Managing Certificates

In this section:

Related articles:

- Generating PKI Certificates
- Certificate Expiry Check - CLI
- Security Configuration - Cert Expiry Check (EMA)
- PKI Security - CLI
- Security Configuration - PKI (EMA)

The SBC Core supports Transport Layer Security (TLS) enabling SIP and HTTP applications to securely communicate on an insecure network, and to reliably verify the identity of a user via digital signatures. SIP/TLS applications may act as a TLS server or client for their respective TLS sessions. HTTP/TLS applications as a web server (for the EMA) always act as a TLS server for a TLS session. As a TLS session is being negotiated, the TLS server presents its digital certificate to its client for authentication and for encryption of client-generated shared secret. Sometimes, a TLS server may also request the client to send its certificate for mutual authentication as in the case of the SIP peering and the management access to the EMA that requires Common Access Card (CAC) based public key (PK) authentication. In any case, a TLS connection's security is not established until every individual certificate on the chain presented by the peer device is successfully authenticated and Online Certificate Status Protocol (OCSP) validated.

The status of the certificates corresponding to established ongoing TLS sessions, however, may change over the lifetime of the TLS session, especially when the sessions are long-lived. The SBC periodically checks all certificates and trust chains associated with ongoing sessions, and then terminates any ongoing sessions if the corresponding certificates are revoked, no longer trusted, or expired.

Remote certificates are installed in the SBC for presentation along with local certificates, installed as trust anchors for the verification of credentials presented by peer devices, and installed as the OCSP responder certificates for the verification of signed OCSP responses. These installed remote certificates are not automatically renewed and thus can expire. The SBC gives the user an alert before any installed certificates are near expiration so the user can take action against them.

Certificates reside in the `/opt/sonus/external` directory.

Certificate Types

Local-Internal Certificates

In previous SBC versions, the RSA key pairs and Certificate Signing Request (CSR) for SBC Core platforms were generated on an external workstation. The CSR was then submitted to a Certificate Authority, and the resulting certificate was received back from the CA, copied onto the workstation, and combined with the private key in a PKCS#12 file which was used to install the key pair and certificate onto the SBC.

The SBC application can now generate and install RSA key pairs and generate Certificate Signing Request (CSR) on the SBC 5000 series system itself. The certificate request is sent to a CA, and the issued certificate is then installed on the SBC. The local-internal certificate option simplifies the certificates and keys managing process and also provides more security since the private key never leaves the SBC. For steps to configure local-internal certificates, refer to Generating PKI Certificates.

Local Certificates

Local certificates are credentials belonging to the local system itself, which it presents to peers in order to prove its identity. You have to download local certificate files to the system before installing the certificates.

Remote Certificates

Remote certificates are credentials belonging to Certificate Authorities (CA). The copies of these certificates are installed in the SBC Core because they are part of a chain of certificates the local system will present to peers, or because the corresponding CAs are trust anchors for the

local system. Certificates belonging to non-CA remote systems should also be installed as trust anchors in this manner.

The Certificate Authority (CA) certificates and trusted remote certificates contain public key certificates; they do not contain the private keys. The CA certificates and remote certificates are Distinguished Encoding Rules (DER) format files; a method for encoding a data object (such as an X.509 certificate) which uses a digital signature to bind together a public key with an identity.

The SBC imports these certificates from Distinguished Encoding Rules (DER) formatted files.

> **Note**
> The SBC supports a maximum of 4,096 TLS certificates/CAs (both local and remote).

RSA Key Pairs and Certificate Signing Requests

In previous SBC versions, the RSA key pairs and Certificate Signing Request (CSR) for SBC Core platforms were generated on an external workstation. The CSR was then submitted to a Certificate Authority, and the resulting certificate was received back from the CA, copied onto the workstation, and combined with the private key in a PKCS#12 file which was used to install the key pair and certificate onto the SBC.

The SBC application can now generate and install RSA key pairs and generate Certificate Signing Request (CSR) on the SBC system itself. The certificate request is sent to a CA, and the issued certificate is then installed on the SBC. This feature simplifies the certificates and keys managing process and also provides more security since the private key never leaves the SBC. To configure PKI certificates, see Generating PKI Certificates.

Certificate Re-Check and Expiry Warning

The SBC has a configurable option to check for expired certificates, trust anchor validity, and if certificates have been revoked if OSCP is enabled. The re-check rate is configurable via CLI from every 8 hours up to every 30 days in increments of 1 hour. The default value is once per 24 hour period.

Upon failure of any one of the checks (for example, the certificate is no longer valid), the SBC terminates the TLS session and logs a MAJOR level event (sonusSbxFailedCertificateReCheck) to alert the user. The one exception will be if OSCP is enabled but SBC does not receive revocation status of successful.good or successful.revoked, the corresponding TLS session continues for SIP/TLS. The SBC Core supports SHA-256 Cryptographic Hash Algorithm for certificate verification.

The SBC Core also includes a configurable option via CLI to set certificate expiry warning rates.

- Use the `expiryWarningThreshold` parameter to set the number of days prior to a certificate expiration to send a warning message.
- Use `expirationPeriodicWarning` parameter to set the frequency, in days, for sending periodic warning reminders once the `expiryWarningThreshold` has been met.

The SBC logs an event in the DBG and SEC logs at a high severity level when a local or remote certification installed on the SBC is within 60 days of its expiration date. The event repeats weekly until the certification is replaced or deleted (even after it has expired).

> Disabled certificates are not included in the certificate expiry warning check.

For configuration details, refer to:

- CLI: Certificate Expiry Check - CLI
- EMA: Security Configuration - Cert Expiry Check

Subjective Alternative Name

The Subjective Alternative Name (SAN) is an X509 version 3 extension which allows an SSL certificate to specify multiple names that the certificate should match against to allow the user to secure a large number of domains with only one certificate.

> The SAN may contain e-mail addresses, IP addresses, regular DNS host name; however, the SBC currently supports DNS Host Names only.

The CLI syntax to enter multiple DNS names is shown below:

```
> request system security pki certificate <certName> generateCSR csrsubj <csrSubj> keysize <keySize>
subjectAlternativeDnsName <dnsName1, dnsName2...>
```

For configuration details, refer to:

- CLI: PKI Security - CLI
- EMA: Security Configuration - PKI