Genesys Security Deployment Guide

Advanced TLS

4/4/2026

Contents

Advanced TLS

This topic contains additional information about TLS.

Tuning Protocol Version Availability

In release 8.5.1, as part of the transition to OpenSSL from RSA Bsafe, the behavior of the **sec-protocol** option has been modified.

> ### Important
> Refer to Security Pack 8.5.100.25 for information on OpenSSL version 1.1.1g and later, TLS 1.3, and SAN certificate.

The availability of a particular protocol setting in **sec-protocol** strongly depends on the actual component version.

Generally, the protocol versions currently available are as follows:

- On UNIX and Linux, TLS 1.3 is the highest available protocol with the OpenSSL Security Pack; TLS 1.1 with the RSA Security Pack.
- On Windows, refer to Microsoft documentation for the list of supported TLS versions for particular Windows deployment. Genesys recommends that you explicitly enable the desired protocol version in the Windows registry; refer to the Windows document TLS/SSL Settings for more information about enabling and disabling protocols in the Windows registry.

> ### Warning
> Genesys components use the Windows implementation of TLS on Windows platforms, and therefore Windows settings take precedence over **sec-protocol** settings. Genesys software is unable to use a protocol version if it is disabled on the Windows operating system level.

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.

The supported protocol version modes can be categorized as one of two types:

- **strict** mode— TLSv12 and TLSv13 are the strict protocol version modes. These settings can be used to enforce a specific protocol version. The connection will not be established if the remote server does not accept the enforced protocol version.
- **compatibility** mode—SSLv23 and the default mode, are compatible with all modes from TLSv1.1 or TLSv1.2 up to and including TLSv13, and will connect with the highest mode offered by the other side of the TLS connection.

Tuning Available Cipher Lists for TLS v1.2

Normally, the set of available ciphers is provided by your InfoSec, and can be configured to the preferences of the user. The **cipher-list** configuration option allows the supporting Genesys component to select a list of cipher suites used in TLSv 1.2 and lower. This option is transferred to a third-party library and describes the set of possible cipher suites.

Cipher List Formatting Rules

> **Important**
>
> This section describes cipher list format for an application using the Genesys common library. If you are configuring a cipher list for the PSDK-based application, refer to the *Platform SDK Developer's Guide* for the proper format, and more information about cipher lists in PSDK.

For applications using the Genesys common library , the cipher list string is a list of cipher operations. Each operation consists of an optional operator character followed by a name. See OpenSSL cipher commands for more information. Cipher list strings must conform to the following formatting rules:

[+] Show rules

- The name is either a valid cipher name or a cipher alias. Valid names contain the characters a-z, A-Z, 0-9, and - (dash).
- List separator characters are used to separate the names and aliases in the list. A list separator character must be a colon.
- Multi-part names are joined with +.

- The character ! appearing immediately after a separator indicates a kill operation. The cipher following the character becomes unavailable.
- The character + appearing immediately after a separator indicates an order operation. This moves the active cipher to the current position in the list of ciphers.
- The character - appearing immediately after a separator indicates a delete operation. The cipher following the character becomes inactive. The cipher remains available for further operations.
- A non-operator character appearing immediately after a separator indicates an add operation. If the cipher following the character is not currently active, the cipher is added as an active cipher to the end of the list of available ciphers.

All operations occur in the order in which they appear in the list. If the cipher corresponding to a name (or part of a name, for multi-part names) is not available in the library, it is ignored during loading. In this situation, no error message is logged.

Aliases

Ciphers also have aliases. The following table details the primary cipher aliases.

[+] Show table

Alias	Description
kRSA, kDHr, kDHd, kEDH	Key exchange types
aRSA, aDSS, aNULL, aDH	Authentication
DES, 3DES, RC4, RC2, eNULL	Ciphers
MD5, SHA1	Message digests

Groups of commonly-used ciphers also have aliases. This enables multiple aliases to be specified easily. The following table details the cipher group aliases.

[+] Show table

Alias	Description
SSLv2	All SSLv2 ciphers
SSLv3	All SSLv3 ciphers
EXP	All export ciphers
LOW	All low strength ciphers (no export ciphers, normally single DES)
MEDIUM	128-bit encryption
HIGH	Triple DES with key lengths larger than 128 bits, and some cipher suites with 128-bit keys

Aliases can also be joined in a colon-separated list to specify the ciphers to add, move, or delete.

Example

The following is an example of a cipher string:

```
!ADH:RC4+RSA:HIGH:MEDIUM:LOW:EXP:+SSLv2:+EXP
```

This cipher string is interpreted in the following sequence:

1. Do not consider any ciphers that do not authenticate.
2. Use ciphers that use RC4 and RSA.
3. Include the HIGH, MEDIUM, and LOW security ciphers.
4. Add all export ciphers.
5. Pull all SSLv2 and export ciphers to the end of the list.

Tuning Available Cipher Lists for TLS v1.3

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 lower).

Applications should use the SSL_CTX_set_ciphersuites() or SSL_set_ciphersuites() functions to configure TLSv1.3 ciphersuites.

> **Important**
>
> The functions SSL_CTX_get_ciphers() and SSL_get_ciphers() return the full list of ciphersuites that have been configured for both TLSv1.2 and lower and TLSv1.3.