

FortiAnalyzer - Dataset Reference

Version 5.6.9



FORTINET DOCUMENT LIBRARY

https://docs.fortinet.com

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

FORTINET BLOG https://blog.fortinet.com

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

FORTINET TRAINING & CERTIFICATION PROGRAM

https://www.fortinet.com/support-and-training/training.html

NSE INSTITUTE https://training.fortinet.com

FORTIGUARD CENTER https://fortiguard.com/

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

FEEDBACK Email: techdocs@fortinet.com



July 18, 2019 FortiAnalyzer 5.6.9 Dataset Reference 05-569-438683-20190718

TABLE OF CONTENTS

Change Log	. 4
Introduction	. 5
Understanding datasets and macros	. 5
Dataset Reference List	. 6
Macro Reference List	69

Change Log

Date	Change Description	
2019-07-18	Initial release.	

Introduction

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

Understanding datasets and macros

FortiAnalyzer datasets are collections of log messages from monitored devices.

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

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

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

Dataset Reference List

The following tables list the available predefined data sets reported by FortiAnalyzer. For documentation and technical support reference purposes, thess tables contain the dataset names, SQL query syntax for each dataset, and the log category of the dataset.

Dataset Name	Description	Log Category
Traffic-Bandwidth-Summary-Day-Of- Month	Traffic bandwidth timeline	traffic
<pre>select \$flex_timescale(times sum(traffic_out) as t sum(traffic in) as tr</pre>	raffic_out,	
from	—	
<pre>###(select \$flex_time 0)) as traffic_out, sum</pre>		<pre>sum(coalesce(sentbyte, 0)) as traffic_in from</pre>

Dataset Name	Description	Log Category
Session-Summary-Day-Of-Month	Number of session timeline	traffic
from \$log where \$filte		n (4, 7, 14,
Dataset Name	Description	Log Category
Top-Users-By-Bandwidth	Bandwidth application top users by bandwidth usage	traffic
select coalesce(

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

```
ipstr(`srcip`)
  ) as user src,
  sum(
    coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
  ) as bandwidth,
  sum(
    coalesce(rcvdbyte, 0)
  ) as traffic in,
  sum(
    coalesce(sentbyte, 0)
  ) as traffic out,
  count(*) as sessions
from
  $log
where
  $filter
  and logid to int(logid) not in (4, 7, 14, 20)
group by
  user src
having
  sum(
    coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
  )& gt; 0
order by
  bandwidth desc
```

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

from

```
$log
where
$filter
and logid_to_int(logid) not in (4, 7, 14, 20)
and nullifna(app) is not null
group by
app_group
having
sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
)& gt; 0
order by
bandwidth desc
```

Dataset Name	Description	Log Category
Top-User-Source-By-Sessions	Top user source by session count	traffic
select		
coalesce(
nullifna(`user`)		
nullifna(`unauth	user`),	
ipstr(`srcip`)		
) as user_src,		
<pre>count(*) as sessic</pre>	ns	
from		
\$log where		
\$filter		
	ogid) not in (4, 7, 14, 20)	
group by	logia, not in (1, ,, in, 20)	
user src		
order by		
sessions desc		
Dataset Name	Description	Log Category
Top-App-By-Sessions	Top applications by session count	traffic
select		
app_group_name(app) as app_group,	
count(*) as sessic	ns	

from

```
where
  $filter
  and logid_to_int(logid) not in (4, 7, 14, 20)
  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(hos), ipstr(dstip)) as domain, count(*) as session from \$log where \$filter and logid_to_int(lo group by domain order by sessions desc</pre>		0)

Dataset Name	Description	Log Category
Top-Destination-Addresses-By- Bandwidth	Top destinations by bandwidth usage	traffic
<pre>select coalesce(nullifna(root_domain(hostn), ipstr(dstip)) as domain,</pre>	ame)	

```
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 logid_to_int(logid) not in (4, 7, 14, 20)
  and coalesce(
    nullifna(
      root domain(hostname)
    ),
    ipstr(`dstip`)
  ) is not null
group by
  domain
having
  sum(
    coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
  )& gt; 0
order by
  bandwidth desc
```

Dataset Name	Description	Log Category
DHCP-Summary-By-Port	Event top dhcp summary	event
drop		
table if exists rp	ot_tmptbl_1;	
drop		
table if exists rp	<pre>>t_tmptbl_2;</pre>	
drop		
table if exists rp	<pre>ot_tmptbl_3; create tempo</pre>	rary table rpt_tmptbl_
1 as ###(select cond	cat(interface, '.', devid) as intf, mac from
<pre>\$log where \$last3day</pre>	_period \$filter and log	id_to_int(logid) =
26001 and dhcp_msg =	<pre>- 'Ack' group by interfac</pre>	e, devid, mac)###; cre-
ate temporary table	<pre>rpt_tmptbl_2 as ###(sele</pre>	ct concat(interface,

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

Dataset Name	Description	Log Category
Top-Wifi-Client-By-Bandwidth	Traffic top WiFi client by bandwidth usage	traffic
<pre>select coalesce(nullifna(`user`), nullifna(`unauthus ipstr(`srcip`)) as user_src, srcssid, devtype, coalesce(nullifna(`srcname` `srcmac`) as hostname_mac, sum(coalesce(sentbyte,) as bandwidth</pre>		
from		
\$log		
where		
\$filter		
and logid_to_int(log and (srcssid is not nul	id) not in (4, 7, 14, 20)	
or dstssid is not		
)		
group by		

```
user_src,
srcssid,
devtype,
hostname_mac
having
sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
)& gt; 0
order by
bandwidth desc
```

Dataset Name	Description	Log Category
Traffic-History-By-Active-User	Traffic history by active user	traffic
<pre>select \$flex_timescale(time count(distinct(user_src)) as total_user</pre>	stamp) as hodex,	
from		
—	logid_to_int(logid) er_src order by times	cip`)) as user_src from not in (4, 7, 14, 20)

Dataset Name	Description	Log Category
Top-Allowed-Websites-By-Requests	UTM top allowed web sites by request	traffic
select		
hostname,		
catdesc,		
count(*) as requests		
from		
\$log		
where		
\$filter		
and logid_to_int(logi	d) not in (4, 7, 14, 20)	
and utmevent in (
'webfilter', 'banne	d-word', 'web-content',	
'command-block', 's	cript-filter'	
)		

```
and hostname is not null
and (
    utmaction not in ('block', 'blocked')
    or action != 'deny'
)
group by
   hostname,
   catdesc
order by
   requests desc
```

Dataset Name	Description	Log Category
Top-50-Websites-By-Bandwidth	Webfilter top allowed web sites by bandwidth usage	webfilter
<pre>select domain, string_agg(distinct sum(bandwidth) as ba sum(traffic_in) as t sum(traffic_out) as from ###(select coalesce) domain, catdesc, sum(coal domain, catdesc, sum(coal (coalesce(sentbyte, 0)) \$filter and logid_to_i tion!='blocked' and (co and (hostname is not r word', 'web-content', by domain, catdesc have</pre>	<pre>catdesc, ', ') as agg_catdesc, andwidth, traffic_in, traffic_out (nullifna(hostname), ipstr(`dstip coalesce(sentbyte, 0)+coalesce(rc esce(rcvdbyte, 0)) as traffic_in) as traffic_out from \$log-traff int(logid) not in (4, 7, 14, 20) countweb>0 or ((logver is null or null or utmevent in ('webfilter', 'command-block', 'script-filter' ving sum(coalesce(sentbyte, 0)+coal pandwidth desc)### t group by dom</pre>	vdbyte, 0)) , sum ic where and utmac- logver<52) 'banned-)))) group alesce(rcvd-

Dataset Name	Description	Log Category
Top-Blocked-Websites	UTM top blocked web sites by request	traffic
<pre>select hostname, count(*) as requests from \$log where</pre>		

```
$filter
and logid_to_int(logid) not in (4, 7, 14, 20)
and utmevent in (
    'webfilter', 'banned-word', 'web-content',
    'command-block', 'script-filter'
)
and hostname is not null
and (
    utmaction in ('block', 'blocked')
    or action = 'deny'
)
group by
hostname
order by
requests desc
```

Dataset Name	Description	Log Category
Top-Web-Users-By-Request	UTM top web users by request	traffic
select		
coalesce(
<pre>nullifna(`user`),</pre>		
nullifna(`unauthu:	ser`),	
ipstr(`srcip`)		
) as user_src,		
devtype,		
srcname,		
count(*) as requests	5	
from		
\$log		
where		
\$filter		
and logid_to_int(logid) not in (4, 7, 14, 20)		
<pre>and utmevent in ('webfilter', 'banned-word', 'web-content',</pre>		
'command-block', 'script-filter'		
	beripe rifeer	
group by		
user src,		
devtype,		
srcname		

```
order by requests desc
```

Dataset Name	Description	Log Category
Top-Allowed-WebSites-By-Bandwidth	UTM top allowed websites by bandwidth usage	traffic
select		
appid,		
hostname,		
catdesc,		
sum(
	0)+ coalesce(rcvdbyte, 0)	
) as bandwidth,		
sum (
coalesce(rcvdbyte,	0)	
) as traffic_in,		
<pre>sum(coalesce(sentbyte,</pre>	0)	
) as traffic out	0)	
from		
\$log		
where		
\$filter		
and logid_to_int(log	id) not in (4, 7, 14, 20)	
and utmevent in (
	ed-word', 'web-content',	
'command-block', '	script-filter'	
)		
and hostname is not a	null	
group by		
appid,		
hostname, catdesc		
having		
sum (
-	0)+ coalesce(rcvdbyte, 0)	
)& gt; 0		
order by		
bandwidth desc		

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

```
select
  coalesce(
    nullifna(`user`),
    nullifna(`unauthuser`),
    ipstr(`srcip`)
  ) as user src,
  devtype,
  srcname,
  count(*) as requests
from
  $log
where
  $filter
  and logid_to_int(logid) not in (4, 7, 14, 20)
  and utmevent in (
    'webfilter', 'banned-word', 'web-content',
    'command-block', 'script-filter'
  )
  and (
    utmaction in ('block', 'blocked')
    or action = 'deny'
  )
group by
  user src,
  devtype,
  srcname
order by
  requests desc
```

Dataset Name	Description	Log Category
Top-20-Web-Users-By-Bandwidth	Webfilter top web users by bandwidth usage	webfilter
<pre>select user_src,</pre>		
sum(bandwidth) as ban		
<pre>sum(traffic_in) as tr sum(traffic_out) as t</pre>	<u> </u>	
from		
###(select coalesce(n	ullifna(`user`), nullifna(`unau	thuser`),
<pre>ipstr(`srcip`)) as user</pre>	_src, sum(coalesce(sentbyte, 0)	+coalesce
(rcvdbyte, 0)) as bandw	<pre>idth, sum(coalesce(rcvdbyte, 0)</pre>) as
<pre>traffic_in, sum(coalesc</pre>	e(sentbyte, 0)) as traffic_out	from \$log-

traffic where \$filter and logid_to_int(logid) not in (4, 7, 14, 20) and (countweb>0 or ((logver is null or logver<52) and (hostname is not null or utmevent in ('webfilter', 'banned-word', 'webcontent', 'command-block', 'script-filter')))) group by user_src having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc)### t group by user_src order by bandwidth desc

```
Dataset Name
                           Description
                                                                Log Category
Top-Web-Users-By-Bandwidth
                           UTM top web users by bandwidth usage
                                                                 traffic
select
  coalesce(
    nullifna(`user`),
    nullifna(`unauthuser`),
    ipstr(`srcip`)
  ) as user src,
  devtype,
  srcname,
  sum(
    coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
  ) as bandwidth,
  sum(
    coalesce(rcvdbyte, 0)
  ) as traffic in,
  sum(
    coalesce(sentbyte, 0)
  ) as traffic out
from
  $log
where
  $filter
  and logid to int(logid) not in (4, 7, 14, 20)
  and utmevent in (
    'webfilter', 'banned-word', 'web-content',
    'command-block', 'script-filter'
  )
group by
  user src,
  devtype,
  srcname
having
  sum(
```

```
coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
)& gt; 0
order by
bandwidth desc
```

Dataset Name Description Log Category Top-Video-Streaming-Websites-By-UTM top video streaming websites by bandwidth usage traffic Bandwidth 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 logid to int(logid) not in (4, 7, 14, 20) and catdesc in ('Streaming Media and Download') group by appid, hostname having sum(coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0))& gt; 0 order by bandwidth desc

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

Dataset Name

```
nullifna(`user`),
    nullifna(`unauthuser`),
    ipstr(`srcip`)
  ) as user src,
  count(*) as requests
from
  $log
where
  $filter
  and logid to int(logid) not in (4, 7, 14, 20)
  and service in (
    'smtp', 'SMTP', '25/tcp', '587/tcp',
    'smtps', 'SMTPS', '465/tcp'
  )
group by
  user src
order by
  requests desc
```

Description Top-Email-Receivers-By-Count Default email top receivers by count select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)

```
) as user src,
  count(*) as requests
from
  $log
where
  $filter
  and logid_to_int(logid) not in (4, 7, 14, 20)
  and service in (
    'pop3', 'POP3', '110/tcp', 'imap',
    'IMAP', '143/tcp', 'imaps', 'IMAPS',
    '993/tcp', 'pop3s', 'POP3S', '995/tcp'
group by
 user src
```

Log Category

traffic

```
order by
```

requests desc

Dataset Name	Description	Log Category
Top-Email-Senders-By-Bandwidth	Default email top senders by bandwidth usage	traffic
	0)+ coalesce(rcvdbyte, 0)	
) as bandwidth from		
\$log		
where		
\$filter		
<pre>and logid_to_int(log: and service in ('smtp', 'SMTP', '25</pre>	id) not in (4, 7, 14, 20)	
'smtps', 'SMTPS',		
)		
group by user_src		
having sum(
<pre>coalesce(sentbyte,) & gt; 0 order by</pre>	0)+ coalesce(rcvdbyte, 0)	
bandwidth desc		

Dataset Name	Description	Log Category
Top-Email-Receivers-By-Bandwidth	Default email top receivers by bandwidth usage	traffic
<pre>select coalesce(nullifna(`user`), nullifna(`unauthuse: ipstr(`srcip`)) as user_src,</pre>	r`),	

```
sum(
    coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
  ) as bandwidth
from
  $log
where
  $filter
  and logid_to_int(logid) not in (4, 7, 14, 20)
  and service in (
    'pop3', 'POP3', '110/tcp', 'imap',
    'IMAP', '143/tcp', 'imaps', 'IMAPS',
    '993/tcp', 'pop3s', 'POP3S', '995/tcp'
  )
group by
 user src
having
  sum(
    coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
  )& gt; 0
order by
 bandwidth desc
```

Dataset Name	Description	Log Category
Top-Malware-By-Name	UTM top virus	virus
<pre>select virus, max(virusid_s) as</pre>	s virusid,	
	n 'Adware' else 'Vir e,	hen 'Spyware' when virus us' end
from		
<pre>\$log-traffic where 14, 20) and utmever</pre>	\$filter and logid_to nt is not null and v	<pre>count(*) as totalnum from o_int(logid) not in (4, 7, irus is not null group by c)### union all ###(select</pre>

as totalnum from \$log-virus where \$filter and (eventtype is null or logver>=52) and nullifna(virus) is not null group by virus,

virus, virusid to str(virusid, eventtype) as virusid s, count(*)

virusid_s order by totalnum desc)###) t group by virus, malware_ type order by totalnum desc

Dataset Name	Description		Log Category
Top-Virus-By-Name	UTM top virus		virus
<pre>select virus, max(virusid_s) a (</pre>	s virusid,		
			when virus
from			
<pre>\$log-traffic where 14, 20) and utmeve virus, virusid_s c virus, virusid_to_ as totalnum from \$ or logver>=52) and</pre>	us, '' as virusid_s, \$filter and logid_t ent is not null and v order by totalnum des str(virusid, eventty log-virus where \$fil nullifna(virus) is talnum desc)###) t g desc	o_int(logid) n virus is not nu sc)### union al vpe) as virusid ter and (event not null group	ot in (4, 7, ll group by l ###(select _s, count(*) type is null by virus, vir

Dataset Name	Description	Log Category
Top-Virus-Victim	UTM top virus user	traffic
<pre>select user_src, sum(totalnum) a</pre>	s totalnum	
from		
(
###(select co	<pre>alesce(nullifna(`user`), nu</pre>	llifna(`unauthuser`),
<pre>ipstr(`srcip`)) a</pre>	<pre>s user_src, count(*) as tot</pre>	alnum from \$log-
traffic where \$fi	<pre>lter and logid_to_int(logid</pre>) not in (4, 7, 14,
20) and utmevent	is not null and virus is no	t null group by user_
src order by tota	<pre>lnum desc)### union all ###</pre>	(select coalesce(nul-
lifna(`user`), ip	<pre>str(`srcip`)) as user_src,</pre>	count(*) as totalnum

from \$log-virus where \$filter and (eventtype is null or logver>-

=52) and nullifna(virus) is not null group by user_src order by totalnum desc)###) t group by user_src order by totalnum desc

Dataset Name	Description	Log Category
Top-Attack-Source	UTM top attack source	attack
<pre>select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, count(*) as totalnum from \$log where \$filter group by user_src order by totalnum desc</pre>		

Dataset Name	Description	Log Category
Top-Attack-Victim	UTM top attack dest	attack
select		
dstip,		
count(*) as totalnum		
from		
\$log		
where		
\$filter		
and dstip is not null		
group by		
dstip		
order by		
totalnum desc		

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

```
select
 vpn name,
  sum (bandwidth) as bandwidth,
  sum(traffic in) as traffic in,
  sum(traffic out) as traffic out
from
  (
    select
      devid,
      vd,
      remip,
      tunnelid,
      vpn name,
        case when min(s time) = max(e time) then max(max traffic
in) else max(max traffic in) - min(min traffic in) end
      ) as traffic in,
        case when min(s time) = max(e time) then max(max traffic
out) else max(max traffic out) - min(min traffic out) end
      ) as traffic out,
        case when min(s time) = max(e time) then max(max traffic
in) + max(max traffic out) else max(max traffic in) - min(min
traffic in) + max(max traffic out) - min(min traffic out) end
      ) as bandwidth
    from
      ###(select devid, vd, remip, vpn trim(vpntunnel) as vpn
name, tunnelid, 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 (tunnelip is null or tunnelip='0.0.0.0') and
action in ('tunnel-stats', 'tunnel-down') and tunnelid is not null
group by devid, vd, remip, vpn name, tunnelid) ### t group by
```

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

devid, vd, remip, vpn name, tunnelid) tt group by vpn name having

sum(traffic in+traffic out)>0 order by bandwidth desc

```
select
 user src,
 remip as remote ip,
 from dtime(
   min(s time)
 ) as start time,
  sum (bandwidth) as bandwidth,
 sum(traffic in) as traffic in,
  sum(traffic out) as traffic out
from
  (
    select
      devid,
      vd,
      remip,
     user src,
      tunnelid,
     min(s time) as s time,
     max(e time) as e time,
      (
        case when min(s time) = max(e time) then max(max traffic
in) + max(max traffic out) else max(max traffic in) - min(min
traffic in) + max(max traffic out) - min(min traffic out) end
      ) as bandwidth,
        case when min(s time) = max(e time) then max(max traffic
in) else max(max traffic in) - min(min traffic in) end
      ) as traffic in,
        case when min(s time) = max(e time) then max(max traffic
out) else max(max traffic out) - min(min traffic out) end
      ) as traffic out
    from
      ###(select devid, vd, remip, coalesce(nullifna(`user`),
ipstr(`remip`)) as user src, tunnelid, 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 sub-
type='vpn' and tunneltype='ssl-tunnel' and action in ('tunnel-
```

ipstr(`remip`)) is not null and tunnelid is not null group by devid, vd, user_src, remip, tunnelid)### t group by devid, vd, user_src, remip, tunnelid) tt group by user_src, remote_ip having sum(bandwidth)>0 order by bandwidth desc

```
Dataset Name
                         Description
                                                             Log Category
Top-Dial-Up-IPSEC-Tunnels-By-
                         Top dial up IPsec tunnels by bandwidth usage
                                                             event
Bandwidth
select
  vpn name,
  sum(bandwidth) as bandwidth,
  sum(traffic in) as traffic in,
  sum(traffic out) as traffic out
from
  (
    select
      devid,
      vd,
      tunnelid,
      remip,
      vpn name,
      (
        case when min(s time) = max(e time) then max(max traffic
in) else max(max traffic in) - min(min traffic in) end
      ) as traffic in,
        case when min(s time) = max(e time) then max(max traffic
out) else max(max traffic out) - min(min traffic out) end
      ) as traffic out,
      (
        case when min(s time) = max(e time) then max(max traffic
in) + max(max traffic out) else max(max traffic in) - min(min
traffic in) + max(max traffic out) - min(min traffic out) end
      ) as bandwidth
    from
      ###(select devid, vd, remip, vpn_trim(vpntunnel) as vpn_
name, tunnelid, 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 nullifna(vpntunnel) is not null
```

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') and tunnelid is not null group by devid, vd, remip, vpn_name, tunnelid)### t 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 bas sum(traffic_in) as t sum(traffic_out) as from</pre>	craffic_in,	
string_agg(disti remip, tunnelid, min(s_time) as s max(e_time) as e (time,	_
<pre>in) + max(max_traffic_c traffic_in) + max(max_t) as bandwidth,</pre>	<pre>(s_time) = max(e_time) then max(m out) else max(max_traffic_in) - m traffic_out) - min(min_traffic_ou (s_time) = max(e_time) then max(m)</pre>	in(min_ t) end

```
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, nullifna(`xauthuser`) as xauthuser_ agg, nullifna(`user`) as user_agg, remip, tunnelid, 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 'ipsec%' and not (tunnelip is null or tunnelip='0.0.0.0') and action in ('tunnelstats', 'tunnel-down', 'tunnel-up') and tunnelid is not null group by devid, vd, xauthuser_agg, user_agg, remip, tunnelid)### t group by devid, vd, remip, tunnelid) tt group by user_src, remip having sum(bandwidth)>0 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 dura sum(bandwidth) as ban sum(traffic_in) as tr</pre>	dwidth, affic_in,	
<pre>sum(traffic_out) as t from (select devid, vd, remip,</pre>	raffic_out	

28

```
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 xau-
thuser agg, nullifna(`user`) as user agg, tunnelid, min(coalesce
(dtime, 0)) as s time, max(coalesce(dtime, 0)) as e time, max
(coalesce(duration, 0)) as max duration, min(coalesce(duration, 0))
```

as min_duration, min(coalesce(sentbyte, 0)) as min_traffic_out, min(coalesce(rcvdbyte, 0)) as min_traffic_in, max(coalesce(sentbyte, 0)) as max_traffic_out, max(coalesce(rcvdbyte, 0)) as max_ traffic_in from \$log where \$filter and subtype='vpn' and tunneltype like 'ipsec%' and not (tunnelip is null or tunnelip='0.0.0.0') and action in ('tunnel-stats', 'tunnel-down', 'tunnel-up') and tunnelid is not null and tunnelid!=0 group by devid, vd, remip, xauthuser_agg, user_agg, tunnelid order by tunnelid)### t group by devid, vd, remip, tunnelid) tt group by user_ src having sum(bandwidth)>0 order by duration desc

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

```
select
 user src,
  remip as remote ip,
  from dtime(
    min(s time)
  ) as start time,
  sum (bandwidth) as bandwidth,
  sum(traffic in) as traffic in,
  sum(traffic out) as traffic out
from
  (
    select
      devid,
      vd,
      user src,
      remip,
      tunnelid,
      min(s time) as s time,
      max(e time) as e time,
      (
        case when min(s time) = max(e time) then max(max traffic
in) + max(max traffic out) else max(max traffic in) - min(min
traffic in) + max(max traffic out) - min(min traffic out) end
      ) as bandwidth,
        case when min(s time) = max(e time) then max(max traffic
in) else max(max traffic in) - min(min traffic in) end
      ) as traffic in,
        case when min(s time) = max(e time) then max(max traffic
out) else max(max traffic out) - min(min traffic out) end
      ) as traffic out
    from
      ###(select devid, vd, coalesce(nullifna(`user`), ipstr
(`remip`)) as user src, remip, tunnelid, min(coalesce(dtime, 0))
as s time, max(coalesce(dtime, 0)) as e time, min(coalesce(sent-
byte, 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='ssl-web' 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)### t group by devid, vd, user_src, remip, tunnelid) tt group by user_src, remote_ip having sum(bandwidth)>0 order by bandwidth desc

Dataset Name	Description	Log Category
Top-SSL-VPN-Web-Mode-Users-By- Duration	Top SSL VPN web mode users by duration	event
<pre>select user_src, remip as remote_ip, from_dtime(min(s_time)) as start_time, (max(e_time) - min(s_) as duration</pre>	_time)	
<pre>from (select devid, vd, user_src, remip, tunnelid, min(s_time) as s_ max(e_time) as e_</pre>	—	
<pre>(`remip`)) as user_src, as s_time, max(coalesce ter and subtype='vpn' a nel-stats', 'tunnel-dow (`user`), ipstr(`remip" group by devid, vd, use</pre>	, vd, coalesce(nullifna(`user` , remip, tunnelid, min(coalesc e(dtime, 0)) as e_time from \$1 and tunneltype='ssl-web' and a wn', 'tunnel-up') and coalesce `)) is not null and tunnelid i er_src, remip, tunnelid)### t emip, tunnelid) tt group by us ation desc	e(dtime, 0)) og where \$fil- ction in ('tun (nullifna s not null group by

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, 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 'ssl%' and action in ('tunnel-up', 'tunnel-stats', 'tunneldown') and coalesce(nullifna(`user`), ipstr(`remip`)) is not null and tunnelid is not null and tunnelid!=0 group by devid, vd, remip, user_src, tunnelid, tunneltype)### t group by devid, vd, remip, user_src, tunnelid, tunneltype) tt group by user_src, tunneltype having sum(traffic_out+traffic_in)>0 order by duration desc

Dataset Name	Description	Log Category
vpn-Top-Dial-Up-VPN-Users-By- Duration	Top dial up VPN users by duratior	event
<pre>select coalesce(xauthuser_agg, user_agg, ipstr(`remip`)) as user_src, t_type as tunneltype, from_dtime(min(s_time)) as start_time, sum(duration) as dura sum(duration) as dura sum(bandwidth) as ban sum(traffic_in) as tr sum(traffic_out) as tr sum(traffic_out) as tr from (select devid, vd, remip, string_agg(disting string_agg(disting)</pre>	ation, adwidth, caffic_in, craffic_out	') as xauthuser_agg, s user_agg,
t_type, tunnelid, min(s_time) as s_ max(e time) as e	-	
(-	then max(max duration)
else max(max_duration)-		—

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, 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 'ssl%' or (tunneltype like 'ipsec%' and not (tunnelip is null or tunnelip='0.0.0.0'))) and action in ('tunnelstats', 'tunnel-down', 'tunnel-up') and tunnelid is not null and tunnelid!=0 group by devid, vd, remip, xauthuser agg, user agg, t type, tunnelid) ### t group by devid, vd, remip, t type, tunnelid) tt group by user src, tunneltype having sum(bandwidth)>0 order by duration desc

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

```
devid,
vd,
remip,
tunnelid,
sum(tunnelup) as total_num,
max(traffic_in) as traffic_in,
max(traffic_out) as traffic_out
```

from

###(select \$flex_timestamp as timestamp, devid, vd, remip, tunnelid, (case when action='tunnel-up' then 1 else 0 end) as tunnelup, max(coalesce(sentbyte, 0)) as traffic_out, max(coalesce (rcvdbyte, 0)) as traffic_in from \$log where \$filter and subtype='vpn' and (tunneltype like 'ipsec%' or tunneltype like 'ssl%') and action in ('tunnel-up', 'tunnel-stats', 'tunnel-down') and tunnelid is not null group by timestamp, action, devid, vd, remip, tunnelid order by timestamp desc)### 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>select f_user, tunneltype, sum(total_num) as too from ###(select coalesce(second) tunneltype, count(*) a type='vpn' and (tunnels) and action in ('ssl-loo (nullifna(`xauthuser`)) user, tunneltype)### t num desc</pre>	- nullifna(`xauthuser`), s total_num from \$log w type='ipsec' or left(tw gin-fail', 'ipsec-logiw , nullifna(`user`)) is	where \$filter and sub- unneltype, 3)='ssl') n-fail') and coalesce not null group by f_

Dataset Name	Description	Log Category
vpn-Authenticated-Logins	VPN authenticated logins	event
<pre>select coalesce(xauthuser_agg,</pre>		

```
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 nullifna(`xauthuser`) as xauthuser agg, nullifna
```
(`user`) as user agg, devid, vd, remip, (case when tunneltype like 'ipsec%' then 'ipsec' else tunneltype end) as t type, tunnelid, sum((case when action='tunnel-up' then 1 else 0 end)) as tunnelup, 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%' 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) ### t group by t type, devid, vd, remip, tunnelid having max(tunnelup) > 0) tt group by f user, tunneltype having sum(bandwidth) > 0 order by total num desc

Dataset Name	Description	Log Category
vpn-Traffic-Usage-Trend-VPN- Summary	VPN traffic usage trend	event
	dwidth) as ssl_bandwidth, bandwidth) as ipsec_bandwid	th
(
select \$flex_timescale devid, vd, remip, tunnelid, (e(timestamp) as hodex,	
case when t t	ype like 'ssl%' then (
case when m in)+ max(max_traffic_	<pre>in(s_time) = max(e_time) the out) else max(max_traffic_: traffic_out) - min(min_traf;</pre>	in)- min(min
case when t_t	ype like 'ipsec%' then (

```
case when min(s_time) = max(e_time) then max(max_traffic_
in) + max(max_traffic_out) else max(max_traffic_in) - min(min_
traffic_in) + max(max_traffic_out) - min(min_traffic_out) end
) else 0 end
) as ipsec_traffic_bandwidth,
min(s_time) as s_time,
max(e_time) as e_time
from
```

###(select \$flex_timestamp as timestamp, devid, vd, remip, tunnelid, (case when tunneltype like 'ipsec%' then 'ipsec' else tunneltype end) as t_type, max(coalesce(sentbyte, 0)) as max_ traffic_out, max(coalesce(rcvdbyte, 0)) as max_traffic_in, min (coalesce(sentbyte, 0)) as min_traffic_out, min(coalesce(rcvdbyte, 0)) as min_traffic_in, min(coalesce(dtime, 0)) as s_time, max (coalesce(dtime, 0)) as e_time from \$log where \$filter and subtype='vpn' and (tunneltype like 'ipsec%' or tunneltype like 'ssl%') and action in ('tunnel-stats', 'tunnel-down') and tunnelid is not null and tunnelid!=0 group by timestamp, devid, vd, remip, t_type, tunnelid order by timestamp desc)### t group by hodex, devid, t_type, vd, remip, tunnelid) tt group by hodex order by hodex

Dataset Name	Description	Log Category
Top-S2S-IPSEC-Tunnels-By- Bandwidth-and-Availability	Top S2S IPsec tunnels by bandwidth usage and avail	event
<pre>select vpntunnel, tunneltype, sum(traffic_out) as sum(traffic_in) as sum(bandwidth) as k sum(uptime) as uption from (select vpntunnel, tunneltype, tunnelid, devid, vd, sum(sent end -</pre>	traffic_in, pandwidth,	

```
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 order by tunnelid)### t group by vpntunnel, tunneltype, tunnelid, devid, vd order by bandwidth desc) t group by vpntunnel, tunneltype order by bandwidth desc

Dataset Name	Description	Log Category
Top-Dialup-IPSEC-By-Bandwidth-and- Availability	Top dialup IPsec users by bandwidth usage and avail	event
select		
user_src, remip, sum(traffic out) as t	raffic out	
<pre>sum(traffic_out) as t sum(traffic_in) as tr sum(bandwidth) as ban</pre>	affic_in,	
<pre>sum(uptime) as uptime from</pre>		
(
select		
user_src, remip,		
tunnelid, devid, vd,		
sum(sent_end - se	nt_beg) as traffic_out, vd_beg) as traffic_in,	
	_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 order by tunnelid)### t group by user_src, remip, tunnelid, devid, vd order by bandwidth desc) t group by user_src, remip order by bandwidth desc

Dataset Name	Description	Log Category
Top-SSL-Tunnel-Mode-By-Bandwidth- and-Availability	Top SSL tunnel users by bandwidth usage and avail	event
<pre>sum(rcvd_end - rc sum(sent_end - sent) as bandwidth,</pre>	affic_in, dwidth,	

###(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 order by tunnelid)### t group by user_src, remote_ip, tunnelid, devid, vd order by bandwidth desc) t group by user_src, remote_ip order by bandwidth desc

```
Dataset Name
                          Description
                                                               Log Category
Top-SSL-Web-Mode-By-Bandwidth-
                          Top SSL web 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='ssl-web' and coalesce(nullifna(`user`), ipstr(`remip`)) is not null and tunnelid is not null group by tunnelid, user_src, remip, devid, vd order by tunnelid)### 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 group by user_src, remote_ip order by bandwidth desc

```
Dataset Name
                          Description
                                                               Log Category
Admin-Login-Summary
                          Event admin login summary
                                                               event
select
  f user,
  ui,
  sum(login) as total num,
  sum(login duration) as total duration,
  sum(config change) as total change
from
  (
    select
      `user` as f user,
      ui,
      (
        case when logid to int(logid) = 32001 then 1 else 0 end
      ) as login,
        case when logid to int(logid) = 32003 then duration else 0
end
      ) as login duration,
      (
        case when logid to int(logid) = 32003
        and state is not null then 1 else 0 end
      ) as config change
    from
      $log
    where
      $filter
      and nullifna(`user`) is not null
      and logid to int(logid) in (32001, 32003)
```

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

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

select

\$flex_timescale(timestamp) as dom, sum(total_num) as total_num, sum(total_change) as total_change

from

###(select timestamp, sum(login) as total_num, sum(config_ change) as total_change from (select \$flex_timestamp as timestamp, (case when logid_to_int(logid)=32001 then 1 else 0 end) as login, (case when logid_to_int(logid)=32003 and state is not null then 1 else 0 end) as 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 order by timestamp desc)### t group by dom order by dom

Dataset Name	Description	Log Category
Admin-Failed-Login-Summary	Event admin failed login summary	event
select		
`user` as f_user,		
ui,		
count(status) as to	tal_failed	
from		
\$log		
where		
\$filter		
and nullifna(`user`) is not null		
and logid_to_int(logid) = 32002		
group by		
ui,		
f_user		

order by

total_failed desc

Dataset Name	Description	Log Category
System-Summary-By-Severity	Event system summary by severity	event
<pre>gency') then 'Critica ='warning' then 'Medi 'Info' end) as severi \$filter and subtype='</pre>	_	High' when level- 'Low' else m from \$log where p order by total_
Dataset Name	Description	Log Category
System-Summary-By-Date	Event system summary by date	event
<pre>('critical', 'alert', sum(case when level = when level = 'warning where \$filter and sub</pre>	itical,	end) as critical, as high, sum(case um from \$log
Dataset Name	Description	Log Category
Important-System-Summary-By-Dat	e Event system summary by date	event
<pre>select \$flex_timescale(tim sum(critical) as cr sum(high) as high, sum(medium) as medi</pre>	itical,	

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 order by timestamp desc)### t group by dom order by dom

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

select

```
msg_desc as msg,
severity_tmp as severity,
sum(count) as counts
```

from

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

Dataset Name	Description	Log Category
System-High-Severity-Events	Event system high severity events	event
<pre>select msg_desc as msg, severity_tmp as sever</pre>	rity,	
<pre>sum(count) as counts from ###(select coalesce(n</pre>	ullifna(logdesc), msg) as msg des	sc, (case
when level='error' then	I', 'alert', 'emergency') then '('High' when level='warning' the	n 'Medium'
<pre>count(*) as count from group by msg_desc, seve</pre>	en 'Low' else 'Info' end) as seven \$log where \$filter and subtype='s erity_tmp order by count desc)### oup by msg, severity tmp order by	system' t where

desc

Dataset Name	Description	Log Category
System-Medium-Severity-Events	Event system medium severity events	event
<pre>select msg_desc as msg, severity_tmp as seve sum(count) as counts</pre>	_	
<pre>from ###(select coalesce(when level in ('critic when level='error' the</pre>	nullifna(logdesc), msg) as ms al', 'alert', 'emergency') th n 'High' when level='warning'	hen 'Critical' ' then 'Medium'
<pre>count(*) as count from group by msg_desc, sev</pre>	en 'Low' else 'Info' end) as \$\$ \$log where \$filter and subty erity_tmp order by count desc group by msg, severity_tmp or	ype='system' c)### t where

```
desc
```

Dataset Name	Description	Log Category
utm-drilldown-Top-Traffic-Summary	UTM drilldown traffic summary	traffic
select		
srcip,		
srcname		
from		
###(select coalesce(n	ullifna(`user`), nullifna(`	unauthuser`),
<pre>ipstr(`srcip`)) as user</pre>	_src, srcip, srcname, sum(c	oalesce(sent-
byte, 0)+coalesce(rcvdb	yte, 0)) as bandwidth from	\$log where \$fil-
<pre>ter and logid_to_int(lc</pre>	gid) not in (4, 7, 14, 20)	group by user_
src, srcip, srcname ord	ler by bandwidth desc)### t	where \$filter-
drilldown group by srci	p, srcname	

Dataset Name	Description	Log Category
utm-drilldown-Top-User-Destination	UTM drilldown top user destination	traffic
<pre>select appid, app, dstip, sum(sessions) as sess sum(bandwidth) as bar</pre>	-	
<pre>from ###(select coalesce(r</pre>	nullifna(`user`), nullifna(`	unauthuser`),

ipstr(`srcip`)) as user_src, appid, app, dstip, count(*) as sessions, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from \$log where \$filter and logid_to_int(logid) not in (4, 7, 14, 20) and dstip is not null and nullifna(app) is not null group by user_src, appid, app, dstip having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc)### t where \$filter-drilldown group by appid, app, dstip order by bandwidth desc

Dataset Name	Description	Log Category		
utm-drilldown-Email-Senders- Summary	UTM drilldown email senders summary	traffic		
<pre>select sum(requests) as requests, 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 logid_to_int(logid) not in (4, 7, 14, 20) 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</pre>				
Dataset Name	Description	Log Category		
utm-drilldown-Email-Receivers- Summary	UTM drilldown email receivers summary	traffic		
<pre>select sum(requests) as re sum(bandwidth) as } from</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 logid_to_int(logid) not in (4, 7, 14, 20) and recipient is not null and service in ('pop3', 'POP3', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/tcp', 'pop3s', 'POP3S', '995/tcp') group by user_src, recipient order by requests desc)### t where \$filter-drilldown

Dataset Name	Description	Log Category	
utm-drilldown-Top-Email-Recipients- By-Bandwidth	UTM drilldown top email recipients	traffic	
<pre>lpstr(`srcip`)) as user (coalesce(sentbyte, 0)- slog where \$filter and and service in ('pop3', '143/tcp', 'imaps', 'IN '995/tcp') group by use t where \$filter-drilldo</pre>	ndwidth nullifna(`user`), nullifna(`unau c_src, recipient, count(*) as re +coalesce(rcvdbyte, 0)) as bandw logid_to_int(logid) not in (4, , 'POP3', '110/tcp', 'imap', 'IM MAPS', '993/tcp', 'pop3s', 'POP3 er_src, recipient order by reque own and recipient is not null gr dth)>0 order by bandwidth desc	quests, sum idth from 7, 14, 20) AP', S', sts desc)###	
Dataset Name	Description	Log Category	
utm-drilldown-Top-Email-Senders-By- Bandwidth	UTM drilldown top email senders	traffic	
<pre>select sender, sum(bandwidth) as bandwidth from ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, sender, count(*) as requests, sum (coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from \$log where \$filter and logid_to_int(logid) not in (4, 7, 14, 20) 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</pre>			
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 logid_to_int(logid) not in (4, 7, 14, 20) and (countweb>0 or ((logver is null or logver<52) and (hostname is not null or utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter')))) and hostname is not null group by user_src, appid, hostname, blocked order by bandwidth desc)### t where \$filter-drilldown and blocked=0 group by appid, hostname order by bandwidth desc

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

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

(

###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, appid, hostname, (case when utmaction='blocked' then 1 else 0 end) as blocked, count(*) as requests from \$log-traffic where \$filter and logid_to_int(logid) not in (4, 7, 14, 20) and utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') and hostname is not null group by user_src, appid, hostname, blocked order by requests desc)### union all ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, 0 as appid, hostname, (case when action-='blocked' then 1 else 0 end) as blocked, count(*) as requests from \$log-webfilter where \$filter and (eventtype is null or logver>=52) and hostname is not null group by user_src, appid, hostname, blocked order by requests desc)###) t where \$filterdrilldown and blocked=1 group by appid, hostname order by requests desc

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

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

###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, virus, count(*) as totalnum from \$log-traffic where \$filter and logid_to_int(logid) not in (4, 7, 14, 20) and utmevent is not null and virus is not null group by user_src, virus order by totalnum desc)### union all ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, virus, count(*) as totalnum from \$log-virus where \$filter and (eventtype is null or logver>=52) and nullifna(virus) is not null group by user_src, virus order by totalnum desc)###) t where \$filter-drilldown group by virus order by totalnum desc

Dataset Name	Description	Log Category
utm-drilldown-Top-Attacks	UTM drilldown top attacks by name	attack
<pre>select attack, sum(attack_count) as from</pre>	attack_count	

###(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
<pre>src, vuln, count(*) a ='vuln-detection' and</pre>	e(nullifna(`user`), ipstr(`srci s totalnum from \$log where \$fi vuln is not null group by use c)### t where \$filter-drilldow	lter and actioner_src, vuln

Dataset Name	Description	Log Category		
utm-drilldown-Top-App-By-Bandwidth	UTM drilldown top applications by bandwidth usage	traffic		
<pre>select appid, app, sum(bandwidth) as bandwidth from ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, appid, app, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, count(*) as sessions from \$log where \$filter and logid_to_int(logid) not in (4, 7, 14, 20) and nullifna(app) is not null group by user_src, appid, app order by sessions desc)### t where \$filter-drilldown group by appid, app having sum(bandwidth)>0 order by bandwidth desc</pre>				
Dataset Name	Description	Log Category		
utm-drilldown-Top-App-By-Sessions	UTM drilldown top applications by session count	traffic		
<pre>select appid, app, sum(sessions) as sessions from ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, appid, app, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, count(*) as sessions from \$log where \$filter and logid_to_int(logid) not in (4, 7, 14, 20) and nullifna(app) is not null group by user_src, appid, app order by sessions desc)### t where \$filter-drilldown group by appid, app order by sessions desc</pre>				
order by sessions desc		у аррій, арр		
-	Description	Log Category		

```
select
  coalesce(
    nullifna(`user`),
    nullifna(`unauthuser`),
    ipstr(`srcip`)
  ) as dldn_user,
    count(*) as session,
```

```
sum(
    coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
  ) as bandwidth,
  sum(
    coalesce(sentbyte, 0)
  ) as traffic out,
  sum(
    coalesce(rcvdbyte, 0)
  ) as traffic in
from
  $log
where
  $filter
  and logid_to_int(logid) not in (4, 7, 14, 20)
group by
  dldn user
having
  sum(
    coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
  )& gt; 0
order by
  bandwidth desc
```

Dataset Name	Description		Log Category
bandwidth-app-Top-App-By- Bandwidth-Sessions	Top applications by bandwidth usage		traffic
<pre>select app_group_name(app) sum(</pre>	<pre>0)+ coalesce(rcvdbyte, 0)</pre>	0)	
from \$log where			

```
$filter
and logid_to_int(logid) not in (4, 7, 14, 20)
and nullifna(app) is not null
group by
app_group
having
sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
)& gt; 0
order by
bandwidth desc
```

Dataset Name	Description	Log Category		
bandwidth-app-Category-By- Bandwidth	Application risk application usage by category	traffic		
<pre>select appcat, sum(coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)</pre>				
) as bandwidth from \$log				
<pre>where \$filter and logid to int(logid)</pre>	gid) not in (4, 7, 14, 20)			
and nullifna(appcat group by	-			
appcat order by bandwidth desc				

Dataset Name	Description	Log Category
bandwidth-app-Top-Users-By- Bandwidth-Sessions	Bandwidth application top users by bandwidth usage	traffic
<pre>select coalesce(nullifna(`user`), nullifna(`unauthus ipstr(`srcip`)) as user_src,</pre>	ser`),	

```
sum(
    coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
  ) as bandwidth,
  sum(
    coalesce(rcvdbyte, 0)
  ) as traffic in,
  sum(
    coalesce(sentbyte, 0)
  ) as traffic out,
  count(*) as sessions
from
  $loq
where
  $filter
  and logid_to_int(logid) not in (4, 7, 14, 20)
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(times</pre>	tamp) as hodex,		
<pre>count(distinct(user_src) }</pre>			
) as total_user from			
<pre>###(select \$flex_timestamp as timestamp, coalesce(nullifna (`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src from \$log where \$filter and logid_to_int(logid) not in (4, 7, 14, 20) group by timestamp, user_src order by timestamp desc)### t group by hodex order by hodex</pre>			

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

```
select
  coalesce(
    nullifna(
      root domain (hostname)
    ),
    ipstr(`dstip`)
  ) as domain,
  sum(
    coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
  ) as bandwidth,
  sum(
    coalesce(rcvdbyte, 0)
  ) as traffic in,
  sum(
    coalesce(sentbyte, 0)
  ) as traffic out,
  count(*) as sessions
from
  $loq
where
  $filter
  and logid to int(logid) not in (4, 7, 14, 20)
group by
  domain
order by
  bandwidth desc
```

	_
Dataset I	Name

Description	١

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

```
select
 coalesce(
    cast(poluuid as text),
   cast (policyid as text)
  ) as polid,
  sum(
    coalesce(rcvdbyte, 0) + coalesce(sentbyte, 0)
  ) as bandwidth,
  sum(
    coalesce(rcvdbyte, 0)
  ) as traffic in,
```

```
sum(
    coalesce(sentbyte, 0)
  ) as traffic_out,
    count(*) as sessions
from
    $log
where
    $filter
    and logid_to_int(logid) not in (4, 7, 14, 20)
group by
    polid
order by
    bandwidth desc
```

Dataset Name	Description	Log Category
bandwidth-app-Traffic-Statistics	Bandwidth application traffic statistics	traffic
<pre>drop table if exists rpt_ 1(total_sessions van total_bandwidth van ave_session varchan ave_bandwidth vard active_date varchan total_users varchan total_app varchan total_dest varchan); insert into rpt_t total sessions, to</pre>	archar(255), ar(255), char(255), ar(255), ar(255), (255), c(255), cmptbl_1 (y table rpt_tmptbl_
ave_session, ave_k)	bandwidth	
<pre>select format_numeric_no_de sum(sessions)) as total_sessions, bandwidth_unit(sum(bandwidth)) as total_bandwidth format_numeric_no_de cast(sum(sessions)/ \$</pre>	а,)

```
)

) 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 logid to int(logid) not in (4, 7, 14, 20))### 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 logid to int(logid) not in (4, 7, 14, 20) 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 logid to int(logid) not in (4, 7, 14, 20) 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 logid to int(logid) not in (4, 7, 14, 20) 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 logid to int(logid) not in (4, 7, 14, 20) 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
scores from $slog$ where	s estamp as timestamp, sum(crsco \$filter and logid_to_int(logi	d) not in (4,
	is not null group by timestam r by timestamp desc)### t grou	
(crscore%65536)>0 order		np by hodex
(crscore%65536)>0 order order by hodex	r by timestamp desc)### t grou	p by hodex Log Category

###(select \$flex_timestamp as timestamp, sum(crscore%65536) as scores, count(*) as totalnum from \$log where \$filter and logid_to_ int(logid) not in (4, 7, 14, 20) 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(`unauthuse ipstr(`srcip`)) as user_src, sum(crscore % 65536) from</pre>		

```
$log
where
$filter
and logid_to_int(logid) not in (4, 7, 14, 20)
and crscore is not null
group by
user_src
having
sum(crscore % 65536)& gt; 0
order by
scores desc
```

Dataset Name	Description	Log Category
Top-Devices-By-Reputation-Scores	Reputation top devices by scores	traffic
<pre>select devtype, coalesce(nullifna(`srcname` nullifna(`srcmac`) ipstr(`srcip`)) as dev_src,</pre>	,	
sum(crscore % 65536) from \$log	as scores	
<pre>where \$filter and logid_to_int(log and crscore is not n</pre>	id) not in (4, 7, 14, 20) ull	
group by devtype, dev src		
having sum(crscore % 65536) order by scores desc	& gt; 0	

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 logid to int(logid) not in (4, 7, 14, 20) 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 logid to int(logid) not in (4, 7, 14, 20) 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, devtype, sum(crscore%65536) as sum rp score from \$log where \$pre period \$filter and logid to int (logid) not in (4, 7, 14, 20) and crscore is not null group by f device, devtype having sum(crscore%65536)>0 order by sum rp score desc) ###; create temporary table rpt tmptbl 2 as ###(select coalesce(nullifna(`srcname`),nullifna(`srcmac`), ipstr(`srcip`)) as f device, devtype, sum(crscore%65536) as sum rp score from \$log where \$filter and logid to int(logid) not in (4, 7, 14, 20) and crscore is not null group by f device, devtype having sum (crscore%65536)>0 order by sum rp score desc)###; select t1.f device, t1.devtype , sum(t1.sum rp score) as t1 sum score, sum (t2.sum rp score) as t2 sum score, (sum(t2.sum rp score)-sum (t1.sum rp score)) as delta from rpt tmptbl 1 as t1 inner join rpt tmptbl 2 as t2 on t1.f device=t2.f device and t1.devtype=t2.devtype where t2.sum rp score > t1.sum rp score group by tl.f device, tl.devtype order by delta desc

Dataset Name	Description	Log Category
Attacks-By-Severity	Threat attacks by severity	attack
select		
(
case when seve:	rity = 'critical' then 'Critic	cal' when severity
= 'high' then 'High	n' when severity = 'medium' th	nen 'Medium' when
severity = 'low' th	nen 'Low' when severity = 'in:	fo' then 'Info' end
) as severity,		
count(*) as total	lnum	
from		
\$log		
where		
\$filter		
group by		
severity		
order by totalnum desc		

Dataset Name	Description	Log Category
Top-Attacks-Detected	Threat top attacks detected	attack
<pre>select attack, attackid, cve, severity, sum(attack_count) as from ###(select attack, at severity = 'critical' t t1.severity = 'medium' else 5 end) as severity t1 left join (select na t1.attack=t2.name where group by attack, attack by severity_level, atta</pre>	<pre>attack_count tackid, t1.severity, cve, (case hen 1 when t1.severity = 'high' then 3 when t1.severity = 'low _level, count(*) as attack_coun me, cve, vuln_type from ips_mda \$filter and nullifna(attack) i id, t1.severity, severity_level ck_count desc)### t group by at</pre>	when t1 then 2 when ' then 4 t from \$log ta) t2 on s not null , cve order tack,
	<pre>ck_count desc)### t group by at erity_level, cve order by sever</pre>	

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

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

Dataset Name	Description	Log Category
Top-Virus-Source	Threat top virus source	traffic
<pre>select srcip, hostname, sum(totalnum) a from (###(select sr</pre>	<pre>s totalnum cip, hostname, count(*) as total</pre>	Lnum from \$log-
<pre>traffic where \$fi 20) and utmevent hostname order by ipstr(`dstip`) as where \$filter and (virus) is not nu</pre>	<pre>lter and logid_to_int(logid) not is not null and virus is not nul totalnum desc)### union all ### hostname, count(*) as totalnum (eventtype is null or logver>=5 ll group by srcip, hostname order by srcip, hostname order by tot</pre>	t in (4, 7, 14, Il group by srcip, \$ (select srcip , from \$log-virus 52) and nullifna er by totalnum

Dataset Name	Description	Log Category
Intrusion-in-Last-7-Days	Threat intrusion timeline	attack
<pre>select \$flex_timescale(times sum(totalnum) as tota from</pre>	-	
—	stamp as timestamp, count(*) as to group by timestamp order by time ex order by hodex	

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

select

```
$flex_datetime(timestamp) as hodex,
  sum(totalnum) as totalnum
from
```

. I ON (

###(select \$flex_timestamp as timestamp, count(*) as totalnum from \$log-traffic where \$filter and logid_to_int(logid) not in (4, 7, 14, 20) and utmevent is not null and virus is not null group by timestamp order by timestamp desc)### union all ###(select \$flex_ timestamp as timestamp, count(*) as totalnum from \$log-virus where \$filter and (eventype is null or logver>=52) and nullifna(virus) is not null group by timestamp order by timestamp desc)###) 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 t	otalnum	
from		
###(select coalesc	ce(nullifna(`user`), ipstr(`	srcip`)) as user_
<pre>src, virus, count(*)</pre>	as totalnum from \$log wher	re \$filter group by
user_src, virus orde	er by totalnum desc)### t wh	nere virus like
'Riskware%' group by	v user_src order by totalnum	n desc
Dataset Name	Description	Log Category

Dataset Name	Description	Log Category
Top-Spyware-by-Name	Threat top spyware by name	virus
<pre>select virus, max(virusid_s) as v: sum(totalnum) as tot</pre>		
from	califant	
<pre>src, virus, virusid_to (*) as totalnum from s virusid_s order by tot</pre>	(nullifna(`user`), ipstr(o_str(virusid, eventtype) \$log where \$filter group talnum desc)### t where w virus order by totalnum d	as virusid_s, count by user_src, virus, virus like

Dataset Name	Description	Log Category
Top-Spyware-Source	Threat top spyware source	traffic
<pre>select srcip, hostname, count(*) as total from \$log where \$filter and logid_to_int(and virus like 'R</pre>	logid) not in (4, 7, 14, 20)	
group by srcip, hostname order by totalnum desc		

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

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

from

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

Dataset Name	Description	Log Category
Top-Adware-Victims	Threat top adware victims	virus
<pre>select user_src, sum(totalnum) as to</pre>	otalnum	
<pre>from ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_ src, virus, count(*) as totalnum from \$log where \$filter group by</pre>		
—	t by totalnum desc)### t wi ser_src order by totalnum o	

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

select

```
virus,
max(virusid_s) as virusid,
sum(totalnum) as totalnum
```

from

###(select 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 order by totalnum desc)### 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
and virus like group by srcip, hostname	t(logid) not in (4, 7, 14, 20)	
order by totalnum desc		
Dataset Name	Description	Log Category
Adware-Time-Line	Threat adware timeline	virus
<pre>sum(totalnum) a from ###(select \$fle</pre>	(timestamp) as hodex, s totalnum x_timestamp as timestamp, count filter and virus like 'Adware%	

timestamp order by timestamp desc)### t group by hodex order by hodex

Dataset Name	Description	Log Category
Intrusions-Timeline-By-Severity	Threat intrusions timeline by severity	attack
<pre>sum(critical) as cr: sum(high) as high, sum(medium) as media sum(low) as low, sum(info) as info from ###(select \$flex_tin = 'critical' then 1 efficient ity = 'high' then 1 efficient 'medium' then 1 else ('notice' then 1 else ()</pre>	um, mestamp as timestamp, sum(cas lse 0 end) as critical, sum(c lse 0 end) as high, sum(case 0 end) as medium, sum(case wh 0 end) as low, sum(case when 'debug' then 1 else 0 end) as y timestamp order by timestam	<pre>case when sever- when severity = nen severity = severity = s info from \$log</pre>
Dataset Name	Description	Log Category
Important-Intrusions-Timeline-By- Severity	Threat intrusions timeline by severity	attack
<pre>select \$flex_timescale(time sum(critical) as cri sum(high) as high, sum(medium) as mediu sum(low) as low, sum(info) as info</pre>		

from

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

Dataset Name	Description	Log Category
Top-Intrusions-By-Types	Threat top intrusions by types	attack
<pre>select vuln_type, count(*) as totaln</pre>	um	
from		
\$log t1		
left join (
select		
name,		
cve,		
vuln_type		
from		
ips mdata		
) t2 on t1.attack	= t2.name	
where		
\$filter		
and vuln type is n	ot null	
group by		
vuln_type		
order by		
totalnum desc		

Dataset Name	Description	Log Category
Critical-Severity-Intrusions	Threat critical severity intrusions	attack
<pre>select attack, attackid, cve, vuln_type, count(*) as totalnum from \$log t1 left join (select</pre>		
name,		
cve,		

```
vuln_type
from
    ips_mdata
) t2 on t1.attack = t2.name
where
    $filter
    and t1.severity = 'critical'
    and nullifna(attack) is not null
group by
    attack,
    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 totalnu	am	
from		
\$log t1		
left join (
select		
name,		
cve,		
vuln_type		
from		
ips_mdata		
) t2 on t1.attack =	= t2.name	
where		
\$filter		
and t1.severity =	'high'	
and nullifna(attac)	k) is not null	
group by		
attack,		
attackid,		

vuln_type, cve order by totalnum desc

Dataset Name	Description	Log Category
Medium-Severity-Intrusions	Threat medium severity intrusions	attack
<pre>select attack, vuln_type, cve, count(*) as total from \$log t1 left join (select name, cve, vuln_type from ips_mdata) t2 on t1.attack where \$filter and t1.severity = and nullifna(atta group by attack, vuln_type, cve order by totalnum desc</pre>	= t2.name 'medium'	
Dataset Name	Description	Log Category
Top-Intrusion-Victims	Threat top intrusion victims	attack
<pre>select victim, sum(cri_num) as c sum(high_num) as sum(med_num) as m</pre>	high,	

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

Dataset Name	Description	Log Category
Top-Intrusion-Sources	Threat top intrusion sources	attack
from ###(select srcip as then 1 else 0 end) as 1 else 0 end) as high else 0 end) as med_nu	igh,	everity='critical' severity='high' then city='medium' then 1 c and severity in

Dataset Name	Description	Log Category
Top-Blocked-Intrusions	Threat top blocked intrusions	attack
<pre>severity = 'high' t 'Medium' when t1.se 'info' then 'Info') as severity_nam count(*) as total vuln_type, (</pre>	num,	= 'medium' then en t1.severity =
case when tl.se	everity = 'critical' then 0 w	hen tl.severity =

```
'high' then 1 when t1.severity = 'medium' then 2 when t1.severity
= 'low' then 3 when t1.severity = 'info' then 4 else 5 end
  ) as severity number
from
  $log t1
  left join (
    select
      name,
      cve,
      vuln type
    from
      ips mdata
  ) t2 on t1.attack = t2.name
where
  $filter
  and nullifna(attack) is not null
  and action not in ('detected', 'pass session')
group by
  attack,
  attackid,
  t1.severity,
  vuln type
order by
  severity number,
  totalnum desc
Dataset Name
                          Description
                                                              Log Category
Top-Monitored-Intrusions
                          Threat top monitored intrusions
                                                              attack
select
  attack,
  attackid,
  (
    case when t1.severity = 'critical' then 'Critical' when t1.-
severity = 'high' then 'High' when t1.severity = 'medium' then
'Medium' when t1.severity = 'low' then 'Low' when t1.severity =
'info' then 'Info' end
  ) as severity name,
  count(*) as totalnum,
  vuln type,
  (
```

```
case when t1.severity = 'critical' then 0 when t1.severity =
```

```
'high' then 1 when t1.severity = 'medium' then 2 when t1.severity
= 'low' then 3 when t1.severity = 'info' then 4 else 5 end
  ) as severity number
from
  $log t1
  left join (
    select
      name,
      cve,
      vuln type
    from
      ips mdata
  ) t2 on t1.attack = t2.name
where
  $filter
  and nullifna(attack) is not null
  and action in ('detected', 'pass session')
group by
  attack,
  attackid,
  t1.severity,
  vuln type
order by
  severity number,
  totalnum desc
Dataset Name
                         Description
                                                             Log Category
Attacks-Over-HTTP-HTTPs
                         Threat attacks over HTTP HTTPs
                                                             attack
select
  attack,
  attackid,
  (
    case when severity = 'critical' then 'Critical' when severity
= 'high' then 'High' when severity = 'medium' then 'Medium' when
severity = 'low' then 'Low' when severity = 'info' then 'Info' end
  ) as severity,
  count(*) as totalnum,
  (
    case when severity = 'critical' then 0 when severity = 'high'
then 1 when severity = 'medium' then 2 when severity = 'low' then
3 when severity = 'info' then 4 else 5 end
```
```
) as severity_number
from
  $log
where
  $filter
  and severity in ('critical', 'high', 'medium')
  and upper(service) in ('HTTP', 'HTTPS')
group by
  attack,
  attackid,
  severity,
  severity,
  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>select (case apstatus when 3 then 'suppressed' else) as ap_full_status, count(*) as totalnum from (select apstatus, bssid, ssid from</pre>	1 then 'rogue' when 2 then 'accep e 'others' end	ted' when
\$log where \$filter and	us, bssid, ssid, count(*) as subt apstatus is not null and apstatus	!=0 and

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

Dataset Name	Description	Log Category
default-AP-Detection-Summary-by- Status-OffWire_table	Default access point detection summary by status off- wire	event
<pre>select (case apstatus when 3 then 'suppressed' els) as ap_full_status, count(*) as totalnum from (select apstatus, bssid,</pre>	1 then 'rogue' when 2 then 'accep e 'others' end	ted' when
<pre>\$log where \$filter and bssid is not null and o (43527, 43521, 43525, 4 43571, 43582, 43583, 43 order by subtotal desc)</pre>	us, bssid, ssid, count(*) as subt apstatus is not null and apstatus nwire='no' and logid_to_int(logid 3563, 43564, 43565, 43566, 43569, 584, 43585) group by apstatus, bs ### t group by apstatus, bssid, s order by totalnum desc	!=0 and) in 43570, sid, ssid

Dataset Name	Description	Log Category
default-AP-Detection-Summary-by- Status-OnWire	Default access point detection summary by status on- wire	event
<pre>3 then 'suppressed' els) as ap_full_status, count(*) as totalnum from (select apstatus,</pre>	1 then 'rogue' when 2 then 'accep e 'others' end	ted' when
apstatus, bssid, ssid from		

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

Dataset Name	Description	Log Category
default-AP-Detection-Summary-by- Status-OnWire_table	Default access point detection summary by status on- wire	event
<pre>3 then 'suppressed' els) as ap_full_status, count(*) as totalnum from (select apstatus, bssid, ssid</pre>	1 then 'rogue' when 2 then 'accer e 'others' end	pted' when
<pre>\$log where \$filter and bssid is not null and c (43527, 43521, 43525, 4 43571, 43582, 43583, 43 order by subtotal desc)</pre>	<pre>hus, bssid, ssid, count(*) as subt apstatus is not null and apstatus nwire='yes' and logid_to_int(log: 3563, 43564, 43565, 43566, 43569, 584, 43585) group by apstatus, bs ### t group by apstatus, bssid, s order by totalnum desc</pre>	s!=0 and id) in , 43570, ssid, ssid

Dataset Name	Description	Log Category
default-Managed-AP-Summary	Default managed access point summary	event
	n%' logid) in (43522, 43551) 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
	n%' logid) in (43522, 43551) else 'Unauthorized' end	
<pre>where \$filter and logid_to_int(logi group by ap_status order by totalnum desc</pre>	d) in (43522, 43551)	

Dataset Name	Description	Log Category
default-Unclassified-AP-Summary	Default unclassified access point summary	event
<pre>select (case onwire when 'n wire' else 'others' end) as ap_status, count(*) as totalnum</pre>	o' then 'off-wire' when 'yes' the	en 'on-

###(select onwire, ssid, bssid, count(*) as subtotal from \$log
where \$filter and apstatus=0 and bssid is not null and logid_to_
int(logid) in (43521, 43525, 43527, 43563, 43564, 43565, 43566,
43569, 43570, 43571, 43582, 43583, 43584, 43585) group by onwire,
ssid, bssid order by subtotal desc)### t group by ap_status order
by totalnum desc

Dataset Name	Description	Log Category
default-Unclassified-AP-Summary_ table	Default unclassified access point summary	event
<pre>wire' else 'others' en) as ap_status, count(*) as totalnum from ###(select onwire, s where \$filter and apst int(logid) in (43521, 43569, 43570, 43571, 4</pre>		al from \$log nd logid_to_ 565, 43566, up by onwire,
Dataset Name	Description	Log Category
default-selected-AP-Details-OffWire	Default selected access point details off-wire	event
select		

```
(
```

case apstatus when 0 then 'unclassified' when 1 then 'rogue'
when 2 then 'accepted' when 3 then 'suppressed' else 'others' end
) as ap_full_status,
 devid,
 vd,
 ssid,
 bssid,
 manuf,
 rssi,
 channel,
 radioband,

```
from dtime(
    min(dtime)
  ) as first seen,
  from dtime(
    max(dtime)
  ) as last seen,
  detectionmethod,
  itime,
  onwire as on wire
from
  $log
where
  $filter
  and apstatus is not null
  and bssid is not null
  and onwire = 'no'
  and logid to int(logid) in (
    43521, 43563, 43564, 43565, 43566, 43569,
    43570, 43571
  )
group by
  ap full status,
  devid,
  vd,
  ssid,
  bssid,
  manuf,
  rssi,
  channel,
  radioband,
  detectionmethod,
  itime,
  onwire,
  apstatus
Dataset Name
                          Description
```

```
      Dataset Name
      Description
      Log Category

      default-selected-AP-Details-OnWire
      Default selected access point details on-wire
      event

      select
      (
      case apstatus when 0 then 'unclassified' when 1 then 'rogue'

      when 2 then 'accepted' when 3 then 'suppressed' else 'others' end
```

```
) as ap full status,
  devid,
  vd,
  ssid,
  bssid,
  manuf,
  rssi,
  channel,
  radioband,
  from dtime(
    min(dtime)
  ) as first seen,
  from dtime(
    max(dtime)
  ) as last seen,
  detectionmethod,
  itime,
  onwire as on wire
from
  $log
where
  $filter
  and apstatus is not null
  and bssid is not null
  and onwire = 'yes'
  and logid to int(logid) in (
    43521, 43563, 43564, 43565, 43566, 43569,
    43570, 43571
  )
group by
  ap full status,
  devid,
  vd,
  ssid,
  bssid,
  manuf,
  rssi,
  channel,
  radioband,
  detectionmethod,
  itime,
```

onwire, apstatus

Dataset Name	Description	Log Category
event-Wireless-Client-Details	Event wireless client details	event
<pre>drop table if exists rpt_t; 1 as</pre>	mptbl_1; create tempo:	rary table rpt_tmptbl_
select		
ip, lower(mac) as lmac,		
sn,		
ssid,		
channel,		
radioband,		
min(dtime) as first,		
<pre>max(dtime) as last from</pre>		
\$log - event		
where		
\$filter		
and ip is not null		
and mac is not null		
and sn is not null		
and ssid is not null		
group by		
ip, lmac,		
sn,		
ssid,		
channel,		
radioband		
order by		
ip;		
select		
user_src, ip,		
lmac,		
sn,		
ssid,		
channel,		

```
radioband,
  from dtime(first) as first seen,
  from dtime(last) as last seen,
  cast(
    volume as decimal(18, 2)
  ) as bandwidth
from
  (
    select
      *
    from
      rpt tmptbl 1
      inner join (
        select
          user src,
          srcip,
          sum(volume) as volume
        from
          ###(select coalesce(nullifna(`user`), nullifna(`un-
authuser`), ipstr(`srcip`)) as user src, srcip, sum(coalesce(sent-
byte, 0)+coalesce(rcvdbyte, 0)) as volume from $log-traffic where
$filter-time and logid to int(logid) not in (4, 7, 14) and srcip
is not null group by user src, srcip having sum(coalesce(sentbyte,
0)+coalesce(rcvdbyte, 0))>0 order by volume desc)### t group by
user src, srcip order by user src, srcip) t on rpt tmptbl 1.ip =
t.srcip) t order by volume desc
```

Dataset Name	Description	Log Category
event-Wireless-Accepted-Offwire	Event wireless accepted off-wire	event
select		
'accepted' as ap_ful?	l_status,	
devid,		
vd,		
ssid,		
bssid,		
manuf,		
channel,		
radioband,		
from_dtime(
<pre>max(last_seen)</pre>		
) as last_seen,		

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

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

Dataset Name	Description	Log Category
event-Wireless-Accepted-Onwire	Event wireless accepted on-wire	event
select		
'accepted' as ap full	status.	
devid,	,	
vd,		
ssid,		
bssid,		
manuf,		
channel,		
radioband,		
from_dtime(
<pre>max(last_seen)</pre>		
) as last_seen,		
detectionmethod,		
snclosest,		
'yes' as on_wire		
from		
	ssid, bssid, manuf, channe	
	est, onwire, apstatus, max(_
_	ilter and bssid is not null	
	3525, 43563, 43564, 43565,	
	devid, vd, ssid, bssid, man	
	hod, snclosest, onwire, aps	-
last_seen desc)### t wh	ere apstatus=2 and onwire='	yes' group by

devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest order by last seen desc

event-Wireless-Rogue-Offwire Event wireless rogue off-wire event select 'rogue' as ap_full_status, devid,
'rogue' as ap_full_status,
devid,
vd,
ssid,
bssid,
manuf,
channel,
radioband,
from_dtime(
max(last_seen)
) as last_seen,
detectionmethod,
snclosest,
'no' as on_wire
from
<pre>###(select devid, vd, ssid, bssid, manuf, channel, radioband,</pre>
detectionmethod, snclosest, onwire, logid, apstatus, max(dtime)
last_seen from \$log where \$filter and bssid is not null and logi
to_int(logid) in (43521, 43525, 43563, 43564, 43565, 43566, 4356 43570, 43571) group by devid, vd, ssid, bssid, manuf, channel,

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</pre>	_status,	
devid, vd, ssid,		
bssid, manuf,		
channel, radioband, from_dtime(

```
max(last_seen)
) as last_seen,
detectionmethod,
snclosest,
'yes' as on_wire
com
```

###(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 fu	ll status,	
devid,	—	
vd,		
ssid,		
bssid,		
manuf,		
channel,		
radioband,		
from_dtime(
<pre>max(last_seen)</pre>		
) as last_seen,		
detectionmethod,		
snclosest,		
'no' as on_wire		
from		
	ssid, bssid, manuf, cha	
detectionmethod, snclos		

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

Dataset Name	Description	Log Category
event-Wireless-Suppressed-Onwire	Event wireless suppressed on-wire	event
select		
'suppressed' as ap_fu	ll_status,	
devid,		
vd,		
ssid,		
bssid,		
manuf,		
channel,		
radioband,		
from_dtime(
<pre>max(last_seen)</pre>		
) as last_seen,		
detectionmethod,		
snclosest,		
'yes' as on_wire		
from		
###(select devid, vd,	ssid, bssid, manuf, cha	annel, radioband,
detectionmethod, snclos	est, onwire, apstatus, m	nax(dtime) as last_

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_: devid, vd, ssid, bssid, manuf,</pre>	full_status,	

```
channel,
radioband,
from_dtime(
    max(last_seen)
) as last_seen,
detectionmethod,
snclosest,
'no' as on_wire
```

```
from
```

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

Dataset Name	Description	Log Category
event-Wireless-Unclassified-Onwire	Event wireless unclassified on-wire	event
select		
'unclassified' as ap_	full status.	
devid,		
vd,		
ssid,		
bssid,		
manuf,		
channel,		
radioband,		
from_dtime(
<pre>max(last_seen)</pre>		
) as last_seen,		
detectionmethod,		
snclosest,		
'yes' as on_wire		
from		
###(select devid, vd,	ssid, bssid, manuf, channel, rad	ioband,
detectionmethod, snclos	est, onwire, apstatus, max(dtime)	as last_
_	ilter and bssid is not null and 1 3525, 43563, 43564, 43565, 43566,	

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
```

###(select devid, vd, nullifna(`xauthuser`) as xauthuser_ agg, nullifna(`user`) as user_agg, remip, tunnelid, 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 'ipsec%' and not (tunnelip is null or tunnelip='0.0.0.0') and action in ('tunnelstats', 'tunnel-down', 'tunnel-up') and tunnelid is not null and tunnelid!=0 group by devid, vd, xauthuser_agg, user_agg, remip, 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>select remip as remote_ip, sum(bandwidth) as bar from (select devid, vd, remip, tunnelid, /</pre>	ndwidth	
<pre>in) else max(max_traffi) as traffic_in, (case when min(s)</pre>	<pre>s_time) = max(e_time) then max(max_ c_in) - min(min_traffic_in) end s_time) = max(e_time) then max(max_ fic_out) - min(min_traffic_out) end</pre>	

```
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, tunnelid, 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 'ssl%' and action in ('tunnel-stats', 'tunnel-down') and remip is not null and tunnelid is not null group by devid, vd, remip, tunnelid order by tunnelid)### t group by devid, vd, remip, tunnelid) tt group by remote_ip having sum(traffic_in+traffic_out)>0 order by bandwidth desc

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

select

```
$flex_timescale(timestamp) as hodex,
  sum(allowed_request) as allowed_request,
  sum(blocked_request) as blocked_request
from
```

(

###(select \$flex_timestamp as timestamp, sum(case when utmaction!='blocked' then 1 else 0 end) as allowed_request, sum(case when utmaction='blocked' then 1 else 0 end) as blocked_request from \$log-traffic where \$filter and logid_to_int(logid) not in (4, 7, 14, 20) and utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') group by timestamp order by timestamp desc)### union all ###(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-webfilter where \$filter and (eventtype is null or logver>=52) group by timestamp order by timestamp desc)###) t group by hodex order by hodex

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

```
select
$flex_timescale(timestamp) as hodex,
cast(
    ebtr_value(
        ebtr_agg_flat(browsetime),
        null,
        $timespan
    )/ 60.0 as decimal(18, 2)
) as browsetime
```

```
from
```

###(select \$flex_timestamp as timestamp, ebtr_agg_flat(\$browse_ time) as browsetime from \$log where \$filter and logid_to_int (logid) not in (4, 7, 14, 20) and \$browse_time is not null group by timestamp order by timestamp desc)### t group by hodex order by hodex

Dataset Name	Description	Log Category
traffic-Browsing-Time-Summary- Enhanced	Traffic browsing time summary enhanced	traffic
time) as browsetime f (logid) not in (4, 7,	rowsetime),	_to_int null group
Dataset Name	Description	Log Category
webfilter-Top-Web-Users-By-Blockec Requests	- Webfilter top web users by blocked requests	webfilter

select

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

(

###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, count(*) as requests from \$logtraffic where \$filter and logid_to_int(logid) not in (4, 7, 14, 20) and utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') and coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) is not null and utmaction='blocked' group by user_src order by requests desc)### union all ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_ src, count(*) as requests from \$log-webfilter where \$filter and (eventtype is null or logver>=52) and coalesce(nullifna(`user`), ipstr(`srcip`)) is not null and action='blocked' group by user_src order by requests desc)###) t group by user_src order by requests desc

Dataset Name	Description	Log Category
webfilter-Top-Web-Users-By-Allowed- Requests	Webfilter top web users by allowed requests	webfilter
<pre>ipstr(`srcip`)) as user traffic where \$filter a 20) and utmevent in ('w 'command-block', 'scrip nullifna(`unauthuser`), tion!='blocked' group b all ###(select coalesce src, count(*) as reques (eventtype is null or l ipstr(`srcip`)) is not</pre>	<pre>ests (nullifna(`user`), nullifna(`u _src, count(*) as requests fro nd logid_to_int(logid) not in ebfilter', 'banned-word', 'web t-filter') and coalesce(nullif ipstr(`srcip`)) is not null a y user_src order by requests d (nullifna(`user`), ipstr(`srci ts from \$log-webfilter where \$ ogver>=52) and coalesce(nullif null and action!='blocked' gro esc)###) t group by user_src or </pre>	om \$log- (4, 7, 14, o-content', and utmac- desc)### union p`)) as user_ Sfilter and Ena(`user`), oup by user_

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

```
select
user_src,
ebtr_value(
    ebtr_agg_flat(browsetime),
    null,
    $timespan
) as browsetime,
    sum(bandwidth) as bandwidth,
    sum(traffic_in) as traffic_in,
    sum(traffic_out) as traffic_out
from
```

###(select user_src, ebtr_agg_flat(browsetime) as browsetime, sum(bandwidth) as bandwidth, sum(traffic_in) as traffic_in, sum (traffic_out) as traffic_out from (select coalesce(nullifna (`user`), ipstr(`srcip`)) as user_src, ebtr_agg_flat(\$browse_time) as browsetime, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce (sentbyte, 0)) as traffic_out from \$log where \$filter and \$browse_ time is not null group by user_src) t group by user_src 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
webfilter-Top-Blocked-Web-Sites-By-
                           Webfilter top blocked web sites by requests
                                                                 webfilter
Requests
select
  domain,
  catdesc,
  sum(requests) as requests
from
  (
    ###(select hostname as domain, catdesc, count(*) as requests
from $log-traffic where $filter and logid to int(logid) not in (4,
7, 14, 20) and utmevent in ('webfilter', 'banned-word', 'web-con-
tent', 'command-block', 'script-filter') and hostname is not null
and utmaction='blocked' group by domain, catdesc order by requests
```

desc)### union all ###(select hostname as domain, catdesc, count
(*) as requests from \$log-webfilter where \$filter and (eventtype
is null or logver>=52) and hostname is not null and catdesc is not
null and action='blocked' group by domain, catdesc order by

requests desc)###) t group by domain, catdesc order by requests
desc

Dataset Name	Description	Log Category
webfilter-Top-Allowed-Web-Sites-By- Requests	Webfilter top allowed web sites by requests	webfilter
select domain,		
•	atdesc, ', ') as agg_catdesc,	

```
sum(requests) as requests
```

```
from (
```

###(select hostname as domain, catdesc, count(*) as requests from \$log-traffic where \$filter and logid_to_int(logid) not in (4, 7, 14, 20) and utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') and hostname is not null and utmaction!='blocked' group by domain, catdesc order by requests desc)### union all ###(select hostname as domain, catdesc, count(*) as requests from \$log-webfilter where \$filter and (eventtype is null or logver>=52) and hostname is not null and catdesc is not null and action!='blocked' group by domain, catdesc order by requests desc)###) t group by domain order by requests desc

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

```
select
```

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

```
from
```

###(select coalesce(nullifna(root_domain(hostname)), 'other') as domain, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out from \$log-traffic where \$filter and logid_to_int(logid) not in (4, 7, 14, 20) and (countweb>0 or ((logver is null or logver<52) and (hostname is not null or utmevent in ('webfilter', 'banned-word', 'web-content', 'commandblock', 'script-filter')))) and catdesc in ('Streaming Media and Download') 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
webfilter-Top-Blocked-Web-Categories	Webfilter top blocked web categories	webfilter
select		
catdesc,		

```
sum(requests) as requests
```

```
from (
```

###(select catdesc, count(*) as requests from \$log-traffic where \$filter and logid_to_int(logid) not in (4, 7, 14, 20) and utmevent in ('webfilter', 'banned-word', 'web-content', 'commandblock', 'script-filter') and catdesc is not null and utmaction='blocked' group by catdesc order by requests desc)### union all ###(select catdesc, count(*) as requests from \$log-webfilter where \$filter and (eventtype is null or logver>=52) and catdesc is not null and action='blocked' group by catdesc order by requests desc)###) t group by catdesc order by requests desc

Dataset Name	Description	Log Category
webfilter-Top-Allowed-Web-Categories	Webfilter top allowed web categories	webfilter
select		
catdesc,		
sum(requests) as reque	ests	
from		
(
###(select catdesc,	<pre>count(*) as requests from \$</pre>	log-traffic
where \$filter and logid	_to_int(logid) not in (4, 7,	14, 20) and
utmevent in ('webfilter	', 'banned-word', 'web-conte	ent', 'command-
block', 'script-filter') and catdesc is not null an	nd utmac-
tion!='blocked' group by	y catdesc order by requests	desc)### union
<pre>all ###(select catdesc,</pre>	<pre>count(*) as requests from \$</pre>	log-webfilter
where \$filter and (even	ttype is null or logver>=52)	and catdesc is
not null and action!='b	locked' group by catdesc ord	ler by requests
desc) ###) t group by ca	tdesc order by requests desc	

Dataset Name	Description	Log Category
traffic-Top-50-Sites-By-Browsing-Time	Traffic top sites by browsing time	traffic
select		
hostname,		
string_agg(distinct c	atdesc, ', ') as agg_catdesc,	
ebtr_value(_	
ebtr_agg_flat(brows	etime),	
null,		
\$timespan		
) as browsetime,		
sum(bandwidth) as ban		
<pre>sum(traffic_in) as tr</pre>	—	
<pre>sum(traffic_out) as t</pre>	raffic_out	
from		
	catdesc, ebtr_agg_flat(browsetime	
	as bandwidth, sum(traffic_in) as	—
—	traffic_out from (select hostnar	
	<pre>owse_time) as browsetime, sum(coa yte, 0)) as bandwidth, sum(coales</pre>	
	ic in, sum(coalesce(sentbyte, 0))	
	here \$filter and logid to int(log	
—	name is not null and \$browse time	
	catdesc) t group by hostname, ca	
	<pre>r_agg_flat(browsetime), null, nul</pre>	

-					_ `			•	
desc)###	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>select hostname, string_agg(distinct c ebtr_value(ebtr_agg_flat(brows) null, \$timespan) as browsetime, sum(bandwidth) as ban sum(traffic_in) as tra- sum(traffic_out) as t</pre>	dwidth, affic_in,	

###(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 logid_to_int(logid) not in (4, 7, 14, 20) and hostname is not null and \$browse_time is not null group by hostname, catdesc) t group by hostname, catdesc order by ebtr_value(ebtr_agg_flat(browsetime), null, null) desc)### t group by hostname order by browsetime desc

Dataset Name	Description	Log Category
traffic-Top-10-Categories-By-Brows Time	sing- Traffic top category by browsing time	traffic
<pre>(bandwidth) as bandw (\$browse_time) as br (rcvdbyte, 0)) as ba int(logid) not in (4 \$browse_time is not order by ebtr_value(</pre>		otr_agg_flat te, 0)+coalesce er and logid_to_ not null and up by catdesc all, null)
Dataset Name	Description	Log Category
traffic-Top-10-Categories-By-Brows Time-Enhanced	sing- Traffic top category by browsing time enhanc	ed traffic
<pre>select catdesc, ebtr_value(</pre>		

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

```
###(select catdesc, ebtr_agg_flat(browsetime) as browsetime, sum
(bandwidth) as bandwidth from (select catdesc, ebtr_agg_flat
($browse_time) as browsetime, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth from $log where $filter and logid_to_
int(logid) not in (4, 7, 14, 20) and catdesc is not null and
$browse_time is not null group by catdesc) t group by catdesc
order by ebtr_value(ebtr_agg_flat(browsetime), null, null)
desc)### t group by catdesc order by browsetime desc
```

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 logid_to_int(logid) not in (4, 7, 14, 20) and \$browse_time is not null group by dstcountry) t group by dstcountry order by ebtr_value(ebtr_agg_flat(browsetime), null, null) desc)### 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
<pre>sum(bandwidth) as bandw (traffic_out) as traffic flat(\$browse_time) as k coalesce(rcvdbyte, 0)) traffic_in, sum(coalesc where \$filter and logic \$browse_time is not nut country order by ebtr_w</pre>	ndwidth, raffic_in,	ic_in, sum ebtr_agg_ te, 0)+- dbyte, 0)) as from \$log 4, 20) and by dst- , null, null)
Dataset Name	Description	Log Category

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

1	۰.		
L	4	ч	l
۰.	,		'

Dataset Name	Description	Log Category
Top-10-Users-Browsing-Time	Estimated browsing time	traffic
select		
user_src,		
ebtr_value(ebtr agg flat(bro	ousetime)	
null,		
\$timespan		
) as browsetime		
from		
	c, ebtr_agg_flat(browsetime) a e(nullifna(`user`), nullifna(`	
	ser src, ebtr agg flat(\$browse	
		—
setime from \$loq whe:	re Sillter and logid to int(lo	(4, 1)
	re \$filter and logid_to_int(lc se time is not null group by u	
7, 14, 20) and \$brows		ser_src) t grou
7, 14, 20) and \$brows by user_src order by	se_time is not null group by u	ser_src) t group setime), null,
7, 14, 20) and \$brows by user_src order by	se_time is not null group by u ebtr_value(ebtr_agg_flat(brow	ser_src) t group setime), null,
7, 14, 20) and \$brows by user_src order by null) desc)### t grou	se_time is not null group by u ebtr_value(ebtr_agg_flat(brow up by user_src order by browse	ser_src) t group setime), null, time desc
7, 14, 20) and \$brows by user_src order by null) desc)### t grow Dataset Name Top-10-Users-Browsing-Time-	se_time is not null group by u ebtr_value(ebtr_agg_flat(brow up by user_src order by browse Description	ser_src) t group setime), null, time desc Log Category
7, 14, 20) and \$brows by user_src order by null) desc)### t grou Dataset Name Top-10-Users-Browsing-Time- Enhanced	se_time is not null group by u ebtr_value(ebtr_agg_flat(brow up by user_src order by browse Description	ser_src) t group setime), null, time desc Log Category
7, 14, 20) and \$brows by user_src order by null) desc)### t grow Dataset Name Top-10-Users-Browsing-Time- Enhanced select user_src, ebtr_value(se_time is not null group by u ebtr_value(ebtr_agg_flat(brow up by user_src order by browse Description Estimated browsing time enhanced	ser_src) t group setime), null, time desc Log Category
7, 14, 20) and \$brows by user_src order by null) desc)### t grow Dataset Name Top-10-Users-Browsing-Time- Enhanced select user_src, ebtr_value(ebtr_agg_flat(brows)	se_time is not null group by u ebtr_value(ebtr_agg_flat(brow up by user_src order by browse Description Estimated browsing time enhanced	ser_src) t group setime), null, time desc Log Category
<pre>7, 14, 20) and \$brows by user_src order by null) desc)### t grow Dataset Name Top-10-Users-Browsing-Time- Enhanced select user_src, ebtr_value(ebtr_agg_flat(brownull,</pre>	se_time is not null group by u ebtr_value(ebtr_agg_flat(brow up by user_src order by browse Description Estimated browsing time enhanced	ser_src) t group setime), null, time desc Log Category
<pre>7, 14, 20) and \$brows by user_src order by null) desc)### t grow Dataset Name Top-10-Users-Browsing-Time- Enhanced select user_src, ebtr_value(ebtr_agg_flat(brownull, \$timespan</pre>	se_time is not null group by u ebtr_value(ebtr_agg_flat(brow up by user_src order by browse Description Estimated browsing time enhanced	ser_src) t group setime), null, time desc Log Category
<pre>7, 14, 20) and \$brows by user_src order by null) desc)### t grow Dataset Name Top-10-Users-Browsing-Time- Enhanced select user_src, ebtr_value(ebtr_agg_flat(brownull,</pre>	se_time is not null group by u ebtr_value(ebtr_agg_flat(brow up by user_src order by browse Description Estimated browsing time enhanced	ser_src) t groug setime), null, time desc Log Category
<pre>7, 14, 20) and \$brows by user_src order by null) desc)### t grow Dataset Name Top-10-Users-Browsing-Time- Enhanced select user_src, ebtr_value(ebtr_agg_flat(brownull, \$timespan) as browsetime from</pre>	se_time is not null group by u ebtr_value(ebtr_agg_flat(brow up by user_src order by browse Description Estimated browsing time enhanced	ser_src) t grou setime), null, time desc Log Category traffic

ipstr(`srcip`)) as user_src, ebtr_agg_flat(\$browse_time) as browsetime from \$log where \$filter and logid_to_int(logid) not in (4, 7, 14, 20) and \$browse_time is not null group by user_src) t group by user_src order by ebtr_value(ebtr_agg_flat(browsetime), null, null) desc)### t group by user src order by browsetime desc

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

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

###(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 logid_to_int(logid) not in (4, 7, 14, 20) 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-Enhanced	Estimated browsing time enhanced	traffic
from (select coalesce(r ipstr(`srcip`)) as user setime from \$log where 7, 14, 20) and \$browse_ by user_src order by eb	<pre>setime), ebtr_agg_flat(browsetime) nullifna(`user`), nullifna c_src, ebtr_agg_flat(\$brow \$filter and logid_to_int(_time is not null group by otr_value(ebtr_agg_flat(brown)); by user_src order by brow</pre>	(`unauthuser`), se_time) as brow- logid) not in (4, user_src) t group owsetime), null,
Dataset Name	Description	Log Category
wifi-Top-AP-By-Bandwidth	Top access point by bandwidth usage	traffic
<pre>select coalesce(ap, srcintf) sum(coalesce(sentbyte,</pre>	<pre>as ap_srcintf, 0)+ coalesce(rcvdbyte, 0)</pre>	

100

```
) as bandwidth
from
  $log
where
  $filter
  and logid to int(logid) not in (4, 7, 14, 20)
  and (
    srcssid is not null
    or dstssid is not null
  )
group by
  ap srcintf
having
  sum(
    coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
  ) & qt; 0
order by
  bandwidth desc
```

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

select

ap_srcintf as srcintf, count(distinct srcmac) as totalnum

from

###(select coalesce(ap, srcintf) as ap_srcintf, srcssid, osname, osversion, devtype, srcmac, count(*) as subtotal from \$log where \$filter and logid_to_int(logid) not in (4, 7, 14, 20) and (srcssid is not null or dstssid is not null) and srcmac is not null group by ap_srcintf, srcssid, osname, osversion, devtype, srcmac order by subtotal desc)### t group by srcintf order by totalnum desc

Dataset Name	Description	Log Category
wifi-Top-SSID-By-Bandwidth	Top SSIDs by bandwidth usage	traffic
<pre>select srcssid, sum(coalesce(sentbyte,) as bandwidth from</pre>	0)+ coalesce(rcvdbyte,	0)

```
$log
where
  $filter
  and logid_to_int(logid) not in (4, 7, 14, 20)
  and srcssid is not null
group by
  srcssid
having
  sum(
     coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
  )& gt; 0
order by
  bandwidth desc
```

Dataset Name	Description	Log Category
wifi-Top-SSID-By-Client	Top SSIDs by client	traffic
<pre>count(*) as subtotal fr (logid) not in (4, 7, 1 is not null) and srcmac osname, osversion, devt</pre>) as totalnum rcssid, osname, osversion, de om \$log where \$filter and log 4, 20) and (srcssid is not nu is not null group by srcintf ype, srcmac order by subtotal ll group by srcssid order by	gid_to_int all or dstssid , srcssid, . desc)### t

p WiFi applications by bandwidth usage	traffic
+ coalesce(rcvdbyte, 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 (
type, 'unknown') case when osve	, srcmac, 'unknown') ' (' ', ' coalesce(osname, '') rsion is null then '' else '	(
end) ')'		
) as client,		
sum (
	e, 0)+ coalesce(rcvdbyte, 0)	
) as bandwidth		
from		
\$log		
where		
\$filter		
and logid to int(lo	ogid) not in (4, 7, 14, 20)	
and (-	
srcssid is not n	ull	
or dstssid is no	t null	
)		
group by		
client		
having		
sum(

```
coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
)& gt; 0
order by
bandwidth desc
```

Dataset Name	Description	Log Category
wifi-Top-OS-By-Bandwidth	Top WiFi os by bandwidth usage	traffic
select		
(
	'unknown') ' ' coales	ce(osversion, '')
) as os,		
sum (
-	e, 0)+ coalesce(rcvdbyte, 0)	
) as bandwidth		
from		
\$log		
where		
\$filter	(1, 2, 2)	
and logid_co_int(ic and (ogid) not in (4, 7, 14, 20)	
srcssid is not nu	11	
or dstssid is not		
)		
, group by		
os		
having		
sum(
coalesce(sentbyte	e, 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>select (coalesce(osname, ') as os, count(distinct srcma from</pre>	unknown') ' ' co c) as totalnum	<pre>palesce(osversion, '')</pre>

###(select srcintf, srcssid, osname, osversion, devtype, srcmac, count(*) as subtotal from \$log where \$filter and logid_to_int (logid) not in (4, 7, 14, 20) and (srcssid is not null or dstssid is not null) and srcmac is not null group by srcintf, srcssid, osname, osversion, devtype, srcmac order by subtotal desc)### t group by os order by totalnum desc

Dataset Name	Description	Log Category
wifi-Top-Device-By-Bandwidth	Top WiFi device by bandwidth usage	traffic
<pre>) as bandwidth from \$log where \$filter and logid_to_int(l and (srcssid is not n or dstssid is no) and devtype is not group by devtype having sum(</pre>	t null	
Dataset Name	Description	Log Category
wifi-Top-Device-By-Client	Top WiFi device by client	traffic
<pre>select devtype, count(distinct src from ###(coloct evoint)</pre>	mac) as totalnum	_

###(select srcintf, srcssid, osname, osversion, devtype, srcmac,

count(*) as subtotal from \$log where \$filter and logid_to_int
(logid) not in (4, 7, 14, 20) and (srcssid is not null or dstssid
is not null) and srcmac is not null group by srcintf, srcssid,
osname, osversion, devtype, srcmac order by subtotal desc)### t
where devtype is not null group by devtype order by totalnum desc

Dataset Name	Description	Log Category
wifi-Overall-Traffic	WiFi overall traffic	traffic
select sum(coalesce(sentbyt) as bandwidth	.e, 0)+ coalesce(rcvdbyte,	0)
from		
\$log		
where		
\$filter		
and logid_to_int(l	ogid) not in (4, 7, 14, 20))
and (
srcssid is not n	ull	
or dstssid is no	t null	
)		

Dataset Name	Description	Log Category
wifi-Num-Distinct-Client	WiFi num distinct client	traffic
<pre>select count(distinct srcmac) as totalnum from ###(select srcintf, srcssid, osname, osversion, devtype, srcmac, count(*) as subtotal from \$log where \$filter and logid_to_int (logid) not in (4, 7, 14, 20) and (srcssid is not null or dstssid is not null) and srcmac is not null group by srcintf, srcssid, osname, osversion, devtype, srcmac order by subtotal desc)### t</pre>		
Dataset Name	Description	Log Category
Top30-Subnets-by-Bandwidth-and- Sessions	Top subnets by application bandwidth	traffic
<pre>select ip_subnet(`srcip`) sum(</pre>	as subnet,	

106

```
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 logid_to_int(logid) not in (4, 7, 14, 20)
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_group, sum(coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)) as bandwidth</pre>				
from				
<pre>\$log where \$filter and logid_to_int(logid) not in (4, 7, 14, 20) and nullifna(app) is not null</pre>				
<pre>group by subnet, app_group</pre>				

```
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`) a app_group_name(app) count(*) as sessions</pre>	as app_group,	
from		
\$log		
where		
\$filter		
and logid_to_int(log and nullifna(app) is	gid) not in (4, 7, 14, 20) s not null	
group by		
subnet,		
app_group		
order by		
sessions desc		

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

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

website,

###(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 logid_to_ int(logid) not in (4, 7, 14, 20) and (countweb>0 or ((logver is null or logver<52) and (hostname is not null or utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-
filter')))) group by 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	traffic
select		
subnet,		
website,		
sum(hits) as hits		
from		
(
— —	et(`srcip`) as subnet, hostname a	
	log-traffic where \$filter and ho	
	Int(logid) not in (4, 7, 14, 20)	
	', 'banned-word', 'web-content',	
) group by subnet, website order	-
	(select ip_subnet(`srcip`) as sub	
	(*) as hits from \$log-webfilter v	
ter and hostname is not	null and (eventtype is null or	logver>=52)

ter and hostname is not null and (eventtype is null or logver>=53 group by subnet, website order by hits desc)###) t group by subnet, website order by hits desc

Dataset Name	Description	Log Category
Top30-Subnets-with-Top10-User-by- Bandwidth	Top users by bandwidth	traffic
select		
<pre>ip_subnet(`srcip`) as coalesce(nullifna(`user`), nullifna(`unauthuse</pre>		
ipstr(`srcip`)		
) as user_src,		
<pre>sum(coalesce(sentbyte,) as bandwidth</pre>	0)+ coalesce(rcvdbyte, 0)	
from		
\$log		
where		
\$filter		
and logid_to_int(logi	ld) not in (4, 7, 14, 20)	

```
and srcip is not null
group by
subnet,
user_src
having
sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
)& gt; 0
order by
bandwidth desc
```

Dataset Name	Description	Log Category
Top30-Subnets-with-Top10-User-by- Sessions	Top users by sessions	traffic
select		
<pre>ip_subnet(`srcip`) as</pre>	s subnet,	
coalesce(
<pre>nullifna(`user`),</pre>		
nullifna(`unauthuse	er`),	
ipstr(`srcip`)		
) as user_src,		
count(*) as sessions		
from		
\$log		
where		
\$filter		
and logid_to_int(logi	d) not in (4, 7, 14, 20)	
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(</pre>		

```
coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
  ) as bandwidth
from
  $loq
where
  $filter
  and logid_to_int(logid) not in (4, 7, 14, 20)
group by
  appcat,
  app
having
  sum(
    coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
  )& gt; 0
order by
  bandwidth desc
```

Dataset Name	Description	Log Category
app-Top-20-Category-and- Applications-by-Session	Top category and applications by session	traffic
<pre>select appcat, app, count(*) as sessions from \$log where \$filter and logid_to_int(logi group by appcat, app order by sessions desc</pre>	.d) not in (4, 7, 14, 20)	

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

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
 $loq
where
  $filter
  and logid to int(logid) not in (4, 7, 14, 20)
 and action in ('accept', 'close', 'timeout')
group by
 timestamp,
 user src,
 appcat,
 app,
 destination
order by
 bandwidth desc
```

Dataset Name	Description	Log Category
app-Top-500-Blocked-Applications-by- Session	Top blocked applications by session	traffic
<pre>select coalesce(nullifna(`user`), nullifna(`unauthuse ipstr(`srcip`)) as user_src, appcat, app, count(*) as sessions</pre>	r`),	

```
from
  $log
where
  $filter
  and logid_to_int(logid) not in (4, 7, 14, 20)
  and action in (
     'deny', 'blocked', 'reset', 'dropped'
  )
group by
  user_src,
  appcat,
  app
order by
  sessions desc
```

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

```
select
```

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

from

###(select dtime, catdesc, hostname, cast(utmaction as text) as status, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from \$log-traffic where \$filter and hostname is not null and logid_to_int(logid) not in (4, 7, 14, 20) and (countweb>0 or ((logver is null or logver<52) and (hostname is not null or utmevent in ('webfilter', 'banned-word', 'web-content', 'commandblock', '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	traffic
select hod, website,		

```
sum(hits) as hits
from
```

(

###(select \$hour_of_day as hod, (hostname || ' (' || coalesce (`catdesc`, 'Unknown') || ')') as website, count(*) as hits from \$log-traffic where \$filter and hostname is not null and logid_to_ int(logid) not in (4, 7, 14, 20) and utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') group by hod, website order by hod, hits desc)### union all ### (select \$hour_of_day as hod, (hostname || ' (' || coalesce(`catdesc`, 'Unknown') || ')') as website , count(*) as hits from \$logwebfilter where \$filter and hostname is not null and (eventtype is null or logver>=52) group by hod, website order by hod, hits desc) ###) t group by hod, website order by hod, hits desc

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

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

###(select hostname as website, catdesc, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from \$log-traffic where \$filter and hostname is not null and logid_to_int(logid) not in (4, 7, 14, 20) and (countweb>0 or ((logver is null or logver<52) and (hostname is not null or utmevent in ('webfilter', 'bannedword', 'web-content', 'command-block', 'script-filter')))) group by website, catdesc order by bandwidth desc)### t group by website, catdesc order by bandwidth desc

Dataset Name	Description	Log Category
web-Top-20-Category-and-Websites- by-Session	Web top category and websites by session	traffic
<pre>select website, catdesc, sum(sessions) as hits</pre>		
from		

(###(select hostname as website, catdesc, count(*) as sessions from \$log-traffic where \$filter and hostname is not null and logid_to_int(logid) not in (4, 7, 14, 20) and utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'scriptfilter') group by website, catdesc order by sessions desc)### union all ###(select hostname as website, catdesc, count(*) as sessions from \$log-webfilter where \$filter and hostname is not null and (eventtype is null or logver>=52) 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
<pre>authuser`), ipstr(`srci desc, sum(coalesce(dura 0)+coalesce(rcvdbyte, 0) hostname is not null ar and action in ('accept' src, website, catdesc h (rcvdbyte, 0))>0 order</pre>	ecimal(18, 2)	vebsite, cat- e(sentbyte, e \$filter and 4, 7, 14, 20) time, user_ +coalesce
Dataset Name	Description	Log Category
web-Top-500-User-Visted-Websites-	Web top user visted websites by bandwidth usage	traffic

select website,

by-Bandwidth

```
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 logid_to_int(logid) not in (4, 7, 14, 20) and (countweb>0 or ((logver is null or logver<52) and (hostname is not null or utmevent in ('webfilter', 'bannedword', '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	traffic
<pre>select website, catdesc, sum(sessions) as sess</pre>	ions	
from		
	e as website, catdesc, count(*) as e \$filter and hostname is not null	

from \$log-traffic where \$filter and hostname is not null and logid_to_int(logid) not in (4, 7, 14, 20) and utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'scriptfilter') group by hostname, catdesc order by sessions desc)### union all ###(select hostname as website, catdesc, count(*) as sessions from \$log-webfilter where \$filter and hostname is not null and (eventtype is null or logver>=52) 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(*) as totalnum from \$log</pre>		

```
where
  $filter
  and clientfeature is not null
group by
  clientfeature
order by
  totalnum desc
```

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

select

```
os,
```

```
count(distinct hostname) as totalnum
```

from

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

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

select

```
fctver as fctver_short,
count(distinct hostname) as totalnum
```

from

###(select hostname, os, fctver from \$log where \$filter group by hostname, os, fctver)### 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
<pre>select profile, count(distinct hostna</pre>	.me) as totalnum	
<pre>from ###(soloct bostnamo </pre>	coalosco(nullifna(usingpolicy) !	No Pro-
<pre>###(select hostname, coalesce(nullifna(usingpolicy), 'No Pro- file') as profile from \$log where \$filter group by hostname, pro- file)### t group by profile order by totalnum desc</pre>		

Dataset Name	Description	Log Category
fct-Client-Summary	Client Summary	fct-event
file, nullifna(`use and os is not null	ume, deviceip, os, nullifna er`) as hostuser, fctver fr group by hostname, devicei group by hostname, device	com \$log where \$filter p, os, profile, hos-
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
<pre>select threat, sum(totalnum) as to</pre>	talnum	
from (
<pre>###(select threat, count(*) as totalnum from \$log-fct-traffic where \$filter and threat is not null and lower(utmevent)-</pre>		
='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 and lower(utmevent) and lower(threat) l group by hostname order by totalnum desc</pre>	null in ('webfilter', 'appfirewall')	
Dataset Name	Description	Log Category

fct-Top10-Infected-Devices-with-Virus- Top Infected Devices with Virus Malware
select
hostname,
sum(totalnum) as totalnum
from
(
 ###(select hostname, count(*) as totalnum from \$log-fcttraffic where \$filter and hostname is not null and lower(utmevent)
in ('antivirus', 'antimalware') group by hostname order by totalnum desc)### union all ###(select hostname, count(*) as totalnum
from \$log-fct-event where \$filter and hostname is not null and
virus is not null group by hostname order by totalnum desc)###) t

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

group by hostname order by totalnum desc

```
select
threat,
hostname,
hostuser,
utmaction
from
```

(

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

Dataset Name	Description	Log Category
fct-Web-Filter-Violations	Web Filter Violations	fct-traffic
<pre>select remotename, hostname, coalesce(nullifna(`user`), 'Unknown') as hostuser, utmaction, count(*) as totalnum from \$log where \$filter and lower(utmevent)= and utmaction = 'bloc </pre>	'webfilter'	
<pre>group by remotename, hostname, hostuser, utmaction order by totalnum desc</pre>		

Dataset Name	Description	Log Category
fct-Application-Firewall	Application Firewall	fct-traffic
select		
threat,		
hostname,		
hostuser,		
utmaction		
6		

from

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

Dataset Name	Description	Log Category
fct-Errors-and-Alerts	Errors and Alerts	fct-event
select		
msg,		
hostname,		
coalesce(
nullifna(`user`)	,	
'Unknown'		
) as hostuser		
from		
\$log		
where		
\$filter		
and level in ('er:	cor', 'alert')	
group by		
msg,		
hostname,		
hostuser		
Dataset Name	Description	Log Category

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

```
count(*) as totalnum
```

```
from
   $log
where
   $filter
   and hostname is not null
   and utmevent is not null
   and utmaction = 'blocked'
group by
   hostname
order by
   totalnum desc
```

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

select

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

from

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

Dataset Name	Description	Log Category
fct-vuln-Category-Type-Vulnerabilities	Vulnerabilities Detected by Category Type	fct-netscan
<pre>select vulncat, count(distinct vulnna from</pre>	me) as totalnum	
lifna(vulncat) is not n	ulnname from \$log where \$filter a ull and nullifna(vulnname) is not ame)### t group by vulncat order	null

Dataset Name	Description	Log Category
fct-vuln-Vulnerabilities-by-OS	Forticlient Vulnerabilities by OS	fct-netscan
select		
os,		
count(distinct vulnna	ame) as totalnum	

from

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

Dataset Name	Description	Log Category
fct-vuln-Vulnerabilities-by-Risk-Level	Number Vulnerability by Device and Risk Level	fct-netscan
select		
vulnseverity,		
(sites louitiesly then 5 other as	
	rity = 'Critical' then 5 when v	-
-	nseverity = 'Medium' then 3 whe	
-	n vulnseverity = 'Info' then 1	else 0 end
) as severity_number,	,	
count(distinct vulnna	ame) as vuln_num,	
count(distinct devid)	as dev_num	
from	—	
###(select vulnsever:	ity, devid, vulnname from \$log	where \$filter

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

Dataset Name	Description	Log Category
fct-vuln-Device-by-Risk-Level	Number Vulnerability by Device and Risk Level	fct-netscan
select		
vulnseverity,		
(
	ity = 'Critical' then 5 when vu	-
_	severity = 'Medium' then 3 when	
_	vulnseverity = 'Info' then 1 e	lse 0 end
) as severity_number,		
count(distinct vulnna	—	
count(distinct devid)	as dev_num	
from		
###(select vulnseveri	ty, devid, vulnname from \$log w	here \$filter
and nullifna(vulnseveri	ty) is not null and nullifna(vu	lnname) is
not null and nullifna(d	evid) is not null group by vuln	severity,
<pre>vulnname, devid) ### t g</pre>	roup by vulnseverity order by d	ev_num desc,

severity number desc



select

```
hostname,
(
    '<div style=word-break:normal>' || vulnname || '</div>'
) as vulnname,
vulnseverity,
vulncat,
string_agg(distinct products, ',') as products,
string_agg(distinct cve_id, ',') as cve_list,
```

(
 'Remediation Info'
) as vendor_link
from

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

Dataset Name	Description	Log Category
fct-vuln-Remediation-by-Device	Remediate The Vulnerability Found on Device	fct-netscan
select		
hostname,		
(
' <div style="word-</td"><td>oreak:normal>' vulnname </td><td></td></div> '	oreak:normal>' vulnname	
) as vulnname,		
vulnseverity,		
—	<pre>vendor_link, ',') as vendor_li</pre>	ink
from		
	, vulnname, vulnseverity, vulni	
where \$filter and vul	nname is not null and hostname	is not null
group by hostname, vul	lnname, vulnseverity, vulnid)##	‡# t1 inner
join fct_mdata t2 on [.]	t1.vulnid=t2.vid::int group by	hostname, vul-
nname, vulnseverity o:	rder by vulnseverity, hostname	
Detected Name	Description	

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

```
select
```

```
(
    '<b>' || vulnname || '</b><br/><br/>' || 'Description<br/>div
style=word-break:normal>' || description || '</div><br/>br/><br/>' ||
'Affected Products<br/>' || products || '<br/><br/>' || 'Recommended Action-
s<br/>' || vendor_link || '<br/><br/><br/>' || 'Recommended Action-
from
    ###(select devid, vulnname, vulnseverity, (case vulnseverity
when 'low' then 1 when 'info' then 2 when 'medium' then 3 when
```

'high' then 4 when 'critical' then 5 else 0 end) as severity_ level, vulnid from \$log where \$filter and vulnname is not null group by devid, vulnname, vulnseverity, severity_level, vulnid order by severity_level)### t1 inner join fct_mdata t2 on t1.vulnid=t2.vid::int group by remediation order by remediation

Dataset Name	Description	Log Category	
fct-vuln-Top-30-Targeted-High-Risk- Vulnerabilities	Top 30 Targeted High Risk Vulnerabilities	fct-netscan	
select			
t3.cve id,			
score,			
<pre>string_agg(distinct products, ',') as products,</pre>			
(
<pre>'<a '="" ,')="" href=" String_agg(vendor_link, " ="">Mitigation</pre>			
Infomation'			
) as vendor_link			
from			
###(select vulnid fro	om \$log where \$filter group	by vulnid)### t1	
inner join fct_mdata t2	2 on t2.vid=t1.vulnid::text	inner join fct_	

inner join fct_mdata t2 on t2.vid=t1.vulnid::text inner join fct_ cve_score t3 on strpos(t2.cve_id, t3.cve_id) > 0 group by t3.cve_ id, score order by score desc, t3.cve_id

Dataset Name	Description	Log Category
os-Detect-OS-Count	Detected operation system count	traffic
select		
(
coalesce(osname,	'Unknown')	
) as os,		
count(*) as totalnu	am	
from		
\$log		
where		
\$filter		
and logid_to_int(lo	ogid) not in (4, 7, 14, 20)	
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
<pre>authuser`), ipstr(`srci policyid, count(*) as s (rcvdbyte, 0)) as bandw and logid_to_int(logid) aser_src, dstip, srcint</pre>	<pre>, coalesce(nullifna(`user`), n p`)) as user_src, dstip, srcin essions, sum(coalesce(sentbyte idth from \$log where \$filter-e not in (4, 7, 14, 20) group b f, dstintf, policyid order by r-drilldown and nullifna(app)</pre>	tf, dstintf, , 0)+coalesce xclude-var y appid, app, sessions
Dataset Name	Description	Log Category
drilldown-Top-App-By-Sessions-Bar	Drilldown top applications by session count	traffic
<pre>uthuser`), ipstr(`srci olicyid, count(*) as s rcvdbyte, 0)) as bandw nd logid_to_int(logid) ser_src, dstip, srcint</pre>	<pre>, coalesce(nullifna(`user`), n p`)) as user_src, dstip, srcin essions, sum(coalesce(sentbyte idth from \$log where \$filter-e not in (4, 7, 14, 20) group b f, dstintf, policyid order by r-drilldown and nullifna(app)</pre>	tf, dstintf, , 0)+coalesc xclude-var y appid, app sessions
Dataset Name	Description	Log Category
drilldown-Top-App-By-Bandwidth- Table	Drilldown top applications by bandwidth usage	traffic
elect appid, app,	dwidth	

```
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 logid_to_int(logid) not in (4, 7, 14, 20) group by appid, app, user_src, dstip, srcintf, dstintf, policyid order by sessions desc)### t where \$filter-drilldown and nullifna(app) is not null group by appid, app having sum(bandwidth)>0 order by bandwidth desc

Dataset Name	Description	Log Category
drilldown-Top-App-By-Bandwidth-Bar	Drilldown top applications by bandwidth usage	traffic
select		
appid,		
app,		
sum(bandwidth) as ban	dwidth	
from		
###(select appid, app	<pre>, coalesce(nullifna(`user`), nul</pre>	llifna(`un-
_	p`)) as user_src, dstip, srcint:	
	essions, sum(coalesce(sentbyte,	
(rcvdbyte, 0)) as bandw	idth from \$log where \$filter-exc	clude-var
and logid_to_int(logid)	not in (4, 7, 14, 20) group by	appid, app,
user_src, dstip, srcint	f, dstintf, policyid order by se	essions
<pre>desc)### t where \$filte</pre>	r-drilldown and nullifna(app) is	s not null
group by appid, app hav	ing sum(bandwidth)>0 order by ba	andwidth
desc		

Dataset Name	Description	Log Category
drilldown-Top-Destination-By- Sessions-Table	Drilldown top destination by session count	traffic
<pre>select dstip, sum(sessions) as sess from</pre>	ions	
<pre>authuser`), ipstr(`srci policyid, count(*) as s (rcvdbyte, 0)) as bandw</pre>	<pre>, coalesce(nullifna(`user`), null p`)) as user_src, dstip, srcintf, essions, sum(coalesce(sentbyte, 0 idth from \$log where \$filter-excl not in (4, 7, 14, 20) group by a</pre>	dstintf,)+coalesce ude-var

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
<pre>authuser`), ipstr(`sr policyid, count(*) as (rcvdbyte, 0)) as ban and logid_to_int(logi user_src, dstip, srci desc)### t where \$fil</pre>	andwidth pp, coalesce(nullifna(`user`), cip`)) as user_src, dstip, srci sessions, sum(coalesce(sentbyt dwidth from \$log where \$filter- d) not in (4, 7, 14, 20) group ntf, dstintf, policyid order by ter-drilldown and dstip is not width)>0 order by bandwidth des	ntf, dstintf, e, 0)+coalesce exclude-var by appid, app, sessions null group by
Dataset Name	Description	Log Category
drilldown-Top-User-By-Sessions-Tab	le Drilldown top user by session count	traffic
<pre>select user_src, sum(sessions) as sessions from ###(select appid, app, coalesce(nullifna(`user`), nullifna(`un- authuser`), 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 logid_to_int(logid) not in (4, 7, 14, 20) 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</pre>		
Dataset Name	Description	Log Category
drilldown-Top-User-By-Sessions-Bar	Drilldown top user by session count	traffic
select		

user_src,

sum(sessions) as sessions
from

###(select appid, app, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, dstip, srcintf, dstintf, policyid, count(*) as sessions, sum(coalesce(sentbyte, 0)+coalesce (rcvdbyte, 0)) as bandwidth from \$log where \$filter-exclude-var and logid_to_int(logid) not in (4, 7, 14, 20) group by appid, app, user_src, dstip, srcintf, dstintf, policyid order by sessions desc)### t where \$filter-drilldown and user_src is not null group by user src order by sessions desc

Dataset Name	Description	Log Category
drilldown-Top-User-By-Bandwidth- Table	Drilldown top user by bandwidth usage	traffic
<pre>authuser`), ipstr(`srci policyid, count(*) as s (rcvdbyte, 0)) as bandw and logid_to_int(logid) user_src, dstip, srcint desc)### t where \$filte</pre>	<pre>dwidth , coalesce(nullifna(`user`), nul p`)) as user_src, dstip, srcintf essions, sum(coalesce(sentbyte, idth from \$log where \$filter-exc not in (4, 7, 14, 20) group by f, dstintf, policyid order by se r-drilldown and user_src is not bandwidth)>0 order by bandwidth</pre>	, dstintf, 0)+coalesce lude-var appid, app, ssions null group

Dataset Name	Description	Log Category
drilldown-Top-User-By-Bandwidth-Bar	Drilldown top user by bandwidth usage	traffic
<pre>select user_src, sum(bandwidth) as ban</pre>	dwidth	
<pre>authuser`), ipstr(`srci policyid, count(*) as s (rcvdbyte, 0)) as bandw</pre>	<pre>, coalesce(nullifna(`user`), p`)) as user_src, dstip, src essions, sum(coalesce(sentby idth from \$log where \$filter</pre>	cintf, dstintf, yte, 0)+coalesce r-exclude-var
	not in (4, 7, 14, 20) group f, dstintf, policyid order b	

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>select user_src, sum(requests) as visi</pre>	ts	
from		

(

###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, hostname, count(*) as requests from \$log-traffic where \$filter-exclude-var and logid_to_int(logid) not in (4, 7, 14, 20) and utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') and hostname is not null group by user_src, hostname order by requests desc)### union all ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, hostname, count(*) as requests from \$log-webfilter where \$filter-exclude-var and (eventtype is null or logver>=52) and hostname is not null group by user_src, hostname order by requests desc)###) t where \$filter-drilldown and user_src is not null group by user_src order by visits desc

Dataset Name	Description	Log Category
drilldown-Top-Web-User-By-Visit-Bar	Drilldown top web user by visit	traffic
select		
user_src,		
sum(requests) as visi	lts	
from		
(
###(select coalesce	e(nullifna(`user`), nullif	<pre>Ena(`unauthuser`),</pre>
<pre>ipstr(`srcip`)) as user</pre>	_src, hostname, count(*)	as requests from
<pre>\$log-traffic where \$fil</pre>	lter-exclude-var and logic	d_to_int(logid) not
in (4, 7, 14, 20) and u	tmevent in ('webfilter',	'banned-word',
'web-content', 'command	d-block', 'script-filter')	and hostname is
not null group by user_	src, hostname order by re	equests desc)###
union all ###(select co	<pre>palesce(nullifna(`user`),</pre>	<pre>ipstr(`srcip`)) as</pre>
user_src, hostname, cou	int(*) as requests from \$1	og-webfilter where
\$filter-exclude-var and	d (eventtype is null or lo	ogver>=52) and host
name is not null group	by user_src, hostname ord	ler by requests

desc)###) t where \$filter-drilldown and user_src is not null group by user src order by visits desc

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

select

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

```
(
```

###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, hostname, count(*) as requests from \$log-traffic where \$filter-exclude-var and logid_to_int(logid) not in (4, 7, 14, 20) and utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') and hostname is not null group by user_src, hostname order by requests desc)### union all ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, hostname, count(*) as requests from \$log-webfilter where \$filter-exclude-var and (eventtype is null or logver>=52) and hostname is not null group by user_src, hostname order by requests desc)###) t where \$filter-drilldown and hostname is not null group by hostname order by visits desc

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

hostname, sum(requests) as visits from

(

###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, hostname, count(*) as requests from \$log-traffic where \$filter-exclude-var and logid_to_int(logid) not in (4, 7, 14, 20) and utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') and hostname is not null group by user_src, hostname order by requests desc)### union all ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, hostname, count(*) as requests from \$log-webfilter where \$filter-exclude-var and (eventtype is null or logver>=52) and hostname is not null group by user_src, hostname order by requests desc)###) t where \$filter-drilldown and hostname is not null group by hostname order by visits desc

Dataset Name	Description	Log Category
drilldown-Top-Email-Sender-By- Volume	Drilldown top email sender by volume	traffic
<pre>select sender, sum(bandwidth) as vo</pre>	lume	
from		
(

###(select sender, recipient, count(*) as requests, sum (coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from \$log-traffic where \$filter-exclude-var and logid_to_int(logid) not in (4, 7, 14, 20) and service in ('smtp', 'SMTP', '25/tcp', '587/tcp', 'smtps', 'SMTPS', '465/tcp') and utmevent in ('generalemail-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 \$filterdrilldown and sender is not null group by sender having sum(bandwidth)>0 order by volume desc

Dataset Name	Description	Log Category
drilldown-Top-Email-Send-Recipient- By-Volume	Drilldown top email send recipient by volume	traffic
<pre>select recipient, sum(bandwidth) as vol finam</pre>	ume	
from (
<pre>(coalesce(sentbyte, 0)+ \$log-traffic where \$fil in (4, 7, 14, 20) and s</pre>	recipient, count(*) as requests, coalesce(rcvdbyte, 0)) as bandwid ter-exclude-var and logid_to_int ervice in ('smtp', 'SMTP', '25/to TTPS', '465/tcp') and utmevent in	dth from (logid) not cp',

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 \$filterdrilldown 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 requ	ests	
from		
(
###(select sender,	recipient, count(*) as reques	ts, sum
<pre>(coalesce(sentbyte, 0)+</pre>	<pre>coalesce(rcvdbyte, 0)) as ban</pre>	dwidth from
-	ter-exclude-var and logid_to_	
	ervice in ('smtp', 'SMTP', '2	
	TPS', '465/tcp') and utmevent	2
	') group by sender, recipient	_
-	all ###(select `from` as sen	
-	requests, sum(coalesce(sentby	
_ ·	as bandwidth from \$log-emailf	
	service in ('smtp', 'SMTP',	
	TPS', '465/tcp') and eventtyp	
	rder by requests desc)###) t	
desc	not null group by sender ord	er by requests
UEBU		

Dataset Name	Description	Log Category
drilldown-Top-Email-Send-Recipient- By-Count	Drilldown top email send recipient by count	traffic
<pre>select recipient, sum(requests) as requ from (</pre>	ests	

###(select sender, recipient, count(*) as requests, sum (coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from \$log-traffic where \$filter-exclude-var and logid_to_int(logid) not in (4, 7, 14, 20) and service in ('smtp', 'SMTP', '25/tcp', '587/tcp', 'smtps', 'SMTPS', '465/tcp') and utmevent in ('generalemail-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 \$filterdrilldown and recipient is not null group by recipient order by requests desc

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

select

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

(

###(select recipient, sender, count(*) as requests, sum (coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from \$log where \$filter-exclude-var and logid_to_int(logid) not in (4, 7, 14, 20) and service in ('pop3', 'POP3', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/tcp', 'pop3s', 'POP3S', '995/tcp') and utmevent in ('general-email-log', 'spamfilter') group by recipient, sender order by requests desc)### union all ###(select `to` as recipient, `from` as sender, count(*) as requests, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from \$log-emailfilter where \$filter-exclude-var and service in ('pop3', 'POP3', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/tcp', 'pop3s', 'POP3S', '995/tcp') and eventtype is null group by `to`, `from` order by requests desc)###) t where \$filter-drilldown and recipient is not null group by recipient having sum(bandwidth)>0 order by volume desc

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

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

###(select recipient, sender, count(*) as requests, sum (coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from \$log where \$filter-exclude-var and logid_to_int(logid) not in (4, 7, 14, 20) and service in ('pop3', 'POP3', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/tcp', 'pop3s', 'POP3S', '995/tcp') and utmevent in ('general-email-log', 'spamfilter') group by recipient, sender order by requests desc)### union all ###(select `to` as recipient, `from` as sender, count(*) as requests, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from \$log-emailfilter where \$filter-exclude-var and service in ('pop3', 'POP3', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/tcp', 'pop3s', 'POP3S', '995/tcp') and eventtype is null group by `to`, `from` order by requests desc)###) t where \$filter-drilldown and sender is not null group by sender having sum(bandwidth)>0 order by volume desc

Dataset Name	Description	Log Category
drilldown-Top-Email-Recipient-By- Count	Drilldown top email receiver by count	traffic
<pre>select recipient, sum(requests) as req</pre>	uests	
from (
	nt, sender, count(*) as requ +coalesce(rcvdbyte, 0)) as b	
7, 14, 20) and service	<pre>lude-var and logid_to_int(lo in ('pop3', 'POP3', '110/to</pre>	cp', 'imap',
'995/tcp') and utmeven	aps', 'IMAPS', '993/tcp', 'p t in ('general-email-log', '	'spamfilter')
<pre>group by recipient, sender order by requests desc)### union all ###(select `to` as recipient, `from` as sender, count(*) as</pre>		
requests, sum(coalesce	<pre>(sentbyte, 0)+coalesce(rcvdb</pre>	oyte, 0)) as band

in ('pop3', 'POP3', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/tcp', 'pop3s', 'POP3S', '995/tcp') and eventtype is

width from \$log-emailfilter where \$filter-exclude-var and service

null group by `to`, `from` order by requests desc)###) t where \$filter-drilldown and recipient is not null group by recipient order by requests desc

Dataset Name	Description	Log Category
drilldown-Top-Email-Receive-Sender- By-Count	Drilldown top email receive sender by count	traffic
<pre>select sender, sum(requests) as requ</pre>	lests	

from (

###(select recipient, sender, count(*) as requests, sum (coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from \$log where \$filter-exclude-var and logid_to_int(logid) not in (4, 7, 14, 20) and service in ('pop3', 'POP3', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/tcp', 'pop3s', 'POP3S', '995/tcp') and utmevent in ('general-email-log', 'spamfilter') group by recipient, sender order by requests desc)### union all ###(select `to` as recipient, `from` as sender, count(*) as requests, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from \$log-emailfilter where \$filter-exclude-var and service in ('pop3', 'POP3', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/tcp', 'pop3s', 'POP3S', '995/tcp') and eventtype is null group by `to`, `from` order by requests desc)###) t where \$filter-drilldown and sender is not null group by sender order by requests desc

Dataset Name	Description	Log Category
drilldown-Top-Attack-Destination	Drilldown top attack dest	attack
select dstip, sum(totalnum) as tota	alnum	
from		
###(select srcip, dst	tip, count(*) as totalnum from	\$log where
\$filter-exclude-var gro	oup by srcip, dstip order by t	otalnum
<pre>desc)### t where \$filte</pre>	er-drilldown and dstip is not	null group by
dstip order by totalnur	n desc	

Dataset Name	Description	Log Category
drilldown-Top-Attack-Source	Drilldown top attack source	attack
<pre>select srcip, sum(totalnum) as to </pre>	otalnum	
<pre>from ###(select srcip, of</pre>	dstip, count(*) as totalnum	from \$log where
	group by srcip, dstip order lter-drilldown and srcip is	-

srcip order by totalnum desc

Dataset Name	Description	Log Category
drilldown-Top-Attack-List	Drilldown top attack list	attack
<pre>select from_itime(itime) attack, srcip, dstip</pre>	as timestamp,	
from		
	attack, srcip, dstip from \$ itime desc)### t where \$fi	-

order by timestamp desc

Dataset Name	Description		Log Category
drilldown-Top-Virus	UTM top virus		virus
select virus,			
<pre>max(virusid_s) as (</pre>	virusid,		
case when virus like 'Adware%' then	like 'Riskware%' th		u virus
) as malware_type,			
<pre>sum(totalnum) as t from</pre>	otalnum		
(
	, '' as virusid_s,		
<pre>\$log-traffic where \$ 14, 20) and utmevent virus, virusid_s ord</pre>	is not null and vi	rus is not null g	roup by

virus, virusid_to_str(virusid, eventtype) as virusid_s, count(*) as totalnum from \$log-virus where \$filter and (eventtype is null or logver>=52) and nullifna(virus) is not null group by virus, virusid_s order by totalnum desc)###) t group by virus, malware_type order by totalnum desc

Dataset Name	Description	Log Category
drilldown-Virus-Detail	Drilldown virus detail	traffic
(`unauthuser`), ip recipient from \$lo not in (4, 7, 14, null order by itim coalesce(nullifna(cast(' ' as char) \$log-virus where \$ nullifna(virus) is) as timestamp, me, virus, coalesce(nullifna str(`srcip`)) as user_src, ds g-traffic where \$filter and 1 20) and utmevent is not null e desc)### union all ###(sele `user`), ipstr(`srcip`)) as u as hostname, cast(' ' as chan filter and (eventtype is null not null order by itime desc r by timestamp desc	stip, hostname, logid_to_int(logid) and virus is not ect itime, virus, user_src, dstip, r) as recipient from l or logver>=52) and
user-drilldown-Top-Blocked-We By-Requests	eb-Sites- User drilldown top blocked web sites by re	equests webfilter
<pre>src, hostname, act ter and hostname i order by requests</pre>	<pre>requests sce(nullifna(`user`), ipstr(` ion, count(*) as requests fro s not null group by user_src, desc)### t where \$filter-drip y hostname order by requests</pre>	om \$log where \$fil- , hostname, action lldown and action-

Dataset Name	Description	Log Category
user-drilldown-Top-Allowed-Web-Sites- By-Requests	User drilldown top allowed web sites by requests	webfilter
src, hostname, action, ter and hostname is not order by requests desc)	ests ullifna(`user`), ipstr(`srcip count(*) as requests from \$loo null group by user_src, host: ### t where \$filter-drilldown tname order by requests desc	g where \$fil- name, action
Dataset Name	Description	Log Category
user-drilldown-Top-Blocked-Web- Categories	User drilldown top blocked web categories	webfilter
src, catdesc, action, c and catdesc is not null oy requests desc)### t	ests ullifna(`user`), ipstr(`srcip ount(*) as requests from \$log group by user_src, catdesc, a where \$filter-drilldown and a desc order by requests desc	where \$filter action order
Dataset Name	Description	Log Category
user-drilldown-Top-Allowed-Web- Categories	User drilldown top allowed web categories	webfilter
<pre>select catdesc, sum(requests) as requ from ###(select coalesce(n)</pre>	ests ullifna(`user`), ipstr(`srcip)) as user

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

Dataset Name			
	Description	Log Category	
user-drilldown-Top-Attacks	User drilldown top attacks by name	attack	
<pre>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 \$fil- ter-drilldown group by attack order by attack_count desc</pre>			
ter-ariiiaown group r	by attack order by attack_count		
Dataset Name	Dy attack order by attack_count Description		
		t desc	

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

Dataset Name	Description	Log Category
user-drilldown-Top-Virus-By-Name	User drilldown top virus	virus
<pre>select virus, max(virusid_s) as vir sum(totalnum) as tota</pre>		
from		
<pre>###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_ src, virus, virusid_to_str(virusid, eventtype) as virusid_s, count (*) as totalnum from \$log where \$filter and nullifna(virus) is not null group by user_src, virus, virusid_s order by totalnum</pre>		

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 \$fil- ter 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 user-drilldown-Count-Spam-Activity-by- Hour-of-Day	Description User drilldown count spam activity by hour of day	Log Category emailfilter		
<pre>select hourstamp, sum(totalnum) as totalnum from ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_ src, \$hour_of_day as hourstamp, count(*) as totalnum from \$log where \$filter and `to` is not null and action in ('detected', 'blocked') group by user_src, hourstamp order by hourstamp)### t where \$filter-drilldown group by hourstamp order by hourstamp</pre>				
Dataset Name	Description	Log Category		
user-drilldown-Top-Spam-Sources	User drilldown top spam sources	emailfilter		
select mf_sender, sum(totalnum) as tota	alnum			

```
from
```

###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_ src, `from` as mf_sender, count(*) as totalnum from \$log where \$filter and `from` is not null and action in ('detected', 'blocked') group by user_src, mf_sender order by totalnum desc)### t where \$filter-drilldown group by mf_sender order by totalnum desc

Dataset Name	Description	Log Category
event-Usage-CPU	Event usage CPU	event
<pre>) as cpu_avg_usage from ###(select \$hour_of_d count(*) as num from \$1</pre>	n(num) as decimal(6, 2) lay as hourstamp, sum(cpu) as cpu_ log where \$filter and subtype='sys oup by hourstamp)### t group by ho	stem' and

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

```
sum(sess usage) / sum(num) as decimal(10, 2)
```

```
) as sess_avg_usage
from
  ###(select $hour_of_day as hourstamp, sum(totalsession) as sess_
usage, count(*) as num from $log where $filter and sub-
type='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,
```

```
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(`unauthuse ipstr(`srcip`)) as user_src, srcip, sum(</pre>	er`),	
<pre>coalesce(sentbyte,) as bandwidth, sum(</pre>	0)+ coalesce(rcvdbyte,	0)
<pre>coalesce(rcvdbyte,) as traffic_in, sum(</pre>	0)	
coalesce(sentbyte,	0)	
```
) as traffic_out
from
  $log
where
  $filter
  and logid to int(logid) not in (4, 7, 14, 20)
  and srcip is not null
group by
 user_src,
  srcip
having
  sum(
    coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
  )& gt; 0
order by
  bandwidth desc
```

Dataset Name	Description	Log Category
App-Risk-Top-User-Source-By- Sessions	Application risk top user source by session count	traffic
and srcip is not null group by srcip, user_src	d) not in (4, 7, 14, 20)	
order by sessions desc		

Dataset Name	Description	Log Category
App-Risk-Top-Users-By-Reputation- Scores-Bar	Application risk reputation top users by scores	traffic
<pre>select coalesce(nullifna(`user`), nullifna(`unauthus ipstr(`srcip`)) as user_src, sum(crscore % 65536) from \$log where \$filter and logid_to_int(log and crscore is not no group by user_src having sum(crscore % 65536) order by scores desc</pre>	as scores id) not in (4, 7, 14, 20) ull	

App-Risk-Top-Devices-By-Reputation- Scores	Application risk reputation top devices by scores	traffic
<pre>select devtype, coalesce(nullifna(`srcname`), nullifna(`srcmac`), ipstr(`srcip`)</pre>	,	

Description

```
) as dev_src,
sum(crscore % 65536) as scores
from
  $log
where
  $filter
  and logid_to_int(logid) not in (4, 7, 14, 20)
  and crscore is not null
```

Log Category

```
group by
  devtype,
  dev_src
having
  sum(crscore % 65536)& gt; 0
order by
  scores desc
```

Dataset Name	Description	Log Category
App-Risk-Application-Usage-By- Category-With-Pie	Application risk application usage by category	traffic
) as bandwidth from \$log where \$filter	0)+ coalesce(rcvdbyte, 0) gid) not in (4, 7, 14, 20) is not null	
bandwidth desc		

Dataset Name	Description	Log Category
App-Risk-App-Usage-by-Category	Application risk application usage by category	traffic
<pre>select appcat, sum(coalesce(sentbyte,) as bandwidth</pre>	0)+ coalesce(rcvdbyte, 0)	
from		
\$log		
where		
\$filter		
and logid_to_int(log and nullifna(appcat)	id) not in (4, 7, 14, 20) 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
<pre>select catdesc, sum(bandwidth) as b</pre>	andwidth	
<pre>0)) as bandwidth from (logid) not in (4, 7, or logver<52) and (ho filter', 'banned-word filter')))) and catde</pre>	<pre>sum(coalesce(sentbyte, 0)+coal \$log-traffic where \$filter and 14, 20) and (countweb>0 or () stname is not null or utmevent ', 'web-content', 'command-bloc sc is not null group by catdes p by catdesc order by bandwidt</pre>	nd logid_to_int (logver is null in ('web- ock', 'script- sc order by band-
Dataset Name	Description	Log Category
App-Risk-Key-Applications-Crossing The-Network	- Application risk application activity	traffic
<pre>select app_group_name(app) appcat, sum(coalesce(sentbyte) as bandwidth, count(*) as num ses</pre>	<pre>, 0)+ coalesce(rcvdbyte, 0)</pre>	
from \$log		
<pre>where \$filter and logid_to_int(lo and nullifna(app) i</pre>	gid) not in (4, 7, 14, 20) s not null	
<pre>group by app_group, appcat order by</pre>		

Dataset Name	Description	Log Category
App-Risk-Applications-Running-Over- HTTP	Application risk applications running over HTTP	traffic
select		
app_group_name(app) a	as app_group,	
service,	_	
count(*) as sessions,	,	
sum(
coalesce(sentbyte,	0)+ coalesce(rcvdbyte, 0)	
) as bandwidth		
from		
\$log		
where		
\$filter		
and logid_to_int(log:	id) not in (4, 7, 14, 20)	
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

Dalasel Name	Description	Log Calegory
App-Risk-Top-Web-Sites-Visited-By- Network-Users-Pie-Cha	Application risk web browsing summary category	traffic
<pre>select catdesc, sum(num_sess) as num_ sum(bandwidth) as ban</pre>		
from		
###(select catdesc, c	ount(*) as num_sess, sum(coales	ce(sentbyte,
0)+coalesce(rcvdbyte, 0)) as bandwidth from \$log-traff	ic where

\$filter and logid_to_int(logid) not in (4, 7, 14, 20) and (countweb>0 or ((logver is null or logver<52) and (hostname is not null or utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter')))) and catdesc is not null group by catdesc order by num_sess desc)### t group by catdesc order by num sess desc

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

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

from

###(select catdesc, count(*) as num_sess, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from \$log-traffic where \$filter and logid_to_int(logid) not in (4, 7, 14, 20) and (countweb>0 or ((logver is null or logver<52) and (hostname is not null or utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter'))) and catdesc is not null group by catdesc order by num_sess desc)### t group by catdesc order by num_sess desc

Dataset Name	Description	Log Category
App-Risk-Web-Browsing-Hostname- Category	Application risk web browsing activity hostname category	traffic
<pre>select domain, catdesc, sum(visits) as visits</pre>		
from		
(
	(nullifna(hostname), ipstr(`dstip	
<pre>domain, catdesc, count(*) as visits from \$log-traffic where \$fil- ter and logid_to_int(logid) not in (4, 7, 14, 20) and utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block',</pre>		
<pre>'script-filter') and catdesc is not null group by domain, catdesc order by visits desc)### union all ###(select coalesce(nullifna (hostname), ipstr(`dstip`)) as domain, catdesc, count(*) as visits</pre>		

from \$log-webfilter where \$filter and (eventtype is null or logver>=52) and catdesc is not null group by domain, catdesc order by visits desc)###) t group by domain, catdesc order by visits desc

Dataset Name	Description	Log Category
Top-Destination-Countries-By- Browsing-Time	Traffic top destination countries by browsing time	traffic
<pre>select dstcountry, ebtr_value(ebtr_agg_flat(brow null, \$timespan) as browsetime, sum(bandwidth) as ba sum(traffic_in) as t sum(traffic_out) as</pre>	andwidth, craffic_in,	
<pre>sum(bandwidth) as band (traffic_out) as traff flat(\$browse_time) as coalesce(rcvdbyte, 0)) traffic_in, sum(coales where \$filter and logi \$browse_time is not nu country order by ebtr_</pre>	<pre>ty, ebtr_agg_flat(browsetime) as dwidth, sum(traffic_in) as traffic fic_out from (select dstcountry, browsetime, sum(coalesce(sentbyt as bandwidth, sum(coalesce(rcvd sce(sentbyte, 0)) as traffic_out d_to_int(logid) not in (4, 7, 14 all group by dstcountry) t group value(ebtr_agg_flat(browsetime), stcountry order by browsetime des</pre>	<pre>c_in, sum ebtr_agg_ e, 0)+- byte, 0)) as from \$log , 20) and by dst- null, null)</pre>

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(brows null, \$timespan) as browsetime,</pre>	setime),	

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

###(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 logid_to_int(logid) not in (4, 7, 14, 20) and \$browse_time is not null group by dstcountry) t group by dstcountry order by ebtr_value(ebtr_agg_flat(browsetime), null, null) desc)### t group by dstcountry order by browsetime desc

Dataset Name	Description	Log Category
App-Risk-Traffic-Top-Hostnames-By- Browsing-Time	Traffic top domains by browsing time	traffic
select		
hostname, ebtr_value(
<pre>ebtr_agg_flat(brows null,</pre>	setime),	
\$timespan		
) as browsetime, sum(bandwidth) as ban	dwidth,	
<pre>sum(traffic_in) as tr</pre>	affic_in,	
<pre>sum(traffic_out) as t from</pre>	raffic_out	
	ebtr_agg_flat(browsetime) width, sum(traffic in) as t	
	.c out from (select hostnar	_ '

(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 logid_to_int(logid) not in (4, 7, 14, 20) and hostname is not null and \$browse_time is not null group by hostname) t group by hostname order by ebtr_value(ebtr_agg_flat(browsetime), null, null) desc)### t group by hostname order by browsetime desc

Dataset Name	Description	Log Category
App-Risk-Traffic-Top-Hostnames-By- Browsing-Time-Enhanced	Traffic top domains by browsing time enhanced	traffic
select		
hostname,		
ebtr_value(
ebtr_agg_flat(brows	setime),	
null,		
Ştimespan		
) as browsetime,		
sum(bandwidth) as bar		
<pre>sum(traffic_in) as tr</pre>	—	
<pre>sum(traffic_out) as t </pre>	craffic_out	
from		
	ebtr_agg_flat(browsetime) as b	
	vidth, sum(traffic_in) as traff c out from (select hostname, e	
—	setime, sum(coalesce(sentbyte,	
—	width, sum(coalesce(sentbyte, 0	
	ce(sentbyte, 0)) as traffic out	
—	$l_{to_int(logid)}$ not in (4, 7, 1	
	nd \$browse time is not null gro	
	ame order by ebtr value(ebtr ag	
setime), null, null) de	esc)### t group by hostname ord	er 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,</pre>		
ref, count(*) as totalnum		
from \$log		
where \$filter		
and severity = 'criti		
<pre>and nullifna(attack) group by attack, severity,</pre>	is not null	
ref order by		
totalnum desc		

Dataset Name	Description	Log Category
App-Risk-Top-High-Threat-Vectors- Crossing-The-Network	Application risk top high threat vectors	attack
<pre>select attack, severity, ref, count(*) as totalnum from \$log where \$filter and severity = 'high and nullifna(attack)</pre>		
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</pre>		
<pre>where \$filter and severity = 'mediu and nullifna(attack)</pre>		
group by attack, severity, ref order by		
totalnum desc		

Dataset Name

App-Risk-Top-Low-Threat-Vectors- Crossing-The-Network Application risk top low threat vectors attack select attack, severity, ref, count(*) as totalnum from \$log where \$filter and severity = 'low' and nullifna(attack) is not null group by group by attack,		•		
<pre>attack, severity, ref, count(*) as totalnum from \$log where \$filter and severity = 'low' and nullifna(attack) is not null group by</pre>		Application risk top low threat vectors	attack	
<pre>from \$log where \$filter and severity = 'low' and nullifna(attack) is not null group by</pre>	attack, severity, ref,			
<pre>\$filter and severity = 'low' and nullifna(attack) is not null group by</pre>	from \$log			
	<pre>\$filter and severity = 'low' and nullifna(attack) group by</pre>	is not null		

Description

Log Category

severity, ref order by totalnum desc

Dataset Name	Description	Log Category
App-Risk-Top-Info-Threat-Vectors- Crossing-The-Network	Application risk top info threat vectors	attack
<pre>select attack, severity, ref, count(*) as totalnum</pre>	1	
from \$log where		
<pre>\$filter and severity = 'info and nullifna(attack)</pre>		
group by attack, severity, ref		
order by totalnum desc		
Dataset Name	Description	Log Category

Dataset Name	Description		Log Category
App-Risk-Top-Virus-By-Name	UTM top virus		virus
select			
virus,			
max(virusid s) as	virusid,		
(
case when virus	like 'Riskware%' t	then 'Spyware'	when virus
like 'Adware%' then	'Adware' else 'Vi	rus' end	
) as malware_type.	,		
sum(totalnum) as	totalnum		
from			
(
###(select virus	s, '' as virusid_s,	, count(*) as	totalnum from
<pre>\$log-traffic where \$</pre>	\$filter and logid_1	to_int(logid)	not in (4, 7,

14, 20) and utmevent is not null and virus is not null group by virus, virusid_s order by totalnum desc)### union all ###(select virus, virusid_to_str(virusid, eventtype) as virusid_s, count(*) as totalnum from \$log-virus where \$filter and (eventtype is null or logver>=52) and nullifna(virus) is not null group by virus, virusid_s order by totalnum desc)###) t group by virus, malware_type order by totalnum desc

Dataset Name	Description	Log Category
App-Risk-Top-Virus-Victim	UTM top virus user	traffic
<pre>select user_src, sum(totalnum) as t from (###(select coale ipstr(`srcip`)) as u traffic where \$filte 20) and utmevent is src order by totalnut lifna(`user`), ipstr from \$log-virus wher >=52) and nullifna(v</pre>		alnum from \$log-) not in (4, 7, 14, t null group by user_ (select coalesce(nul- count(*) as totalnum is null or logver>- by user_src order by

Dataset Name	Description	Log Category
App-Risk-Data-Loss-Prevention-Type- Events	Application risk DLP UTM event	traffic
select utmsubtype,		

```
sum(number) as number
from
(
```

###(select utmsubtype, count(*) as number from \$log-traffic where \$filter and logid_to_int(logid) not in (4, 7, 14, 20) and utmevent='dlp' and utmsubtype is not null group by utmsubtype order by number desc)### union all ###(select subtype::text as utmsubtype, count(*) as number from \$log-dlp where \$filter and subtype is not null group by subtype order by number desc)###) t group by utmsubtype order by number desc

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

Dataset Name	Description	Log Category
App-Risk-Malware-Discovered	Application risk virus discovered	traffic
select		
dom,		
sum(totalnum) as to	otalnum	
from		
(
###(select \$DAY_0	OF_MONTH as dom, count(*) a	s totalnum from
<pre>\$log-traffic where \$filter and logid_to_int(logid) not in (4, 7,</pre>		
14, 20) and utmevent	is not null and virus is n	ot null group by
dom order by totalnur	n desc)### union all ###(se	lect \$DAY_OF_MONTH
as dom, count(*) as t	totalnum from \$log-virus wh	ere \$filter and nul-
lifna(virus) is not r	null and (eventtype is null	or logver>=52)
group by dom order by	y totalnum desc)###) t grou	p 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 logid_to_int(logid) not in (4, 7, 14, 20)
group by
    d_behavior
order by
    number desc
```

Dataset Name

Description

App-Risk-Number-Of-Applications-By- Application risk number of applications by risk behavior traffic Risk-Behavior

```
select
  risk as d risk,
  unnest(
    string to array(behavior, ',')
  ) as f behavior,
  count(*) as number
from
  $log t1
  inner join app mdata t2 on t1.appid = t2.id
where
  $filter
  and logid to int(logid) not in (4, 7, 14, 20)
group by
  risk,
  f behavior
order by
  risk desc,
  number desc
```

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

Log Category

```
select
 risk as d risk,
 behavior as d behavior,
 t2.id,
 t2.name,
 t2.app cat,
 t2.technology,
 sum(
   coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
 ) as bandwidth,
 count(*) as sessions
from
  $log t1
 inner join app_mdata t2 on t1.appid = t2.id
where
 $filter
 and logid to int(logid) not in (4, 7, 14, 20)
 and behavior is not null
group by
 t2.id
order by
 risk desc,
  sessions desc
```

Dataset Name	Description	Log Category
Apprisk-Ctrl-Severe-High-Risk- Application	Severe and high risk applications	traffic
select appcat, count(distinct app) as total_num		
<pre>count(distinct app) as total_num from ###(select appcat, app from \$log where \$filter and app is not null and appcat is not null and logid_to_int(logid) not in (4, 7, 14, 20) and apprisk in ('critical', 'high') group by appcat, app)### t group by appcat order by total_num desc</pre>		

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 total_num 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
<pre>select attack, attackid, vuln_type, cve, severity_number,</pre>		
<pre>count(distinct dstip) count(distinct srcip)</pre>		

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

pprisk-Ctrl-Breakdown-Of-High-Risk- pplication elect appcat, count(distinct app)		traffic
appcat, count(distinct app)		
om	as luldi IIUIII	
all and appcat is not , 20) and apprisk in	<pre>_ pp from \$log where \$filter a null and logid_to_int(logid ('critical', 'high') group cat order by total_num desc</pre>) not in (4, 7,
ataset Name	Description	Log Category
pprisk-Ctrl-Top-20-High-Risk- pplication	Application risk high risk application	traffic
fna(`unauthuser`), i sentbyte, 0)+coalesce ons from \$log where 14, 20) group by lo inner join app_mdat		<pre>sum(coalesce count(*) as se id) not in (4, width desc)### ame) where risk</pre>
ataset Name	Description	Log Category

select behavior,

```
round(
   sum(total_num)* 100 / sum(
      sum(total_num)
   ) over (),
   2
) as percentage
```

###(select (case when lower(appcat)='botnet' then 'malicious'
when lower(appcat)='remote.access' then 'tunneling' when lower
(appcat) in ('storage.backup', 'video/audio') then 'bandwidth-consuming' when lower(appcat)='p2p' then 'peer-to-peer' when lower
(appcat)='proxy' then 'proxy' end) as behavior, count(*) as total_
num from \$log where \$filter and lower(appcat) in ('botnet',
'remote.access', 'storage.backup', 'video/audio', 'p2p', 'proxy')
and logid_to_int(logid) not in (4, 7, 14, 20) and apprisk in
('critical', 'high') group by appcat)### t group by behavior order
by percentage desc

Dataset Name	Description	Log Category
Apprisk-Ctrl-Key-Application-Crossing- The-Network	Key Application Crossing The Network	traffic
select		
risk as d_risk,		
count(distinct user_s	rc) as users,	
id,		
name,		
app_cat,		
technology,		
sum(bandwidth) as ban		
sum(sessions) as sessions		
from		
	<pre>sce(nullifna(`user`), nullif</pre>	
	<pre>p`)) as user_src, sum(coales</pre>	
0)+coalesce(rcvdbyte, 0)) as bandwidth, count(*) as	sessions from
\$log where \$filter and	<pre>logid_to_int(logid) not in (</pre>	(4, 7, 14, 20)
group by app, user_src order by bandwidth desc)### t1 inner join		
app_mdata t2 on t1.app=t2.name group by id, app, app_cat, tech-		
nology, risk order by b	andwidth desc	
Defeest News	Description	

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

```
select
   appcat,
   sum(
      coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
   ) as bandwidth
from
   $log
where
   $filter
   and logid_to_int(logid) not in (4, 7, 14, 20)
   and nullifna(appcat) is not null
group by
   appcat
order by
   bandwidth desc
```

Dataset Name	Description	Log Category
Apprisk-Ctrl-Category-Breakdown-By- Bandwidth	Category breakdown of all applications, sorted by bandwidth	traffic
<pre>authuser`), ipstr(`src: coalesce(rcvdbyte, 0)) \$log where \$filter and</pre>	r) as user_num, ndwidth, num_session pp, coalesce(nullifna(`user`), ip`)) as f_user, sum(coalesce(as bandwidth, count(*) as num logid_to_int(logid) not in (4 s not null group by appcat, ap	sentbyte, 0)+- _session from , 7, 14, 20)
Dataset Name	Description	Log Category
Apprisk-Ctrl-Top-Web-Applications-by- Bandwidth	Top 25 Web Categories by Bandwidtih	traffic

select
 d_risk,
 id,
 name,

```
technology,
  count(distinct f_user) as user_num,
  sum(bandwidth) as bandwidth,
  sum(num_session) as num_session
from
```

```
###(select risk as d_risk, t2.id, t2.name, t2.technology,
coalesce(nullifna(t1.`user`), nullifna(t1.`unauthuser`), ipstr
(t1.`srcip`)) as f_user, sum(coalesce(sentbyte, 0)+coalesce(rcvd-
byte, 0)) as bandwidth, count(*) as num_session from $log t1 inner
join app_mdata t2 on t1.appid=t2.id where $filter and logid_to_int
(logid) not in (4, 7, 14, 20) and nullifna(app) is not null and
service in ('80/tcp', '443/tcp', 'HTTP', 'HTTPS', 'http', 'https')
group by risk, t2.id, t2.name, t2.technology, f_user)### t group
by d_risk, id, name, technology order by bandwidth desc
```

```
Dataset NameDescriptionLog CategoryApprisk-Ctrl-Top-Web-Categories-<br/>VisitedTop 25 Web Categories Visitedtrafficselect<br/>catdesc,<br/>count (distinct f_user) as user_num,<br/>sum (sessions) as sessions,<br/>sum (bandwidth) as bandwidthfrom<br/>### (select catdesc, coalesce (nullifna (`user`), nullifna (`un-<br/>authuser`), 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 logid_to_ int(logid) not in (4, 7, 14, 20) and (countweb>0 or ((logver is null or logver<52) and (hostname is not null or utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'scriptfilter'))) group by f_user, catdesc order by sessions desc)### t group by catdesc order by sessions desc

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

```
(
```

```
case when lower(appcat) = 'botnet' then 'Botnet C&C' else (
     case when virus_s like 'Riskware%' then 'Spyware' when
virus_s like 'Adware%' then 'Adware' else 'Virus' end
   ) end
   ) as malware_type,
   appid,
   app,
   count(distinct dstip) as victims,
   count(distinct srcip) as source,
   sum(total_num) as total_num
from
   (
```

###(select app as virus_s, appcat, appid, app, dstip, srcip, count(*) as total_num from \$log-traffic where \$filter and logid_ to_int(logid) not in (4, 7, 14, 20) 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 logid_to_int(logid) not in (4, 7, 14, 20) and virus is not null 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
<pre>select virus_s, appid, app, count(distinct dstip) count(distinct srcip)</pre>		
<pre>appid, app, dstip, srci \$filter and logid_to_in like '%PossibleThreat.S</pre>	_ cing_to_array(virus, ',')) as vir p, count(*) as total_num from \$2 at(logid) not in (4, 7, 14, 20) a B8%' group by virus_s, dstip, sro s like '%PossibleThreat.SB%' gro	log where and virus cip, appid,

Dataset Name	Description	Log Category
Apprisk-Ctrl-Files-Analyzed-By- FortiCloud-Sandbox	Files analyzed by FortiCloud Sandbox	virus
<pre>select \$DAY_OF_MONTH as do: count(*) as total_n from \$log</pre>		
where \$filter and nullifna(filena:	me) is not null	
and logid_to_int(lo group by dom order by dom		

Dataset Name	Description	Log Category
Apprisk-Ctrl-Malicious-Files-Detected- By-FortiCloud-Sandbox	Files detected by FortiCloud Sandbox	virus

```
select
filename,
analyticscksum,
count(distinct dstip) as victims,
count(distinct srcip) as source
```

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

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

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

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

	Dataset Name	Description	Log Category
	appctrl-Top-Blocked-SIP-Callers	Appctrl top blocked SIP callers	app-ctrl
02	select srcname as caller, count(*) as totalnum		

```
from
  $log
where
  $filter
  and srcname is not null
  and lower(appcat) = 'voip'
  and app = 'sip'
  and action = 'block'
group by
  caller
order by
  totalnum desc
```

```
Dataset Name
                          Description
                                                               Log Category
security-Top20-High-Risk-Application-
                          High risk application in use
                                                               traffic
In-Use
select
  d risk,
  count(distinct f user) as users,
  name,
  app cat,
  technology,
  sum(bandwidth) as bandwidth,
  sum(sessions) as sessions
from
  ###(select risk as d risk, coalesce(nullifna(t1.`user`), nul-
lifna(t1.`unauthuser`), ipstr(t1.`srcip`)) as f user, t2.name,
t2.app cat, t2.technology, sum(coalesce(sentbyte, 0)+coalesce(rcvd-
byte, 0)) as bandwidth, count(*) as sessions from $log t1 inner
join app mdata t2 on t1.appid=t2.id where $filter and risk>='4'
and logid to int(logid) not in (4, 7, 14, 20) group by f user, t2.-
name, t2.app cat, t2.technology, risk)### t group by d risk, name,
app cat, technology order by d risk desc, sessions desc
Dataset Name
                          Description
                                                               Log Category
```

security-High-Risk-Application-By- Category	High risk application by category	traffic
<pre>select app_cat, count(distinct app) a</pre>	as total num	

###(select app_cat, app from \$log t1 inner join app_mdata t2 on t1.appid=t2.id where \$filter and risk>='4' and logid_to_int(logid) not in (4, 7, 14, 20) group by app_cat, app)### t group by app_cat order by total num desc

Description	Log Category
Application risk application usage by category	traffic
0)+ coalesce(rcvdbyte, 0) d) not in (4, 7, 14, 20) is not null	
	Application risk application usage by category 0) + coalesce(rcvdbyte, 0) d) not in (4, 7, 14, 20)

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,</pre>			
count(distinct f user) as user num,			
sum(bandwidth) as bandwidth,			
sum(num_session) as num_session			
from			
<pre>###(select appcat, app, coalesce(nullifna(`user`), nullifna(`un- authuser`), ipstr(`srcip`)) as f_user, sum(coalesce(sentbyte, 0)+- coalesce(rcvdbyte, 0)) as bandwidth, count(*) as num_session from \$log where \$filter and logid_to_int(logid) not in (4, 7, 14, 20)</pre>			

and nullifna(appcat) is not null group by appcat, app, f_user)### t group by appcat order by bandwidth desc

Dataset Name	Description	Log Category
security-Top25-Web-Applications-By- Bandwidth	Top Web Applications by Bandwidtih	traffic
<pre>coalesce(nullifna(t1.`u (t1.`srcip`)) as f_user byte, 0)) as bandwidth, join app_mdata t2 on t1 (logid) not in (4, 7, 1 service in ('80/tcp', ' group by risk, t2.app_c</pre>	dwidth,	<pre>ser`), ipstr +coalesce(rcvd- om \$log t1 inner and logid_to_int not null and 'http', 'https') f_user)### t</pre>
Dataset Name	Description	Log Category
Security-Top25-Web-Categories- Visited	Top 25 Web Categories Visited	traffic
<pre>authuser`), ipstr(`srci (coalesce(sentbyte, 0)+ \$log-traffic where \$fil int(logid) not in (4, 7</pre>	ions,	sessions, sum andwidth from and logid_to_ c ((logver is

('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** Log Category Description security-Top25-Malware-Virus-Botnettraffic Malware: viruses, Bots, Spyware/Adware Spyware select virus s as virus, (case when lower(appcat) = 'botnet' then 'Botnet C&C' else (case when virus s like 'Riskware%' then 'Spyware' when virus s like 'Adware%' then 'Adware' else 'Virus' end) end) as malware type, count(distinct dstip) as victims, count(distinct srcip) as source, sum(total num) as total num from (

###(select app as virus_s, appcat, dstip, srcip, count(*) as total_num from \$log-traffic where \$filter and logid_to_int(logid) not in (4, 7, 14, 20) 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 logid_to_int(logid) not in (4, 7, 14, 20) and virus is not null 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
<pre>select virus, max(virusid_s) as vin malware_type, count(distinct dstip) count(distinct srcip) sum(total_num) as tot</pre>	as victims, as source,	

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

Dataset Name	Description	Log Category
security-Top10-Malware-Botnet	Malware: Botnet	appctrl
select		
app,		
appid,		
malware_type,		
count(distinct dstip)	as victims,	
count(distinct srcip)	as source,	
<pre>sum(total_num) as tot</pre>	al_num	
from		
###(select app, appic	l, cast('Botnet C&C' as char(3	2)) as mal-
<pre>ware_type, srcip, dstip</pre>	, count(*) as total_num from	\$log where
\$filter and lower(appca	<pre>it)='botnet' and nullifna(app)</pre>	is not null
group by app, appid, ma	lware_type, srcip, dstip orde	er by total_num
desc)### t group by app	o, appid, malware_type order b	y total_num
desc		

Dataset Name	Description	Log Category
security-Top10-Victims-of-Malware	Victims of Malware	virus
<pre>select coalesce(nullifna(`user`), nullifna(`unauthuse: ipstr(`srcip`)) as user_src, virus as malware, count(*) as total_num from \$log where \$filter</pre>	r`),	

```
and virus is not null
group by
   user_src,
   malware
order by
   total_num desc
```

Dataset Name	Description	Log Category
security-Top10-Victims-of-Phishing- Site	Victims of Phishing Site	webfilter
<pre>select coalesce(nullifna(`user`), nullifna(`unauthuse ipstr(`srcip`)) as user_src, (lower(service) ') as phishing_site, count(*) as total_num from \$log where \$filter and lower(service) in and hostname is not n and cat in (26, 61)</pre>	<pre>://' hostname url ('http', 'https')</pre>	
<pre>group by user_src, phishing_site order by total_num desc</pre>		

Dataset Name	Description	Log Category
security-Top25-Malicious-Phishing- Sites	Malicious Phishing Site	webfilter
<pre>select phishing_site, count(distinct dstip) count(distinct srcip)</pre>		

```
sum(total) as total_num
from
    ###(select (lower(service) || '://' || hostname || url) as phish-
ing_site, dstip, srcip, count(*) as total from $log where $filter
and lower(service) in ('http', 'https') and hostname is not null
and cat in (26, 61) group by phishing_site, dstip, srcip order by
total desc)### t group by phishing_site order by total_num desc
```

Dataset Name	Description	Log Category
security-Application-Vulnerability	Application vulnerabilities discovered	attack
select		
attack,		
attackid,		
vuln_type,		
cve,		
severity_number,		
count(distinct dstip) as victims,	
count(distinct srcip) as sources,	
sum(totalnum) as tot	calnum	
from		
###(select attack, a	attackid, vuln_type, t2.cve, (case when t1
severity='critical' th	nen 5 when t1.severity='high'	then 4 when t1.
_	n 3 when tl.severity='low' the	
_	en 1 else 0 end) as severity_n	_
_	alnum from \$log t1 left join	
	os_mdata) t2 on t1.attack=t2.n	
	ck) is not null and t1.severit	-
	ckid, vuln_type, t2.cve, t1.se	
<pre>srcip)### t group by</pre>	attack, attackid, vuln_type,	severity_num-

Dataset Name	Description	Log Category
security-Files-Analyzed-By-FortiCloud- Sandbox	Files analyzed by FortiCloud Sandbox	virus
<pre>select \$day_of_week as dow, count(*) as total_num from \$log where</pre>		

ber, cve order by severity number desc, totalnum desc

```
$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
<pre>dstip, srcip, count(*) logid_to_int(logid) nc</pre>	<pre>b) as source, btal_num cring_to_array(virus, ',')) as as total_num from \$log where bt in (4, 7, 14, 20) and virus by virus_s, dstip, srcip, app)</pre>	\$filter and like '%Poss-
Dataset Name	Description	Log Category
security-Data-Loss-Incidents-By- Severity	Data loss incidents summary by severity	dlp

```
select
initcap(severity : :text) as s_severity,
count(*) as total_num
from
   $log
where
   $filter
   and severity is not null
group by
   s_severity
order by
   total_num desc
```

Dataset Name	Description		Log Category
security-Data-Loss-Files-By-Service	Data Lass Files By Service		dlp
<pre>select filename, (case direction when then Unleady and</pre>	'incoming' then	'Download' wh	nen 'outgoing'
<pre>then 'Upload' end) as action, max(filesize) as files service</pre>	size,		
from			
\$log			
where \$filter and filesize is not not	ull		
<pre>group by filename, direction, service order by filesize desc</pre>			

Dataset Name	Description	Log Category
security-Endpoint-Security-Events- Summary	Endpoint Security Events summary	fct-traffic
'webfilter' then 'Malic then 'Risk applications		n 'appfirewall' s incidents'

order by total_num desc

Dataset Name	Description	Log Category
security-Top-Endpoing-Running-High- Risk-Application	Endpoints Running High Risk Application	fct-traffic
<pre>select coalesce(nullifna(`user`), ipstr(`srcip`), 'Unknown') as f_user, coalesce(nullifna(hostname), 'Unknown') as host_name, threat as app, t2.app_cat as appcat, risk as d_risk from \$log t1 inner join app_mdata where \$filter and utmevent = 'appfi and risk & gt;= '4'</pre>	t2 on t1.threat = t2.name rewall'	
<pre>group by f_user, host_name, t1.threat, t2.app_cat, t2.risk order by risk desc</pre>		

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

```
nullifna(`user`),
    ipstr(`deviceip`),
    'Unknown'
  ) as f user,
  coalesce(
    nullifna(hostname),
    'Unknown'
  ) as host_name,
  virus,
  file
from
  $log
where
  $filter
  and clientfeature = 'av'
  and virus is not null
group by
  f user,
  host name,
  virus,
  file
```

Dataset Name	Description	Log Category
security-Top-Endpoints-With-Web- Violateions	Endpoints With Web Violations	fct-traffic
<pre>select f_user, host_name, remotename, sum(total_num) as to from ###(select coalesce)</pre>	otal_num (nullifna(`user`), ipstr(`sr	cip`)) as f_user,
<pre>count(*) as total_num ='webfilter' and remot group by f_user, host_</pre>	<pre>iname), 'Unknown') as host_n from \$log where \$filter and cename is not null and utmac name, remotename order by t st_name, remotename order by</pre>	utmevent- tion='blocked' otal_num desc)###
Datasat Nama	Description	Log Catagory

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

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

Dataset Name	Description	Log Category
content-Count-Total-SCCP-Call- Registrations-by-Hour-of-Day	Content count total SCCP call registrations by hour of day	content
<pre>select \$hour_of_day as hours count(*) as totalnum from \$log where \$filter and proto = 'sccp' and kind = 'register' group by hourstamp order by hourstamp</pre>	tamp,	

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 hours sum(duration) as sccp from \$log where \$filter</pre>	-	
```
and proto = 'sccp'
 and kind = 'call-info'
 and status = 'end'
group by
 hourstamp
order by
 hourstamp
```

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

Dataset Name

Dataset Name	Description	Log Category
content-Count-Total-SIP-Call- Registrations-by-Hour-of-Day	Content count total SIP call registrations by hour of day	content
select		
<pre>\$hour_of_day as hours</pre>	tamp,	
count(*) as totalnum		
from		
\$log		
where		
\$filter		
and proto = 'sip'		
and kind = 'register'		
group by		
hourstamp		

order by

hourstamp

Dataset Name	Description	Log Category
content-Count-Total-SIP-Calls-per- Status	Content count total SIP calls per status	content
<pre>select status, count(*) as totalnum from \$log where \$filter and proto = 'sip' and kind = 'call' group by status order by totalnum desc</pre>		
Dataset Name	Description	Log Category
content-Dist-Total-SIP-Calls-by- Duration	Content dist total SIP calls by duration	content
& lt; 600 then 'LESS_TE	& lt; 60 then 'LESS_ONE_MIN' who N_MIN' when duration & lt; 3600 & gt;= 3600 then 'MORE_ONE_HOU	then 'LESS_

order by totalnum desc

Dataset Name	Description	Log Category
Botnet-Activity-By-Sources	Botnet activity by sources	traffic
select		
app,		
coalesce(
nullifna(`user`)		
nullifna(`unauth	user`),	
ipstr(`srcip`)		
) as user_src,		
count(*) as events		
from		
\$log		
where		
\$filter		
and logid_to_int(1	ogid) not in (4, 7, 14, 20)	
and appcat = 'Botn	et'	
and nullifna(app)	is not null	
group by		
app,		
user_src		
order by		
events desc		

Dataset Name	Description	Log Category
Botnet-Infected-Hosts	Botnet infected hosts	traffic
<pre>select coalesce(nullifna(`user`) nullifna(`unauth ipstr(`srcip`)) as user_src, devtype, coalesce(srcname, count(*) as events</pre>	srcmac) as host_mac,	
from \$log		

where

```
$filter
and logid_to_int(logid) not in (4, 7, 14, 20)
and appcat = 'Botnet'
group by
user_src,
devtype,
host_mac
order by
events desc
```

Dataset Name	Description	Log Category
Detected-Botnet	Detected botnet	traffic
select		
app,		
count(*) as ev	rents	
from		
\$log		
where		
\$filter		
and logid_to_i	nt(logid) not in (4, 7, 14, 20)	
and appcat = '	Botnet'	
and nullifna(a	pp) is not null	
group by		
app		
order by		
events desc		

Dataset Name	Description	Log Category
Botnet-Sources	Botnet sources	traffic
<pre>count(*) as eve from</pre>	stname) as domain, ents	
<pre>\$log where \$filter and logid_to_ir and appcat = 'H and dstip is no</pre>		0)

```
group by
dstip,
domain
order by
events desc
```

Dataset Name	Description		Log Category
Botnet-Victims	Botnet victims		traffic
<pre>select coalesce(nullifna(`use nullifna(`una ipstr(`srcip`) as user_src, count(*) as eve from \$log where \$filter and logid_to_in and appcat = 'B and srcip is no</pre>	uthuser`),) ents et(logid) not in (4, 7 eotnet'	7, 14, 20)	
group by			
—			
-			
<pre>nullifna(`una ipstr(`srcip`) as user_src, count(*) as eve from \$log where \$filter and logid_to_in and appcat = 'B and srcip is no</pre>	uthuser`),) ents et(logid) not in (4, 7 eotnet'	7, 14, 20)	

Dataset Name	Description	Log Category
Botnet-Timeline	Botnet timeline	traffic

select

```
$flex_datetime(timestamp) as hodex,
   sum(events) as events
from
```

```
(
```

###(select \$flex_timestamp as timestamp, count(*) as events from \$log where \$filter and logid_to_int(logid) not in (4, 7, 14, 20) 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)###) t group by hodex order by hodex

Dataset Name	Description	Log Category
Application-Session-History	Application session history	traffic
<pre>select \$flex_timescale(tim sum(counter) as cou</pre>	-	
from		
###(select \$flex_ti	mestamp as timestamp, co	ount(*) as counter
from \$log where \$filt	er and logid_to_int(logi	d) not in (4, 7, 14,
20) group by timestam	p order by timestamp des	c)### t group by
hodex order by hodex		

Dataset Name Description Log Category Application-Usage-List Detailed application usage traffic select appid, app, appcat, (case when (utmaction in ('block', 'blocked') or action = 'deny') then 'Blocked' else 'Allowed' end) as custaction, sum(coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)) as bandwidth, count(*) as num session from \$loa where \$filter and logid to int(logid) not in (4, 7, 14, 20) and nullifna(app) is not null and policyid != 0group by appid, app,

```
appcat,
custaction
order by
bandwidth desc
```

Dataset Name	Description	Log Category
PCI-DSS-Compliance-Summary	PCI DSS Compliance Summary	event
<pre>select status, num_reason as require cast(</pre>		
<pre>num_reason * 100.0 sum(num_reason) c) as decimal(18, 2)) as percent</pre>	over()	
from		
(
select		
(
case when fail_	_count & gt; 0 then 'Non-Complian	nt' else
'Compliant' end		
) as status,		
count(distinct re	eason) as num_reason	
from		
(
select		
ftnt_pci_id,		
(
—	ount) over (partition by ftnt_pc:	i_id)
) as fail_cou	int,	
reason		
from		
<pre>else 0 end) as fail_cou mdata t2 on t1.reason=t type='compliance-check'</pre>	<pre>cnt_pci_id, (case when result='fa unt, reason from \$log t1 inner jo c2.ftnt_id where \$filter and sub- ' group by ftnt_pci_id, result, p</pre>	oin pci_dss_ -
t) t group by status) t	t order by status	

Dataset Name	Description	Log Category
PCI-DSS-Non-Compliant- Requirements-By-Severity	PCI DSS Non-Compliant Requirements by Severity	event

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

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

Dataset Name	Description	Log Category
PCI-DSS-Compliant-Requirements-By- Severity	PCI DSS Compliant Requirements by Severity	event
<pre>with query as (select * from (select ftnt_pci_id, severity, (sum(fail_count reason from</pre>	t) over (partition by ftnt_pci_i	d)
###(select ftnt	_pci_id, t2.severity, (case when	

Fortinet Technologies Inc.

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

Dataset Name	Description	Log Category
PCI-DSS-Fortinet-Security-Best- Practice-Summary	PCI DSS Fortinet Security Best Practice Summary	event
select		
status,		
num_reason as pract.	ices,	
cast(
num_reason * 100.	0 / (
<pre>sum(num_reason)</pre>	over()	
) as decimal(18,	2)	
) as percent		
from		
(
select		
(
case when res	ult = 'fail' then 'Failed' else ':	Passed' end
) as status,		
count(distinct)	reason) as num_reason	
from		
###(select resu	lt, reason from \$log where \$filte:	r and sub-
	k' and result in ('fail','pass') (up by status) t order by status de	

Dataset Name	Description	Log Category
PCI-DSS-Failed-Fortinet-Security- Best-Practices-By-Severity	PCI DSS Failed Fortinet Security Best Practices by Severity	event
select		
status,		
num_reason as practices,		
cast (

```
num_reason * 100.0 /(
    sum(num_reason) over()
    ) as decimal(18, 2)
) as percent
from
    (
    select
    initcap(status) as status,
    count(distinct reason) as num_reason
    from
        ###(select status, reason from $log where $filter and sub-
type='compliance-check' and result='fail' group by status,
reason)### t group by status) t order by status
```

Dataset Name	Description	Log Category
PCI-DSS-Passed-Fortinet-Security- Best-Practices-By-Severity	PCI DSS Passed Fortinet Security Best Practices by Severity	event
<pre>select status, num_reason as practic cast(num_reason * 100.0 sum(num_reason) o) as decimal(18, 2)) as percent from (select initcap(status) a count(distinct re from</pre>	es, /(ver()	er and sub-
<pre>type='compliance-check'</pre>	and result='pass' group by sta tatus) t order by status	

Dataset Name	Description	Log Category
PCI-DSS-Requirements-Compliance- Details	PCI DSS Requirements Compliance Details	event

select

ftnt_pci_id,

```
left(
    string agg(distinct ftnt id, ','),
    120
  ) as practice,
  (
    case when sum(fail count)& gt; 0 then 'Non-Compliant' else
'Compliant' end
  ) as compliance,
  pci requirement
from
  ###(select ftnt pci id, ftnt id, (case when result='fail' then 1
else 0 end) as fail count, pci requirement from $log t1 inner join
pci dss mdata t2 on t1.reason=t2.ftnt id where $filter and sub-
type='compliance-check' group by ftnt_pci_id, ftnt_id, result,
pci requirement)### t group by ftnt pci id, pci requirement order
by ftnt pci id
```

Dataset Name	Description	Log Category
PCI-DSS-Fortinet-Security-Best- Practice-Details	PCI DSS Fortinet Security Best Practice Details	event
select		
reason as ftnt_id, msg,		
initcap(status) as module	status,	
from		
\$log		
where		
\$filter		
and subtype = 'comp	liance-check'	
group by		
reason,		
status,		
module,		
msg		
order by		
ftnt_id		
Dataset Name	Description	Log Category

Dataset Name	Description	Log Calego
DLP-Email-Activity-Details	Email DLP Violations Summary	dlp

```
select
 from itime(itime) as timestamp,
  `from` as sender,
  `to` as receiver,
 regexp replace(filename, '.*/', '') as filename,
 filesize,
 profile,
 action,
 direction
from
  $loq
where
 $filter
 and (
    service in (
      'smtp', 'SMTP', '25/tcp', '587/tcp',
      'smtps', 'SMTPS', '465/tcp'
    )
    or service in (
      'pop3', 'POP3', '110/tcp', 'imap',
      'IMAP', '143/tcp', 'imaps', 'IMAPS',
      '993/tcp', 'pop3s', 'POP3S', '995/tcp'
   )
  )
order by
  timestamp desc
```

Dataset Name	Description	Log Category
Email-DLP-Chart	Email DLP Activity Summary	dlp
<pre>select profile, count(*) as total_num from \$log where \$filter and (service in ('smtp', 'SMTP', ' 'smtps', 'SMTPS',)</pre>	25/tcp', '587/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
<pre>select from_itime(itime) srcip, dstip, hostname, profile, filename, filesize, action, direction</pre>	as timestamp,	
<pre>from \$log where \$filter and lower(service order by timestamp desc</pre>) in ('http', 'https')	

Dataset Name	Description	Log Category
Web-DLP-Chart	Web DLP Activity Summary	dlp
<pre>select profile, count(*) as from \$log where \$filter</pre>	s total_num	

```
and lower(service) in ('http', 'https')
group by
profile
order by
total_num desc
```

Dataset Name	Description	Log Category
DLP-FTP-Activity-Details	Web DLP Violations Summary	dlp
select		
<pre>from_itime(itime)</pre>	as timestamp,	
srcip,		
dstip, filename,		
profile,		
filesize,		
action,		
direction		
from		
\$log		
where		
\$filter		
	in ('ftp', 'ftps')	
order by		
timestamp desc		

Dataset Name	Description	Log Category
FTP-DLP-Chart	FTP DLP Activity Summary	dlp
<pre>select profile, count(*) as total_num from \$log where \$filter and lower(service) in group by profile order by total_num desc</pre>		

Dataset Name	Description	Log Category
top-users-by-browsetime	Top Users by website browsetime	traffic
select		
user src,		
domain,		
ebtr_value(
ebtr_agg_flat(b	rowsetime),	
null,		
\$timespan		
) as browsetime		
from		
###(select user_s)	rc, domain, ebtr_agg_flat(browse	etime) as brow-
setime from (select	<pre>coalesce(nullifna(`user`), ipst</pre>	tr(`srcip`)) as
user_src, coalesce(r	nullifna(hostname), ipstr(`dsti	p`)) as domain,
<pre>ebtr_agg_flat(\$brows</pre>	se_time) as browsetime from \$log	g where \$filter
and \$browse_time is	not null group by user_src, dom	main) t group by
user src, domain ord	der by ebtr value(ebtr agg flat	(browsetime),

user_src, domain order by ebtr_value(ebtr_agg_flat(browsetime), null, null) desc)### t group by user_src, domain order by browsetime desc

Dataset Name	Description	Log Category
wifi-usage-by-hour-authenticated	Wifi Usage by Hour - Authenticated	event
and subtype='wireless	ac) as totalnum _DAY as hod, stamac from \$10 ' and action='client-authen up by hod order by hod	-
Dataset Name	Description	Log Category
wifi-usage-authenticated-timeline	Wifi Usage Timeline - Authenticated	event
select \$flex timescale(time		

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

Dataset Name	Description	Log Category
app-top-user-by-bandwidth	Top 10 Applications Bandwidth by User Drilldown	traffic
select		
app,		
coalesce(
<pre>nullifna(`user`),</pre>		
nullifna(`unauthuse	r`),	
ipstr(`srcip`)		
) as user_src,		
sum(
coalesce(`sentbyte`	<pre>, 0)+ coalesce(`rcvdbyte`, 0)</pre>	
) as bandwidth		
from		
\$log		
where		
\$filter		
	d) not in (4, 7, 14, 20)	
and nullifna(app) is	not null	
group by		
app,		
user_src		
order by		
bandwidth desc		

Dataset Name	Description	Log Category
app-top-user-by-session	Top 10 Application Sessions by User Drilldown	traffic
select		
app,		
coalesce(
nullifna(`user`)	,	
nullifna(`unauth	user`),	
ipstr(`srcip`)		
) as user_src,		
count(*) as session	ns	

```
from
$log
```

-

```
$filter
and logid_to_int(logid) not in (4, 7, 14, 20)
and nullifna(app) is not null
group by
app,
user_src
order by
sessions desc
```

Dataset Name	Description	Log Category
traffic-Interface-Bandwidth-Usage	Interface Bandwidth Usage	traffic
<pre>with qry as (select dom as dom_s, devid as devid_s, vd as vd_s, srcintf, dstintf, total_sent, total_rcvd</pre>		
total_rcvd from		

###(select \$DAY OF MONTH as dom, devid, vd, srcintf, dstintf, sum(coalesce(sentbyte, 0)) as total sent, sum(coalesce(rcvdbyte, 0)) as total rcvd, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as total from \$log where \$filter and logid to int(logid) not in (4, 7, 14, 20) and nullifna(srcintf) is not null and nullifna (dstintf) is not null group by dom, devid, vd, srcintf, dstintf having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by total desc)### t) select dom, unnest(array['download', 'upload']) as type, unnest(array[sum(download), sum(upload)]) as bandwidth from (select coalesce(t1.dom s, t2.dom s) as dom, coalesce (t1.devid_s, t2.devid_s) as devid, coalesce(t1.vd s, t2.vd s) as vd, coalesce(t1.srcintf, t2.dstintf) as intf, sum(coalesce (t1.total sent, 0)+coalesce(t2.total rcvd, 0)) as download, sum (coalesce(t2.total sent, 0)+coalesce(t1.total rcvd, 0)) as upload from qry t1 full join qry t2 on t1.dom s=t2.dom s and t1.srcintf=t2.dstintf group by dom, devid, vd, intf) t where \$filterdrilldown group by dom order by dom

Dataset Name	Description	Log Category
ctap-SB-Files-Needing-Inspection-vs- Others	Files Needing Inspection vs Others	virus
	e', 'jar', 'msi', 'vbs',	
'rar', 'cab', 'dc 'xlsx', 'ppt', 'p 'lnk', 'js'	p', 'lzw', 'tar', pc', 'docx', 'xls', pptx', 'pdf', 'swf', c File Types' else 'Excluded	d Files' end
) as files, sum(total_num) as tot		
num from \$log where \$fi	ext(filename) as suffix, co lter and dtype='fortisandbo group by suffix order by to by total_num desc	ox' and nullifna
Dataset Name	Description	Log Category
ctap-SB-Breakdown-of-File-Types	Breakdown of File Types	virus
<pre>select (case when suffix in 'exe', 'msi', 'up 'dll', 'ps1', 'ja</pre>	x', 'vbs', 'bat', 'cmd',	
) then 'Executable PDF' when suffix in ('s 'doc', 'docx', 'r 'dotm', 'dot') then 'Microsoft W	Files' when suffix in ('pdf wf') then 'Adobe Flash' whe otf', 'dotx', 'docm', Yord' when suffix in (altx', 'xlsm', 'xlsb',	

```
then 'Web Files' when suffix in (
      'cab', 'tgz', 'z', '7z', 'tar', 'lzh',
      'kgb', 'rar', 'zip', 'gz', 'xz', 'bz2'
    ) then 'Archive Files' when suffix in ('apk') then 'Android
Files' else 'Others' end
  ) as filetype,
  sum(total num) as total num
from
  ###(select file name ext(filename) as suffix, count(*) as total
num from $log where $filter and dtype='fortisandbox' and nullifna
(filename) is not null group by suffix order by total num desc) ###
t group by filetype order by total num desc
Dataset Name
                         Description
                                                            Log Category
ctap-SB-Top-Sandbox-Malicious-Exes
                                                            virus
select
  (
    case fsaverdict when '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
  $loq
where
  $filter
  and dtype = 'fortisandbox'
  and file name ext(filename) = 'exe'
  and fsaverdict not in ('clean', 'submission failed')
group by
  filename,
  risk,
  service
order by
  risk desc,
  total num desc,
  filename
```

Dataset Name	Description	Log Category
ctap-SB-Sources-of-Sandbox- Discovered-Malware	Sources of Sandbox Discovered Malware	virus
<pre>select srcip, count(*) as total_num</pre>	n	
from		
\$log		
where		
\$filter		
and dtype = 'fortisar	ndbox'	
and nullifna(filename	e) is not null	
and fsaverdict not ir	n ('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_s	rc) as users,	
id,		
name,		
app_cat,		
technology,		
sum(bandwidth) as ban	dwidth,	
sum(sessions) as sess	ions	
from		
###(select lower(app)	as lowapp, coalesce(nullif	na(`user`), nul-
lifna(`unauthuser`), ip	<pre>str(`srcip`)) as user_src, s</pre>	sum(coalesce
(sentbyte, 0)+coalesce(<pre>rcvdbyte, 0)) as bandwidth,</pre>	count(*) as ses-
sions from \$log where \$	filter and logid_to_int(log:	id) not in (4,
7, 14, 20) group by low	app, user_src order by bandu	width desc)###
t1 inner join app_mdata	t2 on t1.lowapp=lower(t2.na	ame) where risk>-
='4' group by id, name,	app_cat, technology, risk o	order by d_risk

desc, sessions desc

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

```
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
case when malware	cat)= 'botnet' then 'Botnet C&C' like 'Riskware%' then 'Spyware' n 'Adware' else 'Virus' end as victims,	
<pre>count(distinct srcip) sum(total_num) as tot</pre>		

from

(

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

Dataset Name	Description	Log Category
ctap-App-Risk-Reputation-Top- Devices-By-Scores	Reputation Top Devices By-Scores	traffic
select		
<pre>coalesce(nullifna(`srcname ipstr(`srcip`), nullifna(`srcmac`)) as dev_src, sum(crscore % 65536)</pre>)	
from		
\$log		
where		
\$filter		
and logid_to_int(log and crscore is not a	gid) not in (4, 7, 14, 20) 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 i	n ('80/tcp', 'HTTP', 'http') then	'HTTP'

```
else 'HTTPS' end
  ) as service,
  sum(
    coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
  ) as bandwidth
from
  $loq
where
  $filter
  and logid to int(logid) not in (4, 7, 14, 20)
  and nullifna(app) is not null
  and service in (
    '80/tcp', '443/tcp', 'HTTP', 'HTTPS',
    'http', 'https'
  )
group by
  service
having
  sum(
    coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
  )& gt; 0
order by
  bandwidth desc
```

Dataset Name	Description	Log Category
ctap-Top-Source-Countries	Top Source Countries	traffic
<pre>select srccountry, sum(coalesce(sentbyte,) as bandwidth from \$log where \$filter</pre>	0)+ coalesce(rcvdbyte, 0)	
and logid_to_int(logid) not in (4, 7, 14, 20) 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 logid_to_int (logid) not in (4, 7, 14, 20) 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(sent-
byte, 0)+coalesc	e(rcvdbyte, 0)) as bandwidth,	sum(coalesce(rcvd-
byte, 0)) as tra	ffic_in, sum(coalesce(sentbyte	e, 0)) as traffic_
<pre>out, count(*) as</pre>	sessions from \$log where \$fil	lter and logid_to_int
(logid) not in (4, 7, 14, 20) and nullifna(app) is not null group
by app_group hav	<pre>ing sum(coalesce(sentbyte, 0)+</pre>	<pre>+coalesce(rcvdbyte,</pre>
0))>0 order by b	andwidth desc)### t1 inner joi	in app_mdata t2 on
<pre>lower(t1.app_gro</pre>	up)=lower(t2.name) where app_c	cat='Cloud.IT' group
by app_group ord	er 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 lower(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 logid_to_int(logid) not in (4, 7, 14, 20) 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 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
<pre>select name as app_grou sum(bandwidth) a</pre>		
<pre>(sentbyte, 0)+coal (rcvdbyte, 0)) as traffic_out, count logid_to_int(logid null group by app (rcvdbyte, 0))>0 d mdata t2 on t1.app</pre>	r(app_group_name(app)) as app lesce(rcvdbyte, 0)) as bandwid traffic_in, sum(coalesce(sen t(*) as sessions from \$log whe d) not in (4, 7, 14, 20) and r group having sum(coalesce(sen order by bandwidth desc)### t p_group=lower(t2.name) where a er by bandwidth desc	dth, sum(coalesce tbyte, 0)) as ere \$filter and nullifna(app) is not ntbyte, 0)+coalesce 1 inner join app_
Dataset Name	Description	Log Category
ctap-Top-SocialMedia-App-By	- Top SocialMedia Applications by Bandwi	idth Usage traffic

```
select
```

Bandwidth

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

205

```
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 logid_to_int (logid) not in (4, 7, 14, 20) and nullifna(app) is not null group by app_group having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc)### t1 inner join app_mdata t2 on lower(t1.app_group)=lower(t2.name) where app_cat='Social.Media' group by app_group order by bandwidth desc

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

```
select
```

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

from

###(select app_group_name(app) as app_group, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_ out, count(*) as sessions from \$log where \$filter and logid_to_int (logid) not in (4, 7, 14, 20) 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
<pre>select app_group, sum(bandwidth) as bar sum(traffic_in) as tr sum(traffic_out) as tr </pre>	raffic_in,	

```
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 logid_to_int (logid) not in (4, 7, 14, 20) 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
<pre>select app_group, sum(bandwidth) as ba sum(traffic_in) as t sum(traffic_out) as</pre>	raffic_in, traffic_out,	
<pre>sum(sessions) as ses from</pre>	sions	
byte, 0)+coalesce(rcvd	_name(app) as app_group, sum(byte, 0)) as bandwidth, sum(c	coalesce(rcvd-
out, count(*) as sessi	n, sum(coalesce(sentbyte, 0)) ons from \$log where \$filter a 14, 20) and nullifna(app) is	and logid_to_int
0))>0 order by bandwid	<pre>m(coalesce(sentbyte, 0)+coale th desc)### t1 inner join app wer(t2.name) where app_cat='E</pre>	_mdata t2 on

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) sum(sessions) as sess: sum(bandwidth) as banc from</pre>	ions,	

###(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 logid_to_ int(logid) not in (4, 7, 14, 20) and (countweb>0 or ((logver is null or logver<52) and (hostname is not null or utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'scriptfilter'))) group by f_user, catdesc order by sessions desc)### t group by catdesc order by sessions desc

Dataset Name	Description	Log Category
ctap-App-Risk-Applications-Running- Over-HTTP	Application risk applications running over HTTP	traffic
select		
app_group_name(app) a service, count(*) as sessions,	—	
sum(
— ·	0)+ coalesce(rcvdbyte, 0)	
) as bandwidth from		
\$log		
where		
\$filter		
and logid to int(log	id) not in (4, 7, 14, 20)	
and nullifna(app) is		
and service in (
'80/tcp', '443/tcp'	', 'HTTP', 'HTTPS',	
'http', 'https'		
)		
group by		
app_group,		
service		
having		
sum(0)+ coalesce(rcvdbyte, 0)	
) & gt; 0	0) + coaresce(revabyce, 0)	
order by		
bandwidth desc		

Dataset Name	Description	Log Category
ctap-App-Risk-Web-Browsing-Activity- Hostname-Category	Application risk web browsing activity hostname category	traffic
<pre>select domain, catdesc, sum(visits) as visits</pre>		
from		

(

###(select coalesce(nullifna(hostname), ipstr(`dstip`)) as domain, catdesc, count(*) as visits from \$log-traffic where \$filter and logid_to_int(logid) not in (4, 7, 14, 20) and utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') and catdesc is not null group by domain, catdesc order by visits desc)### union all ###(select coalesce(nullifna (hostname), ipstr(`dstip`)) as domain, catdesc, count(*) as visits from \$log-webfilter where \$filter and (eventtype is null or logver>=52) and catdesc is not null group by domain, catdesc order by visits desc)###) t group by domain, catdesc order by visits desc

Dataset Name	Description	Log Category
ctap-Top-Sites-By-Browsing-Time	Traffic top sites by browsing time	traffic
<pre>select hostname, string_agg(distinct ebtr_value(ebtr_agg_flat(brow null, \$timespan) as browsetime,</pre>	catdesc, ', ') as agg_catdes setime),	с,
sum(bandwidth) as ba sum(traffic_in) as t		
<pre>sum(traffic_out) as from</pre>	traffic_out	
<pre>###(select hostname, setime, sum(bandwidth) in, sum(traffic_out) a desc, ebtr_agg_flat(\$b</pre>	<pre>catdesc, ebtr_agg_flat(brow as bandwidth, sum(traffic_i) s traffic_out from (select he rowse_time) as browsetime, so byte, 0)) as bandwidth, sum(</pre>	n) as traffic_ ostname, cat- um(coalesce(sent-

(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out from \$log where \$filter and logid_to_int(logid) not in (4, 7, 14, 20) and hostname is not null and \$browse_time is not null group by hostname, catdesc) t group by hostname, catdesc order by ebtr_value(ebtr_agg_flat(browsetime), null, null) desc)### t group by hostname order by browsetime desc

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

```
select
```

```
hourstamp,
```

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

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

Dataset Name	Description	Log Category	
ctap-Top-Bandwidth-Hosts	Top Bandwidth Hosts	traffic	
<pre>select hostname, sum(coalesce(sentbyte</pre>	e, 0)+ coalesce(rcvdbyte, 0)		
) as bandwidth	, , , , , , ,		
from \$log - traffic			
where \$filter			
and hostname is not null and logid to int(logid) not in (4, 7, 14, 20)			
group by hostname			
having			
<pre>sum(coalesce(sentbyte) & gt; 0</pre>	e, 0)+ coalesce(rcvdbyte, 0)		

order by

bandwidth desc

Dataset Name	Description	Log Category	
saas-Application-Discovered	All Applications Discovered on the Network	traffic	
<pre>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</pre>			
Dataset Name	Description	Log Category	
saas-SaaS-Application-by-Category	Number of SaaS Applications by Category	traffic	
<pre>) as saas_cat_str, count(distinct app_ from ###(select app_s, s as bandwidth, count(s app_s, unnest(saasin; byte, coalesce(rcvdb) and apps is not null)</pre>	en 0 then 'Sanctioned' else 'Uns saas_s%10 as saas_cat, sum(sentb) as total_app from (select unn fo) as saas_s, coalesce(sentbyte yte, 0) as rcvdbyte from \$log wh t where saas_s>=10 group by app esc)### t where saas_cat in (0, as_cat	yte+rcvdbyte) est(apps) as , 0) as sent- ere \$filter p_s, saas_cat	
Dataset Name	Description	Log Category	
saas-SaaS-Application-by-Bandwid	h Number of SaaS Applications by Bandwidth	traffic	
<pre>select (case saas_cat whe) as saas_cat_str, sum(bandwidth) as b</pre>	en 0 then 'Sanctioned' else 'Tol	erated' end	

from

###(select app_s, saas_s%10 as saas_cat, sum(sentbyte+rcvdbyte)
as bandwidth, count(*) as total_app from (select unnest(apps) as
app_s, unnest(saasinfo) as saas_s, coalesce(sentbyte, 0) as sentbyte, coalesce(rcvdbyte, 0) as rcvdbyte from \$log where \$filter
and apps is not null) t where saas_s>=10 group by app_s, saas_cat
order by bandwidth desc)### t where saas_cat in (0, 2) group by
saas cat order by saas cat

```
Dataset Name
                           Description
                                                                 Log Category
saas-SaaS-Application-by-Session
                           Number of SaaS Applications by Session
                                                                 traffic
select
  (
    case saas cat when 0 then '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 sent-
byte, coalesce(rcvdbyte, 0) as rcvdbyte from $log where $filter
and apps is not null) t where saas s \ge 10 group by app s, saas cat
order by bandwidth desc) ### t where saas cat in (0, 2) group by
saas cat order by saas cat
Dataset Name
                           Description
                                                                 Log Category
saas-SaaS-App-Users-vs-Others
                           Number of Users of SaaS Apps vs Others
                                                                 traffic
select
  (
    case is saas when 0 then 'Other Apps' else 'SaaS Apps' end
```

```
) as app_type,
```

count(distinct saasuser) as total_user

```
from
```

###(select saasuser, saas_s/10 as is_saas from (select coalesce (nullifna(`user`), nullifna(`clouduser`), nullifna(`unauthuser`), srcname, ipstr(`srcip`)) as saasuser, unnest(saasinfo) as saas_s from \$log where \$filter and apps is not null) t group by saasuser, is_saas)### t group by app_type

Dataset Name	Description	Log Category	
saas-SaaS-App-Users	Number of Users of SaaS Apps	traffic	
<pre>select (case saas_cat when 0 then 'Sanctioned' when 1 then 'Unsanc- tioned' 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 bar sum(traffic_in) as tr sum(traffic_out) as t sum(sessions) as sess sum(session_block) as (sum(sessions) - sum() as session_pass, count(distinct app s)</pre>	caffic_in, craffic_out, sions, s session_block, (session_block)		

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 (action IN ('deny', 'ip-conn', 'dns') OR (utmaction IN ('block', 'blocked', 'reset', 'dropped'))) THEN 1 ELSE 0 END) as is_blocked from \$log where \$filter and apps is not null) t where saas_s>=10 group by saasuser, app_s order by bandwidth desc)### t group by saasuser order by bandwidth desc

Dataset Name	Description	Log Category	
saas-Top-Category-by-SaaS- Application-Usage	Top Categories by SaaS Application Usage	traffic	
<pre>select app_cat, (case saas_cat when 0 then 'Sanctioned' else 'Unsactioned' end) as saas_cat_str, count(distinct app_s) as total_app from ###(select app_s, saas_s%10 as saas_cat from (select unnest (apps) as app_s, unnest(saasinfo) as saas_s from \$log where \$fil- ter and apps is not null) t where saas_s>=10 group by app_s, saas_ cat)### t1 inner join app_mdata t2 on t1.app_s=t2.name where saas_ cat in (0, 1) group by app_cat, saas_cat order by total_app desc</pre>			
Dataset Name	Description	Log Category	
saas-Top-SaaS-Category-by-Numl of-User	per- Top SaaS Categories by Number of Users	traffic	
) as saas_cat_str, count(distinct saa	en 0 then 'Sanctioned' else 'Uns suser) as total_user	actioned' end	
<pre>unnest(apps) as app_ lifna(`user`), nulli name, ipstr(`srcip`) is not null) t where user)### t1 inner jo</pre>	<pre>saas_s%10 as saas_cat, saasuser s, unnest(saasinfo) as saas_s, c fna(`clouduser`), nullifna(`unau) as saasuser from \$log where \$f saas_s>=10 group by app_s, saas in app_mdata t2 on t1.app_s=t2.m roup by app_cat, saas_cat order</pre>	coalesce(nul- thuser`), src- filter and apps s_cat, saas- name where	

desc

Dataset Name	Description	Log Category
saas-Top-User-by-Number-of-SaaS- Application	Top Users by Number of SaaS Applications	traffic
select		
saasuser,		
(
case saas_cat when	0 then 'Sanctioned' else 'Unsa	ctioned' end
) as saas_cat_str,		
count(distinct app_s)) as total_app	
from		
###(select app_s, saa	as_s%10 as saas_cat, saasuser f	from (select
unnest(apps) as app_s,	unnest(saasinfo) as saas_s, co	alesce(nul-
lifna(`user`), nullifna	a(`clouduser`), nullifna([`] unaut	huser`), src-
<pre>name, ipstr(`srcip`)) a</pre>	as saasuser from \$log where \$fi	lter and apps
is not null) t where sa	aas_s>=10 group by app_s, saas_	cat, saas-

```
user)### t where saas_cat in (0, 1) group by saasuser, saas_cat
order by total_app desc
```

```
Dataset Name
                          Description
                                                              Log Category
saas-Top-SaaS-Application-by-
                                                               traffic
                          Top SaaS Applications by Sessions and Bandwidth
Bandwidth-Session
select
  t2.id as app id,
  app s,
  app cat,
  sum (bandwidth) as bandwidth,
  sum(traffic in) as traffic in,
  sum(traffic out) as traffic out,
  sum(sessions) as sessions,
  sum(session block) as session block,
  (
    sum(sessions) - sum(session block)
  ) as session pass
from
  ###(select app s, sum(sentbyte+rcvdbyte) as bandwidth, sum(rcvd-
byte) as traffic in, sum(sentbyte) as traffic out, count(*) as ses-
sions, sum(is blocked) as session block from (select unnest(apps)
as app s, unnest(saasinfo) as saas s, coalesce(sentbyte, 0) as
sentbyte, coalesce(rcvdbyte, 0) as rcvdbyte, (CASE WHEN (action IN
('deny', 'ip-conn', 'dns') OR (utmaction IN ('block', 'blocked',
```

'reset', 'dropped'))) 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 + rcvdby from (select unnest(apps) as a unnest(saasinfo) coalesce(sentbyte coalesce(rcvdbyte from \$log where \$filter and apps is not n) t where saas_s = 12 group by app_s order by bandwidth desc</pre>	pp_s, as saas_s, , 0) as sentbyte, , 0) as rcvdbyte	
Dataset Name	Description	Log Category
saas-drilldown-Top-Tolerated-SaaS- Application	Top Tolerated SaaS Applications	traffic
select		
app_s,		

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

###(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 (action IN ('deny', 'ip-conn', 'dns') OR (utmaction IN ('block', 'blocked', 'reset', 'dropped'))) THEN 1 ELSE 0 END) as is_blocked from \$log where \$filter and apps is not null) t where saas_s=12 group by saasuser, app_s order by bandwidth desc)### t where \$filter-drilldown group by app_s order by bandwidth desc

Dataset Name	Description	Log Category
saas-Top-User-by-Tolerated-SaaS- Application-Drilldown	Top Users by Tolerated SaaS Applications	traffic

select

saasuser, count(distinct app_s) as total_app

from

###(select saasuser, app_s, sum(sentbyte+rcvdbyte) as bandwidth, sum(rcvdbyte) as traffic_in, sum(sentbyte) as traffic_out, count (*) as sessions, sum(is_blocked) as session_block from (select coalesce(nullifna(`user`), nullifna(`clouduser`), nullifna(`unauthuser`), srcname, ipstr(`srcip`)) as saasuser, unnest(apps) as app_s, unnest(saasinfo) as saas_s, coalesce(sentbyte, 0) as sentbyte, coalesce(rcvdbyte, 0) as rcvdbyte, (CASE WHEN (action IN ('deny', 'ip-conn', 'dns') OR (utmaction IN ('block', 'blocked', 'reset', 'dropped'))) THEN 1 ELSE 0 END) as is_blocked from \$log where \$filter and apps is not null) t where saas_s=12 group by saasuser, app_s order by bandwidth desc)### t group by saasuser order by total_app desc

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

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

###(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 (action IN ('deny', 'ip-conn', 'dns') OR (utmaction IN ('block', 'blocked', 'reset', 'dropped'))) 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
	n '5' then 'Critical' wh n' when '2' then 'Info'	_

```
total_user
from
(
    select
    app_group,
    count(distinct saasuser) as total_user,
    sum(bandwidth) as bandwidth,
    sum(traffic_in) as traffic_in,
    sum(traffic_out) as traffic_out,
    sum(sessions) as sessions,
    sum(session_block) as session_block,
    (
        sum(sessions)- sum(session_block)
    ) as session_pass
```

###(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 (action IN ('deny', 'ip-conn', 'dns') OR (utmaction IN ('block', 'blocked', 'reset', 'dropped'))) 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
	5' then 'Critical' when '4' then when '2' then 'Info' else 'Low' e	-

```
traffic in,
  traffic out,
  sessions,
  session block,
  session pass,
  total user
from
  (
    select
      app group,
      count(distinct saasuser) as total user,
      sum (bandwidth) as bandwidth,
      sum(traffic in) as traffic in,
      sum(traffic out) as traffic out,
      sum(sessions) as sessions,
      sum(session block) as session block,
      (
        sum(sessions) - sum(session block)
      ) as session pass
```

###(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 (action IN ('deny', 'ip-conn', 'dns') OR (utmaction IN ('block', 'blocked', 'reset', 'dropped'))) THEN 1 ELSE 0 END) as is blocked from \$log where \$filter and apps is not null) t where saas $s \ge 10$ 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
<pre>select srccountry, count(distinct devid) from</pre>	as device_count	

###(select srccountry, devid from \$log where \$filter and logid_ to_int(logid) not in (4, 7, 14, 20) and nullifna(srccountry) is not null group by srccountry, devid)### t group by srccountry order by device_count desc

Dataset Name	Description	Log Category
aware-Top-Endpoint-Operating- Systems	Top Endpoint Operating Systems	fct-traffic
<pre>select osl as os, count(distinct hostname) as total_num from ###(select split_part(os, ',', 1) as osl, hostname from \$log where \$filter and nullifna(os) is not null group by osl, host- name)### t group by os order by total num desc</pre>		
Dataset Name	Description	Log Category
aware-Top-Endpoint-Applications-	Top Endpoint Applications Windows	fct-traffic

```
select
```

Windows

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

Dataset Name	Description	Log Category
aware-Top-Endpoint-Applications-Mac	Top Endpoint Applications Mac	fct-traffic
<pre>select srcname, count(*) as total_num from</pre>		

```
$log
where
$filter
and nullifna(srcname) is not null
and lower(os) like '%mac os%'
group by
srcname
order by
total_num desc
```

Dataset Name	Description	Log Category
aware-Summary-Of-Changes	Summary of Changes	event
<pre>select regexp_replace(msg, count(*) as total r</pre>	. '[^]*\$' , '') as msg_tri num	.m,
from		
\$log		
where		
\$filter		
and logid_to_int(lo	ogid)= 44547	
group by		
msg_trim		
order by		
total_num desc		

Dataset Name	Description	Log Category
aware-Change-Details	Change Details	event
<pre>select \$calendar_time as `user`, ui, msg from \$log</pre>	timestamp,	
<pre>where \$filter and logid_to_int(order by timestamp desc</pre>	logid)= 44547	

Dataset Name	Description	Log Category
aware-Vulnerabilities-By-Severity	Vulnerabilities by Security	fct-netscan
from ###(select vulnsev nullifna(vulnname) i	nname) as vuln_num erity, vulnname from \$log w s not null and nullifna(vul verity, vulnname)### t grou sc	nseverity) is not
Dataset Name	Description	Log Category
aware-Vulnerabilities-Trend	Vulnerabilities Trend	fct-netscan
_		

```
select
```

```
$flex_timescale(timestamp) as timescale,
sum(critical) as critical,
sum(high) as high,
sum(medium) as medium,
sum(low) as low
```

###(select \$flex_timestamp as timestamp, sum(case when lower (vulnseverity) = 'critical' then 1 else 0 end) as critical, sum (case when lower(vulnseverity) = 'high' then 1 else 0 end) as high, sum(case when lower(vulnseverity) = 'medium' then 1 else 0 end) as medium, sum(case when lower(vulnseverity) = 'notice' then 1 else 0 end) as Low from \$log where \$filter group by timestamp order by timestamp desc)### t group by timescale order by timescale

Dataset Name	Description	Log Category
aware-Top-Critical-Vulnerabilities	Top Critical Vulnerabilities	fct-netscan
<pre>select hostname, vulnname, vulnseverity, vulncat, count (*) as total_num from \$log</pre>	m	

```
where
  $filter
  and nullifna(hostname) is not null
  and nullifna(vulnname) is not null
  and lower(vulnseverity) = 'critical'
group by
  hostname,
  vulnname,
  vulnname,
  vulnseverity,
  vulncat
order by
  total_num desc
```

Dataset Name	Description	Log Category
aware-Top-Device-Attack-Targets	Top Device Attack Targets	fct-netscan
<pre>select hostname, count(*) as total_nu from \$log where \$filter and nullifna(hostnam and nullifna(vulnnam group by hostname order by total_num desc</pre>	ne) is not null	
Dataset Name	Description	Log Category
aware-Top-Attack-Targets	Top Attack Targets	fct-netscan
(vulnseverity) = 'high	name) as vuln_num, nunseverity) = 'critical' n' then 4 when lower(vulns nseverity) = 'low' then 2	everity) = 'medium'

```
) as severity_num,
string_agg(distinct cve_id, ',') as cve_agg
from
  ###(select hostname, split_part(os, ',', 1) as os1, vulnname,
vulnseverity, vulnid from $log where $filter and nullifna(vul-
nname) 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, os,
vulnseverity order by severity num desc, vuln num desc
```

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

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

```
(
```

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

```
Dataset Name
                           Description
                                                                Log Category
aware-Threats-Type-By-Severity
                           Threats Type by Severity
                                                                 virus
select
  threat type,
  sum(critical) as critical,
  sum(high) as high,
  sum(medium) as medium,
  sum(low) as low
from
  (
    ###(select 'Malware' as threat type, sum(case when crlevel =
'critical' then 1 else 0 end) as critical, sum(case when crlevel =
'high' then 1 else 0 end) as high, sum(case when crlevel =
```

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

Dataset Name	Description	Log Category
aware-Threats-By-Day	Threats by Day	virus
<pre>from \$log-virus when group by daystamp)## stamp, count(*) as # nullifna(attack) is (select \$day_of_wee)</pre>	_of_week as daystam re \$filter and null ## union all ###(se total_num from \$log not null group by c as daystamp, coun ter and lower(appca	<pre>up, count(*) as total_num ifna(virus) is not null elect \$day_of_week as day- r-attack where \$filter and daystamp)### union all ### at(*) as total_num from \$log- at)='botnet' group by day- by daystamp</pre>
Dataset Name	Description	Log Category
aware-Threats-By-Day-Radar	Threats by Day	virus
<pre>select daystamp, sum(total_num) as from (</pre>	total_num	

###(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 \$logapp-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 nu</pre>	m	
from \$log		
where \$filter		
and nullifna(virus)	is not null	
group by virus		
order by total_num desc		

Dataset Name	Description	Log Category
aware-Top-Malware-By-Count	Top Malware by Count	app-ctrl
<pre>select virus, malware_type, risk_level, count(distinct dstip count(distinct srcip sum(total num) as to</pre>) as source,	
from	—	
(
	virus, 'Botnet C&C' as mal	
	level, dstip, srcip, count	—
<pre>nullifna(apprisk::text</pre>	ere \$filter and lower(appca 2) is not null group by app order by total_num desc)##	, malware_type,

(select virus, 'Virus' 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)###) 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 tota</pre>		
<pre>count(status) as tota from \$log</pre>		
<pre>where \$filter and nullifna(`user`) and logid to int(logi</pre>		
<pre>group by ui, f_user, dstip order by total failed desc</pre>		

Dataset Name	Description	Log Category
aware-Top-Denied-Connections	Top Denied Connections	traffic
<pre>select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, service '(' ips dstip, count(*) as total_num from \$log where</pre>	str(srcip) ')' as ir n	iterface,

```
$filter
and logid_to_int(logid) not in (4, 7, 14, 20)
and action = 'deny'
group by
user_src,
interface,
dstip
order by
total_num desc
```

Dataset Name	Description	Log Category
aware-Failed-Compliance-Checked- By-Device	Failed Compliance Checked by Device	event
select		

```
devid,
    'Failed' as results,
    count(distinct reason) as total_num
from
```

```
###(select devid, reason from $log where $filter and sub-
type='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
<pre>select coalesce(nullifna(epname), nullifna(ipstr(`srcip`)), 'Unknown') as epname, total_bl, threats from (select t1.epname,</pre>		
t1.srcip, t2.total bl,		
cz.cocar_br,		

```
t2.total cs,
  t2.max verdict,
  t2.max cs,
  t2.threats
from
  $ADOMTBL PLHD PBD EP t1
  inner join (
    select
      tl.epid,
      total bl,
      total cs,
      max verdict,
      max cs,
      threats
    from
      (
        select
          epid,
          sum(bl count) as total bl,
          sum(cs count) as total cs,
          max(verdict) as max verdict,
          max(cs score) as max cs
        from
          (
            select
              day st as itime,
              epid,
              bl count,
              cs count,
              verdict,
              cs score
            from
               $ADOMTBL_PLHD_PBD_STSUM
          ) t
        where
          $filter
          and $filter - drilldown
        group by
          epid
      ) t1
      inner join (
```

```
select
              epid,
              string agg(name, ',') as threats
            from
               (
                 select
                   epid,
                   thid
                 from
                   (
                     select
                       day st as itime,
                       epid,
                       unnest(threatid) as thid
                     from
                       $ADOMTBL PLHD PBD STSUM
                   ) t
                where
                   $filter
                   and $filter - drilldown
                group by
                   epid,
                   thid
              ) t1
              inner join td threat name mdata t2 on t1.thid =
t2.id
            group by
              epid
          ) t2 on t1.epid = t2.epid
      ) t2 on t1.epid = t2.epid
  ) t
where
  total bl & gt; 0
order by
  total bl desc
```

Dataset Name	Description	Log Category
aware-loc-Potential-Breach-By-Day	IOC Potential Breach by Day	app-ctrl
select number, day_st as itime		

```
from
  (
    select
      count(epid) as number,
      to char(
        from itime(itime),
        'Day'
      ) as day_st
    from
      (
        select
          epid,
          day st as itime
        from
          $ADOMTBL_PLHD_PBD_STSUM
        where
          cs_count & gt; 0
      ) t
    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 num to_char(from_itime(itime 'Day') as day_st</pre>		

```
from
      (
        select
          epid,
          day st as itime
        from
          $ADOMTBL_PLHD_PBD_STSUM
        where
          cs count & gt; 0
      ) t
    where
      $filter
      and $filter - drilldown
    group by
      day st
  ) tt
order by
  itime
```

Description **Dataset Name** Log Category aware-loc-Suspicion-Summary **IOC Suspicion Summary** app-ctrl select coalesce(nullifna(epname), nullifna(ipstr(`srcip`)), 'Unknown') as epname, total cs, max cs, max verdict, threats from (select t1.epname, tl.srcip, t2.total bl, t2.total cs, t2.max verdict,

```
t2.max cs,
  t2.threats
from
  $ADOMTBL PLHD PBD EP t1
  inner join (
    select
      tl.epid,
      total bl,
      total cs,
      max verdict,
      max cs,
      threats
    from
      (
        select
          epid,
          sum(bl count) as total bl,
          sum(cs count) as total cs,
          max(verdict) as max verdict,
          max(cs score) as max cs
        from
          (
            select
               day st as itime,
               epid,
              bl count,
              cs count,
              verdict,
               cs score
            from
               $ADOMTBL PLHD PBD STSUM
          ) t
        where
          $filter
          and $filter - drilldown
        group by
          epid
      ) t1
      inner join (
        select
          epid,
```

```
string agg(name, ',') as threats
            from
               (
                select
                  epid,
                  thid
                 from
                   (
                     select
                       day st as itime,
                       epid,
                       unnest(threatid) as thid
                     from
                       $ADOMTBL PLHD PBD STSUM
                   ) t
                where
                   $filter
                   and $filter - drilldown
                group by
                  epid,
                  thid
              ) t1
              inner join td threat name mdata t2 on t1.thid =
t2.id
            group by
              epid
          ) t2 on t1.epid = t2.epid
      ) t2 on t1.epid = t2.epid
  ) t
where
  total bl = 0
  and total cs & gt; 0
order by
  max verdict desc,
  max cs desc,
  total cs desc
```

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

table if exists rpt tmptbl 1;

drop

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

	Log Category
newthing-New-Devices New devices	fct-traffic

drop

```
table if exists rpt_tmptbl_1;
```

drop

table if exists rpt_tmptbl_2; create temporary table rpt_tmptbl_ 1 as ###(select hostname, os, srcip, fctver from \$log where \$pre_ period \$filter and hostname is not null group by hostname, os, srcip, fctver order by hostname) ###; create temporary table rpt_ tmptbl_2 as ###(select hostname, os, srcip, fctver from \$log where \$filter and hostname is not null group by hostname, os, srcip, fctver order by hostname)###; select hostname, max(fctos_to_devtype(os)) as devtype, string_agg(distinct os, '/') as os_agg, string_agg(distinct ipstr(srcip), '/') as srcip_agg, string_agg (distinct fctver, '/') as fctver_agg from rpt_tmptbl_2 where not exists (select 1 from rpt_tmptbl_1 where rpt_tmptbl_2.hostname=rpt tmptbl 1.hostname) group by hostname order by hostname

Dataset Name	Description	Log Category
newthing-New-Software-Installed	New software installed	fct-traffic

drop

table if exists rpt_tmptbl_1;

drop

table if exists rpt_tmptbl_2; create temporary table rpt_tmptbl_
1 as ###(select srcproduct, hostname from \$log where \$pre_period
\$filter and nullifna(srcproduct) is not null group by srcproduct,
hostname order by srcproduct)###; create temporary table rpt

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

Dataset Name	Description	Log Category
newthing-New-Security-Threats	New security threats	virus
<pre>drop table if exists rpt_ drop table if exists rpt_ 1 as select * from</pre>	_tmptbl_1; _tmptbl_2; create tempora	ary table rpt_tmptbl_
app-ctrl where \$pre_per and lower(appcat)='bot union all ###(select v \$log-virus where \$pre_ null group by threat_n attack as threat_name, \$pre_period \$filter and threat_name, cat_id, s tmptbl_2 as select * f app as threat_name, 1 \$filter and nullifna(a group by daystamp, thr stamp)### union all ## threat_name, 2 as cat_ nullifna(virus) is not id, srcip order by day as daystamp, attack as attack where \$filter a stamp, threat_name, cat threat_name, (case cat	threat_name, 1 as cat_ic eriod \$filter and nullifr inet' group by threat_name, irus as threat_name, 2 as period \$filter and nullifr name, cat_id, srcip)### u 3 as cat_id, srcip from ad nullifna(attack) is not ercip)###) t; create temp from (###(select \$DAY_OF_ as cat_id, srcip from \$1 opp) is not null and lowe eeat_name, cat_id, srcip #(select \$DAY_OF_MONTH as id, srcip from \$log-viru to null group by daystamp, stamp)### union all ### s threat_name, 3 as cat_i and nullifna(attack) is r it_id, srcip order by day c_id when 1 then 'Botnet' tack' end) as threat cat,	<pre>ha(app) is not null me, cat_id, srcip)### as cat_id, srcip from ifna(virus) is not union all ###(select m \$log-attack where of null group by porary table rpt_ MONTH as daystamp, log-app-ctrl where er(appcat)='botnet' order by day- as daystamp, virus as us where \$filter and threat_name, cat_ (select \$DAY_OF_MONTH id, srcip from \$log- not null group by day- ystamp)###) t; select ' when 2 then 'Mal-</pre>

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_namee=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

Fortinet Technologies Inc.

Dataset Name	Description	Log Category
newthing-New-Security-Threats- Timeline	New security threats timeline	virus
drop table if exists r drop		
1 as select	<pre>pt_tmptbl_2; create temporary</pre>	table rpt_tmptb1
from (as threat name 1 as est id a	

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

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

drop

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

table if exists rpt tmptbl 2; create temporary table rpt tmptbl 1 as ###(select vulnid, vulnname, vulnseverity, vulncat, hostname from \$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 msg from \$log where \$filter and msg is not null and pri = 'critical' order by timestamp desc</pre>	timestamp,	
Dataset Name	Description	Log Category
newthing-Configuration-Changes	Configuration Changes	event
<pre>select `user` as f_user, devid, from_dtime(dtime) as ui, msg from \$log where \$filter and cfgtid & gt; 0 order by</pre>	time_s,	

Dataset Name	Description	Log Category
newthing-FortiGate-Upgrades	FortiGate Upgrades	event
<pre>select devid, from_dtime(dtime) as info[1] as intf, info[2] as prev_ver, info[3] as new_ver from</pre>	time_s,	

```
(
   select
    devid,
    dtime,
    regexp_matches(
        msg, 'from ([^ ]+) \\(([^ ]+) -> ([^)]+)\\)'
    ) as info
   from
      $log
   where
      $filter
      and action = 'restore-image'
   ) t
order by
   time s desc
```

Dataset Name	Description	Log Category
newthing-User-Upgrades	User Upgrades	fct-event

```
drop
```

```
table if exists rpt_tmptbl_1;
```

drop

table if exists rpt_tmptbl_2; create temporary table rpt_tmptbl_ 1 as ###(select distinct on (1, 2) fgtserial, hostname, deviceip, os, dtime from \$log where \$pre_period \$filter and 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, t2.os as cur_os, from_dtime(t1.dtime) as time_s from rpt_tmptbl_2 t2 inner join rpt_tmptbl_1 t1 on t2.fgtserial=t1.fgtserial and t2.hostname=t1.hostname and t2.os-!=t1.os order by devid, t2.hostname, t1.dtime desc

Dataset Name	Description	Log Category
GTP-List-of-APN-Used	List of APNs Used	gtp
<pre>select apn, from_dtime(min(first_seen)</pre>		

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

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

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

GTP-Top-APN-by-Duration Top	o APNs by Duration	gtp
<pre>select apn, sum(coalesce(duration, 0)) as total_dura from</pre>		

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

Dataset Name	Description	Log Category
Top10-dns-Botnet-Domain-IP	Top Queried Botnet C&C Domains and IPs	dns
select domain,		

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

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

Dataset Name	Description	Log Category
dns-Botnet-Usage	Top Queried Botnet C&C Domains and IPs	dns
select		
domain, malware_type,		
action,		
<pre>count(distinct srcip) count(distinct source</pre>		
sum(total num) as tot	—	
from	—	
	otnetdomain, ipstr(botnetip)) as	
	<pre>ar(32)) as malware_type, (case w when action='redirect' then 'Re</pre>	
	ction, srcip, (CASE WHEN level]	
ical', 'alert', 'emerge	ncy') THEN 5 WHEN level='error'	THEN 4 WHEN
_	WHEN level='notice' THEN 2 ELSE	
	<pre>omain, ipstr(botnetip)) as source log where \$filter and (botnetdom</pre>	_
—	null) group by domain, action,	
_	sc) ### t group by domain, malwar	-

action order by total_num desc

Dataset Name	Description	Log Category
Dns-Detected-Botnet	Top Queried Botnet C&C Domains and IPs	dns
select		
domain,		
malware_type,		
action,		
count(distinct sr	cip) as victims,	
count(distinct so	urces_s) as sources,	
sum(total_num) as	total_num	
from		
###(select coales	ce(botnetdomain, ipstr(botnetip)) as domain,
cast('Botnet C&C' a	s char(32)) as malware_type, (c	ase when action-
='block' then 'Bloc	ked' when action='redirect' the	n 'Redirected'

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

Dataset Name	Description	Log Category
dns-Botnet-Domain-IP	Queried Botnet C&C Domains and IPs	dns
THEN 'Medium' WHEN '2') as severity from	THEN 'Critical' WHEN 4 THEN 'High THEN 'Info' ELSE 'Low' END	
<pre>cast('Botnet C&C' as c ='block' then 'Blocked else 'Passed' end) as ical', 'alert', 'emerg level='warning' THEN 3</pre>	<pre>botnetdomain, ipstr(botnetip)) as har(32)) as malware_type, (case w ' when action='redirect' then 'Re action, srcip, (CASE WHEN level I ency') THEN 5 WHEN level='error' WHEN level='notice' THEN 2 ELSE domain, ipstr(botnetip)) as sourc</pre>	hen action- directed' N ('crit- THEN 4 WHEN 1 END) as

(*) as total_num from \$log where \$filter and (botnetdomain is not null or botnetip is not null) group by domain, action, srcip, sevid order by sevid desc)### t group by domain, srcip, sevid order by sevid desc, domain

Dataset Name	Description	Log Category
dns-High-Risk-Source	High Risk Sources	dns
select srcip,		
<pre>sum(total_num) as tot sum(</pre>	al_num,	
<pre>case when sevid = 5) as num cri,</pre>	then total_num else 0 end	
sum(then total num else 0 end	
) as num_hig, sum(_	
•	then total_num else 0 end	
from		
<pre>###(select srcip, (CA 'emergency') THEN 5 WHE THEN 3 WHEN level='noti as total_num from \$log by srcip, sevid order b</pre>	SE WHEN level IN ('critical', 'al N level='error' THEN 4 WHEN level ce' THEN 2 ELSE 1 END) as sevid, where \$filter and srcip is not nu y total_num desc)### t where sevi al_num)>0 order by total_num desc	l='warning' count(*) ull group id>=3 group

Dataset Name	Description	Log Category
dns-DNS-Request-Over-Time	DNS Request Over Time	dns
) as num_cri, sum(tamp) as timescale, then total_num else 0 end then total_num else 0 end	
<pre>case when sevid = 3) as num_med,</pre>	then total_num else 0 end	

```
sum(
    case when sevid = 2 then total_num else 0 end
) as num_inf,
    sum(
      case when sevid = 1 then total_num else 0 end
) as num_low
from
    ###(select $flex_timestamp as timestamp, (CASE WHEN level IN
('critical', 'alert', 'emergency') THEN 5 WHEN level='error' THEN
4 WHEN level='warning' THEN 3 WHEN level='notice' THEN 2 ELSE 1
END) as sevid, count(*) as total_num from $log where $filter group
by timestamp, sevid order by total_num desc)### t group by times-
cale order by timescale
```

Dataset Name	Description	Log Category
dns-Top-Queried-Domain	Top Queried Domain	dns
<pre>select qname, count(*) as total_num from \$log where \$filter and qname is not null group by qname order by total_num desc</pre>		

Dataset Name	Description	Log Category
dns-Top-Domain-Lookup-Failure-Bar	Top Domain Lookup Failures	dns
<pre>select qname, srcip, count(*) as total_num from \$log where \$filter and qname is not null</pre>		

```
and (
    action = 'block'
    or logid_to_int(logid) = 54001
)
group by
qname,
srcip
order by
total_num desc
```

Dataset Name Description Log Category dns-Top-Domain-Lookup-Failure-Table Top Domain Lookup Failures dns select qname, srcip, count(*) as total num from \$log where \$filter and qname is not null and (action = 'block' or logid to int(logid) = 54001) group by qname, srcip order by total num desc

dns-Query-Timeout Query Timeout dns select srcip, qname, count(*) as total_num from \$log where	 Dataset Name	Description	Log Category
<pre>select srcip, qname, count(*) as total_num from \$log</pre>			
<pre>srcip, qname, count(*) as total_num from \$log</pre>	uns-Query-Timeout	Query mileout	uis
<pre>qname, count(*) as total_num from \$log</pre>	select		
<pre>count(*) as total_num from \$log</pre>	srcip,		
from \$log	-		
\$log		al_num	
-			
where	-		
	wnere		

```
$filter
and srcip is not null
and logid_to_int(logid) = 54001
group by
qname,
srcip
order by
total_num desc
```

Dataset Name	Description	Log Category
dns-Blocked-Query	Blocked Queries	dns
select srcip, msg, count(*) as tota	l_num	
from		
\$log		
where		
\$filter		
and srcip is not		
and action = 'bl	ock'	
group by		
srcip,		
msg		
order by		
total_num desc		

Dataset Name	Description	Log Category
perf-stat-cpu-usage-drilldown	Fortigate resource detail timeline	event
) as cpu_ave, cast(sum(total_mem)/ sum) as mem_ave, cast(tamp) as hodex, (count) as decimal(6, 0) (count) as decimal(6, 0) m(count) as decimal(6, 0)	

```
) as disk ave,
  cast(
    (
      sum(
        total trate + total erate + total orate
      )
    )/ 100.00 / sum(count) as decimal(10, 2)
  ) as log rate,
  cast(
    sum(totalsession) / sum(count) as decimal(10, 0)
  ) as sessions,
  cast(
    sum(sent) / sum(count) as decimal(10, 0)
  ) as sent kbps,
  cast(
    sum(recv) / sum(count) as decimal(10, 0)
  ) as recv kbps,
  cast(
    sum(sent + recv) / sum(count) as decimal(10, 0)
  ) as transmit kbps,
  max(mem peak) as mem peak,
  max(disk peak) as disk peak,
  max(cpu peak) as cpu peak,
  cast(
    max(lograte peak) / 100.00 as decimal(10, 2)
  ) as lograte peak,
  max(session peak) as session peak,
  max(transmit peak) as transmit kbps peak,
  cast(
    sum(cps) / sum(count) as decimal(10, 0)
  ) as cps ave,
 max(cps peak) as cps peak
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
perf-stat-mem-usage-drilldown
                           Fortigate resource detail 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(
         total trate + total erate + total orate
      )
    )/ 100.00 / sum(count) as decimal(10, 2)
  ) as log rate,
  cast(
    sum(totalsession) / sum(count) as decimal(10, 0)
  ) as sessions,
  cast(
    sum(sent) / sum(count) as decimal(10, 0)
  ) as sent kbps,
  cast(
    sum(recv) / sum(count) as decimal(10, 0)
  ) as recv kbps,
  cast(
```
```
sum(sent + recv)/ sum(count) as decimal(10, 0)
) as transmit_kbps,
max(mem_peak) as mem_peak,
max(disk_peak) as disk_peak,
max(cpu_peak) as cpu_peak,
cast(
    max(lograte_peak)/ 100.00 as decimal(10, 2)
) as lograte_peak,
max(session_peak) as session_peak,
max(transmit_peak) as transmit_kbps_peak,
cast(
    sum(cps)/ sum(count) as decimal(10, 0)
) as cps_ave,
max(cps_peak) as cps_peak
```

###(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
perf-stat-disk-usage-drilldown	Fortigate resource detail timeline	event
<pre>select \$flex_timescale(times devid, cast(sum(total_cpu)/ sum) as cpu_ave,</pre>	stamp) as hodex, n(count) as decimal(6, 0)	

```
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(
        total trate + total erate + total orate
      )
    )/ 100.00 / sum(count) as decimal(10, 2)
  ) as log rate,
  cast(
    sum(totalsession) / sum(count) as decimal(10, 0)
  ) as sessions,
  cast(
    sum(sent) / sum(count) as decimal(10, 0)
  ) as sent kbps,
  cast(
    sum(recv) / sum(count) as decimal(10, 0)
  ) as recv kbps,
  cast(
    sum(sent + recv) / sum(count) as decimal(10, 0)
  ) as transmit kbps,
  max(mem peak) as mem_peak,
  max(disk peak) as disk peak,
  max(cpu peak) as cpu peak,
  cast(
    max(lograte peak) / 100.00 as decimal(10, 2)
  ) as lograte peak,
  max(session peak) as session peak,
  max(transmit peak) as transmit kbps peak,
  cast(
    sum(cps) / sum(count) as decimal(10, 0)
  ) as cps ave,
 max(cps peak) as cps peak
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, '/', 1), '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
perf-stat-sessions-drilldown	Fortigate resource detail timeline	event
<pre>select \$flex_timescale(t: devid, cast(sum(total_cpu)/) as cpu_ave, cast(sum(total_mem)/) as mem_ave, cast(sum(total_disk),) as disk_ave,</pre>	<pre>imestamp) as hodex, sum(count) as decimal(6, 0) sum(count) as decimal(6, 0) f sum(count) as decimal(6, 0)</pre>	
))/ 100.00 / sum) as log_rate, cast(sum(totalsessions,) as sessions, cast(<pre>+ total_erate + total_orate (count) as decimal(10, 2) n)/ sum(count) as decimal(10, count) as decimal(10, 0)</pre>	0)

```
) as sent kbps,
cast(
  sum(recv) / sum(count) as decimal(10, 0)
) as recv kbps,
cast(
  sum(sent + recv) / sum(count) as decimal(10, 0)
) as transmit kbps,
max(mem peak) as mem peak,
max(disk peak) as disk peak,
max(cpu peak) as cpu peak,
cast(
  max(lograte peak) / 100.00 as decimal(10, 2)
) as lograte peak,
max(session peak) as session peak,
max(transmit peak) as transmit kbps peak,
cast(
  sum(cps) / sum(count) as decimal(10, 0)
) as cps ave,
max(cps peak) as cps peak
```

###(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
perf-stat-lograte-drilldown	Fortigate resource detail 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(
        total trate + total erate + total orate
      )
    )/ 100.00 / sum(count) as decimal(10, 2)
  ) as log rate,
  cast(
    sum(totalsession) / sum(count) as decimal(10, 0)
  ) as sessions,
  cast(
    sum(sent) / sum(count) as decimal(10, 0)
  ) as sent kbps,
  cast(
    sum(recv) / sum(count) as decimal(10, 0)
  ) as recv kbps,
  cast(
    sum(sent + recv) / sum(count) as decimal(10, 0)
  ) as transmit kbps,
  max(mem peak) as mem peak,
  max(disk peak) as disk peak,
  max(cpu peak) as cpu peak,
  cast(
    max(lograte peak) / 100.00 as decimal(10, 2)
  ) as lograte peak,
  max(session peak) as session peak,
  max(transmit peak) as transmit kbps peak,
  cast(
    sum(cps) / sum(count) as decimal(10, 0)
```

```
) as cps_ave,
max(cps_peak) as cps_peak
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 $fil-
ter and action='perf-stats' group by timestamp, devid)### t where
$filter-drilldown group by hodex, devid order by hodex
```

Dataset Name	Description	Log Category
perf-stat-connections-drilldown	Fortigate resource detail timeline	event
<pre>) as cpu_ave, cast(sum(total_mem)/ s) as mem_ave, cast(sum(total_disk)/) as disk_ave, cast((sum(total_trate +)</pre>	<pre>sum(count) as decimal(6, 0) sum(count) as decimal(6, 0) sum(count) as decimal(6, 0 total_erate + total_orate</pre>)
)/ 100.00 / sum(c	count) as decimal(10, 2)	

```
) as log rate,
  cast(
    sum(totalsession) / sum(count) as decimal(10, 0)
  ) as sessions,
  cast(
    sum(sent) / sum(count) as decimal(10, 0)
  ) as sent kbps,
  cast(
    sum(recv) / sum(count) as decimal(10, 0)
  ) as recv kbps,
  cast(
    sum(sent + recv) / sum(count) as decimal(10, 0)
  ) as transmit kbps,
  max(mem peak) as mem peak,
  max(disk peak) as disk peak,
  max(cpu peak) as cpu peak,
  cast(
    max(lograte peak) / 100.00 as decimal(10, 2)
  ) as lograte peak,
  max(session peak) as session peak,
  max(transmit peak) as transmit kbps peak,
  cast(
    sum(cps) / sum(count) as decimal(10, 0)
  ) as cps ave,
 max(cps peak) as cps peak
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, '/', 2),
'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
perf-stat-bandwidth-drilldown	Fortigate resource detail timeline	event
select		
	imestamp) as hodex,	
devid,	1 · · · ·	
cast(
sum(total_cpu)/	<pre>sum(count) as decimal(6, 0)</pre>	
) as cpu_ave,		
cast(
_	<pre>sum(count) as decimal(6, 0)</pre>	
) as mem_ave,		
cast(
_	<pre>/ sum(count) as decimal(6, 0)</pre>	
) as disk_ave,		
cast((
sum(
	+ total erate + total orate	
)		
	(count) as decimal(10, 2)	
) as log rate,		
cast(
sum(totalsessior	n)/ sum(count) as decimal(10, 0	0)
) as sessions,		
cast(
	count) as decimal(10, 0)	
) as sent_kbps,		
cast(
	count) as decimal(10, 0)	
) as recv_kbps,		
cast(/ sum(count) as decimal(10, 0))
) as transmit kbps		/
max(mem peak) as r		
max(disk peak) as	—	
max(cpu peak) as o		
cast(_	
<pre>max(lograte_pea)</pre>	<)/ 100.00 as decimal(10, 2)	

```
) as lograte_peak,
max(session_peak) as session_peak,
max(transmit_peak) as transmit_kbps_peak,
cast(
    sum(cps) / sum(count) as decimal(10, 0)
) as cps_ave,
max(cps_peak) as cps_peak
```

###(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
perf-stat-usage-summary-average	Fortigate resource summary view	event
) as cpu_ave, cast(sum(total_mem)/s) as mem_ave, cast(sum(count) as decimal(6, 0 sum(count) as decimal(6, 0 sum(count) as decimal(6,))

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

```
###(select 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 from $log where $filter and action='perf-stats'
group by devid)### t group by devid order by devid
```

```
Dataset Name
                          Description
                                                              Log Category
perf-stat-usage-summary-peak
                          Fortigate resource summary view
                                                              event
select
 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(
        total trate + total erate + total orate
      )
    )/ 100.00 / sum(count) as decimal(10, 2)
  ) as log rate,
  cast(
    sum(totalsession) / sum(count) as decimal(10, 0)
  ) as sessions,
  cast(
    sum(sent) / sum(count) as decimal(10, 0)
  ) as sent kbps,
  cast(
    sum(recv) / sum(count) as decimal(10, 0)
  ) as recv kbps,
  cast(
    sum(sent + recv) / sum(count) as decimal(10, 0)
  ) as transmit kbps,
  max(mem peak) as mem_peak,
  max(disk peak) as disk peak,
 max(cpu peak) as cpu peak,
  cast(
    max(lograte peak) / 100.00 as decimal(10, 2)
  ) as lograte peak,
  max(session peak) as session peak,
  max(transmit peak) as transmit kbps peak
from
```

###(select 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 from \$log where \$filter and action='perf-stats' group by devid)### t group by devid order by devid

Dataset Name	Description	Log Category
perf-stat-usage-details-drilldown- master	Fortigate resource summary view	event
<pre>) as cpu_ave, cast(sum(total_mem)/ s) as mem_ave, cast(sum(total_disk)/) as disk_ave, cast((sum(total_trate -)</pre>	<pre>sum(count) as decimal(6, 0) sum(count) as decimal(6, 0) sum(count) as decimal(6, 0) + total_erate + total_orate count) as decimal(10, 2)</pre>	
<pre>sum(totalsession)) as sessions, cast(</pre>)/ sum(count) as decimal(10,	0)

```
sum(sent) / sum(count) as decimal(10, 0)
 ) as sent kbps,
 cast(
    sum(recv) / sum(count) as decimal(10, 0)
 ) as recv kbps,
 cast(
    sum(sent + recv) / sum(count) as decimal(10, 0)
  ) as transmit kbps,
 max(mem peak) as mem peak,
 max(disk peak) as disk peak,
 max(cpu peak) as cpu peak,
 cast(
   max(lograte peak) / 100.00 as decimal(10, 2)
  ) as lograte peak,
 max(session peak) as session peak,
 max(transmit peak) as transmit kbps peak
from
```

```
###(select 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 from $log where $filter and action='perf-stats'
group by devid)### t group by devid order by devid
```

Dataset Name	Description	Log Category
360-degree-security-Application- Visiblity-and-Control-Summary	Application Visibolity and Control Summary	app-ctrl
<pre>select appcat, count(distinct app) a</pre>	as total_num	
<pre>from ###(select appcat, applicat)</pre>	op from \$log where \$filter and app	is not

null and appcat is not null and logid_to_int(logid) not in (4, 7, 14) group by appcat, app)### t group by appcat order by total_num desc

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

select

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

###(select cast('Malware & Botnet C&C' as char(32)) as threat_ name, app as threats from \$log-app-ctrl where \$filter and lower (appcat)='botnet' group by app)### union all ###(select cast('Malware & Botnet C&C' as char(32)) as threat_name, virus as threats from \$log-virus where \$filter and nullifna(virus) is not null group by virus)### union all ###(select cast('Malicious & Phishing Sites' as char(32)) as threat_name, hostname as threats from \$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 total_num 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>select data_loss, count(*) as total_num from (select (</pre>			
case when sever Exfiltration' else (case when coa nullifna(`u ipstr(`srci	ser`),	then 'Critical D	ata

```
) is not null then 'User Associated Data Loss' else NULL
end
         ) end
      ) as data loss
    from
      $log
    where
      $filter
  ) t
where
  data loss is not null
group by
  data loss
order by
  total num desc
Dataset Name
                          Description
                                                               Log Category
360-degree-security-Endpoint-
                          Endpoint Protection
                                                               fct-traffic
Protection-Summary
select
  blocked event,
  count(*) as total num
from
  (
    select
      (
        case utmevent when 'antivirus' then 'Malware Deteced and
Blocked' when 'appfirewall' then 'Risk Application Blocked' when
'webfilter' then (
           case when coalesce(
             nullifna(`user`),
             ipstr(`srcip`)
           ) is not null then 'Web Sites Violation Blocked' else
'Non User Initiated Web Visits' end
        ) else NULL end
      ) as blocked event
    from
      $log
```

```
and utmaction in ('blocked', 'quarantined')
```

where

\$filter

```
) t
where
blocked_event is not null
group by
blocked_event
order by
total_num desc
```

Macro Reference List

The following table lists the available predefined macros that can be used in a report layout to display the log data as text (XML format) dynamically.

Macro Name	Description	Dataset Used	Log Category
Application Category with Highest Session Count	Application category with the highest session count	App-Sessions-By- Category	Traffic
Application with Highest Bandwidth	Application with the highest bandwidth usage	Top-App-By-Bandwidth	Traffic
Application with Highest Session Count	Applications with the highest session count	Top-App-By-Sessions	Traffic
Attack with Highest Session Count	Attack with highest session count	Utm-Top-Attack-Source	Attack
Botnet with Highest Session Count	Botnet with the highest session count	Detected-Botnet	Traffic
Destination with Highest Bandwidth	Destination with the highest bandwidth usage	Top-Destinations-By- Bandwidth	Traffic
Destination with Highest Session Count	Destination with the highest session count	Top-Destinations-By- Sessions	Traffic
Highest Bandwidth Consumed (Application) Category	Highest bandwidth consumed by application category	App-Risk-App-Usage-By- Category	Traffic
Highest Bandwidth Consumed (Application)	Highest bandwidth consumed by application	Top-App-By-Bandwidth	Traffic
Highest Bandwidth Consumed (Destination)	Highest bandwidth consumed by destination	Top-Destinations-By- Bandwidth	Traffic
Highest Bandwidth Consumed (P2P Application)	Highest bandwidth consumed by P2P application	Top-P2P-App-By- Bandwidth	Traffic
Highest Bandwidth Consumed (Source)	Highest bandwidth consumed by source	Top-Users-By-Bandwidth	Traffic
Highest Bandwidth Consumed ()Web Category)	Highest bandwidth consumed by website category	Top-Web-Category-by- Bandwidth	Web Filter
Highest Bandwidth Consumed (Website)	Highest bandwidth consumed by website	Top-Web-Sites-by- Bandwidth	Web Filter
Highest Risk Application with Highest Bandwidth	Highest risk application with the highest bandwidth usage	High-Risk-Application- By-Bandwidth	Traffic

Macro Name	Description	Dataset Used	Log Category
Highest Risk Application with Highest Session Count	Highest risk application with the highest session count	High-Risk-Application- By-Sessions	Traffic
Highest Session Count by Application Category	Highest session count by application category	App-Sessions-By- Category	Traffic
Highest Session Count by Application	Highest session count by application	Top-App-By-Sessions	Traffic
Highest Session Count by Attack	Highest session count by attack	Utm-Top-Attack-Source	Attack
Highest Session Count by Botnet	Highest session count by botnet	Detected-Botnet	Traffic
Highest Session Count by Destination	Highest session count by destination	Top-Destinations-By- Sessions	Traffic
Highest Session Count by Highest Severity Attack	Highest session count by highest severity attack	Threat-Attacks-By- Severity	Attack
Highest Session Count by P2P Application	Highest session count by P2P application	Top-P2P-App-By- Sessions	Traffic
Highest Session Count by Source	Highest session count by source	Top-User-Source-By- Sessions	Traffic
Highest Session Count by Virus	Highest session count by virus	Utm-Top-Virus	Traffic
Highest Session Count by Web Category	Highest session count by website category	Top-Web-Category-by- Sessions	Web Filter
Highest Session Count by Website	Highest session count by website	Top-Web-Sites-by- Sessions	Web Filter
Highest Severity Attack with Highest Session Count	Highest severity attack with the highest session count	Threat-Attacks-By- Severity	Attack
P2P Application with Highest Bandwidth	P2P applications with the highest bandwidth usage	Top-P2P-App-By- Bandwidth	Traffic
P2P Application with Highest Session Count	P2P applications with the highest session count	Top-P2P-App-By- Sessions	Traffic
Source with Highest Bandwidth	Source with the highest bandwidth usage	Top-Users-By-Bandwidth	Traffic
Source with Highest Session Count	Source with the highest session count	Top-User-Source-By- Sessions	Traffic
Total Number of Attacks	Total number of attacks detected	Total-Attack-Source	Attack
Total Number of Botnet Events	Total number of botnet events	Total-Number-of-Botnet- Events	Traffic

Macro Name	Description	Dataset Used	Log Category
Total Number of Viruses	Total number of viruses detected	Total-Number-of-Viruses	Traffic
User Details	User details of traffic	Traffic-User-Detail	Traffic
Virus with Highest Session Count	Virus with the highest session count	Utm-Top-Virus	Traffic
Web Category with Highest Bandwidth	Web filtering category with the highest bandwidth usage	Top-Web-Category-by- Bandwidth	Web Filter
Web Category with Highest Session Count	Web filtering category with the highest session count	Top-Web-Category-by- Sessions	Web Filter
Website with Highest Bandwidth	Website with the highest bandwidth usage	Top-Web-Sites-by- Bandwidth	Web Filter
Website with Highest Session Count	Website with the highest session count	Top-Web-Sites-by- Sessions	Web Filter





Copyright© 2019 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 and, in such event, only the specific performance metrics expressly identified in such binding written contract shall be binding on Fortinet. For absolute clarity, any such warranty will be limited to performance in the same ideal conditions as in Fortinet's internal lab tests. In no event does Fortinet make any commitment related to future deliverables, features or development, and circumstances may change such that any forward-looking statements herein are not accurate. Fortinet disclaims in full any covenants, representations, and guarantees pursuant hereto, whether express or implied. Fortinet reserves the right to change, modify, transfer, or otherwise revise this publication without notice, and the most current version of the publication shall be applicable.