

在FMC中配置Cisco RADKit集成

目录

[简介](#)

[背景](#)

[功能说明和演练](#)

[FMC REST API](#)

[思科支持：RADKit控制台](#)

[升级和向后兼容性](#)

[故障排除](#)

[诊断概述](#)

[RADKit会话日志](#)

[故障排除演练问题示例](#)

[遥测](#)

[常见问题](#)

简介

本文档介绍7.7版本中添加的Cisco RADKit Integration in FMC功能。

背景

防火墙管理员面临的问题

- 远程自动化开发套件(RADKit)由思科开发，是一个网络范围的协调器，旨在为用户提供以安全方式访问网络设备以及排除网络设备故障的可能性。<https://radkit.cisco.com/>
- 思科安全防火墙管理中心(FMC)管理和操作安全防火墙威胁防御(FTD)设备。单个FMC可以管理不同位置的多个设备。
- 虽然用户可以单独安装RADKit并自注册其FMC和FTD，但将RADKit服务构建到FMC中并以自动方式自注册FMC和所有受管设备(FTD)将给最终用户带来更好的体验。

使用案例

在FMC中集成RADKit后，用户可以从中获益的一些关键功能包括：

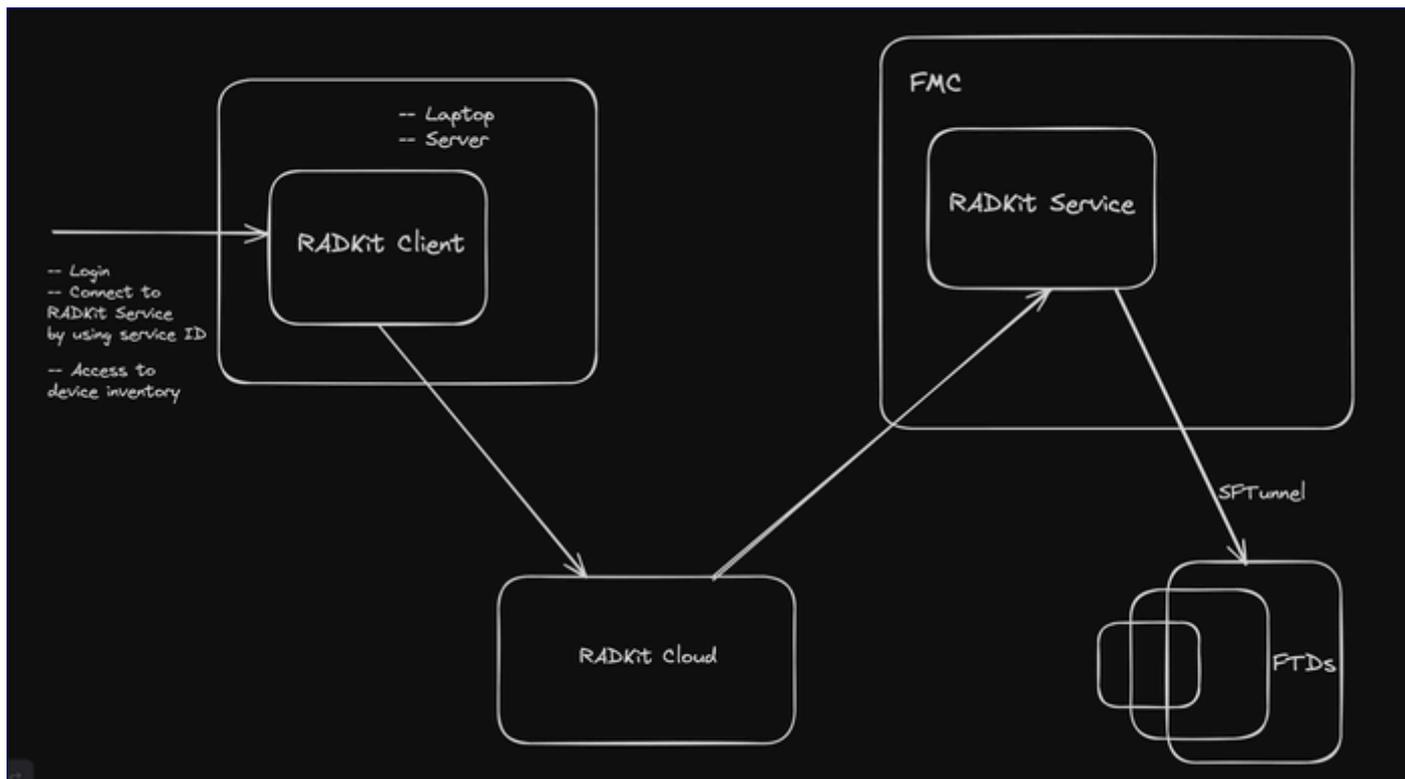
- 能够从RADKit客户端CLI远程访问FMC/FTD。
- 能够向任何有需要的人（例如，思科TAC工程师）提供对FMC/FTD的受控访问。
- 利用自动化功能从RADKit客户端收集数据和诊断问题（可以从RADKit客户端创建和使用在多个设备上执行命令的脚本）。

新功能 — 解决方案

- 从安全防火墙7.7.0开始，远程自动化开发套件(RADKit)服务集成到FMC中。
- 用户可以按需启用或禁用RADKit服务，在RADKit云中注册该服务，以及创建远程用户的授权，以在计划访问持续时间内从RADKit客户端访问特定设备。
 - 可以编辑或撤销授权。
- 还可以选择为设备提供sudo访问，以进行高级故障排除。

FMC图中的RADKit服务集成

下图显示了RADKit如何实现从用户 (TAC工程师) RADKit客户端到生产FTD设备的通信：



基础知识：支持的平台，许可

应用和管理器

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

支持的其他方面

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

要运行的功能的依赖关系

- 最低版本是安全防火墙7.7.0。
- 要连接到FMC中托管的RADKit服务，需要从<https://radkit.cisco.com/downloads/release/>将RADKit客户端安装在支持工程师的计算机上。
- RADKit客户端的首选版本为1.6.10或更高版本。
- 由于RADKit服务向后兼容较旧的RADKit客户端版本，因此可以使用较旧版本的RADKit客户端。

功能说明和演练

功能概述

- RADKit服务集成到FMC中使设备管理员能够向远程用户（思科TAC工程师）提供对其网络中特定FMC和FTD设备的访问权限，以便进行故障排除和自动化。RADKit在故障排除方面比屏幕共享高效得多，它不需要控制用户的计算机，是一种更安全的网络工作方式，并且很好地补充了Webex。
- 这样可改善技术支持体验，因为设备管理员无需单独安装和配置RADKit服务。此外，这还减少了Cisco TAC工程师解决支持问题的支持时间。

配置步骤:概述

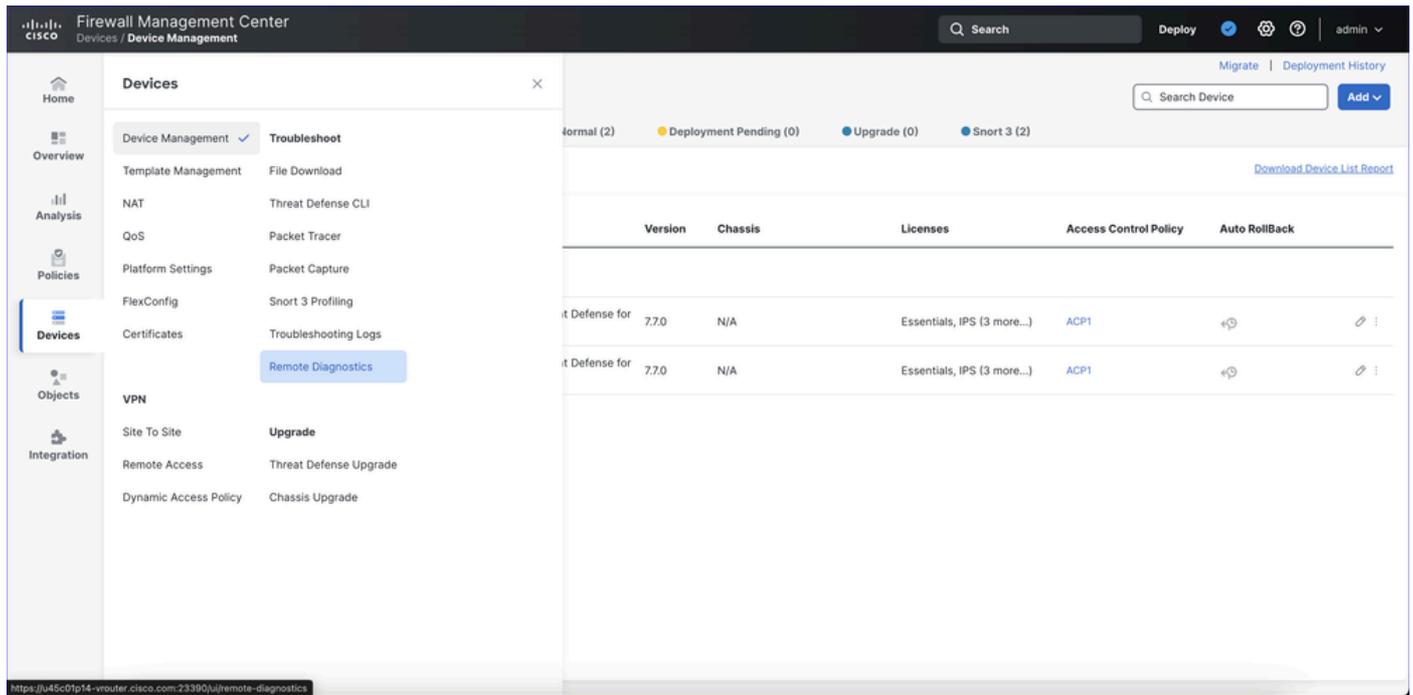
- 1.设备管理员（FMC管理员用户）：在FMC GUI上启用和注册RADKit服务并配置授权。
- 2.思科TAC/思科支持：在其计算机上安装RADKit客户端，从RADKit客户端访问设备并排除设备故

障。

FMC管理员用户：防火墙管理中心演练

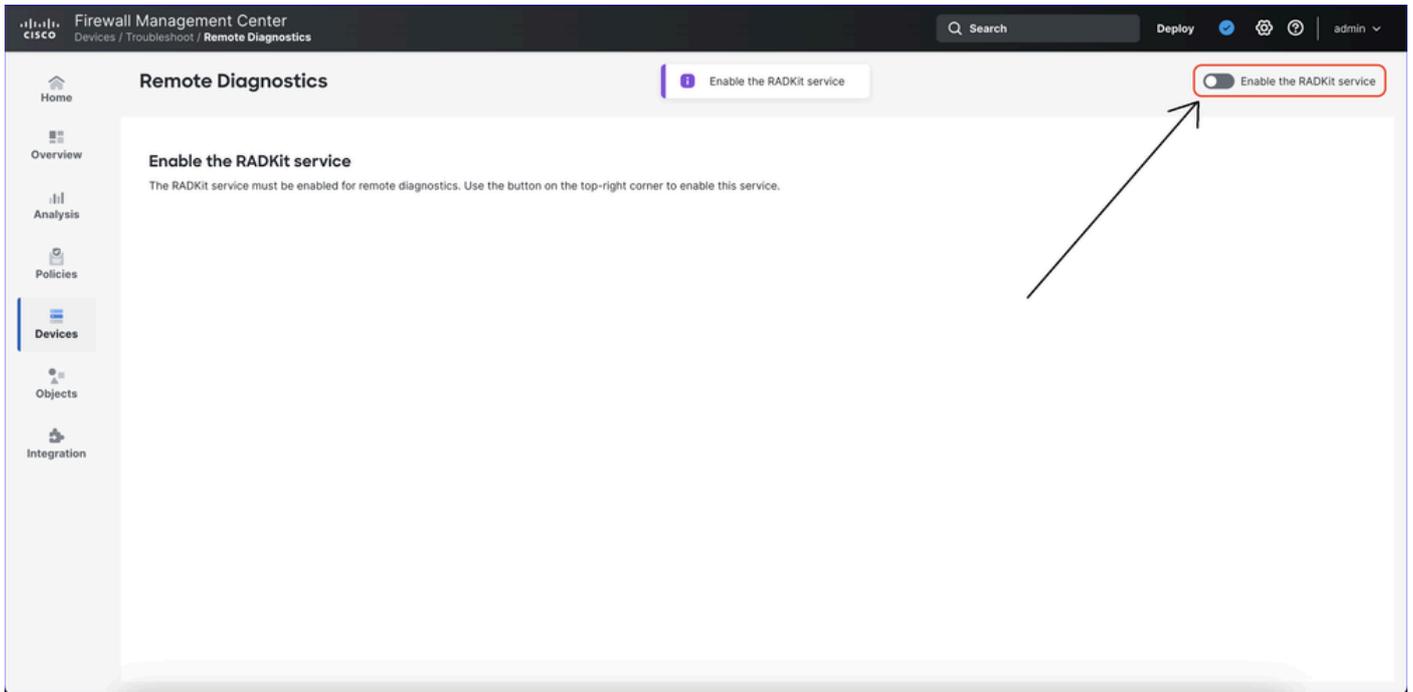
Remote Diagnostics菜单

- 在Devices -> Troubleshoot下，已为此功能添加了一个新的“Remote Diagnostics”菜单项。
- Administrator、Network Admin和Maintenance用户在页面上有读/写权限。
- 安全分析师、安全分析师（只读）和安全审批人用户对该页面具有只读权限。



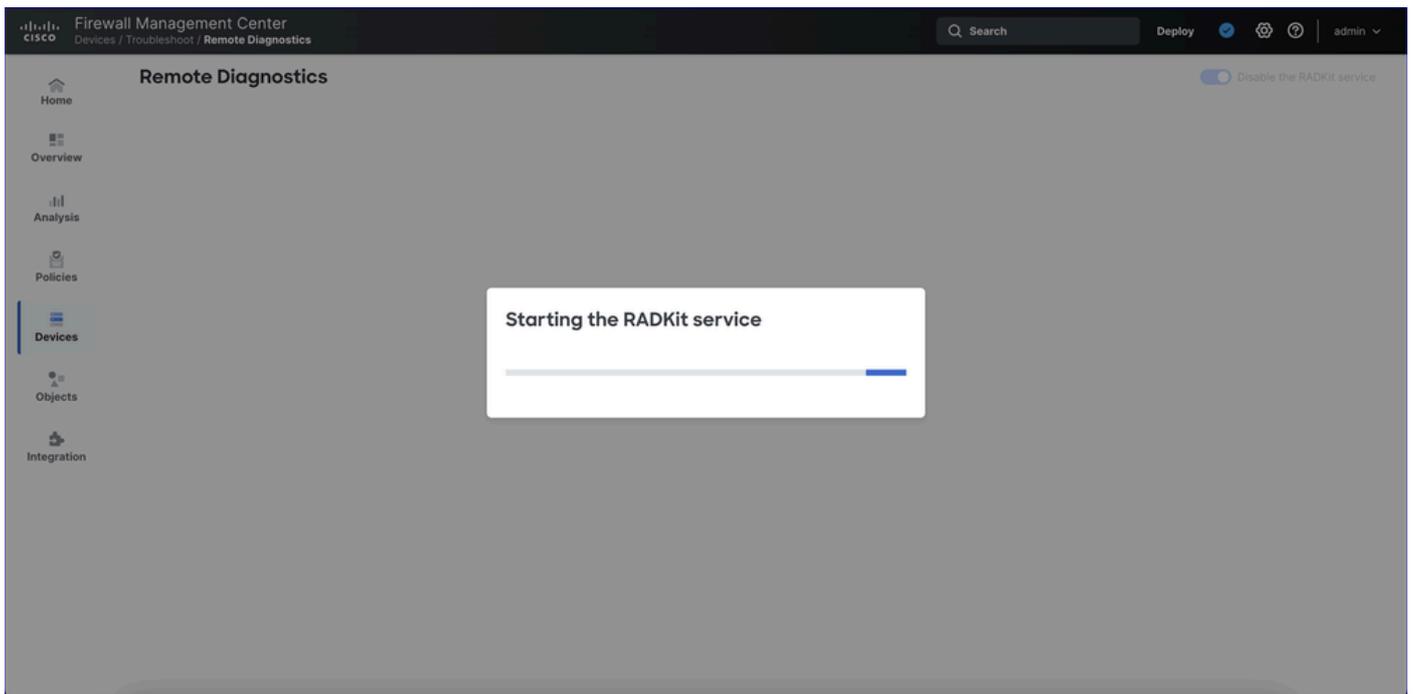
初始远程诊断页面

这是初始Remote Diagnostics页。可通过切换“启用RADKit服务”开关来启用RADKit服务：



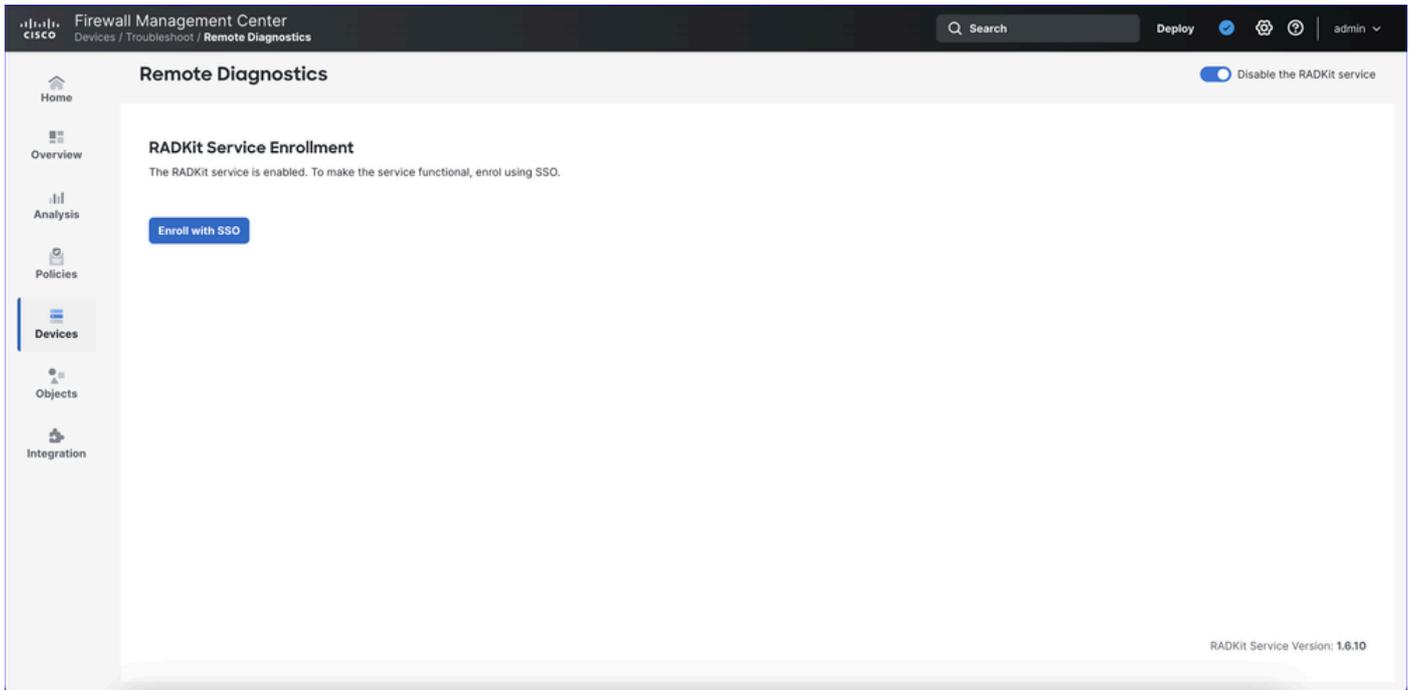
RADKit服务启动

启用RADKit服务后，在RADKit服务启动之前会显示进度条：



RADKit服务已启用

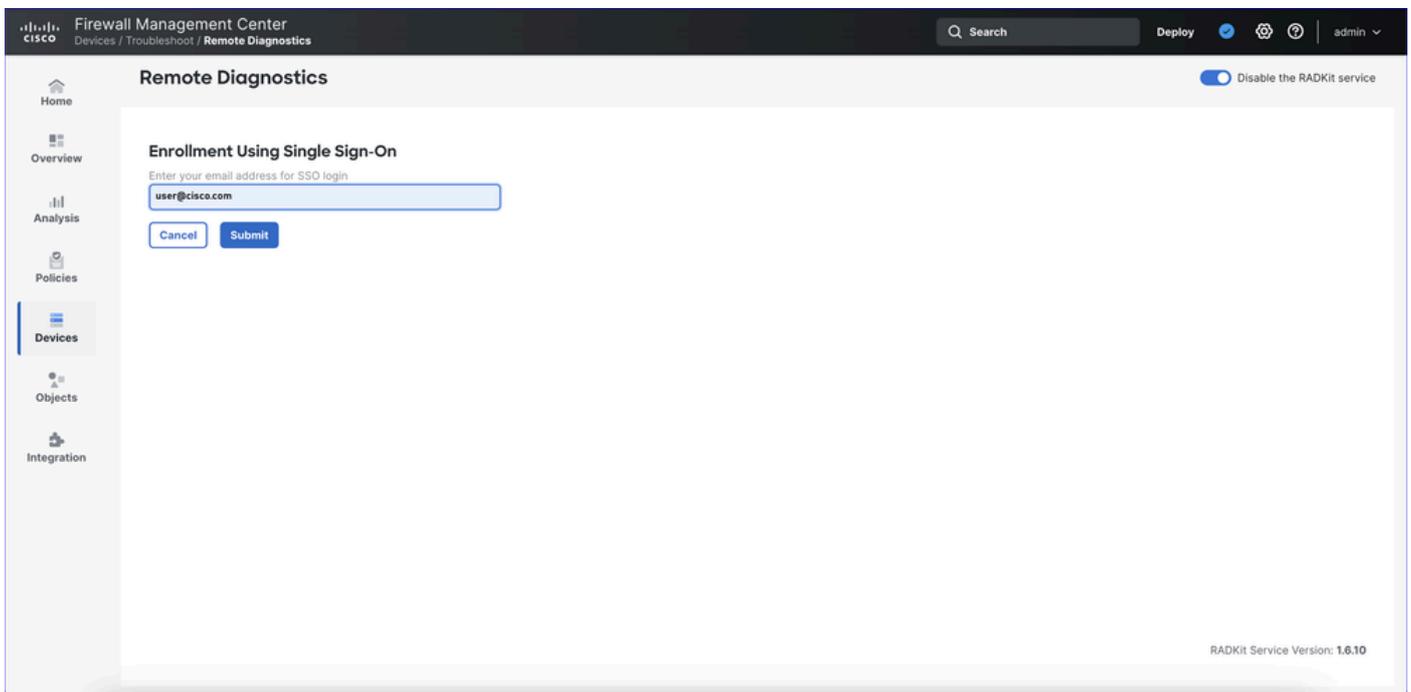
- 当RADKit服务启用过程完成时，显示此页面：



下一步是通过点击“使用SSO注册”按钮在RADKit云中注册。

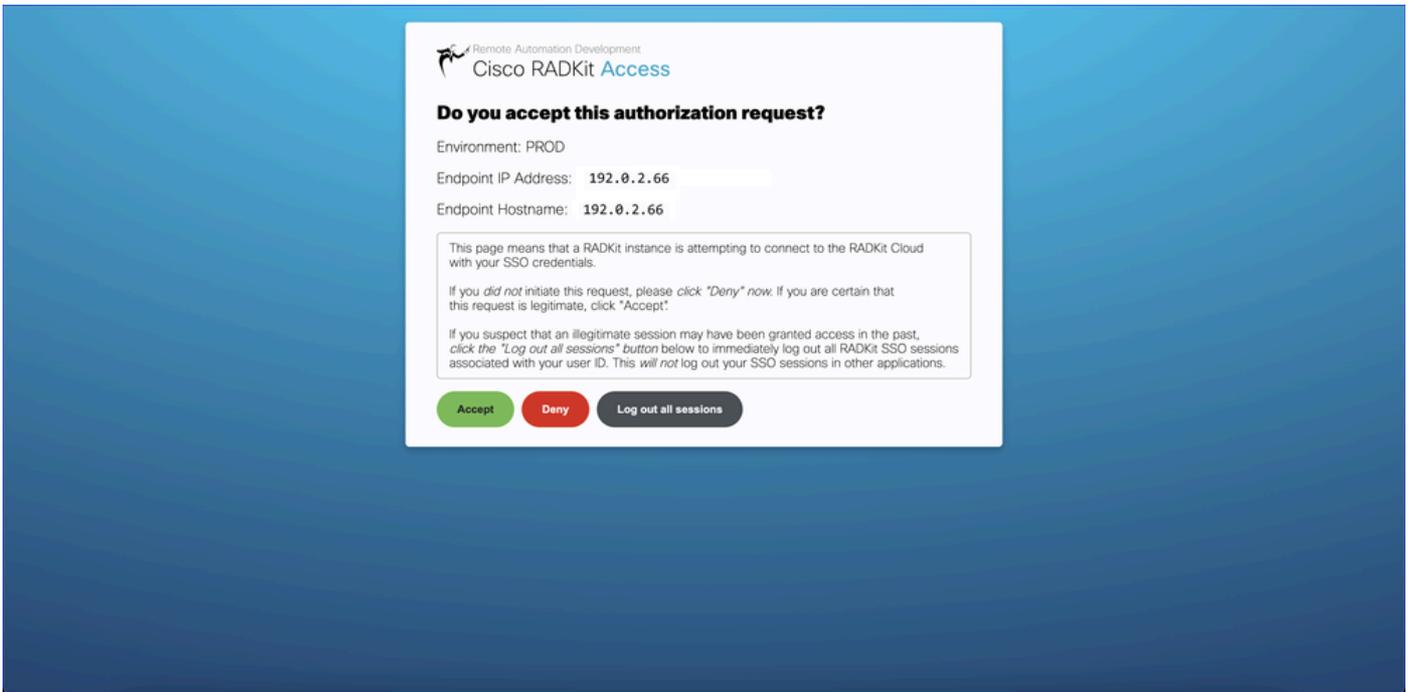
使用SSO注册 — 输入电子邮件地址

注册流程的第1步包括输入RADKit云注册的用户邮件地址：



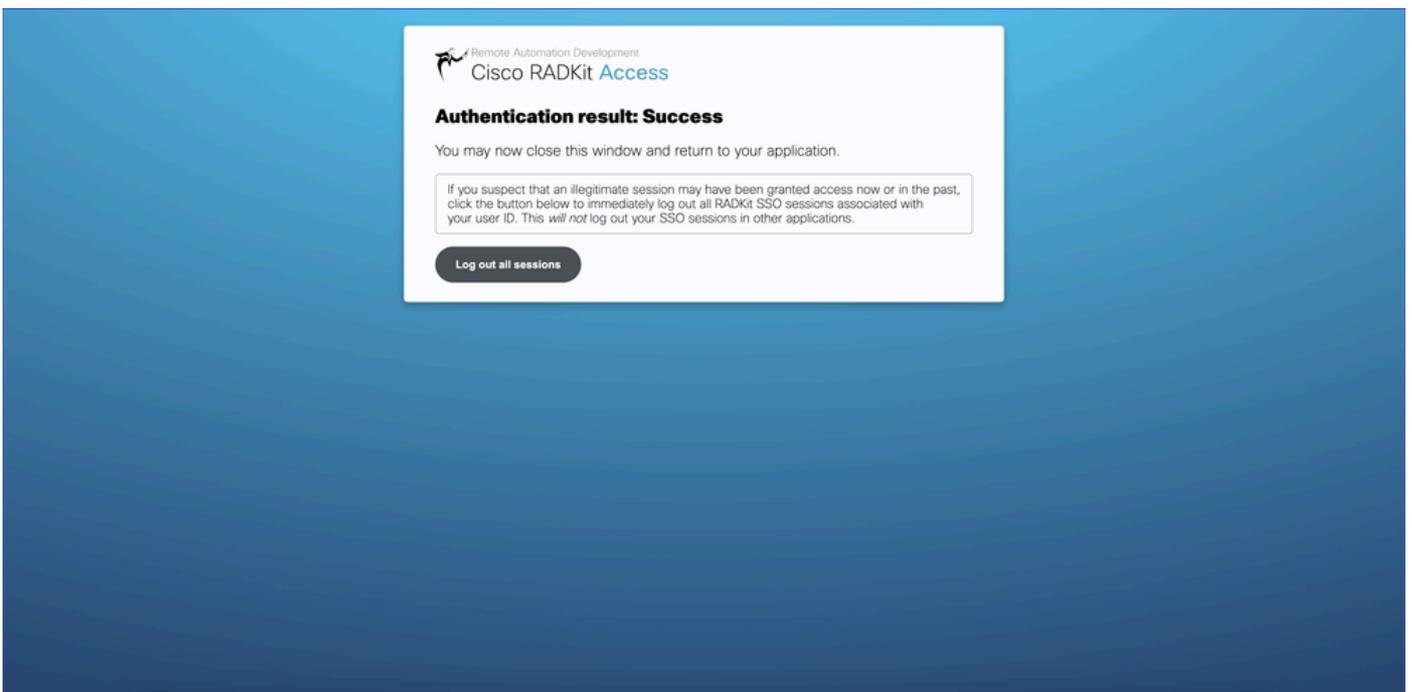
使用SSO注册 — 接受授权请求

新的浏览器选项卡（或窗口，取决于浏览器设置）打开。单击Accept按钮。



使用SSO注册 — 身份验证成功

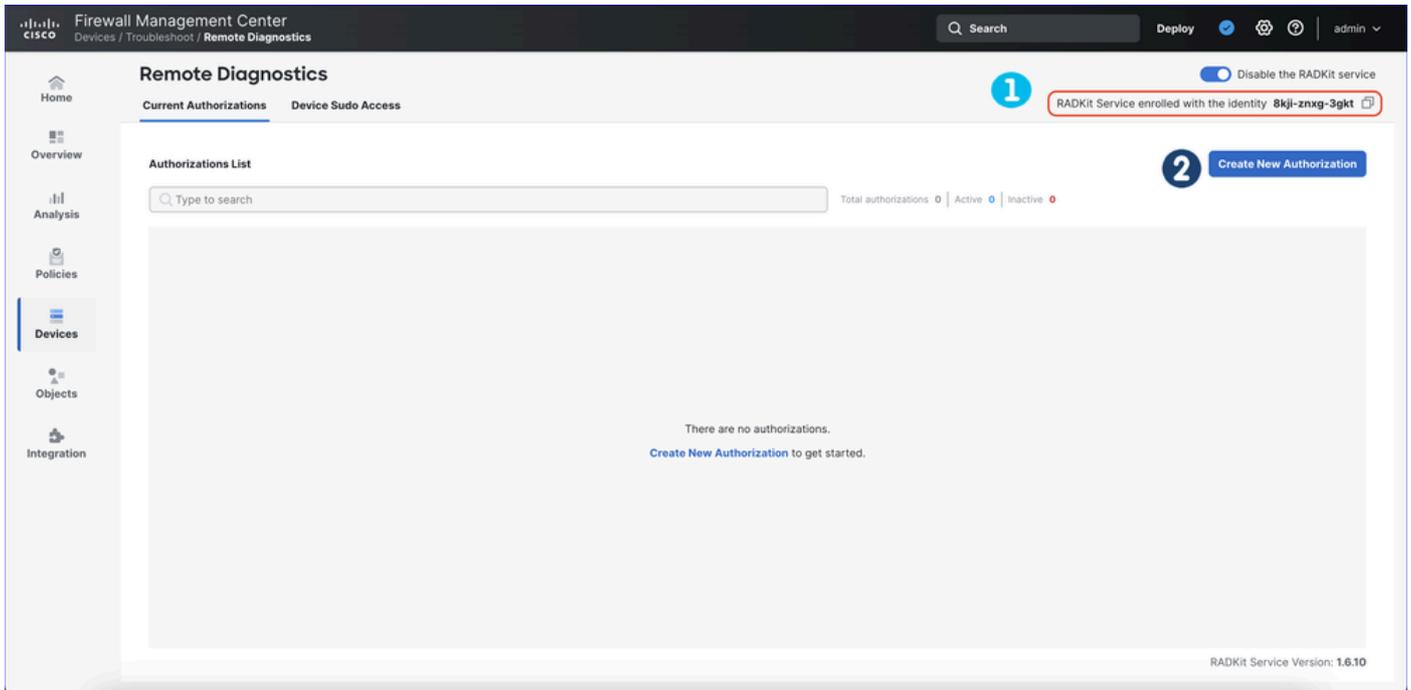
身份验证成功后，用户可关闭浏览器选项卡并返回FMC远程诊断页面。



RADKit服务已注册

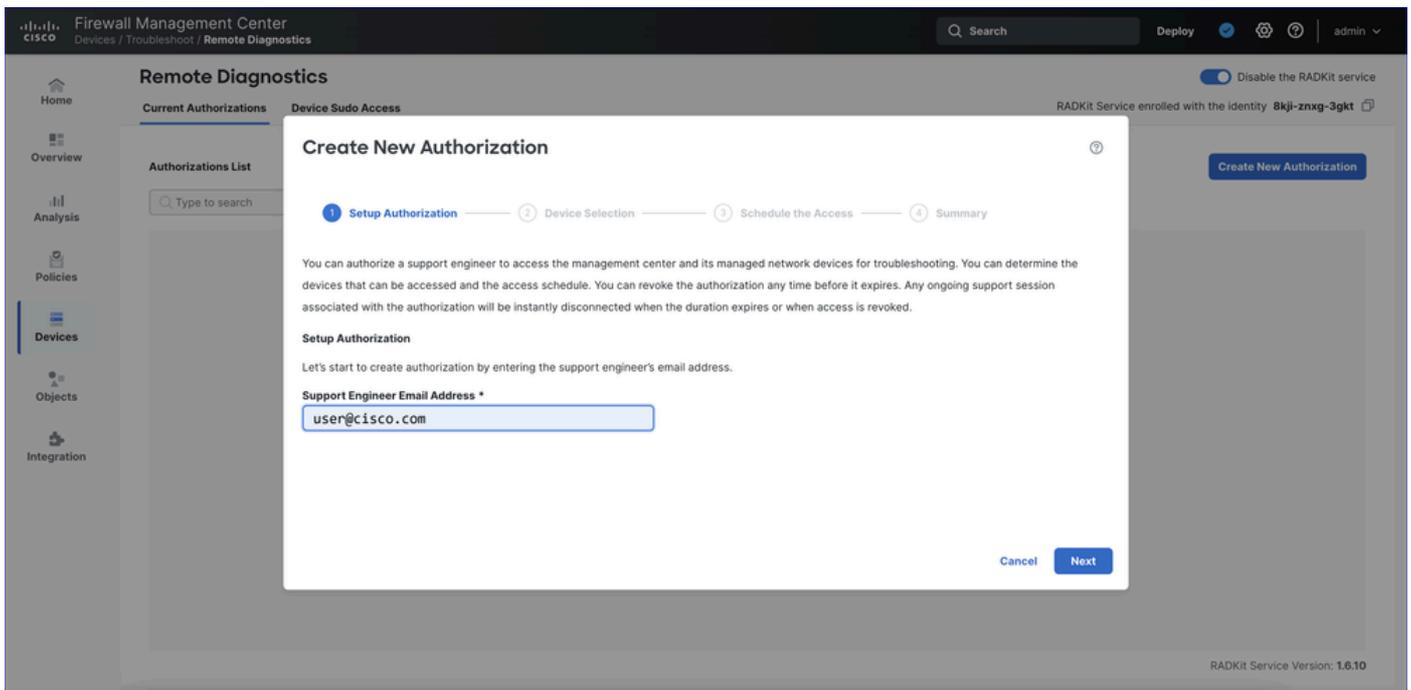
使用指定的服务ID注册RADKit服务（在本示例中，ID为8kji-znxg-3gkt）。可以将ID复制到剪贴板。将其交给思科TAC工程师，这样他们就可以从RADKit客户端连接到RADKit服务。

下一步是通过点击“Create New Authorization”(创建新授权)按钮创建授权：



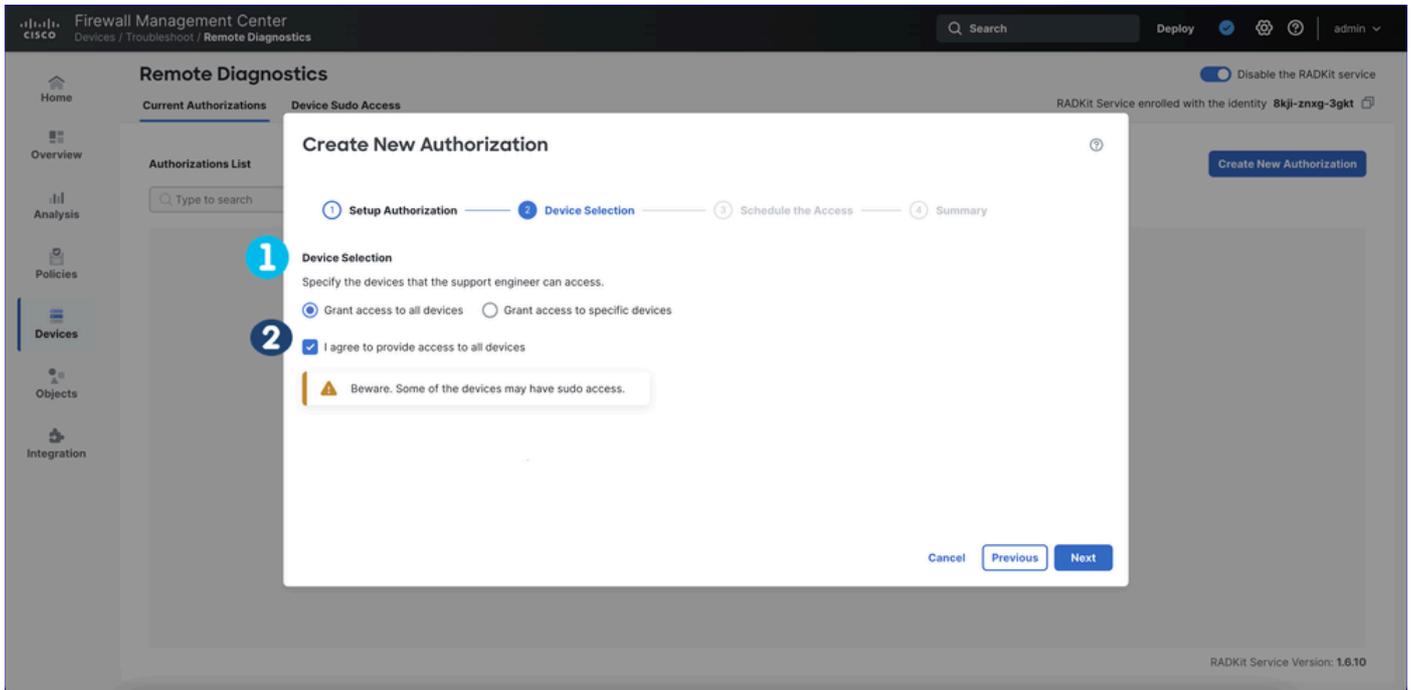
创建新授权：第 1 步

- 要创建新授权，第一步是添加支持工程师电子邮件地址。
- 创建新授权有四个步骤。步骤的进度显示在顶部。



创建新授权：第 2 步

- 步骤 1：指定支持工程师可以访问的设备。或者，如本示例所示，授予对所有设备的访问权限。
- 步骤 2：选中所有或特定设备的单选按钮。对于特定设备，可以选择FMC和/或FTD。注意 Device Sudo Access选项卡中某些设备可以获得sudo访问权限的警告。选中复选框之前，不会启用Next按钮。
- Sudo访问稍后在Device Sudo Access选项卡中按设备提供（在创建授权期间不提供）。

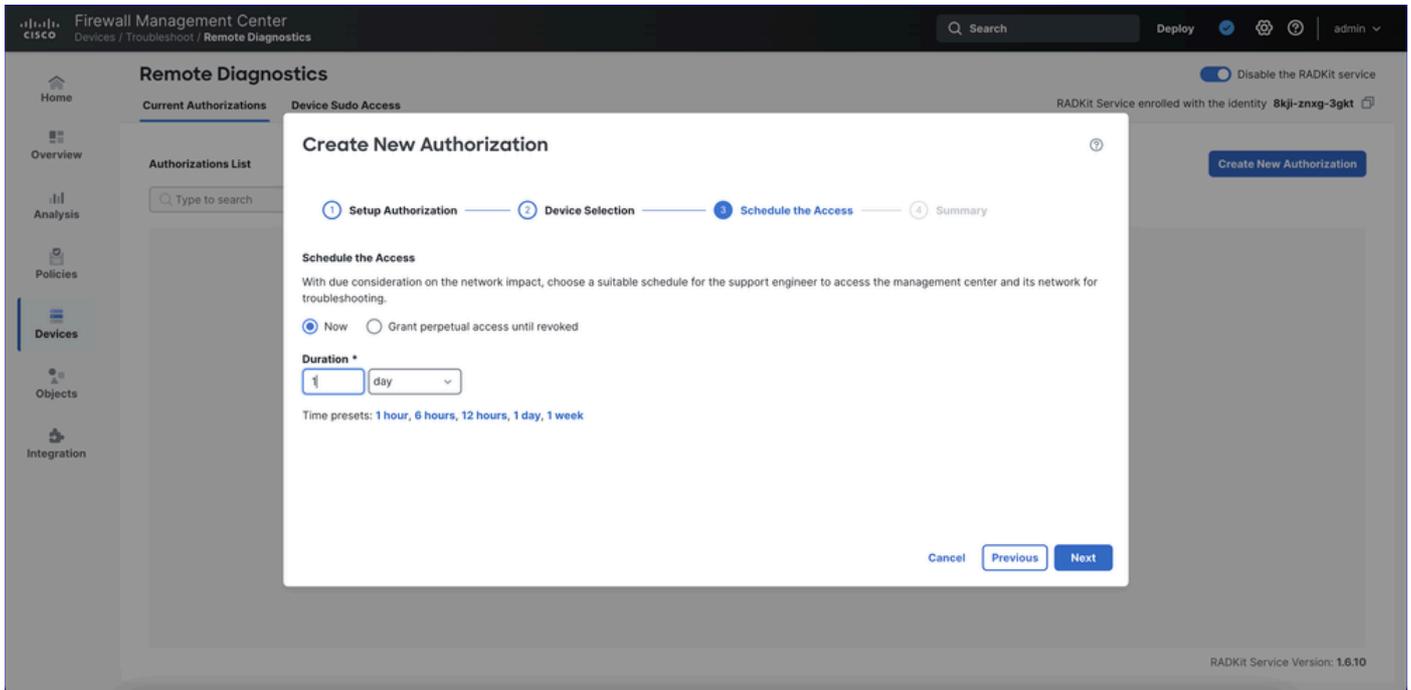


有关领料设备的备注

- 只能选择受支持版本上的设备（例如，在初始版本中，只有7.7.0设备）。
- 已禁用且不可访问的设备不可选择。RADKit依赖sftunnel(TCP 8305)访问设备。
 - 如果存在sftunnel连接问题，则它不起作用，但它仍然显示在RADKit资产中。
 - 如果设备已断电，则根本看不到该设备。
- 如果HA对中有FMC，则只能添加主用/主用。
- 创建/编辑授权时，设备会添加到RADKit资产。从FMC注销设备后，设备会从设备的“资产”中删除。

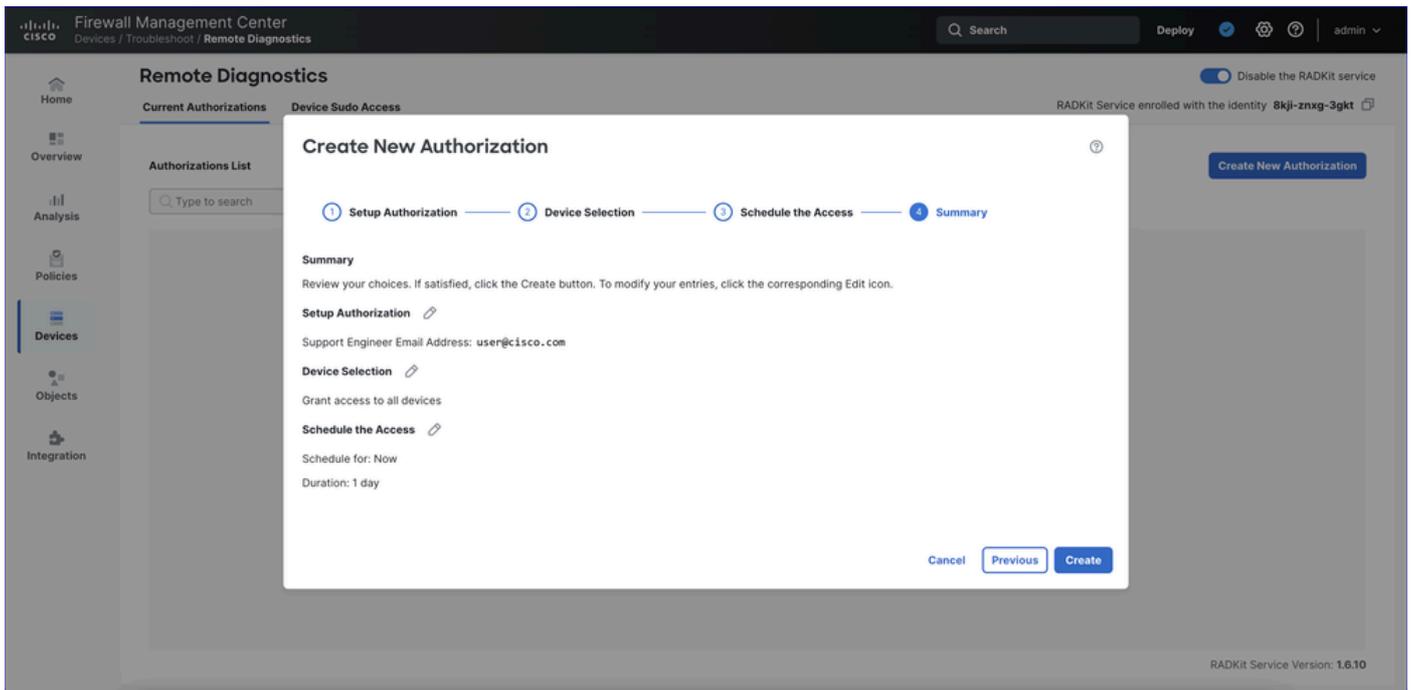
创建新授权：第 3 步

- 步骤 3：指定支持工程师访问设备的持续时间。
- 选择“Now”并指定持续时间，或者，
- 选择“Grant permanual access until revoked”。
- 默认持续时间为1天。可以设置任何持续时间；还有一些预定义的持续时间值。



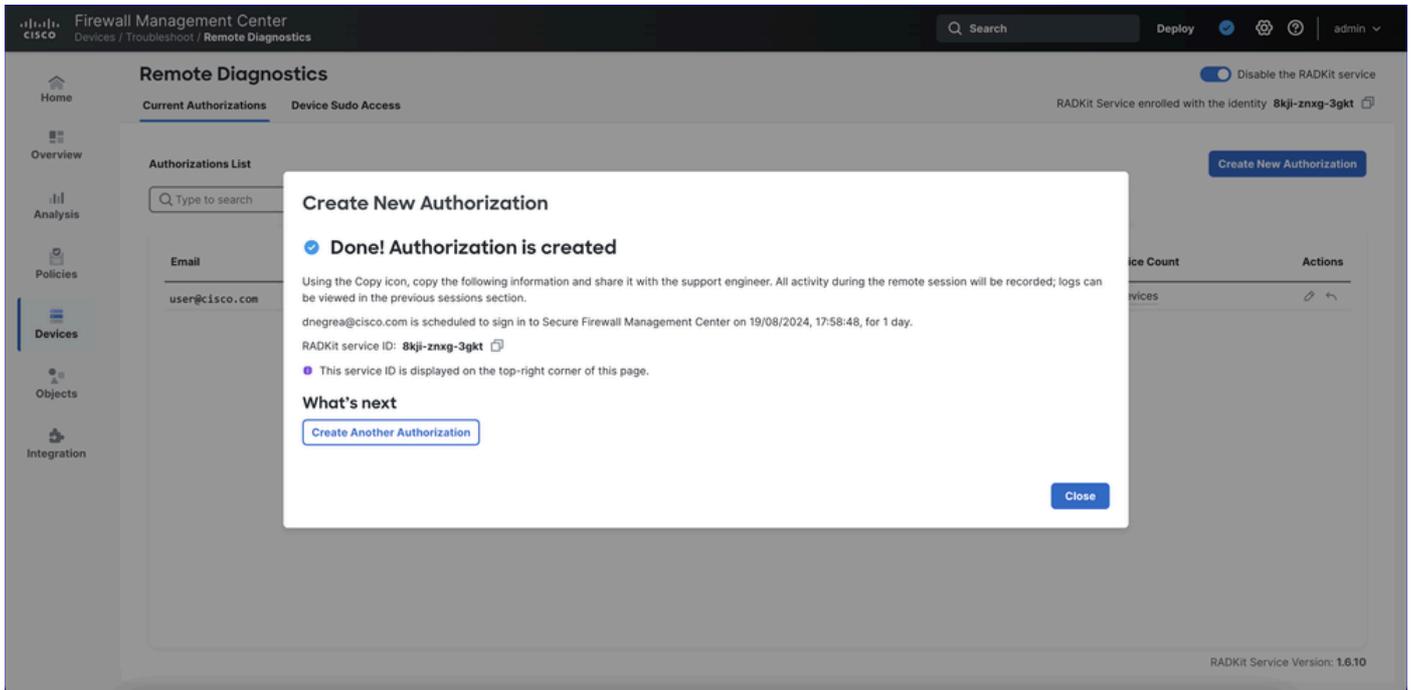
创建新授权摘要

最后一步是授权摘要。用户可以在此处查看和编辑配置。



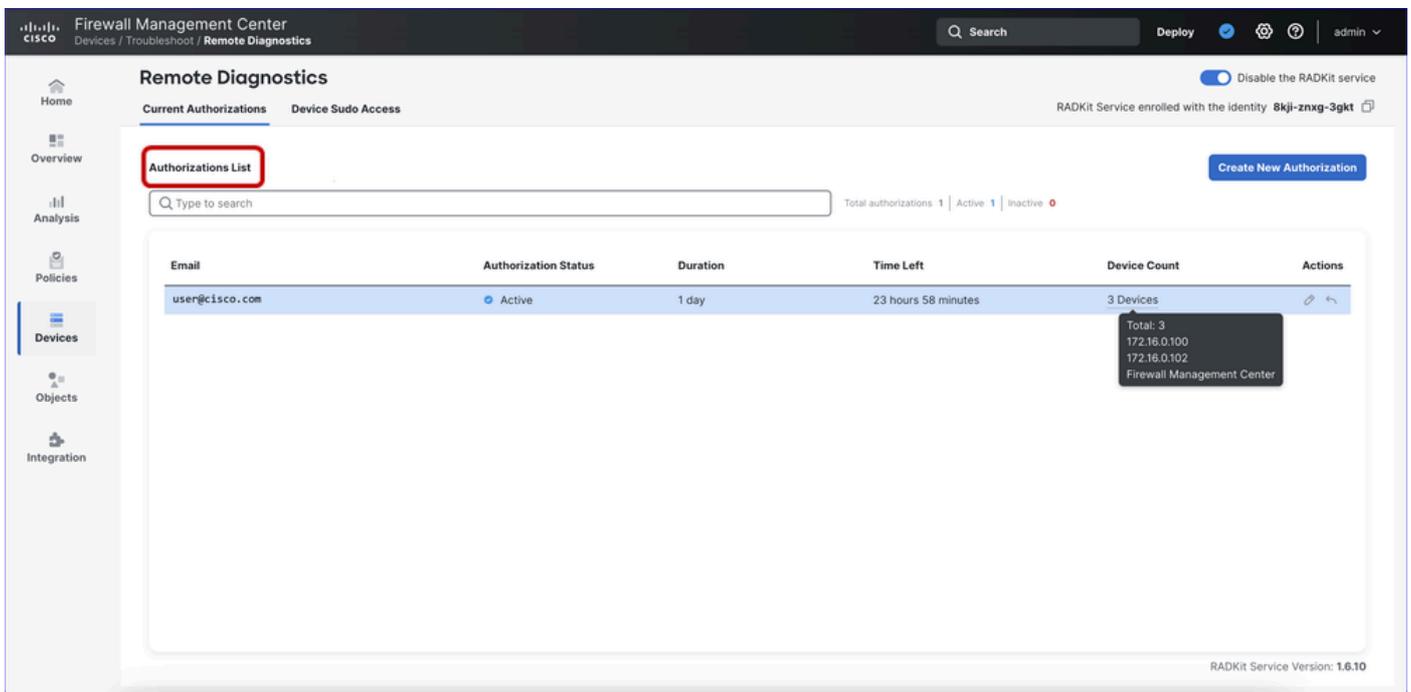
创建新授权已完成

授权创建完成后，将显示确认屏幕：



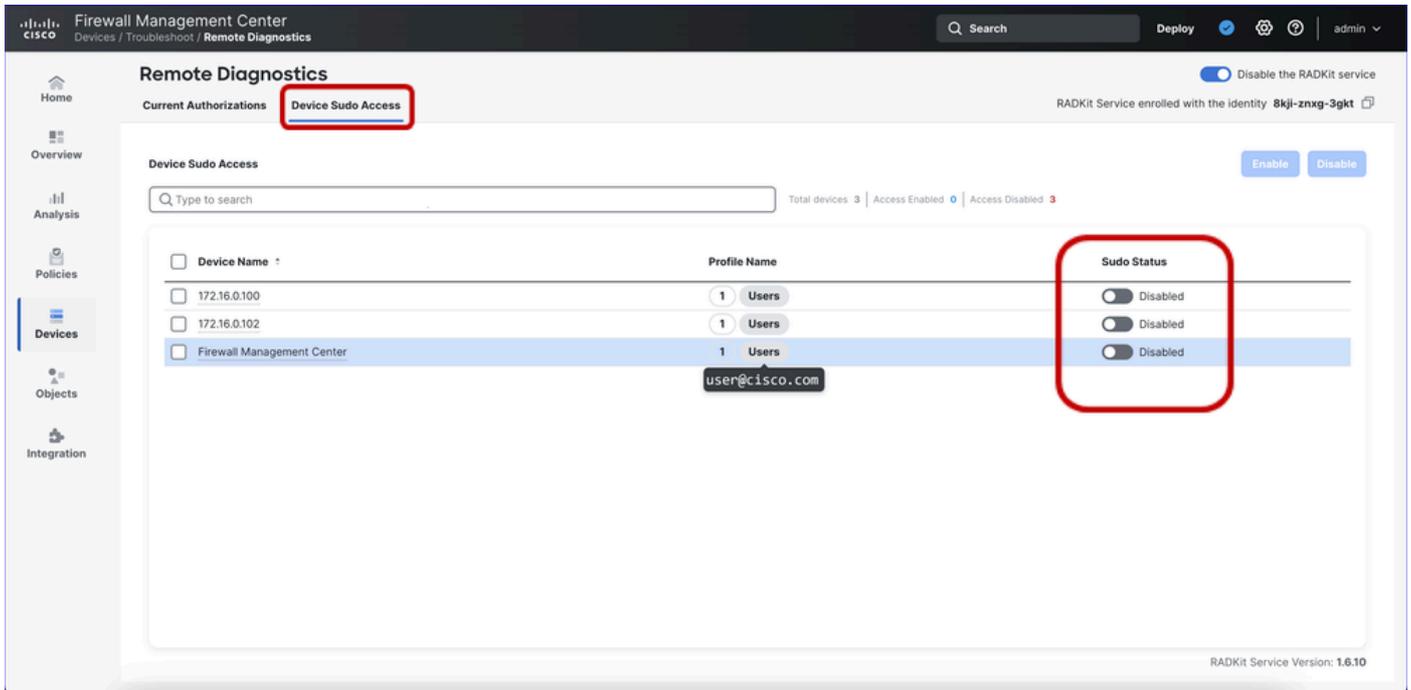
当前授权列表，包括撤销

- 当前授权列表显示在Current Authorizations选项卡中。
- “操作”(Actions) (最右侧的列) 为“撤销访问权限”(Revoke access)和“编辑授权”(Edit authorization)。



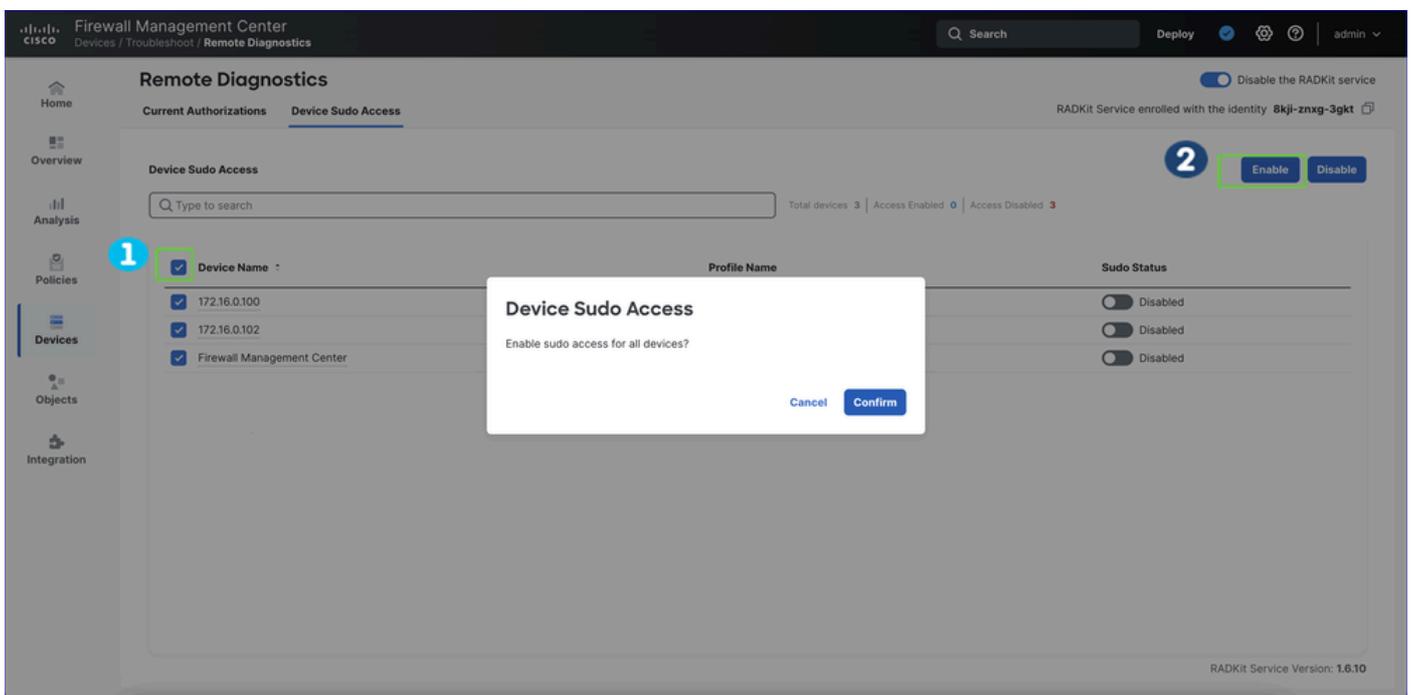
设备Sudo访问列表

- 具有sudo访问设置的设备列表显示在Device Sudo Access选项卡中。
- 使用右列中的切换按钮打开sudo访问。默认情况下关闭。
- 此外，还可以批量启用/禁用sudo访问。



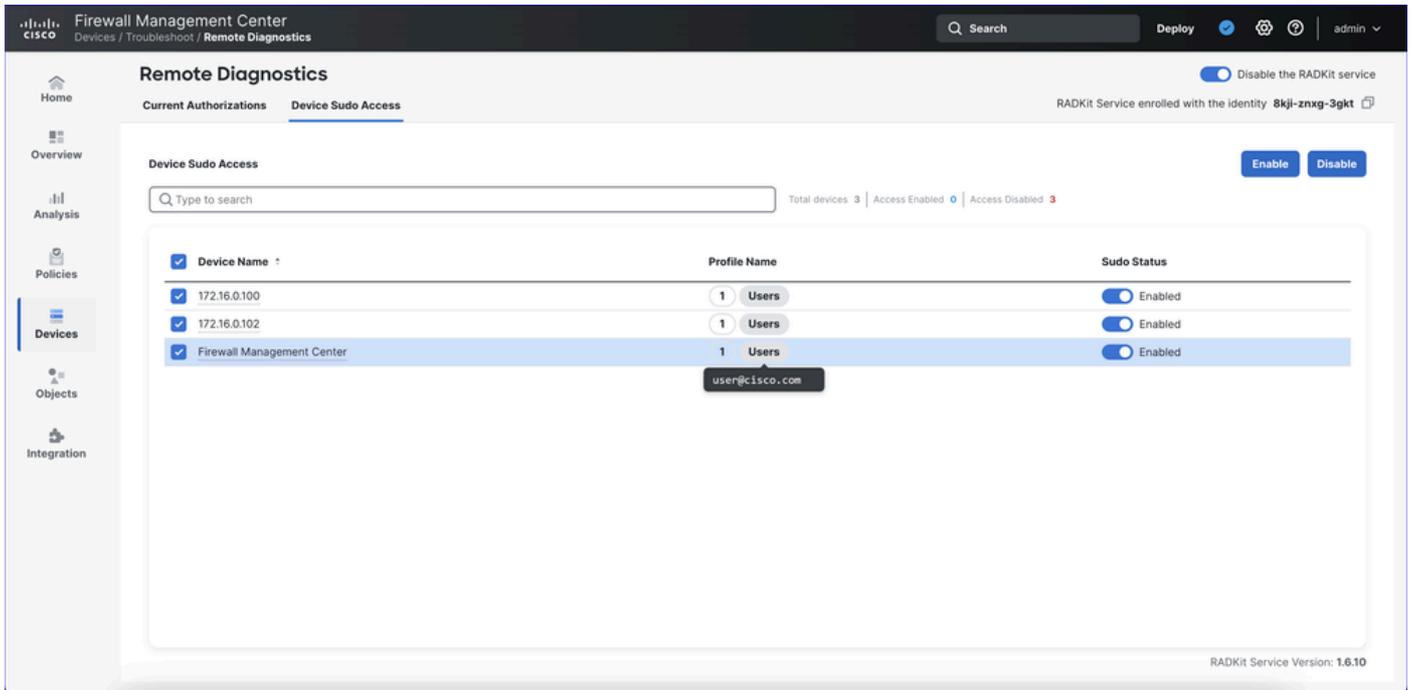
确认启用设备Sudo访问

- 1.通过选择设备，然后点击“启用”按钮，可以对所有设备或仅对某些特定设备启用Sudo访问权限。
- 2.启用时，系统将显示确认对话框，并且需要单击Confirm。



已启用设备Sudo访问

- 启用或禁用设备的sudo访问后，页面右侧的Sudo Status列会更新。
- 支持工程师能够在设备上运行sudo su;这是无密码的。支持工程师不需要具有根密码。



其他注释

- 只有FMC用户有权访问的域中的设备才可见，并且可以授权进行远程访问。
- 如果FMC在HA中：
 - RADKit服务只能在主用/主用上启用。
 - 辅助FMC当前无法添加为要从RADKit客户端访问的设备。
- 一次只能授权一名支持工程师。
 - 如果您需要其他支持工程师来访问，请为其他工程师创建另一个授权。服务ID将相同。

FMC REST API

RADKit服务REST API

为了支持RADKit服务的创建和读取操作，引入了以下新的URL:

- GET `:/api/fmc_troubleshoot/v1/domain/{domainUUID}/radkit/services`
 - 从FMC检索所有RADKit服务数据。
- GET `:/api/fmc_troubleshoot/v1/domain/{domainUUID}/radkit/services/{id}`
 - 从指定的ID获取/检索RADKit服务数据。
- POST `:/api/fmc_troubleshoot/v1/domain/{domainUUID}/radkit/services`
 - 在FMC上创建RADKit服务（启用/禁用该服务）。

RADKit服务模式

RADKit服务模型包括：

- 类型

- ID
- 状态
- isEnrolled
- serviceId
- version

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

思科支持：RADKit客户端使用情况

支持方：安装RADKit客户端

- 要访问FMC/FTD，支持人员需要安装RADKit客户端。
 - 客户端在Windows、Mac和Linux操作系统上运行。
- 支持人员可以从多个用户访问多个设备。每个RADKit授权都有自己的设备“清单”。
 - 对于支持部门想要访问的每个用户设备资产，需要RADKit服务ID。
 - 对于单个资产，可以从RADKit客户端访问FMC及其托管FTD，这是用户在授权访问时指定的。

获取并安装RADKit客户端

RADKit客户端可以从<https://radkit.cisco.com/downloads/release/>本地安装，然后使用命令从终端启动：radkit-client

安装程序可用于Windows、MacOS和Linux。

```
15:07:59.886Z INFO | internal | CXD object created without authentication set, call `<this object>.authenticate()` to set authentication.

Example usage:
client = sso_login("<email_address>") # Open new client and authenticate with SSO
client = certificate_login("<email_address>") # OR authenticate with a certificate
client = access_token_login("<access_token>") # OR authenticate with an SSO Access Token
service = client.service("<serial>") # Then connect to a RADKit Service
service = start_integrated_service() # Immediately login to an integrated session
service = direct_login() # Establish cloud-less direct connection to service.
client.grant_service_otp() # Enroll a new service

>>> 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("8kji-znxg-3gkt")
15:09:03.406Z INFO | internal | Connecting to forwarder [forwarder_base_url='wss://prod.radkit-cloud.cisco.com/forwarder-4/' uri='wss://prod.radkit-cloud.cisco.com/forwarder-4/websocket/']
15:09:03.639Z INFO | internal | Connection to forwarder successful [forwarder_base_url='wss://prod.radkit-cloud.cisco.com/forwarder-4/' uri='wss://prod.radkit-cloud.cisco.com/forwarder-4/websocket/']
15:09:03.727Z INFO | internal | Forwarder client created. [forwarder_base_url='wss://prod.radkit-cloud.cisco.com/forwarder-4/' uri='wss://prod.radkit-cloud.cisco.com/forwarder-4/websocket/']
15:09:04.003Z INFO | internal | Connecting to forwarder [forwarder_base_url='wss://prod.radkit-cloud.cisco.com/forwarder-1/' uri='wss://prod.radkit-cloud.cisco.com/forwarder-1/websocket/']
15:09:04.244Z INFO | internal | Connection to forwarder successful [forwarder_base_url='wss://prod.radkit-cloud.cisco.com/forwarder-1/' uri='wss://prod.radkit-cloud.cisco.com/forwarder-1/websocket/']
15:09:04.332Z INFO | internal | Forwarder client created. [forwarder_base_url='wss://prod.radkit-cloud.cisco.com/forwarder-1/' uri='wss://prod.radkit-cloud.cisco.com/forwarder-1/websocket/']
>>>
>>> service_inventory
<radkit_client.sync.device.DeviceDict object at 0x1154969a0>
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
firepower-1724078669 127.0.0.1 FMC True False False False False firepower False
Untouched inventory from service 8kji-znxg-3gkt.
>>> |
```

RADKit客户端屏幕截图与登录命令 (下一部分中的详细信息)。

RADKit客户端登录命令

- 使用用户在FMC中的授权期间输入的邮件地址。
- RADKit客户端登录并连接到指定的服务ID命令。在本例中，RADKit服务ID 8abc-znxg-3abc必须与防火墙管理员在FMC中看到的内容匹配。

<#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.radkit-cloud.cisco.com/forwarder-4/' uri='wss://prod.radkit-cloud.cisco.com/forwarder-4/websocket/']
15:09:03.727Z INFO | internal | Forwarder client created. [forwarder_base_url='wss://prod.radkit-cloud.cisco.com/forwarder-4/' uri='wss://prod.radkit-cloud.cisco.com/forwarder-4/websocket/']
15:09:04.244Z INFO | internal | Connection to forwarder successful [forwarder_base_url='wss://prod.radkit-cloud.cisco.com/forwarder-1/' uri='wss://prod.radkit-cloud.cisco.com/forwarder-1/websocket/']
```

15:09:04.332Z INFO | internal | Forwarder client created. [forwarder_base_url='wss://prod.radkit-cloud

RADKit客户端服务清单命令

用于列出远程用户 (思科TAC工程师) 有权访问的资产的命令 :

```
<#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.

资产中设备有一个filter命令 (下一部分) 。 使用左列中的名称启动与设备的交互式会话 (命令在即将出现的部分) 。

RADKit客户端 : 过滤设备

用于过滤资产中的设备的命令 :

```
<#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	False	172.16.0.100	False
172-16-0-102-1724078669	127.0.0.2	FTD	True	False	False	False	False	False	172.16.0.102	False

2 device(s) from service 8kji-znxg-3gkt.

RADKit客户端设备交互式会话命令

使用从之前的“service.inventory”命令获取的名称“firepower-1724078669”启动设备（在本例中为 FMC）的交互式会话：

```
<#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/client_
```

```
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 password
```

```
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客户端在设备上执行命令

在设备上执行命令！

```
<#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  
-----
```

从设备获取文件

- 通过RADKit客户端，Cisco TAC工程师可以通过SSH连接到设备并执行各种操作，包括生成故障排除文件。

思科支持：RADKit控制台

使用RADKit网络控制台

- 此外，Cisco TAC支持工程师可以使用RADKit网络控制台，也可以使用RADKit客户端。网络控制台是RADKit客户端的一部分。
- RADKit Network Console的功能为基本RADKit Client功能提供了简单的命令行界面(CLI)。它旨在用于快速连接到RADKit服务实例，建立交互式会话并下载/上传文件，无需任何麻烦和最少培训。
- 使用命令行启动Network Console:radkit-network-console
- 有关详细信息，请查看RADKit文档。

升级和向后兼容性

从7.7升级到7.7

- RADKit Service已添加到安全防火墙7.7.0中。
 - 升级到版本7.7.0+的设备具有RADKit服务所需的配置。

使用不受支持的FTD的经验

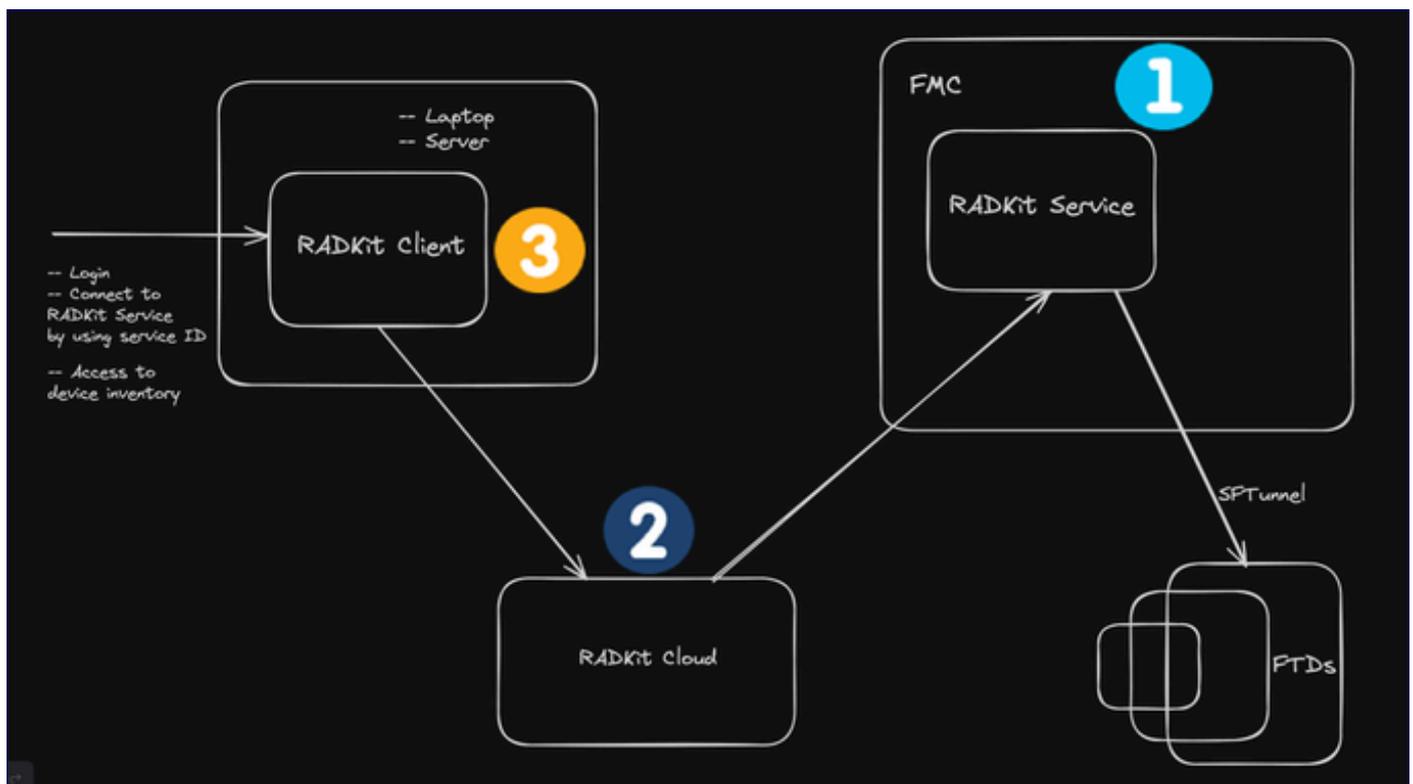
- FMC和FTD必须具有最低版本7.7.0才能使用此功能（版本低于7.7的FTD不能添加到7.7 FMC RADKit授权中）。
- 7.7.0以外的已注册FTD不可用于选择以启用授权。

故障排除

诊断概述

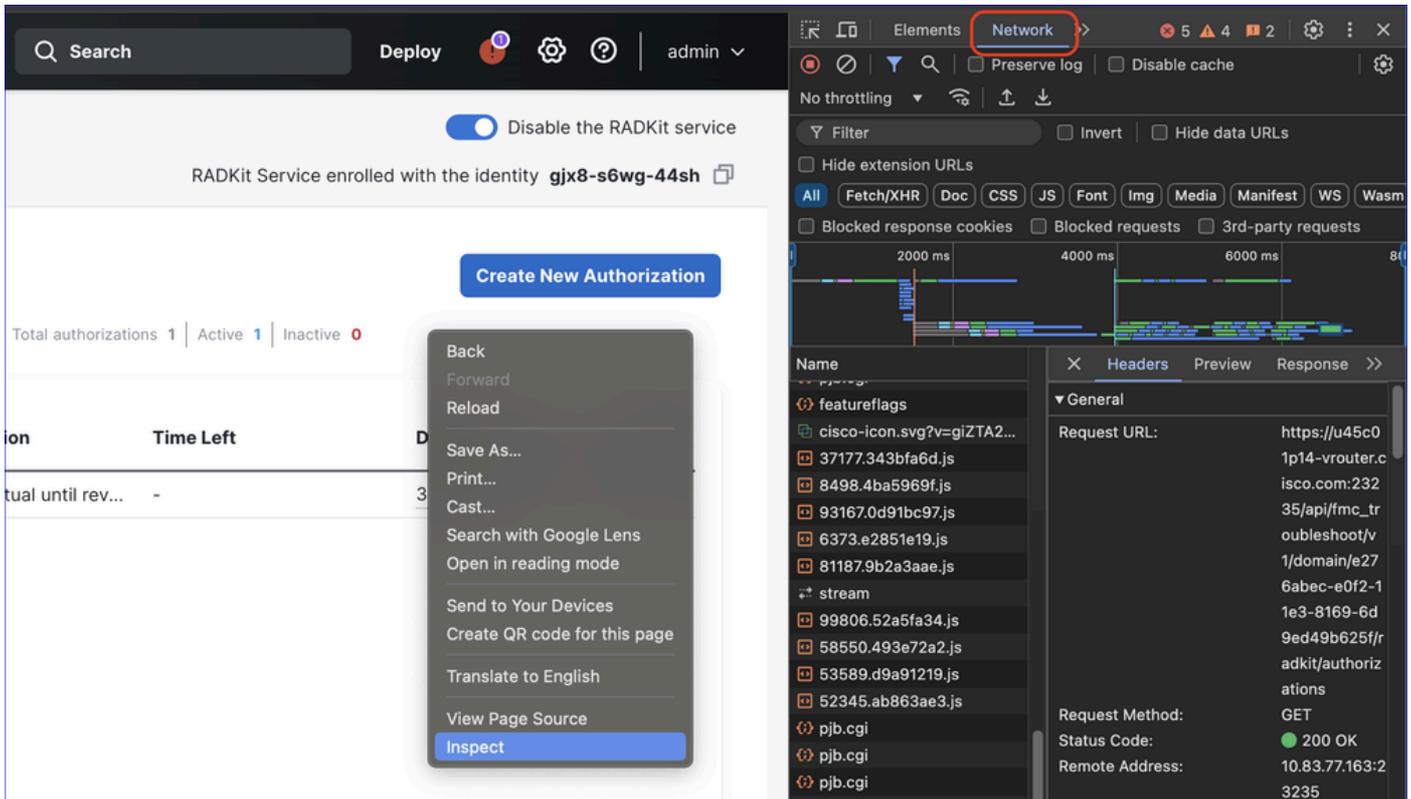
故障排除点

- 1.使用浏览器开发工具和FMC日志查看FMC中发生的情况。
- 2.对于FMC上的RADKit服务、RADKit云和RADKit客户端之间的通信问题，请查看RADKit客户端日志记录。
3. RADKit客户端。



如何排除故障：浏览器的开发工具

- 浏览器的Developer Tools、Network选项卡显示页面上运行的API调用，可用于排除FMC上的问题。可通过右键单击页面然后点击Inspect (检查) 来启动此功能。
- 在Network选项卡中检查API呼叫状态代码和响应预览。



RADKit Service Go中间件API

Go Middleware for the RADKit integration使用未通过FMC API资源管理器公开提供的API调用。Go Middleware API日志位于/var/log/auth-daemon.log。Go中间件执行的功能包括：

- 使用单点登录流程在RADKit云中注册RADKit服务。
- 获取所有远程RADKit用户的授权和相关设备的列表。
- 使用邮件获取特定远程RADKit用户授权和相关设备。
- 创建远程RADKit用户授权，并在指定的时间范围内授予访问设备（所有设备或选定设备列表）的权限。
- 修改远程RADKit用户授权。
- 删除远程RADKit用户授权。

RADKit服务故障排除日志

- 常规FMC日志：从FMC ssh会话发出pigtail命令。
- 访问中间件API:/var/log/auth-daemon.log
- 包含RADKit和auth-daemon处理数据的日志：

/var/log/process_stdout.log

/var/log/process_stderr.log

所有这些日志都包含在FMC/FTD故障排除中。

- 内部RADKit服务日志：/var/lib/radkit/logs/service/
- 从设备（FMC和FTD）上的RADKit客户端执行的操作的日志：
/var/lib/radkit/session_logs/service

要提交至思科TAC的日志

- 错误屏幕截图。
- 问题的描述。
- 复制步骤。
- 包含错误的Pigtail和/var/log/auth-daemon.log日志提取。

监控访问

在FMC审核日志中记录谁被授予了访问权限多长时间以及谁被授予了访问权限。

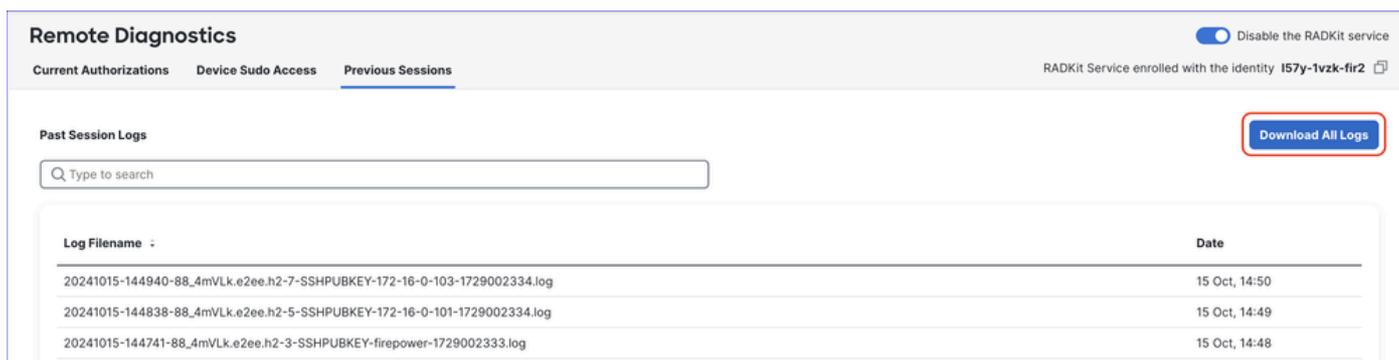
RADKit会话日志

从RADKit客户端在设备（FMC和FTD）上执行的操作的RADKit会话日志显示在FMC的/var/lib/radkit/session_logs/service:

- 日志来自RADKit服务本身。
- 这些日志包含在故障排除捆绑包中。
- 也可以从UI访问日志（请参见下一节）。
- 有多个会话日志文件；每个会话一个。

RADKit以前会话日志

通过点击“下载所有日志”(Download All Logs)按钮，可以下载从RADKit客户端执行的设备操作的RADKit会话日志，作为包含“以前的会话”(Previous Sessions)选项卡中所有日志的存档。



Remote Diagnostics

Current Authorizations Device Sudo Access Previous Sessions

Disable the RADKit service

RADKit Service enrolled with the identity i57y-1vzk-fir2

Past Session Logs

Download All Logs

Q Type to search

Log Filename	Date
20241015-144940-88_4mVlk.e2ee.h2-7-SSHPUBKEY-172-16-0-103-1729002334.log	15 Oct, 14:50
20241015-144838-88_4mVlk.e2ee.h2-5-SSHPUBKEY-172-16-0-101-1729002334.log	15 Oct, 14:49
20241015-144741-88_4mVlk.e2ee.h2-3-SSHPUBKEY-firepower-1729002333.log	15 Oct, 14:48

故障排除演练问题示例

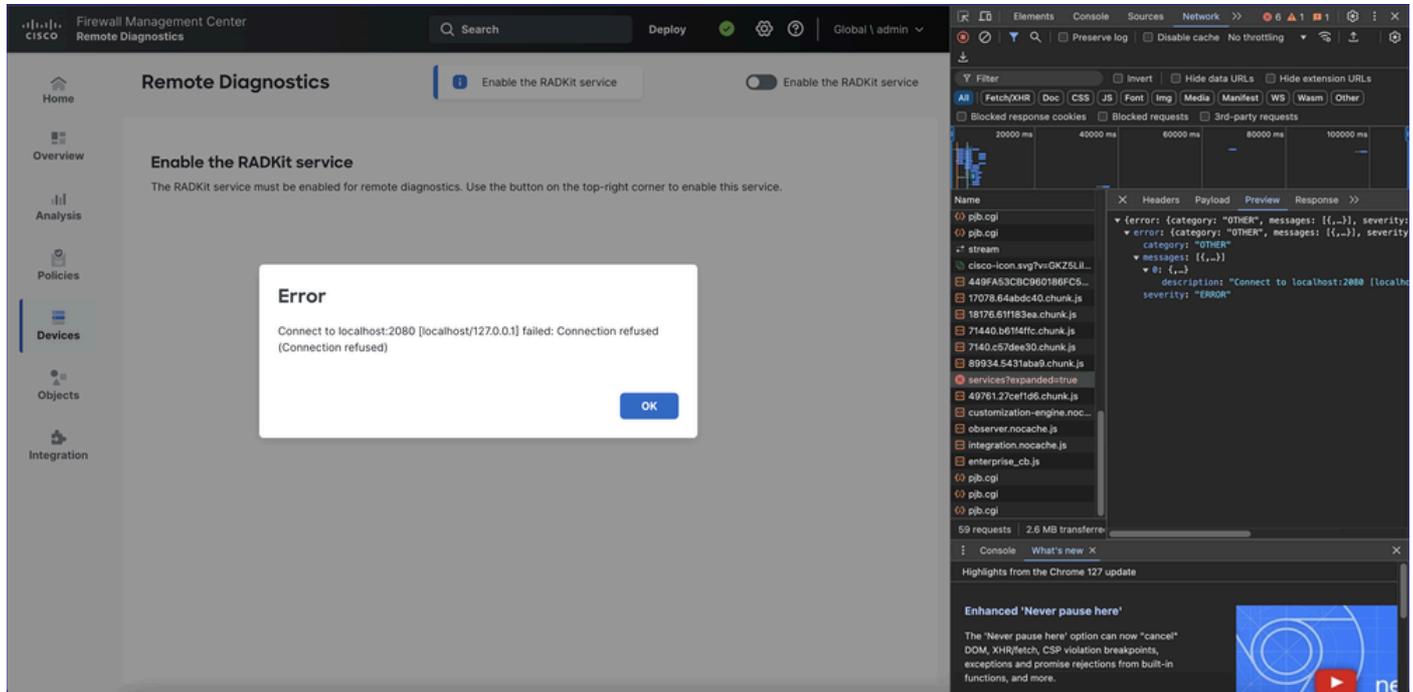
故障排除示例

如果出现类似于“连接到localhost:2080 [localhost/127.0.0.1]失败”的错误：Connection refused(Connection refused)，请尝试从FMC SSH会话重新启动auth-daemon:

```
<#root>
```

```
root@firepower:~$
```

```
sudo pmtool restartbyid auth-daemon
```



遥测

已为此功能添加遥测输出：

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

常见问题

常见问题解答:登录和注册

问：如果FMC无法直接访问互联网，注册是否与代理一起使用？

A.如果代理有权访问prod.radkit-cloud.cisco.com（用于注册流程），则为Yes。

问：用户能否将自己的IdP用于此服务？

答：RADKit云仅接受思科SSO。您可以选择将您的公司帐户与思科帐户关联，以便可以通过非思科

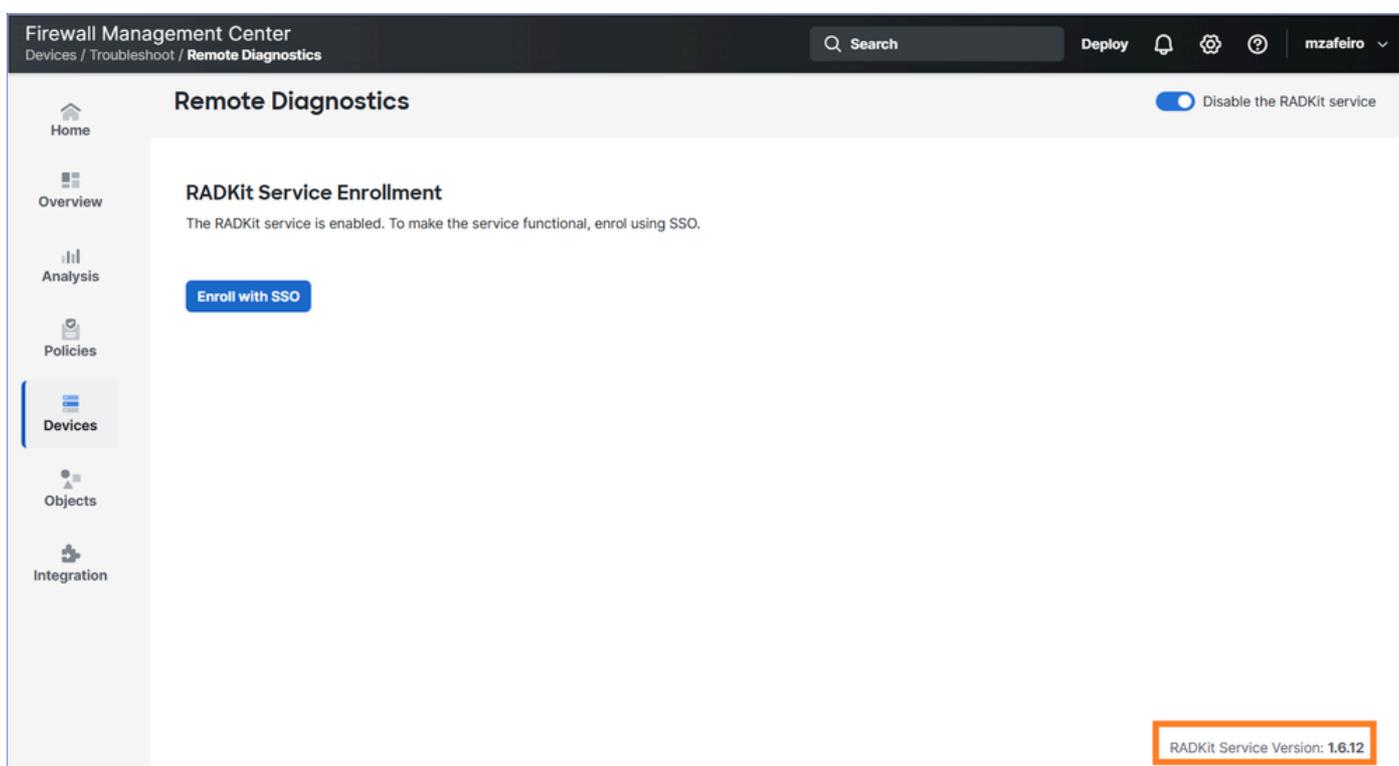
电子邮件进行RADKit服务注册。

常见问题解答:RADKit版本

问：7.7版的FMC中包含哪个版本的RADkit?我们如何知道FMC中包含哪个RADKit版本？这是否可以在不升级FMC的情况下进行更新？

A.

- 7.7.0附带的RADKit版本为1.6.12。
- RADKit服务的版本显示在FMC远程诊断(FMC Remote Diagnostics)页面的底部：“RADKit服务版本：1.6.12”



- RADKit与FMC升级包/修补程序捆绑在一起。不支持在FMC中单独升级RADKit服务。

常见问题解答:Other (其他)

问：是否可以包括非由FMC管理的外部设备？

答：只有由FMC管理的设备才能添加到RADKit资产，然后通过授权进行访问。

问：RADKit配置是否作为FMC备份的一部分进行备份？

A.

- 配置不会作为FMC备份的一部分进行备份。

- 它未备份，因为我们预计通常不会提供永久访问；访问通常只限于有限时间。

参考

实用链接:

- [FMC配置指南 — RADKit](#)
- <https://radkit.cisco.com/>
- <https://radkit.cisco.com/docs/index.html>
- <https://radkit.cisco.com/downloads/release/>
- <https://github.com/Cisco-RADKit/Intro>

关于此翻译

思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。