

Configurar validação de assinatura do pacote IOx

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Informações de Apoio](#)

[Configurar](#)

[Etapa 1. Criar Chave CA e Certificado](#)

[Etapa 2. Gerar âncora de confiança para uso em IOx](#)

[Etapa 3. Importar âncora de confiança no dispositivo IOx](#)

[Etapa 4. Crie uma chave específica de aplicativo e CSR](#)

[Etapa 5. Assinar certificado específico de aplicativo com CA](#)

[Etapa 6. Embale seu aplicativo IOx e assine-o com um certificado específico do aplicativo](#)

[Passo 7. Implante seu pacote IOx assinado em um dispositivo habilitado para assinatura](#)

[Verificar](#)

[Troubleshoot](#)

Introduction

Este documento descreve de forma detalhada como criar e usar pacotes assinados na plataforma IOx.

Prerequisites

Requirements

A Cisco recomenda que você tenha conhecimento destes tópicos:

- Conhecimento básico do Linux
- Entender como os certificados funcionam

Componentes Utilizados

As informações neste documento são baseadas nestas versões de software e hardware:

- Dispositivo compatível com IOx configurado para IOx:
Endereço IP configurado Sistema operacional convidado (GOS) e Cisco Application Framework (CAF) que executam Network Address Translation (NAT) configurada para acesso ao CAF (porta 8443)
- Host Linux com SSL (Secure Sockets Layer) aberta instalado

- Arquivos de instalação do cliente IOx que podem ser baixados de:
<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.

Informações de Apoio

Desde a versão IOx, a assinatura do pacote de aplicativos AC5 é suportada. Esse recurso permite garantir que o pacote de aplicativos seja válido e que o instalado no dispositivo seja obtido de uma origem confiável. Se a validação de assinatura do pacote de aplicativos estiver ativada em uma plataforma, somente os aplicativos assinados poderão ser implantados.

Configurar

Estas etapas são necessárias para usar a validação da assinatura do pacote:

1. Crie uma chave e um certificado de autoridade de certificação (AC).
2. Gerar uma âncora de confiança para uso em IOx.
3. Importe a âncora de confiança em seu dispositivo IOx.
4. Crie uma chave específica do aplicativo e uma solicitação de assinatura de certificado (CSR).
5. Assine o certificado específico do aplicativo com o uso da CA.
6. Embale seu aplicativo IOx, assine-o com o certificado específico do aplicativo.
7. Implante o pacote IOx assinado em um dispositivo habilitado para assinatura.

Note: Para este artigo, uma CA autoassinada é usada em um cenário de produção. A melhor opção é usar uma CA oficial ou a AC da sua empresa para assinar.

Note: As opções para CA, chaves e assinaturas são escolhidas somente para fins de laboratório e podem precisar ser ajustadas para seu ambiente.

Etapa 1. Criar Chave CA e Certificado

A primeira etapa é criar sua própria CA. Isso pode ser feito simplesmente pela geração de uma chave para a CA e um certificado para essa chave:

Para gerar a chave 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)
```

Para gerar o certificado CA:

```
[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 []:
```

Os valores no certificado CA devem ser ajustados para corresponder ao seu caso de uso.

Etapa 2. Gerar âncora de confiança para uso em IOx

Agora que você tem a chave e o certificado necessários para sua CA, você pode criar um pacote de âncora de confiança para uso em seu dispositivo IOx. O pacote de âncora de confiança deve conter a cadeia de assinatura completa da AC (caso certificados intermediários sejam usados para assinatura) e um arquivo info.txt usado para fornecer os metadados (formulário livre).

Primeiro, crie o arquivo info.txt e coloque alguns metadados nele:

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

Opcionalmente, se você tiver vários certificados CA, para formar sua cadeia de certificados CA, será necessário reuni-los em um .pem:

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

Note: Esta etapa não é necessária para este artigo, uma vez que um único certificado raiz de CA é usado para iniciar a sessão, isso não é recomendado para produção e o par de chaves de CA raiz deve sempre ser armazenado offline.

A cadeia de certificados CA precisa ser denominada ca-chain.cert.pem, portanto, prepare este arquivo:

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

Finalmente, você pode combinar o ca-chain.cert.pem e info.txt em um tar com gzipped:

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

Etapa 3. Importar âncora de confiança no dispositivo IOx

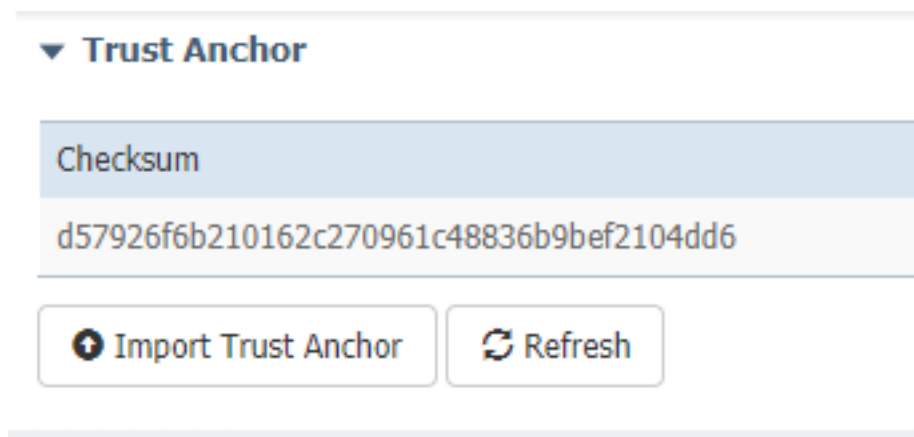
O trustanchorv1.tar.gz que você criou na etapa anterior precisa ser importado em seu dispositivo IOx. Os arquivos no pacote são usados para verificar se um aplicativo foi assinado com um certificado assinado pela CA correto antes de permitir uma instalação.

A importação da âncora confiável pode ser feita através do incidente:

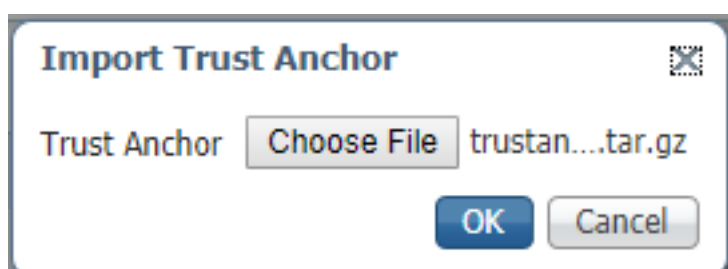
```
[jedepuyd@KJK-SRVIOT-10 signing]$ ioxclient platform signedpackages trustanchor set trustanchorv1.tar.gz
Currently active profile : default
Command Name: plt-sign-pkg-ta-set
Response from the server: Imported trust anchor file successfully
[jedepuyd@KJK-SRVIOT-10 signing]$ ioxclient platform signedpackages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

Outra opção é importar a âncora de confiança através do Gerenciador Local:

Navegue até **Configuração do sistema > Importar âncora de confiança** conforme mostrado na imagem.



Selecione o arquivo que você gerou na Etapa 2. e clique em **OK** conforme mostrado na imagem.




Depois de importar com êxito a âncora de confiança, marque **Enabled (Habilitado)** para **Application Signing Validation (Validação da assinatura do aplicativo)** e clique em **Save Configuration (Salvar configuração)** como mostrado na imagem:

▼ Application Signature Validation

▼ Configuration

Application Signature Validation

Enabled

 Save Configuration

Etapa 4. Crie uma chave específica de aplicativo e CSR

Em seguida, você pode criar um par de chaves e certificados que é usado para entrar em seu aplicativo IOx. A melhor prática é gerar um par de chaves específico para cada aplicativo que você planeja implantar.

Desde que cada um deles esteja assinado com a mesma AC, todos são considerados válidos.

Para gerar a chave específica do aplicativo:

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

Para gerar o 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 []:
```

Assim como com a CA, os valores no certificado do aplicativo devem ser ajustados para corresponder ao seu caso de uso.

Etapa 5. Assinar certificado específico de aplicativo com CA

Agora que você tem os requisitos para seu CA e CSR de aplicativo, você pode assinar o CSR

com o uso de CA. O resultado é um certificado específico de aplicativo assinado:

```
[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
```

Etapa 6. Embale seu aplicativo IOx e assine-o com um certificado específico do aplicativo

Neste ponto, você está pronto para empacotar seu aplicativo IOx e assiná-lo com o par de chaves gerado da Etapa 4. e assinado pela CA na Etapa 5.

O resto do processo para criar a origem e o pacote.yaml para seu aplicativo permanece inalterado.

aplicação IOx de pacote com o uso de par de chaves:

```
[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
```

Passo 7. Implante seu pacote IOx assinado em um dispositivo habilitado para assinatura

A última etapa no processo seria implantar o aplicativo em seu dispositivo IOx. Não há diferença em comparação a uma implantação de aplicativo não assinada:

```
[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
```

Verificar

Use esta seção para confirmar se a sua configuração funciona corretamente.

Para verificar se uma chave de aplicativo está assinada corretamente com sua CA, você pode fazer o seguinte:

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

Troubleshoot

Esta seção disponibiliza informações para a solução de problemas de configuração.

Quando você enfrenta problemas com a implantação de aplicativos, pode ver um destes erros:

```
[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"
}
```

Ocorreu um erro ao assinar o certificado de aplicação com o uso da AC ou não corresponde ao do pacote de âncora fidedigno.

Use as instruções mencionadas na seção Verificar para verificar seus certificados e também o pacote de âncora confiável.

Esse erro indica que o pacote não foi assinado corretamente. Você pode examinar a Etapa 6. novamente.

```
[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"  
}
```