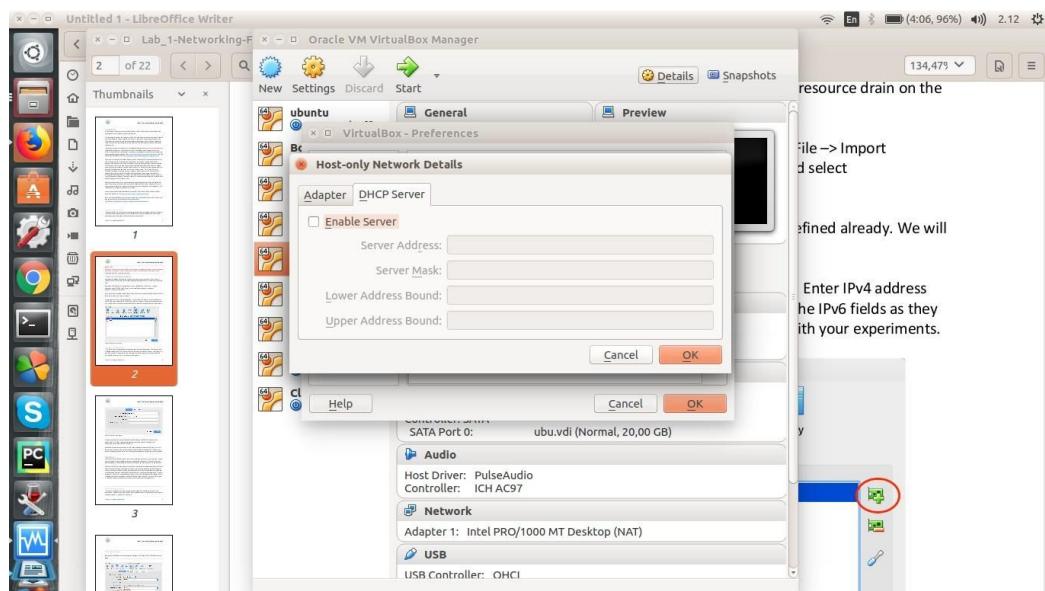
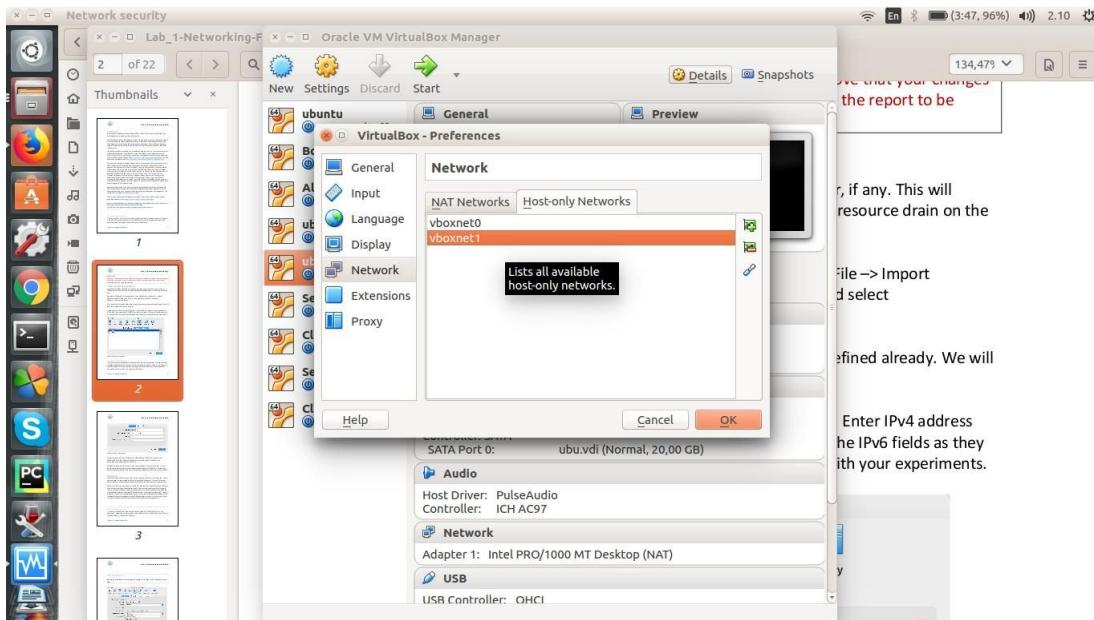
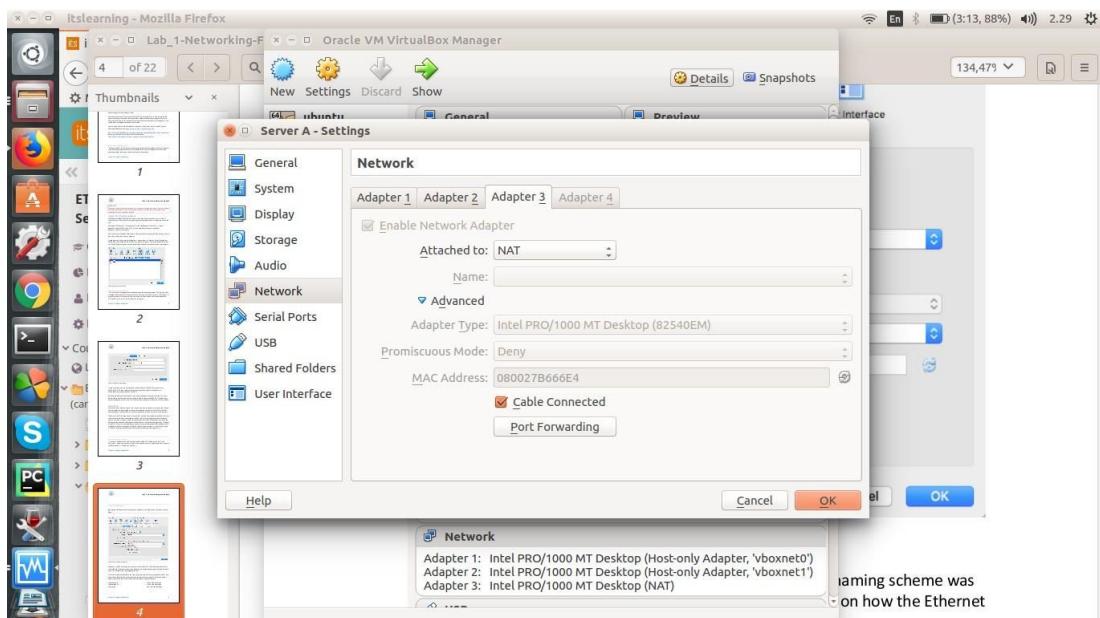
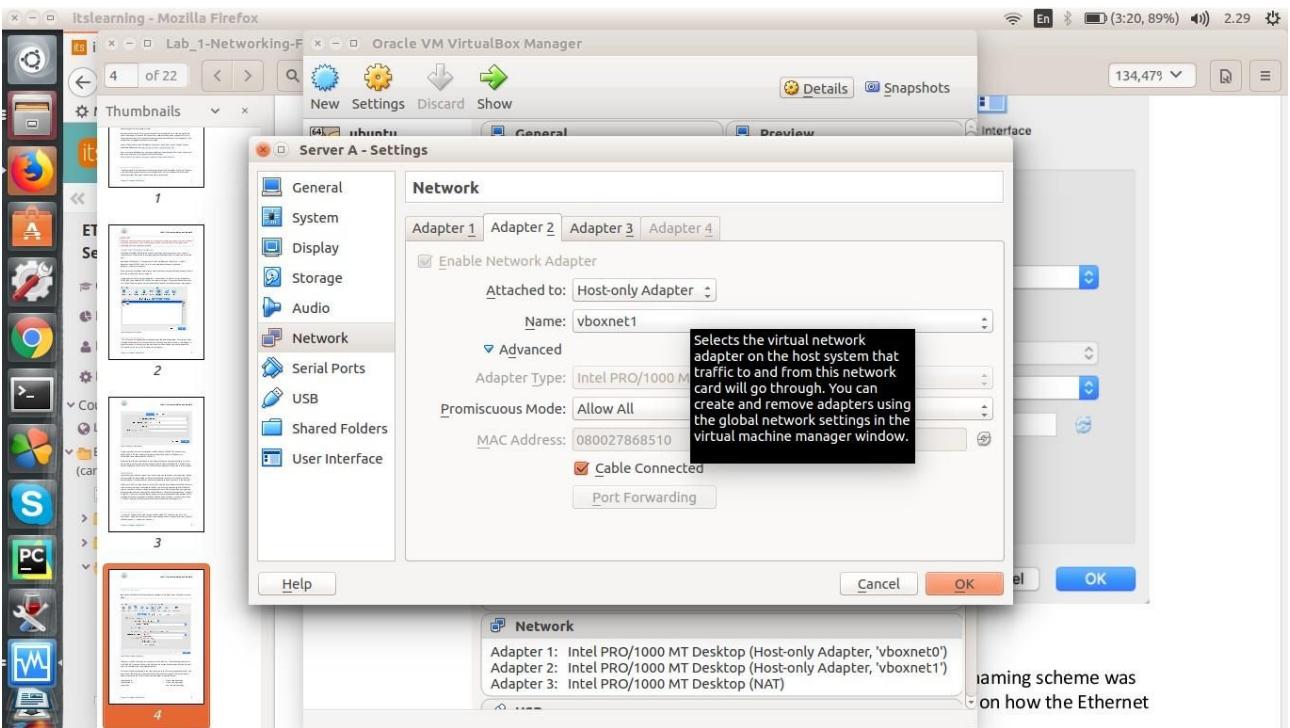


LAB 1.1
(security)
Saddam Hossen
9112301917

Task 1: MAC addresses:

I have created 2 Host only adapter and one NAT adapter.



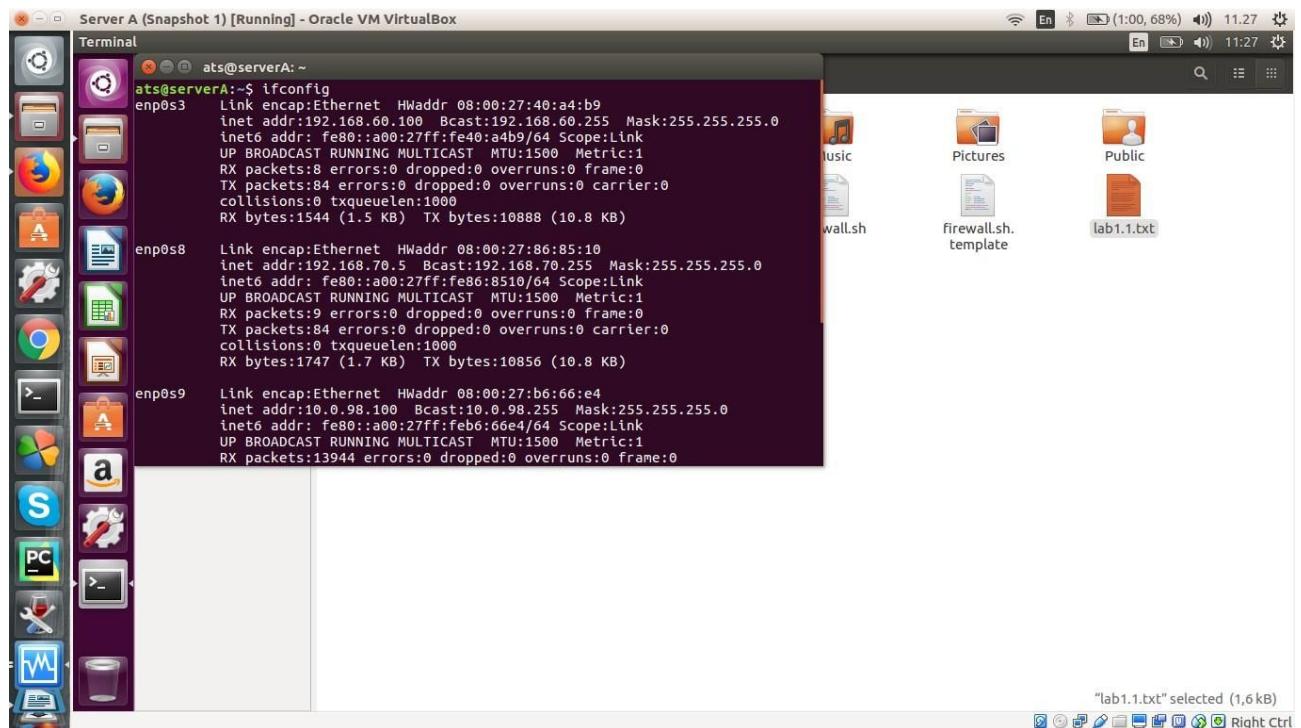


Interface:-----MAC Addresses

vboxnet0 -----08002740A4B9
vboxnet1-----080027868510
NAT-----080027B666E4

Task 2: Network interfaces

iface	ip	mac
enp0s3	192.168.60.100	08002740A4B9---Host Only Adapter
enp0s8	192.168.70.5	080027868510---Host Only Adapter
enp0s9	10.0.98.100	080027B666E4-- NAT Adapter



Task 3: IP addresses, netmasks and subnet

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

Network address: 10.0.98.0

```
dns-nameservers 10.0.98.3
gateway 10.0.98.2
```

```
# Host-only interface (for internal use) auto enp0s3
iface enp0s3 inet static address
192.168.60.100
netmask 255.255.255.0
```

Network address: 10.0.60.0

Host-only interface (for external use) auto enp0s8
iface enp0s8 inet static address
192.168.70.5
netmask 255.255.255.0

Network address: 10.0.70.0

enp0s8

inet addr: 192.168.60.0 → 11000000.10101600.0011100.0110000
Subnet Mask: 255.255.255.0 → 11111111.11111111.11111111.00000000
bitwise AND: 192.168.60.0 → 11000000.10101000.0011100.00000000

Network Address: 192.168.60.0

enp0s8:

inetaddr: 192.168.70.5 → 11000000.1010100001000110.00000101
Network: 255.255.255.0 → 11111111.11111111.11111111.00000000
bitwise AND: 192.168.70.0 → 11000000.10101000.01000110.00000000

enp0s9:

inet Addr: 10.0.98.100 → 00001010.00000000.01100010.01100100
Net Mask: 255.255.255.0 → 11111111.11111111.11111111.00000000
Network Address: 10.0.98.0

do: inet Address: 127.0.0.1 → 0111111.00000000.00000000.00000000
Netmask: 255.255.255.0 → 1111111.1111111.1111111.1111111

Bitwise AND: 127.0.0.1 → 0111111.00000000.00000000.00000000

Network Address: 127.0.0.1



BÄSTA STARTEN PÅ DIN KARRIÄR I SKOLA OCH FÖRSKOLA-

NACKAS KOMMUNALA SKOLOR

Få tips och inspirationsmaterial



Task 4: Host-only interfaces

iface:	ip:	mask::
enp0s3-----	192.168.60.100	255.255.255.0
enp0s8-----	192.168.70.5	255.255.255.0
enp0s9-----	10.0.98.100	255.0.0.0

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	10.0.98.2	0.0.0.0	UG	0	0	0	enp0s9
10.0.98.0	0.0.0.0	255.255.255.0	U	0	0	0	enp0s9
169.254.0.0	0.0.0.0	255.255.0.0	U	1000	0	0	enp0s3
192.168.60.0	0.0.0.0	255.255.255.0	U	0	0	0	enp0s3
192.168.70.0	0.0.0.0	255.255.255.0	U	0	0	0	enp0s8

```
ats@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
```

Host-only interface	-----	-----IPv4 Address-----	-----Subnet Mask-----
vboxnet0		192.168.60.1	255.255.255.0
vboxnet1		192.168.70.1	255.255.255.0
vboxnet2		192.168.80.1	255.255.255.0

By the command ipconfig/all in the host OS, I can get interface list from host OS. The available host-only interfaces are vboxnet0, vboxnet1 and vboxnet2

Through enp0s9 I can reach the default gateway because enp0s9 is the default gateway interface.

Task 5: Routing tables in the host OS

By using netstat -4

-rn route -n

ip -4 route

on host os I got this...

```
saddam@saddam-bth:~$ ip -4 route
default via 192.168.0.1 dev wlp2s0 proto static metric 600
169.254.0.0/16 dev wlp2s0 scope link metric 1000
192.168.0.0/24 dev wlp2s0 proto kernel scope link src 192.168.0.7 metric 600
192.168.56.0/24 dev vboxnet0 proto kernel scope link src 192.168.56.1
192.168.57.0/24 dev vboxnet1 proto kernel scope link src 192.168.57.1
192.168.58.0/24 dev vboxnet2 proto kernel scope link src 192.168.58.1
192.168.122.0/24 dev virbr0 proto kernel scope link src 192.168.122.1 linkdown

saddam@saddam-bth:~$
```

Default gateway to connect the internet is wlp2s0

(wifi). For interface wlp2s0 I can reach by the

default gateway 192.168.0.1

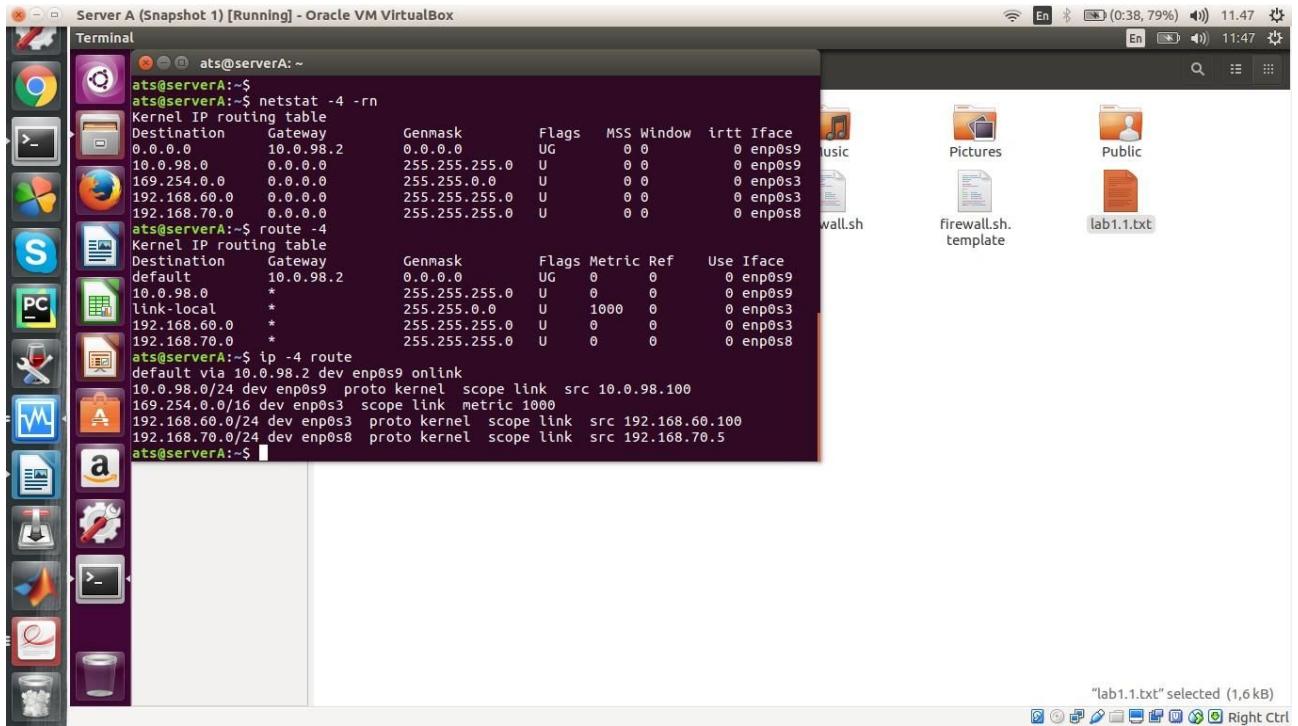
Task 6: Routing tables in the guest OS

By using netstat -4

-rn route -n

ip -4 route

on guest os I got this



I have reached the default gateway 10.0.98.2 by interface enp0s9.

enp0s9 is NAT interface.

Task 7: Ping the host-based host-onlyinterface

Ping accordingly and got below icmp request and reply both Host and Guest OS.

At guest OS, ping the IP address 192.168.60.1 which is corresponding of the host only interface in the host OS. Then I have stopped the ping and taken Wireshark screenshot. From the screenshot .I can observe that the ICMP traffic of the Wireshark are identical.

At Host OS

```
student@serverA:~$ ping 192.168.0.1
PING 192.168.0.1 (192.168.0.1) 56(84) bytes of data.
4 bytes from 192.168.0.1: icmp_seq=1 ttl=63 time=3.61 ms
4 bytes from 192.168.0.1: icmp_seq=2 ttl=63 time=7.49 ms
4 bytes from 192.168.0.1: icmp_seq=3 ttl=63 time=3.44 ms
4 bytes from 192.168.0.1: icmp_seq=4 ttl=63 time=6.52 ms
4 bytes from 192.168.0.1: icmp_seq=5 ttl=63 time=7.08 ms
4 bytes from 192.168.0.1: icmp_seq=6 ttl=63 time=4.51 ms
4 bytes from 192.168.0.1: icmp_seq=7 ttl=63 time=4.27 ms
4 bytes from 192.168.0.1: icmp_seq=8 ttl=63 time=7.10 ms
4 bytes from 192.168.0.1: icmp_seq=9 ttl=63 time=4.58 ms
4 bytes from 192.168.0.1: icmp_seq=10 ttl=63 time=5.67 ms
4 bytes from 192.168.0.1: icmp_seq=11 ttl=63 time=4.12 ms
4 bytes from 192.168.0.1: icmp_seq=12 ttl=63 time=8.31 ms
4 bytes from 192.168.0.1: icmp_seq=13 ttl=63 time=3.21 ms
4 bytes from 192.168.0.1: icmp_seq=14 ttl=63 time=3.79 ms
4 bytes from 192.168.0.1: icmp_seq=15 ttl=63 time=7.42 ms
4 bytes from 192.168.0.1: icmp_seq=16 ttl=63 time=3.28 ms
```

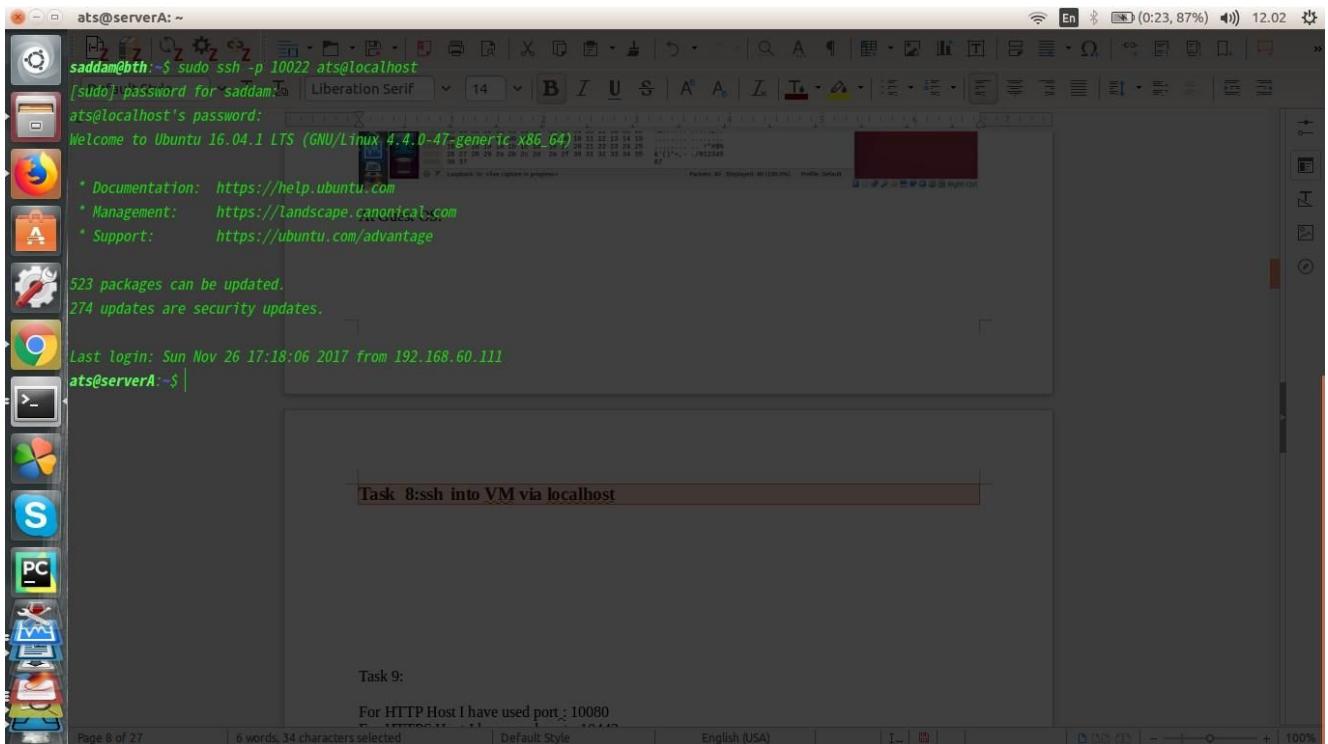
*enp0s9

No.	Time	Source	Destination	Protocol	Length	Info
24	10.024724502	192.168.0.1	10.0.98.100	ICMP	98	Echo (ping)
25	11.022507888	10.0.98.100	192.168.0.1	ICMP	98	Echo (ping)
26	11.030776033	192.168.0.1	10.0.98.100	ICMP	98	Echo (ping)
27	12.024103300	10.0.98.100	192.168.0.1	ICMP	98	Echo (ping)
28	12.027300303	192.168.0.1	10.0.98.100	ICMP	98	Echo (ping)
29	13.025547296	10.0.98.100	192.168.0.1	ICMP	98	Echo (ping)
30	13.029328703	192.168.0.1	10.0.98.100	ICMP	98	Echo (ping)

Task 8: ssh into VM via localhost

I have opened a remote shell to Server A and I can run any console commands in it log into serverA from host OS.

sudo ssh -p 10022 ats@localhost

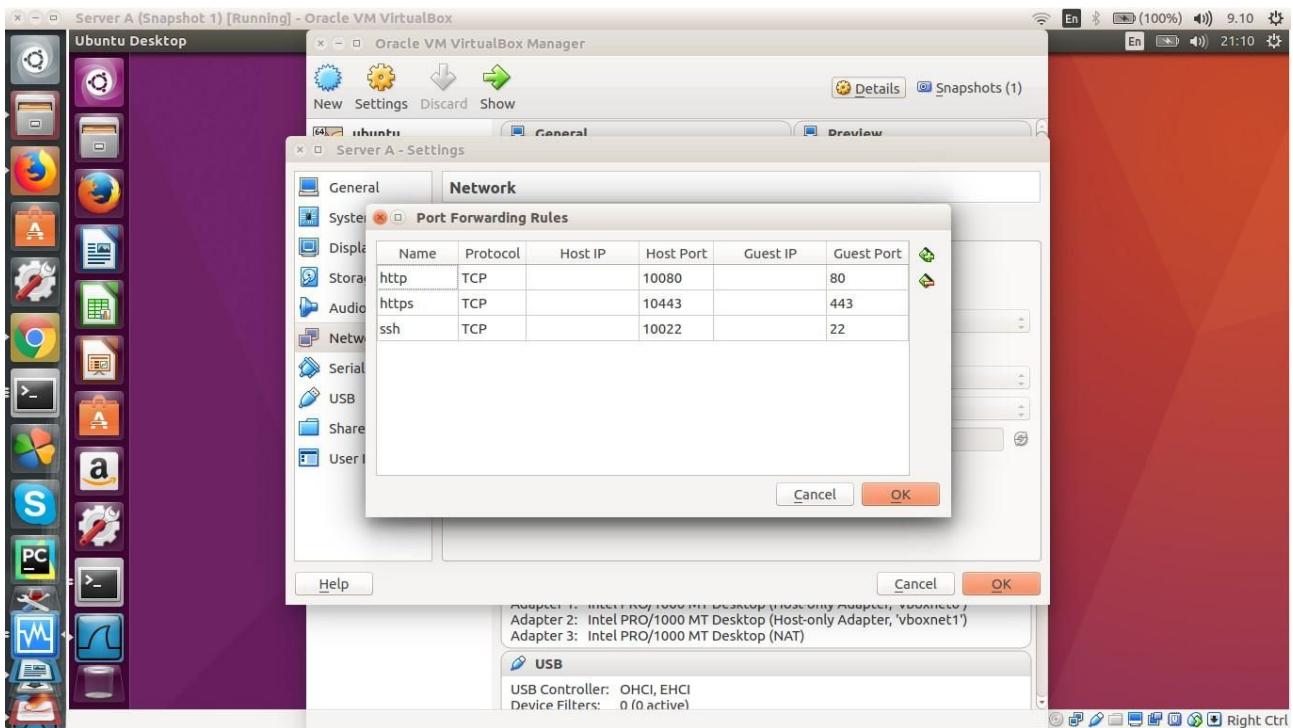


Task 9:

For HTTP Host I have used port :
10080 For HTTPS Host I have used
port : 10443

For HTTP Guest I have used port :
80 For HTTPS Guest I have used port
: 443

Here host port number 10080 for HTTP and 10443 for HTTPS



X

Apache2 Ubuntu Default Page: It works - Mozilla Firefox

192.168.60.1:10080

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

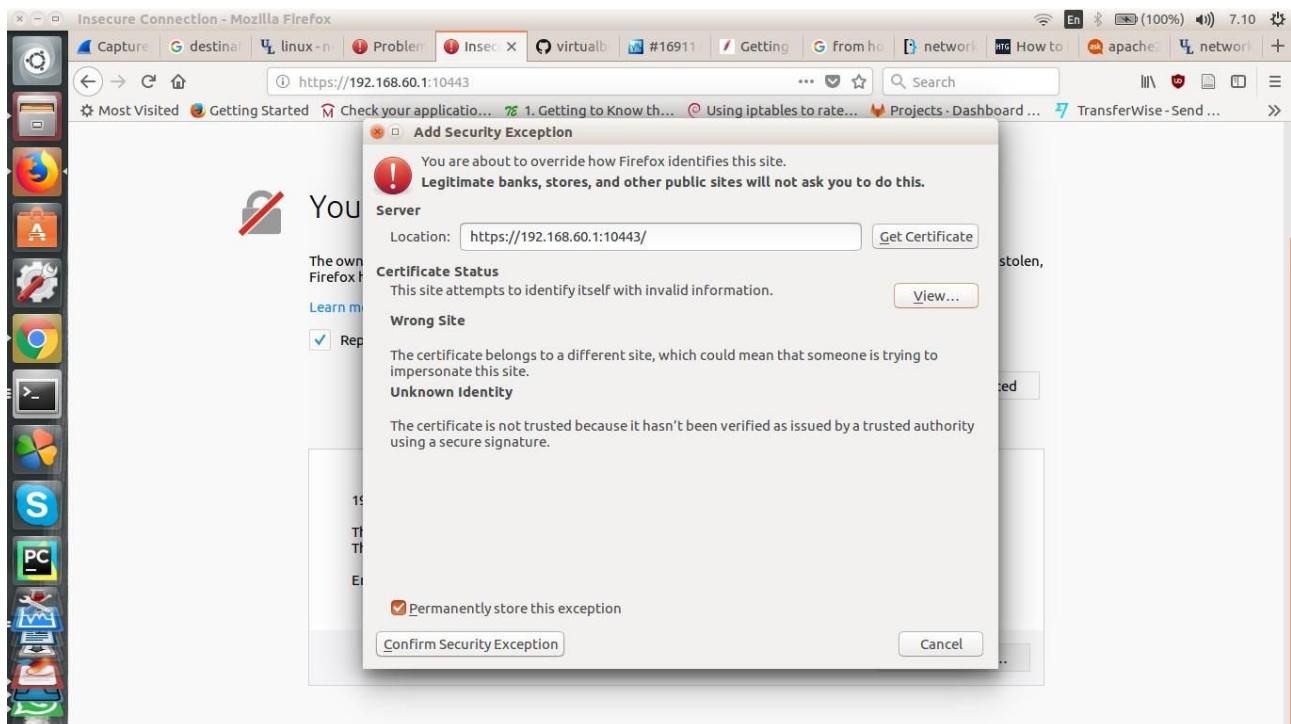
Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. 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:

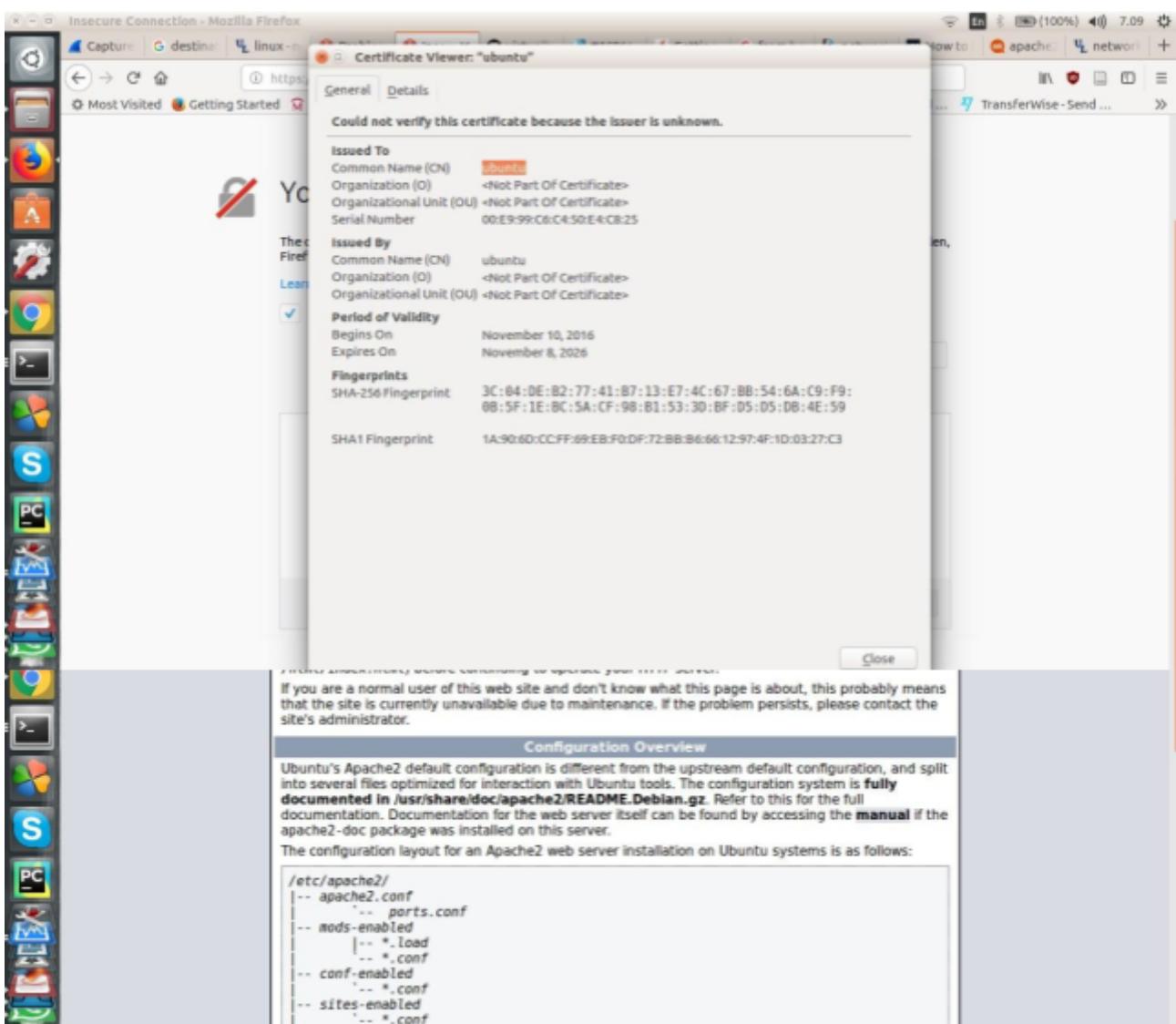
```
/etc/apache2/
|-- apache2.conf
|   '-- ports.conf
|-- mods-enabled
|   '-- *.load
|   '-- *.conf
|-- conf-enabled
|   '-- *.conf
|-- sites-enabled
|   '-- *.conf
```

For Host

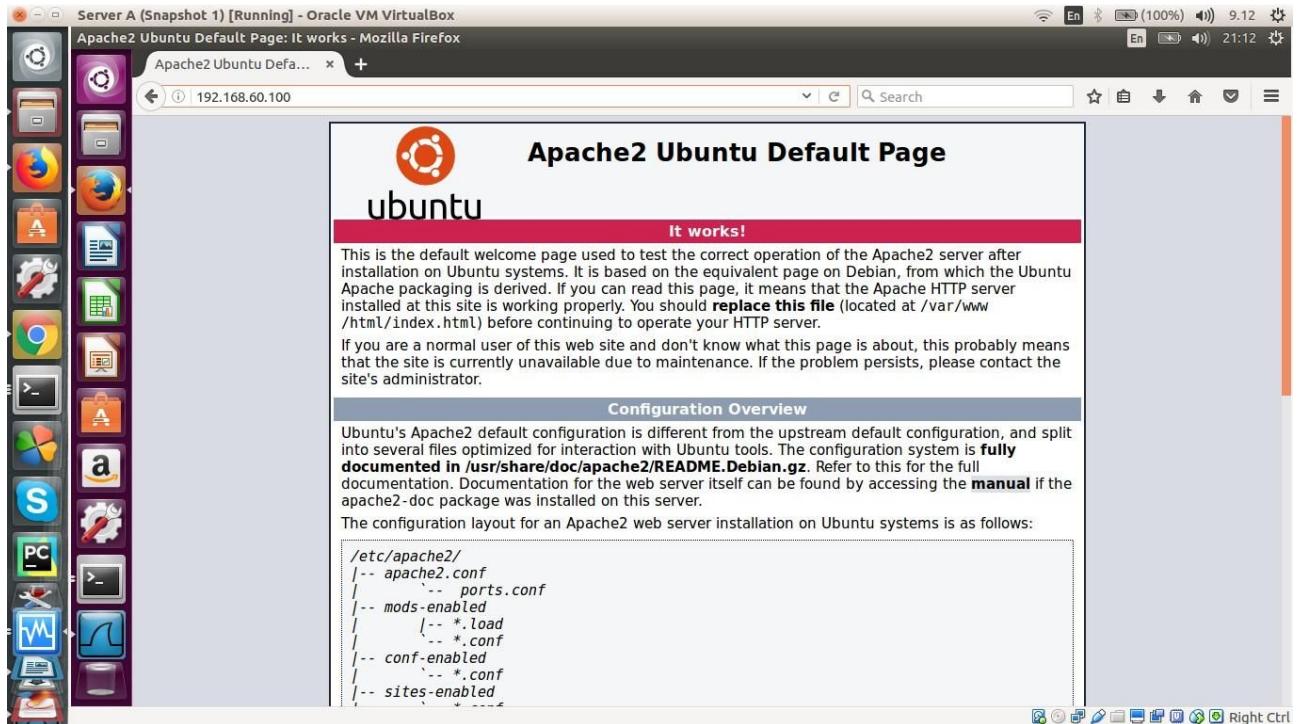
Host HTTPS



HOST HTTPS



Host: Https



Guest Apache2 server.

Task 10:

After giving `sudo iptables --list` command I got the below list and it showing all chain(INPUT,OUTPUT,FORWARD) will accept all data. There is no Drop or return filter used. This is the default policy.

```
ats@serverA:~$ sudo iptables --list
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
ats@serverA:~$
```

I have run the following commands for filter, mangle, and nat rules:

```
sudo iptables -t filter -L  
sudo iptables -t mangle -L  
sudo iptables -t nat -L
```

```
ats@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  
ats@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  
ats@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  
ats@serverA:~$ █
```

Task 11: Block HTTP-browsing in the guest OS

To verify http and https connection to the host and guest os I have used <http://www.httpvshttps.com/> url as per instruction.

For Host : HTTPS

The screenshot shows a web browser window with the URL <https://www.httpvshttps.com/>. The page title is "HTTP vs HTTPS Test". The content area features a large grid of green checkmarks. To the right, a "Page Info" sidebar is open, showing details for the HTTPS version of the page:

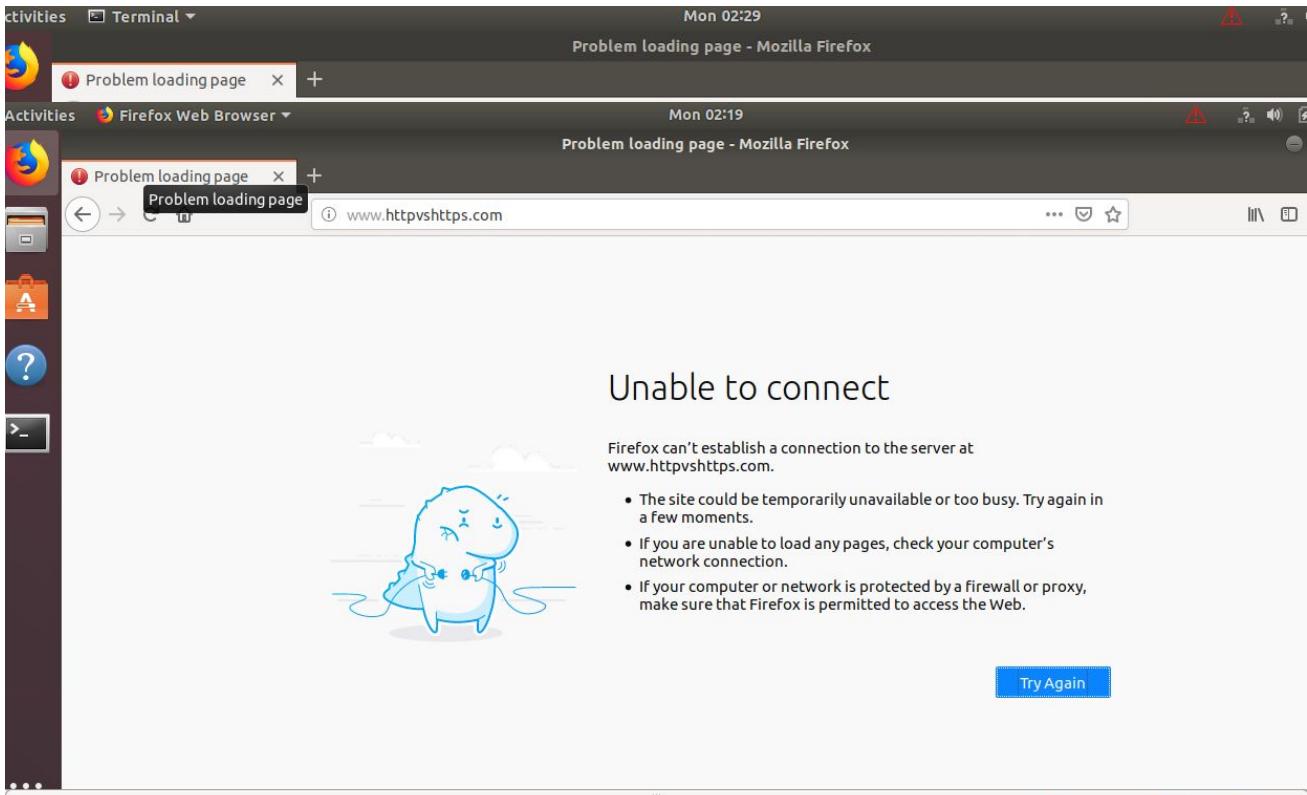
General	Media	Permissions	Security
Title: HTTP vs HTTPS — Test them both yourself			
Address: https://www.httpvshttps.com/			
Type: text/html			
Render Mode: Quirks mode			
Text Encoding: UTF-8			
Size: 7.33 KB (7,501 bytes)			
Referring URL: https://www.google.com/			
Modified: November 12, 2018, 2:24:49 AM GMT+1			
Meta (15 tags)			
Name	Content		
theme-color	#1ac222		
ip	83.253.18.32		
viewport	width=device-width, initial-scale=1, user-scalable=no		
twitter:card	summary		
descripton	Encrypted websites protect our privacy and are significantly faster. Run thi...		
twitter:description	Encrypted websites protect our privacy and are significantly faster. Run thi...		

For Host OS : HTTP

The screenshot shows a web browser window with the URL <http://www.httpvshttps.com/>. The page title is "HTTP vs HTTPS Test". The content area features a large green banner at the top stating "10.716 s" and "998% slower than HTTPS". Below the banner is a grid of green checkmarks. To the right, a "Page Info" sidebar is open, showing details for the HTTP version of the page:

General	Media	Permissions	Security
Title: HTTP vs HTTPS — Test them both yourself			
Address: http://www.httpvshttps.com/			
Type: text/html			
Render Mode: Quirks mode			
Text Encoding: UTF-8			
Size: 7.33 KB (7,501 bytes)			
Modified: November 12, 2018, 2:35:08 AM GMT+1			
Meta (15 tags)			
Name	Content		

For Server A (HTTP)



For Server A (HTTPS)

HTTP vs HTTPS — Test them both yourself - Mozilla Firefox

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 most accurate results, run each test 2-3 times in a private/incognito browser window.

Page Info - https://www.httpvshttps.com/

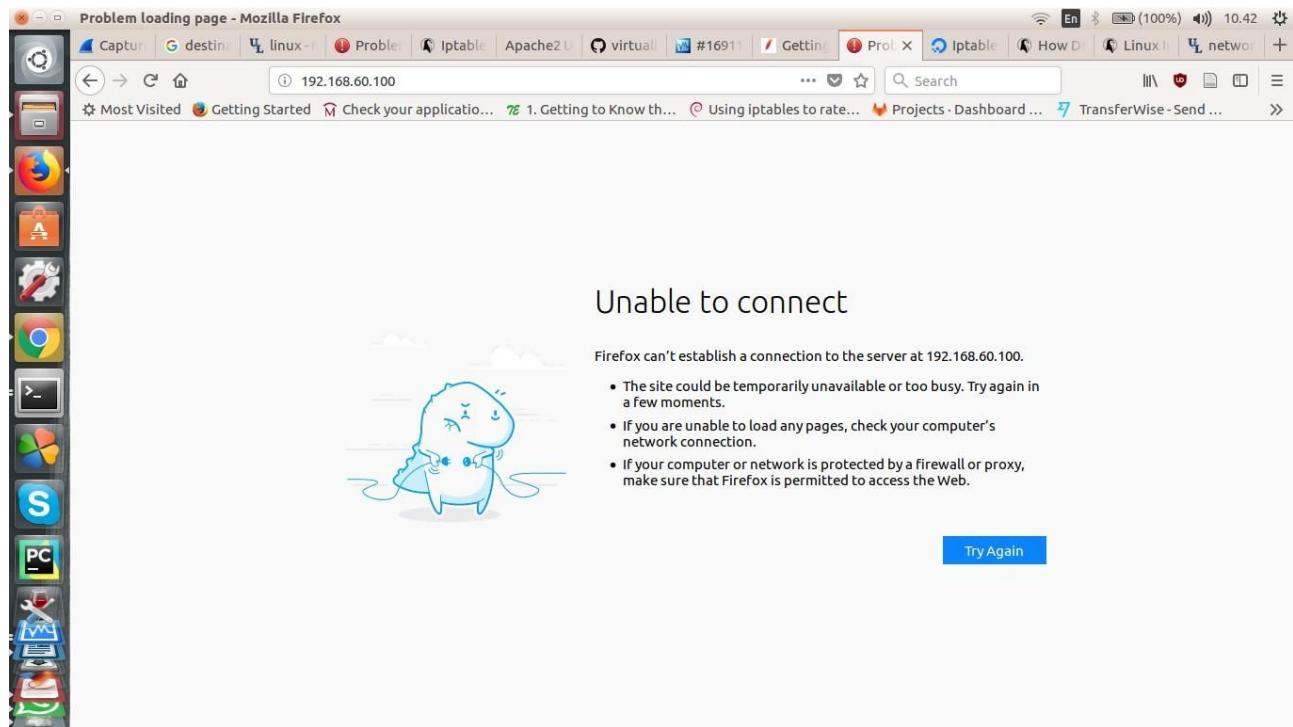
General	Media	Permissions	Security
Title: HTTP vs HTTPS — Test both yourself			
Address: https://www.httpvshttps.com/			
Type: text/html			
Render Mode: Quirks mode			
Text Encoding: UTF-8			
Size: 7.33 KB (7,501 bytes)			
Referring URL: https://www.google.com/			
Modified: November 12, 2018, 2:26:15 AM GMT+1			
Meta (15 tags)			
Name	Content		
theme-color	#1ac222		
ip	83.253.18.32		
viewport	width=device-width, initial-scale=1, user-scalable=no		

After blocking for verification purpose I tried to visit that previous website <http://www.httpvshttps.com> website and its confirmed that it has blocked.

Task 12:

For Blocking apache server I have used below iptable rule.

```
sudo iptables -A INPUT -p tcp --dport http -i enp0s9 -j REJECT
```



Task 13: Unblock HTTP-browsing in the guest OS

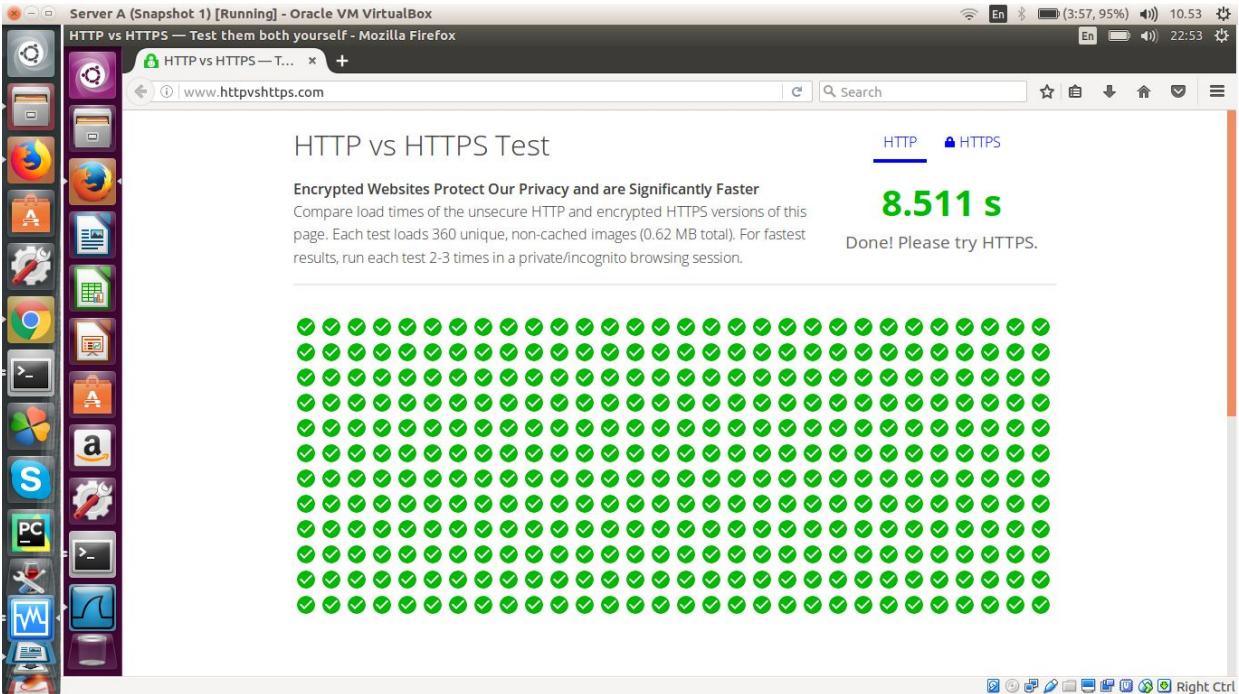
I have blocked the http at task 11 .

Now for unblocking that I have used below command:

```
sudo iptables -D INPUT -p tcp --dport 80 -m conntrack --ctstate NEW,ESTABLISHED  
-j DROP
```

```
sudo iptables -D INPUT -p tcp --sport 80 -m conntrack --ctstate ESTABLISHED -j DROP
```

All http request starts accepting.



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

For the task guest OS can view HTTP and HTTPS pages, but apache2 server is blocked from serving HTTP content I have added the below command to firewall.sh file.

Sudo iptables -A INPUT -p tcp --dport http -i enp0s9 -j REJECT

```

ats@serverA:~$ ifconfig
enp0s3      Link encap:Ethernet HWaddr 00:0C:29:14:4D:9B
            inet addr:192.168.60.100  Bcast:192.168.60.255  Mask:255.255.255.0
            inet6 addr: fe80::a00c:29ff:fe14:4d9b/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:1212 errors:0 drop:0  error rate:0.0%
            TX packets:92 errors:0 drop:0  error rate:0.0%
            collisions:0 txqueuelen:1000
            RX bytes:112312 (112.3 KB)
            RX bytes:112312 (112.3 KB)

enp0s8      Link encap:Ethernet HWaddr 00:0C:29:14:4D:9B
            inet addr:192.168.70.5  Bcast:192.168.70.255  Mask:255.255.255.0
            inet6 addr: fe80::a00c:29ff:fe14:4d9b/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:1290 errors:0 drop:0  error rate:0.0%
            TX packets:92 errors:0 drop:0  error rate:0.0%
            collisions:0 txqueuelen:1000
            RX bytes:116992 (116.9 KB)
            RX bytes:116992 (116.9 KB)

enp0s9      Link encap:Ethernet HWaddr 00:0C:29:14:4D:9B
            inet addr:10.98.100.1  Bcast:10.98.100.255  Mask:255.255.255.0
            inet6 addr: fe80::a00c:29ff:fe14:4d9b/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:16608 errors:0 drop:0  error rate:0.0%
            TX packets:9988 errors:0 drop:0  error rate:0.0%
            collisions:0 txqueuelen:1000
            RX bytes:9944002 (9.9 MB)
            RX bytes:9944002 (9.9 MB)

lo          Link encap:Local Loopback
            inet addr:127.0.0.1  Mask:255.0.0.0
            inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING  MTU:65536  Metric:1
            RX packets:15971 errors:0 drop:0  error rate:0.0%
            TX packets:15971 errors:0 drop:0  error rate:0.0%
            collisions:0 txqueuelen:1
            RX bytes:1338699 (1.3 MB)
            RX bytes:1338699 (1.3 MB)

ats@serverA:~$ cat /etc/firewall.sh
# 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
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:1212 errors:0 drop:0  error rate:0.0%
            TX packets:92 errors:0 drop:0  error rate:0.0%
            collisions:0 txqueuelen:1000
            RX bytes:112312 (112.3 KB)
            RX bytes:112312 (112.3 KB)

# Create logging chains
#$IPT -t filter -N input_log
#$IPT -t filter -N output_log
#$IPT -t filter -N forward_log
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:1290 errors:0 drop:0  error rate:0.0%
            TX packets:92 errors:0 drop:0  error rate:0.0%
            collisions:0 txqueuelen:1000
            RX bytes:116992 (116.9 KB)
            RX bytes:116992 (116.9 KB)

# Set some logging targets for DROPPED packets
#$IPT -t filter -A input_log -j LOG --log-level notice --log-prefix "input drop: "
#$IPT -t filter -A output_log -j LOG --log-level notice --log-prefix "output drop: "
#$IPT -t filter -A forward_log -j LOG --log-level notice --log-prefix "forward drop: "
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:16608 errors:0 drop:0  error rate:0.0%
            TX packets:9988 errors:0 drop:0  error rate:0.0%
            collisions:0 txqueuelen:1000
            RX bytes:9944002 (9.9 MB)
            RX bytes:9944002 (9.9 MB)

# Return from the logging chain to the built-in chain
Link encap:Local Loopback
#$IPT -t filter -A input_log -j RETURN
#$IPT -t filter -A output_log -j RETURN
#$IPT -t filter -A forward_log -j RETURN
UP LOOPBACK RUNNING  MTU:65536  Metric:1
            RX packets:15971 errors:0 drop:0  error rate:0.0%
            TX packets:15971 errors:0 drop:0  error rate:0.0%
            collisions:0 txqueuelen:1
            RX bytes:1338699 (1.3 MB)
            RX bytes:1338699 (1.3 MB)

# 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 TNPINPUT -j input_log
Saving file '/home/ats/firewall.sh'...
sh Tab Width: 8 Ln 41, Col 63 INS

```

I have changed the rule and after giving iptable -L command found below:

```
ats@serverA:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
          tcp   --  anywhere             anywhere            reject-with icmp-port-unreachable

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
          tcp   --  anywhere             anywhere            reject-with icmp-port-unreachable

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

Now I comment out that command and make the firewall initial state.

Task 15: Change default firewall policy to DROP

I have removed the rules from Task 14 as per instruction and executed the script.

Then I have modified the firewall.sh script by the following commands to make default firewall policy to DROP:

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

And I add it to the firewall.sh file.

Now Server A dropping all packet and stop all connection to the outside world.

Don't even get ping from the loopback/localhost.

```
ats@serverA:~$ sudo iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination
          tcp   --  anywhere             anywhere            reject-with icmp-port-unreachable

Chain FORWARD (policy DROP)
target     prot opt source               destination
          tcp   --  anywhere             anywhere            reject-with icmp-port-unreachable

Chain OUTPUT (policy DROP)
target     prot opt source               destination
          tcp   --  anywhere             anywhere            reject-with icmp-port-unreachable
ats@serverA:~$
```

Task 16: Logging DROPPED packets

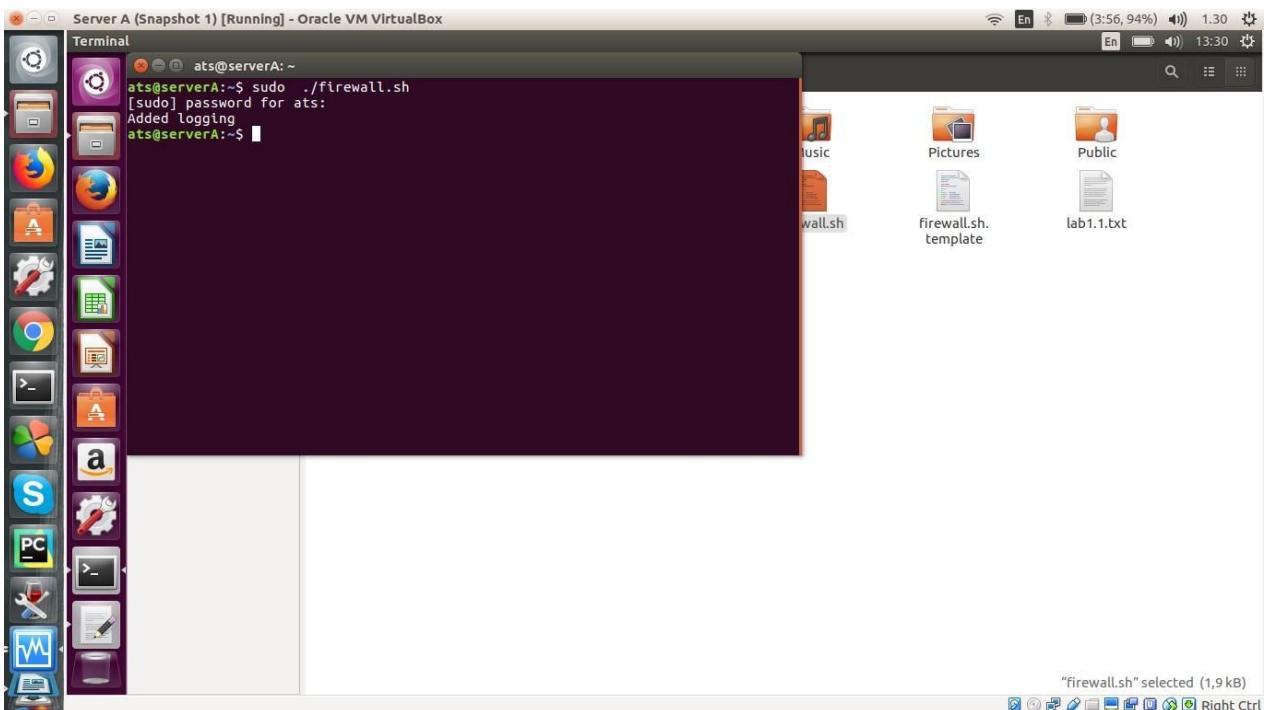
I have modified the firewall script as per instruction and executed it
Then in the terminal window, I have run the following command:

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

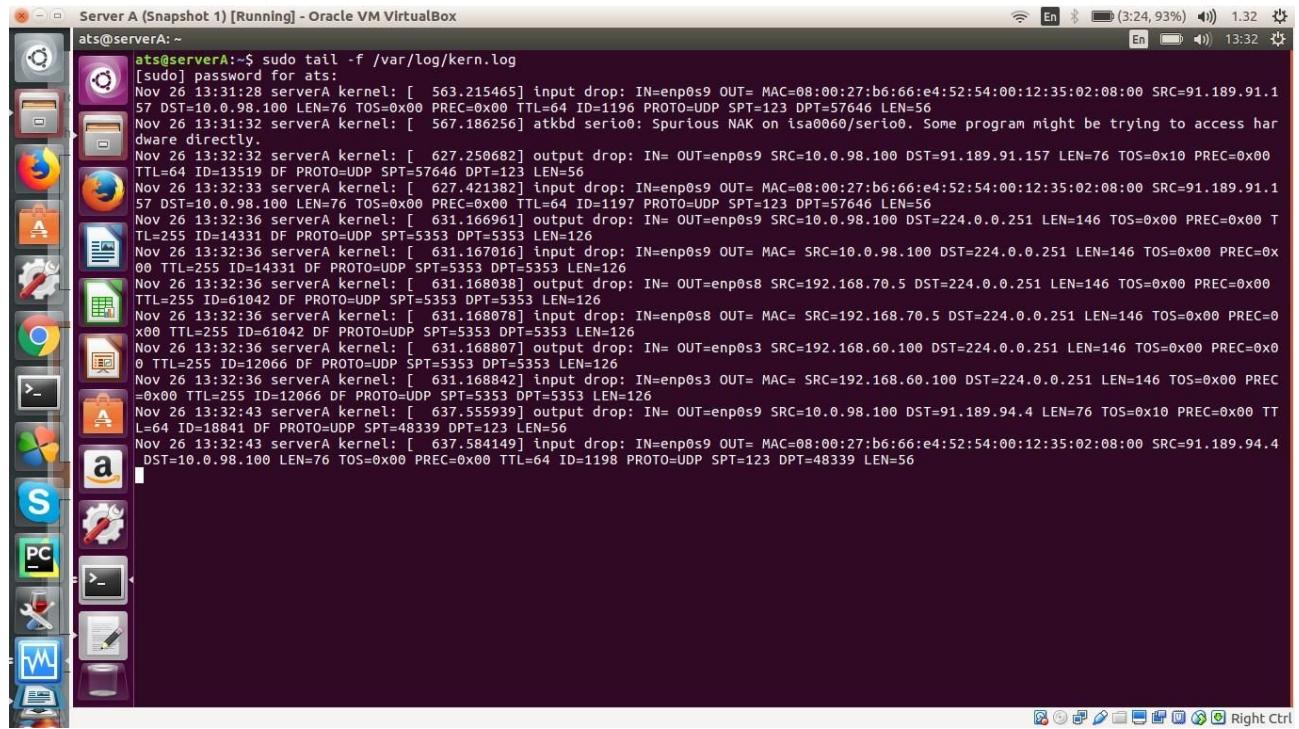
Now I am able to see live logs from the Linux kernel. I have started pinging the loopback interface in another terminal window and I can see all outputs are dropped .

```
CMP TYPE=8 CODE=0 ID=3585 SEQ=18
Mar 25 20:41:03 serverA kernel: [ 5475.872460] output drop: IN= OUT=lo SRC=127.0.0.1 DST=127.0.0.1 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=16451 DF PROTO=I
CMP TYPE=8 CODE=0 ID=3585 SEQ=19
Mar 25 20:41:04 serverA kernel: [ 5476.872411] output drop: IN= OUT=lo SRC=127.0.0.1 DST=127.0.0.1 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=16576 DF PROTO=I
CMP TYPE=8 CODE=0 ID=3585 SEQ=20
Mar 25 20:41:05 serverA kernel: [ 5477.873359] output drop: IN= OUT=lo SRC=127.0.0.1 DST=127.0.0.1 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=16825 DF PROTO=I
CMP TYPE=8 CODE=0 ID=3585 SEQ=21
Mar 25 20:41:06 serverA kernel: [ 5478.872720] output drop: IN= OUT=lo SRC=127.0.0.1 DST=127.0.0.1 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=16942 DF PROTO=I
CMP TYPE=8 CODE=0 ID=3585 SEQ=22
Mar 25 20:41:07 serverA kernel: [ 5479.872607] output drop: IN= OUT=lo SRC=127.0.0.1 DST=127.0.0.1 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=17172 DF PROTO=I
CMP TYPE=8 CODE=0 ID=3585 SEQ=23
Mar 25 20:41:08 serverA kernel: [ 5480.872781] output drop: IN= OUT=lo SRC=127.0.0.1 DST=127.0.0.1 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=17407 DF PROTO=I
CMP TYPE=8 CODE=0 ID=3585 SEQ=24
Mar 25 20:41:09 serverA kernel: [ 5481.872526] output drop: IN= OUT=lo SRC=127.0.0.1 DST=127.0.0.1 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=17429 DF PROTO=I
CMP TYPE=8 CODE=0 ID=3585 SEQ=25
Mar 25 20:41:10 serverA kernel: [ 5482.872512] output drop: IN= OUT=lo SRC=127.0.0.1 DST=127.0.0.1 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=17564 DF PROTO=I
CMP TYPE=8 CODE=0 ID=358
Mar 25 20:41:11 serverA ~ ats@serverA: ~
CMP TYPE=8 CODE=0 ID=358ping: sendmsg: Operation not permitted
Mar 25 20:41:12 serverA ping: sendmsg: Operation not permitted
CMP TYPE=8 CODE=0 ID=358ping: sendmsg: Operation not permitted
Mar 25 20:41:13 serverA ping: sendmsg: Operation not permitted
CMP TYPE=8 CODE=0 ID=358ping: sendmsg: Operation not permitted
Mar 25 20:41:14 serverA ping: sendmsg: Operation not permitted
CMP TYPE=8 CODE=0 ID=358ping: sendmsg: Operation not permitted
Mar 25 20:41:15 serverA ping: sendmsg: Operation not permitted
CMP TYPE=8 CODE=0 ID=358ping: sendmsg: Operation not permitted
Mar 25 20:41:16 serverA ping: sendmsg: Operation not permitted
CMP TYPE=8 CODE=0 ID=358ping: sendmsg: Operation not permitted
```

Added logging:

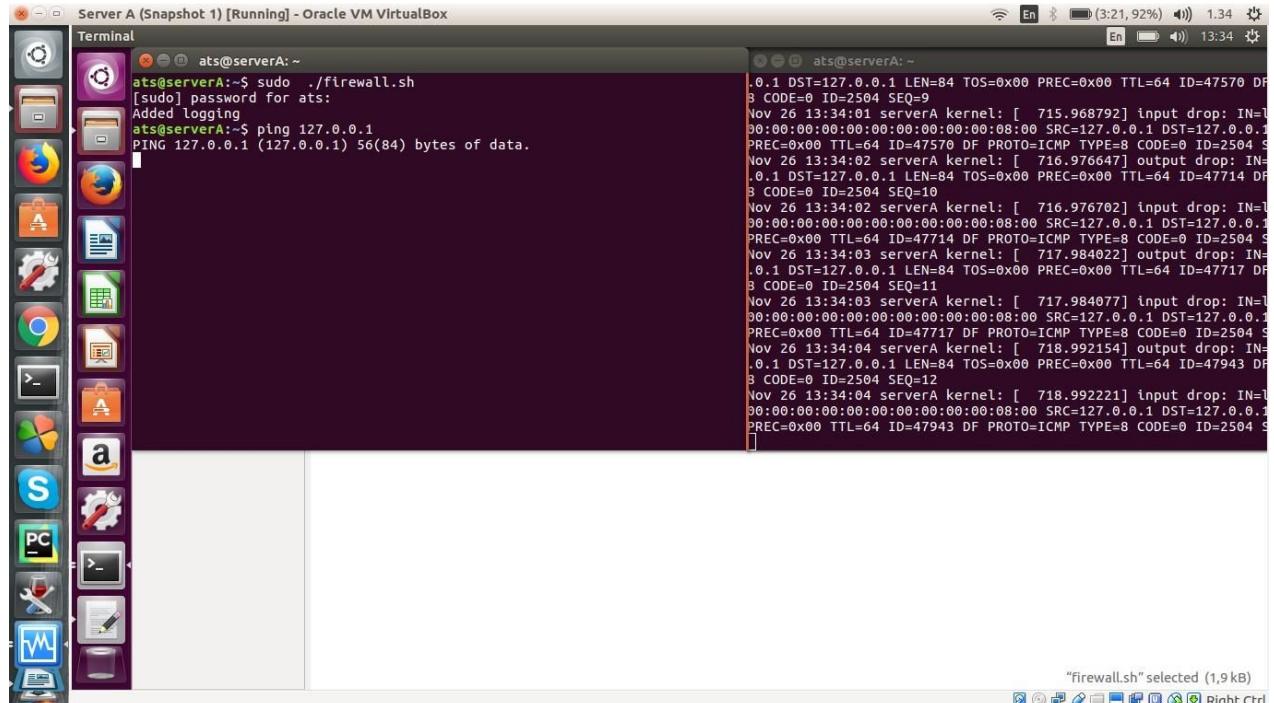


After Blocking I have seen that all inputs are dropped.



```
ats@serverA:~$ sudo tail -f /var/log/kern.log
[sudo] password for ats:
Nov 26 13:31:28 serverA kernel: [ 563.215465] input drop: IN=enp0s9 OUT= MAC=08:00:27:b6:6e:e4:52:54:00:12:35:02:08:00 SRC=91.189.91.1
57 DST=10.0.98.100 LEN=76 TOS=0x00 PREC=0x00 TTL=64 ID=1196 PROTO=UDP SPT=123 DPT=57646 LEN=56
Nov 26 13:31:32 serverA kernel: [ 567.186256] atkbd serio0: Spurious NAK on isa0060/serio0. Some program might be trying to access hardware directly.
Nov 26 13:32:32 serverA kernel: [ 627.250682] output drop: IN= OUT=enp0s9 SRC=10.0.98.100 DST=91.189.91.157 LEN=76 TOS=0x10 PREC=0x00 TTL=64 ID=13519 DF PROTO=UDP SPT=57646 LEN=56
Nov 26 13:32:33 serverA kernel: [ 627.421382] input drop: IN=enp0s9 OUT= MAC=08:00:27:b6:6e:e4:52:54:00:12:35:02:08:00 SRC=91.189.91.1
57 DST=10.0.98.100 LEN=76 TOS=0x00 PREC=0x00 TTL=64 ID=1197 PROTO=UDP SPT=123 DPT=57646 LEN=56
Nov 26 13:32:36 serverA kernel: [ 631.166961] output drop: IN= OUT=enp0s9 SRC=10.0.98.100 DST=224.0.0.251 LEN=146 TOS=0x00 PREC=0x00 TTL=255 ID=14331 DF PROTO=UDP SPT=5353 DPT=5353 LEN=126
Nov 26 13:32:36 serverA kernel: [ 631.167016] input drop: IN=enp0s9 OUT= MAC= SRC=10.0.98.100 DST=224.0.0.251 LEN=146 TOS=0x00 PREC=0x00 TTL=255 ID=14331 DF PROTO=UDP SPT=5353 DPT=5353 LEN=126
Nov 26 13:32:36 serverA kernel: [ 631.168038] output drop: IN= OUT=enp0s8 SRC=192.168.70.5 DST=224.0.0.251 LEN=146 TOS=0x00 PREC=0x00 TTL=255 ID=61042 DF PROTO=UDP SPT=5353 DPT=5353 LEN=126
Nov 26 13:32:36 serverA kernel: [ 631.168078] input drop: IN=enp0s8 OUT= MAC= SRC=192.168.70.5 DST=224.0.0.251 LEN=146 TOS=0x00 PREC=0x00 TTL=255 ID=61042 DF PROTO=UDP SPT=5353 DPT=5353 LEN=126
Nov 26 13:32:36 serverA kernel: [ 631.168807] output drop: IN= OUT=enp0s3 SRC=192.168.60.100 DST=224.0.0.251 LEN=146 TOS=0x00 PREC=0x00 TTL=255 ID=12066 DF PROTO=UDP SPT=5353 DPT=5353 LEN=126
Nov 26 13:32:36 serverA kernel: [ 631.168842] input drop: IN=enp0s3 OUT= MAC= SRC=192.168.60.100 DST=224.0.0.251 LEN=146 TOS=0x00 PREC=0x00 TTL=255 ID=12066 DF PROTO=UDP SPT=5353 DPT=5353 LEN=126
Nov 26 13:32:43 serverA kernel: [ 637.555939] output drop: IN= OUT=enp0s9 SRC=10.0.98.100 DST=91.189.94.4 LEN=76 TOS=0x10 PREC=0x00 TTL=64 ID=18841 DF PROTO=UDP SPT=48339 DPT=123 LEN=56
Nov 26 13:32:43 serverA kernel: [ 637.584149] input drop: IN=enp0s9 OUT= MAC=08:00:27:b6:6e:e4:52:54:00:12:35:02:08:00 SRC=91.189.94.4
DST=10.0.98.100 LEN=76 TOS=0x00 PREC=0x00 TTL=64 ID=1198 PROTO=UDP SPT=123 DPT=48339 LEN=56
```

Try to ping loopback but it also dropping that.



```
ats@serverA:~$ sudo ./firewall.sh
[sudo] password for ats:
Added logging
ats@serverA:~$ ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.

Nov 26 13:34:01 serverA kernel: [ 715.968792] input drop: IN=10:00:00:00:00:00:00:00:00:00:08:00 SRC=127.0.0.1 DST=127.0.0.1 PREC=0x00 TTL=64 ID=47570 DF=0 CODE=0 ID=2504 SEQ=9
Nov 26 13:34:01 serverA kernel: [ 715.968792] output drop: IN=10:00:00:00:00:00:00:00:00:00:08:00 SRC=127.0.0.1 DST=127.0.0.1 PREC=0x00 TTL=64 ID=47570 DF=0 PROTO=ICMP TYPE=8 CODE=0 ID=2504 SEQ=9
Nov 26 13:34:02 serverA kernel: [ 716.976647] input drop: IN=10:00:00:00:00:00:00:00:00:00:08:00 SRC=127.0.0.1 DST=127.0.0.1 PREC=0x00 TTL=64 ID=47714 DF=0 PROTO=ICMP TYPE=8 CODE=0 ID=2504 SEQ=10
Nov 26 13:34:02 serverA kernel: [ 716.976702] output drop: IN=10:00:00:00:00:00:00:00:00:00:08:00 SRC=127.0.0.1 DST=127.0.0.1 PREC=0x00 TTL=64 ID=47714 DF=0 PROTO=ICMP TYPE=8 CODE=0 ID=2504 SEQ=10
Nov 26 13:34:03 serverA kernel: [ 717.984022] input drop: IN=10:00:00:00:00:00:00:00:00:00:08:00 SRC=127.0.0.1 DST=127.0.0.1 PREC=0x00 TTL=64 ID=47717 DF=0 PROTO=ICMP TYPE=8 CODE=0 ID=2504 SEQ=11
Nov 26 13:34:03 serverA kernel: [ 717.984077] output drop: IN=10:00:00:00:00:00:00:00:00:00:08:00 SRC=127.0.0.1 DST=127.0.0.1 PREC=0x00 TTL=64 ID=47717 DF=0 PROTO=ICMP TYPE=8 CODE=0 ID=2504 SEQ=11
Nov 26 13:34:04 serverA kernel: [ 718.992154] input drop: IN=10:00:00:00:00:00:00:00:00:00:08:00 SRC=127.0.0.1 DST=127.0.0.1 PREC=0x00 TTL=64 ID=47943 DF=0 PROTO=ICMP TYPE=8 CODE=0 ID=2504 SEQ=12
Nov 26 13:34:04 serverA kernel: [ 718.992221] output drop: IN=10:00:00:00:00:00:00:00:00:00:08:00 SRC=127.0.0.1 DST=127.0.0.1 PREC=0x00 TTL=64 ID=47943 DF=0 PROTO=ICMP TYPE=8 CODE=0 ID=2504 SEQ=12
"firewall.sh" selected (1.9 kB)
```

Task 17: Enable traffic from loopback interface

For enabling traffic from loopback I have applied below rules:

Sudo iptables -A INPUT -i lo -j ACCEPT

Sudo iptables -A OUTPUT -o lo -j ACCEPT

Fro enabling SSH I have applied below rules:

```
sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
sudo iptables -A OUTPUT -p tcp --sport 22 -j ACCEPT
```

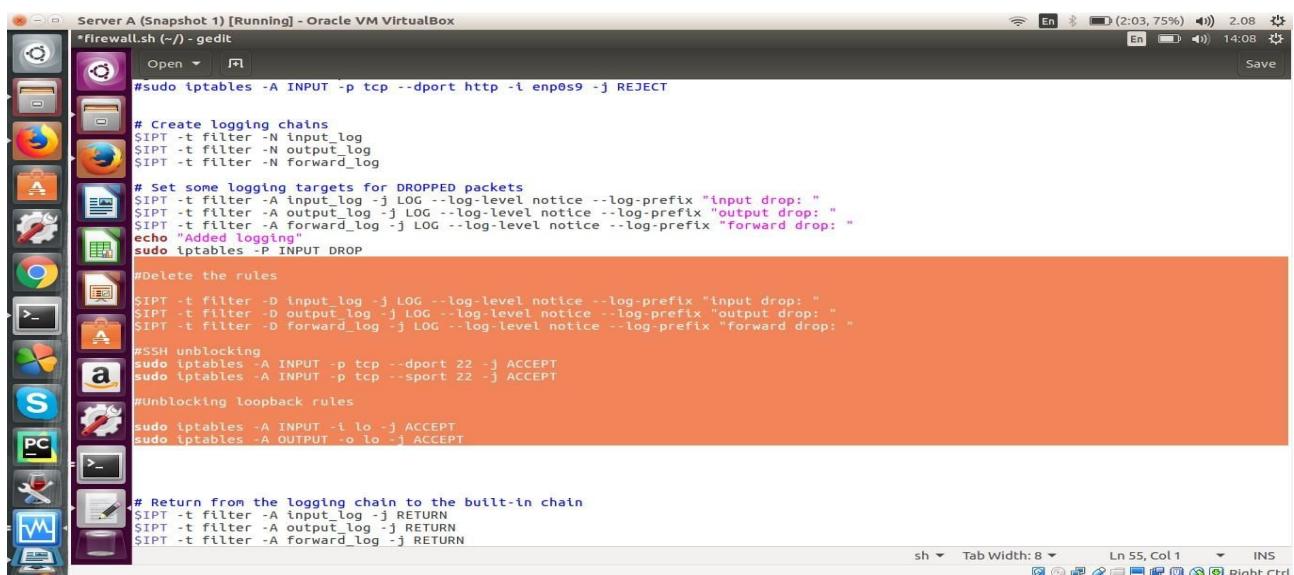
Now I am getting ping to the loopback and ssh from my host.

```
ats@serverA:~$ ssh localhost
ats@localhost's password:
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-104-generic x86_64)

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

416 packages can be updated.
136 updates are security updates.

Last login: Mon Nov 27 18:44:25 2017 from 10.0.98.2
ats@serverA:~$ logout
Connection to localhost closed.
ats@serverA:~$
```



The screenshot shows a terminal window titled "Server A (Snapshot 1) [Running] - Oracle VM VirtualBox". The window contains the following iptables configuration script:

```
#!/bin/sh
#sudo iptables -A INPUT -p tcp --dport http -i enp0s9 -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-level notice --log-prefix "input drop: "
$IPT -t filter -A output_log -j LOG --log-level notice --log-prefix "output drop: "
$IPT -t filter -A forward_log -j LOG --log-level notice --log-prefix "forward drop: "
echo "Added logging"
sudo iptables -P INPUT DROP
#
#Delete the rules
$IPT -t filter -D input_log -j LOG --log-level notice --log-prefix "input drop: "
$IPT -t filter -D output_log -j LOG --log-level notice --log-prefix "output drop: "
$IPT -t filter -D forward_log -j LOG --log-level notice --log-prefix "forward drop: "
#
#SSH unblocking
sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
sudo iptables -A INPUT -p tcp --sport 22 -j ACCEPT
#
#Unblocking loopback rules
sudo iptables -A INPUT -i lo -j ACCEPT
sudo iptables -A OUTPUT -o lo -j ACCEPT
#
# 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
```

Server A (Snapshot 1) [Running] - Oracle VM VirtualBox

Terminal

ats@serverA: ~

```
rtt min/avg/max/mdev = 0.035/0.042/0.050/0.005 ms
ats@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=0.101 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.111 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.165 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.104 ms
64 bytes from 127.0.0.1: icmp_seq=5 ttl=64 time=0.029 ms
64 bytes from 127.0.0.1: icmp_seq=6 ttl=64 time=0.078 ms
64 bytes from 127.0.0.1: icmp_seq=7 ttl=64 time=0.093 ms
64 bytes from 127.0.0.1: icmp_seq=8 ttl=64 time=0.062 ms
64 bytes from 127.0.0.1: icmp_seq=9 ttl=64 time=0.085 ms
64 bytes from 127.0.0.1: icmp_seq=10 ttl=64 time=0.085 ms
64 bytes from 127.0.0.1: icmp_seq=11 ttl=64 time=0.083 ms
64 bytes from 127.0.0.1: icmp_seq=12 ttl=64 time=0.123 ms
64 bytes from 127.0.0.1: icmp_seq=13 ttl=64 time=0.094 ms
64
```

saddam@bth:~\$ sudo ssh -p 10022 ats@localhost

```
[sudo] password for saddam:
```

Sudo aptables -A INPUT -p tcp --dport 22 -j ACCEPT

Sudo aptables -A INPUT -p tcp --sport 22 -j ACCEPT

saddam@bth:~\$ sudo ssh -p 10022 ats@localhost

```
Sudo aptables -A OUTPUT -o lo -j ACCEPT
```

Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-47-generic x86_64)

SIPT -t filter -A input log -j RETURN

SIPT -t filter -A output log -j RETURN

SIPT Documentation: https://help.ubuntu.com

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

These rules must be inserted at the end of the built-in chain to log packets that will be dropped by the default # 523 packages can be updated.

SIPT updates are security updates

SIPT - filter -A OUTPUT -j output_log

SIPT - filter -A FORWARD -j forward_log

Last login: Sat Nov 25 18:46:47 2017 from 10.0.98.2

ats@serverA:~\$

serverA:~\$ sudo tail -f /var/log/kern.log

```
password for ats:
Nov 26 13:50:51 serverA kernel: [ 1725.402717] atkbd serio0: Spuriou serio0. Some program might be trying to access hardware directl
Nov 26 13:52:54 serverA kernel: [ 1849.176136] atkbd serio0: Spuriou serio0. Some program might be trying to access hardware directl
Nov 26 13:53:09 serverA kernel: [ 1863.509384] atkbd serio0: Spuriou serio0. Some program might be trying to access hardware directl
Nov 26 13:53:53 serverA kernel: [ 1908.142046] atkbd serio0: Spuriou serio0. Some program might be trying to access hardware directl
Nov 26 13:55:06 serverA kernel: [ 1981.207361] atkbd serio0: Spuriou serio0. Some program might be trying to access hardware directl
Nov 26 14:01:12 serverA kernel: [ 2346.692166] atkbd serio0: Spuriou serio0. Some program might be trying to access hardware directl
Nov 26 14:01:20 serverA kernel: [ 2354.626623] atkbd serio0: Spuriou serio0. Some program might be trying to access hardware directl
Nov 26 14:01:31 serverA kernel: [ 2365.755675] atkbd serio0: Spuriou serio0. Some program might be trying to access hardware directl
Nov 26 14:03:48 serverA kernel: [ 2502.237276] atkbd serio0: Spuriou serio0. Some program might be trying to access hardware directl
Nov 26 14:04:49 serverA kernel: [ 2563.973891] atkbd serio0: Spuriou serio0. Some program might be trying to access hardware directl
```

sh Tab Width: 8 Ln 63, Col 17 INS

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

For allowing icmp to the outside world I have set below rules:

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

```
ls@serverA:~          En  23: Last login: Sun Mar 25 22:43:18 on console
GNU nano 2.5.3        File: firewall.sh
$IPPT -P INPUT DROP
$IPPT -P OUTPUT DROP
$IPPT -P FORWARD DROP

$IPPT -A INPUT -p tcp --dport 80 -j REJECT
$IPPT -A OUTPUT -p tcp --dport 80 -j REJECT
$IPPT -D INPUT -p tcp --dport 80 -j REJECT

$IPPT -A INPUT -i lo -j ACCEPT
$IPPT -A OUTPUT -o lo -j ACCEPT

$IPPT -A OUTPUT -p icmp --icmp-type echo-request -j ACCEPT
$IPPT -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT

$IPPT -A OUTPUT -p udp -m conntrack --ctstate \NEW,ESTABLISHED -j ACCEP
$IPPT -A INPUT -p udp -m conntrack --ctstate ESTABLISHED,RELATED -j ACC 416 packages can be updated.
136 updates are security updates.

$IPPT -A OUTPUT -p tcp -m conntrack --ctstate \NEW,ESTABLISHED -j ACCEP
$IPPT -A OUTPUT -p tcp -m conntrack --ctstate \NEW,ESTABLISHED -j ACCEP Last login: Sun Mar 25 20:50:23 2018 from 127.0.0.1
$IPPT -A INPUT -p tcp --dport 443 -j ACCEPT
$IPPT -A INPUT -p tcp --dport 22 -j ACCEPT
ats@serverA:~$ █
```

I can ping google dns server 8.8.8.8 from my server A.

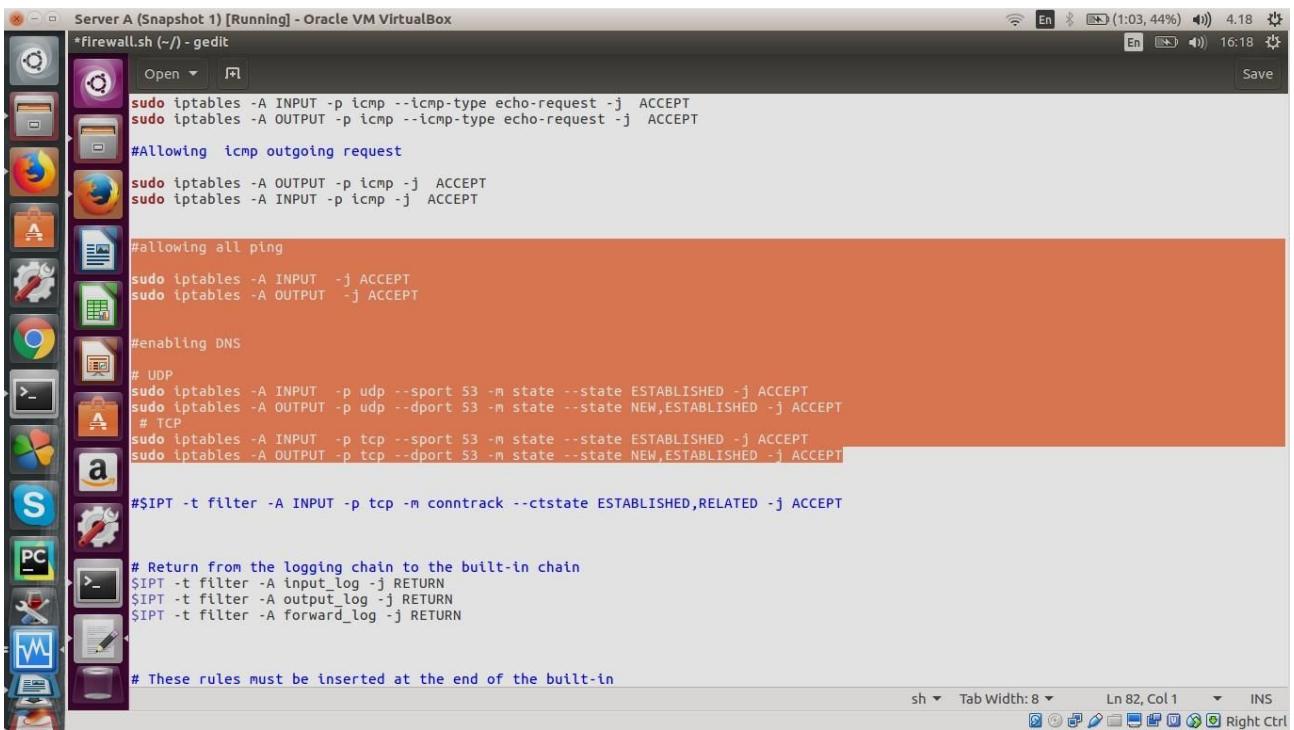
Task 19: Allow Server A to ping all hosts

For allowing all ping and DNS server I have used below rules:

I have modified the firewall script by following rules to allow server A to ping all hosts:

```
$IPT -A OUTPUT -j ACCEPT  
$IPT -A INPUT -j ACCEPT  
$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
```

```
ats@serverA:~$ sudo nano firewall.sh  
ats@serverA:~$ sudo ./firewall.sh  
Added logging  
ats@serverA:~$ ping 10.0.98.100  
PING 10.0.98.100 (10.0.98.100) 56(84) bytes of data.  
64 bytes from 10.0.98.100: icmp_seq=1 ttl=64 time=0.075 ms  
64 bytes from 10.0.98.100: icmp_seq=2 ttl=64 time=0.074 ms  
64 bytes from 10.0.98.100: icmp_seq=3 ttl=64 time=0.061 ms  
64 bytes from 10.0.98.100: icmp_seq=4 ttl=64 time=0.218 ms  
64 bytes from 10.0.98.100: icmp_seq=5 ttl=64 time=0.074 ms  
64 bytes from 10.0.98.100: icmp_seq=6 ttl=64 time=0.087 ms  
64 bytes from 10.0.98.100: icmp_seq=7 ttl=64 time=0.127 ms  
64 bytes from 10.0.98.100: icmp_seq=8 ttl=64 time=0.079 ms  
64 bytes from 10.0.98.100: icmp_seq=9 ttl=64 time=0.073 ms  
64 bytes from 10.0.98.100: icmp_seq=10 ttl=64 time=0.057 ms  
64 bytes from 10.0.98.100: icmp_seq=11 ttl=64 time=0.076 ms  
^C  
--- 10.0.98.100 ping statistics ---
```



```
sudo iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
sudo iptables -A OUTPUT -p icmp --icmp-type echo-request -j ACCEPT

#Allowing icmp outgoing request

sudo iptables -A OUTPUT -p icmp -j ACCEPT
sudo iptables -A INPUT -p icmp -j ACCEPT

#allowing all ping

sudo iptables -A INPUT -j ACCEPT
sudo iptables -A OUTPUT -j ACCEPT

#enabling DNS

# UDP
sudo iptables -A INPUT -p udp --sport 53 -m state --state ESTABLISHED -j ACCEPT
sudo iptables -A OUTPUT -p udp --dport 53 -m state --state NEW,ESTABLISHED -j ACCEPT
# TCP
sudo iptables -A INPUT -p tcp --sport 53 -m state --state ESTABLISHED -j ACCEPT
sudo iptables -A OUTPUT -p tcp --dport 53 -m state --state NEW,ESTABLISHED -j ACCEPT

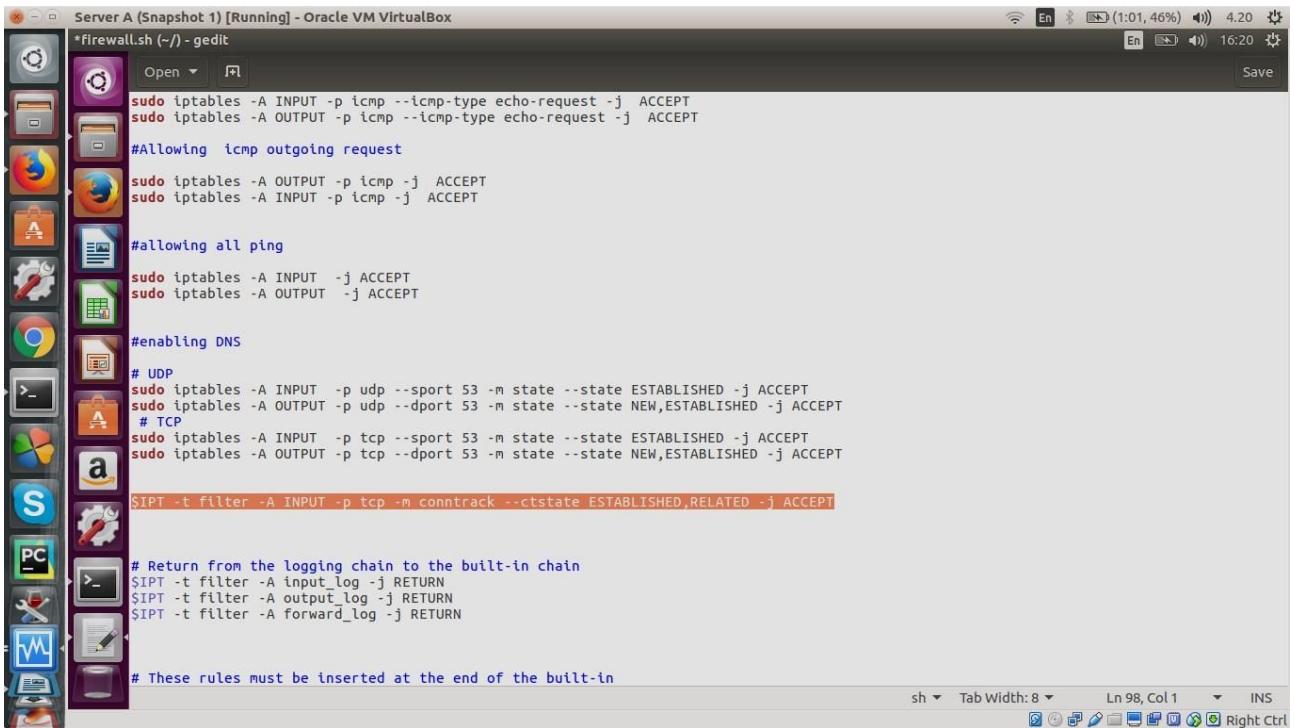
#SIPT -t filter -A INPUT -p tcp -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

# 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
```

Task 20:

Sudo iptables -t filter -A INPUT -p tcp -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT



```
sudo iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
sudo iptables -A OUTPUT -p icmp --icmp-type echo-request -j ACCEPT

#Allowing icmp outgoing request

sudo iptables -A OUTPUT -p icmp -j ACCEPT
sudo iptables -A INPUT -p icmp -j ACCEPT

#allowing all ping

sudo iptables -A INPUT -j ACCEPT
sudo iptables -A OUTPUT -j ACCEPT

#enabling DNS

# UDP
sudo iptables -A INPUT -p udp --sport 53 -m state --state ESTABLISHED -j ACCEPT
sudo iptables -A OUTPUT -p udp --dport 53 -m state --state NEW,ESTABLISHED -j ACCEPT
# TCP
sudo iptables -A INPUT -p tcp --sport 53 -m state --state ESTABLISHED -j ACCEPT
sudo iptables -A OUTPUT -p tcp --dport 53 -m state --state NEW,ESTABLISHED -j ACCEPT

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

# 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
```

Now outside connection has established and I can browse www.facebook.com from server A

Task 21:

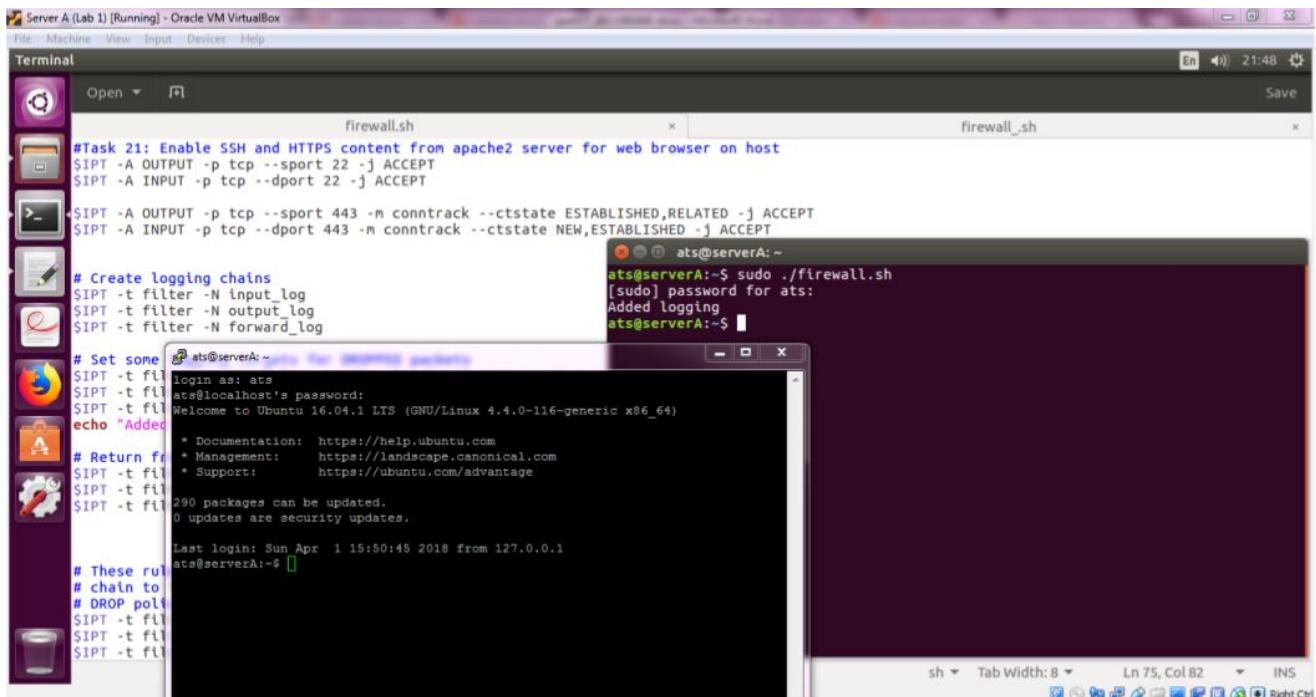
To add the following rules to the firewall.sh script and executing it and thus enable TCP connections to be established to any destination, so we can able to browse websites with the Firefox browser from Server A.

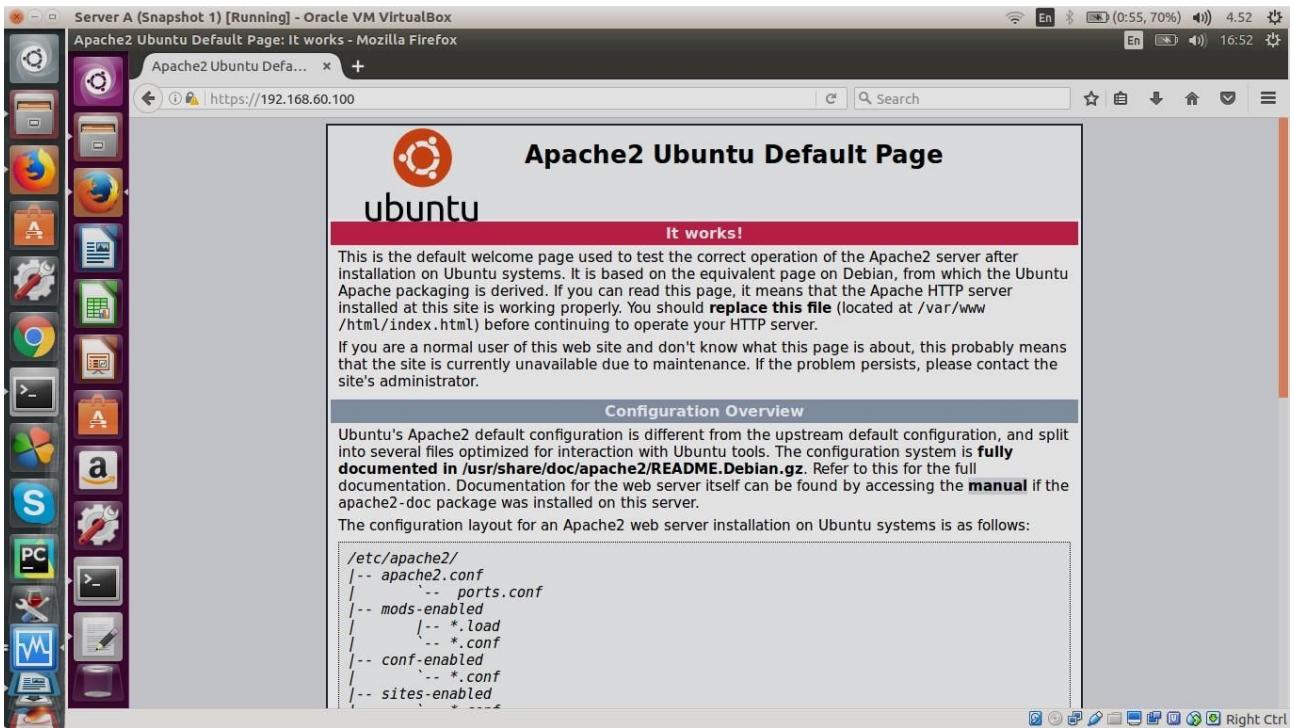
```
$IPT -A OUTPUT -p tcp --sport 22 -j ACCEPT  
$IPT -A INPUT -p tcp --dport 22 -j ACCEPT
```

Sudo iptables -A OUTPUT -p tcp -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT

"sudo iptables -A INPUT -p tcp -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

I can browse https apache2 server.





Task 22:

I can ping from Client A to Server A.

```
ats@serverA:~$ sudo nano firewall.sh
ats@serverA:~$ ping 192.168.60.111
PING 192.168.60.111 (192.168.60.111) 56(84) bytes of data.
64 bytes from 192.168.60.111: icmp_seq=1 ttl=64 time=0.342 ms
64 bytes from 192.168.60.111: icmp_seq=2 ttl=64 time=0.537 ms
64 bytes from 192.168.60.111: icmp_seq=3 ttl=64 time=0.329 ms
64 bytes from 192.168.60.111: icmp_seq=4 ttl=64 time=0.454 ms
64 bytes from 192.168.60.111: icmp_seq=5 ttl=64 time=0.529 ms
64 bytes from 192.168.60.111: icmp_seq=6 ttl=64 time=0.579 ms
64 bytes from 192.168.60.111: icmp_seq=7 ttl=64 time=0.504 ms
64 bytes from 192.168.60.111: icmp_seq=8 ttl=64 time=0.618 ms
64 bytes from 192.168.60.111: icmp_seq=9 ttl=64 time=0.499 ms
64 bytes from 192.168.60.111: icmp_seq=10 ttl=64 time=0.329 ms
64 bytes from 192.168.60.111: icmp_seq=11 ttl=64 time=0.345 ms
64 bytes from 192.168.60.111: icmp_seq=12 ttl=64 time=0.294 ms
64 bytes from 192.168.60.111: icmp_seq=13 ttl=64 time=0.421 ms
64 bytes from 192.168.60.111: icmp_seq=14 ttl=64 time=0.305 ms
64 bytes from 192.168.60.111: icmp_seq=15 ttl=64 time=0.461 ms
64 bytes from 192.168.60.111: icmp_seq=16 ttl=64 time=0.662 ms
64 bytes from 192.168.60.111: icmp_seq=17 ttl=64 time=0.536 ms
64 bytes from 192.168.60.111: icmp_seq=18 ttl=64 time=0.575 ms
64 bytes from 192.168.60.111: icmp_seq=19 ttl=64 time=0.430 ms
64 bytes from 192.168.60.111: icmp_seq=20 ttl=64 time=0.757 ms
64 bytes from 192.168.60.111: icmp_seq=21 ttl=64 time=0.414 ms
64 bytes from 192.168.60.111: icmp_seq=22 ttl=64 time=0.570 ms
```

Task 23:

To fix the firewall rules such that we can SSH from Client A to Server A. Add the following rules to the firewall.

```
sudo iptables -I INPUT -s 192.168.60.111 -p tcp -m tcp --dport 22 -j ACCEPT
```

```
ats@serverA:~$ ats@clientA:~$ sudo ssh ats@192.168.60.100
[sudo] password for ats:
ats@192.168.60.100's password:
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-104-generic x86_64)

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

416 packages can be updated.
136 updates are security updates.

Last login: Sun Mar 25 23:12:31 2018 from 192.168.60.1
ats@serverA:~$
```

```
Server A (Snapshot 1) [Running] - Oracle VM VirtualBox
Client A [Running] - Oracle VM VirtualBox
ats@serverA:~$ 0 packages can be updated.
0 updates are security updates.

Last login: Thu Nov 17 00:51:35 2016 from 10.0.99.2
ats@clientA:~$ sudo ssh -p 10022 ats@192.168.60.100
[sudo] password for ats:
ssh: Connect to host 192.168.60.100 port 10022: Connection refused
ats@clientA:~$ sudo ssh -p 22 ats@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:W+LPjhGRAjAU6ZmmVMzlgjvytXF4mC2eXKlDqKC505U.
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.
ats@192.168.60.100's password:
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-47-generic x86_64)

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

523 packages can be updated.
274 updates are security updates.

Last login: Sun Nov 26 17:13:57 2017 from 10.0.98.2
ats@serverA:~$
```

For ssh I used below command.

Sudo ssh -p 22 ats@192.168.60.100

Task 24:

Forwarded the traffic to Server A by following command.

```
Gateway 192.168.60.100  
dns-nameservers 10.0.98.3
```

The screenshot shows two terminal windows side-by-side. The left window displays the command `netstat -4 -rn` which lists the Kernel IP routing table. The right window shows the `/etc/network/interfaces` file being edited in `GNU nano 2.5.3`. The configuration includes a loopback interface and a host-only interface `enp0s3` with static IP settings, and a gateway entry pointing to the specified address.

```
ats@clientA:~$ netstat -4 -rn  
Kernel IP routing table  
Destination     Gateway         Genmask        Flags   MSS Window irtt Iface  
0.0.0.0         192.168.60.100  0.0.0.0        UG        0 0          0 enp0s3  
169.254.0.0     0.0.0.0        255.255.0.0   U         0 0          0 enp0s3  
192.168.60.0    0.0.0.0        255.255.255.0 U         0 0          0 enp0s3  
ats@clientA:~$  
  
ats@clientA:~$ nano /etc/network/interfaces  
# interfaces(5) file used by ifup(8) and ifdown(8)  
auto lo  
iface lo inet loopback  
  
# Host-only interface  
auto enp0s3  
iface enp0s3 inet static  
address 192.168.60.111  
netmask 255.255.255.0  
  
gateway 192.168.60.100  
dns-nameservers 10.0.98.3
```

Task 25:

Follow these command :

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

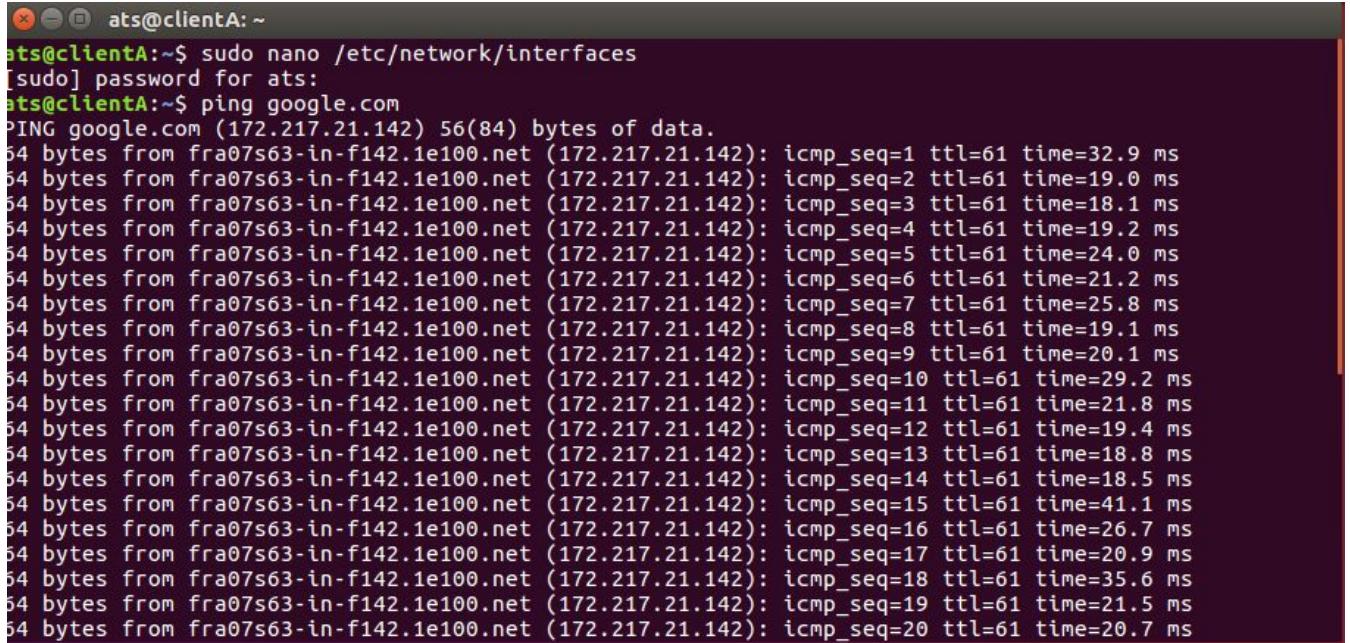
Task 26 & 27:

Using Server A as a router to forward traffic from outside to Client A I have used below command-----

```
sudo iptables -t filter -A FORWARD -i $HIF -j ACCEPT
```

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

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



```
ats@clientA:~$ sudo nano /etc/network/interfaces
[sudo] password for ats:
ats@clientA:~$ ping google.com
PING google.com (172.217.21.142) 56(84) bytes of data.
64 bytes from fra07s63-in-f142.1e100.net (172.217.21.142): icmp_seq=1 ttl=61 time=32.9 ms
64 bytes from fra07s63-in-f142.1e100.net (172.217.21.142): icmp_seq=2 ttl=61 time=19.0 ms
64 bytes from fra07s63-in-f142.1e100.net (172.217.21.142): icmp_seq=3 ttl=61 time=18.1 ms
64 bytes from fra07s63-in-f142.1e100.net (172.217.21.142): icmp_seq=4 ttl=61 time=19.2 ms
64 bytes from fra07s63-in-f142.1e100.net (172.217.21.142): icmp_seq=5 ttl=61 time=24.0 ms
64 bytes from fra07s63-in-f142.1e100.net (172.217.21.142): icmp_seq=6 ttl=61 time=21.2 ms
64 bytes from fra07s63-in-f142.1e100.net (172.217.21.142): icmp_seq=7 ttl=61 time=25.8 ms
64 bytes from fra07s63-in-f142.1e100.net (172.217.21.142): icmp_seq=8 ttl=61 time=19.1 ms
64 bytes from fra07s63-in-f142.1e100.net (172.217.21.142): icmp_seq=9 ttl=61 time=20.1 ms
64 bytes from fra07s63-in-f142.1e100.net (172.217.21.142): icmp_seq=10 ttl=61 time=29.2 ms
64 bytes from fra07s63-in-f142.1e100.net (172.217.21.142): icmp_seq=11 ttl=61 time=21.8 ms
64 bytes from fra07s63-in-f142.1e100.net (172.217.21.142): icmp_seq=12 ttl=61 time=19.4 ms
64 bytes from fra07s63-in-f142.1e100.net (172.217.21.142): icmp_seq=13 ttl=61 time=18.8 ms
64 bytes from fra07s63-in-f142.1e100.net (172.217.21.142): icmp_seq=14 ttl=61 time=18.5 ms
64 bytes from fra07s63-in-f142.1e100.net (172.217.21.142): icmp_seq=15 ttl=61 time=41.1 ms
64 bytes from fra07s63-in-f142.1e100.net (172.217.21.142): icmp_seq=16 ttl=61 time=26.7 ms
64 bytes from fra07s63-in-f142.1e100.net (172.217.21.142): icmp_seq=17 ttl=61 time=20.9 ms
64 bytes from fra07s63-in-f142.1e100.net (172.217.21.142): icmp_seq=18 ttl=61 time=35.6 ms
64 bytes from fra07s63-in-f142.1e100.net (172.217.21.142): icmp_seq=19 ttl=61 time=21.5 ms
64 bytes from fra07s63-in-f142.1e100.net (172.217.21.142): icmp_seq=20 ttl=61 time=20.7 ms
```

I am getting ping and Client A is connected to the world.

```

#!/bin/sh

IPT=/sbin/iptables

# NAT interface
NIF=enp0s9
# NAT IP address
NIP='10.0.98.100'
# Host-only interface
HIF=enp0s3
# Host-only IP address
HIP='192.168.60.100'
# DNS nameserver
NS='10.0.98.3'

## Reset the firewall to an empty, but friendly state
# Flush all chains in FILTER table
$IPT -t filter -F
# Delete any user-defined chains in FILTER table
$IPT -t filter -X
# Flush all chains in NAT table
$IPT -t nat -F
# Delete any user-defined chains in NAT table$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 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: Guest OS can view HTTP and HTTPS pages, but apache2 server is blocked from serving HTTP content.
$IPT -A INPUT -p tcp --dport 80 -j REJECT
#Task 15: Change default firewall policy to DROP
$IPT -t filter -P INPUT DROP
$IPT -t filter -P OUTPUT DROP
$IPT -t filter -P FORWARD DROP

#Task 17: Enable traffic from loopback interface
$IPT -A INPUT -i lo -j ACCEPT$IPT -A OUTPUT -o lo -j ACCEPT
#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
#Task 19: Allow Server A to ping all hosts
$IPT -A OUTPUT -j ACCEPT
$IPT -A INPUT -j ACCEPT
$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
#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#Task 23: SSH from Client A to Server A
$IPT -A INPUT -p tcp -s 192.168.60.111 --dport 22 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT

```

```
#Task 26: Change 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
#Task 27: Enable SNAT on Server A
$IPT -t nat -A POSTROUTING -j SNAT -o $NIF --to $NIP
# 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 drop: "
$IPT -t filter -A output_log -j LOG --log-level notice --log-prefix "output drop: "
$IPT -t filter -A forward_log -j LOG --log-level notice --log-prefix "forward drop: "
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
# 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
```

Thanks

Saddam Hossen

