Configure OVS Connection Using SSL with Self-signed Certificates

To make a SSL/TLS Openflow connection between onos and OVS switches using self-signed certificates, there are five main steps to follow:

1. *Generate SSL key/certificate for onos;*
2. *Test the SSL connection.*
3. *Copy the onos certificate to the appropriate OVS location so that OVS can accept the certificate from onos;*
4. *Generate SSL key/certificate for OVS;*
5. *Copy the OVS certificate to the appropriate onos location so that onos can accept the certificate from OVS;*

The following is an example of the detailed configuration steps.

Step 1. Generating SSL key/certificate for onos. On the host running onos, we generate the SSL key/certificate as the following:

a). Create a folder named “ssl”, in this folder we use "keytool" to generate a .jks keystone:

```
root@devel:/opt/onos# cd ssl
root@devel:/opt/onos/ssl# keytool -genkey -keyalg RSA -alias onos -keystore onos.jks -storepass 111111 -
validity 365 -keysize 2048
What is your first and last name?
  [Unknown]:  Ai Haoyu
What is the name of your organizational unit?
  [Unknown]:  FNLab-BUPT
What is the name of your organization?
  [Unknown]:  BUPT
What is the name of your City or Locality?
  [Unknown]:  Beijing
What is the name of your State or Province?
  [Unknown]:  Beijing
What is the two-letter country code for this unit?
  [Unknown]:  CN
Is CN=Ai Haoyu, OU=FNLab-BUPT, O=BUPT, L=Beijing, ST=Beijing, C=CN correct?
  [no]:  yes


Enter key password for <onos>
        (RETURN if same as keystore password):


Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry
standard format using "keytool -importkeystore -srckeystore onos.jks -destkeystore onos.jks -deststoretype
pkcs12".
root@devel:/opt/onos/ssl# ls
onos.jks
```

b). Covert the .jks keystore (which onos uses) to PEM file (which OVS uses) in a 2-step conversions: from .jks to .p12, then to .pem:

```
root@devel:/opt/onos/ssl# keytool -importkeystore -srckeystore onos.jks -destkeystore onos.p12 -srcstoretype
jks -deststoretype pkcs12
Importing keystore onos.jks to onos.p12...
Enter destination keystore password:
Re-enter new password:
Enter source keystore password:
Entry for alias onos successfully imported.
Import command completed:  1 entries successfully imported, 0 entries failed or cancelled
root@devel:/opt/onos/ssl# ls
onos.jks  onos.p12
root@devel:/opt/onos/ssl# openssl pkcs12 -in onos.p12 -out onos.pem
Enter Import Password:
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
root@devel:/opt/onos/ssl# ls
onos.jks  onos.p12  onos.pem
```

c). Use the certificate portion of the "onos.pem" file to create a new file, called "cacert.pem" - this is the file to be copied over to OVS - it is from "Bag Attributes" to "END CERTIFICATE":

```
root@devel:/opt/onos/ssl# cat onos.pem
<Private key here>
Bag Attributes
    friendlyName: onos
    localKeyID: 54 69 6D 65 20 31 35 33 39 33 32 37 38 34 32 31 39 39
subject=/C=CN/ST=Beijing/L=Beijing/O=BUPT/OU=FNLab-BUPT/CN=Ai Haoyu
issuer=/C=CN/ST=Beijing/L=Beijing/O=BUPT/OU=FNLab-BUPT/CN=Ai Haoyu
-----BEGIN CERTIFICATE-----
MIIDbzCCAlegAwIBAgIEOtHFEDANBgkqhkiG9w0BAQsFADBoMQswCQYDVQQGEwJD
TjEQMA4GA1UECBMHQmVpamluZzEQMA4GA1UEBxMHQmVpamluZzENMAsGA1UEChME
......
-----END CERTIFICATE-----



root@devel:/opt/onos/ssl# sudo vi ./cacert.pem
Bag Attributes
    friendlyName: onos
    localKeyID: 54 69 6D 65 20 31 35 33 39 33 32 37 38 34 32 31 39 39
subject=/C=CN/ST=Beijing/L=Beijing/O=BUPT/OU=FNLab-BUPT/CN=Ai Haoyu
issuer=/C=CN/ST=Beijing/L=Beijing/O=BUPT/OU=FNLab-BUPT/CN=Ai Haoyu
-----BEGIN CERTIFICATE-----
MIIDbzCCAlegAwIBAgIEOtHFEDANBgkqhkiG9w0BAQsFADBoMQswCQYDVQQGEwJD
TjEQMA4GA1UECBMHQmVpamluZzEQMA4GA1UEBxMHQmVpamluZzENMAsGA1UEChME
.......
-----END CERTIFICATE-----
```

Note:

The intermediate key/cert, "onos.p12", and onos.pem, are no longer used and should be discarded.

Step 2. Copy the onos certificate to the appropriate OVS location so that OVS can accept the certificate from onos:

a). In the case of PicaOS, firstly we need to install openssl module, then create controllerca directory and switchca directory, and generate key and certificate for the OVS switch in switchca directory:

```
admin@PICOS-OVS:~$ sudo apt-get install openssl
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssl is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 60 not upgraded.
admin@PICOS-OVS:~$ sudo su
root@PICOS-OVS:/home/admin# ovs-pki init --force
Creating controllerca...
Creating switchca...
root@PICOS-OVS:~$ cd /ovs/var/lib/openvswitch/pki/switchca
root@PICOS-OVS:/ovs/var/lib/openvswitch/pki/switchca# ovs-pki req+sign sc switch
sc-req.pem      Mon Oct 15 16:01:11 UTC 2018
        fingerprint 0d336057404dab9bc7dc158b7f4a7007ced6efd4
root@PICOS-OVS:/ovs/var/lib/openvswitch/pki/switchca# ls -lat
total 76
drwxr-xr-x 6 root root 4096 Oct 15 16:01 .
.....
-rw-r--r-- 1 root root 4002 Oct 15 16:01 sc-cert.pem
-rw-r--r-- 1 root root 3565 Oct 15 16:01 sc-req.pem
-rw------- 1 root root 1675 Oct 15 16:01 sc-privkey.pem
-rw-r--r-- 1 root root 4028 Oct 15 15:56 cacert.pem
....
drwxr-xr-x 4 root root 4096 Oct 15 15:56 ..
```

a). Copy cacert.pem from your ONOS working directory(/opt/ssl) to this directory your OVS machine: "/ovs/var/lib/openvswitch/pki /controllerca/cacert.pem", you can replace or backup the original cacert.pem file.

```
root@PICOS-OVS:/etc# mkdir ovsbackup
root@PICOS-OVS:/etc# cd ovsbackup/
root@PICOS-OVS:/etc/ovsbackup# cd /ovs/var/lib/openvswitch/pki/controllerca
root@PICOS-OVS:/ovs/var/lib/openvswitch/pki/controllerca# ls
ca.cnf  cacert.pem  careq.pem  certs  crl  crlnumber  index.txt  index.txt.attr  index.txt.
old  newcerts  private  serial  serial.old
root@PICOS-OVS:/ovs/var/lib/openvswitch/pki/controllerca# mv cacert.pem /etc/ovsbackup/
root@PICOS-OVS:/ovs/var/lib/openvswitch/pki/controllerca# ls
ca.cnf  careq.pem  certs  crl  crlnumber  index.txt  index.txt.attr  index.txt.
old  newcerts  private  serial  serial.old
root@PICOS-OVS:/ovs/var/lib/openvswitch/pki/controllerca# scp pica8@10.10.50.243:/opt/ssl/cacert.pem .
The authenticity of host '10.10.50.243 (10.10.50.243)' can't be established.
ECDSA key fingerprint is a3:a4:a2:ab:a9:bd:81:02:16:7a:9b:8d:f9:07:e3:90.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.10.50.243' (ECDSA) to the list of known hosts.
pica8@10.10.50.243's password:
cacert.
pem
      100% 1499     1.5KB/s   00:00
root@PICOS-OVS:/ovs/var/lib/openvswitch/pki/controllerca# ls
ca.cnf  cacert.pem  careq.pem  certs  crl  crlnumber  index.txt  index.txt.attr  index.txt.
old  newcerts  private  serial  serial.old
```

c). Make OVS to use the new keys:

```
root@PICOS-OVS:/ovs/var/lib/openvswitch/pki/switchca# ovs-vsctl set-ssl /ovs/var/lib/openvswitch/pki/switchca
/sc-privkey.pem /ovs/var/lib/openvswitch/pki/switchca/sc-cert.pem /ovs/var/lib/openvswitch/pki/controllerca
/cacert.pem
```

Step 3. Copy the OVS certificate to the appropriate onos location so that onos can accept the certificate from OVS:

a). Copy "sc-cert.pem" (the OVS public key just generated in 2.1) to the ONOS host, and import it to onos.jks store with trust:

```
root@pica8:/opt/ssl# scp admin@10.10.51.140:/ovs/var/lib/openvswitch/pki/switchca/sc-cert.pem .
The authenticity of host '10.10.51.140 (10.10.51.140)' can't be established.
ECDSA key fingerprint is SHA256:3hI1TnyjNj2k7QgjCRS5xR/2wh5yxKijgwMHIuCsm0s.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.10.51.140' (ECDSA) to the list of known hosts.
admin@10.10.51.140's password:
sc-cert.
pem
     100% 4002     3.9KB/s   00:00
root@pica8:/opt/ssl# ls
cacert.pem  onos.jks  onos.p12  onos.pem  sc-cert.pem
root@pica8:/opt/ssl# keytool -importcert -file sc-cert.pem -keystore onos.jks
Enter keystore password:
Owner: CN=sc id:2018 Oct 15 16:01:09, OU=Open vSwitch certifier, O=Open vSwitch, ST=CA, C=US
Issuer: CN=OVS switchca CA Certificate (2018 Oct 15 15:56:46), OU=switchca, O=Open vSwitch, ST=CA, C=US
Serial number: 2
Valid from: Tue Oct 16 00:01:11 CST 2018 until: Fri Oct 13 00:01:11 CST 2028
Certificate fingerprints:
         MD5:  AA:53:F9:43:D4:77:07:38:87:25:DC:3D:8A:0A:19:A6
         SHA1: 45:19:64:D8:E1:A7:2A:18:CC:15:25:10:9A:C8:AE:BC:FD:53:AB:CC
         SHA256: 75:BE:13:3D:DB:8E:DC:39:D1:73:01:A8:A3:04:00:9B:D1:7B:98:84:49:85:2A:B9:5C:E9:1D:B6:F6:45:C5:E0
Signature algorithm name: MD5withRSA (weak)
Subject Public Key Algorithm: 2048-bit RSA key
Version: 1

Warning:
The input uses the MD5withRSA signature algorithm which is considered a security risk.

Trust this certificate? [no]:  yes
Certificate was added to keystore


Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry
standard format using "keytool -importkeystore -srckeystore onos.jks -destkeystore onos.jks -deststoretype
pkcs12".
root@pica8:/opt/ssl# keytool -list -keystore onos.jks
Enter keystore password:
Keystore type: jks
Keystore provider: SUN


Your keystore contains 2 entries

onos, Oct 15, 2018, PrivateKeyEntry,
Certificate fingerprint (SHA1): F9:D6:36:06:B8:5A:AE:0C:50:CE:B8:E3:DF:16:0F:9B:5C:8B:2E:7C
mykey, Oct 15, 2018, trustedCertEntry,
Certificate fingerprint (SHA1): 45:19:64:D8:E1:A7:2A:18:CC:15:25:10:9A:C8:AE:BC:FD:53:AB:CC
```

b). In the ONOS host, make sure these applications are activated:

```
onos> apps -s
......
*  21 org.onosproject.hostprovider         1.13.3   Host Location Provider
*  22 org.onosproject.lldpprovider         1.13.3   LLDP Link Provider
*  23 org.onosproject.optical-model        1.13.3   Optical Network Model
*  24 org.onosproject.openflow-base        1.13.3   OpenFlow Base Provider
*  25 org.onosproject.openflow             1.13.3   OpenFlow Provider Suite
......
*  37 org.onosproject.drivers              1.13.3   Default Drivers
......
```

c). Enable ONOS to use OFTLS by configuring "$ONOS_HOME/bin/onos-service":

```
root@pica8:/opt/onos/bin# vi onos-service
#!/bin/bash
# -----------------------------------------------------------------------------
# Starts ONOS Apache Karaf container
# -----------------------------------------------------------------------------


# uncomment the following line for performance testing
# export JAVA_OPTS="${JAVA_OPTS:--Xms16G -Xmx16G -XX:+UseConcMarkSweepGC -XX:+CMSIncrementalMode -XX:
+PrintGCDetails -XX:+PrintGCTimeStamps}"


# uncomment the following line for Netty TLS encryption
# Do modify the keystore location/password and truststore location/password accordingly
#export JAVA_OPTS="${JAVA_OPTS:--DenableNettyTLS=true -Djavax.net.ssl.keyStore=/home/ubuntu/onos.jks -Djavax.
net.ssl.keyStorePassword=222222 -Djavax.net.ssl.trustStore=/home/ubuntu/onos.jks -Djavax.net.ssl.
trustStorePassword=222222}"


export JAVA_OPTS="${JAVA_OPTS:--DenableOFTLS=true -Djavax.net.ssl.keyStore=/opt/ssl/onos.jks -Djavax.net.ssl.
keyStorePassword=111111 -Djavax.net.ssl.trustStore=/opt/ssl/onos.jks -Djavax.net.ssl.trustStorePassword=111111}"


export JAVA_OPTS="${JAVA_OPTS:--Dds.lock.timeout.milliseconds=10000}"

set -e  # exit on error
set -u  # exit on undefined variable
.................
"onos-service" 57L, 2308C written
```

c). Restart the ONOS controller.

Step 4. Testing the SSL connection:

a). On the OVS machine, we create a new bridge named br0, and set it connect to the controller by ssl:

```
admin@PICOS-OVS:/$ ovs-vsctl add-br br0 - set bridge br0 datapath_type=pica8
admin@PICOS-OVS:/$ ovs-vsctl show
5187ffd7-d781-48dc-82a6-37692344a877
    Bridge "br0"
        Port "br0"
            Interface "br0"
                type: internal
admin@PICOS-OVS:/$ ovs-vsctl set-controller br0 ssl:10.10.50.243:6653
```

b). Check the OVS machine log, You should see the following log messages:

```
admin@PICOS-OVS:~$ tail -f /tmp/log/messages
Oct 15 2018 17:39:54 Xorplus daemon.notice : 17:39:54.742|ovs|00879|rconn|INFO|br0<->ssl:10.10.50.243:6653:
connected
Oct 15 2018 17:40:06 Xorplus daemon.notice : 17:40:06.390|ovs|00880|connmgr|INFO|br0<->ssl:10.10.50.243:6653: 3
flow_mods 10 s ago (3 adds)
```

c). Check the device on ONOS host, you can see the OVS machine in the list:

```
onos> devices
id=of:1c480030ab283b4f, available=true, local-status=connected 25m15s ago, role=MASTER, type=SWITCH, mfr=Pica8,
Inc, hw=ag5648, sw=PicOS, serial=A566F6DG17AB00011, chassis=1c480030ab283b4f, driver=default, channelId=10.
10.51.140:36024, managementAddress=10.10.51.140, protocol=OF_14
```

We can see the SSL connection is successful.