Log File Structure and Filtering

https://campus.barracuda.com/doc/79463201/

The following table refers to the CC Syslog Service (For more information, see: Control Center Syslog Service) and explains the log file structure of service processes.

Log File Definition

[moduledir] = /opt/phion/modules/server/msyslog

Process Name	Executable	GUI Log File Name	Description
activate	[moduledir]/bin/activate	[server]_[service]	Configuration activation, on an optional CC HA partner the activation will also trigger the start of process [server]_[service]_sshd on both systems if HA synchronisation is configured as on.
[server]_[service]	[moduledir]/bin/msylogd	[server]_[service]	The actual service running on the active CC partner which is in charge of starting, terminating and monitoring of sub-processes.

[server]_[service]_slgd	/sbin/syslog-ng	[server]_[service]	The subprocess running on the active CC partner that corresponds to the actual syslog engine. This process is in charge of the actual log processing. Depending on the actual configuration settings it may write messages directly to the local disk on the active CC HA partner or transfer all [HA sync] or a filtered subset of messages to external UDP/TCP sockets using syslog protocol or to local TCP listening sockets on the loopback or to named pipes (FIFOs) from where they are read by some of the various sub-processes below.
[server]_[service]_sshc	[moduledir]/ssh/sshc.msyslog	n/a	The subprocess running on the active CC partner that is in charge of transferring log messages to the HA partner via SSHv2 port forwarding (client end).
[server]_[service]_sshd	[moduledir]/ssh/sshd.msyslog	[server]_[service]_ssh	The subprocess running on both CC HA partners that is in charge of receiving log messages from the active CC HA partner via SSHv2 protocol (server end) and forwarding them to the local syslogd process which will in turn write the messages to the local disk on the passive CC HA partner.

[server]_[service]_csslsrv	/usr/sbin/stunnel	[server]_[service]_csslsrv	The subprocess running on the active CC HA partner responsible for the termination and forwarding to the syslog engine of received SSL encapsulated log messages.
[server]_[service]_sslsrv	/usr/sbin/stunnel	[server]_[service]_sslsrv	The subprocess running on the active CC HA partner responsible for the termination of SSL connections (stunnel server) originating from external log host which seek to be fed relayed log messages. The subprocess will read from a named pipe (FIFO) upon successful connection by an external SSL client. Log messages are fed into the pipe by the syslog engine and reach the requestor via an SSL encapsulated log stream.

[server]_[service]_sslclt	/usr/sbin/stunnel	[server]_[service]_sslclt	The subprocess running on the active CC HA partner responsible for originating (stunnel client) SSL connections to external log hosts which are subsequently fed relayed log messages through the SSL connection. The subprocess will listen on a separate TCP listening socket per destination on the loopback for messages sent by the syslog engine and forward the messages via SSL encapsulated log streams to the log hosts.

Supported Ciphers and Cipher Preference by the Stunnel-based Sub-processes

AES128-SHA:DES-CBC3-SHA:AES256-SHA:DH-RSA-AES128-SHA:DHE-RSA-AES128-SHA:IDEA-CBC-SHA:EDH-RSA-DESCBC3-SHA

DES encryption is not supported due to its limited resistance against brute force attacks.

Filtering Policy

Structure of a syslog conformant log line as received by the syslog engine: **'['PRI']'[DATE/TIME] [HOSTNAME] [PROGRAM NAME]['['[PID]']']: [MESSAGE]\n**

'['PRI']'

Two digit decimal number enclosed in angled brackets containing information on both syslog facility and log level.

All logs sent by Barracuda CloudGen Firewall systems conform to syslog facility user.

- The log facility is a parameter that can be used when building filter conditions for log relaying.

[DATE/TIME]

Three letter English month abbreviation 'blank' day of month 'blank' 2-digit-hour [00-23]:2-digit-minute[00-59]:2-digit-second[00-59] example: Jul 31 14:08:01.

[HOSTNAME]

Hostname or IP address of the system the message originates from (possibly also the address of a relay host).

[PROGRAM NAME]]['['[PID]']']

Typically the name of the application the log message originates from. Note that an appended process ID number enclosed by square brackets may be part of this so-called **program name**. A colon follows the program name. The colon is used as indicator that all remaining portions of text actually belong to the actual log message part.

[MESSAGE]

The actual log message data.

Barracuda CloudGen Firewall gateways use the program name to add information as to the origin of a log message. To this end the actual log line is reconstructed before being sent to the gateway's syslog proxy service (*bsyslog*) for external delivery. The reconstruction entails replacing the original program name by the name of the log instance, that is the file, the log message would go into in directory */var/phion/logs* if it were solely written to disk. The original program name and message are simply moved further behind and now together form the new message part.

'['PRI']'[DATE/TIME] [HOSTNAME] [PROGRAM NAME]['['[PID]']']: [MESSAGE]\n

is changed to: **'['PRI']'[DATE/TIME] [HOSTNAME] [LOG-INSTANCE-NAME]: [PROGRAM NAME]['['[PID]']']: [MESSAGE]\n**

An example for a log instance name would be **box_Firewall** referring to log file `/var/phion/logs/box_Firewall.log`. The added **[LOG-INSTANCE-NAME]** is used by the Syslog Proxy service on a Barracuda CloudGen Firewall to find out as to which received log messages are supposed to be sent to which destination. On a per destination basis the program name field may

be overwritten by the syslog proxy before sending the log message on to the destination. The intention behind this is that this information is extracted by the CC Syslog Server to determine the local file underneath `/var/phion/mlogs` into which the log message is written and additionally this information may again be used for filtering purposes when log relaying to external security management systems by the CC is intended. The policy adopted by a Barracuda CloudGen Firewall is as follows:

CC-managed Box

Parameter	Value	Explicit Node Name	Explicit Hierarchy Info	Used Program Name
Add Range/Cluster Info	**yes**	-	-	[box name]/ [LOG-INSTANCE-NAME]
	no	-	-	-
Override Node Name	**no**	-	-	-
	yes	**[NAME]**	Range	[range/[NAME]
			Range and Cluster	[range]/[cluster]/[NAME]
			Range, Cluster and Box	[range]/[cluster]/[box name]/[NAME]
			Box	[box name]/[NAME]

Self-managed Box

Parameter	Value	Explicit Node Name	Explicit Hierarchy Info	Used Program Name
Override Node Name	**no**	-	-	[box name]/[LOG-INSTANCENAME]
	yes	**[NAME]**	none	[range/[NAME]
			Box	[box name]/[NAME]

The log messages received by the CC Syslog server thus contain additional information stored in the program name. First this information is used by the CC syslog server to determine the file into which a particular log message is meant to be written, provided local disc storage is desired. The log file is simply equal to `/var/phion/mlogs/[program name of log message]`.

From the table above it becomes clear that this mechanism allows for hierarchical depositing of log messages. If to override settings are used on the transmitting managed box all streamed log instances of the box are simply replicated under `/var/phion/mlogs/[range]/[cluster]/[box name]`. Yet it may sometimes be desirable to bundle together certain log contents, that are located in different files on the box, either for central storage or relaying purposes.

A good example for this is the firewall log. From the box's point of view firewall related log content goes into several files. On one hand there is the log output generated by the local firewall and on the

other hand there is the log output generated by the forwarding firewall service. In order to collect both outputs into a single file on the CC you would define a filter on the streaming box comprising the aforementioned two logging components and a destination corresponding to the CC where you now make use of the override node name option. Choosing for example 'allfirewall' as an explicit node name you have ascertained that a single file instance will be used on the CC. Depending on your exact intentions you may now adjust the explicit hierarchy information, that is the path information that is prepended to 'allfirewall'.