

LAB-1 Networking and Firewalls

By

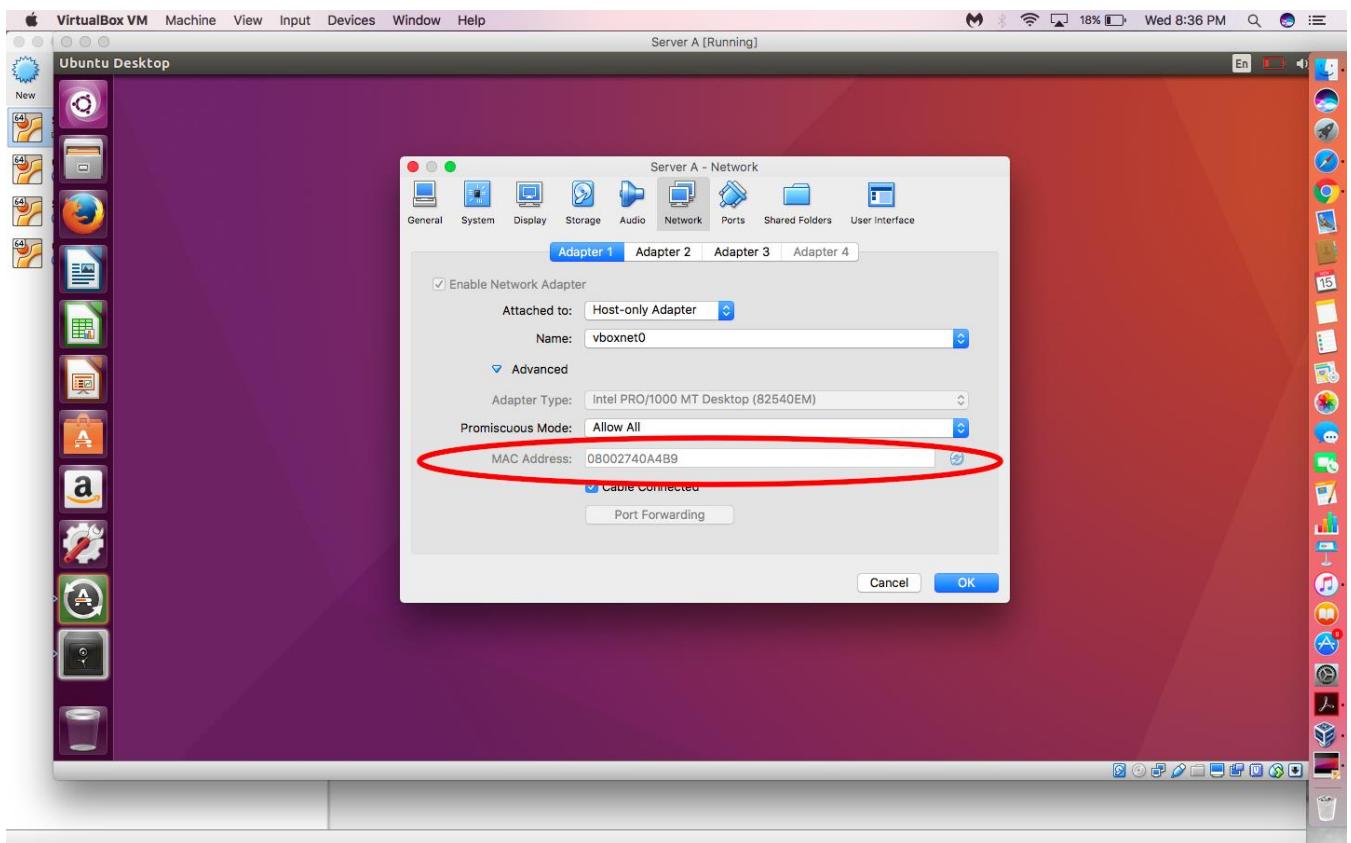
Monica Tamanampudi

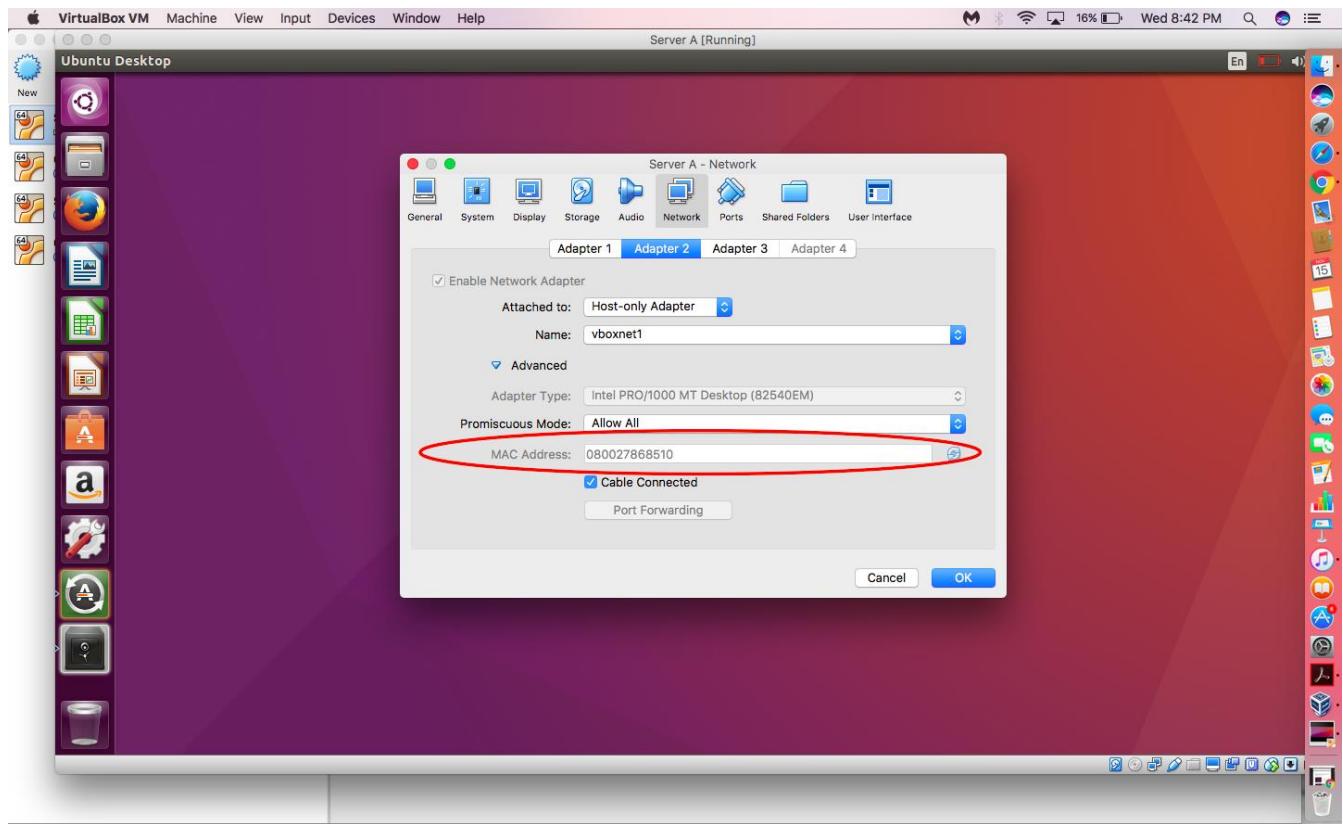
mota17@student.bth.se

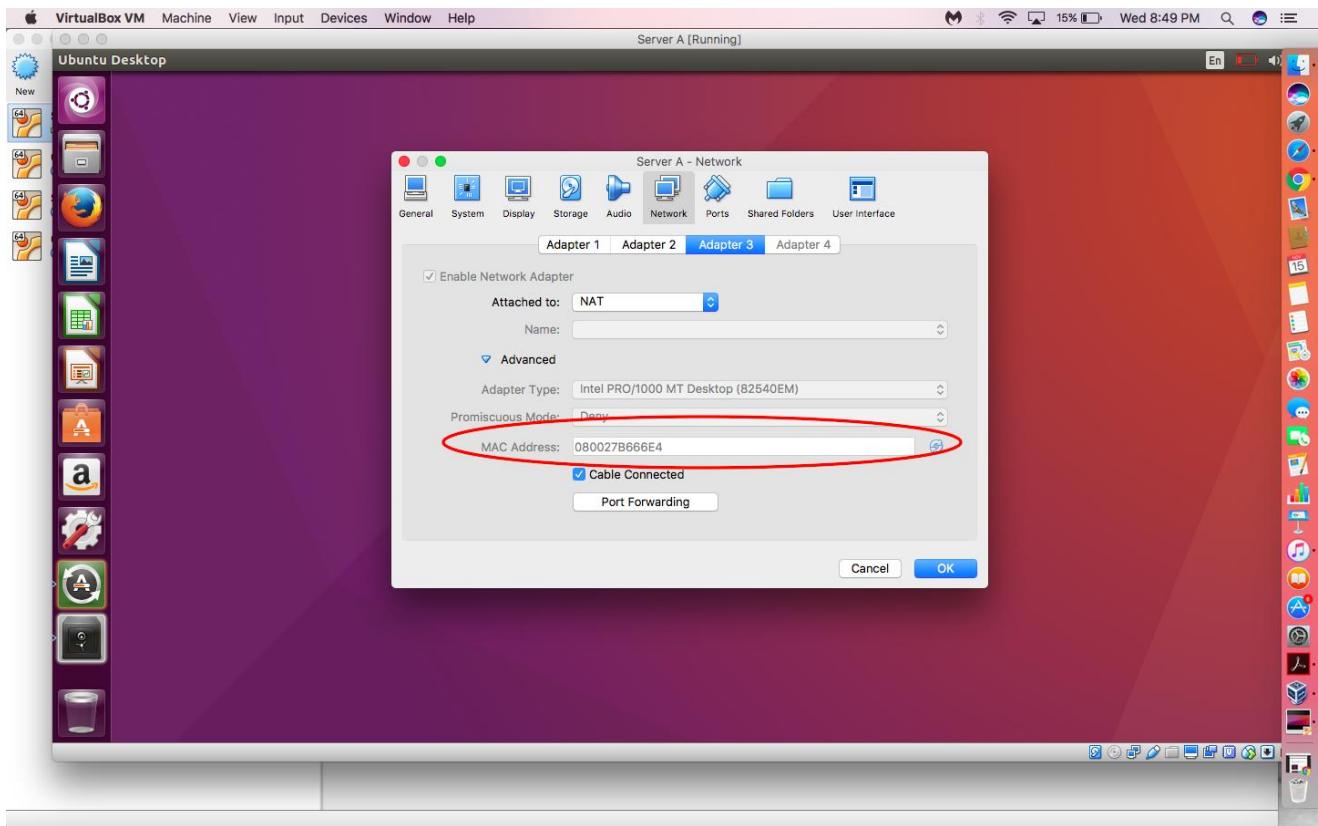
P No 960207-4180

Task 1:

To find out the MAC address of the configured adapters in the web server VM. The figures below show the mac address.



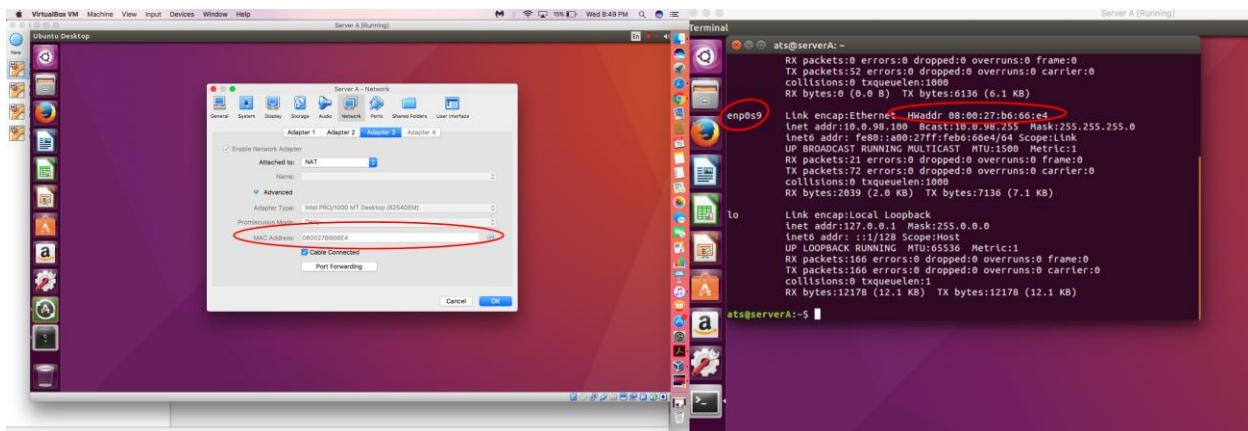
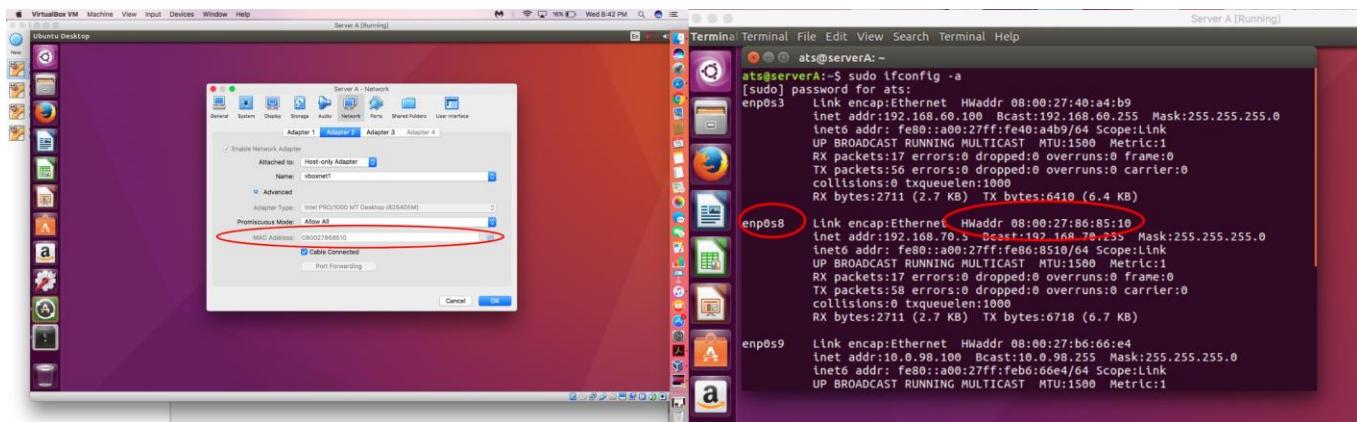
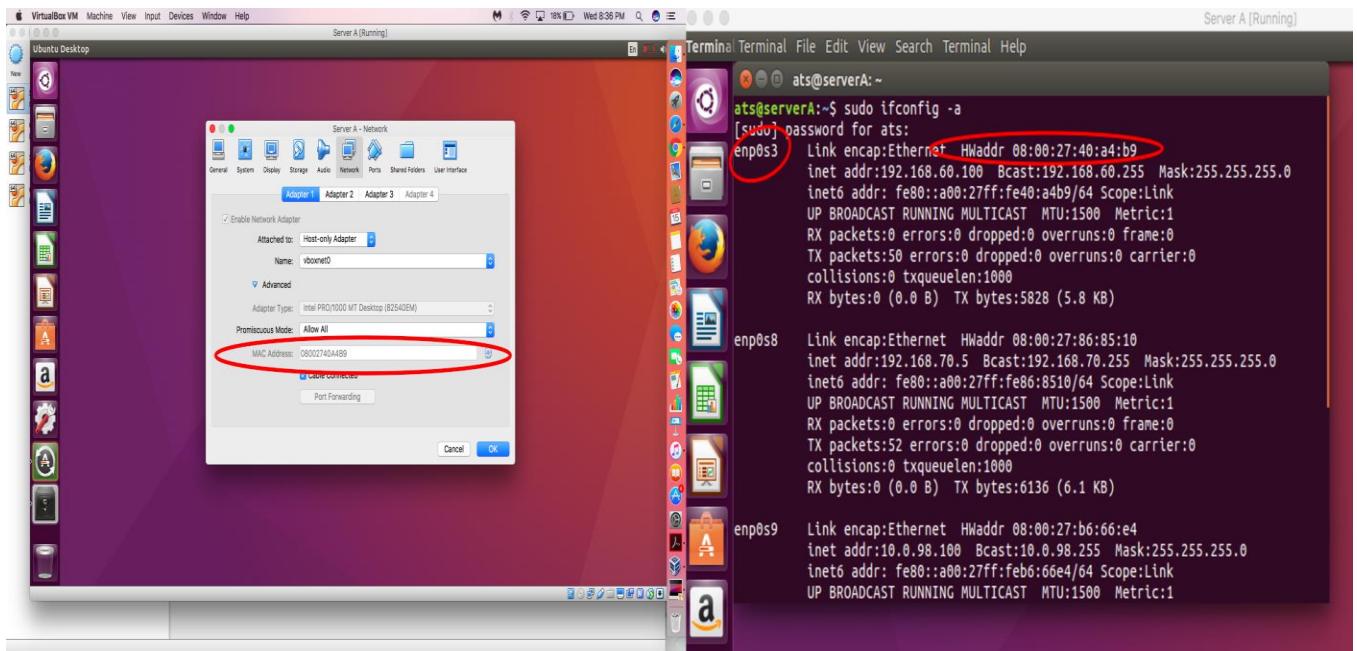




Task 2:

We need to identify which is the NAT interface and which are the host-only interfaces by using the MAC addresses. Enter the following command in the terminal of server A to get the list of available interfaces.

“Sudo ifconfig –a “



Task 3:

The picture below shows the network address corresponding to their IP address in order to find out the network address of each interface associated with their IP address and their netmask

enp0s3

inet add: 192.168.60.100 → 11000000.10101000.00111100.01100100

subnet mask: 255.255.255.0 → 11111111.11111111.11111111.00000000

bitwise AND: 192.168.60.0 → 11000000.10101000.00111100.00000000

Network Address - 192.168.60.0

enp0s8

inet add: 192.168.70.5 → 11000000.10101000.01000110.00000101

subnet mask: 255.255.255.0 → 11111111.11111111.11111111.00000000

bitwise AND: 192.168.70.0 → 11000000.10101000.01000110.00000000

Network Address - 192.168.70.0

enp0s9

inet add: 10.0.98.100 → 00001010.00000000.01100010.01100100

net mask: 255.255.255.0 → 11111111.11111111.11111111.00000000

bitwise AND: 10.0.98.0 → 00001010.00000000.01100010.00000000

Network Address - 10.0.98.0

lo

inet add: 127.0.0.1 → 01111111.00000000.00000000.00000001

net mask: 255.255.255.0 → 11111111.11111111.11111111.00000000

bitwise AND: 127.0.0.0 → 01111111.00000000.00000000.00000000

Network Address: 127.0.0.0

Task 4:

To find out the host-only interfaces in the HOST OS. By entering the following command in the terminal of HOST OS we can know the list of available interfaces and from the available interfaces we can find out the host-only interfaces (vboxnet0, vboxnet1, vboxnet 2)

ifconfig -a

```
option=128<0>RCVTIMEOUT,TXCSUM,TXSTATISTICS,SM_ISTAMP>
inet 127.0.0.1 netmask ff000000
inet6 ::1 prefixlen 128
inet 192.168.0.1 netmask feffff00 broadcast 192.168.0.255
inet6 fe80::1%eth0 prefixlen 64 scoprid 0x1
nd6 options=20<PERFRMRNUD,DAD>
gif0: flags=8010<POINTPOINT,MULTICAST> mtu 1280
stf0: flags=8000 mtu 1280
inet: flags=100<BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
ether c4:b3:01:0a:b2:22
inet6 fe80::c4b3:1ff:fe0a:b222%eth0 prefixlen 64 secured scoprid 0x4
inet 192.168.0.10 netmask ff000000 broadcast 192.168.0.125
nd6 options=20<PERFRMRNUD,DAD>
media: autoselect
status: active
ethtool: link: up speed 1000 duplex full
en1: flags=96400<BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX> mtu 1500
options=40<TSO4,TSO6>
ether 00:0c:29:bd:c9:99
inet 192.168.0.11 netmask ff000000 broadcast 192.168.0.125
nd6 options=20<PERFRMRNUD,DAD>
media: autoselect
status: inactive
p2p0: flags=8010<POINTPOINT,MULTICAST,RUNNING,SIMPLEX,MULTICAST> mtu 2304
ether 06:b3:01:0a:b2:1e
media: autoselect
status: inactive
awd10: flags=894400<BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1484
ether 00:ef:ef:29:8d:c9:99
inet 192.168.0.12 netmask ff000000 broadcast 192.168.0.125
nd6 options=20<PERFRMRNUD,DAD>
media: autoselect
status: inactive
bridge0: flags=883<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
options=63<RCVCSUM,TXCSUM,TS04,TS06>
ether 00:0c:29:00:02:1e
Configuration:
    id 0:0:8:0:8:0 priority 0 holdtime 0
    fddelay 0
    port 0 state xonxoff 100 timeout 1200
    root 0:0:8:0:8:0 priority 0 ifcost 0 port 0
    igfilter disabled Flags #x2
    member en1 flags=1A000000 DISCOVER<
        ifcost 0 priority 5 path cost 0
    nd6 options=20<PERFRMRNUD,DAD>
    media <unspecified type>
    status: inactive
utun0: flags=855<POINTPOINT,RUNNING,MULTICAST> mtu 2000
inet fe80::1:23ff:fe00:1234567890abcdef%utun0 prefixlen 64 scoprid 0x9
nd6 options=20<PERFRMRNUD,DAD>
boxnet0: flags=843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 158
ether 00:0c:29:00:00:00
inet 192.168.0.10 netmask ff000000 broadcast 192.168.0.255
boxnet1: flags=843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 158
ether 00:0c:27:00:00:01
inet 192.168.0.11 netmask ff000000 broadcast 192.168.0.255
boxnet2: flags=843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 158
ether 00:0c:27:00:00:02
inet 192.168.0.12 netmask ff000000 broadcast 192.168.0.255
Monicas-Air:~ monicatamananampudi$
```

```

Monicas-Air:~ monicatamanampudi$ ifconfig
utun0: flags=8051<UP,BROADCAST,RUNNING,MULTICAST> mtu 2000
    inet6 fe80::1e80:1321%utun0 brd fe80::fffe:1321%utun0 mtu 1280
        nd6 options=201<PERFORMNUD,DAD>
        media: <unknown type>
        status: inactive
vboxnet0: flags=8051<UP,BROADCAST,RUNNING,MULTICAST> mtu 2000
    inet6 fe80::1e80:1321%vboxnet0 brd fe80::fffe:1321%vboxnet0 mtu 1500
        nd6 options=201<PERFORMNUD,DAD>
        ether 0a:00:27:00:00:08
        inet 192.168.60.1 netmask 0xffffffff broadcast 192.168.60.255
vboxnet1: flags=8051<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether 0a:00:27:00:00:01
        inet 192.168.70.1 netmask 0xffffffff broadcast 192.168.70.255
vboxnet2: flags=8043<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether 0a:00:27:00:00:02
        inet 192.168.80.1 netmask 0xffffffff broadcast 192.168.80.255
Monicas-Air:~ monicatamanampudi$ 

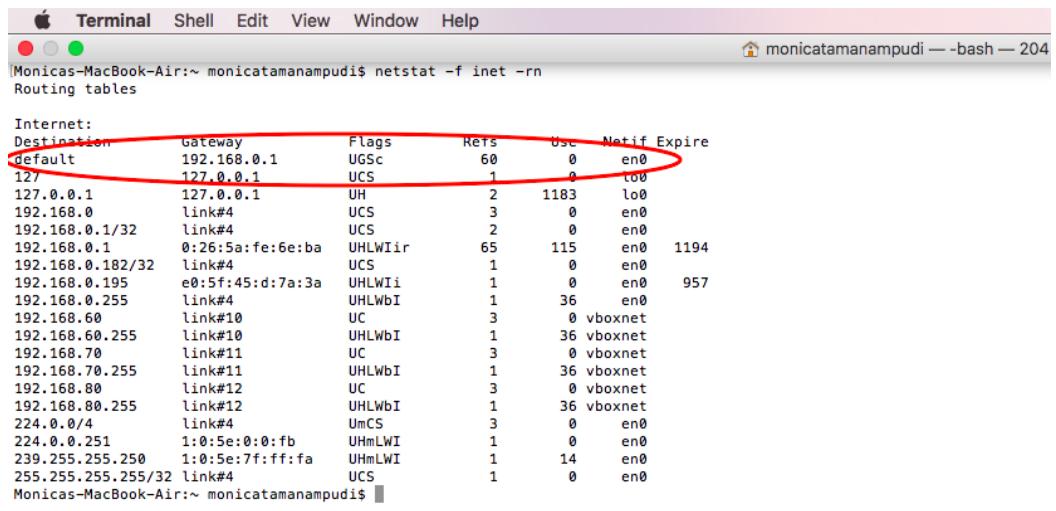
```

The screenshot shows a terminal window on the left displaying network interface configurations. On the right, there are two configuration dialog boxes for 'Adapter' and 'DHCP Server'. The top dialog is for interface 'vboxnet0' with IP 192.168.60.1. The bottom dialog is for interface 'vboxnet1' with IP 192.168.70.1. Both dialogs have fields for IPv4 Address, IPv4 Network Mask, IPv6 Address, and IPv6 Prefix Length. Red circles highlight the 'IPv4 Address' field in the terminal and the 'IPv4 Address' field in both configuration dialogs.

Task 5:

To identify over what interface, we can reach the default gateway for host. enter the following command in the terminal of the HOST OS we can view the routing table.

`netstat -f inet -rn`



```

Terminal Shell Edit View Window Help
monicas-MacBook-Air:~ monicatamanampudi$ netstat -f inet -rn
Routing tables

Internet:
Destination      Gateway        Flags   Rets   Use    Netif  Expire
default          192.168.0.1    UGSc    60     0      en0
127/8           127.0.0.1     UCS     1      0      lo0
127.0.0.1       127.0.0.1     UH      2     1183   lo0
192.168.0       link#4       UCS     3      0      en0
192.168.0.1/32  link#4       UCS     2      0      en0
192.168.0.1     0:26:5a:fe:6e:ba UHLWIir  65     115   en0    1194
192.168.0.182/32 link#4       UCS     1      0      en0
192.168.0.195   e0:5f:45:d:7a:3a UHLWii   1      0      en0    957
192.168.0.255   link#4       UHLWbI   1      36    en0
192.168.60       link#10      UC      3      0      vboxnet
192.168.60.255  link#10      UHLWbI   1      36    vboxnet
192.168.70       link#11      UC      3      0      vboxnet
192.168.70.255  link#11      UHLWbI   1      36    vboxnet
192.168.80       link#12      UC      3      0      vboxnet
192.168.80.255  link#12      UHLWbI   1      36    vboxnet
224.0.0/4        link#4       UmCS    3      0      en0
224.0.0.251     1:0:5e:0:0:fb UHmlWI   1      0      en0
239.255.255.250 1:0:5e:7:f:ff:fa UHmlWI   1      14    en0
255.255.255.255/32 link#4       UCS     1      0      en0
Monicas-MacBook-Air:~ monicatamanampudi$ 

```

Task 6: To identify the interface through which we can reach the default gateway for my host. By entering the following command in the terminal of guest OS we can view the routing table. The default gateway is 10.0.98.2.

“Netstat -4 -rn”

“Route -n”

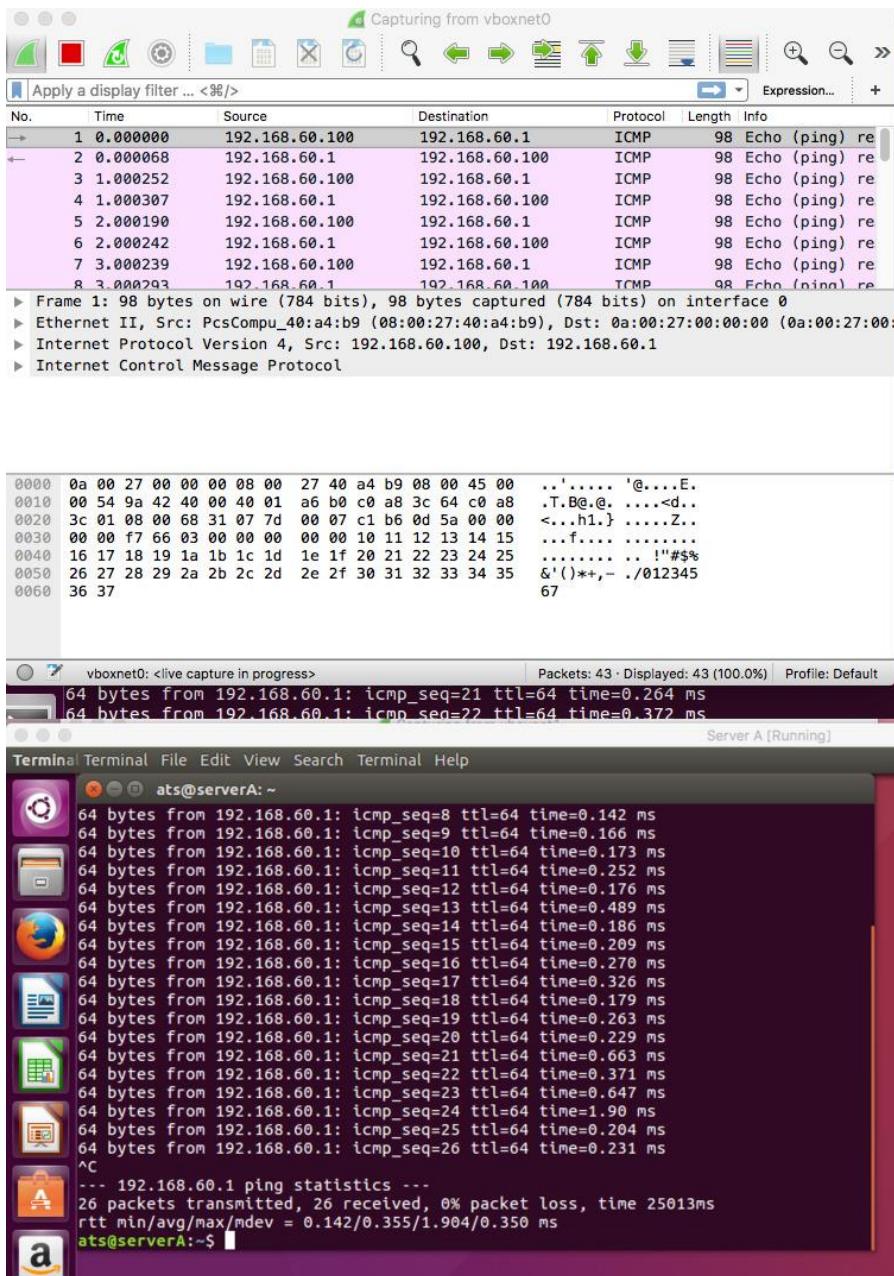
“Ip -4 route”

Ubuntu Desktop Server A [Running]

```
ats@serverA:~$ netstat -4 -rn
Kernel IP routing table
Destination      Gateway      Genmask      Flags   MSS Window irtt 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          0 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:~$ route -n
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
ats@serverA:~$
```

Task 7:

To ping the IP address corresponding to the host-only interface in the host OS and capture the packets in the Wireshark. To examine the icmp traffic



From this we can conclude that the ping and the Wireshark capturing of the icmp traffic are identical

Task 8:

To ssh into VM via localhost from the HOST OS. Enter the following command on the host

`ssh -p 10022 ats@localhost`

Server A - Network

Name	Protocol	Host IP	Host Port	Guest IP	Guest Port
ssh	TCP		10022		22

Cancel OK


```

monicatamanampudi — ats@serverA: ~ — ssh -p 10022 ats@localhost — 80x24
Last login: Thu Nov 16 18:52:22 on ttys000
[host-193-11-184-76:~ monicatamanampudi$ ssh -p 10022 ats@localhost
The authenticity of host '[localhost]:10022 ([127.0.0.1]:10022)' can't be established.
ECDSA key fingerprint is SHA256:W+LPjhGRAjAU6ZmmWMzlgjvytXF4mC2eXK1DqKC505U.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[localhost]:10022' (ECDSA) to the list of known hosts.
[ats@localhost's password:
Permission denied, please try again.
[ats@localhost'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

525 packages can be updated.
279 updates are security updates.

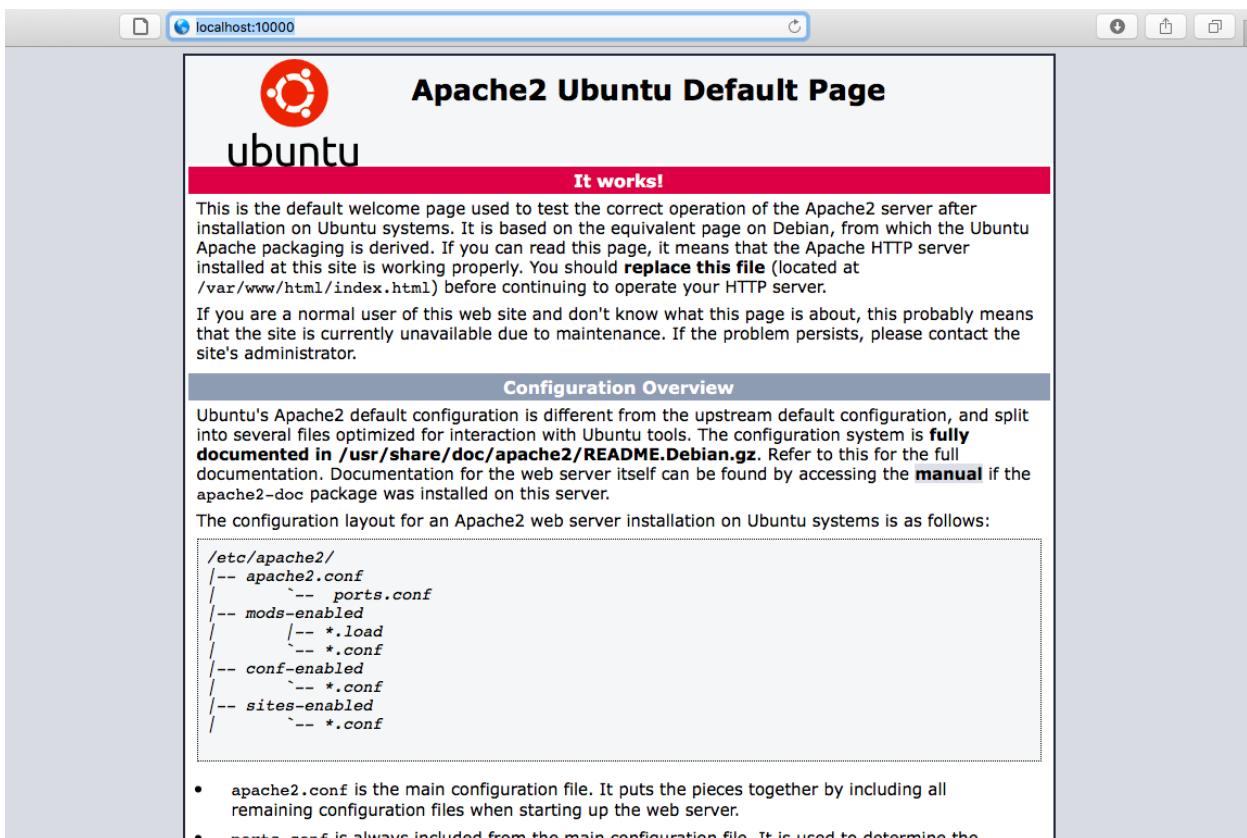
Last login: Thu Nov 17 00:51:35 2016 from 10.0.99.2
ats@serverA:~$ ]

```

Task 9:

To add the forwarding rules for HTTP and HTTPS in Virtual Box, so that the host user can view the HTTP and HTTPS content of the apache2 server in the guest OS. In order to do this we are using port 10000 for HTTP and 10001 for HTTPS in the host OS and forwarding these ports to the official ports for

HTTP (80) and HTTPS (443) of the guest OS. The images below show that the host OS can view the HTTP and HTTPS content of the apache2 server in the host OS.



The screenshot shows a web browser window displaying the Apache2 Ubuntu Default Page. The page features the Ubuntu logo and the text "ubuntu". A red banner at the top right says "It works!". Below it, a paragraph explains the purpose of the page: "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." Another paragraph below states: "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." A section titled "Configuration Overview" provides information about the configuration layout: "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 is listed as follows:

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

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

The screenshot shows the Oracle VM VirtualBox Manager interface. A window titled "Server A - Network" is open, showing the "Network" tab selected among other tabs like General, System, Display, Storage, Audio, Ports, Shared Folders, and User Interface. The table lists three network mappings:

Name	Protocol	Host IP	Host Port	Guest IP	Guest Port
HTTP	TCP		10000		80
HTTPS	TCP		10001		443
SSH	TCP		10022		22

At the bottom of the window are "Cancel" and "OK" buttons. Below this window, another "OK" button is visible.

Task 10:

The commands to view the default rules are.

“*sudo iptables -t filter -L*”

“*sudo iptables -t mangle -L*”

“*sudo iptables -t nat -L*”

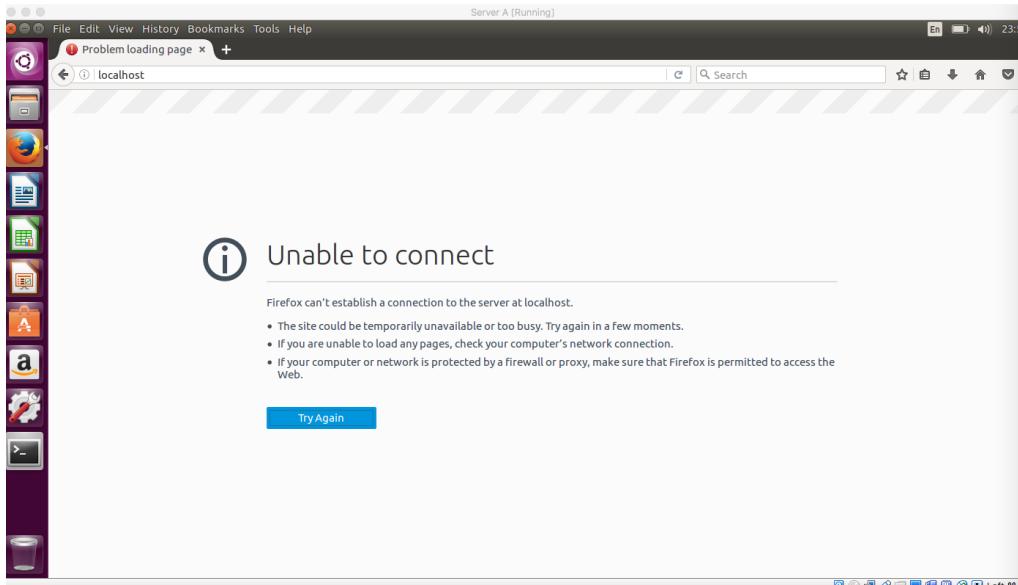
In the picture below the default policy and rules of the tables are shown (filter, mangle, nat)

```
Terminal File Edit View Search Terminal Help
sts@server:~$ sudo iptables -t filter -L
[sudo] password for sts:
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
sts@server:~$ 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
sts@server:~$ sudo iptables -t nat -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
sts@server:~$
```

Task 11:

To block the HTTP-browsing in the guest OS. To block the HTTP browsing we need to block the INPUT chain of the filter table for port number 80. The following command will block the HTTP browsing in the guest OS.

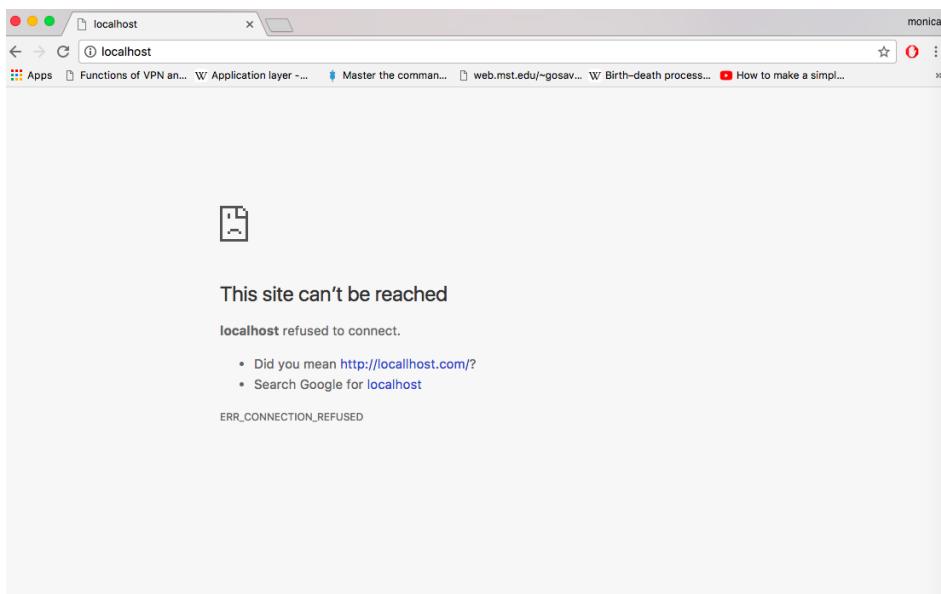
“*sudo iptables -A INPUT -p tcp --dport 80 -j REJECT*”



Task 12:

To block the HTTP-browsing in the host OS. To block the HTTP browsing we need to block the OUTPUT chain of the filter table for port number 80. The following command will block the HTTP browsing in the guest OS.

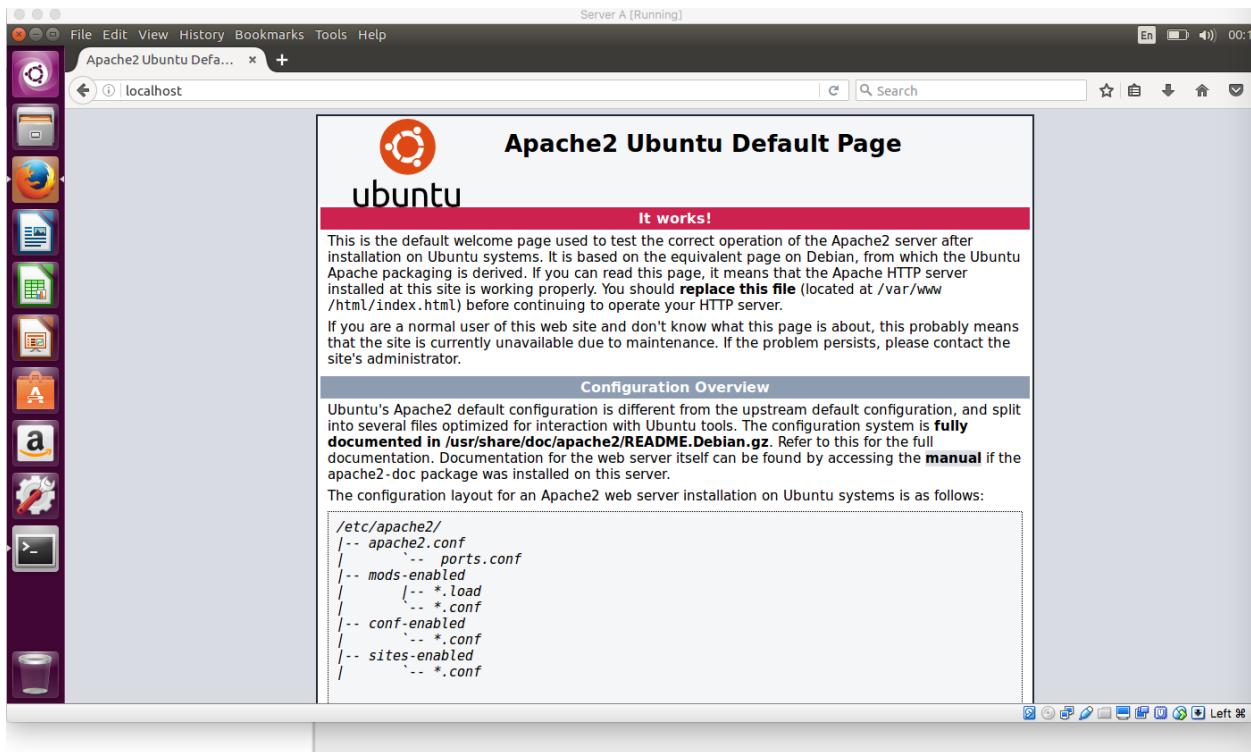
“*sudo iptables -A OUTPUT -p tcp --dport 80 -j REJECT*”



Task 13:

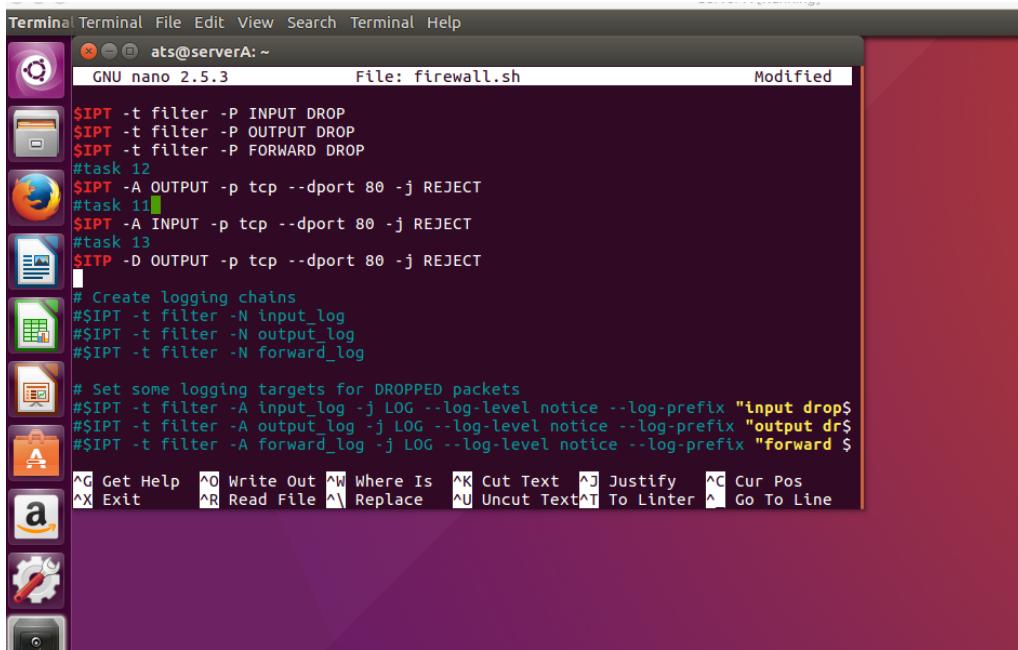
To unblock the HTTP-browsing in the guest OS. To block the HTTP browsing we need to block the INPUT chain of the filter table for port number 80. The following command will unblock the HTTP browsing in the guest OS.

“sudo iptables -D INPUT -p tcp --dport 80 -j REJECT”



Task 14:

To modify the script `firewall.sh` to bring this server A firewall to the state we had in task 13 guest OS can view HTTP and HTTPS pages, but apache2 server is blocked from serving the HTTP content



A screenshot of a Linux desktop environment, likely Ubuntu, showing a terminal window titled "Terminal". The terminal window has a title bar with "Terminal" and "ats@serverA: ~". Below the title bar, it says "GNU nano 2.5.3" and "File: firewall.sh". The status bar at the top right shows "Modified". The main area of the terminal contains a shell script named "firewall.sh". The script includes comments like "#task 12", "#task 11", "#task 13", and "# Create logging chains". It also contains iptables commands such as "\$IPT -t filter -P INPUT DROP", "\$IPT -t filter -P OUTPUT DROP", and "\$IPT -t filter -P FORWARD DROP". The script ends with a section for logging targets and help information. The terminal window is part of a desktop interface with a purple background and various icons in the dock.

```
$IPT -t filter -P INPUT DROP
$IPT -t filter -P OUTPUT DROP
$IPT -t filter -P FORWARD DROP
#task 12
$IPT -A OUTPUT -p tcp --dport 80 -j REJECT
#task 11
$IPT -A INPUT -p tcp --dport 80 -j REJECT
#task 13
$IPT -D OUTPUT -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
#$IPT -t filter -A input_log -j LOG --log-level notice --log-prefix "input drop"
#$IPT -t filter -A output_log -j LOG --log-level notice --log-prefix "output dr"
#$IPT -t filter -A forward_log -j LOG --log-level notice --log-prefix "forward "

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

Task 15:

To change the default firewall policy to DROP. Add the following commands in the firewall.sh script and executing the script, the firewall policy will be changed to DROP.

“*\$IPT –P INPUT DROP*”
“*\$IPT –P OUTPUT DROP*”
“*\$IPT –P FORWARD DROP*”

```
Terminal Terminal File Edit View Search Terminal Help Server A [Running] En
ats@serverA: ~
t-wth icmp-port-unreachable
Chain FORWARD (policy DROP)
target  prot opt source          destination
Chain OUTPUT (policy DROP)
target  prot opt source          destination
ats@serverA:~$ nano firewall.sh
ats@serverA:~$ sudo ./firewall.sh
./firewall.sh: 49: ./firewall.sh: -D: not found
ats@serverA:~$ sudo iptables -L
Chain INPUT (policy DROP)
target  prot opt source          destination
REJECT  tcp  --  anywhere          anywhere      tcp dpt:http reject
t-wth icmp-port-unreachable

Chain FORWARD (policy DROP)
target  prot opt source          destination
Chain OUTPUT (policy DROP)
target  prot opt source          destination
REJECT  tcp  --  anywhere          anywhere      tcp dpt:http reject
ats@serverA:~$
```

Task 16:

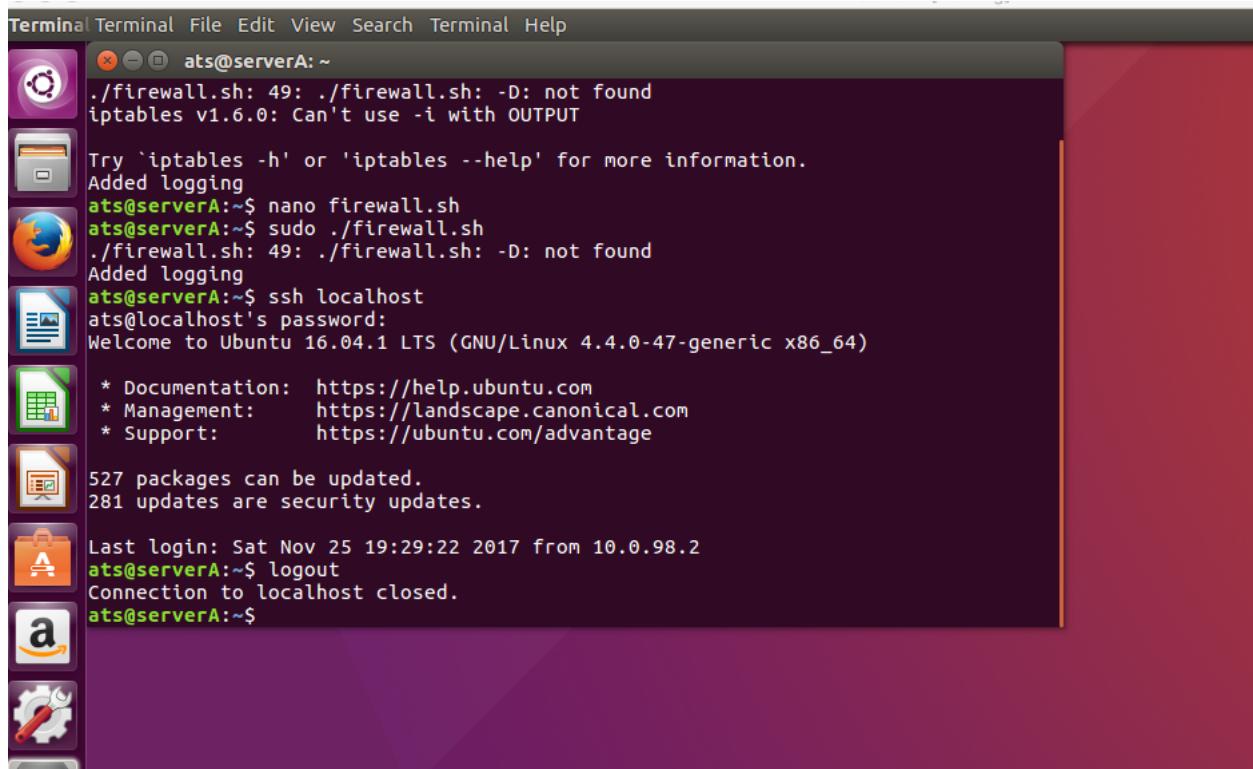
To see the live logs of linux kernel by entering the following command. In task 15 since we have set the default firewall policy to drop and so the packets are being dropped.

“`sudo tail -f /var/log/kern.log`”

Task 17:

To fix the firewall rules such that all type of traffic to and from loopback interface is enabled. By writing the following commands in the firewall script and executing it, we can enable the traffic for loopback interfaces.

```
“$IPT -A INPUT -i lo -j ACCEPT”
“$IPT -A OUTPUT -o lo -j ACCEPT”
```

A screenshot of a Ubuntu desktop environment. A terminal window is open in the top-left corner. The terminal window title bar says "Terminal". Inside the terminal, the user is at the prompt "ats@serverA:~". They run the command "./firewall.sh" which fails because it cannot find the "-D" option. It then shows the usage information for iptables. The user then runs "sudo ./firewall.sh" successfully, which adds logging rules and creates a nano editor file for the firewall script. They then run "ssh localhost" and log in with their password. The terminal shows they are connected to "Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-47-generic x86_64)". It also shows documentation links for help. Below the terminal, the desktop environment shows icons for Dash, Home, Applications, and Settings. The desktop background is a purple gradient.

```
Terminal Terminal File Edit View Search Terminal Help
ats@serverA: ~
./firewall.sh: 49: ./firewall.sh: -D: not found
iptables v1.6.0: Can't use -i with OUTPUT
Try `iptables -h` or `iptables --help` for more information.
Added logging
ats@serverA:~$ nano firewall.sh
ats@serverA:~$ sudo ./firewall.sh
./firewall.sh: 49: ./firewall.sh: -D: not found
Added logging
ats@serverA:~$ ssh localhost
ats@localhost'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

527 packages can be updated.
281 updates are security updates.

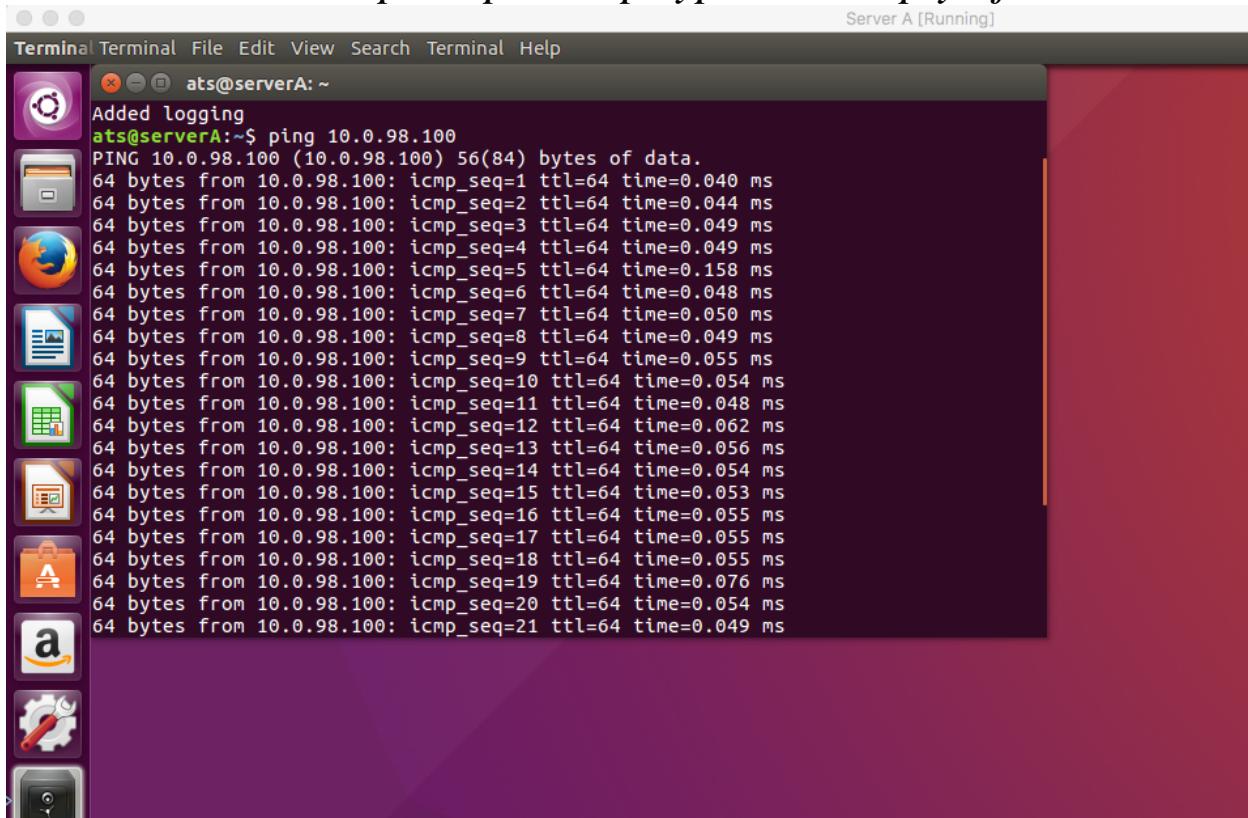
Last login: Sat Nov 25 19:29:22 2017 from 10.0.98.2
ats@serverA:~$ logout
Connection to localhost closed.
ats@serverA:~$
```

Task 18:

To allow ping traffic initiated from Server A. For ping traffic, we need to allow outgoing ICMP Echo Request and incoming ICMP Echo Reply messages. Enter the following commands.

“\$IPT -A OUTPUT -p icmp --icmp-type echo-request -j ACCEPT”

“\$IPT -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT”



The screenshot shows a Linux desktop environment with a terminal window open. The terminal window is titled "Server A [Running]" and contains the following text:

```
Terminal Terminal File Edit View Search Terminal Help
ats@serverA: ~
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=0.040 ms
64 bytes from 10.0.98.100: icmp_seq=2 ttl=64 time=0.044 ms
64 bytes from 10.0.98.100: icmp_seq=3 ttl=64 time=0.049 ms
64 bytes from 10.0.98.100: icmp_seq=4 ttl=64 time=0.049 ms
64 bytes from 10.0.98.100: icmp_seq=5 ttl=64 time=0.158 ms
64 bytes from 10.0.98.100: icmp_seq=6 ttl=64 time=0.048 ms
64 bytes from 10.0.98.100: icmp_seq=7 ttl=64 time=0.050 ms
64 bytes from 10.0.98.100: icmp_seq=8 ttl=64 time=0.049 ms
64 bytes from 10.0.98.100: icmp_seq=9 ttl=64 time=0.055 ms
64 bytes from 10.0.98.100: icmp_seq=10 ttl=64 time=0.054 ms
64 bytes from 10.0.98.100: icmp_seq=11 ttl=64 time=0.048 ms
64 bytes from 10.0.98.100: icmp_seq=12 ttl=64 time=0.062 ms
64 bytes from 10.0.98.100: icmp_seq=13 ttl=64 time=0.056 ms
64 bytes from 10.0.98.100: icmp_seq=14 ttl=64 time=0.054 ms
64 bytes from 10.0.98.100: icmp_seq=15 ttl=64 time=0.053 ms
64 bytes from 10.0.98.100: icmp_seq=16 ttl=64 time=0.055 ms
64 bytes from 10.0.98.100: icmp_seq=17 ttl=64 time=0.055 ms
64 bytes from 10.0.98.100: icmp_seq=18 ttl=64 time=0.055 ms
64 bytes from 10.0.98.100: icmp_seq=19 ttl=64 time=0.076 ms
64 bytes from 10.0.98.100: icmp_seq=20 ttl=64 time=0.054 ms
64 bytes from 10.0.98.100: icmp_seq=21 ttl=64 time=0.049 ms
```

Task 19:

To allow the server to ping all hosts. By adding the following rules to the firewall.sh script and executing it, we are allowing the firewall to accept the outgoing ICMP traffic to any server and corresponding ICMP replies.

“\$IPT -A OUTPUT -p udp -m conntrack --ctstate \NEW,ESTABLISHED -j ACCEPT”

“\$IPT -A INPUT -p udp -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT”

```

Terminal Terminal File Edit View Search Terminal Help
ats@serverA:~$ ping google.com
PING google.com (172.217.22.174) 56(84) bytes of data.
64 bytes from arn09s11-in-f14.1e100.net (172.217.22.174): icmp_seq=1 ttl=63 time
=21.4 ms
64 bytes from arn09s11-in-f14.1e100.net (172.217.22.174): icmp_seq=2 ttl=63 time
=22.1 ms
64 bytes from arn09s11-in-f14.1e100.net (172.217.22.174): icmp_seq=3 ttl=63 time
=21.8 ms
64 bytes from arn09s11-in-f14.1e100.net (172.217.22.174): icmp_seq=4 ttl=63 time
=21.9 ms
64 bytes from arn09s11-in-f14.1e100.net (172.217.22.174): icmp_seq=5 ttl=63 time
=22.1 ms
64 bytes from arn09s11-in-f14.1e100.net (172.217.22.174): icmp_seq=6 ttl=63 time
=21.8 ms
64 bytes from arn09s11-in-f14.1e100.net (172.217.22.174): icmp_seq=7 ttl=63 time
=22.2 ms
64 bytes from arn09s11-in-f14.1e100.net (172.217.22.174): icmp_seq=8 ttl=63 time
=22.2 ms
64 bytes from arn09s11-in-f14.1e100.net (172.217.22.174): icmp_seq=9 ttl=63 time
=22.9 ms
64 bytes from arn09s11-in-f14.1e100.net (172.217.22.174): icmp_seq=10 ttl=63 tim
e=22.3 ms
64 bytes from arn09s11-in-f14.1e100.net (172.217.22.174): icmp_seq=11 ttl=63 tim
e=22.4 ms

Terminal Terminal File Edit View Search Terminal Help
ats@serverA:~$ nano 2.5.3      File: firewall.sh      Modified
GNU nano 2.5.3
$IPT -t filter -P INPUT DROP
$IPT -t filter -P OUTPUT DROP
$IPT -t filter -P FORWARD DROP
#task 12
$IPT -A OUTPUT -p tcp --dport 80 -j REJECT
#task 11
$IPT -A INPUT -p tcp --dport 80 -j REJECT
#task 13
$IPT -D OUTPUT -p tcp --dport 80 -j REJECT
#task 17
$IPT -A INPUT -i lo -j ACCEPT
$IPT -A OUTPUT -o lo -j ACCEPT
#task 18
$IPT -A OUTPUT -p icmp --icmp-type echo-request -j ACCEPT
$IPT -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT
#task 19
$IPT -A OUTPUT -p udp -m conntrack --ctstate \NEW,ESTABLISHED -j ACCEPT
$IPT -A INPUT -p udp -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

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

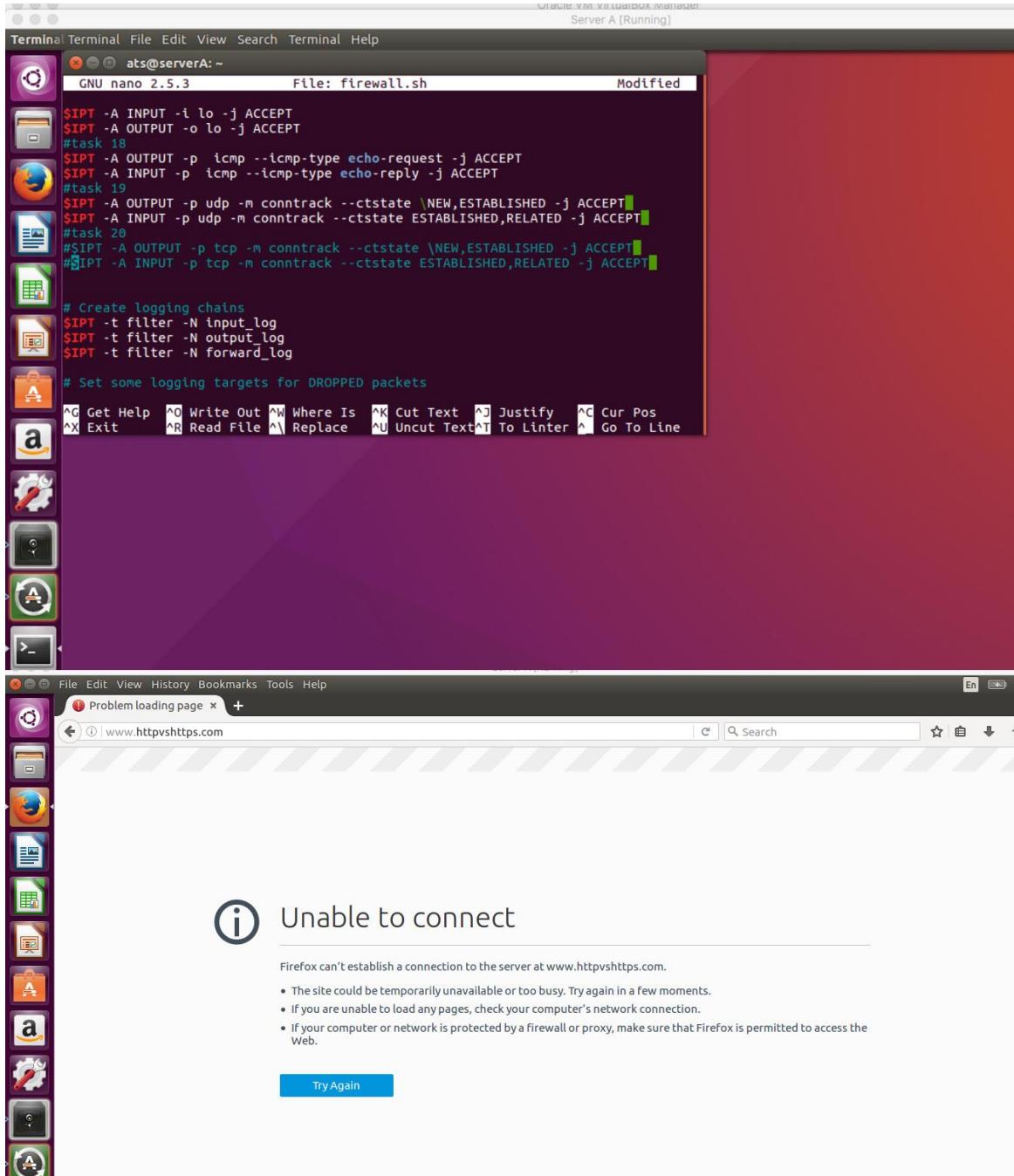
```

Task 20:

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.

```
“$IPT –A OUTPUT –p tcp –m conntrack --ctstate \NEW,ESTABLISHED –j ACCEPT ”
“$IPT –A INPUT –p tcp –m conntrack --ctstate ESTABLISHED,RELATED –j ACCEPT”
```

The images below show that the tcp connection is not established for the website by commenting the rules above (<http://www.httpvshttps.com/>)



The images below show that the tcp connection is established by adding the rules above.

Server A [Running]

The screenshot shows a Linux desktop environment with a purple gradient background. On the left, there is a vertical dock containing icons for various applications like a file manager, terminal, and system settings. In the center, there is a terminal window titled "Terminal" with the command "ats@serverA: ~" and the file "firewall.sh" open. The terminal contains several iptables commands, with the last two lines circled in red:

```
$IPT -A INPUT -p udp -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
#task 2
$IPT -A OUTPUT -p tcp -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
$IPT -A INPUT -p tcp -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
#task 2
$IPT -A INPUT -p tcp --dport 443 -j ACCEPT
$IPT -A INPUT -p tcp --dport 22 -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
$IPT -t filter -A input_log -j LOG --log-level notice --log-prefix "input drop:$
$IPT -t filter -A output_log -j LOG --log-level notice --log-prefix "output dro$ 
$IPT -t filter -A forward_log -j LOG --log-level notice --log-prefix "forward d$ 
echo "Added logging"
```

Below the terminal is a menu bar with options like File, Edit, View, Search, Terminal, Help, and a battery icon. At the bottom of the terminal window are keyboard shortcuts for various functions.

At the top of the screen, there is a title bar "Server A [Running]" and a status bar with battery and signal indicators.

Below the terminal, there is a web browser window titled "HTTP vs HTTPS Test". The URL "https://www.httpvshttps.com" is visible in the address bar. The page content includes a comparison between HTTP and HTTPS, stating "Encrypted Websites Protect Our Privacy and are Significantly Faster". It shows a large grid of green checkmarks representing test results. The HTTPS version is highlighted with a blue underline and a green "0.350 s" response time. A message at the bottom says "Done! Please try HTTP." Below the browser are social sharing buttons for Facebook, Twitter, and LinkedIn.

Task 21:

To Enable SSH and HTTPS content from apache2 server for web browser on HOST, add the commands below.

```
“$IPT -A INPUT -p tcp --dport 443 -j ACCEPT”
“$IPT -A INPUT -p tcp --dport 22 -j ACCEPT”
```

The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "Terminal" and the command line shows the user is at "ats@serverA: ~". Inside the terminal, the user has run the command "nano firewall.sh" to edit a script. The script contains several iptables rules and some logging configurations. A red oval highlights the line "# Task 21" which includes the rule "\$IPT -A INPUT -p tcp --dport 443 -j ACCEPT". Below the terminal, the desktop environment is visible with a taskbar showing icons for a browser, file manager, and system settings. The desktop background is a colorful gradient.

```
Terminal File Edit View Search Terminal Help
ats@serverA: ~
GNU nano 2.5.3      File: firewall.sh

$IPT -A INPUT -p udp -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
#task 20
$IPT -A OUTPUT -p tcp -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
$IPT -A INPUT -p tcp -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
#task 21
$IPT -A INPUT -p tcp --dport 443 -j ACCEPT
$IPT -A INPUT -p tcp -dport 22 -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
$IPT -t filter -A input_log -j LOG --log-level notice --log-prefix "input drop:$"
$IPT -t filter -A output_log -j LOG --log-level notice --log-prefix "output dro$
$IPT -t filter -A forward_log -j LOG --log-level notice --log-prefix "forward d
echo "Added logging"

# Get Help   Write Out  Where Is  Cut Text  Justify  Cur Pos
# Exit    Read File  Replace  Uncut Text  To Linter  Go To Line

[Monicas-Air:~ monicatamanampudi — ats@serverA: ~ — ssh -p 10022 ats@localhost — 80x23
[ats@localhost's password:
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-47-generic x86_64)

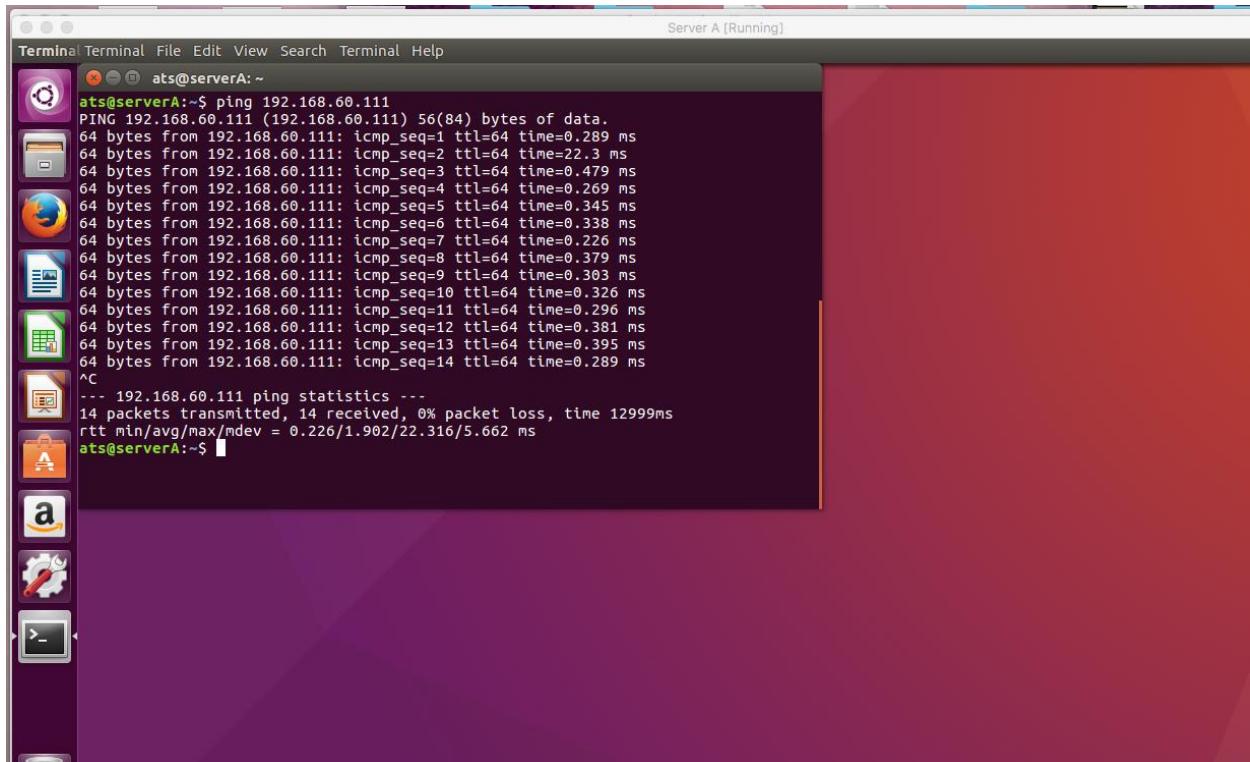
 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

527 packages can be updated.
281 updates are security updates.

Last login: Sat Nov 25 21:39:50 2017 from 127.0.0.1
ats@serverA:~$ ]
```

Task 22:

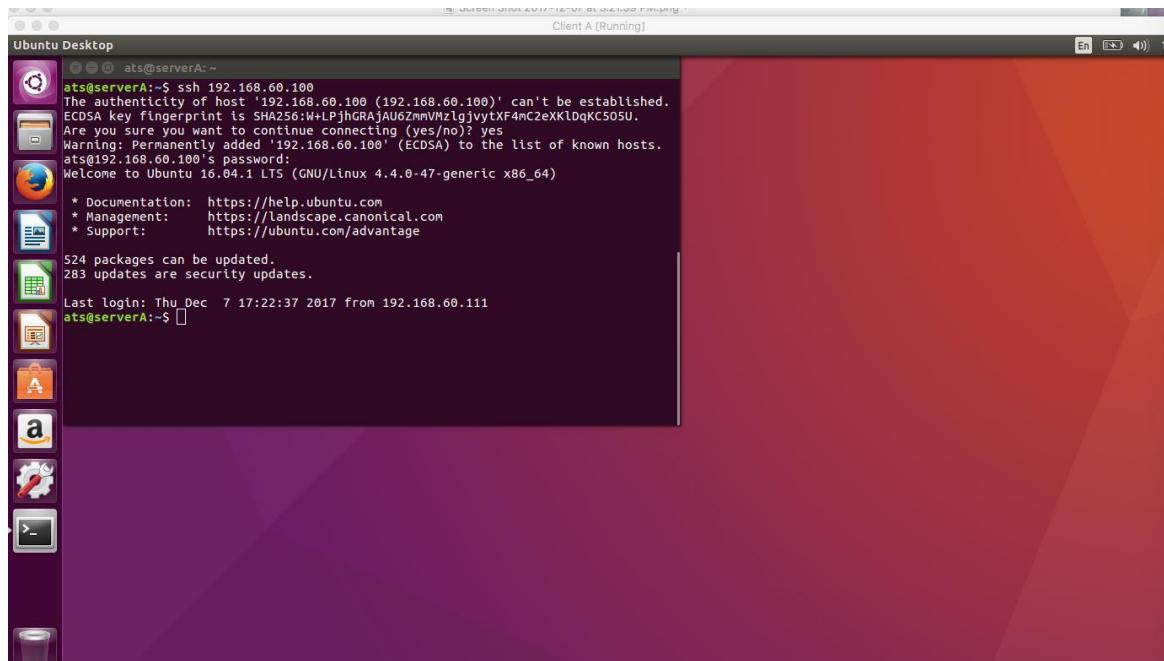
To ping server A from client A add rules in firewall.sh



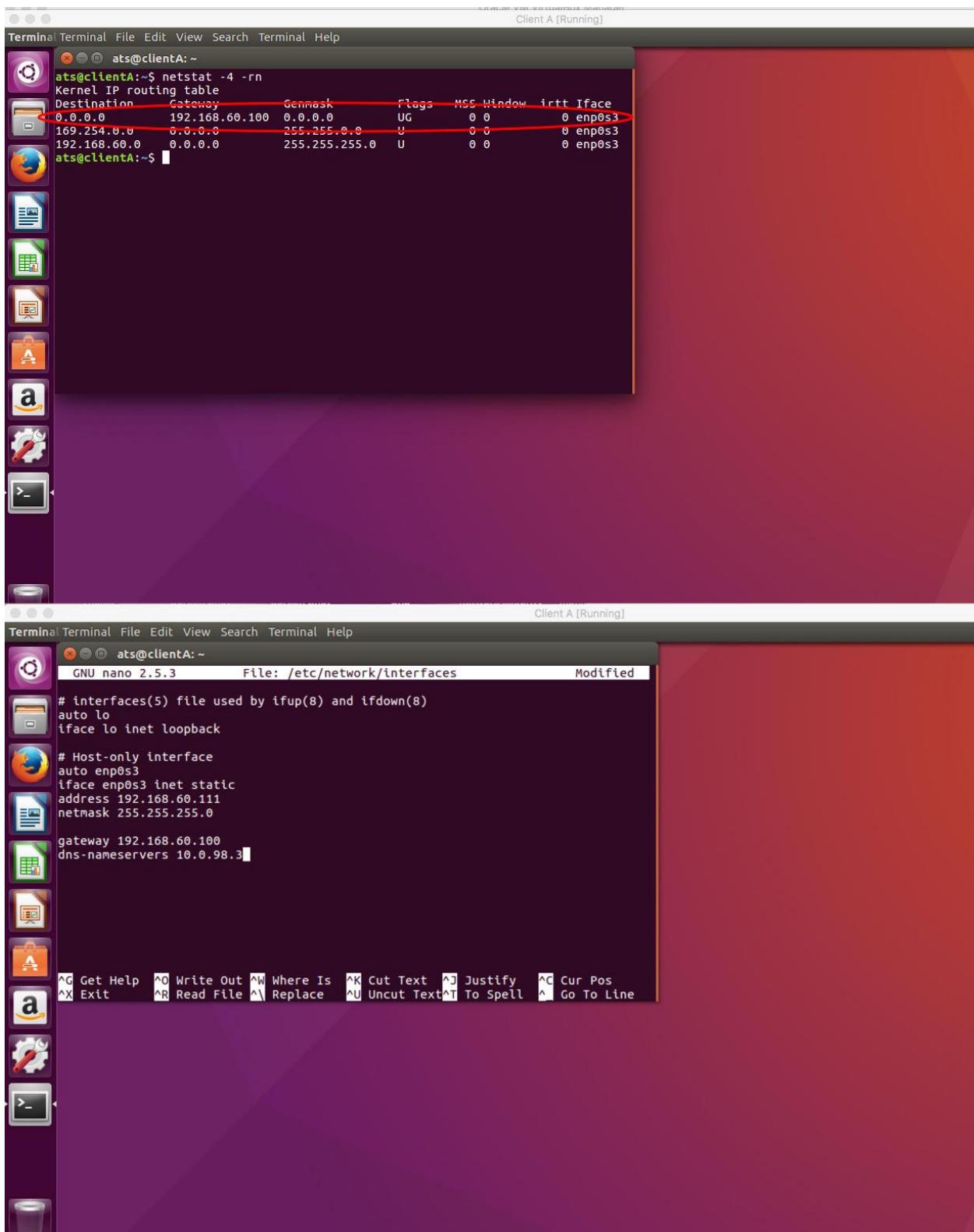
Task 23:

To SSH form server A to client A add the following rules

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



Task 24:
Add the gateway and dns-name server in
`/etc/network/interfaces` file in client A.



Task 25:

Execute the following command in the terminal of Server A so that IP forwarding is enabled on the Server A. This will forward the packets from enp0s3 to enp0s9

“*sudo sysctl -w net.ipv4.ip_forward=1*”

“*sudo sysctl -p*”

Task 26:

Change the iptables rules to forward packets. add the following rules to forward packets from enp0s3 to enp0s9.

“*\$IPT -t filter -A FORWARD -i \$HIF -j ACCEPT*”

“*\$IPT -t filter -A FORWARD -i \$NIF -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT*”

And then the packets are forwarded to the NAT interface. But the problem is that Client A uses private address (192.168.60.111) and all routers will have a basic default rule to drop packets coming from the private addresses. So we need to inform Server A to use NAT (more specifically Source NAT - SNAT). To do this we need to enable the SNAT on Server A.

Task 27:

To fix the problem in the above task you need to inform Server A to do SNAT on the NAT interface and add the following iptables rule.

“*\$IPT -t nat -A POSTROUTING -j SNAT -o \$NIF --to \$NIP*”

