TLS for Signaling

In this section:

Related articles:

- DTLS Profile - CLI
- TLS Profile - CLI
- EMA TLS Profile - CLI
- Security - Dtls Profile (EMA)
- Security Configuration - TLS Profile (EMA)

Overview

The SBC Core supports the exchange of SIP signaling over Transport Layer Security (TLS), an IETF protocol for securing communications across an untrusted network. Normally, SIP packets travel in plain text over TCP or UDP connections. Secure SIP is a security measure that uses TLS, the successor to the Secure Sockets Layer (SSL) protocol. TLS operates just above the transport layer (Layer 4) and provides peer authentication, confidentiality and message integrity.

The SBC supports TLS versions 1.0, 1.1 and 1.2 with server-only authentication (in which only the server is authenticated at the TLS layer) and mutual authentication (in which both the TLS client and server are authenticated at the TLS layer). TLS is an effective measure to a number of threats including theft of service, disruption of service, compromise of confidentiality, and compromise of service integrity.

SIP over TLS may be independently configured on each hop between SIP devices. SIP transport type selection is typically configured via the IP Signaling Profile, and may also be provisioned on the SIP trunk group or identified via a DNS lookup.

Note
Sonus recommends using the highest TLS version supported by both the SBC and the peer equipment.

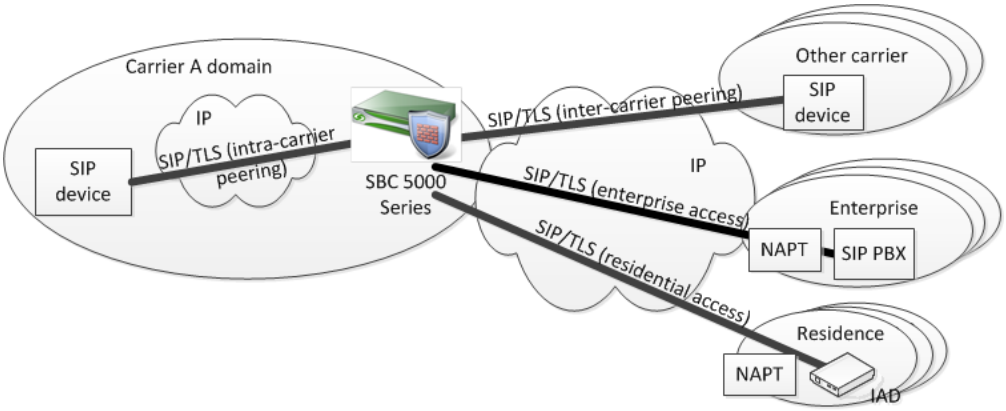
Note
If a zone's `sipSigPort` is configured for `transportProtocolsAllowed` = `sip-tls-tcp`, the SBC increments the configured `port Number` by 1 and uses it as the new port number for SIP over TLS signaling. The SBC then opens a TCP socket for SIP over TLS for the new TCP port number.

Example: When `sipSigPort` is configured with a `portNumber` of 5060 and `transportProtocolsAllowed` = `sip-tls-tcp`, the SBC listens on TCP port 5061 for SIP over TLS.

Usage Scenarios and TLS Roles

The SBC uses SIP over TLS in several scenarios as illustrated in the figure below .

Figure 1: SIP over TLS Usage Scenarios



Note
In most scenarios, the SBC Core does not support ECC certificates for TLS Handshake. Specifically, the SBC Core does not support ECC certificates for TLS handshake when it acts as a TLS “server-only,” although it can support the certificates when acting as TLS client in the configured “server-and-client” role.

The table below describes the interrelationship between each of these scenarios, the TLS role (server or client/server), and the authentication requirements.

Table 1: TLS Usage Scenarios

Usage Scenario	Usage Description	TLS Role	Authentication Requirements
Residential Access	Between a subscriber SIP User Agent (UA) and an SBC.	Server	Server-only authentication. This is intended for use in conjunction with authenticated SIP registration. A peer is blocked from using any services until a successful SIP registration is performed. A separate registrar is deployed to challenge and authenticate the registration; this may be a Sonus ASX or other device. The registrar should be configured to require authentication on the registration; however the SBC does not check or enforce this.
Enterprise Access	Between an enterprise PBX and an SBC.	Server	Mutual TLS authentication for static (non-registering) IP PBX. Server-only Authentication for registering PBX.
Inter-Carrier Peering	Between a SIP proxy or Back-to-Back User Agent (B2B UA) belonging to another administrative domain and an SBC.	Client or Server	Mutual TLS authentication.
Intra-Carrier Peering	Between an SBC and a SIP proxy or a B2B UA belonging to the same administrative domain.	Client or Server	Mutual TLS authentication

Deployments may involve two or more of the above scenarios and include different transports (SIP over TLS, SIP over TCP, or SIP over UDP) simultaneously on separate legs of the same signaling path.

DTLS Crypto Suites

The Datagram Transport Layer Security (DTLS) protocol provides authentication, data integrity, and confidentiality for communications between two applications over an Unreliable Datagram Protocol (UDP). The Secure Real-time Transport Protocol (SRTP) provides encryption, message authentication and integrity, and replay protection to the RTP data in both unicast and multicast applications. DTLS-SRTP is an extension to the DTLS protocol, where DTLS acts as the key management protocol. DTLS protocol is also extended to negotiate the SRTP crypto suites and parameters for use with those keys.

WebRTC is a signaling protocol defined for real-time communication between Web browsers. WebRTC has assigned DTLS-SRTP protocol for the media exchange between the browsers. The SBC includes the following DTLS functionality:

- Real-time communication between the web browsers by using DTLS-SRTP while inter-working with SIP networks.
- DTLS on the media path for key management for the SRTP-based media.
- The self-signed certificates to secure and authenticate DTLS associations. DTLS connections are secured by the two browsers sharing self-signed certificates as part of the media connection during a DTLS handshake between the browsers. The certificates are authenticated by checking a fingerprint, which is passed in the signaling path as part of the Session Description Protocol (SDP).

The SBC includes DTLS crypto suites that define a set of ciphers (algorithms used for encrypting data) which allow the selection of an appropriate level of security. When a TLS connection is established, the client and server exchange information about which cipher suites they have in common.

The following crypto suites are supported:

Table 2: Supported DTLS Crypto Suites

Authentication Mechanism	Public/Private Key Pair	Confidentiality Cipher and Mode	Integrity Cipher
RSA-WITH-NULL-SHA The integrity cipher used for the TLS Record protocol.	RSA	NULL	SHA-1
RSA-WITH-AES-128-CBC-SHA (default) Confidentiality cipher and mode for the TLS Record protocol.	RSA	AES-128-CBC	SHA-1
RSA-WITH-AES-128-CBC-SHA-256 Confidentiality cipher and mode for the TLS Record protocol with SHA-256 as the hash function.	RSA	AES-128-CBC	SHA-256
RSA-WITH-AES-256-CBC-SHA Confidentiality cipher and mode for the TLS Record protocol with AES 256 encryption.	RSA	AES-256-CBC	SHA-1
RSA-WITH-AES-256-CBC-SHA-256* Confidentiality cipher and mode for the TLS Record protocol with AES 256 encryption and SHA-256 as the hash function.	RSA	AES-256-CBC	SHA-256
TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384** Confidentiality cipher and mode for TLS Recod with AES256 CBC and SHA384 as hash function.	ECDH-ECDSA	AES-256-CBC	SHA-384
TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384** Confidentiality cipher and mode for TLS Recod with AES256 GCM and SHA384 as hash function.	ECDH-ECDSA	AES-256-GCM	SHA-384
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA Confidentiality cipher and mode for the TLS Record protocol using ECDHE (Elliptic Curve Diffie-Hellman key Exchange) with AES128 CBC and SHA as hash function.	ECDHE-RSA	AES-128-CBC	SHA-1
TLS-ECDHE-RSA-WITH-AES-256-CBC-SHA-384* Confidentiality cipher and mode for the TLS Record protocol using ECDHE (Elliptic Curve Diffie-Hellman key Exchange) with AES256 CBC and SHA384 as hash function.	ECDHE-RSA	AES-256-CBC	SHA-384

* To use this cipher, TLS version 1.2 must be enabled in the TLS Profile.

** To use this cipher, TLS version 1.2 must be enabled in the TLS Profile and SSL certificates must be created using ECC keys.

Terms used in this table:

RSA – Authentication based on X.509 certificates using RSA public/private key pairs
AES-128 – Advanced Encryption Standard (128-bit key length)
CBC – Cipher Block Chaining
SHA – Secure Hash algorithm

Note
When FIPS-140-2 mode is enabled, do not use the `rsa-with-null-sha` option.

The SBC and its peer devices use X.509 digital certificates to authenticate themselves for TLS. Local certificates in PKSC # 12 format (attesting to the identity of the SBC) and remote Certificate Authority (CA) certificates may be installed on the SBC in a common area (/opt/sonus/external/) where they are available to TLS.