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 Cisco ASR 9000 devices, 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 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.

Configuration Example

```
Router# configure terminal
Router(config)# radius-server host 209.165.200.225 auth-port 1646 key secret007
Router(config)# radius-server vsa attribute ignore unknown
Router(config)# commit
```

Running Configuration

```
Router# show run radius
radius-server host 209.165.200.225 auth-port 1646
  key 7 00171605165E1F565F76
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.

Configuration Example

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

Running Configuration

```
Router# show run aaa
  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.

```
RP/0/RSP0/CPU0:router#crypto key generate rsa asr9k
```

Running Configuration

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

```
RP/0/RSP0/CPU0:router# show crypto key key-name rsa
Key label: asr9k
Type : RSA General purpose
Size : 2048
Data :
30820122 300D0609 2A864886 F70D0101 01050003 82010F00 3082010A 02820101
00BAA4F5 19C1C41A 4A195B31 6722B853 5271EECA B884CC19 CE75FB23 19DC0346
2F90F9B2 CBCB9BA3 4E4DDD46 2C21F555 4C642E3A 98FE0A2F 587D79F5 1D5B898F
893CEC38 B7C8CB03 48D0AEA1 D554DF2B BA751489 3099A890 1A910D25 7DA78F99
E29526FE 6F84C147 4F872715 D3BDE515 FACB28E8 6375BB38 1F3AFDA8 853C6E57
8BDA1800 7DDADFE3 32ABAB4C 3D078342 36E79F05 CAFCE764 26274F41 25F7BC70
```

```
04ABEDFE 96A183EE 23A3D099 2D5741C5 F81747FB 1ED5F672 5449B7AE 8D2E9224
CF12E1CA 9E2373C4 41BF29FA A9DDD930 5A3A2FDE FD1DADE1 2548DEDB 05FC2176
7D5DB337 B1563CA3 A94DF081 5B294D1A A9B70A56 CA5CF7B2 A779F27A 3EE4F568
F1020301 0001
```

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.

```
RP/0/RSP0/CPU0:router# configure terminal
RP/0/RSP0/CPU0:router(config)# crypto ca trustpoint {tp_name}
RP/0/RSP0/CPU0:router(config-trustp)# enrollment url {ca-url}
RP/0/RSP0/CPU0:router(config-trustp)# subject-name {x.500-name}
RP/0/RSP0/CPU0:router(config-trustp)# rsakeypair {keypair-label}
RP/0/RSP0/CPU0:router(config-trustp)# crl optional
RP/0/RSP0/CPU0:router(config-trustp)# commit
```

Running Configuration

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

```
crypto ca trustpoint tp
  crl optional
  subject-name CN=asr9k,OU=BU,O=Govt,L=Newyork,ST=NY,C=US
  enrollment url http://20.30.40.50
  rsakeypair asr9k
   !
```

Configure Domain Name

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

```
RP/0/RSP0/CPU0:router# domain name ca.asr9k.cisco.com
```

Running Configuration

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

```
RP/0/1/CPU0:router# show run domain name
Thu Mar 29 16:10:42.533 IST
domain name asr9k.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.

```
RP/0/RSP0/CPU0:router# crypto ca authenticate {tp_name}
RP/0/RSP0/CPU0:router# crypto ca enroll {tp_name}
```

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 : tp
==================================================
CA certificate
Serial Number        : E0:18:F3:E4:53:17:3E:28
Subject              : subject-name CN=asr9k,OU=BU,O=Govt,L=Newyork,ST=NY,C=US
Issued By            : subject-name CN=asr9k,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=20.30.40.50,unstructuredName=asr9k,
C=US,ST=NY,L=Newyork,O=Govt,OU=BU,CN=asr9k
Issued By            : CN=asr9k,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: tp
```

Configure EAP Profile

You can configure multiple EAP profiles.

```
RP/0/RSP0/CPU0:router# configure terminal
RP/0/RSP0/CPU0:router(config)# eap profile {name}
RP/0/RSP0/CPU0:router(config-eap)# identity {user-name}
RP/0/RSP0/CPU0:router(config-eap)# method tls pki-trustpoint {trustpoint-name}
RP/0/RSP0/CPU0:router(config-eap)# commit
```

Running Configuration

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

```
RP/0/RSP0/CPU0:router# show run eap profile asr9k
eap profile asr9k
method tls pki-trustpoint tp
!
identity CE1
!
```

Configure MACsec using Local EAP-TLS Authentication

You can configure 802.1X profile on an authenticator.

```
RP/0/RP0/CPU0:router(config)# dot1x profile auth
RP/0/RP0/CPU0:router(config-dot1x-auth)# pae authenticator
RP/0/RP0/CPU0:router(config-dot1x-auth)# authenticator
RP/0/RP0/CPU0:router(config-dot1x-auth-auth)# eap profile eap_1
RP/0/RP0/CPU0:router(config-dot1x-auth-auth)# timer reauth-time 3600
RP/0/RP0/CPU0:router(config-dot1x-auth-auth)# commit
```

Running Configuration

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

RP/0/RSP0/CPU0:router# show run dot1x profile auth

```
dot1x profile auth
pae authenticator
authenticator
eap profile eap_1
timer reauth-time 3600
!
!
```

You can configure 802.1X profile on a supplicant.

```
RP/0/RP0/CPU0:router(config)# dot1x profile supp
RP/0/RP0/CPU0:router(config-dot1x-supp)# pae supplicant
RP/0/RP0/CPU0:router(config-dot1x-supp)# supplicant
RP/0/RP0/CPU0:router(config-dot1x-supp-supp)# eap profile eap_1
RP/0/RP0/CPU0:router(config-dot1x-supp-supp)# commit
```

Running Configuration

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

RP/0/RSP0/CPU0:router# show run dot1x profile supp

```
dot1x profile supp
pae supplicant
supplicant
eap profile eap_1
!
!
```

Configure MACsec using Remote EAP-TLS Authentication

You can configure 802.1X profile on an authenticator.

```
RP/0/RP0/CPU0:router(config)# dot1x profile auth
RP/0/RP0/CPU0:router(config-dot1x-auth)# pae authenticator
RP/0/RP0/CPU0:router(config-dot1x-auth)# authenticator
RP/0/RP0/CPU0:router(config-dot1x-auth-auth)# timer reauth-time 3600
RP/0/RP0/CPU0:router(config-dot1x-auth-auth)# commit
```

Running Configuration

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

RP/0/RSP0/CPU0:router# show run dot1x profile auth

```
dot1x profile auth
pae authenticator
authenticator
timer reauth-time 3600
!
!
```

You can configure 802.1X profile on a supplicant.

```
RP/0/RP0/CPU0:router(config)# dot1x profile supp
RP/0/RP0/CPU0:router(config-dot1x-supp)# pae supplicant
RP/0/RP0/CPU0:router(config-dot1x-supp)# supplicant
RP/0/RP0/CPU0:router(config-dot1x-supp-supp)# eap profile eap_1
RP/0/RP0/CPU0:router(config-dot1x-supp-supp)# commit
```

Running Configuration

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

RP/0/RSP0/CPU0:router# show run dot1x profile supp

```
dot1x profile supp
pae supplicant
supplicant
eap profile eap_1
!
!
```

Configure MACSec EAP and 802.1X Profile on Interface

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

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface interface-name
RP/0/RSP0/CPU0:router(config-if)# dot1x profile profile-name
RP/0/RSP0/CPU0:router(config-if)# macsec eap [policy macsec-policy-name]
RP/0/RSP0/CPU0:router(config-if)# commit
```

Running Configuration

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

```
RP/0/RSP0/CPU0:router# show run interface HundredGigE 0/1/1/2
interface HundredGigE 0/1/1/2
   dot1x profile asr9k_prof
   macsec eap
!
```

Verify MACSec EAP and 802.1X Configuration on Interface

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

```
RP/0/RSP0/CPU0:router# show dot1x interface HundredGigE 0/1/1/2 detail

Dot1x info for HundredGigE 0/1/1/2
---------------------------------------------------------------
Interface short name : Hu0/1/1/2
Interface handle     : 0x800020
Interface MAC        : 0201.9ab0.85af
Ethertype            : 888E
PAE                  : Both
Dot1x Port Status    : AUTHORIZED
Dot1x Profile        : asr9k_prof
Supplicant:
Config Dependency    : Resolved
Eap profile          : asr9k
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.

```
RP/0/RSP0/CPU0:router# show macsec mka session interface HundredGigE 0/1/1/2

=======================================================================
Interface Local-TxSCI # Peers Status Key-Server
=======================================================================
Hu0/1/12  0201.9ab0.85af/0001 1 Secured YES
```

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

```
RP/0/RSP0/CPU0:router# show macsec mka session interface HundredGigE 0/1/1/2 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                            : Hu0/1/1/2
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
```