Validación de la firma del paquete de IOx de la configuración

Contenido

Introducción

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

Prerequisites

Requisitos

Cisco recomienda que tenga conocimiento sobre estos temas:

- Conocimiento básico de Linux
- Entienda cómo los Certificados trabajan

Componentes Utilizados

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

- Dispositivo con capacidad de IOx que se configura para IOx:
  Dirección IP configuradaSistema operativo del invitado (GOS) y marco de la aplicación de Cisco (CAF) esos funcionamientosNetwork Address Translation (NAT) configurado para el acceso a CAF (puerto 8443)

- El host de Linux con Secure Sockets Layer (SSL) abierto instaló
- Archivos de la instalación del cliente de IOx de los cuales puede ser descargado: https://software.cisco.com/download/release.html?mdfid=286306005&softwareid=286306762

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si la red está funcionando, asegúrese de haber comprendido el impacto que puede tener cualquier comando.

Antecedentes

Desde la versión de IOx, AC5 la firma del paquete de la aplicación se soporta. Esta característica permite asegurarse de que el paquete de la aplicación es válido y el que está instalado en el dispositivo está obtenido de una fuente confiable. Si la validación de la firma del paquete de la aplicación se gira en una plataforma, sólo las aplicaciones entonces firmadas pueden ser desplegadas.

Configurar

Estos pasos se requieren para utilizar la validación de la firma del paquete:

1. Cree una clave y un certificado del Certificate Authority (CA).
2. Genere un ancla de la confianza para el uso en IOx.
3. Importe el ancla de la confianza en su IOx-dispositivo.
4. Cree una clave y un pedido de firma de certificado específicos a la aplicación (CSR).
5. Firme el certificado específico a la aplicación con el uso de CA.
6. Empaquete su aplicación de IOx, fírmela con el certificado específico a la aplicación.
7. Despliegue su paquete firmado de IOx sobre un dispositivo firma-habilitado.

Note: Para este artículo, CA uno mismo-firmado se utiliza en un escenario de la producción. La mejor opción es utilizar CA oficial o CA de su compañía para firmar.

Note: Las opciones para CA, las claves y las firmas se eligen para los propósitos del laboratorio solamente y pudieron necesitar ser ajustado según su entorno.

Paso 1. Cree la clave y el certificado de CA

El primer paso es crear su propio CA. Esto se puede hacer simplemente por la generación de una clave para CA y de un certificado para esa clave:

Para generar la clave 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)
```

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 en el certificado de CA se deben ajustar para hacer juego su caso del uso.

Paso 2. Genere el ancla de la confianza para el uso en IOx

Ahora que usted tiene la clave y el certificado necesarios para su CA, usted puede crear a un conjunto del ancla de la confianza para el uso en su dispositivo de IOx. El conjunto del ancla de la confianza debe contener CA lleno que firma el encadenamiento (en caso de que el certificado intermedio se utiliza para firmar) y un archivo de info.txt que se utiliza para proporcionar los meta datos (de la manera libre).

Primero, cree el archivo de info.txt y ponga algunos meta datos en él:

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

Opcionalmente, si usted tiene Certificados de CA múltiples, formar su encadenamiento del certificado de CA, usted necesita ponerlos juntos en un .pem:

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

> **Note**: Este paso no se requiere para este artículo, puesto que un solo certificado raíz de CA se utiliza para dirigir la muestra, esto no se recomienda para la producción y raíz CA el keypair se debe salvar siempre off-liné.

El encadenamiento del certificado de CA necesita ser nombrado ca-chain.cert.pem, así que elabore este archivo:

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

Finalmente, usted puede combinar el ca-chain.cert.pem y info.txt en un alquitrán gzipped:

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

Paso 3. Ancla de la confianza de la importación en el IOx-dispositivo

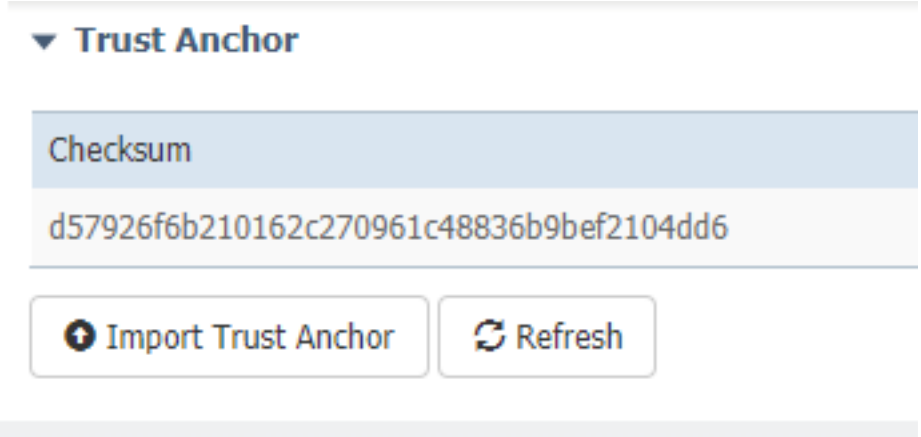
trustanchorv1.tar.gz que usted creó en el paso anterior necesita ser importado sobre su IOx-dispositivo. Los archivos en el conjunto se utilizan para verificar si una aplicación conseguida firmó con a certificado firmado por CA de CA correcto antes de que permita una instalación.

La importación del ancla de la confianza se puede hacer vía el ioxlcient:

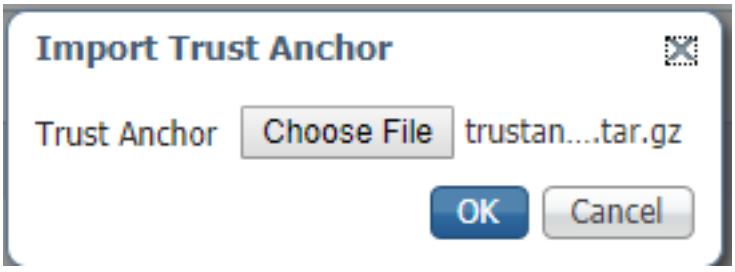
```
[jedepuyd@KJK-SRVIOT-10 signing]$ ioxclient platform signedpackages trustanchor set
trustanchorv1.tar.gz
Currently active profile :  default
Command Name:  plt-sign-pkg-ta-set
Response from the server:  Imported trust anchor file successfully
[jedepuyd@KJK-SRVIOT-10 signing]$ ioxclient platform signedpackages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

Otra opción es importar el ancla de la confianza vía el administrador local:

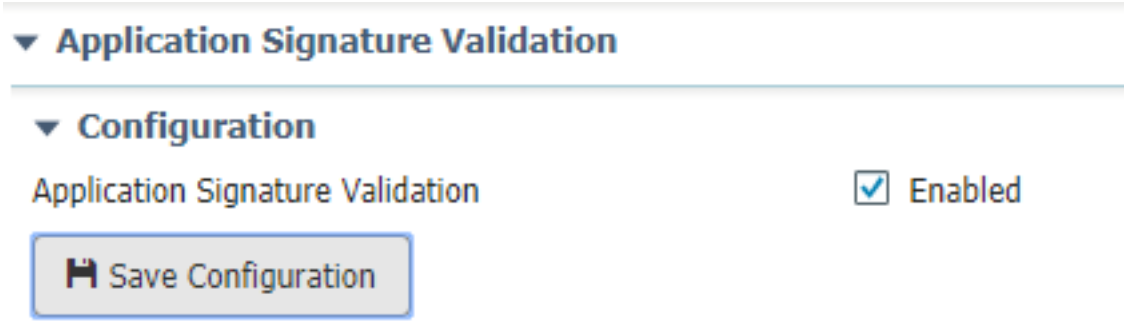
Navegue al **ancla de la confianza de los ajustes de sistema > de la importación** tal y como se muestra en de la imagen.



Seleccione el archivo que usted generó en el paso 2. y hace clic la **AUTORIZACIÓN** tal y como se muestra en de la imagen.



Después de que usted haya importado con éxito el ancla de la confianza, marque **habilitado** para la **validación de firma de la aplicación** y haga clic la **configuración de la salvaguardia** tal y como se muestra en de la imagen:



Paso 4. Cree la clave específica a la aplicación y el CSR

Después, usted puede crear una clave y un par del certificado que se utilice para firmar en su aplicación de IOx. La mejor práctica es generar un keypair específico para cada aplicación que usted planea desplegar.

Mientras cada uno de ésos se firme con mismo CA, todos se consideran como válido.

Para generar la clave específica a 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 el 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 con CA, los valores en el certificado de la aplicación se deben ajustar para hacer juego su caso del uso.

Paso 5. Certificado específico a la aplicación de la muestra con CA

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

el uso de CA. El resultado es un certificado específico a 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. Empaquete su aplicación de IOx y fírmela con el certificado específico a la aplicación

En este momento, usted está listo para empaquetar su aplicación de IOx y para firmarla con el keypair generado del paso 4. y firmó por CA en el paso 5.

Sigue habiendo el resto del proceso para crear la fuente y el package.yaml para su aplicación sin cambiar.

empaquete la aplicación de IOx con el uso del 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
```

Paso 7. Despliegue su paquete firmado de IOx sobre un dispositivo Firma-habilitado

El paso más reciente en el proceso sería desplegar la aplicación a su dispositivo de IOx. No hay diferencia con respecto a un despliegue de la aplicación sin signo:

```
[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

Utilize esta sección para confirmar que su configuración funcione correctamente.

Para verificar si una clave de la aplicación se firma correctamente con su CA, usted puede hacer esto:

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

Troubleshooting

Esta sección proporciona la información que usted puede utilizar para resolver problemas su configuración.

Cuando usted experimenta los problemas con el despliegue de las aplicaciones, usted podría 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 en la firma del certificado de la aplicación con el uso de CA o no hace juego con el que está en el conjunto de confianza del ancla.

Utilice las instrucciones mencionadas adentro verifican la sección, para marcar sus Certificados y también al conjunto de confianza del ancla también.

Este el error indica que su paquete no fue firmado correctamente, usted puede mirar en el paso 6. otra 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"
}
```