تكوين التحقق من صحة توقيع حزمة IOx

المحتويات

المقدمة

يوضح هذا المستند بشكل مفصل كيفية إنشاء الحزم الموقعة واستخدامها على منصة IOx.

المتطلبات الأساسية

المتطلبات

توصي Cisco بأن تكون لديك معرفة بالمواضيع التالية:

- معرفة لينوكس الأساسية
- فهم كيفية عمل الشهادات

المكونات المستخدمة

تستند المعلومات الواردة في هذا المستند إلى إصدارات البرامج والمكونات المادية التالية:

- جهاز قادر على تشغيل IOx تم تكوينه ل IOx:
  تم تكوين عنوان IPظام التشغيل المضيف (GOS) وإطار عمل تطبيقات Cisco (CAF)
  الذي يتم تشغيلهشكلت شبكة عنوان ترجمة (NAT) ل منفذ إلى CAF (ميناء 8443)
- مضيف Linux مع ثبتت طبقة مآخذ التوصيل الآمنة (SSL) المفتوحة
- ملفات تثبيت عميل IOx التي يمكن تنزيلها من:
  https://software.cisco.com/download/release.html?mdfid=286306005&softwareid=286306762

تم إنشاء المعلومات الواردة في هذا المستند من الأجهزة الموجودة في بيئة معملية خاصة.

بدأت جميع الأجهزة المُستخدمة في هذا المستند بتكوين ممسوح (افتراضي). إذا كانت شبكتك مباشرة، فتأكد من فهمك للتأثير المحتمل لأي أمر.

معلومات أساسية

منذ إصدار IOx، يتم دعم توقيع حزمة تطبيق AC5. تتيح هذه الميزة التأكد من أن حزمة التطبيق صالحة وأن حزمة التطبيق المثبتة على الجهاز تم الحصول عليها من مصدر موثوق. في حالة تشغيل التحقق من صحة توقيع حزمة التطبيق في نظام أساسي، يمكن نشر التطبيقات الموقعة فقط.

التكوين

هذه الخطوات مطلوبة لاستخدام التحقق من صحة توقيع الحزمة:

1. إنشاء مفاتيح المرجع المصدق (CA) والشهادة.
2. إنشاء رابط ثقة للاستخدام على IOx.
3. قم باستيراد نقطة إرساء الثقة على جهاز IOx الخاص بك.
4. قم بإنشاء مفاتيح خاص بالتطبيق وطلب توقيع الشهادة (CSR).
5. توقيع شهادة التطبيق المحددة باستخدام المرجع المصدق.
6. قم بحزم تطبيق IOx وتوقيعه باستخدام شهادة التطبيق المحددة.
7. قم بنشر حزمة IOx الموقعة على جهاز يدعم التوقيع.

ملاحظة: بالنسبة لهذه المادة، يستخدم المرجع المصدق الموقع ذاتيا في سيناريو إنتاج. الخيار الأفضل هو إستخدام مرجع مصدق رسمي أو مرجع مصدق لشركتك للتوقيع.

ملاحظة: يتم إختيار خيارات المرجع المصدق والمفاتيح والتواقيع لأغراض معملية فقط وقد يلزم تعديلها لتناسب بيئتك.

الخطوة 1. إنشاء مفاتيح CA والشهادة

الخطوة الأولى هي أن تنشئ المرجع المصدق الخاص بك. ويمكن القيام بذلك ببساطة عن طريق إنشاء مفاتيح للمرجع المصدق وشهادة لذلك المفاتيح:

لإنشاء مفاتيح 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)
```

لإنشاء شهادة المرجع المصدق:

```
[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 []:
```

يجب تعديل القيم الموجودة في شهادة المرجع المصدق لمطابقة حالة الاستخدام الخاصة بك.

الخطوة 2. إنشاء رابط الثقة للاستخدام على IOx

الآن بعد أن أصبح لديك الملفات والشهادة اللازمين لـ CA الخاص بك، يمكنك إنشاء حزمة إرساء ثقة للاستخدام على جهاز IOx الخاص بك. يجب أن تحتوي حزمة إرساء الثقة على ملف info.txt (للتوقيع) وسلسلة توقيع CA الكاملة (في حالة إستخدام الشهادة الوسيطة للتوقيع) الذي يتم إستخدامه لتوفير بيانات التعريف (النموذج الخر).

أولا، قم بإنشاء ملف info.txt ووضع بعض البيانات الأولية فيه:

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

إذا كان لديك عدة شهادات CA، لتشكيل سلسلة شهادات CA الخاصة بك، تحتاج إلى دمج محتوياتها في ملف .pem واحد:

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

> **ملاحظة**: هذه الخطوة غير مطلوبة لهذه المقالة، انظر استخدام شهادة جذر CA واحدة للتوقيع المباشر، فلا يوصى بالإنتاج ويجب تخزين مفاتيح CA الجذر دائما دون اتصال.

يجب تسمية سلسلة شهادات CA باسم ca-series.cert.pem، لذلك قم بإعداد هذا الملف:

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

أخيرا، يمكنك دمج ca-series.cert.pem و info.txt في قطران مضغوط:

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

الخطوة 3. إستيراد إرساء الثقة على IOx-Device

يلزم إستيراد Trustanchorv1.tar.gz الذي تم إنشاؤه في الخطوة السابقة إلى جهاز IOx الخاص بك. تستخدم الملفات الموجودة في الحزمة للتحقق مما إذا تم توقيع تطبيق

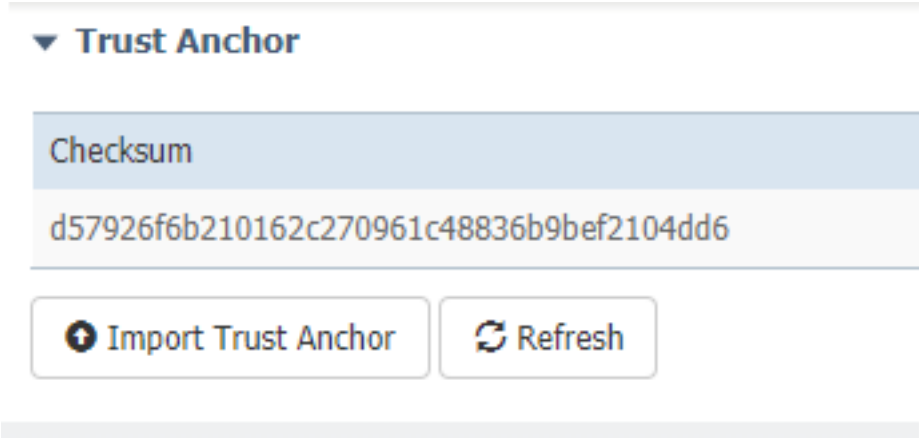
باستخدام شهادة موقعة من المرجع المصدق الصحيح قبل أن تسمح بالتثبيت.

يمكن تنفيذ إستيراد مرساة الثقة من خلال عنصر التحكم:

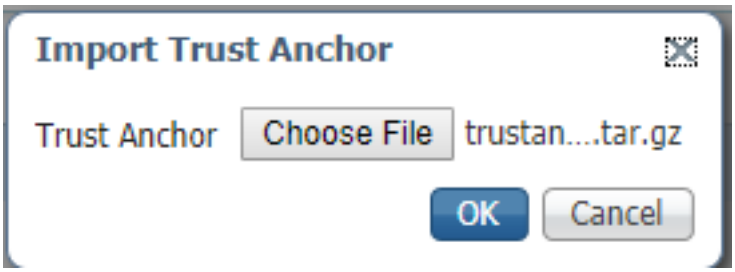
```
[jedepuyd@KJK-SRVIOT-10 signing]$ ioxclient platform signedpackages trustanchor set
trustanchorv1.tar.gz
Currently active profile :  default
Command Name:  plt-sign-pkg-ta-set
Response from the server:  Imported trust anchor file successfully
[jedepuyd@KJK-SRVIOT-10 signing]$ ioxclient platform signedpackages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

خيار آخر هو أن تقوم باستيراد رابط الثقة عبر المدير المحلي:

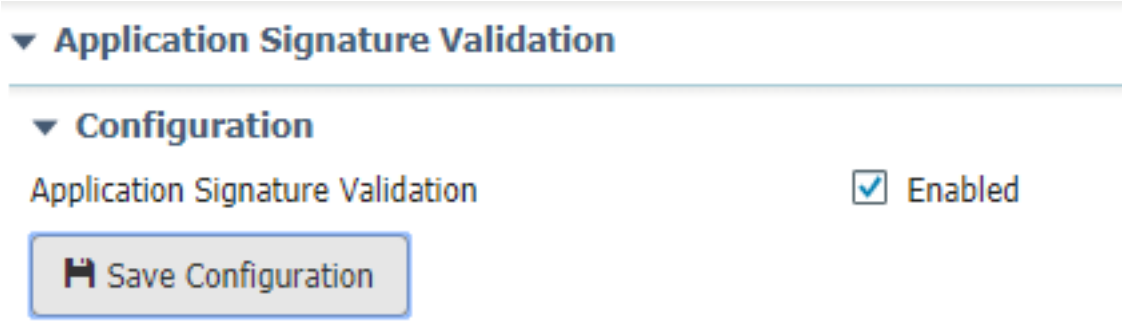
انتقل إلى **إعداد النظام > إستيراد رابط الثقة** كما هو موضح في الصورة.



حدد الملف الذي قمت بتوليده في الخطوة 2. وانقر **موافق** كما هو موضح في الصورة.



بعد أن تقوم باستيراد إرساء الثقة بنجاح، تحقق من **تمكين التحقق من توقيع التطبيق** وانقر فوق **حفظ التكوين** كما هو موضح في الصورة:



الخطوة 4. إنشاء مفتاح خاص بالتطبيق و CSR

بعد ذلك، يمكنك إنشاء زوج مفاتيح وشهادات يتم إستخدامه لتسجيل الدخول إلى تطبيق نشره. أفضل ممارسة هي إنشاء زوج مفاتيح محدد لكل تطبيق تخطط لنشره. IOx الخاص بك.

طالما أن كل واحد من هذه الوثائق قد وقع بنفس المرجع المصدق، فإنها تعتبر جميعا صحيحة.

لإنشاء المفتاح الخاص بالتطبيق:

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

لإنشاء 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 []:
```

تأكد من أن الحالل مع المرجع المصدق، يجب تعديل القيم الموجودة في شهادة التطبيق لتطابق تلك الخاصة بحالة الاستخدام الخاصة بك.

الخطوة 5. توقيع شهادة خاصة بالتطبيق مع CA

الآن بعد أن أصبحت لديك متطلبات CA والتطبيق CSR، يمكنك توقيع CSR باستخدام CA. والنتيجة هي شهادة موقعة خاصة بالتطبيق:

```
[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
```

الخطوة 6. قم بتجميع تطبيق IOx وتوقيعه بشهادة خاصة بالتطبيق

عند هذه النقطة، تكون مستعدا لحزم تطبيق IOx وتوقيعه باستخدام زوج المفاتيح الذي

تم إنشاؤه من الخطوة 4. ومو قبل CA في الخطوة 5.

باقي العملية إنشاء المصدر والحزمة.yaml للتطبيق الخاص بك الا تزال دون تغيير.

تطبيق IOx مع إستخدام ج ز المفاتيح:

```
[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
```

الخطوة 7. نشر حزمة IOx الموقعة على جهاز يدعم التوقيع

تتمثل الخطوة الأخيرة في العملية في نشر التطبيق على جهاز IOx لديك. الفرق الوحيد مقارنة بنشر تطبيق غير موقع:

```
[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
```

التحقق من الصحة

استخدم هذا القسم لتأكيد عمل التكوين بشكل صحيح.

للتحقق من توقيع مفاتيح التطبيق بشكل صحيح مع المرجع المصدق، يمكنك القيام بذلك:

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

استكشاف الأخطاء وإصلاحها

يوفر هذا القسم معلومات يمكن إستخدامها لاستكشاف أخطاء التكوين وإصلاحها.

عندما تواجه مشكلات تتعلق بنشر التطبيقات، يمكن مشاهدة أحد هذه الأخطاء:

```
[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"
}
```

حدث خطأ ما في توقيع شهادة التطبيق باستخدام CA أو أنها لا تطابق شهادة المستخدم الموثوق بها في حزمة الإرساء.

أستخدم التعليمات المذكورة في قسم التحقق، للتحقق من ترخيصاتك وأيضا حزمة الربط الموثوق بها.

يشير هذا الخطأ إلى أنه لم يتم توقيع الحزمة بشكل صحيح، يمكنك النظر في الخطوة 6. مرة أخرى.

```
[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"
}
```

حول هذه الترجمة
ترجمت Cisco هذا المستند باستخدام مجموعة من التقنيات الآلية والبشرية لتقديم محتوى دعم للمستخدمين في جميع أنحاء العالم بلغتهم الخاصة. يُرجى ملاحظة أن أفضل ترجمة آلية لن تكون دقيقة كما هو الحال مع الترجمة الاحترافية التي يقدمها مترجم محترف. تخلي Cisco Systems مسئوليتها عن دقة هذه الترجمات وتُوصي بالرجوع دائمًا إلى المستند الإنجليزي الأصلي (الرابط متوفر).