Configurer la validation des signatures du package IOx

Table des matières

Introduction

Conditions préalables

- Exigences
- Composants utilisés

Informations générales

Configurer

- Étape 1 : création d'une clé et d'un certificat CA
- Étape 2. Génération d'un ancrage de confiance à utiliser sur IOx
- Étape 3. Importation de l'ancrage de confiance sur IOx-Device
- Étape 4. Créer une clé spécifique à l'application et un CSR
- Étape 5. Signature du certificat spécifique à l'application avec l'AC
- Étape 6. Créer un package pour votre application IOx et la signer avec un certificat spécifique à l'application
- Étape 7. Déployez votre package IOx signé sur un périphérique compatible avec les signatures

Vérifier

Dépannage

Introduction

Ce document décrit de manière détaillée comment créer et utiliser des packages signés sur la plate-forme IOx.

Conditions préalables

Exigences

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Connaissances de base de Linux
- Comprendre le fonctionnement des certificats

Composants utilisés

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- Périphérique compatible IOx configuré pour IOx :
  - Adresse IP configurée

    - Système d'exploitation invité (GOS) et Cisco Application Framework (CAF) qui exécute
    - Traduction d'adresses de réseau (NAT) configurée pour l'accès à CAF (port 8443)
- Hôte Linux avec SSL (Secure Sockets Layer) ouvert installé
- Fichiers d'installation du client IOx téléchargeables à l'adresse suivante : https://software.cisco.com/download/release.html?mdfid=286306005&softwareid=286306762

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Informations générales

Depuis la version IOx, la signature de package d'application AC5 est prise en charge. Cette fonctionnalité permet de s'assurer que le package d'application est valide et que celui installé sur le périphérique provient d'une source fiable. Si la validation des signatures des packages d'applications est activée sur une plate-forme, seules les applications signées peuvent être déployées.

Configurer

Les étapes suivantes sont requises pour utiliser la validation de signature de package :

1. Créez une clé et un certificat d'autorité de certification (CA).
2. Générez une ancre d'approbation à utiliser sur IOx.
3. Importez l'ancre de confiance sur votre périphérique IOx.
4. Créez une clé spécifique à l'application et une demande de signature de certificat (CSR).
5. Signez le certificat spécifique à l'application avec l'utilisation de l'autorité de certification.
6. Mettez en package votre application IOx, signez-la avec le certificat spécifique à l'application.
7. Déployez votre package IOx signé sur un périphérique compatible avec les signatures.

Remarque : Pour cet article, une autorité de certification auto-signée est utilisée dans un scénario de production. La meilleure option est d'utiliser une autorité de certification officielle ou l'autorité de certification de votre entreprise pour signer.

Remarque : Les options de l’autorité de certification, des clés et des signatures sont choisies à des fins de travaux pratiques uniquement et peuvent nécessiter un ajustement en fonction de votre environnement.

Étape 1 : création d’une clé et d’un certificat CA

La première étape consiste à créer votre propre autorité de certification. Pour ce faire, il suffit de générer une clé pour l’autorité de certification et un certificat pour cette clé :

Afin de générer la clé CA :

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

Afin de générer le certificat d'autorité de certification :

```
[jedepuyd@KJK-SRVIOT-10 signing]$ openssl req -x509 -new -nodes -key rootca-key.pem -sha256 -days 4096
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 []:
```

Les valeurs du certificat CA doivent être ajustées pour correspondre à votre cas d'utilisation.

Étape 2. Génération d'un ancrage de confiance à utiliser sur IOx

Maintenant que vous disposez de la clé et du certificat nécessaires pour votre autorité de certification, vous pouvez créer un bundle d'ancrage de confiance à utiliser sur votre périphérique IOx. Le bundle d'ancrage de confiance doit contenir la chaîne de signature complète de l'autorité de certification (dans le cas où un certificat intermédiaire est utilisé pour la signature) et un fichier info.txt qui est utilisé pour fournir les métadonnées (format libre).

Commencez par créer le fichier info.txt et y insérer des métadonnées :

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

Si vous disposez de plusieurs certificats d'autorité de certification, vous devez les regrouper en un seul fichier .pem pour former votre chaîne de certificats d'autorité de certification :

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

Remarque : Cette étape n'est pas requise pour cet article, car un seul certificat racine d'autorité de certification est utilisé pour la signature directe, ce qui n'est pas recommandé pour la production et la paire de clés d'autorité de certification racine doit toujours être stockée hors connexion.

La chaîne de certificats de l'autorité de certification doit être nommée ca-chain.cert.pem. Préparez donc ce fichier :

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

Enfin, vous pouvez combiner ca-chain.cert.pem et info.txt dans un tar gzippé :

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

Étape 3. Importation de l'ancrage de confiance sur IOx-Device

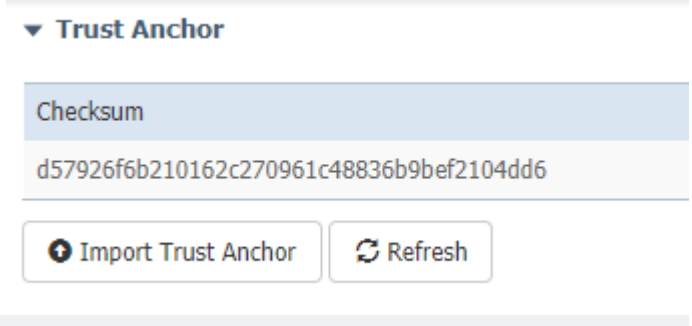
Le fichier trustanchorv1.tar.gz que vous avez créé à l'étape précédente doit être importé sur votre périphérique IOx. Les fichiers du bundle sont utilisés pour vérifier si une application a été signée avec un certificat signé par l'autorité de certification appropriée avant d'autoriser une installation.

L'importation de l'ancre de confiance peut être effectuée via l'ioxlcient :

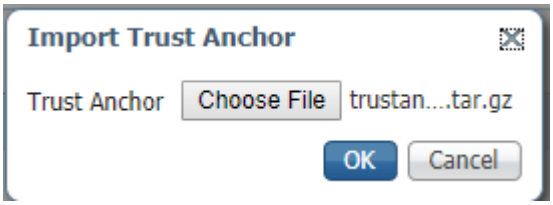
```
[jedepuyd@KJK-SRVIOT-10 signing]$ ioxclient platform signedpackages trustanchor set trustanchorv1.tar.g
Currently active profile :  default
Command Name:  plt-sign-pkg-ta-set
Response from the server:  Imported trust anchor file successfully
[jedepuyd@KJK-SRVIOT-10 signing]$ ioxclient platform signedpackages enable
Currently active profile :  default
Command Name:  plt-sign-pkg-enable
Successfully updated the signed package deployment capability on the device to true
```

Une autre option consiste à importer l'ancre d'approbation via le Gestionnaire local :

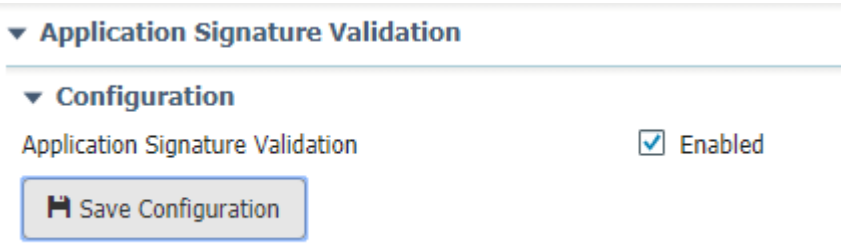
Naviguez jusqu'à System Setting > Import Trust Anchor comme indiqué dans l'image.



Sélectionnez le fichier que vous avez généré à l'étape 2. et cliquez sur OK comme indiqué dans l'image.



Après avoir importé avec succès l'ancre d'approbation, cochez Enabled pour Application Signing Validation et cliquez sur Save Configuration comme indiqué dans l'image :



Étape 4. Créer une clé spécifique à l’application et un CSR

Vous pouvez ensuite créer une paire clé/certificat utilisée pour vous connecter à votre application IOx. La meilleure pratique consiste à générer une paire de clés spécifique pour chaque application que vous prévoyez de déployer.

Tant que chacun d'eux est signé avec la même AC, ils sont tous considérés comme valides.

Afin de générer la clé spécifique à l'application :

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

Afin de générer le 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 []:
```

Comme pour l'autorité de certification, les valeurs du certificat d'application doivent être ajustées pour correspondre à votre cas d'utilisation.

Étape 5. Signature du certificat spécifique à l’application avec l’AC

Maintenant que vous avez les exigences pour votre CA et votre application CSR, vous pouvez signer le CSR avec l'utilisation de CA. Le résultat est un certificat spécifique à l'application signé :

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

Étape 6. Créer un package pour votre application IOx et la signer avec un certificat spécifique à l'application

À ce stade, vous êtes prêt à empaqueter votre application IOx et à la signer avec la paire de clés générée à l'étape 4. et signée par l'autorité de certification à l'étape 5.

Le reste du processus de création de la source et du package.yaml pour votre application reste inchangé.

package l'application IOx avec l'utilisation de la paire de clés :

```
[jedepuyd@KJK-SRVIOT-10 iox_docker_pythonsleep]$ ioxclient package --rsa-key ../signing/app-key.pem --c
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 d
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
```

Étape 7. Déployez votre package IOx signé sur un périphérique compatible avec les signatures

La dernière étape du processus consiste à déployer l'application sur votre périphérique IOx. Il n'y a aucune différence par rapport à un déploiement d'application non signé :

```
[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
```

Vérifier

Utilisez cette section pour confirmer que votre configuration fonctionne correctement.

Afin de vérifier si une clé d'application est correctement signée avec votre CA, vous pouvez faire ceci :

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

Dépannage

Cette section fournit des informations que vous pouvez utiliser pour dépanner votre configuration.

Lorsque vous rencontrez des problèmes avec le déploiement d'applications, vous pouvez voir l'une des erreurs suivantes :

```
[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 certificat
 "errorcode": -1,
 "message": "Invalid Archive file"
}
```

Un problème s'est produit lors de la signature du certificat d'application avec l'utilisation de l'autorité de certification ou il ne correspond pas à celui du bundle d'ancres de confiance.

Suivez les instructions mentionnées dans la section Vérifier pour vérifier vos certificats ainsi que le bundle d'ancres de confiance.

Ces erreurs indiquent que votre package n'a pas été signé correctement. Vous pouvez à nouveau passer à l'étape 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"
}
```

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.