MACSec Using EAP-TLS Authentication

This chapter describes how to achieve MACSec encryption between two Routers using the 802.1x Port-based authentication with Extensible Authentication Protocol-Transport Layer Security (EAP-TLS). EAP-TLS allows mutual authentication using certificates, between the authentication server and the client, and generates the Master Session Key (MSK). This MSK is used to derive the Connectivity Association Key (CAK), and the corresponding Connectivity Association Key Name (CKN) is derived from the EAP session ID.

Guidelines and Limitations for EAP-TLS Authentication

The EAP-TLS authentication has the following guidelines and limitations:

- The IOS-XR software supports 802.1X only on physical ports (Ethernet interfaces).
- The IOS-XR software supports only EAP-TLS authentication method.
- 802.1X Port-based authentication is used only to derive keys for MKA, and does not perform port control.
- The IOS-XR software supports both the PAE roles, as an authenticator and a supplicant.
- The IOS-XR software as an authenticator supports Remote EAP authentication using RADIUS as EAP transport.

- The IOS-XR software supports only single-host mode, and not multi-host mode.

IEEE 802.1X Device Roles

The devices in the network have the following specific roles with IEEE 802.1X authentication.

- Supplicant - An entity at one end of a point-to-point LAN segment that seeks to be authenticated by an Authenticator attached to the other end of that link.
- Authenticator - An entity that facilitates authentication of other entities attached to the same LAN.
- Authentication Server - An entity that provides an authentication service to an Authenticator. This service determines, from the credentials provided by the Supplicant, whether the Supplicant is authorized to access the services provided by the system in which the Authenticator resides.

Prerequisites for MACSec MKA Using EAP-TLS Authentication

- Ensure that a Certificate Authority (CA) server is configured for the network.
- Ensure that the configured CA certificate is valid.
- Ensure that the user has configured Cisco Identity Services Engine (ISE) Release 2.2 onwards or Cisco Secure Access Control Server Release 5.6 onwards as external AAA server.
- Ensure that the remote AAA server is configured with EAP-TLS method.
- Ensure that both the routers, the CA server, and the external AAA server are synchronized using Network Time Protocol (NTP). If time is not synchronized on all these devices, certificates may not be validated.

MACsec with Local EAP-TLS Authentication

In local EAP authentication, the EAP-server is co-located with the authenticator locally on the router. This feature enables the router to authenticate dot1x (802.1x) clients with the EAP-TLS method using TLS Version 1.0 (TLSv1). It provides EAP-TLS based mutual authentication, where a Master Session Key (MSK) is generated on successful authentication.

Configure MACSec Encryption Using EAP-TLS Authentication

Configuring MACSec encryption using EAP-TLS authentication involves the following tasks:

Configure RADIUS Server

To configure RADIUS server pre-shared keys, obtain the pre-shared key values for the remote RADIUS server and perform this task. You can also specify an IPv6 address for the host (radius server).

```
Router# configure terminal
Router(config)# radius-server host 209.165.200.225 key 7 094F471A1A0A57
Router(config)# radius-server vsa attribute ignore unknown
Router(config)# commit
```

Running Configuration

```
Router# show run radius-server
radius-server host 209.165.200.225 auth-port 1646
  key 7 094F471A1A0A57
  radius-server vsa attribute ignore unknown
!
```

Configure 802.1X Authentication Method

You can configure 802.1X authentication method using RADIUS as the protocol. Only default AAA method is supported for 802.1X authentication.

```
Router# configure terminal
Router(config)# aaa authentication dot1x default group radius
Router(config)# commit
```

Running Configuration

```
Router# show run aaa
configure
  aaa authentication dot1x default group radius
```

Generate RSA Key Pair

RSA key pairs are used to sign and encrypt key management messages. This is required before you can obtain a certificate for the node. You must enter the key modulus size when prompted.

```
Router# crypto key generate rsa 8002
Wed Aug  7 10:25:22.461 UTC
The name for the keys will be: 8002
Choose the size of the key modulus in the range of 512 to 4096 for your General Purpose
Keypair.
Choosing a key modulus greater than 512 may take a few minutes.

How many bits in the modulus [2048]: 600
Generating RSA keys ...
Done w/ crypto generate keypair
[OK]
```

To delete the RSA keys, use the `no` form: **no crypto key generate rsa**

The following is a sample output of **show crypto key mypubkey rsa** command.

```
Router# show crypto key mypubkey rsa

Key label: 8002
Type     : RSA General purpose
Size     : 600
Created  : 12:56:29 UTC Wed Aug 07 2019
Data     :
 3067300D 06092A86 4886F70D 01010105 00035600 3053024C 0096DB0F EE3B3233
```

```
 6E5FDA53 0FC504D1 9A056E29 BB703118 C6A8A254 1DC6504B 29CD4DA0 984735C8
 46CD39A1 C379B059 92870F76 693D4A66 D9953F69 450238D4 C57803AF 41160D4F
 C9451945 02030100 01
```

Running Configuration

Configure Trustpoint

Trustpoints let you manage and track CAs and certificates. A trustpoint includes the identity of the CA, CA-specific configuration parameters, and an association with one, enrolled identity certificate. After you have defined a trustpoint, you can reference it by name in commands requiring that you specify a CA.

```
Router# configure terminal
Router(config)# crypto ca trustpoint test2
Router(config-trustp)# enrollment url http://caurl.com
Router(config-trustp)# subject-name CN=8000Series,OU=BU,O=Govt,L=Newyork,ST=NY,C=US
Router(config-trustp)# rsakeypair 8002
Router(config-trustp)# crl optional
Router(config-trustp)# commit
```

You can also specify the enrollment URL as an IP address (*http://10.2.2.2*).

Running Configuration

The following is a sample output of **show run crypto ca trustpoint test2** command.

```
crypto ca trustpoint test2
 crl optional
 subject-name CN=8000Series,OU=BU,O=Govt,L=Newyork,ST=NY,C=US
 enrollment url http://caurl.com
 rsakeypair 8002
   !
```

Configure Domain Name

You can configure a domain name, which is required for certificate enrollment.

```
Router(config)# domain name ca.8000-series.cisco.com
Router(config)# commit
```

Running Configuration

The following is a sample output of **show run domain name** command.

```
Router# show run domain name
Thu Mar 29 16:10:42.533 IST
domain name ca.8000-series.cisco.com
```

Authenticate Certificate Authority and Request Certificates

Certificate enrollment involves the following two steps:

1. Obtain CA certificate for the given trust point, using the **crypto ca authenticate** *tp_name* command.
2. Enroll the device certificate with CA, using the **crypto ca enroll** *tp_name* command.

```
Router# crypto ca authenticate test2
Router# crypto ca enroll test2
```

Running Configuration

The following is a sample output of the **show crypto ca certificates** command.

```
RP/0/RSP0/CPU0:router# show crypto ca certificates
Trustpoint : test2
==================================================
CA certificate
Serial Number        : E0:18:F3:E4:53:17:3E:28
Subject              : subject-name CN=8002,OU=BU,O=Govt,L=Newyork,ST=NY,C=US
Issued By            : subject-name CN=8002,OU=BU,O=Govt,L=Newyork,ST=NY,C=US
Validity Start       : 08:17:32 UTC Fri Jun 24 2016
Validity End         : 08:17:32 UTC Mon Jun 22 2026
SHA1 Fingerprint     : 894ABBFAA3B08E5B7D9E470ECFBBC04576B569F2
Router certificate
Key usage            : General Purpose
Status               : Available
Serial Number        : 03:18
Subject              :
serialNumber=cf302761,unstructuredAddress=209.165.200.225,unstructuredName=8002,
C=US,ST=NY,L=Newyork,O=Govt,OU=BU,CN=8002
Issued By            : CN=8000Series,OU=BU,O=Govt,L=Newyork,ST=NY,C=US
Validity Start       : 13:04:52 UTC Fri Feb 23 2018
Validity End         : 13:04:52 UTC Sat Feb 23 2019
SHA1 Fingerprint     :33B50A59C76CCD87D3D0F0271CD5C81F4A1EE9E1
Associated Trustpoint: test2
```

Configure EAP Profile

You can configure multiple EAP profiles.

```
Router# configure terminal
Router(config)# eap profile 8002
Router(config-eap)# identity CE1
Router(config-eap)# method tls pki-trustpoint test2
Router(config-eap)# commit
```

Running Configuration

The following is sample output of **show run eap** command.

```
Router# show run eap profile 8002
eap profile 8002
method tls pki-trustpoint test2
!
identity CE1
!
```

Configure 802.1X Profile on the Device

```
Router# configure
Router(config)# dot1x profile 8k_prof
Router(config-dot1x-8k_prof)# pae both
Router(config-dot1x-8k_prof)# authenticator timer reauth-time 3600
Router(config-dot1x-8k_prof)# supplicant eap profile 8002
Router(config-dot1x-8k_prof)# exit
```

```
Router(config)# commit
Router(config)# end
```

Running Configuration

The following is a sample output of the **show run dot1x profile 8k_prof** command.

```
Router# show run dot1x profile 8k_prof

dot1x profile 8k_prof
pae both
authenticator
  timer reauth-time 3600
!
supplicant
  eap profile 8002
!
```

Configure MACSec EAP and 802.1X Profile on an Interface

You can attach one of the 802.1X profiles on an interface.

```
Router# configure
Router(config)# interface fourHundredGigE 0/0/0/0
Router(config-if)# dot1x profile 8k_prof
Router(config-if)# macsec eap policy macsec-1
Router(config-if)# commit
```

Running Configuration

The following is a sample output of the **show run interface** command.

```
Router# show run interface HundredGigE 0/0/0/0
interface HundredGigE 0/0/0/0
dot1x profile 8k_prof
macsec eap policy macsec-1
!
```

Verify MACSec EAP and 802.1X Configuration on Interface

The following is a sample output of **show dot1x interface** command.

```
Router# show dot1x interface HundredGigE 0/0/0/24 detail

Dot1x info for HundredGigE 0/0/0/24
-----------------------------------------------------------------
Interface short name : Hu0/0/0/24
Interface handle     : 0x800020
Interface MAC        : 0201.9ab0.85af
Ethertype            : 888E
PAE                  : Both
Dot1x Port Status    : AUTHORIZED
Dot1x Profile        : 8k_prof
Supplicant:
Config Dependency    : Resolved
Eap profile          : 8k
Client List:
Authenticator        : 0257.3fae.5cda
EAP Method           : EAP-TLS
Supp SM State        : Authenticated
```

```
Supp Bend SM State   : Idle
Last authen time     : 2018 Mar 01 13:31:03.380
Authenticator:
Config Dependency    : Resolved
ReAuth               : Enabled, 0 day(s), 01:00:00
Client List:
Supplicant           : 0257.3fae.5cda
Auth SM State        : Authenticated
Auth Bend SM State   : Idle
Last authen time     : 2018 Mar 01 13:33:17.852
Time to next reauth  : 0 day(s), 00:59:57
MKA Interface:
Dot1x Tie Break Role : Auth
EAP Based Macsec     : Enabled
MKA Start time       : 2018 Mar 01 13:33:17.852
MKA Stop time        : NA
MKA Response time    : 2018 Mar 01 13:33:18.357
```

The following is a sample output of **show macsec mka session interface** command.

```
Router# show macsec mka session interface HundredGigE 0/0/0/24

============================================================================
Interface Local-TxSCI # Peers Status Key-Server
============================================================================
Hu0/0/0/24  0201.9ab0.85af/0001 1 Secured YES
```

The following is a sample output of **show macsec mka session interface detail** command.

```
Router# show macsec mka session interface HundredGigE 0/0/0/24 detail

MKA Detailed Status for MKA Session
===================================
Status                                : SECURED - Secured MKA Session with MACsec

Local Tx-SCI                          : 0201.9ab0.85af/0001
Local Tx-SSCI                         : 2
Interface MAC Address                 : 0201.9ab0.85af
MKA Port Identifier                   : 1
Interface Name                        : Hu0/0/0/24
CAK Name (CKN)                        : A94399EE68B2A455F85527A4309485DA
CA Authentication Mode                : EAP
Keychain                              : NA (EAP mode)
Member Identifier (MI)                : 3222A4A7678A6BDA553FDB54
Message Number (MN)                   : 114
Authenticator                         : YES
Key Server                            : YES
MKA Cipher Suite                      : AES-128-CMAC
Configured MACSec Cipher Suite        : GCM-AES-XPN-256
Latest SAK Status                     : Rx & Tx
Latest SAK AN                         : 1
Latest SAK KI (KN)                    : 3222A4A7678A6BDA553FDB5400000001 (1)
Old SAK Status                        : No Rx, No Tx
Old SAK AN                            : 0
Old SAK KI (KN)                       : RETIRED (0)
SAK Transmit Wait Time                : 0s (Not waiting for any peers to respond)
SAK Retire Time                       : 0s (No Old SAK to retire)
Time to SAK Rekey                     : NA
MKA Policy Name                       : *DEFAULT POLICY*
Key Server Priority                   : 16
Delay Protection                      : FALSE
Replay Window Size                    : 64
Include ICV Indicator                 : FALSE
Confidentiality Offset                : 0
Algorithm Agility                     : 80C201
```

```
SAK Cipher Suite                        : 0080C20001000004 (GCM-AES-XPN-256)
MACsec Capability                       : 3 (MACsec Integrity, Confidentiality, & Offset)
MACsec Desired                          : YES

# of MACsec Capable Live Peers          : 1
# of MACsec Capable Live Peers Responded : 1

Live Peer List:
MI                       MN     Rx-SCI (Peer)      SSCI    KS-Priority
-----------------------------------------------------------------------------
86B47DE76B42D9D7AB6805F7  113   0257.3fae.5cda/0001   1        16

Potential Peer List:
MI      MN         Rx-SCI (Peer)             SSCI         KS-Priority
-----------------------------------------------------------------------------

Peers Status:
Last Tx MKPDU   : 2018 Mar 01 13:36:56.450
Peer Count      : 1
RxSCI           : 02573FAE5CDA0001
MI              : 86B47DE76B42D9D7AB6805F7
Peer CAK        : Match
Latest Rx MKPDU : 2018 Mar 01 13:36:56.450
```