# Secure Client 5를 사용하여 MKA 구성 및 문제 해결

## 목차



## 소개

이 문서에서는 Secure Client 5를 신청자로 사용하여 엔드포인트에서 MACsec 암호화를 구성하는 방법에 대해 설명합니다.

### 사전 요구 사항

Cisco에서는 다음 항목에 대한 지식을 권장합니다.

- ISE(Identity Services Engine)
- 802.1x 및 Radius
- MACsec MKA 암호화
- Secure Client 버전 5(이전의 Anyconnect)

### 사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

- ISE(Identity Services Engine) 버전 3.3
- Catalyst 9300 버전 17.06.05
- Cisco Secure Client 5.0.4032

### 배경 정보

MACSec(Media Access Control Security)는 802.1AE IEEE 표준에 정의된 OSI 모델(데이터 링크)의 레이어 2에서 이더넷 프레임에 대한 암호화 및 보호를 제공하는 네트워크 보안 표준입니다.

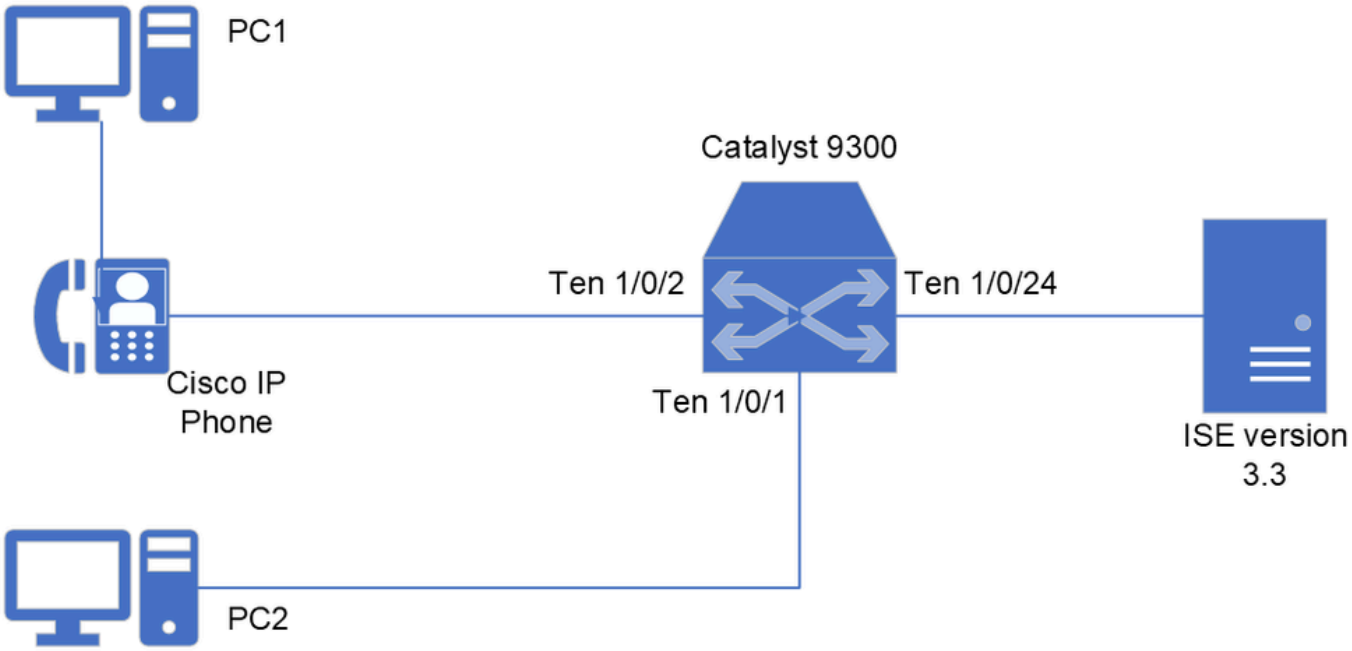
MACSec는 스위치 간 또는 스위치-호스트 간 연결이 될 수 있는 지점 간 연결에서 이 암호화를 제공하므로 이 표준의 적용 범위는 유선 연결로 제한됩니다.

이 표준은 레이어 2 연결에서 전송되는 프레임의 소스 및 대상 MAC 주소를 제외한 전체 데이터를 암호화합니다.

MACsec MKA(Key Agreement) 프로토콜은 MACsec 피어가 링크를 보호하는 데 필요한 보안 키를 협상하는 메커니즘입니다.

## 구성

### 네트워크 다이어그램




참고: 이 문서에서는 PC 디바이스 및 Cisco IP Phone에 대한 Radius 인증에 대해 구성되고 작동하는 규칙이 있는 것으로 간주합니다. 처음부터 컨피그레이션을 설정하려면 ISE Secure Wired Access Predictive Deployment Guide를 참조하여 ID 기반 네트워크 액세스를 위한 Identity Services Engine 및 스위치의 컨피그레이션을 검토하십시오.

## ISE에서 정책 설정

첫 번째 작업은 이전 다이어그램에 표시된 두 PC(및 Cisco IP Phone)에 적용되는 해당 권한 부여 프로파일을 구성하는 것입니다.

이 가상 시나리오에서 PC는 인증 방법으로 802.1X 프로토콜을 사용하고 Cisco IP Phone은 MAB(Mac Address Bypass)를 사용합니다.

ISE는 Radius 프로토콜을 통해 스위치와 통신하며, Radius 세션을 통해 엔드포인트가 연결된 인터페이스에서 스위치가 적용해야 하는 특성에 대해 설명합니다.

호스트의 MACsec 암호화의 경우, 필요한 특성은 cisco-av-pair = linksec-policy이며, 여기에는 다음 3가지 값이 있습니다.

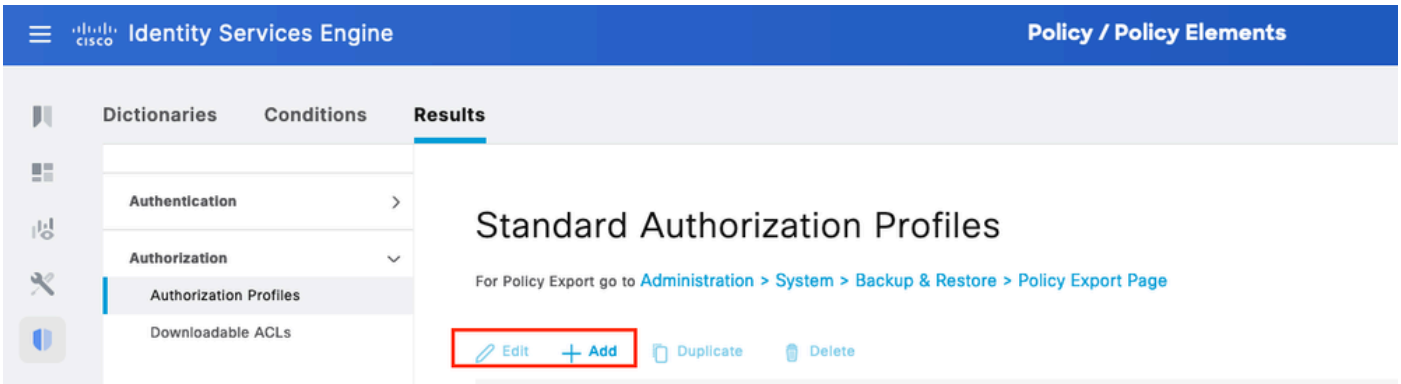
- 안전하지 않아야 함: 스위치는 Radius 세션이 발생하는 인터페이스에서 MKA 암호화를 수행하지 않습니다.

- 보안 필요: 스위치는 Radius 세션과 연결된 트래픽에 암호화를 적용해야 합니다. MKA 세션이 실패하거나 시간 초과가 있는 경우, 연결은 권한 부여 실패로 간주되며 MKA 세션을 설정하려는 또 다른 시도가 이루어집니다.
- 보안 필요: 스위치가 MKA 암호화를 시도합니다. Radius 세션에 연결된 MKA 세션이 성공하면 트래픽이 암호화됩니다. MKA가 실패하거나 시간 초과되면, 스위치는 암호화되지 않은 트래픽이 해당 Radius 세션에 연결되도록 허용합니다.

1단계. 두 PC 모두 MKA 기능이 없는 시스템이 인터페이스 Ten 1/0/1에 연결하는 경우 유연성을 갖도록 보안 MKA 정책을 적용합니다.

옵션으로, 반드시 보안해야 하는 정책을 적용하는 PC2용 정책을 구성할 수 있습니다.

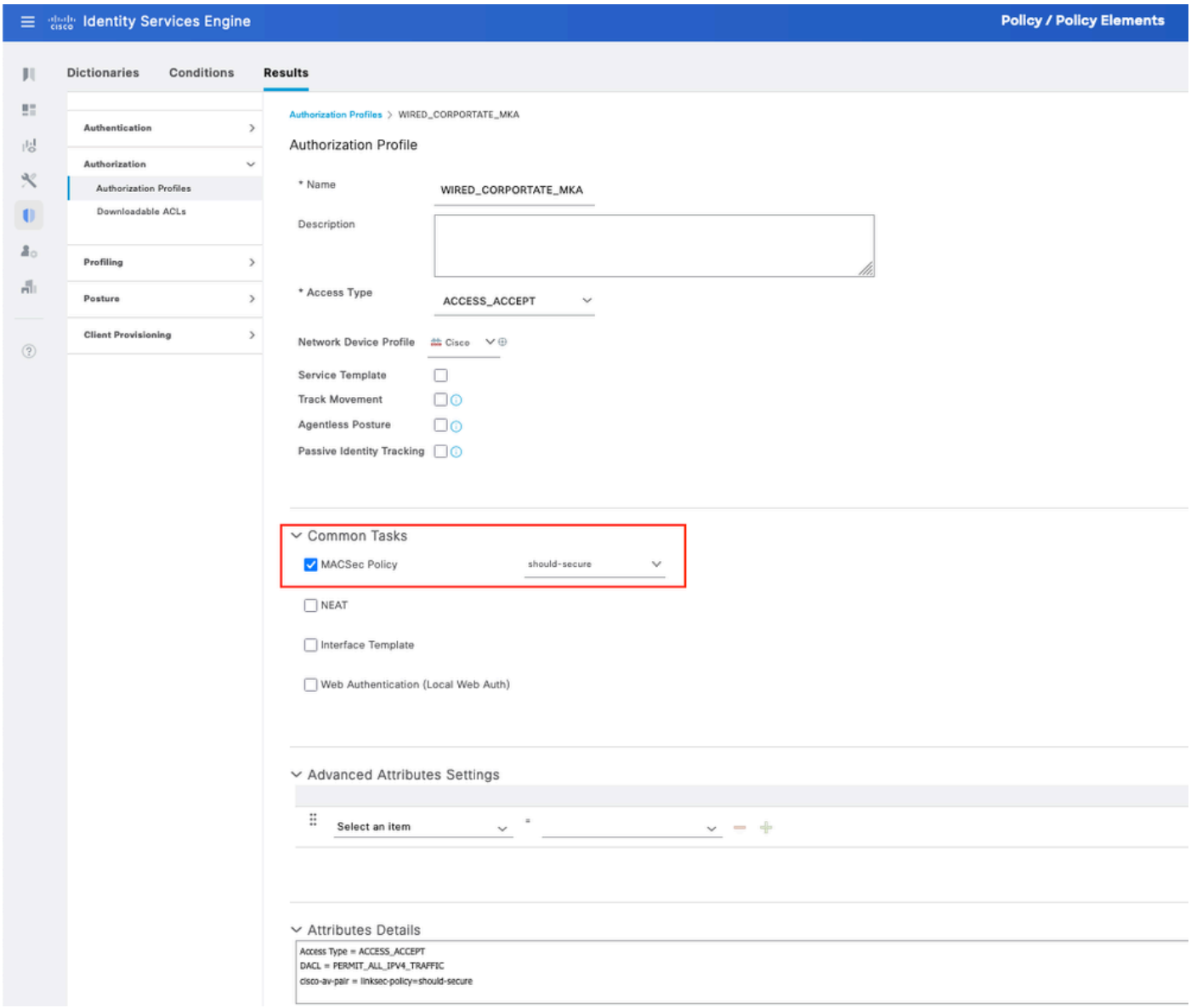
이 예에서는 Policy(정책) > Policy Elements(정책 요소) > Results(결과) > Authorization Profiles(권한 부여 프로파일)와 같이 PC에 대한 정책을 구성한 다음 기존 프로파일을 +Add(추가) 또는 Edit(수정)합니다




2단계. 프로필에 필요한 필드를 완성하거나 사용자 지정합니다.

Common Tasks에서 MACSec 정책과 적용할 해당 정책을 선택했는지 확인합니다.

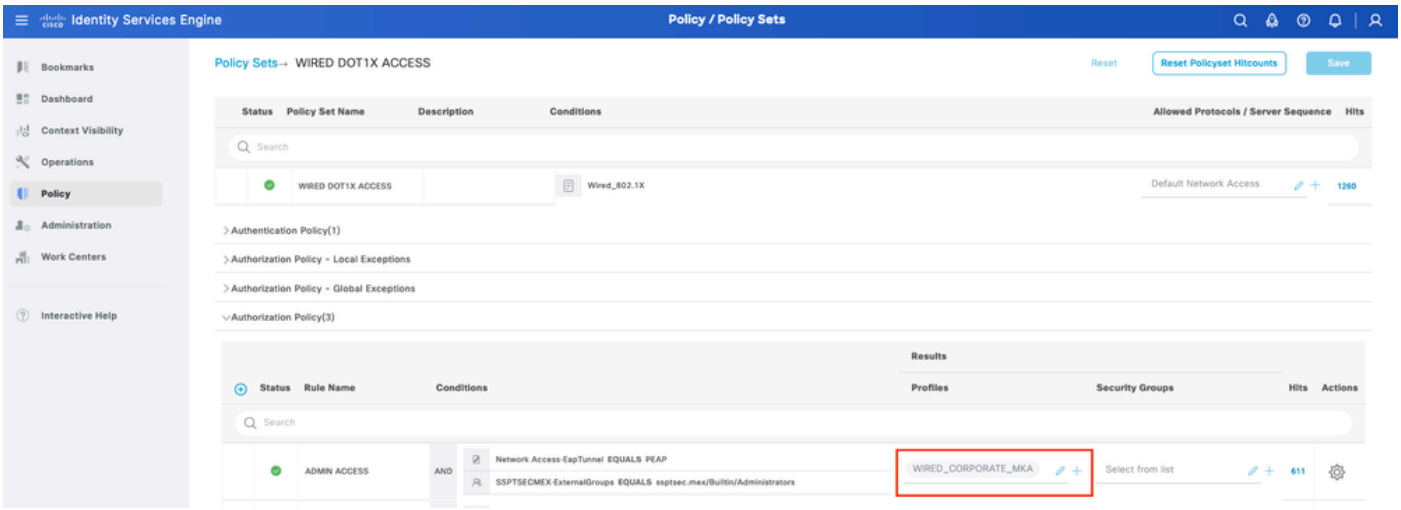
아래로 스크롤하여 컨피그레이션을 저장합니다.




3단계. 디바이스에서 적중하는 권한 부여 규칙에 해당 권한 부여 프로파일을 할당합니다.

Policy > Policy Sets > (Select Policy Set assigned) > Authorization Policy로 이동합니다.

권한 부여 규칙을 권한 부여 프로파일과 MACsec Settings(MACsec 설정)와 연결합니다. 아래로 스크롤하여 구성을 저장합니다.




## Catalyst 9300에서 MKA 설정

1단계. 다음 예제와 같이 새 MKA 정책을 구성합니다.

```
!
  mka policy MKA_PC
  key-server priority 0
  no delay-protection
  macsec-cipher-suite gcm-aes-128
  confidentiality-offset 0
  sak-rekey on-live-peer-loss
  sak-rekey interval 0
  include-icv-indicator
  no send-secure-announcements
  no use-updated-eth-header
  no ssci-based-on-sci
!
```

2단계. PC가 연결된 인터페이스에서 MACsec 암호화를 활성화합니다.

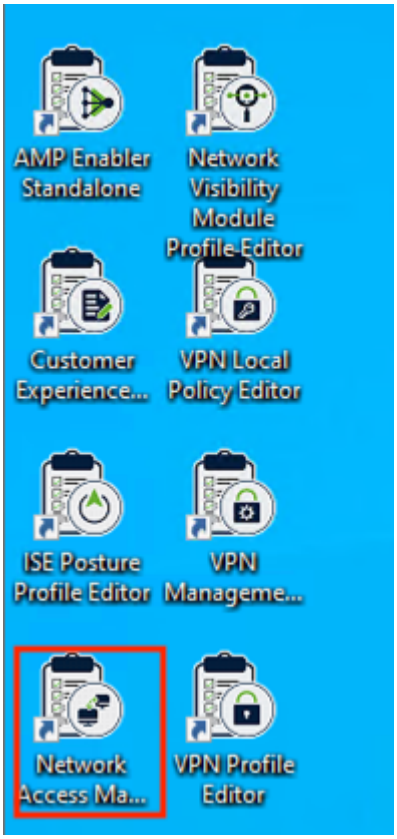
```
!
interface TenGigabitEthernet1/0/1
 macsec
 mka policy MKA_PC
!
```

참고: MKA 컨피그레이션의 명령 및 옵션과 관련된 자세한 내용은 사용하는 스위치 버전에 해당하는 보안 컨피그레이션 가이드를 참조하십시오. 이 예의 이 시나리오에서는 Security Configuration Guide, Cisco IOS XE Bengaluru 17.6.x(Catalyst 9300 스위치)

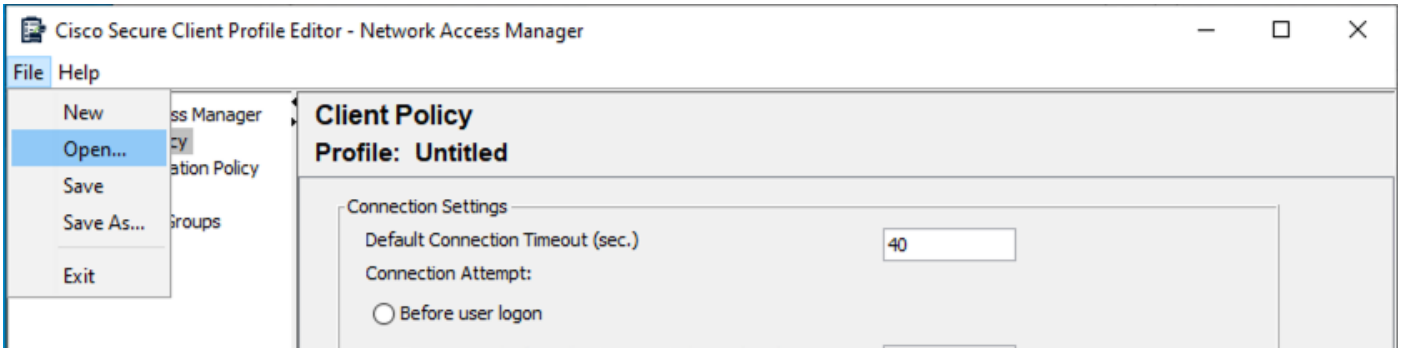
Network Access Manager 프로파일 편집기를 사용하여 MKA를 설정합니다.

1단계. (Cisco의 다운로드 웹 사이트에서) 다운로드하고 사용 중인 Secure Client 버전과 일치하는 프로파일 편집기를 엽니다.

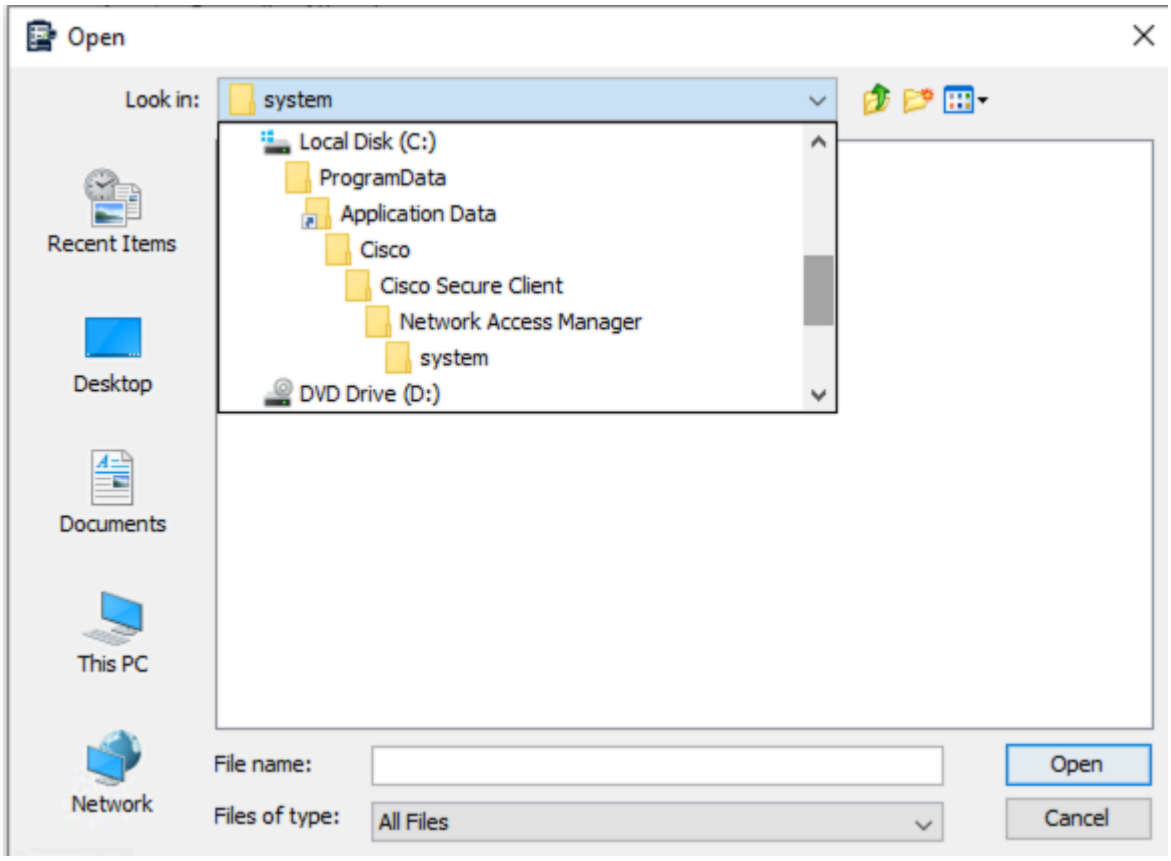
컴퓨터에 이 프로그램을 설치한 후 Cisco Secure Client Profile Editor - Network Access Manager를 엽니다.



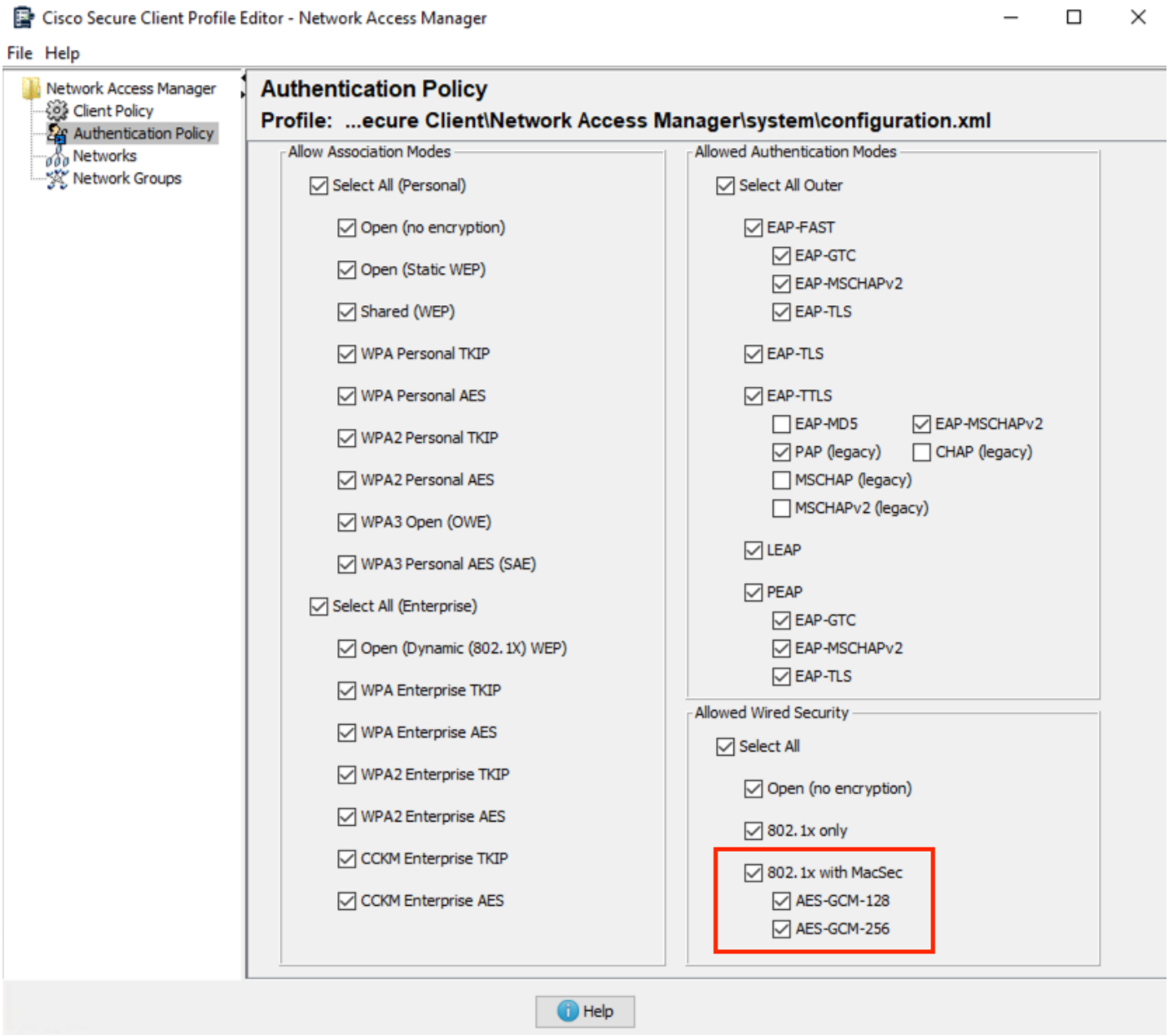

2단계. 파일 > 열기를 선택합니다.




3단계. 이 이미지에 표시되는 폴더 시스템을 선택합니다. 이 폴더에서 configuration.xml이라는 파일을 엽니다.

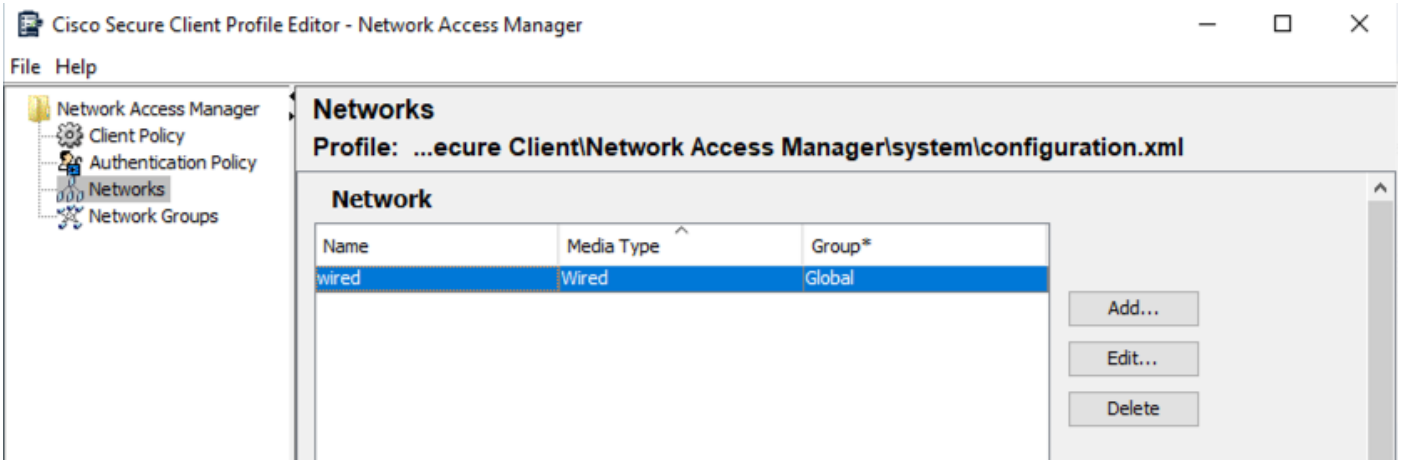



4단계. 프로파일 편집기에서 파일을 로드했으면 Authentication Policy(인증 정책) 옵션을 선택하고 802.1x with MACSec 관련 옵션이 활성화되었는지 확인합니다.



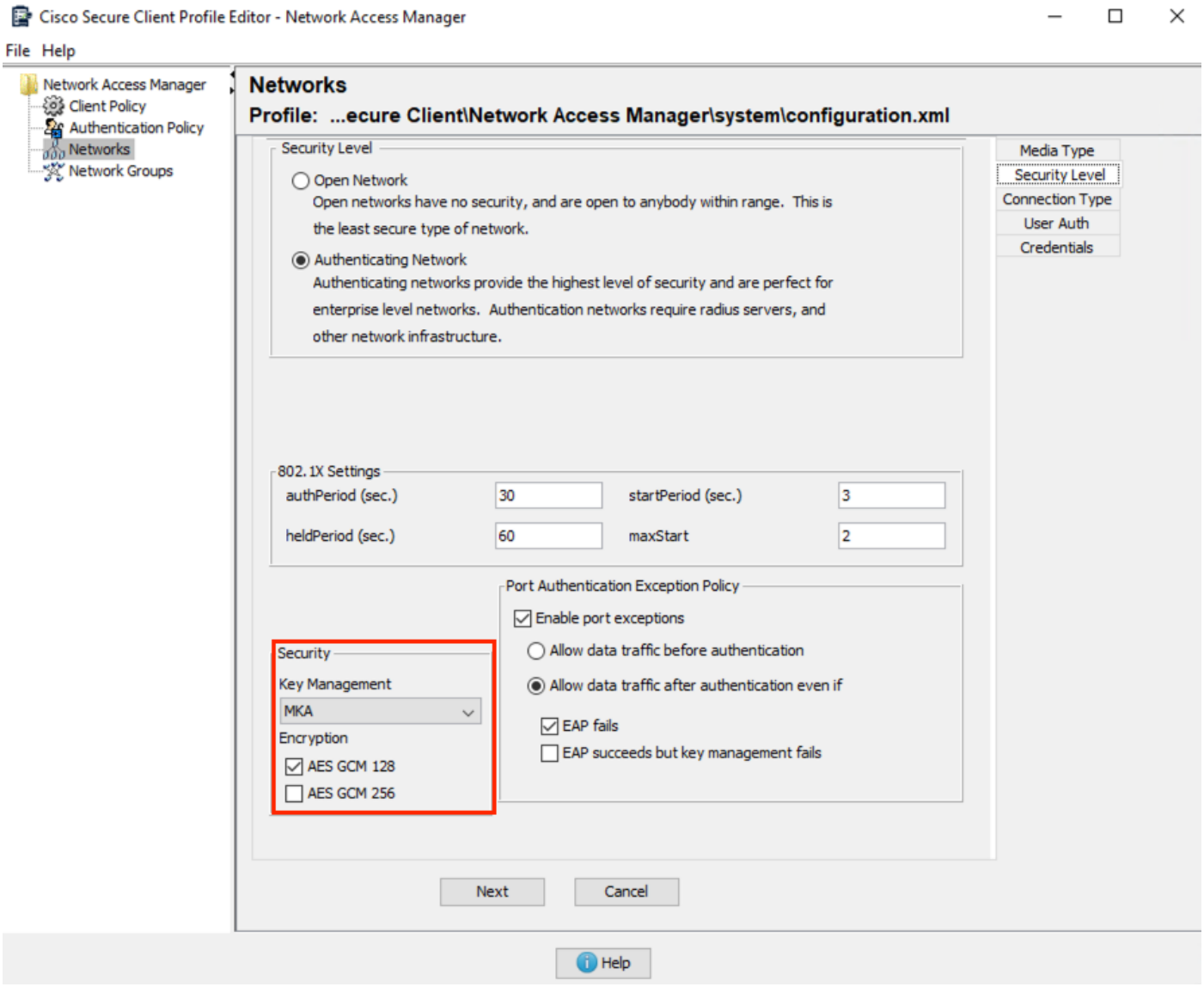

5단계. Networks(네트워크) 섹션으로 진행합니다. 이 부분에서는 유선 연결을 위한 새 프로파일을 추가하거나 Secure Client 5.0에 설치된 기본 유선 프로파일을 편집할 수 있습니다.

이 시나리오에서는 기존 유선 프로파일을 수정하겠습니다.




6단계. 프로파일을 구성합니다. Security Level(보안 레벨) 섹션에서 Key Management(키 관리)를 조정하여 MKA를 사용하고 그 다음에 암호화 AES GCM 128을 사용합니다.

인증 dot1x 및 정책에 대한 다른 매개 변수를 조정합니다.

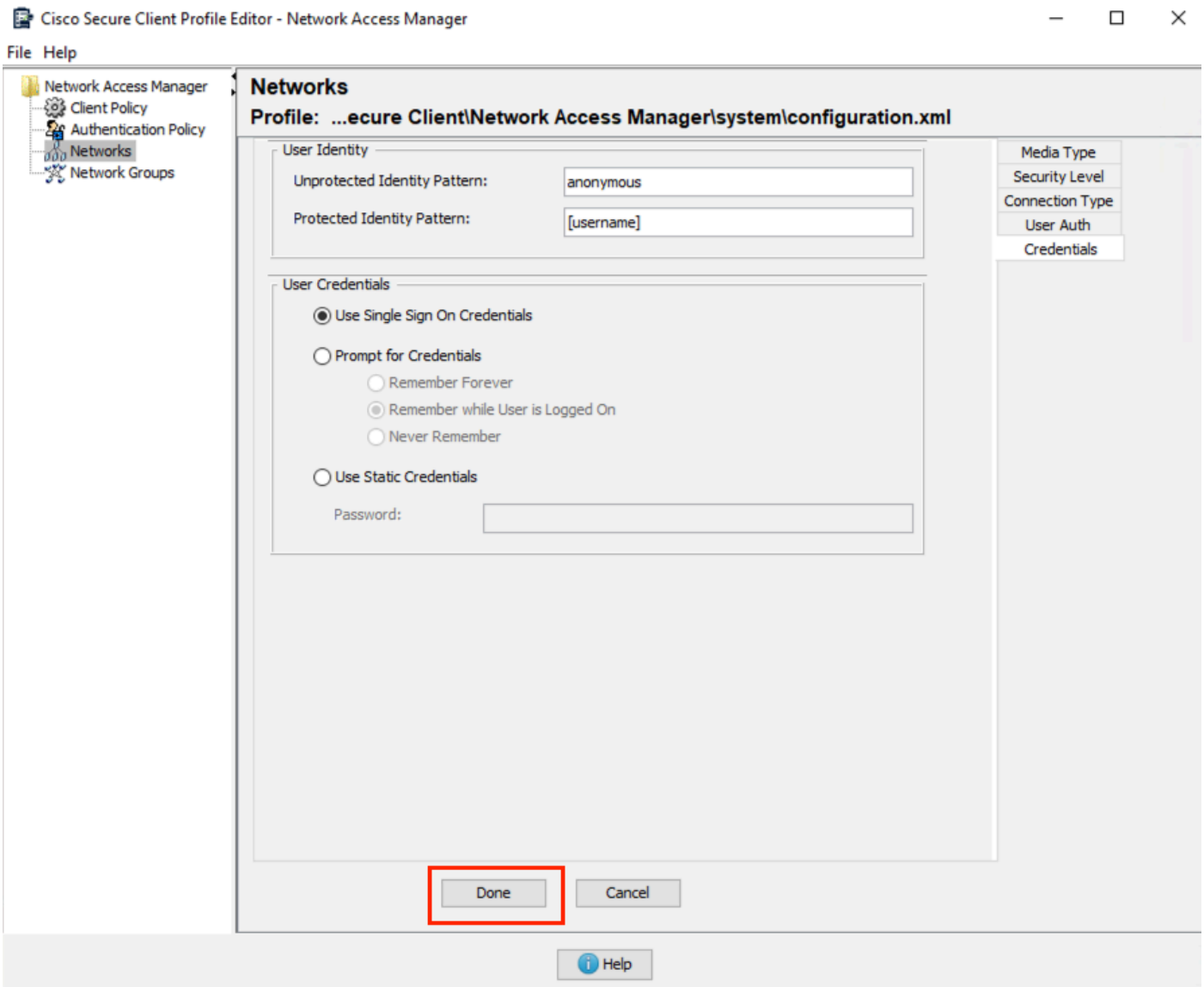


7단계. 연결 유형, 사용자 인증 및 자격 증명에 대한 나머지 섹션을 구성합니다¶

이러한 섹션은 Security Level 섹션에서 선택한 인증 설정에 따라 달라집니다.

컨피그레이션을 마치면 완료를 클릭합니다.

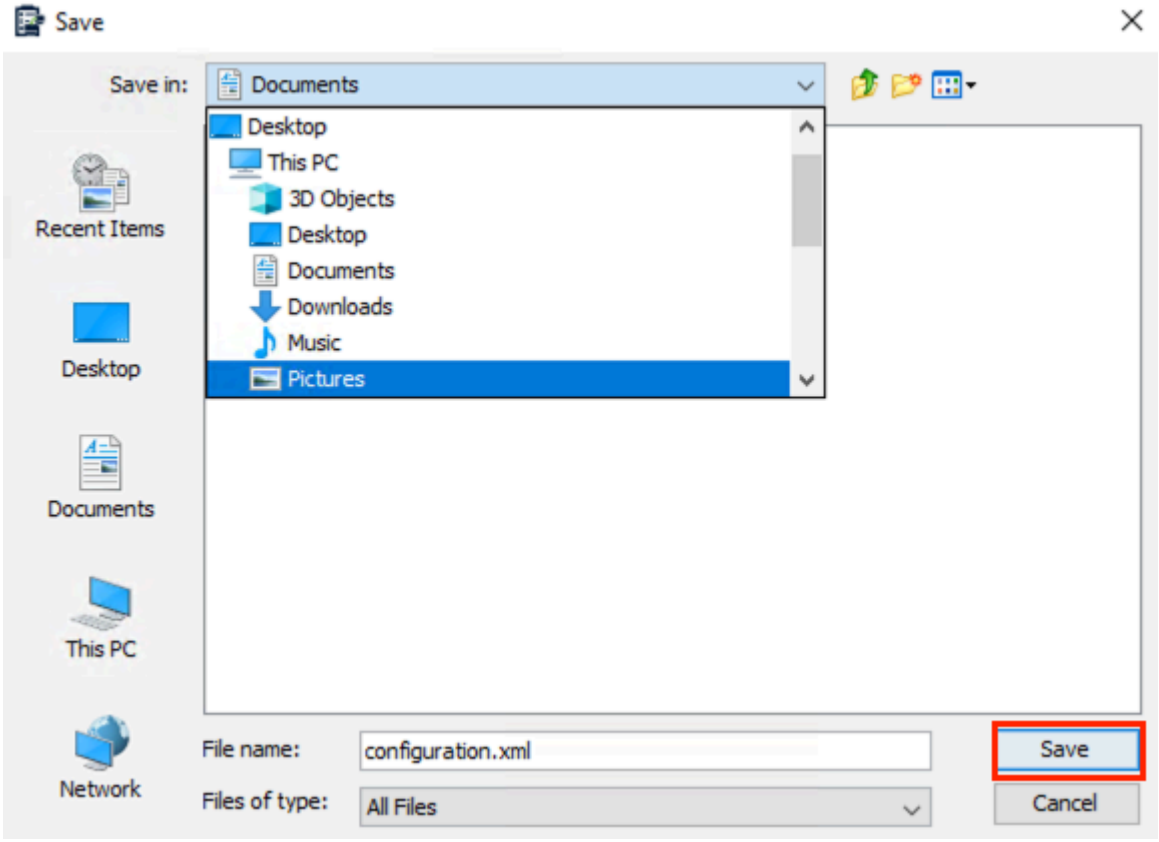
이 시나리오에서는 사용자 자격 증명과 함께 PEAP(Protected Extensible Authentication Protocol)를 사용합니다.




8단계. File(파일) 메뉴로 이동합니다. Save as(다른 이름으로 저장) 옵션을 계속 실행합니다.

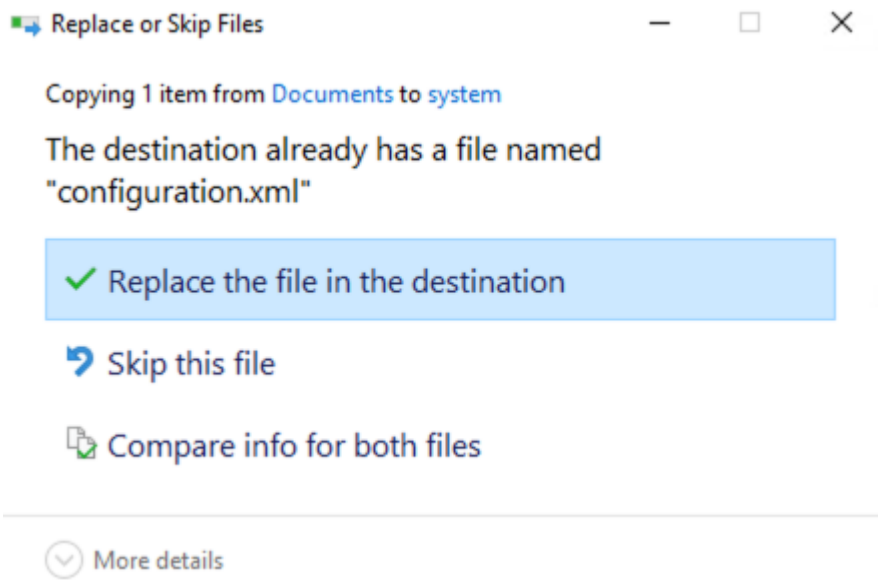
파일 이름을 configuration.xml로 지정하고 ProgramData\Cisco\Cisco Secure Client\Network Access Manager\system과 다른 폴더에 저장합니다.

이 예제에서는 파일이 Documents 폴더에 저장되었습니다. 프로파일을 저장합니다.



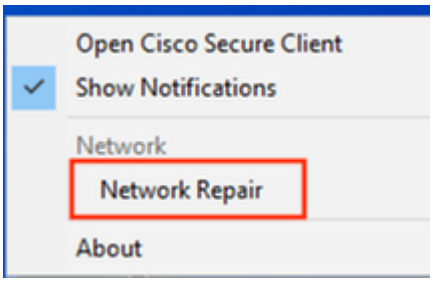

8단계. 프로파일 위치로 이동하여 파일을 복사한 다음 ProgramData\Cisco\Cisco Secure Client\Network Access Manager\system 폴더에 들어 있는 파일을 바꿉니다.

[대상에서 파일 바꾸기] 옵션을 선택합니다.




9단계. Security Client 5.0에서 수정된 프로필을 로드하려면 Windows 시스템의 오른쪽 아래 작업 표시줄에 있는 Secure Client 아이콘을 마우스 오른쪽 단추로 클릭합니다.

네트워크 복구를 수행합니다.

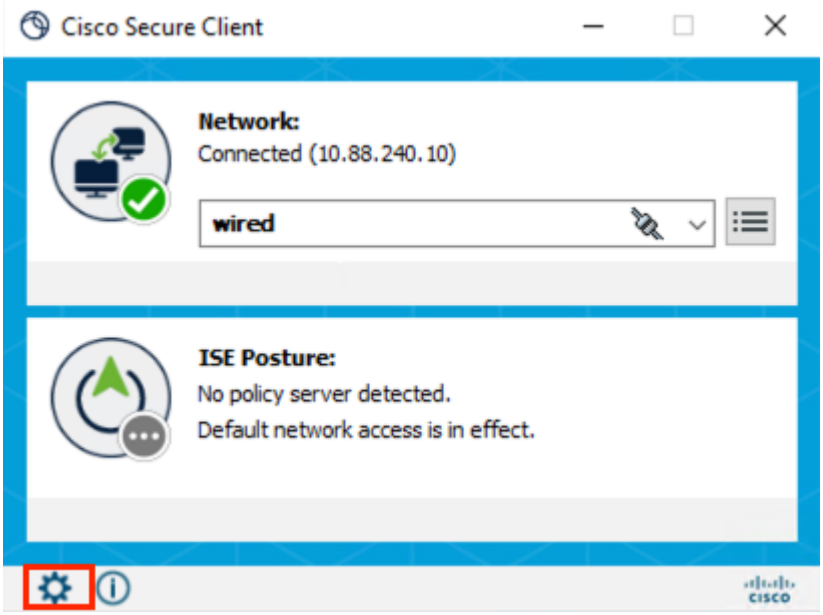



참고: 프로파일 편집기를 통해 구성된 모든 네트워크에는 관리자 네트워크 권한이 있으므로 사용자는 이 도구를 사용하여 구성한 콘텐츠를 사용자 지정/변경할 수 없습니다.

## Network Access Manager를 사용한 MKA 네트워크 설정(선택 사항).

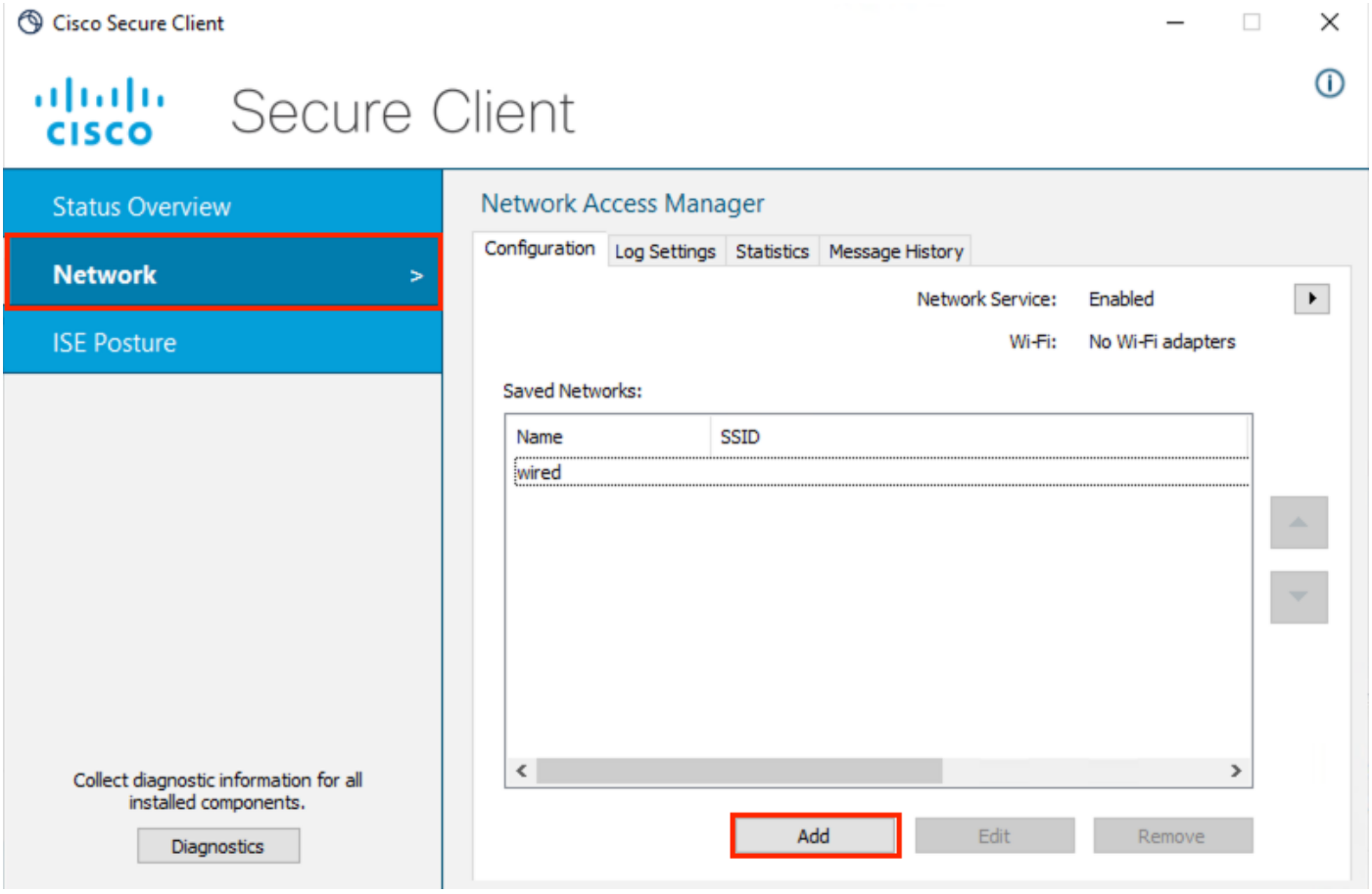
1단계. 프로파일 편집기를 사용하는 MKA 설정의 대안으로 이 도구를 사용하지 않고 네트워크를 추가할 수 있습니다.

Secure Client 제품군에서 기어 아이콘을 선택합니다.



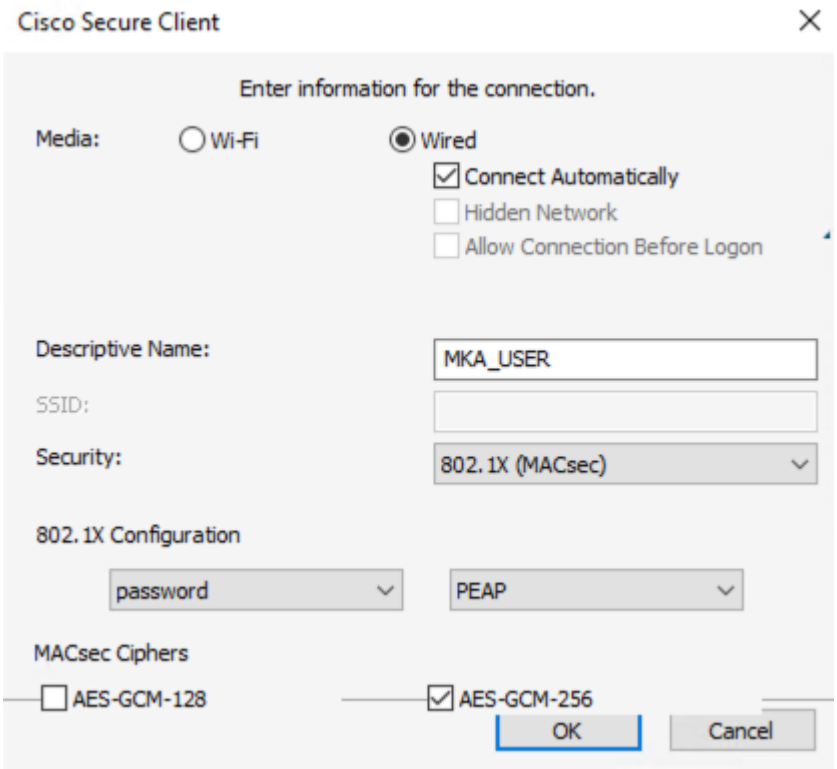

2단계. 표시된 새 창에서 Network(네트워크) 옵션을 선택합니다.

Configuration(컨피그레이션) 섹션에서 Add to ingress an network MKA capable with privileges User Network(사용자 네트워크 권한이 있는 네트워크 MKA를 인그레스하려면 추가) 옵션을 선택합니다.




3단계. 새 컨피그레이션 창에서 연결의 특성을 설정하고 네트워크의 이름을 지정합니다.

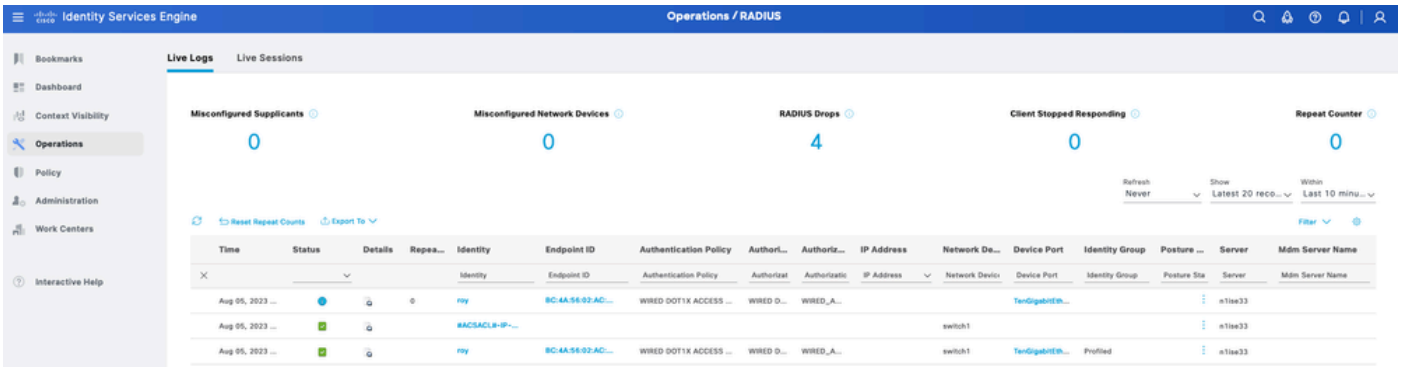
완료되면 확인 단추를 선택합니다.



# 다음을 확인합니다.

## ISE에서 검증

ISE에서 이 플로우의 컨피그레이션이 완료되면 디바이스가 Livelogs에서 인증되고 권한이 부여됨을 확인합니다.




Details of the authentication and the Result(인증 세부사항 및 결과) 섹션으로 이동합니다.

권한 부여 프로파일에 설정된 특성은 NAD(Network Access Device)로 전송되며 하나의 Essential 라이센스를 사용합니다.

| | |
|---|---|
| cisco-av-pair | linksec-policy=should-secure |
| cisco-av-pair | ACS:CiscoSecure-Defined-ACL=#ACSACL#-IP-PERMIT_ALL_IPV4_TRAFFIC-57f6b0d3 |
| MS-MPPE-Send-Key | **** |
| MS-MPPE-Recv-Key | **** |
| LicenseTypes | Essential license consumed. |

## Catalyst 스위치에 대한 검증

이러한 명령은 이 솔루션의 적절한 기능을 검증하는 데 사용할 수 있습니다.

```
switch1#show mka policy
```

```
switch1#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

MKA_PC            0    FALSE 0  FALSE FALSE  GCM-AES-128     Te1/0/1        Te1/0/2
```

switch1#show mka session

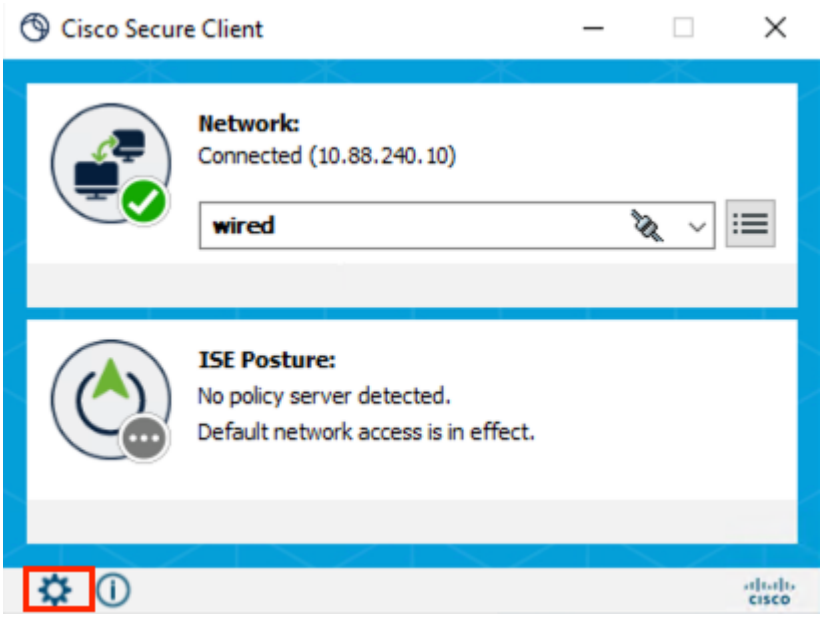
```
switch1#show mka session

Total MKA Sessions..... 2
      Secured Sessions... 2
      Pending Sessions... 0

=============================================================================================
Interface      Local-TxSCI         Policy-Name      Inherited        Key-Server
Port-ID        Peer-RxSCI          MACsec-Peers     Status           CKN
=============================================================================================
Te1/0/1        ac7a.5646.4d01/0002 MKA_PC           NO               YES
2              bc4a.5602.ac25/0000 1                Secured          60E8BC2A9EF5FE11532428862C747640

Te1/0/2        ac7a.5646.4d02/0002 MKA_PC           NO               YES
2              bc4a.5602.ac26/0000 1                Secured          C79300869F539BB534350133064744B6
```

switch1#show authentication session interface <interface_ID> detail

```
switch1#show authentication session interface ten 1/0/2 detail
            Interface:  TenGigabitEthernet1/0/2
                IIF-ID:  0x19FA2D8B
           MAC Address:  bc4a.5602.ac26
          IPv6 Address:
          IPv4 Address:
             User-Name:  alice
                Status:  Authorized
                Domain:  DATA
        Oper host mode:  multi-domain
      Oper control dir:  both
       Session timeout:  N/A
     Common Session ID:  C5AA580A00000030CD72CFE6
       Acct Session ID:  0x00000017
                Handle:  0x62000016
        Current Policy:  POLICY_Te1/0/2


Local Policies:
        Service Template: DEFAULT_LINKSEC_POLICY_SHOULD_SECURE (priority 150)

Server Policies:
      Security Policy:  Should Secure
              ACS ACL: #ACSACL#-IP-PERMIT_ALL_IPV4_TRAFFIC-57f6b0d3
      Security Status:  Link Secured

Method status list:
       Method           State
        dot1x           Authc Success

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

            Interface:  TenGigabitEthernet1/0/2
                IIF-ID:  0x101218F4
           MAC Address:  d4ad.bd2a.cbab
          IPv6 Address:
          IPv4 Address:
             User-Name:  D4-AD-BD-2A-CB-AB
                Status:  Authorized
                Domain:  VOICE
        Oper host mode:  multi-domain
      Oper control dir:  both
       Session timeout:  N/A
     Common Session ID:  C5AA580A00000040CD8001B3
       Acct Session ID:  0x0000001a
                Handle:  0xa1000018
        Current Policy:  POLICY_Te1/0/2

Local Policies:
        Service Template: DEFAULT_LINKSEC_POLICY_SHOULD_SECURE (priority 150)
      Security Policy:  Should Secure
      Security Status:  Link Unsecured

Server Policies:
              ACS ACL: xACSACLx-IP-DENY_ALL_IPV4_TRAFFIC-57f6b0d3

Method status list:
       Method           State
          mab           Authc Success
```

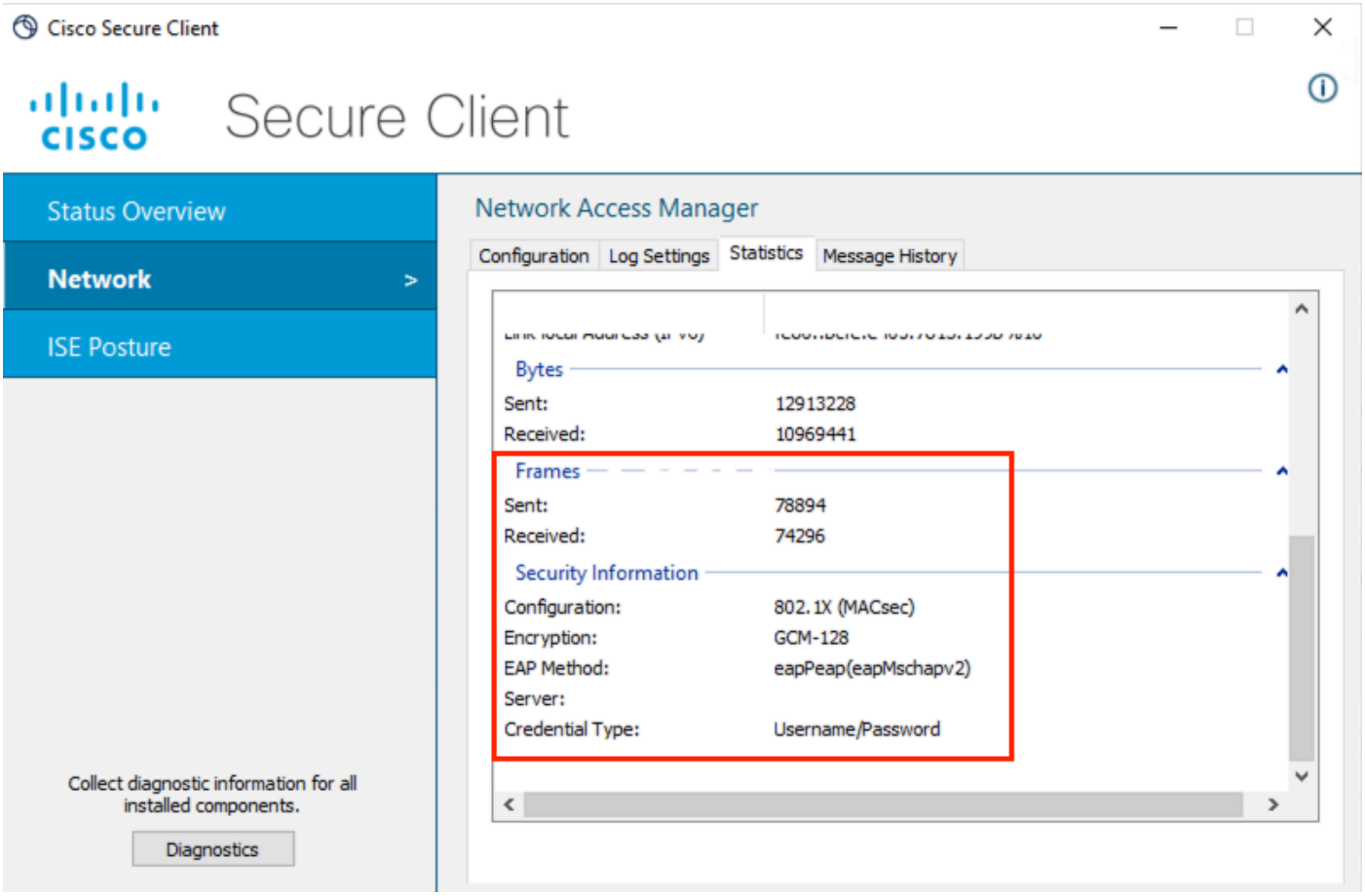
## Secure Client에 대한 검증

MACsec 암호화로 생성된 프로필로 인증에 성공했습니다. 엔진 아이콘을 클릭하면 추가 정보가 표시될 수 있습니다.




Secure Client(보안 클라이언트)에 대해 여기에 표시된 메뉴에서 Network Access Manager(네트워크 액세스 관리자) > Statistics(통계) 섹션에서 암호화 및 해당 MACsec 컨피그레이션을 확인합니다.

레이어 2에서 암호화를 수행할 때 수신 및 전송되는 프레임이 증가합니다.

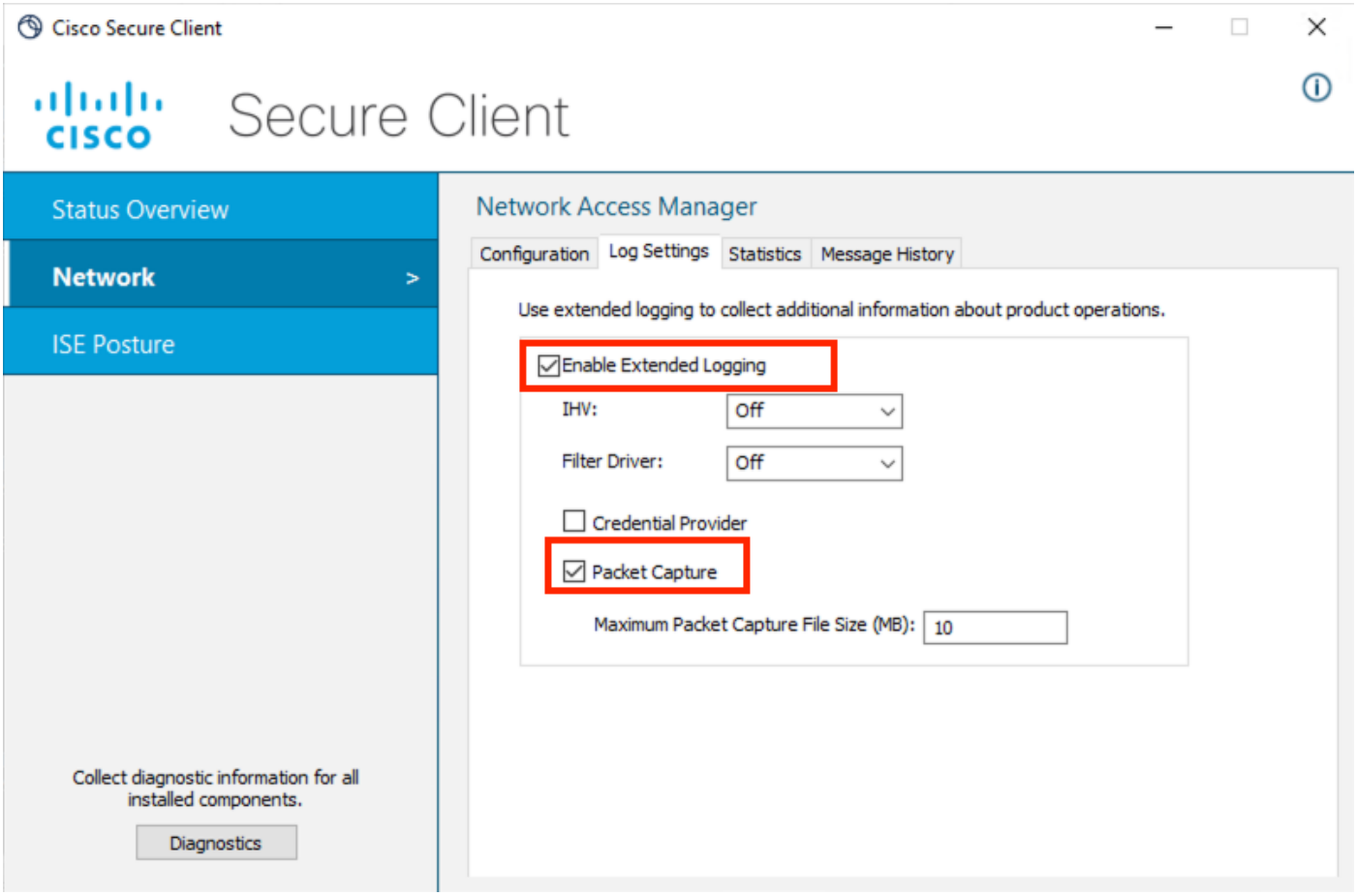



# 문제 해결

참고: 이 섹션에서는 발생할 수 있는 MKA 문제와 관련된 트러블슈팅 부분을 다룹니다. 인증 또는 권한 부여 실패에 직면한 경우 ISE Secure Wired Access Predictive Deployment Guide - Troubleshooting to investigate를 참조하십시오. 이 설명서에서는 인증이 MACsec 암호화 없이 정상적으로 작동한다고 가정합니다.

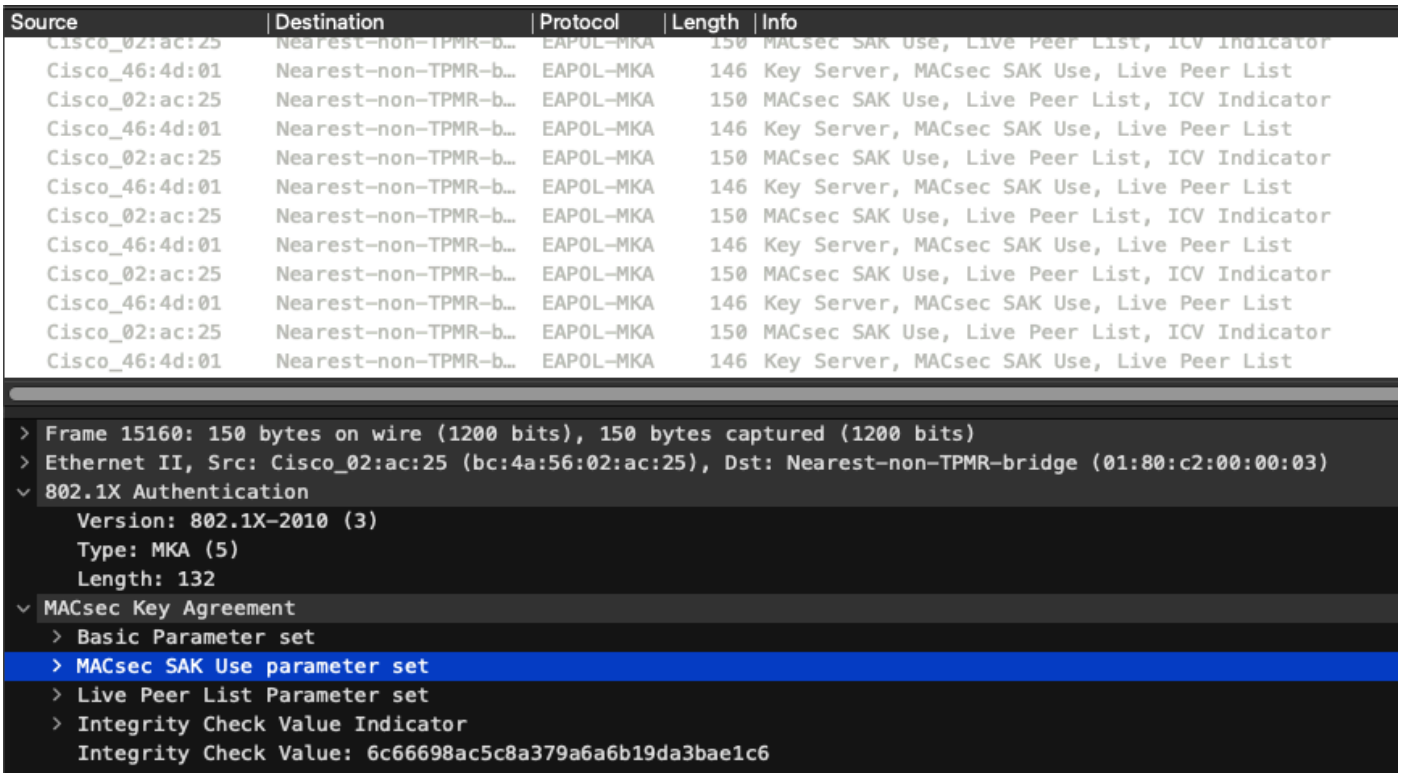
## Cisco Secure Endpoint

- Secure Endpoint 제품군에서 DART 모듈을 활성화합니다.
- 이 메뉴에서 Extended Logging(확장 로깅)을 활성화하여 사용자 MKA 연결에 대한 추가 데이터를 수집합니다. DART 번들에 포함된 패킷 캡처를 활성화할 수도 있습니다.




- configuration.xml 및 Network access Manager 분석을 진행하려면 DART 번들을 수집합니다. 문제 해결을 위한 데이터 수집을 위해 DART 실행 설명서를 참조하십시오

이 예에서는 호스트와 스위치 간의 정보가 패킷으로 표시되는 방식을 표시합니다.




DART 번들에서 NetworkAccessManager.txt라는 로그에서 인증 802.1X 및 MKA 세션에 대한 유용한 정보를 찾을 수 있습니다.

이 정보는 MKA 암호화를 사용하는 성공적인 인증에 표시됩니다.

```
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: PORT (1) net: RECV (status: UP, AUTO) (portMsg.c 709)
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: 8021X (2) RECEIVED SUCCESS (dot1x_util.c 326)
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: STATE (2) current state = AUTHENTICATING (dot1x_sm.c 323)
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: STATE (2) S_enterStateAux called with state = AUTHENTICATI
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: STATE (2) staying in 802.1x state: AUTHENTICATING (dot1x_s
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: 8021X (2) smTimer: sec=30 (dot1x_util.c 454)
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: STATE (2) eap_type<0>, lengths<4,1496> (dot1x_proto.c 90)
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: 8021X (2) smTimer: paused (dot1x_util.c 484)
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: EAP (0) Received EAP-Success. (eap_auth_client.c 835)
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: EAP (0) tlsAuthOnAuthEnd: clear TLS session (eap_auth_tls_
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: EAP (0) tlsAuthOnAuthEnd: successful authentication, save
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: EAP (3) new credential list saved (eapRequest.c 1485)
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: EAP (3) EAP status: AC_EAP_STATUS_EAP_SUCCESS (eapMessage.
%csc_nam-7-DEBUG_MSG: %[tid=9028]: EAP-CB: EAP status notification: session-id=1, handle=04B2DD44, stat
%csc_nam-7-DEBUG_MSG: %[tid=9028]: EAP-CB: sending EapStatusEvent...
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: PORT (2) EAP response received. <len:400> <res:2> (dot1x_p
%csc_nam-7-DEBUG_MSG: %[tid=2716]: EAP: ...received EapStatusEvent: session-id=1, EAP handle=04B2DD44,
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: 8021X (2) smTimer: activated (dot1x_util.c 503)
%csc_nam-6-INFO_MSG: %[tid=2716]: EAP: Eap status AC_EAP_STATUS_EAP_SUCCESS.
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: STATE (2) current state = AUTHENTICATING (dot1x_sm.c 323)
%csc_nam-7-DEBUG_MSG: %[tid=2716]: EAP: processing EapStatusEvent in the subscriber
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: STATE (2) dot1x->eapSuccess is True (dot1x_sm.c 352)
%csc_nam-7-DEBUG_MSG: %[tid=2716]: Auth[wired:user-auth]: Enabling fast reauthentication
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: STATE (2) SUCCESS (dot1x_sm.c 358)
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: STATE (2) S_enterStateAux called with state = AUTHENTICATE
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: STATE (2) S_enterStateAux calling sm8Event8021x due to aut
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: 8021X (2) smTimer: disabled (dot1x_util.c 460)
```

```
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: MKA (0) NASP: dot1xAuthSuccessEvt naspStopEapolAnnouncement
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: NASP (0) >> NASP: naspStopEapolAnnouncement (nasp.c 900)
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: NASP (0) << NASP: naspStopEapolAnnouncement.  err = 0 (nasp
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: 8021X (2) dot1x->config.useMka = 1 (dot1x_main.c 829)
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: INF (2) >> MKA: StartSession (mka.c 511)
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: MKA (2) MKA: bUseMka = 1, bUseMacSec = 1706033334, MacsecSu
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: MKA (2) >> MKA: InitializeContext (mka.c 1247)
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: INF (2) MKA: Changing state to Unconnected (mka.c 1867)
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: INF (2) MKA: Changing Sak State to Idle (mka.c 1271)
%csc_nam-7-DEBUG_MSG: %[tid=2716]: Auth[wired:user-auth]: Fast reauthentication enabled on authenticati
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: MKA (2) << MKA: InitializeContext (mka.c 1293)
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: MKA (2) MKA: Changing state to Need Server (mka.c 1871)
%csc_nam-7-DEBUG_MSG: %[tid=2716]: Auth[wired:user-auth]: Sending NOTIFICATION__SUCCESS to subscribers
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: INF (2) >> MKA: CreateKeySet (mka.c 924)
%csc_nam-7-DEBUG_MSG: %[tid=2716]: Network auth request NOTIFICATION__SUCCESS
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: NASP (0) >> NASP: naspGetNetCipherSuite (nasp.c 569)
%csc_nam-6-INFO_MSG: %[tid=9028][comp=SAE]: MKA (2) MKA: Key length is 16 bytes (mka.c 954)
%csc_nam-7-DEBUG_MSG: %[tid=2716]: Auth[wired:user-auth]: Finishing authentication
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: MKA (2) MKA: MyMac (mka.c 971)
%csc_nam-7-DEBUG_MSG: %[tid=2716]: Auth[wired:user-auth]: Authentication finished
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: API (1) event: STATUS - AC_PORT_STATUS_EAP_SUCCESS (portWo
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: API (1) event: complete (portWorkList.c 130)
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: API (1) event: STATUS - AC_PORT_STATUS_MKA_UNCONNECTED (po
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: API (1) event: complete (portWorkList.c 130)
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: API (1) event: STATUS - AC_PORT_STATUS_MKA_NEED_SERVER (po
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: API (1) event: complete (portWorkList.c 130)
%csc_nam-7-DEBUG_MSG: %[tid=8140][comp=SAE]: NET (0) SscfCallback(1): SSCF_NOTIFICATION_CODE_SEND_PACKE
%csc_nam-7-DEBUG_MSG: %[tid=8140][comp=SAE]: NET (0) CIMD Event: evtSeq#=0 msg=4 ifIndex=1 len=36 (cimd
%csc_nam-7-DEBUG_MSG: %[tid=8140][comp=SAE]: NET (1) cdiEvt:(3,0) dataLen=4 (cimdEvt.c 358)
%csc_nam-7-DEBUG_MSG: %[tid=8140][comp=SAE]: NET (1) cdiEvt:(3,1) dataLen=102 (cimdEvt.c 358)
%csc_nam-7-DEBUG_MSG: %[tid=8140][comp=SAE]: NET (1) netEvent(1): Recv queued (netEvents.c 91)
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: PORT (1) net: RECV (status: UP, AUTO) (portMsg.c 709)
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: MKA (2) >> MKA: EapolInput (mka.c 125)
%csc_nam-7-DEBUG_MSG: %[tid=9028][comp=SAE]: MKA (2) MKA: MKPDU In (mka.c 131)
```

## Cisco IOS 문제 해결

이러한 명령은 NAD(Network Access Device)에서 플랫폼과 신청자 간의 MKA 암호화를 검토하도록 구현할 수 있습니다.

명령에 대한 자세한 내용은 NAD로 사용되는 플랫폼의 해당 컨피그레이션 가이드를 참조하십시오.

```
#show authentication session interface <interface_ID> detail
#show mka summary
#show mka policy
#show mka session interface <interface_ID> detail
#show macsec summary
#show macsec interface <interface_ID>
#debug mka events
#debug mka errors
#debug macsec event
#debug macsec error
```

이는 호스트에 대한 하나의 성공적인 MKA 연결의 디버깅입니다. 이 정보를 참조용으로 사용할 수 있습니다.

```
%LINK-3-UPDOWN: Interface TenGigabitEthernet1/0/1, changed state to down
Macsec interface TenGigabitEthernet1/0/1 is UP
MKA-EVENT: Create session event: derived CKN 9F0DC198A9728FB3DA198711B58570E4, len 16
MKA-EVENT EC000025: SESSION START request received...
NGWC-MACSec: pd get port capability is invoked
MKA-EVENT: New MKA Session on Interface TenGigabitEthernet1/0/1 with Physical Port Number 9 is using th
MKA-EVENT: New VP with SCI AC7A.5646.4D01/0002 on interface TenGigabitEthernet1/0/1
MKA-EVENT: Created New CA 0x80007F30A6B46F20 Participant on interface TenGigabitEthernet1/0/1 with SCI
%MKA-5-SESSION_START: (Te1/0/1 : 2) MKA Session started for RxSCI bc4a.5602.ac25/0000, AuditSessionID C
MKA-EVENT: Started a new MKA Session on interface TenGigabitEthernet1/0/1 for Peer MAC bc4a.5602.ac25 w
MKA-EVENT bc4a.5602.ac25/0000 EC000025: FSM (Init MKA Session) - Successfully derived CAK.
MKA-EVENT bc4a.5602.ac25/0000 EC000025: Successfully initialized a new MKA Session (i.e. CA entry) on i
MKA-EVENT bc4a.5602.ac25/0000 EC000025: FSM (Derive KEK/ICK) - Successfully derived KEK...
MKA-EVENT bc4a.5602.ac25/0000 EC000025: FSM (Derive KEK/ICK) - Successfully derived ICK...
MKA-EVENT bc4a.5602.ac25/0000 EC000025: New Live Peer detected, No potential peer so generate the first
MKA-EVENT bc4a.5602.ac25/0000 EC000025: >> FSM - Generate SAK for CA with CKN 9F0DC198 (Latest AN=0, Ol
MKA-EVENT bc4a.5602.ac25/0000 EC000025: Generation of new Latest SAK succeeded (Latest AN=0, KN=1)...
MKA-EVENT bc4a.5602.ac25/0000 EC000025: >> FSM - Install RxSA for CA with CKN 9F0DC198 on VP with SCI A
MKA-EVENT bc4a.5602.ac25/0000 EC000025: Clean up the Rx for dormant peers
MACSec-IPC: send_xable send msg success for switch=1
MACSec-IPC: blocking enable disable ipc req
MACSec-IPC: watched boolean waken up
MACSec-IPC: geting switch number
MACSec-IPC: switch number is 1
MACSec-IPC: create_tx_sc send msg success
Send create_tx_sc to IOMD successfully
alloc_cache called TxSCI: AC7A56464D010002 RxSCI: BC4A5602AC250000
Enabling replication for slot 1 vlan 330 and the ref count is 1
MACSec-IPC: vlan_replication send msg success
Added replication for data vlan 330
MACSec-IPC: geting switch number
MACSec-IPC: switch number is 1
MACSec-IPC: create_rx_sc send msg success
Sent RXSC request to FED/IOMD
MACSec-IPC: geting switch number
MACSec-IPC: switch number is 1
MACSec-IPC: install_rx_sa send msg success
Sent ins_rx_sa to FED and IOMD
MKA-EVENT bc4a.5602.ac25/0000 EC000025: Requested to install/enable new RxSA successfully (AN=0, KN=1 S
%LINEPROTO-5-UPDOWN: Line protocol on Interface TenGigabitEthernet1/0/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan330, changed state to up
MKA-EVENT bc4a.5602.ac25/0000 EC000025: Sending SAK for AN 0 resp peers 0 cap peers 1
MKA-EVENT bc4a.5602.ac25/0000 EC000025: SAK Wait Timer started for 6 seconds.
MKA-EVENT bc4a.5602.ac25/0000 EC000025: (KS) Received new SAK-Use response to Distributed SAK for AN 0,
MKA-EVENT bc4a.5602.ac25/0000 EC000025: (KS) All 1 peers with the required MACsec Capability have indic
MKA-EVENT: Reqd to Install TX SA for CA 0x80007F30A6B46F20 AN 0 CKN 9F0DC198 - on int(TenGigabitEtherne
MKA-EVENT bc4a.5602.ac25/0000 EC000025: >> FSM - Install TxSA for CA with CKN 9F0DC198 on VP with SCI A
MACSec-IPC: geting switch number
MACSec-IPC: switch number is 1
MACSec-IPC: install_tx_sa send msg success
MKA-EVENT bc4a.5602.ac25/0000 EC000025: Before sending SESSION_SECURED status - SECURED=false, PREVIOUS
MKA-EVENT bc4a.5602.ac25/0000 EC000025: Successfully sent SECURED status for CA with CKN 9F0DC198.
MKA-EVENT: Successfully updated the CKN handle for interface: TenGigabitEthernet1/0/1 with 9F0DC198 (if
%MKA-5-SESSION_SECURED: (Te1/0/1 : 2) MKA Session was secured for RxSCI bc4a.5602.ac25/0000, AuditSessi
MKA-EVENT: MSK found to be same while updating the MSK and EAP Session ID in the subblock
MKA-EVENT bc4a.5602.ac25/0000 EC000025: After sending SESSION_SECURED status - SECURED=true, PREVIOUSLY
```

### ISE(Identity Services Engine) 문제 해결

이 기능과 관련된 트러블슈팅은 cisco-av-pair 특성 linksec-policy=should-secure의 전달로 제한됩니다.

인증 결과가 디바이스가 연결 중인 스위치 포트에 연결된 Radius 세션에 해당 정보를 전송하는지 확인합니다.

ISE에 대한 추가 인증 분석은 ISE에서 트러블슈팅 및 디버그 활성화를 참조하십시오

## 일반적인 문제

### 암호 불일치

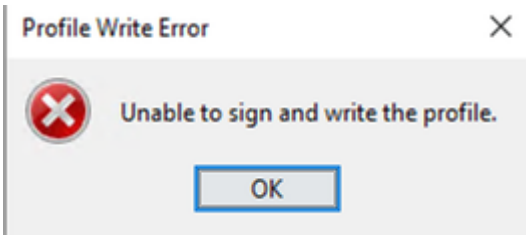
이 로그는 NAD의 MKA 디버그에서 볼 수 있습니다.

```
MKA-4-MKA_MACSEC_CIPHER_MISMATCH: (Te1/0/1 : 30) Lower strength MKA-cipher than macsec-cipher for RxSCI
```

이 시나리오에서 가장 먼저 확인해야 할 사항은 스위치의 MKA 정책 및 보안 클라이언트 프로필에 구성된 암호가 일치한다는 것입니다.

AES-GCM-256 암호화의 경우, 이러한 요구 사항은 (문서에 따라) Cisco Secure Client(AnyConnect 포함) 관리자 가이드, 릴리스 5를 충족해야 합니다

### configuration.xml을 저장할 수 없습니다.




프로파일 쓰기 오류와 관련된 이 문제는 Network Access Manager 프로파일 편집기를 사용하여 MKA의 Setup이라는 configuration.xml을 저장하면 해결됩니다.

사용된 configuration.xml 파일을 수정할 수 없는 오류와 관련이 있습니다. 따라서 다른 위치에 파일을 저장하여 프로파일 교체를 진행하십시오.

## 관련 정보

- MACsec 암호화 구성
- Cat9k 및 ISE를 사용하여 호스트에 MACsec 스위치 구성

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.