Create and Manage Certificates

For supported software information, click here.

On a Director node, you import, create and manage certificates for the web server. The certificates are used for GUI operations, API calls, and communication between Director and Analytics nodes.

When you first install the Director software, it includes only the basic self-signed certificates, which are sufficient for the initial configuration of the system. You can use those certificates for secure data transfer between the Director server and clients using secure socket layer (SSL) encryption.

Certificates aid in authenticating Director nodes and clients before they connect with each other. They also contain keys to establish encrypted and secure connections.

For production environments, you can import an existing private key and a certificate signed by an external Certificate Authority (CA). For proof-of-concept (POC) and demonstration purposes, you can use either a real certificate or a self-signed certificate that is generated by the tools included with the Director software.

Director nodes support X.509 certificates in the following formats:

- DER-encoded certificate—DER encoding is the binary form of the certificate. This format supports the encoding of private keys. Because the files are in binary format, you cannot view the certificate using a text editor. However, most applications recognize and can process DER-encoded certificate files. The Director node uses the versa_director_web_client.cer certificate based on DER encoding.
- PEM-encoded certificate—PEM encoding uses Base64-encoded ASCII files. It does not store the certificate path or private key information. The Director node uses the versa_director_web_client.crt certificate based on PEM encoding.

If you are using a DER-encoded certificate, you must convert it to CER format before you import it to the Director node. To do this, you can use an SSL convertor website to convert, or you can issue the following command from the Linux shell:

```
openssl x509 -in certificate-name.der -inform DER -out certificate-name.cer
```

To import the converted DER-encoded certificate to the Director node, execute the vnms-import-key-cert.sh script on the Director node (type the command on one line):

```
Admin@Director:/opt/versa/vnms/scripts# ./vnms-import-key-cert.sh --key --storepass password--key
key-file-name.key --cert certificate-name.cer --keypass encrypted-private-key-password
```

After you import the DER-encoded certificate, the installation steps are the same for PEM-encoded and DER-encoded certificates.

The Director node uses Java KeyStore (JKS) as the keystore format. JKS is a repository of security certificates that is commonly used with SSL encryption.

On a Director node, the server private key and certificates are installed in the /var/versa/vnms/data/certs directory, in the following files:

```
Admin@VersaDirector:~# cd /var/versa/vnms/data/certs/
Admin@VersaDirector:/var/versa/vnms/data/certs# ll
total 48
drwxr-xr-x 3 root root 4096 Nov 20 08:32 ./
drwxr-xr-x 3 root root 4096 Nov 20 00:13 ../
-rw-r--r-- 1 root root 4390 Nov 20 08:32 tomcat_keystore.jks
-rw-r--r-- 1 root root 1366 Nov 20 08:32 versa_director_client.cer
-rw-r--r-- 1 root root 1907 Nov 20 08:32 versa_director_client.crt
-rw-r--r-- 1 root root 1366 Nov 20 08:32 versa_director_web_client.cer
-rw-r--r-- 1 root root 1907 Nov 20 08:32 versa_director_web_client.crt
-rw-r--r-- 1 root root 4387 Nov 20 08:32 vnms_keystore.jks
-rw-r--r-- 1 root root 1433 Nov 20 08:32 vnms_truststore.ts
```

The Director node uses the following stores for keys and certificates:

Store Name	**Description**
tomcat_keystore.jks	Stores the private key, the public certificate, and root CA certificate for the Tomcat web server.
vnms_keystore.jks	Stores the private key for the karaf server.
vnms_truststore.ts	Stores the public key certificates and root CA certificate for the karaf server.

The Director node uses the following certificates. Note that CRT and CER are two different certificate formats.

Certificate	**Description**
versa_director_client.cer versa_director_client.crt	Certificates used by Director clients to invoke RESTful APIs. These are public certificates for the Kafka server.
versa_director_web_client.cer versa_director_web_client.crt	Public certificates for the Tomcat web server.

For Releases 20.2.1 and later, when the SSL certificate is nearing its expiration, the Director GUI displays an alarm at the following times:

- 30 days or less before expiration, the Director node displays a warning alarm.

- 7 days or less before expiration, the Director node displays a critical alarm.

After you renew the SSL certificate, the Director node clears the alarm.

Install Existing Private Key and CA-Signed Certificate on a Director Node

After the CA returns the signed certificate to you, you install the private key and the certificate on the Director node.

If you are using a high availability (HA) setup, ensure that you create, load, or import the certificates on the active node. After the certificates are placed on the active node, they are automatically synchronized from the active node to the standby node.

To import the private key and CA-signed certificate on the Director node:

1. Ensure that all CA certificates are concatenated into a single file:

   ```
   Admin@Director:~# cat intermediate-certificate.crt root-certificate.crt > CA-certificate
   ```

2. If your setup consists of a single Director node, stop the Director node by issuing the **vsh stop** command. For Releases 20.2 and 21.1, issue the **service vnms stop** command. If you have a HA setup, do not perform this step.

   ```
   Admin@Director:~# vsh stop
   ```

   For example:

   ```
   Admin@Director:~# vsh stop
   Stopping VNMS service
   ----------------------------------
   Stopping VNMS:NCS..........[Stopped]
   Stopping VNMS:REDIS........[Stopped]
   Stopping VNMS:KARAF........[Stopped]
   Stopping VNMS:TOMCAT.......[Stopped]
   ```

3. Verify the certificate:

   ```
   Admin@Director:~# openssl verify -verbose -CAfile CA-certificate CA-signed-certificate
   ```

   For example:

   ```
   Admin@Director:~# openssl verify example.com.crt
   example.com.crt: C = US, ST = California, O = example.com, OU = IT,
   CN = example.com, emailAddress = administrator@example.com
   error 20 at 0 depth lookup:unable to get local issuer certificate
   ```

   Note that OpenSSL verifies only a CA-signed certificate. For example, for a .pem file, the response is as follows:

   ```
   Admin@Director:~# openssl verify remote.site.pem
   remote.site.pem: OK
   ```

4. If you have an external CA-signed certificate, execute the vnms-import-key-cert.sh script (type the command on one line):

   ```
   Admin@Director:/opt/versa/vnms/scripts# ./vnms-import-key-cert.sh --key private-key-file
   --cert CA-signed-cert-file [--keypass encrypted-private-key-password]
   ```

   The script places a backup of the existing certificates and keystores in the /var/versa/vnms/data/certs/backup directory. For example:

   ```
   Admin@Director:/opt/versa/vnms/scripts# ./vnms-import-key-cert.sh --storepass versa1234 --key test-
   private.key --cert test-public.crt --keypass versa123
   => Taking backup of existing certificates and keystores in /var/versa/vnms/data/certs/backup
   => Successfully created pkcs12 file
   => Importing key and certificates to keystores
   => Generating websockify certificates
   => Saving storepass and keypass
   ```

   The script places the generated keystores and truststores in the /var/versa/vnms/data/certs/ directory, and then imports the CA-signed certificate into the Director node into that directory. For example:

   ```
   Admin@VersaDirector:/var/versa/vnms/data/certs# ls -l
   total 52
   drwxr-xr-x 3 root root 4096 Nov 20 09:51 ./
   drwxr-xr-x 3 root root 4096 Nov 20 00:13 ../
   drwxr-xr-x 2 root root 4096 Nov 20 09:47 backup/
   -rw-r--r-- 1 root root 1944 Nov 20 09:37 example.com.crt
   -rw-r--r-- 1 root root 1074 Nov 20 09:08 example.com.csr
   -rw-r--r-- 1 root root 1751 Nov 20 09:08 example.com.key
   -rw-r--r-- 1 root root 2751 Nov 20 09:47 tomcat_keystore.jks
   -rw-r--r-- 1 root root 1393 Nov 20 09:47 versa_director_client.cer
   -rw-r--r-- 1 root root 1944 Nov 20 09:47 versa_director_client.crt
   -rw-r--r-- 1 root root 1393 Nov 20 09:47 versa_director_web_client.cer
   -rw-r--r-- 1 root root 1944 Nov 20 09:47 versa_director_web_client.crt
   -rw-r--r-- 1 root root 2751 Nov 20 09:47 vnms_keystore.jks
   -rw-r--r-- 1 root root 1460 Nov 20 09:47 vnms_truststore.jks
   ```

5. Set the ownership of the files to versa:versa:

   ```
   Admin@VersaDirector:/var/versa/vnms/data/certs# sudo chown -R versa:versa /var/versa/vnms/data/
   certs
   ```

   Alternately, run the script as a Versa user (type the command on one line). Note that **keypass** and **cafile** are optional arguments.

   ```
   Admin@VersaDirector:/var/versa/vnms/data/certs# sudo -u versa vnmsimportkeycert.sh --key private-
   key-filename --cert CA-signed-cert-filename
   [--keypass encrypted-private-key-password] [--cafile full-path-to-CA-root-certificate-file]
   ```

6. For a HA setup, wait for about 5 minutes, and then verify that the certificates and keystores have been synchronized from the active node to the standby node. To do this, check the /var/versa/vnms/data/certs directory on the standby Director node. Note that automatic synchronization between the active and standby Director nodes is performed approximately every 60 seconds.

7. Start the Director node:
    - For a standalone Director node, issue the **service vnms start** command:

      ```
      Admin@VersaDirector:/var/versa/vnms/data/certs# service vnms start
      ```

    - For an HA setup, issue the **vsh stop** command on the standby Director node to stop the Versa services on the standby node, and then issue the **vsh restart** command on the active Director node to start the Versa services on the active node. Issuing the commands in this sequence prevents the standby node from taking on the active role.

8. After few minutes, verify that all the processes have initialized:

    ```
    [Administrator@VersaDirector: ~] $ vsh status
    [sudo] password for Administrator:
    NCS[4.7.10]             [Running]
    POSTGRE[15.3]              [Running]
    NETBOX-IPAM                [Running]
    SPRING-BOOT                [Running]
    REDIS[6.2.6]            [Running]
    APACHE TOMCAT/9.0.75           [Running]
    NODE-EXPORTER                [Running]
    ```

9. For an HA setup:
    a. When all the Versa services on the active node are up, issue the **vsh start** command on the standby node, and then confirm that all the Versa services on the standby node are running.
    b. Issue the **request vnmsha actions check-sync-status** command on both the active and standby nodes to confirm that the two HA Director nodes are in sync.
10. In a browser window, go to https://*hostname*, and verify the certificate information.
11. After you install or update the versa_director_client.cer file, you must import the new certificate on the Analytics node so that you can access the Analytics UI. To do this, issue the following command on the Director node as the user versa. You are prompted for the Analytics cluster name and IP address and the ssh password for connecting to the Analytics node. To find the name of the Analytics cluster, select the Administrator tab in Director view, and then select Connectors > Analytics in the left menu bar.

    ```
    [Administrator@VersaDirector: ~] $ sudo su versa
    [versa@VersaDirector: ~] $ sudo -u versa /opt/versa/vnms/scripts/vnms-cert-sync.sh --sync
    ```

Generate a Key and CSR for Real Certificate Deployment

You can create a certificate, get it signed by a CA, and then use it on a Director node. To do this, you create a private key and a certificate signing request (CSR) on the Director node. The process of generating a CSR creates a public and private key. After you receive a CA-signed certificate, you install the private key and the certificate on the Director node.

Ensure that you create, load, and import the certificates on the active node. In a high availability (HA) setup, the certificates are automatically synchronized from the active node to the standby node. To verify whether the certificates have been synchronized from the active to the standby node, check the /var/versa/vnms/data/certs directory. After you ensure that the certificates have been synchronized, stop the services on the standby node and then restart the services on active node. When the services on the active node are up, restart the services on the standby node.

To create a private key and a CSR, execute the vnms-csrgeh.sh script, specifying the options described in the following table. (Type the command on a single line.) The following example shows the script options:

```
# cd /opt/versa/vnms/scripts
# ./vnms-csrgen.sh --domain director-01 --country US --state CA --locality SC \
--organization versa-networks.com --organizationalunit IT --email admin@versa-networks.com \
--keypass test123 --validity 365 --san director-01,DNS:director-02
```

DN Field	Description
country	Two-letter ISO abbreviation for the country. For example, US.
domain	Common name for the active Director node. This must be an exact match. For example, director-01.
email	Email address of your organization. For example, admin@versa-networks.com.
keypass	Password for the private key encryption. For example, versa123.
locality	City or locality where the organization is legally located. For example, Santa Clara or SC.
organization	Exact legal name of your organization. Do not abbreviate the organization name. For example, versa-networks.com.
organization unit	Section of the organization. For example, IT.
san	Common name of all Directors in a high availability (HA) topology, separated by commas. You may need to add the name source with a colon for non-active Director nodes. For example, director-01,*dns*:director-02.
state	State or province where the organization is legally located. Avoid abbreviations. For example, California.
validity	Validity length of the certificate, in days. For example, 365.

The following sample output shows the generation of a private key and a CSR:

```
Admin@VersaDirector:/opt/versa/vnms/scripts# ./vnms-csrgen.sh --domain director-01 \
--country US --state CA --locality SC --organization versa-networks.com \
--organizationalunit IT --email admin@versa-networks.com --keypass versa123 \
--validity 365 --san director-01,DNS:director-02
=> Generating Key and CSR for request domain: director-01, key_pass: versa123
=> Request details: [/C=US/ST=CA/L=SC/O=versa-networks.com/OU=IT/CN=versa-networks.com/
emailAddress=admin@versa-networks.com]
```

```
=> Generating with password encrypted keypass: versa123 and CSR
=> Successfully generated Key and CSR file: /var/versa/vnms/data/certs/versa-networks.com.key, /var/versa/
vnms/data/certs/versa-networks.com.csr
```

To verify the CSR, issue the following command:

```
# openssl req -noout -text -in /var/versa/vnms/data/certs/name.csr
```

For example, the following script verifies privately generated certificate attributes:

```
# openssl req -noout -text -in /var/versa/vnms/data/certs/director-01.csr
Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject: C=US, ST=California, L=Santa Clara, O=versa-networks.com, OU=IT,
        CN=versa-networks.com/emailAddress=administrator@versa-networks.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:dd:4d:ed:95:8f:4b:cf:af:af:dc:f5:15:75:9a:
                    79:21:62:3c:4d:01:61:a9:1d:54:39:82:e9:2c:e9:
                    38:84:1a:d1:68:1c:a6:45:3d:c8:45:58:73:39:6a:
                    6c:ab:4a:7a:81:d7:17:22:d2:11:0b:5f:66:97:f5:
                    cd:a9:76:59:fc:da:b9:bd:5d:3e:1c:91:9f:ac:4c:
                    23:28:13:c4:c8:29:24:f5:c2:69:54:8e:3c:81:8d:
                      0c:74:7d:1c:fc:5d:0a:af:7d:f0:2e:d6:8b:15:1b:
                    0c:96:30:69:6c:04:01:d2:e2:a8:b3:0b:00:7d:27:
                    6e:ca:66:34:20:e9:8b:bd:99:d9:50:22:c8:e4:cf:
                    d5:1d:10:ac:f3:67:26:42:69:c0:dd:a7:fd:74:fc:
                      5f:84:58:b3:2e:99:0c:39:17:f4:ff:a4:e1:58:89:
                    68:c3:87:c5:e0:62:19:d3:30:39:84:b6:86:82:8b:
                    66:45:9e:6f:35:a2:cb:53:ab:dc:e4:19:4e:f3:a1:
                    2a:38:dd:28:1f:26:6f:1f:07:a9:16:55:5f:ed:51:
                    61:23:03:df:ea:5d:0b:d6:a7:50:b3:81:96:da:c7:
                    d2:c8:5b:0a:03:00:c5:38:a7:91:79:b9:93:29:1a:
                    bc:c3:1e:89:a8:38:2c:45:fe:47:3e:11:7b:c8:da:
                      09:bb
                Exponent: 65537 (0x10001)
              Attributes: a0:00
    Signature Algorithm: sha256WithRSAEncryption
    5a:22:cc:d6:a7:ee:35:92:69:b5:8d:2b:ad:06:68:85:09:04:
    e2:7e:eb:a0:dd:7e:9e:f3:30:80:4b:56:92:01:d0:04:8d:08:
    d5:e5:b0:e0:e2:e7:cb:c2:6d:03:b3:d3:9b:fd:04:b8:2c:5f:
    27:74:1f:17:78:71:76:98:72:9d:13:62:60:82:12:8d:fd:8b:
    6f:6e:05:56:4f:bb:c1:5f:ed:14:2e:2e:3c:59:f9:01:5f:cf:
    cf:cc:37:60:ec:52:e0:33:ad:e7:c5:04:54:63:4d:c3:e2:1e:
    88:9d:f5:fc:c4:ac:f5:d9:96:5d:60:1a:1c:c0:52:01:c3:cf:
    9d:f6:73:85:9b:54:36:ed:28:fc:81:0c:7b:e1:31:ec:a3:33:
    31:6a:13:17:04:53:79:2d:75:16:9e:07:94:4b:08:fd:50:c6:
    61:9c:c1:be:53:57:a5:7a:11:fb:58:5d:63:0a:cb:0d:a6:25:
    a9:c3:01:85:9a:84:48:6d:be:37:9e:17:8b:02:6a:e6:93:2a:
    59:8a:3f:b2:17:e4:ca:4b:fc:c5:30:d7:c4:e5:ec:82:4d:d3:
```

```
17:a1:81:6b:ba:7d:c6:aa:cd:61:2d:2e:ac:d6:6d:bb:f7:7a:
91:20:fe:03:8e:65:5b:41:1a:82:e7:5a:23:b6:78:90:8a:c4:
39:c4:e3:44
```

After you generate the CSR, you must submit it to an external CA, such as Symantec, Thawte, or VeriSign, to have them sign the certificate request. When you choose a CA, ensure that the CA satisfies the following criteria:

- Allows freedom to deploy certificates on any number of servers
- Provides an efficient certification management tool to manage all certificates
- Provides technical support
- Is a publicly trusted CA

Generate Self-Signed Certificates

When you first install the Director software, it includes a complete set of self-signed certificates.

Ensure that you create, load, or import the certificates on the active node. In an HA setup, the certificates are automatically synchronized from the active node to the standby node. To verify whether the certificates have been synchronized from the active to the standby node, check the /var/versa/vnms/data/certs directory. After you ensure that the certificates have been synchronized, stop the services on the standby node and then restart the services on active node. When the services on the active node are up, restart the services on the standby node.

If you need to generate a new self-signed certificate, issue the following commands as the user "versa". Do not generate the certificate using sudo privilege.

```
# su - versa
# cd /opt/versa/vnms/scripts
# ./vnms-certgen.sh --cn active-director --san backup-director --overwrite --storepass password
```

Field	Description
cn	Common name for the active Director node, for example, director-01.
overwrite	Overwrite the existing certificate with the newly generated certificate. If you include the overwrite keyword, a new certificate is generated that replaces the existing certificate when you run the script. If you omit the overwrite keyword, the existing certificate is updated with new attributes when you run the script.
san	Common names of all Director nodes in the Director cluster, separated by a comma. Specify all the node names in the format dns:*director-name.* For example, director-01,dns:director-02
storepass	Password to use to access the certificate through the Java Truststore and the Java Keystore. The default password is versa123, and you must use this password for Releases 16.1R2 through Release 16.1R2S8, and for Releases 20.1. For Release 16.1R2S9 and later versions of Release 16.1R2, you can use any password.

The following example of executing this script generates a certificate with a password of example123 and stores the certificate in the /var/versa/vnms/data/certs directory on the Director node:

```
# su - versa
# cd /opt/versa/vnms/scripts
# ./vnms-certgen.sh --storepass example123
=> Generating certificate for domain: versa-director
=> /var/versa/vnms/data/certs is empty, continue creating new certificates
=> Generating ca_config.cnf
=> Generated CA key and CA cert files
=> Generating SSO certificates
=> Generating websockify certificates
=> Saving storepass and keypass
```

To regenerate the self-signed certificate for a Director server:

Note: It is recommended that you back up the existing certificate and keys before regenerating the self-signed certificate.

1. Ensure that the certificate directory, /var/versa/vnms/data/certs, contains no files and is empty.
2. Run the certificate regeneration script:

   ```
   sudo -u versa /opt/versa/vnms/scripts/vnms-certgen.sh --storepass versa123 --keypass versa123
   ```

3. Install an existing private key and CA signed certificate on the Director server (type the command on a single line):

   ```
   ./vnms-import-key-cert.sh --key private-key-filename --cert CA-signed-certificate-filename
   ```

```
--storepass versa123 [--keypass private-key-password]
```

View Self-Signed Certificates

To view the self-signed certificates, use the Java **keytool** command on the Director node.

To view all certificates, issue the following command:

```
# keytool -list -keystore vnms_keystore.jks -storepass versa123 -keypass versa123
Keystore type: JKS
Keystore provider: SUN
Your keystore contains 1 entry
vnmsserver, Nov 20, 2014, PrivateKeyEntry,
Certificate fingerprint (SHA1): 59:30:F7:EF:CC:6A:E6:EF:8B:14:2F:1C:B3:23:5B:59:5F:93:CA:0
```

To view the information about a specific certificate, specify the certificate name, for example, versa_director_client.crt:

```
# keytool -printcert -file versa_director_client.crt
```

To view the information in the Director self-signed certificate, issue the **keytool –printcert** command. For example:

```
Admin@Director:/var/versa/vnms/data/certs# keytool -printcert
-file versa_director_client.crt
Owner: CN=, OU=VersaDirector, O=versa-networks, ST=California, C=US
Issuer: L=Santa Clara, ST=California, C=US, OU=VersaDirector, O=versa-networks
Serial number: 1
Valid from: Thu Nov 20 00:10:39 EST 2014 until: Sat Nov 19 00:10:39 EST 2016
Certificate fingerprints:
    MD5:  52:AF:0D:1B:85:94:66:EB:19:4E:DE:E6:18:C9:6A:18
SHA1: 8A:5D:61:CD:B0:1D:DC:6D:41:55:45:E9:4D:60:D8:F1:CD:7E:32:8A
    SHA256: EE:B4:FB:58:59:0A:5F:E4:14:84:63:51:BA:2F:17:C9:AD:C4:76:
D6:01:FB:DC:52:53:8B:E5:55:8D:ED:98:DF
    Signature algorithm name: SHA256withRSA
    Version: 3
Extensions:
#1: ObjectId: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier [
KeyIdentifier [
0000: 91 9E B2 0A 45 70 DB 2C   D7 00 65 F9 C5 A0 A3 87  ....Ep.,..e.....
0010: 92 10 4F 3E                                        ..O>
]
[L=Santa Clara, ST=California, C=US, OU=VersaDirector, O=versa-networks]
SerialNumber: [ 9f0a3155 84ae03bb]
]
#2: ObjectId: 2.5.29.19 Criticality=false
BasicConstraints:[
  CA:true
  PathLen:2147483647
]
#3: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 86 26 CE DC A9 D1 3B 9B   8B 0C D2 D5 D9 0C FE 33  .&....;........3
0010: B2 7A DB A3                                        .z..
]
]
```

Best Practices for Managing SSL Certificates

To successfully create and manage SSL certificates on Director nodes, follow these best practices:

- It is recommended that you use a self-signed certificate only for POC and lab testing. When you deploy the product, you should use a certificate signed by a third-party CA.
- Implement stronger cryptography methods. Use 2048-bit private keys for servers. 2048-bit private keys are the minimum recommended standard per NIST special publication 800-131A and per Common Criteria Protection Profile PP_APP_v1.1. They are secure and difficult to hack.
- Use passwords to protect encrypted keys for strengthening security in backup systems.
- Include all possible domain names in the certificates. It is recommended that you use wildcard certificates.
- Create an inventory of the SSL certificates and keys that are deployed on the network. Verify whether the certificates are compliant with your organization's policies.
- Use a reliable certification authority.
- Monitor the expiration date of existing certificates. Use a management application interface to track the life of certificates and keys so that you are aware of when a certificate is about to expire.
- Encrypt the data traffic on your website.
- Check and install system updates and patches frequently.

Supported Software Information

Releases 20.2 and later support all content described in this article, except:

- For Releases 21.2 and later, the **vsh stop** command replaces the **service vnms stop** command.

Additional Information

Configure Certificate Servers
Configure a Common Certificate Authority
Configure CSR Objects
Configure Kerberos Authentication
Perform Initial Software Configuration
Troubleshoot Analytics Access and Certificate Issues