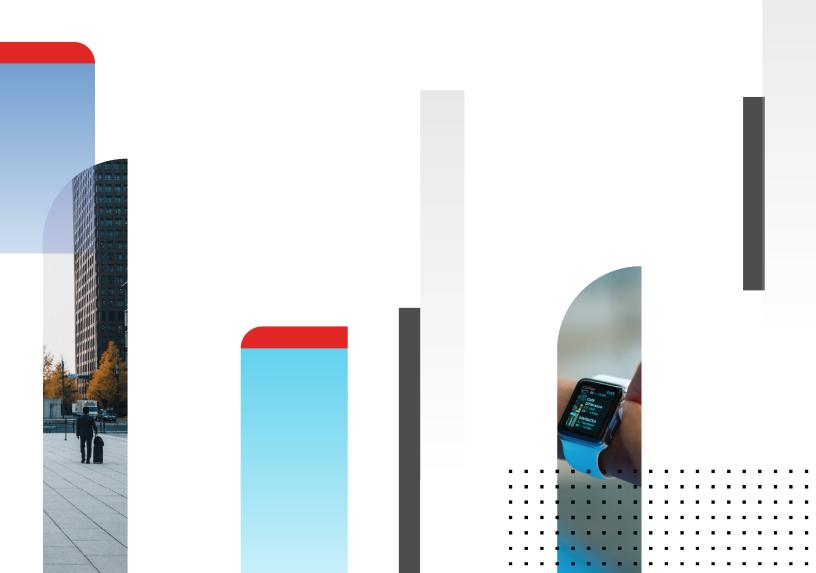
F**E**RTINET.

Dataset Reference

FortiAnalyzer 7.0.4



FORTINET DOCUMENT LIBRARY

https://docs.fortinet.com

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

FORTINET BLOG https://blog.fortinet.com

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

FORTINET TRAINING & CERTIFICATION PROGRAM

https://www.fortinet.com/training-certification

NSE INSTITUTE https://training.fortinet.com

FORTIGUARD CENTER https://www.fortiguard.com

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

FEEDBACK Email: techdoc@fortinet.com



June 8, 2022 FortiAnalyzer 7.0.4 Dataset Reference 05-704-712082-20220608

TABLE OF CONTENTS

Introduction	4
Understanding datasets and macros	4
Dataset Reference List	5
Macro Reference List 29	0
Change Log 29)3

Introduction

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

Understanding datasets and macros

FortiAnalyzer datasets are collections of log messages from monitored devices.

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

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

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

Dataset Reference List

The following tables list the datasets included with FortiAnalyzer. The tables contain the name, SQL query syntax, and log category for each dataset.

Dataset Name	Description	Log Category
Traffic-Bandwidth-Summary-Day-Of- Month	Traffic bandwidth timeline	traffic
<pre>(traffic_in) as traffic_in from timestamp as timestamp, dvid, sr (`unauthuser`), ipstr(`srcip`)) (sentdelta, sentbyte, 0)+coalesc (sentdelta, sentbyte, 0)) as tra in from \$log-traffic where \$filt dstip, epid, euid, user_src, ser</pre>	<pre>t, width) as bandwidth, sum(traffic_out) as ###base(/*tag:rpt_base_t_bndwdth_sess*/se cip, dstip, epid, euid, coalesce(nullifna as user_src, service, count(*) as session e(rcvddelta, rcvdbyte, 0)) as bandwidth, ffic_out, sum(coalesce(rcvddelta, rcvdbyt er and (logflag&(1 32)>0) group by timest vice /*SkipSTART*/order by timestamp desc der by bandwidth desc)### t where \$filter</pre>	<pre>elect \$flex_ a(`user`), nullifna hs, sum(coalesce sum(coalesce ce, 0)) as traffic_ tamp, dvid, srcip, c/*SkipEND*/)base###</pre>

Dataset Name	Description	Log Category
Session-Summary-Day-Of-Month	Number of session timeline	traffic
<pre>select \$flex_timescale(timestamp) as hodex, sum(sessions) as sessions from ###(select timestamp, sum(sessions) as sessions from ###base(/*tag:rpt_base_t_bndwdth_ sess*/select \$flex_timestamp as timestamp, dvid, srcip, dstip, epid, euid, coalesce(nullifna (`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service, count(*) as sessions, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in from \$log-traffic where \$filter and (logflag&(1 32)>0) group by timestamp, dvid, srcip, dstip, epid, euid, user_src, service /*SkipSTART*/order by timestamp desc/*SkipEND*/)base### base_query group by timestamp order by sessions desc)### t where \$filter-drilldown group by hodex order by hodex</pre>		
Dataset Name	Description	Log Category
Top-Users-By-Bandwidth	Bandwidth application top users by bandwidth usage	traffic

select

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

```
sum(traffic_out) as traffic_out,
sum(sessions) as sessions
from
  ###(select user_src, sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_out, sum
  (bandwidth) as bandwidth, sum(sessions) as sessions from ###base(/*tag:rpt_base_t_top_
  app*/select dvid, srcip, dstip, epid, euid, coalesce(nullifna(`user`), nullifna
  (`unauthuser`), ipstr(`srcip`)) as user_src, appid, app, appcat, apprisk, hostname, sum
  (coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as
  traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as
  bandwidth, count(*) as sessions from $log-traffic where $filter and (logflag&(1|32)>0) and
  nullifna(app) is not null group by dvid, srcip, dstip, epid, euid, user_src, appid, app,
  appcat, apprisk, hostname order by sessions desc)base### t group by user_src order by
  sessions desc, bandwidth desc)### t group by user_src having sum(bandwidth)>0 order by
  bandwidth desc
```

Dataset Name	Description	Log Category
Top-App-By-Bandwidth	Top applications by bandwidth usage	traffic
<pre>select app_group_name(app) as app_group_</pre>	up,	
<pre>sum(bandwidth) as bandwidth, sum(traffic_in) as traffic_in, sum(traffic out) as traffic ou</pre>	t,	
sum(sessions) as sessions from		
	apprisk, sum(traffic_in) as traffic_ andwidth, sum(sessions) as sessions f	—
(`unauthuser`), ipstr(`srcip`))	ip, dstip, epid, euid, coalesce(nulli as user_src, appid, app, appcat, appr	isk, hostname, sum
traffic_out, sum(coalesce(sentde	<pre>)) as traffic_in, sum(coalesce(sentde lta, sentbyte, 0)+coalesce(rcvddelta,</pre>	rcvdbyte, 0)) as
nullifna(app) is not null group b	from \$log-traffic where \$filter and (by dvid, srcip, dstip, epid, euid, us	er_src, appid, app,
	by sessions desc)base### t group by a ssions desc, bandwidth desc/*SkipEND* der by bandwidth desc	

Dataset Name	Description	Log Category
Top-User-Source-By-Sessions	Top user source by session count	traffic
count(*) as sessions from \$log	a(`user`), nullifna(`unauthuser`), ips g where \$filter and (logflag&1>0) grou user_src order by sessions desc	

Dataset Name	Description	Log Category
Top-App-By-Sessions	Top applications by session count	traffic
<pre>select app_group,</pre>		

```
sum(sessions) as sessions
from
```

###(select app_group_name(app) as app_group, appcat, service, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from \$log where \$filter and (logflag&(1|32)>0) and nullifna(app) is not null group by app group, appcat, service order by bandwidth desc) ### t group by app group order by sessions desc

Dataset Name	Description	Log Category
Top-Destination-Addresses-By- Sessions	Top destinations by session count	traffic
<pre>select coalesce(nullifna(root_domain(hostname)), ipstr(dstip)) as domain, count(*) as sessions from \$log where \$filter and (logflag&1>0) group by domain order by sessions desc</pre>		
Dataset Name	Description	Log Category

Dataset Name

Dataset Name	200	cription	Log Calegory
Top-Destination-Addresses- Bandwidth	-Ву- Тор	destinations by bandwidth usage	traffic
<pre>select coalesce(nullifna(root_domain(hostr), ipstr(dstip)) as domain, sum(coalesce(sentbyte,) as bandwidth, sum(coalesce(rcvdbyte,) as traffic_in, sum(coalesce(sentbyte,) as traffic_out from</pre>	0)+ coalesce(r 0)	cvdbyte, 0)	

```
$log
where
 $filter
 and (
  logflag&1>0
 )
 and coalesce(
  nullifna(
     root_domain(hostname)
   ),
   ipstr(`dstip`)
 ) is not null
group by
 domain
having
 sum(
   coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
 ) > 0
order by
 bandwidth desc
```

Dataset Name	Description	Log Category
DHCP-Summary-By-Port	Event top dhcp summary	event
<pre>select devintf, mac from ###(select concat(interface \$filter and logid_to_int(logid) group by devintf, mac; create (select concat(interface, '.' int(logid) = 26001 and dhcp_m create temporary table rpt_tm as decimal(18,2)) as percent_ (interface, '.', devid) as devint(logid)=26003 and total>0 order by devintf, itime desc; cli_count from rpt_tmptbl_3 to rpt_tmptbl_2 where not exists</pre>	-	<pre>com \$log where \$last3day_period group by devintf, mac) ### t s select devintf, mac from ### .og where \$filter and logid_to_ :)### t group by devintf, mac; devintf, cast(used*100.0/total distinct on (devintf) concat Elog where \$filter and logid_to_ .time desc/*SkipEND*/)### t percent_of_allocated_ip, new_ punt(mac) as new_cli_count from ere rpt_tmptbl_2.mac=rpt_tmptbl_</pre>

Dataset Name	Description	Log Category
Top-Wifi-Client-By-Bandwidth	Traffic top WiFi client by bandwidth usage	traffic
<pre>select user_src,</pre>		

```
srcssid,
devtype_new,
hostname_mac,
sum(bandwidth) as bandwidth
from
 (
    select
    user_src,
    srcssid,
    get_devtype(srcswversion, osname, devtype) as devtype_new,
    hostname_mac,
    sum(bandwidth) as bandwidth
    from
```

###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user src, ap, srcintf, srcssid, srcssid as ssid, srcmac, srcmac as stamac, coalesce(nullifna (`srcname`), `srcmac`) as hostname mac, max(srcswversion) as srcswversion, max(osname) as osname, max(osversion) as osversion, max(devtype) as devtype, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, count(*) as subtotal from \$log-traffic where \$filter and (logflag&1>0) and (srcssid is not null or dstssid is not null) group by user src, ap, srcintf, srcssid, srcmac, hostname mac /*SkipSTART*/order by bandwidth desc, subtotal desc/*SkipEND*/)### t group by user_src, srcssid, devtype_new, hostname_mac having sum (bandwidth)>0 union all select user_src, ssid as srcssid, null as devtype_new, stamac as hostname mac, sum(bandwidth) as bandwidth from ###(select \$flex timestamp as timestamp, stamac, stamac as srcmac, ap, ssid, ssid as srcssid, user src, sum(coalesce(sentdelta, 0)) as sentdelta, sum(coalesce(rcvddelta, 0)) as rcvddelta, sum(coalesce(sentdelta, 0)+coalesce (rcvddelta, 0)) as bandwidth from (select itime, stamac, ap, ssid, coalesce(`user`, ipstr (`srcip`)) as user src, sentbyte-lag(coalesce(sentbyte, 0)) over (partition by stamac order by itime) as sentdelta, rcvdbyte-lag(coalesce(rcvdbyte, 0)) over (partition by stamac order by itime) as rcvddelta from \$log-event where \$filter and subtype='wireless' and stamac is not null and ssid is not null and action in ('sta-wl-bridge-traffic-stats', 'reassoc-reg', 'assoc-reg')) as t group by timestamp, stamac, ap, ssid, user src /*SkipSTART*/order by bandwidth desc/*SkipEND*/)### t where user src is not null group by user src, ssid, devtype new, stamac having sum(bandwidth)>0) t group by user src, srcssid, devtype new, hostname mac order by bandwidth desc

Dataset Name	Description	Log Category
Traffic-History-By-Active-User	Traffic history by active user	traffic
<pre>select \$flex_timescale(timestamp) as count(distinct(user_src)) as total_user from</pre>		
<pre>bndwdth_sess*/select \$flex_times (nullifna(`user`), nullifna(`una sessions, sum(coalesce(sentdelta bandwidth, sum(coalesce(sentdelta rcvdbyte, 0)) as traffic_in from timestamp, dvid, srcip, dstip, e</pre>	<pre>c, sum(sessions) as sessions from ## stamp as timestamp, dvid, srcip, dst authuser`), ipstr(`srcip`)) as user_ a, sentbyte, 0)+coalesce(rcvddelta, ca, sentbyte, 0)) as traffic_out, su a \$log-traffic where \$filter and (log pid, euid, user_src, service /*Skip ery group by timestamp, user_src or bodex order by hodex</pre>	tip, epid, euid, coalesce src, service, count(*) as rcvdbyte, 0)) as um(coalesce(rcvddelta, ogflag&(1 32)>0) group by pSTART*/order by timestamp

Dataset Name	Description	Log Category
Top-Allowed-Websites-By-Requests	UTM top allowed web sites by request	traffic
select		
hostname, catdesc,		
count(*) as requests		
from		
\$log		
where		
\$filter		
and (
logflag&1>0		
)		
and utmevent in (
•	word', 'web-content', 'command-block',	-
hostname is not null and (utmac hostname, catdesc order by requ	ction not in ('block', 'blocked') or act mests desc	cion!='deny') group by
Dataset Name	Description	Log Category

Top-50-Websites-By-BandwidthWebfilter top allowed web sites by bandwidth usage

select domain,

```
string_agg(
```

```
distinct catdesc,
```

& #039;, ') as agg_catdesc, sum(bandwidth) as bandwidth, sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_out from ###(select coalesce(nullifna(hostname), ipstr(`dstip`)) as domain, catdesc, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum (coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out from \$logtraffic where \$filter and (logflag&l>0) and utmaction!='blocked' and (countweb>0 or ((logver is null or logver<50200000) and (hostname is not null or utmevent in ('webfilter', 'bannedword', 'web-content', 'command-block', 'script-filter'))) group by domain, catdesc having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 /*SkipSTART*/order by bandwidth desc/*SkipEND*/)### t group by domain, catdesc order by bandwidth desc

Dataset Name	Description	Log Category
Top-Blocked-Websites	UTM top blocked web sites by request	traffic
•	ord', 'web-content', 'command-block', 'script- ion in ('block', 'blocked') or action='deny')	

webfilter

Dataset Name	Description	Log Category
Top-Web-Users-By-Request	UTM top web users by request	traffic
<pre>get_devtype(srcswversion, osn (coalesce(sentbyte, 0)+coales traffic_in, sum(coalesce(sent \$filter and (logflag&1>0) and 'command-block', 'script-filt</pre>	a(`user`), nullifna(`unauthuser`), ipst ame, devtype) as devtype_new, srcname, ce(rcvdbyte, 0)) as bandwidth, sum(coal- byte, 0)) as traffic_out, count(*) as r utmevent in ('webfilter', 'banned-word er') group by user_src, devtype_new, sr group by user_src, devtype_new, srcname	action, utmaction, sum esce(rcvdbyte, 0)) as equests from \$log where ', 'web-content', cname, action, utmaction
Dataset Name	Description	Log Category
Top-Allowed-WebSites-By-Bandwidt	h UTM top allowed websites by bandwidth usag	e traffic
<pre>select appid, hostname, catdesc, sum(coalesce(sentbyte, 0)+ co) as bandwidth, sum(coalesce(rcvdbyte, 0)</pre>	alesce(rcvdbyte, 0)	

Dataset Name	Description	Log Category
Top-Blocked-Web-Users	UTM top blocked web users	traffic
<pre>select user_src, devtype_new, srcname, sum(requests) as requests</pre>		

from

###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, get_devtype(srcswversion, osname, devtype) as devtype_new, srcname, action, utmaction, 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 requests from \$log where \$filter and (logflag&1>0) and utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') group by user_src, devtype_new, srcname, action, utmaction order by requests desc)### t where (utmaction in ('block', 'blocked') or action='deny') group by user src, devtype new, srcname order by requests desc

Dataset Name	Description	Log Category
Top-20-Web-Users-By-Bandwidth	Webfilter top web users by bandwidth usage	webfilter
select		
coalesce(
f_user,		
euname,		
ipstr(`srcip`)		
) as user_src,		
coalesce(
epname,		
ipstr(`srcip`)		
) as ep_src,		
sum(bandwidth) as bandwidth,		
<pre>sum(traffic_in) as traffic_in, </pre>		
<pre>sum(traffic_out) as traffic_ou from</pre>		
(
select		
dvid,		
f user,		
srcip,		
ep id,		
eu id,		
sum(bandwidth) as bandwidt	h,	
<pre>sum(traffic_in) as traffic</pre>	_in,	
<pre>sum(traffic_out) as traffi</pre>	.c_out	
from		
	<pre>nullifna(`user`), nullifna(`unauthuser`))</pre>	
	else epid end) as ep_id, (case when euid<10	
	<pre>(sentbyte, 0)+coalesce(rcvdbyte, 0)) as ba</pre>	
	<pre>ic_in, sum(coalesce(sentbyte, 0)) as traff</pre>	—
	ag&1>0) and (countweb>0 or ((logver is nul	
	is not null or utmevent in ('webfilter', '	
	<pre>'script-filter')))) group by dvid, f_user, 0)+coalesce(rcvdbyte, 0))>0 /*SkipSTART*/</pre>	
	lvid, f user, srcip, ep id, eu id order by	
	cmac as epmac, dvid from \$ADOM EPEU DEVMAP	
	l and dm.vd=dt.vd) t2 on t1.ep id=t2.epid a	
	eft join \$ADOM ENDPOINT t3 on t1.ep id=t3.	—
	ENDUSER t4 on t1.eu_id=t4.euid group by us	
order by bandwidth desc		

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

```
select
  user_src,
  devtype_new,
  srcname,
  sum(bandwidth) as bandwidth,
  sum(traffic_in) as traffic_in,
  sum(traffic_out) as traffic_out
from
```

###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, get_devtype(srcswversion, osname, devtype) as devtype_new, srcname, action, utmaction, 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 requests from \$log where \$filter and (logflag&l>0) and utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') group by user_src, devtype_new, srcname, action, utmaction order by requests desc)### t group by user_src, devtype_new, srcname having sum(bandwidth)>0 order by bandwidth desc

Dataset Name	Description	Log Category
Top-Video-Streaming-Websites-By- Bandwidth	UTM top video streaming websites by bandwidth usage	traffic
<pre>select appid, hostname, sum(coalesce(sentbyte, 0)+ coales) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out from \$log where \$filter and (logflag&l>0) and catdesc in (& #039;Streaming Media and De (sentbyte, 0)+coalesce(rcvdbyte, </pre>	ownload') group by appid, hostname having sur	n(coalesce
Dataset Name	Description	Log Category
Top-Email-Senders-By-Count	Default top email senders by count	traffic

```
select
```

```
user_src,
sum(requests) as requests
from
  ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
service, count(*) as requests from $log where $filter and (logflag&1>0) group by user_src,
```

service order by requests desc)### t where service in ('smtp', 'SMTP', '25/tcp', '587/tcp', 'smtps', 'SMTPS', '465/tcp') group by user_src order by requests desc

Dataset Name	Description	Log Category
Top-Email-Receivers-By-Count	Default email top receivers by count	traffic

```
select
```

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

from

###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service, count(*) as requests from \$log where \$filter and (logflag&1>0) group by user_src, service order by requests desc)### t where service in ('pop3', 'POP3', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/tcp', 'pop3s', 'POP3S', '995/tcp') group by user_ src order by requests desc

Dataset Name	Description	Log Category
Top-Email-Senders-By-Bandwidth	Default email top senders by bandwidth usage	traffic
	<pre>lesce(rcvdbyte, 0) tcp', '587/tcp', 'smtps', 'SMTPS', '465/tcp' te, 0)+coalesce(rcvdbyte, 0))>0 order by band</pre>	-
Dataset Name	Description	Log Category
Top-Email-Receivers-By-Bandwidth	Default email top receivers by bandwidth usage	traffic
<pre>select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user src</pre>		

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

\$loq

```
where
 $filter
 and (
   logflag&1>0
 and service in (
   & #039;pop3', 'POP3', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/tcp',
'pop3s', 'POP3S', '995/tcp') group by user src having sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0))>0 order by bandwidth desc
```

Dataset Name	Description	Log Category
Top-Malware-By-Name	UTM top virus	virus
select		

virus, max(virusid s) as virusid, (

case when virus like & #039;Riskware%' then 'Spyware' when virus like 'Adware%' then 'Adware' else 'Virus' end) as malware_type, sum(totalnum) as totalnum from ###(select virus, virusid to str(virusid, eventtype) as virusid s, count(*) as totalnum from \$log where \$filter and (eventtype is null or logver>=502000000) and nullifna(virus) is not null group by virus, virusid s /*SkipSTART*/order by totalnum desc/*SkipEND*/)### t group by virus, malware type order by totalnum desc

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

select

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

case when virus like & #039;Riskware%' then 'Spyware' when virus like 'Adware%' then 'Adware' else 'Virus' end) as malware type, sum(totalnum) as totalnum from ###(select virus, virusid_to_str(virusid, eventtype) as virusid_s, count(*) as totalnum from \$log where \$filter and (eventtype is null or logver>=502000000) and nullifna(virus) is not null group by virus, virusid s /*SkipSTART*/order by totalnum desc/*SkipEND*/)### t group by virus, malware type order by totalnum desc

Dataset Name	Description	Log Category
Top-Virus-Victim	UTM top virus user	virus
select		
user_src,		
sum(totalnum) as tota	lnum	
from		
###(select coalesce(n	ullifna(`user`), ipstr(`srcip`)) as user_	src, eventtype, logver,
virus, count(*) as totalnum from \$log where \$filter group by user_src, eventtype, logver,		
virus /*SkipSTART*/orde	r by totalnum desc/*SkipEND*/)### t where	(eventtype is null or

ere (eve `/)### て logver>=502000000) and nullifna(virus) is not null group by user_src order by totalnum desc

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

```
select
    user_src,
    sum(totalnum) as totalnum
from
    ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, eventtype, logver,
    count(*) as totalnum from $log where $filter group by user_src, eventtype, logver
/*SkipSTART*/order by totalnum desc/*SkipEND*/)### t group by user_src order by totalnum
desc
```

Dataset Name	Description	Log Category
Top-Attack-Victim	UTM top attack dest	attack
<pre>select victim, count(*) as totalnum from (select</pre>		
	x #039;incoming' THEN srcip ELSE d is not null group by victim order	

Dataset Name	Description	Log Category
Top-Static-IPSEC-Tunnels-By- Bandwidth	Top static IPsec tunnels by bandwidth usage	event
<pre>select vpn_name, sum(bandwidth) as bandwidth, sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_out from (select devid, vd, remip, tunnelid, vpn_name, (</pre>	t	
<pre>case when min(s_time) = ma in) - min(min_traffic_in) end) as traffic_in, (</pre>	<pre>ax(e_time) then max(max_traffic_in) else max(</pre>	max_traffic_
<pre>out) - min(min_traffic_out) end) as traffic_out, (</pre>	ax(e_time) then max(max_traffic_out) else max	
<pre>else max(max_traffic_in) - min(min end) as bandwidth from</pre>	<pre>ax(e_time) then max(max_traffic_in)+ max(max_ n_traffic_in)+ max(max_traffic_out)- min(min_ o, vpn_trim(vpntunnel) as vpn_name, tunnelid,</pre>	traffic_out)

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

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

null group by devid, vd, user_src, remip, tunnelid, tunneltype order by max_traffic desc)###
t where tunneltype='ssl-tunnel' group by devid, vd, user_src, remip, tunnelid) tt where
bandwidth>0 group by user_src, remote_ip order by bandwidth desc

Dataset Name	Description	Log Category
Top-Dial-Up-IPSEC-Tunnels-By- Bandwidth	Top dial up IPsec tunnels by bandwidth usage	event
<pre>select vpn_name, sum(bandwidth) as bandwidth, sum(traffic_in) as traffic_in sum(traffic_out) as traffic_of from (select devid, vd, tunnelid, remip, vpn_name, (case when min(s_time)= in) - min(min_traffic_in) end) as traffic_in, (case when min(s_time)= out) - min(min_traffic_out) end) as traffic_out, (case when min(s_time)= else max(max_traffic_in) - min(re end) as bandwidth from ###(select devid, vd, ref (coalesce(sentbyte, 0)) as max_ max(coalesce(crvdbyte, 0)+coale as min_traffic_out, min(coalesce s_time, max(coalesce(dtime, 0))</pre>		<pre>max(max_traffic_ max_traffic_out) nin_traffic_out) lid, tunnelip, max max_traffic_in, esce(sentbyte, 0)) lesce(dtime, 0)) as type='vpn' and</pre>
tunnelip order by max_traffic of	<pre>not null group by devid, vd, remip, vpn_nar desc)### t where not (tunnelip is null or tu _name, tunnelid) tt group by vpn_name having dwidth desc</pre>	unnelip='0.0.0.0')
Dataset Name	Description	Log Category
Top-Dial-Up-IPSEC-Users-By-	Top dial up IPsec users by bandwidth usage	event

```
select
  coalesce(
    xauthuser_agg,
    user_agg,
```

Bandwidth

bandwidth desc

```
ipstr(`remip`)
  ) as user_src,
  remip,
  from dtime(
  min(s time)
  ) as start time,
  sum(bandwidth) as bandwidth,
  sum(traffic in) as traffic in,
  sum(traffic out) as traffic out
from
   select
     devid,
     vd.
     string agg(
       distinct xauthuser agg,
       & #039; ') as xauthuser_agg, string_agg(distinct user_agg, ' ') as user_agg, remip,
tunnelid, min(s time) as s time, max(e time) as e time, (case when min(s time)=max(e time)
then max(max traffic in)+max(max traffic out) else max(max traffic in)-min(min traffic
in)+max(max_traffic_out)-min(min_traffic_out) end) as bandwidth, (case when min(s_time)=max
(e_time) then max(max_traffic_in) else max(max_traffic_in)-min(min_traffic_in) end) as
traffic in, (case when min(s time)=max(e time) then max(max traffic out) else max(max
traffic out)-min(min traffic out) end) as traffic out from ###(select devid, vd, remip,
nullifna(`xauthuser`) as xauthuser agg, nullifna(`user`) as user agg, tunnelid, min(coalesce
(dtime, 0)) as s_time, max(coalesce(dtime, 0)) as e_time, max(coalesce(duration,0)) as max_
duration, min(coalesce(duration,0)) as min duration, min(coalesce(sentbyte, 0)) as min
traffic out, min(coalesce(rcvdbyte, 0)) as min traffic in, max(coalesce(sentbyte, 0)) as
max traffic out, max(coalesce(rcvdbyte, 0)) as max traffic in, max(coalesce(rcvdbyte,
0)+coalesce(sentbyte, 0)) as max traffic 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 max traffic desc)### t group
by devid, vd, remip, tunnelid) tt where bandwidth>0 group by user src, remip order by
```

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

& #039; ') as xauthuser agg, string agg(distinct user agg, ' ') as user agg, tunnelid, min(s time) as s time, max(e time) as e time, (case when min(s time)=max(e time) then max(max duration) else max(max duration)-min(min duration) end) as duration, (case when min(s time)=max(e time) then max(max traffic in)+max(max traffic out) else max(max traffic in)-min(min traffic in)+max(max traffic out)-min(min traffic out) end) as bandwidth, (case when min(s time) = max(e time) then max(max traffic in) else max(max traffic in) - min(min traffic in) end) as traffic in, (case when min(s time)=max(e_time) then max(max_traffic_out) else max(max traffic out)-min(min traffic out) end) as traffic out from ###(select devid, vd, remip, nullifna(`xauthuser`) as xauthuser agg, nullifna(`user`) as user agg, tunnelid, min(coalesce(dtime, 0)) as s time, max(coalesce(dtime, 0)) as e time, max(coalesce (duration, 0)) as max duration, min(coalesce(duration, 0)) as min duration, min(coalesce (sentbyte, 0)) as min traffic out, min(coalesce(rcvdbyte, 0)) as min traffic in, max (coalesce(sentbyte, 0)) as max traffic out, max(coalesce(rcvdbyte, 0)) as max traffic in, max(coalesce(rcvdbyte, 0)+coalesce(sentbyte, 0)) as max traffic 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 max_ traffic desc)### t group by devid, vd, remip, tunnelid) tt where bandwidth>0 group by user src order by duration desc

Dataset Name	Description	Log Category
Top-SSL-VPN-Web-Mode-Users-By- Bandwidth	Top SSL VPN web mode users by bandwidth usage	event
—	ax(e_time) then max(max_traffic_in)+ max(max_t n traffic_in)+ max(max_traffic_out) - min(min t	_
end) as bandwidth, (ax(e_time) then max(max_traffic_in) else max(m	_

```
in) - min(min_traffic_in) end
    ) as traffic_in,
    (
        case when min(s_time) = max(e_time) then max(max_traffic_out) else max(max_traffic_
out) - min(min_traffic_out) end
    ) as traffic_out
    from
```

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

Dataset Name	Description	Log Category
Top-SSL-VPN-Web-Mode-Users-By- Duration	Top SSL VPN web mode users by duration	event
<pre>select user_src, remip as remote_ip, from_dtime(min(s_time)) as start_time, sum(duration) as duration from (select</pre>		
<pre>devid, vd, user_src, remip, tunnelid, min(s_time) as s_time, (case when min(s_time) = m min(min_duration) end</pre>	wax(e_time) then max(max_duration) else max	(max_duration)-
<pre>tunnelid, tunneltype, max(coales as min_duration, min(coalesce(dt (coalesce(sentbyte, 0)) as min_t max(coalesce(sentbyte, 0)) as ma max(coalesce(rcvdbyte, 0)+coales subtype='vpn' and tunneltype lik 'tunnel-up') and coalesce(nullif null group by devid, vd, user_sr</pre>	p, coalesce(nullifna(`user`), ipstr(`remip cce(duration,0)) as max_duration, min(coale time, 0)) as s_time, max(coalesce(dtime, 0) traffic_out, min(coalesce(rcvdbyte, 0)) as ax_traffic_out, max(coalesce(rcvdbyte, 0)) cce(sentbyte, 0)) as max_traffic from \$log te 'ssl%' and action in ('tunnel-stats', 't fna(`user`), ipstr(`remip`)) is not null an cc, remip, tunnelid, tunneltype order by ma pup by devid, vd, user_src, remip, tunnelid fation desc	<pre>esce(duration,0))) as e_time, min min_traffic_in, as max_traffic_in, where \$filter and cunnel-down', dd tunnelid is not ux_traffic desc)###</pre>



Dataset Name	Description	Log Category
vpn-Top-Dial-Up-VPN-Users-By- Duration	Top dial up VPN users by duration	event

```
select
  coalesce(
   xauthuser agg,
    user agg,
    ipstr(`remip`)
  ) as user src,
  t type as tunneltype,
  from dtime(
   min(s time)
  ) as start time,
  sum(duration) as duration,
  sum(bandwidth) as bandwidth,
  sum(traffic in) as traffic in,
  sum(traffic out) as traffic out
from
  (
    select
      devid,
      vd,
      remip,
      string_agg(
        distinct xauthuser agg,
```

& #039; ') as xauthuser agg, string agg(distinct user agg, ' ') as user agg, t type, tunnelid, min(s_time) as s_time, max(e_time) as e_time, (case when min(s_time)=max(e_time) then max(max_duration) else max(max_duration)-min(min_duration) end) as duration, (case when min(s_time) = max(e_time) then max(max_traffic_in) + max(max_traffic_out) else max(max_traffic_ in)-min(min traffic in)+max(max traffic out)-min(min traffic_out) end) as bandwidth, (case when min(s time)=max(e time) then max(max traffic in) else max(max traffic in)-min(min traffic in) end) as traffic in, (case when min(s time)=max(e time) then max(max traffic out) else max(max traffic out)-min(min traffic out) end) as traffic out from ###(select devid, vd, remip, nullifna(`xauthuser`) as xauthuser agg, nullifna(`user`) as user agg, (case when tunneltype like 'ipsec%' then 'ipsec' else tunneltype end) as t type, tunnelid, tunnelip, min(coalesce(dtime, 0)) as s time, max(coalesce(dtime, 0)) as e time, max(coalesce (duration, 0)) as max duration, min(coalesce(duration, 0)) as min_duration, min(coalesce (sentbyte, 0)) as min traffic out, min(coalesce(rcvdbyte, 0)) as min traffic in, max (coalesce(sentbyte, 0)) as max traffic out, max(coalesce(rcvdbyte, 0)) as max traffic in, max(coalesce(rcvdbyte, 0)+coalesce(sentbyte, 0)) as max traffic, sum((case when action='tunnel-up' then 1 else 0 end)) as tunnelup from \$log where \$filter and subtype='vpn' and (tunneltype like 'ipsec%' or tunneltype like 'ssl%') and action in ('tunnel-up', 'tunnel-stats', 'tunnel-down') and tunnelid is not null and tunnelid!=0 group by xauthuser agg, user agg, devid, vd, remip, t type, tunnelid, tunnelip order by max traffic desc)### t where (t type like 'ssl%' or (t type like 'ipsec%' and not (tunnelip is null or tunnelip='0.0.0.0'))) group by devid, vd, remip, t_type, tunnelid) tt where bandwidth>0 group by user src, tunneltype order by duration desc

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

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

###(select \$flex_timestamp as timestamp, devid, vd, remip, tunnelid, max((case when action='tunnel-up' then 1 else 0 end)) as tunnelup, max(coalesce(sentbyte, 0)) as traffic_ out, max(coalesce(rcvdbyte, 0)) as traffic_in from \$log where \$filter and subtype='vpn' and (tunneltype like 'ipsec%' or tunneltype like 'ssl%') and action in ('tunnel-up', 'tunnelstats', 'tunnel-down') and tunnelid is not null group by timestamp, devid, vd, remip, tunnelid /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t group by timestamp, devid, vd, remip, tunnelid having max(tunnelup) > 0 and max(traffic_in)+max(traffic_out)>0) t group by hodex order by total_num desc

Dataset Name	Description	Log Category
vpn-Failed-Login-Atempts	VPN failed logins	event
<pre>select f_user, tunneltype, sum(total num) as total n</pre>	ıım	
from	fna(`xauthuser`), `user`) as f_user	r, tunneltype, count(*) as
(tunneltype, 3)='ssl') and	<pre>filter and subtype='vpn' and (tunne action in ('ssl-login-fail', 'ipsec lifna(`user`)) is not null group by order by total num desc</pre>	c-login-fail') and coalesce

Dataset Name	Description	Log Category
vpn-Authenticated-Logins	VPN authenticated logins	event
select		
coalesce(
xauthuser_agg,		
user_agg,		
ipstr(`remip`)		
) as f_user,		
t_type as tunneltype,		
from_dtime(min(s time)		
) as start time,		
sum(total num) as total num,		
sum(duration) as duration		
from		
(
select		
string_agg(
distinct xauthuser_agg, & #039; ') as xauthuser a	agg, string agg(distinct user agg, ' ') as use	er agg, t type,
devid, vd, remip, tunnelid, min(s	s_time) as s_time, max(e_time) as e_time, (cas uration) else max(max duration)-min(min durat:	se when min(s_

duration, (case when min(s time)=max(e time) then max(max traffic in)+max(max traffic out) else max(max_traffic_in)-min(min_traffic_in)+max(max_traffic_out)-min(min_traffic_out) end) as bandwidth, (case when min(s_time)=max(e_time) then max(max_traffic in) else max(max_ traffic in)-min(min traffic in) end) as traffic in, (case when min(s time)=max(e time) then max(max traffic out) else max(max traffic out)-min(min traffic out) end) as traffic out, sum (tunnelup) as total num from ###(select devid, vd, remip, nullifna(`xauthuser`) as xauthuser agg, nullifna(`user`) as user agg, (case when tunneltype like 'ipsec%' then 'ipsec' else tunneltype end) as t type, tunnelid, tunnelip, min(coalesce(dtime, 0)) as s time, max(coalesce(dtime, 0)) as e time, max(coalesce(duration,0)) as max duration, min (coalesce(duration,0)) as min duration, min(coalesce(sentbyte, 0)) as min traffic out, min (coalesce(rcvdbyte, 0)) as min traffic in, max(coalesce(sentbyte, 0)) as max traffic out, max(coalesce(rcvdbyte, 0)) as max traffic in, max(coalesce(rcvdbyte, 0)+coalesce(sentbyte, 0)) as max traffic, sum((case when action='tunnel-up' then 1 else 0 end)) as tunnelup from \$log where \$filter and subtype='vpn' and (tunneltype like 'ipsec%' or tunneltype like 'ssl%') and action in ('tunnel-up', 'tunnel-stats', 'tunnel-down') and tunnelid is not null and tunnelid!=0 group by xauthuser agg, user agg, devid, vd, remip, t type, tunnelid, tunnelip order by max traffic desc)### t group by t type, devid, vd, remip, tunnelid having max(tunnelup) > 0) tt where bandwidth>0 group by f user, tunneltype order by total num desc

Dataset Name	Description	Log Category
vpn-Traffic-Usage-Trend-VPN- Summary	VPN traffic usage trend	event
<pre>select hodex, sum(ssl_traffic_bandwidth)</pre>		
<pre>sum(ipsec_traffic_bandwidth from (</pre>	as ipsec_bandwidth	

```
select
  $flex_timescale(timestamp) as hodex,
  devid,
  vd,
  remip,
  tunnelid,
  (
```

case when t_type like & #039;ssl%' then (case when min(s time)=max(e time) then max (max_traffic_in) +max(max_traffic_out) else max(max_traffic_in) -min(min_traffic_in) +max(max_ traffic_out)-min(min_traffic_out) end) else 0 end) as ssl_traffic_bandwidth, (case when t_ type like 'ipsec%' then (case when min(s time)=max(e time) then max(max traffic in)+max(max traffic out) else max(max traffic in)-min(min traffic in)+max(max traffic out)-min(min traffic out) end) else 0 end) as ipsec traffic bandwidth, min(s time) as s time, max(e time) as e time from ###(select \$flex timestamp as timestamp, devid, vd, remip, tunnelid, (case when tunneltype like 'ipsec' then 'ipsec' else tunneltype end) as t type, (case when action='tunnel-up' then 1 else 0 end) as tunnelup, max(coalesce(sentbyte, 0)) as max traffic out, max(coalesce(rcvdbyte, 0)) as max traffic in, min(coalesce(sentbyte, 0)) as min_traffic_out, min(coalesce(rcvdbyte, 0)) as min_traffic_in, min(coalesce(dtime, 0)) as s time, max(coalesce(dtime, 0)) as e time from \$log where \$filter and subtype='vpn' and (tunneltype like 'ipsec%' or tunneltype like 'ssl%') and action in ('tunnel-up', 'tunnelstats', 'tunnel-down') and tunnelid is not null and tunnelid!=0 group by timestamp, devid, vd, remip, t type, tunnelid, action /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t group by hodex, devid, t type, vd, remip, tunnelid) tt group by hodex order by hodex

Dataset Name	Description	Log Category
Top-S2S-IPSEC-Tunnels-By- Bandwidth-and-Availability	Top S2S IPsec tunnels by bandwidth usage and avail	event
<pre>sent_beg, max(coalesce(sentbyte, max(coalesce(rcvdbyte, 0)) as rc (coalesce(duration, 0)) as durat action='tunnel-stats' and tunnel tunnelip='0.0.0.0') and nullifna group by tunnelid, tunneltype, v tunnelid/*SkipEND*/)### t group</pre>	s traffic_out, s traffic_in, vd_end - rcvd_beg	as rcvd_beg, n_beg, max ypn' and tunnelid!=0 order by
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,
remip,
sum(traffic_out) as traffic_out,
sum(traffic_in) as traffic_in,
sum(bandwidth) as bandwidth,
sum(uptime) as uptime
from
  (
    select
    user_src,
    remip,
    tunnelid,
    devid,
```

```
vd,
sum(sent_end - sent_beg) as traffic_out,
sum(rcvd_end - rcvd_beg) as traffic_in,
sum(
   sent_end - sent_beg + rcvd_end - rcvd_beg
) as bandwidth,
sum(duration_end - duration_beg) as uptime
from
```

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

Dataset Name	Description	Log Category
Top-SSL-Tunnel-Mode-By-Bandwidth- and-Availability	Top SSL tunnel users by bandwidth usage and avail	event
<pre>select user_src, remote_ip, sum(traffic_out) as traffic_ou sum(traffic_in) as traffic_in, sum(bandwidth) as bandwidth, sum(uptime) as uptime from (select user_src, remip as remote_ip, tunnelid, devid, vd, sum(sent_end - sent_beg) a sum(sent_end - sent_beg + rcm) as bandwidth,</pre>	s traffic_out, s traffic_in, vd_end - rcvd_beg	
sum(duration_end - duration	n_beg) as uptime	

```
from
```

###(select tunnelid, tunneltype, 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 coalesce(nullifna (`user`), ipstr(`remip`)) is not null and tunnelid is not null group by tunnelid, tunneltype, user_src, remip, devid, vd /*SkipSTART*/order by tunnelid/*SkipEND*/)### t where tunneltype in ('ssl-tunnel', 'ssl') group by user_src, remote_ip, tunnelid, devid, vd order by bandwidth desc) t where bandwidth>0 group by user_src, remote_ip order by bandwidth desc

Dataset Name	Description	Log Category
Top-SSL-Web-Mode-By-Bandwidth- and-Availability	Top SSL web users by bandwidth usage and avail	event
<pre>select user_src, remote_ip, sum(traffic_out) as traffic_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 sum(rcvd_end - rcvd_beg) as sum(sent_end - sent_beg + rcv) as bandwidth, sum(duration_end - duration from</pre>	s traffic_out, s traffic_in, rd_end - rcvd_beg	
<pre>###(select tunnelid, tunnel src, remip, devid, vd, min(coales</pre>	<pre>type, coalesce(nullifna(`user`), ipstr(`re sce(sentbyte, 0)) as sent_beg, max(coalesce 0)) as rcvd beg, max(coalesce(rcvdbyte, 0)</pre>	e(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 coalesce(nullifna (`user`), ipstr(`remip`)) is not null and tunnelid is not null group by tunnelid, tunneltype, user_src, remip, devid, vd /*SkipSTART*/order by tunnelid/*SkipEND*/)### t where tunneltype='ssl-web' group by user_src, remote_ip, tunnelid, devid, vd having sum(sent_endsent_beg+rcvd_end-rcvd_beg)>0 order by bandwidth desc) t where bandwidth>0 group by user_ src, remote_ip order by bandwidth desc

Dataset Name	Description	Log Category
Admin-Login-Summary	Event admin login summary	event
<pre>select f_user, ui, sum(login) as total_num, sum(login_duration) as total sum(config_change) as total from (select `user` as f_user, ui, (case when logid_to_in) as login,</pre>		

```
(
        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
      $loq
    where
      $filter
      and nullifna(`user`) is not null
      and logid_to_int(logid) in (32001, 32003)
  ) t
group by
  f user,
  ui
having
  sum(login) + sum(config change) > 0
order by
  total_num desc
```

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

select

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

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

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

order by total_failed desc

Dataset Name	Description	Log Category
System-Summary-By-Severity	Event system summary by severity	event
select		

```
severity_tmp as severity,
sum(count) as total_num
rem
```

from

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

Dataset Name	Description	Log Category
System-Summary-By-Date	Event system summary by date	event
'emergency') then 1 else 0 end) end) as high, sum(case when lev	<pre>timestamp, sum(case when level i as critical, sum(case when level el = 'warning' then 1 else 0 end) oup by timestamp /*SkipSTART*/ord</pre>	= 'error' then 1 else 0 as medium from \$log where
Dataset Name	Description	Log Category
Important-System-Summary-By-Date	Event system summary by date	event
<pre>select \$flex_timescale(timestamp) as sum(critical) as critical, sum(high) as high,</pre>	dom,	

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

Dataset Name	Description	Log Category
System-Critical-Severity-Events	Event system critical severity events	event
select msg_desc as msg,		

```
FortiAnalyzer 7.0.4 Dataset Reference Fortinet Inc.
```

severity tmp as severity,

```
sum(count) as counts
from
```

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

Dataset Name	Description	Log Category
System-High-Severity-Events	Event system high severity events	event
'alert', 'emergency') then 'C then 'Medium' when level='not count from \$log where \$filter	a(logdesc), msg) as msg_desc, (case when Critical' when level='error' then 'High' Lice' then 'Low' else 'Info' end) as seve and subtype='system' group by msg_desc, Nesc/*SkipEND*/)### t where severity_tmp= desc	<pre>when level='warning' rity_tmp, count(*) as severity_tmp</pre>
Detect Name	Description	

Dataset Name	Description	Log Category
System-Medium-Severity-Events	Event system medium severity events	event

```
select
```

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

from

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

Dataset Name	Description	Log Category
utm-drilldown-Top-Traffic-Summary	UTM drilldown traffic summary	traffic
select srcip, srcname from		
<pre>trom ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, srcip, srcname, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from \$log where \$filter and (logflag&1>0) group by user_src, srcip, srcname order by bandwidth desc)### t where \$filter-drilldown group by srcip, srcname</pre>		

Dataset Name	Description	Log Category
utm-drilldown-Top-User-Destination	UTM drilldown top user destination	traffic
<pre>select appid, app, dstip, sum(sessions) as sessions,</pre>		

from

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

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

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

sum (bandwidth) as bandwidth

from

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

Dataset Name	Description	Log Category
utm-drilldown-Email-Receivers- Summary	UTM drilldown email receivers summary	traffic

select

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

###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, recipient, count(*) as requests, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from \$log where \$filter and (logflag&1>0) and recipient is not null and service in ('pop3', 'POP3', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/tcp', 'pop3s', 'POP3S', '995/tcp') group by user_src, recipient order by requests desc)### t where \$filterdrilldown

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

select
 recipient,
 sum(bandwidth) as bandwidth
from
 ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
recipient, count(*) as requests, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as
bandwidth from \$log where \$filter and (logflag&1>0) and recipient is not null and service in
 ('pop3', 'POP3', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/tcp', 'pop3s',
'POP3S', '995/tcp') group by user_src, recipient order by requests desc)### t where \$filterdrilldown group by recipient having sum(bandwidth)>0 order by bandwidth desc

Dataset Name	Description	Log C	ategory
utm-drilldown-Top-Email-Senders-By- Bandwidth	UTM drilldown top email senders	traffic	
select sender, sum(bandwidth) as bandwidth			
<pre>from ###(select coalesce(nullifna(``</pre>	user`), nullifna(`unauthuser`),	<pre>ipstr(`srcip`)) as us</pre>	ser_src,

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

Dataset Name	Description	Log Category		
utm-drilldown-Top-Allowed-Websites- By-Bandwidth	UTM drilldown top allowed web sites by bandwidth	traffic		
<pre>select appid, hostname, sum(bandwidth) as bandwidth from ####(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, appid, hostname, (case when utmaction in ('block', 'blocked') then 1 else 0 end) as blocked, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from \$log-traffic where \$filter and (logflag&1>0) and (countweb>0 or ((logver is null or logver<50200000) 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</pre>				
Dataset Name	Description	Log Category		
utm-drilldown-Top-Blocked-Websites- By-Request	UTM drilldown top blocked web sites by request	webfilter		
select appid, hostname,				

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

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

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

select

virus, sum(totalnum) as totalnum from

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

Dataset Name	Description	Log Category
utm-drilldown-Top-Attacks	UTM drilldown top attacks by name	attack
select attack,		

sum(attack_count) as attack_count

```
from
```

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

Dataset Name	Description	Log Category
utm-drilldown-Top-Vulnerability	UTM drilldown top vulnerability by name	netscan
select vuln, sum(totalnum) as totalnum		
<pre>from ###(select coalesce(nullifna</pre>	(`user`), ipstr(`srcip`)) as user_src, vul	ln, count(*) as

totalnum from \$log where \$filter and action='vuln-detection' and vuln is not null group by user_src, vuln order by totalnum desc)### t where \$filter-drilldown group by vuln order by totalnum desc

Dataset Name	Description	Log Category	
utm-drilldown-Top-App-By-Bandwidth	UTM drilldown top applications by bandwidth usage	traffic	
select appid, app, sum(bandwidth) as bandwidth			
<pre>from ###(select user_src, appid, app, appcat, apprisk, sum(bandwidth) as bandwidth, sum (sessions) as sessions from ###base(/*tag:rpt_base_t_top_app*/select dvid, srcip, dstip,</pre>			

epid, euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, count(*) as sessions from \$log-traffic where \$filter and (logflag&(1|32)>0) and nullifna(app) is not null group by dvid, srcip, dstip, epid, euid, user_src, appid, app, appcat, apprisk, hostname order by sessions desc)base### t group by user_src, appid, app, appcat, apprisk /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t where \$filter-drilldown group by appid, app having sum (bandwidth)>0 order by bandwidth desc

Dataset Name	Description	Log Category
utm-drilldown-Top-App-By-Sessions	UTM drilldown top applications by session count	traffic

```
select
```

```
appid,
app,
sum(sessions) as sessions
from
```

###(select user_src, appid, app, appcat, apprisk, sum(bandwidth) as bandwidth, sum
(sessions) as sessions from ###base(/*tag:rpt_base_t_top_app*/select dvid, srcip, dstip,
epid, euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src,
appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in,
sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte,
0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, count(*) as sessions from \$log-traffic
where \$filter and (logflag&(1|32)>0) and nullifna(app) is not null group by dvid, srcip,
dstip, epid, euid, user_src, appid, app, appcat, apprisk, hostname order by sessions
desc)base### t group by user_src, appid, app, appcat, apprisk /*SkipSTART*/order by sessions
desc, bandwidth desc/*SkipEND*/)### t where \$filter-drilldown group by appid, app order by
sessions desc

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

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

Dataset Name	Description	Log Category
bandwidth-app-Top-App-By- Bandwidth-Sessions	Top applications by bandwidth usage	traffic

select

app_group_name(app) as app_group, sum(bandwidth) as bandwidth, sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_out, sum(sessions) as sessions

from

###(select appid, app, appcat, apprisk, sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_out, sum(bandwidth) as bandwidth, sum(sessions) as sessions from ###base(/*tag:rpt_ base_t_top_app*/select dvid, srcip, dstip, epid, euid, coalesce(nullifna(`user`), nullifna (`unauthuser`), ipstr(`srcip`)) as user_src, appid, app, appcat, apprisk, hostname, sum (coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, count(*) as sessions from \$log-traffic where \$filter and (logflag&(1|32)>0) and nullifna(app) is not null group by dvid, srcip, dstip, epid, euid, user_src, appid, app, appcat, apprisk, hostname order by sessions desc)base### t group by appid, app, appcat, apprisk /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t group by app_ group having sum(bandwidth)>0 order by bandwidth desc

Dataset Name	Description	Log Category
bandwidth-app-Category-By- Bandwidth	Application risk application usage by category	traffic

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

from

###(select appid, app, appcat, apprisk, sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_out, sum(bandwidth) as bandwidth, sum(sessions) as sessions from ###base(/*tag:rpt_ base_t_top_app*/select dvid, srcip, dstip, epid, euid, coalesce(nullifna(`user`), nullifna (`unauthuser`), ipstr(`srcip`)) as user_src, appid, app, appcat, apprisk, hostname, sum (coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, count(*) as sessions from \$log-traffic where \$filter and (logflag&(1|32)>0) and nullifna(app) is not null group by dvid, srcip, dstip, epid, euid, user_src, appid, app, appcat, apprisk, hostname order by sessions desc)base### t group by appid, app, appcat, apprisk /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t where \$filterdrilldown and nullifna(appcat) is not null group by appcat having sum(bandwidth)>0 order by bandwidth desc

Dataset Name	Description	Log Category
bandwidth-app-Top-Users-By- Bandwidth-Sessions	Bandwidth application top users by bandwidth usage	traffic
(bandwidth) as bandwidth, sum app*/select dvid, srcip, dsti (`unauthuser`), ipstr(`srcip` (coalesce(rcvddelta, rcvdbyte traffic_out, sum(coalesce(sen bandwidth, count(*) as sessio nullifna(app) is not null gro appcat, apprisk, hostname ord	in,	ase_t_top_ fna stname, sum htbyte, 0)) as te, 0)) as &(1 32)>0) and appid, app, order by
Dataset Name	Description	Log Category
bandwidth-app-Traffic-By-Active-Use Number	er- Bandwidth application traffic by active user number	traffic

select
 \$flex_timescale(timestamp) as hodex,
 count(
 distinct(user_src)
) as total_user

from

###(select \$flex_timestamp as timestamp, coalesce(nullifna(`user`), nullifna
(`unauthuser`), ipstr(`srcip`)) as user_src from \$log where \$filter and (logflag&(1|32)>0)
group by timestamp, user_src order by timestamp desc)### t group by hodex order by hodex

Dataset Name	Description	Log Category
bandwidth-app-Top-Dest-By- Bandwidth-Sessions	Bandwidth application top dest by bandwidth usage sessions	traffic
select		
coalesce(
nullifna(
root_domain(hostname)		
),		
ipstr(`dstip`)		
) as domain,		
<pre>sum(traffic_in) as traffic_i</pre>	n,	
<pre>sum(traffic_out) as traffic_</pre>	out,	
sum(bandwidth) as bandwidth,		
sum(sessions) as sessions		

from

###(select hostname, dstip, sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_ out, sum(bandwidth) as bandwidth, sum(sessions) as sessions from ###base(/*tag:rpt_base_t_ top_app*/select dvid, srcip, dstip, epid, euid, coalesce(nullifna(`user`), nullifna (`unauthuser`), ipstr(`srcip`)) as user_src, appid, app, appcat, apprisk, hostname, sum (coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, count(*) as sessions from \$log-traffic where \$filter and (logflag&(1|32)>0) and nullifna(app) is not null group by dvid, srcip, dstip, epid, euid, user_src, appid, app, appcat, apprisk, hostname order by sessions desc)base### t group by hostname, dstip order by sessions desc, bandwidth desc)### t group by domain order by bandwidth desc

Dataset Name	Description	Log Category
bandwidth-app-Top-Policies-By- Bandwidth-Sessions	Top policies by bandwidth and sessions	traffic
select		
coalesce(
pol.name,		
cast(policyid as text)		
) as polid,		
sum(bandwidth) as bandwidth,		
<pre>sum(traffic_in) as traffic_in,</pre>		
<pre>sum(traffic_out) as traffic_ou</pre>	it,	
sum(sessions) as sessions		
from		
###(select policyid, poluuid,	<pre>sum(coalesce(rcvdbyte, 0) + coalesce(sentbyte,</pre>	0)) as

bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_ out, count(*) as sessions from \$log where \$filter and (logflag&1>0) group by policyid, poluuid order by bandwidth desc)### t1 left join \$ADOMTBL_PLHD_POLINFO pol on t1.poluuid=pol.uuid group by polid order by bandwidth desc

Dataset Name	Description	Log Category
bandwidth-app-Traffic-Statistics	Bandwidth application traffic statistics	traffic
drop	create temporary table rpt_tmptbl_1(traffic
<pre>select format_numeric_no_decimal(sum(sessions)) as total_sessions, bandwidth_unit(</pre>		

```
sum(bandwidth)
) as total_bandwidth,
format_numeric_no_decimal(
   cast(
     sum(sessions) / $days_num as decimal(18, 0)
   )
) as ave_session,
bandwidth_unit(
   cast(
     sum(bandwidth) / $days_num as decimal(18, 0)
   )
) as ave_bandwidth
```

```
from
```

###(select appid, app, appcat, apprisk, sum(traffic in) as traffic in, sum(traffic out) as traffic out, sum(bandwidth) as bandwidth, sum(sessions) as sessions from ###base(/*tag:rpt base t top app*/select dvid, srcip, dstip, epid, euid, coalesce(nullifna(`user`), nullifna (`unauthuser`), ipstr(`srcip`)) as user src, appid, app, appcat, apprisk, hostname, sum (coalesce(rcvddelta, rcvdbyte, 0)) as traffic in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, count(*) as sessions from \$log-traffic where \$filter and (logflag&(1|32)>0) and nullifna(app) is not null group by dvid, srcip, dstip, epid, euid, user_src, appid, app, appcat, apprisk, hostname order by sessions desc)base### t group by appid, app, appcat, apprisk /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t; update rpt tmptbl 1 set active date=t1.dom from (select dom, sum(sessions) as sessions from ###(select \$DAY OF MONTH as dom, count(*) as sessions from \$log where \$filter and (logflag&(1|32)>0) group by dom order by sessions desc)### t group by dom order by sessions desc limit 1) as tl; update rpt tmptbl 1 set total users=t2.totalnum from (select format numeric no decimal (count(distinct(user src))) as totalnum from ###(select user_src, sum(sessions) as count from ###base(/*tag:rpt base t top app*/select dvid, srcip, dstip, epid, euid, coalesce (nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user src, appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic in, sum(coalesce (sentdelta, sentbyte, 0)) as traffic out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce (rcvddelta, rcvdbyte, 0)) as bandwidth, count(*) as sessions from \$log-traffic where \$filter and (logflag&(1|32)>0) and nullifna(app) is not null group by dvid, srcip, dstip, epid, euid, user src, appid, app, appcat, apprisk, hostname order by sessions desc)base### t group by user src order by count desc)### t) as t2; update rpt tmptbl 1 set total app=t3.totalnum from (select format numeric no decimal(count(distinct(app grp))) as totalnum from ###(select app group name(app) as app grp, sum(sessions) as count from ###base(/*tag:rpt base t top app*/select dvid, srcip, dstip, epid, euid, coalesce(nullifna(`user`), nullifna (`unauthuser`), ipstr(`srcip`)) as user_src, appid, app, appcat, apprisk, hostname, sum (coalesce(rcvddelta, rcvdbyte, 0)) as traffic in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, count(*) as sessions from \$log-traffic where \$filter and (logflag&(1|32)>0) and nullifna(app) is not null group by dvid, srcip, dstip, epid, euid, user_src, appid, app, appcat, apprisk, hostname order by sessions desc)base### t group by app grp order by count desc)### t) as t3; update rpt tmptbl 1 set total dest=t4.totalnum from (select format numeric no decimal(count(distinct(dstip))) as totalnum from ###(select dstip, sum(sessions) as count from ###base(/*tag:rpt_base_t_top_app*/select dvid, srcip, dstip, epid, euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic in, sum(coalesce (sentdelta, sentbyte, 0)) as traffic out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce (rcvddelta, rcvdbyte, 0)) as bandwidth, count(*) as sessions from \$log-traffic where \$filter and (logflag&(1|32)>0) and nullifna(app) is not null group by dvid, srcip, dstip, epid, euid, user src, appid, app, appcat, apprisk, hostname order by sessions desc)base### t where dstip is not null group by dstip order by count desc)### t) as t4; select 'Total Sessions' as summary, total sessions as stats from rpt tmptbl 1 union all select 'Total Bytes

Transferred' as summary, total_bandwidth as stats from rpt_tmptbl_1 union all select 'Most Active Date By Sessions' as summary, active_date as stats from rpt_tmptbl_1 union all select 'Total Users' as summary, total_users as stats from rpt_tmptbl_1 union all select 'Total Applications' as summary, total_app as stats from rpt_tmptbl_1 union all select 'Total Destinations' as summary, total_dest as stats from rpt_tmptbl_1 union all select 'Average Sessions Per Day' as summary, ave_session as stats from rpt_tmptbl_1 union all select 'Average Bytes Per Day' as summary, ave_bandwidth as stats from rpt_tmptbl_1

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

\$flex timescale(timestamp) as hodex,

sum(scores) as scores

from

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

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

select

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

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

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

```
group by
  user_src
having
  sum(crscore % 65536)> 0
order by
  scores desc
```

Dataset Name	Description	Log Category
Top-Devices-By-Reputation-Scores	Reputation top devices by scores	traffic
<pre>Top-Devices-By-Reputation-Scores select max(get_devtype(srcswversion, of) as devtype_new, coalesce(nullifna(`srcname`), nullifna(`srcmac`), ipstr(`srcip`)) as dev_src, sum(crscore % 65536) as score from \$log where \$filter and (logflag&1>0) and crscore is not null group by</pre>	sname, devtype)	traffic
dev_src		
<pre>having sum(crscore % 65536)> 0</pre>		
order by scores desc		

Dataset Name	Description	Log Category
Top-Users-With-Increased-Scores	Reputation top users with increased scores	traffic
<pre>select f_user, sum(sum_rp_score) as sum_rp_sc from ###(select coalesce(nullifna(` sum(crscore%65536) as sum_rp_sco crscore is not null group by f_u desc)### t group by f_user; crea rp_score) as sum_rp_score from # ipstr(`srcip`)) as f_user, sum(c (logflag&1>0) and crscore is not</pre>	<pre>create temporary table rpt_tmptbl_1 as ore user`), nullifna(`unauthuser`), ipstr(`srcip re from \$log where \$pre_period \$filter and ser having sum(crscore%65536)>0 order by sur te temporary table rpt_tmptbl_2 as select f_ ##(select coalesce(nullifna(`user`), nullifn rscore%65536) as sum_rp_score from \$log when null group by f_user having sum(crscore%655 f_user; select t1.f_user, sum(t1.sum_rp_score)</pre>	(logflag&1>0) and m_rp_score _user, sum(sum_ na(`unauthuser`), re \$filter and 536)>0 order by

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		
<pre>table if exists rpt_tmptbl_1;</pre>		
drop		
<pre>table if exists rpt_tmptbl_2;</pre>	create temporary table rpt_tmptbl_1 as	
select		
f_device,		
devtype_new,		
<pre>sum(sum_rp_score) as sum_rp_sc</pre>	core	
from		
###(select coalesce(nullifna(`	<pre>srcname`),nullifna(`srcmac`), ipstr(`srcip`))</pre>	as f_device,
<pre>get_devtype(srcswversion, osname</pre>	e, devtype) as devtype_new, sum(crscore%65536)	as sum_rp_

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

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

```
select
```

(

case when severity =& #039;critical' then 'Critical' when severity='high' then 'High' when severity='medium' then 'Medium' when severity='low' then 'Low' when severity='info' then 'Info' end) as severity, count(*) as totalnum from \$log where \$filter group by severity order by totalnum desc

Dataset Name	Description	Log Category
Top-Attacks-Detected	Threat top attacks detected	attack
<pre>select attack, attackid, cve, severity, sum(attack_count) as attack_count)</pre>	count	
	<pre>t1.severity, cve, (case when t1.sev 2 when t1.severity = 'medium' then</pre>	_

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

Dataset Name	Description	Log Category
Top-Attacks-Blocked	Threat top attacks blocked	attack
<pre>select attack, count(*) as attack_count from \$log where \$filter and nullifna(attack) is not n and action not in (</pre>	ull sion') group by attack order by	attack_count desc

Dataset Name	Description	Log Category
Top-Virus-Source	Threat top virus source	virus
select		

source, hostname, sum(totalnum) as totalnum from

###(select source, ipstr(`victim`) as hostname, sum(totalnum) as totalnum from (select (CASE WHEN direction='incoming' THEN dstip ELSE srcip END) as source, (CASE WHEN direction='incoming' THEN srcip ELSE dstip END) as victim, count(*) as totalnum from \$log where \$filter and (eventtype is null or logver>=502000000) and nullifna(virus) is not null group by source, victim) t group by source, hostname /*SkipSTART*/order by totalnum desc/*SkipEND*/)### t group by source, hostname order by totalnum desc

Dataset Name	Description	Log Category
Intrusion-in-Last-7-Days	Threat intrusion timeline	attack

select

\$flex_timescale(timestamp) as hodex, sum(totalnum) as totalnum from ###(select \$flex timestamp as timest

```
###(select $flex_timestamp as timestamp, count(*) as totalnum from $log where $filter
group by timestamp /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t group by hodex
order by hodex
```

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

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

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

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

user src, sum(totalnum) as totalnum

from

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

Dataset Name	Description	Log Category	
Top-Spyware-by-Name	Threat top spyware by name	virus	
<pre>select virus, max(virusid_s) as virusid, sum(totalnum) as totalnum</pre>			
<pre>from ###(select filename, analyticscksum, service, fsaverdict, dtype, coalesce(nullifna</pre>			
(`user`), ipstr(`srcip`)) as user_src, virus, virusid_to_str(virusid, eventtype) as virusid_			

s, count(*) as totalnum from \$log where \$filter group by filename, analyticscksum, service, fsaverdict, dtype, user src, virus, virusid s /*SkipSTART*/order by totalnum

desc/*SkipEND*/)### t where virus like 'Riskware%' group by virus order by totalnum desc

Dataset Name	Description	Log Category	
Top-Spyware-Source	Threat top spyware source	traffic	
<pre>select srcip, hostname, sum(totalnum) as totalnum from ####(select srcip, hostname, virus, count(*) as totalnum from \$log where \$filter and (logflag&1>0) group by srcip, hostname, virus order by totalnum desc)### t where virus lik 'Riskware%' group by srcip, hostname order by totalnum desc</pre>			
Dataset Name Description Log		Log Category	
Spyware-Time-Line	Threat spyware timeline	virus	
<pre>select \$flex_timescale(timestamp) as hodex, sum(totalnum) as totalnum from ###(select \$flex_timestamp as timestamp, virus, count(*) as totalnum from \$log where \$filter group by timestamp, virus /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t</pre>			

where virus like 'Riskware%' group by hodex order by hodex

Dataset Name	Description	Log Category
Top-Adware-Victims	Threat top adware victims	virus
select		
user src,		

sum(totalnum) as totalnum

from

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

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

select

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

from

###(select filename, analyticscksum, service, fsaverdict, dtype, coalesce(nullifna
(`user`), ipstr(`srcip`)) as user_src, virus, virusid_to_str(virusid, eventtype) as virusid_
s, count(*) as totalnum from \$log where \$filter group by filename, analyticscksum, service,
fsaverdict, dtype, user_src, virus, virusid_s /*SkipSTART*/order by totalnum
desc/*SkipEND*/)### t where virus like 'Adware%' group by virus order by totalnum desc

Dataset Name	Description	Log Category
Top-Adware-Source	Threat top adware source	traffic
<pre>select srcip, hostname,</pre>		
<pre>sum(totalnum) as totalnum from</pre>	virus count(*) as totalnum from \$lo	

###(select srcip, hostname, virus, count(*) as totalnum from \$log where \$filter and (logflag&1>0) group by srcip, hostname, virus order by totalnum desc)### t where virus like 'Adware%' group by srcip, hostname order by totalnum desc

Dataset Name	Description	Log Category
Adware-Time-Line	Threat adware timeline	virus
\$filter group by timesta		-
Dataset Name	Description	

Dataset Name	Description	Log Category
Intrusions-Timeline-By-Severity	Threat intrusions timeline by severity	attack

```
select
  $flex_timescale(timestamp) as timescale,
  sum(critical) as critical,
  sum(high) as high,
  sum(medium) as medium,
  sum(low) as low,
  sum(info) as low,
  from
  ###(select $flex_timestamp as timestamp, sum(case when severity = 'critical' then 1 else 0
end) as critical, sum(case when severity = 'high' then 1 else 0 end) as high, sum(case when
```

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

Dataset Name	Description	Log Category
Important-Intrusions-Timeline-By- Severity	Threat intrusions timeline by severity	attack
<pre>select \$flex_timescale(timestamp) a. sum(critical) as critical, sum(high) as high, sum(medium) as medium, sum(low) as low, sum(info) as info</pre>	s timescale,	
<pre>from ###(select \$flex timestamp as</pre>	s timestamp, sum(case when severity = 'c	ritical' then 1 else 0
end) as critical, sum(case when severity = 'medium' then 1 else	n severity = 'high' then 1 else 0 end) a e 0 end) as medium, sum(case when severi ow, sum(case when severity = 'info' or se	s high, sum(case when ty in ('notice',

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

Dataset Name	Description	Log Category
Top-Intrusions-By-Types	Threat top intrusions by types	attack
<pre>Top-Intrusions-By-Types select vuln_type, count(*) as totalnum from \$log t1 left join (select name, cve, vuln_type from ips_mdata) t2 on t1.attack = t2.name where \$filter and vuln_type is not null</pre>	Threat top intrusions by types	attack
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</pre>			
<pre>\$log t1 left join (select name, cve, vuln_type from ips_mdata) t2 on t1.attack = t2.name</pre>			
<pre>where \$filter and t1.severity = & #039;criti attackid, cve, vuln_type order b</pre>		s not null group by	attack,

Dataset Name	Description	Log Category
High-Severity-Intrusions	Threat high severity intrusions	attack
select		
attack,		
attackid,		
vuln_type,		
cve,		
count(*) as totalnum		
from		
\$log t1		
left join (
select		
name,		
cve,		
vuln_type		
from		
ips_mdata		
) t2 on t1.attack = t2.name		
where		
\$filter		
and t1.severity =& #039;high'	and nullifna(attack) is not null group by at	tack, attackid,
<pre>vuln_type, cve order by totalnum</pre>	desc	

Dataset Name	Description	Log Category
Medium-Severity-Intrusions	Threat medium severity intrusions	attack

```
select
 attack,
 vuln_type,
 cve,
  count(*) as totalnum
from
  $log t1
 left join (
   select
     name,
     cve,
     vuln_type
   from
     ips_mdata
 ) t2 on t1.attack = t2.name
where
  $filter
```

and t1.severity =& #039; medium' and nullifna(attack) is not null group by attack, vuln_type, cve order by totalnum desc

Top-Intrusion-Victims Threat top intrusion victims select victim,	Log Category
	attack
<pre>sum(cri_num) as critical, sum(high_num) as high, sum(med_num) as medium, sum(cri_num + high_num + med_num) as totalnum</pre>	
<pre>from ###(select (CASE WHEN direction='incoming' THEN srcip ELSE dstip END) as v</pre>	victim, sum((case

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

Dataset Name	Description	Log Category
Top-Intrusion-Sources	Threat top intrusion sources	attack
<pre>select source, sum(cri_num) as critical, sum(high_num) as high, sum(med_num) as medium, sum(cri_num + high_num + med_</pre>	_num) as totalnum	
<pre>from ###(select (CASE WHEN direction='incoming' THEN dstip ELSE srcip END) as source, sum(case when severity='critical' then 1 else 0 end) as cri_num, sum(case when severity='high' then else 0 end) as high_num, sum(case when severity='medium' then 1 else 0 end) as med_num from \$log where \$filter and severity in ('critical', 'high', 'medium') group by source)### t group by source order by totalnum desc</pre>		

Dataset Name	Description	Log Category
Top-Blocked-Intrusions	Threat top blocked intrusions	attack

select attack,

```
attackid,
 (
    case when severity =& #039;critical' then 'Critical' when severity='high' then 'High'
when severity='medium' then 'Medium' when severity='low' then 'Low' when severity='info'
then 'Info' end) as severity_name, sum(totalnum) as totalnum, vuln_type, (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 ###
(select attack, attackid, t1.severity, count(*) as totalnum, vuln_type, action from $log t1
left join (select name, cve, vuln_type from ips_mdata) t2 on t1.attack=t2.name where $filter
and nullifna(attack) is not null group by attack, attackid, t1.severity, vuln_type, action
order by totalnum desc)### t where action not in ('detected', 'pass_session') group by
attack, attackid, severity, vuln_type order by severity_number, totalnum desc
```

Dataset Name	Description	Log Category
Top-Monitored-Intrusions	Threat top monitored intrusions	attack
	039;critical' then 'Critical' when sev	
when severity='medium' then	'Medium' when severity='low' then 'Lo	ow' when severity='info'

when severity='medium' then 'Medium' when severity='low' then 'Low' when severity='info' then 'Info' end) as severity_name, sum(totalnum) as totalnum, vuln_type, (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 ### (select attack, attackid, t1.severity, count(*) as totalnum, vuln_type, action from \$log t1 left join (select name, cve, vuln_type from ips_mdata) t2 on t1.attack=t2.name where \$filter and nullifna(attack) is not null group by attack, attackid, t1.severity, vuln_type, action order by totalnum desc)### t where action in ('detected', 'pass_session') group by attack, attackid, severity, vuln_type order by severity_number, totalnum desc

Dataset Name	Description	Log Category
Attacks-Over-HTTP-HTTPs	Threat attacks over HTTP HTTPs	attack
when severity='medium' then 'Med then 'Info' end) as severity, co when severity='high' then 1 when severity='info' then 4 else 5 er in ('critical', 'high', 'medium'	critical' then 'Critical' when seven dium' when severity='low' then 'Low' bunt(*) as totalnum, (case when seven a severity='medium' then 2 when seven ad) as severity_number from \$log when) and upper(service) in ('HTTP', 'H aber order by severity_number, total	when severity='info' erity='critical' then 0 erity='low' then 3 when ere \$filter and severity [TTPS') group by attack,

Dataset Name	Description	Log Category
default-AP-Detection-Summary-by- Status-OffWire	Default access point detection summary by status off- wire	event

```
select
```

(

case apstatus when 1 then & #039;rogue' when 2 then 'accepted' when 3 then 'suppressed' else 'others' end) as ap_full_status, count(*) as totalnum from (select apstatus, bssid, ssid from ###(select apstatus, bssid, ssid, onwire, count(*) as subtotal from \$log where \$filter and apstatus is not null and apstatus!=0 and bssid is not null 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, onwire order by subtotal desc)### t where onwire='no' group by apstatus, bssid, ssid) t group by ap_full_status order by totalnum desc

Dataset Name	Description	Log Category
default-AP-Detection-Summary-by-	Default access point detection summary by status off-	event
Status-OffWire_table	wire	

select

(

case apstatus when 1 then & #039;rogue' when 2 then 'accepted' when 3 then 'suppressed' else 'others' end) as ap_full_status, count(*) as totalnum from (select apstatus, bssid, ssid from ###(select apstatus, bssid, ssid, onwire, count(*) as subtotal from \$log where \$filter and apstatus is not null and apstatus!=0 and bssid is not null 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, onwire order by subtotal desc)### t where onwire='no' group by apstatus, bssid, ssid) t group by ap_full_status order by totalnum desc

Dataset Name	Description	Log Category
default-AP-Detection-Summary-by- Status-OnWire	Default access point detection summary by status on- wire	event

select

(

case apstatus when 1 then & #039;rogue' when 2 then 'accepted' when 3 then 'suppressed' else 'others' end) as ap_full_status, count(*) as totalnum from (select apstatus, bssid, ssid from ###(select apstatus, bssid, ssid, onwire, count(*) as subtotal from \$log where \$filter and apstatus is not null and apstatus!=0 and bssid is not null 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, onwire order by subtotal desc)### t where onwire='yes' group by apstatus, bssid, ssid) t group by ap_full_status order by totalnum desc

Dataset Name	Description	Log Category
default-AP-Detection-Summary-by- Status-OnWire table	Default access point detection summary by status on- wire	event

select

(

case apstatus when 1 then & #039;rogue' when 2 then 'accepted' when 3 then 'suppressed' else 'others' end) as ap_full_status, count(*) as totalnum from (select apstatus, bssid, ssid from ###(select apstatus, bssid, ssid, onwire, count(*) as subtotal from \$log where \$filter and apstatus is not null and apstatus!=0 and bssid is not null 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, onwire order by subtotal desc)### t where onwire='yes' group by apstatus, bssid, ssid) t group by ap_full_status order by totalnum desc

Dataset Name	Description	Log Category
default-Managed-AP-Summary	Default managed access point summary	event
select (case when (
'Authorized' else 'Unauthorized'	and logid_to_int(logid) in (43522, 43551)) t end) as ap_status, count(*) as totalnum from in (43522, 43551) group by ap_status order by	\$log where

Dataset Name	Description	Log Category
default-Managed-AP-Summary_table	Default managed access point summary	event

select

```
(
case when (
```

action like & #039;%join%' and logid_to_int(logid) in (43522, 43551)) then 'Authorized' else 'Unauthorized' end) as ap_status, count(*) as totalnum from \$log where \$filter and logid_to_int(logid) in (43522, 43551) group by ap_status order by totalnum desc

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

select

(

case onwire when & #039;no' then 'off-wire' when 'yes' then 'on-wire' else 'others' end) as ap_status, count(*) as totalnum from ###(select onwire, ssid, bssid, count(*) as subtotal from \$log where \$filter and apstatus=0 and bssid is not null and logid_to_int(logid) in (43521, 43525, 43527, 43563, 43564, 43565, 43566, 43569, 43570, 43571, 43582, 43583, 43584, 43585) group by onwire, ssid, bssid order by subtotal desc)### t group by ap_status order by totalnum desc

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

select

(

case onwire when & #039;no' then 'off-wire' when 'yes' then 'on-wire' else 'others' end) as ap_status, count(*) as totalnum from ###(select onwire, ssid, bssid, count(*) as subtotal from \$log where \$filter and apstatus=0 and bssid is not null and logid_to_int(logid) in (43521, 43525, 43527, 43563, 43564, 43565, 43566, 43569, 43570, 43571, 43582, 43583, 43584, 43585) group by onwire, ssid, bssid order by subtotal desc)### t group by ap_status order by totalnum desc

Dataset Name	Description	Log Category
default-selected-AP-Details-OffWire	Default selected access point details off-wire	event

select

(

case apstatus when 0 then & #039;unclassified' when 1 then 'rogue' when 2 then 'accepted' when 3 then 'suppressed' else 'others' end) as ap_full_status, devid, vd, ssid, bssid, manuf, rssi, channel, radioband, from_dtime(min(first_seen)) as first_seen, from_ dtime(max(last_seen)) as last_seen, detectionmethod, itime, onwire as on_wire from ### (select apstatus, devid, vd, ssid, bssid, manuf, rssi, channel, radioband, min(dtime) as first_seen, max(dtime) as last_seen, detectionmethod, itime, onwire from \$log where \$filter and apstatus is not null and bssid is not null and logid_to_int(logid) in (43521, 43563, 43564, 43565, 43566, 43569, 43570, 43571) group by apstatus, devid, vd, ssid, bssid, manuf, rssi, channel, radioband, detectionmethod, itime, onwire order by itime desc)### t where onwire='no' group by ap_full_status, devid, vd, ssid, bssid, manuf, rssi, channel, radioband, detectionmethod, itime, onwire by itime desc

Dataset Name	Description	Log Category
default-selected-AP-Details-OnWire	Default selected access point details on-wire	event

select

(

case apstatus when 0 then & #039;unclassified' when 1 then 'rogue' when 2 then 'accepted' when 3 then 'suppressed' else 'others' end) as ap_full_status, devid, vd, ssid, bssid, manuf, rssi, channel, radioband, from_dtime(min(first_seen)) as first_seen, from_ dtime(max(last_seen)) as last_seen, detectionmethod, itime, onwire as on_wire from ### (select apstatus, devid, vd, ssid, bssid, manuf, rssi, channel, radioband, min(dtime) as first_seen, max(dtime) as last_seen, detectionmethod, itime, onwire from \$log where \$filter and apstatus is not null and bssid is not null and logid_to_int(logid) in (43521, 43563, 43564, 43565, 43566, 43569, 43570, 43571) group by apstatus, devid, vd, ssid, bssid, manuf, rssi, channel, radioband, detectionmethod, itime, onwire order by itime desc)### t where onwire='yes' group by ap_full_status, devid, vd, ssid, bssid, manuf, rssi, channel, radioband, detectionmethod, itime, onwire by itime desc

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

0)+coalesce(rcvdbyte, 0)) as volume from \$log-traffic where \$filter-time and (logflag&1>0) and srcip is not null group by user_src, srcip having sum(coalesce(sentbyte, 0)+coalesce (rcvdbyte, 0))>0 order by volume desc)### t group by user_src, srcip order by user_src, srcip) t on rpt_tmptbl_1.ip = t.srcip) t order by volume desc

Dataset Name	Description	Log Category
event-Wireless-Accepted-Offwire	Event wireless accepted off-wire	event

select

& #039;accepted' as ap_full_status, devid, vd, ssid, bssid, manuf, channel, radioband, from_dtime(max(last_seen)) as last_seen, detectionmethod, snclosest, 'no' as on_wire from ###(select devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest, onwire, logid, apstatus, max(dtime) as last_seen from \$log where \$filter and bssid is not null and logid_to_int(logid) in (43521, 43525, 43563, 43564, 43565, 43566, 43569, 43570, 43571) group by devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest, onwire, logid, apstatus order by last_seen desc)### t where apstatus=2 and onwire='no' group by devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest order by last_seen desc

Dataset Name	Description	Log Category
event-Wireless-Accepted-Onwire	Event wireless accepted on-wire	event

select

& #039;accepted' as ap_full_status, devid, vd, ssid, bssid, manuf, channel, radioband, from_dtime(max(last_seen)) as last_seen, detectionmethod, snclosest, 'yes' as on_wire from ###(select devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest, onwire, apstatus, max(dtime) as last_seen from \$log where \$filter and bssid is not null and logid_to_int(logid) in (43521, 43525, 43563, 43564, 43565, 43566, 43569, 43570, 43571) group by devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest, onwire, apstatus order by last_seen desc)### t where apstatus=2 and onwire='yes' group by devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest order by last seen desc

Dataset Name	Description	Log Category
event-Wireless-Rogue-Offwire	Event wireless rogue off-wire	event

select

& #039;rogue' as ap_full_status, devid, vd, ssid, bssid, manuf, channel, radioband, from_ dtime(max(last_seen)) as last_seen, detectionmethod, snclosest, 'no' as on_wire from ### (select devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest, onwire, logid, apstatus, max(dtime) as last_seen from \$log where \$filter and bssid is not null and logid_to_int(logid) in (43521, 43525, 43563, 43564, 43565, 43566, 43569, 43570, 43571) group by devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest, onwire, logid, apstatus order by last_seen desc)### t where apstatus=1 and onwire='no' group by devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest order by last seen desc

Dataset Name	Description	Log Category
event-Wireless-Rogue-Onwire	Event wireless rogue on-wire	event

select

& #039;rogue' as ap_full_status, devid, vd, ssid, bssid, manuf, channel, radioband, from_ dtime(max(last_seen)) as last_seen, detectionmethod, snclosest, 'yes' as on_wire from ### (select devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest, onwire, apstatus, max(dtime) as last_seen from \$log where \$filter and bssid is not null and logid_to_int(logid) in (43521, 43525, 43563, 43564, 43565, 43566, 43569, 43570, 43571) group by devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest, onwire, apstatus order by last_seen desc)### t where apstatus=1 and onwire='yes' group by devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest order by last_seen desc

Dataset Name	Description	Log Category
event-Wireless-Suppressed-Offwire	Event wireless suppressed off-wire	event

select

& #039;suppressed' as ap_full_status, devid, vd, ssid, bssid, manuf, channel, radioband, from_dtime(max(last_seen)) as last_seen, detectionmethod, snclosest, 'no' as on_wire from ###(select devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest, onwire, logid, apstatus, max(dtime) as last_seen from \$log where \$filter and bssid is not null and logid_to_int(logid) in (43521, 43525, 43563, 43564, 43565, 43566, 43569, 43570, 43571) group by devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest, onwire, logid, apstatus order by last_seen desc)### t where apstatus=3 and onwire='no' group by devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest order by last_seen desc

Dataset Name	Description	Log Category
event-Wireless-Suppressed-Onwire	Event wireless suppressed on-wire	event

select

& #039;suppressed' as ap_full_status, devid, vd, ssid, bssid, manuf, channel, radioband, from_dtime(max(last_seen)) as last_seen, detectionmethod, snclosest, 'yes' as on_wire from ###(select devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest, onwire, apstatus, max(dtime) as last_seen from \$log where \$filter and bssid is not null and logid_to_int(logid) in (43521, 43525, 43563, 43564, 43565, 43566, 43569, 43570, 43571) group by devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest, onwire, apstatus order by last_seen desc)### t where apstatus=3 and onwire='yes' group by devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest order by last_seen desc

Dataset Name	Description	Log Category
event-Wireless-Unclassified-Offwire	Event wireless unclassified off-wire	event

select

& #039;unclassified' as ap_full_status, devid, vd, ssid, bssid, manuf, channel, radioband, from_dtime(max(last_seen)) as last_seen, detectionmethod, snclosest, 'no' as on_wire from ###(select devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest, onwire, logid, apstatus, max(dtime) as last_seen from \$log where \$filter and bssid is not null and logid_to_int(logid) in (43521, 43525, 43563, 43564, 43565, 43566, 43569, 43570, 43571) group by devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest, onwire, logid, apstatus order by last_seen desc)### t where apstatus=0 and onwire='no' group by devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest order by last_seen desc

Dataset Name	Description	Log Category
event-Wireless-Unclassified-Onwire	Event wireless unclassified on-wire	event

select

& #039;unclassified' as ap_full_status, devid, vd, ssid, bssid, manuf, channel, radioband, from_dtime(max(last_seen)) as last_seen, detectionmethod, snclosest, 'yes' as on_wire from ####(select devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest, onwire, apstatus, max(dtime) as last_seen from \$log where \$filter and bssid is not null and logid_to_int(logid) in (43521, 43525, 43563, 43564, 43565, 43566, 43569, 43570, 43571) group by devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest, onwire, apstatus order by last_seen desc)### t where apstatus=0 and onwire='yes' group by devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest order by last_seen desc

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

```
select
 coalesce(
   xauthuser agg,
   user agg,
   ipstr(`remip`)
  ) as user src,
  from dtime(
   min(s time)
  ) as start time,
  sum(bandwidth) as bandwidth,
  sum(traffic in) as traffic in,
  sum(traffic out) as traffic out
from
  (
   select
     devid,
     vd,
     string agg(
```

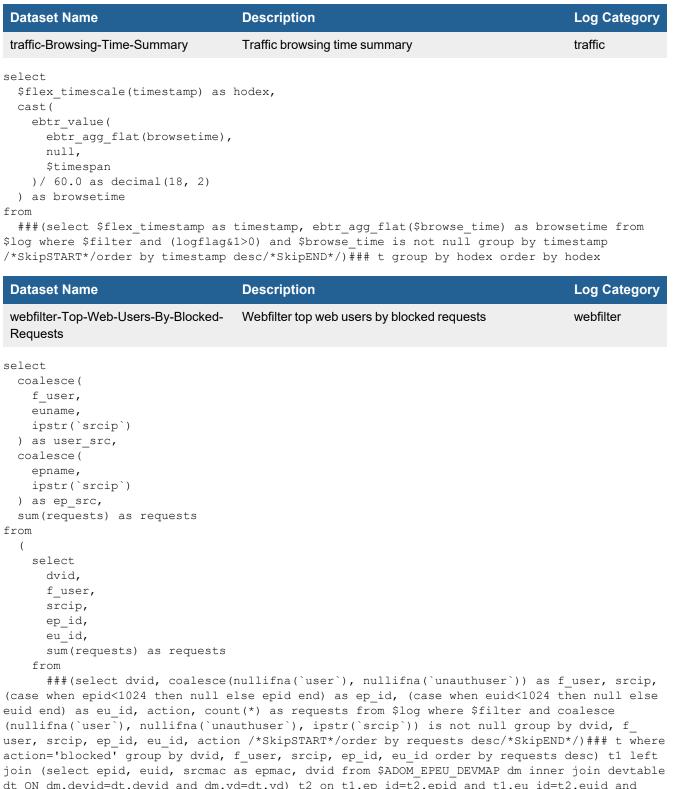
```
distinct xauthuser_agg,
```

& #039; ') as xauthuser_agg, string_agg(distinct user_agg, ' ') as user_agg, remip, tunnelid, min(s_time) as s_time, max(e_time) as e_time, (case when min(s_time)=max(e_time) then max(max traffic in)+max(max traffic out) else max(max traffic in)-min(min traffic in)+max(max traffic out)-min(min traffic out) end) as bandwidth, (case when min(s time)=max (e time) then max(max traffic in) else max(max traffic in)-min(min traffic in) end) as traffic in, (case when min(s time)=max(e time) then max(max traffic out) else max(max traffic out)-min(min traffic out) end) as traffic out from ###(select devid, vd, remip, nullifna(`xauthuser`) as xauthuser agg, nullifna(`user`) as user agg, tunnelid, min(coalesce (dtime, 0)) as s time, max(coalesce(dtime, 0)) as e time, max(coalesce(duration,0)) as max duration, min(coalesce(duration,0)) as min duration, min(coalesce(sentbyte, 0)) as min traffic out, min(coalesce(rcvdbyte, 0)) as min traffic in, max(coalesce(sentbyte, 0)) as max traffic out, max(coalesce(rcvdbyte, 0)) as max traffic in, max(coalesce(rcvdbyte, 0)+coalesce(sentbyte, 0)) as max traffic 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 max traffic desc) ### t group by devid, vd, remip, tunnelid) tt group by user src having sum(bandwidth)>0 order by bandwidth desc

Dataset Name	Description	Log Category
default-Top-Sources-Of-SSL-VPN- Tunnels-By-Bandwidth	Default top sources of SSL VPN tunnels by bandwidth usage	event
<pre>in) - min(min_traffic_in) end) as traffic_in, (case when min(s_time) = out) - min(min_traffic_out) end) as traffic_out, (case when min(s_time) =</pre>	<pre>max(e_time) then max(max_traffic_in) else max(max(e_time) then max(max_traffic_out) else max max(e_time) then max(max_traffic_in)+ max(max_ nin_traffic_in)+ max(max_traffic_out)- min(min_</pre>	(max_traffic_
) as bandwidth from		
<pre>###(select \$flex_timestan tunneltype like 'ipsec%' then ' action='tunnel-up' then 1 else traffic_out, max(coalesce(rcvdk min_traffic_out, min(coalesce(n time, max(coalesce(dtime, 0)) a (tunneltype like 'ipsec%' or tu</pre>	<pre>mp as timestamp, devid, vd, remip, tunnelid, (c ipsec' else tunneltype end) as t_type, (case w 0 end) as tunnelup, max(coalesce(sentbyte, 0)) oyte, 0)) as max_traffic_in, min(coalesce(sentb covdbyte, 0)) as min_traffic_in, min(coalesce(d as e_time from \$log where \$filter and subtype=' unneltype like 'ssl%') and action in ('tunnel-u helid is not null and tunnelid!=0 group by time</pre>	hen as max_ yte, 0)) as time, 0)) as s_ vpn' and p','tunnel-

stats', 'tunnel-down') and tunnelid is not null and tunnelid!=0 group by timestamp, devid, vd, remip, t_type, tunnelid, action /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where t_type like 'ssl%' group by devid, vd, remip, tunnelid) tt group by remote_ip having sum(traffic_in+traffic_out)>0 order by bandwidth desc

Dataset Name	Description	Log Category
webfilter-Web-Activity-Summary-By- Requests	Webfilter web activity summary by requests	webfilter
end) as allowed_request, sum(case request from \$log where \$filter a	l_request,	olocked_ roup by



dt ON dm.devid=dt.devid and dm.vd=dt.vd) t2 on t1.ep_id=t2.epid and t1.eu_id=t2.euid and t1.dvid=t2.dvid left join \$ADOM_ENDPOINT t3 on t1.ep_id=t3.epid and t2.epmac=t3.mac left join \$ADOM_ENDUSER t4 on t1.eu id=t4.euid group by user src, ep 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>select coalesce(f_user, euname, ipstr(`srcip`)) as user_src, coalesce(epname, ipstr(`srcip`)) as ep_src, sum(requests) as requests from (select dvid, f_user, srcip, ep_id, eu_id, sum(requests) as requests from</pre>		
<pre>(case when epid<1024 then null e euid end) as eu_id, action, coun (nullifna(`user`), nullifna(`una user, srcip, ep_id, eu_id, actio action!='blocked' group by dvid, join (select epid, euid, srcmac dt ON dm.devid=dt.devid and dm.v t1.dvid=t2.dvid left join \$ADOM_</pre>	nullifna(`user`), nullifna(`unauthuser`)) as lse epid end) as ep_id, (case when euid<1024 t(*) as requests from \$log where \$filter and uthuser`), ipstr(`srcip`)) is not null group n /*SkipSTART*/order by requests desc/*SkipEr f_user, srcip, ep_id, eu_id order by request as epmac, dvid from \$ADOM_EPEU_DEVMAP dm inne d=dt.vd) t2 on t1.ep_id=t2.epid and t1.eu_id ENDPOINT t3 on t1.ep_id=t3.epid and t2.epmacs d=t4.euid group by user_src, ep_src order by	then null else coalesce by dvid, f_ ND*/)### t where ts desc) t1 left er join devtable =t2.euid and =t3.mac left

Dataset Name	Description	Log Category
traffic-Top-Web-Users-By-Browsing- Time	Traffic top web users by browsing time	traffic
<pre>select user_src, ebtr_value(ebtr_agg_flat(browsetime), null, \$timespan) as browsetime, sum(bandwidth) as bandwidth, sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_out from ###(select user_src, ebtr_agg_; sum(traffic_in) as traffic_in, sum(traffic_in), sum(traffic_in), straffic_in, sum(traffic_in), sum(traffic_in), straffic_in, straffic_in, sum(traffic_in), straffic_in, sum(traffic_in), straffic_in, sum(traffic_in), straffic_in, sum(traffic_in), straffic_in, sum(traffic_in), straffic_in, straffic_in, sum(traffic_in), straffic_in, st</pre>	flat(browsetime) as browsetime, ım(traffic_out) as traffic_out f	from (select coalesce

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

Dataset Name	Description	Log Category
webfilter-Top-Blocked-Web-Sites-By- Requests	Webfilter top blocked web sites by requests	webfilter
filter and (eventtype is null not null group by domain, catde	, catdesc, action, count(*) as requests fr or logver>=502000000) and hostname is not sc, action /*SkipSTART*/order by requests by domain, catdesc order by requests desc	null and catdesc i
Dataset Name	Description	Log Category
webfilter-Top-Allowed-Web-Sites-By- Requests	Webfilter top allowed web sites by requests	webfilter
select domain, string_agg(distinct catdesc, & #039;, ') as agg catdesc,		

& #039;, ') as agg_catdesc, sum(requests) as requests from ###(select hostname as domain, catdesc, action, count(*) as requests from \$log where \$filter and (eventtype is null or logver>=50200000) and hostname is not null and catdesc is not null group by domain, catdesc, action /*SkipSTART*/order by requests desc/*SkipEND*/)### t where action!='blocked' group by domain order by requests desc

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

select

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

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

Dataset Name	Description	Log Category
webfilter-Top-Blocked-Web-Categories	Webfilter top blocked web categories	webfilter
(eventtype is null or logver>=50	unt(*) as requests from \$log-webfilter 2000000) and catdesc is not null group esc/*SkipEND*/)### t where action='blo	by catdesc, action
Dataset Name	Description	Log Category
webfilter-Top-Allowed-Web-Categories	Webfilter top allowed web categories	webfilter

select

catdesc, sum(requests) as requests from

###(select catdesc, action, count(*) as requests from \$log-webfilter where \$filter and (eventtype is null or logver>=50200000) and catdesc is not null group by catdesc, action /*SkipSTART*/order by requests desc/*SkipEND*/)### t where action!='blocked' group by catdesc order by requests desc

Dataset Name	Description	Log Category
traffic-Top-50-Sites-By-Browsing-Time	Traffic top sites by browsing time	traffic
browsetime, sum(bandwidth) as ban traffic_out from ###(select host (bandwidth) as bandwidth, sum(tra from (select hostname, catdesc, e (sentbyte, 0)+coalesce(rcvdbyte, sum(coalesce(sentbyte, 0)) as tra hostname is not null and \$browse	ebtr_value(ebtr_agg_flat(browsetime) ndwidth, sum(traffic_in) as traffic_ name, catdesc, ebtr_agg_flat(browset affic_in) as traffic_in, sum(traffic ebtr_agg_flat(\$browse_time) as brows 0)) as bandwidth, sum(coalesce(rcvd affic_out from \$log where \$filter an _time is not null group by hostname, rder by ebtr_value(ebtr_agg_flat(bro ostname order by browsetime desc	<pre>in, sum(traffic_out) as ime) as browsetime, sum _out) as traffic_out etime, sum(coalesce byte, 0)) as traffic_in, d (logflag&1>0) and catdesc) t group by</pre>

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

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

Dataset Name	Description	Log Category
traffic-Top-Destination-Countries-By- Browsing-Time	Traffic top destination countries by browsing time	traffic
<pre>select dstcountry, ebtr_value(ebtr_agg_flat(browsetime), null, \$timespan) as browsetime, sum(bandwidth) as bandwidth, sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_out)</pre>	t	
<pre>from ###(select dstcountry, ebtr_ag bandwidth, sum(traffic_in) as tr</pre>	<pre>g_flat(browsetime) as browsetime, sum(bandwidt affic_in, sum(traffic_out) as traffic_out from e_time) as browsetime, sum(coalesce(sentbyte,</pre>	(select

(rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce (sentbyte, 0)) as traffic_out from \$log where \$filter and (logflag&1>0) and \$browse_time is not null group by dstcountry) t group by dstcountry /*SkipSTART*/order by ebtr_value(ebtr_ agg_flat(browsetime), null, null) desc/*SkipEND*/)### t group by dstcountry order by browsetime desc

Dataset Name	Description	Log Category
webfilter-Top-Search-Phrases	Webfilter top search phrases	webfilter
<pre>select keyword, count(*) as requests from \$log where \$filter and keyword is not null group by keyword order by requests desc</pre>		
Dataset Name	Description	Log Category
Top-10-Users-Browsing-Time	Estimated browsing time	traffic

```
select
 coalesce(
   f user,
    euname,
    ipstr(`srcip`)
  ) as user src,
  coalesce(
    epname,
   ipstr(`srcip`)
 ) as ep src,
  ebtr value(
    ebtr agg flat (browsetime),
    null,
    $timespan
 ) as browsetime
from
  (
    select
      dvid,
      f user,
      srcip,
      ep id,
      eu id,
      ebtr agg flat (browsetime) as browsetime
    from
```

###(select dvid, f_user, srcip, ep_id, eu_id, ebtr_agg_flat(browsetime) as browsetime from (select dvid, coalesce(nullifna(`user`), nullifna(`unauthuser`)) as f_user, srcip, (case when epid<1024 then null else epid end) as ep_id, (case when euid<1024 then null else euid end) as eu_id, ebtr_agg_flat(\$browse_time) as browsetime from \$log where \$filter and (logflag&1>0) and \$browse_time is not null group by dvid, f_user, srcip, ep_id, eu_id) t group by dvid, f_user, srcip, ep_id, eu_id order by ebtr_value(ebtr_agg_flat(browsetime), null, null) desc)### t group by dvid, f_user, srcip, ep_id, eu_id order by ebtr_value(ebtr_ agg_flat(browsetime), null, null) desc) t1 left join (select epid, euid, srcmac as epmac, dvid from \$ADOM_EPEU_DEVMAP dm inner join devtable dt ON dm.devid=dt.devid and dm.vd=dt.vd) t2 on t1.ep_id=t2.epid and t1.eu_id=t2.euid and t1.dvid=t2.dvid left join \$ADOM_ENDPOINT t3 on t1.ep_id=t3.epid and t2.epmac=t3.mac left join \$ADOM_ENDUSER t4 on t1.eu_id=t4.euid group by user_src, ep_src order by browsetime desc

Dataset Name	Description	Log Category
Estimated-Browsing-Time	Estimated browsing time	traffic
<pre>select coalesce(f_user, euname, ipstr(`srcip`)) as user_src, coalesce(epname, ipstr(`srcip`)) as ep_src, ebtr_value(ebtr_agg_flat(browsetime), null, \$timespan</pre>		

```
) as browsetime
from
(
    select
    dvid,
    f_user,
    srcip,
    ep_id,
    eu_id,
    ebtr_agg_flat(browsetime) as browsetime
from
```

###(select dvid, f_user, srcip, ep_id, eu_id, ebtr_agg_flat(browsetime) as browsetime from (select dvid, coalesce(nullifna(`user`), nullifna(`unauthuser`)) as f_user, srcip, (case when epid<1024 then null else epid end) as ep_id, (case when euid<1024 then null else euid end) as eu_id, ebtr_agg_flat(\$browse_time) as browsetime from \$log where \$filter and (logflag&1>0) and \$browse_time is not null group by dvid, f_user, srcip, ep_id, eu_id) t group by dvid, f_user, srcip, ep_id, eu_id order by ebtr_value(ebtr_agg_flat(browsetime), null, null) desc)### t group by dvid, f_user, srcip, ep_id, eu_id order by ebtr_value(ebtr_ agg_flat(browsetime), null, null) desc) t1 left join (select epid, euid, srcmac as epmac, dvid from \$ADOM_EPEU_DEVMAP dm inner join devtable dt ON dm.devid=dt.devid and dm.vd=dt.vd) t2 on t1.ep_id=t2.epid and t1.eu_id=t2.euid and t1.dvid=t2.dvid left join \$ADOM_ENDPOINT t3 on t1.ep_id=t3.epid and t2.epmac=t3.mac left join \$ADOM_ENDUSER t4 on t1.eu_id=t4.euid group by user_src, ep_src order by browsetime desc

Dataset Name	Description	Log Category
wifi-Top-AP-By-Client	Top access point by client	traffic
<pre>select ap_srcintf as srcintf, count(distinct srcmac) as from (select coalesce(ap, srcintf) srcmac from ###(select coalesce(nt src, ap, srcintf, srcssid, s (`srcname`), `srcmac`) as ho osname, max(osversion) as os 0)+coalesce(rcvdbyte, 0)) as and (logflag&l>0) and (srcss srcintf, srcssid, srcmac, ho desc/*SkipEND*/)### t where ap as ap_srcintf, stamac as stamac as srcmac, ap, ssid, sentdelta, sum(coalesce(rcvd (rcvddelta, 0)) as bandwidtl (`srcip`)) as user_src, sent by itime) as rcvddelta from not null and ssid is not nut</pre>	as ap_srcintf, ullifna(`user`), nullifna(`unauthuser`) srcssid as ssid, srcmac, srcmac as star ostname_mac, max(srcswversion) as srcsu sversion, max(devtype) as devtype, sum s bandwidth, count(*) as subtotal from sid is not null or dstssid is not null) ostname_mac /*SkipSTART*/order by bandu srcmac is not null group by ap_srcint: srcmac from ###(select \$flex_timestamp ssid as srcssid, user_src, sum(coalesce h from (select itime, stamac, ap, ssid, tbyte-lag(coalesce(sentbyte, 0)) over \$log-event where \$filter and subtype= ll and action in ('sta-wl-bridge-traff: timestamp, stamac, ap, ssid, user_src), ipstr(`srcip`)) as user_ mac, coalesce(nullifna wversion, max(osname) as (coalesce(sentbyte, \$log-traffic where \$filter) group by user_src, ap, width desc, subtotal f, srcmac union all (select p as timestamp, stamac, ce(sentdelta, 0)) as (sentdelta, 0)+coalesce , coalesce(`user`, ipstr (partition by stamac order (partition by stamac order 'wireless' and stamac is ic-stats', 'reassoc-req',

Dataset Name	Description	Log Category
wifi-Top-SSID-By-Bandwidth	Top SSIDs by bandwidth usage	traffic
select		
srcssid,		
sum(bandwidth) as bandwidt	h	
from		
(
select		
srcssid,		
sum(bandwidth) as band	width	
from		
###(select coalesce(nu	llifna(`user`), nullifna(`unauthuser`), i	ipstr(`srcip`)) as user_
src, ap, srcintf, srcssid, s	rcssid as ssid, srcmac, srcmac as stamac,	coalesce(nullifna
(`srcname`), `srcmac`) as ho	stname_mac, max(srcswversion) as srcswver	rsion, max(osname) as
osname, max(osversion) as os	version, max(devtype) as devtype, sum(coa	alesce(sentbyte,
0)+coalesce(rcvdbyte, 0)) as	<pre>bandwidth, count(*) as subtotal from \$10</pre>	og-traffic where \$filter
and (logflag&1>0) and (srcss	id is not null or dstssid is not null) g	coup by user_src, ap,
srcintf, srcssid, srcmac, ho	stname_mac /*SkipSTART*/order by bandwidt	ch desc, subtotal

desc/*SkipEND*/)### t where srcssid is not null group by srcssid having sum(bandwidth)>0
union all select ssid as srcssid, sum(bandwidth) as bandwidth from ###(select \$flex_

timestamp as timestamp, stamac, stamac as srcmac, ap, ssid, ssid as srcssid, user_src, sum

(coalesce(sentdelta, 0)) as sentdelta, sum(coalesce(rcvddelta, 0)) as rcvddelta, sum (coalesce(sentdelta, 0)+coalesce(rcvddelta, 0)) as bandwidth from (select itime, stamac, ap, ssid, coalesce(`user`, ipstr(`srcip`)) as user_src, sentbyte-lag(coalesce(sentbyte, 0)) over (partition by stamac order by itime) as sentdelta, rcvdbyte-lag(coalesce(rcvdbyte, 0)) over (partition by stamac order by itime) as rcvddelta from \$log-event where \$filter and subtype='wireless' and stamac is not null and ssid is not null and action in ('sta-wlbridge-traffic-stats', 'reassoc-req', 'assoc-req')) as t group by timestamp, stamac, ap, ssid, user_src /*SkipSTART*/order by bandwidth desc/*SkipEND*/)### t group by ssid having sum(bandwidth)>0) t group by srcssid order by bandwidth desc

Dataset Name	Description	Log Category
wifi-Top-SSID-Bv-Client	Top SSIDs by client	traffic

select

```
srcssid,
count(distinct srcmac) as totalnum
from
  (
    select
    srcssid,
    srcmac
    from
```

###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user src, ap, srcintf, srcssid, srcssid as ssid, srcmac, srcmac as stamac, coalesce(nullifna (`srcname`), `srcmac`) as hostname mac, max(srcswversion) as srcswversion, max(osname) as osname, max(osversion) as osversion, max(devtype) as devtype, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, count(*) as subtotal from \$log-traffic where \$filter and (logflag&1>0) and (srcssid is not null or dstssid is not null) group by user src, ap, srcintf, srcssid, srcmac, hostname mac /*SkipSTART*/order by bandwidth desc, subtotal desc/*SkipEND*/)### t where srcmac is not null group by srcssid, srcmac union all select ssid as srcssid, stamac as srcmac from ###(select \$flex timestamp as timestamp, stamac, stamac as srcmac, ap, ssid, ssid as srcssid, user src, sum(coalesce(sentdelta, 0)) as sentdelta, sum(coalesce(rcvddelta, 0)) as rcvddelta, sum(coalesce(sentdelta, 0)+coalesce (rcvddelta, 0)) as bandwidth from (select itime, stamac, ap, ssid, coalesce(`user`, ipstr (`srcip`)) as user src, sentbyte-lag(coalesce(sentbyte, 0)) over (partition by stamac order by itime) as sentdelta, rcvdbyte-lag(coalesce(rcvdbyte, 0)) over (partition by stamac order by itime) as rcvddelta from \$log-event where \$filter and subtype='wireless' and stamac is not null and ssid is not null and action in ('sta-wl-bridge-traffic-stats', 'reassoc-req', 'assoc-req')) as t group by timestamp, stamac, ap, ssid, user src /*SkipSTART*/order by bandwidth desc/*SkipEND*/)### t where stamac is not null group by ssid, stamac) t where srcssid is not null group by srcssid order by totalnum desc

Dataset Name	Description	Log Category
wifi-Top-App-By-Bandwidth	Top WiFi applications by bandwidth usage	traffic
<pre>select appid, app, sum(coalesce(sentbyte, 0)+ coale) as bandwidth from \$log where</pre>	esce(rcvdbyte, 0)	

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

Dataset Name

wifi-Top-Client-By-Bandwidth

Top WiFi client by bandwidth usage

Description

Log Category

```
select
```

```
client,
sum(bandwidth) as bandwidth
from
  (
    select
    (
    coalesce(
```

```
hostname mac,
```

& #039;unknown') || ' (' || get devtype(srcswversion, osname, devtype) || ', ' || coalesce(osname, '') || (case when srcswversion is null then '' else ' ' || srcswversion end) || ')') as client, sum(bandwidth) as bandwidth from ###(select coalesce(nullifna (`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user src, ap, srcintf, srcssid, srcssid as ssid, srcmac, srcmac as stamac, coalesce(nullifna(`srcname`), `srcmac`) as hostname_mac, max(srcswversion) as srcswversion, max(osname) as osname, max(osversion) as osversion, max (devtype) as devtype, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, count (*) as subtotal from \$log-traffic where \$filter and (logflag&1>0) and (srcssid is not null or dstssid is not null) group by user_src, ap, srcintf, srcssid, srcmac, hostname_mac /*SkipSTART*/order by bandwidth desc, subtotal desc/*SkipEND*/)### t group by client having sum(bandwidth)>0 union all select (coalesce(stamac, 'unknown')) as client, sum(bandwidth) as bandwidth from ###(select \$flex timestamp as timestamp, stamac, stamac as srcmac, ap, ssid, ssid as srcssid, user src, sum(coalesce(sentdelta, 0)) as sentdelta, sum(coalesce(rcvddelta, 0)) as rcvddelta, sum(coalesce(sentdelta, 0)+coalesce(rcvddelta, 0)) as bandwidth from (select itime, stamac, ap, ssid, coalesce(`user`, ipstr(`srcip`)) as user src, sentbyte-lag (coalesce(sentbyte, 0)) over (partition by stamac order by itime) as sentdelta, rcvdbyte-lag (coalesce(rcvdbyte, 0)) over (partition by stamac order by itime) as rcvddelta from \$logevent where \$filter and subtype='wireless' and stamac is not null and ssid is not null and action in ('sta-wl-bridge-traffic-stats', 'reassoc-req', 'assoc-req')) as t group by timestamp, stamac, ap, ssid, user src /*SkipSTART*/order by bandwidth desc/*SkipEND*/)### t group by client having sum(bandwidth) > 0) t where client is not null group by client order by bandwidth desc

Dataset Name	Description	Log Category
wifi-Top-OS-By-Bandwidth	Top WiFi os by bandwidth usage	traffic
andwidth from ###(select of s user_src, ap, srcintf, s hullifna(`srcname`), `srcn osname) as osname, max(os sentbyte, 0)+coalesce(rcvo here \$filter and (logflag ser_src, ap, srcintf, srcs	<pre>' ' coalesce(srcswversion, '')) as os, s coalesce(nullifna(`user`), nullifna(`unauth srcssid, srcssid as ssid, srcmac, srcmac as mac`) as hostname_mac, max(srcswversion) as version) as osversion, max(devtype) as devt dbyte, 0)) as bandwidth, count(*) as subtot &1>0) and (srcssid is not null or dstssid i ssid, srcmac, hostname_mac /*SkipSTART*/ord ## t group by os having sum(bandwidth)>0 or</pre>	<pre>user`), ipstr(`srcip`) s stamac, coalesce s srcswversion, max cype, sum(coalesce cal from \$log-traffic .s not null) group by der by bandwidth desc,</pre>
Dataset Name	Description	Log Category
Dataset Name wifi-Top-OS-By-WiFi-Client	Description Top WiFi os by WiFi client	Log Category traffic
<pre>wifi-Top-OS-By-WiFi-Client elect (coalesce(osname,</pre>		traffic ht(distinct srcmac) as hser`), ipstr(`srcip`)) s stamac, coalesce s srcswversion, max type, sum(coalesce tal from \$log-traffic s not null) group by her by bandwidth desc,
<pre>wifi-Top-OS-By-WiFi-Client elect (coalesce(osname,</pre>	Top WiFi os by WiFi client ' ' coalesce(osversion, '')) as os, cour oalesce(nullifna(`user`), nullifna(`unauthu srcssid, srcssid as ssid, srcmac, srcmac as mac`) as hostname_mac, max(srcswversion) as version) as osversion, max(devtype) as devt dbyte, 0)) as bandwidth, count(*) as subtot &1>0) and (srcssid is not null or dstssid i ssid, srcmac, hostname_mac /*SkipSTART*/ord	traffic ht(distinct srcmac) as hser`), ipstr(`srcip`)) s stamac, coalesce s srcswversion, max type, sum(coalesce tal from \$log-traffic s not null) group by her by bandwidth desc,

select

get_devtype(srcswversion, osname, devtype) as devtype_new, sum(bandwidth) as bandwidth

from

###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, ap, srcintf, srcssid, srcssid as ssid, srcmac, srcmac as stamac, coalesce(nullifna (`srcname`), `srcmac`) as hostname_mac, max(srcswversion) as srcswversion, max(osname) as osname, max(osversion) as osversion, max(devtype) as devtype, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, count(*) as subtotal from \$log-traffic where \$filter and (logflag&l>0) and (srcssid is not null or dstssid is not null) group by user_src, ap, srcintf, srcssid, srcmac, hostname_mac /*SkipSTART*/order by bandwidth desc, subtotal desc/*SkipEND*/)### t where devtype is not null group by devtype_new having sum(bandwidth)>0 order by bandwidth desc

Dataset Name	Description	Log Category
wifi-Top-Device-By-Client	Top WiFi device by client	traffic
<pre>srcmac from ###(select coalesce(src, ap, srcintf, srcssid, (`srcname`), `srcmac`) as osname, max(osversion) as 0)+coalesce(rcvdbyte, 0)) and (logflag&1>0) and (src srcintf, srcssid, srcmac,</pre>	rsion, osname, devtype) as devtype_new, nullifna(`user`), nullifna(`unauthuser`) srcssid as ssid, srcmac, srcmac as stan hostname_mac, max(srcswversion) as srcsw osversion, max(devtype) as devtype, sum as bandwidth, count(*) as subtotal from ssid is not null or dstssid is not null) hostname_mac /*SkipSTART*/order by bandw re srcmac is not null) t where devtype_ne	<pre>mac, coalesce(nullifna wversion, max(osname) as (coalesce(sentbyte, \$log-traffic where \$filter group by user_src, ap, width desc, subtotal</pre>
Dataset Name	Description	Log Category
wifi-Overall-Traffic	WiFi overall traffic	traffic
select sum(bandwidth) as bandwi	dth	

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

sum(bandwidth) as bandwidth

```
from
```

###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user src, ap, srcintf, srcssid, srcssid as ssid, srcmac, srcmac as stamac, coalesce(nullifna (`srcname`), `srcmac`) as hostname mac, max(srcswversion) as srcswversion, max(osname) as osname, max(osversion) as osversion, max(devtype) as devtype, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, count(*) as subtotal from \$log-traffic where \$filter and (logflag&1>0) and (srcssid is not null or dstssid is not null) group by user src, ap, srcintf, srcssid, srcmac, hostname mac /*SkipSTART*/order by bandwidth desc, subtotal desc/*SkipEND*/)### t group by srcssid union all select sum(bandwidth) as bandwidth from ### (select \$flex_timestamp as timestamp, stamac, stamac as srcmac, ap, ssid, ssid as srcssid, user src, sum(coalesce(sentdelta, 0)) as sentdelta, sum(coalesce(rcvddelta, 0)) as rcvddelta, sum(coalesce(sentdelta, 0)+coalesce(rcvddelta, 0)) as bandwidth from (select itime, stamac, ap, ssid, coalesce(`user`, ipstr(`srcip`)) as user_src, sentbyte-lag(coalesce (sentbyte, 0)) over (partition by stamac order by itime) as sentdelta, rcvdbyte-lag(coalesce (rcvdbyte, 0)) over (partition by stamac order by itime) as rcvddelta from \$log-event where \$filter and subtype='wireless' and stamac is not null and ssid is not null and action in ('sta-wl-bridge-traffic-stats', 'reassoc-req', 'assoc-req')) as t group by timestamp, stamac, ap, ssid, user src /*SkipSTART*/order by bandwidth desc/*SkipEND*/)### t) t

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

```
select
  count(distinct srcmac) as totalnum
from
  (
    select
    srcmac
```

from

###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user src, ap, srcintf, srcssid, srcssid as ssid, srcmac, srcmac as stamac, coalesce(nullifna (`srcname`), `srcmac`) as hostname mac, max(srcswversion) as srcswversion, max(osname) as osname, max(osversion) as osversion, max(devtype) as devtype, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, count(*) as subtotal from \$log-traffic where \$filter and (logflag&1>0) and (srcssid is not null or dstssid is not null) group by user src, ap, srcintf, srcssid, srcmac, hostname mac /*SkipSTART*/order by bandwidth desc, subtotal desc/*SkipEND*/)### t where srcmac is not null group by srcmac union all select stamac as srcmac from ###(select \$flex timestamp as timestamp, stamac, stamac as srcmac, ap, ssid, ssid as srcssid, user src, sum(coalesce(sentdelta, 0)) as sentdelta, sum(coalesce(rcvddelta, 0)) as rcvddelta, sum(coalesce(sentdelta, 0)+coalesce(rcvddelta, 0)) as bandwidth from (select itime, stamac, ap, ssid, coalesce(`user`, ipstr(`srcip`)) as user src, sentbyte-lag (coalesce(sentbyte, 0)) over (partition by stamac order by itime) as sentdelta, rcvdbyte-lag (coalesce(rcvdbyte, 0)) over (partition by stamac order by itime) as rcvddelta from \$logevent where \$filter and subtype='wireless' and stamac is not null and ssid is not null and action in ('sta-wl-bridge-traffic-stats', 'reassoc-req', 'assoc-req')) as t group by timestamp, stamac, ap, ssid, user src /*SkipSTART*/order by bandwidth desc/*SkipEND*/)### t where stamac is not null group by stamac) t

```
Dataset Name
                                    Description
                                                                                      Log Category
Top30-Subnets-by-Bandwidth-and-
                                   Top subnets by application bandwidth
                                                                                      traffic
 Sessions
select
  ip subnet(`srcip`) as subnet,
  sum(
    coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
  ) as bandwidth,
  sum(
    coalesce(rcvdbyte, 0)
  ) as traffic in,
  sum(
   coalesce(sentbyte, 0)
  ) as traffic out,
 count(*) as sessions
from
  $log
where
  $filter
  and (
    logflag&1>0
  )
group by
 subnet
having
  sum (
    coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
```

) > 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_ sum(coalesce(sentbyte, 0)+ co) as bandwidth from \$log where \$filter and (logflag&1>0) and nullifna(app) is not nut group by subnet, app_group having sum(coalesce(sentbyte, 0)+ co)> 0 order by bandwidth desc</pre>	_group, balesce(rcvdbyte, 0)	
Dataset Name	Description	Log Category

Dataset Name

Top30-Subnets-by-Application-	

Top applications by sessions

```
Sessions
```

```
select
 ip_subnet(`srcip`) as subnet,
 app_group_name(app) as app_group,
 count(*) as sessions
from
 $log
where
 $filter
 and (
   logflag&1>0
 )
 and nullifna(app) is not null
group by
 subnet,
 app_group
order by
 sessions desc
```

traffic

Dataset Name	Description	Log Category
Top30-Subnets-by-Website-Bandwidth	Top websites and web category by bandwidth	traffic
select subnet, website, sum(bandwidth) as bandwidth		
<pre>from ###(select ip_subnet(`srcip`) as subnet, hostname as website, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from \$log-traffic where \$filter and hostname is not null and (logflag&l>0) and (countweb>0 or ((logver is null or logver<50200000) and (hostname is null and (logflag&l>0)</pre>		

(hostname is not null or utmevent in ('webfilter', 'banned-word', 'web-content', 'commandblock', 'script-filter')))) group by subnet, website order by bandwidth desc)### t group by subnet, website order by bandwidth desc

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

```
website,
sum(hits) as hits
```

from

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

Dataset Name	Description	Log Category
Top30-Subnets-with-Top10-User-by- Bandwidth	Top users by bandwidth	traffic
<pre>Bandwidth select ip_subnet(`srcip`) as subnet, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentbyte, 0)+ coale) as bandwidth from \$log where \$filter and (logflag&1>0</pre>		
) and srcip is not null group by subnet, user_src having		

```
sum(
   coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
 ) > 0
order by
 bandwidth desc
```

Dataset Name	Description	Log Category
Top30-Subnets-with-Top10-User-by- Sessions	Top users by sessions	traffic
<pre>select ip_subnet(`srcip`) as subnet, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, count(*) as sessions from \$log where \$filter and (logflag&1>0) group by subnet, user_src order by sessions desc</pre>		
Dataset Name	Description	Log Category
app-Top-20-Category-and-	Top category and applications by bandwidth usage	traffic

app-1op-20-Category-and- Applications-by-Bandwidth	l op category and a
<pre>select appcat, app, sum(coalesce(sentbyte,) as bandwidth from \$log where \$filter</pre>	0)+ coalesce(rcvdbyte, 0)
YIIICEI	

```
and (
   logflag&1>0
  )
group by
 appcat,
  app
having
  sum(
    coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
```

)> 0 order by bandwidth desc

Description	Log Category
Top category and applications by session	traffic

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

```
select
 from_itime(itime) as timestamp,
 coalesce(
   nullifna(`user`),
  nullifna(`unauthuser`),
   ipstr(`srcip`)
 ) as user_src,
  appcat,
 app,
 coalesce(
  root_domain(hostname),
   ipstr(dstip)
 ) as destination,
 sum(
   coalesce(`sentbyte`, 0)+ coalesce(`rcvdbyte`, 0)
 ) as bandwidth
from
  $log
where
 $filter
 and (
   logflag&1>0
  )
 and action in (
```

Dataset Name	Description	Log Category
app-Top-500-Blocked-Applications-by- Session	Top blocked applications by session	traffic
<pre>select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, appcat, app, count(*) as sessions</pre>		
<pre>irom \$log where \$filter and (logflag&1>0)</pre>		
and action in (set', 'dropped') group by user_src, appcat,	app order by

	•	
web-Detailed-Website-Browsing-Log	Web detailed website browsing log	traffic
(sentbyte, 0)+coalesce(rcvdbyte hostname is not null and (logf) logver<502000000) and (hostname 'web-content', 'command-block',	<pre>c, stname, cast(utmaction as text) as st , 0)) as bandwidth from \$log-traffic ag&1>0) and (countweb>0 or ((logver i is not null or utmevent in ('webfilt 'script-filter')))) group by dtime, ## t group by dtime, catdesc, website</pre>	where \$filter and s null or er', 'banned-word', catdesc, hostname,
Dataset Name	Description	Log Category
web-Hourly-Category-and-Website- Hits-Action	Web hourly category and website hits action	webfilter

```
select
hod,
website,
sum(hits) as hits
```

from

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

```
Dataset Name
                                  Description
                                                                                  Log Category
web-Top-20-Category-and-Websites-
                                  Web top category and websites by bandwidth usage
                                                                                  traffic
 by-Bandwidth
select
  website,
  catdesc,
  sum (bandwidth) as bandwidth
from
  ###(select hostname as website, catdesc, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))
as bandwidth from $log-traffic where $filter and hostname is not null and (logflag&1>0) and
(countweb>0 or ((logver is null or logver<50200000) and (hostname is not null or utmevent
in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter')))) group by
website, catdesc order by bandwidth desc)### t group by website, catdesc order by bandwidth
desc
```

Dataset Name	Description	Log Category
web-Top-20-Category-and-Websites- by-Session	Web top category and websites by session	webfilter
select		

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

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

Dataset Name	Description	Log Category
web-Top-500-Website-Sessions-by- Bandwidth	Web top website sessions by bandwidth usage	traffic
<pre>user_src, hostname as website, ca (sentbyte, 0)+coalesce(rcvdbyte,</pre>		sum(coalesce d hostname is

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

Dataset Name	Description	Log Category
web-Top-500-User-Visted-Websites- by-Bandwidth	Web top user visted websites by bandwidth usage	traffic
<pre>select website, catdesc,</pre>		

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

```
from
```

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

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

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

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

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

select

subtype, count(distinct fctuid) as totalnum from

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

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

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

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

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

select

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

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

Dataset Name	Description	Log Category
fct-Endpoint-Profile-Deployment	Endpoint Profile Deployment	fct-event

select

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

```
from
```

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

Dataset Name	Description	Log Category
fct-Client-Summary	Client Summary	fct-event
<pre>select hostname, deviceip, os_short as os, profile, fctver, from_itime(max(itime)) as last_seen</pre>		
(usingpolicy) as profile, fctv null group by hostname, device	p, regexp_replace(os, '\\(build ver, max(itime) as itime from \$1 eip, os_short, profile, fctver o ofile, fctver order by last seen	og where \$filter and os is not order by itime desc)### t group

Dataset Name	Description	Log Category
fct-Total-Threats-Found	Total Threats Found	fct-traffic
<pre>select utmevent_s as utmevent,</pre>		

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 totalnum</pre>		
from (
(
select threat,		
sum(totalnum) as totalnu from	ım	
<pre>###(select threat, count threat is not null and lower(utr</pre>	(*) as totalnum from \$log-fct-t	
desc)### t group by threat) unio		-

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

Dataset Name	Description	Log Category
fct-Top10-Infected-Devices-with- Botnet	Top Infected Devices with Botnet	fct-traffic
<pre>select hostname, count(*) as totalnum from \$log where \$filter and hostname is not null and lower(utmevent) in (& #039;webfilter', 'appfirew order by totalnum desc</pre>	all') and lower(threat) like '%bo	otnet%' group by hostname

Dataset Name	Description	Log Category
fct-Top10-Infected-Devices-with-Virus- Malware	Top Infected Devices with Virus Malware	fct-traffic
select hostname, sum(totalnum) as totalnum		
from		
(

```
select
```

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

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

Dataset Name	Description	Log Category
fct-All-Antivirus-Antimalware- Detections	All Antivirus and Antimalware Detections	fct-traffic
<pre>select threat, hostname, hostuser, utmaction, from_dtime(max(dtime)) as last_seen from ((select threat, hostname, hostuser, utmaction, max(dtime) as dtime from ###(select threat, hostr utmaction, max(dtime) as dtime f ('antivirus', 'antimalware') gro threat)### t group by threat, host</pre>	name, coalesce(nullifna(`user`), 'Unknown From \$log-fct-traffic where \$filter and lo bup by threat, hostname, hostuser, utmaction batname, hostuser, utmaction) union all (s hax(dtime) as dtime from ###(select virus	ower(utmevent) in ion order by select threat,
as dtime from \$log-fct-event whe	er`), 'Unknown') as hostuser, action as ut ere \$filter and (logflag is null or logfla ame, hostuser, utmaction order by threat)	ag&8=0) and virus is

not null group by threat, hostname, hostuser, utmaction order by threat)### t group by threat, hostname, hostuser, utmaction)) t group by threat, hostname, hostuser, utmaction order by threat

Dataset Name	Description	Log Category
fct-Web-Filter-Violations	Web Filter Violations	fct-traffic
<pre>last_seen from ###(select remo</pre>	utmaction, sum(total) as totalnum, tename, hostname, coalesce(nullifn as total, max(dtime) as dtime from	a(`user`), 'Unknown') as

lower(utmevent)='webfilter' and utmaction='blocked' group by remotename, hostname, hostuser, utmaction order by total desc)### t group by hostuser, hostname, utmaction order by totalnum desc

Dataset Name	Description	Log Category
fct-Application-Firewall	Application Firewall	fct-traffic
<pre>select threat, hostname, hostuser, utmaction, from_dtime(max(dtime)) as last_seen</pre>		
	name, coalesce(nullifna(`user`), 'Unkno Slog where \$filter and lower(utmevent)=	

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

Dataset Name	Description	Log Category
fct-Errors-and-Alerts	Errors and Alerts	fct-event
<pre>select msg, hostname, hostuser, from_dtime(max(dtime)) as last_seen from</pre>		
<pre>###(select msg, hostname, dtime from \$log where \$filt</pre>	<pre>coalesce(nullifna(`user`), 'Unkno er and level in ('error', 'alert') c)### t group by msg, hostname, ho</pre>	group by msg, hostname,

Dataset Name	Description	Log Category
fct-Threats-by-Top-Devices	Threats by Top Devices	fct-traffic
<pre>select hostname, count(*) as totalnum from \$log</pre>		
where \$filter and hostname is not null and utmevent is not null	group by hostname order by totalnum desc	

Dataset Name	Description	Log Category
fct-vuln-Device-Vulnerabilities	Vulnerabilities Detected by User/Device	fct-netscan
WHEN 'Info' THEN 2 WHEN 'Low' vulnname) as vuln_num from ## and nullifna(vulnseverity) is	<pre>#039;Critical' THEN 5 WHEN 'High' THEN 4 W ' THEN 1 ELSE 0 END) as severity_number, co ##(select vulnseverity, devid, vulnname from s not null and nullifna(vulnname) is not nu a) ### t group by vulnseverity order by sevence </pre>	unt(distinct m \$log where \$filter ll group by
Dataset Name	Description	Log Category

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

vulncat, count(distinct vulnname) as totalnum

from

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

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

select

os,

count(distinct vulnname) as totalnum

from

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

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

select

```
vulnseverity,
```

(

case when vulnseverity =& #039;Critical' then 5 when vulnseverity='High' then 4 when vulnseverity='Medium' then 3 when vulnseverity='Low' then 2 when vulnseverity='Info' then 1 else 0 end) as severity_number, count(distinct vulnname) as vuln_num, count(distinct devid) as dev_num from ###(select vulnseverity, devid, vulnname from \$log where \$filter and nullifna(vulnseverity) is not null and nullifna(vulnname) is not null group by vulnseverity, vulnname, devid)### t where nullifna(devid) is not null 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

vulnseverity,

(

case when vulnseverity =& #039;Critical' then 5 when vulnseverity='High' then 4 when vulnseverity='Medium' then 3 when vulnseverity='Low' then 2 when vulnseverity='Info' then 1 else 0 end) as severity_number, count(distinct vulnname) as vuln_num, count(distinct devid) as dev_num from ###(select vulnseverity, devid, vulnname from \$log where \$filter and nullifna(vulnseverity) is not null and nullifna(vulnname) is not null group by vulnseverity, vulnname, devid)### t where nullifna(devid) is not null group by vulnseverity order by dev_ num desc, severity number desc

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

```
select
```

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

from

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

Dataset Name	Description	Log Category
fct-vuln-Details-by-Risk-Level-Device	Vulnerability Details for Each Risk Level by Device	fct-netscan
select		

```
hostname,
os,
vulnseverity,
count(distinct vulnname) as vuln_num,
count(distinct products) as products,
count(distinct cve_id) as cve_count
```

from

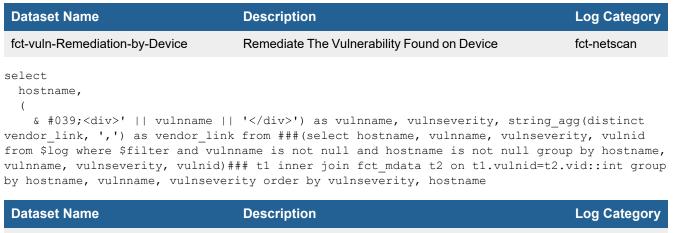
###(select hostname, os, vulnname, vulnseverity, vulnid from \$log where \$filter and vulnname is not null and vulnseverity is not null and hostname is not null group by hostname, os, vulnname, vulnseverity, vulnid)### t1 left join fct_mdata t2 on t1.vulnid=t2.vid::int group by hostname, os, vulnseverity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

Dataset Name	Description	Log Category
fct-vuln-Details-by-Device-User	Vulnerability Details by Device User	fct-netscan
select		

```
hostname,
```

(

& #039;<div>' || vulnname || '</div>') as vulnname, vulnseverity, vulncat, string_agg (distinct products, ',') as products, string_agg(distinct cve_id, ',') as cve_list, ('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 Calegory
fct-vuln-Remediation-by-Vulnerability	Remediation by Vulnerability	fct-netscan

(

Dataset Name	Description	Log Category
fct-vuln-Top-30-Targeted-High-Risk- Vulnerabilities	Top 30 Targeted High Risk Vulnerabilities	s fct-netscan
<pre>Infomation') as vendor_lin vulnid)### t1 inner join fct_r</pre>	<pre>'</pre>	

```
from
```

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

Dataset Name	Description	Log Category
fct-Top-Malware-Detections	Top Infected Devices with Malware	fct-traffic
select		
hostname,		
fctuid,		
sum(totalnum) as totalnum		
from		
(
(
select		
hostname,		
fctuid,		
sum(totalnum) as total	num	
from		
	<pre>tname, coalesce(nullifna(`user`), 'Unkno uid as fatuid as ant (t) as tata laure for</pre>	

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

Dataset Name	Description	Log Category
fct-Top10-Malware-Detections	Top 10 Infected Devices with Malware	fct-traffic
<pre>select threat, hostname, hostuser, utmaction, fctuid, sum(totalnum) as totalnum from</pre>		
((select threat, hostname, hostuser, utmaction, fctuid, sum(totalnum) as totalr	um	
utmaction, max(dtime) as dtime, where \$filter and lower(utmever	name, coalesce(nullifna(`user`), 'Unkn uid as fctuid, count(*) as totalnum f t) in ('antivirus', 'antimalware') gro by threat)### t group by threat, hostn	From \$log-fct-traffic oup by threat, hostname,

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

Dataset Name	Description	Log Category
fct-Devices-with-Botnet	Infected Devices with Botnet	fct-traffic
select		
threat,		
hostname,		
coalesce(
<pre>nullifna(`user`),</pre>		
& #039;Unknown') as hostu	ser, utmaction, uid as fctuid, count(*) as totalnum from \$log
	not null and lower(utmevent) in ('web	
	et%' group by threat, hostname, hostus	er, utmaction, fctuid
order by totalnum desc		

Dataset Name	Description	Log Category
fct-vuln-Vulnerability-by-Hostname	Vulnerability Details for Each Risk Level by Device	fct-netscan

select

```
hostname,
os,
vulnseverity,
count(distinct vulnname) as vuln_num,
count(distinct products) as products,
count(distinct cve_id) as cve_count
from
```

from

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

Dataset Name	Description	Log Category
		209 000090.9
fct-Users-With-Web-Violations	Web Filter Violations	fct-traffic
select		
hostuser,		
hostname,		
string_agg(
distinct remotename,		
& #039;,') as remotename,	utmaction, sum(total) as totalnum	n, from_dtime(max(dtime)) as
last_seen from ###(select remo	otename, hostname, coalesce(nullif	na(`user`), 'Unknown') as
hostuser, utmaction, count(*) as total, max(dtime) as dtime from \$log where \$filter and		
lower(utmevent)='webfilter' and utmaction='blocked' group by remotename, hostname, hostuser,		
utmaction order by total desc)### t group by hostuser, hostname	, utmaction order by totalnum
desc		

Dataset Name	Description	Log Category
fct-Compliance-by-FortiGate	FortiClinet Compliance by FortiGate Enforcing	fct-event
select		
fgtserial,		
count(distinct fctuid) as	totalnum	
from		
(
select		
fgtserial,		
fctuid,		
max(compliance flag) a	s compliance flag	
from	—	
###(select uid as fctu	<pre>id, regexp replace(os, '\\(build.*', '') as os</pre>	s short, fctver,
subtype, fotserial, max(case	when msg like 'Compliance rules%applied' ther	-

subtype, fgtserial, max(case when msg like 'Compliance rules%applied' then 1 else 0 end) as compliance_flag from \$log where \$filter and subtype != 'admin' group by uid, os_short, fctver, subtype, fgtserial order by compliance_flag desc)### tt group by fgtserial, fctuid) t where compliance_flag = 1 group by fgtserial order by totalnum desc

Dataset Name	Description	Log Category
fct-Compliance-Status	Number of FortiClinets by Compliance Status	fct-event

select

(

case compliance_flag when 1 then & #039;Compliant' else 'Non-Compliant' end) as compliance, count(distinct fctuid) as totalnum from (select fctuid, max(compliance_flag) as compliance_flag from ###(select uid as fctuid, regexp_replace(os, '\\(build.*', '') as os_ short, fctver, subtype, fgtserial, max(case when msg like 'Compliance rules%applied' then 1 else 0 end) as compliance_flag from \$log where \$filter and subtype != 'admin' group by uid, os_short, fctver, subtype, fgtserial order by compliance_flag desc)### tt group by fctuid) t group by compliance order by totalnum desc

Dataset Name	Description	Log Category
fct-Non-Compliant-Endpoints	Non-compliant Endpoints	fct-event
<pre>compliance_flag from ###(select short, fctver, subtype, fgtseria else 0 end) as compliance_flag f os_short, fctver, subtype, fgtse fgtserial, fctuid) t1 left join</pre>	is from (select fgtserial, fctuid, max(uid as fctuid, regexp_replace(os, '\\(il, max(case when msg like 'Compliance from \$log where \$filter and subtype != erial order by compliance_flag desc)### \$ADOM_ENDPOINT t2 on t1.fctuid = t2.fc epid where compliance_flag = 0 group by name, t2.mac	build.*', '') as os_ rules%applied' then 1 'admin' group by uid, tt group by tuid left join \$ADOM_

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

```
select
```

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

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

Dataset Name	Description	Log Category
fct-Traffic-Top-Allowed-Web-Cat	Top Visited Web Categories	fct-traffic
<pre>count(*) as requests from \$log v (utmevent)='webfilter' group by</pre>	as category, remotename as website, direction where \$filter and threat is not null and lowe: category, website, direction, utmaction orde: bound' and utmaction='passthrough' group by ca	r r by requests

Dataset Name	Description	Log Category
fct-Traffic-Top-Allowed-Website	Top Visited Websites	fct-traffic

select

```
website,
string agg(
```

```
distinct category,
```

& #039;, ') as agg_category, sum(requests) as requests from ###(select fct_webcat (threat) as category, remotename as website, direction, utmaction, count(*) as requests from \$log where \$filter and threat is not null and lower(utmevent)='webfilter' group by category, website, direction, utmaction order by requests desc)### t where direction='outbound' and utmaction='passthrough' and website is not null group by website order by requests desc

Dataset Name	Description	Log Category
fct-Traffic-Top-Category-By-Website- Session	Top Web Categories by Website Session	fct-traffic
select		

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

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

Dataset Name	Description	Log Category
fct-Traffic-Top-Web-Users-By-Website	Top Web Users by Website	fct-traffic

select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, remotename as website, count(*) as requests from \$log where \$filter

and direction =& #039;outbound' and remotename is not null and utmaction='passthrough' and lower(utmevent)='webfilter' group by user_src, website order by requests desc

Dataset Name	Description	Log Category
os-Detect-OS-Count	Detected operation system count	traffic
select		

```
select
```

```
coalesce(
osname,
```

& #039;Unknown')) as os, count(*) as totalnum from \$log where \$filter and (logflag&1>0) group by os order by totalnum desc

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

select

```
appid,
app,
sum(sessions) as sessions
```

from

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

Dataset Name	Description	Log Category
drilldown-Top-App-By-Sessions-Bar	Drilldown top applications by session count	traffic
select appid, app, sum(sessions) as sessions		
as user_src, dstip, srcintf, dst 0)+coalesce(rcvdbyte, 0)) as bar group by appid, app, user_src, c	ce(nullifna(`user`), nullifna(`unauthuser`), cintf, policyid, count(*) as sessions, sum(c adwidth from \$log where \$filter-exclude-var dstip, srcintf, dstintf, policyid order by s allifna(app) is not null group by appid, app	<pre>oalesce(sentbyte, and (logflag&1>0) essions desc)###</pre>

Dataset Name	Description	Log Category
drilldown-Top-App-By-Bandwidth- Table	Drilldown top applications by bandwidth usage	traffic
select appid, app, sum(bandwidth) as bandwidth		
from		
###(select appid, app, coale	esce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`))
as user_src, dstip, srcintf, c	dstintf, policyid, count(*) as sessions, sum	(coalesce(sentbyte
	dstintf, policyid, count(*) as sessions, sum pandwidth from \$log where \$filter-exclude-va	-

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

Dataset Name	Description	Log Category
drilldown-Top-App-By-Bandwidth-Bar	Drilldown top applications by bandwidth usage	traffic
select appid, app, sum(bandwidth) as bandwidth		
<pre>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(sentbyt 0)+coalesce(rcvdbyte, 0)) as bandwidth from \$log where \$filter-exclude-var and (logflag&l> group by appid, app, user_src, dstip, srcintf, dstintf, policyid order by sessions desc)## t where \$filter-drilldown and nullifna(app) is not null group by appid, app having sum (bandwidth)>0 order by bandwidth desc</pre>		alesce(sentbyte, nd (logflag&1>0) ssions desc)###

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

sum(sessions) as sessions from

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

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

```
dstip,
```

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

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

Dataset Name	Description	Log Category
drilldown-Top-User-By-Sessions-Table	Drilldown top user by session count	traffic
<pre>select user_src, sum(sessions) as sessions</pre>		
<pre>from ###(select appid, app, coalesc</pre>	e(nullifna(`user`), nullifna(`unauthus	ser`), ipstr(`srcip`))

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

Dataset Name	Description	Log Category
drilldown-Top-User-By-Sessions-Bar	Drilldown top user by session count	traffic

select

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

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

Dataset Name	Description	Log Category
drilldown-Top-User-By-Bandwidth-	Drilldown top user by bandwidth usage	traffic
Table		

select

user_src, sum(bandwidth) as bandwidth from

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

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

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

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

Dataset Name	Description	Log Category
drilldown-Top-Web-User-By-Visit- Table	Drilldown top web user by visit	traffic
<pre>select user_src,</pre>		
sum(requests) as visits from		

(

###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_ src, hostname, count(*) as requests from \$log-traffic where \$filter-exclude-var and (logflag&1>0) and utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') and hostname is not null group by user_src, hostname order by requests desc)### union all ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, hostname, count(*) as requests from \$log-webfilter where \$filter-exclude-var and (eventtype is null or logver>=50200000) 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
<pre>select user_src, sum(requests) as visits from (###(select coalesce(nullifna src, hostname, count(*) as reque (logflag&1>0) and utmevent in (' 'script-filter') and hostname is desc)### union all ###(select co hostname, count(*) as requests f is null or logver>=502000000) an</pre>	(`user`), nullifna(`unauthuser`), ipstr sts from \$log-traffic where \$filter-exc webfilter', 'banned-word', 'web-content not null group by user_src, hostname o alesce(nullifna(`user`), ipstr(`srcip`) rom \$log-webfilter where \$filter-exclud d hostname is not null group by user_sr	(`srcip`)) as user_ lude-var and ', 'command-block', rder by requests) as user_src, e-var and (eventtype c, hostname order by
requests desc)###) t where \$filt order by visits desc	er-drilldown and user_src is not null g	roup by user_src

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

```
select
hostname,
sum(requests) as visits
from
  (
    ###(select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_
src, hostname, count(*) as requests from $log-traffic where $filter-exclude-var and
(logflag&1>0) and utmevent in ('webfilter', 'banned-word', 'web-content', 'command-block',
```

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

Dataset Name	Description	Log Category
drilldown-Top-Website-By-Request- Bar	Drilldown top website by request	traffic
<pre>src, hostname, count(*) as reque (logflag&1>0) and utmevent in (' 'script-filter') and hostname is desc)### union all ###(select co hostname, count(*) as requests f is null or logver>=502000000) an</pre>	(`user`), nullifna(`unauthuser`), ipstr(`srcip sts from \$log-traffic where \$filter-exclude-va webfilter', 'banned-word', 'web-content', 'com not null group by user_src, hostname order by alesce(nullifna(`user`), ipstr(`srcip`)) as us rom \$log-webfilter where \$filter-exclude-var a d hostname is not null group by user_src, host er-drilldown and hostname is not null group by	ar and mmand-block', requests ser_src, and (eventtype cname order by

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

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

(

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

Dataset Name	Description	Log Category
drilldown-Top-Email-Send-Recipient- By-Volume	Drilldown top email send recipient by volume	traffic
<pre>(rcvdbyte, 0)) as bandwidth from and service in ('smtp', 'SMTP', ' utmevent in ('general-email-log', desc)### union all ###(select `fr (coalesce(sentbyte, 0)+coalesce(r \$filter-exclude-var and service i '465/tcp') and eventtype is null</pre>	, count(*) as requests, sum(coalesce(sentbyte, \$log-traffic where \$filter-exclude-var and (1 '25/tcp', '587/tcp', 'smtps', 'SMTPS', '465/tc , 'spamfilter') group by sender, recipient orc rom` as sender, `to` as recipient, count(*) as rcvdbyte, 0)) as bandwidth from \$log-emailfilt in ('smtp', 'SMTP', '25/tcp', '587/tcp', 'smtp group by `from`, `to` order by requests desc) is not null group by recipient having sum(band	ogflag&1>0) cp') and der by requests s requests, sum cer where os', 'SMTPS', ###) t where

Dataset Name	Description	Log Category
drilldown-Top-Email-Sender-By-Count	Drilldown top email sender by count	traffic

select
 sender,
 sum(requests) as requests
from

(

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

Dataset Name	Description	Log Category
drilldown-Top-Email-Send-Recipient- By-Count	Drilldown top email send recipient by count	traffic
<pre>select recipient, sum(requests) as requests from</pre>		
<pre>(rcvdbyte, 0)) as bandwidth from and service in ('smtp', 'SMTP', utmevent in ('general-email-log' desc)### union all ###(select `f</pre>	<pre>c, count(*) as requests, sum(coalesce(sentbyte n \$log-traffic where \$filter-exclude-var and ('25/tcp', '587/tcp', 'smtps', 'SMTPS', '465/t ', 'spamfilter') group by sender, recipient or from` as sender, `to` as recipient, count(*) a (rcvdbyte, 0)) as bandwidth from \$log-emailfil</pre>	logflag&1>0) cp') and der by requests s requests, sum

\$filter-exclude-var and service in ('smtp', 'SMTP', '25/tcp', '587/tcp', 'smtps', 'SMTPS', '465/tcp') and eventtype is null group by `from`, `to` order by requests desc)###) t where \$filter-drilldown and recipient is not null group by recipient order by requests desc

Dataset Name	Description	Log Category
drilldown-Top-Email-Recipient-By- Volume	Drilldown top email receiver by volume	traffic
<pre>(rcvdbyte, 0)) as bandwidth from service in ('pop3', 'POP3', '110 '993/tcp', 'pop3s', 'POP3S', '99 group by recipient, sender order recipient, `from` as sender, cou (rcvdbyte, 0)) as bandwidth from ('pop3', 'POP3', '110/tcp', 'ima 'POP3S', '995/tcp') and eventtyp</pre>	<pre>, count(*) as requests, sum(coalesce(sentbyte, \$log where \$filter-exclude-var and (logflag&1 /tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IM 5/tcp') and utmevent in ('general-email-log', by requests desc)### union all ###(select `tc nt(*) as requests, sum(coalesce(sentbyte, 0)+c \$log-emailfilter where \$filter-exclude-var ar p', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993, e is null group by `to`, `from` order by reque cipient is not null group by recipient having sc</pre>	<pre>1>0) and MAPS', 'spamfilter') o`as coalesce nd service in (tcp', 'pop3s', ests desc)###)</pre>

Dataset Name	Description	Log Category
drilldown-Top-Email-Receive-Sender- By-Volume	Drilldown top email receive sender by volume	traffic
<pre>(rcvdbyte, 0)) as bandwidth from service in ('pop3', 'POP3', '110, '993/tcp', 'pop3s', 'POP3S', '999 group by recipient, sender order recipient, `from` as sender, cour (rcvdbyte, 0)) as bandwidth from ('pop3', 'POP3', '110/tcp', 'imag 'POP3S', '995/tcp') and eventtype</pre>	, count(*) as requests, sum(coalesce(sentbyte, \$log where \$filter-exclude-var and (logflag&1 /tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IM 5/tcp') and utmevent in ('general-email-log', by requests desc)### union all ###(select `to nt(*) as requests, sum(coalesce(sentbyte, 0)+c \$log-emailfilter where \$filter-exclude-var an p', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/ e is null group by `to`, `from` order by reque nder is not null group by sender having sum(ba	<pre>>0) and APS', 'spamfilter') ` as oalesce d service in tcp', 'pop3s', sts desc)###)</pre>

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

from (

###(select recipient, sender, count(*) as requests, sum(coalesce(sentbyte, 0)+coalesce (rcvdbyte, 0)) as bandwidth from \$log where \$filter-exclude-var and (logflag&1>0) and service in ('pop3', 'POP3', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/tcp', 'pop3s', 'POP3S', '995/tcp') and utmevent in ('general-email-log', 'spamfilter') group by recipient, sender order by requests desc)### union all ###(select `to` as recipient, `from` as sender, count(*) as requests, sum(coalesce(sentbyte, 0)+coalesce (rcvdbyte, 0)) as bandwidth from \$log-emailfilter where \$filter-exclude-var and service in ('pop3', 'POP3', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/tcp', 'pop3s', 'POP3s', '995/tcp') and eventtype is null group by `to`, `from` order by requests desc)###) t where \$filter-drilldown and recipient is not null group by recipient order by requests desc

```
Dataset Name
                                  Description
                                                                                  Log Category
                                  Drilldown top email receive sender by count
drilldown-Top-Email-Receive-Sender-
                                                                                  traffic
By-Count
select
 sender,
 sum(requests) as requests
from
    ###(select recipient, sender, count(*) as requests, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth from $log where $filter-exclude-var and (logflag&1>0) and
service in ('pop3', 'POP3', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS',
'993/tcp', 'pop3s', 'POP3S', '995/tcp') and utmevent in ('general-email-log', 'spamfilter')
group by recipient, sender order by requests desc)### union all ###(select `to` as
recipient, `from` as sender, count(*) as requests, sum(coalesce(sentbyte, 0)+coalesce
(rcvdbyte, 0)) as bandwidth from $loq-emailfilter where $filter-exclude-var and service in
('pop3', 'POP3', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/tcp', 'pop3s',
'POP3S', '995/tcp') and eventtype is null group by `to`, `from` order by requests desc)###)
```

t where \$filter-drilldown and sender is not null group by sender order by requests desc)###

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

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

###(select (CASE WHEN direction='incoming' THEN dstip ELSE srcip END) as source, (CASE
WHEN direction='incoming' THEN srcip ELSE dstip END) as victim, count(*) as totalnum from
\$log where \$filter-exclude-var group by source, victim order by totalnum desc)### t where
\$filter-drilldown and victim is not null group by victim order by totalnum desc

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

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

###(select (CASE WHEN direction='incoming' THEN dstip ELSE srcip END) as source, (CASE
WHEN direction='incoming' THEN srcip ELSE dstip END) as victim, count(*) as totalnum from
\$log where \$filter-exclude-var group by source, victim order by totalnum desc)### t where
\$filter-drilldown and source is not null group by source order by totalnum desc

Dataset Name	Description	Log Category
drilldown-Top-Attack-List	Drilldown top attack list	attack
<pre>select from_itime(itime) as time attack, source, victim from</pre>	estamp,	
source, (CASE WHEN directio	(CASE WHEN direction='incoming' THEN on='incoming' THEN srcip ELSE dstip END oy itime desc)### t where \$filter-drill) as victim from \$log where

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

case when virus like & #039;Riskware%' then 'Spyware' when virus like 'Adware%' then 'Adware' else 'Virus' end) as malware_type, sum(totalnum) as totalnum from ###(select virus, virusid_to_str(virusid, eventtype) as virusid_s, count(*) as totalnum from \$log where \$filter and (eventtype is null or logver>=502000000) and nullifna(virus) is not null group by virus, virusid_s /*SkipSTART*/order by totalnum desc/*SkipEND*/)### t group by virus, malware type order by totalnum desc

Dataset Name	Description	Log Category
drilldown-Virus-Detail	Drilldown virus detail	virus
select from itime(itime) as ti	mestamp	
virus,	meseump,	
user src,		
victim,		
hostname,		
recipient		
from		
###(select itime, virus	<pre>, coalesce(nullifna(`user`), ipstr((0))</pre>	CASE WHEN direction='incoming'
THEN dstip ELSE srcip END))) as user_src, (CASE WHEN direction	n='incoming' THEN srcip ELSE
dstip END) as victim, cas	t(' ' as char) as hostname, cast(' '	as char) as recipient from
	venttype is null or logver>=50200000	
null order by itime desc)### t where \$filter-drilldown order by timestamp desc		

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

Dataset NameDescriptionLog Categoryuser-drilldown-Top-Allowed-Web-
Sites-By-RequestsUser drilldown top allowed web sites by requestswebfilter

select

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

from

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

Dataset Name	Description	Log Category
user-drilldown-Top-Blocked-Web- Categories	User drilldown top blocked web categories	webfilter
(*) as requests from log where	`user`), ipstr(`srcip`)) as user_src, catd \$filter and catdesc is not null group by ## t where \$filter-drilldown and action='b	user_src, catdesc,
Dataset Name	Description	Log Category
Dataset Name user-drilldown-Top-Allowed-Web- Categories	Description User drilldown top allowed web categories	Log Category webfilter

Dataset Name	Description	Log Category
user-drilldown-Top-Attacks	User drilldown top attacks by name	attack
select attack,		

sum(attack_count) as attack_count

from

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

Dataset Name	Description	Log Category
user-drilldown-Top-Attacks-High- Severity	User drilldown top attacks high severity	attack

select

attack, sum(attack_count) as attack_count

from

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

Dataset Name	Description	Log Category
user-drilldown-Top-Virus-By-Name	User drilldown top virus	virus
(virusid, eventtype) as virusid_	s, count(*) as totalnum f: r_src, virus, virusid_s o:	s user_src, virus, virusid_to_str from \$log where \$filter and nullifna order by totalnum desc)### t where

Dataset Name	Description	Log Category
user-drilldown-Top-Virus-Receivers- Over-Email	User drilldown top virus receivers over email	virus
select receiver, sum(totalnum) as totalnum		
	<pre>`user`), ipstr(`srcip`)) as user_src, `to`</pre>	
	\$filter and subtype='infected' and (servi mtps', 'SMTPS', '465/tcp') or service in (· •

'110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/tcp', 'pop3s', 'POP3S',
'995/tcp')) and nullifna(virus) is not null group by user_src, receiver order by totalnum
desc)### t where \$filter-drilldown group by receiver order by totalnum desc

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

select

```
$hour_of_day(timestamp) as hourstamp,
sum(totalnum) as totalnum
```

from

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

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

```
select
```

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

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

Description	Log Category
Event usage CPU	event
timestamp, as decimal(6, 2) timestamp, devid, slot, sum as total_erate, sum(coalesce e) as last_seen, sum(coalesce sum(coalesce(disk, 0)) as t 0)) as total_cpu, max(coale ate, 0)+coalesce(orate, 0)) on, max(coalesce(totalsessi	event (coalesce(trate, 0)) as total_ e(orate, 0)) as total_orate, min e(mem, 0)) as total_mem, max total_disk, max(coalesce(disk, 0)) esce(cpu, 0)) as cpu_peak, max as lograte_peak, sum(coalesce ton, 0)) as session_peak, sum(cast as sent, sum(cast(coalesce(split)))
coalesce(split_part(bandwidt	<pre>t(coalesce(split_part(bandwidth, ch, '/', 2), '0') as integer)) as .esce(setuprate, 0)) as cps peak,</pre>
	Event usage CPU arstamp, as decimal(6, 2) timestamp, devid, slot, sum as total_erate, sum(coalesce sum(coalesce(disk, 0)) as t 0)) as total_cpu, max(coale ate, 0)+coalesce(orate, 0)) con, max(coalesce(totalsessi '/', 1), '0') as integer)) s integer)) as recv, max(cas coalesce(split_part(bandwidt

count(*) as count from \$log where \$filter and subtype='system' and action='perf-stats' group by timestamp, devid, slot order by total_mem desc)### t group by hourstamp order by hourstamp

Dataset Name	Description	Log Category
event-Usage-Memory	Event usage memory	event
<pre>select \$hour_of_day(timestamp) cast(sum(total_mem) / sum(co) as mem_avg_usage from</pre>	-	

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

Dataset Name	Description	Log Category
event-Usage-Sessions	Event usage sessions	event
<pre>) as sess_avg_usage from ###(select \$flex_timestam trate, sum(coalesce(erate, (itime) as first_seen, max((coalesce(mem, 0)) as mem_p as disk_peak, sum(coalesce((coalesce(trate, 0)+coalesce (totalsession, 0)) as total (coalesce(split_part(bandwi part(bandwidth, '/', 2), '0 '/', 1), '0') as integer)+c transmit_peak, sum(coalesce count(*) as count from \$log</pre>	<pre>as hourstamp, count) as decimal(10, 2) ap as timestamp, devid, slot, sum(counce 0)) as total_erate, sum(coalesce(ora- itime) as last_seen, sum(coalesce(ora- eeak, sum(coalesce(disk, 0)) as tota cpu, 0)) as total_cpu, max(coalesce ee(erate, 0)+coalesce(orate, 0)) as session, max(coalesce(totalsession, dth, '/', 1), '0') as integer)) as coast(coalesce(split_part(bandwidth, e(setuprate, 0)) as cps, max(coalesce where \$filter and subtype='system' order by total_mem desc)### t group 1</pre>	<pre>ate, 0)) as total_orate, min mem, 0)) as total_mem, max al_disk, max(coalesce(disk, 0)) (cpu, 0)) as cpu_peak, max lograte_peak, sum(coalesce 0)) as session_peak, sum(cast sent, sum(cast(coalesce(split_ coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as me(setuprate, 0)) as cps_peak, and action='perf-stats' group</pre>

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

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

from

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

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

```
sum(
   coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
 )> 0
order by
 bandwidth desc
```

Dataset Name	Description	Log Category
App-Risk-Top-User-Source-By- Sessions	Application risk top user source by session count	traffic
<pre>select srcip, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, count(*) as sessions from \$log where \$filter and (logflag&l>0) and srcip is not null group by srcip, user_src order by sessions desc</pre>		

Dataset Name

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

```
sum(crscore % 65536)> 0
order by
scores desc
```

Dataset Name

Description

App-Risk-Top-Devices-By-Reputation- Application risk reputation top devices by scores Scores

Log Category

traffic

```
select
 max(
    get devtype(srcswversion, osname, devtype)
  ) as devtype_new,
  coalesce(
    nullifna(`srcname`),
   nullifna(`srcmac`),
   ipstr(`srcip`)
  ) as dev src,
  sum(crscore % 65536) as scores
from
  $log
where
  $filter
  and (
   logflag&1>0
  )
  and crscore is not null
group by
  dev src
having
  sum(crscore % 65536)> 0
order by
  scores desc
```

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

select

appcat, sum(bandwidth) as bandwidth from

###(select appid, app, appcat, apprisk, sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_out, sum(bandwidth) as bandwidth, sum(sessions) as sessions from ###base(/*tag:rpt_ base_t_top_app*/select dvid, srcip, dstip, epid, euid, coalesce(nullifna(`user`), nullifna (`unauthuser`), ipstr(`srcip`)) as user_src, appid, app, appcat, apprisk, hostname, sum (coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, count(*) as sessions from \$log-traffic where \$filter and (logflag&(1|32)>0) and nullifna(app) is not null group by dvid, srcip, dstip, epid, euid, user_src, appid, app, appcat, apprisk, hostname order by sessions desc)base### t group by appid, app, appcat, apprisk /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t where \$filterdrilldown and nullifna(appcat) is not null group by appcat having sum(bandwidth)>0 order by bandwidth desc

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

```
appcat,
sum(bandwidth) as bandwidth
```

from

###(select appid, app, appcat, apprisk, sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_out, sum(bandwidth) as bandwidth, sum(sessions) as sessions from ###base(/*tag:rpt_ base_t_top_app*/select dvid, srcip, dstip, epid, euid, coalesce(nullifna(`user`), nullifna (`unauthuser`), ipstr(`srcip`)) as user_src, appid, app, appcat, apprisk, hostname, sum (coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, count(*) as sessions from \$log-traffic where \$filter and (logflag&(1|32)>0) and nullifna(app) is not null group by dvid, srcip, dstip, epid, euid, user_src, appid, app, appcat, apprisk, hostname order by sessions desc)base### t group by appid, app, appcat, apprisk /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t where \$filterdrilldown and nullifna(appcat) is not null group by appcat having sum(bandwidth)>0 order by bandwidth desc

Dataset Name	Description	Log Category
Top-20-Categories-By-Bandwidth	Webfilter categories by bandwidth usage	webfilter
a a la a t		

select

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

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

Dataset Name	Description	Log Category
App-Risk-Key-Applications-Crossing- The-Network	Application risk application activity	traffic

select

app_group, appcat, sum(bandwidth) as bandwidth, sum(sessions) as num_session from

###(select app_group_name(app) as app_group, appcat, service, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from \$log where \$filter and (logflag&(1|32)>0) and nullifna(app) is not null group by app_group, appcat, service order by bandwidth desc)### t group by app_group, appcat order by bandwidth desc

Dataset Name	Description	Log Category
App-Risk-Applications-Running-Over- HTTP	Application risk applications running over HTTP	traffic
sentbyte, 0)+coalesce(rcvddelta ccvdbyte, 0)) as traffic_in, su as sessions from \$log where \$fi group by app_group, appcat, ser) as app_group, appcat, service, sum(coalesc , rcvdbyte, 0)) as bandwidth, sum(coalesce(r m(coalesce(sentdelta, sentbyte, 0)) as traff lter and (logflag&(1 32)>0) and nullifna(app vice order by bandwidth desc)### t where ser HTTPS', 'http', 'https') group by app_group,	<pre>ccvddelta, fic_out, count(*) o) is not null cvice in</pre>
(1, 2, 3, 4, 3, 4, 4, 4, 5) > 0		_
<pre>sum(bandwidth)>0 order by bandw Dataset Name</pre>		Log Category
_	idth desc Description Application risk web browsing summary category	Log Category traffic
App-Risk-Top-Web-Sites-Visited-By- Network-Users-Pie-Cha select catdesc, sum(num_sess) as num_sess, sum(bandwidth) as bandwidth from ###(select catdesc, count(*) as bandwidth from \$log-traffic is null or logver<50200000) an word', 'web-content', 'command-	Description	traffic esce(rcvdbyte, eb>0 or ((logve ofilter', 'bann not null group 1

Dataset Name	Description	Log Category
App-Risk-Top-Web-Sites-Visited-By- Network-Users	Application risk web browsing summary category	traffic
as bandwidth from \$log-traffic w is null or logver<50200000) and word', 'web-content', 'command-b	is num_sess, sum(coalesce(sentbyte, 0)+coalesc where \$filter and (logflag&1>0) and (countweb> l (hostname is not null or utmevent in ('webfi wlock', 'script-filter'))) and catdesc is not ## t group by catdesc order by num_sess desc	0 or ((logver lter', 'banned-
Dataset Name	Description	Log Category
App-Risk-Web-Browsing-Hostname-	Application risk web browsing activity hostname category	webfilter

Category

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

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

Dataset Name	Description	Log Category
Top-Destination-Countries-By- Browsing-Time	Traffic top destination countries by browsing time	traffic
<pre>select dstcountry, ebtr_value(ebtr_agg_flat(browsetime), null, \$timespan) as browsetime, sum(bandwidth) as bandwidth, sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_out)</pre>	t	
<pre>bandwidth, sum(traffic_in) as tr dstcountry, ebtr_agg_flat(\$brows</pre>	<pre>g_flat(browsetime) as browsetime, sum(bandwidt affic_in, sum(traffic_out) as traffic_out from e_time) as browsetime, sum(coalesce(sentbyte, (coalesce(rcvdbyte, 0)) as traffic_in, sum(coa</pre>	(select 0)+coalesce

(sentbyte, 0)) as traffic_out from \$log where \$filter and (logflag&1>0) and \$browse_time is not null group by dstcountry) t group by dstcountry /*SkipSTART*/order by ebtr_value(ebtr_ agg_flat(browsetime), null, null) desc/*SkipEND*/)### t group by dstcountry order by browsetime desc

Dataset Name	Description	Log Category
App-Risk-Traffic-Top-Hostnames-By- Browsing-Time	Traffic top domains by browsing time	traffic
<pre>select hostname, ebtr_value(ebtr_agg_flat(browsetime), null, \$timespan) as browsetime, sum(bandwidth) as bandwidth,</pre>		
<pre>sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_ou</pre>		
<pre>sum(traffic_in) as traffic_in, s agg_flat(\$browse_time) as browse bandwidth, sum(coalesce(rcvdbyte</pre>	<pre>flat(browsetime) as browsetime, sum(bandwi um(traffic_out) as traffic_out from (selec time, sum(coalesce(sentbyte, 0)+coalesce(r , 0)) as traffic_in, sum(coalesce(sentbyte (logflag&1>0) and hostname is not null and</pre>	t hostname, ebtr_ cvdbyte, 0)) as , 0)) as traffic_

not null group by hostname) t group by hostname /*SkipSTART*/order by ebtr_value(ebtr_agg_ flat(browsetime), null, null) desc/*SkipEND*/)### t group by hostname order by browsetime desc

Dataset Name	Description	Log Category
App-Risk-Top-Threat-Vectors- Crossing-The-Network	Application risk top threat vectors	attack

```
select
```

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

###(select attack, severity, ref, count(*) as totalnum from \$log where \$filter and nullifna(attack) is not null group by attack, severity, ref order by totalnum desc)### t group by severity order by totalnum desc

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

where severity='critical' group by attack, severity, ref order by totalnum desc

Dataset Name	Description	Log Category
App-Risk-Top-High-Threat-Vectors- Crossing-The-Network	Application risk top high threat vectors	attack
nullifna(attack) is not null gr	ref, count(*) as totalnum from \$log who coup by attack, severity, ref order by t attack, severity, ref order by totalnum	totalnum desc)### t
Dataset Name	Description	Log Category

 App-Risk-Top-Medium-Threat-Vectors-Crossing-The-Network
 Application risk top medium threat vectors
 attack

 select attack,
 attack,
 attack
 attack

```
severity,
```

ref, sum(totalnum) as totalnum from

###(select attack, severity, ref, count(*) as totalnum from \$log where \$filter and nullifna(attack) is not null group by attack, severity, ref order by totalnum desc)### t where severity='medium' group by attack, severity, ref order by totalnum desc

Dataset Name	Description	Log Category
App-Risk-Top-Low-Threat-Vectors- Crossing-The-Network	Application risk top low threat vectors	attack
<pre>select attack, severity, ref, sum(totalnum) as totalnum</pre>		
from		

###(select attack, severity, ref, count(*) as totalnum from \$log where \$filter and nullifna(attack) is not null group by attack, severity, ref order by totalnum desc)### t where severity='low' group by attack, severity, ref order by totalnum desc

Dataset Name	Description	Log Category
App-Risk-Top-Info-Threat-Vectors- Crossing-The-Network	Application risk top info threat vectors	attack
select attack,		

severity,
ref,
sum(totalnum) as totalnum
from

###(select attack, severity, ref, count(*) as totalnum from \$log where \$filter and nullifna(attack) is not null group by attack, severity, ref order by totalnum desc)### t where severity='info' group by attack, severity, ref order by totalnum desc

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

select

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

case when virus like & #039;Riskware%' then 'Spyware' when virus like 'Adware%' then 'Adware' else 'Virus' end) as malware_type, sum(totalnum) as totalnum from ###(select virus, virusid_to_str(virusid, eventtype) as virusid_s, count(*) as totalnum from \$log where \$filter and (eventtype is null or logver>=502000000) and nullifna(virus) is not null group by virus, virusid_s /*SkipSTART*/order by totalnum desc/*SkipEND*/)### t group by virus, malware type order by totalnum desc

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

```
select
    user_src,
    sum(totalnum) as totalnum
from
    ###(select coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, eventtype, logver,
virus, count(*) as totalnum from $log where $filter group by user_src, eventtype, logver,
virus /*SkipSTART*/order by totalnum desc/*SkipEND*/)### t where (eventtype is null or
logver>=50200000) and nullifna(virus) is not null group by user_src order by totalnum desc
```

Dataset Name	Description	Log Category
App-Risk-Data-Loss-Prevention-Type- Events	Application risk DLP UTM event	dlp
210110		

select

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

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

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

Dataset Name	Description	Log Category
App-Risk-Malware-Discovered	Application risk virus discovered	virus
select dom,		
sum(totalnum) as totalnum		

```
from
```

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

Dataset Name	Description	Log Category
App-Risk-Breakdown-Of-Risk- Applications	Application risk breakdown of risk applications	traffic
select unnest(
string_to_array(behavior,		

& #039;,')) as d_behavior, count(*) as number from \$log t1 inner join app_mdata t2 on t1.appid=t2.id where \$filter and (logflag&1>0) group by d_behavior order by number desc

Dataset Name	Description	Log Category
App-Risk-Number-Of-Applications-By-	Application risk number of applications by risk behavior	traffic
Risk-Behavior		

Dataset Name	Description	Log Category
App-Risk-High-Risk-Application	Application risk high risk application	traffic
<pre>select risk as d_risk, behavior as d_behavior, t2.id, t2.name, t2.app_cat, t2.technology, sum(coalesce(sentbyte, 0)+ coale) as bandwidth, count(*) as sessions from \$log t1 inner join app_mdata t2 on t1. where \$filter and (logflag&1>0</pre>		
) and behavior is not null		
group by t2.id		

order by risk desc, sessions desc

Description	Log Category
Severe and high risk applications	traffic
<pre>num , apprisk, sum(bandwidth) as bandwidt _base_t_top_app*/select dvid, srcip, ifna(`unauthuser`), ipstr(`srcip`)) a coalesce(rcvddelta, rcvdbyte, 0)) as affic_out, sum(coalesce(sentdelta, se ndwidth, count(*) as sessions from \$1 ifna(app) is not null group by dvid, cat, apprisk, hostname order by sessi /*SkipSTART*/order by sessions desc,</pre>	<pre>dstip, epid, euid, as user_src, appid, app, traffic_in, sum(coalesce entbyte, 0)+coalesce .og-traffic where \$filter srcip, dstip, epid, .ons desc)base### t group</pre>
	Severe and high risk applications hum , apprisk, sum(bandwidth) as bandwidt _base_t_top_app*/select dvid, srcip, ifna(`unauthuser`), ipstr(`srcip`)) a coalesce(rcvddelta, rcvdbyte, 0)) as affic_out, sum(coalesce(sentdelta, se hdwidth, count(*) as sessions from \$1 ifna(app) is not null group by dvid,

desc/*SkipEND*/)### t where \$filter-drilldown and nullifna(appcat) is not null and apprisk
in ('critical', 'high') group by appcat order by total_num desc

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

select

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

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

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

select appcat,

```
count(distinct app) as total_num from
```

###(select appid, app, appcat, apprisk, sum(bandwidth) as bandwidth, sum(sessions) as sessions from ###base(/*tag:rpt_base_t_top_app*/select dvid, srcip, dstip, epid, euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce (sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce (rcvddelta, rcvdbyte, 0)) as bandwidth, count(*) as sessions from \$log-traffic where \$filter and (logflag&(1|32)>0) and nullifna(app) is not null group by dvid, srcip, dstip, epid, euid, user_src, appid, app, appcat, apprisk, hostname order by sessions desc)base### t group by appid, app, appcat, apprisk /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t where \$filter-drilldown and nullifna(appcat) is not null and apprisk in ('critical', 'high') group by appcat order by total_num desc

Dataset Name	Description	Log Category
Apprisk-Ctrl-Top-20-High-Risk- Application	Application risk high risk application	traffic
<pre>select risk as d_risk, count(distinct user_src) as id, name, app_cat, technology, sum(bandwidth) as bandwidth, sum(sessions) as sessions</pre>		
from		
<pre>user_src, action, utmaction, s count(*) as sessions from \$log</pre>	<pre>lifna(`user`), nullifna(`unauthuser`), i um(coalesce(sentbyte, 0)+coalesce(rcvdby where \$filter and (logflag&1>0) group b dwidth desc)### t1 inner join app mdata</pre>	te, 0)) as bandwidth, by app, user_src,

action, utmaction order by bandwidth desc)### t1 inner join app_mdata t2 on t1.app=t2.name where risk>='4' group by id, name, app_cat, technology, risk order by d_risk desc, sessions desc

Dataset Name	Description	Log Category
Apprisk-Ctrl-High-Risk-Application- Behavioral	Application Behavioral Characteristics	traffic
<pre>(appcat)='remote.access' then 't 'video/audio') then 'bandwidth-o lower(appcat)='proxy' then 'prox (/*tag:rpt_base_t_top_app*/select nullifna(`unauthuser`), ipstr(`s sum(coalesce(rcvddelta, rcvdbyte as traffic_out, sum(coalesce(sen bandwidth, count(*) as sessions nullifna(app) is not null group appcat, apprisk, hostname order 'remote.access', 'storage.backup ('critical', 'high') group by ap 'malicious' as behavior, count((logflag&16>0) and severity in ()</pre>	appcat)='botnet' then 'malicious' when unneling' when lower(appcat) in ('stora consuming' when lower(appcat)='p2p' then cy' end) as behavior, sum(sessions) as t t dvid, srcip, dstip, epid, euid, coale rcip`)) as user_src, appid, app, appcat , 0)) as traffic_in, sum(coalesce(sentd tdelta, sentbyte, 0)+coalesce(rcvddelta from \$log-traffic where \$filter and (lo by dvid, srcip, dstip, epid, euid, user by sessions desc)base### t where lower(', 'video/audio', 'p2p', 'proxy') and a pcat order by total_num desc)### union *) as total_num from \$log-attack where 'critical', 'high') group by behavior o lown group by behavior order by percenta	<pre>ge.backup', 'peer-to-peer' when otal_num from ###base sce(nullifna(`user`), , apprisk, hostname, elta, sentbyte, 0)) , rcvdbyte, 0)) as gflag&(1 32)>0) and _src, appid, app, appcat) in ('botnet', pprisk in all ###(select \$filter and rder by total_num</pre>

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

Dataset Name	Description	Log Category
Apprisk-Ctrl-Risk-Application-Usage- By-Category-With-Pie	Application risk application usage by category	traffic
select appcat, sum(bandwidth) as bandwidth		
traffic_out, sum(bandwidth) as b	apprisk, sum(traffic_in) as traffic_in, s pandwidth, sum(sessions) as sessions from a cip, dstip, epid, euid, coalesce(nullifna(###base(/*tag:rpt_
(coalesce(rcvddelta, rcvdbyte, (<pre>as user_src, appid, app, appcat, apprisk,))) as traffic_in, sum(coalesce(sentdelta, elta, sentbyte, 0)+coalesce(rcvddelta, rcvc</pre>	sentbyte, 0)) as

bandwidth, count(*) as sessions from \$log-traffic where \$filter and (logflag&(1|32)>0) and
nullifna(app) is not null group by dvid, srcip, dstip, epid, euid, user_src, appid, app,
appcat, apprisk, hostname order by sessions desc)base### t group by appid, app, appcat,
apprisk /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t where \$filterdrilldown and nullifna(appcat) is not null group by appcat having sum(bandwidth)>0 order by
bandwidth desc

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

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

from

###(select app, appcat, user_src, sum(bandwidth) as bandwidth, sum(sessions) as sessions from ###base(/*tag:rpt_base_t_top_app*/select dvid, srcip, dstip, epid, euid, coalesce (nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce (sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce (rcvddelta, rcvdbyte, 0)) as bandwidth, count(*) as sessions from \$log-traffic where \$filter and (logflag&(1|32)>0) and nullifna(app) is not null group by dvid, srcip, dstip, epid, euid, user_src, appid, app, appcat, apprisk, hostname order by sessions desc)base### t where nullifna(appcat) is not null group by appcat, user_src order by bandwidth desc)### t where \$filter-drilldown group by appcat order by bandwidth desc

Dataset Name	Description	Log Category
Apprisk-Ctrl-Top-Web-Applications-by- Bandwidth	Top 25 Web Categories by Bandwidtih	traffic
select risk as d_risk,		
id, name, technology,		

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

###(select appid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as f_
user, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, count(*) as num_session
from \$log where \$filter and (logflag&1>0) and nullifna(app) is not null and service in
('80/tcp', '443/tcp', 'HTTP', 'HTTPS', 'http', 'https') group by appid, f_user order by
bandwidth desc)### t1 inner join app_mdata t2 on t1.appid=t2.id group by d_risk, id, name,
technology order by bandwidth desc

Dataset Name	Description	Log Category
Apprisk-Ctrl-Top-Web-Categories- Visited	Top 25 Web Categories Visited	traffic
select catdesc,		

```
count(distinct f_user) as user_num,
sum(sessions) as sessions,
sum(bandwidth) as bandwidth
```

from

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

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

case when lower(appcat)=& #039;botnet' then 'Botnet C&C' else (case when virus_s like 'Riskware%' then 'Spyware' when virus_s like 'Adware%' then 'Adware' else 'Virus' end) end) as malware_type, appid, app, count(distinct dstip) as victims, count(distinct srcip) as source, sum(total_num) as total_num from (###(select app as virus_s, appcat, appid, app, dstip, srcip, count(*) as total_num from \$log-traffic where \$filter and (logflag&l>0) and lower(appcat)='botnet' group by virus_s, appcat, appid, dstip, srcip, app order by total_num desc)### union all ###(select unnest(string_to_array(virus, ',')) as virus_s, appcat, appid, app, dstip, srcip, count(*) as total_num from \$log-traffic where \$filter and (logflag&l>0) and virus is not null group by virus_s, appcat, appid, dstip, srcip, app order by total_num desc)### union all ###(select attack as virus_s, 'botnet' as appcat, 0 as appid, attack as app, dstip, srcip, count(*) as total_num from \$log-attack where \$filter and (logflag&l>0) group by virus_s, appcat, appid, dstip, srcip, app order by total_num desc)### union all ###(select attack as virus_s, 'botnet' as appcat, 0 as appid, attack as app, dstip, srcip, count(*) as total_num from \$log-attack where \$filter and (logflag&l6>0) group by virus_s, appcat, appid, dstip, srcip, app order by total_num desc)###) t group by virus, appid, app, malware type order by total num desc

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

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

Dataset Name	Description	Log Category
Apprisk-Ctrl-Files-Analyzed-By- FortiCloud-Sandbox	Files analyzed by FortiCloud Sandbox	virus
<pre>select \$DAY_OF_MONTH as dom, count(*) as total_num from \$log where \$filter and nullifna(filename) is not and logid to int(logid)= 9233</pre>	null	
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
END) as source, (CASE WHEN dir (*) as totalnum from \$log wher (logid)=9233 and analyticscksu	-	END) as victim, count ogid_to_int cscksum, source,
Dataset Name	Description	Log Category

Apprisk-Ctrl-File-Transferred-By- Application File transferred by applications on the network	app-ctrl

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

Dataset Name	Description	Log Category
appctrl-Top-Blocked-SCCP-Callers	Appctrl top blocked SCCP callers	app-ctrl

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

###(select srcname as caller, app, count(*) as totalnum from \$log where \$filter and srcname is not null and lower(appcat)='voip' and action='block' group by caller, app order by totalnum desc)### t where app='sccp' group by caller order by totalnum desc

Dataset Name	Description	Log Category
appctrl-Top-Blocked-SIP-Callers	Appctrl top blocked SIP callers	app-ctrl
srcname is not null and lower(ap	app, count(*) as totalnum from \$log opcat)='voip' and action='block' gro p='sip' group by caller order by tot	up by caller, app order
Dataset Name	Description	Log Category
	Description High risk application in use	

```
app_cat,
technology,
```

```
sum(bandwidth) as bandwidth,
sum(sessions) as sessions
from
  ###(select risk as d_risk, coalesce(nullifna(t1.`user`), nullifna(t1.`unauthuser`), ipstr
(t1.`srcip`)) as f_user, t2.name, t2.app_cat, t2.technology, sum(coalesce(sentbyte,
0)+coalesce(rcvdbyte, 0)) as bandwidth, count(*) as sessions from $log t1 inner join app_
mdata t2 on t1.appid=t2.id where $filter and risk>='4' and (logflag&1>0) group by f_user,
t2.name, t2.app_cat, t2.technology, risk)### t group by d_risk, name, app_cat, technology
order by d_risk desc, sessions desc
```

Dataset Name	Description	Log Category
security-High-Risk-Application-By- Category	High risk application by category	traffic
—	um log t1 inner join app_mdata t2 on t1.appid=t2 ag&1>0) group by app_cat, app)### t group by	

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

select

```
appcat,
sum(bandwidth) as bandwidth
```

from

###(select appid, app, appcat, apprisk, sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_out, sum(bandwidth) as bandwidth, sum(sessions) as sessions from ###base(/*tag:rpt_ base_t_top_app*/select dvid, srcip, dstip, epid, euid, coalesce(nullifna(`user`), nullifna (`unauthuser`), ipstr(`srcip`)) as user_src, appid, app, appcat, apprisk, hostname, sum (coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, count(*) as sessions from \$log-traffic where \$filter and (logflag&(1|32)>0) and nullifna(app) is not null group by dvid, srcip, dstip, epid, euid, user_src, appid, app, appcat, apprisk, hostname order by sessions desc)base### t group by appid, app, appcat, apprisk /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t where \$filterdrilldown and nullifna(appcat) is not null group by appcat having sum(bandwidth)>0 order by bandwidth desc

Dataset Name	Description	Log Category
Security-Category-Breakdown-By- Bandwidth	Category breakdown of all applications, sorted by bandwidth	traffic
<pre>select appcat, count(distinct app) as app_nu count(distinct user_src) as u sum(bandwidth) as bandwidth, sum(sessions) as num session</pre>		

from

###(select app, appcat, user_src, sum(bandwidth) as bandwidth, sum(sessions) as sessions from ###base(/*tag:rpt_base_t_top_app*/select dvid, srcip, dstip, epid, euid, coalesce (nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce (sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce (rcvddelta, rcvdbyte, 0)) as bandwidth, count(*) as sessions from \$log-traffic where \$filter and (logflag&(1|32)>0) and nullifna(app) is not null group by dvid, srcip, dstip, epid, euid, user_src, appid, app, appcat, apprisk, hostname order by sessions desc)base### t where nullifna(appcat) is not null group by app, appcat, user_src order by bandwidth desc)### t where \$filter-drilldown group by appcat order by bandwidth desc

Dataset Name	Description	Log Category
security-Top25-Web-Applications-By- Bandwidth	Top Web Applications by Bandwidtih	traffic
<pre>(t1.`user`), nullifna(t1.`unauth 0)+coalesce(rcvdbyte, 0)) as ban mdata t2 on t1.appid=t2.id where service in ('80/tcp', '443/tcp',</pre>	<pre>s, app_cat, t2.name, t2.technology, coa user`), ipstr(t1.`srcip`)) as f_user dwidth, count(*) as num_session from \$filter and (logflag&1>0) and nulli 'HTTP', 'HTTPS', 'http', 'https') g ser)### t group by d_risk, name, app</pre>	<pre>, sum(coalesce(sentbyte, \$log t1 inner join app_ fna(app) is not null and roup by risk, t2.app_</pre>
Dataset Name	Description	Log Category
Security-Top25-Web-Categories- Visited	Top 25 Web Categories Visited	traffic

select

```
catdesc,
count(distinct f_user) as user_num,
sum(sessions) as sessions,
sum(bandwidth) as bandwidth
```

from

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

Dataset Name	Description	Log Category
security-Top25-Malware-Virus-Botnet- Spyware	Malware: viruses, Bots, Spyware/Adware	traffic

```
select
virus_s as virus,
(
```

case when lower(appcat)=& #039;botnet' then 'Botnet C&C' else (case when virus_s like 'Riskware%' then 'Spyware' when virus_s like 'Adware%' then 'Adware' else 'Virus' end) end) as malware_type, count(distinct dstip) as victims, count(distinct srcip) as source, sum (total_num) as total_num from (###(select app as virus_s, appcat, dstip, srcip, count(*) as total_num from \$log-traffic where \$filter and (logflag&l>0) and lower(appcat)='botnet' group by virus_s, appcat, dstip, srcip order by total_num desc)### union all ###(select unnest (string_to_array(virus, ',')) as virus_s, appcat, dstip, srcip, count(*) as total_num from \$log-traffic where \$filter and (logflag&l>0) and virus is not null group by virus_s, appcat, dstip, srcip order by total_num desc)### union all ###(select attack as virus_s, 'null' as appcat, dstip, srcip, count(*) as total_num from \$log-attack where \$filter and (logflag&l6>0) group by virus_s, appcat, dstip, srcip order by total_num desc)###) t group by virus, malware type order by total num desc

Dataset Name	Description	Log Category
security-Top10-Malware-Virus- Spyware	Malware: viruses, Spyware/Adware	virus
<pre>select virus, max(virusid_s) as virusid, malware_type, count(distinct victim) as vict count(distinct source) as sour sum(total num) as total num</pre>		
direction='incoming' THEN dstip THEN srcip ELSE dstip END) as vi virus like 'Adware%' then 'Adware	str(virusid, eventtype) as virusid_s, (CASE ELSE srcip END) as source, (CASE WHEN dire ctim, (case when virus like 'Riskware%' th ce' else 'Virus' end) as malware_type, cou ifna(virus) is not null group by virus, vi	ection='incoming' hen 'Spyware' when unt(*) as total_num

Dataset Name	Description	Log Category
security-Top10-Malware-Botnet	Malware: Botnet	appctrl
select		
app,		
appid,		
malware type,		
count(distinct victim) as v	ictims,	
count(distinct source) as s	ource,	
<pre>sum(total_num) as total_num</pre>		
from		
(
###(select app, appid, ca	st('Botnet C&C' as char(32)) as m	nalware_type,(CASE WHEN
direction='incoming' THEN dst	ip ELSE srcip END) as source, (CA	ASE WHEN direction='incoming'
THEN ergin FIGE detin FND) as	wighting = count(*) as total num fu	com ślog-app-strl whore śfilte

victim order by total num desc)### t group by virus, malware type order by total num desc

direction='incoming' THEN dstip ELSE srcip END) as source, (CASE WHEN direction='incoming' THEN srcip ELSE dstip END) as victim, count(*) as total_num from \$log-app-ctrl where \$filter and lower(appcat)='botnet' and nullifna(app) is not null group by app, appid, malware_type, source, victim order by total_num desc)### union all ###(select attack, 0 as appid, cast ('Botnet C&C' as char(32)) as malware_type, (CASE WHEN direction='incoming' THEN dstip ELSE srcip END) as source, (CASE WHEN direction='incoming' THEN srcip ELSE dstip END) as victim,

count(*) as total_num from \$log-attack where \$filter and (logflag&16>0) group by attack, appid, malware_type, source, victim order by total_num desc)###) t group by app, appid, malware_type order by total_num desc

Dataset Name	Description	Log Category
security-Top10-Victims-of-Malware	Victims of Malware	virus
<pre>select coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, virus as malware, count(*) as total_num from \$log where \$filter and virus is not null group by user_src, malware order by total_num desc</pre>		
Dataset Name	Description	Log Category
security-Top10-Victims-of-Phishing- Site	Victims of Phishing Site	webfilter
num from \$log where \$filter and	<pre>// hostname url) as phishing_site, l lower(service) in ('http', 'https') and ser_src, phishing_site order by total_num</pre>	l hostname is not nul:
Dataset Name	Description	Log Category
security-Top25-Malicious-Phishing- Sites	Malicious Phishing Site	webfilter
select phishing_site, count(distinct dstip) as vict count(distinct srcip) as sour		

from
 ###(select (lower(service) || '://' || hostname || url) as phishing_site, dstip, srcip,
count(*) as total from \$log where \$filter and lower(service) in ('http', 'https') and

sum(total) as total_num

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 (、 · · · · ·
(distinct (CASE WHEN direction=	39; incoming' THEN srcip ELSE dstip END) 'incoming' THEN dstip ELSE srcip END))	as sources, sum
5 when severity='high' then 4 w	(select attack, attackid, (case when se hen severity='medium' then 3 when sever nd) as severity number, direction, dsti	ity='low' then 2 when
totalnum from \$log where \$filte	r and nullifna(attack) is not null and rity, direction, dstip, srcip order by	severity is not null
<pre>left join (select name, cve, vu attack, attackid, vuln_type, se</pre>	<pre>ln_type from ips_mdata) t2 on t1.attack verity_number, cve order by severity_nu</pre>	=t2.name group by
desc		

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 \$filter and nullifna(filename) is not and logid_to_int(logid) = 9233 group by dow order by dow</pre>	null	
Dataset Name	Description	Log Category
Security-Zero-Day-Detected-On- Network	Zero-day malware detected on the network	traffic
select		

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

from

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

Dataset Name	Description	Log Category
security-Data-Loss-Incidents-By- Severity	Data loss incidents summary by severity	dlp
<pre>subtype, srcip, dstip, severity, severity='critical' then 'Critica (`user`), ipstr(`srcip`)) is not as data_loss from \$log where \$fi</pre>	_severity, om` as sender, `to` as receiver, profile, act: filename, direction, filesize, (case when al Data Exfiltration' else (case when coalesce null then 'User Associated Data Loss' else Ni lter /*SkipSTART*/order by itime desc/*SkipENi s not null group by s_severity order by total	e(nullifna JLL end) end) D*/)### t where

Dataset Name	Description	Log Category
security-Data-Loss-Files-By-Service	Data Lass Files By Service	dlp

select

filename,

(

case direction when & #039;incoming' then 'Download' when 'outgoing' then 'Upload' end) as action, max(filesize) as filesize, service from ###(select itime, hostname,`from` as sender, `to` as receiver, profile, action, service, subtype, srcip, dstip, severity, filename, direction, filesize, (case when severity='critical' then 'Critical Data Exfiltration' else (case when coalesce(nullifna(`user`), ipstr(`srcip`)) is not null then 'User Associated Data Loss' else NULL end) end) as data_loss from \$log where \$filter /*SkipSTART*/order by itime desc/*SkipEND*/)### t where \$filter-drilldown and filesize is not null group by filename, direction, service order by filesize desc

Dataset Name	Description	Log Category
security-Endpoint-Security-Events- Summary	Endpoint Security Events summary	fct-traffic

select

(

case utmevent when & #039;antivirus' then 'Malware incidents' when 'webfilter' then 'Malicious/phishing websites' when 'appfirewall' then 'Risk applications' when 'dlp' then 'Data loss incidents' when 'netscan' then 'Vulnerability detected' else 'Others' end) as events, count(*) as total_num from \$log where \$filter and utmevent is not null group by events order by total_num desc

Dataset Name	Description	Log Category
security-Top-Endpoing-Running-High- Risk-Application	Endpoints Running High Risk Application	fct-traffic

select coalesce(nullifna(`user`), ipstr(`srcip`), & #039;Unknown') as f_user, coalesce(nullifna(hostname), 'Unknown') as host_name, threat as app, t2.app_cat as appcat, risk as d_risk from \$log t1 inner join app_mdata t2 on t1.threat=t2.name where \$filter and utmevent='appfirewall' and risk>='4' group by f_user, host name, t1.threat, t2.app_cat, t2.risk order by risk desc

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

& #039;Unknown') as f_user, coalesce(nullifna(hostname), 'Unknown') as host_name, virus, file from \$log where \$filter and subtype='av' and virus is not null group by f_user, host_name, virus, file

Dataset Name	Description	Log Category
security-Top-Endpoints-With-Web- Violateions	Endpoints With Web Violations	fct-traffic
<pre>(hostname), 'Unknown') as host_n \$filter and utmevent='webfilter'</pre>	user`), ipstr(`srcip`)) as f_user, coale ame, remotename, count(*) as total_num f and remotename is not null and utmactic	rom \$log where m='blocked' group by
<pre>f_user, host_name, remotename or remotename order by total_num de</pre>	<pre>der by total_num desc)### t group by f_u sc</pre>	iser, host_name,

Dataset Name	Description	Log Category
security-Top-Endpoints-With-Data- Loss-Incidents	Endpoints With Data Loss Incidents	fct-event
<pre>select f_user, host_name, msg, sum(total num) as total num</pre>		
<pre>from ###(select coalesce(nullifna((nullifna(hostname), 'Unknown')</pre>	<pre>`user`), ipstr(`deviceip`), 'Unknown') a as host_name, msg, count(*) as total_nu by f_user, host_name, msg order by tota order by total num desc</pre>	ım from \$log where

Dataset Name	Description	Log Category
content-Count-Total-SCCP-Call- Registrations-by-Hour-of-Day	Content count total SCCP call registrations by hour of day	content
select		
hourstamp, count(totalnum) as totalnum		
from		
	rstamp, proto, kind, status, sum(duration) as content where \$filter group by hourstamp, proto	

order by totalnum desc)### t where proto='sccp' and kind='register' group by hourstamp order by hourstamp

Dataset Name	Description	Log Category
content-Count-Total-SCCP-Calls- Duration-by-Hour-of-Day	Content count total SCCP calls duration by hour of day	content

select

```
hourstamp,
sum(sccp_usage) as sccp_usage
from
```

###(select \$hour_of_day as hourstamp, proto, kind, status, sum(duration) as sccp_usage, count(*) as totalnum from \$log-content where \$filter group by hourstamp, proto, kind, status order by totalnum desc)### t where 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
count(*) as totalnum from \$log	ourstamp, proto, kind, status, sum(duration) as -content where \$filter group by hourstamp, prot here proto='sccp' and kind='call-info' group by	o, kind, status
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
count(*) as totalnum from \$log	ourstamp, proto, kind, status, sum(duration) as -content where \$filter group by hourstamp, prot here proto='sip' and kind='register' group by h	o, kind, status

```
by hourstamp
```

Dataset Name	Description	Log Category
content-Count-Total-SIP-Calls-per- Status	Content count total SIP calls per status	content
count(*) as totalnum from \$log-c	urstamp, proto, kind, status, sum(durati content where \$filter group by hourstamp ere proto='sip' and kind='call' group by	, proto, kind, status

Dataset Name	Description	Log Category
content-Dist-Total-SIP-Calls-by- Duration	Content dist total SIP calls by duration	content

select

(

case when duration<60 then & #039;LESS_ONE_MIN' when duration < 600 then 'LESS_TEN_MIN' when duration < 3600 then 'LESS_ONE_HOUR' when duration >= 3600 then 'MORE_ONE_HOUR' else 'unknown' end) as f_duration, count(*) as totalnum from \$log where \$filter and proto='sip' and kind='call' and status='end' group by f_duration order by totalnum desc

Dataset Name	Description	Log Category
Botnet-Activity-By-Sources	Botnet activity by sources	traffic
<pre>select app, user_src, sum(events) as events</pre>		
from (
select app,		
user_src, sum(totalnum) as events from		
###(select app, appcat,	apprisk, srcip, dstip, coalesce(nu	

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

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

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

Dataset Name	Description	Log Category
Detected-Botnet	Detected botnet	traffic

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

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

Dataset Name	Description	Log Category
Botnet-Sources	Botnet sources	traffic
<pre>select dstip, domain, sum(events) as events from ((</pre>		

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

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

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

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

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

(

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

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

select

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

from

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

Dataset Name	Description	Log Category
Application-Usage-List	Detailed application usage	traffic
<pre>select appid, app, appcat, (case when (utmaction in (</pre>		
custaction, sum(coalesce(sentby	') or action='deny') then 'Blocked' else 'A te, 0)+coalesce(rcvdbyte, 0)) as bandwidth, and (logflag&1>0) and nullifna(app) is not	count(*) as num_

Dataset Name	Description	Log Category	
PCI-DSS-Compliance-Summary	PCI DSS Compliance Summary	event	
<pre>select status, num_reason as requirements, cast(num_reason * 100.0 /(sum(num_reason) over()) as decimal(18, 2)) as percent</pre>			
from (
select			
<pre>(case when fail_count>0 then & #039;Non-Compliant' else 'Compliant' end) as status, count(distinct reason) as num_reason from (select ftnt_pci_id, (sum(fail_count) over (partition by ftnt_pci_id)) as fail_count, reason from ###(select ftnt_pci_id, (case when result='fail' then 1 else 0 end) as fail_count, reason from \$log t1 inner join pci_dss_mdat t2 on t1.reason=t2.ftnt_id where \$filter and subtype='compliance-check' group by ftnt_pci_ id, result, reason)### t) t group by status) t order by status</pre>			

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 order by fail_count desc)### 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
) as fail_count, reason from ###(select ftnt_pci_id, fail_count, reason from \$log t1 \$filter and subtype='compliance- order by fail_count desc)### t) t.reason) as requirements from reason, (case lower(severity) wh	<pre>partition by ftnt_pci_id) t2.severity, (case when result='fail' then 1 inner join pci_dss_mdata t2 on t1.reason=t2.f ccheck' group by ftnt_pci_id, t2.severity, res t where fail_count=0) select t.severity, cour select distinct on (1) reason, severity from en 'high' then 4 when 'critical' then 3 when esc) t group by t.severity order by requiremen</pre>	<pre>tnt_id where sult, reason nt(distinct query order by 'medium' then 2</pre>
Dataset Name	Description	Log Category
PCI-DSS-Fortinet-Security-Best-	PCI DSS Fortinet Security Best Practice Summary	event

```
Practice-Summary
```

```
select
status,
num_reason as practices,
cast(
```

```
num_reason * 100.0 /(
    sum(num_reason) over()
    ) as decimal(18, 2)
    ) as percent
from
  (
    select
    (
```

case when result =& #039;fail' then 'Failed' else 'Passed' end) as status, count (distinct reason) as num_reason from ###(select result, reason from \$log where \$filter and subtype='compliance-check' and result in ('fail','pass') group by result, reason)### t group by status) t order by status desc

Dataset Name	Description	Log Category
PCI-DSS-Failed-Fortinet-Security- Best-Practices-By-Severity	PCI DSS Failed Fortinet Security Best Practices by Severity	event
<pre>select status, num_reason as practices, cast(num_reason * 100.0 /(sum(num_reason) over()) as decimal(18, 2)) as percent</pre>		
<pre>) as percent from (select initcap(status) as status, count(distinct reason) as num_reason from ###(select status, reason, result from \$log where \$filter and subtype='compliance- check' group by status, reason, result)### t where result='fail' group by status) t order by status</pre>		

Dataset Name	Description	Log Category
PCI-DSS-Passed-Fortinet-Security- Best-Practices-By-Severity	PCI DSS Passed Fortinet Security Best Practices by Severity	event
<pre>select status, num_reason as practices, cast(num_reason * 100.0 /(sum(num_reason) over()) as decimal(18, 2)) as percent from (select initcap(status) as status, count(distinct reason) as n from ###(select status, reason,</pre>	num_reason result from \$log where \$filter and subtype='c	ompliance-

check' group by status, reason, result)### t where result='pass' group by status) t order by status

Dataset Name	Description	Log Category
PCI-DSS-Requirements-Compliance- Details	PCI DSS Requirements Compliance Details	event
'Compliant' end) as compliance, when result='fail' then 1 else 0 pci_dss_mdata t2 on t1.reason=t2	<pre>ce, (case when sum(fail_count)>0 then 'Non-Co pci_requirement from ###(select ftnt_pci_id, end) as fail_count, pci_requirement from \$10 .ftnt_id where \$filter and subtype='compliand pci_requirement)### t group by ftnt_pci_id,</pre>	ftnt_id, (case og t1 inner join ce-check' group

Dataset Name	Description	Log Category
PCI-DSS-Fortinet-Security-Best- Practice-Details	PCI DSS Fortinet Security Best Practice Details	event
<pre>select reason as ftnt_id, msg, initcap(status) as status, module from \$log where \$filter and subtype =& #039;compliance id</pre>	-check' group by reason, status, module, msg	order by ftnt_

Dataset Name	Description	Log Category
DLP-Email-Activity-Details	Email DLP Violations Summary	dlp
itime, hostname, `from` as sender srcip, dstip, severity, filename 'Critical Data Exfiltration' els not null then 'User Associated D \$filter /*SkipSTART*/order by it (service in ('smtp', 'SMTP', '25)	<pre>, filesize, profile, action, direc , `to` as receiver, profile, actio , direction, filesize, (case when e (case when coalesce(nullifna(`us ata Loss' else NULL end) end) as d ime desc/*SkipEND*/)### t where \$f /tcp', '587/tcp', 'smtps', 'SMTPS' p', 'IMAP', '143/tcp', 'imaps', 'I</pre>	<pre>n, service, subtype, severity='critical' then er`), ipstr(`srcip`)) is ata_loss from \$log where ilter-drilldown and , '465/tcp') or service in</pre>

Dataset Name	Description	Log Category
Email-DLP-Chart	Email DLP Activity Summary	dlp
select		

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

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

Dataset Name	Description	Log Category
DLP-Web-Activity-Details	Web DLP Violations Summary	dlp
select		
from_itime(itime) as times	tamp,	
srcip,		
dstip, hostname,		
profile,		
filename,		
filesize,		
action,		
direction		
from		
	,`from` as sender, `to` as receiver, pro	
	ity, filename, direction, filesize, (cas	
severity='critical' then 'Cr	itical Data Exfiltration' else (case whe	n coalesce(nullifna
(`user`), ipstr(`srcip`)) is	not null then 'User Associated Data Los	s' else NULL end) end)
as data_loss from \$log where	\$filter /*SkipSTART*/order by itime des	c/*SkipEND*/)### t where
\$filter-drilldown and lower(service) in ('http', 'https') order by t	imestamp desc

Dataset Name	Description	Log Category
Web-DLP-Chart	Web DLP Activity Summary	dlp
<pre>select profile,</pre>		
	e,`from` as sender, `to` as receiver,	
<pre>subtype, srcip, dstip, severity, filename, direction, filesize, (case when severity='critical' then 'Critical Data Exfiltration' else (case when coalesce(nullifna (`user`), ipstr(`srcip`)) is not null then 'User Associated Data Loss' else NULL end) end)</pre>		
	e \$filter /*SkipSTART*/order by itime (service) in ('http', 'https') group b	

Dataset Name	Description	Log Category
DLP-FTP-Activity-Details	Web DLP Violations Summary	dlp
subtype, srcip, dstip, sev severity='critical' then ' (`user`), ipstr(`srcip`)) as data_loss from \$log whe	me,`from` as sender, `to` as receiver, pro rerity, filename, direction, filesize, (cas Critical Data Exfiltration' else (case whe is not null then 'User Associated Data Los pre \$filter /*SkipSTART*/order by itime des er(service) in ('ftp', 'ftps') order by tim	se when en coalesce(nullifna ss' else NULL end) end) sc/*SkipEND*/)### t where
Dataset Name	Description	Log Category
FTP-DLP-Chart	FTP DLP Activity Summary	dlp
select profile,		

as data_loss from \$log where \$filter	/*SkipSTART*/order by itime	<pre>desc/*SkipEND*/)### t where</pre>
<pre>\$filter-drilldown and lower(service)</pre>	in ('ftp', 'ftps') group by	profile order by total_num
desc		

Dataset Name	Description	Log Category
top-users-by-browsetime	Top Users by website browsetime	traffic
<pre>select user_src, domain, ebtr_value(ebtr_agg_flat(browsetime), null, \$timespan) as browsetime from ###(select user_src, domain,</pre>	<pre>ebtr_agg_flat(browsetime) as browsetime `)) as user src, coalesce(nullifna(host)</pre>	from (select coalesce
as domain, ebtr_agg_flat(\$browse_time) as browsetime from \$log where \$filter and \$browse_ time is not null group by user_src, domain) t group by user_src, domain order by ebtr_value (ebtr_agg_flat(browsetime), null, null) desc)### t group by user_src, domain order by browsetime desc		

Dataset Name	Description	Log Category
wifi-usage-by-hour-authenticated	Wifi Usage by Hour - Authenticated	event

select

hod,

count(distinct stamac) as totalnum

from

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

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

select

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

from

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

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

```
select
 app,
 coalesce(
  nullifna(`user`),
  nullifna(`unauthuser`),
   ipstr(`srcip`)
 ) as user src,
 count(*) as sessions
from
  $log
where
 $filter
 and (
  logflag&1>0
 )
 and nullifna(app) is not null
group by
 app,
 user_src
order by
 sessions desc
```

Dataset Name	Description	Log Category
traffic-Interface-Bandwidth-Usage	Interface Bandwidth Usage	traffic
as total_sent, sum(coalesce(rcvc (rcvdbyte, 0)) as total from \$10 not null and nullifna(dstintf) ; sum(coalesce(sentbyte, 0)+coalesce unnest(array['download', 'upload bandwidth from (select coalesce as devid, coalesce(t1.vd_s, t2.v (coalesce(t1.total_sent, 0)+coal sent, 0)+coalesce(t1.total_rcvd,	<pre>dom, devid, vd, srcintf, dstintf, s dbyte, 0)) as total_rcvd, sum(coales og where \$filter and (logflag&1>0) a s not null group by dom, devid, vd sce(rcvdbyte, 0))>0 order by total o d']) as type, unnest(array[sum(down (t1.dom_s, t2.dom_s) as dom, coales rd_s) as vd, coalesce(t1.srcintf, t) esce(t2.total_rcvd, 0)) as download 0)) as upload from qry t1 full jo cintf group by dom, devid, vd, intf dom</pre>	<pre>sce(sentbyte, 0)+coalesce and nullifna(srcintf) is , srcintf, dstintf having desc)### t) select dom, load), sum(upload)]) as ce(t1.devid_s, t2.devid_s) 2.dstintf) as intf, sum d, sum(coalesce(t2.total_ in qry t2 on t1.dom_</pre>
Dataset Name	Description	Log Category

Dataset Name	Description	Log Category
ctap-SB-Files-Needing-Inspection-vs- Others	Files Needing Inspection vs Others	virus
select (

```
case when suffix in (
```

#039;bat','cmd','exe','jar','msi','vbs','7z','zip','gzip','lzw','tar','rar','cab','doc','doc x','xls','xlsx','ppt','pptx','pdf','swf','lnk','js') then 'Higher Risk File Types' else 'Excluded Files' end) as files, sum(total_num) as total_num from ###(select file_name_ext (filename) as suffix, count(*) as total_num from \$log where \$filter and dtype='fortisandbox' and nullifna(filename) is not null group by suffix order by total_num desc)### t group by files order by total num desc

Dataset Name	Description	Log Category
ctap-SB-Breakdown-of-File-Types	Breakdown of File Types	virus

```
select
```

```
( case when suffix in (
```

& #039;exe','msi','upx','vbs','bat','cmd','dll','psl','jar') then 'Executable Files' when suffix in ('pdf') then 'Adobe PDF' when suffix in ('swf') then 'Adobe Flash' when suffix in ('doc','docx','rtf','dotx','docm','dotm','dot') then 'Microsoft Word' when suffix in ('xls','xlsx','xltx','xlsm','xlsb','xlam','xlt') then 'Microsoft Excel' when suffix in ('ppsx','ppt','pptx','potx','sldx','pptm','ppsm','potm','ppam','sldm','pps','pot') then 'Microsoft PowerPoint' when suffix in ('msg') then 'Microsoft Outlook' when suffix in ('htm','js','url','lnk') then 'Web Files' when suffix in

('cab','tgz','z','7z','tar','lzh','kgb','rar','zip','gz','xz','bz2') then 'Archive Files'
when suffix in ('apk') then 'Android Files' else 'Others' end) as filetype, sum(total_num)
as total_num from ###(select file_name_ext(filename) as suffix, count(*) as total_num from
\$log where \$filter and dtype='fortisandbox' and nullifna(filename) is not null group by
suffix order by total_num desc)### t group by filetype order by total_num desc

Dataset Name	Description	Log Category
ctap-SB-Top-Sandbox-Malicious-Exes		virus

select (

case fsaverdict when & #039;malicious' then 5 when 'high risk' then 4 when 'medium risk' then 3 when 'low risk' then 2 else 1 end) as risk, filename, service, count(*) as total_num from \$log where \$filter and dtype='fortisandbox' and file_name_ext(filename)='exe' and fsaverdict not in ('clean','submission failed') group by filename, risk, service order by risk desc, total_num desc, filename

Dataset Name	Description	Log Category
ctap-SB-Sources-of-Sandbox- Discovered-Malware	Sources of Sandbox Discovered Malware	virus
<pre>select source, sum(total_num) as total_num</pre>		
from		
select (
	#039;incoming' THEN dstip ELSE srcip EN ilter and dtype='fortisandbox' and null:	

null and fsaverdict not in ('clean','submission failed') group by source) t group by source
order by total_num desc

Dataset Name	Description	Log Category
ctap-apprisk-ctrl-High-Risk-Application	Application risk high risk application	traffic
select		
risk as d risk,		
count(distinct user src) as us	ers,	
id, —		
name,		
app_cat,		
technology,		
sum(bandwidth) as bandwidth,		
sum(sessions) as sessions		
from		
	<pre>fna(`user`), nullifna(`unauthuser`), ip</pre>	=
	<pre>(coalesce(sentbyte, 0)+coalesce(rcvdbyt</pre>	
action, utmaction order by bandw	here \$filter and (logflag&1>0) group by idth desc)### t1 inner join app_mdata t	2 on t1.app=t2.name

action, utmaction order by bandwidth desc)### t1 inner join app_mdata t2 on t1.app=t2.name where risk>='4' group by id, name, app_cat, technology, risk order by d_risk desc, sessions desc

Dataset Name	Description	Log Category
ctap-apprisk-ctrl-Application- Vulnerability	Application vulnerabilities discovered	attack
<pre>(distinct (CASE WHEN direction (totalnum) as totalnum from ## 5 when severity='high' then 4 severity='info' then 1 else 0 totalnum from \$log where \$filt group by attack, attackid, sev left join (select name, cve, v</pre>	039; incoming' THEN srcip ELSE dstip END) ='incoming' THEN dstip ELSE srcip END)) #(select attack, attackid, (case when se when severity='medium' then 3 when sever end) as severity_number, direction, dsti er and nullifna(attack) is not null and erity, direction, dstip, srcip order by uln_type from ips_mdata) t2 on t1.attack everity_number, cve order by severity_nu	as sources, sum verity='critical' ther ity='low' then 2 when p, srcip, count(*) as severity is not null totalnum desc)### t1 =t2.name group by

Dataset Name	Description	Log Category
ctap-apprisk-ctrl-Common-Virus- Botnet-Spyware	Common Virus Botnet Spyware	app-ctrl
select malware as virus, (

case when lower(appcat) = & #039; botnet' then 'Botnet C&C' else (case when malware like 'Riskware%' then 'Spyware' when malware like 'Adware%' then 'Adware' else 'Virus' end) end) as malware type, appid, app, count(distinct victim) as victims, count(distinct source) as source, sum(total num) as total num from (###(select app as malware, appcat, appid, app, (CASE WHEN direction='incoming' THEN dstip ELSE srcip END) as source, (CASE WHEN direction='incoming' THEN srcip ELSE dstip END) as victim, count(*) as total num from \$logapp-ctrl where \$filter and lower(appcat)='botnet' group by malware, appcat, appid, app, victim, source, app order by total num desc)### union all ###(select virus as malware, 'null' as appcat, 0 as appid, service as app, (CASE WHEN direction='incoming' THEN dstip ELSE srcip END) as source, (CASE WHEN direction='incoming' THEN srcip ELSE dstip END) as victim, count(*) as total num from \$log-virus where \$filter and virus is not null group by malware, appcat, app, appid, victim, source order by total num desc)### union all ###(select attack as malware, 'null' as appcat, 0 as appid, service as app, (CASE WHEN direction='incoming' THEN dstip ELSE srcip END) as source, (CASE WHEN direction='incoming' THEN srcip ELSE dstip END) as victim, count(*) as total num from \$log-attack where \$filter and (logflag&16>0) group by malware, appcat, app, appid, victim, source order by total num desc) ###) t group by malware, malware type, app, appid order by total num desc

Dataset Name	Description	Log Category
ctap-App-Risk-Reputation-Top- Devices-By-Scores	Reputation Top Devices By-Scores	traffic
<pre>select coalesce(nullifna(`srcname`), ipstr(`srcip`), nullifna(`srcmac`)) as dev_src, sum(crscore % 65536) as scores from \$log where \$filter and (logflag&1>0) and crscore is not null group by dev_src having sum(crscore % 65536)> 0 order by scores desc</pre>		
Dataset Name	Description	Log Category
ctap-HTTP-SSL-Traffic-Ratio	HTTP SSL Traffic Ratio	traffic
	ttp') then 'HTTP' else 'HTTPS' end) as servio rcvdbyte, 0)) as bandwidth from \$log where \$f:	

(logflag&1>0) and nullifna(app) is not null and service in ('80/tcp', '443/tcp', 'HTTP',

'HTTPS', 'http', 'https') group by service having sum(coalesce(sentbyte, 0)+coalesce (rcvdbyte, 0))>0 order by bandwidth desc

Dataset Name	Description	Log Category
ctap-Top-Source-Countries	Top Source Countries	traffic
<pre>select srccountry, sum(coalesce(sentbyte, 0)+ coa) as bandwidth from \$log where \$filter and (logflag&1>0) and nullifna(srccountry) is and srccountry <> & #039;Res 0)+coalesce(rcvdbyte, 0))>0 or</pre>	not null erved' group by srccountry hav	-

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

select

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

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

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

select

app_group, sum(bandwidth) as bandwidth from

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

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

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

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

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

select

name as app_group, sum(bandwidth) as bandwidth

from

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

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

select

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

from

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

Dataset Name	Description	Log Category
ctap-Top-Streaming-App-By- Bandwidth	Top Streaming applications by bandwidth usage	traffic
<pre>select app_group, sum(bandwidth) as bandwidth, sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_ou sum(sessions) as sessions</pre>		

from

###(select app_group_name(app) as app_group, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out, count(*) as sessions from \$log where \$filter and (logflag&l>0) and nullifna (app) is not null group by app_group having sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by bandwidth desc)### t1 inner join app_mdata t2 on lower(t1.app_group)=lower (t2.name) where app_cat='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>0)) as bandwidth, sum(coalesce(rd traffic_out, count(*) as sessions (app) is not null group by app_gr 0))>0 order by bandwidth desc)###</pre>	as app_group, sum(coalesce(sentbyte, 0)+coales cvdbyte, 0)) as traffic_in, sum(coalesce(sentby s from \$log where \$filter and (logflag&1>0) and coup having sum(coalesce(sentbyte, 0)+coalesce t 1 inner join app_mdata t2 on lower(t1.app_g: coup by app_group order by bandwidth desc	yte, 0)) as d nullifna (rcvdbyte,

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

select

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

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

Dataset Name	Description	Log Category
ctap-apprisk-ctrl-Top-Web-Categories- Visited	Top 25 Web Categories Visited	traffic
<pre>select catdesc, count(distinct f_user) as user sum(sessions) as sessions, sum(bandwidth) as bandwidth from ###(select catdesc, coalesce(number)</pre>	_num, ullifna(`user`), nullifna(`unauthuser`), ipstu	c(`srcip`)) as

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

Dataset Name	Description	Log Category
ctap-App-Risk-Applications-Running- Over-HTTP	Application risk applications running over HTTP	traffic
<pre>sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum as sessions from \$log where \$fil group by app_group, appcat, serv</pre>	as app_group, appcat, service, sum(coalesce(s rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvd (coalesce(sentdelta, sentbyte, 0)) as traffic_ ter and (logflag&(1 32)>0) and nullifna(app) i ice order by bandwidth desc)### t where servic TTPS', 'http', 'https') group by app_group, se dth desc	delta, out, count(*) s not null e in

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

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

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

Dataset Name	Description	Log Category
ctap-Top-Sites-By-Browsing-Time	Traffic top sites by browsing time	traffic
<pre>browsetime, sum(bandwidth) as ba traffic_out from ###(select host (bandwidth) as bandwidth, sum(tr from (select hostname, catdesc, (sentbyte, 0)+coalesce(rcvdbyte, sum(coalesce(sentbyte, 0)) as tr</pre>	<pre>ebtr_value(ebtr_agg_flat(browsetime), null ndwidth, sum(traffic_in) as traffic_in, su name, catdesc, ebtr_agg_flat(browsetime) a affic_in) as traffic_in, sum(traffic_out) ebtr_agg_flat(\$browse_time) as browsetime, 0)) as bandwidth, sum(coalesce(rcvdbyte, affic_out from \$log where \$filter and (log _time is not null group by hostname, catde</pre>	<pre>im(traffic_out) as as browsetime, sum as traffic_out sum(coalesce 0)) as traffic_in, gflag&1>0) and</pre>

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

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

select

hourstamp,

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

from

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

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

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

select

(

case is_saas when 1 then & #039;SaaS Apps' else 'Other Apps' end) as app_type, count (distinct app_s) as total_num from ###(select app_s, (case when saas_s>=10 then 1 else 0 end) as is_saas from (select unnest(apps) as app_s, unnest(saasinfo) as saas_s from \$log where \$filter and apps is not null) t group by app_s, is_saas order by is_saas desc)### t group by is_saas order by is_saas

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

select

(

case saas_cat when 0 then & #039;Sanctioned' else 'Unsanctioned' end) as saas_cat_str, count(distinct app_s) as num_saas_app from ###(select app_s, saas_s%10 as saas_cat, sum (sentbyte+rcvdbyte) as bandwidth, count(*) as total_app from (select unnest(apps) as app_s, unnest(saasinfo) as saas_s, coalesce(sentbyte, 0) as sentbyte, coalesce(rcvdbyte, 0) as rcvdbyte from \$log where \$filter and apps is not null) t where saas_s>=10 group by app_s, saas_cat order by bandwidth desc)### t where saas_cat in (0, 1) group by saas_cat order by saas_cat

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

select

(

case saas_cat when 0 then & #039;Sanctioned' else 'Tolerated' end) as saas_cat_str, sum (bandwidth) as bandwidth from ###(select app_s, saas_s%10 as saas_cat, sum (sentbyte+rcvdbyte) as bandwidth, count(*) as total_app from (select unnest(apps) as app_s, unnest(saasinfo) as saas_s, coalesce(sentbyte, 0) as sentbyte, coalesce(rcvdbyte, 0) as rcvdbyte from \$log where \$filter and apps is not null) t where saas_s>=10 group by app_s, saas_cat order by bandwidth desc)### t where saas_cat in (0, 2) group by saas_cat order by saas_cat

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

select

(

case saas_cat when 0 then & #039;Sanctioned' else 'Tolerated' end) as saas_cat_str, sum (total_app) as total_app from ###(select app_s, saas_s%10 as saas_cat, sum (sentbyte+rcvdbyte) as bandwidth, count(*) as total_app from (select unnest(apps) as app_s, unnest(saasinfo) as saas_s, coalesce(sentbyte, 0) as sentbyte, coalesce(rcvdbyte, 0) as rcvdbyte from \$log where \$filter and apps is not null) t where saas_s>=10 group by app_s, saas_cat order by bandwidth desc)### t where saas_cat in (0, 2) group by saas_cat order by saas_cat

Dataset Name	Description	Log Category
saas-SaaS-App-Users-vs-Others	Number of Users of SaaS Apps vs Others	traffic

select

(

case is_saas when 0 then & #039;Other Apps' else 'SaaS Apps' end) as app_type, count (distinct saasuser) as total_user from ###(select saasuser, saas_s/10 as is_saas from (select coalesce(nullifna(`user`), nullifna(`clouduser`), nullifna(`unauthuser`), srcname, ipstr(`srcip`)) as saasuser, unnest(saasinfo) as saas_s from \$log where \$filter and apps is not null) t group by saasuser, is_saas)### t group by app_type

Dataset Name	Description	Log Category
saas-SaaS-App-Users	Number of Users of SaaS Apps	traffic

select

(

case saas_cat when 0 then & #039;Sanctioned' when 1 then 'Unsanctioned' else 'Others' end) as app_type, count(distinct saasuser) as total_user from ###(select saasuser, saas_s%10 as saas_cat from (select coalesce(nullifna(`user`), nullifna(`clouduser`), nullifna (`unauthuser`), srcname, ipstr(`srcip`)) as saasuser, unnest(saasinfo) as saas_s from \$log where \$filter and apps is not null) t where saas_s>=10 group by saasuser, saas_cat)### t group by saas_cat order by saas_cat

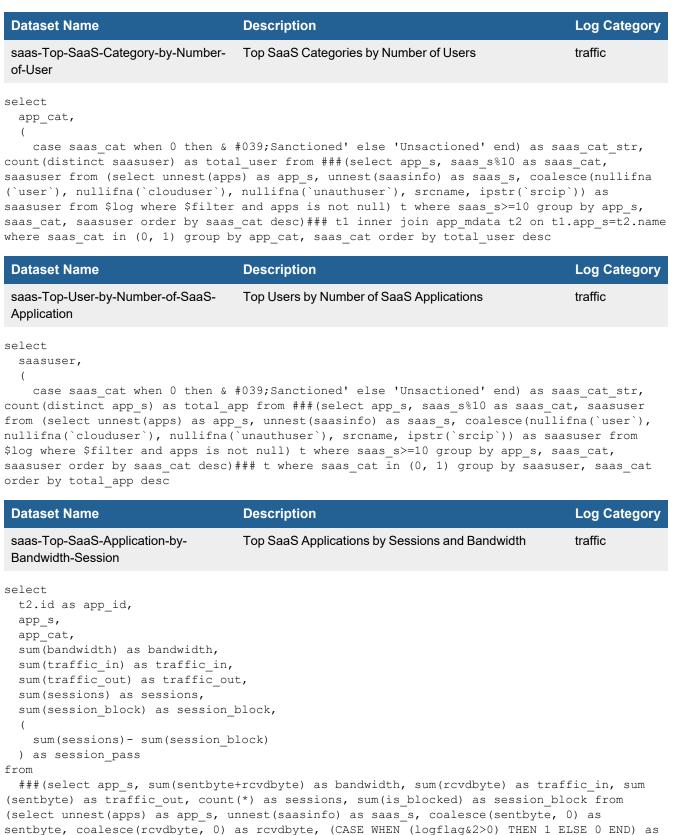
Dataset Name	Description	Log Category
saas-Top-SaaS-User-by-Bandwidth- Session	Top SaaS Users by Bandwidth and Session	traffic
<pre>select saasuser, sum(bandwidth) as bandwidth, sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_ou sum(sessions) as sessions, sum(session_block) as session_i (sum(sessions) - sum(session_b) as session_pass, count(distinct app_s) as total from</pre>	block, lock)	
<pre>###(select saasuser, app_s, sum(sentbyte+rcvdbyte) as bandwidth, sum(rcvdbyte) as traffic_ in, sum(sentbyte) as traffic_out, count(*) as sessions, sum(is_blocked) as session_block from (select coalesce(nullifna(`user`), nullifna(`clouduser`), nullifna(`unauthuser`), srcname, ipstr(`srcip`)) as saasuser, unnest(apps) as app_s, unnest(saasinfo) as saas_s, coalesce(sentbyte, 0) as sentbyte, coalesce(rcvdbyte, 0) as rcvdbyte, (CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END) as is_blocked from \$log where \$filter and apps is not null)</pre>		

Dataset Name	Description	Log Category
saas-Top-Category-by-SaaS- Application-Usage	Top Categories by SaaS Application Usage	traffic
<pre>select app_cat, (</pre>		

t where saas_s>=10 group by saasuser, app_s order by bandwidth desc)### t group by saasuser

case saas_cat when 0 then & #039;Sanctioned' else 'Unsactioned' end) as saas_cat_str, count(distinct app_s) as total_app from ###(select app_s, saas_s%10 as saas_cat from (select unnest(apps) as app_s, unnest(saasinfo) as saas_s from \$log where \$filter and apps is not null) t where saas_s>=10 group by app_s, saas_cat)### t1 inner join app_mdata t2 on t1.app_ s=t2.name where saas_cat in (0, 1) group by app_cat, saas_cat order by total_app desc

order by bandwidth desc



is_blocked from \$log where \$filter and apps is not null) t where saas_s>=10 group by app_

s)### t1 inner join app_mdata t2 on t1.app_s=t2.name group by app_id, app_s, app_cat order by bandwidth desc

Dataset Name	Description	Log Category
saas-Top-Tolerated-SaaS-Application- by-Bandwidth	Top Tolerated SaaS Applications by Bandwidth	traffic
<pre>select app_s, sum(sentbyte + rcvdbyte) as ba from (select unnest(apps) as app_s, unnest(saasinfo) as saas_s coalesce(sentbyte, 0) as s coalesce(rcvdbyte, 0) as r from \$log where \$filter and apps is not null) t where saas_s = 12 group by app_s order by bandwidth desc</pre>	, entbyte,	
Dataset Name	Description	Log Category
saas-drilldown-Top-Tolerated-SaaS- Application	Top Tolerated SaaS Applications	traffic

```
select
```

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

from

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

Dataset Name	Description	Log Category
saas-Top-User-by-Tolerated-SaaS- Application-Drilldown	Top Users by Tolerated SaaS Applications	traffic
select saasuser,		
count(distinct app_s) as tota	l_app	
from	um (sonthutotroudbuto) as handwidth s	

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

Dataset Name	Description	Log Category
saas-drilldown-Top-File-Sharing- SaaS-Application-Detail	Top File Sharing SaaS Applications Detail	traffic
<pre>bandwidth, sum(rcvdbyte) as traf sum(is_blocked) as session_block (`clouduser`), nullifna(`unauthu app_s, unnest(saasinfo) as saas_ as rcvdbyte, (CASE WHEN (logflag \$filter and apps is not null) t</pre>	block,	<pre>(*) as sessions, llifna , unnest(apps) as sce(rcvdbyte, 0) \$log where er order by</pre>

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

```
t2.id as appid,
```

```
(
```

case t2.risk when & #039;5' then 'Critical' when '4' then 'High' when '3' then 'Medium' when '2' then 'Info' else 'Low' end) as risk, app_group, bandwidth, traffic_in, traffic_out, sessions, session_block, session_pass, total_user from (select app_group, count(distinct saasuser) as total_user, sum(bandwidth) as bandwidth, sum(traffic_in) as traffic_in, sum (traffic_out) as traffic_out, sum(sessions) as session, sum(session_block) as session_ block, (sum(sessions)-sum(session_block)) as session_pass from ###(select app_group_name (app_s) as app_group, saasuser, sum(sentbyte+rcvdbyte) as bandwidth, sum(rcvdbyte) as traffic_in, sum(sentbyte) as traffic_out, count(*) as sessions, sum(is_blocked) as session_ block from (select coalesce(nullifna(`user`), nullifna(`clouduser`), nullifna(`unauthuser`), srcname, ipstr(`srcip`)) as saasuser, unnest(apps) as app_s, unnest(saasinfo) as saas_s, coalesce(sentbyte, 0) as sentbyte, coalesce(rcvdbyte, 0) as rcvdbyte, (CASE WHEN (logflag&2>0) THEN 1 ELSE 0 END) as is_blocked from \$log where \$filter and apps is not null) t where saas_s>=10 group by app_group, saasuser order by bandwidth desc)### t group by app_ group) t1 inner join app_mdata t2 on lower(t1.app_group)=lower(t2.name) where t2.app_ cat='Storage.Backup' order by total_user desc, bandwidth desc

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

select

t2.id as appid,

(

case t2.risk when & #039;5' then 'Critical' when '4' then 'High' when '3' then 'Medium'
when '2' then 'Info' else 'Low' end) as risk, app_group, bandwidth, traffic_in, traffic_out,
sessions, session_block, session_pass, total_user from (select app_group, count(distinct
saasuser) as total_user, sum(bandwidth) as bandwidth, sum(traffic_in) as traffic_in, sum
(traffic_out) as traffic_out, sum(sessions) as session, sum(session_block) as session_
block, (sum(sessions)-sum(session_block)) as session_pass from ###(select app_group_name
(app_s) as app_group, saasuser, sum(sentbyte+rcvdbyte) as bandwidth, sum(rcvdbyte) as
traffic_in, sum(sentbyte) as traffic_out, count(*) as sessions, sum(is_blocked) as session_
block from (select coalesce(nullifna(`user`), nullifna(`clouduser`), nullifna(`unauthuser`),
srcname, ipstr(`srcip`)) as saasuser, unnest(apps) as app_s, unnest(saasinfo) as saas_s,
coalesce(sentbyte, 0) as sentbyte, coalesce(rcvdbyte, 0) as rcvdbyte, (CASE WHEN
(logflag&2>0) THEN 1 ELSE 0 END) as is_blocked from \$log where \$filter and apps is not null)
t where saas_s>=10 group by app_group, saasuser order by bandwidth desc)### t group by app_
group) t1 inner join app_mdata t2 on lower(t1.app_group)=lower(t2.name) where t2.app_
cat='Storage.Backup' order by total user desc, bandwidth desc

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

select

& #039;All'::text as country, count(distinct devid) as device_count from ###(select devid from \$log where \$filter group by devid)### t

Dataset Name	Description	Log Category
aware-Network-Endpoint-Devices	Endpoint Devices on Network	
	ategory, 1 as idx, count(distinct ep map_dev.vd, max(lastseen) as itime fr	—

inner join \$ADOM_EPEU_DEVMAP map_dev on t.epid=map_dev.epid where \$filter-drilldown and epname is not null group by epname, map_dev.devid, map_dev.vd) t where \$filter and \$filterdrilldown union all select 'New Devices' as category, 2 as idx, count(distinct epname) as total_num from (select epname, map_dev.devid, map_dev.vd, min(firstseen) as itime from \$ADOM_ENDPOINT t inner join \$ADOM_EPEU_DEVMAP map_dev on t.epid=map_dev.epid where epname is not null group by epname, map_dev.devid, map_dev.vd) t where \$filter and \$filterdrilldown union all select 'Unseen Devices' as category, 3 as idx, count(distinct t1.epname) as total_num from \$ADOM_ENDPOINT t1 where not exists (select 1 from (select epname, map_ dev.devid, map_dev.vd, max(lastseen) as itime from \$ADOM_ENDPOINT t inner join \$ADOM_EPEU_ DEVMAP map_dev on t.epid=map_dev.epid where epname is not null group by epname, map_ dev.devid, map_dev.vd) t2 where \$filter and \$filter-drilldown and t1.epname=t2.epname)) t order by idx

Dataset Name	Description	Log Category
aware-New-Endpoint-Devices	New Endpoint Devices	
<pre>select epid, max(euid) as max_euid from \$ADOM_EPEU_DEVMAP where \$filter - drilldown and euid >= 1024 group by epid); select timestamp, epname as hostname, max(osname) as osname, max(osname) as osname, max(devtype) as devtype, max(srcip) as srcip, string_agg(distinct epname, & #039;,') as user_agg f epdevtype as devtype, epip a epname, max(epdevtype) as ep min(firstseen) as itime from t.epid=map_dev.epid where ep dev.vd) t where \$filter and</pre>	; create temporary table devmap_tmp as rom (select from_itime(itime) as timesta s srcip, epid from (select max(osname) a devtype, max(epip) as epip, t.epid, map \$ADOM_ENDPOINT t inner join \$ADOM_EPEU name is not null group by epname, t.epic \$filter-drilldown) t1 inner join devmap \$ADOM_ENDUSER as teu on devmap_tmp.max_o	amp, osname, epname, as osname, max(epname) as _dev.devid, map_dev.vd, _DEVMAP map_dev on d, map_dev.devid, map_ _tmp on devmap_
Dataset Name	Description	Log Category
aware-New-Endpoint-Devices-Tren	nd New Endpoint Devices Trend	
<pre>select \$flex_timescale(itime) as count(distinct epname) as from (</pre>		

```
select
     epname,
     map_dev.devid,
     map_dev.vd,
     min(firstseen) as itime
   from
     $ADOM_ENDPOINT t
     inner join $ADOM_EPEU_DEVMAP map_dev on t.epid = map_dev.epid
   where
     $filter - drilldown
     and epname is not null
   group by
     epname,
     map_dev.devid,
     map_dev.vd
 ) t
where
 $filter
 and $filter - drilldown
group by
 hodex
order by
 hodex
```

Dataset Name	Description	Log Category
aware-Top-Endpoint-Operating- Systems	Top Endpoint Operating Systems	fct-traffic
—	total_num ,', 1) as osl, hostname from \$log where \$f , hostname)### t group by os order by tota	
Dataset Name	Description	Log Category
Dataset Name aware-Top-Endpoint-Applications- Windows	Description Top Endpoint Applications Windows	Log Category fct-traffic

t group by srcname order by total num desc

Dataset Name	Description	Log Category
aware-Top-Endpoint-Applications-Mac	Top Endpoint Applications Mac	fct-traffic

```
select
srcname1 as srcname,
count(distinct hostname) as total_num
```

from

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

Dataset Name	Description	Log Category
aware-Top-SaaS-Application-by- Number-of-Users	Top SaaS Applications by Number of Users	traffic

select

app_group,

```
count(distinct saasuser) as total_user
```

from

###(select app_group_name(app_s) as app_group, saasuser from (select unnest(apps) as app_ s, unnest(saasinfo) as saas_s, coalesce(nullifna(`user`), nullifna(`clouduser`), nullifna (`unauthuser`), srcname, ipstr(`srcip`)) as saasuser from \$log where \$filter and (logflag&1>0) and apps is not null) t where saas_s>=10 group by app_group, saasuser)### t group by app_group order by total_user desc

Dataset Name	Description	Log Category
aware-Summary-Of-Changes	Summary of Changes	event

select

```
regexp_replace(
```

```
msg,
```

& #039;[^]*\$','') as msg_trim, count(*) as total_num from \$log where \$filter and logid_ to_int(logid)=44547 group by msg_trim order by total_num desc

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

```
select
vulnseverity,
count(distinct vulnname) as vuln_num
from
####(select vulnseverity, vulnname from $log where $filter and nullifna(vulnname) is not
```

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

Dataset Name	Description	Log Category
aware-Vulnerabilities-Trend	Vulnerabilities Trend	fct-netscan
then 1 else 0 end) as critica end) as high, sum(case when 1 (case when lower(vulnseverity	as timescale, as timestamp, sum(case when lower(v 1, sum(case when lower(vulnseverity ower(vulnseverity) = 'medium' then) = 'notice' then 1 else 0 end) as imestamp desc)### t group by timesc	<pre>r) = 'high' then 1 else 0 1 else 0 end) as medium, sum Low from \$log where \$filter</pre>
Dataset Name	Description	Log Category
aware-Top-Critical-Vulnerabilities	Top Critical Vulnerabilities	fct-netscan
select vulnname,		

vulnseverity, vulncat order by total num desc

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

vulnname, vulnseverity, sev_num, vulncat, count(distinct hostname) as total_num from

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

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

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

Dataset Name	Description	Log Category
aware-Top-User-With-Critical- Vulnerabilities	Top Users with Critical Vulnerabilities	fct-netscan
<pre>select hostname, `user` as user_src, vulnname, vulncat, count(*) as total_num from</pre>		
<pre>\$log where \$filter and nullifna(`user`) is not nu and vulnseverity =& #039;Criti total_num desc</pre>	all .cal' group by hostname, user_src, vulnr	name, vulncat order by

Dataset Name	Description	Log Category
aware-Ingress-Data-Flow-By-Zone	Ingress Data Flow By Zone	traffic
select		
app, tag,		

```
sum(rcvdbyte) as rcvdbyte
from
####(select dvid, app, dstintf, sum(coalesce(rcvdbyte, 0)) as rcvdbyte from $log where
$filter group by dvid, app, dstintf having sum(coalesce(rcvdbyte, 0)) > 0 order by rcvdbyte
```

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

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

select

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

###(select dvid, app, srcintf, sum(coalesce(sentbyte, 0)) as sentbyte from \$log where \$filter group by dvid, app, srcintf having sum(coalesce(sentbyte, 0)) > 0 order by sentbyte desc)### ttl inner join (select dvid, intfname, unnest(tags) as tag from intfinfo) tt2 on ttl.dvid=tt2.dvid and ttl.srcintf=tt2.intfname group by app, tag order by sentbyte desc

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

select

hostname, srcip, os, vuln_num, (

CASE sevid WHEN 5 THEN & #039;Critical' WHEN 4 THEN 'High' WHEN 3 THEN 'Medium' WHEN '2' THEN 'Info' ELSE 'Low' END) as vulnseverity, sevid as severity_num, left(cve_agg, 512) as cve_agg from (select hostname, max(srcip) as srcip, string_agg(distinct os1, '/') as os, count(distinct vulnname) as vuln_num, max((CASE vulnseverity WHEN 'Critical' THEN 5 WHEN 'High' THEN 4 WHEN 'Medium' THEN 3 WHEN 'Info' THEN 2 WHEN 'Low' THEN 1 ELSE 0 END)) as sevid, string_agg(distinct cve_id, ',') as cve_agg from ###(select hostname, max(deviceip) as srcip, split_part(os, ',', 1) as os1, vulnname, vulnseverity, vulnid from \$log where \$filter and nullifna(vulnname) is not null and nullifna(vulnseverity) is not null group by hostname, os1, vulnname, vulnseverity, vulnid)### t1 left join fct_mdata t2 on t1.vulnid=t2.vid::int group by hostname) t order by severity_num desc, vuln_num desc

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

select

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

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

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

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

```
Dataset NameDescriptionLog Categoryaware-Threats-By-DayThreats by Dayvirusselect
```

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

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

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

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

(

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

Dataset Name	Description	Log Category
aware-Count-Of-Malware-Events	Count of Malware Events	virus
<pre>select virus, count(*) as total_num from \$log where \$filter and nullifna(virus) is not null group by virus order by</pre>	L	
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 victim) as v count(distinct source) as s sum(total_num) as total_num</pre>	ource,	
from		
(CASE WHEN direction='incomin direction='incoming' THEN src app-ctrl where \$filter and lo malware_type, apprisk, victim	g' THEN dstip ELSE srcip END ip ELSE dstip END) as victim wer(appcat)='botnet' and app , source order by total_num	, count(*) as total_num from \$log- risk is not null group by app,

crlevel::text as risk_level, (CASE WHEN direction='incoming' THEN dstip ELSE srcip END) as source, (CASE WHEN direction='incoming' THEN srcip ELSE dstip END) as victim, 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, victim, source order by total_num desc)### union all ###(select attack as virus, (case when eventtype='botnet' then 'Botnet C&C' else 'Virus' end) as malware_type, crlevel::text as risk_level, (CASE WHEN direction='incoming' THEN dstip ELSE srcip END) as source, (CASE WHEN direction='incoming' THEN srcip ELSE dstip END) as victim, count(*) as total_num from \$log-attack where \$filter and (logflag&16>0) and crlevel is not null group by virus, malware_type, crlevel, victim, source order by total_num desc)###) t group by virus, malware_type, risk_level order by total_num desc

Dataset Name	Description	Log Category
aware-Top-Failed-Login-Attempts	Top Failed Login Attempts	event
select		
`user` as f_user,		
ui,		
dstip,		
count(status) as total_failed	1	
from		
\$log		
where		
\$filter		
and nullifna(`user`) is not n		
and logid_to_int(logid) = 320	002	
group by		
ui,		
f_user,		
dstip		
order by		
total_failed desc		

Dataset Name

	2000111200		
aware-Top-Failed-Authentication- Attempts	VPN failed logins		event
<pre>select f_user, tunneltype, sum(total_num) as total_num</pre>			
<pre>from ###(select coalesce(nullifna(`; total_num from \$log where \$filter (tunneltype, 3)='ssl') and action</pre>	r and subtype='vpn' and	(tunneltype='ipsec'	or left

Description

(nullifna(`xauthuser`), nullifna(`user`)) is not null group by f_user, tunneltype)### t group by f_user, tunneltype order by total_num desc

Dataset Name	Description	Log Category
aware-Top-Denied-Connections	Top Denied Connections	traffic
<pre>select coalesce(nullifna(`user`),</pre>		

Log Category

ipstr(`srcip`)
) as user_src,
service || & #039;(' || ipstr(srcip) || ')' as interface, dstip, count(*) as total_num
from \$log where \$filter and (logflag&1>0) and action = 'deny' group by user_src, interface,
dstip order by total_num desc

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

select

devid,

& #039;Failed' as results, count(distinct reason) as total_num from ###(select devid, reason from \$log where \$filter and subtype='compliance-check' and result='fail' group by devid, reason)### t group by devid, results order by total_num desc

Dataset Name	Description	Log Category
aware-loc-Blacklist-Summary	IOC Blacklist Summary	app-ctrl
drop		
select	create temporary table tmp_ep_eu_map as (
epid, euid		
from		
\$ADOM_EPEU_DEVMAP where		
euid >= 1024		
);		
select		
coalesce(
nullifna (epname),		
nullifna(
ipstr(`srcip`)		
),		
	user_agg, sevid, (CASE sevid WHEN 5 THEN 'Crit	
-	WHEN '2' THEN 'Info' ELSE 'Low' END) as seven at th1.epid, srcip, sevid, bl count, threats fr	
	evid, sum(bl count) as bl count from ((select	
	.ct, unnest(dvid) as dvid s from \$ADOMTBL PLHD	
	ect epid, srcip, day st as itime, bl count, ve	
	PLHD INTERIM IOC VERDICT where bl count>0)) tvo	
	rid s where \$filter and \$filter-drilldown and \$	
	join (select epid, string agg(name, ',') as the	_
	t epid, thid, itime, unnest(dvid) as dvid s fr	
	day st as itime, dvid from \$ADOMTBL PLHD IOC \	
bl_count>0) tal) union all (sele	ect epid, thid, itime, unnest(dvid) as dvid_s f	from (select
	day_st as itime, dvid from \$ADOMTBL_PLHD_INTER	
where bl_count>0) ta2)) t inner	join devtable td on td.dvid = t.dvid_s where	\$filter and
	er group by epid, thid) thr inner join td_threa	
	epid) th2 on th1.epid=th2.epid) t1 left join	
	as user_agg from tmp_ep_eu_map tpu inner joir	
	p by epid) t2 on t2.epid=t1.epid inner join \$7	DOM_ENDPOINT
as tep on tep.epid=t1.epid order	by total_bl desc, sevid 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(
<pre>from_itime(itime),</pre>			
& #039;Day') as day_st from (select epid, day_st as itime, unnest(dvid) as dvid_s			
from \$ADOMTBL_PLHD_INTERIM_IOC_	VERDICT where \$filter-drilldown and	cs_count>0 union all	
(select epid, day_st as itime,	unnest(dvid) as dvid_s from \$ADOMTB	L_PLHD_IOC_VERDICT where	
<pre>\$filter-drilldown and cs_count></pre>	0)) t inner join devtable td on td.	dvid = t.dvid_s where	
\$filter and \$filter-drilldown g	roup by day_st) tt order by itime		

Description	Log Category
IOC Potential Breach by Day	app-ctrl
ERDICT where \$filter-drilldown and nnest(dvid) as dvid_s from \$ADOMTB)) t inner join devtable td on td.o	cs_count>0 union all L_PLHD_IOC_VERDICT where
oup by day_st) tt order by itime	
	· · · · · ·

Dataset Name	Description	Log Category
aware-loc-Suspicion-Summary	IOC Suspicion Summary	app-ctrl
<pre>verdict, threats from (select th from (select epid, srcip, min(it verdict, max(cs_score) as cs_sco verdict, cs_score, unnest(dvid) drilldown and bl_count=0 and cs_</pre>	cs_count as total_cs, cs_score as r l.epid, srcip, itime, cs_count, ven ime) as itime, sum(cs_count) as cs_ ore from ((select epid, srcip, day_s as dvid_s from \$ADOMTBL_PLHD_IOC_VE count>0) union all (select epid, sn c(dvid) as dvid_s from \$ADOMTBL_PLHE	cdict, cs_score, threats count, max(verdict) as st as itime, cs_count, ERDICT where \$filter- ccip, day_st as itime, cs_

where \$filter-drilldown and bl_count=0 and cs_count>0)) tvdt inner join devtable td on td.dvid = tvdt.dvid_s where \$filter and \$filter-drilldown group by epid, srcip) th1 inner join (select epid, string_agg(name, ',') as threats from ((select epid, thid from ((select epid, thid, itime, unnest(dvid) as dvid_s from (select epid, unnest(threatid) as thid, day_ st as itime, dvid from \$ADOMTBL_PLHD_IOC_VERDICT where bl_count=0 and cs_count>0) tal) union all (select epid, thid, itime, unnest(dvid) as dvid_s from (select epid, unnest(threatid) as thid, day_st as itime, dvid from \$ADOMTBL_PLHD_INTERIM_IOC_VERDICT where bl_count=0 and cs_ count>0) ta2)) tt1 inner join devtable td on td.dvid = tt1.dvid_s where \$filter and \$filterdrilldown group by epid, thid) thr inner join td_threat_name_mdata tm on tm.id=thr.thid) tt2 group by epid) th2 on th1.epid=th2.epid) t inner join \$ADOM_ENDPOINT as tep on tep.epid=t.epid order by max verdict desc, max cs desc, total cs desc

Dataset Name	Description	Log Category	
aware-Botnet-IP	Top Source IP Affected by Botnet	virus	
select			
f_user,			
source,			
string_agg(
distinct `virus`,			
& #039;,') as virus_ag	g, count(distinct ipstr(`victim`)) as dsti	p_cnt, max(action) as	
<pre>action, sum(total_num) as total_num, min(from_itime(first_seen)) as first_seen, max(from_</pre>			
itime(last_seen)) as last_seen from ###(select coalesce(nullifna(`user`), nullifna			
(`unauthuser`)) as f_user, virus, (CASE WHEN direction='incoming' THEN dstip ELSE srcip END)			
as source, (CASE WHEN direction='incoming' THEN srcip ELSE dstip END) as victim, max(action)			
as action, count(*) as total_num, min(itime) as first_seen, max(itime) as last_seen from			
<pre>\$log where \$filter and logid in ('0202009248', '0202009249') and virus is not null group by</pre>			
f_user, virus, source, vic	tim order by total_num desc) ### t group by	source, f_user order by	

total num desc

Dataset Name	Description	Log Category
aware-Botnet-Domain	New Botnet Domains	dns
<pre>select botnet, count(distinct `qname`) count(distinct instr(`dstin`</pre>	• _ ·	

```
distinct ipstr(`dstip`)
) as dnssvr_cnt,
sum(total_num) as total_num,
min(
   from_itime(first_seen)
) as first_seen,
max(
   from_itime(last_seen)
) as last_seen
```

from

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

Dataset Name	Description	Log Category
aware-High-Risk-URL-Category	Category of High Risk URLs	webfilter

select
catdesc,
string_agg(
 distinct hostname,
 & #039;,') as hostname_agg, max(action) as action, sum(total_num) as total_num, min
(from_itime(first_seen)) as first_seen, max(from_itime(last_seen)) as last_seen from ###
(select catdesc, hostname, max(action) as action, count(*) as total_num, min(itime) as
first_seen, max(itime) as last_seen from \$log where \$filter and cat in (26, 61, 86, 88, 90,
91, 93) group by catdesc, hostname order by total_num desc)### t group by catdesc order by
total num desc

Dataset Name	Description	Log Category
aware-Malicious-Files	Type of Malicious Files from AV and Sandbox	virus
<pre>select virus, left(url_agg, 1000) as url_agg left(filename_agg, 1000) as fi quarskip, action, from_sandbox, total_num, first_seen, </pre>		
last_seen from		
(select virus, string_agg(distinct url,		
& #039; ') as url_ag max(quarskip) as quarskip, max(a (total_num) as total_num, min(fr 	<pre>gg, string_agg(distinct filename, ') as action) as action, max(from_sandbox) as from_s com_itime(first_seen)) as first_seen, max(from act virus, url, filename, max(quarskip) as qu logid in ('0211009234', '0211009235') then 1 e num, min(itime) as first_seen, max(itime) as not null and logid in ('0211009234', '0201009</br></pre>	sandbox, sum n_itime(last_ uarskip, max else 0 end) as last_seen from

'0211008192', '0211008193', '0211008194', '0211008195') group by virus, url, filename, from_ sandbox order by total_num desc)### t group by virus) t order by total_num desc

Dataset Name	Description	Log Category
newthing-New-Users	New users	fct-traffic
drop		
table if exists rpt_tmptbl_1	;	
drop		
table if exists rpt_tmptbl_2	; create temporary table rpt_tr	mptbl_1 as
select	_	
f_user,		
<pre>min(start_time) as start_tim</pre>	e	
from		
###(select coalesce(nullifna	(`user`), ipstr(`srcip`)) as f	_user, min(dtime) as start_time
from \$log where \$pre_period \$f	ilter group by f_user order by	<pre>start_time desc)### t group by</pre>
<pre>f_user; create temporary table</pre>	<pre>rpt_tmptbl_2 as select f_user;</pre>	, min(start_time) as start_time

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

D	Dataset Name	Description	Log Category
n	ewthing-New-Devices	New devices	fct-traffic

drop

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

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

Dataset Name	Description	Log Category
newthing-New-Software-Installed	New software installed	fct-traffic
<pre>drop table if exists rpt_tmptbl_1; drop table if exists rpt_tmptbl_2; select srcproduct, hostname</pre>	create temporary table rpt_tmptbl_1 a	as
<pre>from ###(select srcproduct, hostname from \$log where \$pre_period \$filter and nullifna (srcproduct) is not null group by srcproduct, hostname order by srcproduct)### t group by srcproduct, hostname; create temporary table rpt_tmptbl_2 as select srcproduct, hostname from ###(select srcproduct, hostname from \$log where \$filter and nullifna(srcproduct) is not null group by srcproduct, hostname order by srcproduct)### t group by srcproduct, hostname; select srcproduct, string_agg(distinct hostname, ',') as host_agg from rpt_tmptbl_2 where not exists (select 1 from rpt_tmptbl_1 where rpt_tmptbl_2.srcproduct=rpt_tmptbl_ 1.srcproduct) group by srcproduct order by srcproduct</pre>		

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

```
drop
  table if exists rpt_tmptbl_1;
drop
  table if exists rpt_tmptbl_2; create temporary table rpt_tmptbl_1 as
select
  threat_name,
  cat_id,
  source
from
  (
```

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

```
Dataset Name
                                  Description
                                                                                 Log Category
 newthing-dns-Botnet-Domain-IP
                                  New Queried Botnet C&C Domains and IPs
                                                                                 dns
drop
  table if exists rpt_tmptbl_1;
drop
  table if exists rpt tmptbl 2; create temporary table rpt tmptbl 1 as
select
 domain,
 malware type,
 action s as action,
 srcip,
  sevid
from
  ###(select coalesce(botnetdomain, ipstr(botnetip)) as domain, cast('Botnet C&C' as char
(32)) as malware type, (case when action='block' then 'Blocked' when action='redirect' then
'Redirected' else 'Passed' end) as action s, srcip, (CASE WHEN level IN ('critical',
'alert', 'emergency') THEN 5 WHEN level='error' THEN 4 WHEN level='warning' THEN 3 WHEN
level='notice' THEN 2 ELSE 1 END) as sevid, coalesce(botnetdomain, ipstr(botnetip)) as
sources s, count(*) as total num from $log where $pre period $filter and (botnetdomain is
```

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

Dataset Name	Description	Log Category
newthing-New-Security-Threats- Timeline	New security threats timeline	virus
drop table if exists rpt tmptbl 1;		
<pre>drop table if exists rpt_tmptbl_2;</pre>	create temporary table rpt_tmptbl_1 as	
<pre>select threat_name,</pre>		
cat_id, source		
from (

###(select app as threat name, 1 as cat id, (CASE WHEN direction='incoming' THEN dstip ELSE srcip END) as source from \$log-app-ctrl where \$pre period \$filter and nullifna(app) is not null and lower(appcat)='botnet' group by threat name, cat id, source)### union all ### (select virus as threat_name, 2 as cat_id, (CASE WHEN direction='incoming' THEN dstip ELSE srcip END) as source from \$log-virus where \$pre period \$filter and nullifna(virus) is not null group by threat name, cat id, source) ### union all ###(select attack as threat name, 3 as cat id, (CASE WHEN direction='incoming' THEN dstip ELSE srcip END) as source from \$logattack where \$pre period \$filter and nullifna(attack) is not null group by threat name, cat id, source)###) t; create temporary table rpt tmptbl 2 as select timestamp, threat name, cat id, source from (###(select \$flex timestamp as timestamp, app as threat name, 1 as cat id, (CASE WHEN direction='incoming' THEN dstip ELSE srcip END) as source from \$log-app-ctrl where \$filter and nullifna(app) is not null and lower(appcat)='botnet' group by timestamp, threat name, cat id, source order by timestamp)### union all ###(select \$flex timestamp as timestamp, virus as threat name, 2 as cat id, (CASE WHEN direction='incoming' THEN dstip ELSE srcip END) as source from \$log-virus where \$filter and nullifna(virus) is not null group by timestamp, threat name, cat id, source order by timestamp) ### union all ###(select \$flex timestamp as timestamp, attack as threat name, 3 as cat id, (CASE WHEN direction='incoming' THEN dstip ELSE srcip END) as source from \$log-attack where \$filter and nullifna(attack) is not null group by timestamp, threat_name, cat_id, source order by timestamp)###) t; select \$flex datetime(timestamp) as timescale, count(distinct source) 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

```
Log Category
 Dataset Name
                                  Description
 newthing-New-Vulnerability
                                  New vulnerabilities
                                                                                   fct-netscan
drop
  table if exists rpt_tmptbl_1;
drop
  table if exists rpt_tmptbl_2; create temporary table rpt_tmptbl_1 as
select
  vulnid,
  vulnname,
  vulnseverity,
  vulncat,
  hostname
from
  ###(select vulnid, vulnname, vulnseverity, vulncat, hostname from $log where $pre period
$filter and nullifna(vulnname) is not null group by vulnid, vulnname, vulnseverity, vulncat,
hostname)### t group by vulnid, vulnname, vulnseverity, vulncat, hostname; create temporary
table rpt tmptbl 2 as select vulnid, vulnname, vulnseverity, vulncat, hostname from ###
```

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

Dataset Name	Description	Log Category
newthing-New-Vulnerability-Graph	New vulnerabilities (Graph)	fct-netscan
drop		
<pre>table if exists rpt_tmptbl_1; drop</pre>		
-	create temporary table rpt tmptbl	1 as
select	create comporting capit ipt_emptor_	
vulnid,		
vulnname,		
vulnseverity,		
vulncat,		
hostname		
from		
<pre>###(select vulnid, vulnname,</pre>	vulnseverity, vulncat, hostname fro	om \$log where \$pre_period
	is not null group by vulnid, vulnna	
	vulnname, vulnseverity, vulncat, h	
	ulnid, vulnname, vulnseverity, vulno	
	severity, vulncat, hostname from \$10	
	group by vulnid, vulnname, vulnseve	1 .
	vulnname, vulnseverity, vulncat, h	
	<pre>rulnid) as vuln_num from rpt_tmptbl_ tmptbl 2.vulnid=rpt tmptbl 1.vulnid)</pre>	_
	cy='Critical' then 5 when vulnseveri	
-	hen vulnseverity='Low' then 2 when v	
else 0 end) desc		

```
Dataset Name
                                                                                      Log Category
                                    Description
 newthing-System-Alerts
                                                                                      local-event
                                    System Alerts
select
 from_itime(itime) as timestamp,
 msg
from
  $log
where
 $filter
 and msg is not null
  and level =& #039; critical' order by timestamp desc
 Dataset Name
                                    Description
                                                                                      Log Category
newthing-Configuration-Changes
                                    Configuration Changes
                                                                                      event
select
 `user` as f_user,
 devid,
 from_dtime(dtime) as time_s,
 ui,
 msg
from
  $log
where
 $filter
 and cfgtid>0
order by
 time s desc
Dataset Name
                                    Description
                                                                                      Log Category
newthing-FortiGate-Upgrades
                                    FortiGate Upgrades
                                                                                      event
select
 devid,
 from dtime(dtime) as time s,
 info[1] as intf,
 info[2] as prev_ver,
 info[3] as new ver
from
```

devid, dtime, regexp_matches(msg, & #039;from ([^]+) \\(([^]+) -> ([^)]+)\\)') as info from \$log where \$filter and action='restore-image') t order by time_s desc
Dataset Name Description Log Category

Dutaset Nume	Description	
newthing-User-Upgrades	User Upgrades	fct-event

(

select

```
drop
  table if exists rpt_tmptbl_1;
drop
  table if exists rpt_tmptbl_2; create temporary table rpt_tmptbl_1 as
select
  fgtserial,
  hostname,
  deviceip,
  os,
  dtime
from
  ###(select distinct on (fgtserial, hostname) fgtserial, hostname, deviceip, os, dtime from
  Slog where Spre period Sfilter and hostname is not null order by fgtserial hostname dtime
```

\$log where \$pre_period \$filter and hostname is not null order by fgtserial, hostname, dtime desc)### t; create temporary table rpt_tmptbl_2 as select fgtserial, hostname, deviceip, os, dtime from ###(select distinct on (fgtserial, hostname) fgtserial, hostname, deviceip, os, dtime from \$log where \$filter and hostname is not null order by fgtserial, hostname, dtime desc)### t; select distinct on (1, 2) t2.fgtserial as devid, t2.hostname, t2.deviceip, t1.os as prev_os, t2.os as cur_os, from_dtime(t1.dtime) as time_s from rpt_tmptbl_2 t2 inner join rpt_tmptbl_1 t1 on t2.fgtserial=t1.fgtserial and t2.hostname=t1.hostname and t2.os!=t1.os order by devid, t2.hostname, t1.dtime desc

Dataset Name	Description	Log Category
GTP-List-of-APN-Used	List of APNs Used	gtp
<pre>select apn, from_dtime(min(first_seen)) as first_seen, from_dtime(max(last_seen)) as last_seen from</pre>		
<pre>###(select apn, min(dtime) as</pre>	first_seen, max(dtime) as last_seen from \$log roup by apn order by last_seen desc)### t group	

by last_seen desc, first_seen

Dataset Name	Description	Log Category
GTP-Top-APN-by-Bytes	Top APNs by Bytes	gtp
<pre>select apn, sum(coalesce(`u-bytes`, 0)) as total_bytes from \$log where \$filter and nullifna(apn) is not null and status =& #039;traffic-cou by total bytes desc</pre>	unt' group by apn having sum(coalesce(`u-bytes`	7, 0))>0 order

Dataset Name	Description	Log Category
GTP-Top-APN-by-Duration	Top APNs by Duration	gtp
<pre>select apn, sum(coalesce(duration, 0)) as total_dura from tage </pre>		
<pre>\$log where \$filter and nullifna(apn) is not n and status =& #039;traffic by total_dura desc</pre>	null c-count' group by apn having sum(coale	esce(duration, 0)) >0 order

Dataset Name	Description	Log Category
GTP-Top-APN-by-Packets	Top APNs by Number of Packets	gtp
<pre>select apn, sum(coalesce(`u-pkts`, 0)) as total_num from \$log where</pre>		
\$filter and nullifna(apn) is not nu	ull -count' group by apn having sum(coalesce)	(`u-pkts`, 0))>0 order

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

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

```
from
```

###(select coalesce(botnetdomain, ipstr(botnetip)) as domain, qname, 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, qname, 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
char(32)) as malware_typ	s_s) as sources, al_num otnetdomain, ipstr(botnetip)) as domain, qname, c be, (case when action='block' then 'Blocked' when Passed' end) as action, srcip, (CASE WHEN level	action='redirect'
evel='notice' THEN 2 EL sources_s, count(*) as t potnetip is not null) gr	HEN 5 WHEN level='error' THEN 4 WHEN level='warni LSE 1 END) as sevid, coalesce(botnetdomain, ipstr cotal_num from \$log where \$filter and (botnetdoma coup by domain, qname, action, srcip, sevid order e_type, action order by total_num desc	ng' THEN 3 WHEN (botnetip)) as in is not null or
evel='notice' THEN 2 EL ources_s, count(*) as t ootnetip is not null) gr	USE 1 END) as sevid, coalesce(botnetdomain, ipstr cotal_num from \$log where \$filter and (botnetdoma coup by domain, qname, action, srcip, sevid order	ng' THEN 3 WHEN (botnetip)) as in is not null or

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

from

###(select coalesce(botnetdomain, ipstr(botnetip)) as domain, qname, 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, qname, 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
<pre>select domain, srcip, sevid, (</pre>		
THEN 'Info' ELSE 'Low' END) (botnetip)) as domain, qname action='block' then 'Blocked	<pre>#039;Critical' WHEN 4 THEN 'High' WHEN 3 T as severity from ###(select coalesce(botnet , cast('Botnet C&C' as char(32)) as malware ' when action='redirect' then 'Redirected' vel IN ('critical', 'alert', 'emergency') T</pre>	domain, ipstr _type, (case when else 'Passed' end) as

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, qname, action, srcip, sevid order by sevid desc)### t group by domain, srcip, sevid order by sevid desc, domain

Dataset Name	Description	Log Category
dns-High-Risk-Source	High Risk Sources	dns

select .

```
srcip,
sum(total_num) as total_num,
sum(
    case when sevid = 5 then total_num else 0 end
) as num_cri,
sum(
    case when sevid = 4 then total_num else 0 end
) as num_hig,
sum(
    case when sevid = 3 then total_num else 0 end
) as num_med
from
```

###(select srcip, (CASE WHEN level IN ('critical', 'alert', 'emergency') THEN 5 WHEN level='error' THEN 4 WHEN level='warning' THEN 3 WHEN level='notice' THEN 2 ELSE 1 END) as sevid, count(*) as total_num from \$log where \$filter and srcip is not null group by srcip, sevid order by total_num desc)### t where sevid>=3 group by srcip having sum(total_num)>0 order by total_num desc

Dataset Name	Description	Log Category
dns-DNS-Request-Over-Time	DNS Request Over Time	dns
<pre>select \$flex_timescale(timestamp) as sum(case when sevid = 5 then to) as num_cri, sum(case when sevid = 4 then to) as num_hig, sum(case when sevid = 3 then to) as num_med, sum(case when sevid = 2 then to) as num_inf, sum(case when sevid = 1 then to) as num_low from</pre>	s timescale, ptal_num else 0 end ptal_num else 0 end ptal_num else 0 end ptal_num else 0 end ptal_num else 0 end stimestamp, (CASE WHEN level ='error' THEN 4 WHEN level='wa. ND) as sevid, count(*) as tota	rning' THEN 3 WHEN l_num from \$log where \$filter

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

Dataset Name	Description	Log Category
dns-Top-Domain-Lookup-Failure-Bar	Top Domain Lookup Failures	dns
select qname, srcip, count(*) as total num		
from		
\$log		
where		
\$filter		
and qname is not null		
and (
-	id_to_int(logid)=54200) group by q	name, srcip order by
total_num desc		

Dataset Name	Description	Log Category
dns-Top-Domain-Lookup-Failure- Table	Top Domain Lookup Failures	dns
<pre>select qname, srcip, count(*) as total_num from \$log where \$filter and qname is not null and (</pre>		
<pre>action =& #039;block' or 1 total_num desc</pre>	logid_to_int(logid)=54200) group by qn	ame, srcip order by
Dataset Name	Description	Log Category

Dataset Name	Description	Log Category
dns-Query-Timeout	Query Timeout	dns

```
select
srcip,
qname,
count(*) as total_num
from
$log
where
$filter
and srcip is not null
and logid_to_int(logid) = 54200
group by
qname,
srcip
order by
total_num desc
```

Dataset Name	Description	Log Category
dns-Blocked-Query	Blocked Queries	dns
<pre>select srcip, msg, count(*) as total_num from \$log where \$filter and srcip is not null and action =& #039;block' gr</pre>	oup by srcip, msg order by total_num desc	

Dataset Name	Description	Log Category
perf-stat-cpu-usage-drilldown	Fortigate resource detail timeline	event
select		
hodex,		
cast(
<pre>sum(cpu_ave)/ count(*) as</pre>	decimal(6, 0)	
) as cpu_ave,		
cast(
<pre>sum(mem_ave)/ count(*) as</pre>	decimal(6, 0)	
) as mem_ave,		
cast (
<pre>sum(disk_ave)/ count(*) a) as disk_ave</pre>	s decimal(6, 0)	
) as disk_ave, cast(
<pre>sum(log rate)/ count(*) a</pre>	s decimal(10 2)	
) as log rate,	5 decimar(10, 2)	
cast(
<pre>sum(sessions)/ count(*) a</pre>	s decimal(10, 0)	
) as sessions,		
cast(
<pre>sum(sent_kbps)/ count(*)</pre>	as decimal(10, 0)	
) as sent_kbps,		
cast(

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

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

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

Dataset Name	Description	Log Category
perf-stat-mem-usage-drilldown	Fortigate resource detail timeline	event
<pre>select hodex, cast(sum(cpu_ave)/ count(*) as c) as cpu_ave, cast(sum(mem_ave)/ count(*) as c) as mem_ave, cast(sum(disk_ave)/ count(*) as) as disk_ave,</pre>	decimal(6, 0)	

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

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

from

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

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

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

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

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

Dataset Name	Description	Log Category
perf-stat-sessions-drilldown	Fortigate resource detail timeline	event
select		
hodex,		
cast(
<pre>sum(cpu_ave) / count(*)</pre>	as decimal(6, 0)	
) as cpu_ave,		
cast(
<pre>sum(mem_ave)/ count(*)</pre>	as decimal(6, 0)	
) as mem_ave,		
cast(
<pre>sum(disk_ave)/ count(*)</pre>	as decimal(6, 0)	
) as disk_ave,		
cast(2a decimal (10 2)	
<pre>sum(log_rate) / count(*)) as log_rate</pre>	as decimar(10, 2)	
) as log_rate, cast(
<pre>sum(sessions)/ count(*)</pre>	as decimal(10 0)	
) as sessions,		
cast(
<pre>sum(sent_kbps)/ count(*</pre>) as decimal(10, 0)	
) as sent kbps,		
cast(
<pre>sum(recv kbps)/ count(*</pre>) as decimal(10, 0)	
) as recv_kbps,		
cast(
<pre>sum(transmit_kbps)/ cou</pre>	nt(*) as decimal(10, 0)	
) as transmit_kbps,		
max(mem_peak) as mem_peak	,	
<pre>max(disk_peak) as disk_pe</pre>		
max(cpu_peak) as cpu_peak		
<pre>max(lograte_peak) as logr</pre>		
max(session_peak) as sess	—	
<pre>max(transmit_kbps_peak) a </pre>	s transmit_kbps_peak,	
cast ($a = d = a = \frac{1}{2} (10 = 0)$	
<pre>sum(cps_ave) / count(*)) as cps ave,</pre>	as decimal(10, 0)	
<pre>max(cps_peak) as cps_peak from</pre>		
(
select		
hodex,		
devid,		
get_fgt_role(devid, s	elot) as role,	
cast (
<pre>sum(cpu_ave) / count</pre>	(*) as decimal(6, 0)	
) as cpu_ave,		
cast(

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

from

Dataset Name	Description	Log Category
perf-stat-lograte-drilldown	Fortigate resource detail timeline	event
select		
hodex,		
cast(
<pre>sum(cpu_ave) / count(*) </pre>	as decimal(6, 0)	
) as cpu_ave,		
<pre>cast(sum(mem ave) / count(*)</pre>	as decimal $(6, 0)$	
) as mem ave,		
cast(
<pre>sum(disk ave) / count(*)</pre>	as decimal(6, 0)	
) as disk_ave,		
cast(
<pre>sum(log_rate)/ count(*)</pre>	as decimal(10, 2)	
) as log_rate,		
cast(
<pre>sum(sessions)/ count(*)</pre>	as decimal(10, 0)	
) as sessions, cast(
sum(sent kbps)/ count(*) as decimal $(10, 0)$	
) as sent kbps,) as accimat(10, 0)	
cast(
<pre>sum(recv_kbps)/ count(*</pre>) as decimal(10, 0)	
) as recv_kbps,		
cast(
—	nt(*) as decimal(10, 0)	
) as transmit_kbps,		
max(mem_peak) as mem_peak		
<pre>max(disk_peak) as disk_pe max(any peak) as any peak</pre>		
<pre>max(cpu_peak) as cpu_peak max(lograte peak) as logr</pre>		
max(session_peak) as sess	—	
max(transmit kbps peak) a		
cast(
<pre>sum(cps_ave)/ count(*)</pre>	as decimal(10, 0)	
) as cps_ave,		
<pre>max(cps_peak) as cps_peak</pre>	:	

```
from
  (
   select
     hodex,
     devid,
     get fgt role(devid, slot) as role,
     cast(
       sum(cpu ave) / count(*) as decimal(6, 0)
     ) as cpu ave,
     cast(
       sum(mem ave) / count(*) as decimal(6, 0)
     ) as mem ave,
     cast(
       sum(disk ave) / count(*) as decimal(6, 0)
     ) as disk ave,
     cast(
       sum(log rate) as decimal(10, 2)
     ) as log rate,
     cast(
       sum(sessions) as decimal(10, 0)
     ) as sessions,
     cast(
       sum(sent kbps) as decimal(10, 0)
     ) as sent kbps,
     cast(
       sum(recv_kbps) as decimal(10, 0)
     ) as recv kbps,
     cast(
       sum(transmit kbps) as decimal(10, 0)
     ) as transmit kbps,
     max(mem peak) as mem peak,
     max(disk peak) as disk peak,
     max(cpu peak) as cpu peak,
     cast(
       max(lograte peak) as decimal(10, 2)
     ) as lograte peak,
     max(session peak) as session peak,
     max(transmit kbps peak) as transmit kbps peak,
     cast(
       sum(cps_ave) as decimal(10, 0)
     ) as cps ave,
     sum(cps peak) as cps peak
    from
      (
        select
          $flex timescale(timestamp) as hodex,
         devid,
          slot,
          sum(total cpu) / sum(count) cpu ave,
          sum(total mem) / sum(count) as mem ave,
          sum(total disk) / sum(count) as disk ave,
          sum(
            total trate + total erate + total orate
          )/ 100.00 / sum(count) as log rate,
          sum(totalsession) / sum(count) as sessions,
          sum(sent) / sum(count) as sent kbps,
```

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

from

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

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

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

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

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

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

```
Dataset NameDescriptionLog Categoryperf-stat-usage-summary-averageFortigate resource summary vieweventselect<br/>devid,<br/>get_fgt_role(devid, slot) as role,<br/>cast(<br/>sum(cpu_ave)/ count(*) as decimal(6, 0)<br/>) as cpu_ave,sum(cpu_ave,
```

```
cast(
    sum(mem ave) / count(*) as decimal(6, 0)
  ) as mem ave,
  cast(
   sum(disk ave) / count(*) as decimal(6, 0)
  ) as disk ave,
  cast(
   sum(log rate) as decimal(10, 2)
 ) as log rate,
 cast(
   sum(sessions) as decimal(10, 0)
  ) as sessions,
 cast(
   sum(sent kbps) as decimal(10, 0)
  ) as sent kbps,
  cast (
   sum(recv kbps) as decimal(10, 0)
  ) as recv kbps,
  cast(
   sum(transmit kbps) as decimal(10, 0)
  ) as transmit_kbps,
 max(mem_peak) as mem_peak,
 max(disk peak) as disk peak,
 max(cpu_peak) as cpu_peak,
 cast(
   max(lograte peak) as decimal(10, 2)
 ) as lograte peak,
 max(session peak) as session peak,
 max(transmit kbps peak) as transmit kbps peak
from
  (
   select
     devid,
     slot,
     sum(total cpu) / sum(count) as cpu ave,
      sum(total_mem) / sum(count) as mem_ave,
     sum(total disk) / sum(count) as disk ave,
     sum(
       total trate + total erate + total orate
     )/ 100.00 / sum(count) as log rate,
     sum(totalsession) / sum(count) as sessions,
      sum(sent) / sum(count) as sent kbps,
     sum(recv) / sum(count) as recv kbps,
     sum(sent + recv) / sum(count) as transmit_kbps,
     max(mem peak) as mem peak,
     max(disk peak) as disk peak,
     max(cpu peak) as cpu peak,
     max(lograte_peak) / 100.00 as lograte_peak,
     max(session peak) as session peak,
     max(transmit peak) as transmit kbps peak
    from
      ###(select $flex timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as
```

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

Dataset Name	Description	Log Category
perf-stat-usage-summary-peak	Fortigate resource summary view	event
	Fortigate resource summary view a role, decimal(6, 0) decimal(6, 0) a decimal(6, 0) 10, 2) (10, 0) (10, 0)	
<pre>max(mem_peak) as mem_peak, max(disk_peak) as disk_peak, max(cpu_peak) as cpu_peak, cast(max(lograte_peak) as decim) as lograte_peak, max(session_peak) as session max(transmit_kbps_peak) as t from (select devid, slot, sum(total_cpu) / sum(cour sum(cour sum(cour) / sum(cour</pre>	nal(10, 2) n_peak, cransmit_kbps_peak nt) as cpu_ave,	

```
sum(total_disk) / sum(count) as disk_ave,
sum(
    total_trate + total_erate + total_orate
) / 100.00 / sum(count) as log_rate,
sum(totalsession) / sum(count) as sessions,
sum(sent) / sum(count) as sent_kbps,
sum(recv) / sum(count) as recv_kbps,
sum(sent + recv) / sum(count) as transmit_kbps,
max(mem_peak) as mem_peak,
max(disk_peak) as disk_peak,
max(cpu_peak) as cpu_peak,
max(lograte_peak) / 100.00 as lograte_peak,
max(session_peak) as session_peak,
max(transmit_peak) as transmit_kbps_peak
from
```

Dataset Name	Description	Log Category
perf-stat-usage-details-drilldown- master	Fortigate resource summary view	event
<pre>select devid, get_fgt_role(devid, slot) as cast(sum(cpu_ave) / count(*) as) as cpu_ave, cast(sum(mem_ave) / count(*) as) as mem_ave, cast(sum(disk_ave) / count(*) as) as disk_ave, cast(sum(log_rate) as decimal(1)) as log_rate, cast(sum(sessions) as decimal(1)) as sessions, cast(sum(sent_kbps) as decimal(2)) as sent_kbps,</pre>	<pre>decimal(6, 0) decimal(6, 0) decimal(6, 0) 0, 2) 0, 0)</pre>	

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

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

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

Dataset Name	Description	Log Category
incident-Incident-Count-by-Status- Donut	Incident status distribution	
<pre>select status, count(*) as cnt from</pre>		
\$incident where \$filter - drilldown		
group by status order by		
status		

Dataset Name	Description	Log Category
incident-Open-Incident-Count-T	imeline Incident count by status over time	
select		
<pre>\$flex_timescale(agg_time</pre>		
max(num_sta_draft) as nu		
<pre>max(num_sta_analysis) as max(num sta response) as</pre>		
max(num_sta_lesponse) as max(num_sta_closed) as m		
max(num sta cancelled) a		
from		
<pre>\$incident_history</pre>		
where		
\$filter - drilldown		
and \$cust_time_filter(ag group by	(g_time)	
hodex		
order by		
hodex		
Dataset Name	Description	Log Category

incident-Closed-Incident-Count-Timeline Incident count by status over time

```
select
 $flex_timescale(agg_time) as hodex,
 max(num_sta_draft) as num_sta_draft,
 max(num_sta_analysis) as num_sta_analysis,
 max(num_sta_response) as num_sta_response,
 max(num_sta_closed) as num_sta_closed,
 max(num sta cancelled) as num sta cancelled
from
 $incident_history
where
 $filter - drilldown
 and $cust_time_filter(agg_time)
group by
 hodex
order by
 hodex
```

Dataset Name	Description	Log Category
Top-10-Apps-by-Bandwidth	Top applications by bandwidth usage	traffic
<pre>select app_group, sum(bandwidth) as bandwidth, sum(traffic_in) as traffic_i sum(traffic_out) as traffic_ sum(sessions) as sessions</pre>	n,	
<pre>sentbyte, 0)+coalesce(rcvddelt rcvdbyte, 0)) as traffic_in, s as sessions from \$log where \$f</pre>	<pre>pp) as app_group, appcat, service, sum(co a, rcvdbyte, 0)) as bandwidth, sum(coale sum(coalesce(sentdelta, sentbyte, 0)) as ilter and (logflag&(1 32)>0) and nullifn ervice order by bandwidth desc)### t group</pre>	sce(rcvddelta, traffic_out, count(*) a(app) is not null

sum(bandwidth)>0 order by bandwidth desc

Dataset Name	Description	Log Category
Top-10-User-by-Bandwidth	Top users by bandwidth usage	traffic
<pre>select coalesce(nullifna(`user`), nullifna(`unauthuse ipstr(`srcip`)) as user_src, srcip, sum(coalesce(sentbyte,) as bandwidth, sum(coalesce(rcvdbyte,) as traffic_in, sum(coalesce(sentbyte,) as traffic_out from</pre>	0)+ coalesce(rcvdbyte, 0) 0)	

```
$loq
where
 $filter
 and (
  logflag&1>0
 )
 and srcip is not null
group by
 user_src,
 srcip
having
 sum(
   coalesce(sentbyte, 0) + coalesce(rcvdbyte, 0)
 ) > 0
order by
 bandwidth desc
```

Dataset Name	Description	Log Category
Top-10-Applications-by-Number-of- Jsers	Top Applications by number of users	traffic

select

```
app_group_name(app) as app_group,
    count(distinct user_src) as number
```

from

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

Dataset Name	Description	Log Category
Top-10-User-by-Session	Top user by session count	traffic
	user`), nullifna(`unauthuser`), ipstr(`srcip` where \$filter and (logflag&1>0) group by user_ er_src order by sessions desc	
Dataset Name	Description	Log Category
Top-10-Apps-by-Session	Top applications by bandwidth usage	traffic
<pre>sentbyte, 0)+coalesce(rcvddelta,</pre>		ddelta,

as sessions from $\log \$ app app group, appcat, service order by bandwidth desc)### t group by app group having sum(bandwidth)>0 order by bandwidth desc

Dataset Name	Description	Log Category
Applications-by-Risk-Level	Applications by Risk Level	traffic
<pre>(risk)='3' then 'Medium' wh (sessions) as sessions, sur bandwidth from ###(select a (coalesce(rcvdbyte, 0)) as bandwidth, count(*) as sess app, appcat order by bandwidth</pre>	pp_group, 039;5' then 'Critical' when max(risk)='4 hen max(risk)='2' then 'Low' else 'Info' h(sent) as sent, sum(received) as received appid, app, appcat, sum(coalesce(sentbyte received, sum(coalesce(sentbyte, 0)+coal sions from \$log where \$filter and (logfla bdth desc)### t1 inner join app_mdata t2 up, appcat order by d_risk desc, bandwidd	<pre>end) as risk_level, sum ed, sum(bandwidth) as e, 0)) as sent, sum lesce(rcvdbyte, 0)) as ag&1>0) group by appid, on lower(t1.app)=lower</pre>
Dataset Name	Description	Log Category
soc-Event-vs-Incident-Today-Tre	nd Events vs Incidents Today Trend	
<pre>(select count(*) from \$ever drilldown and \$cust_time_fi :1 left join devtable t2 or (alerttime,YESTERDAY)) as r (num_cur-num_pre) as num_di filter(createtime,TODAY)) a</pre>	em, num_cur, num_pre, (num_cur-num_pre) a nt t1 left join devtable t2 on t1.dvid=t2 llter(alerttime,TODAY)) as num_cur, (sele n t1.dvid=t2.dvid where \$filter-drilldown num_pre) t union all select 'Incidents' a lff from (select (select count(*) from \$ as num_cur, (select count(*) from \$incide ()) as num_pre) t) t order by item	2.dvid where \$filter- ect count(*) from \$event h and \$cust_time_filter as item, num_cur, num_pre incident where \$cust_time
Dataset Name	Description	Log Category
soc-Event-vs-Incident-History-Tre	end Events vs Incidents History Trend	
select item, num_cur, num_pre, num_diff from (

```
select
```

& #039;Events' as item, num_cur, num_pre, (num_cur-num_pre) as num_diff from (select (select count(*) from \$event t1 left join devtable t2 on t1.dvid=t2.dvid where \$filterdrilldown and \$cust_time_filter(alerttime)) as num_cur, (select count(*) from \$event t1 left join devtable t2 on t1.dvid=t2.dvid where \$filter-drilldown and \$cust_time_filter (alerttime,LAST_N_PERIOD,1)) as num_pre) t union all select 'Incidents' as item, num_cur, num_pre, (num_cur-num_pre) as num_diff from (select (select count(*) from \$incident where \$cust_time_filter(createtime)) as num_cur, (select count(*) from \$incident where \$cust_time_filter(createtime,LAST_N_PERIOD,1)) as num_pre) t) t order by item

Dataset Name	Description	Log Category
soc-Event-vs-Incident-Trend	Events vs Incidents Trend	
(select count(*) from \$event t1 drilldown and \$cust_time_filter t1 left join devtable t2 on t1.c (alerttime,YESTERDAY)) as num_pr (num_cur-num_pre) as num_diff fr filter(createtime,TODAY)) as num filter(createtime,YESTERDAY)) as num_pre, (num_cur-num_pre) as nu join devtable t2 on t1.dvid=t2.c (alerttime)) as num_cur, (select t1.dvid=t2.dvid where \$filter-dr num_pre) t union all select 'Inc diff from (select (select count)		<pre>vid where \$filter- count(*) from \$event nd \$cust_time_filter item, num_cur, num_pre, ident where \$cust_time_ where \$cust_time_ ents' as item, num_cur, from \$event t1 left _time_filter vtable t2 on me,LAST_N_PERIOD,1)) as um_cur-num_pre) as num_ lter(createtime)) as</pre>

Dataset Name	Description	Log Category
soc-Total-Event-by-Severity	Total Events by Severity	
3 THEN 'Low' ELSE NULL END)		EN 'High' WHEN 2 THEN 'Medium' WHEN ts from \$event t1 left join devtable everity order by severity
Dataset Name	Description	Log Category

and Total Event by Coverity I listomy	-
soc-Total-Event-by-Severity-History	

Total Events by Severity History

select dom, (CASE severity WHEN 0 THEN & #039;Critical' WHEN 1 THEN 'High' WHEN 2 THEN 'Medium' WHEN 3 THEN 'Low' ELSE NULL END) as sev, sum(num_events) as num_events from (select dom, unnest (agg_sev) as severity, unnest(agg_num) as num_events from (select \$DAY_OF_MONTH(agg_time) as dom, array[0, 1, 2, 3] as agg_sev, array[max(num_sev_critical), max(num_sev_high), max(num_ sev_medium), max(num_sev_low)] as agg_num from \$event_history where \$filter-drilldown and \$cust_time_filter(agg_time) group by dom order by dom) t) t group by dom, severity order by dom, severity

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

select

(

CASE severity WHEN 0 THEN & #039;Critical' WHEN 1 THEN 'High' WHEN 2 THEN 'Medium' WHEN 3 THEN 'Low' ELSE NULL END) as sev, triggername, count(*) as num_events from \$event t1 left join devtable t2 on t1.dvid=t2.dvid where \$filter-drilldown group by severity, triggername order by severity, triggername

Dataset Name	Description	Log Category
soc-Total-Incident-by-Severity	Total Incidents by Severity	
<pre>select severity, count(*) as num_inc from \$incident where \$filter - drilldown group by severity order by severity</pre>		
Dataset Name	Description	Log Category
soc-Total-Event-vs-Incident-History	Total Events vs Incidents History	
<pre>select coalesce(t1.hodex, t2.hodex) { coalesce(num_event_total, 0) { coalesce(num_inc_total, 0) as coalesce(num_event_high, 0) a from (select \$flex_timescale(agg_time) max(num_total) as num_even max(num_sev_critical + num from \$event_history where</pre>	as num_event_total, num_inc_total, s num_event_high as hodex,	

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

Dataset Name	Description	Log Category
soc-Incident-List	List of Incidents	
<pre>select incid_to_str(incid) as from_itime(createtime) inc_cat_encode(category severity, status, endpoint from \$incident where \$cust_time_filter(creat order by createtime desc</pre>	as timestamp,) as category,	
Dataset Name	Description	Log Category
soc-Incident-by-Severity	Incidents by Severity	
<pre>select severity, count(*) as incnum from \$incident where \$filter - drilldown and \$cust_time_filter(c group by severity</pre>	reatetime)	

order by incnum desc

Dataset Name	Description	Log Category
soc-Incident-by-Status	Incidents by Status	
<pre>select status, count(*) as incnum from \$incident where \$filter - drilldown and \$cust_time_filter(createt: group by status order by incnum desc</pre>		
Dataset Name	Description	Log Category
soc-Incident-by-Category-Unresolved	Unresolved Incidents by Category	
select		
<pre>inc_cat_encode(category) as ca count(*) as incnum from \$incident where \$filter - drilldown and \$cust_time_filter(createt: and status not in (</pre>		
<pre>inc_cat_encode(category) as ca count(*) as incnum from \$incident where \$filter - drilldown and \$cust_time_filter(createt: and status not in (</pre>	ime)	Log Category
<pre>inc_cat_encode(category) as ca count(*) as incnum from \$incident where \$filter - drilldown and \$cust_time_filter(createt: and status not in (& #039;closed', 'cancelled')</pre>	ime)) group by category order by incnum desc	Log Categor
<pre>inc_cat_encode(category) as ca count(*) as incnum from \$incident where \$filter - drilldown and \$cust_time_filter(createt: and status not in (& #039;closed', 'cancelled') Dataset Name soc-Incident-by-Severity-Unresolved select severity, count(*) as incnum from \$incident where \$filter - drilldown and \$cust_time_filter(createt: and status not in (</pre>	ime)) group by category order by incnum desc Description Unresolved Incidents by Severity	Log Category
<pre>inc_cat_encode(category) as ca count(*) as incnum from \$incident where \$filter - drilldown and \$cust_time_filter(createt: and status not in (& #039;closed', 'cancelled') Dataset Name soc-Incident-by-Severity-Unresolved select severity, count(*) as incnum from \$incident where \$filter - drilldown and \$cust_time_filter(createt: and status not in (</pre>	ime)) group by category order by incnum desc Description Unresolved Incidents by Severity	Log Category

```
$flex_timescale(agg_time) as hodex,
```

```
max(num_cat_cat1) as num_cat1,
max(num_cat_cat2) as num_cat2,
max(num_cat_cat3) as num_cat3,
max(num_cat_cat4) as num_cat4,
max(num_cat_cat5) as num_cat5,
max(num_cat_cat6) as num_cat6
from
   $incident_history
where
   $filter - drilldown
   and $cust_time_filter(agg_time)
group by
   hodex
order by
   hodex
```

Dataset Name	Description	Log Category
soc-Incident-List-Unresolved	List of Unresolved Incidents	
<pre>select incid_to_str(incid) as inc from_itime(createtime) as severity, status, endpoint, description from \$incident where \$filter - drilldown and \$cust_time_filter(crea and status not in (& #039;closed', 'cancell</pre>	timestamp,	
Dataset Name	Description	Log Category
fex-RSRQ-timeline	FortiExtender RSRQ timeline	event
<pre>select \$flex_timescale(timestamp) as hodex, cast(sum(rsrq_sum) / sum(count) as decimal(18, 2)) & #039;dB' as rsrq from ###(select \$flex_timestamp(dtime) as timestamp, sum(to_number (rsrq, '999999.99')) as rsrq_sum, sum(to_number(sinr, '999999.99')) as sinr_sum, count(*) as count from \$log where \$filter and logid='0111046409' group by timestamp order by timestamp desc)### t group by hodex order by hodex desc</pre>		

Dataset Name	Description	Log Category
fex-SINR-timeline	FortiExtender SINR timeline	event
<pre>select \$flex_timescale(timestat cast(sum(sinr_sum)/ sum(condition))</pre>	mp) as hodex, unt) as decimal(18, 0)	

) || & #039;dB' as sinr from ###(select \$flex_timestamp(dtime) as timestamp, sum(to_number (rsrq, '999999.99')) as rsrq_sum, sum(to_number(sinr, '999999.99')) as sinr_sum, count(*) as count from \$log where \$filter and logid='0111046409' group by timestamp order by timestamp desc)### t group by hodex order by hodex desc

Dataset Name	Description	Log Category
fgt-device-monitoring-inventory	FortiGate Device Monitoring Inventory	event

select

devname,

(

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

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

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

select

& #039;FortiOS' as sf_name, (platform || ' ' || os) as firmware, count(*) as total_num from \$func-fgt-inventory as t1 where exists (select 1 from devtable t2 where \$dev_filter and t2.devid=t1.devid) group by platform, os order by total num desc

Dataset Name	Description	Log Category
cup-utilization-timeline-for-each-device	FortiGate cpu utilization timeline	event
<pre>select \$flex timescale(timestamp) as 1</pre>	hodex,	

devid,

```
cast(
   sum(total_cpu) / sum(count) as decimal(6, 0)
) as cpu_ave,
cast(
   sum(total_mem) / sum(count) as decimal(6, 0)
) as mem_ave,
cast(
   sum(total_disk) / sum(count) as decimal(6, 0)
) as disk_ave,
cast(
   sum(sent) / sum(count) as decimal(10, 0)
) as sent_kbps,
cast(
   sum(recv) / sum(count) as decimal(10, 0)
) as recv_kbps
```

```
from
```

Dataset Name	Description	Log Category
status-timeline-by-device-cpu- utilization	FortiGate cpu summary view	event
<pre>select devid, cast(sum(total_cpu) / sum(count)) as cpu_ave, max(cpu_peak) as cpu_peak from ###(select Sfley timestamp as</pre>	as decimal(6, 0) timestamp, devid, slot, sum(coalesce	(trate 0)) as total
<pre>trate, sum(coalesce(erate, 0)) (itime) as first_seen, max(itim (coalesce(mem, 0)) as mem_peak, as disk_peak, sum(coalesce(cpu, (coalesce(trate, 0)+coalesce(erate))</pre>	<pre>as total_erate, sum(coalesce(orate, 0) sum(coalesce(disk, 0)) as total_disk, 0)) as total_cpu, max(coalesce(cpu, (rate, 0)+coalesce(orate, 0)) as lograte ion, max(coalesce(totalsession, 0)) as</pre>)) as total_orate, min) as total_mem, max , max(coalesce(disk, 0))))) as cpu_peak, max e_peak, sum(coalesce
<pre>(coalesce(split_part(bandwidth, part(bandwidth, '/', 2), '0') a '/', 1), '0') as integer)+cast(transmit_peak, sum(coalesce(set))</pre>	<pre>'/', 1), '0') as integer)) as sent, s '/', 1), '0') as recv, max(cast(coalesce coalesce(split_part(bandwidth, '/', 2) uprate, 0)) as cps, max(coalesce(setup re \$filter and subtype='system' and ac</pre>	<pre>sum(cast(coalesce(split_ e(split_part(bandwidth,), '0') as integer)) as prate, 0)) as cps_peak,</pre>

by timestamp, devid, slot order by total_mem desc)### t group by devid order by cpu_peak desc

Dataset Name	Description	Log Category
event-cpu-utilization-dev	FortiGate cpu summary view	event
<pre>select devid, cast(sum(total_cpu) / sum(count)) as cpu_ave, max(cpu_peak) as cpu_peak</pre>	as decimal(6, 0)	
<pre>from ###(select \$flex timestamp a</pre>	s timestamp, devid, slot, sum(coalesc	ce(trate, 0)) as total
<pre>trate, sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min (itime) as first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max</pre>		

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

Dataset Name	Description	Log Category
memory-utilization-timeline-for-each- device	FortiGate cpu utilization timeline	event
<pre>select \$flex_timescale(timestamp) as devid, cast(sum(total_cpu)/ sum(count) a) as cpu_ave, cast(sum(total_mem)/ sum(count) a) as mem_ave, cast(sum(total_disk)/ sum(count)) as disk_ave, cast(sum(sent)/ sum(count) as dec) as sent_kbps, cast(sum(recv)/ sum(count) as dec) as recv_kbps from</pre>	as decimal(6, 0) as decimal(6, 0) as decimal(6, 0) cimal(10, 0)	
<pre>###(select \$flex_timestamp as trate, sum(coalesce(erate, 0)) a</pre>	<pre>timestamp, devid, slot, sum(coales as total_erate, sum(coalesce(orate) e) as last_seen, sum(coalesce(mem,</pre>	, 0)) as total_orate, min

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

Dataset Name	Description	Log Category
status-timeline-by-device-mem- utilization	FortiGate memory summary view	event
<pre>trate, sum(coalesce(erate, 0)) a (itime) as first_seen, max(itime (coalesce(mem, 0)) as mem_peak, a as disk_peak, sum(coalesce(cpu, (coalesce(trate, 0)+coalesce(erad (totalsession, 0)) as totalsessio (coalesce(split_part(bandwidth,</pre>	<pre>s decimal(6, 0) timestamp, devid, slot, sum(coalesce(trate, 0) s total_erate, sum(coalesce(orate, 0)) as tota) as last_seen, sum(coalesce(mem, 0)) as tota; sum(coalesce(disk, 0)) as total_disk, max(coalesce)) as total_cpu, max(coalesce(cpu, 0)) as cpu te, 0)+coalesce(orate, 0)) as lograte_peak, su on, max(coalesce(totalsession, 0)) as session_ '/', 1), '0') as integer)) as sent, sum(cast(coalesce(split_pa))) </pre>	al_orate, min l_mem, max lesce(disk, 0)) a_peak, max um(coalesce _peak, sum(cast coalesce(split_
transmit_peak, sum(coalesce(setu)	<pre>balesce(split_part(bandwidth, '/', 2), '0') as prate, 0)) as cps, max(coalesce(setuprate, 0)) e \$filter and subtype='system' and action='per</pre>	as cps_peak,

Dataset Name	Description	Log Category
event-mem-utilization-dev	FortiGate memory summary view	event
<pre>select devid, cast(sum(total_mem) / sum(coun) as mem_ave, max(mem_peak) as mem_peak</pre>	t) as decimal(6, 0)	
<pre>from ###(select \$flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_ trate, sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min (itime) as first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max (coalesce(mem, 0)) as mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum(coalesce(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max (coalesce(trate, 0)+coalesce(erate, 0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce</pre>		

by timestamp, devid, slot order by total mem desc)### t group by devid order by mem peak

desc

(totalsession, 0)) as totalsession, max(coalesce(totalsession, 0)) as session_peak, sum(cast (coalesce(split_part(bandwidth, '/', 1), '0') as integer)) as sent, sum(cast(coalesce(split_ part(bandwidth, '/', 2), '0') as integer)) as recv, max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max(coalesce(setuprate, 0)) as cps_peak, count(*) as count from \$log where \$filter and subtype='system' and action='perf-stats' group by timestamp, devid, slot order by total_mem desc)### t group by devid order by mem_peak desc

Dataset Name	Description	Log Category
disk-utilization-timeline-for-each- device	FortiGate cpu utilization timeline	event
<pre>select \$flex_timescale(timestamp) devid, cast(sum(total_cpu) / sum(count) as cpu_ave, cast(sum(total_mem) / sum(count) as mem_ave, cast(sum(total_disk) / sum(count) as disk_ave, cast(sum(sent) / sum(count) as) as sent_kbps, cast(sum(recv) / sum(count) as) as recv kbps</pre>) as decimal(6, 0)) as decimal(6, 0) t) as decimal(6, 0) decimal(10, 0)	
<pre>trate, sum(coalesce(erate, 0) (itime) as first_seen, max(it (coalesce(mem, 0)) as mem_pea as disk_peak, sum(coalesce(cp (coalesce(trate, 0)+coalesce((totalsession, 0)) as totalse (coalesce(split_part(bandwidt part(bandwidth, '/', 2), '0') '/', 1), '0') as integer)+cas transmit_peak, sum(coalesce(s count(*) as count from \$log w</pre>	as timestamp, devid, slot, sum(coalesce) as total_erate, sum(coalesce(orate, 0 ime) as last_seen, sum(coalesce(mem, 0) k, sum(coalesce(disk, 0)) as total_disk u, 0)) as total_cpu, max(coalesce(cpu, erate, 0)+coalesce(orate, 0)) as lograte ssion, max(coalesce(totalsession, 0)) a h, '/', 1), '0') as integer)) as sent, as integer)) as recv, max(cast(coalesce t(coalesce(split_part(bandwidth, '/', 2 etuprate, 0)) as cps, max(coalesce(setup here \$filter and subtype='system' and as er by total_mem desc)### t where \$filter	<pre>)) as total_orate, min) as total_mem, max , max(coalesce(disk, 0)) 0)) as cpu_peak, max e_peak, sum(coalesce s session_peak, sum(cast sum(cast(coalesce(split_ e(split_part(bandwidth,), '0') as integer)) as prate, 0)) as cps_peak, ction='perf-stats' group</pre>

Dy CIM	, country	act t	~, .	5100	Oracr	2
hodex,	devid	order	by	hode	ex	

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

```
sum(total_disk) / sum(count) as decimal(6, 0)
) as disk_ave,
max(disk_peak) as disk_peak
from
```

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

Dataset Name	Description	Log Category
event-disk-utilization-dev	FortiGate disk summary view	event
select devid,		

```
devid,
cast(
   sum(total_disk) / sum(count) as decimal(6, 0)
) as disk_ave,
   max(disk_peak) as disk_peak
from
```

```
Dataset NameDescriptionLog Categoryevent-total-session-summaryFortiGate Total Sessionseventselect<br/>devid,<br/>max(session_peak) as max_session,<br/>cast(<br/>sum(totalsession)/ sum(count) as decimal(10, 0)<br/>) as sessions,<br/>max(cps_peak) as cps_peak,event
```

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

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

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

select
 devid,
 max(cps_peak) as max_rate
from

Dataset Name	Description	Log Category
event-session-summary-dev	FortiGate Total Sessions	event
<pre>select devid, max(session_peak) as max_session cast(sum(totalsession) / sum(count)) as sessions, max(cps_peak) as cps_peak, cast(sum(cps) / sum(count) as decire) as cps_ave</pre>) as decimal(10, 0)	

from

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

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

select

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

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

Dataset Name	Description	Log Category
fgt-intf-down-timeline-by-device	FortiGate Interface Down by Device	event
select		
devid,		
status,		
sum(total_num) as total_num from	L	
where \$filter and logid_to_in	as timestamp, devid, status, count(*) as t(logid)=20099 and status='DOWN' group by status order by total_num desc	—
Dataset Name	Description	Log Category
Dataset Name fgt-intf-down-dev-donut	Description FortiGate Interface Down by Device	Log Category event
fgt-intf-down-dev-donut	·	
fgt-intf-down-dev-donut	·	
fgt-intf-down-dev-donut select devid, status,	FortiGate Interface Down by Device	
<pre>fgt-intf-down-dev-donut select devid, status, sum(total_num) as total_num</pre>	FortiGate Interface Down by Device	
<pre>fgt-intf-down-dev-donut select devid, status, sum(total_num) as total_num from</pre>	FortiGate Interface Down by Device	event
<pre>fgt-intf-down-dev-donut select devid, status, sum(total_num) as total_num from ###(select \$flex_timestamp</pre>	FortiGate Interface Down by Device	event total_num from \$log

status)### t group by devid, status order by total_num desc

Dataset Name	Description	Log Category
fgt-intf-down-dev-tbl	FortiGate Interface Down by Device	event
<pre>select devid, status, sum(total_num) as total_num from</pre>		
	as timestamp, devid, status, count(*) as	total_num from \$log

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

Dataset Name	Description	Log Category
intf-sent-timeline-for-each-device	FortiGate cpu utilization timeline	event
<pre>select \$flex_timescale(timestamp) a devid, cast(sum(total_cpu)/ sum(count)) as cpu_ave, cast(sum(total_mem)/ sum(count)) as mem_ave, cast(sum(total_disk)/ sum(count) as disk_ave, cast(sum(sent)/ sum(count) as d) as sent_kbps, cast(sum(recv)/ sum(count) as d) as recv_kbps from</pre>	as decimal(6, 0) as decimal(6, 0)) as decimal(6, 0) ecimal(10, 0)	
<pre>trate, sum(coalesce(erate, 0)) (itime) as first_seen, max(itin (coalesce(mem, 0)) as mem_peak as disk_peak, sum(coalesce(cpu (coalesce(trate, 0)+coalesce(e (totalsession, 0)) as totalses (coalesce(split_part(bandwidth part(bandwidth, '/', 2), '0') '/', 1), '0') as integer)+cast transmit_peak, sum(coalesce(se count(*) as count from \$log wh</pre>	<pre>s timestamp, devid, slot, sum(coalesce as total_erate, sum(coalesce(orate, 0 me) as last_seen, sum(coalesce(mem, 0) , sum(coalesce(disk, 0)) as total_disk , 0)) as total_cpu, max(coalesce(cpu, rate, 0)+coalesce(orate, 0)) as lograt sion, max(coalesce(totalsession, 0)) a , '/', 1), '0') as integer)) as sent, as integer)) as recv, max(cast(coalesce (coalesce(split_part(bandwidth, '/', 2 tuprate, 0)) as cps, max(coalesce(setu ere \$filter and subtype='system' and a r by total_mem desc)### t where \$filter</pre>	<pre>D)) as total_orate, min)) as total_mem, max k, max(coalesce(disk, 0)) 0)) as cpu_peak, max te_peak, sum(coalesce as session_peak, sum(cast sum(cast(coalesce(split_ ce(split_part(bandwidth, 2), '0') as integer)) as uprate, 0)) as cps_peak, action='perf-stats' group</pre>

Dataset Name	Description	Log Category
status-timeline-by-device-intf-sent	FortiGate interface summary view	event

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

Description	Log Category
FortiGate cpu utilization timeline	event
<pre>nodex, s decimal(6, 0) s decimal(6, 0) as decimal(6, 0) imal(10, 0) imal(10, 0) cimestamp, devid, slot, sum(coalesce s total_erate, sum(coalesce(orate, (0) a s last_seen, sum(coalesce(mem, 0))</pre>	e(trate, 0)) as total_ 0)) as total_orate, min)) as total_mem, max
	<pre>FortiGate cpu utilization timeline nodex, s decimal(6, 0) s decimal(6, 0) as decimal(6, 0) imal(10, 0) timestamp, devid, slot, sum(coalesce) s total_erate, sum(coalesce(orate, b))</pre>

(coalesce(trate, 0)+coalesce(erate, 0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce (totalsession, 0)) as totalsession, max(coalesce(totalsession, 0)) as session_peak, sum(cast (coalesce(split_part(bandwidth, '/', 1), '0') as integer)) as sent, sum(cast(coalesce(split_ part(bandwidth, '/', 2), '0') as integer)) as recv, max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max(coalesce(setuprate, 0)) as cps_peak, count(*) as count from \$log where \$filter and subtype='system' and action='perf-stats' group by timestamp, devid, slot order by total_mem desc)### t where \$filter-drilldown group by hodex, devid order by hodex

Dataset Name	Description	Log Category
status-timeline-by-device-intf-recv	FortiGate interface summary view	event
<pre>select devid, cast(sum(sent)/ sum(count) as dev) as sent_kbps, cast(sum(recv)/ sum(count) as dev) as recv_kbps, cast(sum(sent + recv)/ sum(count) as transmit_kbps,</pre>	cimal(10, 0)	
<pre>max(transmit_peak) as transmi from</pre>	t_kbps_peak	
<pre>from ###(select \$flex_timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total_ trate, sum(coalesce(erate, 0)) as total_erate, sum(coalesce(orate, 0)) as total_orate, min (itime) as first_seen, max(itime) as last_seen, sum(coalesce(mem, 0)) as total_mem, max (coalesce(mem, 0)) as mem_peak, sum(coalesce(disk, 0)) as total_disk, max(coalesce(disk, 0)) as disk_peak, sum(coalesce(cpu, 0)) as total_cpu, max(coalesce(cpu, 0)) as cpu_peak, max (coalesce(trate, 0)+coalesce(erate, 0)+coalesce(orate, 0)) as lograte_peak, sum(coalesce (totalsession, 0)) as totalsession, max(coalesce(totalsession, 0)) as session_peak, sum(case(coalesce(split_part(bandwidth, '/', 1), '0') as integer)) as sent, sum(cast(coalesce(split_bandwidth, '/', 1), '0'))</pre>		

part(bandwidth, '/', 2), '0') as integer)) as recv, max(cast(coalesce(split_part(bandwidth, '/', 1), '0') as integer)+cast(coalesce(split_part(bandwidth, '/', 2), '0') as integer)) as transmit_peak, sum(coalesce(setuprate, 0)) as cps, max(coalesce(setuprate, 0)) as cps_peak, count(*) as count from \$log where \$filter and subtype='system' and action='perf-stats' group by timestamp, devid, slot order by total_mem desc)### t group by devid order by transmit_kbps peak desc

Dataset Name	Description	Log Category
event-intf-summary-dev	FortiGate interface summary view	event
<pre>select devid, cast(sum(sent)/ sum(count) as de) as sent_kbps, cast(sum(recv)/ sum(count) as de) as recv_kbps, cast(sum(sent + recv)/ sum(count)</pre>	cimal(10, 0)	

```
) as transmit kbps,
 max(transmit_peak) as transmit_kbps_peak
from
  ####(select $flex timestamp as timestamp, devid, slot, sum(coalesce(trate, 0)) as total
trate, sum(coalesce(erate, 0)) as total erate, sum(coalesce(orate, 0)) as total orate, min
(itime) as first seen, max(itime) as last seen, sum(coalesce(mem, 0)) as total mem, max
(coalesce(mem, 0)) as mem peak, sum(coalesce(disk, 0)) as total disk, max(coalesce(disk, 0))
as disk peak, sum(coalesce(cpu, 0)) as total cpu, max(coalesce(cpu, 0)) as cpu peak, max
(coalesce(trate, 0)+coalesce(erate, 0)+coalesce(orate, 0)) as lograte peak, sum(coalesce
(totalsession, 0)) as totalsession, max(coalesce(totalsession, 0)) as session peak, sum(cast
(coalesce(split part(bandwidth, '/', 1), '0') as integer)) as sent, sum(cast(coalesce(split
part(bandwidth, '/', 2), '0') as integer)) as recv, max(cast(coalesce(split part(bandwidth,
'/', 1), '0') as integer)+cast(coalesce(split part(bandwidth, '/', 2), '0') as integer)) as
transmit peak, sum(coalesce(setuprate, 0)) as cps, max(coalesce(setuprate, 0)) as cps peak,
count (*) as count from $log where $filter and subtype='system' and action='perf-stats' group
by timestamp, devid, slot order by total mem desc)### t group by devid order by transmit
kbps peak desc
```

Dataset Name	Description	Log Category
fgt-intf-stats-timeline-util-in-each	FortiGate Interface Statistics Timeline	event

select

\$flex_timescale(tmstamp) as hodex,

devname || & #039;:' || intfname) as dev_intf, cast(sum(bps_out)/sum(interval)/1000 as decimal(10, 0)) as kbps_out_avg, cast(sum(bps_in)/sum(interval)/1000 as decimal(10, 0)) as kbps_in_avg, cast(sum(util_out)/sum(interval)/100 as decimal(10, 2)) as util_out_avg, cast (sum(util_in)/sum(interval)/100 as decimal(10, 2)) as util_in_avg from (select \$flex_ timestamp(timestamp) as tmstamp, dvid, intfname, sum(interval) as interval, sum (sentbps*interval) as bps_out, sum(rcvdbps*interval) as bps_in, sum(sentutil*interval) as util_out, sum(rcvdutil*interval) as util_in from intfstats where \$cust_time_filter (timestamp) group by tmstamp, dvid, intfname) t1 left join devtable t2 on t1.dvid = t2.dvid where \$filter-drilldown group by hodex, dev intf order by hodex

Dataset NameDescriptionLog Categoryfgt-intf-stats-timeline-util-inFortiGate Interface Received Utilizationevent

select

devname || & #039;:' || intfname) as dev_intf, cast(sum(bps_out)/sum(interval)/1000 as decimal(10, 0)) as kbps_out_avg, cast(sum(bps_in)/sum(interval)/1000 as decimal(10, 0)) as kbps_in_avg, cast(sum(util_out)/sum(interval)/100 as decimal(10, 2)) as util_out_avg, cast (sum(util_in)/sum(interval)/100 as decimal(10, 2)) as util_in_avg from (select \$flex_ timestamp(timestamp) as tmstamp, tbl_intf.dvid, intfname, sum(interval) as interval, sum (sentbps*interval) as bps_out, sum(rcvdbps*interval) as bps_in, sum(sentutil*interval) as util_out, sum(rcvdutil*interval) as util_in from (select distinct dvid from ###(select dvid from \$log-event where \$filter and action='perf-stats' group by dvid)### t) tbl_log inner join intfstats tbl_intf.dvid, intfname) t1 left join devtable t2 on t1.dvid = t2.dvid group by dev_intf order by util_in_avg desc, kbps_in_avg desc, kbps_out_avg desc

Dataset Name	Description	Log Category
fgt-intf-stats-timeline-util-out-each	FortiGate Interface Statistics Timeline	event

```
select
   $flex_timescale(tmstamp) as hodex,
   (
      devname || & #039;:' || intfname) as dev_intf, cast(sum(bps_out)/sum(interval)/1000 as
decimal(10, 0)) as kbps_out_avg, cast(sum(bps_in)/sum(interval)/1000 as decimal(10, 0)) as
kbps_in_avg, cast(sum(util_out)/sum(interval)/100 as decimal(10, 2)) as util_out_avg, cast
   (sum(util_in)/sum(interval)/100 as decimal(10, 2)) as util_in_avg from (select $flex_
   timestamp(timestamp) as tmstamp, dvid, intfname, sum(interval) as interval, sum
   (sentbps*interval) as bps_out, sum(rcvdbps*interval) as bps_in, sum(sentutil*interval) as
   util_out, sum(rcvdutil*interval) as util_in from intfstats where $cust_time_filter
   (timestamp) group by tmstamp, dvid, intfname) t1 left join devtable t2 on t1.dvid = t2.dvid
   where $filter-drilldown group by hodex, dev_intf order by hodex
```

Dataset Name	Description	Log Category
fgt-intf-stats-timeline-util-out	FortiGate Interface Sent Utilization	event

select

devname || & #039;:' || intfname) as dev_intf, cast(sum(bps_out)/sum(interval)/1000 as decimal(10, 0)) as kbps_out_avg, cast(sum(bps_in)/sum(interval)/1000 as decimal(10, 0)) as kbps_in_avg, cast(sum(util_out)/sum(interval)/100 as decimal(10, 2)) as util_out_avg, cast (sum(util_in)/sum(interval)/100 as decimal(10, 2)) as util_in_avg from (select \$flex_ timestamp(timestamp) as tmstamp, tbl_intf.dvid, intfname, sum(interval) as interval, sum (sentbps*interval) as bps_out, sum(rcvdbps*interval) as bps_in, sum(sentutil*interval) as util_out, sum(rcvdutil*interval) as util_in from (select distinct dvid from ###(select dvid from \$log-event where \$filter and action='perf-stats' group by dvid)### t) tbl_log inner join intfstats tbl_intf.dvid, intfname) t1 left join devtable t2 on t1.dvid = t2.dvid group by dev_intf order by util_out_avg desc, kbps_out_avg desc, kbps_in_avg desc

Dataset Name	Description	Log Category
fgt-intf-stats-timeline-bit-rate-in-each	FortiGate Interface Statistics Timeline	event
<pre>decimal(10, 0)) as kbps_out_avg kbps_in_avg, cast(sum(util_out) (sum(util_in)/sum(interval)/100 timestamp(timestamp) as tmstamp (sentbps*interval) as bps_out, util_out, sum(rcvdutil*interval (timestamp) group by tmstamp, d</pre>	odex, fname) as dev_intf, cast(sum(bps_out)/s , cast(sum(bps_in)/sum(interval)/1000 as /sum(interval)/100 as decimal(10, 2)) as as decimal(10, 2)) as util_in_avg from , dvid, intfname, sum(interval) as interval, sum(rcvdbps*interval) as bps_in, sum(se) as util_in from intfstats where \$cust vid, intfname) t1 left join devtable t2 y hodex, dev_intf order by hodex	as decimal(10, 0)) as as util_out_avg, cast n (select \$flex_ erval, sum entutil*interval) as t_time_filter
Dataset Name	Description	Log Category

Dataset Name	Description	Log Category
fgt-intf-stats-timeline-bit-rate-in	FortiGate Interface Received Bit Rate	event

select

devname || & #039;:' || intfname) as dev_intf, cast(sum(bps_out)/sum(interval)/1000 as decimal(10, 0)) as kbps_out_avg, cast(sum(bps_in)/sum(interval)/1000 as decimal(10, 0)) as kbps_in_avg, cast(sum(util_out)/sum(interval)/100 as decimal(10, 2)) as util_out_avg, cast (sum(util_in)/sum(interval)/100 as decimal(10, 2)) as util_in_avg from (select \$flex_ timestamp(timestamp) as tmstamp, tbl_intf.dvid, intfname, sum(interval) as interval, sum (sentbps*interval) as bps_out, sum(rcvdbps*interval) as bps_in, sum(sentutil*interval) as util_out, sum(rcvdutil*interval) as util_in from (select distinct dvid from ###(select dvid from \$log-event where \$filter and action='perf-stats' group by dvid)### t) tbl_log inner join intfstats tbl_intf on tbl_log.dvid = tbl_intf.dvid where \$cust_time_filter(timestamp) group by tmstamp, tbl_intf.dvid, intfname) t1 left join devtable t2 on t1.dvid = t2.dvid group by dev_intf order by kbps_in_avg desc

Dataset Name	Description	Log Category
fqt-intf-stats-timeline-bit-rate-out-each	FortiGate Interface Statistics Timeline	event

select

\$flex timescale(tmstamp) as hodex,

```
(
```

devname || & #039;:' || intfname) as dev_intf, cast(sum(bps_out)/sum(interval)/1000 as decimal(10, 0)) as kbps_out_avg, cast(sum(bps_in)/sum(interval)/1000 as decimal(10, 0)) as kbps_in_avg, cast(sum(util_out)/sum(interval)/100 as decimal(10, 2)) as util_out_avg, cast (sum(util_in)/sum(interval)/100 as decimal(10, 2)) as util_in_avg from (select \$flex_ timestamp(timestamp) as tmstamp, dvid, intfname, sum(interval) as interval, sum (sentbps*interval) as bps_out, sum(rcvdbps*interval) as bps_in, sum(sentutil*interval) as util_out, sum(rcvdutil*interval) as util_in from intfstats where \$cust_time_filter (timestamp) group by tmstamp, dvid, intfname) t1 left join devtable t2 on t1.dvid = t2.dvid where \$filter-drilldown group by hodex, dev_intf order by hodex

Dataset Name	Description	Log Category
fgt-intf-stats-timeline-bit-rate-out	FortiGate Interface Sent Bit Rate	event

select

(

devname || & #039;:' || intfname) as dev_intf, cast(sum(bps_out)/sum(interval)/1000 as decimal(10, 0)) as kbps_out_avg, cast(sum(bps_in)/sum(interval)/1000 as decimal(10, 0)) as kbps_in_avg, cast(sum(util_out)/sum(interval)/100 as decimal(10, 2)) as util_out_avg, cast (sum(util_in)/sum(interval)/100 as decimal(10, 2)) as util_in_avg from (select \$flex_ timestamp(timestamp) as tmstamp, tbl_intf.dvid, intfname, sum(interval) as interval, sum (sentbps*interval) as bps_out, sum(rcvdbps*interval) as bps_in, sum(sentutil*interval) as util_out, sum(rcvdutil*interval) as util_in from (select distinct dvid from ###(select dvid from \$log-event where \$filter and action='perf-stats' group by dvid)### t) tbl_log inner join intfstats tbl_intf.dvid, intfname) t1 left join devtable t2 on t1.dvid = t2.dvid group by dev_intf order by kbps_out_avg desc

Dataset Name	Description	Log Category
fgt-intf-stats-summary-view	FortiGate Interface Received Utilization	event

```
select
```

(

devname || & #039;:' || intfname) as dev_intf, cast(sum(bps_out)/sum(interval)/1000 as decimal(10, 0)) as kbps_out_avg, cast(sum(bps_in)/sum(interval)/1000 as decimal(10, 0)) as kbps_in_avg, cast(sum(util_out)/sum(interval)/100 as decimal(10, 2)) as util_out_avg, cast (sum(util_in)/sum(interval)/100 as decimal(10, 2)) as util_in_avg from (select \$flex_ timestamp(timestamp) as tmstamp, tbl intf.dvid, intfname, sum(interval) as interval, sum (sentbps*interval) as bps_out, sum(rcvdbps*interval) as bps_in, sum(sentutil*interval) as util_out, sum(rcvdutil*interval) as util_in from (select distinct dvid from ###(select dvid from \$log-event where \$filter and action='perf-stats' group by dvid)### t) tbl_log inner join intfstats tbl_intf on tbl_log.dvid = tbl_intf.dvid where \$cust_time_filter(timestamp) group by tmstamp, tbl_intf.dvid, intfname) t1 left join devtable t2 on t1.dvid = t2.dvid group by dev_intf order by util_in_avg desc, kbps_in_avg desc, kbps_out_avg desc

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

select

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

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

Dataset Name	Description	Log Category
fgt-ha-failure-summary	FortiGate HA Failure Summary	event
msg_desc from \$log where	me_s, amp as timestamp, dtime, devid, coalesce(n \$filter and subtype='ha' and logid_to_int(7, 37898, 37901, 37902, 37907, 37908) orde:	logid) in (35011, 35012,
Dataset Name	Description	Log Category

Dataset Name	Description	Log Category
fgt-env-faults-power	FortiGate Power Supply Faults	event
<pre>select time_s, devid, msg desc</pre>		
<pre>from ###(select from_dtime(dtim desc, logid_to_int(logid) as</pre>	e) as time_s, devid, coalesce(nullifna(l logid from \$log where \$filter and logid order by time_s desc)### t where logid	l_to_int(logid) in

Dataset Name	Description	Log Category
fgt-env-faults-fan	FortiGate Fan Faults	event
<pre>select time_s, devid, msg_desc</pre>		

from

###(select from_dtime(dtime) as time_s, devid, coalesce(nullifna(logdesc), msg) as msg_
desc, logid_to_int(logid) as logid from \$log where \$filter and logid_to_int(logid) in
(22105, 22107, 22108, 22109) order by time_s desc)### t where logid=22108 order by time_s
desc

Dataset Name	Description	Log Category
fgt-env-faults-temperature	FortiGate Temperatre Too High	event
<pre>select time_s, devid, msg_desc from</pre>		
<pre>###(select from_dtime(dtime) desc, logid_to_int(logid) as logid_to_int(logid)</pre>	as time_s, devid, coalesce(nullifna(ogid from \$log where \$filter and logi rder by time_s desc)### t where logid	d_to_int(logid) in

Dataset Name	Description	Log Category
Behaviour-Banned-Application	Bullying Chat Search and Message Logging	app-ctrl
<pre>string_agg(distinct user_src, ' agg, string_agg(distinct ipstr((select filename, app, itime, cd (`srcip`)) as user_src, `group`, ('facebook_post', 'facebook_chat 'gmail_send.message', 'linkedin_</pre>	<pre>hg_agg(distinct from_itime(itime)::text, ' ') ') as user_agg, string_agg(distinct `group`, `srcip`), ' ') as srcip_agg, count(*) as reque balesce(nullifna(`user`), nullifna(`unauthuser , `srcip` from \$log where \$filter and (lower(a t', 'twitter_post', 'youtube_video.access', 'g post', 'vimeo_video.access', 'google.search_s rder by itime desc)### t where (\$bully_keyword)</pre>	<pre>' ') as group_ ests from ### c`), ipstr app) in gmail_chat', search.phrase',</pre>

Dataset Name	Description	Log Category
Behaviour-Banned-User	Bullying Chat Search and Message Logging	app-ctrl
<pre>string_agg(distinct user_src, agg, string_agg(distinct ipstr (select filename, app, itime, (`srcip`)) as user_src, `group ('facebook_post', 'facebook_ch 'gmail_send.message', 'linkedi</pre>	<pre>ing_agg(distinct from_itime(itime)::text, ' ') as user_agg, string_agg(distinct `gro (`srcip`), ' ') as srcip_agg, count(*) as coalesce(nullifna(`user`), nullifna(`unaut `, `srcip` from \$log where \$filter and (lo tat', 'twitter_post', 'youtube_video.access n_post', 'vimeo_video.access', 'google.sea order by itime desc)### t where (\$bully ke</pre>	<pre>up`, ' ') as group_ requests from ### huser`), ipstr wer(app) in ', 'gmail_chat', rch_search.phrase',</pre>
filename order by requests des		

Dataset Name	Description	Log Category
Behaviour-Banned-User-Drilldown	Bullying Chat Search and Message Logging	app-ctrl
string_agg(distinct user_src,	<pre>ing_agg(distinct from_itime(itime)::text, ' ' ') as user_agg, string_agg(distinct `grou</pre>	
(select filename, app, itime, ((`srcip`)) as user_src, `group ('facebook_post', 'facebook_cha 'gmail_send.message', 'linkedin	<pre>(`srcip`), ' ') as srcip_agg, count(*) as r coalesce(nullifna(`user`), nullifna(`unauth `, `srcip` from \$log where \$filter and (low at', 'twitter_post', 'youtube_video.access' n_post', 'vimeo_video.access', 'google.sear order by itime desc)### t where (\$bully_key c</pre>	<pre>huser`), ipstr ver(app) in (, 'gmail_chat', cch_search.phrase',</pre>
<pre>(select filename, app, itime, ((`srcip`)) as user_src, `group ('facebook_post', 'facebook_cha' 'gmail_send.message', 'linkedin 'bing.search_search.phrase')) of</pre>	<pre>coalesce(nullifna(`user`), nullifna(`unauth `, `srcip` from \$log where \$filter and (low at', 'twitter_post', 'youtube_video.access' n_post', 'vimeo_video.access', 'google.sear order by itime desc)### t where (\$bully_key</pre>	<pre>huser`), ipstr ver(app) in (, 'gmail_chat', cch_search.phrase',</pre>

select

filename,

string agg(

distinct app,

& #039; ') as app_agg, string_agg(distinct from_itime(itime)::text, ' ') as itime agg, string agg(distinct user_src, ' ') as user_agg, string_agg(distinct `group`, ' ') as group_ agg, string agg(distinct ipstr(`srcip`), ' ') as srcip agg, count(*) as requests from ### (select filename, app, itime, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr (`srcip`)) as user src, `group`, `srcip` from \$log where \$filter and (lower(app) in ('facebook post', 'facebook chat', 'twitter post', 'youtube video.access', 'gmail chat', 'gmail send.message', 'linkedin post', 'vimeo video.access', 'google.search search.phrase', 'bing.search search.phrase')) order by itime desc)### t where (\$bully keywords) group by filename order by requests desc

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

select

filename, string agg(distinct app,

& #039; ') as app_agg, string_agg(distinct from_itime(itime)::text, ' ') as itime_agg, string_agg(distinct user_src, ' ') as user_agg, string_agg(distinct `group`, ' ') as group_ agg, string agg(distinct ipstr(`srcip`), ' ') as srcip agg, count(*) as requests from ### (select filename, app, itime, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr (`srcip`)) as user_src, `group`, `srcip` from \$log where \$filter and (lower(app) in ('facebook post', 'facebook_chat', 'twitter_post', 'youtube_video.access', 'gmail_chat', 'gmail_send.message', 'linkedin_post', 'vimeo_video.access', 'google.search_search.phrase', 'bing.search search.phrase')) order by itime desc)### t where (\$banned keywords) group by filename order by requests desc

Dataset Name	Description	Log Category
Self-Harm-Behaviour-Banned- Application-Pie	Self-Harm Chat Search and Message Logging	app-ctrl
<pre>string_agg(distinct user_src, agg, string_agg(distinct ipstr (select filename, app, itime, ((`srcip`)) as user_src, `group ('facebook_post', 'facebook_cha' 'gmail_send.message', 'linkedin'</pre>	<pre>ing_agg(distinct from_itime(itime)::text, ' ' ') as user_agg, string_agg(distinct `group (`srcip`), ' ') as srcip_agg, count(*) as re coalesce(nullifna(`user`), nullifna(`unauthu `, `srcip` from \$log where \$filter and (lowe at', 'twitter_post', 'youtube_video.access', n_post', 'vimeo_video.access', 'google.searc order by itime desc)### t where (\$banned_key c</pre>	<pre>`, ' ') as group_ quests from ### ser`), ipstr r(app) in 'gmail_chat', h_search.phrase',</pre>
Dataset Name	Description	Log Category
Self-Harm-Behaviour-Banned-User- Bar	Self-Harm Chat Search and Message Logging	app-ctrl
<pre>string_agg(distinct user_src, agg, string_agg(distinct ipstr select filename, app, itime, `srcip`)) as user_src, `group 'facebook_post', 'facebook_cha gmail_send.message', 'linkeding</pre>	<pre>ing_agg(distinct from_itime(itime)::text, ' ' ') as user_agg, string_agg(distinct `group (`srcip`), ' ') as srcip_agg, count(*) as re coalesce(nullifna(`user`), nullifna(`unauthu `, `srcip` from \$log where \$filter and (lowe at', 'twitter_post', 'youtube_video.access', n_post', 'vimeo_video.access', 'google.searc order by itime desc)### t where (\$banned_key c</pre>	`, ' ') as group_ quests from ### ser`), ipstr r(app) in 'gmail_chat', h_search.phrase',
Dataset Name	Description	Log Category
Self-Harm-Behaviour-Banned-User- Drilldown	Self-Harm Chat Search and Message Logging	app-ctrl
string_agg(distinct user_src, agg, string_agg(distinct ipstr (select filename, app, itime, o	<pre>ing_agg(distinct from_itime(itime)::text, ' ' ') as user_agg, string_agg(distinct `group (`srcip`), ' ') as srcip_agg, count(*) as re coalesce(nullifna(`user`), nullifna(`unauthu `, `srcip` from \$log where \$filter and (lowe);</pre>	`, ' ') as group_ quests from ### ser`), ipstr r(app) in

'bing.search_search.phrase')) order by itime desc)### t where (\$banned_keywords) group by
filename order by requests desc

Dataset Name	Description	Log Category
Self-Harm-behaviour-banned	Self-Harm Chat Search and Message Logging	app-ctrl
string_agg(distinct user_src, '	ng_agg(distinct from_itime(itime)::text, ' ') as user_agg, string_agg(distinct `grou `srcip`), ' ') as srcip_agg, count(*) as r	p`, ' ') as group_

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

Dataset Name	Description	Log Category
Browsing-Time-per-Social-Media	Browsing Time vs. Domain	traffic

select

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

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

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

select

f user,

sum(bandwidth) as bandwidth

```
from
```

###(select domain, f_user, srcip, ebtr_agg_flat(browsetime) as browsetime, sum(bandwidth)
as bandwidth from (select app_group_name(app) as app_group, coalesce(nullifna(`user`),
nullifna(`unauthuser`), ipstr(`srcip`)) as f_user, srcip, coalesce(nullifna(root_domain
(hostname)), ipstr(dstip), NULL) as domain, ebtr_agg_flat(\$browse_time) as browsetime, sum
(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) as bandwidth from \$log where \$filter and

(logflag&1>0) group by app_group, f_user, hostname, domain, srcip, dstip) t1 inner join app_ mdata t2 on lower(t1.app_group)=lower(t2.name) where app_cat='Social.Media' group by domain, f_user, srcip order by browsetime, bandwidth desc)### t where bandwidth>0 group by f_user order by bandwidth desc

Dataset Name	Description	Log Category
Top-Social-Networking-Durations- Sources-Drilldown	Top Social Networking Durations from Sources Drilldown	traffic
<pre>select f_user, ebtr_value(ebtr_agg_flat(browsetime), null, \$timespan) as browsetime</pre>		
<pre>as bandwidth from (select app_g: nullifna(`unauthuser`), ipstr(`: (hostname)), ipstr(dstip), NULL (coalesce(sentbyte, 0)+coalesce</pre>	cip, ebtr_agg_flat(browsetime) as browsetime, s roup_name(app) as app_group, coalesce(nullifna(srcip`)) as f_user, srcip, coalesce(nullifna(ro) as domain, ebtr_agg_flat(\$browse_time) as bro (rcvdbyte, 0)) as bandwidth from \$log where \$fi p, f_user, hostname, domain, srcip, dstip) t1 i	`user`), ot_domain wsetime, sum lter and

mdata t2 on lower(t1.app_group)=lower(t2.name) where app_cat='Social.Media' group by domain,

f user, srcip order by browsetime, bandwidth desc)### t where \$filter-drilldown and

browsetime is not null group by f user order by browsetime desc

Dataset Name	Description	Log Category
Top-Social-Networking-Durations- Domains-Drilldown	Browsing Time vs. Domain	traffic
<pre>select domain, ebtr_value(ebtr_agg_flat(browsetime), null, \$timespan) as browsetime from</pre>		
<pre>###(select domain, f_user, sr as bandwidth from (select app_g nullifna(`unauthuser`), ipstr(` (hostname)), ipstr(dstip), NULL (coalesce(sentbyte, 0)+coalesce (logflag&1>0) group by app_grou mdata t2 on lower(t1.app_group)</pre>	<pre>cip, ebtr_agg_flat(browsetime) as k roup_name(app) as app_group, coaless srcip`)) as f_user, srcip, coalesce) as domain, ebtr_agg_flat(\$browse_ (rcvdbyte, 0)) as bandwidth from \$1 p, f_user, hostname, domain, srcip, =lower(t2.name) where app_cat='Soci me, bandwidth desc)### t where brow esc</pre>	sce(nullifna(`user`), e(nullifna(root_domain _time) as browsetime, sum log where \$filter and , dstip) t1 inner join app_ ial.Media' group by domain,

Dataset Name	Description	Log Category
Facebook-Posts	Facebook Posts	app-ctrl

select
 i_time,
 f_user,
 srcip,
 filename
from

###(select from_itime(itime) as i_time, coalesce(nullifna(`user`), nullifna(`unauthuser`),
ipstr(`srcip`)) as f_user, srcip, filename, app from \$log where \$filter and filename is not
null order by i time desc)### t where lower(app)=lower('Facebook Post') order by i time desc

Dataset Name	Description	Log Category
Facebook-Chats	Facebook Chats	app-ctrl
<pre>select filename, string_agg(distinct from_itime(itime)</pre>	: :text,	

& #039; ') as itime_agg, string_agg(distinct user_src, ' ') as user_agg, string_agg (distinct `group`, ' ') as group_agg, string_agg(distinct ipstr(srcip), ' ') as srcip_agg, count(*) as requests from ###(select filename, itime, coalesce(nullifna(`user`), nullifna (`unauthuser`), ipstr(`srcip`)) as user_src, `group`, srcip, app from \$log where \$filter and filename is not null order by itime desc)### t where lower(app)=lower('Facebook_Chat') group by filename order by requests desc

Dataset Name	Description	Log Category
Twitter-Posts	Twitter Posts	app-ctrl
<pre>select i_time, f_user, srcip, filename</pre>		

from

###(select from_itime(itime) as i_time, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as f_user, srcip, filename, app from \$log where \$filter and filename is not null order by i_time desc)### t where lower(app)=lower('Twitter_Post') order by i_time desc

Dataset Name	Description	Log Category
LinkedIn-Posts-and-Comments	LinkedIn Posts and Comments	app-ctrl
select filename,		
<pre>string_agg(distinct from_itime(itime)</pre>): :text,	
	string_agg(distinct user_src, ' ') as	
	<pre>pup_agg, string_agg(distinct ipstr(src (select filename, itime, coalesce(null</pre>	
	<pre>)) as user_src, `group`, srcip, app fr itime desc)### t where lower(app)=low</pre>	-

by filename order by requests desc

devid,

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

sum(bibandwidth)/ sum(count) as bibandwidth
from
 ###(select \$flex_timestamp as timestamp, csf, devname, devid, vd, interface, healthcheck
as sla_rule, sum(link_status) as link_status, sum(failed_latency) as failed_latency, sum
(failed_jitter) as failed_jitter, sum(failed_packetloss) as failed_packetloss, sum(latency)
as latency, max(latency) as latency_max, min(latency) as latency_min, sum(jitter) as jitter,
max(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max
(packetloss) as packetloss_max, min(packetloss) as packetloss_min, sum(inbandwidth) as
inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as
count, sum(CASE WHEN link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_status) as
sdwan_status from (select itime, csf, devname, devid, vd, interface, healthcheck, link_
status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link_status=1 THEN packetloss ELSE
100 END) AS packetloss, (CASE WHEN sla failed=1 AND metric='packetloss' THEN 1 ELSE 0 END)

100 END) AS packetloss, (CASE WHEN sla failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed packetloss, (CASE WHEN sla failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed jitter, (CASE WHEN sla failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed latency, (CASE WHEN sla failed=1 THEN 3 ELSE sdwan status END) AS sdwan status, (CASE WHEN link status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth, (CASE WHEN link status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth, (CASE WHEN link status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth from (select itime, csf, devname, devid, vd, interface, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan status, convert unit to num (inbandwidthused) as inbandwidth, convert unit to num(outbandwidthused) as outbandwidth, convert unit to num(bibandwidthused) as bibandwidth from \$log where \$filter and logid to int (logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf, devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filter-drilldown and bibandwidth is not null group by devid having sum(count)>0 order by bibandwidth desc

Dataset Name	Description	Log Category
sdwan-Device-Interface-Latency-Line	SD-WAN Device-Interface Latency Timeline	event
<pre>select \$flex_timescale(timestamp) as t1.interface, min(latency) as latency</pre>	hodex,	
from		
(
select		
timestamp,		
devid,		
interface,		
<pre>sum(latency)/ sum(count) a</pre>	s latency	
from		
###(select \$flex_timestamp	as timestamp, csf, devname, devid, vd, inter	face,
healthcheck as sla_rule, sum(lin	<pre>k_status) as link_status, sum(failed_latency)</pre>	as failed_

latency, sum(failed jitter) as failed jitter, sum(failed packetloss) as failed packetloss, sum(latency) as latency, max(latency) as latency max, min(latency) as latency min, sum (jitter) as jitter, max(jitter) as jitter max, min(jitter) as jitter min, sum(packetloss) as packetloss, max(packetloss) as packetloss max, min(packetloss) as packetloss min, sum (inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN link status=1 THEN 1 ELSE 0 END) AS count linkup, min(sdwan status) as sdwan status from (select itime, csf, devname, devid, vd, interface, healthcheck, link status, (CASE WHEN link status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link status=1 THEN packetloss ELSE 100 END) AS packetloss, (CASE WHEN sla failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed packetloss, (CASE WHEN sla failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed jitter, (CASE WHEN sla failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed latency, (CASE WHEN sla failed=1 THEN 3 ELSE sdwan status END) AS sdwan status, (CASE WHEN link status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth, (CASE WHEN link status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth, (CASE WHEN link status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth from (select itime, csf, devname, devid, vd, interface, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_ unit_to_num(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth from \$log where \$filter and logid to int(logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf, devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t group by timestamp, devid, interface having sum(count)>0) t1 inner join (select interface, count(*) as num intf from ###(select \$flex timestamp as timestamp, csf, devname, devid, vd, interface, healthcheck as sla rule, sum(link status) as link status, sum(failed latency) as failed latency, sum (failed jitter) as failed jitter, sum(failed packetloss) as failed packetloss, sum(latency) as latency, max(latency) as latency max, min(latency) as latency min, sum(jitter) as jitter, max(jitter) as jitter max, min(jitter) as jitter min, sum(packetloss) as packetloss, max (packetloss) as packetloss max, min(packetloss) as packetloss min, sum(inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN link status=1 THEN 1 ELSE 0 END) AS count linkup, min(sdwan status) as sdwan status from (select itime, csf, devname, devid, vd, interface, healthcheck, link status, (CASE WHEN link status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link status=1 THEN packetloss ELSE 100 END) AS packetloss, (CASE WHEN sla failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed packetloss, (CASE WHEN sla failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed jitter, (CASE WHEN sla failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed latency, (CASE WHEN sla failed=1 THEN 3 ELSE sdwan status END) AS sdwan status, (CASE WHEN link status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth, (CASE WHEN link status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth, (CASE WHEN link status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth from (select itime, csf, devname, devid, vd, interface, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan_status, convert_unit_to_num (inbandwidthused) as inbandwidth, convert unit to num(outbandwidthused) as outbandwidth, convert unit to num(bibandwidthused) as bibandwidth from \$log where \$filter and logid to int (logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf, devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filter-drilldown and interface is not null group by interface order by num intf desc limit 10)t2 on t1.interface=t2.interface group by hodex, t1.interface order by hodex

Dataset Name	Description	Log Category
sdwan-Device-Interface-Jitter-Line	SD-WAN Device-Interface Jitter Timeline	event
Suwan-Device-Interface-Jitter-Line	SD-WAN Device-Interface Sitter Timeline	event
<pre>select \$flex timescale(timestamp) as</pre>	bodov	
t1.interface,	nodex,	
min(jitter) as jitter		
from		
(
select		
timestamp,		
devid,		
interface,		
<pre>sum(jitter)/ sum(count) as</pre>	jitter	
from		
	as timestamp, csf, devname, devid, vd, i	
—	k_status) as link_status, sum(failed_late	
	ailed_jitter, sum(failed_packetloss) as	—
	ency) as latency_max, min(latency) as lat	
	as jitter_max, min(jitter) as jitter_min,	
	acketloss_max, min(packetloss) as packetl	—
	m(outbandwidth) as outbandwidth, sum(biba	
	<pre>sum(CASE WHEN link_status=1 THEN 1 ELSE C an status from (select itime, csf, devnam</pre>	—
—	tus, (CASE WHEN link status=1 THEN latence	
	1 THEN jitter ELSE 0 END) AS jitter, (CAS	
—	0 END) AS packetloss, (CASE WHEN sla fail	—
	0 END) AS failed packetloss, (CASE WHEN s	
	D) AS failed jitter, (CASE WHEN sla faile	—
	ND) AS failed latency, (CASE WHEN sla fai	
	s, (CASE WHEN link status=1 THEN inbandw	
inbandwidth, (CASE WHEN link_st	atus=1 THEN outbandwidth ELSE 0 END) AS c	outbandwidth, (CASE
WHEN link_status=1 THEN bibandwi	dth ELSE 0 END) AS bibandwidth from (sele	ect itime, csf,
devname, devid, vd, interface, h	ealthcheck, (CASE WHEN status='down' THEN	1 O ELSE 1 END) AS
	atency, jitter::float as jitter, trim(tra	-
	, (CASE WHEN status='down' THEN 1 WHEN ms	
	la_failed, metric, (CASE WHEN msg LIKE '%	
	vert_unit_to_num(inbandwidthused) as inba	
	outbandwidth, convert_unit_to_num(biband	
	ter and logid_to_int(logid) in (22925, 22	
	up by timestamp, csf, devname, devid, vd,	
	y timestamp desc/*SkipEND*/)### t group k 1 inner join (select interface, count(*)	
	mestamp, csf, devname, devid, vd, interfa	—
	nk status, sum(failed latency) as failed	
	, sum(failed packetloss) as failed packet	
as latency, max(latency) as late	ncy_max, min(latency) as latency_min, sum	n(jitter) as jitter,
	itter) as jitter_min, sum(packetloss) as	
—	<pre>min(packetloss) as packetloss_min, sum(ir</pre>	
	s outbandwidth, sum(bibandwidth) as bibar	
	=1 THEN 1 ELSE 0 END) AS count_linkup, mi	—
—	csf, devname, devid, vd, interface, heal	—
	THEN latency ELSE 0 END) AS latency, (CA	_
	AS jitter, (CASE WHEN link_status=1 THEN	
IUU END) AS PACKETIOSS, (CASE WH	EN sla_failed=1 AND metric='packetloss' 1	. TEN I ELSE V END)

AS failed packetloss, (CASE WHEN sla failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed_jitter, (CASE WHEN sla_failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed latency, (CASE WHEN sla failed=1 THEN 3 ELSE sdwan status END) AS sdwan status, (CASE WHEN link status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth, (CASE WHEN link status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth, (CASE WHEN link status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth from (select itime, csf, devname, devid, vd, interface, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msq LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan_status, convert_unit_to_num (inbandwidthused) as inbandwidth, convert unit to num(outbandwidthused) as outbandwidth, convert unit to num(bibandwidthused) as bibandwidth from \$log where \$filter and logid to int (logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf, devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filter-drilldown and interface is not null group by interface order by num intf desc limit 10)t2 on t1.interface=t2.interface group by hodex, t1.interface order by hodex

Dataset Name	Description	Log Category
sdwan-Device-Interface-Packetloss- Line	SD-WAN Device-Interface Packetloss Timeline	event
<pre>select \$flex_timescale(timestamp) as t1.interface, min(packetloss) as packetloss</pre>	hodex,	
<pre>from (select timestamp, devid, interface,</pre>		

```
sum(packetloss) / sum(count) as packetloss
```

```
from
```

###(select \$flex_timestamp as timestamp, csf, devname, devid, vd, interface, healthcheck as sla rule, sum(link status) as link status, sum(failed latency) as failed latency, sum(failed_jitter) as failed jitter, sum(failed packetloss) as failed packetloss, sum(latency) as latency, max(latency) as latency max, min(latency) as latency min, sum (jitter) as jitter, max(jitter) as jitter max, min(jitter) as jitter min, sum(packetloss) as packetloss, max(packetloss) as packetloss max, min(packetloss) as packetloss min, sum (inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN link status=1 THEN 1 ELSE 0 END) AS count linkup, min(sdwan status) as sdwan status from (select itime, csf, devname, devid, vd, interface, healthcheck, link_status, (CASE WHEN link status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link status=1 THEN packetloss ELSE 100 END) AS packetloss, (CASE WHEN sla failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed packetloss, (CASE WHEN sla failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed jitter, (CASE WHEN sla failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed latency, (CASE WHEN sla failed=1 THEN 3 ELSE sdwan status END) AS sdwan status, (CASE WHEN link status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth, (CASE WHEN link status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth, (CASE WHEN link status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth from (select itime, csf, devname, devid, vd, interface, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA

failed%' THEN 1 ELSE 0 END) AS sla failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_ unit to num(outbandwidthused) as outbandwidth, convert unit to num(bibandwidthused) as bibandwidth from \$log where \$filter and logid to int(logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf, devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t group by timestamp, devid, interface having sum(count)>0) t1 inner join (select interface, count(*) as num intf from ###(select \$flex timestamp as timestamp, csf, devname, devid, vd, interface, healthcheck as sla rule, sum(link status) as link status, sum(failed latency) as failed latency, sum (failed jitter) as failed jitter, sum(failed packetloss) as failed packetloss, sum(latency) as latency, max(latency) as latency max, min(latency) as latency min, sum(jitter) as jitter, max(jitter) as jitter max, min(jitter) as jitter min, sum(packetloss) as packetloss, max (packetloss) as packetloss max, min(packetloss) as packetloss min, sum(inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN link status=1 THEN 1 ELSE 0 END) AS count linkup, min(sdwan status) as sdwan status from (select itime, csf, devname, devid, vd, interface, healthcheck, link status, (CASE WHEN link status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link status=1 THEN packetloss ELSE 100 END) AS packetloss, (CASE WHEN sla failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed packetloss, (CASE WHEN sla failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed jitter, (CASE WHEN sla failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed latency, (CASE WHEN sla failed=1 THEN 3 ELSE sdwan status END) AS sdwan status, (CASE WHEN link status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth, (CASE WHEN link status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth, (CASE WHEN link status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth from (select itime, csf, devname, devid, vd, interface, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan status, convert unit to num (inbandwidthused) as inbandwidth, convert unit to num(outbandwidthused) as outbandwidth, convert unit to num(bibandwidthused) as bibandwidth from \$log where \$filter and logid to int (logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf, devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filter-drilldown and interface is not null group by interface order by num intf desc limit 10)t2 on t1.interface=t2.interface group by hodex, t1.interface order by hodex

Dataset Name	Description	Log Category
sdwan-Device-Latency-Line	SD-WAN Device Latency Timeline	event
healthcheck as sla_rule, sum(lin		ed_latency) as failed_

sum(latency) as latency, max(latency) as latency max, min(latency) as latency min, sum (jitter) as jitter, max(jitter) as jitter max, min(jitter) as jitter min, sum(packetloss) as packetloss, max(packetloss) as packetloss max, min(packetloss) as packetloss min, sum (inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN link status=1 THEN 1 ELSE 0 END) AS count linkup, min(sdwan status) as sdwan status from (select itime, csf, devname, devid, vd, interface, healthcheck, link status, (CASE WHEN link status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link status=1 THEN packetloss ELSE 100 END) AS packetloss, (CASE WHEN sla failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed packetloss, (CASE WHEN sla failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed jitter, (CASE WHEN sla failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed latency, (CASE WHEN sla failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status, (CASE WHEN link_status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth, (CASE WHEN link status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth, (CASE WHEN link status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth from (select itime, csf, devname, devid, vd, interface, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_ unit to num(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth from \$log where \$filter and logid to int(logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf, devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filter-drilldown and latency is not null group by timestamp, devid, interface having sum(count)>0) t1 group by hodex, devid order by hodex

Dataset Name	Description	Log Category
sdwan-Device-Jitter-Line	SD-WAN Device Jitter Timeline	event
<pre>select \$flex_timescale(timestamp) as hodex, devid, min(jitter) as jitter from (select timestamp, devid, interface,</pre>		
<pre>sum(jitter)/ sum(count) as from ###(select \$flex_timestamp</pre>	as timestamp, csf, devname, devid, vd, ir	
<pre>healthcheck as sla_rule, sum(link_status) as link_status, sum(failed_latency) as failed_ latency, sum(failed_jitter) as failed_jitter, sum(failed_packetloss) as failed_packetloss, sum(latency) as latency, max(latency) as latency_max, min(latency) as latency_min, sum (jitter) as jitter, max(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max(packetloss) as packetloss_max, min(packetloss) as packetloss_min, sum</pre>		
(inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN link_status=1 THEN 1 ELSE 0 END) AS count_ linkup, min(sdwan_status) as sdwan_status from (select itime, csf, devname, devid, vd, interface, healthcheck, link_status, (CASE WHEN link_status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link_status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link_		

status=1 THEN packetloss ELSE 100 END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed packetloss, (CASE WHEN sla failed=1 AND

metric='jitter' THEN 1 ELSE 0 END) AS failed jitter, (CASE WHEN sla failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed latency, (CASE WHEN sla failed=1 THEN 3 ELSE sdwan status END) AS sdwan status, (CASE WHEN link status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth, (CASE WHEN link status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth, (CASE WHEN link status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth from (select itime, csf, devname, devid, vd, interface, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msq LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan status, convert unit to num(inbandwidthused) as inbandwidth, convert unit to num(outbandwidthused) as outbandwidth, convert unit to num(bibandwidthused) as bibandwidth from \$log where \$filter and logid to int(logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf, devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filter-drilldown and jitter is not null group by timestamp, devid, interface having sum(count)>0) t1 group by hodex, devid order by hodex

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

###(select \$flex_timestamp as timestamp, csf, devname, devid, vd, interface, healthcheck as sla rule, sum(link status) as link status, sum(failed latency) as failed latency, sum(failed jitter) as failed jitter, sum(failed packetloss) as failed packetloss, sum(latency) as latency, max(latency) as latency max, min(latency) as latency min, sum (jitter) as jitter, max(jitter) as jitter max, min(jitter) as jitter min, sum(packetloss) as packetloss, max(packetloss) as packetloss max, min(packetloss) as packetloss min, sum (inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN link status=1 THEN 1 ELSE 0 END) AS count_ linkup, min(sdwan status) as sdwan status from (select itime, csf, devname, devid, vd, interface, healthcheck, link status, (CASE WHEN link status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link status=1 THEN packetloss ELSE 100 END) AS packetloss, (CASE WHEN sla failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed packetloss, (CASE WHEN sla_failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed jitter, (CASE WHEN sla failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed latency, (CASE WHEN sla failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status, (CASE WHEN link_status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth, (CASE WHEN link status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth, (CASE WHEN link status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth from (select itime, csf, devname, devid, vd, interface, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan status, convert unit to num(inbandwidthused) as inbandwidth, convert

unit_to_num(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth from \$log where \$filter and logid_to_int(logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf, devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filter-drilldown and packetloss is not null group by timestamp, devid, interface having sum(count)>0) t1 group by hodex, devid order by hodex

Dataset Name	Description	Log Category
sdwan-Device-Interface-Summary-by-	SD-WAN Device Interface Summary by Bibandwidth	event
Bibandwidth		

```
select
 devid,
 interface,
 sum(bibandwidth) / sum(count) as bibandwidth,
 cast (
  min(latency min) as decimal(18, 2)
 ) as latency_min,
 cast(
   sum(latency) / sum(count) as decimal(18, 2)
  ) as latency avg,
 cast(
   max(latency max) as decimal(18, 2)
 ) as latency max,
 cast(
   min(jitter min) as decimal(18, 2)
 ) as jitter min,
 cast(
    sum(jitter) / sum(count) as decimal(18, 2)
  ) as jitter avg,
  cast(
  max(jitter max) as decimal(18, 2)
  ) as jitter max,
 cast(
   min(packetloss_min) as decimal(18, 2)
 ) as packetloss min,
 cast(
   sum(packetloss) / sum(count) as decimal(18, 2)
 ) as packetloss avg,
 cast(
   max(packetloss max) as decimal(18, 2)
 ) as packetloss max
from
```

(select \$flex_timestamp as timestamp, csf, devname, devid, vd, interface, healthcheck as sla_rule, sum(link_status) as link_status, sum(failed_latency) as failed_latency, sum (failed_jitter) as failed_jitter, sum(failed_packetloss) as failed_packetloss, sum(latency) as latency, max(latency) as latency_max, min(latency) as latency_min, sum(jitter) as jitter, max(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max (packetloss) as packetloss_max, min(packetloss) as packetloss_min, sum(inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_status) as sdwan_status from (select itime, csf, devname, devid, vd, interface, healthcheck, link_ status, (CASE WHEN link_status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link_ status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link_status=1 THEN packetloss ELSE 100 END) AS packetloss, (CASE WHEN sla failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed packetloss, (CASE WHEN sla failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed jitter, (CASE WHEN sla failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed latency, (CASE WHEN sla failed=1 THEN 3 ELSE sdwan status END) AS sdwan status, (CASE WHEN link status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth, (CASE WHEN link status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth, (CASE WHEN link status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth from (select itime, csf, devname, devid, vd, interface, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msq LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan status, convert unit to num (inbandwidthused) as inbandwidth, convert unit to num(outbandwidthused) as outbandwidth, convert unit to num(bibandwidthused) as bibandwidth from \$log where \$filter and logid to int (logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf, devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filter-drilldown and interface is not null group by devid, interface having sum(count)>0 order by devid, interface

Dataset Name	Description	Log Category	
sdwan-Top-App-By-Bandwidth	Top SD-WAN application by bandwidth	traffic	
<pre>select appid, app_group, sum(bandwidth) as bandwidth, sum(sessions) as sessions from ###(select \$flex_timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group_name(app) as app_group, coalesce (vwlname,vwlservice) as rulename, service, coalesce(nullifna(`srcname`),ipstr (`srcip`),nullifna(`srcmac`)) as dev_src, sum(crscore%65536) as crscore, coalesce(nullifna (`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from \$log-traffic where \$filter and vwlid IS NOT NULL and (logflag&(1 32)>0) group by timestamp, srccountry, dstintf, csf, devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_src, dev_src order by bandwidth desc)### t where \$filter-drilldown group by appid, app_group order by bandwidth desc</pre>			
Dataset Name	Description	Log Category	
sdwan-Top-App-By-Bandwidth-Sankey	Top SD-WAN application by bandwidth usage	traffic	
<pre>select & #039;SD-WAN Utilization' as summary, app_group, devid, dstintf as interface, sum (bandwidth) as bandwidth from ###(select \$flex_timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group_name(app) as app_group, coalesce(vwlname,vwlservice) as rulename, service, coalesce(nullifna (`srcname`),ipstr(`srcip`),nullifna(`srcmac`)) as dev_src, sum(crscore%65536) as crscore, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce (sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce (rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_ out, count(*) as sessions from \$log-traffic where \$filter and vwlid IS NOT NULL and</pre>			

(logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf, devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_src, dev_src order by bandwidth desc)### t where \$filter-drilldown group by app_group, devid, interface order by bandwidth desc

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

select

devid, sum(bibandwidth) / sum(count) as bibandwidth from

###(select \$flex_timestamp as timestamp, csf, devname, devid, vd, interface, healthcheck as sla rule, sum(link status) as link status, sum(failed latency) as failed latency, sum (failed jitter) as failed jitter, sum(failed packetloss) as failed packetloss, sum(latency) as latency, max(latency) as latency max, min(latency) as latency min, sum(jitter) as jitter, max(jitter) as jitter max, min(jitter) as jitter min, sum(packetloss) as packetloss, max (packetloss) as packetloss max, min(packetloss) as packetloss_min, sum(inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN link status=1 THEN 1 ELSE 0 END) AS count linkup, min(sdwan status) as sdwan status from (select itime, csf, devname, devid, vd, interface, healthcheck, link status, (CASE WHEN link status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link status=1 THEN packetloss ELSE 100 END) AS packetloss, (CASE WHEN sla failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed packetloss, (CASE WHEN sla failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed jitter, (CASE WHEN sla failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed latency, (CASE WHEN sla failed=1 THEN 3 ELSE sdwan status END) AS sdwan status, (CASE WHEN link status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth, (CASE WHEN link status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth, (CASE WHEN link status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth from (select itime, csf, devname, devid, vd, interface, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan status, convert unit to num (inbandwidthused) as inbandwidth, convert unit to num(outbandwidthused) as outbandwidth, convert unit to num(bibandwidthused) as bibandwidth from \$log where \$filter and logid to int (logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf, devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filter-drilldown and bibandwidth is not null group by devid having sum(count)>0 order by bibandwidth desc

Dataset Name	Description	Log Category
sdwan-Device-Rules-Donut-Bandwidth	Top SD-WAN Links bandwidth	traffic
<pre>timestamp as timestamp, csf, devi dstintfrole, appid, appcat, app_g as rulename, service, coalesce(nu dev_src, sum(crscore%65536) as c ipstr(`srcip`)) as user_src, sum rcvdbyte, 0)) as bandwidth, sum(c</pre>	<pre>, sum(bandwidth) as bandwidth from ###(select id, vd, srccountry, dstintf, srcintf, srcintf; group_name(app) as app_group, coalesce(vwlname ullifna(`srcname`),ipstr(`srcip`),nullifna(`sr crscore, coalesce(nullifna(`user`), nullifna((coalesce(sentdelta, sentbyte, 0)+coalesce(rcv coalesce(rcvddelta, rcvdbyte, 0)) as traffic_;)) as traffic_out, count(*) as sessions from s</pre>	role, e,vwlservice) rcmac`)) as `unauthuser`), vddelta, in, sum

where \$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf, devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_src, dev_src order by bandwidth desc)### t where \$filter-drilldown group by rulename order by bandwidth desc limit 10

Dataset Name	Description	Log Category
sdwan-device-interface-bandwidth	Top SD-WAN Links bandwidth	traffic
from		
sum(bandwidth) as bandwidth		

interface order by bandwidth desc limit 10

Dataset Name	Description	Log Category
sdwan-Top-Application-Session- Bandwidth	Top SD-WAN application by bandwidth	traffic
<pre>select appid, app_group, sum(bandwidth) as bandwidth, sum(sessions) as sessions</pre>		
	timestamp, csf, devid, vd, srccountry, dstin appcat, app_group_name(app) as app_group, c	

(vwlname,vwlservice) as rulename, service, coalesce(nullifna(`srcname`),ipstr (`srcip`),nullifna(`srcmac`)) as dev_src, sum(crscore%65536) as crscore, coalesce(nullifna (`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from \$log-traffic where \$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf, devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_src, dev_src order by bandwidth desc)### t where \$filter-drilldown group by appid, app_group order by bandwidth desc

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

select

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

###(select \$flex_timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group_name(app) as app_group, coalesce (vwlname,vwlservice) as rulename, service, coalesce(nullifna(`srcname`),ipstr (`srcip`),nullifna(`srcmac`)) as dev_src, sum(crscore%65536) as crscore, coalesce(nullifna (`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from \$log-traffic where \$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf, devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_src, dev_src order by bandwidth desc)### t where \$filter-drilldown group by user_src order by bandwidth desc

Dataset Name	Description	Log Category
sdwan-top-user-app-Drilldown	SD-WAN Top users and Application by bandwidth	traffic
<pre>srcintfrole, dstintfrole, appid (vwlname,vwlservice) as rulenam (`srcip`),nullifna(`srcmac`)) a (`user`), nullifna(`unauthuser` sentbyte, 0)+coalesce(rcvddelta rcvdbyte, 0)) as traffic_in, su as sessions from \$log-traffic w group by timestamp, srccountry, appid, appcat, app_group, rulen</pre>	a timestamp, csf, devid, vd, srccountry, dst l, appcat, app_group_name(app) as app_group, ne, service, coalesce(nullifna(`srcname`),ip as dev_src, sum(crscore%65536) as crscore,), ipstr(`srcip`)) as user_src, sum(coalesce c, rcvdbyte, 0)) as bandwidth, sum(coalesce m(coalesce(sentdelta, sentbyte, 0)) as trans- there \$filter and vwlid IS NOT NULL and (log dstintf, csf, devid, vd, srcintf, srcintfr tame, service, user_src, dev_src order by ba- by user_src, app_group order by bandwidth device the service of the servic	<pre>, coalesce ostr coalesce(nullifna ce(sentdelta, (rcvddelta, ffic_out, count(*) gflag&(1 32)>0) cole, dstintfrole, andwidth desc)### t</pre>
Dataset Name	Description	Log Category

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

```
select
  $flex_timescale(timestamp) as hodex,
  t1.dstintf as interface,
  sum(traffic_out) as bandwidth
from
```

###(select \$flex timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf, srcintfrole, dstintfrole, appid, appcat, app group name(app) as app group, coalesce (vwlname,vwlservice) as rulename, service, coalesce(nullifna(`srcname`),ipstr (`srcip`),nullifna(`srcmac`)) as dev src, sum(crscore%65536) as crscore, coalesce(nullifna (`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic out, count(*) as sessions from \$log-traffic where \$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf, devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app group, rulename, service, user src, dev src order by bandwidth desc) ### t1 inner join (select dstintf, count(*) as num intf from ###(select \$flex timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf, srcintfrole, dstintfrole, appid, appcat, app group name(app) as app group, coalesce(vwlname,vwlservice) as rulename, service, coalesce(nullifna(`srcname`),ipstr(`srcip`),nullifna(`srcmac`)) as dev src, sum (crscore%65536) as crscore, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr (`srcip`)) as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic in, sum(coalesce (sentdelta, sentbyte, 0)) as traffic out, count(*) as sessions from \$log-traffic where \$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf, devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app group, rulename, service, user src, dev src order by bandwidth desc)### t where \$filter-drilldown group by dstintf order by num intf desc limit 10)t2 on t1.dstintf=t2.dstintf group by hodex, tl.dstintf order by hodex

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

select

```
$flex_timescale(timestamp) as hodex,
t1.srcintf as interface,
sum(traffic_in) as bandwidth
from
```

from

###(select \$flex timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf, srcintfrole, dstintfrole, appid, appcat, app group name(app) as app group, coalesce (vwlname,vwlservice) as rulename, service, coalesce(nullifna(`srcname`),ipstr (`srcip`),nullifna(`srcmac`)) as dev src, sum(crscore%65536) as crscore, coalesce(nullifna (`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic out, count(*) as sessions from \$log-traffic where \$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf, devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app group, rulename, service, user src, dev src order by bandwidth desc) ### t1 inner join (select srcintf, count(*) as num intf from ###(select \$flex timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group_name(app) as app_group, coalesce(vwlname,vwlservice) as rulename, service, coalesce(nullifna(`srcname`),ipstr(`srcip`),nullifna(`srcmac`)) as dev src, sum (crscore%65536) as crscore, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr (`srcip`)) as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic in, sum(coalesce

(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from \$log-traffic where \$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf, devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_src, dev_src order by bandwidth desc)### t where \$filter-drilldown and srcintf is not null and srcintfrole ='wan' group by srcintf order by num_intf desc limit 10)t2 on t1.srcintf=t2.srcintf group by hodex, t1.srcintf order by hodex

Dataset Name

Description

Log Category

sdwan-Device-Intfe-traffic-bandwidth- SD-WAN Device-Interface traffic sent bandwidth Timeline traffic Line

select

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

###(select \$flex_timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf, srcintfrole, dstintfrole, appid, appcat, app group name(app) as app group, coalesce (vwlname, vwlservice) as rulename, service, coalesce(nullifna(`srcname`), ipstr (`srcip`),nullifna(`srcmac`)) as dev src, sum(crscore%65536) as crscore, coalesce(nullifna (`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic out, count(*) as sessions from \$log-traffic where \$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf, devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app group, rulename, service, user src, dev src order by bandwidth desc) ### tl inner join (select dstintf, count(*) as num intf from ###(select \$flex timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf, srcintfrole, dstintfrole, appid, appcat, app group name(app) as app group, coalesce(vwlname,vwlservice) as rulename, service, coalesce(nullifna(`srcname`),ipstr(`srcip`),nullifna(`srcmac`)) as dev src, sum (crscore%65536) as crscore, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr (`srcip`)) as user src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic in, sum(coalesce (sentdelta, sentbyte, 0)) as traffic out, count(*) as sessions from \$log-traffic where \$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf, devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app group, rulename, service, user src, dev src order by bandwidth desc) ### t where \$filter-drilldown group by dstintf order by num intf desc limit 10)t2 on t1.dstintf=t2.dstintf group by hodex, tl.dstintf order by hodex

Description	Log Category	
SD-WAN Device Statistic by Bibandwidth	event	
s bibandwidth		
from		
as sla_rule, sum(link_status) as link_status, sum(failed_latency) as failed_latency, sum		
(failed_jitter) as failed_jitter, sum(failed_packetloss) as failed_packetloss, sum(latency)		
as latency, max(latency) as latency_max, min(latency) as latency_min, sum(jitter) as jitter,		
<pre>nax(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max (packetloss) as packetloss, min(packetloss) as packetloss min, sum(inhandwidth) as</pre>		
	SD-WAN Device Statistic by Bibandwidth s bibandwidth timestamp, csf, devname, devid, vd, interface, link_status, sum(failed_latency) as failed_la , sum(failed_packetloss) as failed_packetloss ncy_max, min(latency) as latency_min, sum(jitt	

inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN link status=1 THEN 1 ELSE 0 END) AS count linkup, min(sdwan status) as sdwan status from (select itime, csf, devname, devid, vd, interface, healthcheck, link status, (CASE WHEN link status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link status=1 THEN packetloss ELSE 100 END) AS packetloss, (CASE WHEN sla failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed packetloss, (CASE WHEN sla failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed jitter, (CASE WHEN sla failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed latency, (CASE WHEN sla failed=1 THEN 3 ELSE sdwan status END) AS sdwan status, (CASE WHEN link status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth, (CASE WHEN link status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth, (CASE WHEN link status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth from (select itime, csf, devname, devid, vd, interface, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan status, convert unit to num (inbandwidthused) as inbandwidth, convert unit to num(outbandwidthused) as outbandwidth, convert unit to num(bibandwidthused) as bibandwidth from \$log where \$filter and logid to int (logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf, devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filter-drilldown and bibandwidth is not null group by devid having sum(count)>0 order by bibandwidth desc

Dataset Name	Description	Log Category
sdwan-Device-SLA-Rule-Latency-Line	SD-WAN Device-SLA-Rule Latency Line	event
<pre>select \$flex_timescale(timestamp) as l t1.intf sla,</pre>	hodex,	
<pre>sum(latency) / sum(count) as lat</pre>	tency	
from		
(
select		
timestamp,	ale mule estimate ale com (laterary) estimate	
	<pre>sla_rule as intf_sla, sum(latency) as latency stamp as timestamp, csf, devname, devid, vd,</pre>	
	k status) as link status, sum(failed latency	
_	ailed jitter, sum(failed packetloss) as fai	_
— — — — — — — — — — — — — — — — — — —	ency) as latency max, min(latency) as latency	
(jitter) as jitter, max(jitter)	as jitter_max, min(jitter) as jitter_min, su	m(packetloss) as
packetloss, max(packetloss) as pa	acketloss_max, min(packetloss) as packetloss	_min, sum
(inbandwidth) as inbandwidth, sur	m(outbandwidth) as outbandwidth, sum(bibandw	idth) as
	sum(CASE WHEN link_status=1 THEN 1 ELSE 0 EN	_
	an_status from (select itime, csf, devname,	
—	tus, (CASE WHEN link_status=1 THEN latency E	
	1 THEN jitter ELSE 0 END) AS jitter, (CASE W.	_
-	0 END) AS packetloss, (CASE WHEN sla_failed= 0 END) AS failed packetloss, (CASE WHEN sla	
-	D) AS failed jitter, (CASE WHEN SIA_	
-	ND) AS failed latency, (CASE WHEN sla failed	
_	s, (CASE WHEN link status=1 THEN inbandwidt	
	atus=1 THEN outbandwidth ELSE 0 END) AS outb	
_	dth ELSE 0 END) AS bibandwidth from (select	
—	ealthcheck, (CASE WHEN status='down' THEN 0	
<pre>link_status, latency::float as la</pre>	atency, jitter::float as jitter, trim(traili	ng '%' from

packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msq LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan status, convert unit to num(inbandwidthused) as inbandwidth, convert unit to num(outbandwidthused) as outbandwidth, convert unit to num(bibandwidthused) as bibandwidth from \$log where \$filter and logid to int(logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf, devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where latency is not null group by timestamp, intf sla having sum(count)>0) t1 inner join (select interface || ':' || sla_rule as intf_sla, count(*) as num intf from ###(select \$flex timestamp as timestamp, csf, devname, devid, vd, interface, healthcheck as sla rule, sum(link status) as link status, sum(failed latency) as failed latency, sum(failed jitter) as failed jitter, sum (failed packetloss) as failed packetloss, sum(latency) as latency, max(latency) as latency_ max, min(latency) as latency min, sum(jitter) as jitter, max(jitter) as jitter max, min (jitter) as jitter min, sum(packetloss) as packetloss, max(packetloss) as packetloss max, min(packetloss) as packetloss min, sum(inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN link status=1 THEN 1 ELSE 0 END) AS count linkup, min(sdwan status) as sdwan status from (select itime, csf, devname, devid, vd, interface, healthcheck, link status, (CASE WHEN link status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link status=1 THEN packetloss ELSE 100 END) AS packetloss, (CASE WHEN sla failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed packetloss, (CASE WHEN sla failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed jitter, (CASE WHEN sla failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed latency, (CASE WHEN sla failed=1 THEN 3 ELSE sdwan status END) AS sdwan status, (CASE WHEN link status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth, (CASE WHEN link status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth, (CASE WHEN link status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth from (select itime, csf, devname, devid, vd, interface, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link status, latency::float as latency, jitter::float as jitter, trim (trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan status, convert unit to num(inbandwidthused) as inbandwidth, convert unit to num(outbandwidthused) as outbandwidth, convert unit to num (bibandwidthused) as bibandwidth from \$log where \$filter and logid to int(logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf, devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filterdrilldown and sla_rule is not null group by intf_sla order by num_intf desc limit 10)t2 on t1.intf sla=t2.intf sla group by hodex, t1.intf sla order by hodex

Dataset Name	Description	Log Category
sdwan-Device-SLA-Rule-Jitter-Line	SD-WAN Device-SLA-Rule Jitter Line	event
<pre>select \$flex_timescale(timestamp) as t1.intf_sla, sum(jitter)/ sum(count) as jit from (select timestamp</pre>		
<pre>timestamp, interface & #039;:' sla_rule as intf_sla, sum(jitter) as jitter, sum(count) as count from ###(select \$flex_timestamp as timestamp, csf, devname, devid, vd, interface, healthcheck as sla_rule, sum(link_status) as link_status, sum(failed_latency) as failed_ latency, sum(failed_jitter) as failed_jitter, sum(failed_packetloss) as failed_packetloss, sum(latency) as latency, max(latency) as latency_max, min(latency) as latency_min, sum (jitter) as jitter, max(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss) as</pre>		

packetloss, max(packetloss) as packetloss max, min(packetloss) as packetloss min, sum (inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN link status=1 THEN 1 ELSE 0 END) AS count linkup, min(sdwan status) as sdwan status from (select itime, csf, devname, devid, vd, interface, healthcheck, link status, (CASE WHEN link status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link status=1 THEN packetloss ELSE 100 END) AS packetloss, (CASE WHEN sla failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed packetloss, (CASE WHEN sla failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed jitter, (CASE WHEN sla failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed latency, (CASE WHEN sla failed=1 THEN 3 ELSE sdwan status END) AS sdwan status, (CASE WHEN link status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth, (CASE WHEN link status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth, (CASE WHEN link status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth from (select itime, csf, devname, devid, vd, interface, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msq LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan status, convert unit to num(inbandwidthused) as inbandwidth, convert unit to num(outbandwidthused) as outbandwidth, convert unit to num(bibandwidthused) as bibandwidth from \$log where \$filter and logid_to_int(logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf, devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where jitter is not null group by timestamp, intf sla having sum(count)>0) t1 inner join (select interface || ':' || sla rule as intf sla, count(*) as num intf from ###(select \$flex timestamp as timestamp, csf, devname, devid, vd, interface, healthcheck as sla_rule, sum(link_status) as link_ status, sum(failed latency) as failed latency, sum(failed jitter) as failed jitter, sum (failed packetloss) as failed packetloss, sum(latency) as latency, max(latency) as latency max, min(latency) as latency min, sum(jitter) as jitter, max(jitter) as jitter max, min (jitter) as jitter min, sum(packetloss) as packetloss, max(packetloss) as packetloss max, min(packetloss) as packetloss min, sum(inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN link status=1 THEN 1 ELSE 0 END) AS count linkup, min(sdwan status) as sdwan status from (select itime, csf, devname, devid, vd, interface, healthcheck, link status, (CASE WHEN link status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link status=1 THEN packetloss ELSE 100 END) AS packetloss, (CASE WHEN sla failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed packetloss, (CASE WHEN sla failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed jitter, (CASE WHEN sla failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed latency, (CASE WHEN sla failed=1 THEN 3 ELSE sdwan status END) AS sdwan status, (CASE WHEN link status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth, (CASE WHEN link status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth, (CASE WHEN link status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth from (select itime, csf, devname, devid, vd, interface, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, latency::float as latency, jitter::float as jitter, trim (trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan status, convert unit to num(inbandwidthused) as inbandwidth, convert unit to num(outbandwidthused) as outbandwidth, convert unit to num (bibandwidthused) as bibandwidth from \$log where \$filter and logid_to_int(logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf, devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filterdrilldown and sla rule is not null group by intf sla order by num intf desc limit 10)t2 on t1.intf sla=t2.intf sla group by hodex, t1.intf sla order by hodex

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

```
select
  $flex_timescale(timestamp) as hodex,
  t1.intf_sla,
  sum(packetloss) / sum(count) as packetloss
from
```

(

select

timestamp,

interface || & #039;:' || sla rule as intf sla, sum(packetloss) as packetloss, sum (count) as count from ###(select \$flex timestamp as timestamp, csf, devname, devid, vd, interface, healthcheck as sla rule, sum(link status) as link status, sum(failed latency) as failed latency, sum(failed jitter) as failed jitter, sum(failed packetloss) as failed packetloss, sum(latency) as latency, max(latency) as latency_max, min(latency) as latency_ min, sum(jitter) as jitter, max(jitter) as jitter max, min(jitter) as jitter min, sum (packetloss) as packetloss, max(packetloss) as packetloss max, min(packetloss) as packetloss min, sum(inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum (bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN link status=1 THEN 1 ELSE 0 END) AS count linkup, min(sdwan status) as sdwan status from (select itime, csf, devname, devid, vd, interface, healthcheck, link status, (CASE WHEN link status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link status=1 THEN packetloss ELSE 100 END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed packetloss, (CASE WHEN sla failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed jitter, (CASE WHEN sla failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed latency, (CASE WHEN sla failed=1 THEN 3 ELSE sdwan status END) AS sdwan status, (CASE WHEN link status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth, (CASE WHEN link status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth, (CASE WHEN link status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth from (select itime, csf, devname, devid, vd, interface, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan status, convert unit to num(inbandwidthused) as inbandwidth, convert unit to num(outbandwidthused) as outbandwidth, convert unit to num(bibandwidthused) as bibandwidth from \$log where \$filter and logid to int(logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf, devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where packetloss is not null group by timestamp, intf sla having sum(count)>0) t1 inner join (select interface || ':' || sla rule as intf sla, count(*) as num intf from ###(select \$flex timestamp as timestamp, csf, devname, devid, vd, interface, healthcheck as sla rule, sum(link status) as link status, sum(failed latency) as failed latency, sum(failed jitter) as failed jitter, sum(failed packetloss) as failed packetloss, sum(latency) as latency, max(latency) as latency max, min(latency) as latency min, sum(jitter) as jitter, max(jitter) as jitter max, min(jitter) as jitter min, sum(packetloss) as packetloss, max(packetloss) as packetloss max, min(packetloss) as packetloss min, sum(inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN link status=1 THEN 1 ELSE 0 END) AS count linkup, min(sdwan status) as sdwan status from (select itime, csf, devname, devid, vd, interface, healthcheck, link status, (CASE WHEN link status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link status=1 THEN packetloss ELSE 100 END) AS packetloss, (CASE WHEN sla failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed packetloss, (CASE WHEN sla failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed jitter, (CASE WHEN sla failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed latency, (CASE WHEN sla failed=1 THEN 3 ELSE sdwan_status END) AS sdwan status, (CASE WHEN link status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth, (CASE WHEN link status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth, (CASE WHEN link status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth from (select itime, csf, devname, devid, vd, interface, healthcheck, (CASE WHEN status='down'

THEN 0 ELSE 1 END) AS link_status, latency::float as latency, jitter::float as jitter, trim (trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num(outbandwidthused) as outbandwidth, convert_unit_to_num (bibandwidthused) as bibandwidth from \$log where \$filter and logid_to_int(logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf, devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filter-drilldown and sla_rule is not null group by intf_sla order by num_intf desc limit 10)t2 on t1.intf_sla=t2.intf_sla group by hodex, t1.intf_sla order by hodex

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

###(select \$flex timestamp as timestamp, csf, devname, devid, vd, interface, healthcheck as sla rule, sum(link status) as link status, sum(failed latency) as failed latency, sum (failed jitter) as failed jitter, sum(failed packetloss) as failed packetloss, sum(latency) as latency, max(latency) as latency max, min(latency) as latency min, sum(jitter) as jitter, max(jitter) as jitter max, min(jitter) as jitter min, sum(packetloss) as packetloss, max (packetloss) as packetloss max, min(packetloss) as packetloss min, sum(inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN link status=1 THEN 1 ELSE 0 END) AS count linkup, min(sdwan status) as sdwan status from (select itime, csf, devname, devid, vd, interface, healthcheck, link status, (CASE WHEN link status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link status=1 THEN packetloss ELSE 100 END) AS packetloss, (CASE WHEN sla failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed packetloss, (CASE WHEN sla failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed jitter, (CASE WHEN sla failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed latency, (CASE WHEN sla failed=1 THEN 3 ELSE sdwan status END) AS sdwan status, (CASE WHEN link status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth, (CASE WHEN link status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth, (CASE WHEN link status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth from (select itime, csf, devname, devid, vd, interface, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan status, convert unit to num (inbandwidthused) as inbandwidth, convert unit to num(outbandwidthused) as outbandwidth, convert unit to num(bibandwidthused) as bibandwidth from \$log where \$filter and logid to int (logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf, devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filter-drilldown group by sla rule, interface having sum(count linkup)>0 order by latency desc

Dataset Name	Description	Log Category
sdwan-device-sla-intf-jitter-pass- percent	SD-WAN Device Jitter Pass Percentage by SLA rules and Interface	event
select		
sla rule,		
interface,		
cast(
100 *(
1 - sum(failed_jitter)/	sum(count_linkup)	
) as decimal(18, 2)		
) as jitter		
from		
###(select \$flex_timestamp a	s timestamp, csf, devname, devid, vd, interface	e, healthcheck
as sla_rule, sum(link_status)	as link_status, sum(failed_latency) as failed_l	atency, sum
(failed_jitter) as failed_jitt	er, sum(failed_packetloss) as failed_packetlos	s, sum(latency)
as latency, max(latency) as latency_max, min(latency) as latency_min, sum(jitter) as jitter		
<pre>max(jitter) as jitter_max, min</pre>	(jitter) as jitter_min, sum(packetloss) as pack	etloss, max
(packetloss) as packetloss_max	, min(packetloss) as packetloss_min, sum(inband	lwidth) as
inbandwidth, sum(outbandwidth)	as outbandwidth, sum(bibandwidth) as bibandwid	lth, count(*) a
count, sum(CASE WHEN link stat	us=1 THEN 1 ELSE 0 END) AS count linkup, min(sc	lwan status) as

count, sum(CASE WHEN link_status=1 THEN 1 ELSE 0 END) AS count_linkup, min(sdwan_status) as sdwan status from (select itime, csf, devname, devid, vd, interface, healthcheck, link status, (CASE WHEN link_status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link_ status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link status=1 THEN packetloss ELSE 100 END) AS packetloss, (CASE WHEN sla failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed packetloss, (CASE WHEN sla failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed jitter, (CASE WHEN sla failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed latency, (CASE WHEN sla failed=1 THEN 3 ELSE sdwan status END) AS sdwan status, (CASE WHEN link status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth, (CASE WHEN link status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth, (CASE WHEN link status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth from (select itime, csf, devname, devid, vd, interface, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan status, convert unit to num (inbandwidthused) as inbandwidth, convert unit to num(outbandwidthused) as outbandwidth, convert unit to num(bibandwidthused) as bibandwidth from \$log where \$filter and logid to int (logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf, devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filter-drilldown group by sla rule, interface having sum(count linkup)>0 order by jitter desc

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

from

###(select \$flex_timestamp as timestamp, csf, devname, devid, vd, interface, healthcheck as sla rule, sum(link status) as link status, sum(failed latency) as failed latency, sum (failed jitter) as failed jitter, sum(failed packetloss) as failed packetloss, sum(latency) as latency, max(latency) as latency max, min(latency) as latency min, sum(jitter) as jitter, max(jitter) as jitter max, min(jitter) as jitter min, sum(packetloss) as packetloss, max (packetloss) as packetloss max, min(packetloss) as packetloss min, sum(inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN link status=1 THEN 1 ELSE 0 END) AS count linkup, min(sdwan status) as sdwan status from (select itime, csf, devname, devid, vd, interface, healthcheck, link status, (CASE WHEN link status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link status=1 THEN packetloss ELSE 100 END) AS packetloss, (CASE WHEN sla failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed packetloss, (CASE WHEN sla failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed jitter, (CASE WHEN sla failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed latency, (CASE WHEN sla failed=1 THEN 3 ELSE sdwan status END) AS sdwan status, (CASE WHEN link status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth, (CASE WHEN link status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth, (CASE WHEN link status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth from (select itime, csf, devname, devid, vd, interface, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan status, convert unit to num (inbandwidthused) as inbandwidth, convert unit to num(outbandwidthused) as outbandwidth, convert unit to num(bibandwidthused) as bibandwidth from \$log where \$filter and logid to int (logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf, devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filter-drilldown group by sla rule, interface having sum(count linkup)>0 order by packetloss desc

Dataset Name	Description	Log Category
sdwan-Device-Intf-List-by-Availability	SD-WAN Device Interface List by Availability	event

select

devname || & #039;:' || interface as dev intf, sum(count linkup)/sum(count) as available from ###(select \$flex timestamp as timestamp, csf, devname, devid, vd, interface, healthcheck as sla_rule, sum(link_status) as link_status, sum(failed_latency) as failed_ latency, sum(failed_jitter) as failed_jitter, sum(failed_packetloss) as failed_packetloss, sum(latency) as latency, max(latency) as latency max, min(latency) as latency min, sum (jitter) as jitter, max(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max(packetloss) as packetloss_max, min(packetloss) as packetloss_min, sum (inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN link status=1 THEN 1 ELSE 0 END) AS count linkup, min(sdwan status) as sdwan status from (select itime, csf, devname, devid, vd, interface, healthcheck, link status, (CASE WHEN link status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link status=1 THEN packetloss ELSE 100 END) AS packetloss, (CASE WHEN sla failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed packetloss, (CASE WHEN sla failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed jitter, (CASE WHEN sla failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed latency, (CASE WHEN sla failed=1 THEN 3 ELSE sdwan status END) AS sdwan status, (CASE WHEN link status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth, (CASE WHEN link status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth, (CASE WHEN link status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth from (select itime, csf, devname, devid, vd, interface, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from

packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_ unit_to_num(outbandwidthused) as outbandwidth, convert_unit_to_num(bibandwidthused) as bibandwidth from \$log where \$filter and logid_to_int(logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf, devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filter-drilldown group by dev_intf having sum(count)>0 order by dev_intf

Dataset Name	Description	Log Category
sdwan-Device-Intf-Updown-Timeline	SD-WAN Device Interface Updown Time Line	event

select

\$fv line timescale(timestamp) as hodex,

devname || & #039;:' || interface as dev intf, cast(100*sum(count linkup)/sum(count) as decimal(10,2)) as sdwan status from ###(select \$flex timestamp as timestamp, csf, devname, devid, vd, interface, healthcheck as sla_rule, sum(link_status) as link_status, sum(failed_ latency) as failed_latency, sum(failed_jitter) as failed_jitter, sum(failed_packetloss) as failed packetloss, sum(latency) as latency, max(latency) as latency_max, min(latency) as latency min, sum(jitter) as jitter, max(jitter) as jitter max, min(jitter) as jitter min, sum(packetloss) as packetloss, max(packetloss) as packetloss max, min(packetloss) as packetloss min, sum(inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum (bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN link status=1 THEN 1 ELSE 0 END) AS count linkup, min(sdwan status) as sdwan status from (select itime, csf, devname, devid, vd, interface, healthcheck, link status, (CASE WHEN link status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link status=1 THEN packetloss ELSE 100 END) AS packetloss, (CASE WHEN sla_failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed packetloss, (CASE WHEN sla failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed jitter, (CASE WHEN sla failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed latency, (CASE WHEN sla failed=1 THEN 3 ELSE sdwan status END) AS sdwan status, (CASE WHEN link status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth, (CASE WHEN link status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth, (CASE WHEN link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth from (select itime, csf, devname, devid, vd, interface, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msq LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan status, convert unit to num(inbandwidthused) as inbandwidth, convert unit to num(outbandwidthused) as outbandwidth, convert unit to num(bibandwidthused) as bibandwidth from \$log where \$filter and logid_to_int(logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf, devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t group by hodex, dev intf order by hodex

Dataset Name	Description	Log Categor
sdwan-Device-Availability-status	SD-WAN Device Statistic by Bibandwidth	event
<pre>select devid, sum(bibandwidth)/ sum(count) a from</pre>	s bibandwidth	

###(select \$flex_timestamp as timestamp, csf, devname, devid, vd, interface, healthcheck
as sla_rule, sum(link_status) as link_status, sum(failed_latency) as failed_latency, sum
(failed_jitter) as failed_jitter, sum(failed_packetloss) as failed_packetloss, sum(latency)

as latency, max(latency) as latency_max, min(latency) as latency_min, sum(jitter) as jitter, max(jitter) as jitter max, min(jitter) as jitter min, sum(packetloss) as packetloss, max (packetloss) as packetloss max, min(packetloss) as packetloss min, sum(inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN link status=1 THEN 1 ELSE 0 END) AS count linkup, min(sdwan status) as sdwan status from (select itime, csf, devname, devid, vd, interface, healthcheck, link status, (CASE WHEN link status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link status=1 THEN packetloss ELSE 100 END) AS packetloss, (CASE WHEN sla failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed packetloss, (CASE WHEN sla failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed jitter, (CASE WHEN sla failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed latency, (CASE WHEN sla failed=1 THEN 3 ELSE sdwan status END) AS sdwan status, (CASE WHEN link status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth, (CASE WHEN link status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth, (CASE WHEN link status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth from (select itime, csf, devname, devid, vd, interface, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan status, convert unit to num (inbandwidthused) as inbandwidth, convert unit to num(outbandwidthused) as outbandwidth, convert unit to num(bibandwidthused) as bibandwidth from \$log where \$filter and logid to int (logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf, devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filter-drilldown and bibandwidth is not null group by devid having sum(count)>0 order by bibandwidth desc

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

(select

& #039;SD-WAN' as interface, cast(sum(availcnt)*100.0/sum(count) as decimal(18,2)) as available from (select timestamp, devid, first value(count) OVER (PARTITION BY timestamp, devid ORDER BY link_status/count desc, count desc) as count, first_value(link_status) OVER (PARTITION BY timestamp, devid ORDER BY link status/count desc, count desc) as availant from (select timestamp, devid, interface, sum(link_status) as link_status, sum(count) as count from ###(select \$flex timestamp as timestamp, csf, devname, devid, vd, interface, healthcheck as sla rule, sum(link status) as link status, sum(failed latency) as failed latency, sum(failed jitter) as failed jitter, sum(failed packetloss) as failed packetloss, sum(latency) as latency, max(latency) as latency max, min(latency) as latency min, sum (jitter) as jitter, max(jitter) as jitter max, min(jitter) as jitter min, sum(packetloss) as packetloss, max(packetloss) as packetloss max, min(packetloss) as packetloss min, sum (inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN link status=1 THEN 1 ELSE 0 END) AS count linkup, min(sdwan status) as sdwan status from (select itime, csf, devname, devid, vd, interface, healthcheck, link status, (CASE WHEN link status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link status=1 THEN packetloss ELSE 100 END) AS packetloss, (CASE WHEN sla failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed packetloss, (CASE WHEN sla_failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed jitter, (CASE WHEN sla failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed latency, (CASE WHEN sla failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status, (CASE WHEN link_status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth, (CASE WHEN link status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth, (CASE WHEN link status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth from (select itime, csf,

devname, devid, vd, interface, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan status, convert unit to num(inbandwidthused) as inbandwidth, convert unit to num(outbandwidthused) as outbandwidth, convert unit to num(bibandwidthused) as bibandwidth from \$log where \$filter and logid to int(logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf, devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filter-drilldown and count>0 group by timestamp, devid, interface)t) t group by interface) union all (select interface, cast(sum(link status)*100.0/sum(count) as decimal(18,2)) as available from ### (select \$flex timestamp as timestamp, csf, devname, devid, vd, interface, healthcheck as sla rule, sum(link status) as link status, sum(failed latency) as failed latency, sum (failed jitter) as failed jitter, sum(failed packetloss) as failed packetloss, sum(latency) as latency, max(latency) as latency max, min(latency) as latency min, sum(jitter) as jitter, max(jitter) as jitter max, min(jitter) as jitter min, sum(packetloss) as packetloss, max (packetloss) as packetloss max, min(packetloss) as packetloss min, sum(inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN link status=1 THEN 1 ELSE 0 END) AS count linkup, min(sdwan status) as sdwan status from (select itime, csf, devname, devid, vd, interface, healthcheck, link status, (CASE WHEN link status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link status=1 THEN packetloss ELSE 100 END) AS packetloss, (CASE WHEN sla failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed packetloss, (CASE WHEN sla failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed jitter, (CASE WHEN sla failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed latency, (CASE WHEN sla failed=1 THEN 3 ELSE sdwan status END) AS sdwan status, (CASE WHEN link status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth, (CASE WHEN link status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth, (CASE WHEN link status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth from (select itime, csf, devname, devid, vd, interface, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan status, convert unit to num (inbandwidthused) as inbandwidth, convert unit to num(outbandwidthused) as outbandwidth, convert unit to num(bibandwidthused) as bibandwidth from \$log where \$filter and logid to int (logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf, devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filter-drilldown group by interface order by interface)

Dataset Name Description Log Category sdwan-device-intf-availability-SD-WAN Device Interface Availability Percentage Donut event percentage-donut select interface, unnest(avail) as avail, unnest(val) as val from (select interface, array[& #039; Available', 'Unavailable'] as avail, array[available, 100-available] as val from ((select 'SD-WAN' as interface, cast(sum(availcnt)*100.0/sum(count) as decimal (18,2)) as available from (select timestamp, devid, first_value(count) OVER (PARTITION BY timestamp, devid ORDER BY link status/count desc, count desc) as count, first value(link

status) OVER (PARTITION BY timestamp, devid ORDER BY link status/count desc, count desc) as availcnt from (select timestamp, devid, interface, sum(link status) as link status, sum (count) as count from ###(select \$flex timestamp as timestamp, csf, devname, devid, vd, interface, healthcheck as sla rule, sum(link status) as link status, sum(failed latency) as failed_latency, sum(failed_jitter) as failed_jitter, sum(failed_packetloss) as failed_ packetloss, sum(latency) as latency, max(latency) as latency max, min(latency) as latency min, sum(jitter) as jitter, max(jitter) as jitter max, min(jitter) as jitter min, sum (packetloss) as packetloss, max(packetloss) as packetloss max, min(packetloss) as packetloss min, sum(inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum (bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN link status=1 THEN 1 ELSE 0 END) AS count linkup, min(sdwan status) as sdwan status from (select itime, csf, devname, devid, vd, interface, healthcheck, link status, (CASE WHEN link status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link status=1 THEN packetloss ELSE 100 END) AS packetloss, (CASE WHEN sla failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed packetloss, (CASE WHEN sla failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed jitter, (CASE WHEN sla failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed latency, (CASE WHEN sla failed=1 THEN 3 ELSE sdwan status END) AS sdwan status, (CASE WHEN link status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth, (CASE WHEN link status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth, (CASE WHEN link status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth from (select itime, csf, devname, devid, vd, interface, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_ unit to num(outbandwidthused) as outbandwidth, convert unit to num(bibandwidthused) as bibandwidth from \$log where \$filter and logid to int(logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf, devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filter-drilldown and count>0 group by timestamp, devid, interface)t) t group by interface) union all (select interface, cast(sum(link status)*100.0/sum(count) as decimal(18,2)) as available from ### (select \$flex timestamp as timestamp, csf, devname, devid, vd, interface, healthcheck as sla rule, sum(link status) as link status, sum(failed latency) as failed latency, sum (failed jitter) as failed jitter, sum(failed packetloss) as failed packetloss, sum(latency) as latency, max(latency) as latency max, min(latency) as latency min, sum(jitter) as jitter, max(jitter) as jitter max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max (packetloss) as packetloss max, min(packetloss) as packetloss min, sum(inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN link status=1 THEN 1 ELSE 0 END) AS count linkup, min(sdwan status) as sdwan status from (select itime, csf, devname, devid, vd, interface, healthcheck, link status, (CASE WHEN link status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link status=1 THEN packetloss ELSE 100 END) AS packetloss, (CASE WHEN sla failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed packetloss, (CASE WHEN sla failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed jitter, (CASE WHEN sla failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed latency, (CASE WHEN sla failed=1 THEN 3 ELSE sdwan status END) AS sdwan status, (CASE WHEN link status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth, (CASE WHEN link status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth, (CASE WHEN link_status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth from (select itime, csf, devname, devid, vd, interface, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan status, convert unit to num (inbandwidthused) as inbandwidth, convert unit to num(outbandwidthused) as outbandwidth, convert unit to num(bibandwidthused) as bibandwidth from \$log where \$filter and logid to int (logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf,

devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp
desc/*SkipEND*/)### t where \$filter-drilldown group by interface order by interface)) t) t

Dataset Name	Description	Log Category
sdwan-Device-Application-sdwan- Rules-and-Ports-drilldown	SD-WAN Device Statistic by Bibandwidth	event

select

devid, sum(bibandwidth) / sum(count) as bibandwidth from

###(select \$flex_timestamp as timestamp, csf, devname, devid, vd, interface, healthcheck as sla rule, sum(link status) as link status, sum(failed latency) as failed latency, sum (failed jitter) as failed jitter, sum(failed packetloss) as failed packetloss, sum(latency) as latency, max(latency) as latency max, min(latency) as latency min, sum(jitter) as jitter, max(jitter) as jitter max, min(jitter) as jitter min, sum(packetloss) as packetloss, max (packetloss) as packetloss max, min(packetloss) as packetloss min, sum(inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN link status=1 THEN 1 ELSE 0 END) AS count linkup, min(sdwan status) as sdwan status from (select itime, csf, devname, devid, vd, interface, healthcheck, link status, (CASE WHEN link status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link status=1 THEN packetloss ELSE 100 END) AS packetloss, (CASE WHEN sla failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed packetloss, (CASE WHEN sla failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed jitter, (CASE WHEN sla failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed latency, (CASE WHEN sla failed=1 THEN 3 ELSE sdwan status END) AS sdwan status, (CASE WHEN link status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth, (CASE WHEN link status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth, (CASE WHEN link status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth from (select itime, csf, devname, devid, vd, interface, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan status, convert unit to num (inbandwidthused) as inbandwidth, convert unit to num(outbandwidthused) as outbandwidth, convert unit to num(bibandwidthused) as bibandwidth from \$log where \$filter and logid to int (logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf, devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filter-drilldown and bibandwidth is not null group by devid having sum(count)>0 order by bibandwidth desc

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

select

& #039;SD-WAN Rules' as summary, 'Rule:' || coalesce(rulename, 'Unknown') as rule_name, app_group, devid, dstintf as interface, sum(bandwidth) as bandwidth from ###(select \$flex_ timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group_name(app) as app_group, coalesce(vwlname,vwlservice) as rulename, service, coalesce(nullifna(`srcname`),ipstr(`srcip`),nullifna(`srcmac`)) as dev_src, sum(crscore%65536) as crscore, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum (coalesce(sentdelta, sentbyte, 0)) as traffic out, count(*) as sessions from \$log-traffic where \$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf, devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_src, dev_src order by bandwidth desc)### t where \$filter-drilldown group by rule_name, app_group, devid, interface order by bandwidth desc

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

select

devid,

sum(bibandwidth) / sum(count) as bibandwidth
from

###(select \$flex timestamp as timestamp, csf, devname, devid, vd, interface, healthcheck as sla rule, sum(link status) as link status, sum(failed latency) as failed latency, sum (failed jitter) as failed jitter, sum(failed packetloss) as failed packetloss, sum(latency) as latency, max(latency) as latency max, min(latency) as latency min, sum(jitter) as jitter, max(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max (packetloss) as packetloss_max, min(packetloss) as packetloss_min, sum(inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN link status=1 THEN 1 ELSE 0 END) AS count linkup, min(sdwan status) as sdwan status from (select itime, csf, devname, devid, vd, interface, healthcheck, link status, (CASE WHEN link status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link status=1 THEN packetloss ELSE 100 END) AS packetloss, (CASE WHEN sla failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed packetloss, (CASE WHEN sla failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed jitter, (CASE WHEN sla failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed latency, (CASE WHEN sla failed=1 THEN 3 ELSE sdwan status END) AS sdwan status, (CASE WHEN link status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth, (CASE WHEN link status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth, (CASE WHEN link status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth from (select itime, csf, devname, devid, vd, interface, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan status, convert unit to num (inbandwidthused) as inbandwidth, convert unit to num(outbandwidthused) as outbandwidth, convert unit to num(bibandwidthused) as bibandwidth from \$log where \$filter and logid to int (logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf, devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filter-drilldown and bibandwidth is not null group by devid having sum(count)>0 order by bibandwidth desc

Dataset Name	Description	Log Category
sdwan-Device-Intf-Avail-Percentage- Timeline	SD-WAN Device Interface Availability Percentage Timeline	event
<pre>select hodex, interface, available from ((select \$flex_datetime(timestamp)</pre>) as hodex,	

& #039;SD-WAN' as interface, cast(sum(availcnt)*100.0/sum(count) as decimal(18,2)) as available from (select timestamp, devid, first value(count) OVER (PARTITION BY timestamp, devid ORDER BY link status/count desc, count desc) as count, first value(link status) OVER (PARTITION BY timestamp, devid ORDER BY link status/count desc, count desc) as availant from (select timestamp, devid, interface, sum(link status) as link status, sum(count) as count from ###(select \$flex timestamp as timestamp, csf, devname, devid, vd, interface, healthcheck as sla rule, sum(link status) as link status, sum(failed_latency) as failed_ latency, sum(failed jitter) as failed jitter, sum(failed packetloss) as failed packetloss, sum(latency) as latency, max(latency) as latency max, min(latency) as latency min, sum (jitter) as jitter, max(jitter) as jitter max, min(jitter) as jitter min, sum(packetloss) as packetloss, max(packetloss) as packetloss max, min(packetloss) as packetloss min, sum (inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN link status=1 THEN 1 ELSE 0 END) AS count linkup, min(sdwan status) as sdwan status from (select itime, csf, devname, devid, vd, interface, healthcheck, link status, (CASE WHEN link status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link status=1 THEN packetloss ELSE 100 END) AS packetloss, (CASE WHEN sla failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed packetloss, (CASE WHEN sla failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed jitter, (CASE WHEN sla failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed latency, (CASE WHEN sla_failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status, (CASE WHEN link_status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth, (CASE WHEN link status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth, (CASE WHEN link status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth from (select itime, csf, devname, devid, vd, interface, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_ unit to num(outbandwidthused) as outbandwidth, convert unit to num(bibandwidthused) as bibandwidth from \$log where \$filter and logid to int(logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf, devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filter-drilldown and count>0 group by timestamp, devid, interface)t) t group by hodex order by hodex) union all (select \$flex datetime(timestamp) as hodex, interface, cast(sum(link status)*100.0/sum (count) as decimal(18,2)) as available from ###(select \$flex timestamp as timestamp, csf, devname, devid, vd, interface, healthcheck as sla rule, sum(link status) as link status, sum (failed latency) as failed latency, sum(failed jitter) as failed jitter, sum(failed packetloss) as failed packetloss, sum(latency) as latency, max(latency) as latency max, min (latency) as latency min, sum(jitter) as jitter, max(jitter) as jitter max, min(jitter) as jitter min, sum(packetloss) as packetloss, max(packetloss) as packetloss max, min (packetloss) as packetloss min, sum(inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN link status=1 THEN 1 ELSE 0 END) AS count linkup, min(sdwan status) as sdwan status from (select itime, csf, devname, devid, vd, interface, healthcheck, link_status, (CASE WHEN link_ status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link status=1 THEN packetloss ELSE 100 END) AS packetloss, (CASE WHEN sla failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed packetloss, (CASE WHEN sla failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed jitter, (CASE WHEN sla failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed latency, (CASE WHEN sla failed=1 THEN 3 ELSE sdwan status END) AS sdwan status, (CASE WHEN link status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth, (CASE WHEN link_status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth, (CASE WHEN link status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth from (select itime, csf, devname, devid, vd, interface, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link status, latency::float as latency, jitter::float as jitter, trim (trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla failed, metric, (CASE WHEN msg LIKE '%SLA

status%' THEN 1 ELSE 0 END) AS sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num(outbandwidthused) as outbandwidth, convert_unit_to_num (bibandwidthused) as bibandwidth from \$log where \$filter and logid_to_int(logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf, devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filterdrilldown group by hodex, interface order by hodex)) t order by hodex

Dataset Name	Description	Log Category
sdwan-Device-Intf-Inbandwidth- Timeline	SD-WAN Device-Interface Inbandwidth Timeline	event
<pre>select \$flex_timescale(timestamp) a t1_interface</pre>	as time,	

```
tl.interface,
cast(
   sum(inbandwidth) / sum(count) as decimal(18, 2)
) as inbandwidth
from
   (
   select
    timestamp,
    devid,
    interface,
    sum(count) as count,
    sum(inbandwidth) as inbandwidth
```

from

###(select \$flex timestamp as timestamp, csf, devname, devid, vd, interface, healthcheck as sla rule, sum(link status) as link status, sum(failed latency) as failed latency, sum(failed jitter) as failed jitter, sum(failed packetloss) as failed packetloss, sum(latency) as latency, max(latency) as latency max, min(latency) as latency min, sum (jitter) as jitter, max(jitter) as jitter_max, min(jitter) as jitter_min, sum(packetloss) as packetloss, max(packetloss) as packetloss max, min(packetloss) as packetloss min, sum (inbandwidth) as inbandwidth, sum(outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN link status=1 THEN 1 ELSE 0 END) AS count linkup, min(sdwan_status) as sdwan_status from (select itime, csf, devname, devid, vd, interface, healthcheck, link status, (CASE WHEN link status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link status=1 THEN packetloss ELSE 100 END) AS packetloss, (CASE WHEN sla failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed packetloss, (CASE WHEN sla failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed jitter, (CASE WHEN sla failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed latency, (CASE WHEN sla failed=1 THEN 3 ELSE sdwan status END) AS sdwan status, (CASE WHEN link status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth, (CASE WHEN link status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth, (CASE WHEN link status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth from (select itime, csf, devname, devid, vd, interface, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link_status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan status, convert unit to num(inbandwidthused) as inbandwidth, convert unit to num(outbandwidthused) as outbandwidth, convert unit to num(bibandwidthused) as bibandwidth from \$log where \$filter and logid to int(logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf, devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t group by timestamp, devid, interface) tl inner join (select devid, interface, count(*) as num_intf from ###(select \$flex timestamp as timestamp, csf, devname, devid, vd, interface, healthcheck as sla rule,

sum(link status) as link status, sum(failed latency) as failed latency, sum(failed jitter) as failed jitter, sum(failed packetloss) as failed packetloss, sum(latency) as latency, max (latency) as latency max, min(latency) as latency min, sum(jitter) as jitter, max(jitter) as jitter max, min(jitter) as jitter min, sum(packetloss) as packetloss, max(packetloss) as packetloss max, min(packetloss) as packetloss min, sum(inbandwidth) as inbandwidth, sum (outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN link status=1 THEN 1 ELSE 0 END) AS count linkup, min(sdwan status) as sdwan status from (select itime, csf, devname, devid, vd, interface, healthcheck, link status, (CASE WHEN link status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link status=1 THEN packetloss ELSE 100 END) AS packetloss, (CASE WHEN sla failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed packetloss, (CASE WHEN sla failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed jitter, (CASE WHEN sla failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed latency, (CASE WHEN sla failed=1 THEN 3 ELSE sdwan status END) AS sdwan status, (CASE WHEN link status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth, (CASE WHEN link status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth, (CASE WHEN link status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth from (select itime, csf, devname, devid, vd, interface, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert unit to num (outbandwidthused) as outbandwidth, convert unit to num (bibandwidthused) as bibandwidth from \$log where \$filter and logid to int(logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf, devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filterdrilldown group by devid, interface order by num intf desc limit 10)t2 on t1.interface=t2.interface and t1.devid=t2.devid group by time, t1.interface having sum (count)>0 order by time

Dataset Name	Description	Log Category
sdwan-Device-Intf-Outbandwidth- Timeline	SD-WAN Device-Interface Outbandwidth Timeline	event
<pre>select \$flex_timescale(timestamp) as t1.interface, cast(sum(outbandwidth) / sum(cour) as outbandwidth from (select timestamp, devid, interface, sum(count) as count, sum(outbandwidth) as outb from</pre>	nt) as decimal(18, 2)	
healthcheck as sla_rule, sum(li latency, sum(failed_jitter) as sum(latency) as latency, max(la (jitter) as jitter, max(jitter) packetloss, max(packetloss) as	<pre>mp as timestamp, csf, devname, devid, vd, inte ink_status) as link_status, sum(failed_latency failed_jitter, sum(failed_packetloss) as fai atency) as latency_max, min(latency) as latence) as jitter_max, min(jitter) as jitter_min, su packetloss_max, min(packetloss) as packetloss sum(outbandwidth) as outbandwidth, sum(bibandw</pre>) as failed_ led_packetloss, y_min, sum m(packetloss) as _min, sum

bibandwidth, count(*) as count, sum(CASE WHEN link status=1 THEN 1 ELSE 0 END) AS count linkup, min(sdwan status) as sdwan status from (select itime, csf, devname, devid, vd, interface, healthcheck, link status, (CASE WHEN link status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link status=1 THEN packetloss ELSE 100 END) AS packetloss, (CASE WHEN sla failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed packetloss, (CASE WHEN sla failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed jitter, (CASE WHEN sla failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed latency, (CASE WHEN sla failed=1 THEN 3 ELSE sdwan status END) AS sdwan status, (CASE WHEN link status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth, (CASE WHEN link status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth, (CASE WHEN link status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth from (select itime, csf, devname, devid, vd, interface, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla_failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan status, convert unit to num(inbandwidthused) as inbandwidth, convert unit to num(outbandwidthused) as outbandwidth, convert unit to num(bibandwidthused) as bibandwidth from \$log where \$filter and logid to int(logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf, devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t group by timestamp, devid, interface) t1 inner join (select devid, interface, count(*) as num_intf from ###(select \$flex timestamp as timestamp, csf, devname, devid, vd, interface, healthcheck as sla rule, sum(link status) as link status, sum(failed latency) as failed latency, sum(failed jitter) as failed jitter, sum(failed packetloss) as failed packetloss, sum(latency) as latency, max (latency) as latency_max, min(latency) as latency_min, sum(jitter) as jitter, max(jitter) as jitter max, min(jitter) as jitter min, sum(packetloss) as packetloss, max(packetloss) as packetloss max, min(packetloss) as packetloss min, sum(inbandwidth) as inbandwidth, sum (outbandwidth) as outbandwidth, sum(bibandwidth) as bibandwidth, count(*) as count, sum(CASE WHEN link status=1 THEN 1 ELSE 0 END) AS count linkup, min(sdwan status) as sdwan status from (select itime, csf, devname, devid, vd, interface, healthcheck, link status, (CASE WHEN link status=1 THEN latency ELSE 0 END) AS latency, (CASE WHEN link status=1 THEN jitter ELSE 0 END) AS jitter, (CASE WHEN link status=1 THEN packetloss ELSE 100 END) AS packetloss, (CASE WHEN sla failed=1 AND metric='packetloss' THEN 1 ELSE 0 END) AS failed packetloss, (CASE WHEN sla failed=1 AND metric='jitter' THEN 1 ELSE 0 END) AS failed jitter, (CASE WHEN sla_failed=1 AND metric='latency' THEN 1 ELSE 0 END) AS failed_latency, (CASE WHEN sla_ failed=1 THEN 3 ELSE sdwan_status END) AS sdwan_status, (CASE WHEN link_status=1 THEN inbandwidth ELSE 0 END) AS inbandwidth, (CASE WHEN link status=1 THEN outbandwidth ELSE 0 END) AS outbandwidth, (CASE WHEN link status=1 THEN bibandwidth ELSE 0 END) AS bibandwidth from (select itime, csf, devname, devid, vd, interface, healthcheck, (CASE WHEN status='down' THEN 0 ELSE 1 END) AS link status, latency::float as latency, jitter::float as jitter, trim(trailing '%' from packetloss)::float as packetloss, (CASE WHEN status='down' THEN 1 WHEN msg LIKE '%SLA failed%' THEN 1 ELSE 0 END) AS sla failed, metric, (CASE WHEN msg LIKE '%SLA status%' THEN 1 ELSE 0 END) AS sdwan_status, convert_unit_to_num(inbandwidthused) as inbandwidth, convert_unit_to_num(outbandwidthused) as outbandwidth, convert_unit_to_num (bibandwidthused) as bibandwidth from \$log where \$filter and logid to int(logid) in (22925, 22933, 22936) and interface is not null) t) t group by timestamp, csf, devname, devid, vd, interface, healthcheck /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filterdrilldown group by devid, interface order by num intf desc limit 10)t2 on t1.interface=t2.interface and t1.devid=t2.devid group by time, t1.interface having sum (count) >0 order by time

Dataset Name	Description	Log Category
Top-Web-Sites-by-Bandwidth	Top web sites by bandwidth usage	webfilter

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

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

Dataset Name	Description	Log Category
Top-App-Category-by-Session	Application risk application usage by category	traffic

select

```
appcat,
sum(sessions) as total num
```

from

###(select appid, app, appcat, apprisk, sum(bandwidth) as bandwidth, sum(sessions) as sessions from ###base(/*tag:rpt_base_t_top_app*/select dvid, srcip, dstip, epid, euid, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, appid, app, appcat, apprisk, hostname, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce (sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce (rcvddelta, rcvdbyte, 0)) as bandwidth, count(*) as sessions from \$log-traffic where \$filter and (logflag&(1|32)>0) and nullifna(app) is not null group by dvid, srcip, dstip, epid, euid, user_src, appid, app, appcat, apprisk, hostname order by sessions desc)base### t group by appid, app, appcat, apprisk /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t where \$filter-drilldown group by appcat order by total_num desc

Dataset Name	Description	Log Category
Top-Region-Name-by-Traffic	Traffic top destination countries by browsing time	traffic
select dstcountry, sum(bandwidth) as bandwidth		

from

###(select dstcountry, ebtr_agg_flat(browsetime) as browsetime, sum(bandwidth) as bandwidth, sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_out from (select dstcountry, ebtr_agg_flat(\$browse_time) as browsetime, sum(coalesce(sentbyte, 0)+coalesce (rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce (sentbyte, 0)) as traffic_out from \$log where \$filter and (logflag&1>0) and \$browse_time is not null group by dstcountry) t group by dstcountry /*SkipSTART*/order by ebtr_value(ebtr_ agg_flat(browsetime), null, null) desc/*SkipEND*/)### t where \$filter-drilldown group by dstcountry order by bandwidth desc

Dataset Name	Description	Log Category
Top-App-By-Bandwidth-Chart	Top applications by bandwidth usage	traffic
<pre>select app_group_name(app) as app_ sum(bandwidth) as bandwidth</pre>		

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

```
sum(sessions) as sessions
from
```

###(select appid, app, appcat, apprisk, sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_out, sum(bandwidth) as bandwidth, sum(sessions) as sessions from ###base(/*tag:rpt_ base_t_top_app*/select dvid, srcip, dstip, epid, euid, coalesce(nullifna(`user`), nullifna (`unauthuser`), ipstr(`srcip`)) as user_src, appid, app, appcat, apprisk, hostname, sum (coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, count(*) as sessions from \$log-traffic where \$filter and (logflag&(1|32)>0) and nullifna(app) is not null group by dvid, srcip, dstip, epid, euid, user_src, appid, app, appcat, apprisk, hostname order by sessions desc)base### t group by appid, app, appcat, apprisk /*SkipSTART*/order by sessions desc, bandwidth desc/*SkipEND*/)### t group by app_ group having sum(bandwidth)>0 order by bandwidth desc

Dataset Name	Description	Log Category
Top-Protocols-By-Traffic	Top applications by bandwidth usage	traffic

select

```
service,
```

sum (bandwidth) as bandwidth

from

###(select service, sum(bandwidth) as bandwidth from ###base(/*tag:rpt_base_t_bndwdth_ sess*/select \$flex_timestamp as timestamp, dvid, srcip, dstip, epid, euid, coalesce(nullifna (`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, service, count(*) as sessions, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in from \$log-traffic where \$filter and (logflag&(1|32)>0) group by timestamp, dvid, srcip, dstip, epid, euid, user_src, service /*SkipSTART*/order by timestamp desc/*SkipEND*/)base### base_query group by service order by bandwidth desc)### t where \$filter-drilldown group by service order by bandwidth desc

Dataset Name	Description	Log Category
Top-Web-Sites-by-Sessions	Top web sites by session count	webfilter

select

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

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

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

select

attack, sum(attack_count) as totalnum from

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

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

```
user src,
 sum(totalnum) as totalnum
from
```

####(select \$flex timestamp as timestamp, coalesce(nullifna(`user`), ipstr(`srcip`)) as user src, `from` as mf sender, `to` as mf receiver, action, eventtype, count(*) as totalnum from \$log where \$filter group by timestamp, user src, mf sender, mf receiver, action, eventtype /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filter-drilldown and mf sender is not null group by user src order by totalnum desc

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

select

```
virus,
max(virusid s) as virusid,
```

case when virus like & #039;Riskware%' then 'Spyware' when virus like 'Adware%' then 'Adware' else 'Virus' end) as malware type, sum(totalnum) as totalnum from ###(select virus, virusid to str(virusid, eventtype) as virusid s, count(*) as totalnum from \$log where \$filter and (eventtype is null or logver>=502000000) and nullifna(virus) is not null group by virus, virusid s /*SkipSTART*/order by totalnum desc/*SkipEND*/)### t group by virus, malware type order by totalnum desc

Dataset Name	Description	Log Category
security-Antivirus-Inspections	Antivirus Inspections	virus
<pre>select action, sum(totalnum) as totalnum from</pre>		

####(select \$flex timestamp as timestamp, coalesce(nullifna(`user`), ipstr(`srcip`)) as user src, `from` as mf sender, `to` as mf receiver, action, eventtype, count(*) as totalnum from \$log where \$filter group by timestamp, user src, mf sender, mf receiver, action, eventtype /*SkipSTART*/order by timestamp desc/*SkipEND*/)### t where \$filter-drilldown and action is not null group by action order by totalnum desc

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

severity='critical' then 'Critical Data Exfiltration' else (case when coalesce(nullifna
(`user`), ipstr(`srcip`)) is not null then 'User Associated Data Loss' else NULL end) end)
as data_loss from \$log where \$filter /*SkipSTART*/order by itime desc/*SkipEND*/)### t where
\$filter-drilldown and profile is not null group by profile order by total_num desc

Dataset Name	Description	Log Category
wifi-Top-AP-By-Client	Top access point by client	traffic
select		
ap_srcintf as srcintf,		
count(distinct srcmac) as	totalnum	
from		
(
select		
<pre>coalesce(ap, srcintf)</pre>	as ap_srcintf,	
srcmac		
from		
###(select coalesce(n	ullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_
<pre>src, ap, srcintf, srcssid,</pre>	srcssid as ssid, srcmac, srcmac as sta	mac, coalesce(nullifna
(`srcname`), `srcmac`) as h	ostname_mac, max(srcswversion) as srcs	wversion, max(osname) as
osname, max(osversion) as o	sversion, max(devtype) as devtype, sum	(coalesce(sentbyte,
0)+coalesce(rcvdbyte, 0)) as bandwidth, count(*) as subtotal from \$log-traffic where \$filter		
and (logflag&1>0) and (srcssid is not null or dstssid is not null) group by user_src, ap,		
srcintf, srcssid, srcmac, h	ostname_mac /*SkipSTART*/order by band	width desc, subtotal
<pre>desc/*SkipEND*/)### t where</pre>	srcmac is not null group by ap srcint	f, srcmac union all (select

desc/*SkipEND*/)### t where srcmac is not null group by ap_srcintf, srcmac union all (select ap as ap_srcintf, stamac as srcmac from ###(select \$flex_timestamp as timestamp, stamac, stamac as srcmac, ap, ssid, ssid as srcssid, user_src, sum(coalesce(sentdelta, 0)) as sentdelta, sum(coalesce(rcvddelta, 0)) as rcvddelta, sum(coalesce(sentdelta, 0)+coalesce (rcvddelta, 0)) as bandwidth from (select itime, stamac, ap, ssid, coalesce(`user`, ipstr (`srcip`)) as user_src, sentbyte-lag(coalesce(sentbyte, 0)) over (partition by stamac order by itime) as sentdelta, rcvdbyte-lag(coalesce(rcvdbyte, 0)) over (partition by stamac order by itime) as rcvddelta from \$log-event where \$filter and subtype='wireless' and stamac is not null and ssid is not null and action in ('sta-wl-bridge-traffic-stats', 'reassoc-req', 'assoc-req')) as t group by timestamp, stamac, ap, ssid, user_src /*SkipSTART*/order by bandwidth desc/*SkipEND*/)### t where stamac is not null group by ap, stamac)) t group by srcintf order by totalnum desc

Dataset Name	Description	Log Category
wifi-Top-AP-By-Bandwidth	Top access point by bandwidth usage	traffic
<pre>select ap_srcintf, sum(bandwidth) as bandwidth from (select coalesce(ap, srcintf) as sum(bandwidth) as bandwi from</pre>		
<pre>src, ap, srcintf, srcssid, src (`srcname`), `srcmac`) as host osname, max(osversion) as osve</pre>	ifna(`user`), nullifna(`unauthuser`), ips ssid as ssid, srcmac, srcmac as stamac, c name_mac, max(srcswversion) as srcswversi rsion, max(devtype) as devtype, sum(coale andwidth, count(*) as subtotal from \$log-	coalesce(nullifna .on, max(osname) as esce(sentbyte,

and (logflag&1>0) and (srcssid is not null or dstssid is not null) group by user_src, ap, srcintf, srcssid, srcmac, hostname_mac /*SkipSTART*/order by bandwidth desc, subtotal desc/*SkipEND*/)### t group by ap_srcintf having sum(bandwidth)>0 union all select ap as ap_ srcintf, sum(bandwidth) as bandwidth from ###(select \$flex_timestamp as timestamp, stamac, stamac as srcmac, ap, ssid, ssid as srcssid, user_src, sum(coalesce(sentdelta, 0)) as sentdelta, sum(coalesce(rcvddelta, 0)) as rcvddelta, sum(coalesce(sentdelta, 0)+coalesce (rcvddelta, 0)) as bandwidth from (select itime, stamac, ap, ssid, coalesce(`user`, ipstr (`srcip`)) as user_src, sentbyte-lag(coalesce(sentbyte, 0)) over (partition by stamac order by itime) as sentdelta, rcvdbyte-lag(coalesce(rcvdbyte, 0)) over (partition by stamac order by itime) as rcvddelta from \$log-event where \$filter and subtype='wireless' and stamac is not null and ssid is not null and action in ('sta-wl-bridge-traffic-stats', 'reassoc-req', 'assoc-req')) as t group by timestamp, stamac, ap, ssid, user_src /*SkipSTART*/order by bandwidth desc/*SkipEND*/)### t group by ap having sum(bandwidth)>0) t group by ap_srcintf order by bandwidth desc

Dataset Name	Description	Log Category
wifi-Top-SSID-By-Bandwidth	Top SSIDs by bandwidth usage	traffic
<pre>src, ap, srcintf, srcssid, srcs (`srcname`), `srcmac`) as host osname, max(osversion) as osver 0)+coalesce(rcvdbyte, 0)) as ba and (logflag&l>0) and (srcssid srcintf, srcssid, srcmac, host desc/*SkipEND*/)### t where src union all select ssid as srcssi timestamp as timestamp, stamac, (coalesce(sentdelta, 0)) as ser (coalesce(sentdelta, 0)+coalesc ssid, coalesce(`user`, ipstr(`s (partition by stamac order by i subtype='wireless' and stamac i bridge-traffic-stats', 'reasso</pre>	ifna(`user`), nullifna(`unauthus said as ssid, srcmac, srcmac as name_mac, max(srcswversion) as s rsion, max(devtype) as devtype, andwidth, count(*) as subtotal : is not null or dstssid is not n name_mac /*SkipSTART*/order by b cssid is not null group by srcss id, sum(bandwidth) as bandwidth stamac as srcmac, ap, ssid, se ntdelta, sum(coalesce(rcvddelta, ce(rcvddelta, 0)) as bandwidth : srcip`)) as user_src, sentbyte- time) as sentdelta, rcvdbyte-la time) as rcvddelta from \$log-ev is not null and ssid is not null c-req', 'assoc-req')) as t group der by bandwidth desc/*SkipEND*,	<pre>stamac, coalesce(nullifna srcswversion, max(osname) as sum(coalesce(sentbyte, from \$log-traffic where \$filter null) group by user_src, ap, pandwidth desc, subtotal sid having sum(bandwidth)>0 from ###(select \$flex_ sid as srcssid, user_src, sum , 0)) as rcvddelta, sum from (select itime, stamac, ap, lag(coalesce(sentbyte, 0)) over ag(coalesce(rcvdbyte, 0)) over vent where \$filter and l and action in ('sta-wl- p by timestamp, stamac, ap,</pre>

Dataset Name	Description	Log Category
sdwan-CTAP-Total-Bandwidth- Internal-And-External	CTAP SD-WAN Internal and External Bandwidth	traffic
select interface, bandwidth		

```
from
```

```
(
```

(select

& #039; Internal' as interface, coalesce(sum(bandwidth), 0) as bandwidth from ### (select \$flex timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf, srcintfrole, dstintfrole, appid, appcat, app group name(app) as app group, coalesce (vwlname,vwlservice) as rulename, service, coalesce(nullifna(`srcname`),ipstr (`srcip`),nullifna(`srcmac`)) as dev src, sum(crscore%65536) as crscore, coalesce(nullifna (`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic out, count(*) as sessions from \$log-traffic where \$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf, devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app group, rulename, service, user src, dev src order by bandwidth desc) ### t where \$filter-drilldown and dstintfrole='lan') union all (select 'External' as interface, coalesce(sum(bandwidth), 0) as bandwidth from ###(select \$flex timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf, srcintfrole, dstintfrole, appid, appcat, app group name(app) as app group, coalesce(vwlname,vwlservice) as rulename, service, coalesce(nullifna (`srcname`),ipstr(`srcip`),nullifna(`srcmac`)) as dev_src, sum(crscore%65536) as crscore, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce (sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce (rcvddelta, rcvdbyte, 0)) as traffic in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic out, count(*) as sessions from \$log-traffic where \$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf, devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app group, rulename, service, user src, dev src order by bandwidth desc)### t where \$filter-drilldown and dstintfrole='wan')) t where bandwidth>0

Dataset Name	Description	Log Category
sdwan-CTAP-Total-Bandwidth- External-Business-nonBusiness- Network	CTAP SD-WAN Bandwidth of External Business and nonBusiness	traffic
appcat in ('Mobile','Social.Media 'nonBusiness'when appcat in ('Net coalesce(sum(bandwidth), 0) as ba devid, vd, srccountry, dstintf, s name(app) as app_group, coalesce (`srcname`),ipstr(`srcip`),nullis coalesce(nullifna(`user`), nullis	'Video\/Audio','Game','P2P','unknown') then 'E a','Proxy','Video\/Audio','Game','P2P','unknow twork.Service') then 'Network Service' end) as andwidth from ###(select \$flex_timestamp as ti srcintf, srcintfrole, dstintfrole, appid, appo (vwlname,vwlservice) as rulename, service, coa fna(`srcmac`)) as dev_src, sum(crscore%65536) fna(`unauthuser`), ipstr(`srcip`)) as user_src e(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(c	<pre>mn') then s app_cat, mestamp, csf, cat, app_group_ elesce(nullifna as crscore, c, sum(coalesce</pre>

(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_ out, count(*) as sessions from \$log-traffic where \$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf, devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_src, dev_src order by bandwidth desc)### t where \$filter-drilldown group by app_cat order by bandwidth desc

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

select

& #039;External' as summary, appcat, app_group, sum(bandwidth) as bandwidth from ###
(select \$flex_timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf,
srcintfrole, dstintfrole, appid, appcat, app_group_name(app) as app_group, coalesce
(vwlname,vwlservice) as rulename, service, coalesce(nullifna(`srcname`),ipstr
(`srcip`),nullifna(`srcmac`)) as dev_src, sum(crscore%65536) as crscore, coalesce(nullifna
(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentdelta,
sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta,
rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*)
as sessions from \$log-traffic where \$filter and vwlid IS NOT NULL and (logflag&(1|32)>0)
group by timestamp, srccountry, dstintf, csf, devid, vd, srcintf, srcintfrole, dstintfrole,
appid, appcat, app_group, rulename, service, user_src, dev_src order by bandwidth desc)### t
where \$filter-drilldown and bandwidth>0 group by appcat, app_group order by bandwidth desc

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

select

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

###(select \$flex_timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group_name(app) as app_group, coalesce (vwlname,vwlservice) as rulename, service, coalesce(nullifna(`srcname`),ipstr (`srcip`),nullifna(`srcmac`)) as dev_src, sum(crscore%65536) as crscore, coalesce(nullifna (`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from \$log-traffic where \$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf, devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_src, dev_src order by bandwidth desc)### t1 inner join app_mdata t2 on lower(t1.app_group)=lower(t2.name) where \$filter-drilldown and appcat not in ('Network.Service',

'Mobile','Social.Media','Proxy','Video\/Audio','Game','P2P','unknown') group by app_group order by bandwidth desc, app_group

Dataset Name	Description	Log Category
sdwan-CTAP-Cloud-IT-Apps- Bandwidth	CTAP SD-WAN Cloud IT Application Bandwidth	traffic
srcintfrole, dstintfrole, appid,	<pre>timestamp, csf, devid, vd, srccountry, dst appcat, app_group_name(app) as app_group, , service, coalesce(nullifna(`srcname`), ip</pre>	coalesce

(`srcip`),nullifna(`srcmac`)) as dev_src, sum(crscore%65536) as crscore, coalesce(nullifna (`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from \$log-traffic where \$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf, devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_src, dev_src order by bandwidth desc)### t where \$filter-drilldown and appcat='Cloud.IT' and bandwidth>0 group by app_group order by bandwidth desc

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

select

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

###(select \$flex_timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group_name(app) as app_group, coalesce (vwlname,vwlservice) as rulename, service, coalesce(nullifna(`srcname`),ipstr (`srcip`),nullifna(`srcmac`)) as dev_src, sum(crscore%65536) as crscore, coalesce(nullifna (`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from \$log-traffic where \$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf, devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_src, dev_src order by bandwidth desc)### t where \$filter-drilldown and appcat='Storage.Backup' and bandwidth>0 group by app_group order by bandwidth desc

CTAP SD-WAN Collaboration Application Bandwidth

Dataset Name

```
Description
```

Log Category

sdwan-CTAP-Collaboration-Apps-Bandwidth

select

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

###(select \$flex_timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group_name(app) as app_group, coalesce (vwlname,vwlservice) as rulename, service, coalesce(nullifna(`srcname`),ipstr (`srcip`),nullifna(`srcmac`)) as dev_src, sum(crscore%65536) as crscore, coalesce(nullifna (`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from \$log-traffic where \$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf, devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_src, dev_src order by bandwidth desc)### t where \$filter-drilldown and appcat='Collaboration' and bandwidth>0 group by app_group order by bandwidth desc

Dataset Name

```
Description
```

Log Category

sdwan-CTAP-Top-Streaming-App-By- CTAP SD-WAN Top Streaming Application by Bandwidth traffic Bandwidth

select

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

trom

###(select \$flex_timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group_name(app) as app_group, coalesce (vwlname,vwlservice) as rulename, service, coalesce(nullifna(`srcname`),ipstr (`srcip`),nullifna(`srcmac`)) as dev_src, sum(crscore%65536) as crscore, coalesce(nullifna (`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from \$log-traffic where \$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf, devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_src, dev_src order by bandwidth desc)### t where \$filter-drilldown and appcat='Video\/Audio' and bandwidth>0 group by app_group order by bandwidth desc

Dataset Name	Description	Log Category
sdwan-CTAP-Top-SocialMedia-App- By-Bandwidth	CTAP SD-WAN Top SocialMedia Application by Bandwidth	traffic

select

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

from

###(select \$flex_timestamp as timestamp, csf, devid, vd, srccountry, dstintf, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group_name(app) as app_group, coalesce (vwlname,vwlservice) as rulename, service, coalesce(nullifna(`srcname`),ipstr (`srcip`),nullifna(`srcmac`)) as dev_src, sum(crscore%65536) as crscore, coalesce(nullifna (`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from \$log-traffic where \$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf, devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_src, dev_src order by bandwidth desc)### t where \$filter-drilldown and appcat='Social.Media' and bandwidth>0 group by app_group order by bandwidth desc

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

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

Dataset Name	Description	Log Category
sdwan-CTAP-SB-Top-Sandbox-Files	CTAP SD-WAN Sandbox Top Sandbox Files	virus
select		

```
filename,
analyticscksum,
service,
sum(totalnum) as total_num,
(
```

case fsaverdict when & #039;malicious' then 'Malicious' when 'high risk' then 'High' when 'medium risk' then 'Medium' when 'low risk' then 'Low' else 'Other' end) as risk, (case fsaverdict when 'malicious' then 5 when 'high risk' then 4 when 'medium risk' then 3 when 'low risk' then 2 else 1 end) as risk_level from ###(select filename, analyticscksum, service, fsaverdict, dtype, coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, virus, virusid_to_str(virusid, eventtype) as virusid_s, count(*) as totalnum from \$log where \$filter group by filename, analyticscksum, service, fsaverdict, dtype, user_src, virus, virusid_s /*SkipSTART*/order by totalnum desc/*SkipEND*/)### t where \$filter-drilldown and filename is not null and dtype='fortisandbox' and fsaverdict not in ('clean', 'submission failed') group by filename, analyticscksum, risk_level, risk, service order by risk_level desc, total num desc, service, filename

Dataset Name	Description	Log Category
sdwan-CTAP-SB-Total-Number-of- Malicious-Suspicious-Files	CTAP SD-WAN Sandbox Malicious Suspicious Files Number	virus

select

case fsaverdict when & #039;malicious' then 'Malicious' when 'high risk' then 'High'
when 'medium risk' then 'Medium' when 'low risk' then 'Low' else 'Other' end) as risk, sum
(totalnum) as total_num from ###(select filename, analyticscksum, service, fsaverdict,
dtype, coalesce(nullifna(`user`), ipstr(`srcip`)) as user_src, virus, virusid_to_str
(virusid, eventtype) as virusid_s, count(*) as totalnum from \$log where \$filter group by
filename, analyticscksum, service, fsaverdict, dtype, user_src, virus, virusid_s
/*SkipSTART*/order by totalnum desc/*SkipEND*/)### t where \$filter-drilldown and
dtype='fortisandbox' and fsaverdict not in ('clean','submission failed') group by risk order
by total_num desc

Dataset Name	Description	Log Category
sdwan-CTAP-Top-Source-Countries	CTAP SD-WAN Top Source Countries	traffic
select srccountry, sum(bandwidth) as bandwidth from		
	timestamp, csf, devid, vd, srccountry, , appcat, app_group_name(app) as app_gr	

(vwlname, vwlservice) as rulename, service, coalesce(nullifna(`srcname`),ipstr (`srcip`),nullifna(`srcmac`)) as dev_src, sum(crscore%65536) as crscore, coalesce(nullifna (`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from \$log-traffic where \$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf, devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_src, dev_src order by bandwidth desc)### t where \$filter-drilldown and nullifna(srccountry) is not null and srccountry <> 'Reserved' and bandwidth>0 group by srccountry order by bandwidth desc, srccountry

Dataset Name	Description	Log Category
sdwan-CTAP-Average-Bandwidth- Day-Hour	CTAP SD-WAN Average Bandwidth by Day of Week and Hour	traffic
<pre>select hourstamp, daystamp, round(sum(bandwidth) / count(*)) as bandwidth from (select</pre>		
<pre>\$hour_of_day(timestamp) as \$HOUR_OF_DAY(timestamp) as \$day_of_week(timestamp) as sum(bandwidth) as bandwidt from</pre>	hour_stamp, daystamp,	
<pre>###(select \$flex_timestamp srcintfrole, dstintfrole, appid, (vwlname,vwlservice) as rulename</pre>	as timestamp, csf, devid, vd, srccountry, dst appcat, app_group_name(app) as app_group, coa , service, coalesce(nullifna(`srcname`),ipstr dev_src, sum(crscore%65536) as crscore, coal	lesce

(`srcip`),nullifna(`srcmac`)) as dev_src, sum(crscore%65536) as crscore, coalesce(nullifna (`user`), nullifna(`unauthuser`), ipstr(`srcip`)) as user_src, sum(coalesce(sentdelta, sentbyte, 0)+coalesce(rcvddelta, rcvdbyte, 0)) as bandwidth, sum(coalesce(rcvddelta, rcvdbyte, 0)) as traffic_in, sum(coalesce(sentdelta, sentbyte, 0)) as traffic_out, count(*) as sessions from \$log-traffic where \$filter and vwlid IS NOT NULL and (logflag&(1|32)>0) group by timestamp, srccountry, dstintf, csf, devid, vd, srcintf, srcintfrole, dstintfrole, appid, appcat, app_group, rulename, service, user_src, dev_src order by bandwidth desc)### t where \$filter-drilldown group by hourstamp, hour_stamp, daystamp) t group by hourstamp, daystamp order by hourstamp

Dataset Name	Description	Log Category
sdwan-CTAP-Average-Log-Rate-By- Hour	CTAP SD-WAN Average Log Rate by Hour	event
select		
<pre>\$hour_of_day(timestamp) as how</pre>	irstamp,	
cast(
(
sum(
total_trate + total_erat	te + total_orate	
)		
)/ sum(count)/ 100.0 as decimal(10, 2)		
) as log_rate		
from		
###(select \$flex_timestamp as	timestamp, devid, slot, sum(coalesce(tra	te, 0)) as total_

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

Dataset Name	Description	Log Category
sdwan-CTAP-CPU-Usage-Per-Hour	Event usage CPU	event
select		
<pre>\$hour_of_day(timestamp) as hou cast(</pre>	rstamp,	
<pre>sum(total_cpu) / sum(count) a</pre>	s decimal(6, 2)	
) as cpu_avg_usage from		
	timestamp, devid, slot, sum(coal	esce(trate, 0)) as total
trate, sum(coalesce(erate, 0)) a		—
(itime) as first_seen, max(itime	—	—
(coalesce(mem, 0)) as mem_peak,		-
as disk_peak, sum(coalesce(cpu,	—	—
(coalesce(trate, 0)+coalesce(era		—
(totalsession, 0)) as totalsessi		
(coalesce(split_part(bandwidth,		
part(bandwidth, '/', 2), '0') as	-	
'/', 1), '0') as integer)+cast(c	<pre>palesce(split_part(bandwidth, '/</pre>	', 2), '0') as integer)) as
<pre>transmit_peak, sum(coalesce(setu;</pre>	prate, 0)) as cps, max(coalesce(setuprate, 0)) as cps_peak,

transmit_peak, sum(coalesce(setuprate, 0)) as cps, max(coalesce(setuprate, 0)) as cps_peak, count(*) as count from \$log where \$filter and subtype='system' and action='perf-stats' group by timestamp, devid, slot order by total_mem desc)### t group by hourstamp order by hourstamp

Dataset Name	Description	Log Category
sdwan-CTAP-Memory-Usage-Per- Hour	Event usage memory	event
<pre>crate, sum(coalesce(erate, 0)) as (itime) as first_seen, max(itime (coalesce(mem, 0)) as mem_peak, so as disk_peak, sum(coalesce(cpu, 1)) (coalesce(trate, 0)+coalesce(eras (totalsession, 0)) as totalsession (coalesce(split_part(bandwidth, coart(bandwidth, '/', 2), '0') as /', 1), '0') as integer)+cast(coalesce(coalesce(coalesce))</pre>		e, 0)) as total_orate, min , 0)) as total_mem, max disk, max(coalesce(disk, 0)) pu, 0)) as cpu_peak, max grate_peak, sum(coalesce)) as session_peak, sum(cast nt, sum(cast(coalesce(split_ lesce(split_part(bandwidth, ', 2), '0') as integer)) as

Dataset Name	Description	Log Category
Top-Destination-Addresses-By- Bandwidth-Bar	Top destinations by bandwidth usage	traffic
<pre>select coalesce(nullifna(root_domain(hostname)), ipstr(dstip)) as domain, sum(coalesce(sentbyte, 0)+ co) as bandwidth, sum(coalesce(rcvdbyte, 0)+ co) as traffic_in, sum(coalesce(rcvdbyte, 0)) as traffic_in, sum(coalesce(sentbyte, 0)) as traffic_out from \$log where \$filter and (logflag&1>0) and coalesce(nullifna(</pre>	alesce(rcvdbyte, 0)	

```
root_domain(hostname)
),
ipstr(`dstip`)
) is not null
group by
domain
having
sum(
    coalesce(sentbyte, 0)+ coalesce(rcvdbyte, 0)
)> 0
order by
bandwidth desc
```

Dataset Name	Description	Log Category
intf-Timeline-Sampling	Interface Utilization Timeline by Data Sampling	event
<pre>with base_qry as (select tm, rcvdbps, ntile(100) over (order by rcvdbps) as percentile from (select (timestamp / 300 * 300) sum(rcvdbps) as rcvdbps, 300 as interval from intfstats_billing tb1 join (select ti.dvid, intfname from intfinfo ti</pre>	as tm,	event
where \$dev_filter) tb2 on tb1.dvid = tb2.	.dvid	
<pre>and tb1.intfname = tb2.i where \$cust_time_filter(timest group by tm) tmp</pre>		
<pre>), ref_qry as (select cast(max(rcvdbps)/ 1000000 as c) as ref_val from</pre>	decimal(18, 2)	

```
base_qry
 where
   percentile = 95
)
select
 from_itime(timestamp) as tmstamp,
 cast(
  rcvdbps / 1000000 as decimal(18, 2)
 ) as rcvdbps,
 ref_val
from
 ref_qry,
  (
   select
     tm as timestamp,
     rcvdbps,
     rank() over(
       partition by (tm / 3600)
       order by
         tm
     ) as r
   from
     base_qry
 ) t
where
 r = 1
order by
 tmstamp
```

Dataset Name	Description	Log Category
intf-Util-Histogram	Interface Utilization Value Distribution	event
elect		
cast(
(
(
<pre>max(max_value) over ()</pre>		
)* seq / 100		
) as decimal(16, 0)		
) as value,		
cnt		
rom		
(
select		
<pre>generate_series(0, 100, 2)</pre>	as seq	
) t1		
left join (
select		
perc,		
max_value,		
count(*) as cnt		
from		
(
select		
WIDTH_BUCKET (

```
rcvdbps,
            Ο,
            (
             max(rcvdbps) over ()
            ) + 1,
            50
          )* 2 as perc,
         max(rcvdbps) over () as max value
        from
          (
            select
              (timestamp / 300 * 300) as tm,
              sum(rcvdbps) as rcvdbps,
              300 as interval
            from
              intfstats billing tb1
              join (
                select
                 ti.dvid,
                  intfname
                from
                  intfinfo ti
                  left join devtable td on ti.dvid = td.dvid
                where
                  $dev_filter
              ) tb2 on tb1.dvid = tb2.dvid
              and tb1.intfname = tb2.intfname
            where
              $cust_time_filter(timestamp)
            group by
              tm
          ) tmp
     ) t bucket
   group by
     perc,
     max_value
 ) t2 on t1.seq = t2.perc
order by
 seq
```

Dataset Name

intf-Sorted-Line Interface Utilization Line Sorted by bps event with base_gry as (select	
with base_qry as (select	
<pre>rcvdbps, ntile(100) over (order by rcvdbps) as percentile from (select (timestamp / 300 * 300) as tm, sum(rcvdbps) as rcvdbps,</pre>	

Description

Log Cotogony

```
300 as interval
      from
        intfstats_billing tb1
        join (
         select
           ti.dvid,
            intfname
          from
            intfinfo ti
            left join devtable td on ti.dvid = td.dvid
          where
            $dev filter
        ) tb2 on tb1.dvid = tb2.dvid
        and tb1.intfname = tb2.intfname
     where
        $cust_time_filter(timestamp)
     group by
        tm
   ) tmp
),
ref_qry as (
 select
   cast(
     max(rcvdbps) / 1000000 as decimal(18, 2)
   ) as ref_val
 from
   base qry
 where
   percentile = 95
)
select
 n_perc,
 cast(
   rcvdbps / 1000000 as decimal(18, 2)
 ) as rcvdbps,
 ref_val
from
  (
   select
     seq as n_perc,
     rcvdbps
   from
      (
       select
          generate_series(0, 100, 1) as seq
     ) t1
     left join (
        select
         max(rcvdbps) as rcvdbps,
          percentile
        from
         base_qry
        group by
         percentile
     ) t2 on t1.seq = t2.percentile
 ) t,
```

ref_qry order by n_perc **Dataset Name** Description Log Category event intf-Data-Analysis-Table Interface Utilization Data Analysis with base_qry as (select rcvdbps, interval, ntile(100) over (order by rcvdbps) as percentile from (select (timestamp / 300 * 300) as tm, sum(rcvdbps) as rcvdbps, 300 as interval from intfstats billing tb1 join (select ti.dvid, intfname from intfinfo ti left join devtable td on ti.dvid = td.dvid where \$dev filter) tb2 on tb1.dvid = tb2.dvid and tb1.intfname = tb2.intfname where \$cust_time_filter(timestamp) group by tm) tmp) select min mbps, low ref mbps, mean mbps, ref mbps, peak mbps, actual gb, total from (select cast(min(rcvdbps) / 1000000 as decimal(18, 2)) as min mbps, cast(

```
avg(rcvdbps) / 1000000 as decimal(18, 2)
   ) as mean_mbps,
   cast(
     max(rcvdbps) / 1000000 as decimal(18, 2)
   ) as peak_mbps,
   cast(
     (
        select
         max(rcvdbps)
       from
         base qry
       where
         percentile = 5
     )/ 1000000 as decimal(18, 2)
    ) as low ref mbps,
   cast(
      (
        select
         max(rcvdbps)
        from
          base_qry
        where
         percentile = 95
     )/ 1000000 as decimal(18, 2)
    ) as ref_mbps,
   cast(
     sum(interval * rcvdbps) / 8 /(1024 * 1024 * 1024) as decimal(18, 2)
   ) as actual gb,
   count(*) as total
 from
   base_qry
) t
```

Dataset Name	Description	Log Category
intf-Device-Summary	Interface Utilization Device Summary	event
select devname, t1.intfname,		
rcvd_gb		
from (
select devname, ti.dvid, intfname		
from devtable td join intfinfo ti on ti.	dvid = td dvid	
where \$dev_filter		
) t1 join (
select dvid,		

```
intfname,
     cast(
       sum(interval * rcvdbps)/ 8 /(1024 * 1024 * 1024) as decimal(18, 2)
     ) as rcvd gb
   from
     intfstats_billing
   where
     $cust_time_filter(timestamp)
   group by
     dvid,
     intfname
 ) t2 on t1.dvid = t2.dvid
 and t1.intfname = t2.intfname
order by
 devname,
 rcvd gb desc,
 t1.intfname
```

Dataset Name	Description	Log Category
daily-Summary-Traffic-Bandwidth-Line	Daily Summary - Traffic Bandwidth Line	traffic
<pre>select \$fv_line_timescale(timescale) sum(traffic_in) as traffic_in, sum(traffic_out) as traffic_ou from (union all) t group by time order by time</pre>		
Dataset Name	Description	Log Category
daily-Summary-Top-User	Daily Summary - Top User by Bandwidth	traffic
<pre>select coalesce(nullifna(f_user), ipstr(srcip), & #039;Unknown') as f_user, srcip order by bandwidth dose</pre>	srcip, sum(bandwidth) as bandwidth FROM	1 t group by f_user,

```
srcip order by bandwidth desc
```

Dataset Name	Description	Log Category
daily-Summary-Top-Domain	Daily Summary - Top Domain by Bandwidth	traffic
<pre>select domain, sum(bandwidth) as bandwidth from t where</pre>		

```
domain is not null
group by
 domain
order by
 bandwidth desc
```

Dataset Name	Description	Log Category
daily-Summary-Top-Appcat-Bandwidth	Daily Summary - Top Application Category by Bandwidth	traffic
<pre>select appcat, sum(bandwidth) as bandwidth from (select t1.*, t2.app cat as appcat</pre>		
from t1 left join app_mdata t2 on	t1.app_group = t2.name	
) t where \$filter - drilldown and appcat is not null		
group by appcat order by bandwidth desc		

Dataset Name

Dataset Name	Description	Log Category
daily-Summary-Top-App	Daily Summary - Top Application	traffic
when 4 then 'High' when 5 then bandwidth, sum(traffic_in) as t block) as session_block, (sum(s as sessions from (select t1.*, cat is null then 'Unknown' else else t2.risk::int end) as d_ris	<pre>n & #039;Low' when 2 then 'Elevated' wh 'Critical' else NULL end) as risk, sum raffic_in, sum(traffic_out) as traffic_ essions)-sum(session_block)) as session (case when (d_flags & 1) = 1 then 'Not. t2.app_cat end) as appcat, (case when k from t1 left join app_mdata t2 on t1 y app_group order by max(d_risk) desc,</pre>	(bandwidth) as out, sum(session_ n_pass, sum(sessions) Scanned' when t2.app_ t2.risk is null then 0 Lapp_group=t2.name) t

Dataset Name	Description	Log Category
daily-Summary-Top-Threats	Daily Summary - Top Threats	traffic
<pre>select threat_s as threat, threattype_s as threattype,</pre>		

```
sum(threatweight) as threatweight,
```

```
sum(threat_block) as threat_block,
  (
    sum(threatweight) - sum(threat block)
  ) as threat pass,
  sum(incidents) as incidents,
  sum(incident block) as incident block,
  (
    sum(incidents) - sum(incident block)
  ) as incident pass
from
    union all
     ) t
group by
  threat,
  threattype
order by
  threatweight desc
```

Dataset Name Description Log Category traffic daily-Summary-Top-Compromised-Daily Summary - Top Compromised Hosts Hosts select epid, devid, vd, srcip, devtype, fctuid, euid, bmp logtype as logtype, unauthuser, srcmac, osname, osversion, f user, (case when epid<1024 then ipstr(srcip) else epname end) as epname, threat num, bl count, cs score, cs count, verdict, ip reversed, rescan, (case verdict when 1 then & #039; Low Suspicion' when 2 then 'Medium Suspicion' when 3

case verdict when 1 then & #039;Low Suspicion' when 2 then 'Medium Suspicion' when 3 then 'High Suspicion' when 4 then 'Infected' else 'N/A' end) as verdict_s,ack_time, ack_ note, last_bl as last_detected_time from (SELECT epid, itime, bl_count, cs_score, cs_count, threat_num, bmp_logtype, last_bl, verdict, ip_reversed, rescan, srcip, epname, srcmac, osname, osversion, devtype, fctuid, euid, unauthuser, f_user, ack_note, ack_time, devid, vd, csf, devname FROM (SELECT tvdt.epid, itime, tvdt.bl_count, tvdt.cs_score, tvdt.cs_count, tvdt.threat_num, tvdt.bmp_logtype, tvdt.last_bl, tvdt.verdict, tvdt.ip_reversed,

tvdt.rescan, (CASE WHEN tvdt.epid>1024 THEN tep.epip ELSE tvdt.srcip END) as srcip, tep.epname, tep.mac as srcmac, tep.osname, tep.osversion, tep.epdevtype as devtype, teu.fctuid, teu.euid, teu.unauthuser, (case when teu.euid<=1024 then ipstr(tvdt.srcip) else teu.euname end) as f user, tack.ack note, (case when (tvdt.ack time max=0 or tvdt.ack time min=0) then NULL else tvdt.ack time max end) as ack time,tdev.devid, tdev.vd, tdev.csf, tdev.devname FROM (SELECT epid, srcip, min(day st) as itime, array length(intarr agg (threatid), 1) as threat_num, intarr agg(dvid) as dvid, sum(bl count) as bl count, max(cs score) as cs score, sum(cs count) as cs count, max(last bl) as last bl, max(ack time) as ack time max, min(ack time) as ack time min, bit or(bmp logtype) as bmp logtype, max (verdict) as verdict, max(ip reversed) as ip reversed, max(rescan) as rescan FROM ((SELECT epid, srcip, day st, ack time, threatid, dvid, bl count, cs score, cs count, last bl, bmp logtype, verdict, (case when ioc flags&2>0 then 1 else 0 end) as ip reversed, (case when ioc flags&1>0 then 1 else 0 end) as rescan FROM \$ADOMTBL PLHD IOC VERDICT /*verdict table*/WHERE day_st>=\$start_time and day_st<=\$end_time /*time filter*/) UNION ALL (SELECT epid, srcip, day st, ack time, threatid, dvid,bl count, cs score, cs count, last bl, bmp logtype, verdict, (case when ioc flags&2>0 then 1 else 0 end) as ip reversed, (case when ioc_flags&1>0 then 1 else 0 end) as rescan FROM \$ADOMTBL PLHD INTERIM IOC VERDICT /*verdict intrim table*/WHERE day st>=\$start time and day st<=\$end time /*time filter*/ and verdict>0)) tvdt int GROUP BY epid, srcip) tvdt INNER JOIN /*end points*/ \$ADOM ENDPOINT as tep ON tvdt.epid=tep.epid LEFT JOIN /*end user*/ (select epid, euname, fctuid, euid, unauthuser from (select epid, eu.euid, euname, fctuid, euname as unauthuser, row number() over (partition by epid order by ((case when fctuid is null then 0 else 1 end), lastactive) desc) nth from \$ADOM ENDUSER eu /*end user*/, \$ADOM EPEU DEVMAP as map /*epeu dev map*/ where eu.euid=map.euid and eu.euid>1024) eum where nth=1) teu on tvdt.epid=teu.epid LEFT JOIN /*ack table*/(SELECT epid, srcip, ack time, ack note FROM (SELECT epid, srcip, ack_ time, ack note, row number() over (PARTITION BY epid, srcip order by ack time desc) as ackrank FROM ioc ack WHERE adomoid=\$adom oid) rankqry WHERE ackrank=1) tack ON tvdt.epid=tack.epid and ((tvdt.srcip is null and tack.srcip is null) or tvdt.srcip=tack.srcip) LEFT JOIN devtable tdev ON tdev.dvid = tvdt.dvid[1] WHERE tvdt.dvid && (SELECT array agg(dvid) from devtable WHERE \$filter-drilldown)) tioc) t order by threat num desc

Dataset Name	Description	Log Category
daily-Summary-Incidents-by-Severity	Incidents by Severity	
<pre>select severity, count(*) as incnum from \$incident where \$filter - drilldown and \$cust_time_filter(createt group by severity order by incnum desc</pre>	ime)	
Dataset Name	Description	Log Category
ueba-Asset-Count-by-Detecttype	Asset Count by Detection Type	
select		

case detecttype when & #039; by ip' then 'IP' when 'by mac' then 'MAC' end) as

detecttype, count(distinct epid) as count from \$ADOM_ENDPOINT t1 where epid>1024 and \$filter-drilldown and lastseen>=\$start_time and firstseen<\$end_time and detecttype in ('by_ ip', 'by_mac') group by detecttype order by count desc

Dataset Name	Description Log Category
ueba-Asset-Identification	Asset Count by Identification
<pre>with qualified_ep as (select t2.epid, t2.euid from \$ADOM_ENDPOINT t1 inner join \$ADOM_EPEU_DEVMAP where \$filter - drilldown and lastseen >= \$start_time and firstseen<\$end_time and t2.epid>1024</pre>	t2 on t1.epid = t2.epid
(distinct epid) as count from id	on tl.euid = t2.euid and euname is not null) (select 'Identified' as type, count entified_ep) union all (select 'Unidentified' as type, cour alified ep where epid not in (select * from identified ep))

Dataset Name

Description

Asset Count by Hardware OS

ueba-Asset-Count-by-HWOS

```
select
 osname,
 count(distinct t2.epid) as count
from
 $ADOM ENDPOINT t1
 inner join $ADOM_EPEU_DEVMAP t2 on t1.epid = t2.epid
where
 $filter - drilldown
 and lastseen >= $start time
 and firstseen<$end time
 and osname is not null
 and t2.epid>1024
group by
 osname
order by
 count desc
```

Log Category

Dataset Name	Description	Log Category
ueba-Asset-Count-by-Device-and- Detecttype	Asset Count by Source and Detection Type	
detecttype, count(distinct t1.e DEVMAP t2 on t1.epid=t2.epid in and \$filter-drilldown and t1.le	9;by_ip' then 'IP' when 'by_mac' then 'MAC' epid) as count from \$ADOM_ENDPOINT t1 inner nner join devtable t3 on t2.devid=t3.devid w astseen>=\$start_time and firstseen<\$end_time y_ip', 'by_mac') group by devname, detecttyp	join \$ADOM_EPEU_ where t1.epid>1024 e and devname is
Dataset Name	Description	Log Category

		_	
ueba-User-Count-by-Usergroup	User Count by User Group		

select

coalesce(

eugroup,

& #039;Unknown') as eugroup, count(distinct t1.euid) as count from \$ADOM_ENDUSER t1 inner join \$ADOM_EPEU_DEVMAP t2 ON t1.euid=t2.euid where \$filter-drilldown and t1.euid>1024 and t1.lastseen>=\$start_time and firstseen<\$end_time group by eugroup order by count desc

```
Dataset Name
                                  Description
                                                                                 Log Category
ueba-Asset-User-Count-by-Device
                                 Asset and User Count by Device
select
 devname,
 cnt for,
 sum(count) as count
from
  (
      select
        devname,
        & #039; Endpoint' as cnt for, count(distinct t2.epid) as count from $ADOM ENDPOINT t1
inner join $ADOM EPEU DEVMAP t2 on t1.epid=t2.epid inner join devtable t3 on
t2.devid=t3.devid where $filter-drilldown and t1.lastseen>=$start time and
t1.firstseen<$end time and t2.epid>1024 group by devname order by count desc) union all
(select devname, 'User' as cnt for, count(distinct t1.euid) as count from $ADOM ENDUSER t1
inner join $ADOM EPEU DEVMAP t2 ON t1.euid=t2.euid inner join devtable t3 on
t2.devid=t3.devid where $filter-drilldown and t1.lastseen>=$start time and
tl.firstseen<$end_time and euname != '(none)' and epid>1024 and tl.euid>1024 group by
devname order by count desc)) t group by devname, cnt for order by count desc
```

Dataset Name	Description	Log Category
ueba-Asset-User-Count-by-Device- Interface-and-Detectiontype	Asset and User Count by Source Device Interface and Detection Method	

```
select
  devname,
  srcintf,
  sum(mac_cnt) as mac_cnt,
  sum(ip_cnt) as ip_cnt,
  sum(ep_count) as ep_count,
  sum(eu_count) as eu_count
from
  (
     (
        select
        devname,
```

```
srcintf,
sum(
```

case when detecttype =& #039; by mac' then count else 0 end) as mac cnt, sum(case when detecttype='by ip' then count else 0 end) as ip cnt, sum(count) as ep count, 0 as eu count from (select devname, srcintf, detecttype, count(distinct t1.epid) as count from \$ADOM ENDPOINT t1 inner join \$ADOM EPEU DEVMAP t2 on t1.epid=t2.epid inner join devtable t3 on t2.devid=t3.devid where t1.epid>1024 and \$filter-drilldown and t1.lastseen>=\$start time and firstseen<\$end time and devname is not null and srcintf is not null and detecttype in ('by_ip', 'by_mac') group by devname, srcintf, detecttype order by count desc) t1 group by devname, srcintf order by ep count desc) union all (SELECT devname, srcintf, 0 as mac cnt, 0 as ip cnt, 0 as ep count, count(DISTINCT euid) as eu count from (select euid, euname, t3.epid, eugroup, srcintf, devname, devid from (select t1.euid, euname, epid, eugroup, srcintf, devname, t2.devid from \$ADOM ENDUSER t1 inner join \$ADOM EPEU DEVMAP t2 ON t1.euid=t2.euid inner join devtable t3 on t2.devid=t3.devid where t1.lastseen>=\$start time and t1.firstseen<\$end time and srcintf is not null) t3 LEFT JOIN \$ADOM ENDPOINT t4 ON t3.epid = t4.epid) t5 where euname != '(none)' and epid>1024 and euid>1024 and \$filterdrilldown group by devname, srcintf order by eu count desc)) t group by devname, srcintf order by devname, sum(eu count) + sum(ep count) desc

Dataset Name	Description	Log Category
ueba-Asset-User-Discovery-by-Time	Asset and User Count by Discovery Time	
<pre>select \$flex_timescale(firstseen) as count(distinct epid) as ep_cou count(distinct euid) as eu_cou from ((select firstseen, t1.epid, null as euid from \$ADOM_ENDPOINT t1</pre>	time, nt, nt VMAP t2 on t1.epid = t2.epid rt_time	
and t1.epid>1024		
union all		

```
(
        select
         firstseen,
         null as epid,
          t1.euid
        from
          $ADOM ENDUSER t1
          inner join $ADOM_EPEU_DEVMAP t2 ON t1.euid = t2.euid
       where
          t1.euid>1024
          and $filter - drilldown
          and firstseen >= $start_time
         and firstseen<$end_time
     )
 ) t
group by
 time
order by
 time
```

Dataset Name	Description	Log Category
dns-Security-Domain-Count-by- Threat-Level	Domain Count by Threat level	dns
<pre>botnet or catdesc in ('Malicia catdesc in ('Newly Observed Do or catdesc LIKE '%Dynamic DNS' num from ###(select dvid, qna user, dstip, srcip, catdesc, i null) as is_botnet, min(nanose (eventtime)) as last_seen, cou qname, f_user, dstip, srcip, o</pre>	ain', 'infected-ip', 'infected-url') th ous Websites', 'Phishing', 'Spam URLs') omain', 'Newly Registered Domain', 'Pro %' then 'medium' end) as threat_level, ame, coalesce(nullifna(`user`), nullifn level, tdtype, (botnetdomain is not nu ec_to_sec(eventtime)) as first_seen, ma unt(*) as total_num from \$log-dns where catdesc, level, tdtype, is_botnet order y total_num desc) t where threat_level	then 'high' when bxy Avoidance','Unrated') sum(total_num) as total_ ha(`unauthuser`)) as f_ hall or botnetip is not hax(nanosec_to_sec e \$filter group by dvid, to by total_num desc)### t

Dataset Name	Description	Log Category
dns-Top-Queried-Domain-Bar	Top Queried Domain	dns
<pre>select qname, count(*) as total_num from \$log where \$filter</pre>		

```
and qname is not null
group by
qname
order by
total_num desc
```

Dataset Name	Description	Log Category
dns-Security-Top-Visited-Domain- Categories	Top Visited Domain Categories	dns
<pre>dstip, srcip, catdesc, level, t is_botnet, min(nanosec_to_sec(e last_seen, count(*) as total_nu</pre>	<pre>sce(nullifna(`user`), nullifna(`unauthu dtype, (botnetdomain is not null or bo venttime)) as first_seen, max(nanosec_t m from \$log-dns where \$filter group by dtype, is_botnet order by total_num des der by total_num desc</pre>	tnetip is not null) as o_sec(eventtime)) as dvid, qname, f_user,
Dataset Name	Description	Log Category
dns-Security-Top-Visited-High-Risk- Domain-Categories	Top Visited High Risk Domain Categories	dns
<pre>dstip, srcip, catdesc, level, t is_botnet, min(nanosec_to_sec(e last_seen, count(*) as total_nu dstip, srcip, catdesc, level, t</pre>	<pre>sce(nullifna(`user`), nullifna(`unauthu dtype, (botnetdomain is not null or bo venttime)) as first_seen, max(nanosec_t m from \$log-dns where \$filter group by dtype, is_botnet order by total_num des not null group by catdesc order by tot</pre>	<pre>tnetip is not null) as .o_sec(eventtime)) as dvid, qname, f_user, .c)### t where</pre>
Dataset Name	Description	Log Category
dns-Security-Top-Domain-with-Botnet- CC-Detected	Top Domain with Botnet C&C Detected	dns
<pre>select qname, sum(total_num) as total_num</pre>		

```
from
```

###(select coalesce(botnetdomain, ipstr(botnetip)) as domain, qname, 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, qname, action, srcip, sevid order by sevid desc)### t where qname is not null group by qname order by total_num desc

Dataset Name	Description	Log Category
dns-Security-FortiGate-with-Top-	FortiGate with Top Domain Visited by Source IP	dns
Domain-Visited-by_Source-IP		
select		
devname,		
srcip,		
qname,		
category,		
total_num		
from		
(
select		
devname,		
srcip,		
qname,		
category,		
total_num, row number() over (
partition by devname,		
srcip,		
gname		
order by		
total_num desc,		
qname		
) as rank		
from		
(
select		
devname,		
srcip,		
qname,		
max(catdesc) as categ	ory,	
sum(total_num) as tot	al_num	
from		
user, dstip, srcip, catdesc, le null) as is_botnet, min(nanosec (eventtime)) as last_seen, coun qname, f_user, dstip, srcip, ca	e, coalesce(nullifna(`user`), nullifna(`unau vel, tdtype, (botnetdomain is not null or k _to_sec(eventtime)) as first_seen, max(nanos t(*) as total_num from \$log-dns where \$filte tdesc, level, tdtype, is_botnet order by tot .dvid=t2.dvid where qname is not null and sa	ootnetip is not sec_to_sec er group by dvid, tal_num desc)###
group by devname, srcip, qname srcip, qname	order by total_num desc) t) t where rank=1 o	order by devname,

Dataset Name	Description	Log Category
dns-Security-Top-Domain-Lookup- Failure-by-Count	Top Domain Lookup Failures by Count	dns
<pre>select qname, count(*) as total_num from \$log - dns where</pre>		

```
$filter
and qname is not null
and (
    action =& #039;block' or logid_to_int(logid)=54200) group by qname order by total_num
desc
```

Dataset Name	Description	Log Category
dns-Security-Top-Source-IP-by- Destination-Count	Top Source IP by Destination Count	dns
<pre>dstip, srcip, catdesc, level, t is_botnet, min(nanosec_to_sec(e last_seen, count(*) as total_nu dstip, srcip, catdesc, level, t</pre>	al_num esce(nullifna(`user`), nullifna(`unauthu tdtype, (botnetdomain is not null or bo eventtime)) as first_seen, max(nanosec_t um from \$log-dns where \$filter group by tdtype, is_botnet order by total_num des group by srcip order by total_num desc	<pre>tnetip is not null) as o_sec(eventtime)) as dvid, qname, f_user,</pre>
Dataset Name	Description	Log Category
dns-Security-Top-Destination-IP-by- Source-Count	Top Destination IP by Source Count	dns
<pre>dstip, srcip, catdesc, level, t is_botnet, min(nanosec_to_sec(e last_seen, count(*) as total_nu dstip, srcip, catdesc, level, t</pre>	al_num esce(nullifna(`user`), nullifna(`unauthu tdtype, (botnetdomain is not null or bo eventtime)) as first_seen, max(nanosec_t um from \$log-dns where \$filter group by tdtype, is_botnet order by total_num des group by dstip order by total_num desc	<pre>tnetip is not null) as o_sec(eventtime)) as dvid, qname, f_user, c)### t where srcip is</pre>
Dataset Name	Description	Log Category

dns-Security-Severity-by-High-Risk-Source-IPs-Count

Severity by High Risk Source IPs Count

dns

select

(

CASE sevid WHEN 5 THEN & #039;Critical' WHEN 4 THEN 'High' WHEN 3 THEN 'Medium' WHEN '2' THEN 'Info' ELSE 'Low' END) as severity, count(distinct srcip) as total_num from (select srcip, (CASE WHEN level IN ('critical', 'alert', 'emergency') THEN 5 WHEN level='error' THEN 4 WHEN level='warning' THEN 3 WHEN level='notice' THEN 2 ELSE 1 END) as sevid, count(*) as total_num from ###(select dvid, qname, coalesce(nullifna(`user`), nullifna(`unauthuser`)) as f_user, dstip, srcip, catdesc, level, tdtype, (botnetdomain is not null or botnetip is not null) as is_botnet, min(nanosec_to_sec(eventtime)) as first_seen, max(nanosec_to_sec (eventtime)) as last_seen, count(*) as total_num from \$log-dns where \$filter group by dvid, qname, f_user, dstip, srcip, catdesc, level, tdtype, is_botnet order by total_num desc)### t where level>='warning' and srcip is not null group by srcip, sevid order by total_num desc) t group by severity having sum(total_num)>0 order by total_num desc

Dataset Name	Description	Log Category
dns-Security-Top-DNS-High-Risk- Source-IP	Top DNS High Risk Source IP	dns
<pre>select srcip, sum(case when sevid = 5 then te) as num_cri, sum(case when sevid = 4 then te) as num_hig, sum(case when sevid = 3 then te) as num_med, sum(total num) as total num</pre>	_ otal_num else 0 end	
from		
<pre>level='warning' THEN 3 WHEN level='warning' THEN 3 WHEN levenum from ###(select dvid, qname, specific distribution) as is_botnet, min(nanosed) (eventtime)) as last_seen, courdiname, f_user, dstip, srcip, converse level>='warning' and srcip)</pre>	<pre>lert', 'emergency') THEN 5 WHEN level vel='notice' THEN 2 ELSE 1 END) as se e, coalesce(nullifna(`user`), nullifn evel, tdtype, (botnetdomain is not n c_to_sec(eventtime)) as first_seen, m nt(*) as total_num from \$log-dns wher atdesc, level, tdtype, is_botnet orde ip is not null group by srcip, sevid tal_num)>0 order by total_num desc</pre>	<pre>vid, count(*) as total_ a(`unauthuser`)) as f_ ull or botnetip is not ax(nanosec_to_sec e \$filter group by dvid, or by total_num desc)###</pre>

Dataset Name	Description	Log Category
dns-Security-Top-Infected-Domain-by- Count	Top Infected Domain by Count	dns
select		

```
qname,
count(distinct srcip) as total_num
from
```

###(select dvid, qname, coalesce(nullifna(`user`), nullifna(`unauthuser`)) as f_user, dstip, srcip, catdesc, level, tdtype, (botnetdomain is not null or botnetip is not null) as is_botnet, min(nanosec_to_sec(eventtime)) as first_seen, max(nanosec_to_sec(eventtime)) as last_seen, count(*) as total_num from \$log-dns where \$filter group by dvid, qname, f_user, dstip, srcip, catdesc, level, tdtype, is_botnet order by total_num desc)### t where qname is not null and tdtype='infected-domain' group by qname order by total_num desc

Dataset Name	Description	Log Category
dns-Security-Top-Blocked-Domains- by-Reason	Top Blocked Domains by Reason	dns

```
select
  qname,
  msg,
  count(*) as total_num
from
  $log
where
  $filter
  and qname is not null
  and msg LIKE & #039;Domain was blocked%' group by qname, msg order by total_num desc
```

Dataset Name	Description	Log Category
dns-Security-Top-Users-by-Infected- Domain-Visits	Top Users by Infected Domain Visits	dns
<pre>select coalesce(f_user, instr(`crain`))</pre>		

```
ipstr(`srcip`)
) as user_src,
  count(distinct qname) as total_num
from
```

###(select dvid, qname, coalesce(nullifna(`user`), nullifna(`unauthuser`)) as f_user, dstip, srcip, catdesc, level, tdtype, (botnetdomain is not null or botnetip is not null) as is_botnet, min(nanosec_to_sec(eventtime)) as first_seen, max(nanosec_to_sec(eventtime)) as last_seen, count(*) as total_num from \$log-dns where \$filter group by dvid, qname, f_user, dstip, srcip, catdesc, level, tdtype, is_botnet order by total_num desc)### t where qname is not null and tdtype='infected-domain' and (f_user is not null or srcip is not null) group by user src order by total num desc

```
Dataset Name
                                  Description
                                                                                  Log Category
 dns-Security-Top-Users-and-Infected-
                                  Top Users and Infected Domain by Visit Count
                                                                                  dns
 Domain-by-Visit-Count
select
 coalesce(
   f user,
    ipstr(`srcip`)
  ) as user src,
  qname,
  sum(total_num) as total_num
from
  ####(select dvid, qname, coalesce(nullifna(`user`), nullifna(`unauthuser`)) as f user,
dstip, srcip, catdesc, level, tdtype, (botnetdomain is not null or botnetip is not null) as
is botnet, min(nanosec to sec(eventtime)) as first seen, max(nanosec to sec(eventtime)) as
last seen, count(*) as total num from $log-dns where $filter group by dvid, qname, f user,
dstip, srcip, catdesc, level, tdtype, is botnet order by total num desc)### t where qname is
not null and (f user is not null or srcip is not null) and tdtype='infected-domain' group by
user src, qname order by total num desc
```

Dataset Name	Description	Log Category
dns-Security-Top-Users-by-Visited- Domain-Category-Count	Top Users by Visited Domain Category Count	dns

```
select
  coalesce(
    f_user,
    ipstr(`srcip`)
  ) as user_src,
    count(distinct catdesc) as total_num
from
```

###(select dvid, qname, coalesce(nullifna(`user`), nullifna(`unauthuser`)) as f_user, dstip, srcip, catdesc, level, tdtype, (botnetdomain is not null or botnetip is not null) as is_botnet, min(nanosec_to_sec(eventtime)) as first_seen, max(nanosec_to_sec(eventtime)) as last_seen, count(*) as total_num from \$log-dns where \$filter group by dvid, qname, f_user, dstip, srcip, catdesc, level, tdtype, is_botnet order by total_num desc)### t where catdesc is not null and (f_user is not null or srcip is not null) group by user_src order by total_ num desc

Dataset Name	Description	Log Category
dns-Security-Top-Users-and-Visited- Domain-Category-by-Count	Top Users and Visited Domain Category by Count	dns
<pre>select coalesce(f_user, ipstr(`srcip`)) as user_src, catdesc, srcip, sum(total num) as total num</pre>		
<pre>from ###(select dvid, qname, coales dstip, srcip, catdesc, level, td is_botnet, min(nanosec_to_sec(ev last_seen, count(*) as total_num</pre>	<pre>ce(nullifna(`user`), nullifna(`unauthuser`)) a type, (botnetdomain is not null or botnetip i enttime)) as first_seen, max(nanosec_to_sec(ev from \$log-dns where \$filter group by dvid, qn type, is botnet order by total num desc)### t</pre>	s not null) as renttime)) as ame, f_user,

dstip, srcip, catdesc, level, tdtype, is_botnet order by total_num desc)### t where catdesc is not null and (f_user is not null or srcip is not null) group by user_src, catdesc, srcip order by total_num desc

Dataset Name	Description	Log Category
dns-Security-Top-Newly-Detected- Domain-by-Count	Top Newly Detected Domain by Count	dns
<pre>dstip, srcip, catdesc, level, to is_botnet, min(nanosec_to_sec(ex))</pre>	sce(nullifna(`user`), nullifna(`unauthuser`)) dtype, (botnetdomain is not null or botnetip venttime)) as first_seen, max(nanosec_to_sec(is not null) as eventtime)) as
dstip, srcip, catdesc, level, to	<pre>m from \$log-dns where \$filter group by dvid, dtype, is_botnet order by total_num desc)### n<\$end_time and tdtype is not null and qname</pre>	t where last_

Dataset Name	Description	Log Category
dns-Security-Top-Newly-Detected- Domain-and-Source-IP-with-First- Seen-and-Last-Seen	Top Newly Detected Domain and Source IP with First Seen and Last Seen	dns
<pre>select qname, srcip, from_itime(min(first_seen)) as first_seen, from_itime(max(last_seen)) as last_seen, sum(total num) as total num</pre>		
from	<pre>sce(nullifna(`user`), nullifna(`unauthuser`))</pre>	as f user,
<pre>dstip, srcip, catdesc, level, to is_botnet, min(nanosec_to_sec(ex last_seen, count(*) as total_num</pre>	<pre>dtype, (botnetdomain is not null or botnetip venttime)) as first_seen, max(nanosec_to_sec(e n from \$log-dns where \$filter group by dvid, q dtype, is_botnet order by total_num desc)### t</pre>	is not null) as venttime)) as name, f_user,

group by qname, srcip order by total_num desc			
Dataset Name	Description	Log Category	
360-degree-security-Application- Visiblity-and-Control-Summary	Application Visibolity and Control Summary	app-ctrl	

seen>=\$start time and first seen<\$end time and tdtype is not null and qname is not null

select
 appcat,
 count(distinct app) as total_num
from

###(select appcat, app from \$log where \$filter and app is not null and appcat is not null group by appcat, app)### t group by appcat order by total_num desc

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

select
 threat_name,
 count(distinct threats) as total_num
from

(

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

Dataset Name	Description	Log Category
360-degree-security-Data-Exfiltration- Detection-and-Prevention-Summary	Data Exfiltration Summary	dlp

select
 data_loss,
 count(*) as total_num

```
from
```

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

Dataset Name	Description	Log Category
360-degree-security-Endpoint- Protection-Summary	Endpoint Protection	fct-traffic

select

```
blocked_event,
  count(*) as total_num
from
  (
    select
    (
```

case utmevent when & #039;antivirus' then 'Malware Deteced and Blocked' when 'appfirewall' then 'Risk Application Blocked' when 'webfilter' then (case when coalesce (nullifna(`user`), ipstr(`srcip`)) is not null then 'Web Sites Violation Blocked' else 'Non User Initiated Web Visits' end) else NULL end) as blocked_event from \$log where \$filter and utmaction in ('blocked', 'quarantined')) t where blocked_event is not null group by blocked_ event order by total num desc

Macro Reference List

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

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

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

Macro Name	Description	Dataset Used	Log Category
Web Category with Highest Bandwidth	Web filtering category with the highest bandwidth usage	Top-Web-Category-by- Bandwidth	Web Filter
Web Category with Highest Session Count	Web filtering category with the highest session count	Top-Web-Category-by- Sessions	Web Filter
Website with Highest Bandwidth	Website with the highest bandwidth usage	Top-Web-Sites-by- Bandwidth	Web Filter
Website with Highest Session Count	Website with the highest session count	Top-Web-Sites-by- Sessions	Web Filter

Change Log

Date	Change Description
2022-06-02	Initial release.



www.fortinet.com

Copyright© 2022 Fortinet, Inc. All rights reserved. Fortinet®, FortiGate®, FortiCare® and FortiGuard®, and certain other marks are registered trademarks of Fortinet, Inc., 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. 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.