Managing Cisco NFVI Security

The following topics describe Cisco NFVI network and application security and best practices.

Verifying Management Node Network Permissions

The Cisco NFVI management node stores sensitive information related to Cisco NFVI operations. Access to the management node can be restricted to requests coming from IP addresses known to be used by administrators. The administrator source networks is configured in the setup file, under **[NETWORKING]** using the **admin_source_networks** parameter. To verify this host based firewall setting, log into the management node as an admin user and list the rules currently enforces by iptables. Verify that the source networks match the values configured. If no source networks have been configured, then all source traffic is allowed. However, note that only traffic destined to ports with known admin services is allowed to pass. The **admin_source_networks** value can be set at install time or changed through a reconfigure.

```
[root@j11-control-server-1 ~]# iptables -list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     icmp --  anywhere             anywhere
ACCEPT     tcp  --  10.0.0.0/8           anywhere            tcp dpt:ssh
ACCEPT     tcp  --  172.16.0.0/12        anywhere            tcp dpt:ssh
ACCEPT     tcp  --  10.0.0.0/8           anywhere            tcp dpt:https
ACCEPT     tcp  --  172.16.0.0/12        anywhere            tcp dpt:https
ACCEPT     tcp  --  10.0.0.0/8           anywhere            tcp dpt:4979
ACCEPT     tcp  --  172.16.0.0/12        anywhere            tcp dpt:4979
ACCEPT     tcp  --  10.0.0.0/8           anywhere            tcp dpt:esmagent
```

```
ACCEPT     tcp  --  172.16.0.0/12        anywhere             tcp dpt:esmagent
ACCEPT     tcp  --  10.0.0.0/8           anywhere             tcp dpt:8008
ACCEPT     tcp  --  172.16.0.0/12        anywhere             tcp dpt:8008
ACCEPT     tcp  --  10.0.0.0/8           anywhere             tcp dpt:copy
ACCEPT     tcp  --  172.16.0.0/12        anywhere             tcp dpt:copy
ACCEPT     tcp  --  10.0.0.0/8           anywhere             tcp dpt:22250
ACCEPT     tcp  --  172.16.0.0/12        anywhere             tcp dpt:22250
ACCEPT     all  --  anywhere             anywhere             state RELATED,ESTABLISHED
DROP       all  --  anywhere             anywhere
```

Verifying Management Node File Permissions

The Cisco NFVI management node stores sensitive information related to Cisco NFVI operations. These files are secured by strict file permissions. Sensitive files include secrets.yaml, openrc, *.key, and *.pem. To verify the file permissions, log into the management node as an admin user and list all of the files in the ***~/openstack-configs/*** directory. Verify that only the owner has read and write access to these files. For example:

```
[root@j11-control-server-1 ~]# ls -l ~/openstack-configs
total 172
-rw-------. 1 root root  3272 Jun 21 17:57 haproxy.key
-rw-------. 1 root root  5167 Jun 21 17:57 haproxy.pem
-rw-------. 1 root root   223 Aug  8 18:09 openrc
-rw-------. 1 root root   942 Jul  6 19:44 secrets.yaml

[…]
```

Viewing Administrator Access Attempts

Because the UCS servers are part of the critical Cisco NFVI infrastructure, Cisco recommends monitoring administrator login access periodically. To view the access attempts, use the **journalctl** command to view the log created by sshd. For example:

```
[root@control-server-1 ~]# journalctl –u sshd
-- Logs begin at Tue 2016-06-21 17:39:35 UTC, end at Mon 2016-08-08 17:25:06 UTC. --
Jun 21 17:40:03 hh23-12 systemd[1]: Started OpenSSH server daemon.
Jun 21 17:40:03 hh23-12 systemd[1]: Starting OpenSSH server daemon...
Jun 21 17:40:03 hh23-12 sshd[2393]: Server listening on 0.0.0.0 port 22.
Jun 21 17:40:03 hh23-12 sshd[2393]: Server listening on :: port 22.
Jun 21 17:40:43 hh23-12 sshd[12657]: Connection closed by 171.70.163.201 [preauth]
Jun 21 17:41:13 hh23-12 sshd[12659]: Accepted password for root from 171.70.163.201 port
40499
Jun 21 17:46:41 hh23-12 systemd[1]: Stopping OpenSSH server daemon...
Jun 21 17:46:41 hh23-12 sshd[2393]: Received signal 15; terminating.
Jun 21 17:46:41 hh23-12 systemd[1]: Started OpenSSH server daemon.
Jun 21 17:46:41 hh23-12 systemd[1]: Starting OpenSSH server daemon...
Jun 21 17:46:41 hh23-12 sshd[13930]: Server listening on 0.0.0.0 port 22.
Jun 21 17:46:41 hh23-12 sshd[13930]: Server listening on :: port 22.
Jun 21 17:50:45 hh23-12 sshd[33964]: Accepted password for root from 171.70.163.201 port
40545
Jun 21 17:56:36 hh23-12 sshd[34028]: Connection closed by 192.168.212.20 [preauth]
Jun 21 17:57:08 hh23-12 sshd[34030]: Accepted publickey for root from 10.117.212.20 port
62819
Jun 22 16:42:40 hh23-12 sshd[8485]: Invalid user user1 from 10.117.212.20
Jun 22 16:42:40 hh23-12 sshd[8485]: input_userauth_request: invalid user user1 [preauth]
s
```

Verifying SELinux

To minimize the impact of a security breach on a Cisco NFVI server, the Cisco VM enables SELinux (Security Enhanced Linux) to protect the server resources. To validate that SELinux is configured and running in enforcing mode, use the **sestatus** command to view the status of SELinux and verify that its status is enabled and in enforcing mode. For example:

```
[root@mgmt1 ~]# /usr/sbin/sestatus -v
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   enforcing
Mode from config file:          permissive
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Max kernel policy version:      28
```

Validating Port Listening Services

To prevent access by unauthorized users and processes, Cisco NFVI has no extra services listening on network ports. To verify this, use the netstat -plnt command to get a list of all services listening on the node and verify that no unauthorized services are listening. For example:

```
[root@j11-control-server-1 ~]# netstat -plnt
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address     State       PID/Program
name
tcp        0      0 23.23.4.101:8776        0.0.0.0:*           LISTEN      24468/python2
tcp        0      0 23.23.4.101:5000        0.0.0.0:*           LISTEN      19874/httpd
tcp        0      0 23.23.4.101:5672        0.0.0.0:*           LISTEN      18878/beam.smp

tcp        0      0 23.23.4.101:3306        0.0.0.0:*           LISTEN      18337/mysqld
tcp        0      0 127.0.0.1:11211         0.0.0.0:*           LISTEN      16563/memcached
tcp        0      0 23.23.4.101:11211       0.0.0.0:*           LISTEN      16563/memcached
tcp        0      0 23.23.4.101:9292        0.0.0.0:*           LISTEN      21175/python2
tcp        0      0 23.23.4.101:9999        0.0.0.0:*           LISTEN      28555/python
tcp        0      0 23.23.4.101:80          0.0.0.0:*           LISTEN      28943/httpd
tcp        0      0 0.0.0.0:4369            0.0.0.0:*           LISTEN      18897/epmd

tcp        0      0 127.0.0.1:4243          0.0.0.0:*           LISTEN      14673/docker

tcp        0      0 0.0.0.0:22              0.0.0.0:*           LISTEN      2909/sshd

tcp        0      0 23.23.4.101:4567        0.0.0.0:*           LISTEN      18337/mysqld

tcp        0      0 23.23.4.101:15672       0.0.0.0:*           LISTEN      18878/beam.smp

tcp        0      0 0.0.0.0:35672           0.0.0.0:*           LISTEN      18878/beam.smp

tcp        0      0 127.0.0.1:25            0.0.0.0:*           LISTEN      4531/master

tcp        0      0 23.23.4.101:35357       0.0.0.0:*           LISTEN      19874/httpd

tcp        0      0 23.23.4.101:8000        0.0.0.0:*           LISTEN      30505/python

tcp        0      0 23.23.4.101:6080        0.0.0.0:*           LISTEN      27996/python2
```

```
tcp        0      0 23.23.4.101:9696        0.0.0.0:*               LISTEN      22396/python2
tcp        0      0 23.23.4.101:8004        0.0.0.0:*               LISTEN      30134/python
tcp        0      0 23.23.4.101:8773        0.0.0.0:*               LISTEN      27194/python2
tcp        0      0 23.23.4.101:8774        0.0.0.0:*               LISTEN      27194/python2
tcp        0      0 23.23.4.101:8775        0.0.0.0:*               LISTEN      27194/python2
tcp        0      0 23.23.4.101:9191        0.0.0.0:*               LISTEN      20752/python2
tcp6       0      0 :::9200                 :::*                    LISTEN      18439/xinetd
tcp6       0      0 :::4369                 :::*                    LISTEN      18897/epmd
tcp6       0      0 :::22                   :::*                    LISTEN      2909/sshd
tcp6       0      0 ::1:25                  :::*                    LISTEN      4531/master
```

Validating Non-Root Users for OpenStack Services

To prevent unauthorized access, Cisco NFVI runs OpenStack processes as a non-root user. To verify OpenStack processes are not running as root, use the **ps** command to get a list of all node processes. In the following example the user is 162:

```
[root@j11-control-server-1 ~]# ps -aux | grep nova-api
162      27194  0.6  0.0 360924 132996 ?       S    Aug08  76:58 /usr/bin/python2
/usr/bin/nova-api
162      27231  0.0  0.0 332192 98988 ?        S    Aug08   0:01 /usr/bin/python2
/usr/bin/nova-api
162      27232  0.0  0.0 332192 98988 ?        S    Aug08   0:01 /usr/bin/python2
/usr/bin/nova-api
162      27233  0.0  0.0 332192 98988 ?        S    Aug08   0:01 /usr/bin/python2
/usr/bin/nova-api
```

Verifying Password Strength

Password strength is critical to Cisco NFVI security. Cisco NFVI passwords can be generated in one of two ways during installation:

- The Cisco NFVI installer generates unique passwords automatically for each protected service.
- You can provide an input file containing the passwords you prefer.

Cisco-generated passwords will be unique, long, and contain a mixture of uppercase, lowercase, and numbers. If you provide the passwords, password strength will be your responsibility. You can view the passwords by displaying the secrets.yaml file. For example:

```
[root@mgmt1 ~]# cat ~/openstack-configs/secrets.yaml
ADMIN_USER_PASSWORD: QaZ12n13wvvNY7AH
CINDER_DB_PASSWORD: buJL8pAfytoJ0Icm
CINDER_KEYSTONE_PASSWORD: AYbcB8mx6a5Ot549
CLOUDPULSE_KEYSTONE_PASSWORD: HAT6vbl7Z56yZLtN
COBBLER_PASSWORD: bax8leYFyyDon0ps
CPULSE_DB_PASSWORD: aYGSzURpGChztbMv
DB_ROOT_PASSWORD: bjb3Uvwus6cvaNe5
```

```
ELK_PASSWORD: c50e57Dbm7LF0dRV
[…]
```

Reconfiguring Passwords and OpenStack Configurations

Note This topic does not apply if you installed the optional Cisco Virtual Topology System. For information about use of passwords when VTS is installed, see the *Installing Cisco VTS* section in the *Cisco NFV Infrastructure 2.0 Installation Guide*.

You can reset some configurations after installation including the OpenStack service password and debugs, TLS certificates, ELK configurations, and collectd intervals. Two files, secrets.yaml and openstack_config.yaml, located in : /root/installer-{tag id}/openstack-configs/, contain the passwords, debugs, TLS file location, ELK and collectd configurations. Also, Elasticsearch uses disk space for the data that is sent to it. These files can grow in size, and Cisco VIM has configuration variables that establishes the frequency and file size under which they will be rotated.

The Cisco VIM installer dynamically generates the OpenStack service and database passwords with 16 alphanumeric characters and stores those in /root/openstack-configs/secrets.yaml. You can change the OpenStack service and database passwords using the password reconfigure command on the deployed cloud. The command identifies the containers affected by the password change and restarts them so the new password can take effect. Always schedule password reconfigurations in a maintenance window because container restarts might disrupt the control plane. You can list the password and configuration that can be changed using following:

```
[root@mgmt1 ~]# cd ~/installer-xxxx
[root@mgmt1 installer-xxxx]# ciscovimclient/ciscovim help reconfigure
usage: ciscovim reconfigure [--regenerate_secrets] [--setpassword <secretkey>]
                            [--setopenstackconfig <option>]

Reconfigure the openstack cloud
Optional arguments:
  --regenerate_secrets           Regenerate All Secrets
  --setpassword <secretkey>      Set of secret keys to be changed.
  --setopenstackconfig <option>  Set of Openstack config to be changed.
[root@mgmt1 ~]#ciscovimclient/ciscovim list-openstack-configs
+-------------------------------+----------------------------------------+
|              Name             |                 Option                 |
+-------------------------------+----------------------------------------+
|      CINDER_DEBUG_LOGGING     |                 False                  |
|     KEYSTONE_DEBUG_LOGGING    |                 False                  |
|   CLOUDPULSE_VERBOSE_LOGGING  |                  True                  |
|     MAGNUM_VERBOSE_LOGGING    |                  True                  |
|       NOVA_DEBUG_LOGGING      |                  True                  |
|    NEUTRON_VERBOSE_LOGGING    |                  True                  |
|      external_lb_vip_cert     |  /root/openstack-configs/haproxy.pem   |
|     GLANCE_VERBOSE_LOGGING    |                  True                  |
| COLLECTD_RECONFIGURE_interval |                   30                   |
|     elk_rotation_frequency    |                monthly                 |
|   CEILOMETER_VERBOSE_LOGGING  |                  True                  |
|     elk_rotation_del_older    |                   10                   |
|       HEAT_DEBUG_LOGGING      |                 False                  |
|    KEYSTONE_VERBOSE_LOGGING   |                  True                  |
|     external_lb_vip_cacert    | /root/openstack-configs/haproxy-ca.crt |
|      MAGNUM_DEBUG_LOGGING     |                  True                  |
|     CINDER_VERBOSE_LOGGING    |                  True                  |
|       elk_rotation_size       |                   2                    |
```

```
|    CLOUDPULSE_DEBUG_LOGGING   |                 False                 |
|     NEUTRON_DEBUG_LOGGING     |                  True                 |
|      HEAT_VERBOSE_LOGGING     |                  True                 |
|    CEILOMETER_DEBUG_LOGGING   |                 False                 |
|      GLANCE_DEBUG_LOGGING     |                 False                 |
|      NOVA_VERBOSE_LOGGING     |                  True                 |
+-------------------------------+---------------------------------------+
[root@mgmt1 installer-xxxx]#
[root@mgmt1 installer-xxxx]# ciscovimclient/ciscovim list-password-keys
+----------------------------------+
| Password Keys                    |
+----------------------------------+
| COBBLER_PASSWORD                 |
| CPULSE_DB_PASSWORD               |
| DB_ROOT_PASSWORD                 |
| ELK_PASSWORD                     |
| GLANCE_DB_PASSWORD               |
| GLANCE_KEYSTONE_PASSWORD         |
| HAPROXY_PASSWORD                 |
| HEAT_DB_PASSWORD                 |
| HEAT_KEYSTONE_PASSWORD           |
| HEAT_STACK_DOMAIN_ADMIN_PASSWORD |
| HORIZON_SECRET_KEY               |
| KEYSTONE_ADMIN_TOKEN             |
| KEYSTONE_DB_PASSWORD             |
| METADATA_PROXY_SHARED_SECRET     |
| NEUTRON_DB_PASSWORD              |
| NEUTRON_KEYSTONE_PASSWORD        |
| NOVA_DB_PASSWORD                 |
| NOVA_KEYSTONE_PASSWORD           |
| RABBITMQ_ERLANG_COOKIE           |
| RABBITMQ_PASSWORD                |
| WSREP_PASSWORD                   |
+----------------------------------+
[root@mgmt1 installer-xxxx]#
```

You can change specific password and configuration identified from the available list. The password and configuration values can be supplied on the command line as follows:

```
[root@mgmt1 ~]# ciscovimclient/ciscovim help reconfigure
usage: ciscovim reconfigure [--regenerate_secrets] [--setpassword <secretkey>]
                            [--setopenstackconfig <option>]

Reconfigure the Openstack cloud

Optional arguments:
  --regenerate_secrets           Regenerate All Secrets
  --setpassword <secretkey>      Set of secret keys to be changed.
  --setopenstackconfig <option>  Set of Openstack config to be changed.
[root@mgmt1 ~]# ciscovimclient/ciscovim reconfigure --setpassword
ADMIN_USER_PASSWORD,NOVA_DB_PASSWORD --setopenstackconfig
HEAT_DEBUG_LOGGING,HEAT_VERBOSE_LOGGING
Password for ADMIN_USER_PASSWORD:
Password for NOVA_DB_PASSWORD:
Enter T/F for option HEAT_DEBUG_LOGGING:T
Enter T/F for option HEAT_VERBOSE_LOGGING:T
```

The supplied password must be alphanumeric chars and can be maximum of 32 characters in length. Below are the available configuration parameters for OpenStack:

Configuration Parameter	Allowed Values
CEILOMETER_DEBUG_LOGGING	T/F (True or False)

CEILOMETER_VERBOSE_LOGGING	T/F (True or False)
CINDER_DEBUG_LOGGING	T/F (True or False)
CINDER_VERBOSE_LOGGING	T/F (True or False)
CLOUDPULSE_DEBUG_LOGGING	T/F (True or False)
CLOUDPULSE_VERBOSE_LOGGING	T/F (True or False)
GLANCE_DEBUG_LOGGING	T/F (True or False)
GLANCE_VERBOSE_LOGGING	T/F (True or False)
HEAT_DEBUG_LOGGING	T/F (True or False)
HEAT_VERBOSE_LOGGING	T/F (True or False)
KEYSTONE_DEBUG_LOGGING	T/F (True or False)
KEYSTONE_VERBOSE_LOGGING	T/F (True or False)
MAGNUM_DEBUG_LOGGING	T/F (True or False)
MAGNUM_VERBOSE_LOGGING	T/F (True or False)
NEUTRON_DEBUG_LOGGING	T/F (True or False)
NEUTRON_VERBOSE_LOGGING	T/F (True or False)
NOVA_DEBUG_LOGGING	T/F (True or False)
NOVA_VERBOSE_LOGGING	T/F (True or False)
COLLECTD_RECONFIGURE_interval	Collectd metric gathering interval (seconds)
elk_rotation_del_older	Days after which older logs will be purged
elk_rotation_frequency	Available options: "daily", "weekly", "fortnightly", "monthly"
elk_rotation_size	Gigabytes (entry of type float/int is allowed)
external_lb_vip_cacert	Location of HAProxy CA certificate
external_lb_vip_cert	Location of HAProxy certificate

Alternatively, you can dynamically regenerate all passwords using regenerate_secrets command option as follows:

```
[root@mgmt1 ~]# cd ~/installer-xxxx
[root@mgmt1 ~]# ./ciscovimclient/ciscovim reconfigure --regenerate_secrets
```

In addition to the services passwords, you can change the debug and verbose options for Heat, Glance, Cinder, Nova, Neutron, Keystone and Cloudpulse in /root/openstack-configs/openstack_config.yaml. Other configurations you can modify include ELK configuration parameters, collectd intervals, API and Horizon TLS certificates, and RootCA. , and admin source networks. When reconfiguring these options (for example

TLS), always remember that some control plane downtime will occur, so plan the changes during maintenance windows. The command to reconfigure these elements is:

```
./ciscovimclient/ciscovim reconfigure
```

The command includes a built-in validation to ensure you do not enter typos in the secrets.yaml or openstack_config.yaml files.

When reconfiguration of password or enabling of openstack-services fails, all subsequent pod management operations will be blocked. In this case, it is recommended to contact Cisco TAC to resolve the situation.

Enabling NFVIMON Post Pod Install

Cisco VIM 2.0 can be optionally installed with a 3rd party software (aka NFVIMON), that can monitor the health and performance of the NFV infrastructure. The NFVIMON feature enables extensive monitoring and performance data for various components of the cloud infrastructure including Cisco UCS blade and rack servers, service profiles, Nexus top of rack switches, fabric connections and also the OpenStack instances. The monitoring system is designed such that it can monitor single or multiple Pods from a single management system. NFVIMON can be enabled by extending the setup_data.yaml with relevant information on an existing pod, via the reconfigure option. NFVIMON consists of 4 components: dispatcher, collector, Resource Manager (RM) and control-center with Cisco Zenpacks (CC). Since NVIFMON is a 3rd party software, care has been taken to make sure its integration into VIM is loosely coupled and the VIM automation only deals with installing the minimal software piece (dispatcher) needed to monitor the pod. The installing of the other NFVIMON components (collector, Resource Manager (RM) and control-center with Cisco Zenpacks (CC)), are Cisco Advance Services led activity and those steps are outside the scope of the current install guide.

Before you Begin

Please ensure that you have engaged with Cisco Advance Services on the planning and installation of the NFVIMON accessories along with its network requirements. Also, the image information of collector, Resource Manager (RM) and control-center with Cisco Zenpacks (CC)) is available only through Cisco Advance Services. At a high level, please have a node designated to host a pair of collector VM for each pod, and a common node to host CC and RM VMs, which can aggregate and display monitoring information from multiple pods. In terms of networking, the collectors VMs need to have 2 interfaces: an interface in br_mgmt of the VIM, and another interface that is routable, which can reach the VIM Installer REST API and the RM VMs. Since the collector VM is sitting in an independent node, 4 IPs from the management network of the pod should be pre-planned and reserved. Install steps of the collector, Resource Manager (RM) and control-center with Cisco Zenpacks (CC)) are Cisco Advance Services led activity.

Installation of NFVIMON Dispatcher

The dispatcher is the only component in NFVIMON offering that is managed by VIM orchestrator. While the dispatcher acts as a conduit to pass openstack information of the pod to the collectors, it is the Cisco NFVI Zenpack sitting in the CC/RM node, that gathers the node level information. To enable dispatcher as part of the VIM Install, update the setup_data with the following information:

```
#Define the PODNAME
PODNAME: <PODNAME with no space>; ensure that this is unique across all the pods
NFVIMON:
  MASTER:                  # Master Section
    admin_ip: <IP address of Control Centre VM>
  COLLECTOR:               # Collector Section
```

```
management_vip: <VIP for ceilometer/dispatcher to use> #Should be unique across the VIM
Pod; Should be part of br_mgmt network
    Collector_VM_Info:
      -
        hostname: <hostname of Collector VM 1>
        password: <password_for_collector_vm1>  # max length of 32
       ccuser_password: <password from master for 'ccuser' (to be used for self monitoring)>
  # max length of 32
        admin_ip: <ssh_ip_collector_vm1> # Should be part of br_api network
        management_ip: <mgmt_ip_collector_vm1> # Should be part of br_mgmt network
      -
        hostname: <hostname of Collector VM 2>
        password: <password_for_collector_vm2>  # max length of 32
       ccuser_password: <password from master for 'ccuser' (to be used for self monitoring)>
  # max length of 32
        admin_ip: <ssh_ip_collector_vm2> # Should be part of br_api network
        management_ip: <mgmt_ip_collector_vm2> # Should be part of br_mgmt network
  DISPATCHER:
    rabbitmq_username: admin  # Pod specific user for dispatcher module in
ceilometer-collector
```

To monitor TOR, ensure that the following TORSWITCHINFO sections are defined in the setup_data.yaml.

```
TORSWITHCINFO:
  SWITCHDETAILS:
 -
      hostname: <switch_a_hostname>:     # Mandatory for NFVIMON if switch monitoring is
needed
      username: <TOR switch username>    # Mandatory for NFVIMON if switch monitoring is
needed
      password: <TOR switch password>    # Mandatory for NFVBENCH; Mandatory for NFVIMON
if switch monitoring is needed
      ssh_ip: <TOR switch ssh ip>        # Mandatory for NFVIMON if switch monitoring is
needed
      ....
  -
      hostname: <switch_b_hostname>:     # Mandatory for NFVIMON if switch monitoring is
needed
      username: <TOR switch username>    # Mandatory for NFVIMON if switch monitoring is
needed
      password: <TOR switch password>    # Mandatory for NFVIMON if switch monitoring is
needed
      ssh_ip: <TOR switch ssh ip>        # Mandatory for NFVIMON if switch monitoring is
needed
      ....
```

To initiate the integration of NFVIMON on an existing pod, copy the setupdata into a local dir and update it manually with information listed above, then run reconfiguration command as follows:

```
[root@mgmt1 ~]# cd /root/
[root@mgmt1 ~]# mkdir MyDir
[root@mgmt1 ~]# cd MyDir
[root@mgmt1 ~]# cp /root/openstack-configs/setup_data.yaml <my_setup_data.yaml>
[root@mgmt1 ~]# vi my_setup_data.yaml (update the setup_data to include NFVIMON related
info)
[root@mgmt1 ~]# cd ~/installer-xxxx
[root@mgmt1 ~]# ./ciscovimclient/ciscovim --setupfile ~/MyDir/<my_setup_data.yaml> reconfigure
```

It should be noted that un-configuration of this feature is not supported today. Additionally, NFVIMON is only supported on a pod running keystone v2.

Fernet Key Operations

Keystone fernet token format is based on the cryptographic authentication method - Fernet. Fernet is an implementation of Symmetric Key Encryption. Symmetric key encryption is a cryptographic mechanism that uses the same cryptographic key to encrypt plaintext and the same cryptographic key to decrypt ciphertext. Fernet authentication method also supports multiple keys where it takes a list of symmetric keys, performs all encryption using the first key in a list and attempts to decrypt using all the keys from that list.

The Cisco NFVI pods uses Fernet keys by default. The following operations can be carried out in Cisco NFVI pods.

To check if the fernet keys are successfully synchronized across the keystone nodes.

```
[root@mgmt1 ~]# cd ~/installer-xxxx
[root@mgmt1 ~]# ./ciscovimclient/ciscovim help check-fernet-keys
usage: ciscovim check-fernet-keys

Check whether the fernet keys are successfully synchronized across keystone nodes.
```

To set the fernet key frequency:

```
[root@mgmt1 ~]# cd ~/installer-xxxx
[root@mgmt1 ~]# ./ciscovimclient/ciscovim help period-rotate-fernet-keys
usage: ciscovim period-rotate-fernet-keys <SET_PERIOD_ROTATION_FERNET_KEYS>
Set the frequency of fernet keys rotation on keystone
Positional arguments:
  <SET_PERIOD_ROTATION_FERNET_KEYS>
Frequency to set for period rotation
```

To forcefully rotate the fernet keys:

```
[root@mgmt1 ~]# cd ~/installer-xxxx
[root@mgmt1 ~]# ./ciscovimclient/ciscovim help rotate-fernet-keys
usage: ciscovim rotate-fernet-keys
Trigger rotation of the fernet keys on keystone
```

To resync the fernet keys across the keystone nodes:

```
[root@mgmt1 ~]# cd ~/installer-xxxx
[root@mgmt1 ~]# ./ciscovimclient/ciscovim help resync-fernet-keys
usage: ciscovim resync-fernet-keys
Resynchronize the fernet keys across all the keystone nodes
```

Managing Certificates

When TLS protection is configured for the OpenStack APIs, the two certificate files, haproxy.pem and haproxy-ca.crt, are stored in the /root/openstack-configs/ directory. Clients running on servers outside of the deployed cloud to verify cloud authenticity need a copy of the root certificate (haproxy-ca.crt). If a well-known certificate authority has signed the installed certificate, no additional configuration is needed on client servers. However, if a self-signed or local CA is used, copy haproxy-ca.crt to each client. Following instructions specific to the client operating system or browser to install the certificate as a trusted certificate.

Alternatively, you can explicitly reference the certificate when using the OpenStack CLI by using the environment variable OS_CACERT or command line parameter –cacert.

While Cisco NFVI is operational, a daily check is made to monitor the expiration dates of the installed certificates. If certificates are not nearing expiration, an informational message is logged. As the certificate approaches expiration, an appropriate warning or critical message is logged.

```
2017-04-24T13:56:01 INFO Certificate for OpenStack Endpoints at 192.168.0.2:5000 expires
in 500 days
```

It is important to replace the certificates before they expire. After Cisco NFVI is installed, you can update the certificates by replacing the haproxy.pem and haproxy-ca.crt files and running the reconfigure command:

```
cd ~/installer-xxxx; ./ciscovimclient/ciscovim reconfigure
```

Reconfiguring TLS Certificates

Cisco VIM provides a way to configure TLS certificates on-demand for any reason. For Example: certificate expiration policies governing certificate management.

Reconfiguration of certificates in general is supported in the following components:

- Cisco VIM Rest API endpoints:
  - Steps to be performed to reconfigure certificate and key file are as follows:
  - Operator will copy the new key, CA root and certificate files into the ~/openstack-configs folder under the following filenames

    ```
    cp <new-ca-root-cert> ~/openstack-configs/mercury-ca.crt
    cp <new-key-file> ~/openstack-configs/mercury.key
    cp <new-cert-file> ~/openstack-configs/mercury.crt
    ```

  - Once copied run the reconfigure steps as under:

    ```
    cd ~/installer-xxxx/tools
    ./restapi.py -a reconfigure-tls
    ```

- SwiftStack Service through Horizon and CinderBackup Service.
  - Reconfiguring TLS certificates for SwiftStack mainly involves client side certificate updates. The CA root certificate in both these cases is updated for components within OpenStack that are clients of the SwiftStack service in general.
  - Copy the new CA root certificate to the ~/openstack-configs folder and run reconfigure.

    ```
    cp <new-ca-root-cert> ~/openstack-configs/haproxy-ca.crt
     cd ~/installer-xxxx; ./ciscovimclient/ciscovim reconfigure
    ```

- Logstash service and Logstash forwarder (client-side certificates).
  - For the Logstash service on the management node, both the key and certificate file will be reconfigured as part of the reconfigure operation.
  - For the Logstash forwarder service on the controllers, compute and storage nodes, the certificate file will be reconfigured as part of the reconfigure operation.
  - Copy of the key and certificate files to the ~/openstack-configs folder on the management node and run reconfigure operation.

    ```
    cp <new-key-file> ~/openstack-configs/logstash-forwarder.key
     cp <new-cert-file> ~/openstack-configs/logstash-forwarder.crt
    cd ~/installer-xxxx; ./ciscovimclient/ciscovim reconfigure
    ```

Enabling Keystone v3 on an Existing Install

To continue enhancing our security portfolio, and multi-tenancy with the use of domains, Keystone v3 support has been added in Cisco VIM 2.0 from an authentication end-point. It should be noted that Keystone v2 and v3 are mutually exclusive; i.e. the administrator has to decide during install time the authentication end-point version to go with. By default, VIM orchestrator picks keystone v2 as the authentication end-point. So one can enable Keystonev3 as an install option on day-0 (see 2.0 CiscoVIM install guide), or enable it as a reconfigure option after the pod is installed. To enable Keystone v3 after the pod is installed, one needs to define the following under the optional service section in the setup_data.yam filel.

```
# Optional Services:
OPTIONAL_SERVICE_LIST:
- keystonev3
```

To initiate the integration of Keystone v3 on an existing pod, copy the setupdata into a local dir and update it manually, then run reconfiguration command as follows:

```
[root@mgmt1 ~]# cd /root/
[root@mgmt1 ~]# mkdir MyDir
[root@mgmt1 ~]# cd MyDir
[root@mgmt1 ~]# cp /root/openstack-configs/setup_data.yaml <my_setup_data.yaml>
[root@mgmt1 ~]# vi my_setup_data.yaml (update the setup_data to include keystone v3 info)
[root@mgmt1 ~]# cd ~/installer-xxxx
[root@mgmt1 ~]# ./ciscovimclient/ciscovim --setupfile ~/MyDir/<my_setup_data.yaml> reconfigure
```

It should be noted that un-configuration of this feature is not supported today. Additionally, re-versioning Keystone API from v3 to v2 is also not supported.

LDAP support with Keystone v3

With the introduction of KeystoneV3, the openstack service authentication can now we delegated to an external LDAP server. In Cisco VIM 2.0, this feature has been introduced optionally if the authorization is done by Keystone v3. Just like Keystonev3, this feature can be enabled on an existing pod running Cisco VIM 2.0. To avail of this feature post pod deployment, the setup_data needs to be augmented with the following information during the installation of the pod.

An important pre-requisite for enabling LDAP integration is that the LDAP endpoint MUST be reachable from all the Controller nodes that run OpenStack Keystone Identity Service.

```
LDAP:
  domain: <Domain specific name>
  user_objectclass: <objectClass for Users> # e.g organizationalPerson
  group_objectclass: <objectClass for Groups> # e.g. groupOfNames
  user_tree_dn: '<DN tree for Users>' # e.g. 'ou=Users,dc=cisco,dc=com'
  group_tree_dn: '<DN tree for Groups>' # e.g. 'ou=Groups,dc=cisco,dc=com'
  suffix: '<suffix for DN>' # e.g. 'dc=cisco,dc=com'
  url: '<ldap:// host:port>' # e.g. 'ldap://172.26.233.104:389'
  user: '<DN of bind user>' # e.g. 'dc=admin,dc=cisco,dc=com'
  password: <password> # e.g. password of bind user
```

To initiate the integration of LDAP with Keystone v3 on an existing pod, copy the setupdata into a local dir and update it manually with the relevant LDAP and Keystone v3 (if absent from before) configuration, then run reconfiguration command as follows:

```
[root@mgmt1 ~]# cd /root/
[root@mgmt1 ~]# mkdir MyDir
[root@mgmt1 ~]# cd MyDir
[root@mgmt1 ~]# cp /root/openstack-configs/setup_data.yaml <my_setup_data.yaml>
```

```
[root@mgmt1 ~]# vi my_setup_data.yaml (update the setup_data to include LDAP info)
[root@mgmt1 ~]# cd ~/installer-xxxx
[root@mgmt1 ~]# ./ciscovimclient/ciscovim --setupfile ~/MyDir/<my_setup_data.yaml> reconfigure
```

The reconfigure feature supports a full or partial reconfiguration of the LDAP integration service.

Note All the parameters within the LDAP stanza are configurable with the exception of the domain parameter.

Integrating identity with LDAP over TLS: The automation supports keystone integration with LDAP over TLS. In order to enable TLS, the CA root certificate must be presented as part of the /root/openstack-configs/haproxy-ca.crt file. The url parameter within the LDAP stanza must be set to ldaps.

Additionally, the url parameter supports the following formats: *<ldaps | ldap>://<FQDN | IP-Address>:[port]*.

The protocol can be ldap for non-ssl OR ldaps if tls is to be enabled.

The ldap host can be a fully-qualified domain name or an ip address depending on how the SSL certificates were generated.

The port number is optional and if not provided assumes that the ldap services are running on the default ports. For Example:. 389 for non-ssl and 636 for ssl. However, if these are not the defaults, then the non-standard port numbers must be provided.