Lab 10 NET

This repository contains exercises for the BSc – BSY Laboratory. The exercises focus on networking and routing using the `ip` utility, `nftables`, and namespaces.

Prerequisites

- A ZHAW VPN session
- Any modern web browser
- Any modern SSH client application
- OpenStack Horizon dashboard: https://ned.cloudlab.zhaw.ch
- OpenStack account details
- Username to login with SSH into VMs in ned.cloudlab.zhaw.ch OpenStack cloud from your laptops

Exercises

Task 1 – Setup & Basic Tasks

We want to build a networking situation using netspaces on a single Openstack VM. To do this we need to:

1. Ensure `nftables` is installed. Type `nft --version`. If `nft` is not installed then do so using `sudo apt install nftables`.
2. Clone the appropriate repo using your ZHAW username/password and: `git clone https://github.zhaw.ch/InES-RT-Ethernet/bsy-lab-ip-route-students.git`.
3. In the sub-directory `/netenv` call `sudo bash netenv setup`. This will set up a networking environment with three IP stacks running, each in its own namespace.

Task 2 – Routing

Subtask 2.1 – Setting up the routing

Enter the router namespace and - using the `ip` utility - examine the routing table. The `ip` utility features a number of sub-utilities, for instance `link`, `route`, `address` (addr), `monitor`.

Question: Using the ping utility, are the client and server reachable?

Answer: Yes, the client and server are reachable in the same network. However, the router and server in a different network are not reachable because routing tables are missing.

Subtask 2.2 – Running the application

On the server machine, issue the command: `./srvr/bin/srvr`. On the client machine, issue the command: `./clnt/bin/clnt`. The client part of the round-trip application automatically stops after the incrementing number has reached a value of 1000. If desired, the client can just be restarted and the sequence starts over again.

Task 3 – Understanding the structure of netfilters and nft

Subtask 3.1 – Understanding the structure of netfilters and nft

In the man page of nft, compare the hooks mentioned in the “Address Families” section with the diagram above. Relate the address families and hooks from the man page to the diagram above.

Question: Trace the path of the client UDP frame through the netfilter. Where would be a good place to set up a counter counting all frames entering the netfilter? Where would be a good place to set up counters for the following:

1. Counting all UDP frames entering the IP layer from an external port
2. Counting all UDP frames coming from an application
3. Counting the sum of frames from these two sources?

Answer:

1. To count all UDP frames entering the IP layer from an external port, you can set up a counter in the PREROUTING hook. This hook is triggered before any routing decisions are made, so it allows you to capture all incoming packets, including UDP

frames. By incrementing the counter for each UDP packet, you can keep track of the total count.

2. To count all UDP frames coming from an application, you can set up a counter in the OUTPUT hook. This hook is triggered when a packet is generated by the local system. By incrementSorry for the cut-off. Here's the continuation:

```
2. To count all UDP frames coming from an application, you can set up a
counter in the OUTPUT hook. This hook is triggered when a packet is
generated by the local system. By incrementing the counter for each UDP
packet, you can keep track of the total count generated by the applications.
3. To count the sum of frames from the two sources mentioned above, you can
maintain separate counters in the PREROUTING and OUTPUT hooks as described.
Then, you can periodically add the counts from these two counters to
calculate the total count.

## Assessment

No assessment foreseen for this lab session.

## Time

The entire session will take 90 minutes.

## Additional Documentation

OpenStack Horizon documentation can be found on the following pages:

- User Guide: https://docs.openstack.org/horizon/latest/
```