Framework Deployment Guide

Genesys Implementation of Secure Protocol Connections

5/8/2025

Contents

Genesys Implementation of Secure Protocol Connections

This topic describes how to use the Genesys Security Pack to implement secure connections in Management Framework. More detailed information about OpenSSL, and how to use it to secure connections between Genesys components, is contained in the Secure Connections (TLS) section of the *Genesys Security Deployment Guide*.

Deployment Steps

To deploy Security Pack on UNIX or Linux, run the installation package. After the files have been copied into the target folders, make sure that you set up required environment variables to allow Genesys applications to load shared modules from those locations. For example, on Linux, you might want to include the installation path to the LD_LIBRARY_PATH environment variable and restart affected applications. Note that if you are using LCA to start your applications from the Management Layer, LCA must be restarted first to pick up the changes to the environment variables before you can (re)start the applications.

On Windows operating systems, you do not need to deploy any additional software; secure connections are available for use by any Genesys application that supports them.

Backward Compatibility

The new Security Pack is a drop-in replacement of the existing Security Pack. To upgrade to the OpenSSL version, you replace the binary modules. You do not have to make any change to the configuration of existing deployments.

> **Important**
>
> You must restart those applications using secured connections after upgrading to the new Security Pack.

To ensure backward compatibility, the new Security Pack includes a new mode (referred to as compatibility mode) that restores some behavior of the old Security Pack. This mode is disabled by default.

> **Warning**

- Compatibility mode should be enabled only as a last resort if the new Security Pack is encountering compatibility errors in the customer environment.
- When in compatibility mode, Genesys strongly recommends that you take the necessary actions to avoid long-term usage of this mode.

To enable the Security Pack compatibility mode, set the environment variable GCTI_SECPACK_COMPAT_MODE to 1 before starting the application. Once started, you cannot disable the mode during application runtime.

The following compatibility issue workarounds are enabled by compatibility mode:

- When verifying a peer certificate chain, a chain entry certificate revocation status will be ignored if the certificate is explicitly trusted as a CA in the local configuration (that is, listed in the ca certificate list).
- The peer certificate chain verification process will ignore any non-compatible "Key usage" extension value. For example, a peer certificate without "authentication" usage will be accepted in compatibility mode. RSA did not verify the "Key usage" extension values; OpenSSL does.

If you want to continue using the RSA BSafe implementation instead of OpenSSL, make sure you set up your environment so that shared modules from the **<Security Pack root>/legacy** folder have being loaded instead of the default ones (located in **<Security Pack root>**).

OpenSSL

OpenSSL is the industry standard SSL implementation. It is widely used in both open source and commercial products, and so existing vulnerabilities and issues are promptly discovered and fixed.

OpenSSL is developed in parallel with all the new features introduced into SSL, such as TLSv1.2 protocol version support.

Version Information

The OpenSSL website contains release strategy information, describing the version naming, release schedule and more. See the Links section for additional information regarding version information.

To determine the version of OpenSSL that the Security Pack is using, refer to the most recent Security Pack on UNIX Release Note.

FIPS mode information

OpenSSL provides native support for FIPS mode. Unlike RSA, OpenSSL does not require a special version capable of performing FIPS mode operations. A FIPS-capable version of OpenSSL is used by the Security Pack. For more info on FIPS mode and OpenSSL support of FIPS, please refer to OpenSSL documentation and the FIPS User Guide.

Links

OpenSSL site	https://www.openssl.org/
OpenSSL backwards compatibility analysis dashboard	http://upstream.rosalinux.ru/versions/openssl.html
OpenSSL FIPS User Guide	https://www.openssl.org/docs/fips/UserGuide-2.0.pdf

Protocol Versions Compatibility

The **sec-protocol** option supports the following modes: SSLv23 (the default), SSLv3, TLSv1, TLSv11, and TLSv12.

The availability of a particular protocol setting in **sec-protocol** strongly depends on the actual component version. Older components may not support this option at all. No components except the most recent Management Framework servers support the TLSv12 value. For other components, refer to documentation specific to the component to determine protocols are supported (if any), and for additional information.

Generally, the protocol versions currently available are as follows:

- On UNIX and Linux, TLS 1.2 is the highest available protocol with OpenSSL SecPack; TLS 1.1 with RSA SecPack.
- On Windows, TLS 1.1 and TLS 1.2 are supported starting with Microsoft Vista / Server 2008. However, in most cases these must be enabled in the registry to become available. Genesys recommends that you explicitly enable the desired protocol version in the Windows registry; refer to the following Windows document for more information about enabling and disabling protocols in the Windows registry: TLS/SSL Settings. Note that Genesys components use the Windows implementation of TLS on Windows platforms, and hence Windows settings take precedence over the **sec-protocol** settings. Genesys software is unable to use a protocol version if it is disabled on the Windows operating system level.

The supported protocol version modes can be categorized as one of two types: strict or compatibility. These are described below.

Strict protocol version modes

SSLv3, TLSv1, TLSv11, and TLSv12 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 protocol version modes

SSLv23, the default mode, is compatible with all modes from SSLv2 up to and including TLSv12, and will connect with the highest mode offered by the other server. If SSL 2 ciphers are explicitly specified, the SSL 2 client can connect only to servers running in SSLv23 mode. Otherwise, the SSL 2 mode is deprecated; it is highly vulnerable and is not to be used.