Internetworking Exercise

Offensive Technologies (145519)

Francisco Manuel Colmenar Lamas
219757

Trento, September 2020

Contents

List of Figures

1. INTERNETWORKING EXERCISE'S INSTRUCTIONS

1.1. Introduction.

In this exercise a mini-internetwork, which can be viewed as a small scale model of the Internet, is going to be used to experiment how is the setup of a network in which the equipment is connected but unconfigured.

The features of the network which are going to be configured are *IP addressing*, *route tables*, *private address prohibition*, *network address translation* and *port forwarding*.

Furthermore, the network model which is going to be used in this exercise is composed of five nodes. The following image displays a diagram of the network model.

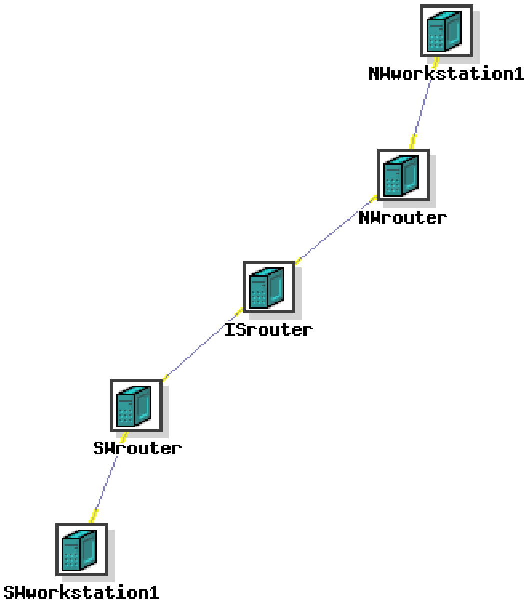


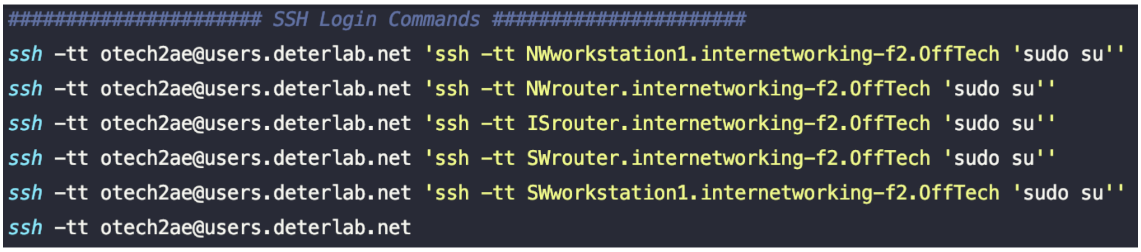
Figure 1.1: Network model used for this exercise.

Some important remarks about the nodes should be made in order to better understand the following steps. All of the five different nodes which are used in this exercise have five ethernet interfaces. However, it cannot be controlled which interfaces are being used or not. Therefore, special care needs to be given to the steps and scripts in which *ethX*, or similar, are used in order to choose the correct ethernet interface according to the current experiment.

1.2. Set up

To start with, in order to interact with the nodes it is needed to gain an interface to each of them. Consequently, to gain a terminal window interface to any of the nodes of the exercise a "double ssh" needs to be run by connecting first to users.isi.deterlab.net and from there to the desired node.

The commands needed to obtain the interface from the nodes are located inside the file *scripts/guide.sh*. It should be noted that the file *scripts/guide.sh* should not be executed as a single script, but instead as a reference for running auxiliary commands, mainly *ping* related commands to check the correctness of the already performed steps. The lines of code which are to be used at this step are the following ones.



```
###################### SSH Login Commands ######################
ssh -tt otech2ae@users.deterlab.net 'ssh -tt NWworkstation1.internetworking-f2.OffTech 'sudo su''
ssh -tt otech2ae@users.deterlab.net 'ssh -tt NWrouter.internetworking-f2.OffTech 'sudo su''
ssh -tt otech2ae@users.deterlab.net 'ssh -tt ISrouter.internetworking-f2.OffTech 'sudo su''
ssh -tt otech2ae@users.deterlab.net 'ssh -tt SWrouter.internetworking-f2.OffTech 'sudo su''
ssh -tt otech2ae@users.deterlab.net 'ssh -tt SWworkstation1.internetworking-f2.OffTech 'sudo su''
ssh -tt otech2ae@users.deterlab.net
```

Figure 1.2: SSH commands to login to each of the different nodes.

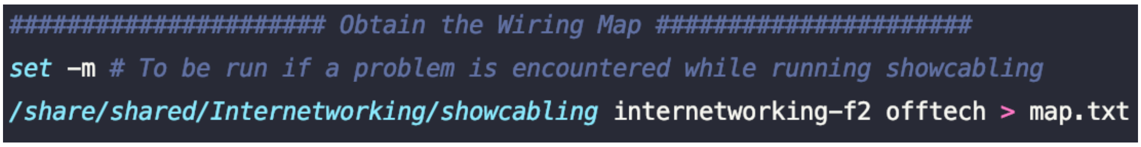
Furthermore, it is recommended to have a different terminal screen for each of the nodes for the ease of the execution of the following steps. Therefore, each of the lines of code of the previous image should be executed in a different terminal screen.

1.3. IP Addressing

In this step of the exercise the IP addresses of the five different nodes of the network are going to be provided with IP addresses. It should be stressed that the addresses need to be assigned to the cabled ethernet interfaces. Otherwise the subsequent steps of this exercise will not work correctly.

In order to determine which are the NICs which are cabled the script */share/shared/Internetworking/showcabling* needs to be executed from users.isi.deterlab.net.

Inside the file *scripts/guide.sh* the following commands can be used for this purpose.



```
###################### Obtain the Wiring Map ######################
set -m # To be run if a problem is encountered while running showcabling
/share/shared/Internetworking/showcabling internetworking-f2 offtech > map.txt
```

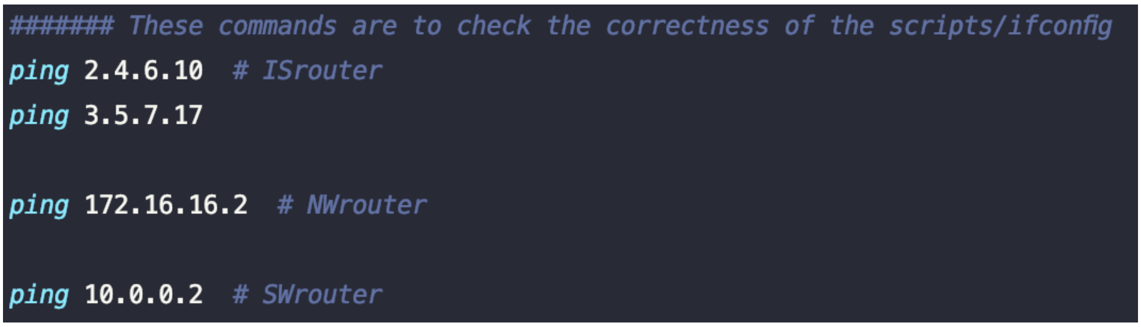
Figure 1.3: Command to obtain the "wiring map".

Two remarks about the above lines of code needs to be provided. First, the script *showcabling* needs two arguments in order to be executed correctly, which they are the experiment name and the project name. Second, the output of *"showcabling"* is saved into the file *map.txt* in order to facilitate its access during the execution of next steps of this exercise.

Once that the interfaces which are "wired" are known, the IPs used by each of the nodes can be assigned. In order to perform this assignment of addresses the scripts located in *"scripts/ifconfig/"* are provided.

Each of the scripts in the before mentioned folder need to be executed from a different node. For example, in ISrouter the script *"scripts/ifconfig/ISrouter.sh"* is the one to be executed.

Finally, to check the correctness of the IP addressing the links which have been created should be tested. Therefore, the following *ping* commands, which are located in the file *scripts/guide.sh*, should be executed in order to verify that all these pings get replies.



```
####### These commands are to check the correctness of the scripts/ifconfig
ping 2.4.6.10  # ISrouter
ping 3.5.7.17

ping 172.16.16.2  # NWrouter

ping 10.0.0.2  # SWrouter
```

Figure 1.4: Commands to ping and check the correctness of IP addressing.

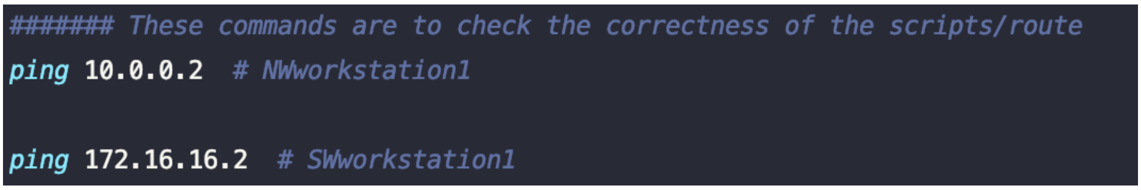
Attention should be given to the IP addresses used in the ping commands. In the case that the same IP addresses as in the *"scripts/ifconfig/"* scripts are the ones used, no change should be made to the *ping* commands. Otherwise, the specific IP addresses chosen needs to be the ones used in the above commands.

1.4. Route Tables

In this section of the exercise the different routing tables are going to be added in order to provide universal connectivity withing the different LANs from the current five node network model. The objective is to allow any node to dispatch a packet to any node from the network even if it does not directly belong to the same LAN.

In order to accomplish the addition of the routing tables, the scripts located in the folder *"scripts/route/"* are provided. As in the previous section, each of the different scripts need to be executed from the different nodes, such as *SWrooter.sh* needs to be executed from the node *SWrooter*.

Ultimately, to test that the route tables work as expected the following *ping* commands can be used.



```
####### These commands are to check the correctness of the scripts/route
ping 10.0.0.2  # NWworkstation1

ping 172.16.16.2  # SWworkstation1
```

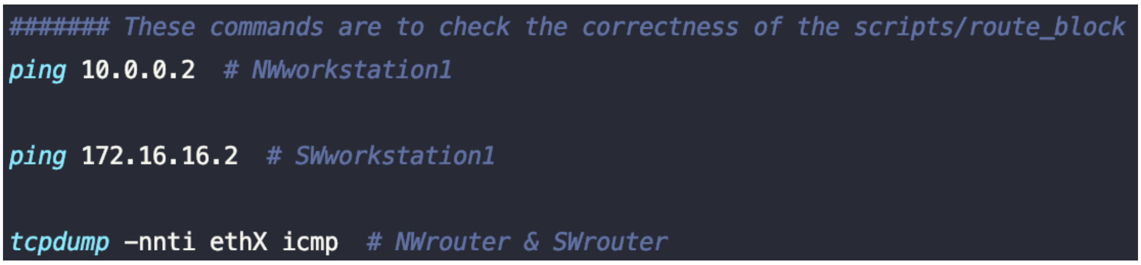
Figure 1.5: Commands to ping and check the correctness of the route tables.

1.5. Private Address Prohibition

In order to provide a more resemblance with the real internet, the private addresses such as 10., 192. and 172. are going to be prohibited in this section. To be able to accomplish this step, several *forward iptables* rules are going to added only at *ISrouter*.

As a consequence that only *ISrouter* needs to check that it does not forward any private address, only one script needs to be executed. The script is *scripts/route_block/ISrouter.sh* containing very straightforward rules which just drop any packet with the source or destination address belonging to a private network.

To test that the private addresses are not going through *ISrouter* the following *ping* commands can be executed.



```
####### These commands are to check the correctness of the scripts/route_block
ping 10.0.0.2  # NWworkstation1

ping 172.16.16.2  # SWworkstation1

tcpdump -nnti ethX icmp  # NWrouter & SWrouter
```

Figure 1.6: Commands to ping and check the correctness of private address prohibition.

In this case, opposed to the previous sections, the *ping* commands will not receive any response meaning that the private address prohibition has been implemented correctly.

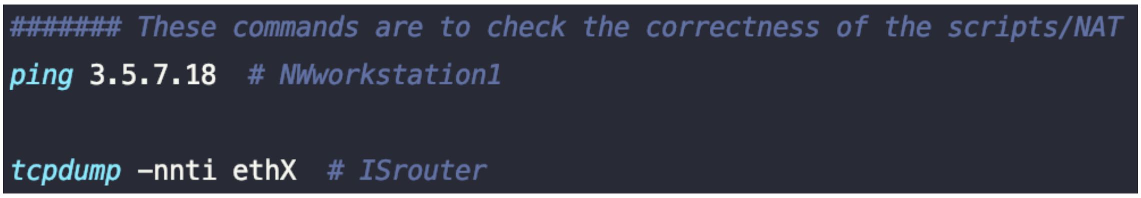
1.6. Network Address Translation

The next step is to perform a Network Address Translation or NAT. The objective of this technique is to overcome the issue originated from the prohibition of private addresses through the *ISrouter*.

The main idea of NATs is to provide the operationally valid public IP addresses from the Internet-facing routers to the nodes which are private in order for them to be able to connect to the Internet.

In order to implement the NAT in the current network model two scripts needs to be executed, one in each of the "Internet-facing routers", which they are *NWrouter* and *SWrouter*. The scripts are inside the folder *scripts/NAT/*.

To check the validity of the created NATs the following *ping* command can be used.



```
####### These commands are to check the correctness of the scripts/NAT
ping 3.5.7.18  # NWworkstation1

tcpdump -nnti ethX  # ISrouter
```

Figure 1.7: Commands to ping and check the correctness of NAT.

In the case that the NAT has been implemented correctly the previous *pings* commands will provide a response.

1.7. Port Forwarding

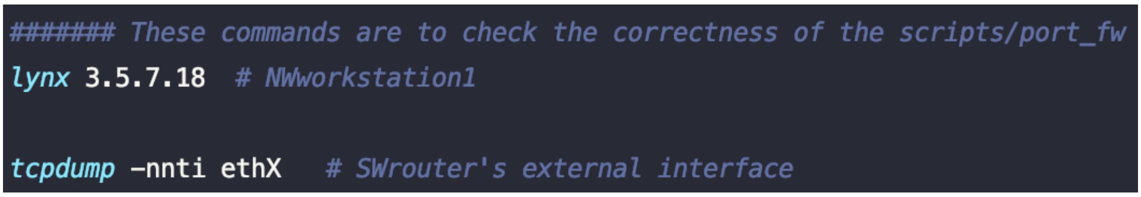
Finally the last step of this exercise is to implement *port forwarding* in order to be able to solve the end-to-end problem based originated by the NAT.

In this case, *SWworkstation1* has the Apache web server running on it and *NWworstation1* has the lynx web browser installed on it. The objective is to allow *NWworstation1* browse *SWworkstation1*.

To accomplish it taking into account that no private address can be sent, such as a packet directly addressed to *SWworkstation1* from *NWworstation1*, *SWrouter* will identify that the packets addressed to it are in fact intended for *NWworstation1*, identifiable by the TCP header. Therefore, *SWrouter* will produce copies of those packets and send them to *SWworkstation1* using its private address.

Only one script needs to be run by *SWrouter*, as the web server is running at *SWworkstation1*, which is *scripts/port_fw/SWrouter.sh*.

To verify that the port forwarding is working correctly the following command running the web browser *lynx* can be run.



```
####### These commands are to check the correctness of the scripts/port_fw
lynx 3.5.7.18  # NWworkstation1

tcpdump -nnti ethX   # SWrouter's external interface
```

Figure 1.8: Commands to run the web browser and check the correctness of port forwarding.

If the previous steps have been done correctly, *NWworstation1* would have to display a success page which has been received by *SWworstation1*.