

Audit and Security Logging

Cloud Execution Environment

INTERWORK DESCRIPTION

Copyright

© Ericsson AB 2016–2018. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

Trademark List

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark Information.



Contents

1	Scope	1
2	General	2
3	Description	4
3.1	Protocol Versions	4
4	Message and Signal Definition	5
5	Logging in CEE	6
5.1	Connecting to syslog Service	6
5.2	Log Filtering	7
5.3	Log Files Written by rsyslog	8
5.4	Log Rotation	10
5.5	Remote Logging	10
5.6	Audit Logging	11
5.7	SBI and SIEM Configuration	12
6	Functions and Procedure Declaration	13
7	Constants Declarations	14
	Reference List	15



1 Scope

This document describes the Northbound Interface (NBI) of the Log Aggregator that is part of the Cloud Execution Environment (CEE).

The arrow between the Log Aggregator and the Log Collector in Figure 1 represents the Northbound Interface of the Log Aggregator.

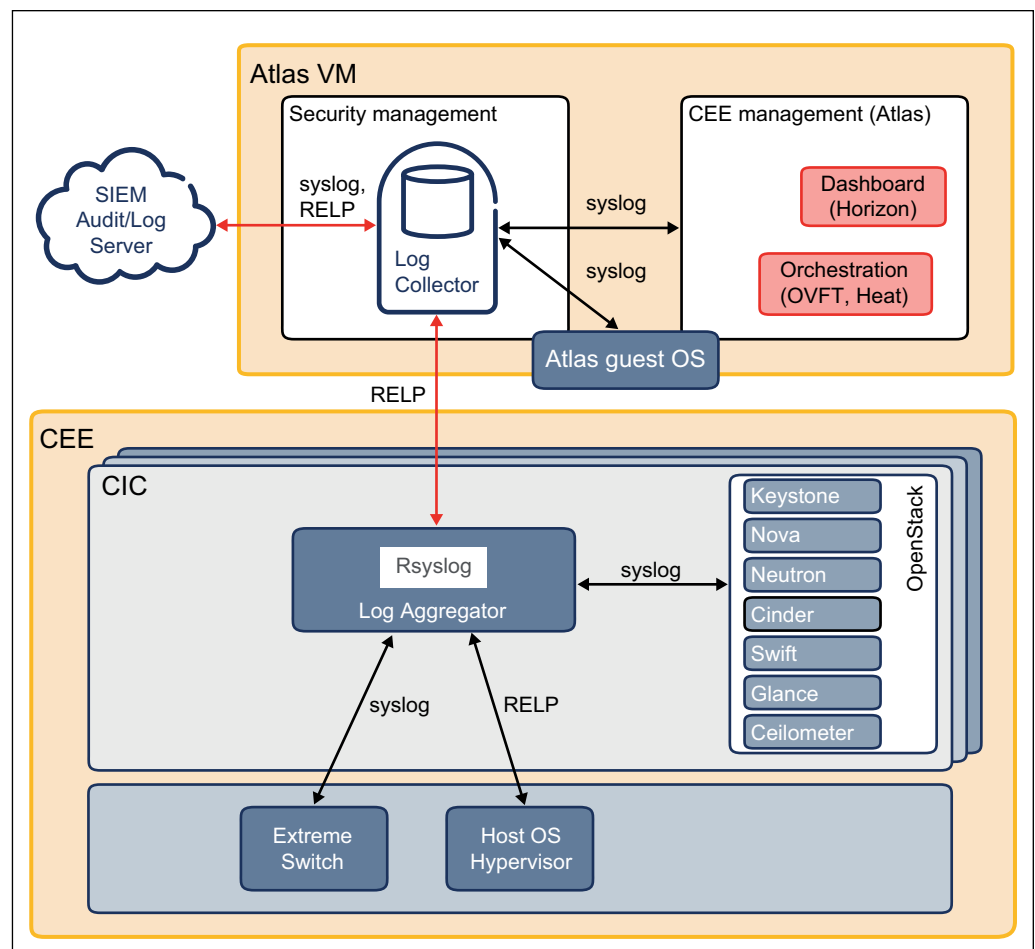


Figure 1 Security and Audit Logging



2 General

The Log Aggregator sends records over the Reliable Event Logging Protocol (RELP).

Event Sources

The audit and security logging system collects logs originating from the following event sources:

- Compute nodes
- Controller nodes
- Traffic switches (Extreme)

Audit and security log records from these nodes are either asynchronously pushed towards the Log Aggregator.

Log Aggregator

Security and Audit event records from the event sources are aggregated at the Log Aggregator and forwarded towards the Log Collector. The Log Aggregator functionality is performed by the `rsyslog` instance on the controllers, and it also acts as the system log. High availability of the Log Aggregator is ensured by the HAProxy front-end that performs health check of the Log Aggregator interfaces and distributes the load among the controllers.

Log Collector

Atlas acts as the collector of all audit and security events in CEE. `rsyslog` is configured to receive audit events over the RELP protocol and store them in a separate file system. The log collector listens on port 20514 for RELP connections. The incoming events are stored in a separate file system mounted at `/log-collector`. The size of the log-collector filesystem is 10% of the mounted data volume. The log file (`/log-collector/audit.log`) is truncated and rotated if it reaches the specified size limit. For an example of `audit.log` contents, see Example 1.

Note: Audit events received by the Log Collector can be forwarded to one or multiple external Security Information and Event Management (SIEM) systems for further analysis. Supported Protocols for event forwarding are RELP and `syslog` over TCP.



```
<14>1 2016-04-22T17:28:33.151518+05:30 localhost audispd - - - node=atlas type=DAEMON_START =>
msg=audit(1461326313.146:1040): auditd start, ver=2.3.2 format=nolog kernel=3.13.0-74-generic =>
auid=4294967295 pid=2039 subj=unconfined res=success
<14>1 2016-04-22T17:28:33.252499+05:30 localhost audispd - - - node=atlas type=KERNEL =>
msg=audit(1461326168.677:1): initialized
<14>1 2016-04-22T17:28:33.252513+05:30 localhost audispd - - - node=atlas type=AVC =>
msg=audit(1461326169.081:2): apparmor="STATUS" info="AppArmor sha1 policy hashing enabled" =>
pid=1 comm="swapper/0"
<14>1 2016-04-22T17:28:33.252523+05:30 localhost audispd - - - node=atlas type=AVC =>
msg=audit(1461326173.687:3): apparmor="STATUS" operation="profile_load" profile="unconfined" =>
name="/sbin/dhclient" pid=497 comm="apparmor_parser"
...
```

Example 1 Typical Contents of audit.log



3 Description

RELP is a networking protocol for computer data logging in computer networks. It is based on the ideas of the `syslog` protocol but extends it to provide reliable delivery of event messages. It is most often used in environments where message loss is not acceptable.

RELP uses a client-server model with (mostly) fixed roles. The initiating part of the connection is called the client, the listening part is called the server.

RELP uses the Transmission Control Protocol (TCP) for message transmission. This provides basic protection against message loss, but does not guarantee delivery under all circumstances. When a connection is aborted, it cannot be reliably detected if the last messages sent have actually reached their destination. Contrary to the `syslog` protocol, RELP works with a backchannel, over which information of messages processed by the receiver is conveyed back to the sender. This enables RELP to always know which messages have been properly received, even in the case of a connection abort.

For more information, see Reference [1] and Reference [2].

3.1 Protocol Versions

For protocol versions, refer to section [Protocol Versions in Security Information and Event Management](#).



4 Message and Signal Definition

RELP employs a command-response model, that is, the client issues commands to which the server responds. Each command is assigned with a (relatively) unique, monotonically increasing ID, called the Transaction Number (TXNR). Each response must include that ID. A command and its response is called a RELP transaction.

For more information, see Reference [1] and Reference [2].



5 Logging in CEE

This section contains technical and implementation details. For a generic overview of logging, refer to the [Security User Guide](#). For details on logging in CEE with SDN, refer to SDN document [Logging, Reference \[3\]](#).

In CEE, applications must be configured to use `syslog` for logging instead of sending logs to final destinations, for example writing directly to files. A centralized logging solution allows full control and end-to-end processing. Benefits of the centralized solution are:

- Writing log files to file system can be enabled or disabled without having to reconfigure applications.
- Logs can be transported off the host without the need to read back and parse log files.
- Log rotation does not restart individual applications, only the `syslog` daemon needs to be signaled to reopen output files.
- Files can be rotated without restarting individual applications or using the `copytruncate` mechanism, temporarily duplicating log files.

CEE uses `syslog` protocol RFC5424 to support handling complex multi-line messages. The messages written by `syslog` must contain a detailed header including:

- Timestamp containing year, month, date, hour, minute, second, and timezone offset
- Fully qualified hostname of the node the log originates from
- Application name

The `syslog` daemon implementation used in CEE is `rsyslog`. Unix DGRAM, UDP, TCP and RELP are available to send logs to `rsyslogd`. The on-wire format ensures proper handling of messages: message boundaries must remain intact and the header must remain unaltered, so the same decision that was made on the originating host can be made on other hosts. This is useful when, for example, logging to vCICs is enabled, so the same rule set can be used on vCICs and other nodes, to create log file hierarchy.

5.1 Connecting to syslog Service

The `syslog` daemon listens on the `/dev/log` socket. This is the preferred logging method for applications running on the same host. Networked `syslog` is possible, but applications cannot use networked `syslog` connections for local logging to reduce networking overhead. Using lossy transport, like UDP, must be avoided. In CEE, networked `syslog` is enabled only on the vCICs. On compute nodes



/dev/log is available, as these nodes are not syslog servers or relays. The following table lists available syslog sinks in CEE.

Table 1 syslog Sinks in CEE

Node Type	Socket and Protocol	Description	Use
vCIC	/dev/log Unix DGRAM/syslog	Default socket used by the libc syslog API	Apps submit logs locally using this socket
vCIC	UDP port 514/syslog	Networks syslog over UDP	Deprecated
vCIC	UDP port 10514/syslog	UDP log sink for audit logging	Deprecated, used by Extreme switches
vCIC	UDP port 8514/syslog	UDP log sink for remote logging	Deprecated, use not recommended
vCIC	TCP port 8514/syslog	TCP receiver for remote logging	Collecting logs from other nodes
vCIC	TCP port 20514/RELP	RELP log receiver	Audit logging
Compute/Cinder/baseOS	/dev/log Unix DGRAM/syslog	Default socket used by the libc syslog API	Apps submit logs locally using this socket
baseOS	TCP port 8514/syslog	TCP receiver	Used by the ScaleIO storage solution to submit logs

5.2 Log Filtering

rsyslogd parses incoming logs and handle logs based on how these were classified. OpenStack components are configured to use a specific syslog facility and filtering is done based on the syslog facility and the application name. For example, OpenStack Nova uses the LOCAL6 syslog facility and the nova application name prefix, so the corresponding rsyslog filter is:

```
if ($syslogfacility == 22) and ($programname startswith "nova")
```

Filtering rules act as a whitelist, so when no rule matches a log, then the log must be dropped.



Note: When local logging is enabled in `config.yaml`, non-matched messages are written to `/var/log/catch-all.log` so logs are not lost. This fallback is deprecated, so component owners must periodically check the file on deployments and when important information is present in this file, then the filtering rule set need to be updated.

5.3 Log Files Written by rsyslog

The following locations are written by `rsyslogd` (file and directory names are relative to `/var/log`):

- `aodh/` (OpenStack AODH component, vCIC-specific)
 - `aodh-api.log`
 - `aodh-dbsync.log`
 - `aodh-listener.log`
 - `aodh-notifier.log`
- `audittrail` (Ericsson logging component, vCIC only)
- `auth.log`
- `ceilometer/` (OpenStack Ceilometer component)
 - `ceilometer-agent-notification.log` (vCIC only)
 - `ceilometer-api.log` (vCIC only)
 - `ceilometer-collector.log` (vCIC only)
 - `ceilometer-dbsync.log` (vCIC only)
 - `ceilometer-expirer.log` (vCIC only)
 - `ceilometer-polling.log`
- `cinder/` (OpenStack Cinder component, vCIC only)
 - `cinder-api.log`
 - `cinder-backup.log`
 - `cinder-manage.log`
 - `cinder-scheduler.log`
 - `cinder-volume.log`
- `cmha.log` (Ericsson CM-HA component, vCIC only)
- `cron.log`



- `daemon.log`
- `dnsmasq/` (vCIC only)
 - `dnsmasq.log`
 - `dnsmasq-dhcp.log`
- `glance/` (OpenStack Glance component, vCIC only)
 - `glance-api.log`
 - `glance-cache-pruner.log`
 - `glance-glare.log`
 - `glance-manage.log`
 - `glance-registry.log`
- `haproxy.log` (vCIC only)
- `keystone/` (OpenStack Identity component, vCIC only)
 - `keystone-admin.log`
 - `keystone-public.log`
- `mongodb/` (vCIC only)
 - `mongod.27017.log`
- `mysqld.log` (MySQL database logfile, vCIC only)
- `neutron/` (OpenStack Neutron component)
 - `neutron-dhcp-agent.log` (vCIC only)
 - `neutron-openvswitch-agent.log`
 - `neutron-server.log` (vCIC only)
- `nova/` (OpenStack Nova component)
 - `nova-api.log` (vCIC only)
 - `nova-cert.log` (vCIC only)
 - `nova-conductor.log` (vCIC only)
 - `nova-consoleauth.log` (vCIC only)
 - `nova-manage.log`
 - `nova-novncproxy.log` (vCIC only)



- `nova-scheduler.log`
 - `pacemaker.log` (Corosync/Pacemaker, vCIC only)
 - `pmapi.log` (Ericsson PMAPI component, vCIC only)
 - `rsyslog-stats.log`
 - `sheriff.log` (Ericsson license management component, vCIC only)
 - `slapd.log` (OpenLDAP server log, vCIC only)
 - `swift-all.log` (OpenStack Swift component, vCIC only)
 - `watchmen.log` (Ericsson Watchmen component, vCIC only)
 - `remote/<scaleio_node>/scaleio/` (remote logs stored on ScaleIO nodes and vCIC)
- For remote logging, see Section 5.5 on page 10.
- `EMC-scaleio.log`

5.4 Log Rotation

Most logfiles written by `rsyslogd` are handled by the generic CEE log rotation job. It runs as a `cron` job every 30 minutes, the configuration file is `/etc/logrotate.d/fuel.nodaily`. `/var/log/cmha.log` and `/var/log/mondogb/*.log` are rotated by their log rotation jobs.

5.5 Remote Logging

Logs are sent off-host when any of the following settings under the logging section is enabled:

- `forward_to_fuel`
- `forward_to_controller`
- `forward_to_external`
- `auditlog_remote`

vCIC nodes are used as log relays when `forward_to_external` is enabled as other nodes are not allowed to access external sites. On the vCICs, logs originating from other nodes are stored under `/var/log/remote/<remote_hostname>`.

The protocol used for remote logging is hardcoded to octet-framed TCP. If a more reliable transport is needed, for example RELP, templates must be updated by hand, and disk-assisted queueing must be changed to disk queueing.



Note: The increased disk I/O can cause performance degradation affecting not just syslog but the entire system.

5.6 Audit Logging

The audit trail of CEE contains the following types of messages:

- Logs submitted by the Linux audit subsystem on infrastructure nodes (vCIC, Cinder, compute, baseOS)
- Logs submitted by Extreme switches
- Logs submitted by the ScaleIO storage solution (ScaleIO nodes, vCIC)

vCIC, compute, Cinder and baseOS nodes are configured to forward audit messages to the infrastructure audit log aggregator. The log aggregator service is running on the internal management VIP address TCP port 20514, haproxy forwards incoming TCP sessions to one of the vCICs. These sessions are load-balanced so multiple vCICs can act as log aggregator at the same time. The log aggregators forward audit logs to the log collector inside the Atlas VM (SBI in `config.yaml`), but audit logs can be written in `/var/log/audittrail` in the vCICs as well. Forwarding is done using the RELP protocol using on-disk queueing to reduce the likelihood of losing messages. The size of the on-disk buffer is limited so local disks do not get filled. When the queue is full, logs are discarded.

The Linux audit subsystem is preconfigured on CEE. `/etc/audit/auditd.conf` is tuned to be capable of handling a large amount of audit events and non-blocking transport is used for event dispatching to ensure that the kernel does not get blocked by auditd. CEE-specific auditd rules are read from the `/etc/audit/audit.rules` file. Audit configuration is locked down after being loaded, so any change requires a reboot. The audit dispatcher daemon is configured to forward logs to syslog using the `/etc/audisp/plugins.d/syslog.conf` file. The rsyslog configuration contains a specific rule matching for messages from the audit dispatcher daemon `audispd`, and rsyslog injects matching logs in the audit trail (`/etc/rsyslog.d/00-01-log-aggregator-client.conf`).

Extreme switches are configured to send syslog messages to the internal management VIP address. The switches support UDP only, so for this purpose the dedicated UDP port 10514 is used. Packets sent to the VIP address (UDP port 10514) are redirected with DNAT on the vCICs to the local IP address, so the packets are received by `rsyslogd` running in the global namespace (the VIP address is configured in the haproxy namespace where there is no application that can receive the UDP packets). `/etc/rsyslog.d/00-00-log-aggregator-server.conf` contains the configuration for receiving logs from Extreme switches and forwarding these messages to the audit trail.



5.7 SBI and SIEM Configuration

The network target of audit trail logging is configured on the vCICs in `/etc/rsyslog.d/00-00-log-aggregator-server.conf`. The RELP protocol is used and the remote log server is configured to be the SBI address specified in `config.yaml`; when installed, the Atlas component uses this address, which is configured to accept incoming logs.

It is recommended to secure logs forwarded to external log collectors like a SIEM solution, to ensure the authenticity of messages. For this purpose `rsyslog` can be configured to use RELP over TLS. In the underlying TLS channel it is recommended to enforce mutual authentication so the endpoints involved can communicate using a verified connection.

To enable TLS between the vCIC and Atlas, add the following options for the `send_to_collector` action defined in the `/etc/rsyslog.d/00-00-log-aggregator-server.conf` file on the vCICs:

```
tls="on"
tls.caCert="/etc/ssl/certs/CEE/ca-bundle.crt"
tls.myCert="/etc/ssl/certs/CEE/ctrl.crt"
tls.myPrivKey="/etc/ssl/certs/CEE/ctrl.crt"
tls.authmode="name"
tls.permittedpeer=["<security_nbi_atlas_hostname>"]
tls.prioritystring="<security_gnutls_priority>"
```

Where `<security_nbi_atlas_hostname>` and `<security_gnutls_priority>` must be replaced with the actual settings from `config.yaml`.

Similarly, update the RELP input configuration in `/etc/rsyslog.d/02_imrelp.conf` on the Atlas host to contain the following:

```
tls.caCert="</etc/ssl/certs/ca.crt>"
tls.myCert="</etc/ssl/certs/atlas.pem>"
tls.myPrivKey="</etc/ssl/private/atlas.key>"
tls.authmode="name"
tls.permittedpeer="<security_nbi_controller_hostname>"
tls.prioritystring="<security_gnutls_priority>"
```

An external SIEM log collector can be enabled on Atlas by configuring the RELP output definition in `/etc/rsyslog.d/39-siem.conf`. It is strongly recommended to enable TLS for this transport.



6 Functions and Procedure Declaration

For more information, see Reference [1] and Reference [2].



7 Constants Declarations

N/A



Reference List

- [1] RELP – The Reliable Event Logging Protocol, Rainer Gerhards,
<http://www.rsyslog.com/doc/relp.html>
- [2] Reliable Event Logging Protocol, from Wikipedia the free encyclopedia,
http://en.wikipedia.org/wiki/Reliable_Event_Logging_Protocol
- [3] Logging, 3/198 22-AXD 101 08/6-V1