TLS Support

Transport Layer Security (TLS) provides mechanisms to secure your phone traffic. Typically it is used to protect your **SIP** and HTTP connections from eavesdropping and tampering.

TLS handshake overview

Initially, when a client wants to establish a secure connection to a server, some parameters are negotiated, this phase is called the handshake procedure. One of these parameters is the supported cipher suites defining the encryption algorithms required to encrypt the traffic. In a next step, the server sends a digital certificate that contains its public encryption key, the trusted certificate authority (issuer) and a signature of the certificate.

The client may reject the connection if the identity of the server or the issuer is unknown. Usually, the client has a list of trusted server certificates and a list of trusted certificate authorities (CA). If the server certificate is considered trusted or if the identification from the server is signed by one of these CAs, the client typically permits the connection and delivers a random number encrypted with the server's public key to the server. As only the server can decrypt this number with its private key, it is used as the session key to encrypt and decrypt the traffic.

Optionally the server can "**ask**" the client for the certificate. In that case the client must send a TLS certificate. Now the server can authenticate or deny the request based on the client certificate.

If the identification from the server is signed by one of these CAs, the client typically permits the connection and delivers a random number encrypted with the server's public key to the server. As only the server can decrypt this number with its private key, it is used as the session key to encrypt and decrypt the traffic.

Your phone acts as a TLS client in various cases:

- **SIP** connections with **TLS** as transport protocol
- Provisioning requests
- **Action URL** HTTPs requests
- HTTPS requests to an URL triggered by function keys
- Minibrowser applications served by an HTTPS server
- **LDAP** server providing the business contacts

When TLS is used a mutual authentication of the client and the server can be performed by the phone and the server.

Setting-Up the phone for TLS

The built-in certificate

Every Snom phone (except the old 3xx series) is produced with a built-in TLS certificate on board.

Every device certificate is issued by the **Snom Certification Authority**. The built-in certificate contains the the device MAC address into the DN x.509 attribute.

Thanks to the built-in certificate, a server using the TLS protocol can:

- verify the issuer of the certificate, checking the certificate signature against the Snom CA: in other words the server can make sure that the request comes from a Snom phone

- verify the DN of the client certificate, checking the MAC within the offered certificate and the requested resource. The server can authorise the specific device to the resource

Uploading the Phone Certificate

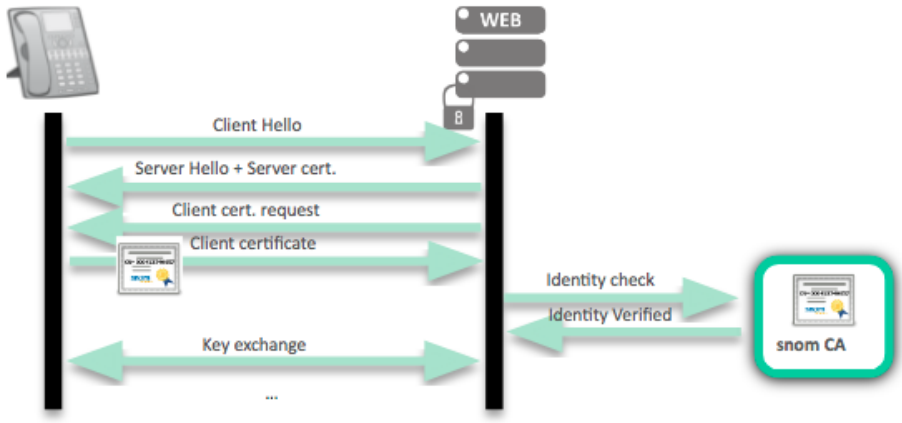
In fact, no special configuration is required. Every out-of-the-box Snom phone has a certificate built-in to the firmware along with its private key. You can retrieve this certificate by pointing your browser to your phone's web interface via secure HTTP (HTTPS). Usually you will have to confirm that you trust this identification since your browser could not identify your phone. After confirming, your connection to the web interface will be encrypted.

You can also specify your own client certificate in the webserver_cert setting. In this case, make sure you are provisioning the certificate and the key in a trusted environment, see below how to provision a custom certificate.

Authentication of the devices

The TLS server can be configured to check the client identity via the TLS authentication: as described into the previous section, during the TLS handshake, the server asks the client for the certificate. The Snom phone will send the built-in certificate, now the server can check the issuer of the client certificate and permit or deny the request.

Since device MAC address is mentioned into the built-in certificate CN, the web server can also authorize or deny the request based on the requested URL and the presented client certificate.



In order to configure the client authentication, you will need to import the Snom Certification Authority certificates into your TLS server.

> **IMPORTANT**
>
> Starting with fw 8.9.3.60 and **10.1.x** we support the **SHA-2** algorithm, new phone models are already equipped with a **SHA-2** built-in certificate. In case you are using devices with SHA-2 certificate you will need to import also the Snom SHA-2 CA public certificate.

Phone model	Supported Hash algorithm	Built-in certificate
Snom 300	SHA-1	SHA-1
Snom 320		
Snom 370		
Snom PA1		
Snom MP		
Snom D120	SHA-1, SHA2	SHA-2
Snom 710 / D710	SHA-1, SHA-2 (fw >= 8.9.3.60)	SHA-1
Snom 715 / D715	SHA-1, SHA2 (fw >= 8.9.3.60)	SHA-1
Snom D712	SHA-1, SHA2 (fw >= 8.9.3.60)	SHA-1
Snom 720	SHA-1, SHA2 (fw >= 8.9.3.60)	SHA-1

Snom 760	SHA-1, SHA-2 (fw >= 8.9.3.60)	SHA-1
Snom D725	SHA-1, SHA2 (fw >= 8.9.3.60)	SHA-1
Snom D735	SHA-1, SHA2	SHA-2
Snom D745	SHA-1, SHA2 (fw >= 8.9.3.60)	SHA-1
Snom D765	SHA-1, SHA2 (fw >= 8.9.3.60)	SHA-1
Snom D785	SHA-1, SHA2	SHA-2
Snom D305	SHA-1, SHA2 (fw >= 8.9.3.60)	SHA-1
Snom D315	SHA-1, SHA2 (fw >= 8.9.3.60)	SHA-1
Snom D345	SHA-1, SHA2 (fw >= 8.9.3.60)	SHA-1
Snom D375	SHA-1, SHA2 (fw >= 8.9.3.60)	SHA-1
Snom D385	SHA-1, SHA2	SHA-2

Warning

The following devices: 300, 320, 370, PA1 and MP are produced with a common Snom certificate, so the built-in certificate isn't unique per-device and doesn't mention the MAC address into the CN

Upgrade to SHA-2

Upgrading a SHA-1 native device to SHA-2 requires a firmware patch specific per device, this patch contains the SHA-2 certificate and must be requested to the Snom tech support.

Snom Certification Authority public certificates

From the following links you can download:

SHA-1 Snom root CA

SHA-1 Snom root CA with intermediate certificates

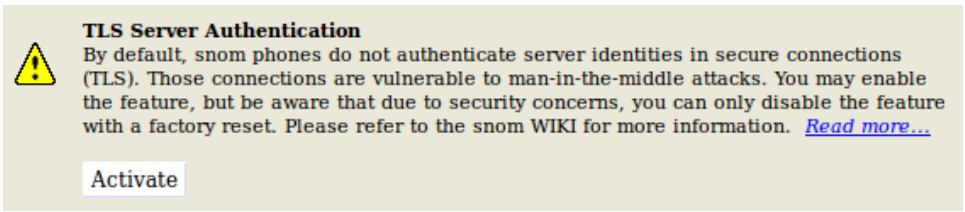
As SHA1 gets widely deprecated and SHA2 becomes more and more the standard, support for it is getting mandatory in real world deployments.

SHA-2 Snom root CA support starts with **v8.9.3.60** and **v10.1.X**.

SHA-2 Snom intermediate

Server Authentication

- ***Version 8.x***
  In version 8.x, you do not need to worry about the server identification. Snom phones do not verify server identities by default. Starting with FW version 8.2.30 you can enable a setting to require verification of server certificates though. You can activate the feature on the certificates page of the web interface:



NOTE

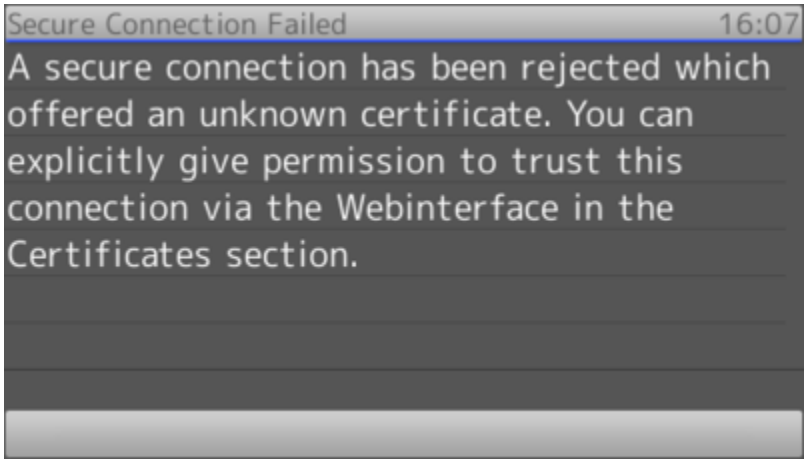
Please be careful when enabling this feature. The phone will reject all secure connections from peers offering an unknown certificate that could not be verified by one of the built-in CAs of the Snom phone. Please refer to the Certificate Authorities tab to see which authorities are supported by the phone. Due to security concerns, you can only disable this feature by resetting the phone to the factory defaults.

- ***Version 10.x***
  As of version 10.x, Snom has decided to force the server identity verification. This verification cannot be disabled because that would create a security weakness for the Snom Phones.

In addition to the server certificate a phone can also verify the identity of the server checking if the certificate DN matches the server name FQDN. This can be done via the setting check_fqdn_against_server_cert. The behaviour of the server name validation can be modified via the setting host_name_validation_flags.

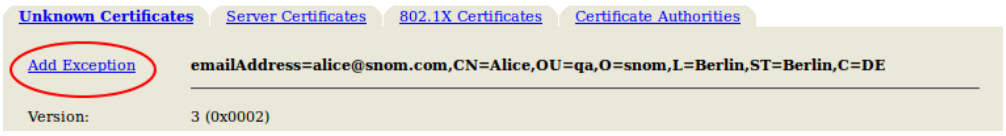
Adding Unknown Certificates

The phone will reject a connection if it cannot verify the identification with the certificate delivered by the server. If this happens, the following notification will appear on the screen:



A certificate is trusted if its signature is signed by a certificate authority. Snom has pre-installed a list of CAs which are listed on the *Certificate Authorities* tab of the *Certificates* page. This list is automatically updated based on the Mozilla official list, at every new firmware upgrade.

All rejected certificates are listed in the *Unknown Certificates* tab. If you want to permanently trust a certificate you can add it as an exception:



After adding it as an exception in the *Server Certificates* tab a connection from a peer using this certificate will no longer be rejected.

Manually Uploading Certificates

In admin mode, you can manually upload certificates signed by one of the phone's accepted authorities or server certificates in the *Unknown Certificates* tab. Every attempt to upload an unknown certificate will fail. In case of upload failures, please refer to the log and make sure your certificate is in DER format and is signed by one of phone's authorities or server certificates.

Uploading Server Certificates via Provisioning

You can upload a server certificate using auto provisioning. For this, the download link to the certificate needs to be placed within the XML.

The certificates settings (<certificates> tag) contains the trusted server certificates. This XML tag can be used either inside the <settings> tag or as an individual XML file whose URL is listed inside <setting-files> tag

The tag contains an attribute with the URL of the certificate file to fetch:

```
<certificate url="http://some.url/certificate.der" />
```

⚠ Please note that the download of the certificate is delayed after all provisioning xml files have been loaded and processed.

Beginning with firmware release 8.7.5.52/8.9.3.41 a second variant of this tag is supported, where the content of the certificate file is included as a base64 encoded string:

```
<certificate type="base64">...</certificate>
```

The benefit of this variant is that the certificate is immediately available after processing the line in the provisioning XML.

INFO

You can get the base64 encoded certificate out of the PEM format, removing the BEGIN / END taglines:

```
-----BEGIN CERTIFICATE-----
```

```
-----END CERTIFICATE-----
```

If you decide to provide the download link(s) in an additional XML file, this is how it could look like:

```
<certificates>
 <certificate url="http://192.168.1.101/trusted_cert1.DER" />
 <certificate url="http://192.168.1.101/trusted_cert2.DER" />
</certificates>
```

The attributes url and base64 can be combined in the same file, as the following example:

```
<?xml version="1.0" encoding="utf-8" ?>
<certificates>
   <certificate url="http://192.168.1.101/trusted_cert1.DER" />
   <certificate url="http://192.168.1.101/trusted_cert2.DER" />
   <certificate type="base64">
 MIIG9zCCBd+gAwIBAgIIUf9BRQhu9JwwDQYJKoZIhvcNAQELBQAwdTELMAkGA1UE
 BhMCREUxJTAjBgNVBAoTHFQtU3lzdGVtcyBJbnRlcm5hdGlvbmFsIEdtYkgxHzAd
 BgNVBAsTFlQtU3lzdGVtcyBUcnVzdCBDZW50ZXIxHjAcBgNVBAMTFVRlbGVTZWMg
 QnVzaW5lc3MgQ0EgMTAeFw0xODA0MTkxMDQ3MTlaFw0yMDA3MTkyMzU5NTlaMIGl
 MQswCQYDVQQGEwJERTEcMBoGA1UEChMTRGV1dHNjaGUgVGVsZWtvbSBBRzEdMBsG
 A1UECxMUU0lQLVRydW5rLnRlbGVrb20uZGUxEjAQBgNVBAsTCVNJUC1UcnVuazEY
 [...]
 [...]
 MBYGA1UEAxMPdGVsLnQtb25saW5lLmRlMRwwGgYDVQQIExNOb3JkcmhlaW4tV2Vz
 dGZhbGVuMQ0wCwYDVQQHEwRCb25uMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
 CgKCAQEAwl6iq3B9EBJe9z34yCikyfla+ZSKE4gQUpo3hLLz2zXKiQildQc6qB6g
 MzYvwjVJI64t5S2CbqEybBtrPn0FiziseDRZKnt+bkuIqZNPOYtkE1akGgdjIieV
 Wjg6oD37+BCCqyq60gq0FbsGgjlwiNb68jL7dUXzRi2lgxtwk86+g/QFg+3rQts/
 3GREGNhwVbu4mUIrnnphaUA8BnUeGi++8j9d21ZF/uW2pIQqVBItYDflBee+qGfk
   </certificate>
</certificates>
```

⚠ All provisioned certificates need to be signed by one of the phone's server or authority certificates (**this restriction was removed starting with firmware revisions 8.7.5.52/8.9.3.41**). Make sure the supplied certificates are in DER format.

Tech Notes

Phone's Web Server Certificate

The phone uses the built-in certificate also as a certificate for the web user interface when accessed via HTTPS. Currently Snom phones come with SHA1 (RSA, 1024 bit key length) or SHA2(RSA, 2048 bit key length) certificates.

Starting from firmware 8.9.3.40 it is possible to switch the certificate via the setting phone_cert_type to a SHA256 certificate .

Supported Cipher Suites

Phone	Firmware	Protocol Version	Key Exchange	Authentication	Block Cipher	Key Length [bit]	Mode of Operation	Message Authentication Code	Notes
snom300	8.7.3.25.9	SSL 3, TLS 1.0	DH, RSA	NULL, RSA	3DES, DES, NULL, RC4	56, 128, 168	CBC	MD5, SHA1	Firmware version not supported anymore (EoL). It is highly recommended to upgrade to a higher version.
snom320									
snom360									
snom370									
snom710									
snom720									
snom760									
snom820									
snom821									
snom870									
snomMP									
snomPA1									
snom300	8.7.5.35	TLS 1.0	DH, RSA	NULL, RSA	3DES, AES, DES, NULL, RC4	56, 128, 168	CBC	MD5, SHA1	* Since firmware version 8.7.5.49 the snom710 has the same TLS-properties as the snom715 (see below). * Starting with version 8.7.5.57 3DES, DES, NULL and RC4 have been removed from the supported block ciphers.
snom320									
snom370									
snom710									
snomPA1									
snom715		TLS 1.0, TLS 1.1, TLS 1.2	DH, ECDH, RSA, SRP	DSS, ECDSA, RSA	3DES, AES, Camellia	128, 168, 256	CBC, GCM	SHA1, SHA256, SHA384	
snom720									
snom725									
snom760									
snomD765									
snom821									
snom870									
snomMP									
snomD305	8.9.3.52	TLS 1.0, TLS 1.1, TLS 1.2	DH, ECDH, RSA	ECDSA, RSA	AES	128, 256	CBC, GCM	SHA1, SHA256, SHA384	
snomD315									
snomD345									
snomD375									
snomD745									
snomD120	10.1.20.0 10.1.33.33	TLS 1.0, TLS 1.1, TLS 1.2	DH, ECDH, RSA	ECDSA, RSA	AES	128, 256	CBC, GCM	SHA1, SHA256, SHA384	
snomD305									
snomD315									
snomD345									

snomD375									
snomD385									
snomD712									
snom715									
snom725									
snomD735									
snomD745									
snomD765									
snomD785									

Limitations

- No support of Certificate Revocation Lists (CRL).
- No support for Certificate Lifecycle Management (e.g. SCEP).

Further Information

- VoIP Essentials
- Visit the Snom Forum
- Open a support ticket
- Find a local partner

Related articles

- <certificates> tag
- <dialplan> tag
- <functionKeys> tag
- <gui-languages>, <web-languages>tag
- <phone-settings> tag
- <ReplacementPlan> tag
- <Setting-Files> tag
- <tbook>,<phone-book> tag
- <uploads> tag
- Action URLs
- Ad-Hoc Conference - V10
- assign-action
- Basic setting provisioning via DHCP
- BLF - Busy lamp field
- Call Features