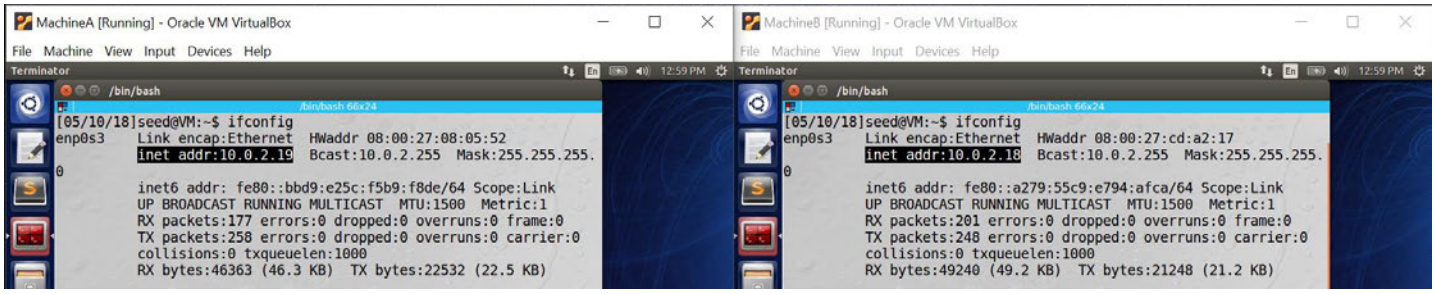
CS 491
Assignment 5

665336275 **Akshay Kataria**

Task 1: VM Setup

VM (Virtual Machine) IP address won't change for the entire lab experiment. We will implement the UFW (Uncomplicated Firewall) on MachineA / Client machine.

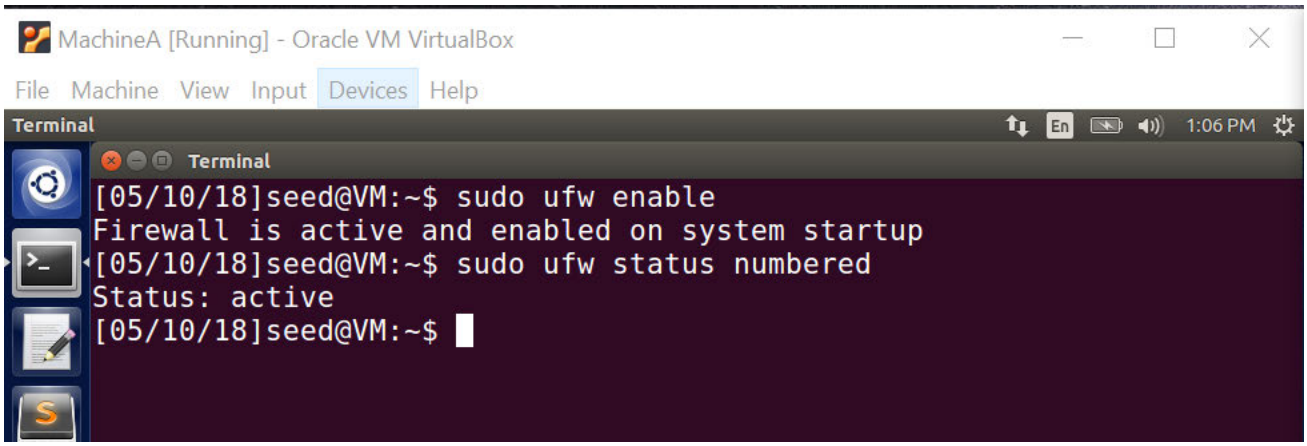
IP address: 10.0.2.19 IP address: 10.0.2.18
Client (MachineA) Server (MachineB)
|_____|



Obs: We can observe that the VM's are up and running and the 2 machines have been assigned the IP address.

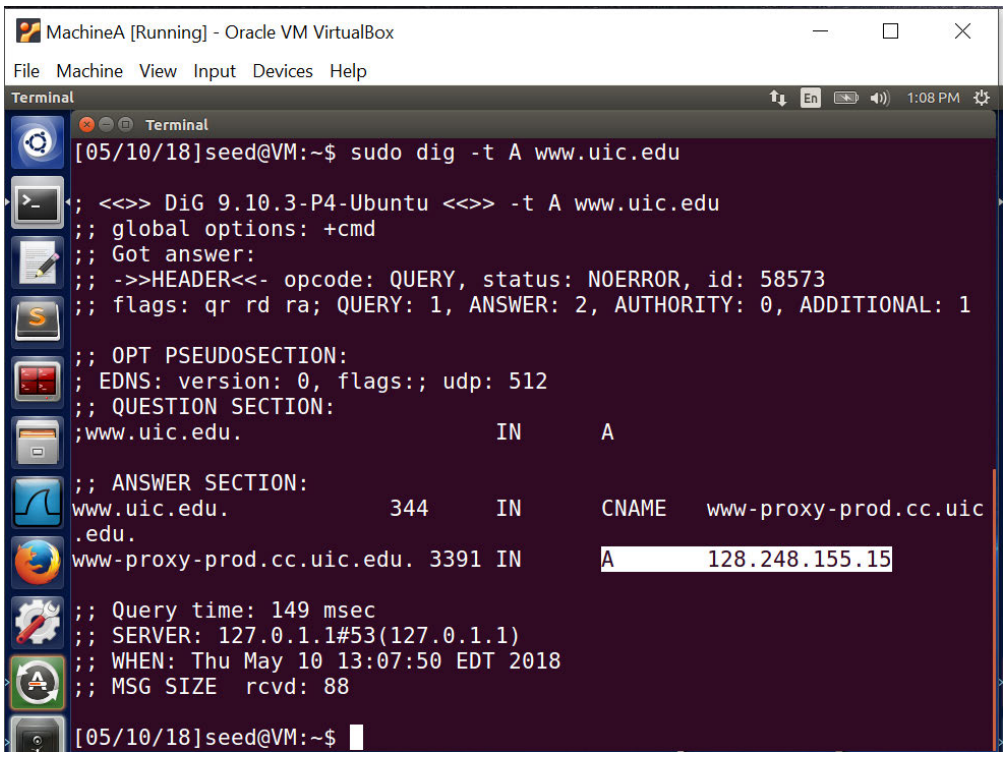
Task 2: Set up the Firewall

Organizations often block the internal users from accessing certin external sites (known as Egress Filtering). For this task, we will be working on the UFW (Uncomplicated Firewall) to block MachineA from reaching out to a particular website.

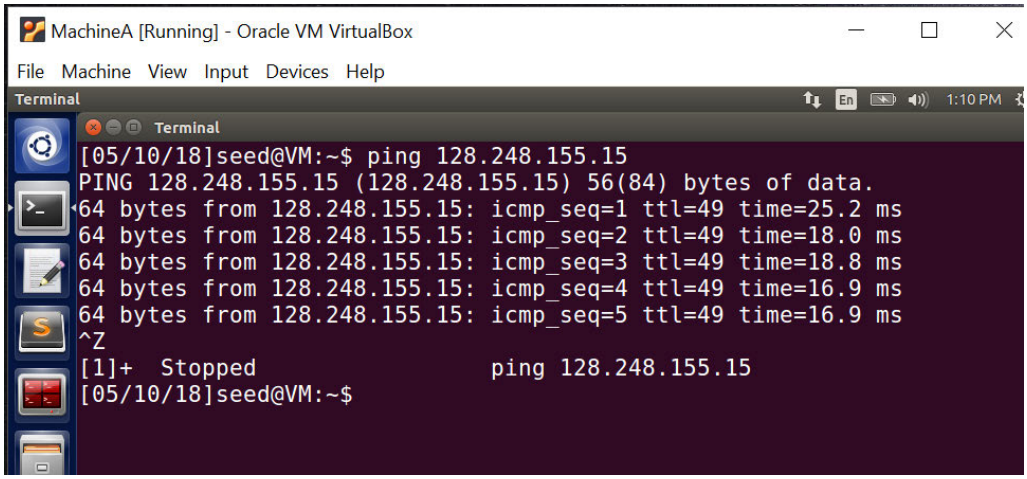


Obs: We can observe that UFW (Uncomplicated Firewall) have been installed and enabled on machineA and is working fine. There is no rule appearing after "sudo ufw status numbered" command because none has been mentioned yet.

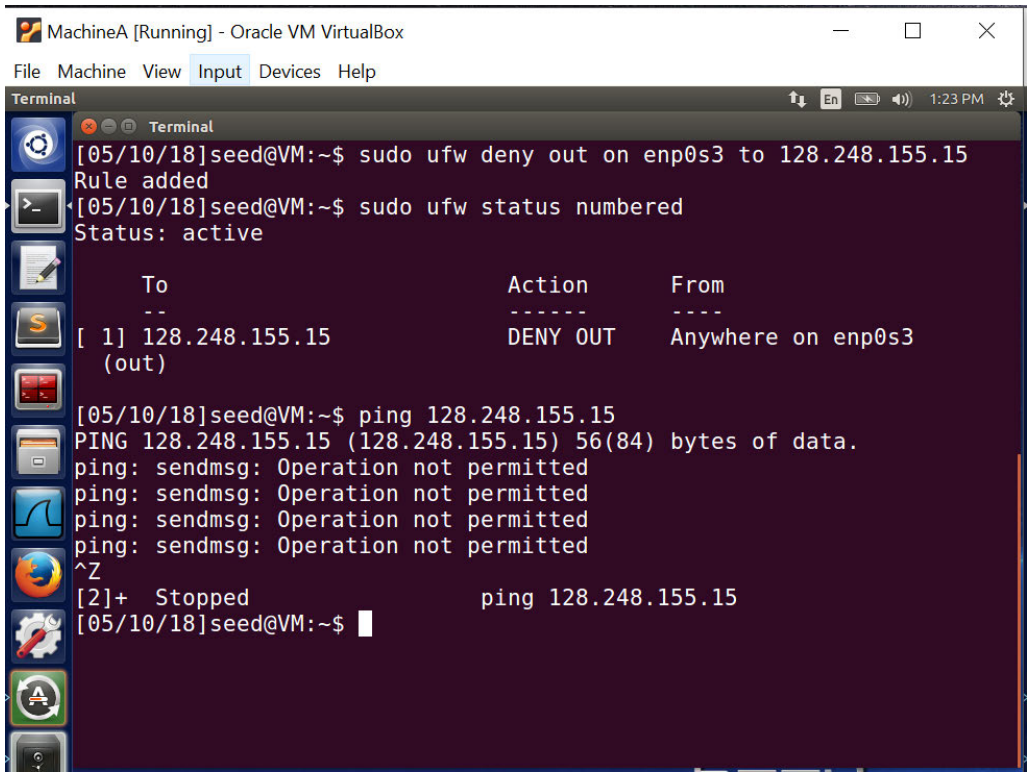
For this Lab experiment, we will try to search the ip address of UIC and will try to block the traffic from machineA to UIC website.



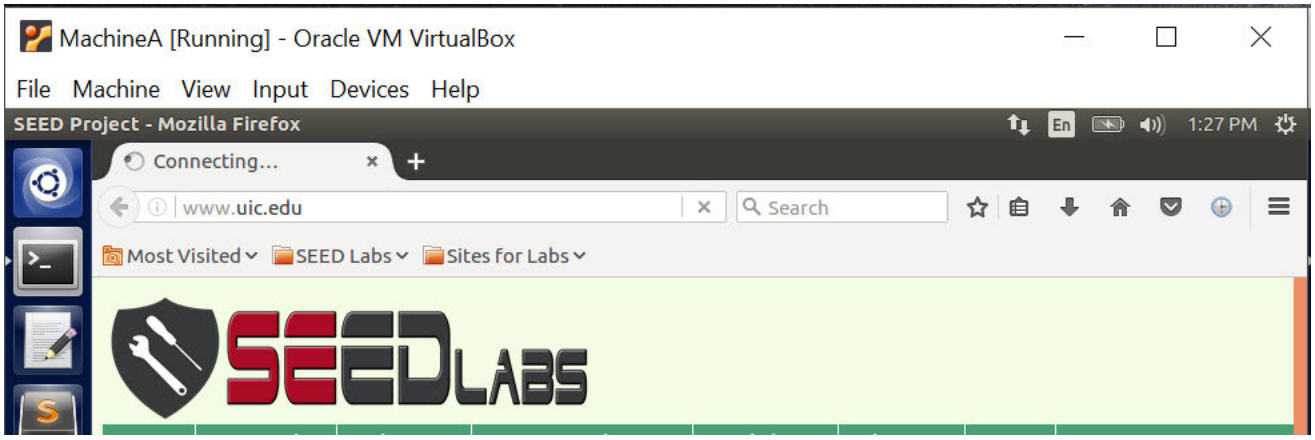
Obs: We are now trying to block access to a particular website from machie A. For our lab we are blocking "www.uic.edu". We found out it's address with the help of 'sudo dig -t A www.uic.edu' command. Address of website to be blocked is: 128.248.155.15.



Obs: We can observe that before any rule in UFW is initialized to block the traffic from machineA, we are able to ping "www.uic.edu" (128.248.155.15) with the help of the ping command. It shows that we able to access this website.



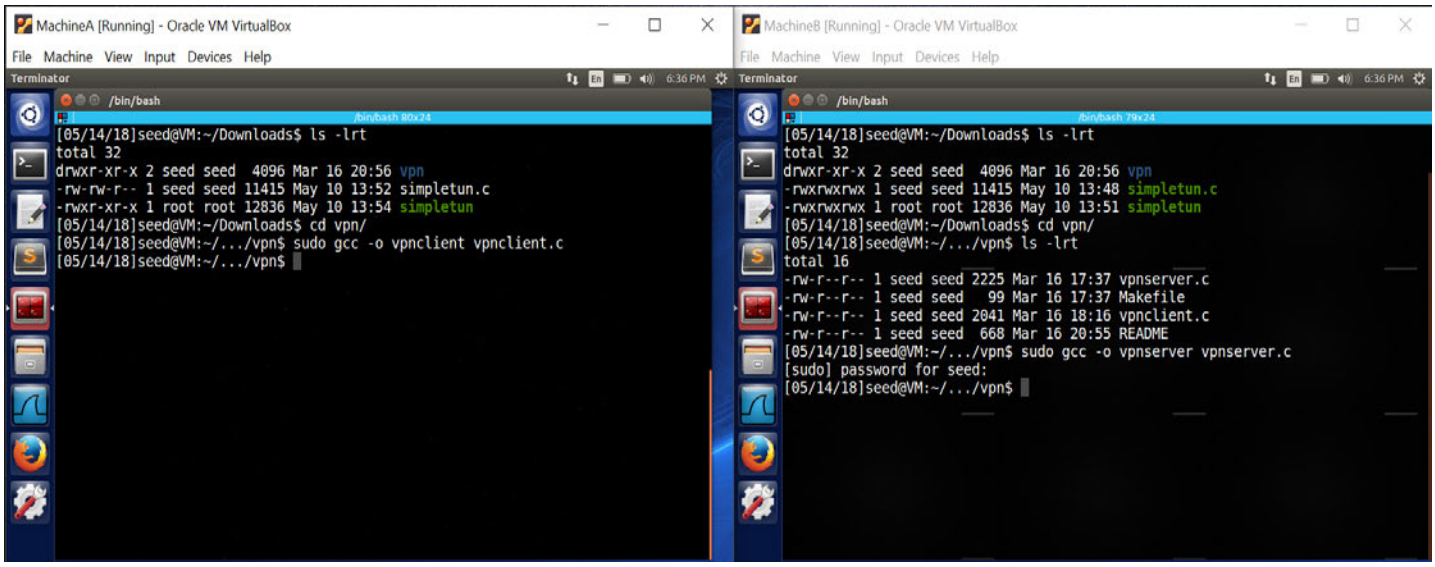
Obs: We can now observe that we are not able to ping “www.uic.edu” (128.248.155.15). This is because we have implemented a rule on ufw (sudo ufw deny out on enp0s3 to 128.248.155.15) that prevents machineA from reaching that particular website. This can be seen as rule 1 in the ufw rule table. You can also observe “Operation not permitted”.



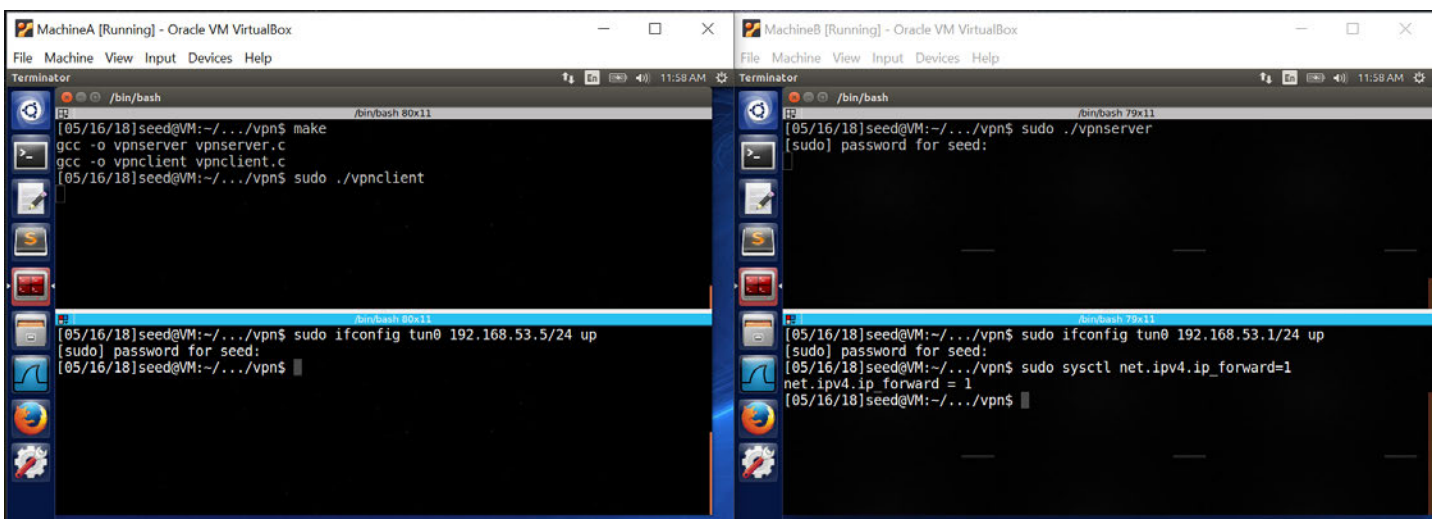
Obs: We can also observe from the web broswer that we are not able to reach out to “www.uic.edu” website.

Task 3: Bypassing Firewall using VPN

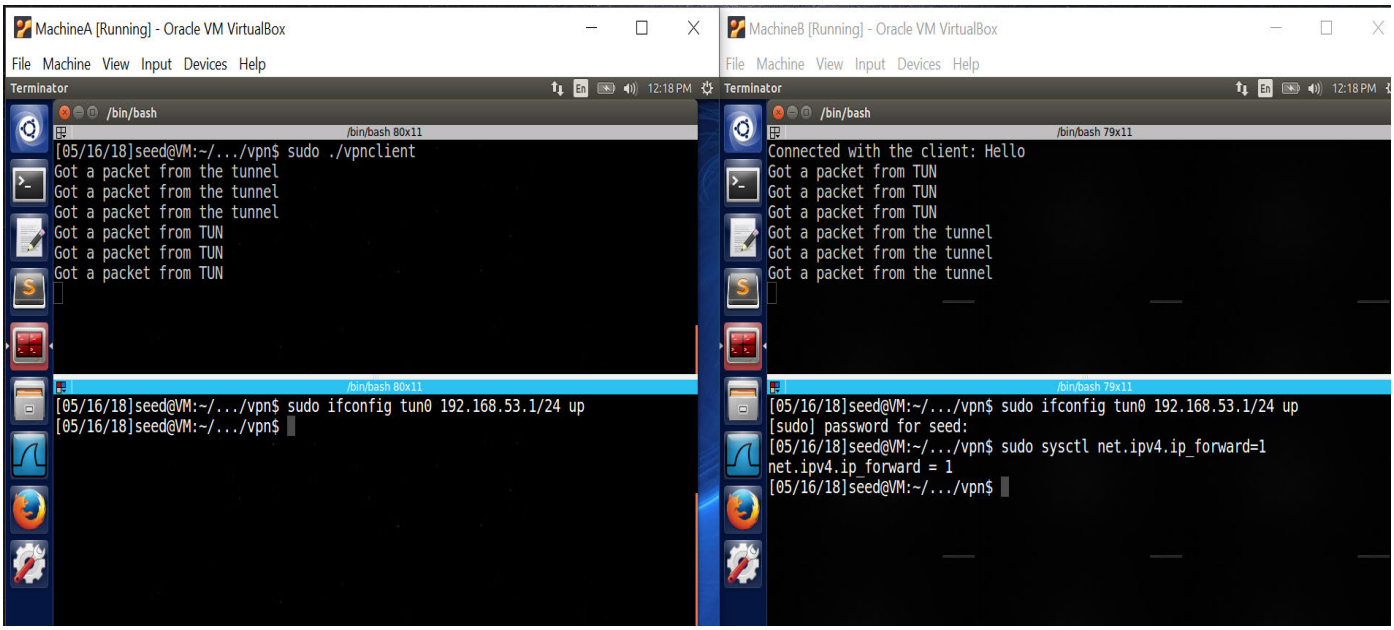
In this task, we will try to bypass the UFW firewall rules by creating a VPN tunnel between machineA (client) and the machineB (server).



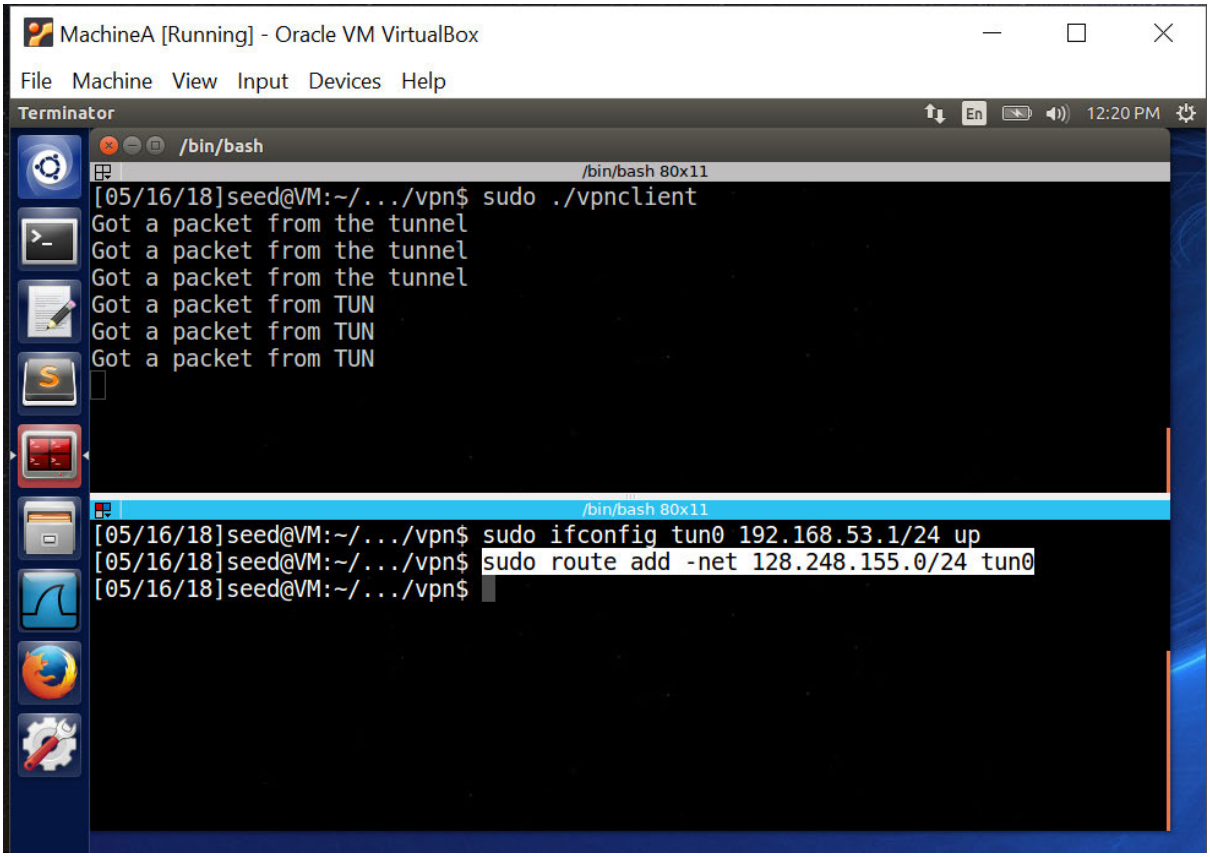
Obs: To implement a VPN tunnel in the virtual machines, “vpnclient.c” was downloaded and complied with root priviledges with the help of “sudo gcc -o vpnclient vpnclient.c” on the vpn client/machineA. Similarly, vpnserver.c was downloaded and compiled on vpn server/machineB.



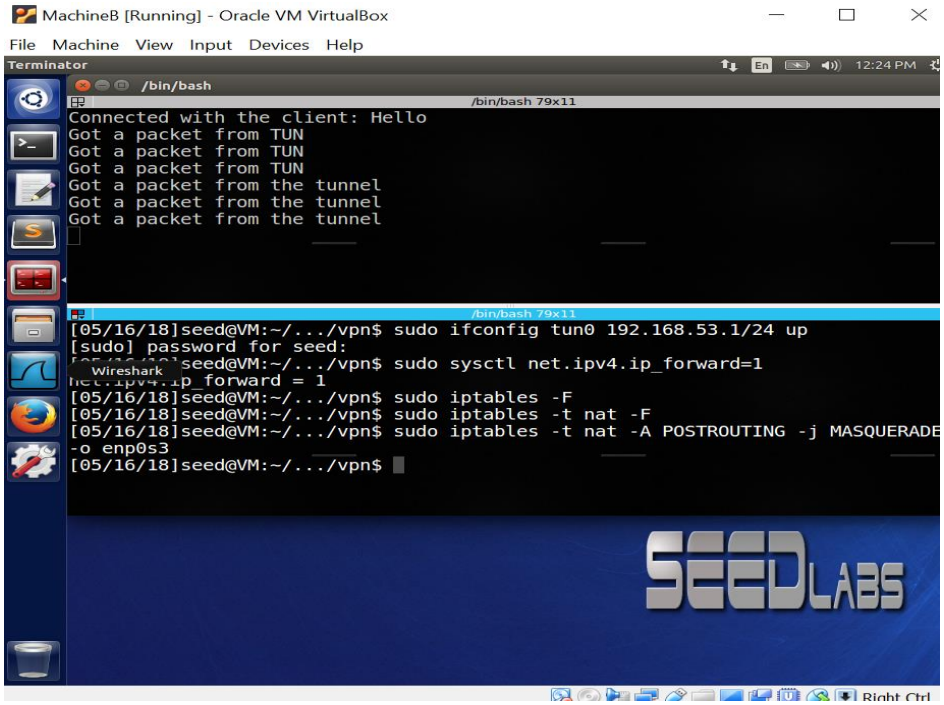
Obs: It can be observed that we first open up the vpnserver by “sudo ./vpnserver” and then assigned that interface an ip address by “sudo ifconfig tun0 192.168.53.1/24 up” in a different terminal window. Similary, we open the vpnclient machine interface for tunneling by “sudo ./vpnclient” and then assigned that interface an ip address by “sudo ifconfig tun0 192.168.53.5/24 up”. After this setup, we want to make sure that we have enabled the IP forwarding on the VPN server using command “sudo sysctl net.ipv4.ip_forward=1”.



Obs: It can be observed that as soon as the tunnel was up, client and server started sending packets to each other. Confirms that the tunnel has been successfully established.



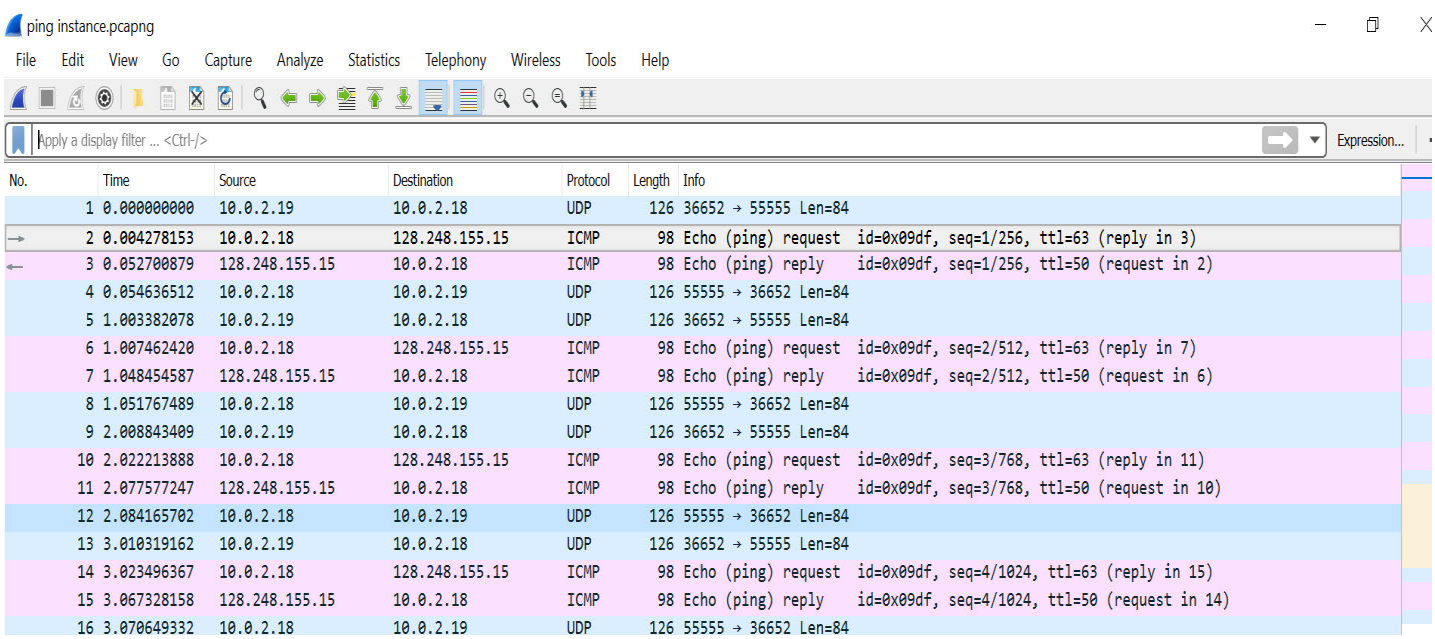
Obs: Now, we can configure a route from the VPN client to the VPN server via VPN tunnel interface “tun0” with the help of the “sudo route add -net 128.248.155.0/24 tun0”.



Obs: It can be observed that after the addition of route, we need to add the NAT on VPN Server. This can be done with the help of following commands:

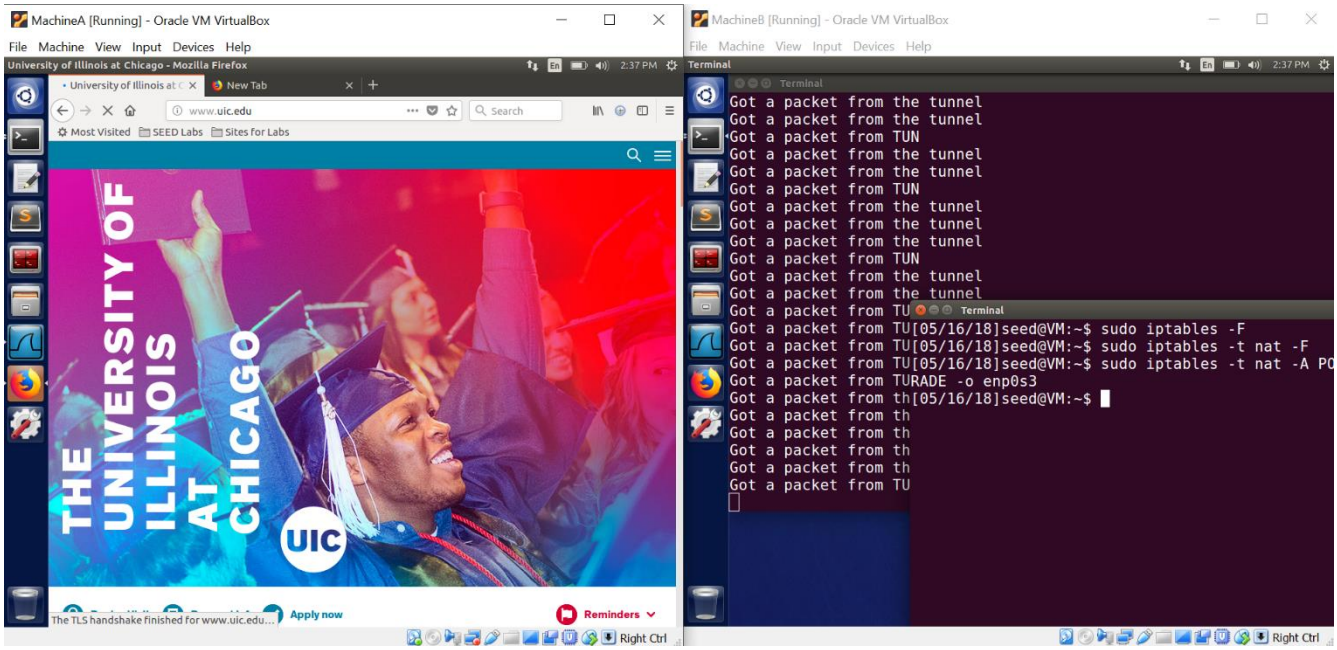
- sudo iptables -F
- sudo iptables -t nat -F
- sudo iptables -t nat -A POSTROUTING -j MASQUERADE -o enp0s3

After the setup is complete, you can ping the external website that has been blocked on the VPN Client and but can be reached out through VPN Server.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.19	10.0.2.18	UDP	126	36652 → 55555 Len=84
2	0.004278153	10.0.2.18	128.248.155.15	ICMP	98	Echo (ping) request id=0x09df, seq=1/256, ttl=63 (reply in 3)
3	0.052700879	128.248.155.15	10.0.2.18	ICMP	98	Echo (ping) reply id=0x09df, seq=1/256, ttl=50 (request in 2)
4	0.054636512	10.0.2.18	10.0.2.19	UDP	126	55555 → 36652 Len=84
5	1.003382078	10.0.2.19	10.0.2.18	UDP	126	36652 → 55555 Len=84
6	1.007462420	10.0.2.18	128.248.155.15	ICMP	98	Echo (ping) request id=0x09df, seq=2/512, ttl=63 (reply in 7)
7	1.048454587	128.248.155.15	10.0.2.18	ICMP	98	Echo (ping) reply id=0x09df, seq=2/512, ttl=50 (request in 6)
8	1.051767489	10.0.2.18	10.0.2.19	UDP	126	55555 → 36652 Len=84
9	2.008843409	10.0.2.19	10.0.2.18	UDP	126	36652 → 55555 Len=84
10	2.022213888	10.0.2.18	128.248.155.15	ICMP	98	Echo (ping) request id=0x09df, seq=3/768, ttl=63 (reply in 11)
11	2.077577247	128.248.155.15	10.0.2.18	ICMP	98	Echo (ping) reply id=0x09df, seq=3/768, ttl=50 (request in 10)
12	2.084165702	10.0.2.18	10.0.2.19	UDP	126	55555 → 36652 Len=84
13	3.010319162	10.0.2.19	10.0.2.18	UDP	126	36652 → 55555 Len=84
14	3.023496367	10.0.2.18	128.248.155.15	ICMP	98	Echo (ping) request id=0x09df, seq=4/1024, ttl=63 (reply in 15)
15	3.067328158	128.248.155.15	10.0.2.18	ICMP	98	Echo (ping) reply id=0x09df, seq=4/1024, ttl=50 (request in 14)
16	3.070649332	10.0.2.18	10.0.2.19	UDP	126	55555 → 36652 Len=84

Obs: It can be clearly observed from the packets captured from wireshark that whenever VPN client (10.0.2.19) tries to ping 128.248.155.15, the ICMP ECHO Request traffic goes through the VPN server (10.0.2.18) and ICMP ECHO Reply reaches VPN Client via VPN Server only. This shows that the tunnel is working perfectly and we are able to bypass the UFW firewall as is doesn't account for the DPI (Deep Packet Inspection).



Obs: I have also tried to reach out to the blocked website through the browser and as we can observe, we are able to resolve the domain name for that website. It can be safely presumed that the VPN tunnel is working just fine.

451	13.698086182	10.0.2.19	192.168.1.254	DNS	71	Standard query 0x5cf1 A www.uic.edu
452	13.700316804	10.0.2.19	192.168.1.254	DNS	71	Standard query 0xb31e AAAA www.uic.edu
453	13.750581446	192.168.1.254	10.0.2.19	DNS	119	Standard query response 0x5cf1 A www.uic.edu CNAME www-proxy-prod.cc.uic.edu A 128.2
454	13.750602165	192.168.1.254	10.0.2.19	DNS	156	Standard query response 0xb31e AAAA www.uic.edu CNAME www-proxy-prod.cc.uic.edu SOA
455	13.790796389	10.0.2.19	72.21.91.29	TCP	54	46106 → 80 [FIN, ACK] Seq=1 Ack=1 Win=29200 Len=0
456	13.792782734	10.0.2.19	52.10.151.19	TLSv1…	85	Encrypted Alert

Obs: From the captured wireshark packets, you can observe that the query about www.uic.edu (128.248.155.15) that was initiated by VPN Client (10.0.2.19) is getting resolved via VPN Server. It can be deduced that the VPN tunnel is working fine.