



# Manual Configuration

---

You can configure Cisco SIP proxy server (SPS) in either of two ways:

- Using the GUI-based provisioning system
- Manually editing text-based files

This appendix describes how to configure Cisco SPS manually. Unless your situation is highly unusual, do not perform manual configuration. Use the GUI-based provisioning system, as described in [Chapter 2, “Configuring Cisco SPS.”](#)



## Caution

All Cisco SPS 1.x versions require manual configuration. Therefore, for backward compatibility, Cisco SPS supports manual editing of all configuration files. However, if you use the GUI-based provisioning system, do not attempt manual editing. Manual changes to any configuration file written by the GUI are overwritten when the GUI is used again.

This appendix contains the following sections:

- [Prerequisites, page B-1](#)
- [How to Define the Cisco SPS Configuration File, page B-2](#)
- [How to Configure the SIP Proxy Server in a Farm, page B-26](#)
- [How to Configure IPSec, page B-28](#)

## Prerequisites

- Install Cisco SPS and activate the license as described in the *Cisco SIP Proxy Server Installation Guide*.
- Change to the directory in which the Cisco SPS configuration file (sipd.conf file) is located and open the file using a text editor such as vi. A default sipd.conf configuration file is copied at installation to the following location:

```
Linux: # /usr/local/sip/conf/  
# vi sipd.conf
```

```
Solaris: # /opt/sip/conf  
# vi sipd.conf
```

**Note**

Do not use a Microsoft Windows DOS text editor. When you save the sipd.conf file using a DOS editor, the <eol> (end of line) characters are changed and Cisco SPS has trouble reading them.

## How to Define the Cisco SPS Configuration File

This section describes the following:

- [Configuring Server-Global Directives](#), page B-3
- [Configuring Host-Specific Directives](#), page B-4
- [Configuring Server-Core Directives](#), page B-6
- [Configuring Standard Directives](#), page B-11

## Cisco SPS Configuration File

The Cisco SPS configuration file is sipd.conf. A default sipd.conf configuration file is copied at installation into the following:

**Linux:** /usr/local/sip/conf/

**Solaris:** /opt/sip/conf.

In most cases, you can use the default configuration for starting Cisco SPS and placing some test registrations and calls through it, but you might need to customize the defaults for your particular environment. If you make changes, you must restart the server in order for the changes to take effect. In general, a graceful restart is sufficient; however, for some changes, a complete restart is required. The directives for which changes require a complete restart are identified below.

## Cisco SPS Directives

Cisco SPS directives are similar to Apache server directives. Directives are grouped into four categories:

- Server-global directives—Define the overall operation of Cisco SPS. See the [“Configuring Server-Global Directives”](#) section on page B-3.
- Host-specific directives—Define basic Cisco SPS operations that are specific to a particular proxy server (rather than to a virtual proxy host that represents more than one server on one machine, as might be the case for companies that share a web server and yet each have their own domain (www.company1.com and www.company2.com) and access to the web server). See the [“Configuring Host-Specific Directives”](#) section on page B-4.
- Server-core directives—Define the primary SIP functionality of Cisco SPS. See the [“Configuring Server-Core Directives”](#) section on page B-6.
- Standard directives—Define Cisco SPS interfaces and additional functionality on a per-module basis. See the [“Configuring Standard Directives”](#) section on page B-11.

Directives are listed below in the order in which they appear in the Apache server (on which Cisco SPS is based). That is also the order in which the directives are written in the sipd.conf file and in which they appear under the various tabs in the GUI-based provisioning system.

## Configuring Server-Global Directives

Server-global directives are generic server directives that define the overall operation of the server. This does not include directives that configure protocol-specific (HTTP or SIP) details.

**Note**

Cisco SPS uses standard Apache directives to configure the SPS global environment. If the default for an Apache directive differs for Cisco SPS, the SPS default is listed below. For more detail on Apache directives, see the Apache website at <http://www.apache.org>.

### Detailed Steps

**Step 1** In the (Linux) `/usr/local/sip/conf/` or (Solaris) `/opt/sip/conf/` directory, open the `sipd.conf` file using a text editor such as `vi`:

```
# vi sipd.conf
```

**Step 2** Set the following directives as needed:

- **<ServerRoot>**—Directory in which Cisco SPS scripts and executables (`bin/`), configuration (`conf/`), and error-log (`logs/`) files reside. Do not add a forward slash (`/`) to the end of the directory path. Default is as follows:

**Linux:** `/usr/local/sip`

**Solaris:** `/opt/sip`

- **LockFile**—Path to the lockfile used when Cisco SPS is compiled with either `USE_FCNTL_SERIALIZED_ACCEPT` or `USE_FLOCK_SERIALIZED_ACCEPT`. (A lockfile is a type of file that Apache uses to let multiple processes access the same network socket.)

You should normally leave this directive at its default value. Change the value, though, if the logs directory is NFS mounted, because the lockfile must be stored on a local disk. For more information on lockfile location, refer to Apache's documentation on this directive at <http://www.apache.org/docs/mod/core.html#lockfile>.

The protocol identifier (PID) of the main server process is automatically appended to the filename. Default is `logs/accept.lock`.

- **PidFile**—Path and file where Cisco SPS records its process ID upon startup. A filename that does not begin with a forward slash (`/`) is assumed to be relative to `<ServerRoot>`. Default is `logs/sipd.pid`.
- **ScoreBoardFile**—Memory-mapped file in which internal server process information is stored. This file is automatically created if your architecture requires it. If the file is automatically created, ensure that no two servers share the same file. Default is `logs/apache_runtime_status`.
- **prefork MPM module**—Module that implements a nonthreaded, preforking web server for handling requests. This directive causes Cisco SPS to monitor child processes and, when necessary, spawn additional child processes to handle incoming SIP requests and responses. If too few requests and responses are being created, Cisco SPS tears down some idle child processes.

Maximum and minimum values for the following prefork MPM module directives depend on your available platform resources. Cisco SPS ignores prefork module directives if the server is run in single-process mode (`/sipd -DONE_PROCESS`) for debugging purposes.

- **StartServers**—Number of child processes that Cisco SPS creates upon startup. Default is 5.
- **MinSpareServers**—Minimum number of idle child processes (that is, processes that do not handle requests). Default is 5.

- **MaxSpareServers**—Maximum number of idle child processes. Such processes that exceed this number are torn down. Do not set this parameter to a large number. Default is 10.
- **MaxClients**—Number of simultaneous requests that Cisco SPS can support; this number must be no greater than the number of child processes to be created. Default is 20.
- **MaxRequestsPerChild**—Maximum number of requests that an individual child process (process that handles UDP traffic, IPC traffic, and timeouts) can handle during its life. If this number is exceeded, the child process is torn down and replaced by a new child process. This directive limits the amount of memory that processes can consume by accidental memory leakage. Timeouts occur every 50 milliseconds even in the absence of SIP traffic, and the counter is updated. Commonly used value is on the order of hours or days, (100000) or days (1000000). Default is 0 (child process is never torn down).
- **Listen**—List of ports or port and IP address combinations that the server listens to. This directive binds the server to specific IP addresses and specifies whether the server should listen to more than one IP address or port. If you specify only a port, the server responds to requests on all IP interfaces on that port. Valid entries include port and IP:port, but not IP only. Default is all requests on all IP interfaces. The following are examples of proper syntax:

```
Listen 3000
Listen 10.23.56.78:5060
```

**Step 3** Save and close the file.

---

## Configuring Host-Specific Directives

Host-specific directives define the basic configuration of Cisco SPS. They define server access, error logs, and the frequency with which logs rotate.



### Note

Cisco SPS uses standard Apache directives to configure the SPS basic configuration. If the default for an Apache directive differs for Cisco SPS, the SPS default is listed below. For more detail on Apache directives, see the Apache website at <http://www.apache.org>.

---

### Detailed Steps

---

**Step 1** In the (Linux) `/usr/local/sip/conf/` or (Solaris) `/opt/sip/conf/` directory, open the `sipd.conf` file using a text editor such as `vi`:

```
# vi sipd.conf
```

**Step 2** Set the following directives as needed:

- **Port**—Port on which Cisco SPS listens. Default is SIP port 5060. If this setting is less than 1023, Cisco SPS (`sipd` daemon) initially must be run as root. This is true even if `sipd` is to run as a different user or group.
- **User/Group**—Name or number of the user and group to run the `sipd` process as when `sipd` is started by the root user. Default is `cps`.
- **ServerName**—Hostname that clients use to create request URIs. This differs from the hostname that the server normally recognizes as its own. For example, the server name might be `sip-proxy.company.com` rather than the host's real name. This directive is useful for building a server

farm and publishing a single hostname for the farm. This directive is optional. Use it only if you have multiple servers in a farm. The `ServerName`, if defined, must be a valid Domain Name System (DNS) name for the host.

- **HostnameLookups**—Logs client DNS hostnames rather than of IP addresses. Valid values are `On` and `Off`. Default is `Off`.
- **ErrorLog**—Location of the error log file that contains Cisco SPS logs debug and error messages. Default is `logs/error_log`.

To automatically rotate error and debug logs daily without having to tear down and gracefully restart the Cisco SPS (`sipd` daemon), enter the following, being sure to specify the full path to both the `rotatelogs` script and the log file that you want to rotate:

**Linux:** `ErrorLog "/usr/local/sip/bin/rotatelogs /usr/local/sip/log/error_log 86400"`

**Solaris:** `ErrorLog "/opt/sip/bin/rotatelogs opt/sip/log/error_log 86400"`

Rotation-time default is 86400 seconds (24 hours).

- **LogLevel**—Verbosity of messages recorded in the error logs. Valid values are the following:
  - `emerg`—Emergencies and when system is unusable
  - `alert`—When action must be taken immediately
  - `crit`—Critical conditions
  - `error`—Error conditions
  - `warn`—Warning conditions (default)
  - `notice`—Normal but significant conditions
  - `info`—Informational
  - `debug`—Debugging messages
- **LogFormat**—Format nicknames of the log file, for use with `CustomLog`.

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
```

- **CustomLog**—Location and format of the access-log file. Default is `logs/access_log` `common`.

The location where transactions are logged depends on whether you define any access-log files within a `<VirtualHost>` container. If you do, transactions are logged in that container. If you do not, they are logged in the location specified for this directive.

To automatically rotate the access log daily without having to tear down and gracefully restart the Cisco SPS (`sipd` daemon), enter the following, being sure to specify the full path to both the `rotatelogs` script and the log file that you want to rotate:

**Linux:** `CommonLog "/usr/local/sip/bin/rotatelogs /usr/local/sip/logs/access_log 86400" common`

**Solaris:** `CommonLog "/opt/sip/bin/rotatelogs /opt/sip/logs/access_log 86400" common`

Rotation-time default is 86400 seconds (24 hours).

For `agent` and `referer` log files, use the following directives by removing the comment markers:

```
CustomLog logs/referer_log referer
CustomLog logs/agent_log agent
```

For a single log file that has access, agent, and referer information (combined log-file format), use the following directive by removing the comment marker:

```
CustomLog logs/access_log combined
```

**Step 3** Save and close the file.

---

## Configuring Server-Core Directives

Server-core directives govern how Cisco SPS functions as a redirect, registrar, or proxy server, either transaction stateful or stateless.

### Detailed Steps

**Step 1** In the (Linux) `/usr/local/sip/conf/` or (Solaris) `/opt/sip/conf` directory, open the `sipd.conf` file using a text editor such as `vi`:

```
# vi sipd.conf
```

**Step 2** Set the following directives as needed:

- **CSPSVersion**—Version of Cisco SPS that matches this configuration file. This directive is read-only; do not manually change it.
- **ProxyDomain**—Default domain to which Cisco SPS belongs. Valid value is a Domain Name System (DNS) domain suffix in standard fully qualified domain name (FQDN) format. There is no default.

#### Examples

```
mydomain.com
company.mydomain.com
```

- **StatefulServer**—Determines whether Cisco SPS is a transaction-stateful or transaction-stateless server. A transaction includes the following: received request, request or requests (if forked) forwarded downstream, responses received from downstream hosts, and best response returned upstream. Valid values are as follows:
  - On (stateful)—Cisco SPS remembers incoming and outgoing requests, provides reliable retransmission of proxied requests, and returns the best final responses.
  - Off (stateless)—Cisco SPS forgets all information once a request or response has been processed. It merely forwards requests and responses.

Default is On. If you change the value of this directive, you must restart the server.
- **SipResolveLocalContactsInRedirectMode**—Cisco SPS, if configured as a redirect server, returns next-hop routing information and updates contact information before returning the SIP 3xx response. Valid values are On and Off. Default is Off.
- **UseCallerPreferences**—Allows user-defined user-agent client (UAC) preferences to override administrator-defined preferences during request handling. Preferences include decisions such as whether to proxy or redirect a request, whether to fork a request (sequential or parallel), whether to recursively search, and to which URI to proxy or redirect a request. Valid values are On (use user-defined preferences) and Off (use administrator-defined preferences; ignore user-defined preferences). Default is On.

- **ServerType**—Determines whether Cisco SPS functions as a proxy server or as a redirect server. A proxy server processes and routes SIP requests. A redirect server provides contact information by means of SIP redirect (3xx) responses. Valid values are Proxy and Redirect. Default is Proxy.
- **Recursive**—Recursively tries addresses returned in a SIP 3xx redirection response. Valid values are On (tries addresses) and Off (SPS returns the lowest-numbered or best response). Default is On.
- **MaxForks**—Maximum number of branches that can be forked when Cisco SPS functions as a stateful server. Range is 1 to 6. Default is 5.
- **NumericUsernameInterpretation**—Lookup order for numeric user information in the Request-URI header field when the “;user=IP/PHONE” parameter is missing. Valid values are the following:
  - IP\_164—Process as URLs first and then as E.164 numbers.
  - E164\_IP—Process as E.164 numbers first and then as URLs (default).
  - IP—Process as URLs only.
  - E164—Process as E.164 numbers.
- **NumericUsernameCharacterSet**—Set of characters that Cisco SPS uses to determine whether the user-information portion of a Request-URI is in a category that applies to the “NumericUsernameInterpretation” processing step. This set does not apply to any user-information parameters. Default is +0123456789.-() (global phone number combinations). For more information on this directive, see the sipd.conf file.
- **OrigUserNameSource**—Origin of the UserName attribute in the accounting request message. Valid values are the following:
  - From—The user part of the URL in the From SIP header is used for authentication and to populate standard RADIUS accounting attribute #1 (UserName).
  - Auth—The user provided for authentication in the authorization or proxy-authorization header is used for authentication and billing. If no proxy-authorization header is present, the user is taken from the From header in the billing records.

Default is Auth.

- **NumExpandAuthUserName**—Applies number-expansion rules to the UserName received in the Authorization or Proxy-Authorization header. Valid values are On and Off. Default is On.
- **SrvForFqdnOnly**—Performs DNS server (SRV) lookups only for hosts that are FQDNs. If the host portion of the Request-URI header field does not contain an IP address but does contain a period, Cisco SPS determines the host to be an FQDN. Valid values are On (perform lookups only on FQDN hosts) and Off (perform lookups for any host that does not contain a target port). Default is Off.
- **SipT1InMs**—Time (in milliseconds) after which a request is first retransmitted if no response is received. Default is 500 (0.5 second).
- **SipT2InMs**—Time (in milliseconds) after which the backoff interval for non-INVITE requests does not increase exponentially. Default is 4000 (4 seconds).
- **SipT3InMs**—Default time (in milliseconds) that Cisco SPS waits after receiving a provisional response when processing an INVITE request. If a client does not include an Expires value in the INVITE, this value is used. Default is 60000 (60 seconds).
- **SipMaxT3InMs**—Maximum time (in milliseconds) that Cisco SPS waits after receiving a provisional response when processing an INVITE request. If a client includes an Expires value greater than this value in an INVITE, this value is used instead. Default is 180000 (180 seconds).
- **SipT4InMs**—Time (in milliseconds) that Cisco SPS maintains the transaction control block (TCB) after proxying a final response to a SIP INVITE request. Default is 32000 (32 seconds).

- **MaxInviteRetxCount**—Number of times that Cisco SPS can retransmit a SIP INVITE request. Range is 0 to 6. Default is 6.
- **MaxNonInviteRetxCount**—Number of times that Cisco SPS can retransmit a SIP request other than an INVITE. Range is 0 to 10. Default is 10.
- **SharedMemorySize**—Size (in bytes) of shared memory to be allocated for TCB. Range is 32000000 to 512000000. Default is 128000000. Recommended size is 128000000 (128 MB).
- **RegistryCleanupRate**—Time (in milliseconds) after which expired or deleted entries are removed from the registry. Default is 180000 (180 seconds).
- **AddRecordRoute**—Adds the Record-Route header to an initial SIP INVITE request. The Record-Route header field contains a globally reachable Request-URI that identifies the proxy server. When the proxy server adds the Request-URI to the Record-Route field in SIP messages, the server is kept in the path of subsequent requests for the same call leg. Valid values are On (add) and Off (do not add). Default is Off. ServerType must be set to Proxy for this directive to apply.
- **AddTransportInRecordRoute**—Forces use of a transport parameter in the Record-Route header. Doing so is useful for when the proxy server does not use the domain name in the path headers (that is, when ProxyAddressResolutionType is NOT SRV). Enabling this directive explicitly allows the proxy server to work with old equipment that does not yet support NAPTR and SRV. Valid values are On (force use) and Off (do not force use). Default is Off.
- **SipRouteHeaderTransportType**—Transport type for routes specified in Route headers of SIP requests handled by Cisco SPS. If the route contains an explicit transport parameter, this directive is ignored and the transport identified in the route header is used. Valid values are TCP, TLS, and UDP. Default is UDP.
- **AllowNaptrLookup**—Enables NAPTR lookup logic on the proxy server. Valid values are On (enable) and Off (disable; use TransportPrefOrder to select transport). Default is On.
- **TransportPrefOrder**—Transport preferences for times when NAPTR cannot be used or is unsuccessful. Valid values are the following: TLS\_TCP\_UDP, TLS\_UDP\_TCP, TCP\_TLS\_UDP, TCP\_UDP\_TLS, UDP\_TLS\_TCP, UDP\_TCP\_TLS, TLS\_TCP, TLS\_UDP, TCP\_TLS, TCP\_UDP, UDP\_TLS, UDP\_TCP, TLS, TCP, UDP. If SipTlsEnable is disabled, a transport preference of TLS is ignored. Default is TLS\_TCP\_UDP.
- **DiffServValue**—Value (in hex) to mark the type-of-service (TOS) byte of the IP header field of the transmitted SIP packets. Default is 0x60.

Values and their meanings are specified in RFC2474, RFC2475, RFC2597, and RFC3246. Valid values are as follows:

- Expedited Forwarding (EF) queue (RFC3246) value: 0xB8
- Assured Forwarding (AF) queue (RFC2597) values:

	Class 1	Class 2	Class 3	Class 4
Low drop	0x28	0x48	0x68	0x88
Medium drop	0x30	0x50	0x70	0x90
High drop	0x38	0x58	0x78	0x98

- IP routing (Class 6) value: 0xC0
- Streaming video (Class 4) value: 0x80
- Telephony signaling (voice & video) Class 3) value: 0x60
- Network management (Class 2) value: 0x40

- Scavenger (Class 1) value: 0x20
- Other (default, Class 0) value: 0x00

Some networks might alternatively recognize the type-of-service (RFC1349, RFC1812) bitmasks as follows.

- Minimize delay: 0x10
- Maximize throughput: 0x08
- Maximize reliability: 0x04
- Minimize cost: 0x02




---

**Note** This directive marks IP packets to a specified value. Marked packets receive special treatment only if network IP routers and switches are configured to provide it.

---

- **Sip\_Token\_Port**—Port that the synchronization server uses. This port must be the same for all servers in a farm. Default is 22794. If you change the value of this directive, you must restart the server.
- **Sip\_Services\_Port**—Port on the synchronization server. Default is 52931. If you change the value of this directive, you must restart the server.
- **RadiusUserNameAttrAddDomain**—Appends the domain in the From header to the username in the RADIUS UserName attribute (user@domain format). Domains other than ProxyDomain (default) or a domain in Virtual\_Proxy\_Domain are not appended. Valid values are On (append) and Off (do not append). Default is On.
- **RadiusRetransmissionInterval**—Time (in milliseconds) between retransmissions to the RADIUS server. Default is 2000.
- **RadiusRetransmissionCount**—Number of times to retransmit before deciding that the RADIUS server is unreachable. Default is 2.
- **RadiusRetransmissionAfterFailure**—Number of times to retransmit the current RADIUS request if all attempts to send the previous request fail. Default is 0.
- **RadiusRetryTime**—Time (in seconds) before retrying the primary RADIUS server, if it is out of service. Default is 300 (5 minutes).
- **ProxyAddressResolutionType**—Type of DNS configuration that is set up for SIP services in the proxy-server domain. Valid values are the following:
  - IP—No DNS configuration is available. The proxy server uses IP addresses in headers.
  - A—DNS is set up with A records corresponding to the ServerName directive. Hence, the proxy server uses the value of this directive in headers. If the ServerName directive is not set, then the proxy server uses its hostname SRV, which denotes that the proxy-server domain has SRV records set, and hence the proxy server uses the value of the ProxyDomain directive in headers.

Set this directive in conjunction with the ProxyDomain directive. Default is IP.

- **IpAddrInPathHeaders**—IP address to use in the Via and Record-Route path headers when ProxyAddressResolutionType is set to IP. Use this directive to control which address is used on multihomed servers. Default is to use the first value returned from gethostbyname.

- **IgnoreProxyRequire**—Behaves as if ProxyRequire headers are not present in the request. For example, suppose that an INVITE request contains ProxyRequire:extension-foo, but the proxy has no formal logic to understand extension-foo. RFC 2543 requires that a 420 response be returned, but with IgnoreProxyRequire configured, the INVITE is processed as if that particular header were not present. Valid values are strings of text that the proxy server is to ignore in headers.
- **SIPStatsLog**—Prints statistics to the stats\_log file. Valid values are On and Off. Default is On.
- **SIPStatsInterval**—Time (in seconds) for which statistics are logged. Default is 3600.
- **SharedMemoryStatsLog**—Enables debugging for shared memory. Valid values are On and Off. Default is Off.
- **SharedMemoryStatsInterval**—Time (in minutes) for which the log is written to the sharedmem\_stats\_log file in the logs directory. Default is 5.
- **SipTcpMaxTCPConnections**—Number of SIP TCP connections that can be open at any time. Default is 128.




---

**Note** This setting is ignored in favor of using limits enforced by the operating system. For both Linux and Solaris, this limit is 1024 (as set by FD\_SETSIZE) by default. Increasing this limit might degrade performance.

---

- **SipTcpMaxConnectTimeout**—Time (in milliseconds) that the server waits to connect to the client. Range is 150 to 10000 (10 seconds). Default is 1000 (1 second).
- **SipTcpReuseConnection**—Reuses the TCP connection for subsequent transactions with the same entity. Valid values are On (reuse) and Off (do not reuse). Default is Off.

All SIP entities using TCP for transport to one another should share the same setting. This prevents performance degradation and potential call failures. Otherwise, a proxy server that has this flag set to On continuously tries to reuse the same connection, even while another hop where this flag is set to Off is being torn down. For best performance, set to On only when all elements support that setting. To interwork with Cisco IOS gateways, set this to Off. If other entities in the network reuse connections, define persistent connections to those entities in the conf/persistent\_tcp.conf file.

- **SipTlsEnable**—Enables TLS. Valid values are On and Off. Default is Off. This directive is read-only during start/restart.
- **AllowSipTlsConversionToSip**—Terminates incoming SIPS requests on SIP contacts. Doing so presents a security risk; do so with caution and only if you know in advance that your endpoints and gateways are incapable of handling SIPS and TLS connections.
- **SipTlsPort**—TLS port. Default is 5061. This directive is read-only during start/restart.
- **SipTlsSessionTimeout**—Server-side-session cache timeout (in seconds). Sessions are not reusable after this timeout. Default is 300.
- **SipTlsCertificateFile**—Location of the PEM-encoded certificate file for the server. This directive is read-only during start/restart.
- **SipTlsCertificateKeyFile**—Location of the PEM-encoded private key file for the server. This directive is read-only during start/restart.
- **SipTlsCACertificateFile**—Location of certificates of certification authorities (CAs) with whose clients Cisco SPS deals. Cisco SPS uses this information for client authentication. This directive is read-only during start/restart.
- **SipTlsMutualAuthentication**—Directs the server to perform mutual authentication with the client. Valid values are On and Off. Default is Off.

- **DebugFlag StateMachine**—Logs LogLevel information on operation of the per-sipd child SIP state machine to <ServerRoot>/logs/error\_log. Valid values are On and Off. Default is Off.
- **DebugFlag Radius**—Logs LogLevel information for RADIUS messages to <ServerRoot>/logs/error\_log. Valid values are On and Off. Default is Off.
- **DebugFlag Parser**—Logs LogLevel information on operation of the per-sipd child SIP parser to <ServerRoot>/logs/error\_log. Valid values are On and Off. Default is Off.
- **DebugFlag SipTcp**—Logs LogLevel information on TCP transport of SIP messages by TCP services to <ServerRoot>/logs/error\_log. Valid values are On and Off. Default is Off.
- **DebugFlag SipTls**—Logs <LogLevel> information about the TLS transport of SIP messages by TCP services to <ServerRoot>/logs/error\_log. Valid values are On and Off. Default is Off.

**Step 3** Save and close the file.

---

## Configuring Standard Directives

Cisco SPS contains eight modules that you can use to configure a variety of interfaces, services, and features. This section includes the following information:

- [Configuring the MySQL Database Subscriber-Table Interface, page B-11](#)
- [Configuring the GKTMP Interface, page B-13](#)
- [Configuring Accounting Services, page B-13](#)
- [Configuring SIP Access Control, page B-15](#)
- [Configuring Authentication and Authorization, page B-17](#)
- [Configuring Preauthentication Query, page B-18](#)
- [Configuring Call Forwarding, page B-19](#)
- [Configuring Number Expansion, page B-20](#)
- [Configuring E.164 to Request-URI Address Translation, page B-21](#)
- [Configuring Next-Hop Routing, page B-22](#)
- [Configuring Registry Services, page B-23](#)
- [Configuring Virtual-Proxy-Server Hosts, page B-24](#)
- [Configuring H.323 RAS, page B-25](#)

## Configuring the MySQL Database Subscriber-Table Interface

You can configure an interface to a MySQL database subscriber table to maintain subscriber records for user authentication, authorization, accounting, and per-user call-forwarding. You can also map field names used by Cisco SPS to an existing MySQL subscriber table.

### Prerequisites

- If a MySQL database subscriber table exists in the network, use directives in the MySQL module to map the field names used by Cisco SPS to those used in the MySQL database subscriber table.
- If a MySQL subscriber table does not exist in the network, create one, using the `install_mysql_db` script (refer to the *Cisco SIP Proxy Server Installation Guide*).

**Note**

- If you use the GUI-based provisioning system, the MySQL database tables are created during provisioning-system installation, and you cannot modify the field names in the tables. Refer to the *Cisco SIP Proxy Server Installation Guide* for details.
- For information on working with MySQL databases, see the MySQL website at <http://www.mysql.com>.
- You can conduct a MySQL query at any time. Use the following information:
  - For terminating features such as the call forwarding: the “user” portion of the Request-URI
  - For originating features such as Authentication: the UserName from the Authorization, Proxy-Authorization Header, or From header

In either case, you may expand the key to a full E.164 number as needed before the MySQL query.

## Detailed Steps

- 
- Step 1** In the (Linux) `/usr/local/sip/conf/` or (Solaris) `/opt/sip/conf` directory, open the `sipd.conf` file using a text editor such as `vi`:

```
# vi sipd.conf
```

- Step 2** Set the following directives as needed:

- **DB\_MySQL**—Enables the MySQL database interface. That is, it establishes a TCP connection with the database. Valid values are On (connect) and Off (do not connect). Default is Off.
- **DB\_MySQL\_HostName**—Hostname or IP address of the system on which the primary MySQL database resides.
- **DB\_MySQL\_Secondary\_HostName**—Hostname or IP address of the system on which the secondary MySQL database resides.
- **DB\_MySQL\_DB**—Name of the database in which the subscriber table is stored and maintained.
- **DB\_MySQL\_Username**—Login username for the database account.
- **DB\_MySQL\_Password**—Login password for the database account.
- **DB\_MySQL\_SubscriberTable**—Name of the table in which the subscriber entries are stored.
- **DB\_MySQL\_Connect\_Timeout**—Timeout value (in seconds) for when Cisco SPS attempts to connect to the MySQL database server. After expiration of this time, SPS marks the connection as bad to prevent more child processes from blocking and resets the connection flag as soon as the server returns online. Adjust this value according to the traffic load on the server. A large value blocks more child processes than does a small value. Default is 3.
- **DB\_MySQL\_XXX\_Field**—Name equivalent in an existing MySQL database subscriber table. Use this directive to map Cisco SPS field names to their equivalent MySQL names.
- **DebugFlag DBMySQL**—Logs `mod-sip-db-mysql` debug messages to `<ServerRoot>/logs/error_log`. Valid values are On and Off. Default is Off.

- Step 3** Save and close the file.
-

## Configuring the GKTMP Interface

You can configure an interface to translate SIP protocol data units (PDUs) to the GateKeeper Transaction Message Protocol (GKTMP) protocol for local-number-portability (LNP) lookups, 1-800 and 1-900 number translations, and endpoint resolutions. A newly started Cisco SPS process initiates a TCP connection with a network-application-manager (NAM) server via the GKTMP interface.

### Detailed Steps

---

**Step 1** In the (Linux) `/usr/local/sip/conf/` or (Solaris) `/opt/sip/conf` directory, open the `sipd.conf` file using a text editor such as `vi`:

```
# vi sipd.conf
```

**Step 2** Set the following directives as needed:

- **GktmpConnection**—Enables the GKTMP interface. Valid values are On (enabled) and Off (disabled). Default is Off.
- **MasterServerHostname**—Hostname of the primary NAM server.
- **MasterServerIpAddress**—IP address of the primary NAM server.
- **MasterServerPort**—Destination port number of the primary NAM server and LNP lookup services.
- **SecondaryServerHostname**—Hostname of the secondary NAM server.
- **SecondaryServerIpAddress**—IP address of the secondary NAM server.
- **SecondaryServerPort**—Destination port number of the secondary NAM server.
- **GktmpTransportType**—Transport type for routes specified in GKTMP responses received by Cisco SPS. Valid values are TCP, TLS, and UDP. Default is UDP.
- **Debug Flag GKTMP**—Prints `mod_sip_gktmp` module debug messages to `logs/error_log`. Valid values are On (print) and Off (do not print). Default is Off.
- **DebugFlag GktmpAPI**—Logs `mod_sip_gktmp` API debug messages to `<ServerRoot>/logs/error_log`. Valid values are On and Off. Default is Off.

**Step 3** Save and close the file.

---

## Configuring Accounting Services

You can configure Cisco SPS to perform accounting services—that is, to generate and forward transaction or call information to a RADIUS server. This information is in the form of a RADIUS accounting-request message that contains standard billing information such as username, IP address of the proxy server that set up the call, message status type, type of port, session time, ID of the endpoint that is called, and ID of the endpoint that calls.

When accounting service is enabled and the interface to the RADIUS server is configured, Cisco SPS creates and sends accounting records to the RADIUS server according to how you set relevant directives (see [Table B-1](#) and [Table B-2](#)).

**Table B-1 AccountingServerSide Directive**

Condition 1	Condition 2	Returned Message	Sent Record
AccountingServerSide Directive is On	—	200 for INVITE upstream	Server-side START record
	—	Final response for BYE upstream (call is successful)	Server-side STOP record
	AccountingUnsuccessful directive is On	Non-200 final response for an INVITE upstream (call is unsuccessful)	

**Table B-2 AccountingClientSide Directive**

Condition 1	Condition 2	Received Message	Sent Record
AccountingClientSide directive is On	—	200 for INVITE from downstream	Client-side START record
	—	Final response for BYE from downstream (call is successful)	Client-side STOP record
	AccountingUnsuccessful directive is On	Non-200 final response for an INVITE from downstream (call is unsuccessful)	

When Cisco SPS receives an INVITE request from itself or a member of its registry or routing farm, server-side accounting is disabled for that INVITE, and no server-side START or STOP records are sent to the RADIUS server. Similarly, when Cisco SPS sends an INVITE to itself or a member of its registry or routing farm, client-side accounting is disabled for that INVITE, and no client-side START or STOP records are sent to the RADIUS server.

## Detailed Steps

**Step 1** In the (Linux) `/usr/local/sip/conf/` or (Solaris) `/opt/sip/conf` directory, open the `sipd.conf` file using a text editor such as `vi`:

```
# vi sipd.conf
```

**Step 2** Set the following directives as needed:

- **Accounting**—Logs accounting information on a RADIUS server. Valid values are On and Off. Default is Off.
- **AccountingServerSide**—Sends server-side accounting records for successful calls. Valid values are On and Off. Default is On.
- **AccountingClientSide**—Sends client-side accounting records for successful calls. Valid values are On and Off. Default is Off.

- **AccountingUnsuccessful**—Sends accounting records for unsuccessful calls. Valid values are On and Off. Default is Off. Operates as follows:
  - On—This directive is interpreted in conjunction with the AccountingServerSide and AccountingClientSide directives. If the AccountingServerSide and this flag are both On, accounting records are sent for server-side unsuccessful calls. If the AccountingClientSide and this flag are both On, accounting records are sent for client-side unsuccessful calls.
  - Off—No accounting records are sent for unsuccessful calls, regardless of the setting for the AccountingServerSide or AccountingClientSide flags.
- **AccountingRecordFormat**—Record format used for accounting. Currently, RADIUS is the only valid option.
- **AccountingTimeFormat**—Specifies whether timestamps are in local or GMT time.
- **PrimaryRadiusAcctIp**—IP address or hostname of the primary RADIUS server to use for accounting.
- **PrimaryRadiusAcctPort**—Destination port number of the primary RADIUS server to use for accounting.
- **PrimaryRadiusAcctSecret**—Secret text string shared between Cisco SPS and the primary RADIUS server to use for accounting.
- **SecondaryRadiusAcctIp**—IP address or hostname of the secondary RADIUS server to use for accounting.
- **SecondaryRadiusAcctPort**—Destination port number of the secondary RADIUS server to use for accounting.
- **SecondaryRadiusAcctSecret**—Secret text string shared between Cisco SPS and the secondary RADIUS server to use for accounting.
- **AcctIncludeSIPHeader**—Sends the SIP header in VSA #1 (AVPair) within RADIUS accounting messages as they are received by the proxy server—that is, with complete header line (from the 200 OK for the start request and from the BYE for the stop request). You can have a maximum of 50 headers in the sipd.conf file. For RADIUS, this directive is included as the value of Cisco AVPair 1 and attribute name sip-hdr.

**Step 3** Save and close the file.

---

## Configuring SIP Access Control

You can configure access to Cisco SPS.

### Detailed Steps

**Step 1** In the (Linux) /usr/local/sip/conf/ or (Solaris) /opt/sip/conf directory, open the sipd.conf file using a text editor such as vi:

```
# vi sipd.conf
```

**Step 2** Set the following directives as needed:

- **Order**—Default access state and the order in which Allow and Deny directives are evaluated. In all cases, every Allow and Deny statement is evaluated. There are two valid values:
  - Deny,Allow—Evaluate Deny directives before Allow directives. Allow access by default. Allow access to any client that matches an Allow directive or does not match a Deny directive.
  - Allow,Deny—Evaluate Allow directives before Deny directives. Deny access by default. Deny access to any client that does not match an Allow directive or matches a Deny directive.




---

**Note** Separate keywords only by a comma. Do not use blank spaces.

---

In the following example, all hosts in the company.com domain are allowed access and all other hosts are denied access.

```
Order Deny,Allow
Deny from all
Allow from company.com
```

In the following example, all hosts in the company.com domain are allowed access, except for hosts that are in the foo.company.com subdomain. All hosts not in the company.com domain are denied access because the default state is to deny access to the server.

```
Order Allow,Deny
Allow from company.com
Deny from foo.company.com
```

If the order in the last example is changed to Deny,Allow, all hosts are allowed access. Regardless of the ordering of the directives in the configuration file, the Allow from company.com is evaluated last and overrides the Deny from foo.company.com. All hosts not in the company.com domain are also allowed access because the default state changes to Allow.

- **Allow**—Which hosts are granted access to an area of the server. Access can be controlled by hostname, IP address, IP address range, or some other characteristic of the client request captured in an environment variable.
 

The first argument to this directive is always the *from* hostname. Subsequent arguments can take two different forms: all and host. If Allow from all is specified, all hosts are allowed access, subject to the Deny and Order settings (see below). To allow only particular hosts or groups of hosts to access the server, specify the host in any of the following formats:

  - Partial domain name (example: Allow from company.com)
  - Full IP address (example: Allow from 10.1.2.3)
  - Partial IP address (example: Allow from 10.1)
  - Network/netmask pair (example: Allow from 10.1.0.0/255.255.0.0)
  - Network/nnn CIDR specification (example: Allow from 10.1.0.0/16)
- **Deny**—Which hosts are denied access to an area of the server. Valid values are identical to those for the Allow directive.
- **Satisfy**—Access policy for both types of access control (Allow and Deny) and authentication checks. Valid values are All (allow and authenticate the sending host) and Any (grant access to the sending host if it passes an access-control-allow or authentication check). With either value, you must turn the authentication module On to ensure that an authentication check is performed.

**Step 3** Save and close the file.

## Configuring Authentication and Authorization

You can configure Cisco SPS to authenticate users or endpoints before it processes a transaction. You can specify that authentication be provided by SPS or a RADIUS server, and that it be done by means of HTTP Digest Authentication or HTTP Basic Authentication.

Authentication is based on the username that Cisco SPS extracts from the From, Authorization, or Proxy-Authorization header, regardless of where authentication takes place.

Cisco SPS expands the name to a full E.164 number before authentication according to the header type and to the rules in the relevant directive:

A Username from This Type of Header...	Expands According to Rules in This Directive...
From	User type and NumericUserNameInterpretation
Authorization or Proxy-Authorization	NumExpandAuthUserName

You can also configure Cisco SPS to ask for the domain of the user (as determined by the host portion of the header) as part of an authentication request.

### Detailed Steps

**Step 1** In the (Linux) `/usr/local/sip/conf/` or (Solaris) `/opt/sip/conf` directory, open the `sipd.conf` file using a text editor such as `vi`.

```
# vi sipd.conf
```

**Step 2** Set the following directives as needed:

- **Authentication**—Whether users must be authenticated before their transactions are processed. Valid values are `On` (authentication required) and `Off` (authentication not required). Default is `Off`.



**Note** User authentication does not occur if the following three conditions are true, because access control is already satisfied:

1. Access control is being used.
2. Hostname or IP address of the sender is covered by a corresponding `Allow` directive.
3. The `Satisfy` directive is set to `Any` instead of `All`.

- **AuthRealm**—Realm used in authentication response headers. Default is `CISCO`.
- **AuthServer**—Server on which user authentication takes place. Valid values are `Radius` and `Proxy`. Default is `Proxy`.
- **AuthScheme**—Authentication method to use when users require authentication before receiving service. Valid values are `HTTP_Digest` and `HTTP_Basic`. Default is `HTTP_Digest`.
- **AuthDigestQop**—Quality-of-protection (QoP) value for a digest-authentication challenge. Indicates the quality of protection supported. Valid values are `auth` (authentication only), `auth-int` (authentication and integrity), and `both` (allow the client to choose). Default is `auth`. For backward compatibility, `none` (from previous releases) is treated as `auth`.

- **AuthDigestAlgorithm**—Value of the algorithm to be included in a Digest Challenge to the user and used in Authentication Response headers. Valid values are MD5 (algorithm="MD5") and MD5-sess (algorithm="MD5-sess"). Default is MD5.
- **AuthConsumeProxyAuth**—Consumes (that is, strips off) the proxy-authorization header before forwarding a request downstream. Valid values are On (consume) and Off (pass downstream). Default is On. When a downstream device needs the header to identify the originator of the request, set to Off.
- **AuthAllow3rdPartyRegistration**—Checks unauthorized redirection of calls by a third-party registration. If set to Off, the username in the To header is matched with the username in the From or Authorization header; if these usernames do not match, registration is rejected, regardless of the form of authentication (Basic or Digest). Default is Off.
- **AuthAllow3rdPartyInvite**—Allows third-party INVITE requests for all forms of authentication (Basic or Digest). Valid values are On (user in the From header can differ from user used for authentication) and Off (user in the From header must match user used for authentication). Default is On.
- **RadiusAuthSkew**—Time (in seconds) for which a challenge is valid. Default is 30.
- **PrimaryRadiusAuthIp**—IP address or hostname of the primary RADIUS server to use for authentication.
- **PrimaryRadiusAuthPort**—Destination port number of the primary RADIUS server to use for authentication.
- **PrimaryRadiusAuthSecret**—Secret text string shared between Cisco SPS and the primary RADIUS server to use for authentication.
- **SecondaryRadiusAuthIp**—IP address or hostname of the secondary RADIUS server to use for authentication.
- **SecondaryRadiusAuthPort**—Destination port number of the secondary RADIUS server to use for authentication.
- **SecondaryRadiusAuthSecret**—Secret text string shared between Cisco SPS and the secondary RADIUS server to use for authentication.
- **AuthIncludeSIPHeader**—Sends the SIP header in VSA #1 (AVPair) within RADIUS accounting messages as they are received by the proxy server—that is, with complete header line (from the 200 OK for the start request and from the BYE for the stop request). You can have a maximum of 50 headers in the sipd.conf file. For RADIUS, this directive is included as the value of Cisco AVPair 1 and attribute name sip-hdr.

**Step 3** Save and close the file.

---

## Configuring Preauthentication Query

You can configure Cisco SPS to send a preauthorization query for a new INVITE request to a resource-pool-manager server (RPMS).

You start by creating a list of previous hops and a list of RPMSs that Cisco SPS is to check against. Then, during system operation, Cisco SPS checks a new INVITE request's previous hop against those in your list of previous hops. If it finds a match, it sends a preauthorization-query RADIUS message on behalf of the INVITE to an RPMS in your list of RPMSs. Cisco SPS does the following:

- If it receives an Accept response from the RPMS, it processes the INVITE normally.

- If it receives a Reject response, it returns a 408 (temporarily unavailable) message to the caller.
- If it receives no response within a specified wait time, it tries the next RPMS in the list. If all RPMSs in the list fail to respond, it processes the INVITE normally, as if the module is Off.
- It designates an RPMS to handle the next call as follows:
  - If any RPMS responds (whether the response is Accept or Reject) within a specified retry time (the counter for which starts when the first RPMS fails to respond), it designates that RPMS.
  - If all RPMSs fail to respond or if the retry time expires, it designates the first RPMS in the list.

Since preauthorization query messages are RADIUS messages, the new RPMS uses the RADIUS module to build and send preauthorization query RADIUS messages.

To see debug messages related to these events, turn on the following debug flags:

- DebugFlag StateMachine
- DebugFlag Radius
- DebugFlag RPMS

## Detailed Steps

- 
- Step 1** In the (Linux) `/usr/local/sip/conf/` or (Solaris) `/opt/sip/conf` directory, open the `sipd.conf` file using a text editor such as `vi`:
- ```
# vi sipd.conf
```
- Step 2** Set the following directives as needed:
- **PreAuthorization**—Preauthorizes new INVITE requests. Valid values are On and Off. Default is Off.
  - **PreAuthRequestType**—Preauthorization request type. Only valid value with this release is Query.
  - **RPMS\_ServerIpPortSecret**—List of RPMS IP addresses, port numbers, and secrets (passwords) for up to 10 servers.
  - **PreAuthPreviousHop**—IP address, hostname, or domain for up to 100 different hops, or the keyword ALL. The format of an entry is the same as that for an access-list entry.
  - **DebugFlag RPMS**—Logs RPMS module debug messages to `<ServerRoot>/logs/error_log`. Valid values are On and Off. Default is Off.
- Step 3** Save and close the file.
- 

## Configuring Call Forwarding

You can configure Cisco SPS to perform call forwarding, providing that you have a MySQL database.



### Note

For call forwarding, you need to define the corresponding subscribers. Rather than editing an existing subscriber, add a new one:

```
user --> 5100
domain --> cisco.com
```

Then add a call-forwarding URL for that user. Enable the corresponding call-forwarding feature in sipd.conf as well.

## Detailed Steps

**Step 1** In the (Linux) /usr/local/sip/conf/ or (Solaris) /opt/sip/conf directory, open the sipd.conf file using a text editor such as vi:

```
# vi sipd.conf
```

**Step 2** Set the following directives as needed:

- **CallForwardUnconditional**—Forwards calls unconditionally. Valid values are On (forward) and Off (do not forward).
- **CallForwardNoAnswer**—Forwards calls when a call is not answered within a designated amount of time. Valid values are On (forward) and Off (do not forward).
- **CallForwardBusy**—Forwards calls when when the called party is busy (a SIP 486 Busy Here response is received). Valid values are On (forward) and Off (do not forward).
- **CallForwardUnavailable**—Forwards calls when a user-agent client (UAC) is unavailable. Calls for users who are listed in the subscriber database but lack a valid registration, or who have a valid registration but do not respond within a designated time, are forwarded to the call-forward-unavailable location. Valid values are On (forward) and Off (do not forward).
- **CallForwardNoAnswerTimer**—Time (in milliseconds) after which to forward an unanswered call. Default is 24000 (24 seconds). This directive requires that CallForwardNoAnswer be set to On.
- **CallForwardUnavailableTimer**—Time (in milliseconds) after which to forward a call when a UAC is unavailable. Default is 24000 (24 seconds). This directive requires that CallForwardUnavailable be set to On.
- **AddDiversionHeader**—Includes the CC-Diversion header in SIP messages. Inclusion enables conveyance of call-redirection information during call setup. Valid values are On (include) and Off (exclude). Default is On if call forwarding is enabled.
- **DiversionHeaderName**—Name used for diversion headers generated by this proxy server. Valid values are Diversion, CC-Diversion, and CC-Redirect. Default is CC-Diversion.

**Step 3** Save and close the file.

## Configuring Number Expansion

You can configure support for global number-expansion plans.

### Detailed Steps

**Step 1** In the (Linux) /usr/local/sip/conf/ or (Solaris) /opt/sip/conf directory, open the sipd.conf file using a text editor such as vi:

```
# vi sipd.conf
```

**Step 2** Set the following directives as needed:

- **Cisco\_Numexpand**—Uses number expansion. Valid values are On and Off. Default is On.

- **DebugFlag Numexpand**—Logs number expansion-related debug messages to <ServerRoot>/logs/error\_log. Valid values are On and Off. Default is Off.

**Step 3** Save and close the file.

**Step 4** Start the number plan by assigning it a unique identifier:

```
<NumberPlan ID>
```

where *ID* is the unique identifier of the number plan (for example, global).

**Step 5** Specify the number plan, using one or more number-expansion directives:

```
NumExp <unexpanded-pattern> <expanded-pattern>
</NumberPlan>
```



**Note** You can use a period (.) as a wildcard to represent any digit.

**Example:**

```
<NumberPlan Global>
  NumExp 2.... +1919392....
  NumExp 7.... +1408527....
  NumExp 8... 5555...
</NumberPlan>
```

## Configuring E.164 to Request-URI Address Translation

You can configure Domain Name System (DNS) lookup for translation of an E.164 number (or any number in a number plan) into a list of Request-URIs.

### Detailed Steps

**Step 1** In the (Linux) /usr/local/sip/conf/ or (Solaris) /opt/sip/conf directory, open the sipd.conf file using a text editor such as vi:

```
# vi sipd.conf
```

**Step 2** Set the following directives as needed:

- **Cisco\_Enum**—Translates E.164 numbers to Request-URI s. Valid values are On and Off. Default is Off.
- **Cisco\_Enum\_Domain**—Private search domain for a private ENUM number plan. This directive is ignored if a Request-URI user begins with the plus (+) character, because the number is part of a global ENUM number plan (e164.arpa).
- **Cisco\_Enum\_Global\_Domain**—Domain to use in either of the following cases:
  - The Request-URI user begins with a plus (+) character (indicating a global domain)
  - A value is not specified for the Cisco\_Enum\_Domain directive
 Default is e164.arpa.
- **DebugFlag Enum**—Logs mod\_sip\_enum API debug messages to <ServerRoot>/logs/error\_log. Valid values are On and Off. Default is Off.

**Step 3** Save and close the file.

---

## Configuring Next-Hop Routing

You can configure Cisco SPS to perform next-hop route lookups for final Request-URIs by means of static route entries. SPS determines static routes by parsing directives such as the destination pattern, transport protocol, and target address.

### Detailed Steps

**Step 1** In the (Linux) `/usr/local/sip/conf/` or (Solaris) `/opt/sip/conf` directory, open the `sipd.conf` file using a text editor such as `vi`:

```
# vi sipd.conf
```

**Step 2** Set the following directives as needed:

- **Cisco\_Routing**—Perform next-hop routing. Valid values are On and Off. Default is On.
- **Cisco\_Routing\_Shared\_Memory\_Address**—Memory location of the routing table. Default is 0x35000000.
- **Cisco\_Routing\_Rendezvous\_Name**—Rendezvous name of the database containing routing information. Default is `routing_db`.
- **Cisco\_Routing\_Rendezvous\_Directory**—Location of the routing database. Default is `<ServerRoot>/data`.
- **Cisco\_Routing\_Remote\_Update\_Port**—Port number of the routing database server for all members of a server farm. The value for this directive must be the same for all members of a farm. Default is 22913. If you change the value of this directive, you must restart the server.
- **Cisco\_Routing\_Use\_Domain\_Routing**—Performs domain next-hop routing. This type of routing uses the host portion of the Request-URI as the key in obtaining the next one or more hops for a request. Valid values are On and Off. Default is Off.
- **Cisco\_Routing\_Farm\_Members**—Names of Cisco SPS proxy-server farm members, excluding the local host. Specify this list on all SPS farm members, being sure to exclude the local host in its own list because it is included implicitly. This list is used by the `sysadmin_sps_regroute` tool. If you change the value of this directive, you must restart the server.
- **Cisco\_Routing\_Max\_DB\_Age\_on\_Boot**—Maximum age (in seconds) of the database-backing store file at system startup. A file whose age exceeds this value is deleted. This value must be greater than the registry ageout value. Default is 86400 (24 hours). If you change the value of this directive, you must restart the server.

The value of this directive is particularly important if you have an external routing process for keeping the database current; if you set the directive unwisely, the externally populated routes might be deleted on system startup.

- **Cisco\_Routing\_Wildcard\_Expand\_Length**—Maximum number of digits in a destination-pattern wildcard entry, when the destination pattern type is phone (see `Static_Route_Type` in the following section). If an entry is longer than this value, Cisco SPS removes the wildcard character (\*) from the entry, and fails to expand the destination pattern entered by the user to a variable length. The portion of the destination pattern entry (without the wildcard) is added or deleted in the routing database as specified in the `Static_Route_Delete_or_Add` directive for this destination-pattern entry. Default is 25.

- **Cisco\_Routing\_Global\_Less\_Specific\_Route\_Search**—Searches the route database using less-specific patterns when all previously returned routes have been tried without final response or only a 5xx response. If any previously tried route has its AllowLessSpecificRoute field set to Off, Cisco SPS still stops the less-specific route search when those routes have been tried without final response or only a 5xx response. If this directive is set to Off, the value of the AllowLessSpecificRoute directive for an individual route has no effect. Valid values are On and Off. Default is Off. If you are configuring TLS, set to Off.
- **DebugFlag Routing**—Logs mod-sip-routing module debug messages to <ServerRoot>/logs/error\_log. Valid values are On and Off. Default is Off.

**Step 3** Save and close the file.

---

## Configuring Registry Services

You can configure Cisco SPS to process requests from user-agent clients (UACs) that register their location. When registry services are configured, Cisco SPS can do the following:

- Add a new registration
- Delete an existing registration
- Update an existing registration
- Delete all registrations of a user
- Return a current list of registrations of a user
- Periodically purge dated or expired registrations

### Detailed Steps

---

**Step 1** In the (Linux) /usr/local/sip/conf/ or (Solaris) /opt/sip/conf directory, open the sipd.conf file using a text editor such as vi:

```
# vi sipd.conf
```

**Step 2** Set the following directives as needed:

- **Cisco\_Registry**—Performs registry services. Valid values are On and Off. Default is On.
- **Cisco\_Registry\_Use\_Virtual\_Proxy\_Host**—Functions as a virtual-proxy-server host. Valid values are On and Off. Default is On.
- **Cisco\_Registry\_Shared\_Memory\_Address**—Memory location of the registration table. Default is 0x30000000.
- **Cisco\_Registry\_Rendezvous\_Name**—Rendezvous name of the database containing registration information. Default is registry\_db.
- **Cisco\_Registry\_Rendezvous\_Directory**—Location of the registration database. Default is <ServerRoot>/data.
- **Cisco\_Registry\_Remote\_Update\_Port**—Port number of the registration database server for all server-farm members. The value for this directive must be the same for all farm members. Default is 22913. If you change the value of this directive, you must restart the server.
- **Cisco\_Registry\_Farm\_Members**—Names or IP addresses of all remote proxy servers (other than this proxy server) that are contained within the same farm as this proxy server. This list is used by the sysadmin\_sps\_regroute tool.

**Note**

You must synchronize the system clock between farm members. We recommend that you use Network Time Protocol (NTP) to do so. It provides accuracy within a millisecond on LANs and up to a few tens of milliseconds on WANs. For more information on NTP, refer to the Network Time Protocol Project website at <http://www.ntp.org/>.

- **Cisco\_Registry\_Max\_DB\_Age\_on\_Boot**—Maximum age (in seconds) of the database-backing store file at system startup. A file whose age exceeds this value is deleted. This value must be greater than the registry ageout value. Default is 86400 (24 hours). If you change this value of this directive, you must restart the server.
- **DebugFlag Registry**—Logs mod\_sip\_registry module debug messages to <ServerRoot>/logs/error\_log. Valid values are On and Off. Default is Off.
- **Virtual\_Proxy\_Domain**—Unique DNS domain for this VirtualProxyHost. This value must be different from the value for ProxyDomain. (For information on the ProxyDomain directive, see the “Configuring Server-Core Directives” section on page B-6.) This directive is required. Examples of appropriate values for this directive are somedomain.org and foo.com.
- **Virtual\_Proxy\_Server\_Name**—Unique name for the virtual-proxy-server host. This value must be different from the actual name of the host. This directive is optional. Examples of appropriate values for this directive are usa.somedomain.org and usa.foo.com.
- **Virtual\_Proxy\_Server\_IP**—Unique IP address for this virtual-proxy-server host. This value must be different from the actual IP address of the host. This directive is optional. Examples of appropriate values for this directive are 10.23.2.2 and 192.168.2.2.

**Step 3** Save and close the file.

## Configuring Virtual-Proxy-Server Hosts

You can configure a virtual-proxy-server host and define up to 10 virtual-proxy-host entries.

### Detailed Steps

**Step 1** Assign a unique identifier to the virtual proxy host:

```
<VirtualProxyHost ID>
```

where *ID* is the unique identifier of this VirtualProxyHost configuration.

**Step 2** Set the following directives:

- **Virtual\_Proxy\_Domain**—Unique domain that Cisco SPS handles as VirtualProxyHost. Specify a value other than the actual domain of the proxy server.
- **Virtual\_Proxy\_Server\_Name**—Unique server name that Cisco SPS handles as VirtualProxyHost. Specify a value other than the actual server name of the proxy server.
- **Virtual\_Proxy\_Server\_IP**—Unique IP address that Cisco SPS handles as VirtualProxyHost. Specify a value other than actual IP address of the proxy server.

**Step 3** At the end of the entry, specify the following:

```
</VirtualProxyHost>
```

**Step 4** Save and close the file.

---

### Configuration Example

The following configuration example shows an entry for a virtual proxy host:

```
<VirtualProxyHost 1.1>
Virtual_Proxy_Domain foo.bar
Virtual_Proxy_Server_Name usa.foo.bar
Virtual_Proxy_Server_IP 61.12.1.1
</VirtualProxyHost>
```

## Configuring H.323 RAS

You can configure communication between Cisco SPS and a H.323 gatekeeper. Communication involves sending ASN.1 encoded Registration, Admission, and Status Protocol (RAS) LRQ messages to a provisioned H.323 gatekeeper and receiving LCF, LR,J or RIP messages from the gatekeeper.

### Detailed Steps

---

**Step 1** In the (Linux) /usr/local/sip/conf/ or (Solaris) /opt/sip/conf directory, open the sipd.conf file using a text editor such as vi:

```
# vi sipd.conf
```

**Step 2** Set the following directives as needed:

- **RASModule**—Enables the RAS module. Valid values are On and Off. Default is Off.
- **RASAcceptLCF**—LCF message to accept. Valid values are First (take the first valid LCF message from any gatekeeper) and Best (wait for the best LCF message within the LRQ time window; “best” means from the gateway that has the lowest cost and highest priority).
- **RASTimeoutInterval**—Time (in milliseconds) to wait for a single response from a gatekeeper. If the timeout expires, Cisco SPS tries another gatekeeper within the same cluster. You must set this directive if you set the RASLRQMethod directive to Sequential. Default is 300.
- **RASLRQMethod**—Method used to send LRQ messages to a gatekeeper. Valid values are Sequential (Cisco SPS sends LRQs sequentially and waits for a response in the RASTimeoutInterval time) and Blast (SPS sends LRQs in parallel before detecting any response).
- **RASLRQWindow**—Maximum time (in milliseconds) to wait for responses from the gatekeepers contacted. Default is 3000. The RIP message can override this value.
- **RASTimeToLive**—TTL (TimeToLive) value (in hops) in the RAS LRQ nonstandard message body. Default is 6.
- **RASAllowTranslation**—Sets the canMapAlias field in the LRQ message. Valid values are the following:
  - On—Set to True. The gatekeeper replaces the dialed phone number or destinationInfo field.
  - Off—Set to False. The gatekeeper does not replace address information.

If the gatekeeper replaces address information and the value of the canMapAlias field is False, the gatekeeper rejects the LRQ.

- **RASGateKeeperCluster**—Sets priorities for gatekeeper clusters so that Cisco SPS can query clusters in priority order. For each gatekeeper, you must specify the IP address and port number. Valid priority values are from 1 to 65535.

**Example**

The following example sets up two clusters, each containing two gatekeepers that use port 1719. Priority values are 1 and 2 respectively.

```
<RASGatekeeperCluster 1>
RASGatekeeper gatekeeper1.company.com 1719
RASGatekeeper gatekeeper2.company.com 1719
</RASGatekeeperCluster>

<RASGatekeeperCluster 2>
RASGatekeeper gatekeeper3.company.com 1719
RASGatekeeper gatekeeper4.company.com 1719
</RASGatekeeperCluster>
```

- **RASGateKeeper**—IP address (or FQDN) and port number of each individual gatekeeper in a gatekeeper cluster. Maximum number of gatekeepers in a cluster is 5.
- **RASDefaultTechPrefixAction**—Default action to take to an outgoing INVITE request or 302 message when a technology prefix exists and no specific local rule applies. Valid values are Strip (remove the prefix) and Include (include the prefix).
- **RASTechPrefix**—Technology prefix to use when the dialed number matches the specified number pattern.

**Example**

```
RASTechPrefix 1919321... 001# INCLUDE
RASTechPrefix 1919456... 002# STRIP
```

- **RASTransportType**—Transport type to use in the route entry that Cisco SPS learns from a gatekeeper via RAS. Valid values are TCP, TLS, and UDP. Default is UDP.
- **DebugFlag RasAPI**—Logs RasAPI debug messages to <ServerRoot>/logs/error\_log. These messages explain how Cisco SPS handles incoming and outgoing RasAPI messages that concern RAS encoding/decoding and sending/receiving. Valid values are On and Off. Default is Off.
- **DebugFlag RAS**—Logs RAS debug messages to <ServerRoot>/logs/error\_log. These messages explain how Cisco SPS handles incoming and outgoing RAS messages that concern module-specific tasks such as module configuration, socket creation, message conversion, and routes insertion. Valid values are On and Off. Default is Off.

**Step 3** Save and close the file.

---

## How to Configure the SIP Proxy Server in a Farm

You can configure registry and routing farms to contain different sets of members, as long as you ensure that configuration is consistent across all farm members.

For example, suppose that your registry contains two members (host1, host2) and the routing farm contains only the local member. You configure the `Cisco_Registry_Farm_Members` directive on each member and remove the comment marker for the `Cisco_Routing_Farm_Members` directive on each member.

**Note**

- Designate one farm member as master and the other as slave. Use the `sysadmin_csp_s_regroute` tool on the master to load any initial registry or routing seed files for the farm and to perform any registry or routing updates for the farm. Do not use the tool on a slave to seed or update registry or routing information. Initially, the master synchronizes any slaves. After that, members synchronize with each other whenever an update occurs.
- Synchronize the system clock between farm members. We recommend that you use Network Time Protocol (NTP) to do so. It provides accuracy within a millisecond on LANs and up to a few tens of milliseconds on WANs. For more information on NTP, refer to the Network Time Protocol Project website at <http://www.ntp.org/>.

**Detailed Steps**

**Step 1** Ensure that the IP network connecting farm members is working.

**Step 2** Edit the configuration file as follows:

- a. In the (Linux) `/usr/local/sip/conf/` or (Solaris) `/opt/sip/conf` directory, open the `sipd.conf` file using a text editor such as `vi`:

```
# vi sipd.conf
```

- b. Set the following directives:

- **Cisco\_Registry\_Farm\_Members**—Hostnames of all servers (excluding the local server)
- **Cisco\_Routing\_Farm\_Members**—Hostnames of all servers (excluding the local server)

**Example**

The following shows directive settings for each of two servers in a farm (`host1.cisco.com` and `host2.cisco.com`).

Member	Directive Setting
<b>Registry Farm</b>	
<code>host1.cisco.com</code>	<code>Cisco_Registry_Farm_Members "host2.cisco.com"</code>
<code>host2.cisco.com</code>	<code>Cisco_Registry_Farm_Members "host1.cisco.com"</code>
<b>Routing Farm</b>	
<code>host1.cisco.com</code>	<code>Cisco_Routing_Farm_Members "host2.cisco.com"</code>
<code>host2.cisco.com</code>	<code>Cisco_Routing_Farm_Members "host1.cisco.com"</code>

- c. Save and close the file.

**Step 3** Configure the master server:

- a. Prepare two seed data files for registry and routing separately for Cisco SPS.

**Note**

For additional information, see the template files `sip_registry.conf-dist` and `sip_routing.conf-dist`.

For upgrades from Cisco SPS 1.0 or Cisco SPS 1.1, copy static entries from `sipd.conf` to two separate files, one each for registry and routing.

- b. Use the `sysadmin_sps_regroute` tool to import the files, or use the provisioning-system GUI to seed the registry and routing entries.



---

**Note** Use seed data only to start the master member for the first time or when existing registry or routing databases are invalid. Seed data is not required between normal shutdown and start. A database is invalid when it contains corrupted data or when it has expired. If a database becomes get corrupted, do the following: 1. Stop Cisco SPS. 2. Remove the database files (`registry_db` or `routing_db` in the (Linux) `usr/local/sip/logs` or (Solaris) `opt/sip/logs` directory). 3. Start Cisco SPS. 4. Seed the database. If the database has expired, the farm member automatically clears its old database and gets the most current database from another farm member.

---

**Step 4** Configure slave servers as needed.



---

**Note** Do not place any seed entries in the default template files or `sipd.conf`, and do not use the `sysadmin_sps_regroute` tool on a slave server to update a slave database. For more information on slave servers, see the `sysadmin_sps_regroute` tool.

---

**Step 5** Synchronize the system clock among all farm members.

**Step 6** Periodically back up the registry and routing database files (use the `sysadmin_sps_regroute` tool to export their contents). You can then use the database files as seed data files if the databases later become corrupted.



---

**Note** For more information on system backups and restores, see the [“How to Back Up and Restore Cisco SPS” section on page 3-8.](#)

---

## How to Configure IPSec

You can configure IP security (IPSec) on your Cisco SPS.



---

**Note** You can configure IpSEC only if you are a system administrator.

---

IPSec is a suite of security protocols that secure and encrypt communication channels and ensure that only authorized parties can communicate on those channels. It enables you to restrict inbound and outbound communication on a port-by-port basis and to offer authenticated hosts different levels of access.

IPSec provides the following optional network-security services. In general, your local security policy determines which services you should use:

- Data confidentiality—The IPSec sender can encrypt packets before transmitting them across a network.
- Data integrity—The IPSec receiver can authenticate packets sent by the IPSec sender to ensure that the data has not been altered during transmission.

- Data-origin authentication—The IPSec receiver can authenticate the source of the IPSec packets sent. This service depends on the data integrity service.
- Anti-replay—The IPSec receiver can detect and reject replayed packets.



**Note** Data-origin authentication (also sometimes called data authentication or data integrity) is understood in this document to include anti-replay services, unless otherwise specified.

Cisco SPS IPSec is supported on network configurations as shown in [Table B-3](#).

**Table B-3 Network Configurations That Support Cisco SPS IPSec**

Network Configuration	Means of Support
Solaris system to Solaris system	Manual keying
Linux system to Linux system	Any of the following: <ul style="list-style-type: none"> <li>• Manual keying</li> <li>• IKE via configuration files</li> <li>• IKE via mod_ipsec_auto.c</li> </ul>
Solaris system to Linux system	Manual keying

Cisco SPS IPSec requires authentication and data encryption as follows:

- Authentication is installed as part of the Solaris Operating Environment 2.8.
- Data encryption is available on a Solaris supplemental CD or by download without charge from the Sun Solaris website at <http://www.sun.com/software/solaris/encryption/>.

## Detailed Steps

- 
- Step 1** Verify that data encryption is installed by checking to see if the following two files exist:
- /kernel/strmod/encrdes
  - /kernel/strmod/encr3des
- Step 2** Configure IPSec for Cisco SPS on a Solaris platform:
- a. Configure the system security policy.
  - b. Install the authentication and data-encryption security keys.
-

