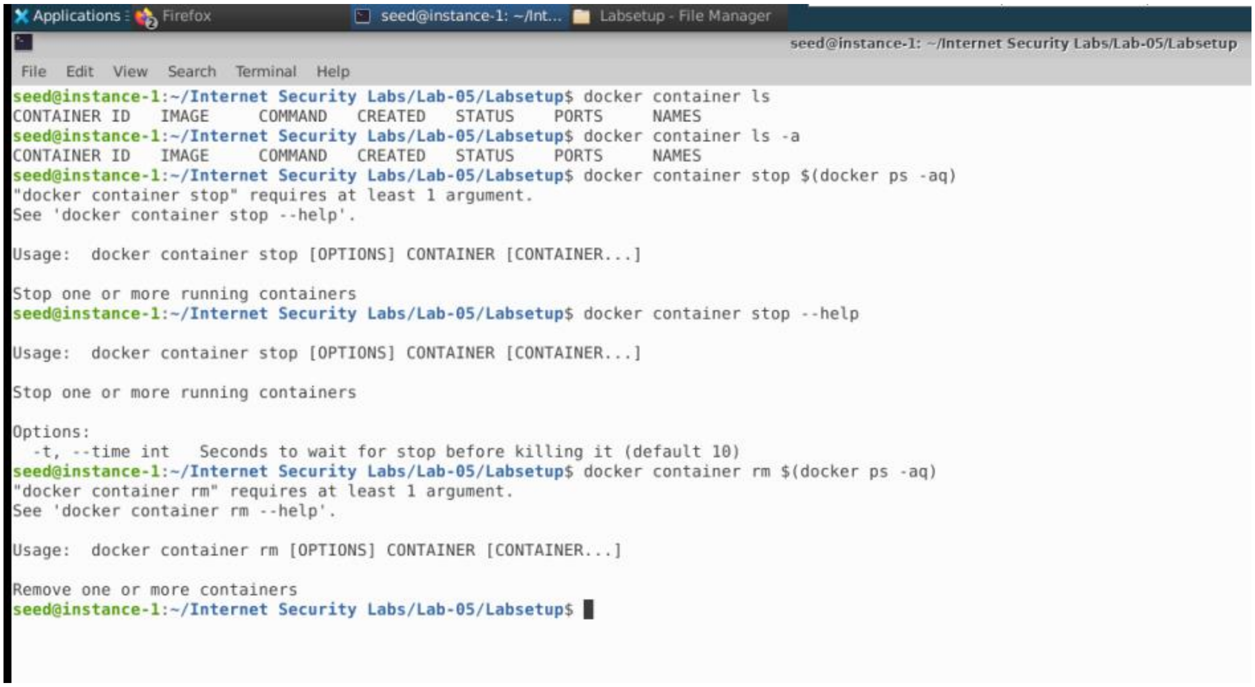
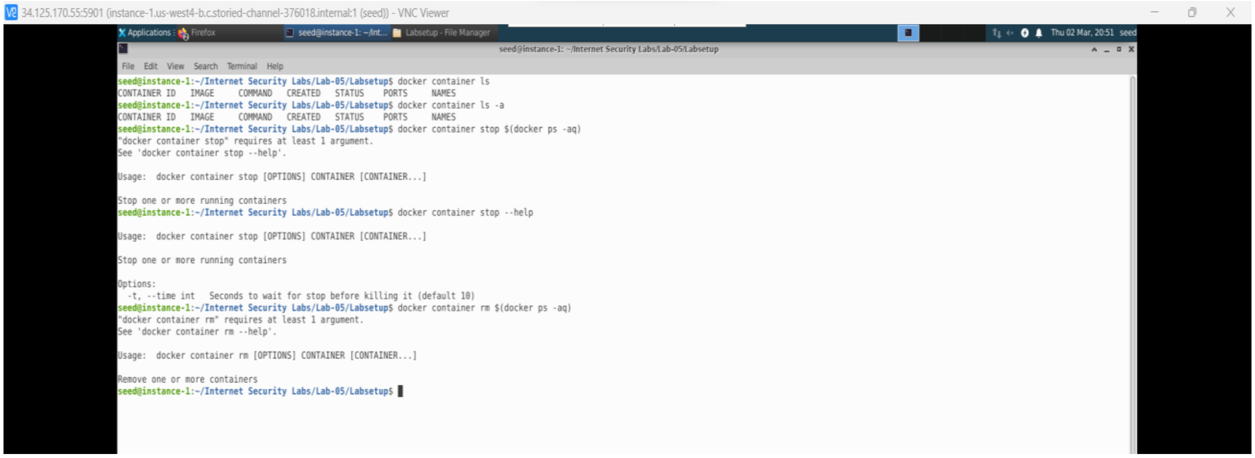
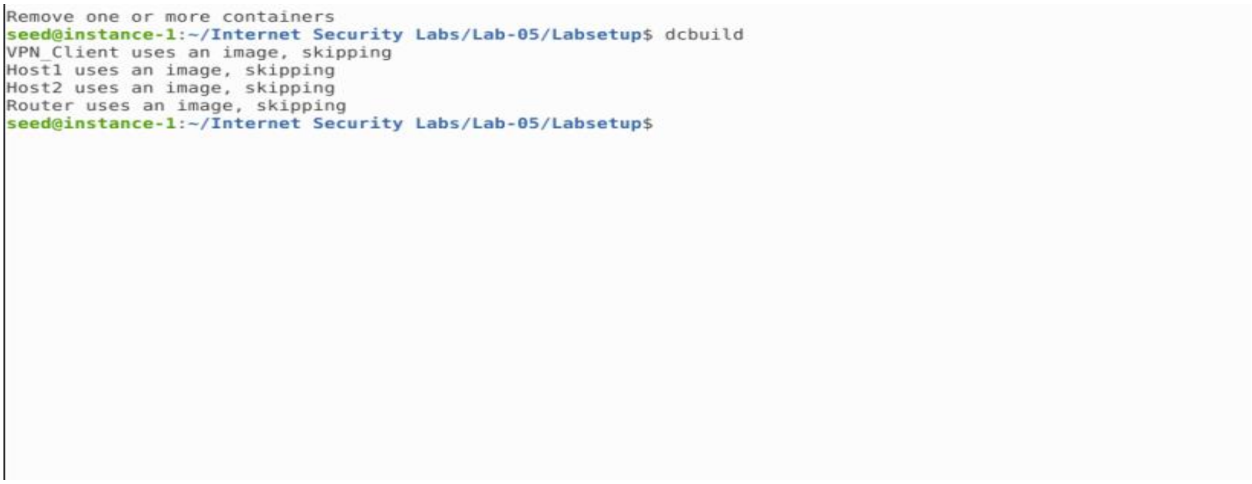
ddocker


Applications
Firefox
seed@instance-1: ~/Int...
Labsetup - File Manager
seed@instance-1: ~/Internet Security Labs/Lab-05/Labsetup
File Edit View Search Terminal Help

```
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docker container ls
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docker container ls -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docker container stop $(docker ps -aq)
"docker container stop" requires at least 1 argument.
See 'docker container stop --help'.

Usage:  docker container stop [OPTIONS] CONTAINER [CONTAINER...]

Stop one or more running containers
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docker container stop --help

Usage:  docker container stop [OPTIONS] CONTAINER [CONTAINER...]

Stop one or more running containers

Options:
  -t, --time int   Seconds to wait for stop before killing it (default 10)
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docker container rm $(docker ps -aq)
"docker container rm" requires at least 1 argument.
See 'docker container rm --help'.

Usage:  docker container rm [OPTIONS] CONTAINER [CONTAINER...]

Remove one or more containers
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ 
```

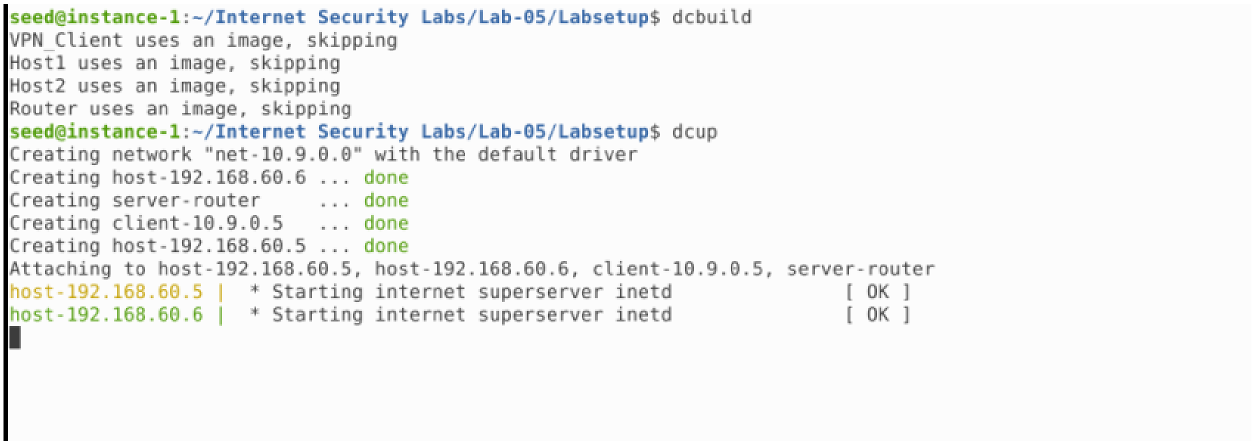

34.125.170.55:5901 (instance-1.us-west4-b.c.storied-channel-376018.internal:1 (seed)) - VNC Viewer
Applications
Firefox
seed@instance-1: ~/Int...
Labsetup - File Manager
Thu 02 Mar, 20:51
seed
seed@instance-1: ~/Internet Security Labs/Lab-05/Labsetup
File Edit View Search Terminal Help

```
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docker container ls
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docker container ls -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docker container stop $(docker ps -aq)
"docker container stop" requires at least 1 argument.
See 'docker container stop --help'.

Usage:  docker container stop [OPTIONS] CONTAINER [CONTAINER...]

Stop one or more running containers
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docker container stop --help

Usage:  docker container stop [OPTIONS] CONTAINER [CONTAINER...]

Stop one or more running containers

Options:
  -t, --time int   Seconds to wait for stop before killing it (default 10)
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docker container rm $(docker ps -aq)
"docker container rm" requires at least 1 argument.
See 'docker container rm --help'.

Usage:  docker container rm [OPTIONS] CONTAINER [CONTAINER...]

Remove one or more containers
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ 
```



```
Remove one or more containers
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ dcbuild
VPN_Client uses an image, skipping
Host1 uses an image, skipping
Host2 uses an image, skipping
Router uses an image, skipping
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$
```

To bring up all the containers, 2 networks and 4 containers

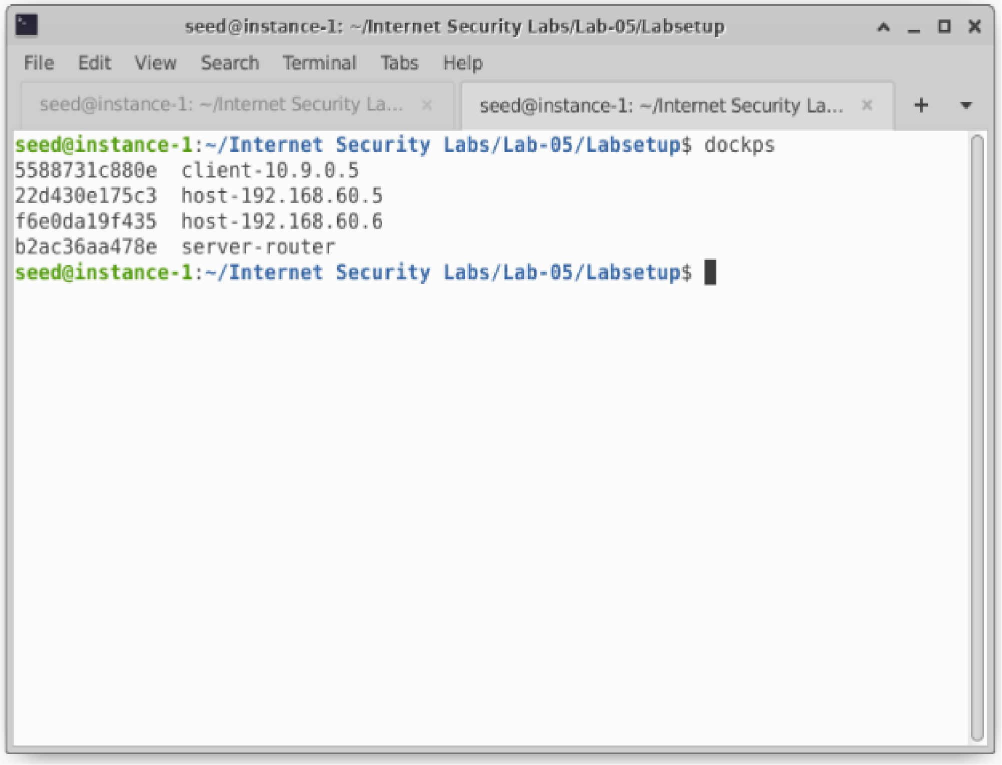


```
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ dcbuild
VPN_Client uses an image, skipping
Host1 uses an image, skipping
Host2 uses an image, skipping
Router uses an image, skipping
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ dcup
Creating network "net-10.9.0.0" with the default driver
Creating host-192.168.60.6 ... done
Creating server-router     ... done
Creating client-10.9.0.5   ... done
Creating host-192.168.60.5 ... done
Attaching to host-192.168.60.5, host-192.168.60.6, client-10.9.0.5, server-router
host-192.168.60.5 |  * Starting internet superserver inetd              [ OK ]
host-192.168.60.6 |  * Starting internet superserver inetd              [ OK ]
```

Here we have a proper network the VPN client, it will setup a pdp connection to the VPN server and try to access the protected host V.

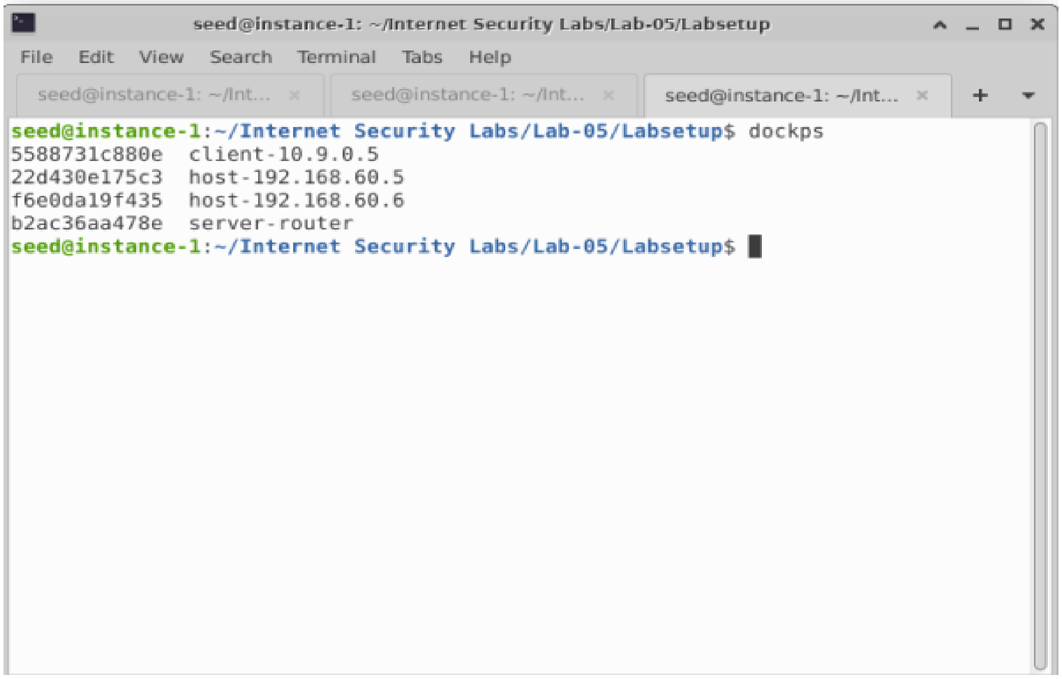
In the image we have 4 containers but in the terminal we have 4 containers.

We have opened a second tab.


seed@instance-1: ~/Internet Security Labs/Lab-05/Labsetup
File Edit View Search Terminal Tabs Help
seed@instance-1: ~/Internet Security La...
seed@instance-1: ~/Internet Security La...

```
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ dockps
5588731c880e  client-10.9.0.5
22d430e175c3  host-192.168.60.5
f6e0da19f435  host-192.168.60.6
b2ac36aa478e  server-router
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$
```

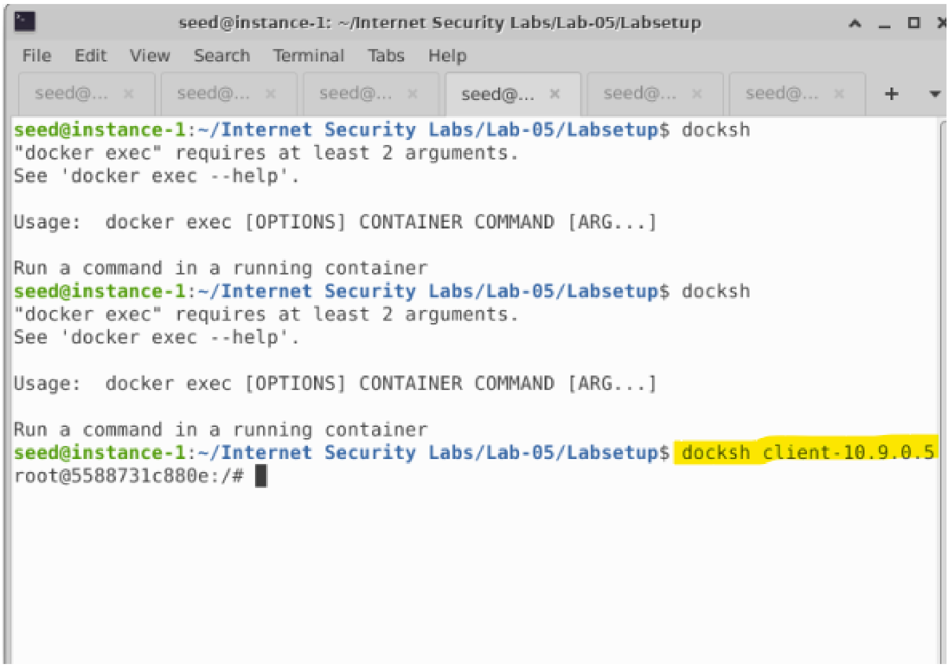
In the 3rd tab


seed@instance-1: ~/Internet Security Labs/Lab-05/Labsetup
File Edit View Search Terminal Tabs Help
seed@instance-1: ~/Int...
seed@instance-1: ~/Int...
seed@instance-1: ~/Int...

```
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ dockps
5588731c880e  client-10.9.0.5
22d430e175c3  host-192.168.60.5
f6e0da19f435  host-192.168.60.6
b2ac36aa478e  server-router
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$
```

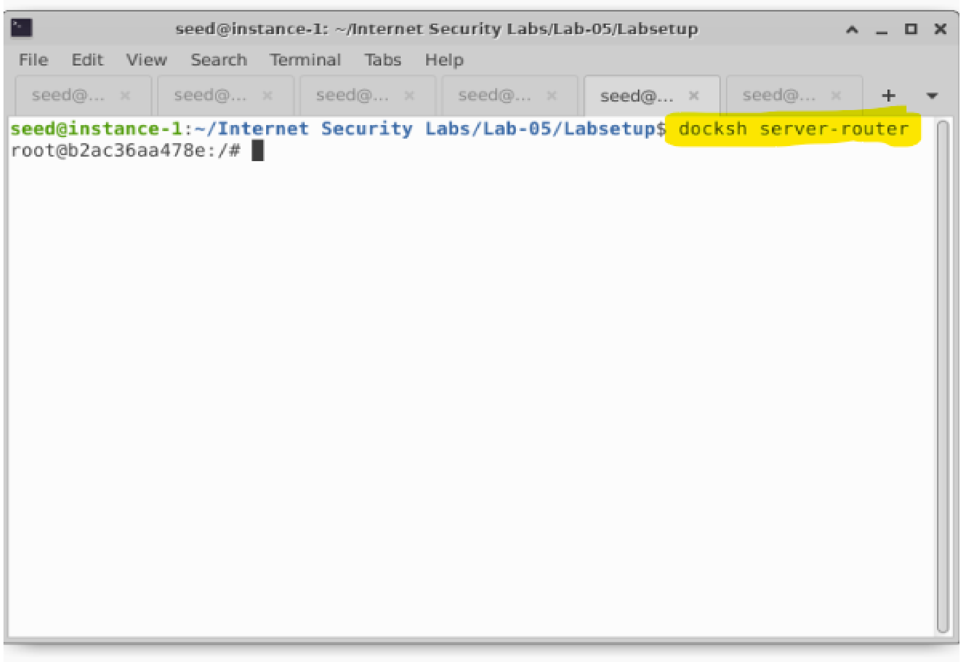
First 3 are containers

4th tab


seed@instance-1: ~/Internet Security Labs/Lab-05/Labsetup
File Edit View Search Terminal Tabs Help
seed@... seed@... seed@... seed@... seed@... seed@...

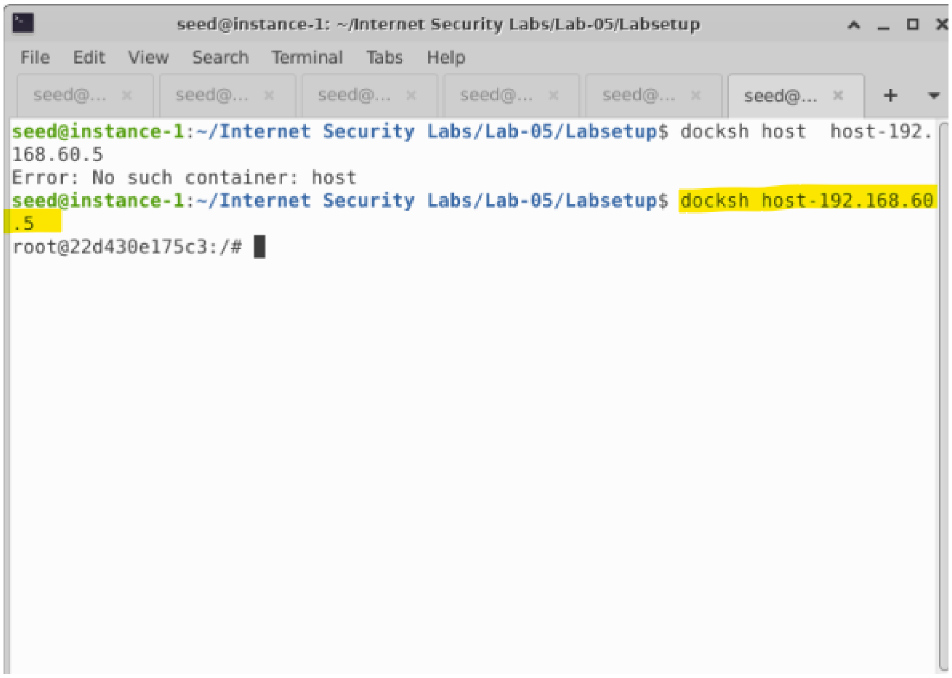
```
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docksh
"docker exec" requires at least 2 arguments.
See 'docker exec --help'.

Usage:  docker exec [OPTIONS] CONTAINER COMMAND [ARG...]

Run a command in a running container
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docksh
"docker exec" requires at least 2 arguments.
See 'docker exec --help'.

Usage:  docker exec [OPTIONS] CONTAINER COMMAND [ARG...]

Run a command in a running container
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docksh client-10.9.0.5
root@5588731c880e:/#
```

5th tab


seed@instance-1: ~/Internet Security Labs/Lab-05/Labsetup
File Edit View Search Terminal Tabs Help
seed@... seed@... seed@... seed@... seed@... seed@...

```
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docksh server-router
root@b2ac36aa478e:/#
```

6th tab

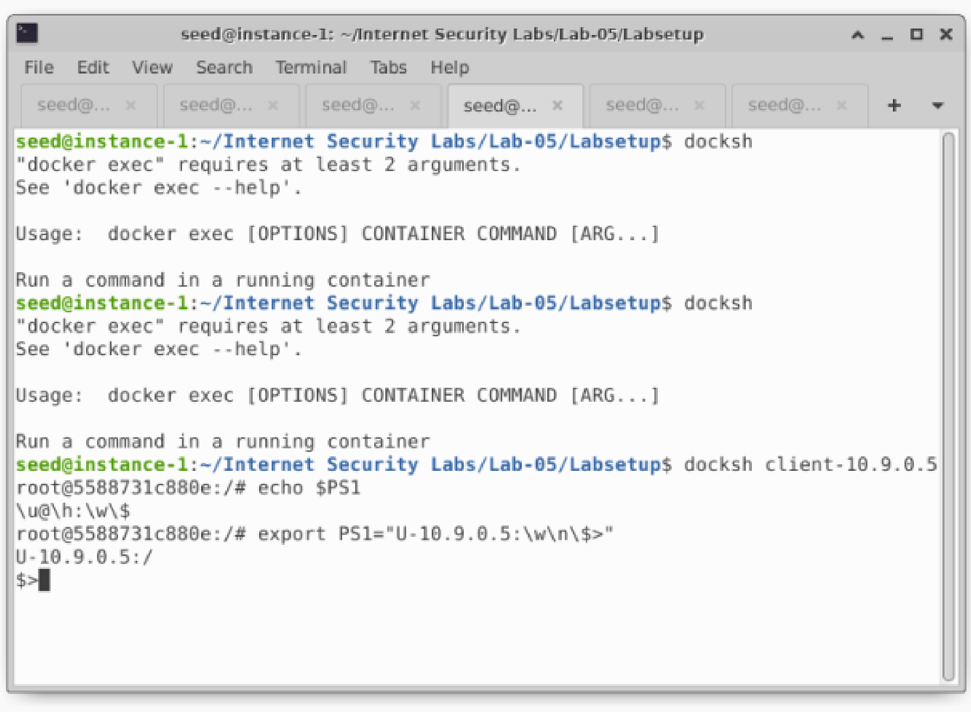

seed@instance-1: ~/Internet Security Labs/Lab-05/Labsetup
File Edit View Search Terminal Tabs Help
seed@... seed@... seed@... seed@... seed@... seed@...

```
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docksh host  host-192.
168.60.5
Error: No such container: host
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docksh host-192.168.60
.5
root@22d430e175c3:/# 
```

Now let us change the prompt so we can see the IP address clearly and the name of this computer.

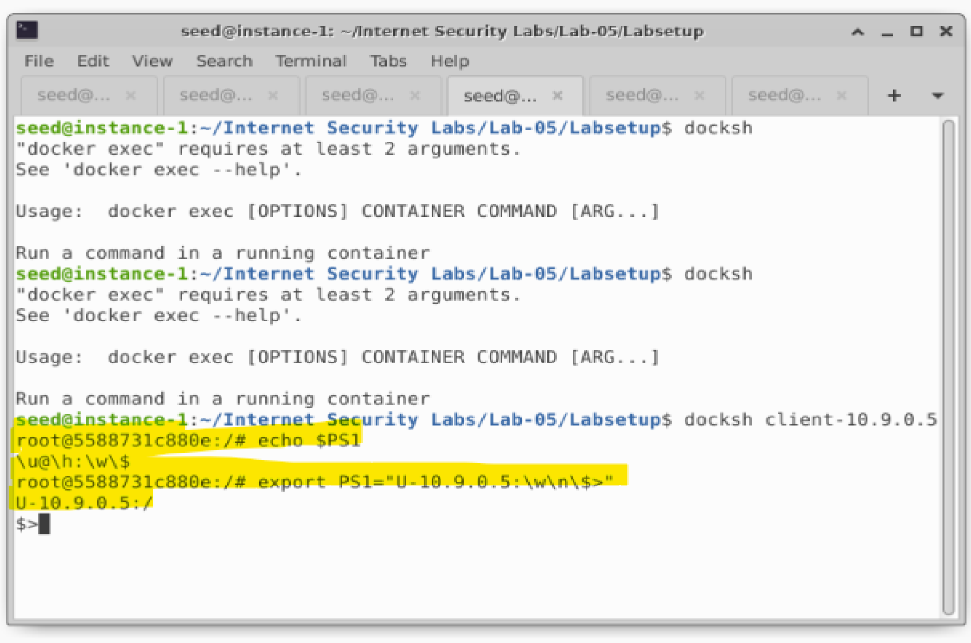
Here we can client or host V with this IP address and this host V with this IP address the router with the two IP addresses.

In the 4th tab, we have



seed@instance-1: ~/Internet Security Labs/Lab-05/Labsetup
File Edit View Search Terminal Tabs Help
seed@... seed@... seed@... seed@... seed@... seed@...

```
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docksh
"docker exec" requires at least 2 arguments.
See 'docker exec --help'.

Usage:  docker exec [OPTIONS] CONTAINER COMMAND [ARG...]

Run a command in a running container
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docksh
"docker exec" requires at least 2 arguments.
See 'docker exec --help'.

Usage:  docker exec [OPTIONS] CONTAINER COMMAND [ARG...]

Run a command in a running container
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docksh client-10.9.0.5
root@5588731c880e:/# echo $PS1
\u@\h:\w\$
root@5588731c880e:/# export PS1="U-10.9.0.5:\w\n\$>"
U-10.9.0.5:/
$>
```

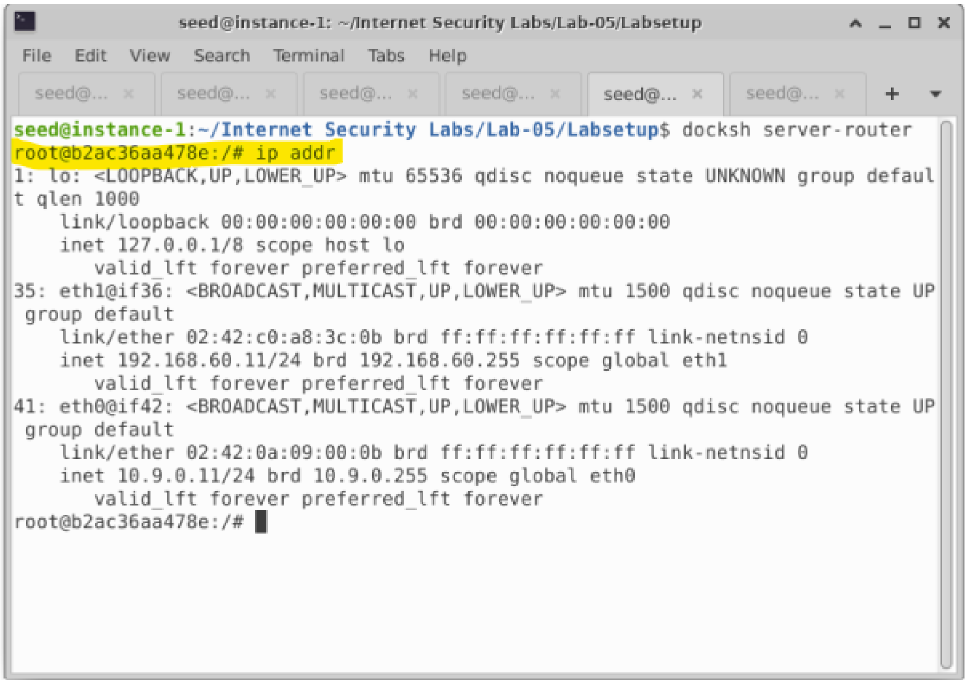


seed@instance-1: ~/Internet Security Labs/Lab-05/Labsetup
File Edit View Search Terminal Tabs Help
seed@... seed@... seed@... seed@... seed@... seed@...

```
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docksh
"docker exec" requires at least 2 arguments.
See 'docker exec --help'.

Usage:  docker exec [OPTIONS] CONTAINER COMMAND [ARG...]

Run a command in a running container
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docksh
"docker exec" requires at least 2 arguments.
See 'docker exec --help'.

Usage:  docker exec [OPTIONS] CONTAINER COMMAND [ARG...]

Run a command in a running container
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docksh client-10.9.0.5
root@5588731c880e:/# echo $PS1
\u@\h:\w\$
root@5588731c880e:/# export PS1="U-10.9.0.5:\w\n\$>"
U-10.9.0.5:/
$>
```

So I will use a host U with the IP address and this is the VPN client, working directory and here I want to add a new line

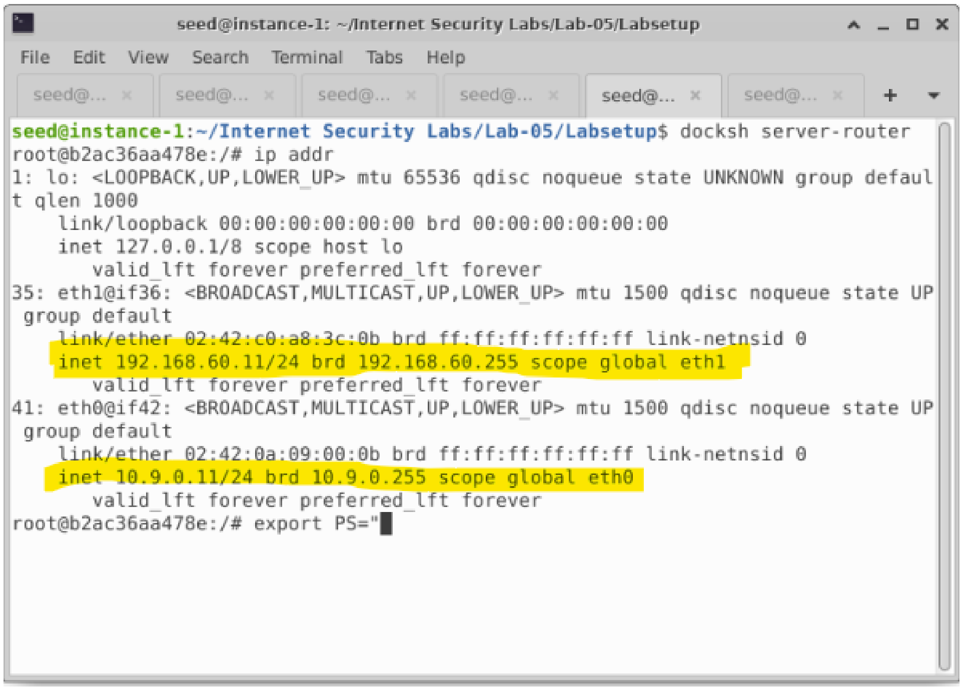
In the 5th tab,for this server-router we have two interfaces


seed@instance-1: ~/Internet Security Labs/Lab-05/Labsetup
File Edit View Search Terminal Tabs Help
seed@... seed@... seed@... seed@... seed@... seed@...

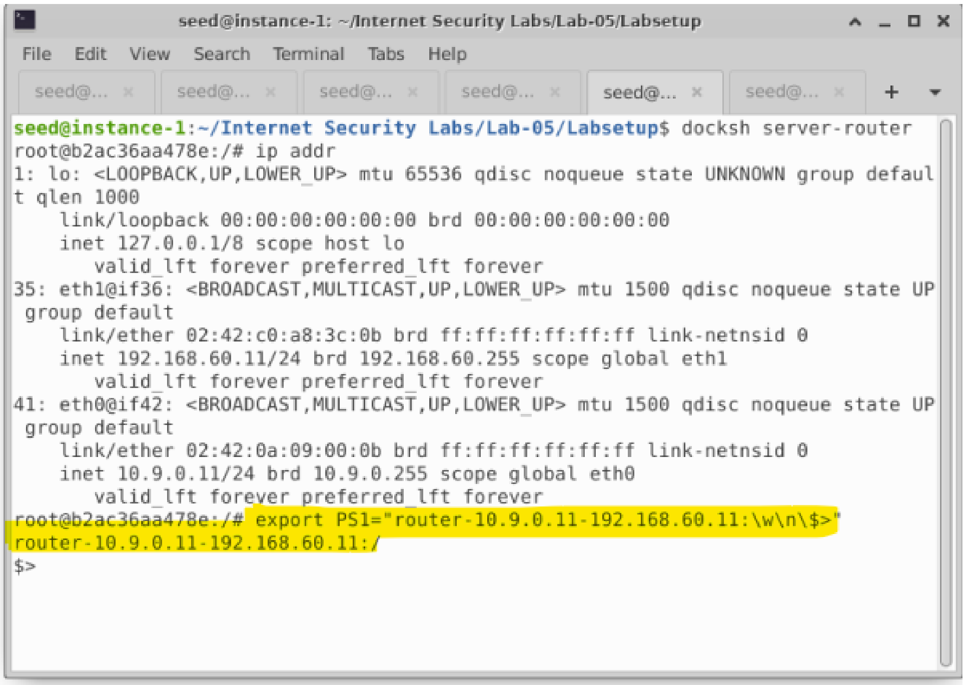
```
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docksh server-router
root@b2ac36aa478e:/# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
35: eth1@if36: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
 group default
    link/ether 02:42:c0:a8:3c:0b brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.168.60.11/24 brd 192.168.60.255 scope global eth1
       valid_lft forever preferred_lft forever
41: eth0@if42: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
 group default
    link/ether 02:42:0a:09:00:0b brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.9.0.11/24 brd 10.9.0.255 scope global eth0
       valid_lft forever preferred_lft forever
root@b2ac36aa478e:/# 
```

//////It has 0 to the auto network, ether1 to the private network

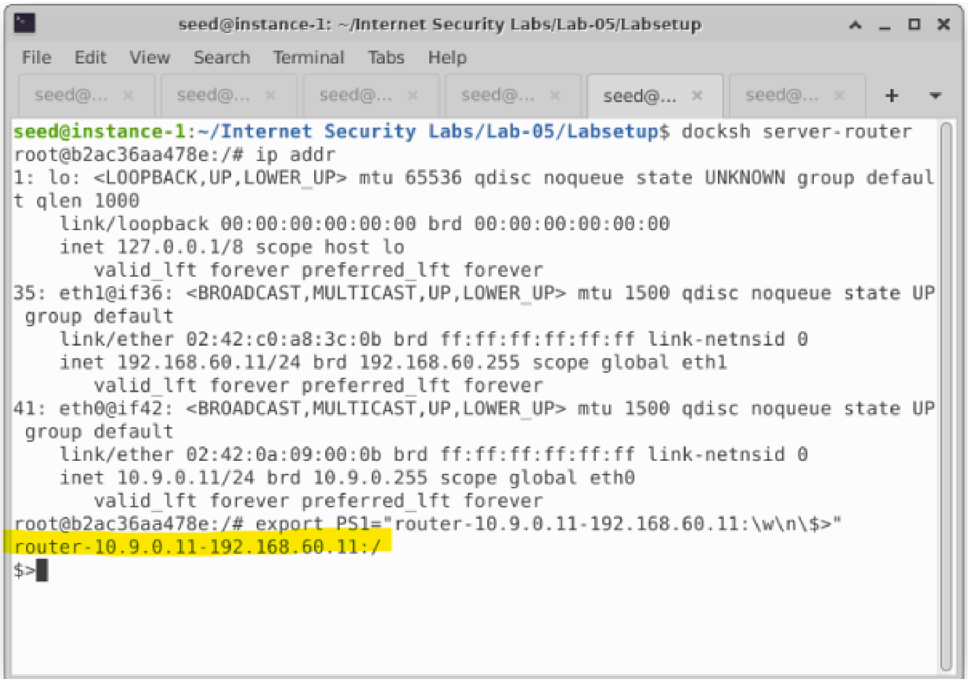
So this is a server for the router


seed@instance-1: ~/Internet Security Labs/Lab-05/Labsetup
File Edit View Search Terminal Tabs Help
seed@... seed@... seed@... seed@... seed@... seed@...

```
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docksh server-router
root@b2ac36aa478e:/# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
35: eth1@if36: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
 group default
    link/ether 02:42:c0:a8:3c:0b brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.168.60.11/24 brd 192.168.60.255 scope global eth1
       valid_lft forever preferred_lft forever
41: eth0@if42: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
 group default
    link/ether 02:42:0a:09:00:0b brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.9.0.11/24 brd 10.9.0.255 scope global eth0
       valid_lft forever preferred_lft forever
root@b2ac36aa478e:/# export PS="
```


seed@instance-1: ~/Internet Security Labs/Lab-05/Labsetup
File Edit View Search Terminal Tabs Help
seed@... seed@... seed@... seed@... seed@... seed@...

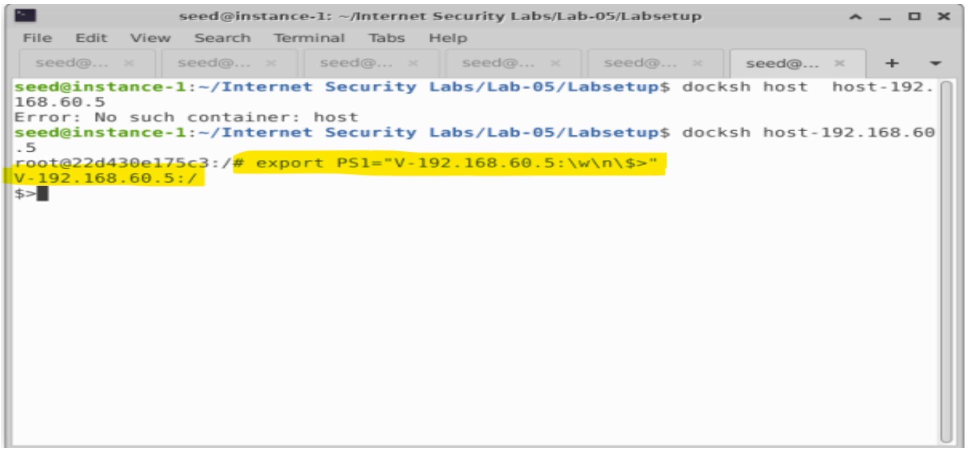
```
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docksh server-router
root@b2ac36aa478e:/# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
35: eth1@if36: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
 group default
    link/ether 02:42:c0:a8:3c:0b brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.168.60.11/24 brd 192.168.60.255 scope global eth1
       valid_lft forever preferred_lft forever
41: eth0@if42: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
 group default
    link/ether 02:42:0a:09:00:0b brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.9.0.11/24 brd 10.9.0.255 scope global eth0
       valid_lft forever preferred_lft forever
root@b2ac36aa478e:/# export PS1="router-10.9.0.11-192.168.60.11:\w\n\$>"
router-10.9.0.11-192.168.60.11:/
$>
```



seed@instance-1: ~/Internet Security Labs/Lab-05/Labsetup
File Edit View Search Terminal Tabs Help
seed@... seed@... seed@... seed@... seed@... seed@...

```
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docksh server-router
root@b2ac36aa478e:/# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
35: eth1@if36: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
 group default
    link/ether 02:42:c0:a8:3c:0b brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.168.60.11/24 brd 192.168.60.255 scope global eth1
       valid_lft forever preferred_lft forever
41: eth0@if42: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
 group default
    link/ether 02:42:0a:09:00:0b brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.9.0.11/24 brd 10.9.0.255 scope global eth0
       valid_lft forever preferred_lft forever
root@b2ac36aa478e:/# export PS1="router-10.9.0.11-192.168.60.11:\w\n\$>"
router-10.9.0.11-192.168.60.11:/
$>
```

Highlighted one is a router on the VPN server



seed@instance-1: ~/Internet Security Labs/Lab-05/Labsetup
File Edit View Search Terminal Tabs Help
seed@... seed@... seed@... seed@... seed@... seed@...

```
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docksh host  host-192.
168.60.5
Error: No such container: host
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docksh host-192.168.60
.5
root@22d430e175c3:/# export PS1="V-192.168.60.5:\w\n\$>"
V-192.168.60.5:/
$>
```

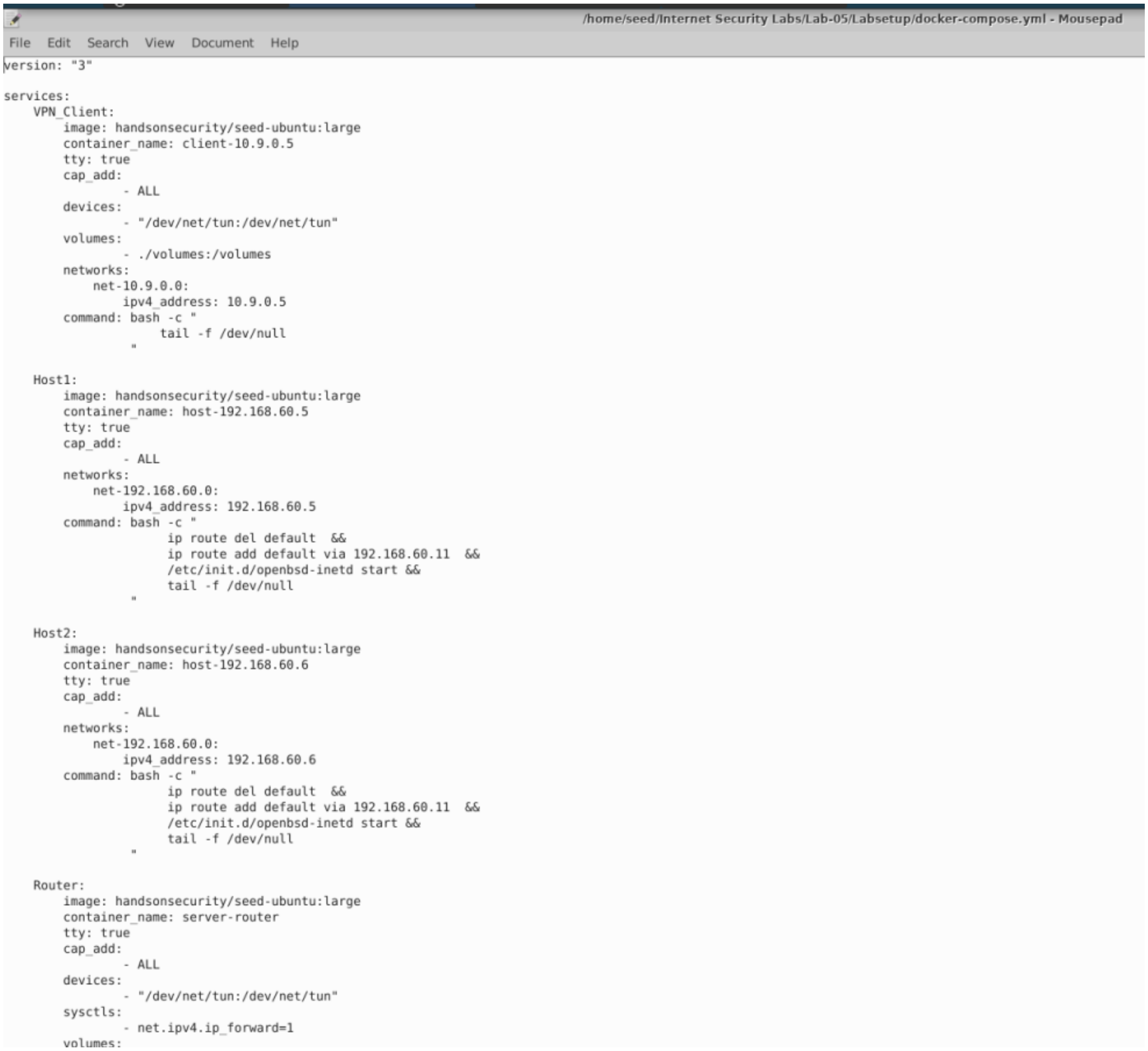
Now this is a router on the VPN server on the host V

1) Now we will be setting up the environment

-Host V and Host U client within client set up within tunnel to the VPN Server and try to access this protected hostway.

In this Lab both the time interface are created with the python code, so now letsgo through Lab menu, the shared folder, the volumes.

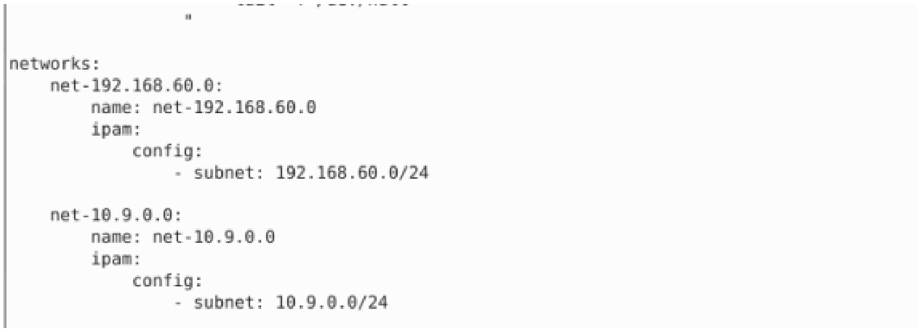
Open the docker-compose.yml file from the Labsetup folder.


/home/seed/Internet Security Labs/Lab-05/Labsetup/docker-compose.yml - Mousepad
File
Edit
Search
View
Document
Help

```
version: "3"

services:
    VPN_Client:
        image: handsonsecurity/seed-ubuntu:large
        container_name: client-10.9.0.5
        tty: true
        cap_add:
                - ALL
        devices:
                - "/dev/net/tun:/dev/net/tun"
        volumes:
                - ./volumes:/volumes
        networks:
            net-10.9.0.0:
                ipv4_address: 10.9.0.5
        command: bash -c "
                     tail -f /dev/null
                 "

    Host1:
        image: handsonsecurity/seed-ubuntu:large
        container_name: host-192.168.60.5
        tty: true
        cap_add:
                - ALL
        networks:
            net-192.168.60.0:
                ipv4_address: 192.168.60.5
        command: bash -c "
                      ip route del default  &&
                      ip route add default via 192.168.60.11  &&
                      /etc/init.d/openbsd-inetd start &&
                      tail -f /dev/null
                 "

    Host2:
        image: handsonsecurity/seed-ubuntu:large
        container_name: host-192.168.60.6
        tty: true
        cap_add:
                - ALL
        networks:
            net-192.168.60.0:
                ipv4_address: 192.168.60.6
        command: bash -c "
                      ip route del default  &&
                      ip route add default via 192.168.60.11  &&
                      /etc/init.d/openbsd-inetd start &&
                      tail -f /dev/null
                 "

    Router:
        image: handsonsecurity/seed-ubuntu:large
        container_name: server-router
        tty: true
        cap_add:
                - ALL
        devices:
                - "/dev/net/tun:/dev/net/tun"
        sysctls:
                - net.ipv4.ip_forward=1
        volumes:
```

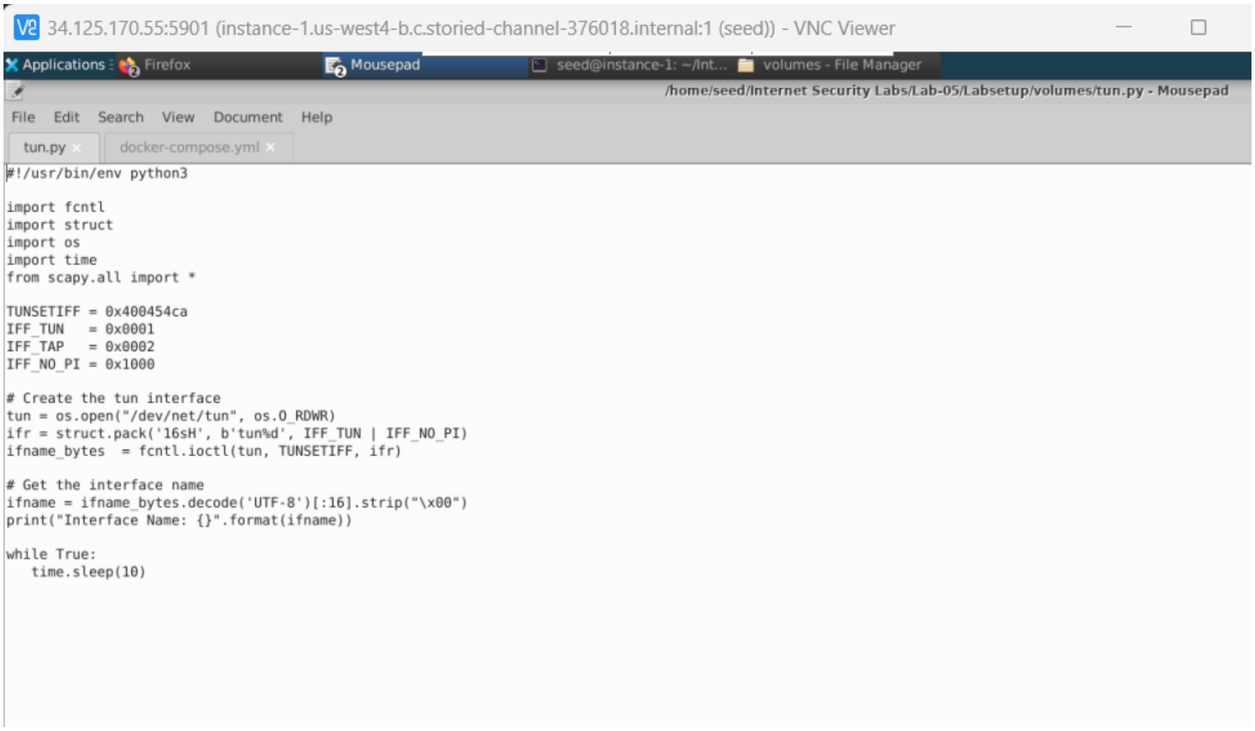
In the docker-composer file the volumes, is used as a VPN client, not by host one and not by host 2, was used by docker and a tool networks

This one is used to simulate public network, let's create a similar major public network and there is private network used to protect the host V



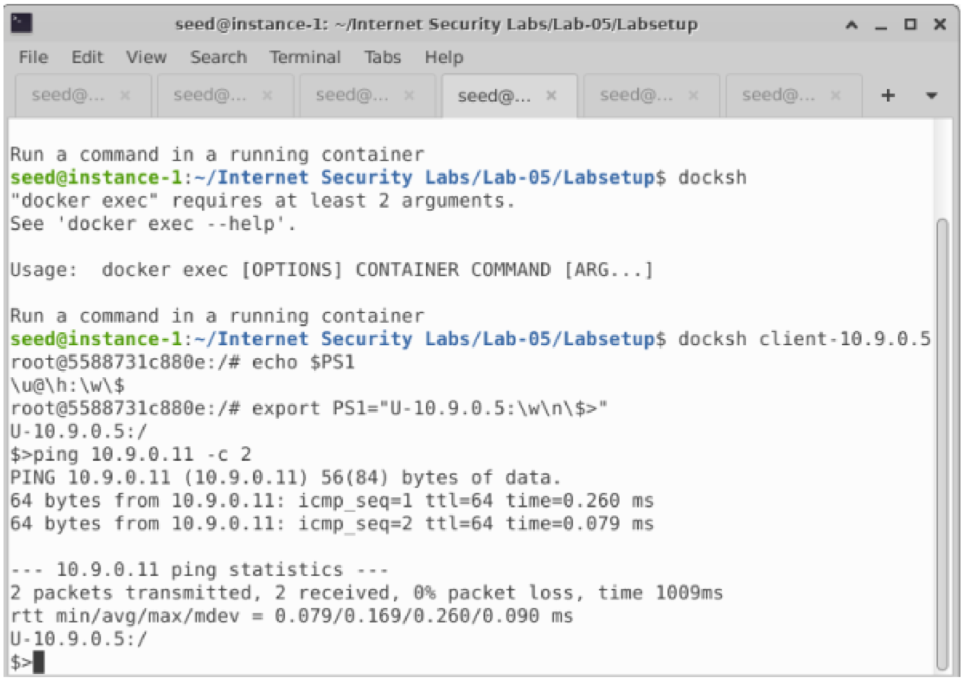
```
                "

networks:
    net-192.168.60.0:
        name: net-192.168.60.0
        ipam:
            config:
                - subnet: 192.168.60.0/24

    net-10.9.0.0:
        name: net-10.9.0.0
        ipam:
            config:
                - subnet: 10.9.0.0/24
```

Inside the volume folder a template is provided called tun.py(tunnel), so we have to write the code here

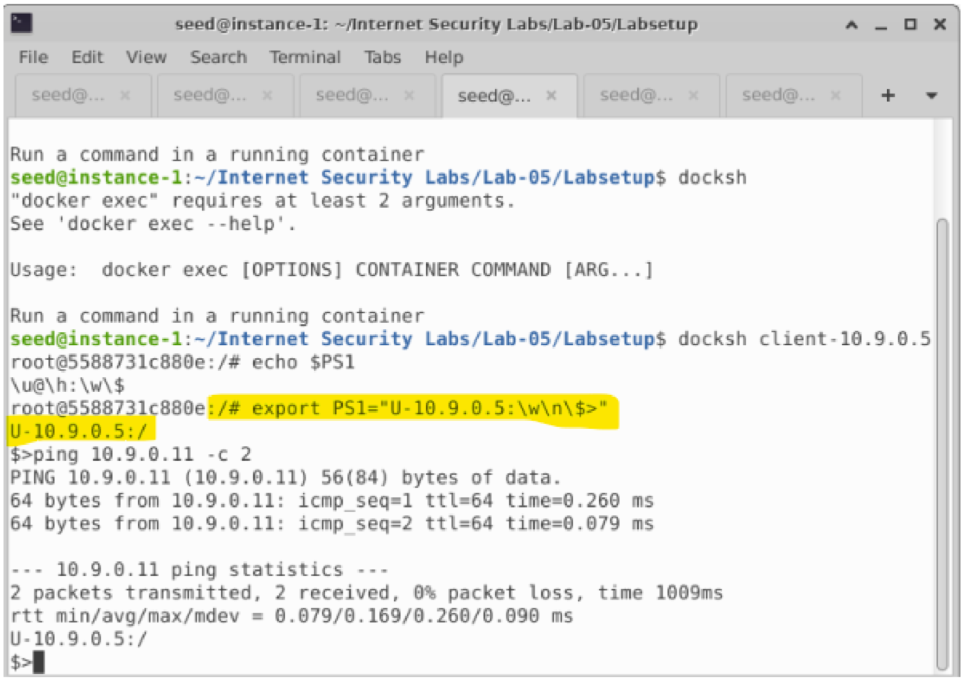

34.125.170.55:5901 (instance-1.us-west4-b.c.storied-channel-376018.internal:1 (seed)) - VNC Viewer
Applications
Firefox
Mousepad
seed@instance-1: ~/Int...
volumes - File Manager
/home/seed/Internet Security Labs/Lab-05/Labsetup/volumes/tun.py - Mousepad
File
Edit
Search
View
Document
Help
tun.py
docker-compose.yml

```
#!/usr/bin/env python3

import fcntl
import struct
import os
import time
from scapy.all import *

TUNSETIFF = 0x400454ca
IFF_TUN   = 0x0001
IFF_TAP   = 0x0002
IFF_NO_PI = 0x1000

# Create the tun interface
tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack('16sH', b'tun%d', IFF_TUN | IFF_NO_PI)
ifname_bytes  = fcntl.ioctl(tun, TUNSETIFF, ifr)

# Get the interface name
ifname = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
print("Interface Name: {}".format(ifname))

while True:
   time.sleep(10)
```

Description:---



```
Run a command in a running container
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docksh
"docker exec" requires at least 2 arguments.
See 'docker exec --help'.

Usage:  docker exec [OPTIONS] CONTAINER COMMAND [ARG...]

Run a command in a running container
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docksh client-10.9.0.5
root@5588731c880e:/# echo $PS1
\u@\h:\w\$
root@5588731c880e:/# export PS1="U-10.9.0.5:\w\n\$>"
U-10.9.0.5:/
$>ping 10.9.0.11 -c 2
PING 10.9.0.11 (10.9.0.11) 56(84) bytes of data.
64 bytes from 10.9.0.11: icmp_seq=1 ttl=64 time=0.260 ms
64 bytes from 10.9.0.11: icmp_seq=2 ttl=64 time=0.079 ms

--- 10.9.0.11 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1009ms
rtt min/avg/max/mdev = 0.079/0.169/0.260/0.090 ms
U-10.9.0.5:/
$>
```


seed@instance-1: ~/Internet Security Labs/Lab-05/Labsetup
File Edit View Search Terminal Tabs Help
seed@... seed@... seed@... seed@... seed@... seed@...

Run a command in a running container
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docksh
"docker exec" requires at least 2 arguments.
See 'docker exec --help'.

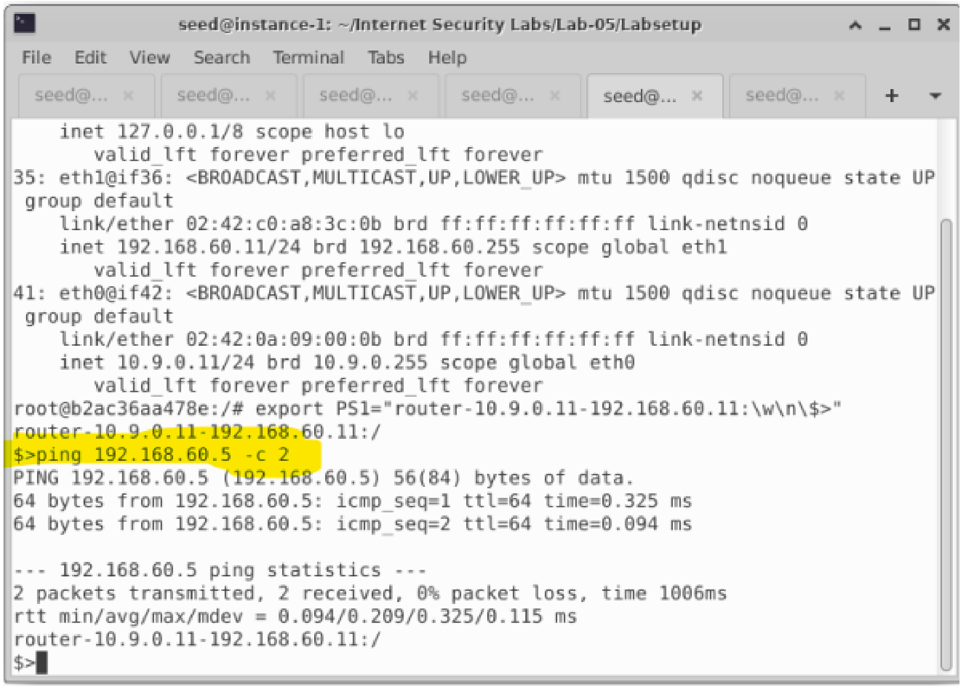
Usage: docker exec [OPTIONS] CONTAINER COMMAND [ARG...]

Run a command in a running container
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docksh client-10.9.0.5
root@5588731c880e:/# echo $PS1
\u@\h:\w\$
root@5588731c880e:/# export PS1="U-10.9.0.5:\w\n\$>"
U-10.9.0.5:/
$>ping 10.9.0.11 -c 2
PING 10.9.0.11 (10.9.0.11) 56(84) bytes of data.
64 bytes from 10.9.0.11: icmp_seq=1 ttl=64 time=0.260 ms
64 bytes from 10.9.0.11: icmp_seq=2 ttl=64 time=0.079 ms

--- 10.9.0.11 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1009ms
rtt min/avg/max/mdev = 0.079/0.169/0.260/0.090 ms
U-10.9.0.5:/
$>

Test-2:- VPN server can communicate with Host V

Here we can in server-router we can ping host V, two packets transmitted and 2 received

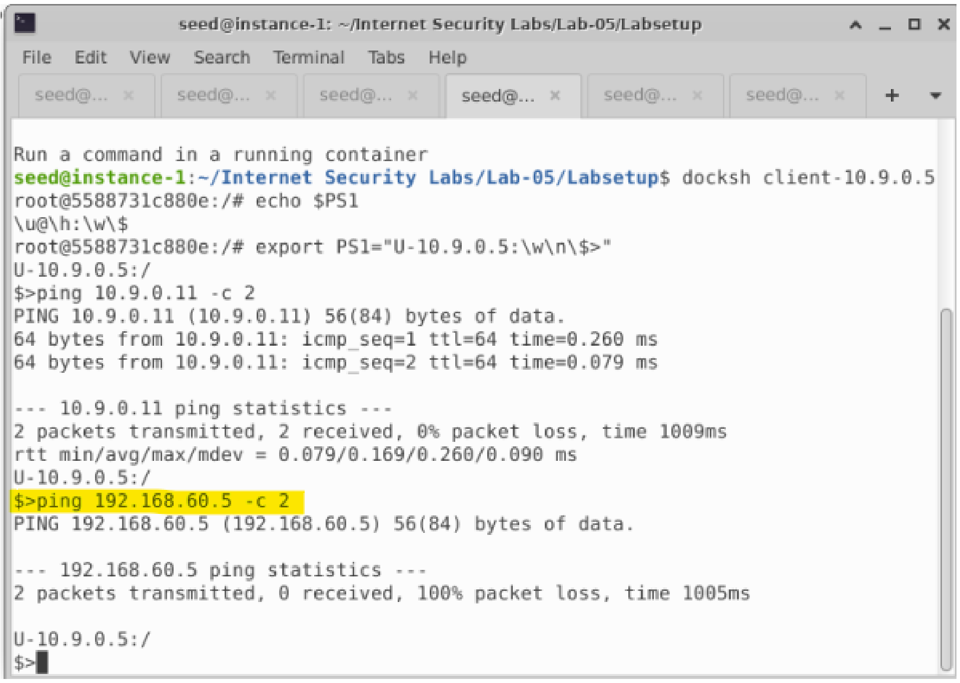

seed@instance-1: ~/Internet Security Labs/Lab-05/Labsetup
File Edit View Search Terminal Tabs Help
seed@... seed@... seed@... seed@... seed@... seed@...

```
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
35: eth1@if36: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
 group default
    link/ether 02:42:c0:a8:3c:0b brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.168.60.11/24 brd 192.168.60.255 scope global eth1
       valid_lft forever preferred_lft forever
41: eth0@if42: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
 group default
    link/ether 02:42:0a:09:00:0b brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.9.0.11/24 brd 10.9.0.255 scope global eth0
       valid_lft forever preferred_lft forever
root@b2ac36aa478e:/# export PS1="router-10.9.0.11-192.168.60.11:\w\n\$>"
router-10.9.0.11-192.168.60.11:/
$>ping 192.168.60.5 -c 2
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
64 bytes from 192.168.60.5: icmp_seq=1 ttl=64 time=0.325 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=64 time=0.094 ms

--- 192.168.60.5 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1006ms
rtt min/avg/max/mdev = 0.094/0.209/0.325/0.115 ms
router-10.9.0.11-192.168.60.11:/
$>
```

Test 3: Host U should not be able to communicate with Host V

The environment we set up it has a private network protector suite and later we will set up VPN tunnel, then host V can communicate with host report currently.

If there is no VPN tunnel host V should not be able to communicate with host V


seed@instance-1: ~/Internet Security Labs/Lab-05/Labsetup
File Edit View Search Terminal Tabs Help
seed@... seed@... seed@... seed@... seed@... seed@...

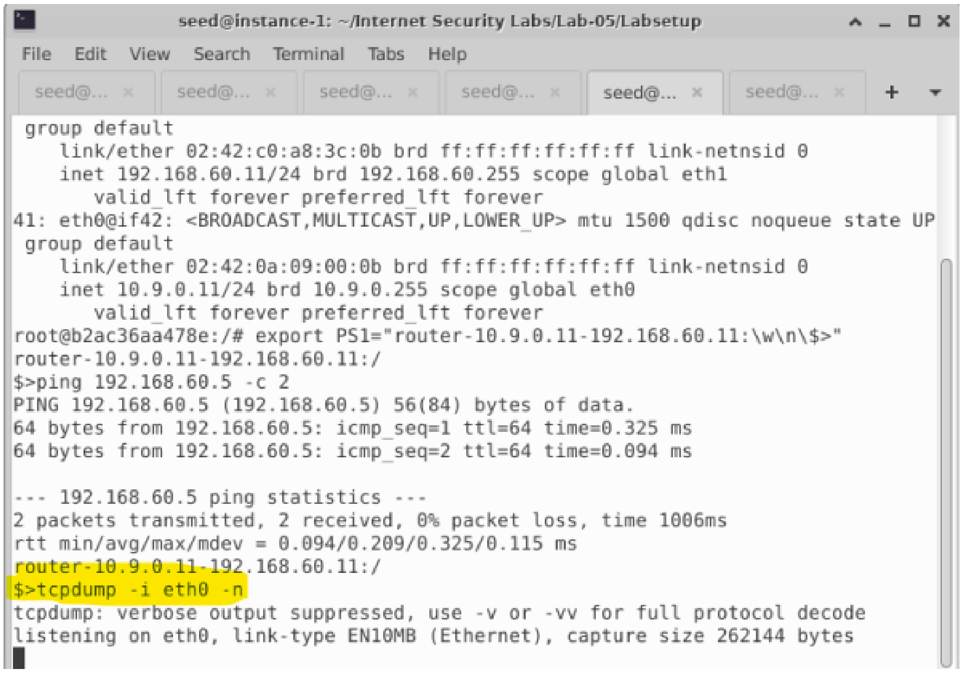
```
Run a command in a running container
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docksh client-10.9.0.5
root@5588731c880e:/# echo $PS1
\u@\h:\w\$
root@5588731c880e:/# export PS1="U-10.9.0.5:\w\n\$>"
U-10.9.0.5:/
$>ping 10.9.0.11 -c 2
PING 10.9.0.11 (10.9.0.11) 56(84) bytes of data.
64 bytes from 10.9.0.11: icmp_seq=1 ttl=64 time=0.260 ms
64 bytes from 10.9.0.11: icmp_seq=2 ttl=64 time=0.079 ms

--- 10.9.0.11 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1009ms
rtt min/avg/max/mdev = 0.079/0.169/0.260/0.090 ms
U-10.9.0.5:/
$>ping 192.168.60.5 -c 2
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.

--- 192.168.60.5 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1005ms

U-10.9.0.5:/
$>
```

We can ping from host U to ping host V

Test 4:- Run tcpdump on the router, and sniff the traffic on each of the network. Show that you can capture packets.

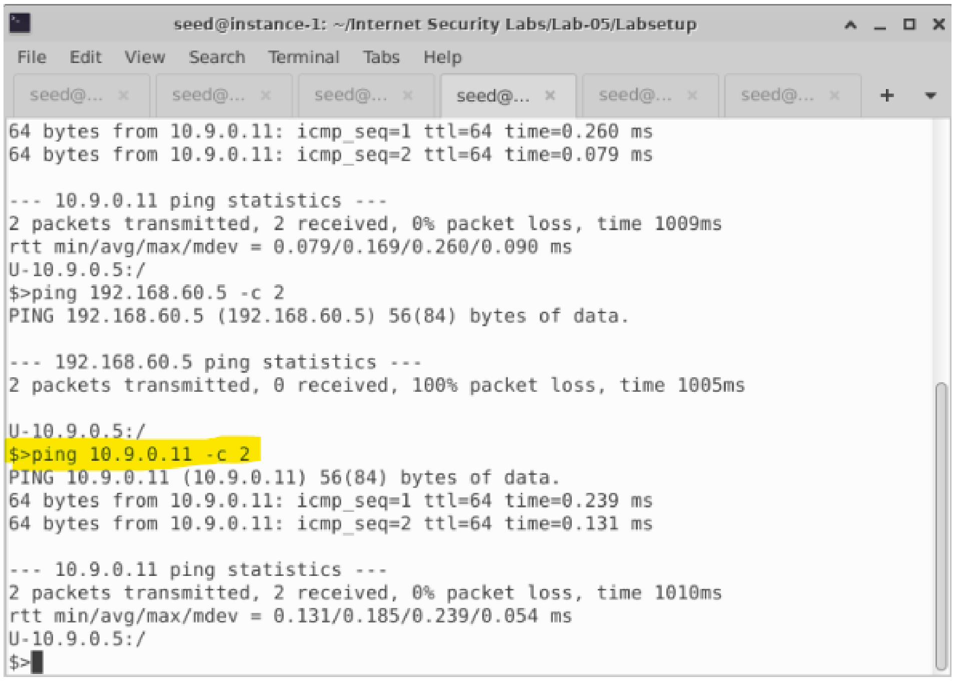
On router we enter as tcpdump -I eth

First we sniff on interface eth0

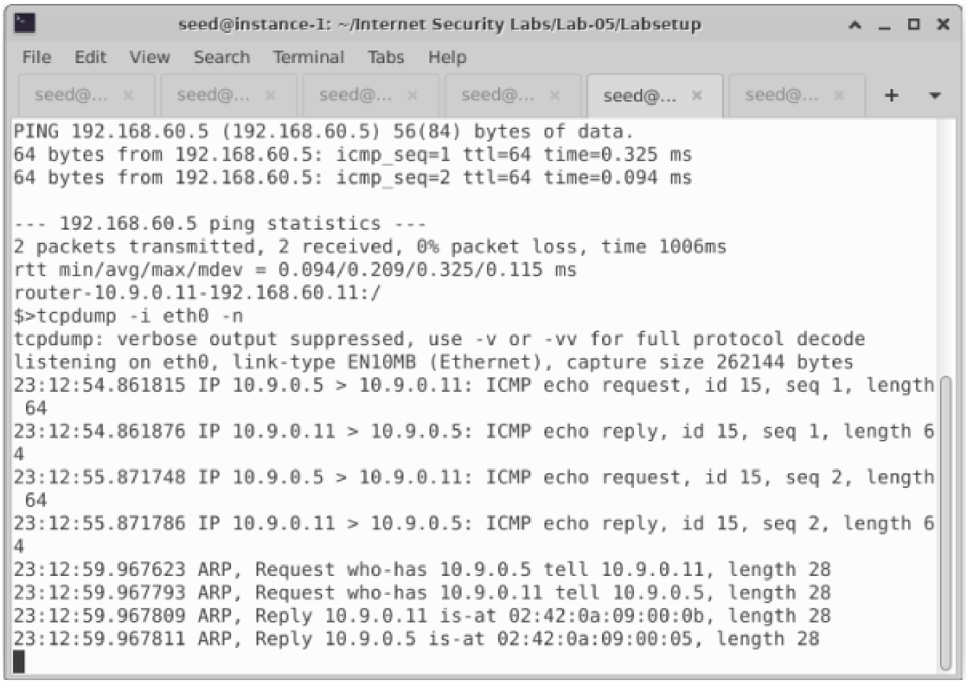

seed@instance-1: ~/Internet Security Labs/Lab-05/Labsetup
File Edit View Search Terminal Tabs Help
seed@... seed@... seed@... seed@... seed@... seed@...

```
 group default
    link/ether 02:42:c0:a8:3c:0b brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.168.60.11/24 brd 192.168.60.255 scope global eth1
       valid_lft forever preferred_lft forever
41: eth0@if42: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
 group default
    link/ether 02:42:0a:09:00:0b brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.9.0.11/24 brd 10.9.0.255 scope global eth0
       valid_lft forever preferred_lft forever
root@b2ac36aa478e:/# export PS1="router-10.9.0.11-192.168.60.11:\w\n\$>"
router-10.9.0.11-192.168.60.11:/
$>ping 192.168.60.5 -c 2
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
64 bytes from 192.168.60.5: icmp_seq=1 ttl=64 time=0.325 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=64 time=0.094 ms

--- 192.168.60.5 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1006ms
rtt min/avg/max/mdev = 0.094/0.209/0.325/0.115 ms
router-10.9.0.11-192.168.60.11:/
$>tcpdump -i eth0 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
```

And then again from host U to ping drop(router)


seed@instance-1: ~/Internet Security Labs/Lab-05/Labsetup
File Edit View Search Terminal Tabs Help
seed@... seed@... seed@... seed@... seed@... seed@...

```
64 bytes from 10.9.0.11: icmp_seq=1 ttl=64 time=0.260 ms
64 bytes from 10.9.0.11: icmp_seq=2 ttl=64 time=0.079 ms

--- 10.9.0.11 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1009ms
rtt min/avg/max/mdev = 0.079/0.169/0.260/0.090 ms
U-10.9.0.5:/
$>ping 192.168.60.5 -c 2
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.

--- 192.168.60.5 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1005ms

U-10.9.0.5:/
$>ping 10.9.0.11 -c 2
PING 10.9.0.11 (10.9.0.11) 56(84) bytes of data.
64 bytes from 10.9.0.11: icmp_seq=1 ttl=64 time=0.239 ms
64 bytes from 10.9.0.11: icmp_seq=2 ttl=64 time=0.131 ms

--- 10.9.0.11 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1010ms
rtt min/avg/max/mdev = 0.131/0.185/0.239/0.054 ms
U-10.9.0.5:/
$>
```

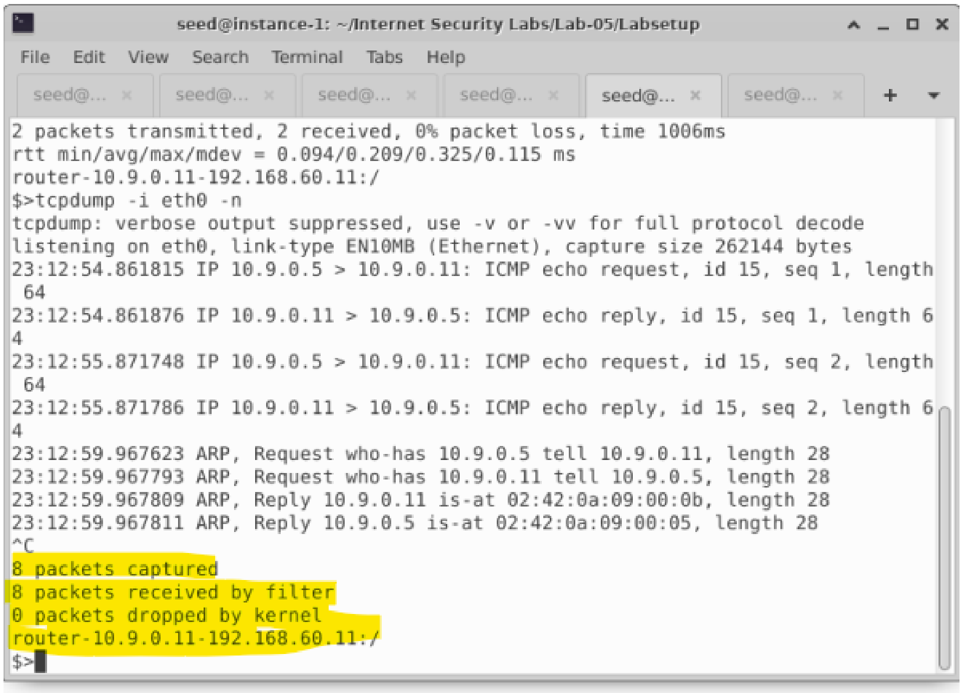

seed@instance-1: ~/Internet Security Labs/Lab-05/Labsetup
File Edit View Search Terminal Tabs Help
seed@... seed@... seed@... seed@... seed@... seed@...

```
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
64 bytes from 192.168.60.5: icmp_seq=1 ttl=64 time=0.325 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=64 time=0.094 ms

--- 192.168.60.5 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1006ms
rtt min/avg/max/mdev = 0.094/0.209/0.325/0.115 ms
router-10.9.0.11-192.168.60.11:/
$>tcpdump -i eth0 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
23:12:54.861815 IP 10.9.0.5 > 10.9.0.11: ICMP echo request, id 15, seq 1, length
 64
23:12:54.861876 IP 10.9.0.11 > 10.9.0.5: ICMP echo reply, id 15, seq 1, length 6
4
23:12:55.871748 IP 10.9.0.5 > 10.9.0.11: ICMP echo request, id 15, seq 2, length
 64
23:12:55.871786 IP 10.9.0.11 > 10.9.0.5: ICMP echo reply, id 15, seq 2, length 6
4
23:12:59.967623 ARP, Request who-has 10.9.0.5 tell 10.9.0.11, length 28
23:12:59.967793 ARP, Request who-has 10.9.0.11 tell 10.9.0.5, length 28
23:12:59.967809 ARP, Reply 10.9.0.11 is-at 02:42:0a:09:00:0b, length 28
23:12:59.967811 ARP, Reply 10.9.0.5 is-at 02:42:0a:09:00:05, length 28
```

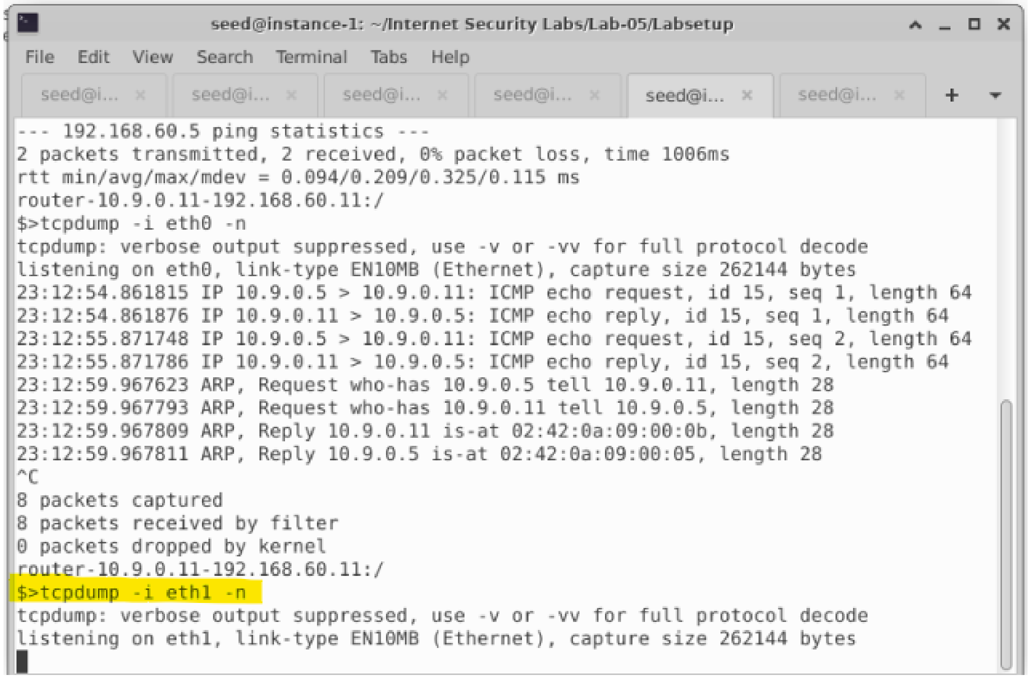
On the router side we will see from host U to the server, that icmp echo request then you'll get a reply from the server from the router to host U

The second ping request and the second ping reply and there are also some other packets ARP.

Now if u want to sniff the packets in the public network, so we need to sniff from ethernet 1 because if eth1 is interfaced to the private network, so just tap Ctrl+C To stop

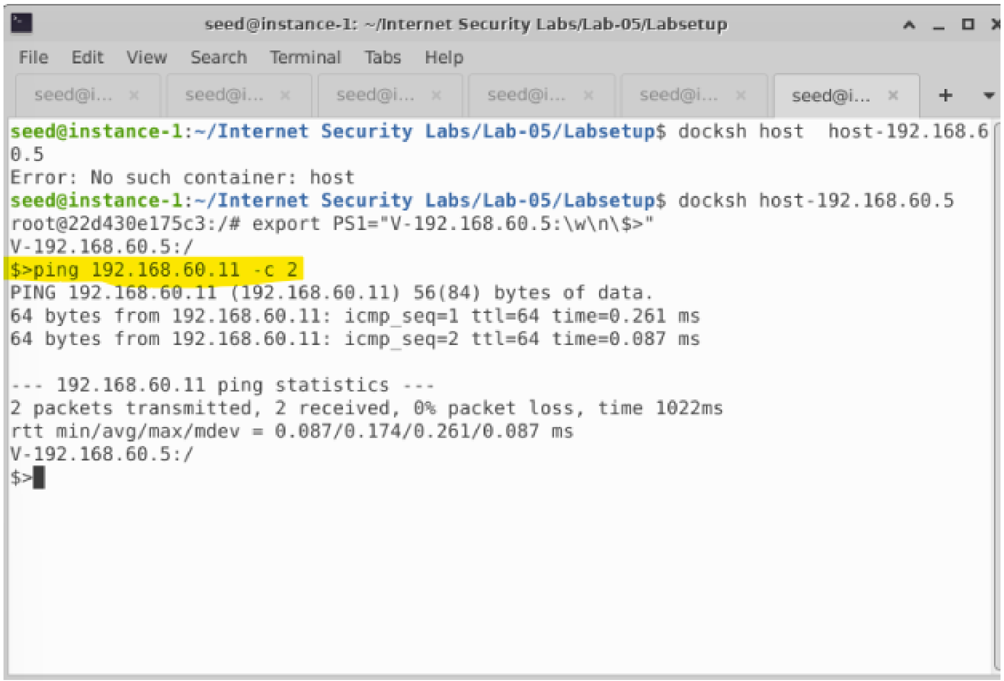

seed@instance-1: ~/Internet Security Labs/Lab-05/Labsetup
File Edit View Search Terminal Tabs Help
seed@... seed@... seed@... seed@... seed@... seed@...

```
2 packets transmitted, 2 received, 0% packet loss, time 1006ms
rtt min/avg/max/mdev = 0.094/0.209/0.325/0.115 ms
router-10.9.0.11-192.168.60.11:/
$>tcpdump -i eth0 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
23:12:54.861815 IP 10.9.0.5 > 10.9.0.11: ICMP echo request, id 15, seq 1, length
 64
23:12:54.861876 IP 10.9.0.11 > 10.9.0.5: ICMP echo reply, id 15, seq 1, length 6
4
23:12:55.871748 IP 10.9.0.5 > 10.9.0.11: ICMP echo request, id 15, seq 2, length
 64
23:12:55.871786 IP 10.9.0.11 > 10.9.0.5: ICMP echo reply, id 15, seq 2, length 6
4
23:12:59.967623 ARP, Request who-has 10.9.0.5 tell 10.9.0.11, length 28
23:12:59.967793 ARP, Request who-has 10.9.0.11 tell 10.9.0.5, length 28
23:12:59.967809 ARP, Reply 10.9.0.11 is-at 02:42:0a:09:00:0b, length 28
23:12:59.967811 ARP, Reply 10.9.0.5 is-at 02:42:0a:09:00:05, length 28
^C
8 packets captured
8 packets received by filter
0 packets dropped by kernel
router-10.9.0.11-192.168.60.11:/
$>
```


seed@instance-1: ~/Internet Security Labs/Lab-05/Labsetup
File Edit View Search Terminal Tabs Help
seed@i... seed@i... seed@i... seed@i... seed@i... seed@i...

```
--- 192.168.60.5 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1006ms
rtt min/avg/max/mdev = 0.094/0.209/0.325/0.115 ms
router-10.9.0.11-192.168.60.11:/
$>tcpdump -i eth0 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
23:12:54.861815 IP 10.9.0.5 > 10.9.0.11: ICMP echo request, id 15, seq 1, length 64
23:12:54.861876 IP 10.9.0.11 > 10.9.0.5: ICMP echo reply, id 15, seq 1, length 64
23:12:55.871748 IP 10.9.0.5 > 10.9.0.11: ICMP echo request, id 15, seq 2, length 64
23:12:55.871786 IP 10.9.0.11 > 10.9.0.5: ICMP echo reply, id 15, seq 2, length 64
23:12:59.967623 ARP, Request who-has 10.9.0.5 tell 10.9.0.11, length 28
23:12:59.967793 ARP, Request who-has 10.9.0.11 tell 10.9.0.5, length 28
23:12:59.967809 ARP, Reply 10.9.0.11 is-at 02:42:0a:09:00:0b, length 28
23:12:59.967811 ARP, Reply 10.9.0.5 is-at 02:42:0a:09:00:05, length 28
^C
8 packets captured
8 packets received by filter
0 packets dropped by kernel
router-10.9.0.11-192.168.60.11:/
$>tcpdump -i eth1 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
```

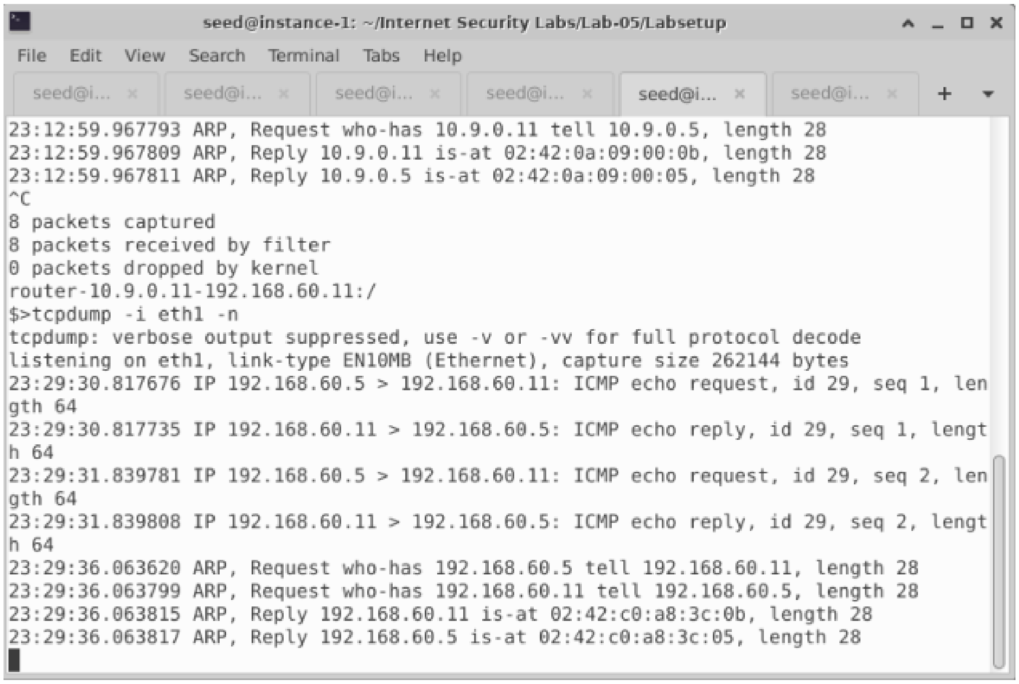
So now from host V we can ping the server, so this is the interface of the router interface to the public network


seed@instance-1: ~/Internet Security Labs/Lab-05/Labsetup
File Edit View Search Terminal Tabs Help
seed@i... seed@i... seed@i... seed@i... seed@i... seed@i...

```
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docksh host  host-192.168.6
0.5
Error: No such container: host
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docksh host-192.168.60.5
root@22d430e175c3:/# export PS1="V-192.168.60.5:\w\n\$>"
V-192.168.60.5:/
$>ping 192.168.60.11 -c 2
PING 192.168.60.11 (192.168.60.11) 56(84) bytes of data.
64 bytes from 192.168.60.11: icmp_seq=1 ttl=64 time=0.261 ms
64 bytes from 192.168.60.11: icmp_seq=2 ttl=64 time=0.087 ms

--- 192.168.60.11 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1022ms
rtt min/avg/max/mdev = 0.087/0.174/0.261/0.087 ms
V-192.168.60.5:/
$>
```

Two transmitted two received right on the server, you'll see from host V to the ………..(17:08)----- **Reminder**

Second request and second reply, Some packets try to ask for the physical address of this IP


seed@instance-1: ~/Internet Security Labs/Lab-05/Labsetup
File Edit View Search Terminal Tabs Help
seed@i... seed@i... seed@i... seed@i... seed@i... seed@i...

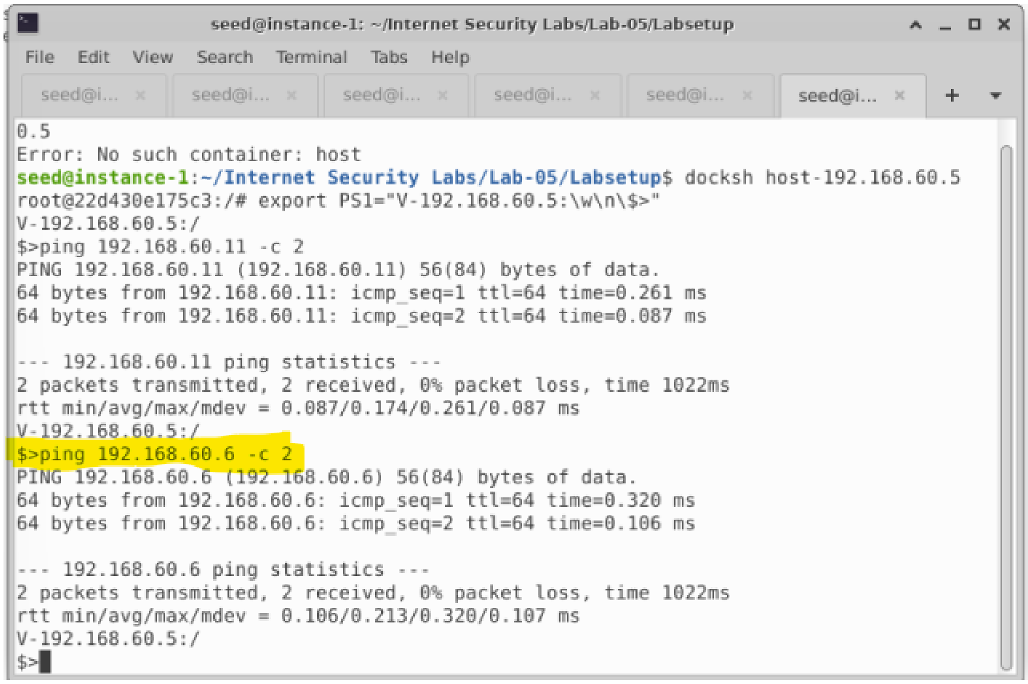
```
23:12:59.967793 ARP, Request who-has 10.9.0.11 tell 10.9.0.5, length 28
23:12:59.967809 ARP, Reply 10.9.0.11 is-at 02:42:0a:09:00:0b, length 28
23:12:59.967811 ARP, Reply 10.9.0.5 is-at 02:42:0a:09:00:05, length 28
^C
8 packets captured
8 packets received by filter
0 packets dropped by kernel
router-10.9.0.11-192.168.60.11:/
$>tcpdump -i eth1 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
23:29:30.817676 IP 192.168.60.5 > 192.168.60.11: ICMP echo request, id 29, seq 1, len
gth 64
23:29:30.817735 IP 192.168.60.11 > 192.168.60.5: ICMP echo reply, id 29, seq 1, lengt
h 64
23:29:31.839781 IP 192.168.60.5 > 192.168.60.11: ICMP echo request, id 29, seq 2, len
gth 64
23:29:31.839808 IP 192.168.60.11 > 192.168.60.5: ICMP echo reply, id 29, seq 2, lengt
h 64
23:29:36.063620 ARP, Request who-has 192.168.60.5 tell 192.168.60.11, length 28
23:29:36.063799 ARP, Request who-has 192.168.60.11 tell 192.168.60.5, length 28
23:29:36.063815 ARP, Reply 192.168.60.11 is-at 02:42:c0:a8:3c:0b, length 28
23:29:36.063817 ARP, Reply 192.168.60.5 is-at 02:42:c0:a8:3c:05, length 28
```

We can see tcp dump can sniff both networks but only those are packets transfer into and out of the router, we can slip rather, well as we just discussed moment ago it cannot be snipped for example,

Example:---Host V to another private host

.6 is another private host and the router will not be able to sniff this ping packets

Here 2 packets are transmitted and 2 received.

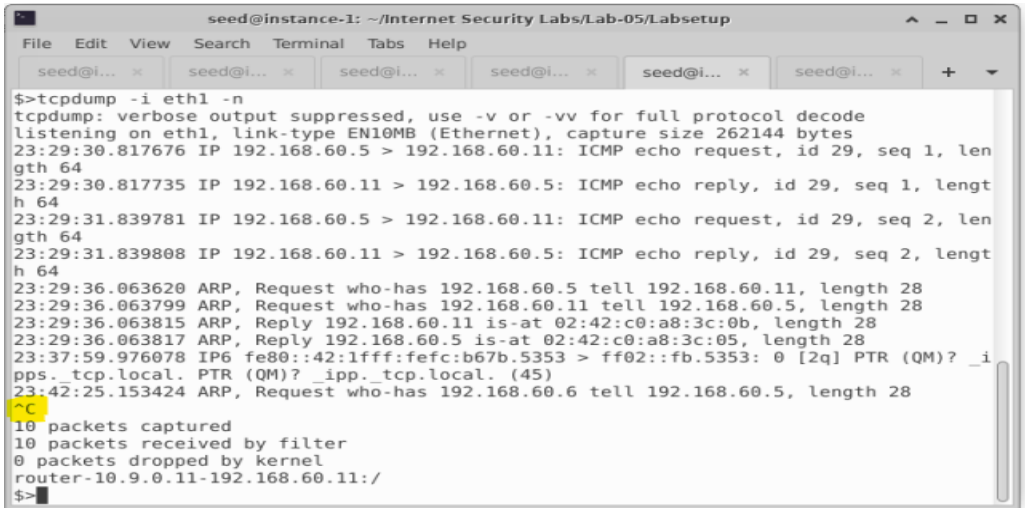

seed@instance-1: ~/Internet Security Labs/Lab-05/Labsetup
File Edit View Search Terminal Tabs Help
seed@i... ×
seed@i... ×
seed@i... ×
seed@i... ×
seed@i... ×
seed@i... ×

```
0.5
Error: No such container: host
seed@instance-1:~/Internet Security Labs/Lab-05/Labsetup$ docksh host-192.168.60.5
root@22d430e175c3:/# export PS1="V-192.168.60.5:\w\n\$>"
V-192.168.60.5:/
$>ping 192.168.60.11 -c 2
PING 192.168.60.11 (192.168.60.11) 56(84) bytes of data.
64 bytes from 192.168.60.11: icmp_seq=1 ttl=64 time=0.261 ms
64 bytes from 192.168.60.11: icmp_seq=2 ttl=64 time=0.087 ms

--- 192.168.60.11 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1022ms
rtt min/avg/max/mdev = 0.087/0.174/0.261/0.087 ms
V-192.168.60.5:/
$>ping 192.168.60.6 -c 2
PING 192.168.60.6 (192.168.60.6) 56(84) bytes of data.
64 bytes from 192.168.60.6: icmp_seq=1 ttl=64 time=0.320 ms
64 bytes from 192.168.60.6: icmp_seq=2 ttl=64 time=0.106 ms

--- 192.168.60.6 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1022ms
rtt min/avg/max/mdev = 0.106/0.213/0.320/0.107 ms
V-192.168.60.5:/
$>
```

Router didn’t see the ping packets, because these packets are not going to this router or out of this router, they just go from .file to .6

Host V to another private host .6

Now lets stop the tcp dump


seed@instance-1: ~/Internet Security Labs/Lab-05/Labsetup
File Edit View Search Terminal Tabs Help
seed@i... seed@i... seed@i... seed@i... seed@i... seed@i...

```
$>tcpdump -i eth1 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
23:29:30.817676 IP 192.168.60.5 > 192.168.60.11: ICMP echo request, id 29, seq 1, len
gth 64
23:29:30.817735 IP 192.168.60.11 > 192.168.60.5: ICMP echo reply, id 29, seq 1, lengt
h 64
23:29:31.839781 IP 192.168.60.5 > 192.168.60.11: ICMP echo request, id 29, seq 2, len
gth 64
23:29:31.839808 IP 192.168.60.11 > 192.168.60.5: ICMP echo reply, id 29, seq 2, lengt
h 64
23:29:36.063620 ARP, Request who-has 192.168.60.5 tell 192.168.60.11, length 28
23:29:36.063799 ARP, Request who-has 192.168.60.11 tell 192.168.60.5, length 28
23:29:36.063815 ARP, Reply 192.168.60.11 is-at 02:42:c0:a8:3c:0b, length 28
23:29:36.063817 ARP, Reply 192.168.60.5 is-at 02:42:c0:a8:3c:05, length 28
23:37:59.976078 IP6 fe80::42:1fff:fefc:b67b.5353 > ff02::fb.5353: 0 [2q] PTR (QM)? _i
pps._tcp.local. PTR (QM)? _ipp._tcp.local. (45)
23:42:25.153424 ARP, Request who-has 192.168.60.6 tell 192.168.60.5, length 28
^C
10 packets captured
10 packets received by filter
0 packets dropped by kernel
router-10.9.0.11-192.168.60.11:/
$>
```

TASK-2 Create and Configure TUN interface

Here we use the template tun.py provided, we can use read and write system calls to receive packets from or send packets to the virtual interface.

Create this tun and tab interface

(Code is already included in the volumes folder in the zip file and as we jus opened it here and tun.py)

Here we need to create tunnel interface then get the interface name and use while loop to keep the program running because we know there is a virtual interface, only exists during the time when the program is running, once the program is stopped, the virtual interface will disappear.

Task 2.a

Run and check the name of the interface(20:48)

Uou should be able to see a interface called tun 0

Now your job in this task is to change the time to pi program so instead of using time as the prefix of the interface name use your last name as prefix, the first file characters are your last name

For ex:------Your last name is smith, you should use smith as the prefix.

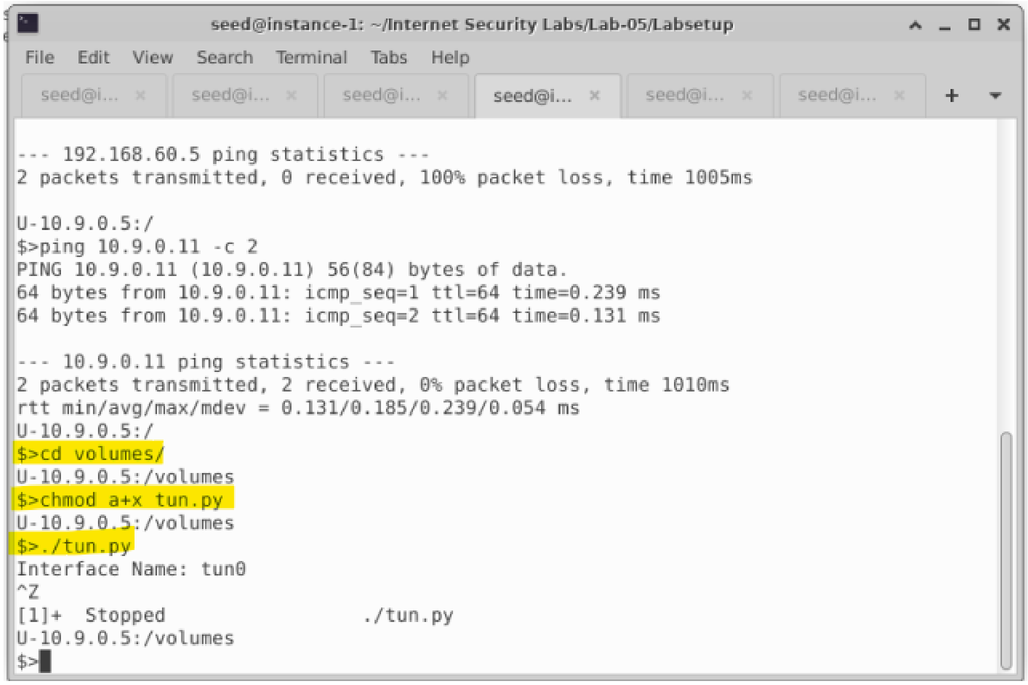
So first lets learn to find the time zero interface.

(We are asked to run tun.py program on Host U)

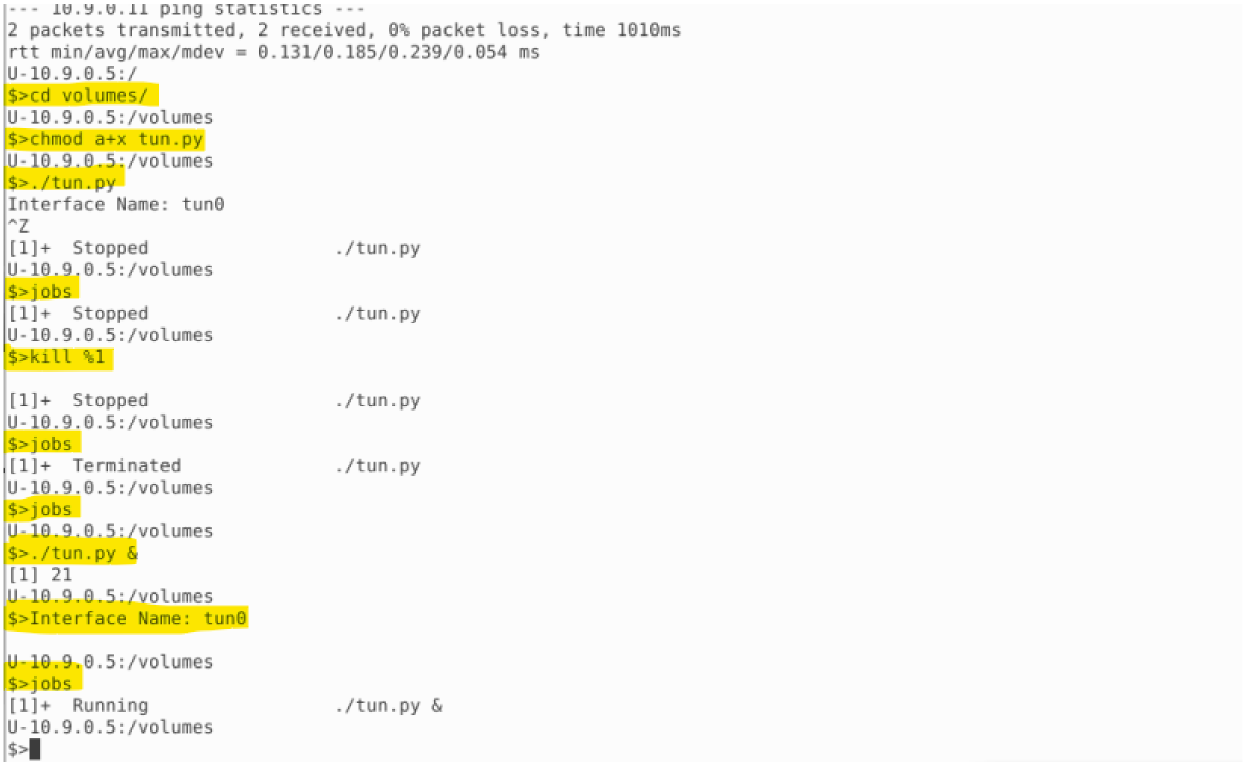
In the first view go to the volumes folder and you will see the tun.py,make it executable then run it, press enter you will see the interface name tun 0, now when you keep it running in that while loop, infinite while loop so lets stop it, press ctrl+C to stop it.

However we are asked to find the time interface how do we find it but now it keeps it running, so lets stop it and run in background.

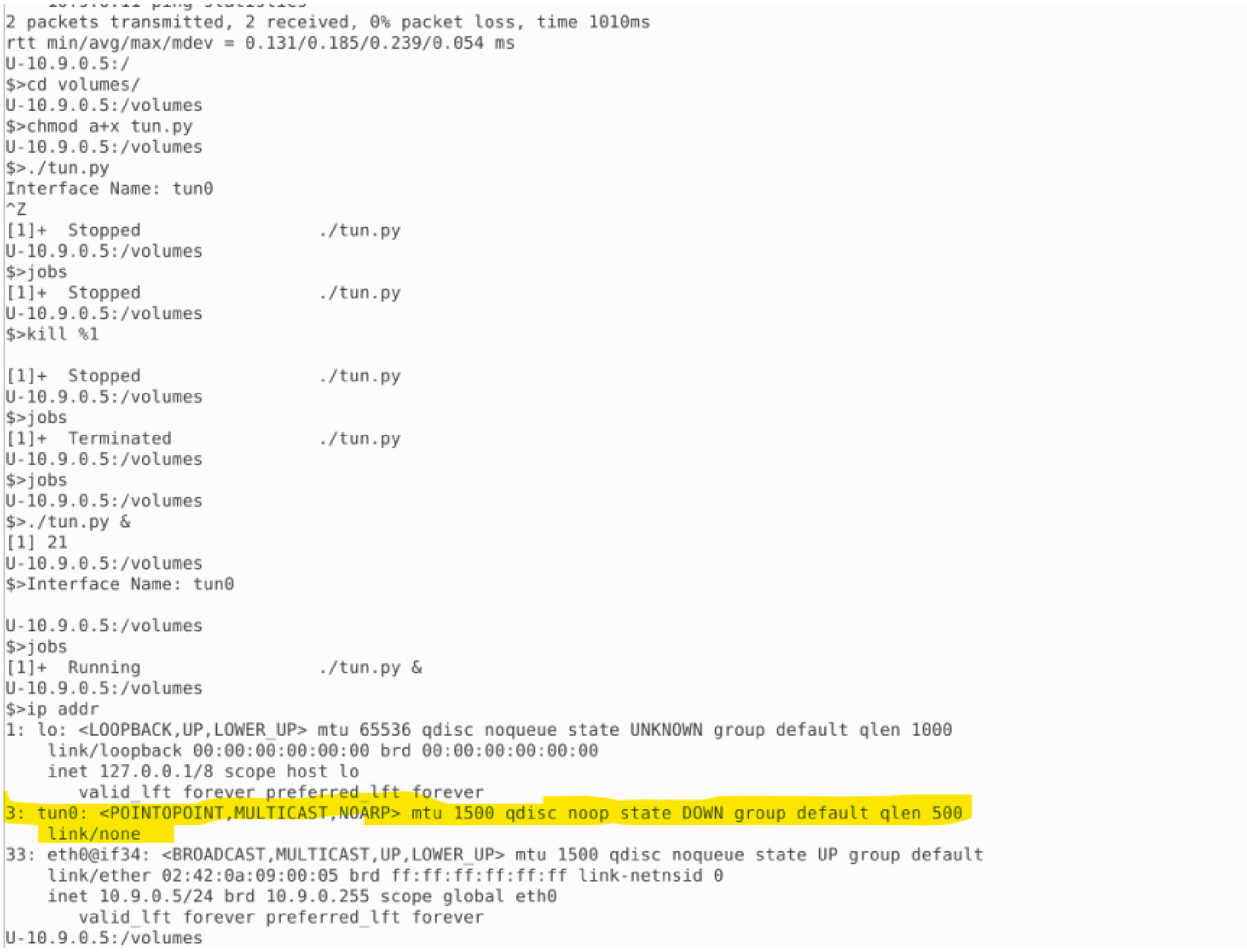
You can use jobs to see if its stopped lets declare it


seed@instance-1: ~/Internet Security Labs/Lab-05/Labsetup
File Edit View Search Terminal Tabs Help
seed@i... × seed@i... × seed@i... × seed@i... × seed@i... × seed@i... ×

```
--- 192.168.60.5 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1005ms

U-10.9.0.5:/
$>ping 10.9.0.11 -c 2
PING 10.9.0.11 (10.9.0.11) 56(84) bytes of data.
64 bytes from 10.9.0.11: icmp_seq=1 ttl=64 time=0.239 ms
64 bytes from 10.9.0.11: icmp_seq=2 ttl=64 time=0.131 ms

--- 10.9.0.11 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1010ms
rtt min/avg/max/mdev = 0.131/0.185/0.239/0.054 ms
U-10.9.0.5:/
$>cd volumes/
U-10.9.0.5:/volumes
$>chmod a+x tun.py
U-10.9.0.5:/volumes
$>./tun.py
Interface Name: tun0
^Z
[1]+  Stopped                 ./tun.py
U-10.9.0.5:/volumes
$>
```



```
--- 10.9.0.11 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1010ms
rtt min/avg/max/mdev = 0.131/0.185/0.239/0.054 ms
U-10.9.0.5:/
$>cd volumes/
U-10.9.0.5:/volumes
$>chmod a+x tun.py
U-10.9.0.5:/volumes
$>./tun.py
Interface Name: tun0
^Z
[1]+  Stopped                 ./tun.py
U-10.9.0.5:/volumes
$>jobs
[1]+  Stopped                 ./tun.py
U-10.9.0.5:/volumes
$>kill %1

[1]+  Stopped                 ./tun.py
U-10.9.0.5:/volumes
$>jobs
[1]+  Terminated              ./tun.py
U-10.9.0.5:/volumes
$>jobs
U-10.9.0.5:/volumes
$>./tun.py &
[1] 21
U-10.9.0.5:/volumes
$>Interface Name: tun0

U-10.9.0.5:/volumes
$>jobs
[1]+  Running                 ./tun.py &
U-10.9.0.5:/volumes
$>
```



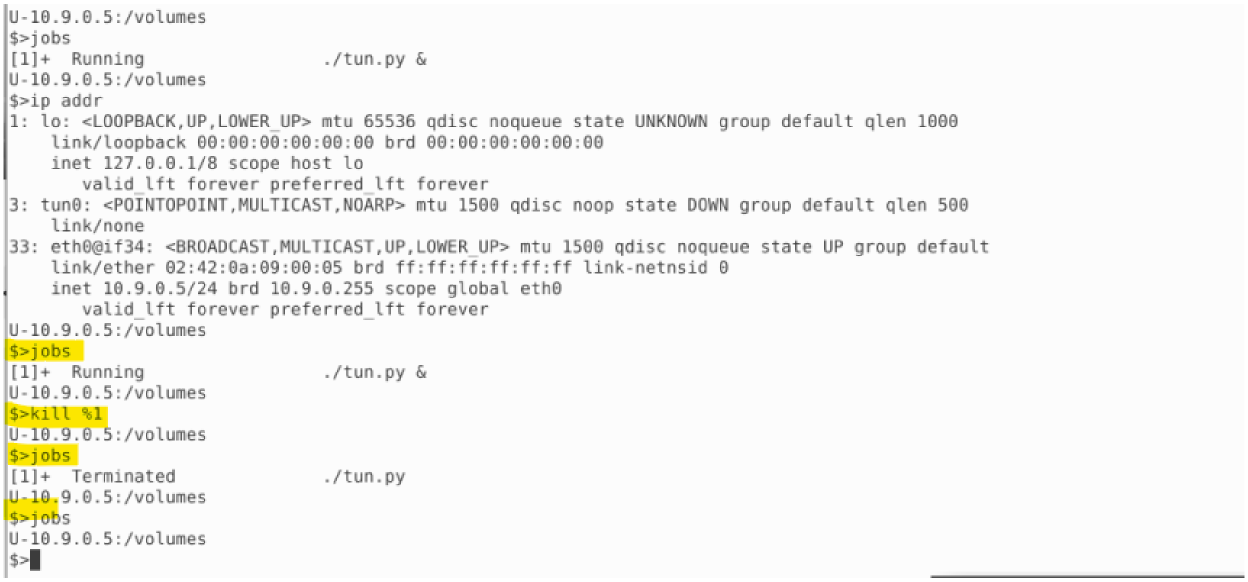
```
2 packets transmitted, 2 received, 0% packet loss, time 1010ms
rtt min/avg/max/mdev = 0.131/0.185/0.239/0.054 ms
U-10.9.0.5:/
$>cd volumes/
U-10.9.0.5:/volumes
$>chmod a+x tun.py
U-10.9.0.5:/volumes
$>./tun.py
Interface Name: tun0
^Z
[1]+  Stopped                 ./tun.py
U-10.9.0.5:/volumes
$>jobs
[1]+  Stopped                 ./tun.py
U-10.9.0.5:/volumes
$>kill %1

[1]+  Stopped                 ./tun.py
U-10.9.0.5:/volumes
$>jobs
[1]+  Terminated              ./tun.py
U-10.9.0.5:/volumes
$>jobs
U-10.9.0.5:/volumes
$>./tun.py &
[1] 21
U-10.9.0.5:/volumes
$>Interface Name: tun0

U-10.9.0.5:/volumes
$>jobs
[1]+  Running                 ./tun.py &
U-10.9.0.5:/volumes
$>ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
3: tun0: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop state DOWN group default qlen 500
    link/none
33: eth0@if34: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:0a:09:00:05 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.9.0.5/24 brd 10.9.0.255 scope global eth0
       valid_lft forever preferred_lft forever
U-10.9.0.5:/volumes
```

……………………………………………………………………………………………

………………………………………………………………………..

…………………………………………………………………………………(24:02)

So now lets stop it



```
U-10.9.0.5:/volumes
$>jobs
[1]+  Running                 ./tun.py &
U-10.9.0.5:/volumes
$>ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
3: tun0: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop state DOWN group default qlen 500
    link/none
33: eth0@if34: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:0a:09:00:05 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.9.0.5/24 brd 10.9.0.255 scope global eth0
       valid_lft forever preferred_lft forever
U-10.9.0.5:/volumes
$>jobs
[1]+  Running                 ./tun.py &
U-10.9.0.5:/volumes
$>kill %1
U-10.9.0.5:/volumes
$>jobs
[1]+  Terminated              ./tun.py
U-10.9.0.5:/volumes
$>jobs
U-10.9.0.5:/volumes
$>
```

Now you are asked to change the name from tun to your lastname first file calculator, so here you can see tun here inside this configuration struct and I change the tum to last contrast and save it