FMCでのCisco RADKit統合の設定

内容

はじめに

- 背景
- 機能説明とウォークスルー
- FMC REST API
- デバイスから詳細を取得する
- シスコサポート：RADKitコンソール
- アップグレードと下位互換性

トラブルシューティング

- 診断概要
- RADKitセッションログ
- トラブルシューティングのウォークスルーの問題例
- テレメトリ

FAQ

はじめに

このドキュメントでは、7.7リリースで追加されたFMCのCisco RADKit統合機能について説明します。

背景

ファイアウォール管理者が直面する問題

- シスコが開発したRemote Automation Development Kit(RADKit)は、ユーザが安全な方法でアクセスし、ネットワークデバイスのトラブルシューティングを行えるように設計されたネットワーク全体のオーケストレータです。https://radkit.cisco.com/
- Cisco Secure Firewall Management Center(FMC)は、Secure Firewall Threat Defense(FTD)デバイスを管理および運用します。1つのFMCで、さまざまな場所にある複数のデバイスを管理できます。
- ユーザがRADKitを個別にインストールし、FMCとFTDをその中にオンボーディングすることは可能ですが、FMCにRADKitサービスを構築し、FMCとすべての管理対象デバイス(FTD)を自動化された方法でオンボーディングすることは、エンドユーザにとってより良い経験になるでしょう。

使用例

FMCにRADKitを統合すると、次のような重要な機能が得られます。

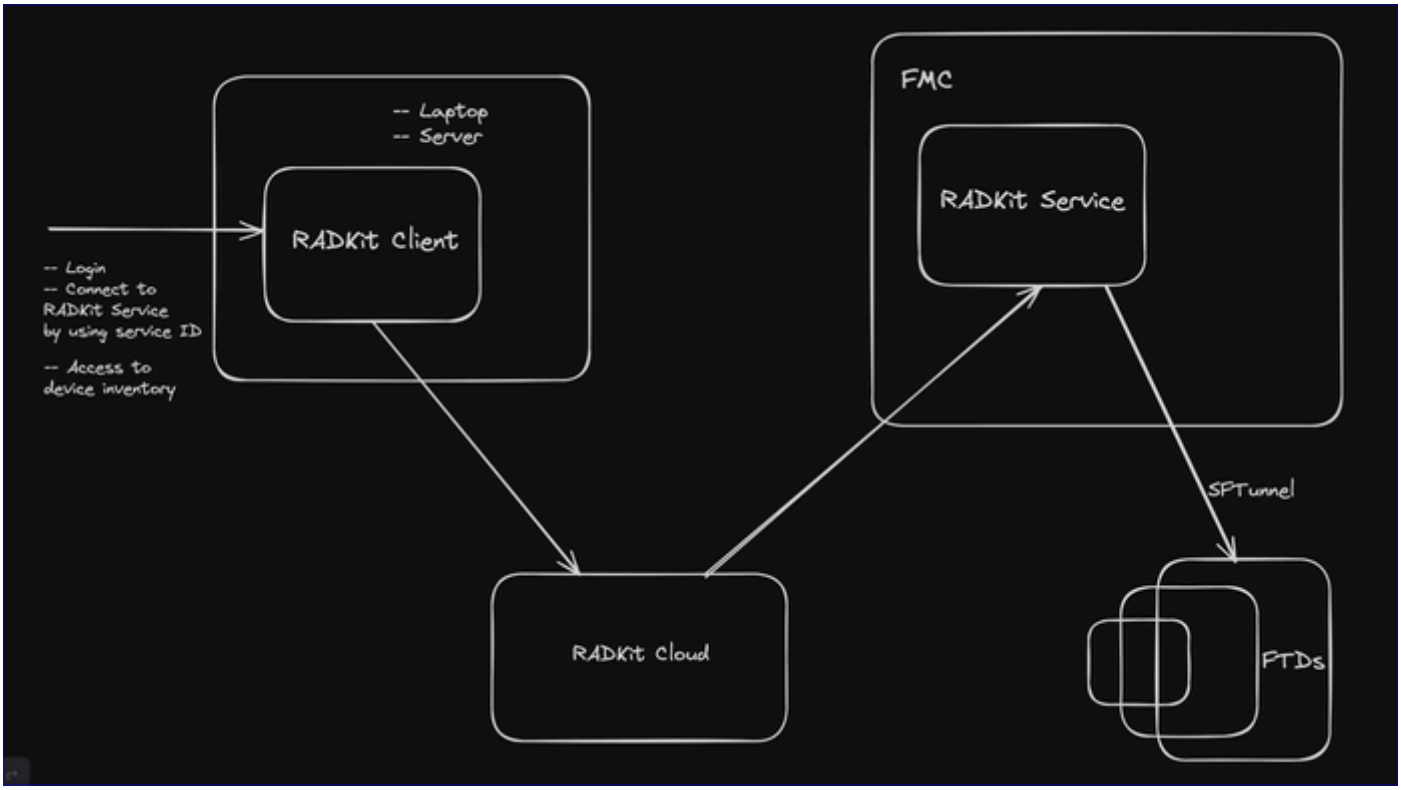
- RADKitクライアントCLIからFMC/FTDにリモートアクセス可能
- FMC/FTDを必要とするユーザ（Cisco TACエンジニアなど）に、FMC/FTDへのアクセス制御を提供する機能。
- RADKitクライアントからデータを収集し、問題を診断するための自動化機能を活用します（複数のデバイスでコマンドを実行するスクリプトをRADKitクライアントから作成して使用できます）。

新機能 – ソリューション

- Secure Firewall 7.7.0以降では、Remote Automation Development Kit(RADKit)サービスがFMCに統合されています。
- ユーザはオンデマンドでRADKitサービスを有効または無効にし、RADKitクラウドに登録し、スケジュールされたアクセス期間にRADKitクライアントから特定のデバイスにアクセスするリモートユーザの承認を作成できます。
    - 承認は、編集または取り消すことができます。
- 高度なトラブルシューティングのためにデバイスにsudoアクセスを提供するオプションもあります。

FMCでのRADKitサービス統合の図

次の図は、RADKitがユーザ（TACエンジニア）のRADKitクライアントから実稼働FTDデバイスへの通信をどのように実現するかを示しています。



基本：サポートされるプラットフォーム、ライセンス

アプリケーションとマネージャ

FTD		ASA	
FMC and FTD Platforms: All		Not supported	
FMC on 7.7.0 FMC REST API	Yes Yes	ASA CLI 9.23.1	No
FTD Supported Versions *(lowest version FMC on 7.7.0 can manage is 7.2)*	7.7.0 only	ASDM 7.23.1	No
Snort Support (Snort 3 is the only Snort version supported in 7.7)	Snort 3 Snort 2 (only for devices on 7.2.x..7.6.x)	CSM 4.30	No
FDM on 7.7.0	No		

サポートのその他の側面

Platforms	
FTD	
Licenses Required	No licensing requirements for this feature.
Works in Evaluation Mode	Yes
IP Addressing	IPv4 IPv6
Multi-instances supported?	Yes
Supported with HA'd devices	Yes
Supported with clustered devices?	Yes
Other (only routed mode | transparent mode), etc.	No Special Notes
Internet access for the RADKit cloud enrollment required	Access to prod.radkit-cloud.cisco.com

機能の動作に関する依存関係

- 最小バージョンはセキュアファイアウォール7.7.0です。
- FMC内でホストされているRADKitサービスに接続するには、サポートエンジニアのコンピュータでhttps://radkit.cisco.com/downloads/release/からRADKitクライアントをインストールする必要があります。
- RADKit Clientの推奨バージョンは1.6.10以降です。
- RADKit Clientの古いバージョンは、RADKit Serviceが古いバージョンのRADKit Clientと下位互換性があるため、使用できます。

機能説明とウォークスルー

機能の概要

- FMCへのRADKitサービスの統合により、デバイス管理者（Cisco TACエンジニア）は、トラブルシューティングと自動化の目的で、ネットワーク内の特定のFMCおよびFTDデバイスへのリモートユーザのアクセスを提供できます。RADKこれは画面共有よりもはるかに効率的で、ユーザーのコンピュータを制御する必要がなく、ネットワーク上で作業する安全な方法であり、Webexを見事に補完します。
- これにより、デバイス管理者はRADKitサービスを個別にインストールして設定する必要がないため、テクニカルサポートのエクスペリエンスが向上します。また、これにより、サポートの問題を解決する際のCisco TACエンジニアのサポート時間が短縮されます。

設定手順：概要

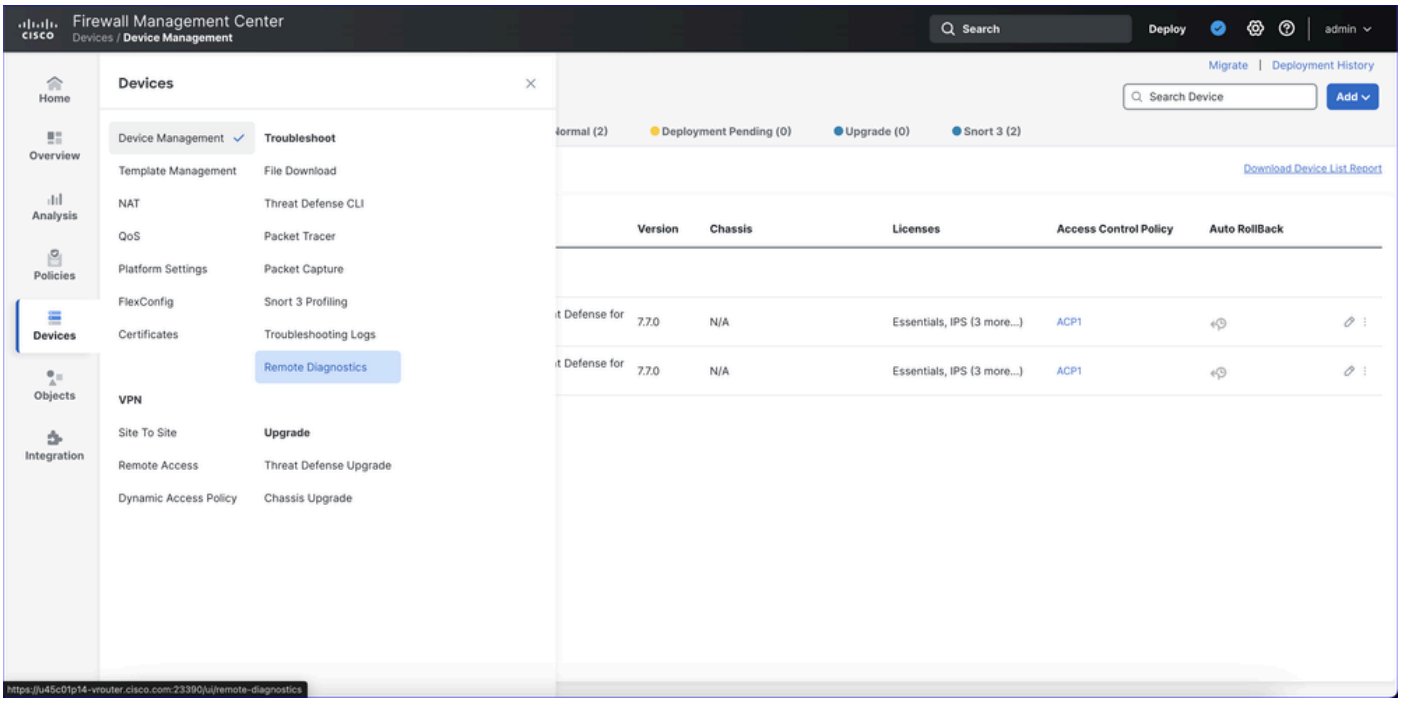
1. デバイス管理者（FMC管理者ユーザ）:RDKitサービスを有効にして登録し、FMC GUIで承認を設定します。

2. Cisco TAC/シスコサポート：RADKit Clientをコンピュータにインストールし、RADKit Clientからデバイスにアクセスしてトラブルシューティングを行います。

FMC管理者ユーザ：ファイアウォール管理センターのウォークスルー

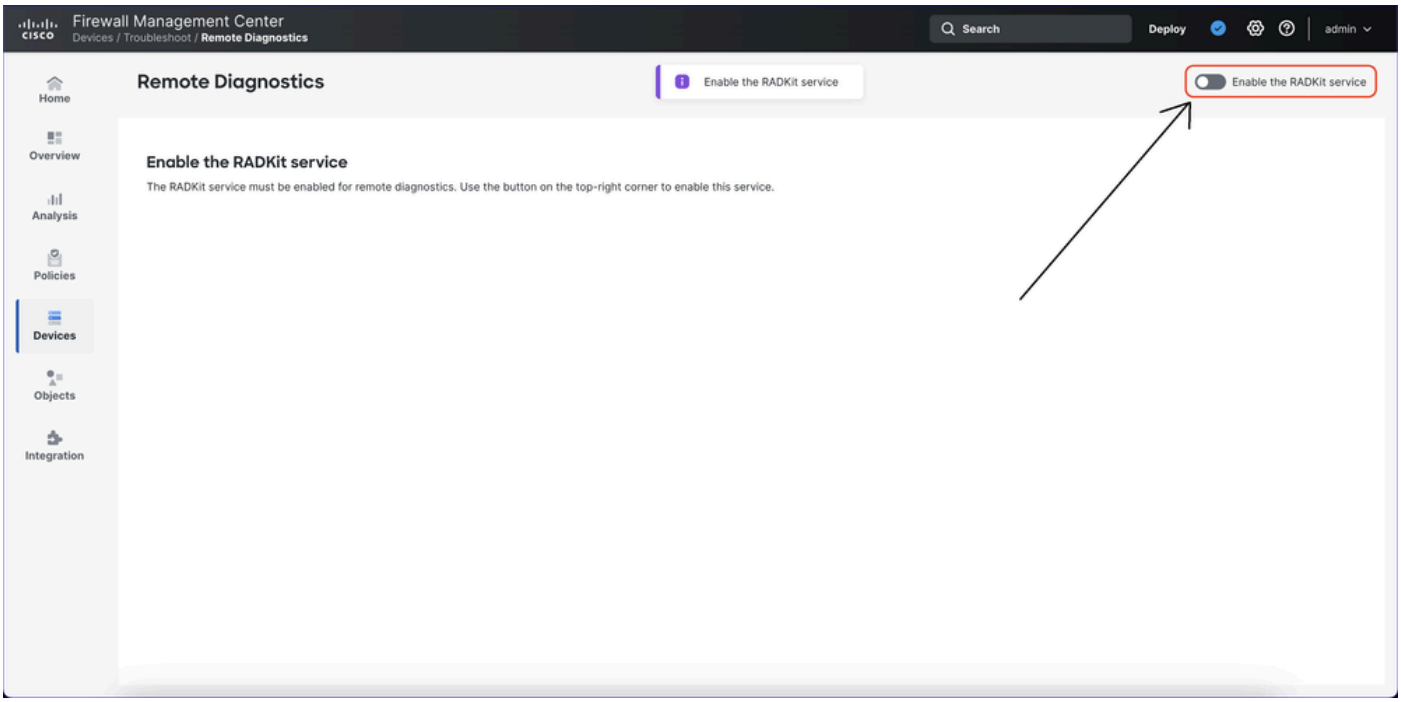
リモート診断メニュー

- Devices -> Troubleshootの下に、この機能のための新しい「Remote Diagnostics」メニュー項目が追加されました。
- 管理者、ネットワーク管理者、およびメンテナンスユーザには、ページに対する読み取り/書き込み権限があります。
- Security Analyst、Security Analyst（読み取り専用）、およびSecurity Approverの各ユーザーには、ページに対する読み取り専用のアクセス許可が与えられます。



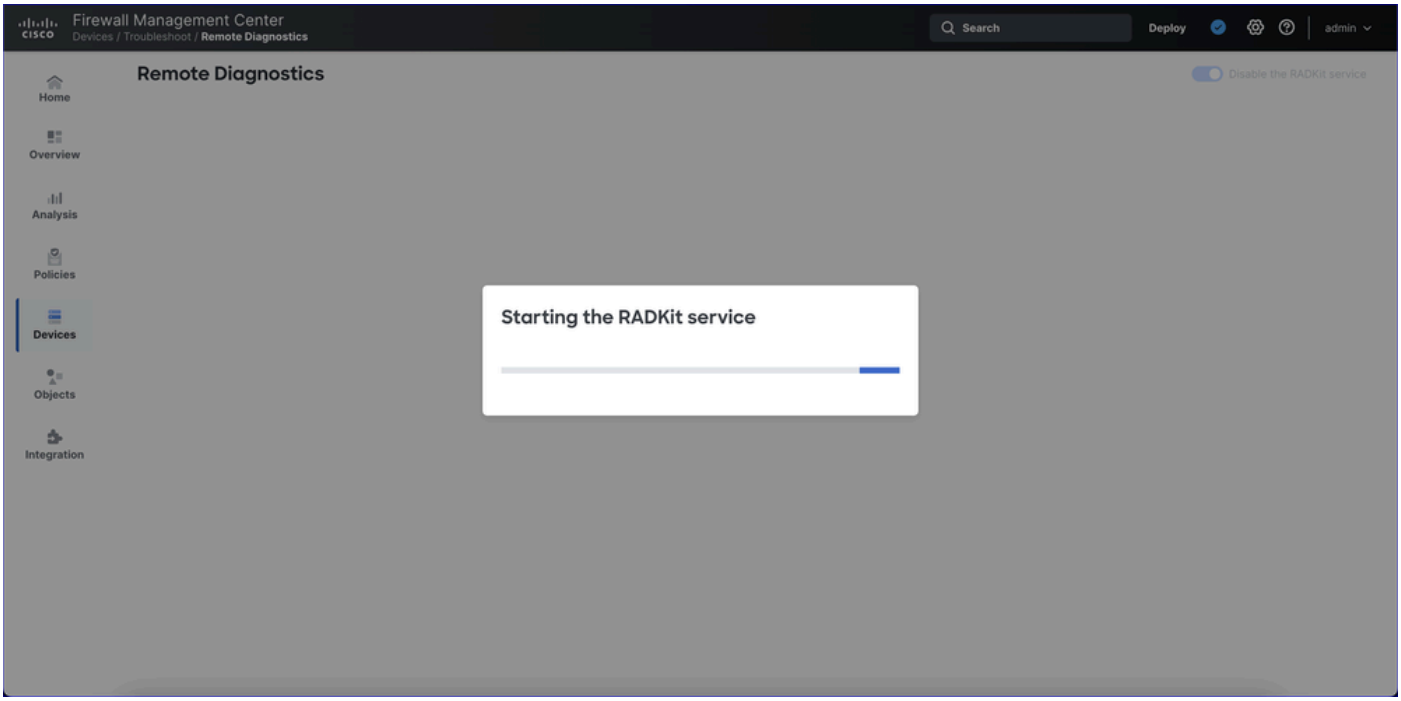
初期リモート診断ページ

これは最初のリモート診断ページです。RADKitサービスを有効にするには、「Enable the RADKit service」スイッチを切り替えます。



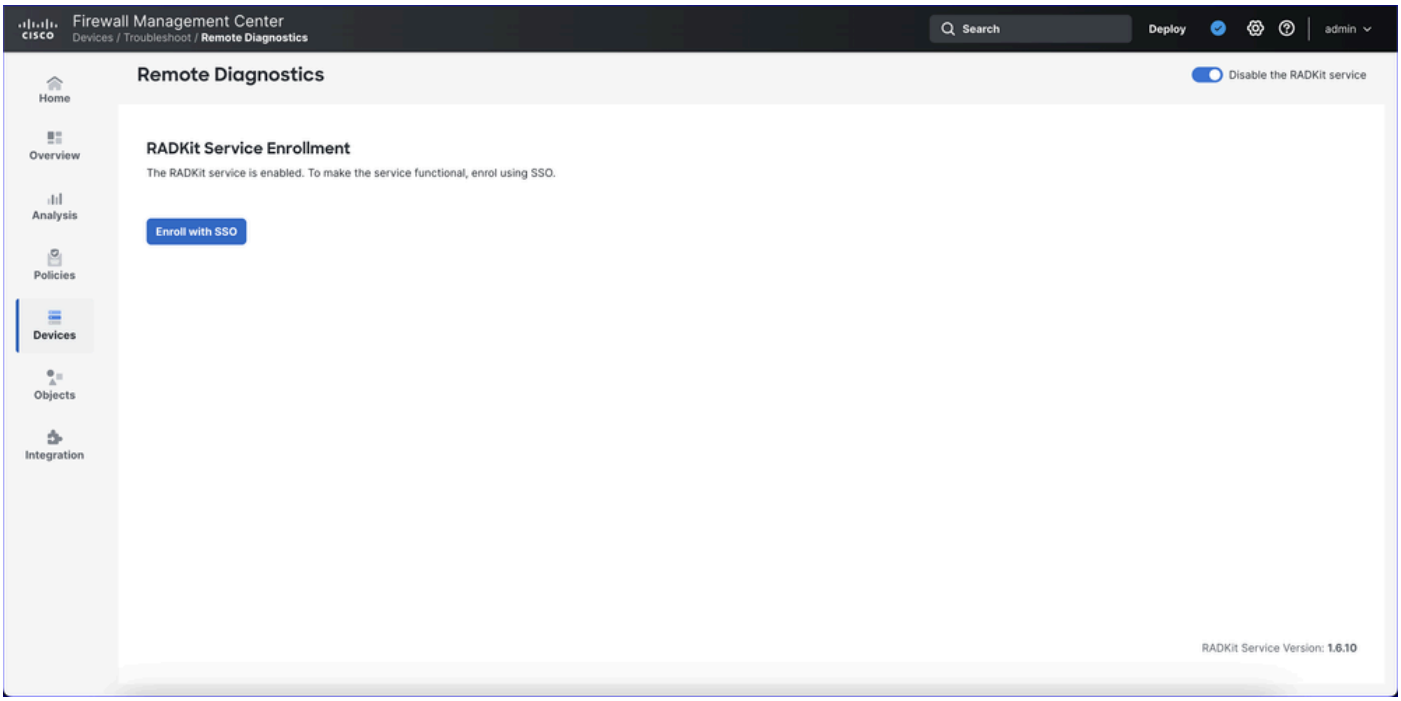
RADKitサービスの開始

RADKitサービスを有効にすると、RADKitサービスが開始されるまで経過表示バーが表示されます。



有効なRADKitサービス

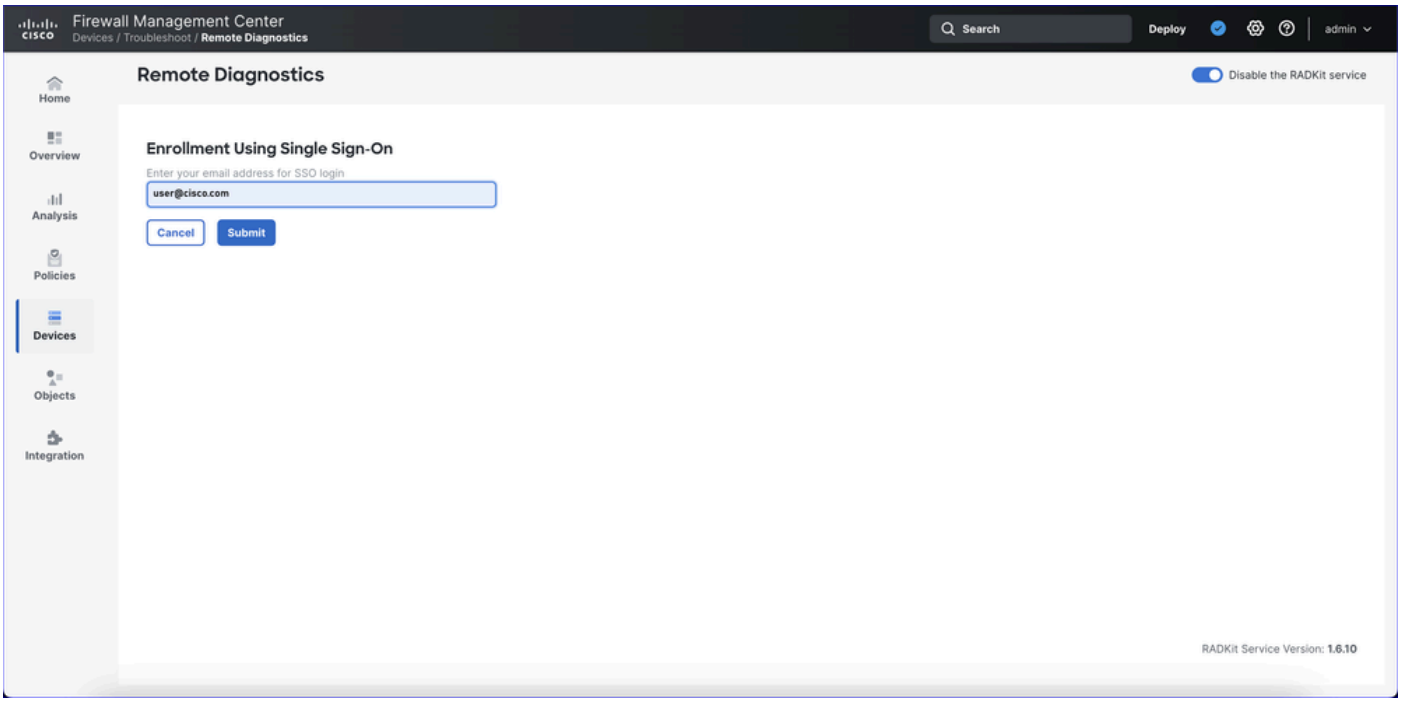
- RADKitサービス有効化プロセスが完了すると、次のページが表示されます。



次のステップは、「Enroll with SSO」ボタンをクリックしてRADKitクラウドに登録することです。

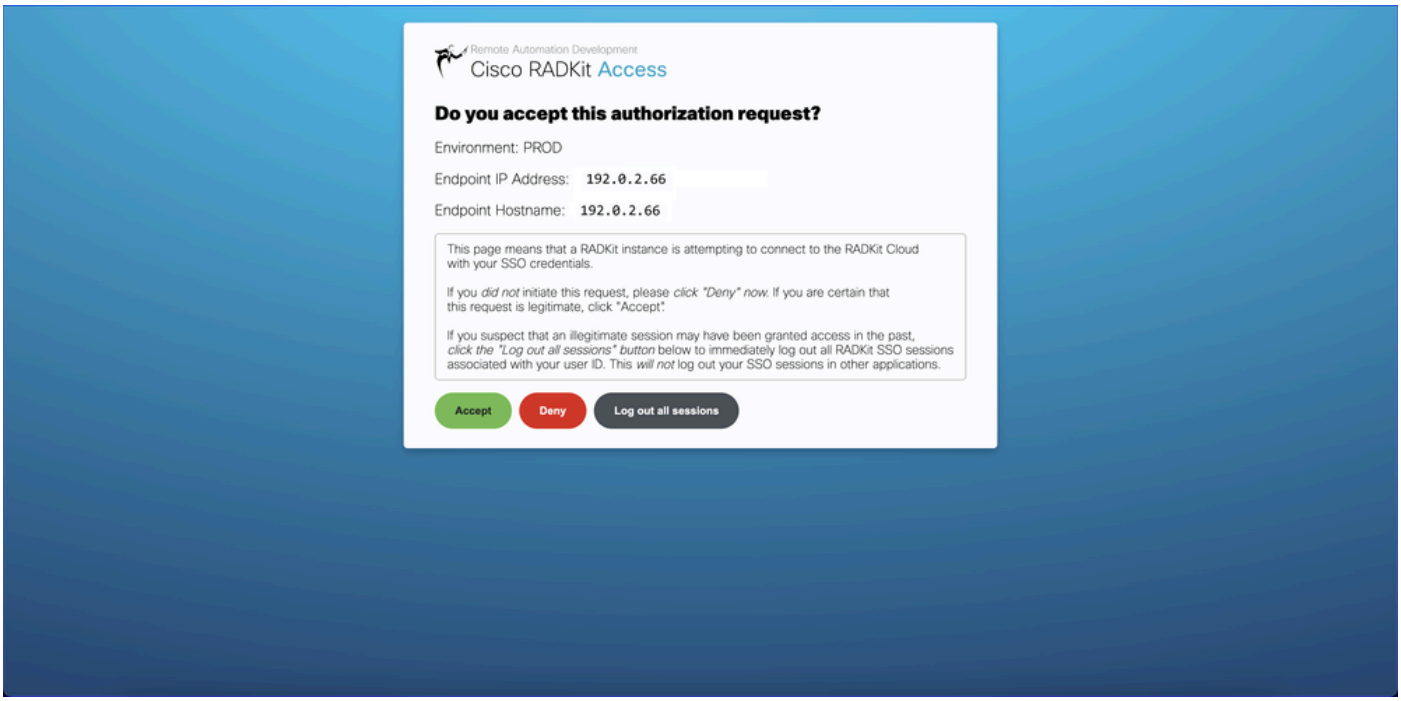
SSOへの登録 – 電子メールアドレスの入力

登録プロセスのステップ1では、RADKitクラウド登録のユーザ電子メールアドレスを入力します。



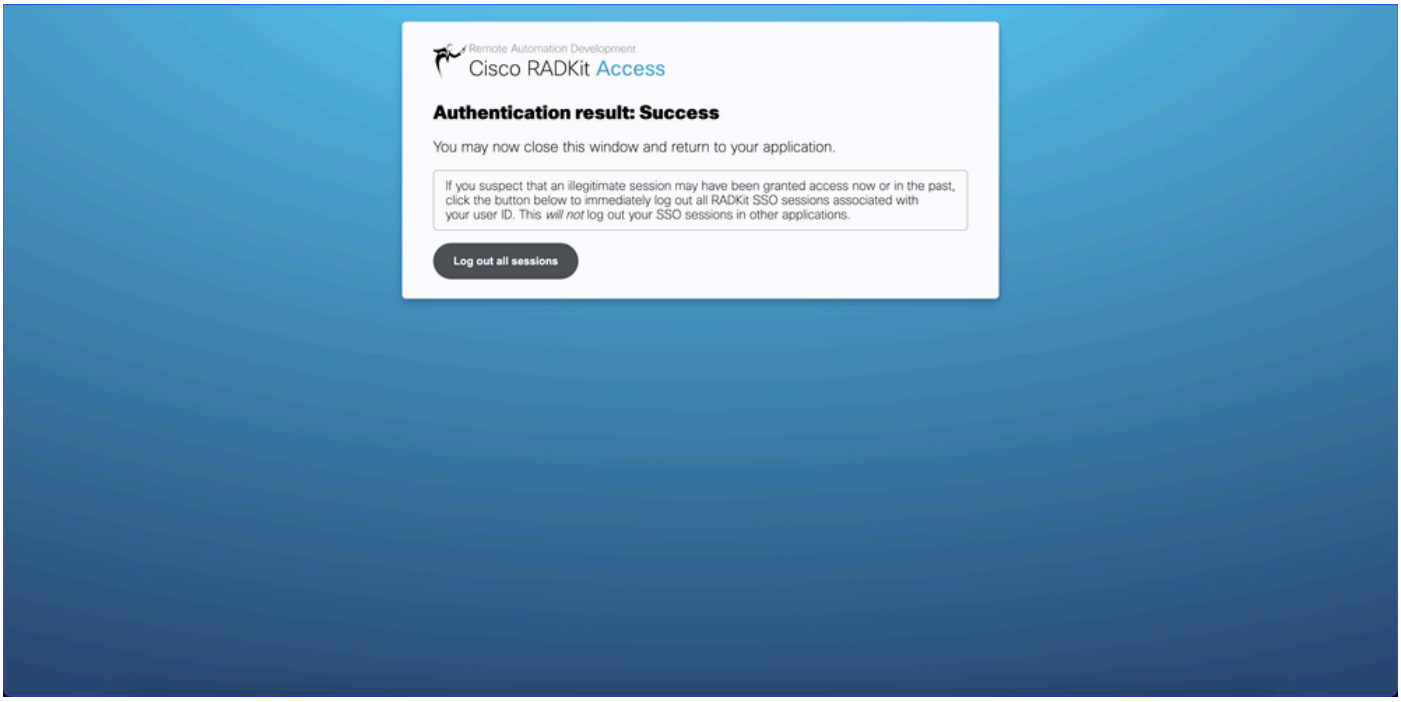
SSOへの登録：認証要求の受け入れ

新しいブラウザタブ（ブラウザの設定によってはウィンドウ）が開きます。Acceptボタンをクリックします。



SSOへの登録 – 認証成功

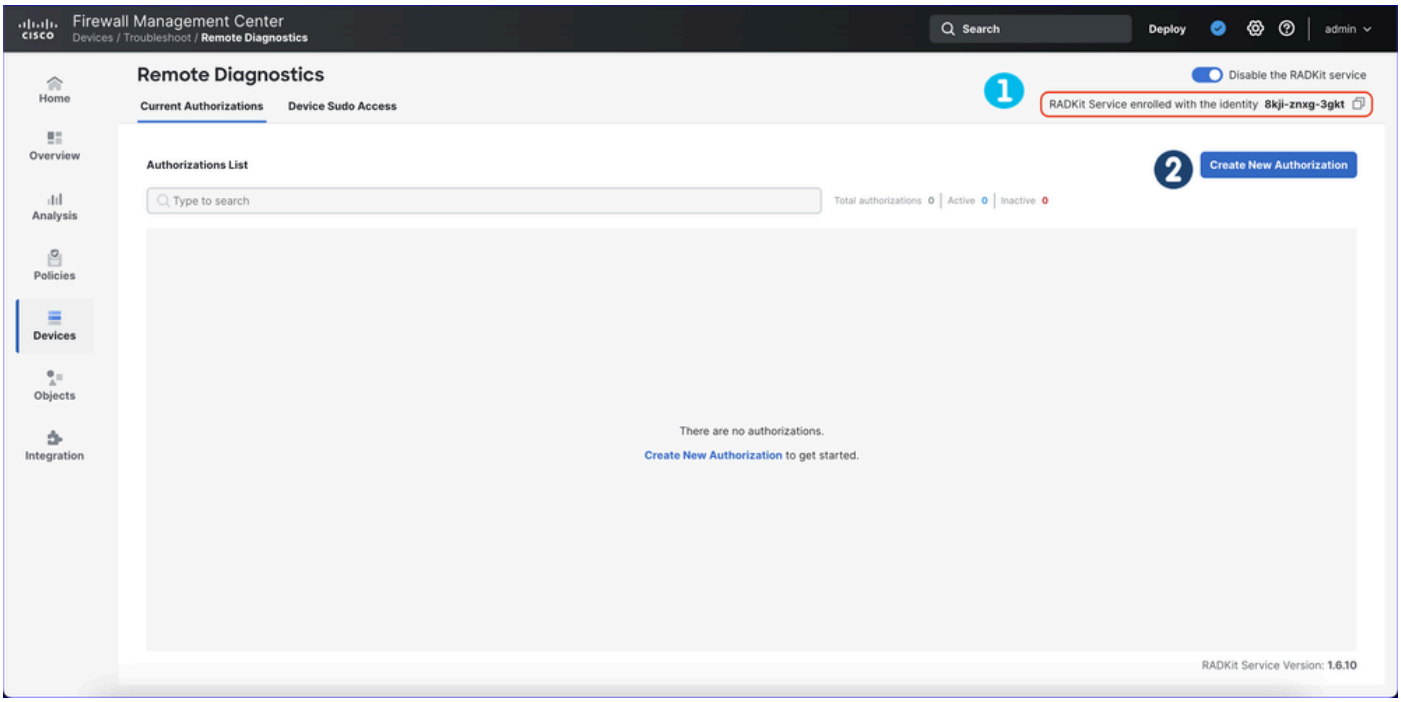
認証に成功した後、ユーザはブラウザタブを閉じて、FMCのリモート診断ページに戻ることができます。



RADKitサービス登録済み

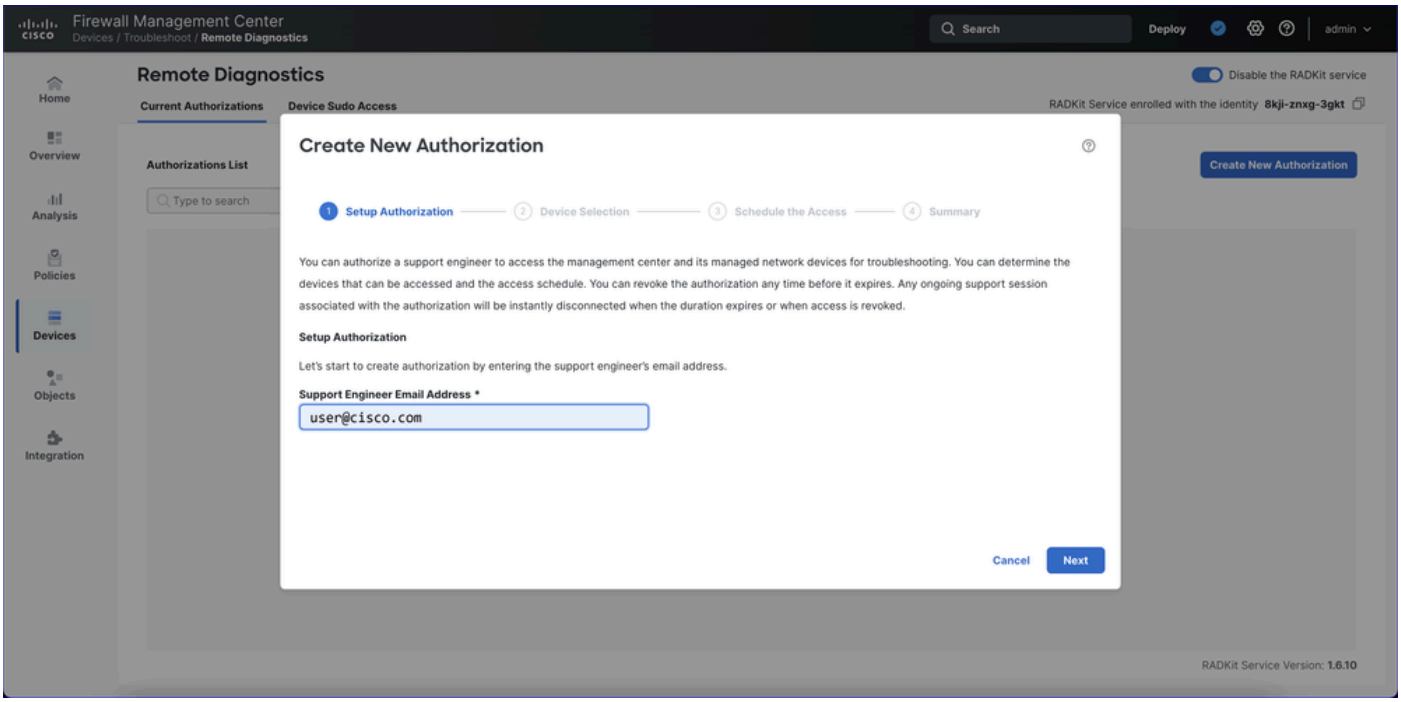
RADKitサービスは、指定されたサービスIDで登録されます（この例では、IDは8kji-znxg-3gktです）。 IDはクリップボードにコピーできます。Cisco TACエンジニアがRADKitクライアントからRADKitサービスに接続できるように、この情報を提供します。

次に、「Create New Authorization」ボタンをクリックして認可を作成します。



新しい許可の作成：手順1

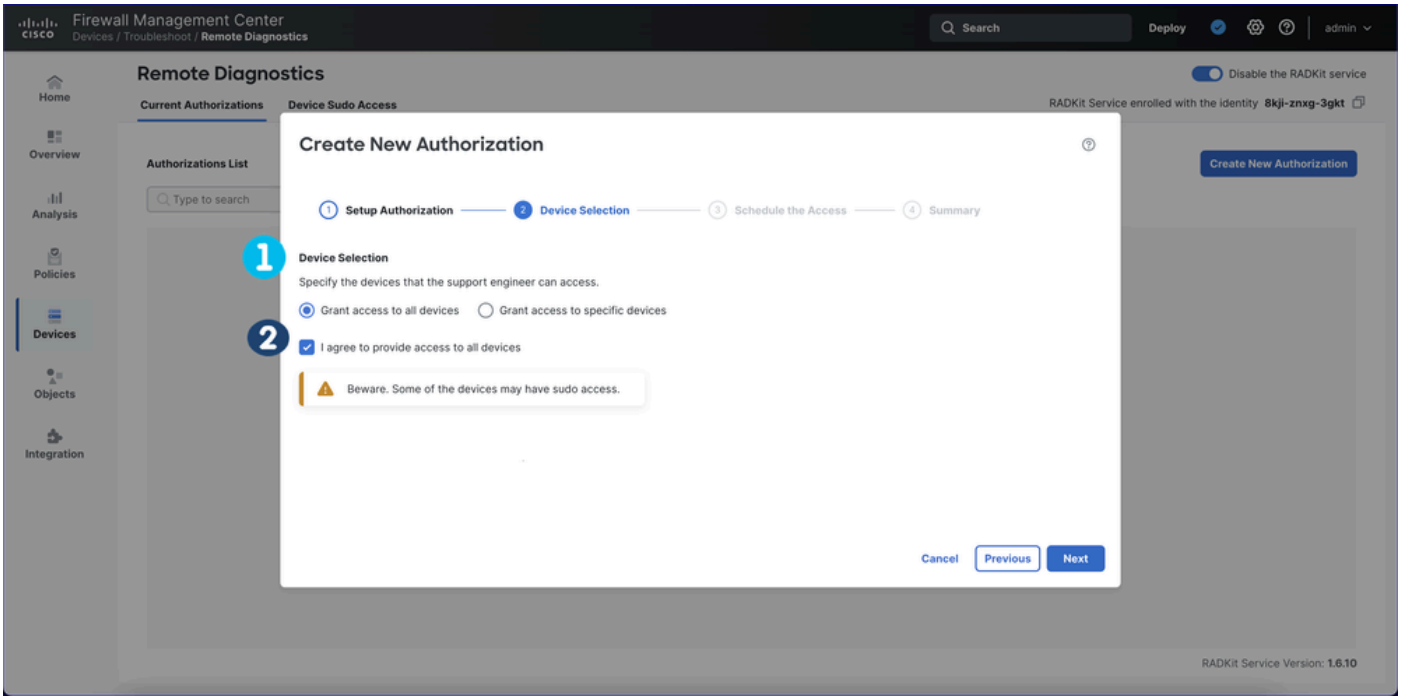
- 新しい承認を作成するには、最初にサポートエンジニアの電子メールアドレスを追加します。
- 新しい許可を作成するには、4つのステップがあります。手順に沿った進行状況が上部に表示されます。



新しい許可の作成：手順2

- ステップ1：サポートエンジニアがアクセスできるデバイスを指定します。または、この例のように、すべてのデバイスにアクセス権を付与します。

- ステップ2：すべてまたは特定のデバイスのオプションボタンをオンにします。特定のデバイスについて、FMCまたはFTD（あるいはその両方）を選択できます。sudoアクセスがデバイスの[sudoアクセス]タブの一部のデバイスに提供する可能性があるという警告に注意してください。チェックボックスをオンにするまで、Nextボタンは有効になりません。
- Sudoアクセスは、後でデバイスのSudoアクセスタブでデバイスごとに提供されます（認可の作成中ではありません）。

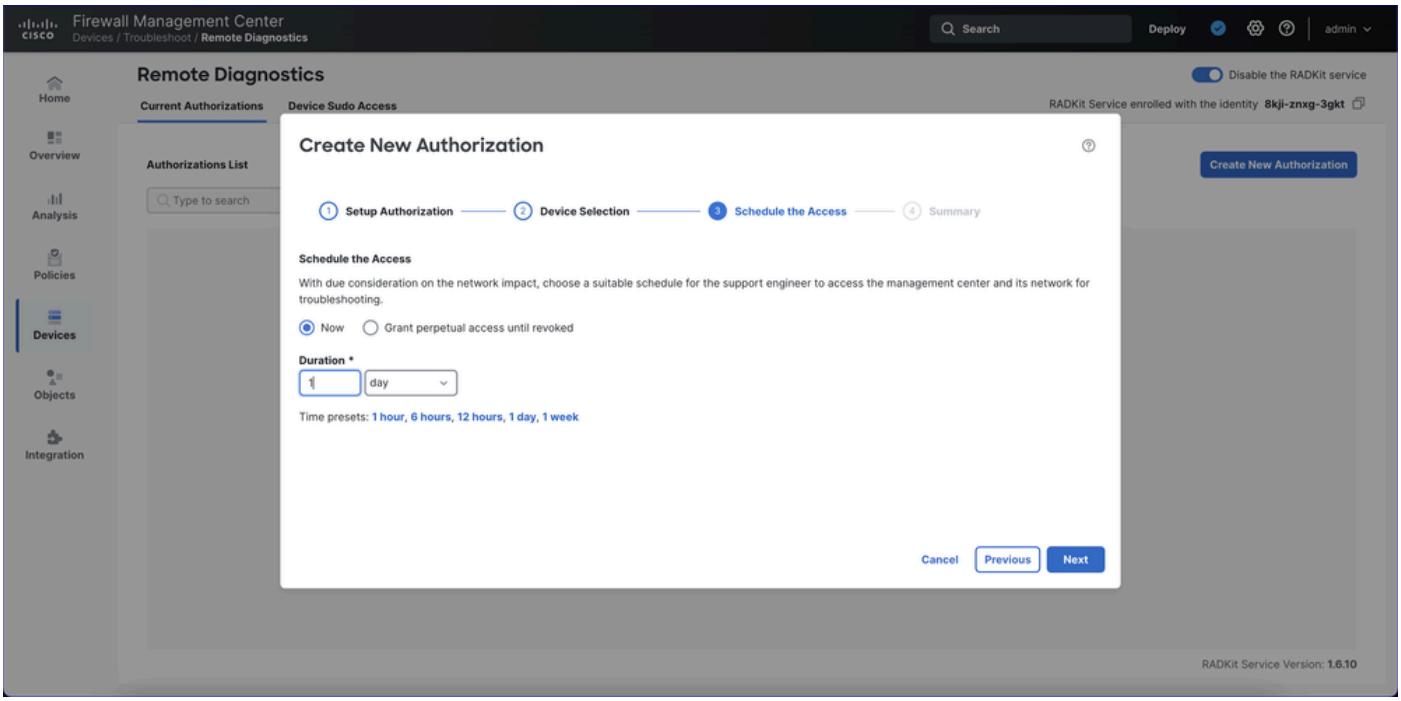


ピッキングデバイスについての注意

- サポートされているビルド上のデバイス（たとえば、初期リリースでは7.7.0デバイス）のみを選択できます。
- 無効で到達不可能なデバイスは選択できません。RADKitはsftunnel(TCP 8305)を使用してデバイスにアクセスします。
  - sftunnelの接続に問題がある場合は機能しませんが、RADKitインベントリには引き続き表示されます。
  - デバイスの電源がオフの場合、デバイスはまったく表示されません。
- HAペアにFMCがある場合は、アクティブ/プライマリだけを追加できます。
- デバイスは、認可の作成/編集時にRADKitインベントリに追加されます。デバイスがFMCから登録解除されると、デバイスの「インベントリ」から削除されます。

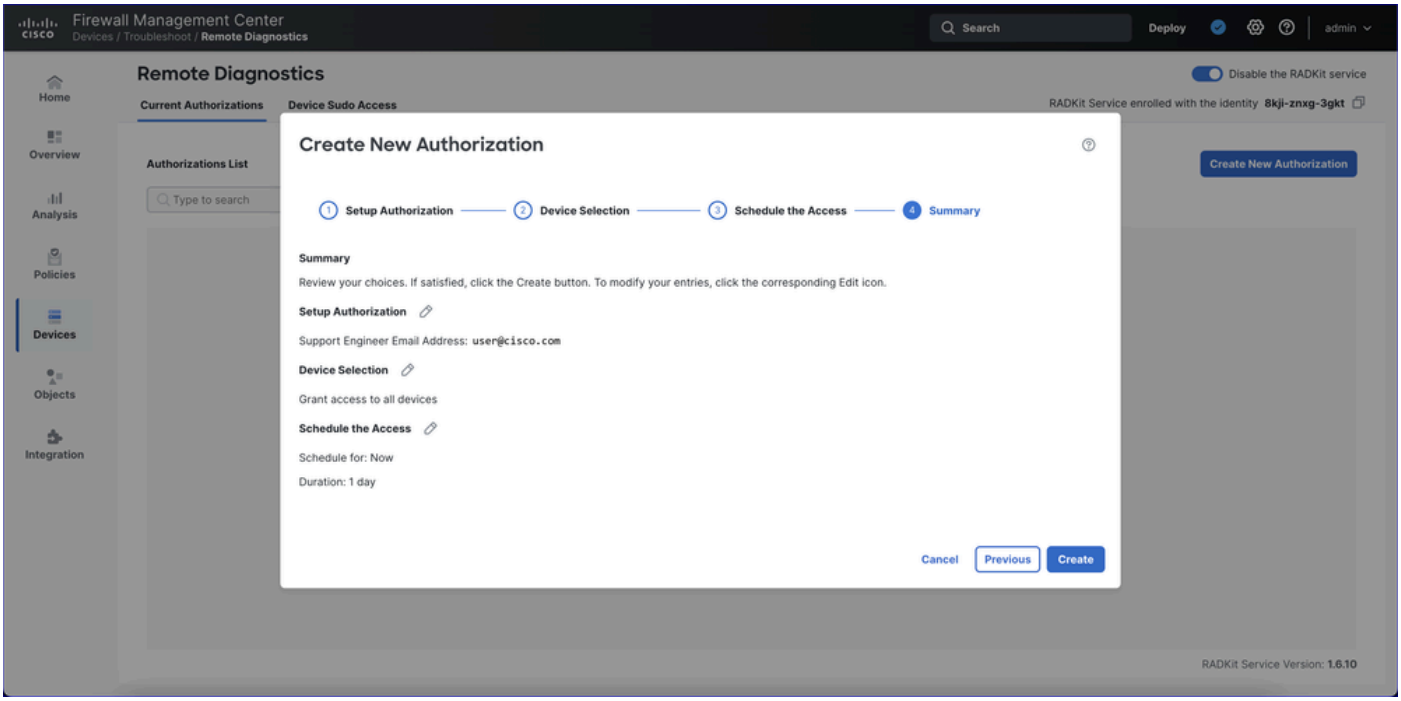
新しい許可の作成：手順3

- ステップ3：サポートエンジニアがデバイスにアクセスする期間を指定します。
- 「今すぐ」を選択して継続時間を指定するか、
- 「失効するまで無制限アクセスを許可する」を選択します。
- デフォルトの期間は1日です。任意の期間を設定できます。また、期間の値も事前に定義されています。



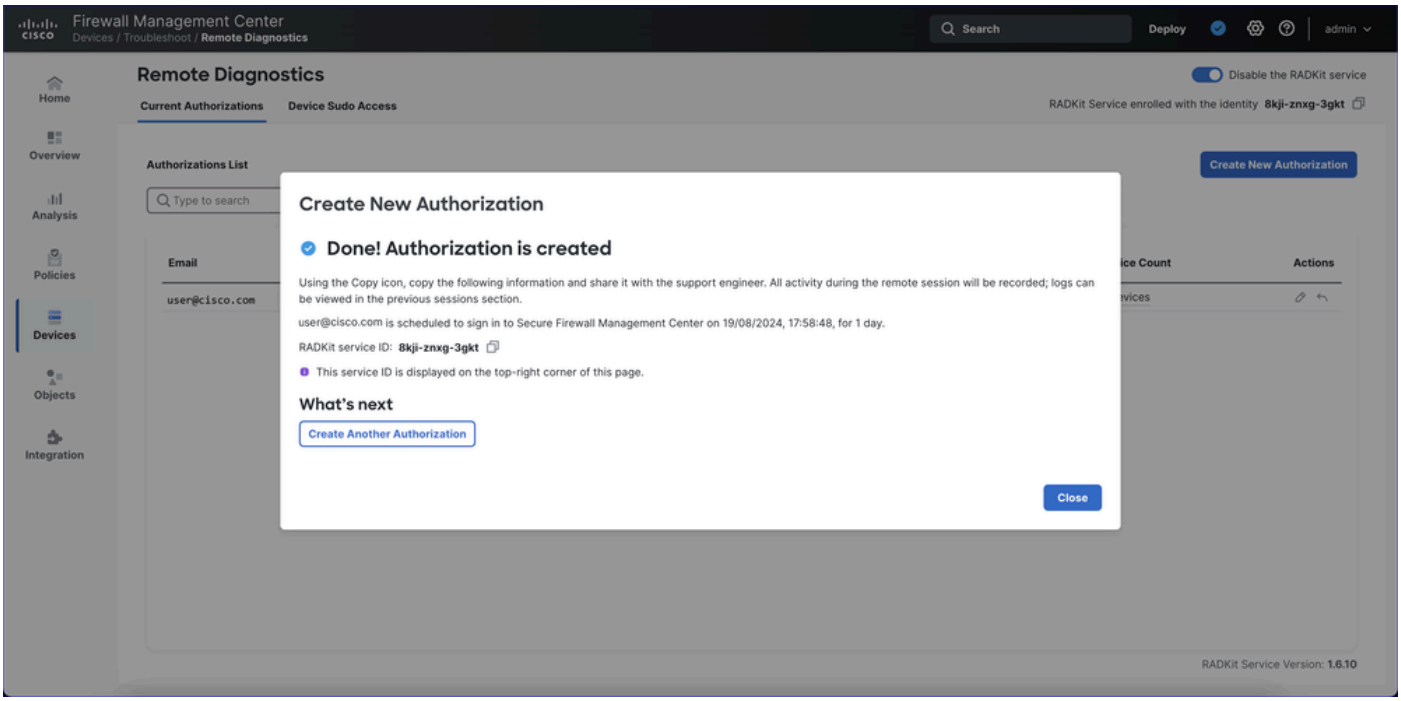
新しい承認の概要の作成

最後のステップは許可の要約です。ここで、ユーザは設定を確認および編集できます。



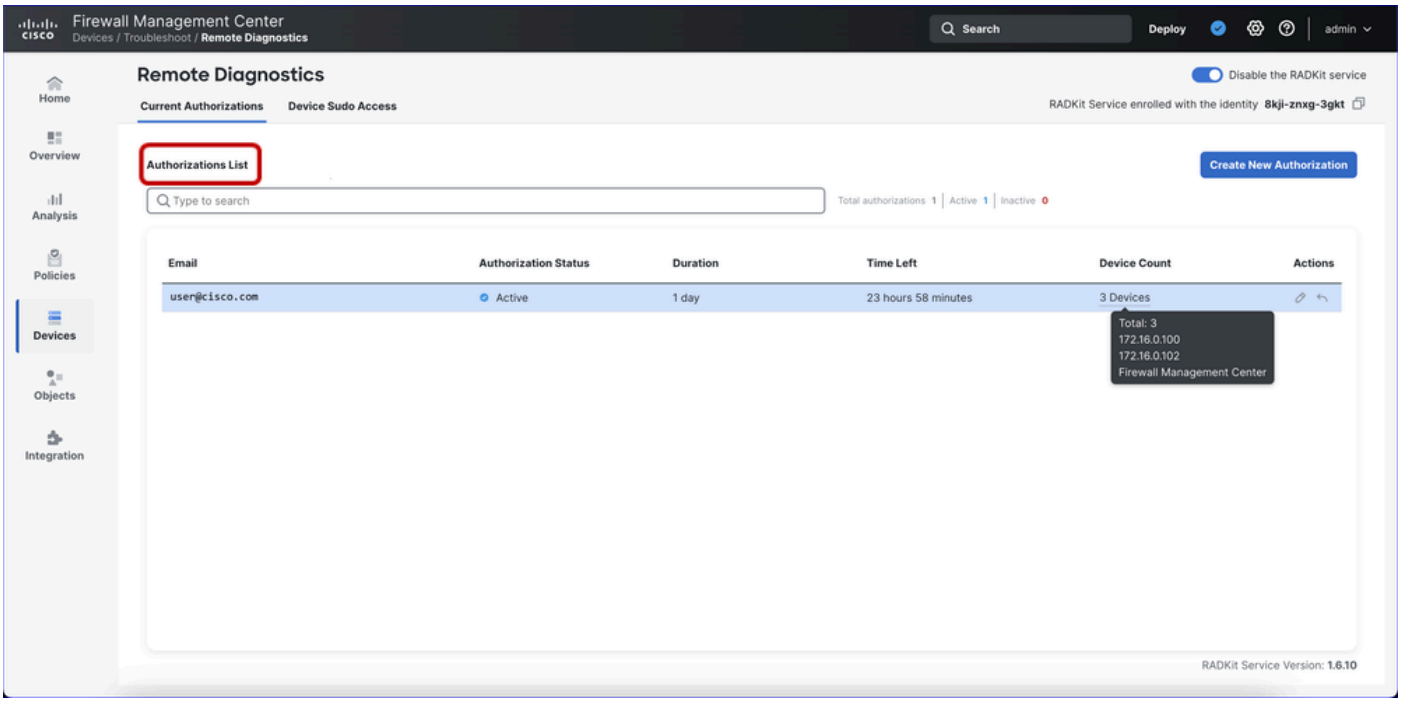
新しい承認の作成が完了しました

許可の作成が完了すると、確認画面が表示されます。



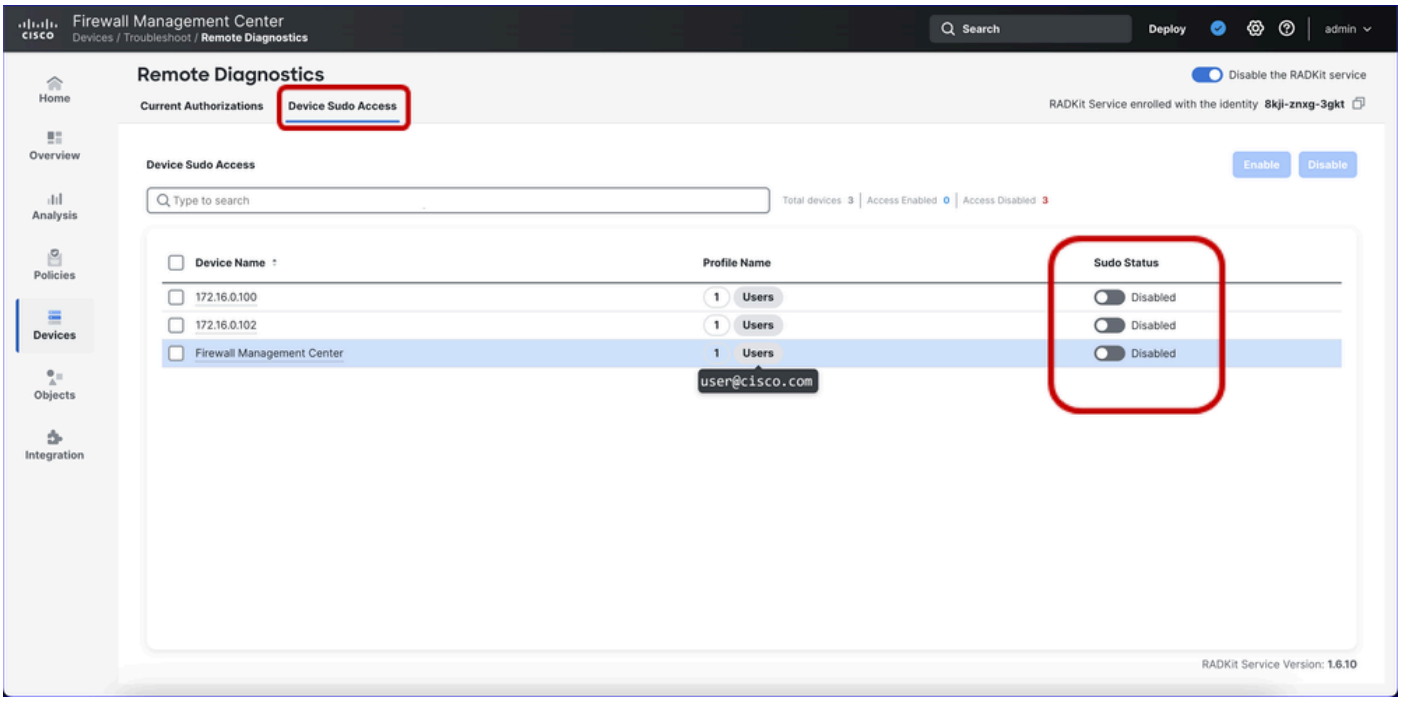
取り消しを含む現在の承認リスト

- 現在の承認のリストが「現在の承認」タブに表示されます。
- アクション（右端の列）は、アクセス権の取り消しと編集の権限です。



デバイスSudoアクセスリスト

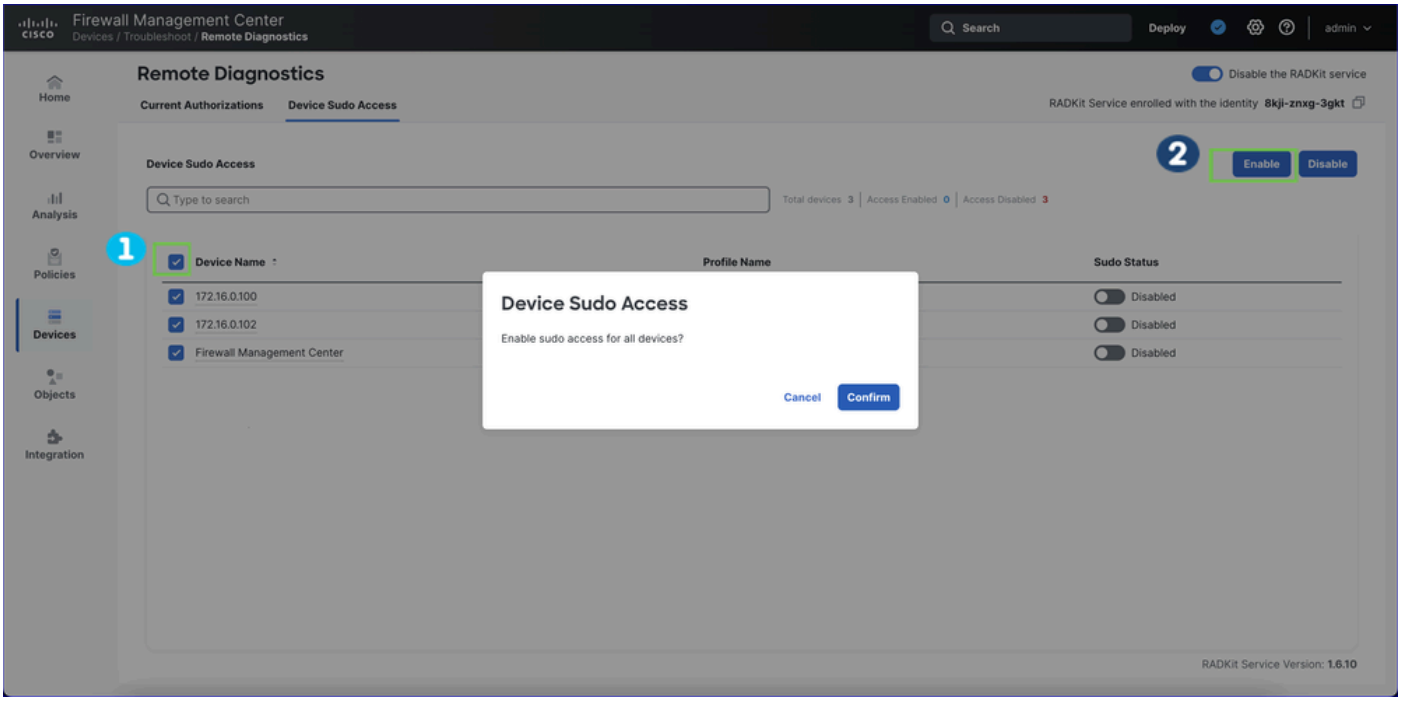
- sudoアクセスが設定されているデバイスのリストは、Device Sudo Accessタブに表示されます。
- sudo accessをオンにするには、右側の列のトグルを使用します。デフォルトではオフになっています。
- また、sudoアクセスを一括で有効/無効にすることもできます。



デバイスのSudoアクセスの有効化の確認

1. Sudoアクセスは、すべてのデバイスに対して、または特定のデバイスに対してのみ有効にすることができます。これを行うには、デバイスを選択して「Enable」ボタンをクリックします。

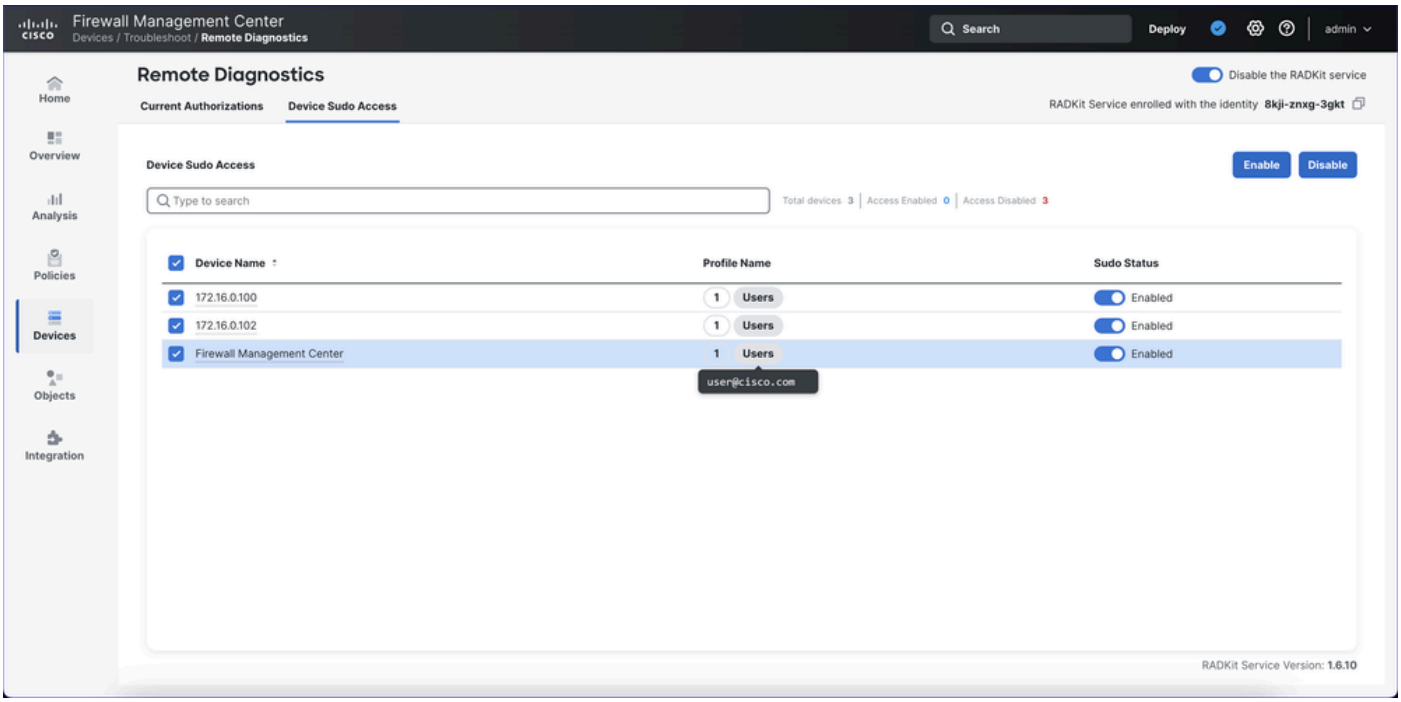
2. 有効にすると、確認ダイアログが表示され、「Confirm」をクリックする必要があります。



Sudoアクセスが有効なデバイス

- デバイスのsudoアクセスを有効または無効にすると、ページ右側のsudo Status列が更新されます。
- サポートエンジニアは、デバイス上でsudo suを実行できます。これはパスワードレスです

。サポートエンジニアはrootパスワードを持つ必要はありません。



その他の注意事項

- FMCユーザがアクセスできるドメイン内のデバイスだけが表示され、リモートアクセスを許可できます。
- FMCがHAの場合：
  - RADKitサービスは、アクティブ/プライマリでのみ有効にできます。
  - セカンダリFMCは現在、RADKitクライアントからアクセスするデバイスとして追加できません。
- 承認は、一度に1人のサポートエンジニアに対してのみ行うことができます。
  - 別のサポートエンジニアがアクセス権を持つ必要がある場合は、追加のエンジニア用に別の権限を作成します。サービスIDは同じです。

FMC REST API

RADKitサービスREST API

RADKitサービスでの作成および読み取り操作をサポートするために、次の新しいURLが導入されました。

- GET: /api/fmc_troubleshoot /v1/domain/{domainUUID}/radkit/services
  - FMCからすべてのRADKitサービスデータを取得します。
- GET: /api/fmc_troubleshoot/v1/domain/{domainUUID}/radkit/services/{id}
  - 指定されたIDからRADKitサービスデータを取得/取得します。
- 投稿:/api/fmc_troubleshoot/v1/domain/{domainUUID}/radkit/services
  - FMCでRADKitサービスを作成します（サービスを有効/無効にします）。

RADKitサービスモデル

RADKitサービスモデルは、次の要素で構成されています。

- 種類
- [id]
- ステータス
- 登録済み
- サービスID
- version


```
{
 "type": "RADKitService",
 "id": "DummyContainerId",
 "status": "RUNNING",
 "isEnrolled": true,
 "serviceId": "8kji-znxg-3gkt",
 "version": "1.6.10"
}
```


シスコサポート：RADKitクライアントの使用状況

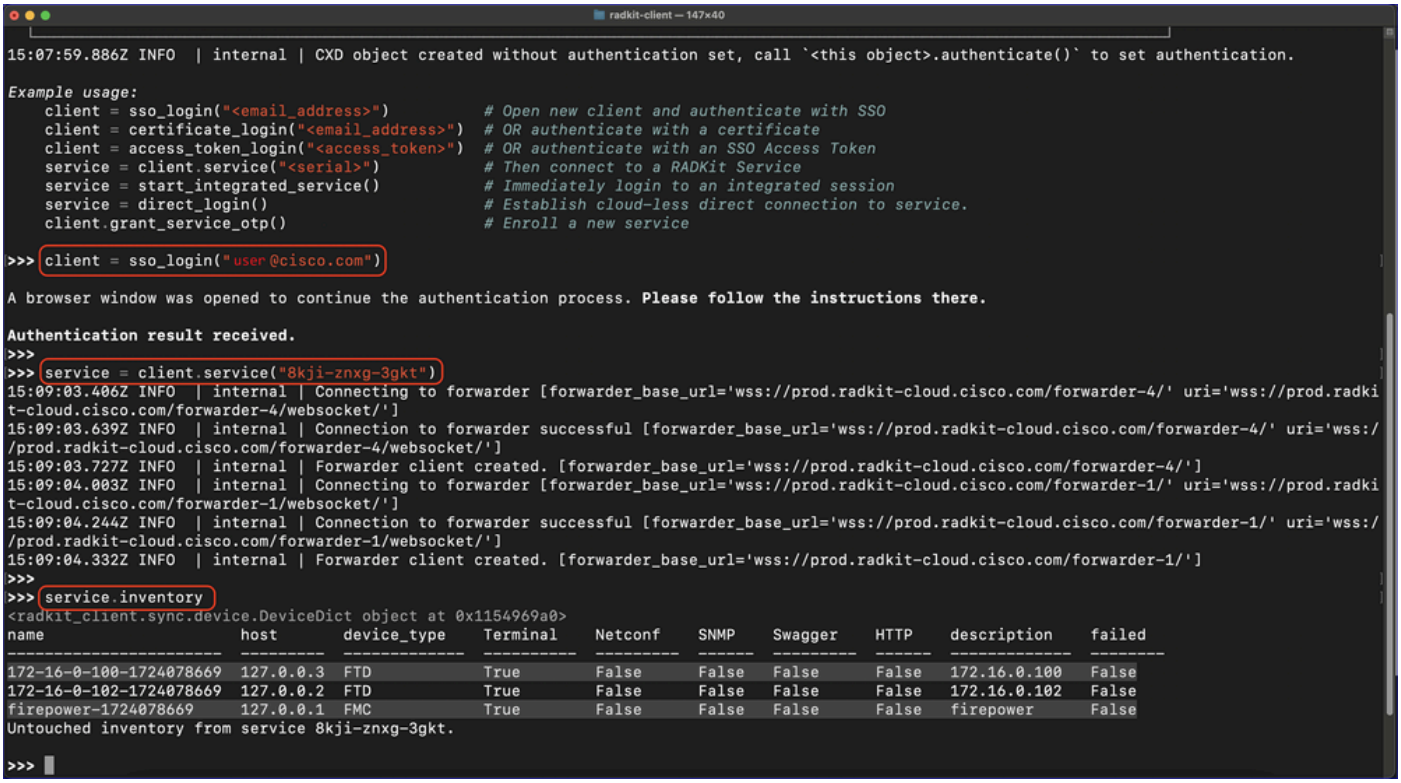
サポート側：RADKit Clientのインストール

- FMC/FTDにアクセスするには、サポートはRADKitクライアントをインストールする必要があります。
  - クライアントは、Windows、Mac、およびLinuxオペレーティングシステムで動作します。
- サポートは、複数のユーザから複数のデバイスにアクセスできます。各RADKit許可には、デバイスの独自の「インベントリ」があります。
  - サポートがアクセスするユーザデバイスインベントリごとに、RADKitサービスIDが必要です。
  - 単一のインベントリの場合は、FMCと管理対象FTDの両方に対して、ユーザがアクセスを許可するときに指定したとおりにRADKitクライアントからアクセスが可能です。

RADKitクライアントの入手とインストール

RADKitクライアントはhttps://radkit.cisco.com/downloads/release/からローカルにインストールでき、次にコマンドradkit-clientを使用して端末から起動します。

インストーラは、Windows、MacOS、およびLinuxで使用できます。



ログインコマンドを使用したRADKitクライアントのスクリーンショット（詳細は次のセクションを参照）

RADKitクライアントのログインコマンド

- FMCでの認証時にユーザが入力した電子メールアドレスを使用します。
- RADKitクライアントがログインし、指定されたサービスIDコマンドに接続します。RADKitサービスID（この例では8abc-znxg-3abc）は、ファイアウォール管理者がFMCで表示する内容と一致する必要があります。

<#root>

>>>

client = sso_login("user@cisco.com")

A browser window was opened to continue the authentication process.

Please follow the instructions there.

```
Authentication result received.


>>>
service = client.service("8abc-znxg-3abc")


15:09:03.639Z INFO  | internal | Connection to forwarder successful [forwarder_base_url='wss://prod.rad
15:09:03.727Z INFO  | internal | Forwarder client created. [forwarder_base_url='wss://prod.radkit-cloud
15:09:04.244Z INFO  | internal | Connection to forwarder successful [forwarder_base_url='wss://prod.rad
15:09:04.332Z INFO  | internal | Forwarder client created. [forwarder_base_url='wss://prod.radkit-cloud
```

RADKitクライアントサービスインベントリコマンド

リモートユーザ（Cisco TACエンジニア）がアクセスを許可されているインベントリをリストするコマンド：

```
<#root>

>>>
service.inventory


<radkit_client.sync.device.DeviceDict object at 0x1154969a0>
name                      host       device_type    Terminal    Netconf    SNMP    Swagger    HTTP    de
------------------------  ---------  -------------  ----------  ---------  ------  ---------  ------  --
172-16-0-100-1724078669   127.0.0.3  FTD            True        False      False   False      False   17
172-16-0-102-1724078669   127.0.0.2  FTD            True        False      False   False      False   17
firepower-1724078669      127.0.0.1  FMC            True        False      False   False      False   fi
Untouched inventory from service 8kji-znxg-3gkt.
```

インベントリ内のデバイスに対するフィルタコマンドがあります（次のセクション）。 左側の列の名前を使用して、デバイスとのインタラクティブセッションを開始します（次のセクションのコマンド）。

ヒント：インベントリが古くなっている場合は、次のコマンドを使用して更新できます。
>>> service.update_inventory()

RADKitクライアント：デバイスのフィルタ

インベントリ内のデバイスをフィルタリングするコマンド：

```
<#root>
```

```
>>>
ftds = service.inventory.filter(attr='name',pattern='172-16-0’)


>>>
ftds


<radkit_client.sync.device.DeviceDict object at 0x111a93130>
name host device_type Terminal Netconf SNMP Swagger HTTP description failed
---------------------- --------- ------------ ---------- --------- ------ --------- ------ ----------
172-16-0-100-1724078669 127.0.0.3 FTD True False False False False 172.16.0.100 False
172-16-0-102-1724078669 127.0.0.2 FTD True False False False False 172.16.0.102 False
2 device(s) from service 8kji-znxg-3gkt.
```

RADKitクライアントデバイスのインタラクティブセッションコマンド

前の「service.inventory」コマンドから取得した「firepower-1724078669」という名前のデバイス（この場合はFMC）のインタラクティブセッションを起動する。

```
<#root>
>>>
service.inventory["firepower-1724078669"].interactive()


08:56:10.829Z INFO | internal | Starting interactive session (will be closed when detached)
08:56:11.253Z INFO | internal | Session log initialized [filepath='/Users/use/.radkit/session_logs/clie

Attaching to firepower-1724078669 ...
Type: ~. to terminate.
~? for other shortcuts.
When using nested SSH sessions, add an extra ~ per level of nesting.

Warning: all sessions are logged. Never type passwords or other secrets, except at an echo-less passwor


Copyright 2004-2024, Cisco and/or its affiliates. All rights reserved.
Cisco is a registered trademark of Cisco Systems, Inc.
All other trademarks are property of their respective owners.

Cisco Firepower Extensible Operating System (FX-OS) v82.17.0 (build 170)
Cisco Secure Firewall Management Center for VMware v7.7.0 (build 1376)
```

RADKitクライアントがデバイス上でコマンドを実行する

デバイスでコマンドを実行します。

```
<#root>

>>>

result = ftds.exec(['show version', 'show interface'])


>>>
>>>

result.status


<RequestStatus.SUCCESS: 'SUCCESS'>

>>>
>>>

result.result['172-16-0-100-1724078669']['show version'].data | print


> show version
-------------------[ firepower ]--------------------
Model : Cisco Secure Firewall Threat Defense for VMware (75) Version 7.7.0 (Build 1376)
UUID : 989b0f82-5e2c-11ef-838b-b695bab41ffa
LSP version : lsp-rel-20240815-1151
VDB version : 392
----------------------------------------------------
```

デバイスから詳細を取得する

このインベントリを考慮すると：

```
<#root>

>>>

service.inventory


[READY] <radkit_client.sync.device.DeviceDict object at 0x192cdb77110>
name                          host      device_type   Terminal   Netconf   SNMP   Swagger   HTTP   desc
----------------------------- --------- ------------- ---------- --------- ------ --------- ------ ----
10-62-184-69-1743156301       127.0.0.4 FTD           True       False     None   False     False  10.6
fmc1700-1-1742391113          127.0.0.1 FMC           True       False     None   False     False  FMC1
ftd3120-3-1743154081          127.0.0.2 FTD           True       False     None   False     False  FTD3
ftd3120-4-1743152281          127.0.0.3 FTD           True       False     None   False     False  FTD3
```

FTDデバイスから「show version」の詳細を取得するには、次のコマンドを実行します。

<#root>

>>>

command = "show version"

>>>

ftds = service.inventory.filter("device_type","FTD").exec(command).wait()

>>>
>>>

 # Print the results

>>>

for key in ftds.result.keys():

...

print(key)

...

ftds.result.get(key).data | print

...

<- Press Enter twice

```
ftd3120-3-1743154081
> show version
-------------------[ FTD3100-3 ]--------------------
Model : Cisco Secure Firewall 3120 Threat Defense (80) Version 7.7.0 (Build 89)
UUID : 123a456a-cccc-bbbb-aaaa-a123456abcde
LSP version : lsp-rel-20250327-1959
VDB version : 404
----------------------------------------------------

>
10-62-184-69-1743156301
> show version
-----------------[ KSEC-FPR1010-10 ]-----------------
Model : Cisco Firepower 1010 Threat Defense (78) Version 7.7.0 (Build 89)
UUID : 123a456a-cccc-bbbb-aaaa-a123456abcde
LSP version : lsp-rel-20250327-1959
VDB version : 404
----------------------------------------------------

>
ftd3120-4-1743152281
> show version
-------------------[ FTD3100-4 ]--------------------
Model : Cisco Secure Firewall 3120 Threat Defense (80) Version 7.7.0 (Build 89)
UUID : 123a456a-cccc-bbbb-aaaa-a123456abcde
LSP version : lsp-rel-20250327-1959
VDB version : 404
```

```
--------------------------------------------------

>
```

別のアプローチ：

<#root>

```
>>> # Get the FTDs. This returns a DeviceDict object:
...
ftds = service.inventory.filter("device_type","FTD")


>>> # Access the dictionary of devices from the _async_object attribute
...
devices_obj = ftds.__dict__['_async_object']


>>> # Extract the 'name' from each AsyncDevice object
...
names = [device.name() for device in devices_obj.values()]


>>> # Get the 'show version' output from all FTD devices:
...
command = "show version"

...
show_ver_ftds = []

...
for name in names:

...
ftd = service.inventory[name]

...
req = ftd.exec(command)

...
req.wait(30)
 # depending on the number of devices you might need to increase the timeout value
...
show_ver_ftds.append(req.result.data)
```

```
>>> # Print the inventory device name + 'show version' output from each device:
...
for name, show_version in zip(names, show_ver_ftds):


...
print(f"Inventory name: {name}")


...
print(show_version[2:-2]) # Remove the leading '> ' and trailing ' \n>'


...
print("\n")


Inventory name: ftd3120-3-1743154081
show version
-------------------[ FTD3100-3 ]--------------------
Model : Cisco Secure Firewall 3120 Threat Defense (80) Version 7.7.0 (Build 89)
UUID : 123a456a-cccc-bbbb-aaaa-a123456abcde
LSP version : lsp-rel-20250327-1959
VDB version : 404
----------------------------------------------------


Inventory name: ftd3120-4-1743152281
show version
-------------------[ FTD3100-4 ]--------------------
Model : Cisco Secure Firewall 3120 Threat Defense (80) Version 7.7.0 (Build 89)
UUID : 123a456a-cccc-bbbb-aaaa-a123456abcde
LSP version : lsp-rel-20250327-1959
VDB version : 404
----------------------------------------------------


Inventory name: 10-62-184-69-1743156301
show version
----------------[ KSEC-FPR1010-10 ]-----------------
Model : Cisco Firepower 1010 Threat Defense (78) Version 7.7.0 (Build 89)
UUID : 123a456a-cccc-bbbb-aaaa-a123456abcde
LSP version : lsp-rel-20250327-1959
VDB version : 404
----------------------------------------------------
```

デバイスからのファイルの取得

- Cisco TACエンジニアは、RADKitクライアントを介してデバイスにSSHで接続し、トラブルシューティングファイルの生成を含むさまざまな操作を実行できます。

シスコサポート：RADKitコンソール

RADKit Network Consoleの使用

- RADKitクライアントを使用する代わりに、Cisco TACサポートエンジニアはRADKit Network Consoleを使用することもできます。Network ConsoleはRADKit Clientの一部です。
- RADKitネットワークコンソールは、基本的なRADKitクライアント機能にシンプルなコマンドラインインターフェイス(CLI)を提供する機能です。RADKit Serviceインスタンスに素早く接続し、インタラクティブなセッションを確立し、面倒な作業や最小限のトレーニングでファイルのダウンロードとアップロードを行うためのものです。
- コマンドラインを使用してネットワークコンソールを起動します： radkit-network-console
- 詳細については、RADKitのドキュメントを参照してください。

アップグレードと下位互換性

7.7以降へのアップグレード

- RADKitサービスはSecure Firewall 7.7.0で追加されました。
  - バージョン7.7.0+にアップグレードされたデバイスには、RADKitサービスに必要な設定があります。

サポートされていないFTDの経験

- この機能が動作するには、FMCとFTDには最低限バージョン7.7.0が必要です（バージョン7.7未満のFTDは7.7 FMC RADKit許可に追加できません）。
- 7.7.0以外の登録済みFTDは、許可を有効にするためのピッキングには使用できません。

トラブルシューティング

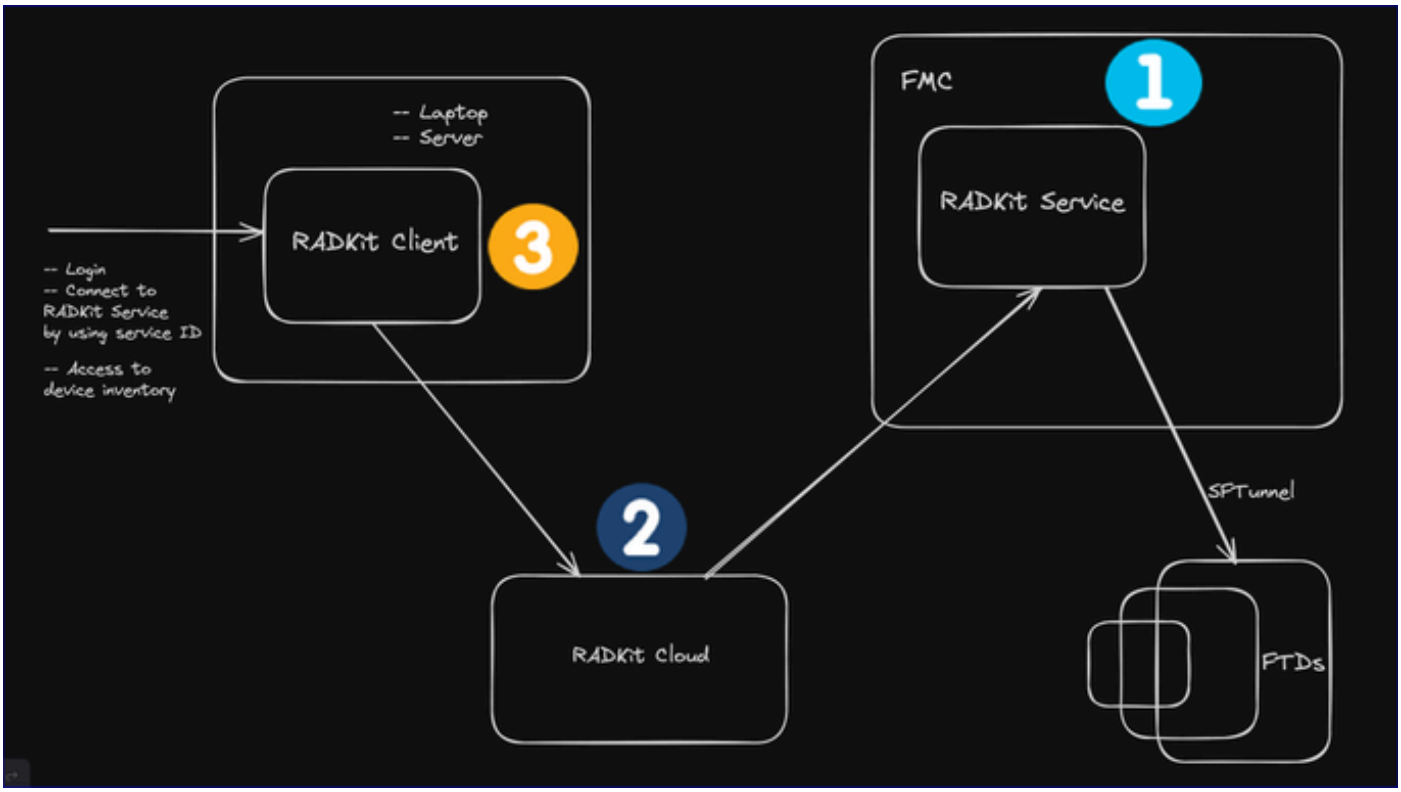
診断概要

トラブルシューティングポイント

1. ブラウザ開発ツールとFMCのログを使用して、FMCで何が起きているかを確認します。

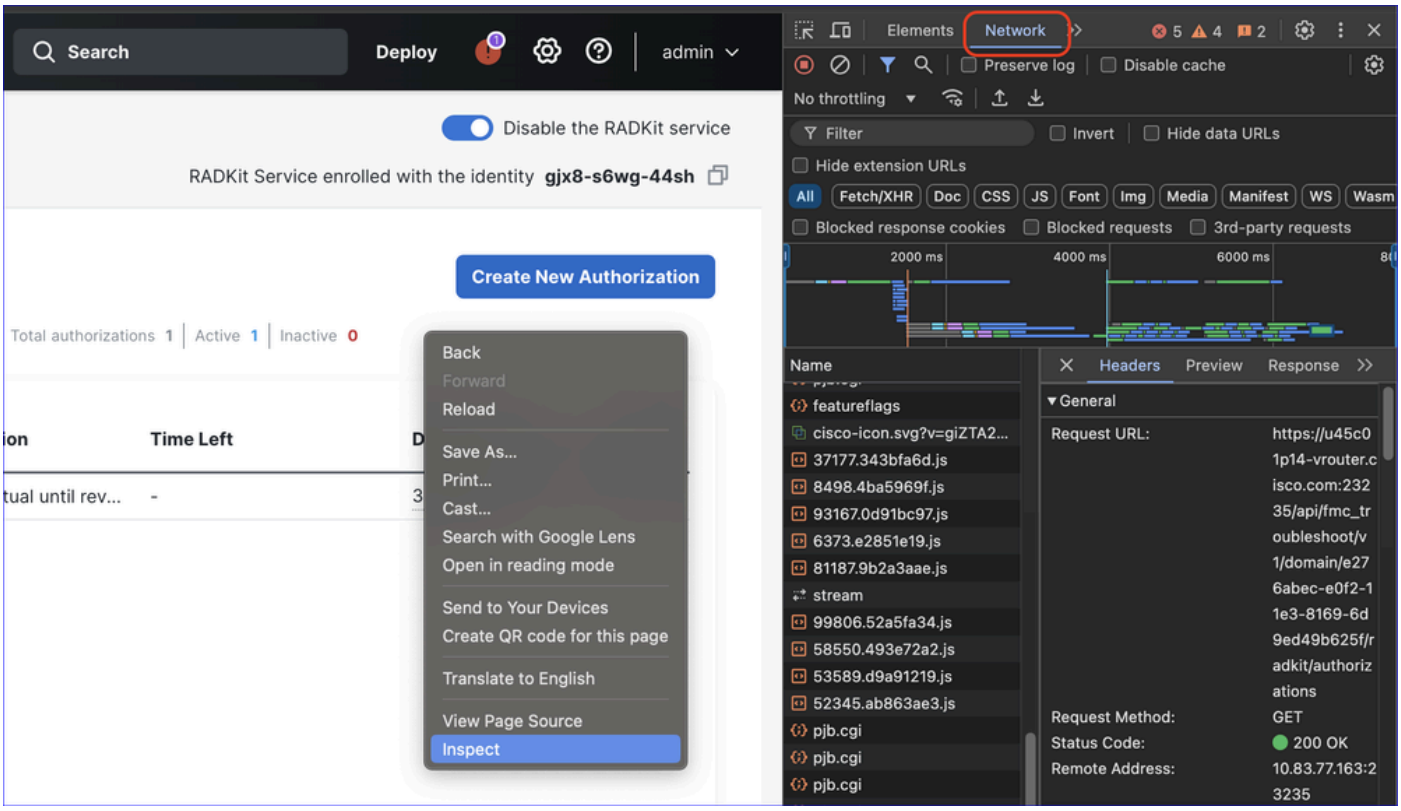
2. FMC上のRADKitサービス、RADkit Cloud、RADKitクライアント間の通信の問題については、RADKitクライアントロギングを参照してください。

3. RADKitクライアント。



トラブルシューティング方法：ブラウザの開発者ツール

- ブラウザのDeveloper ToolsのNetworkタブには、ページで実行されたAPIコールが表示されます。これは、FMCの問題をトラブルシューティングする際に使用できます。これは、ページを右クリックして[Inspect]をクリックすることで起動できます。
- NetworkタブでAPIコールステータスコードと応答プレビューを確認します。



RADKit Service Go Middleware API

Go Middleware for the RADKit統合では、FMC APIエクスプローラでは公開されていないAPI呼び出しを使用します。Go Middleware APIのログは、/var/log/auth-daemon.logで入手できます。Go Middlewareが実行する機能は次のとおりです。

- シングルサインオンプロセスでRADKitクラウドにRADKitサービスを登録します。
- すべてのリモートRADKitユーザの許可と関連デバイスのリストを取得します。
- 電子メールを使用して、特定のリモートRADKitユーザ許可および関連するデバイスを取得します。
- リモートRADKitユーザ許可を作成し、指定した期間にデバイス（すべてのデバイスまたは選択したデバイスのリスト）にアクセスする権限を付与します。
- リモートRADKitユーザ許可を変更します。
- リモートRADKitユーザ許可を削除します。

RADKitサービスをトラブルシューティングするためのログ

- 一般的なFMCのログ：FMC sshセッションからのpigtailコマンド。
- Go Middleware API:/var/log/auth-daemon.log
- RADKitおよびauth-daemonプロセスのデータを含むログ：

/var/log/process_stdout.log

/var/log/process_stderr.log

これらのログはすべて、FMC/FTDのトラブルシューティングに含まれています。

- 内部RADKitサービスログ：/var/lib/radkit/logs/service/
- デバイス（FMCおよびFTD）のRADKitクライアントから実行された操作のログ：/var/lib/radkit/session_logs/service

Cisco TACに提出すべきログ

- エラーのスクリーンショット
- 問題の説明。
- 再現手順。
- エラーを含むピグテールと/var/log/auth-daemon.logログの抽出。

アクセス監視

FMC監査ログには、アクセスを許可されたユーザとアクセスを許可されたユーザのログが記録されます。

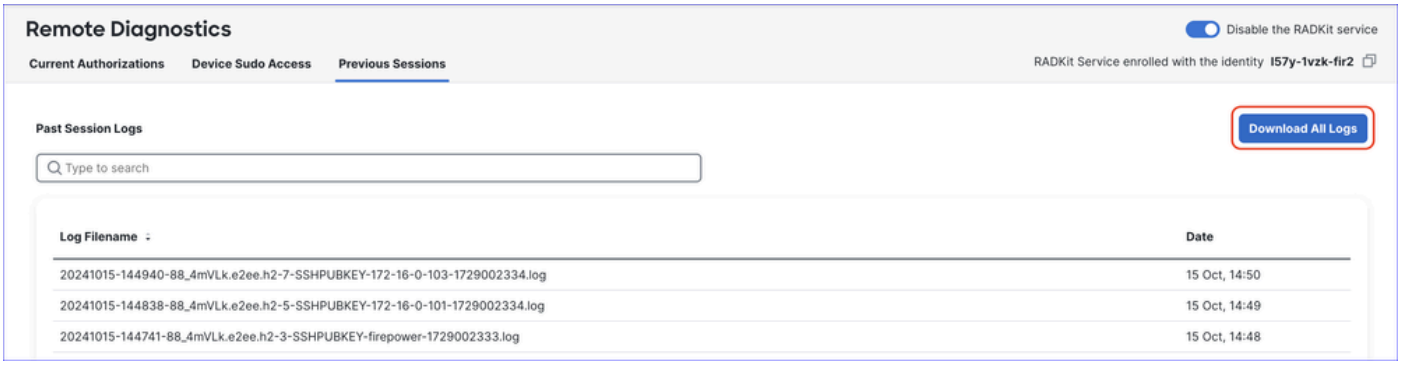
RADKitセッションログ

デバイス（FMCおよびFTD）上のRADKitクライアントから実行された操作のRADKitセッションログは、FMCの/var/lib/radkit/session_logs/service:

- ログはRADKitサービス自体からのものです。
- これらのログは、トラブルシューティングバンドルに含まれています。
- ログはUIからもアクセスできます（次のセクションを参照）。
- 複数のセッション・ログ・ファイルが存在します。セッションごとに1つのログ・ファイルがあります。

RADKitの以前のセッションログ

RADKitクライアントから実行されたデバイス操作のRADKitセッションログを、アーカイブとしてダウンロードできます。このアーカイブには、「Previous Sessions」タブで「Download All Logs」ボタンをクリックすると、すべてのログが含まれます。



トラブルシューティングのウォークスルーの問題例

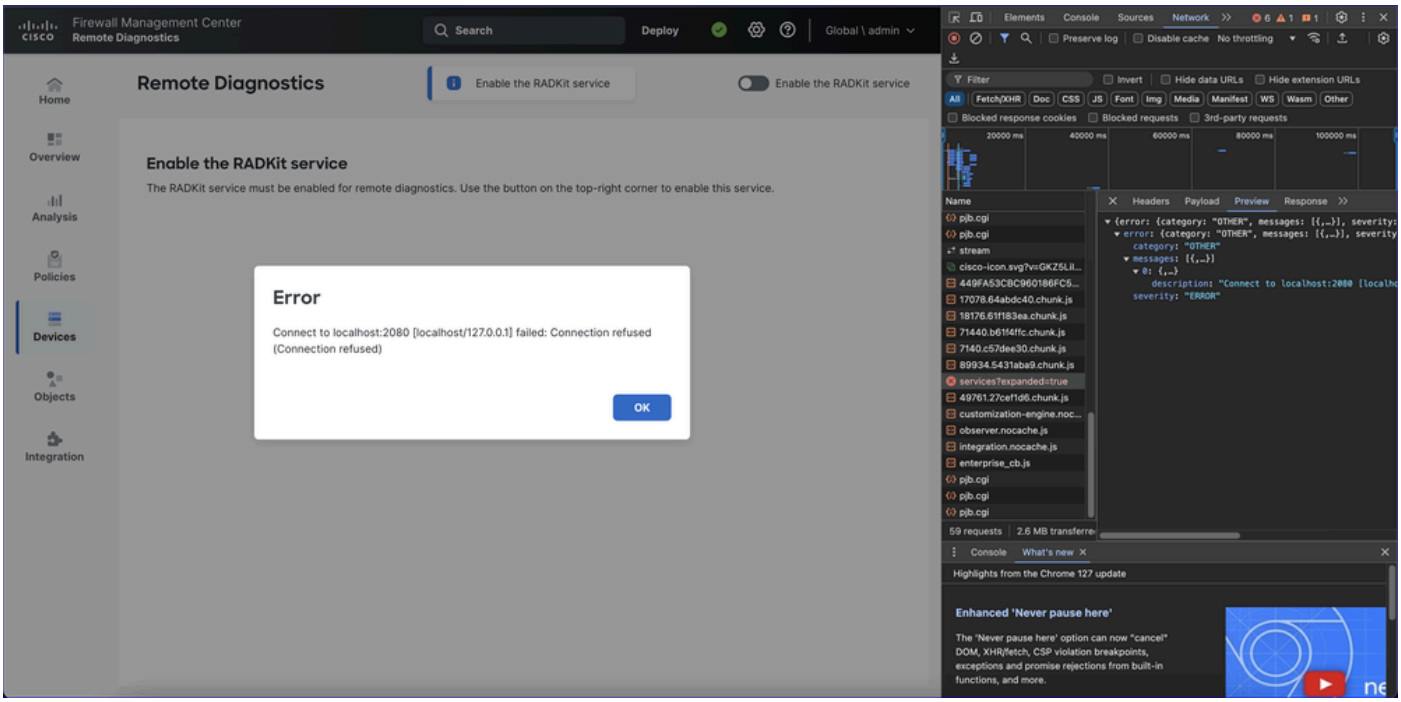
トラブルシューティングの例

「Connect to localhost:2080 [localhost/127.0.0.1] failed: Connection refused (Connection refused)」のようなエラーが発生した場合は、FMCのSSHセッションからauth-daemonを再起動してみてください。

<#root>

root@firepower:~$

sudo pmtool restartbyid auth-daemon



テレメトリ

テレメトリの出力は、この機能のために追加されました。

```
"remoteDiagnostics" : {
   "isRemoteDiagnosticsEnabled": 0 // 0 = false , 1 = true
}
```

FAQ

FAQ：ログインと登録

Q. FMCにインターネットへの直接アクセスがない場合、登録はプロキシで機能しますか。

A.はい。登録プロセスで使用されるprod.radkit-cloud.cisco.comへのアクセス権がプロキシにある場合はアクセスできます。

Q.ユーザは、このサービスに独自のIdPを使用できますか。

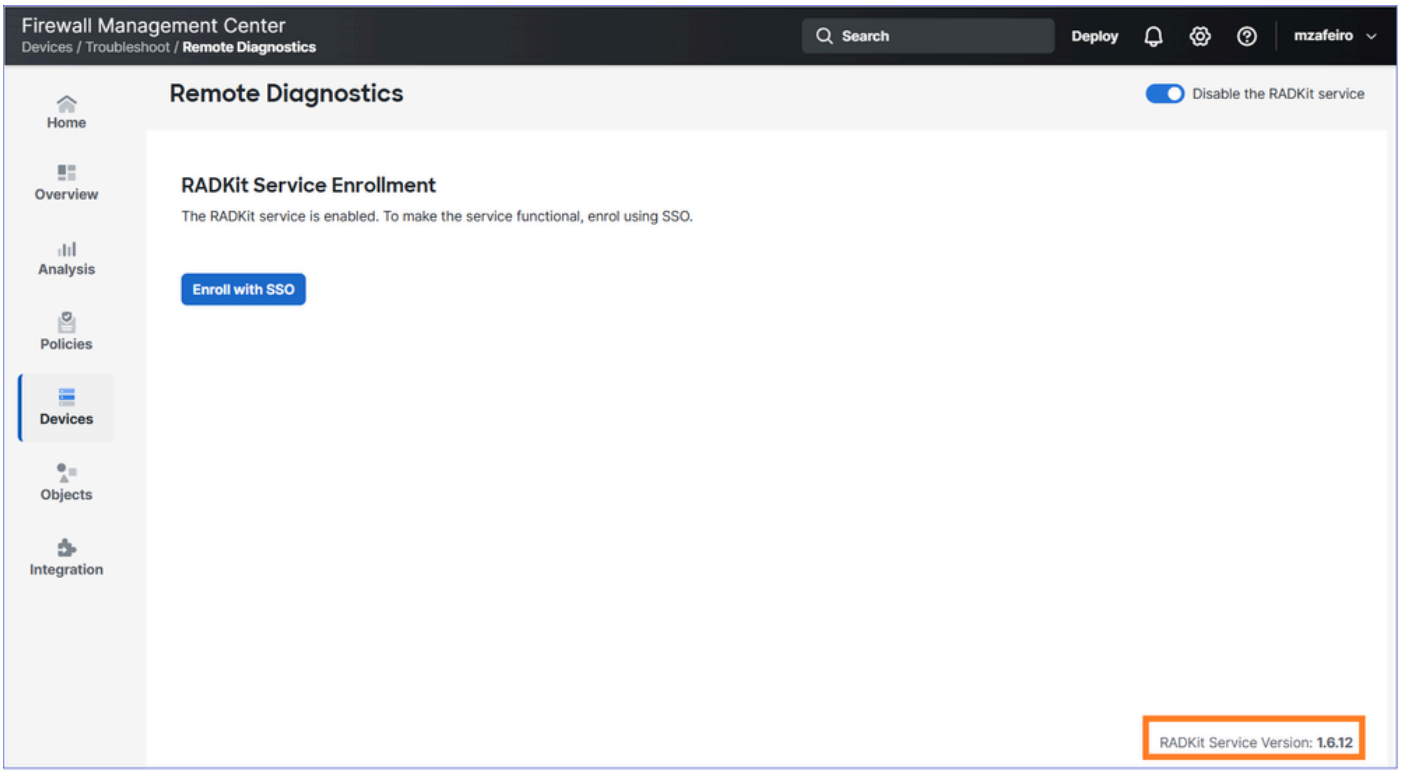
A. RADKitクラウドではCisco SSOのみが受け入れられます。会社のアカウントをシスコのアカウントに関連付けるオプションがあり、RADKitサービスをシスコ以外の電子メールで登録できます。

FAQ:RADKitバージョン

Q. 7.7リリースのFMCには、RADkitのどのバージョンが含まれていますか。FMCに含まれているRADKitのバージョンを調べるにはどうすればいいですか。これは、FMCをアップグレードしなくてもアップデートできますか。

A.

- 7.7.0に付属するRADKitのバージョンは1.6.12です。
- RADKitサービスのバージョンは、FMC Remote Diagnosticsページの下部に「RADKit Service Version: 1.6.12」と表示されます。



- RAID:FMCアップグレードパッケージやホットフィックスにバンドルされています。FMCのRADKitサービスを個別にアップグレードすることはできません。

FAQ：その他

Q. FMCで管理されていない外部デバイスを含めることはできますか。

A. FMCで管理されているデバイスだけをRADKitインベントリに追加し、認証を通じてアクセスできます。

Q. FMCバックアップの一部としてRADKit設定はバックアップされますか。

A.

- 設定は、FMCバックアップの一部としてバックアップされません。
- 通常、無制限のアクセスは提供されず、通常は限られた期間しかアクセスできないと予測さ

れるため、バックアップされません。

参考資料

参考リンク：

- FMC設定ガイド – RADKit
- https://radkit.cisco.com/
- https://radkit.cisco.com/docs/index.html
- https://radkit.cisco.com/downloads/release/
- https://github.com/Cisco-RADKit/Intro

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。