

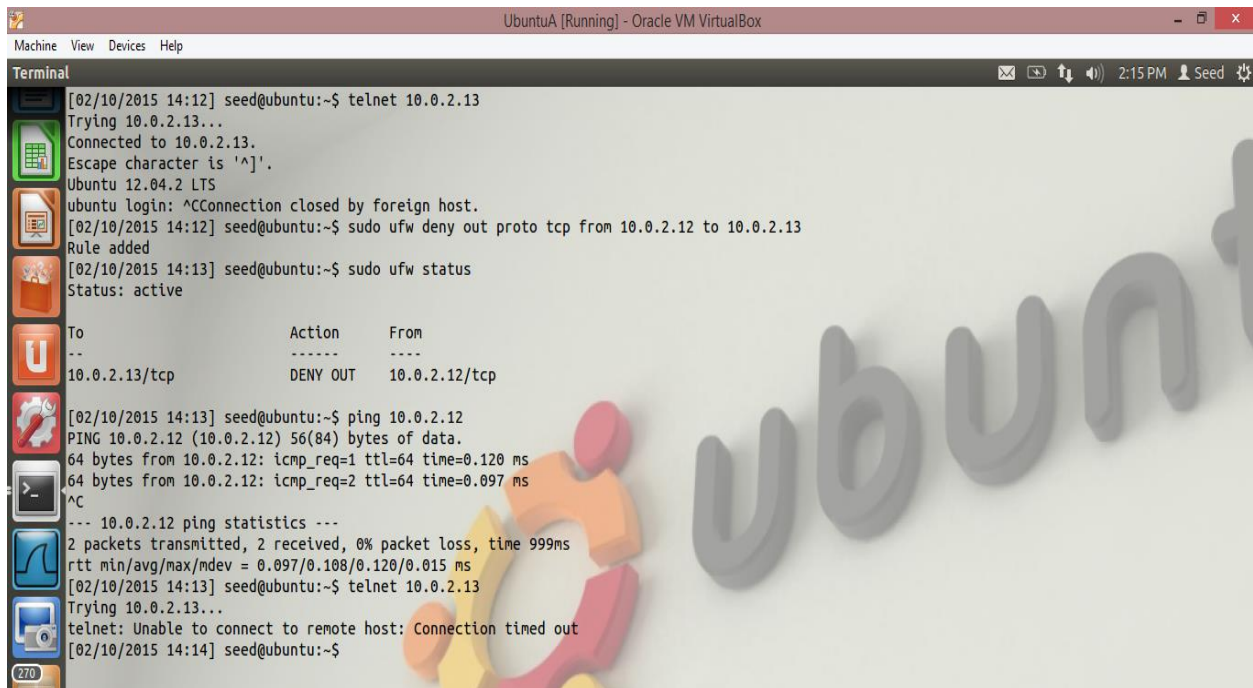
Internet Security - Linux Firewall Exploration Lab

Name – Abhishek Tripathi

SUID: 35081-6306

Task 1: Using Firewall

a) Block telnet

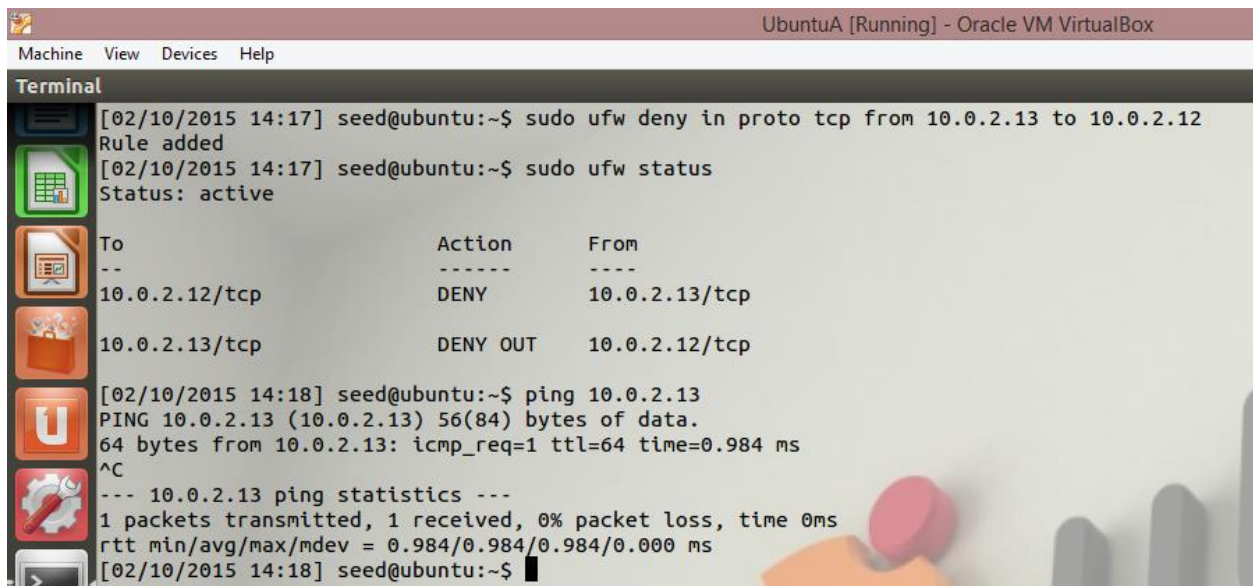


```
UbuntuA [Running] - Oracle VM VirtualBox
Machine View Devices Help
Terminal
[02/10/2015 14:12] seed@ubuntu:~$ telnet 10.0.2.13
Trying 10.0.2.13...
Connected to 10.0.2.13.
Escape character is '^]'.
Ubuntu 12.04.2 LTS
ubuntu login: ^CConnection closed by foreign host.
[02/10/2015 14:12] seed@ubuntu:~$ sudo ufw deny out proto tcp from 10.0.2.12 to 10.0.2.13
Rule added
[02/10/2015 14:13] seed@ubuntu:~$ sudo ufw status
Status: active

To Action From
--
10.0.2.13/tcp DENY OUT 10.0.2.12/tcp

[02/10/2015 14:13] seed@ubuntu:~$ ping 10.0.2.12
PING 10.0.2.12 (10.0.2.12) 56(84) bytes of data.
64 bytes from 10.0.2.12: icmp_req=1 ttl=64 time=0.120 ms
64 bytes from 10.0.2.12: icmp_req=2 ttl=64 time=0.097 ms
^C
--- 10.0.2.12 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.097/0.108/0.120/0.015 ms
[02/10/2015 14:13] seed@ubuntu:~$ telnet 10.0.2.13
Trying 10.0.2.13...
telnet: Unable to connect to remote host: Connection timed out
[02/10/2015 14:14] seed@ubuntu:~$
```

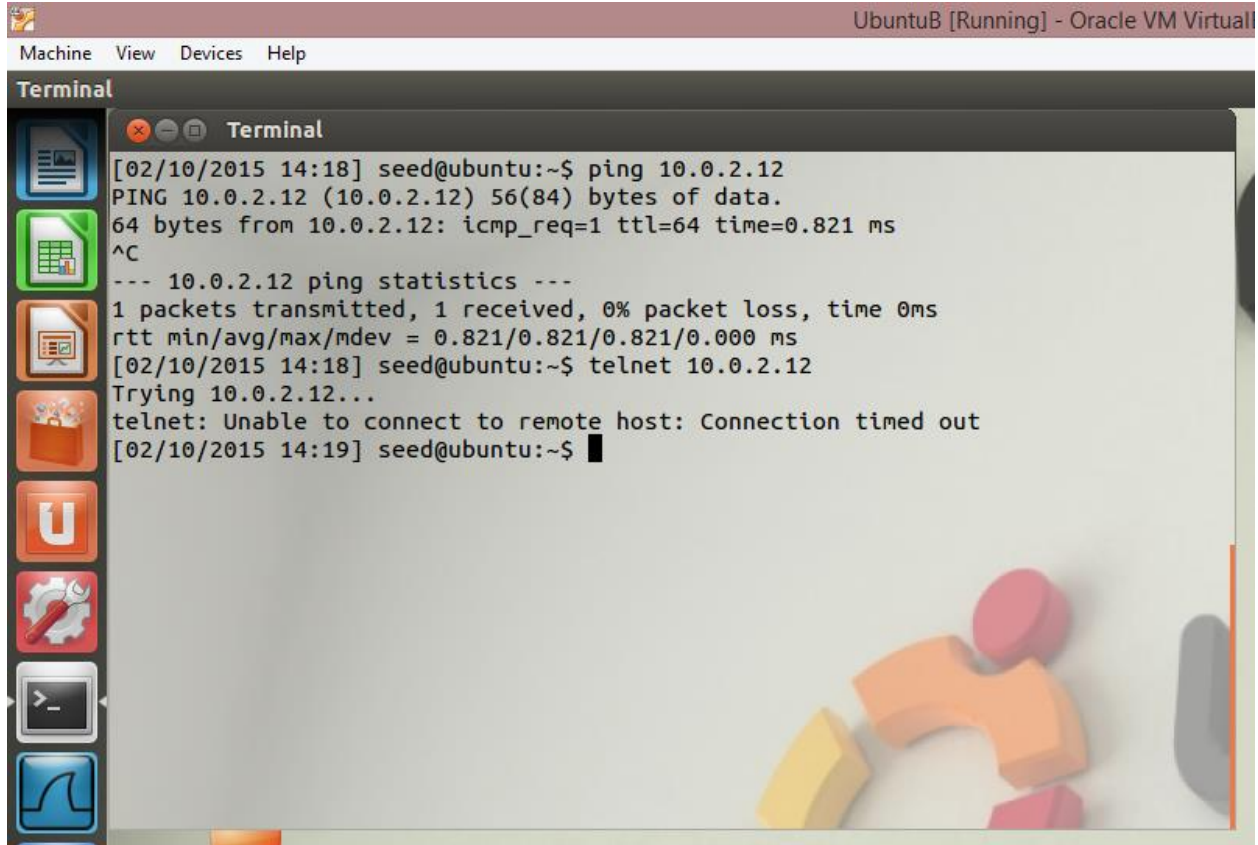
Observation: I created a rule to block the outgoing traffic/packet from UbuntuA to UbuntuB and tried to do a telnet. As I have created an ufw (Ubuntu firewall rule) telnet request gets timed out and does not connect.



```
UbuntuA [Running] - Oracle VM VirtualBox
Machine View Devices Help
Terminal
[02/10/2015 14:17] seed@ubuntu:~$ sudo ufw deny in proto tcp from 10.0.2.13 to 10.0.2.12
Rule added
[02/10/2015 14:17] seed@ubuntu:~$ sudo ufw status
Status: active

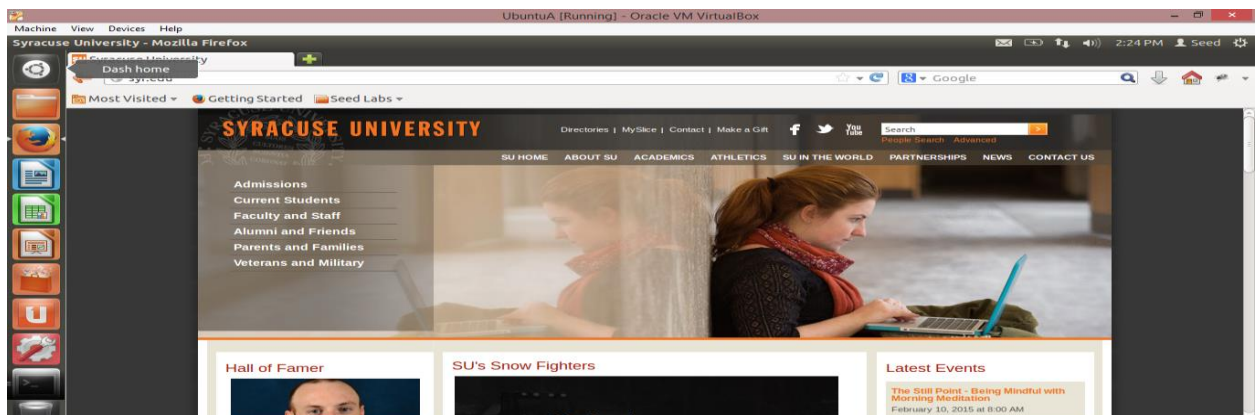
To Action From
--
10.0.2.12/tcp DENY 10.0.2.13/tcp
10.0.2.13/tcp DENY OUT 10.0.2.12/tcp

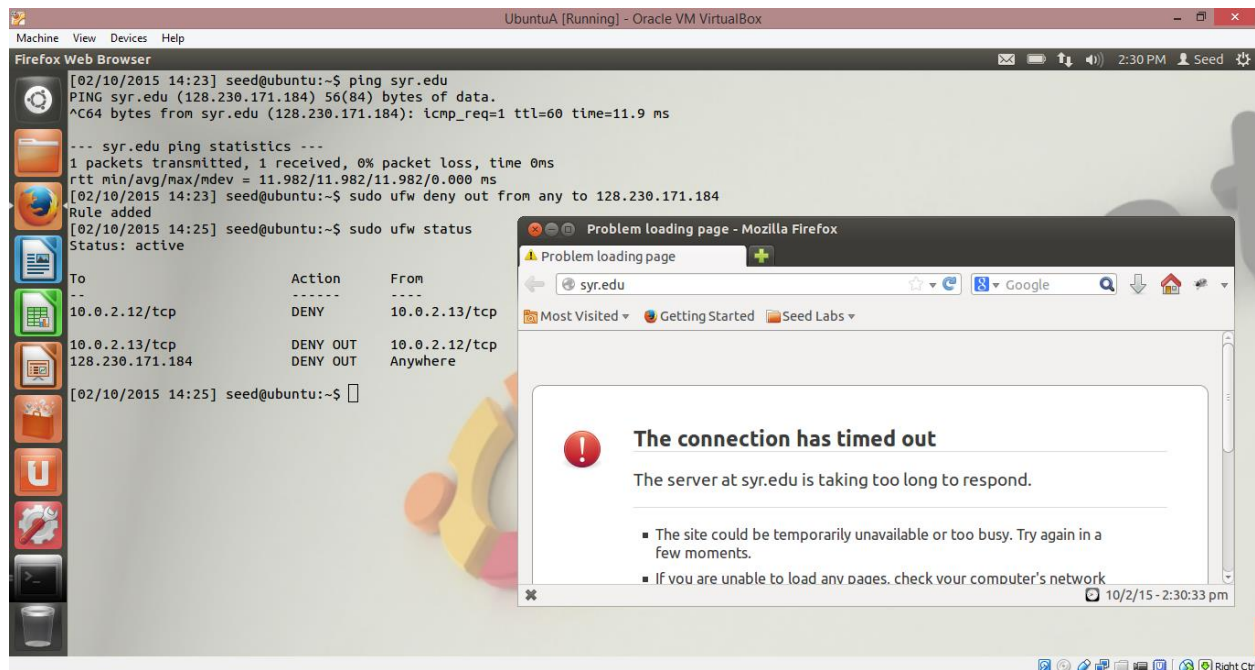
[02/10/2015 14:18] seed@ubuntu:~$ ping 10.0.2.13
PING 10.0.2.13 (10.0.2.13) 56(84) bytes of data.
64 bytes from 10.0.2.13: icmp_req=1 ttl=64 time=0.984 ms
^C
--- 10.0.2.13 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.984/0.984/0.984/0.000 ms
[02/10/2015 14:18] seed@ubuntu:~$
```



Observation: I am trying to lock the incoming traffic to UbuntuA. I created a rule on UbuntuA and trying to create a telnet session from UbuntuB to UbuntuA but did not succeed as the ufw was created.

b) Blocking a website





Observation: we could see that we could see the syr.edu could be assessed from the browser but after creating a firewall rule to block it its inaccessible. Because it is blocking the packets from the particular ip as given in the ufw

Task 2: How Firewall Works

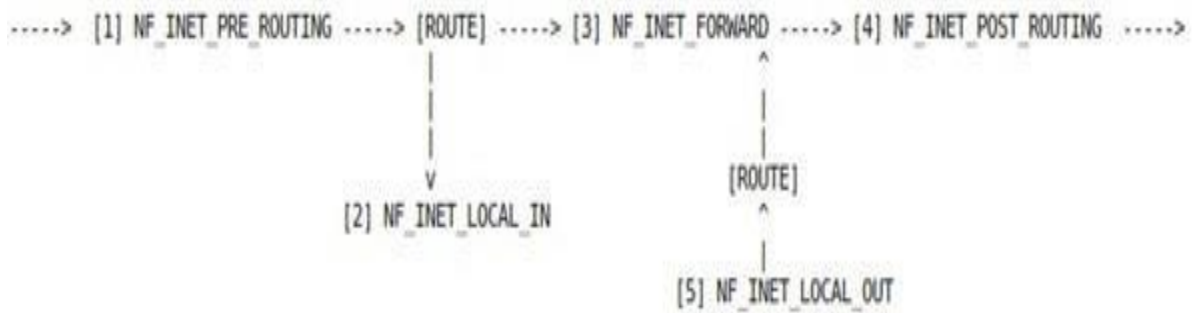
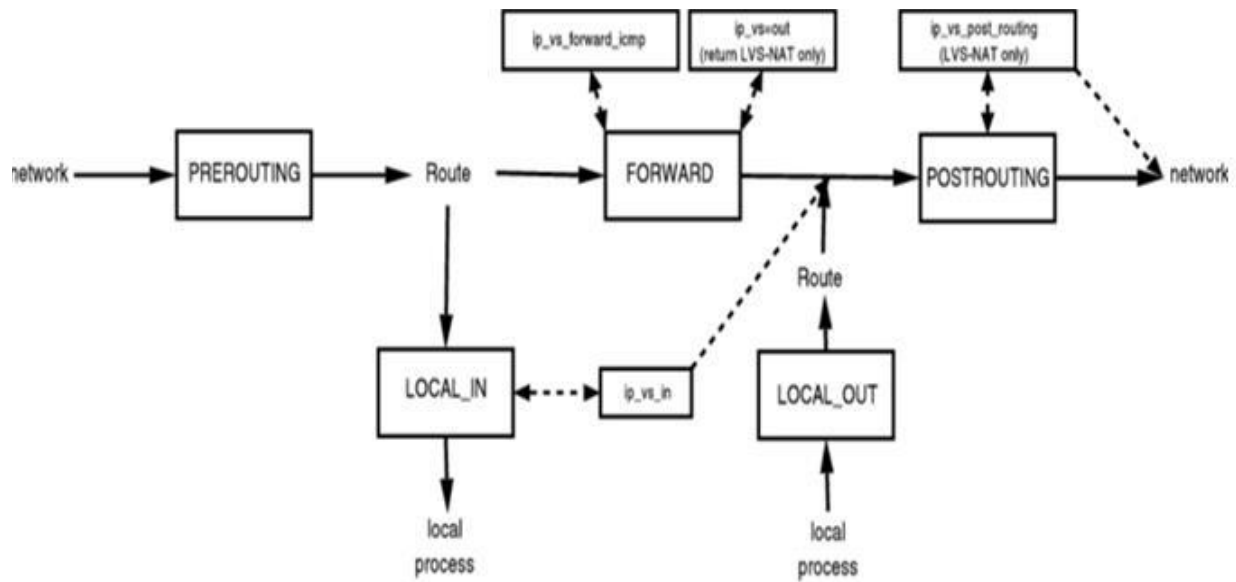
Question 1 : What types of hooks does Netfilter support, and what can you do with these hooks? Please draw a diagram to show how packets flow through these hooks.

Answer :

Netfilter is a framework inside the Linux kernel which offers flexibility for various networking-related operations to be implemented in form of customized handlers.

Different types of hooks supported by Netfilter and their function **are**

- ☐ NF_IP_PRE_ROUTING : This hook called after sanity checks and called before routing decisions.
- ☐ NF_IP_LOCAL_IN : This hook called after routing decisions if packet is for host
- ☐ NF_IP_FORWARD : This hook called if the packet is destined for another interface
- ☐ NF_IP_POST_ROUTING : This hook called for packets coming from local processes on their way out
- ☐ NF_IP_LOCAL_OUT : This hook called for just before outbound packets hit the wire



With all these hooks, netfilter can capture the packet at whatever time it wants and decide whether to drop the packet or process the packet or simply pass it through.

Question 2: Where should you place a hook for ingress filtering, and where should you place a hook for egress filtering?

Ingress filtering placed at the interface between the ISP and the end user.

We can place hook for ingress filtering at the edges of ISPs where appropriate, at the routers connecting LANs to an enterprise network, etc.

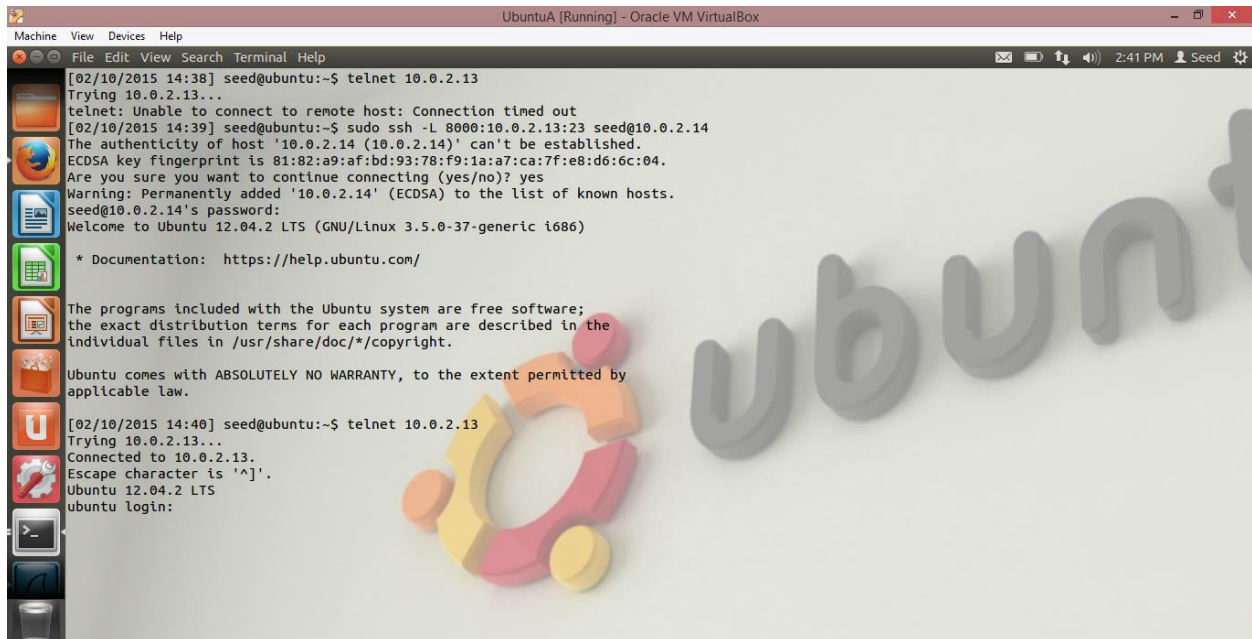
Egress filtering placed at right after the NIC card of my machine and routers.

Question 3: Can you modify packets using Netfilter?

Yes. We can modify packets using Netfilter as we are able to capture it

Task 3: Evading Egress Filtering

Task 3.a: Telnet to Machine B through the firewall



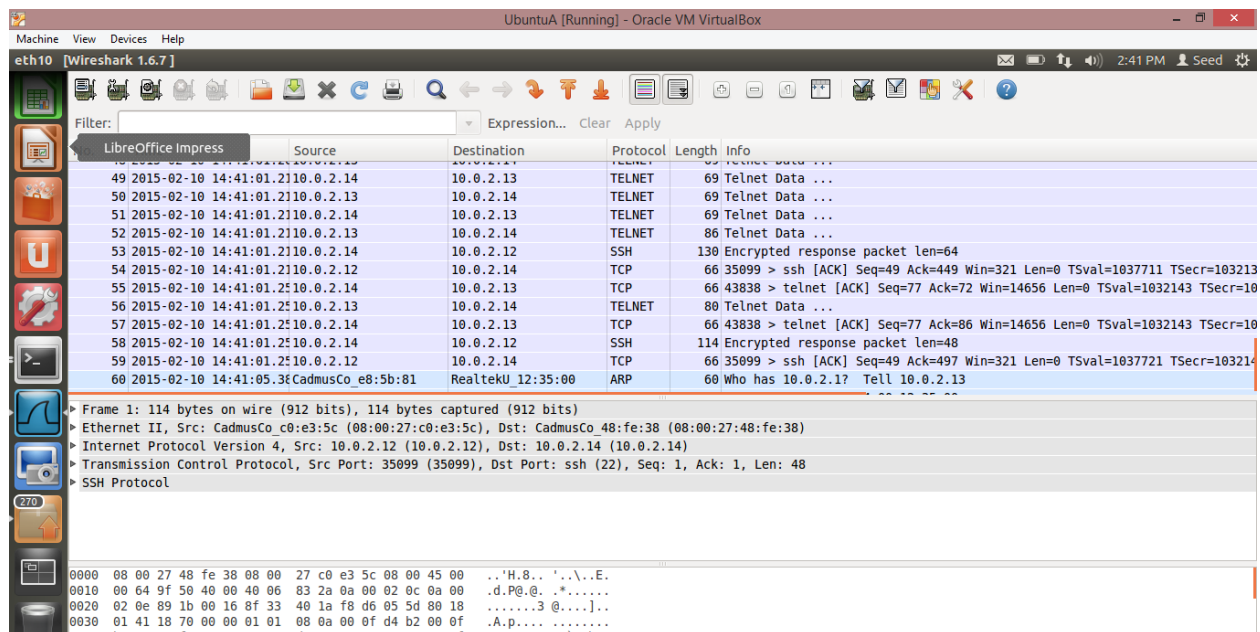
```
Machine View Devices Help
File Edit View Search Terminal Help
[02/10/2015 14:38] seed@ubuntu:~$ telnet 10.0.2.13
Trying 10.0.2.13...
telnet: Unable to connect to remote host: Connection timed out
[02/10/2015 14:39] seed@ubuntu:~$ sudo ssh -L 8000:10.0.2.13:23 seed@10.0.2.14
The authenticity of host '10.0.2.14 (10.0.2.14)' can't be established.
ECDSA key fingerprint is 81:82:a9:af:bd:93:78:f9:1a:a7:ca:7f:e8:d6:6c:04.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.2.14' (ECDSA) to the list of known hosts.
seed@10.0.2.14's password:
Welcome to Ubuntu 12.04.2 LTS (GNU/Linux 3.5.0-37-generic i686)

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

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

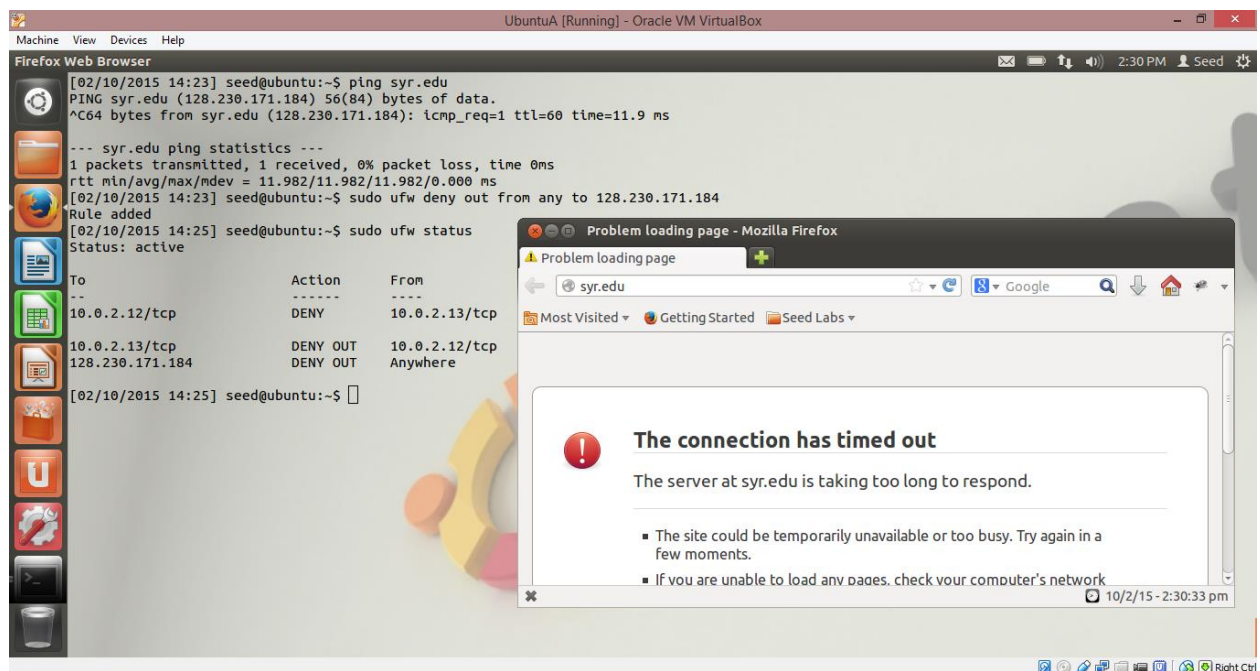
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

[02/10/2015 14:40] seed@ubuntu:~$ telnet 10.0.2.13
Trying 10.0.2.13...
Connected to 10.0.2.13.
Escape character is '^]'.
Ubuntu 12.04.2 LTS
ubuntu login:
```

Observation: we have blocked the telnet in the above question. Now we are trying to use a SSH to pass the traffic. Since SSH is encrypted netfilter is not able to read the content of packet hence the data so it allows the traffic to pass through.

Task 3.b: Connecting to Facebook using SSH Tunnel.



Ufwrule added to block syr.edu

```
Machine View Devices Help
UbuntuA [Running] - Oracle VM VirtualBox
File Edit View Search Terminal Help
[02/10/2015 15:36] seed@ubuntu:~$ ssh -D 9000 -C seed@machine_B
^Z
[1]+  Stopped                  ssh -D 9000 -C seed@machine_B
[02/10/2015 15:37] seed@ubuntu:~$ ssh -D 9000 -C seed@machine_B
^C
[02/10/2015 15:37] seed@ubuntu:~$ ssh -D 9000 -C seed@10.0.2.13
The authenticity of host '10.0.2.13 (10.0.2.13)' can't be established.
ECDSA key fingerprint is 81:82:a9:af:bd:93:78:f9:1a:a7:ca:7f:e8:d6:6c:04.
Are you sure you want to continue connecting (yes/no)? Y
Please type 'yes' or 'no': yes
Warning: Permanently added '10.0.2.13' (ECDSA) to the list of known hosts.
seed@10.0.2.13's password:
Welcome to Ubuntu 12.04.2 LTS (GNU/Linux 3.5.0-37-generic i686)

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

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
```

ssh tunnel created successfully.



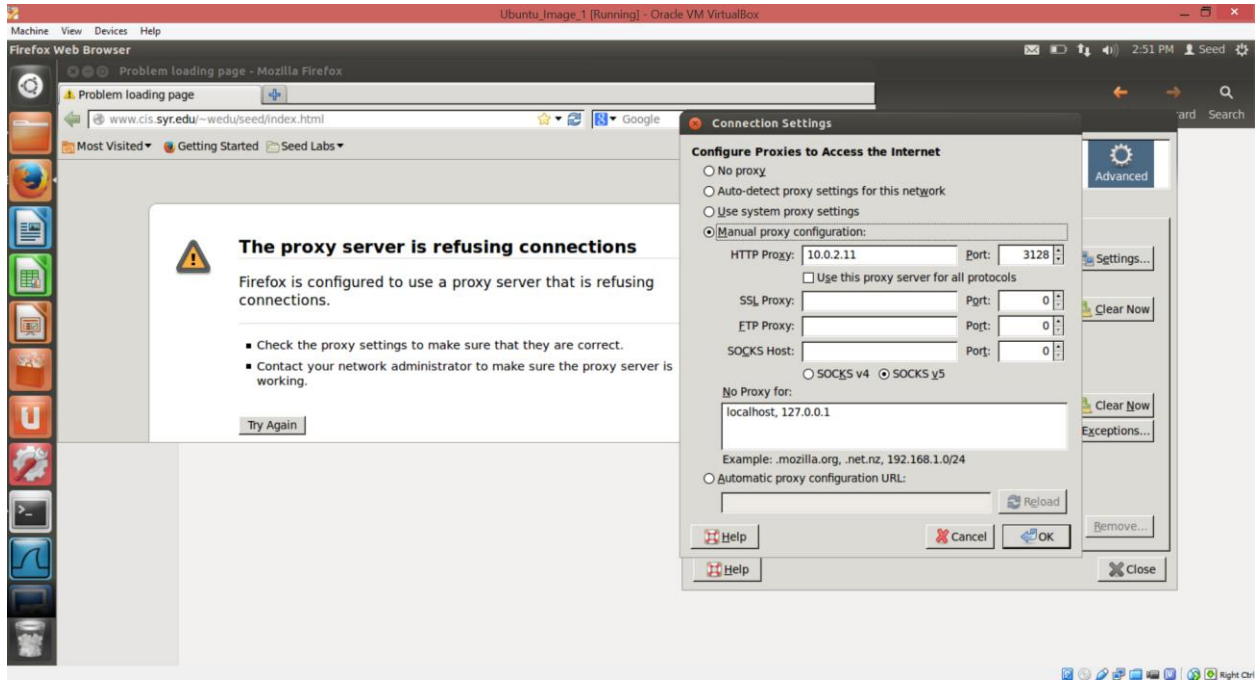
We can access the website syr.edu after the tunneling is done.

Question 4: If ufw blocks the TCP port 22, which is the port used by SSH, can you still set up an SSH tunnel to evade egress filtering?

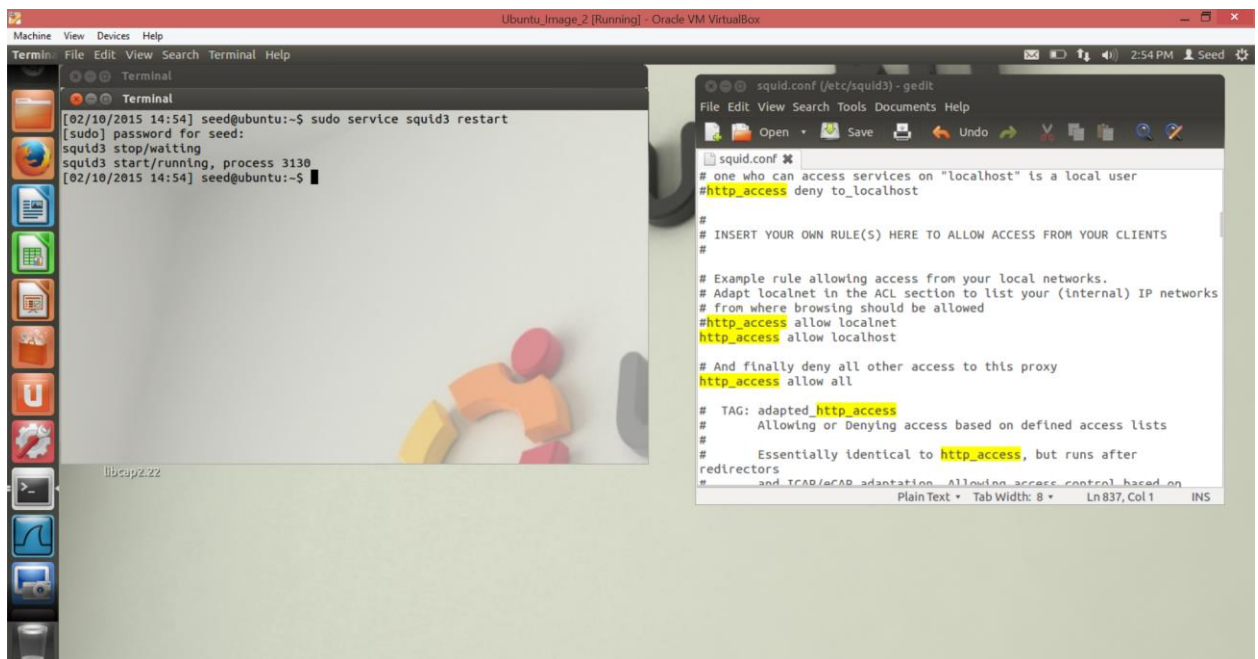
Yes we can change the port and accesses the proxy

Task 4: Web Proxy (Application Firewall)

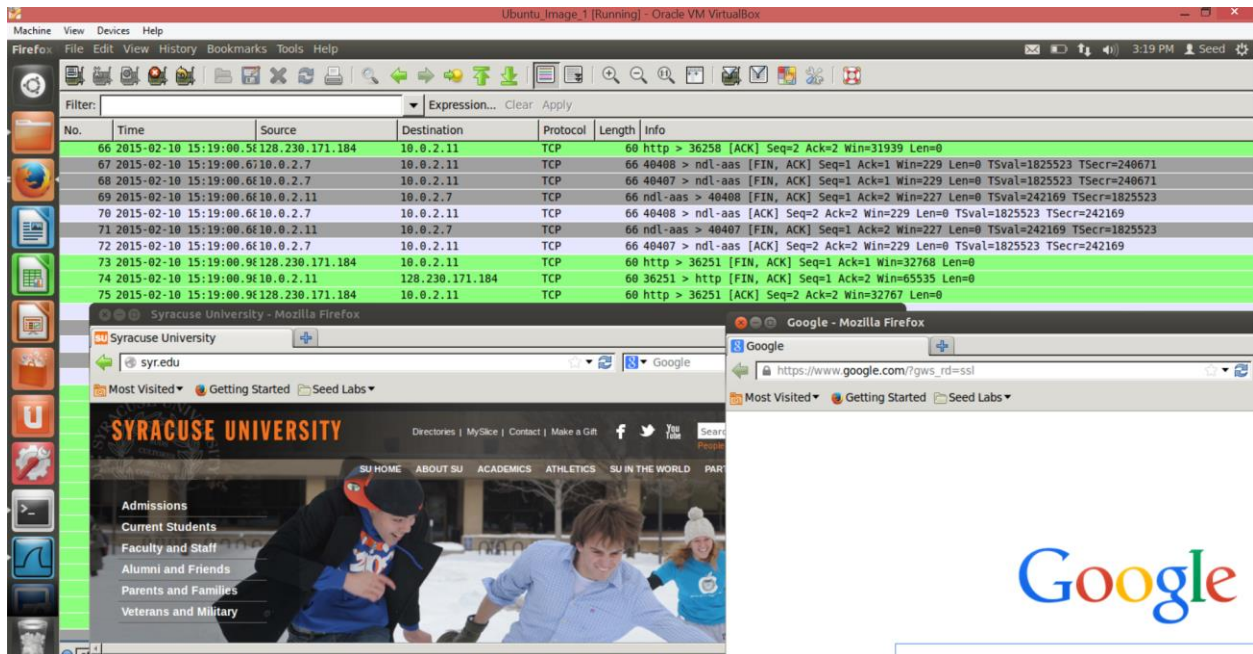
Task 4.a: Setup



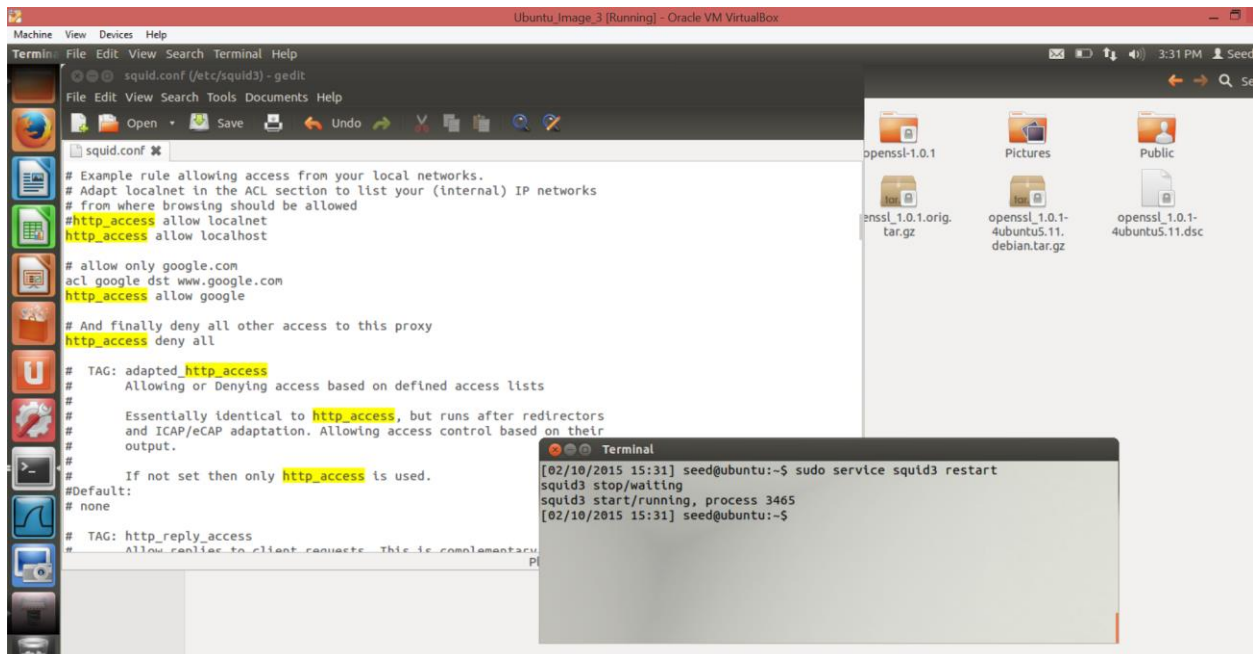
Observation: as the proxy for ubuntuA is set up as UbuntuB and UbuntuB squid3 is set up deny all so the page does not open.



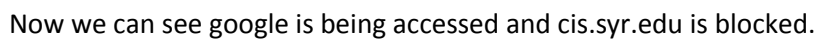
Squid3 is changed to allow all and and squid process is restated.



Now we can see that syr.edu is accessible and wire shark traffic shows between UbuntuA and UbuntuB



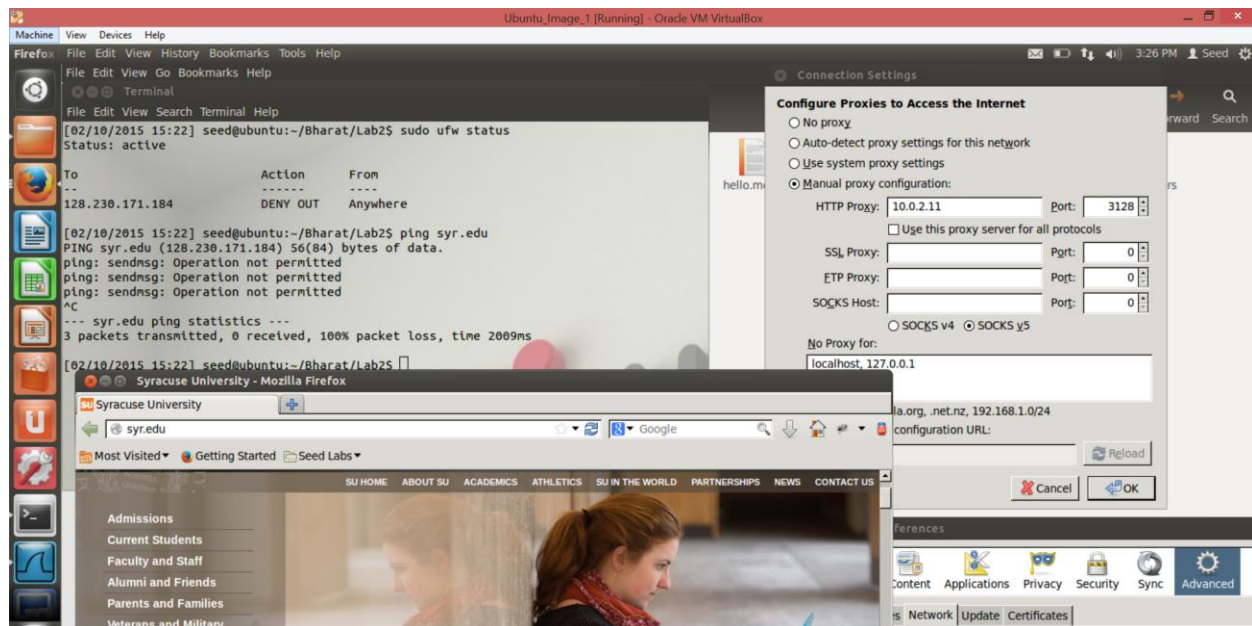
As squid3 changed to allow only google website and other external websites are blocked



The screenshot displays a virtual machine environment with the following components:

- Terminal Window:** Shows the execution of `sudo ufw status` (Status: active) and a `ping` command to `syr.edu`. The output indicates a 100% packet loss and a time-out.
- Firefox Web Browser:** The address bar shows `syr.edu`. A notification banner at the bottom states: "The connection has timed out. The server at syr.edu is taking too long to respond." Below this, a list of troubleshooting steps is provided:
 - The site could be temporarily unavailable or too busy. Try again in a few moments.
 - If you are unable to load any pages, check your computer's network connection.
- Connection Settings Dialog:** A window titled "Connection Settings" is open, showing the "Configure Proxies to Access the Internet" section. The "Use system proxy settings" option is selected. Other options include "No proxy", "Auto-detect proxy settings for this network", and "Manual proxy configuration". The "Manual proxy configuration" section has fields for HTTP, SSL, FTP, and SOCKS proxies, all of which are currently empty. The "SOCKS" section is set to "SOCKS v4". The "No Proxy for:" field contains `localhost, 127.0.0.1`. The "Automatic proxy configuration URL:" field is empty. The "Reload" button is disabled.

Ufw rule was setup as shown to block syr.edu and we are using system proxy is blocked.

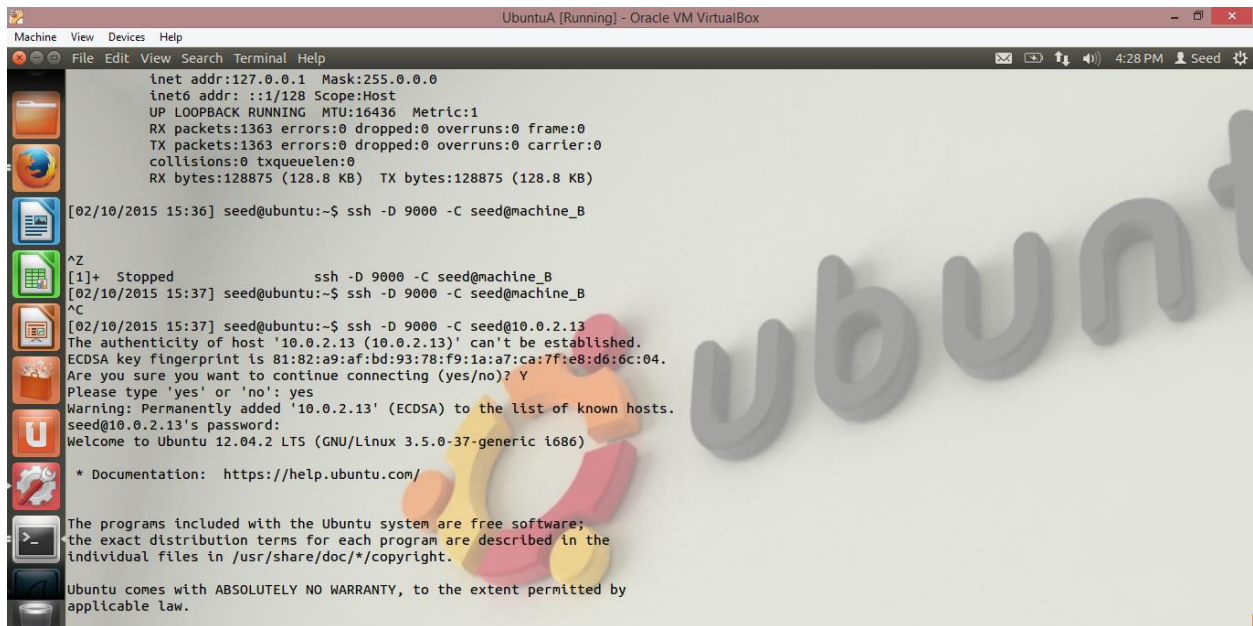


Ufw is used to blocked syracuse.edu. But we setup proxy settings as shown to use web proxy of UbuntuB. And so we can access syracuse.edu

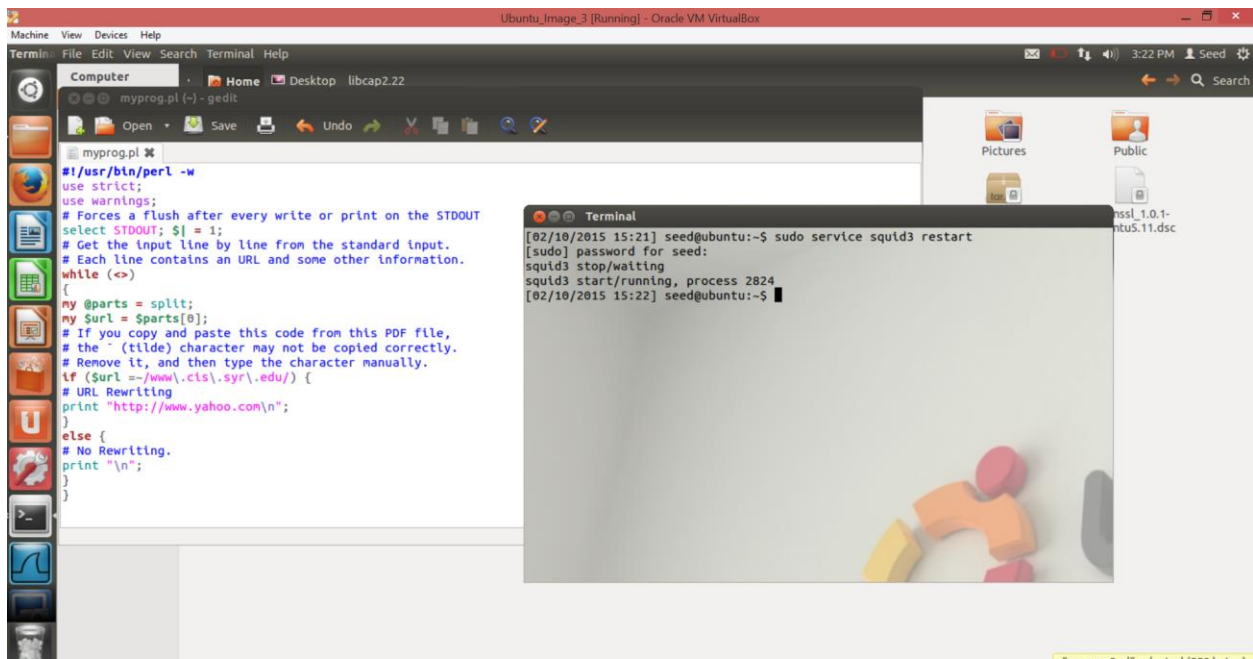
The screenshot shows a Wireshark network capture with the following data:

No.	Time	Source	Destination	Protocol	Length	Info
1	2015-02-10 15:27:47.2610.0.2.7	31.13.71.1	10.0.2.7	TLV1	91	Application Data
2	2015-02-10 15:27:47.2231.13.71.1	10.0.2.7	31.13.71.1	TLV1	91	Application Data
3	2015-02-10 15:27:47.2210.0.2.7	10.0.2.11	128.230.171.184	TCP	54	42554 > https [ACK] Seq=38 Ack=38 Win=40880 Len=0
4	2015-02-10 15:27:47.2210.0.2.7	10.0.2.11	128.230.171.184	HTTP	913	GET http://syr.edu/ HTTP/1.1
5	2015-02-10 15:27:47.2210.0.2.11	128.230.171.184	10.0.2.11	TCP	74	36296 > http [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=373819 TSecr=0 WS=64
6	2015-02-10 15:27:47.22128.230.171.184	10.0.2.11	128.230.171.184	TCP	60	http > 36296 [SYN, ACK] Seq=0 Ack=1 Win=32768 Len=0 MSS=1460
7	2015-02-10 15:27:47.2210.0.2.11	128.230.171.184	10.0.2.11	TCP	60	36296 > http [ACK] Seq=1 Ack=1 Win=14600 Len=0
8	2015-02-10 15:27:47.2210.0.2.11	128.230.171.184	10.0.2.11	HTTP	949	GET / HTTP/1.1
9	2015-02-10 15:27:47.22128.230.171.184	10.0.2.11	128.230.171.184	HTTP	220	HTTP/1.1 304 Not Modified
10	2015-02-10 15:27:47.2210.0.2.11	128.230.171.184	10.0.2.7	TCP	60	36296 > http [ACK] Seq=896 Ack=167 Win=15544 Len=0
11	2015-02-10 15:27:47.2210.0.2.11	10.0.2.7	10.0.2.11	HTTP	385	HTTP/1.0 304 Not Modified
12	2015-02-10 15:27:47.2210.0.2.7	10.0.2.11	10.0.2.11	TCP	66	40488 > ndl-aas [ACK] Seq=848 Ack=240 Win=3327 Len=0 TSval=1957176 TSecr=373823
13	2015-02-10 15:27:47.3110.0.2.7	10.0.2.11	128.230.171.184	HTTP	986	GET http://syr.edu/css/reset.css HTTP/1.1
14	2015-02-10 15:27:47.3110.0.2.7	10.0.2.11	128.230.171.184	HTTP	988	GET http://syr.edu/css/global.css HTTP/1.1
15	2015-02-10 15:27:47.3110.0.2.7	10.0.2.11	128.230.171.184	HTTP	989	GET http://syr.edu/css/modules.css HTTP/1.1
16	2015-02-10 15:27:47.3110.0.2.11	128.230.171.184	10.0.2.7	HTTP	942	/css/reset.css HTTP/1.1
17	2015-02-10 15:27:47.3110.0.2.11	128.230.171.184	10.0.2.7	TCP	74	36297 > http [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=373829 TSecr=0 WS=64
18	2015-02-10 15:27:47.3110.0.2.11	10.0.2.7	10.0.2.11	TCP	66	ndl-aas > 40487 [ACK] Seq=1 Ack=844 Win=418 Len=0 TSval=373829 TSecr=1957181
19	2015-02-10 15:27:47.3110.0.2.7	10.0.2.11	128.230.171.184	HTTP	911	GET http://syr.edu/css/templates.css HTTP/1.1
20	2015-02-10 15:27:47.3110.0.2.11	128.230.171.184	10.0.2.11	TCP	74	36298 > http [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=373829 TSecr=0 WS=64
21	2015-02-10 15:27:47.3110.0.2.7	10.0.2.11	128.230.171.184	HTTP	983	GET http://syr.edu/css/nav.css HTTP/1.1
22	2015-02-10 15:27:47.3110.0.2.11	128.230.171.184	10.0.2.11	TCP	74	36299 > http [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=373829 TSecr=0 WS=64
23	2015-02-10 15:27:47.3110.0.2.7	10.0.2.11	128.230.171.184	HTTP	988	GET http://syr.edu/css/rotator.css HTTP/1.1
24	2015-02-10 15:27:47.3110.0.2.11	128.230.171.184	10.0.2.7	TCP	74	36300 > http [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=373829 TSecr=0 WS=64
25	2015-02-10 15:27:47.3110.0.2.11	10.0.2.7	10.0.2.11	TCP	66	ndl-aas > 40484 [ACK] Seq=1 Ack=843 Win=538 Len=0 TSval=373829 TSecr=1957181
26	2015-02-10 15:27:47.3110.0.2.7	10.0.2.11	128.230.171.184	HTTP	912	GET http://syr.edu/css/patch/safari.css HTTP/1.1
27	2015-02-10 15:27:47.3110.0.2.11	128.230.171.184	10.0.2.11	TCP	74	36301 > http [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=373829 TSecr=0 WS=64
28	2015-02-10 15:27:47.3110.0.2.7	10.0.2.11	128.230.171.184	HTTP	986	GET http://syr.edu/css/print.css HTTP/1.1
29	2015-02-10 15:27:47.3110.0.2.11	10.0.2.7	128.230.171.184	TCP	66	ndl-aas > 40485 [ACK] Seq=1 Ack=847 Win=488 Len=0 TSval=373829 TSecr=1957181
30	2015-02-10 15:27:47.3110.0.2.11	128.230.171.184	10.0.2.7	TCP	74	36302 > http [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=373829 TSecr=0 WS=64

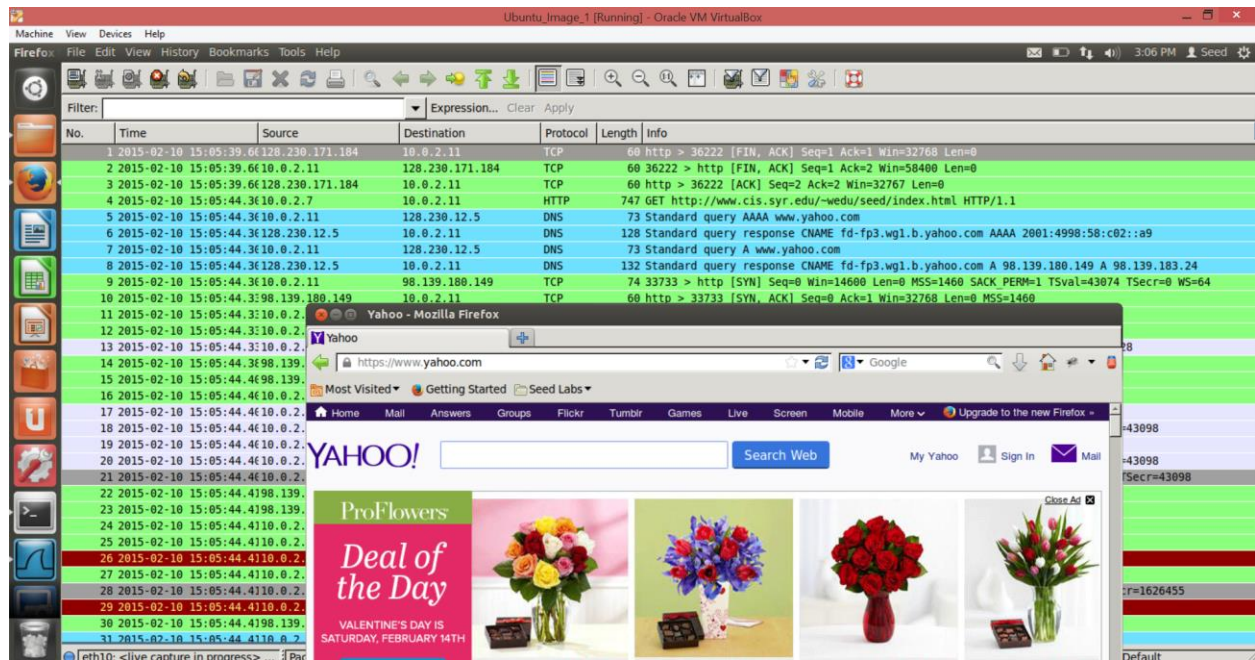
Wireshark session proves that there is traffic between UbuntuA and ubuntuB



Task 4.c Url Rewrite



url rewrite program 'myprog.pl' is used. This program will redirect cis.syr.edu with yahoo.com.
url_rewrite tag was changed to the path of myprog.pl and squid3 was restarted

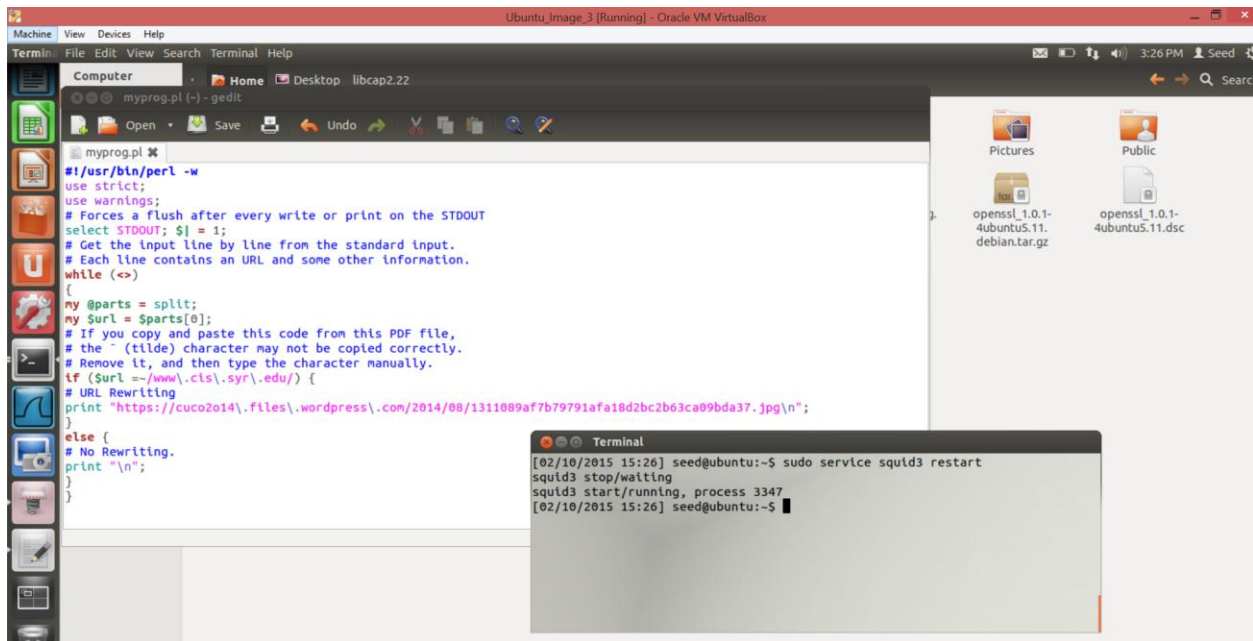


Accessing cis.syr.edu/~wedu/seed/index.html will redirect it to yahoo.com and wireshark session shows the redirection traffic.

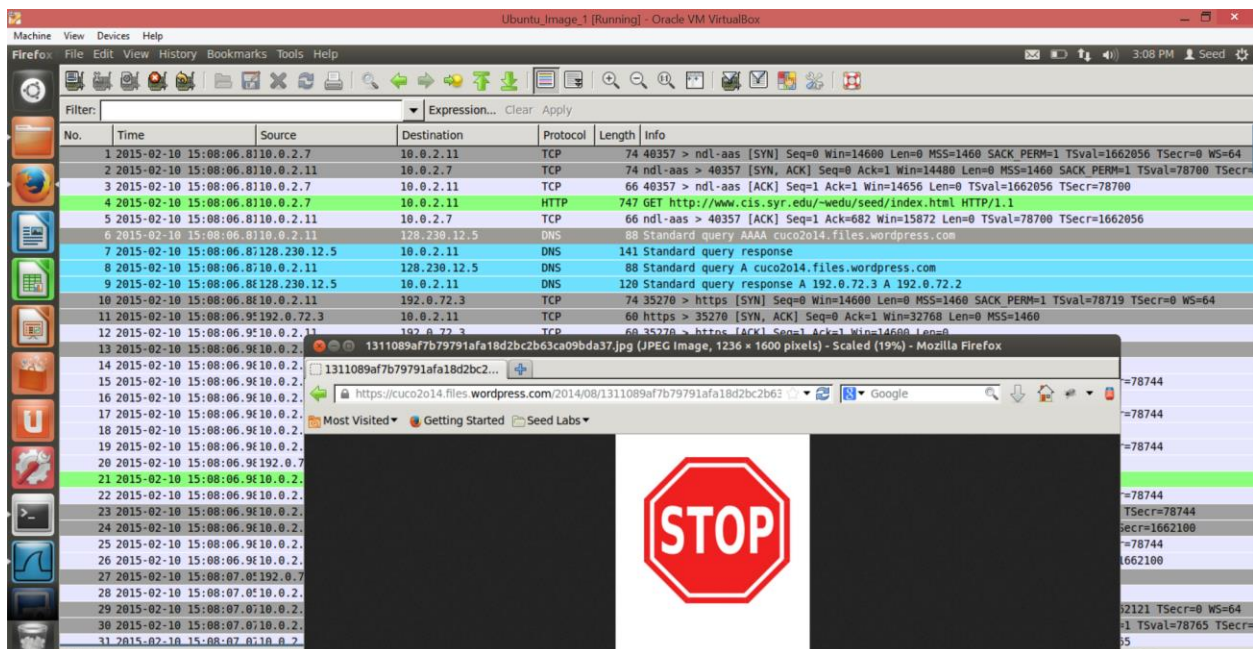
Question 5: If ufw blocks the TCP port 3128, can you still use web proxy to evade the firewall?

Yes we can change the port to evade firewall

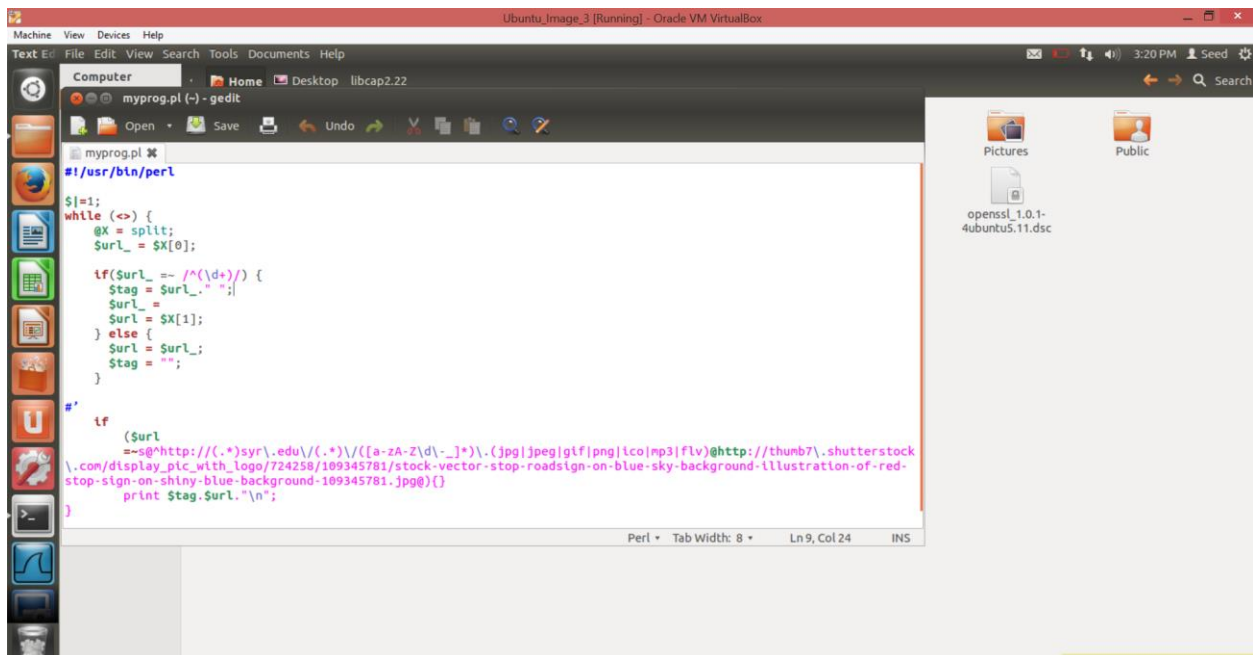
Task 4.c Url redirect to red stop sign



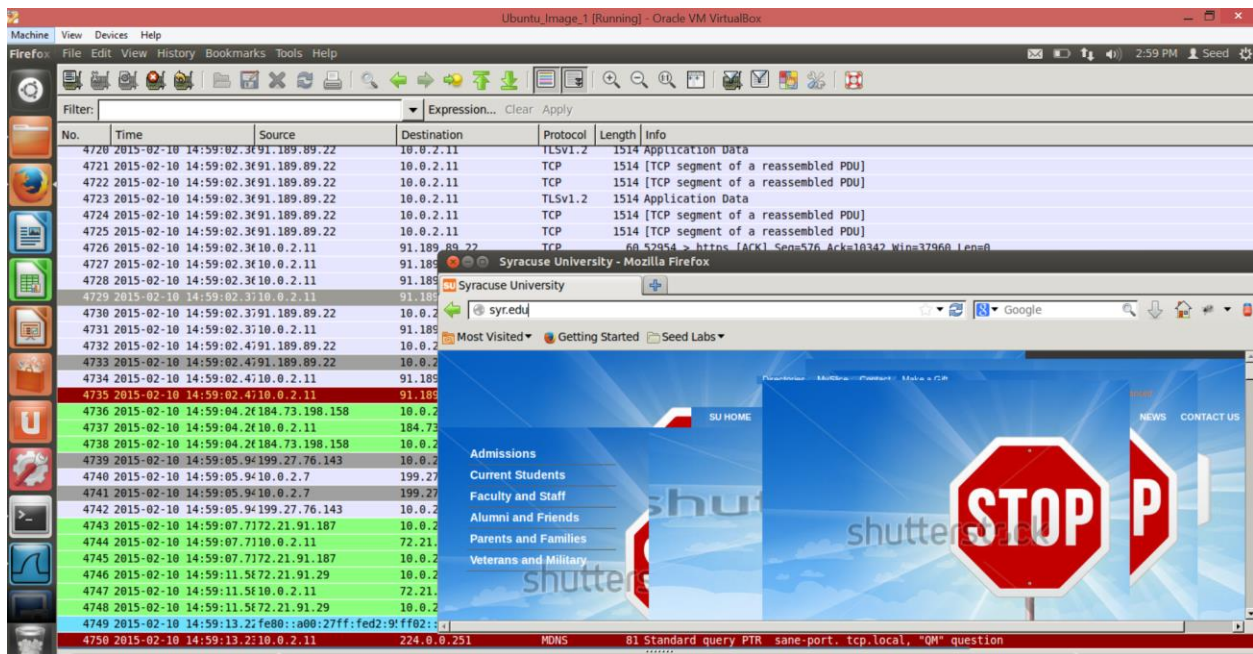
myprog.pl redirect url is changed to red_stop sign and squid3 is restarted.



When we try to access 'cis.syr.edu/~wedu/seed/index.html' red stop sign image is displayed



myprog.pl is modified as shown. When we encounter a .jpg or .jpeg or .gif or .png image then url is redirected to red stop sign image url.



AS we can see all the image requests in syr.edu are redirected to the red stopsign image url and hence we see only the redstop sign image. Wireshark traffic shows the traffic.

Question 6: We can use the SSH and HTTP protocols as tunnels to evade the egress filtering. Can we use the ICMP protocol as a tunnel to evade the egress filtering? Please briefly describe how

Yes, ICMP tunneling works by injecting arbitrary data into an echo packet sent to a remote computer. The remote computer replies in the same manner, injecting an answer into another ICMP packet and sending it back. The client performs all communication using ICMP echo request packets, while the proxy uses echo reply packets (as per Wikipedia).

The client will perform all its communications using ICMP echo request (ping) packets (type 8), whereas the proxy will use echo reply packets (type 0). They all use raw sockets.

