配置IOx包簽名驗證

目錄

[簡介](#)
[必要條件](#)
[需求](#)
[採用元件](#)
[背景資訊](#)
[設定](#)
[步驟1.建立CA金鑰和憑證](#)
[步驟2.生成用於IOx的信任錨](#)
[步驟3.在IOx裝置上匯入信任金鑰](#)
[步驟4.建立應用程式專屬金鑰和CSR](#)
[步驟5.使用CA簽署應用特定證書](#)
[步驟6.將IOx應用程式打包並使用特定於應用程式的證書對其進行簽名](#)
[步驟7.將簽名的IOx包部署到啟用簽名的裝置上](#)
[驗證](#)
[疑難排解](#)

簡介

本文檔詳細介紹如何在IOx平台上建立和使用已簽名的軟體包。

必要條件

需求

思科建議您瞭解以下主題：

- Linux基礎知識
- 瞭解憑證運作方式

採用元件

本文中的資訊係根據以下軟體和硬體版本：

- 為IOx配置的支援IOx的裝置：
  已配置IP地址執行中的訪客作業系統(GOS)和思科應用程式架構(CAF)網路地址轉換(NAT)配置為訪問CAF（埠8443）
- 安裝了開放安全套接字層(SSL)的Linux主機
- IOx客戶端安裝檔案，可從以下位置下載：https://software.cisco.com/download/release.html?mdfid=286306005&softwareid=286306762

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設

）的組態來啟動。如果您的網路正在作用，請確保您已瞭解任何指令可能造成的影響。

背景資訊

自IOx發行以來，支援AC5應用程式套件簽名。此功能可確保應用程式套件有效並且從受信任的來源獲取裝置上安裝的應用程式套件。如果在平台中開啟了應用程式套件簽名驗證，則只有這樣才能部署簽名的應用程式。

設定

使用包簽名驗證需要執行以下步驟：

1. 建立證書頒發機構(CA)金鑰和證書。
2. 生成用於IOx的信任錨。
3. 在IOx裝置上匯入信任金鑰。
4. 建立應用程式專屬金鑰和憑證簽署請求(CSR)。
5. 使用CA對應用程式特定的證書進行簽名。
6. 將IOx應用程式打包，使用特定於應用程式的證書對其進行簽名。
7. 將已簽名的IOx包部署到啟用簽名的裝置上。

附註：對於本文而言，自簽名CA用於生產方案。最佳選擇是使用官方CA或您公司的CA簽署。

附註：選擇CA、金鑰和簽名的選項只是為了實驗目的，可能需要根據您的環境進行調整。

步驟1.建立CA金鑰和憑證

第一步是建立您自己的CA。這只需產生CA的金鑰和該金鑰的憑證即可完成：

若要產生CA金鑰：

```
[jedepuyd@KJK-SRVIOT-10 signing]$ openssl genrsa -out rootca-key.pem 2048
Generating RSA private key, 2048 bit long modulus
.....................................................................................+++
.......+++
e is 65537 (0x10001)
```

若要產生CA憑證：

```
[jedepuyd@KJK-SRVIOT-10 signing]$ openssl req -x509 -new -nodes -key rootca-key.pem -sha256 -
days 4096 -out rootca-cert.pem
You are about to be asked to enter information that is incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name (DN).
There are quite a few fields but you can leave some blank
For some fields there can be a default value,
If you enter '.', the field can be left blank.
-----
```

```
Country Name (2 letter code) [XX]:BE
State or Province Name (full name) []:WVL
Locality Name (eg, city) [Default City]:Kortrijk
Organization Name (eg, company) [Default Company Ltd]:Cisco
Organizational Unit Name (eg, section) []:IOT
Common Name (eg, your name or your server's hostname) []:ioxrootca
Email Address []:
```

必須調整CA證書中的值以匹配您的使用案例。

步驟2.生成用於IOx的信任錨

現在，您已經擁有了您的CA所需的金鑰和證書，可以建立一個用於您的IOx裝置的信任錨點捆綁包。信任錨點捆綁包必須包含完整的CA簽名鏈（如果中間證書用於簽名）和用於提供（自由形式）後設資料的info.txt檔案。

首先，建立info.txt檔案，並在其中放置一些後設資料：

```
[jedepuyd@KJK-SRVIOT-10 signing]$ echo "iox app root ca v1">info.txt
```

或者，如果您有多個CA憑證，若要形成CA憑證鏈結，需要將它們放在一個.pem中：

```
cat first_cert.pem second_cert.pem > combined_cert.pem
```

> **附註**：由於單個CA根證書用於直接簽名，因此本文不需要此步驟，建議不要將其用於生產，並且根CA金鑰對必須始終離線儲存。

CA憑證鏈結需要命名為ca-chain.cert.pem，因此請準備以下檔案：

```
[jedepuyd@KJK-SRVIOT-10 signing]$ cp rootca-cert.pem ca-chain.cert.pem
```

最後，您可以將ca-chain.cert.pem和info.txt合併到gzipped tar中：

```
[jedepuyd@KJK-SRVIOT-10 signing]$ tar -czf trustanchorv1.tar.gz ca-chain.cert.pem info.txt
```

步驟3.在IOx裝置上匯入信任金鑰

在上一步中建立的trustanchorv1.tar.gz需要匯入到IOx裝置中。套件組合中的檔案用於驗證應用程式是否已在允許安裝之前從正確的CA使用CA簽署的憑證簽署。

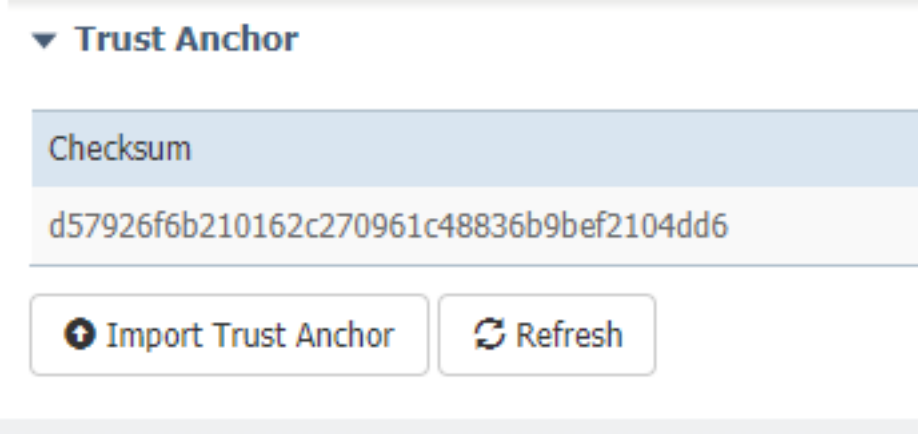
信任錨點的匯入可以通過ioxlicent完成：

```
[jedepuyd@KJK-SRVIOT-10 signing]$ ioxclient platform signedpackages trustanchor set
trustanchorv1.tar.gz
Currently active profile :  default
Command Name:  plt-sign-pkg-ta-set
Response from the server:  Imported trust anchor file successfully
[jedepuyd@KJK-SRVIOT-10 signing]$ ioxclient platform signedpackages enable
```

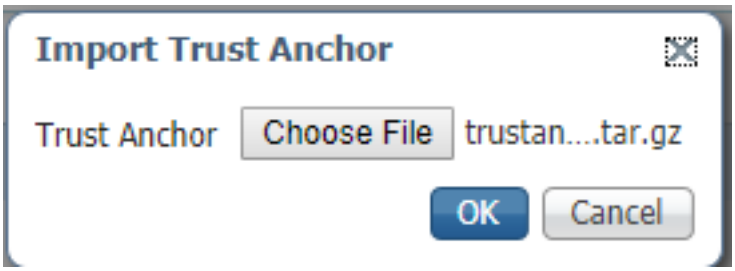
```
Currently active profile :  default
Command Name:  plt-sign-pkg-enable
Successfully updated the signed package deployment capability on the device to true
```

另一種方法是通過本地管理器匯入信任錨點：

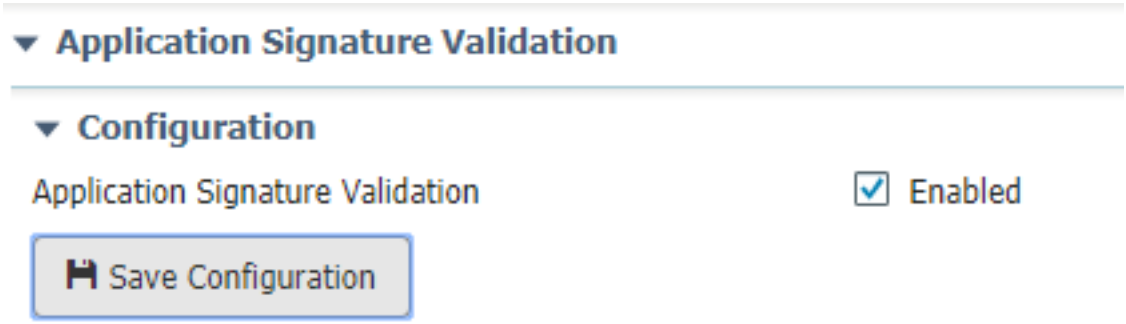
導覽至**System Setting > Import Trust Anchor**，如下圖所示。



選擇在步驟2中生成的檔案。然後按一下**OK**，如下圖所示。



成功匯入信任錨點後，請選中**Enabled**進行**Application Signing Validation**，然後按一下**Save Configuration**，如下圖所示：



步驟4.建立應用程式專屬金鑰和CSR

接下來，您可以建立用於登入IOx應用程式的金鑰和證書對。最佳實踐是為計畫部署的每個應用程式生成一個特定的金鑰對。

只要每個證書使用同一個CA簽名，它們就會被視為有效。

要生成應用程式特定金鑰：

```
[jedepuyd@KJK-SRVIOT-10 signing]$ openssl genrsa -out app-key.pem 2048
Generating RSA private key, 2048 bit long modulus
.......+++
...+++
e is 65537 (0x10001)
```

若要產生CSR:

```
[jedepuyd@KJK-SRVIOT-10 signing]$ openssl req -new -key app-key.pem -out app.csr
You are about to be asked to enter information that is incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name (DN).
There are quite a few fields but you can leave some blank.
For some fields there can be a default value,
If you enter '.', the field can be left blank.
-----
Country Name (2 letter code) [XX]:BE
State or Province Name (full name) []:WVL
Locality Name (eg, city) [Default City]:Kortrijk
Organization Name (eg, company) [Default Company Ltd]:Cisco
Organizational Unit Name (eg, section) []:IOT
Common Name (eg, your name or your server's hostname) []:ioxapp
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

對於CA，必須調整應用證書中的值以匹配您的使用案例。

步驟5.使用CA簽署應用特定證書

現在，您已經具備了您的CA和應用CSR的要求，因此您可以使用CA簽署CSR。結果是簽名的特定於應用程式的證書：

```
[jedepuyd@KJK-SRVIOT-10 signing]$ openssl x509 -req -in app.csr -CA rootca-cert.pem -CAkey
rootca-key.pem -CAcreateserial -out app-cert.pem -days 4096 -sha256
Signature ok
subject=/C=BE/ST=WVL/L=Kortrijk/O=Cisco/OU=IOT/CN=ioxapp
Getting CA Private Key
```

步驟6.將IOx應用程式打包並使用特定於應用程式的證書對其進行簽名

此時，您已經準備就緒，可以打包您的IOx應用程式，並使用步驟4中生成的金鑰對對其進行簽名。然後在步驟5中由CA進行簽名。

為應用程式建立source和package.yaml的其餘過程保持不變。

使用金鑰對封裝IOx應用程式：

```
[jedepuyd@KJK-SRVIOT-10 iox_docker_pythonsleep]$ ioxclient package --rsa-key ../signing/app-
key.pem --certificate ../signing/app-cert.pem .
Currently active profile :  default
Command Name:  package
Using rsa key and cert provided via command line to sign the package
Checking if package descriptor file is present..
Validating descriptor file /home/jedepuyd/iox/iox_docker_pythonsleep/package.yaml with package
```

```
schema definitions
Parsing descriptor file..
Found schema version  2.2
Loading schema file for version  2.2
Validating package descriptor file..
File /home/jedepuyd/iox/iox_docker_pythonsleep/package.yaml is valid under schema version 2.2
Created Staging directory at :  /tmp/666018803
Copying contents to staging directory
Checking for application runtime type
Couldn't detect application runtime type
Creating an inner envelope for application artifacts
Excluding  .DS_Store
Generated  /tmp/666018803/artifacts.tar.gz
Calculating SHA1 checksum for package contents..
Package MetaData file was not found at  /tmp/666018803/.package.metadata
Wrote package metadata file :  /tmp/666018803/.package.metadata
Root Directory :  /tmp/666018803
Output file:  /tmp/096960694
Path:  .package.metadata
SHA1 : 2a64461a921c2d5e8f45e92fe203127cf8a06146
Path:  artifacts.tar.gz
SHA1 : 63da3eb3d81e13249b799bf57845f3fc9f6f2f94
Path:  package.yaml
SHA1 : 0e6259e49ff22d6d38e6d1913759c5674c5cec6d
Generated package manifest at  package.mf
Signed the package and the signature is available at package.cert
Generating IOx Package..
Package generated at /home/jedepuyd/iox/iox_docker_pythonsleep/package.tar
```

步驟7.將簽名的IOx包部署到啟用簽名的裝置上

該過程的最後一步是將應用程式部署到IOx裝置。與未簽名的應用程式部署相比，沒有區別：

```
[jedepuyd@KJK-SRVIOT-10 iox_docker_pythonsleep]$ ioxclient app install test package.tar
Currently active profile :  default
Command Name:  application-install
Saving current configuration
Installation Successful. App is available at :
https://10.50.215.248:8443/iox/api/v2/hosting/apps/test
Successfully deployed
```

驗證

使用本節內容，確認您的組態是否正常運作。

為了驗證應用程式金鑰是否正確與CA簽名，您可以執行以下操作：

```
[jedepuyd@KJK-SRVIOT-10 signing]$ openssl verify -CAfile rootca-cert.pem app-cert.pem
app-cert.pem: OK
```

疑難排解

本節提供的資訊可用於對組態進行疑難排解。

當遇到應用程式部署問題時，您可能會看到以下錯誤之一：

```
[jedepuyd@KJK-SRVIOT-10 iox_docker_pythonsleep]$ ioxclient app install test package.tar
Currently active profile :  default
Command Name:  application-install
Saving current configuration
Could not complete your command :  Error. Server returned 500
{
 "description": "Invalid Archive file: Certificate verification failed: [18, 0, 'self signed
certificate']",
 "errorcode": -1,
 "message": "Invalid Archive file"
}
```

使用CA簽署應用憑證時出現錯誤，或者與受信任的錨點套件中的憑證不匹配。

使用「驗證」部分中提到的指令，檢查您的證書以及受信任的錨點捆綁包。

這些錯誤表示您的程式包沒有正確簽名，您可以再次檢視步驟6。

```
[jedepuyd@KJK-SRVIOT-10 iox_docker_pythonsleep]$ ioxclient app install test2 package.tar
Currently active profile :  default
Command Name:  application-install
Saving current configuration
Could not complete your command :  Error. Server returned 500
{
 "description": "Package signature file package.cert or package.sign not found in package",
 "errorcode": -1009,
 "message": "Error during app installation"
}
```