



Secure Socket Layer HTTP

- [Feature history for Secure Socket Layer HTTP, on page 1](#)
- [Information about Secure Socket Layer HTTP, on page 2](#)
- [How to configure Secure Socket Layer HTTP, on page 6](#)
- [Monitoring secure HTTP server and client status, on page 13](#)
- [Additional references for Secure Socket Layer HTTP, on page 13](#)

Feature history for Secure Socket Layer HTTP

This table provides release and related information for the features explained in this module. These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Table 1: Feature history for Secure Socket Layer HTTP

Release	Feature	Feature information
Cisco IOS XE 26.2.1	Secure Socket Layer HTTP: HTTP with SSL encryption provides a secure connection to allow such functions as configuring a switch from a web browser..	Cisco C9610 Series Smart Switches
Cisco IOS XE Everest 16.5.1a Cisco IOS XE Everest 16.6.1	Secure Socket Layer HTTP: The Secure Socket Layer HTTP feature has been introduced.	Support for this feature was introduced on the C9500-12Q, C9500-16X, C9500-24Q, and C9500-40X models of the Cisco Catalyst 9500 Series Switches.
Cisco IOS XE Fuji 16.8.1a Cisco IOS XE Fuji 16.9.2 Cisco IOS XE Gibraltar 16.11.1	Secure Socket Layer HTTP: The Secure Socket Layer HTTP feature has been introduced.	Support for this feature was introduced on the C9500-24Y4C, C9500-32C, C9500-32QC, and C9500-48Y4C models of the Cisco Catalyst 9500 Series Switches.

Release	Feature	Feature information
Cisco IOS XE Cupertino 17.7.1	Secure Socket Layer HTTP: The Secure Socket Layer HTTP feature has been introduced.	This feature was implemented on supervisor modules C9400X-SUP-2 and C9400X-SUP-2XL. This feature was implemented on the C9500X-28C8D model. This feature was implemented on Cisco Catalyst 9600 Series Supervisor 2 Module (C9600X-SUP-2).
Cisco IOS XE Cupertino 17.9.1	Secure Socket Layer HTTP: The Secure Socket Layer HTTP feature has been introduced.	This feature was implemented on the C9200CX-12P-2X2G, C9200CX-8P-2X2G, and C9200CX-12T-2X2G models.
Cisco IOS XE Dublin 17.10.1b	Secure Socket Layer HTTP: The Secure Socket Layer HTTP feature uses the secure HTTP server and secure HTTP client uses an implementation of SSL Version 3.0 with application-layer encryption. On a secure HTTP connection, data to and from an HTTP server is encrypted before being sent over the Internet.	This feature was implemented on the C9500X-60L4D model.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to <https://cfng.cisco.com/>.

Information about Secure Socket Layer HTTP

This section describes the concepts required to understand Secure Socket Layer HTTP on Cisco devices, including the roles of the secure HTTP server and client, certificate authority trustpoints, CipherSuites, and configuration guidelines.

Secure HTTP servers and clients overview

HTTP with SSL encryption provides a secure connection to allow such functions as configuring a switch from a web browser. HTTP over SSL is abbreviated as HTTPS; the URL of a secure connection begins with https:// instead of http://.

Roles of the secure HTTP server and client



Note Certain device configurations and protocols are identified as insecure and pose a security risk to Cisco devices. While existing deployments continue to function, you must intentionally enable these features for new installations. For more information on remediation, refer to [Cisco C9000 Switching IOSXE – Resilient Infrastructure Playbook](#).

The Cisco implementation of the secure HTTP server and secure HTTP client uses SSL Version 3.0 with application-layer encryption. On a secure HTTP connection, data to and from an HTTP server is encrypted before being sent over the Internet.

The primary role of the HTTP secure server (the switch) is to listen for HTTPS requests on a designated port (the default HTTPS port is 443) and pass the request to the HTTP 1.1 Web server. The HTTP 1.1 server processes requests and passes responses (pages) back to the HTTP secure server, which, in turn, responds to the original request.

The primary role of the HTTP secure client (the web browser) is to respond to Cisco IOS application requests for HTTPS User Agent services, perform HTTPS User Agent services for the application, and pass the response back to the application.

Certificate authority trustpoints

Certificate authorities (CAs) manage certificate requests and issue certificates to participating network devices. These services provide centralized security key and certificate management for the participating devices. Specific CA servers are referred to as trustpoints.

How trustpoints authenticate secure HTTP connections

When a connection attempt is made, the HTTPS server provides a secure connection by issuing a certified X.509v3 certificate, obtained from a specified CA trustpoint, to the client. The client (usually a web browser), in turn, has a public key that allows it to authenticate the certificate.

For secure HTTP connections, we highly recommend that you configure a CA trustpoint. If a CA trustpoint is not configured for the device running the HTTPS server, the server certifies itself and generates the needed RSA key pair. Because a self-certified (self-signed) certificate does not provide adequate security, the connecting client generates a notification that the certificate is self-certified, and the user has the opportunity to accept or reject the connection. This option is useful for internal network topologies (such as testing).

If you do not configure a CA trustpoint, when you enable a secure HTTP connection, either a temporary or a persistent self-signed certificate for the secure HTTP server (or client) is automatically generated.

- If the switch is not configured with a hostname and a domain name, a temporary self-signed certificate is generated. If the switch reboots, any temporary self-signed certificate is lost, and a new temporary self-signed certificate is assigned.
- If the switch has been configured with a host and domain name, a persistent self-signed certificate is generated. This certificate remains active if you reboot the switch or if you disable the secure HTTP server so that it will be there the next time you re-enable a secure HTTP connection.



Note The certificate authorities and trustpoints must be configured on each device individually. Copying them from other devices makes them invalid on the switch.



Note When a new certificate is enrolled, the new configuration change is not applied to the HTTPS server until the server is restarted. You can restart the server using the **reload** command. On restarting the server, the switch starts using the new certificate.

If a self-signed certificate has been generated, this information is included in the output of the **show running-config** privileged EXEC command. This is a partial sample output from that command displaying a self-signed certificate.

```
Device# show running-config

Building configuration...

<output truncated>

crypto pki trustpoint TP-self-signed-3080755072
  enrollment selfsigned
  subject-name cn=IOS-Self-Signed-Certificate-3080755072
  revocation-check none
  rsakeypair TP-self-signed-3080755072
!
!
crypto ca certificate chain TP-self-signed-3080755072
certificate self-signed 01
  3082029F 30820208 A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  59312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 33303830 37353530 37323126 30240609 2A864886 F70D0109
  02161743 45322D33 3535302D 31332E73 756D6D30 342D3335 3530301E 170D3933
  30333031 30303030 35395A17 0D323030 31303130 30303030 305A3059 312F302D

<output truncated>
```

You can remove this self-signed certificate by disabling the secure HTTP server and entering the **no crypto pki trustpoint TP-self-signed-30890755072** global configuration command. If you later re-enable a secure HTTP server, a new self-signed certificate is generated.



Note The values that follow `TP self-signed` depend on the certificate serial number of the device.

You can use the **ip http secure-client-auth** command (optional) to allow the HTTPS server to request an X.509v3 certificate from the client. Authenticating the client provides more security than server authentication by itself.



Note Certain device configurations and protocols are identified as insecure and pose a security risk to Cisco devices. While existing deployments continue to function, you must intentionally enable these features for new installations. For more information on remediation, refer to [Cisco C9000 Switching IOS XE – Resilient Infrastructure Playbook](#).



Note Beginning from Cisco IOS XE Amsterdam 17.3.x, CA's self-signed root certificate must be configured for successful authentication of the client.

CipherSuites

A CipherSuite specifies the encryption algorithm and the digest algorithm to use on an SSL connection. When connecting to the HTTPS server, the client web browser offers a list of supported CipherSuites, and the client and server negotiate the best encryption algorithm to use from those on the list that are supported by both.

CipherSuites supported by the switch

For the best possible encryption, you should use a client browser that supports 128-bit encryption. The SSL_RSA_WITH_DES_CBC_SHA CipherSuite provides less security than the other CipherSuites, as it does not offer 128-bit encryption.

The more secure and more complex CipherSuites require slightly more processing time. This list defines the CipherSuites supported by the switch, ranked from fastest to slowest in terms of router processing load:

Table 2: Supported CipherSuites

Cisco CLI cipher-suite	Description
tls13-aes128-gcm-sha256	TLS 1.3 with AES-128-GCM and SHA-256.
tls13-aes256-gcm-sha384	TLS 1.3 with AES-256-GCM and SHA-384
tls13-chacha20-poly1305-sha256	TLS 1.3 with ChaCha20-Poly1305 and SHA-256
rsa-aes-gcm-sha2	RSA authentication with AES-GCM and SHA-2
ecdhe-ecdsa-aes-gcm-sha2	ECDHE key exchange, ECDSA authentication, AES-GCM, and SHA-2
ecdhe-rsa-aes-gcm-sha2	ECDHE key exchange, RSA authentication, AES-GCM, and SHA-2
dhe-aes-gcm-sha2	DHE key exchange with AES-GCM and SHA-2
rsa-aes-cbc-sha2	RSA authentication with AES-CBC and SHA-2
ecdhe-rsa-aes-cbc-sha2	ECDHE key exchange, RSA authentication, AES-CBC, and SHA-2
dhe-aes-cbc-sha2	DHE key exchange with AES-CBC and SHA-2



Note

The latest versions of Chrome do not support the four original cipher suites, thus disallowing access to both web GUI and guest portals.



Note Certain device configurations and protocols are identified as insecure and pose a security risk to Cisco devices. While existing deployments continue to function, you must intentionally enable these features for new installations. For more information on remediation, refer to [Cisco C9000 Switching IOS XE – Resilient Infrastructure Playbook](#).

RSA (in conjunction with the specified encryption and digest algorithm combinations) is used for both key generation and authentication on SSL connections. This usage is independent of whether or not a CA trustpoint is configured.

Default SSL configuration

The default SSL configuration defines the baseline state of SSL on the device before any administrator-configured changes are applied.

Default SSL configuration settings

These guidelines apply to the default SSL configuration:

- The standard HTTP server is disabled.
- SSL is disabled.
- No CA trustpoints are configured.
- No self-signed certificates are generated.

SSL configuration guidelines

SSL configuration guidelines provide the prerequisites and behavioral constraints that apply when enabling SSL in specific switch deployment scenarios.

SSL behavior in cluster, stack, and trustpoint configurations

- When SSL is used in a switch cluster, the SSL session terminates at the cluster commander. Cluster member switches must run standard HTTP.
- Before you configure a CA trustpoint, you should ensure that the system clock is set. If the clock is not set, the certificate is rejected due to an incorrect date.
- In a switch stack, the SSL session terminates at the active switch.

How to configure Secure Socket Layer HTTP

This section describes the tasks required to configure Secure Socket Layer HTTP on Cisco devices, including CA trustpoint configuration, secure HTTP server setup, and secure HTTP client setup.

Configure a CA trustpoint

For secure HTTP connections, we recommend that you configure an official CA trustpoint. A CA trustpoint is more secure than a self-signed certificate.

Beginning in privileged EXEC mode, follow these steps to configure a CA trustpoint:

Procedure

-
- | | |
|---------------|--|
| Step 1 | enable
Example:
Device> enable

Enables privileged EXEC mode. Enter your password, if prompted. |
| Step 2 | configure terminal
Example:
Device# configure terminal

Enters global configuration mode. |
| Step 3 | hostname <i>hostname</i>
Example:
Device(config)# hostname your_hostname

Specifies the hostname of the switch (required only if you have not previously configured a hostname). The hostname is required for security keys and certificates. |
| Step 4 | ip domain name <i>domain-name</i>
Example:
Device(config)# ip domain name your_domain

Specifies the IP domain name of the switch (required only if you have not previously configured an IP domain name). The domain name is required for security keys and certificates. |
| Step 5 | crypto key generate rsa
Example:
Device(config)# crypto key generate rsa

(Optional) Generates an RSA key pair. RSA key pairs are required before you can obtain a certificate for the switch. RSA key pairs are generated automatically. You can use this command to regenerate the keys, if needed. |
| Step 6 | crypto ca trustpoint <i>name</i>
Example:
Device(config)# crypto ca trustpoint your_trustpoint

Specifies a local configuration name for the CA trustpoint and enters CA trustpoint configuration mode. |
| Step 7 | enrollment url <i>url</i> |

Example:

```
Device(ca-trustpoint)# enrollment url http://your_server:80
```

Specifies the URL to which the switch should send certificate requests.

Step 8 **enrollment http-proxy** *host-name port-number***Example:**

```
Device(ca-trustpoint)# enrollment http-proxy your_host 49
```

(Optional) Configures the switch to obtain certificates from the CA through an HTTP proxy server.

- For *host-name*, specify the proxy server used to get the CA.
- For *port-number*, specify the port number used to access the CA.

Step 9 **crl query** *url***Example:**

```
Device(ca-trustpoint)# crl query ldap://your_host:49
```

Configures the switch to request a certificate revocation list (CRL) to ensure that the certificate of the peer has not been revoked.

Step 10 **primary** *name***Example:**

```
Device(ca-trustpoint)# primary your_trustpoint
```

(Optional) Specifies that the trustpoint should be used as the primary (default) trustpoint for CA requests.

- For *name*, specify the trustpoint that you just configured.

Step 11 **exit****Example:**

```
Device(ca-trustpoint)# exit
```

Exits CA trustpoint configuration mode and returns to global configuration mode.

Step 12 **crypto ca authentication** *name***Example:**

```
Device(config)# crypto ca authentication your_trustpoint
```

Authenticates the CA by getting the public key of the CA. Use the same name used in Step 6.

Step 13 **crypto ca enroll** *name***Example:**

```
Device(config)# crypto ca enroll your_trustpoint
```

Obtains the certificate from the specified CA trustpoint. This command requests a signed certificate for each RSA key pair.

Step 14 **end****Example:**

```
Device(config)# end
```

Exits global configuration mode and returns to privileged EXEC mode.

Configure the secure HTTP server

Use this procedure to enable the HTTPS server and configure options that apply to both standard and secure HTTP servers, including path, access list, maximum connections, and timeout policy.

Before you begin

If you are using a certificate authority for certification, configure the CA trustpoint on the switch before enabling the HTTP server. If you have not configured a CA trustpoint, a self-signed certificate is generated the first time that you enable the secure HTTP server.

To verify the secure HTTP connection by using a web browser, enter **https://URL**, where the URL is the IP address or hostname of the server switch. If you configure a port other than the default port, you must also specify the port number after the URL. For example:



Note The AES256_SHA2 cipher suite is not supported because it is not implemented as a selectable HTTPS cipher suite on this platform. A client that offers only this cipher suite cannot establish an HTTPS connection with the device. Ensure to configure a cipher suite supported by both the device and the client.

```
https://209.165.129:1026
```

or

```
https://host.domain.com:1026
```

The existing **ip http access-class access-list-number** command for specifying the access list (only IPv4 ACLs) is going to be deprecated. You can still use this command to specify an access list to allow access to the HTTP server. Two new commands have been introduced to enable support for specifying IPv4 and IPv6 ACLs:

- **ip http access-class ipv4 access-list-name | access-list-number** for specifying IPv4 ACLs
- **ip http access-class ipv6 access-list-name** for specifying IPv6 ACLs

We recommend using the new CLI to avoid receiving warning messages.



Note Certain device configurations and protocols are identified as insecure and pose a security risk to Cisco devices. While existing deployments continue to function, you must intentionally enable these features for new installations. For more information on remediation, refer to [Cisco C9000 Switching IOS XE – Resilient Infrastructure Playbook](#).

Note these considerations for specifying access lists:

- If you specify an access list that does not exist, the configuration takes place but you receive this warning message: `ACL being attached does not exist, please configure it`
- If you use **ip http access-class ipv4 access-list-name | access-list-number** or **ip http access-class ipv6 access-list-name**, and an access list was already configured using **ip http access-class**, this warning message appears: `Removing ip http access-class <access-list-number>`

Beginning in privileged EXEC mode, follow these steps to configure a secure HTTP server:

Procedure

Step 1

enable

Example:

```
Device> enable
```

Enables privileged EXEC mode. Enter your password, if prompted.

Step 2

show ip http server status

Example:

```
Device# show ip http server status
```

(Optional) Displays the status of the HTTP server to determine if the secure HTTP server feature is supported in the software. You should see one of these lines in the output:

```
HTTP secure server capability: Present
```

or

```
HTTP secure server capability: Not present
```

Step 3

configure terminal

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 4

ip http secure-server

Example:

```
Device(config)# ip http secure-server
```

Enables the HTTPS server if it has been disabled. The HTTPS server is disabled by default.

Step 5

ip http secure-port *port-number*

Example:

```
Device(config)# ip http secure-port 443
```

(Optional) Specifies the port number to be used for the HTTPS server. The default port number is 443. Valid options are 443 or any number in the range 1025 to 65535.

Step 6

ip http secure-ciphersuite

Example:

```
Device(config)# ip http secure-ciphersuite tls13-aes256-gcm-sha384
```

(Optional) Specifies the CipherSuites (encryption algorithms) to be used for encryption over the HTTPS connection. If you do not have a reason to specify a particular CipherSuite, you should allow the server and client to negotiate a CipherSuite that they both support. This is the default.

Step 7

ip http secure-client-auth

Example:

```
Device(config)# ip http secure-client-auth
```

(Optional) Configures the HTTP server to request an X.509v3 certificate from the client for authentication during the connection process. The default is for the client to request a certificate from the server, but the server does not attempt to authenticate the client.

Step 8 **ip http secure-trustpoint** *name***Example:**

```
Device(config)# ip http secure-trustpoint your_trustpoint
```

Specifies the CA trustpoint to use to get an X.509v3 security certificate and to authenticate the client certificate connection.

Note

Use of this command assumes you have already configured a CA trustpoint according to the previous procedure.

Step 9 **ip http path** *path-name***Example:**

```
Device(config)# ip http path /your_server:80
```

(Optional) Sets a base HTTP path for HTML files. The path specifies the location of the HTTP server files on the local system (usually located in system flash memory).

Step 10 **ip http access-class** {**ipv4** {*access-list-number* | *access-list-name*} | **ipv6** {*access-list-name*}}**Example:**

```
Device(config)# ip http access-class ipv4 4
```

(Optional) Specifies an access list to use to allow access to the HTTP server.

Step 11 **ip http max-connections** *value***Example:**

```
Device(config)# ip http max-connections 4
```

(Optional) Sets the maximum number of concurrent connections allowed to the HTTP server. We recommend that the value be at least 10. This is required for the UI to function as expected.

Step 12 **ip http timeout-policy** **idle** *seconds* **life** *seconds* **requests** *value***Example:**

```
Device(config)# ip http timeout-policy idle 120 life 240 requests 1
```

(Optional) Specifies how long a connection to the HTTP server can remain open:

- **idle:** the maximum time period when no data is received or response data cannot be sent. The range is 1 to 600 seconds. The default is 180 seconds (3 minutes).
- **life:** the maximum time period from the time that the connection is established. The range is 1 to 86400 seconds (24 hours). The default is 180 seconds.
- **requests:** the maximum number of requests processed on a persistent connection. The maximum value is 86400. The default is 1.

Step 13 **end**

Example:

```
Device(config)# end
```

Exits global configuration mode and returns to privileged EXEC mode.

Configure the secure HTTP client

The standard HTTP client and secure HTTP client are always enabled. A certificate authority is required for secure HTTP client certification. This procedure assumes that you have previously configured a CA trustpoint on the switch. If a CA trustpoint is not configured and the remote HTTPS server requires client authentication, connections to the secure HTTP client fail.

Beginning in privileged EXEC mode, follow these steps to configure a secure HTTP client:



Note Certain device configurations and protocols are identified as insecure and pose a security risk to Cisco devices. While existing deployments continue to function, you must intentionally enable these features for new installations. For more information on remediation, refer to [Cisco C9000 Switching IOS XE – Resilient Infrastructure Playbook](#).

Procedure

Step 1 enable

Example:

```
Device> enable
```

Enables privileged EXEC mode. Enter your password, if prompted.

Step 2 configure terminal

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 ip http client secure-trustpoint *name*

Example:

```
Device(config)# ip http client secure-trustpoint your_trustpoint
```

(Optional) Specifies the CA trustpoint to be used if the remote HTTP server requests client authentication. Using this command assumes that you have already configured a CA trustpoint by using the previous procedure. The command is optional if client authentication is not needed or if a primary trustpoint has been configured.

Step 4 ip http client secure-ciphersuite

Example:

```
Device(config)# ip http secure-ciphersuite tls13-aes256-gcm-sha384
```

(Optional) Specifies the CipherSuites (encryption algorithms) to be used for encryption over the HTTPS connection. If you do not have a reason to specify a particular CipherSuite, you should allow the server and client to negotiate a CipherSuite that they both support. This is the default.

Step 5 **end**

Example:

```
Device(config)# end
```

Exits global configuration mode and returns to privileged EXEC mode.

Monitoring secure HTTP server and client status

To monitor the SSL secure server and client status, use the privileged EXEC commands in the table "Commands for displaying the SSL secure server and client status".

Table 3: Commands for displaying the SSL secure server and client status

Command	Purpose
show ip http client secure status	Shows the HTTP secure client configuration.
show ip http server secure status	Shows the HTTP secure server configuration.
show running-config	Shows the generated self-signed certificate for secure HTTP connections.

Additional references for Secure Socket Layer HTTP

This topic provides additional references for the Secure Socket Layer HTTP feature, including related documents and technical assistance resources.

Table 4: Related documents

Related topic	Document title
Certification authority	Configuring Certification Authority Interoperability

Table 5: Technical assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support