Configurar a validação da assinatura do pacote de IOx

Índice

Introdução
Pré-requisitos
Requisitos
Componentes Utilizados
Informações de Apoio
Configurar
Etapa 1. Crie a chave e o certificado de CA
Etapa 2. Gerencia a âncora da confiança para o uso em IOx
Etapa 3. Âncora da confiança da importação no IOx-dispositivo
Etapa 4. Crie a chave característica da aplicação e o CSR
Etapa 5. Certificado característico da aplicação do sinal com CA
Etapa 6. Empacote seu aplicativo de IOx e assine-o com certificado característico da aplicação
Etapa 7. Distribua seu pacote assinado de IOx em um dispositivo Assinatura-permitido
Verificar
Troubleshooting

Introdução

Este documento descreve em uma maneira detalhada como criar e usar pacotes assinados na plataforma de IOx.

Pré-requisitos

Requisitos

A Cisco recomenda que você tenha conhecimento destes tópicos:

- Conhecimento básico de Linux
- Compreenda como os Certificados trabalham

Componentes Utilizados

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

- Dispositivo capaz de IOx que é configurado para IOx:
  Endereço IP de Um ou Mais Servidores Cisco ICM NT configuradoSistema operacional do convidado (GOS) e estrutura do aplicativo Cisco (CAF) essas corridasNetwork Address Translation (NAT) configurado para o acesso a CAF (porta 8443)
- Host de Linux com o secure sockets layer aberto (SSL) instalado

- Arquivos da instalação de cliente de IOx de que pode ser transferido: https://software.cisco.com/download/release.html?mdfid=286306005&softwareid=286306762

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a sua rede estiver ativa, certifique-se de que entende o impacto potencial de qualquer comando.

Informações de Apoio

Desde a liberação de IOx, a assinatura do pacote de aplicativo AC5 é apoiada. Esta característica reserva assegurar-se de que o pacote de aplicativo seja válido e esse instalado no dispositivo esteja obtido de um origem confiável. Se a validação da assinatura do pacote de aplicativo é girada SOBRE em uma plataforma, simplesmente os aplicativos então assinados podem ser distribuídos.

Configurar

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

1. Crie uma chave e um certificado do Certificate Authority (CA).
2. Gerencia uma âncora da confiança para o uso em IOx.
3. Importe a âncora da confiança em seu IOx-dispositivo.
4. Crie uma chave e uma solicitação de assinatura de certificado características da aplicação (CSR).
5. Assine o certificado característico da aplicação com o uso de CA.
6. Empacote seu aplicativo de IOx, assine-o com o certificado característico da aplicação.
7. Distribua seu pacote assinado de IOx em um dispositivo assinatura-permitido.

Note: Para este artigo, CA auto-assinado é usado em uma encenação da produção. A melhor opção é usar CA oficial ou CA da sua empresa para assinar.

Note: As opções para CA, as chaves e as assinaturas são escolhidas para finalidades do laboratório somente e puderam precisar de ser ajustado para seu ambiente.

Etapa 1. Crie a chave e o certificado de CA

A primeira etapa é criar seu próprio CA. Isto pode simplesmente ser feito pela geração de uma chave para CA e de um certificado para essa chave:

A fim gerar a chave de 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)
```

A fim gerar o certificado de 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 de CA devem ser ajustados para combinar seu caso do uso.

Etapa 2. Gerencia a âncora da confiança para o uso em IOx

Agora que você tem a chave e o certificado necessários para seu CA, você pode criar um pacote da âncora da confiança para o uso em seu dispositivo de IOx. O pacote da âncora da confiança deve conter CA completo que assina a corrente (caso que o certificado intermediário é usado assinando) e um arquivo de info.txt que seja usado para fornecer os metadata (de forma livre).

Primeiramente, crie o arquivo de info.txt e põe alguns metadata nele:

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

Opcionalmente, se você tem certificados de CA múltiplos, para formar sua corrente de certificado de CA, você precisa de uni-los em um .pem:

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

> **Note**: Esta etapa não é exigida para este artigo, desde que um único certificado de raiz de CA é usado para dirigir o sinal, isto não é recomendada para a produção e o keypair da CA raiz deve sempre ser armazenado off line.

A corrente de certificado de CA precisa de ser nomeada ca-chain.cert.pem, assim que 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 alcatrão gzipped:

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

Etapa 3. Âncora da confiança da importação no IOx-dispositivo

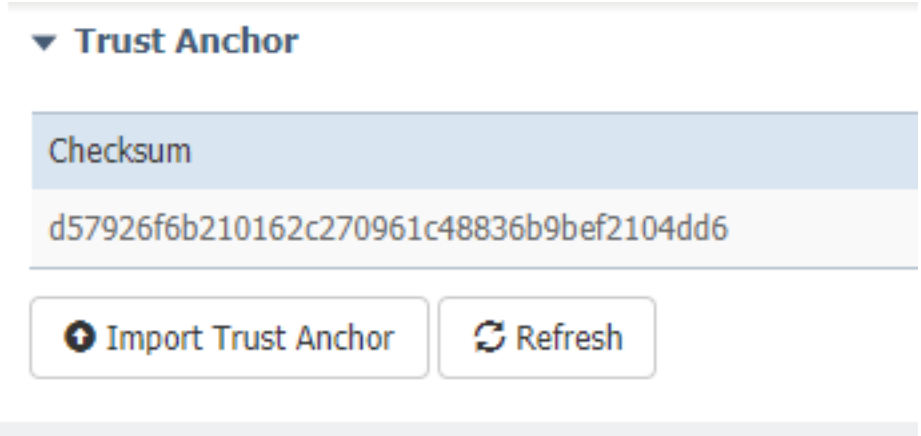
trustanchorv1.tar.gz que você criou na etapa precedente precisa de ser importado em seu IOx-dispositivo. Os arquivos no pacote são usados para verificar se um aplicativo obtido assinou com um certificado assinado CA de CA correto antes que permita uma instalação.

A importação da âncora da confiança pode ser feita através do ioxlcient:

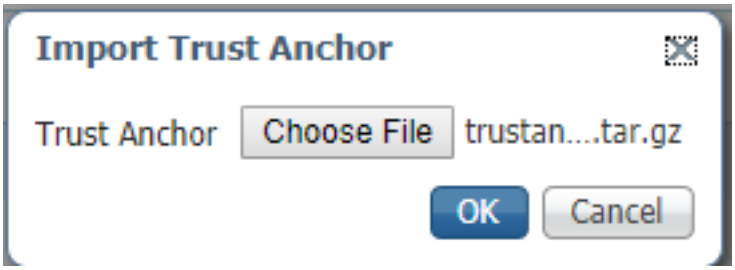
```
[jedepuyd@KJK-SRVIOT-10 signing]$ ioxclient platform signedpackages trustanchor set
trustanchorv1.tar.gz
Currently active profile :  default
Command Name:  plt-sign-pkg-ta-set
Response from the server:  Imported trust anchor file successfully
[jedepuyd@KJK-SRVIOT-10 signing]$ ioxclient platform signedpackages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

Uma outra opção é importar a âncora da confiança através do gerente local:

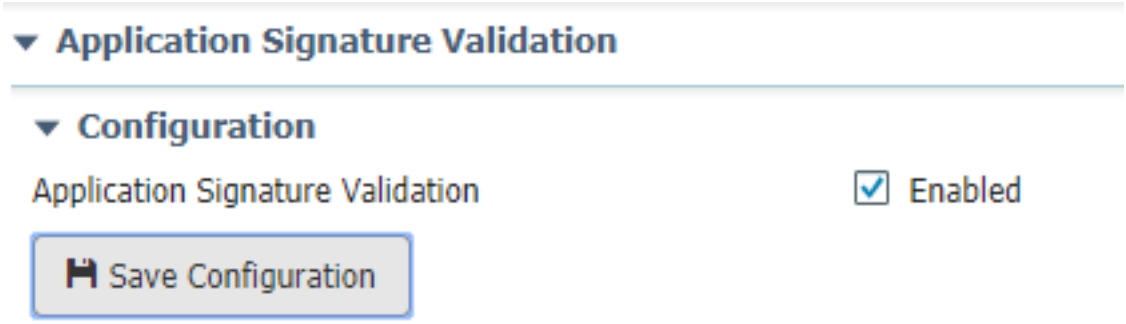
Navegue à **âncora da confiança da configuração de sistema > da importação** segundo as indicações da imagem.



Selecione o arquivo que você gerou em etapa 2. e clica a **APROVAÇÃO** segundo as indicações da imagem.



Depois que você importou com sucesso a âncora da confiança, a verificação **permitida** para a **validação de assinatura do aplicativo** e o clique **salvar a configuração** segundo as indicações da imagem:



Etapa 4. Crie a chave característica da aplicação e o CSR

Em seguida, você pode criar uma chave e um par do certificado que seja usado para assinar em seu aplicativo de IOx. O melhor prática é gerar um keypair específico para cada aplicativo que você planeia distribuir.

Enquanto cada um daquelas é assinada com mesmo CA, todos estão considerados como válido.

A fim gerar a chave característica da aplicação:

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

A fim 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 []:
```

Como com CA, os valores no certificado do aplicativo devem ser ajustados para combinar seu caso do uso.

Etapa 5. Certificado característico da aplicação do sinal com CA

Agora que você tem as exigências para seus CA e aplicativo CSR, você pode assinar o CSR com

o uso de CA. O resultado é um certificado característico da aplicação 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. Empacote seu aplicativo de IOx e assine-o com certificado característico da aplicação

Neste momento, você está pronto para empacotar seu aplicativo de IOx e para assiná-lo com o keypair gerado de etapa 4. e assinou por CA na etapa 5.

O resto do processo para criar a fonte e o package.yaml para seu aplicativo permanece inalterado.

empacote o aplicativo de IOx com o uso do keypair:

```
[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
```

Etapa 7. Distribua seu pacote assinado de IOx em um dispositivo Assinatura-permitido

A última etapa no processo seria distribuir o aplicativo a seu dispositivo de IOx. Não há nenhuma diferença em comparação com um desenvolvimento de aplicativo sem assinatura:

```
[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.

A fim verificar se uma chave do aplicativo é assinada corretamente com seu CA, você pode fazer este:

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

Troubleshooting

Esta seção fornece a informação que você pode se usar a fim pesquisar defeitos sua configuração.

Quando você experimenta edições com o desenvolvimento dos aplicativos, você poderia 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"
}
```

Algo foi mal em assinar o certificado do aplicativo com o uso de CA ou não combina com esse no pacote confiado da âncora.

Use as instruções mencionadas dentro verificam a seção, para verificar também seus Certificados e igualmente o pacote confiado da âncora.

Este o erro indica que seu pacote não esteve assinado corretamente, você pode olhar em etapa 6. outra vez.

```
[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"
}
```