Logging Support

Cisco Unified Border Element (SP Edition) provides various features for working with logs. Logging can be configured so that logs are generated under specified conditions. Logs can also be generated on demand. Information derived from the logs can be used for analyzing and troubleshooting issues pertaining to the operation of the network and for identifying areas for improvement in the network.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to *Cisco Unified Border Element (SP Edition) Command Reference: Unified Model* at:

http://www.cisco.com/en/US/docs/ios/sbc/command/reference/sbcu_book.html

For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

Feature History for Logging Support

Release	Modification
Cisco IOS XE Release 2.x	The Syslog feature was introduced in a release earlier than Cisco IOS XE Release 3.1S.
Cisco IOS XE Release 3.5S	The Call Log Correlation feature was introduced to enable all the correlation logs associated with a particular call to be linked together using a correlator ID. The Alarms feature was enhanced to include new features for working with alarm logs.

Contents

This chapter contains the following sections:

Syslog Capabilities

All the Cisco Unified Border Element (SP Edition) debug messages that are displayed on the console are recorded in the Cisco IOS syslog. All the Cisco IOS syslog commands that configure log size, persistence, and redirection can be used for managing the syslog.

In addition to the console messages, Cisco Unified Border Element (SP Edition) records a log in its own internal buffer. This is known as the problem determination log and is saved in the event of a software-forced reload or as a result of using the **sbc dump-diagnostics** command. When you compile the problem reports, the problem determination log file is included as part of the problem reports.

Internal Log Levels

The Session Border Controller (SBC) application uses an internal log level to control the verbosity of the console and the PD log. Although both the console and problem determination log levels can be changed independently, we do not recommend changing the problem determination log level because the problem determination log buffer is of limited size and important logs may be lost.

The default SBC problem determination logging level is 63 for the console and 60 for the buffer. You can change the default SBC problem determination logging level using the **debug sbc log-level console** command, the **debug sbc log-level filter** command, or the **debug sbc log-level buffer** command.

Log Level	Syslog Level
90	Fatal
80	Error
70	Unexpected
63	Configuration Error
60	Operational
50	Audit
40	Statistics
30	Verbose Operational
20	Verbose Statistics
10	Internal Diagnostic

Enabling the Syslog Functionality

To enable the syslog functionality on the SBC, set the internal log levels, and issue the syslog-specific logging commands. The following example assumes a default problem determination level of 63 (no further action is needed if this is a fresh reboot).

1. Enable logging using the following commands:

```
Router# configure
Router(config)# logging enable
Router(config)# logging standby
```

Note The **logging standby** command allows the synchronization of the active and standby syslog settings.

2. Configure the location to which you want the syslog messages to be sent. Locations can be one of the following:
   - Console: logging console <1-7>

     ```
     Router(config)# logging console severity-level
     ```
   - Buffer: logging buffer <1-7>

     ```
     Router(config)# logging buffered severity-level
     ```

     Note Use the **show logging** command to view the logging statistics and the logging buffer. Use the **clear logging** command to clear the logging buffer.

   - Syslog server: logging trap <1-7>

     ```
     Router(config)# logging host ip_address [tcp[/port] | udp[/port]]
     Router(config)# logging trap severity-level
     Router(config)# logging device-id {hostname | ipaddress interface_name | string
     text | context-name}
     Router(config)# logging facility number
     ```

     Note The **logging device-id** command allows the customization of syslog messages when sending the log to a remote server.

   - Telnet sessions: logging monitor <1-7>

     ```
     Router(config)# logging monitor severity-level
     Router# terminal monitor
     ```
   - SNMP management station: logging history <1-7>

     ```
     Router(config)# logging history severity-level
     ```
   - Supervisor: logging supervisor <1-7>

     ```
     Router(config)# logging supervisor severity-level
     ```

3. Configure specific syslog message manipulation:

```
Router(config)# logging message syslog_id [level severity_level]
Router# show logging message
Router# clear logging
```

4. Configure the global syslog settings:

```
Router(config)# logging queue queue-size
Router# show logging queue
Router(config)# logging timestamp
Router(config)# logging rate-limit {num {interval | level severity_level |
message syslog_id} | unlimited {level severity_level | message syslog_id}}
Router# show logging
```

Call Log Correlation

The Call Log Correlation feature enables all the correlation logs associated with a particular call to be linked together using a correlator ID. This feature also enables real-time filtering of logs on a particular call. A 64-bit diagnostics correlator is assigned to each SIP call, REGISTER, SUBSCRIBE, or NOTIFY messages.

You can set the filters based on the following parameters:

- Dialed or dialing number
- Session Initiation Protocol (SIP) Universal Resource Identifier (URI)
- Remote signaling address
- Remote VPN ID
- Adjacency
- VRF

The logs that match the selected filter type are saved in a separate problem determination trace file and inter process signal (IPS) trace file.

Use the following command to enable the correlation-logs filter:

debug sbc *sbc-name* correlation-logs filter *filter-name* **[pdtrc-log-level** *value***]**

Use the following command to disable the correlation-logs filter:

no debug sbc *sbc-name* correlation-logs filter *filter-name*

Use the following command to display the debug logs, filters, and log levels:

show debugging

Problem Determination Log Levels

You can set the problem determination log level in the filter using the **pdtrc-log-level** option in the **debug sbc** *sbc-name* **correlation-logs filter** *filter-name* **[pdtrc-log-level** *value***]** command. The problem determination trace log level ranges from 0 to 100. The default log level is 60. A log level of 100 indicates that no logs are output, and 0 indicates that all the logs are output.

Table 44-1 lists the problem determination log levels:

Table 44-1 Problem Determination Log Levels

Problem Determination Log Level	Description
90	Critical system errors
80	Major system errors
70	Minor system errors
63	Configuration errors
60	Call errors
55	Call overview
50	Call details
40	Call statistics
30	Verbose operational
20	Verbose statistics
10	Internal diagnostic

Examples of Call Log Correlation Feature

The following example shows the various filters available for filtering the correlation logs:

```
Router# debug sbc test correlation-logs filter ?

   adjacency                 Adjacency, matching calls to or from this adjacency
   dn                        Dialed/dialing number,matching calls to or from this number
   remote-signalling-address Remote signalling address matching to or from this address
   sip-uri                   SIP-URI, matching calls to or from this URI
   vrf                       VRF name
```

The following example shows the filtering of correlation logs based on the adjacency parameter:

```
Router# debug sbc test correlation-logs filter adjacency abc
   Debugging filter log-level set to default level 60

Router# show debugging
   SBC correlator filter Adjacency name is abc
   IpsTracing is enabled
```

The following example shows the filtering of correlation logs based on the dialing number parameter:

```
Router# debug sbc test correlation-logs filter dn aa

   Debugging filter log-level set to default level 60

Router# show debugging
```

```
    SBC correlator filter DN is aa
    IpsTracing is enabled
```

The following example shows the filtering of correlation logs based on the remote signalling address parameter:

```
Router# debug sbc test correlation-logs filter remote-signalling-address ipv4 192.0.2.1

    Debugging filter log-level set to default level 60

Router# show debugging

    SBC buffer log-level is 0
    SBC correlator
    Filter Remote signalling-address ipv4 address is 192.0.2.1
    IpsTracing is enabled
    SBC correlator
    Filter DN is abc
    Pd loglevel is 70
    IpsTracing is enabled
```

The following example shows the filtering of correlation logs based on the SIP URI parameter:

```
Router# debug sbc test correlation-logs filter sip-uri ccc
    Debugging filter log-level set to default level 60

Router# show debugging

    SBC correlator filter Adjacency name is abc
    IpsTracing is enabled
    SBC correlator filter Remote signalling-address ipv4 address is 192.0.2.1
    IpsTracing is enabled
    SBC correlator filter SIP-URI is ccc
    IpsTracing is enabled
    SBC correlator filter DN is aa
    IpsTracing is enabled
```

The following example shows the filtering of correlation logs based on the VRF parameter:

```
Router# debug sbc test correlation-logs filter vrf new ipv4 rsa 192.0.2.1 pdtrc-log-level
70

    Debugging filter log-level set to default level 60

Router# show debugging

    SBC correlator Filter Remote signalling-address ipv4 address is 192.0.2.1
    SBC correlator Filter VRF is new with Vpn(id) = 3
    Pd loglevel is 70
    IpsTracing is enabled
    SBC correlator Filter SIP-URI is 9.0.0.0
    Pd loglevel is 0
    IpsTracing is enabled
```

Alarm Logs

You can configure the SBC to generate alarms for various types of events associated with the operation of the SBC. You can also configure the SBC to log debugging information, which you can use to monitor and tune the functioning of the system. On the basis of the alarms, you can take corrective and preventive action to ensure that the SBC continues functioning according to your business requirements. It is also

important to monitor the alarms generated by the SBC over a period of time and analyze this information. To address this requirement, you can configure the SBC to generate, display, and store alarm logs. The information provided in the alarm logs can help resolve some common issues, such as interoperability problems and incorrect configurations. These logs can also be used to identify issues that might potentially require escalation and investigation by more specialized support staff. Information in the logs can be used to improve the overall efficiency of the system.

Note All alarm log information is lost after a route processor failover.

You can use any combination of the following commands to configure alarm logs:

- Use the **debug sbc alarm-filter** command to specify the alarm types for which alarm logs must be generated.
- Use the **debug sbc alarm-log-level** command to specify the output mode and the alarm severity level for which alarms must be logged.
- The buffer that is used to store alarm logs may run out of free space while log files are written to it. In addition, you may want to store alarm logs for future reference. Use the **sbc periodic-dump-alarms** command to configure periodic movement of alarm log files from the buffer to a file system.
- Use the **sbc dump-alarms** command to move the alarm logs from the buffer to either a file system that you specify or the default file system configured on the router.

Configuring Alarm Logs

This task explains the commands that you can use to configure alarm logs. Note that it is not mandatory to use any particular command described in this task. You can use any combination of these commands to configure alarm logs.

SUMMARY STEPS

1. **debug sbc** *sbc-name* **alarm-filter** *alarm-type*
2. **debug sbc** *sbc-name* **alarm-log-level [buffer | console]** *severity-level*
3. **sbc periodic-dump-alarms {dump-location** *file-system* **[time-period** *time-period*] **| time-period** *time-period*}
4. **sbc dump-alarms [***file-system***]**
5. **show debugging**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**debug sbc** *sbc-name* **alarm-filter** *alarm-type* **Example:** `Router# debug sbc MySbc alarm-filter audit-congestion`	Configures the alarm types for which alarm logs must be generated. • *sbc-name*—Name of the SBC. • *alarm-type*—One of the following alarms: – **audit-congestion**—Call audit congestion. – **blacklist-alert**—Blacklist alert. – **blacklist-event**—Blacklist event. – **h248**—H248 connection failed. – **handled-exception**—Handled exception. – **routing-component**—Routing component set not active. – **routing-config**—Routing config set not active. – **routing-invalid**—Invalid routing configuration. – **sip-congestion**—SIP congestion detection. – **sip-peer**—SIP peer unavailable. – **vqm**—Voice Quality metrics (VQM) threshold exceeded.
Step 2	**debug sbc** *sbc-name* **alarm-log-level [buffer | console]** *severity-level* **Example:** `Router(config)# debug sbc MySbc alarm-log-level console 40`	Configures the output mode and the alarm severity level for which alarms must be logged. • *sbc-name*—Name of the SBC. • **buffer**—Specifies that alarm logs must be stored in the buffer. **Note** The size of a single log file created on the file system cannot exceed 2 MB. When the size of a particular log file reaches 2 MB, a new file is created and logging output is stored in the new file. • **console**—Specifies that logging output must be displayed on the console. • *severity-level*—Alarm severity level for which logs must be generated. The range is from 0 to 100. For alarm logs stored in the buffer, the default is 40. For alarm logs displayed on the console, the default is 80. To disable logging, set the value to 100. If you set the value to 0, logs are generated for all levels of alarm severity.

Command or Action	Purpose
Step 3 **sbc periodic-dump-alarms {dump-location** *file-system* **[time-period** *time-period***] | time-period** *time-period***}** **Example:** `Router(config-sbc)# sbc periodic-dump-alarms dump-location bootflash: time-period 120`	Configures periodic movement of alarm log files from the buffer to a file system. • **dump-location**—Specifies that you want the alarm logs to be stored in a file system. • *file-system*—Name of the file system where you want the alarm logs to be moved. For example, *file-system* can be one of the following: – **bootflash:** – **flash:** – **fpd:** – **ftp:** – **http:** – **https:** – **obfl:** – **pram:** – **rcp:** – **scp:** – **tftp:** • **time-period** *time-period*—Specifies the periodic time interval, in minutes, after you want the logs to be moved. The range is from 0 to 1440. The default is 60. **Note** When you run the **no** form of this command, the time period for moving logs is set to 0 and periodic movement of the logs is disabled.

	Command or Action	Purpose
Step 4	**sbc dump-alarms** [*file-system*] **Example:** `Router(config-sbc-sbe-cacpolicy)# sbc dump-alarms bootflash:`	Moves alarm logs from the buffer to either a file system that you specify or the default file system configured on the router. • *file-system*—Name of the file system where you want the alarm logs to be moved. For example, *file-system* can be one of the following: – **bootflash:** – **flash:** – **fpd:** – **ftp:** – **http:** – **https:** – **obfl:** – **pram:** – **rcp:** – **scp:** – **tftp:**
Step 5	**show debugging** **Example:** `Router# show debugging`	Displays information about the types of debugging that are enabled on the router. The output of this command includes debugging settings created by running the **debug sbc alarm-filter** command and the **debug sbc alarm-log-level** command.

The following sample output of the **show debugging** command shows the debugging settings created by running the **debug sbc alarm-filter** command and the **debug sbc alarm-log-level** command. In this example, these debug commands have been used to specify that logs must be generated for call audit congestion alarms that are of severity level 60 or higher and that these logs must be moved to the specified file system at 120-minute intervals:

```
Router# show debugging

SBC:
  SBC buffer alarm-log-level : 60
  SBC alarm filter 1 : AUDIT CONGESTION
  SBC alarm periodic dump time : 120 min
```