Implementation details and troubleshooting - uploading custom domain certificate instead of using realhostip.com

Implementation Details of uploading custom certificate

API - UploadCustomCertificate

- API has to be invoked multiple times to upload the certificates and has to be done in order following from root cert, 0 or more intermediate certs followed by server cert and private key.
- Uploading root certificate and the server certificates are mandatory steps.
- Uploading intermediate certificates(s) is optional.
- Self-signed certificates are not supported.
- Reverting back to Realhostip.com or the last certificate is not supported.
- Understanding parameters -
    - When API is invoked directly instead if UI make sure the certificates are URL encoded. One way is to use Google chrome - Advanced Rest Client to url encode your certificate (basically it converts the newline into %0A so the certificate becomes 1 line rather than multiple lines).
    - Make sure the certificates are in PEM format.
    - API invocation has to be in order ie first the root certificate with id=1 then zero or more intermediate certificates with id =2, 3, 4 etc.
    - There is no convention for the name parameter but it would help to name the root certificate as "root", intermediate certificates as "intermediate1", "intermediate2" etc. NOTE - Keep the names always unique.
    - domainsuffix should be same as the global config secstorage.ssl.cert.domain/consoleproxy.url.domain = yourdomain.com and for all the API invocations.

DataBase -

'keystore' table stores the certificate.

Schema -

```
mysql> desc keystore;
+---------------+---------------------+------+-----+---------+----------------+
| Field | Type | Null | Key | Default | Extra |
+---------------+---------------------+------+-----+---------+----------------+
| id | bigint(20) unsigned | NO | PRI | NULL | auto_increment |
| name | varchar(64) | NO | UNI | NULL | |
| certificate | text | NO | | NULL | |
| key | text | YES | | NULL | |
| domain_suffix | varchar(256) | NO | | NULL | |
| seq | int(11) | YES | | NULL | |
+---------------+---------------------+------+-----+---------+----------------+
```

SSVM

- **SecStorageSetupCommand -**
    - All ssvms are rebooted when UploadCustomCertificate is invoked for server cert and private key. As part of ssvm agent connect with MS, a command called **SecStorageSetupCommand** containing all the certs and the key is sent to ssvm (you wont see the certs as it is sensitive information and not logged).
    - On the ssvm side the root cert is inserted into the keystore.
    - For apache server a server cert. file, a key file and a certificate chain file (consisting of certs from root cert to the intermediate cert) are created in the appropriate location and location put into the config file.
- **SSVM acts as a server** for download template/volume/iso operations and during copyTemplate if it is in the source zone. It uses an Apache webserver when acting as server.
    - Apache server should have the latest key, server cert and chain under the following location -
        - /etc/ssl/private/cert_apache.key
        - /etc/ssl/certs/cert_apache.crt
        - /etc/ssl/certs/cert_apache_chain.crt
    - /etc/apache2/sites-available/default-ssl is the config file for apache server where the chain location, server cert and private key are configured.
    - When client contacts the apache server it needs to present the entire chain of certificates to the client.
- **It acts as a client** during copyTemplate operation if it is in the destination zone. SSVM uses a Java client when acting as a client.
    - SSVM has a java keystore which should have the root certificate in the keystore so that when it tries to download from the url it can validate the chain presented by the server against the known root.
      Use the command below to list all certs in the keystore

      ```
      keytool -list -keystore /usr/local/cloud/systemvm/certs/realhostip.keystore -storepass vmops.com
      ```

CPVM

- CPVM acts as a server only and uses Java HTTP webserver

- Every time CPVM starts up, the keystore is created in-memory by management server and the data sent to CPVM
- CPVM uses this keystore to configure the Java webserver with right private key and certificate
- The key is saved with label “CPVMCertificate” in keystore
- Every time uploadSSL API is called, CPVM reboots
- Will be enhanced to reboot only when full chain is uploaded
- CPVM has an in-memory keystore (similar to SSVM’s realhostip.keystore)

TroubleShooting

Basics

Any issues with SSVM functionality -

- **Check logs** –
    - Check MS logs (/var/log/cloudstack/management) and
    - ssvm logs (under /var/log/cloud/) to see if you can find any exceptions in programming the certificate.
- **SSVM acting as server** -
    - Via browser or any client, check whether Apache Webserver for SSVM is sending the entire chain. If not,
    - Make sure certs and key are programmed correctly on apache server by tracing command SecStorageSetupCommand. Read SSVM implementation for details.
    - Check SSVM/MS logs to see any exception.
    - Make sure certs are uploaded fine in relevant /etc/ location as discussed in implementation above.
- **SSVM acting as client** –
    - Ensure that realhostip.keystore (Java keystore) on destination SSVM has the root CA certificate : keytool -list -keystore realhostip.keystore -storepass vmops.com
    - Check DB has certs as you uploaded with no spaces or garbage characters, correct domain and sequence.
- **Destroy SSVM** - If nothing is apparent from the logs try destroying ssvm as one of the last resorts. Try this in one of the zones and see if the new ssvm which comes up solves the issues.
- **Check certificates format -** Check in the DB, keystore table that the certificates uploaded are correct format and match the certificates with customers. Some of the things which you can check for are
    - Spaces in the certificate - there shouldnt be any spaces in the certificate. Probably url encoding is incorrect.
    - Missing "-----BEGIN CERTIFICATE-----" OR "-----END CERTIFICATE-----"
    - Incorrect hyphens in the begin certificate above.

Specific Issues seen

1. Download urls point to the old domain.
    a. Reduce the expiration duration of the urls by changing global config extract.url.expiration.interval
    b. And change the frequency for cleanup thread through extract.url.cleanup.interval restart MS.
    c. Wait for the cleanup thread duration and try downloading again. See whether the url is deleted.
    d. DB tables to check (don’t recommend but worst case)
       Version < 4.2 – upload table persists url. Entry is hard deleted on expiration of url.
       Version >= 4.2 –
       template_store_ref, download_url is made null on expiration of url.
       volume_store_ref, entry hard deleted on expiration of url.
2. CopyTemplate giving the exception -
   Connection timed out
   Exception – ‘sun.security.validator.ValidatorException: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target’
    a. Server side validation - Via browser or any client, check whether Apache Webserver for SSVM is sending the entire chain. If not,
        i. Make sure certs and key are programmed correctly on apache server. Read ssvm implementation for details.
        ii. Check ssvm/MS logs to see any exception
        iii. Make sure certs are uploaded fine.
    b. Client side validation – Ensure that realhostip.keystore (Java keystore) on destination SSVM has the root CA certificate : keytool -list -keystore realhostip.keystore -storepass vmops.com
3. Download urls are not working.
    a. Via browser or any client, check whether Apache Webserver for SSVM is sending the entire chain
4. I don’t see the latest systemvm.iso being patched since the /usr/local/cloud/systemvm/config_ssl.sh on ssvm is still the old one.

    a. Depending on version and hypervisor – make sure latest systemvm.iso is propagated properly. Check md5sum of iso on MS and iso on the location eg.
       For XS – its host under /opt/xensource/packages/iso
       For VMware – its all secondary storage under systemvm folder
       For KVM – it should be packaged with cloud-agent package

5. I changed CPVM to work on HTTPS from HTTP, or vice-versa. It does not change.
    a. You may need to destroy and recreate your CPVM when switching between HTTP and HTTPS protocols.
    b. Check if DNS server is working correctly
    c. Check global config – it should have “*” for versions >= CCP 4.3
6. Certificate encoding -
    a. Make sure you have correctly url encoded the certificates. To do this you can check cloud database and check the entries in keystore table and match them with the original certificates you had.
7. Check the locations - Custom Private Key location /etc/ssl/private/cert_apache.key , Custom Private Cert location /etc/ssl/certs/cert_apache.crt and if chain is sent then Custom Cert Chain location is at /etc/ssl/certs/cert_apache_chain.crt

    a. Do note that the chain file should be populated with the root certificate till the intermediate certificate.
8. Uploaded wrong certificate -
    a. In case you uploaded wrong certificates for root/intermediate, you can undo that by calling the api with same name and domainsuffix. In case you uploaded wrong server certificate upload the right one through the UI keeping the same domain suffix.
9. Any other issues with ssvm -
    a. Check logs - Check MS logs (/var/log/cloudstack/management) and ssvm logs (under /var/log/cloud/) to see if you can find any exceptions in programming the certificate.
    b. Destroy ssvm - If nothing is apparent from the logs try destroying ssvm as one of the last resorts. Try this in one of the zones and see if the new ssvm which comes up solves the issues.
10. I have already uploaded certificate and chain in a prior CCP version. Am I all good?
    a. Unfortunately, no. You will need to re-upload the whole chain using the same API parameters (name, id etc.)