FortiAnalyzer - Dataset Reference

Version 6.0.7

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/support-and-training/training.html

NSE INSTITUTE
https://training.fortinet.com

FORTIGUARD CENTER
https://fortiguard.com/

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

FEEDBACK
Email: techdoc@fortinet.com

November 13, 2019
FortiAnalyzer 6.0.7 Dataset Reference
05-606-484314-20191113

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
  ###({{FGT_DATASET_BASE_TRAFFIC_BANDWIDTH_SESSION}})### t 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
  ###({{FGT_DATASET_BASE_TRAFFIC_BANDWIDTH_SESSION}})### t group by hodex order by hodex
```

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

```
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,
  count(*) as sessions
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
group by
  user_src
having
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  )& gt; 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(
    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)
  )& gt; 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)
  )& gt; 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 concat(inter-
face, '.', devid) as intf, mac from $log where $last3day_period $filter  and logid_to_int
(logid) = 26001 and dhcp_msg = 'Ack' group by interface, devid, mac)###; create temporary
table rpt_tmptbl_2 as ###(select concat(interface, '.', devid) as intf, mac from $log where
$filter and logid_to_int(logid) = 26001 and dhcp_msg = 'Ack' group by interface, devid,
mac)###; create temporary table rpt_tmptbl_3 as select distinct on (1) intf, cast
(used*100.0/total as decimal(18,2)) as percent_of_allocated_ip from ###(select distinct on (1)
concat(interface, '.', devid) as intf, used, total, itime from $log where $filter and logid_
to_int(logid)=26003 and total>0 /*SkipSTART*/order by intf, itime desc/*SkipEND*/)### t order
by intf, itime desc; select t1.intf as interface, percent_of_allocated_ip, new_cli_count from
rpt_tmptbl_3 t1 inner join (select intf, 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 intf)
t2 on t1.intf=t2.intf 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
  coalesce(
    nullifna(`user`),
    nullifna(`unauthuser`),
    ipstr(`srcip`)
  ) as user_src,
  srcssid,
  devtype,
  coalesce(
    nullifna(`srcname`),
    `srcmac`
  ) as hostname_mac,
  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
  )
group by
  user_src,
  srcssid,
  devtype,
  hostname_mac
having
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  )& gt; 0
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
  ###({{FGT_DATASET_BASE_TRAFFIC_BANDWIDTH_SESSION}})### t 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 (
    '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, ', ') 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<52) 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 (
    '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,
  devtype,
  srcname,
  count(*) as requests
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
  and utmevent in (
    'webfilter', 'banned-word', 'web-content',
    'command-block', 'script-filter'
  )
group by
```

```
  user_src,
  devtype,
  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 (
    '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)
  )& gt; 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,
  devtype,
  srcname,
  count(*) as requests
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
  and utmevent in (
    'webfilter', 'banned-word', 'web-content',
    'command-block', 'script-filter'
  )
  and (
    utmaction in ('block', 'blocked')
    or action = 'deny'
  )
group by
  user_src,
  devtype,
  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 (log-
flag&1>0) and (countweb>0 or ((logver is null or logver<52) 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,
  devtype,
  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 (
    'webfilter', 'banned-word', 'web-content',
    'command-block', 'script-filter'
  )
group by
  user_src,
  devtype,
  srcname
having
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  )& gt; 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 ('Streaming Media and Download')
group by
```

```
  appid,
  hostname
having
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  )& gt; 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 (
    '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 (
    '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 (
    'smtp', 'SMTP', '25/tcp', '587/tcp',
    'smtps', 'SMTPS', '465/tcp'
  )
group by
  user_src
having
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  )& gt; 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 (
    '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)
  )& gt; 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 '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>=52) and nullifna(virus) is not null group
by virus, virusid_s /*SkipSTART*/order by totalnum desc/*SkipEND*/)### t group by virus, mal-
ware_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 '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>=52) and nullifna(virus) is not null group
by virus, virusid_s /*SkipSTART*/order by totalnum desc/*SkipEND*/)### t group by virus, mal-
ware_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>=52) 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
      ###({{FGT_DATASET_BASE_EVENT_VPN_IPSEC_TUNNEL_BANDWIDTH}})### 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
      ###({{FGT_DATASET_BASE_EVENT_SSL_VPN_TUNNEL_USERS}})### 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
      ###({{FGT_DATASET_BASE_EVENT_VPN_DIAL_UP_IPSEC_TUNNELS}})### 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, ' ') 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
      ###({{FGT_DATASET_BASE_EVENT_VPN_DIAL_UP_IPSEC_BANDWIDTH}})### 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, ' ') 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
      ###({{FGT_DATASET_BASE_EVENT_VPN_DIAL_UP_IPSEC_BANDWIDTH}})### 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
      ###({{FGT_DATASET_BASE_EVENT_SSL_VPN_TUNNEL_USERS}})### 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
      ###({{FGT_DATASET_BASE_EVENT_SSL_VPN_TUNNEL_USERS}})### t where tunneltype='ssl-web'
group by devid, vd, user_src, remip, tunnelid) tt group by user_src, remote_ip order by dur-
ation 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
      ###({{FGT_DATASET_BASE_EVENT_SSL_VPN_TUNNEL_USERS}})### 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, ' ') 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
      ###({{FGT_DATASET_BASE_EVENT_VPN_DIAL_UP_IPSEC_USERS}})### 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 dur-
ation desc
```

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

```
select
  $flex_timescale(timestamp) as hodex,
  sum(total_num) as total_num
from
  (
    select
      timestamp,
      devid,
      vd,
      remip,
      tunnelid,
      sum(tunnelup) as total_num,
      max(traffic_in) as traffic_in,
      max(traffic_out) as traffic_out
    from
      ###(select $flex_timestamp as timestamp, devid, vd, remip, tunnelid, (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 (tun-
neltype like 'ipsec%' or tunneltype like 'ssl%') and action in ('tunnel-up', 'tunnel-stats',
'tunnel-down') and tunnelid is not null group by timestamp, action, 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(`xau-
thuser`), nullifna(`user`)) is not null group by f_user, tunneltype)### t group by f_user, tun-
neltype 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, ' ') 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
      ###({{FGT_DATASET_BASE_EVENT_VPN_DIAL_UP_IPSEC_USERS}})### 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 '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
      ###({{FGT_DATASET_BASE_EVENT_VPN_TRAFFIC_USAGE}})### 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, vpn-
tunnel, devid, vd /*SkipSTART*/order by tunnelid/*SkipEND*/)### t group by vpntunnel, tun-
neltype, 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 tun-
nelid!=0 group by tunnelid, user_src, remip, devid, vd /*SkipSTART*/order by tun-
nelid/*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(dur-
ation, 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 tun-
nelid, 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(dur-
ation, 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)& gt; 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 con-
fig_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
  ###({{FGT_DATASET_BASE_EVENT_SYSTEM_EVENTS}})### 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', 'emer-
gency') 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', 'emer-
gency') 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
  ###({{FGT_DATASET_BASE_EVENT_SYSTEM_EVENTS}})### 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
  ###({{FGT_DATASET_BASE_EVENT_SYSTEM_EVENTS}})### 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
  ###({{FGT_DATASET_BASE_EVENT_SYSTEM_EVENTS}})### 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(rcvd-
byte, 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
  ###({{FGT_DATASET_BASE_TRAFFIC_TOP_EMAIL_SENDERS}})### 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
  ###({{FGT_DATASET_BASE_TRAFFIC_TOP_EMAIL_RECIPIENTS}})### 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
  ###({{FGT_DATASET_BASE_TRAFFIC_TOP_EMAIL_RECIPIENTS}})### 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
  ###({{FGT_DATASET_BASE_TRAFFIC_TOP_EMAIL_SENDERS}})### 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<52) and (hostname is not null
or utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-fil-
ter')))) and hostname is not null group by user_src, appid, hostname, blocked order by band-
width 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>=52) 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 total-
num from $log where $filter and (eventtype is null or logver>=52) 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 total-
num 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
  ###({{FGT_DATASET_BASE_TRAFFIC_TOP_APPS}})### 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
  ###({{FGT_DATASET_BASE_TRAFFIC_TOP_APPS}})### 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)
  )& gt; 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(
    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)
  )& gt; 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(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  ) as bandwidth
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
  and nullifna(appcat) is not null
group by
  appcat
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
  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,
  count(*) as sessions
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
group by
  user_src
having
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  )& gt; 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>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(
    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
  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(
    cast(poluuid as text),
    cast(policyid as text)
  ) as polid,
  sum(
    coalesce(rcvdbyte, 0) + coalesce(sentbyte, 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
  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 count(*) as sessions, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as band-
width from $log where $filter and (logflag&1>0))### t; update rpt_tmptbl_1 set active_date-
e=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>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 total-
num from ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as
user_src, count(*) as count from $log where $filter and (logflag&1>0) 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, count(*) as count from $log where $filter and (logflag&1>0) and nullifna(app) is
not null group by app_grp order by count desc)### t) as t3; update rpt_tmptbl_1 set total_dest-
t=t4.totalnum from (select format_numeric_no_decimal(count(distinct(dstip))) as totalnum from
###(select dstip, count(*) as count from $log where $filter and (logflag&1>0) and 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 sum-
mary, total_bandwidth as stats from rpt_tmptbl_1 union all select 'Most Active Date By Ses-
sions' 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 sum-
mary, 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
  ###({{FGT_DATASET_BASE_TRAFFIC_CLIENT_REPUTATION_INCIDENTS}})### 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
  ###({{FGT_DATASET_BASE_TRAFFIC_CLIENT_REPUTATION_INCIDENTS}})### 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)& gt; 0
```

```
order by
  scores desc
```

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

```
select
  devtype,
  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
  devtype,
  dev_src
having
  sum(crscore % 65536)& gt; 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 coalesce(nul-
lifna(`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)###; create temporary table
rpt_tmptbl_2 as ###(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)###; 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 coalesce(nul-
lifna(`srcname`),nullifna(`srcmac`), ipstr(`srcip`)) as f_device, devtype, 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 having sum(crscore%65536)>0 order by sum_rp_score desc)###; create
temporary table rpt_tmptbl_2 as ###(select coalesce(nullifna(`srcname`),nullifna(`srcmac`),
ipstr(`srcip`)) as f_device, devtype, sum(crscore%65536) as sum_rp_score from $log where $fil-
ter and (logflag&1>0) and crscore is not null group by f_device, devtype having sum
(crscore%65536)>0 order by sum_rp_score desc)###; select t1.f_device, t1.devtype , 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=t2.devtype where t2.sum_rp_score > t1.sum_rp_score group by
t1.f_device, t1.devtype order by delta desc
```

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

```
select
  (
    case when severity = '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 ('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>=52) 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>=52) 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
  ###({{FGT_DATASET_BASE_VIRUS_TOP_MALWARE_VICTIMS}})### 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
  ###({{FGT_DATABASE_VIRUS_TOP_MALWARE_NAME}})### 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 '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
  ###({{FGT_DATASET_BASE_VIRUS_TOP_MALWARE_VICTIMS}})### 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
  ###({{FGT_DATABASE_VIRUS_TOP_MALWARE_NAME}})### 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 '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 = 'notice' 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 = 'notice' 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 = '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 = '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 = '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 sever-
ity='medium' then 1 else 0 end) as med_num from $log where $filter and severity in ('crit-
ical', '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 = '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 = '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 = '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 '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 '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 '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 '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 '%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 '%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 '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 '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 '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 '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,
  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;
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
  '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
  ###({{FGT_DATASET_BASE_EVENT_WIRELESS_ROGUE_OFFWIRE}})### 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
  '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
  ###({{FGT_DATASET_BASE_EVENT_WIRELESS_ROGUE_ONWIRE}})### 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
  '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
  ###({{FGT_DATASET_BASE_EVENT_WIRELESS_ROGUE_OFFWIRE}})### 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
  '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
  ###({{FGT_DATASET_BASE_EVENT_WIRELESS_ROGUE_ONWIRE}})### 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
  '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
  ###({{FGT_DATASET_BASE_EVENT_WIRELESS_ROGUE_OFFWIRE}})### 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
  '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
  ###({{FGT_DATASET_BASE_EVENT_WIRELESS_ROGUE_ONWIRE}})### 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
  '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
  ###({{FGT_DATASET_BASE_EVENT_WIRELESS_ROGUE_OFFWIRE}})### 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
  '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
  ###({{FGT_DATASET_BASE_EVENT_WIRELESS_ROGUE_ONWIRE}})### 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, ' ') 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
      ###({{FGT_DATASET_BASE_EVENT_VPN_DIAL_UP_IPSEC_BANDWIDTH}})### 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
      ###({{FGT_DATASET_BASE_EVENT_VPN_TRAFFIC_USAGE}})### 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>=52) 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
  ###({{FGT_DATABASE_TRAFFIC_BROWSE_TIME}})### t group by hodex order by hodex
```

Dataset Name	Description	Log Category
traffic-Browsing-Time-Summary-Enhanced	Traffic browsing time summary enhanced	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
  ###({{FGT_DATABASE_TRAFFIC_BROWSE_TIME}})### 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>=52) 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>=52) 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(brow-
setime), 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>=52) 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, ', ') 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>=52) 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(sent-
byte, 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<52) and (hostname is not null or utmevent in ('web-
filter', '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>=52) 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>=52) 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, ', ') 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
  ###({{FGT_DATASET_TRAFFIC_TOP_SITES_BY_EB_TIME}})### t group by hostname order by browsetime
desc
```

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

```
select
  hostname,
  string_agg(distinct catdesc, ', ') 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
```

```
  ###({{FGT_DATASET_TRAFFIC_TOP_SITES_BY_EB_TIME}})### 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
  ###({{FGT_DATASET_TRAFFIC_TOP_CATS_BY_EB_TIME}})### t group by catdesc order by browsetime
desc
```

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

```
select
  catdesc,
  ebtr_value(
    ebtr_agg_flat(browsetime),
    null,
    $timespan
  ) as browsetime,
  sum(bandwidth) as bandwidth
from
  ###({{FGT_DATASET_TRAFFIC_TOP_CATS_BY_EB_TIME}})### 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
  ###({{FGT_DATASET_TRAFFIC_TOP_DST_COUNTRY_BY_EB_TIME}})### t group by dstcountry order by
browsetime desc
```

Dataset Name	Description	Log Category
traffic-Top-Destination-Countries-By-Browsing-Time-Enhanced	Traffic top destination countries by browsing time enhanced	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
  ###({{FGT_DATASET_TRAFFIC_TOP_DST_COUNTRY_BY_EB_TIME}})### 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
  ###({{FGT_DATASET_TRAFFIC_EB_TIME}})### t group by user_src order by browsetime desc
```

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

```
select
  user_src,
  ebtr_value(
    ebtr_agg_flat(browsetime),
    null,
    $timespan
  ) as browsetime
from
  ###({{FGT_DATASET_TRAFFIC_EB_TIME}})### 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
  ###({{FGT_DATASET_TRAFFIC_EB_TIME}})### t group by user_src order by browsetime desc
```

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

```
select
  user_src,
  ebtr_value(
    ebtr_agg_flat(browsetime),
    null,
    $timespan
  ) as browsetime
from
  ###({{FGT_DATASET_TRAFFIC_EB_TIME}})### 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
  coalesce(ap, srcintf) as ap_srcintf,
  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
  )
group by
  ap_srcintf
having
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  )& gt; 0
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, srcssid, osname, osversion, devtype, srcmac,
count(*) as subtotal from $log where $filter and (logflag&1>0) and (srcssid is not null or
dstssid is not null) and srcmac is not null group by ap_srcintf, srcssid, osname, osversion,
devtype, srcmac order by subtotal desc)### 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(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  ) as bandwidth
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
  and srcssid is not null
group by
  srcssid
having
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  )& gt; 0
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
  ###({{FGT_DATASET_TRAFFIC_TOP_WIFI_CLIENT}})### 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)
  )& gt; 0
order by
  bandwidth desc
```

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

```
select
  (
    coalesce(srcname, srcmac, 'unknown') || ' (' || coalesce(devtype, 'unknown') || ', ' ||
coalesce(osname, '') || (
      case when osversion is null then '' else ' ' || osversion end
    ) || ')'
  ) as client,
  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
  )
group by
  client
having
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  )& gt; 0
order by
  bandwidth desc
```

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

```
select
  (
    coalesce(osname, 'unknown') || ' ' || coalesce(osversion, '')
  ) as os,
  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
  )
group by
  os
having
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  )& gt; 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, 'unknown') || ' ' || coalesce(osversion, '')
  ) as os,
  count(distinct srcmac) as totalnum
```

```
from
  ###({{FGT_DATASET_TRAFFIC_TOP_WIFI_CLIENT}})### t 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
  devtype,
  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 devtype is not null
group by
  devtype
having
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  )& gt; 0
order by
  bandwidth desc
```

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

```
select
  devtype,
  count(distinct srcmac) as totalnum
from
  ###(select srcintf, srcssid, osname, osversion, devtype, srcmac, count(*) as subtotal from
$log where $filter and (logflag&1>0) and (srcssid is not null or dstssid is not null) and
srcmac is not null group by srcintf, srcssid, osname, osversion, devtype, srcmac order by sub-
total desc)### t where devtype is not null group by devtype order by totalnum desc
```

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

```
select
  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
  )
```

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

```
select
  count(distinct srcmac) as totalnum
from
  ###(select srcintf, srcssid, osname, osversion, devtype, srcmac, count(*) as subtotal from
$log where $filter and (logflag&1>0) and (srcssid is not null or dstssid is not null) and
srcmac is not null group by srcintf, srcssid, osname, osversion, devtype, srcmac order by sub-
total desc)### 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)
  )& gt; 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)
  )& gt; 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<52) and (hostname is not null
or utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-fil-
ter')))) 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>=52) 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)
  )& gt; 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)
  )& gt; 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 ('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 (
    '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(sent-
byte, 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<52) 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>=52) group by hod, website order by hod, hits desc)### t group by hod, web-
site 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 (coun-
tweb>0 or ((logver is null or logver<52) and (hostname is not null or utmevent in ('web-
filter', '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
  ###({{FGT_DATASET_WEBFILTER_TOP_WEBSITES}})### 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(sent-
byte, 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, web-
site, 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 (coun-
tweb>0 or ((logver is null or logver<52) and (hostname is not null or utmevent in ('web-
filter', '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
  ###({{FGT_DATASET_WEBFILTER_TOP_WEBSITES}})### t group by website, catdesc order by sessions
desc
```

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

```
select
  clientfeature,
  count(distinct hostname) as totalnum
from
  ###({{FGT_DATASET_FCTEVENT_FORTICLIENT}})### t where clientfeature is not null group by cli-
entfeature order by totalnum desc
```

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

```
select
  os,
  count(distinct hostname) as totalnum
from
  ###({{FGT_DATASET_FCTEVENT_FORTICLIENT}})### t where os 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 hostname) as totalnum
from
  ###({{FGT_DATASET_FCTEVENT_FORTICLIENT}})### 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 hostname) as totalnum
from
  ###(select hostname, coalesce(nullifna(usingpolicy), 'No Profile') as profile from $log
where $filter group by hostname, 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,
  profile,
  hostuser,
  fctver,
  from_itime(
    max(itime)
  ) as last_seen
from
  ###(select hostname, deviceip, os, nullifna(usingpolicy) as profile, nullifna(`user`) as hos-
tuser, fctver, max(itime) as itime from $log where $filter and os is not null group by host-
name, deviceip, os, profile, hostuser, fctver order by itime desc)### t group by hostname,
deviceip, os, profile, hostuser, 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, 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)### union all
###(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 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 ('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, 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)### union all ###(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 total-
num desc)###) 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, 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)### union all
###(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 order by threat
```

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

```
select
  remotename,
  hostname,
  hostuser,
  utmaction,
  sum(total) as totalnum,
  from_dtime(
    max(dtime)
  ) as last_seen
from
  ###(select remotename, hostname, coalesce(nullifna(`user`), 'Unknown') as hostuser, utmac-
tion, 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 remotename, hostname, hostuser, 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 utmac-
tion='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 = '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,
  count(distinct vulnname) as totalnum
from
  ###(select vulnseverity, vulnname from $log where $filter and nullifna(vulnseverity) is not
null and nullifna(vulnname) is not null group by vulnseverity, vulnname)### t group by
vulnseverity order by totalnum 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 = '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_num-
ber 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 = '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_num-
ber 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(vul-
nname) 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, vul-
nname, 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,
  (
    '<div style=word-break:normal>' || 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 vul-
nname 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,
  (
    '<div style=word-break:normal>' || 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
  (
    '<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, ',') 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
os-Detect-OS-Count	Detected operation system count	traffic

```
select
  (
    coalesce(osname, '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(band-
width)>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(band-
width)>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>=52) and host-
name is not null group by user_src, hostname order by requests desc)###) t where $filter-drill-
down 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>=52) and host-
name is not null group by user_src, hostname order by requests desc)###) t where $filter-drill-
down 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>=52) and host-
name is not null group by user_src, hostname order by requests desc)###) t where $filter-drill-
down 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>=52) and host-
name is not null group by user_src, hostname order by requests desc)###) t where $filter-drill-
down 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 event-
type 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 event-
type 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 event-
type 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 event-
type 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 '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>=52) and nullifna(virus) is not null group
by virus, virusid_s /*SkipSTART*/order by totalnum desc/*SkipEND*/)### t group by virus, mal-
ware_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>=52) 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
  ###({{FGT_DATASET_WEBFILTER_TOP_WEB_BY_REQUEST}})### 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
  ###({{FGT_DATASET_WEBFILTER_TOP_WEB_BY_REQUEST}})### 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
  ###({{FGT_DATASET_WEBFILTER_TOP_WEB_CATS}})### 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
  ###({{FGT_DATASET_WEBFILTER_TOP_WEB_CATS}})### 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
  ###({{FGT_DATASET_ATTACK_TOP_ATTACKS}})### 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
  ###({{FGT_DATASET_ATTACK_TOP_ATTACKS}})### 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(vir-
usid, 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
  hourstamp,
  sum(totalnum) as totalnum
from
  ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, $hour_of_day as
hourstamp, count(*) as totalnum from $log where $filter and `to` is not null and action in
('detected', 'blocked') group by user_src, hourstamp order by hourstamp)### t where $filter-
drilldown 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 coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, `from` as mf_sender,
count(*) as totalnum from $log where $filter and `from` is not null and action in ('detected',
'blocked') group by user_src, mf_sender order by totalnum desc)### t where $filter-drilldown
group by mf_sender order by totalnum desc
```

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

```
select
  hourstamp,
  cast(
    sum(cpu_usage)/ sum(num) as decimal(6, 2)
  ) as cpu_avg_usage
from
  ###(select $hour_of_day as hourstamp, sum(cpu) as cpu_usage, count(*) as num from $log where
$filter and subtype='system' and action='perf-stats' group by hourstamp)### t group by
hourstamp order by hourstamp
```

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

```
select
  hourstamp,
  cast(
    sum(mem_usage)/ sum(num) as decimal(6, 2)
  ) as mem_avg_usage
from
  ###(select $hour_of_day as hourstamp, sum(mem) as mem_usage, count(*) as num from $log where
$filter and subtype='system' and action='perf-stats' group by hourstamp)### t group by
hourstamp order by hourstamp
```

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

```
select
  hourstamp,
  cast(
    sum(sess_usage)/ sum(num) as decimal(10, 2)
  ) as sess_avg_usage
from
  ###(select $hour_of_day as hourstamp, sum(totalsession) as sess_usage, count(*) as num from
$log where $filter and subtype='system' and action='perf-stats' group by hourstamp)### t group
by hourstamp order by hourstamp
```

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

```
select
  hourstamp,
  cast(
    sum(sess_usage)/ sum(num) as decimal(10, 2)
  ) as sess_avg_usage,
  cast(
    sum(cpu_usage)/ sum(num) as decimal(6, 2)
  ) as cpu_avg_usage
from
  ###(select $hour_of_day as hourstamp, sum(cpu) as cpu_usage, sum(totalsession) as sess_
usage, count(*) as num from $log where $filter and subtype='system' and action='perf-stats'
group by hourstamp)### 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)
  )& gt; 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)& gt; 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
  devtype,
  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
  devtype,
  dev_src
```

```
having
  sum(crscore % 65536)& gt; 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(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  ) as bandwidth
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
  and nullifna(appcat) is not null
group by
  appcat
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(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  ) as bandwidth
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
  and nullifna(appcat) is not null
group by
  appcat
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<52) 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 (
    '80/tcp', '443/tcp', 'HTTP', 'HTTPS',
    'http', 'https'
  )
group by
  app_group,
  service
having
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  )& gt; 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<52) 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<52) 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 vis-
its from $log where $filter and (eventtype is null or logver>=52) 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
  ###({{FGT_DATASET_TRAFFIC_TOP_DST_COUNTRY_BY_EB_TIME}})### t group by dstcountry order by
browsetime desc
```

Dataset Name	Description	Log Category
Top-Destination-Countries-By-Browsing-Time-Enhanced	Traffic top destination countries by browsing time enhanced	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
  ###({{FGT_DATASET_TRAFFIC_TOP_DST_COUNTRY_BY_EB_TIME}})### 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
  ###({{FGT_DATASET_TRAFFIC_TOP_DOMAINS_BY_EB_TIME}})### t group by hostname order by brow-
setime desc
```

Dataset Name	Description	Log Category
App-Risk-Traffic-Top-Hostnames-By-Browsing-Time-Enhanced	Traffic top domains by browsing time enhanced	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
  ###({{FGT_DATASET_TRAFFIC_TOP_DOMAINS_BY_EB_TIME}})### t group by hostname order by brow-
setime 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 = '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 = '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 = '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 = '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 = '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 '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>=52) and nullifna(virus) is not null group
by virus, virusid_s /*SkipSTART*/order by totalnum desc/*SkipEND*/)### t group by virus, mal-
ware_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>=52) 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
  utmsubtype,
  sum(number) as number
from
  ###(select subtype::text as utmsubtype, count(*) as number from $log where $filter and sub-
type is not null group by subtype order by number desc)### t group by utmsubtype order by num-
ber 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>=52) 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, ',')
  ) 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, ',')
  ) 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 appcat, app from $log where $filter and app is not null and appcat is not null
and (logflag&1>0) and apprisk in ('critical', 'high') group by appcat, app)### t 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-web-
filter where $filter and cat in (26, 61) group by hostname)### union all ###(select cast('Crit-
ical & 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 total-
num from $log t1 left join (select name, cve, vuln_type from ips_mdata) t2 on t1.at-
tack=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 appcat, app from $log where $filter and app is not null and appcat is not null
and (logflag&1>0) and apprisk in ('critical', 'high') group by appcat, app)### t 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 lower(app) as lowapp, 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 lowapp, user_src order by
bandwidth desc)### t1 inner join app_mdata t2 on t1.lowapp=lower(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, count(*) as total_num from $log where $filter and lower(appcat)
in ('botnet', 'remote.access', 'storage.backup', 'video/audio', 'p2p', 'proxy') and (log-
flag&1>0) and apprisk in ('critical', 'high') group by appcat)### union all ###(select 'mali-
cious' as behavior,  count(*) as total_num from $log-attack where $filter and (logflag&16>0)
and severity in ('critical', 'high') group by behavior)###) t group by behavior order by per-
centage 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(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  ) as bandwidth
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
  and nullifna(appcat) is not null
group by
  appcat
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 f_user) as user_num,
  sum(bandwidth) as bandwidth,
  sum(num_session) as num_session
from
  ###(select appcat, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`))
as f_user, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, count(*) as num_ses-
sion from $log where $filter and (logflag&1>0) and nullifna(appcat) is not null group by
appcat, app, f_user)### t 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<52) 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)= '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 '%Poss-
ibleThreat.SB%' group by virus_s, dstip, srcip, appid, app )### t where virus_s like '%Poss-
ibleThreat.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, ana-
lyticscksum, 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)= '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)= '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(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  ) as bandwidth
from
  $log
```

```
where
  $filter
  and (
    logflag&1>0
  )
  and nullifna(appcat) is not null
group by
  appcat
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 f_user) as user_num,
  sum(bandwidth) as bandwidth,
  sum(num_session) as num_session
from
  ###(select appcat, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`))
as f_user, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, count(*) as num_ses-
sion from $log where $filter and (logflag&1>0) and nullifna(appcat) is not null group by
appcat, app, f_user)### t 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<52) 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)= '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 (log-
flag&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 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 as app, 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 app,
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) || '://' || 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 total-
num from $log t1 left join (select name, cve, vuln_type from ips_mdata) t2 on t1.at-
tack=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
  $log
where
  $filter
  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 'incoming' then 'Download' when 'outgoing' then 'Upload' end
  ) as action,
  max(filesize) as filesize,
  service
from
  $log
where
  $filter
  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 'antivirus' then 'Malware incidents' when 'webfilter' then 'Mali-
cious/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`),
    '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 & gt;= '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`),
    'Unknown'
  ) as f_user,
  coalesce(
    nullifna(hostname),
    'Unknown'
  ) as host_name,
  virus,
  file
from
  $log
where
  $filter
  and clientfeature = '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(host-
name), '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(nul-
lifna(hostname), 'Unknown') as host_name, msg, count(*) as total_num from $log where $filter
and clientfeature='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 = '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 = '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 = '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 = '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 = '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 'LESS_ONE_MIN' when duration<600 then 'LESS_TEN_MIN' when dur-
ation<3600 then 'LESS_ONE_HOUR' when duration & gt;= 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
        ###({{FGT_DATASET_BASE_TRAFFIC_APP_BOTNET}})### t group by app, user_src order by
events desc) union all (select attack as app, user_src, sum(totalnum) as events from ###
({{FGT_DATASET_BASE_ATTACK_APP_BOTNET}})### t group by app, 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,
  host_mac,
  sum(events) as events
from
  (
    ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
devtype, 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, host_mac order by
events desc)### union all ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr
(`srcip`)) as user_src, 'Unknown' as devtype, hostname as host_mac, count(*) as events from
$log-attack where $filter and (logflag&16>0) group by user_src, devtype, host_mac order by
events desc)###) t group by user_src, devtype, 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
        ###({{FGT_DATASET_BASE_TRAFFIC_APP_BOTNET}})### t group by app order by events desc)
union all (select attack as app, sum(totalnum) as events from ###({{FGT_DATASET_BASE_ATTACK_
APP_BOTNET}})### t group by app 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,
      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)###) t1 union all (select dstip, root_domain(hostname) as
domain, sum(totalnum) as events from ###({{FGT_DATASET_BASE_ATTACK_APP_BOTNET}})### t group by
dstip, domain order by events desc)) 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
        ###({{FGT_DATASET_BASE_TRAFFIC_APP_BOTNET}})### t group by user_src) union all (select
user_src, sum(totalnum) as events from ###({{FGT_DATASET_BASE_ATTACK_APP_BOTNET}})### 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 $fil-
ter 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 ('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 '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
        ###({{FGT_DATASET_EVENT_COMPLIANCE_CHECK}})### 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
        ###({{FGT_DATASET_EVENT_COMPLIANCE_CHECK}})### 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 = '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 res-
ult 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 res-
ult='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 res-
ult='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, ','),
    120
  ) as practice,
  (
    case when sum(fail_count)& gt; 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 = '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,
  `from` as sender,
  `to` as receiver,
  regexp_replace(filename, '.*/', '') as filename,
  filesize,
  profile,
  action,
  direction
from
  $log
where
  $filter
  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
  $log
where
  $filter
  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
  $log
where
  $filter
  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
  $log
where
  $filter
  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
```

```
  $log
where
  $filter
  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
  $log
where
  $filter
  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 sub-
type='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 (
      'bat', 'cmd', 'exe', 'jar', 'msi', 'vbs',
      '7z', 'zip', 'gzip', 'lzw', 'tar',
      'rar', 'cab', 'doc', 'docx', '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
  ###({{FGT_DATASET_BASE_VIRUS_FSA_DETECTED_FILE_TYPES}})### 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 (
      '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
  ###({{FGT_DATASET_BASE_VIRUS_FSA_DETECTED_FILE_TYPES}})### 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 '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 = '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 lower(app) as lowapp, 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 lowapp, user_src order by
bandwidth desc)### t1 inner join app_mdata t2 on t1.lowapp=lower(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 total-
num from $log t1 left join (select name, cve, vuln_type from ips_mdata) t2 on t1.at-
tack=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)= '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, ser-
vice as app, dstip, srcip, count(*) as total_num from $log-attack where $filter and (log-
flag&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)& gt; 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 ('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)
  )& gt; 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 & lt;& gt; 'Reserved'
group by
  srccountry
having
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  )& gt; 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
  ###({{FGT_DATASET_BASE_TRAFFIC_APP_BANDWIDTH}})### 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
  ###({{FGT_DATASET_BASE_TRAFFIC_APP_BANDWIDTH}})### 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
  ###({{FGT_DATASET_BASE_TRAFFIC_APP_BANDWIDTH}})### 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
  ###({{FGT_DATASET_BASE_TRAFFIC_APP_BANDWIDTH}})### 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
  ###({{FGT_DATASET_BASE_TRAFFIC_APP_BANDWIDTH}})### 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
  ###({{FGT_DATASET_BASE_TRAFFIC_APP_BANDWIDTH}})### 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
  ###({{FGT_DATASET_BASE_TRAFFIC_APP_BANDWIDTH}})### 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
  ###({{FGT_DATASET_BASE_TRAFFIC_APP_BANDWIDTH}})### 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<52) 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 (
    '80/tcp', '443/tcp', 'HTTP', 'HTTPS',
    'http', 'https'
  )
group by
  app_group,
  service
having
```

```
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  )& gt; 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 vis-
its from $log where $filter and (eventtype is null or logver>=52) 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, ', ') 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
  ###({{FGT_DATASET_TRAFFIC_TOP_SITES_BY_EB_TIME}})### 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 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 'SaaS Apps' else 'Other Apps' end
  ) as app_type,
  count(distinct app_s) as total_num
from
  ###({{FGT_DATASET_TRAFFIC_SAAS_APPS}})### 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 'Sanctioned' else 'Unsanctioned' end
  ) as saas_cat_str,
  count(distinct app_s) as num_saas_app
from
  ###({{FGT_DATASET_TRAFFIC_SAAS_APP_BY_CAT}})### 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 'Sanctioned' else 'Tolerated' end
  ) as saas_cat_str,
  sum(bandwidth) as bandwidth
```

```
from
  ###({{FGT_DATASET_TRAFFIC_SAAS_APP_BY_CAT}})### 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 'Sanctioned' else 'Tolerated' end
  ) as saas_cat_str,
  sum(total_app) as total_app
from
  ###({{FGT_DATASET_TRAFFIC_SAAS_APP_BY_CAT}})### 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 '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 '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(sent-
byte, 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 '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 'Sanctioned' else 'Unsactioned' end
  ) as saas_cat_str,
  count(distinct saasuser) as total_user
from
  ###({{FGT_DATASET_TRAFFIC_SAAS_TOP_USER}})### 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 'Sanctioned' else 'Unsactioned' end
  ) as saas_cat_str,
  count(distinct app_s) as total_app
from
  ###({{FGT_DATASET_TRAFFIC_SAAS_TOP_USER}})### 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(sent-
byte) 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
  ###({{FGT_DATASET_TRAFFIC_TOLERATED_SAAS_APP}})### 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
  ###({{FGT_DATASET_TRAFFIC_TOLERATED_SAAS_APP}})### 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
  ###({{FGT_DATASET_TRAFFIC_SAAS_APP_GROUP}})### 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 '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
      ###({{FGT_DATASET_TRAFFIC_SAAS_APP_GROUP}})### 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 '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
      ###({{FGT_DATASET_TRAFFIC_SAAS_APP_GROUP}})### 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
  '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-Devices	Network 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, max(itime) as max_itime from $log where $pre_period $filter and hostname is not
null group by hostname, os, srcip order by max_itime desc)###; create temporary table rpt_
tmptbl_2 as ###(select hostname, os, srcip, max(itime) as max_itime from $log where $filter
and hostname is not null group by hostname, os, srcip order by max_itime desc)###; select
'Unseen Devices' as category, count(distinct hostname) as total_num from rpt_tmptbl_1 where
not exists (select 1 from rpt_tmptbl_2 where rpt_tmptbl_1.hostname=rpt_tmptbl_2.hostname)
union all select 'New Devices', count(distinct hostname) as total_num from rpt_tmptbl_2 where
not exists (select 1 from rpt_tmptbl_1 where rpt_tmptbl_1.hostname=rpt_tmptbl_2.hostname)
union all select 'Seen Devices', count(distinct t1.hostname) as total_num from rpt_tmptbl_1 t1
inner join rpt_tmptbl_2 t2 on t1.hostname=t2.hostname
```

Dataset Name	Description	Log Category
aware-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, max(srcip) as srcip, max(itime) as max_itime from $log where $pre_period $filter and
hostname is not null group by hostname, os order by max_itime desc)###; create temporary table
rpt_tmptbl_2 as ###(select hostname, os, max(srcip) as srcip, max(itime) as max_itime from
$log where $filter and hostname is not null group by hostname, os order by max_itime desc)###;
select from_itime(max(max_itime)) as timestamp, hostname, max(fctos_to_devtype(os)) as dev-
type, string_agg(distinct os, '/') as os_agg, max(srcip) as srcip 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 timestamp desc
```

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 nul-
lifna(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 nul-
lifna(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 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, '[^ ]*$', '') 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, vul-
nname, 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-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 '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 $fil-
ter 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 day-
stamp, 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 day-
stamp, 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(`xau-
thuser`), nullifna(`user`)) is not null group by f_user, tunneltype)### t group by f_user, tun-
neltype 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 || '(' || 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,
  'Failed' as results,
  count(distinct reason) as total_num
from
  ###(select devid, reason from $log where $filter and subtype='compliance-check' and res-
ult='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
      epeudevmap
    where
      euid & gt;= 1024
  );
select
  coalesce(
    nullifna(epname),
    nullifna(
      ipstr(`srcip`)
    ),
    'Unknown'
  ) as epname,
  user_agg,
  sevid,
  (
    CASE sevid WHEN 5 THEN 'Critical' WHEN 4 THEN 'High' WHEN 3 THEN 'Medium' WHEN '2' THEN
'Info' ELSE 'Low' END
  ) as severity,
  threats,
  bl_count as total_bl
from
  (
    select
      th1.epid,
      srcip,
      sevid,
      bl_count,
      threats
    from
      (
        select
          epid,
          srcip,
          max(verdict)+ 1 as sevid,
          sum(bl_count) as bl_count
        from
          (
            (
              select
                epid,
                srcip,
                day_st as itime,
                bl_count,
                verdict
              from
                $ADOMTBL_PLHD_IOC_VERDICT
              where
                bl_count>0
            )
            union all
              (
                select
                  epid,
                  srcip,
                  day_st as itime,
                  bl_count,
                  verdict
                from
                  $ADOMTBL_PLHD_INTERIM_IOC_VERDICT
                where
                  bl_count>0
              )
          ) tvdt
        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,
                      unnest(threatid) as thid,
                      day_st as itime
                    from
                      $ADOMTBL_PLHD_IOC_VERDICT
                    where
                      bl_count>0
                  )
                  union all
                    (
                      select
                        epid,
                        unnest(threatid) as thid,
                        day_st as itime
                      from
                        $ADOMTBL_PLHD_INTERIM_IOC_VERDICT
                      where
                        bl_count>0
                    )
                ) t
              where
                $filter
                and $filter - drilldown
              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 endusers teu on tpu.euid = teu.euid
    group by
      epid
  ) t2 on t2.epid = t1.epid
  inner join endpoints 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),
        'Day'
      ) as day_st
    from
      (
        (
          select
            epid,
            day_st as itime
          from
            $ADOMTBL_PLHD_INTERIM_IOC_VERDICT
          where
            cs_count>0
        )
        union all
          (
            select
              epid,
              day_st as itime
            from
              $ADOMTBL_PLHD_IOC_VERDICT
            where
              cs_count>0
          )
      ) t
    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),
        'Day'
      ) as day_st
    from
      (
        (
          select
```

```
            epid,
            day_st as itime
          from
            $ADOMTBL_PLHD_INTERIM_IOC_VERDICT
          where
            cs_count>0
        )
        union all
          (
            select
              epid,
              day_st as itime
            from
              $ADOMTBL_PLHD_IOC_VERDICT
            where
              cs_count>0
          )
      ) t
    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`)
    ),
    '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
              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
                from
                  $ADOMTBL_PLHD_INTERIM_IOC_VERDICT
                where
                  bl_count = 0
                  and cs_count>0
              )
          ) tvdt
        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,
                      unnest(threatid) as thid,
                      day_st as itime
                    from
                      $ADOMTBL_PLHD_IOC_VERDICT
                    where
                      bl_count = 0
                      and cs_count>0
                  )
                  union all
                    (
                      select
                        epid,
                        unnest(threatid) as thid,
                        day_st as itime
                      from
                        $ADOMTBL_PLHD_INTERIM_IOC_VERDICT
                      where
                        bl_count = 0
                        and cs_count>0
                    )
                ) tt1
              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 endpoints tep on tep.epid = t.epid
order by
  max_verdict desc,
  max_cs desc,
  total_cs 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 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)###; create temporary table rpt_tmptbl_2 as ###(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)###; 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 $log where $pre_period $filter and hostname is not null group by hostname,
os, srcip, fctver order by hostname)###; create temporary table rpt_tmptbl_2 as ###
(select hostname, os, srcip, fctver from $log where $filter and hostname is not null group by
hostname, os, srcip, fctver order by hostname)###; 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 $log where $pre_period $filter and nullifna(srcproduct) is not null group by
srcproduct, hostname order by srcproduct)###; create temporary table rpt_tmptbl_2 as ###
(select srcproduct, hostname from $log where $filter and nullifna(srcproduct) is not null
group by srcproduct, hostname order by srcproduct)###; 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
  *
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 * 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 day-
stamp)### 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 coalesce(bot-
netdomain, 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
$pre_period $filter and (botnetdomain is not null or botnetip is not null) group by domain,
action, srcip, sevid order by sevid desc)###; create temporary table rpt_tmptbl_2 as ###
(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 'Redir-
ected' else 'Passed' end) as action, srcip, (CASE WHEN level IN ('critical', 'alert', 'emer-
gency') 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)###; 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
  *
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 * 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 $fil-
ter 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, vul-
nname, vulnseverity, vulncat, hostname from $log where $pre_period $filter and nullifna(vul-
nname) is not null group by vulnid, vulnname, vulnseverity, vulncat, hostname)###; create
temporary table rpt_tmptbl_2 as ###(select vulnid, vulnname, vulnseverity, vulncat, hostname
from $log where $filter and nullifna(vulnname) is not null 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(dis-
tinct hostname) as host_num, cve_id from rpt_tmptbl_2 t1 left join fct_mdata t2 on t1.vul-
nid=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, vul-
nname, vulnseverity, vulncat, hostname from $log where $pre_period $filter and nullifna(vul-
nname) is not null group by vulnid, vulnname, vulnseverity, vulncat, hostname)###; create
temporary table rpt_tmptbl_2 as ###(select vulnid, vulnname, vulnseverity, vulncat, hostname
from $log where $filter and nullifna(vulnname) is not null 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 pri = '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, '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 distinct on
(1, 2) fgtserial, hostname, deviceip, os, dtime from $log where $pre_period $filter and host-
name is not null order by fgtserial, hostname, dtime desc)###; create temporary table rpt_
tmptbl_2 as ###(select distinct on (1, 2) fgtserial, hostname, deviceip, os, dtime from $log
where $filter and hostname is not null order by fgtserial, hostname, dtime desc)###; 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.host-
name, 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 = 'traffic-count'
group by
  apn
having
  sum(
    coalesce(`u-bytes`, 0)
  )& gt; 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 = '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 = 'traffic-count'
group by
  apn
having
  sum(
    coalesce(`u-pkts`, 0)
  )& gt; 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
  ###({{FGT_DATASET_DNS_BOTNET_DOMAINS}})### 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
  ###({{FGT_DATASET_DNS_BOTNET_DOMAINS}})### 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
  ###({{FGT_DATASET_DNS_BOTNET_DOMAINS}})### 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 'Critical' WHEN 4 THEN 'High' WHEN 3 THEN 'Medium' WHEN '2' THEN
'Info' ELSE 'Low' END
  ) as severity
from
  ###({{FGT_DATASET_DNS_BOTNET_DOMAINS}})### 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', 'emer-
gency') 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 = 'block'
    or logid_to_int(logid)= 54001
  )
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 = 'block'
    or logid_to_int(logid)= 54001
  )
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)= 54001
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 = '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,
      (
        case when devid like 'F6KF%' then (
          case when slot = 0 then 'MBD' else 'FPC' end
        ) when devid like 'F7KE%' then (
          case when slot in (1, 2) then 'FIM' else 'FPM' end
        ) else null end
      ) 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(
        sum(lograte_peak) as decimal(10, 2)
      ) as lograte_peak,
      sum(session_peak) as session_peak,
      sum(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, count(*) as count, sum
(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) 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, sum(coalesce(trate, 0)) as total_trate, sum(coalesce(erate,
0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, max(coalesce(trate, 0)+coalesce
(erate, 0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalses-
sion, 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 from $log where $filter 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,
      (
        case when devid like 'F6KF%' then (
          case when slot = 0 then 'MBD' else 'FPC' end
        ) when devid like 'F7KE%' then (
          case when slot in (1, 2) then 'FIM' else 'FPM' end
        ) else null end
      ) 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(
        sum(lograte_peak) as decimal(10, 2)
      ) as lograte_peak,
      sum(session_peak) as session_peak,
      sum(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, count(*) as count, sum
(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) 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, sum(coalesce(trate, 0)) as total_trate, sum(coalesce(erate,
0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, max(coalesce(trate, 0)+coalesce
(erate, 0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalses-
sion, 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 from $log where $filter 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,
    (
      case when devid like 'F6KF%' then (
        case when slot = 0 then 'MBD' else 'FPC' end
      ) when devid like 'F7KE%' then (
        case when slot in (1, 2) then 'FIM' else 'FPM' end
      ) else null end
    ) 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(
      sum(lograte_peak) as decimal(10, 2)
    ) as lograte_peak,
    sum(session_peak) as session_peak,
    sum(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, count(*) as count, sum
(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) 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, sum(coalesce(trate, 0)) as total_trate, sum(coalesce(erate,
0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, 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 from $log where $filter 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,
          (
            case when devid like 'F6KF%' then (
              case when slot = 0 then 'MBD' else 'FPC' end
            ) when devid like 'F7KE%' then (
              case when slot in (1, 2) then 'FIM' else 'FPM' end
            ) else null end
          ) 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(
            sum(lograte_peak) as decimal(10, 2)
          ) as lograte_peak,
          sum(session_peak) as session_peak,
          sum(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, count(*) as count, sum
    (coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) 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, sum(coalesce(trate, 0)) as total_trate, sum(coalesce(erate,
    0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, max(coalesce(trate, 0)+coalesce
    (erate, 0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalses-
    sion, 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 from $log where $filter 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,
      (
        case when devid like 'F6KF%' then (
          case when slot = 0 then 'MBD' else 'FPC' end
        ) when devid like 'F7KE%' then (
          case when slot in (1, 2) then 'FIM' else 'FPM' end
        ) else null end
      ) 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(
        sum(lograte_peak) as decimal(10, 2)
      ) as lograte_peak,
      sum(session_peak) as session_peak,
      sum(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, count(*) as count, sum
(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) 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, sum(coalesce(trate, 0)) as total_trate, sum(coalesce(erate,
0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, max(coalesce(trate, 0)+coalesce
(erate, 0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalses-
sion, 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 from $log where $filter 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,
      (
        case when devid like 'F6KF%' then (
          case when slot = 0 then 'MBD' else 'FPC' end
        ) when devid like 'F7KE%' then (
          case when slot in (1, 2) then 'FIM' else 'FPM' end
        ) else null end
      ) 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(
        sum(lograte_peak) as decimal(10, 2)
      ) as lograte_peak,
      sum(session_peak) as session_peak,
      sum(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, count(*) as count, sum
(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) 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, sum(coalesce(trate, 0)) as total_trate, sum(coalesce(erate,
0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, max(coalesce(trate, 0)+coalesce
(erate, 0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalses-
sion, 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 from $log where $filter 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,
      (
        case when devid like 'F6KF%' then (
          case when slot = 0 then 'MBD' else 'FPC' end
        ) when devid like 'F7KE%' then (
          case when slot in (1, 2) then 'FIM' else 'FPM' end
        ) else null end
      ) 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(
        sum(lograte_peak) as decimal(10, 2)
      ) as lograte_peak,
      sum(session_peak) as session_peak,
      sum(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, count(*) as count, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) 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, sum(coalesce(trate, 0)) as total_trate, sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, 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 from $log where $filter 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,
  (
    case when devid like 'F6KF%' then (
      case when slot = 0 then 'MBD' else 'FPC' end
    ) when devid like 'F7KE%' then (
      case when slot in (1, 2) then 'FIM' else 'FPM' end
    ) else null end
  ) 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(
    sum(lograte_peak) as decimal(10, 2)
  ) as lograte_peak,
  sum(session_peak) as session_peak,
  sum(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, count(*) as count, sum(coalesce
(mem, 0)) as total_mem, max(coalesce(mem, 0)) 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, sum(coalesce(trate, 0)) as total_trate, sum(coalesce(erate, 0)) as total_erate,
sum(coalesce(orate, 0)) as total_orate, 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 from $log where $filter 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,
  (
    case when devid like 'F6KF%' then (
      case when slot = 0 then 'MBD' else 'FPC' end
    ) when devid like 'F7KE%' then (
      case when slot in (1, 2) then 'FIM' else 'FPM' end
    ) else null end
  ) 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(
    sum(lograte_peak) as decimal(10, 2)
  ) as lograte_peak,
  sum(session_peak) as session_peak,
  sum(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, count(*) as count, sum(coalesce
(mem, 0)) as total_mem, max(coalesce(mem, 0)) 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, sum(coalesce(trate, 0)) as total_trate, sum(coalesce(erate, 0)) as total_erate,
sum(coalesce(orate, 0)) as total_orate, 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 from $log where $filter 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,
  (
    case when devid like 'F6KF%' then (
      case when slot = 0 then 'MBD' else 'FPC' end
    ) when devid like 'F7KE%' then (
      case when slot in (1, 2) then 'FIM' else 'FPM' end
    ) else null end
  ) 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(
    sum(lograte_peak) as decimal(10, 2)
  ) as lograte_peak,
  sum(session_peak) as session_peak,
  sum(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, count(*) as count, sum(coalesce
(mem, 0)) as total_mem, max(coalesce(mem, 0)) 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, sum(coalesce(trate, 0)) as total_trate, sum(coalesce(erate, 0)) as total_erate,
sum(coalesce(orate, 0)) as total_orate, 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 from $log where $filter 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
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-web-
filter where $filter and cat in (26, 61) group by hostname)### union all ###(select cast('Crit-
ical & 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
      (
        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
  ) t
where
  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 '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 Vis-
its' 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
2019-11-13	Initial release.