Configure WAN MACsec on Catalyst 8500 with Subinterfaces

Contents

Introduction

This document describes the process for configuring WAN Media Access Control Security (MACsec) on Cisco Catalyst 8500 Platforms with subinterfaces.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Advanced networking concepts, including WAN, VLANs, and encryption
- Understanding of MACsec (IEEE 802.1AE) and Key Management (IEEE 802.1X-2010)
- Familiarity with Cisco IOS® XE Command Line Interface (CLI)

Components Used

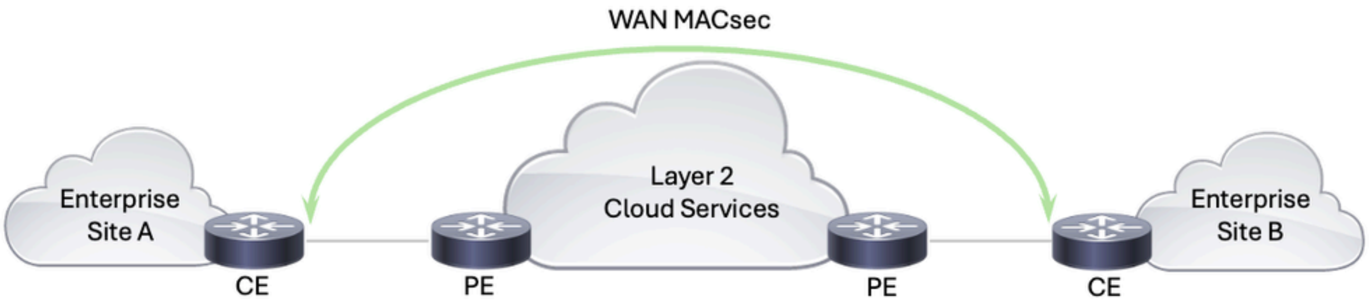
The information in this document is based on these software and hardware versions:

- Cisco Catalyst 8500 Series Edge Platforms
- Cisco IOS XE version 17.14.01a

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

WAN MACsec is a security solution designed to protect network traffic across WAN networks by utilizing the features of MACsec. When using a service provider network to exchange data, it is important to encrypt data in transit to prevent tampering. WAN MACsec is easy to deploy and manage, making it ideal for organizations that need to safeguard their network traffic from data manipulation, such as eavesdropping and man-in-the-middle attacks. It provides seamless, line-rate encryption, ensuring that data remains secure and uncompromised as it traverses various network infrastructures, including service provider networks, cloud environments, and enterprise networks.



WAN MACsec Solution

To share a bit of history, MACsec, defined by the IEEE 802.1AE standard, provides secure communication on Ethernet networks by ensuring data confidentiality, integrity, and origin authenticity for Ethernet frames. Operating at the data link layer (Layer 2) of the Open Systems Interconnection (OSI) model, MACsec encrypts and authenticates Ethernet frames to secure communication between nodes. Originally designed for LANs, MACsec has evolved to support WAN deployments as well. It offers line-rate encryption, ensuring minimal latency and overhead, which is crucial for high-speed networks.

IEEE 802.1X-2010 is an amendment to the original IEEE 802.1X standard, which defines Port-Based Network Access Control. The 2010 revision introduces the MACsec Key Agreement (MKA) protocol, which is essential for managing encryption keys in MACsec implementations. MKA handles the distribution and management of cryptographic keys used by MACsec to encrypt and decrypt data. MKA is a standard that contributes to multivendor interoperability for MACsec deployments, supporting secure key exchanges and rekeying mechanisms, critical for maintaining continuous security in dynamic WAN environments.

In WAN MACsec deployments, IEEE 802.1AE (MACsec) provides the fundamental encryption and security mechanisms at the data link layer, ensuring that all Ethernet frames are protected as they traverse the network. IEEE 802.1X-2010 with the MKA protocol, handles the critical task of distributing and managing the encryption keys necessary for MACsec to function. Together, these standards ensure that WAN MACsec can deliver robust, high-speed encryption across wide area networks, providing comprehensive protection for data in transit while maintaining interoperability and ease of management.

To address the unique challenges of WAN environments, some enhancements were made to the traditional MACsec deployments:

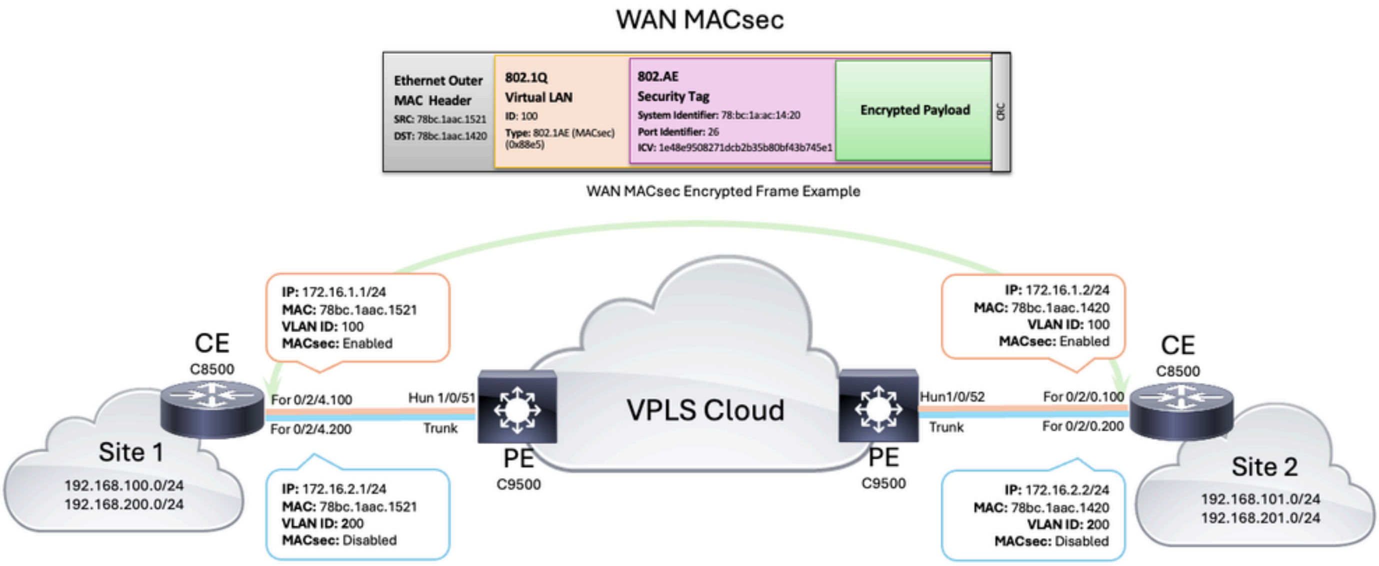
- 802.1Q Tag in the Clear: This feature allows the 802.1Q VLAN tag to be exposed outside the encrypted MACsec header, facilitating more flexible network designs, especially in public Ethernet transport environments. This capability is essential for integrating MACsec with Carrier Ethernet services, as it allows for the coexistence of encrypted and unencrypted traffic on the same network, simplifying network architecture and reducing costs.
- Adaptability Over Public Carrier Ethernet: Modern WAN MACsec implementations can adapt to

public carrier Ethernet services. This adaptability includes modifying the Ethernet Authentication Protocol over LAN (EAPoL) destination address and EtherType, allowing MACsec to function seamlessly over Carrier Ethernet networks that can otherwise consume or block these frames.

WAN MACsec represents a significant advancement in Ethernet encryption, addressing the growing need for high-speed, secure WAN connections. Its ability to provide line-rate encryption, support for flexible network designs, and adaptability to public carrier services make it a critical component of modern network security architectures. By leveraging WAN MACsec, organizations can achieve robust security for their high-speed WAN links while simplifying their network architectures and reducing operational complexity.

Configure

Network Diagram



WAN MACsec Topology

Configurations

Step 1: Basic Device Configuration

To start the configuration, you first need to define the subinterfaces that are going to be used for the traffic segmentation and the connection to the service provider. For this scenario two subinterfaces are defined for VLAN 100 associated to subnet 172.16.1.0/24 and VLAN 200 associated to subnet 172.16.2.0/24 (later on only one subinterface is going to be configured with MACsec).

CE 8500-1	CE 8500-2
<#root> interface FortyGigabitEthernet0/2/4.100 **encapsulation dot1Q 100** **ip address 172.16.1.1 255.255.255.0** ! interface FortyGigabitEthernet0/2/4.200 **encapsulation dot1Q 200** **ip address 172.16.2.1 255.255.255.0**	<#root> interface FortyGigabitEthernet0/2/0.100 **encapsulation dot1Q 100** **ip address 172.16.1.2 255.255.255.0** ! interface FortyGigabitEthernet0/2/0.200 **encapsulation dot1Q 200** **ip address 172.16.2.2 255.255.255.0**

Step 2: Configure MACsec Key Chain

Remember that the IEEE 802.1X-2010 standard specifies that the MACsec Encryption Keys can be derived from a Pre-Shared Key (PSK), by 802.1X Extensible Authentication Protocol (EAP) or chosen and distributed by an MKA key server. In this example, PSKs are used and manually configured through the MACsec key chain, and these are equal to the Connectivity Association Key (CAK), which is the primary key used to derive all other encryption keys used in MACsec.

CE 8500-1	
<#root>	<#root>
8500-1#	8500-2#
configure terminal	**configure terminal**
8500-1(config)#	8500-2(config)#
key chain keychain_vlan100 macsec	**key chain keychain_vlan100**
8500-1(config-keychain-macsec)#	8500-2(config-keychain-mac
key 01	**key 01**
8500-1(config-keychain-macsec-key)#	8500-2(config-keychain-mac
cryptographic-algorithm aes-256-cmac	**cryptographic-algorithm ae**
8500-1(config-keychain-macsec-key)#	8500-2(config-keychain-mac
key-string a5b2df4657bd8c02fcdaaf1212fe27ccc54626ad12d7c3b64c7a93e0113011e1	**key-string a5b2df4657bd8c0**
8500-1(config-keychain-macsec-key)#	8500-2(config-keychain-mac
lifetime 00:00:00 Jun 1 2024 duration 864000	**lifetime 00:00:00 Jun 1 20**
8500-1(config-keychain-macsec-key)#	8500-2(config-keychain-mac
key 02	**key 02**
8500-1(config-keychain-macsec-key)#	8500-2(config-keychain-mac
cryptographic-algorithm aes-256-cmac	**cryptographic-algorithm ae**
8500-1(config-keychain-macsec-key)#	8500-2(config-keychain-mac
key-string b5b2df4657bd8c02fcdaaf1212fe27ccc54626ad12d7c3b64c7a93e0113011e2	**key-string b5b2df4657bd8c0**
8500-1(config-keychain-macsec-key)#	8500-2(config-keychain-mac
lifetime 23:00:00 Jun 1 2024 infinite	**lifetime 23:00:00 Jun 1 20**
8500-1(config-keychain-macsec-key)#	8500-2(config-keychain-mac
exit	**exit**
8500-1(config-keychain-macsec)#	8500-2(config-keychain-mac
exit	**exit**

Note: While configuring the MACsec key chain, remember that the **key-string** must consist of hex digits only, the **aes-128-cmac** cryptographic algorithm requires a key of 32 hex digits and the **aes-256-cmac** cryptographic algorithm requires a key of 64 hex digits.

Note: Remember, when using multiple keys, an overlapping time period between them is needed to achieve a hitless key rollover after the specified key lifetime has expired.

Warning: It is important to ensure that the clocks of both routers are synchronized; therefore, the use of Network Time Protocol (NTP) is highly recommended. Failure to do so can prevent the establishment of MKA sessions or cause them to fail in the future.

Step 3: Configure MKA Policy

While the default MKA policy can be useful for initial setup and simple networks, configuring a custom MKA policy for WAN MACsec is generally recommended to meet specific security, compliance, and performance requirements. Custom policies offer greater flexibility and control, ensuring your network security is robust and customized to your needs.

When configuring your MKA policy there are different elements that can be selected, such as, Key Server Priority, Delay Protection for the MACsec Key Agreement Packet Data Unit (MKPDU), Cipher Suite, among others. In this platform and software versions the next ciphers can be used:

MACsec Cipher	Description
gcm-aes-128	Galois/Counter Mode (GCM) with Advanced Encryption Standard (AES) using a 128-bit key
gcm-aes-256	Galois/Counter Mode (GCM) with AES using a 256-bit key (higher encryption strength)

gcm-aes-xpn-128	Galois/Counter Mode (GCM) with AES using a 128-bit key, with Extended Packet Numbering (XPN)
gcm-aes-xpn-256	Galois/Counter Mode (GCM) with AES using a 256-bit key, with XPN (higher encryption strength)

Note: XPN enhances the GCM-AES cipher by supporting longer packet numbering, which improves security for very long-lived sessions or high-throughput environments. The use of high-speed links, for example 40 Gb/s or 100 Gb/s, can cause very short key rollover times because the Packet Number (PN) within the MACsec frame, typically based on the number of packets sent, could be exhausted quickly at these speeds. XPN extends the packet numbering sequence and eliminate the need for frequent Security Association Key (SAK) rekey that can occur in high capacity links.

In this example, the selected cipher for the MKA policy is **gcm-aes-xpn-256**, and other elements are going to have the default value:

CE 8500-1	**CE 8500-2**
<#root> 8500-1#	<#root> 8500-2#

configure terminal Enter configuration commands, one per line. End with CNTL/Z. 8500-1(config)# **mka policy subint100** 8500-1(config-mka-policy)# **macsec-cipher-suite gcm-aes-xpn-256** 8500-1(config-mka-policy)# **end**	**configure terminal** Enter configuration commands, one per lin 8500-2(config)# **mka policy subint100** 8500-2(config-mka-policy)# **macsec-cipher-suite gcm-aes-xpn-256** 8500-2(config-mka-policy)# **end**

Step 4: Configure MACsec at Interface and Subinterface level

In this scenario, even though the physical interface is not being configured with an IP address, some **macsec** commands need to be applied at this level for the solution to work. The MACsec policy and key chain are applied at the subinterface level (see the configuration example):

CE 8500-1	CE 8500-2
<#root> 8500-1# **configure terminal** 8500-1(config)# **interface FortyGigabitEthernet0/2/4** 8500-1(config-if)# **mtu 9216** 8500-1(config-if)# **cdp enable** 8500-1(config-if)# **macsec dot1q-in-clear 1** 8500-1(config-if)# **macsec access-control should-secure** 8500-1(config-if)# **exit** 8500-1(config)# **interface FortyGigabitEthernet0/2/4.100** 8500-1(config-if)# **eapol destination-address broadcast-address**	<#root> 8500-2# **configure terminal** 8500-2(config)# **interface FortyGigabitEthernet0/2/0** 8500-2(config-if)# **mtu 9216** 8500-2(config-if)# **cdp enable** 8500-2(config-if)# **macsec dot1q-in-clear 1** 8500-2(config-if)# **macsec access-control should-secure** 8500-2(config-if)# **exit** 8500-1(config)# **interface FortyGigabitEthernet0/2/0.100** 8500-2(config-if)# **eapol destination-address broadcast-address**

8500-1(config-if)#	8500-2(config-if)#
eapol eth-type 876F	**eapol eth-type 876F**
8500-1(config-if)#	8500-2(config-if)#
mka policy subint100	**mka policy subint100**
8500-1(config-if)#	8500-2(config-if)#
mka pre-shared-key key-chain keychain_vlan100	**mka pre-shared-key key-chain keychain_vlan100**
8500-1(config-if)#	8500-2(config-if)#
macsec	**macsec**
8500-2(config-if)#	8500-2(config-if)#
end	**end**

Commands Applied at Physical Interface Level

a. MTU is set to 9216 as the service provider used in the topology is allowing jumbo frames, but this is not a requirement

b. The command **macsec dot1q-in-clear** enables the option to have the VLAN (dot1q) tag in the clear (not encrypted)

c. The command **macsec access-control should-secure** allows unencrypted packets from the physical interface or subinterface to be sent or received (this command is needed if some subinterfaces require encryption and some others not, this is due to the default MACsec behavior where it does not allow any unencrypted packets to be transmitted or received from the same physical interface where MACsec is enabled)

Commands Applied at Subinterface Level

a. Now, the command **eapol destination-address broadcast-address** is needed to change the destination MAC address of the EAPoL frames (which by default is a multicast MAC address 01:80:C2:00:00:03) to a broadcast MAC address in order to make sure that the service provider floods them and do not drop or consume them.

b. The command **eapol eth-type 876F**, is used as well to change the default ethernet type of the EAPoL frame (which by default is 0x888E) and change it to 0x876F. This is again needed to prevent the service provider to drop or consume these frames.

c. The commands **mka policy <policy name>** and **mka pre-shared-key key-chain <key chain name>** are used to apply the custom policy and key chain to the subinterface.

d. And last but not least, the **macsec** command enables MACsec at the subinterface level.

In the current setup, without the previous EAPoL changes, the 9500 switches at the service provider side were not forwarding the EAPoL frames.

Note: MACsec commands such as **dot1q-in-clear** and **should-secure** are inherited by the subinterfaces. Additionally, EAPoL commands can be set at the physical interface level, and in such cases, these commands are also inherited by the subinterfaces. However, explicit configuration of EAPoL commands on the subinterface overrides the inherited value or policy for that subinterface.

Verify

Once the configuration is applied, the next output shows the relevant running configuration from each Customer Edge (CE) C8500 router (some configuration was omitted):

CE 8500-1	
<#root> 8500-1# **`show running-config`**	<#root> 8500-2# **`show running-config`**

```
Building configuration...

Current configuration : 8792 bytes
!
!
version 17.14
service timestamps debug datetime msec
service timestamps log datetime msec
service call-home
platform qfp utilization monitor load 80
!
hostname 8500-1
!
boot-start-marker
boot system flash bootflash:c8000aep-universalk9.17.14.01a.SPA.bin
boot-end-marker
!
!
no logging console
no aaa new-model
!
!

key chain keychain_vlan100 macsec
 key 01
  cryptographic-algorithm aes-256-cmac
  key-string a5b2df4657bd8c02fcdaaf1212fe27ccc54626ad12d7c3b64c7a93e0113011e1
  lifetime 00:00:00 Jun 1 2024 duration 864000
 key 02
  cryptographic-algorithm aes-256-cmac
  key-string b5b2df4657bd8c02fcdaaf1212fe27ccc54626ad12d7c3b64c7a93e0113011e2
  lifetime 23:00:00 Jun 1 2024 infinite

!
!
!
!
!
!
license boot level network-premier addon dna-premier
!
!
spanning-tree extend system-id
!

mka policy subint100
 macsec-cipher-suite gcm-aes-xpn-256


!
!
!
!
!
!
cdp run
!
!
!
!
interface Loopback100
 ip address 192.168.100.10 255.255.255.0
!
```

```
Building configuration...

Current configuration : 85
!
!
version 17.14
service timestamps debug d
service timestamps log dat
service call-home
platform qfp utilization m
!
hostname 8500-2
!
boot-start-marker
boot system flash bootflas
boot-end-marker
!
!
no logging console
no aaa new-model
!
!

key chain keychain_vlan100
 key 01
  cryptographic-algorithm
  key-string a5b2df4657bd8
  lifetime 00:00:00 Jun 1
 key 02
  cryptographic-algorithm
  key-string b5b2df4657bd8
  lifetime 23:00:00 Jun 1


!
!
!
!
!
!
license boot level network
!
!
spanning-tree extend syste
!

mka policy subint100
 macsec-cipher-suite gcm-a


!
!
!
!
!
!
cdp run
!
!
!
!
interface Loopback101
 ip address 192.168.101.10
```

```
interface Loopback200
 ip address 192.168.200.10 255.255.255.0
!
!

interface FortyGigabitEthernet0/2/4

 mtu 9216
 no ip address
 no negotiation auto
 cdp enable

 macsec dot1q-in-clear 1
 macsec access-control should-secure

!

interface FortyGigabitEthernet0/2/4.100

 encapsulation dot1Q 100
 ip address 172.16.1.1 255.255.255.0


ip mtu 9184

 eapol destination-address broadcast-address
 eapol eth-type 876F
 mka policy subint100
 mka pre-shared-key key-chain keychain_vlan100
 macsec

!

interface FortyGigabitEthernet0/2/4.200

 encapsulation dot1Q 200
 ip address 172.16.2.1 255.255.255.0
!
!
router eigrp 100
 network 172.16.1.0 0.0.0.255
 network 192.168.0.0 0.0.255.255
!
ip forward-protocol nd
!
!
!
control-plane
!
!
!
!
!
!
line con 0
 exec-timeout 0 0
 logging synchronous
 stopbits 1
line aux 0
line vty 0 4
 login
 transport input ssh
!
!
!
```

```
!
interface Loopback201
 ip address 192.168.201.10
!
!

interface FortyGigabitEthe

 mtu 9216
 no ip address
 no negotiation auto
 cdp enable

 macsec dot1q-in-clear 1
 macsec access-control sho

!

interface FortyGigabitEthe

 encapsulation dot1Q 100
 ip address 172.16.1.2 255


ip mtu 9184

 eapol destination-address
 eapol eth-type 876F
 mka policy subint100
 mka pre-shared-key key-ch
 macsec

!

interface FortyGigabitEthe

 encapsulation dot1Q 200
 ip address 172.16.2.2 255
!
!
router eigrp 100
 network 172.16.1.0 0.0.0.
 network 192.168.0.0 0.0.2
!
ip forward-protocol nd
!
!
!
control-plane
!
!
!
!
!
!
line con 0
 exec-timeout 0 0
 logging synchronous
 stopbits 1
line aux 0
line vty 0 4
 login
 transport input ssh
!
!
```

! ! ! end 8500-1#	! ! ! ! end 8500-2#

Note: Notice that after enabling MACsec, by applying the **macsec** command, the MTU at that interface is automatically adjusted and reduced by 32 bytes to account for the MACsec overhead.

Next, you can find a list of essential commands that can be utilized to check and verify the status of MACsec between peers. These commands provide you with detailed information about the current MACsec sessions, keychains, policies, and statistics:

show mka sessions - This command displays the current MKA sessions status.

show mka sessions detail - This command provides detailed information about each MKA session.

show mka keychains -This command shows the keychains used for MACsec and the assigned interface.

show mka policy - This command displays the policies applied, the interfaces and cipher suite used.

show mka summary - This command provides a summary of the MKA sessions and statistics.

show macsec statistics interface <interface name> - This command shows the MACsec statistics for a specified interface, and it helps to identify if encrypted traffic is being sent and received.

CE 8500-1

<#root>

```
8500-1#

show mka sessions


Total MKA Sessions....... 1
      Secured Sessions... 1
      Pending Sessions... 0

====================================================================================================
Interface       Local-TxSCI          Policy-Name       Inherited          Key-Server
Port-ID         Peer-RxSCI           MACsec-Peers      Status             CKN
====================================================================================================

Fo0/2/4.100

     78bc.1aac.1521/001a

subint100

         NO                   NO

26

              78bc.1aac.1420/001a  1

Secured

            02


8500-1#

show mka sessions detail


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

TX-SSCI.................. 2
Local Tx-SCI............. 78bc.1aac.1521/001a

Interface MAC Address.... 78bc.1aac.1521

MKA Port Identifier...... 26
Interface Name........... FortyGigabitEthernet0/2/4.100
Audit Session ID.........
CAK Name (CKN)........... 02
Member Identifier (MI)... 8387013B6C4D6106D4443285
Message Number (MN)...... 439243
```

```
EAP Role................. NA
Key Server............... NO

MKA Cipher Suite......... AES-256-CMAC


Latest SAK Status........ Rx & Tx
Latest SAK AN............ 0
Latest SAK KI (KN)....... F5720CC2E83183F1E673DACD00000001 (1)
Old SAK Status........... FIRST-SAK
Old SAK AN............... 0
Old SAK KI (KN).......... FIRST-SAK (0)

SAK Transmit Wait Time... 0s (Not waiting for any peers to respond)
SAK Retire Time.......... 0s (No Old SAK to retire)
SAK Rekey Time........... 0s (SAK Rekey interval not applicable)


MKA Policy Name.......... subint100

Key Server Priority...... 0
Delay Protection......... NO
Delay Protection Timer.......... 0s (Not enabled)

Confidentiality Offset... 0
Algorithm Agility........ 80C201
SAK Rekey On Live Peer Loss........ NO
Send Secure Announcement.. DISABLED
SCI Based SSCI Computation.... NO

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.. 0

Live Peers List:
  MI                        MN          Rx-SCI (Peer)        KS        RxSA          SSCI
                                                             Priority  Installed
  ---------------------------------------------------------------------------------------
  F5720CC2E83183F1E673DACD  439222      78bc.1aac.1420/001a  0         YES            1

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

Installed

  ---------------------------------------------------------------------------------------

8500-1#

show mka keychains


MKA PSK Keychain(s) Summary...

Keychain         Latest CKN                                                Interface(s)
Name             Latest CAK                                                Applied
===============================================================================================

keychain_vlan100 02                                                        Fo0/2/4.100
```

```
                 <HIDDEN>

8500-1#

show mka policy


MKA Policy defaults :
         Send-Secure-Announcements: DISABLED

MKA Policy Summary...

Codes : CO - Confidentiality Offset, ICVIND - Include ICV-Indicator,
        SAKR OLPL - SAK-Rekey On-Live-Peer-Loss,
        DP - Delay Protect, KS Prio - Key Server Priority

Policy            KS   DP    CO SAKR  ICVIND Cipher          Interfaces
Name              Prio          OLPL         Suite(s)        Applied
===============================================================================
*DEFAULT POLICY*  0    FALSE 0  FALSE TRUE   GCM-AES-128
                                             GCM-AES-256


subint100         0    FALSE 0  FALSE TRUE   GCM-AES-XPN-256 Fo0/2/4.100




8500-1#

show mka summary


Total MKA Sessions....... 1
      Secured Sessions... 1
      Pending Sessions... 0

====================================================================================================
Interface       Local-TxSCI          Policy-Name       Inherited          Key-Server
Port-ID         Peer-RxSCI           MACsec-Peers      Status             CKN
====================================================================================================
Fo0/2/4.100     78bc.1aac.1521/001a  subint100         NO                 NO
26              78bc.1aac.1420/001a  1                 Secured            02



MKA Global Statistics
=====================
MKA Session Totals
   Secured.................... 14
   Fallback Secured........... 0
   Reauthentication Attempts.. 0

   Deleted (Secured).......... 13
   Keepalive Timeouts......... 0

CA Statistics
   Pairwise CAKs Derived...... 0
   Pairwise CAK Rekeys........ 0
   Group CAKs Generated....... 0
   Group CAKs Received........ 0

SA Statistics
```

```
   SAKs Generated.............. 0
   SAKs Rekeyed................ 2
   SAKs Received............... 18
   SAK Responses Received...... 0
   SAK Rekeyed as KN Mismatch.. 0

MKPDU Statistics


MKPDUs Validated & Rx...... 737255

      "Distributed SAK"..... 18
      "Distributed CAK"..... 0


MKPDUs Transmitted......... 738485

      "Distributed SAK"..... 0
      "Distributed CAK"..... 0

MKA Error Counter Totals
========================
Session Failures
   Bring-up Failures................ 0
   Reauthentication Failures........ 0
   Duplicate Auth-Mgr Handle........ 0

SAK Failures
   SAK Generation................... 0
   Hash Key Generation.............. 0
   SAK Encryption/Wrap.............. 0
   SAK Decryption/Unwrap............ 0
   SAK Cipher Mismatch.............. 0

CA Failures
   Group CAK Generation............. 0
   Group CAK Encryption/Wrap........ 0
   Group CAK Decryption/Unwrap...... 0
   Pairwise CAK Derivation.......... 0
   CKN Derivation................... 0
   ICK Derivation................... 0
   KEK Derivation................... 0
   Invalid Peer MACsec Capability... 0

MACsec Failures
   Rx SC Creation................... 0
   Tx SC Creation................... 0
   Rx SA Installation............... 0
   Tx SA Installation............... 0

MKPDU Failures
   MKPDU Tx............................. 0
   MKPDU Rx ICV Verification............ 0
   MKPDU Rx Fallback ICV Verification..... 0
   MKPDU Rx Validation.................. 0
   MKPDU Rx Bad Peer MN................. 0
   MKPDU Rx Non-recent Peerlist MN........ 0
SAK USE Failures
   SAK USE Latest KN Mismatch............ 0
   SAK USE Latest AN not in USE........... 0

8500-1#
```

```
show macsec statistics interface Fo0/2/4.100

MACsec Statistics for FortyGigabitEthernet0/2/4.100
 SecY Counters
  Ingress Untag Pkts:       0
  Ingress No Tag Pkts:      0
  Ingress Bad Tag Pkts:     0
  Ingress Unknown SCI Pkts: 0
  Ingress No SCI Pkts:      0
  Ingress Overrun Pkts:     0
  Ingress Validated Octets: 0


Ingress Decrypted Octets: 11853398

  Egress Untag Pkts:        0
  Egress Too Long Pkts:     0
  Egress Protected Octets:  0


Egress Encrypted Octets:  11782598


 Controlled Port Counters
  IF In Octets:             14146226
  IF In Packets:            191065
  IF In Discard:            0
  IF In Errors:             0
  IF Out Octets:            14063174
  IF Out Packets:           190042
  IF Out Errors:            0

 Transmit SC Counters (SCI: 78BC1AAC1521001A)
  Out Pkts Protected:       0
  Out Pkts Encrypted:       190048
 Transmit SA Counters (AN 0)
  Out Pkts Protected:       0
  Out Pkts Encrypted:       190048

 Receive SA Counters (SCI: 78BC1AAC1420001A  AN 0)
  In Pkts Unchecked:        0
  In Pkts Delayed:          0
  In Pkts OK:               191069
  In Pkts Invalid:          0
  In Pkts Not Valid:        0
  In Pkts Not using SA:     0
  In Pkts Unused SA:        0
  In Pkts Late:             0
```

Reachability from the different subinterfaces is successful, as well as reachability between the 192.168.0.0/16 subnets. The next ping tests demonstrate the successful connectivity:

<#root>

8500-1#

```
ping 172.16.1.2


Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.2, timeout is 2 seconds:
```

```
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
8500-1#

ping 172.16.2.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.2.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
8500-1#

ping 192.168.101.10 source 192.168.100.10

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.101.10, timeout is 2 seconds:
Packet sent with a source address of 192.168.100.10
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
8500-1#
```

After capturing packets from an ICMP test on the Provider Edge (PE) device, you can compare the encrypted and unencrypted frames. Notice that the Ethernet outer MAC header is the same on both frames, with the dot1q tag visible. However, the encrypted frame shows an EtherType of 0x88E5 (MACsec), while the unencrypted frame displays an EtherType of 0x0800 (IPv4) along with the ICMP protocol information:

Encrypted Frame VLAN 100

```
<#root>

F241.03.03-9500-1#

show monitor capture cap buffer detail | begin Frame 80


Frame 80: 150 bytes on wire (1200 bits), 150 bytes captured (1200 bits) on interface /tmp/epc_ws/wif_to
    Interface id: 0 (/tmp/epc_ws/wif_to_ts_pipe)
        Interface name: /tmp/epc_ws/wif_to_ts_pipe
    Encapsulation type: Ethernet (1)
    Arrival Time: Jul 29, 2024 23:50:16.528191000 UTC
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1722297016.528191000 seconds
    [Time delta from previous captured frame: 0.224363000 seconds]
    [Time delta from previous displayed frame: 0.224363000 seconds]
    [Time since reference or first frame: 21.989269000 seconds]
    Frame Number: 80
    Frame Length: 150 bytes (1200 bits)
    Capture Length: 150 bytes (1200 bits)
    [Frame is marked: False]
    [Frame is ignored: False]


[Protocols in frame: eth:ethertype:vlan:ethertype:macsec:data]

Ethernet II, Src: 78:bc:1a:ac:15:21 (78:bc:1a:ac:15:21), Dst: 78:bc:1a:ac:14:20 (78:bc:1a:ac:14:20)

    Destination: 78:bc:1a:ac:14:20 (78:bc:1a:ac:14:20)
        Address: 78:bc:1a:ac:14:20 (78:bc:1a:ac:14:20)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Source: 78:bc:1a:ac:15:21 (78:bc:1a:ac:15:21)
```

```
        Address: 78:bc:1a:ac:15:21 (78:bc:1a:ac:15:21)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)

    Type: 802.1Q Virtual LAN (0x8100)
802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 100

    000. .... .... .... = Priority: Best Effort (default) (0)
    ...0 .... .... .... = DEI: Ineligible
    .... 0000 0110 0100 = ID: 100

    Type: 802.1AE (MACsec) (0x88e5)
802.1AE Security tag

    0010 11.. = TCI: 0x0b, VER: 0x0, SC, E, C
        0... .... = VER: 0x0
        .0.. .... = ES: Not set
        ..1. .... = SC: Set
        ...0 .... = SCB: Not set
        .... 1... = E: Set
        .... .1.. = C: Set
    .... ..00 = AN: 0x0
    Short length: 0

    Packet number: 147
    System Identifier: 78:bc:1a:ac:15:21 (78:bc:1a:ac:15:21)
    Port Identifier: 26
    ICV: 2b80cff435c7ef5a8d21b473b998cfb4

Data (102 bytes)

0000  99 53 71 3e f6 c7 9b bb 00 21 68 48 d6 ca 26 af   .Sq>.....!hH..&.
0010  80 a5 76 40 19 c9 45 97 b3 5a 48 d3 2d 30 72 a6   ..v@..E..ZH.-0r.
0020  96 47 6e a7 4c 30 90 e5 70 10 80 e8 68 00 5f ad   .Gn.L0..p...h._.
0030  7f dd 4a 70 a8 46 00 ef 7d 56 fe e2 66 ba 6c 1b   ..Jp.F..}V..f.l.
0040  3a 07 44 4e 5e e7 04 cb cb f4 03 71 8d 40 da 55   :.DN^......q.@.U
0050  9f 1b ef a6 3a 1e 42 c7 05 e6 9e d0 39 6e b7 3f   ....:.B.....9n.?
0060  f2 82 cf 66 f2 5b                                 ...f.[
    Data: 9953713ef6c79bbb00216848d6ca26af80a5764019c94597b^@&
    [Length: 102]
```

Related Information

- WAN MACSEC and MKA Support Enhancements

- Innovations in Ethernet Encryption (802.1AE - MACsec) for Securing High Speed (1-100GE) WAN Deployments
- Troubleshoot WAN MACSEC on Routers