openLI

Provisioner Configuration
OpenLI Training: Chapter Seven

Shane Alcock
University of Waikato
New Zealand
shane.alcock@waikato.ac.nz

The Provisioner

- Central orchestrator for OpenLI
  - All other components connect to the provisioner
  - Receive instructions

- First goal: get the provisioner listening for client connections

The Provisioner

- Get a shell on your provisioner container

```
$ docker exec -i -t openli-provisioner /bin/bash

root@cb04f35b37df:/home/openli-prov#
```

The Provisioner

- Query the container's IP address on the openli-lab network

```
# ip addr list eth1

844: eth1@if845: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
state UP group default
    link/ether 02:42:ac:13:00:03 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.19.0.3/16 brd 172.19.255.255 scope global eth0
    valid_lft forever preferred_lft forever
```

The Provisioner

● Query the container's IP address on the openli-lab network

```
# ip addr list eth1

844: eth1@if845: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
state UP group default
    link/ether 02:42:ac:13:00:03 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.19.0.3/16 brd 172.19.255.255 scope global eth0
    valid_lft forever preferred_lft forever
```

Configuration

- Open the provisioner configuration file in a text editor
  - I use vim, but any editor will work fine

```
# vim /etc/openli/provisioner-config.yaml

clientport: <COLLECTORPORT>
clientaddr: <PROVIP>
updateport: <RESTAPIPORT>
updateaddr: <PROVIP>
mediationport: <MEDIATORPORT>
mediationaddr: <PROVIP>

intercept-config-file: /etc/openli/running-intercept-config.yaml
```

WAND

Configuring Listening IPs

● Replace all instances of <PROVIP> with the correct IP address

```
# vim /etc/openli/provisioner-config.yaml

clientport: <COLLECTORPORT>
clientaddr: 172.19.0.3
updateport: <RESTAPIPORT>
updateaddr: 172.19.0.3
mediationport: <MEDIATORPORT>
mediationaddr: 172.19.0.3

intercept-config-file: /etc/openli/running-intercept-config.yaml
```

Configuring Listening IPs

- Replace all instances of <PROVIP> with the correct IP address
  - Remember this IP for later!

```
# vim /etc/openli/provisioner-config.yaml

clientport: <COLLECTORPORT>
clientaddr: 172.19.0.3
updateport: <RESTAPIPORT>
updateaddr: 172.19.0.3
mediationport: <MEDIATORPORT>
mediationaddr: 172.19.0.3

intercept-config-file: /etc/openli/running-intercept-config.yaml
```

WAND

Configuring Ports

- Choose 3 distinct ports for listening services
  - One for the REST API
  - One for collectors to connect to
  - One for mediators to connect to

Configuring Ports

- Choose 3 distinct ports for listening services
  - One for the REST API -- **8080**
  - One for collectors to connect to -- **9001**
  - One for mediators to connect to -- **9002**

- Remember these for later too!

Configuring Ports

- Update the provisioner config with the port numbers

```
# vim /etc/openli/provisioner-config.yaml

clientport: <COLLECTORPORT>
clientaddr: 172.19.0.3
updateport: <RESTAPIPORT>
updateaddr: 172.19.0.3
mediationport: <MEDIATORPORT>
mediationaddr: 172.19.0.3

intercept-config-file: /etc/openli/running-intercept-config.yaml
```

Configuring Ports

- Update the provisioner config with the port numbers

```
# vim /etc/openli/provisioner-config.yaml

clientport: <COLLECTORPORT>
clientaddr: 172.19.0.3
updateport: <RESTAPIPORT>
updateaddr: 172.19.0.3
mediationport: <MEDIATORPORT>
mediationaddr: 172.19.0.3

intercept-config-file: /etc/openli/running-intercept-config.yaml
```

WAND

Configuring Ports

- Update the provisioner config with the port numbers

```
# vim /etc/openli/provisioner-config.yaml

clientport: 9001
clientaddr: 172.19.0.3
updateport: 8080
updateaddr: 172.19.0.3
mediationport: 9002
mediationaddr: 172.19.0.3

intercept-config-file: /etc/openli/running-intercept-config.yaml
```

WAND

Configuration Complete!

- We're done -- save and exit your text editor

Logging with rsyslog

- Copy the provided rsyslog config to `/etc/rsyslog.d/`
  - Restart rsyslog service
  - OpenLI logs will now appear in `/var/log/openli/`

```
# cp /etc/openli/rsyslog.d/10-openli-provisioner.conf /etc/rsyslog.d/

THIS STEP IS NEEDED ONLY WHEN USING THE LAB CONTAINER
# stop_rsyslog.sh

# service rsyslog restart
```

WAND

Starting the Provisioner

- Now we can start the provisioner service
  - Examine logs for any obvious error messages

```
# service openli-provisioner start

 * Starting OpenLI provisioner /etc/openli/provisioner-config.yaml      [ OK
]


# less /var/log/openli/provisioner.log
```

Starting the Provisioner

- Example of "good" looking logs

```
openliprovisioner[22662]: OpenLI: not using OpenSSL TLS for internal communications
openliprovisioner[22662]: OpenLI provisioner: REST API does NOT require authentication
openliprovisioner[22662]: OpenLI: provisioner intercepting RTP comfort noise for all VOIP
intercepts
openliprovisioner[22662]: OpenLI: provisioner listening on 172.19.0.2:9001 successfully.
openliprovisioner[22662]: OpenLI: incoming mediator listening on 172.19.0.2:9002
successfully.
openliprovisioner[22662]: OpenLI: update socket listening on 172.19.0.2:8080 successfully.
```

WAND

Starting the Provisioner

- Example of logs where an error has occurred

```
openliprovisioner[22701]: OpenLI: not using OpenSSL TLS for internal communications
openliprovisioner[22701]: OpenLI provisioner: REST API does NOT require authentication
openliprovisioner[22701]: OpenLI: provisioner intercepting RTP comfort noise for all VOIP
intercepts
openliprovisioner[22701]: OpenLI: Error while trying to bind provisioner listening socket:
Cannot assign requested address.
openliprovisioner[22701]: OpenLI: Error, could not start listening socket.
```

Stopping the Provisioner

- Normally, you would use systemd to halt the service
  - Can't normally stop processes with systemd inside docker
  - I've added a script to the container to do it for you

```
DO THIS NORMALLY
# service openli-provisioner stop

ON THE LAB CONTAINER, DO THIS INSTEAD
# stop_provisioner.sh
```

Next Time

- Configuring the Mediator