Table of Contents

* [Component Certificates Redeployment Lab](https://www.opentlc.com/labs/ocp_operations/02_02_Redeploying_Component_Certificates_Lab.html#_component_certificates_redeployment_lab)
  + [1.1. Prepare Environment](https://www.opentlc.com/labs/ocp_operations/02_02_Redeploying_Component_Certificates_Lab.html#_prepare_environment)
* [1. Examine Existing Certificates](https://www.opentlc.com/labs/ocp_operations/02_02_Redeploying_Component_Certificates_Lab.html#_examine_existing_certificates)
* [2. Redeploy and Validate Certificates](https://www.opentlc.com/labs/ocp_operations/02_02_Redeploying_Component_Certificates_Lab.html#_redeploy_and_validate_certificates)
  + [2.1. Configure Bastion Host **oc**Command with new CA Certificates](https://www.opentlc.com/labs/ocp_operations/02_02_Redeploying_Component_Certificates_Lab.html#_configure_bastion_host_code_oc_code_command_with_new_ca_certificates)
* [3. Configure Web Browser](https://www.opentlc.com/labs/ocp_operations/02_02_Redeploying_Component_Certificates_Lab.html#_configure_web_browser)
  + [3.1. Copy Intermediate CA Certificate](https://www.opentlc.com/labs/ocp_operations/02_02_Redeploying_Component_Certificates_Lab.html#_copy_intermediate_ca_certificate)
  + [3.2. Configure Your Web Browser](https://www.opentlc.com/labs/ocp_operations/02_02_Redeploying_Component_Certificates_Lab.html#_configure_your_web_browser)
  + [3.3. Test Intermediate CA Certificate](https://www.opentlc.com/labs/ocp_operations/02_02_Redeploying_Component_Certificates_Lab.html#_test_intermediate_ca_certificate)
* [4. Summary](https://www.opentlc.com/labs/ocp_operations/02_02_Redeploying_Component_Certificates_Lab.html#_summary)

Component Certificates Redeployment Lab

This lab depends up the first lab, "OpenShift Certificate Authorities Redeployment Lab." If that is not complete, this lab will fail.

Components of the OpenShift system have their own certificate authorities (CAs). In the previous lab, you updated the Platform CA with the Intermediate CA certificate. The system continued to function because deploying the new CA certificates affects only new server certificates signed by this CA, not existing ones.

In this lab, you redeploy the certificates of the rest of the OpenShift cluster. New certificates are generated—each from their own CA certificates appropriate for their component type. You see that some CA certificates change, while other do not.

Finally, you make the Intermediate CA certificate available to your client tools, i.e., your web browser and **oc** command line tools. Without the Intermediate CA certificate available to the client tools, they have no way of verifying the certificate presented by OpenShift. The tools would then present a certificate verification error when accessing the OpenShift, and then prompt you to accept unknown certificates. With the Intermediate CA certificate available to your client tools, the certificates presented by OpenShift can be verified successfully, and the there are no certificate verification errors.

1.1. Prepare Environment

1. Log in to your OpenShift Bastion host and switch to the **root** user.
   * If this was a high availability environment, the OpenShift Master API service would be accessed through the load balancer.
2. Find the proper hostname and port of the Master API from the **openshift\_master\_cluster\_hostname** variable in the Ansible inventory file, and put it in a shell variable for ease of use:

MASTER\_API\_HOSTPORT=$(sed -n 's/openshift\_master\_cluster\_hostname=//p' /etc/ansible/hosts):443; echo ${MASTER\_API\_HOSTPORT}

1. Do the same for the router on the infrastructure node—the port number is **443**:

ROUTER=$(perl -ne 's/^(infranode1\..\*?\.internal).\*$/$1/ and print and last' /etc/ansible/hosts):443; echo ${ROUTER}

1. Set an environment variable with your environment’s GUID.

export GUID=$(hostname | cut -f2 -d.); echo ${GUID}

1. Examine Existing Certificates

1. Examine the existing server certificates using **openssl s\_client** to connect to the master:

echo QUIT | openssl s\_client -connect ${MASTER\_API\_HOSTPORT} 2>&1 | more

**Sample Output**

Thu Sep 7 11:11:49 2017

depth=1 CN = openshift-signer@1504795784

verify error:num=19:self signed certificate in certificate chain

verify return:0s

CONNECTED(00000003)

---

Certificate chain

0 s:/CN=172.30.0.1

i:/CN=openshift-signer@1504795784

1 s:/CN=openshift-signer@1504795784

i:/CN=openshift-signer@1504795784

---

Server certificate

-----BEGIN CERTIFICATE-----

*### -------8<-snip-8<-------*

-----END CERTIFICATE-----

subject=/CN=172.30.0.1

issuer=/CN=openshift-signer@1504795784

---

Acceptable client certificate CA names

/CN=openshift-signer@1504795784

Server Temp Key: ECDH, prime256v1, 256 bits

* + Note that the "Subject" field of the **ca.crt** certificate in the previous lab matches the certificates in the **Certificate chain**section.

1. Examine the router’s certificates:

echo QUIT | openssl s\_client -connect ${ROUTER} 2>&1 | more

**Sample Output**

Certificate chain

0 s:/CN=router.default.svc

i:/CN=openshift-service-serving-signer@1504795784

1 s:/CN=openshift-service-serving-signer@1504795784

i:/CN=openshift-service-serving-signer@1504795784

---

Server certificate

-----BEGIN CERTIFICATE-----

<OMITTED>

-----END CERTIFICATE-----

subject=/CN=router.default.svc

issuer=/CN=openshift-service-serving-signer@1504795784

---

No client certificate CA names sent

Server Temp Key: ECDH, prime256v1, 256 bits

* + Note that the router has a different CA certificate and does *not* express any client CA names.

2. Redeploy and Validate Certificates

1. Run the **redeploy-certificates.yml** Ansible Playbook:

time ansible-playbook -i /etc/ansible/hosts -v -f 20 /usr/share/ansible/openshift-ansible/playbooks/byo/openshift-cluster/redeploy-certificates.yml

* + This command takes about six minutes to complete.

1. Now that the certificates are redeployed, examine the certificates offered by the OpenShift Master API:

echo QUIT | openssl s\_client -connect ${MASTER\_API\_HOSTPORT} 2>&1 | less

**Sample Output**

Certificate chain

0 s:/CN=172.30.0.1

i:/C=US/ST=North Carolina/O=Red Hat, Inc./OU=GPTE DevOps/CN=Red Hat OpenTLC Classroom Intermediate CA/emailAddress=gpte-devops-automation@redhat.com

1 s:/C=US/ST=North Carolina/O=Red Hat, Inc./OU=GPTE DevOps/CN=Red Hat OpenTLC Classroom Intermediate CA/emailAddress=gpte-devops-automation@redhat.com

i:/C=US/ST=North Carolina/L=Raleigh/O=Red Hat, Inc./OU=GPTE DevOps/CN=Red Hat OpenTLC Classroom Root CA/emailAddress=gpte-devops-automation@redhat.com

---

Server certificate

-----BEGIN CERTIFICATE-----

-----END CERTIFICATE-----

subject=/CN=172.30.0.1

issuer=/C=US/ST=North Carolina/O=Red Hat, Inc./OU=GPTE DevOps/CN=Red Hat OpenTLC Classroom Intermediate CA/emailAddress=gpte-devops-automation@redhat.com

---

Acceptable client certificate CA names

/C=US/ST=North Carolina/O=Red Hat, Inc./OU=GPTE DevOps/CN=Red Hat OpenTLC Classroom Intermediate CA/emailAddress=gpte-devops-automation@redhat.com

/CN=openshift-signer@1504795784

Server Temp Key: ECDH, prime256v1, 256 bits

* + Expect the certificate chain in the output to be significantly different from the certificate chain displayed earlier.
  + Expect to see that the Intermediate certificate of the Platform CA signed the server certificate.

2.1. Configure Bastion Host **oc** Command with new CA Certificates

By depoying new CA keys and certificates, the certificate on the Bastion host for the **system:admin** user is now incorrect. The certificate for the **system:admin** user was regenerated on the master. It must be copied to the Bastion host in order to continue using the **oc** occmand on the bastion host.

1. Execute the following steps on the Bastion host to put the proper configuration file with the **system:admin** certificates in place.
2. cd /root/
3. mv /root/.kube/ /root/.oldkube
4. mkdir /root/.kube/
5. ssh master1.${GUID}.internal 'sudo cp /etc/origin/master/admin.kubeconfig /home/ec2-user/; sudo chown ec2-user /home/ec2-user/admin.kubeconfig'

scp master1.${GUID}.internal:/home/ec2-user/admin.kubeconfig /root/.kube/config

|  |  |
| --- | --- |
|  | Use **sudo** to copy the file to the home directory and change its permissions. Then **scp** the file to the proper place on your bastion. |

1. Run the **oc** command once from the Bastion, while indicating the proper CA certificate.

oc login -u user1 -p 'r3dh4t1!' --certificate-authority=/root/certs/ca/intermediate/certs/intermediate.cert.pem

Further **oc** commands on the bastion host root user, even logouts and logins, will not require the **--certificate-authortiy=**option.

3. Configure Web Browser

3.1. Copy Intermediate CA Certificate

1. On the bastion host, copy the Intermediate certificate to your user’s home directory in order to **scp** it to your laptop.
2. sudo -i
3. cp /root/certs/ca/intermediate/certs/intermediate.cert.pem /home/<OpenTLC username>/

chown <OpenTLC username> /home/<OpenTLC username>/intermediate.cert.pem

1. On your laptop, download the following file via scp from your bastion to your laptop using **scp**:

scp -i <Your OpenTLC private key> <OpenTLC username>@bastion.${GUID}.example.opentlc.com:intermediate.cert.pem .

* + This is the Intermediate CA certificate that signed most of OpenShift’s server certificates. Note the location it was downloaded.

3.2. Configure Your Web Browser

Select the following directions that reflect your client platform:

1. Firefox 57 on Fedora
2. Chromium on Fedora
3. Firefox 57 on Mac
4. Chrome on Mac

3.2.1. Configure Firefox Web Browser CA on Linux Fedora 27

1. In Firefox, navigate to **about:preferences#privacy**.
2. Scroll down the page, and select **View Certificates → Authorities**:
3. Click **Import** and select the Intermediate CA certificate file from your download location.
4. Check the **Trust this CA to identify websites** box and click **OK.**
5. Click **OK** again to close the certificates box.
6. The Intermediate CA Certificate is now properly imported into Firefox 57+ on Fedora. You may now proceed to Step 3.3 which tests the Intermediate CA Certificate.

3.2.2. Configure Chromium Web Browser CA on Linux Fedora 27

In this section, you add the Intermediate CA certificate to your Linux operating system’s Chromium browser’s certificate management component.

1. In Chromium, navigate to **chrome://settings/?search=Manage%20certificates** and click **Manage Certificates**.
2. Select **Authorities → Import**.
3. From the file selection dialog box, select the Intermediate CA certificate file and click **Open**.
4. Tick the checkbox that indicates **Trust thie certificate for identifying websites**, and click **OK**.
5. You should see the Red Hat OpenTLC Classroom Intermediate CA" in the list of trusted authorities as depicted here:
6. The Intermediate CA Certificate is now properly imported into Chromium on Fedora. You may now proceed to Step 3.3 which tests the Intermediate CA Certificate.

3.2.3. Configure Firefox Web Browser CA on Mac

1. In Firefox, navigate to **about:preferences#privacy**.
2. Scroll down the page, and select **View Certificates → Authorities**:
3. Click **Import** and select the Intermediate CA certificate file from your download location.
4. Check the **Trust this CA to identify websites** box and click **OK.**
5. Click **OK** again to close the certificates box.
6. The Intermediate CA Certificate is now properly imported into Firefox 57+ on Mac. You may now proceed to Step 3.3 which tests the Intermediate CA Certificate.

3.2.4. Configure Chrome Web Browser CA on Mac

In this section, you add the Intermediate CA certificate to your Mac operating system’s certificate management application.

1. In Chrome, navigate to **chrome://settings/?search=Manage%20certificates** and click **Manage Certificates**.
   * Expect your operating system’s certificate management box to appear.

|  |  |
| --- | --- |
|  | The instructions that follow are for the macOS operating system. |

1. Import the Intermediate CA file into the System keychain. Click the **System** keychain, and then click the padlock above **Click to unlock the System keychain**:
2. Enter your password and click **Modify Keychain**.
3. At the bottom of the box, click the section "Certificates" and then the **+** to add the certificate.
4. From the file selection dialog box, select the Intermediate CA certificate file and click **Open**.
5. If prompted for your macOS password, enter it to import the certificate.
6. Select the name of the new Intermediate CA from the list and type **command+I** to invoke the **Get Info** dialog for this certificate.
7. Click the triangle to expand the **Trust** section.
8. Select the list box next to **Secure Sockets Layer (SSL)** and set it to **Always Trust**, and close the window.
9. Enter your password again if prompted.
10. Your Keychain Access application should include the following highlighted line:
11. Click the **lock** icon to lock the keychain. Enter your password if required.
12. Close the **Keychain Access** window.
13. The Intermediate CA Certificate is now properly imported into Chrome on Mac. You may now proceed to Step 3.3 which tests the Intermediate CA Certificate.

3.3. Test Intermediate CA Certificate

1. Browse to your master’s public hostname at **https://master.$GUID.example.opentlc.com/**, making sure to substitute your GUID for the **$GUID** in the URL.
   * Note that you do not get a certificate error. You are able to log in and browse without warnings or errors.

4. Summary

In this lab we ran through the very typical activities of an OpenShift consultant at a security-aware customer:

* Redployed Component Certificates using the ansible playbook
* Validated the redeployed certificates were indeed the certificate we intended
* Configured our web browsers to work with the new Intermediate CA Certificate

Build Version: be32934b41da1e34c27e21e79bdf57d30c732a3e : Last updated 2018-01-19 05:59:04 EST