

# IOx-pakket-signalering configureren

## Inhoud

[Inleiding](#)

[Voorwaarden](#)

[Vereisten](#)

[Gebruikte componenten](#)

[Achtergrondinformatie](#)

[Configureren](#)

[Stap 1. CA-toets en -certificaat maken](#)

[Stap 2. Generate Trust Anchor voor gebruik op IOx](#)

[Stap 3. Importeer vertrouwen op IOx-apparaat](#)

[Stap 4. Maak een toepassings-specifieke sleutel en CSR](#)

[Stap 5. Signaaltoepassings-specifiek certificaat met CA](#)

[Stap 6. Pak uw IOx-toepassing en teken deze met toepassings-specifiek certificaat](#)

[Stap 7. Plaats uw ondertekende IOx-pakket op een Signature-enabled apparaat](#)

[Verifiëren](#)

[Problemen oplossen](#)

## Inleiding

Dit document beschrijft op een gedetailleerde manier hoe u ondertekende pakketten kunt maken en gebruiken op het IOx-platform.

## Voorwaarden

### Vereisten

Cisco raadt kennis van de volgende onderwerpen aan:

- Basiskennis van Linux
- Begrijp hoe certificaten werken

### Gebruikte componenten

De informatie in dit document is gebaseerd op de volgende software- en hardware-versies:

- IOx-compatibel apparaat dat voor IOx is geconfigureerd:  
IP-adres ingesteld  
Guest Operating System (GOS) en Cisco Application Framework (CAF) dat ondersteuning biedt  
Network-adresomzetting (NAT) ingesteld voor toegang tot CAF (poort 8443)
- Linux-host met geïnstalleerde open Secure Socket Layer (SSL)
- IOx-clientinstallatiebestanden die kunnen worden gedownload van:  
<https://software.cisco.com/download/release.html?mdfid=286306005&softwareid=286306762>

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u de potentiële impact van elke opdracht begrijpen.

## Achtergrondinformatie

Aangezien IOx wordt vrijgegeven, wordt de ondertekening van het AC5-toepassingspakket ondersteund. Deze optie maakt het mogelijk om er zeker van te zijn dat het toepassingspakket geldig is en het pakket dat op het apparaat is geïnstalleerd, afkomstig is van een vertrouwde bron. Als Application Package Signature Validering in een platform wordt ingeschakeld, kunnen alleen dan ondertekende toepassingen worden ingezet.

## Configureren

Deze stappen zijn vereist voor het gebruik van de validatie van de handtekening van een pakket:

1. Maak een certificaat van de autoriteit (CA).
2. Generate een trust anchor voor gebruik op IOx.
3. Importeer het vertrouwde anker op uw IOx-apparaat.
4. Maak een toepassingsspecifieke toets en een certificaatsignaleringsaanvraag (CSR).
5. Teken het specifieke certificaat voor de aanvraag met behulp van de CA.
6. Verpakking van uw IOx-applicatie, teken deze met het aanvraagspecifieke certificaat.
7. Stel uw ondertekende IOx-pakket in op een kenmerkend geactiveerd apparaat.

Opmerking: Voor dit artikel wordt een zelfgetekende CA gebruikt bij een productiescenario. De beste optie is om een officiële CA of de CA van uw bedrijf te gebruiken om te ondertekenen.

Opmerking: De opties voor de CA, toetsen en handtekeningen worden alleen voor laboratoriumdoeleinden gekozen en moeten mogelijk aan uw omgeving worden aangepast.

### Stap 1. CA-toets en -certificaat maken

De eerste stap is het creëren van uw eigen CA. Dit kan simpelweg gedaan worden door het genereren van een sleutel voor de CA en een certificaat voor die sleutel:

Zo genereert u de CA-toets:

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

Zo genereert u het CA-certificaat:

```
[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 []:

De waarden in het CA-certificaat moeten worden aangepast aan uw gebruikcase.

## Stap 2. Generate Trust Anchor voor gebruik op IOx

Nu u de noodzakelijke sleutel en het certificaat voor uw CA hebt, kunt u een bundel van het trust ankerpunt voor gebruik op uw IOx apparaat creëren. De bundel van het vertrouwensankerpunt moet de volledige CA-signaalketen bevatten (indien tussentijdse certificaten voor de ondertekening worden gebruikt) en een info.txt-bestand dat wordt gebruikt om de (vrije vorm) metagegevens te verstrekken.

Maak eerst het info.txt bestand en zet er metagegevens in:

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

Optioneel: als u meerdere CA-certificaten hebt, moet u deze in één .pem samenstellen om uw CA-certificeringsketen te vormen:

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

Opmerking: Deze stap is niet vereist voor dit artikel, aangezien één enkel CA wortel certificaat wordt gebruikt om teken te richten, wordt dit niet aanbevolen voor productie en de wortel CA sleutelbaar moet altijd offline worden opgeslagen.

De CA-certificaatketen moet ca-chain.cert.pem worden genoemd, dus bereid dit bestand voor:

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

Ten slotte kan je de ca-chain.cert.pem en info.txt combineren in een gezipped tar:

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

## Stap 3. Importeer vertrouwen op IOx-apparaat

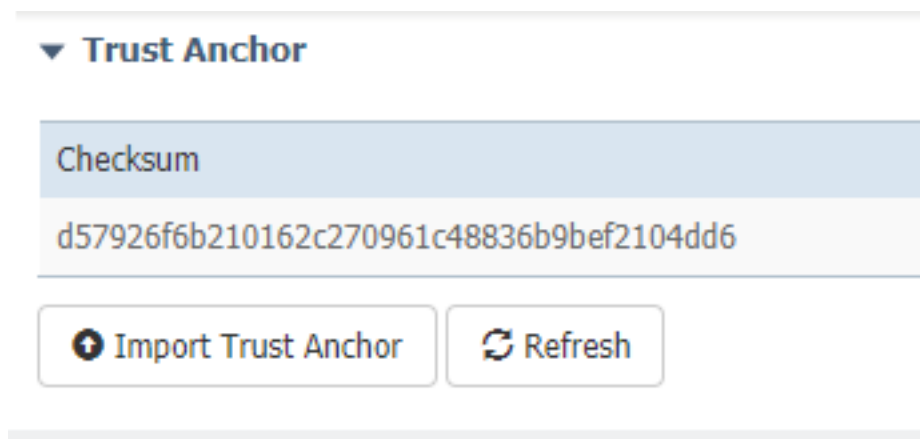
De trustanchorv1.tar.gz die u in de vorige stap hebt gemaakt, moet op uw IOx-apparaat worden geïmporteerd. De bestanden in de bundel worden gebruikt om te controleren of een toepassing is getekend met een door CA ondertekend certificaat van de juiste CA voordat deze een installatie toestaat.

De invoer van het vertrouwensanker kan geschieden via de volgende documenten:

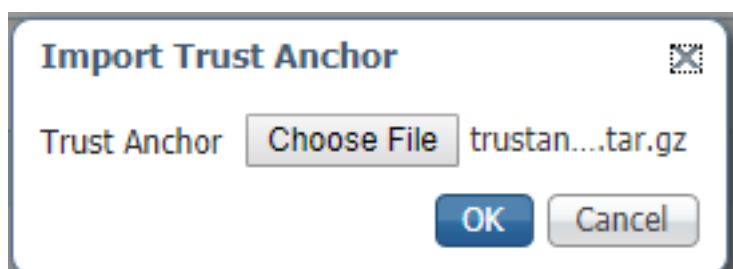
```
[jedepuyd@KJK-SRVIOT-10 signing]$ ioxclient platform signedpackages trustanchor set trustanchorv1.tar.gz
Currently active profile : default
Command Name: plt-sign-pkg-ta-set
Response from the server: Imported trust anchor file successfully
[jedepuyd@KJK-SRVIOT-10 signing]$ ioxclient platform signedpackages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

Een andere optie is om het trust anchor te importeren via Local Manager:

Navigeer naar **stroominstelling > Vertrouwen importeren** zoals in de afbeelding.



Selecteer het bestand dat u in Stap 2 gegenereerd hebt en klik op **OK** zoals in de afbeelding.




Nadat u het vertrouwde anker hebt geïmporteerd, controleert u **Ingeschakeld op Toepassingssignalering** en klikt u op **Configuratie opslaan** zoals in de afbeelding:

## ▼ Application Signature Validation

### ▼ Configuration

Application Signature Validation

Enabled

 Save Configuration

## Stap 4. Maak een toepassings specifieke sleutel en CSR

Daarna kunt u een key en een certificaatpaar maken die gebruikt wordt om in uw IOx-toepassing te tekenen. De beste praktijk is om één specifiek toetsenbord voor elke toepassing te genereren die u van plan bent in te zetten.

Zolang elk van deze overeenkomsten met dezelfde CA is ondertekend, worden ze allemaal als geldig beschouwd.

Zo genereert u de toepassings specifieke toets:

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

Zo genereert u de 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 []:

Net als bij de CA moeten de waarden in het sollicitatiecertificaat worden aangepast aan uw gebruikcase.

## Stap 5. Signaaltoepassings specifiek certificaat met CA

Nu u de vereisten voor uw CA en toepassing CSR hebt, kunt u de CSR met CA ondertekenen. Het resultaat is een ondertekend aanvraagspecifiek certificaat:

```
[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
```

## Stap 6. Pak uw IOx-toepassing en teken deze met toepassings specifiek certificaat

Op dit punt bent u klaar om uw IOx-toepassing te verpakken en het met het gegenereerde sleutelbaar uit Stap 4 te ondertekenen en in Stap 5 door de CA te ondertekenen.

De rest van het proces om de bron en het pakket te maken.yaml voor uw toepassing blijft ongewijzigd.

Verpakking IOx-toepassing met gebruik van toetsencombinatie:

```
[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
```

## Stap 7. Plaats uw ondertekende IOx-pakket op een Signature-enabled apparaat

Laatste stap in het proces zou zijn om de toepassing op uw IOx-apparaat in te voeren. Er is geen

verschil in vergelijking met een niet-ondertekende sollicitatie:

```
[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
```

## Verifiëren

Gebruik dit gedeelte om te bevestigen dat de configuratie correct werkt.

Om te verifiëren of een toepassingsleutel correct met uw CA wordt ondertekend, kunt u dit doen:

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

## Problemen oplossen

Deze sectie verschaft informatie die u kunt gebruiken om problemen met uw configuratie op te lossen.

Wanneer u problemen ondervindt met de toepassing van toepassingen, kunt u één van deze fouten zien:

```
[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"
}
```

Er is iets misgegaan bij het ondertekenen van het sollicitatiecertificaat met het gebruik van de CA of het komt niet overeen met het certificaat in de vertrouwde ankerbundel.

Gebruik de instructies in het gedeelte Verifiëren om uw certificaten en ook de bundel voor het vertrouwde anker te controleren.

Deze fout duidt erop dat uw pakket niet correct is getekend, u kunt ook naar Stap 6 kijken.

```
[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"  
}
```