Secure Client 5を使用したMKAの設定およびトラブルシューティング

内容

はじめに

このドキュメントでは、サプリカントとしてSecure Client 5を使用して、エンドポイントでMACsec暗号化を設定する方法について説明します。

前提条件

次の項目に関する知識があることが推奨されます。

- Identity Services Engine
- 802.1xおよびRadius
- MACsec MKA暗号化
- Secure Clientバージョン5（旧称Anyconnect）

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づいています。

- Identity Services Engine（ISE）バージョン 3.3
- Catalyst 9300バージョン17.06.05
- Cisco Secureクライアント5.0.4032

背景説明

MACSec(Media Access Control Security)は、802.1AE IEEE規格によって定義された、OSIモデル（データリンク）のレイヤ2でのイーサネットフレームの暗号化と保護を提供するネットワークセキュリティ規格です。

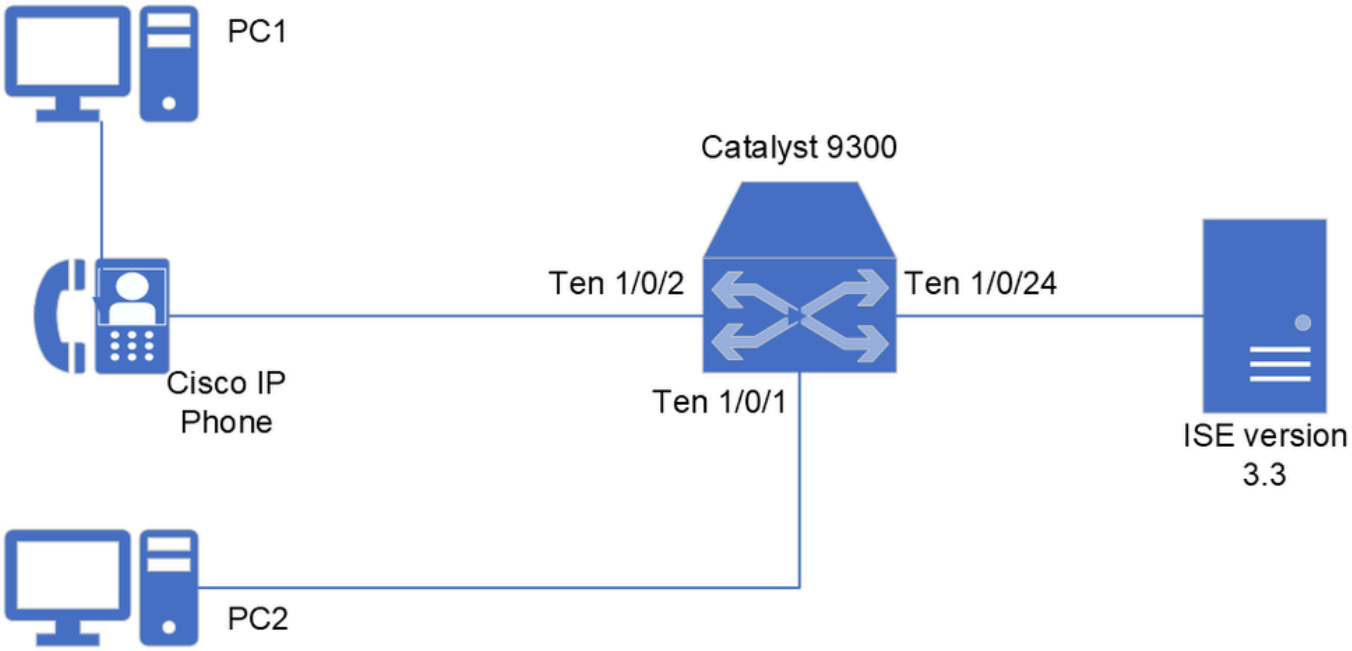
MACSecは、スイッチ間またはスイッチとホスト間の接続を可能にするポイントツーポイント接続でこの暗号化を提供します。したがって、この標準のカバレッジは有線接続に限定されます。

この規格では、レイヤ2接続で送信されるフレームの送信元MACアドレスと宛先MACアドレスを除くデータ全体が暗号化されます。

MACsec Key Agreement(MKA)プロトコルは、MACsecピアがリンクの保護に必要なセキュリティキーをネゴシエートするメカニズムです。

設定

ネットワーク図



注：このドキュメントでは、PCデバイスとCisco IP Phoneに対してRadius認証が設定され、動作しているルールがあると想定しています。最初から設定をセットアップする場合は、『ISEセキュア有線アクセス規範的導入ガイド』を参照して、Identity Services EngineおよびIdentityベースネットワークアクセス用スイッチの設定を確認してください。

ISEでのポリシーの設定

最初の作業は、前の図に示した両方のPC（およびCisco IP Phone）に適用される、対応する認可プロファイルを設定することです。

この架空のシナリオでは、PCは認証方式として802.1Xプロトコルを使用し、Cisco IP PhoneはMACアドレスバイパス(MAB)を使用します。

ISEはRADIUSプロトコルを介してスイッチと通信し、エンドポイントの接続元のインターフェイスでスイッチが適用する必要がある属性についてRadiusセッションを介して通信します。

ホストでのMACsec暗号化の場合、必要な属性はcisco-av-pair = linksec-policyであり、次の3つの値が考えられます。

- Should-not-Secure:スイッチは、RADIUSセッションが発生しているインターフェイスで

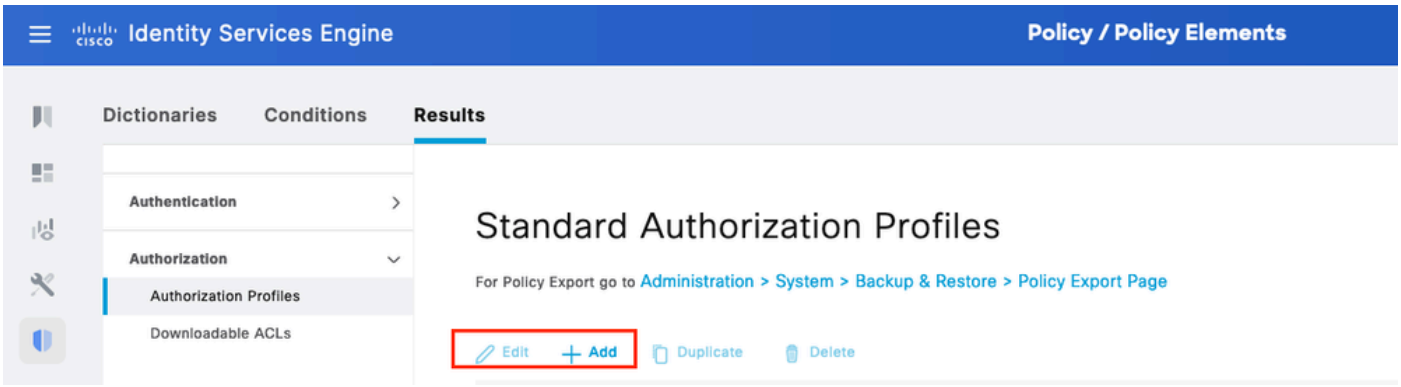
MKA暗号化を実行しません。

- Must-Secure:スイッチは、Radiusセッションにリンクされているトラフィックで暗号化を適用する必要があります。MKAセッションが失敗した（またはタイムアウトした）場合、接続は認証失敗と見なされ、MKAセッションの確立が再度試行されます。
- Should-Secure:スイッチはMKA暗号化を実行しようとします。Radiusセッションにリンクされ たMKAセッションが成功すると、トラフィックは暗号化されます。MKAに障害が発生したり、タイムアウトになったりした場合、スイッチはそのRadiusセッションにリンクされている暗号化されていないトラフィックを許可します。

ステップ 1： 両方のPCで、MKA機能を持たないマシンがインターフェイスTen 1/0/1に接続する場合に柔軟に対応できるように、should-secure MKAポリシーを適用します。

オプションとして、必須セキュリティポリシーを適用するPC2のポリシーを設定できます。

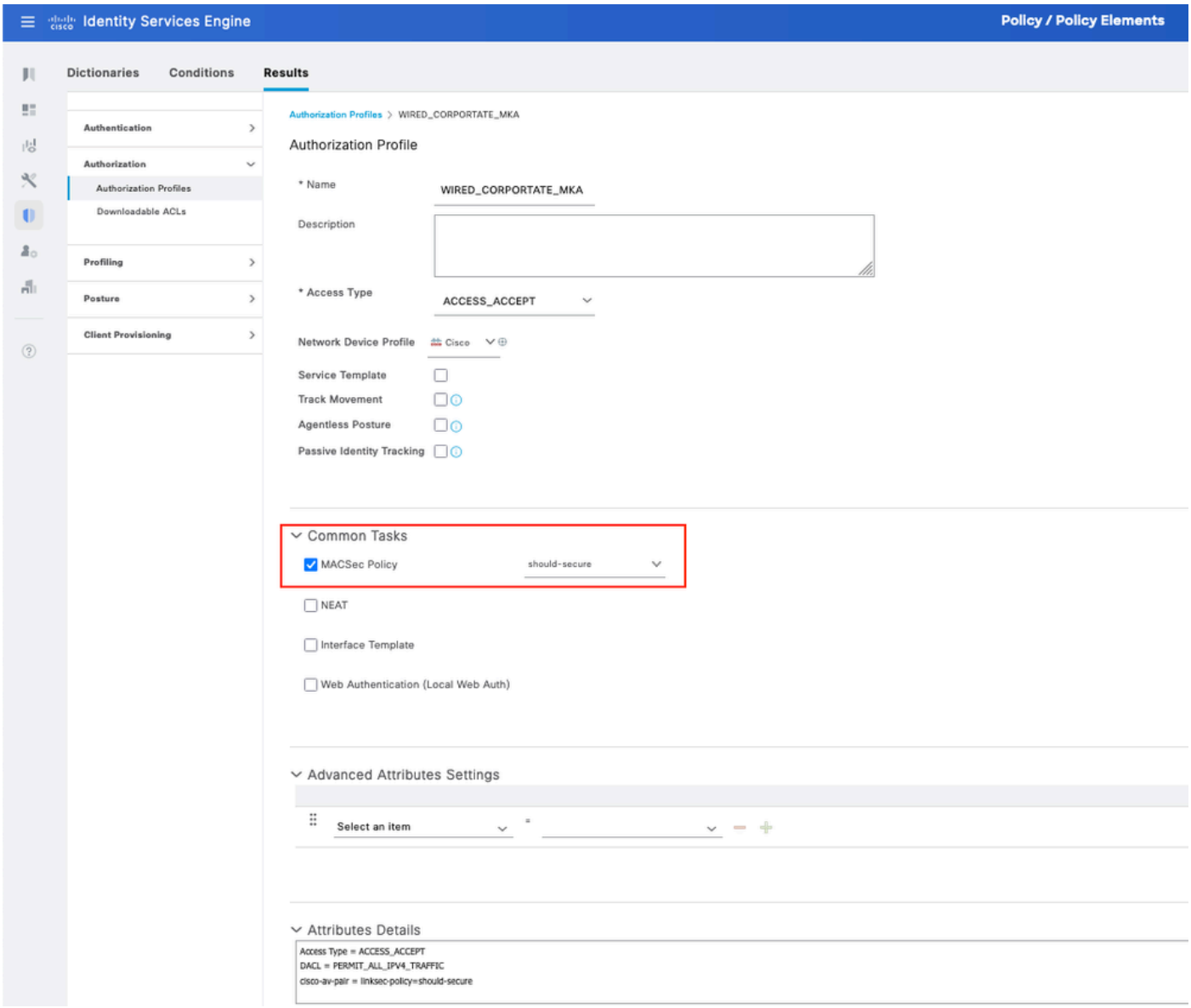
この例では、Policy > Policy Elements > Results > Authorization Profilesの順に選択し、+AddまたはEditで既存のプロファイルを編集して、PCのポリシーを設定します



ステップ 2 プロファイルに必要なフィールドを入力するか、カスタマイズします。

共通タスクで、MACSecポリシーと適用する対応するポリシーを選択していることを確認します。

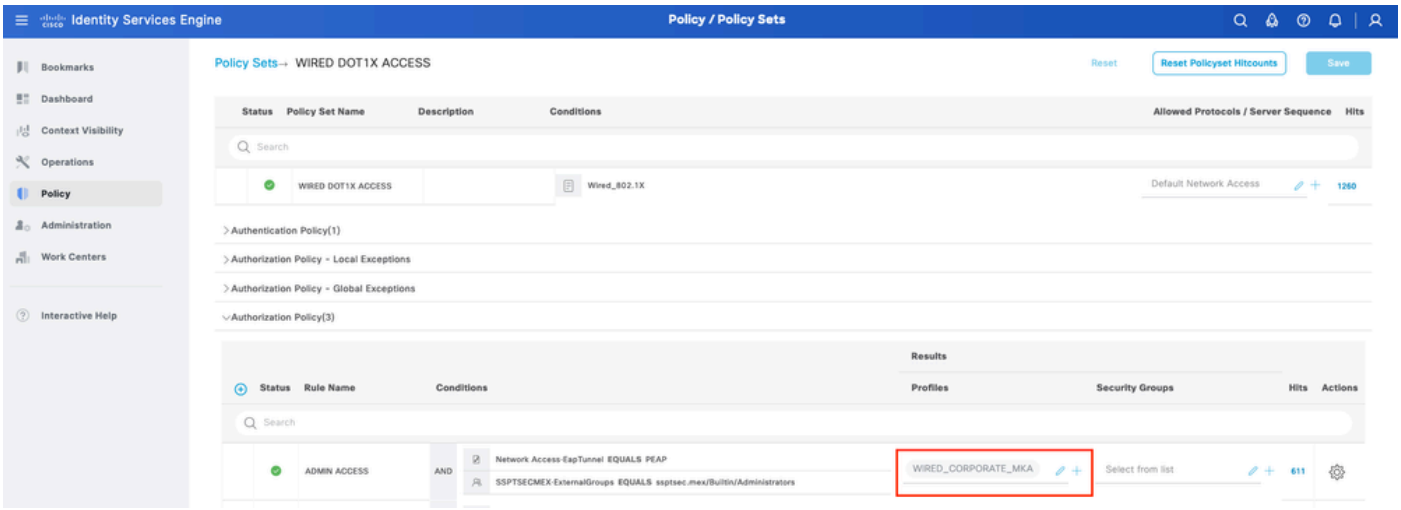
下にスクロールして、設定を保存します。



ステップ 3対応する認可プロファイルを、デバイスがヒットする認可ルールに割り当てます。

Policy > Policy Sets > (Select Policy Set assigned) > Authorization Policyの順に移動します。

認可ルールをMACsec設定の認可プロファイルに関連付けます。下にスクロールして、設定を保存します。



Catalyst 9300でのMKAの設定

ステップ 1： 次の例に示すように、新しいMKAポリシーを設定します。

```
!
  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
!
```

ステップ 2 PCが接続されているインターフェイスでMACsec暗号化を有効にします。

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

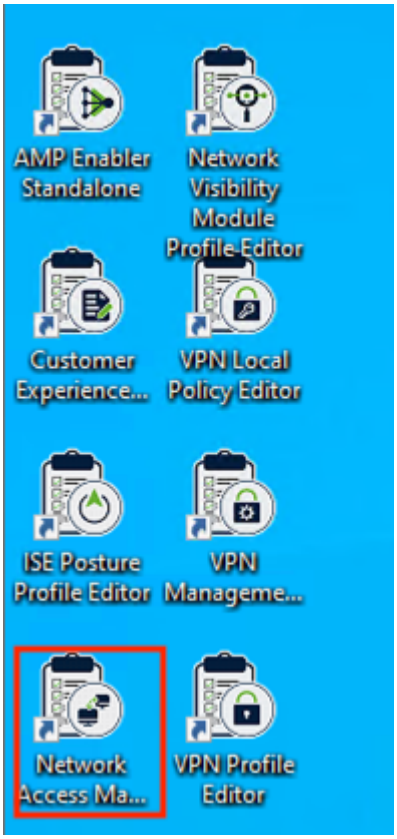
注:MKA設定のコマンドとオプションの詳細については、使用しているスイッチのバージョンに対応する『Security configuration guide』を参照してください。この例のシナリオでは、『セキュリティ設定ガイド、Cisco IOS XE Bengaluru 17.6.x（Catalyst 9300スイ

ッチ）

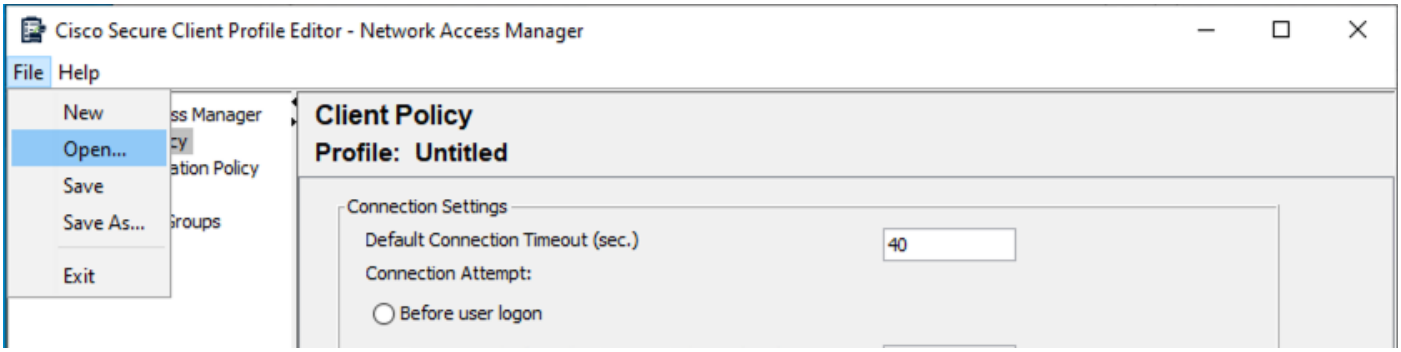
Network Access Manager Profile Editorを使用したMKAのセットアップ

ステップ 1：（シスコのダウンロードサイトから）ダウンロードし、使用しているセキュアクライアントのバージョンに一致するプロファイルエディタを開きます。

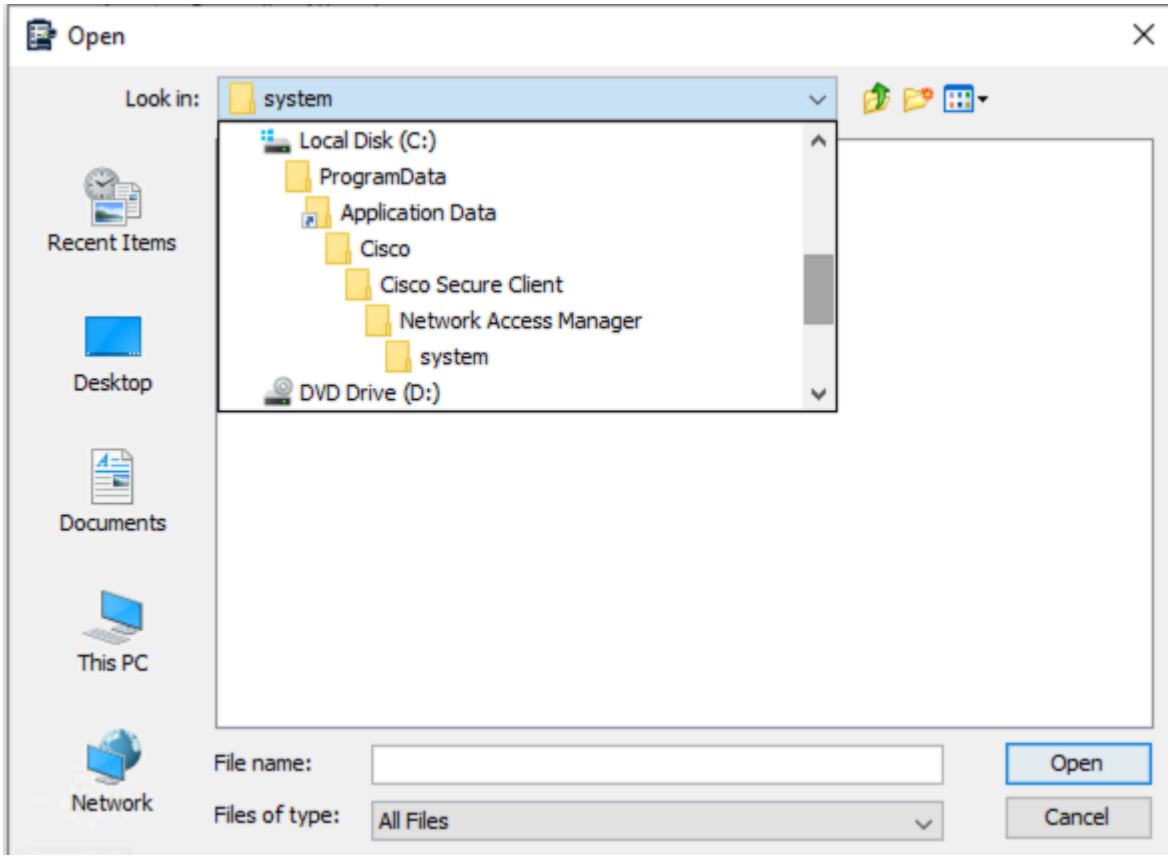
このプログラムをコンピュータにインストールしたら、Cisco Secure Client Profile Editor - Network Access Managerを開きます。



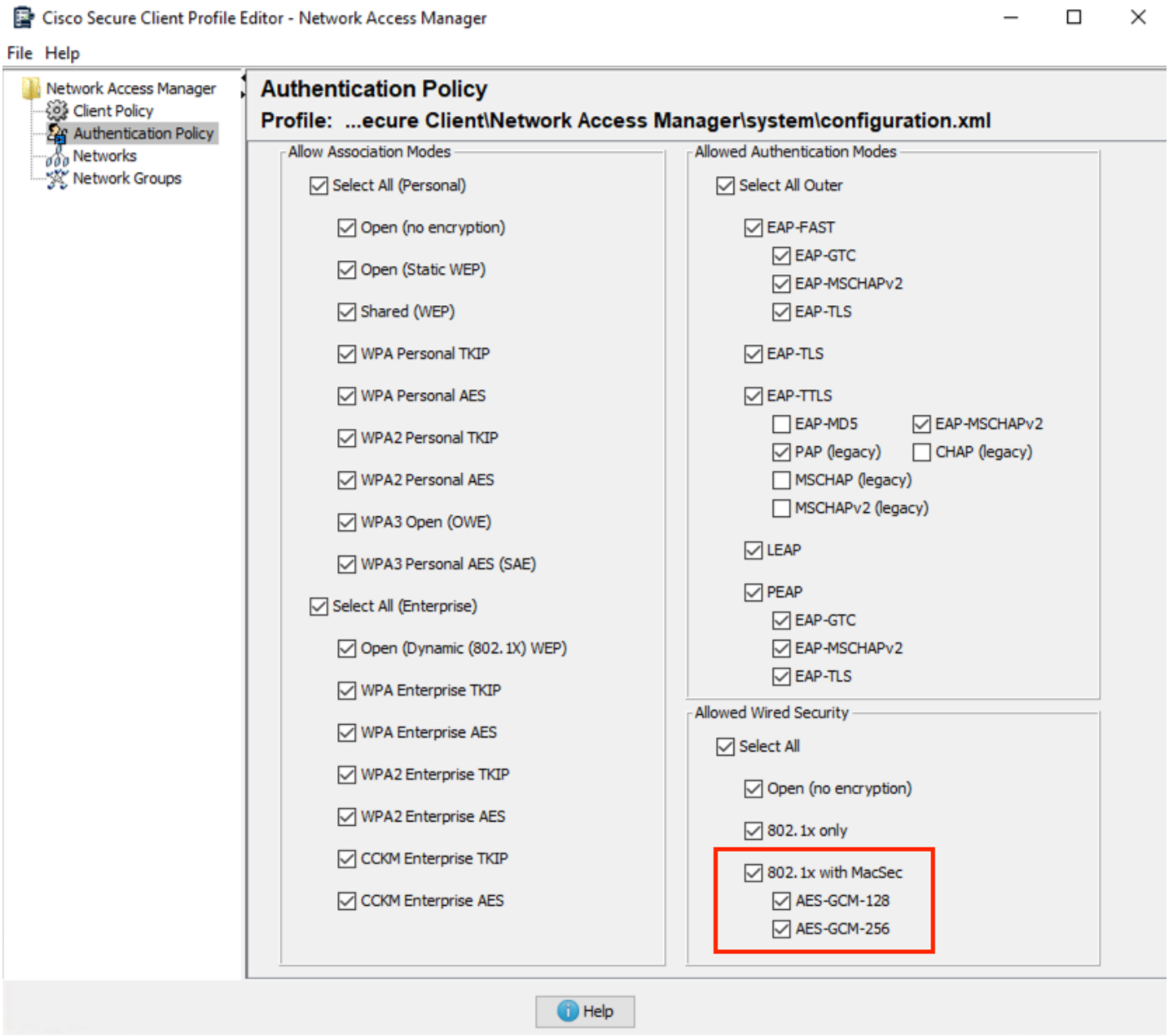
ステップ 2 File > Openの順に選択します。



ステップ 3 このイメージに表示されているフォルダシステムを選択します。このフォルダ内で、configuration.xmlという名前のファイルを開きます。

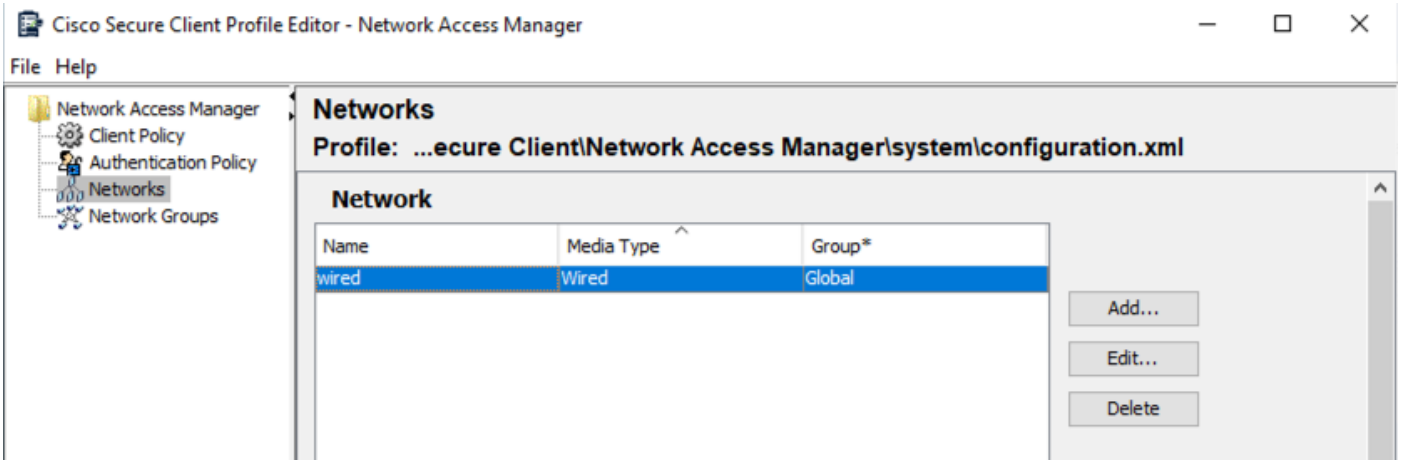


ステップ 4 プロファイルエディタでファイルがロードされたら、Authentication Policyオプションを選択し、802.1x with MACSecに関連するオプションが有効になっていることを確認します。



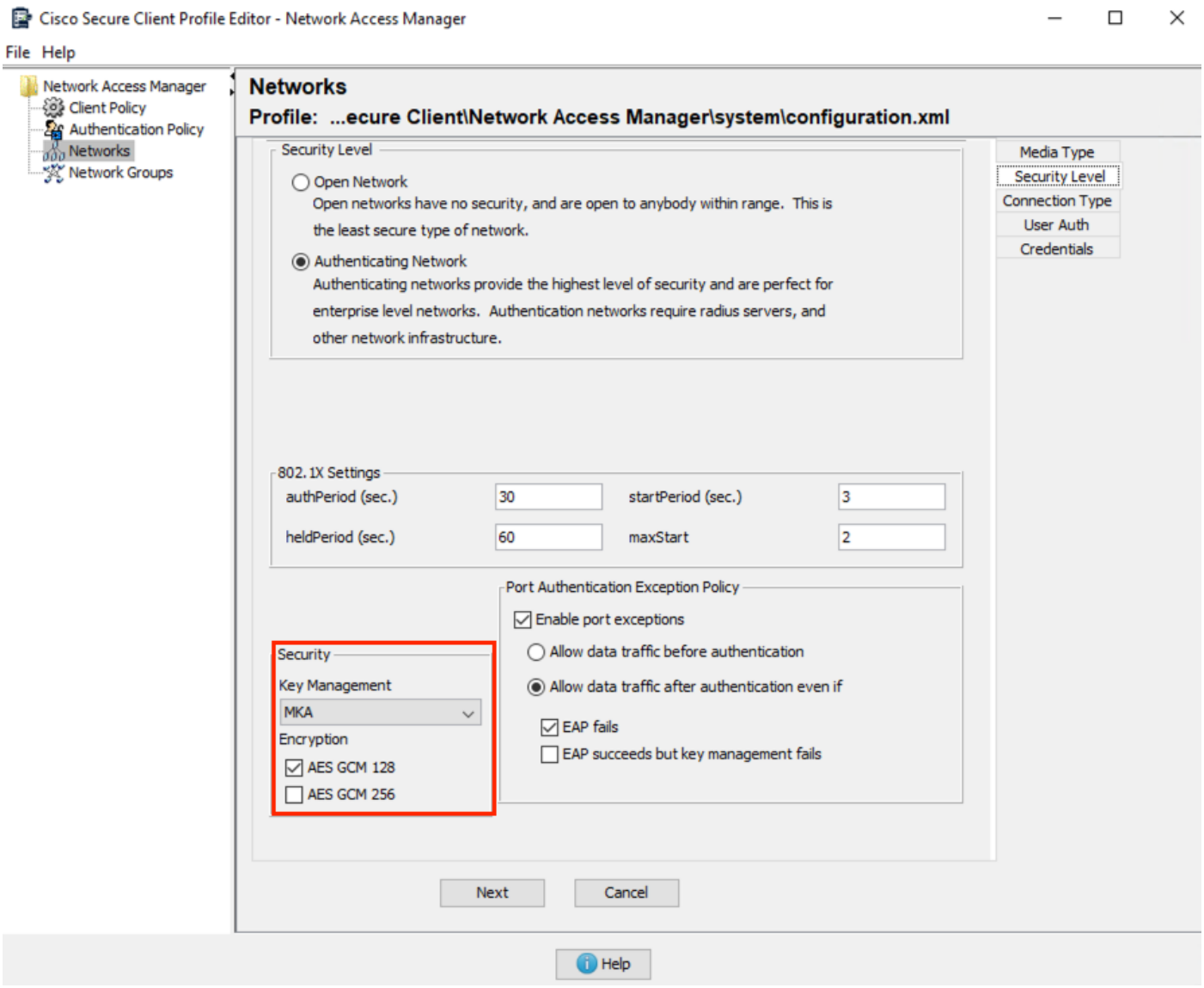
ステップ 5 「ネットワーク」の項に進んでください。ここでは、有線接続用の新しいプロファイルを追加するか、Secure Client 5.0とともにインストールされるデフォルトの有線プロファイルを編集します。

このシナリオでは、既存の有線プロファイルを編集します。



ステップ 6 プロファイルを設定します。セクション「セキュリティレベル」で、MKAに続けて暗号化AES GCM 128を使用するようにキー管理を調整します。

認証dot1xとポリシーのその他のパラメータも調整します。

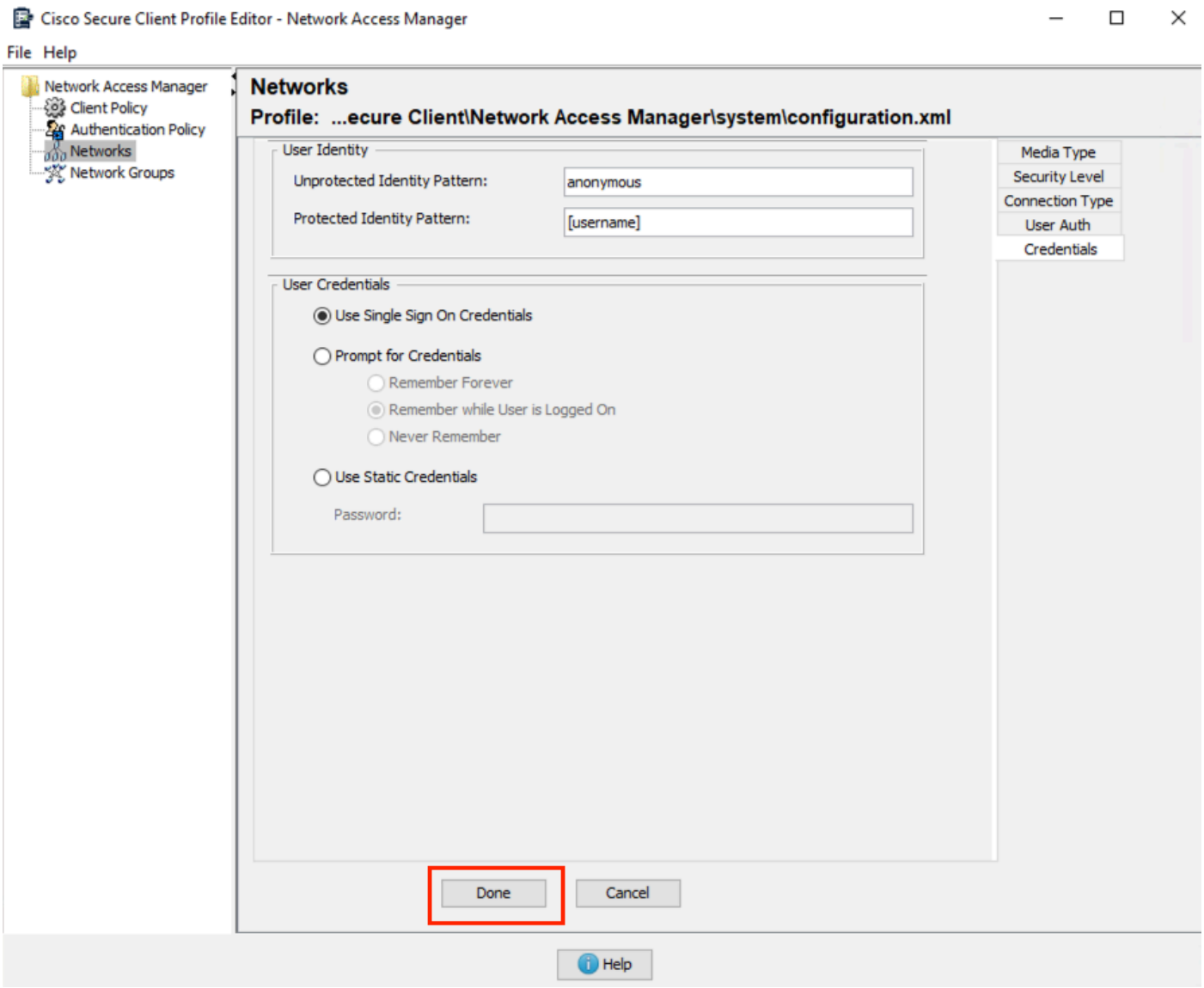


ステップ 7 接続タイプ、ユーザ認証、およびクレデンシャルに関する残りのセクションを設定します。

これらのセクションは、「セキュリティレベル」セクションで選択した認証設定によって異なります。

設定を終了したら、Doneをクリックします。

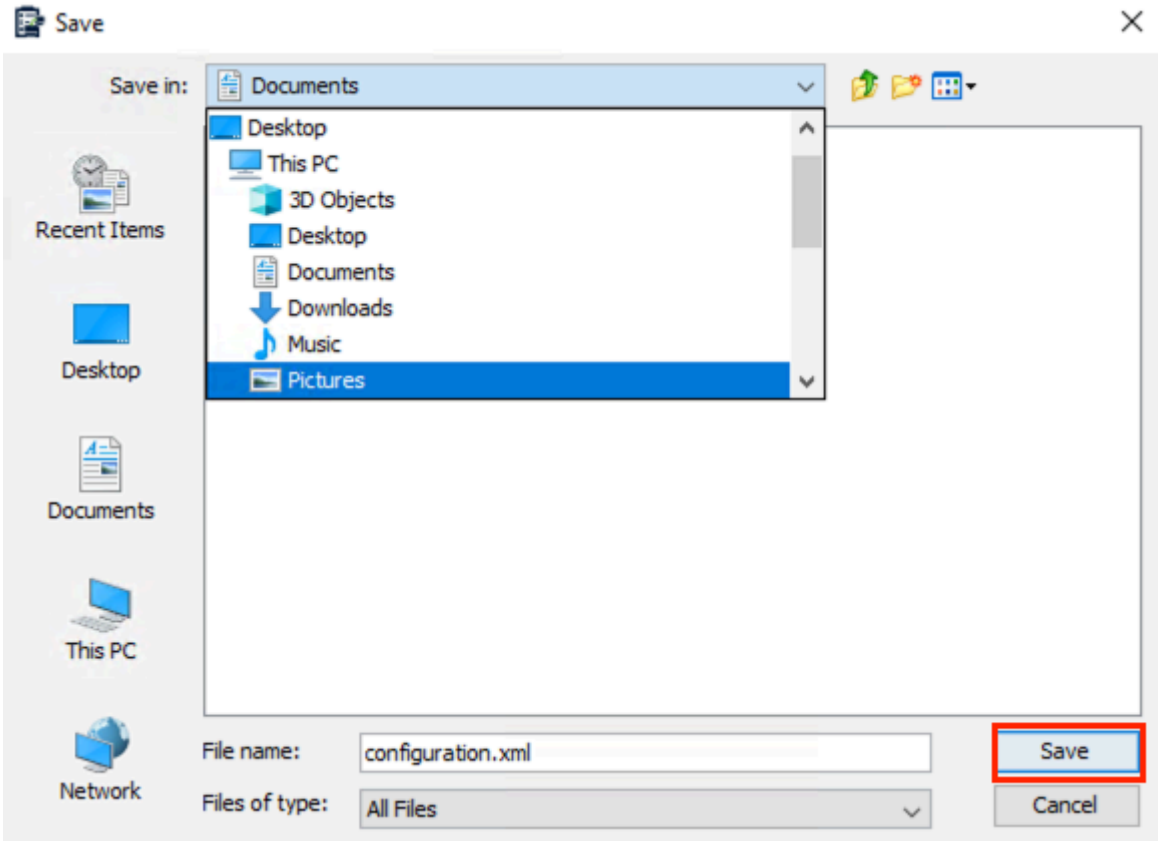
このシナリオでは、ユーザクレデンシャルでProtected Extensible Authentication Protocol(PEAP)を使用しています。



ステップ 8 メニューFileに移動します。「名前を付けて保存」オプションを続行します。

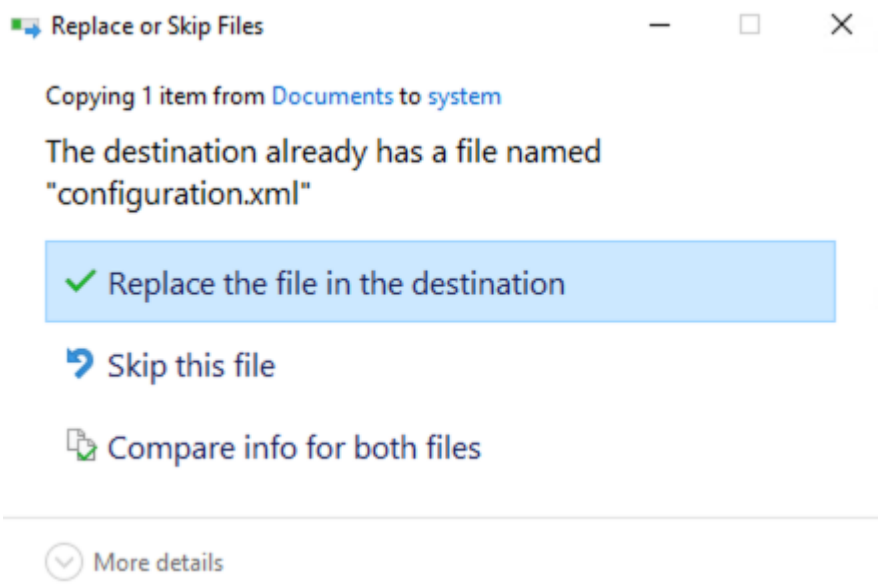
ファイルにconfiguration.xmlという名前を付け、ProgramData\Cisco\Cisco Secure Client\Network Access Manager\systemとは別のフォルダに保存します。

この例では、ファイルはDocumentsフォルダに保存されています。プロファイルを保存します。



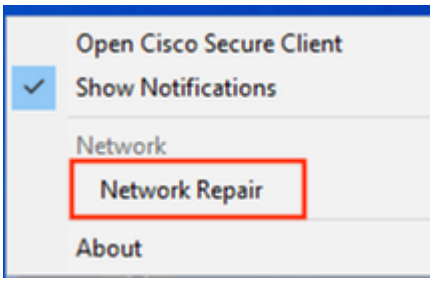
ステップ 8プロファイルの場所に進み、ファイルをコピーして、ProgramData\Cisco\Cisco Secure Client\Network Access Manager\systemフォルダに格納されているファイルを置き換えます。

Replace the file in destinationオプションを選択します。



ステップ 9 Security Client 5.0で変更されたプロファイルをロードするには、Windowsマシンの右下のタスクバーにあるSecure Clientアイコンを右クリックします。

ネットワーク修復を実行します。

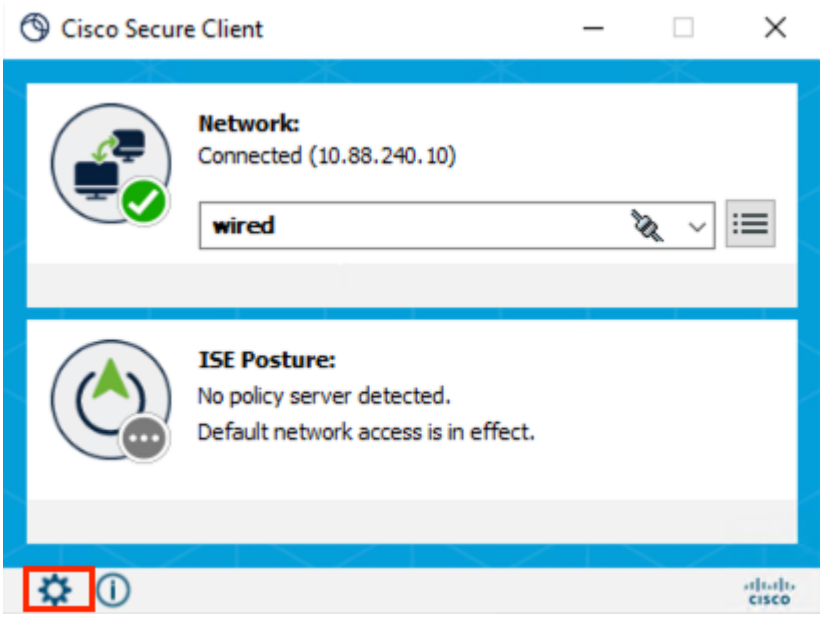


注：プロファイルエディタで設定されたすべてのネットワークにはAdministrator Networkの権限が与えられるため、ユーザはこのツールを使用して設定したコンテンツをカスタマイズまたは変更できません。

Network Access Manager(NAM)を使用したMKAネットワークのセットアップ（オプション）

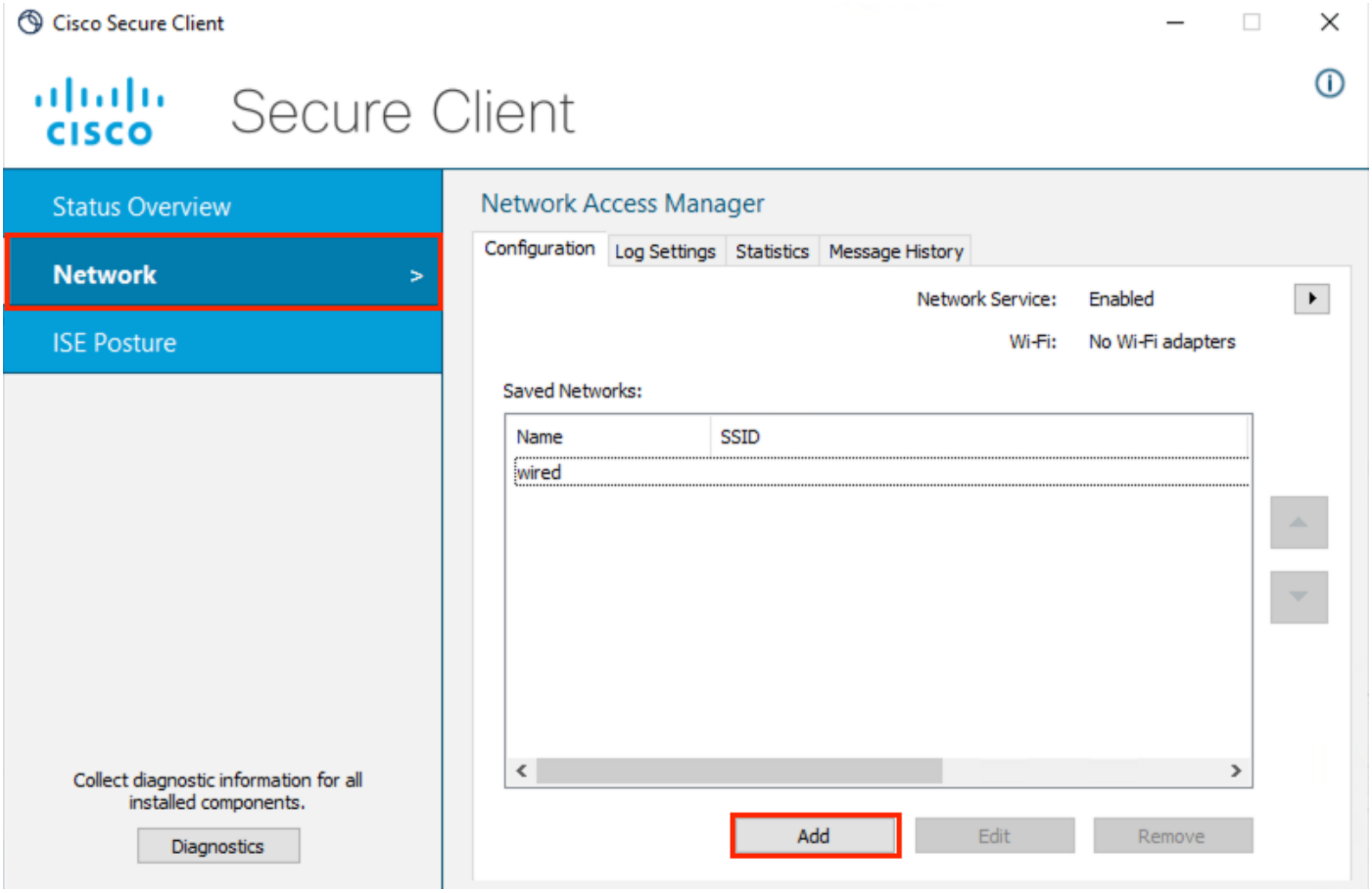
ステップ 1：プロファイルエディタを使用してMKAを設定する代わりに、このツールを使用せずにネットワークを追加できます。

Secure Clientスイートから、歯車アイコンを選択します。



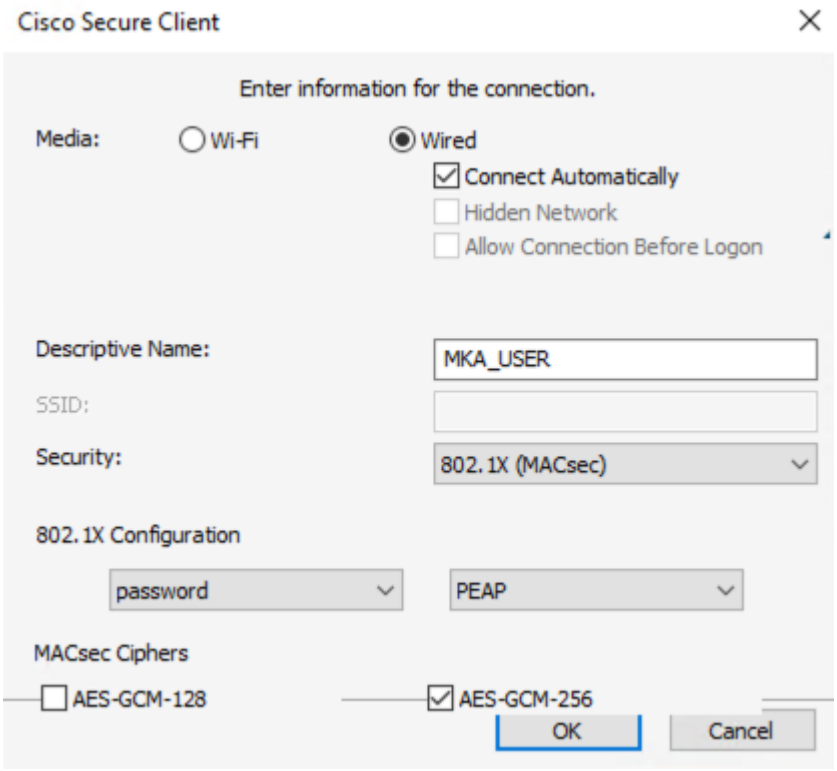
ステップ 2表示された新しいウィンドウで、オプションNetworkを選択します。

Configurationセクションで、Addオプションを選択し、特権User Networkを持つネットワークMKA対応のIPアドレスを入力します。



ステップ 3新しい設定ウィンドウで、接続の特性を設定し、ネットワークに名前を付けます。

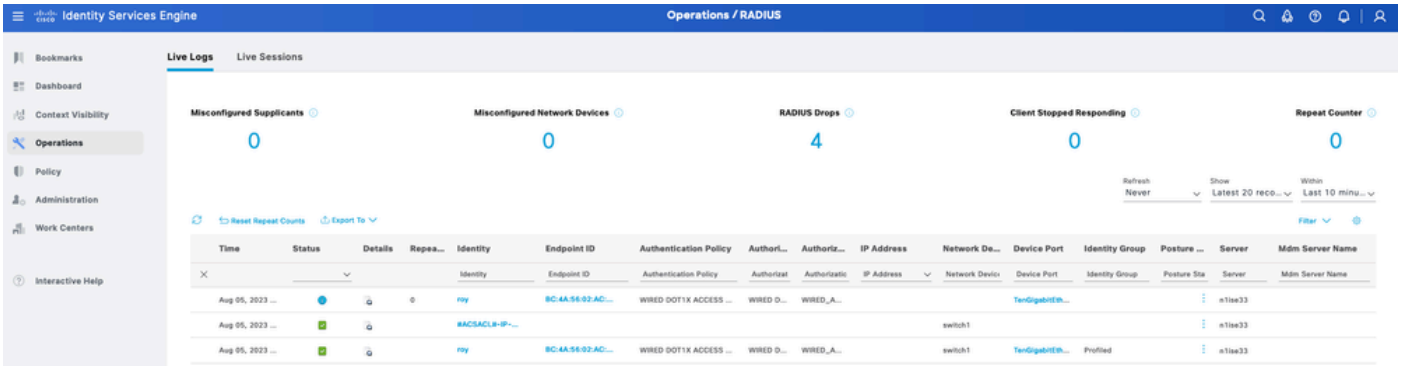
終了したら、OKボタンを選択します。



確認

ISEでの検証

ISEでは、このフローの設定が完了した時点で、Livelogsでデバイスが認証および認可されていることに注意してください。



認証の詳細と結果セクションに移動します。

許可プロファイルに設定された属性は、ネットワークアクセスデバイス(NAD)と、1つのEssentialライセンスの消費に送信されます。

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スイッチの検証。

これらのコマンドは、このソリューションの適切な機能を検証するために使用できます。

```
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
```

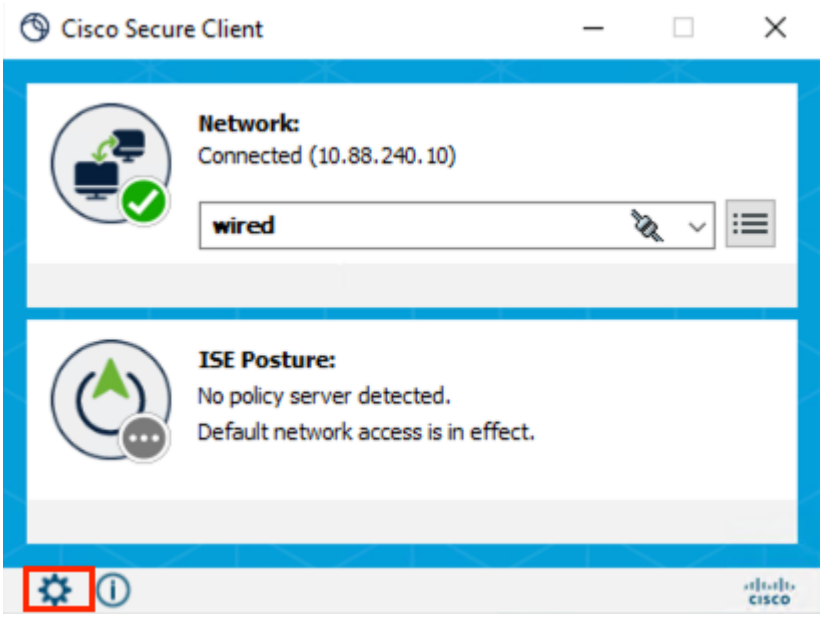
```
switch1#show mka session

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
```

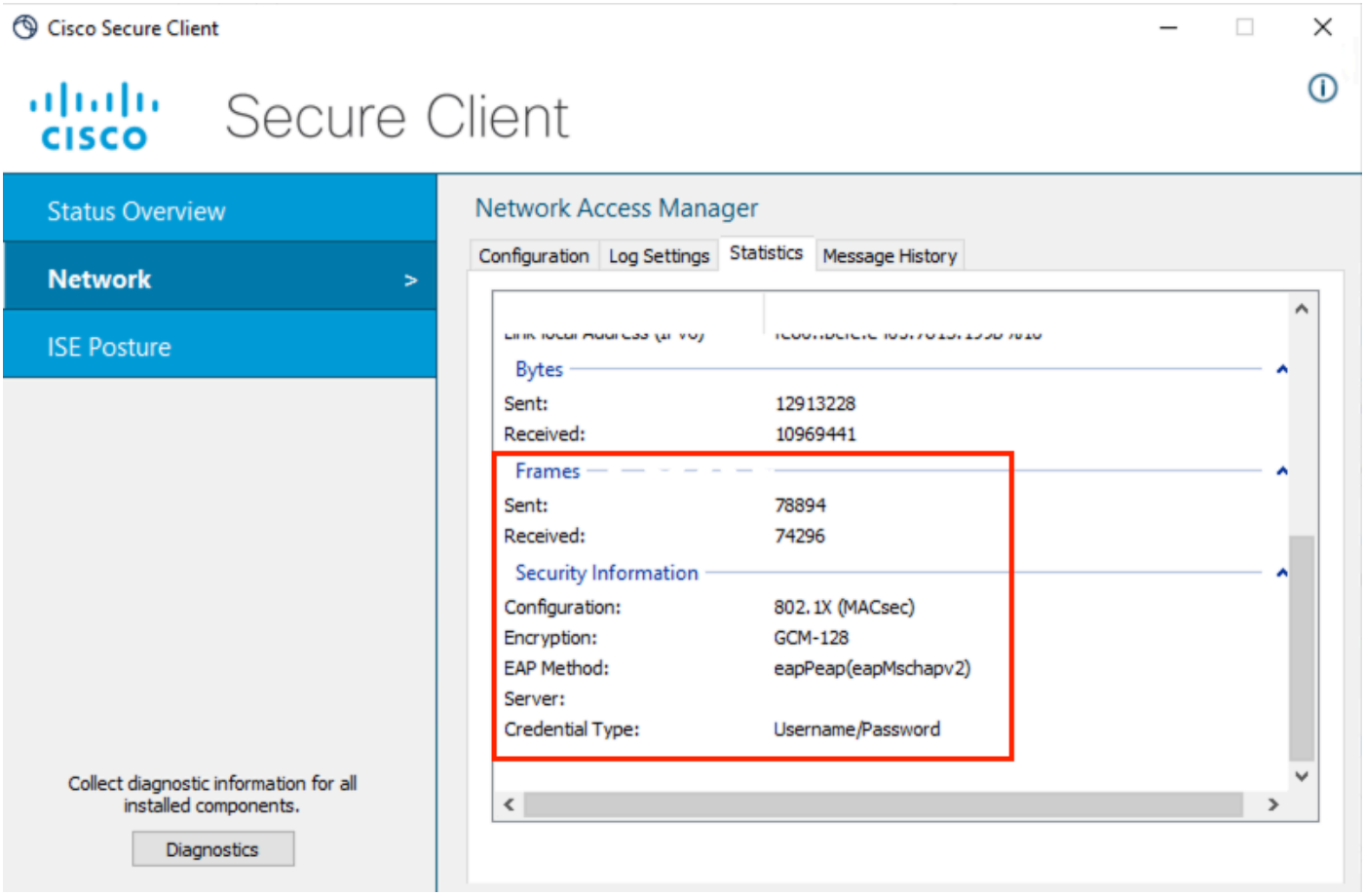
セキュアクライアントの検証。

MACsec暗号化で作成されたプロファイルを使用すると、認証が成功します。エンジンアイコンをクリックすると、詳細情報が表示されます。



ここで表示されるセキュアクライアントのメニューのNetwork Access Manager > Statisticsの項で、暗号化と対応するMACsec設定に注目してください。

送受信されたフレームは、レイヤ2で暗号化が実行されるほど増加します。

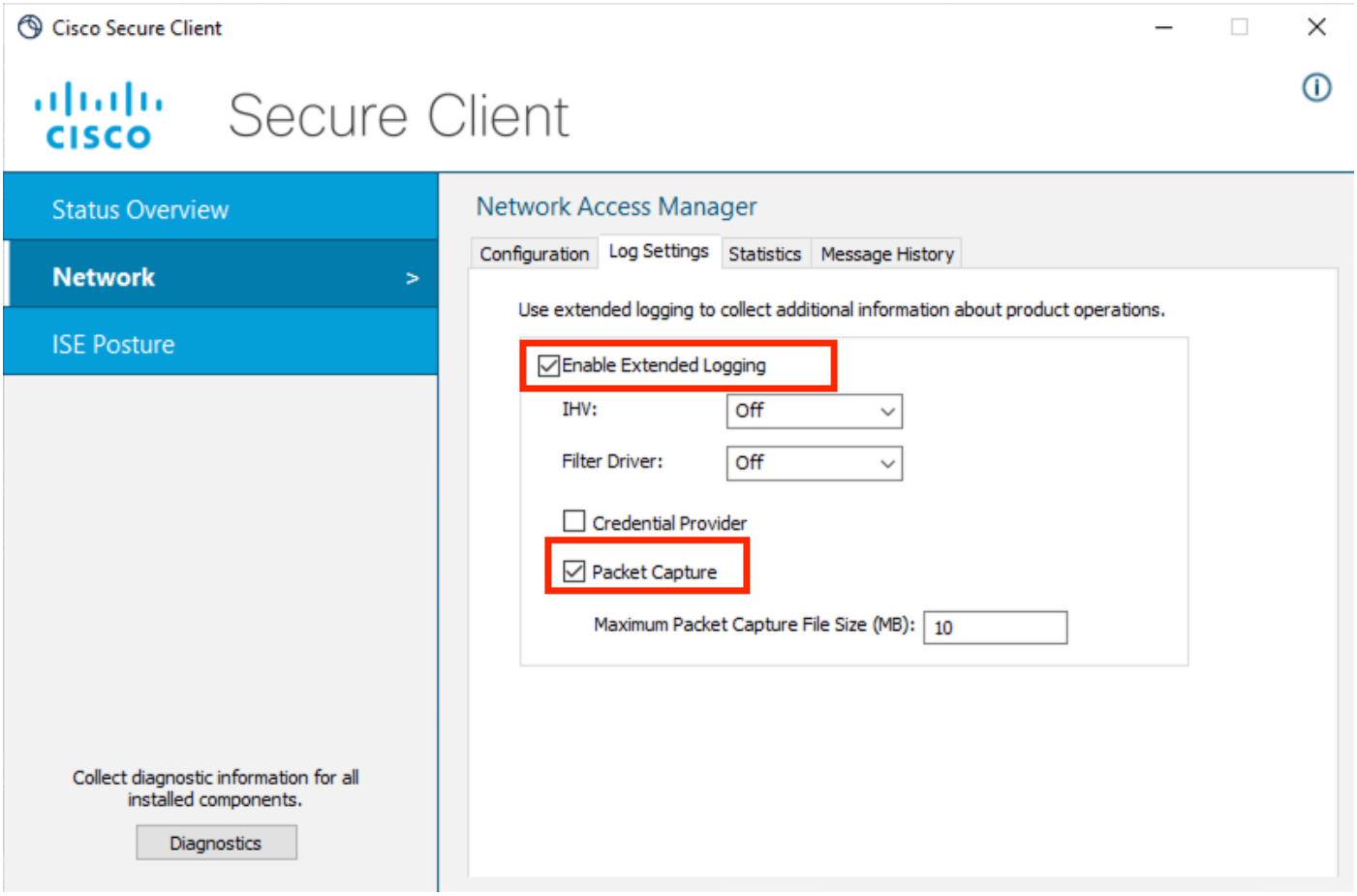


トラブルシュート

注：このセクションでは、発生する可能性があるMKAの問題に関連するトラブルシューティングの部分について説明します。認証または認可の失敗が発生した場合は、『ISEセキュア有線アクセス規範的導入ガイド – トラブルシューティング』を参照して調査してください。このガイドでは、MACsec暗号化を使用しなくても認証が正常に動作することを前提としています。

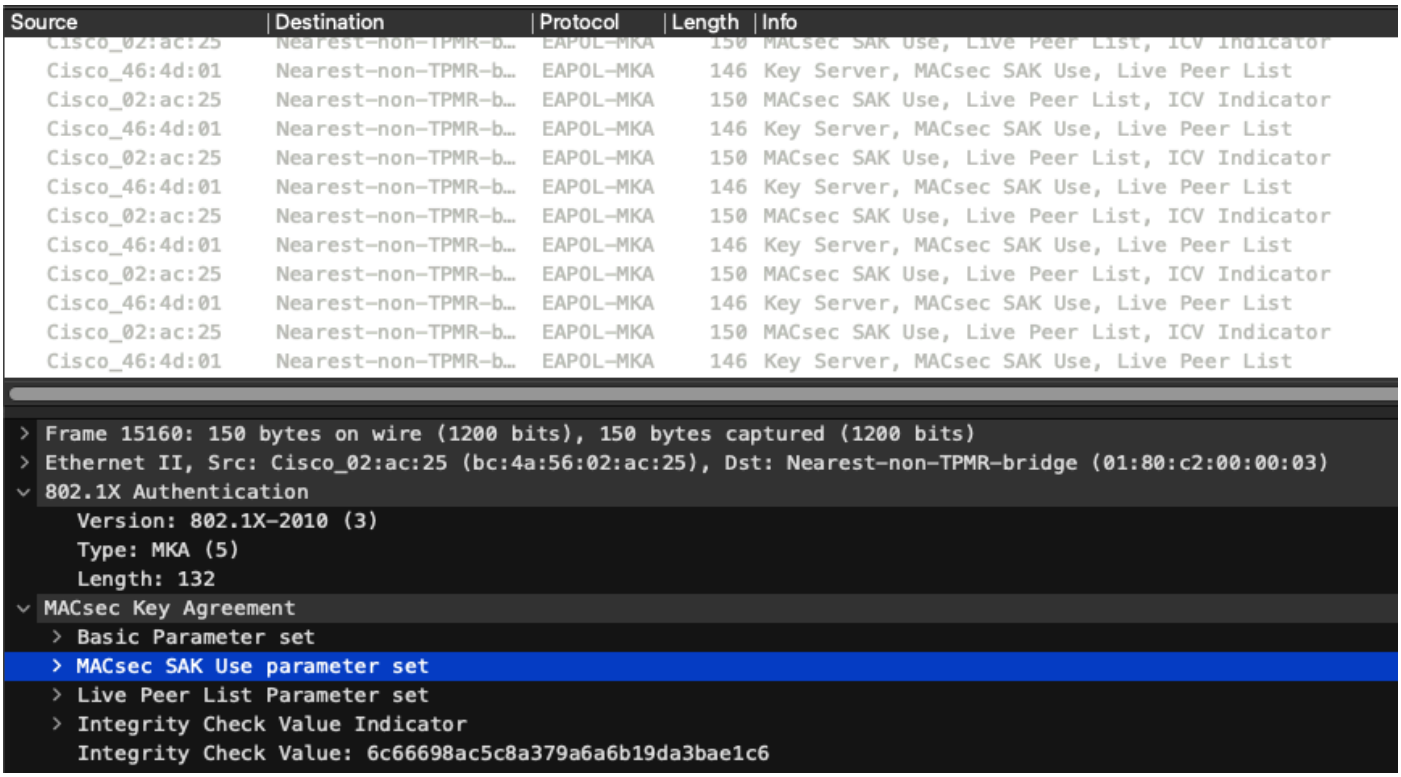
Cisco Secure Endpoint

- Secure Endpoint SuiteでDARTモジュールを有効にします。
- このメニューで、拡張ロギングを有効にして、ユーザMKA接続に関する詳細なデータを収集します。また、DARTバンドルに含まれているパケットキャプチャを有効にすることもできます。



- DARTバンドルを収集して、configuration.xmlとNetwork Access Managerの分析を続行します。ドキュメント『トラブルシューティングのためのデータ収集のためのDARTの実行』を参照してください。

次の例は、ホストとスイッチの間の情報が暗号化されているため、パケットがどのように見えるかを示しています（暗号化は暗号化されていません）。



DARTバンドルからは、NetworkAccessManager.txtという名前のログに、802.1X認証とMKAセッションに関する有益な情報が含まれています。

この情報は、MKA暗号化を使用した認証が成功すると表示されます。

```
%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のトラブルシューティング

これらのコマンドをネットワークアクセスデバイス(NAD)に実装して、プラットフォームとサプリカント間のMKA暗号化を確認できます。

コマンドの詳細については、NADとして使用されるプラットフォームの対応するコンフィギュレーションガイドを確認してください。

```
#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
```

これらは、ホストへの1つの正常なMKA接続のデバッグです。次の情報を参照用として使用できます（図1を参照）。

```
%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_
```

Identity Services Engine(ISE)のトラブルシューティング

この機能に関連するトラブルシューティングは、cisco-av-pair属性linksec-policy=should-secureの配信に限定されます。

デバイスが接続されているスイッチポートにリンクされているRADIUSセッションに許可結果がその情報を送信していることを確認します。

ISEでの認証分析の詳細については、「ISEでのデバッグのトラブルシューティングと有効化」を参照してください。

一般的な問題

暗号の不一致

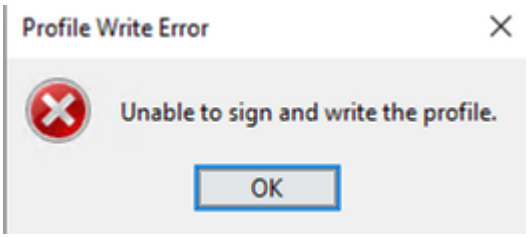
この口グは、NADのMKAデバッグで確認できます。

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

このシナリオで最初に確認するのは、スイッチのMKAポリシーとセキュアクライアントプロファイルで設定されている暗号が一致していることです。

AES-GCM-256暗号化の場合は、次の要件を（ドキュメントごとに）満たす必要があります。『Cisco Secure Client（AnyConnectを含む）管理者ガイドリリース5』

configuration.xmlを保存できません。



プロファイル書き込みエラーに関するこの問題は、Network Access Managerプロファイルエディタを使用して、「MKAの設定」という名前のconfiguration.xml（前述）を保存することで解決できます。

エラーは、使用されているファイルconfiguration.xmlを変更できないことに関連しています。したがって、プロファイルの置き換えを続行するには、ファイルを別の場所に保存してください。

関連情報

- MACsec暗号化の設定
- Cat9kおよびISEを使用したホストへのMACsecスイッチの設定

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。