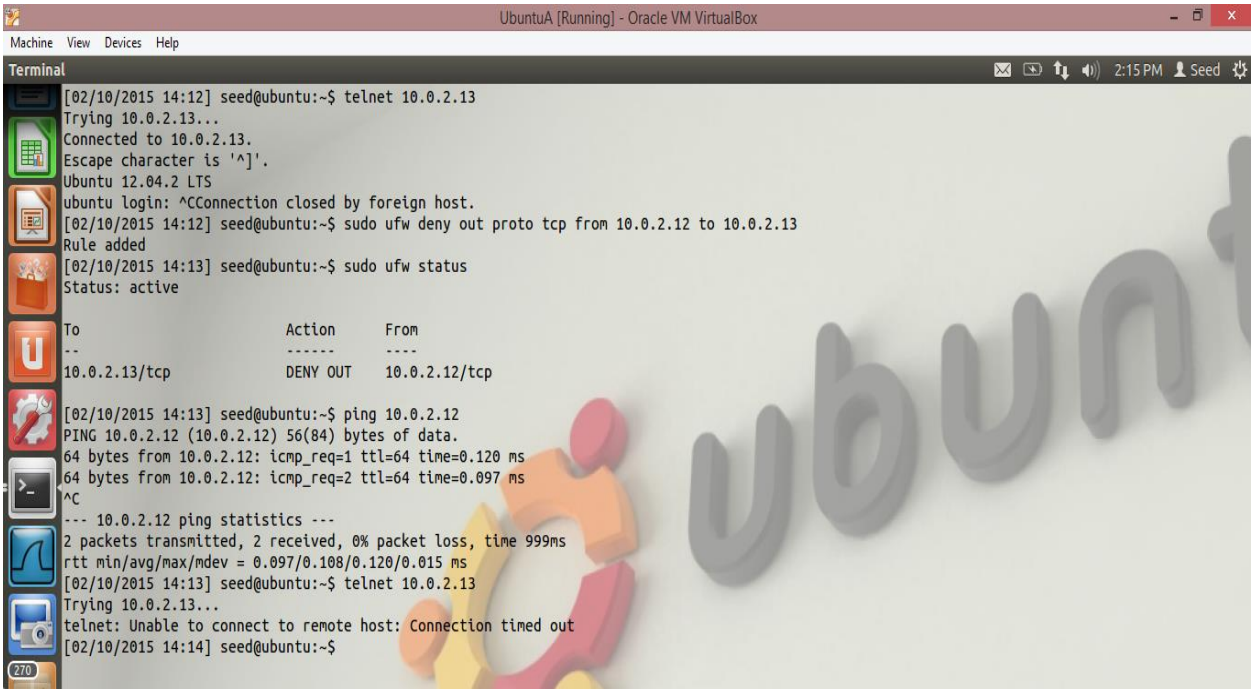
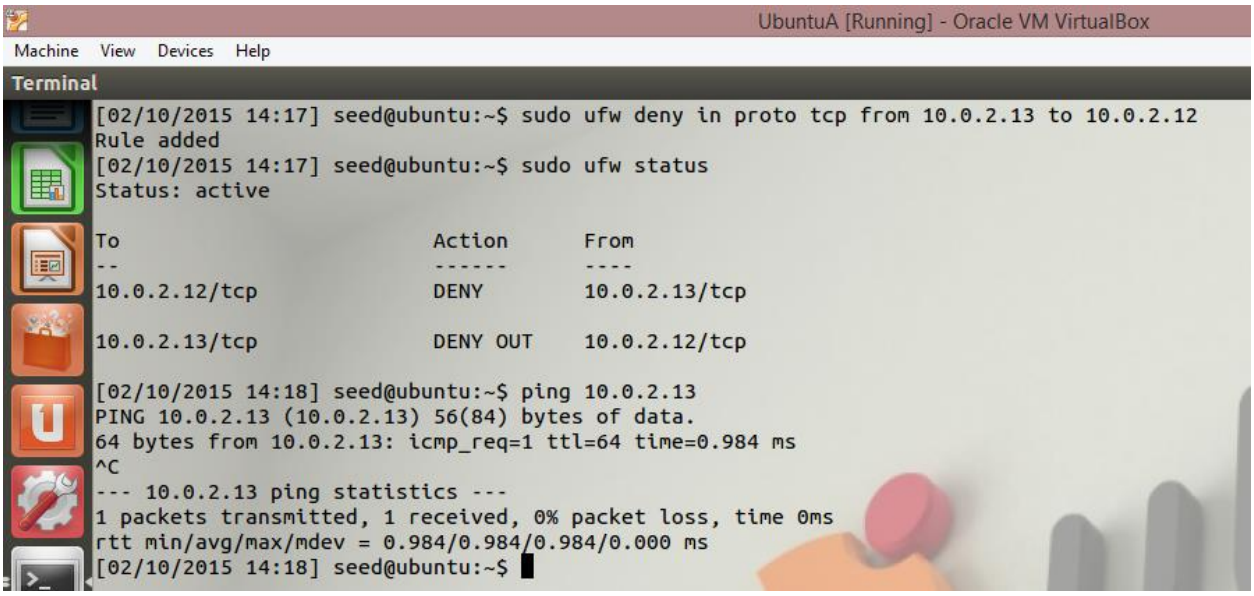
Internet Security - Linux Firewall Exploration Lab
Name – Abhishek Tripathi
SUID: 35081-6306

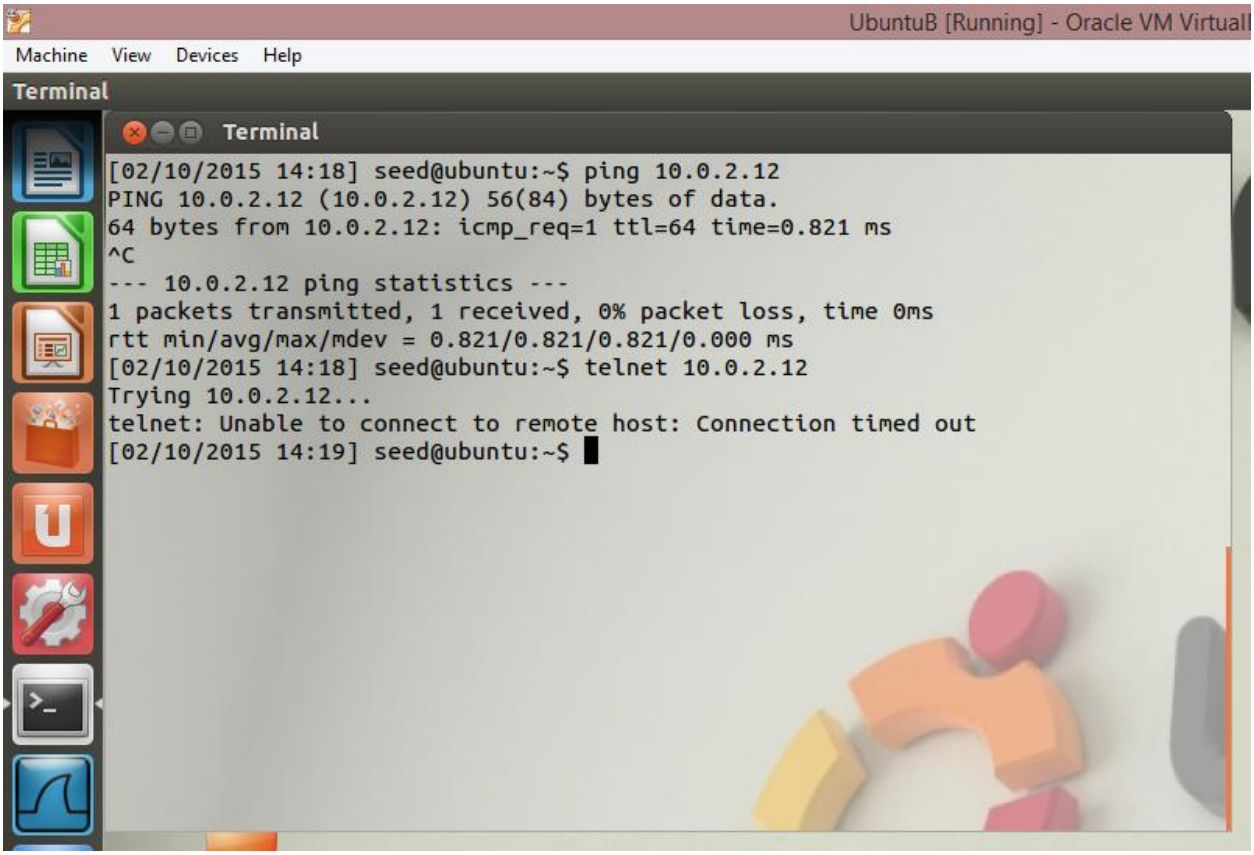
Task 1: Using Firewall

a) Block telnet



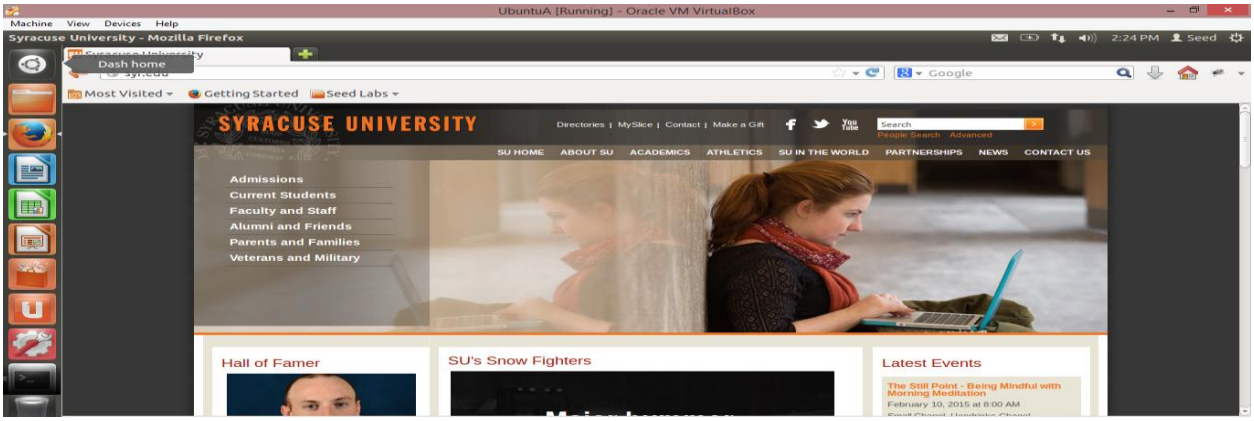
Observation: I created a rule to block the outgoing traffic/packet from UbuntuA to UbuntuB and tried to do a telnet. As I have created an ufw (Ubuntu firewall rule) telnet request gets timed out and does not connect.

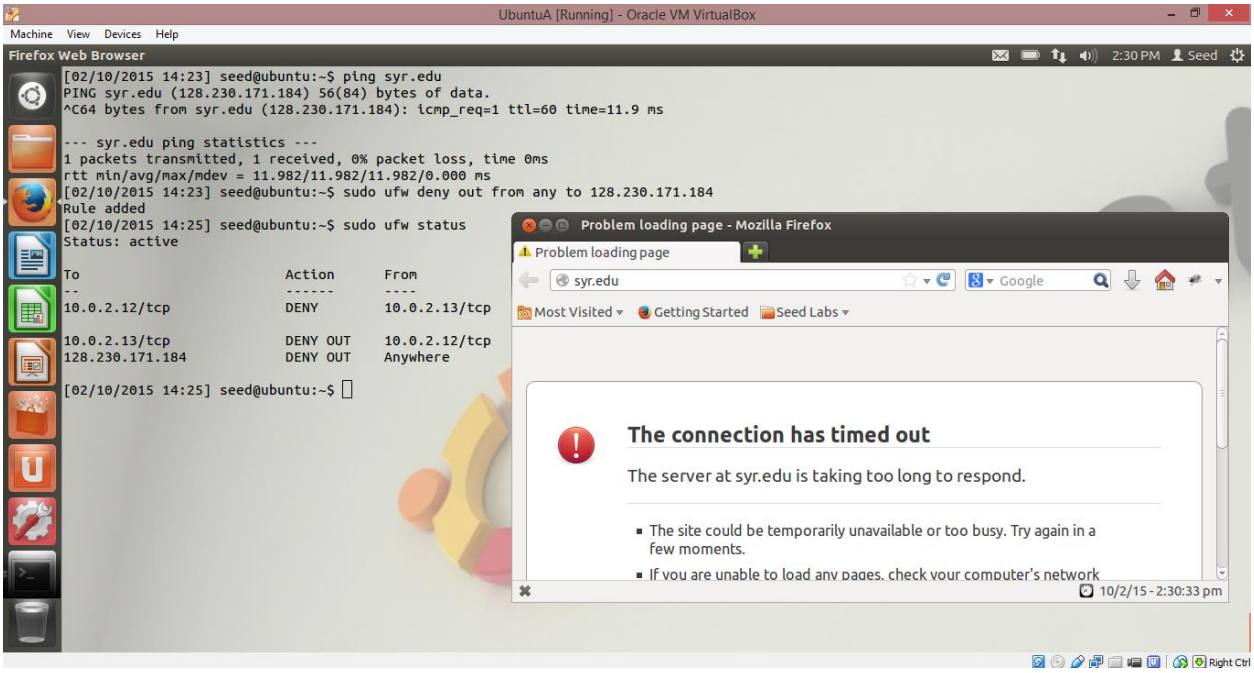




Observation: I am trying to lock the incoming traffic to UbuntuA. I created a rule on UbuntuA and trying to create a telnet session from UbuntuB to UbuntuA but did not successes as the ufw was created.

b) Blocking a wbsite





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

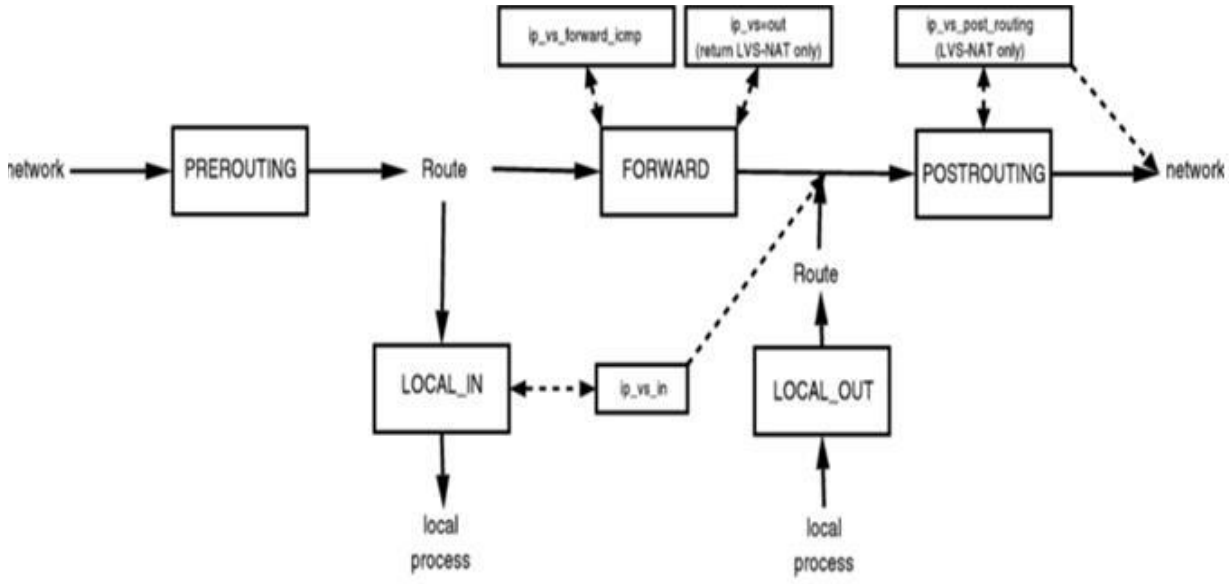
Question 1 : What types of hooks does Netfilter support, and what can you do with these hooks? Please draw a diagram to show how packets flow through these hooks.

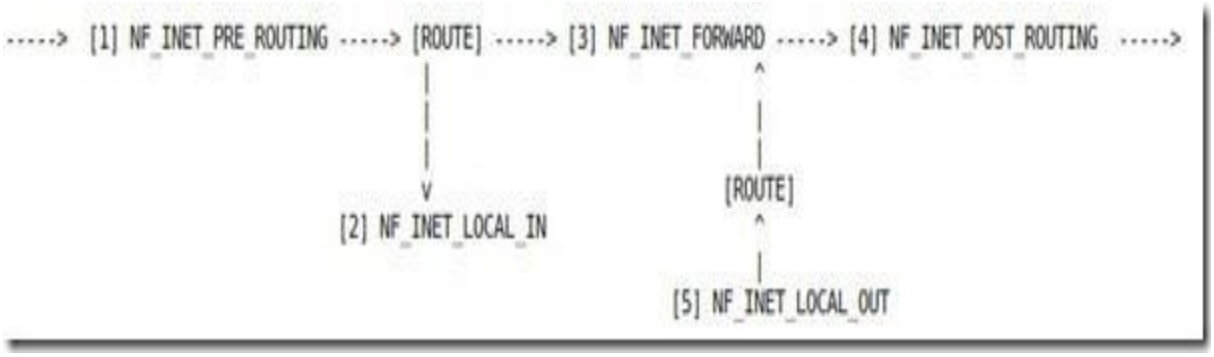
Answer :

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


network
PREROUTING
Route
ip_vs_forward_icmp
ip_vs=out
(return LVS-NAT only)
FORWARD
ip_vs_post_routing
(LVS-NAT only)
POSTROUTING
network
Route
LOCAL_IN
ip_vs_in
LOCAL_OUT
local
process
local
process


-----> [1] NF_INET_PRE_ROUTING -----> [ROUTE] -----> [3] NF_INET_FORWARD -----> [4] NF_INET_POST_ROUTING ----->
[2] NF_INET_LOCAL_IN
[ROUTE]
[5] NF_INET_LOCAL_OUT

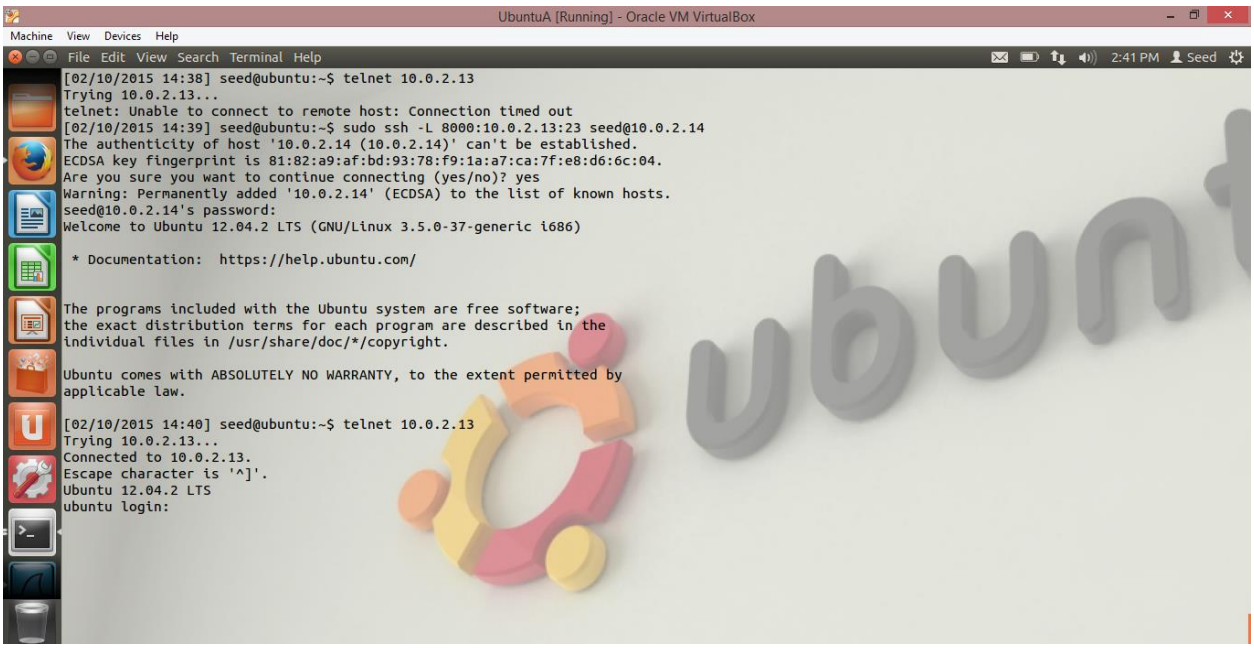
With all these hooks, netfilter can capture the packet at whatever time it wants and d decide whether to drop the packet or process the packet or simply pass it through.

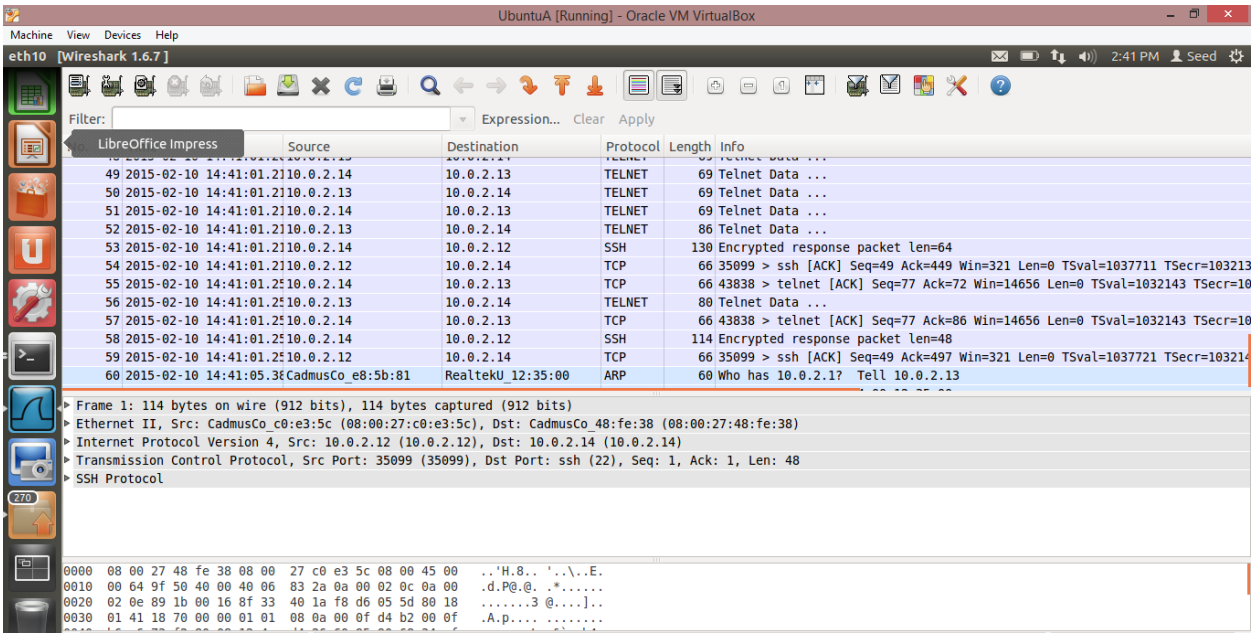
Question 2: Where should you place a hook for ingress filtering, and where should you place a hook for egress filtering?
Ingress filtering placed at the interface between the ISP and the end user.
We can place hook for ingress filtering at the edges of ISPs where appropriate, at the routers connecting 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

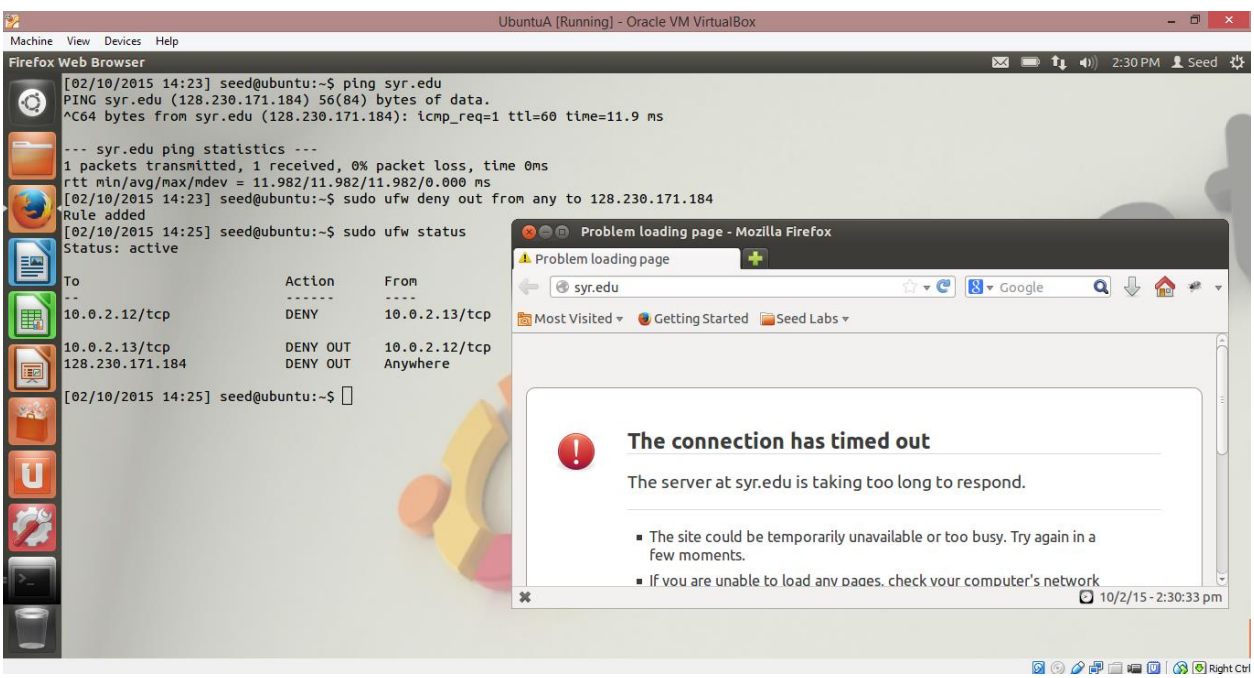
Task 3.a: Telnet to Machine B through the firewal



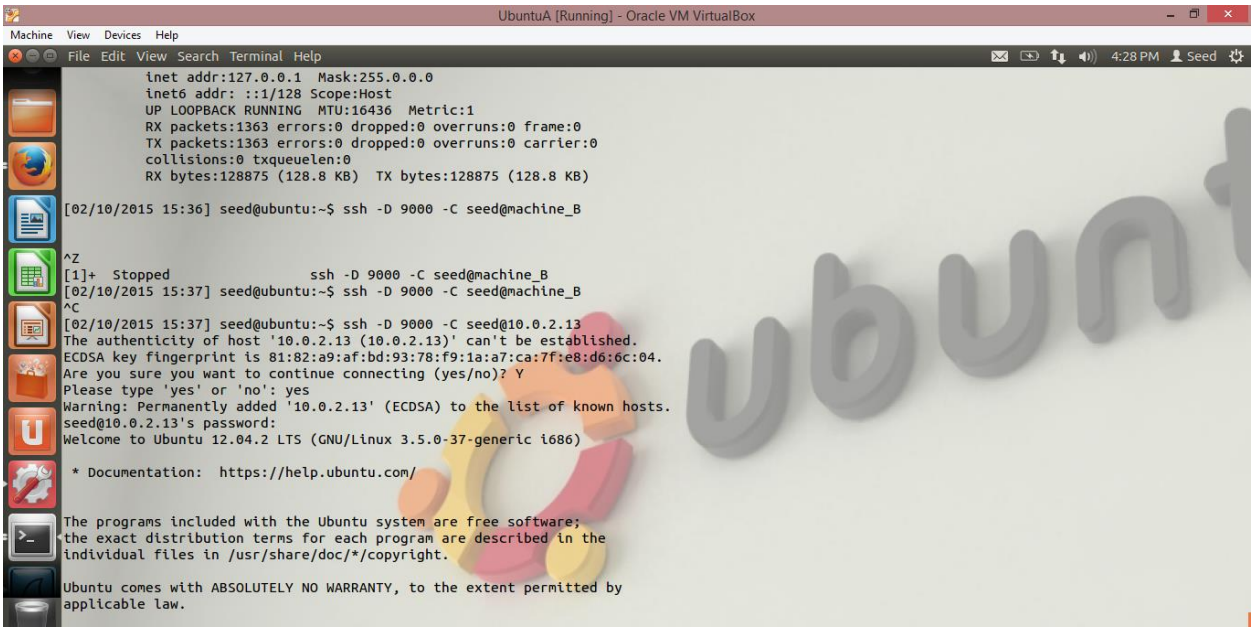


Observation: we have blocked the telnet in the above question. Now we are trying to use a SSH to pass the traffic. Since SSH is encrypted netfilter is not able to read the content of packet hence the data so it allows the traffic to pass through.

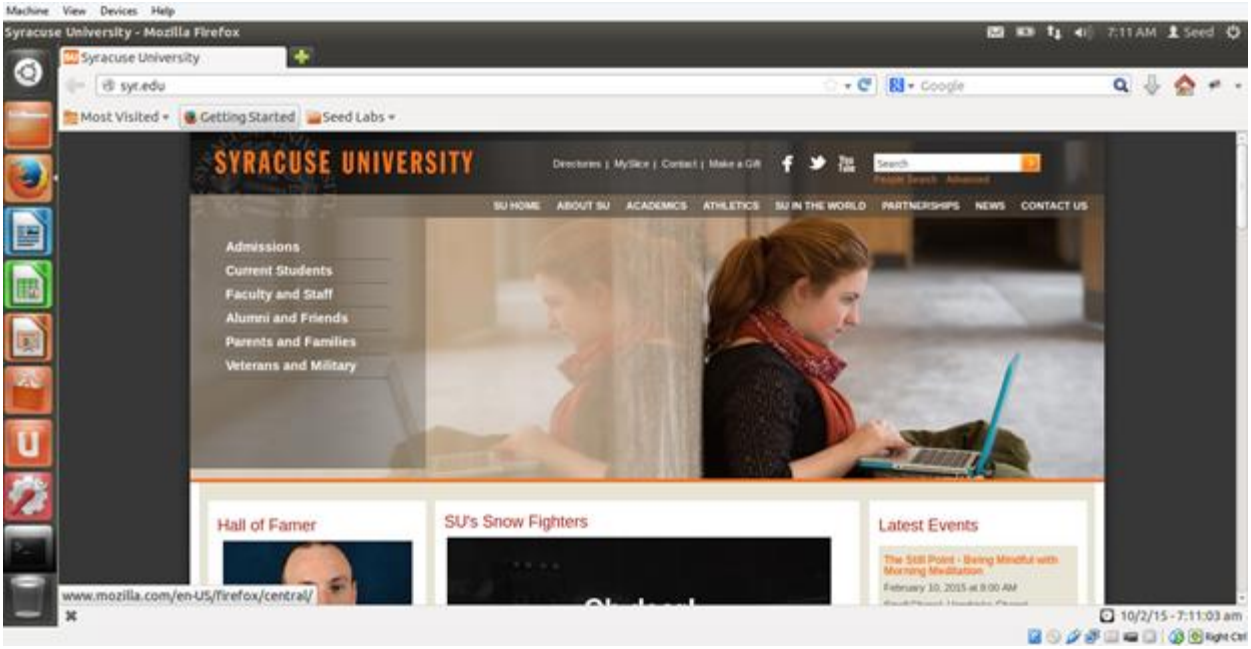
Task 3.b: Connecting to Facebook using SSH Tunnel.



Ufwrule added to block syr.edu



ssh tunnel created successfully.



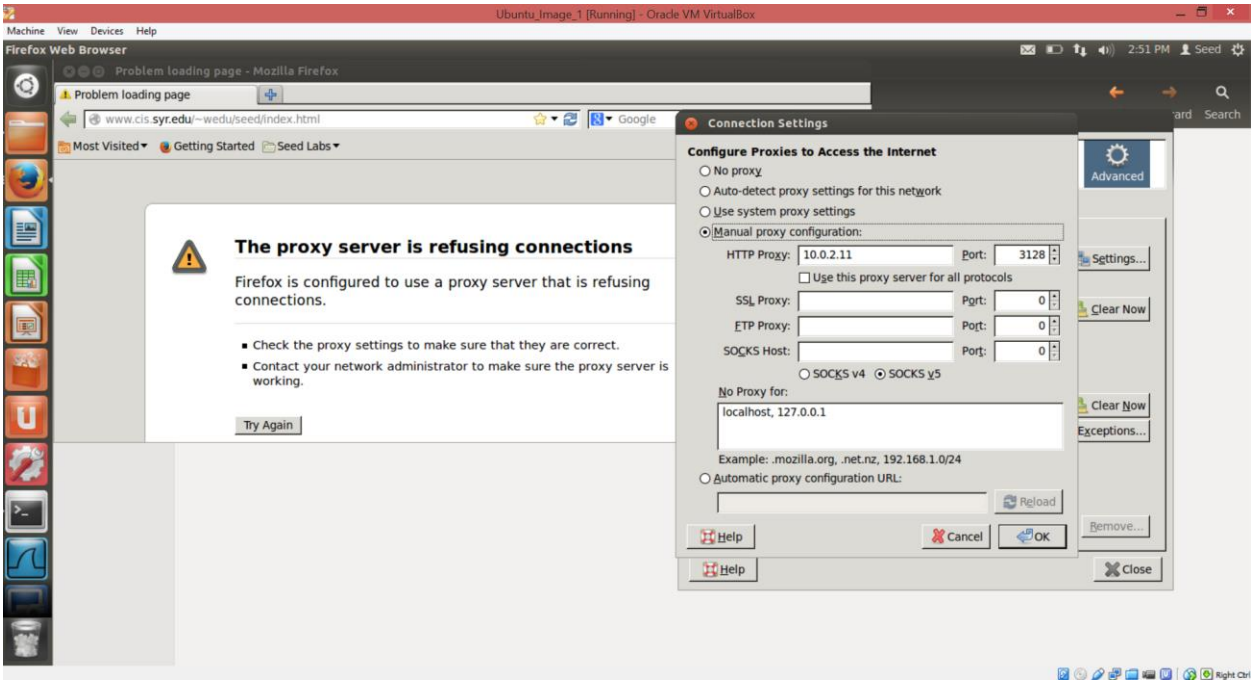
We can access the website syr.edu after the tunneling is done.

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?

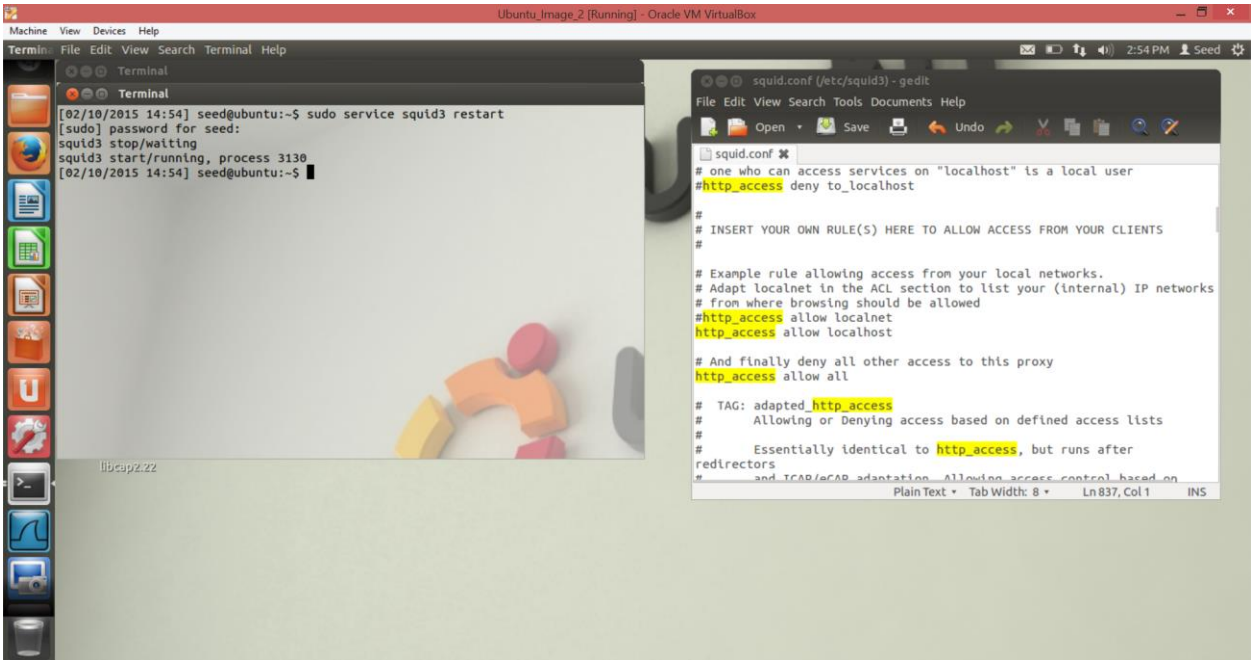
Yes we can change the port and accesses the proxy

Task 4: Web Proxy (Application Firewall)

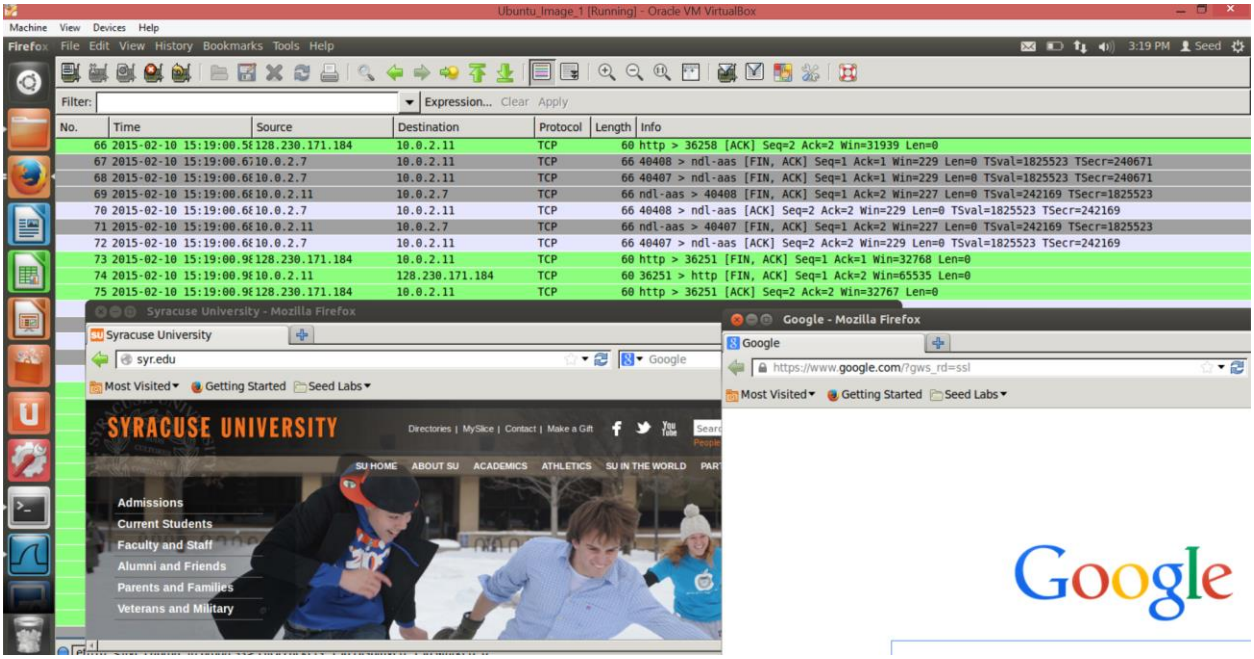
Task 4.a: Setup



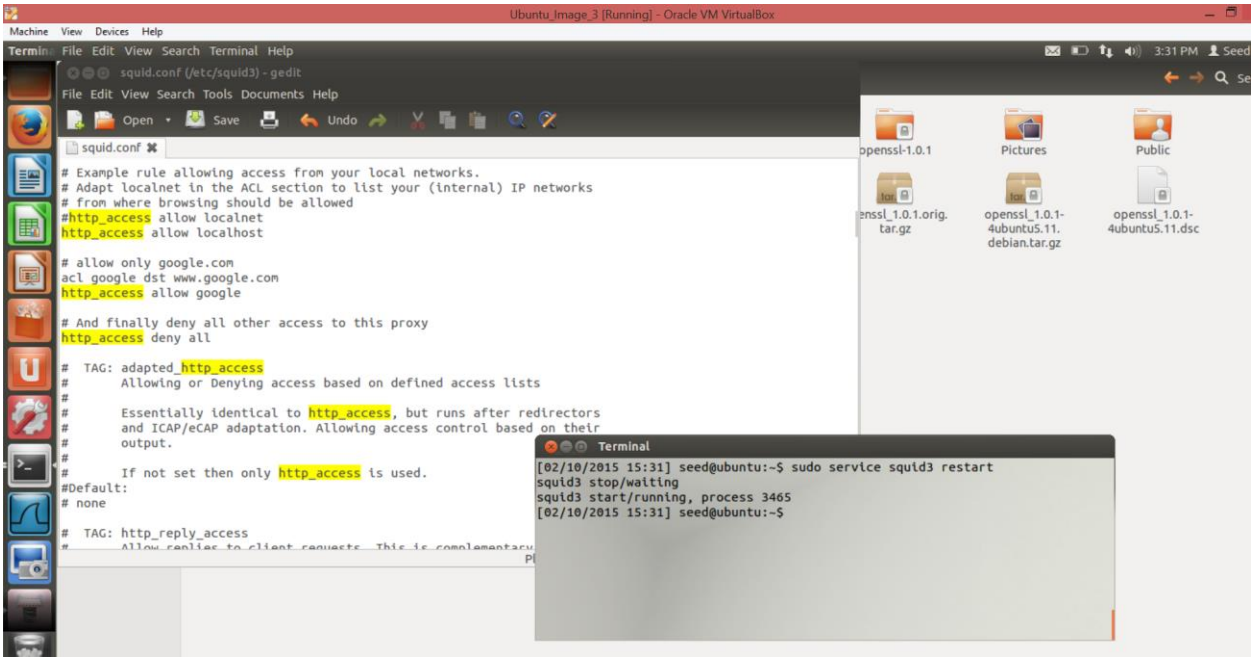
Observation: as the proxy for ubuntuA is set up as UbuntuB and UbuntuB squid3 is set up deny all so the page does not open.



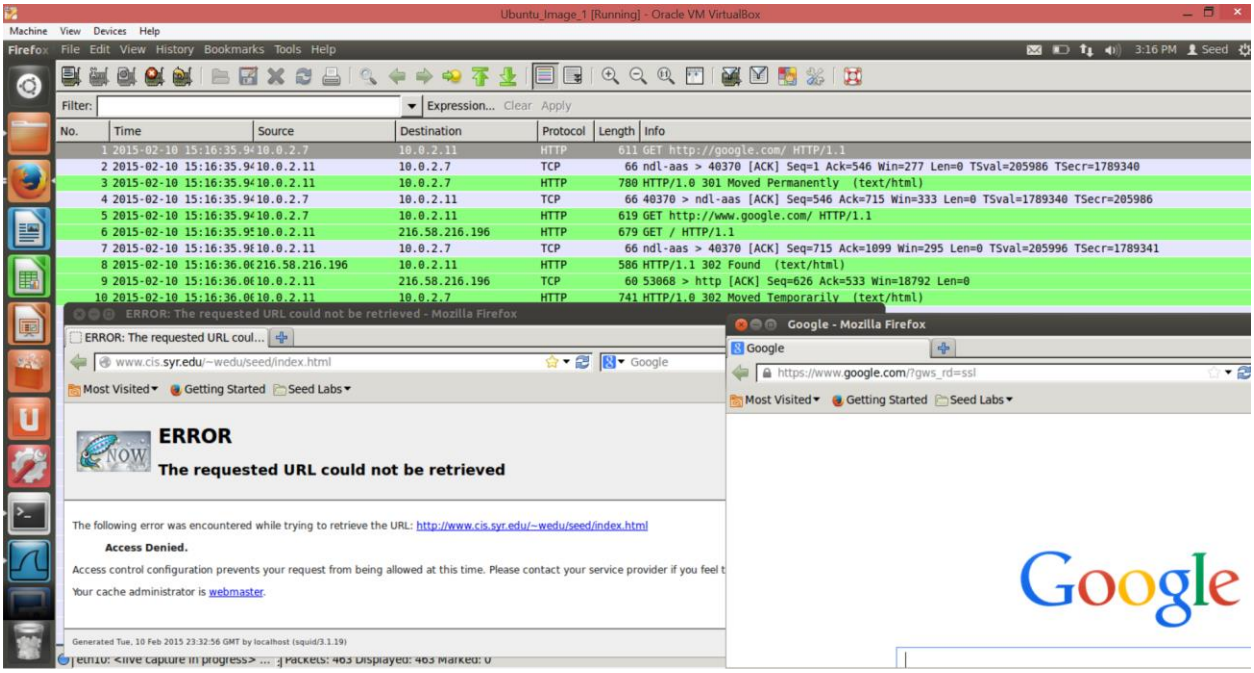
Squid3 is changed to allow all and and sqid process is restated.



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

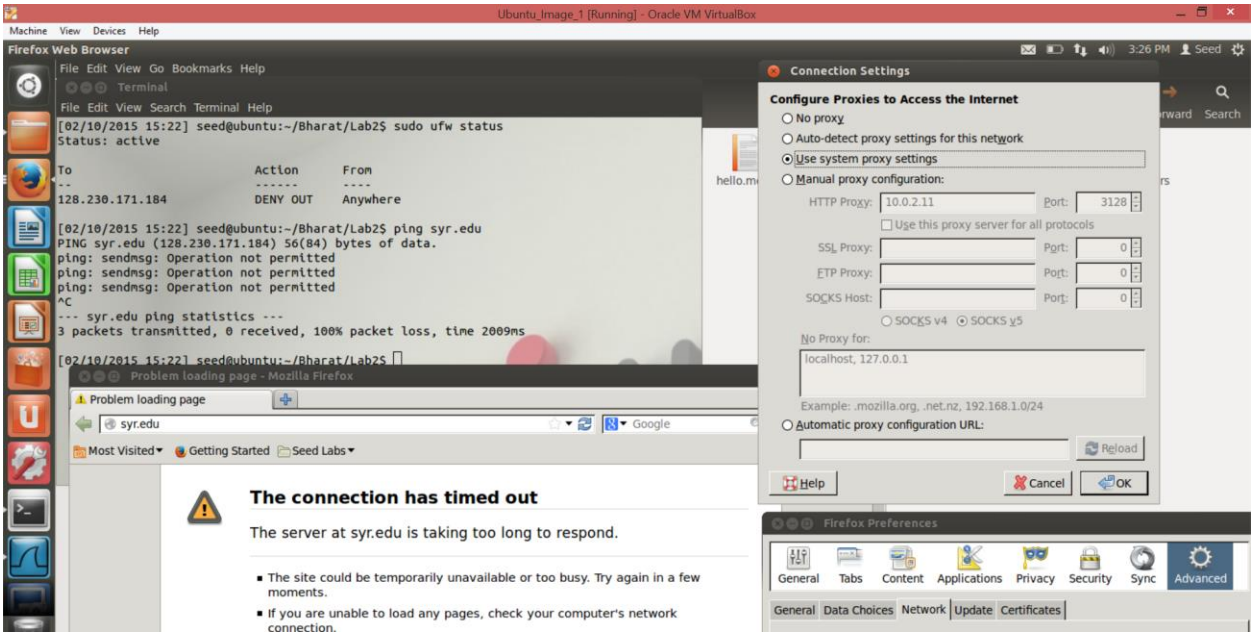


As squid3 changed to allow only google website and other external websites are blocked

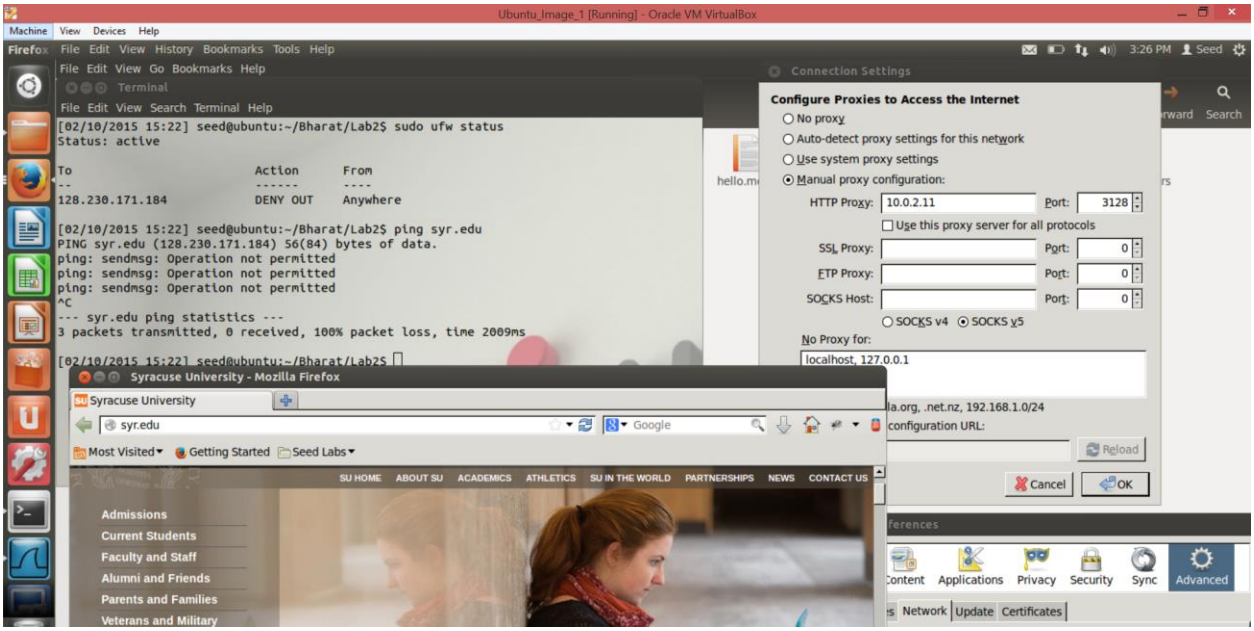


Now we can see google is being accessed and cis.syr.edu is blocked.

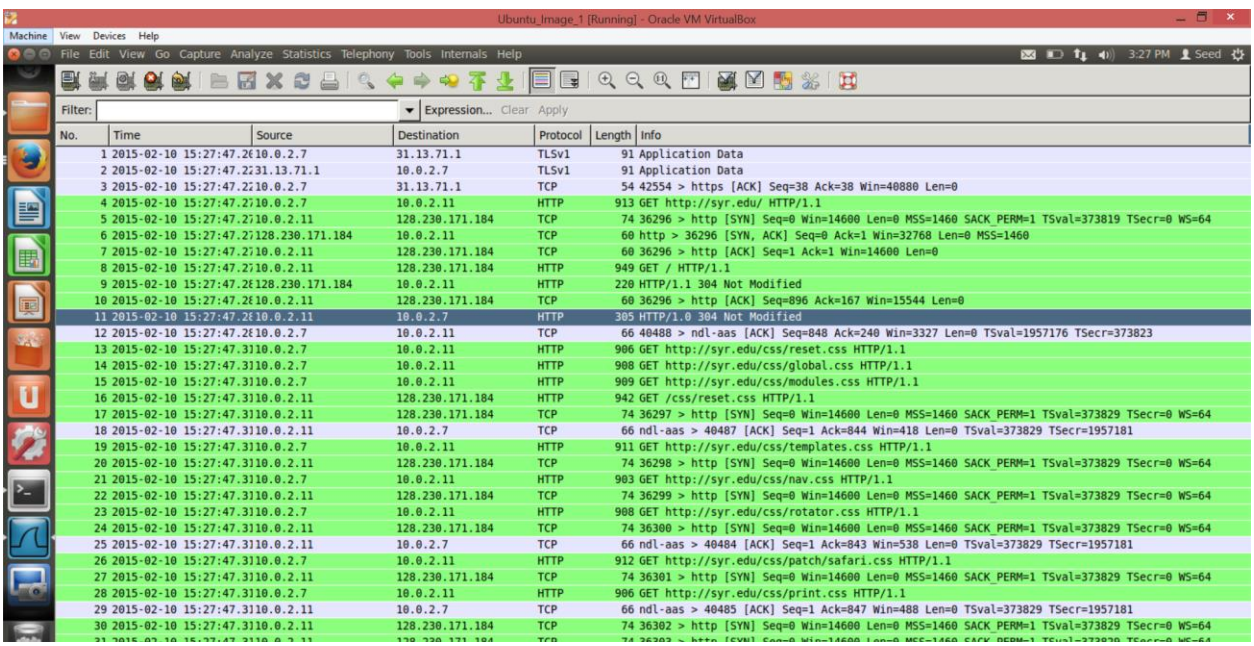
Task 4.b: Using Web Proxy to evade Firewall



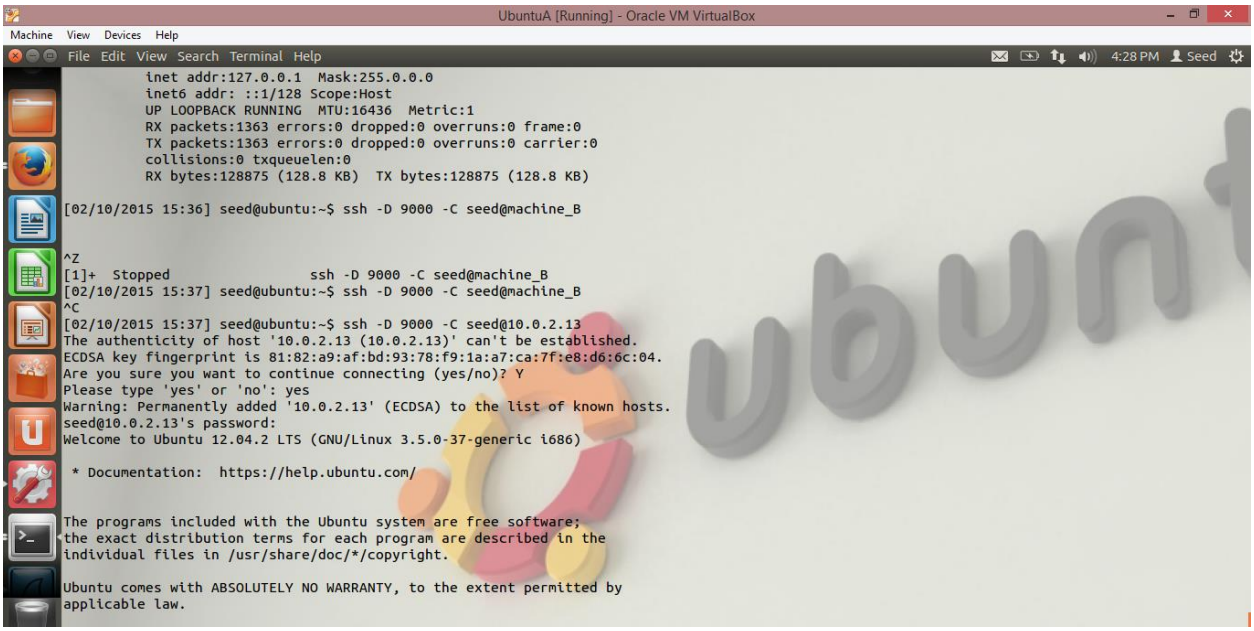
Ufw rule was setup as shown to block syr.edu and we are using system proxy is blocked.



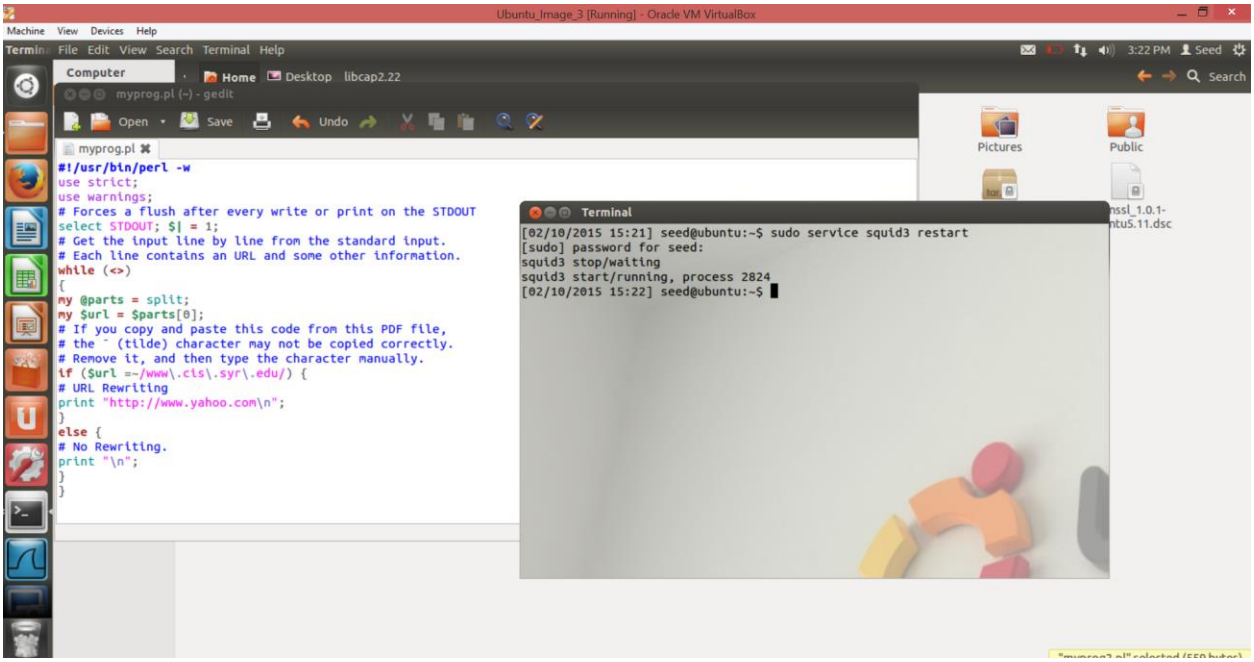
Ufw is used to blocked syr.edu. But we setup proxy settings as shown to use web proxy of UbuntuB. And so we can access syr.edu



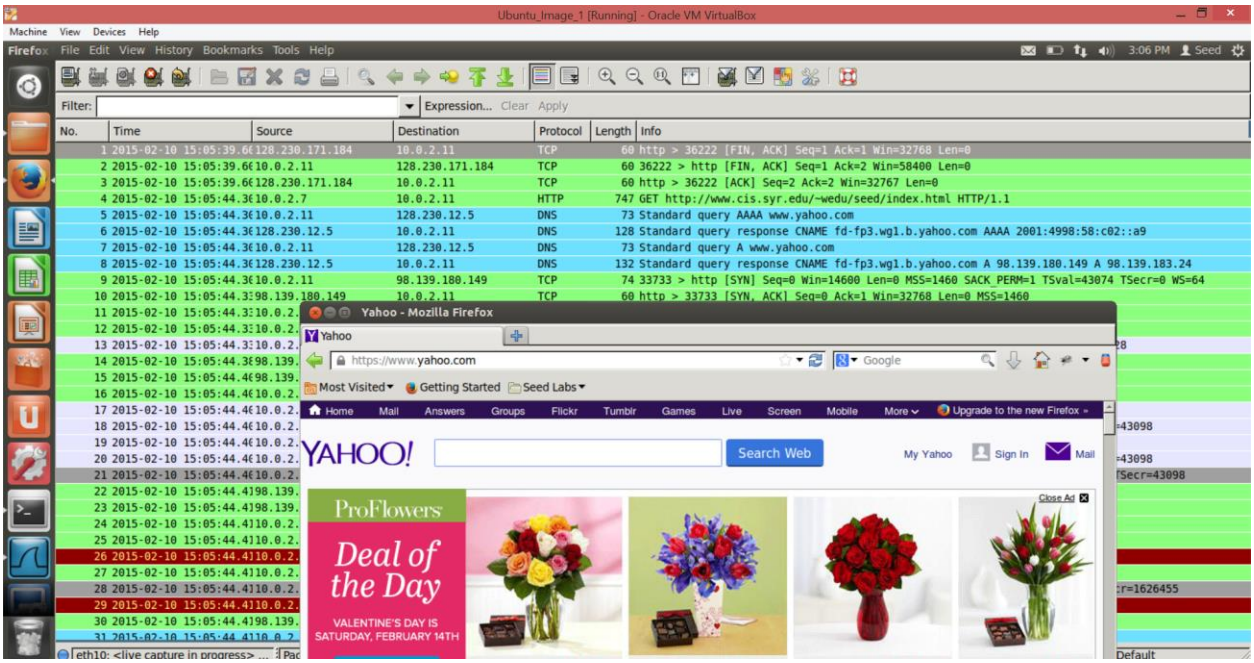
Wireshark session proves that there is traffic between UbuntuA and ubuntuB



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

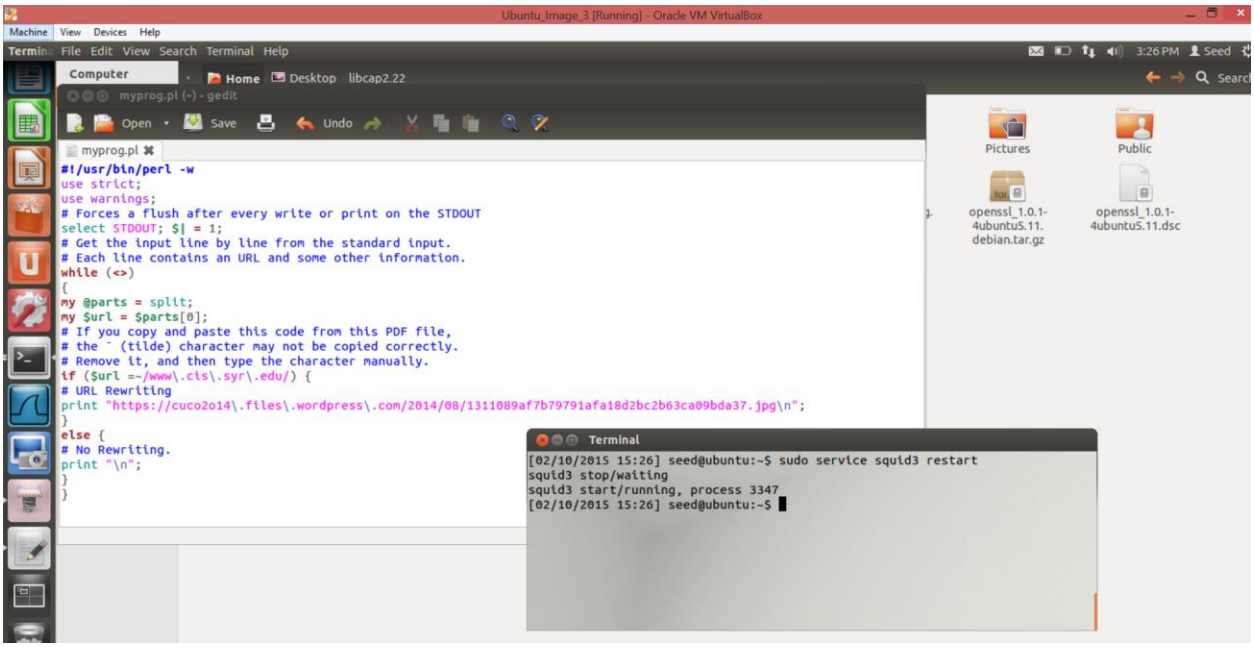


Accessing cis.syr.edu/~wedu/seed/index.html will redirect it to yahoo.com and wireshark session shows the redirection traffic.

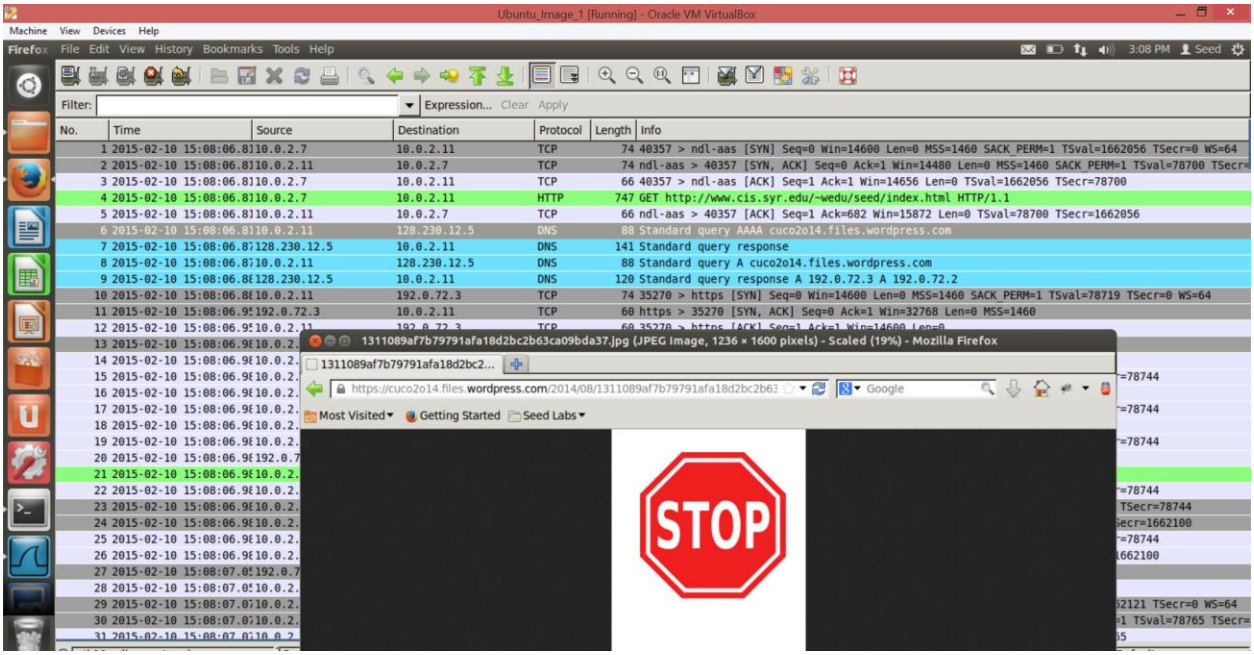
Question 5: If ufw blocks the TCP port 3128, can you still use web proxy to evade the firewall?

Yes we can change the port to evade firewall

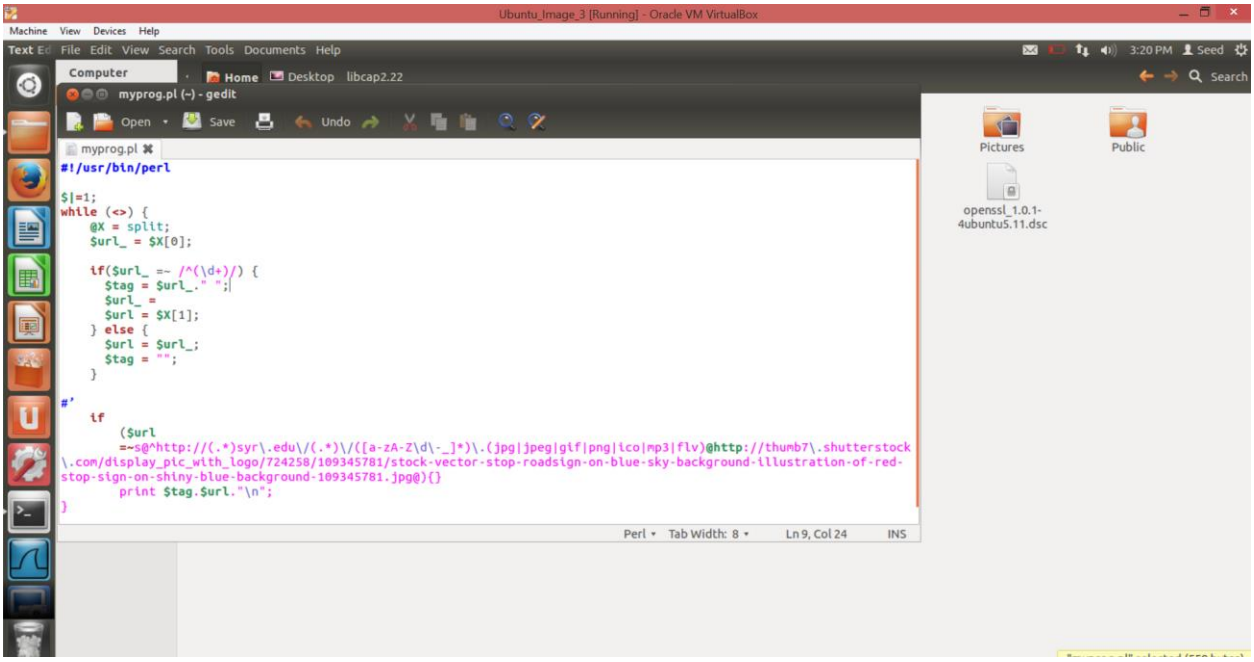
Task 4.c Url redirect to red stop sign



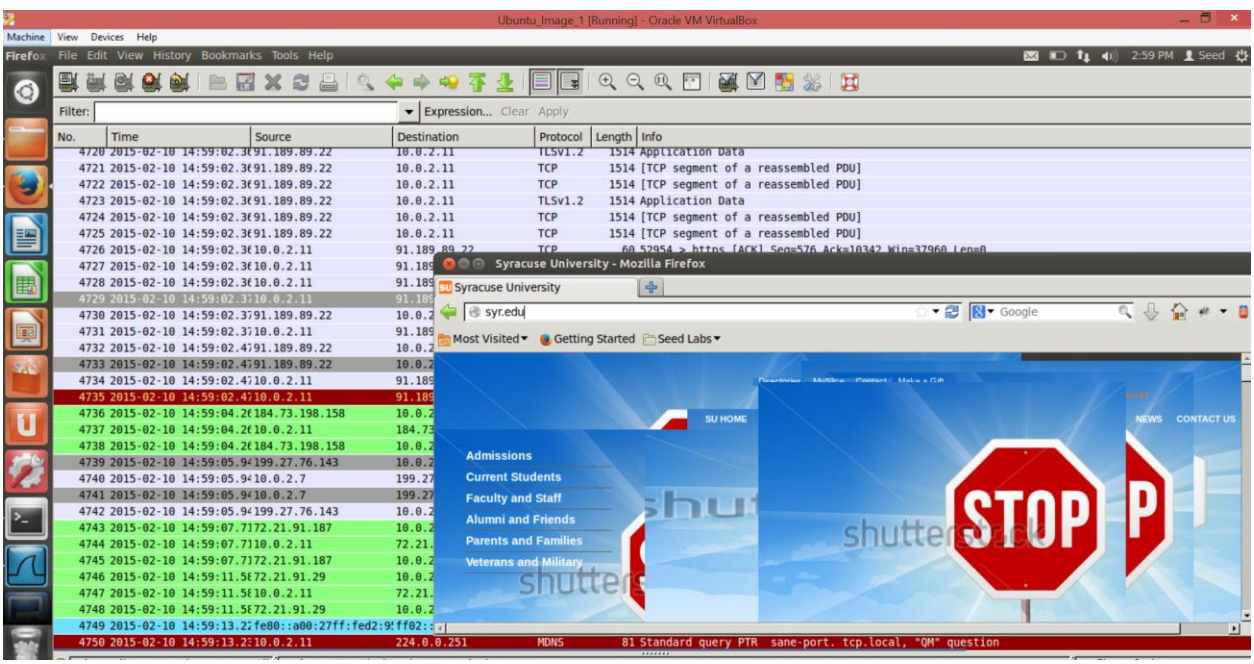
myprog.pl redirect url is changed to red_stop sign and squid3 is restarted.



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.