



FortiAnalyzer - Dataset Reference

Version 6.2.8



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://www.fortiguard.com

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

FEEDBACK Email: techdoc@fortinet.com



May 13, 2021 FortiAnalyzer 6.2.8 Dataset Reference 05-628-547774-20210513

TABLE OF CONTENTS

Introduction	4
Understanding datasets and macros	4
Dataset Reference List	5
Macro Reference List 27	'9
Change Log 28	2

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
<pre>(traffic_in) as traffic_in from timestamp as timestamp, devid, v ipstr(`srcip`)) as user_src, ser 0)+coalesce(rcvdbyte, 0)) as ban (coalesce(rcvdbyte, 0)) as traff timestamp, devid, vd, csf, user_</pre>	<pre>t, width) as bandwidth, sum(traffic_out) as traff ###base(/*tag:rpt_base_t_bndwdth_sess*/select d, csf, coalesce(nullifna(`user`), nullifna(`u vice, count(*) as sessions, sum(coalesce(senth dwidth, sum(coalesce(sentbyte, 0)) as traffic_ ic_in from \$log where \$filter and (logflag&1>0 src, service /*SkipSTART*/order by timestamp ry group by timestamp order by bandwidth desc)</pre>	<pre>\$flex_ inauthuser`), oyte, out, sum) group by</pre>

Dataset Name	Description	Log Category
Session-Summary-Day-Of-Month	Number of session timeline	traffic
<pre>sess*/select \$flex_timestamp as nullifna(`unauthuser`), ipstr(`s (coalesce(sentbyte, 0)+coalesce) traffic_out, sum(coalesce(rcvdby (logflag&1>0) group by timestamp</pre>	<pre>hodex, sions) as sessions from ###base(/*tag:rpt_ba timestamp, devid, vd, csf, coalesce(nullifn srcip`)) as user_src, service, count(*) as s (rcvdbyte, 0)) as bandwidth, sum(coalesce(se yte, 0)) as traffic_in from \$log where \$filt b, devid, vd, csf, user_src, service /*Skips ## base_query group by timestamp order by set</pre>	na(`user`), sessions, sum entbyte, 0)) as ter and START*/order by
Dataset Name	Description	Log Category
Top-Users-By-Bandwidth	Bandwidth application top users by bandwidth usage	traffic

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

```
Dataset Reference List
```

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

Dataset Name	Description		Log Category
Top-App-By-Bandwidth	Top application:	s 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>	<pre>0)+ coalesce(rcvdbyte, 0)</pre>	0)	
) as traffic_out, count(*) as sessions from \$log where	0)		
<pre>\$filter and (logflag&1>0) and nullifna(app) is</pre>	not null		
<pre>group by app_group having sum(coalesce(sentbyte,)& gt; 0</pre>	0)+ coalesce(rcvdbyte,	0)	

order by bandwidth desc

Dataset Name	Description	Log Category
Top-User-Source-By-Sessions	Top user source by session count	traffic
<pre>select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, count(*) as sessions from \$log</pre>		
where \$filter and (logflag&1>0		
) group by user_src order by sessions desc		

Dataset Name	Description	Log Category
Top-App-By-Sessions	Top applications by session count	traffic
<pre>select app_group_name(app) as ap</pre>	n groun.	
count(*) as sessions	₽_3~₽,	
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(hostname)),</pre>		

```
ipstr(dstip)
) as domain,
count(*) as sessions
from
  $log
where
  $filter
  and (
     logflag&1>0
 )
group by
  domain
order by
  sessions desc
```

Dataset Name	Description	Log Category
Top-Destination-Addresses-By- Bandwidth	Top destinations by bandwidth usage	traffic
<pre>select coalesce(</pre>		

```
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; drop table if exists rpt_tmptbl_2; drop table if exists rpt_tmptbl_3; (interface, '.', devid) as intf, int(logid) = 26001 and dhcp_msg table rpt_tmptbl_2 as ###(select \$filter and logid_to_int(logid) mac)###; create temporary table (used*100.0/total as decimal(18, (1) concat(interface, '.', devid logid_to_int(logid)=26003 and to</pre>	<pre>create temporary table rpt_tmptbl_1 as ###(se mac from \$log where \$last3day_period \$filter = 'Ack' group by interface, devid, mac)###; c concat(interface, '.', devid) as intf, mac f = 26001 and dhcp_msg = 'Ack' group by interfa rpt_tmptbl_3 as select distinct on (1) intf, 2)) as percent_of_allocated_ip from ###(selec) as intf, used, total, itime from \$log where tal>0 /*SkipSTART*/order by intf, itime desc/</pre>	<pre>lect concat and logid_to_ reate temporary rom \$log where ce, devid, cast t distinct on \$filter and *SkipEND*/)###</pre>
<pre>count from rpt_tmptbl_3 t1 inner tmptbl_2 where not exists (selec</pre>	<pre>ect t1.intf as interface, percent_of_allocate join (select intf, count(mac) as new_cli_cou t 1 from rpt_tmptbl_1 where rpt_tmptbl_2.mac= ntf=t2.intf order by interface, percent_of_al</pre>	nt from rpt_ rpt_tmptbl_

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, osn coalesce(nullifna(`srcname`), `srcmac`) as hostname_mac, sum(coalesce(sentbyte, 0)+ coal) as bandwidth</pre>	_	
from \$log		
where		
<pre>\$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)</pre>	as hodex,	

count (distinct(user_src)) as total user

from

###(select timestamp, user_src, sum(sessions) as sessions from ###base(/*tag:rpt_base_t_ bndwdth sess*/select \$flex timestamp as timestamp, devid, vd, csf, coalesce(nullifna (`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user src, service, count(*) as sessions, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce (sentbyte, 0)) as traffic_out, sum(coalesce(rcvdbyte, 0)) as traffic_in from \$log where \$filter and (logflag&1>0) group by timestamp, devid, vd, csf, user src, service /*SkipSTART*/order by timestamp desc/*SkipEND*/)base### base query group by timestamp, user src order by sessions desc)### t group by hodex order by hodex

Dataset Name	Description	Log Category
Top-Allowed-Websites-By-Requests	UTM top allowed web sites by request	traffic
<pre>select hostname, catdesc, count(*) as requests from \$log where \$filter and (logflag&1>0) and utmevent in ('webfilter', 'banned-word', 'command-block', 'script-fil) and hostname is not null and (utmaction not in ('block', ' or action != 'deny')</pre>	ter'	
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
<pre>select domain, string_agg(distinct catdesc, sum(bandwidth) as bandwidth, sum(traffic_in) as traffic_in, sum(traffic out) as traffic out)</pre>	,	
<pre>from ###(select coalesce(nullifna(h (sentbyte, 0)+coalesce(rcvdbyte, sum(coalesce(sentbyte, 0)) as th and utmaction!='blocked' and (coalesce)</pre>	nostname), ipstr(`dstip`)) as domain, catdesc, , 0)) as bandwidth, sum(coalesce(rcvdbyte, 0)) raffic_out from \$log-traffic where \$filter and buntweb>0 or ((logver is null or logver<502000 nt in ('webfilter', 'banned-word', 'web-conten	as traffic_in, d (logflag&1>0) 0000) and

block', 'script-filter')))) group by domain, catdesc having sum(coalesce(sentbyte,

0)+coalesce(rcvdbyte, 0))>0 /*SkipSTART*/order by bandwidth desc/*SkipEND*/)### t group by domain, catdesc order by bandwidth desc

Dataset Name	Description	Log Category
Top-Blocked-Websites	UTM top blocked web sites by request	traffic
select hostname, count(*) as requests		
from \$log		
<pre>where \$filter and (logflag&1>0) and utmevent in ('webfilter', 'banned-wo: 'command-block', 'script) and hostname is not null and (utmaction in ('block', or action = 'deny')</pre>	t-filter'	
group by hostname order by requests desc		
Dataset Name	Description	Log Category
Top-Web-Users-By-Request	UTM top web users by request	traffic

```
select
 coalesce(
   nullifna(`user`),
   nullifna(`unauthuser`),
   ipstr(`srcip`)
  ) as user src,
  get devtype(srcswversion, osname, devtype) as devtype new,
 srcname,
 count(*) as requests
from
  $log
where
  $filter
 and (
   logflag&1>0
  )
  and utmevent in (
    'webfilter', 'banned-word', 'web-content',
    'command-block', 'script-filter'
  )
group by
 user_src,
  devtype new,
 srcname
order by
  requests desc
```

Dataset Name Description Log Category Top-Allowed-WebSites-By-Bandwidth UTM top allowed websites by bandwidth usage traffic select appid, hostname, catdesc, sum(coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic in, sum(coalesce(sentbyte, 0)) as traffic out from \$log where \$filter and (logflag&1>0) and utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') and hostname is not null

```
group by
appid,
hostname,
catdesc
having
sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
)& gt; 0
order by
bandwidth desc
```

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

```
select
 coalesce(
   nullifna(`user`),
   nullifna(`unauthuser`),
   ipstr(`srcip`)
 ) as user src,
 get_devtype(srcswversion, osname, devtype) as devtype_new,
 srcname,
  count(*) as requests
from
  $loq
where
 $filter
  and (
    logflag&1>0
 )
  and utmevent in (
    'webfilter', 'banned-word', 'web-content',
    'command-block', 'script-filter'
  )
  and (
   utmaction in ('block', 'blocked')
   or action = 'deny'
 )
group by
 user_src,
 devtype_new,
 srcname
order by
  requests desc
```

Dataset Name	Description	Log Category
Top-20-Web-Users-By-Bandwidth	Webfilter top web users by bandwidth usage	webfilter
<pre>select user_src, sum(bandwidth) as bandwidth, sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_out from</pre>	:	

###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out from \$log-traffic where \$filter and (logflag&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', 'scriptfilter')))) 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
<pre>select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, get_devtype(srcswversion, osna srcname, sum(coalesce(sentbyte, 0)+ coale) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in,</pre>	_	
<pre>sum(coalesce(sentbyte, 0)) as traffic_out from \$log where \$filter and (logflag&1>0)</pre>		
<pre>and utmevent in ('webfilter', 'banned-word', 'command-block', 'script-fil)</pre>		
<pre>group by user_src, devtype_new, srcname</pre>		
<pre>having sum(coalesce(sentbyte, 0)+ coale) & gt; 0 order by bandwidth desc</pre>	esce(rcvdbyte, 0)	

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

```
select
  appid,
  hostname,
  sum(
   coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
  ) as bandwidth,
  sum(
    coalesce(rcvdbyte, 0)
  ) as traffic_in,
  sum(
   coalesce(sentbyte, 0)
  ) as traffic_out
from
  $log
where
 $filter
  and (
    logflag&1>0
  )
  and catdesc in ('Streaming Media and Download')
group by
  appid,
  hostname
having
  sum(
   coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
 )& gt; 0
order by
  bandwidth desc
```

Dataset Name	Description	Log Category
Top-Email-Senders-By-Count	Default top email senders by count	traffic
<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', '2 'smtps', 'SMTPS', '465/tcp') group by user_src order by requests desc</pre>		

Dataset Name	Description	Log Category
Top-Email-Receivers-By-Count	Default email top receivers by count	traffic
select		
coalesce(
<pre>nullifna(`user`),</pre>		
<pre>nullifna(`unauthuser`),</pre>		
ipstr(`srcip`)		
) as user_src,		
count(*) as requests		
from		
\$log		
where		
\$filter		
and (
logflag&1>0		
)		
and service in (
'pop3', 'POP3', '110/tcp'	-	
'IMAP', '143/tcp', 'imaps		
'993/tcp', 'pop3s', 'POP3	S', '995/tcp'	
)		
group by		
user_src order by		
requests desc		
requests desc		
Dataset Name	Description	Log Category

Top-Email-Senders-By-Bandwidth Default email top senders by bandwidth usage select coalesce(nullifna(`user`), nullifna(`unauthuser`),	
<pre>coalesce(nullifna(`user`), nullifna(`unauthuser`),</pre>	ge traffic
<pre>nullifna(`user`), nullifna(`unauthuser`),</pre>	
<pre>nullifna(`unauthuser`),</pre>	
ipstr(`srcip`)	
) as user src,	
sum (
<pre>coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)</pre>	
) 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`),		
<pre>nullifna(`unauthuser`),</pre>		
ipstr(`srcip`)		
) as user_src,		
sum(
<pre>coalesce(sentbyte, 0)+ coale</pre>	esce(rcvdbyte, 0)	
) as bandwidth		
from		
\$log		
where		
\$filter		
and (
logflag&1>0		
)		
and service in (
'pop3', 'POP3', '110/tcp',		
'IMAP', '143/tcp', 'imaps',		
'993/tcp', 'pop3s', 'POP3S',	, '995/tcp'	
)		
group by		
user_src		
having		
sum(
<pre>coalesce(sentbyte, 0) + coale</pre>	esce(rcvdbyte, 0)	
)& gt; 0		
order by		
bandwidth desc		

Dataset Name	Description	Log Category
Top-Malware-By-Name	UTM top virus	virus
select		
virus,		
<pre>max(virusid_s) as virusid</pre>	1	
(
case when virus like 'R	iskware%' then 'Spyware' when	n virus like 'Adware%' then 'Adware'
else 'Virus' end		
) as malware_type,		
sum(totalnum) as totalnum		
from		
###(select virus, virusid	_to_str(virusid, eventtype) a	as virusid_s, count(*) as totalnum
-	rusid_s /*SkipSTART*/order b	r>=502000000) and nullifna(virus) is y totalnum desc/*SkipEND*/)### t

Top-Virus-By-Name UTM top virus virus select
virus,
max(virusid_s) as virusid,
(
case when virus like 'Riskware%' then 'Spyware' when virus like 'Adware%' then 'Adwa
else 'Virus' end
) as malware_type,
sum(totalnum) as totalnum
from
<pre>###(select virus, virusid_to_str(virusid, eventtype) as virusid_s, count(*) as totalnu</pre>

###(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
from \$log where \$filter and (even	user`), ipstr(`srcip`)) as user_src, c nttype is null or logver>=502000000) a pSTART*/order by totalnum desc/*SkipEN	and nullifna(virus) is

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
select dstip,		

```
count(*) as totalnum
from
  $log
where
  $filter
  and dstip is not null
group by
  dstip
order by
  totalnum desc
```

Dataset Name	Description	Log Category
Top-Static-IPSEC-Tunnels-By- Bandwidth	Top static IPsec tunnels by bandwidth usage	event
<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>in) - 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 ma</pre>	
<pre>out) - min(min_traffic_out) end) as traffic_out, (</pre>	ax(e time) then max(max traffic in)+ max(max	
—	n_traffic_in) + max(max_traffic_out) - min(min	

###(select devid, vd, remip, vpn_trim(vpntunnel) as vpn_name, tunnelid, tunnelip, max (coalesce(sentbyte, 0)) as max_traffic_out, max(coalesce(rcvdbyte, 0)) as max_traffic_in, min(coalesce(sentbyte, 0)) as min_traffic_out, min(coalesce(rcvdbyte, 0)) as min_traffic_in, min(coalesce(dtime, 0)) as s_time, max(coalesce(dtime, 0)) as e_time from \$log where \$filter and subtype='vpn' and tunneltype like 'ipsec%' and nullifna(vpntunnel) is not null and action in ('tunnel-stats', 'tunnel-down') and tunnelid is not null group by devid, vd, remip, vpn_name, tunnelid, tunnelip)### t where (tunnelip is null or tunnelip='0.0.0.0') group by devid, vd, remip, vpn_name, tunnelid) tt group by vpn_name having sum(traffic_ in+traffic_out)>0 order by bandwidth desc

Dataset Name	Description	Log Category
Top-SSL-VPN-Tunnel-Users-By- Bandwidth	Top SSL VPN tunnel users by bandwidth usage	event
<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_co from (select devid, vd, remip, user_src, tunnelid, min(s_time) as s_time, max(e_time) as e_time, (</pre>		
—	<pre>max(e_time) then max(max_traffic_in)+ max(m nin_traffic_in)+ max(max_traffic_out)- min(m</pre>	
<pre>) as bandwidth, (</pre>	<pre>max(e_time) then max(max_traffic_in) else m</pre>	ax(max_traffic_
	<pre>max(e_time) then max(max_traffic_out) else</pre>	max(max_traffic_
<pre>tunnelid, tunneltype, max(coale as min_duration, min(coalesce(d) (coalesce(sentbyte, 0)) as min_ max(coalesce(sentbyte, 0)) as min_ from \$log where \$filter and sub stats', 'tunnel-down', 'tunnel- null and tunnelid is not null g t where tunneltype='ssl-tunnel'</pre>	<pre>hip, coalesce(nullifna(`user`), ipstr(`remip esce(duration,0)) as max_duration, min(coale dtime, 0)) as s_time, max(coalesce(dtime, 0) traffic_out, min(coalesce(rcvdbyte, 0)) as max_traffic_out, max(coalesce(rcvdbyte, 0)) otype='vpn' and tunneltype like 'ssl%' and a oup') and coalesce(nullifna(`user`), ipstr(` group by devid, vd, user_src, remip, tunneli group by devid, vd, user_src, remip, tunne remote_ip order by bandwidth desc</pre>	<pre>sce(duration,0))) as e_time, min min_traffic_in, as max_traffic_in ction in ('tunnel- remip`)) is not d, tunneltype)###</pre>

Dataset Name	Description	Log Category
Top-Dial-Up-IPSEC-Tunnels-By- Bandwidth	Top dial up IPsec tunnels by bandwidth usage	event
<pre>select vpn_name,</pre>		

```
sum(bandwidth) as bandwidth,
  sum(traffic in) as traffic in,
  sum(traffic_out) as traffic_out
from
  (
   select
     devid,
     vd,
     tunnelid,
     remip,
     vpn name,
      (
       case when min(s time) = max(e time) then max(max traffic in) else max(max traffic
in) - min(min traffic in) end
     ) as traffic in,
      (
        case when min(s time) = max(e time) then max(max traffic out) else max(max traffic
out) - min(min traffic out) end
      ) as traffic out,
      (
        case when min(s_time) = max(e_time) then max(max_traffic_in) + max(max_traffic_out)
else max(max_traffic_in) - min(min_traffic_in) + max(max_traffic_out) - min(min_traffic_out)
end
     ) as bandwidth
   from
      ###(select devid, vd, remip, vpn_trim(vpntunnel) as vpn_name, tunnelid, tunnelip, max
(coalesce(sentbyte, 0)) as max traffic out, max(coalesce(rcvdbyte, 0)) as max traffic in,
min(coalesce(sentbyte, 0)) as min traffic out, min(coalesce(rcvdbyte, 0)) as min traffic in,
min(coalesce(dtime, 0)) as s time, max(coalesce(dtime, 0)) as e time from $log where $filter
and subtype='vpn' and tunneltype like 'ipsec%' and nullifna(vpntunnel) is not null and
action in ('tunnel-stats', 'tunnel-down') and tunnelid is not null group by devid, vd,
remip, vpn name, tunnelid, tunnelip)### t where not (tunnelip is null or tunnelip='0.0.0.0')
group by devid, vd, remip, vpn name, tunnelid) tt group by vpn name having sum(traffic
out+traffic in)>0 order by bandwidth desc
```

Dataset Name	Description	Log Category
Top-Dial-Up-IPSEC-Users-By- Bandwidth	Top dial up IPsec users by bandwidth usage	event
<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_out from (select</pre>	t	

```
devid,
     vd,
      string_agg(distinct xauthuser_agg, ' ') as xauthuser_agg,
      string agg(distinct user agg, ' ') as user agg,
     remip,
     tunnelid,
     min(s_time) as s_time,
     max(e time) as e_time,
        case when min(s time) = max(e time) then max(max traffic in) + max(max traffic out)
else max(max traffic in) - min(min traffic in) + max(max traffic out) - min(min traffic out)
end
      ) as bandwidth,
      (
        case when min(s time) = max(e time) then max(max traffic in) else max(max traffic
in) - min(min traffic in) end
      ) as traffic in,
      (
        case when min(s time) = max(e time) then max(max traffic out) else max(max traffic
out) - min(min traffic out) end
      ) as traffic out
    from
      ###(select devid, vd, remip, nullifna(`xauthuser`) as xauthuser agg, nullifna(`user`)
as user agg, tunnelid, min(coalesce(dtime, 0)) as s time, max(coalesce(dtime, 0)) as e time,
max(coalesce(duration,0)) as max_duration, min(coalesce(duration,0)) as min_duration, min
(coalesce(sentbyte, 0)) as min_traffic_out, min(coalesce(rcvdbyte, 0)) as min_traffic_in,
max(coalesce(sentbyte, 0)) as max traffic out, max(coalesce(rcvdbyte, 0)) as max traffic in
```

from \$log where \$filter and subtype='vpn' and tunneltype like 'ipsec%' and not (tunnelip is null or tunnelip='0.0.0.0') and action in ('tunnel-stats', 'tunnel-down', 'tunnel-up') and tunnelid is not null and tunnelid!=0 group by devid, vd, remip, xauthuser_agg, user_agg, tunnelid order by tunnelid)### t group by devid, vd, remip, tunnelid) t where bandwidth>0 group by user src, remip order by bandwidth desc

Dataset Name	Description	Log Category
Top-Dial-Up-IPSEC-Users-By- Duration	Top dial up IPsec users by duration	event
<pre>select coalesce(xauthuser_agg, user_agg, ipstr(`remip`)) as user_src, from_dtime(min(s_time)) as start_time, sum(duration) as duration, sum(duration) as duration, sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_out from (select devid, vd,</pre>	z	

```
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
      ###(select devid, vd, remip, nullifna(`xauthuser`) as xauthuser_agg, nullifna(`user`)
as user_agg, tunnelid, min(coalesce(dtime, 0)) as s_time, max(coalesce(dtime, 0)) as e_time,
max(coalesce(duration,0)) as max duration, min(coalesce(duration,0)) as min duration, min
(coalesce(sentbyte, 0)) as min traffic out, min(coalesce(rcvdbyte, 0)) as min traffic in,
max(coalesce(sentbyte, 0)) as max traffic out, max(coalesce(rcvdbyte, 0)) as max traffic in
```

from \$log where \$filter and subtype='vpn' and tunneltype like 'ipsec%' and not (tunnelip is null or tunnelip='0.0.0.0') and action in ('tunnel-stats', 'tunnel-down', 'tunnel-up') and tunnelid is not null and tunnelid!=0 group by devid, vd, remip, xauthuser_agg, user_agg, tunnelid order by tunnelid)### t group by devid, vd, remip, tunnelid) tt where bandwidth>0 group by user src order by duration desc

Dataset Name	Description	Log Category
Top-SSL-VPN-Web-Mode-Users-By- Bandwidth	Top SSL VPN web mode users by bandwidth usage	event
<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, vd, user_src, remip,) }</pre>	Ξ	

```
tunnelid,
     min(s_time) as s_time,
     max(e time) as e time,
      (
        case when min(s time) = max(e time) then max(max traffic in) + max(max traffic out)
else max(max traffic in) - min(min traffic in) + max(max traffic out) - min(min traffic out)
end
      ) as bandwidth,
      (
        case when min(s time) = max(e time) then max(max traffic in) else max(max traffic
in) - min(min traffic in) end
     ) as traffic in,
      (
        case when min(s time) = max(e time) then max(max traffic out) else max(max traffic
out) - min(min traffic out) end
     ) as traffic out
    from
      ###(select devid, vd, remip, coalesce(nullifna(`user`), ipstr(`remip`)) as user src,
tunnelid, tunneltype, max(coalesce(duration,0)) as max duration, min(coalesce(duration,0))
as min duration, min(coalesce(dtime, 0)) as s time, max(coalesce(dtime, 0)) as e time, min
(coalesce(sentbyte, 0)) as min_traffic_out, min(coalesce(rcvdbyte, 0)) as min_traffic_in,
max(coalesce(sentbyte, 0)) as max_traffic_out, max(coalesce(rcvdbyte, 0)) as max_traffic_in
from $log where $filter and subtype='vpn' and tunneltype like 'ssl%' and action in ('tunnel-
```

from Slog where Sfilter and subtype='vpn' and tunneltype like 'ssl%' and action in ('tunnelstats', 'tunnel-down', 'tunnel-up') and coalesce(nullifna(`user`), ipstr(`remip`)) is not null and tunnelid is not null group by devid, vd, user_src, remip, tunnelid, tunneltype)### t group by devid, vd, user_src, remip, tunnelid) tt where bandwidth>0 group by user_src, remote ip order by bandwidth desc

Dataset Name Description	Log Category
Top-SSL-VPN-Web-Mode-Users-By- Top SSL VPN web r Duration	node users by duration event
<pre>select user_src, remip as remote_ip, from_dtime(min(s_time)) as start_time, (max(e_time) - min(s_time)) as duration from (select devid, vd, user_src, remip, tunnelid, min(s_time) as s_time, max(e time) as e time</pre>	
<pre>from ###(select devid, vd, remip, coalesce(nulli tunnelid, tunneltype, max(coalesce(duration,0)) as min duration, min(coalesce(dtime, 0)) as s time </pre>	as max_duration, min(coalesce(duration,0))

(coalesce(sentbyte, 0)) as min_traffic_out, min(coalesce(rcvdbyte, 0)) as min_traffic_in, max(coalesce(sentbyte, 0)) as max_traffic_out, max(coalesce(rcvdbyte, 0)) as max_traffic_in from \$log where \$filter and subtype='vpn' and tunneltype like 'ssl%' and action in ('tunnelstats', 'tunnel-down', 'tunnel-up') and coalesce(nullifna(`user`), ipstr(`remip`)) is not null and tunnelid is not null group by devid, vd, user_src, remip, tunnelid, tunneltype)### t where tunneltype='ssl-web' group by devid, vd, user_src, remip, tunnelid) tt group by user_src, remote_ip order by duration desc

```
Dataset Name
                                  Description
                                                                                 Log Category
 Top-SSL-VPN-Users-By-Duration
                                  Top SSL VPN users by duration
                                                                                 event
select
  user src,
  tunneltype,
  sum(duration) as duration,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out
from
  (
    select
     devid.
      vd,
      remip,
      user src,
      tunneltype,
      tunnelid,
        case when min(s time) = max(e time) then max(max duration) else max(max duration) -
min(min duration) end
      ) as duration,
        case when min(s time) = max(e time) then max(max traffic in) else max(max traffic
in) - min(min traffic in) end
      ) as traffic in,
      (
        case when min(s_time) = max(e_time) then max(max_traffic_out) else max(max_traffic_
out) - min(min traffic out) end
      ) as traffic out,
      (
        case when min(s_time) = max(e_time) then max(max_traffic_in) + max(max_traffic_out)
else max(max traffic in) - min(min traffic in) + max(max traffic out) - min(min traffic out)
end
      ) as bandwidth
    from
      ###(select devid, vd, remip, coalesce(nullifna(`user`), ipstr(`remip`)) as user src,
tunnelid, tunneltype, max(coalesce(duration,0)) as max duration, min(coalesce(duration,0))
as min_duration, min(coalesce(dtime, 0)) as s_time, max(coalesce(dtime, 0)) as e_time, min
(coalesce(sentbyte, 0)) as min traffic out, min(coalesce(rcvdbyte, 0)) as min traffic in,
max(coalesce(sentbyte, 0)) as max traffic out, max(coalesce(rcvdbyte, 0)) as max traffic in
from $log where $filter and subtype='vpn' and tunneltype like 'ssl%' and action in ('tunnel-
stats', 'tunnel-down', 'tunnel-up') and coalesce(nullifna(`user`), ipstr(`remip`)) is not
null and tunnelid is not null group by devid, vd, user src, remip, tunnelid, tunneltype)###
t group by devid, vd, remip, user src, tunnelid, tunneltype) tt where bandwidth>0 group by
user src, tunneltype order by duration desc
```

Dataset Name	Description	Log Category
vpn-Top-Dial-Up-VPN-Users-By- Duration	Top dial up VPN users by duration	event
select		
coalesce(
<pre>xauthuser_agg, user_agg</pre>		
user_agg, ipstr(`remip`)		
) as user_src,		
t_type as tunneltype,		
from_dtime(
<pre>min(s_time)) as start time</pre>		
) as start_time, sum(duration) as duration,		
sum (bandwidth) as bandwidth,		
<pre>sum(traffic_in) as traffic_i</pre>		
<pre>sum(traffic_out) as traffic_</pre>	_out	
from		
(select		
devid,		
vd,		
remip,		
	thuser_agg, ' ') as xauthuser_agg,	
string_agg(distinct use)	r_agg, ' ') as user_agg,	
t_type, tunnelid,		
<pre>min(s_time) as s_time,</pre>		
max(e_time) as e_time,		
(
	= max(e_time) then max(max_duration) else	max(max_duration)-
nin(min_duration) end) as duration,		
(
case when min(s_time)=	= max(e_time) then max(max_traffic_in)+ m	<pre>wax(max_traffic_out)</pre>
else max(max_traffic_in) - min	(min_traffic_in)+ max(max_traffic_out)- m	in(min_traffic_out)
end		
) as bandwidth,		
case when min(s time)=	= max(e_time) then max(max_traffic_in) el	se max(max traffic
in) - min(min traffic in) end		
) as traffic_in,		
(
	<pre>= max(e_time) then max(max_traffic_out) e </pre>	lse max(max_traffic_
out)- min(min_traffic_out) end) as traffic out	1	
from		
	emip, nullifna(`xauthuser`) as xauthuser_	agg, nullifna(`user`)
as user_agg, (case when tunned	ltype like 'ipsec%' then 'ipsec' else tun	neltype end) as t_
time, max(coalesce(duration,0)	<pre>h(coalesce(dtime, 0)) as s_time, max(coal) as max_duration, min(coalesce(duration)</pre>	,0)) as min_duration,
	<pre>min_traffic_out, min(coalesce(rcvdbyte, max_traffic_out, max(coalesce(rcvdbyte,</pre>	

max(coalesce(sentbyte, 0)) as max_traffic_out, max(coalesce(rcvdbyte, 0)) as max_traffic_in, sum((case when action='tunnel-up' then 1 else 0 end)) as tunnelup from \$log where \$filter and subtype='vpn' and (tunneltype like 'ipsec%' or tunneltype like 'ssl%') and action in ('tunnel-up', 'tunnel-stats', 'tunnel-down') and tunnelid is not null and tunnelid!=0 group by xauthuser_agg, user_agg, devid, vd, remip, t_type, tunnelid, tunnelip)### t where (t_type like 'ssl%' or (t_type like 'ipsec%' and not (tunnelip is null or tunnelip='0.0.0.0'))) group by devid, vd, remip, t_type, tunnelid) tt where bandwidth>0 group by user_src, tunneltype order by duration desc

Dataset Name	Description	Log Category
vpn-User-Login-history	VPN user login history	event
select		
<pre>\$flex_timescale(timestamp</pre>) as hodex,	
sum(total_num) as total_n	um	
from		
(
select		
timestamp,		
devid,		
vd,		
remip,		
tunnelid,		
sum(tunnelup) as tota	l_num,	
<pre>max(traffic_in) as tr</pre>	affic_in,	
<pre>max(traffic_out) as t</pre>	raffic_out	
from		
—	stamp as timestamp, devid, vd,	
	lse 0 end) as tunnelup, max(coa	—

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
<pre>total_num from \$log where \$filte (tunneltype, 3)='ssl') and action</pre>	<pre>xauthuser`), `user`) as f_user, tunneltype, of er and subtype='vpn' and (tunneltype='ipsec' of on in ('ssl-login-fail', 'ipsec-login-fail') a (`user`)) is not null group by f_user, tunnel er by total_num desc</pre>	or left and coalesce

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

```
xauthuser_agg,
    user_agg,
    ipstr(`remip`)
  ) as f user,
  t type as tunneltype,
  from dtime(
  min(s time)
  ) as start time,
  sum(total num) as total num,
  sum(duration) as duration
from
  (
    select
      string_agg(distinct xauthuser_agg, ' ') as xauthuser_agg,
      string agg(distinct user agg, ' ') as user agg,
      t type,
      devid,
      vd,
      remip,
      tunnelid,
      min(s_time) as s_time,
      max(e time) as e time,
       case when min(s time) = max(e time) then max(max duration) else max(max duration)-
min(min duration) end
      ) as duration,
      (
        case when min(s time) = max(e time) then max(max traffic in) + max(max traffic out)
else max(max traffic in) - min(min traffic in) + max(max traffic out) - min(min traffic out)
end
      ) as bandwidth,
      (
        case when min(s time) = max(e time) then max(max traffic in) else max(max traffic
in) - min(min traffic in) end
      ) as traffic in,
      (
        case when min(s time) = max(e time) then max(max traffic out) else max(max traffic
out) - min(min traffic out) end
      ) as traffic out,
      sum(tunnelup) as total num
    from
      ###(select devid, vd, remip, nullifna(`xauthuser`) as xauthuser agg, nullifna(`user`)
as user_agg, (case when tunneltype like 'ipsec%' then 'ipsec' else tunneltype end) as t_
type, tunnelid, tunnelip, min(coalesce(dtime, 0)) as s_time, max(coalesce(dtime, 0)) as e_
time, max(coalesce(duration,0)) as max duration, min(coalesce(duration,0)) as min duration,
min(coalesce(sentbyte, 0)) as min traffic out, min(coalesce(rcvdbyte, 0)) as min traffic in,
```

min(coalesce(sentbyte, 0)) as min_traffic_out, min(coalesce(rcvdbyte, 0)) as min_traffic_in, max(coalesce(sentbyte, 0)) as max_traffic_out, max(coalesce(rcvdbyte, 0)) as max_traffic_in, sum((case when action='tunnel-up' then 1 else 0 end)) as tunnelup from \$log where \$filter and subtype='vpn' and (tunneltype like 'ipsec%' or tunneltype like 'ssl%') and action in ('tunnel-up', 'tunnel-stats', 'tunnel-down') and tunnelid is not null and tunnelid!=0 group by xauthuser_agg, user_agg, devid, vd, remip, t_type, tunnelid, tunnelip)### t group by t_ type, devid, vd, remip, tunnelid having max(tunnelup) > 0) tt where bandwidth>0 group by f_ user, tunneltype order by total_num desc

Dataset Name	Description	Log Category
vpn-Traffic-Usage-Trend-VPN- Summary	VPN traffic usage trend	event
<pre>else max(max_traffic_in) - min end) else 0 end) as ssl_traffic_bandwid (case when t_type like case when min(s_time else max(max_traffic_in) - min end) else 0 end) as ipsec_traffic_bandwid min(s_time) as s_time, max(e_time) as s_time, max(e_time) as e_time from ###(select \$flex_timesta tunneltype like 'ipsec%' then action='tunnel-up' then 1 elsa traffic_out, max(coalesce(rcva min_traffic_out, min(coalesce time, max(coalesce(dtime, 0)) (tunneltype like 'ipsec%' or s stats', 'tunnel-down') and tun vd, remip, t_type, tunnelid, a</pre>	<pre>) as ipsec_bandwidth mp) as hodex, 'ssl%' then (e)= max(e_time) then max(max_traffic_in)+ max (min_traffic_in)+ max(max_traffic_out) - min(m. dth, 'ipsec%' then (e)= max(e_time) then max(max_traffic_in)+ max (min_traffic_in)+ max(max_traffic_out) - min(m.</pre>	<pre>in_traffic_out) (max_traffic_out) in_traffic_out) in_traffic_out) (case when e when 0)) as max_ ntbyte, 0)) as e(dtime, 0)) as s_ e='vpn' and l-up','tunnel- imestamp, devid, SkipEND*/)### t</pre>
Dataset Name	Description	Log Category
Top-S2S-IPSEC-Tunnels-By- Bandwidth-and-Availability	Top S2S IPsec tunnels by bandwidth usage and avail	event
<pre>select vpntunnel, tunneltype, sum(traffic_out) as traffic_sum(traffic_in) as traffic_sum(bandwidth) as bandwidth)</pre>	in,	

```
sum(uptime) as uptime
from
  (
    select
     vpntunnel,
     tunneltype,
     tunnelid,
     devid,
     vd,
     sum(sent end - sent beg) as traffic out,
     sum(rcvd end - rcvd beg) as traffic in,
     sum(
       sent end - sent beg + rcvd end - rcvd beg
      ) as bandwidth,
      sum(duration end - duration beg) as uptime
    from
```

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

Dataset Name Description Log Category

Top-Dialup-IPSEC-By-Bandwidth-and-Top dialup IPsec users by bandwidth usage and availeventAvailability

```
select
 user src,
 remip,
 sum(traffic out) as traffic out,
 sum(traffic in) as traffic in,
 sum(bandwidth) as bandwidth,
  sum(uptime) as uptime
from
  (
   select
     user src,
     remip,
     tunnelid,
     devid,
     vd.
      sum(sent end - sent beg) as traffic out,
     sum(rcvd end - rcvd beg) as traffic in,
     sum(
       sent end - sent beg + rcvd end - rcvd beg
      ) as bandwidth,
      sum(duration end - duration beg) as uptime
    from
      ###(select tunnelid, coalesce(nullifna(`xauthuser`), nullifna(`user`), ipstr(`remip`))
as user_src, remip, devid, vd, min(coalesce(sentbyte, 0)) as sent_beg, max(coalesce
(sentbyte, 0)) as sent_end, min(coalesce(rcvdbyte, 0)) as rcvd_beg, max(coalesce(rcvdbyte,
```

0)) as rcvd_end, min(coalesce(duration, 0)) as duration_beg, max(coalesce(duration, 0)) as duration_end from \$log where \$filter and subtype='vpn' and action='tunnel-stats' and tunneltype like 'ipsec%' and not (tunnelip is null or tunnelip='0.0.0.0') and tunnelid is not null and tunnelid!=0 group by tunnelid, user_src, remip, devid, vd /*SkipSTART*/order by tunnelid/*SkipEND*/)### t group by user_src, remip, tunnelid, devid, vd order by bandwidth desc) t where bandwidth>0 group by user_src, remip order by bandwidth desc

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

```
select
 user src,
 remote ip,
 sum(traffic out) as traffic out,
 sum(traffic in) as traffic in,
 sum (bandwidth) as bandwidth,
 sum(uptime) as uptime
from
  (
   select
     user src,
     remip as remote ip,
     tunnelid,
     devid,
     vd,
      sum(sent_end - sent_beg) as traffic_out,
     sum(rcvd end - rcvd beg) as traffic in,
     sum(
       sent end - sent beg + rcvd_end - rcvd_beg
      ) as bandwidth,
      sum(duration end - duration beg) as uptime
    from
```

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

Dataset Name	Description	Log Category
Top-SSL-Web-Mode-By-Bandwidth- and-Availability	Top SSL web users by bandwidth usage and avail	event
<pre>select user_src, remote_ip, sum(traffic_out) as traffic_ou sum(traffic_in) as traffic_in, sum(bandwidth) as bandwidth, sum(uptime) as uptime</pre>		

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

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

Dataset Name	Description	Log Category
Admin-Login-Summary	Event admin login summary	event
<pre>select f_user, ui, sum(login) as total_num, sum(login_duration) as tot sum(config_change) as total from (select `user` as f_user,</pre>	tal_duration,	UVUIL
) as login, (case when logid_to_:	int(logid)= 32001 then 1 else 0 end int(logid)= 32003 then duration else 0 end	
) as login_duration, (case when logid_to_ and state is not nu) as config_change from	int(logid)= 32003 ll then 1 else 0 end	
<pre>\$log where \$filter and nullifna(`user`) and logid_to_int(logic</pre>		

```
) t
group by
 f_user,
 ui
having
 sum(login) + sum(config change)& gt; 0
order by
 total num desc
```

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

```
$flex_timescale(timestamp) as dom,
 sum(total_num) as total_num,
 sum(total change) as total change
from
```

###(select timestamp, sum(login) as total_num, sum(config_change) as total_change from (select \$flex_timestamp as timestamp, (case when logid_to_int(logid)=32001 then 1 else 0 end) as login, (case when logid to int(logid)=32003 and state is not null then 1 else 0 end) as 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
<pre>select `user` as f_user, ui, count(status) as total_faile from \$log where \$filter and nullifna(`user`) is not and logid_to_int(logid) = 3: group by ui, f_user order by total_failed desc</pre>	null	
Dataset Name	Description	Log Category
System-Summary-By-Severity	Event system summary by severity	event
	a(logdesc), msg) as msg_desc, (case when ritical' when level='error' then 'High' w	

```
then 'Medium' when level='notice' then 'Low' else 'Info' end) as severity tmp, count(*) as
```

count from \$log where \$filter and subtype='system' group by msg_desc, severity_tmp
/*SkipSTART*/order by count desc/*SkipEND*/)### t group by severity order by total_num desc

Dataset Name	Description	Log Category
System-Summary-By-Date	Event system summary by date	event
<pre>select \$flex_timescale(timestamp) sum(critical) as critical,</pre>	as dom,	

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

```
from
```

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

Dataset Name	Description	Log Category
Important-System-Summary-By-Date	Event system summary by date	event
<pre>select \$flex_timescale(timestamp) as o sum(critical) as critical, sum(high) as high, sum(medium) as medium</pre>	dom,	
'emergency') then 1 else 0 end) a end) as high, sum(case when level	timestamp, sum(case when level in ('critical', as critical, sum(case when level = 'error' the l = 'warning' then 1 else 0 end) as medium fro up by timestamp /*SkipSTART*/order by timestam om order by dom	en 1 else 0 om \$log where

Dataset Name	Description	Log Category
System-Critical-Severity-Events	Event system critical severity events	event
'alert', 'emergency') then 'Crit then 'Medium' when level='notice count from \$log where \$filter and	ogdesc), msg) as msg_desc, (case when level in ical' when level='error' then 'High' when leve ' then 'Low' else 'Info' end) as severity_tmp, d subtype='system' group by msg_desc, severity /*SkipEND*/)### t where severity_tmp='Critical s desc	el='warning' count(*) as y_tmp

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

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

```
Dataset Name
                                  Description
                                                                                  Log Category
 System-Medium-Severity-Events
                                                                                  event
                                  Event system medium severity events
select
 msg desc as msg,
 severity_tmp as severity,
 sum(count) as counts
from
 ####(select coalesce(nullifna(logdesc), msg) as msg desc, (case when level in ('critical',
'alert', 'emergency') then 'Critical' when level='error' then 'High' when level='warning'
then 'Medium' when level='notice' then 'Low' else 'Info' end) as severity tmp, count(*) as
count from $log where $filter and subtype='system' group by msg_desc, severity_tmp
/*SkipSTART*/order by count desc/*SkipEND*/)### t where severity tmp='Medium' group by msg,
```

severity tmp order by counts desc

Dataset Name	Description	Log Category
utm-drilldown-Top-Traffic-Summary	UTM drilldown traffic summary	traffic
<pre>srcip, srcname, sum(coalesce(se</pre>	`user`), nullifna(`unauthuser`), ips ntbyte, 0)+coalesce(rcvdbyte, 0)) as group by user_src, srcip, srcname c own group by srcip, srcname	bandwidth from \$log
Dataset Name	Description	Log Category
utm-drilldown-Top-User-Destination	UTM drilldown top user destination	traffic
appid, app, dstip, count(*) as	`user`), nullifna(`unauthuser`), ips sessions, sum(coalesce(sentbyte, 0)+ er and (logflag&1>0) and dstip is nc	coalesce(rcvdbyte, 0)) as

is not null group by user src, appid, app, dstip having sum(coalesce(sentbyte, 0)+coalesce

(rcvdbyte, 0))>0 order by bandwidth desc)### t where \$filter-drilldown group by appid, app,
dstip order by bandwidth desc

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

select

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

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

Dataset Name	Description	Log Category		
utm-drilldown-Email-Receivers- Summary	UTM drilldown email receivers summary	traffic		
<pre>select sum(requests) as requests, sum(bandwidth) as bandwidth</pre>				
from				
<pre>###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,</pre>				
<pre>recipient, count(*) as requests, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from \$log where \$filter and (logflag&1>0) and recipient is not null and service in ('pop3', 'POP3', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/tcp', 'pop3s', 'POP3S', '995/tcp') group by user_src, recipient order by requests desc)### t where \$filter- drilldown</pre>				

Dataset Name	Description	Log Category		
utm-drilldown-Top-Email-Recipients- By-Bandwidth	UTM drilldown top email recipients	traffic		
<pre>select recipient, sum(bandwidth) as bandwidth from</pre>	· · · · · · · · · · · · · · · · · · ·			
<pre>###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src recipient, count(*) as requests, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from \$log where \$filter and (logflag&1>0) and recipient is not null and service in</pre>				
('pop3', 'POP3', '110/tcp', 'ima 'POP3S', '995/tcp') group by use	<pre>p', 'IMAP', '143/tcp', 'imaps', 'IMAPS', r_src, recipient order by requests desc) ing sum(bandwidth)>0 order by bandwidth</pre>	'993/tcp', 'pop3s', ### t where \$filter-		

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

select
sender,
sum(bandwidth) as bandwidth
from
####(select_coalesce(nullifna(``

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

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

select

```
appid,
hostname,
sum(bandwidth) as bandwidth
from
```

###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, appid, hostname, (case when utmaction in ('block', 'blocked') then 1 else 0 end) as blocked, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from \$log-traffic where \$filter and (logflag&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', 'commandblock', 'script-filter'))) and hostname is not null group by user_src, appid, hostname, blocked order by bandwidth desc)### t where \$filter-drilldown and blocked=0 group by appid, hostname order by bandwidth desc

Dataset Name	Description	Log Category
utm-drilldown-Top-Blocked-Websites- By-Request	UTM drilldown top blocked web sites by re	equest webfilter
(case when action='blocked' the where \$filter and (eventtype is	(`user`), ipstr(`srcip`)) as user_s en 1 else 0 end) as blocked, count(s null or logver>=502000000) and ho cked order by requests desc)### t w name order by requests desc	(*) as requests from \$log ostname is not null group by
Dataset Name	Description	Log Category
utm-drilldown-Top-Virus-By-Name	UTM drilldown top virus	virus
select		

(virus) is not null group by user_src, virus order by totalnum desc)### t where \$filterdrilldown group by virus order by totalnum desc

Dataset Name	Description	Log Category
utm-drilldown-Top-Attacks	UTM drilldown top attacks by name	attack
select attack,		
sum(attack_count) as atta	ck_count	

from

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

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

select

vuln, sum(totalnum) as totalnum from

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

Dataset Name	Description	Log Category
utm-drilldown-Top-App-By-Bandwidth	UTM drilldown top applications by bandwidth usage	traffic
<pre>nullifna(`unauthuser`), ipstr(`s (coalesce(sentbyte, 0)+coalesce(where \$filter and (logflag&1>0) src, appid, app, appcat, apprisk</pre>	<pre>app*/select devid, vd, csf, coalesce(nullifn srcip`)) as user_src, appid, app, appcat, ap frcvdbyte, 0)) as bandwidth, count(*) as ses and nullifna(app) is not null group by devi- c order by sessions desc)base### t where \$fi bandwidth)>0 order by bandwidth desc</pre>	prisk, sum sions from \$log d, vd, csf, user_
Dataset Name	Description	Log Category
utm-drilldown-Top-App-By-Sessions	UTM drilldown top applications by session count	traffic
	app*/select devid, vd, csf, coalesce(nullifn srcip`)) as user_src, appid, app, appcat, ap	

(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, count(*) as sessions from \$log
where \$filter and (logflag&1>0) and nullifna(app) is not null group by devid, vd, csf, user_
src, appid, app, appcat, apprisk order by sessions desc)base### t where \$filter-drilldown
group by appid, app order by sessions desc

Dataset Name	Description	Log Category
Top5-Users-By-Bandwidth	UTM drilldown top users by bandwidth usage	traffic
<pre>) as bandwidth, sum(coalesce(sentbyte,) as traffic_out, sum(coalesce(rcvdbyte,) as traffic_in from \$log where \$filter and (logflag&1>0)</pre>	0)+ coalesce(rcvdbyte, 0) 0)	
group by dldn_user		
<pre>having sum(coalesce(sentbyte,</pre>	0)+ coalesce(rcvdbyte, 0)	
)& gt; 0 order by	-	
bandwidth desc		

Dataset Name	Description	Log Category traffic	
bandwidth-app-Top-App-By- Bandwidth-Sessions	Top applications by bandwidth usage		
<pre>select app_group_name(app) as app sum(coalesce(sentbyte, 0)+ c) as bandwidth, sum(coalesce(rcvdbyte, 0)</pre>			

```
FortiAnalyzer 6.2.8 Dataset Reference
```

coalesce(sentbyte, 0)

) as traffic in,

sum(

```
) as traffic out,
  count(*) as sessions
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
  and nullifna(app) is not null
group by
  app group
having
  sum(
    coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
  )& gt; 0
order by
  bandwidth desc
```

Dataset Name

bandwidth-app-Category-By-
BandwidthApplication risk application usage by categorytrafficselect
appcat,
sum(bandwidth) as bandwidthsound width)sound width)trafficfrom
###base(/*tag:rpt_base_t_top_app*/select devid, vd, csf, coalesce(nullifna(`user`),
nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, appid, app, appcat, apprisk, sum
(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, count(*) as sessions from \$log
where \$filter and (logflag&1>0) and nullifna(app) is not null group by devid, vd, csf, user_
src, appid, app, appcat, apprisk order by sessions desc)base### t where nullifna(appcat) is
not null group by appcat order by bandwidth desc

Description

Dataset Name		Description	Log Category
bandwidth-app-Top-Users-E Bandwidth-Sessions	Зу-	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)+ coale: 0)	sce(rcvdbyte, 0)	

Log Category

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

Dataset Name	Description	Log Category
bandwidth-app-Traffic-By-Active-User- Number	Bandwidth application traffic by active user number	traffic
<pre>select \$flex_timescale(timestamp) as count(distinct(user_src)) as total_user</pre>	hodex,	
(`unauthuser`), ipstr(`srcip`))	timestamp, coalesce(nullifna(`user`), nullifna as user_src from \$log where \$filter and (logf	lag&1>0) group

by timestamp,	user_src	order b	y timesta	mp desc)###	t	group	by	hodex	order	by	hodex
Dataset Name			Descri	otion							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)+ coal) as bandwidth, sum(coalesce(rcvdbyte, 0)+ coal) as traffic_in, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out, count(*) as sessions from \$log where</pre>	lesce(rcvdbyte, 0)	

\$filter
and (
 logflag&1>0
)
group by
domain
order by
bandwidth desc

Dataset Name

bandwidth-app-Top-Policies-By-Bandwidth-Sessions

select

```
coalesce(
   pol.name,
   cast(policyid as text)
) as polid,
sum(bandwidth) as bandwidth,
sum(traffic_in) as traffic_in,
sum(traffic_out) as traffic_out,
sum(sessions) as sessions
```

from

###(select policyid, poluuid, sum(coalesce(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

Top policies by bandwidth and sessions

Description

Dataset Name	Description	Log Category
bandwidth-app-Traffic-Statistics	Bandwidth application traffic statistics	traffic
<pre>drop table if exists rpt_tmptbl_3 total_sessions varchar(25 total_bandwidth varchar(25 ave_session varchar(255), ave_bandwidth varchar(255), active_date varchar(255), total_users varchar(255), total_dest varchar(255), total_dest varchar(255)); insert into rpt_tmptbl_1 total_sessions, total_band ave_session, ave_bandwidth) select format_numeric_no_decimal(sum(sessions)) as total_sessions, bandwidth_unit(sum(bandwidth)) as total_bandwidth, format_numeric_no_decimal(</pre>	55),), (dwidth,	

Log Category

traffic

```
cast(
    sum(sessions)/ $days_num as decimal(18, 0)
  )
) as ave session,
bandwidth unit(
  cast(
   sum(bandwidth) / $days num as decimal(18, 0)
  )
) as ave bandwidth
```

from

```
####(select count(*) as sessions, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as
bandwidth from $log where $filter and (logflag&1>0))### t; update rpt tmptbl 1 set active
date=t1.dom from (select dom, sum(sessions) as sessions from ###(select $DAY OF MONTH as
dom, count(*) as sessions from $log where $filter and (logflag&1>0) group by dom order by
sessions desc)### t group by dom order by sessions desc limit 1) as t1; update rpt tmptbl 1
set total users=t2.totalnum from (select format numeric no decimal(count(distinct(user
src))) as totalnum from ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr
(`srcip`)) as user src, count(*) as count from $log where $filter and (logflag&1>0) group by
user src order by count desc)### t) as t2; update rpt tmptbl 1 set total app=t3.totalnum
from (select format numeric no decimal(count(distinct(app grp))) as totalnum from ###(select
app_group_name(app) as app_grp, count(*) as count from $log where $filter and (logflag&1>0)
and nullifna(app) is not null group by app_grp order by count desc)### t) as t3; update rpt_
tmptbl 1 set total dest=t4.totalnum from (select format numeric no decimal(count(distinct
(dstip))) as totalnum from ###(select dstip, count(*) as count from $log where $filter and
(logflag&1>0) and dstip is not null group by dstip order by count desc)### t ) as t4; select
'Total Sessions' as summary, total sessions as stats from rpt tmptbl 1 union all select
'Total Bytes Transferred' as summary, total bandwidth as stats from rpt tmptbl 1 union all
select 'Most Active Date By Sessions' as summary, active_date as stats from rpt_tmptbl_1
union all select 'Total Users' as summary, total users as stats from rpt tmptbl 1 union all
select 'Total Applications' as summary, total_app as stats from rpt_tmptbl_1 union all
select 'Total Destinations' as summary, total dest as stats from rpt tmptbl 1 union all
select 'Average Sessions Per Day' as summary, ave_session as stats from rpt_tmptbl_1 union
all select 'Average Bytes Per Day' as summary, ave bandwidth as stats from rpt tmptbl 1
```

Dataset Name	Description	Log Category
Score-Summary-For-All-Users- Devices	Reputation score summary for all users devices	traffic
<pre>select \$flex_timescale(timestamp) as hodex, sum(scores) as scores from ###(select \$flex_timestamp as timestamp, sum(crscore%65536) as s totalnum from \$log where \$filter and (logflag&1>0) and crscore is timestamp having sum(crscore%65536)>0 order by timestamp desc)### hodex</pre>		L group by
Dataset Name	Description	Log Category

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

select

```
$flex timescale(timestamp) as hodex,
sum(scores) as scores,
```

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

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

Dataset Name	Description	Log Category
Top-Users-By-Reputation-Scores	Reputation top users by scores	traffic
<pre>select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, sum(crscore % 65536) as scores from \$log where \$filter and (logflag&1>0) and crscore is not null group by user_src having sum(crscore % 65536)& gt; 0 order by scores desc</pre>		

Dataset	Name
Dutuoot	

Ton-Devices-B	y-Reputation-Score	20
TOP-Devices-Dy	y-nepulation-Score	25

Reputation top devices by scores

Description

traffic

Log Category

```
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

drop

table if exists rpt_tmptbl_1;

drop

table if exists rpt_tmptbl_2; create temporary table rpt_tmptbl_1 as ###(select coalesce (nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as f_user, sum(crscore%65536) as sum_rp_score from \$log where \$pre_period \$filter and (logflag&l>0) and crscore is not null group by f_user having sum(crscore%65536)>0 order by sum_rp_score desc)###; create temporary table rpt_tmptbl_2 as ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr (`srcip`)) as f_user, sum(crscore%65536) as sum_rp_score from \$log where \$filter and (logflag&l>0) and crscore is not null group by f_user having sum(crscore%65536)>0 order by sum_rp_score desc)###; select t1.f_user, sum(t1.sum_rp_score) as t1_sum_score, sum(t2.sum_ rp_score) as t2_sum_score, (sum(t2.sum_rp_score)-sum(t1.sum_rp_score)) as delta from rpt_ tmptbl_1 as t1 inner join rpt_tmptbl_2 as t2 on t1.f_user=t2.f_user where t2.sum_rp_score > t1.sum_rp_score group by t1.f_user order by delta desc

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

drop

table if exists rpt_tmptbl_1;

drop

table if exists rpt_tmptbl_2; create temporary table rpt_tmptbl_1 as ###(select coalesce (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&l>0) and crscore is not null group by f_device, devtype_new having sum(crscore%65536)>0 order by sum_rp_score desc)###; create temporary table rpt_tmptbl_2 as ###(select coalesce(nullifna(`srcname`),nullifna(`srcmac`), ipstr (`srcip`)) as f_device, get_devtype(srcswversion, osname, devtype) as devtype_new, sum (crscore%65536) as sum_rp_score from \$log where \$filter and (logflag&l>0) and crscore is not null group by f_device, devtype_new having sum(crscore%65536)>0 order by sum_rp_score desc)###; 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	

```
count(*) as totalnum
from
   $log
where
   $filter
group by
   severity
order by
   totalnum desc
```

sum(attack_count) as attack_count

Dataset Name	Description	Log Category		
Top-Attacks-Detected	Threat top attacks detected	attack		
select				
attack, attackid,				
cve, severity,				

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

Dataset Name	Description	Log Category
Top-Attacks-Blocked	Threat top attacks blocked	attack
<pre>select attack, count(*) as attack_cour from \$log where \$filter and nullifna(attack) is and action not in ('det group by attack order by attack_count desc</pre>		
Dataset Name	Description	Log Category
Top-Virus-Source	Threat top virus source	virus
<pre>select srcip, hostname, sum(totalnum) as totalr from</pre>	num	

###(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
	as hodex, as timestamp, count(*) as totalnum T*/order by timestamp desc/*SkipE	-

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

select

\$flex datetime(timestamp) as hodex,

sum(totalnum) as totalnum

from

###(select \$flex_timestamp as timestamp, count(*) as totalnum from \$log where \$filter and (eventtype is null or logver>=50200000) and nullifna(virus) is not null group by timestamp /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t group by hodex order by hodex

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

select

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

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

Dataset Name	Description Lo	
Top-Spyware-by-Name	Threat top spyware by name	virus
	(`user`), ipstr(`srcip`)) as user_sr A_s, count(*) as totalnum from \$log	
user_src, virus, virusid_s /*Sk like 'Riskware%' group by virus	tipSTART*/order by totalnum desc/*Sk s order by totalnum desc	<pre>sipEND*/)### t where virus</pre>

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
	as timestamp, count(*) as totalnum fro by timestamp /*SkipSTART*/order by timestamp	
Dataset Name	Description	Log Category
Top-Adware-Victims	Threat top adware victims	virus
select user_src, sum(totalnum) as totalnum from	na(`user`), ipstr(`srcip`)) as user_sr	
totalnum from \$log where \$fi	lter group by user_src, virus /*SkipST. virus like 'Adware%' group by user_src	
otalnum from \$log where \$fi	lter group by user_src, virus /*SkipST.	

```
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 group by user_src, virus, virusid_s /*SkipSTART*/order by totalnum desc/*SkipEND*/)### t where virus like 'Adware%' group by virus order by totalnum desc

Dataset Name	Description	Log Category
Top-Adware-Source	Threat top adware source	traffic
<pre>select srcip, hostname, count(*) as totalnum from \$log where \$filter and (logflag&1>0) and virus like 'Adware%' group by srcip,</pre>		
hostname order by totalnum desc		

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

select

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

from

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

Dataset Name	Description Log C		
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</pre>	timescale,		
<pre>### (select \$flex_timestamp as timestamp, sum(case when severity = 'critical' then 1 else 0 end) as critical, sum(case when severity = 'high' then 1 else 0 end) as high, sum(case when severity = 'medium' then 1 else 0 end) as medium, sum(case when severity = 'notice' then 1 else 0 end) as low, sum(case when severity = 'info' or severity = 'debug' then 1 else 0 end) as info from \$log where \$filter group by timestamp /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t group by timescale order by timescale</pre>			

Dataset Name	Description	
Important-Intrusions-Timeline-By- Severity	Threat intrusions timeline by severity	attack
<pre>select \$flex_timescale(timestamp) a sum(critical) as critical, sum(high) as high, sum(medium) as medium, sum(low) as low, sum(info) as info</pre>	s timescale,	
end) as critical, sum(case when	s timestamp, sum(case when severity = 'c n severity = 'high' then 1 else 0 end) a	as high, sum(case when

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

Dataset Name	Description	Log Category
Top-Intrusions-By-Types	Threat top intrusions by types	attack
select		
vuln_type,		
<pre>count(*) as totalnum from</pre>		
\$log t1		
left join (
select		
name,		
cve,		
vuln_type		
from inc. mdata		
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.		

```
cve,
vuln_type,
count(*) as totalnum
from
```

```
$log t1
  left join (
    select
     name,
     cve,
     vuln_type
    from
     ips mdata
 ) t2 on t1.attack = t2.name
where
 $filter
 and tl.severity = 'critical'
 and nullifna(attack) is not null
group by
 attack,
 attackid,
 cve,
 vuln_type
order by
  totalnum desc
```

Dataset Name	Description	Log Category
High-Severity-Intrusions	Threat high severity intrusions	attack
select		
attack,		
attackid,		
vuln_type,		
cve,		
count(*) as totalnum		
from		
\$log t1		
left join (
select		
name,		
cve,		
vuln_type		
from		
ips_mdata		
) t2 on t1.attack = t2.name		
where		
\$filter		
and tl.severity = 'high'		
and nullifna(attack) is not n	ull	
group by		
attack,		
attackid,		
vuln_type, cve		
order by		
totalnum desc		
Cotarium 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 null
group by
 attack,
 vuln_type,
 cve
order by
 totalnum desc
```

Dataset Name	Description Log C	
Top-Intrusion-Victims	Threat top intrusion victims	attack
<pre>num, sum(case when severity severity='medium' then 1 el</pre>	med_num) as totalnum m, sum((case when severity='critical' ='high' then 1 else 0 end) as high_num se 0 end) as med_num from \$log where \$ m') group by victim)### t group by vic	, sum(case when filter and severity in
Dataset Name	Description	Log Category
Top-Intrusion-Sources Threat top intrusion sources		attack

```
select
source,
sum(cri_num) as critical,
sum(high_num) as high,
sum(med_num) as medium,
sum(cri_num + high_num + med_num) as totalnum
from
####(select srcip as source, sum(case when severity='critical' then 1 else 0 end) as cri_
num, sum(case when severity='high' then 1 else 0 end) as high_num, sum(case when
```

severity='medium' then 1 else 0 end) as med_num from \$log where \$filter and severity in
('critical', 'high', 'medium') group by source)### t group by source order by totalnum desc

Dataset Name	aset Name Description Log Cate			
Top-Blocked-Intrusions	d-Intrusions Threat top blocked intrusions attack			
select				
attack,				
attackid,				
(
<pre>when t1.severity = 'medium t1.severity = 'info' then) as severity_name, count(*) as totalnum, vuln_type, (</pre>		hen 'Low' when		
	= 'critical' then 0 when t1.severity = 'hi			
	n 2 when t1.severity = 'low' then 3 when t	1.severity = 'info' ther		
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.n.	ane			
where \$filter				
and nullifna(attack) is a	not null			
and action not in ('deter				
group by	, pass_session ,			
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,				
(
when tl.severity = 'medium	<pre>= 'critical' then 'Critical' when t1.sever ' then 'Medium' when t1.severity = 'low' t 'Infe! and</pre>			
<pre>t1.severity = 'info' then</pre>	THEO END			

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

_			
Da	+	a t	ma
	105		

Dataset Name	Description	Log Category
Attacks-Over-HTTP-HTTPs	Threat attacks over HTTP HTTPs	attack
<pre>severity = 'medium' then 'M then 'Info' end) as severity, count(*) as totalnum, (case when severity = 'count'</pre>	critical' then 'Critical' when severity = Medium' when severity = 'low' then 'Low' w critical' then 0 when severity = 'high' th ty = 'low' then 3 when severity = 'info'	when severity = 'info' hen 1 when severity =
<pre>\$filter and severity in ('critica and upper(service) in ('F</pre>		
group by attack,		

```
attackid,
severity,
severity_number
order by
severity_number,
totalnum desc
```

Dataset Name	Description	Log Category
default-AP-Detection-Summary-by- Status-OffWire	Default access point detection summary by status off- wire	event
<pre>'others' end) as ap_full_status, count(*) as totalnum from (select apstatus, bssid, ssid from ###(select apstatus, bssid apstatus is not null and apstatus (logid) in (43527, 43521, 43525 43583, 43584, 43585) group by applications of the second secon</pre>	d, ssid, count(*) as subtotal from \$log where us!=0 and bssid is not null and onwire='no' an , 43563, 43564, 43565, 43566, 43569, 43570, 43 pstatus, bssid, ssid order by subtotal desc)## by ap_full_status order by totalnum desc	\$filter and d logid_to_int 571, 43582,
Dataset Name	Description	Log Category
default-AP-Detection-Summary-by- Status-OffWire_table	Default access point detection summary by status off- wire	event

```
select
  (
    case apstatus when 1 then 'rogue' when 2 then 'accepted' when 3 then 'suppressed' else
'others' end
  ) as ap_full_status,
    count(*) as totalnum
```

```
from (
```

select apstatus, bssid,

```
ssid
from
```

###(select apstatus, bssid, ssid, count(*) as subtotal from \$log where \$filter and apstatus is not null and apstatus!=0 and bssid is not null and onwire='no' and logid_to_int (logid) in (43527, 43521, 43525, 43563, 43564, 43565, 43566, 43569, 43570, 43571, 43582, 43583, 43584, 43585) group by apstatus, bssid, ssid order by subtotal desc)### t group by apstatus, bssid, ssid) t group by ap_full_status order by totalnum desc

Dataset Name	Description	Log Category
default-AP-Detection-Summary-by-	Default access point detection summary by status on-	event
Status-OnWire	wire	
select		
(
case apstatus when 1 then '	roque' when 2 then 'accepted' when 3 then 'supp	pressed' else
'others' end	5	
) as ap_full_status,		
count(*) as totalnum		
from		
(
select		
apstatus,		
bssid,		
ssid		
from		
· · · ·	d, ssid, count(*) as subtotal from \$log where \$	
	us!=0 and bssid is not null and onwire='yes' and a state and a state and a state a sta	

apstatus is not null and apstatus!=0 and bssid is not null and onwire='yes' and logid_to_int (logid) in (43527, 43521, 43525, 43563, 43564, 43565, 43566, 43569, 43570, 43571, 43582, 43583, 43584, 43585) group by apstatus, bssid, ssid order by subtotal desc)### t group by apstatus, bssid, ssid) t group by ap_full_status order by totalnum desc

Dataset Name	Description	Log Category
default-AP-Detection-Summary-by- Status-OnWire_table	Default access point detection summary by status on- wire	event
<pre>'others' end) as ap_full_status, count(*) as totalnum from (select apstatus, bssid, ssid from ###(select apstatus, bssid apstatus is not null and apstatus (logid) in (43527, 43521, 43525 43583, 43584, 43585) group by applications of the second sec</pre>	<pre>rogue' when 2 then 'accepted' when 3 then 'supp d, ssid, count(*) as subtotal from \$log where 3 us!=0 and bssid is not null and onwire='yes' ar , 43563, 43564, 43565, 43566, 43569, 43570, 433 pstatus, bssid, ssid order by subtotal desc)### by ap_full_status order by totalnum desc</pre>	filter and nd logid_to_int 571, 43582,

```
    Dataset Name
    Description
    Log Category

    default-Managed-AP-Summary
    Default managed access point summary
    event

    select
    (
    (

    (
    case when (
    *join%'
```

```
and logid_to_int(logid) in (43522, 43551)
) then 'Authorized' else 'Unauthorized' end
) as ap_status,
count(*) as totalnum
from
   $log
where
   $filter
   and logid_to_int(logid) in (43522, 43551)
group by
   ap_status
order by
   totalnum desc
```

Dataset Name	Description	Log Category
default-Managed-AP-Summary_table	Default managed access point summary	event
<pre>select (case when (action like '%join%' and logid_to_int(logid) in</pre>	(43522, 43551)	

```
) then 'Authorized' else 'Unauthorized' end
) as ap_status,
  count(*) as totalnum
from
  $log
where
  $filter
  and logid_to_int(logid) in (43522, 43551)
group by
  ap_status
order by
  totalnum desc
```

Dataset Name	Description	Log Category
default-Unclassified-AP-Summary	Default unclassified access point summary	event
<pre>) as ap_status, count(*) as totalnum from ###(select onwire, ssid, bss apstatus=0 and bssid is not nu 43564, 43565, 43566, 43569, 43</pre>	n 'off-wire' when 'yes' then 'on-wire' els sid, count(*) as subtotal from \$log where all and logid_to_int(logid) in (43521, 435 8570, 43571, 43582, 43583, 43584, 43585) g ### t group by ap_status order by totalnum	\$filter and 25, 43527, 43563, roup by onwire, ssid,
Dataset Name	Description	Log Category

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

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

bssid order by subtotal desc) ### t group by ap_status order by totalnum desc

Dataset Name	Description	Log Category
default-selected-AP-Details-OffWire	Default selected access point details off-wire	event
<pre>3 then 'suppressed' else 'others) as ap_full_status, devid, vd, ssid, bssid, manuf, rssi, channel, radioband, from_dtime(</pre>	nclassified' when 1 then 'rogue' when 2 th	hen 'accepted' when
<pre>min(dtime)) as first_seen, from_dtime(max(dtime)) as last_seen, detectionmethod, itime, onwire as on_wire from \$log</pre>		
<pre>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, 43570, 43571</pre>	43566, 43569,	
) 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
<pre>3 then 'suppressed' else 'others) as ap_full_status,</pre>	nclassified' when 1 then 'rogue' when 2 the ' end	n 'accepted' when
devid, vd, ssid, bssid, manuf, rssi,		
<pre>channel, radioband, from_dtime(min(dtime)) as first_seen,</pre>		
<pre>from_dtime(max(dtime)) as last_seen, detectionmethod, itime,</pre>		
onwire as on_wire from \$log		
<pre>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,</pre>	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
<pre>drop table if exists rpt_tmptbl_1; select ip, lower(mac) as lmac, sn, ssid, channel, radioband, min(dtime) as first,</pre>	create temporary table rpt_tmptbl_1 as	
<pre>max(dtime) as last from \$log - event where \$filter and ip is not null and mac is not null and sn is not null</pre>		
<pre>and ssid is not null group by ip, lmac, sn, ssid, channel,</pre>		
<pre>radioband order by ip; select user_src, ip, lmac, sn,</pre>		
<pre>ssid, channel, radioband, from_dtime(first) as first_se from_dtime(last) as last_seer cast(volume as decimal(18, 2)) as bandwidth</pre>		
<pre>from (select * from rpt_tmptbl_1 inner join (select</pre>		

```
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 \$logtraffic 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
<pre>select 'accepted' as ap_full_status,</pre>		
devid, vd,		
ssid, bssid,		
manuf, channel, radioband,		
from_dtime(max(last seen)		
) as last_seen, detectionmethod,		
snclosest, 'no' as on wire		
from -	acid manuf abannal radioband dat	actionmethod anglesset
onwire, logid, apstatus, max(dt	<pre>ssid, manuf, channel, radioband, det ime) as last_seen from \$log where \$f (42521 42525 42562 42564 42565</pre>	ilter and bssid is not

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

Dataset Name	Description	Log Category
event-Wireless-Accepted-Onwire	Event wireless accepted on-wire	event
<pre>select 'accepted' as ap_full_status, devid, vd, ssid, bssid, manuf, channel, radioband, from_dtime(max(last_seen)) as last_seen, detectionmethod,</pre>		

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

Dataset Name	Descriptior	ı	l	Log Category
event-Wireless-Rogue-Offwire	Event wirele	ss rogue off-wire	(event
select				
'rogue' as ap_full_status,				
devid,				
vd,				
ssid,				
bssid,				
manuf,				
channel,				
radioband,				
<pre>from_dtime(max(last_seen)</pre>				
) as last seen,				
detectionmethod,				
snclosest,				
'no' as on_wire				
from				
### (select devid vd ssid	hesid manuf	channel radioband	detectionmethod	enclosest

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

Dataset Name	Description	Log Category
event-Wireless-Rogue-Onwire	Event wireless rogue on-wire	event
<pre>select 'rogue' as ap_full_status, devid, vd, ssid, bssid, manuf, channel, radioband, from_dtime(max(last_seen)) as last_seen, detectionmethod, snclosest,</pre>		

```
'yes' as on_wire
from
###(select devid
```

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

Dataset Name	Description	Log Category
event-Wireless-Suppressed-Offwire	Event wireless suppressed off-wire	event
select		
'suppressed' as ap_full_status	,	
devid,		
vd,		
ssid,		
bssid,		
manuf,		
channel,		
radioband,		
from_dtime(
max(last_seen)		
) as last_seen,		
detectionmethod,		
snclosest,		
'no' as on_wire		
from		

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

Dataset Name	Description	Log Category
event-Wireless-Suppressed-Onwire	Event wireless suppressed on-wire	event
<pre>select 'suppressed' as ap_full_status devid, vd, ssid, bssid, manuf, channel, radioband, from_dtime(max(last_seen)</pre>		event
) as last_seen, detectionmethod, snclosest, 'yes' as on_wire		

from

###(select devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest, onwire, apstatus, max(dtime) as last_seen from \$log where \$filter and bssid is not null and logid_to_int(logid) in (43521, 43525, 43563, 43564, 43565, 43566, 43569, 43570, 43571) group by devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest, onwire, apstatus order by last_seen desc)### t where apstatus=3 and onwire='yes' group by devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest order by last_seen desc

Dataset Name	Description	Log Category
event-Wireless-Unclassified-Offwire	Event wireless unclassified off-wire	event
<pre>select 'unclassified' as ap_full_stat devid, vd, ssid, bssid, manuf, channel, radioband, from_dtime(max(last_seen)) as last_seen, detectionmethod, snclosest,</pre>	us,	
'no' as on_wire from		
onwire, logid, apstatus, max(dti null and logid_to_int(logid) in 43571) group by devid, vd, ssid, snclosest, onwire, logid, apstat	<pre>sid, manuf, channel, radioband, detecti me) as last_seen from \$log where \$filte (43521, 43525, 43563, 43564, 43565, 435 bssid, manuf, channel, radioband, dete us order by last_seen desc)### t where ssid, bssid, manuf, channel, radioband,</pre>	er and bssid is not 566, 43569, 43570, ectionmethod, apstatus=0 and

snclosest order by la	ast_seen desc
Dataset Name	Description

event-Wireless-Unclassified-Onwire	Event wireless unclassified on-wire	event
elect		
'unclassified' as ap_full_sta	tus,	
devid,		
vd,		
ssid,		
bssid,		
manuf,		
channel,		
radioband,		
from_dtime(
max(last_seen)		
) as last_seen,		
detectionmethod,		
snclosest,		
'yes' as on_wire		
rom		

Log Category

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

Dataset Name

Description

Log Category

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

select coalesce(xauthuser agg, user agg, ipstr(`remip`)) as user src, from dtime(min(s time)) as start time, sum(bandwidth) as bandwidth, sum(traffic in) as traffic in, sum(traffic out) as traffic out from (select devid, vd, string_agg(distinct xauthuser_agg, ' ') as xauthuser_agg, string agg(distinct user agg, ' ') as user_agg, remip, tunnelid, min(s time) as s time, max(e time) as e time, case when min(s time) = max(e time) then max(max traffic in) + max(max traffic out) else max(max traffic in) - min(min traffic in) + max(max traffic out) - min(min traffic out) end) as bandwidth, (case when min(s time) = max(e time) then max(max traffic in) else max(max traffic in) - min(min traffic in) end) as traffic in, (case when min(s time) = max(e time) then max(max traffic out) else max(max traffic out) - min(min traffic out) end) as traffic out from ###(select devid, vd, remip, nullifna(`xauthuser`) as xauthuser agg, nullifna(`user`) as user_agg, tunnelid, min(coalesce(dtime, 0)) as s_time, max(coalesce(dtime, 0)) as e_time, max(coalesce(duration,0)) as max duration, min(coalesce(duration,0)) as min duration, min (coalesce(sentbyte, 0)) as min traffic out, min(coalesce(rcvdbyte, 0)) as min traffic in, max(coalesce(sentbyte, 0)) as max traffic out, max(coalesce(rcvdbyte, 0)) as max traffic in

tunnelid is not null and tunnelid!=0 group by devid, vd, remip, xauthuser_agg, user_agg, tunnelid order by tunnelid)### t group by devid, vd, remip, tunnelid) tt group by user_src having sum(bandwidth)>0 order by bandwidth desc

Dataset Name	Description	Log Category
default-Top-Sources-Of-SSL-VPN- Tunnels-By-Bandwidth	Default top sources of SSL VPN tunnels by bandwidth usage	event
<pre>in) - min(min_traffic_in) end) as traffic_in, (case when min(s_time) = : out) - min(min_traffic_out) end) as traffic_out, (case when min(s_time) = : else max(max_traffic_in) - min(m end) as bandwidth from ###(select \$flex_timestam tunneltype like 'ipsec%' then ' action='tunnel-up' then 1 else traffic_out, max(coalesce(rcvdb min_traffic_out, min(coalesce(r time, max(coalesce(dtime, 0)) a (tunneltype like 'ipsec%' or tu stats', 'tunnel-down') and tunn vd, remip, t_type, tunnelid, ac</pre>	<pre>max(e_time) then max(max_traffic_in) else max(max(e_time) then max(max_traffic_out) else max max(e_time) then max(max_traffic_in)+ max(max_ in_traffic_in)+ max(max_traffic_out)- min(min_ p as timestamp, devid, vd, remip, tunnelid, (c ipsec' else tunneltype end) as t_type, (case w 0 end) as tunnelup, max(coalesce(sentbyte, 0)) yte, 0)) as max_traffic_in, min(coalesce(sentb cvdbyte, 0)) as min_traffic_in, min(coalesce(d s e_time from \$log where \$filter and subtype=' nneltype like 'ssl%') and action in ('tunnel-u elid is not null and tunnelid!=0 group by time tion /*SkipSTART*/order by timestamp desc/*Ski by devid, vd, remip, tunnelid) tt group by rem rder by bandwidth desc</pre>	<pre>(max_traffic_ traffic_out) traffic_out) ase when hen as max_ yte, 0)) as time, 0)) as s_ vpn' and p','tunnel- stamp, devid, pEND*/)### t</pre>
Dataset Name	Description	Log Category
webfilter-Web-Activity-Summary-By- Requests	Webfilter web activity summary by requests	webfilter
select		

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

end) as allowed_request, sum(case when action='blocked' then 1 else 0 end) as blocked_ request from \$log where \$filter and (eventtype is null or logver>=502000000) group by timestamp /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t group by hodex order by hodex

Dataset Name	Description	Log Category
traffic-Browsing-Time-Summary	Traffic browsing time summary	traffic
\$log where \$filter and (logfla		oup by timestamp
Dataset Name	Description	Log Category
traffic-Browsing-Time-Summary- Enhanced	Traffic browsing time summary enhanced	traffic
<pre>select \$flex_timescale(timestamp) a cast(ebtr value(</pre>	s hodex,	

```
ebtr_value(
    ebtr_agg_flat(browsetime),
    null,
    $timespan
)/ 60.0 as decimal(18, 2)
) as browsetime
```

from

###(select \$flex_timestamp as timestamp, ebtr_agg_flat(\$browse_time) as browsetime from \$log where \$filter and (logflag&1>0) and \$browse_time is not null group by timestamp /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t group by hodex order by hodex

Dataset Name	Description	Log Category
webfilter-Top-Web-Users-By-Blocked- Requests	Webfilter top web users by blocked requests	webfilter
<pre>from \$log where \$filter and (eve (`user`), ipstr(`srcip`)) is not</pre>	user`), ipstr(`srcip`)) as user_src, count(*) nttype is null or logver>=502000000) and coal null and action='blocked' group by user_src esc/*SkipEND*/)### t group by user_src order b	esce(nullifna

Dataset Name	Description	Log Category
webfilter-Top-Web-Users-By-Allowed- Requests	Webfilter top web users by allowed requests	webfilter
<pre>from \$log where \$filter and (eve (`user`), ipstr(`srcip`)) is not</pre>	user`), ipstr(`srcip`)) as user_src, count enttype is null or logver>=502000000) and o null and action!='blocked' group by user_ lesc/*SkipEND*/)### t group by user_src ord	coalesce(nullifna _src

Dataset Name	Description	Log Category
traffic-Top-Web-Users-By-Browsing- Time	Traffic top web users by browsing time	traffic
<pre>sum(traffic_in) as traffic_in, s (nullifna(`user`), ipstr(`srcip` sum(coalesce(sentbyte, 0)+coales traffic_in, sum(coalesce(sentbyt time is not null group by user_s</pre>		<pre>lect coalesce ime) as browsetime, esce(rcvdbyte, 0)) as filter and \$browse_ der by ebtr_value</pre>

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 and action='blocked' group by do	catdesc, count(*) as requests from \$log wher 2000000) and hostname is not null and catdesc main, catdesc /*SkipSTART*/order by requests Lomain, catdesc order by requests desc	

Dataset Name	Description	Log Category
webfilter-Top-Allowed-Web-Sites-By- Requests	Webfilter top allowed web sites by requests	webfilter
(eventtype is null or logver>=5	- n, catdesc, count(*) as requests from \$log when 502000000) and hostname is not null and catdesc domain, catdesc /*SkipSTART*/order by requests	c is not null
Dataset Name	Description	Log Category
webfilter-Top-Video-Streaming- Websites-By-Bandwidth	Webfilter top video streaming websites by bandwidth usage	webfilter

select

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

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

```
Dataset NameDescriptionLog Categorywebfilter-Top-Blocked-Web-CategoriesWebfilter top blocked web categorieswebfilterselect<br/>catdesc,<br/>sum(requests) as requestsselect<br/>is null or logver>=50200000) and catdesc is not null and action='blocked' group by catdesc<br/>/*SkipSTART*/order by requests desc/*SkipEND*/)### t group by catdesc order by requests descDataset NameDescriptionLog Category
```

webfilter-Top-Allowed-Web-Categories	Webfilter top allowed web categories	webfilter

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

###(select catdesc, count(*) as requests from \$log-webfilter where \$filter and (eventtype is null or logver>=502000000) and catdesc is not null and action!='blocked' group by catdesc /*SkipSTART*/order by requests desc/*SkipEND*/)### t group by catdesc order by requests desc

Dataset Name	Description		Log Category
traffic-Top-50-Sites-By-Browsing-Time	Traffic top sites by browsing tim	e	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,</pre>	_		
<pre>sum(traffic_out) as traffic_out from ###(select hostname, catdesc, e</pre>		as browsetime,	sum(bandwidth) as

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

Dataset Name	Description	Log Category
traffic-Top-50-Sites-By-Browsing- Time-Enhanced	Traffic top sites by browsing time enhanced	traffic
<pre>bandwidth, sum(traffic_in) as tr hostname, catdesc, ebtr_agg_flat 0)+coalesce(rcvdbyte, 0)) as bar (coalesce(sentbyte, 0)) as traff hostname is not null and \$browse hostname, catdesc /*SkipSTART*/c</pre>	ebtr_agg_flat(browsetime) as browsetime, sum(caffic_in, sum(traffic_out) as traffic_out fro c(\$browse_time) as browsetime, sum(coalesce(se adwidth, sum(coalesce(rcvdbyte, 0)) as traffic fic_out from \$log where \$filter and (logflag&1 e_time is not null group by hostname, catdesc) order by ebtr_value(ebtr_agg_flat(browsetime),	om (select entbyte, _in, sum >0) and t group by
	nostname order by browsetime desc	

Dataset Name	Description	Log Category
traffic-Top-10-Categories-By- Browsing-Time	Traffic top category by browsing time	traffic
<pre>select catdesc, ebtr_value(ebtr_agg_flat(browsetime), null, \$timespan</pre>	,	
) as browsetime, sum(bandwidth) as bandwidth		
—	g_flat(browsetime) as browsetime, sum(bar	

from (select catdesc, ebtr_agg_flat(\$browse_time) as browsetime, sum(coalesce(sentbyte,

0)+coalesce(rcvdbyte, 0)) as bandwidth from \$log where \$filter and (logflag&1>0) and catdesc is not null and \$browse_time is not null group by catdesc) t group by catdesc

/*SkipSTART*/order by ebtr_value(ebtr_agg_flat(browsetime), null, null) desc/*SkipEND*/)###
t group by catdesc order by browsetime desc

Dataset Name	Description	Log Category
traffic-Top-10-Categories-By- Browsing-Time-Enhanced	Traffic top category by browsing time enhanced	traffic
<pre>from (select catdesc, ebtr_agg_f 0)+coalesce(rcvdbyte, 0)) as ban is not null and \$browse_time is</pre>	<pre>lat(browsetime) as browsetime, sum(bandwidth lat(\$browse_time) as browsetime, sum(coalesc dwidth from \$log where \$filter and (logflag& not null group by catdesc) t group by catdes (ebtr_agg_flat(browsetime), null, null) desc setime desc</pre>	e(sentbyte, 1>0) and catdesc c
Dataset Name	Description	Log Category
traffic-Top-Destination-Countries-By-	Traffic top destination countries by browsing time	traffic

Browsing-Time

```
select
dstcountry,
ebtr_value(
   ebtr_agg_flat(browsetime),
   null,
   $timespan
) as browsetime,
   sum(bandwidth) as bandwidth,
```

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

Dataset Name	Description	Log Category
traffic-Top-Destination-Countries-By- Browsing-Time-Enhanced	Traffic top destination countries by browsing time enhanced	traffic
	ut gg_flat(browsetime) as browsetime, sum(bandwi	
<pre>dstcountry, ebtr_agg_flat(\$brows (rcvdbyte, 0)) as bandwidth, sum (sentbyte, 0)) as traffic_out fr not null group by dstcountry) t</pre>	<pre>caffic_in, sum(traffic_out) as traffic_out fr se_time) as browsetime, sum(coalesce(sentbyte a(coalesce(rcvdbyte, 0)) as traffic_in, sum(c com \$log where \$filter and (logflag&1>0) and group by dstcountry /*SkipSTART*/order by eb a) desc/*SkipEND*/)### t group by dstcountry</pre>	, 0)+coalesce oalesce \$browse_time is tr_value(ebtr_

browsetime desc

Dataset Name	Description	Log Category
webfilter-Top-Search-Phrases	Webfilter top search phrases	webfilter
<pre>select keyword, count(*) as requests from \$log where \$filter and keyword is not null group by keyword order by requests desc</pre>		

Dataset Name	Description	Log Category
Top-10-Users-Browsing-Time	Estimated browsing time	traffic
<pre>(nullifna(`user`), nullifna((\$browse_time) as browsetime</pre>	agg_flat(browsetime) as browsetime from `unauthuser`), ipstr(`srcip`)) as user_s: from \$log where \$filter and (logflag&1>)	rc, ebtr_agg_flat 0) and \$browse_time is
—	t group by user_src order by ebtr_value(sc)### t group by user_src order by brows	
Dataset Name	Description	Log Category
Top-10-Users-Browsing-Time- Enhanced	Estimated browsing time enhanced	traffic
select		

```
user_src,
 ebtr_value(
   ebtr_agg_flat(browsetime),
   null,
   $timespan
 ) as browsetime
from
```

###(select user_src, ebtr_agg_flat(browsetime) as browsetime from (select coalesce (nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, ebtr_agg_flat (\$browse_time) as browsetime from \$log where \$filter and (logflag&1>0) and \$browse_time is not null group by user_src) t group by user_src order by ebtr_value(ebtr_agg_flat (browsetime), null, null) desc)### t group by user_src order by browsetime desc

Dataset Name	Description	Log Category
Estimated-Browsing-Time	Estimated browsing time	traffic
<pre>select user_src, ebtr_value(ebtr_agg_flat(browsetime), null, \$timespan) as browsetime from</pre>		
<pre>###(select user_src, ebtr_agg_flat(browsetime) as browsetime from (select coalesce (nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, ebtr_agg_flat (\$browse_time) as browsetime from \$log where \$filter and (logflag&1>0) and \$browse_time is not null group by user_src) t group by user_src order by ebtr_value(ebtr_agg_flat (browsetime), null, null) desc)### t group by user_src order by browsetime desc</pre>		

Dataset Name	Description	Log Category
Estimated-Browsing-Time-Enhanced	Estimated browsing time enhanced	traffic
<pre>(nullifna(`user`), nullifna(`una (\$browse_time) as browsetime from not null group by user_src) t gr</pre>	<pre>flat(browsetime) as browsetime from (se uthuser`), ipstr(`srcip`)) as user_src, m \$log where \$filter and (logflag&1>0) oup by user_src order by ebtr_value(ebt ## t group by user src order by browset</pre>	, ebtr_agg_flat and \$browse_time is tr_agg_flat

Dataset Name	Description	Log Category
wifi-Top-AP-By-Bandwidth	Top access point by bandwidth usage	traffic
<pre>select coalesce(ap, srcintf) as a sum(coalesce(sentbyte, 0)+ a) as bandwidth from \$log where \$filter and (logflag&1>0) and (srcssid is not null or dstssid is not null</pre>		
<pre>) group by ap_srcintf having sum(coalesce(sentbyte, 0)+ 0)& gt; 0 order by bandwidth desc</pre>	coalesce(rcvdbyte, 0)	

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) a</pre>	as totalnum	
<pre>from ###(select coalesce(ap,</pre>	<pre>srcintf) as ap_srcintf, srcssid, osname,</pre>	<pre>srcswversion, get_devtype</pre>

(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)+ c) as bandwidth from \$log where \$filter and (logflag&1>0)</pre>	oalesce(rcvdbyte, 0)	
<pre>and srcssid is not null group by srcssid having sum(coalesce(sentbyte, 0)+ c)& gt; 0 order by bandwidth desc</pre>	oalesce(rcvdbyte, 0)	

Dataset Name	Description	Log Category
wifi-Top-SSID-By-Client	Top SSIDs by client	traffic
subtotal from \$log where \$ not null) and srcmac is no	as totalnum ssid, osname, srcswversion, osversion, devtyr filter and (logflag&1>0) and (srcssid is not ot null group by srcintf, srcssid, osname, so subtotal desc)### t where srcssid is not null	null or dstssid is cswversion, osversion,
Dataset Name	Description	Log Category
wifi-Top-App-By-Bandwidth	Top WiFi applications by bandwidth usage	traffic
<pre>select appid, app,</pre>		

```
sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
) as bandwidth
from
```

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

Dataset Name	Description	Log Category
wifi-Top-Client-By-Bandwidth	Top WiFi client by bandwidth usage	traffic
select		
(
	, 'unknown') ' (' get_devtype(srcswve	rsion, osname,
devtype) ', ' coalesce		
) ')'	is null then '' else ' ' srcswversion e	na
) as client,		
sum (
<pre>coalesce(sentbyte, 0) + co</pre>	palesce(rcvdbyte, 0)	
) as bandwidth		
from		
\$log		
where		
\$filter and (
logflag&1>0		
)		
and (
srcssid is not null		
or dstssid is not null		
)		
group by		
client		
having sum (
coalesce(sentbyte, 0) + co	palesce(rcvdbvte, 0)	
) & gt; 0	(10.00) (10.00) (0)	
order by		

bandwidth desc

Dataset Name	Description	Log Category
wifi-Top-OS-By-Bandwidth	Top WiFi os by bandwidth usage	traffic
select		
(
	<pre>wn') ' ' coalesce(srcswversion, '')</pre>	
) as os, sum(
coalesce(sentbyte, 0)+	coalesce(rcvdbyte, 0)	
) as bandwidth		
from		
\$log		
where		
\$filter		
and (
logflag&1>0		
)		
and (
srcssid is not null		
or dstssid is not null		
) group by		
os		
having		
sum (
coalesce(sentbyte, 0)+	coalesce(rcvdbyte, 0)	
)& gt; 0	-	
order by		
bandwidth desc		

Dataset Name	Description	Log Category
wifi-Top-OS-By-WiFi-Client	Top WiFi os by WiFi client	traffic
<pre>) as os, count(distinct srcmac) as to from ###(select srcintf, srcssid, subtotal from \$log where \$filte not null) and srcmac is not null</pre>	calnum osname, srcswversion, osversion, dev er and (logflag&1>0) and (srcssid is ll group by srcintf, srcssid, osname, cal desc)### t group by os order by t	not null or dstssid is , srcswversion, osversion,
Dataset Name	Description	Log Category
wifi-Top-Device-By-Bandwidth	Top WiFi device by bandwidth usage	traffic

select

```
get_devtype(srcswversion, osname, devtype) as devtype_new,
sum(
    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
 )
 and devtype is not null
group by
 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
select devtype new,		
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, 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
<pre>select sum(coalesce(sentbyte, 0)) as bandwidth from \$log where \$filter and (logflag&1>0) and (srcssid is not null or dstssid is not null)</pre>	+ coalesce(rcvdbyte, 0)	
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-
                                                                                     traffic
                                   Top subnets by application bandwidth
 Sessions
select
  ip subnet(`srcip`) as subnet,
  sum(
    coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
  ) as bandwidth,
  sum(
    coalesce(rcvdbyte, 0)
  ) as traffic in,
  sum(
    coalesce(sentbyte, 0)
  ) as traffic out,
  count(*) as sessions
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
group by
  subnet
having
  sum(
    coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
  )& gt; 0
order by
 bandwidth desc
```

Dataset Name	Description	Log Category
Top30-Subnets-by-Application- Bandwidth	Top applications by bandwidth	traffic
<pre>select ip_subnet(`srcip`) as subnet, app_group_name(app) as app_gr sum(coalesce(sentbyte, 0)+ coal) as bandwidth from \$log where \$filter</pre>	coup,	

```
and (
    logflag&1>0
)
    and nullifna(app) is not null
group by
    subnet,
    app_group
having
    sum(
        coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
    )& gt; 0
order by
    bandwidth desc
```

Dataset Name	Description	Log Category
Top30-Subnets-by-Application- Sessions	Top applications by sessions	traffic
<pre>select ip_subnet(`srcip`) as subnet app_group_name(app) as app_c count(*) as sessions from \$log where \$filter and (logflag&1>0) and nullifna(app) is not nul</pre>	group,	
<pre>group by subnet, app_group order by sessions desc</pre>		

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

select
 subnet,
 website,
 sum(bandwidth) as bandwidth
from

###(select ip_subnet(`srcip`) as subnet, hostname as website, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from \$log-traffic where \$filter and hostname is not null and (logflag&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', 'commandblock', 'script-filter'))) group by subnet, website order by bandwidth desc)### t group by subnet, website order by bandwidth desc

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

select
subnet,
website,
sum(hits) as hits
from
####(select ip_subnet(`srcip`) as subnet, hostname as website, count(*) as hits from \$log
where \$filter and hostname is not null and (eventtype is null or logver>=502000000) group by
subnet, website order by hits desc)### t group by subnet, website order by hits desc

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

count(*) as sessions

from

```
$log
where
  $filter
  and (
    logflag&1>0
)
group by
  subnet,
   user_src
order by
  sessions desc
```

Dataset Name	Description		Log Category
app-Top-20-Category-and- Applications-by-Bandwidth	Top category and	applications by bandwidth usage	traffic
<pre>select appcat, app, sum(coalesce(sentbyte,) as bandwidth from \$log where \$filter and (logflag&1>0) group by appcat, app having sum(</pre>	0)+ coalesce(rcvdbyte, 0))	
<pre>coalesce(sentbyte,)& gt; 0 order by</pre>	0)+ coalesce(rcvdbyte, ())	

bandwidth desc

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)</pre>		

group by

appcat, app order by sessions desc **Dataset Name** Description Log Category app-Top-500-Allowed-Applications-by- Top allowed applications by bandwidth usage traffic Bandwidth select from_itime(itime) as timestamp, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user src, appcat, app, coalesce(root domain (hostname), ipstr(dstip)) as destination, sum(coalesce(`sentbyte`, 0)+ coalesce(`rcvdbyte`, 0)) as bandwidth from \$log where \$filter and (logflag&1>0) and action in ('accept', 'close', 'timeout') group by timestamp, user src, appcat, app, destination order by bandwidth desc Log Category

Dataset Name	D	a	ta	S	et	Ν	la	m	e
--------------	---	---	----	---	----	---	----	---	---

Description

app-Top-500-Blocked-Applications-by- Session	Top blocked applications by session	traffic
<pre>select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, appcat,</pre>		

```
app,
  count(*) as sessions
from
  $log
where
  $filter
  and (
   logflag&1>0
  )
  and action in (
    'deny', 'blocked', 'reset', 'dropped'
  )
group by
 user_src,
 appcat,
  app
order by
  sessions desc
```

Dataset Name

web-Detailed-Website-Browsing-Log

Web detailed website browsing log

Description

Log Category

```
select
from_dtime(dtime) as timestamp,
catdesc,
hostname as website,
status,
sum(bandwidth) as bandwidth
from
```

###(select dtime, catdesc, hostname, cast(utmaction as text) as status, sum(coalesce (sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from \$log-traffic where \$filter and hostname is not null and (logflag&1>0) and (countweb>0 or ((logver is null or logver<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- Hits-Action	Web hourly category and website hits action	webfilter
')') as website , count(*) as hi	l, (hostname ' (' coalesce(`catdesc`, 'U ts from \$log where \$filter and hostname is no 22000000) group by hod, website order by hod, wod, hits desc	t null and

Dataset Name	Description	Log Category
web-Top-20-Category-and-Websites- by-Bandwidth	Web top category and websites by bandwidth usage	traffic
select website,		
catdesc, sum(bandwidth) as bandwidth		

from

###(select hostname as website, catdesc, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))
as bandwidth from \$log-traffic where \$filter and hostname is not null and (logflag&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')))) group by
website, catdesc order by bandwidth desc)### t group by website, catdesc order by bandwidth
desc</pre>

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</pre>		

from

###(select hostname as website, catdesc, count(*) as sessions from \$log where \$filter and hostname is not null and (eventtype is null or logver>=502000000) group by hostname, catdesc order by sessions desc)### t group by website, catdesc order by hits desc

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

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

from

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

Dataset Name	Description	Log Category
web-Top-500-User-Visted-Websites- by-Bandwidth	Web top user visted websites by bandwidth usage	traffic
<pre>select website, catdesc, sum(bandwidth) as bandwidth</pre>		
-	e, catdesc, sum(coalesce(sentbyte, 0)+coales	

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 hostname, catdesc having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc)### t group by website, catdesc order by bandwidth desc

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

```
catdesc,
sum(sessions) as sessions
from
```

###(select hostname as website, catdesc, count(*) as sessions from \$log where \$filter and hostname is not null and (eventtype is null or logver>=502000000) group by hostname, catdesc order by sessions desc)### t group by website, catdesc order by sessions desc

Dataset Name	Description	Log Category
fct-Installed-Feature-Summary	Installed Feature Summary	fct-event
<pre>select clientfeature, count(distinct fctuid) as totalnum from ###(select uid as fctuid, regexp_replace(os, '\\(build.*', '') as os_short, fctver, clientfeature, fgtserial, max(case when msg like 'Compliance rules%applied' then 1 else 0 end) as compliance_flag from \$log where \$filter group by uid, os_short, fctver, clientfeature, fgtserial)### t where clientfeature is not null group by clientfeature order by totalnum desc</pre>		
_	t where clientfeature is not nul	l group by clientfeature order
_	t where clientfeature is not nul Description	l group by clientfeature order Log Category
by totalnum desc		

end) as compliance_flag from \$log where \$filter group by uid, os_short, fctver, clientfeature, fgtserial)### t where os_short is not null group by os order by totalnum desc

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

select

fctver as fctver short, count(distinct fctuid) as totalnum from

###(select uid as fctuid, regexp_replace(os, '\\(build.*', '') as os_short, fctver, clientfeature, fgtserial, max(case when msg like 'Compliance rules%applied' then 1 else 0 end) as compliance flag from \$log where \$filter group by uid, os short, fctver, clientfeature, fgtserial)### t where fctver is not null group by fctver order by totalnum desc

Dataset Name	Description	Log Category
fct-Endpoint-Profile-Deployment	Endpoint Profile Deployment	fct-event
	alnum lesce(nullifna(usingpolicy), 'No Profil , profile)### t group by profile order :	-
Dataset Name	Description	Log Category
fct-Client-Summary	Client Summary	fct-event
fct-Client-SummaryClient Summaryfct-eventselect hostname, deviceip, os_short as os, profile, fctver, from_itime(
Dataset Name	Description	Log Category
fct-Total-Threats-Found	Total Threats Found	fct-traffic

select

```
utmevent s as utmevent,
 count(distinct threat) as totalnum
from
 ###(select coalesce(nullifna(lower(utmevent)), 'unknown') as utmevent s, threat from $log
```

where \$filter and threat is not null and utmaction='blocked' group by utmevent_s, threat)### t group by utmevent order by totalnum desc

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

select
 threat,
 sum(totalnum) as totalnum
from

(

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

Dataset Name	Description	Log Category
fct-Top10-Infected-Devices-with- Botnet	Top Infected Devices with Botnet	fct-traffic
<pre>select hostname, count(*) as totalnum from \$log where \$filter and hostname is not null and lower(utmevent) in ('webfi and lower(threat) like '%botne group by hostname order by totalnum desc</pre>		
Dataset Name	Description	Log Category
fct-Top10-Infected-Devices-with-Virus- Malware	Top Infected Devices with Virus Malware	fct-traffic
hostname is not null and lower(u order by totalnum desc)### union fct-event where \$filter and host	<pre>as totalnum from \$log-fct-traffic where atmevent) in ('antivirus', 'antimalware') a all ###(select hostname, count(*) as tot chame is not null and virus is not null gr coup by hostname order by totalnum desc</pre>	group by hostname alnum from \$log-

Description	Log Category
All Antivirus and Antimalware Detections	fct-traffic
<pre>coalesce(nullifna(`user`), 'Unknown') as hos rom \$log-fct-traffic where \$filter and lower(up by threat, hostname, hostuser, utmaction c virus as threat, hostname, coalesce(nullifna(s utmaction, max(dtime) as dtime from \$log-fc logflag&8=0) and virus is not null group by t</pre>	utmevent) in order by `user`), ct-event where
	<pre>coalesce(nullifna(`user`), 'Unknown') as hos com \$log-fct-traffic where \$filter and lower(up by threat, hostname, hostuser, utmaction c rirus as threat, hostname, coalesce(nullifna(s utmaction, max(dtime) as dtime from \$log-fc</pre>

Dataset Name	Description	Log Category
ct-Web-Filter-Violations	Web Filter Violations	fct-traffic
elect		
hostuser, hostname,		
string_agg(distinct remote	ename, ',') as remotename,	
utmaction, sum(total) as totalnum,		
from_dtime(
max(dtime)		
) as last_seen		

utmaction, count(*) as total, max(dtime) as dtime from \$log where \$filter and lower (utmevent)='webfilter' and utmaction='blocked' group by remotename, hostname, hostuser, utmaction order by total desc)### t group by hostuser, hostname, utmaction order by totalnum desc

Dataset Name	Description	Log Category
fct-Application-Firewall	Application Firewall	fct-traffic
select		
threat,		
hostname,		
hostuser,		
utmaction,		
from_dtime(
max(dtime)		
) as last seen		

from

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

Dataset Name	Description	Log Category
fct-Errors-and-Alerts	Errors and Alerts	fct-event
<pre>select msg, hostname, hostuser, from_dtime(max(dtime))</pre>		
<pre>) as last_seen from ###(select msg, hostn</pre>	ame, coalesce(nullifna(`user`), 'Unknown	') as hostuser, max(dtime) as

###(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>'Info' THEN 2 WHEN 'Low' THEN) as severity_number, count(distinct vulnname) as from</pre>		

(vulnseverity) is not null and nullifna(vulnname) is not null group by vulnseverity, vulnname, devid)### t group by vulnseverity order by severity_number desc

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

select

vulncat,

```
count(distinct vulnname) as totalnum
```

from

###(select vulncat, vulnname from \$log where \$filter and nullifna(vulncat) is not null and nullifna(vulnname) is not null group by vulncat, vulnname)### t group by vulncat order by totalnum desc

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

os, count(distinct vulnname) as totalnum

from

###(select os, vulnname from \$log where \$filter and nullifna(os) is not null and nullifna
(vulnname) is not null group by os, vulnname)### t group by os order by totalnum desc

Dataset Name	Description	Log Category	
fct-vuln-Vulnerabilities-by-Risk-Level	Number Vulnerability by Device and Risk Level	fct-netscan	
<pre>vulnseverity = 'Medium' then 3 y then 1 else 0 end) as severity_number, count(distinct vulnname) as v count(distinct devid) as dev_s from ###(select vulnseverity, devid (vulnseverity) is not null and s</pre>		severity = 'Info' llifna a(devid) is not	
Dataset Name	Description	Log Category	
fct-vuln-Device-by-Risk-Level	Number Vulnerability by Device and Risk Level	fct-netscan	
<pre>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,</pre>			

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

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

select

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

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

Dataset Name	Description	Log Category
fct-vuln-Details-by-Risk-Level-Device	Vulnerability Details for Each Risk Level by Device	fct-netscan
vulnname is not null and vulnseven hostname, os, vulnname, vulnseven	coducts,	roup by m
Dataset Name	Description	Log Category

Dataset Name	Description	LUg Calegory
fct-vuln-Details-by-Device-User	Vulnerability Details by Device User	fct-netscan
select		
hostname,		
(
' <div>' vulnname '<</div>	/div>'	
) as vulnname,		
vulnseverity,		
vulncat,		
string_agg(distinct product	s, ',') as products,	
<pre>string_agg(distinct cve_id,</pre>	',') as cve_list,	
(
<pre>'<a '="" ,')="" href=" String_agg(</pre></td><td>DISTINCT vendor_link, " ="">Remediati</pre>	on Info'	
) as vendor_link		
from		
###(select hostname, vulnna	me, vulnseverity, vulncat, vulnid from \$1	og where \$filter and
vulnname is not null and host	name is not null group by hostname, vulnn	ame, vulnseverity,

vulncat, vulnid)### t1 inner join fct_mdata t2 on t1.vulnid=t2.vid::int group by hostname, vulnname, vulnseverity, vulncat order by hostname

Dataset Name	Description	Log Category
fct-vuln-Remediation-by-Device	Remediate The Vulnerability Found on Device	fct-netscan
<pre>select hostname, ('<div>' vulnname '</div></pre>		

from

###(select hostname, vulnname, vulnseverity, vulnid from \$log where \$filter and vulnname
is not null and hostname is not null group by hostname, vulnname, vulnseverity, vulnid)###
t1 inner join fct_mdata t2 on t1.vulnid=t2.vid::int group by hostname, vulnname,
vulnseverity order by vulnseverity, hostname

Dataset Name	Description	Log Category
fct-vuln-Remediation-by-Vulnerability	Remediation by Vulnerability	fct-netscan
<pre> description ' <b 'Impact ' impact '< ' ') as remediation</br></b </pre>	r/> ' 'Description <di r/>' 'Affected Products ' br/> ' 'Recommended Action</br></di 	products ' >/>'
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
	',') as products, dor_link, ',') '>Mitigation Infomation'	
<pre>) as vendor_link from ###(select vulnid from \$log where \$filter group by vulnid)### t1 inner join fct_mdata t2 on t2.vid=t1.vulnid::text inner join fct_cve_score t3 on strpos(t2.cve_id, t3.cve_id) > 0 group by t3.cve_id, score order by score desc, t3.cve_id</pre>		

Dataset Name	Description	Log Category
fct-Endpoints-by-FortiGate	Endpoints by FortiGate	fct-event
<pre>select fgtserial, count(distinct fctuid) as from</pre>	totalnum	
<pre>###(select uid as fctuid, regexp_replace(os, '\\(build.*', '') as os_short, fctver, clientfeature, fgtserial, max(case when msg like 'Compliance rules%applied' then 1 else 0 end) as compliance_flag from \$log where \$filter group by uid, os_short, fctver, clientfeature, fgtserial)### t where fgtserial is not null group by fgtserial order by</pre>		

totalnum desc

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

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

```
(
```

###(select threat, hostname, coalesce(nullifna(`user`), 'Unknown') as hostuser, 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)### union all ###(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 order by totalnum desc

Dataset Name	Description	Log Category
fct-Top10-Malware-Detections	Top 10 Infected Devices with Malware	fct-traffic
<pre>select threat, hostname, hostuser, utmaction, fctuid,</pre>		
utmaction, max(dtime) as dtime,	, coalesce(nullifna(`user`), 'Unknown') as uid as fctuid, count(*) as totalnum from rus', 'antimalware') group by threat, host	\$log where \$filter
<pre>(nullifna(`user`), 'Unknown') as fctuid, count(*) as totalnum fro logflag&8=0) and virus is not no</pre>	<pre>### union all ###(select virus as threat, s hostuser, action as utmaction, max(dtime om \$log-fct-event where \$filter and (logfl all group by threat, hostname, hostuser, u n != 'pass' group by threat, hostname, hos</pre>) as dtime, uid as ag is null or tmaction, uid order

Dataset Name	Description	Log Category
fct-Devices-with-Botnet	Infected Devices with Botnet	fct-traffic
<pre>elect threat, hostname, coalesce(nullifna(`user`), 'Unknown') as hostuser, utmaction, uid as fctuid, count(*) as totalnum from \$log there \$filter and hostname is not null and lower(utmevent) in ('webf and lower(threat) like '%botn froup by threat,</pre>		
hostname, hostuser, utmaction, fctuid order by totalnum desc		
hostuser, utmaction, fctuid rder by totalnum desc	Description	Log Category
hostuser, utmaction, fctuid order by	Description Vulnerability Details for Each Risk Level by Device	Log Category fct-netscan
<pre>hostuser, utmaction, fctuid rder by totalnum desc Dataset Name fct-vuln-Vulnerability-by-Hostname elect hostname, os, vulnseverity, count(distinct vulnname) as v count(distinct products) as p count(distinct cve_id) as cve rom ###(select hostname, os, vuln ulnname is not null and vulnse ostname, os, vulnname, vulnsev</pre>	Vulnerability Details for Each Risk Level by Device	fct-netscan filter and group by on
<pre>hostuser, utmaction, fctuid rder by totalnum desc Dataset Name fct-vuln-Vulnerability-by-Hostname elect hostname, os, vulnseverity, count(distinct vulnname) as v count(distinct products) as p count(distinct cve_id) as cve rom ###(select hostname, os, vuln ulnname is not null and vulnse ostname, os, vulnname, vulnsev</pre>	Vulnerability Details for Each Risk Level by Device uln_num, roducts, _count name, vulnseverity, vulnid from \$log where \$ verity is not null and hostname is not null erity, vulnid)### t1 left join fct_mdata t2	fct-netscan filter and group by on

```
select
hostuser,
hostname,
string_agg(distinct remotename, ',') as remotename,
utmaction,
```

```
sum(total) as totalnum,
from_dtime(
    max(dtime)
) as last_seen
from
```

###(select remotename, hostname, coalesce(nullifna(`user`), 'Unknown') as hostuser, utmaction, count(*) as total, max(dtime) as dtime from \$log where \$filter and lower (utmevent)='webfilter' and utmaction='blocked' group by remotename, hostname, hostuser, utmaction order by total desc)### t group by hostuser, hostname, utmaction order by totalnum desc

Dataset Name	Description	Log Category	
fct-Compliance-by-FortiGate	FortiClinet Compliance by FortiGate Enforcing	fct-event	
select fgtserial,			
count(distinct fctuid) as t	otalnum		
from			
(
select			
fgtserial,			
fctuid,			
<pre>max(compliance_flag) as</pre>	compliance_flag		
from			
<pre>###(select uid as fctuid, regexp_replace(os, '\\(build.*', '') as os_short, fctver, clientfeature, fgtserial, max(case when msg like 'Compliance rules%applied' then 1 else 0 end) as compliance_flag from \$log where \$filter group by uid, os_short, fctver, clientfeature, fgtserial)### tt group by fgtserial, fctuid) t where compliance_flag = 1 group by fgtserial order by totalnum desc</pre>			
group by ignserial order by t	otainum desc		

Dataset Name	Description	Log Category
fct-Compliance-Status	Number of FortiClinets by Compliance Statu	us fct-event
select		
(
case compliance flag wh	nen 1 then 'Compliant' else 'Non-Complian	it' end
) as compliance,		
count(distinct fctuid) as	s totalnum	
from		
(
select		
fctuid,		
<pre>max(compliance_flag)</pre>	as compliance_flag	
from		
###(select uid as fct	tuid, regexp_replace(os, '\\(build.*', '') as os_short, fctver,
clientfeature, fgtserial, r	<pre>max(case when msg like 'Compliance rules%</pre>	applied' then 1 else 0
end) as compliance_flag fro	om \$log where \$filter group by uid, os_sh	Nort, fctver,
clientfeature, fgtserial)##	## tt group by fctuid) t group by complia	ance order by totalnum
desc		

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

```
select
   t1.fgtserial,
   t3.srcintf,
   t2.epname as hostname,
   t2.mac,
   'Non-Compliant' as status
from
   (
     select
        fgtserial,
        fctuid,
        max(compliance_flag) as compliance_flag
        from
```

###(select uid as fctuid, regexp_replace(os, '\\(build.*', '') as os_short, fctver, clientfeature, fgtserial, max(case when msg like 'Compliance rules%applied' then 1 else 0 end) as compliance_flag from \$log where \$filter group by uid, os_short, fctver, clientfeature, fgtserial)### tt group by fgtserial, fctuid) t1 left join \$ADOM_ENDPOINT t2 on t1.fctuid = t2.fctuid left join \$ADOM_EPEU_DEVMAP t3 on t2.epid = t3.epid where compliance_flag = 0 group by t1.fctuid, t1.fgtserial, t3.srcintf, t2.epname, t2.mac

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

```
select
```

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

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

Dataset Name	Description	Log Category		
fct-Traffic-Top-Allowed-Web-Cat	Top Visited Web Categories	fct-traffic		
<pre>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</pre>				
Dataset Name	Description	Log Category		
fct-Traffic-Top-Allowed-Website	Top Visited Websites			
,		fct-traffic		

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

Description	Log Category
Top Web Categories by Website Session	fct-traffic
11	

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)= 'webfilte</pre>		TCT-traffic
group by user src,		
website order by		
requests desc		

Dataset Name	Description	Log Category
os-Detect-OS-Count	Detected operation system count	traffic
select		
(coalesce(osname, '	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>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</pre>			
Dataset Name	Description	Log Category	
drilldown-Top-App-By-Bandwidth- Table	Drilldown top applications by bandwidth usage	traffic	
<pre>select appid, app, sum(bandwidth) as bandwidth from</pre>			

###(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
<pre>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</pre>		
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
as user_src, dstip, srcintf, dst 0)+coalesce(rcvdbyte, 0)) as ban group by appid, app, user_src, d	e(nullifna(`user`), nullifna(`unauthuser`), ig intf, policyid, count(*) as sessions, sum(coal dwidth from \$log where \$filter-exclude-var and stip, srcintf, dstintf, policyid order by sess tip is not null group by dstip having sum(band	lesce(sentbyte, d (logflag&1>0) sions desc)###

Dataset Name	Description	Log Category
drilldown-Top-User-By-Sessions-Table	Drilldown top user by session count	traffic
as user_src, dstip, srcintf, ds 0)+coalesce(rcvdbyte, 0)) as ba group by appid, app, user_src,	ce(nullifna(`user`), nullifna(`unauthus tintf, policyid, count(*) as sessions, ndwidth from \$log where \$filter-exclude dstip, srcintf, dstintf, policyid ordes ser_src is not null group by user_src o	<pre>sum(coalesce(sentbyte e-var and (logflag&1>0 r by sessions desc)###</pre>
Dataset Name	Description	Log Category
	Drilldown top user by session count	traffic
drilldown-Top-User-By-Sessions-Bar	Dillidowit top user by session count	uanic

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

```
Dataset Name
                                  Description
                                                                                   Log Category
 drilldown-Top-User-By-Bandwidth-
                                  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
<pre>src, hostname, count(*) as reque (logflag&l>0) and utmevent in (' 'script-filter') and hostname is desc)### union all ###(select co hostname, count(*) as requests f is null or logver>=502000000) and</pre>	(`user`), nullifna(`unauthuser`), ipstr(` sts from \$log-traffic where \$filter-exclu webfilter', 'banned-word', 'web-content', not null group by user_src, hostname orc alesce(nullifna(`user`), ipstr(`srcip`)) rom \$log-webfilter where \$filter-exclude- d hostname is not null group by user_src, er-drilldown and user_src is not null group	de-var and 'command-block', der by requests as user_src, -var and (eventtype hostname order by

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 \$filter-drilldown and user_src is not null group by user_src order by visits desc

Dataset Name	Description	Log Category
drilldown-Top-Website-By-Request- Table	Drilldown top website by request	traffic
src, hostname, count(*) as required (logflag&1>0) and utmevent in (a(`user`), nullifna(`unauthuser`), ip; ests from \$log-traffic where \$filter-o 'webfilter', 'banned-word', 'web-conto s not null group by user src, hostname	exclude-var and ent', 'command-block',

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>=50200000) and hostname is not null group by user_src, hostname order by requests desc)###) t where \$filter-drilldown and hostname is not null group by hostname order by visits desc

Dataset Name	Description	Log Category
drilldown-Top-Website-By-Request- Bar	Drilldown top website by request	traffic
<pre>src, hostname, count(*) as reque (logflag&1>0) and utmevent in (' 'script-filter') and hostname is desc)### union all ###(select council hostname, count(*) as requests for</pre>	(`user`), nullifna(`unauthuser`), ipstr(`srcip sts from \$log-traffic where \$filter-exclude-va webfilter', 'banned-word', 'web-content', 'com not null group by user_src, hostname order by alesce(nullifna(`user`), ipstr(`srcip`)) as us rom \$log-webfilter where \$filter-exclude-var a d hostname is not null group by user src, host	ar and mand-block', v requests ser_src, and (eventtype
requests desc)###) t where \$filte	er-drilldown and hostname is not null group by	hostname -

order by visits desc

Dataset Name	Description	Log Category
drilldown-Top-Email-Sender-By- Volume	Drilldown top email sender by volume	traffic
<pre>(rcvdbyte, 0)) as bandwidth fro and service in ('smtp', 'SMTP', utmevent in ('general-email-log desc)### union all ###(select ` (coalesce(sentbyte, 0)+coalesce \$filter-exclude-var and service '465/tcp') and eventtype is nul</pre>	<pre>t, count(*) as requests, sum(coalesce(ser m \$log-traffic where \$filter-exclude-var '25/tcp', '587/tcp', 'smtps', 'SMTPS', ' ', 'spamfilter') group by sender, recipie from` as sender, `to` as recipient, count (rcvdbyte, 0)) as bandwidth from \$log-ema in ('smtp', 'SMTP', '25/tcp', '587/tcp', l group by `from`, `to` order by requests not null group by sender having sum(band)</pre>	<pre>and (logflag&1>0) 465/tcp') and ent order by requests (*) as requests, sum ailfilter where 'smtps', 'SMTPS', s desc)###) t where</pre>
Dataset Name	Description	Log Category
drilldown-Top-Email-Send-Recipient-	Drilldown top email send recipient by volume	traffic

By-Volume

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

(

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

Dataset Name	Description	Log Category
drilldown-Top-Email-Sender-By-Count	Drilldown top email sender by count	traffic
select sender, sum(requests) as requests		

```
from
(
```

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

```
Dataset Name
                                  Description
                                                                                  Log Category
drilldown-Top-Email-Send-Recipient-
                                  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 $log-emailfilter where
$filter-exclude-var and service in ('smtp', 'SMTP', '25/tcp', '587/tcp', 'smtps', 'SMTPS',
'465/tcp') and eventtype is null group by `from`, `to` order by requests desc)###) t where
$filter-drilldown and recipient is not null group by recipient order by requests desc
Dataset Name
                                  Description
                                                                                  Log Category
```

Dataset Name	Description	Log Calegory
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
<pre>(rcvdbyte, 0)) as bandwidth from service in ('pop3', 'POP3', '110, '993/tcp', 'pop3s', 'POP3s', '999 group by recipient, sender order recipient, `from` as sender, cour (rcvdbyte, 0)) as bandwidth from ('pop3', 'POP3', '110/tcp', 'imag 'POP3s', '995/tcp') and eventtype</pre>	, count(*) as requests, sum(coalesce(sentby \$log where \$filter-exclude-var and (logfla /tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 5/tcp') and utmevent in ('general-email-log by requests desc)### union all ###(select nt(*) as requests, sum(coalesce(sentbyte, 0 \$log-emailfilter where \$filter-exclude-var p', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '9 e is null group by `to`, `from` order by re nder is not null group by sender having sum	<pre>g&1>0) and 'IMAPS', ', 'spamfilter') `to` as)+coalesce and service in 93/tcp', 'pop3s', quests desc)###)</pre>

Dataset Name	Description	Log Category
drilldown-Top-Email-Recipient-By- Count	Drilldown top email receiver by count	traffic
<pre>(rcvdbyte, 0)) as bandwidth from service in ('pop3', 'POP3', '110, '993/tcp', 'pop3s', 'POP3S', '995 group by recipient, sender order recipient, `from` as sender, cour</pre>	, count(*) as requests, sum(coalesce(sentbyte, \$log where \$filter-exclude-var and (logflag&1 /tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IM 5/tcp') and utmevent in ('general-email-log', by requests desc)### union all ###(select `to nt(*) as requests, sum(coalesce(sentbyte, 0)+c \$log-emailfilter where \$filter-exclude-var an	>0) and APS', 'spamfilter') ` as oalesce

FortiAnalyzer 6.2.8 Dataset Reference Fortinet Technologies Inc. ('pop3', 'POP3', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/tcp', 'pop3s', 'POP3S', '995/tcp') and eventtype is null group by `to`, `from` order by requests desc)###) t where \$filter-drilldown and recipient is not null group by recipient order by requests desc

Dataset Name Description Log Category drilldown-Top-Email-Receive-Sender-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 \$loq-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

Data	aser Name	Description	Log Calegory
drilld	lown-Top-Attack-Destination	Drilldown top attack dest	attack
seled	ct		

dstip, sum(totalnum) as totalnum from

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

Dataset Name	Description	Log Category
drilldown-Top-Attack-Source	Drilldown top attack source	attack
select srcip, sum(totalnum) as totalnum		

from

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

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

select

from_itime(itime) as timestamp,

attack, srcip, dstip

from

###(select itime, attack, srcip, dstip from \$log where \$filter-exclude-var order by itime
desc)### t where \$filter-drilldown order by timestamp desc

Dataset Name	Description	Log Category
drilldown-Top-Virus	UTM top virus	virus
<pre>select virus, max(virusid_s) as virusid, (case when virus like 'Risk else 'Virus' end) as malware_type, sum(totalnum) as totalnum</pre>	ware%' then 'Spyware' 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
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>=50200000) and nullifna(virus) is not null order by itime desc)### t where \$filter-drilldown order by timestamp desc

Dataset Name	Description	Log Category	
user-drilldown-Top-Blocked-Web- Sites-By-Requests	User drilldown top blocked web sites by requests	webfilter	
<pre>select hostname, sum(requests) as requests from ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, hostname, action, count (*) as requests from \$log where \$filter and hostname is not null group by user_src, hostname, action order by requests desc)### t where \$filter-drilldown and action='blocked' group by hostname order by requests desc</pre>			

Dataset Name	Description	Log Category		
user-drilldown-Top-Allowed-Web- Sites-By-Requests	User drilldown top allowed web sites by requests	webfilter		
<pre>select hostname, sum(requests) as requests from</pre>				
<pre>###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, hostname, action, count (*) as requests from \$log where \$filter and hostname is not null group by user_src, hostname, action order by requests desc)### t where \$filter-drilldown and action!='blocked' group by hostname order by requests desc</pre>				

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

select

```
catdesc,
sum(requests) as requests
```

from

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

Dataset Name	Description	Log Category		
user-drilldown-Top-Allowed-Web- Categories	User drilldown top allowed web categories	webfilter		
<pre>select catdesc, sum(requests) as requests from ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, catdesc, action, count (*) as requests from \$log where \$filter and catdesc is not null group by user_src, catdesc, action order by requests desc)### t where \$filter-drilldown and action!='blocked' group by catdesc order by requests desc</pre>				
Dataset Name	Description	Log Category		
user-drilldown-Top-Attacks	User drilldown top attacks by name	attack		
select attack, sum(attack_count) as attack_count				

from

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

Dataset Name	Description	Log Category
user-drilldown-Top-Attacks-High- Severity	User drilldown top attacks high severity	attack
select		
attack,		
<pre>sum(attack_count) as attack_c</pre>	count	
from		
###(select coalesce(nullifna	(`user`), ipstr(`srcip`)) as user_src, at	tack, (case when
severity in ('critical', 'high') then 1 else 0 end) as high_severity, c	count(*) as attack_
count from \$log where \$filter and nullifna(attack) is not null group by user_src, attack,		
high severity order by attack count desc)### t where \$filter-drilldown and high severity=1		
group by attack order by attack	count desc	

Dataset Name	Description	Log Category
user-drilldown-Top-Virus-By-Name	User drilldown top virus	virus
<pre>select virus, max(virusid_s) as virusid, sum(totalnum) as totalnum</pre>		
<pre>from ###(select coalesce(nullifna())</pre>	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>select receiver, sum(totalnum) as totalnum from ####(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, `to` as receiver, count (*) as totalnum from \$log where \$filter and subtype='infected' and (service in ('smtp', 'SMTP', '25/tcp', '587/tcp', 'smtps', 'SMTPS', '465/tcp') or service in ('pop3', 'POP3', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/tcp', 'pop3s', 'POP3s', '995/tcp')) and nullifna(virus) is not null group by user_src, receiver order by totalnum desc)### t where \$filter-drilldown group by receiver order by totalnum desc</pre>			
Dataset Name	Description	Log Category	
user-drilldown-Count-Spam-Activity- by-Hour-of-Day	User drilldown count spam activity by hour of day	emailfilter	
<pre>select \$hour_of_day(timestamp) as hou sum(totalnum) as totalnum from ###(select \$flex timestamp as</pre>	urstamp, timestamp, coalesce(nullifna(`user`), ipst	r(`srcip`)) as	

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

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

select

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

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

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

select

```
hourstamp,
cast(
   sum(cpu_usage) / sum(num) as decimal(6, 2)
) as cpu_avg_usage
from
```

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

Dataset Name	Description	Log Category
event-Usage-Memory	Event usage memory	event
<pre>select hourstamp, cast(</pre>		
<pre>sum(mem_usage)/ sum(num) as mem_avg_usage </pre>) as decimal(6, 2)	
<pre>from ###(select \$hour_of_day as hourstamp, sum(mem) as mem_usage, count(*) as num from \$log where \$filter and subtype='system' and action='perf-stats' group by hourstamp)### t group hourstamp order by hourstamp</pre>		

Dataset Name	Description	Log Category
event-Usage-Sessions	Event usage sessions	event
select hourstamp,		
cast(

```
sum(sess_usage) / sum(num) as decimal(10, 2)
) as sess_avg_usage
from
####(select $hour of day as hourstamp, sum(total
```

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

Dataset Name Description Log Category event-Usage-CPU-Sessions Event usage CPU sessions event select hourstamp, cast(sum(sess_usage) / sum(num) as decimal(10, 2)) as sess_avg_usage, cast(sum(cpu_usage) / sum(num) as decimal(6, 2)) as cpu_avg_usage from ###(select \$hour of day as hourstamp, sum(cpu) as cpu usage, sum(totalsession) as sess usage, count(*) as num from \$log where \$filter and subtype='system' and action='perf-stats'

group by hourstamp)### t group by hourstamp order by hourstamp

Dataset Name	Description	Log Category
App-Risk-Top-Users-By-Bandwidth	Top users by bandwidth usage	traffic
<pre>select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, srcip, sum(coalesce(sentbyte, 0)+ coal) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic out</pre>	lesce(rcvdbyte, 0)	
<pre>from \$log where \$filter and (logflag&1>0) and srcip is not null group by user_src, srcip having</pre>		

```
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

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`), 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</pre>		

sum(crscore % 65536)& gt; 0

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

		· · · · · · · · · · · · · · · · · · ·
App-Risk-Application-Usage-By- Category-With-Pie	Application risk application usage by category	traffic

Description

select

appcat, sum(bandwidth) as bandwidth from

###base(/*tag:rpt_base_t_top_app*/select devid, vd, csf, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, appid, app, appcat, apprisk, sum (coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, count(*) as sessions from \$log where \$filter and (logflag&1>0) and nullifna(app) is not null group by devid, vd, csf, user_ src, appid, app, appcat, apprisk order by sessions desc)base### t where 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
```

Log Category

from

###base(/*tag:rpt_base_t_top_app*/select devid, vd, csf, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, appid, app, appcat, apprisk, sum (coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, count(*) as sessions from \$log where \$filter and (logflag&1>0) and nullifna(app) is not null group by devid, vd, csf, user_ src, appid, app, appcat, apprisk order by sessions desc)base### t where nullifna(appcat) is not null group by appcat order by bandwidth desc

```
Dataset Name
                                   Description
                                                                                     Log Category
 Top-20-Categories-By-Bandwidth
                                   Webfilter categories by bandwidth usage
                                                                                     webfilter
select
 catdesc,
  sum (bandwidth) as bandwidth
from
  ###(select catdesc, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from
$log-traffic where $filter and (logflag&1>0) and (countweb>0 or ((logver is null or
logver<502000000) and (hostname is not null or utmevent in ('webfilter', 'banned-word',
'web-content', 'command-block', 'script-filter')))) and catdesc is not null group by catdesc
/*SkipSTART*/order by bandwidth desc/*SkipEND*/)### t group by catdesc order by bandwidth
desc
 Dataset Name
                                   Description
                                                                                     Log Category
App-Risk-Key-Applications-Crossing-
                                   Application risk application activity
                                                                                     traffic
 The-Network
select
  app group name(app) as app group,
  appcat,
  sum(
    coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
  ) as bandwidth,
  count(*) as num session
from
  $10a
where
 Sfilter
  and (
    logflag&1>0
  )
  and nullifna(app) is not null
group by
  app_group,
  appcat
order by
  bandwidth desc
```

Dataset NameDescriptionLog CategoryApp-Risk-Applications-Running-Over-
HTTPApplication risk applications running over HTTPtraffic

select

app_group_name(app) as app_group,

```
service,
  count(*) as sessions,
  sum(
   coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
  ) as bandwidth
from
  $log
where
  $filter
  and (
   logflag&1>0
  )
  and nullifna(app) is not null
  and service in (
   '80/tcp', '443/tcp', 'HTTP', 'HTTPS',
   'http', 'https'
  )
group by
  app_group,
  service
having
  sum(
    coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
  )& gt; 0
order by
  bandwidth desc
```

Dataset Name	Description	Log Category
App-Risk-Top-Web-Sites-Visited-By- Network-Users-Pie-Cha	Application risk web browsing summary category	traffic
as bandwidth from \$log-traffic is null or logver<502000000) an	as num_sess, sum(coalesce(sentbyte, 0)+coale where \$filter and (logflag&1>0) and (countwe d (hostname is not null or utmevent in ('web block', 'script-filter'))) and catdesc is n	b>0 or ((logver filter', 'banned-
	### t group by catdesc order by num_sess des	<i>y</i> 1 1
	· · · · · · · · · · · · · · · · · · ·	
catdesc order by num_sess desc)	### t group by catdesc order by num_sess des	sc

word', 'web-content', 'command-block', 'script-filter')))) and catdesc is not null group by catdesc order by num_sess desc)### t group by catdesc order by num_sess desc

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

```
domain,
catdesc,
sum(visits) as visits
from
```

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

Dataset Name	Description	Log Category
Top-Destination-Countries-By- Browsing-Time	Traffic top destination countries by browsing time	traffic
<pre>bandwidth, sum(traffic_in) as tr dstcountry, ebtr_agg_flat(\$brows (rcvdbyte, 0)) as bandwidth, sum (sentbyte, 0)) as traffic_out fr not null group by dstcountry) t</pre>	t g_flat(browsetime) as browsetime, sum(bandwidt affic_in, sum(traffic_out) as traffic_out from e_time) as browsetime, sum(coalesce(sentbyte, (coalesce(rcvdbyte, 0)) as traffic_in, sum(coa om \$log where \$filter and (logflag&1>0) and \$b group by dstcountry /*SkipSTART*/order by ebtr) desc/*SkipEND*/)### t group by dstcountry or	(select 0)+coalesce lesce rowse_time is _value(ebtr_

Dataset Name	Description	Log Category
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,</pre>		

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

Dataset Name	Description	Log Category
App-Risk-Traffic-Top-Hostnames-By- Browsing-Time	Traffic top domains by browsing time	traffic
<pre>sum(traffic_in) as traffic_in, s agg_flat(\$browse_time) as browse bandwidth, sum(coalesce(rcvdbyte out from \$log where \$filter and not null group by hostname) t gr</pre>		ct hostname, ebtr_ rcvdbyte, 0)) as e, 0)) as traffic_ d \$browse_time is r_value(ebtr_agg_
Dataset Name	Description	Log Category
App-Risk-Traffic-Top-Hostnames-By- Browsing-Time-Enhanced	Traffic top domains by browsing time enhanced	traffic
select		

hostname, ebtr_value(ebtr_agg_flat(browsetime), null, \$timespan) as browsetime, sum(bandwidth) as bandwidth, sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_out from ###(select hostname, ebtr_agg_flat(browsetime) as browsetime, sum(bandwidth) as bandwidth, sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_out from (select hostname, ebtr_ agg flat(\$browse time) as browsetime, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_ out from \$log where \$filter and (logflag&1>0) and hostname is not null and \$browse_time is not null group by hostname) t group by hostname /*SkipSTART*/order by ebtr_value(ebtr_agg_ flat(browsetime), null, null) desc/*SkipEND*/)### t group by hostname order by browsetime desc

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

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

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 nu group by attack, severity, ref</pre>	11	

order by totalnum desc

Dataset Name	Description	Log Category
App-Risk-Top-Info-Threat-Vectors- Crossing-The-Network	Application risk top info threat vectors	attack
<pre>select attack, severity, ref, count(*) as totalnum from \$log where \$filter and severity = 'info' and nullifna(attack) is not nu</pre>	111	
group by attack, severity, ref order by totalnum desc		

Dataset Name	Description	Log Category
App-Risk-Top-Virus-By-Name	UTM top virus	virus
select		
virus,		
<pre>max(virusid_s) as virusid,</pre>		
(always & I then I Conversel when with	rus like 'Adware%' then 'Adware'
else 'Virus' end	skwares chen spyware when vi	TUS IIKE Adwares chen Adware
) as malware type,		
sum(totalnum) as totalnum		
from		
	to_str(virusid, eventtype) as v	
-		02000000) and nullifna(virus) is
group by virus, malware type	usid_s /*SkipSTART*/order by tot order by totalnum desc	LAINUM desc/^skipEND^/)### l
<u></u>		
Dataset Name	Description	Log Category
App-Risk-Top-Virus-Victim	UTM top virus user	virus
select		
user_src, sum(totalnum) as totalnum		
from		
	na(`user`), ipstr(`srcip`)) as u	user src, count(*) as totalnum
		02000000) and nullifna(virus) is

from \$log where \$filter and (eventtype is null or logver>=502000000) and nullifna(virus) is
not null group by user_src /*SkipSTART*/order by totalnum desc/*SkipEND*/)### t group by
user_src order by totalnum desc

Dataset Name	Description	Log Category
App-Risk-Data-Loss-Prevention-Type- Events	Application risk DLP UTM event	dlp
<pre>select subtype : :text as utmsubtype, count(*) as number from</pre>		
<pre>subtype, srcip, dstip, severity, severity='critical' then 'Critic (`user`), ipstr(`srcip`)) is not as data_loss from \$log where \$fi</pre>	com` as sender, `to` as receiver, prof filename, direction, filesize, (case cal Data Exfiltration' else (case when c null then 'User Associated Data Loss Elter /*SkipSTART*/order by itime desc s not null group by subtype order by r	e when n coalesce(nullifna s' else NULL end) end) c/*SkipEND*/)### t where

Dataset Name	Description	Log Category
App-Risk-Vulnerability-Discovered	Application risk vulnerability discovered	netscan
select		
vuln,		
vulnref as ref,		
vulncat,		
severity,		
count(*) as totalnum		
from		
\$log		
where		
\$filter		
and vuln is not null		
group by		
vuln,		
vulnref,		
vulncat,		
severity		
order by		
totalnum desc		
Dataset Name	Description	Log Category

Dataset Name	Description	Log Category
App-Risk-Malware-Discovered	Application risk virus discovered	virus
	s dom, count(*) as totalnum from \$log where \$ httype is null or logver>=502000000) group by dom order by totalnum desc	
Dataset Name	Description	Log Category
App-Risk-Breakdown-Of-Risk- Applications	Application risk breakdown of risk applications	traffic

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

Dataset Name	Description	Log Category
App-Risk-Number-Of-Applications-By- Risk-Behavior	Application risk number of applications by risk behavior	traffic
<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. where \$filter and (logflag&1>0) group by risk, f_behavior order by risk desc, number desc</pre>		
Dataset Name	Description	Log Category

Dataset Name	Description	Log Calegory
App-Risk-High-Risk-Application	Application risk high risk application	traffic
<pre>select risk as d_risk, behavior as d_behavior, t2.id, t2.name, t2.app_cat, t2.technology, sum(</pre>		

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

Dataset Name Description Log Category Apprisk-Ctrl-Severe-High-Risktraffic Severe and high risk applications Application select appcat, count(distinct app) as total num from ###(select app, appcat, apprisk, sum(sessions) as sessions from ###base(/*tag:rpt_base_t_ top app*/select devid, vd, csf, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr (`srcip`)) as user src, appid, app, appcat, apprisk, sum(coalesce(sentbyte, 0)+coalesce (rcvdbyte, 0)) as bandwidth, count(*) as sessions from \$log where \$filter and (logflag&1>0) and nullifna(app) is not null group by devid, vd, csf, user_src, appid, app, appcat, apprisk order by sessions desc)base### t group by app, appcat, apprisk order by sessions desc)### t where 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 \$logwebfilter 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
when t1.severity='high' then 4 w 2 when t1.severity='info' then 2 cotalnum from \$log t1 left join	ces, vuln_type, t2.cve, (case when t1.sever: when t1.severity='medium' then 3 when t l else 0 end) as severity_number, dstip (select name, cve, vuln_type from ips	t1.severity='low' then p, srcip, count(*) as _mdata) t2 on
attack, attackid, vuln_type, sev lesc	_type, t2.cve, t1.severity, dstip, src: verity_number, cve order by severity_nu	ip)### t group by umber desc, totalnum
rroup by attack, attackid, vuln_ ttack, attackid, vuln_type, sev lesc Dataset Name Apprisk-Ctrl-Breakdown-Of-High-Risk-	_type, t2.cve, t1.severity, dstip, src: verity_number, cve order by severity_nu Description	ip)### t group by umber desc, totalnum
<pre>proup by attack, attackid, vuln_ ttack, attackid, vuln_type, set lesc Dataset Name Apprisk-Ctrl-Breakdown-Of-High-Risk- Application select appcat, count(distinct app) as total_r from ###(select app, appcat, appris cop_app*/select devid, vd, csf, (`srcip`)) as user_src, appid, a frowdbyte, 0)) as bandwidth, cou and nullifna(app) is not null ge order by sessions desc)base### to attack, attack, atta</pre>	_type, t2.cve, t1.severity, dstip, src: verity_number, cve order by severity_nu Description Severe and high risk applications	<pre>ip)### t group by umber desc, totalnum Log Category traffic base(/*tag:rpt_base_t_ unauthuser`), ipstr tbyte, 0)+coalesce lter and (logflag&1>0) d, app, appcat, appris! by sessions desc)### t</pre>
<pre>proup by attack, attackid, vuln_ tttack, attackid, vuln_type, set lesc Dataset Name Apprisk-Ctrl-Breakdown-Of-High-Risk- Application select appcat, count(distinct app) as total_r from ###(select app, appcat, appris cop_app*/select devid, vd, csf, [`srcip`)) as user_src, appid, a frevdbyte, 0)) as bandwidth, cou and nullifna(app) is not null gr order by sessions desc)base### to here nullifna(appcat) is not null</pre>	<pre>_type, t2.cve, t1.severity, dstip, src: zerity_number, cve order by severity_number, Description Severe and high risk applications hum sk, sum(sessions) as sessions from #### coalesce(nullifna(`user`), nullifna(`u app, appcat, apprisk, sum(coalesce(sent unt(*) as sessions from \$log where \$fil roup by devid, vd, csf, user_src, appic t group by app, appcat, apprisk order b</pre>	<pre>ip)### t group by umber desc, totalnum Log Category traffic base(/*tag:rpt_base_t_ unauthuser`), ipstr tbyte, 0)+coalesce lter and (logflag&1>0) d, app, appcat, apprish by sessions desc)### t</pre>

```
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>(appcat)='remote.access' then 't 'video/audio') then 'bandwidth-c lower(appcat)='proxy' then 'prox (/*tag:rpt_base_t_top_app*/selec (`unauthuser`), ipstr(`srcip`)) (sentbyte, 0)+coalesce(rcvdbyte, \$filter and (logflag&1>0) and nu appid, app, appcat, apprisk orde 'remote.access', 'storage.backup ('critical', 'high') group by ap 'malicious' as behavior, count()</pre>	appcat)='botnet' then 'malicious' when unneling' when lower(appcat) in ('stor onsuming' when lower(appcat)='p2p' the y' end) as behavior, sum(sessions) as t devid, vd, csf, coalesce(nullifna(`u as user_src, appid, app, appcat, appri 0)) as bandwidth, count(*) as session llifna(app) is not null group by devid r by sessions desc)base### where lower ', 'video/audio', 'p2p', 'proxy') and pcat order by total_num desc)### union *) as total_num from \$log-attack where 'critical', 'high') group by behavior) c	<pre>age.backup', n 'peer-to-peer' when total_num from ###base ser`), nullifna sk, sum(coalesce s from \$log where , vd, csf, user_src, (appcat) in ('botnet', apprisk in all ###(select \$filter and</pre>

Dataset Name	Description	Log Category
Apprisk-Ctrl-Key-Application-Crossing- The-Network	Key Application Crossing The Network	traffic
<pre>user_src, sum(coalesce(sentbyte, sessions from \$log where \$filter</pre>	ers, fna(`user`), nullifna(`unauthuser`), ips 0)+coalesce(rcvdbyte, 0)) as bandwidth, and (logflag&1>0) group by app, user_sr t2 on t1.app=t2.name group by id, app,	count(*) as c order by bandwidth

Dataset Name	Description	Log Category
Apprisk-Ctrl-Risk-Application-Usage- By-Category-With-Pie	Application risk application usage by category	traffic
<pre>select appcat, sum(bandwidth) as bandwidth from</pre>		
<pre>###base(/*tag:rpt_base_t_top_a nullifna(`unauthuser`), ipstr(`s (coalesce(sentbyte, 0)+coalesce where \$filter and (logflag&1>0)</pre>	<pre>app*/select devid, vd, csf, coalesce(nullifn srcip`)) as user_src, appid, app, appcat, ap (rcvdbyte, 0)) as bandwidth, count(*) as ses and nullifna(app) is not null group by devi k order by sessions desc)base### t where nul by bandwidth desc</pre>	oprisk, sum ssions from \$log ld, vd, csf, user_

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

select

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

###(select app, appcat, user_src, sum(bandwidth) as bandwidth, sum(sessions) as sessions from ###base(/*tag:rpt_base_t_top_app*/select devid, vd, csf, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, appid, app, appcat, apprisk, sum (coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, count(*) as sessions from \$log where \$filter and (logflag&l>0) and nullifna(app) is not null group by devid, vd, csf, user_ src, appid, app, appcat, apprisk order by sessions desc)base### t where nullifna(appcat) is not null group by app, appcat, user_src order by bandwidth desc)### t group by appcat order by bandwidth desc

Dataset Name	Description	Log Category
Apprisk-Ctrl-Top-Web-Applications-by- Bandwidth	Top 25 Web Categories by Bandwidtih	traffic
<pre>select d_risk, id, name, technology, count(distinct f_user) as user sum(bandwidth) as bandwidth, sum(num session) as num session</pre>		
<pre>sum(num_session) as num_session from ###(select risk as d_risk, t2.id, t2.name, t2.technology, coalesce(nullifna(t1.`user`), nullifna(t1.`unauthuser`), ipstr(t1.`srcip`)) as f_user, sum(coalesce(sentbyte, 0)+coalesce (rcvdbyte, 0)) as bandwidth, count(*) as num_session from \$log t1 inner join app_mdata t2 on t1.appid=t2.id where \$filter and (logflag&1>0) and nullifna(app) is not null and service in</pre>		

('80/tcp', '443/tcp', 'HTTP', 'HTTPS', 'http', 'https') group by risk, t2.id, t2.name, t2.technology, f_user)### t group by d_risk, id, name, technology order by bandwidth desc

Dataset Name	Description	Log Category
Apprisk-Ctrl-Top-Web-Categories- Visited	Top 25 Web Categories Visited	traffic
<pre>select catdesc, count(distinct f_user) as use sum(sessions) as sessions, sum(bandwidth) as bandwidth</pre>	er_num,	

from

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

Dataset Name	Description	Log Category
Apprisk-Ctrl-Common-Virus-Botnet- Spyware	Common virus disvocered, the botnet communictions and the spyware/adware	traffic
<pre>case when virus_s like 'Ris 'Adware' else 'Virus' end) end) as malware_type, appid, app, count(distinct dstip) as victin count(distinct srcip) as source sum(total_num) as total_num from (</pre>	e,	
<pre>appid, app, count(distinct dstip) as victin count(distinct srcip) as source sum(total_num) as total_num from (</pre>		otal_num from

###(select app as virus_s, appcat, appld, app, dstip, srcip, count(*) as total_num from \$log-traffic where \$filter and (logflag&1>0) and lower(appcat)='botnet' group by virus_s, appcat, appid, dstip, srcip, app order by total_num desc)### union all ###(select unnest (string_to_array(virus, ',')) as virus_s, appcat, appid, app, dstip, srcip, count(*) as total_num from \$log-traffic where \$filter and (logflag&1>0) and virus is not null group by virus_s, appcat, appid, dstip, srcip, app order by total_num desc)### union all ###(select attack as virus_s, 'botnet' as appcat, 0 as appid, attack as app, dstip, srcip, count(*) as total_num from \$log-attack where \$filter and (logflag&16>0) group by virus_s, appcat, appid, dstip, srcip, app order by total_num desc)###) t group by virus, appid, app, malware_type order by total num desc

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

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

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

```
source desc
```

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

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

Dataset Name	Description	Log Category
appctrl-Top-Blocked-SCCP-Callers	Appctrl top blocked SCCP callers	app-ctrl
<pre>select srcname as caller, count(*) as totalnum from \$log where \$filter and lower(appcat)= 'voip' and app = 'sccp'</pre>		
<pre>and action = 'block' and srcname is not null group by caller order by totalnum desc</pre>		

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</pre>		

order by totalnum desc

Dataset Name	Description	Log Category
security-Top20-High-Risk-Application- In-Use	High risk application in use	traffic
<pre>(t1.`srcip`)) as f_user, t2.name 0)+coalesce(rcvdbyte, 0)) as ban mdata t2 on t1.appid=t2.id where</pre>	<pre>elesce(nullifna(t1.`user`), nullifna(t1. e, t2.app_cat, t2.technology, sum(coales adwidth, count(*) as sessions from \$log e \$filter and risk>='4' and (logflag&1>0 ogy, risk)### t group by d_risk, name, ag</pre>	ce(sentbyte, t1 inner join app_) group by f_user,
Dataset Name	Description	Log Category
security-High-Risk-Application-By- Category	High risk application by category	traffic

select

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

security-Top10-Application- Application risk application usage by category tra	
Categories-By-Bandwidth	ffic

select
 appcat,
 sum(bandwidth) as bandwidth
from

###base(/*tag:rpt_base_t_top_app*/select devid, vd, csf, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, appid, app, appcat, apprisk, sum (coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, count(*) as sessions from \$log where \$filter and (logflag&1>0) and nullifna(app) is not null group by devid, vd, csf, user_ src, appid, app, appcat, apprisk order by sessions desc)base### t where 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
<pre>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</pre>		
<pre>from ###base(/*tag:rpt_base_t_t</pre>	_src, sum(bandwidth) as bandwidth, sum(sessi cop_app*/select devid, vd, csf, coalesce(nul. `srcip`)) as user_src, appid, app, appcat, ap	lifna(`user`),

nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, appid, app, appcat, apprisk, sum (coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, count(*) as sessions from \$log where \$filter and (logflag&1>0) and nullifna(app) is not null group by devid, vd, csf, user_ src, appid, app, appcat, apprisk order by sessions desc)base### t where nullifna(appcat) is not null group by app, appcat, user_src order by bandwidth desc)### t group by appcat order by bandwidth desc

Dataset Name	Description	Log Category
security-Top25-Web-Applications-By- Bandwidth	Top Web Applications by Bandwidtih	traffic
<pre>(t1.`user`), nullifna(t1.`unauth 0)+coalesce(rcvdbyte, 0)) as ban mdata t2 on t1.appid=t2.id where service in ('80/tcp', '443/tcp',</pre>	s, app_cat, t2.name, t2.technology, coalesce(n user`), ipstr(t1.`srcip`)) as f_user, sum(d dwidth, count(*) as num_session from \$log f \$filter and (logflag&1>0) and nullifna(app 'HTTP', 'HTTPS', 'http', 'https') group by ser)### t group by d_risk, name, app_cat, f	<pre>coalesce(sentbyte, t1 inner join app_ p) is not null and y risk, t2.app_</pre>
Dataset Name	Description	Log Category

Security-Top25-Web-Categories- Visited	Top 25 Web Categories Visited	traffic
<pre>select catdesc, count(distinct f_user) as user sum(sessions) as sessions, sum(bandwidth) as bandwidth from ####(select catdesc, coalesce(r</pre>	_num, nullifna(`user`), nullifna(`unau	thuser`), 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

Dataset Name	Description	Log Category
security-Top25-Malware-Virus-Botnet- Spyware	Malware: viruses, Bots, Spyware/Adware	traffic
		<e 'adware%'="" td="" then<=""></e>
###(select app as virus_s, app as virus_s, app	<pre>opcat, dstip, srcip, count(*) as total_r</pre>	num from \$log-traffic

where \$filter and (logflag&l>0) and lower(appcat)='botnet' group by virus_s, appcat, dstip, srcip order by total_num desc)### union all ###(select unnest(string_to_array(virus, ',')) as virus_s, appcat, dstip, srcip, count(*) as total_num from \$log-traffic where \$filter and (logflag&l>0) and virus is not null group by virus_s, appcat, dstip, srcip order by total_ num desc)### union all ###(select attack as virus_s, 'null' as appcat, dstip, srcip, count (*) as total_num from \$log-attack where \$filter and (logflag&l6>0) group by virus_s, appcat, dstip, srcip order by total_num desc)###) t group by virus, malware_type order by total_num desc

Dataset Name	Description	Log Category
security-Top10-Malware-Virus- Spyware	Malware: viruses, Spyware/Adware	virus
when virus like 'Riskware%' t 'Virus' end) as malware_type	<pre>burce, bo_str(virusid, eventtype) as virusid_s, bhen 'Spyware' when virus like 'Adware%' c, count(*) as total_num from \$log where virus, virusid_s, srcip, dstip order by</pre>	then 'Adware' else \$filter and nullifna
Dataset Name	Description	Log Category

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 as app, 0 as appid, cast('Botnet C&C' as char(32)) as malware_type, srcip, dstip, count(*) as total_num from \$log-attack where \$filter and (logflag&16>0) group by app, appid, malware_type, srcip, dstip order by total_num desc)###) t group by app, appid, malware_type order by total_num desc

Dataset Name	Description	Log Category
security-Top10-Victims-of-Malware	Victims of Malware	virus
<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 total_num desc</pre>		
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) '://' 1) as phishing_site, count(*) as total_num from</pre>	nostname url	

```
$log
where
  $filter
  and lower(service) in ('http', 'https')
  and hostname is not null
  and cat in (26, 61)
group by
  user_src,
  phishing_site
order by
  total num desc
```

Dataset Name	Description	Log Category
security-Top25-Malicious-Phishing- Sites	Malicious Phishing Site	webfilter
count(*) as total from \$log whe	rce, '://' hostname url) as phishi ere \$filter and lower(service) in ('h n (26, 61) group by phishing_site, ds	ttp', 'https') and
Dataset Name	Description	Log Category
security-Application-Vulnerability	Application vulnerabilities discovered	attack
<pre>select attack, attackid, vuln_type, cve, severity_number, count(distinct dstip) as vict</pre>	ims,	

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

###(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

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

totalnum from \$log t1 left join (select name, cve, vuln_type from ips_mdata) t2 on

count(distinct srcip) as sources,

sum(totalnum) as totalnum

from

desc

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

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

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

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

Dataset Name	Description	Log Category	
security-Data-Loss-Incidents-By- Severity	Data loss incidents summary by severity	dlp	
<pre>select initcap(severity : :text) as s_severity, count(*) as total_num from ###(select itime, hostname,`from` as sender, `to` as receiver, profile, action, service, subtype, srcip, dstip, severity, filename, direction, filesize, (case when severity='critical' then 'Critical Data Exfiltration' else (case when coalesce(nullifna (`user`), ipstr(`srcip`)) is not null then 'User Associated Data Loss' else NULL end) end) as data_loss from \$log where \$filter /*SkipSTART*/order by itime desc/*SkipEND*/)### t where \$filter-drilldown and severity is not null group by s_severity order by total_num desc</pre>			
Dataset Name	Description	Log Category	
security-Data-Loss-Files-By-Service	Data Lass Files By Service	dlp	
<pre>select filename, (case direction when 'incomi </pre>	ing then Developed then lettering		

```
max(filesize) as filesize,
```

```
service
```

from

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

Dataset Name	Description	Log Category
security-Endpoint-Security-Events- Summary	Endpoint Security Events summary	fct-traffic
'Malicious/phishing websites' wh	s' then 'Malware incidents' when 'webfilte en 'appfirewall' then 'Risk applications' can' then 'Vulnerability detected' else 'O	when 'dlp' then
Dataset Name	Description	Log Category
security-Top-Endpoing-Running-High- Risk-Application	Endpoints Running High Risk Application	fct-traffic

```
select
 coalesce(
   nullifna(`user`),
    ipstr(`srcip`),
    'Unknown'
  ) as f_user,
  coalesce(
   nullifna(hostname),
   'Unknown'
  ) as host name,
  threat as app,
  t2.app cat as appcat,
  risk as d risk
from
  $log t1
  inner join app mdata t2 on t1.threat = t2.name
where
  $filter
  and utmevent = 'appfirewall'
```

```
and risk & gt;= '4'
group by
f_user,
host_name,
t1.threat,
t2.app_cat,
t2.risk
order by
risk desc
```

Dataset Name	Description	Log Category
security-Top-Endpoints-Infected-With- Malware	Endpoints Infected With Malware	fct-event
<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 clientfeature = '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
select		
f_user,		
host_name,		
remotename,		
sum(total_num) as total_num		
from		
###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as f use	r, coalesce(nullifna
(hostname), 'Unknown') as host	name, remotename, count(*) as tot	al num from \$log where
\$filter and utmevent='webfilter	' and remotename is not null and rder by total_num desc)### t grou	utmaction='blocked' group by

Dataset Name	Description	Log Category
security-Top-Endpoints-With-Data- Loss-Incidents	Endpoints With Data Loss Incidents	fct-event
<pre>(nullifna(hostname), 'Unknown')</pre>	`user`), ipstr(`deviceip`), 'Unknown') a as host_name, msg, count(*) as total_nu group by f_user, host_name, msg order k usg order by total_num desc	um from \$log where
Dataset Name	Description	Log Category
content-Count-Total-SCCP-Call- Registrations-by-Hour-of-Day	Content count total SCCP call registrations by ho	ur of day content

select

```
$hour_of_day as hourstamp,
   count(*) as totalnum
from
   $log
where
   $filter
   and proto = 'sccp'
   and kind = 'register'
group by
   hourstamp
order by
   hourstamp
```

Dataset Name	Description	Log Category
content-Count-Total-SCCP-Calls- Duration-by-Hour-of-Day	Content count total SCCP calls duration by hour of day	content
<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 hourstamp order by hourstamp</pre>		

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'</pre>		

```
and kind = 'call'
group by
status
```

order by totalnum desc

Dataset Name	Description	Log Category
content-Dist-Total-SIP-Calls-by- Duration	Content dist total SIP calls by duration	content
	LESS_ONE_MIN' when duration<600 then JR' when duration & gt;= 3600 then 'M	
Dataset Name	Description	Log Category
Botnet-Activity-By-Sources	Botnet activity by sources	traffic
	apprisk, srcip, dstip, coalesce(null as user src, count(*) as totalnum fro	

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

Dataset Name	Description	Log Category
Botnet-Infected-Hosts	Botnet infected hosts	traffic
select		
user src,		
host_mac,		
sum(events) as events		
from		
(
###(select coalesce(n	ullifna(`user`), nullifna(`unauthuser`),	ipstr(`srcip`)) as user_
<pre>src, get_devtype(srcswver</pre>	sion, osname, devtype) as devtype_new, c	coalesce(srcname, srcmac) a
host_mac, count(*) as eve	nts from \$log-traffic where \$filter and	(logflag&1>0) and
apprat='Botnet' group by	user src devtype new host mac order by	vevents desc)### union all

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

Dataset Name	Description	Log Category
Detected-Botnet	Detected botnet	traffic

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

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

select
 dstip,
 domain,
 sum(events) as events
from

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

Dataset Name	Description	Log Category
Botnet-Victims	Botnet victims	traffic
select		
user src,		
sum(events) as events	5	
from		
(
(
select		
user_src,		
sum(totalnum) a	is events	
from		
###(select app,	appcat, apprisk, srcip, dstip, coalesc	e(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`	<pre>srcip`)) as user_src, count(*) as total</pre>	num from \$log-traffic where
\$filter and (logflag&1>	0) and appcat='Botnet' and nullifna(app) is not null group by app,
appcat, apprisk, srcip,	dstip, user_src order by totalnum desc)### t group by user_src)
union all (select user_	<pre>src, sum(totalnum) as events from ###(s</pre>	elect attack, coalesce

(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, \$flex_timestamp as timestamp, hostname, severity, crlevel, eventtype, service, dstip, srcip, count(*) as totalnum from \$log-attack where \$filter and (logflag&16>0) group by attack, user_src, timestamp, hostname, severity, crlevel, eventtype, service, dstip, srcip order by timestamp desc)### t group by user_src)) t group by user_src order by events desc

Dataset Name	Description	Log Category
Botnet-Timeline	Botnet timeline	traffic
<pre>select \$flex_datetime(timestamp) a sum(events) as events from</pre>	s hodex,	

(

###(select \$flex_timestamp as timestamp, count(*) as events from \$log-traffic where \$filter and (logflag&1>0) and appcat='Botnet' group by timestamp order by timestamp desc)### union all ###(select \$flex_timestamp as timestamp, count(*) as events from \$log-dns where \$filter and (botnetdomain is not null or botnetip is not null) group by timestamp order by timestamp)### union all ###(select \$flex_timestamp as timestamp, count(*) as events from \$log-attack where \$filter and (logflag&16>0) group by timestamp order by timestamp)###) t group by hodex order by hodex

Dataset Name	Description	Log Category
Application-Session-History	Application session history	traffic

select

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

from

###(select \$flex_timestamp as timestamp, count(*) as counter from \$log where \$filter and (logflag&1>0) group by timestamp order by timestamp desc)### t group by hodex order by hodex

Dataset Name	Description	Log Category
Application-Usage-List	Detailed application usage	traffic
<pre>select appid, app, appcat, (case when (utmaction in ('block', or action = 'deny') then 'Blocked' else 'Al) as custaction, sum(coalesce(sentbyte, 0)+ co) as bandwidth, count(*) as num_session from \$log where \$filter and (logflag&l>0) and nullifna(app) is not nu and policyid != 0 group by appid, app, appcat, custaction order by bandwidth desc</pre>	llowed' end balesce(rcvdbyte, 0)	
Dataset Name	Description	Log Category
PCI-DSS-Compliance-Summary	PCI DSS Compliance Summary	event
<pre>select status, num_reason as requirements, cast(num_reason * 100.0 /(</pre>		

```
sum(num_reason) over()
```

```
) as decimal(18, 2)
 ) as percent
from
  (
   select
     (
       case when fail count>0 then 'Non-Compliant' else 'Compliant' end
     ) as status,
     count(distinct reason) as num reason
   from
      (
        select
          ftnt_pci_id,
          (
           sum(fail count) over (partition by ftnt pci id)
          ) as fail count,
          reason
        from
          ###(select ftnt_pci_id, (case when result='fail' then 1 else 0 end) as fail_count,
reason from $log t1 inner join pci_dss_mdata t2 on t1.reason=t2.ftnt_id where $filter and
```

subtype='compliance-check' group by ftnt_pci_id, result, reason)### t) t group by status) t order by status

Dataset Name	Description	Log Category
PCI-DSS-Non-Compliant- Requirements-By-Severity	PCI DSS Non-Compliant Requirements by Severity	event
<pre>with query as (select * from (select ftnt_pci_id, severity, (sum(fail_count) over) as fail_count, reason from from</pre>	<pre>(partition by ftnt_pci_id)</pre>	
<pre>###(select ftnt_pci_id fail_count, reason from \$log t \$filter and subtype='complianc t) t where fail_count>0) selec (select distinct on (1) reason</pre>	<pre>1, t2.severity, (case when result='fail' then 1 1 inner join pci_dss_mdata t2 on t1.reason=t2.se-check' group by ftnt_pci_id, t2.severity, result t.severity, count(distinct t.reason) as require, severity from query order by reason, (case local' then 3 when 'medium' then 2 when 'low' the der by requirements desc</pre>	<pre>ftnt_id where sult, reason)### irements from ower(severity)</pre>

Dataset Name	Description	Log Category
PCI-DSS-Compliant-Requirements- By-Severity	PCI DSS Compliant Requirements by Severity	event
with query as (select		

```
*
from
 (
    select
    ftnt_pci_id,
    severity,
    (
        sum(fail_count) over (partition by ftnt_pci_id)
    ) as fail_count,
    reason
    from
```

###(select ftnt_pci_id, t2.severity, (case when result='fail' then 1 else 0 end) as fail_count, reason from \$log t1 inner join pci_dss_mdata t2 on t1.reason=t2.ftnt_id where \$filter and subtype='compliance-check' group by ftnt_pci_id, t2.severity, result, reason)### t) t where fail_count=0) select t.severity, count(distinct t.reason) as requirements from (select distinct on (1) reason, severity from query order by reason, (case lower(severity) when 'high' then 4 when 'critical' then 3 when 'medium' then 2 when 'low' then 1 else 0 end) desc) t group by t.severity order by requirements desc

Dataset Name	Description	Log Category
PCI-DSS-Fortinet-Security-Best- Practice-Summary	PCI DSS Fortinet Security Best Practice Summary	event
) as status, count(distinct reason) as from ###(select result, reason	' then 'Failed' else 'Passed' end num_reason from \$log where \$filter and subtype='complianc by result, reason)### t group by status) t ord	

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)</pre>		

```
) as percent
from
 (
    select
    initcap(status) as status,
    count(distinct reason) as num_reason
    from
        ###(select status, reason from $log where $filter and subtype='compliance-check' and
result='fail' group by status, reason)### t group by status) t order by status
```

Dataset Name	Description	Log Category
PCI-DSS-Passed-Fortinet-Security- Best-Practices-By-Severity	PCI DSS Passed Fortinet Security Best Practices by Severity	event
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, 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 \$log where \$filter and subtype = 'compliance-ch group by reason, status, module, msg order by ftnt_id</pre>	neck'	
Dataset Name	Description	Log Category
DLP-Email-Activity-Details	Email DLP Violations Summary	dlp
<pre>select from_itime(itime) as timesta sender, receiver, regexp_replace(filename, '.' filesize,</pre>		

```
profile,
action,
direction
```

from

###(select itime, hostname,`from` as sender, `to` as receiver, profile, action, service, subtype, srcip, dstip, severity, filename, direction, filesize, (case when severity='critical' then 'Critical Data Exfiltration' else (case when coalesce(nullifna (`user`), ipstr(`srcip`)) is not null then 'User Associated Data Loss' else NULL end) end) as data_loss from \$log where \$filter /*SkipSTART*/order by itime desc/*SkipEND*/)### t where \$filter-drilldown and (service in ('smtp', 'SMTP', '25/tcp', '587/tcp', 'smtps', 'SMTPS', '465/tcp') or service in ('pop3', 'POP3', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/tcp', 'pop3s', 'POP3S', '995/tcp')) order by timestamp desc

Dataset Name	Description	Log Category
Email-DLP-Chart	Email DLP Activity Summary	dlp
<pre>select profile, count(*) as total_num</pre>		
	ame,`from` as sender, `to` as receiver, pr verity, filename, direction, filesize, (ca	

severity='critical' then 'Critical Data Exfiltration' else (case when coalesce(nullifna
(`user`), ipstr(`srcip`)) is not null then 'User Associated Data Loss' else NULL end) end)
as data_loss from \$log where \$filter /*SkipSTART*/order by itime desc/*SkipEND*/)### t where
\$filter-drilldown and (service in ('smtp', 'SMTP', '25/tcp', '587/tcp', 'smtps', 'SMTPS',
'465/tcp') or service in ('pop3', 'POP3', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps',
'IMAPS', '993/tcp', 'pop3s', 'POP3S', '995/tcp')) group by profile order by total_num desc

Dataset Name	Description	Log Category
DLP-Web-Activity-Details	Web DLP Violations Summary	dlp
select		

from_itime(itime) as timestamp,
srcip,
dstip,
hostname,
profile,
filename,
filesize,
action,
direction

```
from
```

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

Dataset Name	Description	Log Category
Web-DLP-Chart	Web DLP Activity Summary	dlp
<pre>subtype, srcip, dstip, seven severity='critical' then 'Cr (`user`), ipstr(`srcip`)) is as data_loss from \$log where</pre>	e,`from` as sender, `to` as receiver, p rity, filename, direction, filesize, (c ritical Data Exfiltration' else (case w s not null then 'User Associated Data L e \$filter /*SkipSTART*/order by itime d (service) in ('http', 'https') group by	ase when hen coalesce(nullifna oss' else NULL end) end) esc/*SkipEND*/)### t where
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,</pre>	stamp,	

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

Dataset Name	Description	Log Category
FTP-DLP-Chart	FTP DLP Activity Summary	dlp

select

```
profile,
  count(*) as total_num
from
```

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

Dataset Name	Description	Log Category
top-users-by-browsetime	Top Users by website browsetime	traffic
<pre>(nullifna(`user`), ipstr(`srcip as domain, ebtr_agg_flat(\$brows time is not null group by user</pre>	<pre>ebtr_agg_flat(browsetime) as browseti p`)) as user_src, coalesce(nullifna(ho se_time) as browsetime from \$log where _src, domain) t group by user_src, dom ll, null) desc)### t group by user_src</pre>	<pre>stname), ipstr(`dstip`)) \$filter and \$browse_ ain order by ebtr_value</pre>
Dataset Name	Description	Log Category
wifi-usage-by-hour-authenticated	Wifi Usage by Hour - Authenticated	event
<pre>select hod, count(distinct stamac) as to</pre>	talnum	

from

###(select \$HOUR_OF_DAY as hod, stamac from \$log where \$filter and subtype='wireless' and action='client-authentication' group by hod, stamac)### t group by hod order by hod

Dataset Name	Description	Log Category
wifi-usage-authenticated-timeline	Wifi Usage Timeline - Authenticated	event

select

```
$flex_timescale(timestamp) as hodex,
count(distinct stamac) as totalnum
```

from

###(select \$flex_timestamp as timestamp, stamac from \$log where \$filter and subtype='wireless' and action='client-authentication' group by timestamp, stamac order by timestamp desc)### t group by hodex order by hodex

Dataset Name	Description	Log Category
app-top-user-by-bandwidth	Top 10 Applications Bandwidth by User Drilldown	traffic
<pre>select app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(`sentbyte`, 0)+ coa) as bandwidth from \$log where \$filter and (logflag&1>0) and nullifna(app) is not null group by app, user_src order by</pre>	lesce(`rcvdbyte`, 0)	
—		

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 \$log where \$filter</pre>		

```
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 from</pre>		
<pre>###(select \$DAY_OF_MONTH as as total_sent, sum(coalesce(rcvc (rcvdbyte, 0)) as total from \$10 not null and nullifna(dstintf) is sum(coalesce(sentbyte, 0)+coales unnest(array['download', 'upload bandwidth from (select coalesce as devid, coalesce(t1.vd_s, t2.v (coalesce(t1.total_sent, 0)+coal sent, 0)+coalesce(t1.total_rcvd,</pre>	<pre>dom, devid, vd, srcintf, dstintf, dbyte, 0)) as total_rcvd, sum(coale og where \$filter and (logflag&1>0) as not null group by dom, devid, vo ace(rcvdbyte, 0))>0 order by total d']) as type, unnest(array[sum(down t1.dom_s, t2.dom_s) as dom, coales rd_s) as vd, coalesce(t1.srcintf, t esce(t2.total_rcvd, 0)) as downloa 0)) as upload from qry t1 full jo intf group by dom, devid, vd, intf dom</pre>	esce (sentbyte, 0)+coalesce and nullifna (srcintf) is d, srcintf, dstintf having desc)### t) select dom, nload), sum(upload)]) as see (t1.devid_s, t2.devid_s) t2.dstintf) as intf, sum ad, sum(coalesce(t2.total_ pin qry t2 on t1.dom_

Dataset Name	Description	Log Category
ctap-SB-Files-Needing-Inspection-vs- Others	Files Needing Inspection vs Others	virus
select		
(case when suffix in (
'bat', 'cmd', 'exe', 'jar'	, 'msi', 'vbs',	
'7z', 'zip', 'gzip', 'lzw'		
'rar', 'cab', 'doc', 'docx 'xlsx', 'ppt', 'pptx', 'pd		
'lnk', 'js'	r, Swr,	
) then 'Higher Risk File Typ	es' else 'Excluded Files' end	
) as files,		
<pre>sum(total_num) as total_num from</pre>		
	ame) as suffix, count(*) as total_	num from \$log where

\$filter and dtype='fortisandbox' and nullifna(filename) is not null group by suffix order by
total_num desc)### t group by files order by total_num desc

Dataset Name	Description	Log Category
ctap-SB-Breakdown-of-File-Types	Breakdown of File Types	virus
<pre>select (case when suffix in ('exe', 'msi', 'upx', 'vbs 'dll', 'ps1', 'jar') then 'Executable Files' w</pre>	', 'bat', 'cmd', hen suffix in ('pdf') then 'Adobe	PDF' when suffix in ('swf')
<pre>then 'Adobe Flash' when suffix 'doc', 'docx', 'rtf', 'do 'dotm', 'dot') then 'Microsoft Word' whe 'xls', 'xlsx', 'xltx', 'x 'xlam', 'xlt') then 'Microsoft Excel' wh</pre>	in (tx', 'docm', n suffix in (lsm', 'xlsb',	
'ppsx', 'ppt', 'pptx', 'p 'pptm', 'ppsm', 'potm', ' 'pps', 'pot') then 'Microsoft PowerPoin		Microsoft Outlook' when
<pre>'cab', 'tgz', 'z', '7z', 'kgb', 'rar', 'zip', 'gz') then 'Archive Files' when) as filetype,</pre>		
	name) as suffix, count(*) as tota ' and nullifna(filename) is not r iletype order by total_num desc	

D - 4	aset		
	asar	Nan	
Dat	asel		

Description

ctap-SB-Top-Sandbox-Malicious-Exes

```
select
```

(case fsaverdict when 'malicious' then 5 when 'high risk' then 4 when 'medium risk' then 3 when 'low risk' then 2 else 1 end) as risk, filename, service, count(*) as total_num from \$log where \$filter and dtype = 'fortisandbox' and file name ext(filename) = 'exe' and fsaverdict not in ('clean', 'submission failed') group by filename,

Log Category

virus

```
risk,
 service
order by
 risk desc,
 total_num desc,
 filename
```

Dataset Name

Description Log Category virus ctap-SB-Sources-of-Sandbox-Sources of Sandbox Discovered Malware **Discovered-Malware** select srcip, count(*) as total num from \$log where \$filter and dtype = 'fortisandbox' and nullifna(filename) is not null and fsaverdict not in ('clean', 'submission failed') group by srcip order by total_num desc

Dataset Name	Description	Log Category
ctap-apprisk-ctrl-High-Risk-Application	Application risk high risk application	traffic
select		
risk as d_risk,		
count(distinct user_src) as us	ers,	
id,		
name,		
app_cat,		
technology,		
sum(bandwidth) as bandwidth,		
sum(sessions) as sessions		
from		
###(select lower(app) as lowap	<pre>p, coalesce(nullifna(`user`), nullifna</pre>	(`unauthuser`), ipstr
(`srcip`)) as user_src, sum(coal	esce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, count
(*) as sessions from \$log where	\$filter and (logflag&1>0) group by low	app, user_src order by
<pre>bandwidth desc)### t1 inner join</pre>	app_mdata t2 on t1.lowapp=lower(t2.na	me) where risk>='4'
group by id, name, app_cat, tech	nology, risk order by d_risk desc, ses	sions desc

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

```
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
	urce,	re like 'Adware%' then
(connect considering deting considering	

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

Dataset Name	Description	Log Category
ctap-App-Risk-Reputation-Top- Devices-By-Scores	Reputation Top Devices By-Scores	traffic
<pre>select coalesce(nullifna(`srcname`), ipstr(`srcip`), nullifna(`srcmac`)) as dev_src,</pre>		

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

Dataset Name	Description	Log Category
ctap-HTTP-SSL-Traffic-Ratio	HTTP SSL Traffic Ratio	traffic
select		
(
case when service in ('8	80/tcp', 'HTTP', 'http') then 'H	ITP' else 'HTTPS' end
) as service,		
sum(
<pre>coalesce(sentbyte, 0) + c</pre>	coalesce(rcvdbyte, 0)	
) as bandwidth		
irom		
\$log		
here \$filter		
and (
logflag&1>0		
)		
, and nullifna(app) is not r	1	
and service in (· · · ·	
'80/tcp', '443/tcp', 'HI	TP', 'HTTPS',	
'http', 'https'		
)		
roup by		
service		
aving		
sum(
<pre>coalesce(sentbyte, 0) + c</pre>	coalesce(rcvdbyte, 0)	
)& gt; 0		
order by		
bandwidth desc		
Dataset Name	Description	Log Category

Dataset Name	Description	Log Category

```
      ctap-Top-Source-Countries
      Top Source Countries
      traffic

      select
      srccountry,
      sum(
      coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
```

```
) as bandwidth
from
  $log
where
  $filter
  and (
   logflag&1>0
  )
  and nullifna(srccountry) is not null
  and srccountry & lt;& gt; 'Reserved'
group by
  srccountry
having
 sum (
   coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
 )& gt; 0
order by
 bandwidth desc,
  srccountry
```

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

select
 app_group,
 sum(bandwidth) as bandwidth

```
from
```

###(select app_group_name(app) as app_group, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out, count(*) as sessions from \$log where \$filter and (logflag&l>0) and nullifna (app) is not null group by app_group having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc)### t1 inner join app_mdata t2 on lower(t1.app_group)=lower (t2.name) where behavior like '%Cloud%' group by app_group order by bandwidth desc

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

select

app_group,
 sum(bandwidth) as bandwidth
from

###(select app_group_name(app) as app_group, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out, count(*) as sessions from \$log where \$filter and (logflag&1>0) and nullifna (app) is not null group by app_group having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc)### t1 inner join app_mdata t2 on lower(t1.app_group)=lower (t2.name) where app_cat='Cloud.IT' group by app_group order by bandwidth desc

Dataset Name	Description	Log Category
ctap-RAS-Apps	CTAP RAS Apps	traffic
select name as app_group,		

```
sum(bandwidth) as bandwidth
from
```

###(select app_group_name(app) as app_group, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out, count(*) as sessions from \$log where \$filter and (logflag&1>0) and nullifna (app) is not null group by app_group having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc)### t1 inner join app_mdata t2 on lower(t1.app_group)=lower (t2.name) where app_cat='Remote.Access' group by name order by bandwidth desc

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

select

name as app_group, sum(bandwidth) as bandwidth from

###(select app_group_name(app) as app_group, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out, count(*) as sessions from \$log where \$filter and (logflag&l>0) and nullifna (app) is not null group by app_group having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc)### t1 inner join app_mdata t2 on lower(t1.app_group)=lower (t2.name) where app_cat='Proxy' group by name order by bandwidth desc

Dataset Name	Description	Log Category
ctap-Top-SocialMedia-App-By- Bandwidth	Top SocialMedia Applications by Bandwidth Usage	traffic

select

```
app_group,
sum(bandwidth) as bandwidth,
sum(traffic_in) as traffic_in,
sum(traffic_out) as traffic_out,
sum(sessions) as sessions
```

from

###(select app_group_name(app) as app_group, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out, count(*) as sessions from \$log where \$filter and (logflag&1>0) and nullifna (app) is not null group by app_group having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc)### t1 inner join app_mdata t2 on lower(t1.app_group)=lower (t2.name) where app_cat='Social.Media' group by app_group order by bandwidth desc

Dataset Name	Description	Log Category
ctap-Top-Streaming-App-By- Bandwidth	Top Streaming applications by bandwidth usage	traffic
<pre>select app_group, sum(bandwidth) as bandwidth, sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_out sum(sessions) as sessions from ###(select app_group_name(app))</pre>		+coalesce(rcvdbyte,

0)) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out, count(*) as sessions from \$log where \$filter and (logflag&1>0) and nullifna (app) is not null group by app_group having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc)### t1 inner join app_mdata t2 on lower(t1.app_group)=lower (t2.name) where app_cat='Video/Audio' group by app_group order by bandwidth desc

Dataset Name	Description	Log Category
ctap-Top-Game-App-By-Bandwidth	Top Game applications by bandwidth usage	traffic
select app group,		

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

###(select app_group_name(app) as app_group, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out, count(*) as sessions from \$log where \$filter and (logflag&1>0) and nullifna (app) is not null group by app_group having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc)### t1 inner join app_mdata t2 on lower(t1.app_group)=lower (t2.name) where app_cat='Game' group by app_group order by bandwidth desc

Dataset Name	Description	Log Category
ctap-Top-P2P-App-By-Bandwidth	Top P2P applications by bandwidth usage	traffic

select

```
app_group,
sum(bandwidth) as bandwidth,
sum(traffic_in) as traffic_in,
sum(traffic_out) as traffic_out,
sum(sessions) as sessions
```

from

###(select app_group_name(app) as app_group, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out, count(*) as sessions from \$log where \$filter and (logflag&1>0) and nullifna (app) is not null group by app_group having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc)### t1 inner join app_mdata t2 on lower(t1.app_group)=lower (t2.name) where app_cat='P2P' group by app_group order by bandwidth desc

Dataset Name	Description	Log Category
ctap-apprisk-ctrl-Top-Web-Categories- Visited	Top 25 Web Categories Visited	traffic
<pre>select catdesc, count(distinct f_user) as user sum(sessions) as sessions, sum(bandwidth) as bandwidth</pre>	_num,	
<pre>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</pre>		

((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

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_grou service, count(*) as sessions, sum(coalesce(sentbyte, 0)+ coales) 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)	
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- Application risk web browsing activity hostname category webfilter Hostname-Category

```
select
  domain,
  catdesc,
  sum(visits) as visits
from
```

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

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

```
select
hostname,
string_agg(distinct catdesc, ', ') as agg_catdesc,
ebtr_value(
    ebtr_agg_flat(browsetime),
    null,
    $timespan
) as browsetime,
sum(bandwidth) as bandwidth,
sum(traffic_in) as traffic_in,
sum(traffic_out) as traffic_out
```

from

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

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

select

```
hourstamp,
```

```
sum(bandwidth)/ count(distinct daystamp) as bandwidth
```

```
from
```

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

Dataset Name	Description	Log Category
ctap-Top-Bandwidth-Hosts	Top Bandwidth Hosts	traffic
<pre>select hostname, sum(coalesce(sentbyte, 0)+) as bandwidth from \$log - traffic where \$filter and hostname is not null and (logflag&1>0) group by hostname having sum(</pre>	coalesce(rcvdbyte, 0)	

```
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
```

###(select app_s, (case when saas_s>=10 then 1 else 0 end) as is_saas from (select unnest (apps) as app_s, unnest(saasinfo) as saas_s from \$log where \$filter and apps is not null) t group by app_s, is_saas)### t group by is_saas order by is_saas

Dataset Name	Description	Log Category
saas-SaaS-Application-by-Category	Number of SaaS Applications by Category	traffic
<pre>select (case saas_cat when 0 then 'S) as saas_cat_str, count(distinct app s) as num s</pre>	Canctioned' else 'Unsanctioned' end	
<pre>from ###(select app_s, saas_s%10 as total_app from (select unnest(ap 0) as sentbyte, coalesce(rcvdbyte)</pre>	<pre>saas_cat, sum(sentbyte+rcvdbyte) as band ops) as app_s, unnest(saasinfo) as saas_s, se, 0) as rcvdbyte from \$log where \$filter by app_s, saas_cat order by bandwidth desc</pre>	coalesce(sentbyte, and apps is not

```
cat in (0, 1) group by saas_cat order by saas_cat
```

Dataset Name	Description	Log Category
saas-SaaS-Application-by-Bandwidth	Number of SaaS Applications by Bandwidth	traffic
<pre>) as saas_cat_str, sum(bandwidth) as bandwidth from ###(select app_s, saas_s%10 as total_app from (select unnest(app 0) as sentbyte, coalesce(rcvdbyte</pre>	anctioned' else 'Tolerated' end saas_cat, sum(sentbyte+rcvdbyte) as bandwidth ps) as app_s, unnest(saasinfo) as saas_s, coal e, 0) as rcvdbyte from \$log where \$filter and y app_s, saas_cat order by bandwidth desc)### order by saas_cat	esce(sentbyte, apps is not

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

```
select
  (
    case saas_cat when 0 then 'Sanctioned' else 'Tolerated' end
  ) as saas_cat_str,
  sum(total_app) as total_app
from
  ###(select app_s, saas_s%10 as saas_cat, sum(sentbyte+rcvdbyte) as bandwidth, count(*) as
total_app from (select unnest(apps) as app_s, unnest(saasinfo) as saas_s, coalesce(sentbyte,
  0) as sentbyte, coalesce(rcvdbyte, 0) as rcvdbyte from $log where $filter and apps is not
  null) t where saas_s>=10 group by app_s, saas_cat order by bandwidth desc)### t where saas_
cat in (0, 2) group by saas_cat order by saas_cat
```

Dataset Name	Description	Log Category
saas-SaaS-App-Users-vs-Others	Number of Users of SaaS Apps vs Others	traffic
<pre>select (case is_saas when 0 then '0) as app_type,</pre>	ther Apps' else 'SaaS Apps' end	

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>select (case saas_cat when 0 then 'Sanctioned' when 1 then 'Unsanctioned' else 'Others' end) as app_type, count(distinct saasuser) as total_user from ###(select saasuser, saas_s%10 as saas_cat from (select coalesce(nullifna(`user`), nullifna(`clouduser`), nullifna(`unauthuser`), srcname, ipstr(`srcip`)) as saasuser, unnest (saasinfo) as saas_s from \$log where \$filter and apps is not null) t where saas_s>=10 group by saasuser, saas cat)### t group by saas cat order by saas cat</pre>		
Dataset Name	Description	Log Category
saas-Top-SaaS-User-by-Bandwidth- Session	Top SaaS Users by Bandwidth and Session	traffic
<pre>select saasuser, sum(bandwidth) as bandwidth, sum(traffic_in) as traffic_on, sum(traffic_out) as traffic_out sum(sessions) as sessions, sum(session_block) as session_(sum(sessions) - sum(session_block)</pre>	ut, _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>) as saas_cat_str, count(distinct app_s) as t from ###(select app_s, saas_s%1 (saasinfo) as saas_s from \$1</pre>	.0 as saas_cat from (select unnest(apps) a .og where \$filter and apps is not null) t uner join app_mdata t2 on t1.app_s=t2.name	where saas_s>=10 group
Dataset Name	Description	Log Category
saas-Top-SaaS-Category-by-Numl of-User	ber- Top SaaS Categories by Number of Users	traffic
<pre>) as saas_cat_str, count(distinct saasuser) a from ###(select app_s, saas_s%1</pre>	en 'Sanctioned' else 'Unsactioned' end as total_user .0 as saas_cat, saasuser from (select unne coalesce(nullifna(`user`), nullifna(`clou	<u> </u>

```
null) t where saas_s>=10 group by app_s, saas_cat, saasuser)### t1 inner join app_mdata t2
on t1.app_s=t2.name where saas_cat in (0, 1) group by app_cat, saas_cat order by total_user
```

```
desc
```

Dataset Name	Description	Log Category
saas-Top-User-by-Number-of-SaaS- Application	Top Users by Number of SaaS Applications	traffic
select saasuser, (

(`unauthuser`), srcname, ipstr(`srcip`)) as saasuser from \$log where \$filter and apps is not

case saas_cat when 0 then 'Sanctioned' else 'Unsactioned' end) as saas_cat_str, count(distinct app_s) as total_app from

###(select app_s, saas_s%10 as saas_cat, saasuser from (select unnest(apps) as app_s, unnest(saasinfo) as saas_s, coalesce(nullifna(`user`), nullifna(`clouduser`), nullifna (`unauthuser`), srcname, ipstr(`srcip`)) as saasuser from \$log where \$filter and apps is not null) t where saas_s>=10 group by app_s, saas_cat, saasuser)### t where saas_cat in (0, 1) group by saasuser, saas cat order by total app desc

Dataset Name	Description	Log Category
saas-Top-SaaS-Application-by- Bandwidth-Session	Top SaaS Applications by Sessions and Bandwidth	traffic
<pre>select t2.id as app_id, app_s, app_cat, sum(bandwidth) as bandwidth, sum(traffic_in) as traffic_in sum(traffic_out) as traffic_or sum(sessions) as sessions, sum(session_block) as session (sum(sessions)- sum(session_1)) as session_pass</pre>	ut, _block,	
<pre>from ###(select app_s, sum(sentbyte+rcvdbyte) as bandwidth, sum(rcvdbyte) as traffic_in, sum (sentbyte) as traffic_out, count(*) as sessions, sum(is_blocked) as session_block from (select unnest(apps) as app_s, unnest(saasinfo) as saas_s, coalesce(sentbyte, 0) as</pre>		

(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
<pre>select app_s, sum(sentbyte + rcvdbyte) as bas</pre>	ndwidth	
from (
<pre>select unnest(apps) as app_s, unnest(saasinfo) as saas_s coalesce(sentbyte, 0) as s coalesce(rcvdbyte, 0) as r</pre>	entbyte,	
from		

```
where
  $filter
  and apps is not null
```

\$loq

```
) 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_ (</pre>	t,	

```
sum(sessions) - sum(session_block)
) as session_pass
```

```
from
```

###(select saasuser, app_s, sum(sentbyte+rcvdbyte) as bandwidth, sum(rcvdbyte) as traffic_ in, sum(sentbyte) as traffic_out, count(*) as sessions, sum(is_blocked) as session_block from (select coalesce(nullifna(`user`), nullifna(`clouduser`), nullifna(`unauthuser`), srcname, ipstr(`srcip`)) as saasuser, unnest(apps) as app_s, unnest(saasinfo) as saas_s, coalesce(sentbyte, 0) as sentbyte, coalesce(rcvdbyte, 0) as rcvdbyte, (CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END) as is_blocked from \$log where \$filter and apps is not null) t where saas_s=12 group by saasuser, app_s order by bandwidth desc)### t where \$filterdrilldown group by app_s order by bandwidth desc

Dataset Name	Description	Log Category
saas-Top-User-by-Tolerated-SaaS- Application-Drilldown	Top Users by Tolerated SaaS Applications	traffic
select		
saasuser,		
count(distinct app_s) as total	l_app	
from		
###(select saasuser, app_s, su	um(sentbyte+rcvdbyte) as bandwidth, sum(rcv	dbyte) as traffic_
in, sum(sentbyte) as traffic_out, count(*) as sessions, sum(is_blocked) as session_block		
<pre>from (select coalesce(nullifna(`user`), nullifna(`clouduser`), nullifna(`unauthuser`),</pre>		
<pre>srcname, ipstr(`srcip`)) as saas</pre>	suser, unnest(apps) as app_s, unnest(saasir	nfo) as saas_s,
coalesce(sentbyte, 0) as sentbyt	te, coalesce(rcvdbyte, 0) as rcvdbyte, (CAS	SE WHEN
(logflag&2>0) THEN 1 ELSE 0 END)) as is_blocked from \$log where \$filter and	l apps is not null)
t where saas_s=12 group by saasu	user, app_s order by bandwidth desc)### t g	group by saasuser

order by total_app desc

Dataset Name	Description	Log Category
saas-drilldown-Top-File-Sharing- SaaS-Application-Detail	Top File Sharing SaaS Applications Detail	traffic

```
select
saasuser,
sum(bandwidth) as bandwidth,
sum(traffic_in) as traffic_in,
sum(traffic_out) as traffic_out,
sum(sessions) as sessions,
sum(session_block) as session_block,
(
        sum(sessions)- sum(session_block)
) as session_pass
```

from

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

Dataset Name	Description	Log Category
saas-Top-File-Sharing-SaaS- Application	Top File Sharing Applications	traffic
<pre>select t2.id as appid, (case t2.risk when '5' th '2' then 'Info' else 'Low' en) as risk, app_group, bandwidth, traffic_in, traffic_out, session_block, session_pass, total_user from (select</pre>	en 'Critical' when '4' then 'Hid nd	gh' when '3' then 'Medium' when
<pre>app_group, count(distinct saasuse: sum(bandwidth) as band sum(traffic_in) as tra sum(traffic_out) as tra sum(sessions) as sessi sum(session_block) as a (sum(session_block) as a (sum(session_pass from ###(select app_group_n) bandwidth, sum(rcvdbyte) as a sum(session_s)</pre>	width, ffic_in, affic_out, ons, session_block, ession_block) ame(app_s) as app_group, saasus	affic_out, count(*) as sessions,

(`clouduser`), nullifna(`unauthuser`), srcname, ipstr(`srcip`)) as saasuser, unnest(apps) as app_s, unnest(saasinfo) as saas_s, coalesce(sentbyte, 0) as sentbyte, coalesce(rcvdbyte, 0) as rcvdbyte, (CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END) as is_blocked from \$log where \$filter and apps is not null) t where saas_s>=10 group by app_group, saasuser order by bandwidth desc)### t group by app_group) t1 inner join app_mdata t2 on lower(t1.app_ group)=lower(t2.name) where t2.app_cat='Storage.Backup' order by total_user desc, bandwidth desc

Dataset Name	Description	Log Category
saas-Top-File-Sharing-SaaS- Application-Drilldown	Top File Sharing Applications	traffic
<pre>select t2.id as appid, (case t2.risk when '5' th '2' then 'Info' else 'Low' e) as risk, app_group, bandwidth, traffic_in, traffic_out, session_pass, total_user from (select app_group, count(distinct saasuse sum(bandwidth) as band sum(traffic_out) as tra sum(traffic_out) as tra</pre>	r) as total_user, width, ffic_in, affic_out,	hen '3' then 'Medium' when
<pre>sum(sessions) as sessi sum(session_block) as (</pre>	•	
<pre>sum(sessions) - sum(s) as session_pass from</pre>	ession_block)	
<pre>###(select app_group_n bandwidth, sum(rcvdbyte) as sum(is_blocked) as session_b</pre>	<pre>ame(app_s) as app_group, saasuser, s traffic_in, sum(sentbyte) as traffic lock from (select coalesce(nullifna(uthuser`), srcname, ipstr(`srcip`))</pre>	_out, count(*) as sessions, `user`), nullifna

sum(is_blocked) as session_block from (select coalesce(nullifna(`user`), nullifna
(`clouduser`), nullifna(`unauthuser`), srcname, ipstr(`srcip`)) as saasuser, unnest(apps) as
app_s, unnest(saasinfo) as saas_s, coalesce(sentbyte, 0) as sentbyte, coalesce(rcvdbyte, 0)
as rcvdbyte, (CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END) as is_blocked from \$log where
\$filter and apps is not null) t where saas_s>=10 group by app_group, saasuser order by
bandwidth desc)### t group by app_group) t1 inner join app_mdata t2 on lower(t1.app_
group)=lower(t2.name) where t2.app_cat='Storage.Backup' order by total_user desc, bandwidth
desc

Dataset Name	Description	Log Category
aware-Device-By-Location	Device by Location	traffic

```
select
  'All' : :text as country,
   count(distinct devid) as device_count
from
  ###(select devid from $log where $filter group by devid)### t
```

Dataset NameDescriptionLog Categoryaware-Network-Endpoint-DevicesEndpoint Devices on Network

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

```
group by
          epname,
          map_dev.devid,
          map_dev.vd
      ) t
   where
     $filter
      and $filter - drilldown
    union all
    select
      'Unseen Devices' as category,
      3 as idx,
      count(distinct t1.epname) as total_num
    from
      $ADOM ENDPOINT t1
    where
      not exists (
        select
          1
        from
          (
            select
              epname,
              map_dev.devid,
             map_dev.vd,
              max(lastseen) as itime
            from
              $ADOM ENDPOINT t
              inner join $ADOM_EPEU_DEVMAP map_dev on t.epid = map_dev.epid
            where
              epname is not null
            group by
              epname,
              map dev.devid,
              map_dev.vd
          ) t2
        where
          $filter
          and $filter - drilldown
          and t1.epname = t2.epname
      )
 ) t
order by
  idx
```

Dataset Name	Description	Log Category
aware-New-Endpoint-Devices	New Endpoint Devices	
<pre>drop table if exists devmap_tmp; select epid, max(euid) as max_euid from \$ADOM_EPEU_DEVMAP</pre>	create temporary table devmap_tmp as (

```
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	
where		
<pre>epname is not null group by epname, map_dev.devid, map_dev.vd) t here \$filter and \$filter - drilldown coup by hodex cder by hodex</pre>		
Dataset Name	Description	Log Category
aware-Top-Endpoint-Operating- Systems	Top Endpoint Operating Systems	fct-traffic
elect osl as os, count(distinct hostname) as to rom	-	or and pullified
	<pre>1) as os1, hostname from \$log where \$filt hostname)### t group by os order by total_n</pre>	
Dataset Name	Description	Log Category
aware-Top-Endpoint-Applications- Windows	Top Endpoint Applications Windows	fct-traffic
elect srcname1 as srcname, count(distinct hostname) as to	otal_num	

from

```
###(select split_part(srcname, '.', 1) as srcname1, hostname from $log where $filter and
nullifna(srcname) is not null and lower(os) like '%windows%' group by srcname, hostname)###
t group by srcname order by total_num desc
```

Dataset Name	Description	Log Category
aware-Top-Endpoint-Applications-Mac	Top Endpoint Applications Mac	fct-traffic
	_ '.', 1) as srcname1, hostname from \$lo d lower(os) like '%mac os%' group by sr	
Dataset Name	Description	Log Category
aware-Top-SaaS-Application-by- Number-of-Users	Top SaaS Applications by Number of Users	traffic
, unnest(saasinfo) as saas_s, c `unauthuser`), srcname, ipstr(`	- s) as app_group, saasuser from (select oalesce(nullifna(`user`), nullifna(`clo srcip`)) as saasuser from \$log where \$f ll) t where saas_s>=10 group by app_gro	uduser`), nullifna
Dataset Name	Description	Log Category
aware-Summary-Of-Changes	Summary of Changes	event
<pre>select regexp_replace(msg, '[^]*\$', count(*) as total_num from \$log where \$filter and logid_to_int(logid) = 44547 group by msg_trim order by total_num desc</pre>	'') as msg_trim,	
Dataset Name	Description	Log Category
aware-Change-Details	Change Details	event
<pre>select \$calendar_time as timestamp, `user`, ui, msg from \$log where</pre>		

```
$filter
 and logid_to_int(logid) = 44547
order by
  timestamp desc
```

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

count(distinct vulnname) as vuln_num from

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

Dataset Name	Description	Log Category
aware-Vulnerabilities-Trend	Vulnerabilities Trend	fct-netscan
then 1 else 0 end) as critica	as timestamp, sum(case when lo al, sum(case when lower(vulnse	
(case when lower(vulnseverity		then 1 else 0 end) as medium, sun d) as Low from \$log where \$filter timescale order by timescale
Dataset Name	Description	Log Category

Dataset Name	Description	Log Category
aware-Top-Critical-Vulnerabilities	Top Critical Vulnerabilities	fct-netscan
select		
vulnname,		
vulnseverity,		
vulncat,		
count(distinct hostname) as t	otal_num	
from		
	, vulnseverity, vulncat, count(
	name) is not null and vulnsever.	
	y, vulncat order by total_num de	esc)### t group by vulnname,
vulnseverity, vulncat order by	total_num desc	

Dataset Name	Description	Log Category
aware-Top-Vulnerabilities-Last-Period	Top Vulnerabilities Last Period	fct-netscan
select		

```
vulnname,
vulnseverity,
```

```
sev_num,
vulncat,
count(distinct hostname) as total_num
from
```

###(select hostname, vulnname, vulnseverity, (CASE vulnseverity WHEN 'Critical' THEN 5
WHEN 'High' THEN 4 WHEN 'Medium' THEN 3 WHEN 'Info' THEN 2 WHEN 'Low' THEN 1 ELSE 0 END) as
sev_num, vulncat, count(*) as total_num from \$log where \$pre_period \$filter and nullifna
(vulnname) is not null group by hostname, vulname, vulnseverity, vulncat order by sev_num
desc, total_num desc)### t group by vulnname, vulnseverity, sev_num, vulncat order by sev_
num desc, total num desc

Dataset Name	Description	Log Category
aware-Top-New-Vulnerabilities	Top New Vulnerabilities	fct-netscan

drop

table if exists rpt_tmptbl_1;

drop

table if exists rpt_tmptbl_2; create temporary table rpt_tmptbl_1 as ###(select vulnid, vulnname, vulnseverity, vulncat, hostname from \$log where \$pre_period \$filter and nullifna (vulnname) is not null group by vulnid, vulnname, vulnseverity, vulncat, hostname)###; create temporary table rpt_tmptbl_2 as ###(select vulnid, vulnname, vulnseverity, vulncat, hostname from \$log where \$filter and nullifna(vulnname) is not null group by vulnid, vulnname, vulnseverity, vulncat, hostname)###; select vulnname, (case when vulnseverity='Critical' then 5 when vulnseverity='High' then 4 when vulnseverity='Medium' then 3 when vulnseverity='Low' then 2 when vulnseverity='Info' then 1 else 0 end) as sev, vulnseverity, vulncat, count(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 and vulnseverity = 'Critical group by hostname, user_src, vulnname, vulncat order by total_num desc</pre>		

Description	Log Category
Ingress Data Flow By Zone	traffic

from

###(select app, dstintf, sum(coalesce(rcvdbyte, 0)) as rcvdbyte from \$log where \$filter
group by app, dstintf having sum(coalesce(rcvdbyte, 0)) > 0 order by rcvdbyte desc)### tt1
inner join intftags tt2 on tt1.dstintf=tt2.intfname group by app, tag order by rcvdbyte desc

Dataset Name	Description	Log Category
aware-Egress-Data-Flow-By-Zone	Egress Data Flow By Zone	traffic

select

```
app,
tag,
sum(sentbyte) as sentbyte
```

from

###(select app, srcintf, sum(coalesce(sentbyte, 0)) as sentbyte from \$log where \$filter
group by app, srcintf having sum(coalesce(sentbyte, 0)) > 0 order by sentbyte desc)### tt1
inner join intftags tt2 on 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 n and nullifna(vulnname) is n group by hostname order by total_num desc</pre>		
Dataset Name	Description	Log Category
aware-Top-Attack-Targets	Top Attack Targets	fct-netscan
<pre>select hostname, srcip, os, vuln_num, (CASE sevid WHEN 5 THEN '('Info' ELSE 'Low' END</pre>	Critical' WHEN 4 THEN 'High'	WHEN 3 THEN 'Medium' WHEN '2' THEN

```
) as vulnseverity,
 sevid as severity num,
 left(cve_agg, 512) as cve_agg
from
  (
   select
     hostname,
     max(srcip) as srcip,
     string agg(distinct os1, '/') as os,
     count(distinct vulnname) as vuln num,
     max(
        (
         CASE vulnseverity WHEN 'Critical' THEN 5 WHEN 'High' THEN 4 WHEN 'Medium' THEN 3
WHEN 'Info' THEN 2 WHEN 'Low' THEN 1 ELSE 0 END
       )
     ) as sevid,
     string agg(distinct cve id, ',') as cve agg
    from
      ###(select hostname, max(deviceip) as srcip, split part(os, ',', 1) as os1, vulnname,
vulnseverity, vulnid from $log where $filter and nullifna(vulnname) is not null and nullifna
```

(vulnseverity) is not null group by hostname, os1, vulnname, vulnseverity, vulnid)### t1 left join fct_mdata t2 on t1.vulnid=t2.vid::int group by hostname) t order by severity_num desc, vuln_num desc

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

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

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

Dataset Name	Description	Log Category
aware-Threats-Type-By-Severity	Threats Type by Severity	virus
<pre>select threat_type, sum(critical) as critical, sum(high) as high, sum(medium) as medium, sum(low) as low</pre>		
type, sum(case when crlevel = 'c	<pre>ype='botnet' then 'Botnets' else 'Malware' end ritical' then 1 else 0 end) as critical, sum(c nd) as high, sum(case when crlevel = 'medium'</pre>	ase when

end) as medium, sum(case when crlevel = 'low' then 1 else 0 end) as low from \$log-virus
where \$filter and nullifna(virus) is not null group by threat_type)### union all ###(select
'Intrusions' as threat_type, sum(case when severity = 'critical' then 1 else 0 end) as
critical, sum(case when severity = 'high' then 1 else 0 end) as high, sum(case when severity
= 'medium' then 1 else 0 end) as medium, sum(case when severity = 'low' then 1 else 0 end)
as low from \$log-attack where \$filter and nullifna(attack) is not null group by threat_
type)### union all ###(select 'Botnets' as threat_type, sum(case when apprisk = 'critical'
then 1 else 0 end) as critical, sum(case when apprisk = 'high' then 1 else 0 end) as high,
sum(case when apprisk = 'medium' then 1 else 0 end) as medium, sum(case when apprisk = 'low'
then 1 else 0 end) as low from \$log-app-ctrl where \$filter and lower(appcat)='botnet' group
by threat type)###) t group by threat type

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

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

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

```
Dataset Name
                                  Description
                                                                                 Log Category
aware-Threats-By-Day-Radar
                                  Threats by Day
                                                                                 virus
select
 daystamp,
  sum(total num) as total num
from
    ###(select $day of week as daystamp, count(*) as total num from $log-virus where $filter
and nullifna(virus) is not null group by daystamp)### union all ###(select $day of week as
daystamp, count(*) as total num from $log-attack where $filter and nullifna(attack) is not
null group by daystamp)### union all ###(select $day of week as daystamp, count(*) as total
num from $log-app-ctrl where $filter and lower(appcat) = 'botnet' group by daystamp) ###) t
group by daystamp order by daystamp
```

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

group by virus order by total_num desc

Dataset Name Description Log Category aware-Top-Malware-By-Count Top Malware by Count app-ctrl select virus, malware_type, risk level, count (distinct dstip) as victim, count(distinct srcip) as source, sum(total num) as total num from (###(select app as virus, 'Botnet C&C' as malware_type, apprisk::text as risk_level, dstip, srcip, count(*) as total num from \$log-app-ctrl where \$filter and lower

dstip, srcip, count(*) as total_num from \$log-app-ctrl where \$filter and lower (appcat)='botnet' and apprisk is not null group by app, malware_type, apprisk, dstip, srcip order by total_num desc)### union all ###(select virus, (case when eventtype='botnet' then 'Botnet C&C' else 'Virus' end) as malware_type, crlevel::text as risk_level, dstip, srcip, count(*) as total_num from \$log-virus where \$filter and nullifna(virus) is not null and crlevel is not null group by virus, malware_type, crlevel, dstip, srcip order by total_num desc)### union all ###(select attack as virus, (case when eventtype='botnet' then 'Botnet C&C' else 'Virus' end) as malware_type, crlevel::text as risk_level, dstip, srcip, count(*) as total_num from \$log-attack where \$filter and (logflag&16>0) and crlevel is not null group by virus, malware_type, crlevel, dstip, srcip order by total_num desc)###) t group by virus, malware_type, risk_level order by total_num desc

Dataset Name	Description	Log Category
aware-Top-Failed-Login-Attempts	Top Failed Login Attempts	event
<pre>select `user` as f_user, ui, dstip, count(status) as total_failed from \$log where \$filter and nullifna(`user`) is not m and logid_to_int(logid) = 320 group by ui, f_user, dstip order by total failed desc</pre>	ull	even

Dataset Name	Description	Log Category
aware-Top-Failed-Authentication- Attempts	VPN failed logins	event
<pre>select f_user, tunneltype, sum(total_num) as total_num</pre>		
<pre>total_num from \$log where \$filt (tunneltype, 3)='ssl') and activ</pre>	[`xauthuser`), `user`) as f_user, tu ter and subtype='vpn' and (tunneltype on in ('ssl-login-fail', 'ipsec-log a(`user`)) is not null group by f_u der by total_num desc	e='ipsec' or left in-fail') and coalesce

Dataset Name	Description	Log Category
aware-Top-Denied-Connections	Top Denied Connections	traffic
<pre>select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, service '(' ipstr(srcip) dstip, count(*) as total_num from \$log where \$filter and (</pre>	<pre> ')' as interface,</pre>	
logflag&1>0)		
<pre>and action = 'deny' group by user_src, interface, dstip order by total num desc</pre>		
Dataset Name	Description	Log Category
aware-Failed-Compliance-Checked- By-Device	Failed Compliance Checked by Device	event

select devid, 'Failed' as results, count(distinct reason) as total_num from ### (select devid, reason from \$log where \$filter and subtype='compliance-check' and 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
drop
  table if exists tmp_ep_eu_map; create temporary table tmp_ep_eu_map as (
   select
     epid,
     euid
   from
      $ADOM EPEU DEVMAP
    where
     euid & gt;= 1024
  );
select
 coalesce(
  nullifna(epname),
   nullifna(
     ipstr(`srcip`)
   ),
    'Unknown'
  ) as epname,
  user_agg,
  sevid,
  (
    CASE sevid WHEN 5 THEN 'Critical' WHEN 4 THEN 'High' WHEN 3 THEN 'Medium' WHEN '2' THEN
'Info' ELSE 'Low' END
 ) as severity,
 threats,
 bl_count as total_bl
from
  (
    select
     th1.epid,
     srcip,
     sevid,
     bl count,
     threats
    from
      (
        select
          epid,
          srcip,
          max(verdict) + 1 as sevid,
          sum(bl_count) as bl_count
        from
          (
            (
              select
                epid,
                srcip,
                day_st as itime,
                bl count,
                verdict,
                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
```

Dataset Name

aware-loc-Potential-Breach-By-Day

IOC Potential Breach by Day

Description

```
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-Potential-Breach-By-Day- Bar	IOC Potential Breach by Day	app-ctrl
<pre>select number, day_st as itime from (select count(epid) as number, to_char(</pre>		

```
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

Dataset Name	Description	Log Category
aware-loc-Suspicion-Summary	IOC Suspicion Summary	app-ctrl
<pre>aware-loc-Suspicion-Summary select coalesce(nullifna(epname), nullifna(ipstr(`srcip`)), 'Unknown') as epname, cs_count as total_cs, cs_score as max_cs, verdict as max_verdict, threats from</pre>	IOC Suspicion Summary	app-ctrl
(select th1.epid, srcip,		

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

```
inner join (
 select
   epid,
    string_agg(name, ',') as threats
 from
    (
      (
        select
          epid,
          thid
        from
          (
            (
              select
                epid,
                thid,
                itime,
                unnest(dvid) as dvid_s
              from
                (
                  select
                    epid,
                    unnest(threatid) as thid,
                    day_st as itime,
                    dvid
                  from
                    $ADOMTBL PLHD IOC VERDICT
                  where
                    bl count = 0
                    and cs count>0
                ) 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>select botnet, count(distinct `qname`) count(distinct ipstr(`dstip`) as dnssvr_cnt, sum(total_num) as total_ min(</pre>)	

```
from_itime(first_seen)
) as first_seen,
max(
    from_itime(last_seen)
) as last_seen
from
```

###(select coalesce(`botnetdomain`, ipstr(`botnetip`)) as botnet, qname, dstip, count(*)
as total_num, min(nanosec_to_sec(eventtime)) as first_seen, max(nanosec_to_sec(eventtime))
as last_seen from \$log where \$filter and logid in ('1501054601', '1501054600') group by
botnet, qname, dstip order by total num desc)### t group by botnet order by first seen desc

Dataset Name	Description	Log Category
aware-High-Risk-URL-Category	Category of High Risk URLs	webfilter
<pre>select catdesc, string_agg(distinct hostname 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</pre>	, ',') as hostname_agg,	
<pre>from ###(select catdesc, hostname</pre>	, max(action) as action, count(*) a	s total num, min(itime) as
first_seen, max(itime) as last	_seen from \$log where \$filter and c name order by total_num desc)### t	at in (26, 61, 86, 88, 90,

```
total_num desc
```

Dataset Name	Description	Log Category
aware-Malicious-Files	Type of Malicious Files from AV and Sandbox	virus
<pre>select virus, left(url_agg, 1000) as url_ left(filename_agg, 1000) as quarskip, action, from_sandbox, total_num, first_seen, last_seen from (select virus, string_agg(distinct url</pre>	filename_agg,	
string_agg(distinct fil max(quarskip) as quarsk	ename, ' ') as filename_agg, ip,	
<pre>max(action) as action, max(from sandbox) as fr</pre>	om sandbox.	
sum(total_num) as total	—	

```
min(
    from_itime(first_seen)
) as first_seen,
max(
    from_itime(last_seen)
) as last_seen
from
```

###(select virus, url, filename, max(quarskip) as quarskip, max(action) as action, (case when logid in ('0211009234', '0211009235') then 1 else 0 end) as from_sandbox, count (*) as total_num, min(itime) as first_seen, max(itime) as last_seen from \$log where \$filter and virus is not null and logid in ('0211009234', '0201009235', '0211008192', '0211008193', '0211008194', '0211008195') group by virus, url, filename, from_sandbox order by total_num desc)### t group by virus) t order by total num desc

Dataset Name	Description	Log Category
newthing-New-Users	New users	fct-traffic

drop

table if exists rpt_tmptbl_1;

drop

table if exists rpt_tmptbl_2; create temporary table rpt_tmptbl_1 as ###(select coalesce (nullifna(`user`), ipstr(`srcip`)) as f_user, min(dtime) as start_time from \$log where \$pre_ period \$filter group by f_user order by start_time desc)###; create temporary table rpt_ tmptbl_2 as ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as f_user, min(dtime) as start_time from \$log where \$filter group by f_user order by start_time desc)###; select f_ user, from_dtime(min(start_time)) as start_time from rpt_tmptbl_2 where f_user is not null and not exists (select 1 from rpt_tmptbl_1 where rpt_tmptbl_2.f_user=rpt_tmptbl_1.f_user) group by f_user order by start_time desc

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

drop

table if exists rpt_tmptbl_1;

drop

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

Dataset Name	Description	Log Category
newthing-New-Software-Installed	New software installed	fct-traffic
<pre>drop table if exists rpt_tmptbl_1; drop</pre>		
table if exists rpt_tmptbl_2;	create temporary table rpt_tmptbl here \$pre_period \$filter and null	

null group by srcproduct, hostname order by srcproduct)###; create temporary table rpt_
tmptbl_2 as ###(select srcproduct, hostname from \$log where \$filter and nullifna(srcproduct)
is not null group by srcproduct, hostname order by srcproduct)###; select srcproduct,
string_agg(distinct hostname, ',') as host_agg from rpt_tmptbl_2 where not exists (select 1
from rpt_tmptbl_1 where rpt_tmptbl_2.srcproduct=rpt_tmptbl_1.srcproduct) group by srcproduct
order by srcproduct

Dataset Name	Description	Log Category
newthing-New-Security-Threats	New security threats	virus
drop		

table if exists rpt_tmptbl_1;

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

from (

###(select app as threat name, 1 as cat id, srcip from \$log-app-ctrl where \$pre period \$filter and nullifna(app) is not null and lower(appcat)='botnet' group by threat name, cat id, srcip)### union all ###(select virus as threat name, 2 as cat id, srcip from \$log-virus where \$pre period \$filter and nullifna(virus) is not null group by threat name, cat id, srcip)### union all ###(select attack as threat name, 3 as cat id, srcip from \$log-attack where \$pre period \$filter and nullifna(attack) is not null group by threat name, cat id, srcip)###) t; create temporary table rpt_tmptbl_2 as select * from (###(select \$DAY_OF_MONTH as daystamp, app as threat name, 1 as cat id, srcip from \$log-app-ctrl where \$filter and nullifna(app) is not null and lower(appcat)='botnet' group by daystamp, threat name, cat id, srcip order by daystamp)### union all ###(select \$DAY OF MONTH as daystamp, virus as threat name, 2 as cat id, srcip from \$log-virus where \$filter and nullifna(virus) is not null group by daystamp, threat_name, cat_id, srcip order by daystamp)### union all ###(select \$DAY OF MONTH as daystamp, attack as threat name, 3 as cat id, srcip from \$log-attack where \$filter and nullifna(attack) is not null group by daystamp, threat name, cat id, srcip order by daystamp)###) t; select threat_name, (case cat_id when 1 then 'Botnet' when 2 then 'Malware' when 3 then 'Attack' end) as threat cat, count(distinct srcip) as host num, string agg (distinct cve, ',') as cve agg from rpt tmptbl 2 left join ips mdata t2 on rpt tmptbl 2.threat name=t2.name where not exists (select 1 from rpt tmptbl 1 where rpt tmptbl 2.threat_name=rpt_tmptbl_1.threat_name) group by threat_name, threat_cat order by host_num desc

Dataset Name	Description	Log Category
newthing-dns-Botnet-Domain-IP	New Queried Botnet C&C Domains and IPs	dns
,		

drop

```
table if exists rpt_tmptbl_1;
drop
```

table if exists rpt_tmptbl_2; create temporary table rpt_tmptbl_1 as ###(select coalesce (botnetdomain, ipstr(botnetip)) as domain, cast('Botnet C&C' as char(32)) as malware_type, (case when action='block' then 'Blocked' when action='redirect' then 'Redirected' else 'Passed' end) as action, srcip, (CASE WHEN level IN ('critical', 'alert', 'emergency') THEN 5 WHEN level='error' THEN 4 WHEN level='warning' THEN 3 WHEN level='notice' THEN 2 ELSE 1 END) as sevid, coalesce(botnetdomain, ipstr(botnetip)) as sources_s, count(*) as total_num from \$log where \$pre_period \$filter and (botnetdomain is not null or botnetip is not null) group by domain, action, srcip, sevid order by sevid desc)###; create temporary table rpt_ tmptbl 2 as ###(select coalesce(botnetdomain, ipstr(botnetip)) as domain, cast('Botnet C&C' as char(32)) as malware_type, (case when action='block' then 'Blocked' when action='redirect' then 'Redirected' else 'Passed' end) as action, srcip, (CASE WHEN level IN ('critical', 'alert', 'emergency') THEN 5 WHEN level='error' THEN 4 WHEN level='warning' THEN 3 WHEN level='notice' THEN 2 ELSE 1 END) as sevid, coalesce(botnetdomain, ipstr (botnetip)) as sources_s, count(*) as total_num from \$log where \$filter and (botnetdomain is not null or botnetip is not null) group by domain, action, srcip, sevid order by sevid desc)###; 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 *</pre>	; ; create temporary table rpt_tmptbl_1	as
—	ame, 1 as cat_id, srcip from \$log-app- ot null and lower(appcat)='botnet' gro	—

(app id, srcip)### union all ###(select virus as threat name, 2 as cat id, srcip from \$log-virus where \$pre period \$filter and nullifna(virus) is not null group by threat name, cat id, srcip)### union all ###(select attack as threat name, 3 as cat id, srcip from \$log-attack where \$pre period \$filter and nullifna(attack) is not null group by threat_name, cat_id, srcip)###) t; create temporary table rpt_tmptbl_2 as select * from (###(select \$flex_ timestamp as timestamp, app as threat name, 1 as cat id, srcip from \$log-app-ctrl where \$filter and nullifna(app) is not null and lower(appcat)='botnet' group by timestamp, threat name, cat id, srcip order by timestamp)### union all ###(select \$flex timestamp as timestamp, virus as threat name, 2 as cat id, srcip from \$log-virus where \$filter and nullifna(virus) is not null group by timestamp, threat name, cat id, srcip order by timestamp)### union all ###(select \$flex timestamp as timestamp, attack as threat name, 3 as cat id, srcip from \$log-attack where \$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 drop	1;	
	2; create temporary table rpt_tmpt	

vulnname, vulnseverity, vulncat, hostname from \$log where \$pre_period \$filter and nullifna
(vulnname) is not null group by vulnid, vulnname, vulnseverity, vulncat, hostname)###;

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

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

drop

table if exists rpt_tmptbl_1;

```
drop
```

table if exists rpt_tmptbl_2; create temporary table rpt_tmptbl_1 as ###(select vulnid, vulnname, vulnseverity, vulncat, hostname from \$log where \$pre_period \$filter and nullifna (vulnname) is not null group by vulnid, vulnname, vulnseverity, vulncat, hostname)###; create temporary table rpt_tmptbl_2 as ###(select vulnid, vulnname, vulnseverity, vulncat, hostname from \$log where \$filter and nullifna(vulnname) is not null group by vulnid, vulnname, vulnseverity, vulncat, hostname)###; select vulnseverity, count (distinct vulnid) as vuln_num from rpt_tmptbl_2 where not exists (select 1 from rpt_tmptbl_1 where rpt_tmptbl_ 2.vulnid=rpt_tmptbl_1.vulnid) group by vulnseverity order by (case when vulnseverity='Critical' then 5 when vulnseverity='High' then 4 when vulnseverity='Medium' then 3 when vulnseverity='Low' then 2 when vulnseverity='Info' then 1 else 0 end) desc

Dataset Name	Description	Log Category
newthing-System-Alerts	System Alerts	local-event
<pre>select from_itime(itime) as timestamp msg from \$log where \$filter and msg is not null and level = 'critical' order by timestamp desc</pre>	2, ,	
ermestamp dese		
Dataset Name	Description	Log Category
-	Description Configuration Changes	Log Category event

```
where
   $filter
   and cfgtid>0
order by
   time_s desc
```

Dataset Name	Description	Log Category
newthing-FortiGate-Upgrades	FortiGate Upgrades	event
<pre>select devid, from_dtime(dtime) as time_s, info[1] as intf, info[2] as prev_ver, info[3] as new_ver from (select devid, dtime, regexp_matches(msg, 'from ([^]+) \\(0)) as info from \$log where \$filter and action = 'restore-image"</pre>		
) t order by time s desc		

Dataset Name

newthing-User-UpgradesUser Upgradesfct-eventdrop table if exists rpt_tmptbl_1; droptable if exists rpt_tmptbl_2; create temporary table rpt_tmptbl_1 as ###(select distinct on (1, 2) fgtserial, hostname, deviceip, os, dtime from \$log where \$pre_period \$filter and hostname is not null order by fgtserial, hostname, dtime desc)###; create temporary table rpt_tmptbl_2 as ###(select distinct on (1, 2) fgtserial, hostname, deviceip, os, dtime from \$log where \$filter and hostname is not null order by fgtserial, hostname, deviceip, os, dtime from \$log where \$filter and hostname is not null order by fgtserial, hostname, time desc)###; select distinct on (1, 2) t2.fgtserial as devid, t2.hostname, t2.deviceip, t1.os as prev_os			
<pre>table if exists rpt_tmptbl_1; drop table if exists rpt_tmptbl_2; create temporary table rpt_tmptbl_1 as ###(select distinct on (1, 2) fgtserial, hostname, deviceip, os, dtime from \$log where \$pre_period \$filter and hostname is not null order by fgtserial, hostname, dtime desc)###; create temporary table rpt_tmptbl_2 as ###(select distinct on (1, 2) fgtserial, hostname, deviceip, os, dtime from \$log where \$filter and hostname is not null order by fgtserial, hostname, dtime desc)###; select distinct on (1, 2) t2.fgtserial as devid, t2.hostname, t2.deviceip, t1.os as prev_os</pre>	newthing-User-Upgrades	User Upgrades	fct-event
t2.os as cur_os, from_dtime(t1.dtime) as time_s from rpt_tmptbl_2 t2 inner join rpt_tmptbl_ t1 on t2.fgtserial=t1.fgtserial and t2.hostname=t1.hostname and t2.os!=t1.os order by devid t2.hostname, t1.dtime desc	<pre>table if exists rpt_tmpt drop table if exists rpt_tmpt on (1, 2) fgtserial, hostr hostname is not null order rpt_tmptbl_2 as ###(select \$log where \$filter and hos select distinct on (1, 2) t2.os as cur_os, from_dtin t1 on t2.fgtserial=t1.fgts</pre>	- cbl_2; create temporary table rpt hame, deviceip, os, dtime from \$1 r by fgtserial, hostname, dtime de t distinct on (1, 2) fgtserial, he stname is not null order by fgtse t2.fgtserial as devid, t2.hostnam ne(t1.dtime) as time_s from rpt_tage serial and t2.hostname=t1.hostname	where \$pre_period \$filter and esc)###; create temporary table ostname, deviceip, os, dtime from rial, hostname, dtime desc)###; me, t2.deviceip, t1.os as prev_os, mptbl_2 t2 inner join rpt_tmptbl_1

Description

Dataset Name	Description	Log Category
GTP-List-of-APN-Used	List of APNs Used	gtp
select		
apn,		

Log Category

```
from_dtime(
    min(first_seen)
) as first_seen,
from_dtime(
    max(last_seen)
) as last_seen
from
```

###(select apn, min(dtime) as first_seen, max(dtime) as last_seen from \$log where \$filter and nullifna(apn) is not null group by apn order by last_seen desc)### t group by apn order by last_seen desc, first_seen

Dataset Name	Description	Log Category
GTP-Top-APN-by-Bytes	Top APNs by Bytes	gtp
select		
apn,		
sum(
<pre>coalesce(`u-bytes`, 0)</pre>		
) as total_bytes		
from		
\$log		
where		
\$filter		
and nullifna(apn) is not null		
and status = 'traffic-count'		
group by		
apn		
having		
sum(
<pre>coalesce(`u-bytes`, 0)</pre>		
)& gt; 0		
order by		
total_bytes desc		
Dataset Name	Description	Log Category

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 n	ull	
and status = 'traffic-coun	t'	
group by		
apn		
naving		
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
<pre>select apn, sum(coalesce(`u-pkts`, 0)) as total_num from \$log where \$filter and nullifna(apn) is not null and status = 'traffic-count' group by apn having sum(coalesce(`u-pkts`, 0)) & gt; 0 order by total_num desc</pre>		

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 v count(distinct sources_s) sum(total num) as total nu</pre>	as sources,	
<pre>(32)) as malware_type, (case 'Redirected' else 'Passed' e 'emergency') THEN 5 WHEN lev level='notice' THEN 2 ELSE 1 sources_s, count(*) as total</pre>	domain, ipstr(botnetip)) as domain, cast('Bo when action='block' then 'Blocked' when act nd) as action, srcip, (CASE WHEN level IN (el='error' THEN 4 WHEN level='warning' THEN END) as sevid, coalesce(botnetdomain, ipst: _num from \$log where \$filter and (botnetdoma by domain, action, srcip, sevid order by sev ion order by total_num desc	tion='redirect' then 'critical', 'alert', 3 WHEN r(botnetip)) as ain is not null or
Defect Norma	Description	

Dataset Name	Description	Log Category
dns-Botnet-Usage	Top Queried Botnet C&C Domains and IPs	dns
<pre>select domain, malware_type, action,</pre>		

```
count(distinct srcip) as victims,
count(distinct sources_s) as sources,
sum(total_num) as total_num
```

from

###(select coalesce(botnetdomain, ipstr(botnetip)) as domain, cast('Botnet C&C' as char (32)) as malware type, (case when action='block' then 'Blocked' when action='redirect' then 'Redirected' else 'Passed' end) as action, srcip, (CASE WHEN level IN ('critical', 'alert', 'emergency') THEN 5 WHEN level='error' THEN 4 WHEN level='warning' THEN 3 WHEN level='notice' THEN 2 ELSE 1 END) as sevid, coalesce(botnetdomain, ipstr(botnetip)) as sources s, count(*) as total num from \$log where \$filter and (botnetdomain is not null or botnetip is not null) group by domain, action, srcip, sevid order by sevid desc)### t group by domain, malware type, action order by total num desc

Dataset Name	Description	Log Category
Dns-Detected-Botnet	Top Queried Botnet C&C Domains and IPs	dns
<pre>(32)) as malware_type, (c 'Redirected' else 'Passed 'emergency') THEN 5 WHEN level='notice' THEN 2 ELS sources_s, count(*) as to botnetip is not null) gro</pre>	s) as sources,	<pre>tion='redirect' then 'critical', 'alert', 3 WHEN c(botnetip)) as ain is not null or</pre>
Dataset Name	Description	Log Category
dns-Botnet-Domain-IP	Queried Botnet C&C Domains and IPs	dns
<pre>select domain, srcip, sevid, (CASE sevid WHEN 5 THE 'Info' ELSE 'Low' END) as severity from</pre>	N 'Critical' WHEN 4 THEN 'High' WHEN 3 THEN 'Me	edium' WHEN '2' THEN
<pre>###(select coalesce(bot (32)) as malware_type, (c 'Redirected' else 'Passed</pre>	netdomain, ipstr(botnetip)) as domain, cast('Bo ase when action='block' then 'Blocked' when act ' end) as action, srcip, (CASE WHEN level IN ('	cion='redirect' then 'critical', 'alert',

'emergency') THEN 5 WHEN level='error' THEN 4 WHEN level='warning' THEN 3 WHEN level='notice' THEN 2 ELSE 1 END) as sevid, coalesce(botnetdomain, ipstr(botnetip)) as sources s, count(*) as total num from \$log where \$filter and (botnetdomain is not null or botnetip is not null) group by domain, action, srcip, sevid order by sevid desc)### t group by domain, srcip, sevid order by sevid desc, domain

Dataset Name	Description	Log Category
dns-High-Risk-Source	High Risk Sources	dns
) as num_cri, sum(case when sevid = 4 t) as num_hig, sum(_num, hen total_num else 0 end hen total_num else 0 end hen total_num else 0 end	
from ###(select srcip, (CASE	WHEN level IN ('critical', 'ale	rt!. 'emergency') THEN 5 WHEN
<pre>level='error' THEN 4 WHEN sevid, count(*) as total_</pre>	level='warning' THEN 3 WHEN lev num from \$log where \$filter and	<pre>el='notice' THEN 2 ELSE 1 END) as srcip is not null group by srcip, by srcip having sum(total_num)>0</pre>

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) sum(case when sevid = 5 ther) as num_cri, sum(case when sevid = 4 ther) as num_hig, sum(case when sevid = 3 ther) as num_med, sum(case when sevid = 2 ther) as num_inf, sum(case when sevid = 1 ther) as num_low from ###(select \$flex_timestamp 'emergency') THEN 5 WHEN lev level='notice' THEN 2 ELSE 1</pre>	as timescale, h total_num else 0 end h total_num else 0 end h total_num else 0 end h total_num else 0 end	varning' THEN 3 WHEN al_num from \$log where \$filter
Dataset Name	Description	Log Category

Dataset Name	Description	Log Category
dns-Top-Queried-Domain	Top Queried Domain	dns

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

Dataset Name	Description	Log Category
dns-Top-Domain-Lookup-Failure-Bar	Top Domain Lookup Failures	dns
select		
qname,		
srcip,		
count(*) as total_num		
from		
\$log		
where		
\$filter		
and qname is not null		
and (
action = 'block'		
or logid_to_int(logid) = 5400	1	
)		
group by		
qname,		
srcip		
order by		
total_num desc		

Dataset Name	Description	Log Category
dns-Top-Domain-Lookup-Failure- Table	Top Domain Lookup Failures	dns
<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) = 5400) group by qname,</pre>	1	

0

srcip
order by
total_num desc

Dataset Name	Description	Log Category
dns-Query-Timeout	Query Timeout	dns
select		
srcip,		
qname,		
count(*) as total_num		
from		
\$log		
where		
\$filter		
and srcip is not null		
and logid_to_int(logid) = 54003		
group by		
qname,		
srcip		
order by		
total_num desc		

Dataset Name	Description	Log Category
dns-Blocked-Query	Blocked Queries	dns
select srcip,		
<pre>msg, count(*) as total_num from</pre>		
\$log where		
<pre>\$filter and srcip is not null and action = 'block'</pre>		
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(
   sum(lograte peak) as decimal(10, 2)
 ) as lograte peak,
 sum(session peak) as session peak,
 sum(transmit kbps peak) as transmit kbps peak,
 cast(
   sum(cps ave) as decimal(10, 0)
 ) as cps ave,
 sum(cps peak) as cps peak
from
  (
    select
      $flex timescale(timestamp) as hodex,
      devid,
      slot,
      sum(total_cpu) / sum(count) cpu_ave,
      sum(total mem) / sum(count) as mem ave,
      sum(total disk) / sum(count) as disk ave,
      sum(
        total_trate + total_erate + total_orate
      )/ 100.00 / sum(count) as log_rate,
      sum(totalsession) / sum(count) as sessions,
      sum(sent) / sum(count) as sent kbps,
      sum(recv) / sum(count) as recv kbps,
      sum(sent + recv) / sum(count) as transmit kbps,
      max(mem peak) as mem peak,
     max(disk peak) as disk peak,
     max(cpu peak) as cpu peak,
      max(lograte peak) / 100.00 as lograte peak,
      max(session peak) as session peak,
     max(transmit peak) as transmit kbps peak,
      sum(cps) / sum(count) as cps ave,
      max(cps peak) as cps peak
    from
```

Dataset Name	Description	Log Category
perf-stat-mem-usage-drilldown	Fortigate resource detail timeline	event

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

```
) as sessions,
 cast (
   sum(sent kbps) as decimal(10, 0)
  ) as sent kbps,
 cast(
   sum(recv kbps) as decimal(10, 0)
 ) as recv kbps,
 cast(
   sum(transmit kbps) as decimal(10, 0)
 ) as transmit kbps,
 max(mem peak) as mem peak,
 max(disk peak) as disk peak,
 max(cpu peak) as cpu peak,
 cast(
   sum(lograte peak) as decimal(10, 2)
 ) as lograte peak,
 sum(session peak) as session peak,
 sum(transmit kbps peak) as transmit kbps peak,
 cast(
   sum(cps ave) as decimal(10, 0)
  ) as cps_ave,
 sum(cps_peak) as cps_peak
from
  (
    select
      $flex timescale(timestamp) as hodex,
      devid,
      slot,
      sum(total cpu) / sum(count) cpu ave,
      sum(total mem) / sum(count) as mem ave,
      sum(total disk) / sum(count) as disk ave,
      sum(
       total trate + total erate + total orate
      )/ 100.00 / sum(count) as log rate,
      sum(totalsession) / sum(count) as sessions,
      sum(sent) / sum(count) as sent kbps,
      sum(recv) / sum(count) as recv kbps,
      sum(sent + recv) / sum(count) as transmit kbps,
      max(mem peak) as mem peak,
      max(disk peak) as disk peak,
      max(cpu peak) as cpu peak,
      max(lograte peak) / 100.00 as lograte peak,
      max(session peak) as session peak,
     max(transmit_peak) as transmit_kbps peak,
      sum(cps) / sum(count) as cps ave,
      max(cps peak) as cps peak
    from
```

###(select \$flex_timestamp as timestamp, devid, slot, count(*) as count, sum (coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum(coalesce(cpu, 0)) as total_cpu, max (coalesce(cpu, 0)) as cpu_peak, sum(coalesce(trate, 0)) as total_trate, sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, max(coalesce(trate, 0)+coalesce (erate, 0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalsession, max(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part (bandwidth, '/', 1), '0') as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as recv, max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max(coalesce(setuprate, 0)) as cps_peak from \$log where \$filter and action='perf-stats' group by timestamp, devid, slot)### t where \$filterdrilldown group by hodex, devid, slot) t group by hodex, devid, role) t group by hodex order by hodex

Dataset Name	Description	Log Category
perf-stat-disk-usage-drilldown	Fortigate resource detail timeline	event
select		
hodex,		
cast(
<pre>sum(cpu_ave)/ count(*)</pre>	as decimal(6, 0)	
) as cpu_ave,		
cast(
<pre>sum(mem_ave)/ count(*)) as mem_ave,</pre>	as decimal(6, 0)	
cast(
<pre>sum(disk_ave)/ count(*)</pre>	as decimal(6, 0)	
) as disk_ave,		
cast(
<pre>sum(log_rate)/ count(*)</pre>	as decimal(10, 2)	
) as log_rate,		
cast(
<pre>sum(sessions)/ count(*) .</pre>	as decimal(10, 0)	
) as sessions,		
<pre>cast(sum(sent_kbps)/ count(*</pre>	$\lambda = \alpha + \alpha$	
) as sent_kbps,) as decimal(10, 0)	
cast(
<pre>sum(recv kbps)/ count(*</pre>) as decimal(10, 0)	
) as recv kbps,	, as accimat(10, 0)	
cast(
<pre>sum(transmit_kbps)/ cou</pre>	nt(*) as decimal(10, 0)	
) as transmit kbps,		
max(mem_peak) as mem_peak	,	
max(disk_peak) as disk_pe		
max(cpu_peak) as cpu_peak		
max(lograte_peak) as logr		
<pre>max(session_peak) as sess</pre>	ion_peak,	
<pre>max(transmit_kbps_peak) a</pre>	s transmit_kbps_peak,	
cast(
<pre>sum(cps_ave)/ count(*)</pre>	as decimal(10, 0)	
) as cps_ave,		
<pre>max(cps_peak) as cps_peak</pre>		
from		
(
select		
hodex,		
devid, get fgt role(devid, s	lot) as role	
cast(100, 45 IUIC,	
	(*) as decimal(6, 0)	
) as cpu ave,	(,	
cast(
(

```
sum(mem ave) / count(*) as decimal(6, 0)
  ) as mem_ave,
 cast(
   sum(disk ave) / count(*) as decimal(6, 0)
 ) as disk ave,
 cast(
   sum(log rate) as decimal(10, 2)
 ) as log rate,
 cast(
   sum(sessions) as decimal(10, 0)
 ) as sessions,
 cast(
   sum(sent kbps) as decimal(10, 0)
 ) as sent kbps,
 cast(
   sum(recv kbps) as decimal(10, 0)
 ) as recv_kbps,
 cast(
   sum(transmit kbps) as decimal(10, 0)
 ) as transmit_kbps,
 max(mem_peak) as mem_peak,
 max(disk_peak) as disk_peak,
 max(cpu peak) as cpu peak,
 cast(
   sum(lograte_peak) as decimal(10, 2)
 ) as lograte peak,
 sum(session peak) as session peak,
 sum(transmit kbps peak) as transmit kbps peak,
 cast(
   sum(cps_ave) as decimal(10, 0)
 ) as cps ave,
 sum(cps peak) as cps peak
from
  (
   select
      $flex timescale(timestamp) as hodex,
     devid,
      slot,
      sum(total cpu) / sum(count) cpu ave,
      sum(total mem) / sum(count) as mem ave,
      sum(total_disk) / sum(count) as disk_ave,
      sum(
       total_trate + total_erate + total_orate
      )/ 100.00 / sum(count) as log_rate,
      sum(totalsession) / sum(count) as sessions,
      sum(sent) / sum(count) as sent kbps,
      sum(recv) / sum(count) as recv kbps,
      sum(sent + recv) / sum(count) as transmit_kbps,
      max(mem peak) as mem peak,
      max(disk peak) as disk peak,
     max(cpu peak) as cpu peak,
     max(lograte peak) / 100.00 as lograte peak,
      max(session peak) as session peak,
      max(transmit peak) as transmit kbps peak,
      sum(cps) / sum(count) as cps ave,
      max(cps peak) as cps peak
```

from

Dataset Name	Description	Log Category
perf-stat-sessions-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(sum(log_rate)/ count(*)) as log_rate, cast(sum(sessions)/ count(*)) as sessions, cast(sum(sent_kbps)/ count(*)) as sent_kbps, cast(sum(recv_kbps)/ count(*)) as recv_kbps, cast(</pre>	<pre>as decimal(6, 0) as decimal(6, 0) as decimal(6, 0) as decimal(10, 2) as decimal(10, 0) </pre>	event
<pre>max(mem_peak) as mem_peak max(disk_peak) as disk_peak</pre>	eak,	
<pre>max(cpu_peak) as cpu_peak max(lograte_peak) as logn max(session_peak) as sess max(transmit_kbps_peak) a cast(</pre>	ate_peak, sion_peak, us transmit_kbps_peak,	
<pre>sum(cps_ave)/ count(*)) as cps_ave, max(cps_peak) as cps_peał</pre>		

```
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(
       sum(lograte peak) as decimal(10, 2)
     ) as lograte peak,
     sum(session peak) as session peak,
     sum(transmit kbps peak) as transmit kbps peak,
     cast(
       sum(cps_ave) as decimal(10, 0)
     ) as cps ave,
     sum(cps peak) as cps peak
    from
      (
        select
          $flex timescale(timestamp) as hodex,
         devid,
          slot,
          sum(total cpu) / sum(count) cpu ave,
          sum(total mem) / sum(count) as mem ave,
          sum(total disk) / sum(count) as disk ave,
          sum(
            total trate + total erate + total orate
          )/ 100.00 / sum(count) as log rate,
          sum(totalsession) / sum(count) as sessions,
          sum(sent) / sum(count) as sent kbps,
```

```
sum(recv) / sum(count) as recv_kbps,
sum(sent + recv) / sum(count) as transmit_kbps,
max(mem_peak) as mem_peak,
max(disk_peak) as disk_peak,
max(cpu_peak) as cpu_peak,
max(lograte_peak) / 100.00 as lograte_peak,
max(session_peak) as session_peak,
max(transmit_peak) as transmit_kbps_peak,
sum(cps) / sum(count) as cps_ave,
max(cps_peak) as cps_peak
```

from

Dataset Name	Description	Log Category
perf-stat-lograte-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(*) as) as disk_ave, cast(sum(log_rate)/ count(*) as) as log_rate, cast(sum(sessions)/ count(*) as) as sessions, cast(sum(sent_kbps)/ count(*)) as sent_kbps, cast(sum(recv_kbps)/ count(*)) as recv_kbps, cast(sum(transmit_kbps)/ count) as transmit kbps,</pre>	as decimal(6, 0) as decimal(6, 0) as decimal(10, 2) as decimal(10, 0) as decimal(10, 0) as decimal(10, 0)	

```
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(
       sum(lograte_peak) as decimal(10, 2)
     ) as lograte peak,
     sum(session peak) as session peak,
     sum(transmit_kbps_peak) as transmit_kbps_peak,
     cast(
       sum(cps ave) as decimal(10, 0)
     ) as cps_ave,
      sum(cps peak) as cps peak
    from
      (
        select
          $flex timescale(timestamp) as hodex,
```

```
devid,
  slot,
  sum(total_cpu) / sum(count) cpu_ave,
  sum(total mem) / sum(count) as mem ave,
  sum(total disk) / sum(count) as disk ave,
  sum(
   total trate + total erate + total orate
  )/ 100.00 / sum(count) as log rate,
  sum(totalsession) / sum(count) as sessions,
  sum(sent) / sum(count) as sent_kbps,
  sum(recv) / sum(count) as recv kbps,
  sum(sent + recv) / sum(count) as transmit kbps,
  max(mem peak) as mem peak,
  max(disk peak) as disk peak,
  max(cpu peak) as cpu peak,
  max(lograte peak) / 100.00 as lograte peak,
 max(session peak) as session peak,
  max(transmit peak) as transmit kbps peak,
  sum(cps) / sum(count) as cps ave,
  max(cps peak) as cps peak
from
```

Dataset Name	Description	Log Category
perf-stat-connections-drilldown	Fortigate resource detail timeline	event
<pre>select hodex, cast(sum(cpu_ave)/ count(*) a) as cpu_ave, cast(sum(mem_ave)/ count(*) a) as mem_ave, cast(sum(disk_ave)/ count(*)) as disk_ave, cast(sum(log_rate)/ count(*)) as log_rate, cast(sum(sessions)/ count(*)</pre>	as decimal(6, 0) as decimal(6, 0) as decimal(10, 2)	

```
) 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(
        sum(lograte peak) as decimal(10, 2)
      ) as lograte peak,
```

```
sum(session peak) as session peak,
 sum(transmit kbps peak) as transmit kbps peak,
 cast(
   sum(cps ave) as decimal(10, 0)
 ) as cps ave,
  sum(cps peak) as cps peak
from
  (
   select
      $flex timescale(timestamp) as hodex,
      devid,
      slot,
      sum(total_cpu) / sum(count) cpu_ave,
      sum(total mem) / sum(count) as mem ave,
      sum(total disk) / sum(count) as disk ave,
      sum (
       total trate + total erate + total orate
      )/ 100.00 / sum(count) as log rate,
      sum(totalsession) / sum(count) as sessions,
      sum(sent) / sum(count) as sent kbps,
      sum(recv) / sum(count) as recv_kbps,
      sum(sent + recv) / sum(count) as transmit kbps,
      max(mem peak) as mem peak,
      max(disk peak) as disk peak,
      max(cpu_peak) as cpu_peak,
      max(lograte_peak) / 100.00 as lograte peak,
      max(session peak) as session peak,
      max(transmit peak) as transmit kbps peak,
      sum(cps) / sum(count) as cps ave,
      max(cps peak) as cps peak
    from
```

Dataset Name	Description	Log Category
perf-stat-bandwidth-drilldown	Fortigate resource detail timeline	event
<pre>select hodex, cast(sum(cpu_ave)/ count(*)) as cpu_ave, cast(</pre>	as decimal(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(
   sum(lograte peak) as decimal(10, 2)
 ) as lograte peak,
 sum(session peak) as session peak,
  sum(transmit kbps peak) as transmit kbps peak,
 cast(
   sum(cps ave) as decimal(10, 0)
 ) as cps ave,
 sum(cps peak) as cps peak
from
  (
   select
      $flex timescale(timestamp) as hodex,
      devid,
      slot,
      sum(total cpu) / sum(count) cpu ave,
      sum(total mem) / sum(count) as mem ave,
      sum(total_disk) / sum(count) as disk_ave,
      sum(
       total trate + total erate + total orate
      )/ 100.00 / sum(count) as log_rate,
      sum(totalsession) / sum(count) as sessions,
      sum(sent) / sum(count) as sent kbps,
      sum(recv) / sum(count) as recv kbps,
      sum(sent + recv) / sum(count) as transmit kbps,
      max(mem peak) as mem peak,
      max(disk peak) as disk peak,
      max(cpu peak) as cpu peak,
     max(lograte_peak) / 100.00 as lograte peak,
     max(session peak) as session peak,
      max(transmit peak) as transmit kbps peak,
      sum(cps) / sum(count) as cps ave,
     max(cps_peak) as cps_peak
    from
```

Dataset Name	Description	Log Category
perf-stat-usage-summary-average	Fortigate resource summary view	event
select		
devid,		
<pre>get_fgt_role(devid, slot) as</pre>	s role,	
cast(
<pre>sum(cpu_ave)/ count(*) as</pre>	decimal(6, 0)	
) as cpu_ave,		
cast(
<pre>sum(mem_ave)/ count(*) as</pre>	decimal(0, 0)	
) as mem_ave, cast(
<pre>sum(disk_ave)/ count(*) as</pre>	decimal(6, 0)	
) as disk ave,		
cast(
sum(log_rate) as decimal(1	0, 2)	
) as log_rate,		
cast(
sum(sessions) as decimal(1	.0, 0)	
) as sessions,		
cast(
<pre>sum(sent_kbps) as decimal</pre>	(10, 0)	
) as sent_kbps,		
cast(
<pre>sum(recv_kbps) as decimal</pre>	(10, 0)	
) as recv_kbps,		
cast(
sum(transmit_kbps) as deci	.mal(10, 0)	
) as transmit_kbps,		
<pre>max(mem_peak) as mem_peak, max(dick peak) as dick peak</pre>		
<pre>max(disk_peak) as disk_peak, max(cpu peak) as cpu peak,</pre>		
cast(
sum(lograte_peak) as decim	nal(10, 2)	
) as lograte_peak,		
sum(session peak) as sessior	n peak,	
sum(transmit_kbps_peak) as t	—	
from		
(
select		
devid,		
slot,		
<pre>sum(total_cpu) / sum(cour</pre>	—	
<pre>sum(total_mem) / sum(cour</pre>	—	
<pre>sum(total_disk) / sum(cou </pre>	int) as disk_ave,	
sum(
total_trate + total_er	—	
)/ 100.00 / sum(count) a sum(totalsession)/ sum(c	—	
<pre>sum(totalsession)/ sum(c sum(sent)/ sum(count) as</pre>		
<pre>sum(sent)/ sum(count) as sum(recv)/ sum(count) as</pre>		
<pre>sum(recv)/ sum(count) ac sum(sent + recv)/ sum(count)</pre>		
max(mem peak) as mem pea	—	
<pre>max(disk_peak) as disk_p</pre>		
,,		

```
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
```

Dataset Name	Description	Log Category
perf-stat-usage-summary-peak	Fortigate resource summary view	event
<pre>perf-stat-usage-summary-peak select devid, get_fgt_role(devid, slot) as recast(sum(cpu_ave)/ count(*) as ded) as cpu_ave, cast(sum(mem_ave)/ count(*) as ded) as mem_ave, cast(sum(disk_ave)/ count(*) as ded) as disk_ave, cast(sum(log_rate) as decimal(10,) as log_rate, cast(sum(sent_kbps) as decimal(10,) as sent_kbps, cast(sum(recv_kbps) as decimal(10,) as recv_kbps, cast(sum(transmit_kbps) as decimal) as transmit_kbps, max(mem_peak) as mem_peak, max(disk_peak) as disk_peak, max(cpu peak) as cpu peak,</pre>	<pre>>>le, cimal(6, 0) cimal(6, 0) ecimal(6, 0) 2) 0) , 0) , 0)</pre>	event
<pre>cast(sum(lograte_peak) as decimal) as lograte peak,</pre>	(10, 2)	

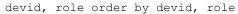
```
sum(session peak) as session peak,
  sum(transmit kbps peak) as transmit kbps peak
from
  (
    select
     devid,
     slot,
     sum(total cpu) / sum(count) as cpu ave,
     sum(total mem) / sum(count) as mem ave,
      sum(total disk) / sum(count) as disk ave,
      sum(
       total trate + total erate + total orate
      )/ 100.00 / sum(count) as log rate,
      sum(totalsession) / sum(count) as sessions,
      sum(sent) / sum(count) as sent kbps,
      sum(recv) / sum(count) as recv kbps,
      sum(sent + recv) / sum(count) as transmit kbps,
     max(mem peak) as mem peak,
     max(disk peak) as disk peak,
     max(cpu peak) as cpu peak,
     max(lograte_peak) / 100.00 as lograte_peak,
     max(session peak) as session peak,
     max(transmit peak) as transmit kbps peak
    from
```

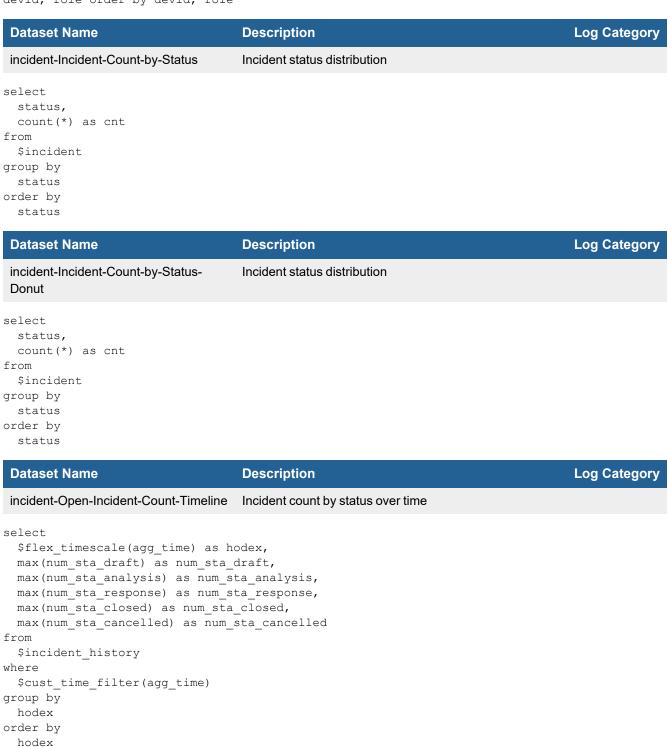
###(select \$flex_timestamp as timestamp, devid, slot, count(*) as count, sum(coalesce (mem, 0)) as total_mem, max(coalesce(mem, 0)) mem_peak, sum(coalesce(disk, 0)) as total_ disk, max(coalesce(disk, 0)) as disk_peak, sum(coalesce(cpu, 0)) as total_cpu, max(coalesce (cpu, 0)) as cpu_peak, sum(coalesce(trate, 0)) as total_trate, sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, max(coalesce(trate, 0)+coalesce(erate, 0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalsession, max (coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as recv, max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast (coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce (setuprate, 0)) as cps, max(coalesce(setuprate, 0)) as cps_peak from \$log where \$filter and action='perf-stats' group by timestamp, devid, slot)### t group by devid, slot) t group by devid, role order by devid, role

Dataset Name	Description	Log Category
perf-stat-usage-details-drilldown- master	Fortigate resource summary view	event
<pre>select devid, get_fgt_role(devid, slot) as r cast(sum(cpu_ave)/ count(*) as de) as cpu_ave, cast(sum(mem_ave)/ count(*) as de) as mem_ave, cast(sum(disk_ave)/ count(*) as d) as disk_ave, cast(</pre>	cimal(6, 0) cimal(6, 0)	

```
sum(log rate) as decimal(10, 2)
  ) as log_rate,
  cast(
    sum(sessions) as decimal(10, 0)
  ) as sessions,
  cast(
   sum(sent kbps) as decimal(10, 0)
  ) as sent kbps,
  cast(
   sum(recv kbps) as decimal(10, 0)
  ) as recv kbps,
  cast(
   sum(transmit kbps) as decimal(10, 0)
  ) as transmit kbps,
  max(mem peak) as mem peak,
 max(disk peak) as disk peak,
 max(cpu peak) as cpu peak,
 cast(
   sum(lograte peak) as decimal(10, 2)
  ) as lograte peak,
  sum(session_peak) as session_peak,
  sum(transmit_kbps_peak) as transmit_kbps_peak
from
  (
    select
     devid,
     slot,
      sum(total cpu) / sum(count) as cpu ave,
     sum(total mem) / sum(count) as mem ave,
     sum(total disk) / sum(count) as disk ave,
     sum(
       total trate + total erate + total orate
     )/ 100.00 / sum(count) as log rate,
      sum(totalsession) / sum(count) as sessions,
      sum(sent) / sum(count) as sent kbps,
      sum(recv) / sum(count) as recv kbps,
      sum(sent + recv) / sum(count) as transmit kbps,
     max(mem peak) as mem peak,
     max(disk peak) as disk peak,
     max(cpu peak) as cpu peak,
     max(lograte peak) / 100.00 as lograte peak,
     max(session peak) as session peak,
     max(transmit peak) as transmit kbps peak
    from
```

###(select \$flex_timestamp as timestamp, devid, slot, count(*) as count, sum(coalesce (mem, 0)) as total_mem, max(coalesce(mem, 0)) mem_peak, sum(coalesce(disk, 0)) as total_ disk, max(coalesce(disk, 0)) as disk_peak, sum(coalesce(cpu, 0)) as total_cpu, max(coalesce (cpu, 0)) as cpu_peak, sum(coalesce(trate, 0)) as total_trate, sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, max(coalesce(trate, 0)+coalesce(erate, 0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalsession, max (coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as recv, max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast (coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce (setuprate, 0)) as cps, max(coalesce(setuprate, 0)) as cps_peak from \$log where \$filter and action='perf-stats' group by timestamp, devid, slot)### t group by devid, slot) t group by





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

```
select
   $flex_timescale(agg_time) as hodex,
   max(num_sta_draft) as num_sta_draft,
   max(num_sta_analysis) as num_sta_analysis,
   max(num_sta_response) as num_sta_response,
   max(num_sta_closed) as num_sta_closed,
   max(num_sta_cancelled) as num_sta_cancelled
from
   $incident_history
where
   $cust_time_filter(agg_time)
group by
   hodex
order by
   hodex
```

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

Dataset Name	Description	Log Category
Top-10-User-by-Bandwidth	Top users by bandwidth usage	traffic
<pre>select coalesce(nullifna(`user`),</pre>		

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

Dataset Name	Description	Log Category
Top-10-Applications-by-Number-o Users	f- Top Applications by number of users	traffic
app, appcat from \$log where		(app) is not null group by
Dataset Name	Description	Log Category
Top-10-User-by-Session	Top user by session count	traffic

```
select
  coalesce(
    nullifna(`user`),
    nullifna(`unauthuser`),
    ipstr(`srcip`)
  ) as user_src,
    count(*) as sessions
from
```

```
$log
where
  $filter
  and (
    logflag&1>0
 )
group by
  user_src
order by
  sessions desc
```

Dataset Name Description Log Category Top-10-Apps-by-Session Top applications by bandwidth usage traffic select app_group_name(app) as app_group, sum(coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic out, count(*) as sessions from \$log where \$filter and (logflag&1>0

```
)
and nullifna(app) is not null
group by
app_group
having
sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
)& gt; 0
order by
bandwidth desc
```

Dataset Name	Description	Log Category
Applications-by-Risk-Level	Applications by Risk Level	traffic
<pre>select app_group_name(app) as app min(id) as id, appcat, max(risk) as d_risk, (</pre>		
	then 'Critical' when max(risk)= '4' trisk)= '2' then 'Low' else 'Info' end	<pre>chen 'High' when max(risk) =</pre>

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

(

```
select
          (
           select
             count(*)
            from
              $incident
            where
              $cust_time_filter(createtime, TODAY)
          ) as num_cur,
          (
            select
             count(*)
           from
              $incident
            where
              $cust_time_filter(createtime, YESTERDAY)
          ) as num_pre
     ) t
 ) t
order by
 item
```

Dataset Name	Description	Log Category
soc-Event-vs-Incident-History-Trend	Events vs Incidents History Trend	
elect		
item,		
num_cur,		
num pre,		
num_diff		
 From		
(
select		
'Events' as item,		
num_cur,		
num_pre,		
(num_cur - num_pre) as nu	m_diff	
from		
(
select		
(
select		
count(*)		
from		
\$event where		
\$cust_time_filter	(alorttime)	
) as num cur,	(dierectime)	
(
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, num_cur, num_pre, (num_cur - num_pre) as num from (select (select (select</pre>		

```
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

```
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
```

Description

```
$event
group by
severity
```

Log Category

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 0 THEN 'Critical' WHEN 1 THEN 'High' WHEN 2 THEN 'Medium' WHEN 3 THEN

```
'Low' ELSE NULL END
) as sev,
triggername,
count(*) as num_events
from
   $event
group by
   severity,
   triggername
order by
   severity,
   triggername
```

Dataset Name	Description	Log Category
soc-Total-Incident-by-Severity	Total Incidents by Severity	
<pre>select severity, count(*) as num_inc from \$incident group by severity order by severity</pre>		
Dataset Name	Description	Log Category
soc-Total-Event-vs-Incident-History	Total Events vs Incidents History	
<pre>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 group by hodex order by hodex</pre>	as num_event_total, num_inc_total, s num_event_high as hodex, nt_total, n_sev_high) as num_event_high	

```
) 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) a from_itime(createtime category, severity, status, endpoint</pre>				
from				
<pre>\$incident where</pre>				
<pre>\$cust_time_filter(cre order by createtime desc</pre>	eatetime)			

Dataset Name	Description	Log Category
fex-RSRQ-timeline	FortiExtender RSRQ timeline	event
<pre>sum, sum(to_number(sinr, '9</pre>		count from \$log where
Dataset Name	Description	Log Category

fex-SINR-timeline	FortiExtender SINR timeline	
<pre>select \$flex_timescale(timestamp) cast(</pre>		
<pre>sum(sinr_sum) / sum(count) 'dB' as sinr from</pre>) as decimal(18, 0)	

event

###(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	
--------------	--

	•	•	
fgt-inventory-hardware	FortiGate Monitoring Inventory Hardware	event	
select			
platform,			
count(*) as total_num			
from			
\$func - fgt - inventory as t1			
where			
exists (
select			
1			
from			
devtable t2			
where			
\$dev_filter			
and t2.devid = t1.devid			
)			
group by			
platform			
order by			
total_num desc			

Description

Log Category

Dataset Name	Description	Log Category
fgt-inventory-software	FortiGate Monitoring Inventory Software	event
<pre>select 'FortiOS' as sf_name, (platform ' ' os) as fir 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</pre>	mware,	
Dataset Name	Description	Log Category
cup-utilization-timeline-for-each-device	FortiGate cpu utilization timeline	event
<pre>select \$flex_timescale(timestamp) as devid, cast(sum(total_cpu)/ sum(count) a</pre>		

```
_cpu)/ sum(count) as decimal(6,
      III (LOLe
 ) as cpu_ave,
 cast(
   sum(total mem) / sum(count) as decimal(6, 0)
 ) as mem_ave,
 cast(
   sum(total disk) / sum(count) as decimal(6, 0)
 ) as disk ave,
 cast(
   sum(sent) / sum(count) as decimal(10, 0)
 ) as sent kbps,
 cast(
   sum(recv) / sum(count) as decimal(10, 0)
 ) as recv kbps
from
```

###(select \$flex_timestamp as timestamp, devid, count(*) as count, sum(coalesce(mem, 0))
as total_mem, max(coalesce(mem, 0)) mem_peak, sum(coalesce(disk, 0)) as total_disk, max
(coalesce(disk, 0)) as disk_peak, sum(coalesce(cpu, 0)) as total_cpu, max(coalesce(cpu, 0))
as cpu_peak, sum(coalesce(trate, 0)) as total_trate, sum(coalesce(erate, 0)) as total_erate,
sum(coalesce(orate, 0)) as total_orate, max(coalesce(trate, 0)+coalesce(erate, 0)+coalesce
(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalsession, max(coalesce
(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part(bandwidth, '/', 1), '0') as

integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as recv, max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_ part(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max(coalesce(setuprate, 0)) as cps_peak from \$log where \$filter and action='perf-stats' group by timestamp, devid)### 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
<pre>select devid, cast(sum(total_cpu) / sum(count)) as cpu_ave, max(cpu_peak) as cpu_peak from</pre>	as decimal(6, 0)	
<pre>###(select min(itime) as first_seen, max(itime) as last_seen, devid, count(*) as count, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum(coalesce(cpu, 0)) as total_cpu, max (coalesce(cpu, 0)) as cpu_peak, sum(coalesce(trate, 0)) as total_trate, sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, max(coalesce(trate, 0)+coalesce (erate, 0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalsession, max(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part (bandwidth, '/', 1), '0') as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/',</pre>		

2), '0') as integer)) as recv, max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max(coalesce(setuprate, 0)) as cps_peak from \$log where \$filter and action='perf-stats' group by devid)### t group by devid order by cpu_peak desc

Dataset Name	Description	Log Category
event-cpu-utilization-dev	FortiGate cpu summary view	event
<pre>select devid, cast(sum(total_cpu)/ sum(cou) as cpu_ave, max(cpu_peak) as cpu_peak from ###(select min(itime) as sum(coalesce(mem, 0)) as to as total_disk, max(coalesce (coalesce(cpu, 0)) as cpu_p 0)) as total_erate, sum(coa (erate, 0)+coalesce(orate, totalsession, max(coalesce((bandwidth, '/', 1), '0') a</pre>	<pre>int) as decimal(6, 0) int first_seen, max(itime) as last_seen, btal_mem, max(coalesce(mem, 0)) mem_p e(disk, 0)) as disk_peak, sum(coalesce beak, sum(coalesce(trate, 0)) as tota clesce(orate, 0)) as total_orate, max 0)) as lograte_peak, sum(coalesce(to ctotalsession, 0)) as session_peak, s as integer)) as sent, sum(cast(coalesce)</pre>	<pre>devid, count(*) as count, peak, sum(coalesce(disk, 0)) ce(cpu, 0)) as total_cpu, max il_trate, sum(coalesce(erate, c(coalesce(trate, 0)+coalesce stalsession, 0)) as sum(cast(coalesce(split_part sce(split_part(bandwidth, '/',</pre>
<pre>2), '0') as integer)) as re integer)+cast(coalesce(spli sum(coalesce(setuprate, 0))</pre>	<pre>as sent, sum(cast(coales ecv, max(cast(coalesce(split_part(ban .t_part(bandwidth, '/', 2), '0') as i as cps, max(coalesce(setuprate, 0)) cats' group by devid)### t group by d</pre>	<pre>idwidth, '/', 1), '0') as integer)) as transmit_peak, as cps_peak from \$log where</pre>

Dataset Name	Description	Log Category
memory-utilization-timeline-for-each- device	FortiGate cpu utilization timeline	event
<pre>select \$flex_timescale(timestamp) as devid, cast(sum(total_cpu)/ sum(count)) as cpu_ave, cast(sum(total_mem)/ sum(count)) as mem_ave, cast(sum(total_disk)/ sum(count)) as disk_ave, cast(sum(sent)/ sum(count) as de) as sent_kbps,</pre>	as decimal(6, 0) as decimal(6, 0) as decimal(6, 0)	
<pre>cast(sum(recv)/ sum(count) as de) as recv_kbps</pre>	cimal(10, 0)	
as total_mem, max(coalesce(mem, (coalesce(disk, 0)) as disk_pea as cpu_peak, sum(coalesce(trate sum(coalesce(orate, 0)) as tota	<pre>timestamp, devid, count(*) as count, 0)) mem_peak, sum(coalesce(disk, 0)) k, sum(coalesce(cpu, 0)) as total_cpu , 0)) as total_trate, sum(coalesce(er l_orate, max(coalesce(trate, 0)+coale n(coalesce(totalsession, 0)) as total</pre>	as total_disk, max , max(coalesce(cpu, 0)) ate, 0)) as total_erate sce(erate, 0)+coalesce

(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as recv, max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_ part(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max(coalesce(setuprate, 0)) as cps_peak from \$log where \$filter and action='perf-stats' group by timestamp, devid)### 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>sum(coalesce(mem, 0)) as total_m</pre>	s decimal(6, 0) seen, max(itime) as last_seen, devid, count(mem, max(coalesce(mem, 0)) mem_peak, sum(coales s, 0)) as disk peak, sum(coalesce(cpu, 0)) as t	sce(disk, 0))
<pre>(coalesce(cpu, 0)) as cpu_peak, 0)) as total_erate, sum(coalesce (erate, 0)+coalesce(orate, 0)) a</pre>	<pre>sum(coalesce(trate, 0)) as total_trate, sum(co (orate, 0)) as total_orate, max(coalesce(trate s lograte_peak, sum(coalesce(totalsession, 0)) session, 0)) as session_peak, sum(cast(coalesce))</pre>	palesce(erate, e, 0)+coalesce) as

(bandwidth, '/', 1), '0') as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as recv, max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max(coalesce(setuprate, 0)) as cps_peak from \$log where \$filter and action='perf-stats' group by devid)### 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)</pre>	as decimal(6, 0)	

```
) as mem ave,
```

max(mem peak) as mem peak

```
from
```

###(select min(itime) as first_seen, max(itime) as last_seen, devid, count(*) as count, sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum(coalesce(cpu, 0)) as total_cpu, max (coalesce(cpu, 0)) as cpu_peak, sum(coalesce(trate, 0)) as total_trate, sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, max(coalesce(trate, 0)+coalesce (erate, 0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalsession, max(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part (bandwidth, '/', 1), '0') as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as recv, max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max(coalesce(setuprate, 0)) as cps_peak from \$log where \$filter and action='perf-stats' group by devid)### 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
<pre>select \$flex_timescale(timestamp) as devid, cast(sum(total_cpu) / sum(count) a) as cpu_ave, cast(sum(total_mem) / sum(count) a) as mem_ave, cast(sum(total_disk) / sum(count)) as disk_ave, cast(sum(sent) / sum(count) as dec) as sent_kbps, cast(sum(recv) / sum(count) as dec) as recv_kbps from</pre>	as decimal(6, 0) as decimal(6, 0) as decimal(6, 0) cimal(10, 0)	
<pre>####(select \$flex_timestamp as</pre>	<pre>timestamp, devid, count(*) as coun 0)) mem_peak, sum(coalesce(disk, 0)</pre>	

(coalesce(disk, 0)) as disk_peak, sum(coalesce(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, sum(coalesce(trate, 0)) as total_trate, sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, max(coalesce(trate, 0)+coalesce(erate, 0)+coalesce (orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalsession, max(coalesce (totalsession, 0)) as session_peak, sum(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as recv, max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_ part(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max(coalesce(setuprate, 0)) as cps_peak from \$log where \$filter and action='perf-stats' group by timestamp, devid)### 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 from</pre>	as decimal(6, 0)	

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

Dataset Name	Description	Log Category
event-disk-utilization-dev	FortiGate disk summary view	event
<pre>sum(coalesce(mem, 0)) as total_ as total_disk, max(coalesce(dis (coalesce(cpu, 0)) as cpu_peak, 0)) as total_erate, sum(coalesce (erate, 0)+coalesce(orate, 0))</pre>		<pre>eak, sum(coalesce(disk, 0)) e(cpu, 0)) as total_cpu, max e_trate, sum(coalesce(erate, coalesce(trate, 0)+coalesce calsession, 0)) as</pre>
(bandwidth, '/', 1), '0') as in	teger)) as sent, sum(cast(coalesc max(cast(coalesce(split part(band	ce(split_part(bandwidth, '/',

integer)+cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max(coalesce(setuprate, 0)) as cps_peak from \$log where \$filter and action='perf-stats' group by devid)### 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_sess cast(sum(totalsession)/ sum(coun) as sessions, max(cps_peak) as cps_peak, cast(sum(cps)/ sum(count) as dec</pre>	t) as decimal(10, 0)	
) as cps_ave from		
<pre>sum(coalesce(mem, 0)) as total_ as total_disk, max(coalesce(dis (coalesce(cpu, 0)) as cpu_peak, 0)) as total_erate, sum(coalesc</pre>	<pre>t_seen, max(itime) as last_seen, d mem, max(coalesce(mem, 0)) mem_pea k, 0)) as disk_peak, sum(coalesce(sum(coalesce(trate, 0)) as total_ e(orate, 0)) as total_orate, max(c as lograte peak, sum(coalesce(tota</pre>	<pre>k, sum(coalesce(disk, 0)) cpu, 0)) as total_cpu, max trate, sum(coalesce(erate, oalesce(trate, 0)+coalesce</pre>
<pre>totalsession, max(coalesce(tota (bandwidth, '/', 1), '0') as in 2), '0') as integer)) as recv, integer)+cast(coalesce(split_pa sum(coalesce(setuprate, 0)) as</pre>	<pre>lsession, 0)) as session_peak, sum teger)) as sent, sum(cast(coalesce max(cast(coalesce(split_part(bandw rt(bandwidth, '/', 2), '0') as int- cps, max(coalesce(setuprate, 0)) a group by devid)### t group by dev</pre>	<pre>(cast(coalesce(split_part (split_part(bandwidth, '/', idth, '/', 1), '0') as eger)) as transmit_peak, s cps_peak from \$log where</pre>

Dataset Name	Description	Log Category
event-session-rate-summary	FortiGate Session Rate	event

select devid, max(cps_peak) as max_rate

```
from
```

desc

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

Dataset Name	Description	Log Category
event-session-summary-dev	FortiGate Total Sessions	event

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

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

Dataset Name	Description	Log Category
fgt-intf-down-timeline-for-each-device	FortiGate Interface Down Timeline	event

select

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

____U

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

Dataset Name	Description	Log Category
fgt-intf-down-timeline-by-device	FortiGate Interface Down by Device	event
where \$filter and logid_to_	um p as timestamp, devid, status, count(*) as int(logid)=20099 and status='DOWN' group b , status order by total_num desc	
Dataset Name	Description	Log Category

Dataset Name	Description	Log Category
fgt-intf-down-dev-donut	FortiGate Interface Down by Device	event

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

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

Dataset Name	Description	Log Category
fgt-intf-down-dev-tbl	FortiGate Interface Down by Device	event

```
select
```

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

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

```
Dataset Name
                                  Description
                                                                                  Log Category
 intf-sent-timeline-for-each-device
                                  FortiGate cpu utilization timeline
                                                                                  event
select
  $flex timescale(timestamp) as hodex,
  devid,
  cast(
    sum(total_cpu) / sum(count) as decimal(6, 0)
  ) as cpu ave,
  cast(
    sum(total mem) / sum(count) as decimal(6, 0)
  ) as mem ave,
  cast(
    sum(total disk) / sum(count) as decimal(6, 0)
  ) as disk ave,
  cast(
    sum(sent) / sum(count) as decimal(10, 0)
  ) as sent kbps,
  cast(
    sum(recv) / sum(count) as decimal(10, 0)
  ) as recv kbps
from
  ###(select $flex timestamp as timestamp, devid, count(*) as count, sum(coalesce(mem, 0))
as total_mem, max(coalesce(mem, 0)) mem_peak, sum(coalesce(disk, 0)) as total_disk, max
(coalesce(disk, 0)) as disk peak, sum(coalesce(cpu, 0)) as total cpu, max(coalesce(cpu, 0))
as cpu peak, sum(coalesce(trate, 0)) as total trate, sum(coalesce(erate, 0)) as total erate,
sum(coalesce(orate, 0)) as total orate, max(coalesce(trate, 0)+coalesce(erate, 0)+coalesce
(orate, 0)) as lograte peak, sum(coalesce(totalsession, 0)) as totalsession, max(coalesce
(totalsession, 0)) as session peak, sum(cast(coalesce(split part(bandwidth, '/', 1), '0') as
integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as
recv, max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_
part(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as
```

cps, max(coalesce(setuprate, 0)) as cps_peak from \$log where \$filter and action='perf-stats' group by timestamp, devid)### 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 d) as sent_kbps, cast(sum(recv) / sum(count) as d) as recv_kbps, cast(sum(sent + recv) / sum(count)) as transmit kbps,</pre>	lecimal(10, 0)	
max(transmit_peak) as transm	it_kbps_peak	
from		
###(select min(itime) as fir	st_seen, max(itime) as last_seen, de	evid, count(*) as count,
<pre>sum(coalesce(mem, 0)) as total</pre>	_mem, max(coalesce(mem, 0)) mem_pea	k, sum(coalesce(disk, 0))
as total disk, max(coalesce(di	sk. ()) as disk peak, sum(coalesce()	cpu, 0)) as total cpu, max

sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum(coalesce(cpu, 0)) as total_cpu, max (coalesce(cpu, 0)) as cpu_peak, sum(coalesce(trate, 0)) as total_trate, sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, max(coalesce(trate, 0)+coalesce (erate, 0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalsession, max(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part (bandwidth, '/', 1), '0') as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as recv, max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max(coalesce(setuprate, 0)) as cps_peak from \$log where \$filter and action='perf-stats' group by devid)### t group by devid order by transmit_kbps_ peak desc

Dataset Name	Description	Log Category
intf-recv-timeline-for-each-device	FortiGate cpu utilization timeline	event
<pre>select \$flex_timescale(timestamp) a devid, cast(sum(total_cpu) / sum(count)) as cpu_ave, cast(sum(total_mem) / sum(count)) as mem_ave, cast(sum(total_disk) / sum(count)) as disk_ave, cast(sum(sent) / sum(count) as d) as sent_kbps, cast(sum(recv) / sum(count) as d) as recv kbps</pre>	as decimal(6, 0) as decimal(6, 0)) as decimal(6, 0) lecimal(10, 0)	

from

###(select \$flex_timestamp as timestamp, devid, count(*) as count, sum(coalesce(mem, 0))
as total_mem, max(coalesce(mem, 0)) mem_peak, sum(coalesce(disk, 0)) as total_disk, max
(coalesce(disk, 0)) as disk_peak, sum(coalesce(cpu, 0)) as total_cpu, max(coalesce(cpu, 0))
as cpu_peak, sum(coalesce(trate, 0)) as total_trate, sum(coalesce(erate, 0)) as total_erate,
sum(coalesce(orate, 0)) as total_orate, max(coalesce(trate, 0)+coalesce(erate, 0)+coalesce
(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as totalsession, max(coalesce
(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part(bandwidth, '/', 1), '0') as
integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as
recv, max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)) as
recv, max(coalesce(setuprate, 0)) as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as
cps, max(coalesce(setuprate, 0)) as cps_peak from \$log where \$filter and action='perf-stats'
group by timestamp, devid)### 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 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 ###(select min(itime) as first sum(coalesce(mem, 0)) as total_mu as total_disk, max(coalesce(disk (coalesce(cpu, 0)) as cou_peak, so (coalesce(cpu, 0)) as cpu_peak, so (coalesce(cpu, 0)) as cpu_peak, so (coalesce(coate, 0)) as totalsession, max(coalesce(totals (bandwidth, '/', 1), '0') as inter 2), '0') as integer)) as recv, mu integer)+cast(coalesce(split_par- sum(coalesce(setuprate, 0)) as cpu_peak, so (coalesce(setuprate, 0)) as cpu_peak</pre>	imal(10, 0) imal(10, 0) as decimal(10, 0)	id, count(*) as count, sum(coalesce(disk, 0)) a, 0)) as total_cpu, max ate, sum(coalesce(erate, lesce(trate, 0)+coalesce ession, 0)) as ast(coalesce(split_part blit_part(bandwidth, '/', ch, '/', 1), '0') as er)) as transmit_peak, cps_peak from \$log where

Dataset Name	Description	Log Category
event-intf-summary-dev	FortiGate interface summary view	event
<pre>select devid, cast(sum(sent) / sum(count)) as sent_kbps, cast(sum(recv) / sum(count)</pre>		

```
) as recv_kbps,
cast(
    sum(sent + recv) / sum(count) as decimal(10, 0)
) as transmit_kbps,
    max(transmit_peak) as transmit_kbps_peak
from
```

```
###(select min(itime) as first_seen, max(itime) as last_seen, devid, count(*) as count,
sum(coalesce(mem, 0)) as total_mem, max(coalesce(mem, 0)) mem_peak, sum(coalesce(disk, 0))
as total_disk, max(coalesce(disk, 0)) as disk_peak, sum(coalesce(cpu, 0)) as total_cpu, max
(coalesce(cpu, 0)) as cpu_peak, sum(coalesce(trate, 0)) as total_trate, sum(coalesce(erate,
0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, max(coalesce(trate, 0)+coalesce
(erate, 0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce(totalsession, 0)) as
totalsession, max(coalesce(totalsession, 0)) as session_peak, sum(cast(coalesce(split_part
(bandwidth, '/', 1), '0') as integer)) as sent, sum(cast(coalesce(split_part(bandwidth, '/',
2), '0') as integer)) as recv, max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as
integer)+cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as transmit_peak,
sum(coalesce(setuprate, 0)) as cps, max(coalesce(setuprate, 0)) as cps_peak from $log where
$filter and action='perf-stats' group by devid)### 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		
hodex,		
dev_intf,		
kbps_out_avg,		
kbps_in_avg,		
util_out_avg,		
util_in_avg from		
(
select		
\$flex timescale(tmstamp)	as hodex,	
(devname ':' intfr		
cast(_	
<pre>sum(bps_out) / sum(inte</pre>	erval)/ 1000 as decimal(10, 0)	
) as kbps_out_avg,		
cast(
	cval)/ 1000 as decimal(10, 0)	
) as kbps_in_avg,		
cast(
—	cerval)/ 100 as decimal(10, 2)	
) as util_out_avg, cast(
	erval)/ 100 as decimal(10, 2)	
) as util in avg		
from		
(
select		
<pre>\$flex_timestamp(time</pre>	estamp) as tmstamp,	
tbl_intf.dvid,		
intfname,		
sum(interval) as int		
sum(sentbps * interv	/al) as ps_out,	

```
sum(rcvdbps * interval) as bps_in,
sum(sentutil * interval) as util_out,
sum(rcvdutil * interval) as util_in
from
```

###(select distinct dvid from \$log-event where \$filter and action='perf-stats')###
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 hodex, dev_intf) t where \$filter-drilldown order by hodex

Dataset Name	Description	Log Category
fgt-intf-stats-timeline-util-in	FortiGate Interface Received Utilization	event
select		
<pre>(devname ':' intfnam cast(</pre>	e) as dev_intf,	
<pre>sum(bps_out) / sum(interv</pre>	al)/ 1000 as decimal(10, 0)	
) as kbps_out_avg,		
cast (1) (1000 - a decimal (10 - 0))	
) as kbps in avg,	1)/ 1000 as decimal(10, 0)	
cast(
<pre>sum(util_out) / sum(inter</pre>	val)/ 100 as decimal(10, 2)	
) as util_out_avg,		
cast(
	al)/ 100 as decimal(10, 2)	
) as util_in_avg From		
(
select		
<pre>\$flex_timestamp(timest</pre>	amp) as tmstamp,	
tbl_intf.dvid,		
intfname,		
<pre>sum(interval) as inter sum(sentbps * interval</pre>		
sum(rcvdbps * interval		
sum(sentutil * interva		
sum(rcvdutil * interva	l) as util_in	
from		
-	id from \$log-event where \$filter and action tbl intf on tbl log.dvid = tbl intf.dvid w	-

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 hodex, dev_intf, kbps_out_avg, kbps_in_avg, util_out_avg, util_in_avg</pre>		

```
from
  (
   select
     $flex timescale(tmstamp) as hodex,
      (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 \$log-event where \$filter and action='perf-stats')###
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 hodex, dev intf) t where \$filter-drilldown order by hodex

Dataset Name	Description	Log Category
fgt-intf-stats-timeline-util-out	FortiGate Interface Sent Utilization	event
<pre>select (devname ':' intfnam cast(sum(bps_out)/ sum(interv.) as kbps_out_avg, cast(sum(bps_in)/ sum(interva) as kbps_in_avg,</pre>		event
) as util_out_avg, cast(val)/ 100 as decimal(10, 2) al)/ 100 as decimal(10, 2)	
<pre>`select \$flex_timestamp(timest</pre>	amp) 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 \$log-event where \$filter and action='perf-stats')###
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
                                   FortiGate Interface Statistics Timeline
fgt-intf-stats-timeline-bit-rate-in-each
                                                                                    event
select
 hodex,
 dev intf,
 kbps out avg,
 kbps in avg,
 util out avg,
 util in avg
from
  (
   select
      $flex timescale(tmstamp) as hodex,
      (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 $log-event where $filter and action='perf-stats')###
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 hodex, dev intf) t where $filter-drilldown order by hodex
 Dataset Name
                                  Description
                                                                                  Log Category
                                  FortiGate Interface Received Bit Rate
fgt-intf-stats-timeline-bit-rate-in
                                                                                  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 $log-event where $filter and action='perf-stats')###
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 hodex, dev_intf, kbps_out_avg, kbps_in_avg, util_out_avg, util_in_avg from (select \$flex_timescale(tmstamp) a: (devname ':' intfname cast(sum(bps_out)/ sum(interve)</pre>		

```
) 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 $log-event where $filter and action='perf-stats')###
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 hodex, dev_intf) t where $filter-drilldown order by hodex
```

Dataset Name	Description	Log Category
fgt-intf-stats-timeline-bit-rate-out	FortiGate Interface Sent Bit Rate	event
elect		
<pre>(devname ':' intfname) cast(</pre>	as dev_intf,	
<pre>sum(bps_out) / sum(interval) as kbps_out_avg,</pre>)/ 1000 as decimal(10, 0)	
<pre>cast(sum(bps in) / sum(interval)</pre>	/ 1000 as decimal(10, 0)	
) as kbps_in_avg,	, 1000 ab accimat (10 , 0,	
cast(
<pre>sum(util_out) / sum(interva) as util_out aug</pre>	1)/ 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	,,	
rom – –		
(
select		
<pre>\$flex_timestamp(timestar</pre>	np) as tmstamp,	
tbl_intf.dvid,		
intfname,		
sum(interval) as interva	•	
<pre>sum(sentbps * interval)</pre>		
<pre>sum(rcvdbps * interval)</pre>		
<pre>sum(sentutil * interval) </pre>	—	
<pre>sum(rcvdutil * interval)</pre>	as utii_in	

from

from

###(select distinct dvid from \$log-event where \$filter and action='perf-stats')###
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

###(select distinct dvid from \$log-event where \$filter and action='perf-stats')###
tbl_log inner join intfstats tbl_intf on tbl_log.dvid = tbl_intf.dvid where \$cust_time_
filter(timestamp) group by tmstamp, tbl_intf.dvid, intfname) t1 left join devtable t2 on
t1.dvid = t2.dvid group by dev_intf order by util_in_avg desc, kbps_in_avg desc, kbps_out_
avg desc

Dataset Name	Description	Log Category
fgt-ha-failure-timeline	FortiGate HA Failure Timeline	event
<pre>select \$flex_timescale(timestamp) count(*) as total_num from</pre>	as hodex,	
<pre>###(select \$flex_timestamp as timestamp, dtime, devid, coalesce(nullifna(logdesc), msg) a 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</pre>		

sum(rcvdutil * interval) as util in

Dataset Name	Description	Log Category
fgt-ha-failure-summary	FortiGate HA Failure Summary	event
msg_desc from \$log where \$filter	timestamp, dtime, devid, coalesce(n and subtype='ha' and logid_to_int(8, 37901, 37902, 37907, 37908) orde	(logid) in (35011, 35012,

Dataset Name	Description	Log Category
fgt-env-faults-power	FortiGate Power Supply Faults	event
select		
from_dtime(dtime) as tim	ne_s,	
devid,		
coalesce(
nullifna(logdesc),		
msg		
) as msg_desc		
from		
\$log		
where		
\$filter		
and logid_to_int(logid)	in (22105, 22107)	
order by		
time_s desc		
Dataset Name	Description	Log Category
fgt-env-faults-fan	FortiGate Fan Faults	event
select		

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

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

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

Dataset Name	Description	Log Category
Behaviour-Banned-Application	Bullying Chat Search and Message Logging	app-ctrl
<pre>select filename, string_agg(distinct app, ' ' string_agg(distinct from_itime(itime) ' ') as itime_agg, string_agg(distinct user_srowstring_agg(distinct `group`, string_agg(distinct ipstr(`srcip`), ' '</pre>	: :text, ; ' ') as user_agg,	
<pre>(`srcip`)) as user_src, `group ('facebook_post', 'facebook_ch 'gmail_send.message', 'linkedi</pre>	<pre>cime, coalesce(nullifna(`user`), nullifna(`u o`, `srcip` from \$log where \$filter and (low hat', 'twitter_post', 'youtube_video.access' n_post', 'vimeo_video.access', 'google.sear and (\$bully_keywords) order by itime desc)# sc</pre>	<pre>ver(app) in , 'gmail_chat', cch_search.phrase',</pre>

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, ' ') as user_agg,	

```
' '
) as srcip_agg,
  count(*) as requests
from
```

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

Dataset Name	Description	Log Category
Behaviour-Banned-User-Drilldown	Bullying Chat Search and Message Logging	app-ctrl
<pre>select filename, string_agg(distinct app, ' ') string_agg(distinct from_itime(itime): ' '</pre>		
) as itime_agg, string_agg(distinct user_src, string_agg(distinct `group`, string_agg(distinct ipstr(`srcip`),		
) as srcip_agg, count(*) as requests		
from		
<pre>(`srcip`)) as user_src, `group` ('facebook_post', 'facebook_cha</pre>	<pre>me, coalesce(nullifna(`user`), nullifna(`` , `srcip` from \$log where \$filter and (lo t', 'twitter_post', 'youtube_video.access post', 'vimeo video.access', 'google.sea</pre>	wer(app) in ', 'gmail_chat',

'gmail_send.message', 'linkedin_post', 'vimeo_video.access', 'google.search_search.phrase', 'bing.search_search.phrase')) and (\$bully_keywords) order by itime desc)### t group by filename order by requests desc

Dataset Name	Description	Log Category
behaviour-banned	Bullying Chat Search and Message Logging	app-ctrl
<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, , ' ') as user_agg,	
) as srcip_agg, count(*) as requests from		

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

Dataset Name	Description	Log Category
Self-Harm-Behaviour-Banned-User- Pie	Self-Harm Chat Search and Message Logging	app-ctrl
<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>	<pre>:text, ' ') as user_agg,</pre>	
) as srcip_agg, count(*) as requests		
<pre>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</pre>		

```
filename order by requests desc
```

Dataset Name	Description	Log Category
Self-Harm-Behaviour-Banned- Application-Pie	Self-Harm Chat Search and Message Logging	app-ctrl
<pre>select filename, string_agg(distinct app, ' ') string_agg(distinct from_itime(itime) ' '</pre>		
<pre>) as itime_agg, string_agg(distinct user_src, string_agg(distinct `group`, string_agg(distinct ipstr(`srcip`), ' '</pre>		
	ime, coalesce(nullifna(`user`), nullifna(`unau `, `srcip` from \$log where \$filter and (lower	

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

Dataset Name	Description	Log Category
Self-Harm-Behaviour-Banned-User- Bar	Self-Harm Chat Search and Message Logging	app-ctrl
<pre>(`srcip`)) as user_src, `group` ('facebook_post', 'facebook_cha' 'gmail_send.message', 'linkedin</pre>	<pre>:text, ' ') as user_agg, ' ') as group_agg, me, coalesce(nullifna(`user`), nullifna(`u , `srcip` from \$log where \$filter and (low t', 'twitter_post', 'youtube_video.access' _post', 'vimeo_video.access', 'google.sear nd (\$banned_keywords) order by itime desc)</pre>	<pre>rer(app) in , 'gmail_chat', ch_search.phrase',</pre>
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, ' ') string_agg(distinct from_itime(itime): ' ') as itime_agg, string_agg(distinct user_src, string_agg(distinct `group`, string_agg(</pre>	:text, ' ') as user_agg,	

###(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',

distinct ipstr(`srcip`),

. .

from

) as srcip_agg, count(*) as requests 'bing.search_search.phrase')) and (\$banned_keywords) order by itime desc)### t group by filename order by requests desc

Dataset Name	Description	Log Category
Self-Harm-behaviour-banned	Self-Harm Chat Search and Message Logging	g app-ctrl
<pre>(`srcip`)) as user_src, `grou ('facebook_post', 'facebook_c' 'gmail_send.message', 'linked'</pre>	<pre>a): :text, rc, ' ') as user_agg, `, ' ') as group_agg, itime, coalesce(nullifna(`user`), nullifn up`, `srcip` from \$log where \$filter and chat', 'twitter_post', 'youtube_video.acc din_post', 'vimeo_video.access', 'google) and (\$banned_keywords) order by itime of)</pre>	<pre>(lower(app) in cess', 'gmail_chat', .search_search.phrase',</pre>
Dataset Name	Description	Log Category
Browsing-Time-per-Social-Media	Browsing Time vs. Domain	traffic
as bandwidth from (select app nullifna(`unauthuser`), ipstr (hostname)), ipstr(dstip), NU (coalesce(sentbyte, 0)+coales (logflag&1>0) group by app_gr mdata t2 on lower(t1.app_grou	<pre>srcip, ebtr_agg_flat(browsetime) as brow p_group_name(app) as app_group, coalesce r(`srcip`)) as f_user, srcip, coalesce(nu JLL) as domain, ebtr_agg_flat(\$browse_timesce(rcvdbyte, 0)) as bandwidth from \$log roup, f_user, hostname, domain, srcip, desce(t2.name) where app_cat='Social etime, bandwidth desc)### t where browset</pre>	<pre>(nullifna(`user`), ullifna(root_domain me) as browsetime, sum where \$filter and stip) t1 inner join app_ .Media' group by domain,</pre>

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

by domain order by browsetime desc

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

###(select domain, f_user, srcip, ebtr_agg_flat(browsetime) as browsetime, sum(bandwidth) as bandwidth from (select app_group_name(app) as app_group, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as f_user, srcip, coalesce(nullifna(root_domain (hostname)), ipstr(dstip), NULL) as domain, ebtr_agg_flat(\$browse_time) as browsetime, sum (coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from \$log where \$filter and (logflag&1>0) group by app_group, f_user, hostname, domain, srcip, dstip) t1 inner join app_ mdata t2 on lower(t1.app_group)=lower(t2.name) where app_cat='Social.Media' group by domain, f_user, srcip order by browsetime, bandwidth desc)### t where bandwidth>0 group by f_user

Dataset Name	Description	Log Category	
Top-Social-Networking-Durations- Sources-Drilldown	Top Social Networking Durations from Sources Drilldown	traffic	
<pre>select f_user, ebtr_value(ebtr_agg_flat(browsetime), null, \$timespan) as browsetime from</pre>			
<pre>from ###(select domain, f_user, srcip, ebtr_agg_flat(browsetime) as browsetime, sum(bandwidth) as bandwidth from (select app_group_name(app) as app_group, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as f_user, srcip, coalesce(nullifna(root_domain (hostname)), ipstr(dstip), NULL) as domain, ebtr_agg_flat(\$browse_time) as browsetime, sum (coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from \$log where \$filter and (logflag&1>0) group by app_group, f_user, hostname, domain, srcip, dstip) t1 inner join app_ mdata t2 on lower(t1.app_group)=lower(t2.name) where app_cat='Social.Media' group by domain, f_user, srcip order by browsetime, bandwidth desc)### t where \$filter-drilldown and</pre>			

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>as bandwidth from (select app_gr nullifna(`unauthuser`), ipstr(`s (hostname)), ipstr(dstip), NULL) (coalesce(sentbyte, 0)+coalesce()</pre>	<pre>ip, ebtr_agg_flat(browsetime) as browse roup_name(app) as app_group, coalesce(nul rcip`)) as f_user, srcip, coalesce(null as domain, ebtr_agg_flat(\$browse_time) rcvdbyte, 0)) as bandwidth from \$log wh o, f_user, hostname, domain, srcip, dsti</pre>	ullifna(`user`), lifna(root_domain) as browsetime, sum here \$filter and

mdata t2 on lower(t1.app_group)=lower(t2.name) where app_cat='Social.Media' group by domain, f_user, srcip order by browsetime, bandwidth desc)### t where browsetime is not null group by domain order by browsetime desc

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

Dataset Name	Description	Log Category
Facebook-Chats	Facebook Chats	app-ctrl
—	er_src, ' ') as user_agg, rroup`, ' ') as group_agg,	
(`srcip`)) as user_src,	time, coalesce(nullifna(`user`), nullif `group`, srcip from \$log where \$filter lename is not null)### t group by filer	and lower(app)=lower

Dataset Name	Description	Log Category
Twitter-Posts	Twitter Posts	app-ctrl
<pre>select from_itime(itime) as i_time, coalesce(nullifna(`user`),</pre>		

```
nullifna(`unauthuser`),
    ipstr(`srcip`)
) as f_user,
    srcip,
    filename
from
    $log
where
    $filter
    and lower(app) = lower('Twitter_Post')
    and filename is not null
order by
    i_time desc
```

Dataset Name	Description	Log Category
LinkedIn-Posts-and-Comments	LinkedIn Posts and Comments	app-ctrl
<pre>select filename, string_agg(distinct from_itime(itim ' ') as itime_agg, string_agg(distinct user_s string_agg(distinct `group string_agg(distinct ipstr(srcip), ' ') as srcip_agg, count(*) as requests from</pre>	erc, ' ') as user_agg,	
(`srcip`)) as user_src, `grc	e, coalesce(nullifna(`user`), nullifna(`una pup`, srcip from \$log where \$filter and low me is not null)### t group by filename ord	wer(app)=lower
(`srcip`)) as user_src, `grc	oup`, srcip from \$log where \$filter and low	wer(app)=lower
(`srcip`)) as user_src, `grc ('LinkedIn_Post') and filena	<pre>oup`, srcip from \$log where \$filter and low me is not null)### t group by filename or</pre>	wer(app)=lower der by requests desc

\$log where \$filter

```
and (
    vwlname is not null
    or vwlservice is not null
)
and (
    logflag&1>0
)
group by
rulename
having
sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
)& gt; 0
order by
bandwidth desc
```

Dataset Name	Description	Log Category
sdwan-Bandwidth-Summary-by- VWLservice-Pie	Total Bandwidth by SD-WAN Service	traffic
<pre>select coalesce(vwlname, vwlservic sum(coalesce(sentbyte, 0)+ cd) as bandwidth, sum(coalesce(sentbyte, 0)) as traffic_out, sum(coalesce(rcvdbyte, 0)) as traffic_in from \$log where \$filter and (vwlname is not null or vwlservice is not null) and (logflag&1>0) group by rulename having sum(coalesce(sentbyte, 0)+ cd) & gt; 0 order by bandwidth desc</pre>	balesce(rcvdbyte, 0)	
Dataset Name	Description	Log Category

Dataset Name	Description	Log Calegory
sdwan-Bandwidth-Detail-by- VWLservice	Total Bandwidth by SD-WAN Service	traffic

```
select
  coalesce(vwlname, vwlservice) as rulename,
  sum(
   coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
  ) as bandwidth,
  sum(
   coalesce(sentbyte, 0)
  ) as traffic out,
  sum(
    coalesce(rcvdbyte, 0)
  ) as traffic in
from
  $log
where
 $filter
  and (
   vwlname is not null
   or vwlservice is not null
  )
  and (
  logflag&1>0
  )
group by
 rulename
having
 sum(
   coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
 )& gt; 0
order by
 bandwidth desc
```

sdwan-Bandwidth-Detail-by- WULservice-Drilldown Total Bandwidth by SD-WAN Service traffic select coalesce(vwlname, vwlservice) as rulename, sum(coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)) as bandwidth, sum(Dataset Name	Description	Log Category
<pre>coalesce(vwlname, vwlservice) as rulename, sum(coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(sentbyte, 0)) as traffic_out, sum(coalesce(rcvdbyte, 0)) as traffic_in from \$log where \$filter and (vwlname is not null or vwlservice is not null)</pre>	-	Total Bandwidth by SD-WAN Service	traffic
ana (<pre>coalesce(vwlname, vwlservi sum(coalesce(sentbyte, 0)+ of) as bandwidth, sum(coalesce(sentbyte, 0)) as traffic_out, sum(coalesce(rcvdbyte, 0)) as traffic_in from \$log where \$filter and (vwlname is not null</pre>	coalesce(rcvdbyte, 0)	

```
logflag&1>0
)
group by
rulename
having
sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
)& gt; 0
order by
bandwidth desc
```

Dataset Name	Description	Log Category
sdwan-VWLservice-by-Firewall- Interface	SD-WAN Service by Firewall and Interface	traffic
<pre>select devid, srcintf, string_agg(distinct dstintf, sum(sessions) as sessions, sum(bandwidth) as bandwidth</pre>	', ') as dstintf,	

from

###(select devid, srcintf, dstintf, coalesce(vwlname,vwlservice) as rulename, count(*) as sessions, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce (sentbyte, 0)) as traffic_out, sum(coalesce(rcvdbyte, 0)) as traffic_in from \$log where \$filter and (vwlname is not null or vwlservice is not null) and (logflag&1>0) group by devid, srcintf, dstintf, rulename)### t where \$filter-drilldown group by devid, srcintf order by bandwidth desc

Dataset Name	Description	Log Category
sdwan-Health-Check-Changes- Timeline	SD-WAN Health Check Changes Timeline	event
<pre>((.*?)\\) ') as hcheck, devid, logdesc='Virtual WAN Link statu</pre>		Eilter and er by num_hcheck
order by timescale		
order by timescale Dataset Name	Description	Log Category
-	Description SD-WAN Service Utilization by Bandwidth	Log Category traffic

```
dvid,
string_agg(distinct dstintf, ', ') as dstintf,
sum(bandwidth) as bandwidth
from
    ###(select app_group_name(app) as app_group, dvid, dstintf, sum(coalesce(sentbyte,
0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(sentbyte, 0)) as traffic_out, sum
(coalesce(rcvdbyte, 0)) as traffic_in, count(*) as sessions from $log where $filter and
(vwlname is not null or vwlservice is not null) and app is not null and (logflag&1>0) group
by app_group, dvid, dstintf having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order
by bandwidth desc)### t where $filter-drilldown group by dvid) t1 left join devtable t2 on
t1.dvid = t2.dvid group by devname, dstintf order by bandwidth desc
```

Dataset Name	Description	Log Category
sdwan-drilldown-Service-Utilization-by- Sessions	SD-WAN Service Utilization by Session	traffic
<pre>select (devname ':' dstintf) as sum(sessions) as sessions from (select dvid, string_agg(distinct dstintf sum(sessions) as sessions</pre>	_	

###(select app_group_name(app) as app_group, dvid, dstintf, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(sentbyte, 0)) as traffic_out, sum (coalesce(rcvdbyte, 0)) as traffic_in, count(*) as sessions from \$log where \$filter and (vwlname is not null or vwlservice is not null) and app is not null and (logflag&l>0) group by app_group, dvid, dstintf having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc)### t where \$filter-drilldown group by dvid) t1 left join devtable t2 on t1.dvid = t2.dvid group by devname, dstintf order by sessions desc

Dataset Name	Description	Log Category
sdwan-drilldown-Bandwidth-Trend	SD-WAN Bandwidth Trend	traffic

select

```
$flex_timescale(timestamp) as timescale,
    sum(bandwidth) as bandwidth
from
```

###(select app_group_name(app) as app_group, \$flex_timestamp as timestamp, sum(coalesce (sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(sentbyte, 0)) as traffic_ out, sum(coalesce(rcvdbyte, 0)) as traffic_in from \$log where \$filter and (vwlname is not null or vwlservice is not null) and app is not null and (logflag&1>0) group by app_group, timestamp having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc)### t where \$filter-drilldown group by timescale order by timescale

Dataset Name	Description	Log Category
sdwan-Top-Application-VWLservice-	SD-WAN Top Applications by Bandwidth	traffic
Summary-by-Bandwidth		

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

###(select app_group_name(app) as app_group, dstintf, sum(coalesce(sentbyte, 0)+coalesce (rcvdbyte, 0)) as bandwidth, sum(coalesce(sentbyte, 0)) as traffic_out, sum(coalesce (rcvdbyte, 0)) as traffic_in, count(*) as sessions from \$log where \$filter and (vwlname is not null or vwlservice is not null) and app is not null and (logflag&1>0) group by app_ group, dstintf having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc)### t where \$filter-drilldown group by app_group order by bandwidth desc

Dataset Name	Description	Log Category
sdwan-Top-Application-over- VWLservice-by-Bandwidth	SD-WAN Top Applications by Bandwidth	traffic

select

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

from

###(select app_group_name(app) as app_group, dstintf, sum(coalesce(sentbyte, 0)+coalesce (rcvdbyte, 0)) as bandwidth, sum(coalesce(sentbyte, 0)) as traffic_out, sum(coalesce (rcvdbyte, 0)) as traffic_in, count(*) as sessions from \$log where \$filter and (vwlname is not null or vwlservice is not null) and app is not null and (logflag&1>0) group by app_ group, dstintf having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc)### t where \$filter-drilldown group by app_group order by bandwidth desc

```
Dataset Name
                                  Description
                                                                                 Log Category
                                 SD-WAN Device-Interface Statistic
sdwan-fw-Device-Interface-Stat
                                                                                 event
 Bibandwidth-drilldown
select
 devid,
  sum(bibandwidth) / sum(count) as bibandwidth
from
  ###(select $flex timestamp as timestamp, devid, interface, min(latency) as latency, max
(latency) as latency max, avg(latency) as latency avg, min(jitter) as jitter, max(jitter) as
jitter max, avg(jitter) as jitter avg, min(packetloss) as packetloss, max(packetloss) as
packetloss_max, avg(packetloss) as packetloss_avg, sum(inbandwidth) as inbandwidth, sum
(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count from
(select itime, devid, interface, cast((substring(msg,'Latency: (\\d+\\.?\\d*), ')::float) as
decimal(18,2)) as latency, cast(substring(msg,'jitter: (\\d+\\.?\\d*), ')::float as decimal
(18,2)) as jitter, cast(substring(msg,'packet loss: (\d+\.?\d*), ') as decimal(18,2)) as
packetloss, (format2bytes(substring(msg,'inbandwidth: (\\d+\\.?\\d*
[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as inbandwidth, (format2bytes(substring(msg,'outbandwidth:
(\\d+\\.?\\d*[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as outbandwidth, (format2bytes(substring
(msg, 'bibandwidth: (\\d+\\.?\\d*[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as bibandwidth from $log
where $filter and logid to int(logid)=22925 and msg is not null) t group by timestamp,
devid, interface /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where $filter-
drilldown and bibandwidth is not null group by devid order by bibandwidth desc
```

Dataset Name

Dataset Name	Description	Log Category
sdwan-fw-Device-Interface-Quality_ Bibandwidth-drilldown	SD-WAN Device-Interface Statistic	event
<pre>(latency) as latency_max, avg(la jitter_max, avg(jitter) as jitte packetloss_max, avg(packetloss) (outbandwidth) as outbandwidth, (select itime, devid, interface, decimal(18,2)) as latency, cast((18,2)) as jitter, cast(substrin packetloss, (format2bytes(substrin [k,K,M,m,g,G,t,T,p,P,e,E])bps')) (\\d+\\.?\\d*[k,K,M,m,g,G,t,T,p, (msg,'bibandwidth: (\\d+\\.?\\d* where \$filter and logid_to_int(1) devid, interface /*SkipSTART*/or</pre>	<pre>s bibandwidth timestamp, devid, interface, min(latency tency) as latency_avg, min(jitter) as ji r_avg, min(packetloss) as packetloss, ma as packetloss_avg, sum(inbandwidth) as i sum(bibandwidth) as bibandwidth, count(* cast((substring(msg,'Latency: (\\d+\\.? substring(msg,'jitter: (\\d+\\.?\\d*), ' g(msg,'packet loss: (\\d+\\.?\\d*)%, ') ing(msg,'inbandwidth: (\\d+\\.?\\d*)) as inbandwidth, (format2bytes(substrint P,e,E])bps'))) as outbandwidth, (format2 [k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as bik ogid)=22925 and msg is not null) t group der by timestamp desc/*SkipEND*/)### t w null group by devid order by bibandwidt</pre>	<pre>tter, max(jitter) as ax(packetloss) as nbandwidth, sum as count from e\\d*), ')::float) as)::float as decimal as decimal(18,2)) as ag(msg,'outbandwidth ebytes(substring bandwidth from \$log by timestamp, where \$filter-</pre>

Description

<pre>select \$flex_timescale(timestamp) as hodex, t1.interface, sum(bibandwidth) / sum(count) as bibandwidth from ###(select \$flex_timestamp as timestamp, devid, interface, min(latency) as latency, max (latency) as latency_max, avg(latency) as latency_avg, min(jitter) as jitter, max(jitter) jitter_max, avg(jitter) as jitter_avg, min(packetloss) as packetloss, max(packetloss) as packetloss_max, avg(packetloss) as packetloss_avg, sum(inbandwidth) as inbandwidth, sum (outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count from (select itime, devid, interface, cast((substring(msg,'Latency: (\\d+\\.?\\d*), ')::float)</pre>	sdwan-Device-Interface-Bibandwidth- Line	SD-WAN Device-Interface Bibandwidth Line	event
<pre>decimal(18,2)) as latency, cast(substring(msg,'jitter: (\\d+\\.?\\d*), ')::float as decima (18,2)) as jitter, cast(substring(msg,'packet loss: (\\d+\\.?\\d*)%, ') as decimal(18,2)) packetloss, (format2bytes(substring(msg,'inbandwidth: (\\d+\\.?\\d* [k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as inbandwidth, (format2bytes(substring(msg,'outbandwidt (\\d+\\.?\\d*[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as outbandwidth, (format2bytes(substring (msg,'bibandwidth: (\\d+\\.?\\d*[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as bibandwidth from \$log where \$filter and logid_to_int(logid)=22925 and msg is not null) t group by timestamp, devid, interface /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t1 inner join (select interface, count(*) as num_intf from ###(select \$flex_timestamp as timestamp, devid, interface, min(latency) as latency, max(latency) as latency_max, avg(latency) as latency_ avg, min(jitter) as jitter, max(jitter) as jitter_max, avg(jitter) as jitter_avg, min (packetloss) as packetloss, max(packetloss) as packetloss_max, avg(packetloss) as packetloss_avg, sum(inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum (bibandwidth) as bibandwidth, count(*) as count from (select itime, devid, interface, cast ((substring(msg,'Latency: (\\d+\\.?\\d*), ')::float) as decimal(18,2)) as latency, cast</pre>	<pre>\$flex_timescale(timestamp) as h t1.interface, sum(bibandwidth)/ sum(count) as from ###(select \$flex_timestamp as t (latency) as latency_max, avg(late jitter_max, avg(jitter) as jitter packetloss_max, avg(packetloss) a (outbandwidth) as outbandwidth, s (select itime, devid, interface, decimal(18,2)) as latency, cast(s (18,2)) as jitter, cast(substring packetloss, (format2bytes(substring packetloss, (latency), cast(s (lk,K,M,m,g,G,t,T,p,P,e,E])bps'))) (\\d+\\.?\\d*[k,K,M,m,g,G,t,T,p,F,f]) where \$filter and logid_to_int(latency) where \$filter and logid_to_int(latency) as latency avg, min(jitter) as jitter, max(filter) packetloss_avg, sum(inbandwidth) (bibandwidth) as bibandwidth, conducted (select loss_avg, sum(inbandwidth), conducted) (select loss_avg, sum(inbandwidth, conducted) (select loss) as packetloss, max(select)) (select loss) as select) (select loss) (sele</pre>	s bibandwidth timestamp, devid, interface, min(latency) tency) as latency_avg, min(jitter) as jit r_avg, min(packetloss) as packetloss, may as packetloss_avg, sum(inbandwidth) as in sum(bibandwidth) as bibandwidth, count(*) cast((substring(msg,'Latency: (\\d+\\.?\ substring(msg,'jitter: (\\d+\\.?\\d*), ') g(msg,'packet loss: (\\d+\\.?\\d*)%, ') a ing(msg,'inbandwidth: (\\d+\\.?\\d*) as inbandwidth, (format2bytes(substring P,e,E])bps'))) as outbandwidth, (format2k [k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as biba bgid)=22925 and msg is not null) t group der by timestamp desc/*SkipEND*/)### t1 f from ###(select \$flex_timestamp as timest cy, max(latency) as latency_max, avg(latency) packetloss) as packetloss_max, avg(packet as inbandwidth, sum(outbandwidth) as out unt(*) as count from (select itime, device	tter, max(jitter) as (packetloss) as hbandwidth, sum) as count from (\d*), ')::float) as)::float as decimal as decimal(18,2)) as g(msg,'outbandwidth: bytes(substring andwidth from \$log by timestamp, inner join (select tamp, devid, ency) as latency_ tter_avg, min tloss) as tbandwidth, sum d, interface, cast

Log Category

(substring(msg,'jitter: (\\d+\\.?\\d*), ')::float as decimal(18,2)) as jitter, cast (substring(msg,'packet loss: (\\d+\\.?\\d*)%, ') as decimal(18,2)) as packetloss, (format2bytes(substring(msg,'inbandwidth: (\\d+\\.?\\d*[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as inbandwidth, (format2bytes(substring(msg,'outbandwidth: (\\d+\\.?\\d* [k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as outbandwidth, (format2bytes(substring(msg,'bibandwidth: (\\d+\\.?\\d*[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as bibandwidth from \$log where \$filter and logid_to_int(logid)=22925 and msg is not null) t group by timestamp, devid, interface /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filter-drilldown and interface is not null group by interface order by num_intf desc limit 10)t2 on t1.interface=t2.interface group by hodex, t1.interface order by hodex

Dataset Name	Description	Log Category
sdwan-Device-Interface-Bibandwidth- Pie	SD-WAN Device-Interface Statistic	event

select

```
devid,
interface,
sum(bibandwidth) / sum(count) as bibandwidth
from
```

###(select \$flex_timestamp as timestamp, devid, interface, min(latency) as latency, max (latency) as latency_max, avg(latency) as latency_avg, min(jitter) as jitter, max(jitter) as jitter_max, avg(jitter) as jitter_avg, min(packetloss) as packetloss, max(packetloss) as packetloss_max, avg(packetloss) as packetloss_avg, sum(inbandwidth) as inbandwidth, sum (outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count from (select itime, devid, interface, cast((substring(msg,'Latency: (\\d+\\.?\\d*), ')::float) as decimal(18,2)) as latency, cast(substring(msg,'jitter: (\\d+\\.?\\d*), ')::float as decimal (18,2)) as jitter, cast(substring(msg,'packet loss: (\\d+\\.?\\d*)%, ') as decimal(18,2)) as packetloss, (format2bytes(substring(msg,'inbandwidth: (\\d+\\.?\\d* [k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as inbandwidth, (format2bytes(substring(msg,'outbandwidth: (\\d+\\.?\\d*[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as outbandwidth, (format2bytes(substring

(msg,'bibandwidth: (\\d+\\.?\\d*[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as bibandwidth from \$log where \$filter and logid_to_int(logid)=22925 and msg is not null) t group by timestamp, devid, interface /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filterdrilldown and bibandwidth is not null group by devid, interface order by bibandwidth desc

Dataset Name	Description	Log Category
sdwan-Device-Interface-Inbandwidth- Line	SD-WAN Device-Interface Inbandwidth Line	event
select		

\$flex_timescale(timestamp) as hodex, t1.interface, sum(inbandwidth)/ sum(count) as inbandwidth from

###(select \$flex_timestamp as timestamp, devid, interface, min(latency) as latency, max (latency) as latency_max, avg(latency) as latency_avg, min(jitter) as jitter, max(jitter) as jitter_max, avg(jitter) as jitter_avg, min(packetloss) as packetloss, max(packetloss) as packetloss_max, avg(packetloss) as packetloss_avg, sum(inbandwidth) as inbandwidth, sum (outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count from (select itime, devid, interface, cast((substring(msg,'Latency: (\\d+\\.?\\d*), ')::float) as decimal(18,2)) as latency, cast(substring(msg,'jitter: (\\d+\\.?\\d*), ')::float as decimal (18,2)) as jitter, cast(substring(msg,'packet loss: (\\d+\\.?\\d*)%, ') as decimal(18,2)) as packetloss, (format2bytes(substring(msg,'inbandwidth: (\\d+\\.?\\d*)

[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as inbandwidth, (format2bytes(substring(msg,'outbandwidth: (\\d+\\.?\\d*[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as outbandwidth, (format2bytes(substring (msg,'bibandwidth: (\\d+\\.?\\d*[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as bibandwidth from \$log where \$filter and logid to int(logid)=22925 and msg is not null) t group by timestamp, devid, interface /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t1 inner join (select interface, count(*) as num intf from ###(select \$flex timestamp as timestamp, devid, interface, min(latency) as latency, max(latency) as latency max, avg(latency) as latency avq, min(jitter) as jitter, max(jitter) as jitter max, avq(jitter) as jitter avq, min (packetloss) as packetloss, max(packetloss) as packetloss max, avg(packetloss) as packetloss avg, sum(inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum (bibandwidth) as bibandwidth, count(*) as count from (select itime, devid, interface, cast ((substring(msg, 'Latency: (\\d+\\.?\\d*), ')::float) as decimal(18,2)) as latency, cast (substring(msg,'jitter: (\\d+\\.?\\d*), ')::float as decimal(18,2)) as jitter, cast (substring(msg, 'packet loss: $(\d+\.?\d*)$, ') as decimal(18,2)) as packetloss, (format2bytes(substring(msg,'inbandwidth: (\\d+\\.?\\d*[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as inbandwidth, (format2bytes(substring(msg,'outbandwidth: $(\d+\.?\d+$ [k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as outbandwidth, (format2bytes(substring(msg,'bibandwidth: (\\d+\\.?\\d*[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as bibandwidth from \$log where \$filter and logid to int(logid)=22925 and msg is not null) t group by timestamp, devid, interface /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filter-drilldown and interface is not null group by interface order by num intf desc limit 10)t2 on t1.interface=t2.interface group by hodex, t1.interface order by hodex

```
Dataset NameDescriptionLog Categorysdwan-Device-Interface-Inbandwidth-<br/>PieSD-WAN Device-Interface Downstream Statisticevent
```

select
 devid,
 interface,
 sum(inbandwidth)/ sum(count) as inbandwidth
from

###(select \$flex_timestamp as timestamp, devid, interface, min(latency) as latency, max (latency) as latency_max, avg(latency) as latency_avg, min(jitter) as jitter, max(jitter) as jitter_max, avg(jitter) as jitter_avg, min(packetloss) as packetloss, max(packetloss) as packetloss_max, avg(packetloss) as packetloss_avg, sum(inbandwidth) as inbandwidth, sum (outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count from (select itime, devid, interface, cast((substring(msg,'Latency: (\\d+\\.?\\d*), ')::float) as decimal(18,2)) as latency, cast(substring(msg,'jitter: (\\d+\\.?\\d*), ')::float as decimal (18,2)) as jitter, cast(substring(msg,'packet loss: (\\d+\\.?\\d*)%, ') as decimal(18,2)) as packetloss, (format2bytes(substring(msg,'inbandwidth: (\\d+\\.?\\d* [k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as inbandwidth, (format2bytes(substring(msg,'outbandwidth: (\\d+\\.?\\d*[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as outbandwidth, (format2bytes(substring (msg,'bibandwidth: (\\d+\\.?\\d*[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as bibandwidth from \$log where \$filter and logid_to_int(logid)=22925 and msg is not null) t group by timestamp, devid, interface /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filterdrilldown and bibandwidth is not null group by devid, interface order by inbandwidth desc

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

select

\$flex_timescale(timestamp) as hodex,

```
t1.interface,
  sum(outbandwidth) / sum(count) as outbandwidth
from
  ###(select $flex timestamp as timestamp, devid, interface, min(latency) as latency, max
(latency) as latency max, avg(latency) as latency avg, min(jitter) as jitter, max(jitter) as
jitter max, avg(jitter) as jitter avg, min(packetloss) as packetloss, max(packetloss) as
packetloss max, avg(packetloss) as packetloss avg, sum(inbandwidth) as inbandwidth, sum
(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count from
(select itime, devid, interface, cast((substring(msg,'Latency: (\\d+\\.?\\d*), ')::float) as
decimal(18,2)) as latency, cast(substring(msg,'jitter: (\\d+\\.?\\d*), ')::float as decimal
(18,2)) as jitter, cast(substring(msg,'packet loss: (\d+\.?\d*)%, ') as decimal(18,2)) as
packetloss, (format2bytes(substring(msg,'inbandwidth: (\\d+\\.?\\d*
[k,K,M,m,q,G,t,T,p,P,e,E])bps'))) as inbandwidth, (format2bytes(substring(msq,'outbandwidth:
(\\d+\\.?\\d*[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as outbandwidth, (format2bytes(substring
(msg, 'bibandwidth: (\\d+\\.?\\d*[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as bibandwidth from $log
where $filter and logid to int(logid)=22925 and msg is not null) t group by timestamp,
devid, interface /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t1 inner join (select
interface, count(*) as num intf from ###(select $flex timestamp as timestamp, devid,
interface, min(latency) as latency, max(latency) as latency max, avg(latency) as latency
avg, min(jitter) as jitter, max(jitter) as jitter max, avg(jitter) as jitter avg, min
(packetloss) as packetloss, max(packetloss) as packetloss_max, avg(packetloss) as
packetloss avg, sum(inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum
(bibandwidth) as bibandwidth, count(*) as count from (select itime, devid, interface, cast
((substring(msg,'Latency: (\\d+\\.?\\d*), ')::float) as decimal(18,2)) as latency, cast
(substring(msg,'jitter: (\d+\.?\d+), ')::float as decimal(18,2)) as jitter, cast
(substring(msg, 'packet loss: (\ (\ )\ )\ , ') as decimal(18,2)) as packetloss,
(format2bytes(substring(msg,'inbandwidth: (\\d+\\.?\\d*[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as
inbandwidth, (format2bytes(substring(msg,'outbandwidth: (\\d+\\.?\\d*
[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as outbandwidth, (format2bytes(substring(msg,'bibandwidth:
(\\d+\\.?\\d*[k,K,M,m,q,G,t,T,p,P,e,E])bps'))) as bibandwidth from $log where $filter and
logid to int(logid)=22925 and msg is not null) t group by timestamp, devid, interface
/*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where $filter-drilldown and interface
is not null group by interface order by num intf desc limit 10)t2 on
t1.interface=t2.interface group by hodex, t1.interface order by hodex
```

Dataset Name	Description	Log Category
sdwan-Device-Interface- Outbandwidth-Pie	SD-WAN Device-Interface Upstream Statistic	event
<pre>select devid, interface, sum(outbandwidth) / sum(count)</pre>	as outbandwidth	
<pre>(latency) as latency_max, avg(la jitter_max, avg(jitter) as jitte packetloss_max, avg(packetloss)</pre>	<pre>timestamp, devid, interface, min(latency) as l tency) as latency_avg, min(jitter) as jitter, r_avg, min(packetloss) as packetloss, max(pack as packetloss_avg, sum(inbandwidth) as inbandw sum(bibandwidth) as bibandwidth, count(*) as c</pre>	<pre>max(jitter) as setloss) as ridth, sum</pre>
<pre>(select itime, devid, interface, decimal(18,2)) as latency, cast((18,2)) as jitter, cast(substring)</pre>	<pre>cast((substring(msg,'Latency: (\\d+\\.?\\d*), substring(msg,'jitter: (\\d+\\.?\\d*), ')::flc g(msg,'packet loss: (\\d+\\.?\\d*)%, ') as dec ing(msg,'inbandwidth: (\\d+\\.?\\d*</pre>	')::float) as at as decimal

[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as inbandwidth, (format2bytes(substring(msg,'outbandwidth: (\\d+\\.?\\d*[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as outbandwidth, (format2bytes(substring (msg,'bibandwidth: (\\d+\\.?\\d*[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as bibandwidth from \$log where \$filter and logid_to_int(logid)=22925 and msg is not null) t group by timestamp, devid, interface /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filterdrilldown and bibandwidth is not null group by devid, interface order by outbandwidth desc

Dataset Name	Description	Log Category
sdwan-Device-Interface-Latency-Line	SD-WAN Device-Interface Latency Line	event
<pre>select \$flex_timescale(timestamp) as t1.interface, min(latency) as latency</pre>	hodex,	
from (
select timestamp,		
devid, interface,		
<pre>sum(latency) / sum(count) a</pre>	as latency	
— — — — — — — — — — — — — — — — — — — —	o as timestamp, devid, interface, min	

max(latency) as latency max, avg(latency) as latency avg, min(jitter) as jitter, max(jitter) as jitter max, avg(jitter) as jitter avg, min(packetloss) as packetloss, max(packetloss) as packetloss max, avg(packetloss) as packetloss avg, sum(inbandwidth) as inbandwidth, sum (outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count from (select itime, devid, interface, cast((substring(msg,'Latency: (\\d+\\.?\\d*), ')::float) as decimal(18,2)) as latency, cast(substring(msg,'jitter: (\\d+\\.?\\d*), ')::float as decimal (18,2)) as jitter, cast(substring(msg,'packet loss: $(\backslash d+ \backslash .? \backslash d*)$ %, ') as decimal(18,2)) as packetloss, (format2bytes(substring(msg,'inbandwidth: (\\d+\\.?\\d* [k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as inbandwidth, (format2bytes(substring(msg,'outbandwidth: (\\d+\\.?\\d*[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as outbandwidth, (format2bytes(substring (msq, 'bibandwidth: (\\d+\\.?\\d*[k,K,M,m,q,G,t,T,p,P,e,E])bps'))) as bibandwidth from \$log where \$filter and logid to int(logid)=22925 and msg is not null) t group by timestamp, devid, interface /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t group by timestamp, devid, interface) t1 inner join (select interface, count(*) as num intf from ###(select \$flex timestamp as timestamp, devid, interface, min(latency) as latency, max(latency) as latency max, avg(latency) as latency avg, min(jitter) as jitter, max(jitter) as jitter max, avg(jitter) as jitter avg, min(packetloss) as packetloss, max(packetloss) as packetloss max, avg(packetloss) as packetloss avg, sum(inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count from (select itime, devid, interface, cast((substring(msg,'Latency: ($\d+\, ?\d+$), ')::float) as decimal(18,2)) as latency, cast(substring(msg,'jitter: $(\d+\...\d+\), ')$::float as decimal(18,2)) as jitter, cast(substring(msg,'packet loss: $(\d+\.?\d*)$, ') as decimal(18,2)) as packetloss, (format2bytes(substring(msg,'inbandwidth: (\\d+\\.?\\d*[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as inbandwidth, (format2bytes(substring(msg,'outbandwidth: (\\d+\\.?\\d* [k,K,M,m,q,G,t,T,p,P,e,E])bps'))) as outbandwidth, (format2bytes(substring(msq,'bibandwidth: (\\d+\\.?\\d*[k,K,M,m,q,G,t,T,p,P,e,E])bps'))) as bibandwidth from \$log where \$filter and logid to int(logid)=22925 and msg is not null) t group by timestamp, devid, interface /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filter-drilldown and interface is not null group by interface order by num intf desc limit 10)t2 on t1.interface=t2.interface group by hodex, t1.interface order by hodex

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

```
select
  $flex_timescale(timestamp) as hodex,
  t1.interface,
  min(jitter) as jitter
from
  (
    select
    timestamp,
    devid,
    interface,
    sum(jitter)/ sum(count) as jitter
  from
```

###(select \$flex_timestamp as timestamp, devid, interface, min(latency) as latency, max(latency) as latency max, avg(latency) as latency avg, min(jitter) as jitter, max(jitter) as jitter_max, avg(jitter) as jitter avg, min(packetloss) as packetloss, max(packetloss) as packetloss max, avg(packetloss) as packetloss avg, sum(inbandwidth) as inbandwidth, sum (outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count from (select itime, devid, interface, cast((substring(msg,'Latency: (\\d+\\.?\\d*), ')::float) as decimal(18,2)) as latency, cast(substring(msg,'jitter: (\\d+\\.?\\d*), ')::float as decimal (18,2)) as jitter, cast(substring(msg,'packet loss: $(\d+\.?\d*)$, ') as decimal(18,2)) as packetloss, (format2bytes(substring(msg,'inbandwidth: (\\d+\\.?\\d* [k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as inbandwidth, (format2bytes(substring(msg,'outbandwidth: (\\d+\\.?\\d*[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as outbandwidth, (format2bytes(substring (msg, 'bibandwidth: (\\d+\\.?\\d*[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as bibandwidth from \$log where \$filter and logid_to_int(logid)=22925 and msg is not null) t group by timestamp, devid, interface /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t group by timestamp, devid, interface) t1 inner join (select interface, count(*) as num intf from ###(select \$flex timestamp as timestamp, devid, interface, min(latency) as latency, max(latency) as latency max, avg(latency) as latency avg, min(jitter) as jitter, max(jitter) as jitter max, avg(jitter) as jitter avg, min(packetloss) as packetloss, max(packetloss) as packetloss max, avg(packetloss) as packetloss avg, sum(inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count from (select itime, devid, interface, cast((substring(msg,'Latency: (\\d+\\.?\\d*), ')::float) as decimal(18,2)) as latency, cast(substring(msg,'jitter: (\\d+\\.?\\d*), ')::float as decimal(18,2)) as jitter, cast(substring(msg,'packet loss: $(\d+\.?\d*)$, ') as decimal(18,2)) as packetloss, (format2bytes(substring(msg,'inbandwidth: (\\d+\\.?\\d*[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as inbandwidth, (format2bytes(substring(msg,'outbandwidth: (\\d+\\.?\\d* [k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as outbandwidth, (format2bytes(substring(msg,'bibandwidth: (\\d+\\.?\\d*[k,K,M,m,q,G,t,T,p,P,e,E])bps'))) as bibandwidth from \$log where \$filter and logid to int(logid)=22925 and msg is not null) t group by timestamp, devid, interface /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filter-drilldown and interface is not null group by interface order by num intf desc limit 10)t2 on t1.interface=t2.interface group by hodex, t1.interface order by hodex

Dataset Name	Description	Log Category
sdwan-Device-Interface-Packetloss- Line	SD-WAN Device-Interface Packetloss Line	event
<pre>select \$flex_timescale(timestamp) as t1.interface, min(packetloss) as packetloss from (select</pre>	hodex,	

```
timestamp,
devid,
interface,
sum(packetloss)/ sum(count) as packetloss
from
```

###(select \$flex timestamp as timestamp, devid, interface, min(latency) as latency, max(latency) as latency max, avg(latency) as latency avg, min(jitter) as jitter, max(jitter) as jitter max, avg(jitter) as jitter avg, min(packetloss) as packetloss, max(packetloss) as packetloss max, avg(packetloss) as packetloss avg, sum(inbandwidth) as inbandwidth, sum (outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count from (select itime, devid, interface, cast((substring(msg,'Latency: (\\d+\\.?\\d*), ')::float) as decimal(18,2)) as latency, cast(substring(msg,'jitter: (\\d+\\.?\\d*), ')::float as decimal (18,2)) as jitter, cast(substring(msg,'packet loss: $(\d+\.?\d*)$, ') as decimal(18,2)) as packetloss, (format2bytes(substring(msg,'inbandwidth: (\\d+\\.?\\d* [k,K,M,m,q,G,t,T,p,P,e,E])bps'))) as inbandwidth, (format2bytes(substring(msq,'outbandwidth: (\\d+\\.?\\d*[k,K,M,m,q,G,t,T,p,P,e,E])bps'))) as outbandwidth, (format2bytes(substring (msg,'bibandwidth: (\\d+\\.?\\d*[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as bibandwidth from \$log where \$filter and logid to int(logid)=22925 and msg is not null) t group by timestamp, devid, interface /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t group by timestamp, devid, interface) t1 inner join (select interface, count(*) as num intf from ###(select \$flex_timestamp as timestamp, devid, interface, min(latency) as latency, max(latency) as latency_max, avg(latency) as latency_avg, min(jitter) as jitter, max(jitter) as jitter_max, avg(jitter) as jitter avg, min(packetloss) as packetloss, max(packetloss) as packetloss max, avg(packetloss) as packetloss avg, sum(inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count from (select itime, devid, interface, cast((substring(msg,'Latency: (\\d+\\.?\\d*), ')::float) as decimal(18,2)) as latency, cast(substring(msg,'jitter: (\\d+\\.?\\d*), ')::float as decimal(18,2)) as jitter, cast(substring(msg,'packet loss: ($\d+\.?\d*$)%, ') as decimal(18,2)) as packetloss, (format2bytes(substring(msg,'inbandwidth: (\\d+\\.?\\d*[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as inbandwidth, (format2bytes(substring(msg,'outbandwidth: $(\d+\.?\d+$ [k,K,M,m,q,G,t,T,p,P,e,E])bps'))) as outbandwidth, (format2bytes(substring(msq,'bibandwidth: (\\d+\\.?\\d*[k,K,M,m,q,G,t,T,p,P,e,E])bps'))) as bibandwidth from \$log where \$filter and logid to int(logid)=22925 and msg is not null) t group by timestamp, devid, interface /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filter-drilldown and interface is not null group by interface order by num intf desc limit 10)t2 on t1.interface=t2.interface group by hodex, t1.interface order by hodex

Dataset Name	Description	Log Category
sdwan-fw-Device-Stat_by_ Bibandwidth	SD-WAN Device Statistic by Bibandwidth	event

select

devid,

sum(bibandwidth) / sum(count) as bibandwidth

from

###(select \$flex_timestamp as timestamp, devid, interface, min(latency) as latency, max (latency) as latency_max, avg(latency) as latency_avg, min(jitter) as jitter, max(jitter) as jitter_max, avg(jitter) as jitter_avg, min(packetloss) as packetloss, max(packetloss) as packetloss_max, avg(packetloss) as packetloss_avg, sum(inbandwidth) as inbandwidth, sum (outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count from (select itime, devid, interface, cast((substring(msg,'Latency: (\\d+\\.?\\d*), ')::float) as decimal(18,2)) as latency, cast(substring(msg,'jitter: (\\d+\\.?\\d*), ')::float as decimal (18,2)) as jitter, cast(substring(msg,'nacket loss: (\\d+\\.?\\d*), ') as decimal(18,2)) as packetloss, (format2bytes(substring(msg,'inbandwidth: (\\d+\\.?\\d* [k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as inbandwidth, (format2bytes(substring(msg,'outbandwidth: (\\d+\\.?\\d*[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as outbandwidth, (format2bytes(substring (msg,'bibandwidth: (\\d+\\.?\\d*[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as bibandwidth from \$log where \$filter and logid_to_int(logid)=22925 and msg is not null) t group by timestamp, devid, interface /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filterdrilldown and bibandwidth is not null group by devid order by bibandwidth desc

Dataset Name	Description	Log Category
sdwan-Device-Latency-Line	SD-WAN Device Latency Line	event
<pre>select \$flex_timescale(timestamp) as 1 devid, min(latency) as latency from (select timestamp, devid, interface, sum(latency) / sum(count) as from ###(select \$flex_timestamp max(latency) as latency_max, avg as jitter_max, avg(jitter) as jis packetloss_max, avg(packetloss) (outbandwidth) as outbandwidth, (select itime, devid, interface, decimal(18,2)) as latency, cast(substring packetloss, (format2bytes(substring packetloss, (format2bytes(substring packetloss, (format2bytes(substring packetloss, (lotter))) (\\d+\\.?\\d*[k,K,M,m,g,G,t,T,p,f]) may (bibandwidth: (\\d+\\.?\\d* where \$filter and logid_to_int(log devid, interface /*SkipSTART*/org </pre>	nodex,	<pre>in(latency) as latency, eer) as jitter, max(jitter) etloss, max(packetloss) as a) as inbandwidth, sum count(*) as count from .d+\\.?\\d*), ')::float) as .d*), ')::float as decimal %, ') as decimal(18,2)) as .* ubstring(msg,'outbandwidth: format2bytes(substring as bibandwidth from \$log : group by timestamp, ## t group by timestamp,</pre>

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

max(latency) as latency_max, avg(latency) as latency_avg, min(jitter) as jitter, max(jitter) as jitter_max, avg(jitter) as jitter_avg, min(packetloss) as packetloss, max(packetloss) as packetloss_max, avg(packetloss) as packetloss_avg, sum(inbandwidth) as inbandwidth, sum (outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count from (select itime, devid, interface, cast((substring(msg,'Latency: (\\d+\\.?\\d*), ')::float) as decimal(18,2)) as latency, cast(substring(msg,'jitter: (\\d+\\.?\\d*)%, ') as decimal(18,2)) as packetloss, (format2bytes(substring(msg,'packet loss: (\\d+\\.?\\d*)%, ') as decimal(18,2)) as packetloss, (format2bytes(substring(msg,'inbandwidth: (\\d+\\.?\\d* [k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as inbandwidth, (format2bytes(substring(msg,'outbandwidth: (\\d+\\.?\\d*[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as outbandwidth, (format2bytes(substring (msg,'bibandwidth: (\\d+\\.?\\d*[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as bibandwidth from \$log where \$filter and logid_to_int(logid)=22925 and msg is not null) t group by timestamp, devid, interface /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t group by timestamp, devid, interface) t1 where \$filter-drilldown and jitter is not null group by hodex, devid order by hodex

Dataset Name	Description	Log Category
sdwan-Device-Packetloss-Line	SD-WAN Device Packet Loss Line	event
select		
<pre>\$flex_timescale(timestamp) as devid,</pre>	hodex,	
min(packetloss) as packetloss		
from		
(
select		
timestamp,		
devid,		
interface,		
sum(packetloss)/ sum(count) as packetloss	
from		
###(select \$flex_timestamp	as timestamp, devid, interface, min	(latency) as latency,
<pre>max(latency) as latency_max, avg</pre>	(latency) as latency_avg, min(jitter	<pre>as jitter, max(jitter)</pre>
	tter_avg, min(packetloss) as packetl	_
	as packetloss_avg, sum(inbandwidth)	•
	sum(bibandwidth) as bibandwidth, cou	
	<pre>cast((substring(msg,'Latency: (\\d+</pre>	
	<pre>substring(msg,'jitter: (\\d+\\.?\\d*</pre>	
-	g(msg,'packet loss: (\\d+\\.?\\d*)%,	') as decimal(18,2)) as
	<pre>ing(msg,'inbandwidth: (\\d+\\.?\\d*</pre>	
) as inbandwidth, (format2bytes(subs	5. 5.
	P,e,E])bps'))) as outbandwidth, (for	
-	[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as	
	ogid)=22925 and msg is not null) t g der by timestamp desc/*SkipEND*/)###	

devid, interface /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t group by timestamp, devid, interface) t1 where \$filter-drilldown and packetloss is not null group by hodex, devid order by hodex

```
Dataset NameDescriptionLog Categorysdwan-Device-Interface-Summary-by-<br/>BibandwidthSD-WAN Device Interface Summary by Bibandwidthevent
```

```
select
devid,
```

```
interface,
  sum(bibandwidth) / sum(count) as bibandwidth,
 min(latency) as latency min,
  cast(
   avg(latency avg) as decimal(18, 2)
 ) as latency avg,
 max(latency max) as latency max,
 min(jitter) as jitter min,
 cast(
   avg(jitter avg) as decimal(18, 2)
 ) as jitter avg,
 max(jitter max) as jitter max,
 min(packetloss) as packetloss min,
 cast(
   avg(packetloss avg) as decimal(18, 2)
 ) as packetloss avg,
 max(packetloss max) as packetloss max
from
```

###(select \$flex_timestamp as timestamp, devid, interface, min(latency) as latency, max (latency) as latency_max, avg(latency) as latency_avg, min(jitter) as jitter, max(jitter) as jitter_max, avg(jitter) as jitter_avg, min(packetloss) as packetloss, max(packetloss) as packetloss_max, avg(packetloss) as packetloss_avg, sum(inbandwidth) as inbandwidth, sum (outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count from (select itime, devid, interface, cast((substring(msg,'Latency: (\\d+\\.?\\d*), ')::float) as decimal(18,2)) as latency, cast(substring(msg,'jitter: (\\d+\\.?\\d*)%, ') as decimal(18,2)) as packetloss, (format2bytes(substring(msg,'packet loss: (\\d+\\.?\\d*) [k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as inbandwidth; (l\d+\\.?\\d* [k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as outbandwidth, (format2bytes(substring (msg,'bibandwidth: (\\d+\\.?\\d*[k,K,M,m,g,G,t,T,p,P,e,E])bps'))) as bibandwidth from \$log where \$filter and logid_to_int(logid)=22925 and msg is not null) t group by timestamp, devid, interface is not null group by devid, interface order by bibandwidth desc

Dataset Name	Description	Log Category
Top-Web-Sites-by-Bandwidth	Top web sites by bandwidth usage	webfilter

select

domain, sum(bandwidth) as bandwidth from

###(select coalesce(nullifna(hostname), ipstr(`dstip`)) as domain, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from \$log-traffic where \$filter and (logflag&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')))) group by domain having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc)### t group by domain order by bandwidth desc

Dataset Name	Description	Log Category
Top-App-Category-by-Session	Application risk application usage by category	traffic
select		

```
select
appcat,
sum(sessions) as total num
```

from

###base(/*tag:rpt_base_t_top_app*/select devid, vd, csf, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, appid, app, appcat, apprisk, sum (coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, count(*) as sessions from \$log where \$filter and (logflag&1>0) and nullifna(app) is not null group by devid, vd, csf, user_ src, appid, app, appcat, apprisk order by sessions desc)base### t where \$filter-drilldown and appcat is not null group by appcat order by total_num desc

Dataset Name	Description	Log Category
Top-Region-Name-by-Traffic	Traffic top destination countries by browsing time	traffic

select

dstcountry, sum(bandwidth) as bandwidth

from

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

Dataset Name	Description	Log Category
Top-App-By-Bandwidth-Chart	Top applications by bandwidth usage	traffic
<pre>Top-App-By-Bandwidth-Chart select app_group_name(app) as app_ sum(coalesce(sentbyte, 0) + cc) 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 group by</pre>	group, palesce(rcvdbyte, 0)	traffic
app_group		
<pre>having sum(coalesce(sentbyte, 0)+ cc)& gt; 0</pre>	palesce(rcvdbyte, 0)	

order by bandwidth desc

Dataset Name	Description	Log Category
Top-Protocols-By-Traffic	Top applications by bandwidth usage	traffic

select

```
service,
```

sum(bandwidth) as bandwidth

from

###base(/*tag:rpt_base_t_bndwdth_sess*/select \$flex_timestamp as timestamp, devid, vd, csf, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service, count(*) as sessions, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(sentbyte, 0)) as traffic_out, sum(coalesce(rcvdbyte, 0)) as traffic_ in from \$log where \$filter and (logflag&1>0) group by timestamp, devid, vd, csf, user_src, service /*SkipSTART*/order by timestamp desc/*SkipEND*/)base### base_query where \$filterdrilldown group by service order by bandwidth desc

Dataset Name	Description	Log Category
Top-Web-Sites-by-Sessions	Top web sites by session count	webfilter

select

```
domain,
sum(sessions) as sessions
from
```

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

Dataset Name	Description	Log Category
Top-Attacks-by-Count	Threat attacks by severity	attack
<pre>select attack, sum(attack_count) as totalnum from ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, attack, (case when severity in ('critical', 'high') then 1 else 0 end) as high_severity, count(*) as attack_ count from \$log where \$filter and nullifna(attack) is not null group by user_src, attack, high_severity order by attack_count desc)### t where \$filter-drilldown and attack is not null group by attack order by totalnum desc</pre>		
Dataset Name Description Log C		
Top-Spams-by-Count	User drilldown top spam sources	emailfilter
<pre>select user_src, sum(totalnum) as totaln from ###(select \$flex_timest</pre>	um amp as timestamp, coalesce(nullifna(`user`)	, ipstr(`srcip`)) as

```
user_src, `from` as mf_sender, `to` as mf_receiver, action, eventtype, count(*) as totalnum from $log where $filter group by timestamp, user_src, mf_sender, mf_receiver, action,
```

eventtype /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filter-drilldown and mf sender is not null group by user src order by totalnum desc

Dataset Name	Description	Log Category
utm-Top-Virus-Count	UTM top virus	virus
<pre>select virus, max(virusid_s) as virusid (case when virus like 'R else 'Virus' end</pre>		virus like 'Adware%' then 'Adware'
) as malware_type, sum(totalnum) as totalnum from		virusid_s, count(*) as totalnum

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

Dataset Name	Description	Log Category
security-Antivirus-Inspections	Antivirus Inspections	virus

select

```
action,
sum(totalnum) as totalnum
```

```
from
```

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

Dataset Name	Description	Log Category
Top-DLP-by-Count	Email DLP Activity Summary	dlp
<pre>select profile, count(*) as total_num from</pre>		

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

Dataset Name	Description	Log Category
wifi-Top-AP-By-Client	Top access point by client	traffic
<pre>select ap_srcintf as srcintf,</pre>		

```
\operatorname{count}(\operatorname{distinct}\operatorname{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-AP-By-Bandwidth	Top access point by bandwidth usage	traffic
<pre>select coalesce(ap, srcintf) as sum(</pre>		
<pre>coalesce(sentbyte, 0)+) as bandwidth</pre>	coalesce(rcvdbyte, 0)	
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)+)& qt; 0</pre>	coalesce(rcvdbyte, 0)	
order by		
bandwidth desc		

Dataset Name	Description	Log Category
Dataset Name	Description	

wifi-Top-SSID-By-Bandwidth	Top SSIDs by bandwidth usage	traffic
elect		
srcssid,		
sum (
<pre>coalesce(sentbyte, 0) + co</pre>	palesce(rcvdbyte, 0)	
) as bandwidth	-	
rom		
\$log		
here		
\$filter		
and (
logflag&1>0		
)		
and srcssid is not null		
roup by		
srcssid		

```
having
  sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  )& gt; 0
order by
  bandwidth desc
```

Dataset Name

	Decemption	
360-degree-security-Application- Visiblity-and-Control-Summary	Application Visibolity and Control Summary	app-ctrl

Description

select

```
appcat,
count(distinct app) as total num
```

from

###(select appcat, app from \$log where \$filter and app is not null and appcat is not null group by appcat, app)### t group by appcat order by total_num desc

Dataset Name	Description	Log Category
360-degree-security-Threats-	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 \$logwebfilter 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
<pre>subtype, srcip, dstip, severity, severity='critical' then 'Critic (`user`), ipstr(`srcip`)) is not as data_loss from \$log where \$fi</pre>	om` as sender, `to` as receiver, pro filename, direction, filesize, (cas al Data Exfiltration' else (case who null then 'User Associated Data Los lter /*SkipSTART*/order by itime des is not null group by data_loss ordes	se when en coalesce(nullifna ss' else NULL end) end) sc/*SkipEND*/)### t where

Log Category

Dataset Name	Description	Log Category
360-degree-security-Endpoint- Protection-Summary	Endpoint Protection	fct-traffic
<pre>then 'Risk Application Blocked' case when coalesce(nullifna(`user`), ipstr(`srcip`)</pre>	ivirus' then 'Malware Deteced an when 'webfilter' then (Web Sites Violation Blocked' else	
<pre>Visits' 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>	d', 'quarantined')	

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.

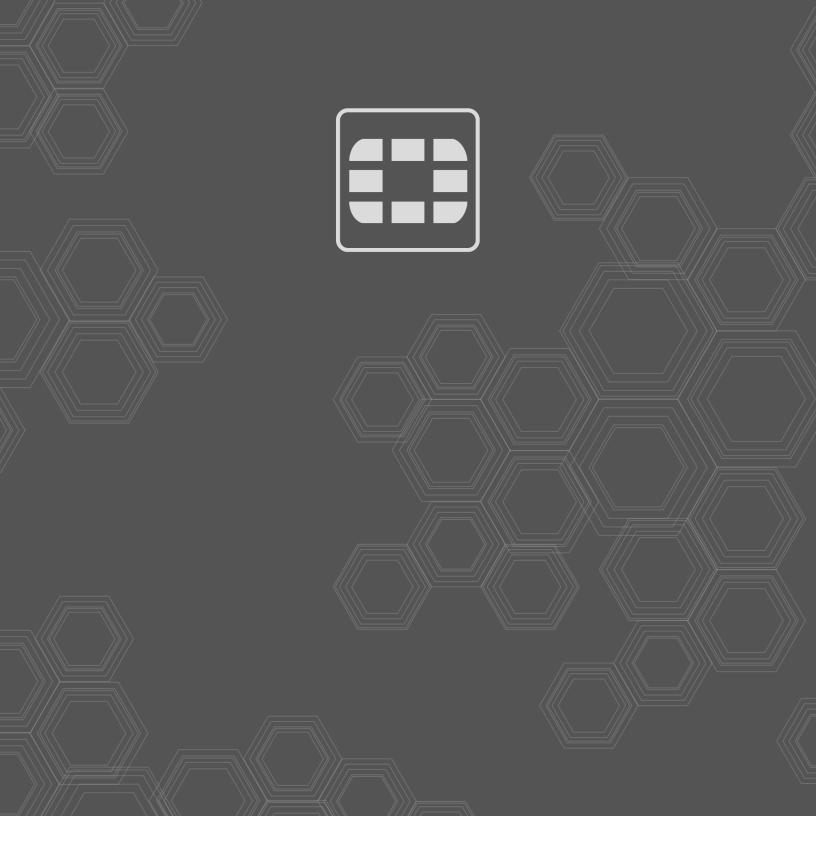
Macro Name	Description	Dataset Used	Log Category
Application Category with Highest Session Count	Application category with the highest session count	App-Sessions-By- Category	Traffic
Application with Highest Bandwidth	Application with the highest bandwidth usage	Top-App-By-Bandwidth	Traffic
Application 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

2021-05-13 Initial release.





Copyright© 2021 Fortinet, Inc. All rights reserved. Fortinet®, FortiGate®, FortiCare® 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 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.