Asset Manager Logging Guide

Asset Manager logs are useful for the purposes of *auditing*, *debugging*, and *notifying*.

1. **Auditing** - Auditing logs help customers with system security by providing a record of system access and changes to system configuration, among other things. The files can be sent to an external syslog server to be monitored in near real-time. An administrator or security officer can be notified of any unexpected events. The files can be used after an event as forensic evidence of nefarious activity.
2. **Debugging** - Asset Manager services produce debugging logs that Lumeta Support, quality engineers (QE) and developers can use to determine the root cause of bugs and anomalous behavior.
3. **Notification** - The CEF log contains notifications of scanning and configuration events in "Common Event Format." The log can be sent to an external syslog server to be monitored and made actionable by third-party software. Users can select which notifications to send to the CEF log on the GUI's Settings > Asset Manager Systems page.

Files and Directories in /var/log

Log files specific to Asset Manager are in **bold** type.

File	Type	Notes
aide/	Intrusion detection (file integrity)	By default, aide is not run, and the file in this directory is empty
anaconda.*	System install logs	
audit/	Linux auditing system	/etc/audit/audit.rules Use *auditctl*, *ausearch*, *aureport*
boot.log	On each reboot	
btmp	Failed login attempts	Read with *utmpdump*.
cef.log	Lumeta CEF logging	
cron	Cron jobs	Records every time a cron job is run
diagnostics/	Diagnostic files	This directory is created when “gather_diagnostics” is run. These are not normal log files but rather a snapshot of the system state when it was run.

File	Type	Notes
discovery-agent.out	Discovery agent debug log	
discovery-filemonitorlog	Trace ingestion log	
dmesg	System kernel buffer ring	
dracut.log	RAM disk created during system install	
lumeta_upgrade.log	Asset Manager upgrade	Exists only after an upgrade attempt
httpd24/	Apache and mod_sec logs	
java_install.log	Java installation	Empty
lastlog	Logins	Read with *last* command
lumeta-queries.log	Query timing log	
lumeta-warehouse-queries.log	Warehouse query timing log	
lumeta-webapp.out	API log	
lumeta-warehouse.out	Warehouse log	
lumeta-webapp-console.log	API stdout and stderr log	Thread dump from running gather diagnostics will go to this file
lumeta-warehouse-console.log	Warehouse stdout and stderr log	
maillog	Email	Only populated if you read the mail cron sends to root
messages	Main Linux log file	Now includes all CLI commands
netboot.log	System install log	

ntpstats/	NTP logging	Empty
performance-data/	Performance logging	
pg_log/	Postgres logs	
sa/	System activity info	Read with *sar* command
secure	Security-related logging	Logs *runuser*, *sudo*, *sshd* and *pam* usage; maybe others

File	Type	Notes
spooler	System spooler	Empty
tallylog	PAM module pam_tally2	For denying access after failed attempts to login. Not used by default.
wtmp	Login info	Read with *utmpdump*
yum.log	System install	

Apache

General httpd logs in error_log and access_log. Security module has two log files, modsec_audit.log and modsec_debug.log.

Httpd logging can be controlled in the apache config. See the following links:
https://httpd.apache.org/docs/1.3/logs.html
http://httpd.apache.org/docs/current/mod/mod_log_config.html

Logging for mod_security is configured in /etc/httpd/conf.d/mod_security.conf. A reference manual can be found here: https://github.com/SpiderLabs/ModSecurity/wiki/Reference-Manual-%28v2.x%29

Crond

The cron daemon logs every command that it runs. However by default it sends output (stderr or stdout) from those commands to root via email. The daemon can be configured to send the output to syslog instead of email by putting the following line in /etc/sysconfig/crond:

CRONDARGS=-s -m off

Postgres

Postgres logs to /var/log/pg_log/postgresql.log. Logging is configured in /var/lib/pgsql/13/data/postgresql.conf. It also logs to the syslog facility LOCAL3. Postgres auditing can be turned on and off from the CLI. It should only be enabled when necessary because of the large amount of data generated.

Auditing

Linux system auditing can be turned on and off via the “system audit” command. Database auditing can be turned on via the “system database-auditing” command.

Name	Path	Method
Auditing	/api/rest/management/auditing?enable=*boolean*	GET
SetDBAuditing	/api/rest/management/DBauditing	PUT
GetDBAuditing	/api/rest/management/DBauditing	GET

- All successful and unsuccessful login attempts
- All logoffs
- Additions, deletions, and modifications to user privileges, access rules, permissions, and user accounts including account attributes (such as type, identifier, and so on)

- Changes to programs and parameters (system, network, and security settings)

- hostname/domain
- /etc/sysconfig/network
- failed access to/deletions of files and programs
- successful file system mounts
- changes to sudoers
- loading/unloading of kernel modules

- time-related changes
- Use of setgid/setuid programs
- /etc/selinux changes

- Changes to critical hardware elements

- All mainframe and server system startups and shutdowns
- All password changes

- Abnormal system events (for example, performance deterioration, files filling up, programs ending abnormally)

- Attempts to perform unauthorized functions (that is, those events which create violations)

- approaching storage capacity & other storage issues

- Privileged account activity

- Audit log activity (initialization, interruption, unexpected stop, etc.)

- audit log volume capacity

Syslog

Out of the box, syslog-ng logs to the following files:

- /dev/console
- /var/log/messages
- /var/log/secure
- /var/log/maillog
- /var/log/spooler
- /var/log/boot.log
- /var/log/cron
- /var/log/kern
- /var/log/mail
- /var/log/snmpd.log

In addition, we use use syslog-ng to send the following files to a remote syslog server, if configured:

- lumeta-webapp.out
- discovery-agent.out
- audit.log
- ospf_ipv4_lsa.log
- ospf_ipv6_lsa.log
- httpd24/access_log
- httpd24/error_log
- httpd24/modsec_audit.log
- httpd24/modsec_debug.log
- lumeta-warehouse.out
- snmpd.log
- lumeta-queries.log

The syslog-ng admin guide has information about customizing the message format.

Syslog-ng Configuration

Syslog-ng uses three parameter types to configure logging: sources, filters and destinations. Sources can be programs that syslog-ng launches, log files from running programs, pipes, snmp traps, local mailboxes, the Linux kernel and the network, among others. A single logical source can be built from many real sources.

```
source s_sys {

file ("/proc/kmsg" program_override("kernel: ")); unix-stream ("/dev/log");

internal();

}

source s_webapp { file("/var/log/lumeta-webapp.out"); };
```

Filters serve to refine what gets logged based on log level, the subsystem that the messages originated from, the host or network the messages are coming from, or the source program of the messages. In addition, programs that log directly to syslog may specify a facility such as local0, local1, through to local7, for which we can filter. Filters can be composed of complex boolean expressions.

```
filter f_warn    { level(warning..emerg); };
filter f_default { level(info..emerg) and not (facility(mail) or facility(authpriv) or facility(cron) or facility
(local3) or facility(daemon)); };
```

Destinations can be files, other syslog or syslog-ng servers on the network, third-party logging software, email, an SQL databases or Unix sockets.

```
destination d_mesg   { file("/var/log/messages"); }; destination d_lumeta { tcp("syslog.corp.lumeta.com" ); };
```

A syslog-ng configuration uses the collection of sources, filters and destinations in a series of log definitions. Each log definition is a set of one or more logical sources plus a single destination and any number of filters (including none).

```
log { source(s_sys); filter(f_default); destination(d_mesg); }; log { source(s_sys); filter(f_warn); destination
(d_lumeta); };
```

Asset Manager

Asset Manager uses a combination of syslog-ng and log4j for logging our own software. CentOS and Linux utilities running on Asset Manager also produce logs as described above and their output can be controlled via syslog-ng, although we don't currently supply any way for users to do so.

The CLI log command can change the logging level of the webapp (API + dojo UI) and discovery components. Those components use the java log4j library. The CLI log command can change the destination of the webapp, discovery and CEF log files. They can remain on the local system or be copied to a remote syslog server. If copied to a remote server, the originals remain on the Asset Manager system.

The gather_diagnostics utility returns the entire contents of /var/log.

Logging Configuration via the CLI

The CLI log command is used to set log levels, set a remote syslog server, enable CEF logging to a remote syslog server and show log contents of the lumeta-webapp.out log file.

The CLI system command is used to enable or disable auditing.

The "log services" command shows services that we don't control with the log command.

The "log level" command sets the logging level for the API (and thus CLI and GUI) and the scan agent.

The CLI calls the helper script *log_config*, which in turn calls the API to make all the logging changes.

Name	Path	Method
SetLogServer	/api/rest/service/log/server	POST
SetCEFLogServer	/api/rest/service/log/server/cef	POST
GetLogLevel	/api/rest/service/log/level/$service	GET
SetLogLevel	/api/rest/service/log/level/$service	POST
GetLogServer	/api/rest/service/log/server	GET
GetCEFLogServer	/api/rest/service/log/server/cef	GET
GetLogServices	/api/rest/service/log/entries	GET
ShowLog	/api/rest/service/log/entries	GET

The API, written in Java, calls the helper script *observer_log* to manipulate the syslog-ng configuration files and to read the log entries from lumeta-webapp.out.

Webapp and Discovery Logging

The lumeta-webapp includes the dojo GUI and the API and anything that calls the API (CLI and scripts, although that's not distinguished in the log file). Both the webapp and the discovery agent (aka scan agent) use the log4j library. They also print to stdout and stderr.

Currently, all the logging output goes to lumeta-webapp.out and discovery-agent.out, respectively. The stdout and stderr output in these log files is not formatted the same way that the log4j messages are.

Log4j is initialized via the log4j.xml files, which are compiled into the .war files for the two applications. We define "appenders" in the log4j.xml file to send the log messages to the console.

The files that start the two applications are *lumeta-webapp* and *discovery-agent* in /etc/init.d/. Within each script is a java command line with the options -outfile and -errfile followed by the name of the appropriate log file. These options control where the console output goes. This is why log4j messages and stdout and stderr messages all go to the same file.

The well-formatted log messages (those from log4j and not stdout and stderr) have the following format:

```
Date                                [x thread         ] level (module                                )
-    message
```

```
Apr 03 2018 11:59:48.568 [A main        ] INFO (DatabaseManager             ) - Initializing Database
```

The letter that precedes the thread name is one of {W, A, U, D} for Webapp, API, UI and Discovery-agent, respectively, indicating the .war file that generated the message. Currently, D messages will only appear in discovery-agent.out and W/A/U messages will only appear in lumeta-webapp.out.

Query Logging

By default, query logging is off, but query timing logging is on. Query timing data gets sent to its own log file, lumeta-queries.log. The queries themselves, if logging is enabled get sent to lumeta-webapp.out.

Here is a synopsis of the CLI commands for query and query timing logging:

All SQL queries go to lumeta-webapp.out:

```
log level set DEBUG API com.lumeta.api.sql
```

Specific SQL query goes to lumeta-webapp.out:

```
log level set DEBUG API com.lumeta.api.sql.SystemDao.getSystems
```

No SQL queries go to lumeta-webapp.out (except those that have been set explicitly):

```
log level set NONE API com.lumeta.api.sql
```

Don't log specific query:

```
log level set NONE API com.lumeta.api.sql.SystemDao.getSystems
```

By default, all query timings go to lumeta-queries.log

Turn off query timings (except those that have been set explicitly):

```
log level set NONE API com.lumeta.api.query
```

Turn on timings for specific query:

```
log level set DEBUG API com.lumeta.api.query.EventDao.addNotification
```

Turn off query timings for specific query

```
log level set NONE API com.lumeta.api.query.EventDao.addNotification
```

In above commands, NONE can be INFO, WARN or ERROR for same effect (disabling logging).

Also log setup time to lumeta-queries.log:

```
log level set TRACE API com.lumeta.api.query
```

```
log level set TRACE API com.lumeta.api.query.EventDao.addNotification
```

CEF Logging

In addition to the logging for webapp and discovery defined above, we log Asset Manager notifications in a “Common Event Format.” The log file can be sent to a remote syslog server for integration with third-party applications and tools. The CEF implementation is described in the Confluence document Common Event Format Support for Asset Manager.

We define a log4j appender in log4j.xml (see next section) which outputs the CEF log messages to syslog-ng using facility LOCAL1. The formatting of the messages is implemented in Java in the CEFLayout.java file.

The following format is used:

Date host **CEF:***CEF version* **|***vendor***|***product*|*Lumeta version***|***notification type***|***notification name***|***priority***|msg=***message* **cat=***category* **dvchost=***system name* **rt=***Date* **cn1=***zone id* **cn1Label=***zone name* **suser=***user name* **dhost=***ip* **c6a3=***ip* **mac=***mac addr*

```
Apr 10 09:53:06 65.246.240.183 CEF:0|Lumeta|ESI|3.3.1.11447|COLLECTOR_UPDATED|Collector
```

```
Updated|5|msg=Collector c Config Inserted cat=CONFIG dvchost=rick1 rt=Apr 10 2018 09:53:06 cn1=1 cn1Label=Zone1
suser=admin
```

In the above message format:

- CEF version is always 0
- vendor is always Asset Manager
- product is always ESI

- priority is always 5
- the following fields are optional and only appear for notifications that contain the data:
    - cn1
    - cn1Label
    - suser
    - dhost
    - c6a3
    - Mac

The remote CEF log server can be defined in the CLI using the “log cefserver” command, or it can be configured in the GUI on the Asset Manager Systems page under the CEF Notifications tab. Also on the CEF Notifications tab, an administrator can configure which device and system notifications get logged to the CEF log server.

More on log4j

On startup, the lumeta-webapp and discovery-agent programs read in their log4j.xml files. These files, in addition to setting up appenders, set the initial log levels for various components. They also set up appenders and levels for the query timings and CEF logging.

Most modules declare loggers using the path name of the class defined in the module. For example, com.lumeta.api.dao.UserDaoImpl. These paths define a hierarchy. We can set log levels for an individual module like UserDaoImpl, or for an entire subtree (for example, com.lumeta.api), which propagates down to the leaves (UserDaoImpl, in this case, but all the other modules under com.lumeta.api) except where the level has already been explicitly set.

When we use the CLI or API to set log levels, the levels get recorded in the database (system.loglevel table) and get reapplied if the applications restart.

Syslog-ng

Although syslog-ng has the concept of filters and log levels, we only modify the syslog-ng configuration to change the remote syslog and CEF servers. Instead, we use log4j to set the log levels and everything we log that’s at or above those levels goes into the log files. If remote logging is set up, then syslog-ng reads the log files and send the messages to the remote server, where the administrator of that server may decide to apply additional filtering.

Is command centre up and fully started?
What log messages to look for when things start? “Started” in lumeta-webapp.out

Look for

INFO (Main) - Starting

and

INFO (Main) - Started

Errors or warnings in between those lines may indicate an issue even when the UI seems to be functioning properly.

Once you see “Started,” the UI and API should be ready for new connections.

Required processes up and running

Here are portions of output from the ps command, annotated to indicate if duplicate lines are expected. Some of the lines have been truncated and may appear that way when viewed with top or ps.