



Transport Layer Security Tunnel Support

- [Transport layer security tunnels, on page 1](#)
- [Configure a transport layer security tunnel, on page 2](#)
- [Transport layer security tunnel verification, on page 4](#)

Transport layer security tunnels

A transport layer security tunnel is a secure communication protocol that

- enables the controller to reach a public cloud automatically
- encrypts all data sent over the TCP connection, and
- ensures reliable and secure communication between the controller and Digital Network Architecture (DNA) on Cloud.

Cisco plug and play solution

The Cisco Catalyst 9800 Series Wireless Controller requires direct access to a public cloud to implement the teleworker solution using Cisco OfficeExtend Access Points (OEAPs). With the introduction of Transport Layer Security (TLS) tunnel support from Cisco IOS XE Amsterdam 17.3.2 onwards, the controller can now reach a public cloud automatically. This helps Cisco Catalyst Center on Cloud to establish TLS communication channels with the controller to monitor and manage wireless solutions.

The TLS connection ensures that the configuration and telemetry are reliably and securely communicated between the controller and the Digital Network Architecture (DNA) on Cloud. The TLS tunnel encrypts all the data that is sent over the TCP connection. The TLS tunnel provides a more secure protocol across the internet. After the controller discovery, the Cisco Catalyst Center on Cloud uses Cisco DNA Assurance and Automation features to manage the controller centrally.

The Cisco Plug and Play solution is a converged solution that provides a highly secure, scalable, seamless, and unified zero-touch deployment experience.

Plug-n-Play Agent

The Cisco Plug and Play (PnP) agent is an embedded software component that is present in all the Cisco network devices that support simplified deployment architecture. The PnP agent understands and interacts only with a PnP server. The PnP agent, using DHCP, DNS, or other such methods, tries to acquire the IP address of the PnP server with which it wants to communicate. After a server is found and a connection is established, the agent communicates with the PnP server to perform deployment-related activities.

For more information on Cisco Plug and Play, see the [Cisco Plug and Play Feature Guide](#).

Supported controllers

The Transport Layer Security Tunnel (TLS) over PnP feature is supported on the following controllers:

- Cisco Catalyst 9800-80 Wireless Controller
- Cisco Catalyst 9800-40 Wireless Controller
- Cisco Catalyst 9800-L Wireless Controller

Configure a transport layer security tunnel

Establish a secure TLS tunnel to protect data exchanged between network devices.

Use this procedure when you need to secure traffic between devices using a TLS tunnel, typically in environments requiring encrypted communication for compliance or security reasons.

Procedure

Step 1 Enter global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Configure a crypto TLS tunnel channel.

Example:

```
Device(config)# crypto tls-tunnel TLS-tunnel-name
```

Example:

```
Device(config)# crypto tls-tunnel cloud-primary
```

Step 3 Specify the server IPv4 address, IPv6 address, or URL name and the port number.

Example:

```
Device(config-crypto-tls-tunnel)# server ipv4 server-ipv4-address port port-number
```

Example:

```
Device(config-crypto-tls-tunnel)# server ipv4 172.31.255.255 port 4043
```

The server can be specified as an IPv4 address, IPv6 address, or URL. The port number must be in the range 1025-65535.

Step 4 Specify the overlay interface and interface number.

Example:

```
Device(config-crypto-tls-tunnel)# overlay interface interface-name
```

Example:

```
Device(config-crypto-tls-tunnel)# overlay interface Loopback0
```

An overlay interface is a logical, multiaccess, multicast-capable interface. It encapsulates Layer 2 frames in IP unicast or multicast headers.

Step 5 Specify the LAN interface type, number, and the priority rank.

Example:

```
Device(config-crypto-tls-tunnel)# local-interface interface-name priority priority-rank
```

Example:

```
Device(config-crypto-tls-tunnel)# local-interface vlan 1 priority 1
```

Currently, the tunnel supports only one WAN interface with priority 1 and does not support the list of WAN interfaces with multiple priorities.

Step 6 Specify a preshared key and password options.

Example:

```
Device(config-crypto-tls-tunnel)# psk id identity key psk-key
```

Example:

```
Device(config-crypto-tls-tunnel)# psk id test key
```

Step 7 Specify the trustpoints for use with the RSA signature authentication method.

Example:

```
Device(config-crypto-tls-tunnel)# pki trustpoint trustpoint-label sign-or-verify
```

Example:

```
Device(config-crypto-tls-tunnel)# pki trustpoint tsp1 sign
```

Use the certificate from the trustpoint for signing or verifying. If the **sign** or **verify** keyword is not specified, the trustpoint is used for both signing and verification. Authentication can be done using either pre-shared key (PSK) or PKI (certificate based).

Step 8 (Optional) Enable common criteria mode (FIPS mode).

Example:

```
Device(config-crypto-tls-tunnel)# cc-mode
```

Indicates a common criteria mode, which is a Federal Information Processing Standards (FIPS) mode.

Step 9 Enable the TLS tunnel.

Example:

```
Device(config-crypto-tls-tunnel)# no shutdown
```

Step 10 Return to privileged EXEC mode.

Example:

```
Device(config-crypto-tls-tunnel)# end
```

The TLS tunnel is now configured and enabled, providing secure encrypted communication between devices.

Transport layer security tunnel verification

This reference describes how to verify a transport layer security (TLS) tunnel by using various show commands to display client summary, session details, statistics, and session summary information.

The TLS client support includes BinOS processes using Linux Tun/Tap Interface. To verify the TLS client summary details, use the following command:

```
Device# show platform software tlsc client summary
TLS Client - Config Summary
```

Name	ID	Gateway	Port	Auth	Trustpoint	DPD Time	Rekey Time	Retry Time
fqdn	0		8443	PSK	N/A	60	300	20

To verify the TLS client session detail, session statistics, tunnel statistics, and DNS counters, use the following command:

```
Device# show platform software client detail <tls-name>
```

```
Session Name      : fqdn
FQDN resolved IP : 10.255.255.255
ID                : 0
Created           : 04/20/21 00:36:42
Updated          : 04/22/21 05:56:03
State             : Up (Rekey)
Up Time          : 04/21/21 20:30:21 (9 hours 25 minutes 45 seconds)
Down Time        : 04/21/21 20:30:01
Rekey Time       : 04/22/21 05:55:51 (15 seconds)
```

```
TLS Session Statistics
Up Notifications   : 3
Down Notifications : 2
Rekey Notifications : 636
DP State Updates  : 0
DPD Cleanups      : 0
```

Packets From	Packets To	Packet Errors To	Bytes From	Bytes To
BinOS	80	0	0	0
IOSd	0	0	0	0
TLS Client	0	0	0	0

```
TLS Tunnel Statistics
```

Type	Tx Packets	Rx Packets
Total	0	80
CSTP Ctrl	3836	3836
CSTP Data	80	0

Type	Requests	Responses
CSTP Cfg	639	639
CSTP DPD	3197	3197

```
Invalid CSTP Rx      : 0
Injected Packet Success : 0
```

```

Injected Packet Failed : 0
Consumed Packets      : 0
    
```

TLS Tunnel DNS Counters

```

DNS Resolve Request Success Count : 641
DNS Resolve Request Failure Count : 0
DNS Resolve Success Count         : 639
DNS Resolve Failure Count         : 2
    
```

To verify the TLS client global statistics, use the following command.

```

Device# show platform software tlsc statistics
TLS Client: Global Statistics
    
```

Session Statistics

```

Up / Down      : 5 / 2
Rekeys        : 636
DP Updates    : 0
DPD Cleanups  : 0
    
```

	Packets From	Packets To	Packet Errors To	Bytes From	Bytes To
BinOS	85	0			0
IOSd	0	0	0	0	0
TLS Client	0	0	0	0	0

Tunnel Statistics

```

SSL Handshake Init / Done : 641 / 641
TCP Connection Req / Done : 641 / 641
Tunnel Packets
Rx / Tx      : 85 / 0
Injected / Failed : 0 / 0
Consumed      : 0
    
```

CSTP Packets

```

Control Rx / Tx : 3839 / 3839
Data   Rx / Tx : 0 / 85
Config Req / Resp : 641 / 641
DPD   Req / Resp : 3198 / 3198
Invalid Rx      : 0
    
```

FQDN Counters

```

Req / Resp / Success : 0 / 0 / 0
    
```

NAT Counters

```

Transalte In / Out : 0 / 0
Ignore   In / Out : 0 / 0
Failed           : 0
Invalid          : 0
No Entry         : 0
Unsupported      : 0
    
```

Internal Counters

Type	Allocated	Freed
EV	1299	1295
Tunnel	5	4
Conn	643	642
Sess	3	2

Config Message Related Counters

```

Type      Success    Failed
    
```

```
-----  
Create      3          0  
Delete      2          0
```

To view the TLS client-session summary, use the following command.

```
Device# show platform software tlsc session summary
```

```
TLS Client - Session Summary
```

```
Name      ID      Created          State      Since          Elapsed  
-----  
fqdn      0      04/20/21 00:36:42 Up         04/21/21 20:30:21 9 hours 26 minutes 44  
seconds
```