

# Lab 1.1: Linux Networking and Firewalls

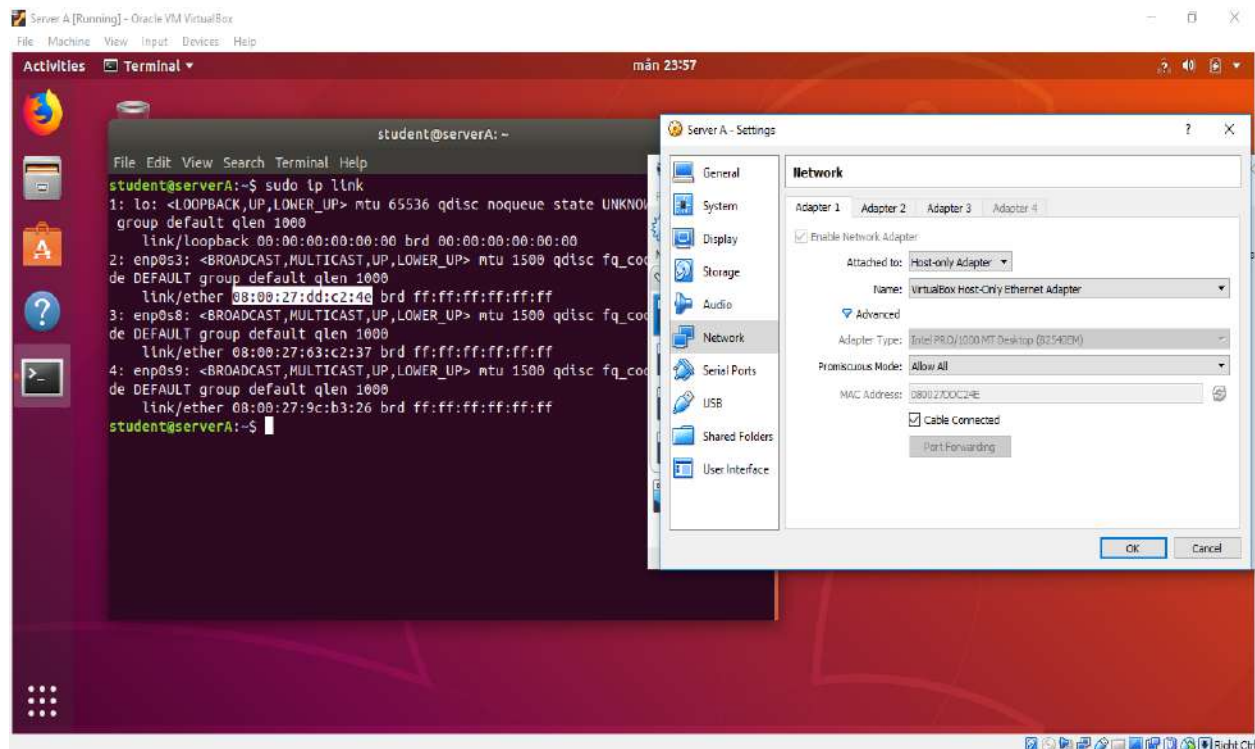
Course: Network Security, ET2540

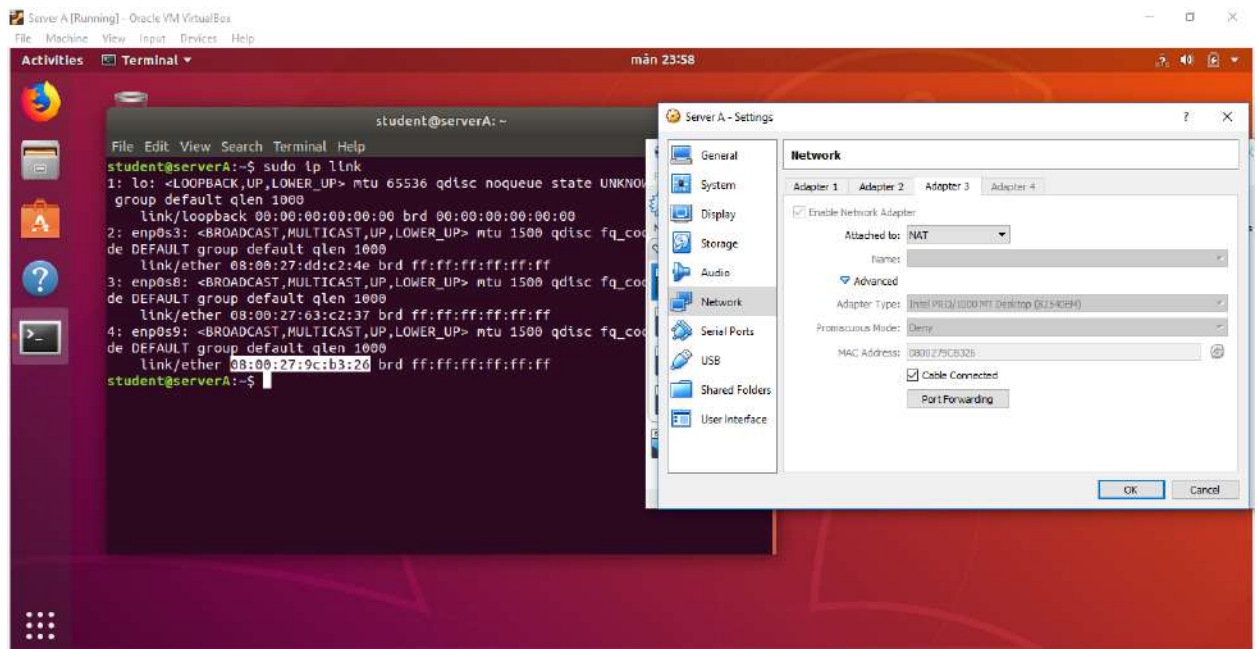
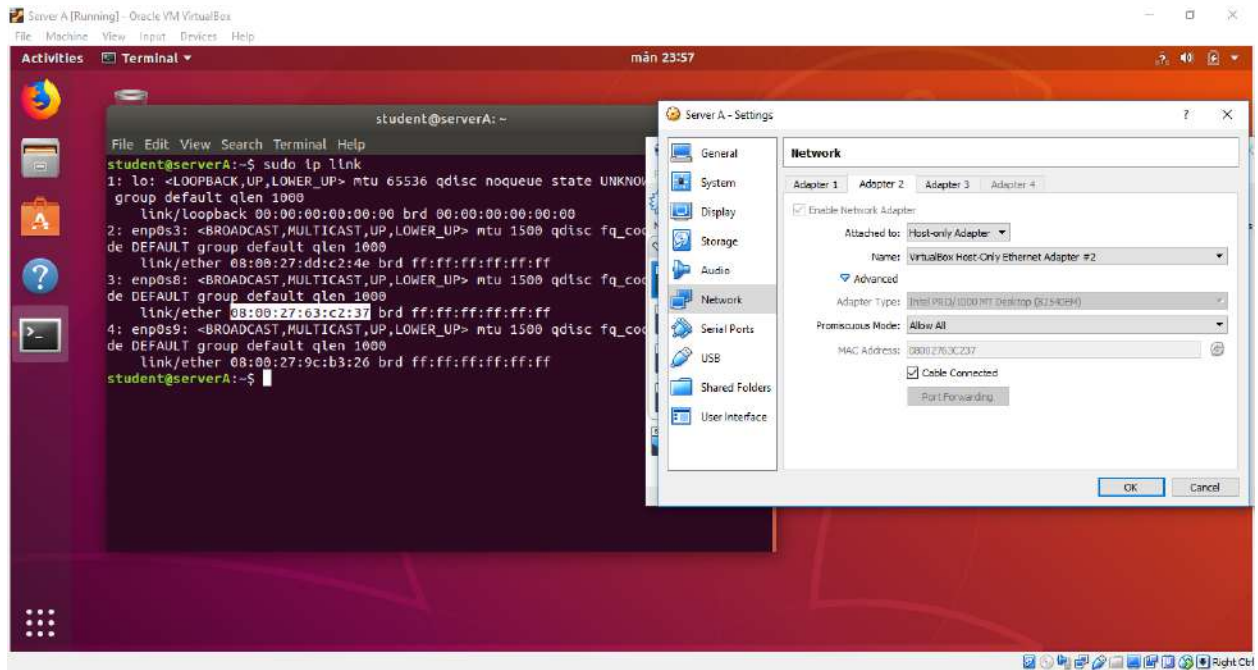
Name: Md Rabiul Ahamed Bin Hanif

Personal Number: 9202122637

## Task 1: MAC Address

| Interface | MAC Address  |
|-----------|--------------|
| Enp0s3    | 080027DDC24E |
| Enp0s8    | 08002763C237 |
| Enp0s9    | 0800279CB326 |





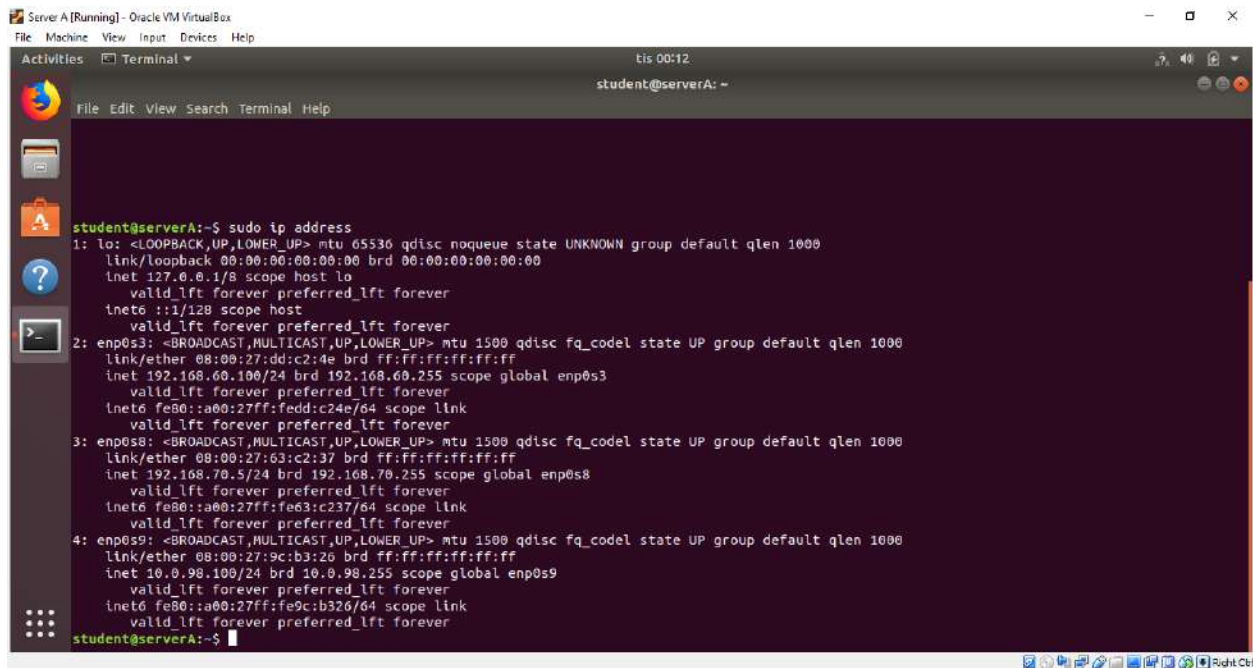
## Task 2: Network Interfaces

Now in guest command window run this command.

```
sudo ip address
```

I can find the Interface, IPv4 address , MAC Address and adapter also.

| Interface | IPv4 Address   | MAC Address  | Adapter   |
|-----------|----------------|--------------|-----------|
| Enp0s3    | 192.168.60.100 | 080027DDC24E | Host Only |
| Enp0s8    | 192.168.70.5   | 08002763C237 | Host Only |
| Enp0s9    | 10.0.98.100    | 0800279CB326 | NAT       |



```
Server A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal
tis 00:12
student@serverA: ~

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:dd:c2:4e 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:fedd:c24e/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 08:00:27:63:c2:37 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:fe63:c237/64 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 08:00:27:9c:b3:26 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:fe9c:b326/64 scope link
        valid_lft forever preferred_lft forever
student@serverA:~$
```

### Task 3: IP addresses, netmasks and subnet

(IPv4 address) AND (Netmasks)= Subnet

| Interface | IPv4 address | MAC Address  | Subnet     |
|-----------|--------------|--------------|------------|
| enp0s3    | 192.168.60.1 | 080027C56E84 | 192.168.60 |
| enp0s8    | 192.168.70.1 | 08002797F5E0 | 192.168.70 |
| enp0s9    | 10.0.98.100  | 0800272841E0 | 10.0.98.0  |

For enp0s3:

|                      |          |          |          |          |
|----------------------|----------|----------|----------|----------|
| IP Address (Decimal) | 192      | 168      | 60       | 1        |
| IP Address (Binary)  | 11000000 | 10101000 | 00111100 | 00000001 |
| Netmasks (B)         | 11111111 | 11111111 | 11111111 | 00000000 |
| AND operation        |          |          |          |          |
| Subnet (B)           | 11000000 | 10101000 | 00111100 | 00000000 |
| Subnet (D)           | 192      | 168      | 60       | 0        |

For enp0s8:

|                      |          |          |          |          |
|----------------------|----------|----------|----------|----------|
| IP Address (Decimal) | 192      | 168      | 70       | 1        |
| IP Address (Binary)  | 11000000 | 10101000 | 01010000 | 00000001 |
| Netmasks (B)         | 11111111 | 11111111 | 11111111 | 00000000 |
| AND operation        |          |          |          |          |
| Subnet (B)           | 11000000 | 10101000 | 01010000 | 00000000 |
| Subnet (D)           | 192      | 168      | 70       | 0        |

For enp0s9:

|                      |          |          |          |          |
|----------------------|----------|----------|----------|----------|
| IP Address (Decimal) | 10       | 0        | 98       | 100      |
| IP Address (Binary)  | 00001010 | 00000000 | 01100010 | 01100100 |
| Netmasks (B)         | 11111111 | 11111111 | 11111111 | 00000000 |
| AND operation        |          |          |          |          |
| Subnet (B)           | 00001010 | 00000000 | 01100010 | 01100100 |
| Subnet (D)           | 10       | 0        | 98       | 0        |

## Task 4: Host-only interfaces

In host OS command window I run this command

```
ipconfig/all
```

I found the host interface list from the host OS.

| Host-Only interface                              | IPv4 Address On Host | Subnet Mask   | Host Only Interface in the Guest | IP address on the Guest |
|--|----------------------|---------------|----------------------------------|-------------------------|
| Ethernet adapter VirtualBox Host-Only Network    | 192.168.60.1         | 255.255.255.0 | enp0s3                           | 192.168.60.100          |
| Ethernet adapter VirtualBox Host-Only Network #2 | 192.168.70.1         | 255.255.255.0 | enp0s8                           | 192.168.70.5            |

The host only interface is Ethernet adapter Virtual Box-Host-Only Network (192.168.60.1) which is connected to enp0s3 (192.168.60.100) of Guest OS & Ethernet adapter Virtual Box Host-Only network #2 (192.168.70.1) which connected to enp0s8 (192.168.70.5) of guest OS.

```
Ethernet adapter VirtualBox Host-Only Network:

Connection-specific DNS Suffix  . : 
Description . . . . . : VirtualBox Host-Only Ethernet Adapter
Physical Address. . . . . : 0A-00-27-00-00-11
DHCP Enabled. . . . . : No
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::5002:c40e:7229:8faa%17(Preferred)
IPv4 Address. . . . . : 192.168.60.1(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 
DHCPv6 IAID . . . . . : 84541479
DHCPv6 Client DUID. . . . . : 00-01-00-01-23-92-69-F6-74-86-7A-59-E9-77
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 VirtualBox Host-Only Network #2:

```
Connection-specific DNS Suffix . :  
Description . . . . . : VirtualBox Host-Only Ethernet Adapter #2  
Physical Address. . . . . : 0A-00-27-00-00-2B  
DHCP Enabled. . . . . : No  
Autoconfiguration Enabled . . . . : Yes  
Link-local IPv6 Address . . . . . : fe80::b9b6:6262:5d93:a485%43(Preferred)  
IPv4 Address. . . . . : 192.168.70.1(Preferred)  
Subnet Mask . . . . . : 255.255.255.0  
Default Gateway . . . . . :  
DHCPv6 IAID . . . . . : 722075687  
DHCPv6 Client DUID. . . . . : 00-01-00-01-23-92-69-F6-74-86-7A-59-E9-77  
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 VirtualBox Host-Only Network #3:

```
Connection-specific DNS Suffix . :  
Description . . . . . : VirtualBox Host-Only Ethernet Adapter #3  
Physical Address. . . . . : 0A-00-27-00-00-2F  
DHCP Enabled. . . . . : No  
Autoconfiguration Enabled . . . . : Yes  
Link-local IPv6 Address . . . . . : fe80::c50:e4e:60b5:3ccb%47(Preferred)  
IPv4 Address. . . . . : 192.168.80.1(Preferred)  
Subnet Mask . . . . . : 255.255.255.0  
Default Gateway . . . . . :  
DHCPv6 IAID . . . . . : 789184551  
DHCPv6 Client DUID. . . . . : 00-01-00-01-23-92-69-F6-74-86-7A-59-E9-77  
DNS Servers . . . . . : fec0:0:0:ffff::1%1  
                        fec0:0:0:ffff::2%1  
                        fec0:0:0:ffff::3%1  
NetBIOS over Tcpip. . . . . : Enabled
```



## Task 5: Routing tables in the host OS

I have run the command `route -4 PRINT`

By the interface 10.59.0.64 and I can reach the gateway 10.59.0.64 for the host OS.

```
ca Command Prompt
C:\Users\purno>route -4 PRINT
=====
Interface List
 6...74 86 7a 59 e9 77 .....Realtek PCIe FE Family Controller
17...0a 00 27 00 00 11 .....VirtualBox Host-Only Ethernet Adapter
43...0a 00 27 00 00 2b .....VirtualBox Host-Only Ethernet Adapter #2
47...0a 00 27 00 00 2f .....VirtualBox Host-Only Ethernet Adapter #3
 9...16 5a 04 ac ad a2 .....Microsoft Wi-Fi Direct Virtual Adapter
 7...64 5a 04 ac ad a2 .....Dell Wireless 1705 802.11b|g|n (2.4GHZ)
14...64 5a 04 ac ad a3 .....Bluetooth Device (Personal Area Network)
 1.....Software Loopback Interface 1
15...00 00 00 00 00 00 e0 Microsoft Teredo Tunneling Adapter
=====

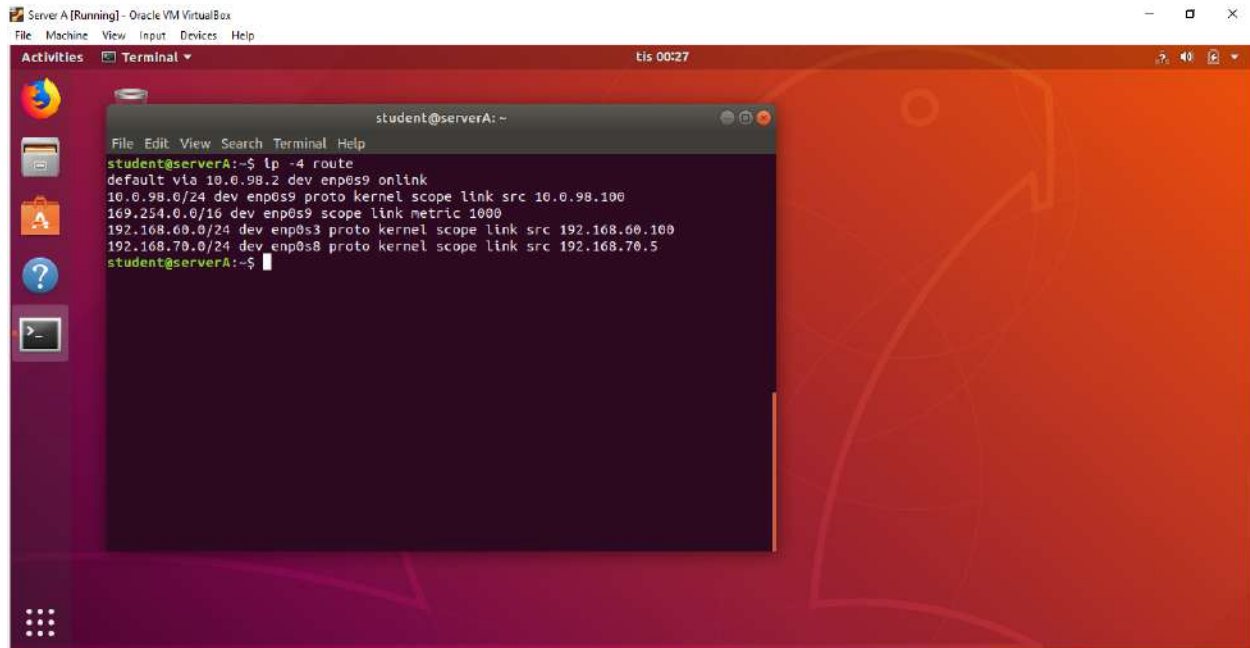
IPv4 Route Table
=====
Active Routes:
Network Destination        Netmask          Gateway          Interface        Metric
0.0.0.0                    0.0.0.0          10.59.0.1        10.59.0.64       55
10.59.0.0                  255.255.255.0    On-link          10.59.0.64       311
10.59.0.64                 255.255.255.255  On-link          10.59.0.64       311
10.59.0.255                255.255.255.255  On-link          10.59.0.64       311
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
192.168.60.0               255.255.255.0    On-link          192.168.60.1     281
192.168.60.1               255.255.255.255  On-link          192.168.60.1     281
192.168.60.255             255.255.255.255  On-link          192.168.60.1     281
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
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          192.168.60.1     281
224.0.0.0                  240.0.0.0        On-link          10.59.0.64       311
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          192.168.60.1     281
255.255.255.255            255.255.255.255  On-link          10.59.0.64       311
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
```

## Task 6: Routing tables in the guest OS

I run in the guest OS this command

```
ip -4 route
```

By the NAT interface enp0s9 I can found default gateway 10.0.98.2.



The screenshot shows a terminal window titled 'student@serverA: ~' within a virtual machine environment. The terminal output of the command 'ip -4 route' is as follows:

```
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 enp0s9 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:~$
```



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

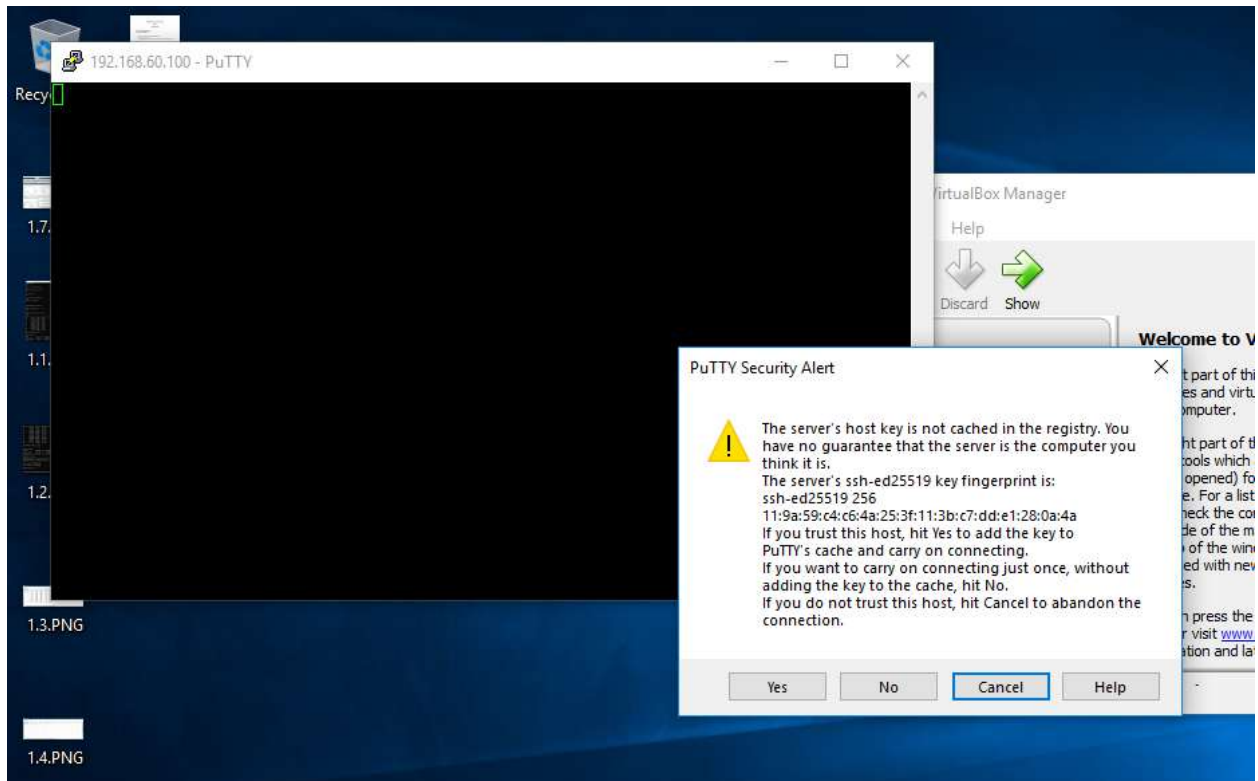
In this task at first turn off the firewall system of the Host OS after that I run the command ping 192.168.60.1 in guest OS beside that I opened both wireshark check the both ping signal no loss.

The screenshot shows a VirtualBox window titled "Server A [Running] - Oracle VM VirtualBox". Inside the VM, a terminal window displays the command `student@serverA:~$ ping 192.168.60.1` and its output, which shows successful ping results with 20 packets and a 1.04 ms round-trip time. Two Wireshark windows are open, both capturing traffic on the `enp0s3` interface. The left Wireshark window shows a display filter of `enp0s3` and a list of captured packets, including ICMP Echo (ping) requests and replies. The right Wireshark window shows a display filter of `enp0s3` and a list of captured packets, including ICMP Echo (ping) requests and replies. The packet details pane in the right Wireshark window shows the selected packet (No. 31) as an ICMP Echo (ping) request from 192.168.60.1 to 192.168.60.255.

The screenshot shows a VirtualBox window titled "Server A [Running] - Oracle VM VirtualBox". Inside the VM, a terminal window displays the command `student@serverA:~$ ping 192.168.60.1` and its output, which shows successful ping results with 20 packets and a 1.04 ms round-trip time. Two Wireshark windows are open, both capturing traffic on the `enp0s3` interface. The left Wireshark window shows a display filter of `enp0s3` and a list of captured packets, including ICMP Echo (ping) requests and replies. The right Wireshark window shows a display filter of `enp0s3` and a list of captured packets, including ICMP Echo (ping) requests and replies. The packet details pane in the right Wireshark window shows the selected packet (No. 31) as an ICMP Echo (ping) request from 192.168.60.1 to 192.168.60.255.

## Task 8: SSH into VM via localhost

I have opened remote shell server use putty software and it successfully worked.



```
student@serverA: ~
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

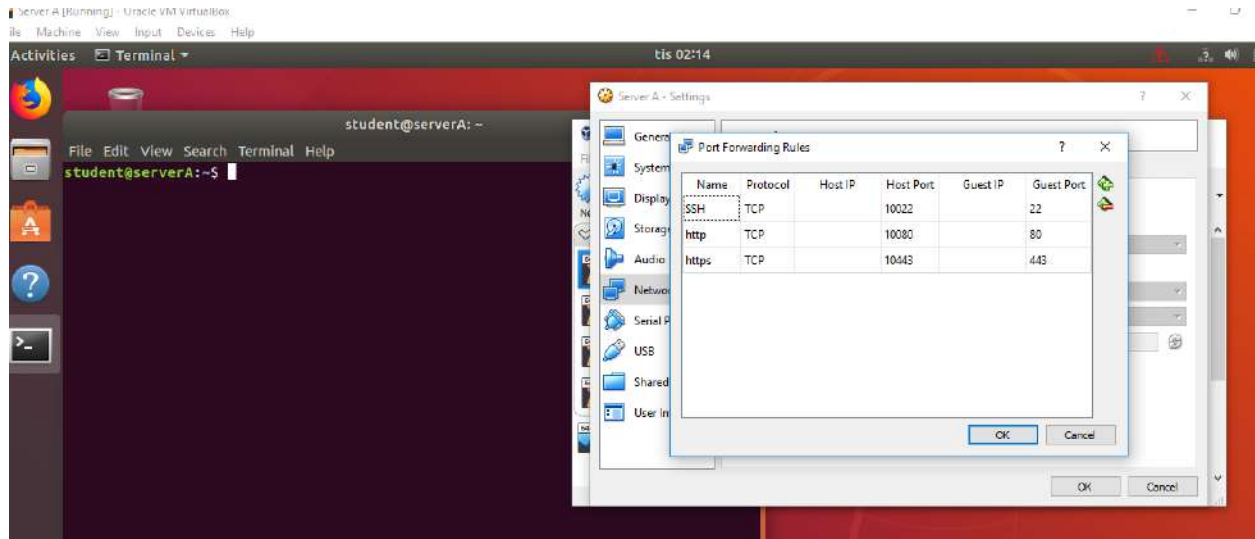
 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

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

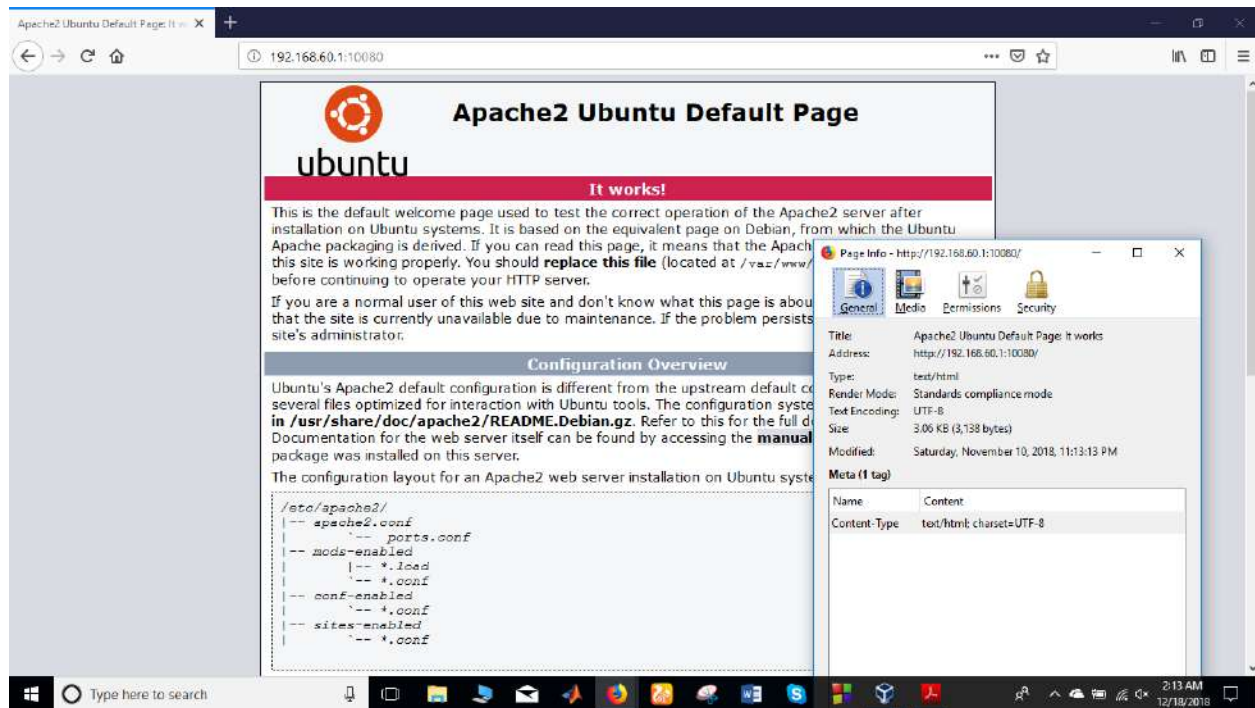
Last login: Sat Nov 10 23:11:06 2018 from 10.0.98.2
student@serverA:~$ sudo ip address
[sudo] password for student:
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        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:dd:c2:4e 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:fedd:c24e/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:e3:c2:37 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:fe63:c237/64 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 08:00:27:9c:b3:26 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:fe9c:b326/64 scope link
        valid_lft forever preferred_lft forever
student@serverA:~$
```

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

I have used host port number 10080 for HTTP and 10443 for HTTPS.

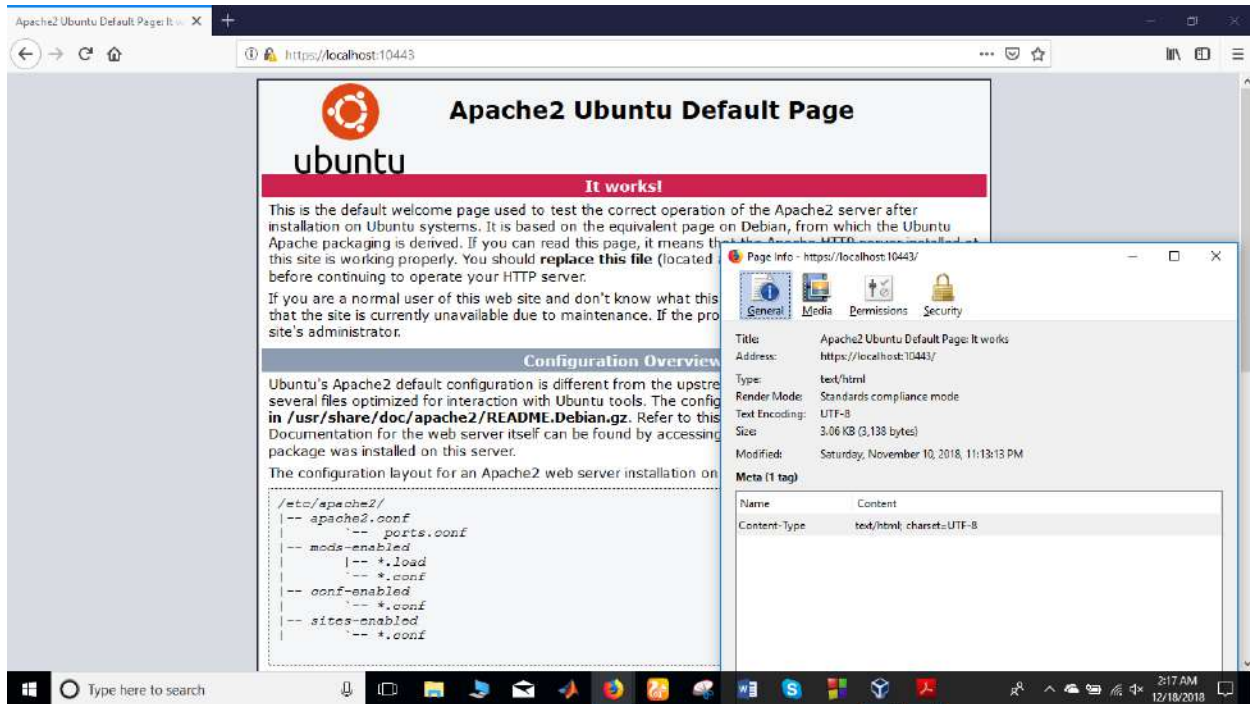


Apache2 server in the host over HTTP

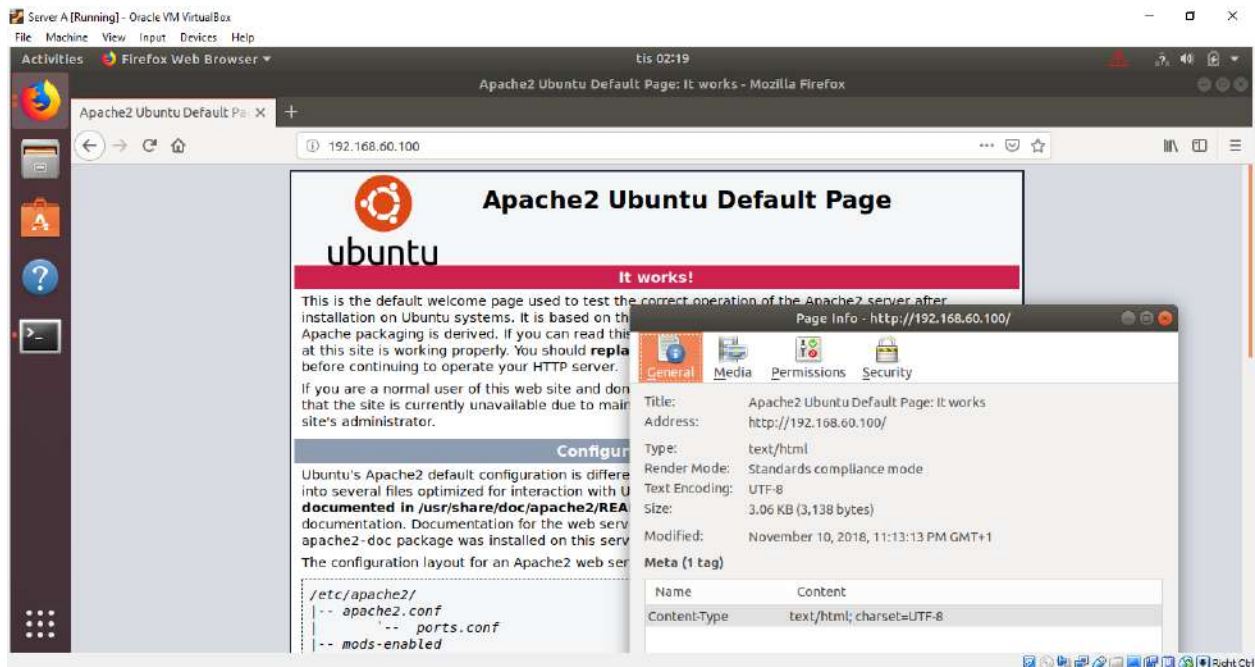




## Apache2 server in the host over HTTPS



## Apache2 server in the guest over HTTP



## Task 10: Default firewall policy and rules

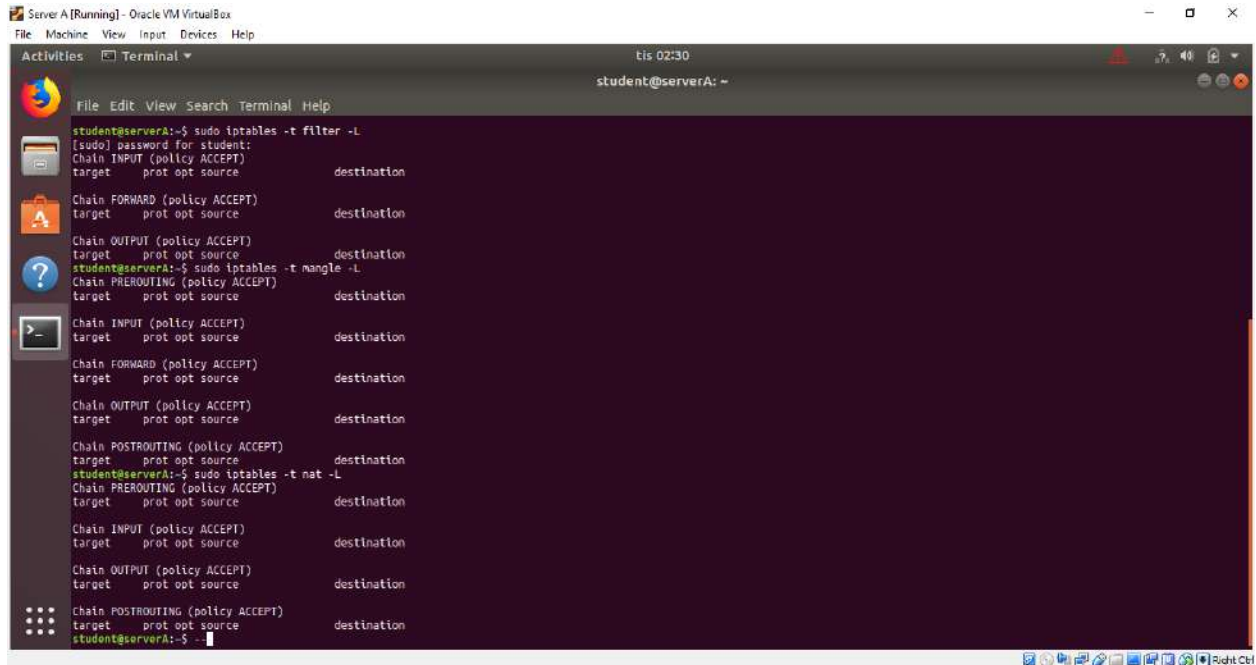
I have run the following commands

```
sudo iptables -t filter -L
```

```
sudo iptables -t mangle -L
```

```
sudo iptables -t nat -L
```

For INPUT, OUTPUT and FORWARD chains, they allow all data, do not drop any data.



The screenshot shows a terminal window titled "Server A [Running] - Oracle VM VirtualBox". The terminal displays the output of three iptables commands: `sudo iptables -t filter -L`, `sudo iptables -t mangle -L`, and `sudo iptables -t nat -L`. Each command output shows the default policy for each chain (ACCEPT) and the target (source or destination) for each rule. The terminal also shows the user prompt `student@serverA: ~` and the time `11:02:30`.

```
student@serverA:~$ sudo iptables -t filter -L
(sudo) password for student:
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 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 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
student@serverA:~$
```

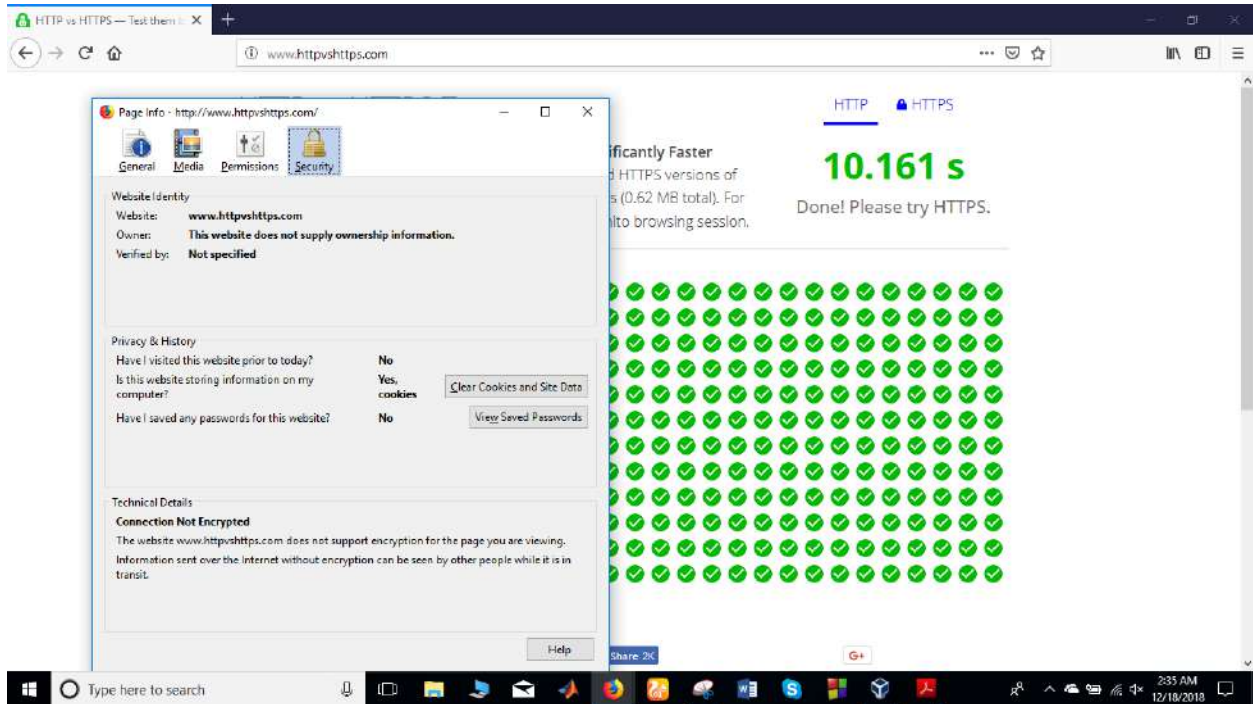
## Task 11: Block HTTP-browsing in the guest OS

I have run these iptables rules

```
sudo iptables -A OUTPUT -p tcp --dport 80 -j REJECT
```

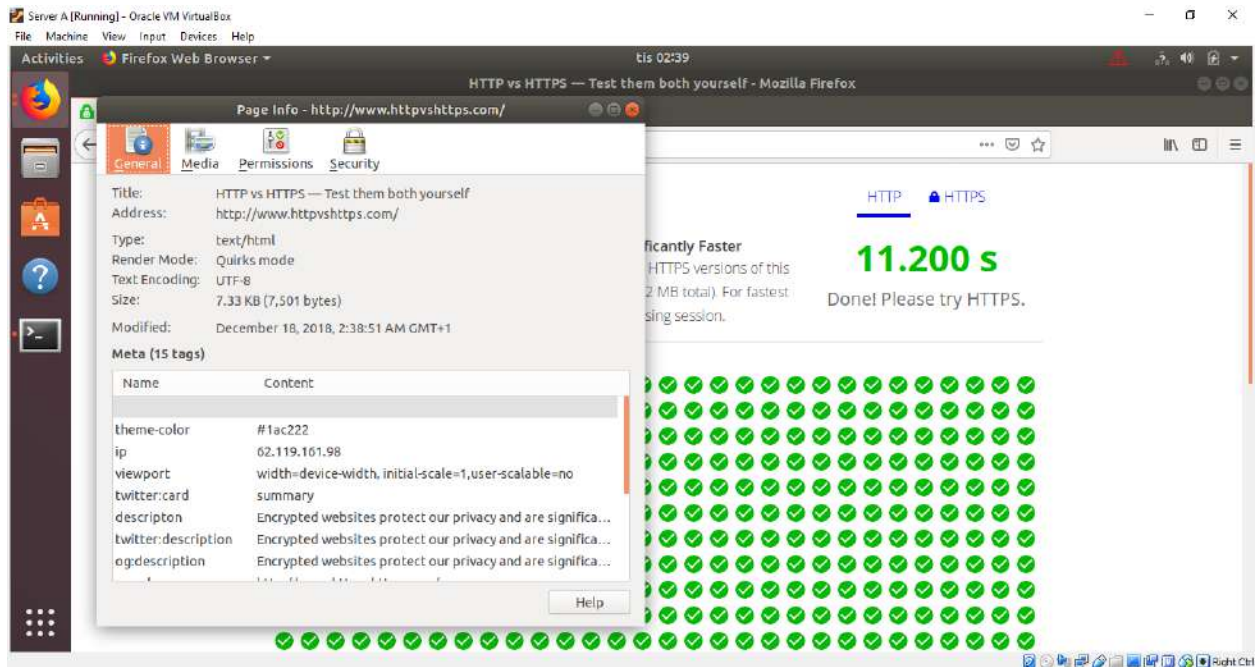
These rules block the user in the guest OS for HTTP.

Web page over HTTP on Host

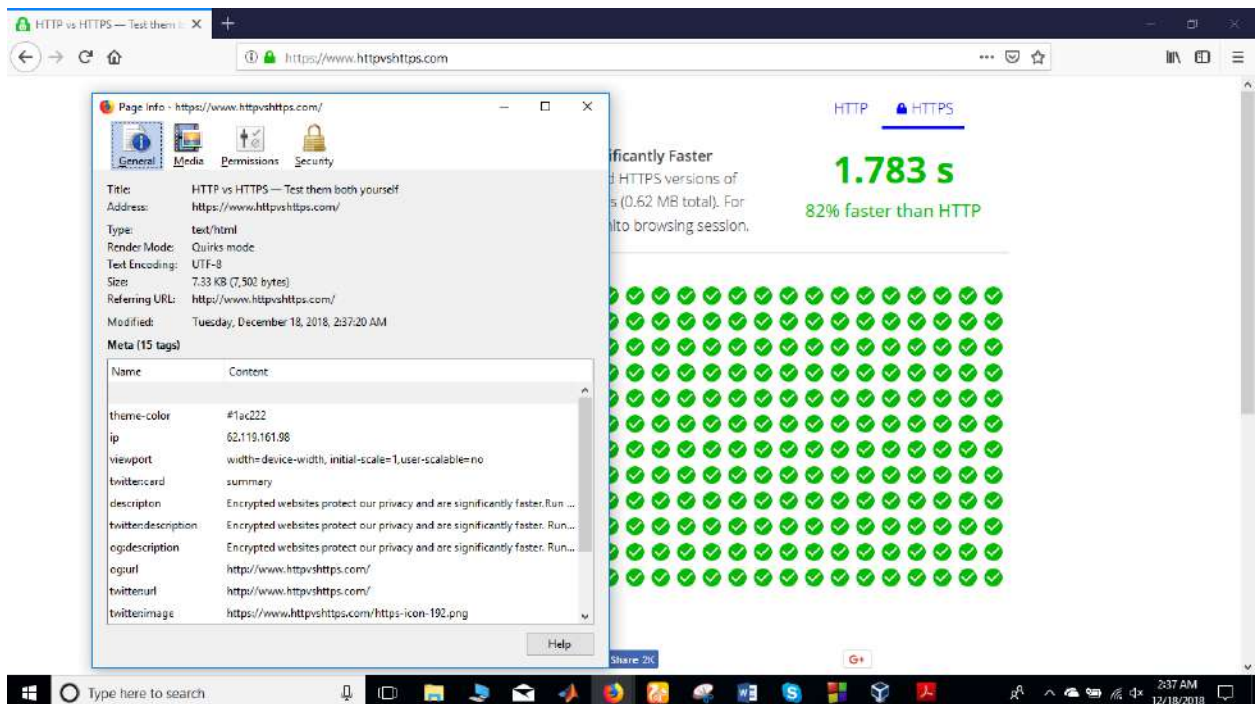




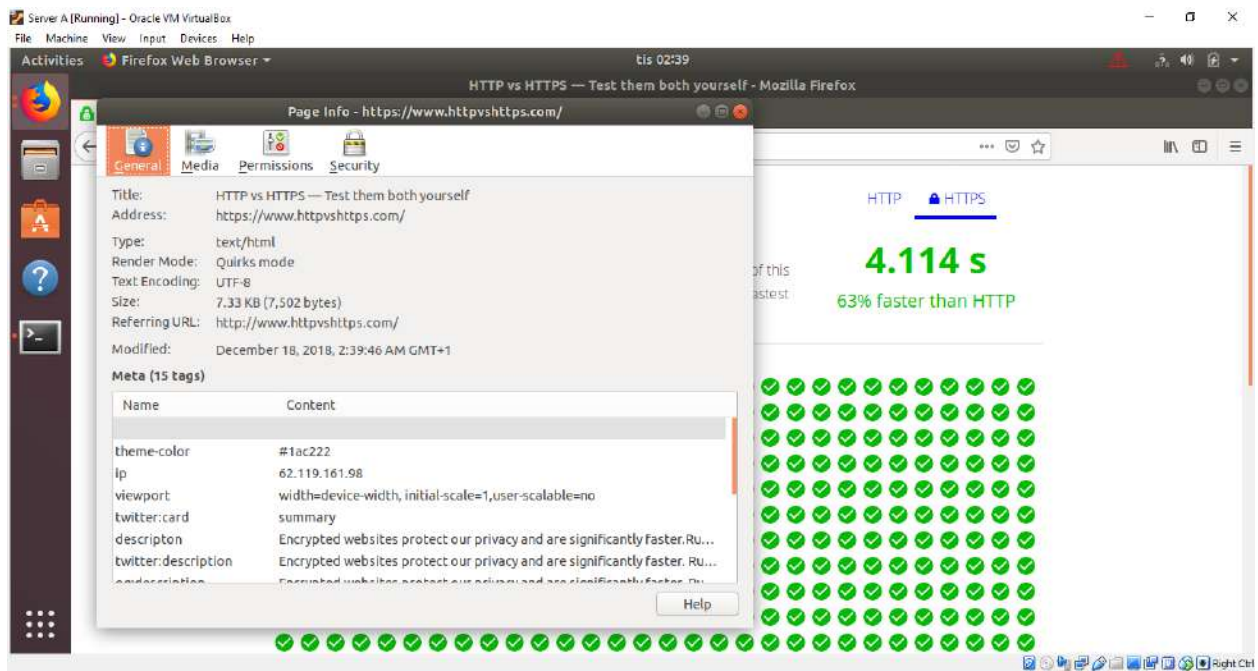
## Web page over HTTP on Guest



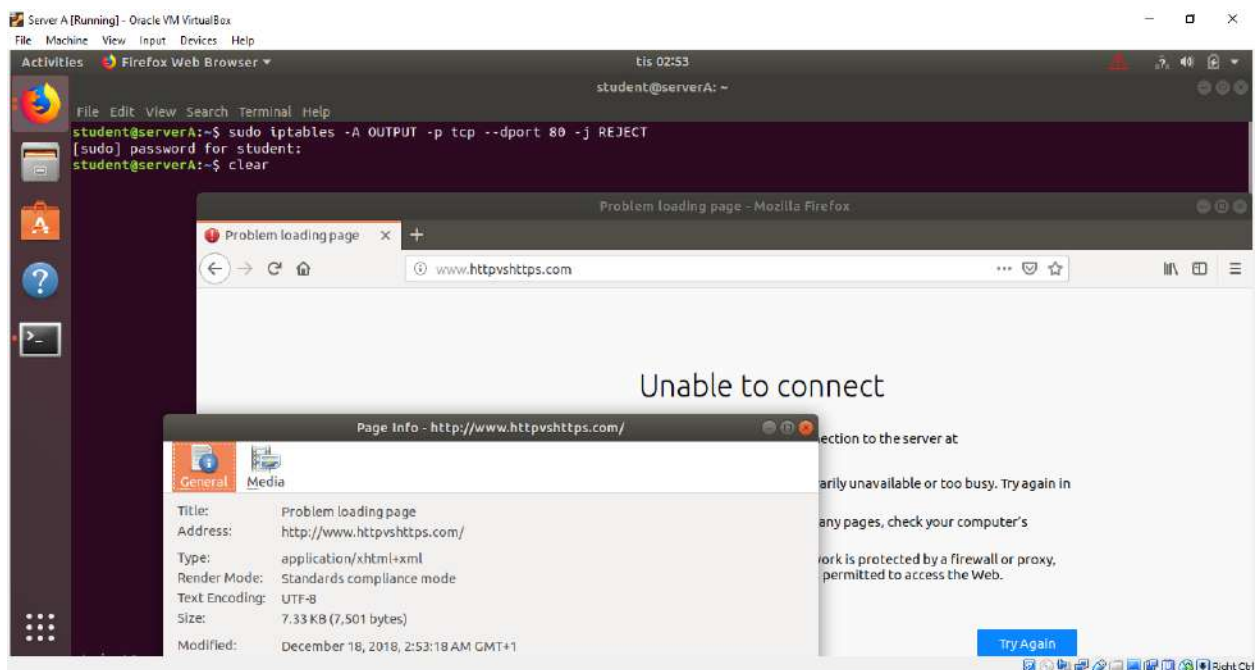
## Web page over HTTPS on Host



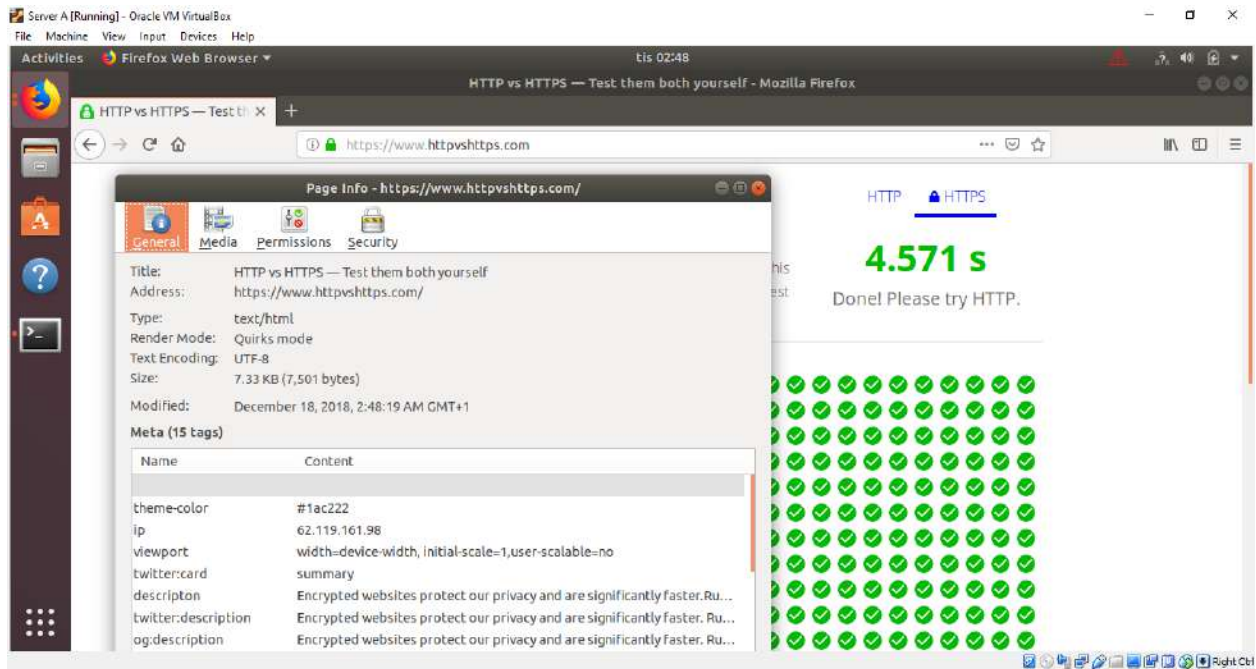
## Web page over HTTPS on Guest



## After running the command block the HTTP



But still can browse HTTPS

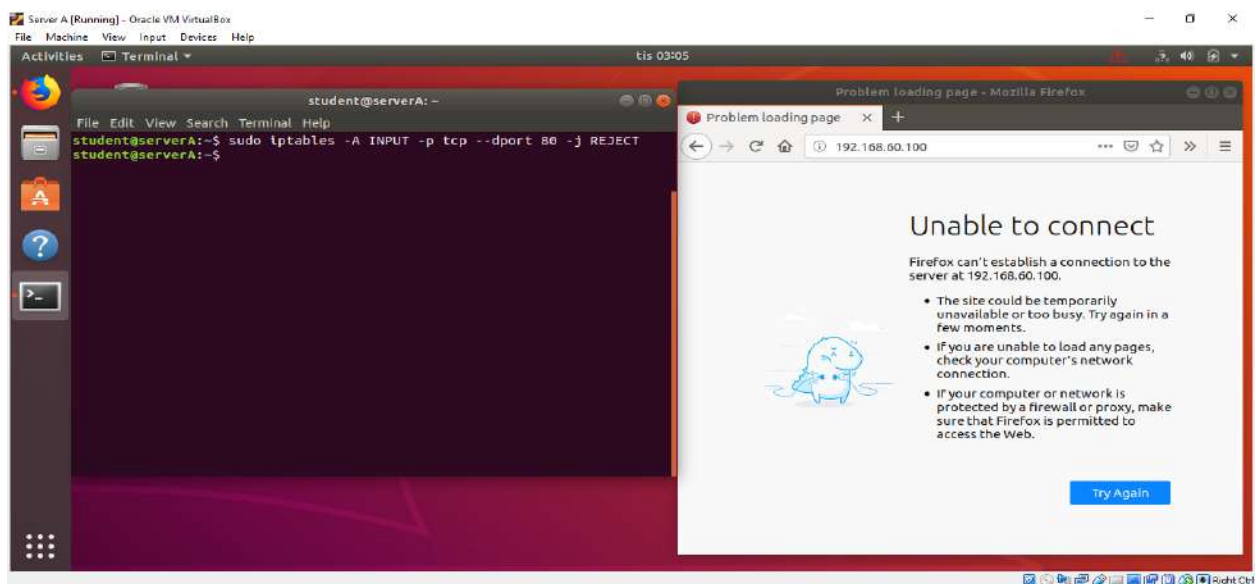


## Task 12: Block Apache web server from serving content over HTTP

In guest command window

```
sudo iptables -A INPUT -p tcp --dport 80 -j REJECT
```

After run upper command host cannot view HTTP content from the apache2 server in the guest OS.



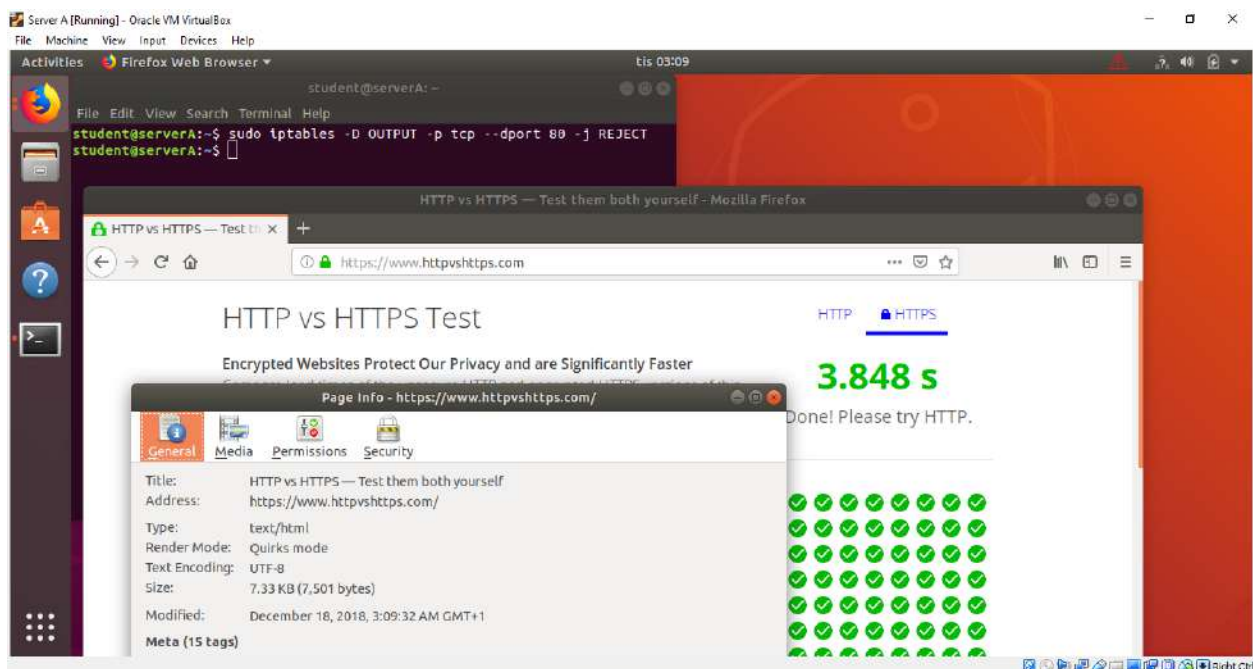
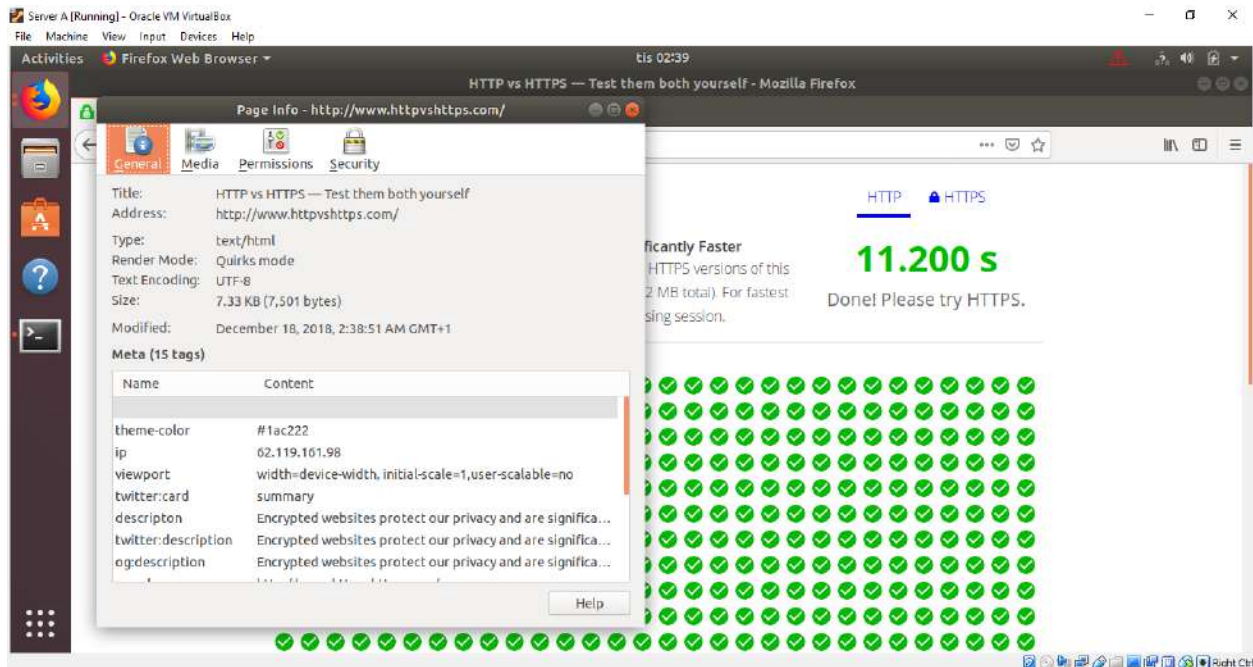


## Task 13: Unblock HTTP-browsing in the guest OS

In guest command window run I have run this rule

```
sudo iptables -D OUTPUT -p tcp --dport 80 -j REJECT
```

After that I can browse both HTTP and HTTPS, so this rule is effective.



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

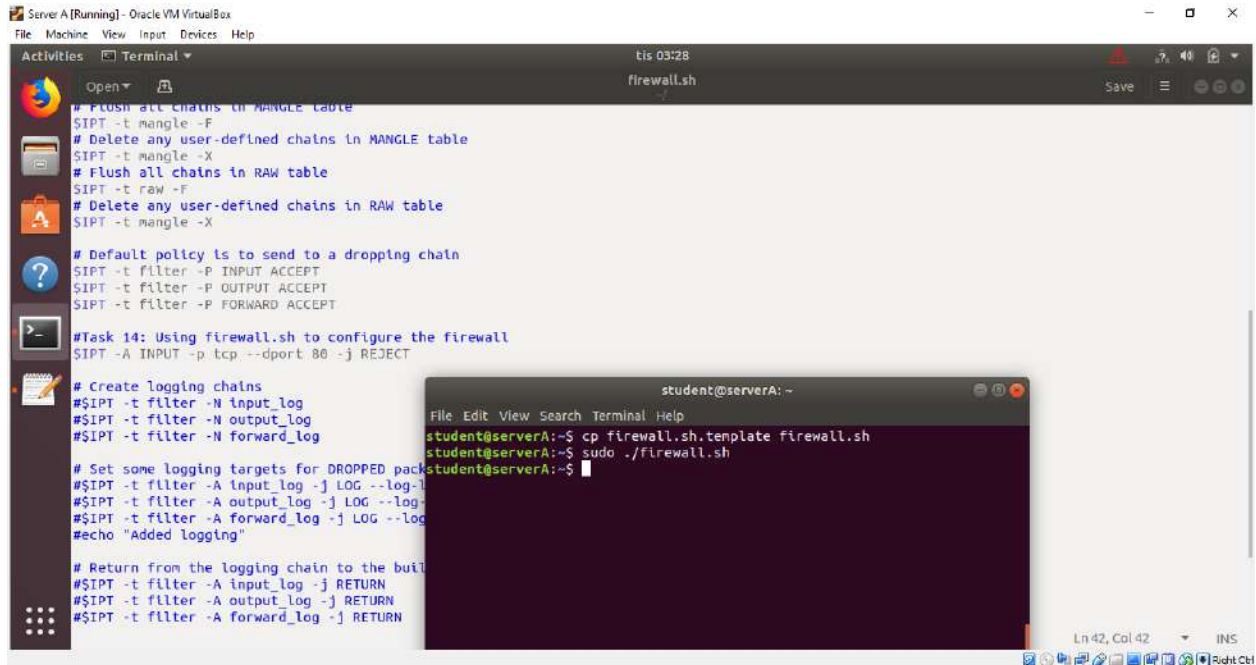
I have modified the firewall.sh script as Task 13

```
$IPT -A INPUT -p tcp --dport 80 -j REJECT
```

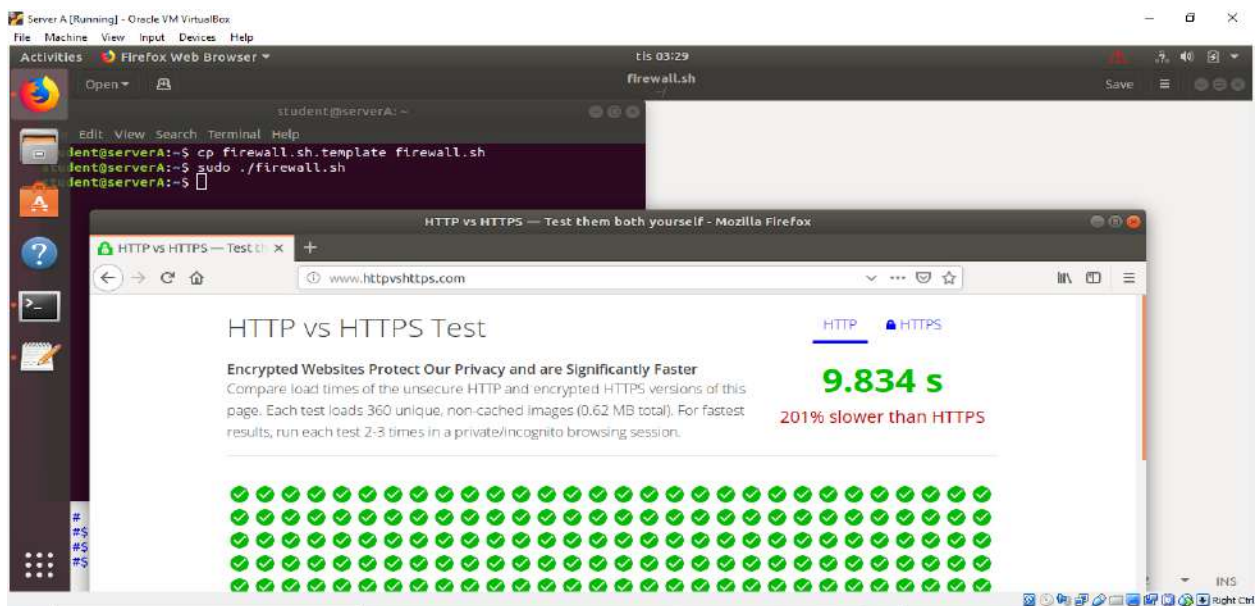
And executed the script by entering in guest command window

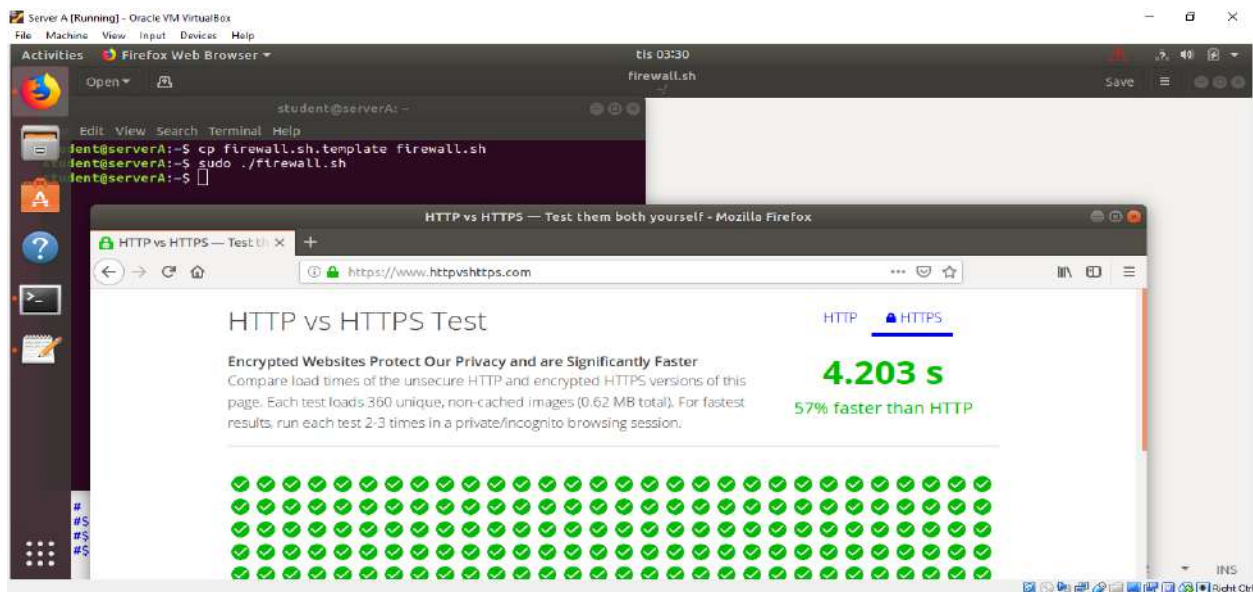
```
sudo ./firewall.sh
```

So, the guest OS can view HTTP and HTTPS pages, but apache2 server is blocked from serving HTTP content.



```
Server A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal
Open
# 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 raw -X
# Default policy is to send to a dropping chain
$IPT -t filter -P INPUT ACCEPT
$IPT -t filter -P OUTPUT ACCEPT
$IPT -t filter -P FORWARD ACCEPT
#Task 14: Using firewall.sh to configure the firewall
$IPT -A INPUT -p tcp --dport 80 -j REJECT
# Create logging chains
#$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
#$IPT -t filter -A input_log -j LOG --log-prefix "INPUT LOG"
#$IPT -t filter -A output_log -j LOG --log-prefix "OUTPUT LOG"
#$IPT -t filter -A forward_log -j LOG --log-prefix "FORWARD LOG"
#echo "Added logging"
# Return from the logging chain to the built-in chains
#$IPT -t filter -A input_log -j RETURN
#$IPT -t filter -A output_log -j RETURN
#$IPT -t filter -A forward_log -j RETURN
student@serverA: ~
File Edit View Search Terminal Help
student@serverA:~$ cp firewall.sh.template firewall.sh
student@serverA:~$ sudo ./firewall.sh
student@serverA:~$
```





## Task 15: Change default firewall policy to DROP

I have removed the rules from Task 14.

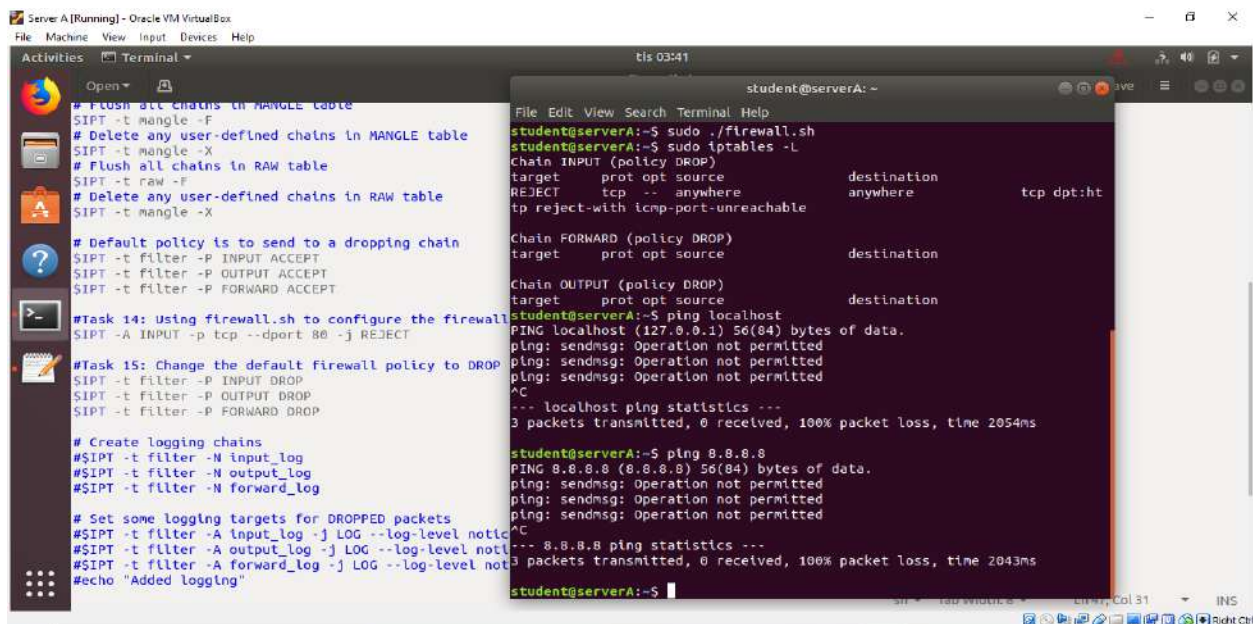
Modify the script

```
$IPT -t filter -P INPUT DROP
```

```
$IPT -t filter -P OUTPUT DROP
```

```
$IPT -t filter -P FORWARD DROP
```

After executed the rule server A cannot ping to the localhost.





## Task 16: Logging DROPPED packets

## Modify my firewall script error

Run the terminal window this code

```
sudo tail -f /var/log/kern.log
```

Now I am able to see live logs from the Linux kernel. I have started ping<sup>1</sup> the loopback interface in another terminal window and I can see all outputs are dropped

The image shows two terminal windows from a virtual machine named 'Server A [Running] - Oracle VM VirtualBox'. The left terminal window shows the output of a ping command from 'student@serverA' to 127.0.0.1, which is successful. Below this, it shows the configuration of iptables rules to log dropped packets. The right terminal window shows the output of the iptables -n -L command, displaying the rules and the corresponding kernel logs for each rule. The logs show that packets are being dropped by the default policy and the FORWARD rule.

```

# These rules must be inserted at the end of the built-in
# chain to log packets that will be dropped by the default
# DROP policy
IPT -t filter -A INPUT -j input_log
IPT -t filter -A OUTPUT -j output_log
IPT -t filter -A FORWARD -j forward_log

```

```

SRC=127.0.0.1 DST=127.0.0.1 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=17591 D
F PROTO=ICMP TYPE=8 CODE=0 ID=9496 SEQ=14
Dec 18 04:50:54 serverA kernel: [19704.198354] output drop: IN= OUT=lo
SRC=127.0.0.1 DST=127.0.0.1 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=17814 D
F PROTO=ICMP TYPE=8 CODE=0 ID=9496 SEQ=15
Dec 18 04:50:55 serverA kernel: [19705.222445] output drop: IN= OUT=lo
SRC=127.0.0.1 DST=127.0.0.1 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=17998 D
F PROTO=ICMP TYPE=8 CODE=0 ID=9496 SEQ=16
Dec 18 04:50:56 serverA kernel: [19706.246334] output drop: IN= OUT=lo
SRC=127.0.0.1 DST=127.0.0.1 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=18627 D
F PROTO=ICMP TYPE=8 CODE=0 ID=9496 SEQ=17
Dec 18 04:50:57 serverA kernel: [19707.270367] output drop: IN= OUT=lo
SRC=127.0.0.1 DST=127.0.0.1 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=18664 D
F PROTO=ICMP TYPE=8 CODE=0 ID=9496 SEQ=18
Dec 18 04:50:58 serverA kernel: [19708.294364] output drop: IN= OUT=lo
SRC=127.0.0.1 DST=127.0.0.1 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=18301 D
F PROTO=ICMP TYPE=8 CODE=0 ID=9496 SEQ=19
Dec 18 04:50:59 serverA kernel: [19709.318358] output drop: IN= OUT=lo
SRC=127.0.0.1 DST=127.0.0.1 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=18399 D
F PROTO=ICMP TYPE=8 CODE=0 ID=9496 SEQ=20

```

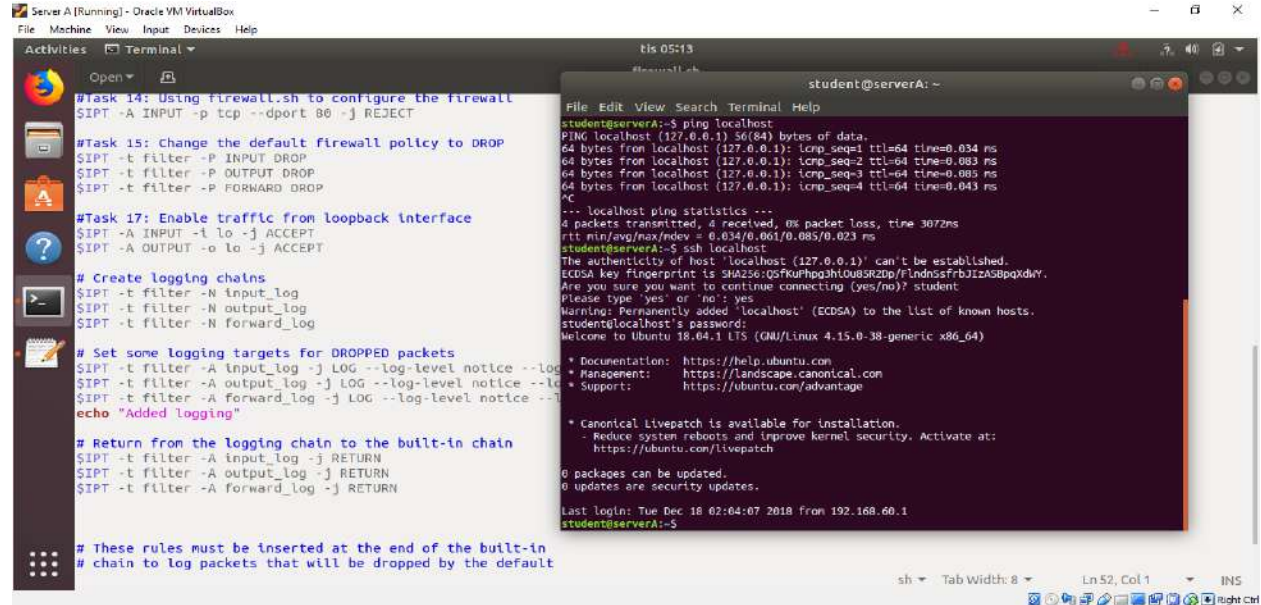
## Task 17: Enable traffic from loopback interface

Modified my firewall script as per instruction and executed it. commands

```
$IPT -A INPUT -i lo -j ACCEPT
```

```
$IPT -A OUTPUT -o lo -j ACCEPT
```

Then I can able lookback localhost and ssh localhost.



The screenshot shows a terminal window with a dark background. The left pane displays a script for Task 17, which configures iptables to allow traffic on the loopback interface (lo). The right pane shows the execution of the script, followed by a ping command to localhost, which succeeds. Below the ping, the user runs 'ssh localhost', and the terminal shows the SSH connection process, including the host key fingerprint and the user's login.

```
#Task 17: Enable traffic from loopback interface
$IPT -A INPUT -i lo -j ACCEPT
$IPT -A OUTPUT -o 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
$IPT -t filter -A input_log -j LOG --log-level notice --log-prefix "input_log"
$IPT -t filter -A output_log -j LOG --log-level notice --log-prefix "output_log"
$IPT -t filter -A forward_log -j LOG --log-level notice --log-prefix "forward_log"
echo "Added logging"

# Return from the logging chain to the built-in chain
$IPT -t filter -A input_log -j RETURN
$IPT -t filter -A output_log -j RETURN
$IPT -t filter -A forward_log -j RETURN

# These rules must be inserted at the end of the built-in
# chain to log packets that will be dropped by the default
```

```
student@serverA:~$ ping localhost
PING localhost (127.0.0.1) 56(84) bytes of data:
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.034 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.083 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.085 ms
64 bytes from localhost (127.0.0.1): icmp_seq=4 ttl=64 time=0.043 ms
^C
--- localhost ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3072ms
rtt min/avg/max/mdev = 0.034/0.061/0.085/0.023 ms
student@serverA:~$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:Q5FKuPhg3hLOU8SR2Dp/FlndssfrbJIZASBpqkdyv.
Are you sure you want to continue connecting (yes/no)? student
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

 * 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 Dec 18 02:04:07 2018 from 192.168.60.1
student@serverA:~$
```

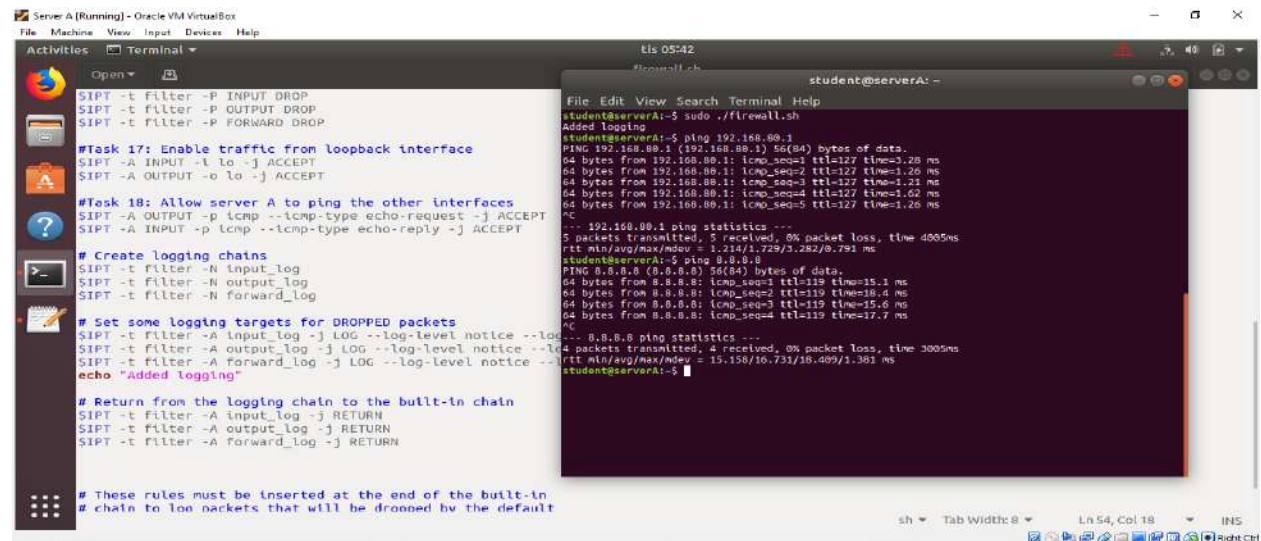
## Task 18: Allow Server A to ping the other interfaces

Modified my firewall script as per instruction and executed it. Commands-

```
$IPT -A OUTPUT -p icmp --icmp-type echo-request -j ACCEPT
```

```
$IPT -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT
```

I can manage ping from server A to the outside world.



The screenshot shows a terminal window with a dark background. The left pane displays a script for Task 18, which configures iptables to allow ICMP echo requests and replies. The right pane shows the execution of the script, followed by a ping command to 192.168.88.1, which succeeds. Below the ping, the user runs 'ping 8.8.8.8', and the terminal shows the ping results to Google's DNS servers.

```
#Task 18: Allow server A to ping the other interfaces
$IPT -A OUTPUT -p icmp --icmp-type echo-request -j ACCEPT
$IPT -A INPUT -p icmp --icmp-type echo-reply -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
$IPT -t filter -A input_log -j LOG --log-level notice --log-prefix "input_log"
$IPT -t filter -A output_log -j LOG --log-level notice --log-prefix "output_log"
$IPT -t filter -A forward_log -j LOG --log-level notice --log-prefix "forward_log"
echo "Added logging"

# Return from the logging chain to the built-in chain
$IPT -t filter -A input_log -j RETURN
$IPT -t filter -A output_log -j RETURN
$IPT -t filter -A forward_log -j RETURN

# These rules must be inserted at the end of the built-in
# chain to log packets that will be dropped by the default
```

```
student@serverA:~$ sudo ./firewall.sh
Added logging
student@serverA:~$ ping 192.168.88.1
PING 192.168.88.1 (192.168.88.1) 56(84) bytes of data:
64 bytes from 192.168.88.1: icmp_seq=1 ttl=127 time=0.28 ms
64 bytes from 192.168.88.1: icmp_seq=2 ttl=127 time=1.26 ms
64 bytes from 192.168.88.1: icmp_seq=3 ttl=127 time=1.21 ms
64 bytes from 192.168.88.1: icmp_seq=4 ttl=127 time=1.62 ms
64 bytes from 192.168.88.1: icmp_seq=5 ttl=127 time=1.26 ms
^C
--- 192.168.88.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 1.214/1.729/3.282/0.791 ms
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=119 time=15.1 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=119 time=16.4 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=119 time=15.6 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=119 time=17.7 ms
^C
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 15.158/16.731/18.409/1.381 ms
student@serverA:~$
```

### Task 19: Allow Server A to ping all hosts

Modified my firewall script as per instruction and executed it. Commands

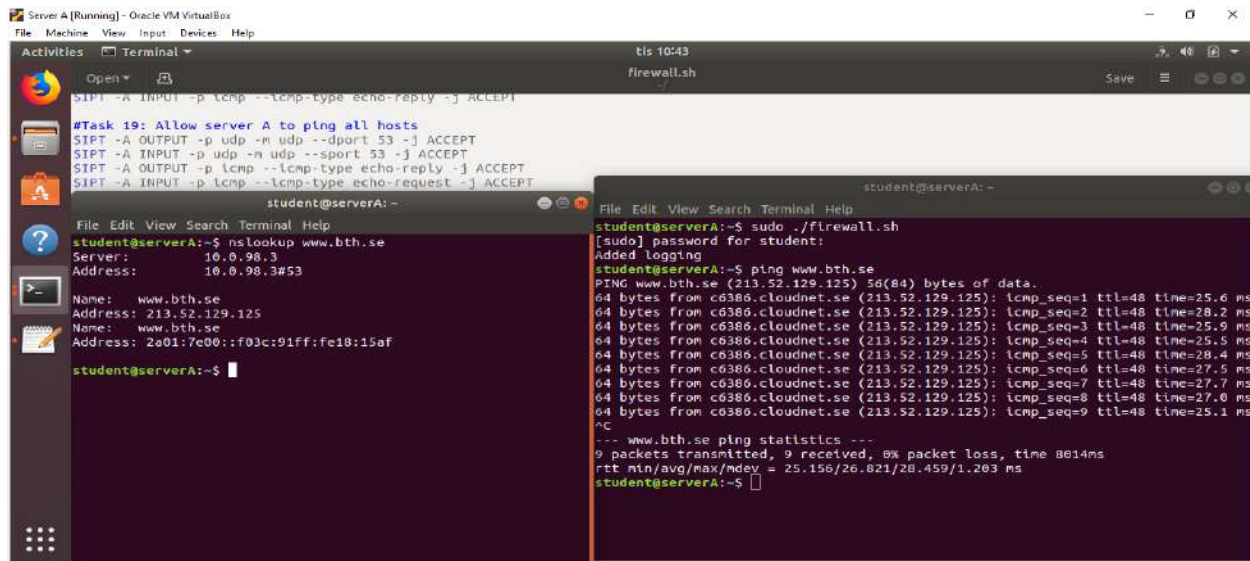
```
$IPT -A OUTPUT -p udp -m udp --dport 53 -j ACCEPT
```

```
$IPT -A INPUT -p udp -m udp --sport 53 -j ACCEPT
```

```
$IPT -A OUTPUT -p icmp --icmp-type echo-reply -j ACCEPT
```

```
$IPT -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
```

Now server A can ping all host

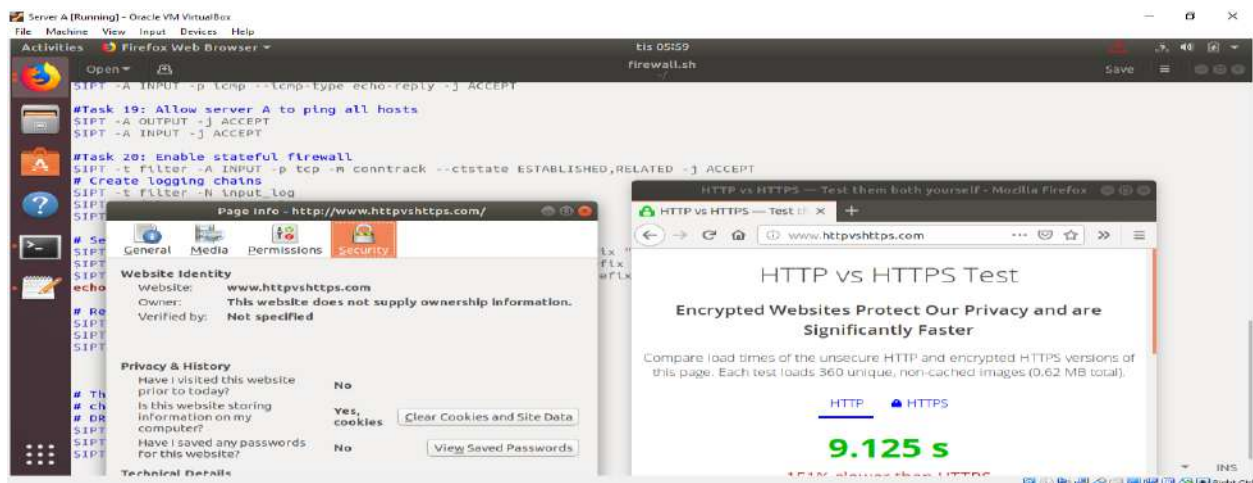


## Task 20: Enable stateful firewall

Modified my firewall script as per instruction and executed it. By commands-

```
$IPT -t filter -A INPUT -p tcp -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
```

After that manage to browse both http and https.





## Task 21: Enable SSH and HTTPS content from apache2 server for web browser on host

Modified my firewall script as per instruction and executed it. Commands-

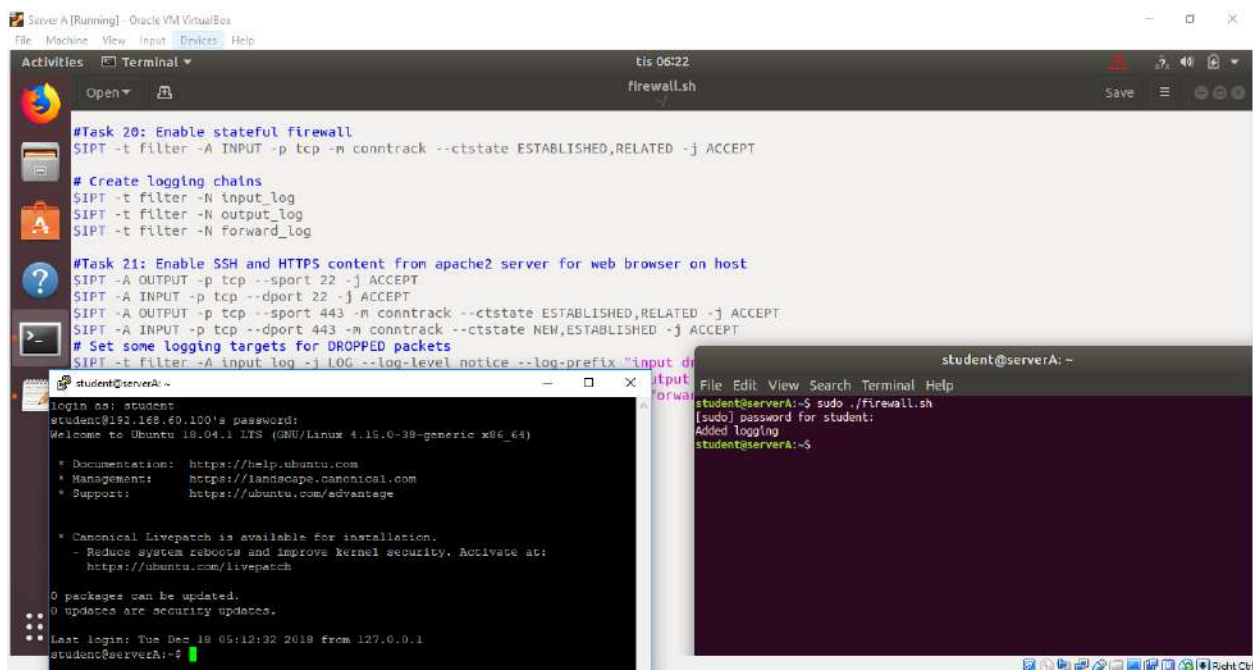
```
$IPT -A OUTPUT -p tcp --sport 22 -j ACCEPT
```

```
$IPT -A INPUT -p tcp --dport 22 -j ACCEPT
```

```
$IPT -A OUTPUT -p tcp --sport 443 -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
```

```
$IPT -A INPUT -p tcp --dport 443 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
```

After that enable SSH and HTTPS content from apache2 server.



The screenshot shows a terminal window titled "Server A [Running] - Oracle VM VirtualBox". The terminal displays the following commands and output:

```
#Task 20: Enable stateful firewall
$IPT -t filter -A INPUT -p tcp -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

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

#Task 21: Enable SSH and HTTPS content from apache2 server for web browser on host
$IPT -A OUTPUT -p tcp --sport 22 -j ACCEPT
$IPT -A INPUT -p tcp --dport 22 -j ACCEPT
$IPT -A OUTPUT -p tcp --sport 443 -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
$IPT -A INPUT -p tcp --dport 443 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT

# Set some logging targets for DROPPED packets
$IPT -t filter -A input_log -j LOG --log-level notice --log-prefix "input dropped"
$IPT -t filter -A output_log -j LOG --log-level notice --log-prefix "output dropped"
$IPT -t filter -A forward_log -j LOG --log-level notice --log-prefix "forwarded"
```

Below the commands, there is a login prompt for a user named "student" on a server with IP 192.168.60.100. The user enters their password and is greeted with the Ubuntu 18.04.1 LTS login message. The terminal also shows system updates and login history.

```
login as: student
student@192.168.60.100's password:
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-38-generic x86_64)

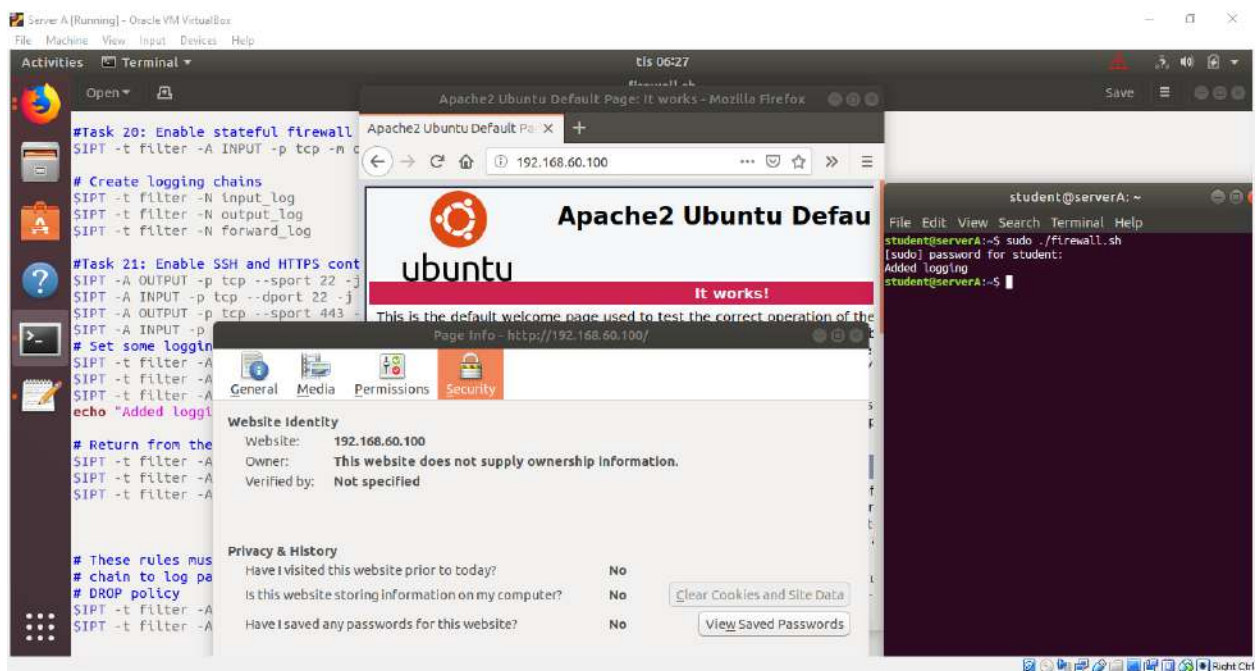
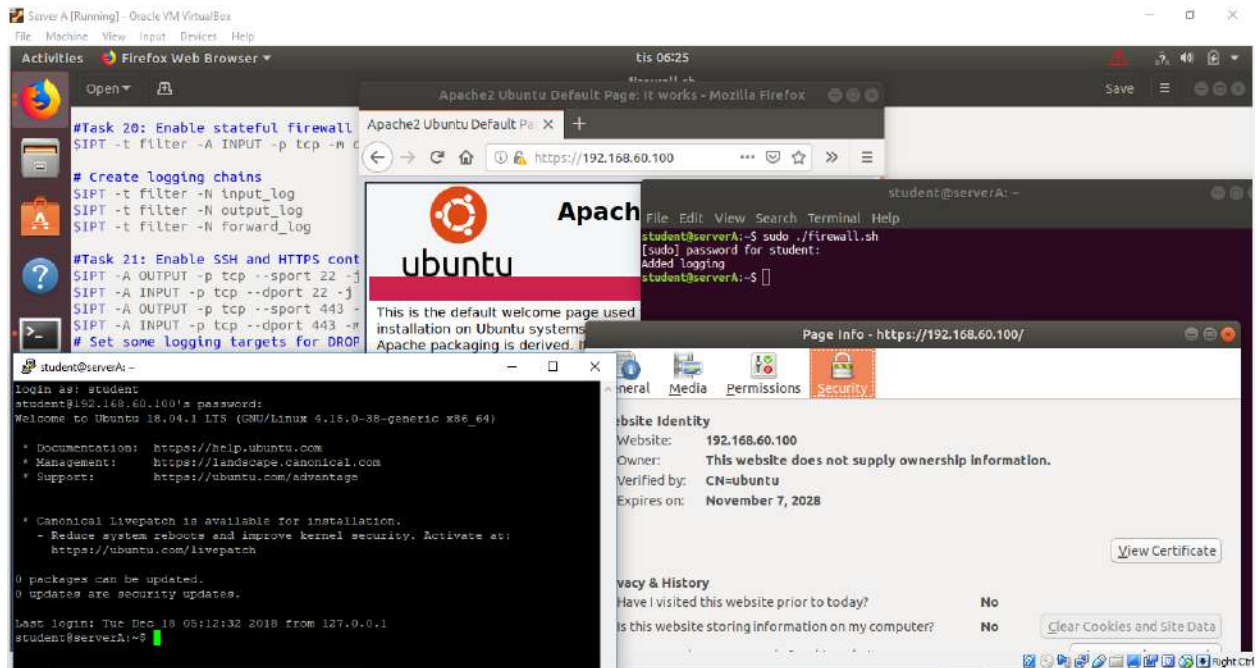
 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 * 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 Dec 18 05:12:32 2018 from 127.0.0.1
student@serverA:~$
```

On the right side of the screenshot, there is a smaller terminal window titled "student@serverA: ~". It shows the user running the command `sudo ./firewall.sh` and receiving the output "Added logging".



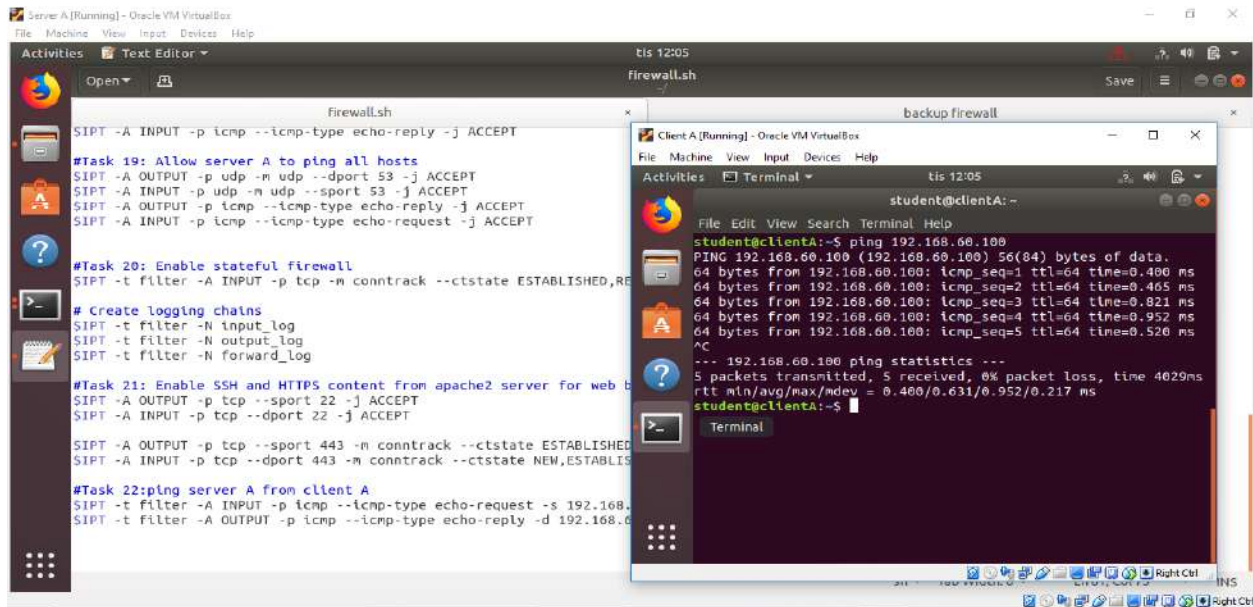
## Task 22: Ping Server A from Client A

Modified my firewall script as per instruction and executed it. Commands-

```
$IPT -t filter -A INPUT -p icmp --icmp-type echo-request -s 192.168.60.111 -m conntrack --ctstate ESTABLISHED,RELATED,NEW -j ACCEPT
```

```
$IPT -t filter -A OUTPUT -p icmp --icmp-type echo-reply -d 192.168.60.111 -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
```

After executing the firewall it's successfully done ping from client A



```
Server A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Text Editor
Open
firewall.sh
#Task 19: Allow server A to ping all hosts
$IPT -A OUTPUT -p udp -m udp --dport 53 -j ACCEPT
$IPT -A INPUT -p udp -m udp --sport 53 -j ACCEPT
$IPT -A OUTPUT -p icmp --icmp-type echo-reply -j ACCEPT
$IPT -A INPUT -p icmp --icmp-type echo-request -j ACCEPT

#Task 20: Enable stateful firewall
$IPT -t filter -A INPUT -p tcp -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

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

#Task 21: Enable SSH and HTTPS content from apache2 server for web browser
$IPT -A OUTPUT -p tcp --sport 22 -j ACCEPT
$IPT -A INPUT -p tcp --dport 22 -j ACCEPT

$IPT -A OUTPUT -p tcp --sport 443 -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
$IPT -A INPUT -p tcp --dport 443 -m conntrack --ctstate NEW,ESTABLISHED,RELATED -j ACCEPT

#Task 22: ping server A from client A
$IPT -t filter -A INPUT -p icmp --icmp-type echo-request -s 192.168.60.111 -m conntrack --ctstate ESTABLISHED,RELATED,NEW -j ACCEPT
$IPT -t filter -A OUTPUT -p icmp --icmp-type echo-reply -d 192.168.60.111 -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

Client A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal
student@clientA:~
student@clientA:~$ 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=0.400 ms
64 bytes from 192.168.60.100: icmp_seq=2 ttl=64 time=0.465 ms
64 bytes from 192.168.60.100: icmp_seq=3 ttl=64 time=0.821 ms
64 bytes from 192.168.60.100: icmp_seq=4 ttl=64 time=0.952 ms
64 bytes from 192.168.60.100: icmp_seq=5 ttl=64 time=0.526 ms
^C
--- 192.168.60.100 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4029ms
rtt min/avg/max/mdev = 0.400/0.631/0.952/0.217 ms
student@clientA:~$
```

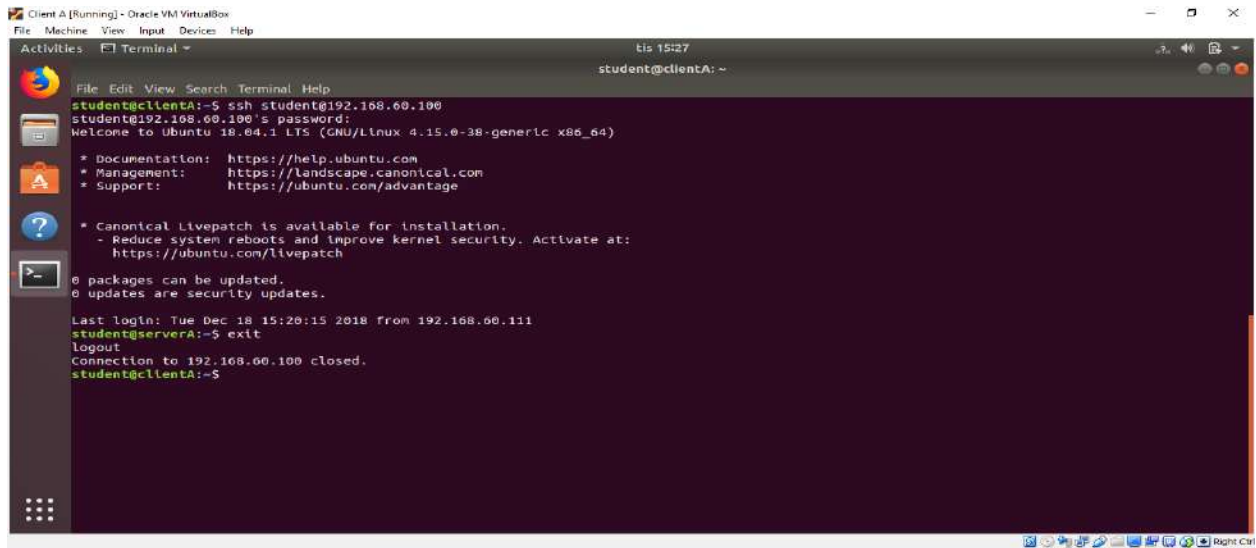


## Task 23: SSH from Client A to Server A

By using this line

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

SSH session to be established from Client A to Server A verify the password allow to access after that exit from server A.



The screenshot shows a terminal window titled "Client A [Running] - Oracle VM VirtualBox". The terminal displays the following text:

```
student@clientA:~$ ssh student@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

 * 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 Dec 18 15:20:15 2018 from 192.168.60.111
student@serverA:~$ exit
logout
Connection to 192.168.60.100 closed.
student@clientA:~$
```

## Task 24: Add gateway and DNS server to Client A

Modified the file `/etc/resolv.conf` on Client A and verify the 10.0.98.3 is listed as DNS server

`nameserver 10.0.98.3`

`gateway 192.168.60.100`



The screenshot shows a terminal window titled "Client A [Running] - Oracle VM VirtualBox". The terminal is running the nano text editor on the file `/etc/resolv.conf`. The prompt is `student@clientA: ~`. The file content is:

```
nameserver 10.0.98.3
```

The terminal window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The status bar at the bottom indicates "File '/etc/resolv.conf' is unwritable".



The screenshot shows a text editor window titled "Client A [Running] - Oracle VM VirtualBox". The text editor is editing the file `/etc/network/interfaces`. The prompt is `student@clientA: ~`. The file content is:

```
# 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
```

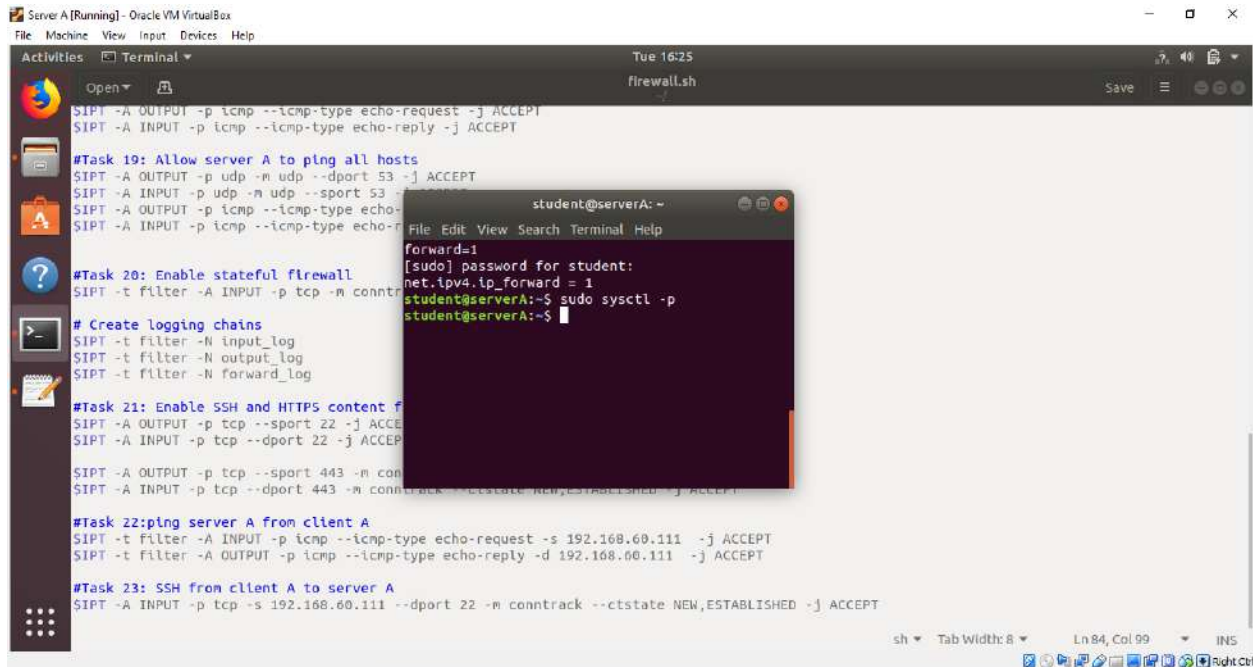
The text editor window has a menu bar with "File", "Machine", "View", "Input", "Devices", and "Help". The status bar at the bottom indicates "Plain Text", "Tab Width: 8", "Ln 1, Col 1", and "INS".

## Task 25: Enable IP forwarding on Server A

For forwarding in server A executed the command in the terminal.

```
sudo sysctl -w net.ipv4.ip_forward=1
```

```
sudo sysctl -p
```



The screenshot shows a terminal window titled "Server A [Running] - Oracle VM VirtualBox". The terminal displays a series of firewall rules and system configuration commands. The rules are for ICMP, UDP, and TCP traffic. The system configuration commands are for enabling IP forwarding and setting the sysctl file. The terminal output shows the following commands and their results:

```
SIPT -A OUTPUT -p icmp --icmp-type echo-request -j ACCEPT
SIPT -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT

#Task 19: Allow server A to ping all hosts
SIPT -A OUTPUT -p udp -m udp --dport 53 -j ACCEPT
SIPT -A INPUT -p udp -m udp --sport 53 -j ACCEPT
SIPT -A OUTPUT -p icmp --icmp-type echo-request -j ACCEPT
SIPT -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT

#Task 20: Enable stateful firewall
SIPT -t filter -A INPUT -p tcp -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT

# Create logging chains
SIPT -t filter -N input_log
SIPT -t filter -N output_log
SIPT -t filter -N forward_log

#Task 21: Enable SSH and HTTPS content filtering
SIPT -A OUTPUT -p tcp --sport 22 -j ACCEPT
SIPT -A INPUT -p tcp --dport 22 -j ACCEPT

SIPT -A OUTPUT -p tcp --sport 443 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
SIPT -A INPUT -p tcp --dport 443 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT

#Task 22: ping server A from client A
SIPT -t filter -A INPUT -p icmp --icmp-type echo-request -s 192.168.60.111 -j ACCEPT
SIPT -t filter -A OUTPUT -p icmp --icmp-type echo-reply -d 192.168.60.111 -j ACCEPT

#Task 23: SSH from client A to server A
SIPT -A INPUT -p tcp -s 192.168.60.111 --dport 22 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
```

The terminal also shows the following commands and their results:

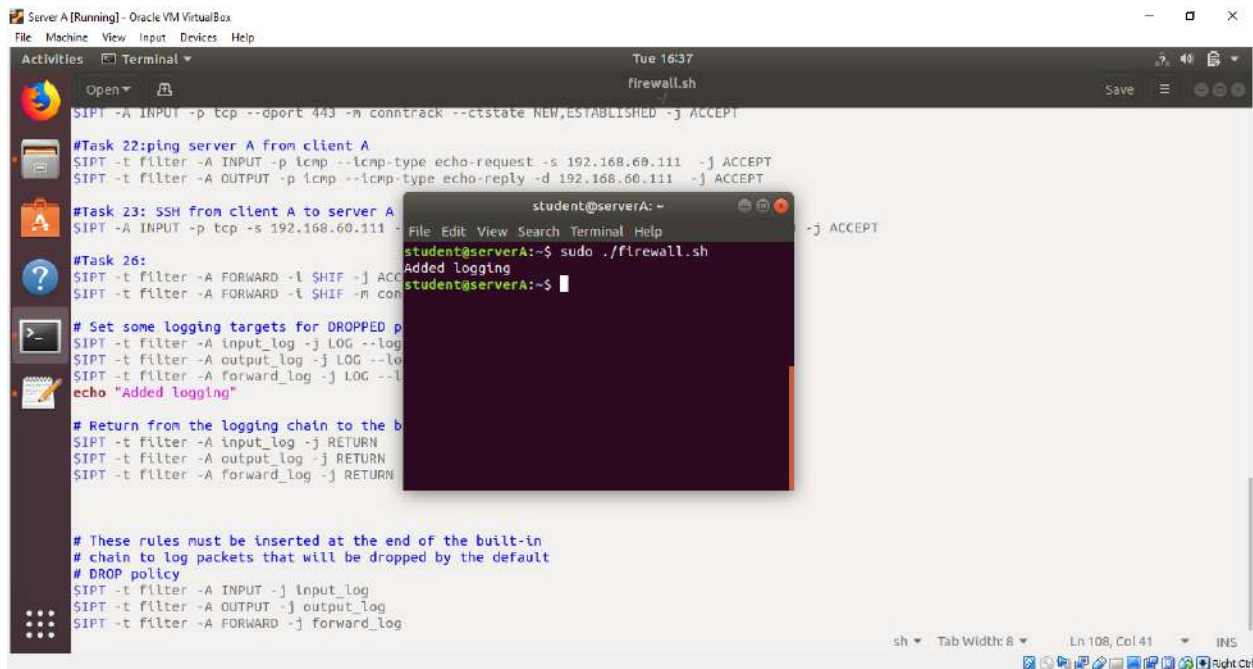
```
forward=1
[sudo] password for student:
net.ipv4.ip_forward = 1
student@serverA:~$ sudo sysctl -p
student@serverA:~$
```

## Task 26: Change iptables to forward packets

I have changed the rules for iptables to forward packets

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

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



The screenshot shows a terminal window titled "Server A [Running] - Oracle VM VirtualBox" with a menu bar (File, Machine, View, Input, Devices, Help) and a toolbar (Activities, Open, Save, etc.). The terminal displays the following iptables rules:

```
S IPT -A INPUT -p tcp --dport 443 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
#Task 22: ping server A from client A
S IPT -t filter -A INPUT -p icmp --icmp-type echo-request -s 192.168.60.111 -j ACCEPT
S IPT -t filter -A OUTPUT -p icmp --icmp-type echo-reply -d 192.168.60.111 -j ACCEPT
#Task 23: SSH from client A to server A
S IPT -A INPUT -p tcp -s 192.168.60.111 -j ACCEPT
#Task 26:
S IPT -t filter -A FORWARD -i $HIF -j ACCEPT
S IPT -t filter -A FORWARD -i $NIF -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
# Set some logging targets for DROPPED packets
S IPT -t filter -A input_log -j LOG --log-prefix "DROPPED INPUT "
S IPT -t filter -A output_log -j LOG --log-prefix "DROPPED OUTPUT "
S IPT -t filter -A forward_log -j LOG --log-prefix "DROPPED FORWARD "
echo "Added logging"
# Return from the logging chain to the built-in chain
S IPT -t filter -A input_log -j RETURN
S IPT -t filter -A output_log -j RETURN
S IPT -t filter -A forward_log -j RETURN
# These rules must be inserted at the end of the built-in chain
# chain to log packets that will be dropped by the default DROP policy
S IPT -t filter -A INPUT -j input_log
S IPT -t filter -A OUTPUT -j output_log
S IPT -t filter -A FORWARD -j forward_log
```

A secondary terminal window titled "student@serverA: ~" is overlaid on the main terminal. It shows the execution of a script:

```
student@serverA: ~$ sudo ./firewall.sh
Added logging
student@serverA: ~$
```

## Task 27: Enable SNAT on Server A

After finished the forwarding rules then I edit my firewall.sh script

```
$IPT -t nat -A POSTROUTING -j SNAT -o $NIF --to $NIP
```

After executed the rules in server A then from client A. I can have managed to use the internet.

```
Server A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Text Editor
Tue 18:33
firewall.sh
Save

#Task 20: Enable stateful firewall
SIPT -t filter -A INPUT -p tcp -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
SIPT -A INPUT -p icmp --icmp-type echo-request -j ACCEPT

#Task 21: Enable SSH and HTTPS content from apache2 server
SIPT -A OUTPUT -p tcp --sport 22 -j ACCEPT
SIPT -A INPUT -p tcp --dport 22 -j ACCEPT

#Task 22: ping server A from client A
SIPT -t filter -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
SIPT -t filter -A OUTPUT -p icmp --icmp-type echo-reply -j ACCEPT

#Task 23: SSH from client A to server A
SIPT -A INPUT -p tcp -s 192.168.60.111 --dport 22 -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

#Task 26:
SIPT -t filter -A FORWARD -i $NIF -j ACCEPT
SIPT -t filter -A FORWARD -o $NIF -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

#Task 27:
SIPT -t nat -A POSTROUTING -j SNAT -o $NIF --to $NIP

# Create logging chains
SIPT -t filter -N input_log
SIPT -t filter -N output_log
```

```
Client A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal
Tue 18:33
student@clientA: ~
student@clientA:~$ ping www.google.com
PING www.google.com (172.217.20.100) 56(84) bytes of data:
64 bytes from fra02s28-in-f4.1e100.net (172.217.20.100): icmp_
seq=1 ttl=50 time=18.2 ms
64 bytes from fra02s28-in-f4.1e100.net (172.217.20.100): icmp_
seq=2 ttl=50 time=18.5 ms
64 bytes from fra02s28-in-f4.1e100.net (172.217.20.100): icmp_
seq=3 ttl=50 time=19.0 ms
64 bytes from fra02s28-in-f4.1e100.net (172.217.20.100): icmp_
seq=4 ttl=50 time=20.2 ms
^C
--- www.google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 18.226/19.009/20.287/0.797 ms
student@clientA:~$
```

Reference:

- ❖ Lab 1: Linux networking and firewalls