Todd Wenker

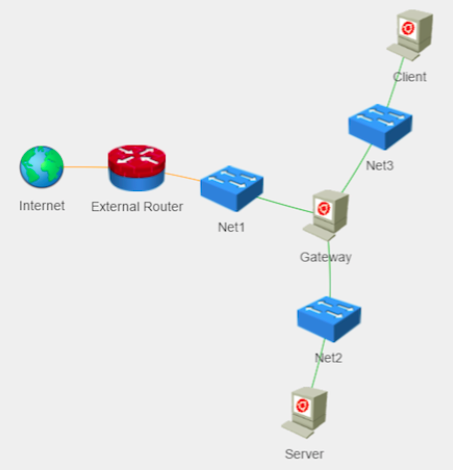
ASU ID: 1206233882

Lab Assignment 5

Firewall Lab

**Summary:** The goal of the lab is to set up the Gateway machine to act as a Firewall for the Server and Client machines. A firewall is a form of network security that tests incoming and outgoing packets based on a set of predefined rules. By using the Iptables tool, rules can set up on the Gateway machine that will block packets that pass through the machine.

**Network Setup:**



The network consists of three hosts: Gateway, Server, and Host. Gateway is set up as a DNS server for the subnet containing Gateway and Server.

**Software Packages Used:**

* Iptables
* SSH

**Task 1 --- Use IPTABLES NAT to make the Client VM and the Server VM access the Internet**

Before the rules can be established, a file titled rc.firewall is created. This file will be a script that contains all of the Iptables commands that will configure the firewall. The file is created using the command *vim rc.firewall* and the command *chmod +x rc.firewall* will change the mode of the script so that it becomes executable. Then the following lines are added:

#!/bin/bash

#####################################################

# Initial Setup

#####################################################

PUBLIC\_IFACE="eth0"

PUBLIC\_IP="192.168.1.5"

PUBLIC\_GW="192.168.1.1"

PUBLIC\_NETWORK="192.168.1.0/24"

LAN\_SERVER\_IFACE="eth1"

LAN\_SERVER\_IP="192.168.2.4"

LAN\_SERVER\_GW="192.168.2.5"

LAN\_SERVER\_NETWORK="192.168.2.0/24"

LAN\_CLIENT\_IFACE="eth2"

LAN\_CLIENT\_IP="192.168.3.4"

LAN\_CLIENT\_GW="192.168.3.5"

LAN\_CLIENT\_NETWORK="192.168.3.0/24"

echo "1" > /proc/sys/net/ipv4/ip\_forward

IPTABLES="/sbin/iptables"

$IPTABLES -F

$IPTABLES -F -t nat

The *#!/bin/bash* denotes that this is a bash script. The following variables are then assigned values that will be used throughout the following Iptables commands, like the IP addresses of each machine. The *echo “1” /proc/sys/net/ipv4/ip\_forward* command sets the ip\_forward flag to one, turning IP forwarding on. The *$IPTABLES –F* and *$IPTABLES \_F –t nat* will flush the rules in the firewall and the NAT table.

Once the general setup is done, rules can be added to the file. To allow the Server and Client machines access to the Internet the following commands are added to rc.firewall:

#client

$IPTABLES -A FORWARD -i eth2 -o eth0 -j ACCEPT

$IPTABLES -A FORWARD -i eth0 -o eth2 -m state –state ESTABLISHED,RELATED -j ACCEPT

#server

$IPTABLES -A FORWARD -i eth1 -o eth0 -j ACCEPT

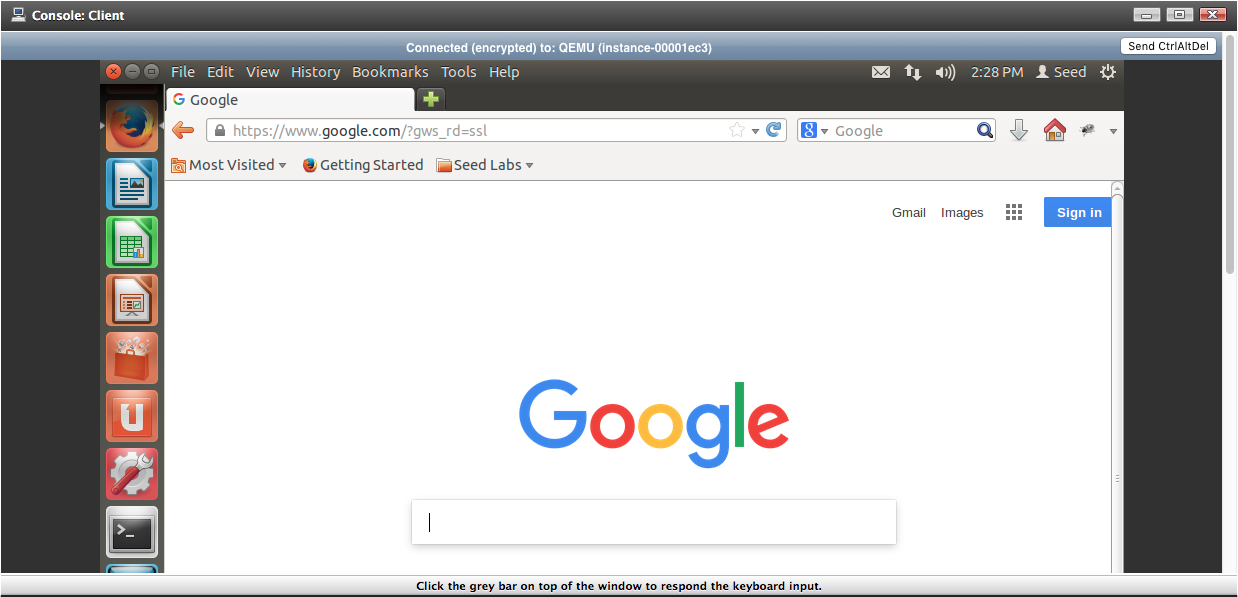
$IPTABLES -A FORWARD -i eth0 -o eth1 -m state –state ESTABLISHED,RELATED -j ACCEPT

#gateway

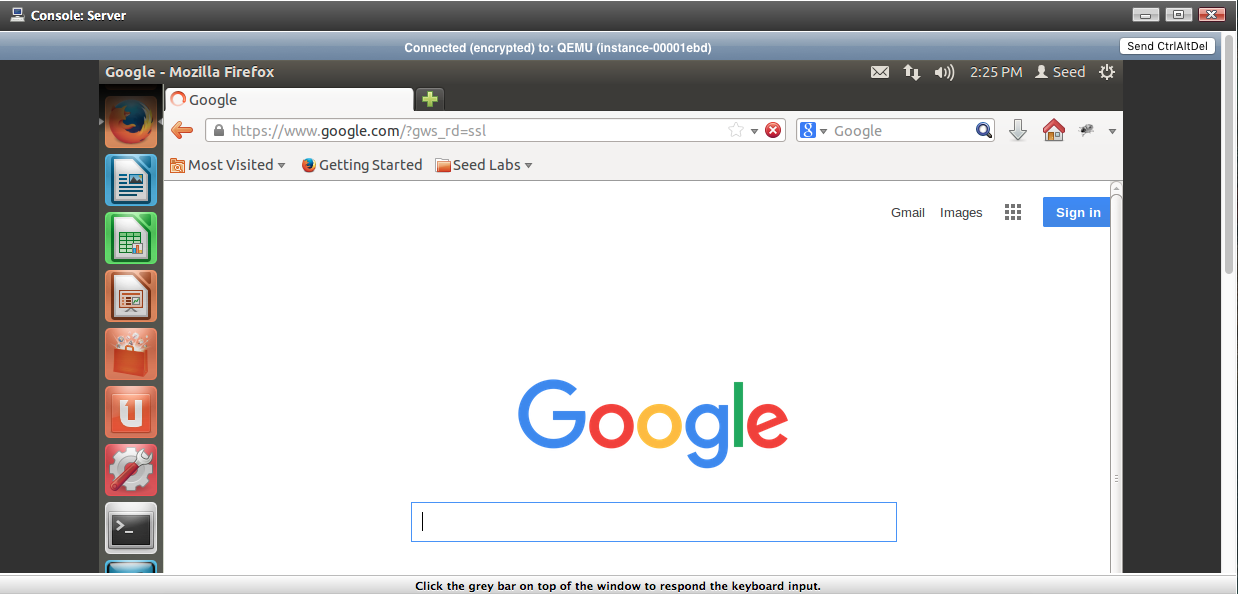
$IPTABLES -A OUTPUT -o eth0 -j ACCEPT

$IPTABLES -A INPUT -i eth0 -j ACCEPT

The first line in the Client and the Server commands sets a rule that allows incoming packets from the interfaces ‘eth2’ and ‘eth1’ to be forwarded through the ‘eth0’ interface, which is the interface connected to the internet. The second command allows the forwarding of packets from the ‘eth0’ interface to the ‘eth2’ and ‘eth1’ interfaces, which are connected to Client and Server respectively, as long as the state of the packets are established or related. This allows the Gateway machine to forward packets to Internet from Client and Server and to Client and Server from the Internet. The commands under the Gateway section accept packets that enter or leave the Gateway machine from the ‘eth0’ interface. After rc.firewall is saved, the command *sudo ./rc.firewall* is used to run the script.



Above is the Client machine accessing the Internet.



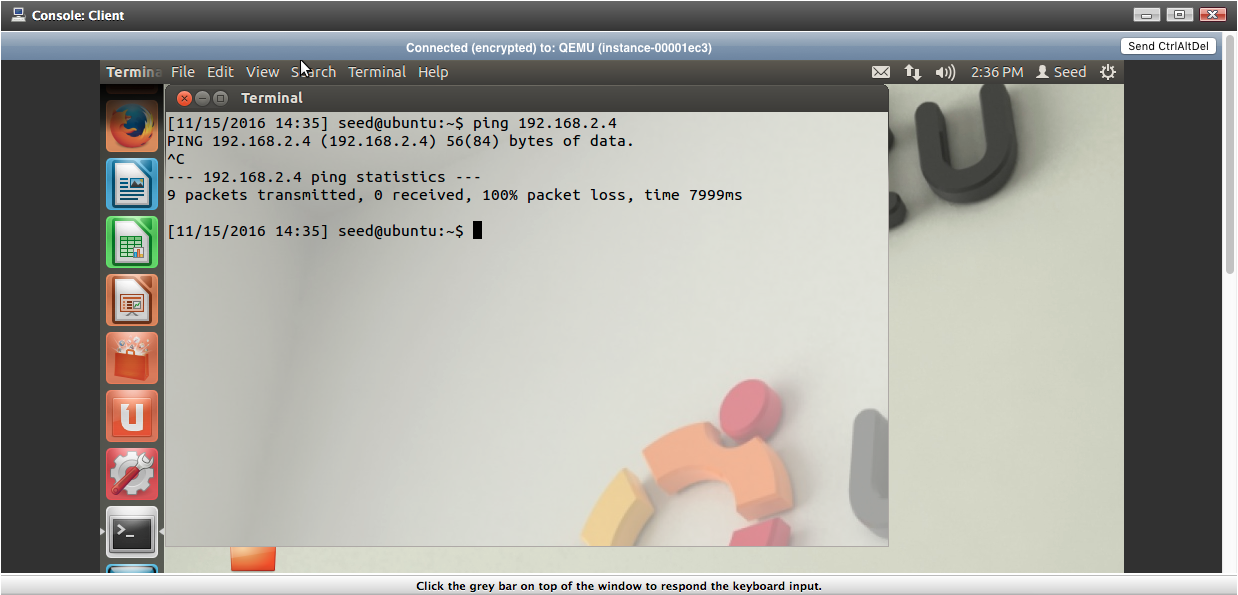
Above is the Server machine accessing the Internet.

**Task 2 --- Use NAT to hide Server VM from Client VM**

Next, to hide the IP address of Client and Server by adding the command:

$IPTABLES --table nat -A POSTROUTING --out-interface eth2 -j MASQUERADE

This command will affect the postrouting phase of the NAT so that outgoing packets bare the IP address of the Gateway machine. The rc.firewall script is run again so these changes take effect.

****

Above is the ping result from Client, attempting to ping Server. The Server does not receive the message.

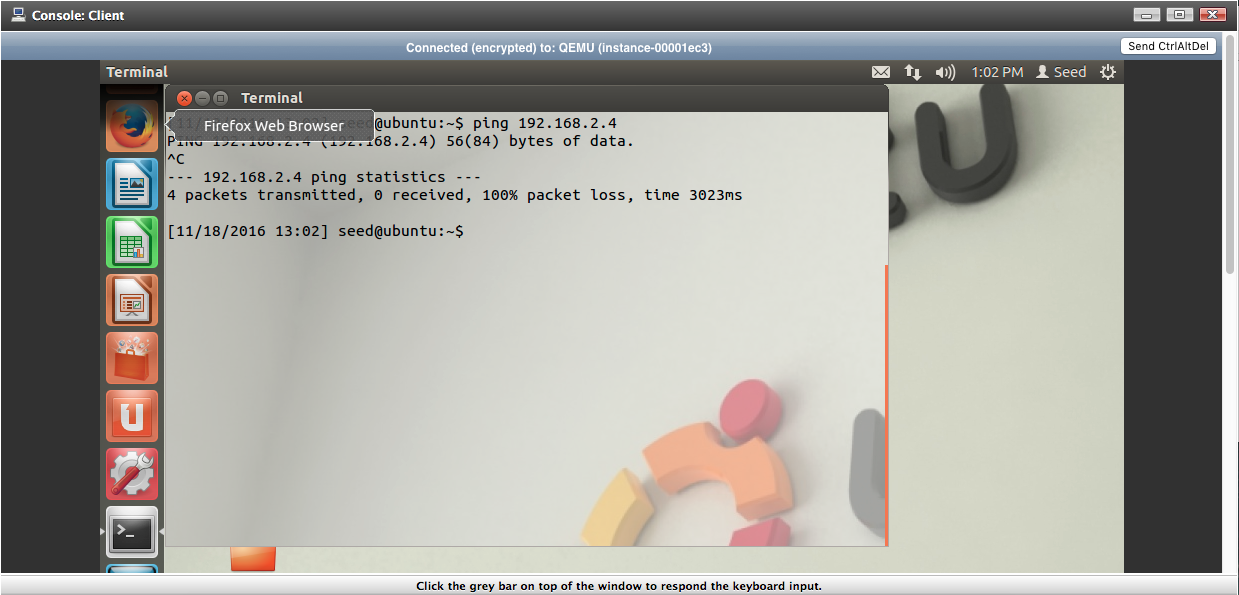
**Task 3 --- Client can ping Gateway but cannot ping Server**

Next, the following command is added to rc.firewall to allowing the Client to ping the Gateway:

$IPTABLES -A INPUT -s $LAN\_CLIENT\_IP -i eth2 -p icmp --icmp-type echo-request -j ACCEPT

$IPTABLES -A OUTPUT -o eth2 -d $LAN\_CLIENT\_IP -p icmp --icmp-type echo-reply -j ACCEPT

The first command accepts input coming from the Client’s IP address along the ‘eth2’ interface, as long as the protocol is ICMP and the type of message is an echo request. The second command accepts output from the Gateway machine using the ‘eth2’ interface to the Client’s IP address as long as the message uses ICMP protocol and is of type echo reply.

****

Above shows that Client can still not ping the Server machine, although it can ping Gateway.

**Task 4 --- Gateway can ping Server. Server can ping Gateway**

To allow the Server and Gateway machines to ping each other, the following commands are added to rc.firewall:

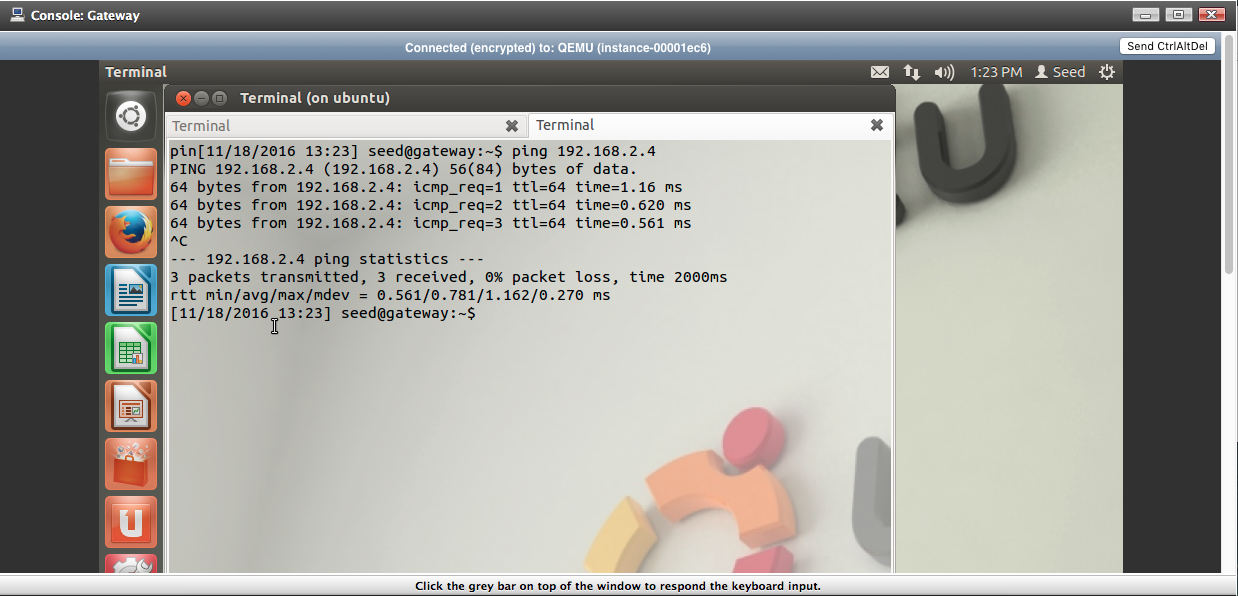
$IPTABLES -A INPUT -s $LAN\_SERVER\_IP -i eth1 -d $LAN\_SERVER\_GW -p icmp --icmp-type echo-request -j ACCEPT

$IPTABLES -A INPUT -s $LAN\_SERVER\_IP -i eth1 -d $SLAN\_ERVER\_GW -p icmp --icmp-type echo-reply -j ACCEPT

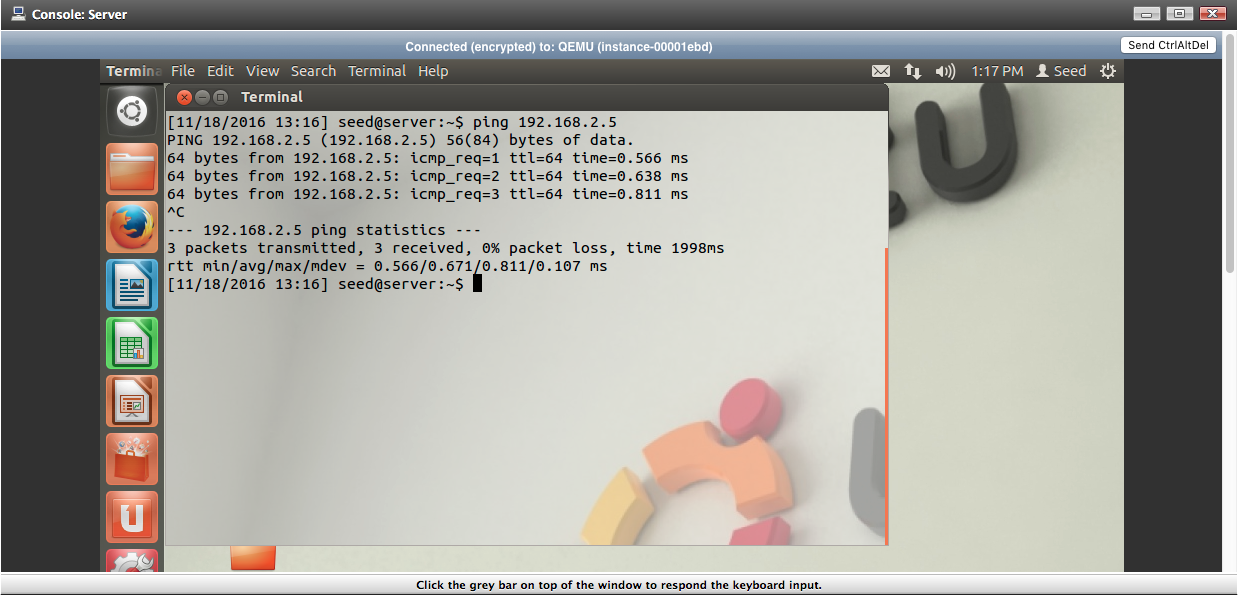
$IPTABLES -A OUTPUT -s $LAN\_SERVER\_GW -o eth1 -d $LAN\_SERVER\_IP -p icmp --icmp-type echo-request -j ACCEPT

$IPTABLES -A OUTPUT -s $LAN\_SERVER\_GW -o eth1 -d $LAN\_SERVER\_IP -p icmp --icmp-type echo-reply -j ACCEPT

The two sets of commands are similar to the two used above, with the exception of changed IP address. The first set allows the Server machine to send ICMP echo-request messages to Gateway and for Gateway to respond with echo-reply. The second set of commands does the inverse, allowing Gateway to ping Server.

****

Gateway using ping on Server’s IP address of 192.168.2.4 and receiving a response.

****

Server using ping on Gateway’s IP address (on that subnet) of 192.168.2.5 and receiving a response.

**Task 5 --- The Server can initiate an SSH connection to the Gateway**

The following commands allow Server to initiate an SSH connection to Gateway:

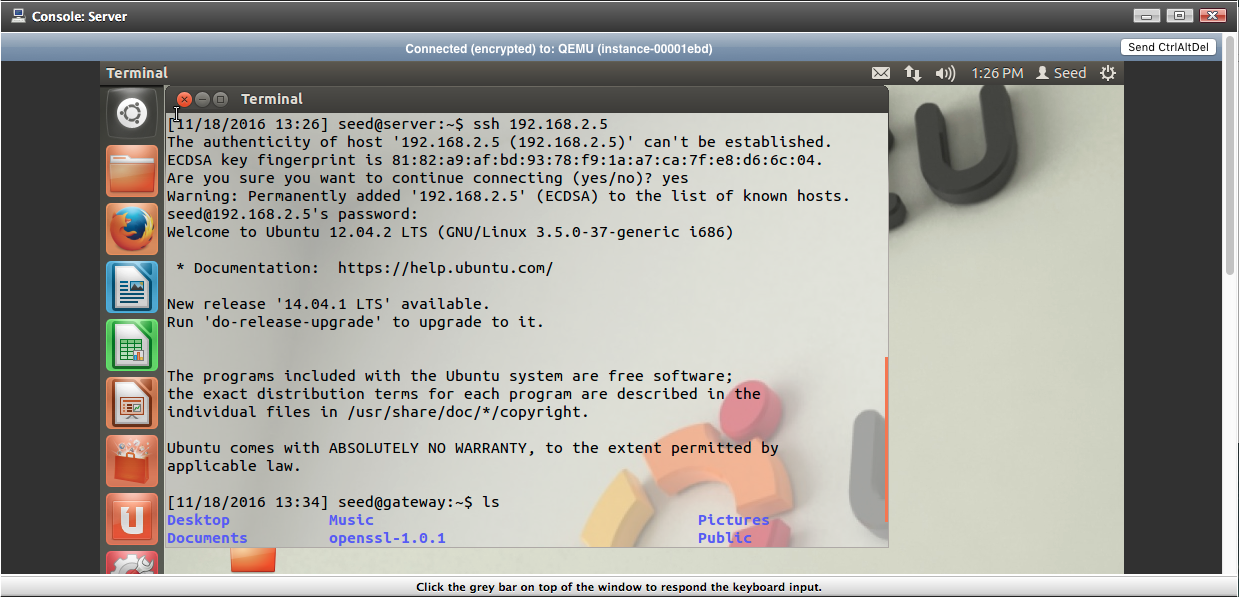
$IPTABLES -A INPUT -s $LAN\_SERVER\_IP -i eth1 -p tcp --dport 22 -j ACCEPT

$IPTABLES -A INPUT -s $LAN\_SERVER\_IP -i eth1 -p tcp -m state --state ESTABLISHED,RELATED -j ACCEPT

$IPTABLES -A OUTPUT -o eth1 -d $LAN\_SERVER\_IP -p tcp --sport 22 -j ACCEPT

$IPTABLES -A OUTPUT -o eth1 -d $LAN\_SERVER\_IP -p tcp -m state --state ESTABLISHED,RELATED -j ACCEPT

The first command tells Gateway to accept messages having Server’s IP, entering on the ‘eth1’ interface, have protocol as TCP, and are connecting to Gateway’s port number 22, which is the port assigned to SSH. The second command has Gateway accept input with the same IP address, interface, and protocol as above but with the state of the connection being either ESTABLISHED or RELATED. The third command allows Gateway to output packets that has TCP as protocol to Server’s IP address along the ‘eth1’ interface and using port number 22 as a source port. The last command allows Gateway to send packets to Server as long as the state are ESTABLISH or RELATED.

****

Above shows the Server establishing an SSH connection to Gateway.

**Task 6 --- The Client can initiate an SSH connection to the Server**

The following commands to allow Client to established an SSH connection to Server are as follows:

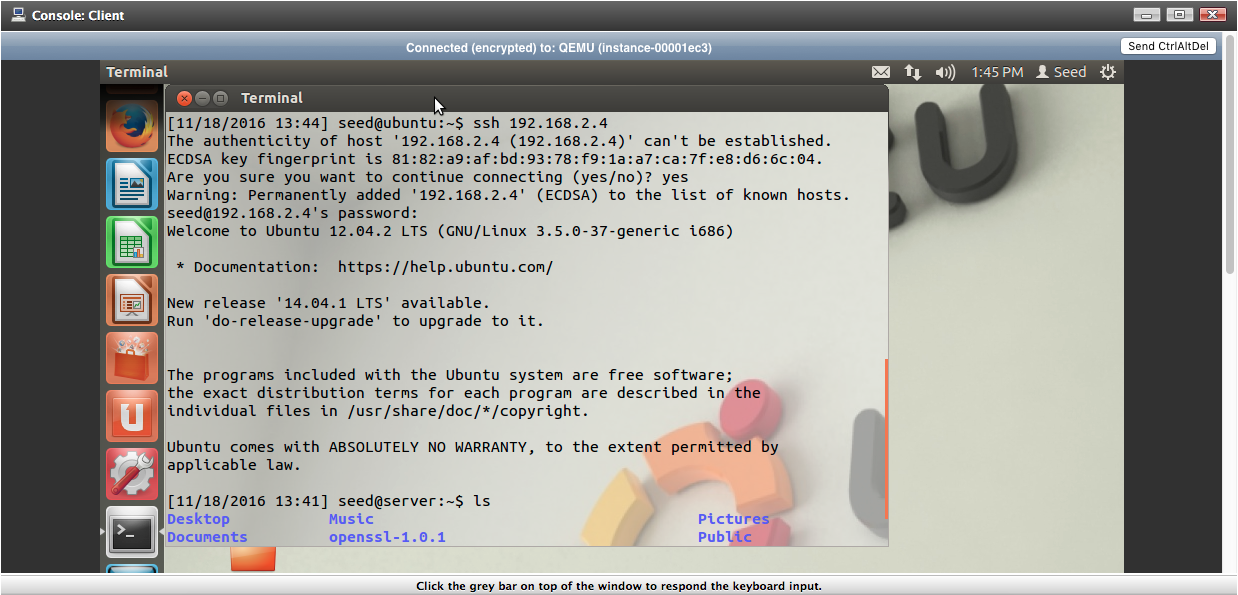
$IPTABLES -A FORWARD -s $LAN\_CLIENT\_IP -i eth2 -d $LAN\_SERVER\_IP -o eth1 -p tcp --dport 22 -j ACCEPT

$IPTABLES -A FORWARD -s $LAN\_CLIENT\_IP -i eth2 -d $LAN\_SERVER\_IP -o eth1 -p tcp -m state --state ESTABLISHED,RELATED -j ACCEPT

$IPTABLES -A FORWARD -d $LAN\_CLIENT\_IP -o eth2 -s $LAN\_SERVER\_IP -i eht1 -p tcp --sport 22 -j ACCEPT

$IPTABLES -A FORWARD -d $LAN\_CLIENT\_IP -o eth2 -s $LAN\_SERVER\_IP -i eth1 -p tcp -m state --state ESTABLISHED,RELATED -j ACCEPT

The commands above are similar to the previous section, with a few important differences. Because the Gateway machine is not the destination or the source of the traffic, the –A flag, which defines the rule chain that will be used, is set to FORWARD. The first two commands accept traffic from Client along the ‘eth2’ interface that is forwarded to Server along the ‘eth1’ interface. The protocol must be TCP, the state needs to be ESTABLISHED or RELATED, and the destination port must be number 22, the SSH designated port. The second pair of commands accepts traffic moving from Server along the ‘eth1’ interface that is forwarded to Client along the ‘eth2’ interface. The protocol must be TCP, the state must be ESTABLISHED or RELATED, and the source port must be number 22.

****

Above shows Client successfully establishing an SSH connection to server by using the command *ssh 192.168.2.4* with the IP address correlating to Server.

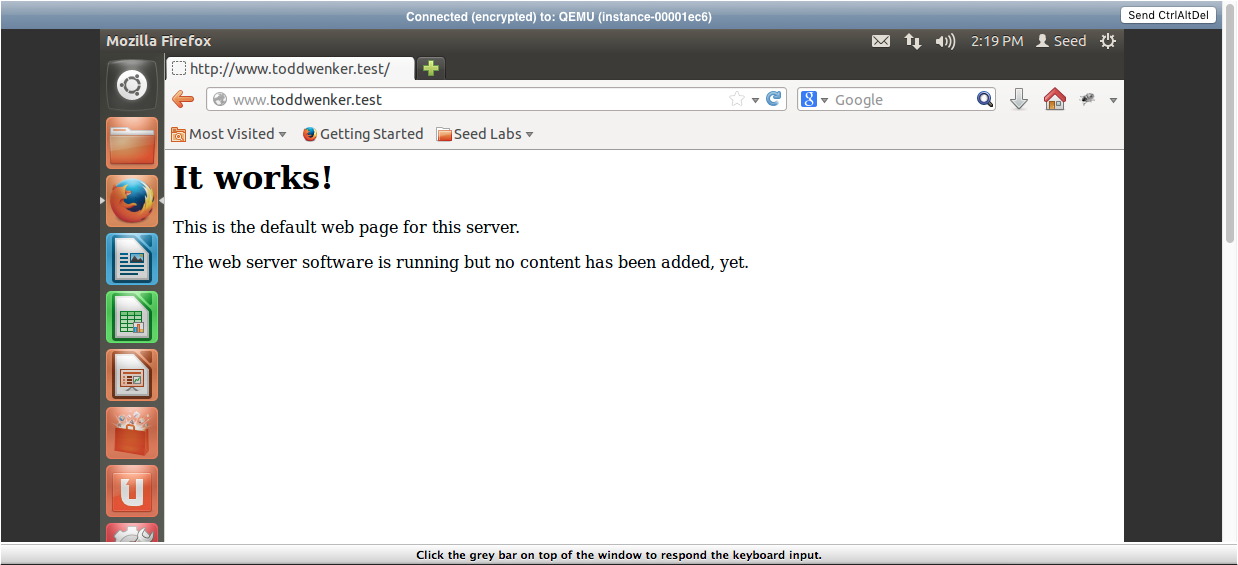
**Task 7 --- The Client can access the website hosted on the Server using port forwarding**

The following commands will allow Client to access the website hosted on Server:

$IPTABLES -A FORWARD -s $LAN\_CLIENT\_IP -i eth2 -d $LAN\_SERVER\_IP -o eth1 -p tcp --dport 80 -j ACCEPT

$IPTABLES -A FORWARD -d $LAN\_CLIENT\_IP -o eth2 -s $SERVER\_IP -i eth1 -p tcp --sport 80 -j ACCEPT

The first command accepts traffic from Client from the ‘eth2’ interface directed towards Server along the ‘eth1’ interface, provided that the protocol being used is TCP and the destination port is number 80. Port 80 is the standard port for HTTP so accessing the Server’s webpage is done through this port. The second does the reverse, allowing traffic from Server to Client provided that the source port on Server is port number 80.

****

**Task 8 --- All other traffic should be dropped**

The following commands will drop all other packets:

$IPTABLES -P INPUT DROP

$IPTABLES -P OUTPUT DROP

$IPTABLES -P FORWARD DROP

The commands set the default policy for the rules INPUT, OUTPUT, and FORWARD. If traffic does not get accepted according to the other established rules, by default, the firewall will drop those packets.

**Conclusion:** By using the Iptables tool, a firewall can be setup by establishing rules that gauge valid and invalid packets. These rules can also be used to establish NAT tables that allows a network to connect to the internet through a single IP address.