BINUS UNIVERSITY INTERNATIONAL

Linux Network Namespace and Virtual Switch

Guest Lecture – Pertamina University

Session 1

Ardimas Purwita
ardimas.andi@binus.ac.id

BINUS UNIVERSITY INTERNATIONAL

Goals

1st

- Linux Network Namespace
- Virtual Switch

2nd

- Openflow
- SDN

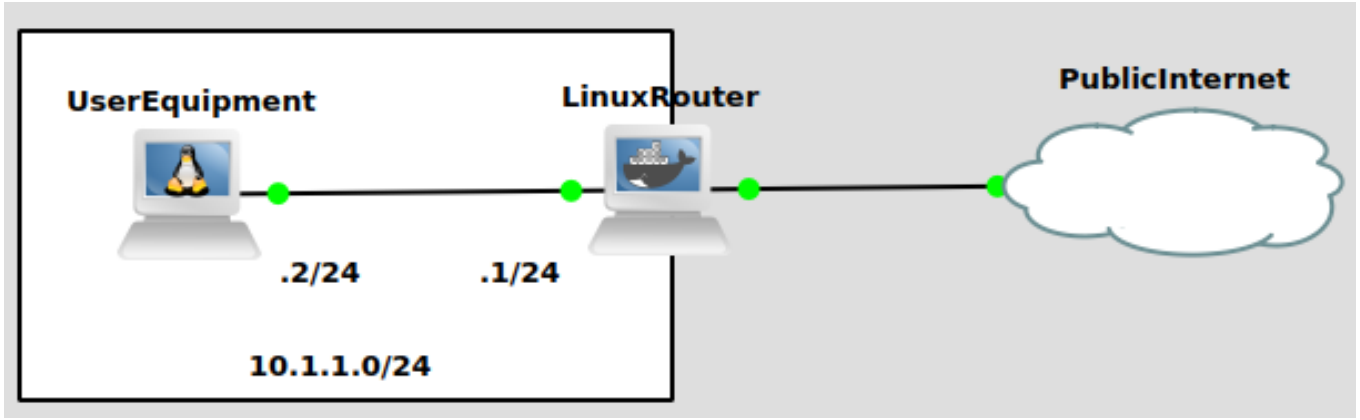
BINUS UNIVERSITY INTERNATIONAL

Objectives

- Familiarize ourselves toward each other
- Familiarize ourselves with the tech
- Obtaining high level intuition for next week's SDN topic

Preamble

- Do you know that your old Linux laptop can be used as a router?
- Now, what can a typical router do?
  - Forwarding
  - NAT



Commands

- To check the forwarding capability of a Linux OS

  sysctl –a | grep –i forward

- Set SNAT

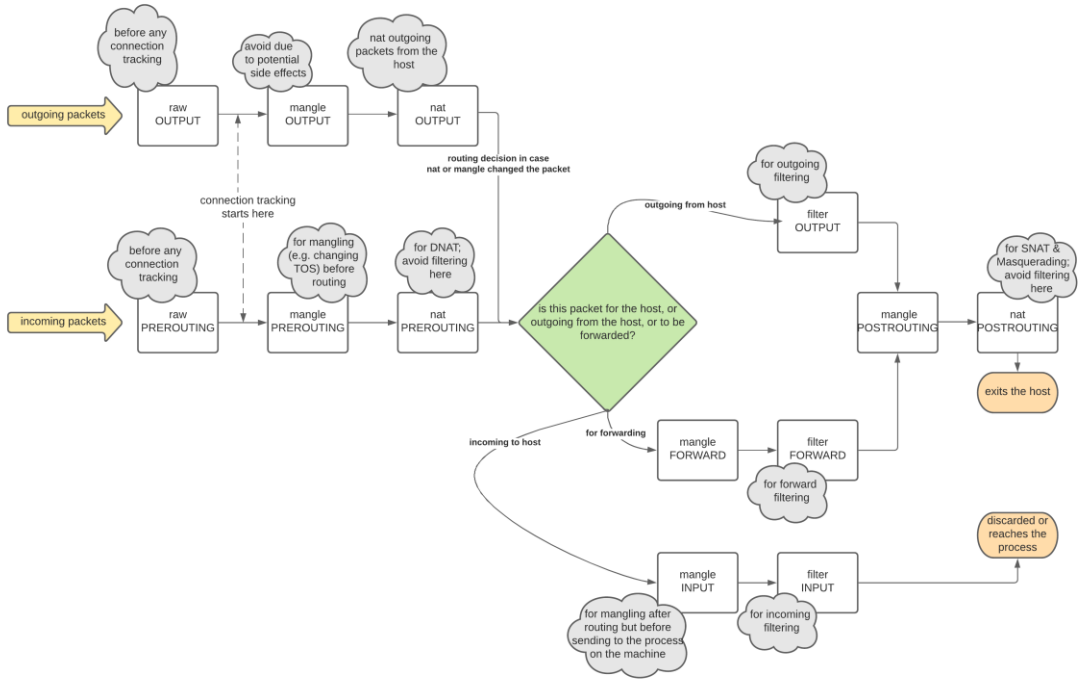
  sudo iptables –t nat –A POSTROUTING –s 10.1.1.0/24 –j MASQUERADE

- To listen to incoming traffic

  tcpdump –i eth0

- To block

  sudo iptables –t filter –A FORWARD –d example.com –i eth0 –j DROP



BINUS UNIVERSITY INTERNATIONAL

Linux Network Namespace and Virtual Ethernet

Try answering the Linux netns and veth questions

See: https://github.com/ardimasp/gl-pu-compnet

Solution: A chain of three netns

```
# create namespace red
ip netns add red
# create veth
ip link add veth-r2g type veth peer name veth-g2r
# attach veth
ip link set veth-r2g netns red
# activate veth
ip netns exec red ip link set veth-r2g up
# assign ip address
ip netns exec red ip a add 192.168.1.2/24 dev veth-r2g


# set routing table
ip netns exec red ip route add 10.1.1.2/32 via 192.168.1.1
```

```
ip netns add green
ip link set veth-g2r netns green
ip netns exec green ip link set dev veth-g2r up
ip netns exec green ip a add 192.168.1.1/24 dev veth-g2r


ip link set veth-g2b netns green
ip netns exec green ip link set dev veth-g2b up
ip netns exec green ip a add 10.1.1.1/24 dev veth-g2b
```

```
ip netns add blue
ip link add veth-b2g type veth peer name veth-g2b
ip link set veth-b2g netns blue
ip netns exec blue ip link set dev veth-b2g up
ip netns exec blue ip a add 10.1.1.2/24 dev veth-b2g


ip netns exec blue ip route add 192.168.1.2/32 via 10.1.1.1
```

Virtual Switch

Try answering the virtual switch questions

Solution: v-bridge

```
#!/bin/sh

# create namespaces
ip netns add red
ip netns add green
ip netns add blue

# create veths
ip link add veth-red type veth peer name veth-red-br
ip link add veth-green type veth peer name veth-green-br
ip link add veth-blue type veth peer name veth-blue-br

# create a virtual bridge to simulate a switch
ip link add v-bridge type bridge

# attach the veths
ip link set veth-red netns red
ip link set veth-red-br master v-bridge
```

```
ip link set veth-green netns green
ip link set veth-green-br master v-bridge

ip link set veth-blue netns blue
ip link set veth-blue-br master v-bridge

# set ip addr inside namespace
ip netns exec red ip addr add 10.1.1.2/24 dev veth-red
ip netns exec green ip addr add 10.1.1.3/24 dev veth-green
ip netns exec blue ip addr add 10.1.1.4/24 dev veth-blue

# set ip addr for the v-bridge
ip addr add 10.1.1.1/24 dev v-bridge
```

```
# activate all interfaces
ip link set dev v-bridge up
ip link set dev veth-red-br up
ip link set dev veth-green-br up
ip link set dev veth-blue-br up
ip netns exec red ip link set dev veth-red up
ip netns exec green ip link set dev veth-green up
ip netns exec blue ip link set dev veth-blue up

# set ip route table
ip netns exec red ip route add default via 10.1.1.1 dev veth-red
ip netns exec green ip route add default via 10.1.1.1 dev veth-gree
ip netns exec blue ip route add default via 10.1.1.1 dev veth-blue

# set nat
iptables -t nat -A POSTROUTING -s 10.1.1.0/24 -j MASQUERADE
```