ISEを使用したIOS XEデバイスでのTACACS+ over TLS 1.3の設定

内容

はじめに

このドキュメントでは、サーバとしてCisco Identity Services Engine(ISE)、クライアントとしてCisco IOS® XEデバイスを使用したTACACS+ over TLSの例について説明します。

概要

Terminal Access Controller Access-Control System Plus(TACACS+)プロトコル[RFC8907]を使用すると、1台以上のTACACS+サーバを介して、ルータ、ネットワークアクセスサーバ、およびその他のネットワークデバイスを一元的に管理できます。認証、許可、アカウンティング(AAA)サービスを提供し、デバイス管理のユースケースに合わせて特別に調整されています。

TACACS+ over TLS 1.3 [RFC8446]は、安全性の高いトランスポート層を導入して機密性の高いデータを保護することで、プロトコルを強化します。この統合により、TACACS+クライアントとサーバ間の接続およびネットワークトラフィックの機密性、整合性、および認証が確保されます。

このガイドの使用方法

このガイドでは、アクティビティを2つの部分に分けて、ISEでCisco IOS XEベースのネットワークデバイスの管理アクセスを管理できるようにします。
- パート1:Device Admin用のISEの設定
- パート2:TACACS+ over TLS用のCisco IOS XEの設定

前提条件

要件

TACACS+ over TLSを設定するための要件：

- ISEおよびネットワークデバイスの証明書に署名するためにTACACS+ over TLSで使用される証明書に署名するための認証局(CA)。
- 認証局(CA)からのルート証明書。

- ネットワークデバイスとISEにはDNS到達可能性があり、ホスト名を解決できます。

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。

- ISE VMware仮想アプライアンス、リリース3.4パッチ2
- Cisco IOS XEソフトウェアバージョン17.15+

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

ライセンス

デバイス管理ライセンスを使用すると、ポリシーサービスノードでTACACS+サービスを使用で

きます。ハイアベイラビリティ(HA)スタンドアロン導入では、デバイス管理ライセンスにより、HAペアの単一のポリシーサービスノードでTACACS+サービスを使用できます。

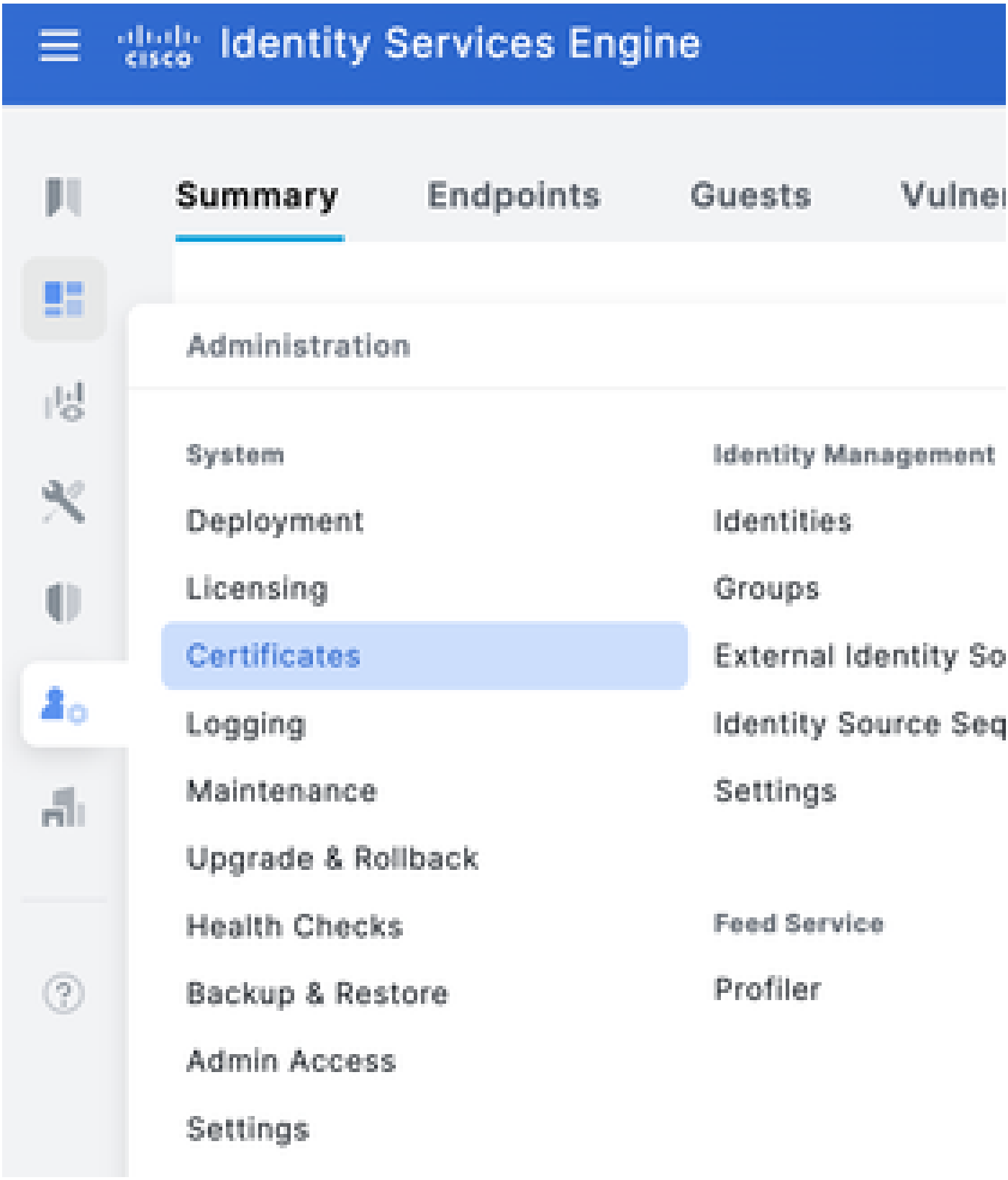
パート1：デバイス管理用のISEの設定

TACACS+サーバ認証用の証明書署名要求の生成

ステップ 1： サポートされているいずれかのブラウザを使用して、ISE管理Webポータルにログインします。

デフォルトでは、ISEはすべてのサービスに自己署名証明書を使用します。最初の手順では、証明書署名要求(CSR)を生成して、認証局(CA)によって署名されるようにします。

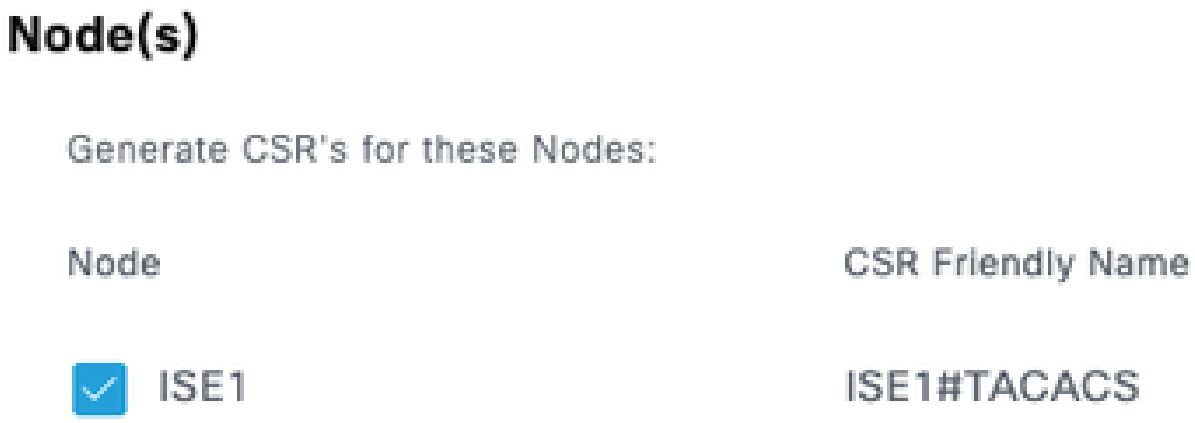
ステップ2:Administration > System > Certificatesの順に選択します。



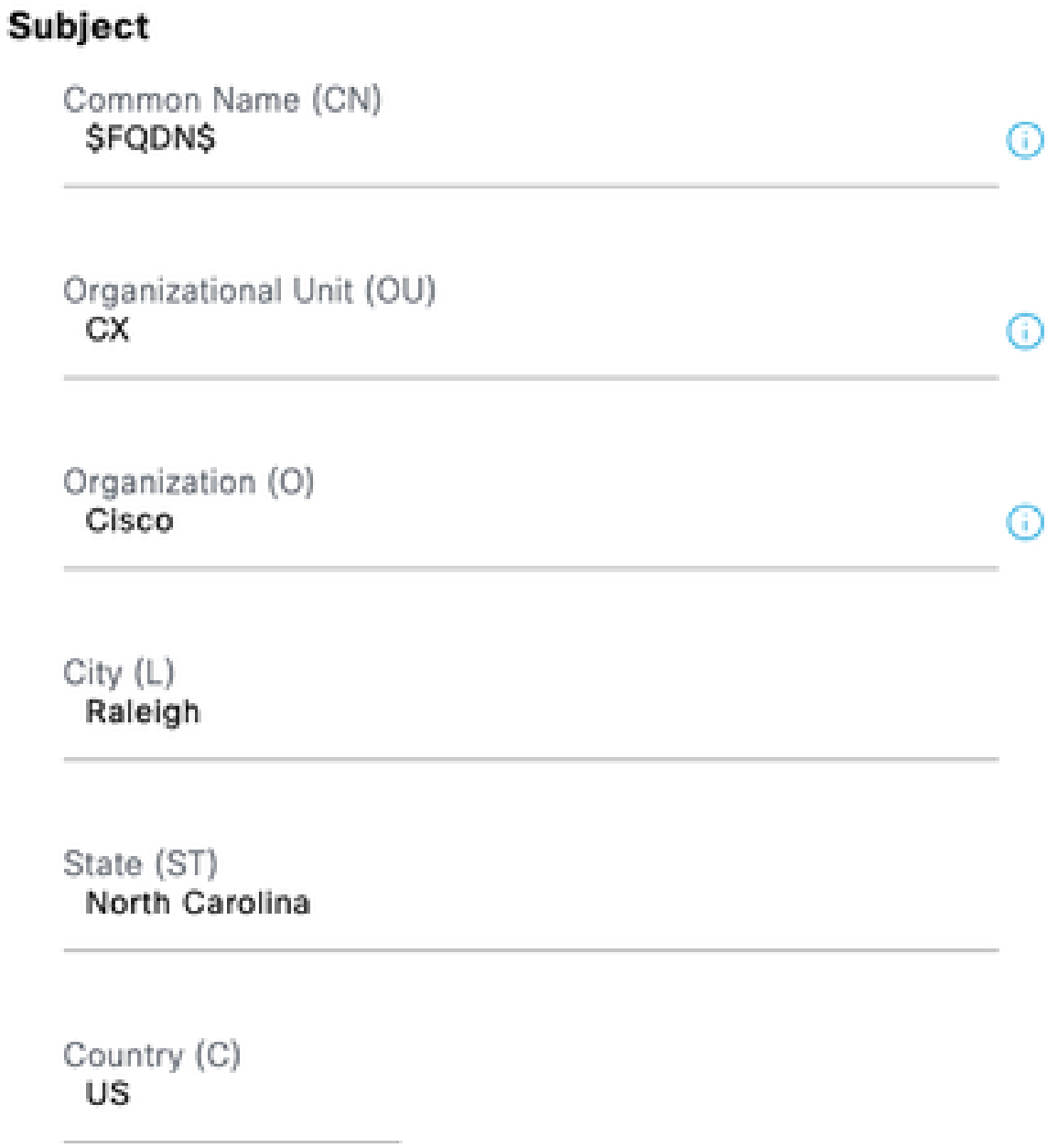
ステップ3:Certificate Signing Requestsの下で、Generate Certificate Signing Requestをクリックします。

ステップ4:TACACS inUsageを選択します。

ステップ5:TACACS+を有効にするPSNを選択します。

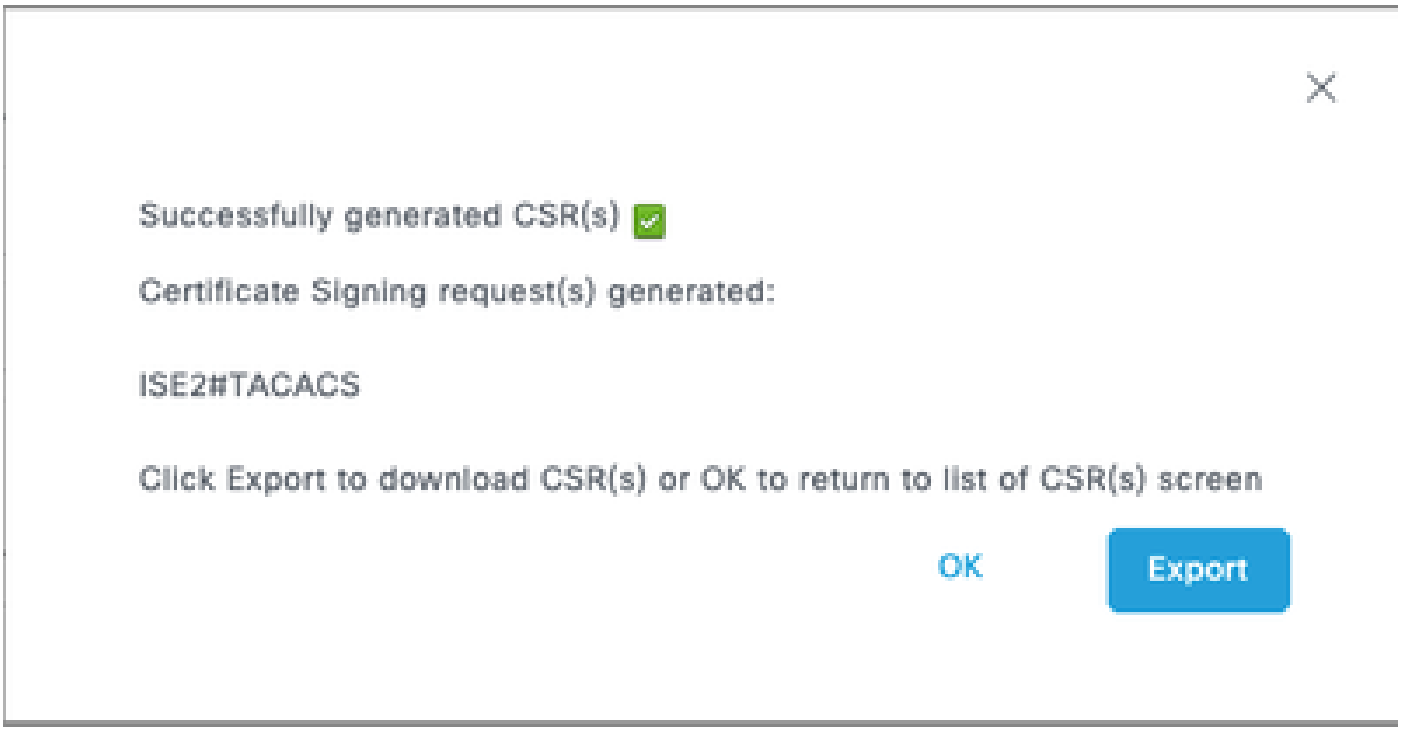


手順 6： Subjectフィールドに、適切な情報を入力します。



ステップ7:サブジェクト代替名(SAN)の下にDNS名とIPアドレスを追加します。

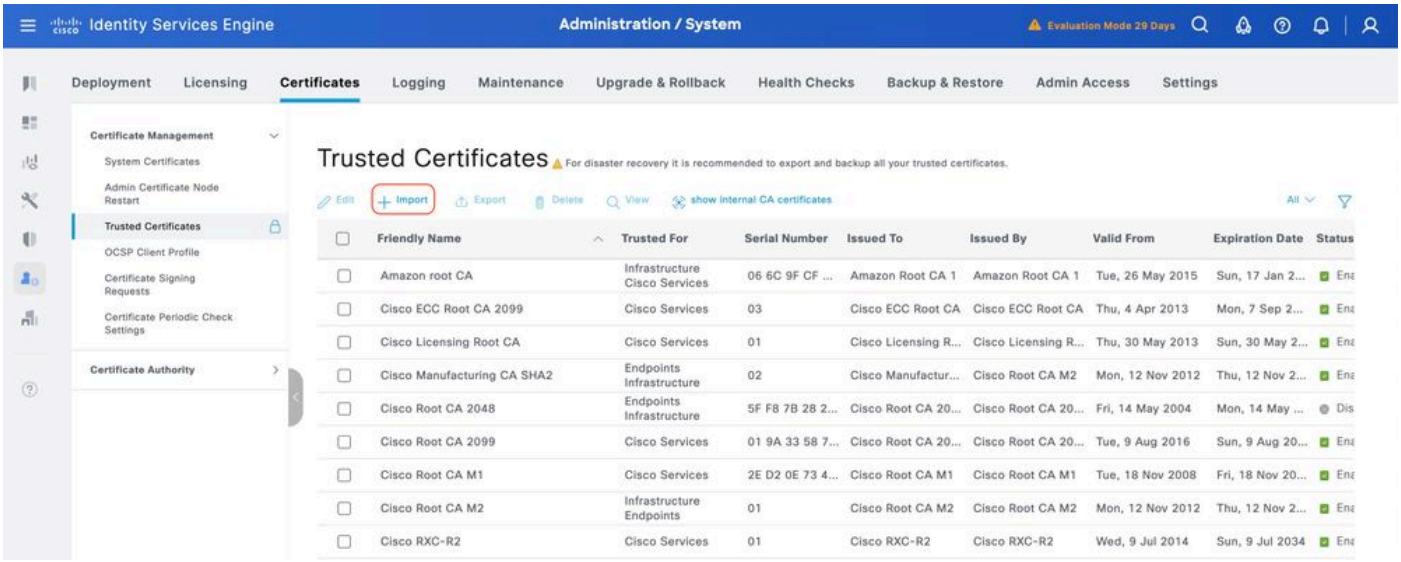
ステップ8:Generateをクリックし、次にExportをクリックします。



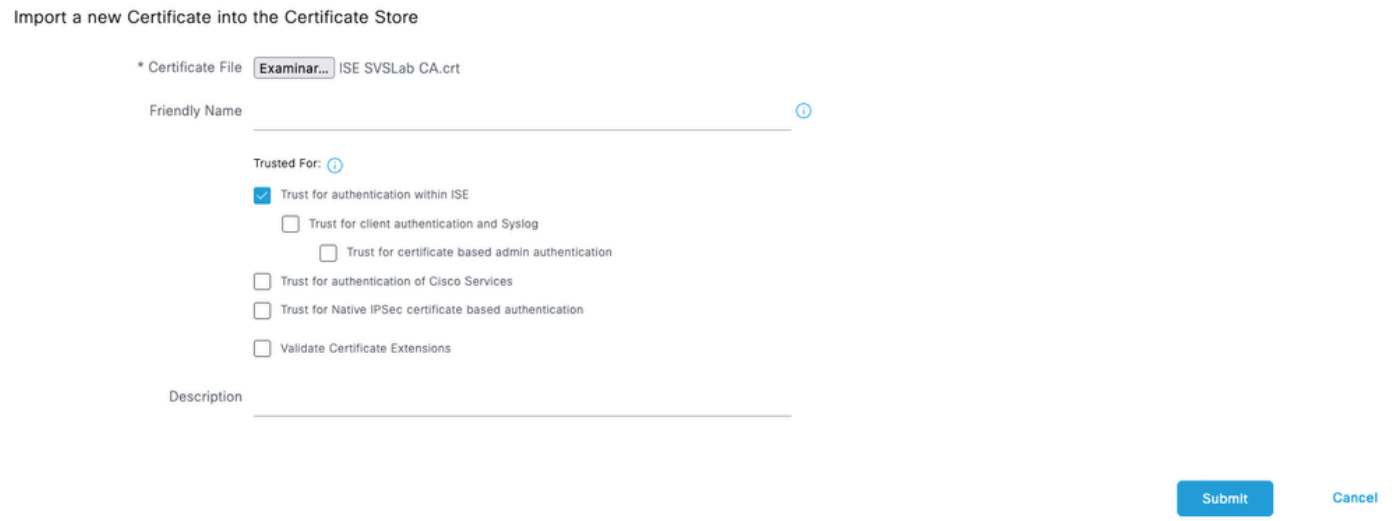
これで、認証局(CA)によって署名された証明書(CRT)を取得できます。

TACACS+サーバ認証用のルートCA証明書のアップロード

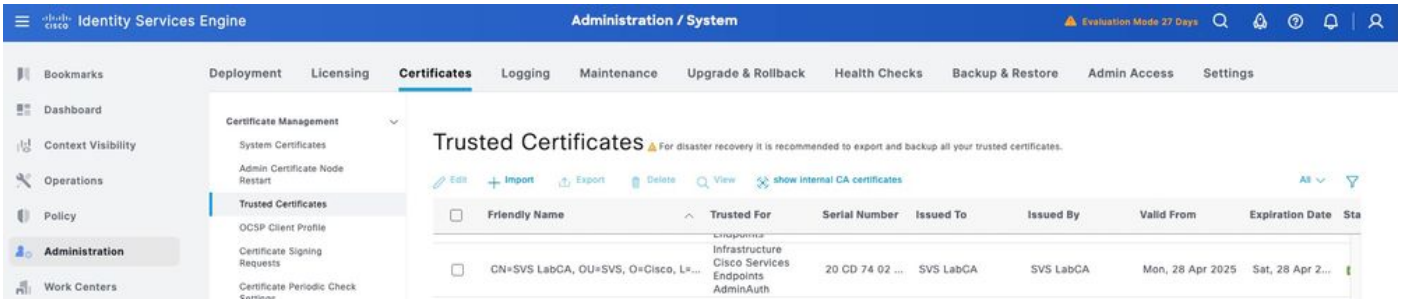
ステップ 1：Administration > System > Certificatesの順に選択します。Trusted Certificatesの下にあるImportをクリックします。



ステップ 2： TACACS証明書署名要求(CSR)に署名した認証局(CA)によって発行された証明書を選択します。 次のことを確認してください。ISE内での認証の信頼性 オプションが有効になっている。



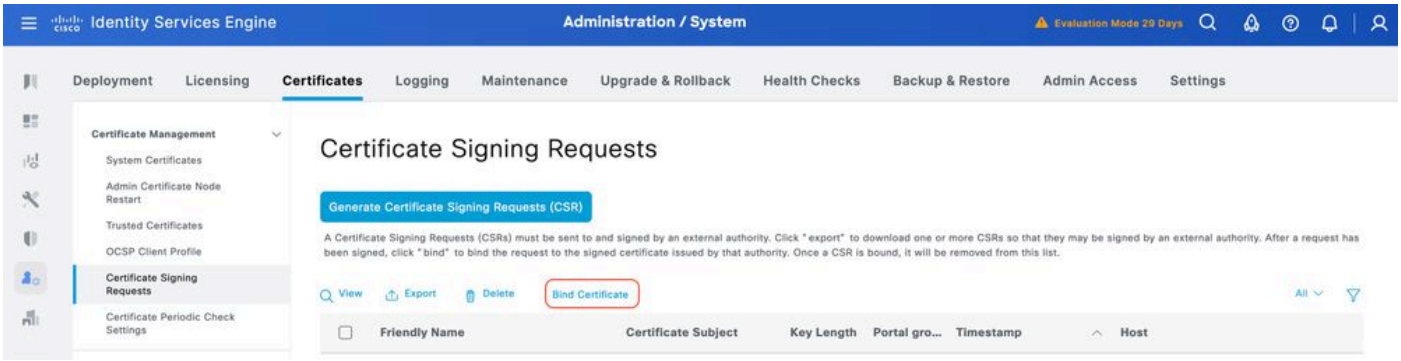
ステップ 3：[Submit] をクリックします。 証明書がTrusted Certificatesの下に表示されている必要があります。



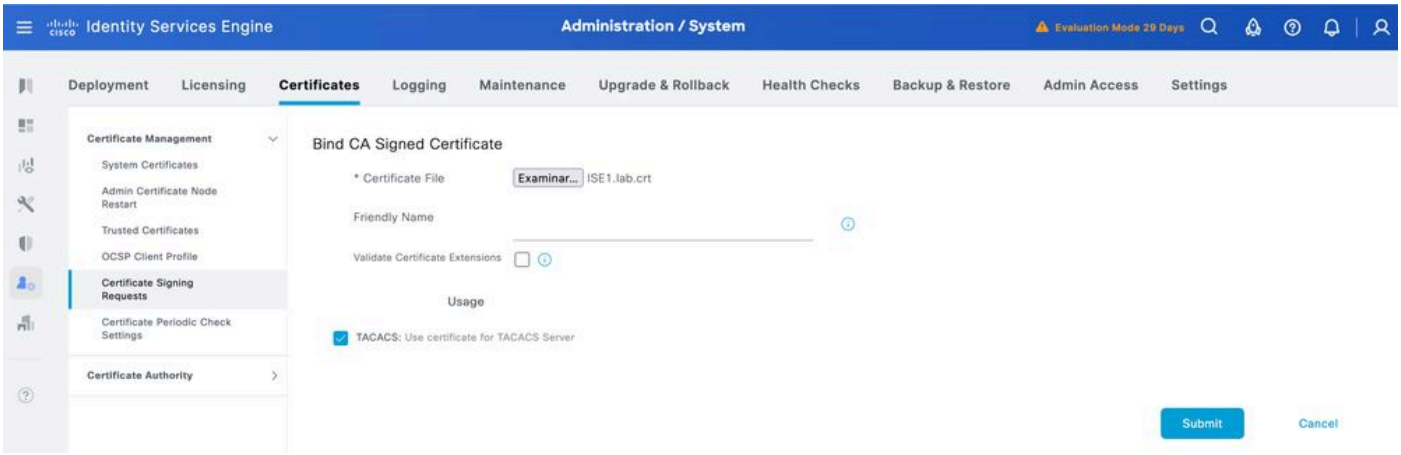
署名付き証明書署名要求(CSR)のISEへのバインド

証明書署名要求(CSR)が署名されたら、署名付き証明書をISEにインストールできます。

ステップ1:Administration > System > Certificatesの順に選択します。Certificate Signing Requestsの下で、前のステップで生成したTACACS CSRを選択し、Bind Certificateをクリックします。

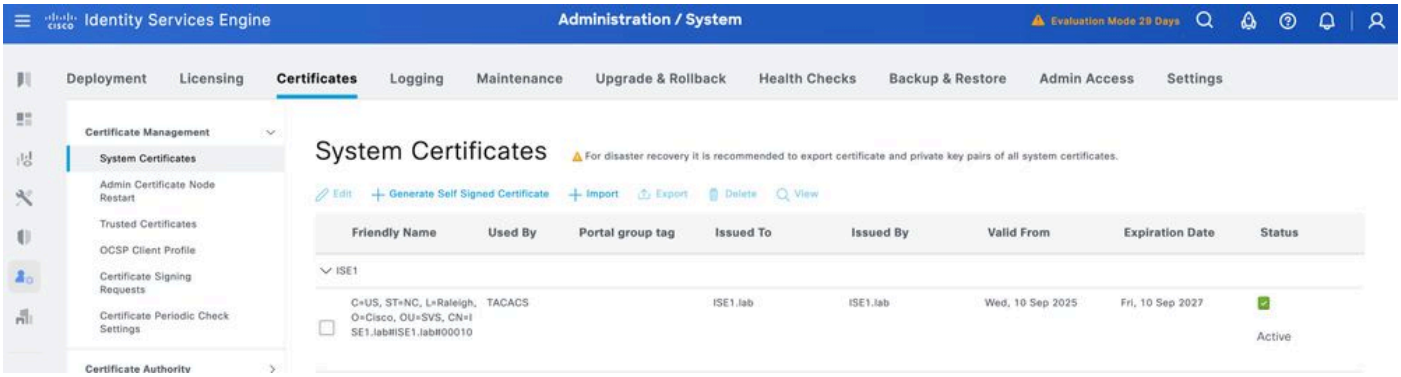


ステップ2:署名付き証明書を選択し、Usage の下のTACACS チェックボックスが選択されたままになっていることを確認します。



ステップ3:Submitをクリックします。既存の証明書の置き換えに関する警告が表示されたら、Yesをクリックして続行します。

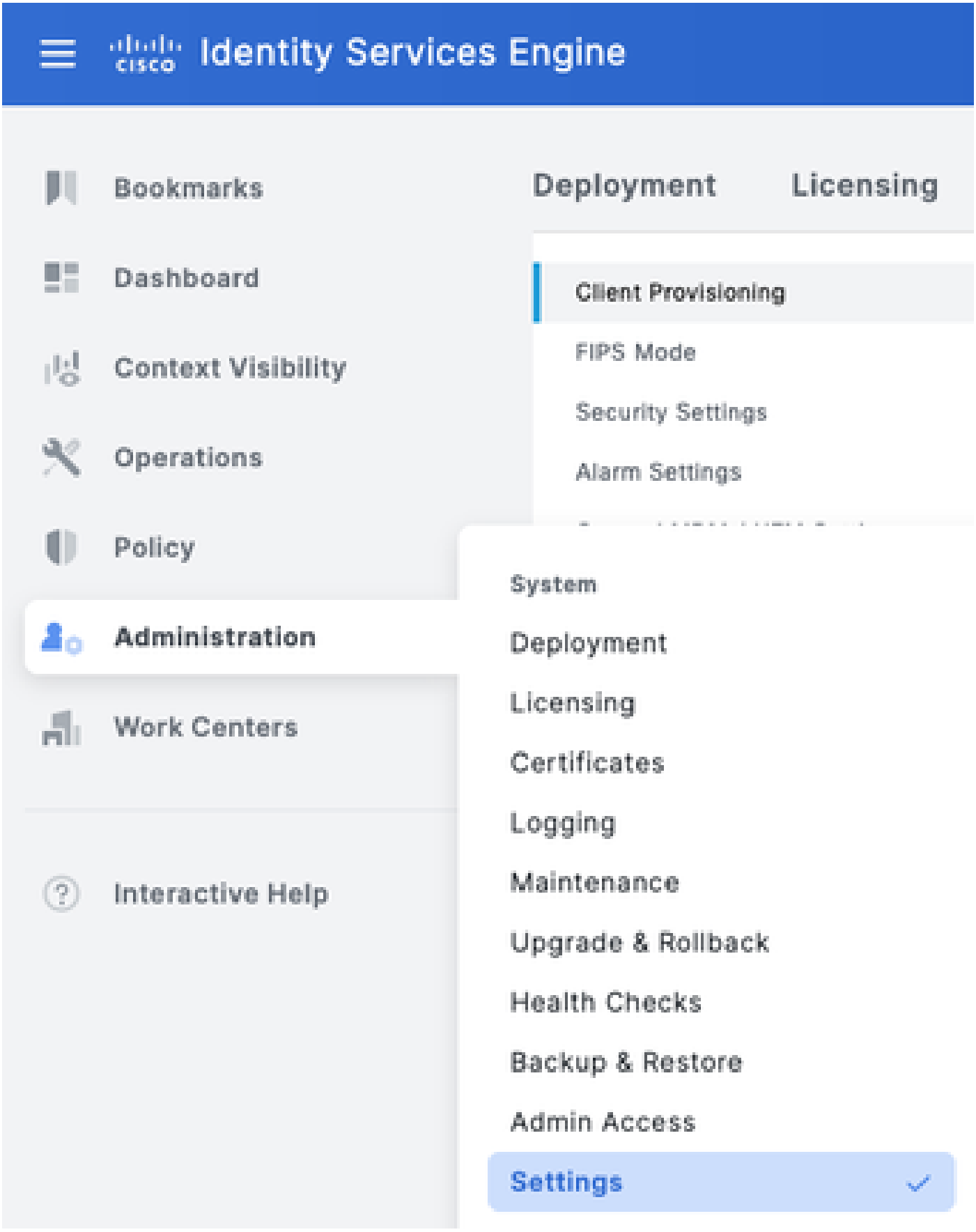
これで、証明書が正しくインストールされました。これは、「システム証明書」で確認できます。



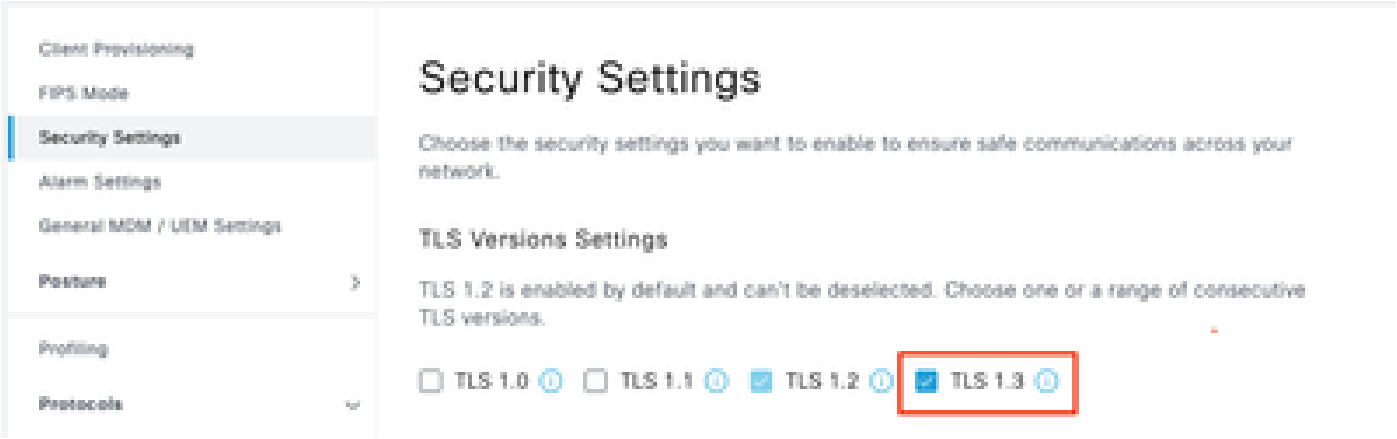
TLS 1.3を有効にする

TLS 1.3は、ISE 3.4.xではデフォルトで有効になっていません。手動で有効にする必要があります。

ステップ1:Administration > System > Settingsの順に選択します。



ステップ2:Security Settingsをクリックし、TLS Version Settingsの下でTLS1.3 の横にあるチェックボックスをオンにして、Saveをクリックします。

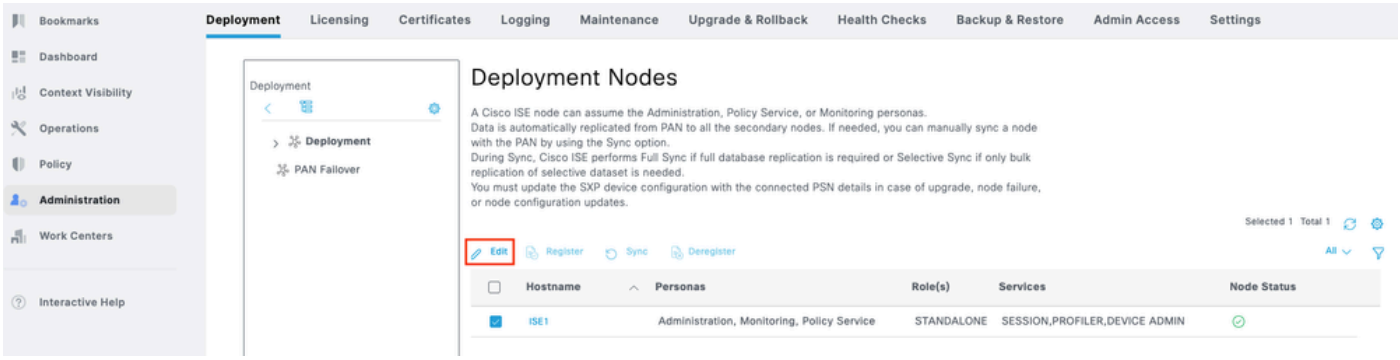


警告:TLSバージョンを変更すると、Cisco ISEアプリケーションサーバがすべてのCisco ISE導入マシンで再起動します。

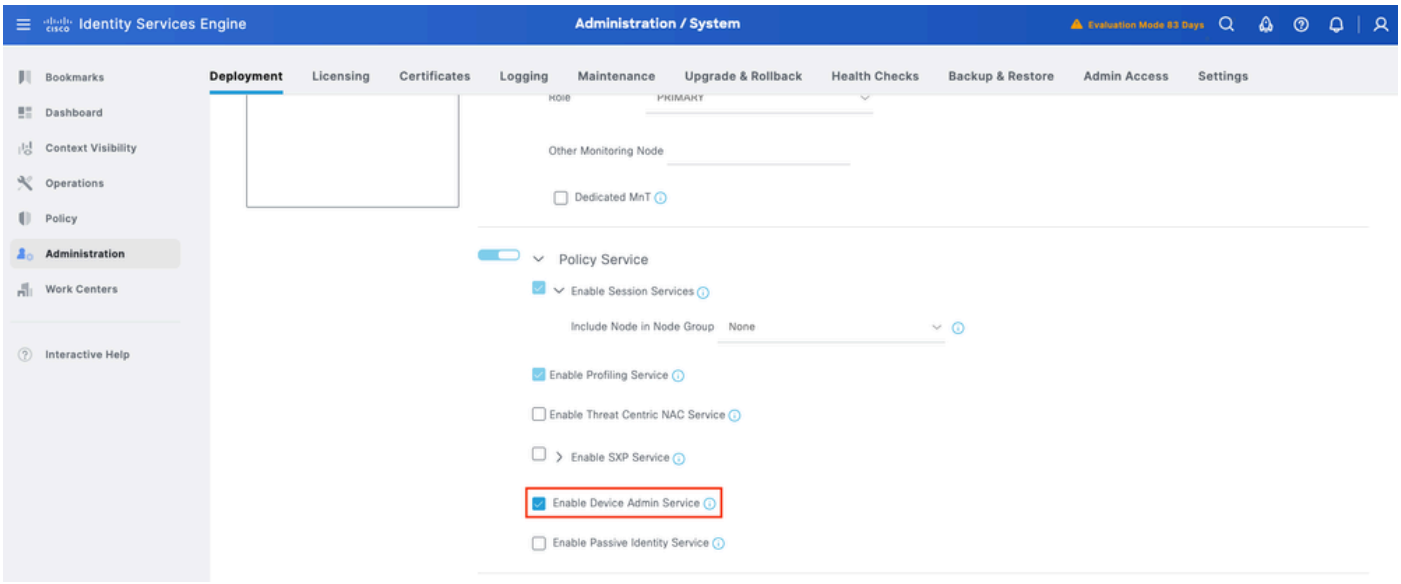
ISEでのデバイス管理の有効化

ISEノードでは、デバイス管理サービス(TACACS+)はデフォルトで有効になっていません。PSNノードでTACACS+を有効にします。

ステップ1:Administration > System > Deploymentの順に選択します。ISEノードの横にあるチェックボックスをオンにし、Editをクリックします。



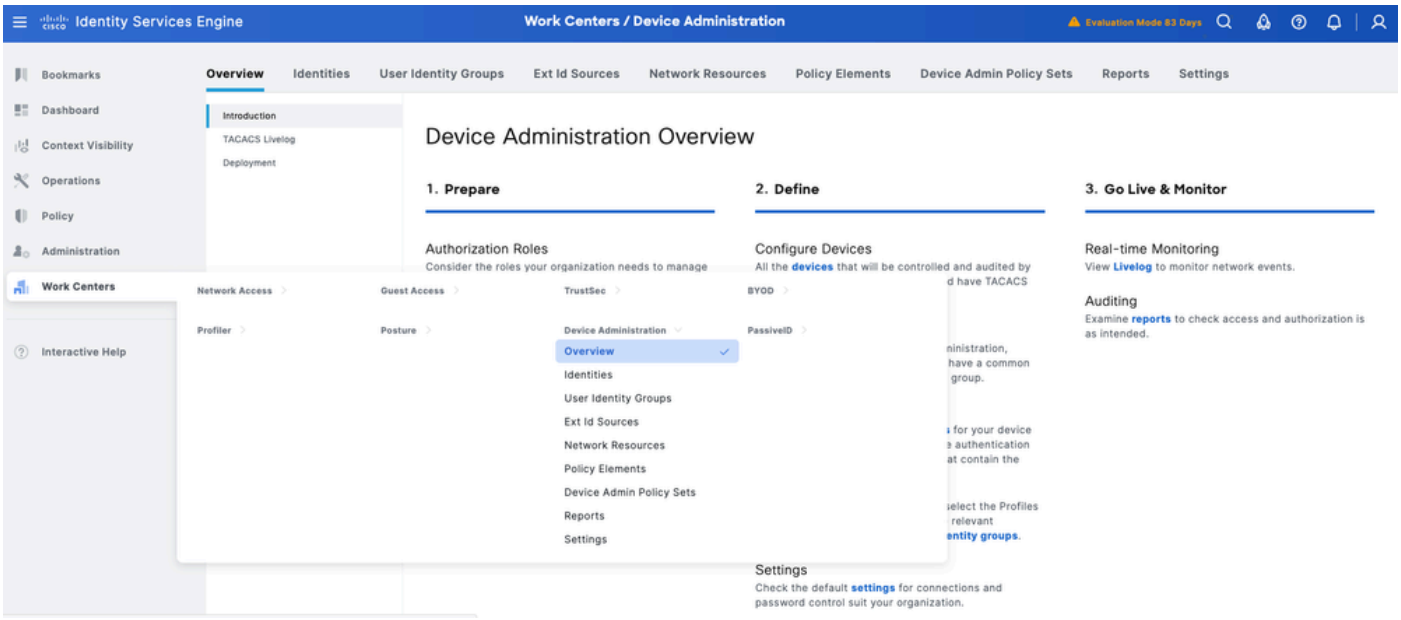
ステップ2:GeneralSettingsで、スクロールダウンしてEnable Device Admin Serviceの横にあるチェックボックスをオンにします。



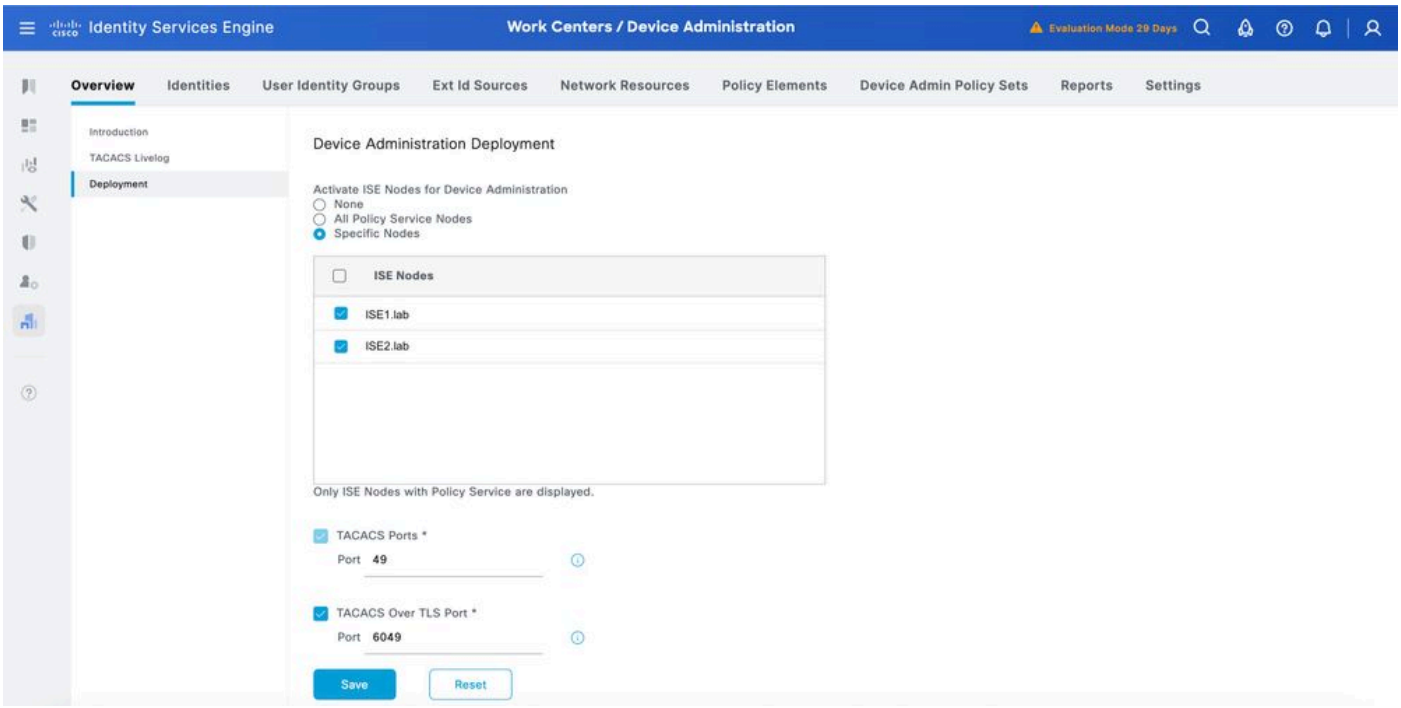
ステップ 3： 設定を保存します。Device Admin ServiceがISEで有効になりました。

TLS経由のTACACSの有効化

ステップ1:Work Centers > Device Administration > Overviewの順に移動します。



ステップ2:Deploymentをクリックします。TACACS over TLSを有効にするPSN ノードを選択します。

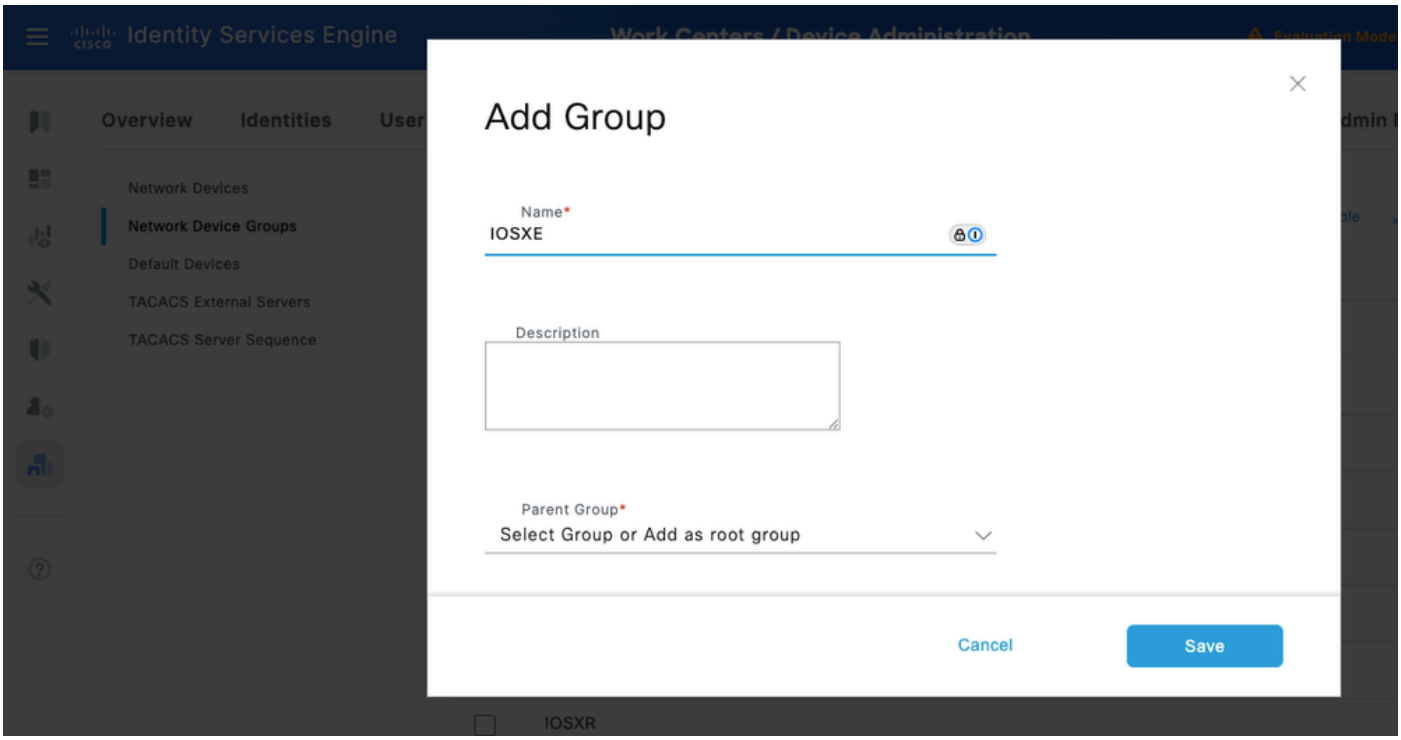


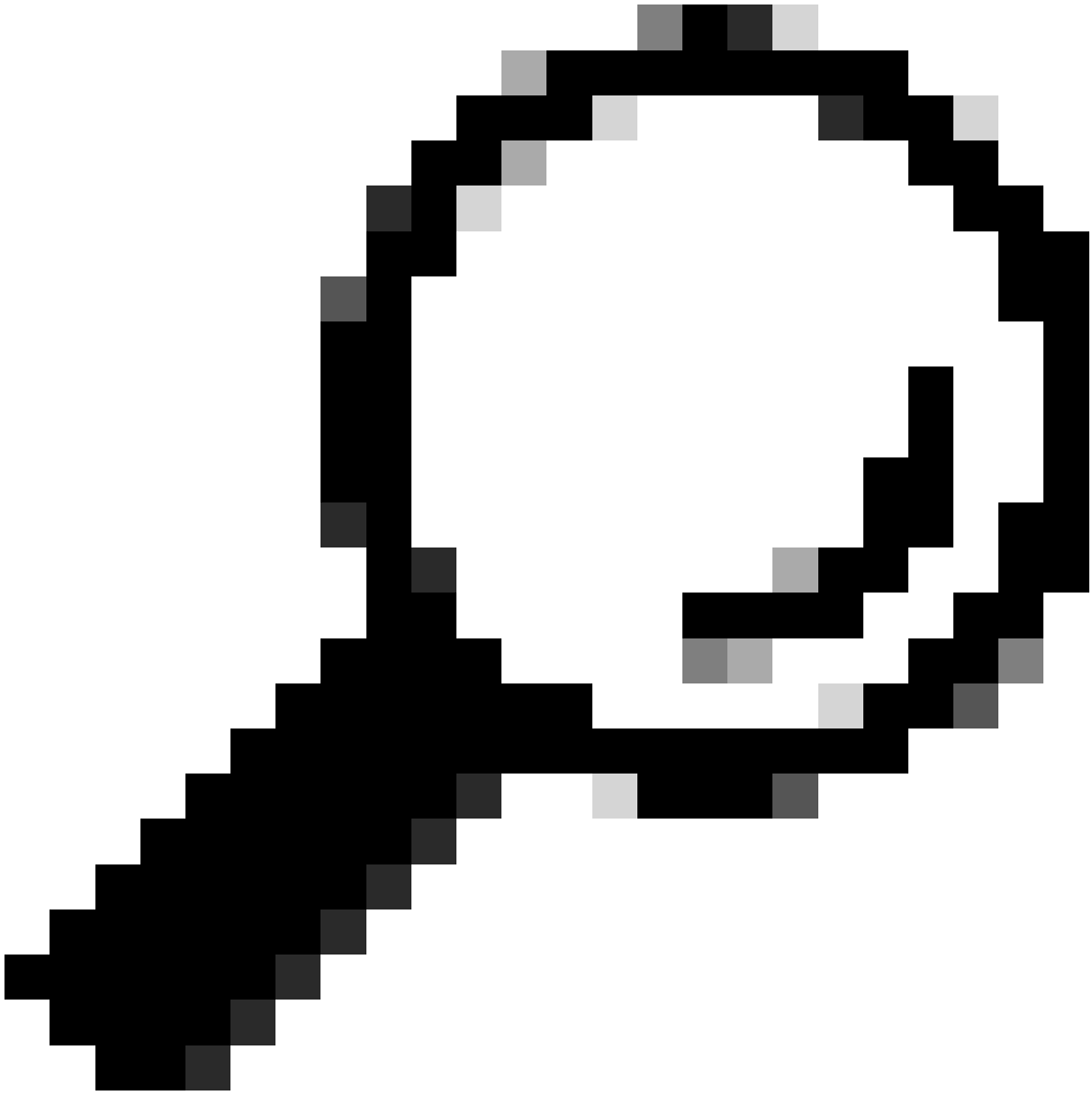
ステップ3：デフォルトのポート6049をそのまま使用するか、TACACS over TLSに別のTCPポートを指定して、Saveをクリックします。

ネットワークデバイスとネットワークデバイスグループの作成

ISEは、複数のデバイスグループ階層を使用した強力なデバイスグループ化を提供します。各階層は、ネットワークデバイスの個別の独立した分類を表します。

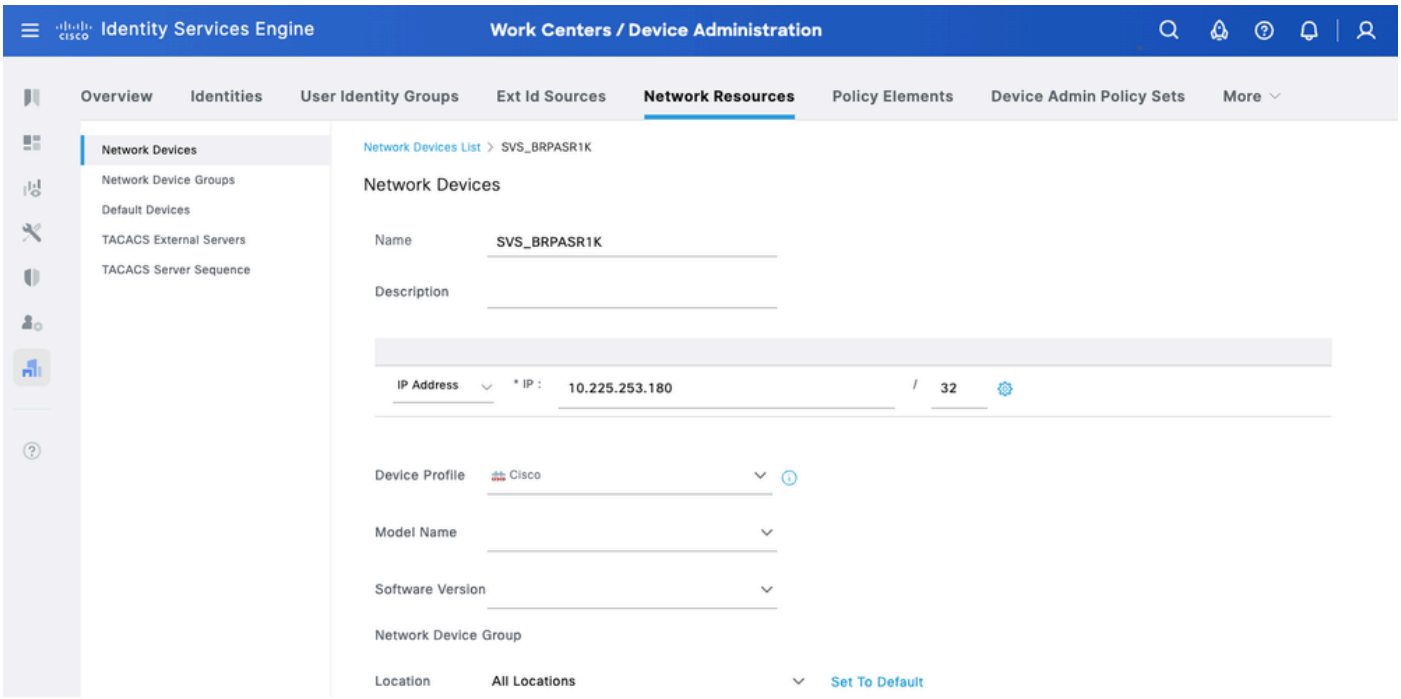
ステップ 1： Work Centers > Device Administration > Network Resourcesの順に移動し、Network Device Groupsをクリックして、IOS XEという名前のグループを作成します。


Identity Services Engine
Overview
Identities
Network Devices
Network Device Groups
Default Devices
TACACS External Servers
TACACS Server Sequence
Add Group
Name*
IOSXE
Description
Parent Group*
Select Group or Add as root group
Cancel
Save
IOSXR

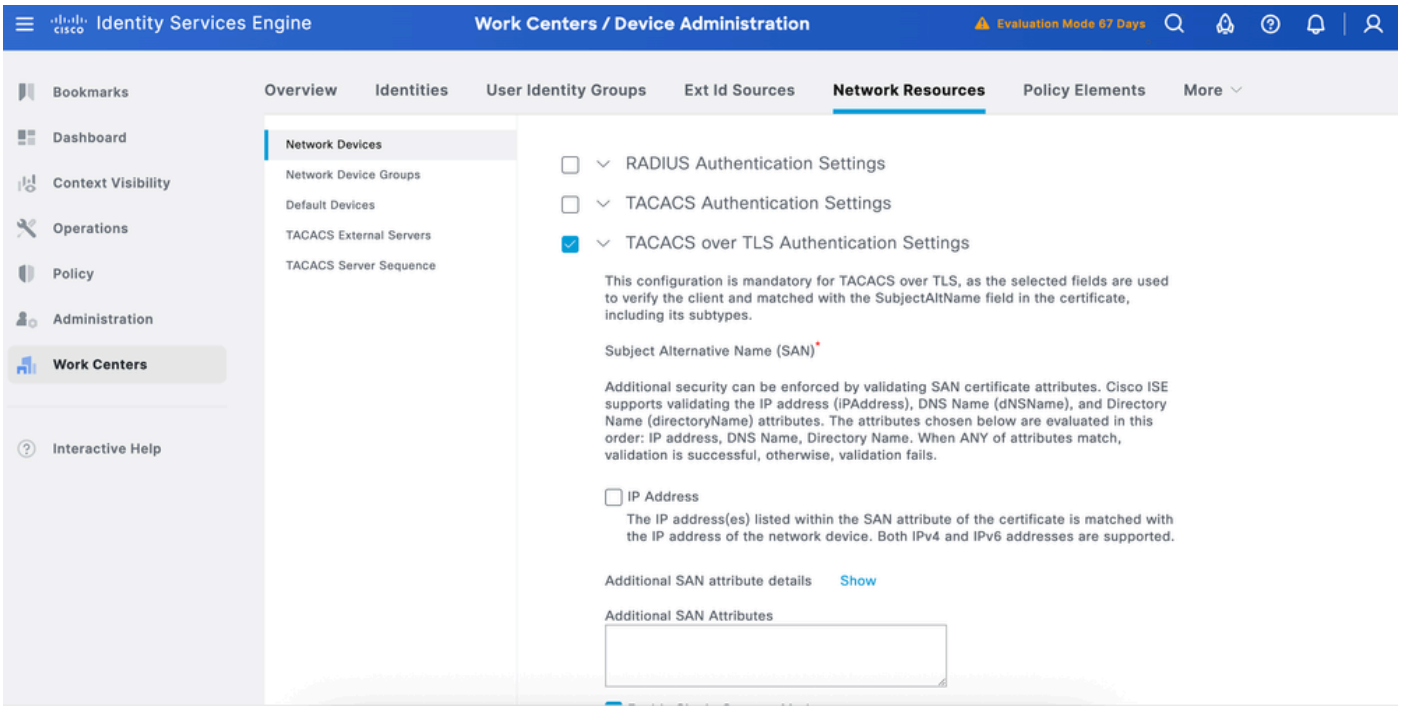


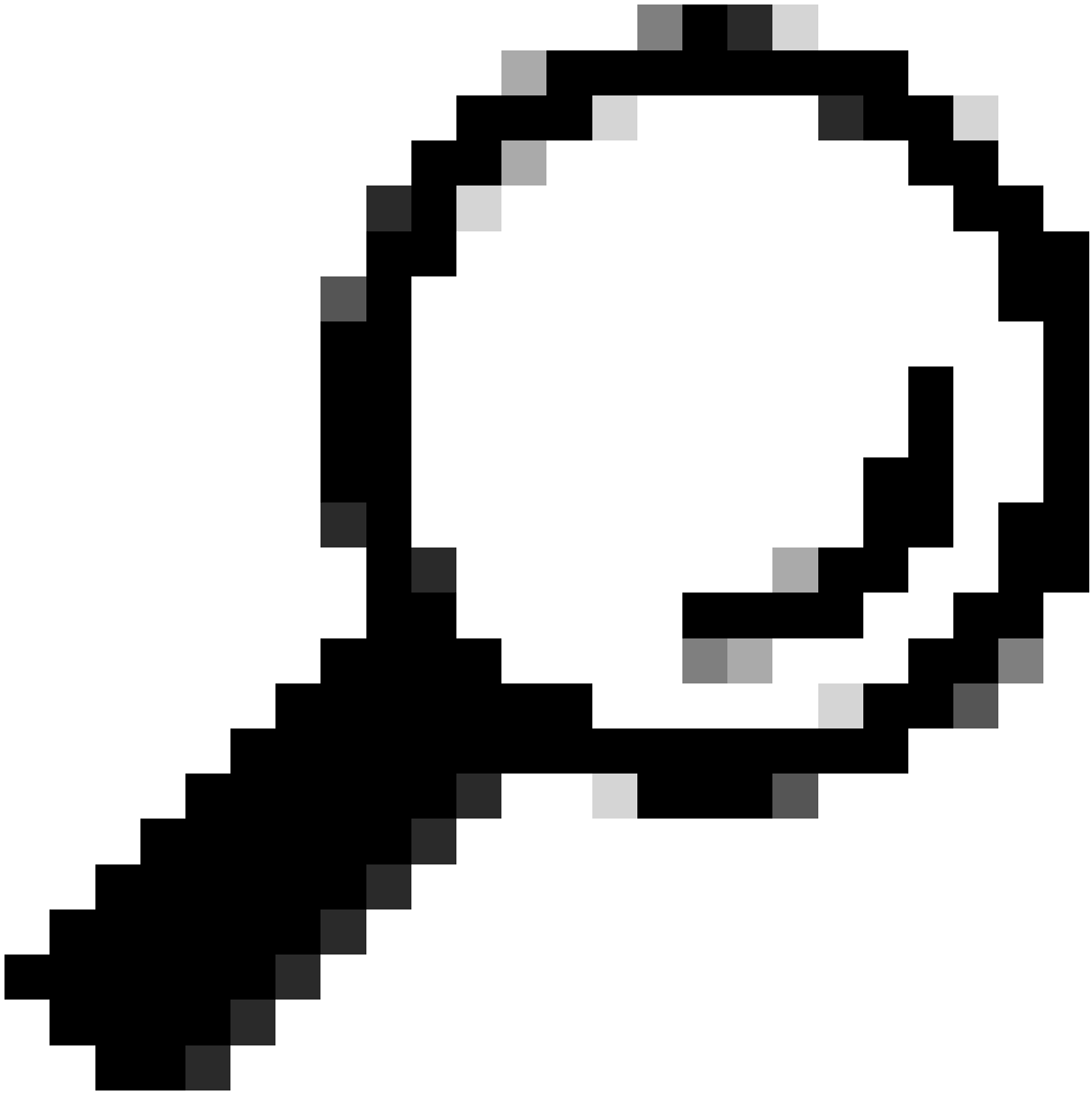
ヒント：すべてのデバイスタイプとすべての口ケーションは、ISEが提供するデフォルトの階層です。独自の階層を追加し、ポリシー条件で後から使用できるネットワークデバイスの識別でさまざまなコンポーネントを定義できます

ステップ2：次に、Cisco IOS XEデバイスをネットワークデバイスとして追加します。[Work Centers] > [Device Administration] > [Network Resources] > [Network Devices] に移動します。Addをクリックして、新しいネットワークデバイスを追加します。このテストでは、SVS_BRPASR1Kとなります。



ステップ3：デバイスのIPアドレスを入力し、デバイスの場所とデバイスタイプ(IOS XE)を必ずマッピングします。最後に、TACACS+ over TLS認証設定を有効にします。



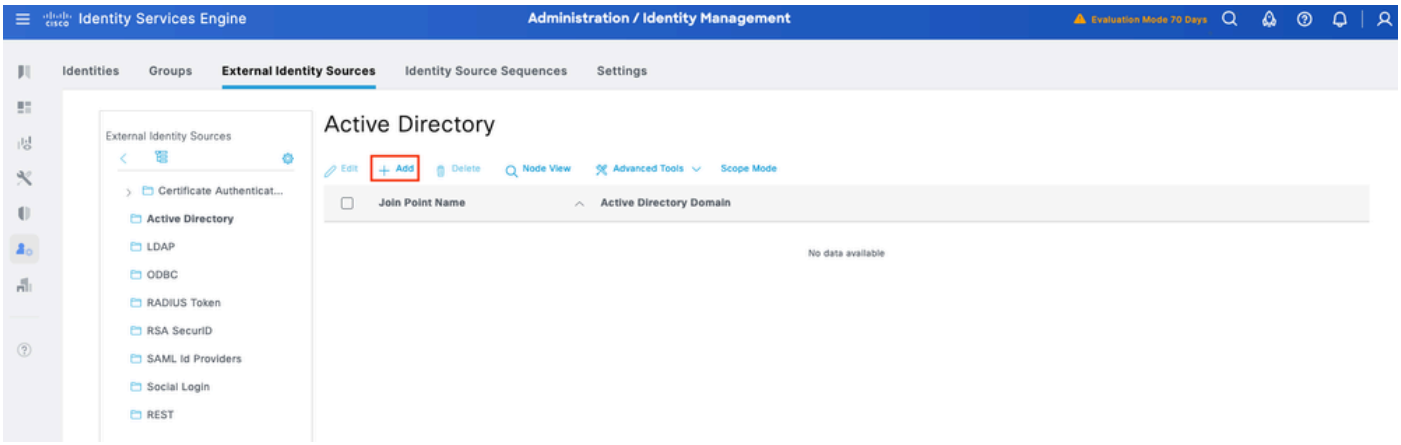


ヒント：デバイスにコマンドが送信されるたびにTCPセッションが再起動しないように、シングルコネクトモードを有効にすることを推奨します。

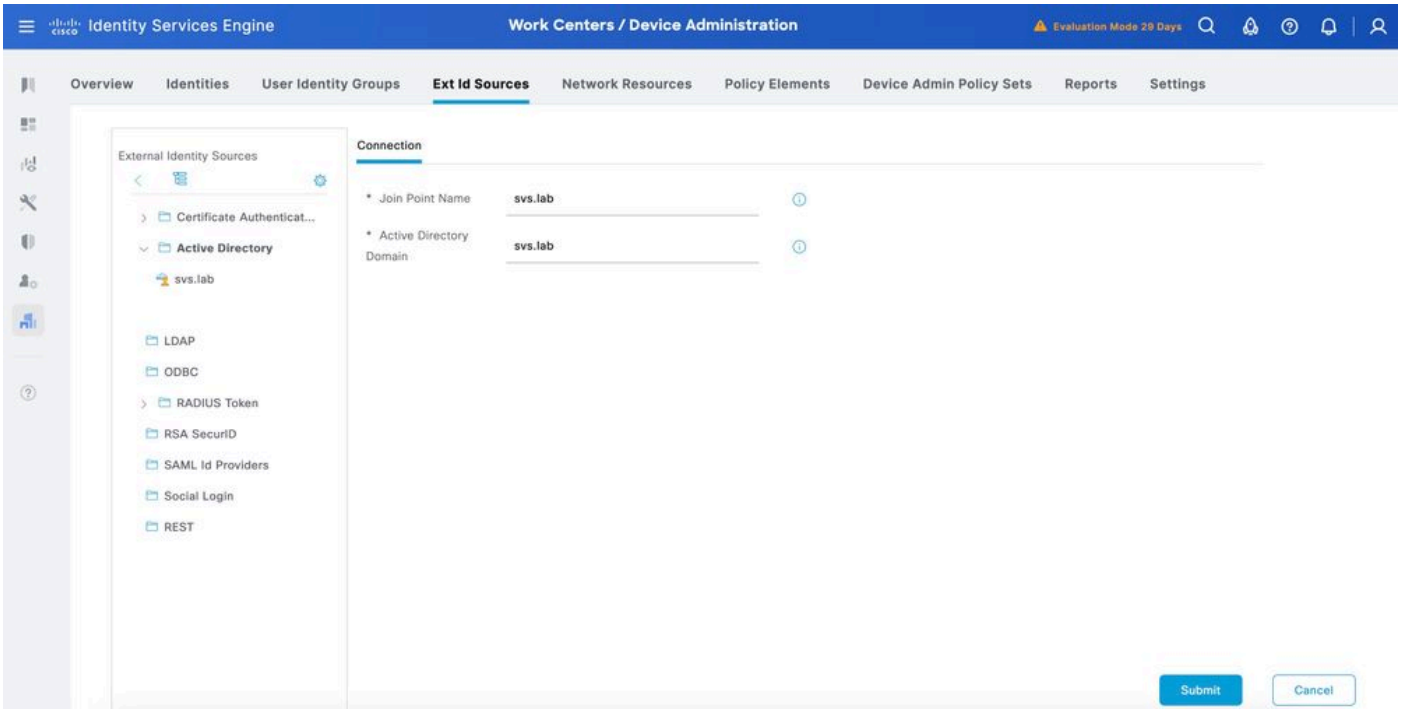
アイデンティティストアの設定

このセクションでは、デバイス管理者用のアイデンティティストアを定義します。これは、ISE内部ユーザおよびサポートされる任意の外部アイデンティティソースにすることができます。ここでは、外部IDソースであるActive Directory(AD)を使用します。

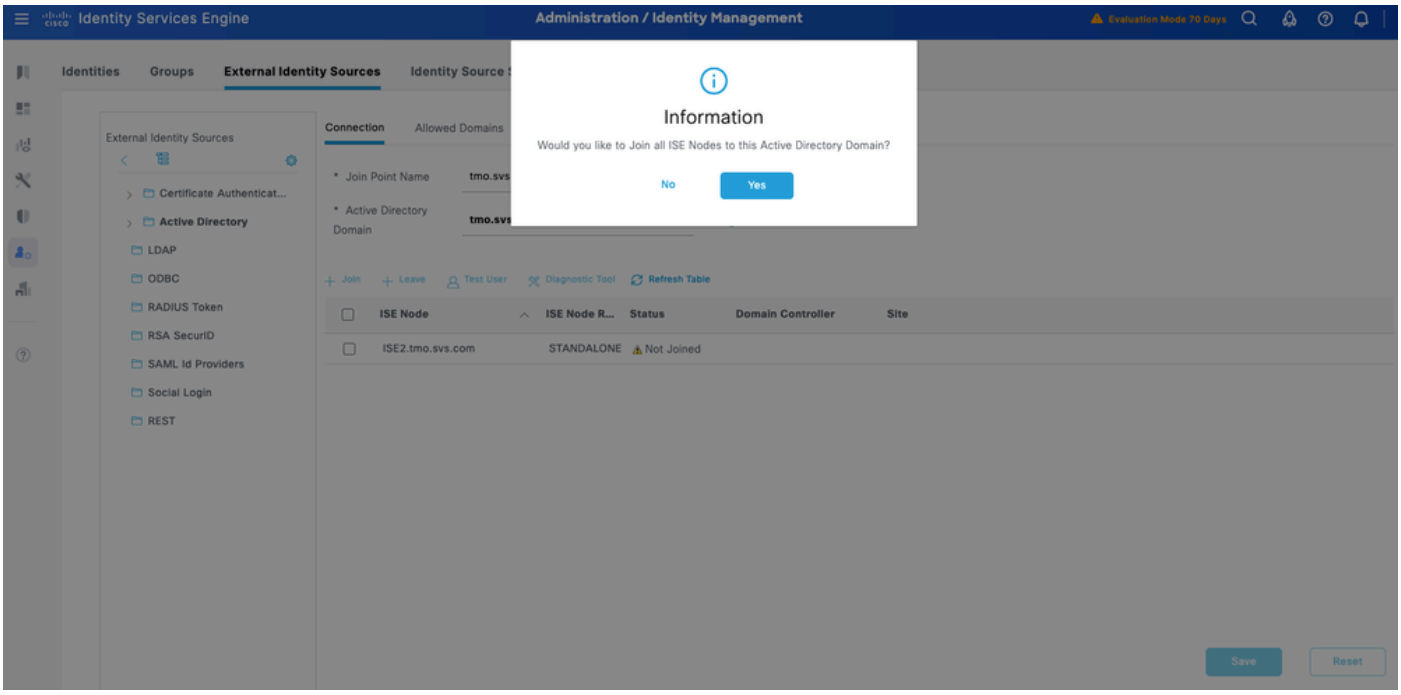
手順 1：Administration > Identity Management > External Identity Stores > Active Directoryの順に移動します。Addをクリックして、新しいADジョイントポイントを定義します。



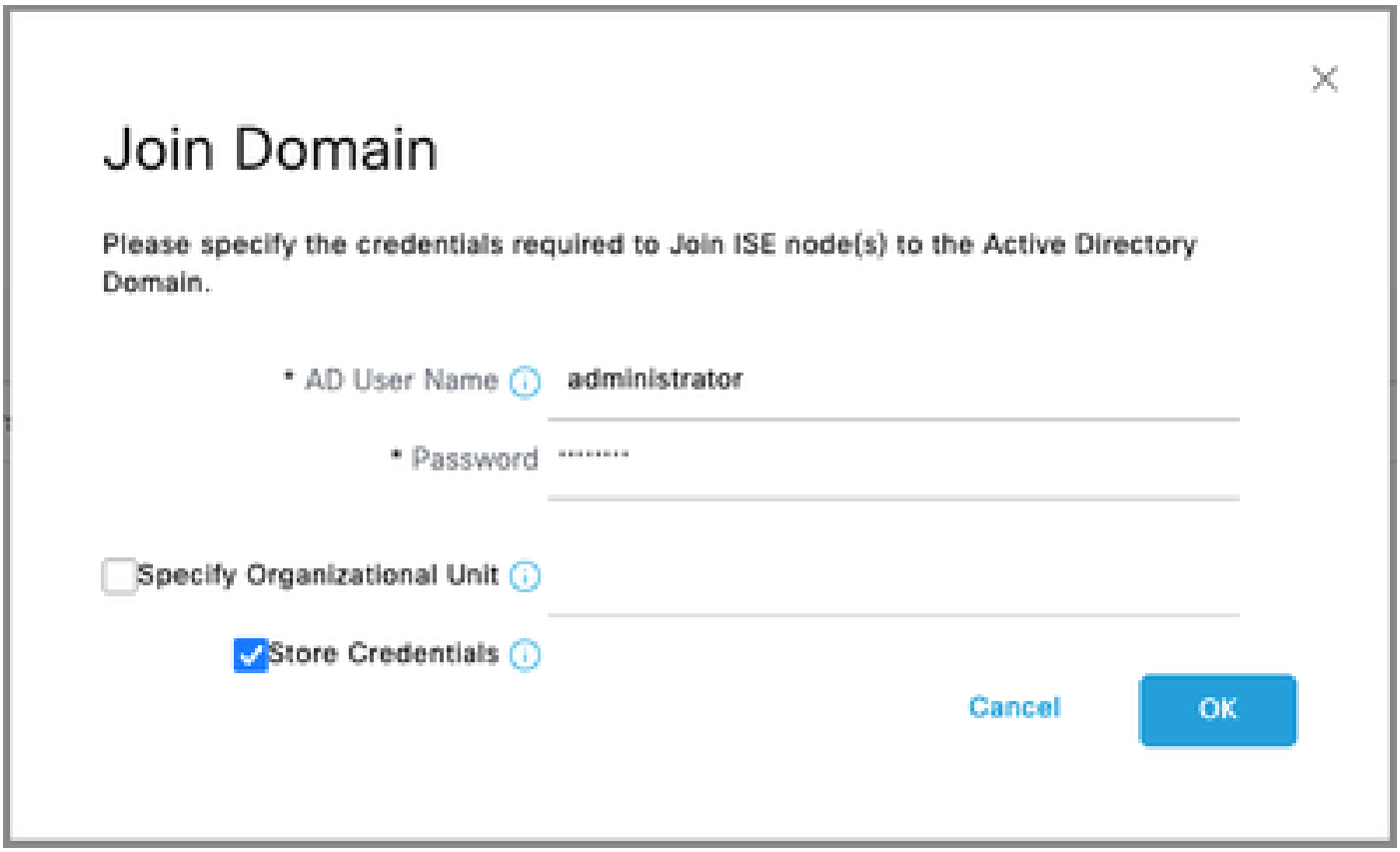
ステップ 2： 参加ポイント名とADドメイン名を指定し、Submitをクリックします。

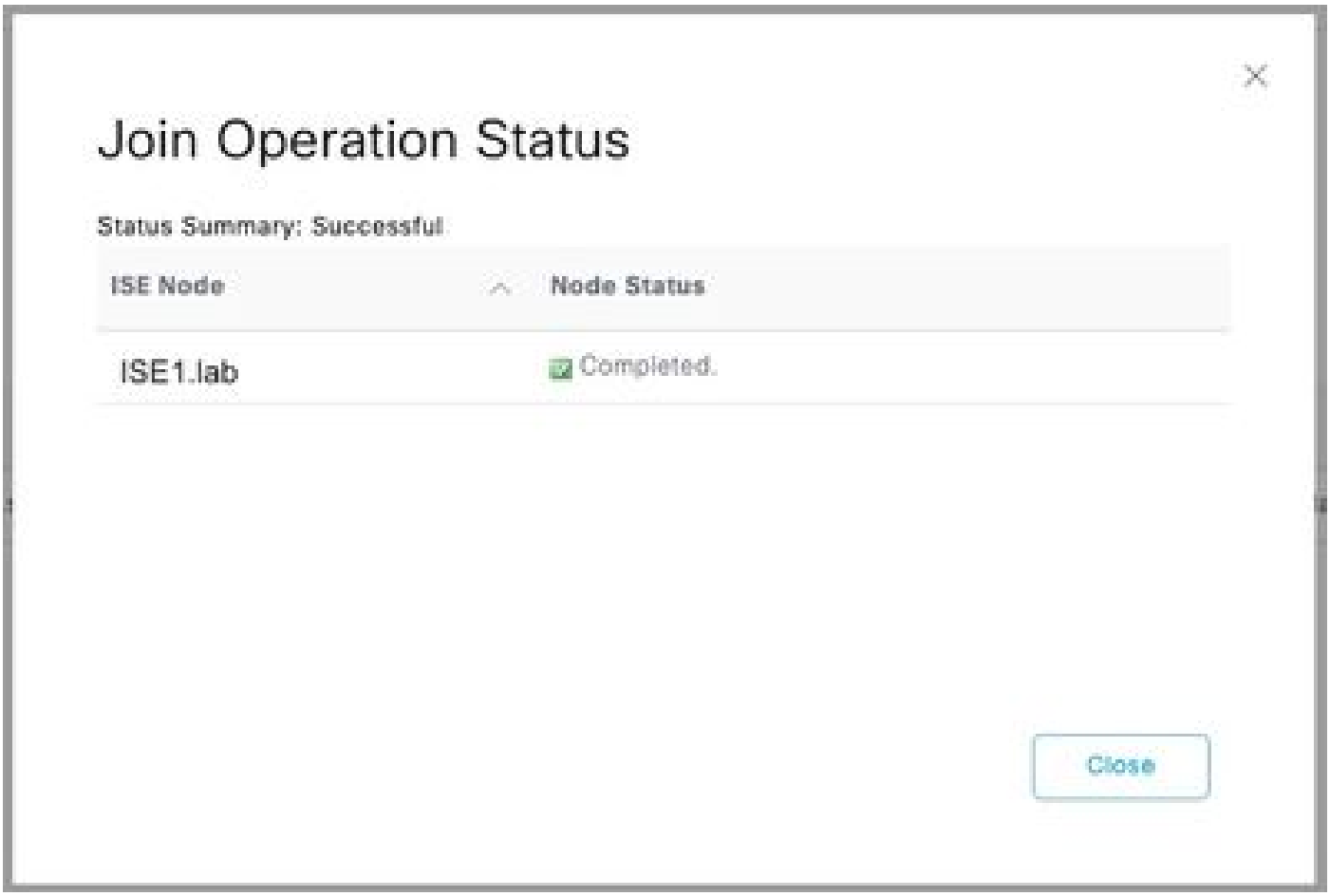


ステップ 3： Would you like to Join all ISE Nodes to this Active Directory Domain?プロンプトが表示されたら、Yesをクリックします。

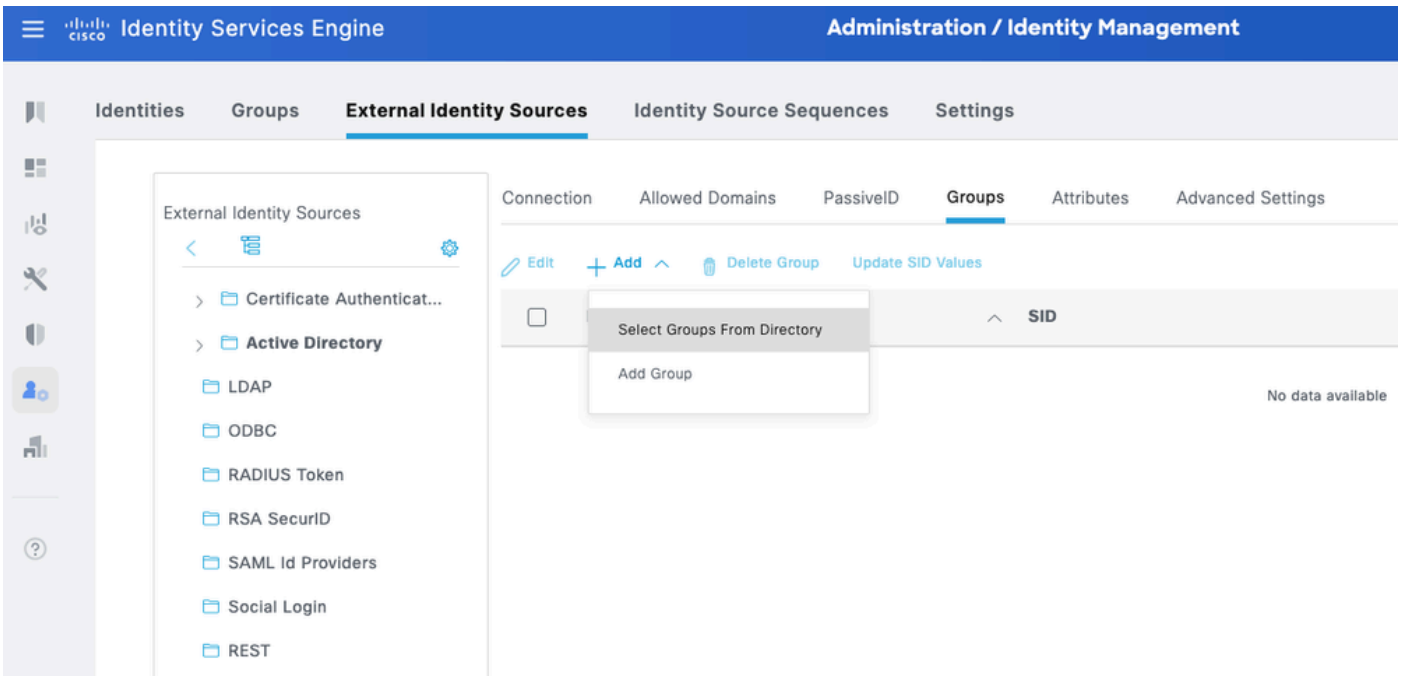


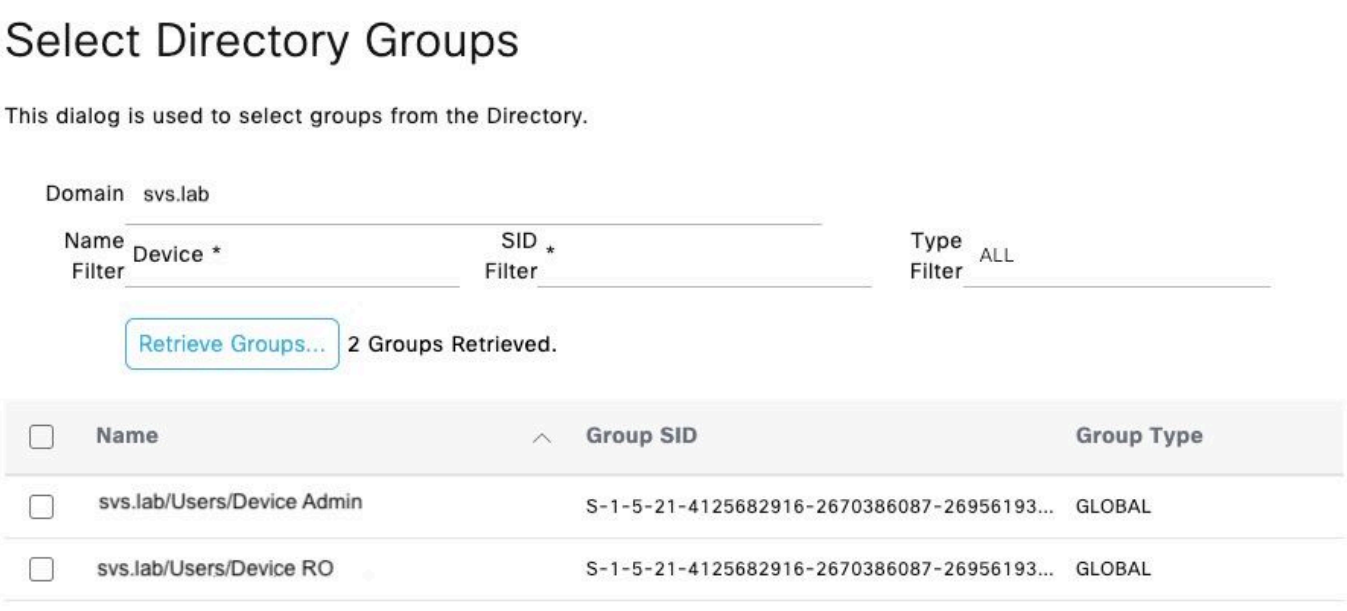
ステップ 4： AD参加権限を持つクレデンシャルを入力し、ISEをADに参加させます。ステータスをチェックして、動作していることを確認します。

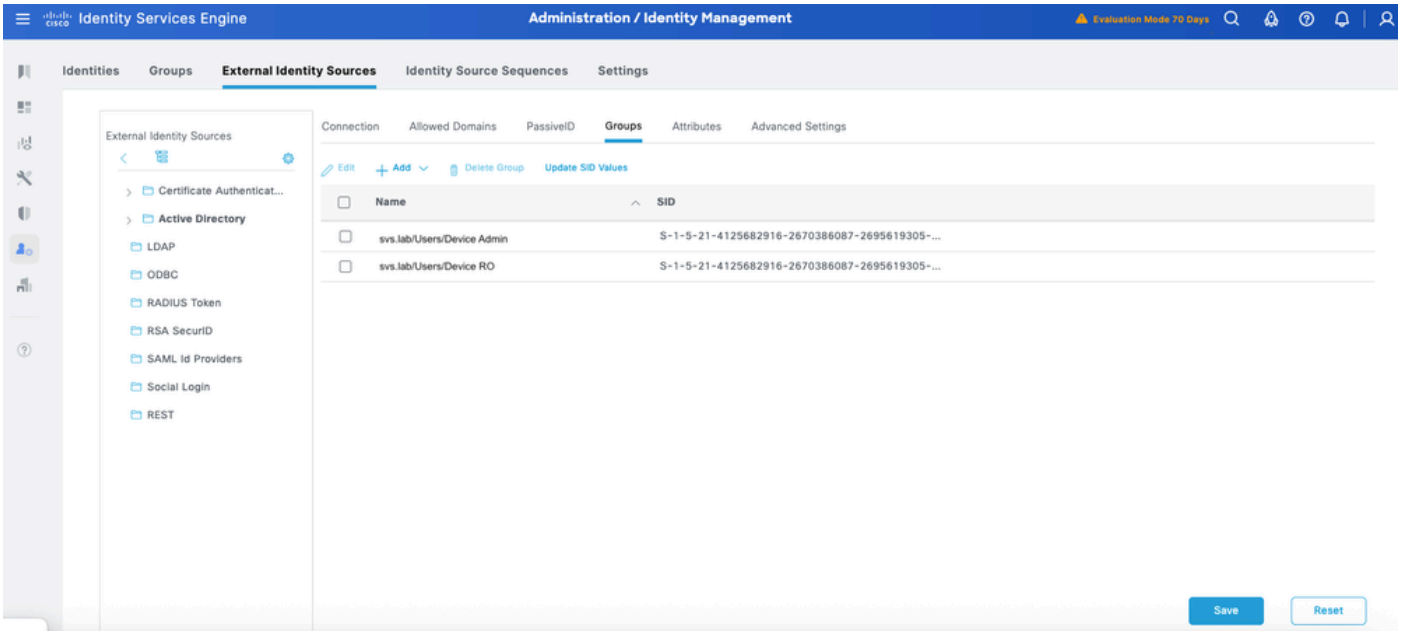




ステップ 5： Groupsタブに移動し、Addをクリックして、デバイスへのアクセスが許可されているユーザに基づいて、必要なすべてのグループを取得します。この例では、このガイドの認可ポリシーで使用されるグループを示します







TACACS+プロファイルの設定

TACACS+プロファイルをCisco IOS XEデバイスの2つの主要なユーザロールにマッピングします。

- ルートシステム管理者 – これは、デバイス内で最も高い権限を持つロールです。root system管理者の役割を持つユーザーは、すべてのシステムコマンドと構成機能に対する完全な管理アクセス権を持ちます。
- オペレータ：このロールは、監視およびトラブルシューティングのためにシステムへの読み取り専用アクセスを必要とするユーザを対象としています。

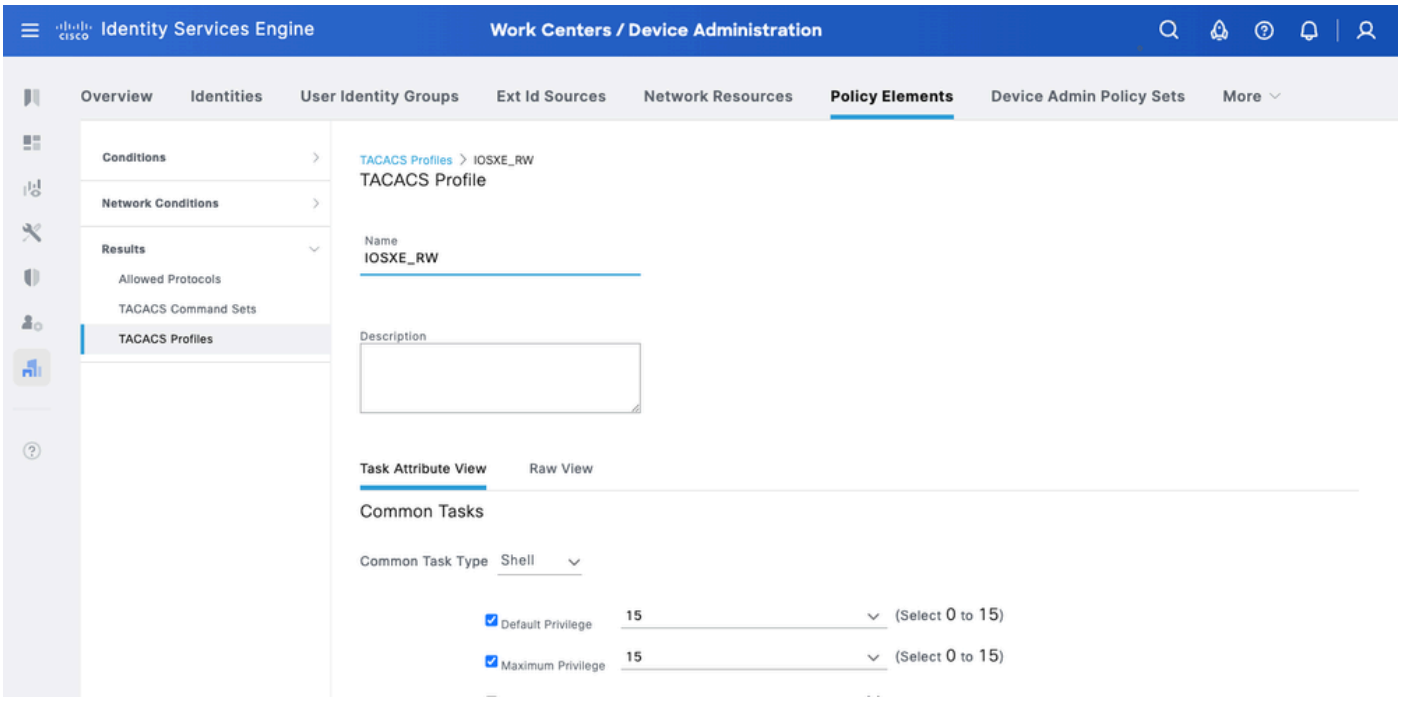
これらは、IOS XE_RWとIOSXR_ROの2つのTACACS+プロファイルとして定義されています。

IOS XE_RW – 管理者プロファイル

ステップ1:Work Centers > Device Administration > Policy Elements > Results > TACACS Profilesの順に移動します。 新しいTACACSプロファイルを追加し、IOS XE_RWという名前を付けます。

ステップ 2： デフォルト権限と最大権限を15に設定します。

ステップ3：設定を確認し、保存します。

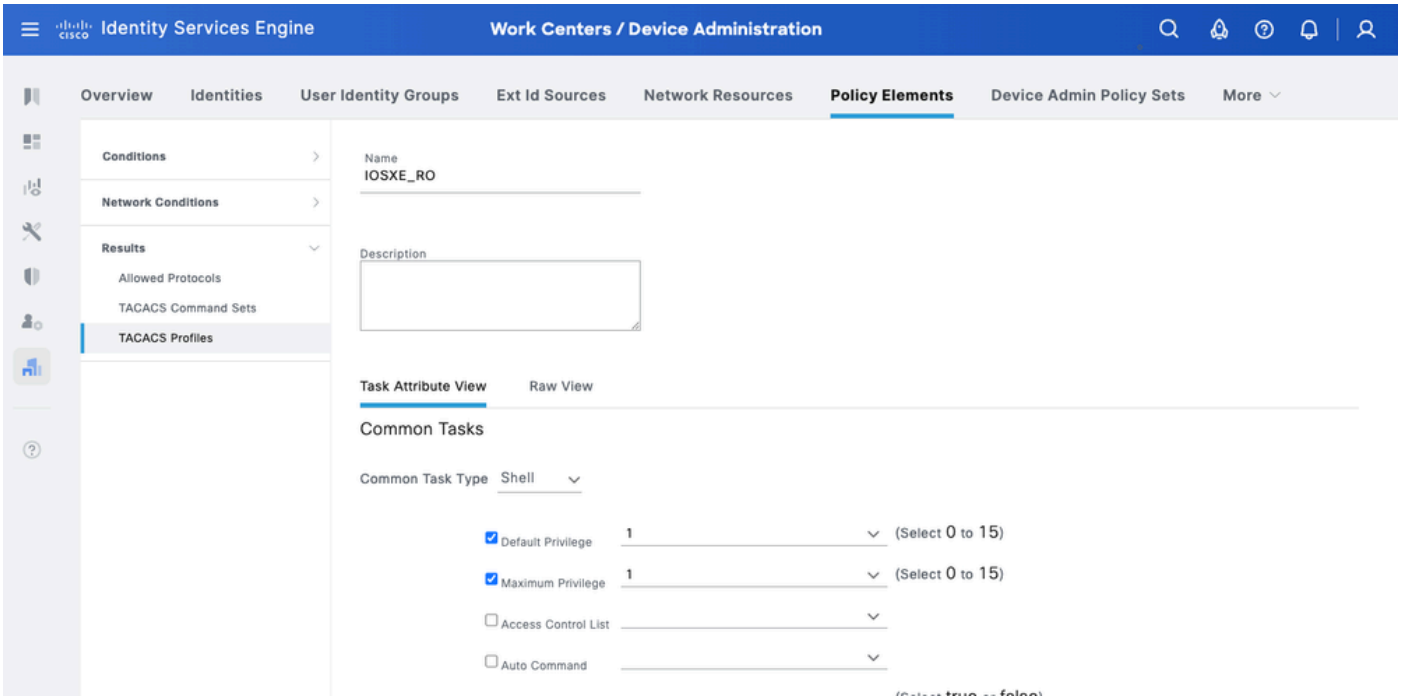


IOS XE_RO – オペレータプロファイル

ステップ1:Work Centers > Device Administration > Policy Elements > Results > TACACS Profilesの順に移動します。 新しいTACACSプロファイルを追加し、IOS XE_ROという名前を付けます。

ステップ 2： デフォルト権限と最大権限を1に設定します。

ステップ3：設定を確認し、保存します。



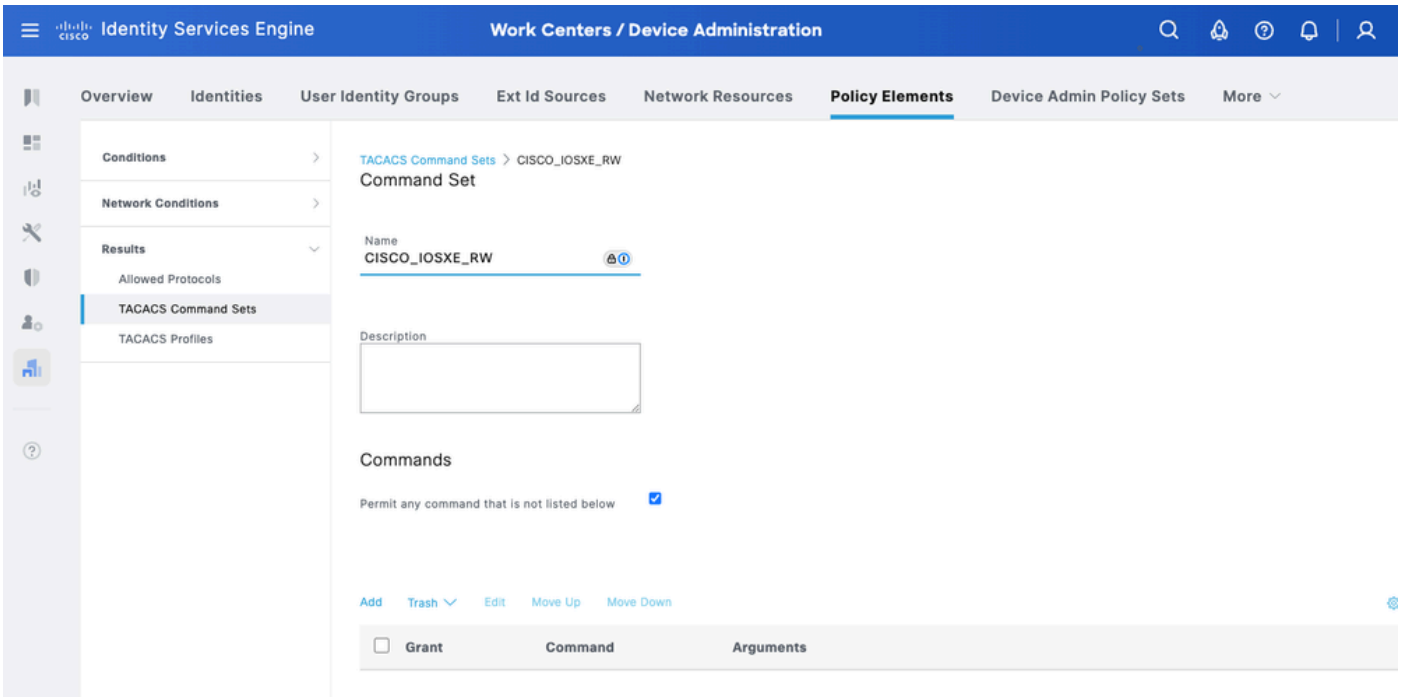
ConfigureTACACS+コマンドセット

これらは、CISCO_IOS XE_RWとCISCO_IOS XE_ROという2つのTACACS+コマンドセットとして定義されています。

CISCO_IOS XE_RW：管理者コマンドセット

ステップ 1： Work Centers > Device Administration > Policy Elements > Results > TACACS Command Setsの順に移動します。 新しいTACACSコマンドセットを追加し、CISCO_IOS XE_RWという名前を付けます。

ステップ 2： Permit any command that is not listed below チェックボックスにチェックマークを付け（これにより、管理者ロールのすべてのコマンドが許可されます）、Saveをクリックします。



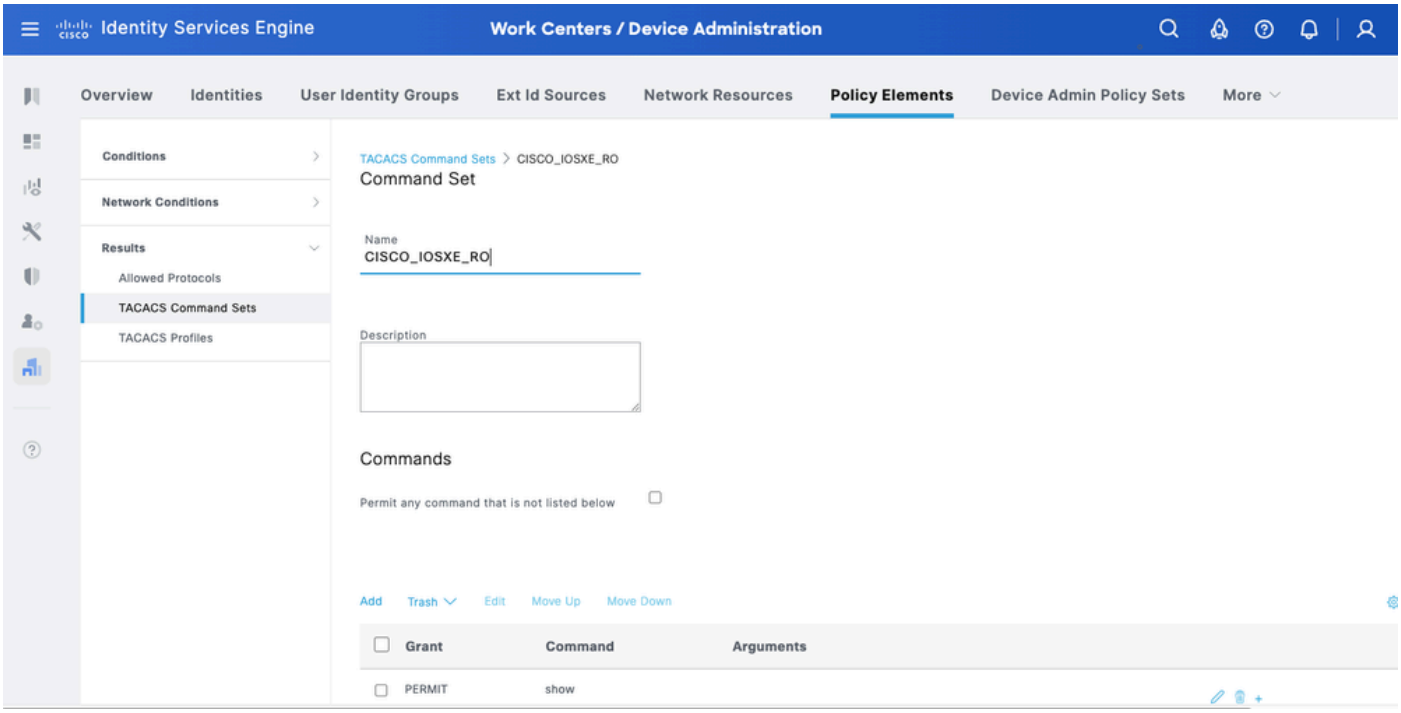
CISCO_IOS XE_RO：オペレータコマンドセット

ステップ1 ISE UIから、Work Centers > Device Administration > Policy Elements > Results > TACACS Command Setsの順に移動します。新しいTACACSコマンドセットを追加し、CISCO_IOS XE_ROという名前を付けます。

ステップ 2： コマンドセクションで、新しいコマンドを追加します。

ステップ 3： [Grant]列のドロップダウンリストから[Permit] を選択し、[Command]列にshowと入力して、チェック矢印をクリックします。

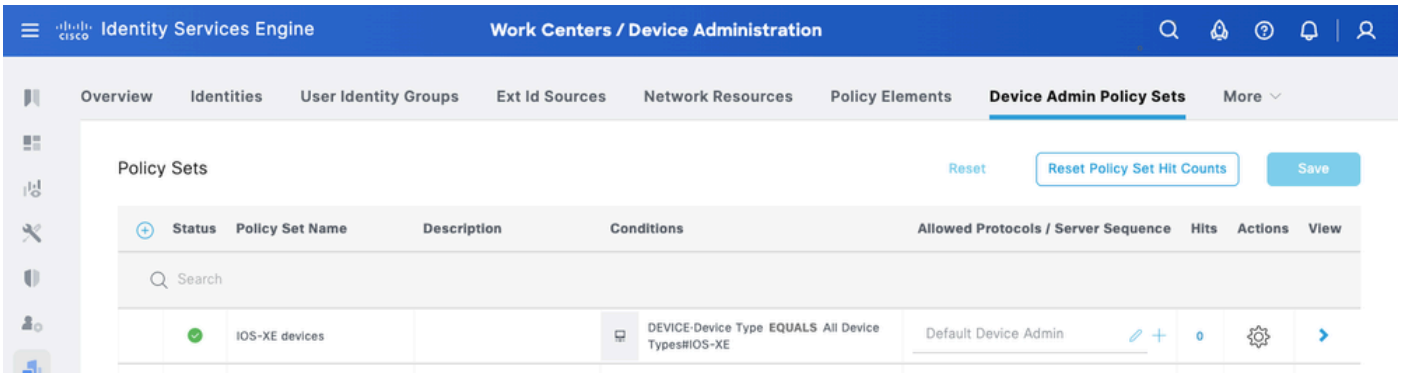
ステップ 4： データを確認して、Saveをクリックします。



デバイス管理ポリシーセットの設定

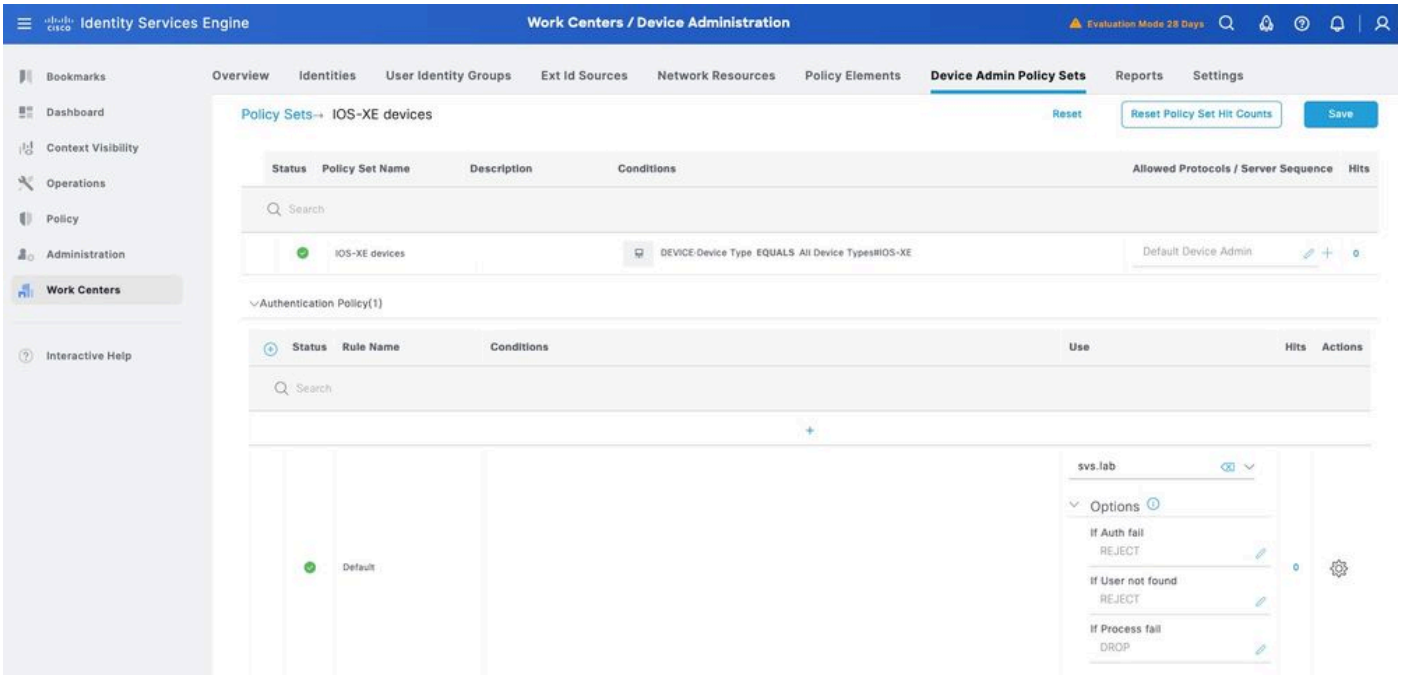
デバイス管理では、ポリシーセットはデフォルトで有効になっています。ポリシーセットは、デバイスタイプに基づいてポリシーを分割できるため、TACACSプロファイルの適用が容易になります。

ステップ1:Work Centers > Device Administration > Device Admin Policy Setsの順に移動します。新しいポリシーセットIOS XEデバイスを追加します。条件でDEVICE:Device Type EQUALS All Device Types#IOS XEを指定します。Allowed Protocolsの下で、Default Device Adminを選択します。



ステップ2:Saveをクリックし、右矢印をクリックしてこのポリシーセットを設定します。

ステップ3:認証ポリシーの作成。認証用に、ID StoreとしてADを使用します。If Auth fail、If User not found、およびIf Process failの下にデフォルトオプションのままにします。



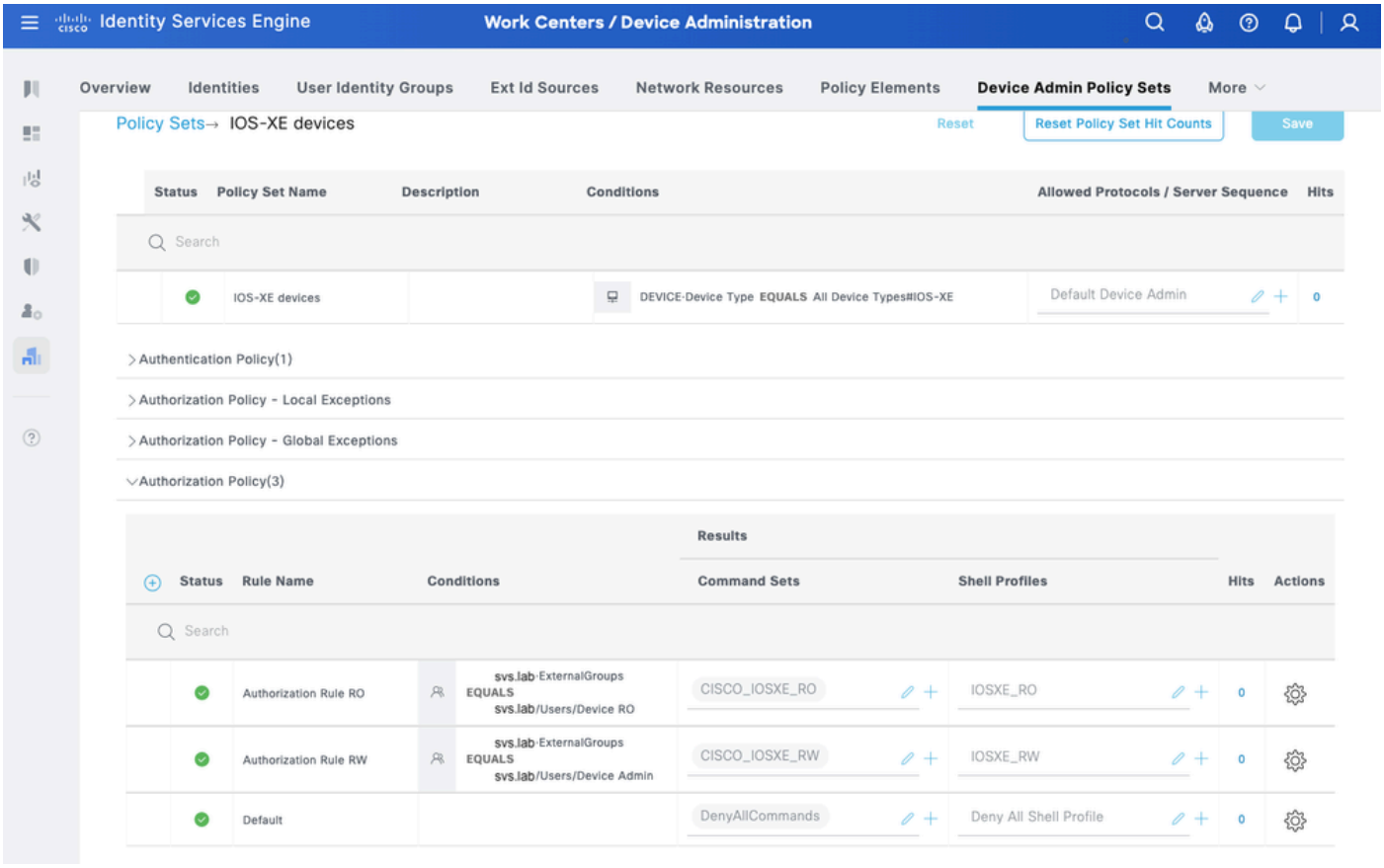
ステップ 4： 許可ポリシーを定義します。

Active Directory(AD)のユーザグループに基づいて許可ポリシーを作成します。

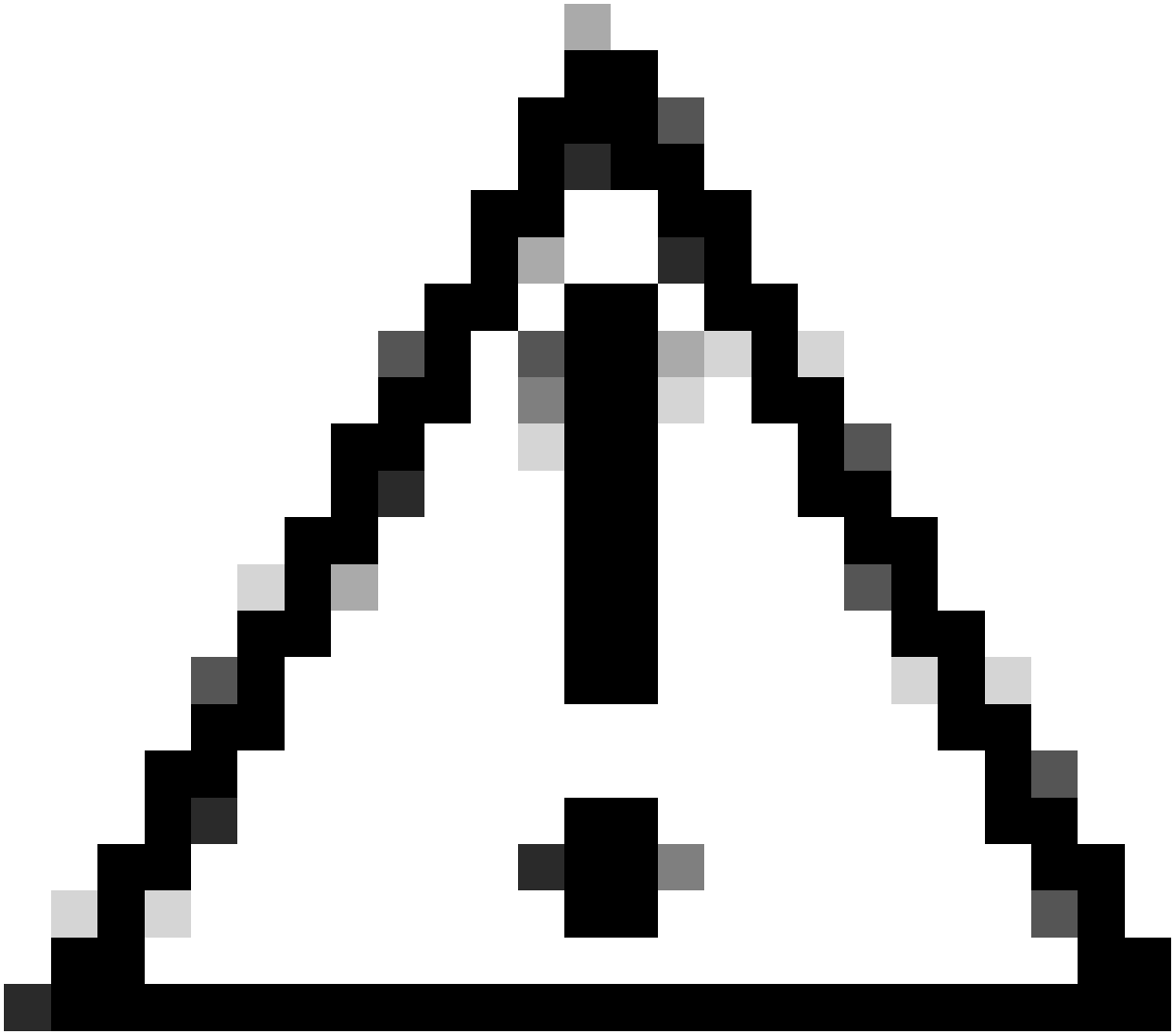
例：

・ ADグループDevice RO のユーザには、CISCO_IOSXR_RO コマンドセットとIOSXR_RO シェルプロファイルが割り当てられます。

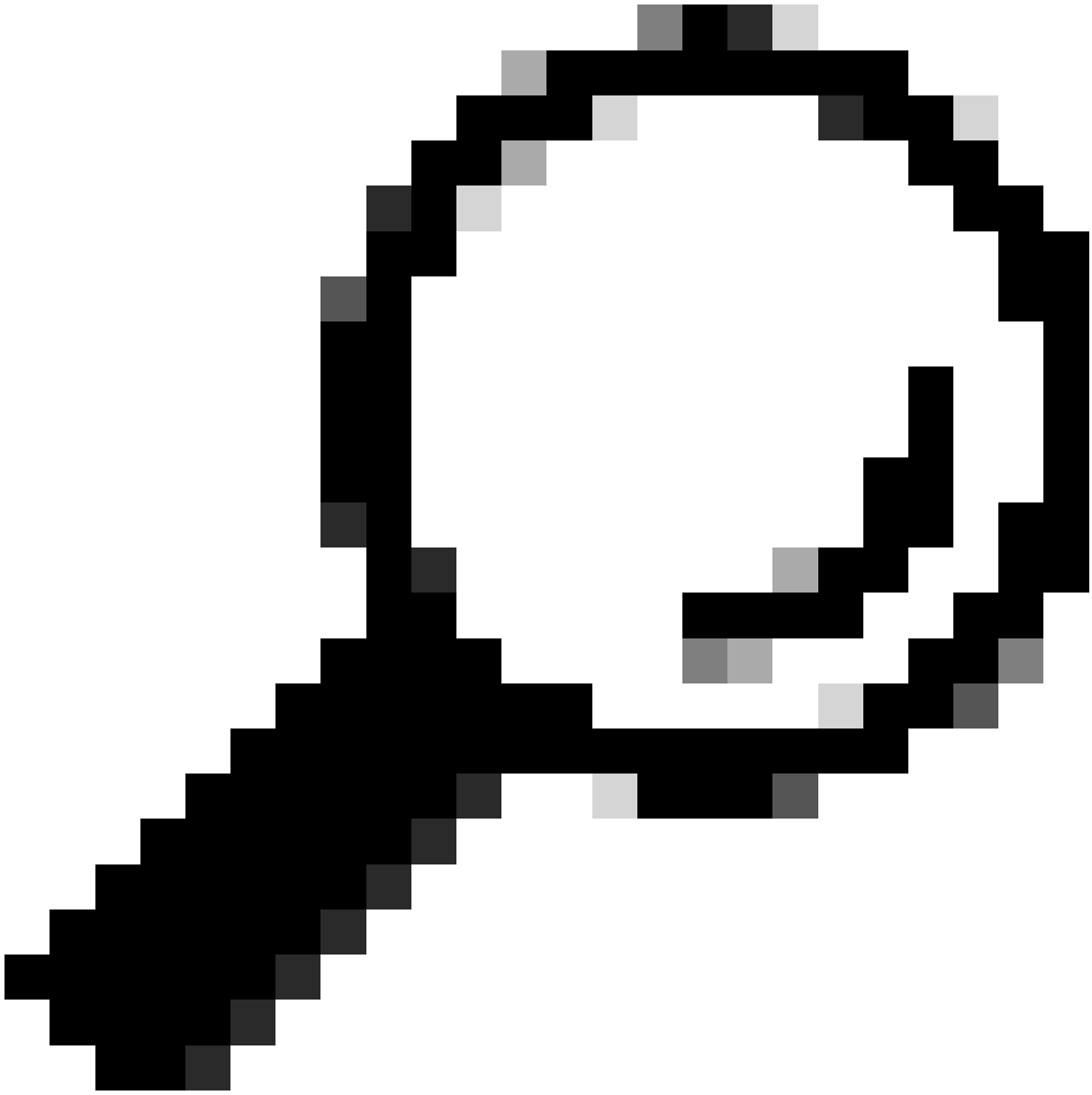
・ ADグループDevice Admin のユーザには、CISCO_IOSXR_RW コマンドセットとIOSXR_RW シェルプロファイルが割り当てられます。



パート2:TACACS+ over TLS 1.3用のCisco IOS XEの設定



注意：コンソール接続が到達可能で、正しく機能していることを確認してください。



ヒント：一時的なユーザを設定し、AAAの認証および認可方式を変更して、設定の変更時にTACACSの代わりにローカルクレデンシャルを使用することで、デバイスからロックアウトされないようにすることをお勧めします。

設定方法1：デバイスが生成したキーペア

TACACS+サーバの設定

手順1 ドメイン名を設定し、ルータのトラストポイントに使用するキーペアを生成します。

```
ip domain name svs.lab

crypto key generate ec keysize 256 label svs-256ec-key
```

トラストポイントの設定

手順1 ルータのトラストポイントを作成し、キーペアを関連付けます。

```
crypto pki trustpoint svs_cat9k
  enrollment terminal pem
  subject-name C=US,ST=NC,L=RTP,O=Cisco,OU=SVS,CN=cat9k.svs.lab
  serial-number none
  ip-address none
  revocation-check none
  eckeypair svs-256ec-key
```

ステップ 2： CA証明書をインストールしてトラストポイントを認証します。

<#root>

cat9k(config)#

crypto pki authenticate svs_cat9k

Enter the base 64 encoded CA certificate.

End with a blank line or the word "quit" on a line by itself

-----BEGIN CERTIFICATE-----

MIIFlDCCA3ygAwIBAgIIIM1OAsTaN/UwDQYJKoZIhvcNAQELBQAwajELMAkGA1UE
BhMCVVMxFzAVBgNVBAgTDk5vcnRoIENhcm9saW5hMRAwDgYDVQQHEwdSYWxlaWdo
MQ4wDAYDVQQKEwVDaXNjbzEMMAoGA1UECxMDU1ZTMRIwEAYDVQQDEwlTVlMgTGFi
QOEwHhcNMjUwNDI4MTcwNTAwWhcNMzUwNDI4MTcwNTAwWjBqMQswCQYDVQQGEwJV
UzEXMBUGA1UECBMOTm9ydGggQ2Fyb2xpbmExEDAOBgNVBAcTB1JhbGVpZ2gxDjAM
BgNVBAoTBUNpc2NvMQwwCgYDVQQLEwNTVlMxEjAQBgNVBAMTCVNWUyBMYWJDQTCC
AiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBAJvZUOyn2vIn6gKbx3M7vaRq
2YjwZlzSH6EkEvxnJTy+kksiFD33GyHQepk7vfp4NFU5OtQ4HC7t/AOv9grDa3QW
VwvV4MBbJhFM3sOJ/ejgDYcMZhIAaPyOZo5WLboOkXEiKjPLatkXojB8FVrhLF3O
jMBSqwa4/Wlniy5S+7s4FFxsCf2OCOWfBAsnrs0tatIIhmcnx+VLJP7MRm8fOw4m
mutNo7IhbJSrgAFXmjlbBjMmgspObULo/wxMHdTbtPBf11HRHTkNIo3qyO4UADL2
WpoGhgT/FaxxBo2UBcnYVaP+jjRE0NYT973MCbVAAxtNVU6bEBROz+LWniACzupm
+qh23SL43uW5A3iSw/BuU1E9p7BOe8oDNKU6gXlojKyLP/gC7j8AePO3ir+KZui8
b8X4iYn/67SbzZFhwxn3chkW4JYhQ4AImW1An2Q1+DMoZL7zRtSqQ3g9ZqRIMzQN
gJ+kQXe7QtT/u6m1MrtjE3gAEVpL334rTIxy9hpKZIkB86t2ZA3JX8CLsbCa13sA
z1XCoONX+6a1ekmXuAOI+t3c1sNbN2AtFi4cJovTAO1xh6OI4QnK+MNQKpTjt/E4
ydHlOrrurXsZummj9QBnkX4pqY7cDLHhdMKpbjDwg7jVLl783nTc9wYptQEPi5sw

```
83g9EMgKV0ARIiVUa/qlAgMBAAGjPjA8MAwGA1UdEwQFMAMBAf8wEQYJYIZIAYb4
QgEBBAQDAgAHMBkGCWCGSAGG+EIBDQQMFgpTVlMgTGFiIENBMA0GCSqGSIb3DQEB
CwUAA4ICAQAIT3O8oL2L6j/7Kk9VdcouuaBsN9o2pNEk3KXeZ8ykarNoxa87sFYr
AwXIwfAtk8uEHfnWu1QcZ3LkEJM9rHVCZuKsYd3D6qojo54HTpxRLgo5oKOdGayi
iSEkSSX9qyfLfINHR2JSVqJU6jLsy86X7q7RmIPMS7XfHzuddFNI4YDoXRX67X+v
O+ja6zTQqjO6lqJhmrSkyFbYf/ZTpe4d1OzJsZjNsNOr8bF9nOA/7qNZLp3Z3cpU
PUOKdbiSvRqnPw3e8TfITVmAzcx8COI2SrYFMSUazolVBvDy+xRKxyAtMbneGz6n
YdykCimThCKoKwp/pWpYBEqIEOf5ay1PKURO/8aj/B7aluJapXkmnj5qPeGhNOpB
Q9r14reov4so2EspkXS7CrH9yGfpIyTprokz1UvZBZ8vloI7YZmjFmem+5rT6Gnk
eU/lX7nV61SYG5W5K+I8uaKuyBHOMn7Amy3DYL5c5GJBqxpSZERbLXV+Q1tIgRU8
8ggzlPOdsS/i6Lo7ypYXOeB9HgVDCkzQsLXQuHGj/2WsgPgdRcjkvnyURk4Jx+Ib
xDrmo7eOXPpSW4l72a6Kl8CR3U2Cr4wsuvndPEq/qd2NRSBWffF0XE/AJHQG7STT
HaXLU9r2Ko6O3oecu8ysGTwLlIt/9T1/FOb0xZRugWcpJrVoTgDGuA==

-----END CERTIFICATE-----

Certificate has the following attributes:
   Fingerprint MD5: D9C404B2 EC08A260 EC3539E7 F54ED17D
   Fingerprint SHA1: 0EB181E9 5A3ED780 3BC5A805 9A854A95 C83AC737

% Do you accept this certificate? [yes/no]:

yes


Trustpoint CA certificate accepted.
% Certificate successfully imported

cat9k(config)#
```

ステップ 3： 証明書署名要求(CSR)を生成します。

<#root>

```
cat9k(config)#

crypto pki enroll svs_cat9k


% Start certificate enrollment ..

% The subject name in the certificate will include: C=US,ST=NC,L=RTP,O=Cisco,OU=SVS,CN=cat9k.svs.lab
% The subject name in the certificate will include: cat9k.svs.lab
Display Certificate Request to terminal? [yes/no]:

yes


Certificate Request follows:

-----BEGIN CERTIFICATE REQUEST-----
MIIBfDCCASMCAQAwgYQxGjAYBgNVBAMTEWNhdDlrLnRtby5zdnMuY29tMQwwCgYD
VQQLEwNTVlMxDjAMBgNVBAoTBUNpc2NvMQwwCgYDVQQHEwNSVFAxCzAJBgNVBAgT
Ak5DMQswCQYDVQQGEwJVUzEgMB4GCSqGSIb3DQEJAhYRY2F0OWsudG1vLnN2cy5j
b20wWTATBgcqhkjOPQIBBggqhkjOPQMBBwNCAATpYE7atscrtl4ddevCh3UgxjYi
4N4oBGWrpJBctKy4so8V5i6RXDt7kHgPzp14Qnf20bcXVODE1wtTAHHBrIXqoDww
OgYJKoZIhvcNAQkOMS0wKzAcBgNVHREEFTATghFjYXQ5ay50bW8uc3ZzLmNvbTAL
BgNVHQ8EBAMCB4AwCgYIKoZIzj0EAwQDRwAwRAIgZqP2QTwM3ZZrmIphJ7+jSTER
4OkTx2DiVs1c1Xf+vR4CIBcSb18DIYz84DmgMHUaf778/cmpe9cWakvdaxMWseBH
-----END CERTIFICATE REQUEST-----
```

```
---End - This line not part of the certificate request---

Redisplay enrollment request? [yes/no]:

no


cat9k(config)#
```

ステップ 4：CA署名付き証明書をインポートします。

```
<#root>

cat9k(config)#

crypto pki import svs_cat9k certificate



Enter the base 64 encoded certificate.
End with a blank line or the word "quit" on a line by itself

-----BEGIN CERTIFICATE-----
MIID8zCCAdugAwIBAgIIKfdYWg5WpskwDQYJKoZIhvcNAQELBQAwajELMAkGA1UE
BhMCVVMxFzAVBgNVBAgTDk5vcnRoIENhcm9saW5hMRAwDgYDVQQHEwdSYWxlaWdo
MQ4wDAYDVQQKEwVDaXNjbzEMMAoGA1UECxMDU1ZTMRIwEAYDVQQDEwlTVlMgTGFi
QOEwHhcNMjUwNTE0MTUxMjAwWhcNMjYwNTE0MTUxMjAwWjCBhDEaMBgGA1UEAxMR
Y2FOOWsudG1vLnN2cy5jb2OxDDAKBgNVBAsTA1NWUzEOMAwGA1UEChMFQ2lzY28x
DDAKBgNVBAcTA1JUUDELMAkGA1UECBMCTkMxCzAJBgNVBAYTAlVTMSAwHgYJKoZI
hvcNAQkCFhFjYXQ5ay5Obw8uc3ZzLmNvbTBZMBMGByqGSM49AgEGCCqGSM49AwEH
AOIABOlgTtq2xyu2Xh1168KHdSDGNiLg3igEZaukkFyOrLiyjxXmLpFcO3uQeA/O
nXhCd/bRtxdU4MTXC1MAccGsheqjTTBLMB4GCWCGSAGG+EIBDQQRFg94Y2EgY2Vy
dGlmaWNhdGUwHAYDVR0RBBUwE4IRY2FOOWsudG1vLnN2cy5jb2OwCwYDVR0PBAQD
AgeAMA0GCSqGSIb3DQEBCwUAA4ICAQBObgKVykeyVC9UsvuuOAUsGaZHGwy2H9Yd
m5vIaui6PJczkCzIoAIghHPGQhIgpEcRqtGyXPZ2r8TCJP11WXNN/G73sFyWAhzY
RtmIM5KIojiDHLtifPayxv9juDuOZRx+wYR2PIQ5eLv1bafg7K8E82sqOCfOtcPr
OcONU8UCxqObdOgu4XsdBN1+wcWFqeQSDLmP7nxvhOOm/LXwCWUHwgVioOAuU2Fe
k5NthtvdxNAhRAImQdTyq6u/yB7vwTwJHcRiJc5USsyzCsTBb6RvL+HsXqBgXGc5
lxCSoLtYOdUxFIpJyK2MOZBY2zq2cNSc8Xbso5/OEQmnHtpWPvij4rSPUhQSY+4m
Qq2Sn3iqf4mGh/AO8T4iXfWDWfNezh7ZxMsCSCK/ZRlELZ2hj6OfzwX1H27Uf8XU
ecrOWx+WzRn7LVRCaGQzFkukfi8S4DLLNtxnNHfsLBVX5yHXCLEL+CQ7n8Z/pxcB
VVrPitwN3ZbO9poZyWiRLTnBsb42xNaWiL9bjQznAOiTDfmfFFourBsaAioz7ouY
2r1Mh+OpE83Uu+4lOTMawDgGiEv7iaiJ6xWc95EC+AdmOx3FvBXMtIM9qr7WwHW6
3C2hVYHJH254elV5+H8iiz7rovEPm8ZDsnvYpJn4Km3iDvBNqp/vvAHOFcyXrvG6
3i/1b9erGQ==
-----END CERTIFICATE-----

% Router Certificate successfully imported

cat9k(config)#
```

TACACSおよびAAAとTLSの設定

ステップ 1：TACACSサーバとAAAグループを作成し、クライアント（ルータ）トラストポイン

トを関連付けます。

```
tacacs server svs_tacacs
 address ipv4 10.225.253.209
 single-connection
 tls port 6049
 tls idle-timeout 60
 tls connection-timeout 60
 tls trustpoint client svs_cat9k
 tls ip tacacs source-interface GigabitEthernet0/0
 tls ip vrf forwarding Mgmt-vrf
!
aaa group server tacacs+ svs_tls
 server name svs_tacacs
 ip vrf forwarding Mgmt-vrf
!
tacacs-server directed-request
```

ステップ 2：AAA方式を設定します。

```
aaa authentication login default group svs_tls local enable
aaa authentication login console local enable
aaa authentication enable default group svs_tls enable
aaa authorization config-commands
aaa authorization exec default group svs_tls local if-authenticated
aaa authorization commands 1 default group svs_tls local if-authenticated
aaa authorization commands 15 default group svs_tls
aaa accounting exec default start-stop group svs_tls
aaa accounting commands 1 default start-stop group svs_tls
aaa accounting commands 15 default start-stop group svs_tls
aaa session-id common
```

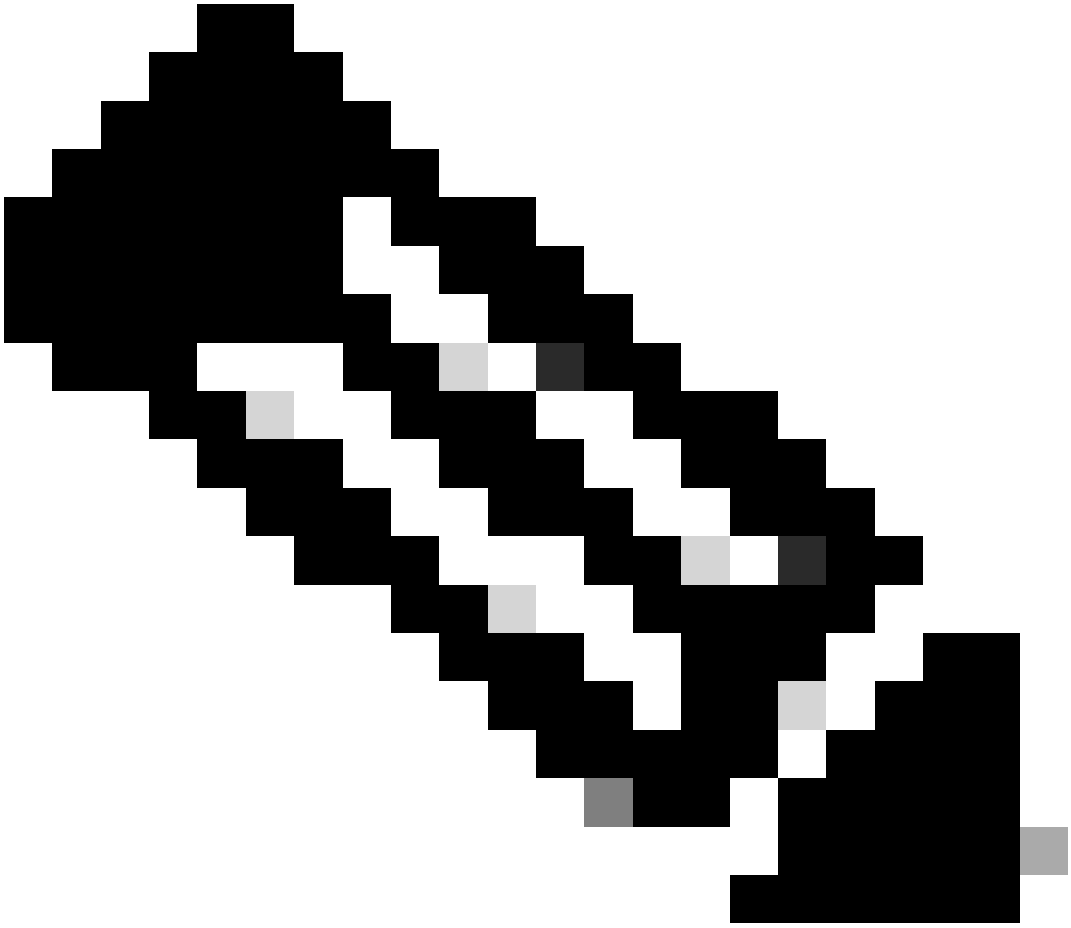
設定方法2:CAが生成したキーペア

CSR方式の代わりにPKCS#12形式でキーおよびデバイス証明書とCA証明書を直接インポートする場合は、この方式を使用できます。

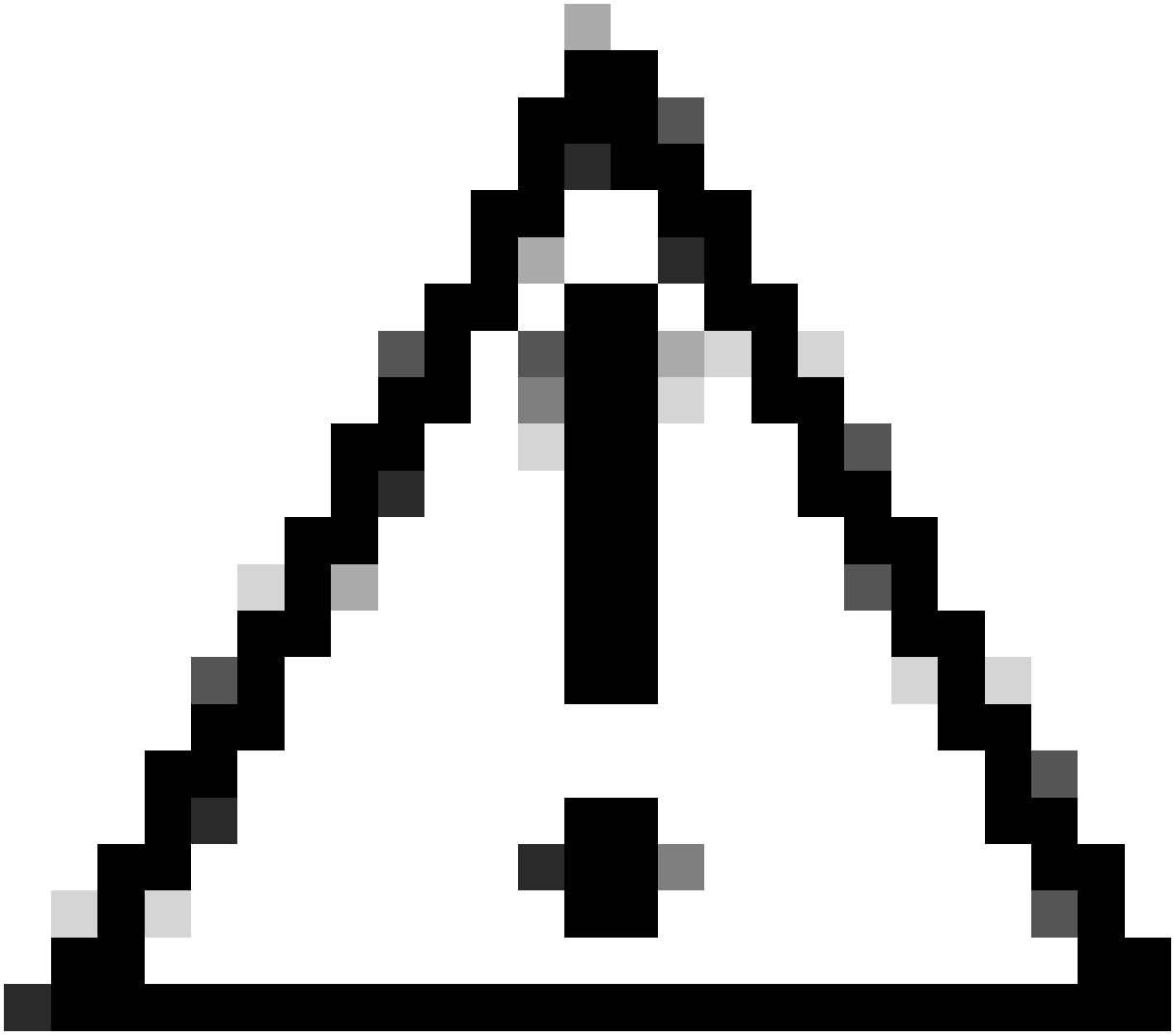
ステップ 1：クライアントトラストポイントを作成します。

```
cat9k(config)#crypto pki trustpoint svs_cat9k_25jun17
cat9k(ca-trustpoint)#revocation-check none
```

ステップ 2：PKCS#12ファイルをブートフラッシュにコピーします。



注:PKCS#12ファイルに完全な証明書チェーンと秘密キーが暗号化ファイルとして含まれていることを確認してください。



注意：インポートされるPKCS#12のキーは、ECCではなく、RSA（例：RSA 2048）タイプである必要があります。

<#root>

cat9k#

copy sftp bootflash: vrf Mgmt-vrf

Address or name of remote host [10.225.253.247]?
Source username [svs-user]?
Source filename [cat9k.svs.lab.pfx]? /home/svs-user/upload/cat9k-25jun17.pfx
Destination filename [cat9k-25jun17.pfx]?
Password:
!
2960 bytes copied in 3.022 secs (979 bytes/sec)

ステップ 3：importコマンドを使用してPKCS#12ファイルをインポートします。

<#root>

```
cat9k#

crypto pki import svs_cat9k_25jun17 pkcs12 bootflash:cat9k-25jun17.pfx

 password C1sco.123
% Importing pkcs12...Reading file from bootflash:cat9k-25jun17.pfx
CRYPTO_PKI: Imported PKCS12 file successfully.
cat9k#

cat9k#

show crypto pki certificates svs_cat9k_25jun17


Certificate
  Status: Available
  Certificate Serial Number (hex): 586OBF33A2O33365
  Certificate Usage: General Purpose
  Issuer:
    cn=SVS LabCA
    ou=SVS
    o=Cisco
    l=Raleigh
    st=North Carolina
    c=US
  Subject:
    Name: cat9k.svs.lab
    e=pkalkur@cisco.com
    cn=cat9k.svs.lab
    ou=svs
    o=cisco
    l=rtp
    st=nc
    c=us
  Validity Date:
    start date: 17:56:00 UTC Jun 17 2025
    end   date: 17:56:00 UTC Jun 17 2026
  Associated Trustpoints: svs_cat9k_25jun17

CA Certificate
  Status: Available
  Certificate Serial Number (hex): 2OCD74O2C4DA37F5
  Certificate Usage: General Purpose
  Issuer:
    cn=SVS LabCA
    ou=SVS
    o=Cisco
    l=Raleigh
    st=North Carolina
    c=US
  Subject:
    cn=SVS LabCA
    ou=SVS
    o=Cisco
    l=Raleigh
    st=North Carolina
    c=US
  Validity Date:
    start date: 17:05:00 UTC Apr 28 2025
    end   date: 17:05:00 UTC Apr 28 2035
  Associated Trustpoints: svs_cat9k_25jun17 svs_cat9k
```

```
  Storage: nvram:SVSLabCA#37F5CA.cer
```

TACACSおよびAAAとTLSの設定

ステップ 1：TACACSサーバとAAAグループを作成し、クライアント（ルータ）トラストポイントを関連付けます。

```
tacacs server svs_tacacs
 address ipv4 10.225.253.209
 single-connection
 tls port 6049
 tls idle-timeout 60
 tls connection-timeout 60
 tls trustpoint client svs_cat9k
 tls ip tacacs source-interface GigabitEthernet0/0
 tls ip vrf forwarding Mgmt-vrf
!
aaa group server tacacs+ svs_tls
 server name svs_tacacs
 ip vrf forwarding Mgmt-vrf
!
tacacs-server directed-request
```

ステップ 2：AAA方式を設定します。

```
aaa authentication login default group svs_tls local enable
aaa authentication login console local enable
aaa authentication enable default group svs_tls enable
aaa authorization config-commands
aaa authorization exec default group svs_tls local if-authenticated
aaa authorization commands 1 default group svs_tls local if-authenticated
aaa authorization commands 15 default group svs_tls
aaa accounting exec default start-stop group svs_tls
aaa accounting commands 1 default start-stop group svs_tls
aaa accounting commands 15 default start-stop group svs_tls
aaa session-id common
```

検証

設定を確認します。

```
show tacacs
show crypto pki certificates <>
show crypto pki trustpoints <>
```

AAAおよびTACACS+のデバッグ。

```
debug aaa authentication
debug aaa authorization
debug aaa accounting
debug aaa subsys
debug aaa protocol local
debug tacacs authentication
debug tacacs authorization
debug tacacs accounting
debug tacacs events
debug tacacs packet
debug tacacs
debug tacacs secure

! Below debugs will be needed only if there is any issue with SSL Handshake
debug ip tcp transactions
debug ip tcp packet
debug crypto pki transactions
debug crypto pki API
debug crypto pki messages
debug crypto pki server
debug ssl openssl errors
debug ssl openssl msg
debug ssl openssl states
clear logging
```

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。