

# 排除证书错误故障"；无法在FMC上配置CA证书"

## 目录

---

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[背景信息](#)

[问题](#)

[解决方案](#)

[步骤1:找到.pfx证书](#)

[第二步：从.pfx文件中提取证书和密钥](#)

[第三步：在文本编辑器中验证证书](#)

[第四步：验证记事本中的私钥](#)

[第五步：拆分CA证书](#)

[第六步：合并PKCS12文件中的证书](#)

[步骤 7.在FMC中导入PKCS12文件](#)

[验证](#)

---

## 简介

本文档介绍如何对由FMC管理的Firepower威胁防御设备上的证书颁发机构(CA)导入错误进行故障排除和修复。

## 先决条件

### 要求

Cisco 建议您了解以下主题：

- 公用密钥基础结构 (PKI)
- Firepower Management Center (FMC)
- Firepower Threat Defense (FTD)
- OpenSSL

### 使用的组件

本文档中的信息基于以下软件版本：

- MacOS x 10.14.6
- FMC 6.4
- OpenSSL

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您使用的是真实网络，请确保您已经了解所有命令的潜在影响。

## 背景信息



注意：在FTD设备上，在生成证书签名请求(CSR)之前需要CA证书。

- 如果CSR在外部服务器（例如Windows Server或OpenSSL）中生成，则手动注册方法将失败，因为FTD不支持手动密钥注册。必须使用其他方法，例如PKCS12。

## 问题

在此特定场景中，FMC在CA证书状态（如图所示）中显示一个红十字，表示证书注册无法安装CA证书。如果证书未正确打包，或PKCS12文件不包含正确的颁发者证书（如图所示），则通常会出现此错误。

Name	Domain	Enrollment Type	Status
wildcard-certificate-2020	Global	PKCS12 file	<span style="color: red;">✘</span> CA <span style="color: blue;">ID</span>



注意：在较新的FMC版本中，已解决此问题以匹配ASA行为，该行为会创建另一个信任点，其根CA包含在.pfx证书的信任链中。

## 解决方案

### 步骤1:找到.pfx证书

获取在FMC GUI中注册的pfx证书保存，然后在Mac终端(CLI)中查找该文件。

```
docs# ls -l
total 16
-rw-r--r--  1 holguins  staff  4701 May 23 15:11 cert.pfx
```

ls

### 第二步：从.pfx文件中提取证书和密钥

从pfx文件提取客户端证书（非CA证书）（需要用于生成.pfx文件的密码）。

```
openssl pkcs12 -in cert.pfx -clcerts -nokeys -out id.pem
```

```
docs# openssl pkcs12 -in cert.pfx -clcerts -nokeys -out id.pem
Enter Import Password:
MAC verified OK
```

身份导出

提取CA证书 ( 而不是客户端证书 ) 。

```
openssl pkcs12 -in cert.pfx -cacerts -nokeys -out certs.pem
```

```
docs# openssl pkcs12 -in cert.pfx -cacerts -nokeys -out certs.pem
Enter Import Password:
MAC verified OK
```

缓存导出

从pfx文件中提取私钥 ( 需要步骤2中的相同口令 ) 。

```
openssl pkcs12 -in cert.pfx -nocerts -out key.pem
```

```
docs# openssl pkcs12 -in cert.pfx -nocerts -out key.pem
Enter Import Password:
MAC verified OK
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
```

密钥导出

现在存在四个文件 : cert.pfx ( 原始pfx捆绑包 ) 、 certs.pem ( CA证书 ) 、 id.pem ( 客户端证书 ) 和 key.pem ( 私钥 ) 。

```
docs# ls -l
total 40
-rw-r--r--  1 holguins  staff  4701 May 23 15:11 cert.pfx
-rw-r--r--  1 holguins  staff  2301 Jun 10 01:34 certs.pem
-rw-r--r--  1 holguins  staff  2410 Jun 10 01:34 id.pem
-rw-r--r--  1 holguins  staff  1958 Jun 10 01:34 key.pem
docs#
```

导出后的ls

**第三步 : 在文本编辑器中验证证书**

使用文本编辑器 ( 例如nano certs.pem ) 验证证书。

对于此特定场景 , certs.pem仅包含子CA ( 颁发CA ) 。

从步骤5开始，本文描述文件certs.pem包含2个证书（一个根CA和一个子CA）的方案的过程。

```
Bag Attributes: <No Attributes>
subject=/C=MX/ST=CDMX/O=Ungu Corp/OU=Ungu Corp Certificate Authority/CN=Ungu Corp Intermediate CA
issuer=/C=MX/ST=CDMX/O=Ungu Corp/OU=Ungu Corp Certificate Authority/CN=Ungu Corp Root CA
-----BEGIN CERTIFICATE-----
MIIF0zCCA7ugAwIBAgICEAUwDQYJKoZIhvcNAQELBQAwdjELMAkGA1UEBhMCTVgx
DTALBgNVBAGMBENETVgxEjAQBgNVBAoMVCVUz3UgQ29ycDEoMCMYGA1UECwwfVW5n
dSBDb3JwIENlcnRpZmljYXRlIEF1dGhvcml0eTEaMBGGA1UEAwwRVW5ndSBDb3Jw
IFJvb3QgQ0EwHhcNMjIwMjI1MDQ4WhcNMjIwMjI1MDQ4WjB+MQswCQYD
VQQGEwJNWDEMAzGA1UECAwEQ0RNWDESMBA1UECgwJVW5ndSBDb3JwMSgwJgYD
VQQLDB9Vbmd1IENvcnAgQ2VydG1maWNhdGUgQXV0aG9yaXR5MSIwIAYDVQQDDb1V
bmd1IENvcnAgS5W0ZXJtZWZlYXRlIENBMTIjANBgkqhkiG9w0BAQEFAAOCAg8A
MIICCgKCAgEAt9zB5lbrhMTEEyGmRVRnuQ+mt86axF3QZEeSYCfV5gZs9R25cw+N
L7U9agbL/bNfvR00N8I8ywVahITWJP9kuzGksEDaUzyHXybDslyPhUNt0fYn5zFi
GGa8lr90KmxSpsXeQB+GB0D8wezA1bAAGSKDiQymtBdQQMpnKTCmCRCjcPD1rBq1
Ewi0/7ePWhHK4KhtBBfSmjQxZYb1QIG5DBWCKA4q2DlME9/o+pL944Utw+HMLrAH
4bT86kT7cYQVbeVSmocastuN+1jux2aJ+4jT0GJM44yn0KzVANo1gEjw/DPhW460
u9I1oJGMCh4j7Efl8bYvHTd+8yEejmHR+ASycsy+8qoymWq3wIPiWJA0r160Hn2c
JOZpu2oQQs+90+wBrzn/yV7aZmVDdbEJSXKHJKIGA7k5VWe/CvXbfExHSCfdZ5EV
uIx4AixdgdwEdd0rghVY0GS1IHBmXNKoPp6s41oLmSm5r8lgZqm5mgdDlUKNA8tG
0jVrURiHLalHhyynoYHHVihEjhPrjNL9T26Dq9iAhX6yMclIXB1QG/QUxef7AL07
nzIBAsrYnAEv+TvqYkRE4Z9gVxYhNLpxnVg0ycHiZbco2IcQzqIwDQAqQS2LRWP
8eNuPd9l+5BgsSYgK3NxpZMXZwmMXgnGye3lueBUL9DSkuknx0aFVMCAwEAAANj
MGEwHQYDVR00BBYEFEDAVTSyUoHTThBtxlvip1L0TEQoMB8GA1UdIwQYMBaAFJMO
DF6TWO6EkboLkLC0t59z01QwMA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/BAQD
AgGGA0GCSqGSIb3DQEBCwUAA4ICAQBUNUuk9jMTGmcP6j/tqBFM3Inhj/84ABMY
T4RBdtxi1v5HPjtknyEIp1B31QxrWi4pLiyh0ILb181mNxnawZDOMvzv7Bsxpvx
xHrGhGac2y4yT72vGcIp/++8H2LatFaGAGePissCjzTcLG9brubP/MXYJ3MrlGXl
FbqvTdDJS5qB0+jRnMbACbV/nTUVXl6f6vb3AW2Zy0/u0+S6VoIB5Uk4xLZuhrwl
IXxSTghQWLqK4FBLj+XxyK2u+10iR3+6JGkkaIbb62zJsklnSJ+gVHgsMhEjATto
H0Zw5+uoJQyl/pa4uk0UaRpkSicH82p+4gPeCg5cEQAcI4niqJgIH0oPYJQszRwD
IB2w3nTAaNMTDyH6Ih/N/MvPihaiYI3jynGEMjansw8zcBPoeak4bTsEx3hu7a/
kWddLmv2TscsfkGCL0XL0fcJLcW4R6QvsZaj3Ia0AsX/Lm0eYb7RnXfjPHenp3rA
a9I0LNe9/AyQrAqp3hQ4XSNs3zgScCja40ZcXiSgJcf1XI58Ml2phT4bob89vY+u
xIawv6bXIteQE7P2RBUeJWPMFclJ75JMplRYsj2xogkneMiPpc9w5moZLxZpvznqgy
aCi37m1d+CT6hYTWxe3HztS03VJ+24IqEr+wmi+FB04VHZtqc/Bpajb0TpGBUGex
wxMFkoFWSA==
-----END CERTIFICATE-----
```

证书视图

#### 第四步：验证记事本中的私钥

使用文本编辑器（例如nano certs.pem）验证key.pem文件的内容。



```
Bag Attributes
  localKeyID: 4B ED BA 56 76 3A C9 22 C3 75 54 A7 0A 1A F1 7D 3B 5E B0 D8
Key Attributes: <No Attributes>
-----BEGIN ENCRYPTED PRIVATE KEY-----
MIIFDjBABgkqhkiG9w0BBQ0wMzAbBgkqhkiG9w0BBQwwDgQI9vQUkrEl0MMCAggA
MBQGCCqGSIB3DQMHBajnRV9YTIYsSwSCBMjqf1Lhs3v0RL0DHkvi7yvWSd3xWLMn
jt1hgOLsU1TDmBAWp/LXpqSP27c4XCQiZc0eiFDqm8aKw9xTDjgkEUBVactZs+Sz
yCE1gcG6NRH9lZFiwOYy+MCR4EPYh06DJPQ+MxLvtjjHrErruyXlAlYwfAtrAcQk
E5tJniCaNTppwFVofLpd/oHa2tF0kBMVVjS3HyxAXEfNThmzMwKRYgsLPUKShTfb
iv0bu8zI6fVfB4db3J/FjqikoiCHKXbWetm/GewZ071H3DW0HamtPw3InUuvYuzv
SM08x+Nji6uo7gtrQ6Rld2z5fN6vTtAw3x10AHjxm+vf0xt95zXhABYkMg2zHaQg
0djTHYFwDhwpdmSSNWM8hWnY8AvFxdjXURp/5MNP+v6ty5W/ETVe6o+Dh1sa9i7v
PzUwIDWs6kt0rxp0v8200lmqSKD6C4UnD1Vf2hH7AyMuwRpYamOEIuPtg8GgeiHJ
6vxpIJ3dY9/s0eyElkvKimZgiXpexBV/nDnksCLJTgyR08AE56iq2+XiBkwIoUai
QTZNi3S+PmPf8glHFtVKR8V6Zk4u8xov3reNTTmKXXcH3mHPaMU/Nhdldn8fpx+
phTzULmdtIpD3r1Hknh0uMvjrw0RYTLp1WGiEo5DU1SyI5jrMcyA0mhufOI7vtPp
rQqXNo6JleXuBteWSIHdqFynrtIdLyUVhK5QwF40m9+OvGkXNuqMDv4fH4+7nv9l
KqK2NS4yUXW1KjbaFe+Cxz9E7stt4NyvwX56l/FpYLHymYDjQA3kFrC3tPHeULjT
fp95fJ+6g2R0nr4yKerHbV5BAaiOV3rRVpBWhgzBK5o3w4+C+QIH6tgD1f2Jp9YA
TZLj/BDxIc40Q6AORATjWcbE1fvuNmNvMEQpDFM0gP8mlqnnBGzc5mwxC1xTNcQD
nmaFYykWvXyCzsvQAgwkvzyZw2mPNQpj3lVIOVRdZy8NWVkkCBLpq2XTSA6AQIK
mnJLY+rSAEi6miVnHeUW683un8KND9+HQ1YZbpKDK+JGcwKp/KhEHKmipEoHS8b5
MLby4tL7qrA3sfddMooJJYsCC372WYrd8xPrDZ9kYJ0N64ks9sYhvRUxRMJaxqaY
Int7b6p90i1r0LpielhUUrEvbu0CudM4sLDyXq8Fqf9G5u8dMuchCjXrEPGhmf4Y
ZhTfQF3xxQYtLBbLfwEQUft6GBsJMLGZFTFPM06/e3vToRu/Opw4Z9hrA6zBfQWa
bcT868DNME+UQxoT825SLwsFFPcj0pixn21FSm6baiq6QWvTV9+5797+AEPnG23i
1x/KKsvTEuwyHGgAX6p9Z0bfP0VcikMzk09MvMDU5MOUm01bnb0zINrrblG0qmRX
SYNNOL7lJ3joAKzv056KURWAMk9tQE8hAefWAZHS1PacwgUIWxOSAszRMkneptiR
VCm5UvzbYiMIA0rJjx6PTakuPIhdfokLyWfMI74ETao0Hl7KdDD1i/w11fAWFqtN
2gzfPw7Q02F7iZiYtxV9ryVBnH4wqut9pFjPYGu2oXC5q4Y7lJ1DrMzc879vAchM
C1dBcaJLWdpdTmrg2WNiao/rv3A20JjP0zAOeUwRo9r50S0oF9ez1ghBpAAtehyi
FGY=
-----END ENCRYPTED PRIVATE KEY-----
```

## 第五步：拆分CA证书

对于certs.pem文件具有2个证书（1个根CA和1个子CA）的情况，需要从信任链中删除根CA才能在FMC中导入pfx格式的证书，同时仅将子CA保留在链中以用于验证。

将certs.pem拆分为多个文件，下一个命令将证书重命名为cacert-XX。

```
split -p "-----BEGIN CERTIFICATE-----" certs.pem cacert-
```

```
docs# split -p "-----BEGIN CERTIFICATE-----" certs.pem cacert-
docs#
```

拆分

```
docs# ls -l
total 56
-rw-r--r--  1 holguins  staff    219 Jun 10 01:46 cacert-aa
-rw-r--r--  1 holguins  staff   2082 Jun 10 01:46 cacert-ab
-rw-r--r--  1 holguins  staff   4701 May 23 15:11 cert.pfx
-rw-r--r--  1 holguins  staff   2301 Jun 10 01:34 certs.pem
-rw-r--r--  1 holguins  staff   2410 Jun 10 01:34 id.pem
-rw-r--r--  1 holguins  staff   1958 Jun 10 01:34 key.pem
docs#
```

拆分后的ls

使用下面描述的命令将.pem扩展名添加到这些新文件。

```
for i in cacert-*;do mv "$i" "$i.pem";done
```

```
docs# for i in cacert-*;do mv "$i" "$i.pem";done
docs#
```

重命名脚本

检查这两个新文件，并使用所述的命令确定哪个文件包含根CA，哪个文件包含子CA。

首先，查找id.pem文件（即身份证书）的颁发者。

```
openssl x509 -in id.pem -issuer -noout
```

```
docs# openssl x509 -in id.pem -issuer -noout
issuer= /C=MX/ST=CDMX/O=Ungu Corp/OU=Ungu Corp Certificate Authority/CN=Ungu Corp Intermediate CA
```

颁发者视图

现在，找到两个cacert-files（CA证书）的主题。

```
openssl x509 -in cacert-aa.pem -subject -noout
openssl x509 -in cacert-ab.pem -subject -noout
```

```
docs# openssl x509 -in cacert-ab.pem -subject -noout
subject= /C=MX/ST=CDMX/O=Ungu Corp/OU=Ungu Corp Certificate Authority/CN=Ungu Corp Intermediate CA
```

主题检查

将Subject与id.pem文件的颁发者（如前面的图像所示）匹配的cacert文件是随后用于创建PFX证书

的子CA。

删除没有匹配主题的cacert文件。在本例中，该证书是cacert-aa.pem。

```
rm -f cacert-aa.pem
```

## 第六步：合并PKCS12文件中的证书

在新的pfx文件中合并子CA证书（在本例中，名称为cacert-ab.pem）以及ID证书(id.pem)和私钥(key.pem)。您必须使用密码保护此文件。如果需要，请更改cacert-ab.pem文件名以匹配您的文件。

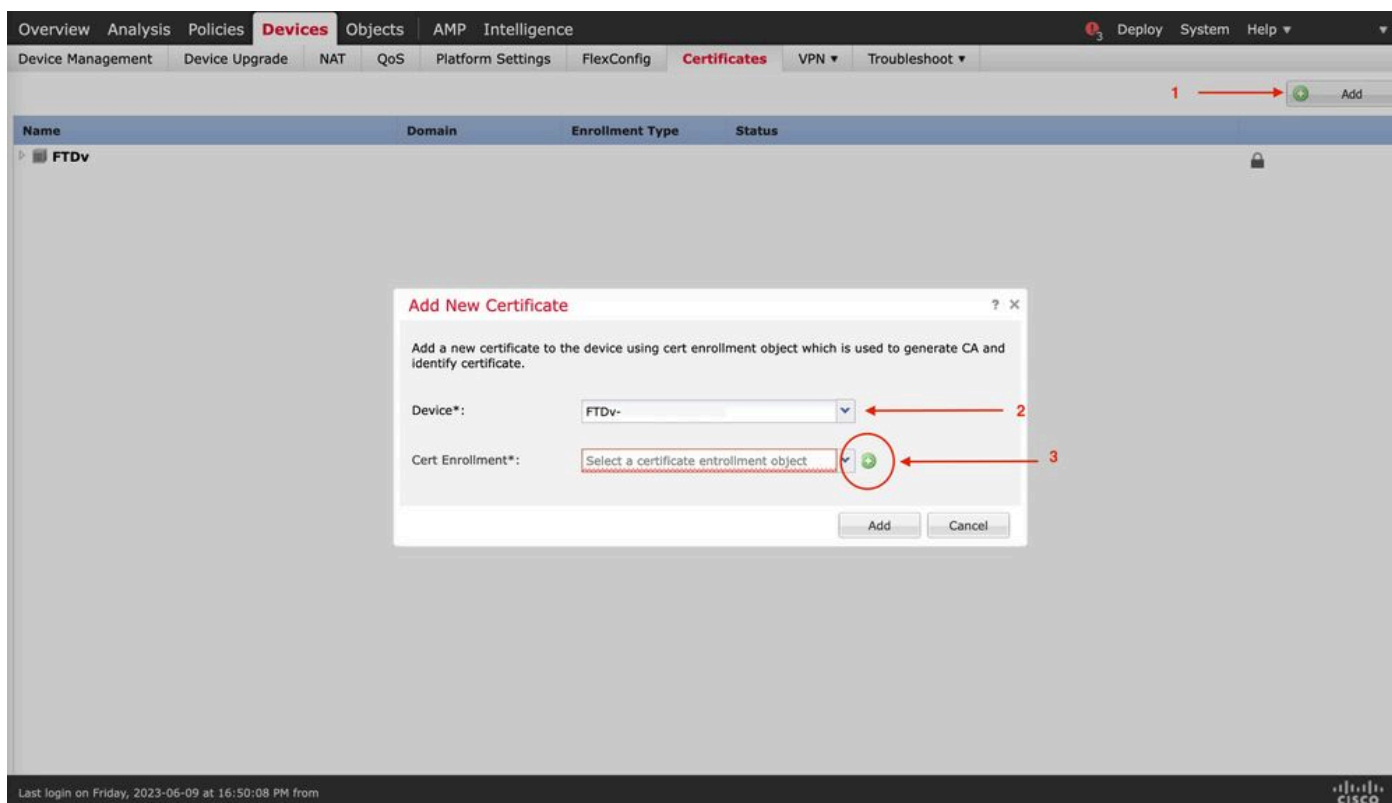
```
openssl pkcs12 -export -in id.pem -certfile cacert-ab.pem -inkey key.pem -out new-cert.pfx
```

```
docs# openssl pkcs12 -export -in id.pem -certfile cacert-ab.pem -inkey key.pem -out new-cert.pfx
Enter Export Password:
Verifying - Enter Export Password:
```

pfx-creation

## 步骤 7.在FMC中导入PKCS12文件

在FMC中，导航到Device > Certificates，并将证书导入所需的防火墙，如图所示。



证书注册

插入新证书的名称。

### Add Cert Enrollment

? X

Name\*

Description

**CA Information** Certificate Parameters Key Revocation

Enrollment Type:

PKCS12 File\*:

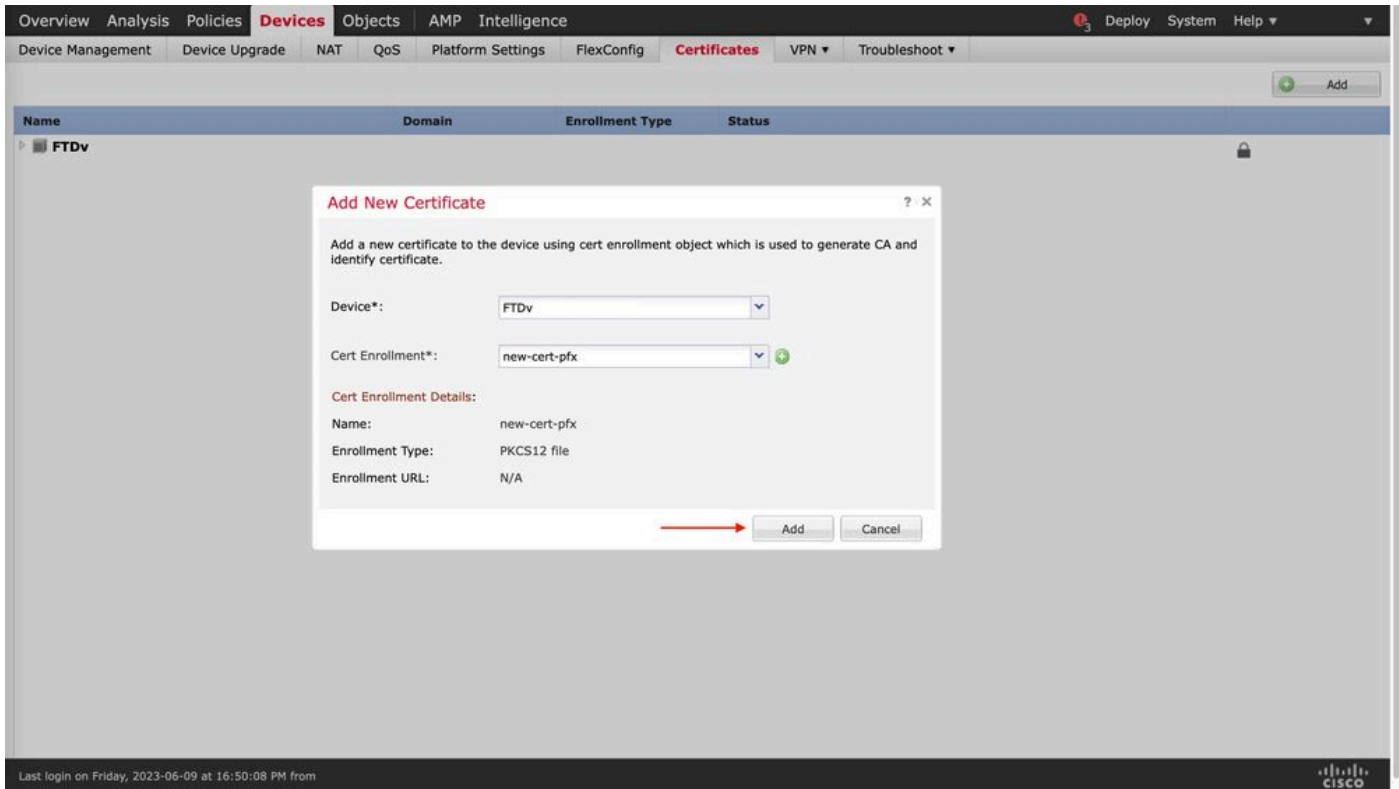
Passphrase:

Allow Overrides

注册

添加新证书，然后等待注册过程将新证书部署到FTD。





new-cert

新证书必须可见，在CA字段中不能有红十字。

## 验证

使用本部分可确认配置能否正常运行。

在Windows中，您可能会遇到以下问题：即使.pfx文件仅包含ID证书，操作系统仍会显示证书的整个链（如果其存储中包含subCA，CA链）。

要检查.pfx文件中的证书列表，可以使用certutil或openssl等工具。

```
certutil -dump cert.pfx
```











certutil是一个命令行实用程序，它提供.pfx文件中的证书列表。您必须看到包含ID、SubCA、CA（如果有）的整个链。

或者，您可以使用openssl命令，如下面的命令所示。

```
openssl pkcs12 -info -in cert.pfx
```

要验证证书状态以及CA和ID信息，您可以选择图标并确认已成功导入：

Device Management NAT VPN QoS Platform Settings FlexConfig **Certificates** + Add

Name	Domain	Enrollment Type	Status	
<b>FPR1k</b>				
wildcard-certificate-2020	Global	PKCS12 file	 CA 	  
new-cert-pfx	Global	PKCS12 file	 CA 	  

## 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）。