Настройте проверку подписи пакета IOx

Содержание

Введение

Этот документ описывает подробным способом, как создать и использовать подписанные пакеты на платформе IOx.

Предварительные условия

Требования

Компания Cisco рекомендует предварительно ознакомиться со следующими предметами:

- Основное знание Linux
- Поймите, как работают сертификаты

Используемые компоненты

Сведения, содержащиеся в данном документе, касаются следующих версий программного обеспечения и оборудования:

- Устройство с поддержкой IOx, которое настроено для IOx:
  IP-адрес настроенГостевая операционная система (GOS) и Платформа приложения Cisco (CAF), которая выполняетсяТехнология NAT настроила для доступа к CAF (порт 8443)
- Хост Linux с открытым установленным Уровнем защищенных сокетов (SSL)
- Файлы установки клиентской части IOx, которые могут быть загружены

от: https://программное обеспечение. cisco . com/download/release.html?mdfid=286306005&softwareid=286306762

Сведения, представленные в этом документе, были получены от устройств, работающих в специальной лабораторной среде. Все устройства, описанные в этом документе, были запущены с чистой (стандартной) конфигурацией. В рабочей сети необходимо изучить потенциальное воздействие всех команд до их использования.

Общие сведения

Начиная с выпуска IOx поддерживается подписание пакета приложений AC5. Эта функция позволяет гарантировать, что пакет приложений допустим, и тот, установленный на устройстве, получен из надежного источника. Если Проверка Подписи Пакета приложений включена в платформе, только тогда приложения со знаком могут быть развернуты.

Настройка

Эти шаги требуются, чтобы использовать проверку подписи пакета:

1. Создайте ключ Центра сертификации (CA) и сертификат.
2. Генерируйте трастовую привязку для использования на IOx.
3. Импортируйте трастовую привязку на своем IOx-устройстве.
4. Создайте специализированный ключ и Запрос подписи сертификата (CSR).
5. Подпишите специализированный сертификат с использованием CA.
6. Упакуйте свое приложение IOx, подпишите его со специализированным сертификатом.
7. Разверните свой пакет IOx со знаком на поддерживающее подпись устройство.

Примечание: Для этой статьи самоподписанный CA используется в производственном сценарии. Наилучший вариант состоит в том, чтобы использовать официальный CA или CA вашей компании для подписания.

Примечание: Опции для CA, ключей и подписей выбраны в целях лабораторной работы только и, возможно, должны были бы быть отрегулированы для вашей среды.

Шаг 1. Создайте ключ CA и сертификат

Первый шаг должен создать вашего собственного CA., Это может быть просто сделано генерацией ключа для CA и сертификата для того ключа:

Для генерации ключа 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)
```

Для генерации сертификата 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 []:
```

Значения в сертификате CA должны быть отрегулированы для соответствия с вариантом использования.

Шаг 2. Генерируйте трастовую привязку для использования на IOx

Теперь, когда у вас есть необходимый ключ и сертификат для вашего CA, можно создать трастовую связку (bundle) привязки для использования на устройстве IOx. Трастовая связка (bundle) привязки должна содержать полную цепочку подписания CA (в случае, если промежуточный сертификат используется для подписания), и info.txt файл, который используется для обеспечения (свободная форма) метаданных.

Во-первых, создайте info.txt файл и поместите некоторые метаданные в него:

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

Дополнительно, если у вас есть множественные сертификаты CA, для формирования цепочки сертификата CA, необходимо соединить их в одном .pem:

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

> **Примечание:** Этот шаг не требуется для этой статьи, так как одиночный корневой сертификат CA используется к прямому знаку, это не рекомендуется для производства, и пара ключей узла CA должна всегда быть сохранена оффлайн.

Цепочку сертификата CA нужно назвать ca-chain.cert.pem, поэтому подготовить этот файл:

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

Наконец, можно объединить ca-chain.cert.pem и info.txt в gzipped tar:

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

Шаг 3. Привязка к доверию импорта на IOx-устройстве

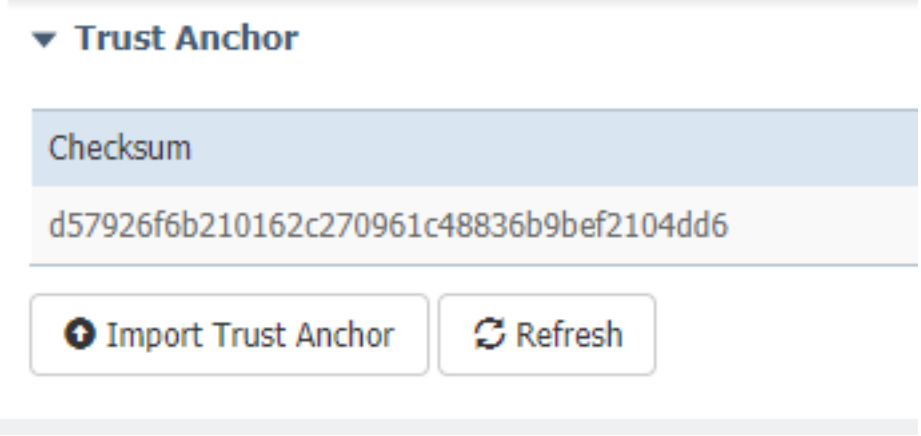
trustanchorv1.tar.gz, который вы создали в предыдущем шаге, должен быть импортирован на ваше IOx-устройство. Файлы в связке (bundle) используются, чтобы проверить, было ли приложение подписано с Сертификатом подписанный ЦС от корректного CA, прежде чем это позволит установку.

Импорт трастовой привязки может быть сделан через ioxlcient:

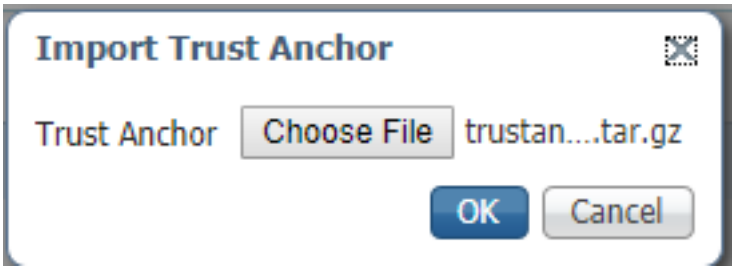
```
[jedepuyd@KJK-SRVIOT-10 signing]$ ioxclient platform signedpackages trustanchor set
trustanchorv1.tar.gz
Currently active profile :  default
Command Name:  plt-sign-pkg-ta-set
Response from the server:  Imported trust anchor file successfully
[jedepuyd@KJK-SRVIOT-10 signing]$ ioxclient platform signedpackages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

Другая опция должна импортировать трастовую привязку через Локального Менеджера:

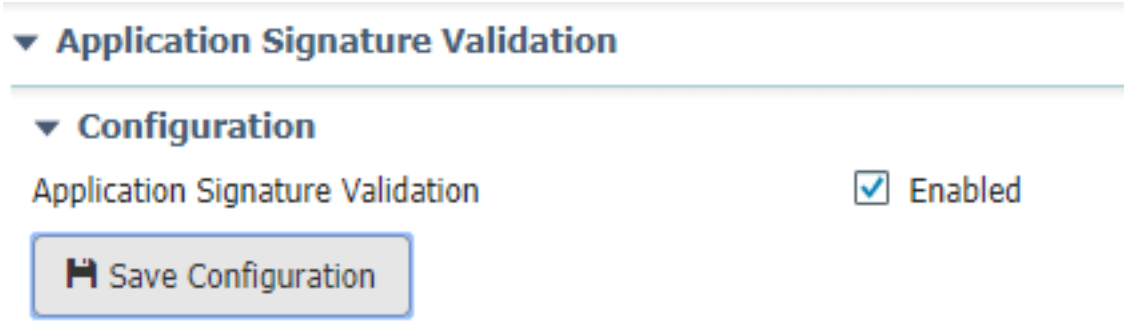
Перейдите к **Системному параметру> Привязка к Доверию Импорта** как показано в образе.



Выберите файл, который вы генерировали в Шаге 2. и нажмите **OK** как показано в образе.



После успешного импорта трастовой привязки проверьте**, Включил** для **Приложения, Подписав Проверку** , и нажмите **Save Configuration** как показано в образе:



Шаг 4. . Создайте специализированный ключ и CSR

Затем, можно создать ключ и пару сертификата, которая используется для подписания в приложение IOx. Оптимальный метод должен генерировать одну определенную пару ключей для каждого приложения, которое вы планируете развернуть.

Пока каждый из тех подписан с тем же CA, их все рассматривают как допустимых.

Для генерации специализированного ключа:

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

Для генерации 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 []:
```

Как с CA, значения в сертификате приложения должны быть отрегулированы для соответствия с вариантом использования.

Шаг 5. . Подпишите специализированный сертификат с CA

Теперь, когда у вас есть требования для вашего CA и CSR приложения, можно подписать

CSR с использованием CA., результатом является специализированный сертификат со знаком:

```
[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
```

Шаг 6. Упакуйте свое Приложение IOx и Знак это со Специализированным Сертификатом

На этом этапе вы готовы упаковать свое приложение IOx и подписать его с генерируемой парой ключей от Шага 4. и подписанный CA в Шаге 5.

Остаток процесса для создания источника и package.yaml для приложения остается неизменным.

пакет приложение IOx с использованием пары ключей:

```
[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
```

Шаг 7. Разверните свой Пакет IOx Со знаком на поддерживающее Подпись Устройство

Последний шаг в процесс должен был бы развернуть приложение на вашем устройстве IOx. Нет никакого различия по сравнению с установкой приложений без знака:

```
[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
```

Проверка

Воспользуйтесь данным разделом для проверки правильности функционирования вашей конфигурации.

Чтобы проверить, подписан ли ключ приложения правильно с вашим CA, можно сделать это:

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

Устранение неполадок

Этот раздел обеспечивает информацию, которую вы можете использовать для того, чтобы устранить неисправность в вашей конфигурации.

При испытании проблем с развертываниями приложений вы видели одну из этих ошибок:

```
[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"
}
```

Что-то пошло не так, как надо в подписании сертификата приложения с использованием CA, или это не совпадает с тем в доверяемой связке (bundle) привязки.

Используйте инструкции, упомянутые в, Проверяют раздел, для проверки сертификатов и также доверяемой связки (bundle) привязки также.

Они ошибка указывает, что ваш пакет не был подписан правильно, можно изучить Шаг 6. снова.

```
[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"
}
```