# FortiAnalyzer - Dataset Reference

Version 6.4.9

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

**FORTINET VIDEO GUIDE**
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

**NSE INSTITUTE**
https://training.fortinet.com

**FORTIGUARD CENTER**
https://www.fortiguard.com

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

**FEEDBACK**
Email: techdoc@fortinet.com

August 31, 2022
FortiAnalyzer 6.4.9 Dataset Reference
05-649-0803421-20220831

# TABLE OF CONTENTS

# 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 from ###base(/*tag:rpt_base_t_bndwdth_sess*/select $flex_
timestamp as timestamp, dvid, srcip, dstip, epid, euid, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, service, count(*) 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, user_src, service /*SkipSTART*/order by timestamp desc/*SkipEND*/)base###
base_query group by timestamp order by bandwidth 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(sessions) as sessions from ###base(/*tag:rpt_base_t_bndwdth_
sess*/select $flex_timestamp as timestamp, dvid, srcip, dstip, epid, euid, coalesce(nullifna
(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service, count(*) 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, user_src, service /*SkipSTART*/order by timestamp
desc/*SkipEND*/)base### base_query group by timestamp order by 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 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 dvid, srcip, dstip, epid, euid, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, 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 dvid, srcip,
dstip, epid, euid, user_src, appid, app, appcat, apprisk, hostname order by sessions
desc)base### t group by user_src order by sessions desc, bandwidth desc)### 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, 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 dvid, srcip, dstip, epid, euid, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, 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 dvid, srcip,
dstip, epid, euid, user_src, appid, app, appcat, apprisk, hostname order by sessions
desc)base### t group by appid, app, appcat, apprisk /*SkipSTART*/order by sessions desc,
bandwidth 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
  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
```

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

```
select
  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
  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 from $log where $last3day_period
$filter and logid_to_int(logid) = 26001 and dhcp_msg = 'Ack' group by devintf, mac)### t
group by devintf, mac; create temporary table rpt_tmptbl_2 as select devintf, mac from ###
(select concat(interface, '.', devid) as devintf, mac from $log where $filter and logid_to_
```

```
int(logid) = 26001 and dhcp_msg = 'Ack' group by devintf, mac)### 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, coalesce
(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service, count(*) 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, user_src, service /*SkipSTART*/order by timestamp
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 | webfilter |

```
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-
traffic 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
  coalesce(
    nullifna(`user`),
    nullifna(`unauthuser`),
    ipstr(`srcip`)
  ) as user_src,
  get_devtype(srcswversion, osname, devtype) as devtype_new,
  srcname,
  count(*) as requests
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
  and utmevent in (
    & #039;webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') 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
  coalesce(
    nullifna(`user`),
    nullifna(`unauthuser`),
    ipstr(`srcip`)
  ) as user_src,
  get_devtype(srcswversion, osname, devtype) as devtype_new,
  srcname,
  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
(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 | webfilter |

```
select
  user_src,
  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,
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')))) group by user_src having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0
/*SkipSTART*/order by bandwidth desc/*SkipEND*/)### t group by user_src order by bandwidth
desc
```

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

```
select
  coalesce(
    nullifna(`user`),
    nullifna(`unauthuser`),
    ipstr(`srcip`)
  ) as user_src,
  get_devtype(srcswversion, osname, devtype) as devtype_new,
  srcname,
  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') group
by user_src, devtype_new, srcname having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>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
  coalesce(
    nullifna(`user`),
    nullifna(`unauthuser`),
    ipstr(`srcip`)
  ) as user_src,
  count(*) as requests
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
  and service in (
    & #039;smtp', 'SMTP', '25/tcp', '587/tcp', 'smtps', 'SMTPS', '465/tcp') 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
  coalesce(
    nullifna(`user`),
    nullifna(`unauthuser`),
    ipstr(`srcip`)
  ) as user_src,
  count(*) as requests
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
  and service in (
    & #039;pop3', 'POP3', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/tcp',
'pop3s', 'POP3S', '995/tcp') 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', '25/tcp', '587/tcp', 'smtps', 'SMTPS', '465/tcp') 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', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/tcp',
'pop3s', 'POP3S', '995/tcp') 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 (eventtype is null or logver>=502000000) 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 (eventtype is null or logver>=502000000) 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, count(*) as totalnum
from $log where $filter and (eventtype is null or logver>=502000000) and nullifna(virus) is
not null group by user_src /*SkipSTART*/order by totalnum desc/*SkipEND*/)### t group by
user_src order by totalnum desc
```

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

```
select
  coalesce(
    nullifna(`user`),
    ipstr(`srcip`)
  ) as user_src,
  count(*) as totalnum
from
  $log
where
  $filter
group by
  user_src
order by
  totalnum desc
```

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

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

```
where
  $filter
  and dstip is not null
group by
  dstip
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,
      tunnelid,
      vpn_name,
      (
        case when min(s_time)= max(e_time) then max(max_traffic_in) else max(max_traffic_
in)- min(min_traffic_in) end
      ) as traffic_in,
      (
        case when min(s_time)= max(e_time) then max(max_traffic_out) else max(max_traffic_
out)- min(min_traffic_out) end
      ) as traffic_out,
      (
        case when min(s_time)= max(e_time) then max(max_traffic_in)+ max(max_traffic_out)
else max(max_traffic_in)- min(min_traffic_in)+ max(max_traffic_out)- min(min_traffic_out)
end
      ) as bandwidth
    from
      ###(select devid, vd, remip, vpn_trim(vpntunnel) as vpn_name, tunnelid, tunnelip, max
(coalesce(sentbyte, 0)) as max_traffic_out, max(coalesce(rcvdbyte, 0)) as max_traffic_in,
min(coalesce(sentbyte, 0)) as min_traffic_out, min(coalesce(rcvdbyte, 0)) as min_traffic_in,
min(coalesce(dtime, 0)) as s_time, max(coalesce(dtime, 0)) as e_time from $log where $filter
and subtype='vpn' and tunneltype like 'ipsec%' and nullifna(vpntunnel) is not null  and
action in ('tunnel-stats', 'tunnel-down') and tunnelid is not null group by devid, vd,
remip, vpn_name, tunnelid, tunnelip)### t where (tunnelip is null or tunnelip='0.0.0.0')
group by devid, vd, remip, vpn_name, tunnelid) tt group by vpn_name having sum(traffic_
in+traffic_out)>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,
      user_src,
      tunnelid,
      min(s_time) as s_time,
      max(e_time) as e_time,
      (
        case when min(s_time)= max(e_time) then max(max_traffic_in)+ max(max_traffic_out)
else max(max_traffic_in)- min(min_traffic_in)+ max(max_traffic_out)- min(min_traffic_out)
end
      ) as bandwidth,
      (
        case when min(s_time)= max(e_time) then max(max_traffic_in) else max(max_traffic_
in)- min(min_traffic_in) end
      ) as traffic_in,
      (
        case when min(s_time)= max(e_time) then max(max_traffic_out) else max(max_traffic_
out)- min(min_traffic_out) end
      ) 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, min
(coalesce(sentbyte, 0)) as min_traffic_out, min(coalesce(rcvdbyte, 0)) as min_traffic_in,
max(coalesce(sentbyte, 0)) as max_traffic_out, max(coalesce(rcvdbyte, 0)) as max_traffic_in
from $log where $filter and subtype='vpn' and tunneltype like 'ssl%' and action in ('tunnel-
stats', 'tunnel-down', 'tunnel-up') and coalesce(nullifna(`user`), ipstr(`remip`)) is not
null and tunnelid is not null group by devid, vd, user_src, remip, tunnelid, tunneltype)###
t where tunneltype='ssl-tunnel' group by devid, vd, user_src, remip, tunnelid) tt where
bandwidth>0 group by user_src, remote_ip 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,
      tunnelid,
      remip,
      vpn_name,
      (
        case when min(s_time)= max(e_time) then max(max_traffic_in) else max(max_traffic_
in)- min(min_traffic_in) end
      ) as traffic_in,
      (
        case when min(s_time)= max(e_time) then max(max_traffic_out) else max(max_traffic_
out)- min(min_traffic_out) end
      ) as traffic_out,
      (
        case when min(s_time)= max(e_time) then max(max_traffic_in)+ max(max_traffic_out)
else max(max_traffic_in)- min(min_traffic_in)+ max(max_traffic_out)- min(min_traffic_out)
end
      ) as bandwidth
    from
      ###(select devid, vd, remip, vpn_trim(vpntunnel) as vpn_name, tunnelid, tunnelip, max
(coalesce(sentbyte, 0)) as max_traffic_out, max(coalesce(rcvdbyte, 0)) as max_traffic_in,
min(coalesce(sentbyte, 0)) as min_traffic_out, min(coalesce(rcvdbyte, 0)) as min_traffic_in,
min(coalesce(dtime, 0)) as s_time, max(coalesce(dtime, 0)) as e_time from $log where $filter
and subtype='vpn' and tunneltype like 'ipsec%' and nullifna(vpntunnel) is not null and
action in ('tunnel-stats', 'tunnel-down') and tunnelid is not null group by devid, vd,
remip, vpn_name, tunnelid, tunnelip)### t where not (tunnelip is null or tunnelip='0.0.0.0')
group by devid, vd, remip, vpn_name, tunnelid) tt group by vpn_name having sum(traffic_
out+traffic_in)>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, (case when min(s_time)=max(e_time)
then max(max_traffic_in)+max(max_traffic_out) else max(max_traffic_in)-min(min_traffic_
in)+max(max_traffic_out)-min(min_traffic_out) end) as bandwidth, (case when min(s_time)=max
(e_time) then max(max_traffic_in) else max(max_traffic_in)-min(min_traffic_in) end) as
traffic_in, (case when min(s_time)=max(e_time) then max(max_traffic_out) else max(max_
traffic_out)-min(min_traffic_out) end) as traffic_out from ###(select devid, vd, remip,
nullifna(`xauthuser`) as xauthuser_agg, nullifna(`user`) as user_agg, tunnelid, 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, min(coalesce(sentbyte, 0)) as min_
traffic_out, min(coalesce(rcvdbyte, 0)) as min_traffic_in, max(coalesce(sentbyte, 0)) as
max_traffic_out, max(coalesce(rcvdbyte, 0)) as max_traffic_in from $log where $filter and
subtype='vpn' and tunneltype like 'ipsec%' and not (tunnelip is null or tunnelip='0.0.0.0')
and  action in ('tunnel-stats', 'tunnel-down', 'tunnel-up') and tunnelid is not null and
tunnelid!=0 group by devid, vd, remip, xauthuser_agg, user_agg, tunnelid order by
tunnelid)### 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, (case when
min(s_time)=max(e_time) then max(max_traffic_in)+max(max_traffic_out) else  max(max_traffic_
in)-min(min_traffic_in)+max(max_traffic_out)-min(min_traffic_out) end) as bandwidth, (case
when min(s_time)=max(e_time) then max(max_traffic_in) else max(max_traffic_in)-min(min_
traffic_in) end) as traffic_in, (case when min(s_time)=max(e_time) then max(max_traffic_out)
else max(max_traffic_out)-min(min_traffic_out) end) as traffic_out from ###(select devid,
vd, remip, nullifna(`xauthuser`) as xauthuser_agg, nullifna(`user`) as user_agg, tunnelid,
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, min(coalesce
(sentbyte, 0)) as min_traffic_out, min(coalesce(rcvdbyte, 0)) as min_traffic_in, max
(coalesce(sentbyte, 0)) as max_traffic_out, max(coalesce(rcvdbyte, 0)) as max_traffic_in
```

```
from $log where $filter and subtype='vpn' and tunneltype like 'ipsec%' and not (tunnelip is
null or tunnelip='0.0.0.0') and  action in ('tunnel-stats', 'tunnel-down', 'tunnel-up') and
tunnelid is not null and tunnelid!=0 group by devid, vd, remip, xauthuser_agg, user_agg,
tunnelid order by tunnelid)### 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,
      user_src,
      remip,
      tunnelid,
      min(s_time) as s_time,
      max(e_time) as e_time,
      (
        case when min(s_time)= max(e_time) then max(max_traffic_in)+ max(max_traffic_out)
else max(max_traffic_in)- min(min_traffic_in)+ max(max_traffic_out)- min(min_traffic_out)
end
      ) as bandwidth,
      (
        case when min(s_time)= max(e_time) then max(max_traffic_in) else max(max_traffic_
in)- min(min_traffic_in) end
      ) as traffic_in,
      (
        case when min(s_time)= max(e_time) then max(max_traffic_out) else max(max_traffic_
out)- min(min_traffic_out) end
      ) 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, min
(coalesce(sentbyte, 0)) as min_traffic_out, min(coalesce(rcvdbyte, 0)) as min_traffic_in,
max(coalesce(sentbyte, 0)) as max_traffic_out, max(coalesce(rcvdbyte, 0)) as max_traffic_in
from $log where $filter and subtype='vpn' and tunneltype like 'ssl%' and action in ('tunnel-
stats', 'tunnel-down', 'tunnel-up') and coalesce(nullifna(`user`), ipstr(`remip`)) is not
null and tunnelid is not null group by devid, vd, user_src, remip, tunnelid, tunneltype)###
t group by devid, vd, user_src, remip, tunnelid) tt where bandwidth>0 group by user_src,
remote_ip 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,
  (
    max(e_time)- min(s_time)
  ) as duration
from
  (
    select
      devid,
      vd,
      user_src,
      remip,
      tunnelid,
      min(s_time) as s_time,
      max(e_time) as e_time
    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, min
(coalesce(sentbyte, 0)) as min_traffic_out, min(coalesce(rcvdbyte, 0)) as min_traffic_in,
max(coalesce(sentbyte, 0)) as max_traffic_out, max(coalesce(rcvdbyte, 0)) as max_traffic_in
from $log where $filter and subtype='vpn' and tunneltype like 'ssl%' and action in ('tunnel-
stats', 'tunnel-down', 'tunnel-up') and coalesce(nullifna(`user`), ipstr(`remip`)) is not
null and tunnelid is not null group by devid, vd, user_src, remip, tunnelid, tunneltype)###
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,
      (
        case when min(s_time)= max(e_time) then max(max_traffic_in) else max(max_traffic_
in)- min(min_traffic_in) end
      ) as traffic_in,
      (
        case when min(s_time)= max(e_time) then max(max_traffic_out) else max(max_traffic_
out)- min(min_traffic_out) end
      ) as traffic_out,
      (
        case when min(s_time)= max(e_time) then max(max_traffic_in)+ max(max_traffic_out)
else max(max_traffic_in)- min(min_traffic_in)+ max(max_traffic_out)- min(min_traffic_out)
end
      ) 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, min
(coalesce(sentbyte, 0)) as min_traffic_out, min(coalesce(rcvdbyte, 0)) as min_traffic_in,
max(coalesce(sentbyte, 0)) as max_traffic_out, max(coalesce(rcvdbyte, 0)) as max_traffic_in
from $log where $filter and subtype='vpn' and tunneltype like 'ssl%' and action in ('tunnel-
stats', 'tunnel-down', 'tunnel-up') and coalesce(nullifna(`user`), ipstr(`remip`)) is not
null and tunnelid is not null group by devid, vd, user_src, remip, tunnelid, tunneltype)###
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-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, (case when
min(s_time)=max(e_time) then max(max_traffic_in)+max(max_traffic_out) else  max(max_traffic_
in)-min(min_traffic_in)+max(max_traffic_out)-min(min_traffic_out) end) as bandwidth, (case
when min(s_time)=max(e_time) then max(max_traffic_in) else max(max_traffic_in)-min(min_
traffic_in) end) as traffic_in, (case when min(s_time)=max(e_time) then max(max_traffic_out)
else max(max_traffic_out)-min(min_traffic_out) end) 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, min(coalesce
(sentbyte, 0)) as min_traffic_out, min(coalesce(rcvdbyte, 0)) as min_traffic_in, max
(coalesce(sentbyte, 0)) as max_traffic_out, max(coalesce(rcvdbyte, 0)) as max_traffic_in,
sum((case when action='tunnel-up' then 1 else 0 end)) as tunnelup 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 and tunnelid!=0 group
by xauthuser_agg, user_agg, devid, vd, remip, t_type, tunnelid, tunnelip)### 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-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(tunnelup) > 0 and 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='ipsec' or left
(tunneltype, 3)='ssl') and action in ('ssl-login-fail', 'ipsec-login-fail') and coalesce
(nullifna(`xauthuser`), nullifna(`user`)) is not null group by f_user, tunneltype)### t
group by f_user, tunneltype order by total_num desc
```

| 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,
  sum(total_num) 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 max(max_duration) else max(max_duration)-min(min_duration) end) as
duration, (case when min(s_time)=max(e_time) then max(max_traffic_in)+max(max_traffic_out)
else  max(max_traffic_in)-min(min_traffic_in)+max(max_traffic_out)-min(min_traffic_out) end)
as bandwidth, (case when min(s_time)=max(e_time) then max(max_traffic_in) else max(max_
traffic_in)-min(min_traffic_in) end) as traffic_in, (case when min(s_time)=max(e_time) then
max(max_traffic_out) else max(max_traffic_out)-min(min_traffic_out) end) as traffic_out, sum
(tunnelup) 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, min(coalesce(sentbyte, 0)) as min_traffic_out, min
(coalesce(rcvdbyte, 0)) as min_traffic_in, max(coalesce(sentbyte, 0)) as max_traffic_out,
max(coalesce(rcvdbyte, 0)) as max_traffic_in, sum((case when action='tunnel-up' then 1 else
0 end)) as tunnelup 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 and tunnelid!=0 group by xauthuser_agg, user_agg, devid, vd, remip, t_
type, tunnelid, tunnelip)### t group by t_type, devid, vd, remip, tunnelid having max
(tunnelup) > 0) tt where bandwidth>0 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(ssl_traffic_bandwidth) as ssl_bandwidth,
  sum(ipsec_traffic_bandwidth) as ipsec_bandwidth
from
  (
    select
      $flex_timescale(timestamp) as hodex,
      devid,
      vd,
      remip,
      tunnelid,
      (
        case when t_type like & #039;ssl%' then (case when min(s_time)=max(e_time) then max
(max_traffic_in)+max(max_traffic_out) else max(max_traffic_in)-min(min_traffic_in)+max(max_
traffic_out)-min(min_traffic_out) end)  else 0 end) as ssl_traffic_bandwidth, (case when t_
type like 'ipsec%' then (case when min(s_time)=max(e_time) then max(max_traffic_in)+max(max_
traffic_out) else max(max_traffic_in)-min(min_traffic_in)+max(max_traffic_out)-min(min_
traffic_out) end)  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, max(coalesce(sentbyte, 0)) as max_
traffic_out, max(coalesce(rcvdbyte, 0)) as max_traffic_in, min(coalesce(sentbyte, 0)) as
min_traffic_out, min(coalesce(rcvdbyte, 0)) as min_traffic_in, min(coalesce(dtime, 0)) as s_
time, max(coalesce(dtime, 0)) as e_time 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 and tunnelid!=0 group by timestamp, devid,
vd, remip, t_type, tunnelid, action /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t
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, 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 tunneltype in ('ssl-tunnel', 'ssl')
and coalesce(nullifna(`user`), ipstr(`remip`)) is not null and tunnelid is not null group by
tunnelid, user_src, remip, devid, vd /*SkipSTART*/order by tunnelid/*SkipEND*/)### t 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, 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 tunneltype='ssl-web' and coalesce
(nullifna(`user`), ipstr(`remip`)) is not null and tunnelid is not null group by tunnelid,
user_src, remip, devid, vd /*SkipSTART*/order by tunnelid/*SkipEND*/)### t 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', '25/tcp',
```

```
'587/tcp', 'smtps', 'SMTPS', '465/tcp') 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', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/tcp', 'pop3s',
'POP3S', '995/tcp') 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', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/tcp', 'pop3s',
'POP3S', '995/tcp') 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', '25/tcp',
'587/tcp', 'smtps', 'SMTPS', '465/tcp') 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 (eventtype is null or logver>=502000000) 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 (eventtype is null or logver>=502000000) 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 dvid, srcip, dstip,
epid, euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
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 dvid, srcip, dstip, epid, euid, user_src, appid, app, appcat, apprisk,
hostname order by 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 dvid, srcip, dstip,
epid, euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
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 dvid, srcip, dstip, epid, euid, user_src, appid, app, appcat, apprisk,
hostname order by 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, 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 dvid, srcip, dstip, epid, euid, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, 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 dvid, srcip,
```

```
dstip, epid, euid, user_src, appid, app, appcat, apprisk, hostname order by sessions
desc)base### t group by appid, app, appcat, apprisk /*SkipSTART*/order by sessions desc,
bandwidth 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, 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 dvid, srcip, dstip, epid, euid, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, 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 dvid, srcip,
dstip, epid, euid, user_src, appid, app, appcat, apprisk, hostname order by 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
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 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 dvid, srcip, dstip, epid, euid, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, 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 dvid, srcip,
dstip, epid, euid, user_src, appid, app, appcat, apprisk, hostname order by sessions
desc)base### t group by user_src order by sessions desc, bandwidth desc)### 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 from $log where $filter and (logflag&(1|32)>0)
group by timestamp, user_src order by timestamp 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 domain,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out,
  sum(bandwidth) as bandwidth,
  sum(sessions) as sessions
from
  ###(select 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 dvid, srcip, dstip, epid, euid, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, 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 dvid, srcip,
dstip, epid, euid, user_src, appid, app, appcat, apprisk, hostname order by sessions
desc)base### t group by hostname, dstip order by sessions desc, bandwidth desc)### t group
by domain 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 appid, app, appcat, apprisk, 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 dvid, srcip, dstip, epid, euid, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, 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 dvid, srcip,
dstip, epid, euid, user_src, appid, app, appcat, apprisk, hostname order by sessions
desc)base### t group by appid, app, appcat, apprisk /*SkipSTART*/order by sessions desc,
bandwidth desc/*SkipEND*/)### 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 user_src, sum(sessions) as count from ###base(/*tag:rpt_base_t_top_
app*/select dvid, srcip, dstip, epid, euid, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, 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 dvid, srcip,
dstip, epid, euid, user_src, appid, app, appcat, apprisk, hostname order by sessions
desc)base### t group by user_src order by count desc)### t) as t2; update rpt_tmptbl_1 set
total_app=t3.totalnum from (select format_numeric_no_decimal(count(distinct(app_grp))) as
totalnum from ###(select app_group_name(app) as app_grp, sum(sessions) as count from ###base
(/*tag:rpt_base_t_top_app*/select dvid, srcip, dstip, epid, euid, coalesce(nullifna(`user`),
nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, 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 dvid, srcip,
dstip, epid, euid, user_src, appid, app, appcat, apprisk, hostname order by sessions
desc)base### t group by app_grp order by count desc)### 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 dstip, sum(sessions) as count from ###base(/*tag:rpt_base_t_top_
app*/select dvid, srcip, dstip, epid, euid, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, 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 dvid, srcip,
dstip, epid, euid, user_src, appid, app, appcat, apprisk, hostname order by sessions
desc)base### t where dstip is not null group by dstip order by count desc)### 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 |
|---|---|---|
| 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
  srcip,
```

```
  hostname,
  sum(totalnum) as totalnum
from
  ###(select srcip , ipstr(`dstip`) as hostname, count(*) as totalnum from $log where
$filter and (eventtype is null or logver>=502000000) and nullifna(virus) is not null group
by srcip, hostname /*SkipSTART*/order by totalnum desc/*SkipEND*/)### t group by srcip,
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
(eventtype is null or logver>=502000000) 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,
  count(*) as totalnum
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
  and virus like & #039;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, count(*) as totalnum from $log where $filter and
virus like 'Riskware%' group by timestamp /*SkipSTART*/order by timestamp
desc/*SkipEND*/)### t 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,
  count(*) as totalnum
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
  and virus like & #039;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, count(*) as totalnum from $log where $filter and
virus like 'Adware%' group by timestamp /*SkipSTART*/order by timestamp desc/*SkipEND*/)###
t 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 dstip 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)### 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 srcip 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)### 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 t1.severity =& #039;critical' then 'Critical' when t1.severity='high' then
'High' when t1.severity='medium' then 'Medium' when t1.severity='low' then 'Low' when
t1.severity='info' then 'Info' end) as severity_name, count(*) as totalnum, vuln_type, (case
when t1.severity='critical' then 0 when t1.severity='high' then 1 when t1.severity='medium'
then 2 when t1.severity='low' then 3 when t1.severity='info' then 4 else 5 end) as severity_
number 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 and action not in
('detected', 'pass_session') group by attack, attackid, t1.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 t1.severity =& #039;critical' then 'Critical' when t1.severity='high' then
'High' when t1.severity='medium' then 'Medium' when t1.severity='low' then 'Low' when
t1.severity='info' then 'Info' end) as severity_name, count(*) as totalnum, vuln_type, (case
when t1.severity='critical' then 0 when t1.severity='high' then 1 when t1.severity='medium'
then 2 when t1.severity='low' then 3 when t1.severity='info' then 4 else 5 end) as severity_
number 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 and action in ('detected',
'pass_session') group by attack, attackid, t1.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, count(*) as subtotal from $log where $filter and
apstatus is not null and apstatus!=0 and bssid is not null and onwire='no' 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 order by subtotal desc)### t 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, count(*) as subtotal from $log where $filter and
apstatus is not null and apstatus!=0 and bssid is not null and onwire='no' 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 order by subtotal desc)### t 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, count(*) as subtotal from $log where $filter and
apstatus is not null and apstatus!=0 and bssid is not null and onwire='yes' 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 order by subtotal desc)### t 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, count(*) as subtotal from $log where $filter and
apstatus is not null and apstatus!=0 and bssid is not null and onwire='yes' 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 order by subtotal desc)### t 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(dtime)) as first_seen, from_dtime(max
(dtime)) as last_seen, detectionmethod, itime, onwire as on_wire from $log where $filter and
apstatus is not null and bssid is not null and onwire='no' and logid_to_int(logid) in
(43521, 43563, 43564, 43565, 43566, 43569, 43570, 43571) group by ap_full_status, devid, vd,
ssid, bssid, manuf, rssi, channel, radioband, detectionmethod, itime, onwire,apstatus
```

| 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(dtime)) as first_seen, from_dtime(max
(dtime)) as last_seen, detectionmethod, itime, onwire as on_wire from $log where $filter and
apstatus is not null and bssid is not null and onwire='yes' and logid_to_int(logid) in
(43521, 43563, 43564, 43565, 43566, 43569, 43570, 43571) group by ap_full_status, devid, vd,
ssid, bssid, manuf, rssi, channel, radioband, detectionmethod, itime, onwire,apstatus
```

| 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 ip, lower(mac) as lmac, sn, ssid, channel, radioband, min(dtime) as first, max
(dtime) as last from $log-event where $filter and ip is not null and mac is not null and sn
is not null and ssid is not null group by ip, lmac, sn, ssid, channel, radioband order by
ip)### 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, 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 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, 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 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, 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 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, 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 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, (case when min(s_time)=max(e_time)
then max(max_traffic_in)+max(max_traffic_out) else max(max_traffic_in)-min(min_traffic_
in)+max(max_traffic_out)-min(min_traffic_out) end) as bandwidth, (case when min(s_time)=max
(e_time) then max(max_traffic_in) else max(max_traffic_in)-min(min_traffic_in) end) as
traffic_in, (case when min(s_time)=max(e_time) then max(max_traffic_out) else max(max_
traffic_out)-min(min_traffic_out) end) as traffic_out from ###(select devid, vd, remip,
nullifna(`xauthuser`) as xauthuser_agg, nullifna(`user`) as user_agg, tunnelid, 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, min(coalesce(sentbyte, 0)) as min_
traffic_out, min(coalesce(rcvdbyte, 0)) as min_traffic_in, max(coalesce(sentbyte, 0)) as
max_traffic_out, max(coalesce(rcvdbyte, 0)) as max_traffic_in from $log where $filter and
subtype='vpn' and tunneltype like 'ipsec%' and not (tunnelip is null or tunnelip='0.0.0.0')
and  action in ('tunnel-stats', 'tunnel-down', 'tunnel-up') and tunnelid is not null and
tunnelid!=0 group by devid, vd, remip, xauthuser_agg, user_agg, tunnelid order by
tunnelid)### 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
      devid,
      vd,
      remip,
      tunnelid,
      (
        case when min(s_time)= max(e_time) then max(max_traffic_in) else max(max_traffic_
in)- min(min_traffic_in) end
      ) as traffic_in,
      (
        case when min(s_time)= max(e_time) then max(max_traffic_out) else max(max_traffic_
out)- min(min_traffic_out) end
      ) as traffic_out,
      (
        case when min(s_time)= max(e_time) then max(max_traffic_in)+ max(max_traffic_out)
else max(max_traffic_in)- min(min_traffic_in)+ max(max_traffic_out)- min(min_traffic_out)
end
      ) 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, max(coalesce(sentbyte, 0)) as max_
traffic_out, max(coalesce(rcvdbyte, 0)) as max_traffic_in, min(coalesce(sentbyte, 0)) as
min_traffic_out, min(coalesce(rcvdbyte, 0)) as min_traffic_in, min(coalesce(dtime, 0)) as s_
time, max(coalesce(dtime, 0)) as e_time 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 and tunnelid!=0 group by timestamp, devid,
vd, remip, t_type, tunnelid, action /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t
where t_type like 'ssl%' group by devid, vd, remip, tunnelid) tt group by remote_ip having
sum(traffic_in+traffic_out)>0 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 and (eventtype is null or logver>=502000000) 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
  user_src,
  sum(requests) as requests
from
  ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, count(*) as requests
from $log where $filter and (eventtype is null or logver>=502000000) and coalesce(nullifna
(`user`), ipstr(`srcip`)) is not null and action='blocked' group by user_src
/*SkipSTART*/order by requests desc/*SkipEND*/)### t group by user_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
  user_src,
  sum(requests) as requests
from
  ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, count(*) as requests
from $log where $filter and (eventtype is null or logver>=502000000) and coalesce(nullifna
(`user`), ipstr(`srcip`)) is not null and action!='blocked' group by user_src
/*SkipSTART*/order by requests desc/*SkipEND*/)### t group by user_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, count(*) as requests from $log where $filter and
(eventtype is null or logver>=502000000) and hostname is not null and catdesc is not null
and action='blocked' group by domain, catdesc /*SkipSTART*/order by requests
desc/*SkipEND*/)### t 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, count(*) as requests from $log where $filter and (eventtype is null or
logver>=502000000) and hostname is not null and catdesc is not null and action!='blocked'
group by domain, catdesc /*SkipSTART*/order by requests desc/*SkipEND*/)### t 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, count(*) as requests from $log-webfilter where $filter and (eventtype
is null or logver>=502000000) and catdesc is not null and action='blocked' group by catdesc
/*SkipSTART*/order by requests desc/*SkipEND*/)### t 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, count(*) as requests from $log-webfilter where $filter and (eventtype
is null or logver>=502000000) and catdesc is not null and action!='blocked' group by catdesc
/*SkipSTART*/order by requests desc/*SkipEND*/)### t 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
  user_src,
  ebtr_value(
    ebtr_agg_flat(browsetime),
    null,
    $timespan
  ) as browsetime
from
  ###(select user_src, ebtr_agg_flat(browsetime) as browsetime from (select coalesce
(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, ebtr_agg_flat
($browse_time) as browsetime from $log where $filter and (logflag&1>0) and $browse_time is
not null group by user_src) t group by user_src order by ebtr_value(ebtr_agg_flat
(browsetime), null, null) desc)### t group by user_src order by browsetime desc
```

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

```
select
  user_src,
  ebtr_value(
    ebtr_agg_flat(browsetime),
```

```
    null,
    $timespan
  ) as browsetime
from
  ###(select user_src, ebtr_agg_flat(browsetime) as browsetime from (select coalesce
(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, ebtr_agg_flat
($browse_time) as browsetime from $log where $filter and (logflag&1>0) and $browse_time is
not null group by user_src) t group by user_src order by ebtr_value(ebtr_agg_flat
(browsetime), null, null) desc)### t group by user_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 and (eventtype is null or logver>=502000000) 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(itime) 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, 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 dtime, catdesc, hostname,
utmaction order by dtime desc)### t group by dtime, catdesc, website, status order by dtime
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 and
(eventtype is null or logver>=502000000) 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 and (eventtype is null or logver>=502000000) 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(dtime) as timestamp,
  user_src,
  website,
  catdesc,
  cast(
    sum(dura)/ 60 as decimal(18, 2)
  ) as dura,
  sum(bandwidth) as bandwidth
from
  ###(select dtime, 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 dtime, user_
src, website, catdesc having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by
bandwidth desc)### t group by dtime, 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 and (eventtype is null or logver>=502000000) group by hostname, catdesc
order by sessions desc)### t 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)### 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)### 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)### 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 from
$log where $filter group by uid, profile)### 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 from $log
where $filter and threat is not null and utmaction='blocked' group by utmevent_s, threat)###
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 from $log where $filter
and nullifna(vulnseverity) is not null and nullifna(vulnname) is not null group by
vulnseverity, vulnname, devid)### 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 from $log where $filter and nullifna(vulncat) is not null and
nullifna(vulnname) is not null group by vulncat, vulnname)### 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 from $log where $filter and nullifna(os) is not null and nullifna
(vulnname) is not null group by os, vulnname)### 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 from $log where $filter and
nullifna(vulnseverity) is not null and nullifna(vulnname) is not null and nullifna(devid) is
not null group by vulnseverity, vulnname, devid)### t 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 from $log where $filter and
nullifna(vulnseverity) is not null and nullifna(vulnname) is not null and nullifna(devid) is
not null group by vulnseverity, vulnname, devid)### t 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 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)### 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 from $log where $filter
and vulnname is not null and hostname is not null group by hostname, vulnname, vulnseverity,
vulncat, vulnid)### 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
from $log where $filter and vulnname is not null and hostname is not null group by hostname,
vulnname, vulnseverity, vulnid)### 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, ('<a href=' || String_agg(vendor_link, ',') || '>Mitigation
Infomation</a>') as vendor_link from ###(select vulnid from $log where $filter group 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)### 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 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)### 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)### 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)### 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)### 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, count(*) as requests
from $log where $filter and direction='outbound' and threat is not null and
utmaction='passthrough' and lower(utmevent)='webfilter' group by category, website order by
requests desc)### t 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, count(*) as requests from $log where $filter
and direction='outbound' and threat is not null and utmaction='passthrough' and lower
(utmevent)='webfilter' group by category, website order by requests desc)### t where 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
  fct_webcat(threat) as category,
  remotename as website,
  count(*) as requests
from
  $log
where
  $filter
  and nullifna(threat) is not null
  and lower(utmevent)=& #039;webfilter' 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 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)### union all ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src,
hostname, count(*) as requests from $log-webfilter where $filter-exclude-var and (eventtype
is null or logver>=502000000) and hostname is not null group by user_src, hostname order by
requests desc)###) t where $filter-drilldown and 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 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)### union all ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src,
hostname, count(*) as requests from $log-webfilter where $filter-exclude-var and (eventtype
is null or logver>=502000000) and hostname is not null group by user_src, hostname order by
requests desc)###) t where $filter-drilldown and 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 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)### union all ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src,
hostname, count(*) as requests from $log-webfilter where $filter-exclude-var and (eventtype
is null or logver>=502000000) and hostname is not null group by user_src, hostname order by
requests desc)###) t where $filter-drilldown and 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 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)### union all ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src,
hostname, count(*) as requests from $log-webfilter where $filter-exclude-var and (eventtype
is null or logver>=502000000) and hostname is not null group by user_src, hostname order by
requests desc)###) t where $filter-drilldown and 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, 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', '25/tcp', '587/tcp', 'smtps', 'SMTPS', '465/tcp') and
utmevent in ('general-email-log', 'spamfilter') group by sender, recipient order by requests
desc)### union all ###(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', '25/tcp', '587/tcp', 'smtps', 'SMTPS',
'465/tcp') and eventtype is null group by `from`, `to` order by requests desc)###) t where
$filter-drilldown and 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, 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', '25/tcp', '587/tcp', 'smtps', 'SMTPS', '465/tcp') and
utmevent in ('general-email-log', 'spamfilter') group by sender, recipient order by requests
desc)### union all ###(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', '25/tcp', '587/tcp', 'smtps', 'SMTPS',
'465/tcp') and eventtype is null group by `from`, `to` order by requests desc)###) t where
$filter-drilldown and 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, 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', '25/tcp', '587/tcp', 'smtps', 'SMTPS', '465/tcp') and
utmevent in ('general-email-log', 'spamfilter') group by sender, recipient order by requests
desc)### union all ###(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', '25/tcp', '587/tcp', 'smtps', 'SMTPS',
'465/tcp') and eventtype is null group by `from`, `to` order by requests desc)###) t where
$filter-drilldown and 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, 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', '25/tcp', '587/tcp', 'smtps', 'SMTPS', '465/tcp') and
utmevent in ('general-email-log', 'spamfilter') group by sender, recipient order by requests
desc)### union all ###(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', '25/tcp', '587/tcp', 'smtps', 'SMTPS',
'465/tcp') and eventtype is null group by `from`, `to` order by requests desc)###) t where
$filter-drilldown and 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, 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', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS',
'993/tcp', 'pop3s', 'POP3S', '995/tcp') and utmevent in ('general-email-log', 'spamfilter')
group by recipient, sender order by requests desc)### union all ###(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', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/tcp', 'pop3s',
'POP3S', '995/tcp') and eventtype is null group by `to`, `from` order by requests desc)###)
t where $filter-drilldown and 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, 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', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS',
'993/tcp', 'pop3s', 'POP3S', '995/tcp') and utmevent in ('general-email-log', 'spamfilter')
group by recipient, sender order by requests desc)### union all ###(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', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/tcp', 'pop3s',
'POP3S', '995/tcp') and eventtype is null group by `to`, `from` order by requests desc)###)
t where $filter-drilldown and 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, 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', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS',
'993/tcp', 'pop3s', 'POP3S', '995/tcp') and utmevent in ('general-email-log', 'spamfilter')
group by recipient, sender order by requests desc)### union all ###(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', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/tcp', 'pop3s',
'POP3S', '995/tcp') and eventtype is null group by `to`, `from` order by requests desc)###)
t where $filter-drilldown and 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, 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', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS',
'993/tcp', 'pop3s', 'POP3S', '995/tcp') and utmevent in ('general-email-log', 'spamfilter')
group by recipient, sender order by requests desc)### union all ###(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', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/tcp', 'pop3s',
'POP3S', '995/tcp') and eventtype is null group by `to`, `from` order by requests desc)###)
t where $filter-drilldown and 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
  dstip,
  sum(totalnum) as totalnum
from
  ###(select srcip, dstip, count(*) as totalnum from $log where $filter-exclude-var group by
srcip, dstip order by totalnum desc)### t where $filter-drilldown and dstip is not null
group by dstip order by totalnum desc
```

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

```
select
  srcip,
  sum(totalnum) as totalnum
from
  ###(select srcip, dstip, count(*) as totalnum from $log where $filter-exclude-var group by
srcip, dstip order by totalnum desc)### t where $filter-drilldown and srcip is not null
group by srcip 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,
  srcip,
  dstip
from
  ###(select itime, attack, srcip, dstip 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 (eventtype is null or logver>=502000000) 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,
  dstip,
  hostname,
  recipient
from
  ###(select itime, virus, coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, dstip,
cast(' ' as char) as hostname, cast(' ' as char) as recipient from $log where $filter and
(eventtype is null or logver>=502000000) 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 user_src, hostname, action, count
(*) as requests from $log where $filter and hostname is not null group by user_src,
hostname, action order by requests desc)### t where $filter-drilldown and action='blocked'
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 user_src, hostname, action, count
(*) as requests from $log where $filter and hostname is not null group by user_src,
hostname, action order by requests desc)### t where $filter-drilldown and action!='blocked'
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 subtype='infected' and (service in ('smtp',
'SMTP', '25/tcp', '587/tcp', 'smtps', 'SMTPS', '465/tcp') or service in ('pop3', 'POP3',
'110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/tcp', 'pop3s', 'POP3S',
'995/tcp')) 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)### 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)### 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)### 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)### 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, 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 dvid, srcip, dstip, epid, euid, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, 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 dvid, srcip,
dstip, epid, euid, user_src, appid, app, appcat, apprisk, hostname order by 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
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, 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 dvid, srcip, dstip, epid, euid, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, 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 dvid, srcip,
dstip, epid, euid, user_src, appid, app, appcat, apprisk, hostname order by 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
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 | webfilter |

```
select
  catdesc,
  sum(bandwidth) as bandwidth
```

```
from
  ###(select catdesc, 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
/*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_name(app) as app_group,
  appcat,
  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
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_name(app) as app_group,
  service,
  count(*) as sessions,
  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 (
```

```
    & #039;80/tcp', '443/tcp', 'HTTP', 'HTTPS', 'http', 'https') group by app_group, service
having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>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
  domain,
  catdesc,
  sum(visits) as visits
from
  ###(select coalesce(nullifna(hostname), ipstr(`dstip`)) as domain, catdesc, count(*) as
visits from $log where $filter and (eventtype is null or logver>=502000000) and catdesc is
not null group by domain, catdesc order by visits desc)### t group by domain, catdesc 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,
  count(*) as totalnum
from
  $log
where
```

```
  $filter
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,
  count(*) as totalnum
from
  $log
where
  $filter
  and severity =& #039;critical' and nullifna(attack) is not null 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,
  count(*) as totalnum
from
  $log
where
  $filter
  and severity =& #039;high' and nullifna(attack) is not null 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,
  count(*) as totalnum
from
  $log
where
  $filter
  and severity =& #039;medium' and nullifna(attack) is not null 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,
  count(*) as totalnum
from
  $log
where
  $filter
  and severity =& #039;low' and nullifna(attack) is not null 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,
  count(*) as totalnum
from
  $log
where
  $filter
  and severity =& #039;info' and nullifna(attack) is not null 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 (eventtype is null or logver>=502000000) 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, count(*) as totalnum
from $log where $filter and (eventtype is null or logver>=502000000) and nullifna(virus) is
not null group by user_src /*SkipSTART*/order by totalnum desc/*SkipEND*/)### t 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 and (eventtype is null or logver>=502000000) 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-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 dvid, srcip, dstip, epid, euid,
coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, 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 dvid, srcip, dstip, epid, euid, user_src, appid, app, appcat, apprisk,
hostname order by 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 cast('Malware & Botnet C&C' as char(32)) as threat_name, app as threats from
$log-app-ctrl where $filter and lower(appcat)='botnet' group by app)### union all ###(select
cast('Malware & Botnet C&C' as char(32)) as threat_name, virus as threats from $log-virus
where $filter and nullifna(virus) is not null group by virus)### union all ###(select cast
('Malicious & Phishing Sites' as char(32)) as threat_name, hostname as threats from $log-
webfilter where $filter and cat in (26, 61) group by hostname)### union all ###(select cast
('Critical & High Intrusion Attacks' as char(32)) as threat_name, attack as threats from
$log-attack where $filter and severity in ('critical', 'high') group by attack)###) t group
by threat_name order by total_num desc
```

| Dataset Name | Description | Log Category |
|---|---|---|
| Apprisk-Ctrl-Application-Vulnerability | Application vulnerabilities discovered | attack |

```
select
  attack,
  attackid,
  vuln_type,
  cve,
  severity_number,
  count(distinct dstip) as victims,
  count(distinct srcip) as sources,
```

```
  sum(totalnum) as totalnum
from
  ###(select attack, attackid, vuln_type, t2.cve, (case when t1.severity='critical' then 5
when t1.severity='high' then 4 when t1.severity='medium' then 3 when t1.severity='low' then
2 when t1.severity='info' then 1 else 0 end) as severity_number, dstip, srcip, 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 nullifna(attack) is not null and t1.severity is not null
group by attack, attackid, vuln_type, t2.cve, t1.severity, dstip, srcip )### t group by
attack, attackid, vuln_type, severity_number, cve order by severity_number desc, totalnum
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 dvid, srcip, dstip, epid, euid,
coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, 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 dvid, srcip, dstip, epid, euid, user_src, appid, app, appcat, apprisk,
hostname order by 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-Top-20-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 | Application Behavioral Characteristics | traffic |

```
select
  behavior,
  round(
    sum(total_num)* 100 / sum(
      sum(total_num)
    ) over (),
    2
  ) as percentage
from
  (
    ###(select (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_top_app*/select dvid, srcip, dstip, epid, euid, coalesce(nullifna(`user`),
nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, 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 dvid, srcip,
dstip, epid, euid, user_src, appid, app, appcat, apprisk, hostname order by sessions
desc)base### t where lower(appcat) in ('botnet', 'remote.access', 'storage.backup',
'video/audio', 'p2p', 'proxy') and apprisk in ('critical', 'high') group by appcat order by
total_num desc)### union all ###(select 'malicious' as behavior,  count(*) as total_num from
$log-attack where $filter and (logflag&16>0) and severity in ('critical', 'high') group by
behavior)###) t where $filter-drilldown 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, 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 dvid, srcip, dstip, epid, euid, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, 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 dvid, srcip,
dstip, epid, euid, user_src, appid, app, appcat, apprisk, hostname order by 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
group by appcat having sum(bandwidth)>0 order by bandwidth desc
```

| 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 dvid, srcip, dstip, epid, euid, coalesce
(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, 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 dvid, srcip, dstip, epid, euid, user_src, appid, app, appcat, apprisk,
hostname order by sessions desc)base### t where nullifna(appcat) is not null 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-Top-Web-Applications-by-Bandwidth | Top 25 Web Categories by Bandwidtih | traffic |

```
select
  d_risk,
  id,
  name,
```

```
  technology,
  count(distinct f_user) as user_num,
  sum(bandwidth) as bandwidth,
  sum(num_session) as num_session
from
  ###(select risk as d_risk, t2.id, t2.name, t2.technology, coalesce(nullifna(t1.`user`),
nullifna(t1.`unauthuser`), ipstr(t1.`srcip`)) as f_user, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth, count(*) as num_session from $log t1 inner join app_mdata t2 on
t1.appid=t2.id where $filter and (logflag&1>0) and nullifna(app) is not null and service in
('80/tcp', '443/tcp', 'HTTP', 'HTTPS', 'http', 'https') group by risk, t2.id, t2.name,
t2.technology, f_user)### t group by d_risk, id, name, technology 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 |
|---|---|---|
| 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 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)### union all ###(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)### union all ###(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, 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 )### 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-Malicious-Files-Detected-By-FortiCloud-Sandbox | Files detected by FortiCloud Sandbox | virus |

```
select
  filename,
  analyticscksum,
  count(distinct dstip) as victims,
  count(distinct srcip) as source
from
  ###(select filename, analyticscksum, dstip, srcip from $log where $filter and filename is
not null and logid_to_int(logid)=9233 and analyticscksum is not null group by filename,
analyticscksum, srcip, dstip)### t group by filename, analyticscksum order by victims desc,
source desc
```

| 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
  srcname as caller,
  count(*) as totalnum
from
  $log
where
  $filter
  and lower(appcat)=& #039;voip' and app='sccp' and action='block' and srcname is not null
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
  srcname as caller,
  count(*) as totalnum
from
  $log
where
  $filter
  and srcname is not null
  and lower(appcat)=& #039;voip' and app='sip' and action='block' group by caller order by
totalnum 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)### 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 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)### 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, 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 dvid, srcip, dstip, epid, euid, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, 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 dvid, srcip,
dstip, epid, euid, user_src, appid, app, appcat, apprisk, hostname order by 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
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 dvid, srcip, dstip, epid, euid, coalesce
(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, 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 dvid, srcip, dstip, epid, euid, user_src, appid, app, appcat, apprisk,
hostname order by sessions desc)base### t where nullifna(appcat) is not null 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 Bandwidtih | traffic |

```
select
  d_risk,
  name,
  app_cat,
  technology,
  count(distinct f_user) as users,
  sum(bandwidth) as bandwidth,
  sum(num_session) as sessions
from
  ###(select risk as d_risk, t2.app_cat, t2.name, t2.technology, coalesce(nullifna
(t1.`user`), nullifna(t1.`unauthuser`), ipstr(t1.`srcip`)) as f_user, sum(coalesce(sentbyte,
0)+coalesce(rcvdbyte, 0)) as bandwidth, count(*) as num_session from $log t1 inner join app_
mdata t2 on t1.appid=t2.id where $filter and (logflag&1>0) and nullifna(app) is not null and
service in ('80/tcp', '443/tcp', 'HTTP', 'HTTPS', 'http', 'https') group by risk, t2.app_
cat, t2.name, t2.technology, f_user)### t group by d_risk, name, app_cat, technology order
by 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 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)### union all ###(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)### union all ###(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, 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 dstip) as victims,
  count(distinct srcip) as source,
  sum(total_num) as total_num
from
  ###(select virus, virusid_to_str(virusid, eventtype) as virusid_s, srcip, dstip, (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, srcip, dstip 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 dstip) as victims,
  count(distinct srcip) as source,
  sum(total_num) as total_num
from
  (
    ###(select app, appid, cast('Botnet C&C' as char(32)) as malware_type, srcip, dstip,
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, srcip, dstip order by total_num
```

```
desc)### union all ###(select attack, 0 as appid, cast('Botnet C&C' as char(32)) as malware_
type, srcip, dstip, count(*) as total_num from $log-attack where $filter and (logflag&16>0)
group by attack, appid, malware_type, srcip, dstip order by total_num desc)###) 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,
  (
    lower(service) || & #039;://' || hostname || url) 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
```

| 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 (lower(service) || '://' || hostname || url) as phishing_site, dstip, srcip,
```

```
count(*) as total from $log where $filter and lower(service) in ('http', 'https') and
hostname is not null 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 dstip) as victims,
  count(distinct srcip) as sources,
  sum(totalnum) as totalnum
from
  ###(select attack, attackid, vuln_type, t2.cve, (case when t1.severity='critical' then 5
when t1.severity='high' then 4 when t1.severity='medium' then 3 when t1.severity='low' then
2 when t1.severity='info' then 1 else 0 end) as severity_number, dstip, srcip, 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 nullifna(attack) is not null and t1.severity is not null
group by attack, attackid, vuln_type, t2.cve, t1.severity, dstip, srcip )### t 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)### 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 from $log where $filter and subtype='av' and virus is not null group by f_user, host_
name, virus, file
```

| 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
  $hour_of_day as hourstamp,
  count(*) as totalnum
from
  $log
where
  $filter
  and proto =& #039;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
  $hour_of_day as hourstamp,
  sum(duration) as sccp_usage
from
  $log
where
  $filter
  and proto =& #039;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(*) as totalnum
from
  $log
where
  $filter
  and proto =& #039;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
  $hour_of_day as hourstamp,
  count(*) as totalnum
from
  $log
where
```

```
  $filter
  and proto =& #039;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(*) as totalnum
from
  $log
where
  $filter
  and proto =& #039;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, dstip,
srcip, count(*) as totalnum from $log-attack where $filter and (logflag&16>0) group by
attack, user_src, timestamp, hostname, severity, crlevel, eventtype, service, dstip, srcip
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 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)### union all
###(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 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, dstip, srcip, count
(*) as totalnum from $log-attack where $filter and (logflag&16>0) group by attack, user_src,
timestamp, hostname, severity, crlevel, eventtype, service, dstip, srcip 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 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, dstip, srcip, count(*) as
totalnum from $log-attack where $filter and (logflag&16>0) group by attack, user_src,
timestamp, hostname, severity, crlevel, eventtype, service, dstip, srcip 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, dstip, srcip, count(*) as
totalnum from $log-attack where $filter and (logflag&16>0) group by attack, user_src,
timestamp, hostname, severity, crlevel, eventtype, service, dstip, srcip 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 $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)###
union all ###(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)### union all ###(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 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 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)### 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)###
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)###
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 from $log where $filter and
subtype='compliance-check' and result in ('fail','pass') group by result, reason)### 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 from $log where $filter and subtype='compliance-check' and
result='fail' group by status, reason)### t 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 from $log where $filter and subtype='compliance-check' and
result='pass' group by status, reason)### t 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 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)### 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', '25/tcp', '587/tcp', 'smtps', 'SMTPS', '465/tcp') or service in
('pop3', 'POP3', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/tcp', 'pop3s',
'POP3S', '995/tcp')) 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', '25/tcp', '587/tcp', 'smtps', 'SMTPS',
'465/tcp') or service in ('pop3', 'POP3', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps',
'IMAPS', '993/tcp', 'pop3s', 'POP3S', '995/tcp')) 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 from $log where $filter and subtype='wireless' and
action='client-authentication' group by hod, stamac)### 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-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','doc
x','xls','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 file_name_ext
(filename) as suffix, count(*) as total_num from $log where $filter and dtype='fortisandbox'
and nullifna(filename) is not null group by suffix 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 file_name_ext(filename) as suffix, count(*) as total_num from
$log where $filter and dtype='fortisandbox' and nullifna(filename) is not null group by
suffix 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 $log where $filter and dtype='fortisandbox' and file_name_ext(filename)='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
  srcip,
  count(*) as total_num
from
  $log
where
  $filter
  and dtype =& #039;fortisandbox' and nullifna(filename) is not null and fsaverdict not in
('clean','submission failed') group by srcip order by total_num desc
```

| 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 dstip) as victims,
  count(distinct srcip) as sources,
  sum(totalnum) as totalnum
from
  ###(select attack, attackid, vuln_type, t2.cve, (case when t1.severity='critical' then 5
when t1.severity='high' then 4 when t1.severity='medium' then 3 when t1.severity='low' then
2 when t1.severity='info' then 1 else 0 end) as severity_number, dstip, srcip, 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 nullifna(attack) is not null and t1.severity is not null
group by attack, attackid, vuln_type, t2.cve, t1.severity, dstip, srcip )### t group by
attack, attackid, vuln_type, severity_number, cve order by severity_number desc, totalnum
desc
```

| Dataset Name | Description | Log Category |
|---|---|---|
| ctap-apprisk-ctrl-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 dstip) as victims, count(distinct srcip) as
```

```
source, sum(total_num) as total_num from (###(select app as malware, appcat, appid, app,
dstip, srcip, count(*) as total_num from $log-app-ctrl where $filter and lower
(appcat)='botnet' group by malware, appcat, appid, app, dstip, srcip, app order by total_num
desc)### union all ###(select virus as malware, 'null' as appcat, 0 as appid, service as
app, dstip, srcip, count(*) as total_num from $log-virus where $filter and virus is not null
group by malware, appcat, app, appid, dstip, srcip order by total_num desc)### union all ###
(select attack as malware, 'null' as appcat, 0 as appid, service as app, dstip, srcip, count
(*) as total_num from $log-attack where $filter and (logflag&16>0) group by malware, appcat,
app, appid, dstip, srcip order by total_num desc)###) t group by malware, malware_type, app,
appid order by total_num desc
```

| 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-HTTP-SSL-Traffic-Ratio | HTTP SSL Traffic Ratio | traffic |

```
select
  (
    case when service in (
      & #039;80/tcp', '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 ('80/tcp', '443/tcp', '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-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-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-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-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-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-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-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-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-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_name(app) as app_group,
  service,
  count(*) as sessions,
  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 (
    & #039;80/tcp', '443/tcp', 'HTTP', 'HTTPS', 'http', 'https') group by app_group, service
having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>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
  domain,
  catdesc,
  sum(visits) as visits
from
  ###(select coalesce(nullifna(hostname), ipstr(`dstip`)) as domain, catdesc, count(*) as
visits from $log where $filter and (eventtype is null or logver>=502000000) and catdesc is
not null group by domain, catdesc order by visits desc)### t group by domain, catdesc 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-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(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  ) as bandwidth
from
  $log - traffic
where
  $filter
  and hostname is not null
  and (
    logflag&1>0
  )
group by
  hostname
having
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  )>0
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)### 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 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)### 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 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)### 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 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)### 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)### 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)### 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)### 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)### 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 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
    where
      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
      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 from $log where $filter and nullifna
(os) is not null group by os1, hostname)### 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 from $log where $filter and
nullifna(srcname) is not null and lower(os) like '%windows%' group by srcname, hostname)###
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 from $log where $filter and
```

```
nullifna(srcname) is not null and lower(os) like '%mac os%' group by srcname, hostname)### 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 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)### 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
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 from $log where $filter and nullifna(vulnname) is not
null and nullifna(vulnseverity) is not null group by vulnseverity, vulnname)### 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 from $log where $pre_period
$filter and nullifna(vulnname) is not null group by vulnid, vulnname, vulnseverity, vulncat,
hostname)### 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 from $log where $filter and
nullifna(vulnname) is not null group by vulnid, vulnname, vulnseverity, vulncat,
hostname)### 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 intfinfo) 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 intfinfo) 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 from $log where
$filter and nullifna(vulnname) is not null and nullifna(vulnseverity) is not null group by
hostname, os1, vulnname, vulnseverity, vulnid)### 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 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)### union all ###(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)### union all ###(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 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 (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)### union all ###(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)### union all ###(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)###) 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 $day_of_week as daystamp, count(*) as total_num from $log-virus where $filter
and nullifna(virus) is not null group by daystamp)### union all ###(select $day_of_week as
daystamp, count(*) as total_num from $log-attack where $filter and nullifna(attack) is not
null group by daystamp)### union all ###(select $day_of_week as daystamp, count(*) as total_
num from $log-app-ctrl where $filter and lower(appcat)='botnet' 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 $day_of_week as daystamp, count(*) as total_num from $log-virus where $filter
and nullifna(virus) is not null group by daystamp)### union all ###(select $day_of_week as
daystamp, count(*) as total_num from $log-attack where $filter and nullifna(attack) is not
null group by daystamp)### union all ###(select $day_of_week as daystamp, count(*) as total_
num from $log-app-ctrl where $filter and lower(appcat)='botnet' 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 dstip) as victim,
  count(distinct srcip) as source,
  sum(total_num) as total_num
from
  (
    ###(select app as virus, 'Botnet C&C' as malware_type, apprisk::text as risk_level,
dstip, srcip, 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, dstip, srcip
order by total_num desc)### union all ###(select virus, (case when eventtype='botnet' then
'Botnet C&C' else 'Virus' end) as malware_type, crlevel::text as risk_level, dstip, srcip,
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, dstip, srcip order by total_num
desc)### union all ###(select attack as virus, (case when eventtype='botnet' then 'Botnet
C&C' else 'Virus' end) as malware_type, crlevel::text as risk_level, dstip, srcip, count(*)
as total_num from $log-attack where $filter and (logflag&16>0) and crlevel is not null group
```

```
by virus, malware_type, crlevel, dstip, srcip order by total_num desc)###) 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='ipsec' or left
(tunneltype, 3)='ssl') and action in ('ssl-login-fail', 'ipsec-login-fail') and coalesce
(nullifna(`xauthuser`), nullifna(`user`)) is not null group by f_user, tunneltype)### 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 from $log where $filter and subtype='compliance-check' and result='fail' group by
devid, reason)### 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
    where
      euid >= 1024
  );
select
  coalesce(
    nullifna(epname),
    nullifna(
      ipstr(`srcip`)
    ),
    & #039;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 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 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 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 cs_count>0 union all (select epid, day_st as
itime, unnest(dvid) as dvid_s from $ADOMTBL_PLHD_IOC_VERDICT where cs_count>0)) t inner join
devtable 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 cs_count>0 union all (select epid, day_st as
itime, unnest(dvid) as dvid_s from $ADOMTBL_PLHD_IOC_VERDICT where cs_count>0)) t inner join
devtable 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(
    nullifna(epname),
    nullifna(
      ipstr(`srcip`)
    ),
    & #039;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 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 bl_count=0 and cs_
count>0)) tvdt inner join devtable 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 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 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=0 and cs_count>0) ta2)) tt1 inner join
devtable 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,
  srcip,
  string_agg(
    distinct `virus`,
    & #039;,') as virus_agg, count(distinct ipstr(`dstip`)) 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, srcip, virus, dstip, 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 srcip, f_user, virus, dstip
order by total_num desc)### t group by srcip, 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,
  srcip
from
  (
    ###(select app as threat_name, 1 as cat_id, srcip from $log-app-ctrl where $pre_period
$filter and nullifna(app) is not null and lower(appcat)='botnet' group by threat_name, cat_
id, srcip)### union all ###(select virus as threat_name, 2 as cat_id, srcip from $log-virus
where $pre_period $filter and nullifna(virus) is not null group by threat_name, cat_id,
srcip)### union all ###(select attack as threat_name, 3 as cat_id, srcip from $log-attack
where $pre_period $filter and nullifna(attack) is not null group by threat_name, cat_id,
srcip)###) t; create temporary table rpt_tmptbl_2 as select daystamp, threat_name, cat_id,
srcip from (###(select $DAY_OF_MONTH as daystamp, app as threat_name, 1 as cat_id, srcip
from $log-app-ctrl where $filter and nullifna(app) is not null and lower(appcat)='botnet'
group by daystamp, threat_name, cat_id, srcip order by daystamp)### union all ###(select
$DAY_OF_MONTH as daystamp, virus as threat_name, 2 as cat_id, srcip from $log-virus where
$filter and nullifna(virus) is not null group by daystamp, threat_name, cat_id, srcip order
by daystamp)### union all ###(select $DAY_OF_MONTH as daystamp, attack as threat_name, 3 as
cat_id, srcip from $log-attack where $filter and nullifna(attack) is not null group by
daystamp, threat_name, cat_id, srcip order by daystamp)###) 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 srcip) 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,
  srcip
from
  (
    ###(select app as threat_name, 1 as cat_id, srcip from $log-app-ctrl where $pre_period
$filter and nullifna(app) is not null and lower(appcat)='botnet' group by threat_name, cat_
id, srcip)### union all ###(select virus as threat_name, 2 as cat_id, srcip from $log-virus
where $pre_period $filter and nullifna(virus) is not null group by threat_name, cat_id,
srcip)### union all ###(select attack as threat_name, 3 as cat_id, srcip from $log-attack
where $pre_period $filter and nullifna(attack) is not null group by threat_name, cat_id,
srcip)###) t; create temporary table rpt_tmptbl_2 as select timestamp, threat_name, cat_id,
srcip from (###(select $flex_timestamp as timestamp, app as threat_name, 1 as cat_id, srcip
from $log-app-ctrl where $filter and nullifna(app) is not null and lower(appcat)='botnet'
group by timestamp, threat_name, cat_id, srcip order by timestamp)### union all ###(select
$flex_timestamp as timestamp, virus as threat_name, 2 as cat_id, srcip from $log-virus where
$filter and nullifna(virus) is not null group by timestamp, threat_name, cat_id, srcip order
by timestamp)### union all ###(select $flex_timestamp as timestamp, attack as threat_name, 3
as cat_id, srcip from $log-attack where $filter and nullifna(attack) is not null group by
timestamp, threat_name, cat_id, srcip order by timestamp)###) t; select $flex_datetime
(timestamp) as timescale, count(distinct srcip) 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 from $log where $pre_period
$filter and nullifna(vulnname) is not null group by vulnid, vulnname, vulnseverity, vulncat,
hostname)### 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 from $log where $filter and
nullifna(vulnname) is not null group by vulnid, vulnname, vulnseverity, vulncat,
hostname)### 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 from $log where $pre_period
$filter and nullifna(vulnname) is not null group by vulnid, vulnname, vulnseverity, vulncat,
hostname)### 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 from $log where $filter and
nullifna(vulnname) is not null group by vulnid, vulnname, vulnseverity, vulncat,
hostname)### 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 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, 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, srcip, 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 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, 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, srcip, 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 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, 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, srcip, 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 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, 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, srcip, 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)### 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)### 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)### 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)### 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)### 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)### 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)### 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)### 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)### 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)### 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,
  count(*) as cnt
from
  $incident
group by
  status
order by
  status
```

| Dataset Name | Description | Log Category |
|---|---|---|
| incident-Incident-Count-by-Status-Donut | Incident status distribution | |

```
select
  status,
  count(*) as cnt
from
  $incident
group by
  status
order by
  status
```

| Dataset Name | Description | Log Category |
|---|---|---|
| incident-Open-Incident-Count-Timeline | Incident count by status over time | |

```
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
  $cust_time_filter(agg_time)
group by
  hodex
order by
  hodex
```

| Dataset Name | Description | Log Category |
|---|---|---|
| incident-Closed-Incident-Count-Timeline | Incident count by status over time | |

```
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
  $cust_time_filter(agg_time)
group by
  hodex
order by
  hodex
```

| Dataset Name | Description | Log Category |
|---|---|---|
| Top-10-Apps-by-Bandwidth | Top applications by bandwidth usage | traffic |

```
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
```

| Dataset Name | Description | Log Category |
|---|---|---|
| Top-10-User-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 |
|---|---|---|
| Top-10-Applications-by-Number-of-Users | Top Applications by number of users | traffic |

```
select
  app_group_name(app) as app_group,
  count(distinct user_src) as number
from
  ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
app, appcat from $log where $filter and (logflag&1>0) and nullifna(app) is not null group by
user_src, app, appcat)### t group by app_group order by number desc
```

| Dataset Name | Description | Log Category |
|---|---|---|
| Top-10-User-by-Session | Top user by session count | traffic |

```
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
```

| Dataset Name | Description | Log Category |
|---|---|---|
| Top-10-Apps-by-Session | Top applications by bandwidth usage | traffic |

```
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
```

| Dataset Name | Description | Log Category |
|---|---|---|
| Applications-by-Risk-Level | Applications by Risk Level | traffic |

```
select
  app_group_name(app) as app_group,
  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(sent) as sent, sum(received) as received, sum(bandwidth) as
bandwidth from ###(select appid, app, appcat, sum(coalesce(sentbyte, 0)) as sent, sum
(coalesce(rcvdbyte, 0)) as received, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as
bandwidth, count(*) as sessions from $log where $filter and (logflag&1>0) group by appid,
app, appcat order by bandwidth desc)### t1 inner join app_mdata t2 on lower(t1.app)=lower
(t2.name) group by app_group, appcat order by d_risk desc, bandwidth desc
```

| 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
  (
    select
      & #039;Events' as item, num_cur, num_pre, (num_cur-num_pre) as num_diff from (select
```

```
(select count(*) from $event where $cust_time_filter(alerttime,TODAY)) as num_cur, (select
count(*) from $event where $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) 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
  (
    select
      & #039;Events' as item, num_cur, num_pre, (num_cur-num_pre) as num_diff from (select
(select count(*) from $event where $cust_time_filter(alerttime)) as num_cur, (select count
(*) from $event where $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)
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
  (
    select
      & #039;Events' as item, num_cur, num_pre, (num_cur-num_pre) as num_diff from (select
(select count(*) from $event where $cust_time_filter(alerttime,TODAY)) as num_cur, (select
count(*) from $event where $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) t1 full join (select 'Events' as item, num_cur, num_pre, (num_cur-num_pre) as num_diff
from (select (select count(*) from $event where $cust_time_filter(alerttime)) as num_cur,
(select count(*) from $event where $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) t2 on t1.item=t2.item order by t1.item
```

| Dataset Name | Description | Log Category |
|---|---|---|
| soc-Total-Event-by-Severity | Total Events by Severity | |

```
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 group by severity
order by severity
```

| 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 (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 $cust_time_filter(agg_
time) group by dom order by dom) t) 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
  (
    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 group by
severity, triggername order by severity, triggername
```

| Dataset Name | Description | Log Category |
|---|---|---|
| soc-Total-Incident-by-Severity | Total Incidents by Severity | |

```
select
  severity,
  count(*) as num_inc
from
  $incident
group by
  severity
order by
  severity
```

| Dataset Name | Description | Log Category |
|---|---|---|
| soc-Total-Event-vs-Incident-History | Total Events vs Incidents History | |

```
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
```

| Dataset Name | Description | Log Category |
|---|---|---|
| soc-Incident-List | List of Incidents | |

```
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
```

| Dataset Name | Description | Log Category |
|---|---|---|
| soc-Incident-by-Severity | Incidents by Severity | |

```
select
  severity,
  count(*) as incnum
from
  $incident
where
  $cust_time_filter(createtime)
group by
  severity
order by
  incnum desc
```

| Dataset Name | Description | Log Category |
|---|---|---|
| soc-Incident-by-Status | Incidents by Status | |

```
select
  status,
  count(*) as incnum
from
  $incident
where
  $cust_time_filter(createtime)
group by
  status
order by
  incnum desc
```

| Dataset Name | Description | Log Category |
|---|---|---|
| soc-Incident-by-Category-Unresolved | Unresolved Incidents by Category | |

```
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
```

| Dataset Name | Description | Log Category |
|---|---|---|
| soc-Incident-by-Severity-Unresolved | Unresolved Incidents by Severity | |

```
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
```

| Dataset Name | Description | Log Category |
|---|---|---|
| soc-Incident-Timeline-by-Category | Incidents Timeline by Category | |

```
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
```

| Dataset Name | Description | Log Category |
|---|---|---|
| soc-Incident-List-Unresolved | List of Unresolved Incidents | |

```
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
```

| 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,
  (
    & #039; ' || devid) as id_devid, ip, platform, os, '1' as total_num from $func-fgt-
inventory as t1 where exists (select 1 from devtable t2 where $dev_filter and
t2.devid=t1.devid) order by devname
```

| Dataset Name | Description | Log Category |
|---|---|---|
| fgt-inventory-hardware | FortiGate Monitoring Inventory Hardware | event |

```
select
  platform,
  count(*) as total_num
from
  $func - fgt - inventory as t1
where
  exists (
    select
      1
    from
      devtable t2
    where
      $dev_filter
      and t2.devid = t1.devid
  )
group by
  platform
order by
  total_num desc
```

| Dataset Name | Description | Log Category |
|---|---|---|
| fgt-inventory-software | FortiGate Monitoring Inventory Software | event |

```
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 t2 where $dev_filter and
t2.devid=t1.devid) group by platform, os 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)### 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)### 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)### 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)### 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)### 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)### 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)### 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)### 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)### 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)### 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)### 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)### 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)### 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)### 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)### 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)### 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)### 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)### 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)### 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)### 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)### 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 (select $flex_
timestamp(timestamp) as tmstamp, dvid, 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 intfstats where $cust_time_filter
(timestamp) group by tmstamp, dvid, intfname) t1 left join devtable t2 on t1.dvid = t2.dvid
where $filter-drilldown 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 (select $flex_
timestamp(timestamp) as tmstamp, tbl_intf.dvid, 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 (select distinct dvid from ###(select dvid
from $log-event where $filter and action='perf-stats' group by dvid)### t) tbl_log inner
join intfstats tbl_intf on tbl_log.dvid = tbl_intf.dvid where $cust_time_filter(timestamp)
group by tmstamp, tbl_intf.dvid, intfname) t1 left join devtable t2 on t1.dvid = t2.dvid
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 (select $flex_
timestamp(timestamp) as tmstamp, dvid, 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 intfstats where $cust_time_filter
(timestamp) group by tmstamp, dvid, intfname) t1 left join devtable t2 on t1.dvid = t2.dvid
where $filter-drilldown 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 (select $flex_
timestamp(timestamp) as tmstamp, tbl_intf.dvid, 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 (select distinct dvid from ###(select dvid
from $log-event where $filter and action='perf-stats' group by dvid)### t) tbl_log inner
join intfstats tbl_intf on tbl_log.dvid = tbl_intf.dvid where $cust_time_filter(timestamp)
group by tmstamp, tbl_intf.dvid, intfname) t1 left join devtable t2 on t1.dvid = t2.dvid
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 (select $flex_
timestamp(timestamp) as tmstamp, dvid, 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 intfstats where $cust_time_filter
(timestamp) group by tmstamp, dvid, intfname) t1 left join devtable t2 on t1.dvid = t2.dvid
where $filter-drilldown 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 (select $flex_
timestamp(timestamp) as tmstamp, tbl_intf.dvid, 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 (select distinct dvid from ###(select dvid
from $log-event where $filter and action='perf-stats' group by dvid)### t) tbl_log inner
join intfstats tbl_intf on tbl_log.dvid = tbl_intf.dvid where $cust_time_filter(timestamp)
group by tmstamp, tbl_intf.dvid, intfname) t1 left join devtable t2 on t1.dvid = t2.dvid
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 (select $flex_
timestamp(timestamp) as tmstamp, dvid, 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 intfstats where $cust_time_filter
(timestamp) group by tmstamp, dvid, intfname) t1 left join devtable t2 on t1.dvid = t2.dvid
where $filter-drilldown 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 (select $flex_
timestamp(timestamp) as tmstamp, tbl_intf.dvid, 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 (select distinct dvid from ###(select dvid
from $log-event where $filter and action='perf-stats' group by dvid)### t) tbl_log inner
join intfstats tbl_intf on tbl_log.dvid = tbl_intf.dvid where $cust_time_filter(timestamp)
group by tmstamp, tbl_intf.dvid, intfname) t1 left join devtable t2 on t1.dvid = t2.dvid
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 (select $flex_
timestamp(timestamp) as tmstamp, tbl_intf.dvid, 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 (select distinct dvid from ###(select dvid
from $log-event where $filter and action='perf-stats' group by dvid)### t) tbl_log inner
join intfstats tbl_intf on tbl_log.dvid = tbl_intf.dvid where $cust_time_filter(timestamp)
group by tmstamp, tbl_intf.dvid, intfname) t1 left join devtable t2 on t1.dvid = t2.dvid
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
  from_dtime(dtime) as time_s,
  devid,
  coalesce(
    nullifna(logdesc),
    msg
  ) as msg_desc
from
  $log
where
  $filter
  and logid_to_int(logid) in (22105, 22107)
order by
  time_s desc
```

| Dataset Name | Description | Log Category |
|---|---|---|
| fgt-env-faults-fan | FortiGate Fan Faults | event |

```
select
  from_dtime(dtime) as time_s,
  devid,
  coalesce(
    nullifna(logdesc),
    msg
  ) as msg_desc
from
  $log
where
  $filter
  and logid_to_int(logid)= 22108
```

```
order by
  time_s desc
```

| Dataset Name | Description | Log Category |
|---|---|---|
| fgt-env-faults-temperature | FortiGate Temperatre Too High | event |

```
select
  from_dtime(dtime) as time_s,
  devid,
  coalesce(
    nullifna(logdesc),
    msg
  ) as msg_desc
from
  $log
where
  $filter
  and logid_to_int(logid)= 22109
order by
  time_s desc
```

| Dataset Name | Description | Log Category |
|---|---|---|
| Behaviour-Banned-Application | Bullying Chat Search and Message Logging | app-ctrl |

```
select
  filename,
  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 $filter and (lower(app) in
('facebook_post', 'facebook_chat', 'twitter_post', 'youtube_video.access', 'gmail_chat',
'gmail_send.message', 'linkedin_post', 'vimeo_video.access', 'google.search_search.phrase',
'bing.search_search.phrase')) and ($bully_keywords) order by itime desc)### t group by
filename order by requests desc
```

| Dataset Name | Description | Log Category |
|---|---|---|
| Behaviour-Banned-User | Bullying Chat Search and Message Logging | app-ctrl |

```
select
  filename,
  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 (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')) and ($bully_keywords) order by itime desc)### t group by
filename order by requests desc
```

| Dataset Name | Description | Log Category |
|---|---|---|
| Behaviour-Banned-User-Drilldown | Bullying Chat Search and Message Logging | app-ctrl |

```
select
  filename,
  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 $filter and (lower(app) in
('facebook_post', 'facebook_chat', 'twitter_post', 'youtube_video.access', 'gmail_chat',
'gmail_send.message', 'linkedin_post', 'vimeo_video.access', 'google.search_search.phrase',
'bing.search_search.phrase')) and ($bully_keywords) order by itime desc)### t group by
filename order by requests desc
```

| Dataset Name | Description | Log Category |
|---|---|---|
| behaviour-banned | Bullying Chat Search and Message Logging | app-ctrl |

```
select
  filename,
  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 $filter and (lower(app) in
('facebook_post', 'facebook_chat', 'twitter_post', 'youtube_video.access', 'gmail_chat',
'gmail_send.message', 'linkedin_post', 'vimeo_video.access', 'google.search_search.phrase',
'bing.search_search.phrase')) and ($bully_keywords) order by itime desc)### t 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 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 (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')) and ($banned_keywords) order by itime desc)### 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 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 $filter and (lower(app) in
('facebook_post', 'facebook_chat', 'twitter_post', 'youtube_video.access', 'gmail_chat',
'gmail_send.message', 'linkedin_post', 'vimeo_video.access', 'google.search_search.phrase',
'bing.search_search.phrase')) and ($banned_keywords) order by itime desc)### 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 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 $filter and (lower(app) in
('facebook_post', 'facebook_chat', 'twitter_post', 'youtube_video.access', 'gmail_chat',
'gmail_send.message', 'linkedin_post', 'vimeo_video.access', 'google.search_search.phrase',
'bing.search_search.phrase')) and ($banned_keywords) order by itime desc)### 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 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 (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')) and ($banned_keywords) order by itime desc)### 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 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 $filter and (lower(app) in
('facebook_post', 'facebook_chat', 'twitter_post', 'youtube_video.access', 'gmail_chat',
'gmail_send.message', 'linkedin_post', 'vimeo_video.access', 'google.search_search.phrase',
'bing.search_search.phrase')) and ($banned_keywords) order by itime desc)### t group by
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
  from_itime(itime) as i_time,
  coalesce(
    nullifna(`user`),
    nullifna(`unauthuser`),
    ipstr(`srcip`)
  ) as f_user,
  srcip,
  filename
from
  $log
where
  $filter
  and lower(app)= lower(
    & #039;Facebook_Post') and filename is not null 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 from $log where $filter and
lower(app)=lower('Facebook_Chat') and filename is not null)### t group by filename order by
requests desc
```

| Dataset Name | Description | Log Category |
|---|---|---|
| Twitter-Posts | Twitter Posts | app-ctrl |

```
select
  from_itime(itime) as i_time,
  coalesce(
    nullifna(`user`),
    nullifna(`unauthuser`),
    ipstr(`srcip`)
  ) as f_user,
  srcip,
  filename
from
  $log
where
  $filter
  and lower(app)= lower(
    & #039;Twitter_Post') and filename is not null 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 from $log where $filter and
lower(app)=lower('LinkedIn_Post') and filename is not null)### t 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, devid, vd, interface, 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, min(sdwan_status) as sdwan_status from (select itime, csf, devid, vd, interface,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE 0 END) AS latency,
(CASE WHEN link_status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link_status=1 THEN
packetloss ELSE 100 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 from (select itime, csf, devid, vd, interface, coalesce
(healthcheck, name) as healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_
status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN msg LIKE '%SLA failed%' THEN 1 WHEN
status='down' 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(inbandwidth) as inbandwidth,
convert_unit_to_num(outbandwidth) as outbandwidth, convert_unit_to_num(bibandwidth) as
bibandwidth from $log where $filter and logid_to_int(logid)=22925 AND status IS NOT NULL AND
interface IS NOT NULL AND (healthcheck IS NOT NULL OR name IS NOT NULL)) t) t group by
timestamp, csf, devid, vd, interface, 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,
      sum(latency)/ sum(count) as latency
    from
      ###(select $flex_timestamp as timestamp, csf, devid, vd, interface, 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, min(sdwan_status) as sdwan_status from (select itime, csf, devid, vd, interface,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE 0 END) AS latency,
(CASE WHEN link_status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link_status=1 THEN
packetloss ELSE 100 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 from (select itime, csf, devid, vd, interface, coalesce
(healthcheck, name) as healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_
status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN msg LIKE '%SLA failed%' THEN 1 WHEN
status='down' 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(inbandwidth) as inbandwidth,
convert_unit_to_num(outbandwidth) as outbandwidth, convert_unit_to_num(bibandwidth) as
bibandwidth from $log where $filter and logid_to_int(logid)=22925 AND status IS NOT NULL AND
interface IS NOT NULL AND (healthcheck IS NOT NULL OR name IS NOT NULL)) t) t group by
timestamp, csf, devid, vd, interface, 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, devid, vd, interface, 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, min(sdwan_status) as
sdwan_status from (select itime, csf, devid, vd, interface, healthcheck, link_status, (CASE
WHEN link_status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link_status=1 THEN jitter
ELSE 0 END) AS jitter, (CASE WHEN link_status=1 THEN packetloss ELSE 100 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
from (select itime, csf, devid, vd, interface, coalesce(healthcheck, name) as healthcheck,
(CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, latency::float as latency,
jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE WHEN
```

```
msg LIKE '%SLA failed%' THEN 1 WHEN status='down' 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
(inbandwidth) as inbandwidth, convert_unit_to_num(outbandwidth) as outbandwidth, convert_
unit_to_num(bibandwidth) as bibandwidth from $log where $filter and logid_to_int
(logid)=22925 AND status IS NOT NULL AND interface IS NOT NULL AND (healthcheck IS NOT NULL
OR name IS NOT NULL)) t) t group by timestamp, csf, devid, vd, interface, 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 10)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,
      sum(jitter)/ sum(count) as jitter
    from
      ###(select $flex_timestamp as timestamp, csf, devid, vd, interface, 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, min(sdwan_status) as sdwan_status from (select itime, csf, devid, vd, interface,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE 0 END) AS latency,
(CASE WHEN link_status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link_status=1 THEN
packetloss ELSE 100 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 from (select itime, csf, devid, vd, interface, coalesce
(healthcheck, name) as healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_
status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN msg LIKE '%SLA failed%' THEN 1 WHEN
status='down' 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(inbandwidth) as inbandwidth,
convert_unit_to_num(outbandwidth) as outbandwidth, convert_unit_to_num(bibandwidth) as
bibandwidth from $log where $filter and logid_to_int(logid)=22925 AND status IS NOT NULL AND
interface IS NOT NULL AND (healthcheck IS NOT NULL OR name IS NOT NULL)) t) t group by
timestamp, csf, devid, vd, interface, 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, devid, vd, interface, 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, min(sdwan_status) as
sdwan_status from (select itime, csf, devid, vd, interface, healthcheck, link_status, (CASE
WHEN link_status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link_status=1 THEN jitter
ELSE 0 END) AS jitter, (CASE WHEN link_status=1 THEN packetloss ELSE 100 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
from (select itime, csf, devid, vd, interface, coalesce(healthcheck, name) as healthcheck,
(CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, latency::float as latency,
jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE WHEN
msg LIKE '%SLA failed%' THEN 1 WHEN status='down' 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
(inbandwidth) as inbandwidth, convert_unit_to_num(outbandwidth) as outbandwidth, convert_
unit_to_num(bibandwidth) as bibandwidth from $log where $filter and logid_to_int
(logid)=22925 AND status IS NOT NULL AND interface IS NOT NULL AND (healthcheck IS NOT NULL
OR name IS NOT NULL)) t) t group by timestamp, csf, devid, vd, interface, 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 10)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,
      sum(packetloss)/ sum(count) as packetloss
    from
      ###(select $flex_timestamp as timestamp, csf, devid, vd, interface, 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, min(sdwan_status) as sdwan_status from (select itime, csf, devid, vd, interface,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE 0 END) AS latency,
(CASE WHEN link_status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link_status=1 THEN
packetloss ELSE 100 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 from (select itime, csf, devid, vd, interface, coalesce
(healthcheck, name) as healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_
status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN msg LIKE '%SLA failed%' THEN 1 WHEN
status='down' 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(inbandwidth) as inbandwidth,
convert_unit_to_num(outbandwidth) as outbandwidth, convert_unit_to_num(bibandwidth) as
bibandwidth from $log where $filter and logid_to_int(logid)=22925 AND status IS NOT NULL AND
interface IS NOT NULL AND (healthcheck IS NOT NULL OR name IS NOT NULL)) t) t group by
timestamp, csf, devid, vd, interface, 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, devid, vd, interface, 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, min(sdwan_status) as
sdwan_status from (select itime, csf, devid, vd, interface, healthcheck, link_status, (CASE
WHEN link_status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link_status=1 THEN jitter
ELSE 0 END) AS jitter, (CASE WHEN link_status=1 THEN packetloss ELSE 100 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
from (select itime, csf, devid, vd, interface, coalesce(healthcheck, name) as healthcheck,
(CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, latency::float as latency,
jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE WHEN
msg LIKE '%SLA failed%' THEN 1 WHEN status='down' 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
(inbandwidth) as inbandwidth, convert_unit_to_num(outbandwidth) as outbandwidth, convert_
unit_to_num(bibandwidth) as bibandwidth from $log where $filter and logid_to_int
(logid)=22925 AND status IS NOT NULL AND interface IS NOT NULL AND (healthcheck IS NOT NULL
OR name IS NOT NULL)) t) t group by timestamp, csf, devid, vd, interface, 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 10)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,
      sum(latency)/ sum(count) as latency
    from
      ###(select $flex_timestamp as timestamp, csf, devid, vd, interface, 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, min(sdwan_status) as sdwan_status from (select itime, csf, devid, vd, interface,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE 0 END) AS latency,
(CASE WHEN link_status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link_status=1 THEN
packetloss ELSE 100 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 from (select itime, csf, devid, vd, interface, coalesce
(healthcheck, name) as healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_
status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN msg LIKE '%SLA failed%' THEN 1 WHEN
status='down' 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(inbandwidth) as inbandwidth,
convert_unit_to_num(outbandwidth) as outbandwidth, convert_unit_to_num(bibandwidth) as
bibandwidth from $log where $filter and logid_to_int(logid)=22925 AND status IS NOT NULL AND
interface IS NOT NULL AND (healthcheck IS NOT NULL OR name IS NOT NULL)) t) t group by
timestamp, csf, devid, vd, interface, 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,
      sum(jitter)/ sum(count) as jitter
    from
      ###(select $flex_timestamp as timestamp, csf, devid, vd, interface, 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, min(sdwan_status) as sdwan_status from (select itime, csf, devid, vd, interface,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE 0 END) AS latency,
(CASE WHEN link_status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link_status=1 THEN
packetloss ELSE 100 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 from (select itime, csf, devid, vd, interface, coalesce
(healthcheck, name) as healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_
status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN msg LIKE '%SLA failed%' THEN 1 WHEN
status='down' 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(inbandwidth) as inbandwidth,
convert_unit_to_num(outbandwidth) as outbandwidth, convert_unit_to_num(bibandwidth) as
bibandwidth from $log where $filter and logid_to_int(logid)=22925 AND status IS NOT NULL AND
interface IS NOT NULL AND (healthcheck IS NOT NULL OR name IS NOT NULL)) t) t group by
timestamp, csf, devid, vd, interface, 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,
      sum(packetloss)/ sum(count) as packetloss
    from
      ###(select $flex_timestamp as timestamp, csf, devid, vd, interface, 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, min(sdwan_status) as sdwan_status from (select itime, csf, devid, vd, interface,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE 0 END) AS latency,
(CASE WHEN link_status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link_status=1 THEN
packetloss ELSE 100 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 from (select itime, csf, devid, vd, interface, coalesce
```

```
(healthcheck, name) as healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_
status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN msg LIKE '%SLA failed%' THEN 1 WHEN
status='down' 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(inbandwidth) as inbandwidth,
convert_unit_to_num(outbandwidth) as outbandwidth, convert_unit_to_num(bibandwidth) as
bibandwidth from $log where $filter and logid_to_int(logid)=22925 AND status IS NOT NULL AND
interface IS NOT NULL AND (healthcheck IS NOT NULL OR name IS NOT NULL)) t) t group by
timestamp, csf, devid, vd, interface, 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
  devid,
  interface,
  sum(bibandwidth)/ sum(count) as bibandwidth,
  cast(
    min(latency_min) as decimal(18, 2)
  ) as latency_min,
  cast(
    sum(latency)/ sum(count) 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(
    sum(jitter)/ sum(count) 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(
    sum(packetloss)/ sum(count) 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, devid, vd, interface, 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, min(sdwan_status) as sdwan_status from (select itime, csf, devid, vd, interface,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE 0 END) AS latency,
(CASE WHEN link_status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link_status=1 THEN
packetloss ELSE 100 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 from (select itime, csf, devid, vd, interface, coalesce
(healthcheck, name) as healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_
status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN msg LIKE '%SLA failed%' THEN 1 WHEN
status='down' 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(inbandwidth) as inbandwidth,
convert_unit_to_num(outbandwidth) as outbandwidth, convert_unit_to_num(bibandwidth) as
bibandwidth from $log where $filter and logid_to_int(logid)=22925 AND status IS NOT NULL AND
interface IS NOT NULL AND (healthcheck IS NOT NULL OR name IS NOT NULL)) t) t group by
timestamp, csf, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp
desc/*SkipEND*/)### t where $filter-drilldown and interface is not null group by devid,
interface having sum(count)>0 order by devid, interface
```

| Dataset Name | Description | Log Category |
|---|---|---|
| sdwan-Top-App-By-Bandwidth | Top SD-WAN application by bandwidth | traffic |

```
select
  appid,
  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
(vwlname,vwlservice) 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)### t where $filter-drilldown
group by appid, 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(vwlname,vwlservice) 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)###
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, devid, vd, interface, 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, min(sdwan_status) as sdwan_status from (select itime, csf, devid, vd, interface,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE 0 END) AS latency,
(CASE WHEN link_status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link_status=1 THEN
packetloss ELSE 100 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 from (select itime, csf, devid, vd, interface, coalesce
(healthcheck, name) as healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_
status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN msg LIKE '%SLA failed%' THEN 1 WHEN
status='down' 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(inbandwidth) as inbandwidth,
convert_unit_to_num(outbandwidth) as outbandwidth, convert_unit_to_num(bibandwidth) as
bibandwidth from $log where $filter and logid_to_int(logid)=22925 AND status IS NOT NULL AND
interface IS NOT NULL AND (healthcheck IS NOT NULL OR name IS NOT NULL)) t) t group by
timestamp, csf, devid, vd, interface, 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;Unknown') 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(vwlname,vwlservice)
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)### 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
  interface,
  sum(bandwidth) as bandwidth
from
  (
    (
      select
        srcintf 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
(vwlname,vwlservice) 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)### t where srcintfrole='wan'
and $filter-drilldown group by interface) union all (select 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(vwlname,vwlservice) 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)###
t where $filter-drilldown group by interface)) t group by interface order by bandwidth desc
limit 10
```

| Dataset Name | Description | Log Category |
|---|---|---|
| sdwan-Top-Application-Session-Bandwidth | Top SD-WAN application by bandwidth | traffic |

```
select
  appid,
  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
(vwlname,vwlservice) 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)### t where $filter-drilldown
group by appid, 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
(vwlname,vwlservice) 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)### 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
(vwlname,vwlservice) 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)### 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
(vwlname,vwlservice) 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)### 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(vwlname,vwlservice) 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)###
t where $filter-drilldown group by dstintf order by num_intf desc limit 10)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
(vwlname,vwlservice) 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)### 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(vwlname,vwlservice) 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)###
t where $filter-drilldown and srcintf is not null and srcintfrole ='wan' group by srcintf
order by num_intf desc limit 10)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 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
(vwlname,vwlservice) 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)### 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(vwlname,vwlservice) 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)###
t where $filter-drilldown group by dstintf order by num_intf desc limit 10)t2 on
t1.dstintf=t2.dstintf group by hodex, t1.dstintf 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, devid, vd, interface, 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, min(sdwan_status) as sdwan_status from (select itime, csf, devid, vd, interface,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE 0 END) AS latency,
(CASE WHEN link_status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link_status=1 THEN
packetloss ELSE 100 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 from (select itime, csf, devid, vd, interface, coalesce
(healthcheck, name) as healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_
status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN msg LIKE '%SLA failed%' THEN 1 WHEN
status='down' 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(inbandwidth) as inbandwidth,
convert_unit_to_num(outbandwidth) as outbandwidth, convert_unit_to_num(bibandwidth) as
bibandwidth from $log where $filter and logid_to_int(logid)=22925 AND status IS NOT NULL AND
interface IS NOT NULL AND (healthcheck IS NOT NULL OR name IS NOT NULL)) t) t group by
timestamp, csf, devid, vd, interface, 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,
  sum(latency)/ sum(count) as latency
from
  (
    select
      timestamp,
      interface || & #039;:' || sla_rule as intf_sla, sum(latency) as latency, sum(count) as
count from ###(select $flex_timestamp as timestamp, csf, devid, vd, interface, 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, min(sdwan_status) as sdwan_status from (select itime, csf, devid, vd, interface,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE 0 END) AS latency,
(CASE WHEN link_status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link_status=1 THEN
packetloss ELSE 100 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 from (select itime, csf, devid, vd, interface, coalesce
```

```
(healthcheck, name) as healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_
status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN msg LIKE '%SLA failed%' THEN 1 WHEN
status='down' 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(inbandwidth) as inbandwidth,
convert_unit_to_num(outbandwidth) as outbandwidth, convert_unit_to_num(bibandwidth) as
bibandwidth from $log where $filter and logid_to_int(logid)=22925 AND status IS NOT NULL AND
interface IS NOT NULL AND (healthcheck IS NOT NULL OR name IS NOT NULL)) t) t group by
timestamp, csf, devid, vd, interface, 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, devid, vd, interface, 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, min(sdwan_status) as sdwan_status from (select itime, csf, devid, vd, interface,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE 0 END) AS latency,
(CASE WHEN link_status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link_status=1 THEN
packetloss ELSE 100 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 from (select itime, csf, devid, vd, interface, coalesce
(healthcheck, name) as healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_
status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN msg LIKE '%SLA failed%' THEN 1 WHEN
status='down' 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(inbandwidth) as inbandwidth,
convert_unit_to_num(outbandwidth) as outbandwidth, convert_unit_to_num(bibandwidth) as
bibandwidth from $log where $filter and logid_to_int(logid)=22925 AND status IS NOT NULL AND
interface IS NOT NULL AND (healthcheck IS NOT NULL OR name IS NOT NULL)) t) t group by
timestamp, csf, devid, vd, interface, 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 10)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,
  sum(jitter)/ sum(count) as jitter
from
  (
    select
      timestamp,
      interface || & #039;:' || sla_rule as intf_sla, sum(jitter) as jitter, sum(count) as
count from ###(select $flex_timestamp as timestamp, csf, devid, vd, interface, 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, min(sdwan_status) as sdwan_status from (select itime, csf, devid, vd, interface,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE 0 END) AS latency,
(CASE WHEN link_status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link_status=1 THEN
packetloss ELSE 100 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 from (select itime, csf, devid, vd, interface, coalesce
(healthcheck, name) as healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_
status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN msg LIKE '%SLA failed%' THEN 1 WHEN
status='down' 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(inbandwidth) as inbandwidth,
convert_unit_to_num(outbandwidth) as outbandwidth, convert_unit_to_num(bibandwidth) as
bibandwidth from $log where $filter and logid_to_int(logid)=22925 AND status IS NOT NULL AND
interface IS NOT NULL AND (healthcheck IS NOT NULL OR name IS NOT NULL)) t) t group by
timestamp, csf, devid, vd, interface, 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, devid, vd, interface, 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, min(sdwan_status) as sdwan_status from (select itime, csf, devid, vd, interface,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE 0 END) AS latency,
(CASE WHEN link_status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link_status=1 THEN
packetloss ELSE 100 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 from (select itime, csf, devid, vd, interface, coalesce
(healthcheck, name) as healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_
status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN msg LIKE '%SLA failed%' THEN 1 WHEN
status='down' 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(inbandwidth) as inbandwidth,
convert_unit_to_num(outbandwidth) as outbandwidth, convert_unit_to_num(bibandwidth) as
bibandwidth from $log where $filter and logid_to_int(logid)=22925 AND status IS NOT NULL AND
interface IS NOT NULL AND (healthcheck IS NOT NULL OR name IS NOT NULL)) t) t group by
timestamp, csf, devid, vd, interface, 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 10)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,
  sum(packetloss)/ sum(count) as packetloss
from
  (
    select
      timestamp,
      interface || & #039;:' || sla_rule as intf_sla, sum(packetloss) as packetloss, sum
(count) as count from ###(select $flex_timestamp as timestamp, csf, devid, vd, interface,
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, min(sdwan_status) as sdwan_status from (select itime, csf,
devid, vd, interface, healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE 0
END) AS latency, (CASE WHEN link_status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN
link_status=1 THEN packetloss ELSE 100 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 from (select itime, csf,
devid, vd, interface, coalesce(healthcheck, name) as healthcheck, (CASE WHEN status='down'
THEN 0 ELSE 1 END) AS link_status, latency::float as latency, jitter::float as jitter, trim
(trailing '%' from packetloss)::float as packetloss, (CASE WHEN msg LIKE '%SLA failed%' THEN
1 WHEN status='down' 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(inbandwidth) as
inbandwidth, convert_unit_to_num(outbandwidth) as outbandwidth, convert_unit_to_num
(bibandwidth) as bibandwidth from $log where $filter and logid_to_int(logid)=22925 AND
status IS NOT NULL AND interface IS NOT NULL AND (healthcheck IS NOT NULL OR name IS NOT
NULL)) t) t group by timestamp, csf, devid, vd, interface, 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, devid, vd, interface,
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, min(sdwan_status) as sdwan_status from (select itime, csf,
devid, vd, interface, healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE 0
END) AS latency, (CASE WHEN link_status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN
link_status=1 THEN packetloss ELSE 100 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 from (select itime, csf,
devid, vd, interface, coalesce(healthcheck, name) as healthcheck, (CASE WHEN status='down'
THEN 0 ELSE 1 END) AS link_status, latency::float as latency, jitter::float as jitter, trim
(trailing '%' from packetloss)::float as packetloss, (CASE WHEN msg LIKE '%SLA failed%' THEN
1 WHEN status='down' 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(inbandwidth) as
inbandwidth, convert_unit_to_num(outbandwidth) as outbandwidth, convert_unit_to_num
(bibandwidth) as bibandwidth from $log where $filter and logid_to_int(logid)=22925 AND
status IS NOT NULL AND interface IS NOT NULL AND (healthcheck IS NOT NULL OR name IS NOT
NULL)) t) t group by timestamp, csf, devid, vd, interface, 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 10)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)
    ) as decimal(18, 2)
  ) as latency
from
  ###(select $flex_timestamp as timestamp, csf, devid, vd, interface, 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, min(sdwan_status) as sdwan_status from (select itime, csf, devid, vd, interface,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE 0 END) AS latency,
(CASE WHEN link_status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link_status=1 THEN
packetloss ELSE 100 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 from (select itime, csf, devid, vd, interface, coalesce
(healthcheck, name) as healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_
status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN msg LIKE '%SLA failed%' THEN 1 WHEN
status='down' 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(inbandwidth) as inbandwidth,
convert_unit_to_num(outbandwidth) as outbandwidth, convert_unit_to_num(bibandwidth) as
bibandwidth from $log where $filter and logid_to_int(logid)=22925 AND status IS NOT NULL AND
interface IS NOT NULL AND (healthcheck IS NOT NULL OR name IS NOT NULL)) t) t group by
timestamp, csf, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp
```

```
desc/*SkipEND*/)### t where $filter-drilldown group by sla_rule, interface having sum
(count)>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)
    ) as decimal(18, 2)
  ) as jitter
from
  ###(select $flex_timestamp as timestamp, csf, devid, vd, interface, 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, min(sdwan_status) as sdwan_status from (select itime, csf, devid, vd, interface,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE 0 END) AS latency,
(CASE WHEN link_status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link_status=1 THEN
packetloss ELSE 100 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 from (select itime, csf, devid, vd, interface, coalesce
(healthcheck, name) as healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_
status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN msg LIKE '%SLA failed%' THEN 1 WHEN
status='down' 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(inbandwidth) as inbandwidth,
convert_unit_to_num(outbandwidth) as outbandwidth, convert_unit_to_num(bibandwidth) as
bibandwidth from $log where $filter and logid_to_int(logid)=22925 AND status IS NOT NULL AND
interface IS NOT NULL AND (healthcheck IS NOT NULL OR name IS NOT NULL)) t) t group by
timestamp, csf, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp
desc/*SkipEND*/)### t where $filter-drilldown group by sla_rule, interface having sum
(count)>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)
    ) as decimal(18, 2)
  ) as packetloss
from
  ###(select $flex_timestamp as timestamp, csf, devid, vd, interface, 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, min(sdwan_status) as sdwan_status from (select itime, csf, devid, vd, interface,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE 0 END) AS latency,
(CASE WHEN link_status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link_status=1 THEN
packetloss ELSE 100 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 from (select itime, csf, devid, vd, interface, coalesce
(healthcheck, name) as healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_
status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN msg LIKE '%SLA failed%' THEN 1 WHEN
status='down' 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(inbandwidth) as inbandwidth,
convert_unit_to_num(outbandwidth) as outbandwidth, convert_unit_to_num(bibandwidth) as
bibandwidth from $log where $filter and logid_to_int(logid)=22925 AND status IS NOT NULL AND
interface IS NOT NULL AND (healthcheck IS NOT NULL OR name IS NOT NULL)) t) t group by
timestamp, csf, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp
desc/*SkipEND*/)### t where $filter-drilldown group by sla_rule, interface having sum
(count)>0 order by packetloss desc
```

| 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, devid, vd, interface, 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, min(sdwan_status) as sdwan_status from (select itime, csf, devid, vd, interface,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE 0 END) AS latency,
(CASE WHEN link_status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link_status=1 THEN
packetloss ELSE 100 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 from (select itime, csf, devid, vd, interface, coalesce
(healthcheck, name) as healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_
status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN msg LIKE '%SLA failed%' THEN 1 WHEN
status='down' 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(inbandwidth) as inbandwidth,
convert_unit_to_num(outbandwidth) as outbandwidth, convert_unit_to_num(bibandwidth) as
bibandwidth from $log where $filter and logid_to_int(logid)=22925 AND status IS NOT NULL AND
interface IS NOT NULL AND (healthcheck IS NOT NULL OR name IS NOT NULL)) t) t group by
timestamp, csf, devid, vd, interface, 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, devid, vd, interface, 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, min(sdwan_status) as sdwan_status from (select itime, csf, devid, vd, interface,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE 0 END) AS latency,
(CASE WHEN link_status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link_status=1 THEN
packetloss ELSE 100 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 from (select itime, csf, devid, vd, interface, coalesce
(healthcheck, name) as healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_
status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN msg LIKE '%SLA failed%' THEN 1 WHEN
status='down' 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(inbandwidth) as inbandwidth,
convert_unit_to_num(outbandwidth) as outbandwidth, convert_unit_to_num(bibandwidth) as
bibandwidth from $log where $filter and logid_to_int(logid)=22925 AND status IS NOT NULL AND
interface IS NOT NULL AND (healthcheck IS NOT NULL OR name IS NOT NULL)) t) t group by
timestamp, csf, devid, vd, interface, 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, devid, vd, interface, 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, min(sdwan_status) as
sdwan_status from (select itime, csf, devid, vd, interface, healthcheck, link_status, (CASE
WHEN link_status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link_status=1 THEN jitter
ELSE 0 END) AS jitter, (CASE WHEN link_status=1 THEN packetloss ELSE 100 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
from (select itime, csf, devid, vd, interface, coalesce(healthcheck, name) as healthcheck,
(CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, latency::float as latency,
jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE WHEN
msg LIKE '%SLA failed%' THEN 1 WHEN status='down' 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
(inbandwidth) as inbandwidth, convert_unit_to_num(outbandwidth) as outbandwidth, convert_
unit_to_num(bibandwidth) as bibandwidth from $log where $filter and logid_to_int
(logid)=22925 AND status IS NOT NULL AND interface IS NOT NULL AND (healthcheck IS NOT NULL
OR name IS NOT NULL)) t) t group by timestamp, csf, devid, vd, interface, 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, devid, vd, interface,
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, min(sdwan_status) as sdwan_status from (select itime, csf,
devid, vd, interface, healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE 0
END) AS latency, (CASE WHEN link_status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN
```

```
link_status=1 THEN packetloss ELSE 100 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 from (select itime, csf,
devid, vd, interface, coalesce(healthcheck, name) as healthcheck, (CASE WHEN status='down'
THEN 0 ELSE 1 END) AS link_status, latency::float as latency, jitter::float as jitter, trim
(trailing '%' from packetloss)::float as packetloss, (CASE WHEN msg LIKE '%SLA failed%' THEN
1 WHEN status='down' 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(inbandwidth) as
inbandwidth, convert_unit_to_num(outbandwidth) as outbandwidth, convert_unit_to_num
(bibandwidth) as bibandwidth from $log where $filter and logid_to_int(logid)=22925 AND
status IS NOT NULL AND interface IS NOT NULL AND (healthcheck IS NOT NULL OR name IS NOT
NULL)) t) t group by timestamp, csf, devid, vd, interface, 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, devid, vd, interface, 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, min(sdwan_status) as
sdwan_status from (select itime, csf, devid, vd, interface, healthcheck, link_status, (CASE
WHEN link_status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link_status=1 THEN jitter
ELSE 0 END) AS jitter, (CASE WHEN link_status=1 THEN packetloss ELSE 100 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
from (select itime, csf, devid, vd, interface, coalesce(healthcheck, name) as healthcheck,
(CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, latency::float as latency,
jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE WHEN
msg LIKE '%SLA failed%' THEN 1 WHEN status='down' 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
(inbandwidth) as inbandwidth, convert_unit_to_num(outbandwidth) as outbandwidth, convert_
unit_to_num(bibandwidth) as bibandwidth from $log where $filter and logid_to_int
(logid)=22925 AND status IS NOT NULL AND interface IS NOT NULL AND (healthcheck IS NOT NULL
OR name IS NOT NULL)) t) t group by timestamp, csf, devid, vd, interface, 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, devid, vd, interface, 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, min(sdwan_status) as sdwan_status from (select itime, csf, devid, vd, interface,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE 0 END) AS latency,
(CASE WHEN link_status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link_status=1 THEN
packetloss ELSE 100 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 from (select itime, csf, devid, vd, interface, coalesce
(healthcheck, name) as healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_
status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN msg LIKE '%SLA failed%' THEN 1 WHEN
status='down' 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(inbandwidth) as inbandwidth,
convert_unit_to_num(outbandwidth) as outbandwidth, convert_unit_to_num(bibandwidth) as
bibandwidth from $log where $filter and logid_to_int(logid)=22925 AND status IS NOT NULL AND
interface IS NOT NULL AND (healthcheck IS NOT NULL OR name IS NOT NULL)) t) t group by
timestamp, csf, devid, vd, interface, 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, 'Unknown') 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(vwlname,vwlservice)
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)### 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-test2 | SD-WAN Device-Interface Statistic | event |

```
select
  devid,
  sum(bibandwidth)/ sum(count) as bibandwidth
```

```
from
  ###(select $flex_timestamp as timestamp, csf, devid, vd, interface, 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, min(sdwan_status) as sdwan_status from (select itime, csf, devid, vd, interface,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE 0 END) AS latency,
(CASE WHEN link_status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link_status=1 THEN
packetloss ELSE 100 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 from (select itime, csf, devid, vd, interface, coalesce
(healthcheck, name) as healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_
status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN msg LIKE '%SLA failed%' THEN 1 WHEN
status='down' 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(inbandwidth) as inbandwidth,
convert_unit_to_num(outbandwidth) as outbandwidth, convert_unit_to_num(bibandwidth) as
bibandwidth from $log where $filter and logid_to_int(logid)=22925 AND status IS NOT NULL AND
interface IS NOT NULL AND (healthcheck IS NOT NULL OR name IS NOT NULL)) t) t group by
timestamp, csf, devid, vd, interface, 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-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, devid, vd, interface, 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, min(sdwan_status) as sdwan_status from (select itime, csf, devid, vd, interface,
healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE 0 END) AS latency,
(CASE WHEN link_status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link_status=1 THEN
packetloss ELSE 100 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 from (select itime, csf, devid, vd, interface, coalesce
(healthcheck, name) as healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_
status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN msg LIKE '%SLA failed%' THEN 1 WHEN
status='down' 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(inbandwidth) as inbandwidth,
convert_unit_to_num(outbandwidth) as outbandwidth, convert_unit_to_num(bibandwidth) as
bibandwidth from $log where $filter and logid_to_int(logid)=22925 AND status IS NOT NULL AND
interface IS NOT NULL AND (healthcheck IS NOT NULL OR name IS NOT NULL)) t) t group by
timestamp, csf, devid, vd, interface, 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, devid, vd, interface, 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, min
(sdwan_status) as sdwan_status from (select itime, csf, devid, vd, interface, healthcheck,
link_status, (CASE WHEN link_status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link_
status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link_status=1 THEN packetloss ELSE
100 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 from (select itime, csf, devid, vd, interface, coalesce(healthcheck,
name) as healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status,
latency::float as latency, jitter::float as jitter, trim(trailing '%' from
packetloss)::float as packetloss, (CASE WHEN msg LIKE '%SLA failed%' THEN 1 WHEN
status='down' 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(inbandwidth) as inbandwidth,
convert_unit_to_num(outbandwidth) as outbandwidth, convert_unit_to_num(bibandwidth) as
bibandwidth from $log where $filter and logid_to_int(logid)=22925 AND status IS NOT NULL AND
interface IS NOT NULL AND (healthcheck IS NOT NULL OR name IS NOT NULL)) t) t group by
timestamp, csf, devid, vd, interface, 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 |
|---|---|---|
| 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 dvid, srcip, dstip, epid, euid,
coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, 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 dvid, srcip, dstip, epid, euid, user_src, appid, app, appcat, apprisk,
hostname order by 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, 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 dvid, srcip, dstip, epid, euid, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src, 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 dvid, srcip,
dstip, epid, euid, user_src, appid, app, appcat, apprisk, hostname order by sessions
desc)base### t group by appid, app, appcat, apprisk /*SkipSTART*/order by sessions desc,
bandwidth 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, coalesce(nullifna
(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service, count(*) 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, user_src, service /*SkipSTART*/order by timestamp
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 and (eventtype is null or logver>=502000000) 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 (eventtype is null or logver>=502000000) 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
  interface,
```

```
  bandwidth
from
  (
    (
      select
        & #039;Internal' 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
(vwlname,vwlservice) 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)### t where $filter-drilldown
and dstintfrole='lan') union all (select 'External' 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(vwlname,vwlservice) 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)### t where $filter-drilldown
and dstintfrole='wan')) t where bandwidth>0
```

| 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(vwlname,vwlservice) 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)###
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
(vwlname,vwlservice) 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)### 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
(vwlname,vwlservice) 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)### 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
(vwlname,vwlservice) 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)### 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
(vwlname,vwlservice) 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)### 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
(vwlname,vwlservice) 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)### 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
(vwlname,vwlservice) 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)### 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
(vwlname,vwlservice) 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)### 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
(vwlname,vwlservice) 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)### 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
(vwlname,vwlservice) 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)### 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)### 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)### 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)### t group by hourstamp order by hourstamp
```

| Dataset Name | Description | Log Category |
|---|---|---|
| Top-Destination-Addresses-By-Bandwidth-Bar | 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 |
|---|---|---|
| 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
        (timestamp / 300 * 300) as tm,
        sum(rcvdbps) as rcvdbps,
        300 as interval
      from
        intfstats_billing tb1
        join (
          select
            ti.dvid,
            intfname
          from
            intfinfo ti
            left join devtable td on ti.dvid = td.dvid
          where
            $dev_filter
        ) tb2 on tb1.dvid = tb2.dvid
        and tb1.intfname = tb2.intfname
      where
        $cust_time_filter(timestamp)
      group by
        tm
    ) tmp
),
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
              (timestamp / 300 * 300) as tm,
              sum(rcvdbps) as rcvdbps,
              300 as interval
            from
              intfstats_billing tb1
              join (
```

```
                select
                  ti.dvid,
                  intfname
                from
                  intfinfo ti
                  left join devtable td on ti.dvid = td.dvid
                where
                  $dev_filter
              ) tb2 on tb1.dvid = tb2.dvid
              and tb1.intfname = tb2.intfname
            where
              $cust_time_filter(timestamp)
            group by
              tm
          ) tmp
      ) 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
        (timestamp / 300 * 300) as tm,
        sum(rcvdbps) as rcvdbps,
        300 as interval
      from
        intfstats_billing tb1
        join (
          select
            ti.dvid,
            intfname
          from
            intfinfo ti
            left join devtable td on ti.dvid = td.dvid
          where
            $dev_filter
        ) tb2 on tb1.dvid = tb2.dvid
        and tb1.intfname = tb2.intfname
      where
        $cust_time_filter(timestamp)
      group by
```

```
        tm
    ) tmp
),
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,
    interval,
    ntile(100) over (
      order by
        rcvdbps
    ) as percentile
  from
```

```
      (
        select
          (timestamp / 300 * 300) as tm,
          sum(rcvdbps) as rcvdbps,
          300 as interval
        from
          intfstats_billing tb1
          join (
            select
              ti.dvid,
              intfname
            from
              intfinfo ti
              left join devtable td on ti.dvid = td.dvid
            where
              $dev_filter
          ) tb2 on tb1.dvid = tb2.dvid
          and tb1.intfname = tb2.intfname
        where
          $cust_time_filter(timestamp)
        group by
          tm
      ) tmp
  )
  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(interval * rcvdbps)/ 8 /(1024 * 1024 * 1024) as decimal(18, 2)
      ) as actual_gb,
      count(*) as total
    from
      base_qry
  ) t
```

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

# 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 |
| Highest Risk Application with Highest Session Count | Highest risk application with the highest session count | High-Risk-Application-By-Sessions | Traffic |

| Macro Name | Description | Dataset Used | Log Category |
|---|---|---|---|
| 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 |
| 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 |

| Macro Name | Description | Dataset Used | Log Category |
|---|---|---|---|
| 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 |

# Change Log

| Date | Change Description |
| --- | --- |
| 2022-08-31 | Initial release. |
| | |