OVA Images and Upgrade Scripts Verification

Introduction

Starting from Cisco IoT FND 4.9.0, you can verify the integrity of the OVA images and upgrade scripts before the installation or upgrade of IoT FND.

For more information, refer to:

- Verifying the OVA Signature, on page 2
- Verifying the Upgrade-Scripts RPM Signature, on page 3
- Verifying the CGMS Tools RPM for Postgres Signature, on page 4

Note From FND release 4.12 onwards, the Secure Hash Algorithm is SHA256 and the earlier FND releases use SHA1.

Table 1: OVA Images and Upgrade Scripts Zip File Contents

Zip File Contents	Description
CISCO-IOTFND-V-K9-<release>-<build number>.zip	Includes Oracle for Mesh management (CGR, IR5xx) use case.
1. iot-fnd-oracle-<release>-<build number>_SHA1_signed.ova **2.** iot-tps-<release>-<build number>_SHA1_signed.ova	
CISCO-IOTFND-VPI-K9-<release>-<build number>.zip	Includes Postgres / Influx for gateway management (IR8xx, IR1101, IC3K) use case.

<table>
<tr><th>Zip File Contents</th><th>Description</th></tr>
<tr><td colspan="2">1. iot-fnd-<release>-<build number>_SHA256_signed.ova
2. iot-tps-<release>-<build number>_SHA256_signed.ova</td></tr>
<tr><td>CISCO-IOTFND-VPI-K9-CGMS-TOOLS-<release>-<build number>.zip
Attention The CGMS tools file is bundled with `CISCO-IOTFND-VPI-K9-UPGRADE-SCRIPTS-<release>-<build number>.zip`.</td><td>Includes cgms tools rpm for Postgres deployments.</td></tr>
<tr><td colspan="2">1. cgms-tools-<release>-<build number>.x86_64.rpm
2. FND_RPM_SIGN-CCO_RELEASE.pem — Cisco signed x.509 end-entity certificate containing public key that is used to verify the signature. This certificate is chained to Cisco root CA and sub CA posted on https://www.cisco.com/security/pki/.
3. cisco_openpgp_verify_release.py — Signature verification program for verifying the Open-pgp Complaint Public Key against x.509 end-entity certificate.
4. cisco_openpgp_verify_release.py.signature — Signature generated for the script cisco_openpgp_verify_release.py.
5. FND-rel-binary.gpg — Open-pgp public key is used for verification of signed RPM.
6. FND-rel-ascii.gpg — Open-pgp public key is used for verification of signed RPM.</td></tr>
<tr><td>CISCO-IOTFND-VPI-K9-UPGRADE-SCRIPTS-<release>-<build number>.zip</td><td>Includes upgrade scripts for upgrading FND-Postgres / Influx OVA.</td></tr>
<tr><td colspan="2">1. upgrade-ova-<release>-<build number>.rpm — Signature embedded RPM image.
2. FND_RPM_SIGN-CCO_RELEASE.pem — Cisco signed x.509 end-entity certificate containing public key that is used to verify the signature. This certificate is chained to Cisco root CA and sub CA posted on https://www.cisco.com/security/pki/.
3. cisco_openpgp_verify_release.py — Signature verification program for verifying the Open-pgp Complaint Public Key against x.509 end-entity certificate.
4. cisco_openpgp_verify_release.py.signature — Signature generated for the script cisco_openpgp_verify_release.py.
5. FND-rel-binary.gpg — Open-pgp public key is used for verification of signed RPM.
6. FND-rel-ascii.gpg — Open-pgp public key is used for verification of signed RPM.</td></tr>
</table>

Verifying the OVA Signature

To verify the OVA signature:

Step 1 Install the `ovftool`.

Step 2 Run the command to verify the signed `ova` file.

```
ovftool iot-fnd-<release>-<build number>_SHA256_signed.ova
```

Verifying the Upgrade-Scripts RPM Signature

Prerequisites:

- Python 2.7.x
- OpenSSL
- Verification scripts running on customer-premises need internet connection to reach Cisco to download root and sub-CA certs

To verify the upgrade-scripts RPM signature:

Step 1 Unzip the file `CISCO-IOTFND-VPI-K9-UPGRADE-SCRIPTS-<release>-<build number>.zip`.

Step 2 Change directory (cd) to `CISCO-IOTFND-VPI-K9-UPGRADE-SCRIPTS-<release>-<build number>` folder.

Step 3 Extract the public key from the public cert:

```
openssl x509 -pubkey -noout -in FND_RPM_SIGN-CCO_RELEASE.pem > FND-EE-cert.pubkey
```

Expected Result:

```
FND-EE-cert.pubkey is created under the same folder
```

Step 4 Verify the verification script using the public key and the signature files.

```
openssl dgst -sha512 -verify FND-EE-cert.pubkey -signature
cisco_openpgp_verify_release.py.signature cisco_openpgp_verify_release.py
```

Expected Result:

```
Verified OK
```

Step 5 Verify if the delivered binary and ASCII keys have matching fingerprints.

a) `gpg FND-rel-binary.gpg`

Expected Result:

```
pub 2048R/F7D5ED29 2017-01-01 identity-name  (FND.rel) identity-name@cisco.com
```

b) `gpg FND-rel-ascii.gpg`

Expected Result:

```
pub 2048R/F7D5ED29 2017-01-01 identity-name (FND.rel) identity-name@cisco.com
```

Step 6 Verify the binary GPG key against EE cert.

```
./cisco_openpgp_verify_release.py -e FND_RPM_SIGN-CCO_RELEASE.pem -G
FND-rel-binary.gpg
```

Expected Result:

```
 Downloading CA certificate from http://www.cisco.com/security/pki/certs/crcam2.cer ...

Successfully downloaded crcam2.cer.

Downloading SubCA certificate from http://www.cisco.com/security/pki/certs/innerspace.cer ...

Successfully downloaded innerspace.cer.

Successfully verified Cisco root, subca and end-entity certificate chain.

Successfully fetched a public key from FND_RPM_SIGN-CCO_RELEASE.pem.

Successfully authenticated FND-rel-binary.gpg key using Cisco X.509 certificate trust chain.
```

Step 7 Verify the RPM Signature using the GPG ASCII key.

```
sudo rpm --import FND-rel-ascii.gpg
rpm -K upgrade-ova-<release>-<build number>.rpm
```

Expected Result:

```
upgrade-ova-<release>-<build number>.rpm: rsa sha1 (md5) pgp md5 OK
```

Step 8 Once the RPM is verified, you can upgrade OVA using the RPM.

Verifying the CGMS Tools RPM for Postgres Signature

Prerequisites:

- Python 2.7.x
- OpenSSL
- Verification scripts running on customer-premises need an internet connection to reach Cisco to download root and sub-CA certs

To verify the cgms tools rpm for Postgres signature:

Step 1 Unzip the file `CISCO-IOTFND-VPI-K9-CGMS-TOOLS-<release>-<build number>.zip .`

Step 2 Change directory (cd) to `CISCO-IOTFND-VPI-K9-CGMS-TOOLS-<release>-<build number>.zip` folder.

Step 3 Extract the public key from the public cert:

```
openssl x509 -pubkey -noout -in FND_RPM_SIGN-CCO_RELEASE.pem > FND-EE-cert.pubkey
```

Expected Result:

```
FND-EE-cert.pubkey is created under the same folder
```

Step 4 Verify the verification script using the public key and the signature files.

```
openssl dgst -sha512 -verify FND-EE-cert.pubkey -signature
cisco_openpgp_verify_release.py.signature cisco_openpgp_verify_release.py
```

Expected Result:

```
Verified OK
```

Step 5 Verify if the delivered binary and ASCII keys have matching fingerprints.

a) `gpg FND-rel-binary.gpg`

Expected Result:

```
pub 2048R/F7D5ED29 2017-01-01 identity-name  (FND.rel) identity-name@cisco.com
```

b) `gpg FND-rel-ascii.gpg`

Expected Result:

```
pub 2048R/F7D5ED29 2017-01-01 identity-name (FND.rel) identity-name@cisco.com
```

Step 6 Verify the binary GPG key against EE cert.

```
./cisco_openpgp_verify_release.py -e FND_RPM_SIGN-CCO_RELEASE.pem -G
FND-rel-binary.gpg
```

Expected Result:

```
 Downloading CA certificate from http://www.cisco.com/security/pki/certs/crcam2.cer ...

Successfully downloaded crcam2.cer.

Downloading SubCA certificate from http://www.cisco.com/security/pki/certs/innerspace.cer ...

Successfully downloaded innerspace.cer.

Successfully verified Cisco root, subca and end-entity certificate chain.

Successfully fetched a public key from FND_RPM_SIGN-CCO_RELEASE.pem.

Successfully authenticated FND-rel-binary.gpg key using Cisco X.509 certificate trust chain.
```

Step 7 Verify the RPM Signature using the GPG ASCII key.

```
sudo rpm --import FND-rel-ascii.gpg

rpm -K cgms-tools-<release>-<build number>.x86_64.rpm
```

Expected Result:

```
upgrade-cgms-tools-<release>-<build number>.x86_64.rpm: rsa sha1 (md5) pgp md5 OK
```

Step 8 Once the RPM is verified, you can upgrade cgms-tools using the RPM.