

Configuración de la Validación de la Firma del Paquete IOx

Contenido

[Introducción](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Antecedentes](#)

[Configurar](#)

[Paso 1. Crear clave de CA y certificado](#)

[Paso 2. Generar anclaje de confianza para su uso en IOx](#)

[Paso 3. Importar Trust Anchor en IOx-Device](#)

[Paso 4. Crear clave específica de aplicación y CSR](#)

[Paso 5. Firmar certificado específico de la aplicación con CA](#)

[Paso 6. Empaquetar la aplicación IOx y firmarla con un certificado específico de la aplicación](#)

[Paso 7. Implemente el paquete IOx firmado en un dispositivo habilitado para firmas](#)

[Verificación](#)

[Troubleshoot](#)

Introducción

Este documento describe de manera detallada cómo crear y utilizar paquetes firmados en la plataforma IOx.

Prerequisites

Requirements

Cisco recomienda que tenga conocimiento sobre estos temas:

- Conocimiento básico de Linux
- Entender cómo funcionan los certificados

Componentes Utilizados

La información que contiene este documento se basa en las siguientes versiones de software y hardware.

- Dispositivo compatible con E/Sx configurado para E/Sx:
Dirección IP configurada Sistema operativo invitado (GOS) y Cisco Application Framework (CAF) que se ejecutan Traducción de direcciones de red (NAT) configurada para el acceso a CAF (puerto 8443)

- Host Linux con Secure Sockets Layer (SSL) abierto instalado
- Archivos de instalación del cliente IOx que se pueden descargar desde:
<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.

Antecedentes

Desde la versión IOx, se admite la firma de paquetes de aplicaciones AC5. Esta función permite asegurarse de que el paquete de la aplicación es válido y que el que está instalado en el dispositivo se obtiene de un origen de confianza. Si la validación de firma del paquete de aplicación está activada en una plataforma, sólo se pueden implementar las aplicaciones firmadas.

Configurar

Estos pasos son necesarios para utilizar la validación de la firma del paquete:

1. Cree una clave y un certificado de autoridad certificadora (CA).
2. Genere un anclaje de confianza para su uso en IOx.
3. Importe el ancla de confianza en su dispositivo IOx.
4. Cree una clave específica de la aplicación y una solicitud de firma de certificado (CSR).
5. Firme el certificado específico de la aplicación con el uso de la CA.
6. Empaquetar la aplicación IOx y firmarla con el certificado específico de la aplicación.
7. Implemente el paquete IOx firmado en un dispositivo con firma habilitada.

Nota: Para este artículo, se utiliza una CA autofirmada en un escenario de producción. La mejor opción es utilizar una CA oficial o la CA de su empresa para firmar.

Nota: Las opciones para la CA, las claves y las firmas se eligen únicamente con fines de laboratorio y es posible que deban ajustarse para su entorno.

Paso 1. Crear clave de CA y certificado

El primer paso es crear su propia CA. Esto se puede hacer simplemente generando una clave para la CA y un certificado para esa clave:

Para generar la clave 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 generar el 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 []:
```

Los valores del certificado de CA deben ajustarse para que coincidan con el caso práctico.

Paso 2. Generar anclaje de confianza para su uso en IOx

Ahora que tiene la clave y el certificado necesarios para su CA, puede crear un paquete de anclaje de confianza para su uso en su dispositivo IOx. El paquete de anclaje de confianza debe contener la cadena de firma completa de CA (en caso de que se utilice el certificado intermedio para la firma) y un archivo info.txt que se utiliza para proporcionar los metadatos (de forma libre).

Primero, cree el archivo info.txt y coloque algunos metadatos en él:

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

Opcionalmente, si tiene varios certificados de CA, para formar su cadena de certificados de CA, debe juntarlos en un .pem:

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

Nota: Este paso no es necesario para este artículo, ya que se utiliza un solo certificado raíz de CA para el signo directo, esto no se recomienda para la producción y el par de claves de CA raíz siempre se debe almacenar sin conexión.

La cadena de certificados de CA debe llamarse ca-chain.cert.pem, así que prepare este archivo:

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

Por último, puede combinar ca-chain.cert.pem e info.txt en un archivo .tar comprimido:

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

Paso 3. Importar Trust Anchor en IOx-Device

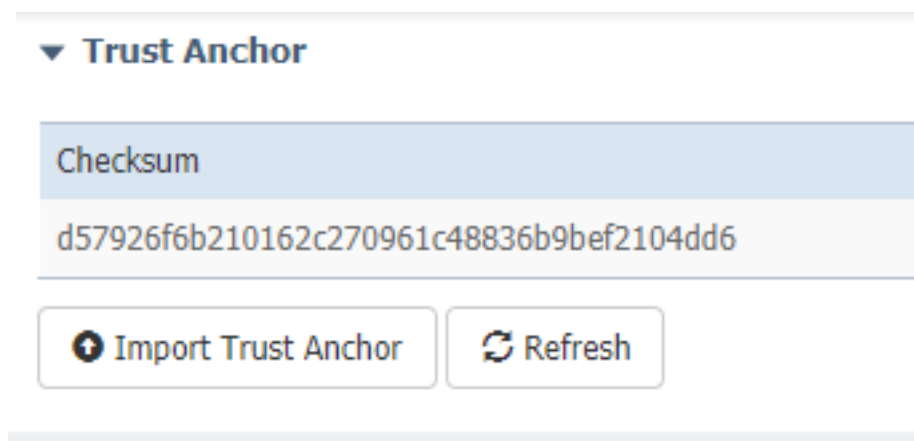
El trustanchorv1.tar.gz que creó en el paso anterior debe importarse en su dispositivo IOx. Los archivos del paquete se utilizan para verificar si una aplicación se firmó con un certificado firmado por CA de la CA correcta antes de permitir una instalación.

La importación del anclaje de confianza puede realizarse a través del cliente:

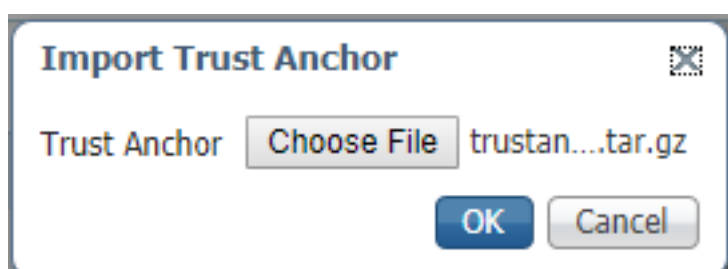
```
[jedepuyd@KJK-SRVIOT-10 signing]$ ioxclient platform signedpackages trustanchor set trustanchorv1.tar.gz
Currently active profile : default
Command Name: plt-sign-pkg-ta-set
Response from the server: Imported trust anchor file successfully
[jedepuyd@KJK-SRVIOT-10 signing]$ ioxclient platform signedpackages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

Otra opción es importar el anclaje de confianza a través del Administrador local:

Vaya a **Configuración del sistema > Importar anclaje de confianza** como se muestra en la imagen.



Seleccione el archivo que generó en el paso 2. y haga clic en **Aceptar** como se muestra en la imagen.




Después de haber importado correctamente el anclaje de confianza, verifique **Enabled for Application Signing Validation** y haga clic en **Save Configuration** como se muestra en la imagen:

▼ Application Signature Validation

▼ Configuration

Application Signature Validation

Enabled

 Save Configuration

Paso 4. Crear clave específica de aplicación y CSR

A continuación, puede crear un par de claves y certificados que se utiliza para iniciar sesión en la aplicación IOx. La mejor práctica es generar un par de claves específico para cada aplicación que planea implementar.

Siempre y cuando cada uno de ellos se firme con la misma CA, todos se considerarán válidos.

Para generar la clave específica de la aplicación:

```
[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 generar la RSE:

```
[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 []:
```

Al igual que con la CA, los valores del certificado de aplicación deben ajustarse para que coincidan con el caso práctico.

Paso 5. Firmar certificado específico de la aplicación con CA

Ahora que tiene los requisitos para su CA y CSR de aplicación, puede firmar el CSR con el uso de

CA. El resultado es un certificado específico de la aplicación firmado:

```
[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
```

Paso 6. Empaquetar la aplicación IOx y firmarla con un certificado específico de la aplicación

En este momento, está preparado para empaquetar su aplicación IOx y firmarla con el par de claves generado del paso 4. y firmado por la CA en el Paso 5.

El resto del proceso para crear el origen y package.yaml para la aplicación permanece inalterado.

aplicación IOx del paquete con el uso del par de llaves:

```
[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
```

Paso 7. Implemente el paquete IOx firmado en un dispositivo habilitado para firmas

El último paso del proceso sería implementar la aplicación en su dispositivo IOx. No hay diferencia en comparación con una implementación de aplicación sin firmar:

```
[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
```

Verificación

Utilice esta sección para confirmar que su configuración funcione correctamente.

Para verificar si una clave de aplicación está correctamente firmada con su CA, puede hacer lo siguiente:

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

Troubleshoot

En esta sección se brinda información que puede utilizar para resolver problemas en su configuración.

Cuando experimenta problemas con la implementación de aplicaciones, puede ver uno de estos errores:

```
[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 salió mal al firmar el certificado de aplicación con el uso de la CA o no coincide con el del paquete de anclaje de confianza.

Utilice las instrucciones mencionadas en la sección Verificación para comprobar los certificados y también el paquete de anclaje de confianza.

Este error indica que el paquete no se firmó correctamente; puede ver el Paso 6. de nuevo.

```
[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"  
}
```