OpenAPIを使用したISE 3.3でのISE証明書情報の取得

内容

はじめに

このドキュメントでは、openAPIを使用してCisco Identity Services Engine(ISE)証明書を管理する手順について説明します。

背景

企業ネットワークのセキュリティと管理がますます複雑化する中、Cisco ISE 3.1では、証明書ライフサイクル管理を合理化するOpenAPI形式のAPIを導入し、効率的で安全な証明書操作のための標準化および自動化されたインターフェイスを提供して、管理者が強力なセキュリティ対策を実施し、ネットワークコンプライアンスを維持できるようにします。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- Cisco Identity Services Engine（ISE）
- REST API
- Python

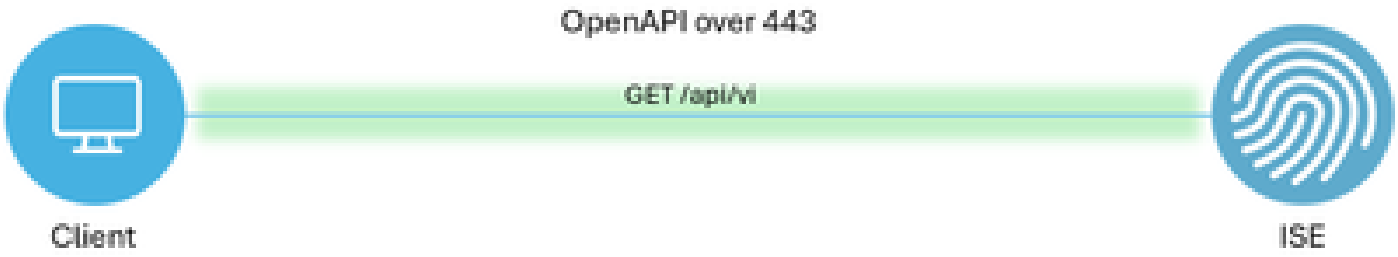
使用するコンポーネント

- ISE 3.3
- Python 3.10.0

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。

設定

ネットワーク図

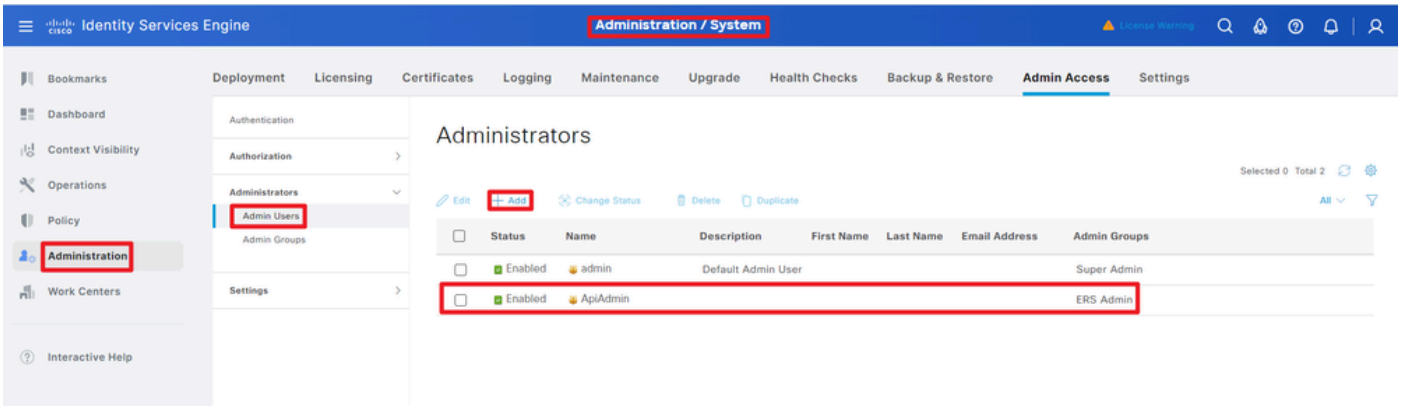


トポロジ

ISEでの設定

ステップ1:Open API adminアカウントを追加します。

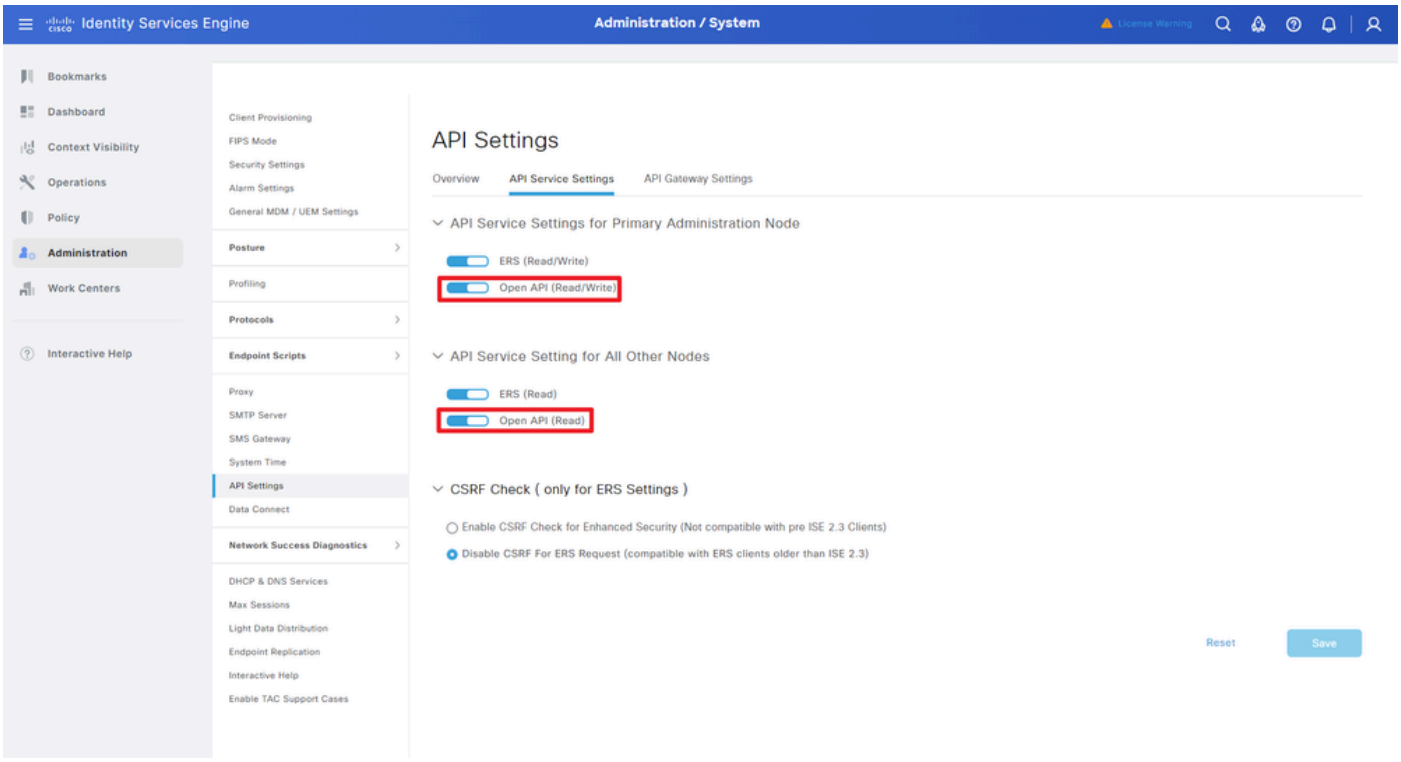
API管理者を追加するには、Administration -> System -> Administration -> Administrators -> Admin Users -> Addの順に移動します。



API管理者

ステップ2:ISEでOpen APIを有効にする

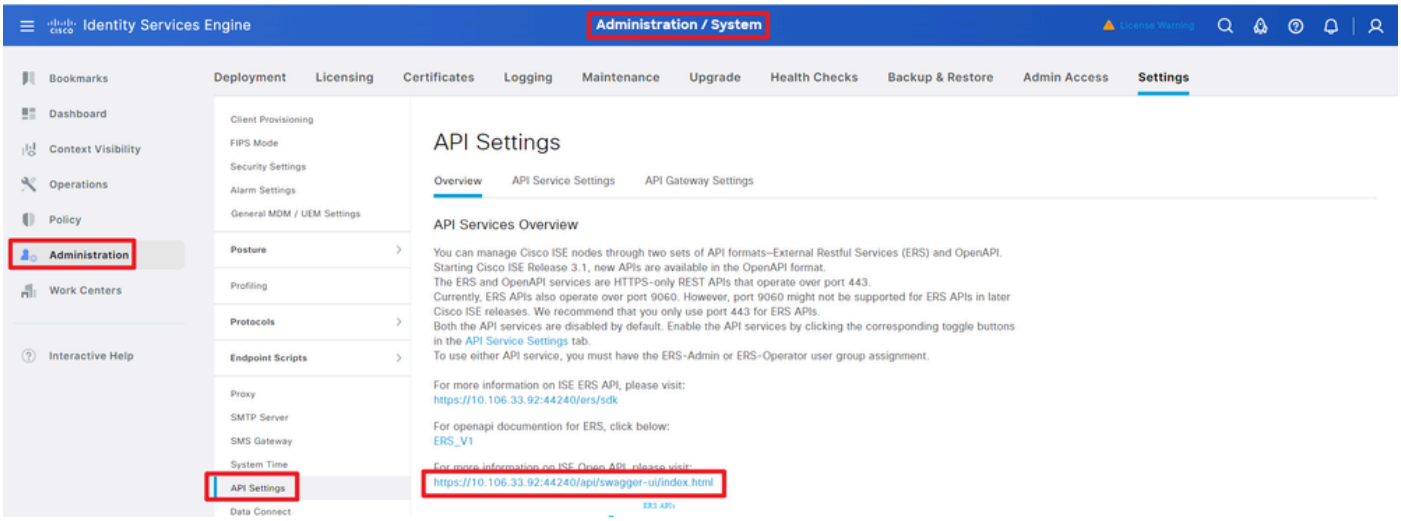
ISEでは、オープンAPIはデフォルトで無効になっています。これを有効にするには、Administration > System > API Settings > API Service Settingsの順に移動します。Open APIオプションを切り替えます。[Save] をクリックします。



OpenAPIの有効化

ステップ3:ISEオープンAPIを調べる

Administration > System > API Settings > Overviewの順に移動します。「APIを開く」をクリックします。



OpenAPIにアクセス

Pythonの例

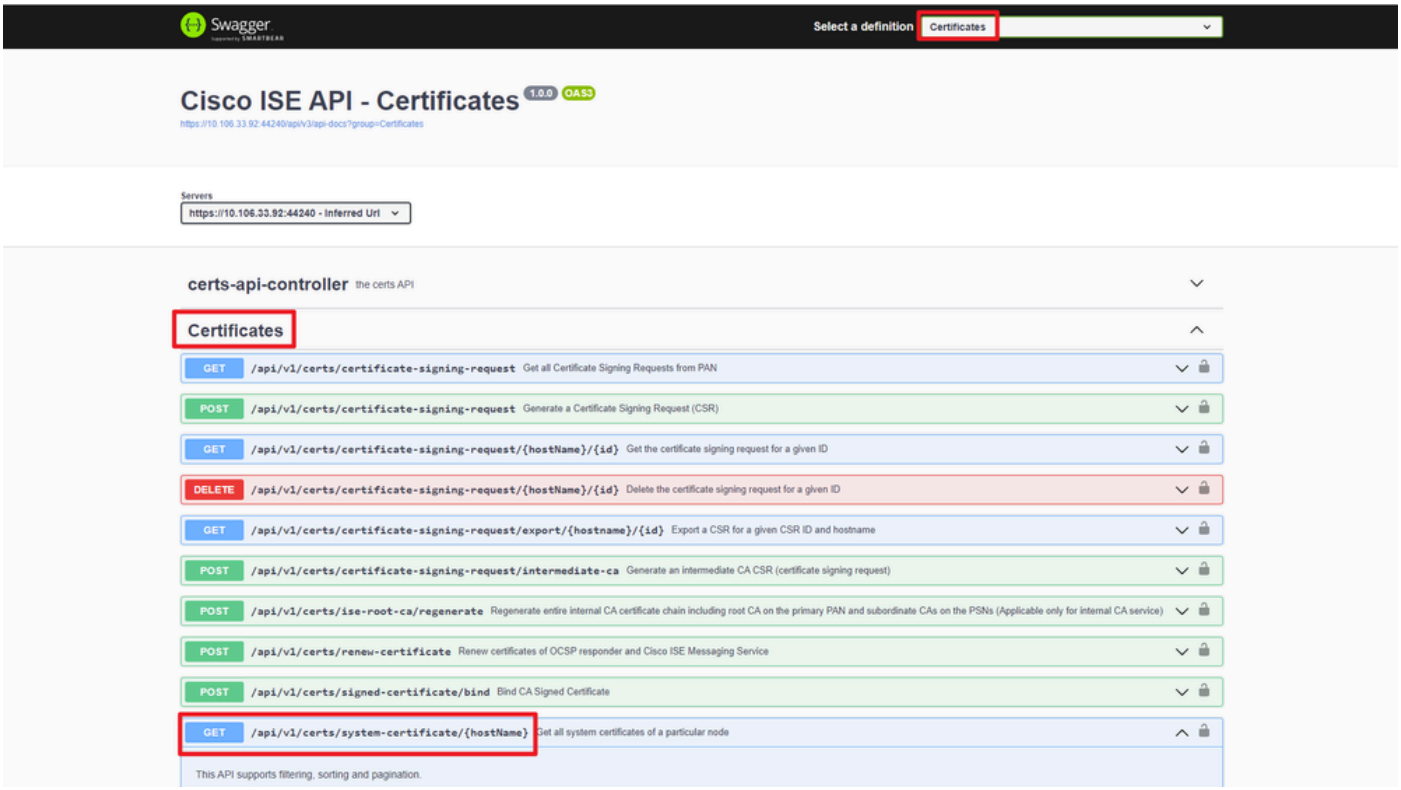
特定のノードのすべてのシステム証明書を取得する

APIは、特定のISEノードのすべての証明書をリストします。

ステップ1:APIコールに必要な情報。

メソッド	GET
URL	https://<ISE-PAN-IP>/api/v1/certs/system-certificate/<ISE-Node-Hostname>
Credentials	Open APIアカウントの資格情報を使用する
ヘッダー	受け入れ:application/json Content-Type:application/json

ステップ2：特定のISEノードの証明書を取得するために使用されるURLを特定します。



API URI(API URI)

ステップ3:Pythonコードの例を次に示します。コンテンツをコピーして貼り付けます。ISEのIP、ユーザ名、パスワードを置き換えます。実行するPythonファイルとして保存します。

ISEとPythonコード例を実行しているデバイスの間の接続が良好であることを確認します。

<#root>

```
from requests.auth import HTTPBasicAuth
import requests

requests.packages.urllib3.disable_warnings()

if __name__ == "__main__":

    url = "
https://10.106.33.92/api/v1/certs/system-certificate/ISE-DLC-CFME02-PSN
"
```

```
    headers = {
"Accept": "application/json", "Content-Type": "application/json"
}
    basicAuth = HTTPBasicAuth(
"ApiAdmin", "Admin123"
)
    response = requests.get(url=url, auth=basicAuth, headers=headers, verify=False)
    print("Return Code:")
    print(response.status_code)
    print("Expected Outputs:")
    print(response.json())
```

次に、予想される出力の例を示します。

Return Code:
200
Expected Outputs:
{'response': [{'id': '5b5b28e4-2a51-495c-8413-610190e1070b', 'friendlyName': 'Default self-signed saml server certificate - CN=SAML_ISE-DLC-CFME0

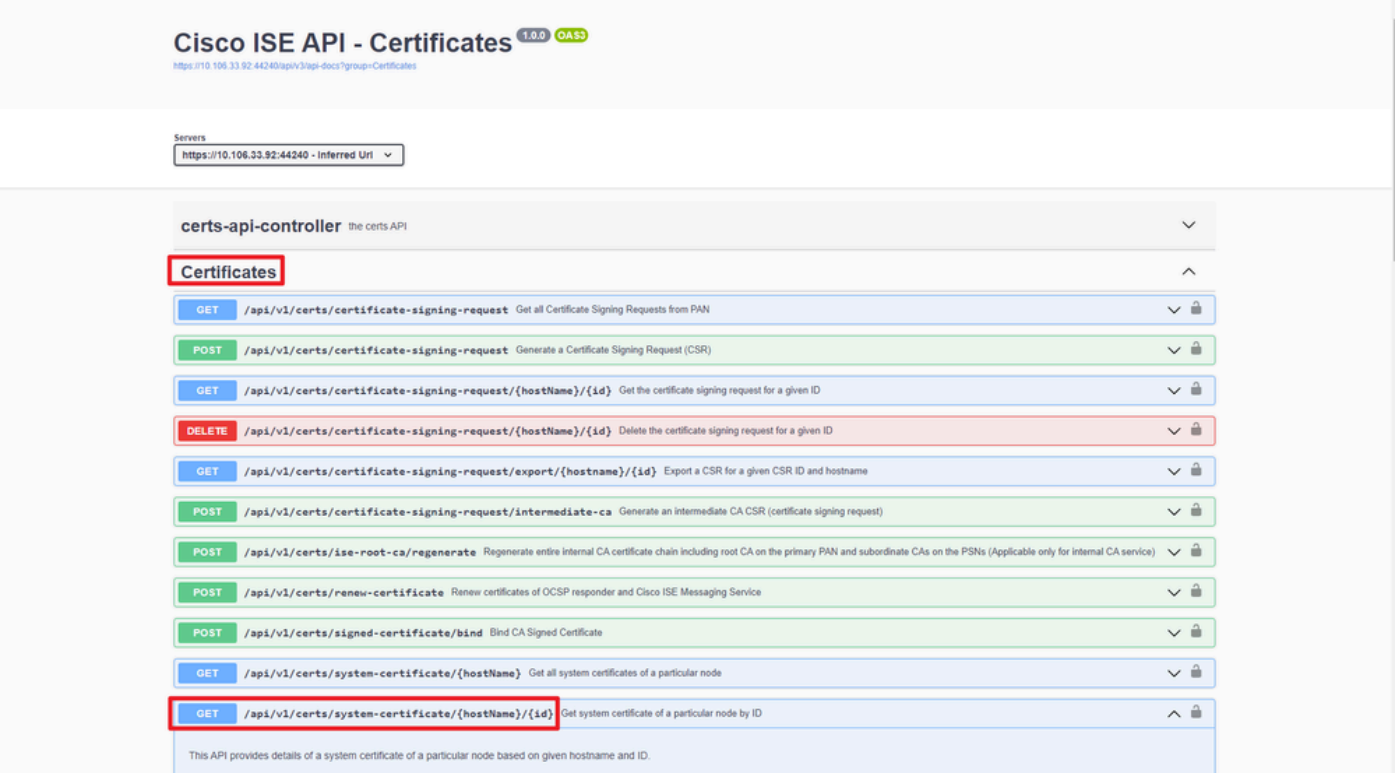
特定のノードのシステム証明書をIDで取得する

このAPIは、指定されたホスト名とIDに基づいて、特定のノードのシステム証明書の詳細を提供します。

ステップ1:APIコールに必要な情報。

メソッド	GET
URL	https://<ISE-PAN-IP>/api/v1/certs/system-certificate/<ISE-Node-Hostname>/<ID-Of-Certificate>
Credentials	Open APIアカウントの資格情報を使用する
ヘッダー	受け入れ:application/json Content-Type:application/json

ステップ2：指定されたホスト名とIDに基づいて特定のノードの証明書を取得するために使用されるURLを見つけます。



API URI(API URI)

ステップ3：ここはPythonコードの例です。コンテンツをコピーして貼り付けます。ISEのIP、ユーザ名、パスワードを置き換えます。実行するPythonファイルとして保存します。

ISEとPythonコード例を実行しているデバイスの間の接続が良好であることを確認します。

<#root>

from requests.auth import HTTPBasicAuth import requests requests.packages.urllib3.disable_warnings() if __name__ == "__main__": url = "

https://10.106.33.92/api/v1/certs/system-certificate/ISE-DLC-CFME02-PSN/5b5b28e4-2a51-495c-8413-610190e1

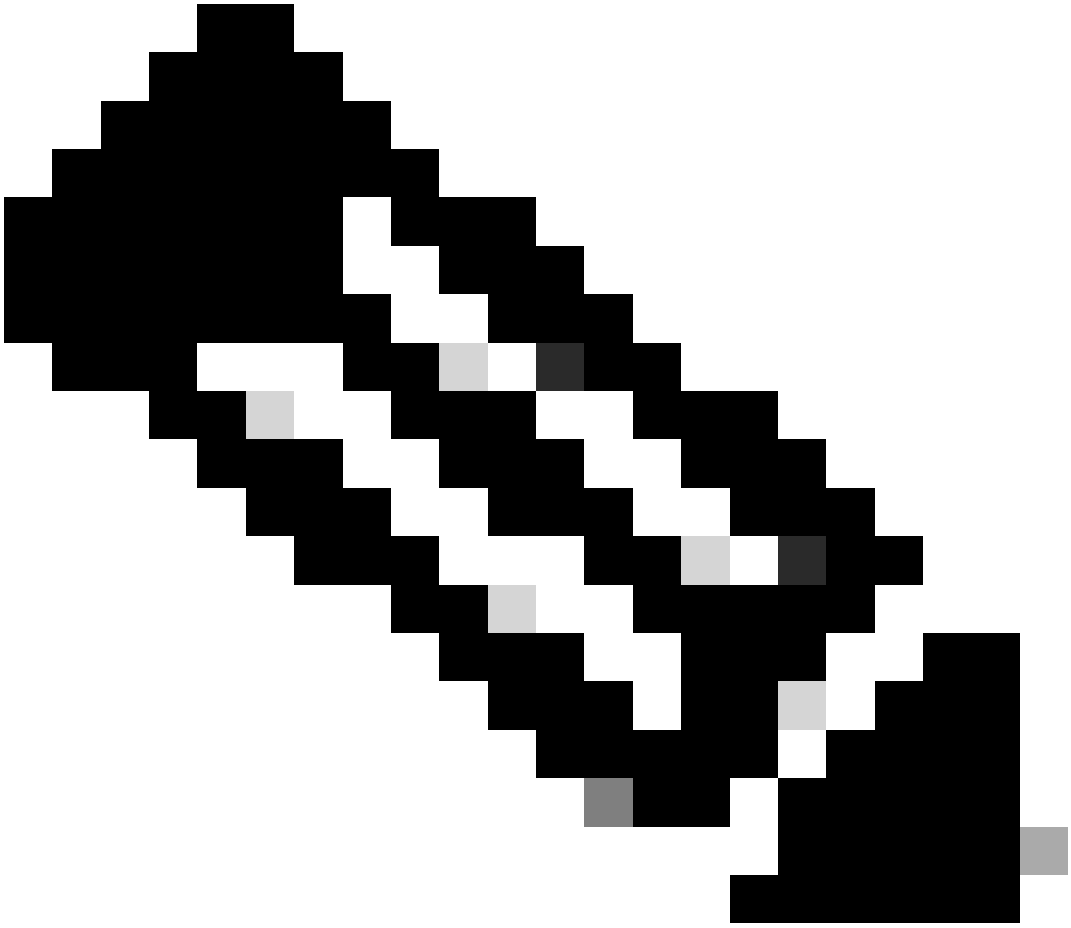
" headers = {

"Accept": "application/json", "Content-Type": "application/json"

} basicAuth = HTTPBasicAuth(

"ApiAdmin", "Admin123"

) response = requests.get(url=url, auth=basicAuth, headers=headers, verify=False) print("Return Code:")



注：このIDは、「特定のノードのすべてのシステム証明書の取得」のステップ3のAPI出力からのものです。たとえば、5b5b28e4-2a51-495c-8413-610190e1070bは「デフォルトの自己署名証明書 – CN=SAML_ISE-DLC-CFME02-PSN.cisco.com」です。

次に、予想される出力の例を示します。

Return Code:
200
Expected Outputs:
{'response': {'id': '5b5b28e4-2a51-495c-8413-610190e1070b', 'friendlyName': 'Default self-signed saml server certificate - CN=SAML_ISE-DLC-CFME02

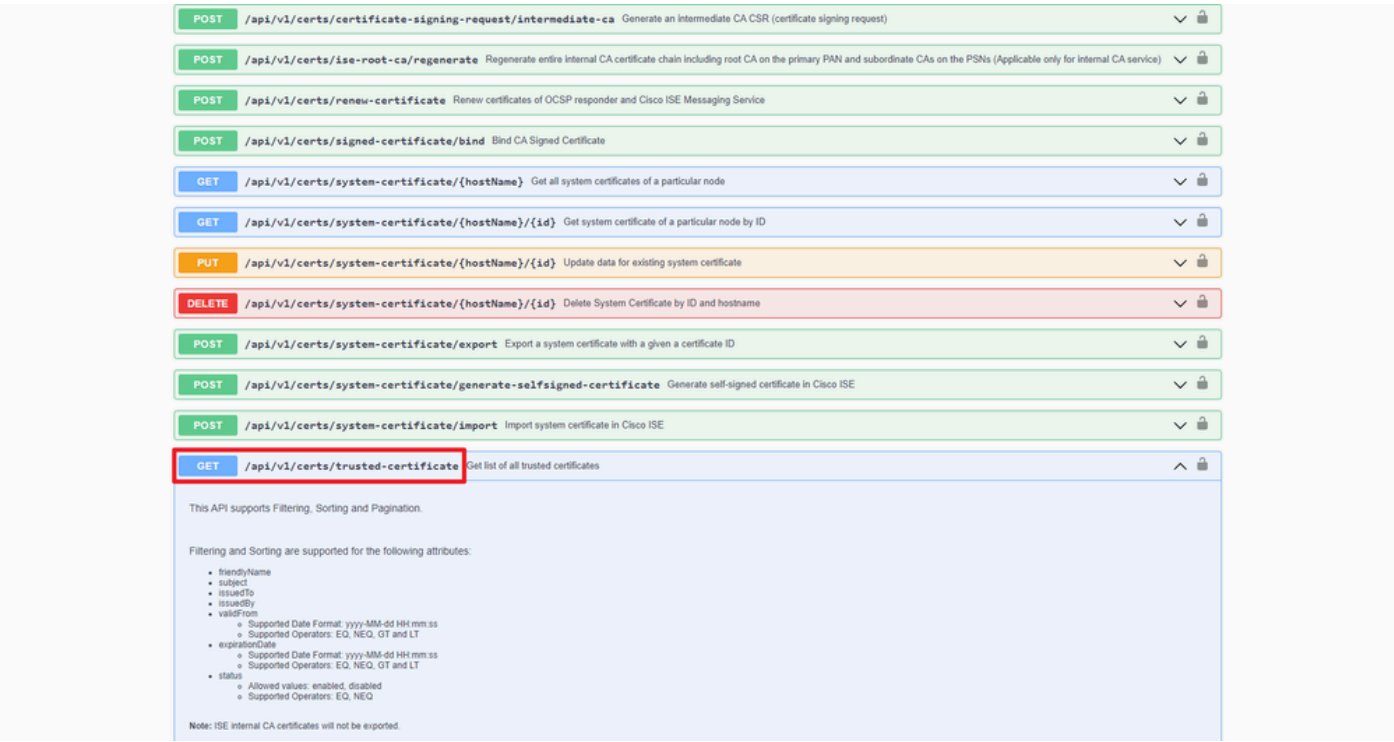
すべての信頼できる証明書の一覧を取得する

APIは、ISEクラスタのすべての信頼できる証明書をリストします。

ステップ1:APIコールに必要な情報。

メソッド	GET
URL	https://<ISE-PAN-IP>/api/v1/certs/trusted-certificate（信頼できる証明書）
Credentials	Open APIアカウントの資格情報を使用する
ヘッダー	受け入れ:application/json Content-Type:application/json

ステップ2：信頼できる証明書を取得するために使用されるURLを見つけます。



API URI(API URI)

ステップ3：ここはPythonコードの例です。コンテンツをコピーして貼り付けます。ISEのIP、ユーザ名、パスワードを置き換えます。実行するPythonファイルとして保存します。

ISEとPythonコード例を実行しているデバイスの間の接続が良好であることを確認します。

<#root>

from requests.auth import HTTPBasicAuth import requests requests.packages.urllib3.disable_warnings() if __name__ == "__main__": url = "

https://10.106.33.92/api/v1/certs/trusted-certificate

" headers = {

"Accept": "application/json", "Content-Type": "application/json"

} basicAuth = HTTPBasicAuth(

"ApiAdmin", "Admin123"

```
) response = requests.get(url=url, auth=basicAuth, headers=headers, verify=False) print("Return Code:")
```

想定される出力例を以下に示します（略）。

```
Return Code:
200
Expected Outputs:
{'response': [{'id': '147d97cc-6ce9-43d7-9928-8cd0fa83e140', 'friendlyName': 'VeriSign Class 3 Public Primary Certification Authority', 'subject': 'CN=Ver
```

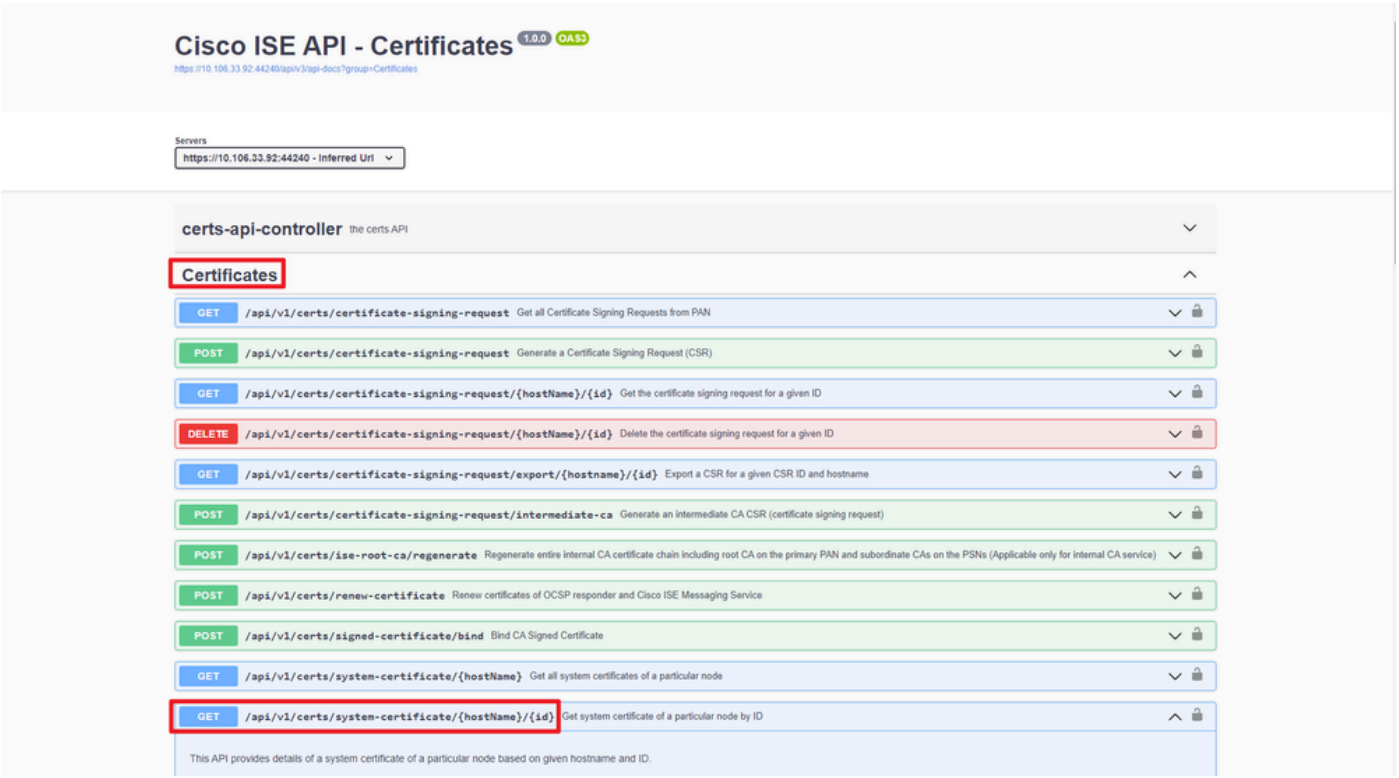
IDによる信頼証明書の取得

このAPIは、指定されたIDに基づいて信頼証明書の詳細を表示できます。

ステップ1:APIコールに必要な情報。

メソッド	GET
URL	https://<ISE-PAN-IP>/api/v1/certs/trusted-certificate/<ID-Of-Certificate>
Credentials	Open APIアカウントの資格情報を使用する
ヘッダー	受け入れ:application/json Content-Type:application/json

手順2：展開情報の取得に使用するURLを見つけます。



API URI(API URI)

ステップ3：ここはPythonコードの例です。コンテンツをコピーして貼り付けます。ISEのIP、ユーザ名、パスワードを置き換えます。実行するPythonファイルとして保存します。

ISEとPythonコード例を実行しているデバイスの間の接続が良好であることを確認します。

<#root>

from requests.auth import HTTPBasicAuth import requests requests.packages.urllib3.disable_warnings() if __name__ == "__main__": url = "

https://10.106.33.92/api/v1/certs/trusted-certificate/147d97cc-6ce9-43d7-9928-8cd0fa83e140

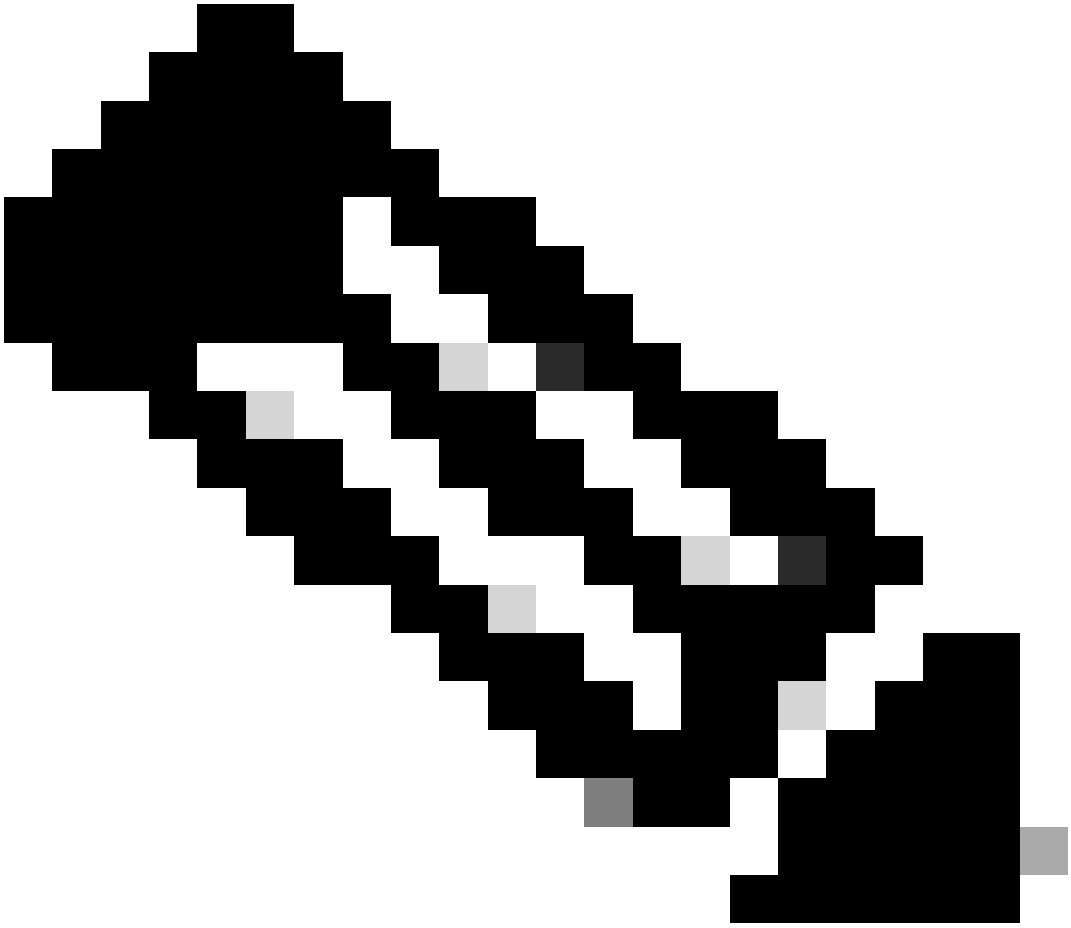
" headers = {

"Accept": "application/json", "Content-Type": "application/json"

} basicAuth = HTTPBasicAuth(

"ApiAdmin", "Admin123"

) response = requests.get(url=url, auth=basicAuth, headers=headers, verify=False) print("Return Code:")



注：このIDは、「すべての信頼できる証明書のリストの取得」のステップ3のAPI出力からのものです。たとえば、147d97cc-6ce9-43d7-9928-8cd0fa83e140は、「VeriSign Class 3 Public Primary Certification Authority」です。

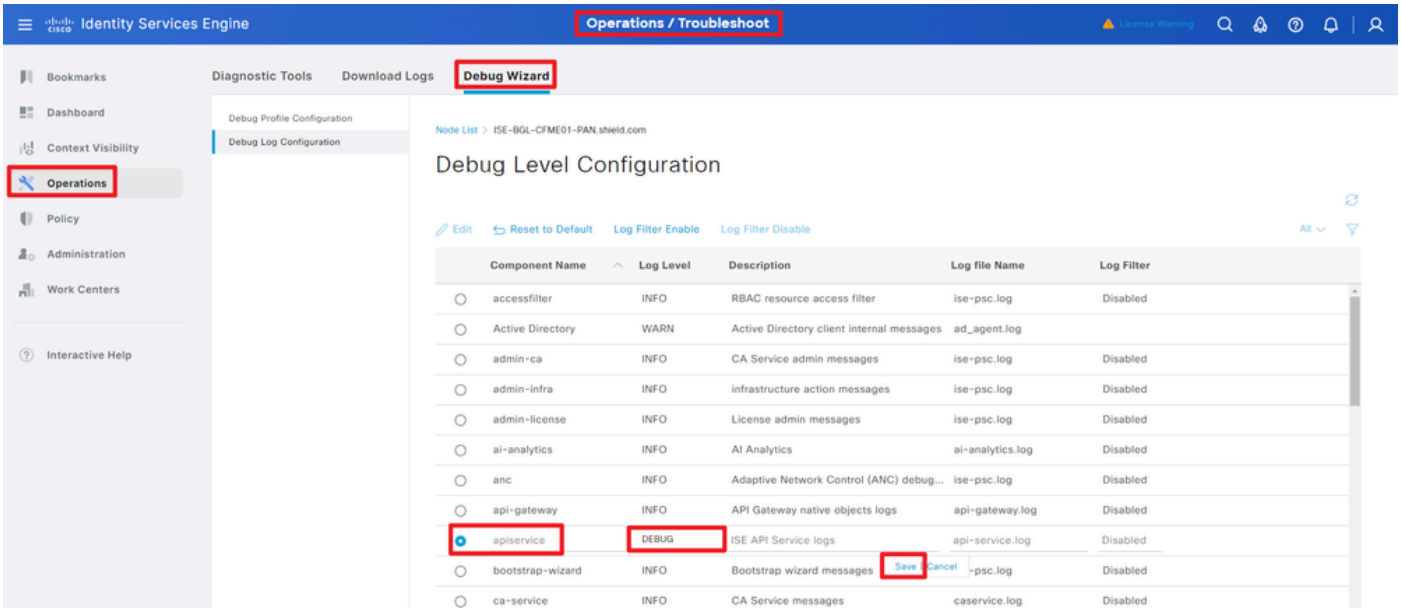
次に、予想される出力の例を示します。

Return Code: 200 Expected Outputs: {'response': {'id': '147d97cc-6ce9-43d7-9928-8cd0fa83e140', 'friendlyName': 'VeriSign Class 3 Public Primary Certifi

トラブルシュート

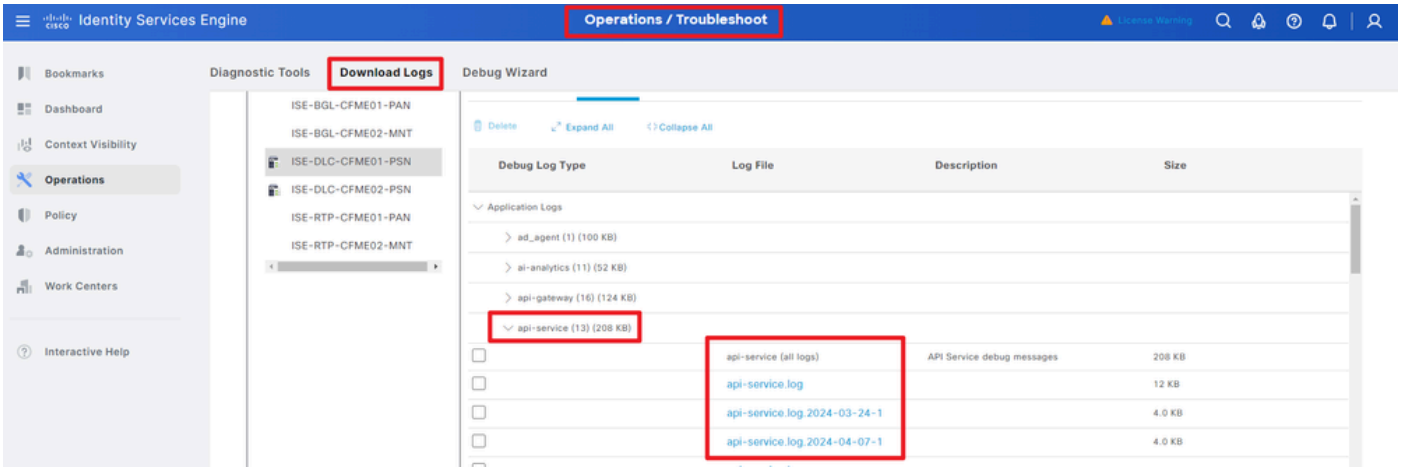
オープンAPIに関連する問題をトラブルシューティングするには、デバッグログ設定ウィンドウで**theapiservicecomponent**のログレベルをDEBUGに設定します。

デバッグを有効にするには、**Operations -> Troubleshoot -> Debug Wizard -> Debug Log Configuration -> ISE Node -> apiservice**の順に移動します。



APIサービスのデバッグ

デバッグログをダウンロードするには、**Operations -> Troubleshoot -> Download Logs -> ISE PAN Node -> Debug Logs**の順に移動します。



デバッグログのダウンロード

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。