



FortiAnalyzer - Dataset Reference

Version 6.4.4



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



December 16, 2020 FortiAnalyzer 6.4.4 Dataset Reference 05-644-622945-20201216

TABLE OF CONTENTS

Introduction	
Understanding datasets and macros	
Dataset Reference List	5
Macro Reference List	
Change Log	

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
(traffic_in) as traffic_in from	ut, dwidth) as bandwidth, sum(traffic {{FGT_DATASET_BASE_TRAFFIC_BANDW dwidth desc)### t where \$filter-d	IDTH_SESSION}} base_query
Dataset Name	Description	Log Category
Session-Summary-Day-Of-Month	Number of session timeline	traffic
<pre>select \$flex_timescale(timestamp) as sum(sessions) as sessions from</pre>	hodex,	

###(select timestamp, sum(sessions) as sessions from {{FGT_DATASET_BASE_TRAFFIC_BANDWIDTH_ SESSION}} base_query group by timestamp order by sessions desc)### t where \$filter-drilldown group by hodex order by hodex

Dataset Name	Description	Log Category
Top-Users-By-Bandwidth	Bandwidth application top users by bandwidth usage	traffic
<pre>select coalesce(nullifna(`user`), nullifna(`unauthuse ipstr(`srcip`)) as user_src, sum(coalesce(sentbyte,) as bandwidth, sum(coalesce(rcvdbyte,) as traffic_in, sum(coalesce(sentbyte,) as traffic_out, count(*) as sessions</pre>	0)+ coalesce(rcvdbyte, 0) 0)	

```
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
<pre>select app_group_name(app) a sum(coalesce(sentbyte,) as bandwidth, sum(coalesce(rcvdbyte,) as traffic_in, sum(coalesce(sentbyte,</pre>	0)+ coalesce(rcvdbyte, 0) 0)	
<pre>) as traffic_out, count(*) as sessions from \$log where \$filter and (logflag&1>0</pre>		
) and nullifna(app) is group by app_group having sum(not null	
<pre>coalesce(sentbyte,)& gt; 0 order by bandwidth desc</pre>	0)+ coalesce(rcvdbyte, 0)	

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

```
nullifna(`user`),
```

Dataset Name	Description	Log Category
Top-App-By-Sessions	Top applications by session count	traffic
select		
app_group_name(app) as ap count(*) as sessions	p_group,	
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
<pre>select coalesce(nullifna(root_domain(hostnamo)</pre>		

```
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-Bytraffic Top destinations by bandwidth usage Bandwidth 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
<pre>drop table if exists rpt_tmptbl_1; </pre>		
<pre>drop table if exists rpt_tmptbl_2;</pre>		

```
drop
  table if exists rpt_tmptbl_3; create temporary table rpt_tmptbl_1 as
select
  devintf,
  mac
from
```

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

Dataset Name	Description	Log Category
Top-Wifi-Client-By-Bandwidth	Traffic top WiFi client by bandwidth usage	traffic
<pre>select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, srcssid, get_devtype(srcswversion, os coalesce(nullifna(`srcname`), `srcmac`) as hostname_mac, sum(coalesce(sentbyte, 0)+ coa) as bandwidth</pre>	name, devtype) as devtype_new, lesce(rcvdbyte, 0)	
<pre>from \$log where \$filter and (logflag&1>0) and (srcssid is not null or dstssid is not null</pre>		
) group by user_src, srcssid, devtype_new, 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
<pre>select \$flex_timescale(timestamp) as count(distinct(user_src)) as total user</pre>	hodex,	
<pre>from ###(select timestamp, user_sr</pre>	c, sum(sessions) as sessions from {{FGT group by timestamp, user_src order by s x 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',		
'command-block', 'script-filt	ter'	
)		
and hostname is not null		
and (
utmaction not in ('block', ') or action != 'deny'	DLOCKEd')	
)		
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<50200000) 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
<pre>select hostname, count(*) as requests from \$log where \$filter and (logflag&1>0) and utmevent in ('webfilter', 'banned-word 'command-block', 'script-) and hostname is not null and (utmaction in ('block', 'b or action = 'deny') group by</pre>	filter'	
hostname order by		
requests desc		

Dataset Name Description Log Category

Top-Web-Users-By-Request	UTM top web users by request	traffic
<pre>select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, get_devtype(srcswversion, srcname, count(*) as requests from</pre>	osname, devtype) as devtype_new,	

Dataset Reference List

```
$log
where
  $filter
  and (
  logflag&1>0
  )
  and utmevent in (
    'webfilter', 'banned-word', 'web-content',
    'command-block', 'script-filter'
  )
group by
 user_src,
 devtype_new,
 srcname
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 (a_{α} (readbut a_{α} 0)	
<pre>coalesce(sentbyte, 0)+ coales) as bandwidth,</pre>	ce(icvabyte, 0)	
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-filt		
)		
and hostname is not null		
group by		
appid,		
hostname,		
catdesc		
having		
<pre>sum(coalesce(sentbyte, 0)+ coales</pre>	c_{0} (readbut c_{0})	
)& qt; 0		
order by		
bandwidth desc		

Dataset Name	Description	Log Category
Top-Blocked-Web-Users	UTM top blocked web users	traffic
select		
coalesce(
nullifna(`user`),		
<pre>nullifna(`unauthuser`),</pre>		
ipstr(`srcip`)		
) as user_src,	sname douture) as douture new	
srcname,	sname, devtype) as devtype_new,	
count(*) as requests		
from		
\$log		
here		
\$filter		
and (
logflag&1>0		
)		
and utmevent in (
<pre>'webfilter', 'banned-word 'command-block', 'script-</pre>		
)		
, and (
utmaction in ('block', 'bl	locked')	
or action = 'deny'		
)		
roup by		
user_src,		
devtype_new,		
srcname		
order by		
requests desc		
	Description	Log Category

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,		
<pre>sum(traffic_in) as traffic_in,</pre>	,	
<pre>sum(traffic_out) as traffic_ou</pre>	lt	
from		
###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srci	p`)) as user_src,
<pre>sum(coalesce(sentbyte, 0)+coales</pre>	<pre>sce(rcvdbyte, 0)) as bandwidth, sum(coalesce</pre>	e(rcvdbyte, 0)) as
<pre>traffic_in, sum(coalesce(sentby)</pre>	te, 0)) as traffic_out from \$log-traffic whe	ere \$filter and (log
(1, 1, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,		

flag&l>0) and (countweb>0 or ((logver is null or logver<502000000) and (hostname is not null or utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter')))) group by user_src having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 /*SkipSTART*/order by bandwidth desc/*SkipEND*/)### t group by user_src order by bandwidth desc

Dataset Name	Description	Log Category
Top-Web-Users-By-Bandwidth	UTM top web users by bandwidth usage	traffic
select		
<pre>coalesce(nullifna(`user`),</pre>		
<pre>nullifna(`unauthuser`), ipstr(`srcip`)</pre>		
) as user_src,		
	sname, devtype) as devtype new,	
srcname,	_	
sum(
<pre>coalesce(sentbyte, 0) + coa</pre>	alesce(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_new,		
srcname naving		
sum (
coalesce(sentbyte, 0) + coa	alesce(rcvdbvte, 0)	
)& gt; 0		
order by		
bandwidth desc		
B (())		

Dataset Name	Description	Log Category
Top-Video-Streaming-Websites-By- Bandwidth	UTM top video streaming websites by bandwidth usage	traffic
<pre>select appid, hostname, sum(coalesce(sentbyte, 0)+ coales) as bandwidth, sum(coalesce(rcvdbyte, 0)</pre>	ce(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
<pre>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', '58 'smtps', 'SMTPS', '465/tcp')</pre>	7/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(
<pre>nullifna(`user`),</pre>		

```
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
<pre>else 'Virus' end) as malware_type, sum(totalnum) as totalnum from ###(select virus, virusic \$log where \$filter and (even</pre>	Riskware%' then 'Spyware' when w n L_to_str(virusid, eventtype) as nttype is null or logver>=50200 .d_s /*SkipSTART*/order by total	virus like 'Adware%' then 'Adware' virusid_s, count(*) as totalnum from 00000) and nullifna(virus) is not Lnum desc/*SkipEND*/)### t group by
Dataset Name	Description	Log Category

Dataset Name	Description	Log Category
Top-Virus-By-Name	UTM top virus	virus
<pre>select virus, max(virusid_s) as virusid, (</pre>		

```
FortiAnalyzer 6.4.4 Dataset Reference
Fortinet Technologies Inc.
```

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>=502000000) and nullifna(virus) is not null group by virus, virusid_s /*SkipSTART*/order by totalnum desc/*SkipEND*/)### t group by virus, malware_type order by totalnum desc

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

```
select
```

user_src, sum(totalnum) as totalnum

```
from
```

###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, count(*) as totalnum from \$log where \$filter and (eventype is null or logver>=502000000) and nullifna(virus) is not null group by user_src /*SkipSTART*/order by totalnum desc/*SkipEND*/)### t group by user_src order by totalnum desc

Dataset Name	Description	Log Category
Top-Attack-Source	UTM top attack source	attack
<pre>select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, count(*) as totalnum from \$log where \$filter group by user_src order by totalnum desc</pre>		
Dataset Name	Description	Log Category
Top-Attack-Victim	UTM top attack dest	attack
<pre>select dstip, count(*) as totalnum from \$log where \$filter and dstip is not null group by dstip</pre>		

order by totalnum desc

Dataset Name	Description	Log Category
Top-Static-IPSEC-Tunnels-By- Bandwidth	Top static IPsec tunnels by bandwidth usage	event
<pre>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,</pre>		
<pre>min(min_traffic_in) end) as traffic_in, (case when min(s_time) = ma out) - min(min_traffic_out) end) as traffic_out, (</pre>	<pre>x(e_time) then max(max_traffic_in) else max(ma x(e_time) then max(max_traffic_out) else max(m x(e_time) then max(max_traffic_in)+ max(max_tr</pre>	ax_traffic_
	fic_in) + max(max_traffic_out) - min(min_traffic	

###({{FGT_DATASET_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
<pre>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,</pre>		

```
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_DATASE_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 bandwidt	ch,	
<pre>sum(traffic_in) as traffic</pre>	_in,	
<pre>sum(traffic_out) as traffi</pre>		
from		
(
select		
devid,		
vd,		
tunnelid,		
remip,		
vpn_name,		
(
—	e) = max(e_time) then max(max_traffic_in) else ma	x(max_traffic_in)-
min(min_traffic_in) end		
) as traffic_in,		
(e) = max(e time) then max(max traffic out) else m	av/may traffic
out) - min(min traffic out) e		
) as traffic out,		
(
case when min(s time	e) = max(e time) then max(max traffic in) + max(ma	x traffic out) else
—	n traffic in) + max(max traffic out) - min(min tra	
) as bandwidth		_ ′
from		
###(//FCT DATASET EVEN	NT VPN DIAL UP IPSEC TUNNELS}})### t where not (tunnelin 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
<pre>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_ou from</pre>	t	
(select devid, vd,	ser_agg, ' ') as xauthuser_agg, gg, ' ') as user_agg,	
	ax(e_time) then max(max_traffic_in)+ max(max_ ffic_in)+ max(max_traffic_out)- min(min_traf;	
<pre>min(min_traffic_in) end) as traffic_in, (</pre>	<pre>ax(e_time) then max(max_traffic_in) else max ax(e_time) then max(max_traffic_out) else max</pre>	
<pre>out) - min(min_traffic_out) end) as traffic_out from ###({{FGT_DATASET_EVENT_VP</pre>	N_DIAL_UP_IPSEC_BANDWIDTH}})### t group by de	evid, vd, remip,

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 EVENT VPN DIAL UP IPSEC BANDWIDTH}})### t group by devid, vd, remip,
```

###({{FGT_DATASET_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
<pre>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 (</pre>		

```
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_DATASE_EVENT_SSL_VPN_TUNNEL_USERS}})### t group by devid, vd, user_src, remip,
```

###({{FGT_DATASE_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
	_VPN_TUNNEL_USERS}})### t where tunnel elid) tt group by user_src, remote_ip	
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_DATASE_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
<pre>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(duration) as duration, sum(bandwidth) as bandwidth, sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_out</pre>		

```
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 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 duration desc

Dataset Name	Description	Log Category
vpn-User-Login-history	VPN user login history	event
<pre>select \$flex_timescale(timestamp sum(total_num) as total_n</pre>		
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
```

(tunneltype like 'ipsec%' or tunneltype like 'ssl%') and action in ('tunnel-up', 'tunnelstats', '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(`xauthuser`), nullifna(`user`)) is not null group by f_user, tunneltype)### t group by f_user, tunneltype order by total_num desc

Dataset Name	Description	Log Category
vpn-Authenticated-Logins	VPN authenticated logins	event
<pre>select coalesce(xauthuser_agg, user_agg, ipstr(`remip`)) as f_user, t_type as tunneltype, from_dtime(min(s_time)) as start_time, cum(total num) as total num</pre>		
<pre>sum(total_num) as total_num sum(duration) as duration</pre>	,	
<pre>string_agg(distinct use t_type, devid, vd, remip, tunnelid, min(s_time) as s_time, max(e_time) as e_time, (</pre>	thuser_agg, ' ') as xauthuser_agg, r_agg, ' ') as user_agg,	
<pre>case when min(s_time): (min_duration) end) as duration,</pre>	= max(e_time) then max(max_duration)	else max(max_duration) - min
	<pre>= max(e_time) then max(max_traffic_in traffic_in) + max(max_traffic_out) - min</pre>	

order by total num desc

```
Dataset Name
                                   Description
                                                                                     Log Category
 vpn-Traffic-Usage-Trend-VPN-
                                   VPN traffic usage trend
                                                                                     event
 Summary
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 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='tunnelstats' and tunneltype like 'ipsec%' and (tunnelip is null or tunnelip='0.0.0.0') and nullifna (`user`) is null and tunnelid is not null and tunnelid!=0 group by tunnelid, tunneltype, vpntunnel, devid, vd /*SkipSTART*/order by tunnelid/*SkipEND*/)### t group by vpntunnel, tunneltype, tunnelid, devid, vd order by bandwidth desc) t where bandwidth>0 group by vpntunnel, tunneltype order by bandwidth desc

Dataset Name	Description	Log Category
Top-Dialup-IPSEC-By-Bandwidth-and- Availability	Top dialup IPsec users by bandwidth usage and avail	event
<pre>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 sum(rcvd_end - rcvd_beg) as sum(sent_end - sent_beg + rcvd </pre>	<pre>traffic_out, traffic_in,</pre>	

```
) 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-Bandwidtl and-Availability	h- Top SSL tunnel users by bandwidth usage and avail	event
<pre>select user_src, remote_ip, sum(traffic_out) as traffic_i sum(traffic_in) as traffic_i sum(bandwidth) as bandwidth, sum(uptime) as uptime from (select user_src, remip as remote_ip, tunnelid, devid, vd, sum(sent_end - sent_beg) sum(rcvd_end - rcvd_beg) sum(sent_end - sent_beg +) as bandwidth, sum(duration_end - durat from</pre>	as traffic_out, as traffic_in, rcvd_end - rcvd_beg	
<pre>devid, vd, min(coalesce(sentby (coalesce(rcvdbyte, 0)) as rcv ation, 0)) as duration_beg, ma and subtype='vpn' and action=' coalesce(nullifna(`user`), ips</pre>	<pre>lesce(nullifna(`user`), ipstr(`remip`)) as user_s te, 0)) as sent_beg, max(coalesce(sentbyte, 0)) a d_beg, max(coalesce(rcvdbyte, 0)) as rcvd_end, mi x(coalesce(duration, 0)) as duration_end from \$1c tunnel-stats' and tunneltype in ('ssl-tunnel', 's tr(`remip`)) is not null and tunnelid is not null yd /*SkipSTART*/order by tunnelid/*SkipEND*/)###</pre>	as sent_end, min n(coalesce(dur- og where \$filter ssl') and 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(duration, 0)) as duration_beg, max(coalesce(duration, 0)) as duration_end from \$log where \$filter and subtype='vpn' and action='tunnel-stats' and tunneltype='ssl-web' and coalesce(nullifna (`user`), ipstr(`remip`)) is not null and tunnelid is not null group by tunnelid, user_src, remip, devid, vd /*SkipSTART*/order by tunnelid/*SkipEND*/)### t group by user_src, remote_ip, tunnelid, devid, vd having sum(sent_end-sent_beg+rcvd_end-rcvd_beg)>0 order by bandwidth desc) t where bandwidth>0 group by user_src, remote_ip order by bandwidth desc

```
Dataset Name
                                    Description
                                                                                      Log Category
                                    Event admin login summary
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
<pre>select \$flex_timescale(timestamp) as sum(total_num) as total_num, sum(total_change) as total_cl</pre>		
from	nin) og hatal som over (som fig skapas) og h	tatal abanan fuam
<pre>###(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)</pre>		

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

Dataset Name	Description	Log Category
Admin-Failed-Login-Summary	Event admin failed login summary	event
select		
`user` as f_user,		
ui,		
count(status) as total_failed		
from		
\$log		
where		
\$filter		
and nullifna(`user`) is not n	ull	
and logid_to_int(logid) = 320	02	
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
```

sum(high) as high, sum(medium) as medium

from

###({{FGT_DATASET_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
<pre>select \$flex_timescale(timestamp) sum(critical) as critical,</pre>	as dom,	

from

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

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

select

```
$flex_timescale(timestamp) as dom,
sum(critical) as critical,
sum(high) as high,
sum(medium) as medium
From
```

from

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

Dataset Name	Description	Log Category
System-Critical-Severity-Events	Event system critical severity events	event
select		
msg_desc as msg,		
severity_tmp as severity,		
sum(count) as counts		

```
from
```

###({{FGT_DATASET_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
<pre>select msg_desc as msg,</pre>		

```
severity_tmp as severity,
  sum(count) as counts
  from
```

###({{FGT_DATASET_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
<pre>select msg_desc as msg, severity_tmp as severity, sum(count) as counts</pre>		
from	EM EVENTS}})### t where severity tmp='Mediu	

severity tmp order by counts desc

Dataset Name	Description	Log Category	
utm-drilldown-Top-Traffic-Summary	UTM drilldown traffic summary	traffic	
<pre>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</pre>			
Dataset Name	Description	Log Category	
utm-drilldown-Top-User-Destination	UTM drilldown top user destination	traffic	
select appid, app,			

app, dstip, sum(sessions) as sessions, sum(bandwidth) as bandwidth

from

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

Dataset Name	Description	Log Category
utm-drilldown-Email-Senders- Summary	UTM drilldown email senders summary	traffic
<pre>select sum(requests) as requests,</pre>		

sum(bandwidth) as bandwidth from

###({{FGT_DATASET_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_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
<pre>select recipient,</pre>		

sum(bandwidth) as bandwidth

```
from
```

###({{FGT_DATASET_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
	EMAIL_SENDERS}})### t where \$filter-drilldown a sum(bandwidth)>0 order by bandwidth desc	and sender is
Dataset Name	Description	Log Category
Dataset Name utm-drilldown-Top-Allowed-Websites- By-Bandwidth	Description UTM drilldown top allowed web sites by bandwidth	Log Category traffic
utm-drilldown-Top-Allowed-Websites-		

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

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

```
select
 appid,
 hostname,
 sum(requests) as requests
```

from

###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user src, 0 as appid, hostname, (case when action='blocked' then 1 else 0 end) as blocked, count(*) as requests from \$log where \$filter and (eventtype is null or logver>=502000000) and hostname is not null group by user src, appid, hostname, blocked order by requests desc)### t where \$filter-drilldown and blocked=1 group by appid, hostname order by requests desc

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

select

```
virus,
```

```
sum(totalnum) as totalnum
```

```
from
```

###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user src, virus, count(*) as totalnum from \$log where \$filter and (eventtype is null or logver>=502000000) and nullifna(virus) is not null group by user src, virus order by totalnum desc)### t where \$filter-drilldown group by virus order by totalnum desc

Dataset Name	Description	Log Category
utm-drilldown-Top-Attacks	UTM drilldown top attacks by name	attack
attack_count from \$log where \$fil	nt ser`), ipstr(`srcip`)) as user_src, attack, cou ter and nullifna(attack) is not null group by u)### t where \$filter-drilldown group by attack	user_src,

Dataset Name	Description	Log Category
utm-drilldown-Top-Vulnerability	UTM drilldown top vulnerability by name	netscan
select vuln, sum(totalnum) as totalnum		
<pre>from ###(select coalesce(nullifna</pre>	(`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 De	escription	Log Category
utm-drilldown-Top-App-By-Bandwidth U	TM drilldown top applications by bandwidth usage	traffic

select

```
appid,
app,
```

sum(bandwidth) as bandwidth

from

###(select appid, app, appcat, apprisk, sum(bandwidth) as bandwidth, sum(sessions) as sessions from {{FGT_DATASET_BASE_TRAFFIC_TOP_APPS}} t group by appid, app, appcat, apprisk /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t where \$filter-drilldown group by appid, app having sum(bandwidth)>0 order by bandwidth desc

Dataset Name	Description	Log Category
utm-drilldown-Top-App-By-Sessions	UTM drilldown top applications by session count	traffic
<pre>select appid, app, sum(sessions) as sessions</pre>		
<pre>from ###(select appid, app, appcat, sions from {{FGT_DATASET_BASE_TR</pre>	<pre>apprisk, sum(bandwidth) as bandwidth, sum(se AFFIC_TOP_APPS}} t group by appid, app, appca esc, bandwidth desc/*SkipEND*/)### t where \$f</pre>	t, apprisk

group by appid, app order by sessions desc

 Dataset Name
 Description
 Log Categor

Dataset Name	Description	Log Category
Top5-Users-By-Bandwidth	UTM drilldown top users by bandwidth usage	traffic
<pre>select coalesce(nullifna(`user`), nullifna(`unauthuse</pre>	r`),	
<pre>ipstr(`srcip`)) as dldn_user, count(*) as session, sum(</pre>		
<pre>coalesce(sentbyte,) as bandwidth, sum(</pre>	0)+ coalesce(rcvdbyte, 0)	
<pre>coalesce(sentbyte,) as traffic_out, sum(</pre>	0)	
<pre>coalesce(rcvdbyte,) as traffic_in from \$log</pre>	0)	
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

Dataset Name	Description	Log Category
bandwidth-app-Top-App-By Bandwidth-Sessions	Top applications by bandwidth usage	traffic
<pre>) as bandwidth, sum(coalesce(rcvdbyte,) as traffic_in, sum(coalesce(sentbyte,) as traffic_out, count(*) as sessions from \$log where \$filter and (logflag&1>0) and nullifna(app) is group by app_group having sum(</pre>	<pre>0)+ coalesce(rcvdbyte, 0) 0) 0)</pre>	
bandwidth desc		

Dataset Name

Dataset Name	Description	Log Category
bandwidth-app-Category-By- Bandwidth	Application risk application usage by category	traffic
select appcat, sum(bandwidth) as bandwidth		
	apprisk, sum(bandwidth) as bandwidth, sum(sess. FFIC_TOP_APPS}} t group by appid, app, appcat,	

/*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t where \$filter-drilldown
and nullifna(appcat) is not null group by appcat order by bandwidth desc

bandwidth-app-Top-Users-By- Bandwidth-Sessions Bandwidth application top users by bandwidth usage traffic select coalesce(nullifna('user'), nullifna('user'), ipstr('srcip')) as user_src, sum(coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(sentbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out, count(*) as sessions Image: Second S	Dataset Name	Description	Log Category
<pre>coalesce(nullifna('user'), nullifna('uauthuser'), 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&l>0) group by user_src having sum(coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)) & gt; 0 order by bandwidth desc </pre>		Bandwidth application top users by bandwidth usage	traffic
Dataset Name Description Log Category	<pre>coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentbyte, 0)+ coal) 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)+ coal)& gt; 0 order by</pre>		
	Dataset Name	Description	Log Category

bandwidth-app-Traffic-By-Active-User- Bandwidth application traffic by active user number traffic
Number

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
<pre>select coalesce(nullifna(root_domain(hostname)), ipstr(`dstip`)) as domain, sum(coalesce(sentbyte, 0)+ coaless) 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</pre>	ce(rcvdbyte, 0)	
Dataset Name	Description	Log Category
bandwidth-app-Top-Policies-By- Bandwidth-Sessions	Top policies by bandwidth and sessions	traffic
<pre>select coalesce(pol.name, cast(policyid as text)) as polid, sum(bandwidth) as bandwidth, sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_out sum(sessions) as sessions from</pre>	,	

from

###(select policyid, poluuid, 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 policyid, poluuid order by bandwidth desc)### t1 left join \$ADOMTBL_PLHD_POLINFO pol on t1.poluuid=pol.uuid group by polid order by bandwidth desc

Dataset Name	Description	Log Category
bandwidth-app-Traffic-Statistics	Bandwidth application traffic statistics	traffic
drop		
<pre>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),</pre>	5),	
<pre>total_dest varchar(255)); insert into rpt_tmptbl_1 total_sessions, total_bandw ave_session, ave_bandwidth</pre>	width,	
)		
<pre>select format_numeric_no_decimal(sum(sessions)</pre>		
) as total sessions,		
bandwidth unit(
sum (bandwidth)		
) as total_bandwidth,		
format_numeric_no_decimal(
cast(
<pre>sum(sessions)/ \$days_num</pre>	as decimal(18, 0)	
)		
) as ave_session,		
bandwidth_unit(
cast(
sum(bandwidth)/ \$days_nur	m as decimal(18, 0)	
)		
) as ave_bandwidth		
from		
	ons, sum(coalesce(sentbyte, 0)+coalesce(rcvd	-
	and (logflag&1>0))### t; update rpt_tmptbl_1	
	(sessions) as sessions from ###(select \$DAY_	- —
-	where \$filter and (logflag&1>0) group by do	_
	by sessions desc limit 1) as t1; update rpt_ format numeric no decimal(count(distinct(us	
<pre>num from ###(select coalesce(n)</pre>	ullifna(`user`), nullifna(`unauthuser`), ips	str(`srcip`)) as
—	om \$log where \$filter and (logflag&1>0) grou	—
	<pre>date rpt_tmptbl_1 set total_app=t3.totalnum nct(app_grp))) as totalnum from ###(select a</pre>	
as app_grp, count(*) as count :	from \$log where \$filter and (logflag&1>0) an	d nullifna(app) is
	r by count desc)### t) as t3; update rpt_tmp	
	<pre>mat_numeric_no_decimal(count(distinct(dstip))</pre>	
	count from \$log where \$filter and (logflag&1	
	ount desc)### t) as t4; select 'Total Sessi	
—	pt_tmptbl_1 union all select 'Total Bytes Tr	
—	from rpt_tmptbl_1 union all select 'Most Ac	
	as stats from rpt_tmptbl_1 union all select from rpt tmptbl 1 union all select 'Total Ap	
Junundry, LUCAL_USELS as Stats .	TTOW THE CUMPONT I MITON ATT PETECE TODAT AD	"PIICACIONS as SUNT

summary, total_users as stats from rpt_tmptbl_1 union all select 'Total Applications' as summary, total_app as stats from rpt_tmptbl_1 union all select 'Total Destinations' as summary, total_dest as stats from rpt_tmptbl_1 union all select 'Average Sessions Per Day' as summary, ave_session as stats from rpt_tmptbl_1 union all select 'Average Bytes Per Day' as summary, ave_bandwidth as stats from rpt_tmptbl_1

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

select

```
$flex_timescale(timestamp) as hodex,
sum(scores) as scores
from
```

###({{FGT_DATASET_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_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(
<pre>nullifna(`user`), nullifna(`unauthuser`),</pre>		
ipstr(`srcip`)		
) as user src,		
sum(crscore % 65536) as scores		
from		
\$log		
vhere \$filter		
and (
logflag&1>0		
)		
and crscore is not null		
group by		
user_src naving		
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
  get_devtype(srcswversion, osname, devtype) as devtype_new,
  coalesce(
  nullifna(`srcname`),
   nullifna(`srcmac`),
   ipstr(`srcip`)
 ) as dev src,
  sum(crscore % 65536) as scores
from
  $log
where
  $filter
 and (
  logflag&1>0
  )
  and crscore is not null
group by
 devtype_new,
 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
<pre>select f_user, sum(sum_rp_score) as sum_rp_sc from ###(select coalesce(nullifna((crscore%65536) as sum_rp_score crscore is not null group by f_user; create tempor sum_rp_score from ###(select coalesce(cip`)) as f_user, sum(crscore%65) and crscore is not null group by desc)### t group by f_user; sele (t2.sum_rp_score) as t2_sum_score</pre>	`user`), nullifna(`unauthuser`), ipstr(`sro from \$log where \$pre_period \$filter and (1 user having sum(crscore%65536)>0 order by s rary table rpt_tmptbl_2 as select f_user, s alesce(nullifna(`user`), nullifna(`unauthus 5536) as sum_rp_score from \$log where \$filt y f_user having sum(crscore%65536)>0 order ect t1.f_user, sum(t1.sum_rp_score) as t1_s re, (sum(t2.sum_rp_score)-sum(t1.sum_rp_sco ot_tmptbl_2 as t2 on t1.f_user=t2.f_user wh	<pre>logflag&1>0) and sum_rp_score desc)### sum(sum_rp_score) as ser`), ipstr(`sr- cer and (logflag&1>0) by sum_rp_score sum_score, sum ore)) as delta from</pre>
Dataset Name	Description	Log Category

```
      Dataset Name
      Description
      Log Category

      Top-Devices-With-Increased-Scores
      Reputation top devices with increased scores
      traffic

      drop
      table if exists rpt_tmptbl_1;
      drop
      table if exists rpt_tmptbl_2; create temporary table rpt_tmptbl_1 as
```

```
select
  f_device,
  devtype_new,
  sum(sum_rp_score) as sum_rp_score
from
```

###(select coalesce(nullifna(`srcname`),nullifna(`srcmac`), ipstr(`srcip`)) as f_device, get_devtype(srcswversion, osname, devtype) as devtype_new, sum(crscore%65536) as sum_rp_score from \$log where \$pre_period \$filter and (logflag&1>0) and crscore is not null group by f_ device, devtype_new having sum(crscore%65536)>0 order by sum_rp_score desc)### t group by f_ device, devtype_new; create temporary table rpt_tmptbl_2 as select f_device, devtype_new, sum (sum_rp_score) as sum_rp_score from ###(select coalesce(nullifna(`srcname`),nullifna(`srcmac`), ipstr(`srcip`)) as f_device, get_devtype(srcswversion, osname, devtype) as devtype_ new, sum(crscore%65536) as sum_rp_score from \$log where \$filter and (logflag&1>0) and crscore is not null group by f_device, devtype_new having sum(crscore%65536)>0 order by sum_rp_score desc)### t group by f_device, devtype_new; select t1.f_device, t1.devtype_new , sum(t1.sum_rp_ score) as t1_sum_score, sum(t2.sum_rp_score) as t2_sum_score, (sum(t2.sum_rp_score)-sum (t1.sum_rp_score)) as delta from rpt_tmptbl_1 as t1 inner join rpt_tmptbl_2 as t2 on t1.f_ device=t2.f_device and t1.devtype_new=t2.devtype_new where t2.sum_rp_score > t1.sum_rp_score group by t1.f_device, t1.devtype_new order by delta desc

Dataset Name	Description	Log Category
Attacks-By-Severity	Threat attacks by severity	attack
_	critical' then 'Critical' when severity = Medium' when severity = 'low' then 'Low' w	
Dataset Name	Description	Log Category
Top-Attacks-Detected	Threat top attacks detected	attack
<pre>select attack, attackid, cve, severity, sum(attack_count) as att from</pre>	ack_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

Description	Log Category
Threat top attacks blocked	attack
ot null ted', 'pass_session')	
	ot null

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>=50200000) 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
<pre>select \$flex_timescale(timestamp) sum(totalnum) as totalnum from</pre>	as hodex,	
	as timestamp, count(*) as totalnum	
by timestamp /*SkipSTART*/ord	er by timestamp desc/*SkipEND*/)###	t group by hodex order by

hodex

Dataset Name	Description	Log Category
Virus-Time-Line	Threat virus timeline	virus
<pre>select \$flex_datetime(timestam sum(totalnum) as totaln from ###(select \$flex_timest</pre>	-	n from \$log where \$filter and

(eventtype is null or logver>=502000000) and nullifna(virus) is not null group by timestamp /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t group by hodex order by hodex

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

NGOR

```
user_src,
sum(totalnum) as totalnum
```

from

###({{FGT_DATASET_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
<pre>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</pre>		

Dataset Name	Description	Log Category
Spyware-Time-Line	Threat spyware timeline	virus
<pre>select \$flex_timescale(timestamp) sum(totalnum) as totalnum</pre>	as hodex,	

```
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_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
<pre>select virus, max(virusid_s) as virusid, sum(totalnum) as totalnum</pre>		

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
<pre>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</pre>		

Dataset Name	Description	Log Category
Adware-Time-Line	Threat adware timeline	virus
<pre>select \$flex_timescale(timestamp)</pre>	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
<pre>select \$flex_timescale(timestamp) as 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</pre>	timescale, timestamp, sum(case when severity = 'criti	cal' then 1 else 0
<pre>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</pre>		

Dataset Name	Description	Log Category
Important-Intrusions-Timeline-By- Severity	Threat intrusions timeline by severity	attack
end) as critical, sum(case when severity = 'medium' then 1 else else 0 end) as low, sum(case wh	s timestamp, sum(case when severity = h severity = 'high' then 1 else 0 end) e 0 end) as medium, sum(case when seve hen severity = 'info' or severity = 'd r group by timestamp /*SkipSTART*/orde	as high, sum(case when rity = 'notice' then 1 ebug' then 1 else 0 end)
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		
<pre>and t1.severity = 'critical' and nullifna(attack) is not nul</pre>	11	
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 tl.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 tl.severity = 'medium'		
and nullifna(attack) is not nul	1	
group by		
attack,		
vuln_type,		
cve		
order by		
totalnum desc		
Dataset Name	Description	Log Category
Top-Intrusion-Victims	Threat top intrusion victims	attack

select
victim,
sum(cri_num) as critical,
sum(high_num) as high,
sum(med_num) as medium,
sum(cri_num + high_num + med_num) as totalnum
from

###(select dstip as victim, sum((case when severity='critical' then 1 else 0 end)) as cri_ num, sum(case when severity='high' then 1 else 0 end) as high_num, sum(case when severity='medium' then 1 else 0 end) as med_num from \$log where \$filter and severity in ('critical', 'high', 'medium') group by victim)### t group by victim order by totalnum desc

Dataset Name	Description	Log Category
Top-Intrusion-Sources	Threat top intrusion sources	attack
<pre>select source, sum(cri_num) as critical, sum(high num) as high,</pre>		

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 tl.severity = 'medium' then 'Medium' when tl.severity = 'low' then 'Low' when tl.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,		
(
-	'critical' then 'Critical' when t1.severity	
	then 'Medium' when tl.severity = 'low' then	n 'Low' when tl.severity
= 'info' then 'Info' end		
) as severity_name,		
count(*) as totalnum,		
vuln_type, (
-	'critical' then 0 when t1.severity = 'high	' then 1 when t1 -
	when t1.severity = 'low' then 3 when t1.sever	
else 5 end		
) as severity_number		
from		
\$log t1		
left join (
select		
name,		
cve,		
vuln_type		
from		
ips_mdata		
) t2 on t1.attack = t2.nam	16	
where \$filter		
and nullifna(attack) is no	t ull	
and action in ('detected',		
group by		
attack,		
attackid,		
tl.severity,		
vuln_type		
order by		
severity_number,		
totalnum desc		

Dataset Name	Description	Log Category
Attacks-Over-HTTP-HTTPs	Threat attacks over HTTP HTTPs	attack
<pre>severity = 'medium' then 'Me 'Info' end) as severity, count(*) as totalnum, (case when severity = 'cn</pre>	-	n severity = 'info' then 1 when severity =
Dataset Name	Description	Log Category
default-AP-Detection-Summary-b Status-OffWire	y- Default access point detection summary by st wire	atus off- event
<pre>'others' end) as ap_full_status, count(*) as totalnum from (select apstatus, bssid, ssid from ###(select apstatus, k apstatus is not null and aps (logid) in (43527, 43521, 43 43583, 43584, 43585) group k</pre>	en 'rogue' when 2 then 'accepted' when 3 th ossid, ssid, count(*) as subtotal from \$log status!=0 and bssid is not null and onwire= 3525, 43563, 43564, 43565, 43566, 43569, 43 by apstatus, bssid, ssid order by subtotal oup by ap_full_status order by totalnum des	where \$filter and 'no' and logid_to_int 570, 43571, 43582, desc)### t group by

Dataset Name	Description	Log Category
default-AP-Detection-Summary-by- Status-OffWire_table	Default access point detection summary by status off- wire	event
elect		
(
case apstatus when 1 then 'r	cogue' when 2 then 'accepted' when 3 then 'suppre	essed' else
others' end		
) as ap_full_status,		
count(*) as totalnum		
from		
(
select		
apstatus,		
bssid,		
ssid		
from		
###(select apstatus, bssic	d, ssid, count(*) as subtotal from \$log where \$fi	ilter and
pstatus is not null and apstatu	us!=0 and bssid is not null and onwire='no' and l	logid_to_int

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
<pre>'others' end) as ap_full_status,</pre>	ogue' when 2 then 'accepted' when 3 then 'suppre	ssed' else
<pre>count(*) as totalnum from (select</pre>		
apstatus, bssid, ssid		
from ###(select apstatus, bssid apstatus is not null and apstatu (logid) in (43527, 43521, 43525, 43583, 43584, 43585) group by ap	, ssid, count(*) as subtotal from \$log where \$fi s!=0 and bssid is not null and onwire='yes' and 43563, 43564, 43565, 43566, 43569, 43570, 43571 status, bssid, ssid order by subtotal desc)### t by ap full status order by totalnum desc	logid_to_int , 43582,

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 43565 43566 43566 43560 43570 43571 43582
```

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, ssid order by totalnum desc

Dataset Name	Description	Log Category
default-Managed-AP-Summary	Default managed access point summary	event
<pre>select (case when (action like '%join%' and logid_to_int(logid) in) then 'Authorized' else 'Un) as ap_status, count(*) as totalnum from \$log where \$filter and logid_to_int(logid) in (43 group by ap_status order by totalnum desc</pre>	authorized' end	

Dataset Name	Description	Log Category
default-Managed-AP-Summary_table	Default managed access point summary	event
<pre>select (case when (action like '%join%' and logid_to_int(logid) in) then 'Authorized' else 'Unau) as ap_status, count(*) as totalnum from \$log where \$filter and logid_to_int(logid) in (4352)</pre>	thorized' end	

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
rom
```

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
  $loq
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 (nclassified' when 1 then 'rogue' when 2 the	
<pre>radioband, from_dtime(min(dtime)) as first_seen, from_dtime(max(dtime)) as last_seen, detectionmethod,</pre>		

itime,
onwire as on wire
from
\$log
where
\$filter
and apstatus is not null
and bssid is not null
and onwire = 'yes'
and logid_to_int(logid) in (
43521, 43563, 43564, 43565, 43566, 43569,
43570, 43571
)
group by
ap_full_status,
devid,
vd,
ssid,
bssid,
manuf,
rssi,
channel,
radioband,
detectionmethod,
itime,
onwire,
apstatus

Dataset Name Description Log Category event-Wireless-Client-Details Event wireless client details event drop table if exists rpt_tmptbl_1; create temporary table rpt_tmptbl_1 as select ip, lmac, sn, ssid, channel, radioband, min(first) as first, max(last) as last from ###(select ip, lower(mac) as lmac, sn, ssid, channel, radioband, min(dtime) as first, max (dtime) as last from \$log-event where \$filter and ip is not null and mac is not null and sn is not null and ssid is not null group by ip, lmac, sn, ssid, channel, radioband order by ip)### t group by ip, lmac, sn, ssid, channel, radioband; select user src, ip, lmac, sn, ssid, channel, radioband, from_dtime(first) as first_seen, from_dtime(last) as last_seen, cast(volume as decimal(18,2)) as bandwidth from (select * from rpt tmptbl 1 inner join (select user src, srcip, sum(volume) as volume from ###(select coalesce(nullifna(`user`), nullifna(`unauthuser), ipstr(`srcip`)) as user src, srcip, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as volume from \$log-traffic where \$filter-time and (logflag&1>0) and srcip is not null group by user src, srcip having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by volume desc) ### t group by user src, srcip order by user src, srcip) t on rpt tmptbl 1.ip =

t.srcip) t order by volume desc

Dataset Name	Description	Log Category
event-Wireless-Accepted-Offwire	Event wireless accepted off-wire	event
select		
'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		
	ESS_ROGUE_OFFWIRE}}) ### t where apstatu:	

```
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		
	<pre>SS_ROGUE_ONWIRE}})### t where apstatus manuf, channel, radioband, detectionm</pre>	

```
by last_seen desc
```

Dataset Name	Description	Log Category
event-Wireless-Rogue-Offwire	Event wireless rogue off-wire	event
<pre>select 'rogue' as ap_full_status, devid, vd,</pre>		

```
ssid,
bssid,
manuf,
channel,
radioband,
from_dtime(
    max(last_seen)
) as last_seen,
detectionmethod,
snclosest,
'no' as on_wire
from
###(//FCT_DATASET
```

```
###({{FGT_DATASET_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 EVENT WIRE	LESS ROGUE ONWIRE}})### t where apstatu	us=1 and onwire='ves'

###({{FGT_DATASET_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
<pre>select 'suppressed' as ap_full_status, devid, vd, ssid, bssid, manuf, channel, radioband, from_dtime(max(last_seen)) as last_seen,</pre>		

```
detectionmethod,
  snclosest,
  'no' as on_wire
  from
```

###({{FGT_DATASET_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
<pre>select 'suppressed' as ap_full_status, devid, vd, ssid, bssid,</pre>		
<pre>manuf, channel, radioband, from_dtime(max(last_seen)</pre>		
) as last_seen, detectionmethod, snclosest, 'yes' as on wire		
from ###({{FGT_DATASET_EVENT_WIRELES	S_ROGUE_ONWIRE}})### t where apstatus=3 and onw manuf, channel, radioband, detectionmethod, snc	

```
by last_seen desc
```

Dataset Name	Description	Log Category
event-Wireless-Unclassified-Offwire	Event wireless unclassified off-wire	event
select		
'unclassified' as ap_full_statu	lS,	
devid,		
vd,		
ssid, bssid,		
manuf,		
channel,		
radioband,		
from_dtime(
<pre>max(last_seen)</pre>		
) as last_seen,		
detectionmethod,		
snclosest,		
'no' as on_wire from		
###({{FGT_DATASET_EVENT_WIRELES	SS_ROGUE_OFFWIRE}})### t where apstatus=0 and manuf, channel, radioband, detectionmethod,	

event
)st

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
<pre>string_agg(distinct user_ag remip, tunnelid, min(s_time) as s_time, max(e_time) as e_time, (</pre>	er_agg, ' ') as xauthuser_agg, g, ' ') as user_agg, x (e time) then max(max traffic in)+ max(max tra	ffic out) else
	fic_in) + max(max_traffic_out) - min(min_traffic_	_

```
(
    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
    ###(((ECT DATAGET EVENT VDN DIAL UD IDEEC DANDWIDTU)))### t group by devid ud remin
```

###({{FGT_DATASET_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
<pre>select remip as remote_ip, sum(bandwidth) as bandwidth from (select devid, vd, remip, tunnelid, (case when min(s_time) = max min(min_traffic_in) end) as traffic_in, (case when min(s_time) = max out) - min(min_traffic_out) end) as traffic_out, (case when min(s_time) = max </pre>	<pre>x(e_time) then max(max_traffic_in) else max(max x(e_time) then max(max_traffic_out) else max(ma x(e_time) then max(max_traffic_in)+ max(max_tra fic_in)+ max(max_traffic_out) - min(min_traffic_</pre>	x_traffic_ ffic_out) else
	_TRAFFIC_USAGE}})### t where t_type like 'ssl%' oup by remote_ip having sum(traffic_in+traffic_	

Dataset Name	Description		Log Category
webfilter-Web-Activity-Summary-By- Requests	Webfilter web activity summa	ary by requests	webfilter
select \$flex_timescale(timestamp) as h sum(allowed_request) as allowed sum(blocked_request) as blocked	_request,		
<pre>from ###(select Sfley timestamp as t</pre>	imestamo sum (case when	action = 'blocked' the	n 1 else () end)

```
###(select $flex_timestamp as timestamp, sum(case when action!='blocked' then 1 else 0 end)
as allowed_request, sum(case when action='blocked' then 1 else 0 end) as blocked_request from
```

\$log where \$filter and (eventtype is null or logver>=502000000) group by timestamp
/*SkipSTART*/order by timestamp desc/*SkipEND*/)### t group by hodex order by hodex

Dataset Name	Description	Log Category		
traffic-Browsing-Time-Summary	Traffic browsing time summary	traffic		
<pre>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</pre>				
Dataset Name	Description	Log Category		
traffic-Browsing-Time-Summary- Enhanced	Traffic browsing time summary enhanced	traffic		
<pre>select \$flex_timescale(timestamp) as h</pre>	odex,			

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

```
select
user_src,
sum(requests) as requests
from
  ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, count(*) as requests from
$log where $filter and (eventtype is null or logver>=502000000) and coalesce(nullifna(`user`),
ipstr(`srcip`)) is not null and action='blocked' group by user_src /*SkipSTART*/order by
```

requests desc/*SkipEND*/)### t group by user_src order by requests desc

Dataset NameDescriptionLog Categorywebfilter-Top-Web-Users-By-Allowed-
RequestsWebfilter top web users by allowed requestswebfilter

Requests

```
select
  user_src,
  sum(requests) as requests
from
```

sum(traffic_out) as traffic_out

###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, count(*) as requests from \$log where \$filter and (eventype is null or logver>=502000000) and coalesce(nullifna(`user`), ipstr(`srcip`)) is not null and action!='blocked' group by user_src /*SkipSTART*/order by requests desc/*SkipEND*/)### t group by user_src order by requests desc

Dataset Name	Description	Log Category
traffic-Top-Web-Users-By-Browsing- Time	Traffic top web users by browsing time	traffic
<pre>select user_src, ebtr_value(ebtr_agg_flat(browsetime), null, \$timespan) as browsetime, sum(bandwidth) as bandwidth, sum(traffic in) as traffic in,</pre>		

from

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

Dataset Name	Description	Log Category
webfilter-Top-Blocked-Web-Sites-By- Requests	Webfilter top blocked web sites by requests	webfilter
(eventtype is null or logver>=50 action='blocked' group by domain	quests cct main, tdesc, m(requests) as requests	
Dataset Name	Description	Log Category

	Description	
webfilter-Top-Allowed-Web-Sites-By- Requests	Webfilter top allowed web sites by requests	webfilter
<pre>select domain, string_agg(distinct catdesc, ',</pre>	') 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>=50200000) and hostname is not null and catdesc is not null and action!='blocked' group by domain, catdesc /*SkipSTART*/order by requests desc/*SkipEND*/)### t group by domain order by requests desc

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

select

```
domain,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out
from
```

###(select coalesce(nullifna(root_domain(hostname)), 'other') as domain, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum (coalesce(sentbyte, 0)) as traffic_out from \$log-traffic where \$filter and (logflag&l>0) and (countweb>0 or ((logver is null or logver<50200000) and (hostname is not null or utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter')))) and catdesc in ('Streaming Media and Download') group by domain having sum(coalesce(sentbyte, 0)+coalesce (rcvdbyte, 0))>0 /*SkipSTART*/order by bandwidth desc/*SkipEND*/)### t group by domain order by bandwidth desc

Dataset Name	Description	Log Category
webfilter-Top-Blocked-Web-Categories	Webfilter top blocked web categories	webfilter
null or logver>=502000000) and ca	requests from \$log-webfilter where \$filter and tdesc is not null and action='blocked' group by sc/*SkipEND*/)### t group by catdesc order by r	catdesc
Dataset Name	Description	Log Category
webfilter-Top-Allowed-Web-Categories	Webfilter top allowed web categories	webfilter
null or logver>=502000000) and ca	requests from \$log-webfilter where \$filter and tdesc is not null and action!='blocked' group b sc/*SkipEND*/)### t group by catdesc order by r	y catdesc
Dataset Name	Description	Log Category
traffic-Top-50-Sites-By-Browsing-Time		

```
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
###//(from paradorm moderned mode armon py pp mumple))#
```

###({{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(
<pre>ebtr_agg_flat(browsetime),</pre>		
null,		
\$timespan		
) as browsetime,		
sum(bandwidth) as bandwidth,		
<pre>sum(traffic_in) as traffic_in,</pre>		
<pre>sum(traffic_out) as traffic_out</pre>		
from		
###({{FGT_DATASET_TRAFFIC_TOP_D	ST_COUNTRY_BY_EB_TIME}})### t group by dstcount	ry 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
<pre>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_D browsetime desc</pre>	DST_COUNTRY_BY_EB_TIME}})### t group by dstcount	cry order by

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
<pre>select user_src, ebtr_value(ebtr_agg_flat(browsetime), null, \$timespan) as browsetime from ####({{FGT_DATASET_TRAFFIC_EB_TIMESTATASET_TTAFFIC_EB_TIMESTATASET_TTAFFIC_EB_TIMESTATASET_TTAFFIC_EB_TIMESTATASET_TTAFFIC_EB_TIMESTATASET_TTAFFIC_EB_TIMESTATASET_TTAFFIC_EB_TIMESTATASET_TTAFFIC_EB_TIMESTATASET_TTAFFIC_EB_TIMESTATASET_TTAFFIC_EB_TIMESTATASET_TTAFFIC_EB_TIMESTATASET_TTAFFIC_EB_TIMESTATASET_TTAFFIC_EB_TIMESTATASET_TTAFFIC_EB_TIMESTATASET_TTAFFIC_EB_TIMESTATASET_TTAFFIC_EB_TIMESTATASET_TTAFFIC_EB_TIMESTATASET_TTAFFIC_EB_TIMESTATASET_TTAFFIC_EB_TIMESTATASET_TTAFFIC_EB_TIMESTATASET_TTAFFIC_</pre>	ME}})### t group by user_src order by browsetime	e desc

Dataset Name	Description	Log Category
Top-10-Users-Browsing-Time- Enhanced	Estimated browsing time enhanced	traffic
<pre>select user_src, ebtr_value(ebtr_agg_flat(browsetime), null, \$timespan</pre>		

```
) 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
<pre>select user_src, ebtr_value(ebtr_agg_flat(browsetime), null, \$timespan) as browsetime from ###({{FGT_DATASET_TRAFFIC_EB_TIMESTATESTATESTATESTATESTATESTATESTATESTA</pre>	ME}})### t group by user_src order by browsetime	e desc

Dataset Name	Description	Log Category
Estimated-Browsing-Time-Enhanced	Estimated browsing time enhanced	traffic
<pre>select user_src, ebtr_value(ebtr_agg_flat(browsetime), null, \$timespan) as browsetime from ###({{FGT_DATASET_TRAFFIC_EB_T</pre>	IME}})### 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
<pre>sum(coalesce(sentbyte, 0)+ coale) 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</pre>	sce(rcvdbyte, 0)	
<pre>sum(coalesce(sentbyte, 0)+ coale) & gt; 0 order by</pre>	sce(rcvdbyte, 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, srcswversion, get_devtype
(srcswversion, osname, devtype) as devtype_new, 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, srcswversion, devtype_new, 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
<pre>select srcssid, sum(coalesce(sentbyte, 0)+ cd) as bandwidth from \$log where \$filter and (logflag&1>0)</pre>	balesce(rcvdbyte, 0)	
<pre>and srcssid is not null group by srcssid having sum(coalesce(sentbyte, 0)+ co)& gt; 0 order by bandwidth desc</pre>	balesce(rcvdbyte, 0)	

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
<pre>select appid, app, sum(coalesce(sentbyte, 0)+ coalesce)) as bandwidth from \$log where \$filter and (logflag&1>0) and (srcssid is not null</pre>	ce(rcvdbyte, 0)	

```
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') || ' (' || get_devtype(srcswversion, osname, devtype) || ', ' || coalesce(osname, '') || (case when srcswversion is null then '' else ' ' || srcswversion end) || ')') as client, sum(coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)) as bandwidth from \$loq 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

Defect News	Description	
Dataset Name	Description	Log Category
wifi-Top-OS-By-Bandwidth	Top WiFi os by bandwidth usage	traffic
<pre>select (coalesce(osname, 'unknown')) as os, sum(</pre>	' ' coalesce(srcswversion, '')	

```
FortiAnalyzer 6.4.4 Dataset Reference
Fortinet Technologies Inc.
```

```
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

<pre>wifi-Top-Device-By-Bandwidth Top WiFi device by bandwidth usage traffic select get_devtype(srcswversion, osname, devtype) as devtype_new, sum(coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)) as bandwidth from \$log where \$filter and (logflag&1>0)</pre>
<pre>get_devtype(srcswversion, osname, devtype) as devtype_new, sum(coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)) as bandwidth from \$log where \$filter and (</pre>
and (srcssid is not null or dstssid is not null) and devtype is not null

```
devtype_new
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
<pre>select devtype_new, count(distinct srcmac) as</pre>	totalnum	

from

###(select srcintf, srcssid, osname, srcswversion, get_devtype(srcswversion, osname, devtype) as devtype_new, srcmac, count(*) as subtotal from \$log where \$filter and (logflag&l>0) and (srcssid is not null or dstssid is not null) and srcmac is not null group by srcintf, srcssid, osname, srcswversion, devtype, srcmac order by subtotal desc)### t where devtype_new is not null group by devtype new order by totalnum desc

Dataset Name	Description	Log Category
wifi-Overall-Traffic	WiFi overall traffic	traffic
select		
sum (
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, srcswversion, get_devtype(srcswversion, osname, devtype) as devtype_new, 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, srcswversion, devtype_new, srcmac order by subtotal desc)### t

Dataset Name	Description	Log Category
Top30-Subnets-by-Bandwidth-and- Sessions	Top subnets by application bandwidth	traffic
<pre>select ip_subnet(`srcip`) as subnet, sum(coalesce(sentbyte, 0)+ coale) 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 naving sum(coalesce(sentbyte, 0)+ coale) & gt; 0 order by bandwidth desc</pre>		
Dataset Name	Description	Log Category

Dataset Name

	Decemption	_09 0 0 0 9 0 1 9
Top30-Subnets-by-Application- Bandwidth	Top applications by bandwidth	traffic
<pre>select ip_subnet(`srcip`) as subnet app_group_name(app) as app_g sum(coalesce(sentbyte, 0)+ coa) as bandwidth from \$log where \$filter and (logflag&1>0) and nullifna(app) is not nul group by subnet, app_group having</pre>	roup, lesce(rcvdbyte, 0)	

```
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
<pre>select ip_subnet(`srcip`) as subnet app_group_name(app) as app_g count(*) as sessions from \$log where \$filter and (logflag&1>0) and nullifna(app) is not nul group by subnet, app_group order by sessions desc</pre>	roup,	
Dataset Name	Description	Log Category

Top30-Subnets-by-Website-Bandwidth	Top websites and web category by bandwidth	traffic
<pre>coalesce(rcvdbyte, 0)) as bandwidd and (logflag&1>0) and (countweb>0 not null or utmevent in ('webfilte</pre>	s subnet, hostname as website, sum(coalesce(th from \$log-traffic where \$filter and hostn or ((logver is null or logver<502000000) an er', 'banned-word', 'web-content', 'command- ite order by bandwidth desc)### t group by s	ame is not null d (hostname is block', 'script-

Dataset Name	Description	Log Category
Top30-Subnets-by-Website-Hits	Top websites and web category by sessions	webfilter
<pre>select subnet, website, sum(hits) as hits from ###(select ip subnet(`srcip`) a</pre>	as subnet, hostname as website, count(*) as hit	s from \$log

where \$filter and hostname is not null and (eventtype is null or logver>=502000000) group by subnet, website order by hits desc)### t group by subnet, website order by hits desc

Dataset Name	Description	Log Category
Top30-Subnets-with-Top10-User-by- Bandwidth	Top users by bandwidth	traffic
<pre>select ip_subnet(`srcip`) as subnet, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentbyte, 0)+ coales) 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)+ coales) & gt; 0 order by bandwidth desc</pre>		

Dataset	Name

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		ige traffic
<pre>select appcat, app, sum(coalesce(sentbyte,) as bandwidth from \$log where \$filter and (logflag&1>0) group by appcat, app</pre>	0)+ coalesce(rcvdbyte, 0)	
having sum(0)+ coalesce(rcvdbyte, 0)	

Dataset Name

Dataset Name	Description	Log Category
app-Top-20-Category-and- Applications-by-Session	Top category and applications by session	traffic
<pre>select appcat, app, count(*) as sessions from \$log where \$filter and (logflag&1>0) group by appcat, app order by sessions desc</pre>		

Dataset Name	Description	Log Category
app-Top-500-Allowed-Applications-by- Bandwidth	Top allowed applications by bandwidth usage	traffic
select		
<pre>from_itime(itime) as timestamp</pre>	1	
coalesce(
<pre>nullifna(`user`),</pre>		
<pre>nullifna(`unauthuser`), insta(`unauthuser`)</pre>		
ipstr(`srcip`)		
) as user_src,		
appcat,		
app, coalesce(
root domain(hostname),		
ipstr(dstip)		
) as destination,		
sum (
<pre>coalesce(`sentbyte`, 0)+ coa</pre>	lesce(`rcvdbyte`, 0)	
) as bandwidth		
from		
\$log		
where		
\$filter		
and (
logflag&1>0		
)		
and action in ('accept', 'clos	e', 'limeoul')	
group by timestamp,		
user_src,		
appcat,		
app,		
destination		
order by		
bandwidth desc		
Dataset Name	Description	Log Category
ann Tan 500 Plackad Applications by	Top blocked applications by session	troffic

app-Top-500-Blocked-Applications-by- SessionTop blocked applications by sessiontrafficselect coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, appcat, app, count(*) as sessions*********************************		•		
<pre>coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, appcat, app, count(*) as sessions from \$log where</pre>		Top blocked applications by session	traffic	
	<pre>coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, appcat, app, count(*) as sessions from \$log where</pre>			

```
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
<pre>select from_dtime(dtime) as timestamp, catdesc.</pre>		

```
catdesc,
hostname as website,
status,
sum(bandwidth) as bandwidth
from
```

###(select dtime, catdesc, hostname, cast(utmaction as text) as status, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from \$log-traffic where \$filter and hostname is not null and (logflag&1>0) and (countweb>0 or ((logver is null or logver<50200000) 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-
                                    Web hourly category and website hits action
                                                                                        webfilter
Hits-Action
select
 hod,
 website,
  sum(hits) as hits
from
  ###(select $hour_of_day as hod, (hostname || ' (' || coalesce(`catdesc`, 'Unknown') || ')')
as website , count(*) as hits from $log where $filter and hostname is not null and (eventtype
is null or logver>=502000000) group by hod, website order by hod, hits desc) ### t group by
hod, website order by hod, hits desc
Dataset Name
                                    Description
                                                                                        Log Category
                                                                                        traffic
web-Top-20-Category-and-Websites-
                                    Web top category and websites by bandwidth usage
by-Bandwidth
select
 website,
  catdesc,
  sum (bandwidth) as bandwidth
from
```

###(select hostname as website, catdesc, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from \$log-traffic where \$filter and hostname is not null and (logflag&1>0) and (countweb>0 or ((logver is null or logver<50200000) and (hostname is not null or utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter')))) group by website, catdesc order by bandwidth desc)### t group by website, catdesc order by bandwidth desc

Dataset Name	Description	Log Category
web-Top-20-Category-and-Websites- by-Session	Web top category and websites by session	webfilter
<pre>select website, catdesc, sum(sessions) as hits from ###({{FGT_DATASET_WEBFILTER_TOP desc</pre>	?_WEBSITES}})### t group by website, ca	atdesc order by hits
Dataset Name	Description	Log Category
web-Top-500-Website-Sessions-by-	Web top website sessions by bandwidth usage	e traffic

. Bandwidth

select

```
from_dtime(dtime) as timestamp,
user_src,
website,
catdesc,
cast(
   sum(dura)/ 60 as decimal(18, 2)
) as dura,
sum(bandwidth) as bandwidth
```

from

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

Dataset Name	Description	Log Category
web-Top-500-User-Visted-Websites- by-Bandwidth	Web top user visted websites by bandwidth usage	traffic
select website, catdesc, sum(bandwidth) as bandwidth		
from		
bandwidth from \$log-traffic where	catdesc, sum(coalesce(sentbyte, 0)+coalesce(r \$filter and hostname is not null and (logflag ver<502000000) and (hostname is not null or ut	£1>0) and (coun-

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		
<pre>from ###({{FGT_DATASET_WEBFILTER_TOP</pre>	<pre>P_WEBSITES}})### t group by website,</pre>	catdesc order by sessions

desc

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

select

```
subtype,
count(distinct fctuid) as totalnum
```

from

###({{FCT_DATASET_EP_STATUS}})### t where subtype is not null group by subtype order by totalnum desc

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

select

```
os_short as os,
count(distinct fctuid) as totalnum
```

from

###({{FCT_DATASET_EP_STATUS}})### t where os_short is not null group by os order by totalnum
desc

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

select

fctver as fctver_short,

```
count(distinct fctuid) as totalnum
```

from

```
###({{FCT_DATASET_EP_STATUS}})### t where fctver is not null group by fctver order by total-
num desc
```

Dataset Name	Description	Log Category
fct-Endpoint-Profile-Deployment	Endpoint Profile Deployment	fct-event
select profile,		
count(distinct fctuid) as to	talnum	

from

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

Dataset Name	Description	Log Category
fct-Client-Summary	Client Summary	fct-event
select		
hostname,		
deviceip,		
os short as os,		
profile,		
fctver,		
from_itime(
max(itime)		
) as last_seen		
from		
	regexp_replace(os, '\\(build.*', '') as os_sh max(itime) as itime from Slog where Sfilter	

(usingpolicy) as profile, fctver, max(itime) as itime from \$log where \$filter and os is not null group by hostname, deviceip, os_short, profile, fctver order by itime desc)### t group by hostname, deviceip, os, profile, fctver order by last seen desc

Dataset NameDescriptionLog Categoryfct-Total-Threats-FoundTotal Threats Foundfct-traffic

select

utmevent_s as utmevent, count(distinct threat) as totalnum from

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

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

```
threat,
sum(totalnum) as totalnum
from
 (
    (
    select
    threat,
    sum(totalnum) as totalnum
    from
```

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

Dataset Name	Description	Log Category
fct-Top10-Infected-Devices-with- Botnet	Top Infected Devices with Botnet	fct-traffic
<pre>select hostname, count(*) as totalnum from \$log where \$filter and hostname is not null and lower(utmevent) in ('web and lower(threat) like '%bot group by hostname order by totalnum desc</pre>		
Dataset Name	Description	Log Category
fct-Top10-Infected-Devices-with-Viru	s- Top Infected Devices with Virus Malware	fct-traffic

```
Malware
```

```
select
hostname,
sum(totalnum) as totalnum
from
(
    (
    select
    hostname,
    sum(totalnum) as totalnum
from
```

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

Dataset Name	Description	Log Category
fct-All-Antivirus-Antimalware- Detections	All Antivirus and Antimalware Detections	fct-traffic
<pre>select threat, hostname, hostuser, utmaction, from_dtime(max(dtime)) as last_seen</pre>		

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

Dataset Name	Description	Log Category
fct-Web-Filter-Violations	Web Filter Violations	fct-traffic
<pre>select hostuser, hostname, string_agg(distinct remotename, utmaction, sum(total) as totalnum, from_dtime(max(dtime)) as last scorp</pre>	',') as remotename,	
) as last_seen from		
tion, count(*) as total, max(dtim	<pre>, coalesce(nullifna(`user`), 'Unknown') as host e) as dtime from \$log where \$filter and lower(u ed' group by remetorance bestrance bestrance</pre>	itmevent) -

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 hostuser, hostname, utmaction order by totalnum desc

Dataset Name	Description	Log Category
fct-Application-Firewall	Application Firewall	fct-traffic
<pre>select threat, hostname, hostuser, utmaction, from_dtime(max(dtime)) as last_seen</pre>		
	ame, coalesce(nullifna(`user`), 'Unknow	

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

Dataset Name	Description	Log Category
fct-Errors-and-Alerts	Errors and Alerts	fct-event
<pre>select msg, hostname, hostuser, from_dtime(max(dtime)) as last seen</pre>		
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
<pre>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</pre>		
Dataset Name	Description	Log Category
fct-vuln-Device-Vulnerabilities	Vulnerabilities Detected by User/Device	fct-netscan
<pre>select 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 severity_number, count(distinct vulnname) as vuln_num from ### (select vulnseverity, devid, vulnname from \$log where \$filter and nullifna(vulnseverity) is not null and nullifna(vulnname) is not null group by vulnseverity, vulnname, devid)### t group by vulnseverity order by severity_number desc</pre>		

Dataset Name	Description	Log Category
fct-vuln-Category-Type-Vulnerabilities	Vulnerabilities Detected by Category Type	fct-netscan
<pre>select vulncat, count(distinct vulnname) as tot</pre>	calnum	
<pre>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</pre>		

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_number desc

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

###(select vulnseverity, devid, vulnname from \$log where \$filter and nullifna(vulnseverity)
is not null and nullifna(vulnname) is not null and nullifna(devid) is not null group by
vulnseverity, vulnname, devid)### t group by vulnseverity order by dev_num desc, severity_number desc

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

select

\$flex_timescale(timestamp) as hodex, count(distinct vulnname) as total_num

from

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

Dataset Name	Description	Log Category
fct-vuln-Details-by-Risk-Level-Device	Vulnerability Details for Each Risk Level by Device	fct-netscan
<pre>select hostname, os,</pre>		
<pre>vulnseverity, count(distinct vulnname) as vuln_num, count(distinct products) as products,</pre>		
<pre>count(distinct cve_id) as cve_c from ###(select hostname, os, vulnna</pre>	ount me, vulnseverity, vulnid from \$log where \$filt;	er and vulnname

is not null and vulnseverity is not null and hostname is not null group by hostname, os, vulnname, vulnseverity, vulnid)### t1 left join fct_mdata t2 on t1.vulnid=t2.vid::int group by hostname, os, vulnseverity order by vuln_num desc, hostname

Dataset Name	Description	Log Category
fct-vuln-Details-by-Device-User	Vulnerability Details by Device User	fct-netscan
select		
hostname,		
(
' <div>' vulnname '<!--</td--><td>'div>'</td><td></td></div>	'div>'	
) as vulnname,		
vulnseverity,		
vulncat,		
string_agg(distinct products	s, ',') as products,	
string_agg(distinct cve_id,	',') as cve_list,	
(
	DISTINCT vendor_link, ',') '>Remediation	n Info'
) as vendor_link		
from		
###(select hostname, vulnname, vulnseverity, vulncat, vulnid from \$log where \$filter and vul-		
	e is not null group by hostname, vulnname,	-
	ndata 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
<pre>select hostname, ('<div>' vulnname '<!--) as vulnname, vulnseverity, string_agg(distinct vendor_l</pre--></div></pre>		
	e, vulnseverity, vulnid from \$log where \$filte	

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/>'
) 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 vulnid from \$log where \$filter group by vulnid)### t1 inner join fct_mdata t2 on t2.vid=t1.vulnid::text inner join fct_cve_score t3 on strpos(t2.cve_id, t3.cve_id) > 0 group by t3.cve_id, score order by score desc, t3.cve_id

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

select
fgtserial,

```
count (distinct fctuid) as totalnum
```

from

###({{FCT_DATASET_EP_STATUS}})### t where fgtserial is not null group by fgtserial order by totalnum desc

Dataset Name	Description	Log Category
fct-Top-Malware-Detections	Top Infected Devices with Malware	fct-traffic
<pre>select hostname, fctuid, sum(totalnum) as totalnum from ((select hostname, fctuid,</pre>		
sum(totalnum) as totalnum		
tion, max(dtime) as dtime, uid as	<pre>me, coalesce(nullifna(`user`), 'Unki fctuid, count(*) as totalnum from s 'antimalware') group by threat bosi</pre>	\$log where \$filter and

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

Dataset Name	Description	Log Category
fct-Top10-Malware-Detections	Top 10 Infected Devices with Malware	fct-traffic
select		
threat,		
hostname,		
hostuser,		
utmaction,		
fctuid,		
sum(totalnum) as totalnum		
from		
(
(
select		
threat,		
hostname,		
hostuser,		
utmaction,		
fctuid,		
sum(totalnum) as totalnum from		
	<pre>me, coalesce(nullifna(`user`), 'Un</pre>	aknown!) as hostusor
### (Serect threat, hostha	me, coaresce (nurrillia (user), "of	iknown / as nostuser,

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

Dataset Name	Description	Log Category
fct-Devices-with-Botnet	Infected Devices with Botnet	fct-traffic
select		
threat,		
hostname,		
coalesce(
<pre>nullifna(`user`),</pre>		
'Unknown'		
) as hostuser,		
utmaction,		
uid as fctuid,		
count(*) as totalnum		
from \$log		
\$⊥09 where		
\$filter		
and hostname is not null		
and lower (utmevent) in ('we	bfilter'. 'appfirewall')	
and lower(threat) like '%bo		
group by		
threat,		
hostname,		
hostuser,		
utmaction,		
fctuid		
order by		
totalnum desc		
Dataset Name	Description	Log Category
fct-vuln-Vulnerability-by-Hostname	Vulnerability Details for Each Risk Level by Device	fct-netscan

```
select
```

```
hostname,
os,
vulnseverity,
count(distinct vulnname) as vuln_num,
count(distinct products) as products,
count(distinct cve_id) as cve_count
from
  ####(select hostname, os, vulnname, vulnseverity, vulnid from $log where $filter and vulnname
is not null and vulnseverity is not null and hostname is not null group by hostname, os, 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-Users-With-Web-Violations	Web Filter Violations	fct-traffic
<pre>select hostuser, hostname, string_agg(distinct remotename utmaction, sum(total) as totalnum, from_dtime(max(dtime)) as last_seen</pre>	, ',') as remotename,	
	e, coalesce(nullifna(`user`), 'Unknown' me) as dtime from \$log where \$filter ar	

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 hostuser, hostname, utmaction order by totalnum desc

Dataset Name	Description	Log Category
fct-Compliance-by-FortiGate	FortiClinet Compliance by FortiGate Enforcing	fct-event
select		
fgtserial,		
count(distinct fctuid) as to	talnum	
from		
(
select		
fgtserial,		
fctuid,		
<pre>max(compliance_flag) as</pre>	compliance_flag	
from		
###({{FCT_DATASET_EP_STA	TUS}})### tt group by fgtserial, fctuid) t when	re compliance_flag
= 1 group by fgtserial order b	y totalnum desc	

Dataset Name	Description	Log Category
fct-Compliance-Status	Number of FortiClinets by Compliance Status	fct-event
<pre>) as compliance, count(distinct fctuid) as total from (select fctuid, max(compliance_flag) as con from</pre>		e order by

Dataset Name	Description	Log Category
fct-Non-Compliant-Endpoints	Non-compliant Endpoints	fct-event
select		
tl.fgtserial,		
t3.srcintf,		
t2.epname as hostname,		
t2.mac,		
'Non-Compliant' as status		
from		
(
select		
fqtserial,		
fctuid,		
max(compliance flag) as co	mpliance flag	
from	1 5	
###({{FCT_DATASET_EP_STATU	JS}})### tt group by fgtserial, fctu	- —

ENDPOINT t2 on t1.fctuid = t2.fctuid left join \$ADOM_EPEU_DEVMAP t3 on t2.epid = t3.epid where compliance_flag = 0 group by t1.fctuid, t1.fgtserial, t3.srcintf, t2.epname, t2.mac

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

```
select
```

```
$flex_timescale(timestamp) as hodex,
sum(requests) as requests
```

from

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

Description	Log Category
Fop Visited Web Categories	fct-traffic
	•

select

```
category,
sum(requests) as requests
from
```

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

Dataset Name	Description	Log Category
fct-Traffic-Top-Allowed-Website	Top Visited Websites	fct-traffic
<pre>select website, string_agg(distinct category, sum(requests) as requests from</pre>	', ') as agg_category,	
<pre>###(select fct_webcat(threat)</pre>	as category, remotename as website, count(*)	as requests from

\$log where \$filter and direction='outbound' and threat is not null and utmaction='passthrough'
and lower(utmevent)='webfilter' group by category, website order by requests desc)### t where
website is not null group by website order by requests desc

Dataset Name	Description	Log Category
fct-Traffic-Top-Category-By-Website- Session	Top Web Categories by Website Session	fct-traffic
<pre>select fct_webcat(threat) as category, remotename as website, count(*) as requests from \$log where \$filter and nullifna(threat) is not nul and lower(utmevent)= 'webfilter group by category, website order by requests desc</pre>		
Dataset Name	Description	Log Category
fct-Traffic-Top-Web-Users-By-Website	Top Web Users by Website	fct-traffic
<pre>select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, remotename as website, count(*) as requests from \$log where \$filter and direction = 'outbound' and remotename is not null and utmaction = 'passthrough' and lower(utmevent) = 'webfilter group by user_src, website order by requests desc</pre>	,	
	Description	Log Category
Dataset Name		

```
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
<pre>user_src, dstip, srcintf, dstint: coalesce(rcvdbyte, 0)) as bandwid group by appid, app, user_src, ds</pre>	e(nullifna(`user`), nullifna(`unauthuser`), 5, policyid, count(*) as sessions, sum(coale dth from \$log where \$filter-exclude-var and stip, srcintf, dstintf, policyid order by se ifna(app) is not null group by appid, app or	esce(sentbyte, 0)+- (logflag&1>0) essions desc)### t
Dataset Name	Description	Log Category

	•	
drilldown-Top-App-By-Bandwidth- Table	Drilldown top applications by bandwidth usage	traffic
select appid,		

```
app,
sum(bandwidth) as bandwidth
```

from

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

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

select appid, app, sum(bandwidth) as bandwidth from

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

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

select

dstip, sum(sessions) as sessions from

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

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

dstip,

sum (bandwidth) as bandwidth

from

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

Dataset Name	Description	Log Category
drilldown-Top-User-By-Sessions-Table	Drilldown top user by session count	traffic
<pre>select user_src, sum(sessions) as sessions </pre>		
user_src, dstip, srcintf, dstintf	(nullifna(`user`), nullifna(`unauthuso , policyid, count(*) as sessions, sum th from \$log where \$filter-exclude-va:	(coalesce(sentbyte, 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

```
Log Category
 Dataset Name
                                   Description
 drilldown-Top-User-By-Bandwidth-
                                   Drilldown top user by bandwidth usage
                                                                                     traffic
 Table
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,
    our (no muchts)
```

```
sum(requests) as visits
from
```

(

###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, hostname, count(*) as requests from \$log-traffic where \$filter-exclude-var and (logflag&1>0) and utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') and hostname is not null group by user_src, hostname order by requests desc)### union all ### (select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, hostname, count(*) as requests from \$log-webfilter where \$filter-exclude-var and (eventtype is null or logver>=502000000) and hostname is not null group by user_src, hostname order by requests desc)###) t where \$filterdrilldown 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&l>0) and utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') and hostname is not null group by user_src, hostname order by requests desc)### union all ### (select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, hostname, count(*) as requests from \$log-webfilter where \$filter-exclude-var and (eventtype is null or logver>=502000000) and hostname is not null group by user_src, hostname order by requests desc)###) t where \$filterdrilldown 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		
(
<pre>###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user src,</pre>		
hostname, count(*) as requests from \$log-traffic where \$filter-exclude-var and (logflag&1>0)		
and utmevent in ('webfilter', 'k	oanned-word', 'web-content', 'command-b	lock', 'script-filter')

and utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') and hostname is not null group by user_src, hostname order by requests desc)### union all ### (select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, hostname, count(*) as requests from \$log-webfilter where \$filter-exclude-var and (eventtype is null or logver>=502000000) and hostname is not null group by user_src, hostname order by requests desc)###) t where \$filterdrilldown and hostname is not null group by hostname order by visits desc

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

```
select
```

```
hostname,
sum(requests) as visits
from
```

```
(
```

###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, hostname, count(*) as requests from \$log-traffic where \$filter-exclude-var and (logflag&1>0) and utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') and hostname is not null group by user_src, hostname order by requests desc)### union all ### (select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, hostname, count(*) as requests from \$log-webfilter where \$filter-exclude-var and (eventtype is null or logver>=502000000) and hostname is not null group by user_src, hostname order by requests desc)###) t where \$filterdrilldown 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&l>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-excludevar 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,	<pre>count(*) as requests, sum(coalesce(sentbyte, 0</pre>)+coalesce
(rcvdbyte, 0)) as bandwidth from	<pre>\$log-traffic where \$filter-exclude-var and (log</pre>	flag&1>0) and
service in ('smtp', 'SMTP', '25/t	.cp', '587/tcp', 'smtps', 'SMTPS', '465/tcp') an	d utmevent in
	') group by sender, recipient order by requests nder, `to` as recipient, count(*) as requests,	

(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from \$log-emailfilter where \$filter-excludevar 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-
                                   Drilldown top email send recipient by count
                                                                                    traffic
By-Count
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 $loq-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-
                                   Drilldown top email receiver by volume
                                                                                    traffic
Volume
select
 recipient,
 sum (bandwidth) as volume
from
    ###(select recipient, sender, count(*) as requests, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth from $log where $filter-exclude-var and (logflag&1>0) and service
```

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

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

select
 sender,
 sum(bandwidth) as volume
from
 (

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

Dataset Name	Description	Log Category
drilldown-Top-Email-Recipient-By- Count	Drilldown top email receiver by count	traffic
<pre>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&l>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</pre>		
Dataset Name	Description	Log Category

drilldown-Top-Email-Receive-Sender-	Drilldown top email receive sender by count	traffic
By-Count		

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

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

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

dstip, sum(totalnum) as totalnum from

###(select srcip, dstip, count(*) as totalnum from \$log where \$filter-exclude-var group by srcip, dstip order by totalnum desc)### t where \$filter-drilldown and dstip is not null group by dstip order by totalnum desc

Dataset Name	Description	Log Category	
drilldown-Top-Attack-Source	Drilldown top attack source	attack	
<pre>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</pre>			
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
```

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

```
select
```

```
from_itime(itime) as timestamp,
virus,
user_src,
dstip,
hostname,
recipient
```

```
from
```

###(select itime, virus, coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, dstip, cast(' ' as char) as hostname, cast(' ' as char) as recipient from \$log where \$filter and (eventtype is null or logver>=502000000) and nullifna(virus) is not null order by itime desc)### t where \$filter-drilldown order by timestamp desc

```
Dataset Name
                                     Description
                                                                                        Log Category
 user-drilldown-Top-Blocked-Web-Sites- User drilldown top blocked web sites by requests
                                                                                        webfilter
 By-Requests
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
                                                                                        Log Category
                                     Description
 user-drilldown-Top-Allowed-Web-Sites- User drilldown top allowed web sites by requests
                                                                                        webfilter
 By-Requests
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
<pre>select catdesc, sum(requests) as requests</pre>		

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(virusid, eventtype) as virusid_s, count(*) as totalnum from \$log where \$filter and nullifna (virus) is not null group by user_src, virus, virusid_s order by totalnum desc)### t where \$filter-drilldown group by virus order by totalnum desc

Dataset Name	Description	Log Category
user-drilldown-Top-Virus-Receivers- Over-Email	User drilldown top virus receivers over email	virus
<pre>(*) as totalnum from \$log where \$i 'SMTP', '25/tcp', '587/tcp', 'smtp '110/tcp', 'imap', 'IMAP', '143/tc '995/tcp')) and nullifna(virus) is</pre>	Ser`), ipstr(`srcip`)) as user_src, `to` as rec filter and subtype='infected' and (service in o os', 'SMTPS', '465/tcp') or service in ('pop3', cp', 'imaps', 'IMAPS', '993/tcp', 'pop3s', 'POF s not null group by user_src, receiver order by n group by receiver order by totalnum desc	('smtp', 'POP3', 23S',

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

select

\$hour_of_day(timestamp) as hourstamp, sum(totalnum) as totalnum from

###({{FGT_DATASET_EMAILFILTER_USER_SENDER}})### t where \$filter-drilldown and mf_receiver is not null and action in ('detected', 'blocked') group by hourstamp order by hourstamp

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

select

```
mf_sender,
sum(totalnum) as totalnum
```

from

###({{FGT_DATASET_EMAILFILTER_USER_SENDER}})### t where \$filter-drilldown and mf_sender is not null and action in ('detected', 'blocked') group by mf_sender order by totalnum desc

Dataset Name	Description	Log

Event usage CPU

```
event-Usage-CPU
```

```
select
  $hour_of_day(timestamp) as hourstamp,
  cast(
    sum(total_cpu) / sum(count) as decimal(6, 2)
 ) as cpu_avg_usage
from
  ###(({FGT_DATASET_EVENT_DISK_LOGRATE_CPU_MEM}))### t group by hourstamp order by hourstamp
```

Category

event

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

select

```
$hour_of_day(timestamp) as hourstamp,
cast(
   sum(total_mem) / sum(count) as decimal(6, 2)
) as mem avg usage
```

from

###({{FGT_DATASET_EVENT_DISK_LOGRATE_CPU_MEM}})### t group by hourstamp order by hourstamp

Dataset Name	Description	Log Category
event-Usage-Sessions	Event usage sessions	event
<pre>select \$hour_of_day(timestamp) as hourstamp, cast(sum(totalsession) / sum(count) as decimal(10, 2)) as sess_avg_usage</pre>		

###({{FGT_DATASET_EVENT_DISK_LOGRATE_CPU_MEM}})### t group by hourstamp order by hourstamp

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

select

```
$hour_of_day(timestamp) as hourstamp,
cast(
    sum(totalsession) / sum(count) as decimal(10, 2)
) as sess_avg_usage,
cast(
    sum(total_cpu) / sum(count) as decimal(6, 2)
) as cpu_avg_usage
from
```

###({{FGT_DATASET_EVENT_DISK_LOGRATE_CPU_MEM}})### t group by hourstamp order by hourstamp

Dataset Name	Description	Log Category
App-Risk-Top-Users-By-Bandwidth	Top users by bandwidth usage	traffic
<pre>select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, srcip, sum(coalesce(sentbyte, 0)+ coales) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in,</pre>	ce(rcvdbyte, 0)	

```
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
<pre>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</pre>		
Dataset Name	Description	Log Category

Dataset Name	Description	Log Category
App-Risk-Top-Users-By-Reputation- Scores-Bar	Application risk reputation top users by scores	traffic
<pre>select coalesce(nullifna(`user`),</pre>		

```
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 traffic App-Risk-Top-Devices-By-Reputation-Application risk reputation top devices by scores Scores select get devtype (srcswversion, osname, devtype) as devtype new, coalesce(nullifna(`srcname`), nullifna(`srcmac`), ipstr(`srcip`)) as dev_src, sum(crscore % 65536) as scores from \$log where \$filter and (logflag&1>0) and crscore is not null group by devtype new, 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(bandwidth) as bandwidth		

from

###(select appid, app, appcat, apprisk, sum(bandwidth) as bandwidth, sum(sessions) as sessions from {{FGT_DATASET_BASE_TRAFFIC_TOP_APPS}} t group by appid, app, appcat, apprisk /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t where \$filter-drilldown and nullifna(appcat) is not null group by appcat order by bandwidth desc

Dataset Name	Description	Log Category
App-Risk-App-Usage-by-Category	Application risk application usage by category	traffic

select

appcat,

sum(bandwidth) as bandwidth

from

###(select appid, app, appcat, apprisk, sum(bandwidth) as bandwidth, sum(sessions) as sessions from {{FGT_DATASET_BASE_TRAFFIC_TOP_APPS}} t group by appid, app, appcat, apprisk /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t where \$filter-drilldown and nullifna(appcat) is not null group by appcat 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 \$logtraffic where \$filter and (logflag&1>0) and (countweb>0 or ((logver is null or logver<502000000) and (hostname is not null or utmevent in ('webfilter', 'banned-word', 'webcontent', '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
<pre>select app_group_name(app) as app_grou appcat, sum(coalesce(sentbyte, 0)+ coales) as bandwidth, count(*) as num_session</pre>		
from		
\$log where		
<pre>\$filter and (logflag&1>0)</pre>		
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
<pre>select app_group_name(app) as app_group service, count(*) as sessions, sum(coalesce(sentbyte, 0)+ coalese) as bandwidth from \$log where \$filter and (logflag&1>0) and nullifna(app) is not null and service in ('80/tcp', '443/tcp', 'HTTP', 'http', 'https')</pre>	sce(rcvdbyte, 0)	
<pre>group by app_group, service having sum(coalesce(sentbyte, 0)+ coales) & gt; 0 order by bandwidth desc</pre>	sce(rcvdbyte, 0)	

Dataset Name

Description

Log Category

```
App-Risk-Top-Web-Sites-Visited-By-<br/>Network-Users-Pie-ChaApplication risk web browsing summary category
```

select
 catdesc,
 sum(num_sess) as num_sess,
 sum(bandwidth) as bandwidth
from

###(select catdesc, count(*) as num_sess, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))
as bandwidth from \$log-traffic where \$filter and (logflag&1>0) and (countweb>0 or ((logver is
null or logver<502000000) and (hostname is not null or utmevent in ('webfilter', 'bannedword', '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</pre>

Dataset Name	Description	Log Category
App-Risk-Top-Web-Sites-Visited-By- Network-Users	Application risk web browsing summary category	traffic
<pre>select catdesc, sum(num_sess) as num_sess, sum(bandwidth) as bandwidth from ###(select catdesc, count(*) a</pre>	as num sess, sum(coalesce(sentbyte, 0)+coalesce(r	
as bandwidth from \$log-traffic w hull or logver<502000000) and (h word', 'web-content', 'command-b	where \$filter and (logflag&1>0) and (countweb>0 on nostname is not null or utmevent in ('webfilter', plock', 'script-filter')))) and catdesc is not nu ### t group by catdesc order by num_sess desc	r ((logver is 'banned-
as bandwidth from \$log-traffic w hull or logver<502000000) and (h yord', 'web-content', 'command-b	where \$filter and (logflag&1>0) and (countweb>0 o nostname is not null or utmevent in ('webfilter', plock', 'script-filter')))) and catdesc is not nu	r ((logver is 'banned-
as bandwidth from \$log-traffic w hull or logver<502000000) and (h word', 'web-content', 'command-b catdesc order by num_sess desc)#	where \$filter and (logflag&1>0) and (countweb>0 on hostname is not null or utmevent in ('webfilter', block', 'script-filter'))) and catdesc is not nu ### t group by catdesc order by num_sess desc	r ((logver is 'banned- ll group by Log Category

```
from
```

###(select coalesce(nullifna(hostname), ipstr(`dstip`)) as domain, catdesc, count(*) as visits from \$log where \$filter and (eventtype is null or logver>=502000000) and catdesc is not null group by domain, catdesc order by visits desc)### t group by domain, catdesc order by visits desc

	Description	Log Category
Top-Destination-Countries-By- Browsing-Time	Traffic top destination countries by browsing time	traffic
<pre>delect 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_I prowsetime desc</pre>	E DST_COUNTRY_BY_EB_TIME}})### t group by dstcount	try order by

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

```
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,		
<pre>sum(traffic_in) as traffic_in,</pre>		
<pre>sum(traffic_out) as traffic_out</pre>		
from		
###({{FGT_DATASET_TRAFFIC_TOP_D	OMAINS_BY_EB_TIME}})### t group by h	nostname order by brow-

setime desc

	Description	Log Category
App-Risk-Traffic-Top-Hostnames-By- Browsing-Time-Enhanced	Traffic top domains by browsing time enhanced	traffic
<pre>elect 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 rom</pre>		

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
<pre>select attack, severity, ref, count(*) as totalnum from</pre>		
<pre>\$log where \$filter and severity = 'critical' ord multifue (attack) is not multi-</pre>		
<pre>and nullifna(attack) is not nul group by attack, severity, ref order by totalnum desc</pre>	Ţ	

Dataset Name	Description	Log Category
App-Risk-Top-High-Threat-Vectors- Crossing-The-Network	Application risk top high threat vectors	attack
<pre>select attack, severity, ref, count(*) as totalnum</pre>		
from \$log		
<pre>where \$filter and severity = 'high' and nullifna(attack) is not nu</pre>	11	
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
<pre>select attack, severity, ref, count(*) as totalnum from \$log where \$filter and severity = 'medium' and nullifna(attack) is not null group by attack,</pre>	L	
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
<pre>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</pre>	1	

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
<pre>select virus, max(virusid_s) as virusid, (case when virus like 'Ris else 'Virus' end) as malware_type, sum(totalnum) as totalnum</pre>	kware%' then 'Spyware' y	when virus like 'Adware	%' then 'Adware'
from			

###(select virus, virusid_to_str(virusid, eventtype) as virusid_s, count(*) as totalnum from \$log where \$filter and (eventtype is null or logver>=502000000) and nullifna(virus) is not null group by virus, virusid_s /*SkipSTART*/order by totalnum desc/*SkipEND*/)### t group by virus, malware type order by totalnum desc

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

```
select
```

```
user_src,
sum(totalnum) as totalnum
from
###(calcot calcocc(cullif)
```

###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, count(*) as totalnum from \$log where \$filter and (eventype is null or logver>=502000000) and nullifna(virus) is not null group by user_src /*SkipSTART*/order by totalnum desc/*SkipEND*/)### t group by user_src order by totalnum desc

Dataset Name	Description	Log Category
App-Risk-Data-Loss-Prevention-Type- Events	Application risk DLP UTM event	dlp
<pre>select subtype : :text as utmsubtype,</pre>		

```
subtype : :text as utmsubtype,
count(*) as number
from
```

###({{FGT_DATASET_DLP_VIOLATION_SUMMARY}})### t where \$filter-drilldown and subtype is not null group by subtype order by number desc

Dataset Name	Description	Log Category
App-Risk-Vulnerability-Discovered	Application risk vulnerability discovered	netscan
<pre>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</pre>		

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

select
 dom,
 sum(totalnum) as totalnum

from

###(select \$DAY_OF_MONTH as dom, count(*) as totalnum from \$log where \$filter and nullifna
(virus) is not null and (eventtype is null or logver>=502000000) group by dom order by totalnum desc)### t group by dom order by totalnum desc

Dataset Name	Description	Log Category
App-Risk-Breakdown-Of-Risk- Applications	Application risk breakdown of risk applications	traffic
<pre>select unnest(string_to_array(behavior, ') as d_behavior, count(*) as number from \$log t1 inner join app_mdata t2 on t1 where \$filter and (logflag&1>0)</pre>		

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
<pre>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.a; where \$filter and (logflag&1>0) group by risk, f_behavior order by risk desc, number desc</pre>		

Dataset Name	Description	
--------------	-------------	--

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)+ coal	esce(rcvdbyte, 0)	
) as bandwidth,		
count(*) as sessions		
from		
\$log t1		
inner join app_mdata t2 on t3	appid = t2.id	
where		
\$filter		
and (
logflag&1>0		
) and behavior is not null		
group by		

Log Category

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
<pre>select appcat, count(distinct app) as total_r from</pre>	um	

###(select appid, app, appcat, apprisk, sum(bandwidth) as bandwidth, sum(sessions) as sessions from {{FGT_DATASET_BASE_TRAFFIC_TOP_APPS}} t group by appid, app, appcat, apprisk /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t where \$filter-drilldown and nullifna(appcat) is not null and apprisk in ('critical', 'high') group by appcat order by total_num desc

Dataset Name	Description	Log Category
Apprisk-Ctrl-Threats-Prevention	Threat Prevention	app-ctrl

select

```
threat_name,
count(distinct threats) as total_num
from
  (
```

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

Dataset Name	Description	Log Category
Apprisk-Ctrl-Application-Vulnerability	Application vulnerabilities discovered	attack
select		
attack,		
attackid,		
vuln_type,		
cve,		
severity_number,		
count(distinct dstip) as victir	ns,	
count(distinct srcip) as source	es,	
sum(totalnum) as totalnum		
from		
<pre>###(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</pre>		

totalnum from \$log t1 left join (select name, cve, vuln_type from ips_mdata) t2 on t1.attack=t2.name where \$filter and nullifna(attack) is not null and t1.severity is not null group by attack, attackid, vuln_type, t2.cve, t1.severity, dstip, srcip)### t group by attack, attackid, vuln_type, severity_number, cve order by severity_number desc, totalnum desc

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

count(distinct app) as total_num

```
from
```

###(select appid, app, appcat, apprisk, sum(bandwidth) as bandwidth, sum(sessions) as sessions from {{FGT_DATASET_BASE_TRAFFIC_TOP_APPS}} t group by appid, app, appcat, apprisk /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t where \$filter-drilldown and nullifna(appcat) is not null and apprisk in ('critical', 'high') group by appcat order by total_num desc

Dataset Name	Description	Log Category
Apprisk-Ctrl-Top-20-High-Risk- Application	Application risk high risk application	traffic

select

```
risk as d_risk,
count(distinct user_src) as users,
id,
name,
app_cat,
technology,
sum(bandwidth) as bandwidth,
sum(sessions) as sessions
```

from

###(select 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
<pre>select behavior, round(sum(total_num)* 100 / sum(sum(total_num)) over (), 2) as percentage from (</pre>		

###(select (case when lower(appcat)='botnet' then 'malicious' when lower(appcat)-='remote.access' then 'tunneling' when lower(appcat) in ('storage.backup', 'video/audio') then 'bandwidth-consuming' when lower(appcat)='p2p' then 'peer-to-peer' when lower(appcat)='proxy' then 'proxy' end) as behavior, sum(sessions) as total_num from {{FGT_DATASET_BASE_TRAFFIC_TOP_ APPS}} t where lower(appcat) in ('botnet', 'remote.access', 'storage.backup', 'video/audio', 'p2p', 'proxy') and apprisk in ('critical', 'high') group by appcat order by total_num desc)### union all ###(select 'malicious' as behavior, count(*) as total_num from \$log-attack where \$filter and (logflag&16>0) and severity in ('critical', 'high') group by behavior)###) t where \$filter-drilldown group by behavior order by percentage desc

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

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
<pre>sions from {{FGT_DATASET_BASE_TR /*SkipSTART*/order by sessions d</pre>	apprisk, sum(bandwidth) as bandwidth, sum(se AFFIC_TOP_APPS}} t group by appid, app, appca esc, bandwidth desc/*SkipEND*/)### t where \$f group by appcat order by bandwidth desc	t, apprisk
Dataset Name	Description	Log Category
Apprisk-Ctrl-Category-Breakdown-By- Bandwidth	Category breakdown of all applications, sorted by bandwidth	traffic

select

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

from

###(select app, appcat, user_src, sum(bandwidth) as bandwidth, sum(sessions) as sessions from {{FGT_DATASET_BASE_TRAFFIC_TOP_APPS}} t where nullifna(appcat) is not null group by app, appcat, user_src order by bandwidth desc)### t where \$filter-drilldown group by appcat order by bandwidth desc

Dataset Name	Description	Log Category
Apprisk-Ctrl-Top-Web-Applications-by- Bandwidth	Top 25 Web Categories by Bandwidtih	traffic
<pre>lifna(t1.`unauthuser`), ipstr(t1. byte, 0)) as bandwidth, count(*) c1.appid=t2.id where \$filter and ('80/tcp', '443/tcp', 'HTTP', 'HT</pre>		+coalesce(rcvd- ta t2 on d service in 2.name, t2
Dataset Name	Description	Log Category
Apprisk-Ctrl-Top-Web-Categories- Visited	Top 25 Web Categories Visited	traffic
<pre>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<50200000) 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</pre>		
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)= 'bot	net' then 'Botnet C&C' else (kware%' then 'Spyware' when virus s like 'Adwa	

```
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-
                                    Zero-day malware detected on the network
                                                                                        traffic
Network
select
 virus s,
  appid,
  app,
  count (distinct dstip) as victims,
  count (distinct srcip) as source,
  sum(total num) as total num
from
  ###(select unnest(string to array(virus, ',')) as virus s, appid, app, dstip, srcip, count
```

(*) as total_num from \$log where \$filter and (logflag&1>0) and virus like '%PossibleThreat.SB%' group by virus_s, dstip, srcip, appid, app)### t where virus_s like '%PossibleThreat.SB%' group by virus_s, appid, app order by total_num desc

Dataset Name	Description	Log Category
Apprisk-Ctrl-Files-Analyzed-By- FortiCloud-Sandbox	Files analyzed by FortiCloud Sandbox	virus
<pre>select \$DAY_OF_MONTH as dom, count(*) as total_num from \$log where \$filter and nullifna(filename) is not n and logid_to_int(logid)= 9233 group by dom order by dom</pre>	ull	

Dataset Name	Description	Log Category
Apprisk-Ctrl-Malicious-Files-Detected- By-FortiCloud-Sandbox	Files detected by FortiCloud Sandbox	virus
not null and logid_to_int(logid)		by filename, ana-
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
<pre>select srcname as caller, count(*) as totalnum from \$log where \$filter and lower(appcat)= 'voip' and app = 'sccp' and action = 'block'</pre>		

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
<pre>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</pre>		
Dataset Name	Description	Log Category
security-Top20-High-Risk-Application- In-Use	High risk application in use	traffic
<pre>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&l>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</pre>		
Dataset Name	Description	Log Category
security-High-Risk-Application-By- Category	High risk application by category	traffic
select		

```
select
   app_cat,
   count(distinct app) as total_num
from
```

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

Dataset Name	Description	Log Category
security-Top10-Application-Categories- By-Bandwidth	Application risk application usage by category	traffic

select

appcat,

sum (bandwidth) as bandwidth

from

###(select appid, app, appcat, apprisk, sum(bandwidth) as bandwidth, sum(sessions) as sessions from {{FGT DATASET BASE TRAFFIC TOP APPS}} t group by appid, app, appcat, apprisk /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t where \$filter-drilldown and nullifna(appcat) is not null group by appcat order by bandwidth desc

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

```
appcat,
  count (distinct app) as app num,
 count(distinct user src) as user num,
 sum(bandwidth) as bandwidth,
 sum(sessions) as num session
from
```

###(select app, appcat, user src, sum(bandwidth) as bandwidth, sum(sessions) as sessions from {{FGT DATASET BASE TRAFFIC TOP APPS}} t where nullifna(appcat) is not null group by app, appcat, user src order by bandwidth desc)### t where \$filter-drilldown group by appcat order by bandwidth desc

Dataset Name	Description	Log Category
security-Top25-Web-Applications-By- Bandwidth	Top Web Applications by Bandwidtih	traffic
<pre>select d_risk, name, app_cat, technology, count(distinct f_user) as users sum(bandwidth) as bandwidth, sum(num_session) as sessions</pre>	3,	
<pre>nullifna(t1.`unauthuser`), ipstr (rcvdbyte, 0)) as bandwidth, cour t1.appid=t2.id where \$filter and ('80/tcp', '443/tcp', 'HTTP', 'HT</pre>	app_cat, t2.name, t2.technology, coalesce(nu (t1.`srcip`)) as f_user, sum(coalesce(sentby (t(*) as num_session from \$log t1 inner join (logflag&1>0) and nullifna(app) is not null TTPS', 'http', 'https') group by risk, t2.app o by d_risk, name, app_cat, technology order	te, 0)+coalesce app_mdata t2 on and service in p_cat, t2.name,

Dataset Name	Description	Log Category
Security-Top25-Web-Categories- Visited	Top 25 Web Categories Visited	traffic
user, count(*) as sessions, su Slog-traffic where \$filter and ((logver is null or logver<502 'banned-word', 'web-content',	er_num, (nullifna(`user`), nullifna(`unauthuser` m(coalesce(sentbyte, 0)+coalesce(rcvdbyte catdesc is not null and (logflag&1>0) an 000000) and (hostname is not null or utme 'command-block', 'script-filter')))) grow roup by catdesc order by sessions desc	e, 0)) as bandwidth from nd (countweb>0 or event in ('webfilter',
Dataset Name	Description	Log Category
security-Top25-Malware-Virus-Botne	t- Malware: viruses, Bots, Spyware/Adware	traffic
Spyware select virus_s as virus, (
<pre>select virus_s as virus, (case when lower(appcat)= ' case when virus_s like ' 'Adware' else 'Virus' end) end) as malware_type, count(distinct dstip) as vic count(distinct dstip) as vic count(distinct srcip) as sou sum(total_num) as total_num from (###(select app as virus_s, where \$filter and (logflag&l>0 srcip order by total_num desc) virus_s, appcat, dstip, srcip, flag&l>0) and virus is not nul desc)### union all ###(select total_num from \$log-attack whe</pre>	<pre>rce, appcat, dstip, srcip, count(*) as total) and lower(appcat)='botnet' group by vi ### union all ###(select unnest(string_to count(*) as total_num from \$log-traffic l group by virus_s, appcat, dstip, srcip attack as virus_s, 'null' as appcat, dsti re \$filter and (logflag&16>0) group by v</pre>	_num from \$log-traffic rus_s, appcat, dstip, p_array(virus, ',')) as where \$filter and (log- order by total_num ip, srcip, count(*) as irus_s, appcat, dstip,
<pre>select virus_s as virus, (case when lower(appcat)= ' case when virus_s like ' Adware' else 'Virus' end) end) as malware_type, count(distinct dstip) as vic count(distinct srcip) as sou sum(total_num) as total_num from (###(select app as virus_s, where \$filter and (logflag&l>0 srcip order by total_num desc) virus_s, appcat, dstip, srcip, flag&l>0) and virus is not nul desc)### union all ###(select cotal_num from \$log-attack whe</pre>	Riskware%' then 'Spyware' when virus_s l. tims, rce,) and lower(appcat)='botnet' group by vi ### union all ###(select unnest(string_te count(*) as total_num from \$log-traffic l group by virus_s, appcat, dstip, srcip attack as virus_s, 'null' as appcat, dst	_num from \$log-traffic rus_s, appcat, dstip, p_array(virus, ',')) as where \$filter and (log- order by total_num ip, srcip, count(*) as irus_s, appcat, dstip,

```
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
  com
```

from

###(select virus, virusid_to_str(virusid, eventtype) as virusid_s, srcip, dstip, (case when virus like 'Riskware%' then 'Spyware' when virus like 'Adware%' then 'Adware' else 'Virus' end) as malware_type, count(*) as total_num from \$log where \$filter and nullifna(virus) is not null group by virus, virusid_s, srcip, dstip order by total_num desc)### t group by virus, malware type order by total num desc

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

```
select
```

```
app,
appid,
malware_type,
count(distinct dstip) as victims,
count(distinct srcip) as source,
sum(total_num) as total_num
from
```

(

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

Dataset Name	Description	Log Category
security-Top10-Victims-of-Malware	Victims of Malware	virus
<pre>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</pre>	Victims of Malware	virus
total_num desc		

Dataset Name	Description	Log Category
security-Top10-Victims-of-Phishing- Site	Victims of Phishing Site	webfilter
<pre>select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, (lower(service) '://' h) as phishing_site, count(*) as total_num from \$log where \$filter and lower(service) in ('http', and hostname is not null and cat in (26, 61) group by user_src, phishing_site order by total_num desc</pre>		
Dataset Name	Description	Log Category
security-Top25-Malicious-Phishing- Sites	Malicious Phishing Site	webfilter
count(*) as total from \$log wher	e, '://' hostname url) as phishing_s e \$filter and lower(service) in ('http' group by phishing_site, dstip, srcip or	, 'https') and hostname
Dataset Name	Description	Log Category
security-Application-Vulnerability	Annie stien underenskilitige die seuene d	ette els
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
####(coloct ottool, ottool)
```

###(select attack, attackid, vuln_type, t2.cve, (case when t1.severity='critical' then 5
when t1.severity='high' then 4 when t1.severity='medium' then 3 when t1.severity='low' then 2
when t1.severity='info' then 1 else 0 end) as severity_number, dstip, srcip, count(*) as totalnum from \$log t1 left join (select name, cve, vuln_type from ips_mdata) t2 on t1.attack=t2.name where \$filter and nullifna(attack) is not null and t1.severity is not null group
by attack, attackid, vuln_type, t2.cve, t1.severity, dstip, srcip)### t group by attack,
attackid, vuln type, severity number, cve order by severity number desc, totalnum desc

```
Dataset Name
                                     Description
                                                                                          Log Category
 security-Files-Analyzed-By-FortiCloud-
                                     Files analyzed by FortiCloud Sandbox
                                                                                          virus
 Sandbox
select
  $day of week as dow,
  count(*) as total num
from
  $loq
where
  Sfilter
  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
total_num from \$log where \$filte		ibleThreat.SB%' group
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
  ####({{FGT_DATASET_DLP_VIOLATION_SUMMARY}})### t where $filter-drilldown and severity is not
null group by s severity order by total num desc
```

```
Dataset Name
                                                                                      Log Category
                                    Description
 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
  ###({{FGT DATASET DLP VIOLATION SUMMARY}})### t where $filter-drilldown and filesize is not
null group by filename, direction, service order by filesize desc
 Dataset Name
                                    Description
                                                                                      Log Category
                                                                                      fct-traffic
security-Endpoint-Security-Events-
                                    Endpoint Security Events summary
Summary
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
 $loq
```

```
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
<pre>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</pre>		

```
$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
<pre>select coalesce(nullifna(`user`), ipstr(`deviceip`), 'Unknown') as f_user, coalesce(nullifna(hostname), 'Unknown') as host_name, virus, file from \$log where \$filter and subtype = 'av' and virus is not null group by f_user, host_name, virus, file</pre>		
Dataset Name	Description	Log Category
security-Top-Endpoints-With-Web- Violateions	Endpoints With Web Violations	fct-traffic
<pre>select f_user, host_name, remotename, sum(total_num) as total_num from ###(select coalesce(nullifna(`u</pre>	.ser`), ipstr(`srcip`)) as f_user, c	oalesce(nullifna

(hostname), 'Unknown') as host_name, remotename, count(*) as total_num from \$log where \$filter and utmevent='webfilter' and remotename is not null and utmaction='blocked' group by f_user, host_name, remotename order by total_num desc)### t group by f_user, host_name, remotename order by total_num desc

Dataset Name	Description	Log Category
security-Top-Endpoints-With-Data- Loss-Incidents	Endpoints With Data Loss Incidents	fct-event
<pre>select f_user, host_name, msg,</pre>		
lifna(hostname), 'Unknown') as h	user`), ipstr(`deviceip`), 'Unknown') as f_ ost_name, msg, count(*) as total_num from \$ r, host_name, msg order by total_num desc)# tal_num desc	log where \$filter

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
<pre>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</pre>		

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
<pre>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</pre>		

hourstamp order by hourstamp

Dataset Name	Description	Log Category
content-Count-Total-SCCP-Calls-per- Status	Content count total SCCP calls per status	content
<pre>select status, count(*) as totalnum from \$log where \$filter and proto = 'sccp' and kind = 'call-info' group by status order by totalnum desc</pre>		
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
<pre>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</pre>		
Dataset Name	Description	Log Category
content-Count-Total-SIP-Calls-per- Status	Content count total SIP calls per status	content
<pre>select status, count(*) as totalnum from \$log where \$filter and proto = 'sip' and kind = 'call'</pre>		

group by status order by totalnum desc **Dataset Name** Description Log Category content-Dist-Total-SIP-Calls-by-Content dist total SIP calls by duration content Duration select (case when duration<60 then 'LESS_ONE_MIN' when duration<600 then 'LESS_TEN_MIN' when duration<3600 then 'LESS_ONE_HOUR' when duration & gt;= 3600 then 'MORE_ONE_HOUR' else 'unknown' end) as f duration, count(*) as totalnum from \$loq 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 traffic Botnet activity by sources

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

###({{FGT_DATASET_TRAFFIC_APP_BOTNET}})### t group by app, user_src order by events
desc) union all (select attack, user_src, sum(totalnum) as events from ###({{FGT_DATASET_
ATTACK_APP_BOTNET})### t group by attack, user_src order by events desc)) t group by app,
user_src order by events desc

Dataset Name	Description	Log Category
Botnet-Infected-Hosts	Botnet infected hosts	traffic
<pre>select user_src,</pre>		

```
devtype_new,
 host_mac,
 sum(events) as events
from
  (
```

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

Dataset Name	Description	Log Category
Detected-Botnet	Detected botnet	traffic
<pre>select app, sum(events) as events from ((</pre>		
	_APP_BOTNET}})### t group by app order by event as events from ###({{FGT_DATASET_ATTACK_APP_BO	

group by attack order by events desc)) t group by app order by events desc

Dataset Name	Description	Log Category
Botnet-Sources	Botnet sources	traffic
<pre>select dstip, domain, sum(events) as events</pre>		
from		
(
(
select		
dstip,		
domain,		
sum(events) as e	events	
from		
###(select dstip	, root_domain(hostname) as domain, count((*) as events from \$log-
traffic where \$filter ar	nd (logflag&1>0) and appcat='Botnet' and c	dstip is not null group by
dstip, domain order by e	events desc)### t group by dstip, domain)	union all (select dstip,
root domain(hostname) as	domain, sum(totalnum) as events from ###	({{FGT DATASET ATTACK APP

Dataset Name	Description	Log Category
Botnet-Victims	Botnet victims	traffic

BOTNET}})### t group by dstip, domain)) t group by dstip, domain order by events desc

```
select
  user_src,
  sum(events) as events
from
  (
     (
        select
        user_src,
        sum(totalnum) as events
        from
```

###({{FGT_DATASET_TRAFFIC_APP_BOTNET}})### t group by user_src) union all (select user_src, sum(totalnum) as events from ###({{FGT_DATASET_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 \$filter and (logflag&1>0) and appcat='Botnet' group by timestamp order by timestamp desc)### union all ###(select \$flex_timestamp as timestamp, count(*) as events from \$log-dns where \$filter and (botnetdomain is not null or botnetip is not null) group by timestamp order by timestamp)### union all ###(select \$flex_timestamp as timestamp, count(*) as events from \$logattack 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
<pre>select appid, app, appcat, (case when (utmaction in ('block', 'bl or action = 'deny') then 'Blocked' else 'Allow) as custaction,</pre>		

```
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 /(
<pre>sum(num_reason) over()</pre>		
) as decimal(18, 2)		
) as percent		
from		
(
select		
	the Die Compliant of the Compliant	
—	then 'Non-Compliant' else 'Compliant'	ena
) as status,	num x02000	
count(distinct reason) as from	num_reason	
(
select		
ftnt pci id,		
(
sum(fail count) ove:	r (partition by ftnt pci id)	
) as fail count,		
reason		
from		
###(select ftnt_pci_id	d, (case when result='fail' then 1 els	e 0 end) as fail_count,
reason from \$log t1 inner join }	pci_dss_mdata t2 on t1.reason=t2.ftnt_	id where \$filter and sub-
type='compliance-check' group by	y 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
) as fail_count, reason from ###({{FGT_DATASET_EV ity, count(distinct t.reason from query order by reason,	<pre>rer (partition by ftnt_pci_id) YENT_COMPLIANCE_CHECK}})### t) t where fail_count> a) as requirements from (select distinct on (1) re (case lower(severity) when 'high' then 4 when 'cr .ow' then 1 else 0 end) desc) t group by t.severit</pre>	ason, severity itical' then 3

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
      ###(//FCT_DATASET_EVENT_COMPLIANCE_CHECK)))### t) t
```

###({{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
    (
    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-
```

###(select result, reason from \$log where \$filter and subtype='compliance-check' and result in ('fail','pass') group by result, reason)### t group by status) t order by status desc

Dataset Name	Description	Log Category
PCI-DSS-Failed-Fortinet-Security- Best-Practices-By-Severity	PCI DSS Failed Fortinet Security Best Practices by Severity	event
<pre>select status, num_reason as practices, cast(num_reason * 100.0 /(sum(num_reason) over()) as decimal(18, 2)) as percent form</pre>		
from (

```
(
  select
    initcap(status) as status,
    count(distinct reason) as num_reason
  from
```

###(select status, reason from \$log where \$filter and subtype='compliance-check' and result='fail' group by status, reason)### t group by status) t order by status

Dataset Name	Description	Log Category
PCI-DSS-Passed-Fortinet-Security- Best-Practices-By-Severity	PCI DSS Passed Fortinet Security Best Practices by Severity	event
<pre>select status, num_reason as practices, cast(num_reason * 100.0 /(sum(num_reason) over()) as decimal(18, 2)) as percent from (</pre>		

```
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-
                                   PCI DSS Requirements Compliance Details
                                                                                     event
 Details
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,
```

###(select fint_pci_id, fint_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
<pre>select reason as ftnt_id, msg, initcap(status) as status, module from</pre>		
<pre>\$log where \$filter and subtype = 'compliance-check</pre>	,	
<pre>group by reason, status, module, msg order by ftnt id</pre>		

Dataset Name	Description	Log Category
DLP-Email-Activity-Details	Email DLP Violations Summary	dlp
<pre>select from_itime(itime) as timestamp, sender,</pre>		

```
receiver,
regexp_replace(filename, '.*/', '') as filename,
filesize,
profile,
action,
direction
from
####({{FGT_DATASET_DLP_VIOLATION_SUMMARY}})### t where $filter-drilldown and (service in
('smtp', 'SMTP', '25/tcp', '587/tcp', 'smtps', 'SMTPS', '465/tcp') or service in ('pop3',
'POP3', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/tcp', 'pop3s', 'POP3S',
'995/tcp')) order by timestamp desc
```

Dataset Name	Description	Log Category
Email-DLP-Chart	Email DLP Activity Summary	dlp
('smtp', 'SMTP', '25/tcp', '587/	N_SUMMARY}})### t where \$filter-dri tcp', 'smtps', 'SMTPS', '465/tcp') o ', '143/tcp', 'imaps', 'IMAPS', '993 er by total_num desc	or service in ('pop3',

Dataset Name	Description	Log Category
DLP-Web-Activity-Details	Web DLP Violations Summary	dlp
<pre>select from_itime(itime) as timestamp, srcip, dstip, hostname, profile, filename, filesize, action, direction</pre>		
<pre>from ###({{FGT_DATASET_DLP_VIOLATION in ('http', 'https') order by tin</pre>	I_SUMMARY}})### t where \$filter-drilldown westamp desc	and lower(service)

Dataset Name	Description	Log Category
Web-DLP-Chart	Web DLP Activity Summary	dlp
	VIOLATION_SUMMARY}})### t where \$filter-drill up by profile order by total_num desc	ldown and lower(service)

Dataset Name	Description	Log Category
DLP-FTP-Activity-Details	Web DLP Violations Summary	dlp
<pre>select from_itime(itime) as times srcip, dstip, filename, profile, filesize, action, direction from</pre>	tamp,	

###({{FGT_DATASET_DLP_VIOLATION_SUMMARY}})### t where \$filter-drilldown and lower(service)
in ('ftp', 'ftps') order by timestamp desc

Dataset Name	Description	Log Category	
FTP-DLP-Chart	FTP DLP Activity Summary	dlp	
<pre>select profile, count(*) as total_num</pre>			
<pre>from ###({{FGT_DATASET_DLP_VIOLATION_SUMMARY}})### t where \$filter-drilldown and lower(service) in ('ftp', 'ftps') group by profile order by total num desc</pre>			

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	n, ebtr_agg_flat(browsetime) as browsetime	from (select coalesce
<pre>(nullifna(`user`), ipstr(`sre</pre>	cip`)) as user_src, coalesce(nullifna(hostn	<pre>ame), ipstr(`dstip`))</pre>
as domain, ebtr_agg_flat(\$browse_time) as browsetime from \$log where \$filter and \$browse_tim		
is not null group by user_src, domain) t group by user_src, domain order by ebtr_value(ebtr		
agg_flat(browsetime), null, n	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
<pre>select hod, count(distinct stamac) as tota from</pre>	lnum	

select
 app,
 coalesce(

nullifna(`user`),
nullifna(`unauthuser`),

###(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
<pre>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</pre>		
Dataset Name	Description	Log Category
app-top-user-by-bandwidth	Top 10 Applications Bandwidth by User Drilldown	traffic

```
ipstr(`srcip`)
 ) as user src,
  sum(
    coalesce(`sentbyte`, 0)+ coalesce(`rcvdbyte`, 0)
  ) as bandwidth
from
  $loq
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
<pre>select app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, count(*) as sessions from</pre>		

```
$loq
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
<pre>with qry as (select dom as dom_s, devid as devid_s, vd as vd_s, srcintf, dstintf, total_sent, total_rcvd</pre>		
as total_sent, sum(coalesce(rcvd (rcvdbyte, 0)) as total from \$10 null and nullifna(dstintf) is no (coalesce(sentbyte, 0)+coalesce((array['download', 'upload']) as from (select coalesce(t1.dom_s, coalesce(t1.vd_s, t2.vd_s) as vd (t1.total_sent, 0)+coalesce(t2.t coalesce(t1.total_rcvd, 0)) as u	<pre>dom, devid, vd, srcintf, dstintf, sur byte, 0)) as total_rcvd, sum(coalesce g where \$filter and (logflag&1>0) and t null group by dom, devid, vd, srcin rcvdbyte, 0))>0 order by total desc) type, unnest(array[sum(download), su t2.dom_s) as dom, coalesce(t1.devid_s , coalesce(t1.srcintf, t2.dstintf) as otal_rcvd, 0)) as download, sum(coale pload from qry t1 full join qry t2 or om, devid, vd, intf) t where \$filter-</pre>	<pre>e(sentbyte, 0)+coalesce d nullifna(srcintf) is not ntf, dstintf having sum ### t) select dom, unnest um(upload)]) as bandwidth s, t2.devid_s) as devid, s intf, sum(coalesce esce(t2.total_sent, 0)+- n t1.dom_s=t2.dom_s and</pre>

order by dom

Description

Dataset Name ctap-SB-Files-Needing-Inspection-vs-Files Needing Inspection vs Others Others

```
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
```

Log Category

virus

from

###({{FGT_DATASET_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	s', 'bat', 'cmd',	
'dll', 'ps1', 'jar'		
) then 'Executable Files' w	when suffix in ('pdf') then 'Adobe PDF'	' when suffix in ('swf')
then 'Adobe Flash' when suffix	in (
'doc', 'docx', 'rtf', 'do	otx', 'docm',	
'dotm', 'dot'		
) then 'Microsoft Word' whe		
'xls', 'xlsx', 'xltx', 'x	<lsm', 'xlsb',<="" th=""><td></td></lsm',>	
'xlam', 'xlt'		
) then 'Microsoft Excel' wh	,	
'ppsx', 'ppt', 'pptx', 'p		
'pptm', 'ppsm', 'potm', '	ppam', 'slom',	
'pps', 'pot'	nt' when suffix in ('msg') then 'Micros	oft Outlook! when ouffin
	then 'Web Files' when suffix in (Solt Outlook when Sullix
'cab', 'tgz', 'z', '7z',		
'kqb', 'rar', 'zip', 'qz'		
	n suffix in ('apk') then 'Android Files	s' else 'Others' end
) as filetype,		
sum(total num) as total num		
from		
###({{FGT DATASET VIRUS FSA I	DETECTED FILE TYPES}})### t group by fi	iletype order by total num

###({{FGT_DATASET_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
7		

```
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
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

ctap-SB-Sources-of-Sandbox-Discovered-Malware

Description

```
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
<pre>select risk as d_risk, count(distinct user_src) as use id, name, app_cat, technology, sum(bandwidth) as bandwidth,</pre>	rs,	
sum(sessions) as sessions		
<pre>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</pre>		

Dataset Name	Description	Log Category
ctap-apprisk-ctrl-Application- Vulnerability	Application vulnerabilities discovered	attack
select attack, attackid, vuln_type,		

Log Category

virus

```
cve,
severity_number,
count(distinct dstip) as victims,
count(distinct srcip) as sources,
sum(totalnum) as totalnum
from
```

###(select attack, attackid, vuln_type, t2.cve, (case when t1.severity='critical' then 5
when t1.severity='high' then 4 when t1.severity='medium' then 3 when t1.severity='low' then 2
when t1.severity='info' then 1 else 0 end) as severity_number, dstip, srcip, count(*) as totalnum from \$log t1 left join (select name, cve, vuln_type from ips_mdata) t2 on t1.attack=t2.name where \$filter and nullifna(attack) is not null and t1.severity is not null group
by attack, attackid, vuln_type, t2.cve, t1.severity, dstip, srcip)### t group by attack,
attackid, vuln_type, severity_number, cve order by severity_number desc, totalnum desc

Dataset Name	Description	Log Category
ctap-apprisk-ctrl-Common-Virus- Botnet-Spyware	Common Virus Botnet Spyware	app-ctrl
<pre>case when malware like 'R 'Adware' else 'Virus' end) end) as malware_type, appid, app, count(distinct dstip) as vict count(distinct srcip) as sour sum(total_num) as total_num from (###(select app as malware, \$log-app-ctrl where \$filter and dstip, srcip, app order by tota appcat, 0 as appid, service as \$filter and virus is not null g total_num desc)### union all ## vice as app, dstip, srcip, coun flag&16>0) group by malware, ap</pre>		nt(*) as total_num from are, appcat, appid, app, irus as malware, 'null' as num from \$log-virus where dstip, srcip order by appcat, 0 as appid, ser- ere \$filter and (log-
Dataset Name	Description	Log Category

Dataset Name	Description	Log Category
ctap-App-Risk-Reputation-To Devices-By-Scores	p- Reputation Top Devices By-Scores	traffic
<pre>select coalesce(nullifna(`srcname`), ipstr(`srcip`), nullifna(`srcmac`)) as dev_src, sum(crscore % 65536) as</pre>	s 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) as service,	O/tcp', 'HTTP', 'http') then '	HTTP' else 'HTTPS' end	
) as service, sum(
coalesce(sentbyte, 0)+ co	alesce(rcvdbyte, 0)		
) as bandwidth			
from			
\$log			
where			
\$filter			
and (
logflag&1>0			
and nullifna(app) is not nu and service in (
'80/tcp', '443/tcp', 'HT	רסי ואייסא		
'http', 'https'	. ,,		
)			
group by			
service			
having			
sum(
coalesce(sentbyte, 0) + co	palesce(rcvdbyte, 0)		
)& gt; 0			
order by bandwidth desc			
Sandwilden dese			

Dataset Name	Description	Log Category
ctap-Top-Source-Countries	Top Source Countries	traffic
<pre>select srccountry, sum(coalesce(sentbyte, 0)+ co) as bandwidth</pre>	palesce(rcvdbyte, 0)	

```
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

app_group, sum(bandwidth) as bandwidth

```
from
```

###({{FGT_DATASET_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 laaS Apps	traffic	
<pre>select app_group, sum(bandwidth) as bandwidth from ###({{FGT_DATASET_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</pre>			
Dataset Name	Description		
Bataset Maine	Becomption	Log Category	
ctap-RAS-Apps	CTAP RAS Apps	traffic	

Dataset Name	Description	Log Category
ctap-Proxy-Apps	CTAP Proxy Apps	traffic

```
name as app_group,
sum(bandwidth) as bandwidth
```

from

###({{FGT_DATASET_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_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-	Top Streaming applications by bandwidth usage	traffic
Bandwidth		

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_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_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

```
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_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
user, count(*) as sessions, sum(c \$log-traffic where \$filter and ca ((logver is null or logver<502000	llifna(`user`), nullifna(`unauthuser`), ip palesce(sentbyte, 0)+coalesce(rcvdbyte, 0) tdesc is not null and (logflag&1>0) and (0 000) and (hostname is not null or utmevent)) as bandwidth from countweb>0 or t in ('webfilter',
	<pre>mmand-block', 'script-filter')))) group by p by catdesc order by sessions desc</pre>	y I_user, catdesc
Dataset Name	Description	Log Category
ctap-App-Risk-Applications-Running- Over-HTTP	Application risk applications running over HTTP	traffic
<pre>select app_group_name(app) as app_group service, count(*) as sessions, sum(coalesce(sentbyte, 0)+ coales) as bandwidth from</pre>		
\$log where		
<pre>%Here \$filter and (logflag&1>0) and nullifna(app) is not null and service in ('80/tcp', '443/tcp', 'HTTP', 'http', 'https'</pre>	'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
<pre>select domain, catdesc, sum(visits) as visits</pre>		
<pre>from ###(select coalesce(nullifna(hc))</pre>	stname), ipstr(`dstip`)) as domain, catdesc, com	unt(*) as vis-

its from \$log where \$filter and (eventtype is null or logver>=502000000) and catdesc is not null group by domain, catdesc order by visits desc)### t group by domain, catdesc order by visits desc

Dataset Name	Description	Log Category
ctap-Top-Sites-By-Browsing-Time	Traffic top sites by browsing time	traffic
<pre>select hostname, string_agg(distinct 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</pre>	') as agg_catdesc,	
from	ITES_BY_EB_TIME}})### t group by hostname order	r by browsetime

de	S	С	

Dataset Name	Description	Log Category
ctap-Average-Bandwidth-Hour	Average Bandwidth Hour	traffic

```
select
```

hourstamp,

 $\operatorname{sum}\left(\operatorname{bandwidth}\right)/\operatorname{count}\left(\operatorname{distinct}\,\operatorname{daystamp}\right)$ as bandwidth from

###(select to_char(from_dtime(dtime), 'HH24:00') as hourstamp, to_char(from_dtime(dtime),
'DD Mon') as daystamp, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from \$log
where \$filter and (logflag&1>0) group by hourstamp, daystamp having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by hourstamp)### t group by hourstamp order by hourstamp

Dataset Name	Description	Log Category
ctap-Top-Bandwidth-Hosts	Top Bandwidth Hosts	traffic

```
select
 hostname,
 sum(
   coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
  ) as bandwidth
from
  $log - traffic
where
 $filter
 and hostname is not null
 and (
   logflag&1>0
 )
group by
 hostname
having
 sum(
    coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
 )>0
order by
 bandwidth desc
```

Dataset Name	Description	Log Category
saas-Application-Discovered	All Applications Discovered on the Network	traffic

```
select
```

```
(
   case is_saas when 1 then '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
```

```
order by saas_cat
```

Dataset Name	Description	Log Category
saas-SaaS-Application-by-Bandwidth	Number of SaaS Applications by Bandwidth	traffic

```
(
   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		

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
<pre>) as app_type, count(distinct saasuser) as to from ###(select saasuser, saas_s%10 (`clouduser`), nullifna(`unauthu</pre>	_ as saas_cat from (select coalesce(nulli ser`), srcname, ipstr(`srcip`)) as saasu er and apps is not null) t where saas_s>=	fna(`user`), nullifna ser, unnest(saasinfo)
Dataset Name	Description	Log Category
saas-Top-SaaS-User-by-Bandwidth- Session	Top SaaS Users by Bandwidth and Session	traffic
select		

```
saasuser,
sum(bandwidth) as bandwidth,
sum(traffic_in) as traffic_in,
```

```
sum(traffic_out) as traffic_out,
sum(sessions) as sessions,
sum(session_block) as session_block,
(
    sum(sessions) - sum(session_block)
) as session_pass,
count(distinct app_s) as total_app
from
```

###(select saasuser, app_s, sum(sentbyte+rcvdbyte) as bandwidth, sum(rcvdbyte) as traffic_ in, sum(sentbyte) as traffic_out, count(*) as sessions, sum(is_blocked) as session_block from (select coalesce(nullifna(`user`), nullifna(`clouduser`), nullifna(`unauthuser`), srcname, ipstr(`srcip`)) as saasuser, unnest(apps) as app_s, unnest(saasinfo) as saas_s, coalesce(sentbyte, 0) as sentbyte, coalesce(rcvdbyte, 0) as rcvdbyte, (CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END) as is_blocked from \$log where \$filter and apps is not null) t where saas_s>=10 group by saasuser, app_s order by bandwidth desc)### t group by saasuser order by bandwidth desc

Dataset Name	Description	Log Category
saas-Top-Category-by-SaaS- Application-Usage	Top Categories by SaaS Application Usage	traffic
<pre>select app_cat, (case saas_cat when 0 the) as saas_cat_str, count(distinct app_s) as t</pre>	n 'Sanctioned' else 'Unsactioned' end otal_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
) as saas_cat_str, count(distinct saasuser) as tot from ###({{FGT_DATASET_TRAFFIC_SAAS_	nctioned' else 'Unsactioned' end al_user TOP_USER}})### t1 inner join app_mdata t2 o 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,		
<pre>sum(traffic_in) as traffic_in,</pre>		
<pre>sum(traffic_out) as traffic_ou</pre>	t,	
sum(sessions) as sessions,		
<pre>sum(session_block) as session_</pre>	block,	
(
<pre>sum(sessions) - sum(session_b</pre>	lock)	
) as session_pass		
from		
###(select app s, sum(sentbyte	+rcvdbvte) as bandwidth, sum(rcvdbvte) as ti	raffic in. sum(sent

###(select app_s, sum(sentbyte+rcvdbyte) as bandwidth, sum(rcvdbyte) as traffic_in, sum(sentbyte) as traffic_out, count(*) as sessions, sum(is_blocked) as session_block from (select unnest(apps) as app_s, unnest(saasinfo) as saas_s, coalesce(sentbyte, 0) as sentbyte, coalesce (rcvdbyte, 0) as rcvdbyte, (CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END) as is_blocked from \$log where \$filter and apps is not null) t where saas_s>=10 group by app_s)### t1 inner join app_ mdata t2 on t1.app_s=t2.name group by app_id, app_s, app_cat order by bandwidth desc

Dataset Name	Description	Log Category
saas-Top-Tolerated-SaaS-Application- by-Bandwidth	Top Tolerated SaaS Applications by Bandwidth	traffic
select		
app_s,		
<pre>sum(sentbyte + rcvdbyte) as ban</pre>	dwidth	
from		
(
select		
unnest(apps) as app_s,		
unnest(saasinfo) as saas_s,		
coalesce(sentbyte, 0) as se		
coalesce(rcvdbyte, 0) as rc	vdbyte	
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
<pre>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_bloc (sum(sessions) - sum(session_bloc) as session_pass from ###({{FGT_DATASET_TRAFFIC_TOLER; order by bandwidth desc </pre>	lock,	rilldown group by app_s

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 session,
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
<pre>select t2.id as appid, (case t2.risk when '5' the then 'Info' else 'Low' end) as risk, app_group, bandwidth, traffic_in, traffic_out, sessions, session_block, session_pass,</pre>	en 'Critical' when '4' then 'High' when	'3' then 'Medium' when '2'
total_user		
from		
<pre>(select app_group, count(distinct saasuse: sum(bandwidth) as bandu sum(traffic_in) as tra: sum(traffic_out) as tra: sum(sessions) as session sum(session_block) as session (sum(sessions) - sum(second)) as session_pass</pre>	width, ffic_in, affic_out, ons, session_block,	
from		
	FIC_SAAS_APP_GROUP}))### t group by 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
<pre>select t2.id as appid, (case t2.risk when '5' then then 'Info' else 'Low' end) as risk, app_group, bandwidth, traffic_in, traffic_out, sessions, session_block, session_pass, total_user</pre>	'Critical' when '4' then 'High' whe	en '3' then 'Medium' when '2'

```
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
<pre>select 'All' : :text as country, count(distinct devid) as device from ###(select devid from \$log wher</pre>	_	
Dataset Name	Description	Log Category
aware-Network-Endpoint-Devices	Endpoint Devices on Network	
<pre>select category, total_num from (select 'Seen Devices' as category, 1 as idx, count(distinct epname) as t from (select epname, map_dev.devid, map_dev.vd, max(lastseen) as itime from \$ADOM_ENDPOINT t inner join \$ADOM_EPEU_D where epname is not null group by epname, map_dev.devid,</pre>	otal_num EVMAP map_dev on t.epid = map_dev.epid	

```
map_dev.vd
  ) t
where
 $filter
  and $filter - drilldown
union all
select
 'New Devices' as category,
  2 as idx,
 count(distinct epname) as total_num
from
  (
   select
     epname,
     map dev.devid,
     map dev.vd,
     min(firstseen) as itime
    from
      $ADOM ENDPOINT t
      inner join $ADOM_EPEU_DEVMAP map_dev on t.epid = map_dev.epid
    where
     epname is not null
    group by
      epname,
      map_dev.devid,
      map_dev.vd
  ) t
where
 $filter
 and $filter - drilldown
union all
select
  'Unseen Devices' as category,
  3 as idx,
  count (distinct t1.epname) as total num
from
  $ADOM_ENDPOINT t1
where
 not exists (
   select
      1
    from
      (
        select
          epname,
          map dev.devid,
          map dev.vd,
          max(lastseen) as itime
        from
          $ADOM ENDPOINT t
          inner join $ADOM_EPEU_DEVMAP map_dev on t.epid = map_dev.epid
        where
          epname is not null
        group by
          epname,
          map_dev.devid,
```

```
map_dev.vd
          ) t2
        where
          $filter
          and $filter - drilldown
          and t1.epname = t2.epname
      )
  ) t
order by
  idx
```

Dataset Name

aware-New-Endpoint-Devices

```
Description
New Endpoint Devices
```

Log Category

```
drop
  table if exists devmap_tmp; create temporary table devmap_tmp as (
   select
     epid,
     max(euid) as max_euid
    from
      $ADOM EPEU DEVMAP
   where
     euid & gt;= 1024
   group by
     epid
 );
select
 timestamp,
 epname as hostname,
 max(osname) as osname,
 max(devtype) as devtype,
 max(srcip) as srcip,
  string_agg(distinct epname, ',') as user_agg
from
  (
    select
     from_itime(itime) as timestamp,
     osname,
     epname,
      epdevtype as devtype,
      epip as srcip,
      epid
    from
      (
        select
         max(osname) as osname,
          max(epname) as epname,
         max(epdevtype) as epdevtype,
         max(epip) as epip,
         t.epid,
          map dev.devid,
          map dev.vd,
          min(firstseen) as itime
        from
          $ADOM ENDPOINT t
```

```
inner join $ADOM_EPEU_DEVMAP map_dev on t.epid = map_dev.epid
        where
          epname is not null
        group by
         epname,
         t.epid,
         map dev.devid,
         map_dev.vd
     ) t
    where
     $filter
     and $filter - drilldown
  ) t1
  inner join devmap_tmp on devmap_tmp.epid = t1.epid
  inner join $ADOM_ENDUSER as teu on devmap_tmp.max_euid = teu.euid
group by
 timestamp,
 hostname
order by
 timestamp desc
```

Dataset Name	Description	Log Category
aware-New-Endpoint-Devices-Trend	New Endpoint Devices Trend	
<pre>select \$flex_timescale(itime) as hodex count(distinct epname) as total from (select epname, map_dev.devid, man_dev.vd, min(firstseen) as itime from \$ADOM_ENDPOINT t inner join \$ADOM_EPEU_DEVMA where epname is not null group by epname, map_dev.devid, map_dev.vd) t where \$filter and \$filter - drilldown group by hodex order by hodex </pre>		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	
	- '.', 1) as srcname1, hostname from \$log where ower(os) like '%windows%' group by srcname, hos		
Dataset Name	Description	Log Category	
aware-Top-Endpoint-Applications-Mac	Top Endpoint Applications Mac	fct-traffic	
	- '.', 1) as srcname1, hostname from \$log where ower(os) like '%mac os%' group by srcname, host		
Dataset Name	Description	Log Category	
aware-Top-SaaS-Application-by- Number-of-Users	Top SaaS Applications by Number of Users	traffic	
<pre>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(`un- authuser`), srcname, ipstr(`srcip`)) as saasuser from \$log where \$filter and (logflag&l>0) and apps is not null) t where saas_s>=10 group by app_group, saasuser)### t group by app_group order by total_user desc</pre>			
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
<pre>select \$calendar_time as timestamp, `user`, ui, msg from \$log where \$filter and logid_to_int(logid) = 44547 order by timestamp desc</pre>		
Dataset Name	Description	Log Category

Dataset Name	Description	Log Category
aware-Vulnerabilities-By-Severity	Vulnerabilities by Security	fct-netscan

```
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, sur

###(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
<pre>select vulnname, vulnseverity, vulncat, count(distinct hostname) as t</pre>	otal_num	
<pre>from ###(select hostname, vulname, 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,</pre>		

vulncat order by total_num desc

Dataset Name	Description	Log Category
aware-Top-Vulnerabilities-Last-Period	Top Vulnerabilities Last Period	fct-netscan
<pre>select vulnname, vulnseverity, sev_num, vulncat, count(distinct hostname) as tot</pre>	.al_num	
	vulnseverity, (CASE vulnseverity WHEN 'Critica 3 WHEN 'Info' THEN 2 WHEN 'Low' THEN 1 ELSE 0	

'High' THEN 4 WHEN 'Medium' THEN 3 WHEN 'Info' THEN 2 WHEN 'Low' THEN 1 ELSE 0 END) as sev_ num, vulncat, count(*) as total_num from \$log where \$pre_period \$filter and nullifna(vulnname) is not null group by hostname, vulnname, vulnseverity, vulncat order by sev_num desc, total_ num desc)### t group by vulnname, vulnseverity, sev_num, vulncat order by sev_num desc, total_ num desc

Dataset Name	Description	Log Category
aware-Top-New-Vulnerabilities	Top New Vulnerabilities	fct-netscan
<pre>drop table if exists rpt_tmptbl_1; drop table if exists rpt_tmptbl_2; select vulnid, vulnname, vulnseverity, vulncat, hostname from ###(select vulnid, vulnname, v</pre>	create temporary table rpt_tmptbl_1 ulnseverity, vulncat, hostname from	as \$log where \$pre_period
<pre>hostname)### t group by vulnid, table rpt_tmptbl_2 as select vul vulnid, vulnname, vulnseverity, nname) is not null group by vuln vulnid, vulnname, vulnseverity,</pre>	s not null group by vulnid, vulnname vulnname, vulnseverity, vulncat, how nid, vulnname, vulnseverity, vulncar vulncat, hostname from \$log where \$ id, vulnname, vulnseverity, vulncat vulncat, hostname; select vulnname, everity='High' then 4 when vulnsever	<pre>stname; create temporary t, hostname from ###(select filter and nullifna(vul- , hostname)### t group by (case when vulnsever-</pre>

vulnseverity='Low' then 2 when vulnseverity='Info' then 1 else 0 end) as sev, vulnseverity, vulncat, count(distinct hostname) as host_num, cve_id from rpt_tmptbl_2 t1 left join fct_mdata t2 on t1.vulnid=t2.vid::int where not exists (select 1 from rpt_tmptbl_1 where t1.vulnid=rpt_ tmptbl_1.vulnid) group by vulnname, sev, vulnseverity, vulncat, cve_id order by sev desc, host_num desc

Dataset Name	Description	Log Category
aware-Top-User-With-Critical- Vulnerabilities	Top Users with Critical Vulnerabilities	fct-netscan
<pre>select hostname, `user` as user_src, vulnname, vulncat, count(*) as total_num from \$log where \$filter and nullifna(`user`) is not null</pre>	L	
<pre>and vulnseverity = 'Critical' group by hostname, user_src, vulnname, vulncat order by total_num desc</pre>		

Dataset Name	Description	Log Category
aware-Ingress-Data-Flow-By-Zone	Ingress Data Flow By Zone	traffic
<pre>select app, tag, sum(rcvdbyte) as rcvdbyte from ###(select dvid, app, dstintf, ter group by dvid, app, dstintf) desc)### tt1 inner join (select d tt1.dvid=tt2.dvid and tt1.dstint)</pre>	dvid, intfname, unnest(tags) as) > 0 order by rcvdbyte tag from intfinfo) tt2 on
Dataset Name	Description	Log Category
aware-Egress-Data-Flow-By-Zone	Egress Data Flow By Zone	traffic
<pre>select app, tag, sum(sentbyte) as sentbyte from ###(select dvid, app, srcintf, sum(coalesce(sentbyte, 0)) as sentbyte from \$log where \$fil- ter group by dvid, app, srcintf having sum(coalesce(sentbyte, 0)) > 0 order by sentbyte</pre>		

desc)### tt1 inner join (select dvid, intfname, unnest(tags) as tag from intfinfo) tt2 on tt1.dvid=tt2.dvid and tt1.srcintf=tt2.intfname group by app, tag order by sentbyte desc

Dataset Name	Description	Log Category
aware-Top-Device-Attack-Targets	Top Device Attack Targets	fct-netscan
<pre>select hostname, count(*) as total_num from \$log where \$filter and nullifna(hostname) is no and nullifna(vulnname) is no group by hostname order by total_num desc</pre>		
Dataset Name	Description	Log Category
aware-Top-Attack-Targets	Top Attack Targets	fct-netscan
select hostname, srcip, os,		

(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,

Dataset Name	Description	Log Category
aware-Threats-By-Severity	Threats by Severity	attack

```
initcap(sev) as severity,
  sum(total_num) as total_num
from
```

###(select crlevel::text as sev, count(*) as total_num from \$log-virus where \$filter and nullifna(virus) is not null and crlevel is not null group by sev order by total_num desc)### union all ###(select severity::text as sev, count(*) as total_num from \$log-attack where \$filter and nullifna(attack) is not null and severity is not null group by sev order by total_num desc)### union all ###(select apprisk::text as sev, count(*) as total_num from \$log-app-ctrl where \$filter and lower(appcat)='botnet' and apprisk is not null group by sev order by total_ num desc)###) t group by severity order by total_num desc

Dataset Name	Description	Log Category
aware-Threats-Type-By-Severity	Threats Type by Severity	virus
<pre>select threat_type, sum(critical) as critical, sum(high) as high, sum(medium) as medium, sum(low) as low</pre>		
from (
	type='botnet' then 'Botnets' els critical' then 1 else 0 end) as	

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 = '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 NameDescriptionLog Categoryaware-Threats-By-DayThreats by Dayvirus
```

```
select
  daystamp,
  sum(total_num) as total_num
from
  (
```

###(select \$day_of_week as daystamp, count(*) as total_num from \$log-virus where \$filter
and nullifna(virus) is not null group by daystamp)### union all ###(select \$day_of_week as daystamp, count(*) as total num from \$log-attack where \$filter and nullifna(attack) is not null

group by daystamp)### union all ###(select \$day_of_week as daystamp, count(*) as total_num
from \$log-app-ctrl where \$filter and lower(appcat)='botnet' group by daystamp)###) t group by
daystamp order by daystamp

Dataset Name	Description	Log Category
aware-Threats-By-Day-Radar	Threats by Day	virus
select daystamp,		

```
sum(total_num) as total_num
from
```

(

###(select \$day_of_week as daystamp, count(*) as total_num from \$log-virus where \$filter and nullifna(virus) is not null group by daystamp)### union all ###(select \$day_of_week as daystamp, count(*) as total_num from \$log-attack where \$filter and nullifna(attack) is not null group by daystamp)### union all ###(select \$day_of_week as daystamp, count(*) as total_num from \$log-app-ctrl where \$filter and lower(appcat)='botnet' group by daystamp)###) t group by daystamp order by daystamp

Dataset Name	Description	Log Category
aware-Count-Of-Malware-Events	Count of Malware Events	virus
<pre>select virus, count(*) as total_num from \$log where \$filter and nullifna(virus) is not null group by virus order by total_num desc</pre>		

Dataset Name	Description	Log Category
aware-Top-Malware-By-Count	Top Malware by Count	app-ctrl
<pre>select virus, malware_type, risk_level, count(distinct dstip) as victim count(distinct srcip) as source sum(total num) as total num</pre>		
from		
<pre>srcip, count(*) as total_num from apprisk is not null group by app, desc)### union all ###(select vir 'Virus' end) as malware_type, crl</pre>	<pre>net C&C' as malware_type, apprisk::text a \$log-app-ctrl where \$filter and lower(ap malware_type, apprisk, dstip, srcip orde us, (case when eventtype='botnet' then 'H evel::text as risk_level, dstip, srcip, o nullifna(virus) is not null and crlevel</pre>	<pre>ppcat)='botnet' and er by total_num Botnet C&C' else count(*) as total_num</pre>

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

Dataset Name	Description	Log Category
aware-Top-Failed-Authentication- Attempts	VPN failed logins	event
<pre>num from \$log where \$filter an 3)='ssl') and action in ('ssl-</pre>	(`xauthuser`), `user`) as f_user, tu d subtype='vpn' and (tunneltype='ips login-fail', 'ipsec-login-fail') and not null group by f_user, tunneltyp c	<pre>sec' or left(tunneltype, d coalesce(nullifna(`xau-</pre>
Dataset Name	Description	Log Category
aware-Top-Denied-Connections	Top Denied Connections	traffic
select		

```
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 tota. from	l_num	
###(select devid, reason from	Slog where \$filter and subtype='complian	ce-check' and res-

###(select devid, reason from \$log where \$filter and subtype='compliance-check' and result='fail' group by devid, reason)### t group by devid, results order by total_num desc

Dataset Name	Description	Log Category
aware-loc-Blacklist-Summary	IOC Blacklist Summary	app-ctrl
select epid, euid from \$ADOM_EPEU_DEVMAP where	create temporary table tmp_ep_eu_map as (
<pre>euid & gt;= 1024); select coalesce(nullifna(epname), nullifna(ipstr(`srcip`)</pre>		
), 'Unknown') as epname, user_agg, sevid, (CASE sevid WHEN 5 THEN 'Critic 'Info' ELSE 'Low' END	cal' WHEN 4 THEN 'High' WHEN 3 THEN 'Medium' WHE	EN '2' THEN

```
) as severity,
 threats,
 bl_count as total_bl
from
  (
    select
     th1.epid,
     srcip,
     sevid,
     bl count,
     threats
    from
      (
        select
          epid,
          srcip,
          max(verdict) + 1 as sevid,
          sum(bl_count) as bl_count
        from
          (
            (
              select
               epid,
                srcip,
                day_st as itime,
                bl_count,
                verdict,
                unnest(dvid) as dvid_s
              from
                $ADOMTBL PLHD IOC VERDICT
              where
                bl_count>0
            )
            union all
              (
                select
                 epid,
                  srcip,
                  day_st as itime,
                  bl_count,
                  verdict,
                  unnest(dvid) as dvid_s
                from
                  $ADOMTBL_PLHD_INTERIM_IOC_VERDICT
                where
                  bl count>0
              )
          ) tvdt
          inner join devtable td on td.dvid = tvdt.dvid s
        where
          $filter
          and $filter - drilldown
          and $dev filter
        group by
          epid,
          srcip
```

```
) th1
inner join (
  select
    epid,
    string_agg(name, ',') as threats
  from
    (
      (
        select
         epid,
          thid
        from
          (
            (
              select
                epid,
                thid,
                itime,
                unnest(dvid) as dvid s
              from
                (
                  select
                    epid,
                    unnest(threatid) as thid,
                    day_st as itime,
                    dvid
                  from
                    $ADOMTBL_PLHD_IOC_VERDICT
                  where
                    bl count>0
                ) tal
            )
            union all
              (
                select
                  epid,
                  thid,
                  itime,
                  unnest(dvid) as dvid_s
                from
                   (
                    select
                      epid,
                      unnest(threatid) as thid,
                      day_st as itime,
                      dvid
                    from
                       $ADOMTBL_PLHD_INTERIM_IOC_VERDICT
                    where
                      bl count>0
                  ) ta2
              )
          ) t
          inner join devtable td on td.dvid = t.dvid s
        where
          $filter
```

```
and $filter - drilldown
                and $dev_filter
              group by
                epid,
                thid
            ) thr
           inner join td threat name mdata tm on tm.id = thr.thid
         ) t
       group by
         epid
     ) th2 on th1.epid = th2.epid
  ) t1
 left join (
   select
     epid,
     string_agg(distinct euname, ',') as user_agg
    from
     tmp_ep_eu_map tpu
     inner join $ADOM_ENDUSER as teu on tpu.euid = teu.euid
   group by
     epid
  ) t2 on t2.epid = t1.epid
  inner join $ADOM_ENDPOINT as tep on tep.epid = t1.epid
order by
  total_bl desc,
  sevid desc
```

aware-loc-Potential-Breach-By-Day elect number, day_st as itime rom (IOC Potential Breach by Day	app-ctrl
number, day_st as itime rm		
day_st as itime rom		
rom		
rom		
(
select		
count(epid) as number,		
to_char(
<pre>from_itime(itime),</pre>		
'Day'		
) as day_st		
from		
(
select		
epid,		
day_st as itime,		
unnest(dvid) as dvid_s		
from		
\$ADOMTBL_PLHD_INTERIM_IO	C_VERDICT	
where		
cs_count>0		
union all		
(
select		
epid,		

```
day_st as itime,
              unnest(dvid) as dvid_s
            from
              $ADOMTBL_PLHD_IOC_VERDICT
            where
              cs count>0
          )
      ) t
      inner join devtable td on td.dvid = t.dvid s
    where
      $filter
      and $filter - drilldown
   group by
     day_st
  ) tt
order by
  itime
```

Dataset Name Description Log Category aware-loc-Potential-Breach-By-Day-IOC Potential Breach by Day app-ctrl Bar 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, unnest(dvid) as dvid_s from \$ADOMTBL PLHD INTERIM IOC VERDICT where cs count>0 union all (select epid, day st as itime, unnest(dvid) as dvid_s from \$ADOMTBL_PLHD_IOC_VERDICT where cs_count>0)) t

```
inner join devtable td on td.dvid = t.dvid_s
where
    $filter
    and $filter - drilldown
    group by
        day_st
) tt
order by
    itime
```

Dataset Name	Description	Log Category
aware-loc-Suspicion-Summary	IOC Suspicion Summary	app-ctrl
<pre>select coalesce(nullifna(epname), nullifna(ipstr(`srcip`)),</pre>		
'Unknown') as epname, cs_count as total_cs,		
cs_score as max_cs, verdict as max_verdict, threats		
from		
<pre>(select th1.epid, srcip, itime, cs_count, verdict, cs_score, threats from (select epid, srcip, min(itime) as itime, } }</pre>		
<pre>sum(cs_count) as cs_co max(verdict) as verdic max(cs_score) as cs_so from (</pre>	ct,	
(select epid, srcip, day_st as itime, cs_count, verdict, cs_score, unnest(dvid) as		

```
from
          $ADOMTBL_PLHD_IOC_VERDICT
        where
          bl count = 0
          and cs_count>0
      )
      union all
        (
          select
            epid,
            srcip,
            day st as itime,
            cs_count,
            verdict,
            cs score,
            unnest(dvid) as dvid s
          from
            $ADOMTBL_PLHD_INTERIM_IOC_VERDICT
          where
            bl_count = 0
            and cs_count>0
        )
    ) tvdt
    inner join devtable td on td.dvid = tvdt.dvid_s
  where
    $filter
    and $filter - drilldown
  group by
    epid,
    srcip
) th1
inner join (
  select
    epid,
    string_agg(name, ',') as threats
  from
    (
      (
        select
          epid,
          thid
        from
          (
            (
              select
                epid,
                thid,
                itime,
                unnest(dvid) as dvid s
              from
                (
                  select
                    epid,
                    unnest(threatid) as thid,
                    day_st as itime,
                    dvid
```

```
from
                           $ADOMTBL_PLHD_IOC_VERDICT
                        where
                          bl count = 0
                          and cs_count>0
                      ) tal
                  )
                  union all
                     (
                      select
                        epid,
                        thid,
                        itime,
                        unnest(dvid) as dvid_s
                      from
                         (
                          select
                            epid,
                            unnest(threatid) as thid,
                            day_st as itime,
                            dvid
                          from
                             $ADOMTBL_PLHD_INTERIM_IOC_VERDICT
                          where
                            bl_count = 0
                             and cs_count>0
                        ) ta2
                    )
                ) tt1
                inner join devtable td on td.dvid = tt1.dvid s
              where
                $filter
                and $filter - drilldown
              group by
                epid,
                thid
            ) thr
            inner join td threat name mdata tm on tm.id = thr.thid
          ) tt2
        group by
          epid
      ) th2 on th1.epid = th2.epid
  ) t
  inner join $ADOM_ENDPOINT as tep on tep.epid = t.epid
order by
 max verdict desc,
 max cs desc,
 total_cs desc
```

Dataset Name

Description

Log Category

aware-Botnet-IP

Top Source IP Affected by Botnet

virus

select f user, srcip,

```
string_agg(distinct `virus`, ',') as virus_agg,
count(
    distinct ipstr(`dstip`)
) as dstip_cnt,
max(action) as action,
sum(total_num) as total_num,
min(
    from_itime(first_seen)
) as first_seen,
max(
    from_itime(last_seen)
) as last_seen
from
```

###(select coalesce(nullifna(`user`), nullifna(`unauthuser`)) as f_user, srcip, virus, dstip, max(action) as action, count(*) as total_num, min(itime) as first_seen, max(itime) as last_seen from \$log where \$filter and logid in ('0202009248', '0202009249') and virus is not null group by srcip, f_user, virus, dstip order by total_num desc)### t group by srcip, f_user order by total num desc

Dataset Name	Description	Log Category
aware-Botnet-Domain	New Botnet Domains	dns
<pre>total_num, min(nanosec_to_sec(eve last_seen from \$log where \$filter</pre>	<pre>in`, ipstr(`botnetip`)) as botnet, qname, dstip nttime)) as first_seen, max(nanosec_to_sec(ever and logid in ('1501054601', '1501054600') grou esc)### t group by botnet order by first_seen of</pre>	nttime)) as up by botnet,
Dataset Name	Description	Log Category
aware-High-Risk-URL-Category	Category of High Risk URLs	webfilter

select
catdesc,
string_agg(distinct hostname, ',') as hostname_agg,
max(action) as action,
sum(total_num) as total_num,
min(
 from_itime(first_seen)
) as first_seen,
max(
 from_itime(last_seen)

```
) as last_seen from
```

###(select catdesc, hostname, max(action) as action, count(*) as total_num, min(itime) as first_seen, max(itime) as last_seen from \$log where \$filter and cat in (26, 61, 86, 88, 90, 91, 93) group by catdesc, hostname order by total_num desc)### t group by catdesc order by total_num desc

Dataset Name	Description	Log Category
aware-Malicious-Files	Type of Malicious Files from AV and S	Sandbox virus
<pre>string_agg(distinc max(quarskip) as q max(action) as act max(from_sandbox) sum(total_num) as min(from_itime(first) as first_seen, max(from_itime(last_) as last_seen from ###(select virus, when logid in ('02110092 total_num, min(itime) as is not null and logid in '0211008194', '021100819 desc)### t group by viru</pre>	<pre>0) as filename_agg, t url, ' ') as url_agg, t filename, ' ') as filename_agg, uarskip, ion, as from_sandbox, total_num, _seen) seen) url, filename, max(quarskip) as quarskip 34', '0211009235') then 1 else 0 end) as first_seen, max(itime) as last_seen from ('0211009234', '0201009235', '021100819 5') group by virus, url, filename, from_ s) t order by total_num desc</br></br></pre>	s from_sandbox, count(*) as om \$log where \$filter and virus 92', '0211008193', _sandbox order by total_num
Dataset Name	Description	Log Category
newthing-New-Users	New users	fct-traffic
drop table if exists rpt_tm	ptbl_1;	

```
idual if exists rpt_tmptbl_1;
drop
table if exists rpt_tmptbl_2; create temporary table rpt_tmptbl_1 as
select
f_user,
min(start_time) as start_time
```

from

###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as f_user, min(dtime) as start_time from \$log where \$pre_period \$filter group by f_user order by start_time desc)### t group by f_ user; create temporary table rpt_tmptbl_2 as select f_user, min(start_time) as start_time from ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as f_user, min(dtime) as start_time from \$log where \$filter group by f_user order by start_time desc)### t group by f_user; select f_ user, from_dtime(min(start_time)) as start_time from rpt_tmptbl_2 where f_user is not null and not exists (select 1 from rpt_tmptbl_1 where rpt_tmptbl_2.f_user=rpt_tmptbl_1.f_user) group by f_user order by start_time desc

Dataset Name	Description	Log Category
newthing-New-Devices	New devices	fct-traffic

drop

```
idea if exists rpt_tmptbl_1;
drop
table if exists rpt_tmptbl_2; create temporary table rpt_tmptbl_1 as
select
hostname,
os,
srcip,
fctver
from
```

###(select hostname, os, srcip, fctver from \$log where \$pre_period \$filter and hostname is not null group by hostname, os, srcip, fctver order by hostname)### t group by hostname, os, srcip, fctver; create temporary table rpt_tmptbl_2 as select hostname, os, srcip, fctver from ###(select hostname, os, srcip, fctver from \$log where \$filter and hostname is not null group by hostname, os, srcip, fctver order by hostname)### t group by hostname, os, srcip, fctver; select hostname, max(fctos_to_devtype(os)) as devtype, string_agg(distinct os, '/') as os_agg, string_agg(distinct ipstr(srcip), '/') as srcip_agg, string_agg(distinct fctver, '/') as fctver_agg from rpt_tmptbl_2 where not exists (select 1 from rpt_tmptbl_1 where rpt_tmptbl_ 2.hostname=rpt_tmptbl_1.hostname) group by hostname order by hostname

```
Dataset Name
                                   Description
                                                                                    Log Category
newthing-New-Software-Installed
                                   New software installed
                                                                                    fct-traffic
drop
  table if exists rpt tmptbl 1;
drop
 table if exists rpt tmptbl 2; create temporary table rpt tmptbl 1 as
select
 srcproduct,
 hostname
from
  ###(select srcproduct, hostname from $log where $pre period $filter and nullifna(srcproduct)
is not null group by srcproduct, hostname order by srcproduct) ### t group by srcproduct, host-
name; create temporary table rpt tmptbl 2 as select srcproduct, hostname from ###(select
srcproduct, hostname from $log where $filter and nullifna(srcproduct) is not null group by
srcproduct, hostname order by srcproduct)### t group by srcproduct, hostname; select
srcproduct, string agg(distinct hostname, ',') as host agg from rpt tmptbl 2 where not exists
(select 1 from rpt tmptbl 1 where rpt tmptbl 2.srcproduct=rpt tmptbl 1.srcproduct) group by
srcproduct order by srcproduct
```

```
Dataset Name
                                   Description
                                                                                   Log Category
 newthing-New-Security-Threats
                                   New security threats
                                                                                   virus
drop
  table if exists rpt_tmptbl_1;
drop
  table if exists rpt_tmptbl_2; create temporary table rpt_tmptbl_1 as
select
 threat name,
 cat id,
 srcip
from
  (
    ###(select app as threat_name, 1 as cat_id, srcip from $log-app-ctrl where $pre_period
$filter and nullifna(app) is not null and lower(appcat)='botnet' group by threat name, cat id,
srcip)### union all ###(select virus as threat name, 2 as cat id, srcip from $log-virus where
$pre period $filter and nullifna(virus) is not null group by threat name, cat id, srcip)###
union all ###(select attack as threat name, 3 as cat id, srcip from $log-attack where $pre
period $filter and nullifna(attack) is not null group by threat name, cat id, srcip)###) t;
create temporary table rpt_tmptbl_2 as select daystamp, threat_name, cat_id, srcip from (###
(select $DAY_OF_MONTH as daystamp, app as threat_name, 1 as cat_id, srcip from $log-app-ctrl
where $filter and nullifna(app) is not null and lower(appcat)='botnet' group by daystamp,
threat name, cat id, srcip order by daystamp)### union all ###(select $DAY OF MONTH as day-
```

stamp, 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
                                   New Queried Botnet C&C Domains and IPs
newthing-dns-Botnet-Domain-IP
                                                                                    dns
drop
  table if exists rpt tmptbl 1;
drop
  table if exists rpt_tmptbl_2; create temporary table rpt_tmptbl_1 as
select
 domain,
 malware_type,
 action s as action,
 srcip,
 sevid
from
  ###(select coalesce(botnetdomain, ipstr(botnetip)) as domain, cast('Botnet C&C' as char(32))
as malware type, (case when action='block' then 'Blocked' when action='redirect' then 'Redir-
```

as malware_type, (case when action='block' then 'Blocked' when action='redirect' then 'Redirected' else 'Passed' end) as action_s, srcip, (CASE WHEN level IN ('critical', 'alert', 'emergency') THEN 5 WHEN level='error' THEN 4 WHEN level='warning' THEN 3 WHEN level='notice' THEN 2 ELSE 1 END) as sevid, coalesce(botnetdomain, ipstr(botnetip)) as sources_s, count(*) as total_num from \$log where \$pre_period \$filter and (botnetdomain is not null or botnetip is not null) group by domain, action_s, srcip, sevid order by sevid desc)### t group by domain, malware_type, action, srcip, sevid; create temporary table rpt_tmptbl_2 as select domain, malware_type, action_s as action, srcip, sevid from ###(select coalesce(botnetdomain, ipstr (botnetip)) as domain, cast('Botnet C&C' as char(32)) as malware_type, (case when action-='block' then 'Blocked' when action='redirect' then 'Redirected' else 'Passed' end) as action_ s, srcip, (CASE WHEN level IN ('critical', 'alert', 'emergency') THEN 5 WHEN level='error' THEN 4 WHEN level='warning' THEN 3 WHEN level='notice' THEN 2 ELSE 1 END) as sevid, coalesce (botnetdomain, ipstr(botnetip)) as sources_s, count(*) as total_num from \$log where \$filter and (botnetdomain is not null or botnetip is not null) group by domain, action_s, srcip, sevid order by sevid desc)### t group by domain, malware_type, action, srcip, sevid; select domain, srcip, sevid, (CASE sevid WHEN 5 THEN 'Critical' WHEN 4 THEN 'High' WHEN 3 THEN 'Medium' WHEN '2' THEN 'Info' ELSE 'Low' END) as severity from rpt_tmptbl_2 where (domain is not null and not exists (select 1 from rpt_tmptbl_1 where rpt_tmptbl_2.domain=rpt_tmptbl_1.domain)) or (srcip is not null and not exists (select 1 from rpt_tmptbl_1 where rpt_tmptbl_2.srcip=rpt_ tmptbl 1.srcip)) group by domain, srcip, sevid order by sevid desc, domain

Dataset Name	Description	Log Category
newthing-New-Security-Threats- Timeline	New security threats timeline	virus
<pre>drop table if exists rpt_tmptbl_1 drop table if exists rpt_tmptbl_2 select threat_name, cat_id, srcip</pre>	; ; create temporary table rpt_tmptbl_1 as	
from (
	ame, 1 as cat_id, srcip from \$log-app-ct ot null and lower(appcat)='botnet' group	··

lower(appcat)= (app) potnet srcip)### union all ###(select virus as threat name, 2 as cat id, srcip from \$log-virus where \$pre period \$filter and nullifna(virus) is not null group by threat name, cat id, srcip)### union all ###(select attack as threat_name, 3 as cat_id, srcip from \$log-attack where \$pre_ period \$filter and nullifna(attack) is not null group by threat name, cat id, srcip)###) t; create temporary table rpt tmptbl 2 as select timestamp, threat name, cat id, srcip from (### (select \$flex timestamp as timestamp, app as threat name, 1 as cat id, srcip from \$log-appctrl where \$filter and nullifna(app) is not null and lower(appcat)='botnet' group by timestamp, threat name, cat id, srcip order by timestamp)### union all ###(select \$flex timestamp as timestamp, virus as threat name, 2 as cat id, srcip from \$log-virus where \$filter and nullifna (virus) is not null group by timestamp, threat name, cat id, srcip order by timestamp) ### union all ### (select \$flex timestamp as timestamp, attack as threat name, 3 as cat id, srcip from \$log-attack where \$filter and nullifna(attack) is not null group by timestamp, threat name, cat id, srcip order by timestamp)###) t; select \$flex datetime (timestamp) as timescale, count(distinct srcip) as host num, (case cat id when 1 then 'Botnet' when 2 then 'Malware' when 3 then 'Attack' end) as threat cat from rpt tmptbl 2 where not exists (select 1 from rpt tmptbl 1 where rpt tmptbl 2.threat name=rpt tmptbl 1.threat name) group by timescale, cat id order by timescale, cat id

Dataset Name	Description	Log Category
newthing-New-Vulnerability	New vulnerabilities	fct-netscan

```
drop
  table if exists rpt_tmptbl_1;
drop
  table if exists rpt_tmptbl_2; create temporary table rpt_tmptbl_1 as
select
  vulnid,
  vulnname,
  vulnseverity,
  vulncat,
  hostname
from
```

###(select vulnid, vulnname, vulnseverity, vulncat, hostname from \$log where \$pre_period \$filter and nullifna(vulnname) is not null group by vulnid, vulnname, vulnseverity, vulncat, hostname)### t group by vulnid, vulnname, vulnseverity, vulncat, hostname; create temporary table rpt_tmptbl_2 as select vulnid, vulnname, vulnseverity, vulncat, hostname from ###(select vulnid, vulnname, vulnseverity, vulncat, hostname from \$log where \$filter and nullifna(vulnname) is not null group by vulnid, vulnname, vulnseverity, vulncat, hostname)### t group by vulnid, vulnname, vulnseverity, vulncat, hostname; select vulnname, (case when vulnseverity='Critical' then 5 when vulnseverity='High' then 4 when vulnseverity='Medium' then 3 when vulnseverity='Low' then 2 when vulnseverity='Info' then 1 else 0 end) as sev, vulnseverity, vulncat, count(distinct hostname) as host_num, cve_id from rpt_tmptbl_2 t1 left join fct_mdata t2 on t1.vulnid=t2.vid::int where not exists (select 1 from rpt_tmptbl_1 where t1.vulnid=rpt_ tmptbl_1.vulnid) group by vulnname, sev, vulnseverity, vulncat, cve_id order by sev desc, host num desc

Dataset Name	Description	Log Category
newthing-New-Vulnerability-Graph	New vulnerabilities (Graph)	fct-netscan
drop		
<pre>table if exists rpt_tmptbl_1;</pre>		
drop		
	create temporary table rpt_tmptbl_1	as
select		
vulnid,		
vulnname,		
vulnseverity, vulncat,		
hostname		
from		
	vulnseverity, vulncat, hostname from	\$log where \$pre period
	is not null group by vulnid, vulnname	
	vulnname, vulnseverity, vulncat, hos	
table rpt tmptbl 2 as select vul	lnid, vulnname, vulnseverity, vulncat	;, hostname from ###(select
	vulncat, hostname from \$log where \$f	
nname) is not null group by vul	nid, vulnname, vulnseverity, vulncat,	hostname)### t group by
vulnid, vulnname, vulnseverity,	vulncat, hostname; select vulnseveri	ty, count (distinct vul-
	1_2 where not exists (select 1 from r	· _ · _ · _ · _

tmptbl_2.vulnid=rpt_tmptbl_1.vulnid) group by vulnseverity order by (case when vulnseverity='Critical' then 5 when vulnseverity='High' then 4 when vulnseverity='Medium' then 3 when vulnseverity='Low' then 2 when vulnseverity='Info' then 1 else 0 end) desc

Dataset Name	Description	Log Category
newthing-System-Alerts	System Alerts	local-event

```
select
from_itime(itime) as timestamp,
msg
from
$log
where
$filter
and msg is not null
and level = 'critical'
order by
timestamp desc
```

Dataset Name	Description	Log Category
newthing-Configuration-Changes	Configuration Changes	event
<pre>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</pre>		

Description	Log Category
FortiGate Upgrades	event
]+) -> ([^)]+)\\)'	

```
Log Category
 Dataset Name
                                   Description
 newthing-User-Upgrades
                                                                                   fct-event
                                   User Upgrades
drop
  table if exists rpt_tmptbl_1;
drop
 table if exists rpt_tmptbl_2; create temporary table rpt_tmptbl_1 as
select
 fgtserial,
 hostname,
 deviceip,
 os,
 dtime
from
  ###(select distinct on (fgtserial, hostname) fgtserial, hostname, deviceip, os, dtime from
Slog where Spre period Sfilter and hostname is not null order by fgtserial, hostname, dtime
desc)### t; create temporary table rpt tmptbl 2 as select fgtserial, hostname, deviceip, os,
dtime from ###(select distinct on (fgtserial, hostname) fgtserial, hostname, deviceip, os,
dtime from $log where $filter and hostname is not null order by fgtserial, hostname, dtime
desc)### t; select distinct on (1, 2) t2.fgtserial as devid, t2.hostname, t2.deviceip, t1.os
as prev_os, t2.os as cur_os, from_dtime(t1.dtime) as time_s from rpt_tmptbl_2 t2 inner join
```

order by devid, t2.hostname, t1.dtime desc			
Dataset Name	Description	Log Category	
GTP-List-of-APN-Used	List of APNs Used	gtp	
<pre>select apn, from_dtime(min(first_seen)) as first_seen, from_dtime(max(last_seen)</pre>			

rpt_tmptbl_1 t1 on t2.fgtserial=t1.fgtserial and t2.hostname=t1.hostname and t2.os!=t1.os

) 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
<pre>select apn, sum(coalesce(`u-bytes`, 0)) as total_bytes from \$log where \$filter and nullifna(apn) is not null</pre>		

```
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 (
<pre>coalesce(`u-pkts`, 0)</pre>		
) as total_num		
from		
\$log		
where		
\$filter		
and nullifna(apn) is not null		
and status = 'traffic-count'		
group by		
apn		
having		
sum(
<pre>coalesce(`u-pkts`, 0)</pre>		
)& 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
<pre>select domain, malware_type, action, count(distinct srcip) as vict count(distinct sources_s) as</pre>	•	

```
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
<pre>select domain, malware_type, action, count(distinct srcip) as victims count(distinct sources_s) as sou sum(total_num) as total_num from ###({{FGT_DATASET_DNS_BOTNET_DOM</pre>		tion order by

total_num desc

Dataset Name	Description	Log Category
Dns-Detected-Botnet	Top Queried Botnet C&C Domains and IPs	dns
<pre>select domain, malware_type, action, count(distinct srcip) as victin count(distinct sources_s) as so sum(total_num) as total_num</pre>	•	
<pre>from ###({{FGT_DATASET_DNS_BOTNET_DO total_num desc</pre>	OMAINS}})### t group by domain, malware_type, ac	ction order by

Description	Log Category
Queried Botnet C&C Domains and IPs	dns
	·

```
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

dns-High-Risk-Source

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
<pre>select \$flex timescale(timestamp)</pre>	as timescale,	
<pre>sum(case when sevid = 5 then) as num_cri, sum(</pre>	total_num else 0 end	
<pre>case when sevid = 4 then) as num_hig, sum(</pre>	total_num else 0 end	
<pre>case when sevid = 3 then) as num_med, sum(</pre>	_	
<pre>case when sevid = 2 then) as num_inf, sum(</pre>	_	
<pre>case when sevid = 1 then) as num_low from</pre>	total_num else 0 end	
	as timestamp, (CASE WHEN level IN rror' THEN 4 WHEN level='warning' T	

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
<pre>select qname, count(*) as total_num from \$log where \$filter and qname is not null group by qname order by total_num desc</pre>		
Dataset Name	Description	Log Category
dns-Top-Domain-Lookup-Failure-Bar	Top Domain Lookup Failures	dns
<pre>select qname, srcip, count(*) as total_num from \$log where \$filter and qname is not null and (action = 'block' or logid_to_int(logid)= 54200) group by qname, srcip order by total_num desc</pre>		
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) = 54200
)
group by
qname,
srcip
order by
total_num desc
```

Dataset Name	Description	Log Category
dns-Query-Timeout	Query Timeout	dns
elect		
srcip,		
qname,		
count(*) as total_num		
irom —		
\$log		
here		
\$filter		
and srcip is not null		
and logid_to_int(logid) = 5	54200	
roup 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,
     get_fgt_role(devid, slot) as role,
     cast(
       sum(cpu ave) / count(*) as decimal(6, 0)
     ) as cpu ave,
     cast(
       sum(mem_ave)/ count(*) as decimal(6, 0)
     ) as mem ave,
     cast(
       sum(disk_ave)/ count(*) as decimal(6, 0)
      ) as disk ave,
      cast(
       sum(log rate) as decimal(10, 2)
      ) as log rate,
      cast(
       sum(sessions) as decimal(10, 0)
      ) as sessions,
      cast(
```

```
sum(sent kbps) as decimal(10, 0)
 ) as sent kbps,
 cast(
   sum(recv kbps) as decimal(10, 0)
 ) as recv kbps,
 cast(
   sum(transmit kbps) as decimal(10, 0)
 ) as transmit kbps,
 max(mem peak) as mem peak,
 max(disk peak) as disk peak,
 max(cpu peak) as cpu peak,
 cast(
  max(lograte_peak) as decimal(10, 2)
 ) as lograte peak,
 max(session peak) as session peak,
 max(transmit kbps peak) as transmit kbps peak,
 cast(
   sum(cps ave) as decimal(10, 0)
 ) as cps ave,
 sum(cps peak) as cps peak
from
  (
   select
     $flex timescale(timestamp) as hodex,
     devid,
     slot,
      sum(total cpu) / sum(count) cpu ave,
      sum(total mem) / sum(count) as mem ave,
      sum(total disk) / sum(count) as disk ave,
      sum (
       total trate + total erate + total orate
     )/ 100.00 / sum(count) as log rate,
      sum(totalsession) / sum(count) as sessions,
      sum(sent) / sum(count) as sent kbps,
      sum(recv) / sum(count) as recv kbps,
      sum(sent + recv) / sum(count) as transmit_kbps,
     max(mem peak) as mem peak,
     max(disk peak) as disk peak,
     max(cpu peak) as cpu peak,
     max(lograte_peak) / 100.00 as lograte_peak,
     max(session peak) as session peak,
     max(transmit peak) as transmit kbps peak,
     sum(cps) / sum(count) as cps ave,
     max(cps_peak) as cps_peak
   from
      ###({{FGT DATASET EVENT DISK LOGRATE CPU MEM}})### t where $filter-drilldown group
```

by hodex, devid, slot) t group by hodex, devid, role) t group by hodex order by hodex

	Description	Log Category
perf-stat-mem-usage-drilldown	Fortigate resource detail timeline	event
elect		
hodex,		
<pre>cast(sum(cpu ave) / count(*) as de</pre>		

```
) as cpu_ave,
 cast(
   sum(mem_ave)/ count(*) as decimal(6, 0)
  ) as mem ave,
 cast(
   sum(disk ave) / count(*) as decimal(6, 0)
 ) as disk ave,
 cast(
   sum(log rate) / count(*) as decimal(10, 2)
 ) as log_rate,
 cast(
   sum(sessions)/ count(*) as decimal(10, 0)
 ) as sessions,
 cast(
   sum(sent kbps)/ count(*) as decimal(10, 0)
 ) as sent kbps,
 cast(
   sum(recv kbps) / count(*) as decimal(10, 0)
  ) as recv kbps,
 cast(
   sum(transmit_kbps)/ count(*) as decimal(10, 0)
 ) as transmit_kbps,
 max(mem peak) as mem peak,
 max(disk_peak) as disk_peak,
 max(cpu_peak) as cpu_peak,
 max(lograte_peak) as lograte_peak,
 max(session peak) as session peak,
 max(transmit kbps peak) as transmit kbps peak,
 cast(
   sum(cps ave) / count(*) as decimal(10, 0)
 ) as cps ave,
 max(cps_peak) as cps_peak
from
  (
   select
     hodex,
     devid,
     get fgt role(devid, slot) as role,
     cast(
       sum(cpu ave)/ count(*) as decimal(6, 0)
     ) as cpu ave,
     cast(
       sum(mem ave)/ count(*) as decimal(6, 0)
     ) as mem ave,
     cast(
       sum(disk ave) / count(*) as decimal(6, 0)
     ) as disk ave,
     cast(
       sum(log rate) as decimal(10, 2)
      ) as log rate,
     cast(
       sum(sessions) as decimal(10, 0)
      ) as sessions,
      cast(
       sum(sent kbps) as decimal(10, 0)
      ) as sent kbps,
```

```
cast(
        sum(recv kbps) as decimal(10, 0)
      ) as recv kbps,
      cast(
        sum(transmit kbps) as decimal(10, 0)
      ) as transmit kbps,
      max(mem peak) as mem peak,
      max(disk peak) as disk peak,
     max(cpu peak) as cpu peak,
     cast(
       max(lograte peak) as decimal(10, 2)
      ) as lograte peak,
     max(session peak) as session peak,
     max(transmit kbps peak) as transmit kbps peak,
      cast(
       sum(cps ave) as decimal(10, 0)
      ) as cps ave,
      sum(cps_peak) as cps_peak
    from
      (
        select
          $flex timescale(timestamp) as hodex,
          devid,
          slot,
          sum(total_cpu) / sum(count) cpu_ave,
          sum(total mem) / sum(count) as mem ave,
          sum(total disk) / sum(count) as disk ave,
          sum(
            total trate + total erate + total orate
          )/ 100.00 / sum(count) as log rate,
          sum(totalsession) / sum(count) as sessions,
          sum(sent) / sum(count) as sent kbps,
          sum(recv) / sum(count) as recv kbps,
          sum(sent + recv) / sum(count) as transmit kbps,
          max(mem peak) as mem peak,
          max(disk_peak) as disk_peak,
          max(cpu peak) as cpu peak,
          max(lograte peak) / 100.00 as lograte peak,
          max(session peak) as session peak,
          max(transmit_peak) as transmit_kbps_peak,
          sum(cps) / sum(count) as cps ave,
          max(cps peak) as cps peak
        from
          ###({{FGT_DATASET_EVENT_DISK_LOGRATE_CPU_MEM}})### t where $filter-drilldown group
by hodex, devid, slot) t group by hodex, devid, role) t group by hodex order by hodex
```

```
Dataset NameDescriptionLog Categoryperf-stat-disk-usage-drilldownFortigate resource detail timelineeventselect<br/>hodex,<br/>cast(<br/>sum(cpu_ave)/ count(*) as decimal(6, 0)<br/>) as cpu_ave,<br/>cast(selecimal(6, 0)
```

```
sum(mem ave) / count(*) as decimal(6, 0)
 ) as mem_ave,
 cast(
   sum(disk ave) / count(*) as decimal(6, 0)
  ) as disk ave,
 cast(
  sum(log rate) / count(*) as decimal(10, 2)
 ) as log rate,
 cast(
   sum(sessions)/ count(*) as decimal(10, 0)
 ) as sessions,
 cast(
   sum(sent kbps)/ count(*) as decimal(10, 0)
 ) as sent kbps,
 cast(
   sum(recv kbps) / count(*) as decimal(10, 0)
 ) as recv_kbps,
 cast(
   sum(transmit kbps)/ count(*) as decimal(10, 0)
  ) as transmit_kbps,
 max(mem_peak) as mem_peak,
 max(disk_peak) as disk_peak,
 max(cpu peak) as cpu peak,
 max(lograte_peak) as lograte_peak,
 max(session_peak) as session_peak,
 max(transmit_kbps_peak) as transmit_kbps_peak,
 cast(
   sum(cps ave) / count(*) as decimal(10, 0)
 ) as cps ave,
 max(cps peak) as cps peak
from
  (
   select
     hodex,
     devid,
     get fgt role(devid, slot) as role,
     cast(
       sum(cpu ave) / count(*) as decimal(6, 0)
     ) as cpu ave,
     cast(
       sum(mem ave)/ count(*) as decimal(6, 0)
      ) as mem ave,
     cast(
       sum(disk_ave)/ count(*) as decimal(6, 0)
     ) as disk ave,
     cast(
       sum(log_rate) as decimal(10, 2)
     ) as log_rate,
     cast(
       sum(sessions) as decimal(10, 0)
      ) as sessions,
      cast(
       sum(sent kbps) as decimal(10, 0)
      ) as sent kbps,
      cast(
       sum(recv kbps) as decimal(10, 0)
```

```
) as recv kbps,
      cast(
       sum(transmit kbps) as decimal(10, 0)
      ) as transmit kbps,
     max(mem peak) as mem_peak,
     max(disk peak) as disk peak,
     max(cpu peak) as cpu peak,
     cast(
       max(lograte peak) as decimal(10, 2)
     ) as lograte peak,
     max(session peak) as session peak,
     max(transmit kbps peak) as transmit kbps peak,
     cast(
       sum(cps ave) as decimal(10, 0)
     ) as cps ave,
      sum(cps peak) as cps peak
   from
      (
       select
          $flex timescale(timestamp) as hodex,
         devid,
         slot,
          sum(total cpu) / sum(count) cpu ave,
          sum(total mem) / sum(count) as mem ave,
          sum(total_disk) / sum(count) as disk_ave,
          sum(
            total trate + total_erate + total_orate
          )/ 100.00 / sum(count) as log_rate,
          sum(totalsession) / sum(count) as sessions,
          sum(sent) / sum(count) as sent_kbps,
          sum(recv) / sum(count) as recv kbps,
          sum(sent + recv) / sum(count) as transmit kbps,
          max(mem peak) as mem peak,
          max(disk peak) as disk peak,
          max(cpu peak) as cpu peak,
         max(lograte_peak) / 100.00 as lograte_peak,
         max(session peak) as session peak,
          max(transmit peak) as transmit kbps peak,
          sum(cps) / sum(count) as cps ave,
         max(cps_peak) as cps_peak
        from
          ###({{FGT DATASET EVENT DISK LOGRATE CPU MEM}})### t where $filter-drilldown group
by hodex, devid, slot) t group by hodex, devid, role) t group by hodex order by hodex
```

Dataset Name	Description	Log Category
perf-stat-sessions-drilldown	Fortigate resource detail timeline	event
select		
hodex,		
cast (

```
sum(cpu_ave)/ count(*) as decimal(6, 0)
) as cpu_ave,
cast(
   sum(mem_ave)/ count(*) as decimal(6, 0)
) as mem ave,
```

```
cast(
   sum(disk_ave)/ count(*) as decimal(6, 0)
 ) as disk ave,
 cast(
   sum(log rate) / count(*) as decimal(10, 2)
 ) as log rate,
 cast(
   sum(sessions)/ count(*) as decimal(10, 0)
 ) as sessions,
 cast(
   sum(sent kbps)/ count(*) as decimal(10, 0)
  ) as sent kbps,
 cast(
   sum(recv kbps) / count(*) as decimal(10, 0)
 ) as recv kbps,
 cast(
   sum(transmit kbps)/ count(*) as decimal(10, 0)
 ) as transmit kbps,
 max(mem peak) as mem peak,
 max(disk_peak) as disk_peak,
 max(cpu_peak) as cpu_peak,
 max(lograte_peak) as lograte_peak,
 max(session peak) as session peak,
 max(transmit_kbps_peak) as transmit_kbps_peak,
 cast(
   sum(cps_ave)/ count(*) as decimal(10, 0)
 ) as cps ave,
 max(cps_peak) as cps_peak
from
  (
   select
     hodex,
     devid,
     get fgt role(devid, slot) as role,
     cast(
       sum(cpu ave) / count(*) as decimal(6, 0)
     ) as cpu ave,
      cast(
       sum(mem ave)/ count(*) as decimal(6, 0)
      ) as mem ave,
     cast(
       sum(disk ave) / count(*) as decimal(6, 0)
     ) as disk ave,
     cast(
       sum(log_rate) as decimal(10, 2)
     ) as log_rate,
     cast(
       sum(sessions) as decimal(10, 0)
      ) as sessions,
      cast(
       sum(sent kbps) as decimal(10, 0)
      ) as sent kbps,
      cast(
       sum(recv kbps) as decimal(10, 0)
      ) as recv kbps,
      cast(
```

```
sum(transmit kbps) as decimal(10, 0)
  ) as transmit_kbps,
  max(mem_peak) as mem_peak,
  max(disk peak) as disk peak,
 max(cpu peak) as cpu peak,
  cast(
  max(lograte peak) as decimal(10, 2)
 ) as lograte peak,
 max(session peak) as session peak,
 max(transmit kbps peak) as transmit kbps peak,
 cast(
   sum(cps ave) as decimal(10, 0)
 ) as cps ave,
  sum(cps_peak) as cps_peak
from
  (
   select
      $flex timescale(timestamp) as hodex,
     devid,
      slot,
      sum(total_cpu) / sum(count) cpu_ave,
      sum(total_mem) / sum(count) as mem ave,
      sum(total disk) / sum(count) as disk ave,
      sum(
        total_trate + total_erate + total_orate
      )/ 100.00 / sum(count) as log rate,
      sum(totalsession) / sum(count) as sessions,
      sum(sent) / sum(count) as sent kbps,
      sum(recv) / sum(count) as recv kbps,
      sum(sent + recv) / sum(count) as transmit kbps,
      max(mem peak) as mem peak,
     max(disk peak) as disk peak,
     max(cpu peak) as cpu peak,
     max(lograte peak) / 100.00 as lograte peak,
      max(session peak) as session peak,
     max(transmit peak) as transmit kbps peak,
      sum(cps) / sum(count) as cps ave,
     max(cps peak) as cps peak
    from
```

###({{FGT_DATASET_EVENT_DISK_LOGRATE_CPU_MEM}})### 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
<pre>select hodex, cast(sum(cpu_ave)/ count(*) as dec) as cpu_ave, cast(sum(mem_ave)/ count(*) as dec) as mem_ave, cast(sum(disk_ave)/ count(*) as dec</pre>	imal(6, 0)	

```
) as disk ave,
 cast(
   sum(log_rate) / count(*) as decimal(10, 2)
  ) as log rate,
 cast(
   sum(sessions)/ count(*) as decimal(10, 0)
 ) as sessions,
 cast(
   sum(sent kbps)/ count(*) as decimal(10, 0)
 ) as sent kbps,
 cast(
   sum(recv kbps)/ count(*) as decimal(10, 0)
 ) as recv kbps,
 cast(
  sum(transmit kbps)/ count(*) as decimal(10, 0)
 ) as transmit kbps,
 max(mem_peak) as mem_peak,
 max(disk_peak) as disk_peak,
 max(cpu peak) as cpu peak,
 max(lograte_peak) as lograte_peak,
 max(session_peak) as session_peak,
 max(transmit_kbps_peak) as transmit_kbps_peak,
 cast(
   sum(cps ave) / count(*) as decimal(10, 0)
 ) as cps_ave,
 max(cps_peak) as cps_peak
from
  (
   select
     hodex,
     devid,
     get fgt role(devid, slot) as role,
     cast(
       sum(cpu ave) / count(*) as decimal(6, 0)
     ) as cpu ave,
     cast(
       sum(mem ave) / count(*) as decimal(6, 0)
      ) as mem ave,
     cast(
       sum(disk_ave) / count(*) as decimal(6, 0)
     ) as disk_ave,
     cast(
       sum(log rate) as decimal(10, 2)
     ) as log_rate,
     cast(
       sum(sessions) as decimal(10, 0)
     ) as sessions,
     cast(
       sum(sent kbps) as decimal(10, 0)
      ) as sent kbps,
      cast(
       sum(recv kbps) as decimal(10, 0)
      ) as recv kbps,
      cast(
       sum(transmit kbps) as decimal(10, 0)
      ) as transmit kbps,
```

```
max(mem peak) as mem peak,
  max(disk_peak) as disk_peak,
 max(cpu_peak) as cpu_peak,
 cast(
   max(lograte_peak) as decimal(10, 2)
 ) as lograte peak,
 max(session peak) as session peak,
 max(transmit kbps peak) as transmit kbps peak,
 cast(
   sum(cps ave) as decimal(10, 0)
 ) as cps ave,
  sum(cps peak) as cps peak
from
  (
   select
      $flex timescale(timestamp) as hodex,
     devid,
      slot,
      sum(total cpu) / sum(count) cpu ave,
      sum(total mem) / sum(count) as mem ave,
      sum(total_disk) / sum(count) as disk_ave,
      sum(
        total trate + total erate + total orate
      )/ 100.00 / sum(count) as log rate,
      sum(totalsession) / sum(count) as sessions,
      sum(sent) / sum(count) as sent kbps,
      sum(recv) / sum(count) as recv kbps,
      sum(sent + recv) / sum(count) as transmit kbps,
      max(mem peak) as mem peak,
      max(disk peak) as disk peak,
      max(cpu peak) as cpu peak,
     max(lograte_peak) / 100.00 as lograte peak,
     max(session peak) as session peak,
     max(transmit peak) as transmit kbps peak,
      sum(cps) / sum(count) as cps ave,
      max(cps peak) as cps peak
    from
      ###({{FGT DATASET EVENT DISK LOGRATE CPU MEM}})### t where $filter-drilldown group
```

###({{FGT_DATASET_EVENT_DISK_LOGRATE_CPU_MEM}})### 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
<pre>select hodex, cast(sum(cpu_ave)/ count(*)) as cpu_ave, cast(sum(mem_ave)/ count(*)) as mem_ave, cast(sum(disk_ave)/ count(*)) as disk_ave, cast(</pre>	as decimal(6, 0)	

```
sum(log rate) / count(*) as decimal(10, 2)
 ) as log_rate,
 cast(
   sum(sessions)/ count(*) as decimal(10, 0)
  ) as sessions,
 cast(
  sum(sent kbps)/ count(*) as decimal(10, 0)
 ) as sent kbps,
 cast(
   sum(recv kbps) / count(*) as decimal(10, 0)
 ) as recv kbps,
 cast(
   sum(transmit kbps)/ count(*) as decimal(10, 0)
 ) as transmit kbps,
 max(mem peak) as mem peak,
 max(disk peak) as disk peak,
 max(cpu_peak) as cpu_peak,
 max(lograte_peak) as lograte_peak,
 max(session peak) as session peak,
 max(transmit kbps peak) as transmit kbps peak,
 cast(
  sum(cps_ave) / count(*) as decimal(10, 0)
 ) as cps ave,
 max(cps_peak) as cps_peak
from
  (
   select
     hodex,
     devid,
     get fgt role(devid, slot) as role,
     cast(
       sum(cpu ave) / count(*) as decimal(6, 0)
     ) as cpu_ave,
     cast(
       sum(mem ave) / count(*) as decimal(6, 0)
      ) as mem ave,
      cast(
       sum(disk ave) / count(*) as decimal(6, 0)
      ) as disk ave,
     cast(
       sum(log rate) as decimal(10, 2)
      ) as log rate,
     cast(
       sum(sessions) as decimal(10, 0)
     ) as sessions,
     cast(
       sum(sent kbps) as decimal(10, 0)
      ) as sent_kbps,
      cast(
       sum(recv kbps) as decimal(10, 0)
      ) as recv kbps,
      cast(
       sum(transmit kbps) as decimal(10, 0)
      ) as transmit kbps,
     max(mem peak) as mem peak,
     max(disk peak) as disk peak,
```

```
max(cpu_peak) as cpu peak,
 cast(
   max(lograte_peak) as decimal(10, 2)
  ) as lograte_peak,
 max(session peak) as session peak,
 max(transmit kbps peak) as transmit kbps peak,
 cast(
   sum(cps ave) as decimal(10, 0)
 ) as cps ave,
 sum(cps peak) as cps peak
from
  (
   select
      $flex timescale(timestamp) as hodex,
     devid,
      slot,
      sum(total_cpu) / sum(count) cpu_ave,
      sum(total mem) / sum(count) as mem ave,
      sum(total disk) / sum(count) as disk ave,
      sum(
        total_trate + total_erate + total_orate
      )/ 100.00 / sum(count) as log_rate,
      sum(totalsession) / sum(count) as sessions,
      sum(sent) / sum(count) as sent kbps,
      sum(recv) / sum(count) as recv_kbps,
      sum(sent + recv) / sum(count) as transmit_kbps,
      max(mem peak) as mem peak,
      max(disk peak) as disk peak,
     max(cpu peak) as cpu peak,
     max(lograte peak) / 100.00 as lograte peak,
     max(session peak) as session peak,
     max(transmit peak) as transmit kbps peak,
      sum(cps) / sum(count) as cps ave,
      max(cps peak) as cps peak
    from
```

###({{FGT_DATASET_EVENT_DISK_LOGRATE_CPU_MEM}})### 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
<pre>select hodex, cast(sum(cpu_ave)/ count(*) as) as cpu_ave, cast(sum(mem_ave)/ count(*) as) as mem_ave, cast(sum(disk_ave)/ count(*) a) as disk_ave, cast(sum(log_rate)/ count(*) a) as log_rate,</pre>	decimal(6, 0) s decimal(6, 0)	

```
cast(
   sum(sessions)/ count(*) as decimal(10, 0)
 ) as sessions,
 cast(
   sum(sent kbps)/ count(*) as decimal(10, 0)
 ) as sent kbps,
 cast(
   sum(recv kbps) / count(*) as decimal(10, 0)
 ) as recv kbps,
 cast(
   sum(transmit kbps)/ count(*) as decimal(10, 0)
 ) as transmit kbps,
 max(mem_peak) as mem_peak,
 max(disk_peak) as disk peak,
 max(cpu peak) as cpu peak,
 max(lograte peak) as lograte peak,
 max(session_peak) as session_peak,
 max(transmit_kbps_peak) as transmit_kbps_peak,
 cast(
   sum(cps ave) / count(*) as decimal(10, 0)
 ) as cps_ave,
 max(cps_peak) as cps_peak
from
  (
   select
     hodex,
     devid,
     get fgt role(devid, slot) as role,
     cast(
       sum(cpu ave) / count(*) as decimal(6, 0)
     ) as cpu ave,
     cast(
       sum(mem ave) / count(*) as decimal(6, 0)
      ) as mem ave,
      cast(
       sum(disk ave) / count(*) as decimal(6, 0)
      ) as disk ave,
      cast(
       sum(log rate) as decimal(10, 2)
      ) as log_rate,
     cast(
       sum(sessions) as decimal(10, 0)
      ) as sessions,
     cast(
       sum(sent_kbps) as decimal(10, 0)
      ) as sent kbps,
     cast(
       sum(recv_kbps) as decimal(10, 0)
      ) as recv kbps,
      cast(
       sum(transmit kbps) as decimal(10, 0)
      ) as transmit kbps,
     max(mem peak) as mem peak,
     max(disk peak) as disk peak,
     max(cpu_peak) as cpu_peak,
      cast(
```

```
max(lograte_peak) as decimal(10, 2)
  ) as lograte_peak,
 max(session_peak) as session_peak,
 max(transmit kbps peak) as transmit kbps peak,
 cast(
   sum(cps ave) as decimal(10, 0)
 ) as cps ave,
 sum(cps peak) as cps peak
from
  (
   select
     $flex timescale(timestamp) as hodex,
     devid,
     slot,
      sum(total cpu) / sum(count) cpu ave,
      sum(total mem) / sum(count) as mem ave,
      sum(total_disk) / sum(count) as disk_ave,
      sum(
        total trate + total erate + total orate
      )/ 100.00 / sum(count) as log_rate,
      sum(totalsession) / sum(count) as sessions,
      sum(sent) / sum(count) as sent_kbps,
      sum(recv) / sum(count) as recv kbps,
      sum(sent + recv) / sum(count) as transmit kbps,
      max(mem_peak) as mem_peak,
     max(disk_peak) as disk_peak,
      max(cpu peak) as cpu peak,
     max(lograte peak) / 100.00 as lograte peak,
     max(session peak) as session peak,
     max(transmit peak) as transmit kbps peak,
      sum(cps) / sum(count) as cps ave,
     max(cps peak) as cps peak
    from
      ###({{FGT DATASET EVENT DISK LOGRATE CPU MEM}})### 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
<pre>select devid, get_fgt_role(devid, slot) as ro cast(sum(cpu_ave) / count(*) as dec) as cpu_ave, cast(sum(mem_ave) / count(*) as dec) as mem_ave, cast(sum(disk_ave) / count(*) as de) as disk_ave, cast(sum(log_rate) as decimal(10,) as log_rate, cast(</pre>	<pre>imal(6, 0) imal(6, 0) cimal(6, 0)</pre>	

```
sum(sessions) as decimal(10, 0)
 ) as sessions,
 cast(
   sum(sent kbps) as decimal(10, 0)
  ) as sent kbps,
 cast(
   sum(recv kbps) as decimal(10, 0)
 ) as recv kbps,
 cast(
   sum(transmit kbps) as decimal(10, 0)
 ) as transmit kbps,
 max(mem peak) as mem peak,
 max(disk peak) as disk peak,
 max(cpu_peak) as cpu_peak,
 cast(
  max(lograte peak) as decimal(10, 2)
 ) as lograte_peak,
 max(session peak) as session peak,
 max(transmit kbps peak) as transmit kbps peak
from
  (
   select
     devid,
     slot,
     sum(total_cpu) / sum(count) as cpu_ave,
     sum(total_mem) / sum(count) as mem_ave,
      sum(total_disk) / sum(count) as disk_ave,
     sum(
       total trate + total erate + total orate
     )/ 100.00 / sum(count) as log_rate,
      sum(totalsession) / sum(count) as sessions,
      sum(sent) / sum(count) as sent kbps,
      sum(recv) / sum(count) as recv kbps,
      sum(sent + recv) / sum(count) as transmit kbps,
     max(mem peak) as mem peak,
     max(disk_peak) as disk_peak,
     max(cpu peak) as cpu peak,
     max(lograte peak) / 100.00 as lograte peak,
     max(session peak) as session peak,
     max(transmit_peak) as transmit_kbps_peak
   from
      ###({{FGT DATASET EVENT DISK LOGRATE CPU MEM}})### 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
<pre>select devid, get_fgt_role(devid, slot) as cast(sum(cpu_ave) / count(*) as) as cpu_ave, cast(sum(mem ave) / count(*) as</pre>	decimal(6, 0)	

```
) as mem_ave,
 cast(
   sum(disk ave)/ count(*) as decimal(6, 0)
  ) as disk ave,
 cast(
   sum(log rate) as decimal(10, 2)
 ) as log rate,
 cast(
   sum(sessions) as decimal(10, 0)
 ) as sessions,
 cast(
   sum(sent kbps) as decimal(10, 0)
 ) as sent kbps,
 cast(
   sum(recv kbps) as decimal(10, 0)
 ) as recv kbps,
 cast(
   sum(transmit kbps) as decimal(10, 0)
  ) as transmit kbps,
 max(mem_peak) as mem_peak,
 max(disk_peak) as disk_peak,
 max(cpu_peak) as cpu_peak,
 cast(
  max(lograte peak) as decimal(10, 2)
 ) as lograte_peak,
 max(session_peak) as session_peak,
 max(transmit kbps peak) as transmit kbps peak
from
  (
   select
     devid,
     slot,
     sum(total cpu) / sum(count) as cpu ave,
      sum(total mem) / sum(count) as mem ave,
      sum(total disk) / sum(count) as disk ave,
     sum(
       total trate + total erate + total orate
     )/ 100.00 / sum(count) as log rate,
      sum(totalsession) / sum(count) as sessions,
     sum(sent) / sum(count) as sent kbps,
     sum(recv) / sum(count) as recv kbps,
      sum(sent + recv) / sum(count) as transmit kbps,
     max(mem peak) as mem peak,
     max(disk_peak) as disk_peak,
     max(cpu_peak) as cpu_peak,
     max(lograte peak) / 100.00 as lograte peak,
     max(session peak) as session peak,
     max(transmit_peak) as transmit_kbps_peak
   from
      ###({{FGT DATASET EVENT DISK LOGRATE CPU MEM}})### t group by devid, slot) t group by
devid, role order by devid, role
```

Dataset Name	Description	Log Category
perf-stat-usage-details-drilldown- master	Fortigate resource summary view	event

```
select
 devid,
 get_fgt_role(devid, slot) as role,
 cast(
  sum(cpu ave) / count(*) as decimal(6, 0)
 ) as cpu ave,
 cast(
   sum(mem ave) / count(*) as decimal(6, 0)
 ) as mem ave,
 cast(
   sum(disk ave) / count(*) as decimal(6, 0)
  ) as disk ave,
 cast(
   sum(log rate) as decimal(10, 2)
 ) as log rate,
 cast(
   sum(sessions) as decimal(10, 0)
 ) as sessions,
 cast(
   sum(sent kbps) as decimal(10, 0)
 ) as sent kbps,
 cast(
   sum(recv kbps) as decimal(10, 0)
 ) as recv_kbps,
 cast(
   sum(transmit kbps) as decimal(10, 0)
 ) as transmit kbps,
 max(mem peak) as mem peak,
 max(disk peak) as disk peak,
 max(cpu peak) as cpu peak,
 cast(
  max(lograte peak) as decimal(10, 2)
 ) as lograte peak,
 max(session peak) as session peak,
 max(transmit kbps peak) as transmit kbps peak
from
  (
   select
     devid,
     slot,
     sum(total_cpu) / sum(count) as cpu ave,
      sum(total mem) / sum(count) as mem ave,
      sum(total_disk) / sum(count) as disk_ave,
     sum(
       total_trate + total_erate + total_orate
      )/ 100.00 / sum(count) as log rate,
      sum(totalsession) / sum(count) as sessions,
      sum(sent) / sum(count) as sent_kbps,
      sum(recv) / sum(count) as recv_kbps,
      sum(sent + recv) / sum(count) as transmit kbps,
     max(mem peak) as mem peak,
     max(disk peak) as disk peak,
     max(cpu peak) as cpu peak,
     max(lograte peak) / 100.00 as lograte peak,
     max(session peak) as session peak,
     max(transmit peak) as transmit kbps peak
```

from

```
###({{FGT_DATASET_EVENT_DISK_LOGRATE_CPU_MEM})### t group by devid, slot) t group by
devid, role order by devid, role
```

Dataset Name	Description	Log Catego
incident-Incident-Count-by-Status	Incident status distribution	
select status, count(*) as cnt from \$incident group by status order by status		
Dataset Name	Description	Log Catego
incident-Incident-Count-by-Status- Donut	Incident status distribution	
select status, count(*) as cnt		

Dataset Name	Description	Log Category
incident-Open-Incident-Count- Timeline	Incident count by status over time	
<pre>select \$flex_timescale(agg_time) as max(num_sta_draft) as num_st max(num_sta_analysis) as num max(num_sta_response) as num max(num_sta_closed) as num_s max(num_sta_cancelled) as nu</pre>	a_draft, _sta_analysis, _sta_response, ta_closed,	
from \$incident history		
where		
<pre>\$cust_time_filter(agg_time)</pre>		
group by		
hodex		

order by hodex

Dataset Name	Description	Log Category
incident-Closed-Incident-Count- Timeline	Incident count by status over time	
<pre>select \$flex_timescale(agg_time) as how max(num_sta_draft) as num_sta_draft) as num_sta_draft max(num_sta_analysis) as num_sta max(num_sta_response) as num_sta_draft max(num_sta_closed) as num_sta_draft max(num_sta_cancelled) as num_sta_from \$incident_history where \$cust_time_filter(agg_time) group by hodex order by hodex</pre>	caft, a_analysis, a_response, closed,	

Dataset Name	Description	Log Category
Top-10-Apps-by-Bandwidth	Top applications by bandwidth usage	traffic
<pre>select app_group_name(app) a sum(coalesce(sentbyte,) as bandwidth, sum(coalesce(rcvdbyte,) as traffic_in, sum(coalesce(sentbyte,</pre>	0)+ coalesce(rcvdbyte, 0) 0)	
<pre>) as traffic_out, count(*) as sessions from \$log where \$filter and (</pre>		
<pre>logflag&1>0) and nullifna(app) is group by app_group having sum(</pre>	not null	
	0)+ coalesce(rcvdbyte, 0)	

<pre>coalesce(nullifaa('user'), nullifaa('user'), 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 rom sing sing(iogflagi>0) and srcip is not null roup by user_src, srcip aving sum(coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)) & gft 0 roder by bandwidth desc Dataset Name Description Log Category Top-10-Applications-by-Number-of- Users Top for app_group_name(app) as app_group, count(distinct user_src) as number rom ##f(select coalesce(nullifna('user'), nullifna('unauthuser'), ipstr('arcip')) as user_src, pp, appcat from Slog where Sfilter and (logflagi>0) and nullifna(app) is not null group by ae_src, app, appcat)##t t group by app_group order by number desc Dataset Name Description Log Category</pre>	Dataset Name	Description	Log Category
<pre>coalesce(nullifaa('user'), nullifaa('user'), 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 rom sing sing(iogflagi>0) and srcip is not null roup by user_src, srcip aving sum(coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)) & gft 0 roder by bandwidth desc Dataset Name Description Log Category Top-10-Applications-by-Number-of- Users Top for app_group_name(app) as app_group, count(distinct user_src) as number rom ##f(select coalesce(nullifna('user'), nullifna('unauthuser'), ipstr('arcip')) as user_src, pp, appcat from Slog where Sfilter and (logflagi>0) and nullifna(app) is not null group by ae_src, app, appcat)##t t group by app_group order by number desc Dataset Name Description Log Category</pre>	Top-10-User-by-Bandwidth	Top users by bandwidth usage	traffic
<pre>nullifna('user'), nullifna('user'), ipst('srcip')) as user_src, sum(coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(sentbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out room %log index for for for sum(coalesce(sentbyte, 0)) as traffic_out room %lig index is not null room by user_src, srcip sum(coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)) & gtr 0 rder by bandwidth desc Dataset Name Description Log Category Top-10-Applications-by-Number-of- users traffic telect app_group_name(app) as app_group, count(distinct user_src) as number room ##f(select coalesce(nullifna('user'), nullifna('unauthuser'), ipstr('srcip')) as user_src, pp, appcat from %log where %filter and (logflag&l>0) and nullifna(app) is not null group by user_src, app, appcat)##f t group by app_group order by number desc Dataset Name Description Log Category</pre>	select		
<pre>nullifna('unauthuser'), ijstr('scrip')) as user_src, srcip, sum(coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out rom %log here %filter and crcip is not null roup by user_src, srcip aving sum(coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)) % gf: 0 roder by bandwidth desc Dataset Name Description Log Category Top-10-Applications-by-Number-of- Top Applications by number of users traffic users relet app-group_name(app) as app_group, count(distinct user_src) as number rom ##f(select coalesce(nullifna('user'), nullifna('unauthuser'), ipstr('srcip')) as user_src, pp, appcat from \$log where \$filter and (logflagsl>0) and nullifna(app) is not null group by user_src, app, appcat)### group by app_group order by number desc Dataset Name Description Log Category the stription of users traffic the stription of user stription of user_src, app-appcat from \$log where \$filter and (logflagsl>0) and nullifna(app) is not null group by as user_src, app, appcat)### group by app_group order by number desc Dataset Name Description Log Category dataset Name Description Log Category dataset Name Description Stription of user stription</pre>			
<pre>ipstr('arcip')) as user_src, srcip, sum(ccalesce(sentbyte, 0)+ ccalesce(rcvdbyte, 0)) as bandwidth, sum(ccalesce(rcvdbyte, 0)) as traffic_in, sum(ccalesce(sentbyte, 0)) as traffic_out rom Slag here \$filter and (logflags1>0) and srcip is not null roup by user_src, srcip aving sum(ccalesce(sentbyte, 0)+ ccalesce(rcvdbyte, 0)) a strific rom Top-10-Applications-by-Number-of- Top Applications by number of users traffic Users traffic traffi</pre>			
<pre>) sa user_src, srclp, sum(coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out rom slog there sfilter and (logflagal>0) and srcip is not null roup by user_src, srcip aud coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)) % gc; 0 reder by bandwidth desc Dataset Name Description Log Category Top-10-Applications-by-Number-of- Users traffic users traffic telect app_group_name(app) as app_group, count(distinct user_src) as number rom ##f (select coalesce(nullifna('user'), nullifna('unauthuser'), ipstr('srcip')) as user_src, pp, appcat from %log where %filter and (logflag\$1>0) and nullifna(app) is not null group by user_src, app, appcat)### t group by app_group order by number desc Dataset Name Description Log Category</pre>			
<pre>srcip, sum(coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(sentbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out room Slog hhere Sfilter and (logflags120) and srcip is not null roup by user_src, srcip aving sum(coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)) & gt; 0 rder by bandwidth desc Dataset Name Description Log Category Top-10-Applications-by-Number-of- Users traffic user set to the set the set to the set to</pre>			
<pre>coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out rom Slog there Sfilter and (logflags1>0) and scip is not null roup by user_src, srcip aving sum(coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)) & git 0 roter by bandwidth desc Dataset Name Description Top Applications by number of users traffic elect app_group_name(app) as app_group, count distinct user_src) as number rom ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, pp, appcat from \$log where Sfilter and (logflag&l>0) and nullifna(app) is not null group by user_src, app, appcat)### t group by app_group order by number desc Dataset Name Description Log Category </pre>	—		
<pre>) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out rom Slog here Sfilter and (logflag&1>0) and srcip is not null rroup by user_src, ercip aving sum(coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0))& gt; 0 rrder by bandwidth desc Dataset Name Description Log Category Top-10-Applications-by-Number-of- Top Applications by number of users traffic Users traffic users traffic users traffic traffic</pre>	sum(
<pre>sum(coalesce(rcvdbyte, 0) as traffic_in, sum(coalesce(sentbyte, 0) as traffic_out rom Slog there Sfilter and (logflags1>0) and srcip is not null roup by user_src, srcip sum(coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0))& gt; 0 roder by bandwidth desc Dataset Name Description Log Category Top-10-Applications-by-Number-of- Top Applications by number of users traffic relect app_group_name(app) as app_group, count(distinct user_src) as number rom ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, pp, appcat from Slog where Sfilter and (logflags1>0) and nullifna(app) is not null group by user_src, app, appcat)### t group by app_group order by number desc Dataset Name Description Log Category</pre>	_	esce(rcvdbyte, 0)	
<pre>coalesce(rcvdbyte, 0) } as traffic_in, sum(coalesce(sentbyte, 0) } as traffic_out rom %log here %filter and (logflag%l>0 } and srcip is not null roup by user_src, srcip sum(coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0) } & gt: 0 rder by bandwidth desc Dataset Name Description Log Category Top-10-Applications-by-Number-of- Top Applications by number of users traffic users telect app_group_name(app) as app_group, count (distinct user_src) as number rom ###(select coalesce(nullifna('user'), nullifna('unauthuser'), ipstr('srcip')) as user_src, pp, appcat from %log where %filter and (logflag%l>0) and nullifna(app) is not null group by user_src, asser_are, app, appcat)### t group by app_group order by number desc Dataset Name Description Log Category Log Catego</pre>			
<pre>) se traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out rom Slog there Sfilter and (logflagilo)) and srcip is not null roup by user_src, srcip saving sum(coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0))% gt; 0 roder by bandwidth desc Dataset Name Description Log Category Top-10-Applications-by-Number-of- Top Applications by number of users traffic users telect app_group_name(app) as app_group, count (distinct user_src) as number rom ###(select coalesce(nullifna('user'), nullifna('unauthuser'), ipstr('srcip')) as user_src, pp, appcat from Slog where Sfilter and (logflagilo) and nullifna(app) is not null group by user_src, app, appcat)### t group by app_group order by number desc Dataset Name Description Log Category</pre>			
<pre>sum(coalesce(sentbyte, 0)) as traffic_out from Slog there \$filter and (logflag&100) and srcip is not null roup by user_src, srcip auding sum(coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)) & gt; 0 rder by bandwidth desc Dataset Name Description Log Category Top-10-Applications-by-Number-of- Top Applications by number of users traffic Users traffic elect app_group_name(app) as app_group, count(distinct user_src) as number rem ###f(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, upp, appcat from \$log where \$filter and (logflag&1>0) and nullifna(app) is not null group by user_src, app, appcat)### t group by app_group order by number desc Dataset Name Description Log Category</pre>			
<pre>) as traffic_out irom %log here \$filter and (logflag\$1>0) and srcip is not null roup by user_src, srcip aving sum(coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)) & gt; 0 rder by bandwidth desc Dataset Name Description Log Category Top-10-Applications-by-Number-of- Users Top Applications by number of users traffic users relect app_group_name(app) as app_group, count(distinct user_src) as number irom ### (select coalesce(nullifna('user'), nullifna('unauthuser'), ipstr('srcip')) as user_src, pp, appcat from %log where \$filter and (logflag\$1>0) and nullifna(app) is not null group by iser_src, app, appcat)### t group by app_group order by number desc Dataset Name Description Log Category</pre>	—		
<pre>irom \$log here \$filter and (logflag&l>0 } and srcip is not null rroup by user_src, srcip user_src, srcip coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0))& gt; 0 roder by bandwidth desc Dataset Name Description Log Category rop-10-Applications-by-Number-of- Top Applications by number of users traffic users telect app_group_name(app) as app_group, count(distinct user_src) as number irom ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, pp, appcat from %log where \$filter and (logflag&l>0) and nullifna(app) is not null group by ser_src, app, appcat)### t group by app_group order by number desc Dataset Name Description Log Category </pre>	coalesce(sentbyte, 0)		
<pre>\$log here \$filter and (logflag&1>0 } and srcip is not null roup by user_src, srcip aaving sum(coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)) & gt; 0 order by bandwidth desc Dataset Name Description Log Category Top-10-Applications-by-Number-of- Top Applications by number of users traffic users traffic users traffic select app_group_name(app) as app_group, count(distinct user_src) as number rom ### (select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, upp, appcat from %log where \$filter and (logflag&1>0) and nullifna(app) is not null group by user_src, app, appcat)### t group by app_group order by number desc Dataset Name Description Log Category</pre>) as traffic_out		
<pre>shere \$filter and (logflag&1>0) and srcip is not null rroup by user_src, srcip sum(coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0))& gt: 0 order by bandwidth desc Dataset Name Description Log Category traffic select app_group_name(app) as app_group, count(distinct user_src) as number rrom ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, gp, appcat from \$filter and (logflag&1>0) and nullifna(app) is not null group by user_src, app, appcat)### t group by app_group order by number desc Dataset Name Description Log Category</pre>	irom		
<pre>\$filter and (logflag&1>0) and srcip is not null roup by user_src, srcip aving sum(ccalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)) & gt; 0 rder by bandwidth desc Dataset Name Description Log Category Top-10-Applications-by-Number-of- Top Applications by number of users traffic Users traffic users traffic traff</pre>			
and (logflag&1>0) and srcip is not null roup by user_src, srcip iaving sum(coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0))& gt; 0 trder by bandwidth descLog CategoryDataset NameDescriptionLog CategoryTop-10-Applications-by-Number-of- usersTop Applications by number of userstraffictelect app_group_name(app) as app_group, count (distinct user_src) as number irom ###(select coalesce(nullifna('user'), nullifna('unauthuser'), ipstr('srcip')) as user_src, pp, appcat from \$log where \$filter and (logflag&1>0) and nullifna(app) is not null group by user_src, app, appcat)### t group by app_group order by number descDataset NameDescriptionLog Category			
<pre>logflag&1>0 } and srcip is not null rroup by user_src, srcip uaving sum(coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)) & gt; 0 rrder by bandwidth desc Dataset Name Description Log Category Top-10-Applications-by-Number-of- Top Applications by number of users traffic users traffic users traffic telect app_group_name(app) as app_group, count(distinct user_src) as number rrom ###(select coalesce(nullifna('user'), nullifna('unauthuser'), ipstr('srcip')) as user_src, pp, appcat from \$log where \$filter and (logflag&1>0) and nullifna(app) is not null group by user_src, app, appcat)### t group by app_group order by number desc Dataset Name Description</pre>			
<pre> and srcip is not null group by user_src, srcip coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0))& gt; 0 rder by bandwidth desc Dataset Name Description Log Category Top-10-Applications-by-Number-of- Top Applications by number of users traffic users telect app_group_name(app) as app_group, count(distinct user_src) as number irom ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, pp, appcat from \$log where \$filter and (logflag&i>0) and nullifna(app) is not null group by user_src, app, appcat)### t group by app_group order by number desc Dataset Name Description Log Category Log Category </pre>	-		
proup by user_src, srcip awing sum(coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)) & gt; 0 order by bandwidth desc Dataset Name Description Log Category Top-10-Applications-by-Number-of- Users Top Applications by number of users traffic select app_group_name(app) as app_group, count (distinct user_src) as number "**** (select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, upp, appcat from \$log where \$filter and (logflag&l>0) and nullifna(app) is not null group by user_src, app, appcat) *** t group by app_group order by number desc Dataset Name Description Log Category			
user_src, srcip laving sum(coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0))& gt; 0 order by bandwidth desc Description Log Category Dataset Name Description Log Category Top-10-Applications-by-Number-of- Users Top Applications by number of users traffic #elect app_group_name(app) as app_group, count (distinct user_src) as number traffic ('unauthuser'), ipstr('srcip')) as user_src, ipp, appcat from \$log where \$filter and (logflag&l>0) and nullifna(app) is not null group by iser_src, app, appcat)### t group by app_group order by number desc Dataset Name Description Log Category	and srcip is not null		
srcip iaving sum(coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)) & gt; 0 order by bandwidth desc Dataset Name Description Log Category Top-10-Applications-by-Number-of- Top Applications by number of users traffic Users traffic eelect app_group_name(app) as app_group, count(distinct user_src) as number from ###*(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, upp, appcat from \$log where \$filter and (logflag&l>0) and nullifna(app) is not null group by user_src, app, appcat)### t group by app_group order by number desc Dataset Name Description Log Category	group by		
avving sum(coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0))& gt; 0 order by bandwidth descDescriptionLog CategoryDataset NameDescriptionTop Applications by number of userstrafficTop-10-Applications-by-Number-of- UsersTop Applications by number of userstrafficreelect app_group_name(app) as app_group, count (distinct user_src) as number from ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, upp, apcat from \$log where \$filter and (logflag\$1>0) and nullifna(app) is not null group by user_src, app, appcat)### t group by app_group order by number descDataset NameDescriptionLog Category			
sum(coalesce (sentbyte, 0)+ coalesce (rcvdbyte, 0)) & gt; 0 order by bandwidth desc Description Log Category Top-10-Applications-by-Number-of- Top Applications by number of users traffic select app_group_name(app) as app_group, count (distinct user_src) as number rest of the select (light of the select coalesce (nullifna (`user`), nullifna (`unauthuser`), ipstr(`srcip`)) as user_src, upp, appcat from \$log where \$filter and (logflag&l>0) and nullifna (app) is not null group by user_src, app, appcat) ### t group by app_group order by number desc Dataset Name Description Log Category	-		
coalesce (sentbyte, 0) + coalesce (rcvdbyte, 0))& gt; 0order bybandwidth descDataset NameDescriptionLog CategoryTop-10-Applications-by-Number-of- UsersTop Applications by number of users userstrafficrelect app_group_name(app) as app_group, count (distinct user_src) as numbertraffia (`unauthuser`), ipstr(`srcip`)) as user_src, upp, appcat from \$log where \$filter and (logflag&l>0) and nullifna (app) is not null group by user_src, app, appcat)### t group by app_group order by number descDataset NameDescriptionLog Category			
)& gt; 0 Description Log Category Dataset Name Description Tog Category Top-10-Applications-by-Number-of- Users Top Applications by number of users traffic telect app_group_name(app) as app_group, count (distinct user_src) as number traffic ('user'), nullifna ('unauthuser'), ipstr('srcip')) as user_src, ### (select coalesce (nullifna ('user'), nullifna ('unauthuser'), ipstr('srcip')) as user_src, pp, appcat from \$log where \$filter and (logflag&1>0) and nullifna (app) is not null group by user_src, app, appcat)### t group by app_group order by number desc Dataset Name Description Log Category	-	esce(rcvdbyte, 0)	
bandwidth descDataset NameDescriptionLog CategoryTop-10-Applications-by-Number-of- UsersTop Applications by number of userstrafficrelect app_group_name(app) as app_group, count (distinct user_src) as number from ### (select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, npp, appcat from \$log where \$filter and (logflag&l>0) and nullifna(app) is not null group by user_src, app, appcat)### t group by app_group order by number descDataset NameDescriptionLog Category			
Dataset NameDescriptionLog CategoryTop-10-Applications-by-Number-of- UsersTop Applications by number of userstrafficreelect app_group_name (app) as app_group, count (distinct user_src) as numberselectselect### (select coalesce (nullifna (`user`), nullifna (`unauthuser`), ipstr(`srcip`)) as user_src, upp, appcat from \$log where \$filter and (logflag&l>0) and nullifna (app) is not null group by user_src, app, appcat)### t group by app_group order by number descLog CategoryDataset NameDescriptionLog Category	order by		
Top-10-Applications-by-Number-of- UsersTop Applications by number of userstrafficselect app_group_name(app) as app_group, count(distinct user_src) as number	bandwidth desc		
Users welect app_group_name(app) as app_group, count(distinct user_src) as number from ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, upp, appcat from \$log where \$filter and (logflag&1>0) and nullifna(app) is not null group by user_src, app, appcat)### t group by app_group order by number desc Dataset Name Description Log Category	Dataset Name	Description	Log Category
<pre>app_group_name(app) as app_group, count(distinct user_src) as number from ####(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, upp, appcat from \$log where \$filter and (logflag&1>0) and nullifna(app) is not null group by iser_src, app, appcat)### t group by app_group order by number desc Dataset Name Description Log Category</pre>		Top Applications by number of users	traffic
<pre>count(distinct user_src) as number from ####(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, upp, appcat from \$log where \$filter and (logflag&1>0) and nullifna(app) is not null group by user_src, app, appcat)### t group by app_group order by number desc Dataset Name Description Log Category</pre>	select		
<pre>from ####(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, pp, appcat from \$log where \$filter and (logflag&1>0) and nullifna(app) is not null group by user_src, app, appcat)### t group by app_group order by number desc Dataset Name Description Log Category</pre>			
<pre>###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, upp, appcat from \$log where \$filter and (logflag&1>0) and nullifna(app) is not null group by user_src, app, appcat)### t group by app_group order by number desc Dataset Name Description Log Category</pre>	—	umber	
upp, appcat from \$log where \$filter and (logflag&1>0) and nullifna(app) is not null group byuser_src, app, appcat)### t group by app_group order by number descDataset NameDescriptionLog Category		`usor`\ nullifna(`unouthusor`\ instr(`s	rain'll as yoor arc
Dataset Name Description Log Category	app, appcat from \$log where \$fil	lter and (logflag&1>0) and nullifna(app)	
	aser_src, app, appcat)### t grou	up by app_group order by number desc	
Top-10-User-by-Session Top user by session count traffic	Dataset Name	Description	Log Category
	Top-10-User-by-Session	Top user by session count	traffic

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

Dataset Name

Top-10-Apps-by-Session

Top applications by bandwidth usage

Description

Log Category 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
  $loq
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
Applications-by-Risk-Level	Applications by Risk Level	traffic

```
select
  app_group_name(app) as app_group,
  min(id) as id,
  appcat,
  max(risk) as d_risk,
  (
    case when max(risk)= '5' then 'Critical' when max(risk)= '4' then 'High' when max(risk)=
 '3' then 'Medium' when max(risk)= '2' then 'Low' else 'Info' end
 ) as risk_level,
  sum(sessions) as sessions,
  sum(sent) as sent,
  sum(received) as received,
  sum(bandwidth) as bandwidth
from
```

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

Dataset Name	Description	Log Category
soc-Event-vs-Incident-Today-Trend	Events vs Incidents Today Trend	
select		
item,		
num cur,		
num pre,		
num diff		
from		
(
select		
'Events' as item,		
num_cur,		
num_pre,		
(num_cur - num_pre) as num	_diff	
from		
(
select		
(
select		
count(*)		
from		
\$event		
where Squat time filter	alorttimo TODAY)	
<pre>\$cust_time_filter(</pre>	aleittime, iODAI)	
) as num_cur,		
select		
count (*)		
from		
\$event		
where		
	alerttime, YESTERDAY)	
) as num pre	• •	
) t		

```
union all
    select
     'Incidents' as item,
     num_cur,
     num_pre,
      (num_cur - num_pre) as num_diff
    from
      (
        select
          (
            select
             count(*)
           from
              $incident
            where
              $cust_time_filter(createtime, TODAY)
          ) as num_cur,
          (
            select
              count(*)
            from
              $incident
            where
              $cust_time_filter(createtime, YESTERDAY)
          ) as num_pre
      ) t
 ) t
order by
 item
```

Dataset Name	Description	Log Category
soc-Event-vs-Incident-History-Trend	Events vs Incidents History Trend	
select item,		
num_cur,		
num_pre, num_diff		
from (
<pre>select 'Events' as item,</pre>		
num_cur,		
num_pre, (num_cur - num_pre) as nur	n_diff	
from (
select (
select count(*)		
from		
\$event where		
<pre>\$cust_time_filter</pre>	(alerttime)	

```
) as num_cur,
          (
            select
              count(*)
            from
              $event
            where
              $cust_time_filter(alerttime, LAST_N_PERIOD, 1)
          ) as num_pre
      ) t
    union all
    select
     'Incidents' as item,
     num_cur,
     num pre,
      (num_cur - num_pre) as num_diff
    from
      (
        select
         (
            select
              count(*)
            from
              $incident
            where
             $cust_time_filter(createtime)
          ) as num cur,
          (
            select
              count(*)
            from
              $incident
            where
              $cust_time_filter(createtime, LAST_N_PERIOD, 1)
          ) as num pre
      ) t
  ) t
order by
 item
```

Dataset Name	Description	Log Category
soc-Event-vs-Incident-Trend	Events vs Incidents Trend	
<pre>select t1.item, t1.num_cur as num_today, t1.num_pre as num_yesterday, t1.num_diff as num_diff1, t2.num_cur as num_this_period, t2.num_pre as num_last_period, t2.num_diff as num_diff2 from (select 'Events' as item,</pre>		

```
num_cur,
   num_pre,
    (num_cur - num_pre) as num_diff
  from
    (
      select
       (
          select
           count(*)
          from
            $event
          where
            $cust_time_filter(alerttime, TODAY)
        ) as num_cur,
        (
          select
            count(*)
          from
            $event
          where
            $cust_time_filter(alerttime, YESTERDAY)
        ) as num_pre
   ) t
  union all
 select
   'Incidents' as item,
   num cur,
   num pre,
    (num_cur - num_pre) as num_diff
 from
    (
      select
        (
          select
            count(*)
          from
            $incident
          where
           $cust_time_filter(createtime, TODAY)
        ) as num_cur,
        (
          select
           count(*)
         from
            $incident
          where
            $cust_time_filter(createtime, YESTERDAY)
        ) as num_pre
   ) t
) t1 full
join (
 select
   'Events' as item,
   num cur,
   num_pre,
    (num_cur - num_pre) as num_diff
```

```
from
      (
        select
          (
           select
              count(*)
            from
              $event
           where
              $cust_time_filter(alerttime)
          ) as num cur,
          (
           select
              count(*)
           from
              $event
           where
              $cust_time_filter(alerttime, LAST_N_PERIOD, 1)
          ) as num_pre
     ) t
    union all
   select
     'Incidents' as item,
     num_cur,
     num_pre,
     (num cur - num pre) as num diff
    from
      (
        select
          (
           select
              count(*)
            from
              $incident
           where
              $cust_time_filter(createtime)
          ) as num_cur,
          (
            select
             count(*)
            from
              $incident
           where
              $cust_time_filter(createtime, LAST_N_PERIOD, 1)
          ) as num_pre
     ) t
  ) t2 on t1.item = t2.item
order by
 t1.item
```

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

(

```
CASE severity WHEN 0 THEN 'Critical' WHEN 1 THEN 'High' WHEN 2 THEN 'Medium' WHEN 3 THEN

'Low' ELSE NULL END

) as sev,

count(*) as num_events

from

$event

group by

severity

order by

severity

Dataset Name Description Log Category

soc-Total-Event-by-Severity-History Total Events by Severity History
```

```
select
 dom,
 (
    CASE severity WHEN 0 THEN 'Critical' WHEN 1 THEN 'High' WHEN 2 THEN 'Medium' WHEN 3 THEN
'Low' ELSE NULL END
 ) as sev,
  sum(num_events) as num_events
from
  (
    select
      dom,
     unnest(agg_sev) as severity,
      unnest(agg num) as num events
    from
      (
        select
          $DAY OF MONTH(agg time) as dom,
          array[0,
          1,
          2,
          3] as agg sev,
          array[max(num_sev_critical),
          max(num_sev_high),
          max(num_sev_medium),
          max(num_sev_low) ] as agg_num
        from
          $event_history
        where
          $cust time filter(agg time)
        group by
          dom
        order by
          dom
      ) t
  ) t
group by
 dom,
 severity
order by
 dom,
  severity
```

Dataset Name	Description	Log Category
soc-Total-Event-by-Severity-Category	Total Events Count by Severity and Category	
select		
CASE severity WHEN U THEN 'C 'Low' ELSE NULL END	ritical' WHEN 1 THEN 'High' WHEN 2 THEN 'Medium	' WHEN 3 THEN
) as sev,		
triggername,		
count(*) as num events		
- From		
\$event		
group by		
severity,		
triggername		
order by		
severity, triggername		
Dataset Name	Description	Log Category
soc-Total-Incident-by-Severity	Total Incidents by Severity	
select severity,		
count(*) as num_inc		
from		
\$incident		
group by		
severity		
order by		
severity		
Dataset Name	Description	Log Category
and Total Event ve Incident History		
soc-Total-Event-vs-Incident-History	Total Events vs Incidents History	
	Total Events vs Incidents History	
select		
select coalesce(t1.hodex, t2.hodex) a	as hodex,	
select coalesce(t1.hodex, t2.hodex) a coalesce(num_event_total, 0) a	as hodex, as num_event_total,	
<pre>select coalesce(t1.hodex, t2.hodex) a coalesce(num_event_total, 0) a coalesce(num_inc_total, 0) as</pre>	as hodex, as num_event_total, num_inc_total,	
select coalesce(t1.hodex, t2.hodex) a coalesce(num_event_total, 0) a coalesce(num_inc_total, 0) as coalesce(num_event_high, 0) as	as hodex, as num_event_total, num_inc_total,	
<pre>select coalesce(t1.hodex, t2.hodex) a coalesce(num_event_total, 0) a coalesce(num_inc_total, 0) as coalesce(num_event_high, 0) as from (</pre>	as hodex, as num_event_total, num_inc_total,	
<pre>select coalesce(t1.hodex, t2.hodex) a coalesce(num_event_total, 0) a coalesce(num_inc_total, 0) as coalesce(num_event_high, 0) as from (select</pre>	as hodex, as num_event_total, num_inc_total, s num_event_high	
<pre>select coalesce(t1.hodex, t2.hodex) a coalesce(num_event_total, 0) a coalesce(num_inc_total, 0) as coalesce(num_event_high, 0) as from (select \$flex_timescale(agg_time)</pre>	as hodex, num_inc_total, num_event_high as hodex,	
<pre>select coalesce(t1.hodex, t2.hodex) a coalesce(num_event_total, 0) a coalesce(num_inc_total, 0) as coalesce(num_event_high, 0) as from (select \$flex_timescale(agg_time) max(num_total) as num_even</pre>	as hodex, num_inc_total, num_event_high as hodex, nt_total,	
<pre>select coalesce(t1.hodex, t2.hodex) a coalesce(num_event_total, 0) a coalesce(num_inc_total, 0) as coalesce(num_event_high, 0) as from (select \$flex_timescale(agg_time) max(num_total) as num_even max(num_sev_critical + num</pre>	as hodex, num_inc_total, num_event_high as hodex,	
<pre>select coalesce(t1.hodex, t2.hodex) a coalesce(num_event_total, 0) a coalesce(num_inc_total, 0) as coalesce(num_event_high, 0) as from (select \$flex_timescale(agg_time) max(num_total) as num_even max(num_sev_critical + num from</pre>	as hodex, num_inc_total, num_event_high as hodex, nt_total,	
<pre>select coalesce(t1.hodex, t2.hodex) a coalesce(num_event_total, 0) a coalesce(num_inc_total, 0) as coalesce(num_event_high, 0) as from (select \$flex_timescale(agg_time) max(num_total) as num_even max(num_sev_critical + num</pre>	as hodex, num_inc_total, num_event_high as hodex, nt_total,	
<pre>select coalesce(t1.hodex, t2.hodex) a coalesce(num_event_total, 0) a coalesce(num_inc_total, 0) as coalesce(num_event_high, 0) as from (select \$flex_timescale(agg_time) max(num_total) as num_even max(num_sev_critical + num from \$event_history where</pre>	as hodex, num_inc_total, num_inc_total, num_event_high as hodex, nt_total, n_sev_high) as num_event_high	
<pre>select coalesce(t1.hodex, t2.hodex) a coalesce(num_event_total, 0) a coalesce(num_inc_total, 0) as coalesce(num_event_high, 0) as from (select \$flex_timescale(agg_time) max(num_total) as num_even max(num_sev_critical + num from \$event_history</pre>	as hodex, num_inc_total, num_inc_total, num_event_high as hodex, nt_total, n_sev_high) as num_event_high	
<pre>select coalesce(t1.hodex, t2.hodex) a coalesce(num_event_total, 0) a coalesce(num_inc_total, 0) as coalesce(num_event_high, 0) as from (select \$flex_timescale(agg_time) max(num_total) as num_even max(num_sev_critical + num from \$event_history where \$cust_time_filter(agg_time)</pre>	as hodex, num_inc_total, num_inc_total, num_event_high as hodex, nt_total, n_sev_high) as num_event_high	

```
hodex
 ) t1 full
 join (
   select
     $flex_timescale(agg_time) as hodex,
     max(
      num_sev_high + num_sev_medium + num_sev_low
     ) as num_inc_total
   from
     $incident_history
   where
     $cust_time_filter(agg_time)
   group by
     hodex
   order by
     hodex
 ) t2 on t1.hodex = t2.hodex
order by
 hodex
```

Dataset Name	Description	Log Category
soc-Incident-List	List of Incidents	
<pre>select incid_to_str(incid) as incnum, from_itime(createtime) as times category, severity, status, endpoint from \$incident where \$cust_time_filter(createtime) order by</pre>	stamp,	
createtime desc		
-	Description	Log Category
createtime desc	Description Incidents by Severity	Log Category

incnum desc

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

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

Dataset Name	Description	Log Category
soc-Incident-by-Category-Unresolved	Unresolved Incidents by Category	
select		
category,		
count(*) as incnum		
from		
\$incident		
where		
<pre>\$cust_time_filter(createtime)</pre>		
and status not in ('closed', '	cancelled')	
group by		

group by category

order by

incnum desc

Dataset Name	Description	Log Category
soc-Incident-by-Severity-Unresolved	Unresolved Incidents by Severity	
<pre>select severity, count(*) as incnum from \$incident where \$cust_time_filter(createtime) and status not in ('closed', ' group by severity order by incnum desc</pre>	cancelled')	
Dataset Name	Description	Log Category
soc-Incident-Timeline-by-Category	Incidents Timeline by Category	
<pre>select \$flex_timescale(agg_time) as h max(num_cat_cat1) as num_cat1, max(num_cat_cat2) as num_cat2, max(num_cat_cat3) as num_cat3, max(num_cat_cat4) as num_cat4,</pre>	odex,	

```
max(num_cat_cat5) as num_cat5,
max(num_cat_cat6) as num_cat6
from
    $incident_history
where
    $cust_time_filter(agg_time)
group by
    hodex
order by
    hodex
```

Dataset Name	Description	Log Category
soc-Incident-List-Unresolved	List of Unresolved Incidents	
select		
<pre>incid_to_str(incid) as inc</pre>	num,	
from_itime(createtime) as	timestamp,	
severity,		
status,		
endpoint,		
description		
from		
\$incident		
where		
\$cust_time_filter(createti	me)	
and status not in ('closed	', 'cancelled')	
order by		

severity desc

Dataset Name	Description	Log Category
fex-RSRQ-timeline	FortiExtender RSRQ timeline	event
<pre>select \$flex_timescale(timestamp) as hodex, cast(sum(rsrq_sum) / sum(count) as decimal(18, 2)) 'dB' as rsrq from ####(select \$flex_timestamp(dtime) as timestamp, sum(to_number(rsrq, '999999.99')) as rsrq_ sum, sum(to_number(sinr, '999999.99')) as sinr_sum, count(*) as count from \$log where \$filter and logid='0111046409' group by timestamp order by timestamp desc)### t group by hodex order by hodex desc</pre>		
Dataset Name	Description	Log Category
fex-SINR-timeline	FortiExtender SINR timeline	event
<pre>select \$flex_timescale(timesta cast(sum(sinr_sum) / sum(co) 'dB' as sinr from</pre>	mp) as hodex, ount) as decimal(18, 0)	

####(select \$flex_timestamp(dtime) as timestamp, sum(to_number(rsrq, '999999.99')) as rsrq_

sum, sum(to_number(sinr, '999999.99')) as sinr_sum, count(*) as count from \$log where \$filter and logid='0111046409' group by timestamp order by timestamp desc)### t group by hodex order by hodex desc

Dataset Name	Description	Log Category
fgt-device-monitoring-inventory	FortiGate Device Monitoring Inventory	event
<pre>select devname, (' ' devid) as id_devid, ip, platform, os, '1' as total_num from \$func - fgt - inventory as t1 where exists (select 1 from devtable t2 where \$dev_filter and t2.devid = t1.devid</pre>		
) order by devname		

Dataset Name	Description	Log Category
fgt-inventory-hardware	FortiGate Monitoring Inventory Hardware	event
<pre>select platform, count(*) as total_num from \$func - fgt - inventory as t1 where exists (select 1 from devtable t2 where \$dev_filter and t2.devid = t1.devid) group by platform order by total_num desc</pre>		
Dataset Name	Description	Log Category
fgt-inventory-software	FortiGate Monitoring Inventory Software	event

```
select
 'FortiOS' as sf_name,
 (platform || ' ' || os) as firmware,
 count(*) as total_num
from
  $func - fgt - inventory as t1
where
 exists (
   select
     1
   from
     devtable t2
   where
     $dev filter
     and t2.devid = t1.devid
  )
group by
 platform,
 os
order by
 total_num desc
```

Dataset Name	Description	Log Category
cup-utilization-timeline-for-each-device	FortiGate cpu utilization timeline	event

```
select
```

```
$flex timescale(timestamp) as hodex,
 devid,
 cast(
  sum(total_cpu) / sum(count) as decimal(6, 0)
 ) as cpu ave,
 cast(
   sum(total_mem) / sum(count) as decimal(6, 0)
 ) as mem_ave,
 cast(
  sum(total disk) / sum(count) as decimal(6, 0)
 ) as disk_ave,
 cast(
   sum(sent) / sum(count) as decimal(10, 0)
 ) as sent_kbps,
 cast(
  sum(recv) / sum(count) as decimal(10, 0)
 ) as recv kbps
from
```

```
###({{FGT_DATASET_EVENT_DISK_LOGRATE_CPU_MEM}})### t where $filter-drilldown group by hodex,
devid order by hodex
```

Dataset Name	Description	Log Category
status-timeline-by-device-cpu- utilization	FortiGate cpu summary view	event
select devid, cast(

```
sum(total_cpu)/ sum(count) as decimal(6, 0)
) as cpu_ave,
max(cpu_peak) as cpu_peak
from
```

###({{FGT DATASET EVENT DISK LOGRATE CPU MEM}})### t group by devid order by cpu peak desc

Dataset Name	Description	Log Category
event-cpu-utilization-dev	FortiGate cpu summary view	event
select		
devid, cast(

```
sum(total_cpu)/ sum(count) as decimal(6, 0)
```

) as cpu_ave,

```
max(cpu_peak) as cpu_peak
```

```
from
```

###({{FGT_DATASET_EVENT_DISK_LOGRATE_CPU_MEM}})### t group by devid order by cpu_peak desc

Dataset Name	Description	Log Category
memory-utilization-timeline-for-each-	FortiGate cpu utilization timeline	event
device		

```
select
 $flex_timescale(timestamp) as hodex,
 devid,
 cast(
  sum(total_cpu) / sum(count) as decimal(6, 0)
 ) as cpu_ave,
 cast(
   sum(total mem) / sum(count) as decimal(6, 0)
 ) as mem ave,
 cast(
   sum(total disk) / sum(count) as decimal(6, 0)
 ) as disk ave,
 cast(
   sum(sent) / sum(count) as decimal(10, 0)
 ) as sent kbps,
 cast(
   sum(recv) / sum(count) as decimal(10, 0)
 ) as recv kbps
from
```

###({{FGT_DATASET_EVENT_DISK_LOGRATE_CPU_MEM}})### t where \$filter-drilldown group by hodex, devid order by hodex

Dataset Name	Description	Log Category
status-timeline-by-device-mem- utilization	FortiGate memory summary view	event
<pre>select devid, cast(sum(total_mem)/ sum(count) as) as mem_ave,</pre>	decimal(6, 0)	

```
max(mem_peak) as mem_peak
from
```

###({{FGT_DATASET_EVENT_DISK_LOGRATE_CPU_MEM}})### t group by devid order by mem_peak desc

Dataset Name	Description	Log Category
event-mem-utilization-dev	FortiGate memory summary view	event
<pre>select devid, cast(sum(total_mem) / sum(count) as) as mem_ave, max(mem_peak) as mem_peak from</pre>	decimal(6, 0)	

###({{FGT_DATASET_EVENT_DISK_LOGRATE_CPU_MEM}})### t group by devid order by mem_peak desc

Dataset Name	Description	Log Category
disk-utilization-timeline-for-each- device	FortiGate cpu utilization timeline	event

select

```
$flex timescale(timestamp) as hodex,
 devid,
 cast(
  sum(total_cpu) / sum(count) as decimal(6, 0)
 ) as cpu_ave,
 cast(
   sum(total_mem) / sum(count) as decimal(6, 0)
 ) as mem ave,
 cast(
   sum(total disk) / sum(count) as decimal(6, 0)
 ) as disk ave,
 cast(
   sum(sent) / sum(count) as decimal(10, 0)
 ) as sent kbps,
 cast(
   sum(recv) / sum(count) as decimal(10, 0)
 ) as recv kbps
from
```

###({{FGT_DATASET_EVENT_DISK_LOGRATE_CPU_MEM}})### t where \$filter-drilldown group by hodex, devid order by hodex

Dataset Name	Description	Log Category
status-timeline-by-device-disk- utilization	FortiGate disk summary view	event
<pre>select devid, cast(sum(total_disk) / sum(count)) as disk_ave, max(disk_peak) as disk_peak</pre>	as decimal(6, 0)	

from

###({{FGT_DATASET_EVENT_DISK_LOGRATE_CPU_MEM}})### t group by devid order by disk_peak desc

Dataset Name	Description	Log Category
event-disk-utilization-dev	FortiGate disk summary view	event
<pre>select devid, cast(sum(total_disk) / sum(count) as disk_ave, max(disk_peak) as disk_peak</pre>) as decimal(6, 0)	

from

###({{FGT_DATASET_EVENT_DISK_LOGRATE_CPU_MEM}})### t group by devid order by disk_peak desc

Dataset Name	Description	Log Category
event-total-session-summary	FortiGate Total Sessions	event
<pre>select devid, max(session_peak) as max_session cast(sum(totalsession) / sum(count)) as sessions, max(cps_peak) as cps_peak, cast(sum(cps) / sum(count) as decime) as cps_ave</pre>	as decimal(10, 0)	
<pre>from ###({{FGT_DATASET_EVENT_DISK_LOG desc</pre>	GRATE_CPU_MEM}})### t group by devid order	by max_session

Dataset Name	Description	Log Category
event-session-rate-summary	FortiGate Session Rate	event
<pre>select devid, max(cps_peak) as max_rate from ###({{FGT_DATASET_EVENT_DI;</pre>	SK_LOGRATE_CPU_MEM}})### t group by de	vid order by max_rate desc
Dataset Name	Description	Log Category
event-session-summary-dev	FortiGate Total Sessions	event

```
select
  devid,
  max(session_peak) as max_session,
  cast(
    sum(totalsession) / sum(count) as decimal(10, 0)
  ) as sessions,
  max(cps_peak) as cps_peak,
  cast(
```

```
sum(cps)/ sum(count) as decimal(10, 0)
) as cps_ave
from
```

###({{FGT_DATASET_EVENT_DISK_LOGRATE_CPU_MEM}})### t group by devid order by max_session
desc

Dataset Name	Description	Log Category
fgt-intf-down-timeline-for-each-device	FortiGate Interface Down Timeline	event
<pre>select \$flex_timescale(timestamp) as h devid, sum(total num) as total num</pre>	odex,	
<pre>sum(total_num) as total_num from ###(select \$flex_timestamp as timestamp, devid, status, count(*) as total_num from \$log where \$filter and logid_to_int(logid)=20099 and status='DOWN' group by timestamp, devid, status)### t where \$filter-drilldown group by hodex, devid order by hodex</pre>		
Dataset Name	Description	Log Category

FortiGate Interface Down by Device

```
select
  devid,
  status,
  sum(total_num) as total_num
from
```

fgt-intf-down-timeline-by-device

###(select \$flex_timestamp as timestamp, devid, status, count(*) as total_num from \$log
where \$filter and logid_to_int(logid)=20099 and status='DOWN' group by timestamp, devid,
status)### t group by devid, status order by total_num desc

Dataset Name	Description	Log Category
fgt-intf-down-dev-donut	FortiGate Interface Down by Device	event
—	num mp as timestamp, devid, status, count(*) as to int(logid)=20099 and status='DOWN' group by t	—
	d, status order by total_num desc	-
status)### t group by devia Dataset Name fgt-intf-down-dev-tbl		Log Category event

event

where \$filter and logid_to_int(logid)=20099 and status='DOWN' group by timestamp, devid, status)### t group by devid, status order by total_num desc

Dataset Name	Description	Log Category
intf-sent-timeline-for-each-device	FortiGate cpu utilization timeline	event
select		
<pre>\$flex_timescale(timestamp) as devid,</pre>	hodex,	
<pre>cast(sum(total_cpu) / sum(count) a</pre>	s decimal(6, 0)	
) as cpu_ave, cast(
<pre>sum(total_mem) / sum(count) a) as mem ave,</pre>	s decimal(6, 0)	
<pre>cast(sum(total disk) / sum(count)</pre>	as decimal(6, 0)	
) as disk_ave, cast(
<pre>sum(sent) / sum(count) as dec) as sent kbps,</pre>	imal(10, 0)	
cast(im_{2} (10 0)	
<pre>sum(recv) / sum(count) as dec) as recv_kbps</pre>	inal(10, 0)	
from		

###({{FGT_DATASET_EVENT_DISK_LOGRATE_CPU_MEM}})### t where \$filter-drilldown group by hodex, devid order by hodex

Dataset Name	Description	Log Category
status-timeline-by-device-intf-sent	FortiGate interface summary view	event
<pre>select devid, cast(sum(sent) / sum(count) as deci) as sent_kbps, cast(sum(recv) / sum(count) as deci) as recv_kbps, cast(sum(sent + recv) / sum(count)) as transmit_kbps, max(transmit_peak) as transmit_ from ###({{FGT_DATASET_EVENT_DISK_LO peak desc</pre>	mal(10, 0) as decimal(10, 0) kbps_peak	devid order by transmit_kbps_

Dataset Name	Description	Log Category
intf-recv-timeline-for-each-device	FortiGate cpu utilization timeline	event
<pre>select \$flex_timescale(timestamp)</pre>	as hodex,	

```
devid,
```

```
cast(
   sum(total_cpu)/ sum(count) as decimal(6, 0)
 ) as cpu_ave,
 cast(
   sum(total_mem) / sum(count) as decimal(6, 0)
 ) as mem_ave,
 cast(
   sum(total disk) / sum(count) as decimal(6, 0)
 ) as disk_ave,
 cast(
   sum(sent) / sum(count) as decimal(10, 0)
 ) as sent kbps,
 cast(
   sum(recv) / sum(count) as decimal(10, 0)
 ) as recv kbps
from
```

###({{FGT_DATASET_EVENT_DISK_LOGRATE_CPU_MEM}})### t where \$filter-drilldown group by hodex, devid order by hodex

Dataset Name	Description	Log Category
status-timeline-by-device-intf-recv	FortiGate interface summary view	event
<pre>select devid, cast(sum(sent) / sum(count) as deci) as sent_kbps, cast(sum(recv) / sum(count) as deci) as recv_kbps, cast(sum(sent + recv) / sum(count)) as transmit_kbps, max(transmit_peak) as transmit_ from</pre>	mal(10, 0) as decimal(10, 0)	
###({{FGT_DATASET_EVENT_DISK_LC	GRATE_CPU_MEM}})### t group by d	levid order by transmit_kbps_

peak d	lesc
--------	------

Dataset Name	Description	Log Category
event-intf-summary-dev	FortiGate interface summary view	event
<pre>select devid, cast(sum(sent) / sum(count) as dec) as sent_kbps, cast(sum(recv) / sum(count) as dec) as recv_kbps, cast(sum(sent + recv) / sum(count)) as transmit_kbps, max(transmit_peak) as transmit_from</pre>	imal(10, 0) as decimal(10, 0)	

###({{FGT_DATASET_EVENT_DISK_LOGRATE_CPU_MEM}})### t group by devid order by transmit_kbps_
peak desc

Dataset Name	Description	Log Category
fgt-intf-stats-timeline-util-in-each	FortiGate Interface Statistics Timeline	event
select \$flex_timescale(tmstamp) as h (devname ':' intfname) cast(sum(bps_out)/ sum(interval)	as dev_intf,	
) as kbps_out_avg, cast(/ 1000 as decimal(10, 0)	
<pre>sum(bps_in)/ sum(interval)/) as kbps_in_avg, cast(</pre>	1000 as decimal(10, 0)	
<pre>sum(util_out)/ sum(interval) as util_out_avg,</pre>)/ 100 as decimal(10, 2)	
<pre>cast(sum(util_in)/ sum(interval)) as util_in_avg</pre>	/ 100 as decimal(10, 2)	
from		
(select \$flex_timestamp(timestamp) as tmstamp.	
dvid, intfname,		
<pre>sum(interval) as interval sum(sentbps * interval) a sum(rcvdbps * interval) a sum(sentutil * interval)</pre>	s bps_out, s bps_in,	
<pre>sum(rcvdutil * interval) from</pre>	—	
intfstats where		
\$cust_time_filter(timesta	mp)	
group by		
tmstamp, dvid,		
intfname		
) t1		
left join devtable t2 on t1.d	vid = t2.dvid	
where		
\$filter - drilldown group by		
hodex,		
dev_intf		
order by		
hodex		
Dataset Name	Description	Log Category
fgt-intf-stats-timeline-util-in	FortiGate Interface Received Utilization	event

```
select
  (devname || ':' || intfname) as dev_intf,
  cast(
    sum(bps out) / sum(interval) / 1000 as decimal(10, 0)
  ) as kbps out avg,
  cast(
   sum(bps in) / sum(interval) / 1000 as decimal(10, 0)
  ) as kbps in avg,
  cast(
    sum(util out) / sum(interval) / 100 as decimal(10, 2)
  ) as util out avg,
 cast(
   sum(util in) / sum(interval) / 100 as decimal(10, 2)
  ) as util in avg
from
  (
   select
      $flex_timestamp(timestamp) as tmstamp,
     tbl intf.dvid,
     intfname,
      sum(interval) as interval,
      sum(sentbps * interval) as bps out,
      sum(rcvdbps * interval) as bps in,
      sum(sentutil * interval) as util out,
      sum(rcvdutil * interval) as util_in
    from
      (
        select
          distinct dvid
        from
          ### (select dvid from $log-event where $filter and action='perf-stats' group by
```

dvid)### t) tbl_log inner join intfstats tbl_intf on tbl_log.dvid = tbl_intf.dvid where \$cust_ time_filter(timestamp) group by tmstamp, tbl_intf.dvid, intfname) t1 left join devtable t2 on t1.dvid = t2.dvid group by dev_intf order by util_in_avg desc, kbps_in_avg desc, kbps_out_avg desc

Dataset Name	Description	Log Category
fgt-intf-stats-timeline-util-out-each	FortiGate Interface Statistics Timeline	event
<pre>select \$flex_timescale(tmstamp) as hode (devname ':' intfname) as cast(sum(bps_out) / sum(interval) / 1) as kbps_out_avg, cast(sum(bps_in) / sum(interval) / 1) as kbps_in_avg, cast(sum(util_out) / sum(interval) /) as util_out_avg, cast(sum(util_in) / sum(interval) / 1) as util_in_avg from</pre>	<pre>dev_intf, 1000 as decimal(10, 0) 000 as decimal(10, 0) 100 as decimal(10, 2)</pre>	

```
(
   select
     $flex_timestamp(timestamp) as tmstamp,
     dvid,
     intfname,
     sum(interval) as interval,
     sum(sentbps * interval) as bps out,
     sum(rcvdbps * interval) as bps in,
     sum(sentutil * interval) as util out,
     sum(rcvdutil * interval) as util_in
   from
     intfstats
   where
     $cust_time_filter(timestamp)
    group by
     tmstamp,
     dvid,
     intfname
 ) t1
  left join devtable t2 on t1.dvid = t2.dvid
where
 $filter - drilldown
group by
 hodex,
 dev_intf
order by
 hodex
```

Dataset Name	Description	Log Category
fgt-intf-stats-timeline-util-out	FortiGate Interface Sent Utilization	event
<pre>select (devname ':' intfname) cast(sum(bps_out) / sum(interval) as kbps_out_avg, cast(sum(bps_in) / sum(interval)) as kbps_in_avg, cast(sum(util_out) / sum(interval) as util_out_avg, cast(sum(util_in) / sum(interval) as util_in_avg from (select \$flex_timestamp(timestam tbl_intf.dvid, intfname, sum(interval) as interval sum(sentbps * interval) sum(sentutil * interval)</pre>	<pre></pre>	

```
sum(rcvdutil * interval) as util_in
from
(
    select
    distinct dvid
    from
```

###(select dvid from \$log-event where \$filter and action='perf-stats' group by dvid)### t) tbl_log inner join intfstats tbl_intf on tbl_log.dvid = tbl_intf.dvid where \$cust_ time_filter(timestamp) group by tmstamp, tbl_intf.dvid, intfname) t1 left join devtable t2 on t1.dvid = t2.dvid group by dev_intf order by util_out_avg desc, kbps_out_avg desc, kbps_in_avg desc

Dataset Name	Description	Log Category
fgt-intf-stats-timeline-bit-rate-in-each	FortiGate Interface Statistics Timeline	event
<pre>select \$flex_timescale(tmstamp) as hod (devname ':' intfname) as cast(sum(bps_out) / sum(interval) /) as kbps_out_avg, cast(sum(bps_in) / sum(interval) / 1) as kbps_in_avg, cast(sum(util_out) / sum(interval) /) as util_out_avg, cast(sum(util_in) / sum(interval) /) as util_in_avg</pre>	<pre>dev_intf, 1000 as decimal(10, 0) 000 as decimal(10, 0) 100 as decimal(10, 2)</pre>	
) as util_in_avg from (
<pre>select \$flex_timestamp(timestamp) dvid, intfname, sum(interval) as interval, sum(sentbps * interval) as sum(rcvdbps * interval) as sum(sentutil * interval) as sum(rcvdutil * interval) as </pre>	<pre>bps_out, bps_in, util_out,</pre>	
from intfstats		
where \$cust_time_filter(timestamp group by tmstamp, dvid, intfname)	
) t1 left join devtable t2 on t1.dvi	d = t2.dvid	
where \$filter - drilldown		
group by hodex,		

dev_intf order by hodex

Dataset Name Description Log Category fgt-intf-stats-timeline-bit-rate-in FortiGate Interface Received Bit Rate event select (devname || ':' || intfname) as dev_intf, cast(sum(bps out) / sum(interval) / 1000 as decimal(10, 0)) as kbps out avg, cast(sum(bps_in) / sum(interval) / 1000 as decimal(10, 0)) as kbps_in_avg, cast(sum(util_out) / sum(interval) / 100 as decimal(10, 2)) as util_out_avg, cast(sum(util in) / sum(interval) / 100 as decimal(10, 2)) as util in avg from (select \$flex timestamp(timestamp) as tmstamp, tbl intf.dvid, intfname, sum(interval) as interval, sum(sentbps * interval) as bps out, sum(rcvdbps * interval) as bps in, sum(sentutil * interval) as util out, sum(rcvdutil * interval) as util_in from (select distinct dvid from ###(select dvid from \$log-event where \$filter and action='perf-stats' group by dvid)### t) tbl_log inner join intfstats tbl_intf on tbl_log.dvid = tbl_intf.dvid where \$cust_ time_filter(timestamp) group by tmstamp, tbl_intf.dvid, intfname) t1 left join devtable t2 on t1.dvid = t2.dvid group by dev_intf order by kbps_in_avg desc

Dataset Name	Description	Log Category
fgt-intf-stats-timeline-bit-rate-out-each	FortiGate Interface Statistics Timeline	event
<pre>select \$flex_timescale(tmstamp) as hode (devname ':' intfname) as cast(sum(bps_out) / sum(interval) / 1) as kbps_out_avg, cast(sum(bps_in) / sum(interval) / 10) as kbps_in_avg,</pre>	dev_intf, .000 as decimal(10, 0)	

```
cast(
    sum(util_out) / sum(interval) / 100 as decimal(10, 2)
  ) as util_out_avg,
  cast(
   sum(util_in) / sum(interval) / 100 as decimal(10, 2)
  ) as util in avg
from
  (
    select
     $flex_timestamp(timestamp) as tmstamp,
     dvid,
     intfname,
     sum(interval) as interval,
      sum(sentbps * interval) as bps_out,
      sum(rcvdbps * interval) as bps in,
      sum(sentutil * interval) as util out,
      sum(rcvdutil * interval) as util_in
    from
     intfstats
    where
      $cust_time_filter(timestamp)
    group by
     tmstamp,
     dvid,
     intfname
  ) t1
  left join devtable t2 on t1.dvid = t2.dvid
where
  $filter - drilldown
group by
 hodex,
 dev intf
order by
  hodex
```

Dataset Name	Description	Log Category
fgt-intf-stats-timeline-bit-rate-out	FortiGate Interface Sent Bit Rate	event
select		
(devname ':' intfname)	as dev_intf,	
cast(
<pre>sum(bps_out)/ sum(interval)</pre>	/ 1000 as decimal(10, 0)	
) as kbps_out_avg,		
cast(
<pre>sum(bps_in)/ sum(interval),</pre>	/ 1000 as decimal(10, 0)	
) as kbps_in_avg,		
cast(
<pre>sum(util_out)/ sum(interval</pre>	L)/ 100 as decimal(10, 2)	
) as util_out_avg,		
cast(
<pre>sum(util_in) / sum(interval)</pre>	/ 100 as decimal(10, 2)	
) as util_in_avg		
from		
(
select		

```
$flex_timestamp(timestamp) as tmstamp,
tbl_intf.dvid,
intfname,
sum(interval) as interval,
sum(sentbps * interval) as bps_out,
sum(rcvdbps * interval) as bps_in,
sum(sentutil * interval) as util_out,
sum(rcvdutil * interval) as util_in
from
  (
    select
    distinct dvid
    from
    ###(select dvid from $log-event wh
```

###(select dvid from \$log-event where \$filter and action='perf-stats' group by dvid)### t) tbl_log inner join intfstats tbl_intf on tbl_log.dvid = tbl_intf.dvid where \$cust_ time_filter(timestamp) group by tmstamp, tbl_intf.dvid, intfname) t1 left join devtable t2 on t1.dvid = t2.dvid group by dev_intf order by kbps_out_avg desc

Dataset Name	Description	Log Category
fgt-intf-stats-summary-view	FortiGate Interface Received Utilization	event
<pre>) as kbps_out_avg, cast(sum(bps_in) / sum(interva) as kbps_in_avg, cast(sum(util_out) / sum(inter) as util_out_avg, cast(sum(util_in) / sum(interv</pre>	e) as dev_intf, al)/ 1000 as decimal(10, 0) 1)/ 1000 as decimal(10, 0) val)/ 100 as decimal(10, 2) al)/ 100 as decimal(10, 2)	
<pre>) as util_in_avg from (select \$flex_timestamp(timest thl_itt6_deid)</pre>	amp) as tmstamp,	
<pre>tbl_intf.dvid, intfname, sum(interval) as inter sum(sentbps * interval sum(rcvdbps * interval sum(sentutil * interva sum(rcvdutil * interva</pre>) as bps_out,) as bps_in, 1) as util_out,	
<pre>from (select distinct dvid from ###(select dvid fr dvid)### t) tbl_log inner jo</pre>	om \$log-event where \$filter and action='perf- in intfstats tbl_intf on tbl_log.dvid = tbl_in by tmstamp, tbl_intf.dvid, intfname) t1 left	ntf.dvid where \$cust_

t1.dvid = t2.dvid group by dev_intf order by util_in_avg desc, kbps_in_avg desc, kbps_out_avg
desc

Dataset Name	Description	Log Category
fgt-ha-failure-timeline	FortiGate HA Failure Timeline	event

select

```
$flex_timescale(timestamp) as hodex,
count(*) as total_num
```

from

###(select \$flex_timestamp as timestamp, dtime, devid, coalesce(nullifna(logdesc), msg) as msg_desc from \$log where \$filter and subtype='ha' and logid_to_int(logid) in (35011, 35012, 35013, 37892, 37893, 37897, 37898, 37901, 37902, 37907, 37908) order by dtime desc)### t group by hodex order by hodex

Dataset Name	Description	Log Category
fgt-ha-failure-summary	FortiGate HA Failure Summary	event
select		

```
from_dtime(dtime) as time_s,
    devid,
```

```
msg_desc
```

```
from
```

###(select \$flex_timestamp as timestamp, dtime, devid, coalesce(nullifna(logdesc), msg) as
msg_desc from \$log where \$filter and subtype='ha' and logid_to_int(logid) in (35011, 35012,
35013, 37892, 37893, 37897, 37898, 37901, 37902, 37907, 37908) order by dtime desc)### t order
by time_s desc

Dataset Name	Description	Log Category
fgt-env-faults-power	FortiGate Power Supply Faults	event
<pre>select from_dtime(dtime) as ti devid, coalesce(nullifna(logdesc), msg) as msg_desc from \$log where \$filter and logid_to_int(logid) order by time_s desc</pre>		
Dataset Name	Description	Log Category

Dataset Name	Description	Log Category
fgt-env-faults-fan	FortiGate Fan Faults	event
select from dtime(dtime) as time s,		
devid,		

```
coalesce(
    nullifna(logdesc),
    msg
) as msg_desc
from
    $log
where
    $filter
    and logid_to_int(logid)= 22108
order by
    time s desc
```

Dataset Name	Description	Log Category
fgt-env-faults-temperature	FortiGate Temperatre Too High	event
<pre>select from_dtime(dtime) as time_s, devid, coalesce(</pre>		
<pre>nullifna(logdesc), msg) as msg_desc</pre>		
from \$log where		
\$filter and logid_to_int(logid)= 22109		
order by time_s desc		
Dataset Name	Description	Log Category

```
Behaviour-Banned-Application
                                   Bullying Chat Search and Message Logging
                                                                                    app-ctrl
select
 filename,
  string agg(distinct app, ' ') as app agg,
  string agg(
   distinct from itime(itime): :text,
    1 1
  ) as itime agg,
  string_agg(distinct user_src, ' ') as user_agg,
  string agg(distinct `group`, ' ') as group agg,
  string agg(
   distinct ipstr(`srcip`),
    1 1
  ) as srcip_agg,
  count(*) as requests
from
  ###(select filename, app, itime, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr
(`srcip`)) as user_src, `group`, `srcip` from $log where $filter and (lower(app) in ('face-
book_post', 'facebook_chat', 'twitter_post', 'youtube_video.access', 'gmail_chat', 'gmail_
send.message', 'linkedin_post', 'vimeo_video.access', 'google.search_search.phrase',
'bing.search_search.phrase')) and ($bully_keywords) order by itime desc)### t group by file-
name order by requests desc
```

Dataset Name	Description	Log Category
Behaviour-Banned-User	Bullying Chat Search and Message Logging	app-ctrl
<pre>select filename, string_agg(distinct app, ' string_agg(distinct from_itime(itime) ' ') as itime_agg, string_agg(distinct user_src string_agg(distinct `group`, string_agg(distinct ipstr(`srcip`), ' '</pre>): :text, c, ' ') as user_agg,	
(`srcip`)) as user_src, `group book_post', 'facebook_chat', send.message', 'linkedin_post	time, coalesce(nullifna(`user`), nullifna(`u p`, `srcip` from \$log where \$filter and (low 'twitter_post', 'youtube_video.access', 'gma ', 'vimeo_video.access', 'google.search_sear and (\$bully_keywords) order by itime desc)#	wer(app) in ('face- ail_chat', 'gmail_ cch.phrase',

Dataset Name	Description	Log Category
Behaviour-Banned-User-Drilldown	Bullying Chat Search and Message Logging	app-ctrl
<pre>select filename, string_agg(distinct app, ' ') string_agg(distinct from_itime(itime): ' ') as itime_agg, string_agg(distinct user_src, string_agg(distinct `group`, string_agg(distinct ipstr(`srcip`), ' ') as srcip_agg, count(*) as requests from ###(select filename, app, iti (`srcip`)) as user_src, `group` book_post', 'facebook_chat', 't send.message', 'linkedin_post',</pre>	<pre>as app_agg, :text, ' ') as user_agg, ' ') as group_agg, ' ') as group_agg, me, coalesce(nullifna(`user`), nullifna(`uu , `srcip` from \$log where \$filter and (lowe witter_post', 'youtube_video.access', 'gma: 'vimeo_video.access', 'google.search_search</pre>	<pre>hauthuser`), ipstr er(app) in ('face- il_chat', 'gmail_ ch.phrase',</pre>
'bing.search_search.phrase')) a name order by requests desc	nd (\$bully_keywords) order by itime desc)#:	## t group by file-
Dataset Name	Description	Log Category

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

```
select
  filename,
  string_agg(distinct app, ' ') as app_agg,
  string agg(
    distinct from itime(itime): :text,
    1 1
  ) as itime agg,
  string agg(distinct user src, ' ') as user agg,
  string agg(distinct `group`, ' ') as group agg,
  string agg(
   distinct ipstr(`srcip`),
    1 1
  ) as srcip agg,
  count(*) as requests
from
  ### (select filename, app, itime, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr
(`srcip`)) as user_src, `group`, `srcip` from $log where $filter and (lower(app) in ('face-
book_post', 'facebook_chat', 'twitter_post', 'youtube_video.access', 'gmail_chat', 'gmail_
```

book_post', 'facebook_chat', 'twitter_post', 'youtube_video.access', 'gmail_chat', 'gmail_ send.message', 'linkedin_post', 'vimeo_video.access', 'google.search_search.phrase', 'bing.search_search.phrase')) and (\$bully_keywords) order by itime desc)### t group by filename order by requests desc

Dataset Name	Description	Log Category
Self-Harm-Behaviour-Banned-User-Pie	Self-Harm Chat Search and Message Logging	app-ctrl

```
select
```

```
filename,
  string agg(distinct app, ' ') as app agg,
  string agg(
   distinct from itime(itime): :text,
    1 1
  ) as itime agg,
  string_agg(distinct user_src, ' ') as user agg,
  string agg(distinct `group`, ' ') as group agg,
  string agg(
   distinct ipstr(`srcip`),
    1 1
 ) as srcip_agg,
  count(*) as requests
from
  ###(select filename, app, itime, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr
(`srcip`)) as user src, `group`, `srcip` from $log where $filter and (lower(app) in ('face-
book post', 'facebook chat', 'twitter post', 'youtube video.access', 'gmail chat', 'gmail
send.message', 'linkedin post', 'vimeo video.access', 'google.search search.phrase',
'bing.search search.phrase')) and ($banned keywords) order by itime desc)### t group by file-
name order by requests desc
```

Dataset Name	Description	Log Category
Self-Harm-Behaviour-Banned- Application-Pie	Self-Harm Chat Search and Message Logging	app-ctrl
<pre>select filename, string_agg(distinct app, ' '</pre>) as app_agg,	

```
string_agg(
    distinct from_itime(itime): :text,
    ' '
) as itime_agg,
    string_agg(distinct user_src, ' ') as user_agg,
    string_agg(distinct `group`, ' ') as group_agg,
    string_agg(
        distinct ipstr(`srcip`),
        ' '
) as srcip_agg,
        count(*) as requests
from
```

###(select filename, app, itime, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr
(`srcip`)) as user_src, `group`, `srcip` from \$log where \$filter and (lower(app) in ('facebook_post', 'facebook_chat', 'twitter_post', 'youtube_video.access', 'gmail_chat', 'gmail_
send.message', 'linkedin_post', 'vimeo_video.access', 'google.search_search.phrase',
'bing.search_search.phrase')) and (\$banned_keywords) order by itime desc)### t group by filename order by requests desc

```
Dataset NameDescriptionLog CategorySelf-Harm-Behaviour-Banned-User-BarSelf-Harm Chat Search and Message Loggingapp-ctrl
```

select filename, string_agg(distinct app, ' ') as app_agg, string_agg(distinct from_itime(itime): :text, ' ') as itime_agg, string_agg(distinct user_src, ' ') as user_agg, string_agg(distinct `group`, ' ') as group_agg, string_agg(distinct ipstr(`srcip`), ' ') as srcip_agg, count(*) as requests

from

###(select filename, app, itime, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr
(`srcip`)) as user_src, `group`, `srcip` from \$log where \$filter and (lower(app) in ('facebook_post', 'facebook_chat', 'twitter_post', 'youtube_video.access', 'gmail_chat', 'gmail_
send.message', 'linkedin_post', 'vimeo_video.access', 'google.search_search.phrase',
'bing.search_search.phrase')) and (\$banned_keywords) order by itime desc)### t group by filename order by requests desc

Dataset Name	Description	Log Category
Self-Harm-Behaviour-Banned-User- Drilldown	Self-Harm Chat Search and Message Logging	app-ctrl
<pre>select filename, string_agg(distinct app, ' ') a string_agg(distinct from_itime(itime): : ' '</pre>	_	

```
) as itime_agg,
string_agg(distinct user_src, ' ') as user_agg,
string_agg(distinct `group`, ' ') as group_agg,
string_agg(
    distinct ipstr(`srcip`),
    ' '
) as srcip_agg,
count(*) as requests
from
    ###(select filename, app, itime, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr
(`srcip`)) as user_src, `group`, `srcip` from $log where $filter and (lower(app) in ('face-
book_post', 'facebook_chat', 'twitter_post', 'youtube_video.access', 'gmail_chat', 'gmail_
send.message', 'linkedin_post', 'vimeo_video.access', 'google.search_search.phrase',
'bing.search_search.phrase')) and ($banned_keywords) order by itime desc)### t group by file-
name order by requests desc
```

Dataset Name	Description	Log Category
Self-Harm-behaviour-banned	Self-Harm Chat Search and Message Logging	app-ctrl

```
select
filename,
string_agg(distinct app, ' ') as app_agg,
string_agg(
    distinct from_itime(itime): :text,
    ' '
) as itime_agg,
string_agg(distinct user_src, ' ') as user_agg,
string_agg(distinct `group`, ' ') as group_agg,
string_agg(
    distinct ipstr(`srcip`),
    ' '
) as srcip_agg,
    count(*) as requests
from
```

###(select filename, app, itime, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr
(`srcip`)) as user_src, `group`, `srcip` from \$log where \$filter and (lower(app) in ('facebook_post', 'facebook_chat', 'twitter_post', 'youtube_video.access', 'gmail_chat', 'gmail_
send.message', 'linkedin_post', 'vimeo_video.access', 'google.search_search.phrase',
'bing.search_search.phrase')) and (\$banned_keywords) order by itime desc)### t group by filename order by requests desc

Dataset Name	Description	Log Category
Browsing-Time-per-Social-Media	Browsing Time vs. Domain	traffic
<pre>select domain, ebtr_value(ebtr_agg_flat(browsetime), null, \$timespan) as browsetime</pre>		
<pre>from ###({{FGT_DATASET_TRAFFIC_TOP_ group by domain order by browset</pre>	DOMAIN_USER_BY_EB_TIME}})### t where desc	e browsetime is not null

Dataset Name	Description	Log Category
Social-Networking-Bar-Graph	Social Networking Browsing Time	traffic

select

f_user,

sum(bandwidth) as bandwidth

from

###({{FGT_DATASET_TRAFFIC_TOP_DOMAIN_USER_BY_EB_TIME}})### t where bandwidth>0 group by f_
user order by bandwidth desc

Dataset Name	Description	Log Category
Top-Social-Networking-Durations- Sources-Drilldown	Top Social Networking Durations from Sources Drilldown	traffic
select		

```
f_user,
ebtr_value(
   ebtr_agg_flat(browsetime),
   null,
   $timespan
) as browsetime
```

from

###({{FGT_DATASET_TRAFFIC_TOP_DOMAIN_USER_BY_EB_TIME}})### t where \$filter-drilldown and browsetime is not null group by f_user order by browsetime desc

Dataset Name	Description	Log Category
Top-Social-Networking-Durations- Domains-Drilldown	Browsing Time vs. Domain	traffic
<pre>select domain, ebtr_value(ebtr_agg_flat(browsetime), null, \$timespan) as browsetime from ###({{FGT_DATASET_TRAFFIC_TOP group by domain order by browse</pre>	_DOMAIN_USER_BY_EB_TIME}})### t where time desc	e browsetime is not null
Dataset Name	Description	Log Category

	app-ctrl
Facebook-PostsFacebook Postsa	••
<pre>select from_itime(itime) as i_time, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)</pre>	

) as f_user, srcip,

```
filename
from
 $log
where
 $filter
 and lower(app) = lower('Facebook_Post')
 and filename is not null
order by
 i_time desc
```

Dataset Name	Description	Log Category
Facebook-Chats	Facebook Chats	app-ctrl
<pre>select filename, string_agg(distinct from_itime(itime ' ') as itime_agg, string_agg(distinct user_sr string_agg(distinct `group` string_agg(distinct ipstr(srcip), ' ') as srcip_agg, count(*) as requests</pre>	c, ' ') as user_agg,	
		ullifna(`unauthuser`), ipstr(`sr- and lower(app)=lower('Facebook_

Chat') and filename is not null) ### t group by filename order by requests desc

Dataset Name	Description	Log Category
Twitter-Posts	Twitter Posts	app-ctrl
<pre>select from_itime(itime) as i_ coalesce(nullifna(`user`), nullifna(`unauthuser` ipstr(`srcip`)) as f_user, srcip, filename from \$log where \$filter and lower(app)= lower(' and filename is not nul order by i_time desc</pre>), Twitter_Post')	
Dataset Name	Description	Log Category

LinkedIn-Posts-and-Comments	
-----------------------------	--

LinkedIn Posts and Comments

app-ctrl

```
select
 filename,
  string_agg(
   distinct from itime(itime): :text,
    1 1
  ) as itime agg,
  string agg(distinct user src, ' ') as user agg,
  string agg(distinct `group`, ' ') as group agg,
  string agg(
   distinct ipstr(srcip),
    1 1
  ) as srcip_agg,
  count(*) as requests
from
  ###(select filename, itime, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`sr-
cip`)) as user src, `group`, srcip from $log where $filter and lower(app)=lower('LinkedIn
```

Post') and filename is not null)### t group by filename order by requests desc

Dataset Name	Description	Log Category
sdwan-fw-Device-Interface-Quality_	SD-WAN Device-Interface Statistic	event
Bibandwidth-drilldown		

select

devid,

sum(bibandwidth) / sum(count_linkup) as bibandwidth

from

###({{FGT_DATASET_EVENT_SD_WAN_DEV_STAT}})### t where \$filter-drilldown and bibandwidth is not null group by devid having sum(count_linkup)>0 order by bibandwidth desc

Dataset Name	Description	Log Category
sdwan-Device-Interface-Latency-Line	SD-WAN Device-Interface Latency Timeline	event
<pre>select \$flex_timescale(timestamp) as h t1.interface, min(latency) as latency</pre>	nodex,	
from		
(
select		
timestamp,		
devid,		
interface,		
<pre>sum(latency)/ sum(count_linkup) as latency</pre>		
from		
###({{FGT_DATASET_EVENT_SD_	_WAN_DEV_STAT}})### t group by timestamp, de	vid, interface hav-
<pre>ing sum(count_linkup)>0) t1 inner join (select interface, count(*) as num_intf from ###({{FGT_</pre>		
DATASET_EVENT_SD_WAN_DEV_STAT}}) ### t where \$filter-drilldown and interface is not null group		
by interface order by num_intf desc limit 10)t2 on t1.interface=t2.interface group by hodex,		
t1.interface order by hodex		

Dataset Name	Description	Log Category
sdwan-Device-Interface-Jitter-Line	SD-WAN Device-Interface Jitter Timeline	event

```
select
  $flex_timescale(timestamp) as hodex,
  t1.interface,
  min(jitter) as jitter
from
  (
    select
    timestamp,
    devid,
    interface,
    sum(jitter) / sum(count_linkup) as jitter
  from
    ###({{FGT DATASET EVENT SD WAN DEV STAT}}
```

###({{FGT_DATASET_EVENT_SD_WAN_DEV_STAT}))### t group by timestamp, devid, interface having sum(count_linkup)>0) t1 inner join (select interface, count(*) as num_intf from ###({{FGT_ DATASET_EVENT_SD_WAN_DEV_STAT}))### t where \$filter-drilldown and interface is not null group by interface order by num_intf desc limit 10)t2 on t1.interface=t2.interface group by hodex, t1.interface order by hodex

Dataset Name	Description	Log Category
sdwan-Device-Interface-Packetloss- Line	SD-WAN Device-Interface Packetloss Timeline	event
<pre>select \$flex_timescale(timestamp) as t1.interface, min(packetloss) as packetloss</pre>	hodex,	
from		
select		
timestamp,		
devid,		
interface,		
<pre>sum(packetloss)/ sum(count</pre>	_linkup) as packetloss	
from		
	_WAN_DEV_STAT}})### t group by timestamp, de	
	r join (select interface, count(*) as num_ir	_
DATASET EVENT SD WAN DEV STAT } })	### t where \$filter-drilldown and interface	is not null group

ing sum(count_linkup)>0) t1 inner join (select interface, count(*) as num_intf from ###({{FGT_ DATASET_EVENT_SD_WAN_DEV_STAT}))### t where \$filter-drilldown and interface is not null group by interface order by num_intf desc limit 10)t2 on t1.interface=t2.interface group by hodex, t1.interface order by hodex

Dataset Name	Description	Log Category
sdwan-Device-Latency-Line	SD-WAN Device Latency Timeline	event
<pre>select \$flex_timescale(timestamp) devid, min(latency) as latency from (select timestamp, devid, interface, sum(latency) / sum(count)</pre>		

from

###({{FGT_DATASET_EVENT_SD_WAN_DEV_STAT}})### t where \$filter-drilldown and latency is not null group by timestamp, devid, interface having sum(count_linkup)>0) t1 group by hodex, devid order by hodex

Dataset Name	Description	Log Category
sdwan-Device-Jitter-Line	SD-WAN Device Jitter Timeline	event
<pre>select \$flex_timescale(timestamp) a devid, min(jitter) as jitter from</pre>	s hodex,	
(
select timestamp,		
devid,		
interface,		
<pre>sum(jitter)/ sum(count_l from</pre>	inkup) as jitter	
###({{FGT_DATASET_EVENT_	SD_WAN_DEV_STAT}})### t where \$filter-	_
not null group by timestamp, d	levid, interface having sum(count_linku	up)>0) t1 group by hodex,

devid order by hodex

Dataset Name	Description	Log Category
sdwan-Device-Packetloss-Line	SD-WAN Device Packet Loss Timeline	event
from ###({{FGT_DATASET_EVENT_		
Dataset Name	Description	Log Category

```
sdwan-Device-Interface-Summary-by- SD-WAN Device Interface Summary by Bibandwidth event
Bibandwidth
```

```
select
  devid,
  interface,
  sum(bibandwidth) / sum(count_linkup) as bibandwidth,
  cast(
     min(latency_min) as decimal(18, 2)
```

```
) as latency_min,
 cast(
   sum(latency) / sum(count_linkup) as decimal(18, 2)
 ) as latency_avg,
 cast(
  max(latency max) as decimal(18, 2)
 ) as latency max,
 cast(
  min(jitter min) as decimal(18, 2)
 ) as jitter min,
 cast(
   sum(jitter)/ sum(count linkup) as decimal(18, 2)
 ) as jitter avg,
 cast(
  max(jitter max) as decimal(18, 2)
 ) as jitter max,
 cast(
  min(packetloss_min) as decimal(18, 2)
 ) as packetloss min,
 cast(
   sum(packetloss) / sum(count_linkup) as decimal(18, 2)
 ) as packetloss_avg,
 cast(
   max(packetloss_max) as decimal(18, 2)
 ) as packetloss_max
from
```

###({{FGT_DATASET_EVENT_SD_WAN_DEV_STAT}})### t where \$filter-drilldown and interface is not null group by devid, interface having sum(count_linkup)>0 order by devid, interface

Dataset Name	Description	Log Category
sdwan-Top-App-By-Bandwidth	Top SD-WAN application by bandwidth	traffic
<pre>select appid, app_group, sum(bandwidth) as bandwidth, sum(sessions) as sessions from ###({{FGT_DATASET_TRAFFIC_SD_Y is not null group by appid, app</pre>	WAN_RULES_INTF_USR}})### t where \$filter-dr: _group order by bandwidth desc	illdown and rulename
Dataset Name	Description	Log Category
sdwan-Top-App-By-Bandwidth-Sankey	y Top SD-WAN application by bandwidth usage	traffic
select 'SD-WAN Utilization' as summa		
<pre>app_group, devid, dstintf as interface, sum(bandwidth) as bandwidth from</pre>	ry,	

###({{FGT_DATASET_TRAFFIC_SD_WAN_RULES_INTF_USR}})### t where \$filter-drilldown and rulename
is not null group by app_group, devid, interface order by bandwidth desc

Dataset Name	Description	Log Category
sdwan-Device-Interface-bandwidth- Drilldown	SD-WAN Device Statistic by Bibandwidth	event
Dillidowit		

select

devid,

sum(bibandwidth) / sum(count_linkup) as bibandwidth

from

###({{FGT_DATASET_EVENT_SD_WAN_DEV_STAT}})### t where \$filter-drilldown and bibandwidth is not null group by devid having sum(count_linkup)>0 order by bibandwidth desc

Dataset Name	Description	Log Category
sdwan-Device-Rules-Donut-Bandwidth	Top SD-WAN Links bandwidth	traffic

select

rulename,

sum(bandwidth) as bandwidth

from

###({{FGT_DATASET_TRAFFIC_SD_WAN_RULES_INTF_USR}})### t where \$filter-drilldown and rulename
is not null group by rulename order by bandwidth desc limit 10

Dataset Name	Description	Log Category
sdwan-device-interface-bandwidth	Top SD-WAN Links bandwidth	traffic
<pre>select interface, sum(bandwidth) as bandwidth from (((</pre>		
	dth C_SD_WAN_RULES_INTF_USR}})### t where	

\$filter-drilldown and rulename is not null group by interface) union all (select dstintf as interface, sum(bandwidth) as bandwidth from ###({{FGT_DATASET_TRAFFIC_SD_WAN_RULES_INTF_ USR}})### t where dstintfrole='wan' and \$filter-drilldown and rulename is not null group by interface)) t group by interface order by bandwidth desc limit 10

Dataset Name	Description	Log Category
sdwan-Top-Application-Session- Bandwidth	Top SD-WAN application by bandwidth	traffic
<pre>select appid, app_group, sum(bandwidth) as bandwidth, sum(sessions) as sessions from</pre>		

###({{FGT_DATASET_TRAFFIC_SD_WAN_RULES_INTF_USR}})### t where \$filter-drilldown and rulename
is not null group by appid, app_group order by bandwidth desc

Dataset Name	Description	Log Category
sdwan-Top-Users-By-Bandwidth-Bar	SD-WAN Top users by bandwidth usage	traffic

select

```
user_src,
```

sum(bandwidth) as bandwidth

```
from
```

###({{FGT_DATASET_TRAFFIC_SD_WAN_RULES_INTF_USR}})### t where \$filter-drilldown and rulename
is not null group by user_src order by bandwidth desc

Dataset Name	Description	Log Category
sdwan-top-user-app-Drilldown	SD-WAN Top users and Application by bandwidth	traffic

select

user_src, app_group, sum(bandwidth) as bandwidth

from

###({{FGT_DATASET_TRAFFIC_SD_WAN_RULES_INTF_USR}})### t where \$filter-drilldown and rulename
is not null group by user_src, app_group order by bandwidth desc

Dataset Name	Description	Log Category
sdwan-Device-Intfe-traffic-out-	SD-WAN Device-Interface traffic sent bandwidth	traffic
bandwidth-Line	Timeline	

select

\$flex_timescale(timestamp) as hodex, t1.dstintf as interface, sum(traffic_out) as bandwidth

from

###({{FGT_DATASET_TRAFFIC_SD_WAN_RULES_INTF_USR})### t1 inner join (select dstintf, count
(*) as num_intf from ###({{FGT_DATASET_TRAFFIC_SD_WAN_RULES_INTF_USR})### t where \$filterdrilldown and dstintf is not null and dstintfrole ='wan' and rulename is not null group by
dstintf order by num_intf desc limit 10)t2 on t1.dstintf=t2.dstintf where rulename is not null
group by hodex, t1.dstintf order by hodex

Dataset Name	Description	Log Category
sdwan-Device-Intfe-traffic-in- bandwidth-Line	SD-WAN Device-Interface traffic received bandwidth Timeline	traffic
(*) as num_intf from ###({{FGT_	WAN_RULES_INTF_USR})### t1 inner join (select s DATASET_TRAFFIC_SD_WAN_RULES_INTF_USR})### t wh lll and srcintfrole ='wan' and rulename is not nu	ere \$filter-

srcintf order by num_intf desc limit 10)t2 on t1.srcintf=t2.srcintf where rulename is not null
group by hodex, t1.srcintf order by hodex

Dataset Name	Description	Log Category
sdwan-Device-Intfe-traffic-bandwidth- Line	SD-WAN Device-Interface traffic sent bandwidth Timeline	traffic

select

\$flex_timescale(timestamp) as hodex, t1.dstintf as interface, sum(traffic_out) as bandwidth

from

###({{FGT_DATASET_TRAFFIC_SD_WAN_RULES_INTF_USR}))### t1 inner join (select dstintf, count
(*) as num_intf from ###({{FGT_DATASET_TRAFFIC_SD_WAN_RULES_INTF_USR}))### t where \$filterdrilldown and dstintf is not null and dstintfrole ='wan' and rulename is not null group by
dstintf order by num_intf desc limit 10)t2 on t1.dstintf=t2.dstintf where rulename is not null
group by hodex, t1.dstintf order by hodex

Dataset Name	Description	Log Category
sdwan-Device-SLA-Interface-	SD-WAN Device Statistic by Bibandwidth	event
bandwidth-Drilldown		

select

devid,

```
sum(bibandwidth) / sum(count_linkup) as bibandwidth
from
```

###({{FGT_DATASET_EVENT_SD_WAN_DEV_STAT}})### t where \$filter-drilldown and bibandwidth is not null group by devid having sum(count_linkup)>0 order by bibandwidth desc

Dataset Name	Description	Log Category
sdwan-Device-SLA-Rule-Latency-Line	SD-WAN Device-SLA-Rule Latency Line	event

```
select
$flex_timescale(timestamp) as hodex,
t1.sla_rule,
min(latency) as latency
from
(
    select
    timestamp,
    devid,
    sla_rule,
    sum(latency) / sum(count_linkup) as latency
from
    ###(//FGT_DATASET_EVENT_SD_WAN_DEV_STATL));
```

###({{FGT_DATASET_EVENT_SD_WAN_DEV_STAT}))### t group by timestamp, devid, sla_rule having sum(count_linkup)>0) t1 inner join (select sla_rule, count(*) as num_intf from ###({{FGT_ DATASET_EVENT_SD_WAN_DEV_STAT}))### t where \$filter-drilldown and sla_rule is not null group by sla_rule order by num_intf desc limit 10)t2 on t1.sla_rule=t2.sla_rule group by hodex, t1.sla_rule order by hodex

Dataset Name	Description	Log Category
sdwan-Device-SLA-Rule-Jitter-Line	SD-WAN Device-SLA-Rule Jitter Line	event

```
select
  $flex_timescale(timestamp) as hodex,
  t1.sla_rule,
  min(jitter) as jitter
from
  (
    select
    timestamp,
    devid,
    sla_rule,
    sum(jitter)/ sum(count_linkup) as jitter
  from
    ###({{FGT_DATASET_EVENT_SD_WAN_DEV_STAT}))### t group by timestamp, devid, sla_rule hav-
```

ing sum(count_linkup)>0) t1 inner join (select sla_rule, count(*) as num_intf from ###({{FGT_ DATASET_EVENT_SD_WAN_DEV_STAT}))### t where \$filter-drilldown and sla_rule is not null group by sla_rule order by num_intf desc limit 10)t2 on t1.sla_rule=t2.sla_rule group by hodex, t1.sla_rule order by hodex

Dataset Name	Description	Log Category
sdwan-Device-SLA-Rule-Packetloss- Line	SD-WAN Device-SLA-Rule Packetloss Line	event
<pre>select \$flex_timescale(timestamp) as h t1.sla_rule, min(packetloss) as packetloss</pre>	odex,	
from		
(select timestamp,		
devid, sla_rule,		
<pre>sum(packetloss) / sum(count_ from</pre>	linkup) as packetloss	
<pre>###({{FGT_DATASET_EVENT_SD_ ing sum(count_linkup)>0) t1 inner</pre>	WAN_DEV_STAT}})### t group by timestamp, c join (select sla_rule, count(*) as num_ir	ntf from ###({{FGT_

ing sum(count_linkup)>0) t1 inner join (select sla_rule, count(*) as num_intf from ###({{FGT_ DATASET_EVENT_SD_WAN_DEV_STAT}))### t where \$filter-drilldown and sla_rule is not null group by sla_rule order by num_intf desc limit 10)t2 on t1.sla_rule=t2.sla_rule group by hodex, t1.sla_rule order by hodex

Dataset Name	Description	Log Category
sdwan-device-sla-intf-latency-pass- percent	SD-WAN Device Latency Pass Percentage by SLA rules and Interface	event
<pre>select sla_rule, interface, cast(100 *(1 - sum(failed_latency)/ su) as decimal(18, 2)) as latency from</pre>	ım(count_linkup)	

###({{FGT_DATASET_EVENT_SD_WAN_DEV_STAT}})### t where \$filter-drilldown group by sla_rule, interface having sum(count_linkup)>0 order by latency desc

Dataset Name	Description	Log Category
sdwan-device-sla-intf-jitter-pass- percent	SD-WAN Device Jitter Pass Percentage by SLA rules and Interface	event
<pre>select sla_rule, interface, cast(100 *(1 - sum(failed_jitter)/ su) as decimal(18, 2)) as jitter</pre>	m(count_linkup)	
<pre>from ###({{FGT DATASET EVENT SD WAN</pre>	DEV STAT}})### t where \$filter-drilldown group }	oy sla rule,

interface having sum(count linkup)>0 order by jitter desc

Dataset Name	Description	Log Category
sdwan-device-sla-intf-packetloss-pass- percent	SD-WAN Device Packet Loss Pass Percentage by SLA rules and Interface	event
<pre>select sla_rule, interface, cast(100 *(1 - sum(failed_packetloss)/) as decimal(18, 2)) as packetloss from ####({{FGT_DATASET_EVENT_SD_WAN_ interface having sum(count linkup</pre>	 DEV_STAT}})### t where \$filter-drilldown group	by sla_rule,

Dataset Name	Description	Log Category
sdwan-Device-Availability-status	SD-WAN Device Statistic by Bibandwidth	event

select devid,

sum(bibandwidth) / sum(count_linkup) as bibandwidth

from

###({{FGT_DATASET_EVENT_SD_WAN_DEV_STAT}})### t where \$filter-drilldown and bibandwidth is not null group by devid having sum(count_linkup)>0 order by bibandwidth desc

Dataset Name	Description	Log Category
sdwan-device-intf-availability- percentage-bar	SD-WAN Device Interface Availability Percentage	event

```
(
select
'SD-WAN' as interface,
```

```
cast(
   sum(availcnt)* 100.0 / sum(count) as decimal(18, 2)
 ) as available
from
  (
   select
     timestamp,
     devid,
     first value(count) OVER (
       PARTITION BY timestamp,
       devid
       ORDER BY
         link_status / count desc,
         count desc
      ) as count,
      first value(link status) OVER (
       PARTITION BY timestamp,
       devid
       ORDER BY
         link_status / count desc,
         count desc
     ) as availcnt
   from
      (
        select
         timestamp,
         devid,
         interface,
         sum(link_status) as link_status,
         sum(count) as count
        from
          ###({{FGT_DATASET_EVENT_SD_WAN_DEV_STAT}})### t where $filter-drilldown and
```

count>0 group by timestamp, devid, interface)t) t group by interface) union all (select interface, cast(sum(link_status)*100.0/sum(count) as decimal(18,2)) as available from ###(({FGT_ DATASET_EVENT_SD_WAN_DEV_STAT}))### t where \$filter-drilldown group by interface order by interface)

Dataset Name	Description	Log Category
sdwan-device-intf-availability- percentage-donut	SD-WAN Device Interface Availability Percentage Donut	event
<pre>select interface, unnest(avail) as avail, unnest(val) as val from (select interface, array['Available', 'Unavailable'] as avail, array[available, 100 - available] as val from (</pre>		

```
(
          select
            'SD-WAN' as interface,
            cast(
              sum(availcnt)* 100.0 / sum(count) as decimal(18, 2)
            ) as available
          from
            (
              select
               timestamp,
                devid,
                first value (count) OVER (
                 PARTITION BY timestamp,
                  devid
                  ORDER BY
                   link status / count desc,
                   count desc
                ) as count,
                first value(link status) OVER (
                  PARTITION BY timestamp,
                  devid
                  ORDER BY
                    link status / count desc,
                   count desc
                ) as availcnt
              from
                (
                  select
                    timestamp,
                    devid,
                    interface,
                    sum(link status) as link status,
                    sum(count) as count
                  from
                    ###({{FGT DATASET EVENT SD WAN DEV STAT}})### t where $filter-drilldown
and count>0 group by timestamp, devid, interface)t) t group by interface) union all (select
interface, cast(sum(link_status)*100.0/sum(count) as decimal(18,2)) as available from ###
```

({{FGT_DATASET_EVENT_SD_WAN_DEV_STAT}})### t where \$filter-drilldown group by interface order by interface)) t) t

Dataset Name	Description	Log Category
sdwan-Device-Application-sdwan- Rules-and-Ports-drilldown	SD-WAN Device Statistic by Bibandwidth	event
<pre>select devid, sum(bibandwidth) / sum(count_li</pre>	nkup) as bibandwidth	
<pre>from ###({{FGT_DATASET_EVENT_SD_WAN</pre>	[_DEV_STAT}})### t where \$filter-drilldown	and bibandwidth is

not null group by devid having sum(count_linkup)>0 order by bibandwidth desc

Dataset Name	Description	Log Category
sdwan-Device-Interface-Application- Traffic-Sankey	Top SD-WAN application by bandwidth sankey	traffic

select
 'SD-WAN Rules' as summary,
 'Rule:' || rulename as rule_name,
 app_group,
 devid,
 dstintf as interface,
 sum(bandwidth) as bandwidth
from
 ####({{FGT_DATASET_TRAFFIC_SD_WAN_RULES_INTF_USR}})### t where \$filter-drilldown and rulename
is not null group by rulename, app_group, devid, interface order by bandwidth desc

Dataset Name	Description	Log Category
sdwan-fw-Device-Interface-test2	SD-WAN Device-Interface Statistic	event

select

devid,

sum(bibandwidth) / sum(count_linkup) as bibandwidth

```
from
```

###({{FGT_DATASET_EVENT_SD_WAN_DEV_STAT}})### t where \$filter-drilldown and bibandwidth is not null group by devid having sum(count linkup)>0 order by bibandwidth desc

Dataset Name	Description	Log Category
sdwan-Device-Intf-Avail-Percentage- Timeline	SD-WAN Device Interface Availability Percentage Timeline	event
<pre>Timeline select hodex, interface, available from ((select \$flex_datetime(timestamp) 'SD-WAN' as interface, cast(sum(availcnt)* 100.0 / x) as available from (select timestamp, devid, first_value(count) OVI PARTITION BY timesta devid</pre>	Timeline as hodex, sum(count) as decimal(18, 2) ER (event
ORDER BY link_status / cour count desc) as count, first_value(link_statu PARTITION BY timesta devid ORDER BY	us) OVER (

```
link_status / count desc,
    count desc
) as availent
from
  (
    select
    timestamp,
    devid,
    interface,
    sum(link_status) as link_status,
    sum(count) as count
    from
    ###({{FGT_DATASET_EVENT_SD_WAN_DEV_STAT}))### t where $filter-drilldown and
```

count>0 group by timestamp, devid, interface)t) t group by hodex order by hodex) union all (select \$flex_datetime(timestamp) as hodex, interface, cast(sum(link_status)*100.0/sum(count) as decimal(18,2)) as available from ###(({FGT_DATASET_EVENT_SD_WAN_DEV_STAT}))### t where \$filter-drilldown group by hodex, interface order by hodex)) t order by hodex

Dataset Name	Description	Log Category
Top-Web-Sites-by-Bandwidth	Top web sites by bandwidth usage	webfilter
<pre>0)+coalesce(rcvdbyte, 0)) as b (countweb>0 or ((logver is nut ('webfilter', 'banned-word',</pre>	a(hostname), ipstr(`dstip`)) as domain, sum bandwidth from \$log-traffic where \$filter a ll or logver<502000000) and (hostname is no 'web-content', 'command-block', 'script-fil htbyte, 0)+coalesce(rcvdbyte, 0))>0 order k width desc	and (logflag&1>0) and ot null or utmevent in .ter')))) group by
Dataset Name	Description	Log Category
Top-App-Category-by-Session	Application risk application usage by category	traffic

```
select
```

```
appcat,
sum(sessions) as total_num
from
```

###(select appid, app, appcat, apprisk, sum(bandwidth) as bandwidth, sum(sessions) as sessions from {{FGT_DATASET_BASE_TRAFFIC_TOP_APPS}} t group by appid, app, appcat, apprisk /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t where \$filter-drilldown group by appcat order by total_num desc

```
Dataset NameDescriptionLog CategoryTop-Region-Name-by-TrafficTraffic top destination countries by browsing timetrafficselect<br/>dstcountry,<br/>sum (bandwidth) as bandwidthsbandwidthtrafficfrom<br/>#### ({{FGT_DATASET_TRAFFIC_TOP_DST_COUNTRY_BY_EB_TIME}})### t where $filter-drilldown group<br/>by dstcountry order by bandwidth desctraffic
```

Dataset Name	Description	Log Category
Top-App-By-Bandwidth-Chart	Top applications by bandwidth usage	traffic
select		
<pre>app_group_name(app) as app_g sum(</pre>	group,	
coalesce(sentbyte, 0)+ co	alesce(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 nu	11	
tronb pà		
app_group		
naving		
sum (
<pre>coalesce(sentbyte, 0)+ coalesce(sentbyte, 0)+ coalesce</pre>	alesce(rcvabyte, U)	
)& gt; 0 order by		
bandwidth desc		

Dataset Name

Top-Protocols-By-Trafiic

Top applications by bandwidth usage

select

```
service,
  sum(bandwidth) as bandwidth
from
```

###(select service, sum(bandwidth) as bandwidth from {{FGT_DATASET_BASE_TRAFFIC_BANDWIDTH_ SESSION}} base_query group by service order by bandwidth desc)### t where \$filter-drilldown group by service order by bandwidth desc

Description

Dataset Name	Description	Log Category
Top-Web-Sites-by-Sessions	Top web sites by session count	webfilter
select domain,		
sum(sessions) as sessions		

```
from
```

###(select coalesce(nullifna(hostname), ipstr(`dstip`)) as domain, count(*) as sessions
from \$log where \$filter and (eventtype is null or logver>=502000000) group by domain order by
sessions desc)### t group by domain order by sessions desc

Log Category

traffic

Dataset Name	Description	Log Category
Top-Attacks-by-Count	Threat attacks by severity	attack

attack,

sum(attack_count) as totalnum

from

###({{FGT_DATASET_ATTACK_TOP_ATTACKS}})### t where \$filter-drilldown and attack is not null
group by attack order by totalnum desc

Dataset Name	Description	Log Category
Top-Spams-by-Count	User drilldown top spam sources	emailfilter

select

```
user_src,
sum(totalnum) as totalnum
from
```

###({{FGT_DATASET_EMAILFILTER_USER_SENDER}})### t where \$filter-drilldown and mf_sender is not null group by user_src order by totalnum desc

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

select

```
virus,
```

```
max(virusid_s) as virusid,
```

```
(
```

case when virus like 'Riskware%' then 'Spyware' when virus like 'Adware%' then 'Adware' else 'Virus' end

) as malware_type, sum(totalnum) as totalnum

from

###(select virus, virusid_to_str(virusid, eventtype) as virusid_s, count(*) as totalnum from \$log where \$filter and (eventtype is null or logver>=502000000) and nullifna(virus) is not null group by virus, virusid_s /*SkipSTART*/order by totalnum desc/*SkipEND*/)### t group by virus, malware_type order by totalnum desc

Dataset Name	Description	Log Category
security-Antivirus-Inspections	Antivirus Inspections	virus
<pre>select action, sum(totalnum) as totalnum from ###({{FGT_DATASET_EMAILFILD null group by action order by</pre>	TER_USER_SENDER}})### t where \$filte 7 totalnum desc	er-drilldown and action is not
Dataset Name	Description	Log Category

Dataset Name	Description	Log Calegory
Top-DLP-by-Count	Email DLP Activity Summary	dlp

```
select
profile,
```

count(*) as total_num

```
from
```

###({{FGT_DATASET_DLP_VIOLATION_SUMMARY}})### t where \$filter-drilldown and profile is not null group by profile order by total_num desc

Dataset Name	Description	Log Category
wifi-Top-AP-By-Client	Top access point by client	traffic
<pre>select ap_srcintf as srcintf, count(distinct srcmac) as to</pre>	talnum	
	<pre>ntf) as ap_srcintf, srcssid, osname,) as devtype_new, srcmac, count(*) as</pre>	
\$filter and (logflag&1>0) and	(srcssid is not null or dstssid is no	t null) and srcmac is not

null group by ap_srcintf, srcssid, osname, srcswversion, devtype_new, srcmac order by subtotal
desc)### t group by srcintf order by totalnum desc

Dataset Name	Description	Log Category
wifi-Top-AP-By-Bandwidth	Top access point by bandwidth usage	traffic
select		
coalesce(ap, srcintf) as ap	_srcintf,	
sum(
coalesce(sentbyte, 0)+ co	alesce(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(
<pre>coalesce(sentbyte, 0)+ co</pre>	alesce(rcvdbyte, 0)	
)& gt; 0		
order by		
bandwidth 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
  $loq
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
sdwan-CTAP-Total-Bandwidth- Internal-And-External	CTAP SD-WAN Internal and External Bandwidth	traffic
<pre>select interface, bandwidth from (</pre>		
(select		
'Internal' as interface, coalesce(sum(bandwidth),		
0) as bandwidth		
	SD_WAN_RULES_INTF_USR}})### t where \$filter-d	

dstintfrole='lan') union all (select 'External' as interface, coalesce(sum(bandwidth), 0) as bandwidth from ###({{FGT_DATASET_TRAFFIC_SD_WAN_RULES_INTF_USR}})### t where \$filter-drilldown and dstintfrole='wan')) t where bandwidth>0

Dataset Name	Description	Log Category
sdwan-CTAP-Total-Bandwidth- External-Business-nonBusiness- Network	CTAP SD-WAN Bandwidth of External Business and nonBusiness	traffic
<pre>select (case when appcat not in ('Network.Service', 'Mobile' 'Proxy', 'Video\/Audio', 'O 'P2P', 'unknown') then 'Business' when appcat</pre>	Game',	

```
'Mobile', 'Social.Media', 'Proxy',
    'Video\/Audio', 'Game', 'P2P', 'unknown'
    ) then 'nonBusiness' when appcat in ('Network.Service') then 'Network Service' end
    ) as app_cat,
    coalesce(
        sum(bandwidth),
        0
    ) as bandwidth
from
    ###({{FGT DATASET TRAFFIC SD WAN RULES INTF USR}))### t where $filter-drilldown and dstint-
```

###({{FGT_DATASET_TRAFFIC_SD_WAN_RULES_INTF_USR}))### t where \$filter-drilldown and dstintfrole='wan' group by app_cat order by bandwidth desc

Dataset Name	Description	Log Category
sdwan-CTAP-Top-Appcat-Appgroup- By-Bandwidth-Sankey	CTAP SD-WAN Top SD-WAN application by bandwidth usage	traffic
select		

```
'External' as summary,
appcat,
app_group,
sum(bandwidth) as bandwidth
```

from

###({{FGT_DATASET_TRAFFIC_SD_WAN_RULES_INTF_USR}})### t where \$filter-drilldown and dstintfrole='wan' and bandwidth>0 group by appcat, app_group order by bandwidth desc

Dataset Name	Description	Log Category
sdwan-CTAP-Business-Apps- Bandwidth	CTAP SD-WAN Business Application with Bandwidth	traffic

select

```
app_group,
sum(bandwidth) as bandwidth
```

from

###({{FGT_DATASET_TRAFFIC_SD_WAN_RULES_INTF_USR}})### t1 inner join app_mdata t2 on lower (t1.app_group)=lower(t2.name) where \$filter-drilldown and appcat not in ('Network.Service', 'Mobile','Social.Media','Proxy','Video\/Audio','Game','P2P','unknown') and dstintfrole='wan' group by app_group order by bandwidth desc, app_group

Dataset Name	Description	Log Category
sdwan-CTAP-Cloud-IT-Apps- Bandwidth	CTAP SD-WAN Cloud IT Application E	Bandwidth traffic
	lth C_SD_WAN_RULES_INTF_USR}})### t where =='wan' and bandwidth>0 group by app_g	
Dataset Name	Description	Log Category

Dataset Name	Description	Log Category
sdwan-CTAP-Storage-Backup-Apps- Bandwidth	CTAP SD-WAN Storage Backup Application Bandwidth	traffic

app_group,

sum(bandwidth) as bandwidth

```
from
```

###({{FGT_DATASET_TRAFFIC_SD_WAN_RULES_INTF_USR}})### t where \$filter-drilldown and appcat-='Storage.Backup' and dstintfrole='wan' and bandwidth>0 group by app_group order by bandwidth desc

Dataset Name	Description	Log Category
Dataset Name	Description	Log Calegory
sdwan-CTAP-Collaboration-Apps- Bandwidth	CTAP SD-WAN Collaboration Application Bandwidth	traffic
	N_RULES_INTF_USR}})### t where \$filter-drilldowr 'wan' and bandwidth>0 group by app_group order b	
Dataset Name	Description	Log Category
sdwan-CTAP-Top-Streaming-App-By- Bandwidth	CTAP SD-WAN Top Streaming Application by Bandwidth	traffic
	N_RULES_INTF_USR}})### t where \$filter-drilldowr wan' and bandwidth>0 group by app_group order by	
Dataset Name	Description	Log Category
sdwan-CTAP-Top-SocialMedia-App- By-Bandwidth	CTAP SD-WAN Top SocialMedia Application by Bandwidth	traffic

select

```
app_group,
  sum(bandwidth) as bandwidth
from
```

```
###({{FGT_DATASET_TRAFFIC_SD_WAN_RULES_INTF_USR}})### t where $filter-drilldown and appcat-
='Social.Media' and dstintfrole='wan' and bandwidth>0 group by app_group order by bandwidth
desc
```

Dataset Name	Description	Log Category
sdwan-CTAP-App-Risk-Reputation- Top-Devices-By-Scores	Reputation Top Devices By-Scores	traffic
select coalesce(

```
nullifna(`srcname`),
```

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

Dataset Name	Description	Log Category
sdwan-CTAP-SB-Top-Sandbox-Files	CTAP SD-WAN Sandbox Top Sandbox Files	virus
<pre>select filename, analyticscksum, service, sum(totalnum) as total_num, (case fsaverdict when 'maliciou 'medium risk' then 'Medium' when) as risk, (</pre>	us' then 'Malicious' when 'high risk' ther 'low risk' then 'Low' else 'Other' end us' then 5 when 'high risk' then 4 when 'r	n 'High' when
<pre>from ###({{FGT DATABASE VIRUS TOP MAI</pre>	LWARE NAME}})### t where \$filter-drilldown	n and filename is

###({{FGT_DATABASE_VIRUS_TOP_MALWARE_NAME}))### t where \$filter-drilldown and filename is not null and dtype='fortisandbox' and fsaverdict not in ('clean', 'submission failed') group by filename, analyticscksum, risk_level, risk, service order by risk_level desc, total_num desc, service, filename

Dataset Name	Description	Log Category
sdwan-CTAP-SB-Total-Number-of- Malicious-Suspicious-Files	CTAP SD-WAN Sandbox Malicious Suspicious Files Number	virus
<pre>'medium risk' then 'Medium' when) as risk, sum(totalnum) as total_num from</pre>	us' then 'Malicious' when 'high risk' then 'Hig 'low risk' then 'Low' else 'Other' end LWARE_NAME}})### t where \$filter-drilldown and	

='fortisandbox' and fsaverdict not in ('clean','submission failed') group by risk order by total_num desc

Dataset Name	Description	Log Category
sdwan-CTAP-Top-Source-Countries	CTAP SD-WAN Top Source Countries	traffic

select

srccountry,

sum(bandwidth) as bandwidth

from

###({{FGT_DATASET_TRAFFIC_SD_WAN_RULES_INTF_USR}})### t where \$filter-drilldown and nullifna
(srccountry) is not null and srccountry <> 'Reserved' and bandwidth>0 group by srccountry
order by bandwidth desc, srccountry

Dataset Name	Description	Log Category
sdwan-CTAP-Average-Bandwidth-Day- Hour	CTAP SD-WAN Average Bandwidth by Day of Week and Hour	traffic
<pre>select hourstamp, daystamp, round(sum(bandwidth) / count(*)) as bandwidth from (</pre>		
	hour_stamp, daystamp,	

Dataset Name	Description	Log Category
sdwan-CTAP-Average-Log-Rate-By- Hour	CTAP SD-WAN Average Log Rate by Hour	event
<pre>select \$hour_of_day(timestamp) as hour cast((sum(total_trate + total_erate))/ sum(count) as decimal(10,) as log_rate</pre>	e + total_orate	
<pre>from ###({{FGT_DATASET_EVENT_DISK_LC hourstamp order by hourstamp</pre>	DGRATE_CPU_MEM}})### t where \$filter-dril	lldown group by

) as domain,

) as bandwidth,

) as traffic in,

) as traffic out

logflag&1>0

and coalesce(nullifna(

coalesce(rcvdbyte, 0)

coalesce(sentbyte, 0)

sum(

sum(

sum(

from \$log where \$filter and (

)

),

```
Log Category
 Dataset Name
                                    Description
sdwan-CTAP-CPU-Usage-Per-Hour
                                    Event usage CPU
                                                                                      event
select
 $hour_of_day(timestamp) as hourstamp,
 cast(
   sum(total_cpu) / sum(count) as decimal(6, 2)
 ) as cpu avg usage
from
  ###({{FGT DATASET EVENT DISK LOGRATE CPU MEM}})### t group by hourstamp order by hourstamp
 Dataset Name
                                                                                      Log Category
                                    Description
sdwan-CTAP-Memory-Usage-Per-Hour Event usage memory
                                                                                      event
select
 $hour of day(timestamp) as hourstamp,
 cast(
   sum(total_mem) / sum(count) as decimal(6, 2)
 ) as mem_avg_usage
from
  ###({{FGT_DATASET_EVENT_DISK_LOGRATE_CPU_MEM}})### t group by hourstamp order by hourstamp
 Dataset Name
                                    Description
                                                                                      Log Category
Top-Destination-Addresses-By-
                                                                                      traffic
                                   Top destinations by bandwidth usage
 Bandwidth-Bar
select
 coalesce(
   nullifna(
     root domain (hostname)
   ),
   ipstr(dstip)
```

FortiAnalyzer 6.4.4 Dataset Reference Fortinet Technologies Inc.

root_domain(hostname)

coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)

```
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
intf-Timeline-Sampling	Interface Utilization Timeline by Data Sampling	event
with base_qry as (
select		
tm,		
rcvdbps,		
ntile(100) over (
order by		
rcvdbps		
) as percentile		
from		
(
select (timestamp / 300 *	200) ac tm	
sum(rcvdbps) as rcv		
300 as interval		
from		
intfstats_billing t	bl	
join (
select		
ti.dvid,		
intfname		
from		
intfinfo ti		
	able td on ti.dvid = td.dvid	
where		
\$dev_filter) tb2 on tb1.dvid =		
and tb1.intfname =		
where	CD2.Inciname	
\$cust_time_filter(t	imestamp)	
group by	Server	
tm		
) tmp		
),		
ref_qry as (
select		
<pre>max(rcvdbps) as ref_val</pre>		
from		
base_qry		
where percentile = 95		
)		
1		

```
select
 from_itime(timestamp) as tmstamp,
 rcvdbps,
 ref_val
from
 ref_qry,
  (
   select
     tm as timestamp,
     rcvdbps,
     rank() over(
      partition by (tm / 3600)
       order by
         tm
     ) as r
   from
     base_qry
 ) t
where
 r = 1
order by
 tmstamp
```

Dataset Name	Description	Log Category
intf-Util-Histogram	Interface Utilization Value Distribution	event
select		
cast (
(
(
<pre>max(max_value) over ()</pre>		
)* seq / 100		
) as decimal(16, 0)		
) as value, cnt		
from		
(
select		
generate_series(0, 100, 2)	as seq	
) t1	-	
left join (
select		
perc,		
max_value,		
count(*) as cnt		
from		
select		
WIDTH BUCKET(
rcvdbps,		
0,		
(
max(rcvdbps) over	()	
) + 1,		
50		

```
)* 2 as perc,
         max(rcvdbps) over () as max_value
        from
          (
            select
              (timestamp / 300 * 300) as tm,
              sum(rcvdbps) as rcvdbps,
              300 as interval
            from
              intfstats_billing tb1
              join (
                select
                  ti.dvid,
                  intfname
                from
                  intfinfo ti
                  left join devtable td on ti.dvid = td.dvid
                where
                  $dev filter
              ) tb2 on tb1.dvid = tb2.dvid
              and tb1.intfname = tb2.intfname
            where
              $cust_time_filter(timestamp)
            group by
              tm
          ) tmp
      ) t bucket
   group by
     perc,
     max value
  ) t2 on t1.seq = t2.perc
order by
  seq
```

Dataset Name

Dataset Name	Description	
intf-Sorted-Line	Interface Utilization Line Sorted by bps	event
with base_qry as (
select		
rcvdbps,		
ntile(100) over (
order by		
rcvdbps		
) as percentile		
from		
(
select		
(timestamp / 300 * 300) as tm,	
sum(rcvdbps) as rcvdbp	5,	
300 as interval		
from		
intfstats_billing tb1		
join (
select		
ti.dvid,		

Description

Log Category

```
intfname
          from
            intfinfo ti
            left join devtable td on ti.dvid = td.dvid
          where
            $dev_filter
        ) tb2 on tb1.dvid = tb2.dvid
        and tb1.intfname = tb2.intfname
      where
        $cust_time_filter(timestamp)
      group by
        tm
   ) tmp
),
ref_qry as (
 select
   max(rcvdbps) as ref_val
  from
   base_qry
 where
   percentile = 95
)
select
 n_perc,
 rcvdbps,
 ref_val
from
  (
   select
     seq as n perc,
     rcvdbps
    from
      (
        select
         generate series(0, 100, 1) as seq
      ) t1
      left join (
       select
         max(rcvdbps) as rcvdbps,
         percentile
        from
         base_qry
        group by
         percentile
     ) t2 on t1.seq = t2.percentile
  ) t,
  ref_qry
order by
```

```
n_perc
```

Dataset Name	Description	Log Category
intf-Data-Analysis-Table	Interface Utilization Data Analysis	event
with base_qry as (select		

```
rcvdbps,
    interval,
    ntile(100) over (
      order by
        rcvdbps
    ) as percentile
  from
    (
      select
        (timestamp / 300 * 300) as tm,
        sum(rcvdbps) as rcvdbps,
        300 as interval
      from
        intfstats_billing tb1
        join (
          select
            ti.dvid,
            intfname
          from
            intfinfo ti
            left join devtable td on ti.dvid = td.dvid
          where
            $dev filter
        ) tb2 on tb1.dvid = tb2.dvid
        and tb1.intfname = tb2.intfname
      where
        $cust time filter(timestamp)
      group by
        tm
    ) tmp
)
select
 min mbps,
 low ref mbps,
 mean mbps,
 ref mbps,
 peak_mbps,
 actual gb,
  total
from
  (
    select
     cast(
       min(rcvdbps) / 1000000 as decimal(18, 2)
     ) as min_mbps,
      cast(
        avg(rcvdbps) / 1000000 as decimal(18, 2)
      ) as mean_mbps,
      cast(
        max(rcvdbps) / 1000000 as decimal(18, 2)
      ) as peak mbps,
      cast(
        (
          select
            max(rcvdbps)
          from
```

```
base_qry
        where
          percentile = 5
      )/ 1000000 as decimal(18, 2)
    ) as low ref mbps,
    cast(
      (
        select
         max(rcvdbps)
        from
         base qry
       where
         percentile = 95
     )/ 1000000 as decimal(18, 2)
    ) as ref mbps,
    cast(
      sum(interval * rcvdbps)/ 8 /(1024 * 1024 * 1024) as decimal(18, 2)
    ) as actual gb,
    count(*) as total
  from
   base_qry
) t
```

```
Dataset NameDescriptionLog Category360-degree-security-Application-<br/>Visibility-and-Control-SummaryApplication Visibolity and Control Summaryapp-ctrl
```

```
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-	Threat Prevention	app-ctrl
Detection-and-Prevention-Summary		

select

```
threat_name,
    count(distinct threats) as total_num
from
```

(

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

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

```
data_loss,
  count(*) as total_num
```

from

###({{FGT_DATASET_DLP_VIOLATION_SUMMARY}})### t where \$filter-drilldown and data_loss is not null group by data_loss order by total_num desc

Dataset Name	Description	Log Category
360-degree-security-Endpoint- Protection-Summary	Endpoint Protection	fct-traffic
<pre>then 'Risk Application Blocked</pre>	tivirus' then 'Malware Deteced and ' when 'webfilter' then (Web Sites Violation Blocked' else '	
<pre>its' end) else NULL end) as blocked_event from \$log where \$filter and utmaction in ('blocked) t where blocked_event is not null group by blocked_event order by total_num desc</pre>		

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.

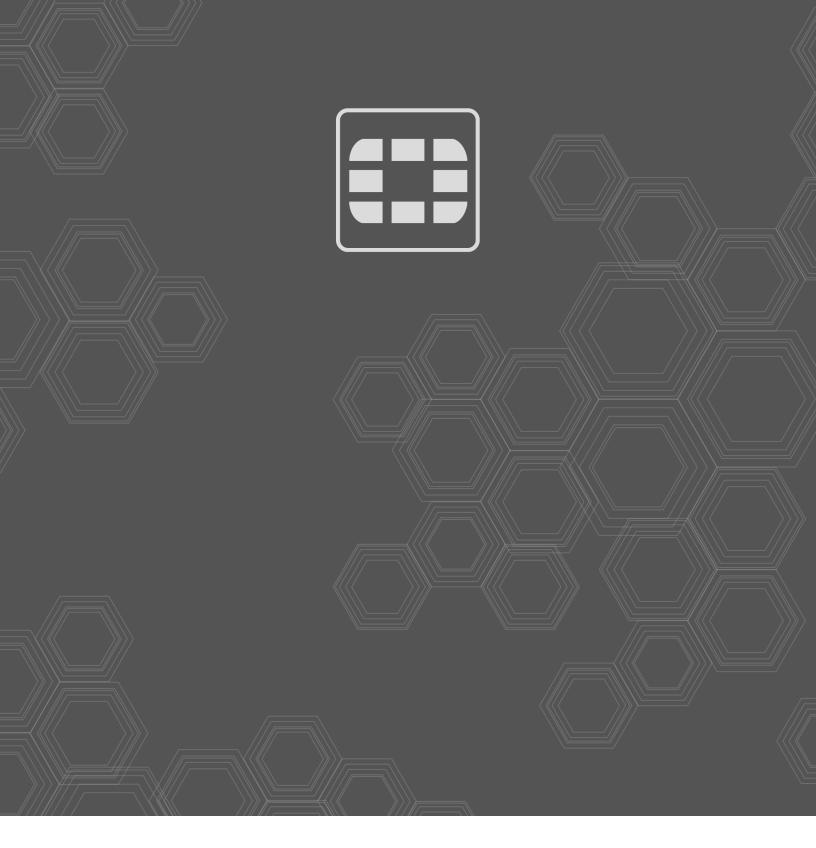
Macro Name	Description	Dataset Used	Log Category
Application Category with Highest Session Count	Application category with the highest session count	App-Sessions-By- Category	Traffic
Application with Highest Bandwidth	Application with the highest bandwidth usage	Top-App-By-Bandwidth	Traffic
Application 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
2020-12-16	Initial release.





Copyright© 2020 Fortinet, Inc. All rights reserved. Fortinet®, FortiGate®, FortiGare® and FortiGuard®, and certain other marks are registered trademarks of Fortinet, Inc., in the U.S. and other jurisdictions, and other Fortinet names herein may also be registered and/or common law trademarks of Fortinet. All other product or company names may be trademarks of their respective owners. Performance and other metrics contained herein were attained in internal lab tests under ideal conditions, and actual performance and other results may vary. Network variables, different network environments and other conditions may affect performance results. Nothing herein represents any binding commitment by Fortinet, and Fortinet disclaims all warranties, whether express or implied, except to the extent Fortinet enters a binding written contract, signed by Fortinet's General Counsel, with a purchaser that expressly warrants that the identified product will perform according to certain expressly-identified performance metrics and, in such event, only the specific performance metrics expressly identified in such binding written contract shall be binding on Fortinet. For absolute clarity, any such warranty will be limited to performance in the same ideal conditions as in Fortinet's internal lab tests. In no event does Fortinet make any commitment related to future deliverables, features or development, and circumstances may change such that any forward-looking statements herein are not accurate. Fortinet disclaims in full any covenants, representations, and guarantees pursuant hereto, whether express or implied. Fortinet reserves the right to change, modify, transfer, or otherwise revise this publication without notice, and the most current version of the publication shall be applicable.