Configuration et dépannage de MKA à l'aide de Secure Client 5

Table des matières

- Introduction
  - Conditions préalables
  - Composants utilisés
  - Informations générales
- Configurer
  - Diagramme du réseau
  - Configuration des stratégies sur ISE
  - Configuration de MKA dans Catalyst 9300
  - Configuration de MKA à l'aide de l'Éditeur de profil Network Access Manager.
  - Configuration des réseaux MKA à l'aide du Gestionnaire d'accès réseau (facultatif).
- Vérifier
  - Validation sur ISE
  - Validation sur le commutateur Catalyst.
  - Validation sur le client sécurisé.
- Dépannage
  - Terminaux sécurisés Cisco
  - Dépannage de Cisco IOS
  - Dépannage d'Identity Services Engine (ISE)
- Problèmes courants
  - Non-concordance de chiffrement
  - Impossible d'enregistrer le fichier configuration.xml.
- Informations connexes

Introduction

Ce document décrit comment configurer le chiffrement MACsec dans un terminal utilisant le client sécurisé 5 comme demandeur.

Conditions préalables

Cisco recommande des connaissances sur les sujets suivants :

- Plateforme de services d’identité
- 802.1x et Radius
- Chiffrement MACsec MKA
- Secure Client version 5 (anciennement Anyconnect)

Composants utilisés

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

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

Informations générales

MACSec (Media Access Control Security) est une norme de sécurité réseau qui assure le cryptage et la protection des trames Ethernet au niveau de la couche 2 du modèle OSI (liaison de données), défini par la norme IEEE 802.1AE.

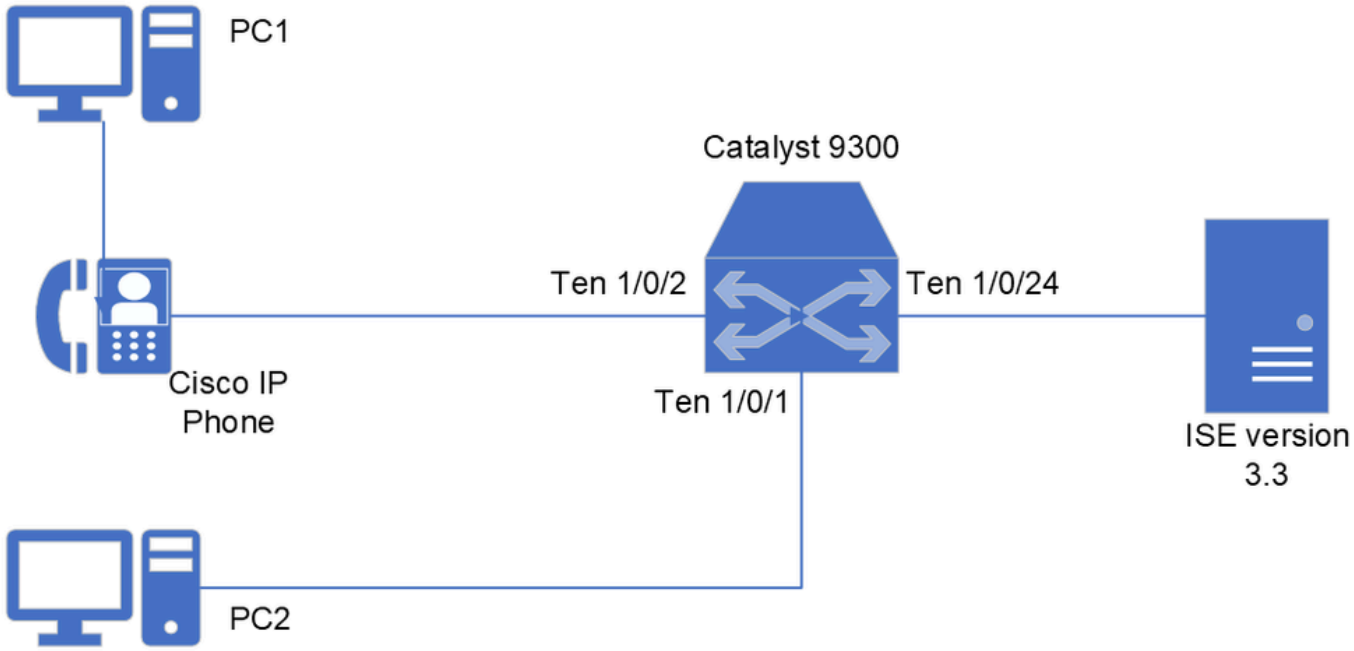
MACSec fournit ce chiffrement dans une connexion point à point qui peut être une connexion commutateur à commutateur ou commutateur à hôte, par conséquent la couverture de cette norme est limitée aux connexions filaires.

Cette norme chiffre l’ensemble des données, à l’exception des adresses MAC source et de destination des trames transmises dans une connexion de couche 2.

Le protocole MACsec Key Agreement (MKA) est le mécanisme à partir duquel les homologues MACsec vont négocier les clés de sécurité nécessaires pour sécuriser la liaison.

Configurer

Diagramme du réseau



Remarque : Cette documentation considère que vous avez des règles configurées et fonctionnant pour l'authentification Radius pour les périphériques PC et le téléphone IP Cisco. Pour configurer une configuration à partir de zéro, veuillez vous reporter au Guide de déploiement prescriptif d'accès filaire sécurisé ISE pour examiner la configuration dans Identity Services Engine and Switch for Identity-Based Network Access.

Configuration des stratégies sur ISE

La première tâche consiste à configurer les profils d'autorisation correspondants qui sont appliqués aux deux PC (ainsi qu'au téléphone IP Cisco) affichés dans le schéma précédent.

Dans ce scénario hypothétique, les PC vont utiliser le protocole 802.1X comme méthode d'authentification et le téléphone IP Cisco utilise le contournement d'adresse MAC (MAB).

ISE communique avec le commutateur via le protocole Radius sur les attributs que le commutateur doit appliquer dans l'interface à partir de laquelle le point d'extrémité est connecté via une session Radius.

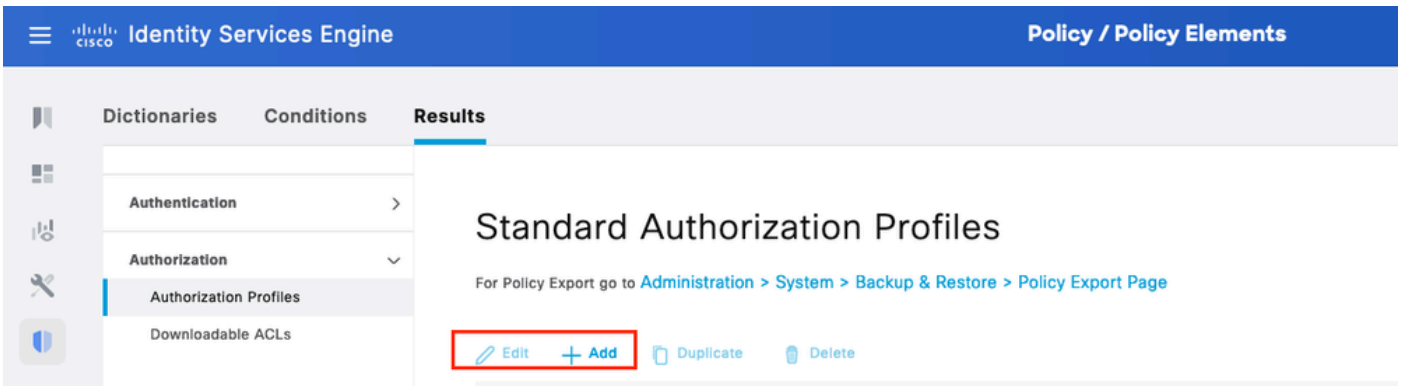
Pour le chiffrement MACsec dans les hôtes, l'attribut requis est cisco-av-pair = linksec-policy, qui a ces 3 valeurs possibles :

- Ne doit pas sécuriser : Le commutateur n'effectue pas le chiffrement MKA dans l'interface où la session Radius a lieu.
- Doit-être sécurisé : Le commutateur doit appliquer le chiffrement dans le trafic lié à la session Radius. Si la session MKA échoue (ou a un délai d'attente), la connexion est considérée comme un échec d'autorisation et une nouvelle tentative est effectuée pour établir la session MKA.
- À sécuriser : Le commutateur tente d'effectuer le chiffrement MKA. Si la session MKA liée à la session Radius réussit, le trafic est chiffré. Si le MKA échoue ou expire, le commutateur autorise ce trafic non chiffré lié à cette session Radius.

Étape 1. Dans les deux PC, appliquez une stratégie MKA sécurisée pour avoir de la flexibilité dans le cas où une machine sans capacités MKA se connecte à l'interface Ten 1/0/1.

En option, vous pouvez configurer une stratégie pour PC2 qui applique une stratégie de sécurité obligatoire.

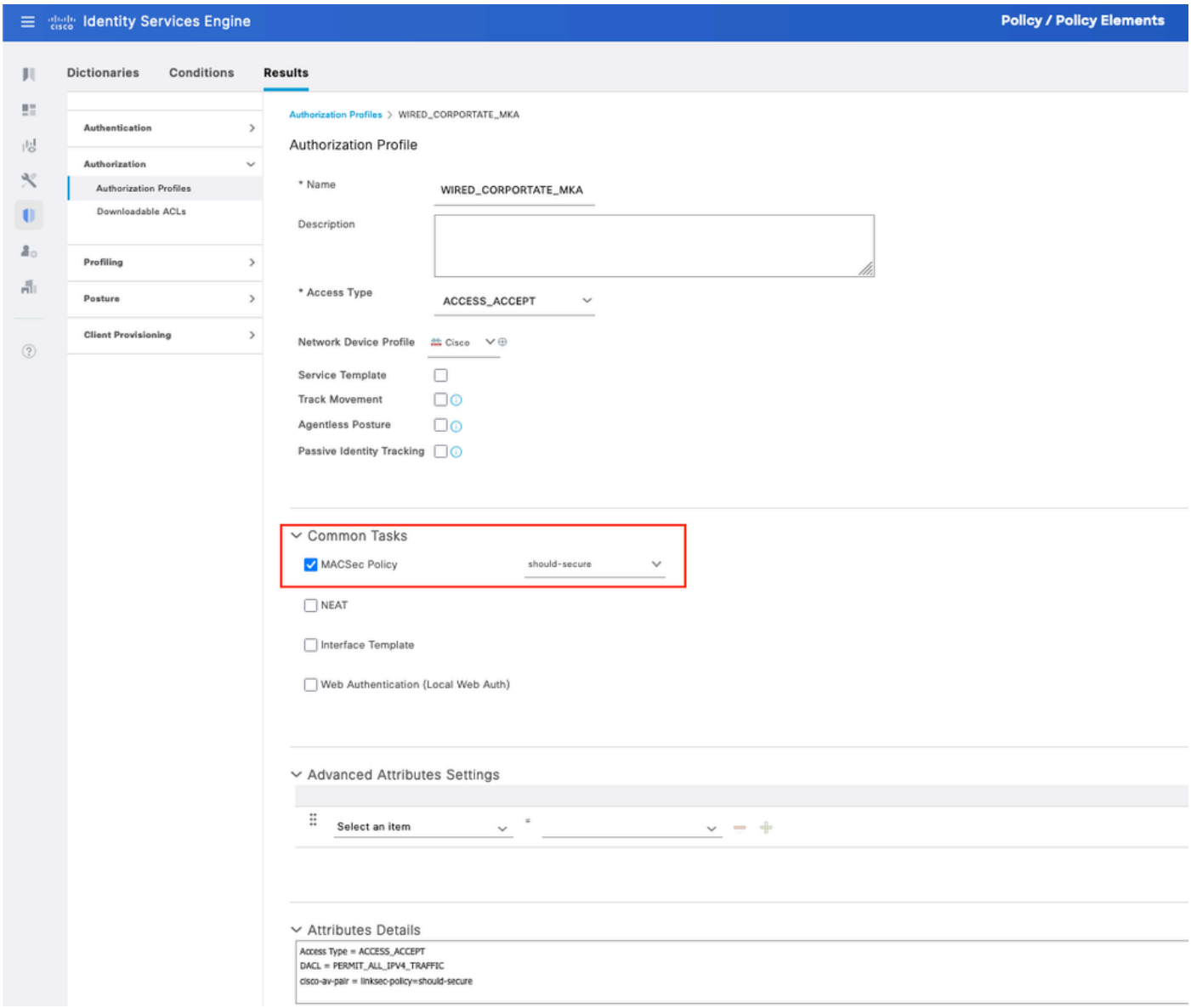
Dans cet exemple, configurez la stratégie pour les PC comme dans Policy > Policy Elements > Results > Authorization Profiles puis +Add or Edit a existing profile



Étape 2. Renseignez ou personnalisez les champs requis pour le profil.

Assurez-vous que dans Tâches communes vous avez sélectionné Stratégie MACSec et la stratégie correspondante à appliquer.

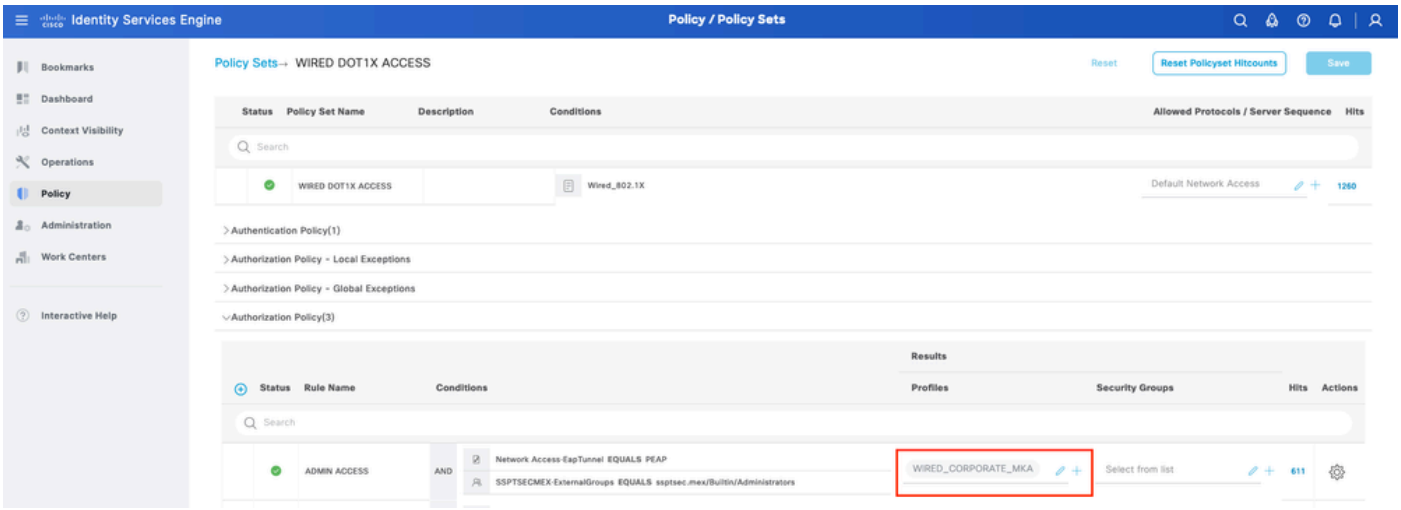
Faites défiler vers le bas et enregistrez la configuration.



Étape 3. Attribuez le profil d'autorisation correspondant aux règles d'autorisation qui sont atteintes par les périphériques.

Naviguez jusqu'à Policy > Policy Sets > (Sélectionnez Policy Set assign) > Authorization Policy.

Associez la règle d'autorisation au profil d'autorisation avec les paramètres MACsec. Faites défiler vers le bas et enregistrez votre configuration.



Configuration de MKA dans Catalyst 9300

Étape 1. Configurez une nouvelle stratégie MKA comme le suggère cet exemple :

```
!
  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
!
```

Étape 2. Activez le chiffrement MACsec dans l'interface à laquelle les PC sont connectés.

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

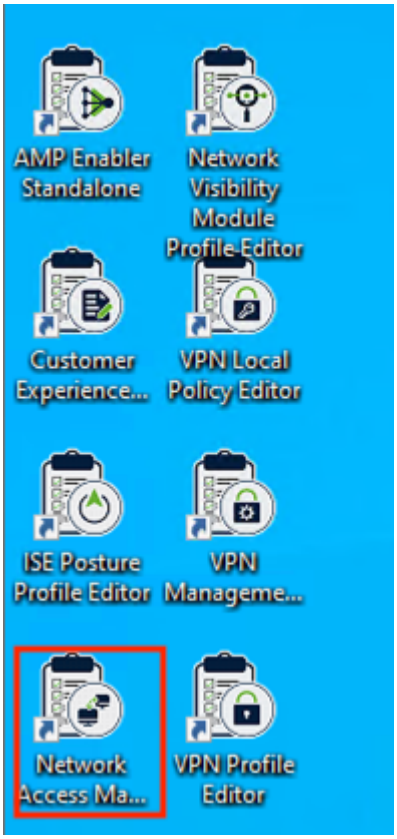
Remarque : Pour plus d'informations sur les commandes et les options de la configuration MKA, consultez le guide de configuration de la sécurité correspondant à la version du commutateur que vous utilisez. Dans ce scénario pour cet exemple, Guide de

[configuration de sécurité, Cisco IOS XE Bengaluru 17.6.x (commutateurs Catalyst 9300)]

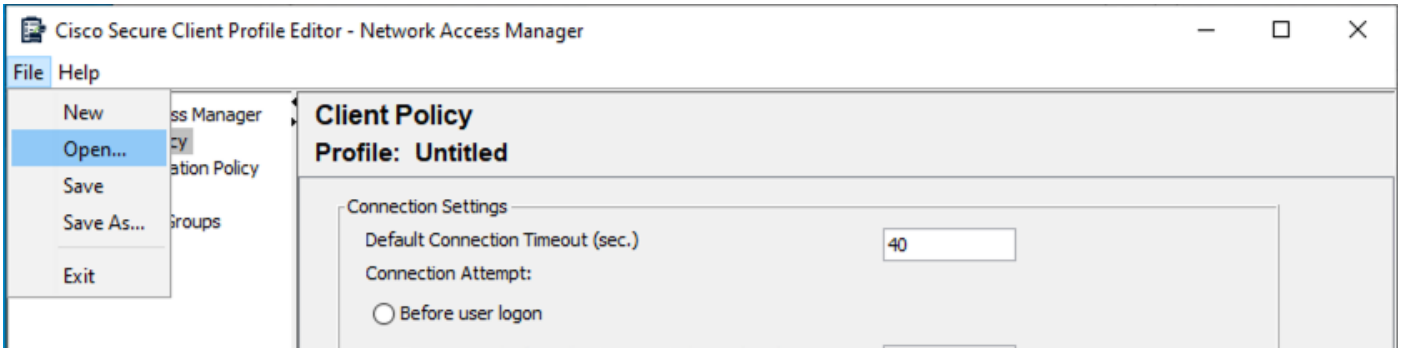
Configuration de MKA à l'aide de l'Éditeur de profil Network Access Manager.

Étape 1. Téléchargez (à partir du site Web de téléchargement de Cisco) et ouvrez l'Éditeur de profil correspondant à la version du client sécurisé utilisée.

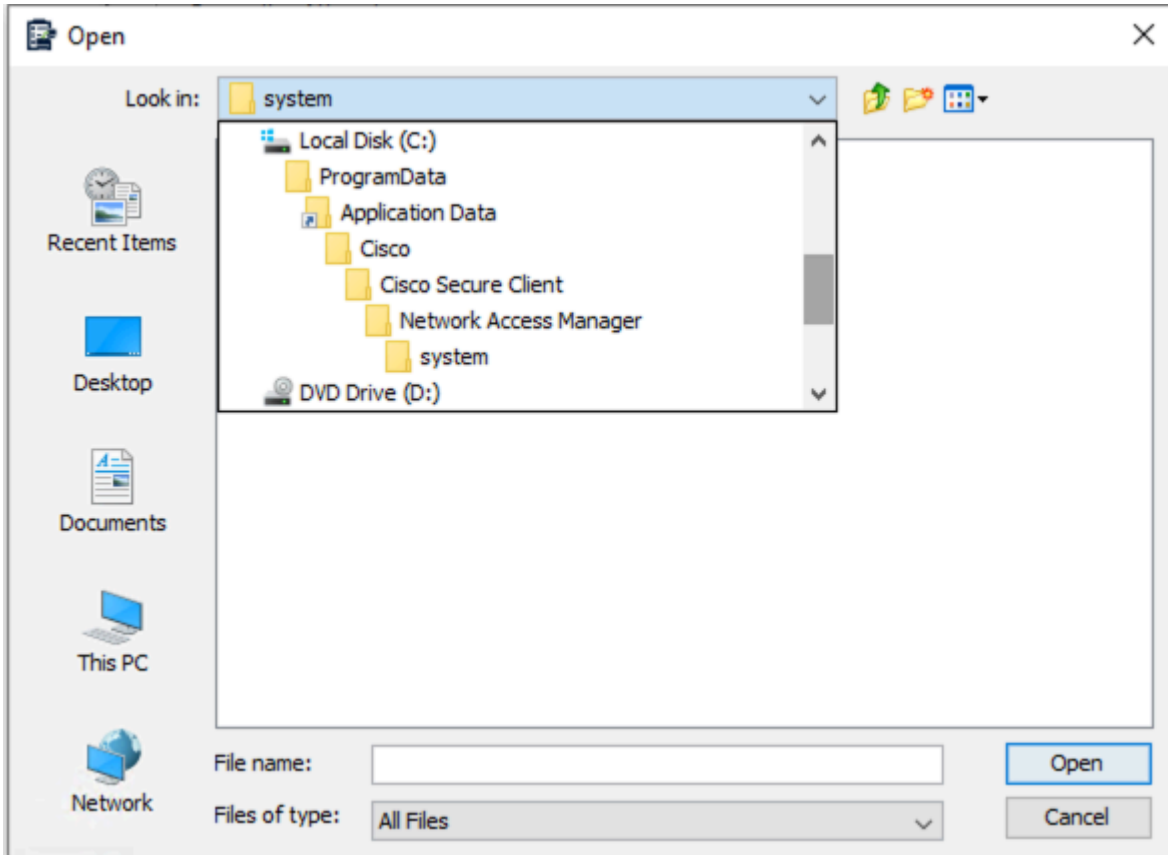
Une fois que vous avez installé ce programme sur votre ordinateur, continuez à ouvrir l'Éditeur de profil client sécurisé Cisco - Gestionnaire d'accès réseau.



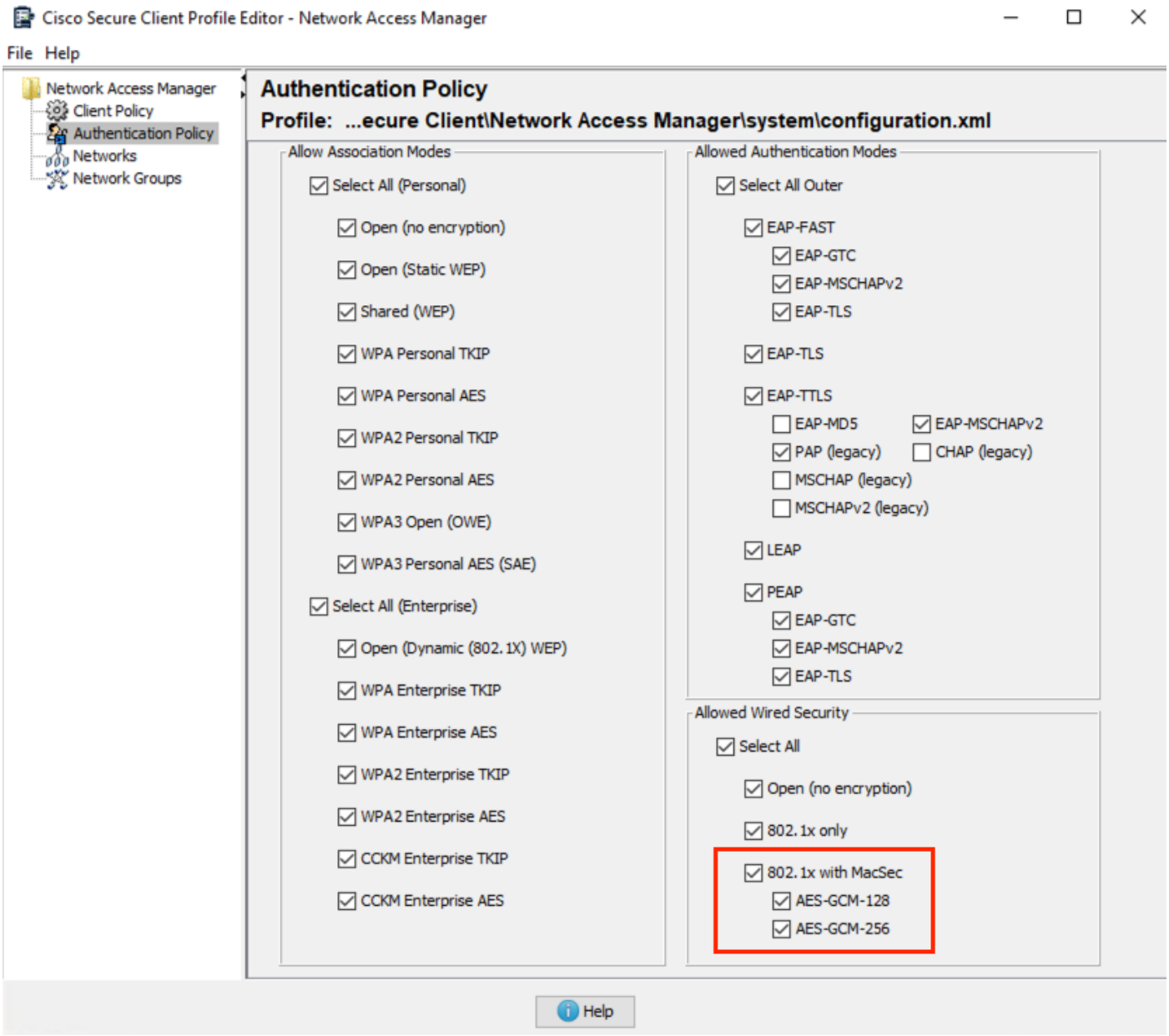
Étape 2. Sélectionnez Fichier > Ouvrir.



Étape 3. Sélectionnez le système de dossiers affiché dans cette image. Dans ce dossier, ouvrez le fichier nommé configuration.xml.

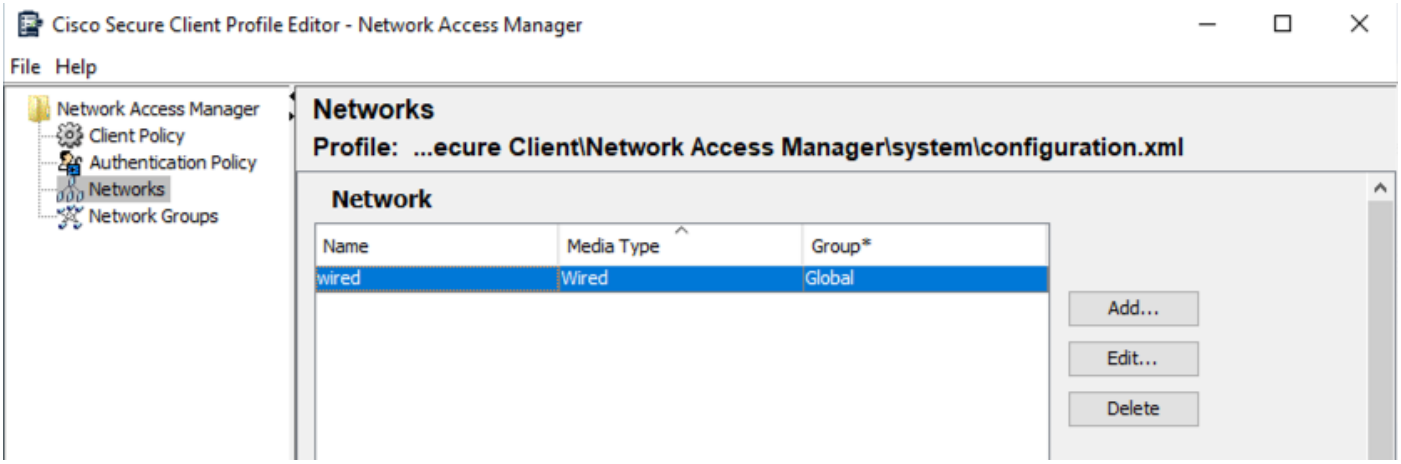


Étape 4. Une fois le fichier chargé par l'Éditeur de profil, sélectionnez l'option Authentication Policy, et assurez-vous que l'option associée à 802.1x avec MACSec est activée.



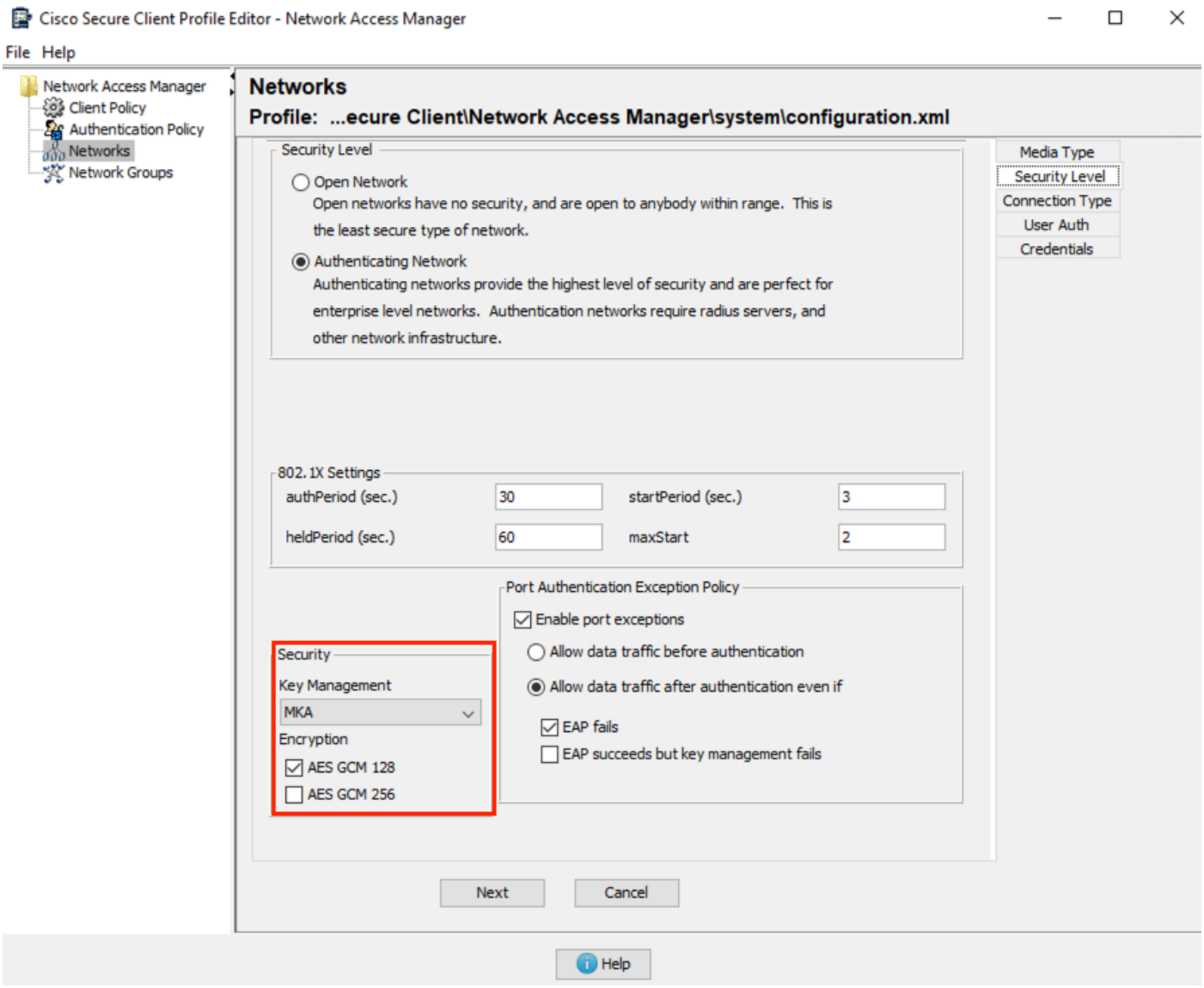
Étape 5. Passez à la section Réseaux. Dans cette partie, vous pouvez Ajouter un nouveau profil pour une connexion filaire ou Modifier le profil filaire par défaut qui est installé avec Secure Client 5.0 .

Dans ce scénario, nous allons modifier le profil filaire existant.



Étape 6. Configuration du profil Dans la section Niveau de sécurité, ajustez la Gestion des clés pour utiliser MKA suivi d'un cryptage AES GCM 128.

Ajustez également les autres paramètres pour l'authentification dot1x et les stratégies.

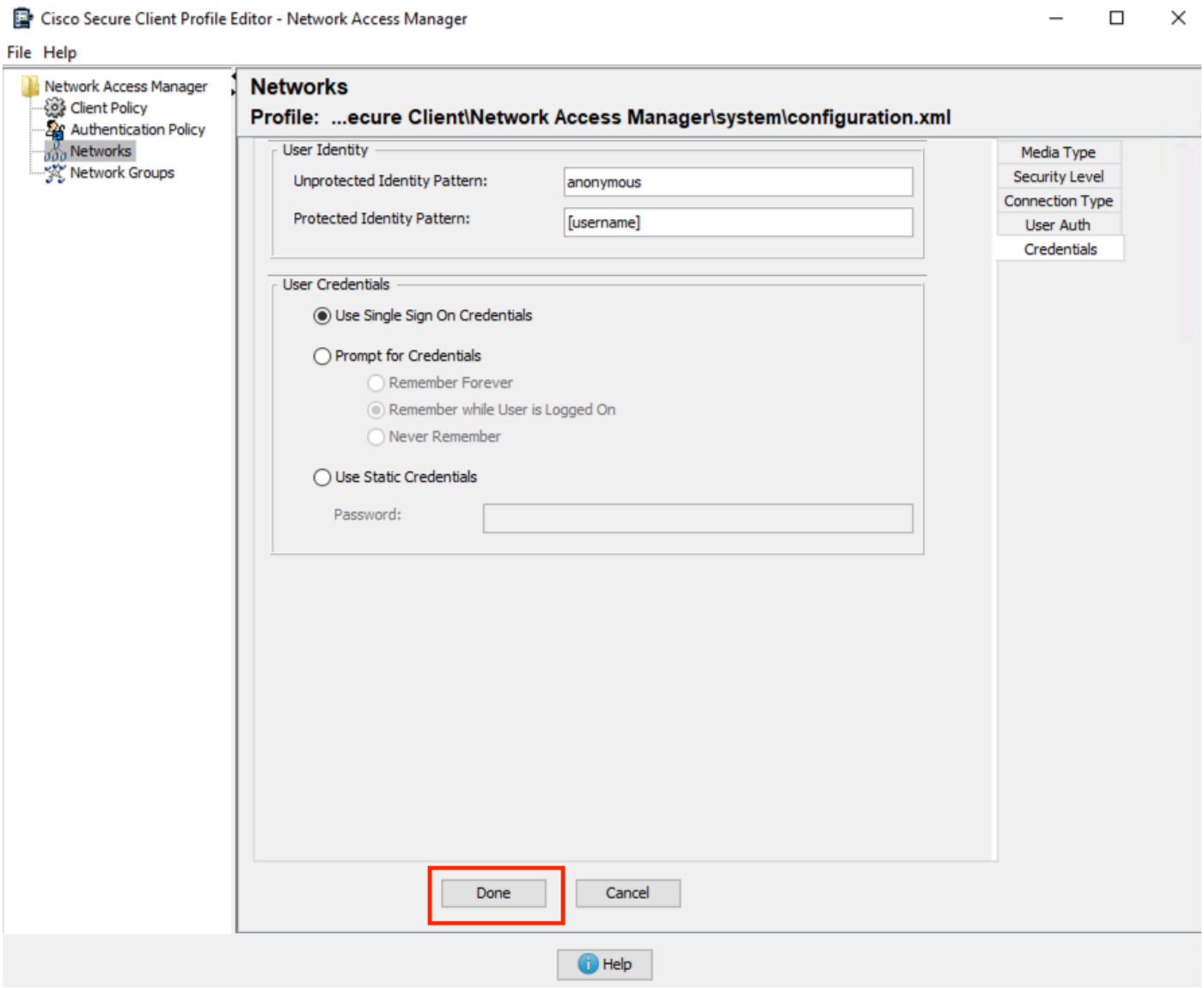


Étape 7. Configurez les sections restantes concernant le type de connexion, l'authentification de l'utilisateur et les informations d'identification.

Ces sections varient en fonction des paramètres d'authentification sélectionnés dans la section Niveau de sécurité.

Une fois les configurations terminées, cliquez sur Done (Terminé).

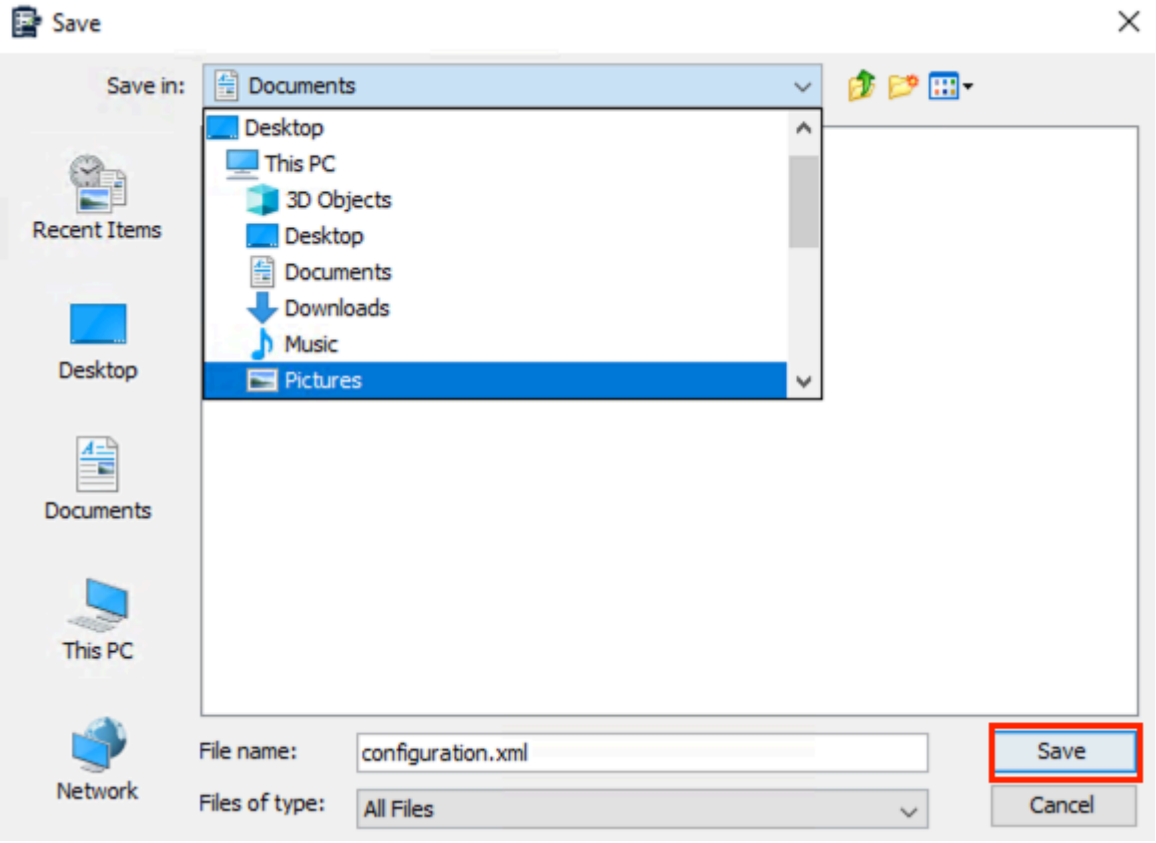
Pour ce scénario, nous utilisons le protocole PEAP (Protected Extensible Authentication Protocol) avec les informations d'identification de l'utilisateur.



Étape 8. Accédez au menu Fichier. Passez à l'option Enregistrer sous.

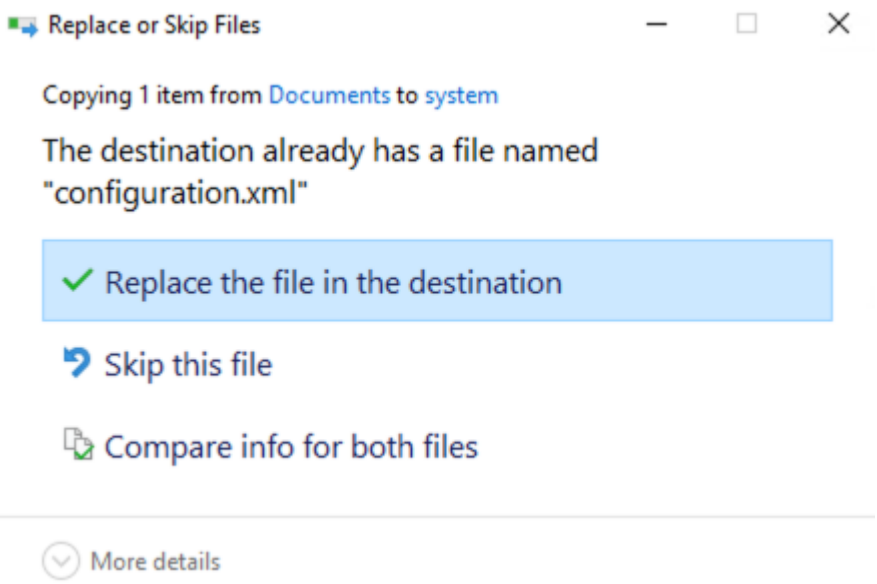
Nommez le fichier configuration.xml et enregistrez-le dans un dossier différent de ProgramData\Cisco\Cisco Secure Client\Network Access Manager\system.

Dans cet exemple, le fichier a été enregistré dans le dossier Documents. Enregistrez le profil.



Étape 8. Accédez à l'emplacement du profil, copiez le fichier et remplacez le fichier contenu dans le dossier ProgramData\Cisco\Cisco Secure Client\Network Access Manager\system.

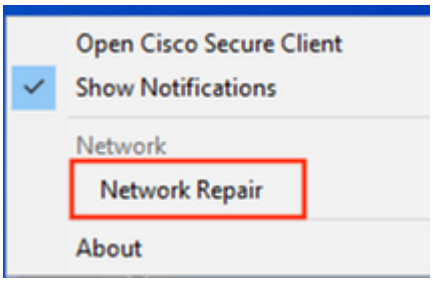
Sélectionnez l'option Remplacer le fichier dans la destination.



Étape 9. Pour charger le profil modifié dans Security Client 5.0, cliquez avec le bouton droit sur

l'icône Secure Client située dans la barre des tâches inférieure droite de votre ordinateur Windows.

Effectuer une réparation du réseau.

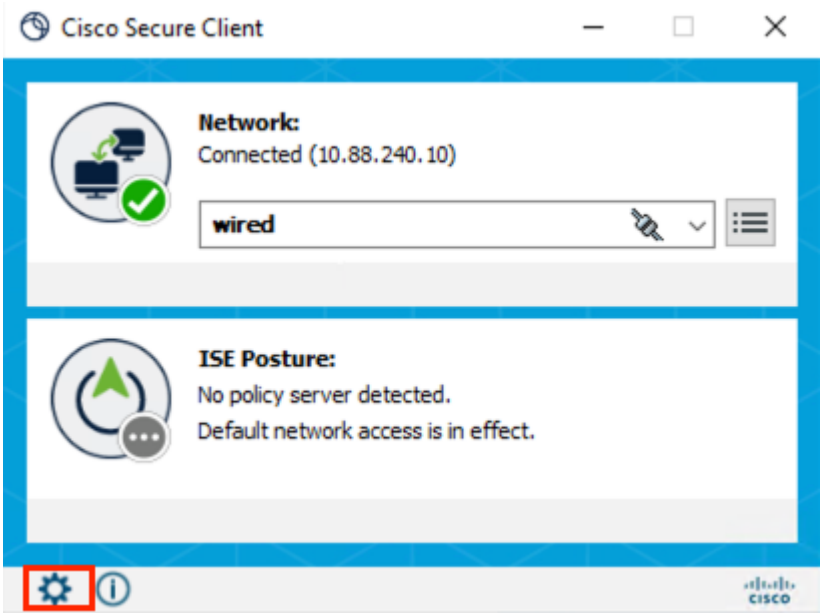


Remarque : Tous les réseaux configurés via l'éditeur de profil disposent des privilèges d'administrateur réseau. Les utilisateurs ne peuvent donc pas personnaliser/modifier le contenu que vous avez configuré à l'aide de cet outil.

Configuration des réseaux MKA à l'aide du Gestionnaire d'accès réseau (facultatif).

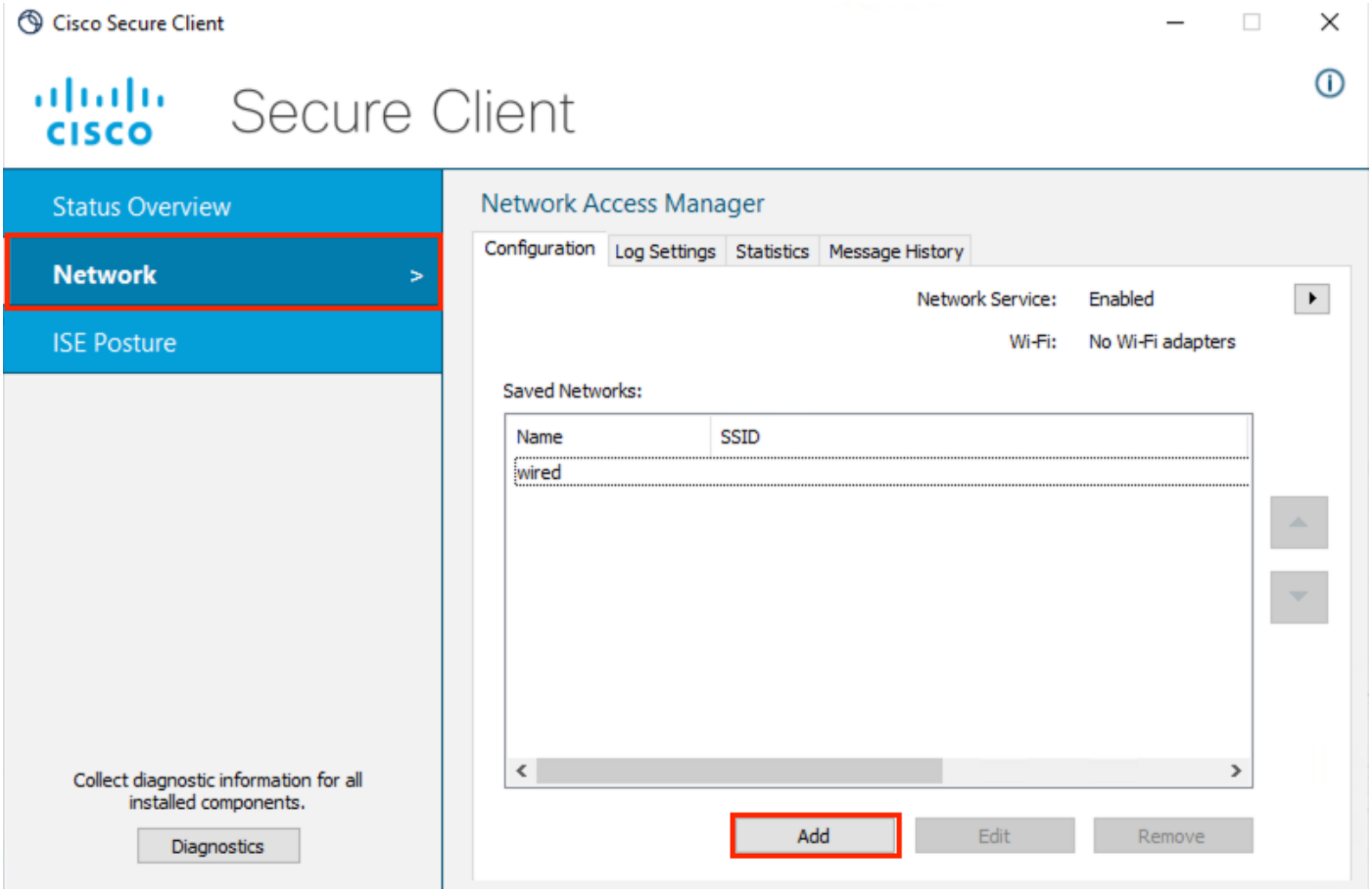
Étape 1. Comme alternative à la configuration MKA à l'aide de l'Éditeur de profil, vous pouvez ajouter des réseaux sans utiliser cet outil.

Dans la suite Secure Client, sélectionnez l'icône de l'engrenage.



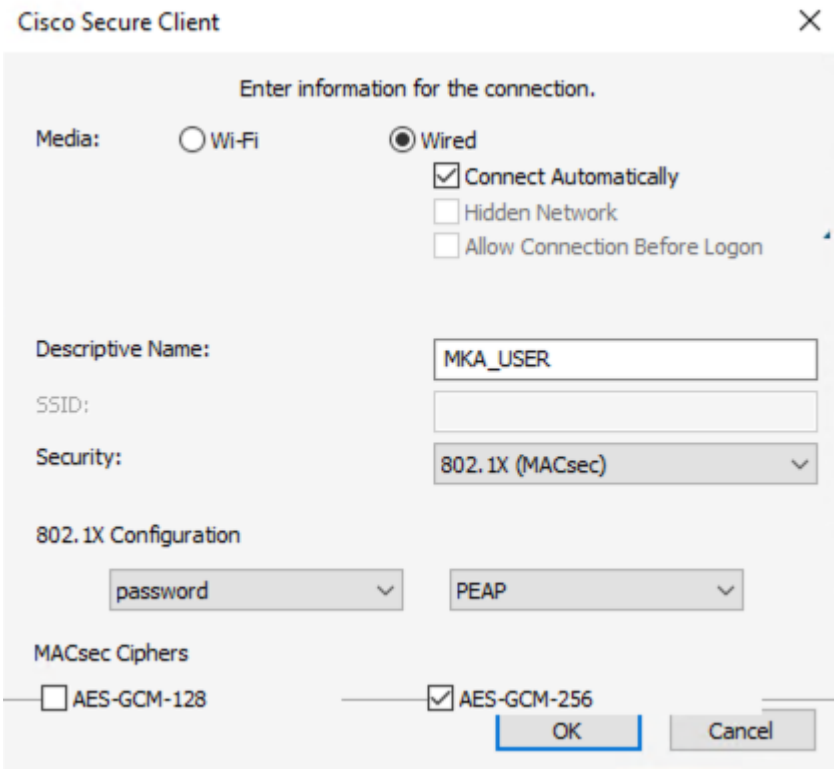
Étape 2. Dans la nouvelle fenêtre affichée, sélectionnez l'option Réseau.

Dans la section Configuration, sélectionnez l'option Add to ingress a network MKA compatible with privileges User Network.



Étape 3. Dans la nouvelle fenêtre de configuration, définissez les caractéristiques de votre connexion et nommez le réseau.

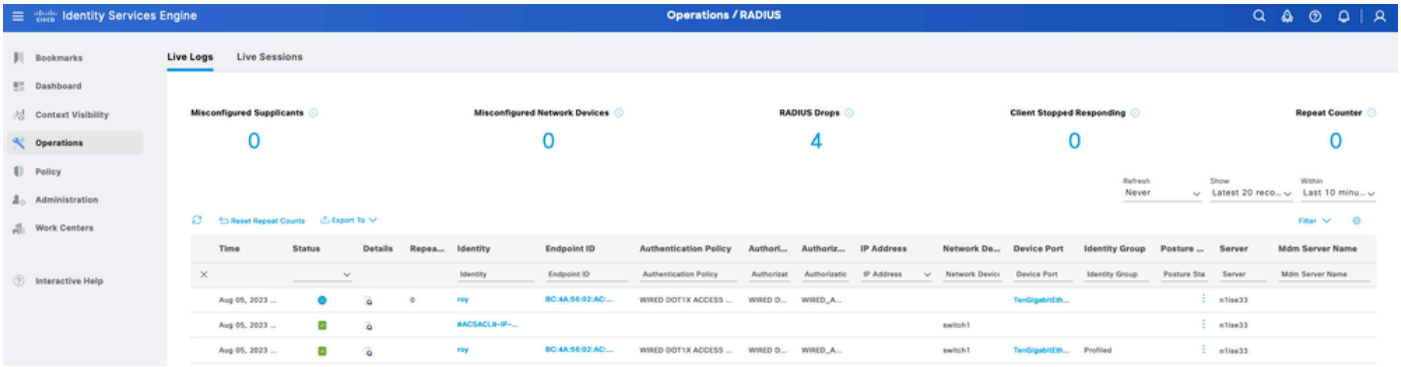
Lorsque vous avez terminé, cliquez sur le bouton OK.



Vérifier

Validation sur ISE

Dans ISE, une fois la configuration de ce flux terminée, notez que le périphérique est authentifié et autorisé dans Livelogs.



Accédez à la section Détails de l'authentification et à la section Résultat.

Les attributs définis dans le profil d'autorisation sont envoyés au périphérique d'accès réseau (NAD) ainsi que la consommation d'une licence Essential.

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.

Validation sur le commutateur Catalyst.

Ces commandes permettent de valider le bon fonctionnement de cette solution.

```
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
```

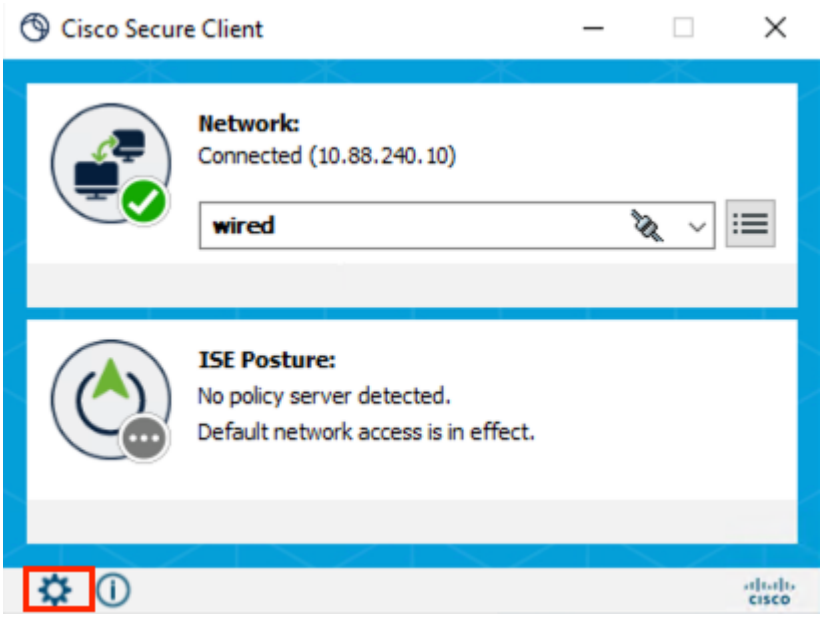
```
switch1#show mka session

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
```

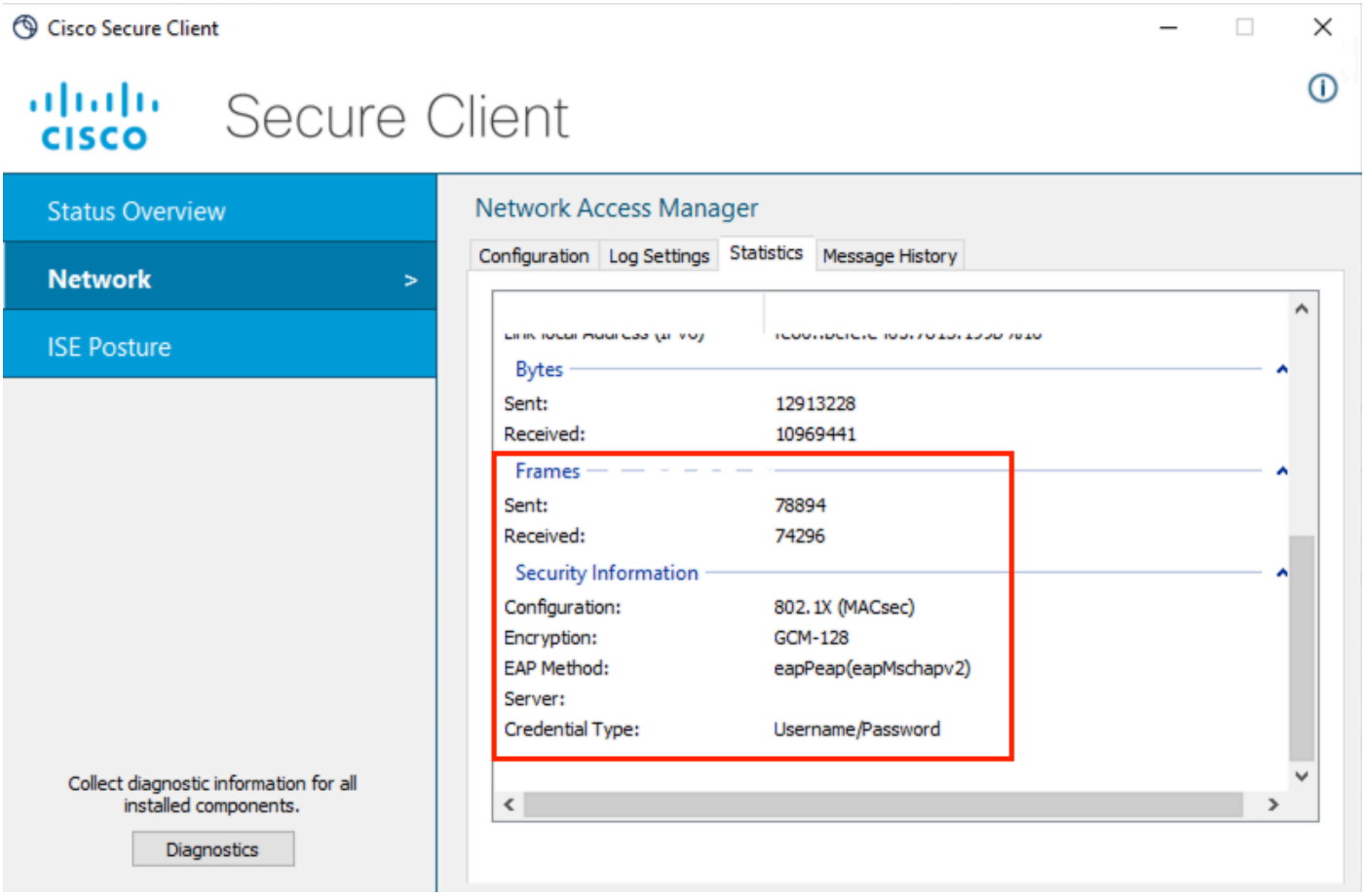
Validation sur le client sécurisé.

L'authentification a réussi avec le profil créé avec le chiffrement MACsec. Si vous cliquez sur l'icône du moteur, des informations supplémentaires peuvent s'afficher.



Dans le menu affiché ici pour le client sécurisé, dans la section Gestionnaire d'accès réseau > Statistiques, notez le Chiffrement et la configuration MACsec correspondante.

Les trames reçues et envoyées augmentent à mesure que le chiffrement est effectué au niveau de la couche 2.

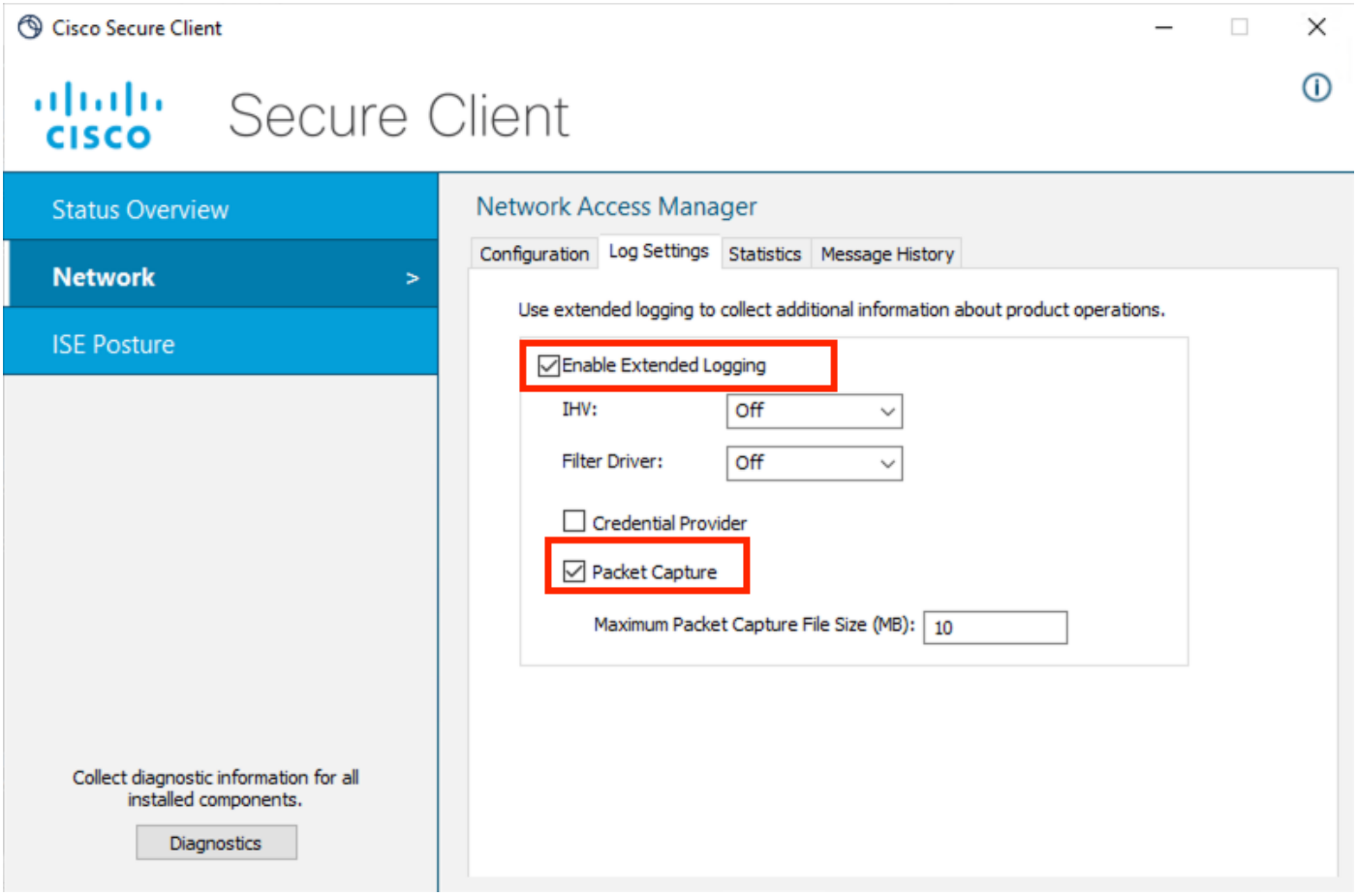


Dépannage

Remarque : Cette section couvre la partie de dépannage liée aux problèmes MKA qui peuvent émerger. Si vous rencontrez un problème d'authentification ou d'autorisation, reportez-vous au Guide de déploiement prescriptif d'accès filaire sécurisé ISE - Dépannage pour en savoir plus. Ce guide suppose que les authentifications fonctionnent correctement sans chiffrement MACsec.

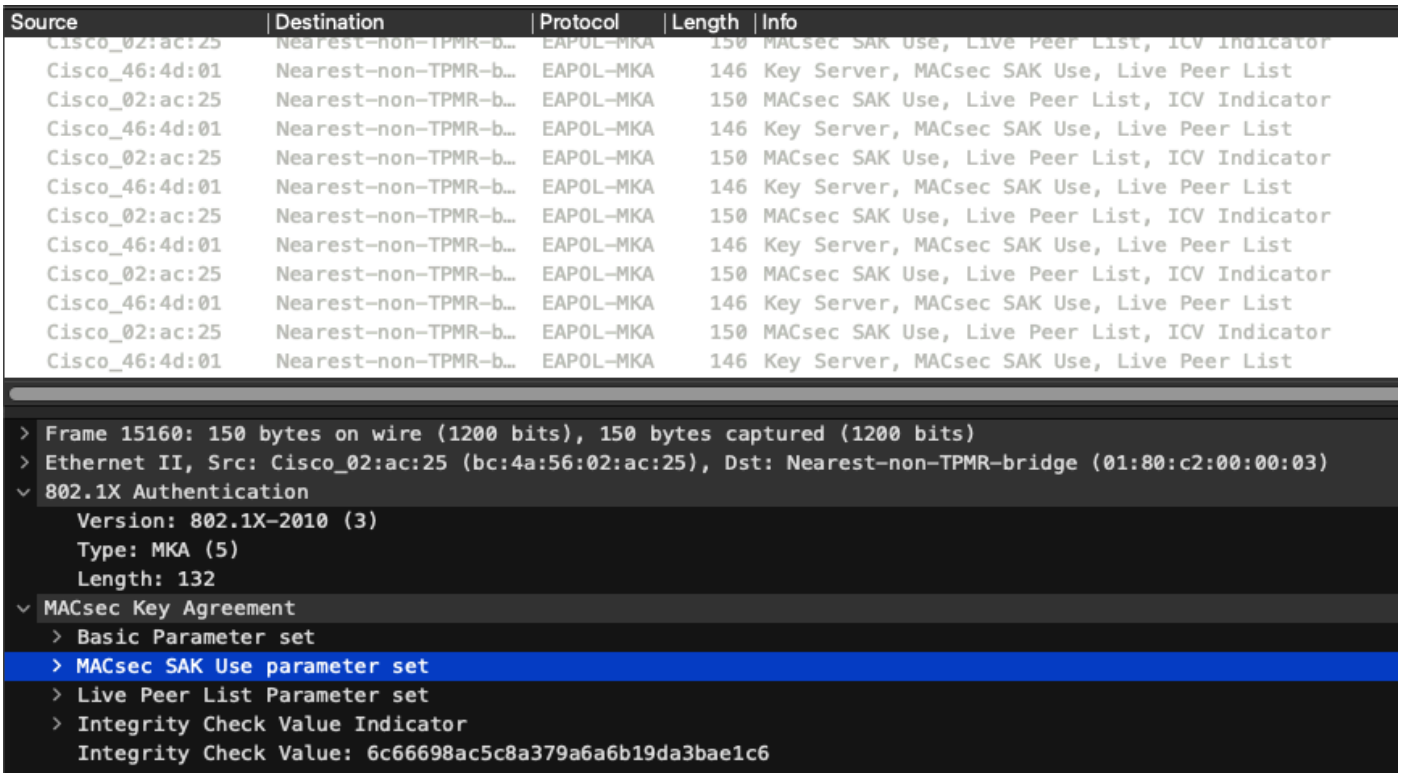
Terminaux sécurisés Cisco

- Activez le module DART dans la suite Secure Endpoint.
- Dans ce menu, activez la consignation étendue pour collecter davantage de données sur la connexion MKA de l'utilisateur. Vous pouvez également activer une capture de paquets contenue dans le bundle DART.



- Collectez l'offre groupée DART pour poursuivre l'analyse du fichier configuration.xml et du Gestionnaire d'accès réseau. Reportez-vous à la documentation [Exécuter DART pour collecter des données pour le dépannage]

Cet exemple montre comment les paquets sont perçus comme les informations entre l'hôte et le commutateur sont chiffrées :



À partir du bundle DART, nous pouvons trouver des informations utiles pour l'authentification 802.1X et la session MKA dans le journal nommé NetworkAccessManager.txt.

Ces informations sont affichées dans une authentification réussie avec chiffrement MKA.

```
%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)
```

Dépannage de Cisco IOS

Ces commandes peuvent être implémentées dans le périphérique d'accès réseau (NAD) pour examiner le chiffrement MKA entre la plate-forme et le demandeur.

Pour plus d'informations sur les commandes, consultez le guide de configuration correspondant de la plate-forme utilisée comme NAD.

```
#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
```

Il s'agit des débogages d'une connexion MKA réussie vers un hôte. Vous pouvez l'utiliser comme référence pour l'affichage :

```
%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
```

Dépannage d'Identity Services Engine (ISE)

Le dépannage lié à cette fonctionnalité est limité à la livraison de l'attribut cisco-av-pair linksec-policy=should-secure.

Assurez-vous que le résultat de l'autorisation envoie ces informations à la session Radius liée aux ports de commutation où les périphériques sont connectés.

Pour une analyse plus approfondie de l'authentification sur ISE, consultez Dépannage et activation des débogages sur ISE

Problèmes courants

Non-concordance de chiffrement

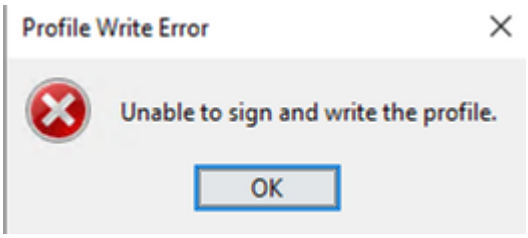
Ce journal peut être vu dans les débogages MKA dans NAD.

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

La première chose à vérifier dans ce scénario est que les chiffrements configurés dans la stratégie MKA dans le commutateur et dans le profil Secure Client correspondent.

Dans le cas du cryptage AES-GCM-256, ces exigences doivent être respectées (conformément à la documentation) Guide de l'administrateur du client sécurisé Cisco (y compris AnyConnect), version 5

Impossible d'enregistrer le fichier configuration.xml.



Ce problème lié à l'erreur d'écriture de profil est résolu en enregistrant le fichier configuration.xml (comme décrit précédemment) nommé Setup of MKA à l'aide de l'Éditeur de profil Network Access Manager.

L'erreur est liée au fait que le fichier configuration.xml utilisé ne peut pas être modifié. Par conséquent, enregistrez le fichier à un autre emplacement pour procéder au remplacement des profils.

Informations connexes

- Configuration du chiffrement MACsec
- Configuration du commutateur MACsec sur l'hôte avec Cat9k et ISE

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.