MiVoice MX-ONE

Certificate Management - Operational Directions

Release 7.2
November 6, 2019

Contents

General

Certificates are used to authenticate the communicating parties. In MX-ONE the certificate handshake procedure is carried out via TLS.

openssl is used by the key management scripts and is also used as a component doing the TLS handshake and handling the encode/decode of messages.

In SIPLP and CSTServer openssl is handled via the SIP stack, reSIProcate, for securing SIP signaling. In IPLP/TLP65 openssl is handled via the common H323 stack. These programs are in this document called “TLS supporting programs”.

For more information, see http://www.openssl.org/.

Certificates and Key Management

X.509 v3 is the standard used for the certificates.

There are two types of certificates according to X.509 v3; Certificate Authority (CA) and signed certificate. The CA has the X.509v3 extension->X.509v3 Basic Constraints set to CA:TRUE while the same parameter for the signed certificate is CA:FALSE.

The root CA can sign another CA and thereby create a chain of trust. TLS clients trusting the CA will thereby trust the servers signed by the Intermediate certificate. The certificate management tool can not create chain of trust between CAs, but it supports importing intermediate CA and root CA.The MX-ONE Enterprise CA is only a root CA.

Both client and server certificate are signed certificates. In 802.1x we the phone will have client certificates as they are validated as clients of the radius server who validate which clients that shall have network access. The MX-ONE Service Node is validated as a server, in relation to other servers or phones.

The certificate management tool allows to either create a generic wild card certificate distributed to all MX-ONE servers (auto) and/or a local certificate for a certain server which can use a certain root CA, which may be required for a certain TLS SIP route.

MX-ONE stores certificates in the native format for openssl, PEM, which is an encode 64 format. The format is easily verified when looking at the file using an ASCII editor as the certificate bob is wrapped with ------BEGIN CERTIFICATE---- and -----END CERTIFICATE-----.

The PEM file used by MX-ONE (by the TLS supporting programs) is a “PEM with Certificate chain” stored at /etc/opt/eri_sn/certs/eri_sn_cert.pem. This file includes in the following order, the server certificate private key, the server certificate and the CA (Certificate Authority). Note, the CA private key is not needed.

TLS Handshake

A certificate is used for public-key cryptography, where each certificate has a private key and a public key. The sender (client) of a message will start a TLS handshake where the initial message is encrypted with the server certificate public key. Only the owner of the server certificate private key may decrypt the message.

The initial message may include a symmetric secret session key, which will be used for this TLS session. Session keys are required for persistent TLS on phones to work as the server will reuse the session originally setup by the phone when sending messages.

A simple TLS handshake is where only the client checks the validity of the server. This makes sense when the client is a web browser where the focus is to protect the web client.

In order to better secure the MX-ONE server it makes sense to validate the client's certificate in the TLS handshake

Mutual TLS (MTLS) is a TLS handshake where the TLS server sends its server certificate as normal, but also request the client to send its certificate. In a SIP route both endpoints will have certificates and can honor this handshake, but IP phones that uses persistent TLS only needs the public root CA and have traditionally not made use of MTLS even though it is possible to enable for some SIP clients.

SIPLP will not support MTLS, but as TLS is terminated at the destination of the IP header, an SBC or Security gateway may be terminating TLS and can direct the package according to the destination in the SIP request URI header. An SBC or security gateway may support MTLS for SIP phones if the client certificate is validated based on that it is signed by the same CA as the SBC server certificate and that it is not expired.

SIPLP (reSIProcate) has the same check for the client certificate as it would towards a server certificate, which is to validate the destination against the DNS. As phones usually acquire their IP address via DHCP, the IP can not be part of the client certificate (CN or SAN:IP).

In SIPLP, MTLS is configured per LIM. In order to avoid conflicts with phones, you can setup a LIM dedicated to a SIP route were MTLS is required. LIM configuration, */etc/opt/eri_sn/ip_telephony.conf*{TlsClientVerificationMode: none|optional|mandatory}.

For an outbound message to a SIP route, reSIProcate validates the other party's server certificate in a TLS handshake. An outbound SIP message request URI host must be resolvable in a DNS server and the host shall match the certificate Common Name (CN) or Subject Alternative Name SAN:IP or SAN:DNS of the TLS destination. If the request URI host is an IP address, the certificate of the TLS destination must contain that IP address in either CN or SAN

When using auto MX-ONE server certificate, the SAN:DNS is a (Fully Qualified Domain Name) FQDN format, which shall match the server's DNS A or AAAA Record pointing to its IPv4 and IPv6 address respectively.

As other option is to use DNS SIP SRV Records. The SIP service request, _sip._<protocol>.<SRV DNS name>, will result in a list of access points offering SIP service. In MX-ONE this would be a list of LIM IP addresses or FQDN DNS names.

Preparation and Work Process

In MX-ONE all certificate management is handled via *mxone_certificate* and *mxone_-maintenance* script. This script can be run from root or from mxone_admin as sudo -H (*mxone_certificate* or *mxone_- maintenance*). The certificate handling for MX-ONE web based management is part of “Web server config”, which is described in *AD Authentication, Description*. The certificate handling for securing call control signaling, phone configuration and phone key management is run from mxone_certificate and is further described in this document. TLS is configured for the TLS supporting program units (see section *General, page 3* or the `mxone_certificate` command).

NOTE: Consult your PKI or IS/IT department as they may want to sign your server certificates instead using an MX-ONE CA.

The output of certificate management is to activate TLS for MX-ONE VoIP. What certificates to create are described in chapter *Strategies for certificate handling, page 7* . How to execute the strategies is described in chapter *Execution, page 9* .

No extra preparation is needed to use the HTTPS feature for XML keys for SIP phones.

Additional preparation for VoIP communication when certificates are activated:

1. The license VOIP-SECURITY is not enabled in trial licenses due to export control on media encryption. For SIP the license is only required for secure media (SRTP). However, H.323 also requires this license to enable secure signaling (TLS). Check whether VOIP-SECURITY is set to yes using command (`license_status`).
2. Follow the instructions in this document to create server certificate.
3. When a certificate is created by the script, the CA is extracted and stored at */etc/opt/eri_sn/certs/ca.pem*.
   – Procedure for TLS on terminals. No handling is required if a commercial CA (Verisign, Thawte etc), preloaded on the phone FW, was used to sign the server certificate. Otherwise the CA, */etc/opt/eri_sn/certs/ca.pem*, needs to be copied to the key storage of the terminal. This is usually the same directory as where the terminal’s configuration files are stored. See Installation Instruction of your SIP or H.323 model. As the terminal normally only will have a CA (not a client certificate), the TLS method to be used is ‘persistent TLS’. This means that the TLS session setup by the client is kept as long as the terminal is logged on and that negotiated session keys are reused by the server when signaling to the client.
   – Procedure for TLS on route (trunk). The same root- or intermediate certificate needs to sign the Call Managers’ server certificates accessing the route.
4. Enable secure VoIP by entering the following commands:
   – `media_encryption_enable -type extension`
   – `media_encryption_enable -type extension` (And other type of end-point should be secured (such as `media_encryption_enable -type route` and `media_encryption_enable -type intermgw`).
   – `data_backup`
   – `reload --system`
   (Or, run *reload -u <TLS supporting program>)* to enable TLS for specific program units only.

5. Check that MX-ONE has TLS ports activated: SIP - port 5061 is used for both extension and trunk. H.323 - port 1300 is used for extension and 1301 is used for trunk.

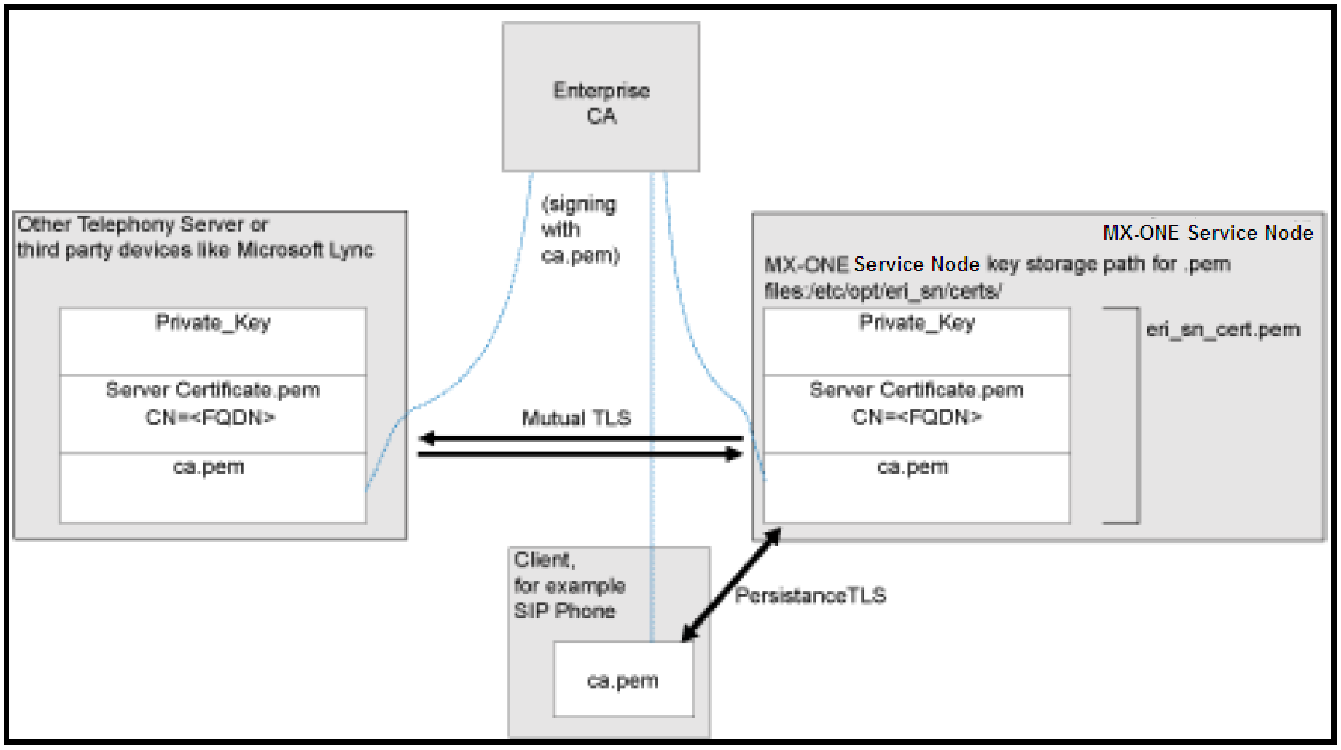
Use the following command to list the listening TCP ports.

- *netstat -ltn tcp <own-server-ip-address>:5061 LISTEN*

NOTE: TLS is based on TCP.

Strategies for Certificate Handling

Figure 3.1: Solution Overview


Enterprise CA
(signing with ca.pem)
Other Telephony Server or third party devices like Microsoft Lync
Private_Key
Server Certificate.pem CN=<FQDN>
ca.pem
Mutual TLS
MX-ONE Service Node
MX-ONE Service Node key storage path for .pem files:/etc/opt/eri_sn/certs/
Private_Key
Server Certificate.pem CN=<FQDN>
ca.pem
eri_sn_cert.pem
Client, for example SIP Phone
ca.pem
PersistanceTLS

SIPLP only accepts inbound SIP messages that are directed to the MX-ONE system. The official identification of the MX-ONE system are the LIMs host name, limX,. where X is the LIM number. So if a SIP phone shall logon to LIM2 in domain, mx.corp.net, it shall be addressed as lim2.mx.corp.net, matching name in the SAN field.

As the server certificate is to be used only for the MX-ONE LIMs, the MX-ONE system should be a sub-domain in the corporate network.

Setup SIP Route

For SIP route, both end points needs to trust the CA, signing their server certificates.

If your company have a Public Key Infrastructure (PKI), responsible for authentication of your network elements, choose to send a certificate request to their CA according to chapter Execution, “Create configuration file with default setting”.

If two MX-ONE systems shall have TLS SIP route between them, one of the system shall be chosen to act as PKI and have the root CA. See Execution, “Using MX-ONE as PKI”.

The route is setup towards the DNS FQDN of the call manager’s host (For SIP trunk this is set by command, *sip_route -uristringX*), and the TLS handshake will verify that the server certificate’s CN or SAN:DNS name field matches the DNS FQDN of the destination of the route.

The default MX-ONE server certificate will have the official FQDN DNS name as SAN:DNS.

if the official FQDN is not known by the DNS used in the network, the server certificate must match what the LIM is addressed as.

Assume that the FQDN used for a LIM is malmo.mx.corp.net, this must be a domain accepted as being part of MX-ONE, sip_domain --local -domain malmo.mx.corp.net. If the other SIP route server supports SAN the default MX-ONE System server certificate may be modified and you can add an extra SAN:DNS: malmo.mx.corp.net. See Execution, “Create configuration file with default setting”. From the default config file, you may do the changes required before creating the CSR based on the modified configuration file.

A valid approach if there is no DNS is to assign the server certificate using the IP address of the MX-ONE Service Node as CN. Per default SAN:IP is already set in an MX.ONE server certificate.

Execution

The chapters in the execution describes the headings in the certificate management tool

Create and Install Default Certificate and Activate TLS

auto - Create and install default certificate and activate TLS

This script combines the default choices of the following individual choices:

- Manage Root Certificate -> Create Root Certificate
- Manage Server Certificate -> Create Server Certificate

You will be prompted to enter a root CA password and a server certificate password. The validity is set to 364 days for both the root CA and the server certificate.

The root certificate created is an Enterprise CA (MXOneEnterpriseCA), which signs the server certificate in the next step.

Per default, the server certificate will be the same for all LIMs. The Common Name (CN) is *.<mxone-domain>, i.e. a wild card certificate. All LIMs FQDNs and IP addresses are set in the extension, Subject Alternative Name (SAN), IP and DNS

When the certificate is activated you are asked to check what is described in chapter “Preparation and work process”.

Create Root Certificate

Manage Root Certificates>Create Root Certificate

Create or Renew Certificate

The definition for renewal of a server certificate is that only validity dates (“Note Before” and “Not After” date and time) are updated. The same procedure applies for updating a certificate as for installing a certificate. The CA is kept consistent as long as the CA private key is kept the same.

Create default server certificate signed by MXOneEnterpriseCA:

- **Manage Server Certificate>Create Server Certificate**

Create a server certificate based on manual changes to the configuration file being input to the signed server certificate:

- Manage Server Certificate>Create configuration file with default settings

Manage Certificates, Create CSR

The Certificate Signing Request (CSR) contains the 509 v3 headers that you want to be signed as the server certificate to use.

Manage Certificates>Create CSR (Certificate Signing Request)

The script will use a certificate private key used earlier or create a new one which is tied to the CSR. Only this private key can be used with the signed server certificate. The script will create both the configuration file which and a CSR file which is the output of the configuration file. If the collected data for CN and SAN needs to be changed, you can change the configuration file and choose “Create CSR from configuration file”.:

Send the *certificate_request.csr* to the Certificate Authority (CA). The CSR file is in encode 64 format. Sometimes the PKI uses a web page to collect CSR, then you often paste the content, which starts with --BEGIN CERTIFICATE REQUEST -- and ends with --END CERTIFICATE REQUEST--

If your Enterprise CA is installed on a Microsoft Server, you can order a server certificate by browsing to the server with to http://<your company’s enterprise ca/certsrv.

Import of Signed Certificate

Create an import directory.

Here you put the imported signed certificates and the CA. When using pem files, you have one server certificate pem file and a CA pem file.

If you were to use a commercial CA or Mitel CA (which Mitel Canada may sign) the CA pem file actually is both an Intermediate CA and a CA which have a trust chain, where the root CA has signed the Intermediate CA. The file has two certificates, i.e. two blobs, --BEGIN CERTIFICATE --, --END CERTIFICATE--

If another MX-ONE has signed the certificate, the CA pem file will only contain a root CA.

The import signed certificate script also supports other certificate formats. Chained certificate (*.p7b) PKCS7 in DER format, containing both server certificate and the CA certificate(s) in one file. PKCS12 (*.pfx) which is a complete signed certificate and does not require a CSR. Input may just be an e-mail as the PKCS12 includes the signed certificate private key. PKCS12 is mostly used when deploying certificates for clients which cannot create a CSR.

The following two scripts are used to install the certificates:

- Certificate Management -> Import Signed Certificate
- Certificate Management -> Install Certificate

The output of the installed certificate which is used by the TLS-supporting programs is */etc/opt/eri_sn/certs/eri_sn_cert.pem* and the configuration file */etc/opt/eri_sn/tls_tele-phony.conf.* The PEM encoded file, *eri_sn_cert.pem*, contains the following concatenated certificate files in the exact order;

- The **private_key** obtained during the CSR generation
- The signed server certificate.
- The single or multiple intermediate CA. If multiple, the secondary should be before the primary.
- The root CA.

NOTE: Dependent on your certificate provider you might have to assure the "-----BEGIN CERTIFI-CATE-----” and “-----END CERTIFICATE-----” statements are on their own separate lines in the Allcerts.pem file.

Manage TLS in MX-ONE

Because the units will be reloaded, perform a data backup using the `data_backup` command.

Enable the use of installed certificates:

- Manage TLS in MX-ONE -> Configure MX-ONE to use TLS

Disable the use of installed certificates:

- Manage TLS in MX-ONE -> Configure MX-ONE not to use TLS

Complete the change by reloading the program unit(s) supporting and using TLS:

reload -u SIPLP,IPLP,TLP65,CSTServer.

Mitel
Powering connections

mitel.com