FortiAnalyzer

Datasets

RELEASE 5.0.8

Reference Manual

Datasets

v5.0.8

September 16, 2014

05-508-248138-20140916

Performance metrics contained herein were attained in internal lab tests under ideal conditions, and performance 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 the performance metrics herein. 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 guarantees. 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.

Technical Documentation	http://help.fortinet.com
Forums	https://support.fortinet.com/forum
Customer Service & Support	https://support.fortinet.com
Training	http://training.fortinet.com
FortiGuard Threat Research & Response	http://www.fortiguard.com
License	http://www.fortinet.com/doc/legal/EULA.pdf
Document Feedback	Email: techdocs@fortinet.com

Contents

Introduction

This document provides information about the various types of FortiAnalyzer datasets which are created based on the FortiGate log SQL tables and messages. These datasets are used to create charts and reports.

It describes the procedure for creating custom datasets, and also lists the types of log tables used to assist in writing SQL queries to create the datasets.

Overview

FortiAnalyzer uses the PostgreSQL and remote MySQL databases to store the log data generated by the FortiGate.
To create a chart based on the FortiGate logs in a local or remote database, you can use either the predefined datasets, or create your own custom datasets by querying the logs in the SQL database in FortiAnalyzer.

This chapter includes the following topics:

Understanding Datasets and Macros

FortiAnalyzer datasets are collections of log messages from monitored devices.
If the FortiAnalyzer unit is not receiving data from a device, or logging is not enabled under *System > Config > SQL Database*, it does not create log tables for that device.

Charts in FortiAnalyzer are generated based on the datasets. To create a chart, you can use either the predefined datasets, or 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 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 etc.

To view and configure datasets, go to ***Reports > Advanced > Dataset*** in the left navigation pane of the web-based manager. For more information, refer to the *Dataset* section in the *FortiAnalyzer Administration Guide.*

To view and configure macros, go to ***Reports> Macro Library*** in the left navigation pane of the web-based manager. For more information, refer to the *Macro Librar*y section in the *FortiAnalyzer Administration Guide.*

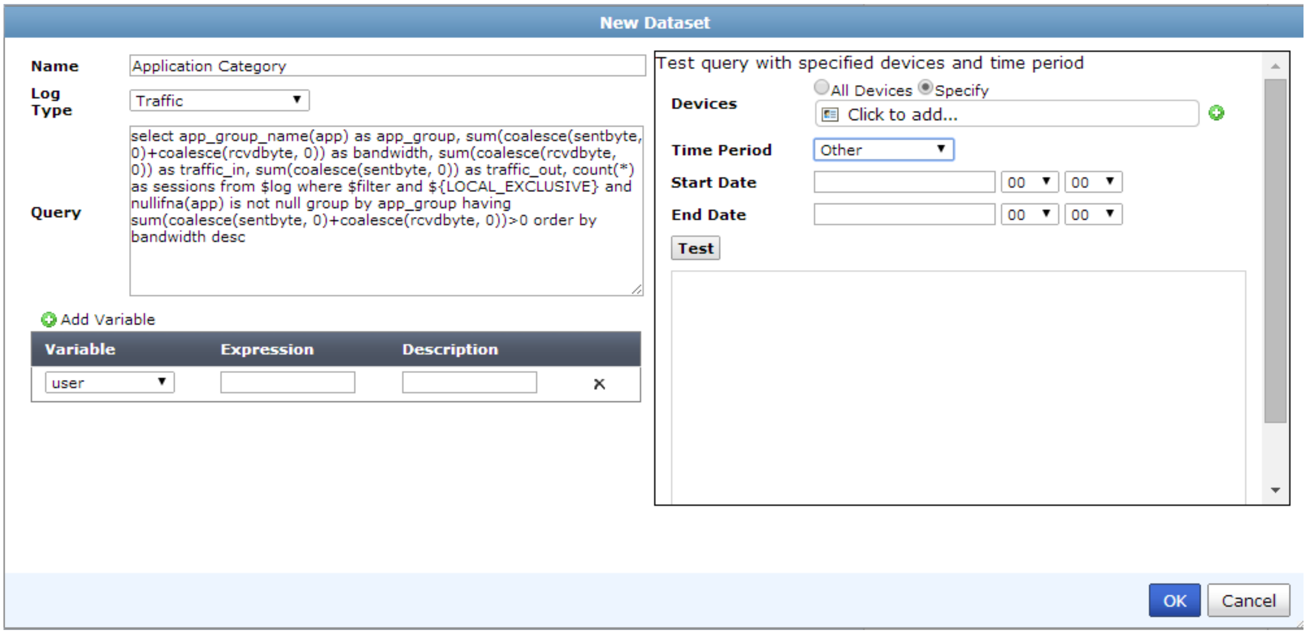
Note: FortiAnalyzer v5.0 Patch Release 5 introduced new datasets for SIP and SCCP. FortiAnalyzer v5.0 Patch Release 6 introduces new datasets for Botnet (Botnet-Activity-By-Sources, Botnet-Infected-Hosts, Botnet-Sources, Botnet-Timeline, and Detected-Botnet).

Creating Custom Datasets

This section describes the procedure to create datasets in the FortiAnalyzer web-based manager. You can also use the CLI command `config sql-report dataset` to create datasets.

To create a custom dataset in the web-based manager

1. Go to ***Reports> Advanced > Dataset.***
2. Click ***Create New.***
3. Configure the following, then click ***OK.***



New Dataset

Name: Application Category

Log Type: Traffic

Query:

```
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 ${LOCAL_EXCLUSIVE} and
nullifna(app) is not null group by app_group having
sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 order by
bandwidth desc
```

Variable	Expression	Description
user		

Test query with specified devices and time period

Devices: ○ All Devices ● Specify

Time Period: Other

Start Date: 00 00

End Date: 00 00

The following table describes the GUI fields of the **New Dataset** dialog box.

Field	Description
Name	Name of the data set.
Log Type	Log Type to be used for the data set. $log is used in the SQL query to represent the log type you select, and it is run against all tables of this type.
Devices	Select **All Devices** to create datasets on all of FortiAnalyzer managed devices. or select **Specify** to choose a device on which you want to create the dataset.
Query	Enter the SQL query syntax to retrieve the log data you want from the SQL database.
Time Period	Select to use logs from a time frame. Select **Other** to define a custom time frame by selecting the **Start Time** and **End Time**. $filter is used in the SQL query "where" clause to limit the results to the period you select.
Test	Click to test whether or not the SQL query is successful.

Testing SQL Query

You can verify the SQL query that you used to create the custom dataset before saving the dataset configuration by testing and viewing the query results.

To test a SQL query:

1. Click ***Test*** after entering the SQL query in the **New Dataset** dialog box.

The query results are displayed. If the query is not successful, an error message appears in the results pane.

Examples of SQL Query Errors

Here are some example error messages and possible causes:

Syntax Errors

`You have an error in your SQL syntax (remote/MySQL) or` ERROR: `syntax error at or near... (local/PostgreSQL)`

• Check that SQL keywords are spelled correctly, and that the query is well-formed.
• Table and column names are demarked by grave accent (`) characters. Single (') and double (") quotation marks will cause an error.

`No data is covered.`

• The query is correctly formed, but no data has been logged for the log type. Check that you have configured the FortiAnalyzer unit to save that log type. Under ***System > Config > SQL Database***, ensure that the log type is checked.

Connection Errors

If well formed queries do not produce results, and logging is turned on for the log type, there may be a database configuration problem with the remote database.

Ensure that:
• MySQL is running and using the default port`3306`.
• You have created an empty database and a user with create permissions for the database.

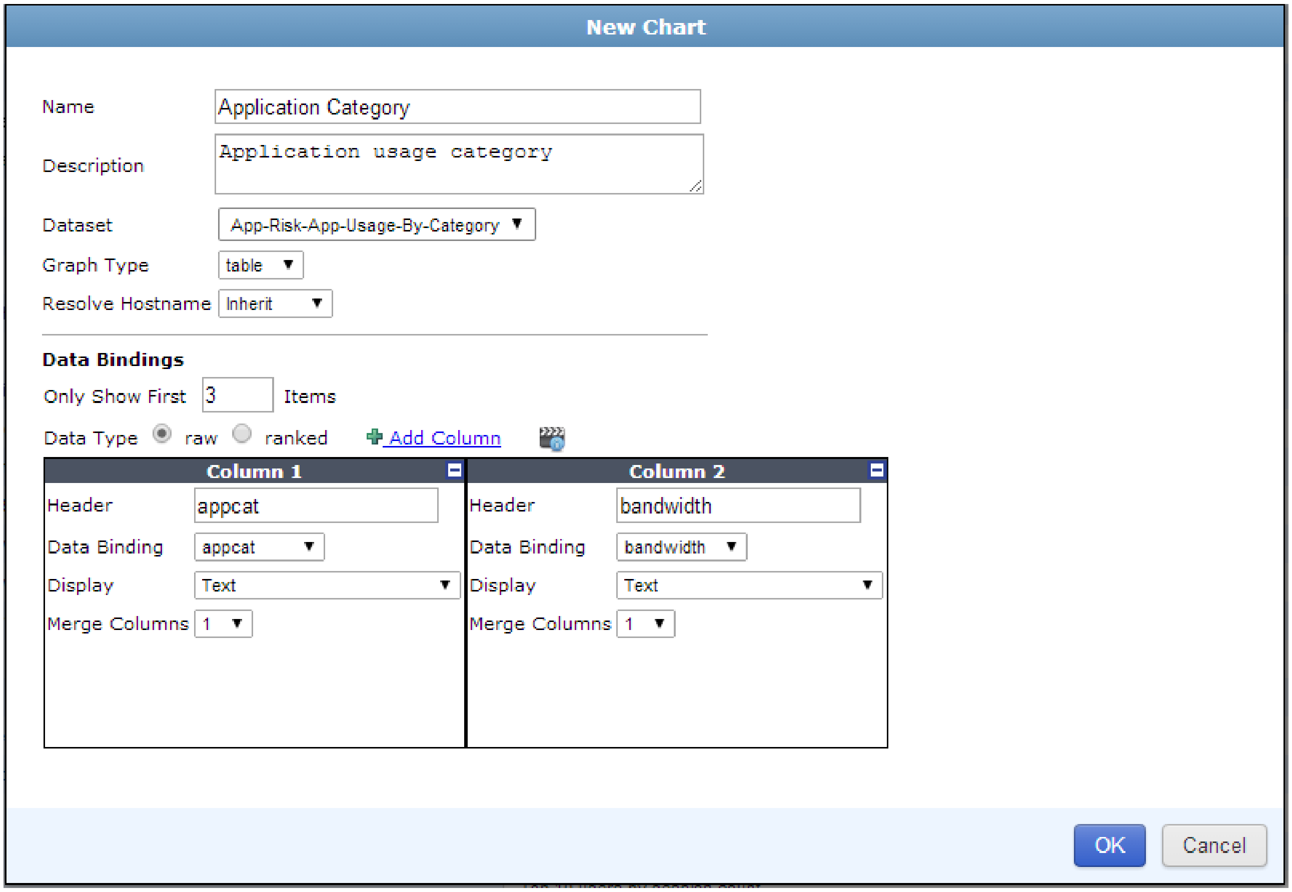
Here is an example of creating a new MySQL database named fazlogs, and adding a user for the database:

```
#Mysql –u root –p
mysql> Create database fazlogs;
mysql> Grant all privileges on fazlogs.* to ‘fazlogger’@’*’
identified by ‘fazpassword’;
mysql> Grant all privileges on fazlogs.* to
‘fazlogger’@’localhost’ identified by ‘fazpassword’;
```

For more information about using SQL queries for creating datasets, refer to the *FortiAnalyzer™ and FortiGate™ Version 4.0 MR2 SQL Log Database Query Technical Note* on the Fortinet Documentation Library at docs.fortinet.com.

Examples of Custom Datasets

The following examples illustrate how to create custom datasets using the web-based manager GUI and CLI. Once created, you can use the datasets to configure chart templates under ***Reports > Chart Library***.



New Chart

Name	Application Category
Description	Application usage category
Dataset	App-Risk-App-Usage-By-Category
Graph Type	table
Resolve Hostname	Inherit

Data Bindings

Only Show First 3 Items

Data Type ● raw ○ ranked Add Column

Column 1		Column 2	
Header	appcat	Header	bandwidth
Data Binding	appcat	Data Binding	bandwidth
Display	Text	Display	Text
Merge Columns	1	Merge Columns	1

Example 1: Distribution of applications by type in the last 24 hours

GUI Procedure

1. Go to ***Reports> Advanced > Dataset.***
2. Click ***Create New.***
3. Select ***Application Control*** under **Log Type**.
4. Enter a name, such as "***apps_type_24hrs***".
5. Select ***Last N Hours*** under **Time Period**.
6. Enter the query:

```
SELECT app_type, COUNT( * ) AS totalnum
FROM $log
WHERE $filter
```

```
AND app_type IS NOT NULL
GROUP BY app_type
ORDER BY totalnum DESC
```

CLI Procedure

To perform the same task using the CLI, use these commands:

```
config sql-report dataset
 edit apps_type_24hrs

      set log-type app-ctrl

      set time-period last-n-hours

      set period-last-n 24

      set query "SELECT app_type, COUNT( * ) AS totalnum
     FROM $log
     WHERE $filter AND app_type IS NOT NULL
     GROUP BY app_type
     ORDER BY totalnum DESC"

end
```

Notes:

- $filter restricts the query result to the time period specified; in this case, it’s the past 24 hours.
- $log queries all application control logs
- The application control module classifies each firewall session in app_type. One firewall session may be classified to multiple app_types. For example, an HTTPsession can be classified to: HTTP, Facebook, etc.
- Some app/app_types may not be able to detected, then the ‘app_type’ field may be null or ‘N/A’. These will be ignored by this query.
- The result is ordered by the total session number of the same app_type. The most frequent app_types will appear first.

Example 2: Top 100 applications by bandwidth in the last 24 hours

GUI Procedure

1. Go to ***Reports> Advanced > Dataset.***
2. Click ***Create New.***
3. Select ***Application Control*** under **Log Type**.
4. Enter a name, such as "***top_100_aps_24hrs***".
5. Select ***Last N Hours*** under **Time Period**.
6. Enter the query:

```
SELECT (
TIMESTAMP - TIMESTAMP %3600
) AS hourstamp, app, service, SUM( sent + rcvd ) AS volume
FROM $log
WHERE $filter and app IS NOT NULL
GROUP BY app
ORDER BY volume DESC
LIMIT 100
```

CLI Procedure

To perform the same task using the CLI, use these commands:

```
config sql-report dataset
 edit top_100_apps_24hrs

     set log-type traffic

     set time-period last-n-hours

     set period-last-n 24

     set query "SELECT ( TIMESTAMP - TIMESTAMP %3600 ) AS

     hourstamp, app, service, SUM( sent + rcvd ) AS volume

     FROM $log
     WHERE $filter and app IS NOT NULL
     GROUP BY app

     ORDER BY volume DESC LIMIT 100"

end
```

Notes:

- (timestamp-timestamp%3600) as hourstamp - this calculates an "hourstamp" to indicate bandwidth per hour.
- `SUM( sent + rcvd ) AS volume` - this calculates the total sent and received bytes.
- `ORDER BY volume DESC` - this orders the results by descending volume (largest volume first).
- `LIMIT 100` - this lists only the top 100 applications.

Log Database Tables

The FortiAnalyzer and FortiGate units create a database table for each managed device and each log type to record log data.

SQL tables follow the naming convention of `[Device Name]-[SQL table type]-[timestamp]`, where the SQL table type is one of the types listed in the table below.

Log Type	SQL Table Type	Description
Traffic	tlog	The traffic log records all traffic to and through the FortiGate interface.
Event	elog	The event log records management and activity events. For example, when an administrator logs in or logs out of the web-based manager.
Antivirus	vlog	The antivirus log records virus incidents in Web, FTP, and email traffic.
Webfilter	wlog	The web filter log records HTTP FortiGate log rating errors including web content blocking actions that the FortiGate unit performs.
Attack	attack_log	The attack log records attacks that are detected and prevented by the FortiGate unit.
Data Leak Prevention	dlog	The Data Leak Prevention log records log data that is considered sensitive and that should not be made public. This log also records data that a company does not want entering their network.
Application Control	app_control_log	The application control log records data detected by the FortiGate unit and the action taken against the network traffic depending on the application that is generating the traffic, for example, instant messaging software, such as MSN Messenger.
Spamfilter	spamfilter_log	The spam filter log records blocking of email address patterns and content in SMTP, IMAP, and POP3 traffic.

To view all the tables created in a database, use the following commands:

- local (PostgreSQL) database: `SELECT * FROM pg_tables`
- remote (MySQL): `SHOW TABLES`

FortiAnalyzer and FortiGate logs also include log sub-types, which are types of log messages that are within the main log type. For example, in the event log type there are the subtype admin log messages.

For more information on FortiGate Log Types and Messages, refer to the FortiOS/FortiGate *Log Message Reference Guide* on the Fortinet Documentation Library at: docs.fortinet.com.

Log Type	Sub Type
traffic (Traffic Log)	allowed - Policy allowed traffic violation - Policy violation traffic Other
event (Event Log)	For FortiGate devices: • system - System activity event • ipsec - IPSec negotiation event • dhcp - DHCP service event • ppp - L2TP/PPTP/PPPoE service event • admin - admin event • ha - HA activity event • auth - Firewall authentication event • pattern - Pattern update event • alertemail - Alert email notifications • chassis - FortiGate-4000 and FortiGate-5000 series chassis event • sslvpn-user - SSL VPN user event • sslvpn-admin - SSL VPN administration event • sslvpn-session - SSL VPN session even • his-performance - performance statistics • vipssl - VIP SSL events • ldb-monitor - LDB monitor events

Log Type	Sub Type
dlp (Data Leak Prevention)	• dlp - Data Leak Prevention
app-crtl (Application Control Log)	• app-crtl-all - All application control
virus (Antivirus Log)	• infected - Virus infected • filename - Filename blocked • oversize - File oversized
webfilter (Web Filter Log)	• content - content block • urlfilter - URL filter • FortiGuard block • FortiGuard allowed • FortiGuard error • ActiveX script filter • Cookie script filter • Applet script filter
ips (Attack Log)	• signature - Attack signature • anomaly - Attack anomaly
emailfilter (Spam Filter Log)	• SMTP • POP3 • IMAP

Dataset Reference List

The following table lists the available predefined datasets applicable to a FortiGate device reported by FortiAnalyzer.

For documentation and technical support reference purposes, this table contains the dataset names, SQL query syntax for each dataset, and the log category of the dataset.

Dataset Name	Description	Log Category	Query Syntax

App-Risk-App-Usage-By-Category	Application risk application usage by category	Traffic	SELECT appcat,
			    sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth
			FROM $log
			WHERE $filter
			  AND logid_to_int(logid) NOT IN (4,
			                                  7,
			                                  14)
			  AND nullifna(appcat) IS NOT NULL
			GROUP BY appcat
			ORDER BY bandwidth DESC
			SELECT app_group_name(app) AS app_group,
			    appcat,
			    sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth,
			    count(*) AS num_session
			FROM $log
			WHERE $filter
			  AND logid_to_int(logid) NOT IN (4,
			                                  7,
			                                  14)
			  AND nullifna(app) IS NOT NULL
			GROUP BY app_group,
			     appcat
			ORDER BY bandwidth DESC

App-Risk-Application-Activity-APP	Application risk application activity	Traffic	SELECT app_group_name(app) AS app_group,
			    appcat,
			    sum(coalesce(sentbyte, 0)+COALESCE (rcvdbyte, 0)) AS bandwidth,
			    count(*) AS num_session
			FROM $log
			WHERE $filter
			  AND logid_to_int(logid) NOT IN (4,
			                                  7,
			                                  14)
			  AND nullifna(app) IS NOT NULL
			GROUP BY app_group,
			     appcat
			ORDER BY bandwidth DESC

App-Risk-Applications-Running-Over-HTTP	Application risk applications running over HTTP	Traffic	SELECT app_group_name(app) AS app_group,
			    service,
			    count(*) AS sessions,
			    sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth
			FROM $log

Dataset Name	Description	Log Category	Query Syntax
			WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND nullifna(app) IS NOT NULL AND service IN ('80/tcp', '443/tcp', 'HTTP', 'HTTPS', 'http', 'https') GROUP BY app_group, service HAVING sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 ORDER BY bandwidth DESC
App-Risk-Breakdown-Of-Risk-Applications	Application risk breakdown of risk applications	Traffic	SELECT d_behavior, count(*) AS number FROM $log t1 INNER JOIN app_mdata t2 ON t1.appid=t2.id WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND d_risk>0 GROUP BY d_behavior ORDER BY number DESC

App-Risk-DLP-UTM-Event	Application risk DLP UTM event	Traffic	SELECT utmsubtype,
			sum(number) AS number
			FROM (###
			(SELECT utmsubtype,
			count(*) AS number
			FROM $log-traffic
			WHERE $filter
			AND logid_to_int(logid) NOT IN (4,
			7,
			14)
			AND utmevent='dlp'
			AND utmsubtype IS NOT NULL
			GROUP BY utmsubtype
			ORDER BY number DESC)###
			UNION ALL ###
			(SELECT subtype AS utmsubtype,
			count(*) AS number
			FROM $log-dlp
			WHERE $filter
			AND subtype IS NOT NULL
			GROUP BY subtype
			ORDER BY number DESC)###) t
			GROUP BY utmsubtype
			ORDER BY number DESC

App-Risk-High-Risk-Application	Application risk high risk application	Traffic	SELECT d_risk, d_behavior, t2.id, t2.name, t2.app_cat, t2.technology, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth,

Dataset Name	Description	Log Category	Query Syntax
			count(*) AS sessions FROM $log t1 INNER JOIN app_mdata t2 ON t1.appid=t2.id WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND d_behavior IS NOT NULL GROUP BY t2.id ORDER BY d_risk DESC, sessions DESC
App-Risk-Number-Of-Applications-By-Risk-Behavior	Application risk number of applications by risk behavior	Traffic	SELECT d_risk, coalesce(d_behavior, 'Other Applications') AS f_behavior, count(*) AS number FROM $log t1 INNER JOIN app_mdata t2 ON t1.appid=t2.id WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) GROUP BY d_risk, d_behavior ORDER BY d_risk DESC, number DESC
App-Risk-Reputation-Top-Devices-By-Scores	Application risk reputation top devices by scores	Traffic	SELECT devtype, coalesce(nullifna(`srcname`),nullifna(`srcmac`), ipstr(`srcip`)) AS dev_src, sum(crscore%65536) AS scores FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND crscore IS NOT NULL GROUP BY devtype, dev_src HAVING sum(crscore%65536)>0 ORDER BY scores DESC

App-Risk-Reputation-Top-Users-By-Scores	Application risk reputation top users by scores	Traffic	SELECT coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) AS user_src,
			sum(crscore%65536) AS scores
			FROM $log
			WHERE $filter
			AND logid_to_int(logid) NOT IN (4,
			7,
			14)
			AND crscore IS NOT NULL
			GROUP BY user_src HAVING sum(crscore%65536)>0
			ORDER BY scores DESC

App-Risk-Top-Critical-Threat-Vectors	Application risk top critical threat vectors	attack	SELECT attack, severity, REF, count(*) AS totalnum FROM $log WHERE $filter AND severity='critical' AND nullifna(attack) IS NOT NULL GROUP BY attack, severity,

Dataset Name	Description	Log Category	Query Syntax
			REF ORDER BY totalnum DESC
App-Risk-Top-High-Threat-Vectors	Application risk top high threat vectors	attack	SELECT attack, severity, REF, count(*) AS totalnum FROM $log WHERE $filter AND severity='high' AND nullifna(attack) IS NOT NULL GROUP BY attack, severity, REF ORDER BY totalnum DESC
App-Risk-Top-Info-Threat-Vectors	Application risk top info threat vectors	attack	SELECT attack, severity, REF, count(*) AS totalnum FROM $log WHERE $filter AND severity='info' AND nullifna(attack) IS NOT NULL GROUP BY attack, severity, REF ORDER BY totalnum DESC
App-Risk-Top-Low-Threat-Vectors	Application risk top low threat vectors	attack	SELECT attack, severity, REF, count(*) AS totalnum FROM $log WHERE $filter AND severity='low' AND nullifna(attack) IS NOT NULL GROUP BY attack, severity, REF ORDER BY totalnum DESC
App-Risk-Top-Medium-Threat-Vectors	Application risk top medium threat vectors	attack	SELECT attack, severity, REF, count(*) AS totalnum FROM $log WHERE $filter AND severity='medium' AND nullifna(attack) IS NOT NULL GROUP BY attack, severity, REF ORDER BY totalnum DESC

App-Risk-Top-Threat-Vectors	Application risk top threat vectors	attack	SELECT severity,
			count(*) AS totalnum
			FROM $log
			WHERE $filter
			GROUP BY severity
			ORDER BY totalnum DESC

Dataset Name	Description	Log Category	Query Syntax
App-Risk-Top-User-Source-By-Sessions	Application risk top user source by session count	Traffic	SELECT srcip, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) AS user_src, count(*) AS sessions FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND srcip IS NOT NULL GROUP BY srcip, user_src ORDER BY sessions DESC
App-Risk-Virus-Discovered	Application risk virus discovered	Traffic	SELECT dom, sum(totalnum) AS totalnum FROM (### (SELECT $DAY_OF_MONTH AS dom, count(*) AS totalnum FROM $log-traffic WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND utmevent IS NOT NULL AND virus IS NOT NULL GROUP BY dom ORDER BY totalnum DESC)### UNION ALL ### (SELECT $DAY_OF_MONTH AS dom, count(*) AS totalnum FROM $log-virus WHERE $filter AND nullifna(virus) IS NOT NULL AND (eventtype IS NULL OR logver = 52) GROUP BY dom ORDER BY totalnum DESC)###) t GROUP BY dom ORDER BY totalnum DESC
App-Risk-Vulnerability-Discovered	Application risk vulnerability discovered	netscan	SELECT vuln, vulncat, severity, count(*) AS totalnum FROM $log WHERE $filter AND vuln IS NOT NULL GROUP BY vuln, vulncat, severity ORDER BY totalnum DESC
App-Risk-Web-Browsing-Activity-Hostname-Category	Application risk web browsing activity hostname category	Traffic	SELECT DOMAIN, catdesc, sum(visits) AS visits FROM (### (SELECT coalesce(nullifna(hostname), ipstr(`dstip`)) AS DOMAIN, catdesc, count(*) AS visits

Dataset Name	Description	Log Category	Query Syntax
			FROM $log-traffic WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND utmevent IN ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') AND catdesc IS NOT NULL AND catdesc !='Unrated' GROUP BY DOMAIN, catdesc ORDER BY visits DESC)### UNION ALL ### (SELECT coalesce(nullifna(hostname), ipstr(`dstip`)) AS DOMAIN, catdesc, count(*) AS visits FROM $log-webfilter WHERE $filter AND (eventtype IS NULL OR logver = 52) AND catdesc IS NOT NULL AND catdesc !='Unrated' GROUP BY DOMAIN, catdesc ORDER BY visits DESC)###) t GROUP BY DOMAIN, catdesc ORDER BY visits DESC
App-Risk-Web-Browsing-Summary-Category	Application risk web browsing summary category	Traffic	SELECT catdesc, sum(num_sess) AS num_sess, sum(bandwidth) AS bandwidth FROM (### (SELECT catdesc, count(*) AS num_sess, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth FROM $log-traffic WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND utmevent IN ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') AND catdesc IS NOT NULL AND catdesc !='Unrated' GROUP BY catdesc ORDER BY num_sess DESC)### UNION ALL ### (SELECT catdesc, count(*) AS num_sess, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte,

Dataset Name	Description	Log Category	Query Syntax

			0)) AS bandwidth
			FROM $log-webfilter
			WHERE $filter
			AND (eventtype IS NULL
			OR logver = 52)
			AND catdesc IS NOT NULL
			AND catdesc !='Unrated'
			GROUP BY catdesc
			ORDER BY num_sess DESC)###) t
			GROUP BY catdesc
			ORDER BY num_sess DESC

App-Sessions-By-Category	Application sessions by category	Traffic	SELECT appcat,
			count(*) AS sessions
			FROM $log
			WHERE $filter
			AND logid_to_int(logid) NOT IN (4,
			7,
			14)
			AND nullifna(appcat) IS NOT NULL
			GROUP BY appcat
			ORDER BY sessions DESC

app-Top-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 logid_to_int(logid) NOT IN (4, 7, 14) AND action IN ('accept', 'close', 'timeout') GROUP BY TIMESTAMP, user_src, appcat, app, destination ORDER BY bandwidth DESC
app-Top-Blocked-Applications-by-Session	Top blocked applications by session	Traffic	SELECT coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) AS user_src, appcat, app, count(*) AS sessions FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND action IN ('deny', 'blocked', 'reset', 'dropped')

Dataset Name	Description	Log Category	Query Syntax

			GROUP BY user_src,
			appcat,
			app
			ORDER BY sessions DESC

app-Top-Category-and-Applications-by-Bandwidth	Top category and applications by bandwidth usage	Traffic	SELECT appcat, app, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) GROUP BY appcat, app HAVING sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 ORDER BY bandwidth DESC
app-Top-Category-and-Applications-by-Session	Top category and applications by session	Traffic	SELECT appcat, app, count(*) AS sessions FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) GROUP BY appcat, app ORDER BY sessions DESC

appctrl-Top-Blocked-SCCP-Callers	Appctrl top blocked SCCP callers	app-ctrl	SELECT srcname AS caller,
			count(*) AS totalnum
			FROM $log
			WHERE $filter
			AND lower(appcat)='voip'
			AND app='sccp'
			AND action='block'
			AND srcname IS NOT NULL
			GROUP BY caller
			ORDER BY totalnum DESC

appctrl-Top-Blocked-SIP-Callers	Appctrl top blocked SIP callers	app-ctrl	SELECT srcname AS caller,
			count(*) AS totalnum
			FROM $log
			WHERE $filter
			AND srcname IS NOT NULL
			AND lower(appcat)='voip'
			AND app='sip'
			AND action='block'
			GROUP BY caller
			ORDER BY totalnum DESC

Application-Session-History	Application session history	Traffic	SELECT $flex_timescale AS hodex,
			count(*) AS counter
			FROM $log
			WHERE $filter
			AND logid_to_int(logid) NOT IN (4,
			7,
			14)
			GROUP BY hodex
			ORDER BY hodex

Dataset Name	Description	Log Category	Query Syntax
bandwidth-app-Top-Dest-By-Bandwidth-Sessions	Bandwidth application top dest by bandwidth usage sessions	Traffic	SELECT coalesce(nullifna(root_domain(hostname)), ipstr(`dstip`)) AS DOMAIN, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth, sum(coalesce(rcvdbyte, 0)) AS traffic_in, sum(coalesce(sentbyte, 0)) AS traffic_out, count(*) AS sessions FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) GROUP BY appid, DOMAIN HAVING sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 ORDER BY bandwidth DESC
bandwidth-app-Top-Users-By-Bandwidth	Bandwidth application top users by bandwidth usage	Traffic	SELECT coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) AS user_src, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth, sum(coalesce(rcvdbyte, 0)) AS traffic_in, sum(coalesce(sentbyte, 0)) AS traffic_out, count(*) AS sessions FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) GROUP BY user_src HAVING sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 ORDER BY bandwidth DESC

bandwidth-app-Traffic-By-Active-User-Number	Bandwidth application traffic by active user number	Traffic	SELECT hodex,
			count(distinct(user_src)) AS total_user
			FROM ###
			(SELECT $flex_timescale AS hodex,
			coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) AS user_src
			FROM $log
			WHERE $filter
			AND logid_to_int(logid) NOT IN (4,
			7,
			14)
			GROUP BY hodex,
			user_src
			ORDER BY hodex)### t
			GROUP BY hodex
			ORDER BY hodex

bandwidth-app-Traffic-Statistics	Bandwidth application traffic statistics	Traffic	DROP TABLE IF EXISTS stats_temp;
			
			CREATE
			TEMPORARY TABLE stats_temp(total_sessions varchar(255), total_bandwidth varchar(255), ave_session varchar(255), ave_bandwidth varchar(255), active_date varchar(255), total_users varchar(255), total_app varchar(255), total_dest varchar(255));
			
			INSERT INTO stats_temp (total_sessions, total_bandwidth, ave_session, ave_bandwidth)

Dataset Name	Description	Log Category	Query Syntax

			SELECT format_numeric_no_decimal(sum(sessions)) AS total_sessions,
			bandwidth_unit(sum(bandwidth)) AS total_bandwidth,
			format_numeric_no_decimal(cast(sum(sessions)/$days_num AS decimal(18, 0))) AS ave_session,
			bandwidth_unit(cast(sum(bandwidth)/$days_num AS decimal(18, 0))) AS ave_bandwidth
			FROM ###
			(SELECT count(*) AS sessions,
			sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth
			FROM $log
			WHERE $filter
			AND ${LOCAL_EXCLUSIVE})### t;
			
			UPDATE stats_temp
			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 ${LOCAL_EXCLUSIVE}
			GROUP BY dom
			ORDER BY sessions)### t
			GROUP BY dom
			ORDER BY sessions DESC LIMIT 1) AS t1;
			
			UPDATE stats_temp
			SET total_users=t2.totalnum
			FROM
			(SELECT format_numeric_no_decimal(count(distinct(user_src))) AS totalnum
			FROM ###
			(SELECT distinct(coalesce(nullifna(\`user\`), nullifna(\`unauthuser\`), ipstr(\`srcip\`))) AS user_src
			FROM $log
			WHERE $filter
			AND ${LOCAL_EXCLUSIVE})### t) AS t2;
			
			UPDATE stats_temp
			SET total_app=t3.totalnum
			FROM
			(SELECT format_numeric_no_decimal(count(distinct(app_group_name(app)))) AS totalnum
			FROM ###
			(SELECT distinct(app_group_name(app)) AS app
			FROM $log
			WHERE $filter
			AND ${LOCAL_EXCLUSIVE})### t) AS t3;
			
			UPDATE stats_temp
			SET total_dest=t4.totalnum
			FROM
			(SELECT format_numeric_no_decimal(count(distinct(dstip))) AS totalnum

Dataset Name	Description	Log Category	Query Syntax

			FROM ###
			  (SELECT distinct(dstip) AS dstip
			   FROM $log
			   WHERE $filter
			    AND ${LOCAL_EXCLUSIVE})### t) AS t4;
			
			SELECT 'Total Sessions' AS summary,
			       total_sessions AS stats
			FROM stats_temp
			UNION ALL
			SELECT 'Total Bytes Transferred' AS summary,
			       total_bandwidth AS stats
			FROM stats_temp
			UNION ALL
			SELECT 'Most Active Date By Sessions' AS summary,
			       active_date AS stats
			FROM stats_temp
			UNION ALL
			SELECT 'Total Users' AS summary,
			       total_users AS stats
			FROM stats_temp
			UNION ALL
			SELECT 'Total Applications' AS summary,
			       total_app AS stats
			FROM stats_temp
			UNION ALL
			SELECT 'Total Destinations' AS summary,
			       total_dest AS stats
			FROM stats_temp
			UNION ALL
			SELECT 'Average Sessions Per Day' AS summary,
			       ave_session AS stats
			FROM stats_temp
			UNION ALL
			SELECT 'Average Bytes Per Day' AS summary,
			       ave_bandwidth AS stats
			FROM stats_temp

Botnet-Activity-By-Sources	Botnet activity by sources	Traffic	SELECT app,
			       coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) AS user_src,
			       count(*) AS events
			FROM $log
			WHERE $filter
			  AND logid_to_int(logid) NOT IN (4,
			                                 7,
			                                 14)
			  AND appcat='Botnet'
			  AND nullifna(app) IS NOT NULL
			GROUP BY app,
			         user_src
			ORDER BY events DESC

Botnet-Infected-Hosts	Botnet infected hosts	Traffic	SELECT coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) AS user_src,
			       devtype,
			       coalesce(srcname, srcmac) AS host_mac,
			       count(*) AS events
			FROM $log
			WHERE $filter

Dataset Name	Description	Log Category	Query Syntax
			AND logid_to_int(logid) NOT IN (4, 7, 14) AND appcat='Botnet' GROUP BY user_src, devtype, host_mac ORDER BY events DESC
Botnet-Sources	Botnet sources	Traffic	SELECT dstip, root_domain(hostname) AS DOMAIN, count(*) AS events FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND appcat='Botnet' AND dstip IS NOT NULL GROUP BY dstip, DOMAIN ORDER BY events DESC

Botnet-Timeline	Botnet timeline	Traffic	SELECT $flex_timescale AS hodex,
			count(*) AS events
			FROM $log
			WHERE $filter
			AND logid_to_int(logid) NOT IN (4,
			7,
			14)
			AND appcat='Botnet'
			GROUP BY hodex
			ORDER BY hodex DESC

Botnet-Victims	Botnet victims	Traffic	SELECT coalesce(nullifna(\`user\`), nullifna(\`unauthuser\`), ipstr(\`srcip\`)) AS user_src, count(*) AS events FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND appcat='Botnet' AND srcip IS NOT NULL GROUP BY user_src ORDER BY events DESC

content-Count-Total-SCCP-Call-Registrations-by-Hour-of-Day	Content count total SCCP call registrations by hour of day	content	SELECT $hour_of_day AS hourstamp,
			count(*) AS totalnum
			FROM $log
			WHERE $filter
			AND proto='sccp'
			AND kind='register'
			GROUP BY hourstamp
			ORDER BY hourstamp

content-Count-Total-SCCP-Calls-Duration-by-Hour-of-Day	Content count total SCCP calls duration by hour of day	content	SELECT $hour_of_day AS hourstamp,
			sum(duration) AS sccp_usage
			FROM $log
			WHERE $filter
			AND proto='sccp'

Dataset Name	Description	Log Category	Query Syntax

			AND kind='call-info'
			AND status='end'
			GROUP BY hourstamp
			ORDER BY hourstamp

content-Count-Total-SCCP-Calls-per-Status	Content count total SCCP calls per status	content	SELECT status,
			count(*) AS totalnum
			FROM $log
			WHERE $filter
			AND proto='sccp'
			AND kind='call-info'
			GROUP BY status
			ORDER BY totalnum DESC

content-Count-Total-SIP-Call-Registrations-by-Hour-of-Day	Content count total SIP call registrations by hour of day	content	SELECT $hour_of_day AS hourstamp,
			count(*) AS totalnum
			FROM $log
			WHERE $filter
			AND proto='sip'
			AND kind='register'
			GROUP BY hourstamp
			ORDER BY hourstamp

content-Count-Total-SIP-Calls-per-Status	Content count total SIP calls per status	content	SELECT status,
			count(*) AS totalnum
			FROM $log
			WHERE $filter
			AND proto='sip'
			AND kind='call'
			GROUP BY status
			ORDER BY totalnum DESC

content-Dist-Total-SIP-Calls-by-Duration	Content dist total SIP calls by duration	content	SELECT (CASE WHEN duration < 60 THEN 'LESS_ONE_MIN' WHEN duration < 600 THEN 'LESS_TEN_MIN' WHEN 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
default-AP-Detection-Summary-by-Status-OffWire	Default access point detection summary by status off-wire	event	SELECT (CASE apstatus WHEN 1 THEN 'rogue' WHEN 2 THEN 'accepted' WHEN 3 THEN 'suppressed' ELSE 'others' END) AS ap_full_status, count(*) AS totalnum FROM (SELECT apstatus, bssid, ssid FROM ### (SELECT apstatus,

Dataset Name	Description	Log Category	Query Syntax
			bssid, ssid, count(*) AS subtotal FROM $log WHERE $filter AND apstatus IS NOT NULL AND apstatus!=0 AND bssid IS NOT NULL AND onwire='no' AND logid_to_int(logid) IN (43527, 43521, 43525) GROUP BY apstatus, bssid, ssid ORDER BY subtotal DESC)### t GROUP BY apstatus, bssid, ssid) t GROUP BY ap_full_status ORDER BY totalnum DESC
default-AP-Detection-Summary-by-Status-OnWire	Default access point detection summary by status on-wire	event	SELECT (CASE apstatus WHEN 1 THEN 'rogue' WHEN 2 THEN 'accepted' WHEN 3 THEN 'suppressed' ELSE 'others' END) AS ap_full_status, count(*) AS totalnum FROM (SELECT apstatus, bssid, ssid FROM ### (SELECT apstatus, bssid, ssid, count(*) AS subtotal FROM $log WHERE $filter AND apstatus IS NOT NULL AND apstatus!=0 AND bssid IS NOT NULL AND onwire='yes' AND logid_to_int(logid) IN (43527, 43521, 43525) GROUP BY apstatus, bssid, ssid ORDER BY subtotal DESC)### t GROUP BY apstatus, bssid, ssid) t GROUP BY ap_full_status ORDER BY totalnum DESC
default-Email-Top-Receivers-By-Band-	Default email top receivers by bandwidth	Traffic	SELECT coalesce(nullifna(\`user\`), nullifna(\`unauthuser\`), ipstr(\`srcip\`)) AS user_src, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS band-

Dataset Name	Description	Log Category	Query Syntax
width	usage		width FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND service IN ('pop3', 'POP3', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/tcp', 'pop3s', 'POP3S', '995/tcp') GROUP BY user_src HAVING sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 ORDER BY bandwidth DESC
default-Email-Top-Receivers-By-Count	Default email top receivers by count	Traffic	SELECT coalesce(nullifna(\`user\`), nullifna(\`unauthuser\`), ipstr(\`srcip\`)) AS user_src, count(*) AS requests FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND service IN ('pop3', 'POP3', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/tcp', 'pop3s', 'POP3S', '995/tcp') GROUP BY user_src ORDER BY requests DESC
default-Email-Top-Senders-By-Bandwidth	Default email top senders by bandwidth usage	Traffic	SELECT coalesce(nullifna(\`user\`), nullifna(\`unauthuser\`), ipstr(\`srcip\`)) AS user_src, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND service IN ('smtp', 'SMTP', '25/tcp', '587/tcp', 'smtps', 'SMTPS',

Dataset Name	Description	Log Category	Query Syntax

			'465/tcp')
			GROUP BY user_src HAVING sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0
			ORDER BY bandwidth DESC

default-Managed-AP-Summary	Default managed access point summary	event	SELECT (CASE
			WHEN (action LIKE '%join%'
			AND logid_to_int(logid)=43522) THEN 'Authorized'
			ELSE 'Unauthorized'
			END) AS ap_status,
			count(*) AS totalnum
			FROM $log
			WHERE $filter
			AND logid_to_int(logid)=43522
			GROUP BY ap_status
			ORDER BY totalnum DESC

default-SELECTed-AP-Details-OffWire	Default SELECTed access point details off-wire	event	SELECT (CASE apstatus WHEN 0 THEN 'unclassified' WHEN 1 THEN 'rogue' WHEN 2 THEN 'accepted' WHEN 3 THEN 'suppressed' ELSE 'others' END) AS ap_full_status, devid, vd, ssid, bssid, manuf, rssi, channel, radioband, FROM_dtime(min(dtime)) AS first_seen, FROM_dtime(max(dtime)) AS last_seen, detectionmethod, itime, onwire AS on_wire FROM $log WHERE $filter AND apstatus IS NOT NULL AND bssid IS NOT NULL AND onwire='no' AND logid_to_int(logid)=43521 GROUP BY ap_full_status, devid, vd, ssid, bssid, manuf, rssi, channel, radioband, detectionmethod, itime, onwire, apstatus
default-SELECTed-AP-Details-OnWire	Default SELECTed access point	event	SELECT (CASE apstatus WHEN 0 THEN 'unclassified' WHEN 1 THEN 'rogue' WHEN 2 THEN 'accepted'

Dataset Name	Description	Log Category	Query Syntax
	details on-wire		WHEN 3 THEN 'suppressed' ELSE 'others' END) AS ap_full_status, devid, vd, ssid, bssid, manuf, rssi, channel, radioband, FROM_dtime(min(dtime)) AS first_seen, FROM_dtime(max(dtime)) AS last_seen, detectionmethod, itime, onwire AS on_wire FROM $log WHERE $filter AND apstatus IS NOT NULL AND bssid IS NOT NULL AND onwire='yes' AND logid_to_int(logid)=43521 GROUP BY ap_full_status, devid, vd, ssid, bssid, manuf, rssi, channel, radioband, detectionmethod, itime, onwire, apstatus
default-Top-Dial-Up-User-Of-Vpn-Tunnel-By-Bandwidth	Default top dial up user of VPN tunnel by bandwidth usage	Traffic	SELECT coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) AS user_src, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND vpntype IN ('ipsec-dynamic', 'sslvpn') GROUP BY user_src HAVING sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 ORDER BY bandwidth DESC
default-Top-Email-Senders-By-Count	Default top email senders by count	Traffic	SELECT coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) AS user_src, count(*) AS requests FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND service IN ('smtp',

Dataset Name	Description	Log Category	Query Syntax
			'SMTP', '25/tcp', '587/tcp', 'smtps', 'SMTPS', '465/tcp') GROUP BY user_src ORDER BY requests DESC
default-Top-IPSEC-Vpn-Dial-Up-User-By-Bandwidth	Default top IPsec VPN dial up user by bandwidth usage	event	SELECT user_src, sum(sent_end-sent_beg+rcvd_end-rcvd_beg) AS bandwidth FROM ### (SELECT coalesce(nullifna(`xauthuser`), nullifna(`user`), ipstr(`remip`)) AS user_src, tunnelid, 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 FROM $log WHERE $filter AND subtype='vpn' AND tunneltype LIKE 'ipsec%' AND action='tunnel-stats' AND NOT (tunnelip IS NULL OR (tunnelip='0.0.0.0' AND coalesce(logver, 0)!=52)) AND tunnelid IS NOT NULL GROUP BY user_src, tunnelid ORDER BY tunnelid)### t GROUP BY user_src HAVING sum(sent_end-sent_beg+rcvd_end-rcvd_beg)>0 ORDER BY bandwidth DESC
default-Top-Sources-Of-SSL-VPN-Tunnels-By-Bandwidth	Default top sources of SSL VPN tunnels by bandwidth usage	event	SELECT remip AS remote_ip, sum(sent_end-sent_beg+rcvd_end-rcvd_beg) AS bandwidth FROM ### (SELECT remip, tunnelid, 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 FROM $log WHERE $filter AND tunneltype LIKE 'ssl%' AND remip IS NOT NULL AND subtype='vpn' AND action='tunnel-stats' AND tunnelid IS NOT NULL GROUP BY tunnelid,

Dataset Name	Description	Log Category	Query Syntax

			remip
			ORDER BY tunnelid)### t
			GROUP BY remote_ip HAVING sum(sent_end-sent_beg+rcvd_end-rcvd_beg)>0
			ORDER BY bandwidth DESC

default-Unclassified-AP-Summary	Default unclassified access point summary	event	SELECT (CASE onwire WHEN 'no' THEN 'off-wire' WHEN 'yes' THEN 'on-wire' ELSE 'others' END) AS ap_status, count(*) AS totalnum FROM ### (SELECT onwire, ssid, bssid, count(*) AS subtotal FROM $log WHERE $filter AND apstatus=0 AND bssid IS NOT NULL AND logid_to_int(logid) IN (43521, 43525, 43527) GROUP BY onwire, ssid, bssid ORDER BY subtotal DESC)### t GROUP BY ap_status ORDER BY totalnum DESC
Detailed-Application-Usage	Detailed application usage	Traffic	SELECT appid, app, appcat, (CASE utmaction WHEN 'blocked' THEN 'Blocked' ELSE 'Allowed' END) AS custaction, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth, count(*) AS num_session FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND nullifna(app) IS NOT NULL AND policyid != 0 GROUP BY appid, app, appcat, custaction ORDER BY bandwidth DESC
Detected-Botnet	Detected botnet	Traffic	SELECT app, count(*) AS events FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14)

Dataset Name	Description	Log Category	Query Syntax

			AND appcat='Botnet'
			AND nullifna(app) IS NOT NULL
			GROUP BY app
			ORDER BY events DESC

drilldown-Top-App-By-Bandwidth	Drilldown top applications by bandwidth usage	Traffic	SELECT appid, app, sum(bandwidth) AS bandwidth FROM ### (SELECT appid, app, coalesce(nullifna(\`user\`), nullifna(\`unauthuser\`), ipstr(\`srcip\`)) AS user_src, dstip, srcintf, dstintf, policyid, count(*) AS sessions, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth FROM $log WHERE $filter-exclude-var AND logid_to_int(logid) NOT IN (4, 7, 14) GROUP BY appid, app, user_src, dstip, srcintf, dstintf, policyid ORDER BY sessions DESC)### t WHERE $filter-var-ONLY AND nullifna(app) IS NOT NULL GROUP BY appid, app HAVING sum(bandwidth)>0 ORDER BY bandwidth DESC
drilldown-Top-App-By-Sessions	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 logid_to_int(logid) NOT IN (4, 7,

Dataset Name	Description	Log Category	Query Syntax
			14) GROUP BY appid, app, user_src, dstip, srcintf, dstintf, policyid ORDER BY sessions DESC)### t WHERE $filter-var-ONLY AND nullifna(app) IS NOT NULL GROUP BY appid, app ORDER BY sessions DESC

drilldown-Top-Attack-Dest	Drilldown top attack dest	attack	SELECT dstip,
			sum(totalnum) AS totalnum
			FROM ###
			(SELECT srcip,
			dstip,
			count(*) AS totalnum
			FROM $log
			WHERE $filter-exclude-var
			GROUP BY srcip,
			dstip
			ORDER BY totalnum DESC)### t
			WHERE $filter-var-ONLY
			AND dstip IS NOT NULL
			GROUP BY dstip
			ORDER BY totalnum DESC

drilldown-Top-Attack-List	Drilldown top attack list	attack	SELECT FROM_itime(itime) AS TIMESTAMP,
			attack,
			srcip,
			dstip
			FROM ###
			(SELECT itime,
			attack,
			srcip,
			dstip
			FROM $log
			WHERE $filter-exclude-var
			ORDER BY itime DESC)### t
			WHERE $filter-var-ONLY
			ORDER BY itime DESC

drilldown-Top-Attack-Source	Drilldown top attack source	attack	SELECT srcip,
			sum(totalnum) AS totalnum
			FROM ###
			(SELECT srcip,
			dstip,
			count(*) AS totalnum
			FROM $log
			WHERE $filter-exclude-var
			GROUP BY srcip,
			dstip
			ORDER BY totalnum DESC)### t
			WHERE $filter-var-ONLY
			AND srcip IS NOT NULL
			GROUP BY srcip
			ORDER BY totalnum DESC

Dataset Name	Description	Log Category	Query Syntax
drilldown-Top-Destination-By-Bandwidth	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 logid_to_int(logid) NOT IN (4, 7, 14) GROUP BY appid, app, user_src, dstip, srcintf, dstintf, policyid ORDER BY sessions DESC)### t WHERE $filter-var-ONLY AND dstip IS NOT NULL GROUP BY dstip HAVING sum(bandwidth)>0 ORDER BY bandwidth DESC
drilldown-Top-Destination-By-Sessions	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 logid_to_int(logid) NOT IN (4, 7, 14) GROUP BY appid, app, user_src, dstip, srcintf, dstintf,

Dataset Name	Description	Log Category	Query Syntax

			policyid
			ORDER BY sessions DESC)### t
			WHERE $filter-var-ONLY
			AND dstip IS NOT NULL
			GROUP BY dstip
			ORDER BY sessions DESC

drilldown-Top-Email-Receiver-By-Count	Drilldown top email receiver by count	Traffic	SELECT recipient, sum(requests) AS requests FROM (### (SELECT recipient, sender, count(*) AS requests, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth FROM $log WHERE $filter-exclude-var AND logid_to_int(logid) NOT IN (4, 7, 14) 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\`

Dataset Name	Description	Log Category	Query Syntax

			ORDER BY requests DESC)###) t
			WHERE $filter-var-ONLY
			AND recipient IS NOT NULL
			GROUP BY recipient
			ORDER BY requests DESC

drilldown-Top-Email-Receiver-By-Volume	Drilldown top email receiver by volume	Traffic	SELECT recipient, sum(bandwidth) AS volume FROM (### (SELECT recipient, sender, count(*) AS requests, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth FROM $log WHERE $filter-exclude-var AND logid_to_int(logid) NOT IN (4, 7, 14) 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

Dataset Name	Description	Log Category	Query Syntax

			WHERE $filter-var-ONLY
			AND recipient IS NOT NULL
			GROUP BY recipient HAVING sum(bandwidth)>0
			ORDER BY volume DESC

drilldown-Top-Email-Receive-Sender-By-Count	Drilldown top email receive sender by count	Traffic	SELECT sender, sum(requests) AS requests FROM (### (SELECT recipient, sender, count(*) AS requests, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth FROM $log WHERE $filter-exclude-var AND logid_to_int(logid) NOT IN (4, 7, 14) 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-var-ONLY

Dataset Name	Description	Log Category	Query Syntax

			AND sender IS NOT NULL
			GROUP BY sender
			ORDER BY requests DESC

drilldown-Top-Email-Receive-Sender-By-Volume	Drilldown top email receive sender by volume	Traffic	SELECT sender, sum(bandwidth) AS volume FROM (### (SELECT recipient, sender, count(*) AS requests, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth FROM $log WHERE $filter-exclude-var AND logid_to_int(logid) NOT IN (4, 7, 14) 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-var-ONLY AND sender IS NOT NULL

Dataset Name	Description	Log Category	Query Syntax

			GROUP BY sender HAVING sum(bandwidth)>0
			ORDER BY volume DESC

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 logid_to_int(logid) NOT IN (4, 7, 14) 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-var-ONLY AND sender IS NOT NULL GROUP BY sender ORDER BY requests DESC
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

Dataset Name	Description	Log Category	Query Syntax
			FROM $log-traffic WHERE $filter-exclude-var AND logid_to_int(logid) NOT IN (4, 7, 14) 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-var-ONLY AND sender IS NOT NULL GROUP BY sender HAVING sum(bandwidth)>0 ORDER BY volume DESC
drilldown-Top-Email-Send-Recipient-By-Count	Drilldown top email send recipient by count	Traffic	SELECT recipient, sum(requests) AS requests FROM (### (SELECT sender, recipient, count(*) AS requests, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth FROM $log-traffic WHERE $filter-exclude-var AND logid_to_int(logid) NOT IN (4, 7, 14) AND service IN ('smtp', 'SMTP', '25/tcp', '587/tcp', 'smtps', 'SMTPS',

Dataset Name	Description	Log Category	Query Syntax
			'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-var-ONLY AND recipient IS NOT NULL GROUP BY recipient ORDER BY requests DESC
drilldown-Top-Email-Send-Recipient-By-Volume	Drilldown top email send recipient by volume	Traffic	SELECT recipient, sum(bandwidth) AS volume FROM (### (SELECT sender, recipient, count(*) AS requests, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth FROM $log-traffic WHERE $filter-exclude-var AND logid_to_int(logid) NOT IN (4, 7, 14) 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,

Dataset Name	Description	Log Category	Query Syntax
			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-var-ONLY AND recipient IS NOT NULL GROUP BY recipient HAVING sum(bandwidth)>0 ORDER BY volume DESC
drilldown-Top-User-By-Bandwidth	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 logid_to_int(logid) NOT IN (4, 7, 14) GROUP BY appid, app, user_src, dstip, srcintf, dstintf, policyid ORDER BY sessions DESC)### t WHERE $filter-var-ONLY AND user_src IS NOT NULL GROUP BY user_src HAVING sum(bandwidth)>0 ORDER BY bandwidth DESC
drilldown-Top-User-By-Sessions	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,

Dataset Name	Description	Log Category	Query Syntax
			dstip, srcintf, dstintf, policyid, count(*) AS sessions, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth FROM $log WHERE $filter-exclude-var AND logid_to_int(logid) NOT IN (4, 7, 14) GROUP BY appid, app, user_src, dstip, srcintf, dstintf, policyid ORDER BY sessions DESC)### t WHERE $filter-var-ONLY AND user_src IS NOT NULL GROUP BY user_src ORDER BY sessions DESC
drilldown-Top-Website-By-Request	Drilldown top website by request	Traffic	SELECT hostname, sum(requests) AS visits FROM (### (SELECT coalesce(nullifna(\`user\`), nullifna(\`unauthuser\`), ipstr(\`srcip\`)) AS user_src, hostname, count(*) AS requests FROM $log-traffic WHERE $filter-exclude-var AND logid_to_int(logid) NOT IN (4, 7, 14) 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 AND hostname IS NOT NULL GROUP BY user_src, hostname

Dataset Name	Description	Log Category	Query Syntax

			ORDER BY requests DESC)###) t
			WHERE $filter-var-ONLY
			AND hostname IS NOT NULL
			GROUP BY hostname
			ORDER BY visits DESC

drilldown-Top-Web-User-By-Visit	Drilldown top web user by visit	Traffic	SELECT user_src, sum(requests) AS visits FROM (### (SELECT coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) AS user_src, hostname, count(*) AS requests FROM $log-traffic WHERE $filter-exclude-var AND logid_to_int(logid) NOT IN (4, 7, 14) 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 AND hostname IS NOT NULL GROUP BY user_src, hostname ORDER BY requests DESC)###) t WHERE $filter-var-ONLY AND user_src IS NOT NULL GROUP BY user_src ORDER BY visits DESC
drilldown-Virus-Detail	Drilldown virus detail	Traffic	SELECT FROM_itime(itime) AS TIMESTAMP, virus, user_src, dstip, hostname, recipient FROM (### (SELECT itime, virus, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) AS user_src, dstip, hostname, recipient FROM $log-traffic

Dataset Name	Description	Log Category	Query Syntax
			WHERE $filter-exclude-var AND logid_to_int(logid) NOT IN (4, 7, 14) AND utmevent IS NOT NULL AND virus IS NOT NULL ORDER BY itime DESC)### UNION ALL ### (SELECT itime, virus, coalesce(nullifna(`user`), ipstr(`srcip`)) AS user_src, dstip, cast(' ' AS char) AS hostname, cast(' ' AS char) AS recipient FROM $log-virus WHERE $filter-exclude-var AND (eventtype IS NULL OR logver = 52) AND nullifna(virus) IS NOT NULL ORDER BY itime DESC)###) t WHERE $filter-var-ONLY ORDER BY itime DESC

Estimated-Browsing-Time	Estimated browsing time	Traffic	SELECT coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) AS user_src,
			sum($browse_time) AS browsetime
			FROM $log
			WHERE $filter
			AND logid_to_int(logid) NOT IN (4,
			7,
			14)
			GROUP BY user_src HAVING sum($browse_time)>0
			ORDER BY browsetime DESC

Estimated-Browsing-Time-Enhanced	Estimated browsing time enhanced	Traffic	SELECT coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) AS user_src,
			sum($browse_time2) AS browsetime
			FROM $log
			WHERE $filter
			AND logid_to_int(logid) NOT IN (4,
			7,
			14)
			GROUP BY user_src HAVING sum($browse_time2)>0
			ORDER BY browsetime DESC

event-Admin-Failed-Login-Summary	Event admin failed login summary	event	SELECT `user` AS f_user,
			ui,
			count(status) AS total_failed
			FROM $log
			WHERE $filter
			AND nullifna(`user`) IS NOT NULL
			AND logid_to_int(logid) = 32002
			GROUP BY ui,
			f_user
			ORDER BY total_failed DESC

event-Admin-Login-Summary	Event admin login summary	event	SELECT `user` AS f_user, ui, sum(CASE WHEN logid_to_int(logid)=32001 THEN 1 ELSE 0 END) AS total_num,

Dataset Name	Description	Log Category	Query Syntax
			sum(CASE WHEN logid_to_int(logid)=32003 THEN duration ELSE 0 END) AS total_duration, count(STATE) AS total_change FROM $log WHERE $filter AND nullifna(`user`) IS NOT NULL AND logid_to_int(logid) IN (32001, 32003) GROUP BY f_user , ui HAVING sum(CASE WHEN logid_to_int(logid)=32001 THEN 1 ELSE 0 END)>0 ORDER BY total_num DESC
event-Admin-Login-Summary-By-Date	Event admin login summary by date	event	SELECT $flex_timescale AS dom, sum(CASE WHEN logid_to_int(logid)=32001 THEN 1 ELSE 0 END) AS total_num, count(STATE) AS total_change FROM $log WHERE $filter AND nullifna(`user`) IS NOT NULL AND logid_to_int(logid) IN (32001, 32003) GROUP BY dom HAVING sum(CASE WHEN logid_to_int(logid) =32001 THEN 1 ELSE 0 END)>0 ORDER BY dom
event-System-Critical-Severity-Events	Event system critical severity events	event	SELECT msg_desc AS msg, 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, COUNT(*) AS COUNT FROM $log WHERE $filter AND subtype='system' GROUP BY msg_desc, severity ORDER BY COUNT DESC)### t WHERE severity='Critical' GROUP BY msg, severity ORDER BY counts DESC
event-System-High-Severity-Events	Event system high severity events	event	SELECT msg_desc AS msg, 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,

Dataset Name	Description	Log Category	Query Syntax

			COUNT(*) AS COUNT
			FROM $log
			WHERE $filter
			AND subtype='system'
			GROUP BY msg_desc,
			severity
			ORDER BY COUNT DESC)### t
			WHERE severity='High'
			GROUP BY msg,
			severity
			ORDER BY counts DESC

event-System-Medium-Severity-Events	Event system medium severity events	event	SELECT msg_desc AS msg, 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-L='warning' THEN 'Medium' WHEN LEVEL='notice' THEN 'Low' ELSE 'Info' END) AS severity, COUNT(*) AS COUNT FROM $log WHERE $filter AND subtype='system' GROUP BY msg_desc, severity ORDER BY COUNT DESC)### t WHERE severity='Medium' GROUP BY msg, severity ORDER BY counts DESC
event-System-Summary-By-Date	Event system summary by date	event	SELECT $flex_timescale AS dom, 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, sum(CASE WHEN LEVEL = 'notice' THEN 1 ELSE 0 END) AS low, sum(CASE WHEN LEVEL = 'information' OR LEVEL = 'debug' THEN 1 ELSE 0 END) AS info FROM $log WHERE $filter AND subtype='system' GROUP BY dom ORDER BY dom
event-System-Summary-By-Severity	Event system summary by severity	event	SELECT (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'

Dataset Name	Description	Log Category	Query Syntax

			END) AS severity,
			count(*) AS total_num
			FROM $log
			WHERE $filter
			AND subtype='system'
			GROUP BY severity
			ORDER BY total_num DESC

event-Top-DHCP-Summary	Event top dhcp summary	event	DROP TABLE IF EXISTS pre_clt_list; DROP TABLE IF EXISTS cur_clt_list; DROP TABLE IF EXISTS allocated_ip; CREATE TEMPORARY TABLE pre_clt_list AS ### (SELECT concat(interface, '.', devid) AS intf, mac FROM $log WHERE $last3day_period $filter AND logid_to_int(logid) = 26001 AND dhcp_msg = 'Ack' GROUP BY interface, devid, mac)###; CREATE TEMPORARY TABLE cur_clt_list AS ### (SELECT concat(interface, '.', devid) AS intf, mac FROM $log WHERE $filter AND logid_to_int(logid) = 26001 AND dhcp_msg = 'Ack' GROUP BY interface, devid, mac)###; CREATE TEMPORARY TABLE allocated_ip AS ### (SELECT t31.intf, percent_of_allocated_ip FROM (SELECT concat(interface, '.', devid) AS intf, CAST((CAST(used AS float)/CAST (total AS float)*100) AS decimal(10,2)) AS percent_of_allocated_ip, itime FROM $log WHERE $filter AND logid_to_int(logid) = 26003 AND total != 0 GROUP BY interface, devid, percent_of_allocated_ip, itime) t31 INNER JOIN (SELECT concat(interface,'.', devid) AS intf,

Dataset Name	Description	Log Category	Query Syntax
			max(itime) AS max_itime FROM $log WHERE $filter AND logid_to_int(logid) = 26003 GROUP BY interface, devid) t32 ON t31.intf = t32.intf AND t31.itime=t32.max_itime)###; SELECT t41.intf AS interface, percent_of_allocated_ip, count(mac) AS new_cli_count FROM (SELECT intf, percent_of_allocated_ip FROM allocated_ip) t41 INNER JOIN (SELECT intf, mac FROM cur_clt_list WHERE (mac NOT IN (SELECT mac FROM pre_clt_list)) GROUP BY intf, mac) t42 ON t41.intf = t42.intf GROUP BY interface, percent_of_allocated_ip ORDER BY interface , percent_of_allocated_ip DESC

event-Usage-CPU	Event usage CPU	event	SELECT hourstamp,
			cast(sum(cpu_usage)/sum(num) AS decimal(6,2)) AS cpu_avg_usage
			FROM ###
			(SELECT $hour_of_day AS hourstamp,
			sum(cpu) AS cpu_usage,
			count(*) AS num
			FROM $log
			WHERE $filter
			AND subtype='system'
			AND action='perf-stats'
			GROUP BY hourstamp)### t
			GROUP BY hourstamp
			ORDER BY hourstamp

event-Usage-CPU-Sessions	Event usage CPU sessions	event	SELECT hourstamp, cast(sum(sess_usage)/sum(num) AS decimal(10,2)) AS sess_avg_usage, cast(sum(cpu_usage)/sum(num) AS decimal(6,2)) AS cpu_avg_usage FROM ### (SELECT $hour_of_day AS hourstamp, sum(cpu) AS cpu_usage, sum(totalsession) AS sess_usage, count(*) AS num FROM $log WHERE $filter AND subtype='system' AND action='perf-stats' GROUP BY hourstamp)### t GROUP BY hourstamp

Dataset Name	Description	Log Category	Query Syntax
			ORDER BY hourstamp

event-Usage-Mem	Event usage memory	event	SELECT hourstamp,
			      cast(sum(mem_usage)/sum(num) AS decimal(6,2)) AS mem_avg_usage
			FROM ###
			  (SELECT $hour_of_day AS hourstamp,
			                sum(mem) AS mem_usage,
			                      count(*) AS num
			   FROM $log
			   WHERE $filter
			     AND subtype='system'
			     AND action='perf-stats'
			   GROUP BY hourstamp)### t
			GROUP BY hourstamp
			ORDER BY hourstamp

event-Usage-Sessions	Event usage sessions	event	SELECT hourstamp,
			      cast(sum(sess_usage)/sum(num) AS decimal(10,2)) AS sess_avg_usage
			FROM ###
			  (SELECT $hour_of_day AS hourstamp,
			                sum(totalsession) AS sess_usage,
			                         count(*) AS num
			   FROM $log
			   WHERE $filter
			     AND subtype='system'
			     AND action='perf-stats'
			   GROUP BY hourstamp)### t
			GROUP BY hourstamp
			ORDER BY hourstamp

event-Wireless-Accepted-Offwire	Event wireless accepted off-wire	event	SELECT 'accepted' AS ap_full_status,
			      devid,
			      vd,
			      ssid,
			      bssid,
			      manuf,
			      channel,
			      radioband,
			      FROM_dtime(max(last_seen)) AS last_seen,
			      detectionmethod,
			      snclosest,
			      'no' AS on_wire
			FROM ###
			  (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

Dataset Name	Description	Log Category	Query Syntax
			AND logid_to_int(logid) IN (43521, 43525) 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
event-Wireless-Accepted-Onwire	Event wireless accepted on-wire	event	SELECT 'accepted' AS ap_full_status, devid, vd, ssid, bssid, manuf, channel, radioband, FROM_dtime(max(last_seen)) AS last_seen, detectionmethod, snclosest, 'yes' AS on_wire FROM ### (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) GROUP BY devid, vd,

Dataset Name	Description	Log Category	Query Syntax
			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
event-Wireless-Client-Details	Event wireless client details	event	DROP TABLE IF EXISTS ip_list; CREATE TEMPORARY TABLE ip_list AS SELECT ip, lower(mac) AS lmac, sn, ssid, channel, radioband, min(dtime) AS FIRST, max(dtime) AS LAST FROM $log-event WHERE $filter AND ip IS NOT NULL AND mac IS NOT NULL AND sn IS NOT NULL AND ssid IS NOT NULL GROUP BY ip, lmac, sn, ssid, channel, radioband ORDER BY ip; SELECT user_src, ip, lmac, sn, ssid, channel, radioband, FROM_dtime(FIRST) AS first_seen, FROM_dtime(LAST) AS last_seen, cast(volume AS decimal(18,2)) AS bandwidth FROM

Dataset Name	Description	Log Category	Query Syntax

			(SELECT *
			 FROM ip_list
			 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 logid_to_int(logid) NOT IN (4,
			                                       7,
			                                       14)
			       AND srcip IS NOT NULL
			      GROUP BY user_src,
			           srcip HAVING sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0
			      ORDER BY volume DESC)### t
			    GROUP BY user_src,
			         srcip
			    ORDER BY user_src,
			         srcip) t ON ip_list.ip = t.srcip) t
			ORDER BY volume DESC

event-Wireless-Rogue-Offwire	Event wireless rogue off-wire	event	SELECT 'rogue' AS ap_full_status,
			    devid,
			    vd,
			    ssid,
			    bssid,
			    manuf,
			    channel,
			    radioband,
			    FROM_dtime(max(last_seen)) AS last_seen,
			    detectionmethod,
			    snclosest,
			    'no' AS on_wire
			FROM ###
			  (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 IN ('43521',
			          '43525')

Dataset Name	Description	Log Category	Query Syntax
			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
event-Wireless-Rogue-Onwire	Event wireless rogue on-wire	event	SELECT 'rogue' AS ap_full_status, devid, vd, ssid, bssid, manuf, channel, radioband, FROM_dtime(max(last_seen)) AS last_seen, detectionmethod, snclosest, 'yes' AS on_wire FROM ### (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) GROUP BY devid, vd, ssid, bssid,

Dataset Name	Description	Log Category	Query Syntax
			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
event-Wireless-Suppressed-Offwire	Event wireless suppressed offwire	event	SELECT 'suppressed' AS ap_full_status, devid, vd, ssid, bssid, manuf, channel, radioband, FROM_dtime(max(last_seen)) AS last_seen, detectionmethod, snclosest, 'no' AS on_wire FROM ### (SELECT devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest, onwire, logid, apstatus, max(dtime) AS last_seen FROM $log WHERE $filter AND bssid IS NOT NULL AND logid_to_int(logid) IN (43521, 43525) GROUP BY devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod,

Dataset Name	Description	Log Category	Query Syntax
			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
event-Wireless-Suppressed-Onwire	Event wireless suppressed onwire	event	SELECT 'suppressed' AS ap_full_status, devid, vd, ssid, bssid, manuf, channel, radioband, FROM_dtime(max(last_seen)) AS last_seen, detectionmethod, snclosest, 'yes' AS on_wire FROM ### (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) GROUP BY devid, vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest, onwire, apstatus ORDER BY last_seen DESC)### t

Dataset Name	Description	Log Category	Query Syntax

			WHERE apstatus=3
			  AND onwire='yes'
			GROUP BY devid,
			         vd,
			         ssid,
			         bssid,
			         manuf,
			         channel,
			         radioband,
			         detectionmethod,
			         snclosest
			ORDER BY last_seen DESC

event-Wireless-Unclassified-Offwire	Event wireless unclassified off-wire	event	SELECT 'unclassified' AS ap_full_status,
			       devid,
			       vd,
			       ssid,
			       bssid,
			       manuf,
			       channel,
			       radioband,
			       FROM_dtime(max(last_seen)) AS last_seen,
			       detectionmethod,
			       snclosest,
			       'no' AS on_wire
			FROM ###
			  (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)
			   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,

Dataset Name	Description	Log Category	Query Syntax
			vd, ssid, bssid, manuf, channel, radioband, detectionmethod, snclosest ORDER BY last_seen DESC
event-Wireless-Unclassified-Onwire	Event wireless unclassified on-wire	event	SELECT 'unclassified' AS ap_full_status, devid, vd, ssid, bssid, manuf, channel, radioband, FROM_dtime(max(last_seen)) AS last_seen, detectionmethod, snclosest, 'yes' AS on_wire FROM ### (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) 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,

Dataset Name	Description	Log Category	Query Syntax
			radioband, detectionmethod, snclosest ORDER BY last_seen DESC
High-Risk-Application-By-Bandwidth	High risk application by bandwidth usage	Traffic	SELECT t2.name, d_behavior, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth FROM $log t1 INNER JOIN app_mdata t2 ON t1.appid=t2.id WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND d_behavior IS NOT NULL GROUP BY t2.name, d_behavior ORDER BY bandwidth DESC
High-Risk-Application-By-Sessions	High risk application by session count	Traffic	SELECT t2.name, d_behavior, count(*) AS sessions FROM $log t1 INNER JOIN app_mdata t2 ON t1.appid=t2.id WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND d_behavior IS NOT NULL GROUP BY t2.name, d_behavior ORDER BY sessions DESC

number-of-session-timeline	Number of session timeline	Traffic	SELECT $flex_timescale AS hodex,
			count(*) AS sessions
			FROM $log
			WHERE $filter
			AND logid_to_int(logid) NOT IN (4,
			7,
			14)
			GROUP BY hodex
			ORDER BY hodex

os-Detect-OS-Count	Detected operation system count	Traffic	SELECT (coalesce(osname, 'Unknown')) AS os,
			count(*) AS totalnum
			FROM $log
			WHERE $filter
			AND logid_to_int(logid) NOT IN (4,
			7,
			14)
			GROUP BY os
			ORDER BY totalnum DESC

reputation-Number-Of-Incidents-For-All-Users-Devices	Reputation number of incidents for all users devices	Traffic	SELECT $flex_timescale AS hodex, sum(crscore%65536) AS scores, count(*) AS totalnum FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14)

Dataset Name	Description	Log Category	Query Syntax

			AND crscore IS NOT NULL
			GROUP BY hodex HAVING sum(crscore%65536)>0
			ORDER BY hodex

reputation-Score-Summary-For-All-Users-Devices	Reputation score summary for all users devices	Traffic	SELECT $flex_timescale AS hodex,
			sum(crscore%65536) AS scores
			FROM $log
			WHERE $filter
			AND logid_to_int(logid) NOT IN (4,
			7,
			14)
			AND crscore IS NOT NULL
			GROUP BY hodex HAVING sum(crscore%65536)>0
			ORDER BY hodex

reputation-Top-Devices-By-Scores	Reputation top devices by scores	Traffic	SELECT devtype, coalesce(nullifna(`srcname`),nullifna(`srcmac`), ipstr(`srcip`)) AS dev_src, sum(crscore%65536) AS scores FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND crscore IS NOT NULL GROUP BY devtype, dev_src HAVING sum(crscore%65536)>0 ORDER BY scores DESC
reputation-Top-Devices-With-Increased-Scores	Reputation top devices with increased scores	Traffic	DROP TABLE IF EXISTS prd1_dev_tbl; DROP TABLE IF EXISTS prd2_dev_tbl; CREATE TEMPORARY TABLE prd1_dev_tbl AS ### (SELECT coalesce(nullifna(`srcname`),nullifna(`srcmac`), ipstr(`srcip`)) AS f_device, devtype, sum (crscore%65536) AS sum_rp_score FROM $log WHERE $pre_period $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND crscore IS NOT NULL GROUP BY f_device, devtype HAVING sum(crscore%65536)>0 ORDER BY sum_rp_score DESC)###; CREATE TEMPORARY TABLE prd2_dev_tbl AS ### (SELECT coalesce(nullifna(`srcname`),nullifna(`srcmac`), ipstr(`srcip`)) AS f_device, devtype, sum (crscore%65536) AS sum_rp_score FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4,

Dataset Name	Description	Log Category	Query Syntax
			7, 14) AND crscore IS NOT NULL GROUP BY f_device, devtype HAVING sum(crscore%65536)>0 ORDER BY sum_rp_score DESC)###; SELECT t1.f_device, t1.devtype , sum(t1.sum_rp_score) AS t1_sum_score, sum(t2.sum_rp_score) AS t2_sum_score, (sum(t2.sum_rp_score)-sum(t1.sum_rp_score)) AS delta FROM prd1_dev_tbl AS t1 INNER JOIN prd2_dev_tbl AS t2 ON t1.f_device=t2.f_device AND t1.devtype=t2.devtype WHERE t2.sum_rp_score > t1.sum_rp_score GROUP BY t1.f_device, t1.devtype ORDER BY delta DESC

reputation-Top-Users-By-Scores	Reputation top users by scores	Traffic	SELECT coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) AS user_src,
			sum(crscore%65536) AS scores
			FROM $log
			WHERE $filter
			AND logid_to_int(logid) NOT IN (4,
			7,
			14)
			AND crscore IS NOT NULL
			GROUP BY user_src HAVING sum(crscore%65536)>0
			ORDER BY scores DESC

reputation-Top-Users-With-Increased-Scores	Reputation top users with increased scores	Traffic	DROP TABLE IF EXISTS prd1_usr_tbl;
			
			DROP TABLE IF EXISTS prd2_usr_tbl;
			
			CREATE TEMPORARY TABLE prd1_usr_tbl AS ###
			(SELECT coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) AS f_user,
			sum(crscore%65536) AS sum_rp_score
			FROM $log
			WHERE $pre_period $filter
			AND logid_to_int(logid) NOT IN (4,
			7,
			14)
			AND crscore IS NOT NULL
			GROUP BY f_user HAVING sum(crscore%65536)>0
			ORDER BY sum_rp_score DESC)###;
			
			CREATE TEMPORARY TABLE prd2_usr_tbl AS ###
			(SELECT coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) AS f_user,
			sum(crscore%65536) AS sum_rp_score
			FROM $log
			WHERE $filter
			AND logid_to_int(logid) NOT IN (4,

Dataset Name	Description	Log Category	Query Syntax

			7,
			14)
			AND crscore IS NOT NULL
			GROUP BY f_user HAVING sum(crscore%65536)>0
			ORDER BY sum_rp_score DESC)###;
			
			SELECT t1.f_user,
			sum(t1.sum_rp_score) AS t1_sum_score,
			sum(t2.sum_rp_score) AS t2_sum_score,
			(sum(t2.sum_rp_score)-sum(t1.sum_rp_score)) AS delta
			FROM prd1_usr_tbl AS t1
			INNER JOIN prd2_usr_tbl 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

threat-Adware-Timeline	Threat adware timeline	virus	SELECT $flex_timescale AS hodex,
			count(*) AS totalnum
			FROM $log
			WHERE $filter
			AND virus LIKE 'Adware%'
			GROUP BY hodex
			ORDER BY hodex DESC

threat-Attacks-By-Severity	Threat attacks by severity	attack	SELECT (CASE
			WHEN severity='critical' THEN 'Critical'
			WHEN severity='high' THEN 'High'
			WHEN severity='medium' THEN 'Medium'
			WHEN severity='low' THEN 'Low'
			WHEN severity='info' THEN 'Info'
			END) AS severity,
			count(*) AS totalnum
			FROM $log
			WHERE $filter
			GROUP BY severity
			ORDER BY totalnum DESC

threat-Attacks-Over-HTTP-HTTPs	Threat attacks over HTTP HTTPs	attack	SELECT attack, (CASE WHEN severity='critical' THEN 'Critical' WHEN severity='high' THEN 'High' WHEN severity='medium' THEN 'Medium' WHEN severity='low' THEN 'Low' WHEN severity='info' THEN 'Info' END) AS severity, count(*) AS totalnum, (CASE WHEN severity='critical' THEN 0 WHEN severity='high' THEN 1 WHEN severity='medium' THEN 2 WHEN severity='low' THEN 3 WHEN severity='info' THEN 4 ELSE 5 END) AS severity_number FROM $log WHERE $filter AND severity IN ('critical', 'high', 'medium') AND upper(service) IN ('HTTP', 'HTTPS')

Dataset Name	Description	Log Category	Query Syntax
			GROUP BY attack, severity, severity_number ORDER BY severity_number, totalnum DESC

threat-Critical-Severity-Intrusions	Threat critical severity intrusions	attack	SELECT attack,
			vuln_type,
			count(*) AS totalnum
			FROM $log t1
			LEFT JOIN ips_mdata t2 ON t1.attack=t2.name
			WHERE $filter
			AND t1.severity = 'critical'
			GROUP BY attack,
			vuln_type
			ORDER BY totalnum DESC

threat-High-Severity-Intrusions	Threat high severity intrusions	attack	SELECT attack,
			vuln_type,
			count(*) AS totalnum
			FROM $log t1
			LEFT JOIN ips_mdata t2 ON t1.attack=t2.name
			WHERE $filter
			AND t1.severity='high'
			GROUP BY attack,
			vuln_type
			ORDER BY totalnum DESC

threat-Intrusions-Timeline-By-Severity	Threat intrusions timeline by severity	attack	SELECT $flex_timescale AS timescale, (CASE WHEN severity='critical' THEN 'Critical' WHEN severity='high' THEN 'High' WHEN severity='medium' THEN 'Medium' WHEN severity='low' THEN 'Low' WHEN severity='info' THEN 'Info' END) AS severity, count(*) AS totalnum FROM $log WHERE $filter GROUP BY timescale, severity ORDER BY timescale

threat-Intrusion-Timeline	Threat intrusion timeline	attack	SELECT $flex_timescale AS hodex,
			count(*) AS totalnum
			FROM $log
			WHERE $filter
			GROUP BY hodex
			ORDER BY hodex

threat-Low-Severity-Intrusions	Threat low severity intrusions	attack	SELECT attack,
			vuln_type,
			count(*) AS totalnum
			FROM $log t1
			LEFT JOIN ips_mdata t2 ON t1.attack=t2.name
			WHERE $filter
			AND t1.severity='low'
			GROUP BY attack,
			vuln_type
			ORDER BY totalnum DESC

threat-	Threat	attack	SELECT attack, vuln_type,

Dataset Name	Description	Log Category	Query Syntax

Medium-Severity-Intrusions	medium severity intrusions		count(*) AS totalnum
			FROM $log t1
			LEFT JOIN ips_mdata t2 ON t1.attack=t2.name
			WHERE $filter
			AND t1.severity='medium'
			GROUP BY attack,
			vuln_type
			ORDER BY totalnum DESC

threat-Spyware-Timeline	Threat spyware timeline	virus	SELECT $flex_timescale AS hodex,
			count(*) AS totalnum
			FROM $log
			WHERE $filter
			AND virus LIKE 'Riskware%'
			GROUP BY hodex
			ORDER BY hodex DESC

threat-Top-Adware-by-Name	Threat top adware by name	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
			GROUP BY user_src,
			virus
			ORDER BY totalnum DESC)### t
			WHERE virus LIKE 'Adware%'
			GROUP BY virus
			ORDER BY totalnum DESC

threat-Top-Adware-Source	Threat top adware source	Traffic	SELECT srcip, hostname, count(*) AS totalnum FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND virus LIKE 'Adware%' GROUP BY srcip, hostname ORDER BY totalnum DESC

threat-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
			ORDER BY totalnum DESC)### t
			WHERE virus LIKE 'Adware%'
			GROUP BY user_src
			ORDER BY totalnum DESC

threat-Top-Attacks-	Threat top attacks	attack	SELECT attack, count(*) AS attack_count

Dataset Name	Description	Log Category	Query Syntax

Blocked	blocked		FROM $log
			WHERE $filter
			  AND nullifna(attack) IS NOT NULL
			  AND action IN ('deny',
			               'blocked',
			               'reset',
			               'dropped')
			GROUP BY attack
			ORDER BY attack_count DESC

threat-Top-Attacks-Detected	Threat top attacks detected	attack	SELECT attack,
			       severity,
			       sum(attack_count) AS attack_count
			FROM ###
			  (SELECT attack, severity, (CASE WHEN severity = 'critical' THEN 1 WHEN severity = 'high' THEN 2 WHEN severity = 'medium' THEN 3 WHEN severity = 'low' THEN 4 ELSE 5 END) AS severity_level,
			
			count(*) AS attack_count
			   FROM $log
			   WHERE $filter
			     AND nullifna(attack) IS NOT NULL
			   GROUP BY attack,
			            severity,
			            severity_level
			   ORDER BY severity_level,
			            attack_count DESC)### t
			GROUP BY attack,
			         severity,
			         severity_level
			ORDER BY severity_level,
			         attack_count DESC

threat-Top-Blocked-Intrusions	Threat top blocked intrusions	attack	SELECT attack,
			       (CASE
			            WHEN t1.severity='critical' THEN 'Critical'
			            WHEN t1.severity='high' THEN 'High'
			            WHEN t1.severity='medium' THEN 'Medium'
			            WHEN t1.severity='low' THEN 'Low'
			            WHEN t1.severity='info' THEN 'Info'
			        END) AS severity_name,
			       count(*) AS totalnum,
			       vuln_type,
			       (CASE
			            WHEN t1.severity='critical' THEN 0
			            WHEN t1.severity='high' THEN 1
			            WHEN t1.severity='medium' THEN 2
			            WHEN t1.severity='low' THEN 3
			            WHEN t1.severity='info' THEN 4
			            ELSE 5
			        END) AS severity_number
			FROM $log t1
			LEFT JOIN ips_mdata t2 ON t1.attack=t2.name
			WHERE $filter
			  AND nullifna(attack) IS NOT NULL
			  AND action IN ('deny',
			               'blocked',
			               'reset',
			               'dropped')

Dataset Name	Description	Log Category	Query Syntax
			GROUP BY attack, t1.severity, vuln_type ORDER BY severity_number, totalnum DESC

threat-Top-Intrusions-By-Types	Threat top intrusions by types	attack	SELECT vuln_type,
			count(*) AS totalnum
			FROM $log t1
			LEFT JOIN ips_mdata t2 ON t1.attack=t2.name
			WHERE $filter
			AND vuln_type IS NOT NULL
			GROUP BY vuln_type
			ORDER BY totalnum DESC

threat-Top-Intrusion-Sources	Threat top intrusion sources	attack	SELECT SOURCE, sum(cri_num) AS critical, sum(high_num) AS high, sum(med_num) AS medium, sum(cri_num + high_num + med_num) AS totalnum FROM ### (SELECT srcip AS SOURCE, sum(CASE WHEN severity='critical' THEN 1 ELSE 0 END) AS cri_num, sum(CASE WHEN severity='high' THEN 1 ELSE 0 END) AS high_num, sum(CASE WHEN severity='medium' THEN 1 ELSE 0 END) AS med_num FROM $log WHERE $filter AND severity IN ('critical', 'high', 'medium') GROUP BY SOURCE)### t GROUP BY SOURCE ORDER BY totalnum DESC
threat-Top-Intrusion-Victims	Threat top intrusion victims	attack	SELECT victim, sum(cri_num) AS critical, sum(high_num) AS high, sum(med_num) AS medium, sum(cri_num + high_num + med_num) AS totalnum FROM ### (SELECT dstip AS victim, sum((CASE WHEN severity='critical' THEN 1 ELSE 0 END)) AS cri_num, sum(CASE WHEN severity='high' THEN 1 ELSE 0 END) AS high_num, sum(CASE WHEN severity='medium' THEN 1 ELSE 0 END) AS med_num FROM $log WHERE $filter AND severity IN ('critical', 'high', 'medium') GROUP BY victim)### t GROUP BY victim ORDER BY totalnum DESC

Dataset Name	Description	Log Category	Query Syntax
threat-Top-Monitored-Intrusions	Threat top monitored intrusions	attack	SELECT attack, (CASE WHEN t1.severity='critical' THEN 'Critical' WHEN t1.severity='high' THEN 'High' WHEN t1.severity='medium' THEN 'Medium' WHEN t1.severity='low' THEN 'Low' WHEN t1.severity='info' THEN 'Info' END) AS severity_name, count(*) AS totalnum, vuln_type, (CASE WHEN t1.severity='critical' THEN 0 WHEN t1.severity='high' THEN 1 WHEN t1.severity='medium' THEN 2 WHEN t1.severity='low' THEN 3 WHEN t1.severity='info' THEN 4 ELSE 5 END) AS severity_number FROM $log t1 LEFT JOIN ips_mdata t2 ON t1.attack=t2.name WHERE $filter AND nullifna(attack) IS NOT NULL AND action NOT IN ('deny', 'blocked', 'reset', 'dropped') GROUP BY attack, t1.severity, vuln_type ORDER BY severity_number, totalnum DESC

threat-Top-Spyware-by-Name	Threat top spyware by name	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
			GROUP BY user_src,
			virus
			ORDER BY totalnum DESC)### t
			WHERE virus LIKE 'Riskware%'
			GROUP BY virus
			ORDER BY totalnum DESC

threat-Top-Spyware-Source	Threat top spyware source	Traffic	SELECT srcip, hostname, count(*) AS totalnum FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND virus LIKE 'Riskware%' GROUP BY srcip, hostname ORDER BY totalnum DESC
threat-Top-	Threat top	virus	SELECT user_src,

Dataset Name	Description	Log Category	Query Syntax

Spyware-Victims	spyware victims		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
			   ORDER BY totalnum DESC)### t
			WHERE virus LIKE 'Riskware%'
			GROUP BY user_src
			ORDER BY totalnum DESC

threat-Top-Virus-Source	Threat top virus source	Traffic	SELECT srcip,
			      hostname,
			      sum(totalnum) AS totalnum
			FROM (###
			       (SELECT srcip,
			              hostname,
			              count(*) AS totalnum
			        FROM $log-traffic
			        WHERE $filter
			          AND logid_to_int(logid) NOT IN (4,
			                                          7,
			                                          14)
			          AND utmevent IS NOT NULL
			          AND virus IS NOT NULL
			        GROUP BY srcip,
			                 hostname
			        ORDER BY totalnum DESC)###
			      UNION ALL ###
			       (SELECT srcip ,
			              ipstr(\`dstip\`) AS hostname,
			                                count(*) AS totalnum
			        FROM $log-virus
			        WHERE $filter
			          AND (eventtype IS NULL
			              OR logver = 52)
			          AND nullifna(virus) IS NOT NULL
			        GROUP BY srcip,
			                 hostname
			        ORDER BY totalnum DESC)###) t
			GROUP BY srcip,
			        hostname
			ORDER BY totalnum DESC

threat-Virus-Timeline	Threat virus timeline	virus	SELECT hodex,
			      sum(totalnum) AS totalnum
			FROM (###
			       (SELECT $flex_timescale AS hodex,
			                                 count(*) AS totalnum
			        FROM $log-traffic
			        WHERE $filter
			          AND logid_to_int(logid) NOT IN (4,
			                                          7,
			                                          14)
			          AND utmevent IS NOT NULL
			          AND virus IS NOT NULL
			        GROUP BY hodex

Dataset Name	Description	Log Category	Query Syntax

			ORDER BY hodex DESC)###
			UNION ALL ###
			(SELECT $flex_timescale AS hodex,
			count(*) AS totalnum
			FROM $log-virus
			WHERE $filter
			AND (eventtype IS NULL
			OR logver = 52)
			AND nullifna(virus) IS NOT NULL
			GROUP BY hodex
			ORDER BY hodex DESC)###) t
			GROUP BY hodex
			ORDER BY hodex DESC

Top-App-By-Bandwidth	Top applications by bandwidth usage	Traffic	SELECT app_group_name(app) AS app_group, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth, sum(coalesce(rcvdbyte, 0)) AS traffic_in, sum(coalesce(sentbyte, 0)) AS traffic_out, count(*) AS sessions FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND nullifna(app) IS NOT NULL GROUP BY app_group HAVING sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 ORDER BY bandwidth DESC

Top-App-By-Sessions	Top applications by session count	Traffic	SELECT app_group_name(app) AS app_group,
			count(*) AS sessions
			FROM $log
			WHERE $filter
			AND logid_to_int(logid) NOT IN (4,
			7,
			14)
			AND nullifna(app) IS NOT NULL
			GROUP BY app_group
			ORDER BY sessions DESC

Top-Destinations-By-Bandwidth	Top destinations by bandwidth usage	Traffic	SELECT coalesce(nullifna(root_domain(hostname)), ipstr(dstip)) AS DOMAIN, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth, sum(coalesce(rcvdbyte, 0)) AS traffic_in, sum(coalesce(sentbyte, 0)) AS traffic_out FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) 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
Top-Destinations-By-Sessions	Top destinations by session	Traffic	SELECT coalesce(nullifna(root_domain(hostname)), ipstr(dstip)) AS DOMAIN, count(*) AS sessions

Dataset Name	Description	Log Category	Query Syntax

	count		FROM $log
			WHERE $filter
			AND logid_to_int(logid) NOT IN (4,
			7,
			14)
			GROUP BY DOMAIN
			ORDER BY sessions DESC

Top-P2P-App-By-Bandwidth	Top P2P applications by bandwidth usage	Traffic	SELECT app_group_name(app) AS app_group, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth, sum(coalesce(rcvdbyte, 0)) AS traffic_in, sum(coalesce(sentbyte, 0)) AS traffic_out, count(*) AS sessions FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND nullifna(app) IS NOT NULL AND lower(appcat)='p2p' AND action='accept' GROUP BY app_group HAVING sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 ORDER BY bandwidth DESC
Top-P2P-App-By-Sessions	Top P2P applications by session count	Traffic	SELECT app_group_name(app) AS app_group, count(*) AS sessions FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND nullifna(app) IS NOT NULL AND lower(appcat)='p2p' AND action='accept' GROUP BY app_group ORDER BY sessions DESC

Top-User-By-Sessions	Top user by session count	Traffic	SELECT coalesce(nullifna(\`user\`), nullifna(\`unauthuser\`), ipstr(\`srcip\`)) AS user_src,
			count(*) AS sessions
			FROM $log
			WHERE $filter
			AND logid_to_int(logid) NOT IN (4,
			7,
			14)
			GROUP BY user_src
			ORDER BY sessions DESC

Top-Users-By-Bandwidth	Top users by bandwidth usage	Traffic	SELECT coalesce(nullifna(\`user\`), nullifna(\`unauthuser\`), ipstr(\`srcip\`)) AS user_src, srcip, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth, sum(coalesce(rcvdbyte, 0)) AS traffic_in, sum(coalesce(sentbyte, 0)) AS traffic_out FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7,

Dataset Name	Description	Log Category	Query Syntax
			14) AND srcip IS NOT NULL GROUP BY user_src, srcip HAVING sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 ORDER BY bandwidth DESC

Top-User-Source-By-Sessions	Top user source by session count	Traffic	SELECT coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) AS user_src,
			count(*) AS sessions
			FROM $log
			WHERE $filter
			AND logid_to_int(logid) NOT IN (4,
			7,
			14)
			GROUP BY user_src
			ORDER BY sessions DESC

Top-Web-Category-by-Bandwidth	Top web category by bandwidth usage	webfilter	SELECT catdesc, sum(bandwidth) AS bandwidth FROM (### (SELECT catdesc, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth FROM $log-traffic WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND utmevent IN ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') GROUP BY catdesc HAVING sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 ORDER BY bandwidth DESC)### UNION ALL ### (SELECT catdesc, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth FROM $log-webfilter WHERE $filter AND (eventtype IS NULL OR logver = 52) GROUP BY catdesc HAVING sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 ORDER BY bandwidth DESC)###) t GROUP BY catdesc ORDER BY bandwidth DESC
Top-Web-Category-by-Sessions	Top web category by session count	webfilter	SELECT catdesc, sum(sessions) AS sessions FROM (### (SELECT catdesc, count(*) AS sessions FROM $log-traffic WHERE $filter AND logid_to_int(logid) NOT IN (4, 7,

Dataset Name	Description	Log Category	Query Syntax
			14) AND utmevent IN ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') GROUP BY catdesc ORDER BY sessions DESC)### UNION ALL ### (SELECT catdesc, count(*) AS sessions FROM $log-webfilter WHERE $filter AND (eventtype IS NULL OR logver = 52) GROUP BY catdesc ORDER BY sessions DESC)###) t GROUP BY catdesc ORDER BY sessions DESC
Top-Web-Sites-by-Bandwidth	Top web sites by bandwidth usage	webfilter	SELECT DOMAIN, sum(bandwidth) AS bandwidth FROM (### (SELECT coalesce(nullifna(hostname), ipstr(`srcip`)) AS DOMAIN, sum(coalesce(sentbyte, 0) +coalesce(rcvdbyte, 0)) AS bandwidth FROM $log-traffic WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND 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)### UNION ALL ### (SELECT coalesce(nullifna(hostname), ipstr(`srcip`)) AS DOMAIN, sum(coalesce(sentbyte, 0) +coalesce(rcvdbyte, 0)) AS bandwidth FROM $log-webfilter WHERE $filter AND (eventtype IS NULL OR logver = 52) GROUP BY DOMAIN HAVING sum(coalesce(sentbyte, 0) +coalesce(rcvdbyte, 0))>0 ORDER BY bandwidth DESC)###) t GROUP BY DOMAIN ORDER BY bandwidth DESC

Top-Web-Sites-by-Sessions	Top web sites by session count	webfilter	SELECT DOMAIN,
			sum(sessions) AS sessions
			FROM (###
			(SELECT coalesce(nullifna(hostname), ipstr(`srcip`)) AS DOMAIN,

Dataset Name	Description	Log Category	Query Syntax
			count(*) AS sessions FROM $log-traffic WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND utmevent IN ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') GROUP BY DOMAIN ORDER BY sessions DESC)### UNION ALL ### (SELECT coalesce(nullifna(hostname), ipstr(`srcip`)) AS DOMAIN, count(*) AS sessions FROM $log-webfilter WHERE $filter AND (eventtype IS NULL OR logver = 52) GROUP BY DOMAIN ORDER BY sessions DESC)###) t GROUP BY DOMAIN ORDER BY sessions DESC

Total-Attack-Source	Total attack source	attack	SELECT count(*) AS totalnum
			FROM $log
			WHERE $filter

Total-Number-of-Botnet-Events	Total number of botnet events	Traffic	SELECT count(*) AS events FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND appcat='Botnet' AND nullifna(app) IS NOT NULL
Total-Number-of-Viruses	Total number of viruses	Traffic	SELECT sum(totalnum) AS totalnum FROM (### (SELECT count(*) AS totalnum FROM $log-traffic WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND utmevent IS NOT NULL AND virus IS NOT NULL ORDER BY totalnum DESC)### UNION ALL ### (SELECT count(*) AS totalnum FROM $log-virus WHERE $filter AND (eventtype IS NULL OR logver = 52) AND nullifna(virus) IS NOT NULL ORDER BY totalnum DESC)###) t
Traffic-band-	Traffic band-	Traffic	SELECT $flex_timescale AS hodex,

Dataset Name	Description	Log Category	Query Syntax
width-timeline	width timeline		sum(coalesce(sentbyte, 0)) AS traffic_out, sum(coalesce(rcvdbyte, 0)) AS traffic_in FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) GROUP BY hodex HAVING sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 ORDER BY hodex

Traffic-Browsing-Time-Summary	Traffic browsing time summary	Traffic	SELECT hodex,
			cast(sum(delta)/60.0 AS decimal(18, 2)) AS browsetime
			FROM ###
			(SELECT $flex_timescale AS hodex,
			sum($browse_time) AS delta
			FROM $log
			WHERE $filter
			AND logid_to_int(logid) NOT IN (4,
			7,
			14)
			GROUP BY hodex HAVING sum($browse_time)>0
			ORDER BY delta DESC)### t
			GROUP BY hodex
			ORDER BY hodex

Traffic-Browsing-Time-Summary-Enhanced	Traffic browsing time summary enhanced	Traffic	SELECT hodex,
			cast(sum(delta)/60.0 AS decimal(18, 2)) AS browsetime
			FROM ###
			(SELECT $flex_timescale AS hodex,
			sum($browse_time2) AS delta
			FROM $log
			WHERE $filter
			AND logid_to_int(logid) NOT IN (4,
			7,
			14)
			GROUP BY hodex HAVING sum($browse_time2)>0
			ORDER BY delta DESC)### t
			GROUP BY hodex
			ORDER BY hodex

Traffic-History-By-Active-User	Traffic history by active user	Traffic	SELECT hodex,
			count(distinct(user_src)) AS total_user
			FROM ###
			(SELECT $flex_timescale AS hodex,
			coalesce(nullifna(\`user\`), nullifna(\`unauthuser\`), ipstr(\`srcip\`)) AS user_src
			FROM $log
			WHERE $filter
			AND logid_to_int(logid) NOT IN (4,
			7,
			14)
			GROUP BY hodex,
			user_src
			ORDER BY hodex)### t
			GROUP BY hodex
			ORDER BY hodex

Traffic-Top-Category-By-Browsing-	Traffic top category by browsing	Traffic	SELECT catdesc, sum(delta) AS browsetime, sum(bandwidth) AS bandwidth

Dataset Name	Description	Log Category	Query Syntax

Time	time		FROM ###
			(SELECT catdesc,
			sum($browse_time) AS delta,
			sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth
			FROM $log
			WHERE $filter
			AND logid_to_int(logid) NOT IN (4,
			7,
			14)
			AND catdesc IS NOT NULL
			GROUP BY catdesc HAVING sum($browse_time)>0
			ORDER BY delta DESC)### t
			GROUP BY catdesc
			ORDER BY browsetime DESC

Traffic-Top-Category-By-Browsing-Time-Enhanced	Traffic top category by browsing time enhanced	Traffic	SELECT catdesc, sum(delta) AS browsetime, sum(bandwidth) AS bandwidth FROM ### (SELECT catdesc, sum($browse_time2) AS delta, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND catdesc IS NOT NULL GROUP BY catdesc HAVING sum($browse_time2)>0 ORDER BY delta DESC)### t GROUP BY catdesc ORDER BY browsetime DESC
Traffic-Top-Destination-Countries-By-Browsing-Time	Traffic top destination countries by browsing time	Traffic	SELECT dstcountry, sum(delta) AS browsetime, sum(bandwidth) AS bandwidth, sum(traffic_in) AS traffic_in, sum(traffic_out) AS traffic_out FROM ### (SELECT dstcountry, sum($browse_time) AS delta, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth, sum(coalesce (rcvdbyte, 0)) AS traffic_in, sum(coalesce(sentbyte, 0)) AS traffic_out FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) GROUP BY dstcountry HAVING sum($browse_time)>0 ORDER BY delta DESC)### t GROUP BY dstcountry ORDER BY browsetime DESC
Traffic-Top-	Traffic top	Traffic	SELECT dstcountry,

Dataset Name	Description	Log Category	Query Syntax
Destination-Countries-By-Browsing-Time-Enhanced	destination countries by browsing time enhanced		sum(delta) AS browsetime, sum(bandwidth) AS bandwidth, sum(traffic_in) AS traffic_in, sum(traffic_out) AS traffic_out FROM ### (SELECT dstcountry, sum($browse_time2) AS delta, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth, sum(coalesce (rcvdbyte, 0)) AS traffic_in, sum(coalesce(sentbyte, 0)) AS traffic_out FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) GROUP BY dstcountry HAVING sum($browse_time2)>0 ORDER BY delta DESC)### t GROUP BY dstcountry ORDER BY browsetime DESC
Traffic-Top-Domains-By-Browsing-Time	Traffic top domains by browsing time	Traffic	SELECT hostname, sum($browse_time) AS browsetime, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth, sum(coalesce(rcvdbyte, 0)) AS traffic_in, sum(coalesce(sentbyte, 0)) AS traffic_out FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND hostname IS NOT NULL GROUP BY hostname HAVING sum($browse_time)>0 ORDER BY browsetime DESC
Traffic-Top-Domains-By-Browsing-Time-Enhanced	Traffic top domains by browsing time enhanced	Traffic	SELECT hostname, sum($browse_time2) AS browsetime, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth, sum(coalesce(rcvdbyte, 0)) AS traffic_in, sum(coalesce(sentbyte, 0)) AS traffic_out FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND hostname IS NOT NULL GROUP BY hostname HAVING sum($browse_time2)>0 ORDER BY browsetime DESC
Traffic-Top-Sites-By-Browsing-Time	Traffic top sites by browsing time	Traffic	SELECT hostname, string_agg(DISTINCT catdesc, ', ') AS agg_catdesc, sum(delta) AS browsetime, sum(bandwidth) AS bandwidth, sum(traffic_in) AS traffic_in, sum(traffic_out) AS traffic_out FROM ###

Dataset Name	Description	Log Category	Query Syntax
			(SELECT hostname, catdesc, sum($browse_time) AS delta, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth, sum(coalesce (rcvdbyte, 0)) AS traffic_in, sum(coalesce(sentbyte, 0)) AS traffic_out FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND hostname IS NOT NULL GROUP BY hostname, catdesc HAVING sum($browse_time)>0 ORDER BY delta DESC)### t GROUP BY hostname ORDER BY browsetime DESC
Traffic-Top-Sites-By-Browsing-Time-Enhanced	Traffic top sites by browsing time enhanced	Traffic	SELECT hostname, string_agg(DISTINCT catdesc, ', ') AS agg_catdesc, sum(delta) AS browsetime, sum(bandwidth) AS bandwidth, sum(traffic_in) AS traffic_in, sum(traffic_out) AS traffic_out FROM ### (SELECT hostname, catdesc, sum($browse_time2) AS delta, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth, sum(coalesce (rcvdbyte, 0)) AS traffic_in, sum(coalesce(sentbyte, 0)) AS traffic_out FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND hostname IS NOT NULL GROUP BY hostname, catdesc HAVING sum($browse_time2)>0 ORDER BY delta DESC)### t GROUP BY hostname ORDER BY browsetime DESC
Traffic-Top-Users-By-Bandwidth	Traffic top users by bandwidth usage	Traffic	SELECT coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr (`srcip`)) AS user_src, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) GROUP BY user_src HAVING sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0

<table>
<tr><th>Dataset Name</th><th>Description</th><th>Log Category</th><th>Query Syntax</th></tr>
<tr><td></td><td></td><td></td><td>ORDER BY bandwidth DESC</td></tr>
<tr><td>Traffic-Top-Web-Users-By-Browsing-Time</td><td>Traffic top web users by browsing time</td><td>Traffic</td><td>SELECT user_src,
sum(delta) AS browsetime,
sum(bandwidth) AS bandwidth,
sum(traffic_in) AS traffic_in,
sum(traffic_out) AS traffic_out
FROM ###
(SELECT coalesce(nullifna(`user`), ipstr(`srcip`)) AS user_src,
sum($browse_time) AS delta,
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 $filter
GROUP BY user_src HAVING sum($browse_time)>0
ORDER BY delta DESC)### t
GROUP BY user_src
ORDER BY browsetime DESC</td></tr>
<tr><td>Traffic-Top-WiFi-Client-By-Bandwidth</td><td>Traffic top WiFi client by bandwidth usage</td><td>Traffic</td><td>SELECT coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) AS user_src,
srcssid,
devtype,
coalesce(nullifna(`srcname`), `srcmac`) AS hostname_mac,
sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth
FROM $log
WHERE $filter
AND logid_to_int(logid) NOT IN (4,
7,
14)
AND (srcssid IS NOT NULL
OR dstssid IS NOT NULL)
GROUP BY user_src,
srcssid,
devtype,
hostname_mac HAVING sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0
ORDER BY bandwidth DESC</td></tr>
<tr><td>Traffic-User-Detail</td><td>Traffic user detail</td><td>Traffic</td><td>SELECT 'User: ' || string_agg(DISTINCT coalesce(nullifna(`user`), 'Unknown'), '/') || '
' || 'Source IP: ' || string_agg(DISTINCT coalesce(ipstr(srcip), 'Unknown'), '/') || '
' || 'Hostname (MAC): ' || string_agg(DISTINCT coalesce(host_dev, 'Unknown'), '/') || '
' || 'Source Interface: ' || string_agg(DISTINCT coalesce(nullifna(srcintf), 'Unknown'), '/') || '
|| 'Devices: ' || string_agg(distinct coalesce(devid, 'UNKNOWN'), '/') AS user_detail
FROM ###
(SELECT `user`,
srcip,
coalesce(nullifna(`srcname`),nullifna(`srcmac`)) AS host_dev,
srcintf,
devid,</td></tr>
</table>

Dataset Name	Description	Log Category	Query Syntax
			count(*) AS events FROM $log WHERE $filter GROUP BY \`user\`, srcip, host_dev, srcintf, devid ORDER BY events DESC)### t

user-drill-down-Count-Spam-Activity-by-Hour-of-Day	User drill-down count spam activity by hour of day	emailfilter	SELECT hourstamp,
			sum(totalnum) AS totalnum
			FROM ###
			(SELECT coalesce(nullifna(\`user\`), ipstr(\`srcip\`)) AS user_src,
			$hour_of_day AS hourstamp,
			count(*) AS totalnum
			FROM $log
			WHERE $filter-exclude-var
			AND \`to\` IS NOT NULL
			AND action IN ('detected',
			'blocked')
			GROUP BY user_src,
			hourstamp
			ORDER BY hourstamp)### t
			WHERE $filter-var-ONLY
			GROUP BY hourstamp
			ORDER BY hourstamp

user-drill-down-Top-Allowed-Web-Categories	User drill-down top allowed web categories	webfilter	SELECT catdesc,
			sum(requests) AS requests
			FROM ###
			(SELECT coalesce(nullifna(\`user\`), ipstr(\`srcip\`)) AS user_src,
			catdesc,
			action,
			count(*) AS requests
			FROM $log
			WHERE $filter-exclude-var
			AND catdesc IS NOT NULL
			GROUP BY user_src,
			catdesc,
			action
			ORDER BY requests DESC)### t
			WHERE $filter-var-ONLY
			AND action!='blocked'
			GROUP BY catdesc
			ORDER BY requests DESC

user-drill-down-Top-Allowed-Web-Sites-By-Requests	User drill-down top allowed web sites by requests	webfilter	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-exclude-var AND hostname IS NOT NULL GROUP BY user_src, hostname, action ORDER BY requests DESC)### t

Dataset Name	Description	Log Category	Query Syntax

			WHERE $filter-var-ONLY
			AND action!='blocked'
			GROUP BY hostname
			ORDER BY requests DESC

user-drill-down-Top-Attacks-By-Name	User drill-down 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-exclude-var
			AND nullifna(attack) IS NOT NULL
			GROUP BY user_src,
			attack,
			high_severity
			ORDER BY attack_count DESC)### t
			WHERE $filter-var-ONLY
			GROUP BY attack
			ORDER BY attack_count DESC

user-drill-down-Top-Attacks-High-Severity	User drill-down 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-exclude-var
			AND nullifna(attack) IS NOT NULL
			GROUP BY user_src,
			attack,
			high_severity
			ORDER BY attack_count DESC)### t
			WHERE $filter-var-ONLY
			AND high_severity=1
			GROUP BY attack
			ORDER BY attack_count DESC

user-drill-down-Top-Blocked-Web-Categories	User drill-down top blocked web categories	webfilter	SELECT catdesc, sum(requests) AS requests FROM ### (SELECT coalesce(nullifna(\`user\`), ipstr(\`srcip\`)) AS user_src, catdesc, action, count(*) AS requests FROM $log WHERE $filter-exclude-var AND catdesc IS NOT NULL GROUP BY user_src, catdesc, action ORDER BY requests DESC)### t

Dataset Name	Description	Log Category	Query Syntax

			WHERE $filter-var-ONLY
			AND action='blocked'
			GROUP BY catdesc
			ORDER BY requests DESC

user-drill-down-Top-Blocked-Web-Sites-By-Requests	User drill-down top blocked web sites by requests	webfilter	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-exclude-var
			AND hostname IS NOT NULL
			GROUP BY user_src,
			hostname,
			action
			ORDER BY requests DESC)### t
			WHERE $filter-var-ONLY
			AND action='blocked'
			GROUP BY hostname
			ORDER BY requests DESC

user-drill-down-Top-Spam-Sources	User drill-down top spam sources	emailfilter	SELECT mf_sender,
			sum(totalnum) AS totalnum
			FROM ###
			(SELECT coalesce(nullifna(`user`), ipstr(`srcip`)) AS user_src,
			`FROM` AS mf_sender,
			count(*) AS totalnum
			FROM $log
			WHERE $filter-exclude-var
			AND `FROM` IS NOT NULL
			AND action IN ('detected',
			'blocked')
			GROUP BY user_src,
			mf_sender
			ORDER BY totalnum DESC)### t
			WHERE $filter-var-ONLY
			GROUP BY mf_sender
			ORDER BY totalnum DESC

user-drill-down-Top-Virus	User drill-down 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-exclude-var
			AND nullifna(virus) IS NOT NULL
			GROUP BY user_src,
			virus
			ORDER BY totalnum DESC)### t
			WHERE $filter-var-ONLY
			GROUP BY virus
			ORDER BY totalnum DESC

user-drill-down-Top-Virus-Receiv-	User drill-down top virus receiv-	virus	SELECT receiver,
			sum(totalnum) AS totalnum
			FROM ###

Dataset Name	Description	Log Category	Query Syntax
ers-Over-Email	ers over email		(SELECT coalesce(nullifna(`user`), ipstr(`srcip`)) AS user_src, `to` AS receiver, count(*) AS totalnum FROM $log WHERE $filter-exclude-var AND subtype='infected' AND (service IN ('smtp', 'SMTP', '25/tcp', '587/tcp', 'smtps', 'SMTPS', '465/tcp') OR service IN ('pop3', 'POP3', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/tcp', 'pop3s', 'POP3S', '995/tcp')) AND nullifna(virus) IS NOT NULL GROUP BY user_src, receiver ORDER BY totalnum DESC)### t WHERE $filter-var-ONLY GROUP BY receiver ORDER BY totalnum DESC
utm-drilldown-Email-Receivers-Summary	UTM drill-down 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-exclude-var AND logid_to_int(logid) NOT IN (4, 7, 14) AND recipient IS NOT NULL AND service IN ('pop3', 'POP3', '110/tcp', 'imap', 'IMAP', '143/tcp', 'imaps', 'IMAPS', '993/tcp', 'pop3s',

Dataset Name	Description	Log Category	Query Syntax

			'POP3S',
			'995/tcp')
			GROUP BY user_src,
			recipient
			ORDER BY requests DESC)### t
			WHERE $filter-var-ONLY

utm-drilldown-Email-Senders-Summary	UTM drill-down email senders summary	Traffic	SELECT sum(requests) AS requests, sum(bandwidth) AS bandwidth FROM ### (SELECT coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) AS user_src, sender, count(*) AS requests, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth FROM $log WHERE $filter-exclude-var AND logid_to_int(logid) NOT IN (4, 7, 14) 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-var-ONLY
utm-drilldown-Top-Allowed-Web-Sites-By-Request	UTM drill-down top allowed web sites by request	Traffic	SELECT appid, hostname, sum(requests) AS requests FROM (### (SELECT coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) AS user_src, appid, hostname, (CASE WHEN utmaction='blocked' THEN 1 ELSE 0 END) AS blocked, count(*) AS requests FROM $log-traffic WHERE $filter-exclude-var AND logid_to_int(logid) NOT IN (4, 7, 14) AND utmevent IN ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') AND hostname IS NOT NULL GROUP BY user_src, appid, hostname, blocked ORDER BY requests DESC)### UNION ALL ###

Dataset Name	Description	Log Category	Query Syntax
			(SELECT coalesce(nullifna(`user`), ipstr(`srcip`)) AS user_src, 0 AS appid, hostname, (CASE WHEN action='blocked' THEN 1 ELSE 0 END) AS blocked, count(*) AS requests FROM $log-webfilter WHERE $filter-exclude-var AND (eventtype IS NULL OR logver = 52) AND hostname IS NOT NULL GROUP BY user_src, appid, hostname, blocked ORDER BY requests DESC)###) t WHERE $filter-var-ONLY AND blocked=0 GROUP BY appid, hostname ORDER BY requests DESC
utm-drilldown-Top-App-By-Bandwidth	UTM drill-down top applications by bandwidth usage	Traffic	SELECT appid, app, sum(bandwidth) AS bandwidth FROM ### (SELECT coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) AS user_src, appid, app, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth, count(*) AS sessions FROM $log WHERE $filter-exclude-var AND logid_to_int(logid) NOT IN (4, 7, 14) AND nullifna(app) IS NOT NULL GROUP BY user_src, appid, app ORDER BY sessions DESC)### t WHERE $filter-var-ONLY GROUP BY appid, app HAVING sum(bandwidth)>0 ORDER BY bandwidth DESC
utm-drilldown-Top-App-By-Sessions	UTM drill-down top applications by session count	Traffic	SELECT appid, app, sum(sessions) AS sessions FROM ### (SELECT coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) AS user_src, appid, app, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth,

Dataset Name	Description	Log Category	Query Syntax
			count(*) AS sessions FROM $log WHERE $filter-exclude-var AND logid_to_int(logid) NOT IN (4, 7, 14) AND nullifna(app) IS NOT NULL GROUP BY user_src, appid, app ORDER BY sessions DESC)### t WHERE $filter-var-ONLY GROUP BY appid, app ORDER BY sessions DESC

utm-drilldown-Top-Attacks-By-Name	UTM drill-down 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-exclude-var
			AND nullifna(attack) IS NOT NULL
			GROUP BY user_src,
			attack
			ORDER BY attack_count DESC)### t
			WHERE $filter-var-ONLY
			GROUP BY attack
			ORDER BY attack_count DESC

utm-drilldown-Top-Blocked-Web-Sites-By-Request	UTM drill-down top blocked web sites by request	Traffic	SELECT appid, hostname, sum(requests) AS requests FROM (### (SELECT coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) AS user_src, appid, hostname, (CASE WHEN utmaction='blocked' THEN 1 ELSE 0 END) AS blocked, count(*) AS requests FROM $log-traffic WHERE $filter-exclude-var AND logid_to_int(logid) NOT IN (4, 7, 14) AND utmevent IN ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') AND hostname IS NOT NULL GROUP BY user_src, appid, hostname, blocked ORDER BY requests DESC)### UNION ALL ### (SELECT coalesce(nullifna(`user`), ipstr(`srcip`)) AS user_src,

Dataset Name	Description	Log Category	Query Syntax

			0 AS appid, hostname, (CASE WHEN action='blocked' THEN 1 ELSE 0 END) AS blocked,
			
			count(*) AS requests
			        FROM $log-webfilter
			        WHERE $filter-exclude-var
			          AND (eventtype IS NULL
			               OR logver = 52)
			          AND hostname IS NOT NULL
			        GROUP BY user_src,
			                 appid,
			                 hostname,
			                 blocked
			        ORDER BY requests DESC)###) t
			WHERE $filter-var-ONLY
			  AND blocked=1
			GROUP BY appid,
			        hostname
			ORDER BY requests DESC

utm-drilldown-Top-Email-Recipients	UTM drill-down 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-exclude-var
			     AND logid_to_int(logid) NOT IN (4,
			                                     7,
			                                     14)
			     AND service IN ('pop3',
			                     'POP3',
			                     '110/tcp',
			                     'imap',
			                     'IMAP',
			                     '143/tcp',
			                     'imaps',
			                     'IMAPS',
			                     '993/tcp',
			                     'pop3s',
			                     'POP3S',
			                     '995/tcp')
			   GROUP BY user_src,
			            recipient
			   ORDER BY requests DESC)### t
			WHERE $filter-var-ONLY
			  AND recipient IS NOT NULL
			GROUP BY recipient HAVING sum(bandwidth)>0
			ORDER BY bandwidth DESC

utm-drilldown-Top-Email-Senders	UTM drill-down top email senders	Traffic	SELECT sender,
			       sum(bandwidth) AS bandwidth
			FROM ###
			  (SELECT coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) AS user_src,

Dataset Name	Description	Log Category	Query Syntax
			sender, count(*) AS requests, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth FROM $log WHERE $filter-exclude-var AND logid_to_int(logid) NOT IN (4, 7, 14) 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-var-ONLY AND sender IS NOT NULL GROUP BY sender HAVING sum(bandwidth)>0 ORDER BY bandwidth DESC
utm-drilldown-Top-User-Destination	UTM drill-down top user destination	Traffic	SELECT appid, app, dstip, sum(sessions) AS sessions, sum(bandwidth) AS bandwidth FROM ### (SELECT coalesce(nullifna(\`user\`), nullifna(\`unauthuser\`), ipstr(\`srcip\`)) AS user_src, appid, app, dstip, count(*) AS sessions, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth FROM $log WHERE $filter-exclude-var AND logid_to_int(logid) NOT IN (4, 7, 14) 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-var-ONLY GROUP BY appid, app, dstip ORDER BY bandwidth DESC
utm-drilldown-	UTM drill-	Traffic	SELECT coalesce(nullifna(\`user\`), nullifna(\`unauthuser\`), ipstr

Dataset Name	Description	Log Category	Query Syntax
Top-Users-By-Bandwidth	down top users by bandwidth usage		(`srcip`)) AS dldn_user, count(*) AS SESSION, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth, sum(coalesce(sentbyte, 0)) AS traffic_out, sum(coalesce(rcvdbyte, 0)) AS traffic_in FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) GROUP BY dldn_user HAVING sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 ORDER BY bandwidth DESC
utm-drilldown-Top-Virus	UTM drill-down top virus	Traffic	SELECT virus, sum(totalnum) AS totalnum FROM (### (SELECT coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) AS user_src, virus, count(*) AS totalnum FROM $log-traffic WHERE $filter-exclude-var AND logid_to_int(logid) NOT IN (4, 7, 14) AND utmevent IS NOT NULL AND virus IS NOT NULL GROUP BY user_src, virus ORDER BY totalnum DESC)### UNION ALL ### (SELECT coalesce(nullifna(`user`), ipstr(`srcip`)) AS user_src, virus, count(*) AS totalnum FROM $log-virus WHERE $filter-exclude-var AND (eventtype IS NULL OR logver = 52) AND nullifna(virus) IS NOT NULL GROUP BY user_src, virus ORDER BY totalnum DESC)###) t WHERE $filter-var-ONLY GROUP BY virus ORDER BY totalnum DESC
utm-drilldown-Top-Vulnerability-By-Name	UTM drill-down top vulnerability by name	netscan	SELECT vuln, sum(totalnum) AS totalnum FROM ### (SELECT coalesce(nullifna(`user`), ipstr(`srcip`)) AS user_src, vuln, count(*) AS totalnum FROM $log WHERE $filter-exclude-var AND action='vuln-detection' AND vuln IS NOT NULL

Dataset Name	Description	Log Category	Query Syntax

			GROUP BY user_src,
			vuln
			ORDER BY totalnum DESC)### t
			WHERE $filter-var-ONLY
			GROUP BY vuln
			ORDER BY totalnum DESC

utm-drilldown-Traffic-Summary	UTM drill-down traffic summary	Traffic	SELECT srcip, srcname FROM (SELECT * FROM (### (SELECT coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) AS user_src, srcip, srcname FROM $log WHERE $filter-exclude-var AND logid_to_int(logid) NOT IN (4, 7, 14) GROUP BY user_src, srcip, srcname)###) t WHERE $filter-var-ONLY GROUP BY user_src, srcip, srcname) t GROUP BY srcip, srcname
utm-Top-Allowed-Websites-By-Bandwidth	UTM top allowed websites by bandwidth usage	Traffic	SELECT appid, hostname, catdesc, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth, sum(coalesce(rcvdbyte, 0)) AS traffic_in, sum(coalesce(sentbyte, 0)) AS traffic_out FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND utmevent IN ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') AND hostname IS NOT NULL GROUP BY appid, hostname, catdesc HAVING sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 ORDER BY bandwidth DESC

utm-Top-Allowed-Web-Sites-By-Request	UTM top allowed web sites by request	Traffic	SELECT hostname,
			catdesc,
			count(*) AS requests
			FROM $log
			WHERE $filter
			AND logid_to_int(logid) NOT IN (4,

Dataset Name	Description	Log Category	Query Syntax
			7, 14) AND utmevent IN ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') AND hostname IS NOT NULL AND utmaction!='blocked' GROUP BY hostname, catdesc ORDER BY requests DESC

utm-Top-Attack-Dest	UTM top attack dest	attack	SELECT dstip,
			count(*) AS totalnum
			FROM $log
			WHERE $filter
			AND dstip IS NOT NULL
			GROUP BY dstip
			ORDER BY totalnum DESC

utm-Top-Attack-Source	UTM top attack source	attack	SELECT coalesce(nullifna(\`user\`), ipstr(\`srcip\`)) AS user_src,
			count(*) AS totalnum
			FROM $log
			WHERE $filter
			GROUP BY user_src
			ORDER BY totalnum DESC

utm-Top-Blocked-Web-Sites-By-Request	UTM top blocked web sites by request	Traffic	SELECT hostname, count(*) AS requests FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND utmevent IN ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') AND hostname IS NOT NULL AND utmaction='blocked' GROUP BY hostname ORDER BY requests DESC
utm-Top-Blocked-Web-Users	UTM top blocked web users	Traffic	SELECT coalesce(nullifna(\`user\`), nullifna(\`unauthuser\`), ipstr(\`srcip\`)) AS user_src, devtype, srcname, count(*) AS requests FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND utmevent IN ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') AND utmaction='blocked'

Dataset Name	Description	Log Category	Query Syntax

			GROUP BY user_src,
			devtype,
			srcname
			ORDER BY requests DESC

utm-Top-Video-Streaming-Websites-By-Bandwidth	UTM top video streaming websites by bandwidth usage	Traffic	SELECT appid, hostname, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth, sum(coalesce(rcvdbyte, 0)) AS traffic_in, sum(coalesce(sentbyte, 0)) AS traffic_out FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND catdesc IN ('Streaming Media and Download') GROUP BY appid, hostname HAVING sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 ORDER BY bandwidth DESC
utm-Top-Virus	UTM top virus	Traffic	SELECT virus, (CASE WHEN virus LIKE 'Riskware%' THEN 'Spyware' WHEN virus LIKE 'Adware%' THEN 'Adware' ELSE 'Virus' END) AS malware_type, sum(totalnum) AS totalnum FROM (### (SELECT virus, count(*) AS totalnum FROM $log-traffic WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND utmevent IS NOT NULL AND virus IS NOT NULL GROUP BY virus ORDER BY totalnum DESC)### UNION ALL ### (SELECT virus, count(*) AS totalnum FROM $log-virus WHERE $filter AND (eventtype IS NULL OR logver = 52) AND nullifna(virus) IS NOT NULL GROUP BY virus ORDER BY totalnum DESC)###) t GROUP BY virus, malware_type ORDER BY totalnum DESC
utm-Top-Virus-User	UTM top virus user	Traffic	SELECT user_src, sum(totalnum) AS totalnum FROM (### (SELECT coalesce(nullifna(\`user\`), nullifna(\`unauthuser\`), ipstr(\`srcip\`)) AS user_src, count(*) AS total-

Dataset Name	Description	Log Category	Query Syntax
			num FROM $log-traffic WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND utmevent IS NOT NULL AND virus IS NOT NULL GROUP BY user_src ORDER BY totalnum DESC)### UNION ALL ### (SELECT coalesce(nullifna(`user`), ipstr(`srcip`)) AS user_src, count(*) AS totalnum FROM $log-virus WHERE $filter AND (eventtype IS NULL OR logver = 52) AND nullifna(virus) IS NOT NULL GROUP BY user_src ORDER BY totalnum DESC)###) t GROUP BY user_src ORDER BY totalnum DESC
utm-Top-Web-Users-By-Bandwidth	UTM top web users by bandwidth usage	Traffic	SELECT coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) AS user_src, devtype, srcname, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth, sum(coalesce(rcvdbyte, 0)) AS traffic_in, sum(coalesce(sentbyte, 0)) AS traffic_out FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND utmevent IN ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') GROUP BY user_src, devtype, srcname HAVING sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 ORDER BY bandwidth DESC
utm-Top-Web-Users-By-Request	UTM top web users by request	Traffic	SELECT coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) AS user_src, devtype, srcname, count(*) AS requests FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND utmevent IN ('webfilter', 'banned-word',

Dataset Name	Description	Log Category	Query Syntax
			'web-content', 'command-block', 'script-filter') GROUP BY user_src, devtype, srcname ORDER BY requests DESC
vpn-Authenticated-Logins	VPN authenticated logins	event	SELECT f_user, tunneltype, sum(total) AS total_num, sum(dura) AS duration FROM ### (SELECT t1.f_user AS f_user, t1.tunneltype AS tunneltype, t1.total AS total, t2.dura AS dura FROM ((SELECT coalesce(nullifna(`xauthuser`), `user`) AS f_user, tunneltype, count(*) AS total, tunnelid FROM $log WHERE $filter AND subtype='vpn' AND (tunneltype='ipsec' OR tunneltype='ssl-web') AND action='tunnel-up' AND coalesce(nullifna(`xauthuser`), nullifna(`user`)) IS NOT NULL GROUP BY f_user, tunneltype, tunnelid ORDER BY tunnelid) AS t1 INNER JOIN (SELECT tunnelid, sum(dura_end-dura_beg) AS dura FROM (SELECT coalesce(nullifna(`xauthuser`),`user`) AS f_user, tunneltype, min(coalesce(duration, 0)) AS dura_beg, max(coalesce(duration,0)) AS dura_end, tunnelid, 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 FROM $log WHERE $filter AND subtype='vpn' AND tunneltype IN ('ipsec',

Dataset Name	Description	Log Category	Query Syntax
			'ssl-web') AND coalesce(nullifna(`xauthuser`), nullifna(`user`)) IS NOT NULL AND action='tunnel-stats' AND tunnelid IS NOT NULL GROUP BY f_user, tunneltype, tunnelid ORDER BY tunnelid) tt GROUP BY tunnelid HAVING sum(sent_end-sent_beg+rcvd_end-rcvd_beg)>0) AS t2 ON t1.tunnelid=t2.tunnelid))### t GROUP BY f_user, tunneltype ORDER BY total_num DESC
vpn-Failed-Logins	VPN failed logins	event	SELECT f_user, tunneltype, sum(total_num) AS total_num FROM ### (SELECT coalesce(nullifna(`xauthuser`), `user`) AS f_user, tunneltype, count(*) AS total_num FROM $log WHERE $filter AND subtype='vpn' AND (tunneltype='ipsec' OR left(tunneltype, 3)='ssl') AND action IN ('ssl-login-fail', 'ipsec-login-fail') AND coalesce(nullifna(`xauthuser`), nullifna(`user`)) IS NOT NULL GROUP BY f_user, tunneltype)### t GROUP BY f_user, tunneltype ORDER BY total_num DESC
vpn-Top-Dial-Up-IPSEC-Tunnels-By-Bandwidth	Top dial up IPsec tunnels by bandwidth usage	event	SELECT vpn_name, sum(sent_end-sent_beg+rcvd_end-rcvd_beg) AS bandwidth, sum(rcvd_end-rcvd_beg) AS traffic_in, sum(sent_end-sent_beg) AS traffic_out FROM ### (SELECT vpn_trim(vpntunnel) AS vpn_name, tunnelid, 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 FROM $log WHERE $filter AND nullifna(vpntunnel) IS NOT NULL AND subtype='vpn' AND tunneltype LIKE 'ipsec%' AND NOT (tunnelip IS NULL OR (tunnelip='0.0.0.0' AND coalesce(logver, 0)!=52))

Dataset Name	Description	Log Category	Query Syntax

			AND action='tunnel-stats'
			AND tunnelid IS NOT NULL
			GROUP BY vpn_name,
			tunnelid
			ORDER BY tunnelid)### t
			GROUP BY vpn_name HAVING sum(sent_end-sent_beg+rcvd_end-rcvd_beg)>0
			ORDER BY bandwidth DESC

vpn-Top-Dial-Up-IPSEC-Users-By-Bandwidth	Top dial up IPsec users by bandwidth usage	event	SELECT user_src, remip, sum(sent_end-sent_beg+rcvd_end-rcvd_beg) AS bandwidth, sum(rcvd_end-rcvd_beg) AS traffic_in, sum(sent_end-sent_beg) AS traffic_out FROM ### (SELECT coalesce(nullifna(`xauthuser`), nullifna(`user`), ipstr(`remip`)) AS user_src, remip, tunnelid, 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 FROM $log WHERE $filter AND subtype='vpn' AND tunneltype LIKE 'ipsec%' AND NOT (tunnelip IS NULL OR (tunnelip='0.0.0.0' AND coalesce(logver, 0)!=52)) AND action='tunnel-stats' AND tunnelid IS NOT NULL GROUP BY user_src, remip, tunnelid ORDER BY tunnelid)### t GROUP BY user_src, remip HAVING sum(sent_end-sent_beg+rcvd_end-rcvd_beg)>0 ORDER BY bandwidth DESC
vpn-Top-Dialup-IPSEC-Users-By-Bandwidth-and-Avail	Top dialup IPsec users by bandwidth usage and avail	event	SELECT user_src, remip, sum(traffic_out) AS traffic_out, sum(traffic_in) AS traffic_in, sum(bandwidth) AS bandwidth, sum(uptime) AS uptime FROM (SELECT user_src, remip, tunnelid, devid, vd, sum(sent_end-sent_beg) AS traffic_out, sum(rcvd_end-rcvd_beg) AS traffic_in,

Dataset Name	Description	Log Category	Query Syntax
			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 coalesce(logver, 0)!=52)) AND tunnelid IS NOT NULL GROUP BY tunnelid, user_src, remip, devid, vd ORDER BY tunnelid)### t GROUP BY user_src, remip, tunnelid, devid, vd ORDER BY bandwidth DESC) t GROUP BY user_src, remip ORDER BY bandwidth DESC
vpn-Top-Dial-Up-IPSEC-Users-By-Duration	Top dial up IPsec users by duration	event	SELECT user_src, sum(dura_end-dura_beg) AS duration, sum(sent_end-sent_beg+rcvd_end-rcvd_beg) AS bandwidth FROM ### (SELECT coalesce(nullifna(`xauthuser`), nullifna(`user`), ipstr(`remip`)) AS user_src, tunnelid, min(coalesce(duration, 0)) AS dura_beg, max(coalesce(duration,0)) AS dura_end, min(coalesce(sentbyte, 0)) AS sent_beg,

Dataset Name	Description	Log Category	Query Syntax

			max(coalesce(sentbyte, 0)) AS sent_end,
			
			min(coalesce(rcvdbyte, 0)) AS rcvd_beg,
			max(coalesce(rcvdbyte, 0)) AS rcvd_end
			  FROM $log
			WHERE $filter
			AND subtype='vpn'
			AND tunneltype LIKE 'ipsec%'
			AND NOT (tunnelip IS NULL
			    OR (tunnelip='0.0.0.0'
			      AND coalesce(logver, 0)!=52))
			AND action='tunnel-stats'
			AND tunnelid IS NOT NULL
			  GROUP BY user_src,
			      tunnelid
			  ORDER BY tunnelid)### t
			GROUP BY user_src HAVING sum(sent_end-sent_beg+rcvd_end-rcvd_beg)>0
			ORDER BY duration DESC

vpn-Top-Dial-Up-VPN-Users-By-Duration	Top dial up VPN users by duration	event	SELECT user_src,
			    tunneltype,
			    sum(dura_end-dura_beg) AS duration,
			    sum(sent_end-sent_beg+rcvd_end-rcvd_beg) AS bandwidth
			FROM ###
			 (SELECT coalesce(nullifna(`xauthuser`), nullifna(`user`), ipstr(`remip`)) AS user_src,
			                tunneltype,
			                tunnelid,
			                min(coalesce(duration, 0)) AS dura_beg,
			
			max(coalesce(duration,0)) AS dura_end,
			
			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
			  FROM $log
			WHERE $filter
			AND subtype='vpn'
			AND (tunneltype LIKE 'ssl%'
			   OR (tunneltype LIKE 'ipsec%'
			    AND NOT (tunnelip IS NULL
			       OR (tunnelip='0.0.0.0'
			        AND coalesce(logver, 0)!=52))))
			AND action='tunnel-stats'
			AND tunnelid IS NOT NULL
			  GROUP BY user_src,
			      tunneltype,
			      tunnelid
			  ORDER BY tunnelid)### t
			GROUP BY user_src,
			    tunneltype HAVING sum(sent_end-sent_beg+rcvd_end-rcvd_beg)>0

Dataset Name	Description	Log Category	Query Syntax
			ORDER BY duration DESC
vpn-Top-S2S-IPSEC-Tunnels-By-Bandwidth-and-Avail	Top S2S IPsec tunnels by bandwidth usage and avail	event	SELECT vpntunnel, tunneltype, sum(traffic_out) AS traffic_out, sum(traffic_in) AS traffic_in, sum(bandwidth) AS bandwidth, sum(uptime) AS uptime FROM (SELECT vpntunnel, tunneltype, tunnelid, devid, vd, sum(sent_end-sent_beg) AS traffic_out, sum(rcvd_end-rcvd_beg) AS traffic_in, sum(sent_end-sent_beg+rcvd_end-rcvd_beg) AS bandwidth, sum(duration_end-duration_beg) AS uptime FROM ### (SELECT tunnelid, tunneltype, vpntunnel, devid, vd, min(coalesce(sentbyte, 0)) AS sent_beg, max(coalesce(sentbyte, 0)) AS sent_end, min(coalesce(rcvdbyte, 0)) AS rcvd_beg, max(coalesce(rcvdbyte, 0)) AS rcvd_end, min(coalesce(duration, 0)) AS duration_beg, max(coalesce(duration, 0)) AS duration_end FROM $log WHERE $filter AND subtype='vpn' AND action='tunnel-stats' AND tunneltype LIKE 'ipsec%' AND (tunnelip IS NULL OR (tunnelip='0.0.0.0' AND coalesce(logver, 0)!=52)) AND nullifna(`user`) IS NULL AND tunnelid IS NOT NULL GROUP BY tunnelid, tunneltype, vpntunnel, devid, vd ORDER BY tunnelid)### t GROUP BY vpntunnel, tunneltype, tunnelid, devid, vd ORDER BY bandwidth DESC) t GROUP BY vpntunnel,

Dataset Name	Description	Log Category	Query Syntax
			tunneltype ORDER BY bandwidth DESC

vpn-Top-SSL-Tunnel-Users-By-Bandwidth-and-Avail	Top SSL tunnel users by bandwidth usage and avail	event	SELECT user_src,
			    remote_ip,
			    sum(traffic_out) AS traffic_out,
			    sum(traffic_in) AS traffic_in,
			    sum(bandwidth) AS bandwidth,
			    sum(uptime) AS uptime
			FROM
			  (SELECT user_src,
			      remip AS remote_ip,
			      tunnelid,
			      devid,
			      vd,
			      sum(sent_end-sent_beg) AS traffic_out,
			      sum(rcvd_end-rcvd_beg) AS traffic_in,
			      sum(sent_end-sent_beg+rcvd_end-rcvd_beg) AS band-width,
			      sum(duration_end-duration_beg) AS uptime
			   FROM ###
			    (SELECT tunnelid,
			        coalesce(nullifna(\`user\`), ipstr(\`remip\`)) AS user_src,
			                              remip,
			                              devid,
			                              vd,
			                              min(coalesce(sentbyte, 0)) AS sent_beg,
			                                                  max(coalesce(sentbyte, 0)) AS sent_end,
			min(coalesce(rcvdbyte, 0)) AS rcvd_beg,
			max(coalesce(rcvdbyte, 0)) AS rcvd_end,
			min(coalesce(duration, 0)) AS duration_beg,
			max(coalesce(duration, 0)) AS duration_end
			     FROM $log
			WHERE $filter
			AND subtype='vpn'
			AND action='tunnel-stats'
			AND tunneltype IN ('ssl-tunnel',
			            'ssl')
			AND coalesce(nullifna(\`user\`), ipstr(\`remip\`)) IS NOT NULL
			AND tunnelid IS NOT NULL
			     GROUP BY tunnelid,
			          user_src,
			          remip,
			          devid,
			          vd
			     ORDER BY tunnelid)### t
			   GROUP BY user_src,
			      remote_ip,
			      tunnelid,
			      devid,
			      vd
			   ORDER BY bandwidth DESC) t
			GROUP BY user_src,
			   remote_ip

Dataset Name	Description	Log Category	Query Syntax
			ORDER BY bandwidth DESC
vpn-Top-SSL-VPN-Tunnel-Users-By-Bandwidth	Top SSL VPN tunnel users by bandwidth usage	event	SELECT user_src, remote_ip, sum(sent_end-sent_beg+rcvd_end-rcvd_beg) AS bandwidth, sum(rcvd_end-rcvd_beg) AS traffic_in, sum(sent_end-sent_beg) AS traffic_out FROM ### (SELECT coalesce(nullifna(`user`), ipstr(`remip`)) AS user_src, remip AS remote_ip, tunnelid, 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 FROM $log WHERE $filter AND subtype='vpn' AND tunneltype LIKE 'ssl-tunnel' AND action='tunnel-stats' AND coalesce(nullifna(`user`), ipstr(`remip`)) IS NOT NULL GROUP BY tunnelid, user_src, remip ORDER BY tunnelid)### t GROUP BY user_src, remote_ip HAVING sum(sent_end-sent_beg+rcvd_end-rcvd_beg)>0 ORDER BY bandwidth DESC
vpn-Top-SSL-VPN-Users-By-Bandwidth	Top SSL VPN users by bandwidth usage	event	SELECT user_src, remote_ip, sum(sent_end-sent_beg+rcvd_end-rcvd_beg) AS bandwidth, sum(rcvd_end-rcvd_beg) AS traffic_in, sum(sent_end-sent_beg) AS traffic_out FROM ### (SELECT coalesce(nullifna(`user`), ipstr(`remip`)) AS user_src, remip AS remote_ip, tunnelid, 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 FROM $log WHERE $filter AND subtype='vpn' AND tunneltype LIKE 'ssl%' AND action='tunnel-stats' AND coalesce(nullifna(`user`), ipstr(`remip`)) IS NOT NULL

Dataset Name	Description	Log Category	Query Syntax

			AND tunnelid IS NOT NULL
			GROUP BY tunnelid,
			user_src,
			remip
			ORDER BY tunnelid)### t
			GROUP BY user_src,
			remote_ip HAVING sum(sent_end-sent_beg+rcvd_end-rcvd_beg)>0
			ORDER BY bandwidth DESC

vpn-Top-SSL-VPN-Users-By-Duration	Top SSL VPN users by duration	event	SELECT user_src, tunneltype, sum(dura_end-dura_beg) AS duration, sum(sent_end-sent_beg+rcvd_end-rcvd_beg) AS bandwidth FROM ### (SELECT coalesce(nullifna(\`user\`), ipstr(\`remip\`)) AS user_src, tunneltype, min(coalesce(duration, 0)) AS dura_beg, max(coalesce(duration, 0)) AS dura_end, tunnelid, 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 FROM $log WHERE $filter AND subtype='vpn' AND tunneltype LIKE 'ssl%' AND action='tunnel-stats' AND coalesce(nullifna(\`user\`), ipstr(\`remip\`)) IS NOT NULL AND tunnelid IS NOT NULL GROUP BY tunnelid, user_src, tunneltype ORDER BY tunnelid)### t GROUP BY user_src, tunneltype HAVING sum(sent_end-sent_beg+rcvd_end-rcvd_beg)>0 ORDER BY duration DESC
vpn-Top-SSL-VPN-Web-Mode-Users-By-Bandwidth	Top SSL VPN web mode users by bandwidth usage	event	SELECT user_src, remote_ip, sum(sent_end-sent_beg+rcvd_end-rcvd_beg) AS bandwidth, sum(rcvd_end-rcvd_beg) AS traffic_in, sum(sent_end-sent_beg) AS traffic_out FROM ### (SELECT coalesce(nullifna(\`user\`), ipstr(\`remip\`)) AS user_src, remip AS remote_ip, tunnelid, min(coalesce(sentbyte, 0)) AS sent_beg,

Dataset Name	Description	Log Category	Query Syntax
			max (coalesce(sentbyte, 0)) AS sent_end, min(coalesce(rcvdbyte, 0)) AS rcvd_beg, max(coalesce(rcvdbyte, 0)) AS rcvd_end FROM $log WHERE $filter AND subtype='vpn' AND tunneltype='ssl-web' AND action='tunnel-stats' AND coalesce(nullifna(`user`), ipstr(`remip`)) IS NOT NULL AND tunnelid IS NOT NULL GROUP BY tunnelid, user_src, remip ORDER BY tunnelid)### t GROUP BY user_src, remote_ip HAVING sum(sent_end-sent_beg+rcvd_end-rcvd_beg)>0 ORDER BY bandwidth DESC
vpn-Top-SSL-Web-Users-By-Bandwidth-and-Avail	Top SSL web users by bandwidth usage and avail	event	SELECT user_src, remote_ip, sum(traffic_out) AS traffic_out, sum(traffic_in) AS traffic_in, sum(bandwidth) AS bandwidth, sum(uptime) AS uptime FROM (SELECT user_src, remip AS remote_ip, tunnelid, devid, vd, sum(sent_end-sent_beg) AS traffic_out, sum(rcvd_end-rcvd_beg) AS traffic_in, sum(sent_end-sent_beg+rcvd_end-rcvd_beg) AS bandwidth, sum(duration_end-duration_beg) AS uptime FROM ### (SELECT tunnelid, coalesce(nullifna(`user`), ipstr(`remip`)) AS user_src, remip, devid, vd, min(coalesce(sentbyte, 0)) AS sent_beg, max(coalesce (sentbyte, 0)) AS sent_end, min(coalesce(rcvdbyte, 0)) AS rcvd_beg, max(coalesce(rcvdbyte, 0)) AS rcvd_end, min(coalesce(duration, 0)) AS duration_beg, max(coalesce(duration, 0)) AS duration_end FROM $log WHERE $filter AND subtype='vpn'

Dataset Name	Description	Log Category	Query Syntax
			AND action='tunnel-stats' AND tunneltype='ssl-web' AND coalesce(nullifna(`user`), ipstr(`remip`)) IS NOT NULL AND tunnelid IS NOT NULL GROUP BY tunnelid, user_src, remip, devid, vd ORDER BY tunnelid)### t GROUP BY user_src, remote_ip, tunnelid, devid, vd HAVING sum(sent_end-sent_beg+rcvd_end-rcvd_beg)>0 ORDER BY bandwidth DESC) t GROUP BY user_src, remote_ip ORDER BY bandwidth DESC
vpn-Top-Static-IPSEC-Tunnels-By-Bandwidth	Top static IPsec tunnels by bandwidth usage	event	SELECT vpn_name, sum(sent_end-sent_beg+rcvd_end-rcvd_beg) AS bandwidth, sum(rcvd_end-rcvd_beg) AS traffic_in, sum(sent_end-sent_beg) AS traffic_out FROM ### (SELECT vpn_trim(vpntunnel) AS vpn_name, tunnelid, 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 FROM $log WHERE $filter AND subtype='vpn' AND tunneltype LIKE 'ipsec%' AND (tunnelip IS NULL OR (tunnelip='0.0.0.0' AND coalesce(logver, 0)!=52)) AND action='tunnel-stats' AND tunnelid IS NOT NULL GROUP BY vpn_name, tunnelid ORDER BY tunnelid)### t GROUP BY vpn_name HAVING sum(sent_end-sent_beg+rcvd_end-rcvd_beg)>0 ORDER BY bandwidth DESC
vpn-Traffic-Usage-Trend-VPN	VPN traffic usage trend	event	SELECT hodex, sum(coalesce(ssl_bandwidth, 0)) AS ssl_bandwidth, sum(coalesce(ipsec_bandwidth, 0)) AS ipsec_bandwidth FROM ### (SELECT coalesce(t1.hodex, t2.hodex) AS hodex, ssl_bandwidth, ipsec_bandwidth FROM

Dataset Name	Description	Log Category	Query Syntax
			(SELECT hodex, sum(sent_end-sent_beg+rcvd_end-rcvd_beg) AS ssl_bandwidth FROM (SELECT $flex_timescale AS hodex, tunnelid, 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 FROM $log WHERE $filter AND subtype='vpn' AND action='tunnel-stats' AND tunnelid IS NOT NULL AND tunneltype LIKE 'ssl%' GROUP BY hodex, tunnelid, devid, vd ORDER BY tunnelid) t_ssl GROUP BY hodex) AS t1 FULL JOIN (SELECT hodex, sum(sent_end-sent_beg+rcvd_end-rcvd_beg) AS ipsec_bandwidth FROM (SELECT $flex_timescale AS hodex, tunnelid, 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 FROM $log WHERE $filter AND subtype='vpn' AND action='tunnel-stats' AND tunnelid IS NOT NULL AND tunneltype LIKE 'ipsec%' GROUP BY hodex, tunnelid, devid, vd ORDER BY tunnelid) t_ipsec GROUP BY hodex) AS t2 ON t1.hodex = t2.hodex)### t GROUP BY hodex ORDER BY hodex
vpn-User-	VPN user	event	SELECT hodex,

Dataset Name	Description	Log Category	Query Syntax
Login-history	login history		count(*) AS total_num FROM ### (SELECT t1.hodex AS hodex FROM ((SELECT $flex_timescale AS hodex, tunnelid FROM $log WHERE $filter AND subtype='vpn' AND (tunneltype='ipsec' OR tunneltype='ssl-web') AND action='tunnel-up' AND coalesce(nullifna(`xauthuser`), nullifna(`user`)) IS NOT NULL GROUP BY hodex, tunnelid ORDER BY hodex DESC) AS t1 INNER JOIN (SELECT tunnelid FROM (SELECT tunnelid, 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 FROM $log WHERE $filter AND subtype='vpn' AND tunneltype IN ('ipsec', 'ssl-web') AND action='tunnel-stats' AND tunnelid IS NOT NULL GROUP BY tunnelid ORDER BY tunnelid) tt GROUP BY tunnelid HAVING sum(sent_end-sent_beg+rcvd_end-rcvd_beg)>0) AS t2 ON t1.tunnelid=t2.tunnelid))### t GROUP BY hodex ORDER BY total_num DESC
web-Detailed-Website-Browsing-Log	Web detailed website browsing log	Traffic	SELECT FROM_dtime(dtime) AS TIMESTAMP, catdesc, hostname AS website, action AS status, sum(bandwidth) AS bandwidth FROM (### (SELECT dtime, catdesc, hostname, action, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth FROM $log-traffic WHERE $filter AND hostname IS NOT NULL AND logid_to_int(logid) NOT IN (4, 7,

Dataset Name	Description	Log Category	Query Syntax
			14) AND utmevent IN ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') GROUP BY dtime, catdesc, hostname, action ORDER BY dtime DESC)### UNION ALL ### (SELECT dtime, catdesc, hostname, action, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth FROM $log-webfilter WHERE $filter AND hostname IS NOT NULL AND (eventtype IS NULL OR logver=52) GROUP BY dtime, catdesc, hostname, action ORDER BY dtime DESC)###) t GROUP BY dtime, catdesc, website, status ORDER BY dtime DESC
webfilter-Categories-By-Bandwidth	Webfilter categories by bandwidth usage	webfilter	SELECT catdesc, sum(bandwidth) AS bandwidth FROM (### (SELECT catdesc, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth FROM $log-traffic WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND utmevent IN ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') AND catdesc IS NOT NULL GROUP BY catdesc ORDER BY bandwidth DESC)### UNION ALL ### (SELECT catdesc, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth FROM $log-webfilter WHERE $filter

Dataset Name	Description	Log Category	Query Syntax

			AND (eventtype IS NULL
			OR logver = 52)
			AND catdesc IS NOT NULL
			GROUP BY catdesc
			ORDER BY bandwidth DESC)###) t
			GROUP BY catdesc
			ORDER BY bandwidth DESC

webfilter-Top-Allowed-Web-Categories	Webfilter top allowed web categories	webfilter	SELECT catdesc, sum(requests) AS requests FROM (### (SELECT catdesc, count(*) AS requests FROM $log-traffic WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND utmevent IN ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') AND catdesc IS NOT NULL AND utmaction!='blocked' GROUP BY catdesc ORDER BY requests DESC)### UNION ALL ### (SELECT catdesc, count(*) AS requests FROM $log-webfilter WHERE $filter AND (eventtype IS NULL OR logver = 52) AND catdesc IS NOT NULL AND action!='blocked' GROUP BY catdesc ORDER BY requests DESC)###) t GROUP BY catdesc ORDER BY requests DESC
webfilter-Top-Allowed-Web-Sites-by-Bandwidth	Webfilter top allowed web sites by bandwidth usage	webfilter	SELECT DOMAIN, string_agg(DISTINCT catdesc, ', ') AS agg_catdesc, sum(bandwidth) AS bandwidth, sum(traffic_in) AS traffic_in, sum(traffic_out) AS traffic_out FROM (### (SELECT coalesce(nullifna(hostname), ipstr(`srcip`)) AS DOMAIN, catdesc, sum(coalesce(sentbyte, 0) +coalesce(rcvdbyte, 0)) AS bandwidth, sum(coalesce(rcvdbyte, 0)) AS traffic_in, sum(coalesce(sentbyte, 0)) AS traffic_out FROM $log-traffic WHERE $filter AND logid_to_int(logid) NOT IN (4, 7,

Dataset Name	Description	Log Category	Query Syntax
			14) AND utmevent IN ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') AND utmaction!='blocked' GROUP BY DOMAIN, catdesc HAVING sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 ORDER BY bandwidth DESC)### UNION ALL ### (SELECT coalesce(nullifna(hostname), ipstr(`srcip`)) AS DOMAIN, catdesc, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth, sum(coalesce(rcvdbyte, 0)) AS traffic_in, sum(coalesce(sentbyte, 0)) AS traffic_out FROM $log-webfilter WHERE $filter AND (eventtype IS NULL OR logver = 52) AND action!='blocked' GROUP BY DOMAIN, catdesc HAVING sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 ORDER BY bandwidth DESC)###) t GROUP BY DOMAIN, catdesc ORDER BY bandwidth DESC
webfilter-Top-Allowed-Web-Sites-By-Requests	Webfilter top allowed web sites by requests	webfilter	SELECT DOMAIN, string_agg(DISTINCT catdesc, ', ') AS agg_catdesc, sum(requests) AS requests FROM (### (SELECT hostname AS DOMAIN, catdesc, count(*) AS requests FROM $log-traffic WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND utmevent IN ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') AND hostname IS NOT NULL AND utmaction!='blocked' GROUP BY DOMAIN, catdesc ORDER BY requests DESC)### UNION ALL ### (SELECT hostname AS DOMAIN, catdesc,

Dataset Name	Description	Log Category	Query Syntax
			count(*) AS requests FROM $log-webfilter WHERE $filter AND (eventtype IS NULL OR logver = 52) AND hostname IS NOT NULL AND catdesc IS NOT NULL AND action!='blocked' GROUP BY DOMAIN, catdesc ORDER BY requests DESC)###) t GROUP BY DOMAIN ORDER BY requests DESC
webfilter-Top-Blocked-Web-Categories	Webfilter top blocked web categories	webfilter	SELECT catdesc, sum(requests) AS requests FROM (### (SELECT catdesc, count(*) AS requests FROM $log-traffic WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND utmevent IN ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') AND catdesc IS NOT NULL AND utmaction='blocked' GROUP BY catdesc ORDER BY requests DESC)### UNION ALL ### (SELECT catdesc, count(*) AS requests FROM $log-webfilter WHERE $filter AND (eventtype IS NULL OR logver = 52) AND catdesc IS NOT NULL AND action='blocked' GROUP BY catdesc ORDER BY requests DESC)###) t GROUP BY catdesc ORDER BY requests DESC
webfilter-Top-Blocked-Web-Sites-By-Requests	Webfilter top blocked web sites by requests	webfilter	SELECT DOMAIN, catdesc, sum(requests) AS requests FROM (### (SELECT hostname AS DOMAIN, catdesc, count(*) AS requests FROM $log-traffic WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND utmevent IN ('webfilter',

Dataset Name	Description	Log Category	Query Syntax
			'banned-word', 'web-content', 'command-block', 'script-filter') AND hostname IS NOT NULL AND utmaction='blocked' GROUP BY DOMAIN, catdesc ORDER BY requests DESC)### UNION ALL ### (SELECT hostname AS DOMAIN, catdesc, count(*) AS requests FROM $log-webfilter WHERE $filter AND (eventtype IS NULL OR logver = 52) AND hostname IS NOT NULL AND catdesc IS NOT NULL AND action='blocked' GROUP BY DOMAIN, catdesc ORDER BY requests DESC)###) t GROUP BY DOMAIN, catdesc ORDER BY requests DESC

webfilter-Top-Search-Phrases	Webfilter top search phrases	webfilter	SELECT keyword,
			count(*) AS requests
			FROM $log
			WHERE $filter
			AND keyword IS NOT NULL
			GROUP BY keyword
			ORDER BY requests DESC

webfilter-Top-Video-Streaming-Websites-By-Bandwidth	Webfilter top video streaming websites by bandwidth usage	webfilter	SELECT DOMAIN, sum(bandwidth) AS bandwidth, sum(traffic_in) AS traffic_in, sum(traffic_out) AS traffic_out FROM (### (SELECT coalesce(nullifna(root_domain(hostname)), 'other') AS DOMAIN, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth, sum(coalesce(rcvdbyte, 0)) AS traffic_in, sum(coalesce(sentbyte, 0)) AS traffic_out FROM $log-traffic WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND 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)

Dataset Name	Description	Log Category	Query Syntax
			+coalesce(rcvdbyte, 0))>0 ORDER BY bandwidth DESC)### UNION ALL ### (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-webfilter WHERE $filter AND (eventtype IS NULL OR logver = 52) AND catdesc IN ('Streaming Media and Download') GROUP BY DOMAIN HAVING sum(coalesce(sentbyte, 0) +coalesce(rcvdbyte, 0))>0 ORDER BY bandwidth DESC)###) t GROUP BY DOMAIN ORDER BY bandwidth DESC
webfilter-Top-Web-Users-By-Allowed-Requests	Webfilter top web users by allowed requests	webfilter	SELECT user_src, devtype, hostname_mac, sum(requests) AS requests FROM (### (SELECT coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) AS user_src, devtype, coalesce(nullifna(`srcname`), `srcmac`) AS hostname_mac, count(*) AS requests FROM $log-traffic WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND utmevent IN ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') AND coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) IS NOT NULL AND utmaction!='blocked' GROUP BY user_src, devtype, hostname_mac ORDER BY requests DESC)### UNION ALL ### (SELECT coalesce(nullifna(`user`), ipstr(`srcip`)) AS user_src, '0' AS devtype, ipstr(`srcip`) AS hostname_mac, count(*) AS requests FROM $log-webfilter WHERE $filter AND (eventtype IS NULL OR logver = 52) AND coalesce(nullifna(`user`), ipstr(`srcip`)) IS NOT NULL

Dataset Name	Description	Log Category	Query Syntax
			AND action!='blocked' GROUP BY user_src, devtype, hostname_mac ORDER BY requests DESC)###) t GROUP BY user_src, devtype, hostname_mac ORDER BY requests DESC
webfilter-Top-Web-Users-By-Bandwidth	Webfilter top web users by bandwidth usage	webfilter	SELECT user_src, devtype, hostname_mac, 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, devtype, coalesce(nullifna(\`srcname\`), \`srcmac\`) AS hostname_mac, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth, sum(coalesce(rcvdbyte, 0)) AS traffic_in, sum(coalesce(sentbyte, 0)) AS traffic_out FROM $log-traffic WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND utmevent IN ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') GROUP BY user_src, devtype, hostname_mac HAVING sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 ORDER BY bandwidth DESC)### UNION ALL ### (SELECT coalesce(nullifna(\`user\`), ipstr(\`srcip\`)) AS user_src, '0' AS devtype, ipstr(\`srcip\`) AS hostname_mac, 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-webfilter WHERE $filter AND (eventtype IS NULL OR logver = 52) GROUP BY user_src,

Dataset Name	Description	Log Category	Query Syntax
			devtype, hostname_mac HAVING sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 ORDER BY bandwidth DESC)###) t GROUP BY user_src, devtype, hostname_mac ORDER BY bandwidth DESC
webfilter-Top-Web-Users-By-Blocked-Requests	Webfilter top web users by blocked requests	webfilter	SELECT user_src, devtype, hostname_mac, sum(requests) AS requests FROM (### (SELECT coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) AS user_src, devtype, coalesce(nullifna(`srcname`), `srcmac`) AS hostname_mac, count(*) AS requests FROM $log-traffic WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND utmevent IN ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') AND coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) IS NOT NULL AND utmaction='blocked' GROUP BY user_src, devtype, hostname_mac ORDER BY requests DESC)### UNION ALL ### (SELECT coalesce(nullifna(`user`), ipstr(`srcip`)) AS user_src, '0' AS devtype, ipstr(`srcip`) AS hostname_mac, count(*) AS requests FROM $log-webfilter WHERE $filter AND (eventtype IS NULL OR logver = 52) AND coalesce(nullifna(`user`), ipstr(`srcip`)) IS NOT NULL AND action='blocked' GROUP BY user_src, devtype, hostname_mac ORDER BY requests DESC)###) t GROUP BY user_src, devtype, hostname_mac

Dataset Name	Description	Log Category	Query Syntax
			ORDER BY requests DESC
webfilter-Web-Activity-Summary-By-Requests	Webfilter web activity summary by requests	webfilter	SELECT hodex, sum(coalesce(allowed_request, 0)) AS allowed_request, sum(coalesce(blocked_request, 0)) AS blocked_request FROM (### (SELECT coalesce(t1.hodex, t2.hodex) AS hodex, allowed_request, blocked_request FROM (SELECT $flex_timescale AS hodex, count(*) AS allowed_request FROM $log-traffic WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND utmevent IN ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') AND utmaction!='blocked' GROUP BY hodex ORDER BY hodex) AS t1 FULL JOIN (SELECT $flex_timescale AS hodex, count(*) AS blocked_request FROM $log-traffic WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND utmevent IN ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') AND utmaction='blocked' GROUP BY hodex ORDER BY hodex) AS t2 ON t1.hodex = t2.hodex)### UNION ALL ### (SELECT coalesce(t1.hodex, t2.hodex) AS hodex, allowed_request, blocked_request FROM (SELECT $flex_timescale AS hodex, count(*) AS allowed_request FROM $log-webfilter WHERE $filter AND (eventtype IS NULL OR logver = 52) AND action!='blocked' GROUP BY hodex ORDER BY hodex) AS t1 FULL JOIN (SELECT $flex_timescale AS hodex, count(*) AS blocked_request FROM $log-webfilter WHERE $filter

Dataset Name	Description	Log Category	Query Syntax

			AND (eventtype IS NULL
			OR logver = 52)
			AND action='blocked'
			GROUP BY hodex
			ORDER BY hodex) AS t2 ON t1.hodex = t2.hodex)###) t
			GROUP BY hodex
			ORDER BY hodex

web-Hourly-Category-and-Website-Hits-Action	Web hourly category and website hits action	Traffic	SELECT hod, website, sum(hits) AS hits FROM (### (SELECT $hour_of_day AS hod, (hostname || ' (' || coalesce (`catdesc`, 'Unknown') || ')') AS website, count(*) AS hits FROM $log-traffic WHERE $filter AND hostname IS NOT NULL AND logid_to_int(logid) NOT IN (4, 7, 14) AND utmevent IN ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') GROUP BY hod, website ORDER BY hod, hits DESC)### UNION ALL ### (SELECT $hour_of_day AS hod, (hostname || ' (' || coalesce (`catdesc`, 'Unknown') || ')') AS website , count(*) AS hits FROM $log-webfilter WHERE $filter AND hostname IS NOT NULL AND (eventtype IS NULL OR logver=52) GROUP BY hod, website ORDER BY hod, hits DESC)###) t GROUP BY hod, website ORDER BY hod, hits DESC
web-Top-Category-and-Websites-by-Bandwidth	Web top category and websites by bandwidth usage	Traffic	SELECT website, catdesc, sum(bandwidth) AS bandwidth FROM (### (SELECT hostname AS website, catdesc, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth FROM $log-traffic WHERE $filter

Dataset Name	Description	Log Category	Query Syntax
			AND hostname IS NOT NULL AND logid_to_int(logid) NOT IN (4, 7, 14) AND utmevent IN ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') GROUP BY website, catdesc ORDER BY bandwidth DESC)### UNION ALL ### (SELECT hostname AS website, catdesc, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth FROM $log-webfilter WHERE $filter AND hostname IS NOT NULL AND (eventtype IS NULL OR logver=52) GROUP BY website, catdesc ORDER BY bandwidth DESC)###) t GROUP BY website, catdesc ORDER BY bandwidth DESC
web-Top-Category-and-Websites-by-Session	Web top category and websites by session	Traffic	SELECT website, catdesc, sum(hits) AS hits FROM (### (SELECT hostname AS website, catdesc, count(*) AS hits FROM $log-traffic WHERE $filter AND hostname IS NOT NULL AND logid_to_int(logid) NOT IN (4, 7, 14) AND utmevent IN ('webfilter', 'banned-word', 'web-content', 'command-block', 'script-filter') GROUP BY website, catdesc ORDER BY hits DESC)### UNION ALL ### (SELECT hostname AS website, catdesc, count(*) AS hits FROM $log-webfilter WHERE $filter AND hostname IS NOT NULL AND (eventtype IS NULL OR logver=52) GROUP BY website,

Dataset Name	Description	Log Category	Query Syntax

			catdesc
			ORDER BY hits DESC)###) t
			GROUP BY website,
			catdesc
			ORDER BY hits DESC

web-Top-User-Visted-Websites-by-Bandwidth	Web top user visted websites by bandwidth usage	Traffic	SELECT website, catdesc, sum(bandwidth) AS bandwidth FROM (### (SELECT hostname AS website, catdesc, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth FROM $log-traffic WHERE $filter AND hostname IS NOT NULL AND logid_to_int(logid) NOT IN (4, 7, 14) AND 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)### UNION ALL ### (SELECT hostname AS website, catdesc, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth FROM $log-webfilter WHERE $filter AND hostname IS NOT NULL AND (eventtype IS NULL OR logver=52) GROUP BY hostname, catdesc ORDER BY bandwidth DESC)###) t GROUP BY website, catdesc ORDER BY bandwidth DESC
web-Top-User-Visted-Websites-by-Session	Web top user visted websites by session	Traffic	SELECT website, catdesc, sum(sessions) AS sessions FROM (### (SELECT hostname AS website, catdesc, count(*) AS sessions FROM $log-traffic WHERE $filter AND hostname IS NOT NULL AND logid_to_int(logid) NOT IN (4, 7, 14) AND utmevent IN ('webfilter',

Dataset Name	Description	Log Category	Query Syntax
			'banned-word', 'web-content', 'command-block', 'script-filter') GROUP BY hostname, catdesc ORDER BY sessions DESC)### UNION ALL ### (SELECT hostname AS website, catdesc, count(*) AS sessions FROM $log-webfilter WHERE $filter AND hostname IS NOT NULL AND (eventtype IS NULL OR logver=52) GROUP BY hostname, catdesc ORDER BY sessions DESC)###) t GROUP BY website, catdesc ORDER BY sessions DESC
web-Top-Web-site-Sessions-by-Bandwidth	Web top web-site sessions by bandwidth usage	Traffic	SELECT FROM_dtime(dtime) AS TIMESTAMP, user_src, website, catdesc, cast(sum(dura)/60 AS decimal(18, 2)) AS dura, sum(bandwidth) AS bandwidth FROM ###(SELECT dtime, coalesce(nullifna(`user`), nullifna(`unauthuser`), ipstr(`srcip`)) AS user_src, hostname AS website, catdesc, sum(coalesce(duration, 0)) AS dura, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth FROM $log WHERE $filter AND hostname IS NOT NULL AND logid_to_int(logid) NOT IN (4, 7, 14) AND action IN ('accept', 'close', 'timeout') GROUP BY dtime, user_src, website, catdesc HAVING sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))> 0 ORDER BY bandwidth DESC)### t GROUP BY dtime, user_src, website, catdesc ORDER BY bandwidth DESC
wifi-Num-	WiFi num dis-	Traffic	SELECT count(DISTINCT srcmac) AS totalnum

Dataset Name	Description	Log Category	Query Syntax
Distinct-Client	tinct client		FROM ### (SELECT srcintf, srcssid, osname, osversion, devtype, srcmac, count(*) AS subtotal FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND (srcssid IS NOT NULL OR dstssid IS NOT NULL) AND srcmac IS NOT NULL GROUP BY srcintf, srcssid, osname, osversion, devtype, srcmac ORDER BY subtotal DESC)### t
wifi-Overall-Traffic	WiFi overall Traffic	Traffic	SELECT sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND (srcssid IS NOT NULL OR dstssid IS NOT NULL)
wifi-Top-AP-By-Bandwidth	Top access point by bandwidth usage	Traffic	SELECT srcintf, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND (srcssid IS NOT NULL OR dstssid IS NOT NULL) GROUP BY srcintf HAVING sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0 ORDER BY bandwidth DESC
wifi-Top-AP-By-Client	Top access point by client	Traffic	SELECT srcintf, count(DISTINCT srcmac) AS totalnum FROM ### (SELECT srcintf, srcssid, osname, osversion, devtype, srcmac, count(*) AS subtotal FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4,

Dataset Name	Description	Log Category	Query Syntax
			7, 14) AND (srcssid IS NOT NULL OR dstssid IS NOT NULL) AND srcmac IS NOT NULL GROUP BY srcintf, srcssid, osname, osversion, devtype, srcmac ORDER BY subtotal DESC)### t GROUP BY srcintf ORDER BY totalnum DESC
wifi-Top-App-By-Bandwidth	Top WiFi applications by bandwidth usage	Traffic	SELECT appid, app, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) 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
wifi-Top-Client-By-Bandwidth	Top WiFi client by bandwidth usage	Traffic	SELECT (coalesce(srcname, srcmac, 'unknown') || ' (' || coalesce (devtype, 'unknown') || ', ' || coalesce(osname, '') || (CASE WHEN osversion IS NULL THEN '' ELSE ' ' || osversion END) || ')') AS client, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND (srcssid IS NOT NULL OR dstssid IS NOT NULL) GROUP BY client HAVING sum(coalesce(sentbyte, 0)+coalesce (rcvdbyte, 0))>0 ORDER BY bandwidth DESC
wifi-Top-Device-By-Bandwidth	Top WiFi device by bandwidth usage	Traffic	SELECT devtype, sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth FROM $log WHERE $filter AND logid_to_int(logid) NOT IN (4, 7, 14) AND (srcssid IS NOT NULL OR dstssid IS NOT NULL) AND devtype IS NOT NULL

<table>
<tr><th>Dataset Name</th><th>Description</th><th>Log Category</th><th>Query Syntax</th></tr>
<tr><td></td><td></td><td></td><td>GROUP BY devtype HAVING sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0
ORDER BY bandwidth DESC</td></tr>
<tr><td>wifi-Top-Device-By-Client</td><td>Top WiFi device by client</td><td>Traffic</td><td>SELECT devtype,
count(DISTINCT srcmac) AS totalnum
FROM ###
(SELECT srcintf,
srcssid,
osname,
osversion,
devtype,
srcmac,
count(*) AS subtotal
FROM $log
WHERE $filter
AND logid_to_int(logid) NOT IN (4,
7,
14)
AND (srcssid IS NOT NULL
OR dstssid IS NOT NULL)
AND srcmac IS NOT NULL
GROUP BY srcintf,
srcssid,
osname,
osversion,
devtype,
srcmac
ORDER BY subtotal DESC)### t
WHERE devtype IS NOT NULL
GROUP BY devtype
ORDER BY totalnum DESC</td></tr>
<tr><td>wifi-Top-OS-By-Bandwidth</td><td>Top WiFi os by bandwidth usage</td><td>Traffic</td><td>SELECT (coalesce(osname, 'unknown') || ' ' || coalesce(osversion, '')) AS os,
sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth
FROM $log
WHERE $filter
AND logid_to_int(logid) NOT IN (4,
7,
14)
AND (srcssid IS NOT NULL
OR dstssid IS NOT NULL)
GROUP BY os HAVING sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0
ORDER BY bandwidth DESC</td></tr>
<tr><td>wifi-Top-OS-By-WiFi-Client</td><td>Top WiFi os by WiFi client</td><td>Traffic</td><td>SELECT (coalesce(osname, 'unknown') || ' ' || coalesce(osversion, '')) AS os,
count(DISTINCT srcmac) AS totalnum
FROM ###
(SELECT srcintf,
srcssid,
osname,
osversion,
devtype,
srcmac,
count(*) AS subtotal
FROM $log</td></tr>
</table>

Dataset Name	Description	Log Category	Query Syntax

			WHERE $filter
			  AND logid_to_int(logid) NOT IN (4,
			                  7,
			                  14)
			  AND (srcssid IS NOT NULL
			     OR dstssid IS NOT NULL)
			  AND srcmac IS NOT NULL
			 GROUP BY srcintf,
			      srcssid,
			      osname,
			      osversion,
			      devtype,
			      srcmac
			 ORDER BY subtotal DESC)### t
			GROUP BY os
			ORDER BY totalnum DESC

wifi-Top-SSID-By-Bandwidth	Top SSIDs by bandwidth usage	Traffic	SELECT srcssid,
			    sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0)) AS bandwidth
			FROM $log
			WHERE $filter
			 AND logid_to_int(logid) NOT IN (4,
			                 7,
			                 14)
			 AND srcssid IS NOT NULL
			GROUP BY srcssid HAVING sum(coalesce(sentbyte, 0)+coalesce(rcvdbyte, 0))>0
			ORDER BY bandwidth DESC

wifi-Top-SSID-By-Client	Top SSIDs by client	Traffic	SELECT srcssid,
			    count(DISTINCT srcmac) AS totalnum
			FROM ###
			 (SELECT srcintf,
			     srcssid,
			     osname,
			     osversion,
			     devtype,
			     srcmac,
			     count(*) AS subtotal
			  FROM $log
			  WHERE $filter
			  AND logid_to_int(logid) NOT IN (4,
			                  7,
			                  14)
			  AND (srcssid IS NOT NULL
			     OR dstssid IS NOT NULL)
			  AND srcmac IS NOT NULL
			 GROUP BY srcintf,
			      srcssid,
			      osname,
			      osversion,
			      devtype,
			      srcmac
			 ORDER BY subtotal DESC)### t
			WHERE srcssid IS NOT NULL
			GROUP BY srcssid
			ORDER BY totalnum 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
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

Macro Name	Description	Dataset Used	Log Category
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
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