Catalyst 9000에서 MACsec 문제 해결

목차

소개

이 문서에서는 MACsec 기능, 활용 사례, Catalyst 9000 스위치의 기능 트러블슈팅 방법에 대해 설명합니다.

사전 요구 사항

요구 사항

이 문서에 대한 특정 요건이 없습니다.

사용되는 구성 요소

- C9300
- C9400
- C9500
- C9600

참고: 다른 Cisco 플랫폼에서 이러한 기능을 활성화하는 데 사용되는 명령은 해당 설정 가이드를 참조하십시오.

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

배경 정보

이 문서의 범위는 두 스위치/라우터 간의 LAN상의 MACsec(Media Access Security Control)입니다.

일반 텍스트 데이터 통신은 보안 위협에 취약합니다. 보안 침해는 OSI 모델의 모든 레이어에서 발생할 수 있습니다. 레이어 2의 일반적인 보안 침해 중 일부는 스니핑, 패킷 도청, 변조, 주입, MAC 주소 스푸핑, ARP 스푸핑, DHCP 서버에 대한 DoS(Denial of Service) 공격, VLAN 호핑입니다.

MACsec은 IEEE 802.1AE 표준에 설명된 L2 암호화 기술입니다. MACsec은 물리적 미디어의 데이터를 보호하며, 상위 레이어에서 데이터가 손상되는 것을 방지합니다. 따라서 MACsec 암호화는 IPsec 및 SSL과 같은 상위 레이어의 다른 암호화 방법보다 우선 순위를 갖습니다.

MACsec의 장점

클라이언트 지향 모드: MACsec은 서로 피어링하는 두 스위치가 키를 교환하기 전에 키 서버 또는 키 클라이언트로 번갈아 사용될 수 있는 설정에 사용됩니다. 키 서버는 두 피어 간의 CAK를 생성하고 유지 관리합니다.

데이터 무결성 검사: MACsec은 MKA를 사용하여 포트에 도착하는 프레임에 대한 ICV(Integrity Check Value)를 생성합니다. 생성된 ICV가 프레임의 ICV와 동일하면 프레임이 수락되고 그렇지 않으면 삭제됩니다.

데이터 암호화: MACsec은 스위치 인터페이스에 포트 레벨 암호화를 제공합니다. 즉, 구성된 포트에서 전송된 프레임이 암호화되고 포트에서 수신된 프레임이 해독됩니다. 또한 MACsec은 암호화된 프레임만 구성하거나 모두 구성할 수 있는 메커니즘을 제공합니다

인터페이스에서 프레임(암호화 및 일반)을 허용합니다.

재생 보호: 네트워크를 통해 프레임을 전송하는 경우 프레임이 순서가 지정된 시퀀스에서 벗어날 가능성이 있습니다. MACsec은 지정된 수의 시퀀스 외 프레임을 허용하는 구성 가능한 창을 제공합니다.

MACsec 및 MTU

MACsec 헤더는 최대 32바이트의 헤더 오버헤드를 추가합니다. MACsec 헤더에서 추가된 추가 오버헤드를 감안하려면 경로의 스위치에서 더 큰 시스템/인터페이스 MTU(Maximum Transmission Unit)를 고려하십시오. MTU가 너무 낮으면 더 높은 MTU를 사용해야 하는 애플리케이션에서 예기치 않은 패킷 손실/지연을 볼 수 있습니다.

참고: MACsec과 관련된 문제가 있는 경우 호환성 매트릭스에 따라 양쪽 끝에 있는 GBIC(Gigabyte Interface Converter)가 지원되는지 확인하십시오.

MACsec 사용 위치

캠퍼스 활용 사례

- 호스트-스위치
- 사이트 또는 건물 간
- 다중 테넌시의 계층 간

데이터 센터 활용 사례

- 데이터 센터 상호 연결
- 서버-스위치

WAN 활용 사례

- 데이터 센터 상호 연결
- 캠퍼스 인터커넥트
- 허브 스포크

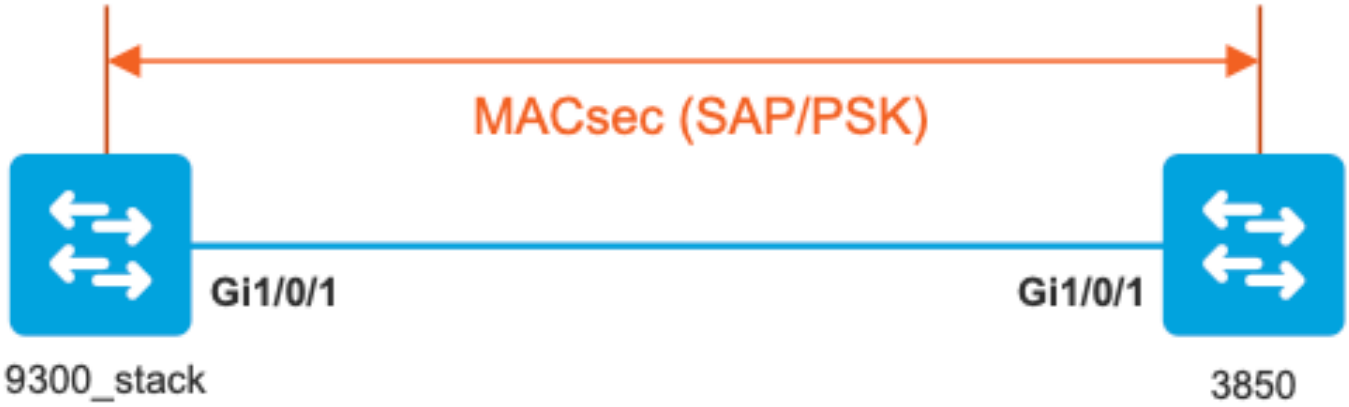
용어

MKA	MACsec 키 계약	MACsec 피어 검색 및 키 협상을 위한 키 계약 프로토콜로 IEEE 802.1X REV-2010에 정의됨
CAK	연결 연결 키	MACsec에 사용되는 다른 모든 키를 생성하는 데 사용되는 장기 기본 키. LAN 구현에서는 MSK에서 파생됩니다(EAP 교환 중에 생성됨).
PMK	쌍방향 기본 키	트래픽 암호화에 사용되는 세션 키를 파생시키는 데 사용되는 구성 요소 중 하나입니다. 수동으로 구성되었거나 802.1X에서 파생됨
CKN	CAK 키 이름	키 값 또는 CAK를 구성하는 데 사용됩니다. 최대 64자의 짝수 16진수 문자만 허용됩니다.
삭크	보안 연결 키	선택한 키 서버가 CAK에서 파생되며 라우터/엔드 디바이스가 지정된 세션의 트래픽을 암호화하는 데 사용하는 키입니다.
ICV	무결성 검사 값 키	CAK에서 파생되며 모든 데이터/제어 프레임에 태그가 지정되어 프레임이 인증된 피어에서 온 것임을 입증합니다. 암호 그룹에 따라

		8~16바이트
작은 나무통	키 암호화 키	CAK(사전 공유 키)에서 파생되며 MACsec 키를 보호하는 데 사용됩니다.
SCI	보안 채널 식별자	각 가상 포트는 16비트 포트 ID가 연결된 물리적 인터페이스의 MAC 주소를 기반으로 고유한 SCI(Secure Channel Identifier)를 수신합니다

시나리오 1: PSK(Pre-Shared Key) 모드에서 SAP를 사용하여 MACsec 스위치에서 스위치 링크 보안

토폴로지



1단계. 링크의 양쪽에서 컨피그레이션을 확인합니다.

<#root>

9300_stack#

show run interface gig 1/0/1

```
interface GigabitEthernet1/0/1
description MACsec_manual_3850-2-gi1/0/1
switchport access vlan 10
switchport mode trunk
```

cts manual

no propagate sgt

sap pmk

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

mode-list gcm-encrypt <-- use full packet encrypt mode

3850#

show run interface gig1/0/1

interface GigabitEthernet1/0/1
description 9300-1gi1/0/1 MACsec manual
switchport access vlan 10
switchport mode trunk

cts manual

no propagate sgt

sap pmk

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

mode-list gcm-encrypt

NOTE:

cts manual

<-- Supplies local configuration for Cisco TrustSec parameters

no propagate sgt

<-- disable SGT tagging on a manually-configured TrustSec-capable interface,

if you do not need to propage the SGT tags.

sap pmk AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA mode-list gcm-encrypt

<--

Use the sap command to manually specify the Pairwise Primary Key (PMK) and the Security Association Prot

authentication and encryption modes to negotiate MACsec link encryption between two interfaces.

The default encryption is sap modelist gcm-encrypt null

9300_stack#(config-if-cts-manual)#

sap pmk fa mode-list

?
gcm-encrypt GCM authentication, GCM encryption
gmac GCM authentication, no encryption
no-encap No encapsulation
null Encapsulation present, no authentication, no encryption

Use "gcm-encrypt" for full GCM-AES-128 encryption.

These protection levels are supported when you configure SAP pairwise primary key (sap pmk):

SAP is not configured— no protection.
sap mode-list gcm-encrypt gmac no-encap—protection desirable but not mandatory.
sap mode-list gcm-encrypt gmac—confidentiality preferred and integrity required.
The protection is selected by the supplicant according to supplicant preference.
sap mode-list gmac —integrity only.
sap mode-list gcm-encrypt-confidentiality required.
sap mode-list gmac gcm-encrypt-integrity required and preferred, confidentiality optional.

2단계. MACsec 상태를 확인하고 매개변수/카운터가 올바른지 확인합니다.

<#root>

Ping issued between endpoints to demonstrate counters

Host-1#

ping 10.10.10.12 <-- sourced from Host-1 IP 10.10.10.11

!!!!!!!!!!!!!!!!!!!!

9300_stack#

sh MACsec summary

Interface

Transmit SC Receive SC <-- Secure Channel (SC) flag is set for transmit and receive

GigabitEthernet1/0/1

1 1

9300_stack#

sh MACsec interface gigabitEthernet 1/0/1

MACsec is enabled

Replay protect : enabled
Replay window : 0
Include SCI : yes
Use ES Enable : no
Use SCB Enable : no
Admin Pt2Pt MAC : forceTrue(1)
Pt2Pt MAC Operational : no

Cipher : GCM-AES-128

Confidentiality Offset : 0
!

Capabilities

ICV length : 16
Data length change supported: yes
Max. Rx SA : 16
Max. Tx SA : 16
Max. Rx SC : 8
Max. Tx SC : 8
Validate Frames : strict
PN threshold notification support : Yes

Ciphers supported :

GCM-AES-128

GCM-AES-256

GCM-AES-XPN-128

GCM-AES-XPN-256

!

Transmit Secure Channels

SCI : 682C7B9A4D010000
SC state : notInUse(2)

Elapsed time : 03:17:50

Start time : 7w0d
Current AN: 0
Previous AN: 1
Next PN: 185

```
   SA State: notInUse(2)
   Confidentiality : yes
   SAK Unchanged : no

  SA Create time : 03:58:39

   SA Start time : 7w0d

   SC Statistics
   Auth-only Pkts : 0
   Auth-only Bytes : 0
   Encrypt Pkts : 2077

   Encrypt Bytes : 0
!
SA Statistics

   Auth-only Pkts : 0

Encrypt Pkts : 184

<-- packets are being encrypted and transmitted on this link

!
Port Statistics
   Egress untag pkts 0
   Egress long pkts 0
!
Receive Secure Channels

   SCI : D0C7897OC3810000
   SC state : notInUse(2)
   Elapsed time : 03:17:50
   Start time : 7w0d
   Current AN: 0
   Previous AN: 1
   Next PN: 2503
   RX SA Count: 0
   SA State: notInUse(2)
   SAK Unchanged : no

SA Create time : 03:58:39

   SA Start time : 7w0d

   SC Statistics
   Notvalid pkts 0
   Invalid pkts 0
   Valid pkts 28312
```

```
   Valid bytes 0
   Late pkts 0
   Uncheck pkts 0
   Delay pkts 0
   UnusedSA pkts 0
   NousingSA pkts 0
   Decrypt bytes 0
!
```

SA Statistics

```
      Notvalid pkts 0
      Invalid pkts 0
```

Valid pkts 2502

<-- number of valid packets received on this link

```
      UnusedSA pkts 0
      NousingSA pkts 0
!
Port Statistics
Ingress untag pkts 0
Ingress notag pkts 36
Ingress badtag pkts 0
Ingress unknownSCI pkts 0
Ingress noSCI pkts 0
Ingress overrun pkts 0
!

9300_stack#
```

sh cts interface summary

```
Global Dot1x feature is Disabled
CTS Layer2 Interfaces
---------------------
Interface  Mode   IFC-state  dot1x-role   peer-id  IFC-cache  Critical-Authentication
-------------------------------------------------------------------------------
Gi1/0/1
```

MANUAL   OPEN

```
      unknown    unknown    invalid     Invalid

CTS Layer3 Interfaces
---------------------
Interface IPv4 encap IPv6 encap IPv4 policy IPv6 policy
-------------------------------------------------------------------------
!

9300_stack#
```

sh cts interface gigabitEthernet 1/0/1

```
Global Dot1x feature is Disabled
Interface GigabitEthernet1/0/1:
```

CTS is enabled, mode: MANUAL

IFC state: OPEN

Interface Active for 04:10:15.723 <--- Uptime of MACsec port

Authentication Status: NOT APPLICABLE
Peer identity: "unknown"
Peer's advertised capabilities: "sap"
Authorization Status: NOT APPLICABLE
!

SAP Status: SUCCEEDED <-- SAP is successful

   Version: 2
   Configured pairwise ciphers:
   gcm-encrypt
!

Replay protection: enabled

Replay protection mode: STRICT

!
Selected cipher: gcm-encrypt
!
Propagate SGT: Disabled
Cache Info:
Expiration : N/A
Cache applied to link : NONE
!
Statistics:
   authc success: 0
   authc reject: 0
   authc failure: 0
   authc no response: 0
   authc logoff: 0

sap success: 1 <-- Negotiated once

sap fail: 0   <-- No failures

   authz success: 0

   authz fail: 0

   port auth fail: 0

   L3 IPM: disabled

3단계. 링크가 표시되면 소프트웨어 디버그를 검토합니다.

<#root>

Verify CTS and SAP events

```
debug cts sap events
debug cts sap packets
```

Troubleshoot MKA session bring up issues

```
debug mka event
debug mka errors
debug mka packets
```

Troubleshoot MKA keep-alive issues

```
debug mka linksec-interface
debug mka MACsec
debug MACsec
```

```
*May 8 00:48:04.843: %LINK-3-UPDOWN: Interface GigabitEthernet1/0/1, changed state to down
*May 8 00:48:05.324:  interface GigabitEthernet1/0/1 is UP
```

***May 8 00:48:05.324: CTS SAP ev (Gi1/0/1): Session started (new).**

***May 8 00:48:05.324: cts_sap_session_start CTS SAP ev (Gi1/0/1) peer:0000.0000.0000**
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

```
CTS SAP ev (Gi1/0/1): Old state: [waiting to restart],
event: [restart timer expired], action:
```

[send message #0] succeeded.

```
New state: [waiting to receive message #1].
```

***May 8 00:48:05.449: CTS SAP ev (Gi1/0/1): EAPOL-Key message from D0C7.8970.C381   <-- MAC of peer switc**

***May 8 00:48:05.449: CTS SAP ev (Gi1/0/1): EAPOL-Key message #0 parsed and validated.**

***May 8 00:48:05.449: CTS SAP ev (Gi1/0/1): Our MAC = 682C.7B9A.4D01   <-- MAC of local inte**

```
peer's MAC = D0C7.8970.C381.
CTS SAP ev (Gi1/0/1): Old state: [waiting to receive message #1],
```

event: [received message #0], action: [break tie] succeeded.

New state: [determining role].

***May 8 00:48:05.449: cts_sap_generate_pmkid_and_sci CTS SAP ev (Gi1/0/1) auth:682c.7b9a.4d01 supp:d0c7.8**
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

CTS SAP ev (Gi1/0/1): Old state: [determining role],

event: [change to authenticator], action: [send message #1] succeeded.

New state: [waiting to receive message #2].

***May 8 00:48:05.457: CTS SAP ev (Gi1/0/1): EAPOL-Key message from D0C7.8970.C381.**

CTS SAP ev (Gi1/0/1): New keys derived:
KCK = 700BEF1D 7A8E10F7 1243A168 883C74FB,
KEK = C207177C B6091790 F3C5B4B1 D51B75B8,
TK = 1B0E17CD 420D12AE 7DE06941 B679ED22,

***May 8 00:48:05.457: CTS SAP ev (Gi1/0/1): EAPOL-Key message #2 parsed and validated.**

***May 8 00:48:05.457: CTS-SAP ev: cts_sap_action_program_msg_2: (Gi1/0/1) GCM is allowed.**

*May 8 00:48:05.457: MACsec-IPC: sending clear_frames_option
*May 8 00:48:05.457: MACsec-IPC: geting switch number
*May 8 00:48:05.457: MACsec-IPC: switch number is 1
*May 8 00:48:05.457: MACsec-IPC: clear_frame send msg success
*May 8 00:48:05.457: MACsec-IPC: getting MACsec clear frames response
*May 8 00:48:05.457: MACsec-IPC: watched boolean waken up
*May 8 00:48:05.457: MACsec-CTS: create_sa invoked for SA creation
*May 8 00:48:05.457: MACsec-CTS: Set up TxSC and RxSC before we installTxSA and RxSA
*May 8 00:48:05.457: MACsec-CTS: create_tx_sc, avail=yes sci=682C7B9A
*May 8 00:48:05.457: NGWC-MACsec: create_tx_sc vlan invalid
*May 8 00:48:05.457: NGWC-MACsec: create_tx_sc client vlan=1, sci=0x682C7B9A4D010000
*May 8 00:48:05.457: MACsec-IPC: sending create_tx_sc
*May 8 00:48:05.457: MACsec-IPC: geting switch number
*May 8 00:48:05.457: MACsec-IPC: switch number is 1
*May 8 00:48:05.457: MACsec-IPC: create_tx_sc send msg success
*May 8 00:48:05.458: MACsec API blocking the invoking context
*May 8 00:48:05.458: MACsec-IPC: getting MACsec sa_sc response
*May 8 00:48:05.458: MACsec_blocking_callback
*May 8 00:48:05.458: Wake up the blocking process
*May 8 00:48:05.458: MACsec-CTS: create_rx_sc, avail=yes sci=D0C78970
*May 8 00:48:05.458: NGWC-MACsec: create_rx_sc client vlan=1, sci=0xD0C78970C3810000
*May 8 00:48:05.458: MACsec-IPC: sending create_rx_sc
*May 8 00:48:05.458: MACsec-IPC: geting switch number
*May 8 00:48:05.458: MACsec-IPC: switch number is 1
*May 8 00:48:05.458: MACsec-IPC: create_rx_sc send msg success
*May 8 00:48:05.458: MACsec API blocking the invoking context
*May 8 00:48:05.458: MACsec-IPC: getting MACsec sa_sc response
*May 8 00:48:05.458: MACsec_blocking_callback
*May 8 00:48:05.458: Wake up the blocking process
*May 8 00:48:05.458: MACsec-CTS: create_tx_rx_sa, txsci=682C7B9A, an=0

```
*May 8 00:48:05.458: MACsec-IPC: sending install_tx_sa
*May 8 00:48:05.458: MACsec-IPC: geting switch number
*May 8 00:48:05.458: MACsec-IPC: switch number is 1
*May 8 00:48:05.459: MACsec-IPC: install_tx_sa send msg success
*May 8 00:48:05.459: NGWC-MACsec:Sending authorized event to port SM
*May 8 00:48:05.459: MACsec API blocking the invoking context
*May 8 00:48:05.459: MACsec-IPC: getting MACsec sa_sc response
*May 8 00:48:05.459: MACsec_blocking_callback
*May 8 00:48:05.459: Wake up the blocking process
*May 8 00:48:05.459: MACsec-CTS: create_tx_rx_sa, rxsci=D0C78970, an=0
*May 8 00:48:05.459: MACsec-IPC: sending install_rx_sa
*May 8 00:48:05.459: MACsec-IPC: getting switch number
*May 8 00:48:05.459: MACsec-IPC: switch number is 1
*May 8 00:48:05.460: MACsec-IPC: install_rx_sa send msg success
*May 8 00:48:05.460: MACsec API blocking the invoking context
*May 8 00:48:05.460: MACsec-IPC: getting MACsec sa_sc response
*May 8 00:48:05.460: MACcsec_blocking_callback
*May 8 00:48:05.460: Wake up the blocking process
CTS SAP ev (Gi1/0/1): Old state: [waiting to receive message #2],
event: [received message #2], action: [program message #2] succeeded.
New state: [waiting to program message #2].
CTS SAP ev (Gi1/0/1): Old state: [waiting to program message #2],
event: [data path programmed], action: [send message #3] succeeded.
```

New state: [waiting to receive message #4].

***May 8 00:48:05.467: CTS SAP ev (Gi1/0/1): EAPOL-Key message from D0C7.8970.C381.**

***May 8 00:48:05.467: CTS SAP ev (Gi1/0/1): EAPOL-Key message #4 parsed and validated.**

***May 8 00:48:05.473: CTS-SAP ev: cts_sap_sync_sap_info: incr sync msg sent for Gi1/0/1**

***May 8 00:48:07.324: %LINK-3-UPDOWN: Interface GigabitEthernet1/0/1, changed state to up**

4단계. 링크가 나타날 때 플랫폼 레벨 추적을 검토합니다.

<#root>

9300_stack#

sh platform software fed switch 1 ifm mappings

Interface	IF_ID	Inst	Asic	Core	Port	SubPort	Mac	Cntx	LPN	GPN	Type	Active
GigabitEthernet1/0/1	0x8	1	0	1	0	0	26	6	1	1	NIF	Y

Note the IF_ID for respective intf

- This respective IF_ID shows in MACsec FED traces seen here.

9300_stack#

set platform software trace fed switch 1 cts_aci verbose

9300_stack#

set platform software trace fed switch 1 MACsec verbose

<-- switch number with MACsec port

9300_stack#

request platform software trace rotate all

/// shut/no shut the MACsec interface ///

9300_stack#

show platform software trace message fed switch 1

2019/05/08 01:08:50.688 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): FED sent MACsec_

2019/05/08 01:08:50.688 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): FED sending MACs

2019/05/08 01:08:50.688 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Running Instal
2019/05/08 01:08:50.688 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Processing job
2019/05/08 01:08:50.688 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Install RxSA c
2019/05/08 01:08:50.688 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Processing SPI
2019/05/08 01:08:50.688 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): MACSec install
2019/05/08 01:08:50.688 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): Entering ins_rx_

2019/05/08 01:08:50.688 {fed_F0-0}{1}: [l2tunnel_bcast] [16837]: UUID: 0, ra: 0, TID: 0 (ERR): port_idMA

2019/05/08 01:08:50.687 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): FED sent macsec_

2019/05/08 01:08:50.687 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): FED sending macs

2019/05/08 01:08:50.687 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): if_id = 8, cts_

2019/05/08 01:08:50.686 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Calling Install

2019/05/08 01:08:50.686 {fed_F0-0}{1}: [sec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): sci=0x682c7b9a4d01

2019/05/08 01:08:50.686 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Processing job

2019/05/08 01:08:50.686 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Create time of

2019/05/08 01:08:50.686 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): sci=0x682c7b9a4

2019/05/08 01:08:50.686 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Install TxSA ca

2019/05/08 01:08:50.686 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Processing SPI

2019/05/08 01:08:50.686 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): MACSec install T

2019/05/08 01:08:50.686 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): Entering ins_tx_

2019/05/08 01:08:50.686 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): FED sent macsec_

2019/05/08 01:08:50.686 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): FED sending macs

2019/05/08 01:08:50.686 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Conf_Offset in

2019/05/08 01:08:50.686 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Successfully in

2019/05/08 01:08:50.686 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Secy policy han

2019/05/08 01:08:50.686 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Install policy

2019/05/08 01:08:50.686 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Attach policy

2019/05/08 01:08:50.686 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Creating drop e

2019/05/08 01:08:50.686 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): if_id = 8, cts_

2019/05/08 01:08:50.686 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): sci=0x682c7b9a4

```
2019/05/08 01:08:50.686 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Create RxSC ca
2019/05/08 01:08:50.686 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Processing SPI
2019/05/08 01:08:50.686 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): MACSec create R
2019/05/08 01:08:50.686 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): Entering cre_rx
2019/05/08 01:08:50.685 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): FED sent macsec
2019/05/08 01:08:50.685 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): FED sending mac
2019/05/08 01:08:50.685 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): txSC setting x
```

2019/05/08 01:08:50.685 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Conf_Offset in

2019/05/08 01:08:50.685 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): if_id = 8, cts_

2019/05/08 01:08:50.685 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): secy created su

2019/05/08 01:08:50.685 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): if_id = 8, cts_

2019/05/08 01:08:50.685 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): if_id = 8, cts_

2019/05/08 01:08:50.685 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): is_remote is 0

2019/05/08 01:08:50.685 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Create TxSC cal

```
2019/05/08 01:08:50.685 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Processing SPI
2019/05/08 01:08:50.685 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): MACSec create T
2019/05/08 01:08:50.685 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): Entering cre_tx
2019/05/08 01:08:50.685 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): FED sent clear_
2019/05/08 01:08:50.685 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): FED sending mac
2019/05/08 01:08:50.685 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Processing job
2019/05/08 01:08:50.685 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (debug): Processing SPI
2019/05/08 01:08:50.685 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): MACSec clear_fr
2019/05/08 01:08:50.685 {fed_F0-0}{1}: [MACsec] [16837]: UUID: 0, ra: 0, TID: 0 (info): Entering clear_
2019/05/08 01:08:50.527 {fed_F0-0}{1}: [pm_xcvr] [17885]: UUID: 0, ra: 0, TID: 0 (note): XCVR POST:XCVR
speed_auto Oper Speed:speed_gbps1 Autoneg Mode:Unknown autonegmode type
2019/05/08 01:08:50.525 {fed_F0-0}{1}: [xcvr] [17885]: UUID: 0, ra: 0, TID: 0 (note): ntfy_lnk_status:
```

2019/05/08 01:08:48.142 {fed_F0-0}{1}: [pm_xcvr] [16837]: UUID: 0, ra: 0, TID: 0 (note): Enable XCVR for

2019/05/08 01:08:48.142 {fed_F0-0}{1}: [pm_tdl] [16837]: UUID: 0, ra: 0, TID: 0 (note): Received PM port

5단계. 하드웨어에서 MACsec 인터페이스의 상태를 확인합니다.

<#root>

9300_stack#

sh platform pm interface-numbers

```
interface iif-id gid slot unit slun   HWIDB-Ptr      status   status2   state snmp-if-index
---------------------------------------------------------------------------------------------------

Gi1/0/1     8     1   1    1    1    0x7F2C90D7C600  0x10040  0x20001B   0x4       8



9300_stack#
sh pl software fed switch 1 ifm if-id 8 <-- iif-id 8 maps to gig1/0/1


Interface IF_ID : 0x0000000000000008
Interface Name : GigabitEthernet1/0/1


Interface Block Pointer : 0x7f4a6c66b1b8
Interface Block State : READY
Interface State : Enabled


Interface Status : ADD, UPD
Interface Ref-Cnt : 8
Interface Type : ETHER
Port Type : SWITCH PORT
Port Location : LOCAL


Slot : 1
Unit : 0
Slot Unit : 1
SNMP IF Index : 8
GPN : 1
EC Channel : 0
EC Index : 0
Port Handle : 0x4e00004c


LISP v4 Mobility : false
LISP v6 Mobility : false
QoS Trust Type : 3
!
Port Information
Handle ............ [0x4e00004c]
Type .............. [Layer2]
Identifier ........ [0x8]
Slot .............. [1]
Unit .............. [1]

Port Physical Subblock
Affinity .......... [local]
Asic Instance ..... [1 (A:0,C:1)]
AsicPort .......... [0]
AsicSubPort ....... [0]
MacNum ............ [26]
ContextId ......... [6]
LPN ............... [1]
```

```
GPN ............... [1]
Speed ............. [1GB]
type .............. [NIF]

PORT_LE ........... [0x7f4a6c676bc8]



<--- port_LE


L3IF_LE ........... [0x0]
DI ................ [0x7f4a6c67d718]
SubIf count ....... [0]

Port L2 Subblock
Enabled ............. [Yes]
Allow dot1q ......... [Yes]
Allow native ........ [Yes]
Default VLAN ........ [1]
Allow priority tag ... [Yes]
Allow unknown unicast [Yes]
Allow unknown multicast[Yes]
Allow unknown broadcast[Yes]
Allow unknown multicast[Enabled]
Allow unknown unicast [Enabled]
Protected ............ [No]
IPv4 ARP snoop ....... [No]
IPv6 ARP snoop ....... [No]
Jumbo MTU ............ [1500]
Learning Mode ........ [1]
Vepa ................. [Disabled]

Port QoS Subblock
Trust Type ..................... [0x2]
Default Value .................. [0]
Ingress Table Map .............. [0x0]
Egress Table Map ............... [0x0]
Queue Map ...................... [0x0]
Port Netflow Subblock
Port Policy Subblock
List of Ingress Policies attached to an interface
List of Egress Policies attached to an interface


Port CTS Subblock


Disable SGACL ..................... [0x0]
Trust ............................. [0x0]
Propagate ......................... [0x0]
%Port SGT ......................... [-1717360783]


Physical Port Macsec Subblock <-- This block is not present when MACsec is not enabled



MACsec Enable .... [Yes]



MACsec port handle.... [0x4e00004c] <-- Same as PORT_LE
```

MACsec Virtual port handles....

.....[0x11000005]

MACsec Rx start index.... [0]
MACsec Rx end index.... [6]
MACsec Tx start index.... [0]
MACsec Tx end index.... [6]

Ref Count : 8 (feature Ref Counts + 1)
IFM Feature Ref Counts
FID : 102 (AAL_FEATURE_SRTP), Ref Count : 1
FID : 59 (AAL_FEATURE_NETFLOW_ACL), Ref Count : 1
FID : 95 (AAL_FEATURE_L2_MULTICAST_IGMP), Ref Count : 1
FID : 119 (AAL_FEATURE_PV_HASH), Ref Count : 1
FID : 17 (AAL_FEATURE_PBB), Ref Count : 1
FID : 83 (AAL_FEATURE_L2_MATM), Ref Count : 1
FID : 30 (AAL_FEATURE_URPF_ACL), Ref Count : 1
IFM Feature Sub block information
FID : 102 (AAL_FEATURE_SRTP), Private Data : 0x7f4a6c9a0838
FID : 59 (AAL_FEATURE_NETFLOW_ACL), Private Data : 0x7f4a6c9a00f8
FID : 17 (AAL_FEATURE_PBB), Private Data : 0x7f4a6c9986b8
FID : 30 (AAL_FEATURE_URPF_ACL), Private Data : 0x7f4a6c9981c8

9300_stack#

sh pl hard fed switch 1 fwd-asic abstraction print-resource-handle 0x7f4a6c676bc8 1 <-- port_LE handle

Handle:0x7f4a6c676bc8 Res-Type:ASIC_RSC_PORT_LE Res-Switch-Num:0 Asic-Num:1 Feature-ID:AL_FID_IFM Lkp-f
priv_ri/priv_si Handle: (nil)Hardware Indices/Handles: index1:0x0 mtu_index/l3u_ri_index1:0x2 sm handle
Detailed Resource Information (ASIC# 1)

snip
LEAD_PORT_ALLOW_CTS value 0 Pass
LEAD_PORT_ALLOW_NON_CTS value 0 Pass

LEAD_PORT_CTS_ENABLED value 1 Pass <-- Flag = 1 (CTS enabled)

LEAD_PORT_MACsec_ENCRYPTED value 1 Pass <-- Flag = 1 (MACsec encrypt enabled)

LEAD_PORT_PHY_MAC_SEC_SUB_PORT_ENABLED value 0 Pass
LEAD_PORT_SGT_ALLOWED value 0 Pass

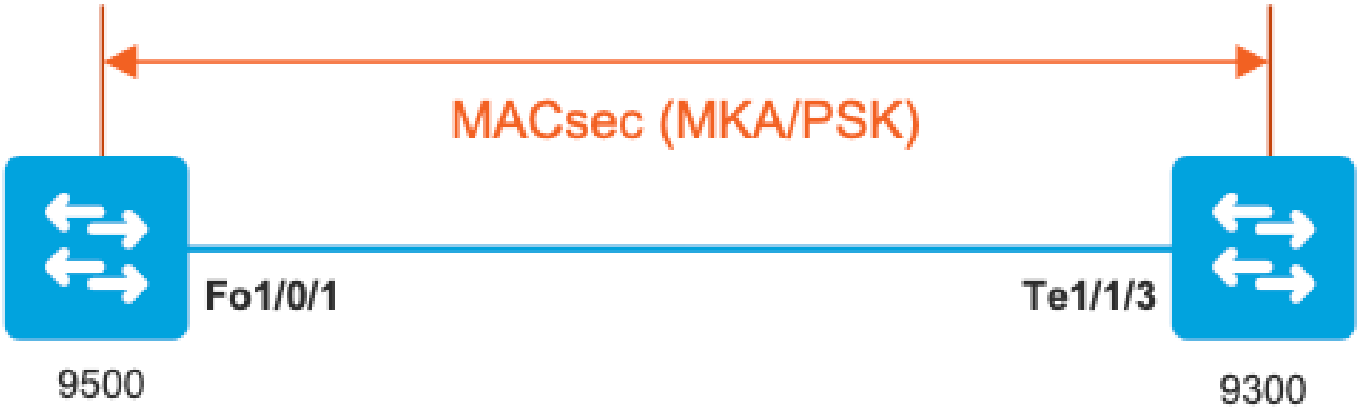
LEAD_PORT_EGRESS_MAC_sec_ENABLE_WITH_SCI value 1 Pass <-- Flag = 1 (MACsec with SCI enabled)

LEAD_PORT_EGRESS_MAC_sec_ENABLE_WITHOUT_SCI value 0 Pass
LEAD_PORT_EGRESS_MAC_sec_SUB_PORT value 0 Pass
LEAD_PORT_EGRESS_MACsec_ENCRYPTED value 0 Pass
snip

시나리오 2: PSK(Pre-Shared Key) 모드의 MKA를 통한 MACsec

스위치 간 링크 보안

토폴로지



1단계. 링크의 양쪽에서 컨피그레이션을 확인합니다.

<#root>

C9500#

sh run | sec key chain

key chain KEY MACsec
 key 01
   cryptographic-algorithm aes-256-cmac
  key-string 7 101C0B1A0343475954532E2E767B3233214105150555030A0004500B514B175F5B05515153005E0E5E505C52

lifetime local 00:00:00 Aug 21 2019 infinite <-- use NTP to sync the time for key chains

mka policy MKA

key-server priority 200
MACsec-cipher-suite gcm-aes-256
confidentiality-offset 0

C9500#

sh run interface fo1/0/1

interface fo1/0/1

MACsec network-link

mka policy MKA

mka pre-shared-key key-chain KEY

C9300#

sh run interface te1/1/3

interface te1/1/3

MACsec network-link

mka policy MKA

mka pre-shared-key key-chain KEY

2단계. MACsec이 활성화되었으며 모든 매개변수/카운터가 올바른지 확인합니다.

<#root>

This example shows the output from one side, verify on both ends of MACsec tunnel

C9500#

sh MACsec summary

Interface Transmit SC Receive SC

FortyGigabitEthernet1/0/1 1 1

C9500#

sh MACsec interface fortyGigabitEthernet 1/0/1

MACsec is enabled

Replay protect : enabled
Replay window : 0
Include SCI : yes
Use ES Enable : no
Use SCB Enable : no
Admin Pt2Pt MAC : forceTrue(1)
Pt2Pt MAC Operational : no

Cipher : GCM-AES-256

Confidentiality Offset : 0

Capabilities

ICV length : 16
Data length change supported: yes
Max. Rx SA : 16
Max. Tx SA : 16
Max. Rx SC : 8
Max. Tx SC : 8
Validate Frames : strict
PN threshold notification support : Yes

Ciphers supported : GCM-AES-128

GCM-AES-256

GCM-AES-XPN-128

GCM-AES-XPN-256

Transmit Secure Channels

SCI : 0CD0F8DCDC010008
SC state : notInUse(2)

Elapsed time : 00:24:38

Start time : 7w0d
Current AN: 0
Previous AN: -
Next PN: 2514
SA State: notInUse(2)
Confidentiality : yes
SAK Unchanged : yes

SA Create time : 1d01h

SA Start time : 7w0d

SC Statistics

Auth-only Pkts : 0
Auth-only Bytes : 0

Encrypt Pkts : 3156 <-- can increment with Tx traffic

Encrypt Bytes : 0

SA Statistics

Auth-only Pkts : 0

Encrypt Pkts : 402 <-- can increment with Tx traffic

Port Statistics

Egress untag pkts 0
Egress long pkts 0

Receive Secure Channels

SCI : A0F8490EA91F0026
SC state : notInUse(2)

Elapsed time : 00:24:38

Start time : 7w0d
Current AN: 0
Previous AN: -
Next PN: 94
RX SA Count: 0
SA State: notInUse(2)
SAK Unchanged : yes
SA Create time : 1d01h
SA Start time : 7w0d

SC Statistics

Notvalid pkts 0
Invalid pkts 0
Valid pkts 0
Valid bytes 0
Late pkts 0
Uncheck pkts 0
Delay pkts 0
UnusedSA pkts 0
NousingSA pkts 0
Decrypt bytes 0

SA Statistics

Notvalid pkts 0
Invalid pkts 0

Valid pkts 93

UnusedSA pkts 0
NousingSA pkts 0

!

Port Statistics

Ingress untag pkts 0

Ingress notag pkts 748

Ingress badtag pkts 0
Ingress unknownSCI pkts 0
Ingress noSCI pkts 0
Ingress overrun pkts 0

C9500#

sh mka sessions interface fortyGigabitEthernet 1/0/1

Summary of All Currently Active MKA Sessions on Interface FortyGigabitEthernet1/0/1...
==========
Interface Local-TxSCI

Policy-Name

Inherited Key-Server
Port-ID Peer-RxSCI MACsec-Peers Status CKN
==========

Fo1/0/1 0cd0.f8dc.dc01/0008

MKA

NO YES

8 a0f8.490e.a91f/0026 1 Secured01 <-- CKN number must match on both sides

0cd0.f8dc.dc01

<--

MAC of local interface

a0f8.490e.a91f

<--

MAC of remote neighbor

8

<-- indicates IIF_ID of respective local port (here IF_ID is 8 for local port fo1/0/1)

C9500#

sh platform pm interface-numbers | in iif|1/0/1

interface

iif-id

  gid slot unit slun   HWIDB-Ptr   status   status2 state snmp-if-index

Fo1/0/1

8

     1   1   1   1   0x7EFF3F442778 0x10040 0x20001B 0x4   8

C9500#

sh mka sessions interface fortyGigabitEthernet 1/0/1 detail

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

Local Tx-SCI..... 0cd0.f8dc.dc01/0008

Interface MAC Address.... 0cd0.f8dc.dc01

MKA Port Identifier..... 8

Interface Name..... FortyGigabitEthernet1/0/1

Audit Session ID.....
CAK Name (CKN)..... 01
Member Identifier (MI)... DFDC62E026E0712F0F096392

Message Number (MN)..... 536   <-- can increment as message numbers increment

EAP Role..... NA
Key Server..... YES

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

Latest SAK Status..... Rx & Tx
Latest SAK AN..... 0
Latest SAK KI (KN)..... DFDC62E026E0712F0F09639200000001 (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.......... MKA
Key Server Priority...... 200
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
SAK Cipher Suite......... 0080C20001000002 (GCM-AES-256)
MACsec Capability........ 3 (MACsec Integrity, Confidentiality, & Offset)
MACsec Desired........... YES
```

# of MACsec Capable Live Peers............ 1 <-- Peers capable of MACsec

# of MACsec Capable Live Peers Responded.. 1 <-- Peers that responded to MACsec negotiation

Live Peers List:

```
MI                        MN    Rx-SCI (Peer)        KS           RxSA
                                                    Priority     Installed
--------------------------------------------------------------------------------
```

ACF0BD8ECCA391A197F4DF6B 537 a0f8.490e.a91f/0026   200   YES <-- One live peer

!

Potential Peers List:

```
MI                        MN    Rx-SCI (Peer)        KS           RxSA
                                                    Priority     Installed
--------------------------------------------------------------------------------
```

Check the MKA policy and ensure that it is applied to expected interface

C9500#

sh mka policy MKA

```
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

==========

MKA

200 FALSE 0 FALSE TRUE

GCM-AES-256

Fo1/0/1 <-- Applied to Fo1/0/1

Ensure that PDU counters are incrementing at Tx/Rx at both sides.

This is useful to determine the direction of issues at transport. ###

C9500#

sh mka statistics | sec PDU

MKPDU Statistics

MKPDUs Validated & Rx..... 2342 <-- can increment

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

MKPDUs Transmitted..... 4552 <-- can increment

MKA Error Counters

C9500#

show mka statistics

** snip***

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 Validation.............. 0
MKPDU Rx Bad Peer MN............. 0
MKPDU Rx Non-recent Peerlist MN.. 0
```

3단계 - 5단계

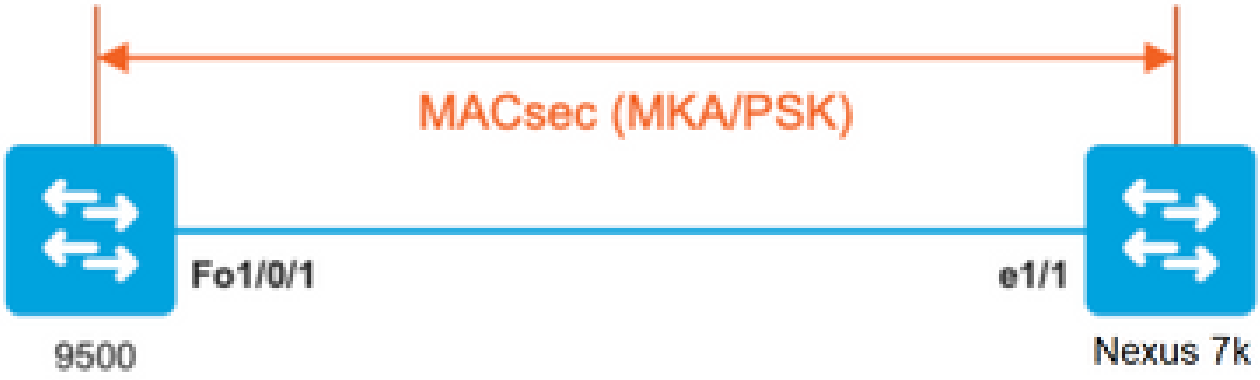
시나리오 1에서 설명한 것과 동일한 지침을 사용합니다.

경고: 상호운용성을 위해 일부 플랫폼은 패딩을 수행하고 일부 플랫폼은 패딩을 수행하지 않는다는 점에 유의하십시오. 이로 인해 mka 세션이 초기화 상태로 유지되는 주요 문제가 발생할 수 있습니다. show mka sessions 명령을 사용하여 이를 확인할 수 있습니다.

Padding 문제 예

이 활용 사례에서는 NX-OS 8.2(2)의 Catalyst 9500 및 Nexus 7k를 보여주지만 C3560CX와 같은 Catalyst 디바이스에서도 발생할 수 있습니다.

(Cisco 버그 ID CSCvs92023은 문제를 문서화합니다.)



- 시나리오 2에 제시된 컨피그레이션을 사용하는 경우 키 불일치로 인해 MKA가 터널을 설정할 수 없습니다.
- 이 디바이스에서는 패딩을 수행하지 않으므로 9500 측에서 0으로 키를 수동으로 완료해야 합니다.

Catalyst 9500

<#root>

```
conf t
  key chain MACsec1 MACsec
     key

01000000000000000000000000000000000000000000000000000000000000000000 --> device does not do padding automati

       key-string 12345678901234567890123456789012
       end
```

Nexus 7k

<#root>

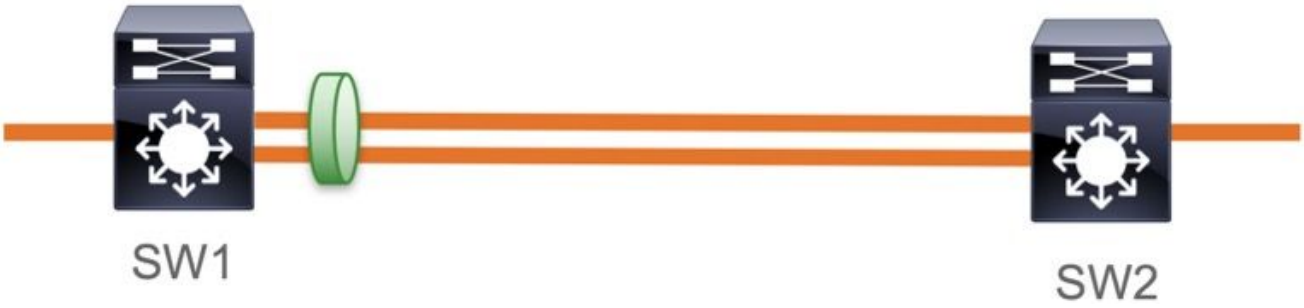
```
conf t
  key chain MACsec1 MACsec


key 01  --> Device does automatic padding.

      key-octet-string 12345678901234567890123456789012
      end
```

기타 구성 옵션

번들/포트 채널 인터페이스의 MKA를 통한 MACsec 스위치-스위치 링크 보안



- L3 및 L2 포트 채널(LACP, PAgP 및 Mode ON)
- 암호화 유형(AES-128 및 AES-256, AES-256은 Advantage 라이센스에 적용)
- 키 교환 MKA PSK만

지원되는 플랫폼:

- Catalyst 9200(AES-128만 해당)
- Catalyst 9300
- Catalyst 9400
- Catalyst 9500 및 Catalyst 9500H
- Catalyst 9600

스위치에서 스위치로의 Etherchannel 컨피그레이션 샘플

키 체인 및 MKA 정책 컨피그레이션은 앞서 MKA 컨피그레이션 섹션에서 설명한 것과 동일하게 유지됩니다.

<#root>

interface <> <-- This is the physical member link. MACsec encrypts on the individual links

MACsec network-link

```
mka policy <policy-name>
mka pre-shared-key key-chain <key-chain name>
macsec replay-protection window-size frame number
```

channel-group

```
mode active <-- Adding physical member to the port-channel
```

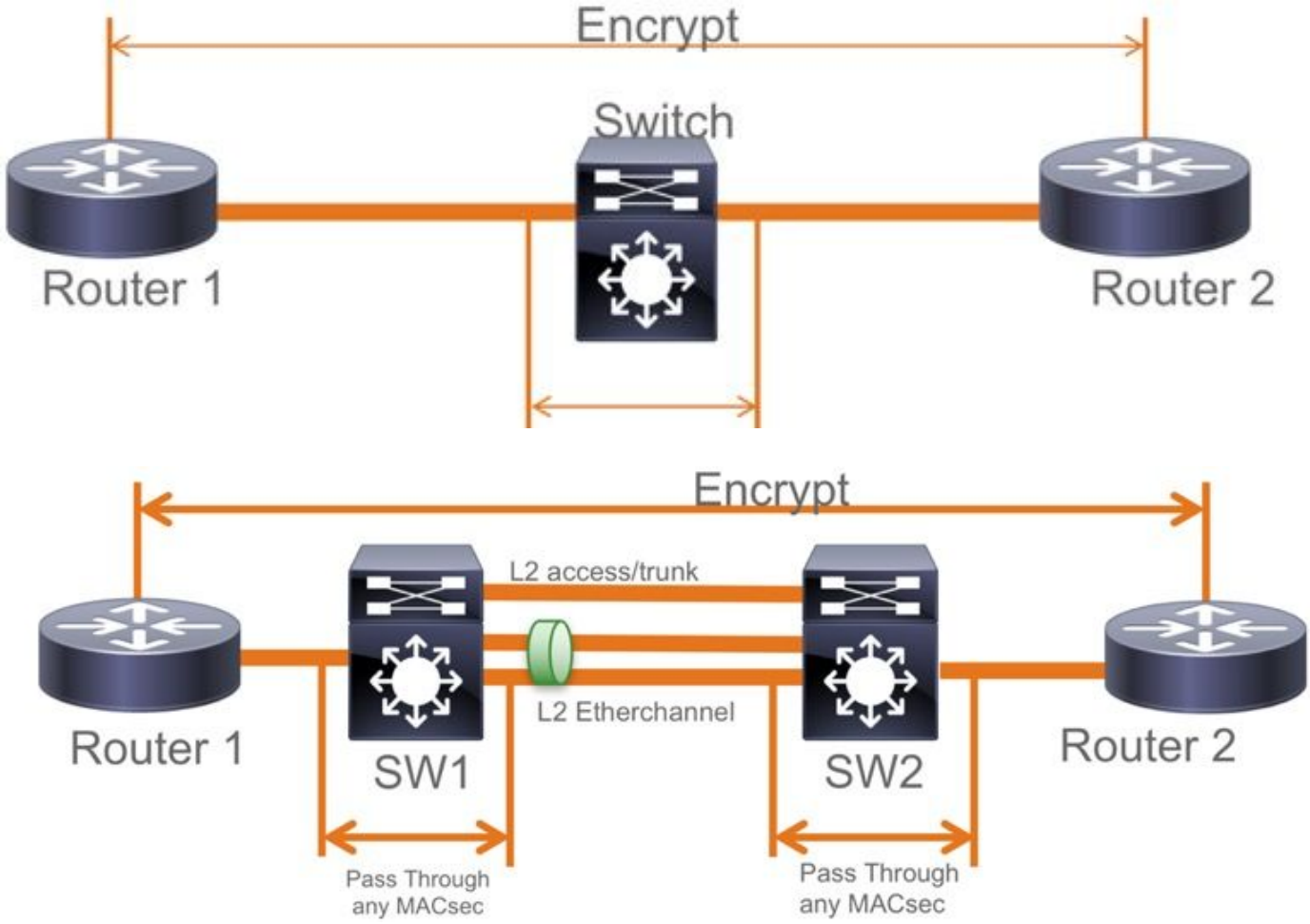
L2 중간 스위치, PSK 모드의 MACsec 스위치 간 링크 보안

이 섹션에서는 Cat9K가 암호화된 패킷을 투명하게 전달해야 하는 지원되는 WAN MACsec 시나리오 중 일부를 다룹니다.

라우터가 직접 연결되지 않았지만 L2 중간 스위치가 있는 경우가 있으며 L2 스위치는 암호화 처리 없이 암호화된 패킷을 우회할 수 있습니다.

Catalyst 9000 스위치는 16.10(1)부터 시작되는 Clear Tag를 사용하여 투명한 패킷을 전달합니다.

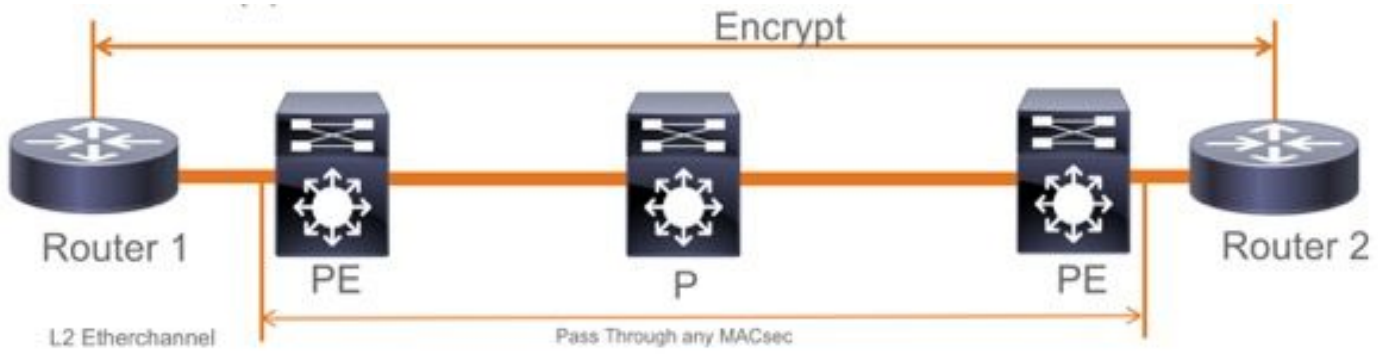
- MKA/SAP에 대해 통과가 지원됩니다.
- L2 액세스, 트렁크 또는 Etherchannel에서 지원
- 기본적으로 지원됨(활성화/비활성화할 컨피그레이션 CLI 없음)
- 라우터가 기본이 아닌(0x888E) 이더 타입의 EAPOL 프레임을 전송하는지 확인



EoMPLS/VPLS 토폴로지

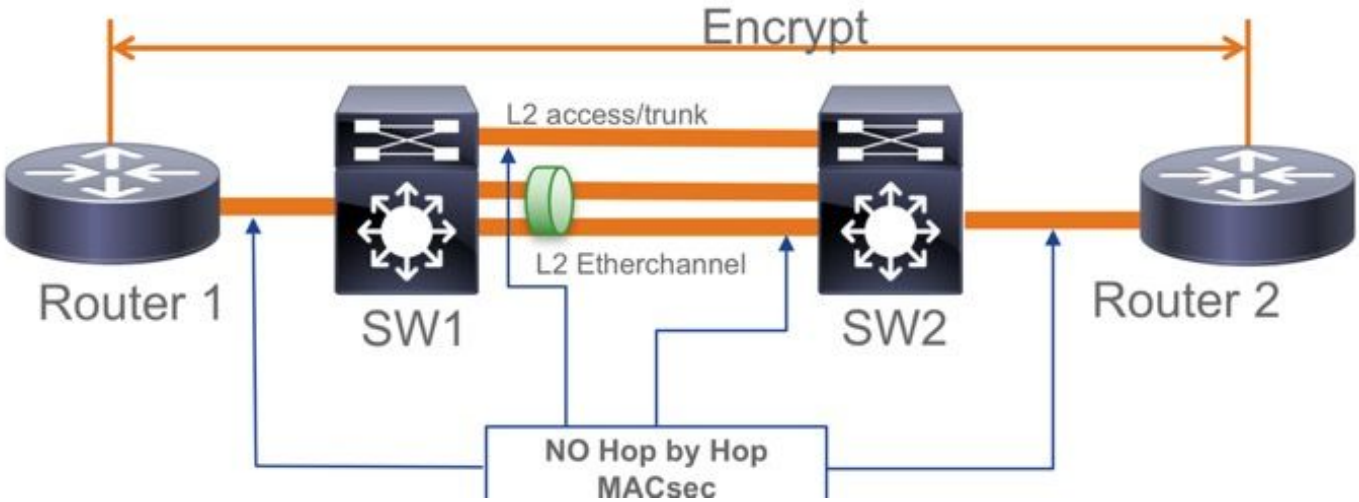
지원되는 플랫폼 Cat 9300/9400,9500/9500H(PE 또는 P 디바이스)

- VPLS
- EoMPLS
- 기본적으로 지원됨(활성화/비활성화할 컨피그레이션 CLI 없음)
- 시작 16.10(1)

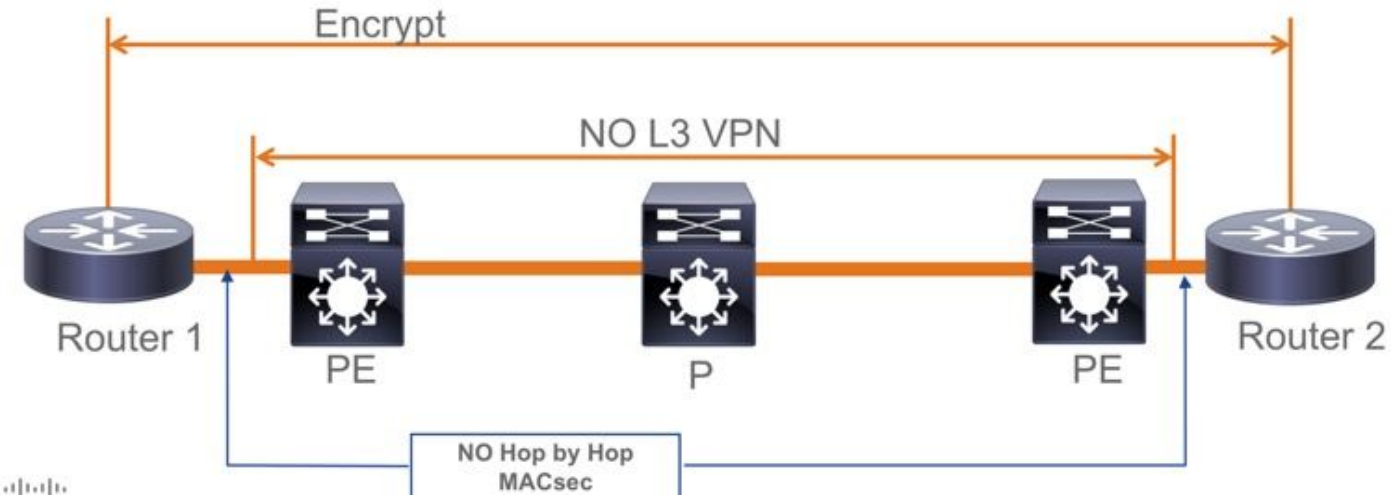


제약 조건

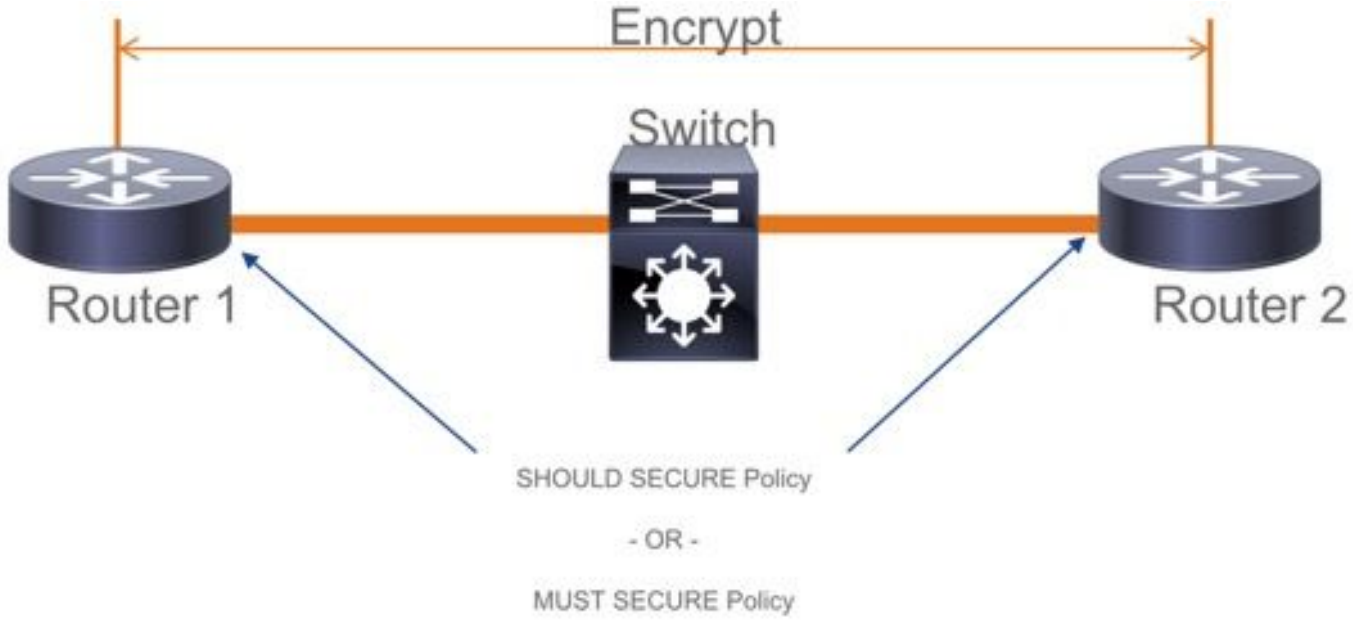
이중 암호화는 지원되지 않습니다. Clear 태그가 있는 엔드 투 엔드 MACsec에서는 L2 직접 연결된 링크에서 Hop by Hop 스위치가 활성화되지 않아야 합니다.



- ClearTag + EoMPLS(중간 레이어 2 전용 스위치 사용), MACsec은 CE-PE 링크에서 활성화할 수 없음
- ClearTag + L3VPN(중간 스위치 사용)은 지원되지 않음



- PSK 모드에서는 Should Secure가 지원되지 않습니다. Must Secure가 기본 모드입니다.
- Must Secure 정책은 MACsec 설정을 협상하기 위해 EAPoL만 암호화하지 않습니다.

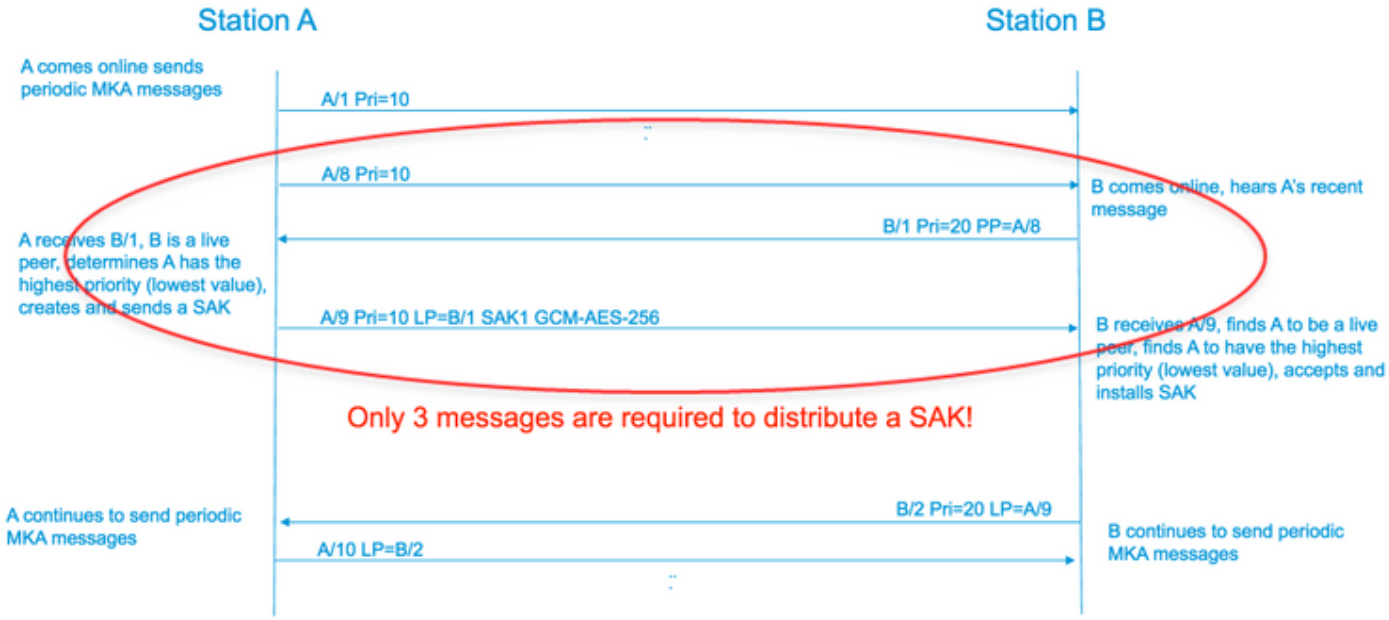


MACsec 운영 정보

작업 순서

1. 링크 및 양쪽 엔드 디바이스가 가동되면 MKA 프레임을 교환합니다(ethertype = 0x888E, 패킷 유형이 MKA인 EAPOL과 동일). 다중 지점 간 협상 프로토콜입니다. CAK 키 값(일반적으로 정적 사전 공유), 키 이름(CKN)이 일치해야 하며 피어를 검색하고 수락하려면 ICV가 유효해야 합니다.
2. Key Server 우선순위가 가장 낮은(기본값 = 0) 디바이스가 Key Server로 선택됩니다. 키 서버는 SAK를 생성하고 MKA 메시지를 통해 배포합니다. SCI(Secure Channel Identifier)가 가장 높은 값을 얻은 경우
3. 이후, 모든 MACsec 보안 프레임은 SAC(Symmetric Cyprintography)로 암호화됩니다. 별도의 TX 및 RX 보안 채널이 생성됩니다. 그러나 동일한 키 SAK가 암호화 및 암호 해독 모두에 사용됩니다.
4. EAPOL-MKA 메시지를 통해 다중 액세스 LAN에서 새 장치가 탐지되면 키 서버는 모든 장치

에서 사용할 새 키를 생성합니다. 새 키는 모든 장치에서 인식한 후에 사용됩니다(IEEE Std 802.1X-2010의 섹션 9.17.2 참조).



MACsec 패킷

제어 프레임(EAPOL-MKA)

- EAPOL 대상 MAC = 01:80:C2:00:00:03 - 여러 대상에 패킷을 멀티캐스트합니다.
- EAPOL 이더넷 유형 = 0x888E

컨트롤 프레임 형식의 L2 페이로드.

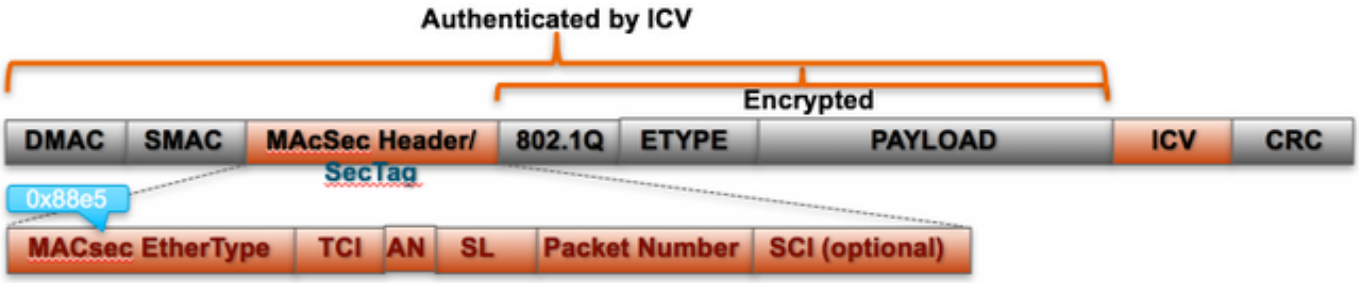
Protocol Version		
Packet Type = EAPOL-MKA		
Packet Body Length		Size
Packet Body (MKPDU)	Basic Parameter Set	Multiple of 4 octets
	Parameter Set	Multiple of 4 octets
	Parameter Set	Multiple of 4 octets
	ICV	16 octets

데이터 프레임

MACsec은 최대 오버헤드가 32바이트(최소 16바이트)인 데이터 프레임에 두 개의 추가 태그를 삽입합니다.

- SecTag = 8~16바이트(8바이트 SCI는 선택 사항)

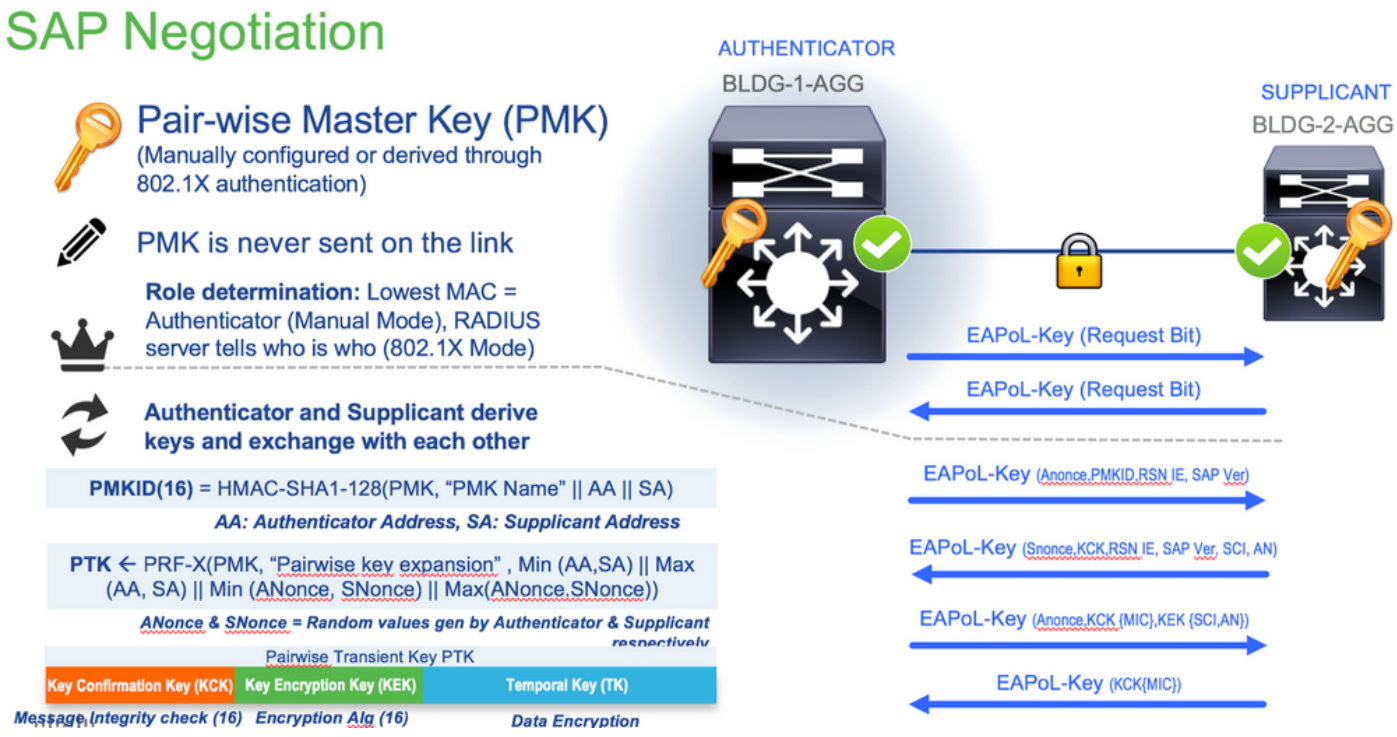
- ICV = 암호 수트를 기준으로 8~16바이트(AES128/256)



MACsec Tag Format

Field	Size	Description
Ethertype	16 bit	MAC length/type value for MACsec packet Ethertype = 88-E5
TCI	6 bit	Tag control info contains: Version, ES, SC, SCB, E, C (indicates how frame is protected)
AN	2 bit	Association number
SL	8 bit	Short Length Indicates MSDU length of 1-48 octets 0 indicates MSDU length > 48 octets
PN	32 bit	Packet sequence number
SCI	64 bit	Secure channel identified (optional)

SAP 협상



키 교환

MACsec Key Derivation Schemes

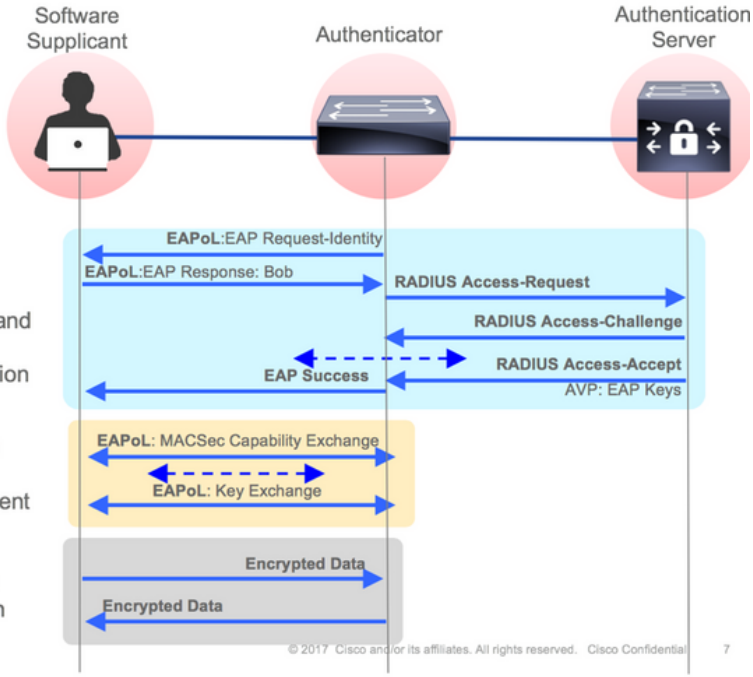
Session Key Agreement Protocols

SAP — **Security Association Protocol** is Cisco proprietary protocol for MACSec Key negotiation.

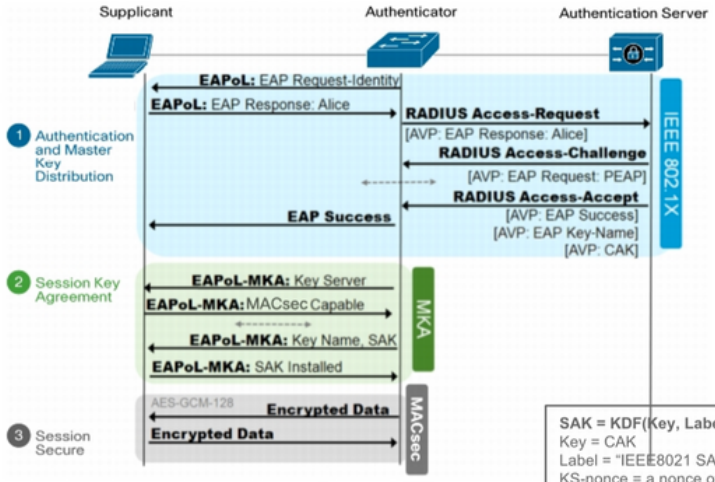
Used only for Switch-to-Switch encryptions.

MKA — MKA (**MACsec Key Agreement**) is defined in IEEE 802.1X-2010.

Used today for Switch-to-Host encryptions. Router MACsec uses MKA



MKA Exchange



ICV = AES-CMAC(ICK, M, 128)
M = DA + SA + (MSDU – ICV)

A pairwise CAK (Connectivity Association Key) is derived directly from the EAP MSK:
CAK = KDF(Key, Label, mac1 | mac2, CAKlength)

Key = MSK[0-15] for a 128 bit CAK, MSK[0-31] for a 256 bit CAK
Label = "IEEE8021 EAP CAK"
mac1 = the lesser of the two source MAC addr used in the EAPOL-EAP exchange
mac2 = the greater of the two source MAC addr used in the EAPOL-EAP exchange
CAKlength = two octets representing an integer value (128 for a 128 bit CAK, 256 for a 256 bit CAK) with the most significant octet first

The KEK(Key Encryption Key) is derived from the CAK using the following transform:
KEK = KDF(Key, Label, Keyid, KEKLength)

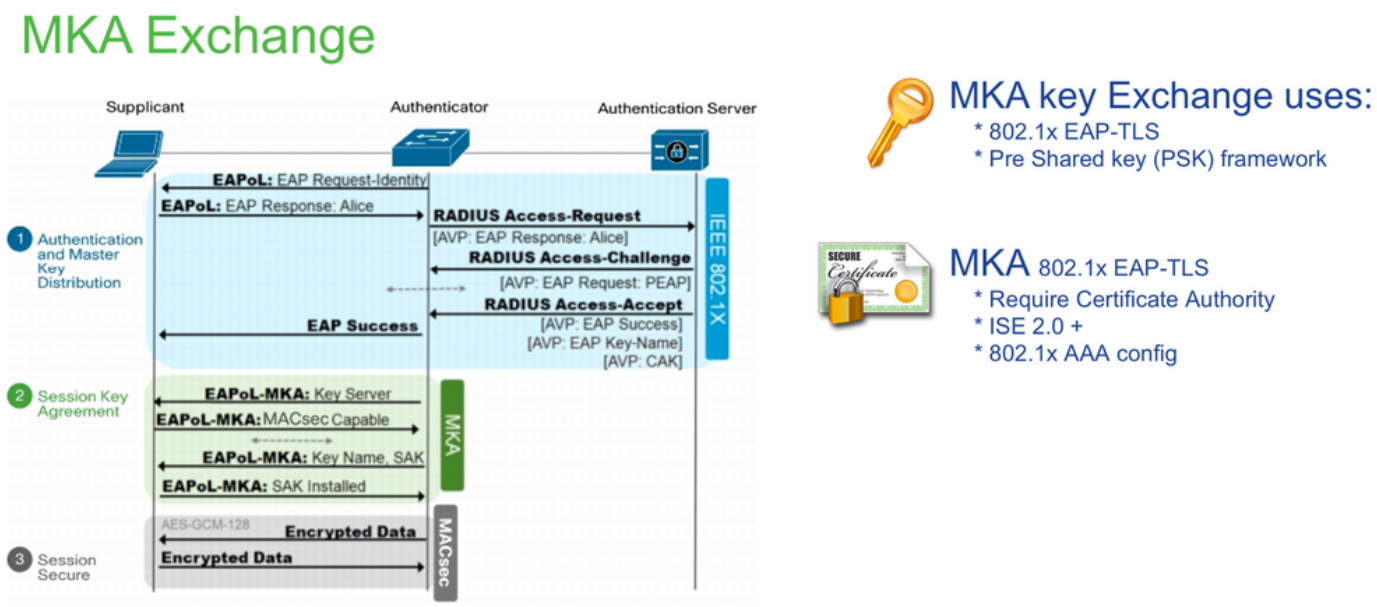
Key = CAK
Label = "IEEE8021 KEK"
Keyid = the first 16 octets of the CKN, with null octets appended to pad to 16 octets
KEKLength = two octets representing an integer value (128 for a 128 bit KEK, 256 for a 256 bit KEK) with the most significant octet first

The ICK (ICV Key) is derived from the CAK using the following transform:

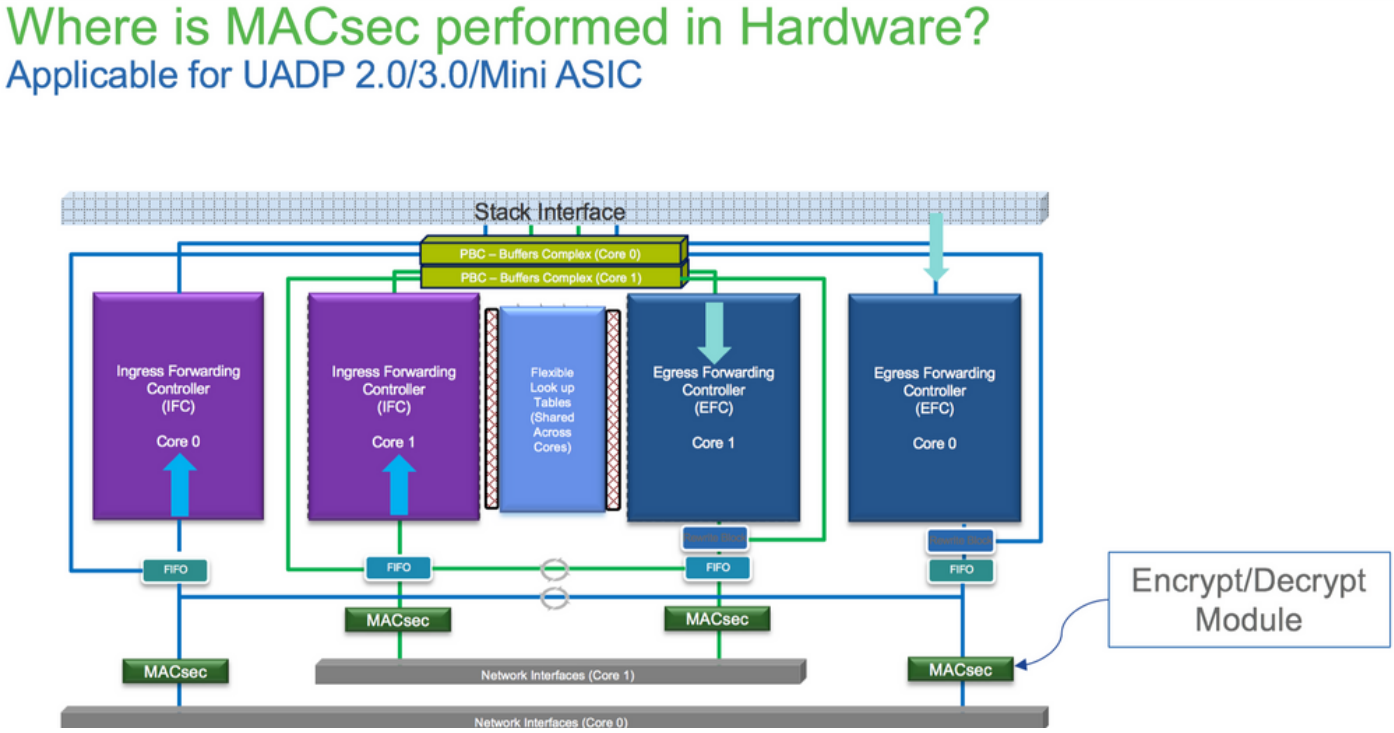
ICK = KDF(Key, Label, Keyid, ICKLength)

Key = CAK
Label = "IEEE8021 ICK"
Keyid = the first 16 octets of the CKN, with null octets appended to pad to 16
ICKLength = two octets representing an integer value (128 for a 128 bit ICK, 256 for a 256 bit ICK) with the most significant octet first

SAK = KDF(Key, Label, KS-nonce | MI-value list | KN, SAKlength)
Key = CAK
Label = "IEEE8021 SAK"
KS-nonce = a nonce of the same size as the required SAK, obtained from an RNG each time an SAK is generated.
MI-value list = a concatenation of MI values (in no particular order) from all live participants
KN = four octets, the Key Number assigned by the Key Server as part of the KI
SAKlength = two octets representing an integer value (128 for a 128 bit SAK, 256 for a 256 bit SAK) with the most significant octet first.



플랫폼의 MACsec



제품 호환성 매트릭스

LAN MACsec Support per Platform

	MACsec	Cat 9200		Cat 9300		Cat 9400		Cat 9500		Cat 9500H / 9600	
		SW	License	SW	License	SW	License	SW	License	SW	License
Switch to Switch	128 Bits SAP	16.10.1 +	NE	16.6.1 +	NE	16.10.1 +	NE	16.6.1 +	NE	16.9.1 + / 16.11.1 +	NE
	128 Bits MKA	16.10.1 +	NE	16.6.1 +	NE	16.10.1 +	NE	16.6.1 +	NE	16.9.1 + / 16.11.1 +	NE
	256 Bits MKA	Not Supported		16.6.1 +	NA	16.10.1 +	NA	16.6.1 +	NA	16.9.1 + / 16.11.1 +	NA
	ClearTag Pass Through	16.10.1 +	NE	16.10.1 +	NE	16.10.1 +	NE	16.10.1 +	NE	16.10.1 + / 16.11.1 +	NE
Host to Switch	128 Bits MKA	16.10.1 +	NE	16.8.1 +	NE	16.9.1 +	NE	16.8.1 +	NE	16.9.1 + / 16.11.1 +	NE
	256 Bits MKA	Not Supported		16.9.1 +	NA	16.10.1 +	NA	16.9.1 +	NA	16.9.1 + / 16.11.1 +	NA

NE – Network Essentials. NA – Network Advantage.
C9300 Stackwise 480 / C9500 SWV High Availability is not supported for MACsec
C9400 Sup 1XL-Y does not Support MACsec on any Supervisor ports
C9400 Sup 1 and 1XL support MACsec for only for interfaces with speed 10/40 Gbps

LAN MACsec Performance Data

	MACsec	Cat 9200	Cat 9300	Cat 9400	Cat 9500	Cat 9500H / 9600
Switch to Switch	128 Bits SAP	Line Rate	Line Rate	Line Rate	Line Rate	Line Rate
	128 Bits MKA	Line Rate	Line Rate	Line Rate	Line Rate	Line Rate
	256 Bits MKA	Not Supported	Line Rate	Line Rate	Line Rate	Line Rate
Host to Switch	128 Bits MKA	Line Rate	Line Rate	Line Rate	Line Rate	Line Rate
	256 Bits MKA	Not Supported	Line Rate	Line Rate	Line Rate	Line Rate

C9400 Sup 1XL-Y does not Support MACsec on any Supervisor ports
C9400 Sup 1 and 1XL support MACsec for only for interfaces with speed 10/40 Gbps

NE – Network Essentials. NA – Network Advantage.
Line rate is calculated with the additional MACsec header overhead

관련 정보

[보안 컨피그레이션 가이드, Cisco IOS® XE Gibraltar 16.12.x(Catalyst 9300 스위치)]

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.