ET2595 Network and System Security

Linux Networking and Firewalls

Laboratory 1

(**Date: 20th November 2022**)

Gattupalli Monica

(moga20@student.bth.se)

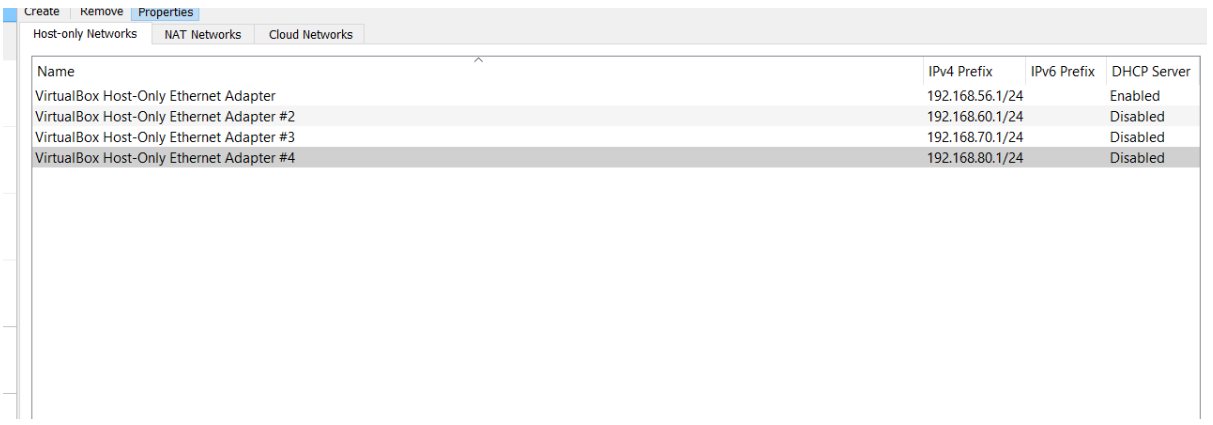
(19991130-5002)

Introduction

To achieve the main goal of the Assignment and get familiar with the basic Linux utilities and diagnose the network system and firewall, we need to set the lab environment in our main OS.

In the lab environment virtual box appliance is provided that is need to be downloaded and extract which gives the 4 ubuntu guest OS named as server A, client A, server B, client B. The setting of the VirtualBox needs updated by setting the network as Host-only networks.

We need to create the 3 host only network with the IP address as 192.168.60.1, 192.168.70.1,192.168.80.1 and the net mask is same for 3 host only networks which is 255.255.255.0.



Name	IPv4 Prefix	IPv6 Prefix	DHCP Server
VirtualBox Host-Only Ethernet Adapter	192.168.56.1/24		Enabled
VirtualBox Host-Only Ethernet Adapter #2	192.168.60.1/24		Disabled
VirtualBox Host-Only Ethernet Adapter #3	192.168.70.1/24		Disabled
VirtualBox Host-Only Ethernet Adapter #4	192.168.80.1/24		Disabled

Figure 1: Adding a Host only networks

Virtual box, Wireshark, putty tools are used in the achieving the goal of the assignment.

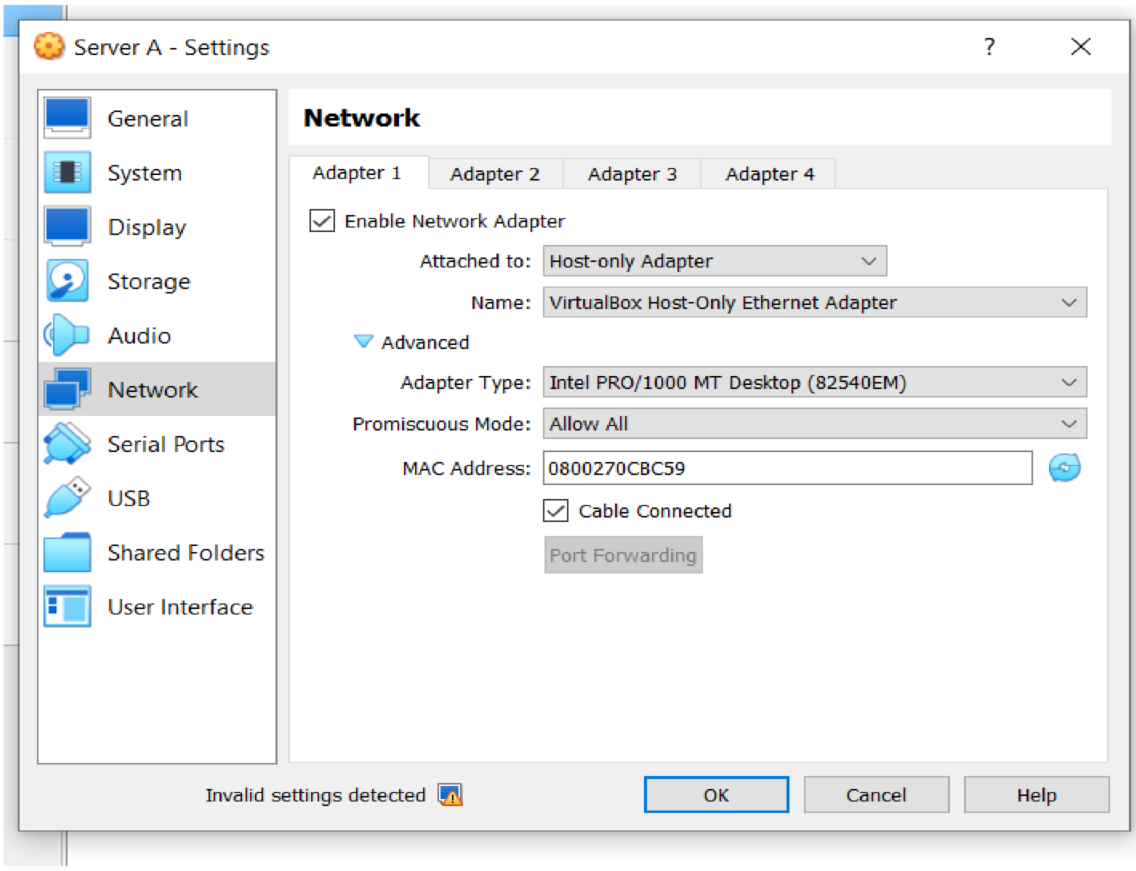
Task-1: MAC Address

To get the MAC address of the adapters we need to open the settings and go to networks then later click on the Advanced to get the MAC address.

The below tables give the MAC address of the created Adapters

Adapters	MAC address
Adapter 1	0800270CBC59
Adapter 2	080027FC27F5
Adapter 3	0800272841E0

The below are screenshots of the 3 Adapters which have the MAC address.



☑ Enable Network Adapter

Attached to: Host-only Adapter

Name: VirtualBox Host-Only Ethernet Adapter

Advanced

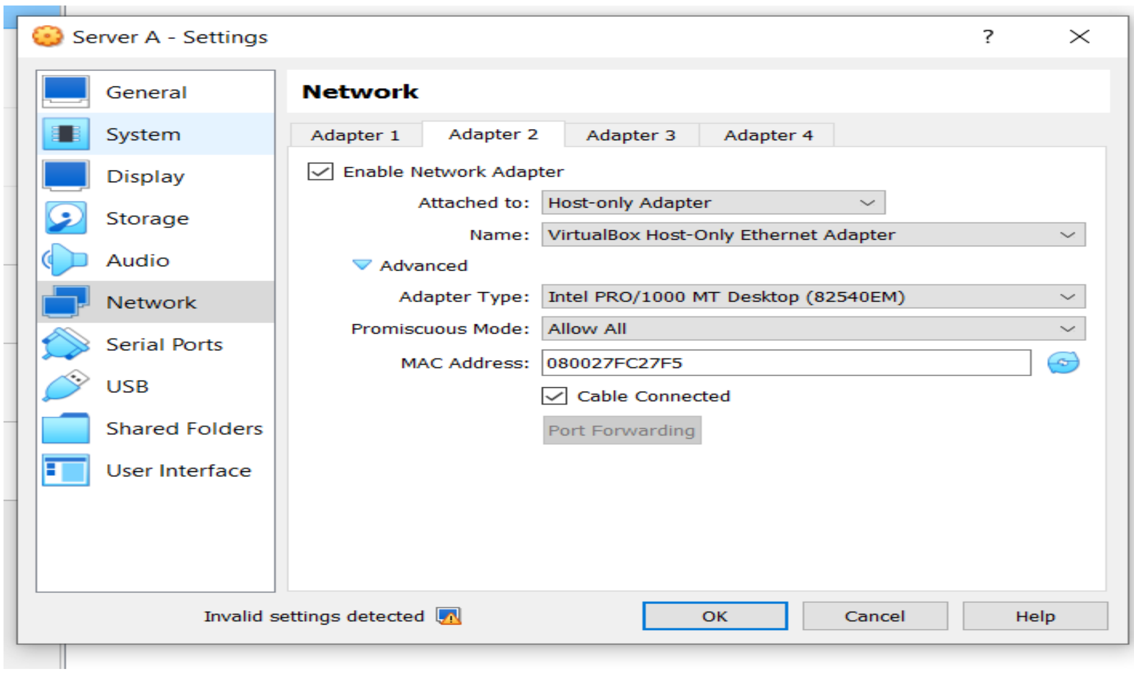
Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

Promiscuous Mode: Allow All

MAC Address: 0800270CBC59

☑ Cable Connected

Figure-2: MAC address of the Adapter 1.



☑ Enable Network Adapter

Attached to: Host-only Adapter

Name: VirtualBox Host-Only Ethernet Adapter

Advanced

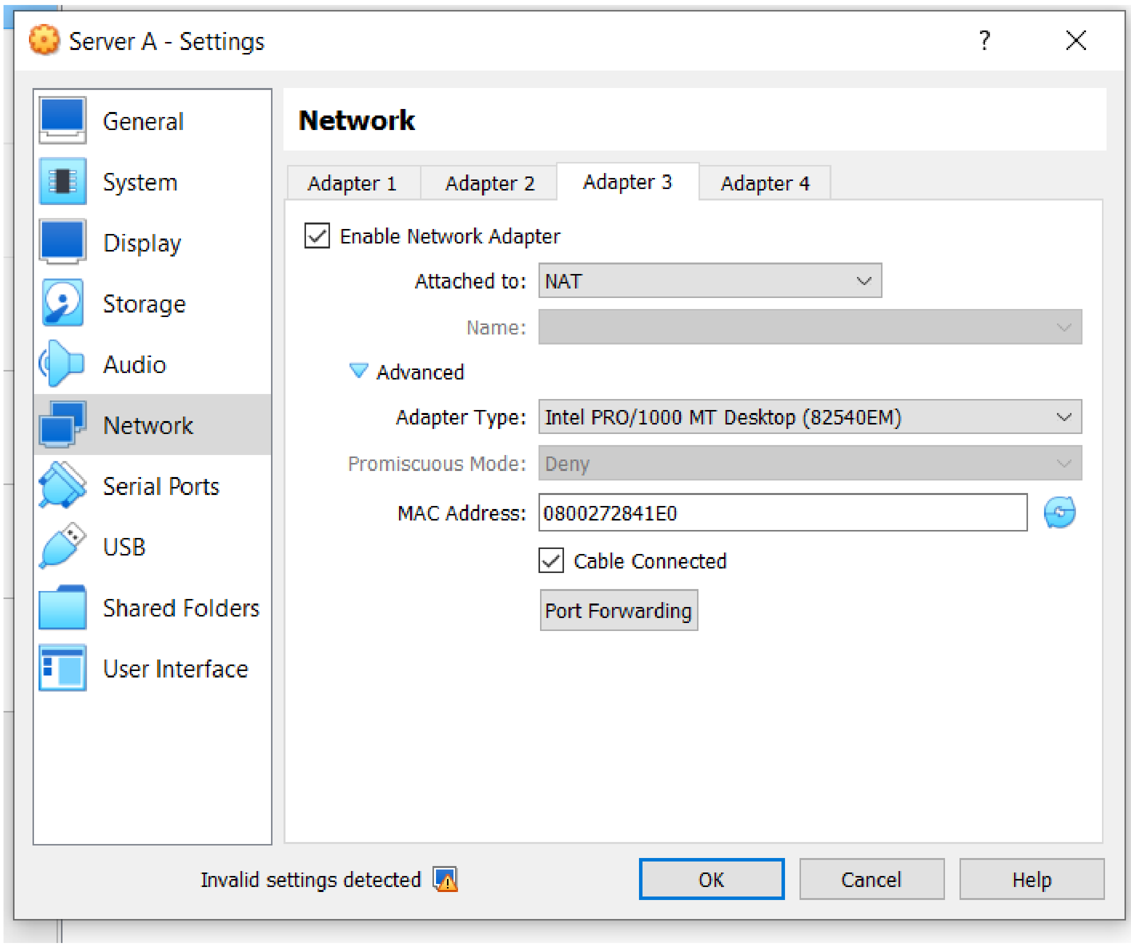
Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

Promiscuous Mode: Allow All

MAC Address: 080027FC27F5

☑ Cable Connected

Figure-3: MAC address in Adapter 2


Server A - Settings

Network

Adapter 1 | Adapter 2 | Adapter 3 | Adapter 4

☑ Enable Network Adapter

Attached to: NAT

Name:

Advanced

Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

Promiscuous Mode: Deny

MAC Address: 0800272841E0

☑ Cable Connected

Port Forwarding

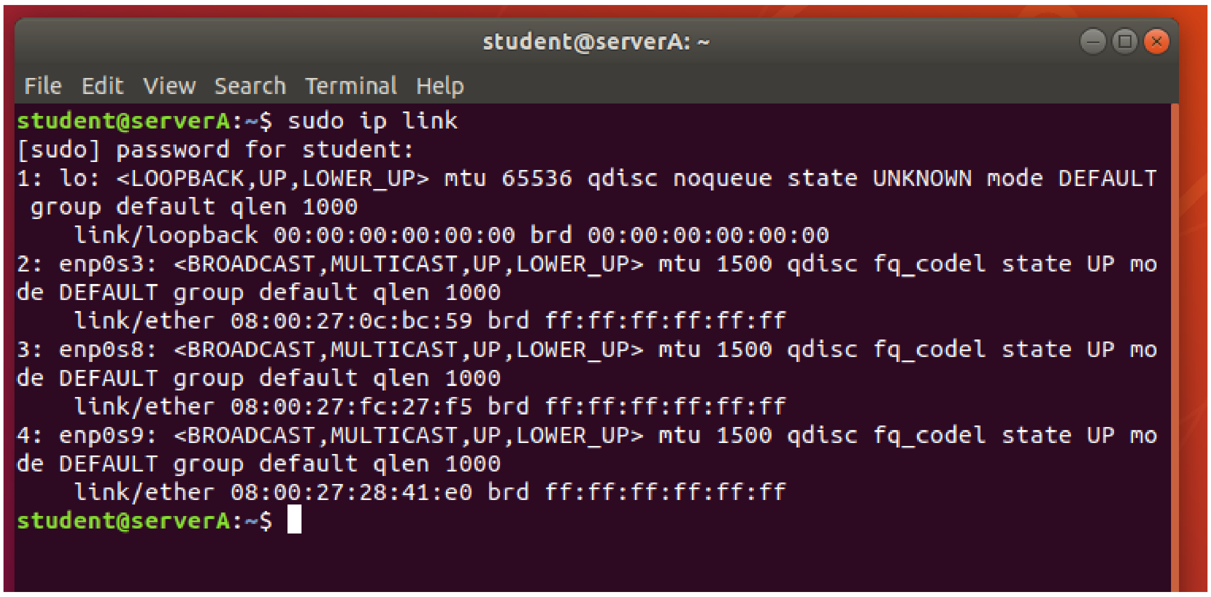
Invalid settings detected

Figure-4: MAC address in Adapter 3.

Task-2: Network Interface

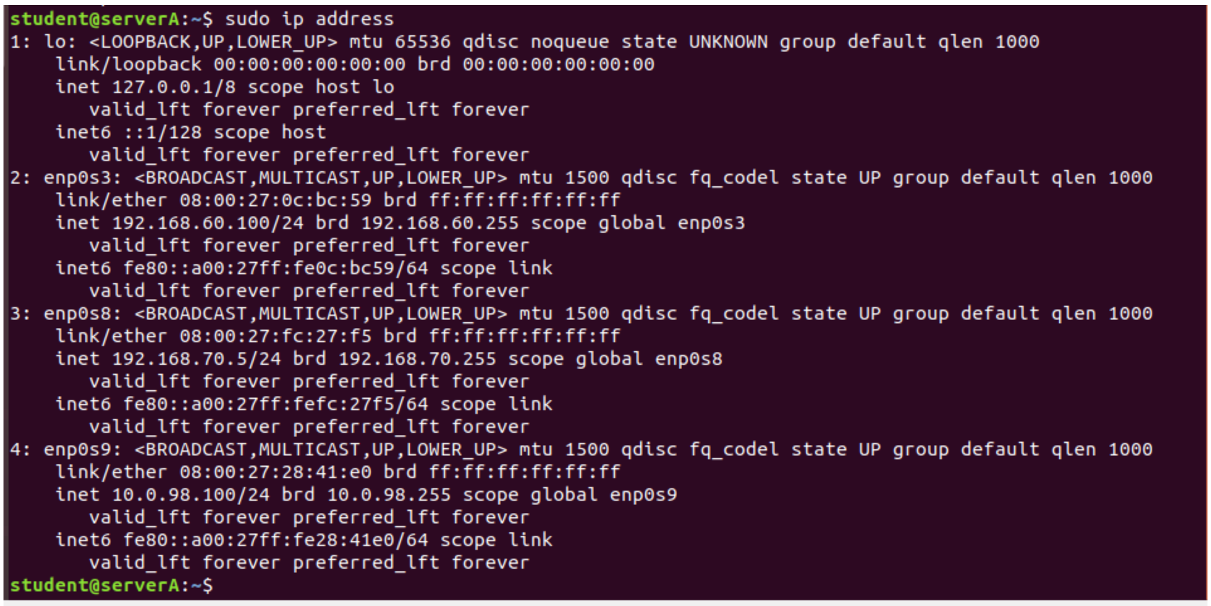
In the terminal **“sudo ip link”** command is entered. After entering command we can see 3 MAC addresses in which 2 are Host only network and one NAT interface.

1. enp0s9 is NAT interface.
2. enp0s3 is Host only interface.
3. enp0s8 is Host only interface.



```
student@serverA:~$ sudo ip link
[sudo] password for student:
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
 group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mo
de DEFAULT group default qlen 1000
    link/ether 08:00:27:0c:bc:59 brd ff:ff:ff:ff:ff:ff
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mo
de DEFAULT group default qlen 1000
    link/ether 08:00:27:fc:27:f5 brd ff:ff:ff:ff:ff:ff
4: enp0s9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mo
de DEFAULT group default qlen 1000
    link/ether 08:00:27:28:41:e0 brd ff:ff:ff:ff:ff:ff
student@serverA:~$
```

Figure-5: output of the command “sudo ip link”.



```
student@serverA:~$ sudo ip address
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
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:0c:bc:59 brd ff:ff:ff:ff:ff:ff
    inet 192.168.60.100/24 brd 192.168.60.255 scope global enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe0c:bc59/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:fc:27:f5 brd ff:ff:ff:ff:ff:ff
    inet 192.168.70.5/24 brd 192.168.70.255 scope global enp0s8
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fefc:27f5/64 scope link
       valid_lft forever preferred_lft forever
4: enp0s9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:28:41:e0 brd ff:ff:ff:ff:ff:ff
    inet 10.0.98.100/24 brd 10.0.98.255 scope global enp0s9
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe28:41e0/64 scope link
       valid_lft forever preferred_lft forever
student@serverA:~$
```

Figure-6: output of the command “sudo ip address”

Task-3: IP address, netmask, and subnet

To derive the subnets we need to perform the and operation between the IP address and the Net mask by converting them to binary format.

In the AND operation if the both the inputs are 1 then the result will be in remaining all cases the output will be 0.

Net mask for all IP address is 255.255.255.0.

1. enp0s3: The Ip address is 192.168.60.100.

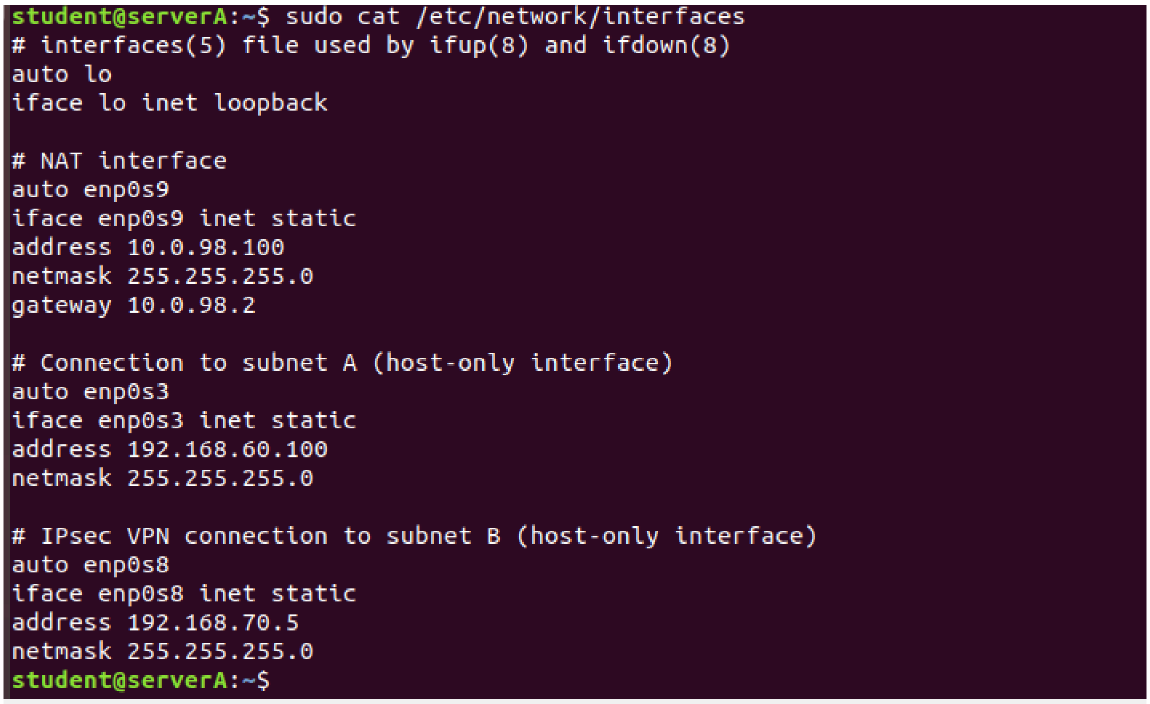
IP address	192.168.60.100	1100 0000.1010 1000.0011 1100.0110 0100
Net Mask	255.255.255.0	1111 1111.1111 1111.1111 1111.0000 0000
subnet	192.168.60.0	1100 0000.1010 1000.0011 1100.0000 0000

2. enp0s8: The Ip address is 192.168.70.5.

IP address	192.168.70.5	1100 0000.1010 1000.0100 0110.0000 0101
Net Mask	255.255.255.0	1111 1111.1111 1111.1111 1111.0000 0000
subnet	192.168.70.0	1100 0000.1010 1000.0100 0110.0000 0000

3. enp09: The Ip address is 10.0.98.100.

IP address	10.0.98.100	1100 0000.0000 0000.0110 0010.0110 0100
Net Mask	255.255.255.0	1111 1111.1111 1111.1111 1111.0000 0000
subnet	10.0.98.0	1100 0000.0000 0000.0110 0010.0000 0000



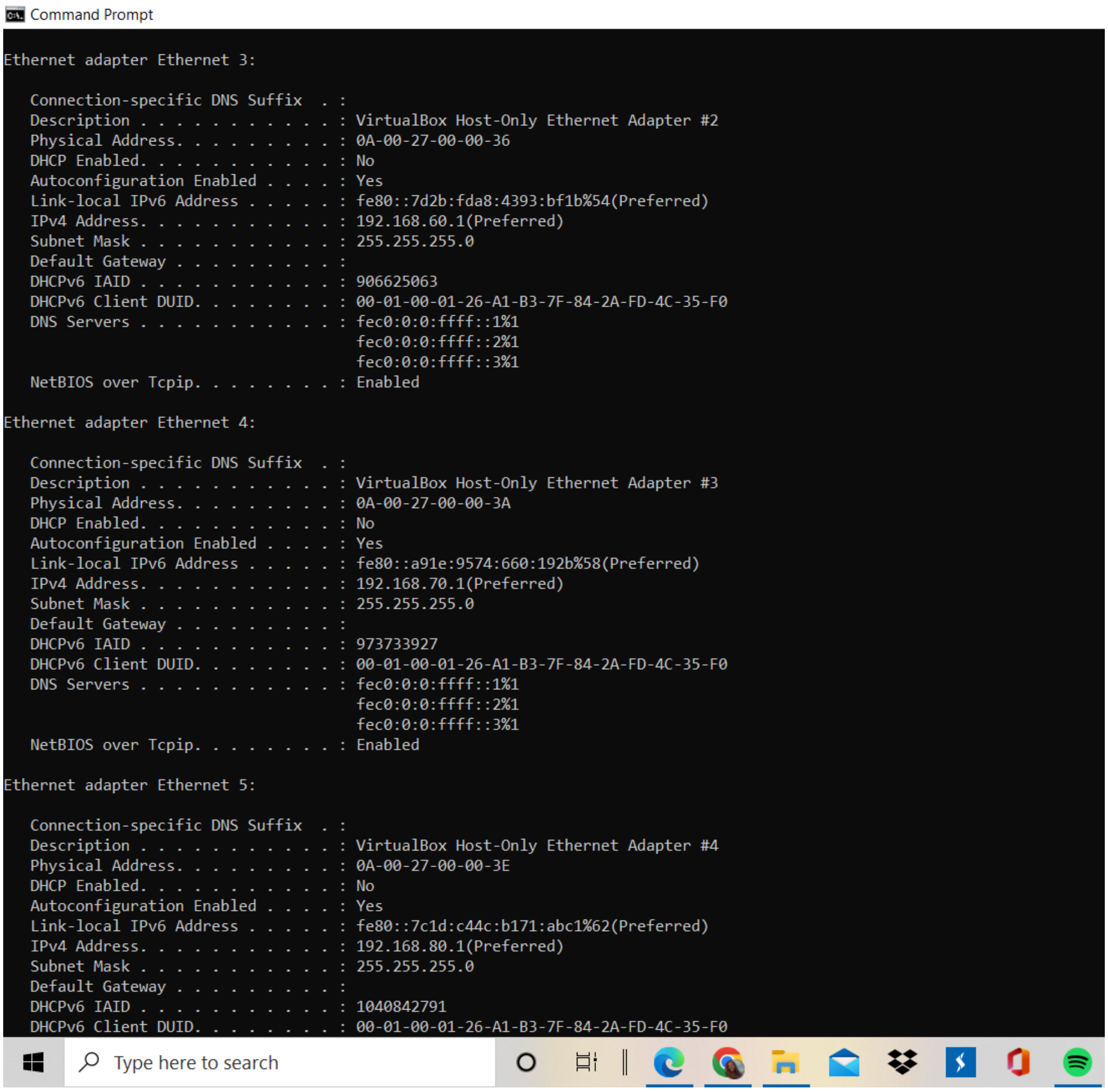
```
student@serverA:~$ sudo cat /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback

# NAT interface
auto enp0s9
iface enp0s9 inet static
address 10.0.98.100
netmask 255.255.255.0
gateway 10.0.98.2

# Connection to subnet A (host-only interface)
auto enp0s3
iface enp0s3 inet static
address 192.168.60.100
netmask 255.255.255.0

# IPsec VPN connection to subnet B (host-only interface)
auto enp0s8
iface enp0s8 inet static
address 192.168.70.5
netmask 255.255.255.0
student@serverA:~$
```

Figure-7: viewing the interfaces in the server A

Task-4: Host only interfaces

I am using windows Host so I entered the **“ipconfig/all”** command in the command prompt of the main host, in the below figure we can see the IP address and the subnet mask.

From the below figure we can understand that the Adapters are connected to the Host only interfaces of Guest OS (192.168.60.0, 192.168.70.0).

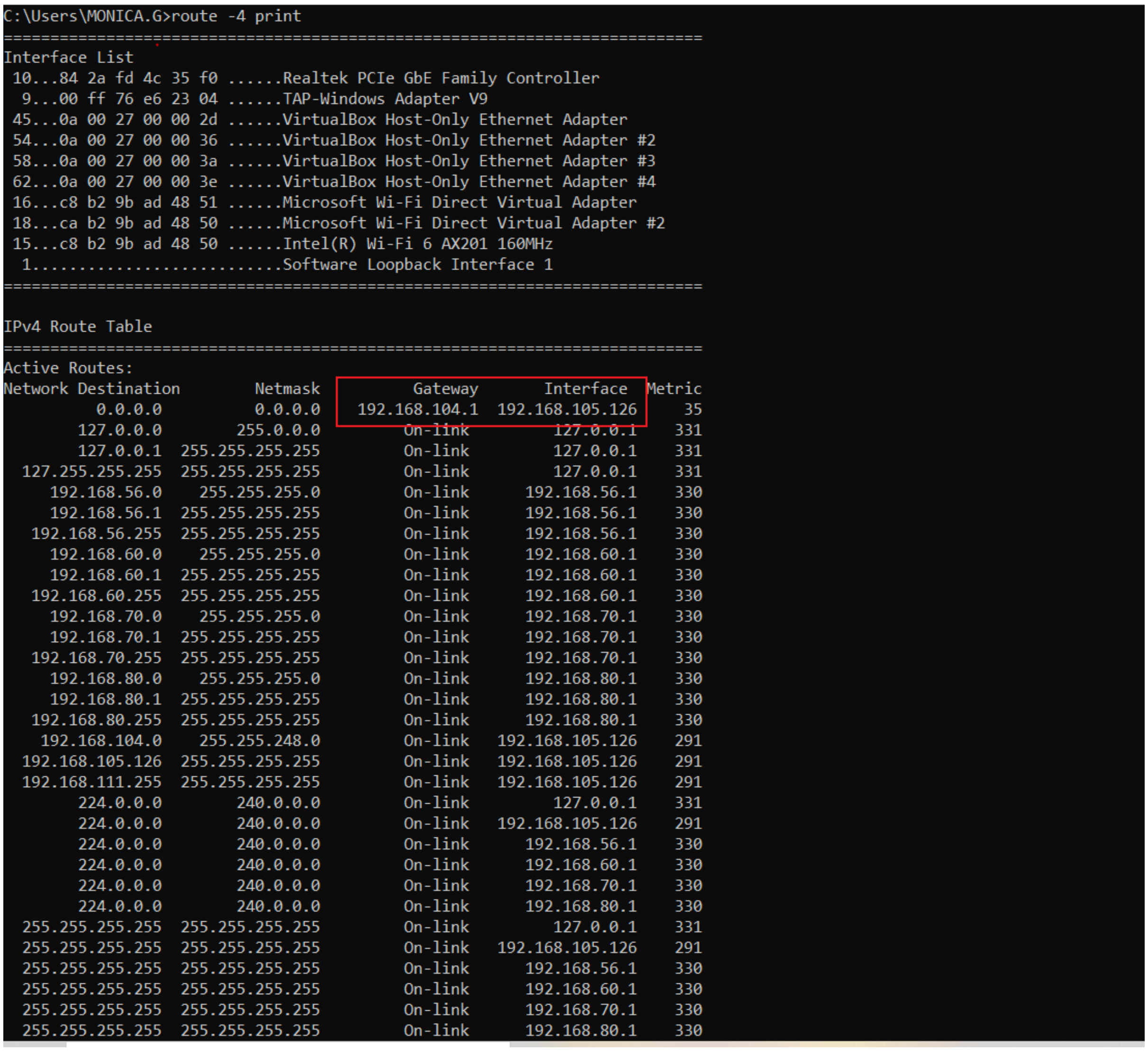


```
Command Prompt

Ethernet adapter Ethernet 3:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : VirtualBox Host-Only Ethernet Adapter #2
   Physical Address. . . . . . . . . : 0A-00-27-00-00-36
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::7d2b:fda8:4393:bf1b%54(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.60.1(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :
   DHCPv6 IAID . . . . . . . . . . . : 906625063
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-26-A1-B3-7F-84-2A-FD-4C-35-F0
   DNS Servers . . . . . . . . . . . : fec0:0:0:ffff::1%1
                                       fec0:0:0:ffff::2%1
                                       fec0:0:0:ffff::3%1
   NetBIOS over Tcpip. . . . . . . . : Enabled

Ethernet adapter Ethernet 4:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : VirtualBox Host-Only Ethernet Adapter #3
   Physical Address. . . . . . . . . : 0A-00-27-00-00-3A
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::a91e:9574:660:192b%58(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.70.1(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :
   DHCPv6 IAID . . . . . . . . . . . : 973733927
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-26-A1-B3-7F-84-2A-FD-4C-35-F0
   DNS Servers . . . . . . . . . . . : fec0:0:0:ffff::1%1
                                       fec0:0:0:ffff::2%1
                                       fec0:0:0:ffff::3%1
   NetBIOS over Tcpip. . . . . . . . : Enabled

Ethernet adapter Ethernet 5:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : VirtualBox Host-Only Ethernet Adapter #4
   Physical Address. . . . . . . . . : 0A-00-27-00-00-3E
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::7c1d:c44c:b171:abc1%62(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.80.1(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :
   DHCPv6 IAID . . . . . . . . . . . : 1040842791
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-26-A1-B3-7F-84-2A-FD-4C-35-F0
```

Figure-8: output of the command “ipconfig/all”.

Task-5: Routing tables in the host OS

As my OS is windows so I used **“route -4 print”** command. In the below figure we can see that the gateway is 192.168.104.1 and the interface is 192.168.105.126. so, from this interface we can reach the default gateway.

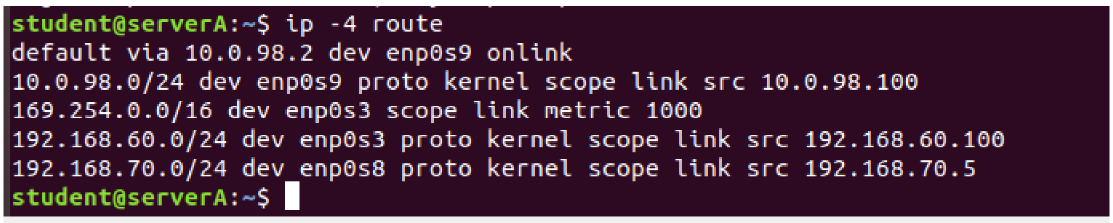


```
C:\Users\MONICA.G>route -4 print
===========================================================================
Interface List
 10...84 2a fd 4c 35 f0 ......Realtek PCIe GbE Family Controller
  9...00 ff 76 e6 23 04 ......TAP-Windows Adapter V9
 45...0a 00 27 00 00 2d ......VirtualBox Host-Only Ethernet Adapter
 54...0a 00 27 00 00 36 ......VirtualBox Host-Only Ethernet Adapter #2
 58...0a 00 27 00 00 3a ......VirtualBox Host-Only Ethernet Adapter #3
 62...0a 00 27 00 00 3e ......VirtualBox Host-Only Ethernet Adapter #4
 16...c8 b2 9b ad 48 51 ......Microsoft Wi-Fi Direct Virtual Adapter
 18...ca b2 9b ad 48 50 ......Microsoft Wi-Fi Direct Virtual Adapter #2
 15...c8 b2 9b ad 48 50 ......Intel(R) Wi-Fi 6 AX201 160MHz
  1...........................Software Loopback Interface 1
===========================================================================

IPv4 Route Table
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0    192.168.104.1  192.168.105.126     35
        127.0.0.0        255.0.0.0         On-link         127.0.0.1    331
        127.0.0.1  255.255.255.255         On-link         127.0.0.1    331
  127.255.255.255  255.255.255.255         On-link         127.0.0.1    331
     192.168.56.0    255.255.255.0         On-link      192.168.56.1    330
     192.168.56.1  255.255.255.255         On-link      192.168.56.1    330
   192.168.56.255  255.255.255.255         On-link      192.168.56.1    330
     192.168.60.0    255.255.255.0         On-link      192.168.60.1    330
     192.168.60.1  255.255.255.255         On-link      192.168.60.1    330
   192.168.60.255  255.255.255.255         On-link      192.168.60.1    330
     192.168.70.0    255.255.255.0         On-link      192.168.70.1    330
     192.168.70.1  255.255.255.255         On-link      192.168.70.1    330
   192.168.70.255  255.255.255.255         On-link      192.168.70.1    330
     192.168.80.0    255.255.255.0         On-link      192.168.80.1    330
     192.168.80.1  255.255.255.255         On-link      192.168.80.1    330
   192.168.80.255  255.255.255.255         On-link      192.168.80.1    330
    192.168.104.0    255.255.248.0         On-link   192.168.105.126    291
  192.168.105.126  255.255.255.255         On-link   192.168.105.126    291
  192.168.111.255  255.255.255.255         On-link   192.168.105.126    291
        224.0.0.0        240.0.0.0         On-link         127.0.0.1    331
        224.0.0.0        240.0.0.0         On-link   192.168.105.126    291
        224.0.0.0        240.0.0.0         On-link      192.168.56.1    330
        224.0.0.0        240.0.0.0         On-link      192.168.60.1    330
        224.0.0.0        240.0.0.0         On-link      192.168.70.1    330
        224.0.0.0        240.0.0.0         On-link      192.168.80.1    330
  255.255.255.255  255.255.255.255         On-link         127.0.0.1    331
  255.255.255.255  255.255.255.255         On-link   192.168.105.126    291
  255.255.255.255  255.255.255.255         On-link      192.168.56.1    330
  255.255.255.255  255.255.255.255         On-link      192.168.60.1    330
  255.255.255.255  255.255.255.255         On-link      192.168.70.1    330
  255.255.255.255  255.255.255.255         On-link      192.168.80.1    330
```

Figure-9: output of the command “route -4 print”.

Task-6: Routing tables in the guest OS

The command **“ip -4 route”** is used in the guest OS, and the enp0s9 (NAT) is the default gateway of the host.

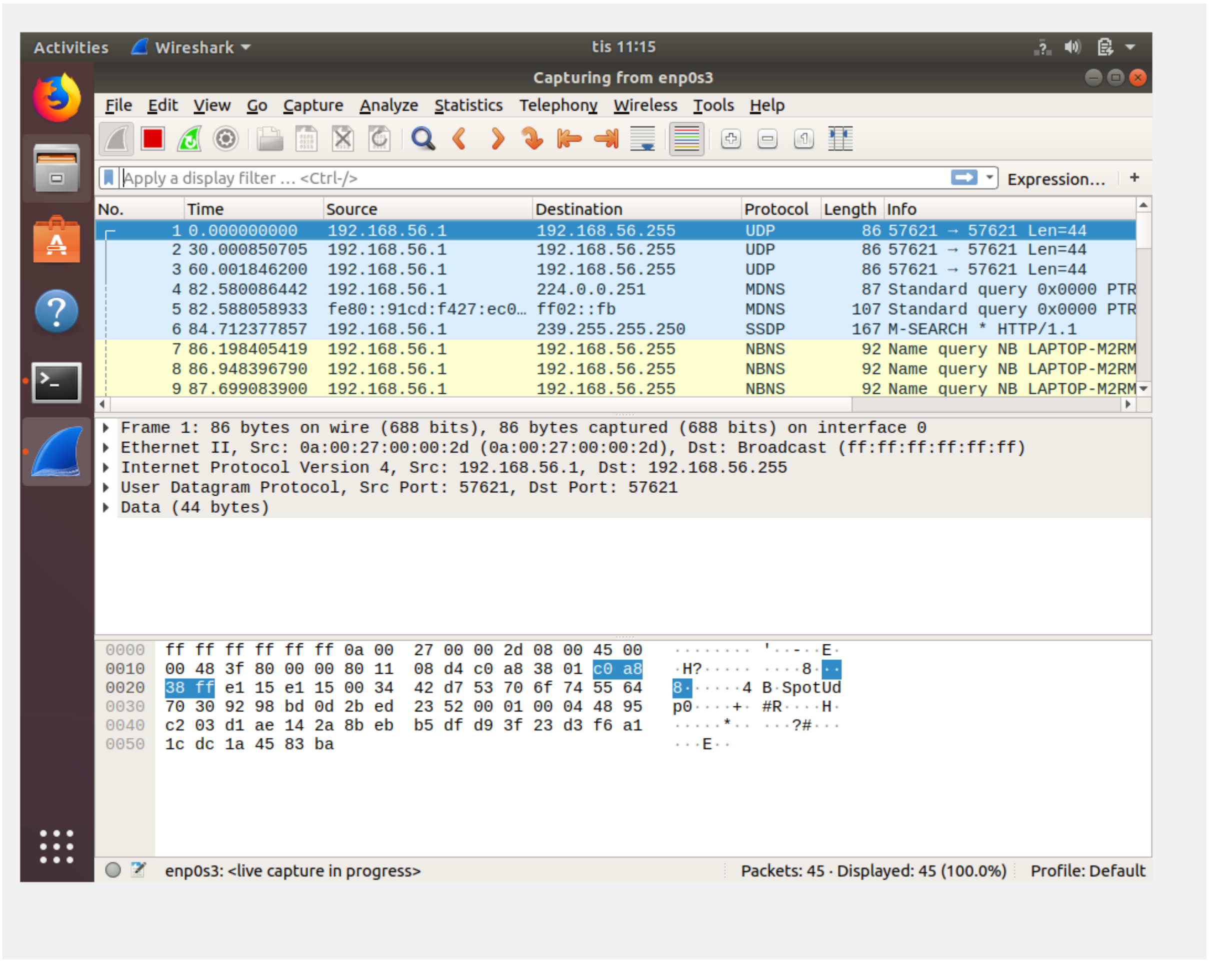


```
student@serverA:~$ ip -4 route
default via 10.0.98.2 dev enp0s9 onlink
10.0.98.0/24 dev enp0s9 proto kernel scope link src 10.0.98.100
169.254.0.0/16 dev enp0s3 scope link metric 1000
192.168.60.0/24 dev enp0s3 proto kernel scope link src 192.168.60.100
192.168.70.0/24 dev enp0s8 proto kernel scope link src 192.168.70.5
student@serverA:~$
```

Figure-10: output of the command “ip -4 route”

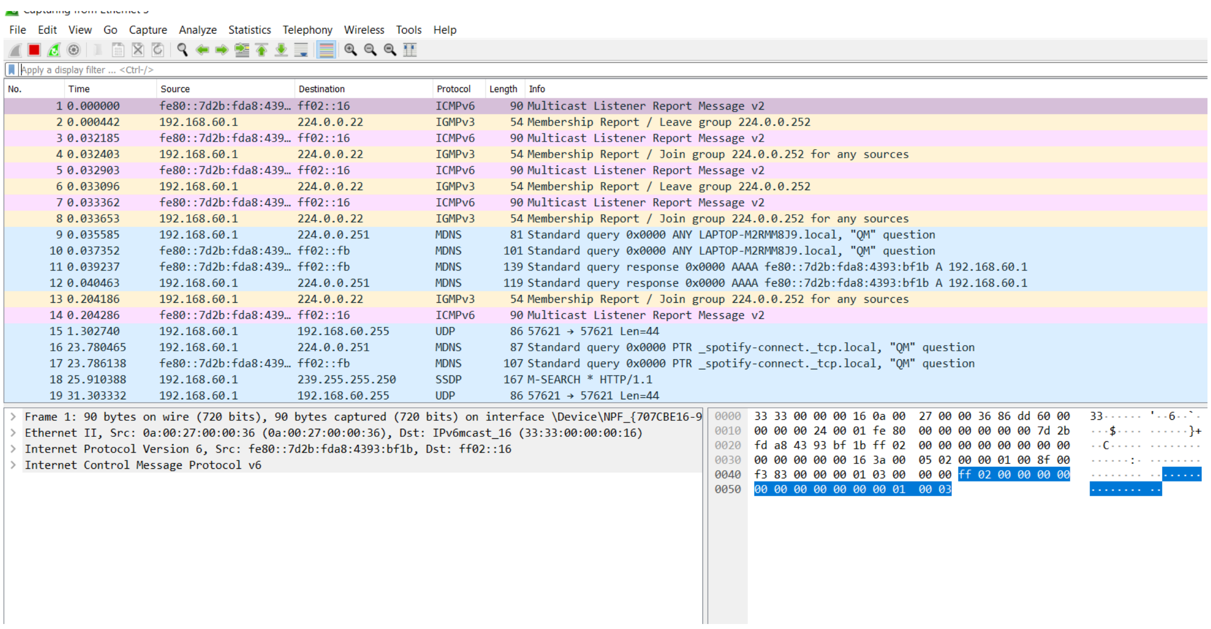
Task-7: Ping the host-based host-only interface

The figure 11 and figure 12 proves that the ICMP traffic of the two wire shark instances identical displays the information in the wire shark after pinging and the figure-12 gives the ICMP traffic of the main OS.


Activities
Wireshark
tis 11:15
Capturing from enp0s3
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Expression…

```
No.   Time            Source                 Destination        Protocol  Length Info
    1 0.000000000     192.168.56.1           192.168.56.255     UDP        86 57621 → 57621 Len=44
    2 30.000850705    192.168.56.1           192.168.56.255     UDP        86 57621 → 57621 Len=44
    3 60.001846200    192.168.56.1           192.168.56.255     UDP        86 57621 → 57621 Len=44
    4 82.580086442    192.168.56.1           224.0.0.251        MDNS       87 Standard query 0x0000 PTR
    5 82.588058933    fe80::91cd:f427:ec0…   ff02::fb           MDNS      107 Standard query 0x0000 PTR
    6 84.712377857    192.168.56.1           239.255.255.250    SSDP      167 M-SEARCH * HTTP/1.1
    7 86.198405419    192.168.56.1           192.168.56.255     NBNS       92 Name query NB LAPTOP-M2RM
    8 86.948396790    192.168.56.1           192.168.56.255     NBNS       92 Name query NB LAPTOP-M2RM
    9 87.699083900    192.168.56.1           192.168.56.255     NBNS       92 Name query NB LAPTOP-M2RM

▸ Frame 1: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
▸ Ethernet II, Src: 0a:00:27:00:00:2d (0a:00:27:00:00:2d), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▸ Internet Protocol Version 4, Src: 192.168.56.1, Dst: 192.168.56.255
▸ User Datagram Protocol, Src Port: 57621, Dst Port: 57621
▸ Data (44 bytes)

0000  ff ff ff ff ff ff 0a 00  27 00 00 2d 08 00 45 00   ········ '··-··E·
0010  00 48 3f 80 00 00 80 11  08 d4 c0 a8 38 01 c0 a8   ·H?····· ····8···
0020  38 ff e1 15 e1 15 00 34  42 d7 53 70 6f 74 55 64   8······4 B·SpotUd
0030  70 30 92 98 bd 0d 2b ed  23 52 00 01 00 04 48 95   p0····+· #R····H·
0040  c2 03 d1 ae 14 2a 8b eb  b5 df d9 3f 23 d3 f6 a1   ·····*·· ···?#···
0050  1c dc 1a 45 83 ba                                  ···E··

enp0s3: <live capture in progress>     Packets: 45 · Displayed: 45 (100.0%)     Profile: Default
```

Figure-11: Wire shark in the guest OS.

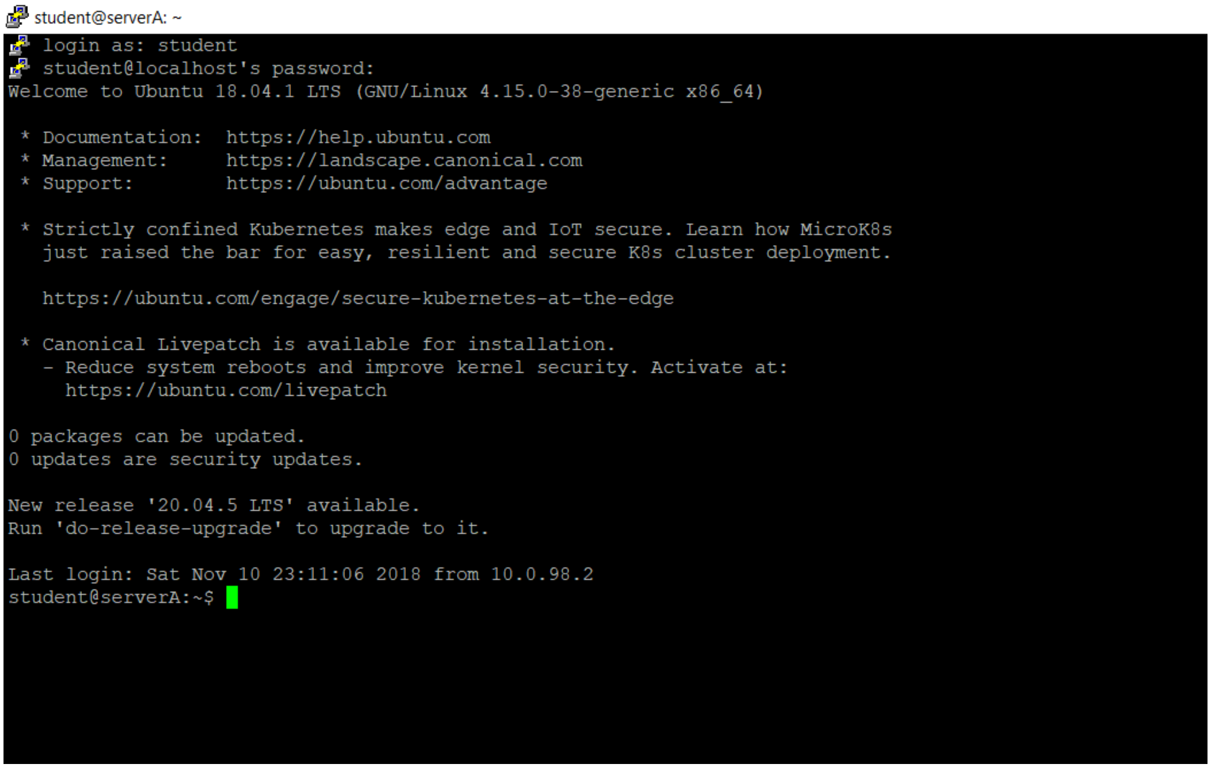

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.   Time          Source                  Destination       Protocol  Length Info
    1 0.000000      fe80::7d2b:fda8:439…    ff02::16          ICMPv6      90 Multicast Listener Report Message v2
    2 0.000442      192.168.60.1            224.0.0.22        IGMPv3      54 Membership Report / Leave group 224.0.0.252
    3 0.032185      fe80::7d2b:fda8:439…    ff02::16          ICMPv6      90 Multicast Listener Report Message v2
    4 0.032403      192.168.60.1            224.0.0.22        IGMPv3      54 Membership Report / Join group 224.0.0.252 for any sources
    5 0.032903      fe80::7d2b:fda8:439…    ff02::16          ICMPv6      90 Multicast Listener Report Message v2
    6 0.033096      192.168.60.1            224.0.0.22        IGMPv3      54 Membership Report / Leave group 224.0.0.252
    7 0.033362      fe80::7d2b:fda8:439…    ff02::16          ICMPv6      90 Multicast Listener Report Message v2
    8 0.033653      192.168.60.1            224.0.0.22        IGMPv3      54 Membership Report / Join group 224.0.0.252 for any sources
    9 0.035585      192.168.60.1            224.0.0.251       MDNS        81 Standard query 0x0000 ANY LAPTOP-M2RMM8J9.local, "QM" question
   10 0.037352      fe80::7d2b:fda8:439…    ff02::fb          MDNS       101 Standard query 0x0000 ANY LAPTOP-M2RMM8J9.local, "QM" question
   11 0.039237      fe80::7d2b:fda8:439…    ff02::fb          MDNS       139 Standard query response 0x0000 AAAA fe80::7d2b:fda8:4393:bf1b A 192.168.60.1
   12 0.040463      192.168.60.1            224.0.0.251       MDNS       119 Standard query response 0x0000 AAAA fe80::7d2b:fda8:4393:bf1b A 192.168.60.1
   13 0.204186      192.168.60.1            224.0.0.22        IGMPv3      54 Membership Report / Join group 224.0.0.252 for any sources
   14 0.204286      fe80::7d2b:fda8:439…    ff02::16          ICMPv6      90 Multicast Listener Report Message v2
   15 1.302740      192.168.60.1            192.168.60.255    UDP         86 57621 → 57621 Len=44
   16 23.780465     192.168.60.1            224.0.0.251       MDNS        87 Standard query 0x0000 PTR _spotify-connect._tcp.local, "QM" question
   17 23.786138     fe80::7d2b:fda8:439…    ff02::fb          MDNS       107 Standard query 0x0000 PTR _spotify-connect._tcp.local, "QM" question
   18 25.910388     192.168.60.1            239.255.255.250   SSDP       167 M-SEARCH * HTTP/1.1
   19 31.303332     192.168.60.1            192.168.60.255    UDP         86 57621 → 57621 Len=44

> Frame 1: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface \Device\NPF_{707CBE16-9
> Ethernet II, Src: 0a:00:27:00:00:36 (0a:00:27:00:00:36), Dst: IPv6mcast_16 (33:33:00:00:00:16)
> Internet Protocol Version 6, Src: fe80::7d2b:fda8:4393:bf1b, Dst: ff02::16
> Internet Control Message Protocol v6

0000  33 33 00 00 00 16 0a 00  27 00 00 36 86 dd 60 00   33······ '··6··`·
0010  00 00 00 24 00 01 fe 80  00 00 00 00 00 00 7d 2b   ···$···· ······}+
0020  fd a8 43 93 bf 1b ff 02  00 00 00 00 00 00 00 00   ··C····· ········
0030  00 00 00 00 00 16 3a 00  05 02 00 00 01 00 8f 00   ······:· ········
0040  f3 83 00 00 00 01 03 00  00 00 ff 02 00 00 00 00   ········ ········
0050  00 00 00 00 00 00 00 01  00 03                     ········ ··
```

Figure-12: wire shark in the main OS.

Task-8: ssh into VM via local host

As I am using Windows host in order to connect to ssh, putty is used. Start the putty and enter the localhost in the Host name and enter the port number “10022”. After the authentication is done then the shell is opened.

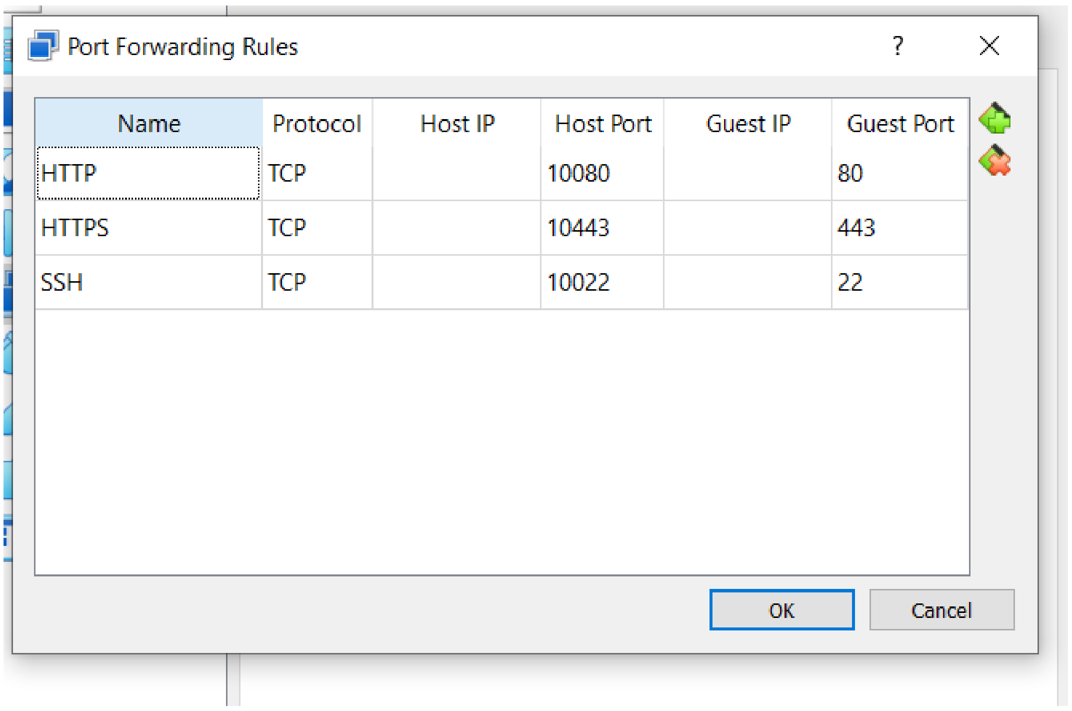


```
login as: student
student@localhost's password:
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-38-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

0 packages can be updated.
0 updates are security updates.

New release '20.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sat Nov 10 23:11:06 2018 from 10.0.98.2
student@serverA:~$
```

Figure-13: remote shell in server A

Task-9: Add forwarding rules for HTTP and HTTPS in VirtualBox

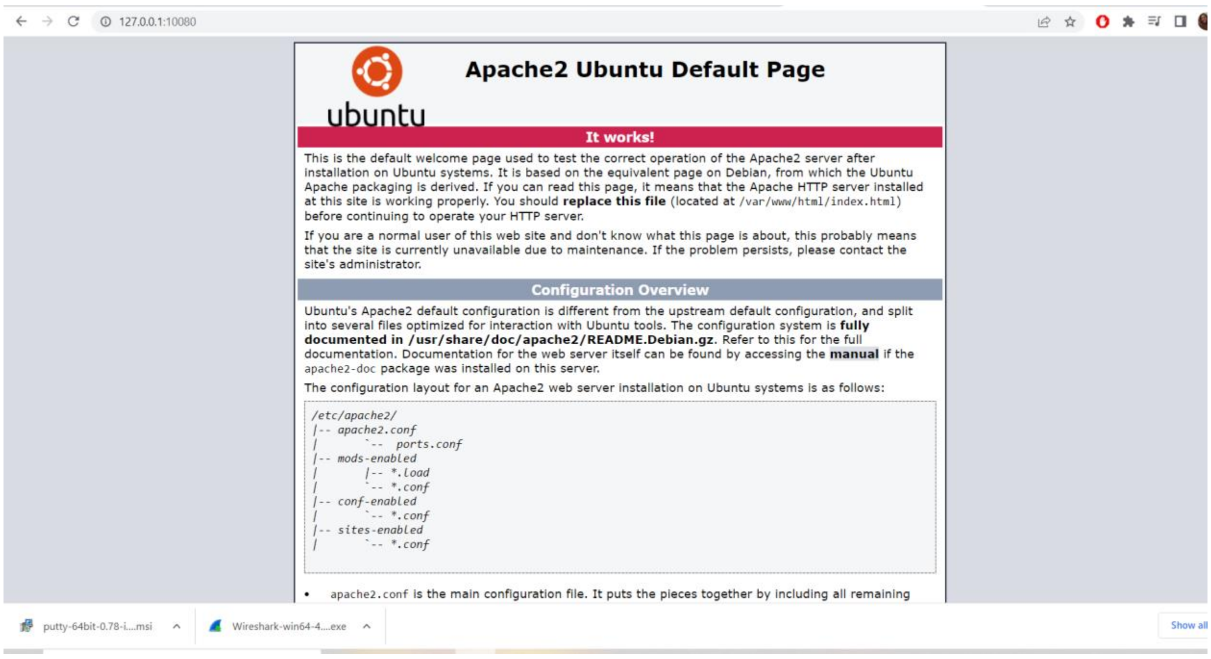
The port Forwarding rules for HTTP and HTTPS. The official port numbers of the HTTP and HTTPS are added to 10000(general port number of http is 80 and it is added to 10000 with results in 10080 and used as the host port) and they are set as the Host port.



Name	Protocol	Host IP	Host Port	Guest IP	Guest Port
HTTP	TCP		10080		80
HTTPS	TCP		10443		443
SSH	TCP		10022		22

Figure-14: Port forwarding rules.

By setting the port forwarding rules, it allows the host to browse the webserver in the VM guest OS over HTTP and HTTPS.



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

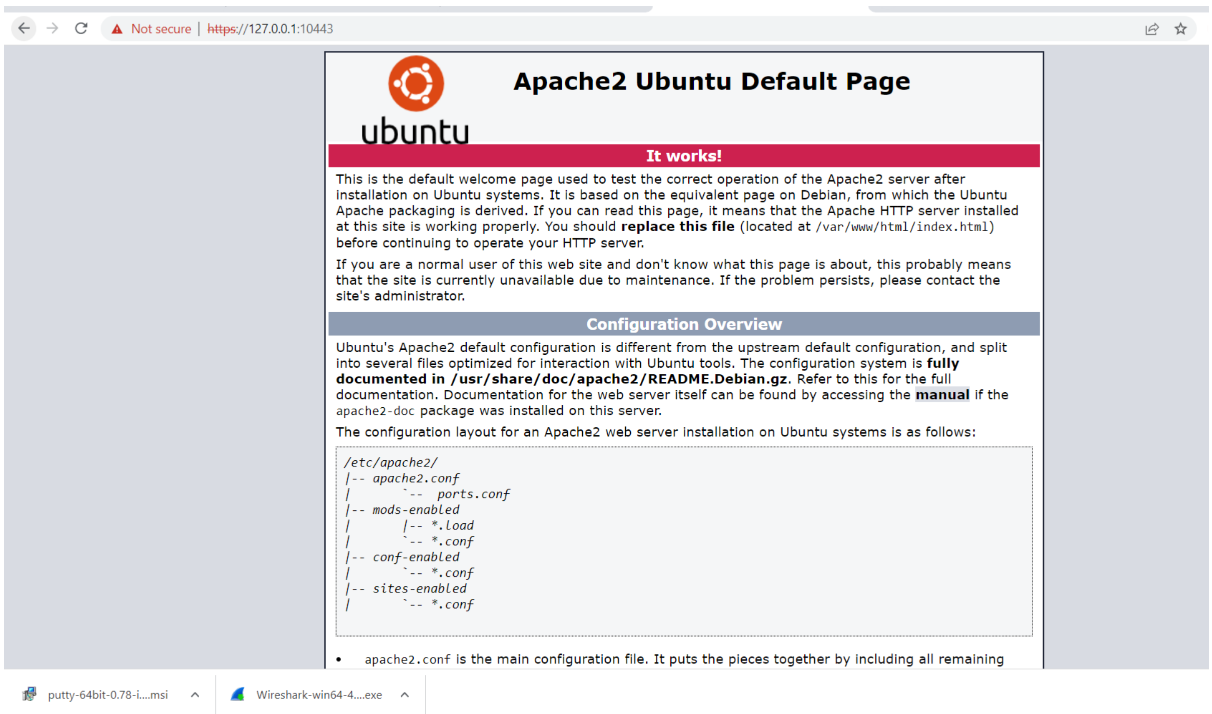
Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining

Figure-15: apache2 server over HTTP


Not secure | ~~https~~://127.0.0.1:10443
putty-64bit-0.78-i....msi
Wireshark-win64-4....exe

Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining

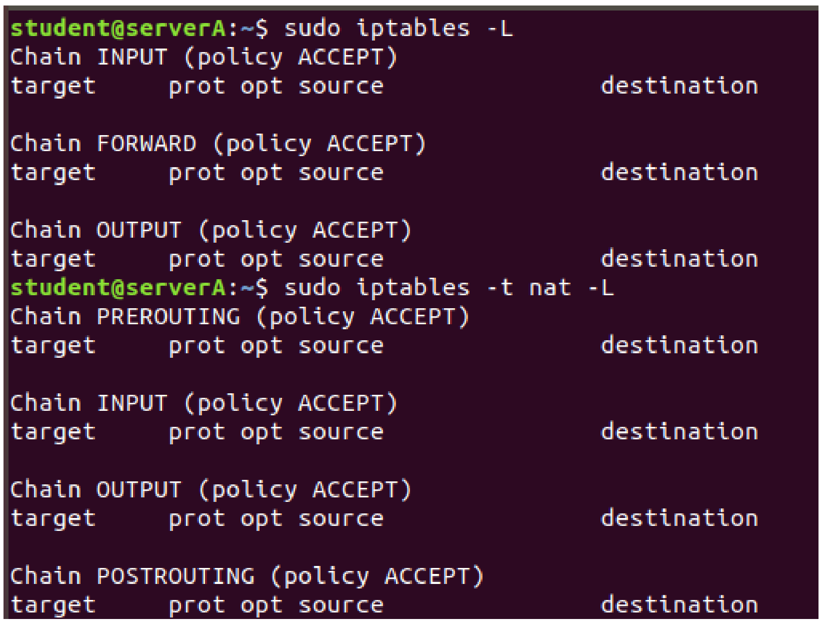
Figure-16: apache2 server over HTTPS

Task-10: Default firewall policy and rules

To find the default policy and the rules installed by default in the VM in the tables filters, mangle, and nat. The following commands are used to find the default policies.

Sudo iptables -L

Sudo iptables -t nat -L

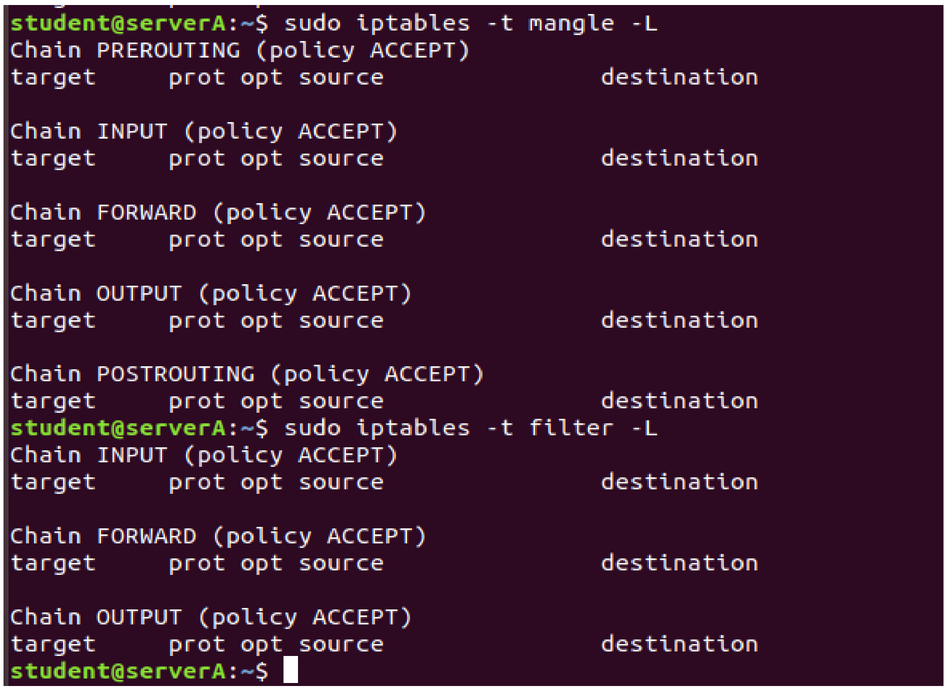


```
student@serverA:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
student@serverA:~$ sudo iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
```

Figure-17: Iptables

Sudo iptables -t mangle -L

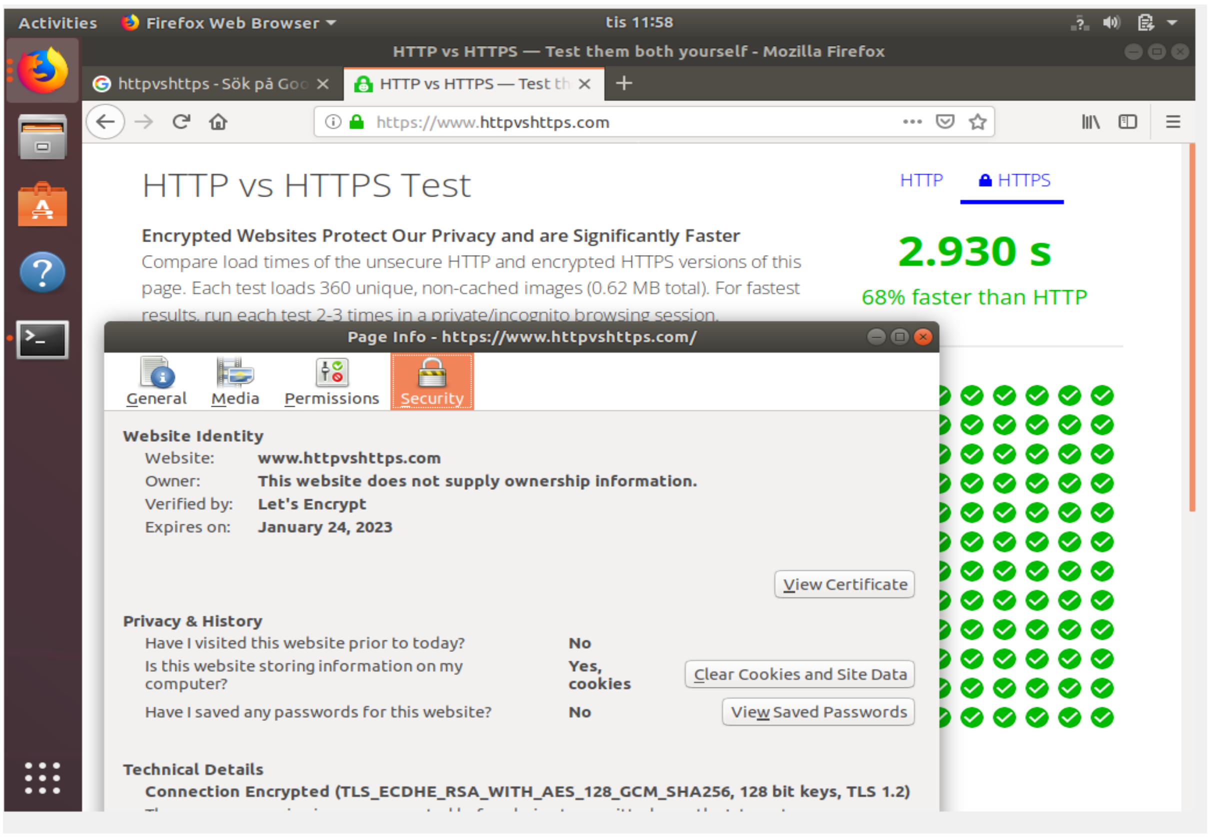
Sudo iptables -t filters -L



```
student@serverA:~$ sudo iptables -t mangle -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
student@serverA:~$ sudo iptables -t filter -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
student@serverA:~$
```

Figure-18: iptables of mangle, filters

Now we need to verify the websites over HTTP and HTTPS in server A. The links http://www.httpvshttps.com and https://www.httpvshttps.com are searched in the web browser on the guest OS.



HTTP vs HTTPS Test

Encrypted Websites Protect Our Privacy and are Significantly Faster

Compare load times of the unsecure HTTP and encrypted HTTPS versions of this page. Each test loads 360 unique, non-cached images (0.62 MB total). For fastest results, run each test 2-3 times in a private/incognito browsing session.

2.930 s

68% faster than HTTP

Website Identity

Website:	www.httpvshttps.com
Owner:	This website does not supply ownership information.
Verified by:	Let's Encrypt
Expires on:	January 24, 2023

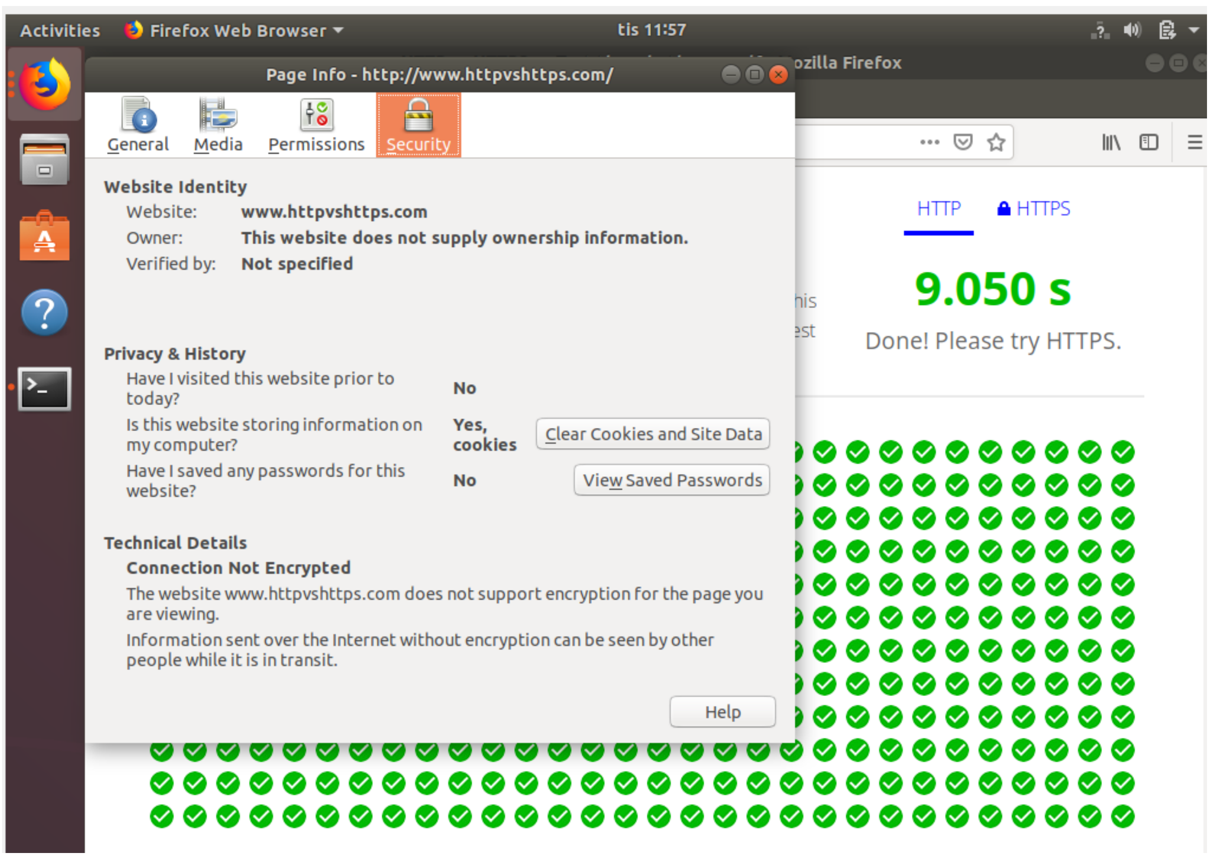
Privacy & History

Have I visited this website prior to today?	No
Is this website storing information on my computer?	Yes, cookies
Have I saved any passwords for this website?	No

Technical Details

Connection Encrypted (TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256, 128 bit keys, TLS 1.2)

Figure-19: The HTTPS link to verify.



Website Identity
Website: **www.httpvshttps.com**
Owner: **This website does not supply ownership information.**
Verified by: **Not specified**

Privacy & History
Have I visited this website prior to today? **No**
Is this website storing information on my computer? **Yes, cookies**
Have I saved any passwords for this website? **No**

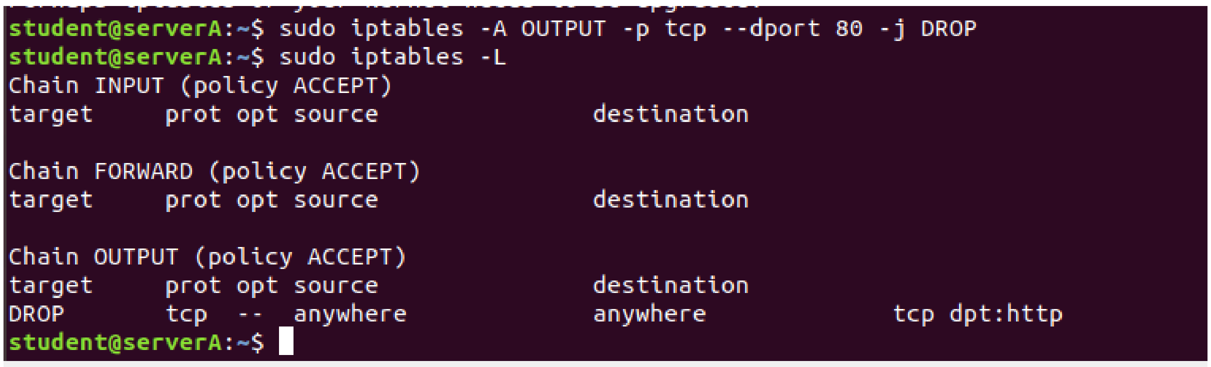
Technical Details
Connection Not Encrypted
The website www.httpvshttps.com does not support encryption for the page you are viewing.
Information sent over the Internet without encryption can be seen by other people while it is in transit.

Figure-20: The HTTP link to verify.

Task-11: Block HTTP-browsing in the guest OS

In this task we need to Block the HTTP-browsing in the Guest OS, but retain the ability to browse with HTTPS. To do that the following command need to be entered in the terminal.

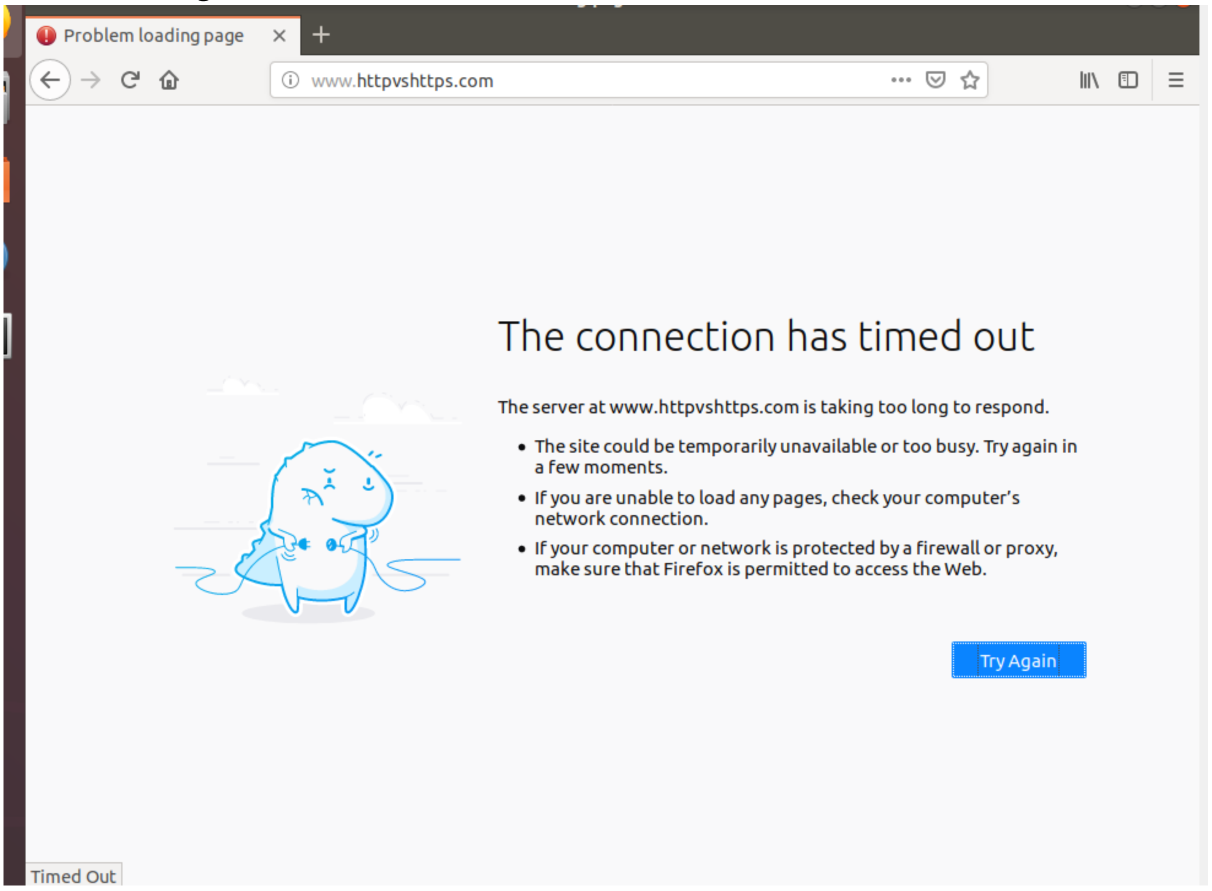
Sudo iptables -A OUTPUT -p tcp –dport 80 -j DROP



```
student@serverA:~$ sudo iptables -A OUTPUT -p tcp --dport 80 -j DROP
student@serverA:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  anywhere             anywhere             tcp dpt:http
student@serverA:~$
```

Figure-21: command used to block the HTTP -browsing.

In the below figure we can see that HTTP is blocked.



The connection has timed out

The server at www.httpvshttps.com is taking too long to respond.

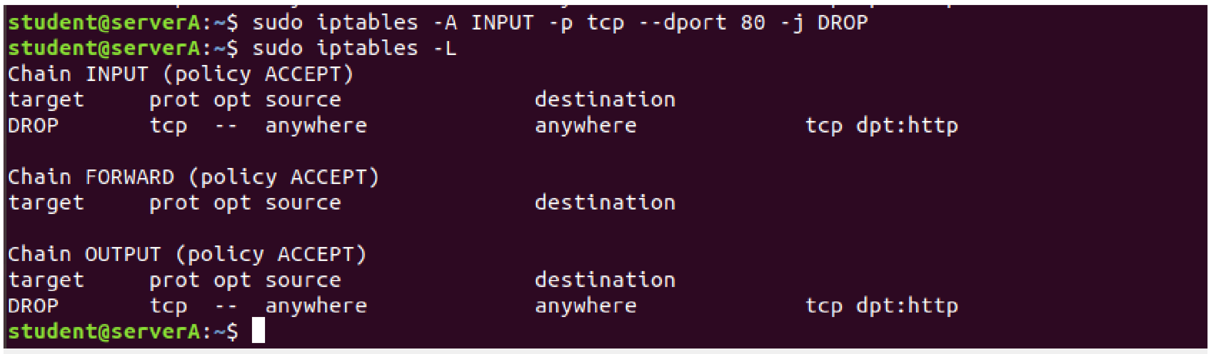
- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the Web.

Figure-22: Problem in loading page of HTTP server.

Task-12: Block Apache web server from serving content over HTTP

In this task we need to block Apache web server from serving content over HTTP, but retains the ability to see content served over HTTPS. The following command is used to blocking the HTTP serving the Apache.

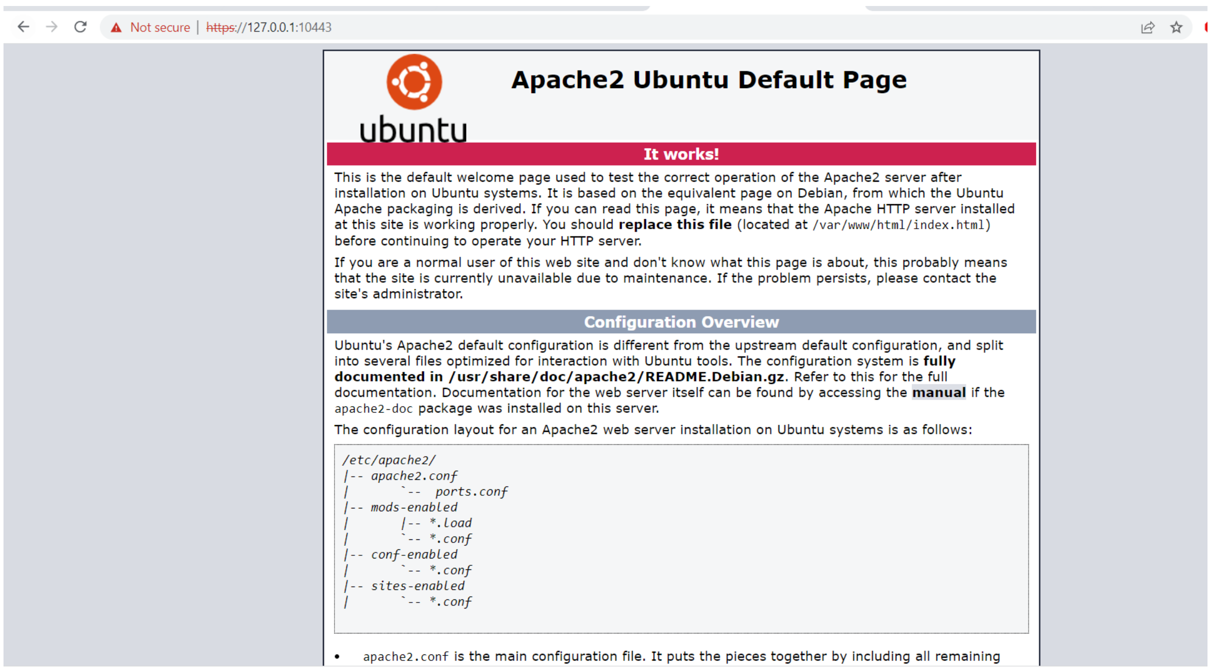
Sudo iptables -A INPUT -p tcp –dport 80 -j DROP



```
student@serverA:~$ sudo iptables -A INPUT -p tcp --dport 80 -j DROP
student@serverA:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  anywhere             anywhere             tcp dpt:http

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  anywhere             anywhere             tcp dpt:http
student@serverA:~$
```

Figure-23: Command to block the Apache webserver from serving the HTTP.

In the below figure we can see that still HTTPS retain the ability to serve even though we blocked the HTTP.


Not secure | ~~https~~://127.0.0.1:10443

Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

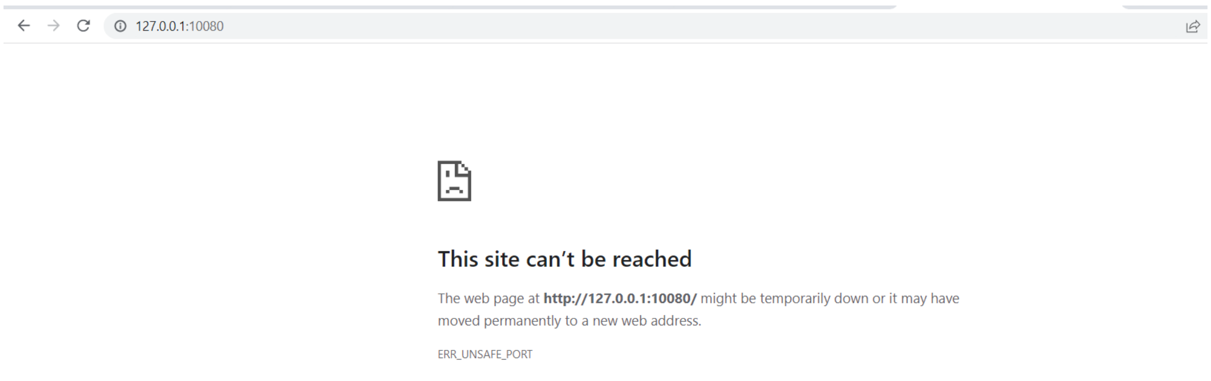
The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining

Figure24: HTTPS is still able to served.

In the below figure 25 the HTTP is unable to serve because we blocked the HTTP from serving.


127.0.0.1:10080

This site can’t be reached

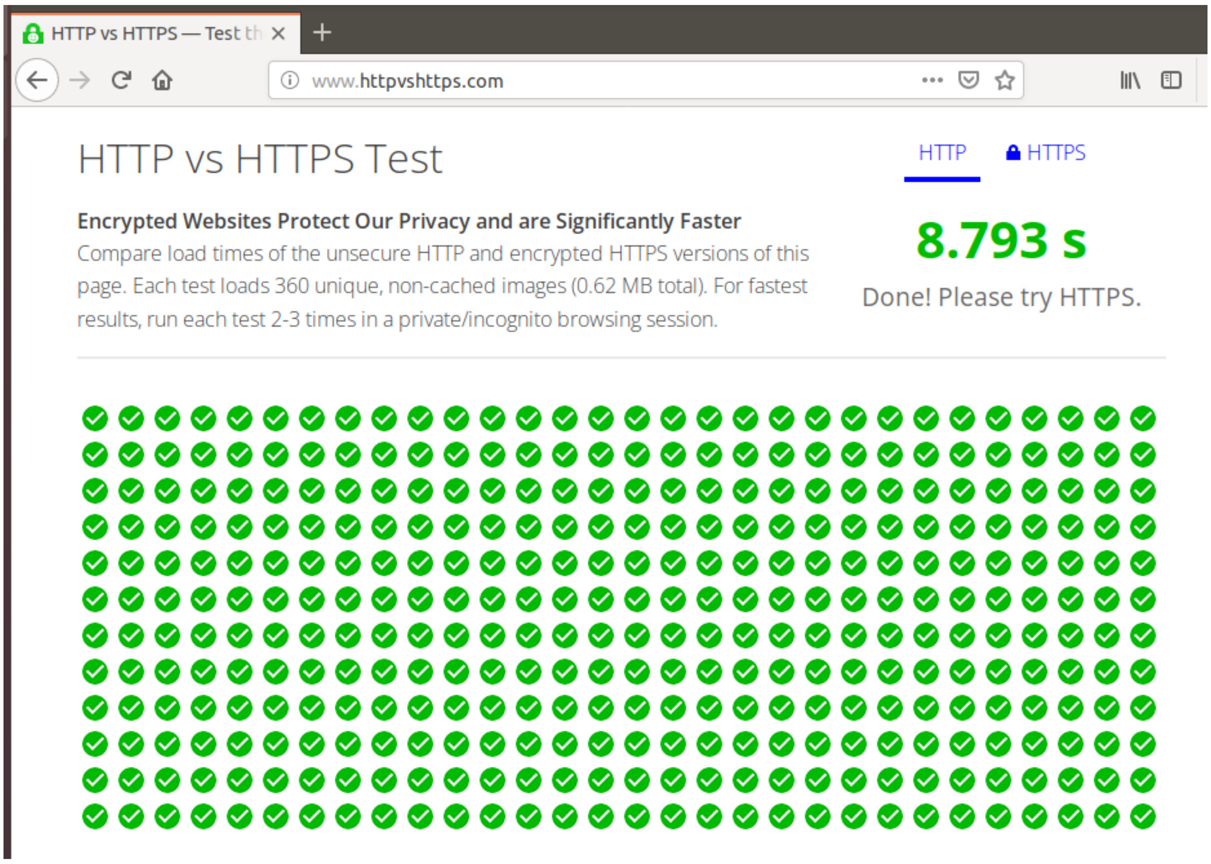
The web page at **http://127.0.0.1:10080/** might be temporarily down or it may have moved permanently to a new web address.

ERR_UNSAFE_PORT

Figure 25: blocking the HTTP serving

Task-13: Unblock HTTP-browsing in the quest OS

In task-11 we blocked the HTTP-browsing, in the task-13 we need to undo the task so that the browsing with the HTTP is done. To check HTTP-browsing the link http://www.httpvshttps.com is used. In the below figure we see the HTTP-browsing is checked.



HTTP vs HTTPS Test

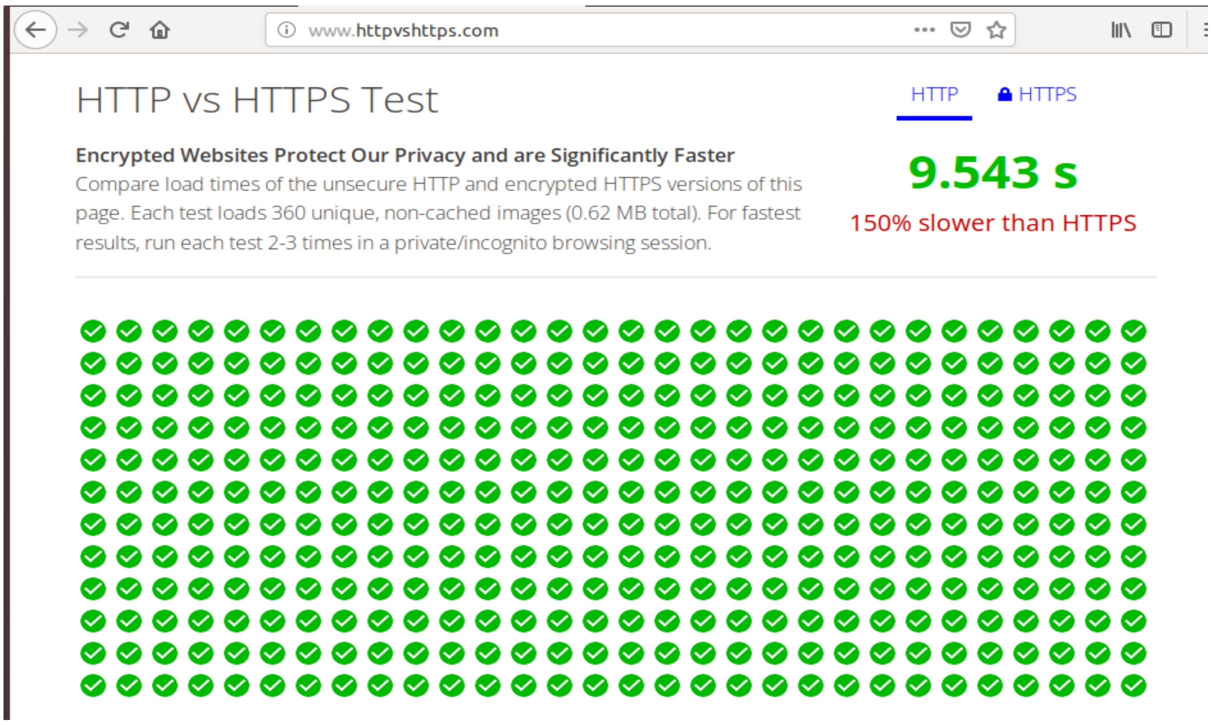
Encrypted Websites Protect Our Privacy and are Significantly Faster
Compare load times of the unsecure HTTP and encrypted HTTPS versions of this page. Each test loads 360 unique, non-cached images (0.62 MB total). For fastest results, run each test 2-3 times in a private/incognito browsing session.

8.793 s
Done! Please try HTTPS.

Figure 26: Check the HTTP-browsing.

Task-14: Use firewall.sh to configure the firewall

In this task we need to modify the script and we need to make sure that guest OS can view HTTP and HTTPS pages, but the apache2 server is blocked from the serving HTTP content.


www.httpvshttps.com

HTTP vs HTTPS Test

HTTP HTTPS

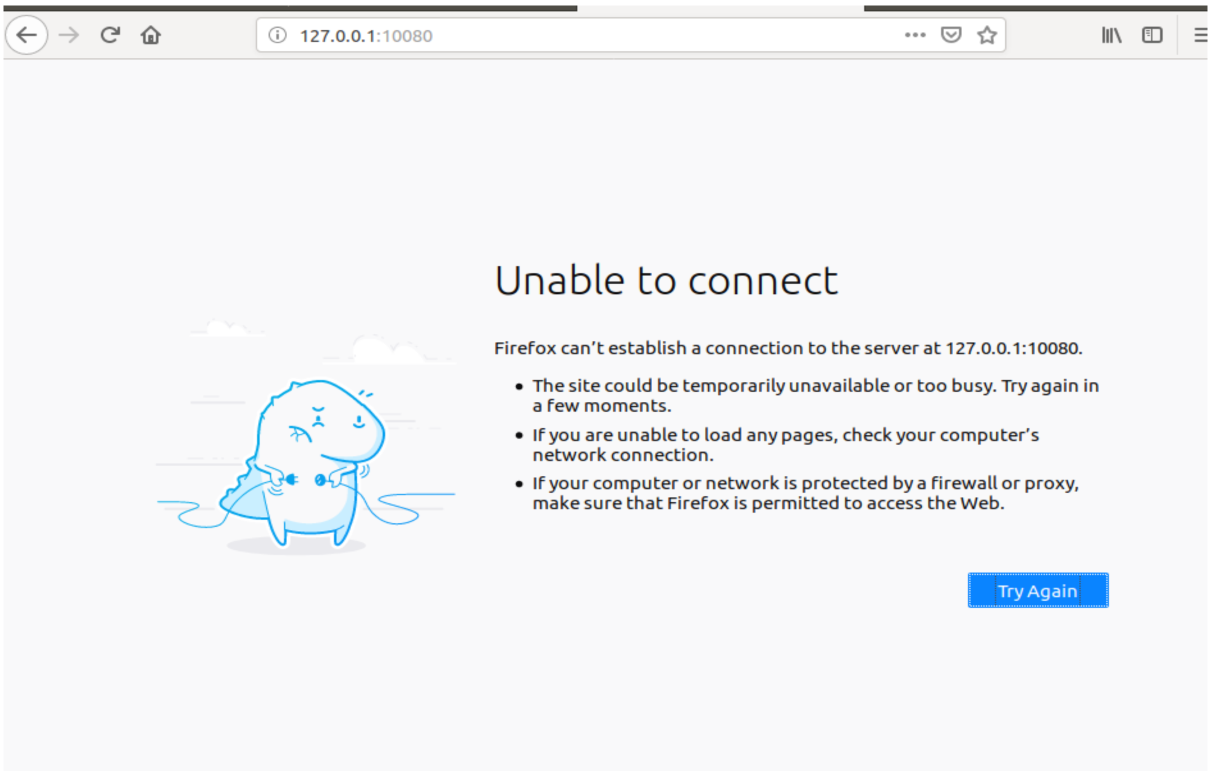
Encrypted Websites Protect Our Privacy and are Significantly Faster
Compare load times of the unsecure HTTP and encrypted HTTPS versions of this page. Each test loads 360 unique, non-cached images (0.62 MB total). For fastest results, run each test 2-3 times in a private/incognito browsing session.

9.543 s

150% slower than HTTPS

Figure 27: http and https pages can be viewed.

In the below figure28 we can see that the serving HTTP content is blocked


127.0.0.1:10080

Unable to connect

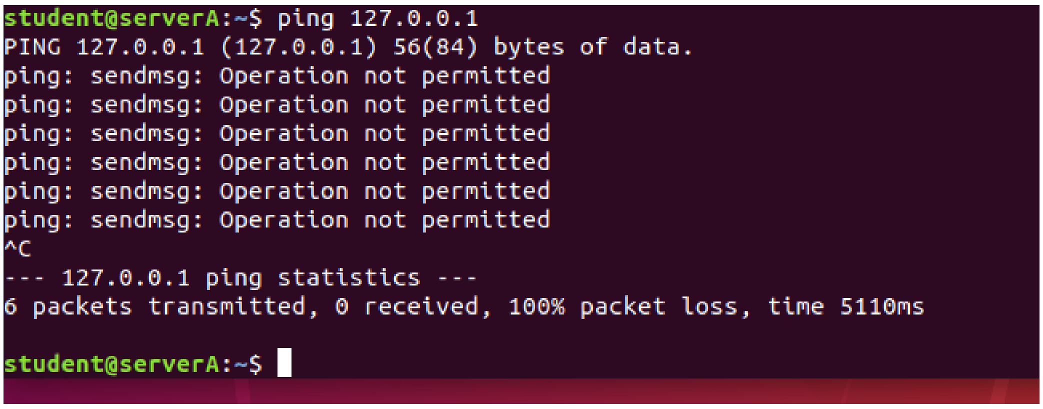
Firefox can't establish a connection to the server at 127.0.0.1:10080.

- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the Web.

Figure 28: Apache 2 is blocked in HTTP.

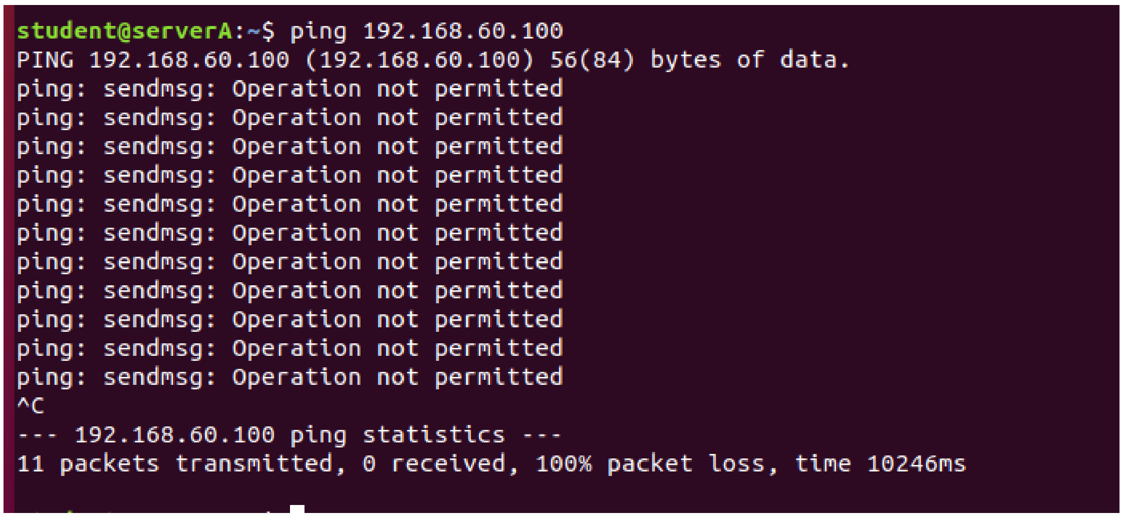
Task-15: change the default firewall policy to drop

In this task we need to change the default firewall policy to Drop. Then the ping is checked, the server A is not connected to the outside world and the ping the loopback interface.



```
student@serverA:~$ ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
^C
--- 127.0.0.1 ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 5110ms

student@serverA:~$
```

Figure29: ping loopback interface in guest OS



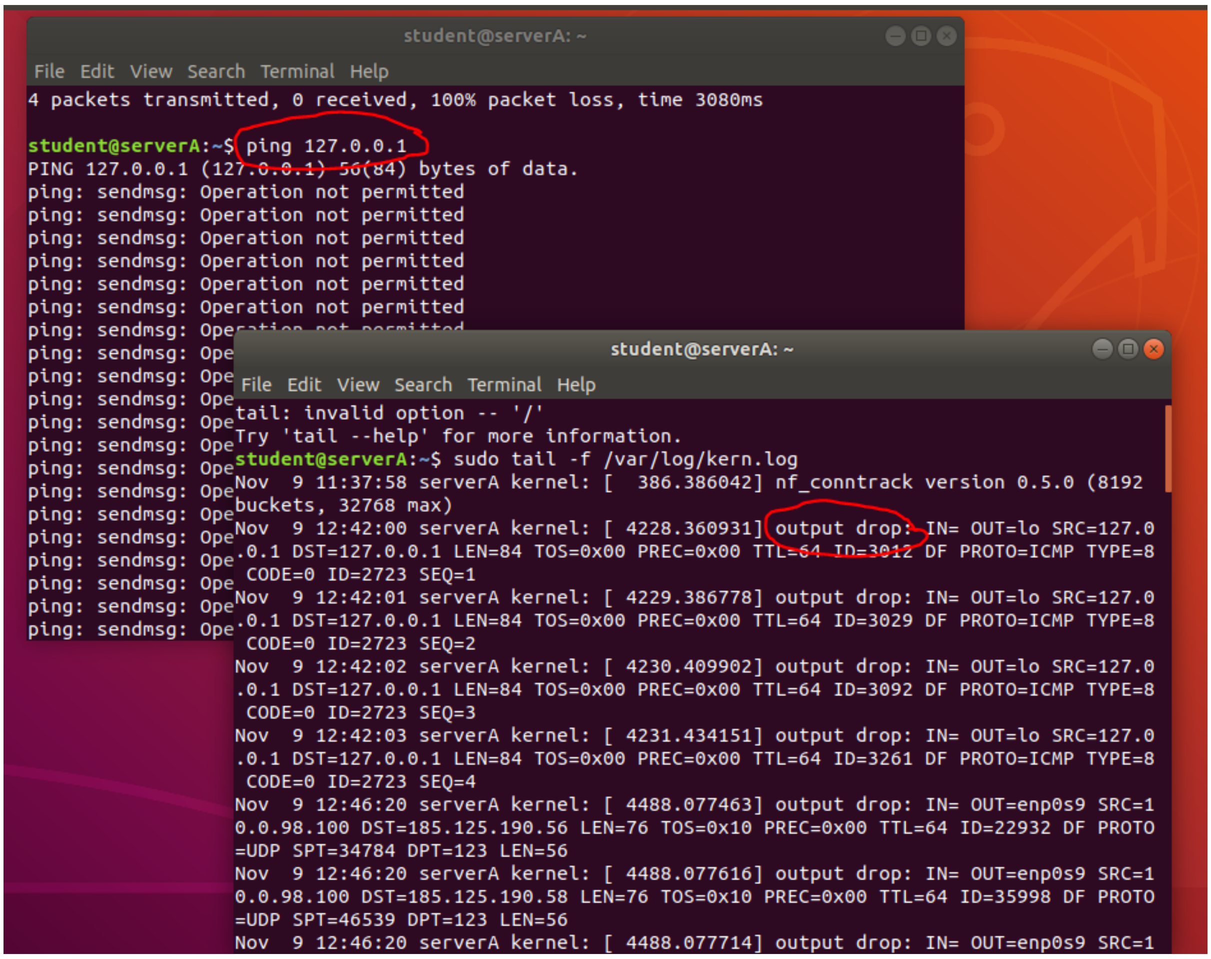
```
student@serverA:~$ ping 192.168.60.100
PING 192.168.60.100 (192.168.60.100) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
^C
--- 192.168.60.100 ping statistics ---
11 packets transmitted, 0 received, 100% packet loss, time 10246ms
```

Figure30: ping 192.168.60.100 in guest OS

Task-16: Logging Dropped packets

In this task firewall.sh is executed, one new terminal is opened and the command **"sudo tail -f /var/log/kern.log"**.

Now we can see the logs in the Linux kernels. Loopback interface is pinged into the terminal.



```
4 packets transmitted, 0 received, 100% packet loss, time 3080ms

student@serverA:~$ ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Ope
ping: sendmsg: Ope
ping: sendmsg: Ope
ping: sendmsg: Ope
ping: sendmsg: Ope
ping: sendmsg: Ope
ping: sendmsg: Ope
ping: sendmsg: Ope
ping: sendmsg: Ope
ping: sendmsg: Ope
ping: sendmsg: Ope
ping: sendmsg: Ope
ping: sendmsg: Ope
```

```
tail: invalid option -- '/'
Try 'tail --help' for more information.
student@serverA:~$ sudo tail -f /var/log/kern.log
Nov  9 11:37:58 serverA kernel: [  386.386042] nf_conntrack version 0.5.0 (8192
buckets, 32768 max)
Nov  9 12:42:00 serverA kernel: [ 4228.360931] output drop: IN= OUT=lo SRC=127.0
.0.1 DST=127.0.0.1 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=3012 DF PROTO=ICMP TYPE=8
 CODE=0 ID=2723 SEQ=1
Nov  9 12:42:01 serverA kernel: [ 4229.386778] output drop: IN= OUT=lo SRC=127.0
.0.1 DST=127.0.0.1 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=3029 DF PROTO=ICMP TYPE=8
 CODE=0 ID=2723 SEQ=2
Nov  9 12:42:02 serverA kernel: [ 4230.409902] output drop: IN= OUT=lo SRC=127.0
.0.1 DST=127.0.0.1 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=3092 DF PROTO=ICMP TYPE=8
 CODE=0 ID=2723 SEQ=3
Nov  9 12:42:03 serverA kernel: [ 4231.434151] output drop: IN= OUT=lo SRC=127.0
.0.1 DST=127.0.0.1 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=3261 DF PROTO=ICMP TYPE=8
 CODE=0 ID=2723 SEQ=4
Nov  9 12:46:20 serverA kernel: [ 4488.077463] output drop: IN= OUT=enp0s9 SRC=1
0.0.98.100 DST=185.125.190.56 LEN=76 TOS=0x10 PREC=0x00 TTL=64 ID=22932 DF PROTO
=UDP SPT=34784 DPT=123 LEN=56
Nov  9 12:46:20 serverA kernel: [ 4488.077616] output drop: IN= OUT=enp0s9 SRC=1
0.0.98.100 DST=185.125.190.58 LEN=76 TOS=0x10 PREC=0x00 TTL=64 ID=35998 DF PROTO
=UDP SPT=46539 DPT=123 LEN=56
Nov  9 12:46:20 serverA kernel: [ 4488.077714] output drop: IN= OUT=enp0s9 SRC=1
```

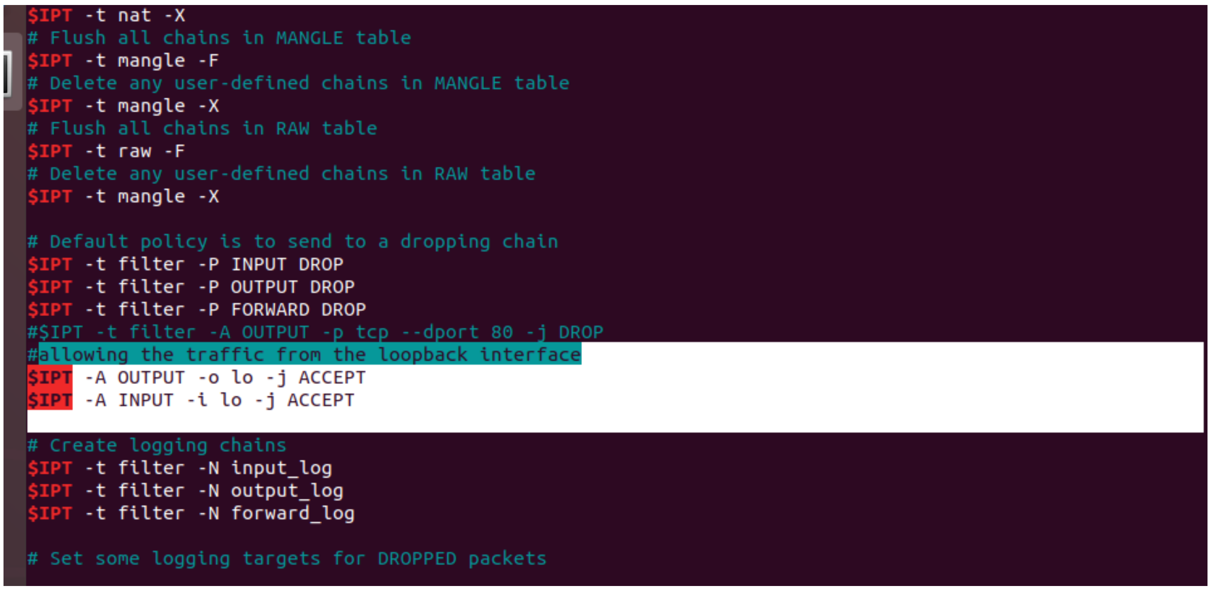
Figure 31: Kernel Logs.

Task-17: enable traffic from loop back interface

In this task we need to enable the traffic from the loopback interface and we need ping the localhost. To do that the following rules needs to be added to the firwall.sh file.

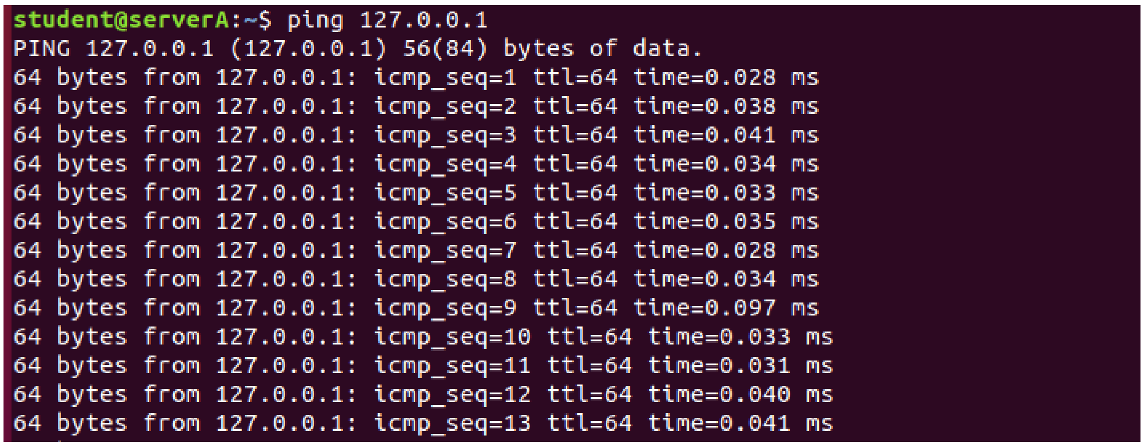
$IPT -A OUTPUT -o lo -j ACCEPT

$IPT -A INPUT -i lo -j ACCEPT



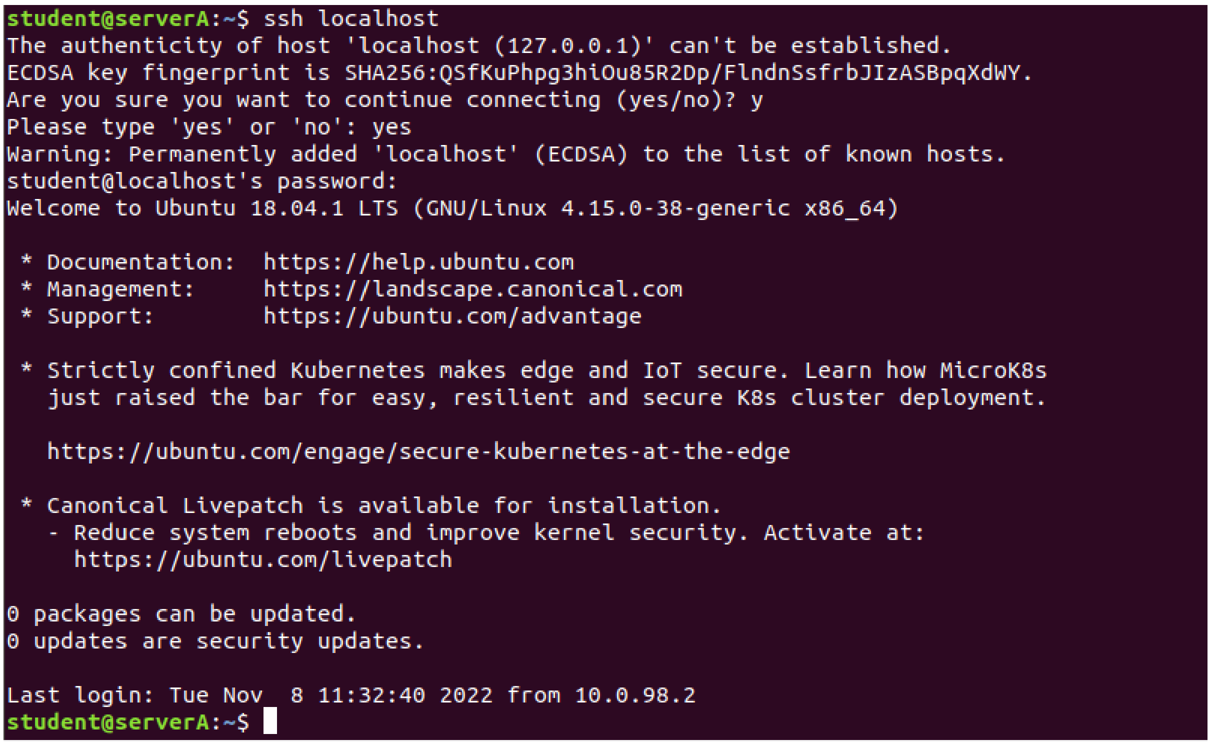
```
$IPT -t nat -X
# Flush all chains in MANGLE table
$IPT -t mangle -F
# Delete any user-defined chains in MANGLE table
$IPT -t mangle -X
# Flush all chains in RAW table
$IPT -t raw -F
# Delete any user-defined chains in RAW table
$IPT -t mangle -X

# Default policy is to send to a dropping chain
$IPT -t filter -P INPUT DROP
$IPT -t filter -P OUTPUT DROP
$IPT -t filter -P FORWARD DROP
#$IPT -t filter -A OUTPUT -p tcp --dport 80 -j DROP
#allowing the traffic from the loopback interface
$IPT -A OUTPUT -o lo -j ACCEPT
$IPT -A INPUT -i lo -j ACCEPT

# Create logging chains
$IPT -t filter -N input_log
$IPT -t filter -N output_log
$IPT -t filter -N forward_log

# Set some logging targets for DROPPED packets
```

Figure 32: code to enable the traffic.



```
student@serverA:~$ ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.028 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.038 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.041 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.034 ms
64 bytes from 127.0.0.1: icmp_seq=5 ttl=64 time=0.033 ms
64 bytes from 127.0.0.1: icmp_seq=6 ttl=64 time=0.035 ms
64 bytes from 127.0.0.1: icmp_seq=7 ttl=64 time=0.028 ms
64 bytes from 127.0.0.1: icmp_seq=8 ttl=64 time=0.034 ms
64 bytes from 127.0.0.1: icmp_seq=9 ttl=64 time=0.097 ms
64 bytes from 127.0.0.1: icmp_seq=10 ttl=64 time=0.033 ms
64 bytes from 127.0.0.1: icmp_seq=11 ttl=64 time=0.031 ms
64 bytes from 127.0.0.1: icmp_seq=12 ttl=64 time=0.040 ms
64 bytes from 127.0.0.1: icmp_seq=13 ttl=64 time=0.041 ms
```

Figure33: ping the loopback interface.



```
student@serverA:~$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:QSfKuPhpg3hiOu85R2Dp/FlndnSsfrbJIzASBpqXdWY.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
student@localhost's password:
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-38-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

0 packages can be updated.
0 updates are security updates.

Last login: Tue Nov  8 11:32:40 2022 from 10.0.98.2
student@serverA:~$
```

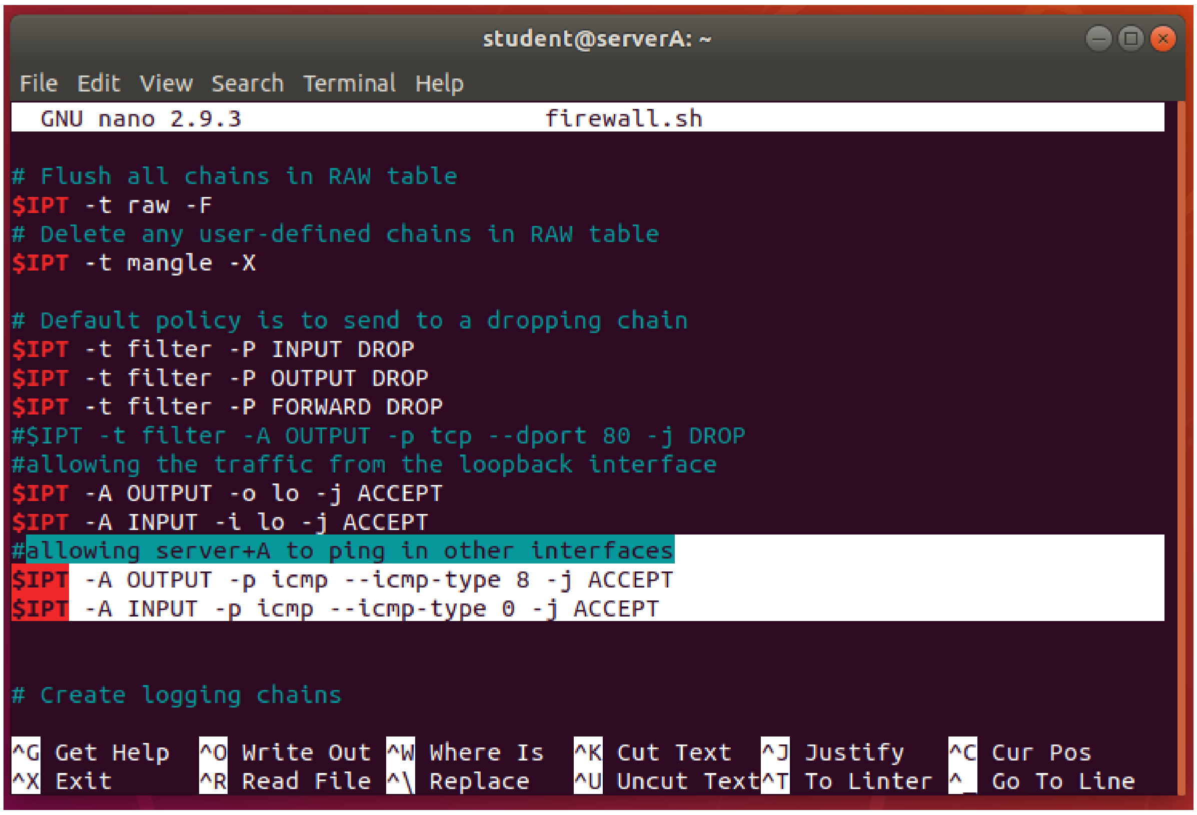
Figure 34: connecting the SSH into localhost.

Task-18: Allow the server A to ping to all interfaces

In this task we need to allow the server A to ping the other interfaces. To do that the following rules need to be added to eth firewall.sh file.

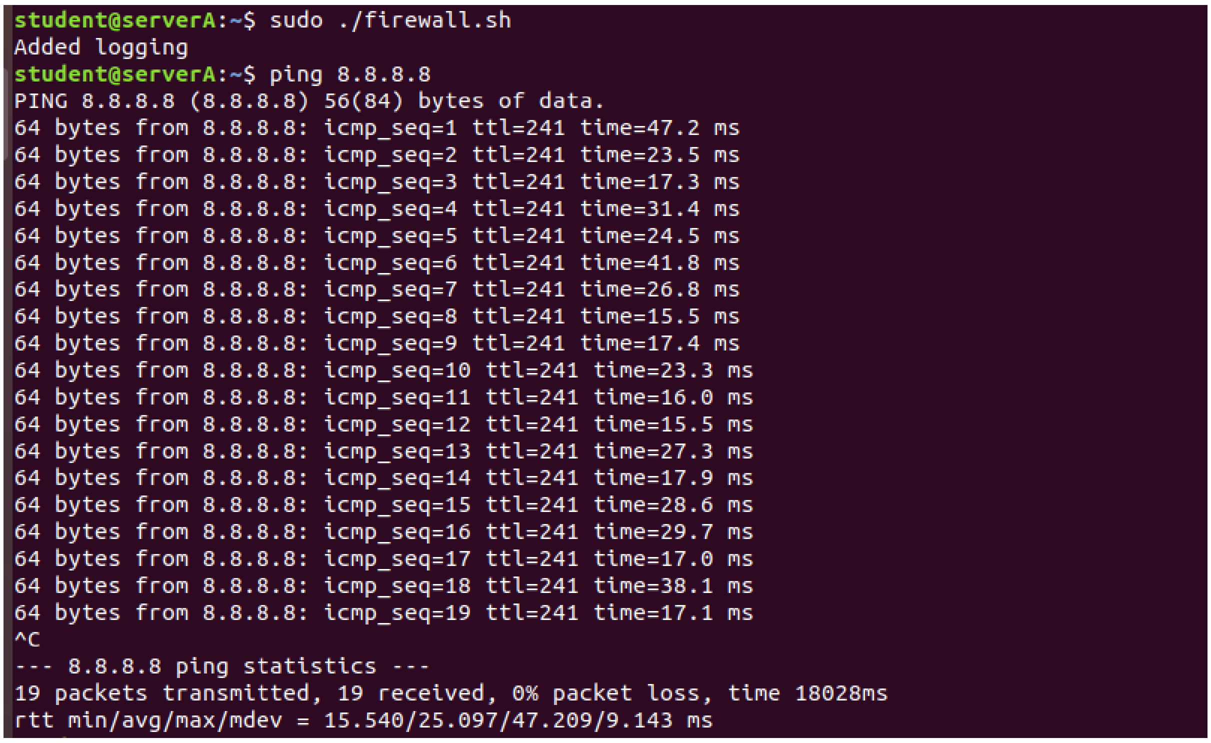
$IPT -A OUTPUT -p icmp –icmp -type 8 -j ACCEPT

$IPT -A INPUT -p icmp –icmp -type 0 -j ACCEPT



```
  GNU nano 2.9.3                     firewall.sh

# Flush all chains in RAW table
$IPT -t raw -F
# Delete any user-defined chains in RAW table
$IPT -t mangle -X

# Default policy is to send to a dropping chain
$IPT -t filter -P INPUT DROP
$IPT -t filter -P OUTPUT DROP
$IPT -t filter -P FORWARD DROP
#$IPT -t filter -A OUTPUT -p tcp --dport 80 -j DROP
#allowing the traffic from the loopback interface
$IPT -A OUTPUT -o lo -j ACCEPT
$IPT -A INPUT -i lo -j ACCEPT
#allowing server+A to ping in other interfaces
$IPT -A OUTPUT -p icmp --icmp-type 8 -j ACCEPT
$IPT -A INPUT -p icmp --icmp-type 0 -j ACCEPT


# Create logging chains

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text^T To Linter   ^_ Go To Line
```

Figure 35: firewall rules



```
student@serverA:~$ sudo ./firewall.sh
Added logging
student@serverA:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=241 time=47.2 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=241 time=23.5 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=241 time=17.3 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=241 time=31.4 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=241 time=24.5 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=241 time=41.8 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=241 time=26.8 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=241 time=15.5 ms
64 bytes from 8.8.8.8: icmp_seq=9 ttl=241 time=17.4 ms
64 bytes from 8.8.8.8: icmp_seq=10 ttl=241 time=23.3 ms
64 bytes from 8.8.8.8: icmp_seq=11 ttl=241 time=16.0 ms
64 bytes from 8.8.8.8: icmp_seq=12 ttl=241 time=15.5 ms
64 bytes from 8.8.8.8: icmp_seq=13 ttl=241 time=27.3 ms
64 bytes from 8.8.8.8: icmp_seq=14 ttl=241 time=17.9 ms
64 bytes from 8.8.8.8: icmp_seq=15 ttl=241 time=28.6 ms
64 bytes from 8.8.8.8: icmp_seq=16 ttl=241 time=29.7 ms
64 bytes from 8.8.8.8: icmp_seq=17 ttl=241 time=17.0 ms
64 bytes from 8.8.8.8: icmp_seq=18 ttl=241 time=38.1 ms
64 bytes from 8.8.8.8: icmp_seq=19 ttl=241 time=17.1 ms
^C
--- 8.8.8.8 ping statistics ---
19 packets transmitted, 19 received, 0% packet loss, time 18028ms
rtt min/avg/max/mdev = 15.540/25.097/47.209/9.143 ms
```

Figure 36: ping traffic.

Task-19: Allow server A to ping all hosts

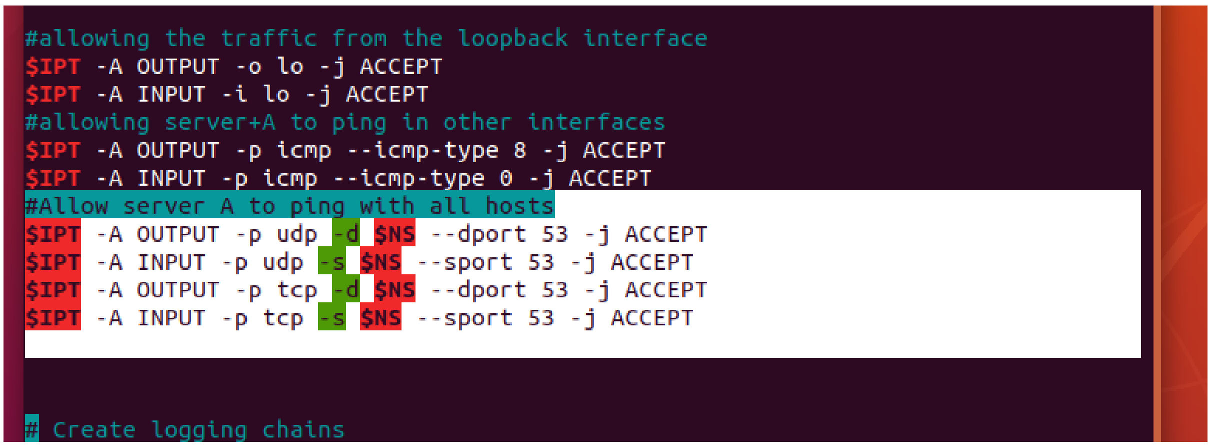
In this task we need to allow the server A to ping all hosts. To achieve it we need to modify the ICMP rules.

$IPT -A OUTPUT -p udp -d $NS –dport 53 -j ACCEPT

$IPT -A INPUT -p udp -s $NS –sport 53 -j ACCEPT

$IPT -A OUTPUT -p udp -d $NS –dport 53 -j ACCEPT

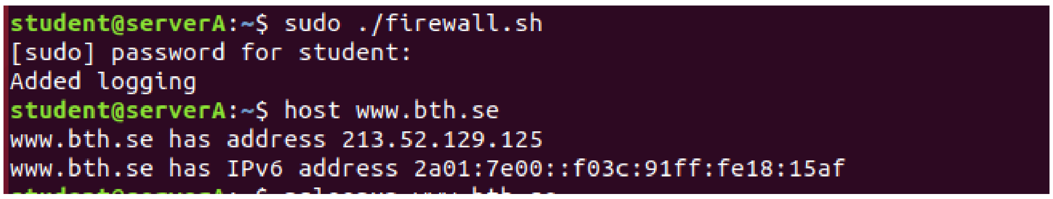
$IPT -A INPUT -p udp -s $NS –sport 53 -j ACCEPT



```
#allowing the traffic from the loopback interface
$IPT -A OUTPUT -o lo -j ACCEPT
$IPT -A INPUT -i lo -j ACCEPT
#allowing server+A to ping in other interfaces
$IPT -A OUTPUT -p icmp --icmp-type 8 -j ACCEPT
$IPT -A INPUT -p icmp --icmp-type 0 -j ACCEPT
#Allow server A to ping with all hosts
$IPT -A OUTPUT -p udp -d $NS --dport 53 -j ACCEPT
$IPT -A INPUT -p udp -s $NS --sport 53 -j ACCEPT
$IPT -A OUTPUT -p tcp -d $NS --dport 53 -j ACCEPT
$IPT -A INPUT -p tcp -s $NS --sport 53 -j ACCEPT


# Create logging chains
```

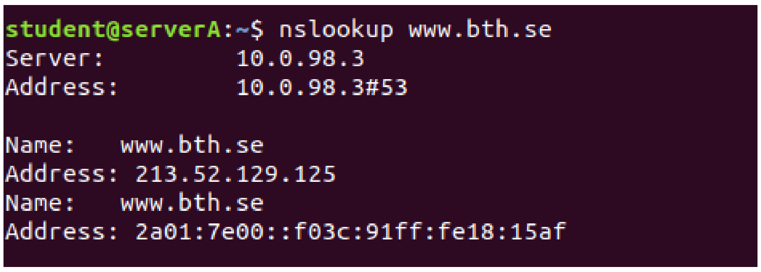
Figure 37: firewall rules.

After adding the rules to the file and executed them. Then we can verify the DNS rules.



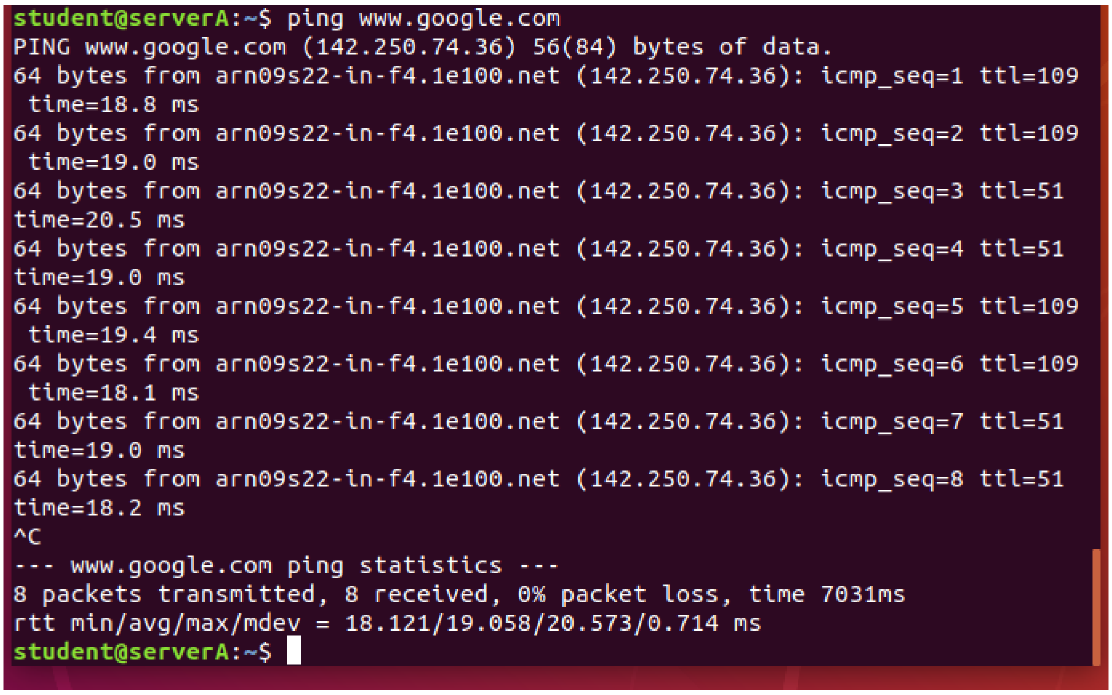
```
student@serverA:~$ sudo ./firewall.sh
[sudo] password for student:
Added logging
student@serverA:~$ host www.bth.se
www.bth.se has address 213.52.129.125
www.bth.se has IPv6 address 2a01:7e00::f03c:91ff:fe18:15af
```

Figure 38: checking the “host www.bth.se”



```
student@serverA:~$ nslookup www.bth.se
Server:         10.0.98.3
Address:        10.0.98.3#53

Name:   www.bth.se
Address: 213.52.129.125
Name:   www.bth.se
Address: 2a01:7e00::f03c:91ff:fe18:15af
```

Figure39: checking the “www.bth.se”



```
student@serverA:~$ ping www.google.com
PING www.google.com (142.250.74.36) 56(84) bytes of data.
64 bytes from arn09s22-in-f4.1e100.net (142.250.74.36): icmp_seq=1 ttl=109
 time=18.8 ms
64 bytes from arn09s22-in-f4.1e100.net (142.250.74.36): icmp_seq=2 ttl=109
 time=19.0 ms
64 bytes from arn09s22-in-f4.1e100.net (142.250.74.36): icmp_seq=3 ttl=51
time=20.5 ms
64 bytes from arn09s22-in-f4.1e100.net (142.250.74.36): icmp_seq=4 ttl=51
time=19.0 ms
64 bytes from arn09s22-in-f4.1e100.net (142.250.74.36): icmp_seq=5 ttl=109
 time=19.4 ms
64 bytes from arn09s22-in-f4.1e100.net (142.250.74.36): icmp_seq=6 ttl=109
 time=18.1 ms
64 bytes from arn09s22-in-f4.1e100.net (142.250.74.36): icmp_seq=7 ttl=51
time=19.0 ms
64 bytes from arn09s22-in-f4.1e100.net (142.250.74.36): icmp_seq=8 ttl=51
time=18.2 ms
^C
--- www.google.com ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7031ms
rtt min/avg/max/mdev = 18.121/19.058/20.573/0.714 ms
student@serverA:~$
```

Figure 40: pining the “www.google.com”

Task-20: Enable state full firewall

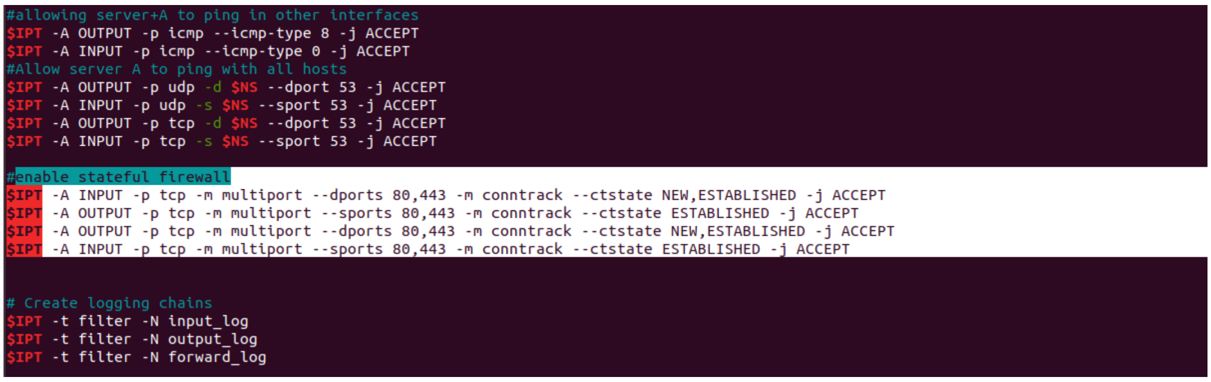
In this task we need to add the rules in the firewall.sh file to enable the TCP connection. The following rules are added

$IPT -A INPUT -p tcp -m multiport –dports 80,443 -m conntrack –ctstate NEW, ESTABLISHED -j ACCEPT

$IPT -A OUTPUT -p tcp -m multiport –dports 80,443 -m conntrack –ctstate ESTABLISHED -j ACCEPT

$IPT -A OUTPUT -p tcp -m multiport –dports 80,443 -m conntrack –ctstate NEW, ESTABLISHED -j ACCEPT

$IPT -A INPUT -p tcp -m multiport –dports 80,443 -m conntrack –ctstate ESTABLISHED -j ACCEPT



```
#allowing server+A to ping in other interfaces
$IPT -A OUTPUT -p icmp --icmp-type 8 -j ACCEPT
$IPT -A INPUT -p icmp --icmp-type 0 -j ACCEPT
#Allow server A to ping with all hosts
$IPT -A OUTPUT -p udp -d $NS --dport 53 -j ACCEPT
$IPT -A INPUT -p udp -s $NS --sport 53 -j ACCEPT
$IPT -A OUTPUT -p tcp -d $NS --dport 53 -j ACCEPT
$IPT -A INPUT -p tcp -s $NS --sport 53 -j ACCEPT

#enable stateful firewall
$IPT -A INPUT -p tcp -m multiport --dports 80,443 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
$IPT -A OUTPUT -p tcp -m multiport --sports 80,443 -m conntrack --ctstate ESTABLISHED -j ACCEPT
$IPT -A OUTPUT -p tcp -m multiport --dports 80,443 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
$IPT -A INPUT -p tcp -m multiport --sports 80,443 -m conntrack --ctstate ESTABLISHED -j ACCEPT


# Create logging chains
$IPT -t filter -N input_log
$IPT -t filter -N output_log
$IPT -t filter -N forward_log
```

Figure 41: firewall rules.

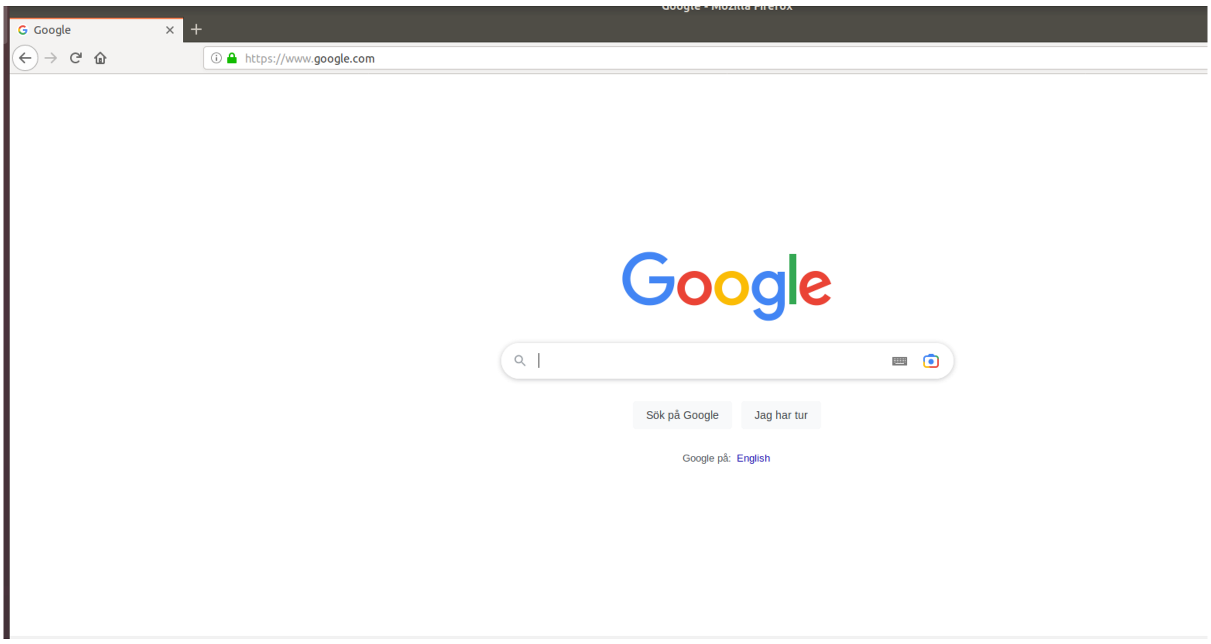
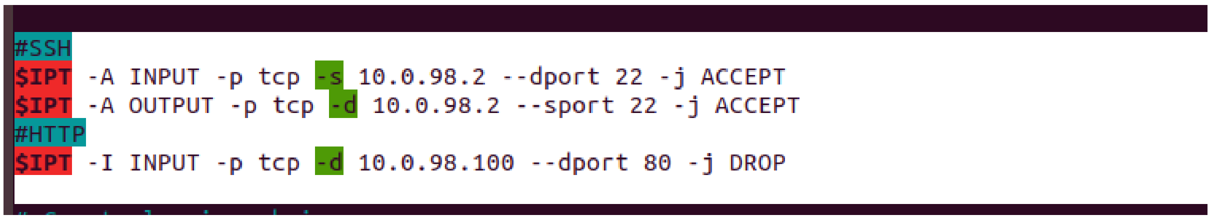


Figure 42: Browsing google website in the Firefox.

Task-21: enable SSH and HTTPS content from apache2 server for web browser on host

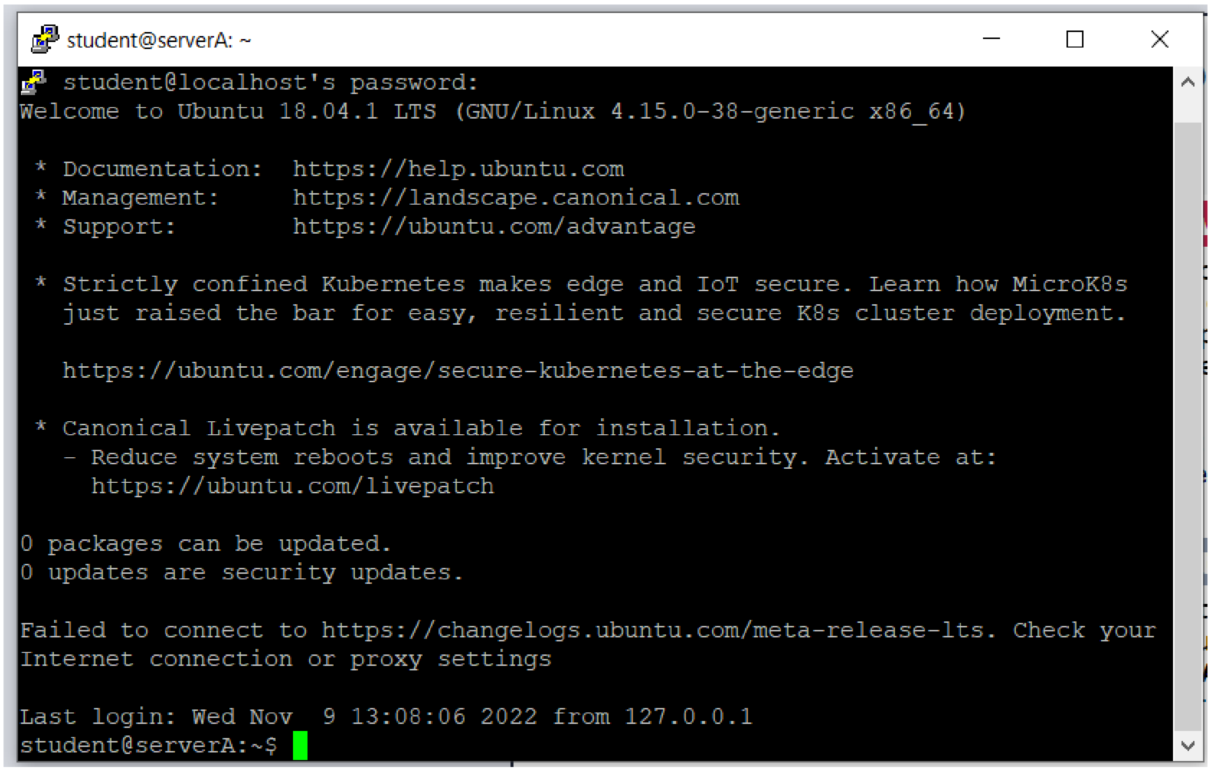
In this task we need enable the SSH and HTTPS but not the HTTP, to do it we need to add the rules in firewall.sh file and execute it. To do that both the SSH and HTTPS rules are accepted and HTTP rules is dropped



```
#SSH
$IPT -A INPUT -p tcp -s 10.0.98.2 --dport 22 -j ACCEPT
$IPT -A OUTPUT -p tcp -d 10.0.98.2 --sport 22 -j ACCEPT
#HTTP
$IPT -I INPUT -p tcp -d 10.0.98.100 --dport 80 -j DROP
```

Figure 43: firewall rules

The below figure44 is enabled the SSH

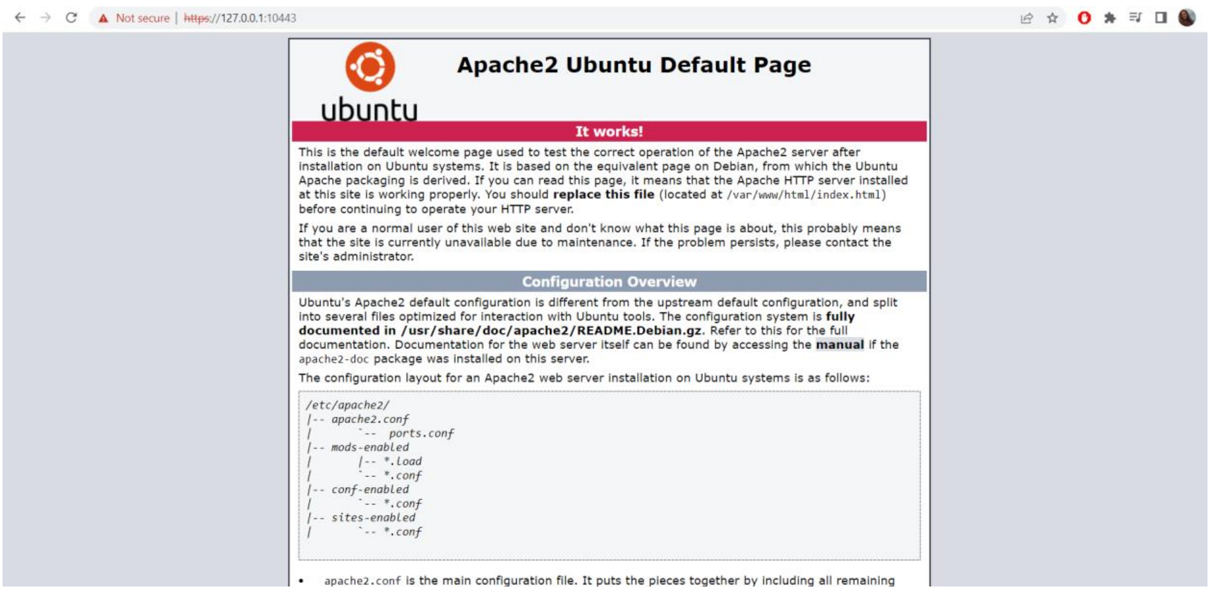


student@serverA: ~

```
student@localhost's password:
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-38-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

0 packages can be updated.
0 updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings

Last login: Wed Nov  9 13:08:06 2022 from 127.0.0.1
student@serverA:~$
```

Figure:44 Connecting SSH

The figure 45 enabled the HTTPS to serve the Apache2.



Not secure | ~~https~~://127.0.0.1:10443

Apache2 Ubuntu Default Page

ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at /var/www/html/index.html) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

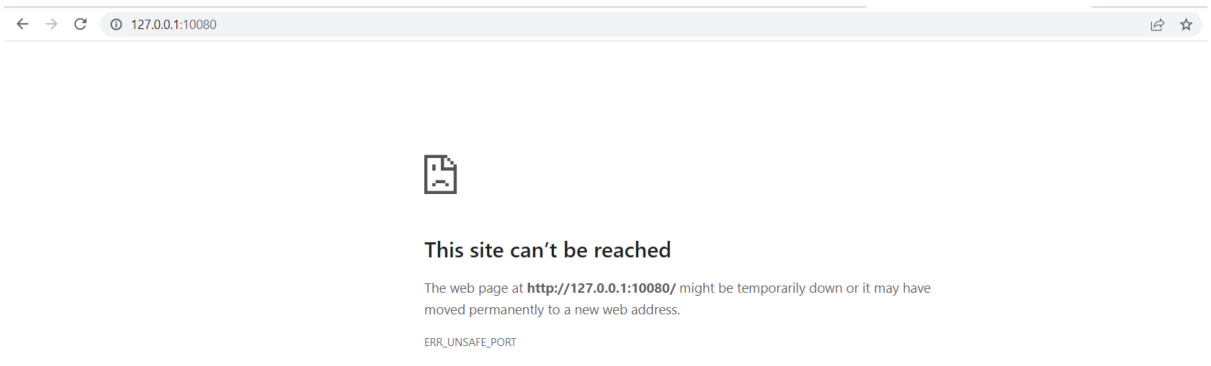
The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- apache2.conf is the main configuration file. It puts the pieces together by including all remaining

Figure 45: Connecting to Apache2 by HTTPS

The below figure 46 blocked HTTP.



```
127.0.0.1:10080
```

This site can't be reached

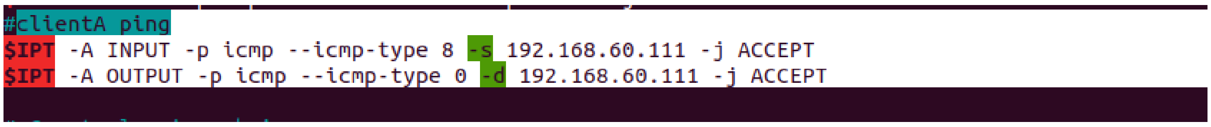
The web page at **http://127.0.0.1:10080/** might be temporarily down or it may have moved permanently to a new web address.

ERR_UNSAFE_PORT

Figure 46: unable to connect from HTTP

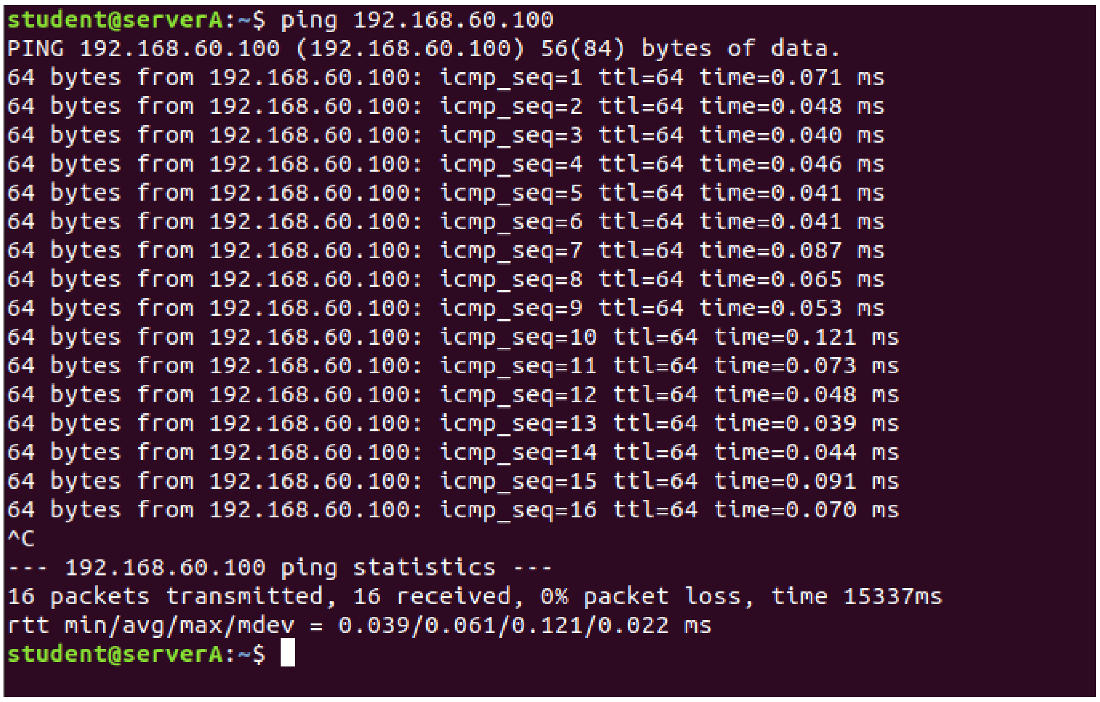
Task-22: ping Server A from client A

In this task we need to ping Server A from client A, the below figure shows the firewall rules.



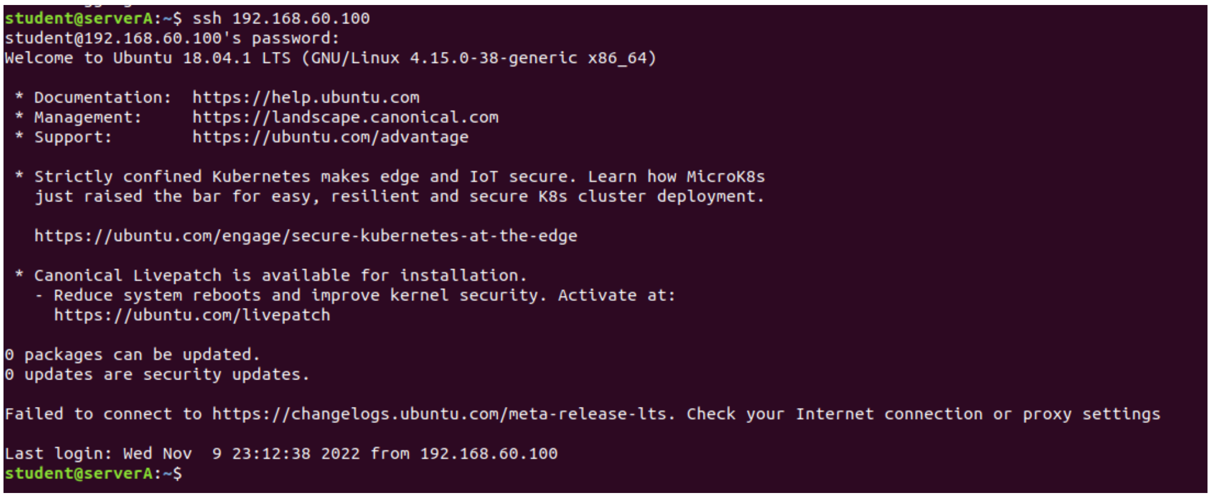
```
#clientA ping
$IPT -A INPUT -p icmp --icmp-type 8 -s 192.168.60.111 -j ACCEPT
$IPT -A OUTPUT -p icmp --icmp-type 0 -d 192.168.60.111 -j ACCEPT
```

Figure 47: firewall rule



```
student@serverA:~$ ping 192.168.60.100
PING 192.168.60.100 (192.168.60.100) 56(84) bytes of data.
64 bytes from 192.168.60.100: icmp_seq=1 ttl=64 time=0.071 ms
64 bytes from 192.168.60.100: icmp_seq=2 ttl=64 time=0.048 ms
64 bytes from 192.168.60.100: icmp_seq=3 ttl=64 time=0.040 ms
64 bytes from 192.168.60.100: icmp_seq=4 ttl=64 time=0.046 ms
64 bytes from 192.168.60.100: icmp_seq=5 ttl=64 time=0.041 ms
64 bytes from 192.168.60.100: icmp_seq=6 ttl=64 time=0.041 ms
64 bytes from 192.168.60.100: icmp_seq=7 ttl=64 time=0.087 ms
64 bytes from 192.168.60.100: icmp_seq=8 ttl=64 time=0.065 ms
64 bytes from 192.168.60.100: icmp_seq=9 ttl=64 time=0.053 ms
64 bytes from 192.168.60.100: icmp_seq=10 ttl=64 time=0.121 ms
64 bytes from 192.168.60.100: icmp_seq=11 ttl=64 time=0.073 ms
64 bytes from 192.168.60.100: icmp_seq=12 ttl=64 time=0.048 ms
64 bytes from 192.168.60.100: icmp_seq=13 ttl=64 time=0.039 ms
64 bytes from 192.168.60.100: icmp_seq=14 ttl=64 time=0.044 ms
64 bytes from 192.168.60.100: icmp_seq=15 ttl=64 time=0.091 ms
64 bytes from 192.168.60.100: icmp_seq=16 ttl=64 time=0.070 ms
^C
--- 192.168.60.100 ping statistics ---
16 packets transmitted, 16 received, 0% packet loss, time 15337ms
rtt min/avg/max/mdev = 0.039/0.061/0.121/0.022 ms
student@serverA:~$
```

Figure 48: ping 192.168.60.100



```
student@serverA:~$ ssh 192.168.60.100
student@192.168.60.100's password:
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-38-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

0 packages can be updated.
0 updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Wed Nov  9 23:12:38 2022 from 192.168.60.100
student@serverA:~$
```

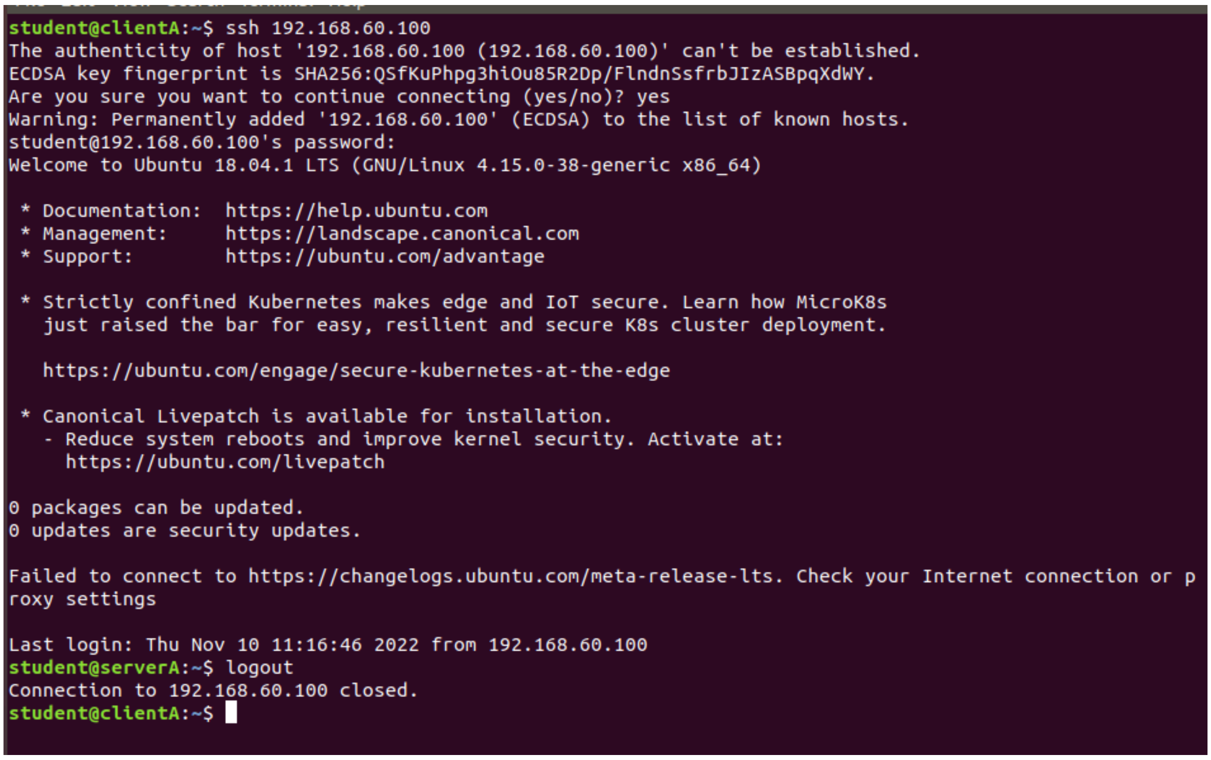
Figure 49: SSH from server A to client A

Task-23:SSH from client A to Server A

In this task we need to connect the SSH from client A to server A, in order to achieve it rules need to be added in the firewall.sh file which is on the server A

$IPT -A INPUT – p tcp -s 192.168.60.111 –dport 22 -m conntrack –ctstate NEW, ESTABLISHED -j ACCEPT

$IPT -A OUTPUT – p tcp -d 192.168.60.111 –sport 22 -m conntrack –ctstate NEW, ESTABLISHED -j ACCEPT

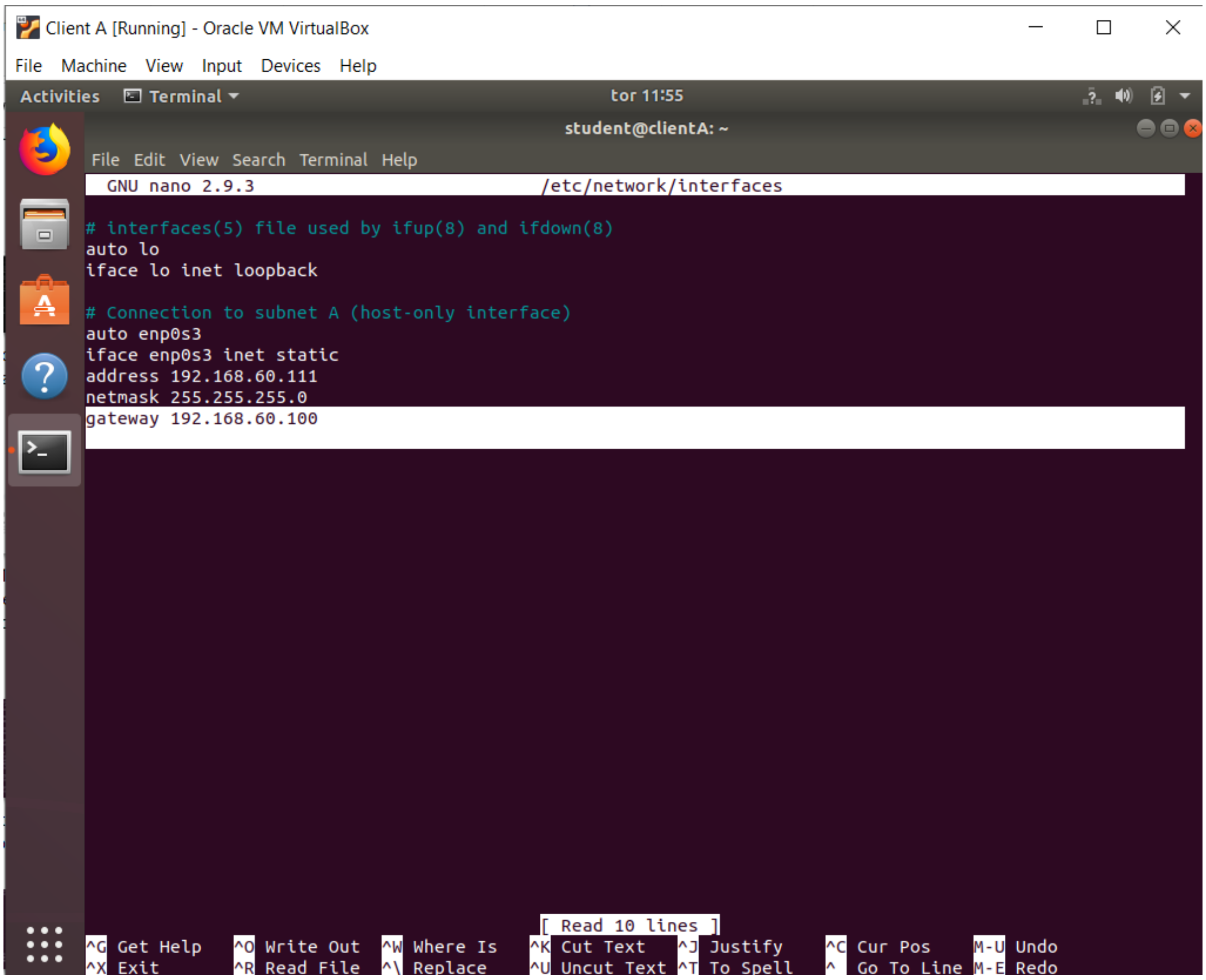


```
student@clientA:~$ ssh 192.168.60.100
The authenticity of host '192.168.60.100 (192.168.60.100)' can't be established.
ECDSA key fingerprint is SHA256:QSfKuPhpg3hiOu85R2Dp/FlndnSsfrbJIzASBpqXdWY.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.60.100' (ECDSA) to the list of known hosts.
student@192.168.60.100's password:
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-38-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

0 packages can be updated.
0 updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or p
roxy settings

Last login: Thu Nov 10 11:16:46 2022 from 192.168.60.100
student@serverA:~$ logout
Connection to 192.168.60.100 closed.
student@clientA:~$
```

Figure 50: connecting SSH from the client A

Task-24: Add gateway and DNS server to client A

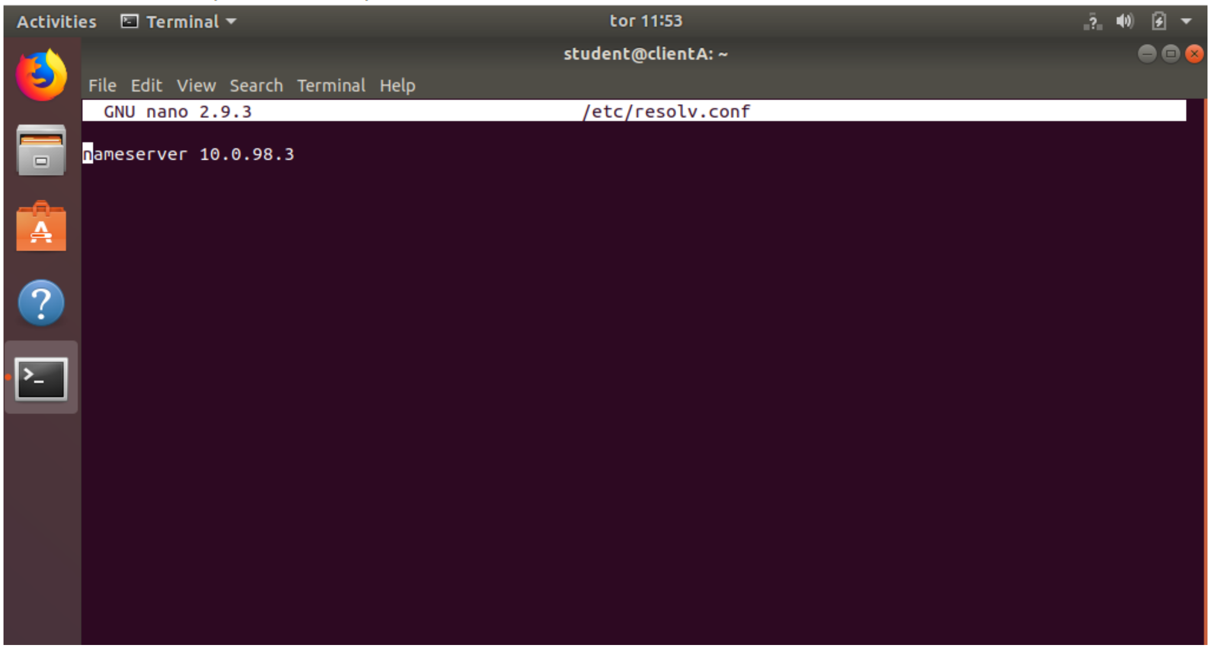
In this task gateway and DNS server need to be added to the client A .to add the gate way we need edit the file **"/etc/network/interfaces"** on client A. The **"gateway 192.168.60.100"** need to be added.


Client A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal
tor 11:55
student@clientA: ~
File Edit View Search Terminal Help

```
  GNU nano 2.9.3                          /etc/network/interfaces

# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback

# Connection to subnet A (host-only interface)
auto enp0s3
iface enp0s3 inet static
address 192.168.60.111
netmask 255.255.255.0
gateway 192.168.60.100

                                    [ Read 10 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos    M-U Undo
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line M-E Redo
```

Figure 51: Adding gateway to client A.

After adding the gateway, list of DNS server needs to be verified weather it has "**10.0.93.3**" by opening the file in the location "**/etc/resolv.conf**".



```
  GNU nano 2.9.3                           /etc/resolv.conf

nameserver 10.0.98.3
```

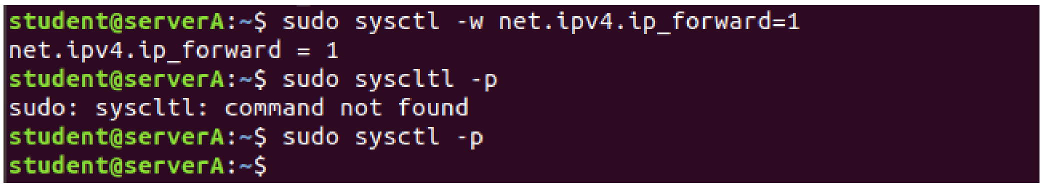
Figure 52: nameserver in the list of DNS.

Task-25: enable IP forwarding on Server A

In this task we need to enable the IP forwarding on serverA by entering the following command on ServerA.

Sudo sysctl -w net.ipv4.ip_forward=1

Sudo sysctl -p



```
student@serverA:~$ sudo sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
student@serverA:~$ sudo sysctltl -p
sudo: sysctltl: command not found
student@serverA:~$ sudo sysctl -p
student@serverA:~$
```

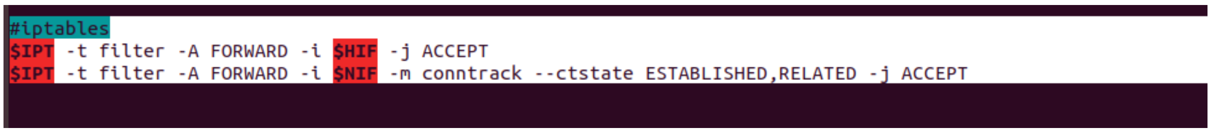
Figure 53: changes on kernel

Task-26: change iptables to forward packets

In this task we need enable the server A to forward the packets to achieve it we need to update the iptables with the following commands.

$IPT -t filter -A FORWARD -i $HIF -j ACCEPT

$IPT -t filter -A FORWARD -i $NIF -m conntrack –ctstate ESTABLISHED, RELATED -j ACCEPT



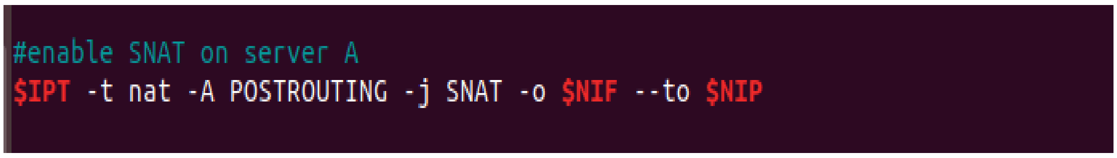
```
#iptables
$IPT -t filter -A FORWARD -i $HIF -j ACCEPT
$IPT -t filter -A FORWARD -i $NIF -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
```

Figure 54: iptables update.

Task-27: enable SNAT on server A

In this task we need to enable the SNAT on server A to access the internet from client, to achieve the task we need add the following command in the firewall code.

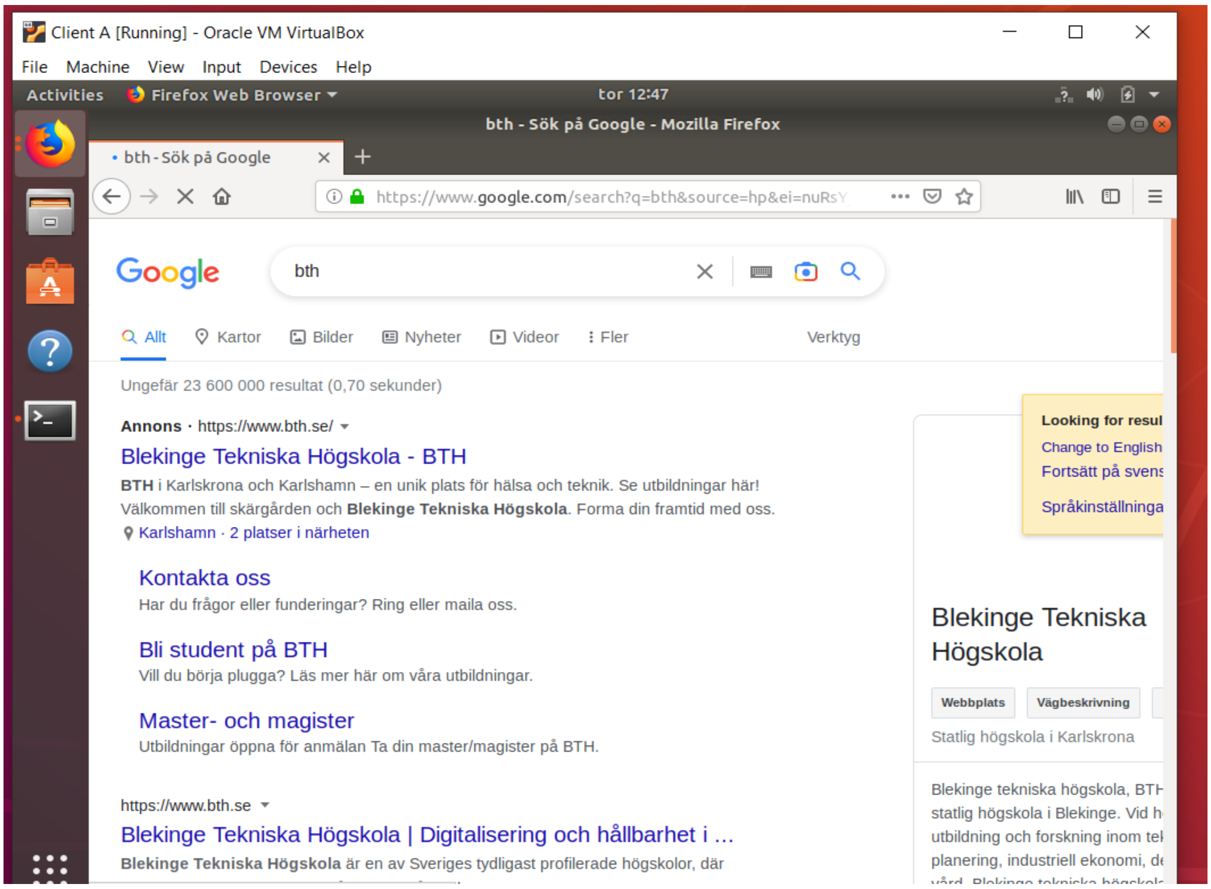
$IPT -t net -A POSTROUTING -j SNAT -o $NIF –to $NIP



```
#enable SNAT on server A
$IPT -t nat -A POSTROUTING -j SNAT -o $NIF --to $NIP
```

Figure 55: firewall rules.

After executing the firewall.sh file in the server A, is we tried to open any website from the Mozilla Firefox we can open it, which means that the internet access is there for the client A



Client A [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Activities Firefox Web Browser tor 12:47

bth - Sök på Google - Mozilla Firefox

bth - Sök på Google

https://www.google.com/search?q=bth&source=hp&ei=nuRsY

bth

Allt Kartor Bilder Nyheter Videor Fler Verktyg

Ungefär 23 600 000 resultat (0,70 sekunder)

Annons · https://www.bth.se/

Blekinge Tekniska Högskola - BTH

BTH i Karlskrona och Karlshamn – en unik plats för hälsa och teknik. Se utbildningar här! Välkommen till skärgården och Blekinge Tekniska Högskola. Forma din framtid med oss.

Karlshamn · 2 platser i närheten

Kontakta oss

Har du frågor eller funderingar? Ring eller maila oss.

Bli student på BTH

Vill du börja plugga? Läs mer här om våra utbildningar.

Master- och magister

Utbildningar öppna för anmälan Ta din master/magister på BTH.

https://www.bth.se

Blekinge Tekniska Högskola | Digitalisering och hållbarhet i ...

Blekinge Tekniska Högskola är en av Sveriges tydligast profilerade högskolor, där

Looking for resul

Change to English

Fortsätt på svens

Språkinställninga

Blekinge Tekniska Högskola

Webbplats Vägbeskrivning

Statlig högskola i Karlskrona

Blekinge tekniska högskola, BTH statlig högskola i Blekinge. Vid h utbildning och forskning inom tek planering, industriell ekonomi, de

Figure 55: internet access to the client A.