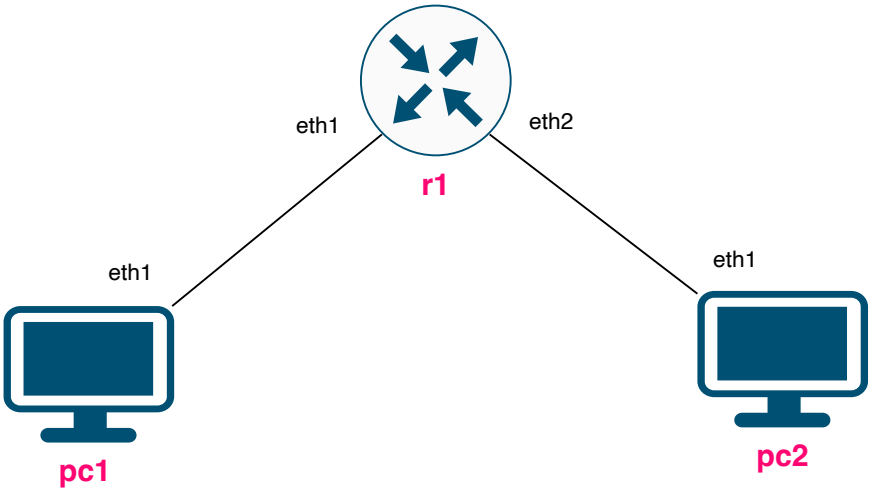
S2E2 - tcpdump

Nicholas Morrison nick@nanocat.net

Topology



Connecting to the lab server

Connect to the lab server:

```
local$ ssh-keygen -R netlab.nanocat.net
local$ ssh lab@netlab.nanocat.net
Password: (see discord)
```

Connect to your router:

```
lab@netlab$ list-devices
lab@netlab$ connect DEVICE
```

Goal

- understand network namespaces
- understand tcpdump
- understand wireshark

About Linux network namespaces

- network namespaces (netns)
- Linux feature since 2002 (linux kernel version 2.4)
- groups and segregates network interfaces
- used in container contexts, like containerlab!
- `ip netns list` lists the existing network namespaces
  - use `ip netns list | grep pod5` to filter for your pod
- `ip netns exec NAMESPACE [some_command]` executes a command in that namespace
- `ip netns exec NAMESPACE ip -c link` lists interfaces in a namespace
- `ip netns exec NAMESPACE ip -c address` lists interfaces and their addresses in a namespace

About tcpdump

- venerable tool, around since 1988
- uses libpcap, same as `wireshark`
- not usually installed by default
- add it to your linux with `apt install tcpdump` (or your local equiv)
- installed by default in our cEOS and PC containers
- we will use `ip netns` to capture from OUTSIDE our containers
- cheat sheet here https://cdn.comparitech.com/wp-content/uploads/2019/06/tcpdump-cheat-sheet-1.pdf

About wireshark

- GUI packet analysis tool
- very powerful filter system

Initial configuration

- by default, r1 acts as a switch (all ports untagged in VLAN 1)
  - so no configuration is required for r1
- configure pc1 with 192.168.0.1/24
- configure pc2 with 192.168.0.2/24

Configure the PCs

Configure with:

```
ip address add 192.168.0.1/24 dev eth1
ip route delete default
```

Check the config with:

```
ip address
ip route
```

Delete a mistake if you need to

```
ip address delete 192.168.0.1/24 dev eth1
```

Open at least two terminals

You'll need two terminals for the next bit.

Generate some packets

In one terminal, `ping 192.168.0.2` from pc1.

In the other terminal, *don't connect to a device*, but stay "outside" in your shell.

Using tcpdump and netns

In your other terminal window, execute tcpdump from inside the pc1 network namespace.

Example:

```
ip netns exec clab-pod1-pc1 ip -c link
```

```
ip netns exec clab-pod1-pc1 tcpdump -i eth1
```

Produce some different packet types

Stop your ping on pc1 with `ctrl-c` and execute:

```
ping 192.168.0.111
```

What kind of packets do you see in your tcpdump?

Capture some packets to disk

Capture 20 packets:

```
ip netns exec clab-podX-pc1 tcpdump -i eth1 -c 20 -w CLEVER_FILENAME.pcap
```

Copy them to your local computer. Locally,

```
scp lab@netlab.nanocat.net:CLEVER_FILENAME.pcap .
```

Open this file locally in Wireshark.