SIP Endpoint SDK Developer's Guide

Configuring secure connections (TLS) for SIP

4/30/2025

Contents

Configuring secure connections (TLS) for SIP

The TLS support in SIP Endpoint SDK and Genesys Softphone on macOS is based on the Genesys Security Pack implementation, which relies on the OpenSSL library, a de-facto industry standard implementation on UNIX platforms. Currently, the macOS Keychain system is not supported. Instead, users should provide the client-side certificate and trusted CA list as files, using one of the file formats supported by the OpenSSL project.

To secure the connection to the SIP Server, users should set the **protocol** parameter to TLS in the corresponding Connectivity element of the **Basic** container in the SIP Endpoint SDK configuration file, whether the connection is direct or via SIP Proxy or SBC.

```
<Connectivity user="{dn}" server="{server:port}" protocol="TLS"/>
```

For Mutual TLS, you should also specify the **certificate** and **certificate-key** options, which refer to the public and private parts of the client-side certificate.

If these options are left empty, outgoing TLS connections will be used, but incoming TLS connections will not be possible. However, most deployments do not encounter this problem because Genesys SIP Server, SIP Proxy, and supported SBCs reuse the client-originated TLS connection by default without opening another TLS connection for delivering incoming SIP messages. In either case, the **trusted-ca** option is mandatory and should include the public key of the Certificate Authority (CA) used to sign the server-side certificate.

> ### Important
>
> The TLS configuration settings for securing the SIP connection can be found in the **system.security** section.

certificate

Valid Values: String

Full path pointing to the certificate file, which is used as a client-side certificate for outgoing TLS connections in case of Mutual TLS, and server-side certificate for incoming TLS connections in case when SBC is configured to not reuse the client TLS connection. For example, `/Users/jdoe/gcti/certificate/hostname_cer.pem`.

This option replaces the `cert_file` option from previous versions. For backwards compatibility, the SDK accepts `certificate` or `cert_file` and the former takes priority.

certificate-key

Valid Values: String

Full path to the endpoint Private Key file, corresponding to the Public Key in the certificate specified by **certificate** option, or, if the Private Key is stored with the certificate, the full path to the certificate .pem file. For example, `/Users/jdoe/gcti/certificate/hostname_priv_key.pem`

This option replaces the `priv_key_file` option from previous versions. For backwards compatibility, the SDK accepts both `certificate-key` and `priv_key_file` option names and the former takes priority.

trusted-ca

Valid Values: String

Full path to the Certificate Authority's (CA) list file. For example, `/users/jdoe/gcti/certificate/ca/ca_cert.pem`

This option replaces the `ca_list_file` option from previous versions. For backwards compatibility, the SDK accepts `trusted-ca` or `ca_list_file` option names and the former takes priority.

sec-protocol

Valid Values: SSLv23, TLSv12, TLSv13 or an empty string

Default Value: an empty string

Specifies the protocol used by the component to set up secure connections:

- SSLv23 - The highest TLS protocol version supported by both sides of communication, from TLS 1.1 and up (remains for backward compatibility, not recommended for new deployments).
- empty string - the default Security Pack settings (currently the highest TLS version supported by both sides from 1.2 upwards).
- TLSv12 - TLS version 1.2.
- TLSv13 - TLS version 1.3.

cipher-list

Specifies the defined list of ciphers for TLSv1.2 and earlier protocols. The cipher list must be in a valid format for OpenSSL library. See the cipher-list setting defined in the Security Deployment Guide and OpenSSL documentation.

ciphersuites

Valid Values: The colon-separated list of TLSv1.3 ciphersuite names, as defined in RFC 8446, in preference order. The list may include one or more of the following:

- TLS_AES_256_GCM_SHA384
- TLS_CHACHA20_POLY1305_SHA256

- TLS_AES_128_GCM_SHA256
- TLS_AES_128_CCM_8_SHA256
- TLS_AES_128_CCM_SHA256

Default Value: empty string, which is equivalent to `TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256`

Specifies the defined list of ciphersuites to be used for TLSv1.3, if that TLS version is supported by both side of the connection (and negotiated during handshake). This option supplements `cipher-list` option (which is still applicable for TLSv1.2 and below).

tls-crl

Specifies the full path to the file that contains one or more certificates defining the Certificate Revocation List (CRL). See the tls-crl setting defined in the Security Deployment Guide and OpenSSL documentation for more information.

tls-target-name-check

Valid Values: `no`, `host`

Default Value: `no`

Specifies if the Common Name in the subject field and/or the Subject Alternate Names of the server's certificate will be compared to the target host name (option value `host`). If they are not identical, the connection fails. If the option is set to `no`, a comparison is not made, and the connection is allowed (provided the server's certificate is valid and signed by trusted CA, as disabling target name check does not affect other certificate verification steps).

> **Important**
>
> The default value for this option is 'no' only to ensure backward compatibility with previous releases. Genesys recommends to always set the value as 'host' in production environments for security reasons to avoid any man-in-the-middle attack attempts.

> **Important**
>
> If encryption is enabled in SIP mode, the user workstation may connect to the CRL systems of the Certificate Authorities that issued the SSL certificates for the SIP User Agents (SIP UAs).

For additional information, refer to:

- Genesys Secure Connection (TLS) guide
- OpenSSL project
- certificate formats (on OpenSSL documentation)