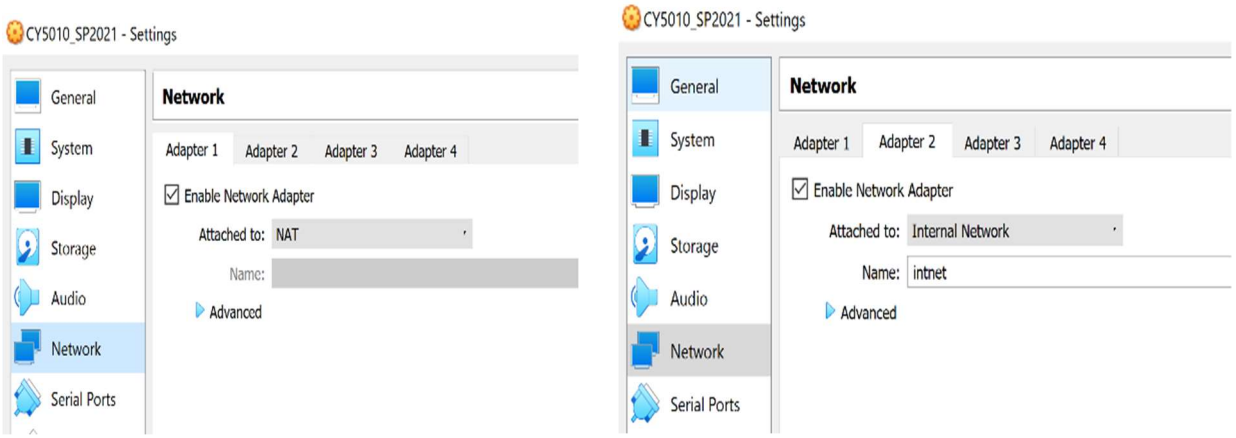
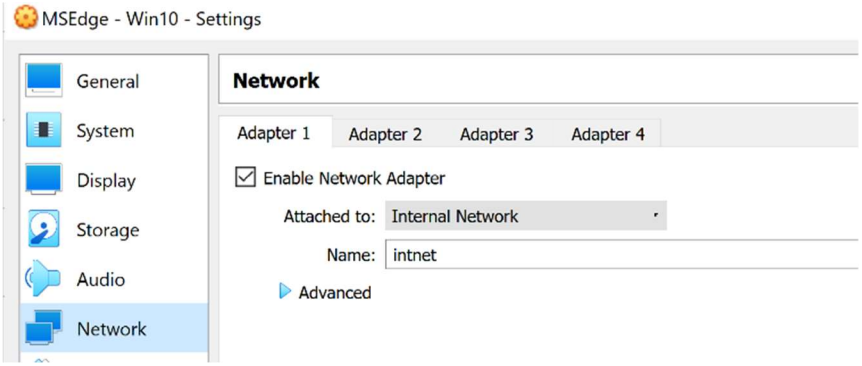
Lab 4: Network Basics - Linux Gateway, DNS and Firewall

Setup:

In Linux VM, we configured 2 network interfaces in VirtualBox.
Adapter 1 as NAT.
Adapter 2 as Internal Network named intnet.



In Windows VM, we configured 1 network interface only.
Adapter 1 as internal network named intnet as well.



Part 1 – Network Configuration

1. In Linux VM, we setup a network adapter enp0s8 to a static IP address 10.0.100.5/24. Used the same .yaml file in /etc/netplan i.e 01-netcfg.yaml file.

```
01-netcfg.yaml
user@ubuntu:/etc/netplan$ cat 01-netcfg.yaml
# This file describes the network interfaces available on your system
# For more information, see netplan(5).
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: yes
    enp0s8:
      addresses: [10.0.100.5/24]
user@ubuntu:/etc/netplan$ _
```

a. We restarted the VM to make sure our changes persisted. Then used ifconfig to verify.

```
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fe10:4d9e  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:10:4d:9e  txqueuelen 1000  (Ethernet)
        RX packets 11  bytes 2180 (2.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 23  bytes 2574 (2.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.100.5  netmask 255.255.255.0  broadcast 10.0.100.255
        inet6 fe80::a00:27ff:fec8:946f  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:c8:94:6f  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 10  bytes 796 (796.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

b. We then verified that our linux VM can still access the internet by pinging google.com and that we were able to ssh into the master server.

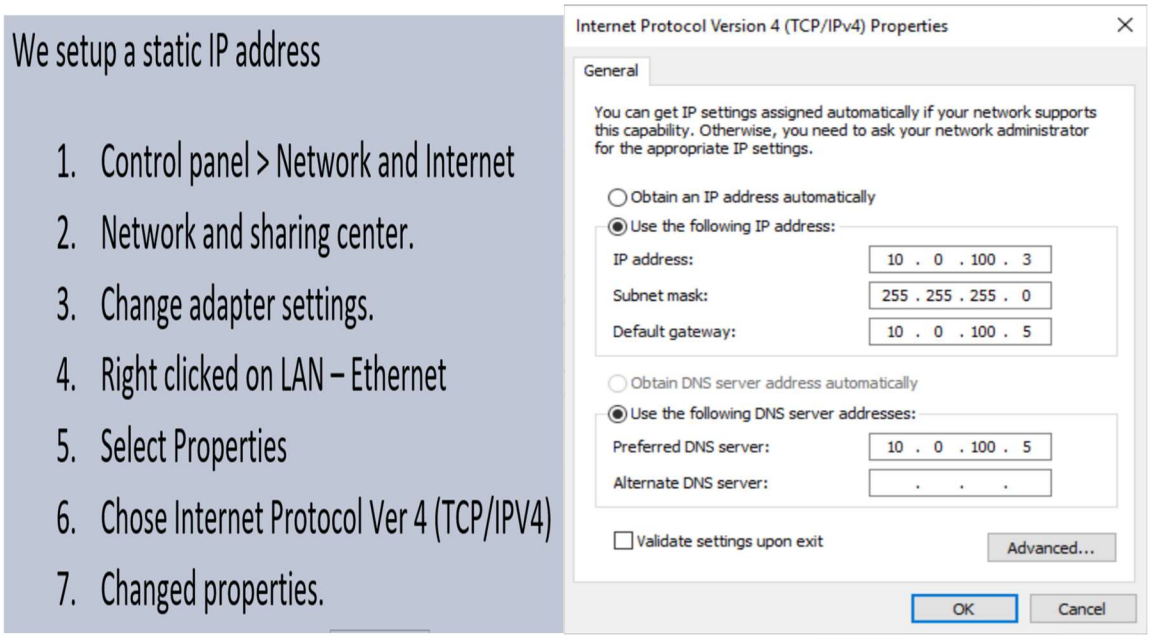
```
root@ubuntu:/home/user# ping google.com
PING google.com (172.217.10.238) 56(84) bytes of data.
64 bytes from lga25s59-in-f14.1e100.net (172.217.10.238): icmp_seq=1 ttl=115 time=79.0 ms
64 bytes from lga25s59-in-f14.1e100.net (172.217.10.238): icmp_seq=2 ttl=115 time=114 ms
64 bytes from lga25s59-in-f14.1e100.net (172.217.10.238): icmp_seq=3 ttl=115 time=58.8 ms
64 bytes from lga25s59-in-f14.1e100.net (172.217.10.238): icmp_seq=4 ttl=115 time=60.5 ms
64 bytes from lga25s59-in-f14.1e100.net (172.217.10.238): icmp_seq=5 ttl=115 time=74.4 ms
^C
--- google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4056ms
rtt min/avg/max/mdev = 58.826/77.486/114.563/20.108 ms
root@ubuntu:/home/user#
```

```
user@ubuntu:~$ ssh -p 17001 team8@cy5010.ccs.neu.edu
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-126-generic x86_64)

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

  System information as of Sun Feb 28 20:00:52 EST 2021

  System load:  0.0               Processes:             102
  Usage of /:   40.9% of 15.68GB  Users logged in:       0
  Memory usage: 25%               IP address for ens160: 10.0.10.140
  Swap usage:   2%                IP address for docker0: 172.17.0.1

 * Introducing self-healing high availability clusters in MicroK8s.
   Simple, hardened, Kubernetes for production, from RaspberryPi to DC.

     https://microk8s.io/high-availability

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

25 packages can be updated.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable

New release '20.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


*** System restart required ***
Last login: Sat Feb 27 21:27:34 2021 from 10.15.185.27
team8@SP21CY5010master:~$
```

2. Enabled IP forwarding to allow Linux VM act as a router, routing packets between NAT and internal network interfaces.

```
user@ubuntu:~$ sudo sysctl -w net.ipv4.ip_forward=1
[sudo] password for user:
net.ipv4.ip_forward = 1
user@ubuntu:~$
user@ubuntu:~$
```

3. In Windows VM, we configured the static IP address to 10.0.100.3. We used the setup process on the left image to configure the IP and DNS addresses on the right image.



Part 2: DNS Configuration

1. Installed Bind9 on the VM using the following commands:

```
sudo apt-get update
sudo apt-get install bind9 bind9utils bind9-doc
```

2. Configured bind9 as the caching DNS server by editing the named.conf.options file. We included 8.8.8.8 as the DNS address and restarted the bind service.

```
user@ubuntu:/etc/bind$ cat named.conf.options
options {
        directory "/var/cache/bind";

        // If there is a firewall between you and nameservers you want
        // to talk to, you may need to fix the firewall to allow multiple
        // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

        // If your ISP provided one or more IP addresses for stable
        // nameservers, you probably want to use them as forwarders.
        // Uncomment the following block, and insert the addresses replacing
        // the all-0's placeholder.

        forwarders {
                8.8.8.8;
        };

        //========================================================================
        // If BIND logs error messages about the root key being expired,
        // you will need to update your keys.  See https://www.isc.org/bind-keys
        //========================================================================
        dnssec-validation auto;

        auth-nxdomain no;    # conform to RFC1035
        listen-on-v6 { any; };
};

user@ubuntu:/etc/bind$
```

3. We tested it by running the “dig northeastern.edu” command. As shown below the time elapsed on the first query is longer than the second query. First query took 17msec while second query took 0 msec.

```
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;northeastern.edu.              IN      A

;; ANSWER SECTION:
northeastern.edu.       3600    IN      A       155.33.17.68

;; Query time: 17 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Tue Mar 02 07:47:23 EST 2021
;; MSG SIZE  rcvd: 61

user@ubuntu:/etc/bind$ sudo service bind9 restart
user@ubuntu:/etc/bind$ dig northeastern.edu

; <<>> DiG 9.11.3-1ubuntu1.14-Ubuntu <<>> northeastern.edu
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 30201
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;northeastern.edu.              IN      A

;; ANSWER SECTION:
northeastern.edu.       3071    IN      A       155.33.17.68

;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Tue Mar 02 07:56:12 EST 2021
;; MSG SIZE  rcvd: 61

user@ubuntu:/etc/bind$ _
```

Part 3 - Setup Routing

1. Route all traffic from the internal network to NAT network.

```
user@ubuntu:/$ sudo iptables -A FORWARD -i enp0s8 -o enp0s3 -j ACCEPT
user@ubuntu:/$
user@ubuntu:/$
```

2. Route only RELATED and ESTABLISHED connections from NAT to the internal network.

```
user@ubuntu:/$
user@ubuntu:/$ sudo iptables -A FORWARD -i enps03 -o enps08 -m state --state RELATED,ESTABLISHED -j
ACCEPT
user@ubuntu:/$
```

3. Perform IP masquerading for all traffic leaving the NAT network.

```
user@ubuntu:/$
user@ubuntu:/$ sudo iptables -t nat -A POSTROUTING -o enps03 -j MASQUERADE
user@ubuntu:/$
```

a. We verified that we could still connect to the internet from our linux VM.

```
root@ubuntu:/home/user# ping google.com
PING google.com (172.217.10.110) 56(84) bytes of data.
64 bytes from lga34s15-in-f14.1e100.net (172.217.10.110): icmp_seq=1 ttl=115 time=26.9 ms
64 bytes from lga34s15-in-f14.1e100.net (172.217.10.110): icmp_seq=2 ttl=115 time=23.6 ms
64 bytes from lga34s15-in-f14.1e100.net (172.217.10.110): icmp_seq=3 ttl=115 time=42.9 ms
^C
--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2045ms
rtt min/avg/max/mdev = 23.655/31.193/42.977/8.442 ms
root@ubuntu:/home/user#
```

b. We verified that we could still ssh into the master server.

```
user@ubuntu:/$ sudo iptables -A FORWARD -i enps03 -o enps08 -m state --state RELATED,ESTABLISHED -j
ACCEPT
user@ubuntu:/$ sudo iptables -t nat -A POSTROUTING -o enps03 -j MASQUERADE
user@ubuntu:/$ ssh team8@cy5010.ccs.neu.edu -p 17001
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-126-generic x86_64)

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

  System information as of Tue Mar  2 12:06:35 EST 2021

  System load:  0.17              Processes:             107
  Usage of /:   41.0% of 15.68GB  Users logged in:       1
  Memory usage: 25%               IP address for ens160: 10.0.10.140
  Swap usage:   2%                IP address for docker0: 172.17.0.1

 * Introducing self-healing high availability clusters in MicroK8s.
   Simple, hardened, Kubernetes for production, from RaspberryPi to DC.

     https://microk8s.io/high-availability

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

25 packages can be updated.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable

New release '20.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


*** System restart required ***
Last login: Tue Mar  2 12:05:34 2021 from 10.15.185.190
team8@SP21CY5010master:~$ _
```

Part 4 – Setup Firewall

1. We added additional rules to route traffic from internal network to NAT ONLY when the destination is SSH (17001/tcp), DNS (53/udp), HTTPS (443/tcp) and drop other packets by default.

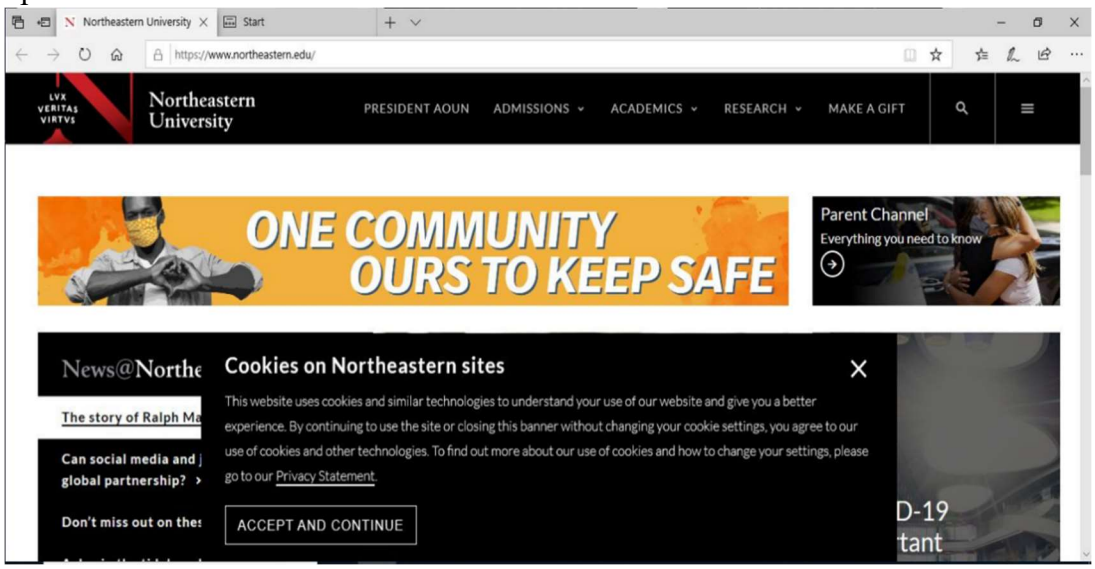
```
root@ubuntu:/home/user# iptables -A FORWARD -i enp0s8 -p tcp -m tcp --dport 17001 -j ACCEPT
root@ubuntu:/home/user# iptables -A FORWARD -i enp0s8 -p udp -m udp --dport 53 -j ACCEPT
root@ubuntu:/home/user# iptables -A FORWARD -i enp0s8 -p tcp -m tcp --dport 443 -j ACCEPT
root@ubuntu:/home/user# iptables -P FORWARD DROP
```

a. Our iptables looked as the figure below with the new rules implemented.

```
root@ubuntu:/home/user# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere             state RELATED,ESTABLISHED
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:17001
ACCEPT     udp  --  anywhere             anywhere             udp dpt:domain
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:https

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

Part 5 – Testing

1. To test the gateway and Windows VM setup were configured correctly, we successfully opened northeastern.edu website on the Windows VM as seen below.



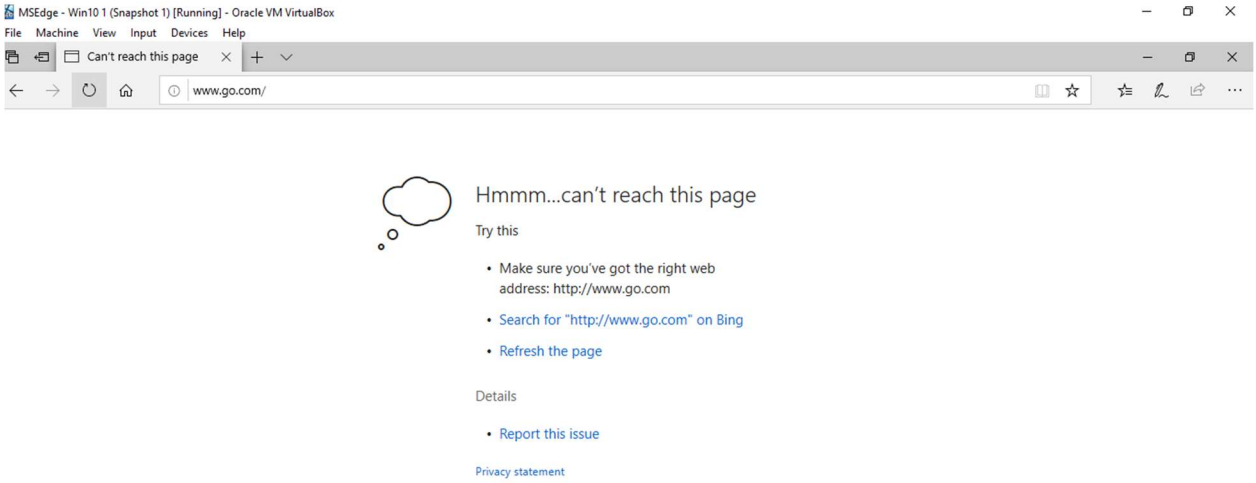
2. We tried to ping google.com and got request timed out since ICMP packets are dropped by default.

```
C:\Users\IEUser>ping google.com

Pinging google.com [172.217.10.110] with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 172.217.10.110:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Users\IEUser>
```

3. We successfully SSH into master server through the windows VM.

```
C:\Users\IEUser>ssh team8@cy5010.ccs.neu.edu -p 17001
team8@cy5010.ccs.neu.edu's password:
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-126-generic x86_64)

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

  System information as of Mon Mar  1 21:46:33 EST 2021

  System load:  0.03              Processes:             102
  Usage of /:   41.0% of 15.68GB  Users logged in:       0
  Memory usage: 27%               IP address for ens160:  10.0.10.140
  Swap usage:   2%                IP address for docker0: 172.17.0.1

 * Introducing self-healing high availability clusters in MicroK8s.
   Simple, hardened, Kubernetes for production, from RaspberryPi to DC.

     https://microk8s.io/high-availability

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

25 packages can be updated.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable

New release '20.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


*** System restart required ***
Last login: Mon Mar  1 20:53:41 2021 from 10.15.185.190
team8@SP21CY5010master:~$
```

4. We tried to connect to a http website (on port 80) and we were unable to do so.



5. We ran the tcpdump command on our Linux VM to listen to the incoming traffic on the internal interface (enp0s8) while we connected to northeastern.edu website from the windows VM. We can see the incoming packets with IP 10.0.100.3 i.e our Windows Vm.

```
root@ubuntu:/home/user# tcpdump -i enp0s8 port 443
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s8, link-type EN10MB (Ethernet), capture size 262144 bytes
20:41:55.780664 IP 10.0.100.3.55667 > lga34s30-in-f14.1e100.net.https: Flags [P.], seq 2419252036:24
19252433, ack 120000233, win 65535, length 397
20:41:55.781706 IP lga34s30-in-f14.1e100.net.https > 10.0.100.3.55667: Flags [.], ack 397, win 65535
, length 0
20:41:55.840387 IP lga34s30-in-f14.1e100.net.https > 10.0.100.3.55667: Flags [P.], seq 1:410, ack 39
7, win 65535, length 409
20:41:55.841265 IP 10.0.100.3.55667 > lga34s30-in-f14.1e100.net.https: Flags [.], ack 410, win 65535
, length 0
20:41:55.846074 IP lga34s30-in-f14.1e100.net.https > 10.0.100.3.55667: Flags [P.], seq 410:1840, ack
 397, win 65535, length 1430
20:41:55.846792 IP 10.0.100.3.55667 > lga34s30-in-f14.1e100.net.https: Flags [.], ack 1840, win 6553
5, length 0
20:41:55.847056 IP lga34s30-in-f14.1e100.net.https > 10.0.100.3.55667: Flags [P.], seq 1840:4005, ac
k 397, win 65535, length 2165
20:41:55.847853 IP 10.0.100.3.55667 > lga34s30-in-f14.1e100.net.https: Flags [.], ack 4005, win 6553
5, length 0
20:41:55.869577 IP lga34s30-in-f14.1e100.net.https > 10.0.100.3.55667: Flags [P.], seq 4005:4051, ac
k 397, win 65535, length 46
20:41:55.870594 IP 10.0.100.3.55667 > lga34s30-in-f14.1e100.net.https: Flags [.], ack 4051, win 6553
5, length 0
20:41:56.048206 IP 10.0.100.3.55667 > lga34s30-in-f14.1e100.net.https: Flags [P.], seq 397:443, ack
4051, win 65535, length 46
20:41:56.049285 IP lga34s30-in-f14.1e100.net.https > 10.0.100.3.55667: Flags [.], ack 443, win 65535
, length 0
^C
12 packets captured
12 packets received by filter
0 packets dropped by kernel
root@ubuntu:/home/user#
```

Questions:

1. **Explain why was IP forwarding enabled on the Linux VM?**

   In Linux VM we used the following command to allow IP forwarding

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

   IP forwarding allowed the Linux VM to receive packets from the Windows VM and send packets to the Windows VM. This feature is needed for the Linux VM to act as the gateway between the internet and the Windows VM.

2. **Explain the rule you have used to route ESTABLISHED AND related TRAFFIC from NAT to the internal network. Why is it important for network security?**

   The rule we used was:
   "$iptables -A FORWARD -i enp0s3 -o enp0s8 -m state --state ESTABLISHED, RELATED -j ACCEPT"
   Established and Related belong to a list of possible states that a network packet can be. In this case established means that the packet is associated with an existing connection that has seen packets in both NAT and internal network. Related, is a packet that is starting a new connection while associated with an existing connection.
   Here the iptables uses the state of packets rule to determine whether to allow or not allow them. Thus, helping system and security administrators by reducing the amount of work to be done. This process is very similar to stateful inspection firewall.