Vikash Deo
N15708475
LAB7 - IPTables

For outgoing traffic (from 10.20.111.0/24 to 10.10.111.0/24) - your internal machine should be able to communicate with the external network and the external machines without restrictions.

The screenshot shows the stateful rule being appended to the FORWARD chain that allows all the traffic to go from the 10.20.111.0/24 incoming at eth1 and outgoing at eth0.

iptables –A OUTPUT –s 10.10.111.0/24 –o eth0 –m state --state NEW,ESTABLISHED –j ACCEPT

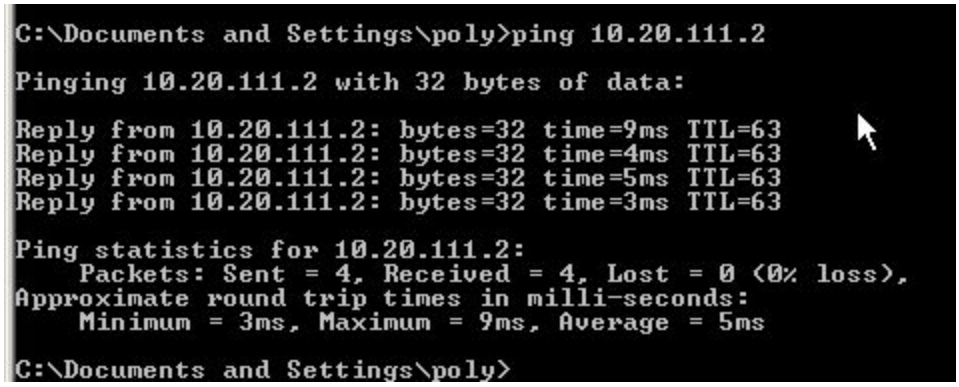
```
router:~# iptables -A OUTPUT -s 10.20.111.0/24 -o eth0 -m state --state NEW,ESTA
BLISHED -j ACCEPT
router:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  10.20.111.0/24       anywhere            state NEW,ESTABLISH
ED
router:~# _
```

Part A

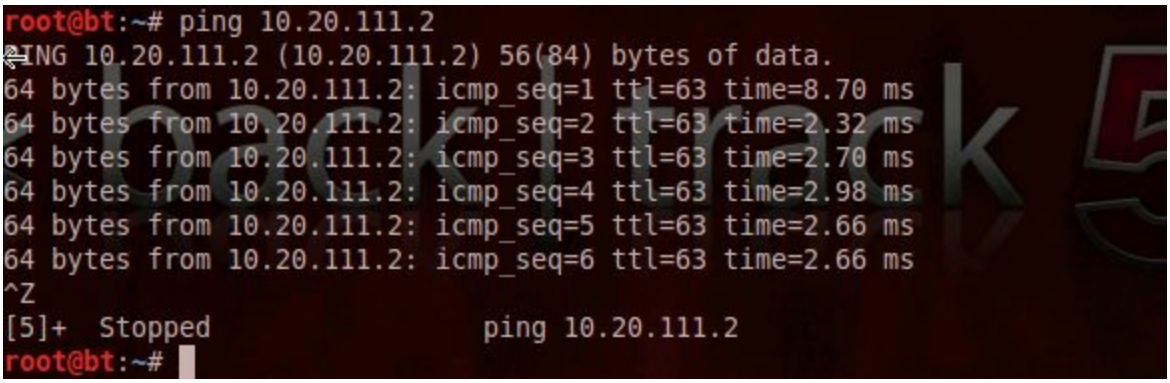
1. **The internal machines should respond to a ping from 10.10.111.0/24**
   The entries made were verified by using the iptables –L command

```
router:~# iptables -A INPUT -i eth0 -p icmp --icmp-type echo-reply -s 10.10.111.
0/24 -m state --state NEW,ESTABLISHED -j ACCEPT
router:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     icmp --  10.10.111.0/24       anywhere            icmp echo-reply sta
te NEW,ESTABLISHED

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  10.10.111.0/24       anywhere            state NEW,ESTABLISH
ED
router:~# _
```

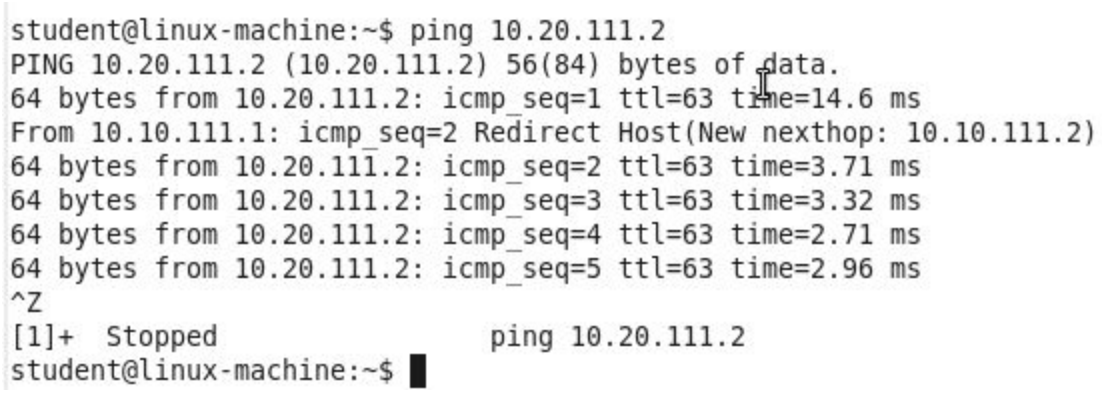
Ping from windows machine.



Ping from backtrack machine.



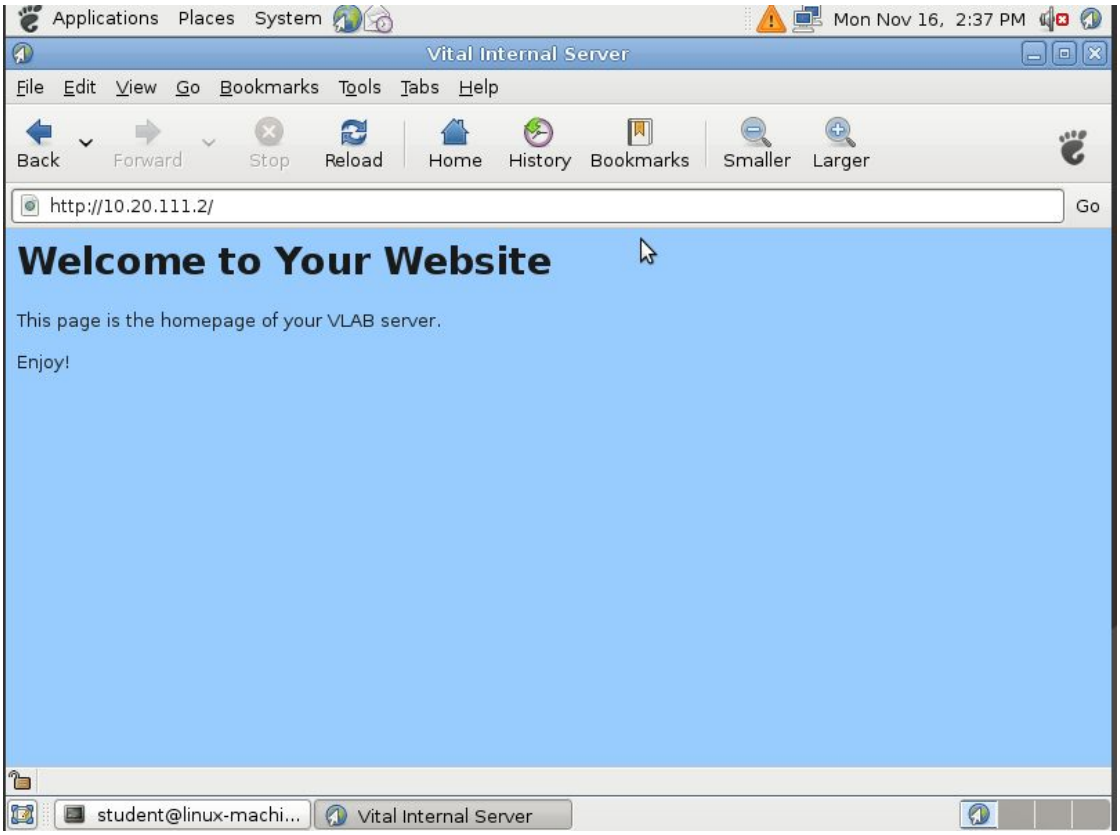
Ping from student linux machine.



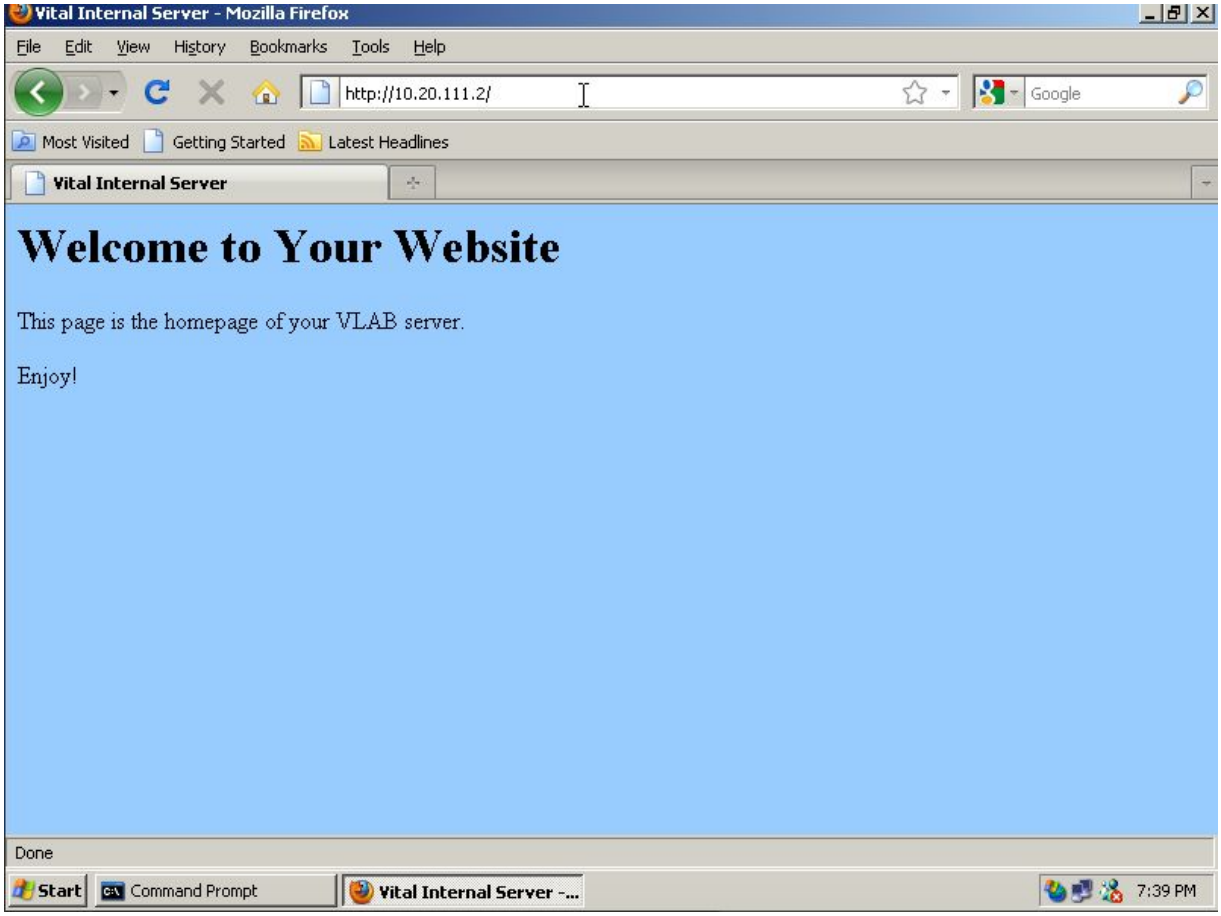
2. The internal machine (10.20.111.2) should accept all incoming SSH and HTTP requests from 10.10.111.0/24

```
router:~# iptables -A INPUT -i eth0 -p tcp -s 10.10.111.0/24 -j ACCEPT -m state
--state NEW --dport 22 --sport 1024:65535
router:~# iptables -A INPUT -i eth0 -p tcp -s 10.10.111.0/24 -j ACCEPT -m state
--state NEW --dport 80 --sport 1024:65535
router:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     icmp --  10.10.111.0/24       anywhere            icmp echo-reply sta
te NEW,ESTABLISHED
ACCEPT     tcp  --  10.10.111.0/24       anywhere            state NEW tcp spts:
1024:65535 dpt:ssh
ACCEPT     tcp  --  10.10.111.0/24       anywhere            state NEW tcp spts:
1024:65535 dpt:www

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  10.10.111.0/24       anywhere            state NEW,ESTABLISH
ED
router:~# _
```

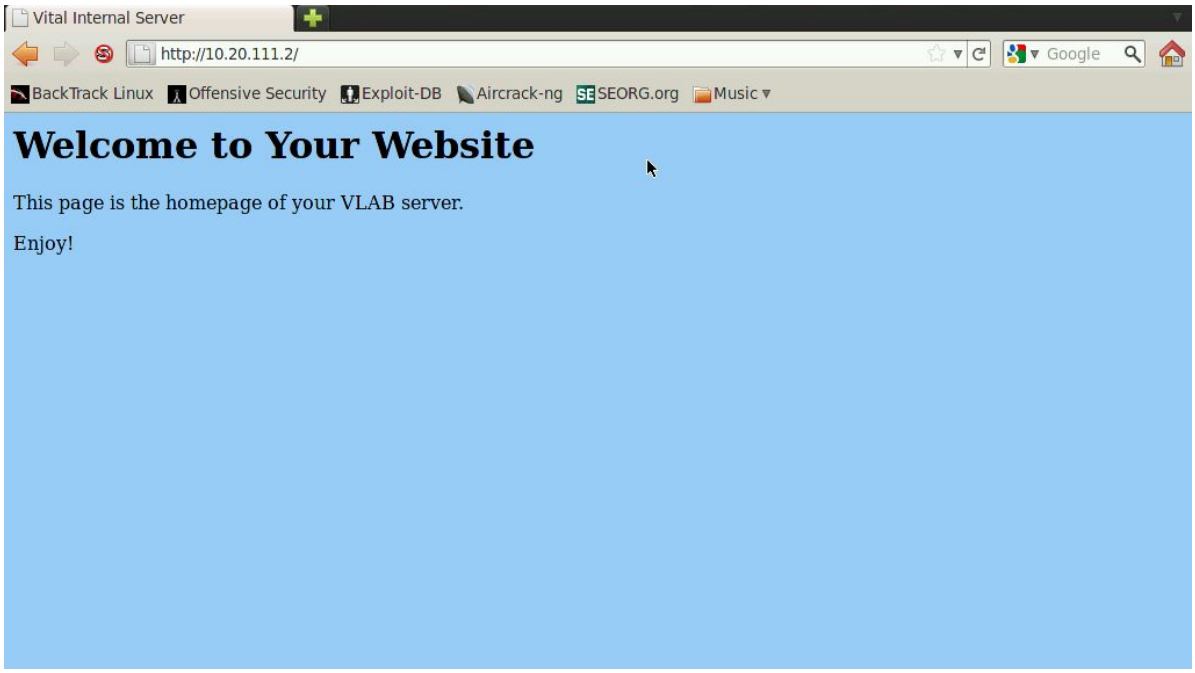
HTTP from Linux machine



HTTP from Windows machine



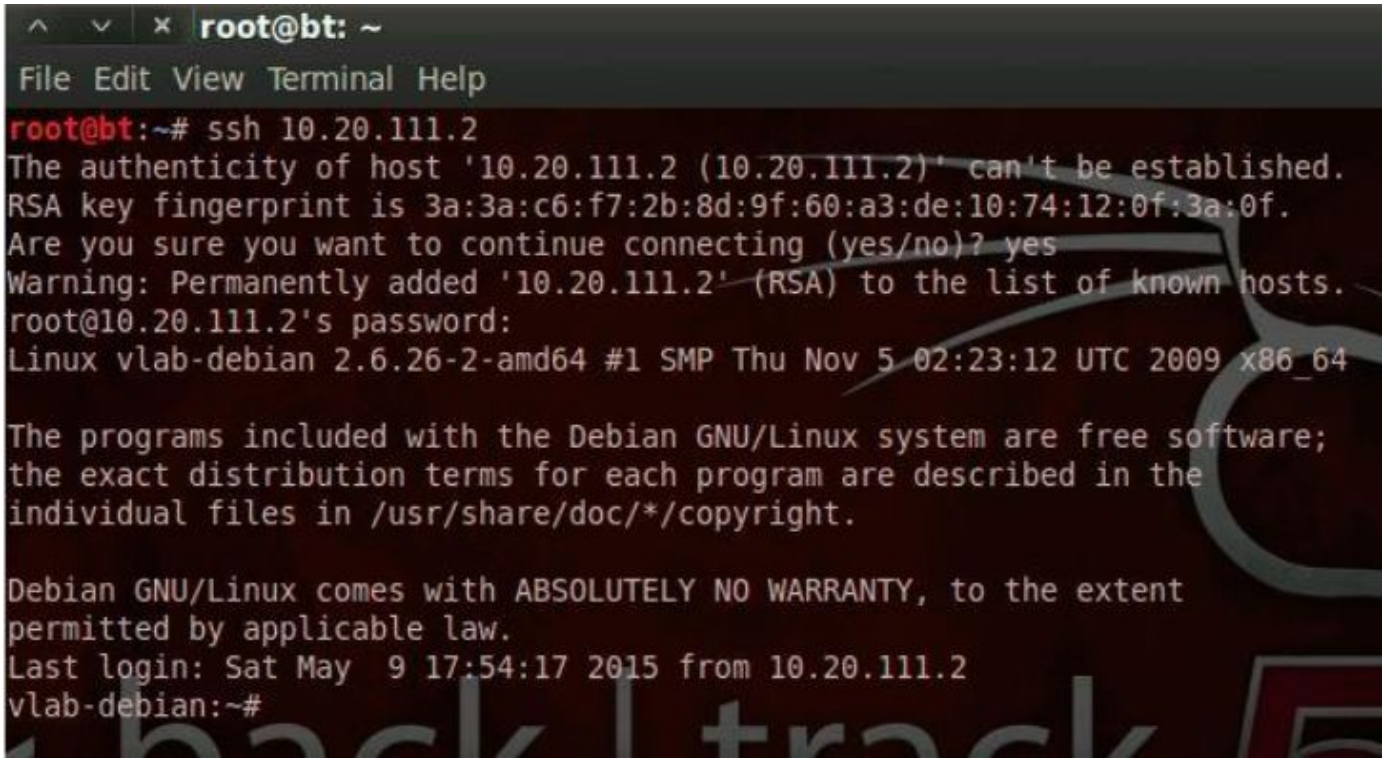
HTTP from Backtrack Machine



SSH from an outer machine (Linux Machine)

```
student@linux-machine:~$ sudo ssh 10.20.111.2
The authenticity of host '10.20.111.2 (10.20.111.2)' can't be established.
RSA key fingerprint is 3a:3a:c6:f7:2b:8d:9f:60:a3:de:10:74:12:0f:3a:0f.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.20.111.2' (RSA) to the list of known hosts.
root@10.20.111.2's password:
Permission denied, please try again.
root@10.20.111.2's password:
Permission denied, please try again.
root@10.20.111.2's password:
Linux vlab-debian 2.6.26-2-amd64 #1 SMP Thu Nov 5 02:23:12 UTC 2009 x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Nov 16 14:44:47 2015 from 10.10.111.107
vlab-debian:~#
```

SSH form backtrack machine.



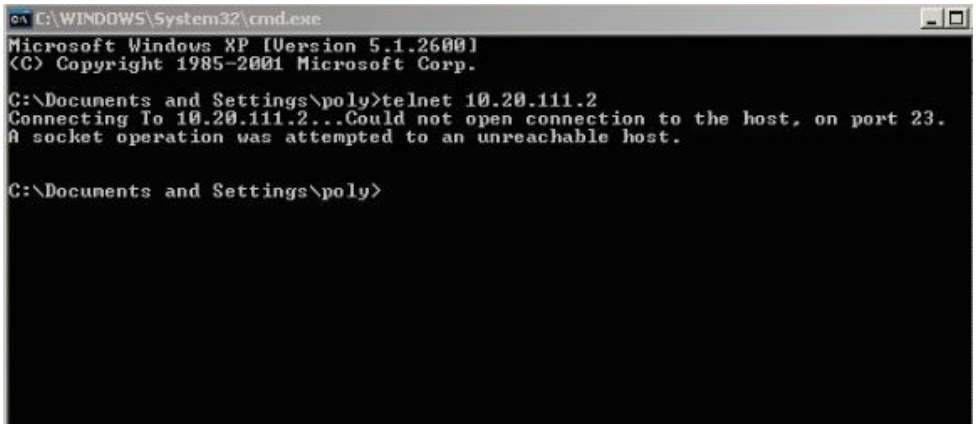
3. The internal machine should accept telnet connections from the BT Machine only.

```
router:~#
router:~# iptables -A INPUT -i eth0 -p tcp -s 10.10.111.107/24 -j ACCEPT -m stat
e --state NEW --dport telnet --sport 1024:65535
router:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     icmp --  10.10.111.0/24       anywhere            icmp echo-reply sta
te NEW,ESTABLISHED
ACCEPT     tcp  --  10.10.111.0/24       anywhere            state NEW tcp spts:
1024:65535 dpt:ssh
ACCEPT     tcp  --  10.10.111.0/24       anywhere            state NEW tcp spts:
1024:65535 dpt:www
ACCEPT     tcp  --  10.10.111.0/24       anywhere            state NEW tcp spts:
1024:65535 dpt:telnet
ACCEPT     tcp  --  10.10.111.0/24       anywhere            state NEW tcp spts:
1024:65535 dpt:telnet

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  10.10.111.0/24       anywhere            state NEW,ESTABLISH
ED
router:~# _
```

Telnet from backtrack machine.

```
root@bt:~# telnet 10.20.111.2
Trying 10.20.111.2...
Connected to 10.20.111.2.
Escape character is '^]'.
Debian GNU/Linux 5.0
vlab-debian login:
```

Telnet from windows is unsuccessful.



Part B

Flushing all the rules from the iptables by using the following commands.
iptables -F
iptables -t nat -F

I ran the command "iptables -t nat -L -nv" to make sure that there were not any previous set rules.

```
router:~#
router:~# iptables -F
router:~# iptables -t nat -F
router:~# iptables -t nat -L -nv
Chain PREROUTING (policy ACCEPT 35 packets, 4844 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 4 packets, 228 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 2 packets, 142 bytes)
 pkts bytes target     prot opt in     out     source               destination

router:~# _
```

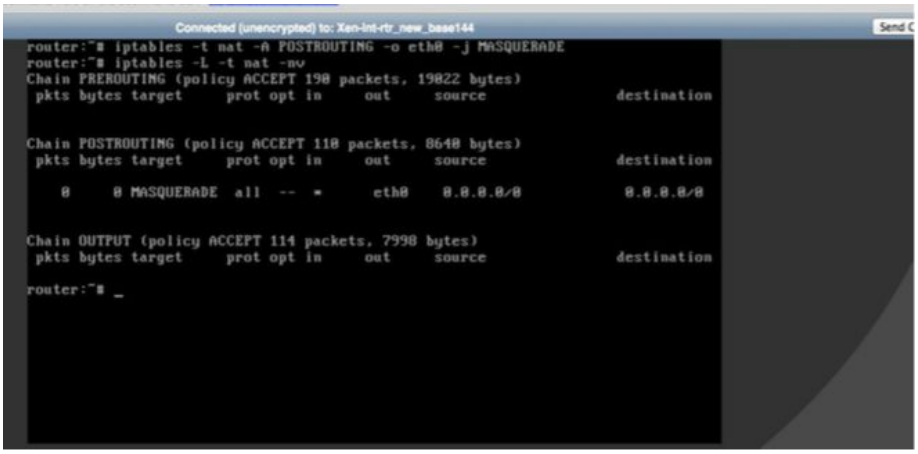
We translate traffic from the internal machine with source IP address 10.20.111.2 to a source ip address of 10.10.111.x (it is the address of the outside interface of the firewall).

iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

where -t nat : allows to alter NAT table of iptables

POSTROUTING: address translation occurs before routing.

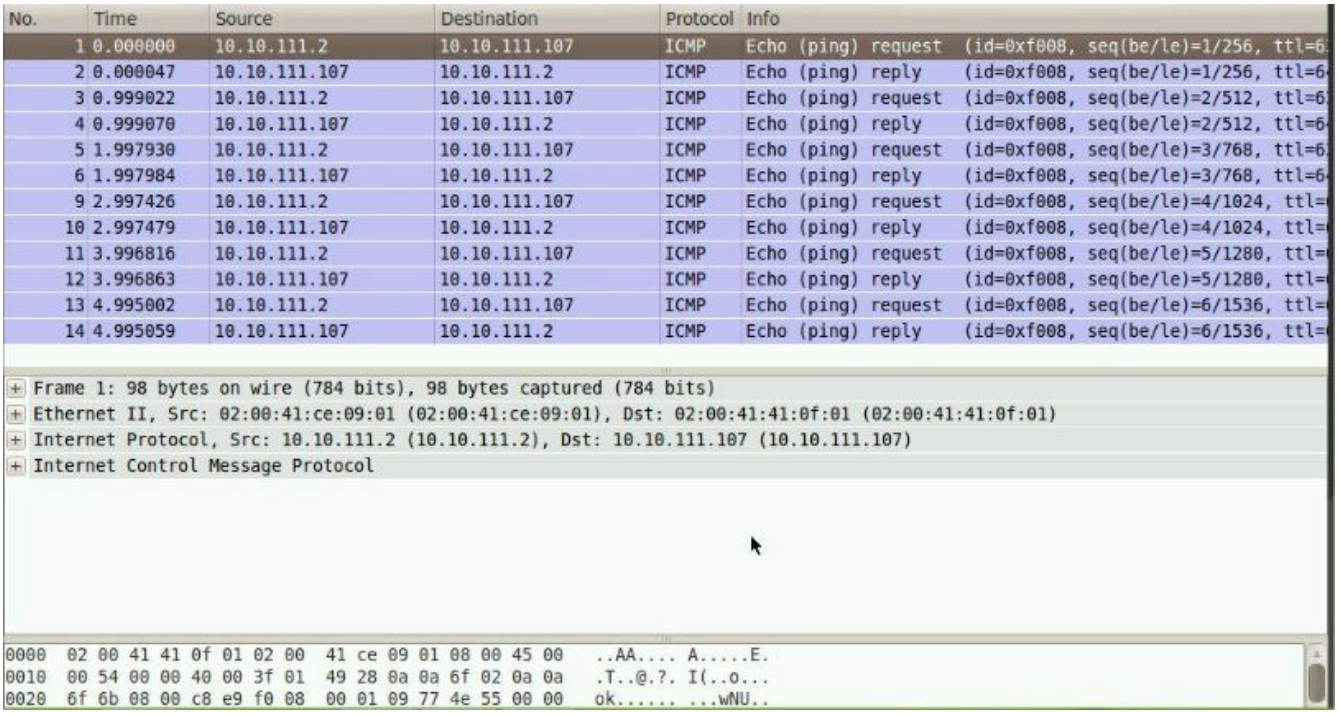
MASQUERADE: used to do source network translation



ping command being run from 10.20.111.2

```
vlab-debian:~# ping 10.10.111.107
PING 10.10.111.107 (10.10.111.107) 56(84) bytes of data.
64 bytes from 10.10.111.107: icmp_seq=1 ttl=63 time=2.94 ms
64 bytes from 10.10.111.107: icmp_seq=2 ttl=63 time=3.17 ms
64 bytes from 10.10.111.107: icmp_seq=3 ttl=63 time=3.12 ms
64 bytes from 10.10.111.107: icmp_seq=4 ttl=63 time=3.94 ms
64 bytes from 10.10.111.107: icmp_seq=5 ttl=63 time=4.07 ms
64 bytes from 10.10.111.107: icmp_seq=6 ttl=63 time=3.06 ms
64 bytes from 10.10.111.107: icmp_seq=7 ttl=63 time=3.68 ms
64 bytes from 10.10.111.107: icmp_seq=8 ttl=63 time=3.56 ms
c64 bytes from 10.10.111.107: icmp_seq=9 ttl=63 time=3.99 ms
64 bytes from 10.10.111.107: icmp_seq=10 ttl=63 time=3.30 ms
^C
--- 10.10.111.107 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 8991ms
rtt min/avg/max/mdev = 2.942/3.486/4.076/0.405 ms
vlab-debian:~# _
```

The output captured the ip addresses 10.10.111.107 and 10.10.111.2 showing conversion of the ip address 10.20.111.2 to 10.10.111.1 and can see source address of the incoming packet is 10.10.111.2, which is the IP of outer interface of the internal router.

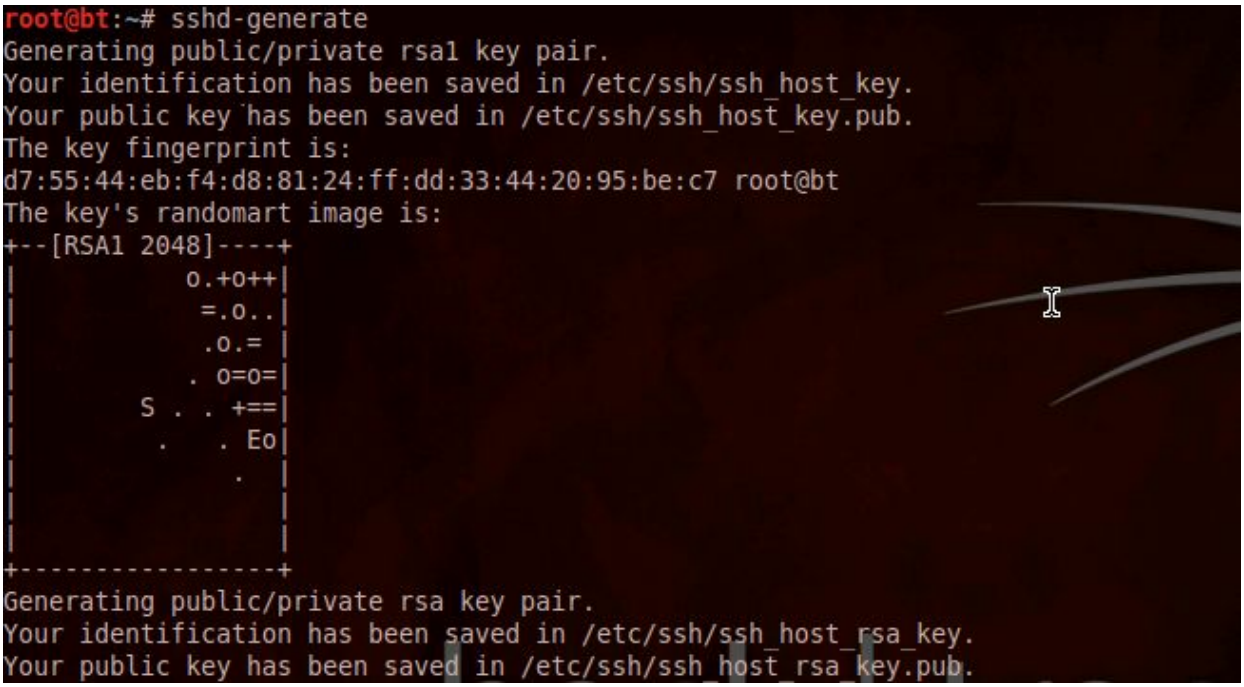


No.	Time	Source	Destination	Protocol	Info
1	0.000000	10.10.111.2	10.10.111.107	ICMP	Echo (ping) request (id=0xf008, seq(be/le)=1/256, ttl=6
2	0.000047	10.10.111.107	10.10.111.2	ICMP	Echo (ping) reply (id=0xf008, seq(be/le)=1/256, ttl=6
3	0.999022	10.10.111.2	10.10.111.107	ICMP	Echo (ping) request (id=0xf008, seq(be/le)=2/512, ttl=6
4	0.999070	10.10.111.107	10.10.111.2	ICMP	Echo (ping) reply (id=0xf008, seq(be/le)=2/512, ttl=6
5	1.997930	10.10.111.2	10.10.111.107	ICMP	Echo (ping) request (id=0xf008, seq(be/le)=3/768, ttl=6
6	1.997984	10.10.111.107	10.10.111.2	ICMP	Echo (ping) reply (id=0xf008, seq(be/le)=3/768, ttl=6
9	2.997426	10.10.111.2	10.10.111.107	ICMP	Echo (ping) request (id=0xf008, seq(be/le)=4/1024, ttl=
10	2.997479	10.10.111.107	10.10.111.2	ICMP	Echo (ping) reply (id=0xf008, seq(be/le)=4/1024, ttl=
11	3.996816	10.10.111.2	10.10.111.107	ICMP	Echo (ping) request (id=0xf008, seq(be/le)=5/1280, ttl=
12	3.996863	10.10.111.107	10.10.111.2	ICMP	Echo (ping) reply (id=0xf008, seq(be/le)=5/1280, ttl=
13	4.995002	10.10.111.2	10.10.111.107	ICMP	Echo (ping) request (id=0xf008, seq(be/le)=6/1536, ttl=
14	4.995059	10.10.111.107	10.10.111.2	ICMP	Echo (ping) reply (id=0xf008, seq(be/le)=6/1536, ttl=

```
⊞ Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)
⊞ Ethernet II, Src: 02:00:41:ce:09:01 (02:00:41:ce:09:01), Dst: 02:00:41:41:0f:01 (02:00:41:41:0f:01)
⊞ Internet Protocol, Src: 10.10.111.2 (10.10.111.2), Dst: 10.10.111.107 (10.10.111.107)
⊞ Internet Control Message Protocol
```

```
0000  02 00 41 41 0f 01 02 00  41 ce 09 01 08 00 45 00   ..AA.... A.....E.
0010  00 54 00 00 40 00 3f 01  49 28 0a 0a 6f 02 0a 0a   .T..@.?. I(..o...
0020  6f 6b 08 00 c8 e9 f0 08  00 01 09 77 4e 55 00 00   ok...... ...wNU..
```

we open an ssh command from the internal machine to the external backtrack machine 10.10.111.10



```
root@bt:~# sshd-generate
Generating public/private rsa1 key pair.
Your identification has been saved in /etc/ssh/ssh_host_key.
Your public key has been saved in /etc/ssh/ssh_host_key.pub.
The key fingerprint is:
d7:55:44:eb:f4:d8:81:24:ff:dd:33:44:20:95:be:c7 root@bt
The key's randomart image is:
+--[RSA1 2048]----+
|           o.+o++|
|            =.o..|
|             .o.= |
|            . o=o=|
|        S . . +==|
|         .   . Eo|
|               . |
|                 |
|                 |
+-----------------+
Generating public/private rsa key pair.
Your identification has been saved in /etc/ssh/ssh_host_rsa_key.
Your public key has been saved in /etc/ssh/ssh_host_rsa_key.pub.
```

```
Generating public/private rsa key pair.
Your identification has been saved in /etc/ssh/ssh_host_rsa_key.
Your public key has been saved in /etc/ssh/ssh_host_rsa_key.pub.
The key fingerprint is:
f2:20:a8:03:97:82:ea:d7:25:c3:e8:79:ee:0f:2a:c2 root@bt
The key's randomart image is:
+--[ RSA 2048]----+
|                 |
|                 |
|                 |
|.  o             |
|+ + + o S        |
|o+ . = =         |
|= . o.+ .        |
|oE +.o.          |
|..o.+o..         |
+-----------------+
Generating public/private dsa key pair.
Your identification has been saved in /etc/ssh/ssh_host_dsa_key.
Your public key has been saved in /etc/ssh/ssh_host_dsa_key.pub.
The key fingerprint is:
b8:c4:7a:7a:01:b9:e9:7c:7d:e6:c4:c8:37:50:e0:11 root@bt
The key's randomart image is:
+--[ DSA 1024]----+
|        E.       |
|       . o       |
|      .  . .     |
|    o. . .       |
|     ++ S        |
|    oo.o +       |
|   o. ooo =      |
|    ooo .oo.     |
|    .o   +.      |
+-----------------+
root@bt:~#
```

```
root@bt:~# /etc/init.d/ssh start
 * Starting OpenBSD Secure Shell server sshd
root@bt:~#
```

Perform ssh from internal linux machine using the following command:

```
###############################################################################
[*] Welcome to the BackTrack 5 Distribution, Codename "Revolution"

[*] Official BackTrack Home Page: http://www.backtrack-linux.org

[*] Official BackTrack Training : http://www.offensive-security.com
###############################################################################

[*] To start a graphical interface, type "startx".
[*] The default root password is "toor".

Last login: Mon Nov 16 20:26:50 2015
root@bt:~# _
```

Part C

1) Iptables can be used as a packet filtering tool. A privileged user can define certain rules applicable to the incoming and outgoing packets to and from the system. In addition, it can also perform network address and port translation and packet mangling.
It can work either as a host based firewall or a network based firewall depending on whether it is looking at the packets that are going to and from the host or being routed through the network.
The rules in iptables are grouped together into chains. A chain is a set of rules that determines the action to be taken on a packet.

Each rule in a chain consists of the specification of the packets those will match with it. As a packet traverse a chain, it is checked against each rule sequentially. If a rule matches a packet then corresponding action is taken on the packet. Once a matching rule is found for a packet, it is further not checked with the other rules coming after the current matched rule. If none of the existing rule matches the packet, then it is processed according to the default rule set.

A table in turn is formed from the chains. It has different types of table - filter, NAT, mangle, raw. The different types of chains in iptables are - INPUT, OUTPUT and FORWARD. If a packet is entering a host with iptables firewall, the rule goes to INPUT chain. If a packet is leaving a host with iptables firewall, the rule goes to OUTPUT chain. Also, if a packet is going through the firewall, the rule goes to forward chain.

2) INPUT chain - This applies for all the packets which are destined to the host with iptables.
OUTPUT chain - This applies for all the packets that are leaving the host with iptables.
FORWARD chain - This applies for all the packets going through the firewall. It is for the packets that are neither emitted by the host nor directed to the host.

3) Deny - The packet is discarded, dropped to the floor, assigned to oblivion. No reply packet of any kind is sent. There is no response to the SYN packet. If a rule is set which matches a particular source address with DENY, then the system is seen as down to the source address. But, if we set DENY for some port ranges, then it allows the source address to know that we are running a firewall.

Reject - An ICMP port unreachable packet is sent to the source address. This tells the remote host that the system is up and running with a firewall.

Accept - When a service is not listening on a port number, the response to the SYN packet (S) from a source address is a connection reset (R). When a service is listening on a port number, a successful TCP connection is established.

4) Some alternative network based firewalls

i) Firewall-1: It is one of the most powerful commercial firewall.

Pros: Good documentation, Professional support, Training services available, includes high-end features such as load balancing and redundancy, excellent reputation

Cons: Requires intensive training for the administrator, very expensive

ii) PfSense: PfSense is BSD based firewall. It is slightly more secure than a firewall using Iptables would be, because it makes a connection harder to spoof by tracking TCP sequence numbers.

iii) Smoothwall Advanced: It is used as an open source firewall. It is designed to be used as a corporate office firewall, with the options to create authentication-based access to different parts of the network and its web proxy and email filtering systems.

Cons: Not very suitable for use as small office firewall or in home.