traffic

```
select
  risk as d_risk,
  name,
  app_cat,
  technology,
  max(srcip) as asset,
  sum(bandwidth) as bandwidth,
  sum(session_block) as session_block,
  sum(sessions)- sum(session_block) as session_pass,
  sum(sessions) as sessions
from
  ###(select $flex_timestamp as timestamp, dvid, epid as ep_id, srcip, dstip, osname, proto,
dstport, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
(CASE WHEN utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3
ELSE 0 END) as action_flag, sum(crscore%65536) as scores, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block,
count(*) as sessions from $log where $filter and (logflag&1>0) and epid>1024 group by timestamp,
dvid, ep_id, srcip, dstip, osname, proto, dstport, app, user_src, action_flag order by bandwidth
desc)### t1 inner join (select epid, min(purduelevel::float/10) as purduelevel from $ADOM_ENDPOINT
ep where purduelevel in (40, 50) group by epid) t2 on t1.ep_id=t2.epid inner join app_mdata t3 on
t1.app=t3.name where risk>='3' group by d_risk, name, app_cat, technology order by d_risk desc,
bandwidth desc
```

Dataset Name	Description	Log Category
IT-Top-Asset-by-Threat-Score	Top IT Zone Assets by Threat Score	traffic

```
select
  srcip,
  & #039;Level ' || min(purduelevel) as purduelevel, sum(scores) as scores from ###(select $flex_
timestamp as timestamp, dvid, epid as ep_id, srcip, dstip, osname, proto, dstport, app, coalesce
(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, (CASE WHEN
utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3 ELSE 0 END)
as action_flag, sum(crscore%65536) as scores, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as
bandwidth, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block, count(*) as sessions
from $log where $filter and (logflag&1>0) and epid>1024 group by timestamp, dvid, ep_id, srcip,
dstip, osname, proto, dstport, app, user_src, action_flag order by bandwidth desc)### t1 inner
join (select epid, min(purduelevel::float/10) as purduelevel from $ADOM_ENDPOINT ep where
purduelevel in (40, 50) group by epid) t2 on t1.ep_id=t2.epid inner join app_mdata t3 on
t1.app=t3.name group by srcip having sum(scores)>0 order by scores desc
```

Dataset Name	Description	Log Category
IT-Top-High-Risk-Apps-By-Bandwidth-Timeline	Top IT Applications by Bandwidth Timeline	traffic

```
select
  hodex,
  t1.app,
  t1.bandwidth
from
  (
    select
      $flex_timescale(timestamp) as hodex,
      app,
      sum(bandwidth) as bandwidth
    from
      ###(select $flex_timestamp as timestamp, dvid, epid as ep_id, srcip, dstip, osname, proto,
dstport, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
(CASE WHEN utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3
ELSE 0 END) as action_flag, sum(crscore%65536) as scores, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block,
count(*) as sessions from $log where $filter and (logflag&1>0) and epid>1024 group by timestamp,
dvid, ep_id, srcip, dstip, osname, proto, dstport, app, user_src, action_flag order by bandwidth
desc)### t1 inner join (select epid, min(purduelevel::float/10) as purduelevel from $ADOM_ENDPOINT
ep where purduelevel in (40, 50) group by epid) t2 on t1.ep_id=t2.epid inner join app_mdata t3 on
t1.app=t3.name group by hodex, app having sum(bandwidth)>0 order by hodex) t1 inner join (select
app, sum(bandwidth) as bandwidth from ###(select $flex_timestamp as timestamp, dvid, epid as ep_
id, srcip, dstip, osname, proto, dstport, app, coalesce(nullifna(`user`), nullifna(`unauthuser`),
ipstr(`srcip`)) as user_src, (CASE WHEN utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2
WHEN utmaction='reset' THEN 3 ELSE 0 END) as action_flag, sum(crscore%65536) as scores, sum
(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum((CASE WHEN (logflag&2>0) THEN 1
ELSE 0 END)) AS session_block, count(*) as sessions from $log where $filter and (logflag&1>0) and
epid>1024 group by timestamp, dvid, ep_id, srcip, dstip, osname, proto, dstport, app, user_src,
action_flag order by bandwidth desc)### t1 inner join (select epid, min(purduelevel::float/10) as
purduelevel from $ADOM_ENDPOINT ep where purduelevel in (40, 50) group by epid) t2 on t1.ep_
id=t2.epid inner join app_mdata t3 on t1.app=t3.name group by app order by bandwidth desc limit
$ddown-top) t2 on t1.app=t2.app order by hodex
```

Dataset Name	Description	Log Category
IT-Top-High-Risk-Apps-by-Bandwidth	Top IT Zone High Risk Applications by Bandwidth	traffic

```
select
  risk as d_risk,
  name,
  app_cat,
  sum(bandwidth) as bandwidth,
  sum(sessions) as sessions
from
  ###(select $flex_timestamp as timestamp, dvid, epid as ep_id, srcip, dstip, osname, proto,
dstport, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
(CASE WHEN utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3
ELSE 0 END) as action_flag, sum(crscore%65536) as scores, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block,
count(*) as sessions from $log where $filter and (logflag&1>0) and epid>1024 group by timestamp,
dvid, ep_id, srcip, dstip, osname, proto, dstport, app, user_src, action_flag order by bandwidth
desc)### t1 inner join (select epid, min(purduelevel::float/10) as purduelevel from $ADOM_ENDPOINT
ep where purduelevel in (40, 50) group by epid) t2 on t1.ep_id=t2.epid inner join app_mdata t3 on
t1.app=t3.name group by d_risk, name, app_cat order by bandwidth desc, d_risk desc
```

Dataset Name	Description	Log Category
IT-Top-High-Risk-Apps-By-Sessions-Timeline	Top IT Applications by Sessions Timeline	traffic

```
select
  hodex,
  t1.app,
  t1.sessions
from
  (
    select
      $flex_timescale(timestamp) as hodex,
      app,
      sum(sessions) as sessions
    from
      ###(select $flex_timestamp as timestamp, dvid, epid as ep_id, srcip, dstip, osname, proto,
dstport, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
(CASE WHEN utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3
ELSE 0 END) as action_flag, sum(crscore%65536) as scores, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block,
count(*) as sessions from $log where $filter and (logflag&1>0) and epid>1024 group by timestamp,
dvid, ep_id, srcip, dstip, osname, proto, dstport, app, user_src, action_flag order by bandwidth
desc)### t1 inner join (select epid, min(purduelevel::float/10) as purduelevel from $ADOM_ENDPOINT
ep where purduelevel in (40, 50) group by epid) t2 on t1.ep_id=t2.epid inner join app_mdata t3 on
t1.app=t3.name group by hodex, app having sum(sessions)>0 order by hodex) t1 inner join (select
app, sum(sessions) as sessions from ###(select $flex_timestamp as timestamp, dvid, epid as ep_id,
```

```
srcip, dstip, osname, proto, dstport, app, coalesce(nullifna(`user`), nullifna(`unauthuser`),
ipstr(`srcip`)) as user_src, (CASE WHEN utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2
WHEN utmaction='reset' THEN 3 ELSE 0 END) as action_flag, sum(crscore%65536) as scores, sum
(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum((CASE WHEN (logflag&2>0) THEN 1
ELSE 0 END)) AS session_block, count(*) as sessions from $log where $filter and (logflag&1>0) and
epid>1024 group by timestamp, dvid, ep_id, srcip, dstip, osname, proto, dstport, app, user_src,
action_flag order by bandwidth desc)### t1 inner join (select epid, min(purduelevel::float/10) as
purduelevel from $ADOM_ENDPOINT ep where purduelevel in (40, 50) group by epid) t2 on t1.ep_
id=t2.epid inner join app_mdata t3 on t1.app=t3.name group by app order by sessions desc limit
$ddown-top) t2 on t1.app=t2.app order by hodex
```

Dataset Name	Description	Log Category
IT-Top-High-Risk-Apps-by-Sessions	Top IT Zone High Risk Applications by Sessions	traffic

```
select
  risk as d_risk,
  name,
  app_cat,
  sum(sessions) as sessions,
  sum(bandwidth) as bandwidth
from
  ###(select $flex_timestamp as timestamp, dvid, epid as ep_id, srcip, dstip, osname, proto,
dstport, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
(CASE WHEN utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3
ELSE 0 END) as action_flag, sum(crscore%65536) as scores, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block,
count(*) as sessions from $log where $filter and (logflag&1>0) and epid>1024 group by timestamp,
dvid, ep_id, srcip, dstip, osname, proto, dstport, app, user_src, action_flag order by bandwidth
desc)### t1 inner join (select epid, min(purduelevel::float/10) as purduelevel from $ADOM_ENDPOINT
ep where purduelevel in (40, 50) group by epid) t2 on t1.ep_id=t2.epid inner join app_mdata t3 on
t1.app=t3.name group by d_risk, name, app_cat order by sessions desc, d_risk desc
```

Dataset Name	Description	Log Category
IT-Traffic-Flow-by-Bandwidth	Top IT Zone Traffic Flow by Bandwidth	traffic

```
select
  srcip,
  dstip,
  sum(bandwidth) as bandwidth
from
  ###(select $flex_timestamp as timestamp, dvid, epid as ep_id, srcip, dstip, osname, proto,
dstport, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
(CASE WHEN utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3
ELSE 0 END) as action_flag, sum(crscore%65536) as scores, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block,
count(*) as sessions from $log where $filter and (logflag&1>0) and epid>1024 group by timestamp,
```

```
dvid, ep_id, srcip, dstip, osname, proto, dstport, app, user_src, action_flag order by bandwidth
desc)### t1 inner join (select epid, min(purduelevel::float/10) as purduelevel from $ADOM_ENDPOINT
ep where purduelevel in (40, 50) group by epid) t2 on t1.ep_id=t2.epid inner join app_mdata t3 on
t1.app=t3.name where risk>='3' group by srcip, dstip order by bandwidth desc
```

Dataset Name	Description	Log Category
IT-Top-Asset-by-Last-External-Connection	Top IT Zone Assets by Last External Connection	traffic

```
select
  srcip,
  min(purduelevel) as purduelevel,
  sum(sessions) as sessions,
  max(last_app) as last_app,
  from_dtime(
    max(timestamp)
  ) as last_seen
from
  (
    select
      srcip,
      timestamp,
      purduelevel,
      sessions,
      first_value(app) over (
        PARTITION by srcip
        order by
          timestamp desc RANGE BETWEEN UNBOUNDED PRECEDING
          AND UNBOUNDED FOLLOWING
      ) as last_app
    from
      ###(select $flex_timestamp as timestamp, dvid, epid as ep_id, srcip, dstip, osname, proto,
dstport, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
(CASE WHEN utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3
ELSE 0 END) as action_flag, sum(crscore%65536) as scores, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block,
count(*) as sessions from $log where $filter and (logflag&1>0) and epid>1024 group by timestamp,
dvid, ep_id, srcip, dstip, osname, proto, dstport, app, user_src, action_flag order by bandwidth
desc)### t1 inner join (select epid, min(purduelevel::float/10) as purduelevel from $ADOM_ENDPOINT
ep where purduelevel in (40, 50) group by epid) t2 on t1.ep_id=t2.epid inner join app_mdata t3 on
t1.app=t3.name where risk>='3') t group by srcip order by last_seen desc, sessions desc
```

Dataset Name	Description	Log Category
IT-Top-Asset-by-Bandwidth	Top IT Zone Assets by Bandwidth	traffic

```
select
  srcip,
```

```
  min(purduelevel) as purduelevel,
  sum(bandwidth) as bandwidth,
  max(last_app) as last_app,
  sum(sessions) as sessions,
  max(dstip) as dstip,
  from_dtime(
    max(timestamp)
  ) as last_seen
from
  (
    select
      srcip,
      dstip,
      timestamp,
      purduelevel,
      sessions,
      bandwidth,
      first_value(app) over (
        PARTITION by srcip
        order by
          timestamp desc RANGE BETWEEN UNBOUNDED PRECEDING
          AND UNBOUNDED FOLLOWING
      ) as last_app
    from
      ###(select $flex_timestamp as timestamp, dvid, epid as ep_id, srcip, dstip, osname, proto,
dstport, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
(CASE WHEN utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3
ELSE 0 END) as action_flag, sum(crscore%65536) as scores, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block,
count(*) as sessions from $log where $filter and (logflag&1>0) and epid>1024 group by timestamp,
dvid, ep_id, srcip, dstip, osname, proto, dstport, app, user_src, action_flag order by bandwidth
desc)### t1 inner join (select epid, min(purduelevel::float/10) as purduelevel from $ADOM_ENDPOINT
ep where purduelevel in (40, 50) group by epid) t2 on t1.ep_id=t2.epid inner join app_mdata t3 on
t1.app=t3.name where risk>='3') t group by srcip order by last_seen desc, bandwidth desc
```

Dataset Name	Description	Log Category
threat-Reconnaissance-Activities-by-Country	Reconnaissance Activities by Country	attack

```
select
  srccountry,
  sum(incidents) as incidents
from
  ###(select $flex_timestamp as timestamp, attack, (case when action in ('clear_session', 'pass_
session') then 1 when action in ('reset', 'reset_client', 'reset_server') then 2 else 3 end) as
action_flag, srcip, dstip, srccountry, count(*) as incidents from $log-attack where $filter and
attack is not null and msg like 'anomaly%' group by timestamp, attack, action_flag, srcip, dstip,
srccountry order by incidents desc)### t where srccountry is not null group by srccountry order by
incidents desc
```

Dataset Name	Description	Log Category
threat-Reconnaissance-Activities-by-Attack	Reconnaissance Activities by Attack	attack

```
select
  attack,
  sum(incidents) as incidents
from
  ###(select $flex_timestamp as timestamp, attack, (case when action in ('clear_session', 'pass_
session') then 1 when action in ('reset', 'reset_client', 'reset_server') then 2 else 3 end) as
action_flag, srcip, dstip, srccountry, count(*) as incidents from $log-attack where $filter and
attack is not null and msg like 'anomaly%' group by timestamp, attack, action_flag, srcip, dstip,
srccountry order by incidents desc)### t group by attack order by incidents desc
```

Dataset Name	Description	Log Category
threat-Reconnaissance-Activities-by-Action	Reconnaissance Activities by Action	attack

```
select
  (
    case when action_flag = 1 then & #039;Allow' when action_flag=1 then 'Reset' else 'Block' end)
as action, sum(incidents) as incidents from ###(select $flex_timestamp as timestamp, attack, (case
when action in ('clear_session', 'pass_session') then 1 when action in ('reset', 'reset_client',
'reset_server') then 2 else 3 end) as action_flag, srcip, dstip, srccountry, count(*) as incidents
from $log-attack where $filter and attack is not null and msg like 'anomaly%' group by timestamp,
attack, action_flag, srcip, dstip, srccountry order by incidents desc)### t group by action order
by incidents desc
```

Dataset Name	Description	Log Category
threat-Top-Reconnaissance-Activities-by-Occurrences	Top Reconnaissance Activities by Occurrences	attack

```
select
  attack,
  srcip,
  dstip,
  sum(incidents_pass) as incidents_pass,
  sum(incidents)- sum(incidents_pass) as incidents_block,
  sum(incidents) as incidents
from
  (
    select
      attack,
      srcip,
      dstip,
```

```
      sum(
        case when action_flag = 1 then incidents else 0 end
      ) as incidents_pass,
      sum(
        case when action_flag != 2 then incidents else 0 end
      ) as incidents
    from
      ###(select $flex_timestamp as timestamp, attack, (case when action in ('clear_session',
'pass_session') then 1 when action in ('reset', 'reset_client', 'reset_server') then 2 else 3 end)
as action_flag, srcip, dstip, srccountry, count(*) as incidents from $log-attack where $filter and
attack is not null and msg like 'anomaly%' group by timestamp, attack, action_flag, srcip, dstip,
srccountry order by incidents desc)### t group by attack, srcip, dstip order by incidents desc) t
group by attack, srcip, dstip order by incidents desc
```

Dataset Name	Description	Log Category
threat-High-Risk-Web-Access-Attempts-by-Category	High Risk Web Access Attempts by Category	webfilter

```
select
  catdesc,
  sum(total_num) as total_num
from
  ###(select $flex_timestamp as timestamp, coalesce(nullifna(`user`), nullifna(`unauthuser`),
ipstr(`srcip`)) as user_src, catdesc, action, count(*) as total_num from $log-webfilter where
$filter and catdesc in ('Dynamic DNS', 'Malicious Websites', 'Newly Observed Domain', 'Newly
Registered Domain', 'Phishing', 'Spam URLs') group by timestamp, user_src, catdesc, action order
by total_num desc)### t group by catdesc order by total_num desc
```

Dataset Name	Description	Log Category
threat-High-Risk-Web-Access-Attempts-by-Action	High Risk Web Access Attempts by Action	webfilter

```
select
  (
    case when action =& #039;blocked' then 'Block' else 'Allow' end) as action, sum(total_num) as
total_num from ###(select $flex_timestamp as timestamp, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, catdesc, action, count(*) as total_num from $log-
webfilter where $filter and catdesc in ('Dynamic DNS', 'Malicious Websites', 'Newly Observed
Domain', 'Newly Registered Domain', 'Phishing', 'Spam URLs') group by timestamp, user_src,
catdesc, action order by total_num desc)### t group by action order by total_num desc
```

Dataset Name	Description	Log Category
threat-Top-High-Risk-Web-Users-by-Access-Attempts	Top High Risk Website Users by Requests	webfilter

```
select
  user_src,
  sum(total_num) as total_num
from
  ###(select $flex_timestamp as timestamp, coalesce(nullifna(`user`), nullifna(`unauthuser`),
ipstr(`srcip`)) as user_src, catdesc, action, count(*) as total_num from $log-webfilter where
$filter and catdesc in ('Dynamic DNS', 'Malicious Websites', 'Newly Observed Domain', 'Newly
Registered Domain', 'Phishing', 'Spam URLs') group by timestamp, user_src, catdesc, action order
by total_num desc)### t group by user_src order by total_num desc
```

Dataset Name	Description	Log Category
threat-High-Risk-Apps-by-Risk-Level	High Risk Applications by Risk Level	traffic

```
select
  (
    case when risk =& #039;5' then 'Critical' when risk='4' then 'High' else '0' end) as severity,
risk as d_risk, sum(sessions) as sessions from ###(select $flex_timestamp as timestamp, app,
coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, (CASE WHEN
utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3 ELSE 0 END)
as action_flag, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, count(*) as
sessions from $log where $filter and (logflag&1>0) group by timestamp, app, user_src, action_flag
order by bandwidth desc)### t1 inner join app_mdata t2 on t1.app=t2.name where risk>='4' group by
severity, d_risk order by sessions desc
```

Dataset Name	Description	Log Category
threat-High-Risk-Apps-by-Category	High Risk Applications by Category	traffic

```
select
  app_cat,
  sum(sessions) as sessions
from
  ###(select $flex_timestamp as timestamp, app, coalesce(nullifna(`user`), nullifna(`unauthuser`),
ipstr(`srcip`)) as user_src, (CASE WHEN utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2
WHEN utmaction='reset' THEN 3 ELSE 0 END) as action_flag, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth, count(*) as sessions from $log where $filter and (logflag&1>0) group
by timestamp, app, user_src, action_flag order by bandwidth desc)### t1 inner join app_mdata t2 on
t1.app=t2.name where risk>='4' group by app_cat order by sessions desc
```

Dataset Name	Description	Log Category
threat-High-Risk-Apps-by-Action	High Risk Applications by Action	traffic

```
select
  (
    case when action_flag = 1 then & #039;Block' when action_flag=2 then 'Allow' else 'Reset' end)
```

```
as action, sum(sessions) as sessions from ###(select $flex_timestamp as timestamp, app, coalesce
(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, (CASE WHEN
utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3 ELSE 0 END)
as action_flag, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, count(*) as
sessions from $log where $filter and (logflag&1>0) group by timestamp, app, user_src, action_flag
order by bandwidth desc)### t1 inner join app_mdata t2 on t1.app=t2.name where risk>='4' and
action_flag>0 group by action order by sessions desc
```

Dataset Name	Description	Log Category
threat-Top-High-Risk-Apps-by-Risk	Top High Risk Applications by Risk	traffic

```
select
  risk as d_risk,
  name,
  app_cat,
  count(distinct user_src) as users,
  sum(bandwidth) as bandwidth,
  sum(session_block) as session_block,
  sum(sessions)- sum(session_block) as session_pass,
  sum(sessions) as sessions
from
  ###(select $flex_timestamp as timestamp, app, coalesce(nullifna(`user`), nullifna(`unauthuser`),
ipstr(`srcip`)) as user_src, (CASE WHEN utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2
WHEN utmaction='reset' THEN 3 ELSE 0 END) as action_flag, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block,
count(*) as sessions from $log where $filter and (logflag&1>0) group by timestamp, app, user_src,
action_flag order by bandwidth desc)### t1 inner join app_mdata t2 on t1.app=t2.name where
risk>='4' group by d_risk, name, app_cat order by d_risk desc, bandwidth desc
```

Dataset Name	Description	Log Category
threat-Top-High-Risk-App-Users	Top High Risk Application Users	traffic

```
select
  user_src,
  count(distinct app) as total_num
from
  ###(select $flex_timestamp as timestamp, app, coalesce(nullifna(`user`), nullifna(`unauthuser`),
ipstr(`srcip`)) as user_src, (CASE WHEN utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2
WHEN utmaction='reset' THEN 3 ELSE 0 END) as action_flag, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth, count(*) as sessions from $log where $filter and (logflag&1>0) group
by timestamp, app, user_src, action_flag order by bandwidth desc)### t1 inner join app_mdata t2 on
t1.app=t2.name where risk>='4' group by user_src order by total_num desc
```

Dataset Name	Description	Log Category
threat-First-Stage-Timeline	Cyber Kill Chain First Stage Timeline	traffic

```
select
  $flex_timestamp(timestamp) as hodex,
  type,
  sum(totalnum) as totalnum
from
  (
    (
      select
        timestamp,
        & #039;Reconnaissance Activities' as type, sum(incidents) as totalnum from ###(select
$flex_timestamp as timestamp, attack, (case when action in ('clear_session', 'pass_session') then
1 when action in ('reset', 'reset_client', 'reset_server') then 2 else 3 end) as action_flag,
srcip, dstip, srccountry, count(*) as incidents from $log-attack where $filter and attack is not
null and msg like 'anomaly%' group by timestamp, attack, action_flag, srcip, dstip, srccountry
order by incidents desc)### t group by timestamp, type order by totalnum desc) union all (select
timestamp, 'Access Attempts to High-Risk Websites' as type, sum(total_num) as totalnum from ###
(select $flex_timestamp as timestamp, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr
(`srcip`)) as user_src, catdesc, action, count(*) as total_num from $log-webfilter where $filter
and catdesc in ('Dynamic DNS', 'Malicious Websites', 'Newly Observed Domain', 'Newly Registered
Domain', 'Phishing', 'Spam URLs') group by timestamp, user_src, catdesc, action order by total_num
desc)### t group by timestamp, type order by totalnum desc) union all (select timestamp, 'High-
Risk Applications' as type, sum(sessions) as totalnum from ###(select $flex_timestamp as
timestamp, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
(CASE WHEN utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3
ELSE 0 END) as action_flag, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, count
(*) as sessions from $log where $filter and (logflag&1>0) group by timestamp, app, user_src,
action_flag order by bandwidth desc)### t1 inner join app_mdata t2 on t1.app=t2.name where
risk>='4' group by timestamp, type order by totalnum desc)) t group by hodex, type order by hodex
```

Dataset Name	Description	Log Category
threat-Application-Vulnerabilities-by-Risk-Level	Application Vulnerabilities by Risk Level	attack

```
select
  severity,
  (
    case when severity =& #039;critical' then 5 when severity='high' then 4 when severity='medium'
then 3 when severity='low' then 2 when severity='info' then 1 else 0 end) as severity_number, sum
(totalnum) as totalnum from ###(select $flex_timestamp as timestamp, attack, attackid, severity,
direction, dstip, srcip, srccountry, (case when action in ('clear_session', 'pass_session') then 1
when action in ('reset', 'reset_client', 'reset_server') then 2 else 3 end) as action_flag, count
(*) as totalnum from $log-attack where $filter and nullifna(attack) is not null and severity is
not null group by timestamp, attack, attackid, severity, direction, dstip, srcip, srccountry,
action_flag order by totalnum desc)### t1 left join (select name, cve, vuln_type from ips_mdata)
t2 on t1.attack=t2.name group by severity, severity_number order by severity_number desc
```

Dataset Name	Description	Log Category
threat-Application-Vulnerabilities-by-Type	Application Vulnerabilities by Type	attack

```
select
  vuln_type,
  sum(totalnum) as totalnum
from
  ###(select $flex_timestamp as timestamp, attack, attackid, severity, direction, dstip, srcip,
srccountry, (case when action in ('clear_session', 'pass_session') then 1 when action in ('reset',
'reset_client', 'reset_server') then 2 else 3 end) as action_flag, count(*) as totalnum from $log-
attack where $filter and nullifna(attack) is not null and severity is not null group by timestamp,
attack, attackid, severity, direction, dstip, srcip, srccountry, action_flag order by totalnum
desc)### t1 left join (select name, cve, vuln_type from ips_mdata) t2 on t1.attack=t2.name where
vuln_type is not null group by vuln_type order by totalnum desc
```

Dataset Name	Description	Log Category
threat-Application-Vulnerabilities-by-Action	Application Vulnerabilities by Action	attack

```
select
  (
    case when action_flag = 1 then & #039;Allow' when action_flag=2 then 'Reset' else 'Block' end)
as action, sum(totalnum) as totalnum from ###(select $flex_timestamp as timestamp, attack,
attackid, severity, direction, dstip, srcip, srccountry, (case when action in ('clear_session',
'pass_session') then 1 when action in ('reset', 'reset_client', 'reset_server') then 2 else 3 end)
as action_flag, count(*) as totalnum from $log-attack where $filter and nullifna(attack) is not
null and severity is not null group by timestamp, attack, attackid, severity, direction, dstip,
srcip, srccountry, action_flag order by totalnum desc)### t1 left join (select name, cve, vuln_
type from ips_mdata) t2 on t1.attack=t2.name group by action order by totalnum desc
```

Dataset Name	Description	Log Category
threat-Top-Application-Vulnerability-by-Risk-Level	Top Application Vulnerabilities by Risk Level	attack

```
select
  attack,
  attackid,
  vuln_type,
  cve,
  (
    case when severity =& #039;critical' then 5 when severity='high' then 4 when severity='medium'
then 3 when severity='low' then 2 when severity='info' then 1 else 0 end) as severity_number,
count(distinct (CASE WHEN direction='incoming' THEN srcip ELSE dstip END)) as victims, count
(distinct (CASE WHEN direction='incoming' THEN dstip ELSE srcip END)) as sources, sum(case when
```

```
action_flag=3 then totalnum else 0 end) as total_block, sum(case when action_flag!=3 then totalnum
else 0 end) as total_allow, sum(totalnum) as totalnum from ###(select $flex_timestamp as
timestamp, attack, attackid, severity, direction, dstip, srcip, srccountry, (case when action in
('clear_session', 'pass_session') then 1 when action in ('reset', 'reset_client', 'reset_server')
then 2 else 3 end) as action_flag, count(*) as totalnum from $log-attack where $filter and
nullifna(attack) is not null and severity is not null group by timestamp, attack, attackid,
severity, direction, dstip, srcip, srccountry, action_flag order by totalnum desc)### t1 left join
(select name, cve, vuln_type from ips_mdata) t2 on t1.attack=t2.name group by attack, attackid,
vuln_type, severity_number, cve order by severity_number desc, totalnum desc
```

Dataset Name	Description	Log Category
threat-Top-Application-Vulnerability-Sources	Top Application Vulnerability Sources	attack

```
select
  srcip,
  srccountry,
  sum(totalnum) as totalnum
from
  ###(select $flex_timestamp as timestamp, attack, attackid, severity, direction, dstip, srcip,
srccountry, (case when action in ('clear_session', 'pass_session') then 1 when action in ('reset',
'reset_client', 'reset_server') then 2 else 3 end) as action_flag, count(*) as totalnum from $log-
attack where $filter and nullifna(attack) is not null and severity is not null group by timestamp,
attack, attackid, severity, direction, dstip, srcip, srccountry, action_flag order by totalnum
desc)### t1 left join (select name, cve, vuln_type from ips_mdata) t2 on t1.attack=t2.name where
srcip is not null group by srcip, srccountry order by totalnum desc
```

Dataset Name	Description	Log Category
threat-Top-Application-Vulnerability-Destinations	Top Application Vulnerability Destinations	attack

```
select
  dstip,
  sum(totalnum) as totalnum
from
  ###(select $flex_timestamp as timestamp, attack, attackid, severity, direction, dstip, srcip,
srccountry, (case when action in ('clear_session', 'pass_session') then 1 when action in ('reset',
'reset_client', 'reset_server') then 2 else 3 end) as action_flag, count(*) as totalnum from $log-
attack where $filter and nullifna(attack) is not null and severity is not null group by timestamp,
attack, attackid, severity, direction, dstip, srcip, srccountry, action_flag order by totalnum
desc)### t1 left join (select name, cve, vuln_type from ips_mdata) t2 on t1.attack=t2.name group
by dstip order by totalnum desc
```

Dataset Name	Description	Log Category
threat-Malware-Types-by-Occurrences	Malware Types by Occurrences	traffic

```
select
  (
    case when virus like & #039;Riskware%' then 'Spyware' when virus like 'Adware%' then 'Adware'
else 'Virus' end) as malware_type, sum(sessions) as sessions from ###(select $flex_timestamp as
timestamp, virus, app, (CASE WHEN utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2 WHEN
utmaction='reset' THEN 3 ELSE 0 END) as action_flag, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, srcip, dstip, sum((CASE WHEN (logflag&2>0) THEN 1
ELSE 0 END)) as session_block, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions from
$log where $filter and (logflag&(1|32)>0) and nullifna(app) is not null and virus is not null
group by timestamp, virus, app, user_src, srcip, dstip, action_flag order by sessions desc)### t
group by malware_type order by sessions desc
```

Dataset Name	Description	Log Category
threat-Malware-Actions-by-Occurrences	Malware Actions by Occurrences	traffic

```
select
  (
    case when action_flag = 1 then & #039;Block' when action_flag=2 then 'Allow' else 'Reset' end)
as action, sum(sessions) as sessions from ###(select $flex_timestamp as timestamp, virus, app,
(CASE WHEN utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3
ELSE 0 END) as action_flag, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as
user_src, srcip, dstip, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) as session_block, sum
(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions from $log where $filter and (logflag&
(1|32)>0) and nullifna(app) is not null and virus is not null group by timestamp, virus, app,
user_src, srcip, dstip, action_flag order by sessions desc)### t where action_flag>0 group by
action order by sessions desc
```

Dataset Name	Description	Log Category
threat-Top-Malwares-by-Occurrences	Top Malwares by Occurrences	traffic

```
select
  virus,
  (
    case when virus like & #039;Riskware%' then 'Spyware' when virus like 'Adware%' then 'Adware'
else 'Virus' end) as malware_type, count(distinct srcip) as victims, count(distinct dstip) as
sources, sum(session_block) as session_block, sum(sessions)-sum(session_block) as session_pass,
sum(sessions) as sessions from ###(select $flex_timestamp as timestamp, virus, app, (CASE WHEN
utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3 ELSE 0 END)
as action_flag, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
srcip, dstip, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) as session_block, sum(CASE WHEN
(logflag&1>0) THEN 1 ELSE 0 END) as sessions from $log where $filter and (logflag&(1|32)>0) and
nullifna(app) is not null and virus is not null group by timestamp, virus, app, user_src, srcip,
dstip, action_flag order by sessions desc)### t group by virus, malware_type order by sessions
desc
```

Dataset Name	Description	Log Category
threat-Top-Malware-Victims-by-Occurrences	Top Malware Victims by Occurrences	traffic

```
select
  user_src,
  sum(sessions) as sessions
from
  ###(select $flex_timestamp as timestamp, virus, app, (CASE WHEN utmaction='block' THEN 1 WHEN
utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3 ELSE 0 END) as action_flag, coalesce
(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, srcip, dstip, sum((CASE
WHEN (logflag&2>0) THEN 1 ELSE 0 END)) as session_block, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0
END) as sessions from $log where $filter and (logflag&(1|32)>0) and nullifna(app) is not null and
virus is not null group by timestamp, virus, app, user_src, srcip, dstip, action_flag order by
sessions desc)### t where user_src is not null group by user_src order by sessions desc
```

Dataset Name	Description	Log Category
threat-Second-Stage-Timeline	Cyber Kill Chain Second Stage Timeline	traffic

```
select
  $flex_timestamp(timestamp) as hodex,
  type,
  sum(totalnum) as totalnum
from
  (
    (
      select
        timestamp,
        & #039;Application Vulnerabilities' as type, sum(totalnum) as totalnum from ###(select
$flex_timestamp as timestamp, attack, attackid, severity, direction, dstip, srcip, srccountry,
(case when action in ('clear_session', 'pass_session') then 1 when action in ('reset', 'reset_
client', 'reset_server') then 2 else 3 end) as action_flag, count(*) as totalnum from $log-attack
where $filter and nullifna(attack) is not null and severity is not null group by timestamp,
attack, attackid, severity, direction, dstip, srcip, srccountry, action_flag order by totalnum
desc)### t1 left join (select name, cve, vuln_type from ips_mdata) t2 on t1.attack=t2.name group
by timestamp, type order by totalnum desc) union all (select timestamp, 'Malware Detected' as
type, sum(sessions) as totalnum from ###(select $flex_timestamp as timestamp, virus, app, (CASE
WHEN utmaction='block' THEN 1 WHEN utmaction='allow' THEN 2 WHEN utmaction='reset' THEN 3 ELSE 0
END) as action_flag, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_
src, srcip, dstip, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) as session_block, sum(CASE
WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions from $log where $filter and (logflag&(1|32)>0)
and nullifna(app) is not null and virus is not null group by timestamp, virus, app, user_src,
srcip, dstip, action_flag order by sessions desc)### t group by timestamp, type order by totalnum
desc)) t group by hodex, type order by hodex
```

Dataset Name	Description	Log Category
threat-CC-Domain-DNS-Resolutions-by-Severity	DNS Resolutions for C&C Domain by Severity	dns

```
select
  sevid,
  sum(total_num) as total_num
from
  ###(select $flex_timestamp as timestamp, coalesce(botnetdomain, ipstr(botnetip)) as domain,
qname, cast('Botnet C&C' as char(32)) as malware_type, (case when action='block' then 'Blocked'
when action='redirect' then 'Redirected' else 'Passed' end) as action, srcip, coalesce(nullifna
(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, (CASE WHEN level IN ('critical',
'alert', 'emergency') THEN 5 WHEN level='error' THEN 4 WHEN level='warning' THEN 3 WHEN
level='notice' THEN 2 ELSE 1 END) as sevid, coalesce(botnetdomain, ipstr(botnetip)) as sources_s,
count(*) as total_num from $log-dns where $filter and (botnetdomain is not null or botnetip is not
null) group by timestamp, domain, qname, action, srcip, user_src, sevid order by sevid desc)### t
group by sevid order by total_num desc
```

Dataset Name	Description	Log Category
threat-CC-Domain-DNS-Resolutions-by-Action	DNS Resolutions for C&C Domain by Action	dns

```
select
  action,
  sum(total_num) as total_num
from
  ###(select $flex_timestamp as timestamp, coalesce(botnetdomain, ipstr(botnetip)) as domain,
qname, cast('Botnet C&C' as char(32)) as malware_type, (case when action='block' then 'Blocked'
when action='redirect' then 'Redirected' else 'Passed' end) as action, srcip, coalesce(nullifna
(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, (CASE WHEN level IN ('critical',
'alert', 'emergency') THEN 5 WHEN level='error' THEN 4 WHEN level='warning' THEN 3 WHEN
level='notice' THEN 2 ELSE 1 END) as sevid, coalesce(botnetdomain, ipstr(botnetip)) as sources_s,
count(*) as total_num from $log-dns where $filter and (botnetdomain is not null or botnetip is not
null) group by timestamp, domain, qname, action, srcip, user_src, sevid order by sevid desc)### t
group by action order by total_num desc
```

Dataset Name	Description	Log Category
threat-Top-CC-Domain-DNS-Resolution-Users-by-Attempts	Top DNS Resolution for C&C Domain Users by Attempts	dns

```
select
  user_src,
  sum(total_num) as total_num
from
  ###(select $flex_timestamp as timestamp, coalesce(botnetdomain, ipstr(botnetip)) as domain,
```

```
qname, cast('Botnet C&C' as char(32)) as malware_type, (case when action='block' then 'Blocked'
when action='redirect' then 'Redirected' else 'Passed' end) as action, srcip, coalesce(nullifna
(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, (CASE WHEN level IN ('critical',
'alert', 'emergency') THEN 5 WHEN level='error' THEN 4 WHEN level='warning' THEN 3 WHEN
level='notice' THEN 2 ELSE 1 END) as sevid, coalesce(botnetdomain, ipstr(botnetip)) as sources_s,
count(*) as total_num from $log-dns where $filter and (botnetdomain is not null or botnetip is not
null) group by timestamp, domain, qname, action, srcip, user_src, sevid order by sevid desc)### t
group by user_src order by total_num desc
```

Dataset Name	Description	Log Category
threat-CC-Sites-Connections-by-Severity	Connections to C&C Sites by Severity	attack

```
select
  severity,
  sum(totalnum) as totalnum
from
  ###(select $flex_timestamp as timestamp, attack, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, severity, service, dstip, srcip, (case when action in
('clear_session', 'pass_session') then 1 when action in ('reset', 'reset_client', 'reset_server')
then 2 else 3 end) as action_flag, count(*) as totalnum from $log-attack where $filter and subtype
= 'ips' and nullifna(attack) is not null and severity is not null group by timestamp, attack,
user_src, severity, service, dstip, srcip, action_flag order by totalnum desc)### t group by
severity order by severity desc
```

Dataset Name	Description	Log Category
threat-CC-Sites-Connections-by-Action	Connections to C&C Sites by Action	attack

```
select
  (
    case when action_flag = 1 then & #039;Allow' when action_flag=2 then 'Reset' else 'Block' end)
as action, sum(totalnum) as totalnum from ###(select $flex_timestamp as timestamp, attack,
coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, severity, service,
dstip, srcip, (case when action in ('clear_session', 'pass_session') then 1 when action in
('reset', 'reset_client', 'reset_server') then 2 else 3 end) as action_flag, count(*) as totalnum
from $log-attack where $filter and subtype = 'ips' and nullifna(attack) is not null and severity
is not null group by timestamp, attack, user_src, severity, service, dstip, srcip, action_flag
order by totalnum desc)### t group by action order by totalnum desc
```

Dataset Name	Description	Log Category
threat-Top-CC-Sites-Connections-by-Risk-Level	Top Connections to C&C Sites by Risk Level	attack

```
select
  severity,
  attack,
  sum(
    case when action_flag != 3 then totalnum else 0 end
  ) as total_allow,
  sum(
    case when action_flag = 3 then totalnum else 0 end
  ) as total_block,
  sum(totalnum) as totalnum
from
  ###(select $flex_timestamp as timestamp, attack, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, severity, service, dstip, srcip, (case when action in
('clear_session', 'pass_session') then 1 when action in ('reset', 'reset_client', 'reset_server')
then 2 else 3 end) as action_flag, count(*) as totalnum from $log-attack where $filter and subtype
= 'ips' and nullifna(attack) is not null and severity is not null group by timestamp, attack,
user_src, severity, service, dstip, srcip, action_flag order by totalnum desc)### t group by
severity, attack order by severity desc, totalnum desc
```

Dataset Name	Description	Log Category
threat-Top-CC-Sites-Users-by-Occurrences	Top Connection to C&C Sites Users by Occurrences	attack

```
select
  user_src,
  sum(totalnum) as totalnum
from
  ###(select $flex_timestamp as timestamp, attack, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, severity, service, dstip, srcip, (case when action in
('clear_session', 'pass_session') then 1 when action in ('reset', 'reset_client', 'reset_server')
then 2 else 3 end) as action_flag, count(*) as totalnum from $log-attack where $filter and subtype
= 'ips' and nullifna(attack) is not null and severity is not null group by timestamp, attack,
user_src, severity, service, dstip, srcip, action_flag order by totalnum desc)### t group by user_
src order by totalnum desc
```

Dataset Name	Description	Log Category
threat-Top-Successful-CC-Sites-Connections-by-Risk-Level	Top Successful Connections to C&C Sites by Risk Level	attack

```
select
  severity,
  attack,
  service,
  count(distinct srcip) as sources,
  sum(totalnum) as totalnum
from
  ###(select $flex_timestamp as timestamp, attack, coalesce(nullifna(`user`), nullifna
```

```
(`unauthuser`), ipstr(`srcip`)) as user_src, severity, service, dstip, srcip, (case when action in
('clear_session', 'pass_session') then 1 when action in ('reset', 'reset_client', 'reset_server')
then 2 else 3 end) as action_flag, count(*) as totalnum from $log-attack where $filter and subtype
= 'ips' and nullifna(attack) is not null and severity is not null group by timestamp, attack,
user_src, severity, service, dstip, srcip, action_flag order by totalnum desc)### t where action_
flag = 1 group by severity, attack, service order by severity desc
```

Dataset Name	Description	Log Category
threat-Third-Stage-Timeline	Cyber Kill Chain Third Stage Timeline	traffic

```
select
  $flex_timestamp(timestamp) as hodex,
  type,
  sum(totalnum) as totalnum
from
  (
    (
      select
        timestamp,
        & #039;DNS Resolutions for C&C Domains' as type, sum(total_num) as totalnum from ###
(select $flex_timestamp as timestamp, coalesce(botnetdomain, ipstr(botnetip)) as domain, qname,
cast('Botnet C&C' as char(32)) as malware_type, (case when action='block' then 'Blocked' when
action='redirect' then 'Redirected' else 'Passed' end) as action, srcip, coalesce(nullifna
(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, (CASE WHEN level IN ('critical',
'alert', 'emergency') THEN 5 WHEN level='error' THEN 4 WHEN level='warning' THEN 3 WHEN
level='notice' THEN 2 ELSE 1 END) as sevid, coalesce(botnetdomain, ipstr(botnetip)) as sources_s,
count(*) as total_num from $log-dns where $filter and (botnetdomain is not null or botnetip is not
null) group by timestamp, domain, qname, action, srcip, user_src, sevid order by sevid desc)### t
group by timestamp, type order by totalnum desc) union all (select timestamp, 'Connections to C&C
Sites' as type, sum(totalnum) as totalnum from ###(select $flex_timestamp as timestamp, attack,
coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, severity, service,
dstip, srcip, (case when action in ('clear_session', 'pass_session') then 1 when action in
('reset', 'reset_client', 'reset_server') then 2 else 3 end) as action_flag, count(*) as totalnum
from $log-attack where $filter and subtype = 'ips' and nullifna(attack) is not null and severity
is not null group by timestamp, attack, user_src, severity, service, dstip, srcip, action_flag
order by totalnum desc)### t group by timestamp, type order by totalnum desc)) t group by hodex,
type order by hodex
```

Dataset Name	Description	Log Category
ztna-Public-Private-SaaS-App-Bandwidth-Timeline	Public, Private and SaaS Apps Bandwidth Timeline	traffic

```
with qry as (
  select
    $flex_timestamp(timestamp) as hodex,
    app_group,
    saasname,
```

```
    sum(bandwidth) as bandwidth
  from
    ###(select $flex_timestamp as timestamp, coalesce(nullifna(`user`), nullifna(`unauthuser`),
ipstr(`srcip`)) as user_src, srcip, dstip, accessproxy, saasname, app_group_name(app) as app_
group, (CASE WHEN (logflag&2>0) THEN msg ELSE null END) posture, max(coalesce(srcname, srcmac)) as
dev_src, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block, sum(CASE WHEN
(logflag&1>0) THEN 1 ELSE 0 END) as sessions, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum
(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in from $log-traffic where $filter and (logflag&
(1|32)>0) and accessproxy IS NOT NULL group by timestamp, user_src, srcip, dstip, accessproxy,
saasname, app_group, posture /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t
group by hodex, app_group, saasname) select hodex, type, sum(bandwidth) as bandwidth from ((select
hodex, 'Private Access' as type, sum(bandwidth) as bandwidth from qry t group by hodex, type order
by bandwidth desc) union all (select hodex, 'Public Access' as type, sum(bandwidth) as bandwidth
from qry t1 inner join shadowit_application t2 on t1.app_group=t2.appname where app_group is not
null group by hodex, type order by bandwidth desc) union all (select hodex, 'SaaS Access' as type,
sum(bandwidth) as bandwidth from qry t where saasname is not null group by hodex, type order by
bandwidth desc)) t group by hodex, type order by hodex
```

Dataset Name	Description	Log Category
ztna-Top-Private-Apps-by-Session	Top Private Applications by Session	traffic

```
select
  accessproxy,
  dstip,
  session_block ||& #039;/'||session_pass as sessions from (select accessproxy, max(dstip) as
dstip, sum(sessions) as sessions, (sum(sessions)-sum(session_block)) as session_pass, sum(session_
block) as session_block from ###(select $flex_timestamp as timestamp, coalesce(nullifna(`user`),
nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, srcip, dstip, accessproxy, saasname, app_
group_name(app) as app_group, (CASE WHEN (logflag&2>0) THEN msg ELSE null END) posture, max
(coalesce(srcname, srcmac)) as dev_src, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS
session_block, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions, sum(coalesce(sentdelta,
sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(sentdelta, sentbyte, 0))
as traffic_out, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in from $log-traffic where
$filter and (logflag&(1|32)>0) and accessproxy IS NOT NULL group by timestamp, user_src, srcip,
dstip, accessproxy, saasname, app_group, posture /*SkipSTART*/order by sessions desc, bandwidth
desc/*SkipEND*/)### t group by accessproxy order by sessions desc) t
```

Dataset Name	Description	Log Category
ztna-Top-Public-Apps-by-Session	Top Public Applications by Session	traffic

```
select
  appid,
  app_group,
  session_block ||& #039;/'||session_pass as sessions from (select appid, app_group, sum(sessions)
as sessions, (sum(sessions)-sum(session_block)) as session_pass, sum(session_block) as session_
block from ###(select $flex_timestamp as timestamp, coalesce(nullifna(`user`), nullifna
```

```
(`unauthuser`), ipstr(`srcip`)) as user_src, srcip, dstip, accessproxy, saasname, app_group_name
(app) as app_group, (CASE WHEN (logflag&2>0) THEN msg ELSE null END) posture, max(coalesce
(srcname, srcmac)) as dev_src, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block,
sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions, sum(coalesce(sentdelta, sentbyte,
0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(sentdelta, sentbyte, 0)) as
traffic_out, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in from $log-traffic where $filter
and (logflag&(1|32)>0) and accessproxy IS NOT NULL group by timestamp, user_src, srcip, dstip,
accessproxy, saasname, app_group, posture /*SkipSTART*/order by sessions desc, bandwidth
desc/*SkipEND*/)### t1 inner join shadowit_application t2 on t1.app_group=t2.appname where app_
group is not null group by appid, app_group order by sessions desc) t
```

Dataset Name	Description	Log Category
ztna-Top-SaaS-Apps-by-Blocked-Allowed-Session	Top SaaS Applications by Blocked and Allowed Session	traffic

```
select
  appid,
  saasname,
  session_block ||& #039;/'||session_pass as sessions from (select appid, saasname, sum(sessions)
as sessions, (sum(sessions)-sum(session_block)) as session_pass, sum(session_block) as session_
block from ###(select $flex_timestamp as timestamp, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, srcip, dstip, accessproxy, saasname, app_group_name
(app) as app_group, (CASE WHEN (logflag&2>0) THEN msg ELSE null END) posture, max(coalesce
(srcname, srcmac)) as dev_src, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block,
sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions, sum(coalesce(sentdelta, sentbyte,
0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(sentdelta, sentbyte, 0)) as
traffic_out, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in from $log-traffic where $filter
and (logflag&(1|32)>0) and accessproxy IS NOT NULL group by timestamp, user_src, srcip, dstip,
accessproxy, saasname, app_group, posture /*SkipSTART*/order by sessions desc, bandwidth
desc/*SkipEND*/)### t1 left join shadowit_application t2 on lower(t1.saasname)=lower(t2.appname)
where saasname is not null group by appid, saasname order by sessions desc) t
```

Dataset Name	Description	Log Category
ztna-Top-Public-Private-SaaS-Access-Failure-Timeline	Top Private, Public and SaaS Access Failure by Count Timeline	traffic

```
with qry as (
  select
    $flex_timescale(timestamp) as hodex,
    accessproxy,
    app_group,
    saasname,
    sum(session_block) as session_block
  from
    ###(select $flex_timestamp as timestamp, coalesce(nullifna(`user`), nullifna(`unauthuser`),
ipstr(`srcip`)) as user_src, srcip, dstip, accessproxy, saasname, app_group_name(app) as app_
group, (CASE WHEN (logflag&2>0) THEN msg ELSE null END) posture, max(coalesce(srcname, srcmac)) as
```

```
dev_src, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block, sum(CASE WHEN
(logflag&1>0) THEN 1 ELSE 0 END) as sessions, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum
(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in from $log-traffic where $filter and (logflag&
(1|32)>0) and accessproxy IS NOT NULL group by timestamp, user_src, srcip, dstip, accessproxy,
saasname, app_group, posture /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t
group by hodex, accessproxy, app_group, saasname) select hodex, t1.app, t1.session_block from
(select hodex, app, sum(session_block) as session_block from ((select hodex, accessproxy as app,
sum(session_block) as session_block from qry t group by hodex, app order by session_block desc)
union all (select hodex, app_group as app, sum(session_block) as session_block from qry t1 inner
join shadowit_application t2 on t1.app_group=t2.appname where app_group is not null group by
hodex, app order by session_block desc) union all (select hodex, saasname as app, sum(session_
block) as session_block from qry t where saasname is not null group by hodex, app order by
session_block desc)) t group by hodex, app order by session_block desc) t1 inner join (select app,
sum(session_block) as session_block from ((select accessproxy as app, sum(session_block) as
session_block from qry t group by app order by session_block desc) union all (select app_group as
app, sum(session_block) as session_block from qry t1 inner join shadowit_application t2 on t1.app_
group=t2.appname where app_group is not null group by app order by session_block desc) union all
(select saasname as app, sum(session_block) as session_block from qry t where saasname is not null
group by app order by session_block desc)) t group by app having sum(session_block)>0 order by
session_block desc limit $ddown-top) t2 on t1.app=t2.app order by hodex
```

Dataset Name	Description	Log Category
ztna-Top-Devices-by-Failed-Posture-Check	Top Known Devices with Failed Posture by Count	traffic

```
select
  dev_src,
  sum(session_block) as session_block
from
  ###(select $flex_timestamp as timestamp, coalesce(nullifna(`user`), nullifna(`unauthuser`),
ipstr(`srcip`)) as user_src, srcip, dstip, accessproxy, saasname, app_group_name(app) as app_
group, (CASE WHEN (logflag&2>0) THEN msg ELSE null END) posture, max(coalesce(srcname, srcmac)) as
dev_src, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block, sum(CASE WHEN
(logflag&1>0) THEN 1 ELSE 0 END) as sessions, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum
(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in from $log-traffic where $filter and (logflag&
(1|32)>0) and accessproxy IS NOT NULL group by timestamp, user_src, srcip, dstip, accessproxy,
saasname, app_group, posture /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t
where dev_src is not null group by dev_src having sum(session_block)>0 order by session_block desc
```

Dataset Name	Description	Log Category
ztna-Top-Failed-Posture-Check-by-Count	Top Failed Posture Checkes by Count	traffic

```
select
  posture,
```

```
  sum(session_block) as session_block
from
  ###(select $flex_timestamp as timestamp, coalesce(nullifna(`user`), nullifna(`unauthuser`),
ipstr(`srcip`)) as user_src, srcip, dstip, accessproxy, saasname, app_group_name(app) as app_
group, (CASE WHEN (logflag&2>0) THEN msg ELSE null END) posture, max(coalesce(srcname, srcmac)) as
dev_src, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block, sum(CASE WHEN
(logflag&1>0) THEN 1 ELSE 0 END) as sessions, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum
(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in from $log-traffic where $filter and (logflag&
(1|32)>0) and accessproxy IS NOT NULL group by timestamp, user_src, srcip, dstip, accessproxy,
saasname, app_group, posture /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t
where posture is not null group by posture having sum(session_block)>0 order by session_block desc
```

Dataset Name	Description	Log Category
ztna-Top-SaaS-Apps-by-Bandwidth	Top SaaS Applications by Bandwidth	traffic

```
select
  saasname,
  sum(bandwidth) as bandwidth
from
  ###(select $flex_timestamp as timestamp, coalesce(nullifna(`user`), nullifna(`unauthuser`),
ipstr(`srcip`)) as user_src, srcip, dstip, accessproxy, saasname, app_group_name(app) as app_
group, (CASE WHEN (logflag&2>0) THEN msg ELSE null END) posture, max(coalesce(srcname, srcmac)) as
dev_src, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block, sum(CASE WHEN
(logflag&1>0) THEN 1 ELSE 0 END) as sessions, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum
(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in from $log-traffic where $filter and (logflag&
(1|32)>0) and accessproxy IS NOT NULL group by timestamp, user_src, srcip, dstip, accessproxy,
saasname, app_group, posture /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t
where saasname is not null group by saasname order by bandwidth desc
```

Dataset Name	Description	Log Category
ztna-Top-SaaS-Apps-by-User-Count	Top SaaS Applications by User Count	traffic

```
select
  saasname,
  count(distinct user_src) as total_num
from
  ###(select $flex_timestamp as timestamp, coalesce(nullifna(`user`), nullifna(`unauthuser`),
ipstr(`srcip`)) as user_src, srcip, dstip, accessproxy, saasname, app_group_name(app) as app_
group, (CASE WHEN (logflag&2>0) THEN msg ELSE null END) posture, max(coalesce(srcname, srcmac)) as
dev_src, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block, sum(CASE WHEN
(logflag&1>0) THEN 1 ELSE 0 END) as sessions, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum
(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in from $log-traffic where $filter and (logflag&
(1|32)>0) and accessproxy IS NOT NULL group by timestamp, user_src, srcip, dstip, accessproxy,
```

```
saasname, app_group, posture /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t
where saasname is not null group by saasname order by total_num desc
```

Dataset Name	Description	Log Category
ztna-Top-SaaS-Apps-by-Session	Top SaaS Applications by Session	traffic

```
select
  saasname,
  sum(sessions) as sessions,
  (
    sum(sessions)- sum(session_block)
  ) as session_pass,
  sum(session_block) as session_block
from
  ###(select $flex_timestamp as timestamp, coalesce(nullifna(`user`), nullifna(`unauthuser`),
ipstr(`srcip`)) as user_src, srcip, dstip, accessproxy, saasname, app_group_name(app) as app_
group, (CASE WHEN (logflag&2>0) THEN msg ELSE null END) posture, max(coalesce(srcname, srcmac)) as
dev_src, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block, sum(CASE WHEN
(logflag&1>0) THEN 1 ELSE 0 END) as sessions, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum
(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in from $log-traffic where $filter and (logflag&
(1|32)>0) and accessproxy IS NOT NULL group by timestamp, user_src, srcip, dstip, accessproxy,
saasname, app_group, posture /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t
where saasname is not null group by saasname order by sessions desc
```

Dataset Name	Description	Log Category
ztna-Top-SaaS-App-Users-by-Bandwidth-Session	Top SaaS App Users by Bandwidth and Session	traffic

```
select
  user_src,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out,
  sum(sessions) as sessions,
  (
    sum(sessions)- sum(session_block)
  ) as session_pass,
  sum(session_block) as session_block
from
  ###(select $flex_timestamp as timestamp, coalesce(nullifna(`user`), nullifna(`unauthuser`),
ipstr(`srcip`)) as user_src, srcip, dstip, accessproxy, saasname, app_group_name(app) as app_
group, (CASE WHEN (logflag&2>0) THEN msg ELSE null END) posture, max(coalesce(srcname, srcmac)) as
dev_src, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block, sum(CASE WHEN
(logflag&1>0) THEN 1 ELSE 0 END) as sessions, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum
(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in from $log-traffic where $filter and (logflag&
```

```
(1|32)>0) and accessproxy IS NOT NULL group by timestamp, user_src, srcip, dstip, accessproxy,
saasname, app_group, posture /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t
where saasname is not null group by user_src order by bandwidth desc, sessions desc
```

Dataset Name	Description	Log Category
ztna-Top-SaaS-Apps-by-Blocked-Session	Top SaaS Applications by Blocked Session	traffic

```
select
  saasname,
  sum(session_block) as session_block
from
  ###(select $flex_timestamp as timestamp, coalesce(nullifna(`user`), nullifna(`unauthuser`),
ipstr(`srcip`)) as user_src, srcip, dstip, accessproxy, saasname, app_group_name(app) as app_
group, (CASE WHEN (logflag&2>0) THEN msg ELSE null END) posture, max(coalesce(srcname, srcmac)) as
dev_src, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block, sum(CASE WHEN
(logflag&1>0) THEN 1 ELSE 0 END) as sessions, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum
(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in from $log-traffic where $filter and (logflag&
(1|32)>0) and accessproxy IS NOT NULL group by timestamp, user_src, srcip, dstip, accessproxy,
saasname, app_group, posture /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t
where saasname is not null group by saasname having sum(session_block)>0 order by session_block
desc
```

Dataset Name	Description	Log Category
ztna-Top-SaaS-App-Users-by-Blocked-Session	Top SaaS App Users by Blocked Session	traffic

```
select
  user_src,
  sum(session_block) as session_block,
  string_agg(
    distinct posture,
    & #039;, ') as postures from ###(select $flex_timestamp as timestamp, coalesce(nullifna
(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, srcip, dstip, accessproxy,
saasname, app_group_name(app) as app_group, (CASE WHEN (logflag&2>0) THEN msg ELSE null END)
posture, max(coalesce(srcname, srcmac)) as dev_src, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0
END)) AS session_block, sum(CASE WHEN (logflag&1>0) THEN 1 ELSE 0 END) as sessions, sum(coalesce
(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(sentdelta,
sentbyte, 0)) as traffic_out, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in from $log-
traffic where $filter and (logflag&(1|32)>0) and accessproxy IS NOT NULL group by timestamp, user_
src, srcip, dstip, accessproxy, saasname, app_group, posture /*SkipSTART*/order by sessions desc,
bandwidth desc/*SkipEND*/)### t where saasname is not null group by user_src having sum(session_
block)>0 order by session_block desc
```

Dataset Name	Description	Log Category
ztna-Top-Private-Public-Apps-by-Bandwidth	Top Private and Public Applications by Bandwidth	traffic

```
with qry as (
  select
    accessproxy,
    app_group,
    sum(bandwidth) as bandwidth
  from
    ###(select $flex_timestamp as timestamp, coalesce(nullifna(`user`), nullifna(`unauthuser`),
ipstr(`srcip`)) as user_src, srcip, dstip, accessproxy, saasname, app_group_name(app) as app_
group, (CASE WHEN (logflag&2>0) THEN msg ELSE null END) posture, max(coalesce(srcname, srcmac)) as
dev_src, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block, sum(CASE WHEN
(logflag&1>0) THEN 1 ELSE 0 END) as sessions, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum
(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in from $log-traffic where $filter and (logflag&
(1|32)>0) and accessproxy IS NOT NULL group by timestamp, user_src, srcip, dstip, accessproxy,
saasname, app_group, posture /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t
group by accessproxy, app_group) select app, sum(bandwidth) as bandwidth from ((select accessproxy
as app, sum(bandwidth) as bandwidth from qry t group by app order by bandwidth desc) union all
(select app_group as app, sum(bandwidth) as bandwidth from qry t1 inner join shadowit_application
t2 on t1.app_group=t2.appname where app_group is not null group by app order by bandwidth desc)) t
group by app order by bandwidth desc
```

Dataset Name	Description	Log Category
ztna-Top-Private-Public-Apps-by-User-Count	Top Private and Public Applications by User Count	traffic

```
with qry as (
  select
    user_src,
    accessproxy,
    app_group,
    sum(bandwidth) as bandwidth
  from
    ###(select $flex_timestamp as timestamp, coalesce(nullifna(`user`), nullifna(`unauthuser`),
ipstr(`srcip`)) as user_src, srcip, dstip, accessproxy, saasname, app_group_name(app) as app_
group, (CASE WHEN (logflag&2>0) THEN msg ELSE null END) posture, max(coalesce(srcname, srcmac)) as
dev_src, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block, sum(CASE WHEN
(logflag&1>0) THEN 1 ELSE 0 END) as sessions, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum
(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in from $log-traffic where $filter and (logflag&
(1|32)>0) and accessproxy IS NOT NULL group by timestamp, user_src, srcip, dstip, accessproxy,
saasname, app_group, posture /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t
group by user_src, accessproxy, app_group) select app, count(distinct user_src) as total_num from
((select accessproxy as app, user_src, sum(bandwidth) as bandwidth from qry t group by app, user_
src order by bandwidth desc) union all (select app_group as app, user_src, sum(bandwidth) as
```

```
bandwidth from qry t1 inner join shadowit_application t2 on t1.app_group=t2.appname where app_
group is not null group by app, user_src order by bandwidth desc)) t group by app order by total_
num desc
```

Dataset Name	Description	Log Category
ztna-Top-Private-Public-Apps-by-Session	Top Private and Public Applications by Session	traffic

```
with qry as (
  select
    accessproxy,
    app_group,
    sum(sessions) as sessions
  from
    ###(select $flex_timestamp as timestamp, coalesce(nullifna(`user`), nullifna(`unauthuser`),
ipstr(`srcip`)) as user_src, srcip, dstip, accessproxy, saasname, app_group_name(app) as app_
group, (CASE WHEN (logflag&2>0) THEN msg ELSE null END) posture, max(coalesce(srcname, srcmac)) as
dev_src, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block, sum(CASE WHEN
(logflag&1>0) THEN 1 ELSE 0 END) as sessions, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum
(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in from $log-traffic where $filter and (logflag&
(1|32)>0) and accessproxy IS NOT NULL group by timestamp, user_src, srcip, dstip, accessproxy,
saasname, app_group, posture /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t
group by accessproxy, app_group) select app, sum(sessions) as sessions from ((select accessproxy
as app, sum(sessions) as sessions from qry t group by app order by sessions desc) union all
(select app_group as app, sum(sessions) as sessions from qry t1 inner join shadowit_application t2
on t1.app_group=t2.appname where app_group is not null group by app order by sessions desc)) t
group by app order by sessions desc
```

Dataset Name	Description	Log Category
ztna-Top-Private-Public-App-Users-by-Bandwidth-Session	Top Private and Public App Users by Bandwidth and Session	traffic

```
with qry as (
  select
    accessproxy,
    app_group,
    user_src,
    sum(bandwidth) as bandwidth,
    sum(traffic_in) as traffic_in,
    sum(traffic_out) as traffic_out,
    sum(session_block) as session_block,
    sum(sessions) as sessions
  from
    ###(select $flex_timestamp as timestamp, coalesce(nullifna(`user`), nullifna(`unauthuser`),
ipstr(`srcip`)) as user_src, srcip, dstip, accessproxy, saasname, app_group_name(app) as app_
group, (CASE WHEN (logflag&2>0) THEN msg ELSE null END) posture, max(coalesce(srcname, srcmac)) as
```

```
dev_src, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block, sum(CASE WHEN
(logflag&1>0) THEN 1 ELSE 0 END) as sessions, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum
(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in from $log-traffic where $filter and (logflag&
(1|32)>0) and accessproxy IS NOT NULL group by timestamp, user_src, srcip, dstip, accessproxy,
saasname, app_group, posture /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t
group by user_src, accessproxy, app_group) select user_src, sum(bandwidth) as bandwidth, sum
(traffic_in) as traffic_in, sum(traffic_out) as traffic_out, sum(sessions) as sessions, (sum
(sessions)-sum(session_block)) as session_pass, sum(session_block) as session_block from ((select
accessproxy as app, user_src, sum(bandwidth) as bandwidth, sum(sessions) as sessions, sum(traffic_
in) as traffic_in, sum(traffic_out) as traffic_out, sum(session_block) as session_block from qry t
group by app, user_src order by bandwidth desc, sessions desc) union all (select app_group as app,
user_src, sum(bandwidth) as bandwidth, sum(sessions) as sessions, sum(traffic_in) as traffic_in,
sum(traffic_out) as traffic_out, sum(session_block) as session_block from qry t1 inner join
shadowit_application t2 on t1.app_group=t2.appname where app_group is not null group by app, user_
src order by bandwidth desc, sessions desc)) t group by user_src order by bandwidth desc, sessions
desc
```

Dataset Name	Description	Log Category
ztna-Top-Private-Public-Apps-by-Blocked-Session	Top Private and Public Applications by Blocked Session	traffic

```
with qry as (
  select
    accessproxy,
    app_group,
    sum(session_block) as session_block
  from
    ###(select $flex_timestamp as timestamp, coalesce(nullifna(`user`), nullifna(`unauthuser`),
ipstr(`srcip`)) as user_src, srcip, dstip, accessproxy, saasname, app_group_name(app) as app_
group, (CASE WHEN (logflag&2>0) THEN msg ELSE null END) posture, max(coalesce(srcname, srcmac)) as
dev_src, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block, sum(CASE WHEN
(logflag&1>0) THEN 1 ELSE 0 END) as sessions, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum
(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in from $log-traffic where $filter and (logflag&
(1|32)>0) and accessproxy IS NOT NULL group by timestamp, user_src, srcip, dstip, accessproxy,
saasname, app_group, posture /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t
group by accessproxy, app_group) select app, sum(session_block) as session_block from ((select
accessproxy as app, sum(session_block) as session_block from qry t group by app order by session_
block desc) union all (select app_group as app, sum(session_block) as session_block from qry t1
inner join shadowit_application t2 on t1.app_group=t2.appname where app_group is not null group by
app order by session_block desc)) t group by app having sum(session_block)>0 order by session_
block desc
```

Dataset Name	Description	Log Category
ztna-Top-Private-Public-App-Users-by-Blocked-Session	Top Private and Public App Users by Blocked Session	traffic

```
with qry as (
  select
    user_src,
    accessproxy,
    app_group,
    posture,
    sum(session_block) as session_block
  from
    ###(select $flex_timestamp as timestamp, coalesce(nullifna(`user`), nullifna(`unauthuser`),
ipstr(`srcip`)) as user_src, srcip, dstip, accessproxy, saasname, app_group_name(app) as app_
group, (CASE WHEN (logflag&2>0) THEN msg ELSE null END) posture, max(coalesce(srcname, srcmac)) as
dev_src, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) AS session_block, sum(CASE WHEN
(logflag&1>0) THEN 1 ELSE 0 END) as sessions, sum(coalesce(sentdelta, sentbyte, 0)+coalesce
(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum
(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in from $log-traffic where $filter and (logflag&
(1|32)>0) and accessproxy IS NOT NULL group by timestamp, user_src, srcip, dstip, accessproxy,
saasname, app_group, posture /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t
group by user_src, accessproxy, app_group, posture) select user_src, sum(session_block) as
session_block, string_agg(distinct posture, ', ') as postures from ((select accessproxy as app,
user_src, posture, sum(session_block) as session_block from qry t group by app, user_src, posture
having sum(session_block)>0) union all (select app_group as app, user_src, posture, sum(session_
block) as session_block from qry t1 inner join shadowit_application t2 on t1.app_group=t2.appname
where app_group is not null group by app, user_src, posture having sum(session_block)>0)) t group
by user_src order by session_block desc
```

Dataset Name	Description	Log Category
dlp-Total-DLP-Events-by-Destination-Countries	Total DLP Events by Destination Countries	dlp

```
select
  dstcountry,
  sum(sessions) as sessions
from
  (
    (
      select
        dstcountry,
        sum(sessions) as sessions
      from
        ###(select $flex_timestamp as timestamp, hostname, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry, (case when action = 'block' then 'Block'
else 'Allow' end) as action_type, severity, coalesce(sensitivity, 'Unclassified') as sen, service,
profile, (case when direction='incoming' then 'Download' else 'Upload' end) as traffic_direction,
filename, sum(filesize) as filesize, count(*) as sessions from $log-dlp where $filter and hostname
is not null and action in ('log-only', 'exempt', 'block') group by timestamp, hostname, user_src,
dstcountry, action_type, severity, sen, service, profile, traffic_direction, filename order by
sessions desc)### t where dstcountry is not null group by dstcountry) union all (select
dstcountry, sum(sessions) as sessions from ###(select $flex_timestamp as timestamp, app, siappid,
filename, profile, service, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as
user_src, dstcountry, action, sum(case when cloudaction='upload' then coalesce(filesize, 0) else 0
```

```
end) as upload_size, sum(case when cloudaction='download' or (cloudaction='others' and app like
'%Video%') then coalesce(filesize, 0) else 0 end) as download_size, sum(filesize) as filesize,
count(*) as sessions from $log-app-ctrl where $filter and siappid is not null and action in
('pass', 'block', 'reset') group by timestamp, app, siappid, filename, profile, service, action,
user_src, dstcountry order by sessions desc)### t1 inner join shadowit_application t2 on
t1.siappid=t2.appid where dstcountry is not null and filename is not null group by dstcountry)
union all (select dstcountry, sum(sessions) as sessions from ###(select $flex_timestamp as
timestamp, saasname, srcip, dstip, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr
(`srcip`)) as user_src, dstcountry, service, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END)) as
session_block, count(*) as sessions, sum(case when accessctrl='upload' THEN coalesce(sentbyte, 0)
ELSE 0 END) AS upload_size, sum(case when accessctrl='download' THEN coalesce(rcvdbyte, 0) ELSE 0
END) AS download_size from $log-traffic where $filter and (logflag&1>0) and saasname is not null
group by timestamp, saasname, srcip, dstip, user_src, dstcountry, service order by sessions
desc)### t where dstcountry is not null group by dstcountry)) t where dstcountry <> 'Reserved'
group by dstcountry order by sessions desc
```

Dataset Name	Description	Log Category
dlp-Trends-by-Action-Timeline	DLP Trends by Action Timeline	dlp

```
select
  hodex,
  action_type,
  sum(sessions) as sessions
from
  (
    (
      select
        $flex_timestamp(timestamp) as hodex,
        action_type,
        sum(sessions) as sessions
      from
        ###(select $flex_timestamp as timestamp, hostname, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry, (case when action = 'block' then 'Block'
else 'Allow' end) as action_type, severity, coalesce(sensitivity, 'Unclassified') as sen, service,
profile, (case when direction='incoming' then 'Download' else 'Upload' end) as traffic_direction,
filename, sum(filesize) as filesize, count(*) as sessions from $log-dlp where $filter and hostname
is not null and action in ('log-only', 'exempt', 'block') group by timestamp, hostname, user_src,
dstcountry, action_type, severity, sen, service, profile, traffic_direction, filename order by
sessions desc)### t group by hodex, action_type) union all (select $flex_timestamp(timestamp) as
hodex, (case when action != 'pass' then 'Block' else 'Allow' end) as action_type, sum(sessions) as
sessions from ###(select $flex_timestamp as timestamp, app, siappid, filename, profile, service,
coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry,
action, sum(case when cloudaction='upload' then coalesce(filesize, 0) else 0 end) as upload_size,
sum(case when cloudaction='download' or (cloudaction='others' and app like '%Video%') then
coalesce(filesize, 0) else 0 end) as download_size, sum(filesize) as filesize, count(*) as
sessions from $log-app-ctrl where $filter and siappid is not null and action in ('pass', 'block',
'reset') group by timestamp, app, siappid, filename, profile, service, action, user_src,
dstcountry order by sessions desc)### t1 inner join shadowit_application t2 on t1.siappid=t2.appid
where filename is not null group by hodex, action_type) union all (select hodex, unnest(action) as
action_type, unnest(session) as sessions from (select $flex_timestamp(timestamp) as hodex, array
```

```
['Block', 'Allow'] as action, array[sum(session_block), (sum(sessions)-sum(session_block))] as
session from ###(select $flex_timestamp as timestamp, saasname, srcip, dstip, coalesce(nullifna
(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry, service, sum((CASE WHEN
(logflag&2>0) THEN 1 ELSE 0 END)) as session_block, count(*) as sessions, sum(case when
accessctrl='upload' THEN coalesce(sentbyte, 0) ELSE 0 END) AS upload_size, sum(case when
accessctrl='download' THEN coalesce(rcvdbyte, 0) ELSE 0 END) AS download_size from $log-traffic
where $filter and (logflag&1>0) and saasname is not null group by timestamp, saasname, srcip,
dstip, user_src, dstcountry, service order by sessions desc)### t group by hodex, action) t)) t
group by hodex, action_type order by hodex
```

Dataset Name	Description	Log Category
dlp-Upload-vs-Download-Trends-Timeline	Upload vs Download DLP Trends Timeline	dlp

```
select
  hodex,
  direction,
  sum(bytes) as bytes
from
  (
    (
      select
        $flex_timestamp(timestamp) as hodex,
        traffic_direction as direction,
        sum(filesize) as bytes
      from
        ###(select $flex_timestamp as timestamp, hostname, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry, (case when action = 'block' then 'Block'
else 'Allow' end) as action_type, severity, coalesce(sensitivity, 'Unclassified') as sen, service,
profile, (case when direction='incoming' then 'Download' else 'Upload' end) as traffic_direction,
filename, sum(filesize) as filesize, count(*) as sessions from $log-dlp where $filter and hostname
is not null and action in ('log-only', 'exempt', 'block') group by timestamp, hostname, user_src,
dstcountry, action_type, severity, sen, service, profile, traffic_direction, filename order by
sessions desc)### t group by hodex, direction) union all (select hodex, unnest(directions) as
direction, unnest(bytes) as bytes from (select $flex_timestamp(timestamp) as hodex, array
['Upload', 'Download'] as directions, array[sum(upload_size), sum(download_size)] as bytes from
###(select $flex_timestamp as timestamp, app, siappid, filename, profile, service, coalesce
(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry, action, sum
(case when cloudaction='upload' then coalesce(filesize, 0) else 0 end) as upload_size, sum(case
when cloudaction='download' or (cloudaction='others' and app like '%Video%') then coalesce
(filesize, 0) else 0 end) as download_size, sum(filesize) as filesize, count(*) as sessions from
$log-app-ctrl where $filter and siappid is not null and action in ('pass', 'block', 'reset') group
by timestamp, app, siappid, filename, profile, service, action, user_src, dstcountry order by
sessions desc)### t1 inner join shadowit_application t2 on t1.siappid=t2.appid where filename is
not null group by hodex, directions) t) union all (select hodex, unnest(directions) as direction,
unnest(bytes) as bytes from (select $flex_timestamp(timestamp) as hodex, array['Upload',
'Download'] as directions, array[sum(upload_size), sum(download_size)] as bytes from ###(select
$flex_timestamp as timestamp, saasname, srcip, dstip, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry, service, sum((CASE WHEN (logflag&2>0)
THEN 1 ELSE 0 END)) as session_block, count(*) as sessions, sum(case when accessctrl='upload' THEN
```

```
coalesce(sentbyte, 0) ELSE 0 END) AS upload_size, sum(case when accessctrl='download' THEN
coalesce(rcvdbyte, 0) ELSE 0 END) AS download_size from $log-traffic where $filter and
(logflag&1>0) and saasname is not null group by timestamp, saasname, srcip, dstip, user_src,
dstcountry, service order by sessions desc)### t group by hodex, directions) t)) t group by hodex,
direction order by hodex
```

Dataset Name	Description	Log Category
dlp-Top-DLP-Events-by-File-Size	Top DLP Events by File Size	dlp

```
select
  severity,
  hostname,
  user_src,
  sensitivity,
  service,
  action_type,
  sum(bytes) as bytes
from
  (
    (
      select
        severity,
        hostname,
        user_src,
        sen as sensitivity,
        service,
        action_type,
        sum(filesize) as bytes
      from
        ###(select $flex_timestamp as timestamp, hostname, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry, (case when action = 'block' then 'Block'
else 'Allow' end) as action_type, severity, coalesce(sensitivity, 'Unclassified') as sen, service,
profile, (case when direction='incoming' then 'Download' else 'Upload' end) as traffic_direction,
filename, sum(filesize) as filesize, count(*) as sessions from $log-dlp where $filter and hostname
is not null and action in ('log-only', 'exempt', 'block') group by timestamp, hostname, user_src,
dstcountry, action_type, severity, sen, service, profile, traffic_direction, filename order by
sessions desc)### t group by severity, hostname, user_src, sensitivity, service, action_type
having sum(filesize)>0) union all (select (case when riskscore between 1 and 15 then 'info' when
riskscore between 16 and 30 then 'low' when riskscore between 31 and 50 then 'medium' when
riskscore between 51 and 70 then 'high' when riskscore between 71 and 100 then 'citical' else
'info' end)::fgt_enum_severity as severity, app as hostname, user_src, 'Unclassified' as
sensitivity, service, action::text as action_type, sum(filesize) as bytes from ###(select $flex_
timestamp as timestamp, app, siappid, filename, profile, service, coalesce(nullifna(`user`),
nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry, action, sum(case when
cloudaction='upload' then coalesce(filesize, 0) else 0 end) as upload_size, sum(case when
cloudaction='download' or (cloudaction='others' and app like '%Video%') then coalesce(filesize, 0)
else 0 end) as download_size, sum(filesize) as filesize, count(*) as sessions from $log-app-ctrl
where $filter and siappid is not null and action in ('pass', 'block', 'reset') group by timestamp,
app, siappid, filename, profile, service, action, user_src, dstcountry order by sessions desc)###
t1 inner join shadowit_application t2 on t1.siappid=t2.appid where filename is not null group by
```

```
severity, hostname, user_src, sensitivity, service, action_type having sum(filesize)>0)) t group
by severity, hostname, user_src, sensitivity, service, action_type order by bytes desc
```

Dataset Name	Description	Log Category
dlp-Top-Users-by-DLP-Events	Top Users by DLP Events	dlp

```
select
  user_src,
  sum(sessions) as sessions
from
  (
    (
      select
        user_src,
        sum(sessions) as sessions
      from
        ###(select $flex_timestamp as timestamp, app, siappid, filename, profile, service,
coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry,
action, sum(case when cloudaction='upload' then coalesce(filesize, 0) else 0 end) as upload_size,
sum(case when cloudaction='download' or (cloudaction='others' and app like '%Video%') then
coalesce(filesize, 0) else 0 end) as download_size, sum(filesize) as filesize, count(*) as
sessions from $log-app-ctrl where $filter and siappid is not null and action in ('pass', 'block',
'reset') group by timestamp, app, siappid, filename, profile, service, action, user_src,
dstcountry order by sessions desc)### t1 inner join shadowit_application t2 on t1.siappid=t2.appid
where filename is not null group by user_src) union all (select user_src, sum(sessions) as
sessions from ###(select $flex_timestamp as timestamp, saasname, srcip, dstip, coalesce(nullifna
(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry, service, sum((CASE WHEN
(logflag&2>0) THEN 1 ELSE 0 END)) as session_block, count(*) as sessions, sum(case when
accessctrl='upload' THEN coalesce(sentbyte, 0) ELSE 0 END) AS upload_size, sum(case when
accessctrl='download' THEN coalesce(rcvdbyte, 0) ELSE 0 END) AS download_size from $log-traffic
where $filter and (logflag&1>0) and saasname is not null group by timestamp, saasname, srcip,
dstip, user_src, dstcountry, service order by sessions desc)### t group by user_src) union all
(select user_src, sum(sessions) as sessions from ###(select $flex_timestamp as timestamp,
hostname, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
dstcountry, (case when action = 'block' then 'Block' else 'Allow' end) as action_type, severity,
coalesce(sensitivity, 'Unclassified') as sen, service, profile, (case when direction='incoming'
then 'Download' else 'Upload' end) as traffic_direction, filename, sum(filesize) as filesize,
count(*) as sessions from $log-dlp where $filter and hostname is not null and action in ('log-
only', 'exempt', 'block') group by timestamp, hostname, user_src, dstcountry, action_type,
severity, sen, service, profile, traffic_direction, filename order by sessions desc)### t group by
user_src)) t group by user_src order by sessions desc
```

Dataset Name	Description	Log Category
dlp-Top-Destination-Countries-by-DLP-Events-Sankey	Top Destination Countries by DLP Events (Sankey)	dlp

```
select
  user_src,
  dstcountry,
  action_type,
  sum(sessions) as sessions
from
  (
    (
      select
        user_src,
        dstcountry,
        action_type,
        sum(sessions) as sessions
      from
        ###(select $flex_timestamp as timestamp, hostname, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry, (case when action = 'block' then 'Block'
else 'Allow' end) as action_type, severity, coalesce(sensitivity, 'Unclassified') as sen, service,
profile, (case when direction='incoming' then 'Download' else 'Upload' end) as traffic_direction,
filename, sum(filesize) as filesize, count(*) as sessions from $log-dlp where $filter and hostname
is not null and action in ('log-only', 'exempt', 'block') group by timestamp, hostname, user_src,
dstcountry, action_type, severity, sen, service, profile, traffic_direction, filename order by
sessions desc)### t where dstcountry is not null group by user_src, dstcountry, action_type) union
all (select user_src, dstcountry, (case when action='pass' then 'Allow' else 'Block' end) as
action_type, sum(sessions) as sessions from ###(select $flex_timestamp as timestamp, app, siappid,
filename, profile, service, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as
user_src, dstcountry, action, sum(case when cloudaction='upload' then coalesce(filesize, 0) else 0
end) as upload_size, sum(case when cloudaction='download' or (cloudaction='others' and app like
'%Video%') then coalesce(filesize, 0) else 0 end) as download_size, sum(filesize) as filesize,
count(*) as sessions from $log-app-ctrl where $filter and siappid is not null and action in
('pass', 'block', 'reset') group by timestamp, app, siappid, filename, profile, service, action,
user_src, dstcountry order by sessions desc)### t1 inner join shadowit_application t2 on
t1.siappid=t2.appid where dstcountry is not null and filename is not null group by user_src,
dstcountry, action_type) union all (select user_src, dstcountry, unnest(action) as action_type,
unnest(session) as sessions from (select user_src, dstcountry, array['Block', 'Allow'] as action,
array[sum(session_block), (sum(sessions)-sum(session_block))] as session, sum(sessions) as
sessions from ###(select $flex_timestamp as timestamp, saasname, srcip, dstip, coalesce(nullifna
(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry, service, sum((CASE WHEN
(logflag&2>0) THEN 1 ELSE 0 END)) as session_block, count(*) as sessions, sum(case when
accessctrl='upload' THEN coalesce(sentbyte, 0) ELSE 0 END) AS upload_size, sum(case when
accessctrl='download' THEN coalesce(rcvdbyte, 0) ELSE 0 END) AS download_size from $log-traffic
where $filter and (logflag&1>0) and saasname is not null group by timestamp, saasname, srcip,
dstip, user_src, dstcountry, service order by sessions desc)### t where dstcountry is not null
group by user_src, dstcountry, action) t)) t where dstcountry <> 'Reserved' group by user_src,
dstcountry, action_type order by sessions desc
```

Dataset Name	Description	Log Category
dlp-Top-Protocols-by-DLP-Events	Top Protocols by DLP Events	dlp

```
select
  service,
```

```
  sum(sessions) as sessions
from
  (
    (
      select
        service,
        sum(sessions) as sessions
      from
        ###(select $flex_timestamp as timestamp, app, siappid, filename, profile, service,
coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry,
action, sum(case when cloudaction='upload' then coalesce(filesize, 0) else 0 end) as upload_size,
sum(case when cloudaction='download' or (cloudaction='others' and app like '%Video%') then
coalesce(filesize, 0) else 0 end) as download_size, sum(filesize) as filesize, count(*) as
sessions from $log-app-ctrl where $filter and siappid is not null and action in ('pass', 'block',
'reset') group by timestamp, app, siappid, filename, profile, service, action, user_src,
dstcountry order by sessions desc)### t1 inner join shadowit_application t2 on t1.siappid=t2.appid
where filename is not null group by service) union all (select service, sum(sessions) as sessions
from ###(select $flex_timestamp as timestamp, saasname, srcip, dstip, coalesce(nullifna(`user`),
nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry, service, sum((CASE WHEN
(logflag&2>0) THEN 1 ELSE 0 END)) as session_block, count(*) as sessions, sum(case when
accessctrl='upload' THEN coalesce(sentbyte, 0) ELSE 0 END) AS upload_size, sum(case when
accessctrl='download' THEN coalesce(rcvdbyte, 0) ELSE 0 END) AS download_size from $log-traffic
where $filter and (logflag&1>0) and saasname is not null group by timestamp, saasname, srcip,
dstip, user_src, dstcountry, service order by sessions desc)### t group by service) union all
(select service, sum(sessions) as sessions from ###(select $flex_timestamp as timestamp, hostname,
coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry, (case
when action = 'block' then 'Block' else 'Allow' end) as action_type, severity, coalesce
(sensitivity, 'Unclassified') as sen, service, profile, (case when direction='incoming' then
'Download' else 'Upload' end) as traffic_direction, filename, sum(filesize) as filesize, count(*)
as sessions from $log-dlp where $filter and hostname is not null and action in ('log-only',
'exempt', 'block') group by timestamp, hostname, user_src, dstcountry, action_type, severity, sen,
service, profile, traffic_direction, filename order by sessions desc)### t group by service)) t
group by service order by sessions desc
```

Dataset Name	Description	Log Category
dlp-Top-Applications-by-DLP-Events	Top Applications by DLP Events	dlp

```
select
  app,
  sum(upload_bytes) as upload_bytes,
  sum(download_bytes) as download_bytes,
  sum(total_users) as total_users,
  riskscore,
  sum(session_block) as session_block,
  sum(session_pass) as session_pass,
  sum(sessions) as sessions
from
  (
    (
```

```
      select
        app,
        sum(upload_size) as upload_bytes,
        sum(download_size) as download_bytes,
        count(distinct user_src) as total_users,
        sum(
          case when action =& #039;pass' then sessions else 0 end) as session_pass, sum(case when
action!='pass' then sessions else 0 end) as session_block, sum(sessions) as sessions from ###
(select $flex_timestamp as timestamp, app, siappid, filename, profile, service, coalesce(nullifna
(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry, action, sum(case when
cloudaction='upload' then coalesce(filesize, 0) else 0 end) as upload_size, sum(case when
cloudaction='download' or (cloudaction='others' and app like '%Video%') then coalesce(filesize, 0)
else 0 end) as download_size, sum(filesize) as filesize, count(*) as sessions from $log-app-ctrl
where $filter and siappid is not null and action in ('pass', 'block', 'reset') group by timestamp,
app, siappid, filename, profile, service, action, user_src, dstcountry order by sessions desc)###
t1 inner join shadowit_application t2 on t1.siappid=t2.appid where filename is not null group by
app) union all (select saasname as app, sum(upload_size) as upload_bytes, sum(download_size) as
download_bytes, count(distinct user_src) as total_users, sum(session_block) as session_block, (sum
(sessions)-sum(session_block)) as session_pass, sum(sessions) as sessions from ###(select $flex_
timestamp as timestamp, saasname, srcip, dstip, coalesce(nullifna(`user`), nullifna(`unauthuser`),
ipstr(`srcip`)) as user_src, dstcountry, service, sum((CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END))
as session_block, count(*) as sessions, sum(case when accessctrl='upload' THEN coalesce(sentbyte,
0) ELSE 0 END) AS upload_size, sum(case when accessctrl='download' THEN coalesce(rcvdbyte, 0) ELSE
0 END) AS download_size from $log-traffic where $filter and (logflag&1>0) and saasname is not null
group by timestamp, saasname, srcip, dstip, user_src, dstcountry, service order by sessions
desc)### t group by app)) t1 inner join shadowit_application t2 on lower(t1.app)=lower(t2.appname)
group by app, riskscore order by sessions desc
```

Dataset Name	Description	Log Category
dlp-Top-Policies-by-DLP-Events-Sankey	DLP Policy Hits to Protocols (Sankey)	dlp

```
select
  profile,
  service,
  action_type,
  sum(sessions) as sessions
from
  ###(select $flex_timestamp as timestamp, hostname, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry, (case when action = 'block' then 'Block'
else 'Allow' end) as action_type, severity, coalesce(sensitivity, 'Unclassified') as sen, service,
profile, (case when direction='incoming' then 'Download' else 'Upload' end) as traffic_direction,
filename, sum(filesize) as filesize, count(*) as sessions from $log-dlp where $filter and hostname
is not null and action in ('log-only', 'exempt', 'block') group by timestamp, hostname, user_src,
dstcountry, action_type, severity, sen, service, profile, traffic_direction, filename order by
sessions desc)### t where profile is not null group by profile, service, action_type order by
sessions desc
```

Dataset Name	Description	Log Category
dlp-Top-Sensitive-Files-by-Severity	Top Sensitive Files by Severity	dlp

```
select
  filename,
  severity,
  hostname,
  sum(session_block) as session_block,
  sum(session_pass) as session_pass,
  sum(sessions) as sessions
from
  (
    (
      select
        filename,
        severity,
        hostname,
        sum(
          case when action_type = & #039;Allow' then sessions else 0 end) as session_pass, sum
(case when action_type = 'Block' then sessions else 0 end) as session_block, sum(sessions) as
sessions from ###(select $flex_timestamp as timestamp, hostname, coalesce(nullifna(`user`),
nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry, (case when action = 'block' then
'Block' else 'Allow' end) as action_type, severity, coalesce(sensitivity, 'Unclassified') as sen,
service, profile, (case when direction='incoming' then 'Download' else 'Upload' end) as traffic_
direction, filename, sum(filesize) as filesize, count(*) as sessions from $log-dlp where $filter
and hostname is not null and action in ('log-only', 'exempt', 'block') group by timestamp,
hostname, user_src, dstcountry, action_type, severity, sen, service, profile, traffic_direction,
filename order by sessions desc)### t where filename is not null group by filename, severity,
hostname order by severity desc) union all (select filename, (case when riskscore between 1 and 15
then 'info' when riskscore between 16 and 30 then 'low' when riskscore between 31 and 50 then
'medium' when riskscore between 51 and 70 then 'high' when riskscore between 71 and 100 then
'citical' else 'info' end)::fgt_enum_severity as severity, app as hostname, sum(case when action =
'pass' then sessions else 0 end) as session_pass, sum(case when action != 'pass' then sessions
else 0 end) as session_block, sum(sessions) as sessions from ###(select $flex_timestamp as
timestamp, app, siappid, filename, profile, service, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, dstcountry, action, sum(case when
cloudaction='upload' then coalesce(filesize, 0) else 0 end) as upload_size, sum(case when
cloudaction='download' or (cloudaction='others' and app like '%Video%') then coalesce(filesize, 0)
else 0 end) as download_size, sum(filesize) as filesize, count(*) as sessions from $log-app-ctrl
where $filter and siappid is not null and action in ('pass', 'block', 'reset') group by timestamp,
app, siappid, filename, profile, service, action, user_src, dstcountry order by sessions desc)###
t1 inner join shadowit_application t2 on t1.siappid=t2.appid group by filename, severity, hostname
having sum(filesize)>0)) t where filename is not null group by filename, severity, hostname order
by severity desc
```

Macro Reference List

The following table lists the available predefined macros that can be used in a report layout to display the log data as text (XML format) dynamically.

Macro Name	Description	Dataset Used	Log Category
Application Category with Highest Session Count	Application category with the highest session count	App-Sessions-By-Category	Traffic
Application with Highest Bandwidth	Application with the highest bandwidth usage	Top-App-By-Bandwidth	Traffic
Application with Highest Session Count	Applications with the highest session count	Top-App-By-Sessions	Traffic
Attack with Highest Session Count	Attack with highest session count	Utm-Top-Attack-Source	Attack
Botnet with Highest Session Count	Botnet with the highest session count	Detected-Botnet	Traffic
Destination with Highest Bandwidth	Destination with the highest bandwidth usage	Top-Destinations-By-Bandwidth	Traffic
Destination with Highest Session Count	Destination with the highest session count	Top-Destinations-By-Sessions	Traffic
Highest Bandwidth Consumed (Application) Category	Highest bandwidth consumed by application category	App-Risk-App-Usage-By-Category	Traffic
Highest Bandwidth Consumed (Application)	Highest bandwidth consumed by application	Top-App-By-Bandwidth	Traffic
Highest Bandwidth Consumed (Destination)	Highest bandwidth consumed by destination	Top-Destinations-By-Bandwidth	Traffic
Highest Bandwidth Consumed (P2P Application)	Highest bandwidth consumed by P2P application	Top-P2P-App-By-Bandwidth	Traffic
Highest Bandwidth Consumed (Source)	Highest bandwidth consumed by source	Top-Users-By-Bandwidth	Traffic
Highest Bandwidth Consumed ()Web Category)	Highest bandwidth consumed by website category	Top-Web-Category-by-Bandwidth	Web Filter
Highest Bandwidth Consumed (Website)	Highest bandwidth consumed by website	Top-Web-Sites-by-Bandwidth	Web Filter
Highest Risk Application with Highest Bandwidth	Highest risk application with the highest bandwidth usage	High-Risk-Application-By-Bandwidth	Traffic

Macro Name	Description	Dataset Used	Log Category
Highest Risk Application with Highest Session Count	Highest risk application with the highest session count	High-Risk-Application-By-Sessions	Traffic
Highest Session Count by Application Category	Highest session count by application category	App-Sessions-By-Category	Traffic
Highest Session Count by Application	Highest session count by application	Top-App-By-Sessions	Traffic
Highest Session Count by Attack	Highest session count by attack	Utm-Top-Attack-Source	Attack
Highest Session Count by Botnet	Highest session count by botnet	Detected-Botnet	Traffic
Highest Session Count by Destination	Highest session count by destination	Top-Destinations-By-Sessions	Traffic
Highest Session Count by Highest Severity Attack	Highest session count by highest severity attack	Threat-Attacks-By-Severity	Attack
Highest Session Count by P2P Application	Highest session count by P2P application	Top-P2P-App-By-Sessions	Traffic
Highest Session Count by Source	Highest session count by source	Top-User-Source-By-Sessions	Traffic
Highest Session Count by Virus	Highest session count by virus	Utm-Top-Virus	Traffic
Highest Session Count by Web Category	Highest session count by website category	Top-Web-Category-by-Sessions	Web Filter
Highest Session Count by Website	Highest session count by website	Top-Web-Sites-by-Sessions	Web Filter
Highest Severity Attack with Highest Session Count	Highest severity attack with the highest session count	Threat-Attacks-By-Severity	Attack
P2P Application with Highest Bandwidth	P2P applications with the highest bandwidth usage	Top-P2P-App-By-Bandwidth	Traffic
P2P Application with Highest Session Count	P2P applications with the highest session count	Top-P2P-App-By-Sessions	Traffic
Source with Highest Bandwidth	Source with the highest bandwidth usage	Top-Users-By-Bandwidth	Traffic
Source with Highest Session Count	Source with the highest session count	Top-User-Source-By-Sessions	Traffic
Total Number of Attacks	Total number of attacks detected	Total-Attack-Source	Attack

Macro Name	Description	Dataset Used	Log Category
Total Number of Botnet Events	Total number of botnet events	Total-Number-of-Botnet-Events	Traffic
Total Number of Viruses	Total number of viruses detected	Total-Number-of-Viruses	Traffic
User Details	User details of traffic	Traffic-User-Detail	Traffic
Virus with Highest Session Count	Virus with the highest session count	Utm-Top-Virus	Traffic
Web Category with Highest Bandwidth	Web filtering category with the highest bandwidth usage	Top-Web-Category-by-Bandwidth	Web Filter
Web Category with Highest Session Count	Web filtering category with the highest session count	Top-Web-Category-by-Sessions	Web Filter
Website with Highest Bandwidth	Website with the highest bandwidth usage	Top-Web-Sites-by-Bandwidth	Web Filter
Website with Highest Session Count	Website with the highest session count	Top-Web-Sites-by-Sessions	Web Filter

www.fortinet.com

Copyright© 2026 Fortinet, Inc. All rights reserved. Fortinet®, FortiGate®, FortiCare® and FortiGuard®, and certain other marks are registered trademarks of Fortinet, Inc., and other Fortinet names herein may also be registered and/or common law trademarks of Fortinet. All other product or company names may be trademarks of their respective owners. Performance and other metrics contained herein were attained in internal lab tests under ideal conditions, and actual performance and other results may vary. Network variables, different network environments and other conditions may affect performance results. Nothing herein represents any binding commitment by Fortinet, and Fortinet disclaims all warranties, whether express or implied, except to the extent Fortinet enters a binding written contract, signed by Fortinet's Chief Legal Officer, with a purchaser that expressly warrants that the identified product will perform according to certain expressly-identified performance metrics and, in such event, only the specific performance metrics expressly identified in such binding written contract shall be binding on Fortinet. For absolute clarity, any such warranty will be limited to performance in the same ideal conditions as in Fortinet's internal lab tests. Fortinet disclaims in full any covenants, representations, and guarantees pursuant hereto, whether express or implied. Fortinet reserves the right to change, modify, transfer, or otherwise revise this publication without notice, and the most current version of the publication shall be applicable.