Product Overview

This chapter provides an overview of the Cisco SIP Proxy Server (CSPS), its features, and prerequisites. This chapter contains the following sections:

- What is SIP?, page 1-1
- What is the Cisco SIP Proxy Server (CSPS)?, page 1-4
- CSPS Components, page 1-4
- CSPS Features, page 1-5
- Requirements, page 1-16
- Reference Links, page 1-17

Note

For installation updates of Release 2.0, refer to the README file.

What is SIP?

Session Initiation Protocol (SIP) is the Internet Engineering Task Force’s (IETF’s) standard for multimedia conferencing over IP. SIP is an ASCII-based, application-layer control protocol (defined in RFC 2543) that can be used to establish, maintain, and terminate calls between two or more end points. Like other VoIP protocols, SIP is designed to address the functions of signaling and session management within a packet telephony network. Signaling allows call information to be carried across network boundaries. Session management provides the ability to control the attributes of an end-to-end call.

SIP provides the following capabilities:

- Determine the location of the target end point—Supports address resolution, name mapping, and call redirection.
- Determine the media capabilities of the target end point—By using Session Description Protocol (SDP), SIP determines the highest level of common services between the end points. Conferences are established using only the media capabilities that can be supported by all end points.
- Determine the availability of the target end point—If a call cannot be completed because the target end point is unavailable, SIP determines whether the called party is already on the phone or did not answer in the allotted number of rings. It then returns a message indicating why the target end point is unavailable.
What is SIP?

- Establish a session between the originating and target end point—If the call can be completed, SIP establishes a session between the end points. SIP also supports mid-call changes, such as the addition of another end point to the conference or the changing of a media characteristic or codec.
- Handle the transfer and termination of calls—SIP supports the transfer of calls from one end point to another. During a call transfer, SIP simply establishes a session between the transferee and a new end point (specified by the transferring party) and terminates the session between the transferee and the transferring party. At the end of a call, SIP terminates the sessions between all parties.

Conferences can consist of two or more users and can be established using multicast or multiple unicast sessions.

Note

The term conference means an established session (or call) between two or more end points. In this document, the terms conference and call are used interchangeably.

Components of SIP

SIP is a peer-to-peer protocol. The peers in a session are called User Agents (UAs). A user agent can function in one of the following roles:

- User agent client (UAC)—A client application that initiates the SIP request.
- User agent server (UAS)—A server application that contacts the user when a SIP request is received and that returns a response on behalf of the user.

Typically, a SIP end point is capable of functioning as both a UAC and a UAS, but functions only as one or the other per transaction. Whether the endpoint functions as a UAC or a UAS depends on the UA that initiated the request.

From an architecture standpoint, the physical components of a SIP network can be grouped into two categories: clients and servers. Figure 1-1 illustrates the architecture of a SIP network.

Note

In addition, the SIP servers can interact with other application services, such as Lightweight Directory Access Protocol (LDAP) servers, location servers, a database application, or an extensible markup language (XML) application. These application services provide back-end services such as directory, authentication, and billing services.
SIP Clients

SIP clients include the following:

- Phones—Can act as either a UAS or UAC. Softphones (PCs that have phone capabilities installed) and Cisco SIP IP phones can initiate SIP requests and respond to requests.
- Gateways—Provide call control. Gateways provide many functionalities. The most common one is a translation function between SIP conferencing endpoints and other terminal types. This function includes translation between transmission formats and between communications procedures. In addition, the gateway also translates between audio and video codecs and performs call setup and clearing on both the LAN side and the switched-circuit network side.

SIP Servers

SIP servers include the following:

- Proxy server—The proxy server is an intermediate device that receives SIP requests from a client and then forwards the requests on the client’s behalf. Basically, proxy servers receive SIP messages and forward them to the next SIP server in the network. Proxy servers can provide functions such as authentication, authorization, network access control, routing, reliable request retransmission, and security.
- Redirect server—Provides the client with information about the next hop or hops that a message should take, then the client contacts the next hop server or UAS directly.
- Registrar server—Processes requests from UACs for registration of their current location. Registrar servers are often co-located with a redirect or proxy server.
What is the Cisco SIP Proxy Server (CSPS)?

The CSPS provides the primary capabilities required for call session management in a VoIP network and processes SIP requests and responses as described in RFC 2543. Powered by Apache, the CSPS can be configured to operate as a transaction stateful or stateless server. The CSPS can also be configured to provide additional server modes and features. For example, the CSPS can be configured with the following capabilities:

- Function as a redirect or registrar server
- Translate E.164 numbers to URL via location server protocols such as Telephone Number Mapping (ENUM)
- Perform gateway and Domain Name System (DNS) routing

Note

The CSPS includes software developed by the Apache Software Foundation (http://www.apache.org/).

CSPS Components

This section describes the main components in the CSPS and how they interact. See also Chapter 3, “Provisioning System GUI”.

- SIP Proxy Server (sipd)—handles all call processing and SIP messages. The provisioning client for sipd (spa) is installed if the CSPS GUI-based provisioning system is being used. SNMP is not automatically installed when sipd is installed. It must be manually installed as needed. See Cisco SIP Proxy Server Installation Guide.
- Provisioning Server (pserver)—main server used by the CSPS GUI-based provisioning system. A license manager (lm) component is automatically installed when the pserver is installed.
- MySQL—This database can be used to store and access provisioning system and subscriber feature data. If the provisioning system is installed, subscriber features (call forwarding and local authentication, not RADIUS) are automatically included since mysql already exists.
- Provisioning GUI client—provisioning client of the CSPS GUI-based provisioning system. It can be installed independent of the pserver. It requires installation of the correct version of JRE -1.3.1 (Java Runtime Environment 1.3.1).

The GUI retrieves and displays the current information in MySQL via the pserver. This information may be modified by the end user, prompting the GUI to send the updates to the pserver. If the information changed is licensing related, the license manager which resides with the pserver, does additional processing on this information. The pserver then updates the MySQL database with this information. Meanwhile, each spa (can be more than one on a multi-member farm) has registered for specific changes to the MySQL database. If a spa is notified of a change, it requests the pserver for the changed data and updates sipd.conf and/or routing and registry data. sipd can then use the new registry and routing entries immediately, and it uses the new sipd.conf if gracefully restarted.

Note

Changes to the MySQL database that spa does not register for are subscriber information such as call forwarding destinations and authentication passwords. sipd receives these information directly from MySQL.
CSPS Features

The CSPS offers the following features:

- Ability to function as a transaction stateful or stateless proxy server, stateful or stateless redirect server, and registrar server
- Call forwarding
- Choice of GUI based provisioning system or CLI tools for the following:
  - server configuration
  - embedded routing and registry databases
  - embedded subscriber database
- Ability to fork requests and distinguish spiralled requests from looped requests
- Address translation
  - Registry database (static and dynamic registry entries for contact points)
  - Gatekeeper Transaction Message Protocol (GKTMP) interface with Cisco NAM for 1-800, 1-900, and LNP lookups as well as least-cost routing
  - E.164 to URL address translation (via location server protocols such as ENUM and GKTMP)
  - Number expansion
- Next-hop routing
  - Static E.164 routes (dial plans)
  - Static domain routes
  - DNS SRV and DNS A record lookup and IP resolution
  - RAS module that allows communications between a CSPS and a H.323 directory gatekeeper
- Authentication and authorization via HTTP (Hypertext Transfer Protocol) Digest or HTTP Basic with a backend MySQL database, or via HTTP Digest, HTTP Basic or CHAP with a backend Radius server.
- Accounting via Radius
- Server farm support for sharing database information (registry and routing databases)
- Support for SIP over the following transport protocols:
  - User Datagram Protocol (UDP)
  - Transmission Control Protocol (TCP)
- Interoperability with Cisco SIP gateways, SIP IP phones, and unified messaging
- IP security (IPSec) for SIP signaling messages
- Access and error logging
- Support for rport and received parameters aiding in NAT traversal
- Domain-Specific registration, authentication, accounting, and subscriber database
- Pre-authorization query to a resource management system (RPMS)
- SNMP interface via CIAgent with basic platform MIBs for server status and start/stop/restart
**Server Modes**

The CSPS can be configured to function as one of the following types of servers:

- **Transaction stateful or stateless proxy server**
  When configured to be a stateful proxy server, the proxy server creates a transaction control block (TCB) and remembers incoming and outgoing requests, providing reliable retransmission of proxied requests and returning the best final response or responses back upstream. One transaction encompasses the received request, the request or requests (if forked) forwarded downstream, responses received from downstream hosts, and the best response returned upstream.

  When configured to be a stateless proxy server, the CSPS forgets all information once a request or response has been processed. As a stateless proxy server, the CSPS will just forward requests and responses.

  The CSPS also performs functions such as authentication, authorization, and access control.

- **Transaction stateful or stateless redirect server**
  As a redirect server, the CSPS accepts SIP requests, maps the address in the Request-URI to zero or more new addresses, and returns these addresses as Contacts in a SIP 3xx response to the UAC.

  When configured as a stateless redirect server, the CSPS does not create a TCB on receiving an INVITE request.

- **Registrar server**
  As a registrar server, the CSPS processes requests from UACs for registration of their current location. The CSPS also maintains registration information and can share this information with other registrar servers in its server farm. As a registrar server, the CSPS provides location services to the proxy server.

  **Note**
  In the CSPS, registrar servers must also be configured to function as either a proxy or redirect server. Standalone registrar servers are not supported.

For more information on configuring server modes, see Chapter 3, “Provisioning System GUI” and Chapter 4, “Configuring the Cisco SIP Proxy Server (CSPS)”.

**Protocol Support**

The CSPS supports the following network and transport protocols:

- **Internet Protocol (IP)**
  IP is a network layer protocol that sends datagram packets between nodes on the Internet. IP also provides features for addressing, type-of-service (ToS) specification, fragmentation and reassembly, and security. The CSPS supports IP as it is defined in RFC 791 for SIP signaling.

- **User Datagram Protocol (UDP)**
  UDP is a protocol that exchanges data packets without acknowledgments or guaranteed delivery. SIP can use UDP as the underlying transport protocol. With UDP, retransmissions are used to ensure reliability. The CSPS supports UDP as it is defined in RFC 768 for SIP signaling.

- **Transport Control Protocol (TCP)**
  TCP is a protocol that exchanges data packets with guaranteed delivery. The CSPS supports TCP as it is defined in RFC 793 for SIP signaling.

- **Domain Name System (DNS)**
DNS is used in the Internet for translating names of network nodes into addresses. SIP uses DNS to resolve the host names of endpoints to IP addresses. The resolution may include DNS SRV and DNS A records.

- Session Description Protocol (SDP)
  SDP is an ASCII-based protocol that describes multimedia sessions and their related scheduling information. The Cisco SIP IP phone uses SDP for session description.

**Spiralled and Loop Request Detection**

A spiralled request is a request that revisits the CSPS with a new request-uri and is considered a new transaction by the proxy. A looped request is a request that contains the proxy's via header, and the request-uri is the same as it was when it first visited the proxy.

At the proxy, this capability has the following effects:

- Allows the same call to be logged more than once
- May cause invocation of different features
- If Record-Route field in the INVITE message is enabled at the proxy, Record-Route procedure will be performed more than once.

**Note**
The Call-ID, From, To and CSeq.seqnum fields of a spiralled request are the same as that of the request which visited the proxy the first time.

**Address Translation, Routing, and IP Resolution**

The following steps are performed by the CSPS to deliver messages from endpoint to endpoint:

1. Translation—Process of translating an incoming Request-URI into an outgoing Request-URI.
2. Next hop routing—Process of obtaining a set of Fully Qualified Domain Names (FQDNs) or IP addresses with port numbers for each of the SIP entities found in the translation step. Among others, this step involves the following features:
   a. SRV lookup for static route—Process of a SRV lookup on the Static_Route_NextHop field if the Static_Route_NextHopPort field in a static route is not specified or is zero.
   b. RAS LRQ message transmission to gatekeeper—Process of sending the LRQ message to the H.323 gatekeeper to obtain the next hop gateway transport address.
3. IP resolution—Process of converting each next hop found in the next hop route lookup step into an IP address.

**Note**
If a Route header is present in the SIP request message, the CSPS translation and next hop routing functions are bypassed. Record-Route must be enabled on the server for subsequent requests to contain a route header.

**Translation**

During the translation step, the Request-URI of an incoming request is processed and a list of Contacts returned, each providing a URL to use in the outgoing request.
If Number Expansion is enabled, the global set of expansion rules is applied to the user portion of the relevant URLs for which the host portion is the CSPS. For REGISTER messages, this applies to the To, From, Contact, and optionally the Authorization headers. For INVITE messages, this applies to the Request URI, From, and optionally the Proxy-Authorization headers.

**Note**
The headers are not rewritten. The expanded versions are used internally for authentication, accounting, translation, and routing purposes.

The CSPS translation modules, in the order that they are called, are as follows.

- Call Forward Unconditional (mod_sip_call_forward)
- Registry (mod_sip_registry)
- ENUM (mod_sip_enum)
- GKTMP (mod_sip_gktmp)

The first module to return one or more Contacts completes the translation step, and the remaining modules are not called. For example, if the Registry module returns a Contact, then neither the ENUM or GKTMP module are called. If none of the translation modules returns a Contact, the core proxy module (mod_sip) returns a Contact based on the incoming Request-URI and that Request-URI is used in the next hop routing step.

**Next Hop Routing**
This step in SIP request processing is to determine the next hop route for each Contact. Next hop routing takes each translated Request-URI (Contact) and locates a set of next hop SIP entities capable of routing to the new Request-URI. This step involves two advanced features as follows.

**SRV Lookup for Static Route**—If the Static_Route_NextHopPort field of a static route is not specified or is zero, the CSPS performs an SRV lookup on the Static_Route_NextHop field.

**H.323 RAS module**—The RAS module allows communications between a SIP proxy server and a H.323 gatekeeper. The CSPS can send ASN.1 encoded RAS LRQ message to a H.323 gatekeeper which responds with a RAS LCF message. Figure 1-2 illustrates how the proxy server sends a LRQ H.225 RAS message to a gatekeeper which responds with a LCF message that can contain a gateway transport address.
Figure 1-2 CSPS and H.323 gatekeeper Message Processing

The RAS module supports the following:

**H.323 RIP (Request in Progress) message**—This message contains a time-delay value. Upon receipt of this message, the CSPS resets its timer and waits for the final response from the gatekeeper.

**Directory gatekeeper**—This gatekeeper forwards requests to the other gatekeepers and returns the highest priority and lowest cost gateway information to the requesting endpoint. Value of the TTL (time-to-live) field in the LRQ message must be greater than zero. The default is 6 hops.

**Sequential LRQ**—The CSPS can send LRQ requests sequentially based on the priority specified for the gatekeeper clusters in sipd.conf file. Configurable parameters include timeout value for each individual request, total LRQ request window, and whether to wait for the best LCF within the LRQ time window or to return the first valid LCF response to the proxy server.

**Blast LRQ**—The CSPS can send LRQ requests to all gatekeeper clusters before listening to the responses from the gatekeepers. It can also be configured to either take the first valid LCF response or wait for the best response within the LRQ time window.

**Multi-alternate endpoint**—If the selected LCF from a gatekeeper contains alternate endpoints, the CSPS can parse these endpoints and store them in the route table. The routes in the alternate endpoints have lower priority than the route for the primary endpoint.

**Redundant gatekeeper**—This can be configured by maintaining a list of prioritized gatekeeper clusters in the CSPS configuration. Once this is configured, the following trial process occurs.

The proxy server tries one of the gatekeepers randomly in the cluster which has the highest priority. If this trial expires the timeout value specified in the RASTimeoutInterval directive, the proxy server tries the next gatekeeper in the same cluster in a round-robin manner. If LRJ message is received by the proxy server, it tries a gatekeeper in the next highest priority cluster and so forth. Figure 1-3 illustrates the configuration.
Tech prefix—The CSPS prepends the technology prefix (one, two or three digits + ‘#’) to the expanded dialed number to the LRQ request. It can also be included in the Request-URI of the INVITE message forwarded to the SIP/H.323 gateway. Multiple technology prefixes can be configured in the sipd.conf file based on the expanded dialed number.

See directive RASTechPrefix in “Configuring E.164 to Request-URI Address Translation” section on page 4-18 and Chapter 3, “Provisioning System GUI”.

Pavo extension—The CSPS can include Pavo extensions (CallIdentifier, RedirectIEInfo, CallingOctet3a) in the LRQ from the CC-Diversion header of the incoming INVITE message.

To locate the next hop SIP entities, the following actions occur:

- If the host portion of the new Request-URI is the address (FQDN or IP address) of the server itself, the next hop routing is performed, using the user portion of the Request-URI.
  
  E.164 routing is an implementation of this method.

- If the Static_Route_NextHopPort field of a static route is not specified or has the value of 0, the CSPS tries to do SRV lookup on the Static_Route_NextHop field.

  If this lookup is successful, the algorithm outlined in RFC 2782 is used to select one destination. If this lookup fails, alternate destinations will be tried and the proxy will do simple DNS ‘A’ lookup on the Static_Route_NextHop field of the static route. This field should contain a name for the ‘A’ lookup.

- If no route is found in the E.164 routing, RAS LRQ message is sent by the CSPS to the H.323 gatekeeper cluster which is identified to have the highest priority gateway.

  According to configurations and availability of the gatekeepers being contacted, one of the following messages is returned before the time limit in the configured timeout window expires.

  - LCF (Location Confirm)
  - LRJ (Location Reject)
  - RIP (Response in Progress)

  The RIP message is activated when the gatekeeper cluster is connected to a remote gatekeeper (by UDP). This is related to the number of hops to be attempted. See directive RASTimeToLive in “Configuring E.164 to Request-URI Address Translation” section on page 4-18 and Chapter 3, “Provisioning System GUI”.

- If the host portion of the new Request-URI is NOT the address (FQDN or IP address) of the server itself, domain routing is performed using the host portion of the Request-URI.
Domain routing and next hop routing via DNS SRV are implementations of this method.

**IP Resolution**

This step in processing SIP requests is to convert each next hop found via next hop routing into an IP address. Standard IP resolution (via gethostbyname) is performed via DNS, NIS, NIS+, or host file, depending on the IP resolution subsystem configured on the system where CSPS is located.

For more information on configuring directives pertaining to address translation and routing, see “Configuring E.164 to Request-URI Address Translation” section on page 4-18 and Chapter 3, “Provisioning System GUI”.

**Authentication, Authorization**

The Cisco SIP Server can be configured to provide authentication and authorization. The authentication can occur at a Radius Server or at the Proxy. Three types of authentication are supported by the Proxy in conjunction with the appropriate Radius Server.

- CHAP-Password Authentication
- HTTP Digest Authentication
- HTTP Basic Authentication

Only HTTP Digest Authentication and HTTP Basic Authentication are supported by the Proxy (without a Radius Server and via a MySQL database).

HTTP Digest Authentication and HTTP Basic Authentication are supported by the Proxy.

Radius is an IETF protocol based on UDP. It functions by exchanging a set of attribute/value pairs between the client and server. For example, a CSPS acting as a Radius client exchanges attribute/value pairs with a Radius Server to provide authentication.

For Radius supported authentication, the UAC password is stored at the Radius Server. For Proxy supported authentication, the UAC password is stored in a subscriber table in a MySQL database.

**Authentication**

The default authentication scheme is the HTTP Digest Authentication performed at the CSPS. HTTP Digest Authentication and HTTP Basic Authentication are performed as described in RFC 2617.

When Digest Authentication and Basic Authentication are used at the Proxy, the Username as found in the Authorization Header or the Proxy-Authorization Header is the key to query the MySQL database.

If the authentication takes place at the Radius Server, the Username is passed as one of the attribute/value pairs from the CSPS to the Radius Server where it can be used to key the User Search, before performing authentication. Additionally, CSPS can be configured to add any desired SIP headers as VSAs in the authentication request to the Radius Server.

The UserName may be expanded by the CSPS before performing the MySQL lookup, or before it is passed to the Radius Server. This enables phone numbers to be expanded to fully expanded E.164 numbers before processing.

If the virtual proxy host feature is turned on, the Username@domain (Username found in the Authorization/proxy-Authorization header and the domain name found in the From header) is the key used to query the MySQL database. This the same with authentication performed on a radius server and the RadiusUserNameAttrAddDomain directive turned on. Username@domain is passed as one of the attribute/value pairs from the CSPS to the Radius Server where it can be used as the key for the user search feature.
CSPS Features

Note

Native Apache based virtual host refers to providing the allusion of more than one server on one system, as differentiated by their hostname. For example, companies sharing a web server can have their own domains (www.company1.com and www.company2.com) and access to the web server. Native Apache based virtual hosts are not supported in CSPS.

The CSPS provides access control lists and user authentication which can be combined to provide more complex access control. Based on Apache access control mechanism, some SIP devices such as SIP gateways, do not authenticate themselves. They may be specified in the access control list, and instruct the CSPS to accept INVITEs from them.

REGISTERs and INVITEs from other devices may be challenged by the CSPS for user authentication. This is configurable.

Authorization

An authenticated user is authorized. Currently, there is no authorization of specific user capabilities for the service provider voice applications.

Accounting

The CSPS uses basic start-stop records with a combination of standard attributes and Cisco Vendor Specific Attributes (VSAs). The server can be configured to provide accounting services via a Radius server. When accounting is enabled, Call Detail Records (CDR) generated during calls are logged on a Radius accounting server as described in RFC 2159.

Accounting uses standard RADIUS attributes where possible. Additionally, CSPS can be configured to add any desired SIP headers as VSAs in the accounting requests. When the accounting services are enabled and the interface to the Radius server is configured, the CSPS creates and sends the Accounting records to the Radius server described as follows.

- The start record is written when a 200 OK is received by the CSPS for an INVITE.
- The stop record is written when BYE is received by the CSPS.

Note

The CSPS receives the BYE only if Record-Route is enabled. See Configuring E.164 to Request-URI Address Translation, page 4-18 and Chapter 3, “Provisioning System GUI”.

Server Farm Support

A server farm is a group of CSPSs that share database information, provide redundancy and high availability. When the members of a farm are defined, all updates to a member database are replicated across every member of the farm. During replication, any member of the farm that is out of synchronization with the other members, will be resynchronized.

By default, a farm is created for the routing and registry databases. To configure a farm, the names of the members of the farm and the ports at which to lock and update the farm members is required.

Database information is shared by mirroring all updates to databases for all members of the farm and by keeping the databases synchronized.
Registrar for Multiple Domain

The CSPS can act as a registrar for multiple domains. These domains can be a single domain (ProxyDomain) for fully expanded E.164 numbers, and zero or more additional domains within which private name spaces can exist.

For example, the E.164 domain can be cisco.com. The additional private domains can be a.com and b.com. The CSPS accepts REGISTER messages for the following domains:

- <*>@a.com
- <*>@b.com
- <*>@cisco.com
- <e.164-number>@cisco.com
- <e.164-number>@a.com
- <e.164-number>@b.com

If an INVITE is received for <e.164-number>.*.com, it is treated as an INVITE for <e.164-number>@cisco.com. All registrations for <e.164-number>.*.com are represented by a single entry in the registration database. If an INVITE is received for user@a.com, or user@b.com, or user@cisco.com, it is treated as an INVITE for user@a.com, or user@b.com, or user@cisco.com, respectively. Registrations for user@a.com, user@b.com, and user@cisco.com result in three separate entries in the registry database.

The creation and administration of multiple domains of overlapping name spaces is a coordinated effort among the administrator of the CSPS and the administrators of the domains the CSPS supports. For example, CompanyA and CompanyB are in area code 408. CompanyA may use 4-digit extensions 2000-7999 and CompanyB may also use 4-digit extensions, 2000-7999. There are no DIDs associated with these extensions. For users at CompanyA and CompanyB who need DIDs, CompanyA gets 1-408-555-[8000-8999] and CompanyB gets 1-408-666-[8000-8999].

Registration occurs as follows.

- 2xxx at CompanyA registers as 2xxx@companyA.com
- 2xxx at CompanyB registers as 2xxx@companyB.com
- 8xxx at CompanyA registers as +14085558xxx@companyA.com
- 8xxx at CompanyB registers as +14086668xxx@companyB.com

Note

xxx can be replaced by any number. For example, 2xxx can be 2000, 2001, 2999, etc.

Call processing occurs as follows.

2001 calls 2000

- 2001 at CompanyA calls 2000 at CompanyA as 2000@companyA.com [1]
- 2001 at CompanyA calls 2000 at CompanyB as 2000@companyB.com [2]
- 2001 at CompanyB calls 2000 at CompanyB as 2000@companyB.com [1]
- 2001 at CompanyB calls 2000 at CompanyA as 2000@companyA.com [2]
8001 calls 2000
- 8001 at CompanyA calls 2000 at CompanyA as 2000@companyA.com [1]
- 8001 at CompanyA calls 2000 at CompanyB as 2000@companyB.com [2]
- 8001 at CompanyB calls 2000 at CompanyB as 2000@companyB.com [1]
- 8001 at CompanyB calls 2000 at CompanyA as 2000@companyA.com [2]

2001 calls 8000
- 2001 at CompanyA calls 8000 at CompanyA as +14085558000@companyA.com[3]
- 2001 at CompanyA calls 8000 at CompanyB as +14086668000@companyA.com[4]
- 2001 at CompanyB calls 8000 at CompanyB as +14086668000@companyB.com[3]
- 2001 at CompanyB calls 8000 at CompanyA as +14085558000@companyB.com[4]

8001 calls 8000
- 8001 at CompanyA calls 8000 at CompanyA as +14085558000@companyA.com[3]
- 8001 at CompanyA calls 8000 at CompanyB as +14086668000@companyA.com[4]
- 8001 at CompanyB calls 8000 at CompanyB as +14086668000@companyB.com[3]
- 8001 at CompanyB calls 8000 at CompanyA as +14085558000@companyB.com[4]

2000 and 8000 call 18183635839
- 2000 at CompanyA calls 18183635839 as +18183635839@companyA.com [5]
- 8000 at CompanyA calls 18183635839 as +18183635839@companyA.com [5]
- 2000 at CompanyB calls 18183635839 as +18183635839@companyB.com [5]
- 8000 at CompanyB calls 18183635839 as +18183635839@companyB.com [5]

John Doe, who is not with CompanyA or CompanyB, but is an authorized user, places calls as follows.
- Calls 2000 at CompanyA as 2000@companyA.com [2]
- Calls 2000 at CompanyB as 2000@companyB.com [2]
- Calls 8000 at CompanyA as +14085558000@<proxy>.com [6]
- Calls 8000 at CompanyB as +14086668000@<proxy>.com [6]
Note

[1] The phones at both companies expand 2xxx to 2xxx@<proxy>

[2] This is done via full URL dialing

[3] The phones at company A expand 8xxx to +14085558xxx@<proxy> the phones at company B expand 8xxx to +140866668xxx@<proxy>

[4] The phones at both companies expand [2-9]xxxxx to +1408xxxxxxx@<proxy>. All authenticated users, regardless of domain, have access to the same routes for forwarding calls for +1408xxxxxxx

[5] The phones at both companies expand 1xxxxxxxxx to +1xxxxxxxxxx@<proxy>. All authenticated users, regardless of domain, have access to the same routes for forwarding calls for +1xxxxxxxxxx

[6] CSPS supports a default domain for e.164 numbers. These are designated by the plus (+) sign. Refer to Figure 1-4 for detail.

Figure 1-4 Multiple Domains Registration
IP Security

IP Security (IPSec) provides security for transmission of sensitive information over unprotected networks such as the Internet. IPSec acts at the network layer, protecting and authenticating IP packets between participating IPSec devices (peers), such as the CSPS and a UAC, SIP gateway, or another CSPS. With IPSec, data can be transmitted across a public network without fear of observation, modification, or spoofing.

All IPSec combinations have been successfully tested and proven to secure traffic to and from the CSPS on the following platforms:

- Solaris 2.8 Operating Environment (via manually-keyed key management method)
- Redhat 7.1 with Linux FreeS/WAN Release 1.5 (via manually keyed and Internet Key Exchange (IKE) key management methods)

Access and Error Logging

The CSPS uses the standard Apache logging functionality in addition to a SIP-specific logging functionality. The standard Apache logging functionality is defined for the CSPS as a whole, while the SIP logging functionality can be configured on a per-module basis.

By default, internal CSPS errors are logged to /usr/local/sip/logs/error_log (on Linux) and /opt/sip/logs/error_log (on Solaris). Access records are logged to /usr/local/sip/logs/access_log (on Linux) and /opt/sip/logs/access_log (on Solaris). For information on interpreting the information in the log files, and customizing the type and amount of information they include, see Chapter 3, “Provisioning System GUI,” Chapter 4, “Configuring the Cisco SIP Proxy Server (CSPS),” and Chapter 5, “Maintaining the Cisco SIP Proxy Server (CSPS).”

Log rotation and size are per standard apache. For information on Apache and Apache logs, see URL www.apache.org.

Requirements

The CSPS can be run on a Solaris or Linux platform. The following sections list the minimum hardware and software requirements for each platform. Before installing and configuring the CSPS, one of the following sets of requirements must be met.

For a list of Cisco supported platforms, see the Cisco SIP Proxy Server (CSPS) Release Note.

Solaris Platform Requirements

For Solaris, the following is required:

- Workstation—Sparc or UltraSparc server class system with a minimum of 256 MB of RAM and 1 GB of disk space.
- Solaris 2.8
- IPSec encryption requires the installation of supplemental encryption software. Additional details and free downloads of the software are available at: http://wwws.sun.com/software/solaris/encryption/
Chapter 1      Product Overview

Linux Platform Requirements

For Linux, the following is required:

- PC—Intel Pentium III processor operating with a minimum of 128 MB of RAM and 1 GB of disk space.
- Redhat 7.1
- For IPSec, Linux FreeS/WAN with Redhat Linux (See URL www.freeswan.org for the latest implementation of Linux FreeS/WAN IPSec and its targeted Redhat Linux version).

Reference Links

The following is a list of URLs on companies and additional information on technologies referenced within the CSPS software and documentation:

- http://www.apache.org
- http://www.mysql.com
- http://www.sun.com
- http://www.linux.org
- http://www.redhat.com
- http://www.freeswan.org
- http://wwws.sun.com