

Firewall Support of Skinny Client Control Protocol

The Firewall Support of Skinny Client Control Protocol feature enables the Cisco IOS XE firewall to support VoIP and the Skinny Client Control Protocol (SCCP). Cisco IP phones use the SCCP to connect with and register to Cisco Unified Communications Manager. To be able to configure Cisco IOS XE firewall between the IP phone and Cisco Unified Communications Manager in a scalable environment, the firewall needs to be able to detect SCCP and understand the information passed within the messages. With the Firewall Support of Skinny Client Control Protocol feature, the firewall inspects Skinny control packets that are exchanged between Skinny clients (such as IP Phones) and the Cisco Unified Communications Manager and configures the router to enable Skinny data channels to traverse through the router. This feature extends the support of SCCP to accommodate video channels.

- Prerequisites for Firewall Support of Skinny Client Control Protocol, on page 1
- Restrictions for Firewall Support of Skinny Client Control Protocol, on page 2
- Information About Firewall Support of Skinny Client Control Protocol, on page 2
- How to Configure Firewall Support of Skinny Client Control Protocol, on page 4
- Configuration Examples for Firewall Support of Skinny Control Protocol, on page 8
- Additional References for Firewall Support of Skinny Client Control Protocol, on page 9
- Feature Information for Firewall Support for Skinny Client Control Protocol, on page 10

Prerequisites for Firewall Support of Skinny Client Control Protocol

- Your system must be running Cisco IOS XE Release 2.1 or a later release.
- You must enable the firewall for the SCCP application-level gateway (ALG) to work.
- You must enable the TFTP ALG for SCCP to work because IP phones that use Skinny need the TFTP configuration file from the Cisco Unified Communications Manager.

Restrictions for Firewall Support of Skinny Client Control Protocol

- IPv6 address inspection and translation is not supported.
- TCP segmentation is not supported.

Information About Firewall Support of Skinny Client Control Protocol

Application-Level Gateways

An application-level gateway (ALG), also known as an application-layer gateway, is an application that translates the IP address information inside the payload of an application packet. An ALG is used to interpret the application-layer protocol and perform firewall and Network Address Translation (NAT) actions. These actions can be one or more of the following depending on your configuration of the firewall and NAT:

- Allow client applications to use dynamic TCP or UDP ports to communicate with the server application.
- Recognize application-specific commands and offer granular security control over them.
- Synchronize multiple streams or sessions of data between two hosts that are exchanging data.
- Translate the network-layer address information that is available in the application payload.

The firewall opens a pinhole, and NAT performs translation service on any TCP or UDP traffic that does not carry the source and destination IP addresses in the application-layer data stream. Specific protocols or applications that embed IP address information require the support of an ALG.

SCCP Inspection Overview

SCCP inspection enables voice communication between two SCCP clients by using the Cisco Unified Communications Manager. The Cisco Unified Communications Manager uses the TCP port 2000 (the default SCCP port) to provide services to SCCP clients. Initially, the SCCP client connects to the primary Cisco Unified Communications Manager by establishing a TCP connection and, if available, connects to a secondary Cisco Unified Communications Manager. After the TCP connection is established, the SCCP client registers with the primary Cisco Unified Communications Manager, which is used as the controlling Cisco Unified Communications Manager until it reboots or a keepalive failure occurs. Thus, the TCP connection between the SCCP client and the Cisco Unified Communications Manager exists forever and is used to establish calls coming to or from the client. If a TCP connection fails, the secondary Cisco Unified Communications Manager is used. All data channels established with the initial Cisco Unified Communications Manager remain active and will be closed after the call ends.

The SCCP protocol inspects the locally generated or terminated SCCP control channels and opens or closes pinholes for media channels that originate from or are destined to the firewall. Pinholes are ports that are opened through a firewall to allow an application controlled access to a protected network.

The table below lists the set of messages that are necessary for the data sessions to open and close. SCCP inspection will examine the data sessions that are used for opening and closing the access list pinholes.

Table 1: SCCP Data Session Messages

Skinny Inspection Message	Description
CloseReceiveChannel	Indicates that the call should be aborted. Any intermediate sessions created by the firewall and NAT have to be cleaned up when this message is received.
OpenReceiveChannelACK	Indicates that the phone is acknowledging the OpenReceiveChannel message that it received from the Cisco Unified Communications Manager.
StartMediaTransmission	Contains the Realtime Transport Protocol (RTP) information of the phone that is the source or destination of the call. The message contains the IP address, the RTP port that the other phone is listening on, and the Call ID that uniquely identifies the call.
StopMediaTransmission	Indicates that the call has ended. Sessions can be cleaned up after receiving this message.
StationCloseReceiveChannel	Instructs the Skinny client (on the basis of the information in this message) to close the receiving channel.
StationOpenMultiMediaReceiveChannelAck	Contains the IP address and port information of the Skinny client sending this message. It also contains the status of whether the client is willing to receive video and data channels.
StationOpenReceiveChannelAck	Contains the IP address and port information of the Skinny client sending this message. This message also contains the status of whether or not the client is willing to receive voice traffic.
StationStartMediaTransmission	Contains the IP address and port information of the remote Skinny client.
StationStartMultiMediaTransmit	Indicates that the Cisco Unified Communications Manager received an OpenLogicalChannelAck message for the video or the data channel.
StationStopMediaTransmission	Instructs the Skinny client (on the basis of the information in this message) to stop transmitting voice traffic.
StationStopSessionTransmission	Instructs the Skinny client (on the basis of the information in this message) to end the specified session.

ALG--SCCP Version 17 Support

The ALG—SCCP Version 17 Support feature enables the SCCP ALG to parse SCCP Version 17 packets. Cisco Unified Communications Manager 7.0 and the IP phones that use Cisco Unified Communications

Manager 7.0 support only SCCP Version 17 messages. The format of SCCP changed from Version 17 to support IPv6. The SCCP ALG checks for the SCCP version in the prefix of a message before parsing it according to the version. The SCCP message version is extracted from the message header and if it is greater than Version 17, the message is parsed by using the Version 17 format and the IPv4 address and port information is extracted. The SCCP ALG supports the inspection and translation of IPv4 address information in SCCP messages.

Note IPv6 address inspection and translation are not supported.

The IP address format of the following SCCP ALG-handled messages changed in Version 17:

- StationOpenMultiMediaReceiveChannelAck
- StationOpenReceiveChannelAckMessage
- StationRegisterMessage
- StationStartMediaTransmissionAckMessage
- StationStartMultiMediaTransmissionAckMessage
- StationStartMediaTransmissionMessage
- StationStartMultiMediaTransmissionMessage

How to Configure Firewall Support of Skinny Client Control Protocol

Configuring a Skinny Class Map and Policy Map

When you enable SCCP (through the **match protocol** command) in a firewall configuration, you must enable TFTP (through the **match protocol** command); otherwise, the IP phones that use SCCP cannot communicate with the Cisco Unified Communications Manager. SCCP enables voice communication between two Skinny clients through the use of a Cisco Unified Communications Manager.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect match-any** *class-map-name*
4. **match protocol** *protocol-name*
5. **match protocol** *protocol-name*
6. **exit**
7. **policy-map type inspect** *policy-map-name*
8. **class type inspect** *class-map-name*
9. **inspect**
10. **exit**

11. **class class-default**
12. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map type inspect match-any** *class-map-name* **Example:** `Router(config)# class-map type inspect match-any cmap1`	Creates an inspect type class map and enters class map configuration mode.
Step 4	**match protocol** *protocol-name* **Example:** `Router(config-cmap)# match protocol skinny`	Configures the match criterion for a Skinny class map.
Step 5	**match protocol** *protocol-name* **Example:** `Router(config-cmap)# match protocol tftp`	Configures the match criterion for a TFTP class map.
Step 6	**exit** **Example:** `Router(config-cmap)# exit`	Exits class map configuration mode.
Step 7	**policy-map type inspect** *policy-map-name* **Example:** `Router(config)# policy-map type inspect pmap1`	Creates an inspect type policy map and enters policy map configuration mode.
Step 8	**class type inspect** *class-map-name* **Example:** `Router(config-pmap)# class type inspect cmap1`	Specifies the class on which the action is performed and enters policy-map class configuration mode.
Step 9	**inspect** **Example:** `Router(config-pmap-c)# inspect`	Enables stateful packet inspection.

	Command or Action	Purpose
Step 10	**exit** **Example:** `Router(config-pmap-c)# exit`	Exits policy-map class configuration mode and enters policy map configuration mode.
Step 11	**class class-default** **Example:** `Router(config-pmap)# class class-default`	Specifies that these policy map settings apply to the predefined default class. • If traffic does not match any of the match criteria in the configured class maps, it is directed to the predefined default class.
Step 12	**end** **Example:** `Router(config-pmap)# end`	Exits policy map configuration mode and enters privileged EXEC mode.

Configuring a Zone Pair and Attaching an SCCP Policy Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** {*zone-name* | **default**}
4. **exit**
5. **zone security** {*zone-name* | **default**}
6. **exit**
7. **zone-pair security** *zone-pair-name* [**source** {*source-zone-name* | **self** | **default**} **destination** [*destination-zone-name* | **self** | **default**]]
8. **service-policy type inspect** *policy-map-name*
9. **exit**
10. **interface** *type number*
11. **zone-member security** *zone-name*
12. **exit**
13. **interface** *type number*
14. **zone-member security** *zone-name*
15. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**zone security** {*zone-name* | **default**} **Example:** `Router(config)# zone security zone1`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode.
Step 4	**exit** **Example:** `Router(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 5	**zone security** {*zone-name* | **default**} **Example:** `Router(config)# zone security zone2`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode.
Step 6	**exit** **Example:** `Router(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 7	**zone-pair security** *zone-pair-name* [**source** {*source-zone-name* | **self** | **default**} **destination** [*destination-zone-name* | **self** | **default**]] **Example:** `Router(config)# zone-pair security in-out source zone1 destination zone2`	Creates a zone pair and enters security zone pair configuration mode. **Note** To apply a policy, you must configure a zone pair.
Step 8	**service-policy type inspect** *policy-map-name* **Example:** `Router(config-sec-zone-pair)# service-policy type inspect pmap1`	Attaches a firewall policy map to the destination zone pair. **Note** If a policy is not configured between a pair of zones, traffic is dropped by default.
Step 9	**exit** **Example:** `Router(config-sec-zone-pair)# exit`	Exits security zone-pair configuration mode and enters global configuration mode.
Step 10	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/0`	Configures an interface and enters interface configuration mode.
Step 11	**zone-member security** *zone-name* **Example:** `Router(config-if)# zone-member security zone1`	Assigns an interface to a specified security zone. **Note** When you make an interface a member of a security zone, all traffic in and out of that interface (except traffic bound

	Command or Action	Purpose
		for the router or initiated by the router) is dropped by default. To let traffic through the interface, you must make the zone part of a zone pair to which you apply a policy. If the policy permits traffic, traffic can flow through that interface.
Step 12	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 13	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/1/1`	Configures an interface and enters interface configuration mode.
Step 14	**zone-member security** *zone-name* **Example:** `Router(config-if)# zone-member security zone2`	Assigns an interface to a specified security zone.
Step 15	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.

Configuration Examples for Firewall Support of Skinny Control Protocol

Example: Configuring an SCCP Class Map and a Policy Map

```
Router# configure terminal
Router(config)# class-map type inspect match-any cmap1
Router(config-cmap)# match protocol skinny
Router(config-cmap)# match protocol tftp
Router(config-cmap)# exit
Router(config)# policy-map type inspect pmap1
Router(config-pmap)# class type inspect cmap1
Router(config-pmap-c)# inspect
Router(config-pmap-c)# exit
Router(config-pmap)# class class-default
Router(config-pmap)# end
```

Example: Configuring a Zone Pair and Attaching an SCCP Policy Map

```
Router# configure terminal
Router(config)# zone security zone1
Router(config-sec-zone)# exit
Router(config)# zone security zone2
Router(config-sec-zone)# exit
Router(config)# zone-pair security in-out source zone1 destination zone2
Router(config-sec-zone-pair)# service-policy type inspect pmap1
Router(config-sec-zone-pair)# exit
Router(config)# interface gigabitethernet 0/0/0
Router(config-if)# zone-member security zone1
Router(config-if)# exit
Router(config)# interface gigabitethernet 0/1/1
Router(config-if)# zone-member security zone2
Router(config-if)# end
```

Additional References for Firewall Support of Skinny Client Control Protocol

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Firewall Support for Skinny Client Control Protocol

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 2: Feature Information for Firewall Support for Skinny Client Control Protocol

Feature Name	Releases	Feature Information
ALG—SCCP V17 Support	Cisco IOS XE Release 3.5S	The ALG—SCCP Version 17 Support feature enables the SCCP ALG to parse SCCP version 17 packets. The SCCP format has changed from version 17 to support IPv6.
Firewall—SCCP Video ALG Support	Cisco IOS XE Release 2.4	SCCP enables voice communication between two Skinny clients through the use of a Cisco Unified Communications Manager. This feature enables Cisco firewalls to inspect Skinny control packets that are exchanged between a Skinny client and the Cisco Unified Communications Manager. The following command was modified: **match protocol**.

Feature Name	Releases	Feature Information
Firewall Support for Skinny Client Control Protocol	Cisco IOS XE Release 2.1	The Firewall Support of Skinny Client Control Protocol feature enables the Cisco IOS XE firewall to support VoIP and SCCP. Cisco IP phones use the SCCP to connect with and register to Cisco Unified Communications Manager. To be able to configure Cisco IOS XE firewall between the IP phone and Cisco Unified Communications Manager in a scalable environment, the firewall needs to be able to detect SCCP and understand the information passed within the messages. With the Firewall Support of Skinny Client Control Protocol feature, the firewall inspects Skinny control packets that are exchanged between Skinny clients (such as IP Phones) and the Cisco Unified Communications Manager and configures the router to enable Skinny data channels to traverse through the router. This feature extends the support of SCCP to accommodate video channels..