INTRANETWORKING

Vinci Nicolò 220229

nicolo.vinci@studenti.unitn.it

Offensive Technologies 2021/2022

24 September 2021

Contents

1 Solution

The exercise consists of assigning IP addresses, defining routes, configuring NAT and port forwarding. The network topology with custom interfaces names is reported in figure 1.

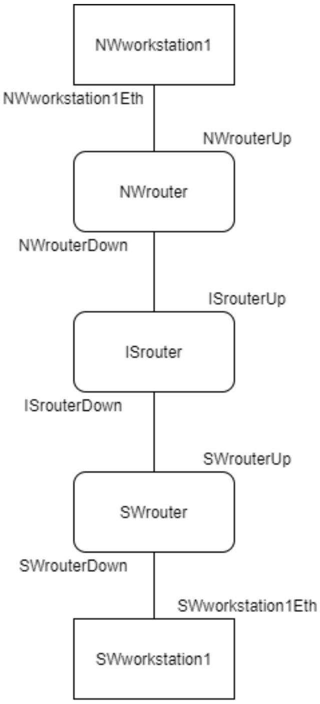
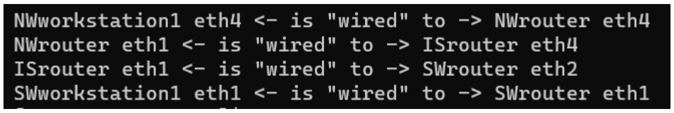


Figure 1: Network topology

A bash script has been developed in order to automate the exercise solution. The script has been called *config.sh* and it can be run with *./config.sh* in the Deterlab machine of *otech2ah* user. However, each network machine in the topology has a lot of interfaces. So, the following command executes a script to discover the actual linked interfaces.

```
$ /share/shared/Internetworking/showcabling intranetworking—vn offtech
```

The output of the script is reported at figure 2.



```
NWworkstation1 eth4 <- is "wired" to -> NWrouter eth4
NWrouter eth1 <- is "wired" to -> ISrouter eth4
ISrouter eth1 <- is "wired" to -> SWrouter eth2
SWworkstation1 eth1 <- is "wired" to -> SWrouter eth1
```

Figure 2: Output script

The output has been parsed exploiting *sed* and *tr* command and saved in the file *eth.txt*. Regarding the output, the piece of string <- *is "wired" to* -> has been substituted with the character ˆ thanks to the first *sed*. Then, the character ˆ has been transformed into new line \n with the command *tr*. Then, the file has been analyzed exploiting *sed* and *cut* in order to retrieve the interfaces and assign them to the corresponding variables in the bash script. Each line of the file *eth.txt* have been analyzed with *sed* and each interface has been retrieved with *cut* command exploiting the blank space. After that, one bash script for each machine has been prepared in order to assign IP addresses, routes, configuring NAT and port forwarding. IP addresses assignment is reported in table 1.

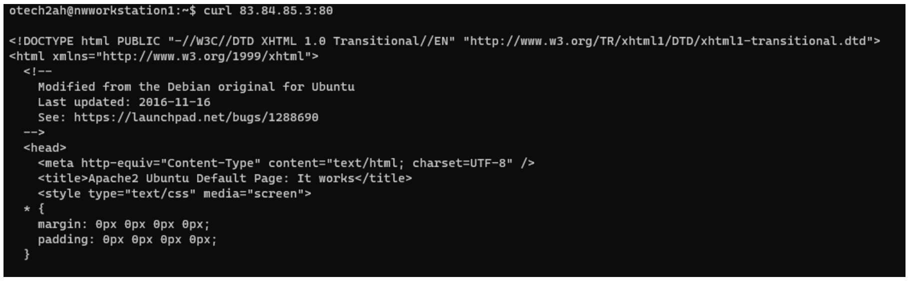
	IP address	Netmask
NWworkstation1Eth	10.10.1.2	255.255.255.0
NWrouterUp	10.10.1.1	255.255.255.0
NWrouterDown	93.94.95.3	255.255.255.0
ISrouterUp	93.94.95.2	255.255.255.0
ISrouterDown	83.84.85.2	255.255.255.0
SWrouterUp	83.84.85.3	255.255.255.0
SWrouterDown	172.16.1.1	255.255.255.0
SWworkstation1Eth	172.16.1.2	255.255.255.0

Table 1: IP addressing

Each script is made executable and it will be launched to the right machine via *ssh* command
ssh [machine name].intranetworking-vn.offtech ”sudo su -c ./[script name].sh”

2 Results

Results are reported in this section.
First of all, the *NWworkstation1* is able to perform the http request to Apache web server located on the *SWworkstation1* thanks to the NAT performed by *NWrouter* and *SWrouter* and the port forwarding performed by *SWrouter*.

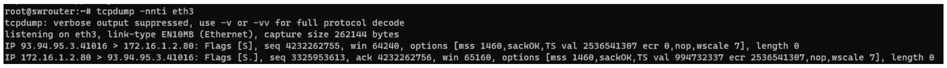


```
otech2ah@nwworkstation1:~$ curl 83.84.85.3:80

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <!--
    Modified from the Debian original for Ubuntu
    Last updated: 2016-11-16
    See: https://launchpad.net/bugs/1288690
  -->
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Apache2 Ubuntu Default Page: It works</title>
    <style type="text/css" media="screen">
  * {
    margin: 0px 0px 0px 0px;
    padding: 0px 0px 0px 0px;
  }
```

Figure 3: Curl request by NWworkstation1

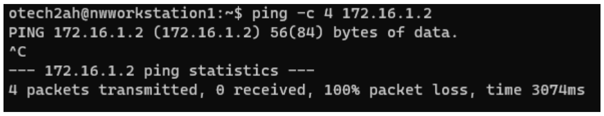
The http request performed by *NWworkstation1* is forwarded by *SWrouter* directly to the *SWworkstation1* thanks to the port forwarding defined. Indeed, in the figure 4 the *SWworkstation1* replies to *93.94.95.3*, which is the IP address of *NWworkstation1* natted by *NWrouter*.



```
root@swrouter:~# tcpdump -nnti eth3
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth3, link-type EN10MB (Ethernet), capture size 262144 bytes
IP 93.94.95.3.41016 > 172.16.1.2.80: Flags [S], seq 4232262755, win 64240, options [mss 1460,sackOK,TS val 2536541307 ecr 0,nop,wscale 7], length 0
IP 172.16.1.2.80 > 93.94.95.3.41016: Flags [S.], seq 3325953613, ack 4232262756, win 65160, options [mss 1460,sackOK,TS val 994732337 ecr 2536541307,nop,wscale 7], length 0
```

Figure 4: SWworkstation1 replies to natted NWworkstation1

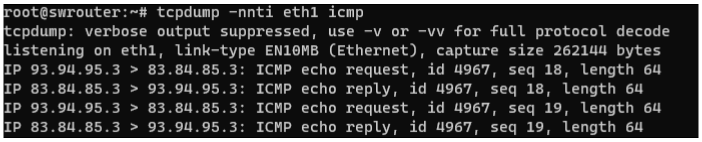
However, the *NWworkstation1* can not directly ping the *SWworkstation1*, because the destination address of the latter is blocked by the *ISrouter*.



```
otech2ah@nwworkstation1:~$ ping -c 4 172.16.1.2
PING 172.16.1.2 (172.16.1.2) 56(84) bytes of data.
^C
--- 172.16.1.2 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3074ms
```

Figure 5: Ping from NWorkstation1 directly to SWworkstation1

Anyway, the *NWworkstation1* is able to ping the *SWrouter* thanks to the NAT perfomed by *NWrouter* and also the *SWworkstation1* is able to ping the *NWrouter* thanks to the NAT performed by *SWrouter*. Indeed, in figure 6 the *SWrouter* is able to reply to ping requests performed by *NWworkstation1*. The *93.94.95.3* address is the natted IP of *NWworkstation1*.



```
root@swrouter:~# tcpdump -nnti eth1 icmp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
IP 93.94.95.3 > 83.84.85.3: ICMP echo request, id 4967, seq 18, length 64
IP 83.84.85.3 > 93.94.95.3: ICMP echo reply, id 4967, seq 18, length 64
IP 93.94.95.3 > 83.84.85.3: ICMP echo request, id 4967, seq 19, length 64
IP 83.84.85.3 > 93.94.95.3: ICMP echo reply, id 4967, seq 19, length 64
```

Figure 6: Ping from NWworkstation1 to SWrouter

3 Improvements

Some suggestions to improve the bash script.
The *config.sh* script checks only if the file *eth.txt* exists. However, there are no controls if some routes or iptables rules already exist in the script launched for a machine in the network. So, routes and iptables should be checked before adding new routes or iptables rules. Indeed, if a script is trying to add an already existed route, there will be an error. Instead, if a script wants to add an already existed iptables rule, there will be two equal rules in the corresponding chain of the iptables.
Then, it may be boring to swap in and swap out continuously to refresh machines in order to test the *config.sh* script. So, it may design a new script to clean IP addresses, routes and iptables automatically.