openLI

The OpenLI Training Lab

OpenLI Training: Chapter Six

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

Goals

- Sandbox environment for experimenting with OpenLI
  - Docker containers for each component
  - A fake LEA that will receive and output intercepts
  - No special hardware required, just a Linux server or laptop
  - Does NOT require a real network tap

Goals

- Practice
  - Basic configuration of each OpenLI component
  - Starting the components
  - Checking logs for errors
  - Using the REST API to connect to an LEA
  - Using the REST API to provision new intercepts

Goals

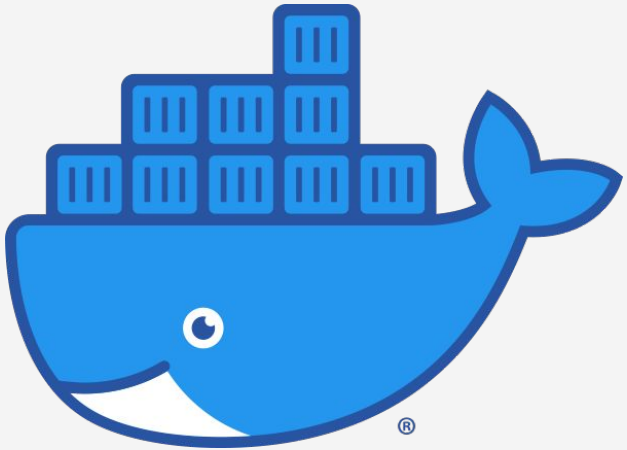
- Observe
  - Replay specific traffic patterns on demand
  - See interception results on the provided “fake” LEA
  - Experiment with pcaps from your own network

Goals

- Expand
  - Introduce more advanced capabilities
  - TLS, authentication, message brokering
  - IPSec tunnels for agency connections

Docker

- Virtualised containers
  - Isolated
  - Lightweight
  - Shareable
  - Open source



Docker

- Installing Docker for Linux
  - https://docs.docker.com/engine/install/
  - https://docs.docker.com/engine/install/linux-postinstall/

Build the Training Lab

- Download the latest version of the lab setup script
  - Install git first, if required

```
$ git clone https://github.com/wanduow/openli-training-lab.git
```

Build the Training Lab

- Run the setup script
  - Read the script first, just to be safe!

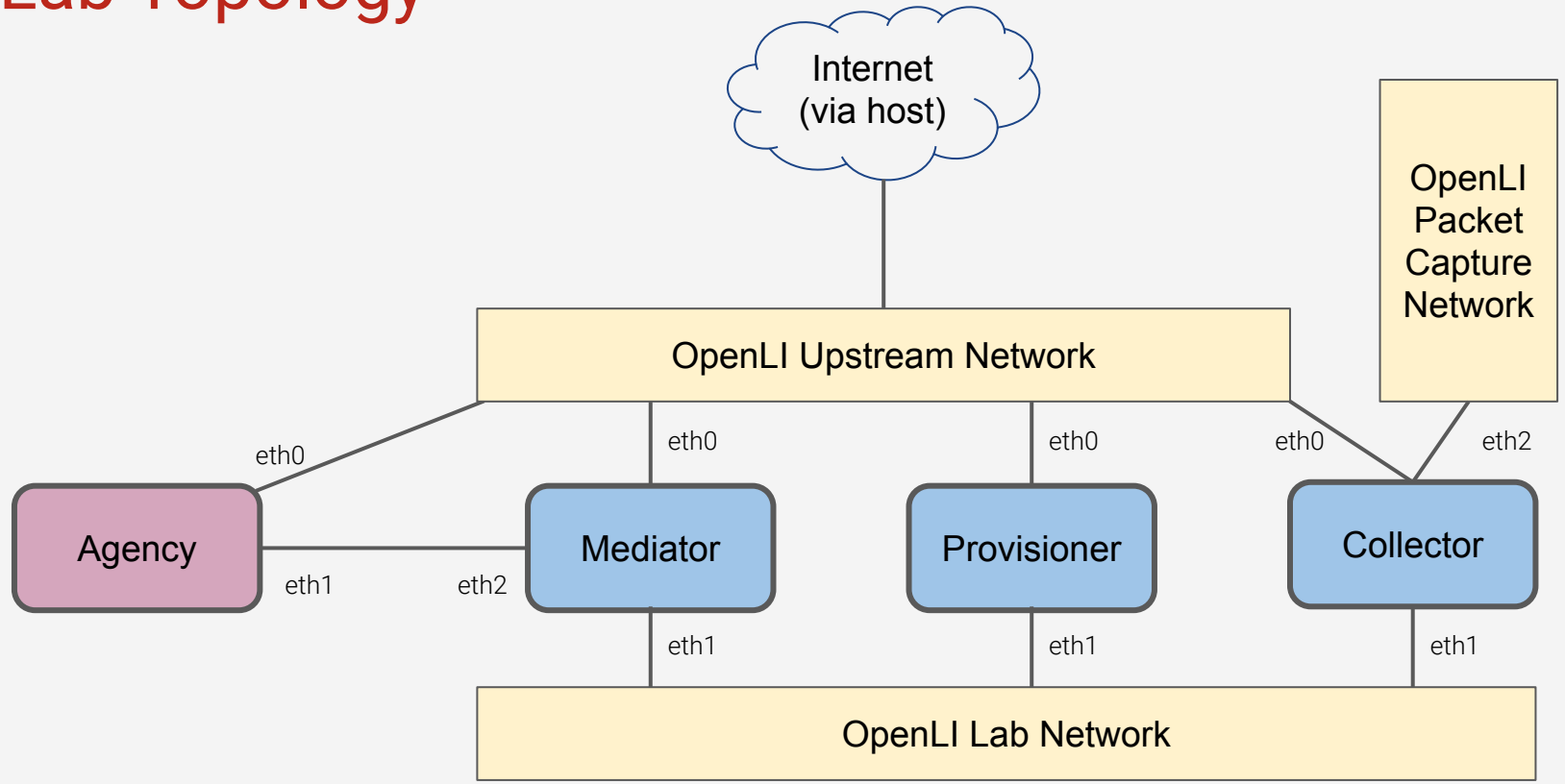
```
$ cd openli-training-lab
$ ./setup.sh
```

Build the Training Lab

- Four containers are now running on your host
  - Provisioner
  - Collector
  - Mediator
  - A “pretend” LEA

- OpenLI packages are already installed

Lab Topology



Topology Notes

- eth0: access to upstream Internet
  - e.g. for installing software packages
- eth1: inter-component communication
- eth2 on mediator: direct connection to the LEA
- eth2 on collector: interface for packet capture

Container Login

- Use `docker exec` to run a shell on a container
  - Container names are:
    - `openli-provisioner`
    - `openli-collector`
    - `openli-mediator`
    - `openli-agency`

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

root@627ad9000dc2:/home/openli-testagency#
```

Container Removal

- Use `docker stop` to halt a container
  - Once stopped, any local changes to the container will be lost

- Running setup.sh again will stop and recreate the containers
  - Clean slate approach

```
$ docker stop openli-agency
```

Next Time

- Configuring the Provisioner