IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 20, Issue 3, Ver. I (May. - June. 2018), PP 67-77
www.iosrjournals.org

An Approach for Preventing Dos Attacks in ISP Companies

Dr. Mohsen Hossien Moh'd [1], Dr. Khaled Mahmood ALadhal[2] ,
Amjed Zaid Thabet[3]
[1](IT Department, *Faculty of Engineering*/ Aden University , Aden-Yemen)
[2]*(Computer Department Faculty of Computer/ Aden University , Aden-Yemen)*
[3](IT Department, *Faculty of Engineering*/ Aden University , Aden-Yemen)
Corresponding Author: Dr. Mohsen Hossien Moh'd

Abstract: *Distributed Denial of service (DDOS) has the most dangerous economics damages DDoS Attacks have plagued the Internet, corporate websites, and networks for more than a decade. Although DDoS attacks aren't new, modern threats and tactics are more advanced than ever. DDoS attacks are occurring with increasing frequency and causing a great damage against a rapidly growing number of targets worldwide.*

The most difficult problem against the defense of the Distributed Denial of service attack is how to distinguish between the legitimate traffic and the real traffic? To protect ISP companies against such attack, an IPS system implementation are needed to detect and prevent flooding attack, as it is one of the challenging issue. This paper introduces the major attacks types and tools used by attacker and study the mitigation techniques by implementing snort as IPS system. Moreover, the paper examines various mechanisms of distributed denial of service attacks, its detection, and various approaches to handle these attacks.

Keywords: *DOS and DDOS Attacks, Denial of Service (DOS), Detection, Prevention, Mitigation, TCP, UDP and ICMP Attack, Echo Request, Snort, IPS and IDS*

Date of Submission: 14-05-2018 Date of acceptance: 31-05-2018

I. INTRODUCTION

The last few years have witnessed a transformation in the profile of the attackers themselves, as well as the types of *Distributed Denial of service (*DDoS) attacks that they are carrying out, as internet communication has become more dependent nowadays for the modern life. Any disruption like DDoS attacks can cause various harms to the social and business networks. The DDoS attack leverages multiple sources to create the denial-of-service condition. By using multiple sources to attack a victim, the mastermind behind the attack is not only able to amplify the magnitude of the attack, but also can better hide his/her actual source IP address. Although the methods and motives behind Denial of Service attacks have changed, the fundamental goal of attacks, is to deny legitimate users resources or service. The main purpose of the attacker is to disable popular sites, banks, … etc. for the financial gain, economic gain, revenge, … etc.. During DDoS attacks, attackers bombard their target with a massive number of requests or data – exhausting its network or computing resources and preventing legitimate users from having access. Simply, a DDoS attack is occurred when an attacker uses a single machine's resources to exhaust those of another machine in order to prevent them from functioning normally.

II. Literature Review

Distributed denial of service (DDoS) attacks continue to harm enterprises around the world. The obvious damage caused by DDoS attacks is bad enough, including headline-grabbing, and multigigabit/second volumetric attacks that crash critical business and government systems. The most insidious is the use of DDoS as a component of advanced targeted attacks. Many of these attacks include DDoS components designed to stay beneath the network security radar, mimicking legitimate user traffic to escape detection. Protective security services start failing or, are blocked altogether. During the confusion caused by the DDoS, the real infiltration takes place: Malware and attacks infect web applications or dig deeper into the network during the confusion [11].

DDoS is no longer just a tool for political and social attacks to make a statement and shut down a site. It is now an insidious and easy-to-execute component of attacks that today's protective measures are often unable to detect [3].

ATA research being conducted by attackers has also changed the nature of DDOS attacks. Brute force, DDOS exploits that bombard targets with large amounts of data at high speed, continue to grow both in sheer bandwidth and frequency. Now, attackers have also begun to analyze the application logic running on the target's website. This allows the attacker to cause denial of service through "resource depletion" attacks such as starting user authentication processes, site-wide searches, or a new account creation processes that consume high percentages of server CPU cycles or memory space without requiring large amounts of network traffic from the attacker. Such an attack achieves the attacker's goal (denial of service) but is more difficult to detect and mitigate than a simple brute force attack.

The Defiance Advanced Research Projects Agency (DARPA) sponsored the MIT Lincoln Labs to create the first well-known IDS evaluation procedure. This was in 1998 when they created the DARPA Evaluation 1998. The aim was to "perform a comprehensive technical evaluation of intrusion detection technology" [12].

One year later, MIT Lincoln Labs started the DARAP 1999 Evaluation with the aim to provide an "unbiased measurement of current performance levels." Another objective was to produce an experimental dataset that could be used by others searchers [13]. It was composed of four targets running with the four most popular operating systems at that moment (Linux 2.0.27, SunOS 4.1.4, Sun Solaris 2.5.1, and Windows NT 4.0). It also contains two sniffers, one in each side of the gateway routers. Inside the network, the DARPA 1999 team setup hundreds simulated PC and workstation. On the other side of the gateway, they setup thousands of simulated web servers that will represent the Internet [12].

III. METHODOLOGY

3.1 Test Bed Description

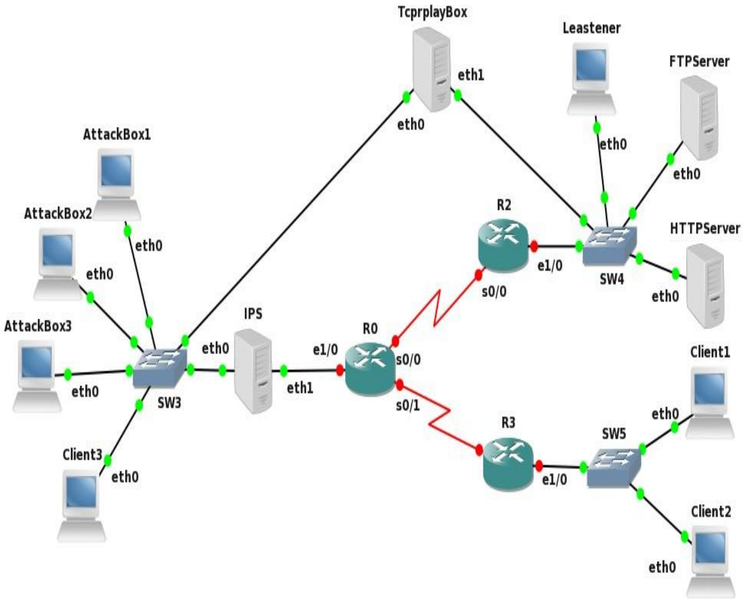


Fig.1 Design of Test bed

A computer with VMware workstation are used to create a private virtual network in order to conduct the experiment. Two VLANs were setup inside of VMware Workstation environment. The software VMware workstation 12 PRO (version 12.0.1 BUILD-3160714) has been used for this purpose four virtual machines are required: the first machine used to run the different attack, the second machine to run Snort, the third machine to act as the DUT (device under attacks), or target of attacks, and the fourth machine to generate background traffic. Three machines are used for running the kali GNU Linux Distribution, and the fourth machine is running with Ubuntu and snort installed and configured as bridge. The specifications for the four machines are presented in Table 3.1.

Table 3.1 The specifications for the four machines.

Table 5.2.1 – Specifications of Virtual Machines in the two VLANs	**Virtual Machines MAC ADDRESS**	**Operating System**	**CPU (shared)**	**Memory**
ATTACKER VM 1	00:0C:29:BB:DE:BC IP ADDRESS: 192.168.101.100	Kali GNU/ Linux ROLLING 32-BIT Distribution HPING3 VER.3.0.0-ALPHA-2	Intel Core i7-2600 CPU @ 3.40GHz NUMBER OF PROCESSOR: 1 NUMBER OF CORES	1003.2 MB

			PER PROCESSOR: 1	
SNORT VM 2	00:0C:29:71:33:4C 00:0C:29:71:33:42	UBUNTU 16.04 LTS SNORT VER.2.9.9.0 GRE (BUILD 56) USING LIBPCAP VER. 1.7.4 USING PCRE VER. 8.38 USING ZLIB VER. 1.2.8 (CONFIGURED AS BIRDGE B/W THE TWO Vmnet)	Intel Core i7-2600 CPU @ 3.40GHz NUMBER OF PROCESSOR: 1 NUMBER OF CORES PER PROCESSOR: 4	6000MB
VICTUM SERVER VM 3	00:0C:29:E7:A5:7D 192.168.100.135	Kali GNU/ Linux ROLLING 32-BIT Distribution IPTRAF-NG VER.1.1.4 WIRESHARK NETWORK PROTOCOL ANALYZER VER.2.0.1	Intel Core i7-2600 CPU @ 3.40GHz NUMBER OF PROCESSOR: 1 NUMBER OF CORES PER PROCESSOR: 2	3000 MB
background traffic generater (DARPA Dataset) VM 4	00:0C:29:6F:DE:AA 00:0C:29:6F:DE:A0 IP ADDRESS:192.168.100.191 ADDRESS:192.168.100.190 DARPA	Kali GNU/ Linux ROLLING 32-BIT Distribution	Intel Core i7-2600 CPU @ 3.40GHz NUMBER OF PROCESSOR: 1 NUMBER OF CORES PER PROCESSOR: 1	1003.2 MB

As the (figure1) above shows, the VM 4 virtual machine will generate background traffic, and VM 1 will sending all attacks to VM 3 through VM 2, while VM 2 is configured to run Snort and monitor all network traffic seen in this virtual environment. In the target machine VM 3, the server is configured to accept both TCP and UDP connections in order to realistically allow for VM 1 to generate attacks on it. After each test is ran with a variation in background traffic playback speed, VM 2 will run snort in two modes, monitor and active (drop mode).

Table 3.2: The specifications in the Hping3 packet attack generator.

Attack	Source		Destination		Port
	M A C address	IP address	M A C address	IP address	
UDP	00:0C:29:BB:DE:BC	192.168.101.100	00:0C:29:E7:A5:7D	192.168.100.135	Random
ICMP	00:0C:29:BB:DE:BC	192.168.101.100	00:0C:29:E7:A5:7D	192.168.100.135	Random
TCP	00:0C:29:BB:DE:BC	192.168.101.100	00:0C:29:E7:A5:7D	192.168.100.135	Random

From the experiments through a virtual environment, the ability to ensure no unexpected traffic had seen on the network achieved. Furthermore, a greater control over experiments was seemed possible since the four machines form a private virtual network. However, it was acknowledged that one major limitation in carrying out the experiments in this environment that it may not accurately reflect a real-life network. Unfortunately, due to time limitations, testing using real-life equipment in a laboratory setting was unable to be achieved.

3.1.1 IMPLEMENTATION SCENARIOS

Scenarios No: 1

- Applying only background traffic without any attack traffic, and snort will be sit in alert mode, record and capture network traffic using Wireshark and iptraf-ng at the DUT for further detailed analysis.

Scenarios no: 2

- Applying background traffic along with flood attack traffic (ICMP, TCP and UDP flood attack once at a time), and snort will be set alert mode to record and capture network traffic using Wireshark and iptraf-ng at the DUT for further detailed analysis.

Scenarios no: 3

- Applying background traffic along with flood attack traffic (ICMP, TCP, UDP flood attack once at a time), and snort will be set drop mode (drop mode), record and capture network traffic using Wireshark and iptraf-ng at the DUT for further detailed analysis.

3.2 ATTACK AND DETECTION PHASE

3.2.1 Attack Component Design

The tool Hping3 can be used in this experiment and has a scripting capacity over the TCP/IP stack [4] and allows the creation of nearly any kind of IP packet. This can be seen as a scriptable TCP/IP stack [5], and can be used to generate, in real time, crafted packets that will match the behavior of a DDoS profile. Three different attacks will be used: TCP-based; UDP-based, ICMP-based. The TCP-based, UDP-based, and ICMP-based will be against the server. By using three different protocols in this experiment, it will be possible to figure out if the protocols have an impact on the capacity to handle attacks on the DUT.

Table 3.3: DDoS attack used

Distributed Denial of Service	Description
TCP-based	TCP packets and SYN flag on flood attack.
UDP-based	UDP packets flood attack.
ICMP-based	ICMP echo request packets.

IV. Type of attack and tools used:

There are several flavors of Denial of Service that could disrupt a normal service. The attacking methods are classified into two methods according to Erikson Jon [1].

- The first method is to flood the network not leaving enough bandwidth for the legitimate packets to get through. This could also be termed as Flooding.
- The other method is to crash a hardware or software item and make it inoperable. Web servers, routing devices, DNS look up servers are the common targets that could be crashed during an attack.

This experiment has investigated both the scenarios and has analyzed its effects. The DDoS paper published by Lee Garber talks about the mechanisms involved in some common attack types.

The attack from hping3 packet generator could be initiated from the Linux terminal. The following commands decide what kind of attack to be launched against the server under attack. The characters and the keywords used for various attacks are first explained below. Hping3 is a network tool able to send custom ICMP/UDP/TCP packets and to display target replies like ping do with ICMP replies. Hping3 handles fragmentation, arbitrary packet body and size and can be used in order to transfer files under supported protocols [8].

Hping3 can be used among other things to [14]:

- Test firewall rules.
- [spoofed] port scanning.
- Test net performance using different protocols, packet size, TOS (type of service) and fragmentation.
- Path MTU discovery.
- Files transferring even between really fascist firewall rules.
- Trace route like under different protocols.
- Fire walk like usage.
- Remote OS fingerprint.
- TCP/IP stack auditing.

Background Traffic Component Design

The defiance Advanced Research Projects Agency (DARPA) sponsored the MIT Lincoln Labs to create the first well-known IDS evaluation procedure. This was in 1998 when they created the DARPA Evaluation1998. The aim was to "perform a comprehensive technical evaluation of intrusion detection technology" [12]. One year later, MIT Lincoln Labs started the DARPA 1999 Evaluation with the aim to provide an "unbiased measurement of current performance levels." Another objective was to produce an experimental dataset that could be used by other searchers [13]. It was composed of four targets running with the four most popular operating systems at that moment (Linux 2.0.27, SunOS 4.1.4, Sun Solaris 2.5.1, and Windows NT 4.0). It also contained two sniffers, one of each side of the gateway routers. Inside the network, the DARPA 1999 team setup hundreds simulated PCs and workstations. On the other side of the gateway, they setup thousands of simulated web servers that will represent the Internet [12].

In these experiments, DARPA dataset inside.tcpdump.gz 1999 week 1, Tuesday, march3 will be used to generate the background network traffic. Also, Tcpprep, Tcprewrite, and Tcpreplay have been used in these experiments to playback the modified dataset as background traffic which offers the possibility to repeat dataset with different options (such as for network speed).

V. Results

Each experiment has been carried out as described in implementation scenarios. The IPS was stopped and all data recorded for further analyses later on. In addition, the machine hosting the IPS was rebooted between each experiment to ensure that the environment is the same between each of them.

5.1 Scenarios No: 1

- When Applying only background traffic without any attack traffic, and snort was set in alert mode, while record and capture network traffic using Wireshark and iptraf-ng at the DUT for further detailed analysis.

All normal background traffic packets and other normal communication control packets were captured during approximately 10 minutes of network activities, while Snort in alert mode, and no flood attacking traffic. This traffic was generated by tcpreply using DARPA Dataset Monday. Furthermore, it can be observed that approximately 541016 packets also recorded by iptraf-ng 1.1.4 during the approximately the same period of time which consist of 502088 TCP packets, 38444 UDP packets, and 484 ICMP packets.

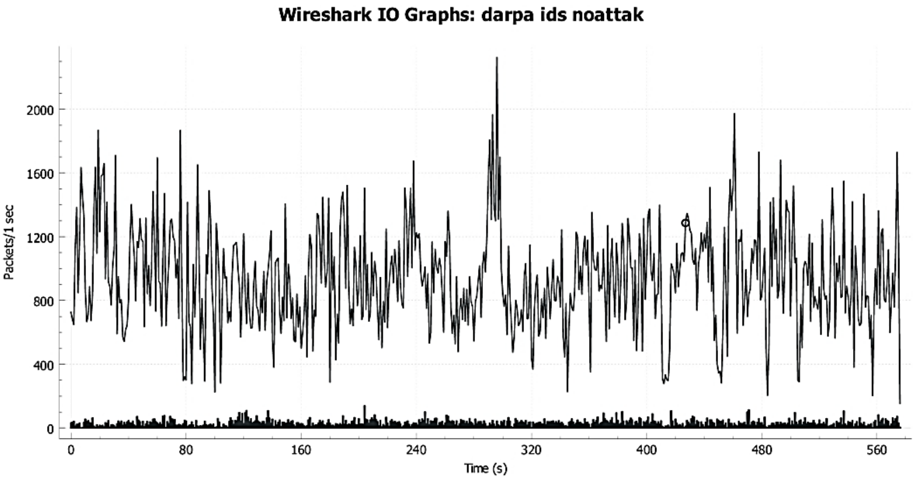


Fig. 2: IO graph network traffic rate monitored without attack while Snort in alert mode

Figure 2 above shows us the network activities, the packets start to transmit from 0 up to 560 second, we can observe that the traffic rate below 1600 packets/sec

5.2 Scenarios No: 2

5.2.1 Smurf attack (ICMP Flooding)

The Smurf attack is a distributed denial-of-service attack in which large numbers of Internet Control Message Protocol (ICMP) packets with the intended victim's spoofed source IP are broadcast to a computer network using an IP broadcast address. Most devices on a network will, by default, respond to this by sending a reply to the source IP address. If the number of machines on the network that receive and respond to these packets is very large, the victim's computer will be flooded with traffic. This can slow down the victim's computer to the point where it becomes impossible to work on [9].

The background traffic combined with legitimate ICMP flood attack traffic using hping3 utilities using the command: hping3 -1 192.168.100.135 --flood. Hping3 generates random ICMP packets and targets against the server IP address 192.168.100.135, for approximately 10 minutes of network activities, while Snort set in alert mode.

All ICMP packets were captured during ICMP flood attack against the server. All packets were destined towards the IP address 192.168.100.135 which is the server IP address and the attacker was also from the same network making a direct impact over the server’s performance. Figure 3 shows when the attack has exactly, when its triggered by hiping3 tools using the command line interface. The ICMP traffic that was initially around zero kbps suddenly increased up to approximately 4800 packets/sec flooding the entire network. But the same attack could be strengthened if a number of attackers (zombies) increases at the same time towards the server. When such a cohesive attack was attempted, the server crashed without being able to handle the continuous packets.

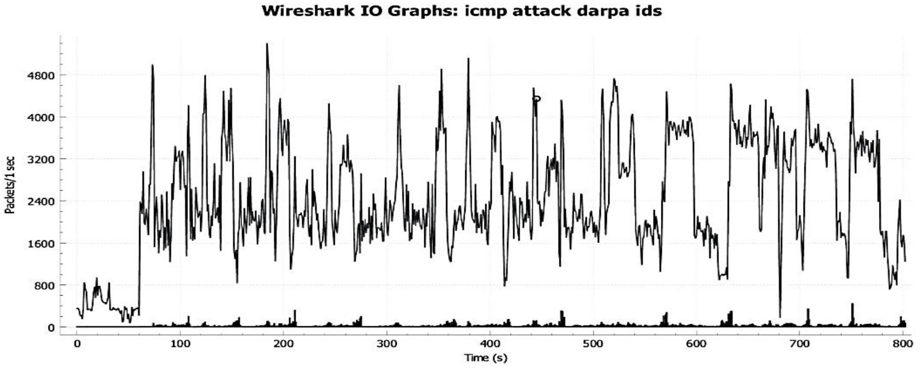


Fig 3: IO graph network traffic rate monitored during ICMP attack while snort set in alert mode

5.2.2 TCP (SYN) FLOOD TTACK

TCP SYN flood (a.k.a. SYN flood) is a type of Distributed Denial of Service (DDoS) attack that exploits part of the normal TCP "three-way handshake" to consume resources on the targeted server and render it unresponsive. Essentially, with SYN flood DDoS, the offender sends TCP connection requests faster than the targeted machine can process them, causing network saturation.

Attack Description

When a client and server establish a normal TCP "three-way handshake," the exchange looks like the following:

1. Client requests connection by sending SYN (synchronize) message to the server.
2. Server acknowledges by sending SYN-ACK (synchronize-acknowledge) message back to the client.
3. Client responds with an ACK (acknowledge) message, and the connection is established.

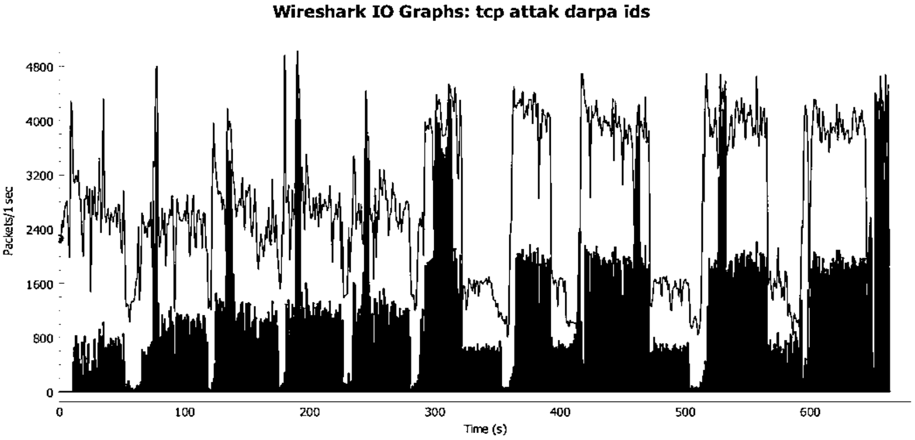


Fig 4: IO graph network traffic rate monitored during TCP SYN flood attack while snort set in alert mode

5.2.3 UDP FLOOD ATTACK

UDP flooding is similar to ping flood. Instead of ping packets, UDP packets are bombarded against the server. UDP could be a lot more effective than ICMP in smaller networks as the size of the UDP packets are enormous. The packet size could be set up to 65000 bytes. which could easily flood a given Ethernet network when multiple zombies are set up. This paper has analyzed all the above described attacks and has brought down some interesting observations.

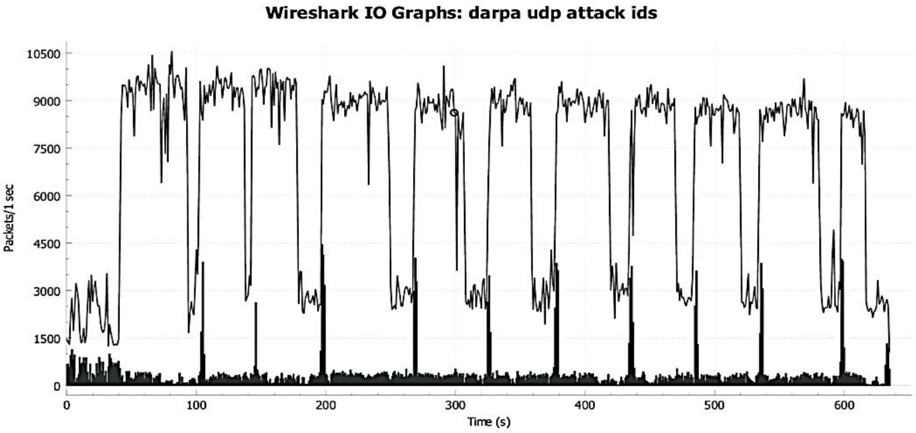


Fig 5: Network traffic rate monitored during UDP flood attack while snort set in alert mode

5.3 Scenarios No: 3

Applying background traffic along with flood attack traffic (ICMP, TCP and UDP flood attack once at a time), and snort will be set inline mode (drop mode), recording and capturing network traffic using Wireshark and Iptraf-ng at the DUT for further detailed analysis.

In this Scenarios, Snort will configure to run in inline mode and few modifications to **snort.conf** will be made, and the afpacket DAQ libraries should be available.

The **-Q** is noticed, and the **-i eth33: eth38** flags are new. The **-Q** flag tells Snort to run in inline mode, while **-i eth33: eth38** tells snort to bridge those two interfaces (to be in line between those two interfaces).

To have Snort drop traffic, there will be a need to modify the rule created above in the local rules from alert to drop. It should now look like the following:

We add the following drop rules to our local rules file and making sure that Snort loads it by testing our configuration and scrolling up to see that the rule is loaded. This rule will drop and alert to console whenever it sees an ICMP, TCP and UDP traffic attack.

The attacks will use three different protocols: ICMP, TCP and, UDP. The packets that will be used during the attacks are known. So, it is possible to analyze them and produce rules that will match the malicious packets pattern. For ICMP-based attack a default rules already exist that match these packets:

drop icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING NMAP"; dsize:0; itype:8; reference:arachnids,162; classtype:attempted-recon; sid:469; rev:4;)

For the TCP and UDP the following drop rules will be used:

drop icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING NMAP"; dsize:0; itype:8; reference:arachnids,162; classtype:attempted-recon; sid:469; rev:4;)

drop icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING NMAP"; dsize:0; itype:8; reference:arachnids,162; classtype:attempted-recon; sid:469; rev:4;)

5.3.1 ICMP Flood Attack

During an ICMP flood attack while snort act as IPS inline mode (drop mode) the following graph was obtained, which shows the traffic pattern. From figure 6, it could be seen that the Network traffic seems normal with peak rate of approximately 1200 packet/sec thereby the attack doing the minimum damage to the server. Figure 7 shows that snort drops 96.332% of the ICMP attack traffic. It could be observed that snort inline block huge attack traffic which could affect the network and server performance.

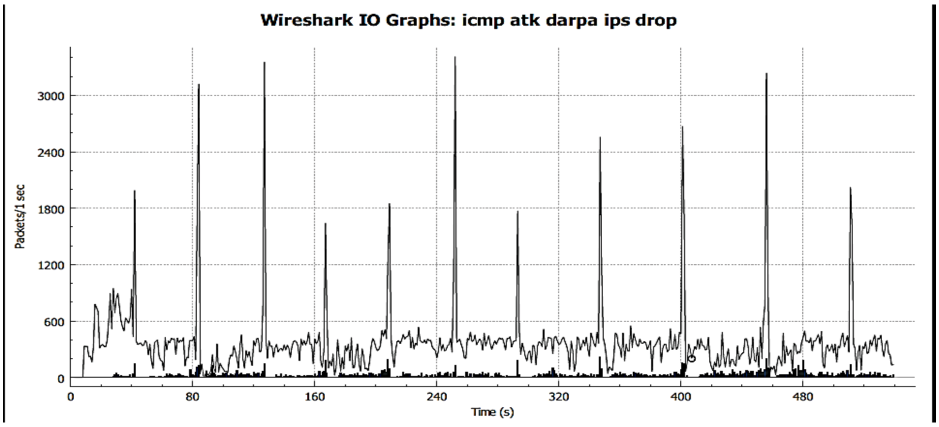
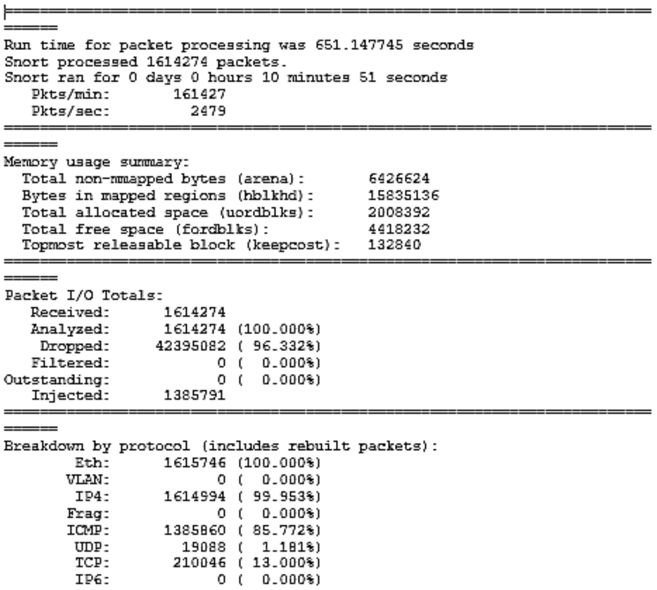


Fig 6: Network traffic rate monitored during ICMP flood attack while snort inline.



```
===================================================================
======
Run time for packet processing was 651.147745 seconds
Snort processed 1614274 packets.
Snort ran for 0 days 0 hours 10 minutes 51 seconds
   Pkts/min:       161427
   Pkts/sec:         2479
===================================================================
======
Memory usage summary:
  Total non-mmapped bytes (arena):       6426624
  Bytes in mapped regions (hblkhd):      15835136
  Total allocated space (uordblks):      2008392
  Total free space (fordblks):           4418232
  Topmost releasable block (keepcost):   132840
===================================================================
======
Packet I/O Totals:
   Received:      1614274
   Analyzed:      1614274 (100.000%)
    Dropped:     42395082 ( 96.332%)
   Filtered:            0 (  0.000%)
Outstanding:            0 (  0.000%)
   Injected:      1385791
===================================================================
======
Breakdown by protocol (includes rebuilt packets):
        Eth:      1615746 (100.000%)
       VLAN:            0 (  0.000%)
        IP4:      1614994 ( 99.953%)
       Frag:            0 (  0.000%)
       ICMP:      1385860 ( 85.772%)
        UDP:        19088 (  1.181%)
        TCP:       210046 ( 13.000%)
        IP6:            0 (  0.000%)
```

Fig 7: Screenshot of Snort console report after ICMP attack

5.3.2 TCP (SYN) Flood Attack

During a TCP flood attack while snort act as IPS inline mode (drop mode) the following graph shows the traffic pattern. From figure 8, it could be seen that the network traffic seems to be decreased from peak 2800 packet/sec to normal rate, approximately, 1200 packet/sec, thereby, the attack is doing the minimum damage to the server. Figure 9 shows that snort drop 9.73% of the TCP attack traffic. It could be observed that snort inline blocked huge attack traffic which could be effect the network and the server performance.

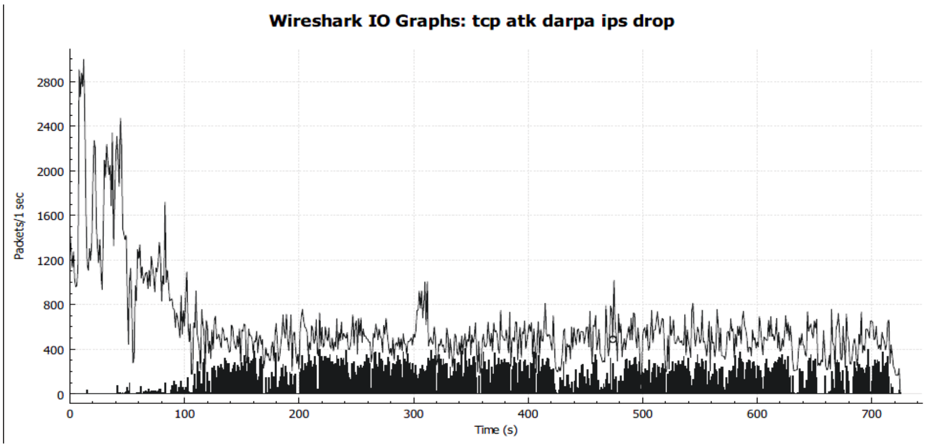
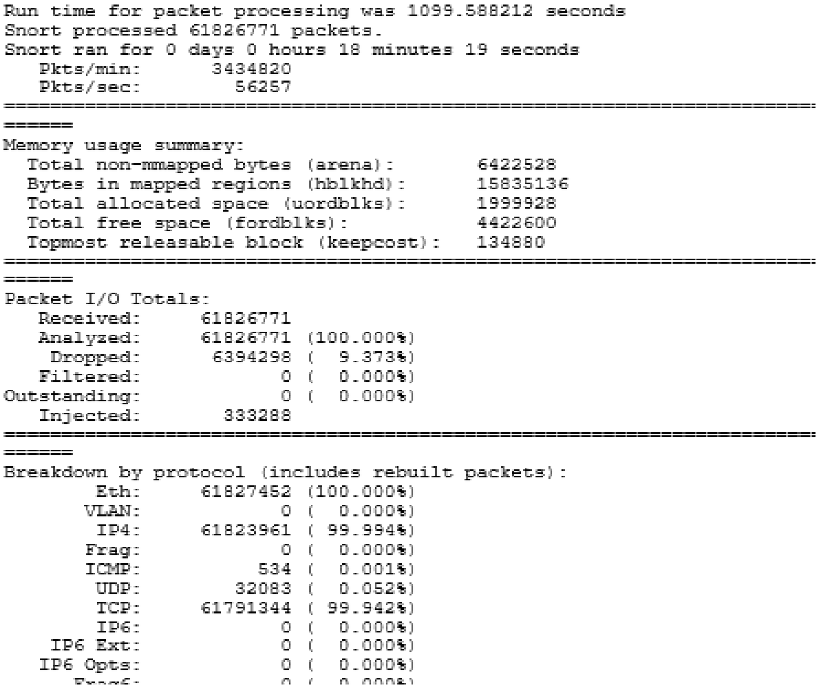


Fig 8: Network traffic rate monitored during TCP flood attack while snort inline.



```
Run time for packet processing was 1099.588212 seconds
Snort processed 61826771 packets.
Snort ran for 0 days 0 hours 18 minutes 19 seconds
   Pkts/min:      3434820
   Pkts/sec:        56257
=======================================================================
======
Memory usage summary:
  Total non-mmapped bytes (arena):       6422528
  Bytes in mapped regions (hblkhd):      15835136
  Total allocated space (uordblks):      1999928
  Total free space (fordblks):           4422600
  Topmost releasable block (keepcost):   134880
=======================================================================
======
Packet I/O Totals:
   Received:     61826771
   Analyzed:     61826771 (100.000%)
    Dropped:      6394298 (  9.373%)
   Filtered:            0 (  0.000%)
Outstanding:            0 (  0.000%)
   Injected:       333288
=======================================================================
======
Breakdown by protocol (includes rebuilt packets):
        Eth:     61827452 (100.000%)
       VLAN:            0 (  0.000%)
        IP4:     61823961 ( 99.994%)
       Frag:            0 (  0.000%)
       ICMP:          534 (  0.001%)
        UDP:        32083 (  0.052%)
        TCP:     61791344 ( 99.942%)
        IP6:            0 (  0.000%)
    IP6 Ext:            0 (  0.000%)
   IP6 Opts:            0 (  0.000%)
      Frag6:            0 (  0.000%)
```

Fig 9: Screenshot of Snort console report after TCP attack

5.3.3 UDP FLOOD ATTACK

During a UDP flood attack while snort act as IPS inline mode (drop mode), the following graph was obtained, which shows the traffic pattern. From figure 10, it could be seen that the network traffic seems to be decreased from peak 1600 packet/sec to normal rate approximately, 400 packets/sec, thereby, the attack is doing the minimum damage to the server. Figure 11 shows that snort drops 96.575% of the UDP attack traffic. It could be observed that snort inline block huge attack traffic which could affect the network and server performance.

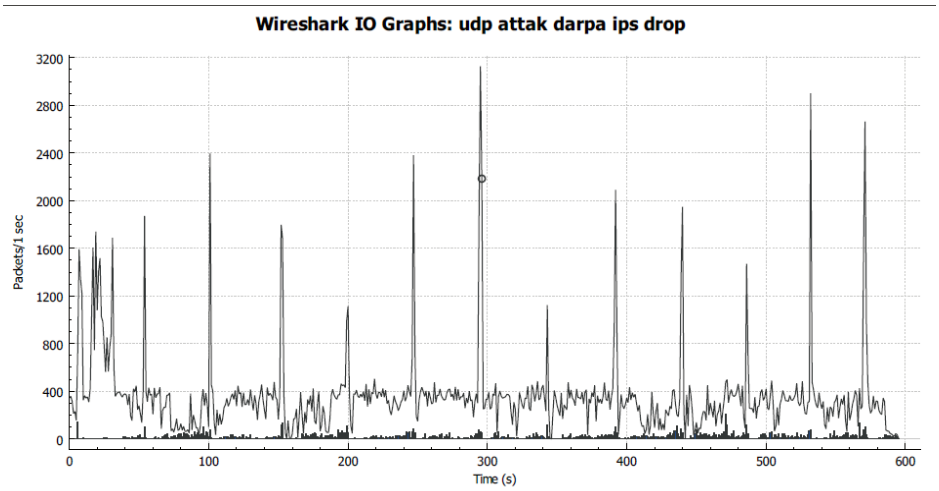
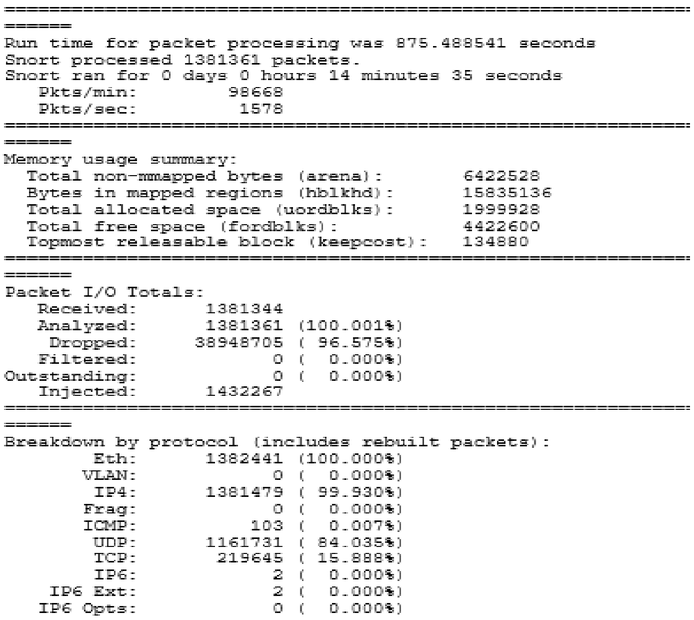


Fig 10: Network traffic rate monitored during UDP flood attack while snort inline.



```
==============================================================
======
Run time for packet processing was 875.488541 seconds
Snort processed 1381361 packets.
Snort ran for 0 days 0 hours 14 minutes 35 seconds
   Pkts/min:        98668
   Pkts/sec:         1578
==============================================================
======
Memory usage summary:
  Total non-mmapped bytes (arena):        6422528
  Bytes in mapped regions (hblkhd):       15835136
  Total allocated space (uordblks):       1999928
  Total free space (fordblks):            4422600
  Topmost releasable block (keepcost):    134880
==============================================================
======
Packet I/O Totals:
   Received:      1381344
   Analyzed:      1381361 (100.001%)
    Dropped:     38948705 ( 96.575%)
   Filtered:            0 (  0.000%)
Outstanding:            0 (  0.000%)
   Injected:      1432267
==============================================================
======
Breakdown by protocol (includes rebuilt packets):
        Eth:      1382441 (100.000%)
       VLAN:            0 (  0.000%)
        IP4:      1381479 ( 99.930%)
       Frag:            0 (  0.000%)
       ICMP:          103 (  0.007%)
        UDP:      1161731 ( 84.035%)
        TCP:       219645 ( 15.888%)
        IP6:            2 (  0.000%)
    IP6 Ext:            2 (  0.000%)
   IP6 Opts:            0 (  0.000%)
```

Fig 11: Screenshot of Snort console report after UDP attack

VI. Conclusion:

We can conclude from the Snort console report that snort drop 96.332% of the ICMP attack traffic, while snort drop 9.73% of the TCP attack traffic and drop 96.575% of the UDP attack traffic. This method is indeed the most effective as it prevents the traffic from attacking the network completely. This idea fails in the event of Distributed Denial of Service attacks as the zombies are spread across different networks.

The technique of the given scenario with only few sub networks and fewer hosts could be satisfying. However, many networks and hosts will be kept out of the server connection Cybersecurity. It is a must for a safer cyberspace; DDOS attacks, which exploit inherent weaknesses in the design and organization of the Internet. It can be easily launched using readily available tools against individual Web sites, chat servers, email servers. The growing dependence on the Internet makes the impact of these attacks increasingly painful for service providers, enterprises, hosting centers and government agencies alike.

Responding to and defeating these attacks in a timely and effective manner is the primary challenge confronting Internet–dependent organizations today. Traditional perimeter security technologies such as firewalls and IPS do not provide adequate DDoS protection. Filtering solutions such as router–based access control lists (ACLs) cannot separate good traffic from bad for most attacks, resulting in legitimate transactions being blocked. To protect them, businesses require a next–generation architecture, purpose–built specifically to detect and defeat increasingly sophisticated, complex and deceptive attacks without impacting ongoing business operations.

It is therefore recommended to implement a combination several techniques in the philosophy of a defense in-depth system aiming at hindering any kind of cyber-attacks

VII. Future works

This paper gives the reader adequate knowledge on how to implement Denial of Service attacks. It highlights various attack tools and the ways to identify the same in a given network. It also suggests basic mitigation strategies that could be adopted in order to defend attacks. However, serious challenges arise when IPv6 needs to be established globally and transition from version 4 to version 6 has to be done. IPv6 introduces six optional headers like Routing header, Authentication header etc [7].

In spite of providing better security with authentication, encryption and encapsulation techniques, IPv6 also brings out serious complications. The following two types of Denial of Service attacks could be implemented if IPv6 is used.

References

[1]. C. D. (1997, January). 1997 Tech Tip: Denial of Service Attacks. Retrieved August 23, 2017, from https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=496599

[2]. S. F., L. G., & P. O. (2005). Network Security and DoS Attacks (2nd ed.).

[3]. Bhattacharyya, ,. %., & Kalita, J. K. (2016). DDoS Attacks, Evolution, Detection,Prevention, Reaction,and Tolerance.

[4]. Parker, D. (2003). Hping. Retrieved February 12, 2010, from Hping: http://gd.tuwien.ac.at/www.hping.org/hping_conv.pdf

[5]. Sanfilippo, S. (2006). hping. Retrieved November 03, 2009, from hping: http://www.hping.org/documentation.php

[6]. Chen, T. & Walsh, P. J. (2009). Guarding against network intrusion. In Vacca (Ed.), Computer and information security handbook, Burlington, MA: Morgan Kaufmann

[7]. Dr Nick Zakhleniuk. University of Essex, "TP Networking and Applications," (Chapter 3) IPv6 Extension Headers (EHs) - IPv6 Security, March 2011.

[8]. hping3. (2014.). Retrieved October 12, 2017, from https://tools.kali.org/information-gathering/hping3

[9]. Smurf attack. (2017, October 29). In Wikipedia, The Free Encyclopedia. Retrieved 00:59, December 19, 2017, from https://en.wikipedia.org/w/index.php?title=Smurf_attack&oldid=807638289

[10]. Allen, J., Christie, A., Fithen, W., McHugh, J., Pickel, J., & Stoner, E. (2000). State of the Practice of Intrusion Detection Technologies. Pittsburgh: Carnegie Mellon University.

[11]. Pescatore, J. (2013). How DDoS Detection and Mitigation Can Fight Advanced Targeted Attacks, SANS Institute: 2.

[12]. Lippmann, R., Haines, J. W., Fried, F. D., Korba, J., & Das, K. (2000). The 1999 DARPA off-line intrusion detection evaluation. Computer Networks , 34 (4), 579-595.

[13]. McHugh, J. (2000). Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory. ACM Transactions on Information and System Security , 3 (4), 262-294.

[14]. Alexander Khalimonenko, O. K., Kirill Ilganaev (February 6, 2018).DDoS attacks in Q4 2017.

Dr. Mohsen Hossien Moh’d "An Approach for Preventing Dos Attacks in ISP Companies." IOSR Journal of Computer Engineering (IOSR-JCE) 20.3 (2018): 67-77.