Managing TLS and trusted CA certificates

TLS certificates are used by the Management Node and each Conferencing Node to verify their identity to clients connecting to them over HTTPS (web) or SIP TLS. These clients include:

- video endpoints
- web browsers (including the Infinity Connect web app)
- Infinity Connect mobile clients (certificates are mandatory for these clients)
- third-party video network infrastructure devices
- Outlook clients (if the VMR Scheduling for Exchange service is enabled).

You can use Pexip Infinity's inbuilt Certificate Signing Request (CSR) generator to assist in acquiring a server certificate from a Certificate Authority.

This topic covers:

For information about how to configure Pexip Infinity to verify peer certificates, and mutual TLS authentication, see Verifying SIP TLS connections with peer systems.

Note that communication between the Management Node and Conferencing Nodes, and between Conferencing Nodes themselves, does not rely on TLS certificates; instead it uses an IPsec transport. For more information see Encryption methodologies.

Certificates overview

The Public Key Infrastructure (PKI) provides a framework for digital certificate management. It provides the following benefits:

- **Authentication**: identities are validated to ensure that only authorized users and devices have access to a server.

- **Encryption**: sessions can be encrypted, so information can be transmitted privately.
- **Data integrity**: ensures that any messages or data transferred to and from devices and servers is not altered.

The primary elements of the PKI are:

- **Public/private key pairs**: public and private keys are used to encrypt and decrypt the information being transferred to a server. Only the private key, which is kept secret by the server, can decrypt the information that is encrypted by the public key. This mechanism is known as asymmetric cryptography (as the encryption is done using non-identical keys); the two keys are mathematically related, and whatever is encrypted with a public key can only be decrypted by its corresponding private key and vice versa.
- **Certificate**: a certificate contains the public key and information about its owner (often referred to as the subject and is typically expressed as a hostname or domain name) and its issuer (typically a trusted, third-party Certificate Authority). This certificate metadata is formatted according to the ITU-T X.509 international standard. A certificate is not considered valid unless it has been directly or indirectly signed by a trusted CA.
- **Certificate Authority (CA)**: a CA is an organization such as Symantec or Comodo that issues digital certificates to corporations after having verified the applicant's identity. The certificate is proof that a certain server or website is owned by a certain organization. A CA can use its own private key to sign the certificates it issues. That signed certificate can then be verified as being signed by a trusted CA by checking the signature against the CA’s own root certificate (via its public key). However, many CAs do not sign with their root certificate, but instead with an intermediate certificate — an intermediate authority is a certificate issuer that has itself been issued by a root or another higher level intermediate authority. This method creates a chain of trust.
- **Chain of trust**: when a device validates a certificate, it compares the certificate issuer with its list of trusted CAs. If a match is not found, it checks if the certificate of the issuing CA was issued by a trusted CA, and so on until the end of the certificate chain. The top of the chain, the root certificate, must be issued by a trusted Certificate Authority. Web browsers and other clients typically have a list of CA certificates that they trust, and can thus verify the certificates of individual servers, however, the server is often required to present a full certificate chain along with its server certificate.

TLS certificate usage within Pexip Infinity

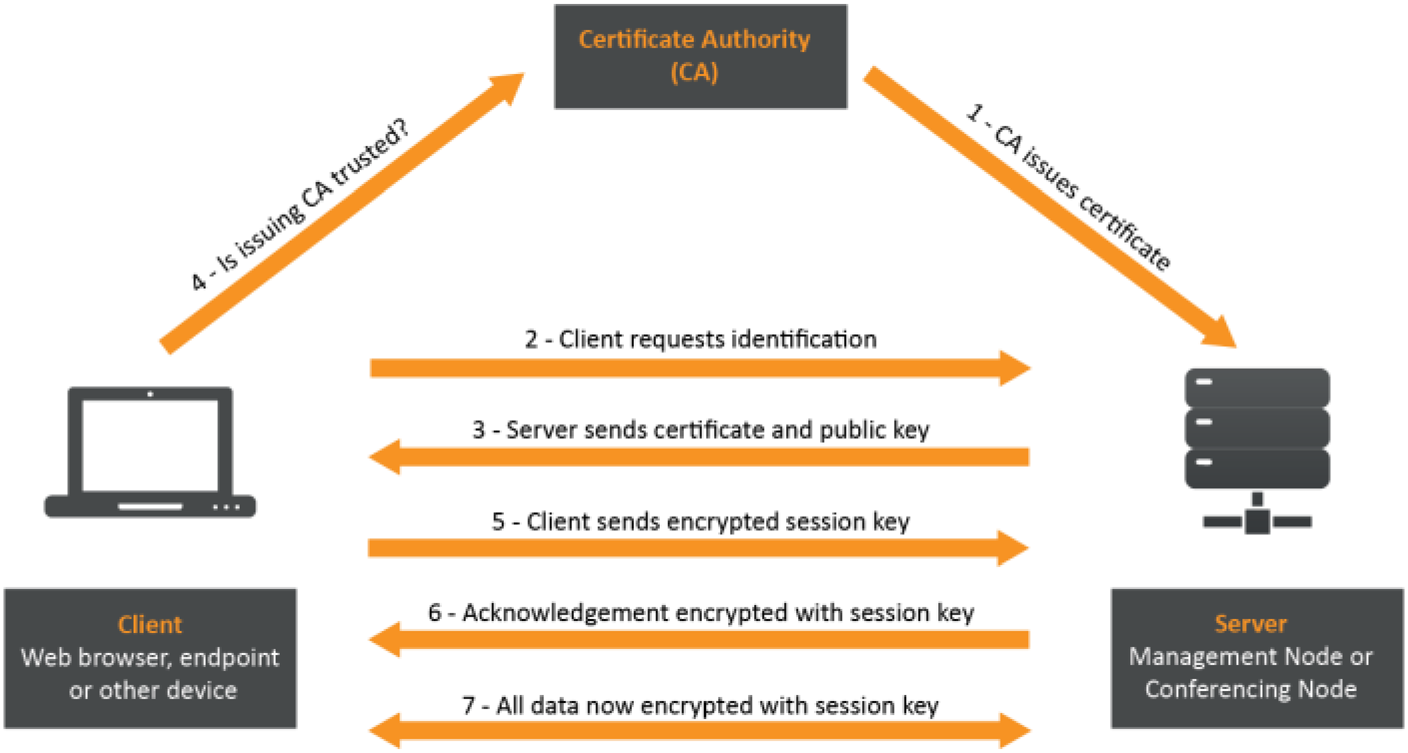
The certificates used within Pexip Infinity are standard digital certificates, but they are referred to as TLS certificates as they are used when establishing a TLS connection to a Pexip node.

The clients that connect to Pexip Infinity over TLS must trust the identity of the Certificate Authority (CA) that signed the node's TLS certificate. The Pexip Infinity platform ships with a self-signed certificate for the Management Node, and each Conferencing Node is deployed with a self-signed certificate. These certificates have a 4096 bit public key and are also appended with 2048 bit Diffie-Hellman parameters.

As these certificates are self-signed, they will not be trusted by clients. We therefore recommend that you replace these certificates with your own certificates that have been signed by either an external CA or a trusted internal CA.

You can use a tool such as https://www.sslshopper.com/ssl-checker.html to verify certificates and the chain of trust (specify port 5061 i.e. use the format <domain>:5061 for the server hostname to ensure that SIP TLS connections are checked).

The Management Node and Conferencing Nodes enable HSTS (HTTP Strict Transport Security) to ensure greater security. This means that if your deployment moves from using a valid TLS certificate to using an invalid certificate (e.g. you redeploy a Conferencing Node, or your certificate expires or is invalidated for some reason) then certain web browsers will stop you from accessing that node via the web when using the DNS name of that node, until you correct the certificate issue. You may browse directly to the IP address of the node in the meantime.


Certificate Authority (CA)
4 - Is issuing CA trusted?
1 - CA issues certificate
2 - Client requests identification
3 - Server sends certificate and public key
5 - Client sends encrypted session key
6 - Acknowledgement encrypted with session key
7 - All data now encrypted with session key
Client
Web browser, endpoint or other device
Server
Management Node or Conferencing Node

In general, to achieve encrypted communication using TLS the following must happen:

1. The CA issues a signed certificate which is uploaded to the server.
2. When a client needs to communicate with the server, it sends a request to the server asking it to provide identification.
3. The server sends back a copy of its TLS certificate and its public key.
4. The client checks whether the CA that issued the certificate is one that it trusts.

5. If the CA is trusted, and if the certificate is otherwise valid, the client creates a session key encrypted with the server's public key and sends it to the server.

6. The server decrypts the session key. It then uses the session key to encrypt an acknowledgment which it sends to the client in order to initiate the encrypted communication.

7. The server and the client now encrypt all communication using the session key.

Certificate usage guidelines

When requesting/generating certificates for your Pexip Infinity platform:

- Do not use SHA-1 certificates on your Conferencing Nodes — use SHA-256 certificates instead. (Some clients, such as iOS devices, already mandate the use of SHA-256 certificates, and browsers will soon stop accepting SHA-1 certificates.)

- If browsers (Infinity Connect web app clients, for example) need to access your Pexip Infinity platform, ensure that your certificates contain at least one subject alternative name (SAN) entry — typically a repeat of the Common Name. (Chrome and Firefox browsers are dropping support for Common Name matching.)

Wildcard TLS certificates are not supported in SIP or Microsoft Skype for Business / Lync environments (as per RFC 5922). If you are using SIP or Skype for Business / Lync, your Conferencing Nodes must not use wildcard TLS certificates.

Alarms

An alarm is raised on the Management Node if:

- the Management Node or a Conferencing Node has no associated TLS certificate

- a TLS certificate has an incomplete chain of trust to the root CA certificate

- one or more of your trusted CA certificates or TLS certificates is due to expire within the next 30 days.

Managing a node's TLS server certificate

You can upload, view/modify, delete and download the TLS server certificates that are used by the Management Node and by each Conferencing Node. You can also generate a certificate signing request for an existing certificate / subject name.

Note that after making any changes to certificates, you need to wait for the files to be synchronized to the relevant Management Node or Conferencing Node (typically after approximately one minute). If changing the certificates in a chain, a reboot of the associated Conferencing Nodes may be required if the changes do not produce the desired effect.

Uploading a TLS server certificate

To upload a new TLS server certificate for the Management Node or a Conferencing Node:

1. From the Pexip Infinity Administrator interface, go to Platform > TLS certificates.
2. Select Add TLS certificate.
3. Complete the following fields:

TLS certificate	Paste the PEM-formatted certificate into the text area or alternatively select the file containing the new TLS certificate. You must upload the certificate file that you have obtained from the Certificate Authority (typically with a .CRT or .PEM extension). Do not upload your certificate signing request (.CSR file). The certificate must be valid for the hostname or FQDN of the Management Node or Conferencing Node to which it will be assigned. You can paste multiple certificates into the text area, but one of those certificates must pair with the associated private key.
Private key	Paste the PEM-formatted private key into the text area or alternatively select the file containing the private key that is associated with the new TLS certificate. Private key files typically have a .KEY or .PEM extension. Pexip Infinity supports RSA and ECDSA keys.
Private key passphrase	If the private key is encrypted, you must also supply the associated passphrase.
TLS parameters	Optionally, paste any additional PEM-formatted parameters into the text area or alternatively select the file containing the parameters that are to be associated with the new TLS certificate. Custom DH parameters and an EC curve name for ephemeral keys can be added. Such parameters can be generated through the OpenSSL toolkit using the commands openssl dhparam and openssl ecparam. For example, the command openssl dhparam -2 -outform PEM 2048 generates 2048 bit DH parameters. Note that these parameters can alternatively be added 'as is' to the end of the TLS certificate.
Nodes	Select one or more nodes to which the new TLS certificate is to be applied. If required, you can upload a certificate and then apply it to a node later.

4. Select Save.

If a certificate with the same subject name already exists (e.g. when replacing an expired certificate), the new certificate is uploaded alongside the original certificate (unless the issuer and serial number details are identical, in which case the existing certificate is updated with the new contents from the file). If the original TLS certificate was assigned to one or more Conferencing Nodes you need to move those node assignments over to the new certificate.

Viewing or modifying existing TLS certificates and changing node assignments

To view information about an existing TLS certificate, change a certificate's contents, or change the nodes to which a certificate is applied:

1. Go to Platform > TLS certificates.

   By default you are shown a list and the current status of **All certificates** that have been uploaded. Status values include:

   - Good: it is a good, valid certificate.
   - Temporary: it is a self-signed certificate.
   - Weak signature: the certificate is signed with SHA-1 or another old signature. We recommend that you use SHA-256 certificates instead.
   - Empty subject: the certificate only has SANs. It is a valid certificate but many browsers will not accept it.
   - <n> days left: when the certificate is due to expire within the next 60 days.
   - Expired: when the certificate has expired.

   You can alternatively select to view Certificates by Node to see which certificate has been assigned to the Management Node or to a particular Conferencing Node.

2. Select the subject name of the certificate you want to view or modify.

   The decoded certificate data is shown, including any chain of trust information.

3. If required, you can modify the:
   - Nodes to which the certificate is assigned. If you assign a certificate to a node, it will automatically replace any other certificate that was previously assigned to that node.
   - TLS certificate data (by expanding the PEM-formatted data section).
   - TLS parameters associated with the certificate (by expanding the PEM-formatted data section).

   You cannot modify the private key.

4. Select Save.

Uploading multiple TLS certificate files

To upload a batch of TLS certificates:

1. Go to Platform > TLS certificates.

2. Select Import files.

3. Select Choose Files to pick one or more PEM-formatted text files that you want to import.
    - The files should contain server TLS certificates with matching private keys.
    - Private keys can be uploaded as separate files or appended to the server TLS certificate file(s).
    - DH or EC parameters may be appended to each server TLS certificate.

    Note that trusted CA certificate files can also be imported via this method if required.

4. Enter the Private key passphrase if the private key is encrypted.

5. Select Import.

    This adds the certificates in the selected files to the existing list of TLS certificates (or to the list of trusted CA certificates, if appropriate).

6. You can then select each imported TLS certificate in turn and assign it to a Management Node or one or more Conferencing Nodes as appropriate.

If a certificate with the same subject name already exists (e.g. when replacing an expired certificate), the new certificate is uploaded alongside the original certificate (unless the issuer and serial number details are identical, in which case the existing certificate is updated with the new contents from the file). If the original TLS certificate was assigned to one or more Conferencing Nodes you need to move those node assignments over to the new certificate.

Deleting an existing TLS certificate

To delete one or more TLS certificates:

1. Go to Platform > TLS certificates.

2. Select the boxes next to the certificates to be deleted, and from the Action drop-down menu select Delete selected TLS certificates and select Go.

Downloading an existing TLS certificate

To download an existing TLS certificate:

1. Go to Platform > TLS certificates.

2. Select the subject name of the certificate you want to download.

    The certificate data is shown.

3. Go to the bottom of the page and select Download.

4. By default the certificate itself and any intermediate certificates are selected to be included in the download.

   You can also choose to include the private key (in which case you must also supply a passphrase).

5. Select the download format, either PEM or PKCS # 12 (PFX).

6. Select Download.

   A file called <subject_name>_certificate or <subject_name>_keycert (if the private key was included) with either a .pem or .pfx extension is downloaded. This file contains the certificate and/or the private key as requested.

   You can also download multiple certificates into one file: select the boxes next to the certificates to be downloaded, and from the Action drop-down menu select Download and select Go. In this case the generated file is called all_certificates or all_keycerts (if private keys were included).

Note that you cannot download a temporary / self-signed certificate.

You can use the Pexip Infinity Management Node to convert PEM certificates to PFX format (or vice versa), by uploading a PEM-formatted certificate and then downloading it again in PFX format. When downloading you can also include the necessary intermediate certificates in the PFX bundle.

Replacing an existing certificate (by generating a new certificate signing request)

You can generate a certificate signing request (CSR) for an existing certificate / subject name, for example if your current certificate is soon due to expire and you want to replace it. Before generating the CSR you can change the certificate data to be included in the new request, such as adding extra subject alternative names (SANs) to those already present in the existing certificate.

For instructions on how to do this, see Requesting a certificate signing request (CSR) for an existing certificate / subject name.

Managing trusted CA certificates

Trusted CA certificates are used within Pexip Infinity to:

- verify client certificates presented to Pexip Infinity when SIP TLS verification mode is enabled
- provide a certificate chain of trust when clients connect to a Conferencing Node over SIP TLS
- verify the server certificate on a video network infrastructure device when a Conferencing Node makes a SIP TLS outbound connection to that device, and that device chooses a cipher suite that requires authentication and a certificate is exchanged
- verify connections to an LDAP server.

Pexip Infinity ships with an inbuilt list of trusted CA certificates. This list is based on the Mozilla CA Certificate Store and cannot be modified.

In addition, you can upload a customized set of trusted CA certificates to the Pexip Infinity platform.

Chain of trust

For a server's TLS certificate to be trusted by a client, the client must be configured to trust the Certificate Authority (CA) that signed the server certificate. Many CAs do not sign with their root certificate, but instead with an intermediate certificate. Clients, however, may only trust the root CA. Therefore the server (in this case the Management Node or Conferencing Node) is often required to present a full certificate chain along with their TLS server certificate.

When this is the case, the chain of intermediate CA certificates must be installed on the Management Node to ensure that the certificate chain of trust is properly established when clients connect to a Conferencing Node over SIP TLS.

To do this you must upload a custom **Trusted CA certificates** file that contains all the required CA certificates, one after each other, in PEM format.

Uploading and managing additional trusted CA certificates

You can upload a customized set of trusted CA certificates to the Pexip Infinity platform. Any trusted CA certificates uploaded here are used in addition to the default set of trusted CA certificates that ships with Pexip Infinity.

To manage the set of custom trusted CA certificates, go to Platform > Trusted CA certificates. This shows a list and the current status of all the trusted CA certificates that have been uploaded. From here you can:

- **Upload a file of Trusted CA certificates**: select Import files, select Choose Files to pick one or more PEM files that you want to import, and then select Import.

  This adds the certificates in the selected files to the existing list of trusted CA certificates (or to the list of TLS certificates, depending on the certificate types contained in the file). If a certificate with the same subject name already exists (e.g. when replacing an expired certificate), the new certificate is uploaded alongside the original certificate (unless the issuer and serial number details are identical, in which case the existing certificate is updated with the new contents from the file).

- **View or modify an existing certificate**: select the **Subject name** of the certificate you want to view. The decoded certificate data is shown.

  If required, you can modify the PEM-formatted certificate data and select Save.

- **Download all certificates**: select Export. A ca-certificates.pem file containing all of the custom-added certificates in PEM format is created and automatically saved to your local file system.

- **Delete one or more certificates**: select the boxes next to the certificates to be deleted, and from the Action drop-down menu select Delete selected Trusted CA certificates and select Go.