Ddos: Distributed Denial Of Service

TEAM NO: 17
Aakash Arora : 01fb14ecs003
Divya Ranjan : 01fb14ecs069
Harish Reddy : 01fb14ecs079

What exactly is Ddos ?

A Distributed Denial of Service (DDoS) is a cyber-attack where the perpetrator seeks to make a machine or network resource unavailable to its intended users by temporarily or indefinitely disrupting services of a host connected to the Internet.A DDoS attack is launched from numerous compromised devices, often distributed globally in what is referred to as a botnet..

Types of Ddos

- UDP Flood
- SYN Flood
- Ping of Death
- Slowloris
- NTP Amplification

Initial Attack Phase

– – –

- We created a single python DOS script for UDP Flooding with network bandwidth of about 50Kbps.
- We then simulated small scale Ddos attack by running multiple instance of it.

Task Manager

File Options View

Processes Performance App history Startup Users Details Services

CPU
1% 1.59 GHz

Memory
3.8/11.9 GB (32%)

Disk 0 (C: D: E:)
19%

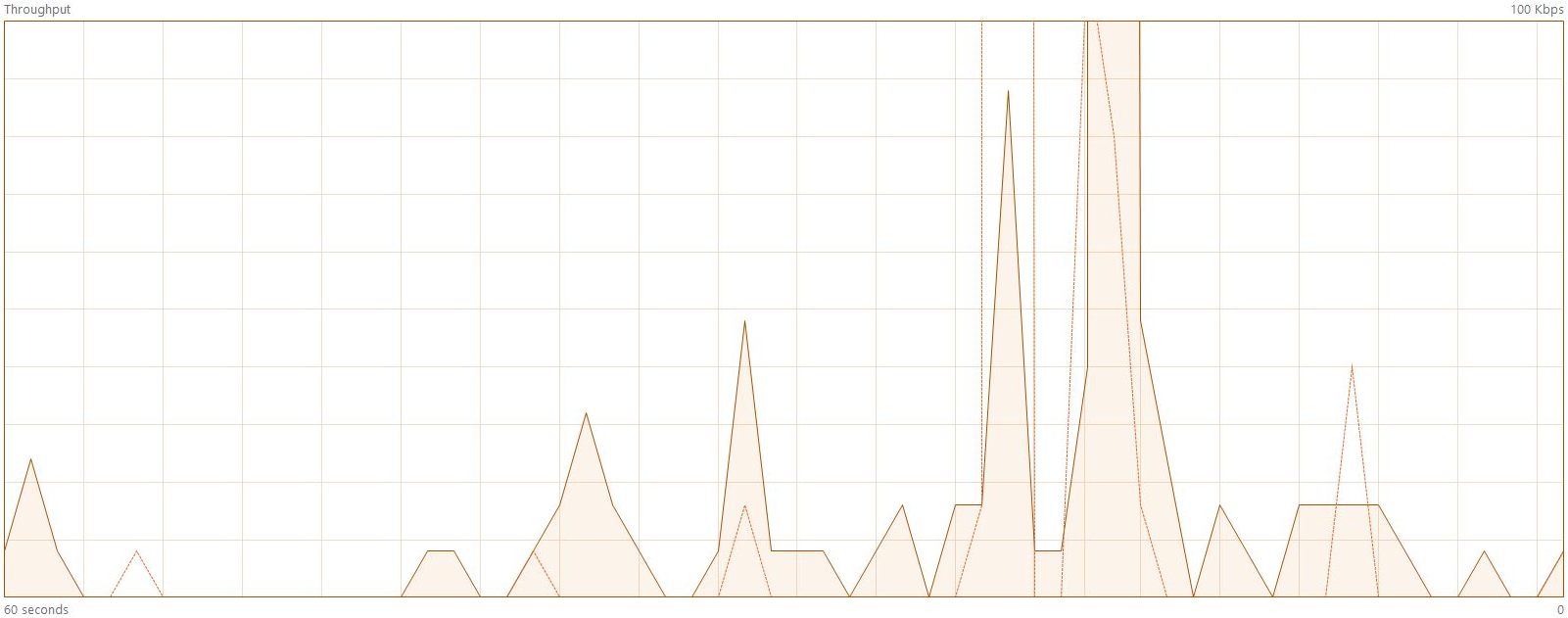
Ethernet
Not connected

Wi-Fi
S: 8.0 R: 8.0 Kbps

Bluetooth
Not connected

Wi-Fi

Intel(R) Dual Band Wireless-AC 3160



Send
8.0 Kbps

Receive
8.0 Kbps

Adapter name:	Wi-Fi
SSID:	maalware
Connection type:	802.11n
IPv4 address:	192.168.0.104
IPv6 address:	fe80::cdcd:d891:bc45:f405%9
Signal strength:	

Fewer details | Open Resource Monitor

9:52 AM
4/24/2017

Transition In Attack Phase (50 kbps to 1 Gbps)

– – –

- We ran a multiple instances of a simple scraping script using beautiful - soup in Amazon Web Services
- We live tested it on webindia123.com
- Result : Site crashed in around 4 minutes

Is Yellowpages.webindia123.com Down?

Yellowpages.webindia123.com seems to be down.

We have tried accessing the Yellowpages.webindia123.com website using our servers and we were unable to connect to the website. If Yellowpages.webindia123.com is also down for you then there is likely a problem with their servers. If you think this is an error you may proceed to the troubleshooting section o try to diagnose and resolve the problem.

This website was last checked: **2 secs ago**

Hit the check button to update this page.

Check Website

Information

Website Status	Reponse Time	Response Code
Currently Down	Error	Error

Recently Checked

- yellowpages.webindia123.com
- openweathermap.org
- apetitar.com.br
- pizzacesar.apetitar.com.br
- 50.63.202.13
- punters.com.au

Attack Phase exploration

- We searched for various Ddos simulation tools and came across Bonesi.
- BoNeSi is a network traffic generator for different protocol types. The attributes of the created packets and connections can be controlled by several parameters like send rate or payload size or they are determined by chance. It spoofs the source ip addresses even when generating tcp traffic.

Task Manager

File Options View

Processes Performance App history Startup Users Details Services

CPU
5% 3.46 GHz

Memory
6.0/11.9 GB (50%)

Disk 0 (C: D: E:)
0%

Ethernet
Not connected

Wi-Fi
S: 0 R: 6.9 Mbps

Bluetooth
Not connected

Wi-Fi

Intel(R) Dual Band Wireless-AC 3160

Throughput

11 Mbps

7.7 Mbps

60 seconds

0

Send
16.0 Kbps

Receive
6.9 Mbps

Adapter name:	Wi-Fi
SSID:	maalware
Connection type:	802.11n
IPv4 address:	192.168.0.106
IPv6 address:	fe80::cdcd:d891:bc45:f405%9
Signal strength:	

Fewer details | Open Resource Monitor

8:26 PM
4/18/2017

Task Manager

File Options View

Processes Performance App history Startup Users Details Services

CPU
6% 3.46 GHz

Memory
6.0/11.9 GB (50%)

Disk 0 (C: D: E:)
0%

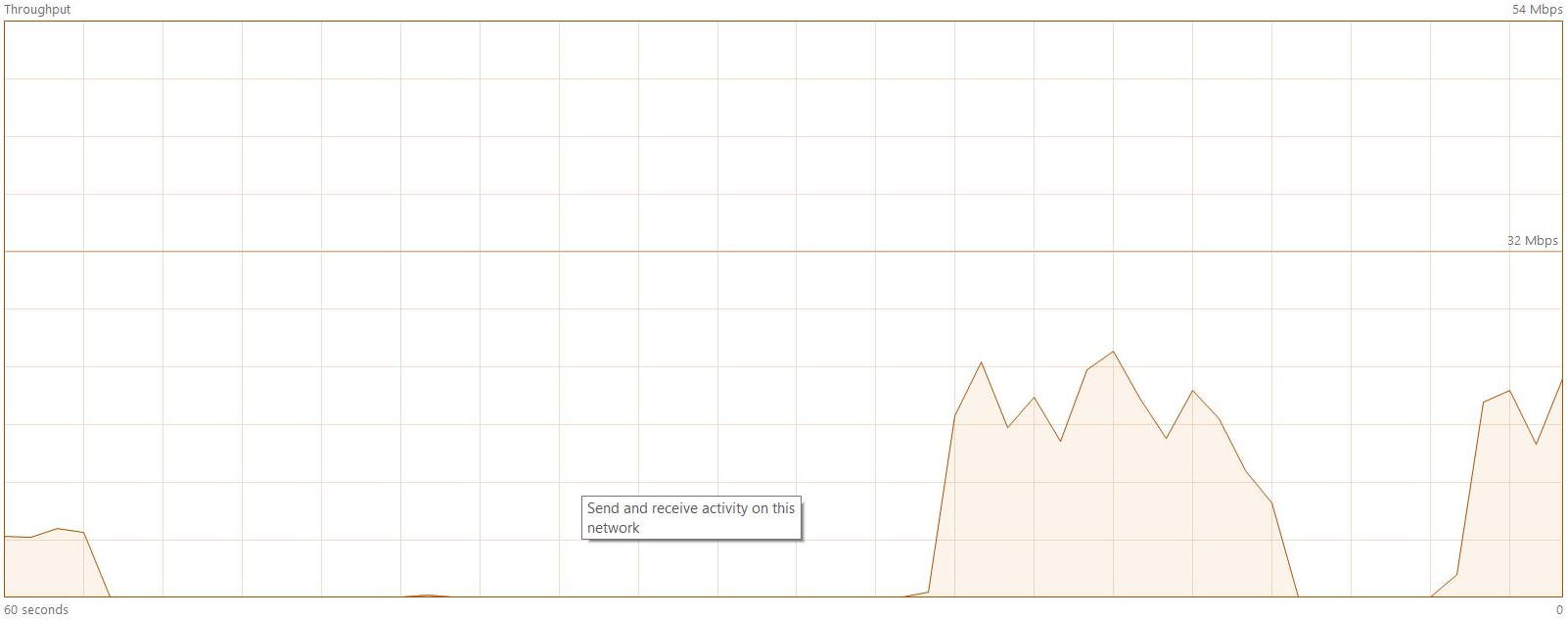
Ethernet
Not connected

Wi-Fi
S: 0 R: 20.5 Mbps

Bluetooth
Not connected

Wi-Fi

Intel(R) Dual Band Wireless-AC 3160



Send
0 Kbps

Receive
20.5 Mbps

Adapter name:	Wi-Fi
SSID:	maalware
Connection type:	802.11n
IPv4 address:	192.168.0.106
IPv6 address:	fe80::cdcd:d891:bc45:f405%9
Signal strength:	

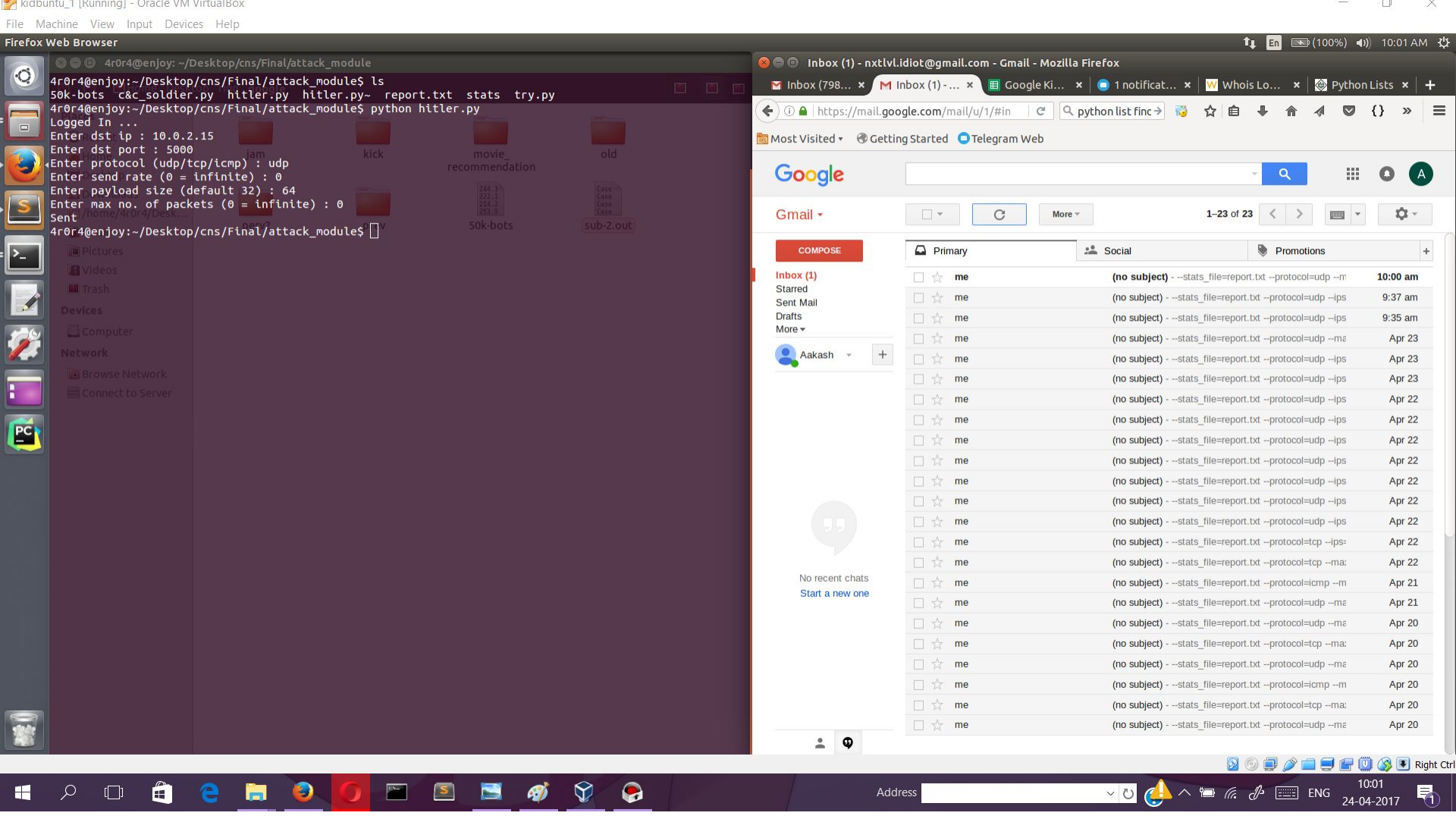
Fewer details | Open Resource Monitor

8:27 PM
4/18/2017

What features/changes we added to Bonesi

– – –

- We created a command & control script for Bonesi - using gmail
- A C&C infrastructure consists of servers and other technical infrastructure used to control malware in general, and, in particular, botnets.
- Inspired by GCAT backdoor - gmail traffic usually unblocked in company infrastructures.
- We also scaled down the no. of bots generated and IPs spoofed for increasing efficiency at defense phase at our local machines.


kidbuntu_1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Firefox Web Browser
4r0r4@enjoy: ~/Desktop/cns/Final/attack_module
4r0r4@enjoy:~/Desktop/cns/Final/attack_module$ ls
50k-bots c&c_soldier.py hitler.py hitler.py~ report.txt stats try.py
4r0r4@enjoy:~/Desktop/cns/Final/attack_module$ python hitler.py
Logged In ...
Enter dst ip : 10.0.2.15
Enter dst port : 5000
Enter protocol (udp/tcp/icmp) : udp
Enter send rate (0 = infinite) : 0
Enter payload size (default 32) : 64
Enter max no. of packets (0 = infinite) : 0
Sent
4r0r4@enjoy:~/Desktop/cns/Final/attack_module$
jam
kick
movie_recommendation
old
50k-bots
sub-2.out
Pictures
Videos
Trash
Devices
Computer
Network
Browse Network
Connect to Server
Inbox (1) - nxtlvl.idiot@gmail.com - Gmail - Mozilla Firefox
Inbox (798...
Inbox (1) - ...
Google Ki...
1 notificat...
Whois Lo...
Python Lists
https://mail.google.com/mail/u/1/#in
python list finc
Most Visited
Getting Started
Telegram Web
Google
Gmail
More
1–23 of 23
COMPOSE
Inbox (1)
Starred
Sent Mail
Drafts
More
Aakash
Primary
Social
Promotions
me (no subject) - --stats_file=report.txt --protocol=udp --m 10:00 am
me (no subject) - --stats_file=report.txt --protocol=udp --ips 9:37 am
me (no subject) - --stats_file=report.txt --protocol=udp --ips 9:35 am
me (no subject) - --stats_file=report.txt --protocol=udp --ma Apr 23
me (no subject) - --stats_file=report.txt --protocol=udp --ips Apr 23
me (no subject) - --stats_file=report.txt --protocol=udp --ips Apr 23
me (no subject) - --stats_file=report.txt --protocol=udp --ips Apr 22
me (no subject) - --stats_file=report.txt --protocol=udp --ips Apr 22
me (no subject) - --stats_file=report.txt --protocol=udp --ips Apr 22
me (no subject) - --stats_file=report.txt --protocol=udp --ips Apr 22
me (no subject) - --stats_file=report.txt --protocol=udp --ips Apr 22
me (no subject) - --stats_file=report.txt --protocol=udp --ips Apr 22
me (no subject) - --stats_file=report.txt --protocol=udp --ips Apr 22
me (no subject) - --stats_file=report.txt --protocol=tcp --ips- Apr 22
me (no subject) - --stats_file=report.txt --protocol=tcp --ma Apr 22
me (no subject) - --stats_file=report.txt --protocol=icmp --m Apr 21
me (no subject) - --stats_file=report.txt --protocol=udp --ma Apr 21
me (no subject) - --stats_file=report.txt --protocol=udp --ma Apr 20
me (no subject) - --stats_file=report.txt --protocol=tcp --ma Apr 20
me (no subject) - --stats_file=report.txt --protocol=udp --ma Apr 20
me (no subject) - --stats_file=report.txt --protocol=icmp --m Apr 20
me (no subject) - --stats_file=report.txt --protocol=tcp --ma Apr 20
me (no subject) - --stats_file=report.txt --protocol=udp --ma Apr 20
No recent chats
Start a new one
Right Ctrl
Address
ENG
10:01
24-04-2017

Defense Phase

There are a series of steps taken to mitigate the attack which are as follows:

- First blocking all the ports which are uncommon, as most of the ports after 1024 are uncommon or rarely used,hence blocking all the ports after 1024.
- Now for blocking the ports we have used iptables

4r0r4@enjoy: ~/Desktop/cns/try/defense_module

```
4r0r4@enjoy:~/Desktop/cns/try/defense_module$ python scrap_blacklist.py
Downloading: blacklist.xml Bytes: 201064405
  47497216  [23.62%]
```

```
from bs4 import BeautifulSoup
from urllib import urlopen
import requests


url = 'http://www.justdial.com/Delhi-NCR/IIT-Tutorials-<near>-Tilak-Nagar/ct-11669'

headers = {'User-agent': 'Mozilla/5.0'}
webpage = requests.get( url, headers=headers )

soup = BeautifulSoup(webpage.content, "html.parser")
#soup = BeautifulSoup(open("/home/4r0r4/Desktop/Tutorials For Pre Medical in Tilak Nagar, Delhi | Justdial.html"))

print(soup)

stores = soup.find_all("div", class_="store-details")

count = 0
details=[]      #list of dicts  [{},{}...]
for store in stores:

        store_details={}

        store_details['name'] = store.find('h4', class_="store-name").find('span', class_="jcn").get_text()
        print(store_details['name'])

        store_details['contact'] = store.find('p', class_="contact-info").get_text()
        print(store_details['contact'])

        address = store.find('p', class_="address-info").get_text()
        display_address = address[50:81].rstrip().lstrip()
        full_address = address[288:].rstrip()
        store_details['address'] = [display_address, full_address]      #address as list of display addr and full addr
        print(store_details['address'][0])
        print(store_details['address'][1])

        store_details['estd_year'] = store.find('span', class_="year").get_text()
        print(store_details['estd_year'])

        #store_details['images'] = images[0].find('img')['src']
        #count += 1
        #print(store_details['images'])
```

Python ▾ Tab Width: 8 ▾ Ln 6, Col 1 INS

~/Desktop/cns/try/defense_module/scrap_blacklist.py - Sublime Text (UNREGISTERED)

scrap_blacklist.py × | hitler.py × | c&c_soldier.py × | try.py ×

```
import urllib2

url = "http://www.unsubscore.com/blacklist.xml"

file_name = url.split('/')[-1]
u = urllib2.urlopen(url)
f = open(file_name, 'wb')
meta = u.info()
file_size = int(meta.getheaders("Content-Length")[0])
print "Downloading: %s Bytes: %s" % (file_name, file_size)

file_size_dl = 0
block_sz = 8192
while True:
    buffer = u.read(block_sz)
    if not buffer:
        break

    file_size_dl += len(buffer)
    f.write(buffer)
    status = r"%10d  [%3.2f%%]" % (file_size_dl, file_size_dl * 100. / file_size)
    status = status + chr(8)*(len(status)+1)
    print status,

f.close()
```

Line 7, Column 19 Tab Size: 4 Python

kidbuntu_1 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Terminal

4r0r4@enjoy: ~/Desktop/cns/Final/attack_module

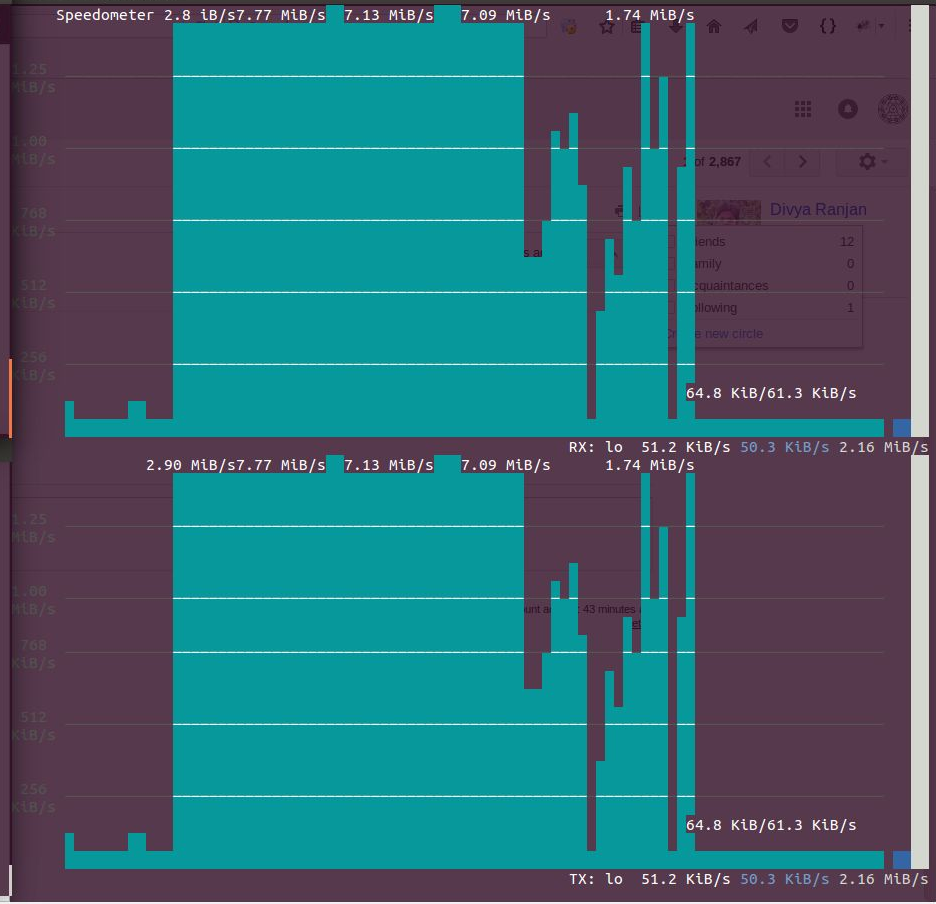
```
1336 packets in 1.000002 seconds
19829 packets in 1.000001 seconds
22629 packets in 1.000143 seconds
910 packets in 1.001291 seconds
924 packets in 1.000764 seconds
1003 packets in 1.000319 seconds
1021 packets in 1.000755 seconds
805 packets in 1.001844 seconds
903 packets in 1.001899 seconds
879 packets in 1.000565 seconds
890 packets in 1.000282 seconds
932 packets in 1.000928 seconds
996 packets in 1.000552 seconds
889 packets in 1.000402 seconds
984 packets in 1.000644 seconds
986 packets in 1.001678 seconds
955 packets in 1.000635 seconds
918 packets in 1.000715 seconds
849 packets in 1.000264 seconds
694 packets in 1.001969 seconds
926 packets in 1.008153 seconds
872 packets in 1.000014 seconds
814 packets in 1.000517 seconds
```

BlockIP.py BlockIP.pyc defense.py log.txt monitor.py port_block.py stats

4r0r4@enjoy: ~/Desktop/cns/Final/defense_module

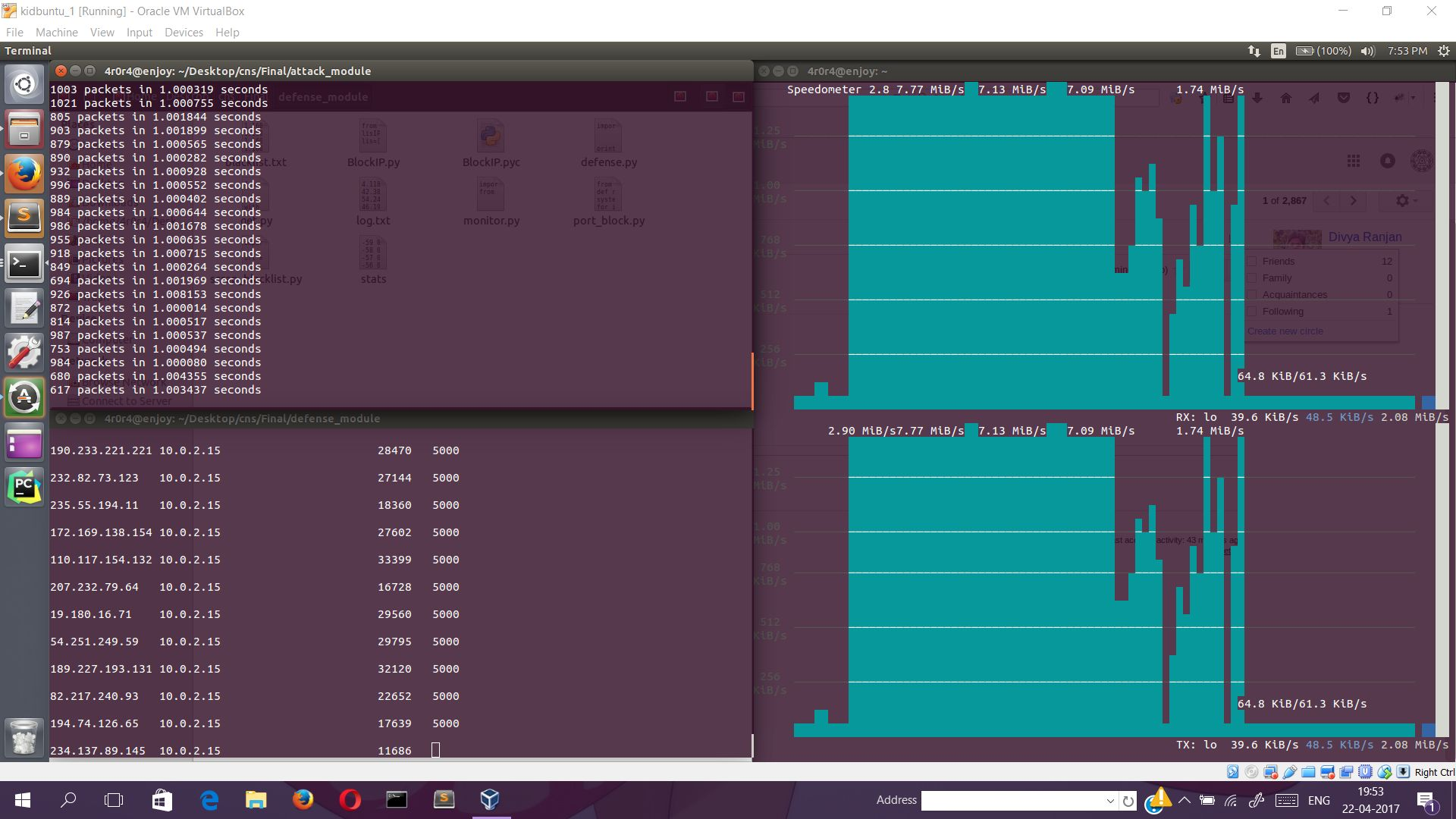
```
190.233.221.221 10.0.2.15                31032   5000

232.82.73.123   10.0.2.15                10068   5000

235.55.194.11   10.0.2.15                26639   5000

172.169.138.154 10.0.2.15                15713   5000

110.117.154.132 10.0.2.15                10139   5000

207.232.79.64   10.0.2.15                10993   5000

19.180.16.71    10.0.2.15                10869   5000

54.251.249.59   10.0.2.15                32088   5000

189.227.193.131 10.0.2.15                17433   5000

82.217.240.93   10.0.2.15                24085   5000

194.74.126.65   10.0.2.15                29762   5000

234.137.89.145  10.0.2.15                24101   5000
```

4r0r4@enjoy: ~



Address

ENG 19:53 22-04-2017


kidbuntu_1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
4r0r4@enjoy: ~/Desktop/cns/Final/attack_module
1003 packets in 1.000319 seconds
1021 packets in 1.000755 seconds
805 packets in 1.001844 seconds
903 packets in 1.001899 seconds
879 packets in 1.000565 seconds
890 packets in 1.000282 seconds
932 packets in 1.000928 seconds
996 packets in 1.000552 seconds
889 packets in 1.000402 seconds
984 packets in 1.000644 seconds
986 packets in 1.001678 seconds
955 packets in 1.000635 seconds
918 packets in 1.000715 seconds
849 packets in 1.000264 seconds
694 packets in 1.001969 seconds
926 packets in 1.008153 seconds
872 packets in 1.000014 seconds
814 packets in 1.000517 seconds
987 packets in 1.000537 seconds
753 packets in 1.000494 seconds
984 packets in 1.000080 seconds
680 packets in 1.004355 seconds
617 packets in 1.003437 seconds
BlockIP.py
BlockIP.pyc
defense.py
log.txt
monitor.py
port_block.py
stats
4r0r4@enjoy: ~/Desktop/cns/Final/defense_module
190.233.221.221 10.0.2.15 28470 5000
232.82.73.123 10.0.2.15 27144 5000
235.55.194.11 10.0.2.15 18360 5000
172.169.138.154 10.0.2.15 27602 5000
110.117.154.132 10.0.2.15 33399 5000
207.232.79.64 10.0.2.15 16728 5000
19.180.16.71 10.0.2.15 29560 5000
54.251.249.59 10.0.2.15 29795 5000
189.227.193.131 10.0.2.15 32120 5000
82.217.240.93 10.0.2.15 22652 5000
194.74.126.65 10.0.2.15 17639 5000
234.137.89.145 10.0.2.15 11686
4r0r4@enjoy: ~
Speedometer 2.8 7.77 MiB/s 7.13 MiB/s 7.09 MiB/s 1.74 MiB/s
64.8 KiB/61.3 KiB/s
RX: lo 39.6 KiB/s 48.5 KiB/s 2.08 MiB/s
2.90 MiB/s7.77 MiB/s 7.13 MiB/s 7.09 MiB/s 1.74 MiB/s
64.8 KiB/61.3 KiB/s
TX: lo 39.6 KiB/s 48.5 KiB/s 2.08 MiB/s
Divya Ranjan
Friends 12
Family 0
Acquaintances 0
Following 1
Create new circle
Right Ctrl
Address
ENG
19:53
22-04-2017

What are iptables

- Iptables as the name suggests contains a list of rules that has to be followed when a packet arrives to the network. We can take decision on what to do with the packets received with the help of iptables
- So,through iptables we are setting up our own firewall that allows connection packets to be received only at ports less than 1025, packets coming at ports will simply be dropped

SETTING UP THE FIREWALL

_ _ _

- After completing the first step, the problem arises if the attack happens at the common ports, in this case ports above 1024. For that, while attack is happening, we will capture the source ip and destination port of the packets coming to our IP address. For the capturing of data , we have used t-shark(a command line interface of wireshark). Data captured through t-shark is dumped into a log file.

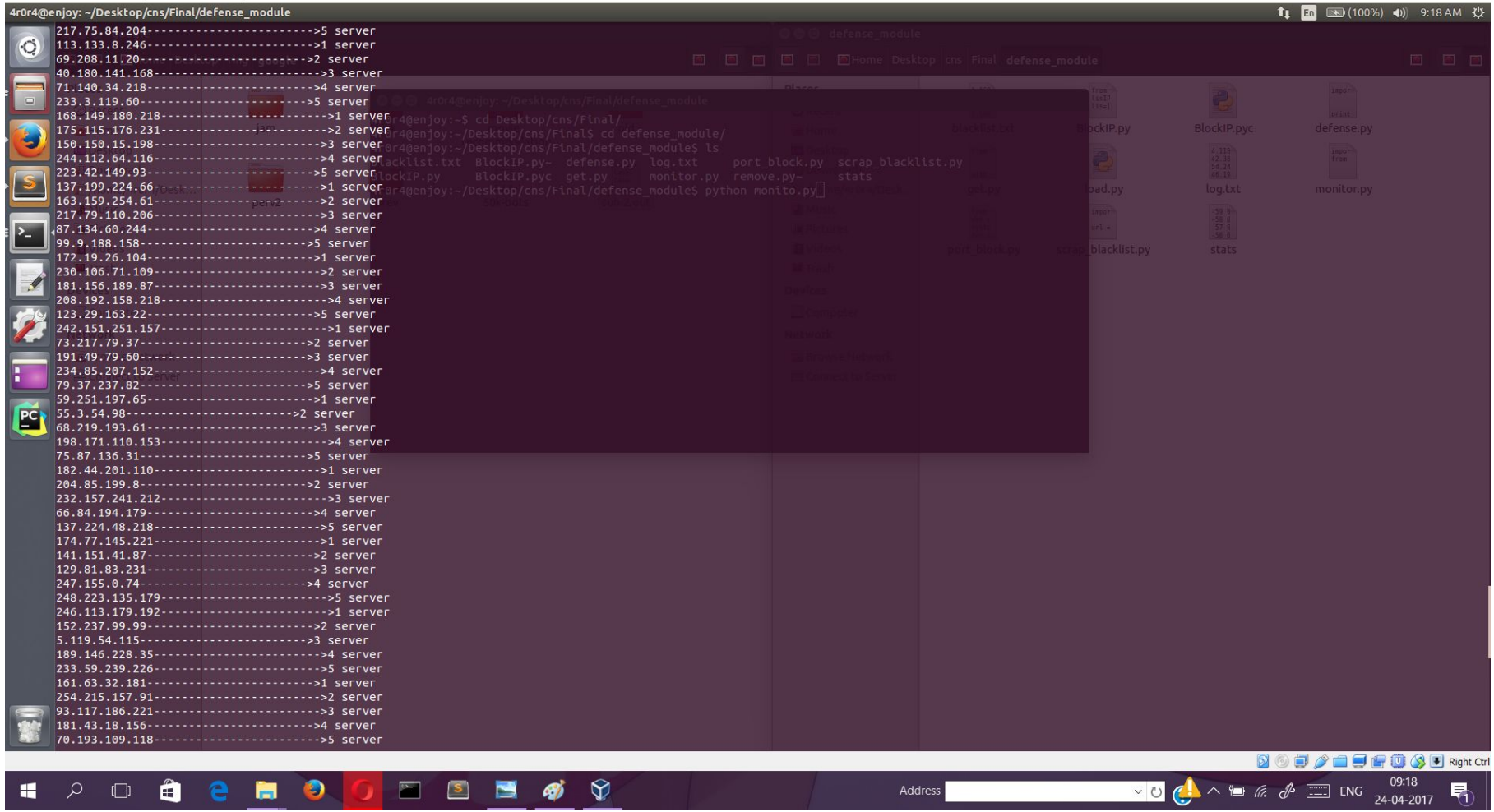
FIREWALL CONTINUATION...

– – –

- Now to actually block the attack, we have written a python script get.py which takes values from the log and invokes specific functions (as the data as argument) to block a particular port(ignore packets coming on that port) or to simply reject all the packets coming from a particular ip address.
- However in some cases, it might not be necessary to block ip as whole,but just a combination the source ip address and destination ports.

LOAD DISTRIBUTION

- A load balancer is a device that acts as a reverse proxy and distributes network or application traffic across a number of servers. Load balancers are used to increase capacity (concurrent users) and reliability of applications.
- We wrote a simple python script to simulate the distribution of the load based on the number of servers based in a round robin fashion.


kidbuntu_1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
4r0r4@enjoy: ~/Desktop/cns/Final/defense_module
9:18 AM
217.75.84.204----------->5 server
113.133.8.246----------->1 server
69.208.11.20----------->2 server
40.180.141.168----------->3 server
71.140.34.218----------->4 server
233.3.119.60----------->5 server
168.149.180.218----------->1 server
175.115.176.231----------->2 server
150.150.15.198----------->3 server
244.112.64.116----------->4 server
223.42.149.93----------->5 server
137.199.224.66----------->1 server
163.159.254.61----------->2 server
217.79.110.206----------->3 server
87.134.60.244----------->4 server
99.9.188.158----------->5 server
172.19.26.104----------->1 server
230.106.71.109----------->2 server
181.156.189.87----------->3 server
208.192.158.218----------->4 server
123.29.163.22----------->5 server
242.151.251.157----------->1 server
73.217.79.37----------->2 server
191.49.79.60----------->3 server
234.85.207.152----------->4 server
79.37.237.82----------->5 server
59.251.197.65----------->1 server
55.3.54.98----------->2 server
68.219.193.61----------->3 server
198.171.110.153----------->4 server
75.87.136.31----------->5 server
182.44.201.110----------->1 server
204.85.199.8----------->2 server
232.157.241.212----------->3 server
66.84.194.179----------->4 server
137.224.48.218----------->5 server
174.77.145.221----------->1 server
141.151.41.87----------->2 server
129.81.83.231----------->3 server
247.155.0.74----------->4 server
248.223.135.179----------->5 server
246.113.179.192----------->1 server
152.237.99.99----------->2 server
5.119.54.115----------->3 server
189.146.228.35----------->4 server
233.59.239.226----------->5 server
161.63.32.181----------->1 server
254.215.157.91----------->2 server
93.117.186.221----------->3 server
181.43.18.156----------->4 server
70.193.109.118----------->5 server
4r0r4@enjoy:~$ cd Desktop/cns/Final/
4r0r4@enjoy:~/Desktop/cns/Final$ cd defense_module/
4r0r4@enjoy:~/Desktop/cns/Final/defense_module$ ls
blacklist.txt BlockIP.py~ defense.py log.txt port_block.py scrap_blacklist.py
BlockIP.py BlockIP.pyc get.py monitor.py remove.py~ stats
4r0r4@enjoy:~/Desktop/cns/Final/defense_module$ python monito.py
Right Ctrl
Address
ENG
09:18
24-04-2017

DDOS attack Mitigation in Real World

1. **Specialized On-Premises Equipment.** Enterprise do all the work to stop the attack, but instead of relying on scripts or an existing firewall, they purchase and deploy dedicated DDoS mitigation appliances. These are specialized hardware that sit in an enterprise’s data center in front of the normal servers and routers and are specifically built to detect and filter the malicious traffic.

DDOS attack Mitigation in Real World

– – –

2.Cloud Mitigation Provider. Cloud mitigation providers are experts at providing DDoS mitigation from the cloud. This means they have built out massive amounts of network bandwidth and DDoS mitigation capacity at multiple sites around the Internet that can take in any type of network traffic, whether you use multiple ISP’s, your own data center or any number of cloud providers. They can scrub the traffic for you and only send “clean” traffic to your data center.

WHY DDOS CAN’T BE STOPPED COMPLETELY ???

– – –

THANK YOU