12-1-2013

A Survey on Detection and Defense of Application Layer DDoS Attacks

Naga Shalini Vadlamani
University of Nevada, Las Vegas, shalinivadlamani@hotmail.com

A SURVEY ON DETECTION AND DEFENSE OF APPLICATION LAYER DDoS ATTACKS

By

Naga Shalini Vadlamani

Bachelor of Technology, Information Technology
Jawaharlal Nehru Technological University, India
2011

A thesis submitted in partial fulfillment
of the requirements for the

Master of Science - Computer Science

School of Computer Science
Howard R. Hughes College of Engineering
The Graduate College

University of Nevada, Las Vegas
December 2013

UNLV

UNIVERSITY OF NEVADA LAS VEGAS

THE GRADUATE COLLEGE

We recommend the thesis prepared under our supervision by

Naga Shalini Vadlamani

entitled

A Survey on Detection and Defense of Application Layer DDoS Attacks

is approved in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science

Department of Computer Science

Ju-Yeon Jo, Ph.D., Committee Chair

Yoohwan Kim, Ph.D., Committee Member

Laxmi Gewali, Ph.D., Committee Member

Venkatesan Muthukumar, Ph.D., Graduate College Representative

Kathryn Hausbeck Korgan, Ph.D., Interim Dean of the Graduate College

December 2013

ABSTRACT

A Survey on Detection and Defense of Application Layer DDoS Attacks

By

Naga Shalini Vadlamani

Dr. Ju-Yeon Jo, Examination Committee Chair

Associate Professor, Department of Computer Science

University of Nevada, Las Vegas

As the time is passing on, the effect of DDoS attacks on Internet security is growing tremendously. Within a very little span there is a huge increase in the size and frequency of DDoS attacks. With the new technologies and new techniques, the attackers are finding more sophisticated ways to attack the servers. In this situation, it is necessary to come up with various mechanisms to detect and defend these DDoS attacks and protect the servers from the attackers. Many researches have been carried out to detect the DDoS attack traffic in transport layer, which is more vulnerable to DDoS attacks. DDoS attacks are more common in transport layer. Coming to application layer, they incur huge loss and it is very difficult to mitigate DDoS attacks even under the presence of strong firewalls and Intrusion Prevention Security. Researches are being conducted to mitigate application layer DDoS attacks.

This Research contains a discussion of various types of DDoS attacks, their detection, and defense and prevention methods proposed by various researchers.

ACKNOWLEDGEMENTS

I would like to thank Dr. Ju-Yeon Jo, my research advisor for all the support and guidance she has offered me during the course of my graduate studies at University of Nevada, Las Vegas. Her encouragement and valuable suggestions have helped me immensely in seeking the right direction for this thesis. I would like to thank Dr. Yoohwan Kim, who deserves special recognition for his wholehearted guidance and support throughout my thesis.

I would also like to thank Dr. Laxmi Gewali, Dr. Yoohwan Kim and Dr. Venkatesan Muthukumar for serving my committee and reviewing my thesis. I am grateful to Dr. Ajoy K. Datta for all his support and guidance through my Master’s program.

This thesis has been a challenging experience to me and was accomplished through help of many people. In particular, I would like to express my sincere gratitude to Ms Ambu Yegappan for providing me assistantship at CAEO. I extend my appreciation and thanks to Amritha Premnath for all her support.

My deepest gratitude to my parents Seshagiri Rao Vadlamani and Kamala Vadlamani for their love, care and opportunities they have provided me at every stage of my life. I would also like to thank my brother Uday Vadlamani for his support and guidance in building up my career. I shall remain ever obliged to my uncle, Vishanatham Peri and all my cousins for their continuous support in all my endeavors. Last but not the least; I would like to thank all my friends and roommates for their support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The most common hurdle the internet services facing today comes from DDoS attacks. There are various tools that overwhelm the servers by launching Denial of Service attacks. With increased technology and sophisticated techniques, it became easy for the attackers to launch these attacks. When it comes to large network environments, it becomes even harder to detect these attacks. Hence, these attacks have become serious threats causing huge revenue losses to the Internet today. These attacks mainly target transport layer, network layer and application layer. In order to overcome this problem, we need more sophisticated methods to detect and defend these attacks. This research gives an insight about the approaches that are proposed by various researchers to detect and defend these kinds of attacks. This research mainly focuses on Application Layer DDoS attacks and their defense mechanisms.

1.1 Outline

Chapter 2 discusses in detail about the attacks which includes various types of network attacks and a brief introduction to DDoS attacks. Chapter 3 discusses about Introduction to DDoS attacks in Network layer, Transport layer and Application layer. Chapter 4 demonstrates an attack model, the experiment conducted under DDoS attacks on TCP in transport layer. Chapter 5 discusses in depth about attacks in Application layer which includes types of attacks and various mechanisms to detect and defend against DDoS attacks, comparison between various approaches and the final result. Chapter 6 discusses about the importance and need to develop new approaches to protect the web services from DDoS attacks.

CHAPTER 2

BACKGROUND AND LITERATURE

This chapter gives an insight about attack, various types of attacks in a network, and gives an introduction to DDoS attacks.

2.1 Attack

In computer networks, an Attack [1] refers to an attempt to destroy, expose, alter, disable, steal or gain unauthorized access to or make an unauthorized use of an asset. Usually attacks can be classified into two types, one is an attack which targets particular software and other is an attack which targets the protocols and web services.

2.1.1 Types of Attacks

Data is usually subject to attacks when there is least security. The intensity and survival of attacks differ depending on the security provided in the network. The following are the attacks that are most common in a network. Most of these attacks can be mitigated by following various approaches like increasing security, using firewalls etc. The following is the description of each attack.

2.1.1.1 Malware

Malware [2] is a malicious stuff that comes along with good stuff when a user attaches his devices to internet. This malware can enter a system through E-mails, web pages etc. Once it enters the system, it can perform many harmful things like deleting the files, installing spyware to detect the keystrokes and extract passwords, credit card details etc. The malware can spread in the form of Virus, Worms or Trojan horse.

2.1.1.2 IP Spoofing

IP Spoofing [2] is common in physical, network and link layers. In IP spoofing attack, the message appears as if it came from a different source. The main purpose of this attack is to conceal the identity of the sender. This kind of attack is widely used in Denial-of-Service attacks. IP address is used as a source of validation to identify whether the user is a legitimate user or not in all the operating systems and networks. Attackers can spoof the IP address and present it as a valid IP and get access to the system. Once the attackers get access to the system they can make any changes to the system like modifying the data or deleting the data which incurs a huge loss. Packet filtering is one of the techniques used to defend against IP Spoofing.

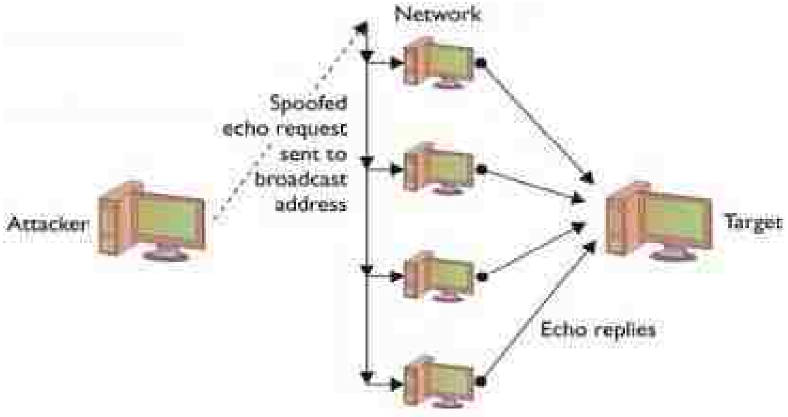


Fig 1: Example to Demonstrate IP Spoofing

2.1.1.3 IP Sniffing

IP Sniffing [2] is common in physical, network and link layers. In this kind of attack, the attacker analyses the network traffic and targets various protocols, services and captures sensitive information like user name, password, e-mails etc. It usually targets low level layers. Wireshark is one of the packet sniffer used to capture packets.

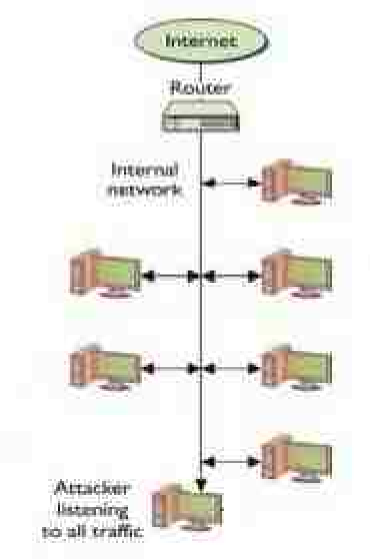


Fig 2: Example to Demonstrate IP Sniffing

2.1.1.4 Password Based Attack

Most of the operating systems are secured using passwords. Depending on the username and password which a user gives, the access rights are assigned to the user. Once if the attacker gets to know the username and password of a valid user, he can create an account for himself and provide all the rights provided to a legitimate user. Now the attacker can use the system as a legitimate user and make many changes to the computer. The attacker can gather the information about the legitimate users, modify the network connections and configurations, modify or delete important files.

2.1.1.5 Man-in-the-middle Attack

In man-in-the-middle attack [2], the attacker monitors, captures and controls the communication without the knowledge of sender or receiver. In lower levels of network layer, the computers may not know with whom they are communicating with. Here, the Man-in-the-middle responds actively to the sender creating an impression that he is the receiver. The attacker can introduce viruses into the system and can alter/modify the data.

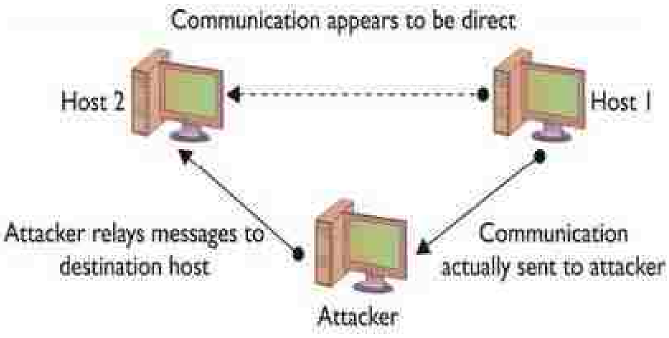

Communication appears to be direct
Host 2
Host 1
Attacker relays messages to destination host
Communication actually sent to attacker
Attacker

Fig 3: Example to Demonstrate Man-in-the-middle Attack

2.1.1.6 Denial of Service Attack

Denial of service attack's [2] main purpose is to degrade an application or a computer system. It can be accomplished in various ways. This can be achieved by depleting various resources like CPU, memory, disk space, network bandwidth etc. Denial of Service can be of many forms. SYN flooding, UDP flooding, ICMP flooding etc. comes under denial of service attacks. Web servers, E-mail servers, DNS servers etc. are subjected to DOS attacks.

According to [2] usually DOS attacks are classified into three types.

Vulnerability Attack: It involves sending messages to a vulnerable application or a system. If enough number of messages is sent, there is high chance of the host to crash or the services to stop.

Bandwidth Flooding: It involves sending a huge number of packets to the targeted host in order to make the target link to get clogged. As a result the legitimate users cannot reach the server.

Connection Flooding: It involves opening a huge number of bogus TCP connections at the target server. With these huge half-open or full-open connections, the host server becomes busy in handling them and as a result it could not accept requests from the legitimate users.

According to [3], Denial of Service is classified into three types namely, DoS (Denial of Service), DDoS (Distributed Denial of Service), DRDoS (Distributed Reflected Denial of Service). Among these types, DRDoS is a very rare attack. In [3], Chen proposed various methods to defend SYN flooding attack. SYN flood attack can be prevented by reducing SYN timeout time, setting SYN cookie. But these methods don’t work efficiently in all the cases. Hence, various preventive measures are explained in this paper. Using a firewall or using a router which carries out preventive NBR, we can prevent SYN flooding attacks.

CHAPTER 3

DDoS ATTACKS IN NETWORK AND TRANSPORT LAYERS

In this chapter, the first section gives introduction to DDoS attacks in various layers; the second section gives insight about various types of DDoS attacks.

3.1 Botnets

A large set of compromised computers that are controlled by attackers for various purposes to carry DDoS attacks are called “Botnets”. Usually these are huge in number and play a very important role in committing DDoS attacks. Normal computers usually get infected by various malwares like virus which spread out through email attachments, various links. These infected computers join Botnets. Botnets have a multi-tier architecture. From the figure, we can observe that the attacker contacts clients and issues the instructions to daemons. As a result, attack is carried out by flooding the victim with too many requests.

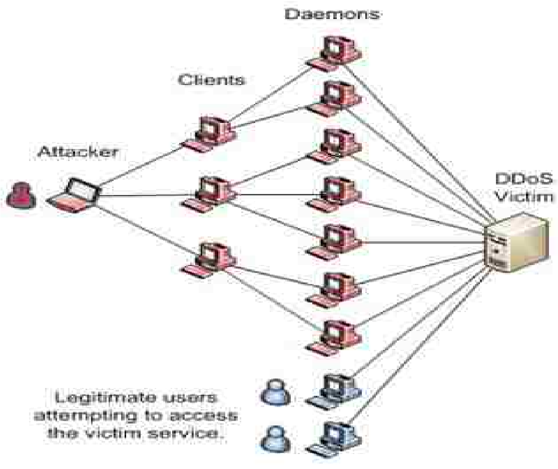


Fig 4: Example of Botnet

3.2 Distributed Denial of Service Attack

It is an attempt to make the resources or services unavailable to the legitimate users by making the system or the server busy with overwhelmed traffic. In DDoS attacks, many computers and many internet connections are used to flood the target with overwhelmed traffic. There are various techniques available to defend against these DDoS attacks and many researches are being conducted.

3.2.1 History of DoS and DDoS Attacks

Initially, in early 1990's DoS attacks started with a single user attacking another user just with a single click of a button. In late 1990's, a set of compromised computers which are controlled by attackers, technically called as "Botnets" were formed. These Botnets resulted in the formation of Distributed Denial of Service attacks. In the year of 2000, the first large scale DDoS attack was committed against various companies like CNN, Yahoo, eBay, Amazon.com etc. Almost all these companies had significant presence in internet. In year 2004, these attacks were used for hire and extortion. Most recently, in years 2007 and 2008, these attacks were widely used against political dissident groups and even against Republic of Georgia during military conflict with Russia.

According to survey conducted by Arbor Networks [4], it has shown that DDoS attacks have been growing rapidly since 2001 and among these, the application layer attacks are on the top. The survey shows that DDoS attacks increased ten times in size from 2005 to 2010. Arbor determined a graph which shows the statistics of increase in application layer DDoS attacks for specific applications.

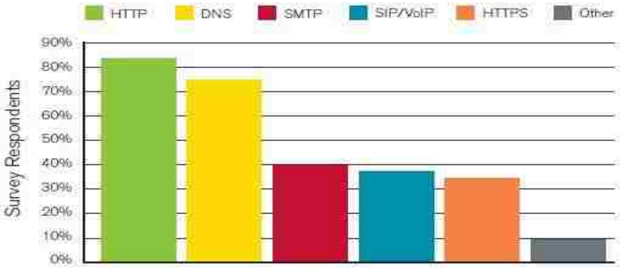


Fig 5: Application Layer DDoS Attacks on Rise

According to a survey conducted by Corero Network Security [5], 38% of U.S enterprises have suffered from DDoS attacks within last one year and 42% of them are victims of multiple attacks. The below figure show the percentage of various organizations that are subjected to risk.

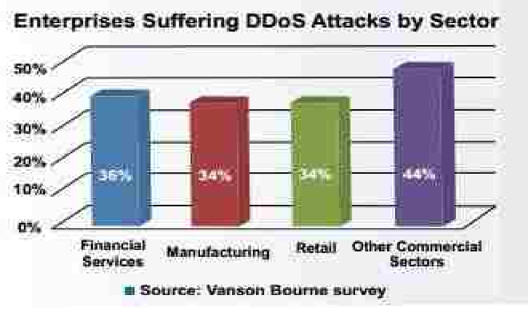


Fig 6: Organizations at Risk

According to [5], the motivation behind DDoS attacks is mainly Political/Ideological, or for the Financial gain, Competitive advantage. The graph below show the percentage of each motive behind DDoS attacks.

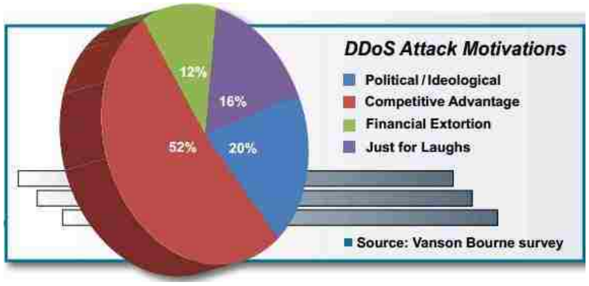


Fig 7: DDoS Attack Motivations

3.1 **DDoS Attacks in Network and Transport Layers**

There are several types of DDoS attacks. Each of them can be committed by using a single attacker or using Botnet.

1. TCP SYN flood
2. Smurf IP
3. UDP flood
4. Ping of death

3.3.1 TCP SYN Flood

This kind of attack affects the hosts running TCP server processes. The main idea of this attack is to make the host retain various unnecessary connections and use all the resources so that the legitimate users do not have enough resources to establish new connections. The attacker keeps sending too many requests to the server and does not respond with an ACK. Thus, makes the server wait for long time keeping the connections open for unnecessary traffic. Many methods have been developed to reduce the effect of SYN flooding. The following figure depicts the TCP SYN flooding attack.

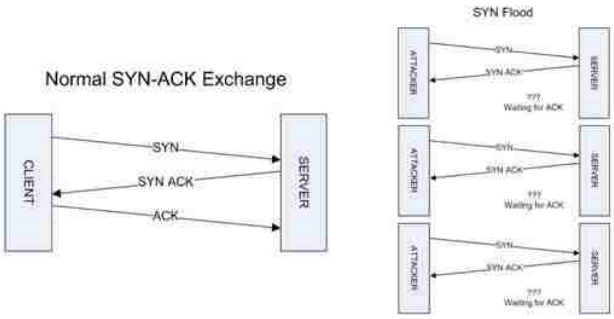


Fig 8: TCP SYN Flood

3.3.2 Smurf IP

In a Smurf IP attack, a ping request is sent by attacker to the broadcast address, modifying the packet to have the victim's IP address as the source. Because the ping was sent to a broadcast address, it will be received by all the machines on the subnet. They read the source IP address, belonging to the victim, and all of them send replies to the victim, overwhelming it with replies. The following figure depicts the Smurf IP attack

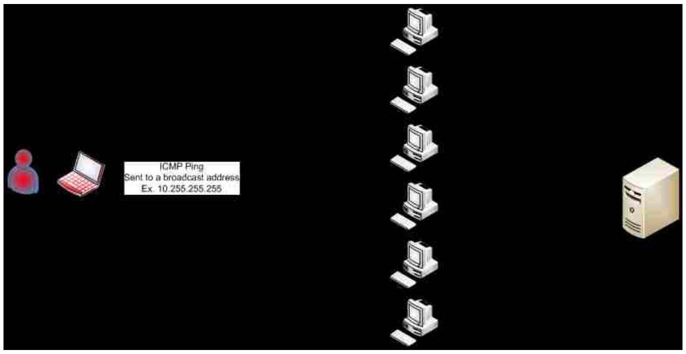


Fig 9: Smurf IP

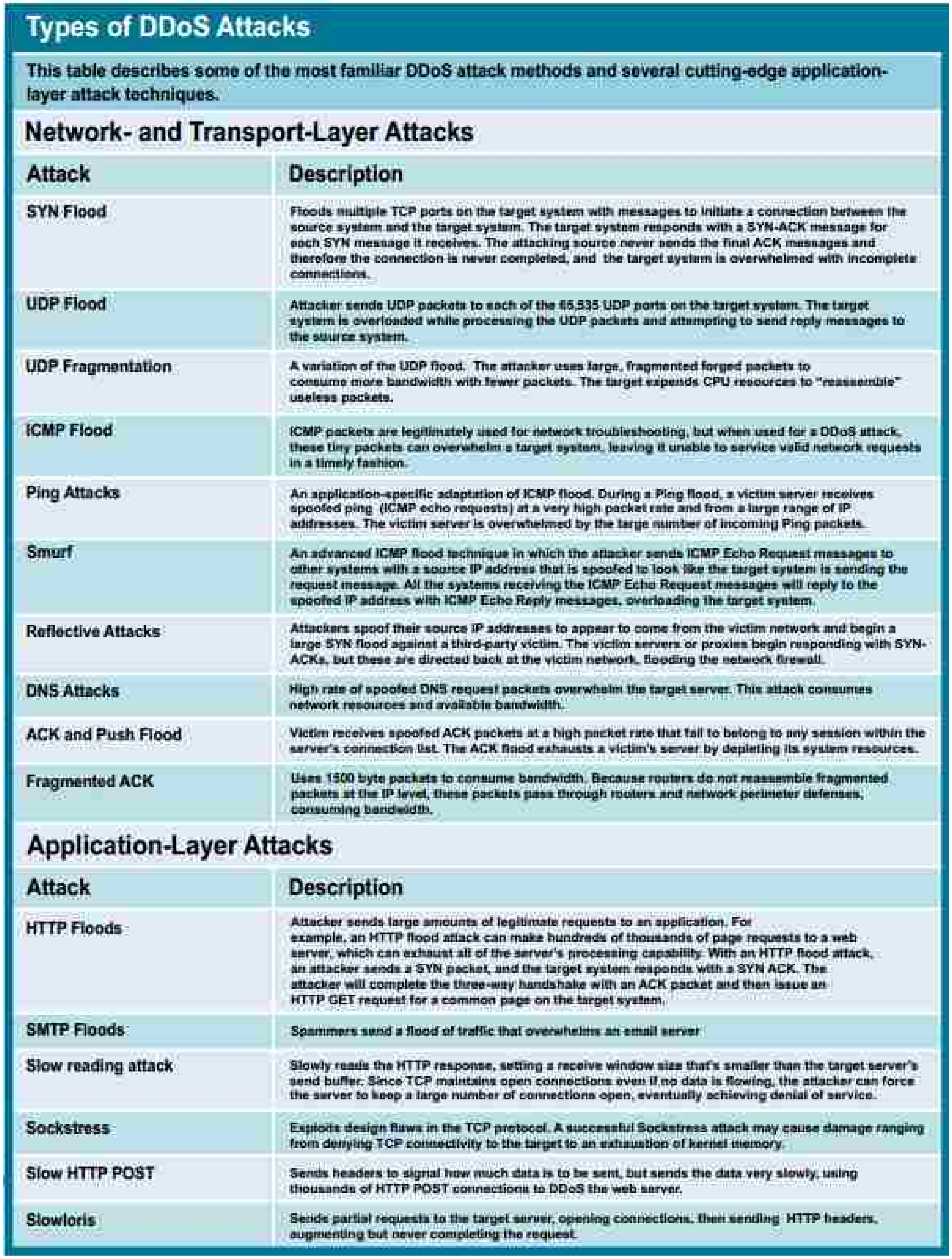
3.3.3 UDP Flood

Its main purpose is to flood a service with huge number of UDP packets. This type of Denial of Service attack fires UDP packets at the victim, attempting to overwhelm a service that is listening for UDP packets. Echo/ Chargen are well known exploits. Chargen is an exploit which generates continuous stream of characters to a network output. Echo is an exploit which reads from the network and “echoes” back what it has read.

3.3.4 Ping of Death

In this kind of attack, the attacker sends larger ping packets/ requests than which is allowed. This results in buffer overflow which leads to system crash. It is very easy to commit this kind of Denial of Service attack. It was very difficult situation in 1990’s. Now, there are various methods to defend against this attack.

In [5], it summarized all these attacks in a table. The following table describes each of the DDoS attacks in brief.



Types of DDoS Attacks

This table describes some of the most familiar DDoS attack methods and several cutting-edge application-layer attack techniques.

Network- and Transport-Layer Attacks

Attack	Description
SYN Flood	Floods multiple TCP ports on the target system with messages to initiate a connection between the source system and the target system. The target system responds with a SYN-ACK message for each SYN message it receives. The attacking source never sends the final ACK messages and therefore the connection is never completed, and the target system is overwhelmed with incomplete connections.
UDP Flood	Attacker sends UDP packets to each of the 65,535 UDP ports on the target system. The target system is overloaded while processing the UDP packets and attempting to send reply messages to the source system.
UDP Fragmentation	A variation of the UDP flood. The attacker uses large, fragmented forged packets to consume more bandwidth with fewer packets. The target expends CPU resources to "reassemble" useless packets.
ICMP Flood	ICMP packets are legitimately used for network troubleshooting, but when used for a DDoS attack, these tiny packets can overwhelm a target system, leaving it unable to service valid network requests in a timely fashion.
Ping Attacks	An application-specific adaptation of ICMP flood. During a Ping flood, a victim server receives spoofed ping (ICMP echo requests) at a very high packet rate and from a large range of IP addresses. The victim server is overwhelmed by the large number of incoming Ping packets.
Smurf	An advanced ICMP flood technique in which the attacker sends ICMP Echo Request messages to other systems with a source IP address that is spoofed to look like the target system is sending the request message. All the systems receiving the ICMP Echo Request messages will reply to the spoofed IP address with ICMP Echo Reply messages, overloading the target system.
Reflective Attacks	Attackers spoof their source IP addresses to appear to come from the victim network and begin a large SYN flood against a third-party victim. The victim servers or proxies begin responding with SYN-ACKs, but these are directed back at the victim network, flooding the network firewall.
DNS Attacks	High rate of spoofed DNS request packets overwhelm the target server. This attack consumes network resources and available bandwidth.
ACK and Push Flood	Victim receives spoofed ACK packets at a high packet rate that fail to belong to any session within the server's connection list. The ACK flood exhausts a victim's server by depleting its system resources.
Fragmented ACK	Uses 1500 byte packets to consume bandwidth. Because routers do not reassemble fragmented packets at the IP level, these packets pass through routers and network perimeter defenses, consuming bandwidth.

Application-Layer Attacks

Attack	Description
HTTP Floods	Attacker sends large amounts of legitimate requests to an application. For example, an HTTP flood attack can make hundreds of thousands of page requests to a web server, which can exhaust all of the server's processing capability. With an HTTP flood attack, an attacker sends a SYN packet, and the target system responds with a SYN ACK. The attacker will complete the three-way handshake with an ACK packet and then issue an HTTP GET request for a common page on the target system.
SMTP Floods	Spammers send a flood of traffic that overwhelms an email server
Slow reading attack	Slowly reads the HTTP response, setting a receive window size that's smaller than the target server's send buffer. Since TCP maintains open connections even if no data is flowing, the attacker can force the server to keep a large number of connections open, eventually achieving denial of service.
Sockstress	Exploits design flaws in the TCP protocol. A successful Sockstress attack may cause damage ranging from denying TCP connectivity to the target to an exhaustion of kernel memory.
Slow HTTP POST	Sends headers to signal how much data is to be sent, but sends the data very slowly, using thousands of HTTP POST connections to DDoS the web server.
Slowloris	Sends partial requests to the target server, opening connections, then sending HTTP headers, augmenting but never completing the request.

Fig 10: Types of DDoS Attacks

CHAPTER 4

DEMONSTRATION OF DDoS ATTACKS ON TCP

This chapter demonstrates an attack committed while transferring files over TCP under DDoS attacks. This experiment shows how a server is attacked using packet flooding. It demonstrates how to capture the packets using wireshark and how to view the details about lost packets. It also demonstrates the setup, tools used to perform the experiment. An image file is transferred from server to client on TCP at a very low bandwidth of 10KBPS.

The following are the requirements to conduct the experiment,

1. Three Computers
2. A Switch
3. Linux Operating System
4. Wireshark

Three machines are setup which act as client, server and attacker. A programmable switch is used in order to reduce the network bandwidth to 10kbps. Experiment is conducted in Linux environment as it has various tools like wireshark, netcat, hping etc. which are used in the experiment.

4.1 Wireshark

Wireshark is an open-source packet analyzer tool which is used to collect packets exchanged over a network and monitor the traffic. It provides various features similar to that of tcpdump; additionally it also provides a graphical interface. It provides various extra features like filtering, sorting etc. These filters are used to refine the data display. Using wireshark, we can view the TCP flow, Time-Sequence graphs, conversation lists etc.

4.2 netcat

Netcat is used to scan various ports, to transfer files, to listen to various ports etc. Using netcat, we can create a client-server message chat communication.

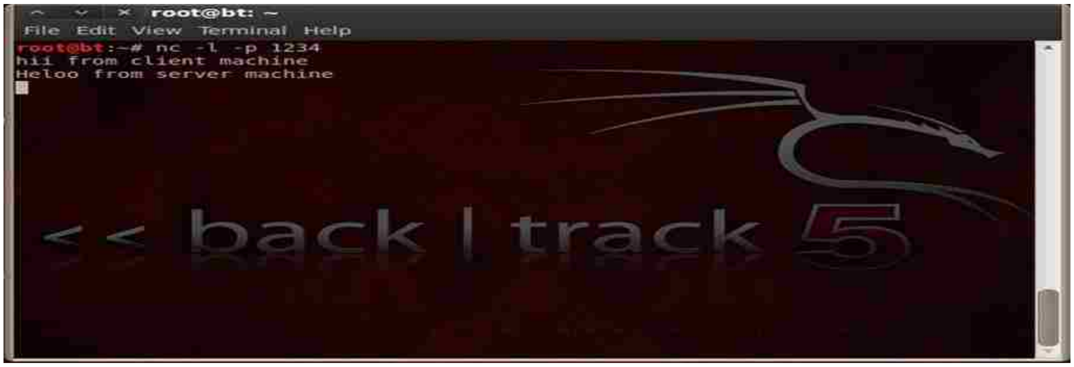
4.3 Creating Client-Server Message Chat Communication

Initially, a message chat communication is created between client machine and server machine in order to check if both the machines are connected and ping each other. The following table explains the order in which the commands are executed at each machine.

S No	Server	S No	Client
1	***nc –l –p 1234***	2	***nc 10.18.22.83 1234***
4	Hello	3	Hi

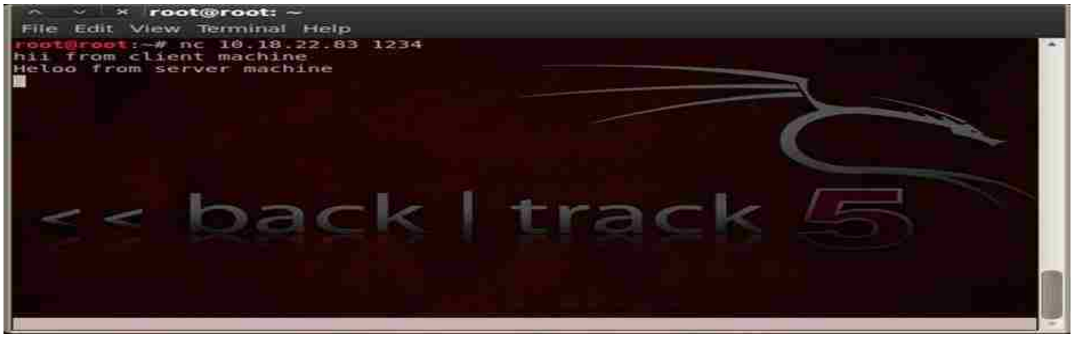
Table1: Client-Server Message Chat Communication

In Table1, *“nc”* represents netcat, *-l –p* tells the machine to listen to a particular port and *1234* is the port number. At client machine, 10.18.22.83 represents IP address of the Server.



```
root@bt:~# nc -l -p 1234
hii from client machine
Heloo from server machine
```

Fig 11: Messages at Server Machine



```
root@root:~# nc 10.18.22.83 1234
hii from client machine
Heloo from server machine
```

Fig 12: Messages at Client Machine

4.4 Transferring an Image File from Server to Client

Initially, an image file named image.jpg is copied to root folder of server machine. Then, terminal is opened in server machine and type the following command.

“*nc –l –p 1234 < image.jpg"*

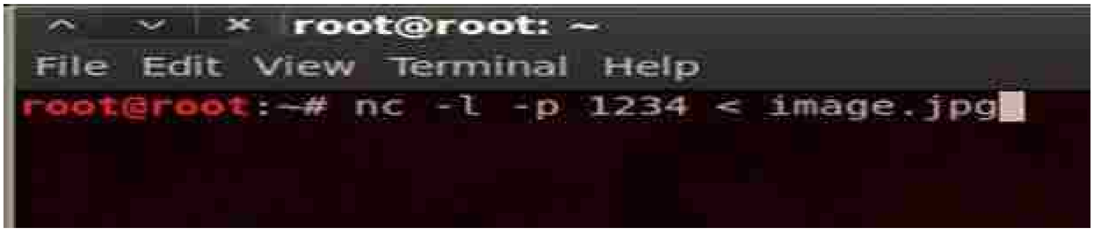
Now, open terminal in client machine and type in the following command.

“nc 10.18.22.105 > image.jpg”

Now, the image.jpg file is copied from server’s root directory to client’s root directory.

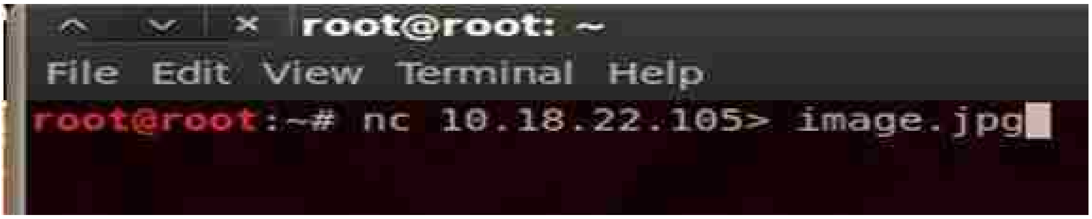
S No	Server	S No	Client
1	Copy an image to root folder		
2	Open terminal	4	Open terminal
3	Type in the following command ***nc –l –p 1234 < image.jpg***	5	***nc 10.18.22.105 > image.jpg***

Table 2: Image Transfer from Server to Client



```
root@root:~# nc -l -p 1234 < image.jpg
```

Fig 13: Command at Server Machine



```
root@root:~# nc 10.18.22.105> image.jpg
```

Fig 14: Command at Client Machine

4.4.1 Capturing the Packet Data

While transferring the image file from server to client, open terminal at both client and server and type “Wireshark” command. It opens the wireshark tool, which is used to capture the packets. When the transfer begins, run the wireshark at both sending and receiving ends. Save them as Server_Capture and Client_Capture. Now the Client_Capture file is on client machine. We need to transfer it to server machine. Follow the same process which we used to transfer image file.

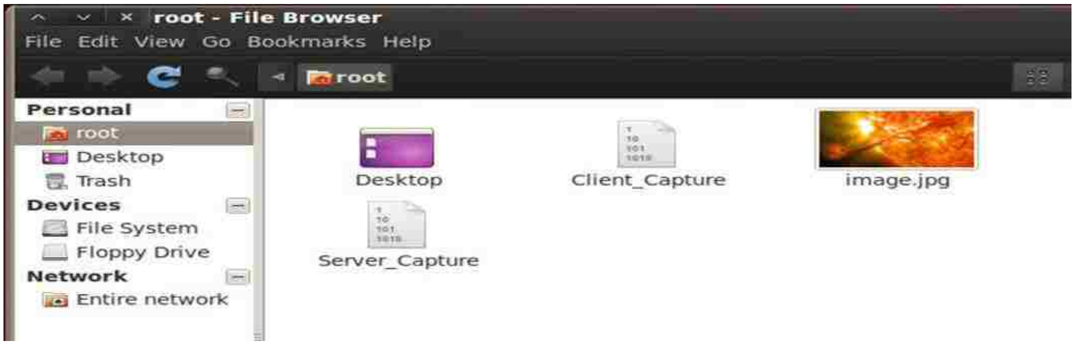
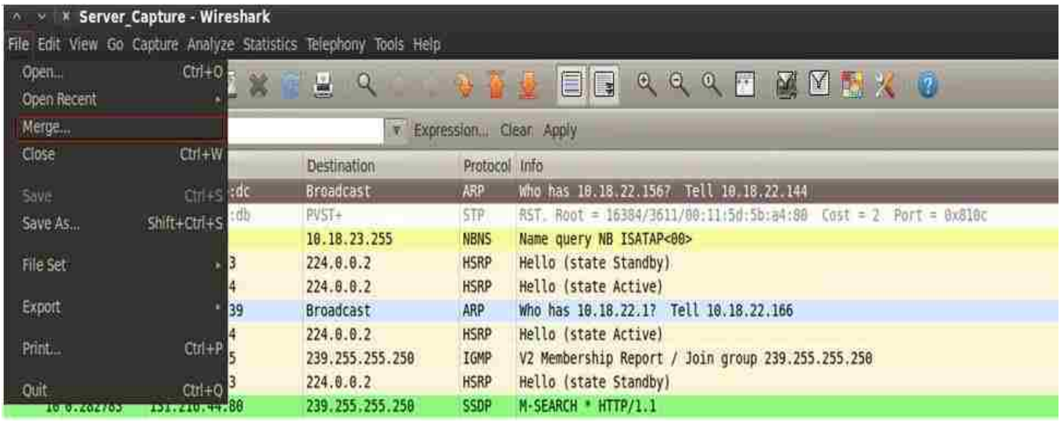


Fig 15: Saving Files at Server’s Root Folder

From the above screenshot, we can see the Client_Capture, image files transferred to/from the server.

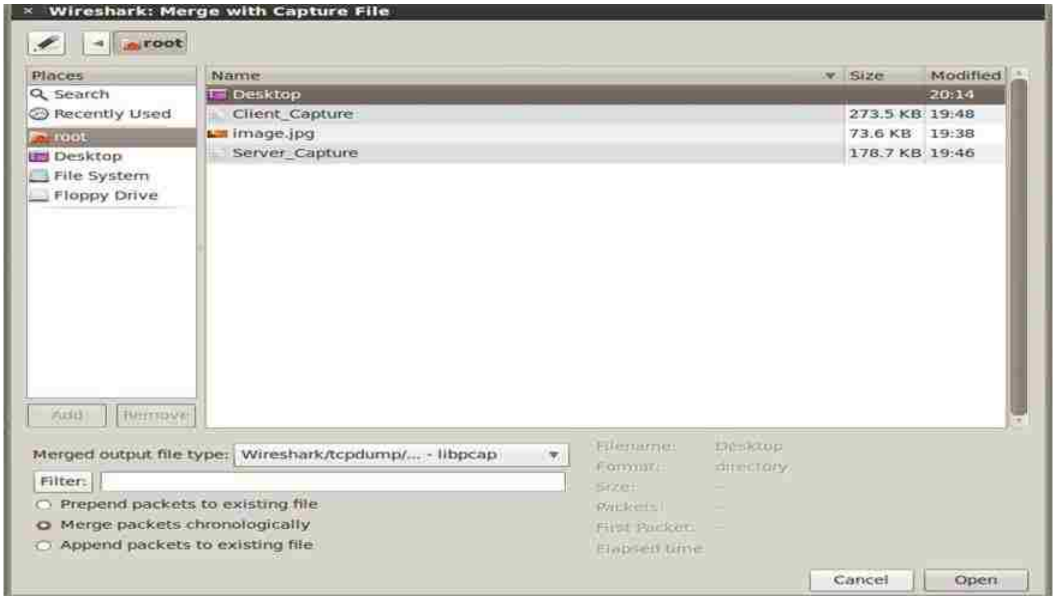
4.4.2 Merging the Packets

Now, the Client_Capture and Server_Capture files are in the root directory of Server machine. We need to merge these files for further comparison. Open Server_Capture in wireshark and from the file menu, select *"Merge"* option. It opens up the open window which allows us to select the file that is to be merged. Then, select the Client_Capture and open it. Now, in the wireshark we have the merged the captured packets of both client and server. Save these merged packets as Merged_Capture in the root folder of server.



	Destination	Protocol	Info
:dc	Broadcast	ARP	Who has 10.18.22.156? Tell 10.18.22.144
:db	PVST+	STP	RST. Root = 16384/3611/00:11:5d:5b:a4:80 Cost = 2 Port = 0x810c
	10.18.23.255	NBNS	Name query NB ISATAP<00>
3	224.0.0.2	HSRP	Hello (state Standby)
4	224.0.0.2	HSRP	Hello (state Active)
39	Broadcast	ARP	Who has 10.18.22.1? Tell 10.18.22.166
4	224.0.0.2	HSRP	Hello (state Active)
5	239.255.255.250	IGMP	V2 Membership Report / Join group 239.255.255.250
3	224.0.0.2	HSRP	Hello (state Standby)
10 0.282785 151.210.44.80	239.255.255.250	SSDP	M-SEARCH * HTTP/1.1

Fig 16: Merge Option in Wireshark



Name	Size	Modified
Desktop		20:14
Client_Capture	273.5 KB	19:48
image.jpg	73.6 KB	19:38
Server_Capture	178.7 KB	19:46

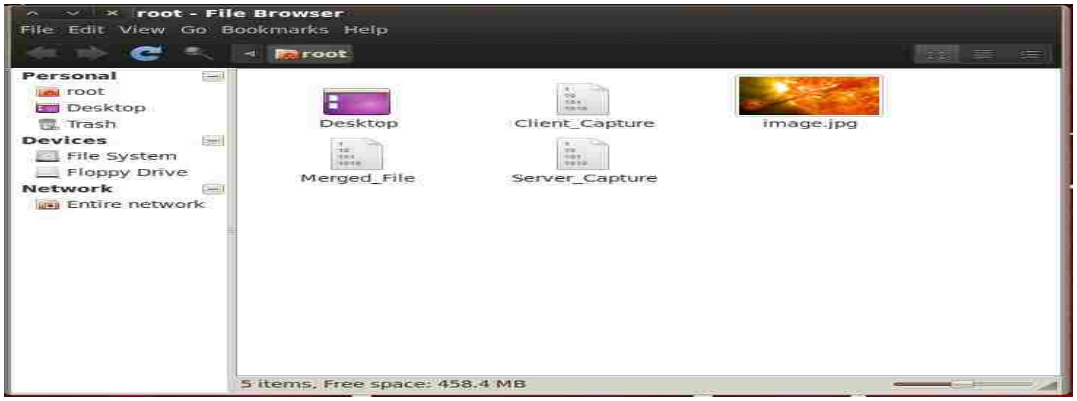
Merged output file type: Wireshark/tcpdump/... - libpcap

Filter:

○ Prepend packets to existing file
● Merge packets chronologically
○ Append packets to existing file

Filename: Desktop
Format: directory
Size: -
Packets: -
First Packet: -
Elapsed time:

Fig 17: Selecting Files to be Merged

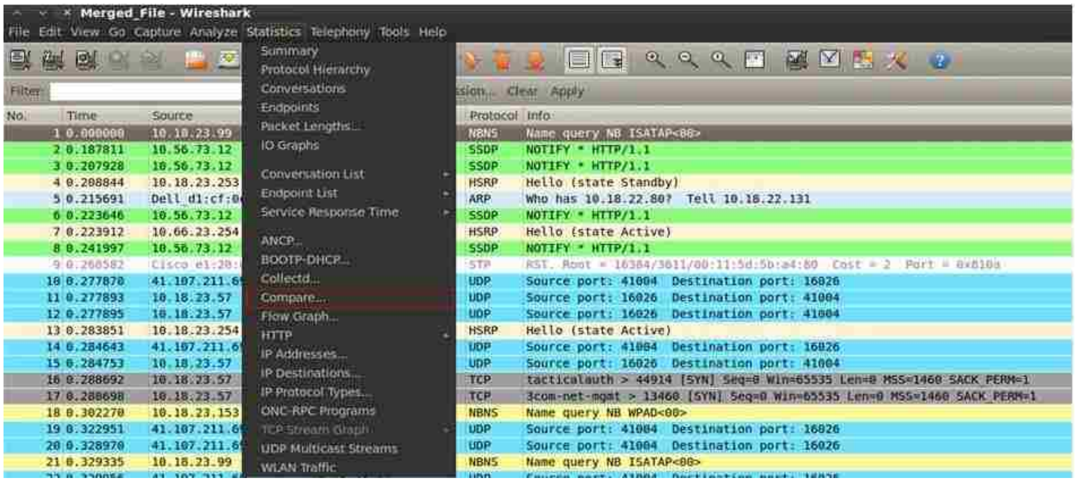


5 items, Free space: 458.4 MB

Fig 18: Saving the Merged File in Root Folder

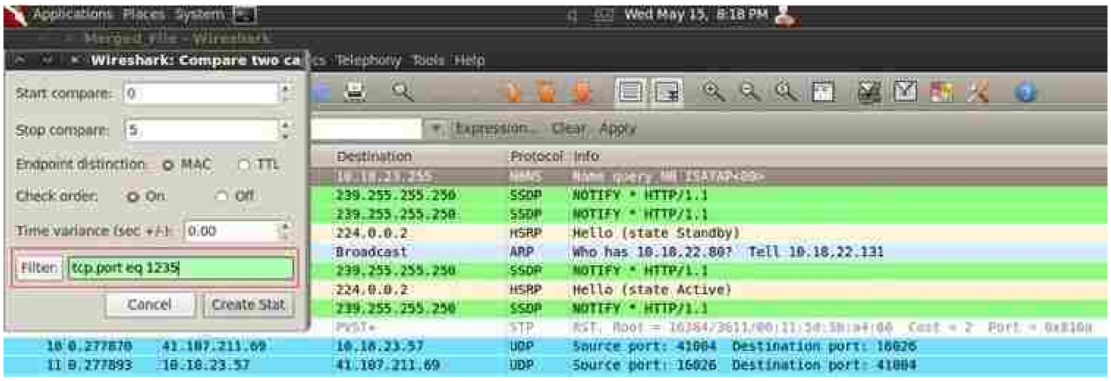
4.4.3 Comparing the Packets

Now, once we get the merged file, we need to do further comparisons. Now, open Merged_File. Select statistics tab and then select compare option. The *"start compare"* and *"stop compare"* values are assigned. Give the filter as "***tcp.port eq 1234***" since the transfer was made on port 1234. When we hit compare, it gives the statistics like number of packets lost and the sequence numbers etc. Using these statistics, we can easily know how many packets were lost or out of order.



```
No.  Time      Source           Protocol Info
   1 0.000000  10.18.23.99      NBNS     Name query NB ISATAP<00>
   2 0.187811  10.56.73.12      SSDP     NOTIFY * HTTP/1.1
   3 0.207928  10.56.73.12      SSDP     NOTIFY * HTTP/1.1
   4 0.208844  10.18.23.253     HSRP     Hello (state Standby)
   5 0.215691  Dell_d1:cf:0     ARP      Who has 10.18.22.80?  Tell 10.18.22.131
   6 0.223646  10.56.73.12      SSDP     NOTIFY * HTTP/1.1
   7 0.223912  10.66.23.254     HSRP     Hello (state Active)
   8 0.241997  10.56.73.12      SSDP     NOTIFY * HTTP/1.1
   9 0.260582  Cisco_e1:20:     STP      RST. Root = 16384/3611/00:11:5d:5b:a4:80  Cost = 2  Port = 0x810a
  10 0.277870  41.107.211.6     UDP      Source port: 41004  Destination port: 16026
  11 0.277893  10.18.23.57      UDP      Source port: 16026  Destination port: 41004
  12 0.277895  10.18.23.57      UDP      Source port: 16026  Destination port: 41004
  13 0.283851  10.18.23.254     HSRP     Hello (state Active)
  14 0.284643  41.107.211.6     UDP      Source port: 41004  Destination port: 16026
  15 0.284753  10.18.23.57      UDP      Source port: 16026  Destination port: 41004
  16 0.288692  10.18.23.57      TCP      tacticalauth > 44914 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
  17 0.288698  10.18.23.57      TCP      3com-net-mgmt > 13460 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
  18 0.302270  10.18.23.153     NBNS     Name query NB WPAD<00>
  19 0.322951  41.107.211.6     UDP      Source port: 41004  Destination port: 16026
  20 0.328970  41.107.211.6     UDP      Source port: 41004  Destination port: 16026
  21 0.329335  10.18.23.99      NBNS     Name query NB ISATAP<00>
```

Fig 19: Selecting Compare Option


Applications Places System
Wed May 15, 8:18 PM
Merged_File - Wireshark
Wireshark: Compare two ca
...cs Telephony Tools Help
Expression... Clear Apply
Cancel
Create Stat

Start compare: 0

Stop compare: 5

Endpoint distinction: ● MAC ○ TTL

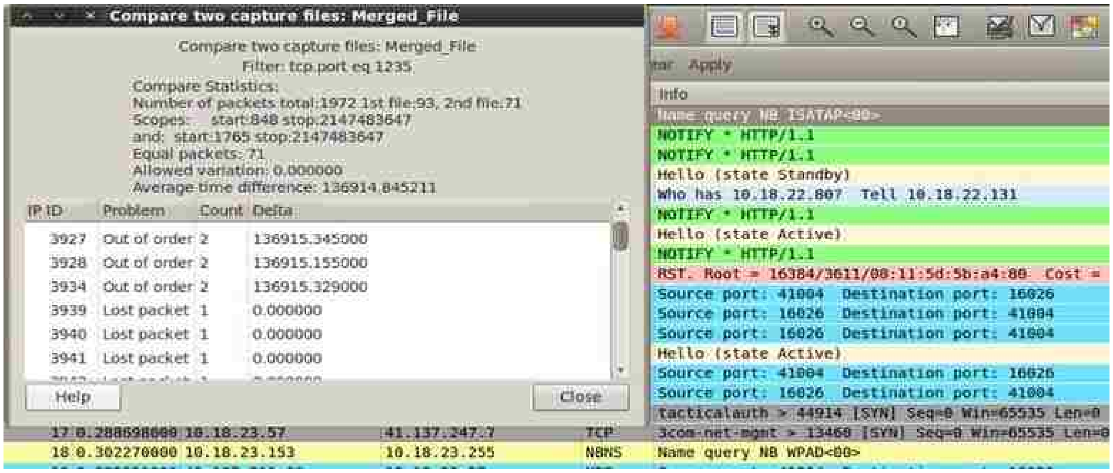
Check order: ● On ○ Off

Time variance (sec +/-): 0.00

Filter: tcp.port eq 1235

```
                                Destination      Protocol Info
                                10.18.23.255     NBNS     Name query NB ISATAP<00>
                                239.255.255.250  SSDP     NOTIFY * HTTP/1.1
                                239.255.255.250  SSDP     NOTIFY * HTTP/1.1
                                224.0.0.2        HSRP     Hello (state Standby)
                                Broadcast        ARP      Who has 10.18.22.80?  Tell 10.18.22.131
                                239.255.255.250  SSDP     NOTIFY * HTTP/1.1
                                224.0.0.2        HSRP     Hello (state Active)
                                239.255.255.250  SSDP     NOTIFY * HTTP/1.1
                                PVST+            STP      RST. Root = 16384/3611/00:11:5d:5b:a4:80  Cost = 2  Port = 0x810a
  10 0.277870  41.107.211.69    10.18.23.57      UDP      Source port: 41004  Destination port: 16026
  11 0.277893  10.18.23.57      41.107.211.69    UDP      Source port: 16026  Destination port: 41004
```

Fig 20: Comparing Merged Files



```
Compare two capture files: Merged_File
Filter: tcp.port eq 1235
Compare Statistics:
Number of packets total:1972 1st file:93, 2nd file:71
Scopes:    start:848 stop:2147483647
and:  start:1765 stop:2147483647
Equal packets: 71
Allowed variation: 0.000000
Average time difference: 136914.845211

IP ID  Problem       Count  Delta
 3927  Out of order  2      136915.345000
 3928  Out of order  2      136915.155000
 3934  Out of order  2      136915.329000
 3939  Lost packet   1      0.000000
 3940  Lost packet   1      0.000000
 3941  Lost packet   1      0.000000
```

```
Info
Name query NB ISATAP<00>
NOTIFY * HTTP/1.1
NOTIFY * HTTP/1.1
Hello (state Standby)
Who has 10.18.22.80?  Tell 10.18.22.131
NOTIFY * HTTP/1.1
Hello (state Active)
NOTIFY * HTTP/1.1
RST. Root = 16384/3611/00:11:5d:5b:a4:80  Cost =
Source port: 41004  Destination port: 16026
Source port: 16026  Destination port: 41004
Source port: 16026  Destination port: 41004
Hello (state Active)
Source port: 41004  Destination port: 16026
Source port: 16026  Destination port: 41004
tacticalauth > 44914 [SYN] Seq=0 Win=65535 Len=0
3com-net-mgmt > 13460 [SYN] Seq=0 Win=65535 Len=0
Name query NB WPAD<00>

17 0.288698000 10.18.23.57    41.137.247.7   TCP
18 0.302270000 10.18.23.153   10.18.23.255   NBNS
```

Fig 21: Comparison Results

4.5 Outline of the Experiment

Setup three machines which acts as client, server and attacker and runs on LINUX environment. All these three machines are connected to the network through a switch. The bandwidth of the network is configured to 10KBPS. Initially the client machine requests a file from server machine. In this experiment, an image file of size 2MB is considered. When the file is requested, the request goes through the switch and reaches the server. In the meanwhile, the attacker floods huge amount of packets to the server using hping3 attack command. This attack makes the server overwhelm with lots of packets. Due to huge packet flooding, packets are lost at the switch. When a large file is being transferred from machine to another at reduced bandwidth under DDoS attack, due to heavy traffic we can observe packet loss. This results in denial of service from the server to a legitimate user.

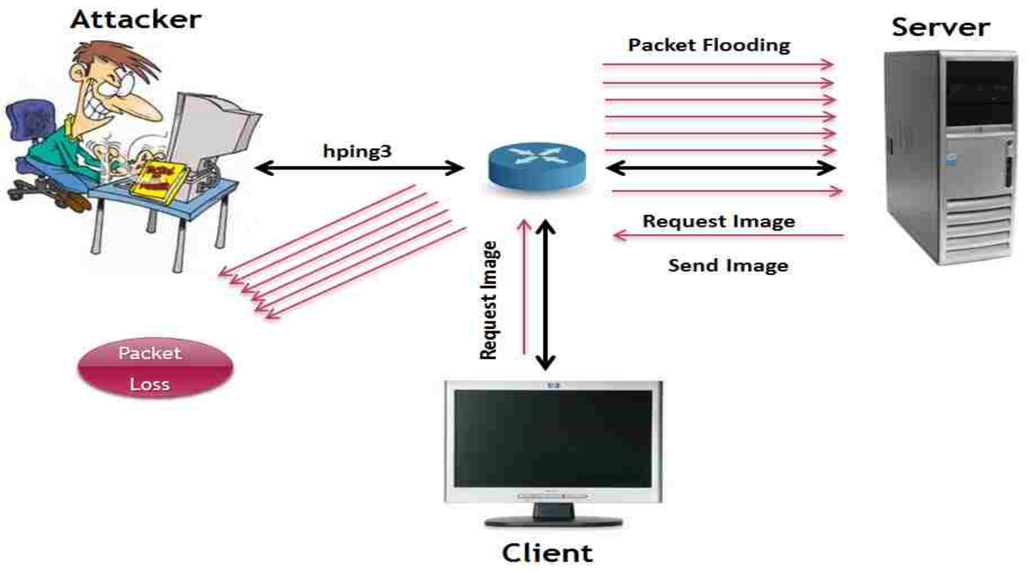
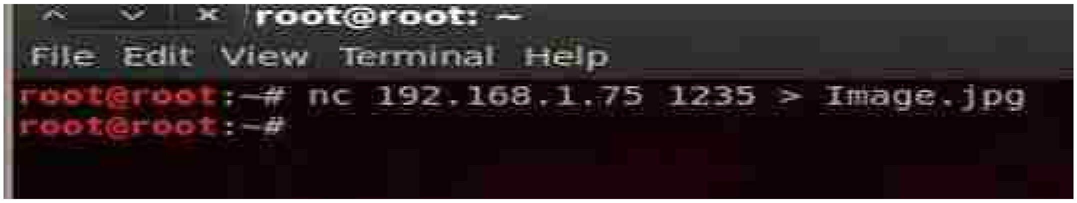


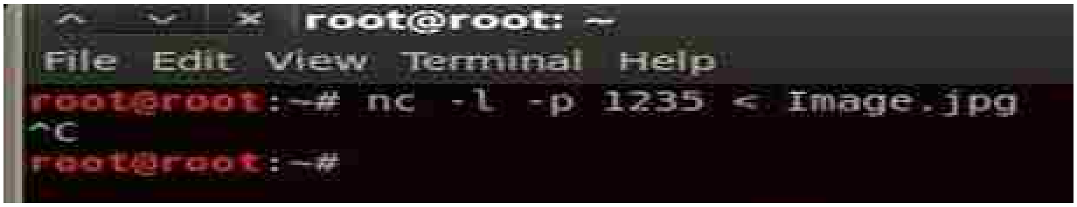
Fig 22: DDoS Attack on Server while Transferring an Image File using TCP

Initially, server is having the image file which is to be transferred to the client machine. Below are the screenshots which describes the entire process.



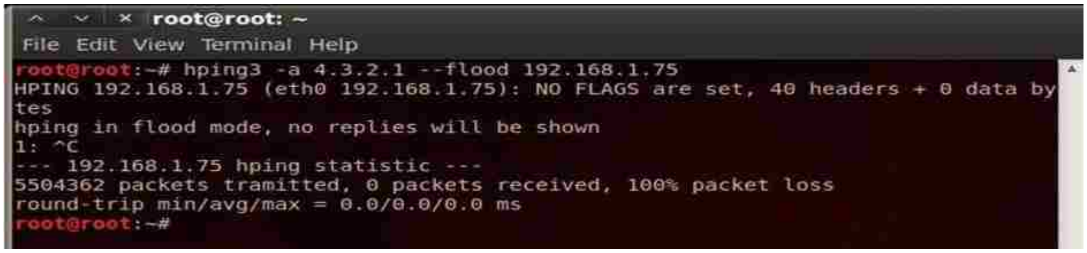
```
root@root:~# nc 192.168.1.75 1235 > Image.jpg
root@root:~#
```

Fig 23: Command at Client’s Machine



```
root@root:~# nc -l -p 1235 < Image.jpg
^C
root@root:~#
```

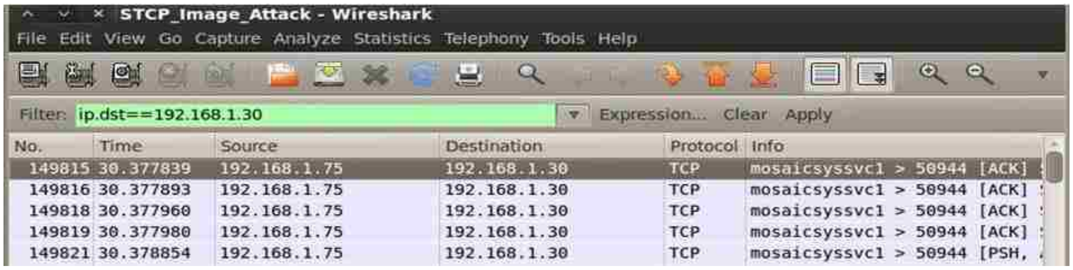
Fig 24: Command at Server’s Machine



```
root@root:~# hping3 -a 4.3.2.1 --flood 192.168.1.75
HPING 192.168.1.75 (eth0 192.168.1.75): NO FLAGS are set, 40 headers + 0 data by
tes
hping in flood mode, no replies will be shown
1: ^C
--- 192.168.1.75 hping statistic ---
5504362 packets tramitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@root:~#
```

Fig 25: Command at Attacker’s Machine

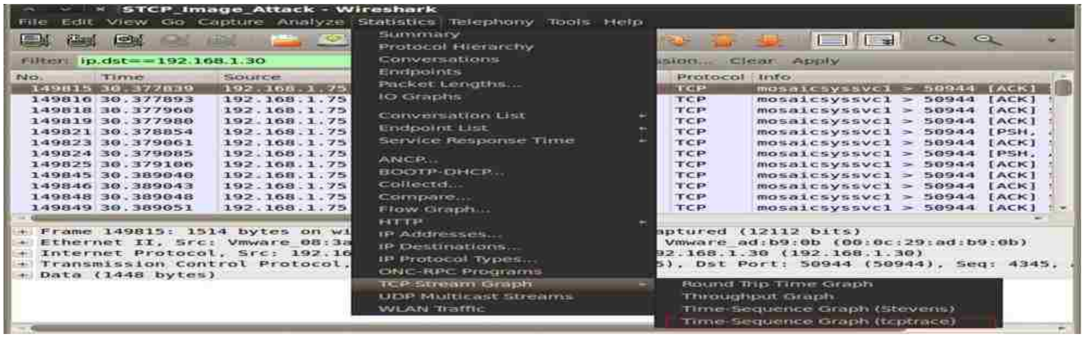
Before executing these commands, open the wireshark application on client, server and attacker machines to capture the packets. Below are the screenshots of captured packets.


STCP_Image_Attack - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter: ip.dst==192.168.1.30
Expression... Clear Apply

```
No.     Time       Source        Destination   Protocol Info
149815 30.377839 192.168.1.75  192.168.1.30  TCP      mosaicsyssvc1 > 50944 [ACK]
149816 30.377893 192.168.1.75  192.168.1.30  TCP      mosaicsyssvc1 > 50944 [ACK]
149818 30.377960 192.168.1.75  192.168.1.30  TCP      mosaicsyssvc1 > 50944 [ACK]
149819 30.377980 192.168.1.75  192.168.1.30  TCP      mosaicsyssvc1 > 50944 [ACK]
149821 30.378854 192.168.1.75  192.168.1.30  TCP      mosaicsyssvc1 > 50944 [PSH,
```

Fig 26: Packets captured at Server's Machine

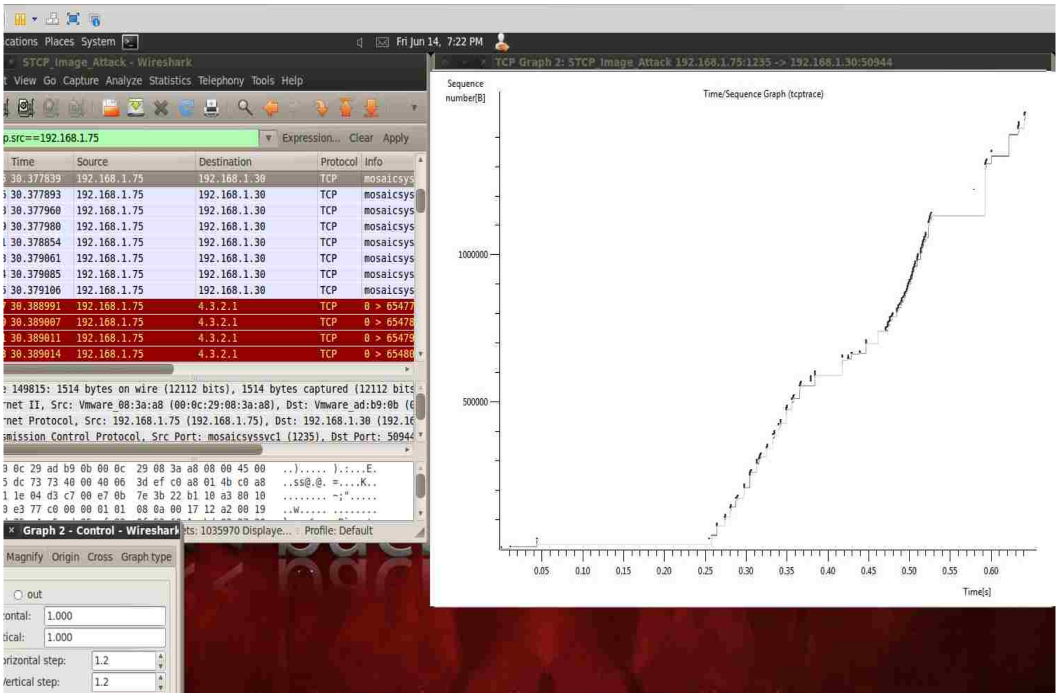
Now, using the previous method, merge both the server and client files.



```
No.     Time       Source          Protocol Info
149815 30.377839 192.168.1.75    TCP      mosaicsyssvc1 > 50944 [ACK]
149816 30.377893 192.168.1.75    TCP      mosaicsyssvc1 > 50944 [ACK]
149818 30.377960 192.168.1.75    TCP      mosaicsyssvc1 > 50944 [ACK]
149819 30.377980 192.168.1.75    TCP      mosaicsyssvc1 > 50944 [PSH,
149821 30.378854 192.168.1.75    TCP      mosaicsyssvc1 > 50944 [ACK]
149823 30.379061 192.168.1.75    TCP      mosaicsyssvc1 > 50944 [PSH,
149824 30.379085 192.168.1.75    TCP      mosaicsyssvc1 > 50944 [ACK]
149825 30.379106 192.168.1.75    TCP      mosaicsyssvc1 > 50944 [ACK]
149845 30.389040 192.168.1.75    TCP      mosaicsyssvc1 > 50944 [ACK]
149846 30.389043 192.168.1.75    TCP      mosaicsyssvc1 > 50944 [ACK]
149848 30.389048 192.168.1.75    TCP      mosaicsyssvc1 > 50944 [ACK]
149849 30.389051 192.168.1.75    TCP      mosaicsyssvc1 > 50944 [ACK]

+ Frame 149815: 1514 bytes on wi                 aptured (12112 bits)
+ Ethernet II, Src: Vmware_08:3a                  Vmware_ad:b9:0b (00:0c:29:ad:b9:0b)
+ Internet Protocol, Src: 192.16                 92.168.1.30 (192.168.1.30)
+ Transmission Control Protocol,                 5), Dst Port: 50944 (50944), Seq: 4345,
+ Data (1448 bytes)
```

Fig 27: Time-Sequence Graph

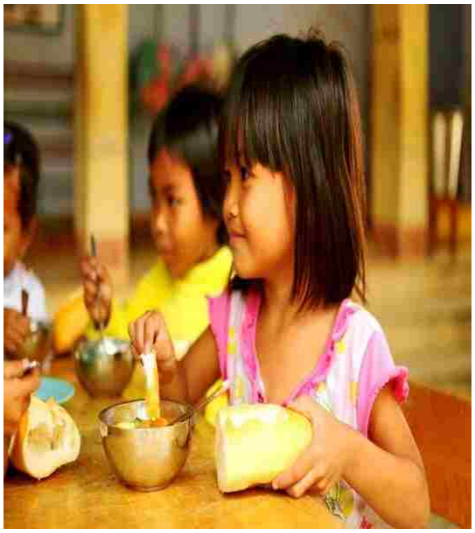
Once the files are merged, we can observe various things like how many packets are lost, how many packets were sent out of order, how long it took for the file transfer etc. Using TCP traces, we can observe the TCP slow start. Due to large amount of packet flooding from the attacker, the packets are lost while transferring from server to client. Thus, the client cannot receive the file completely. Thus, this chapter gives an insight about how an attack is committed using hping3 tool and how the server denies processing the requests of the client. This entire experiment is conducted in Linux environment.



cations Places System
Fri Jun 14, 7:22 PM
STCP_Image_Attack - Wireshark
t View Go Capture Analyze Statistics Telephony Tools Help
Expression... Clear Apply
TCP Graph 2: STCP_Image_Attack 192.168.1.75:1235 -> 192.168.1.30:50944
Sequence number[B]
Time/Sequence Graph (tcptrace)
1000000
500000
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45
0.50
0.55
0.60
Time[s]
Graph 2 - Control - Wireshark
Magnify Origin Cross Graph type
out

```
p.src==192.168.1.75

  Time        Source         Destination    Protocol Info
5 30.377839   192.168.1.75   192.168.1.30   TCP      mosaicsys
5 30.377893   192.168.1.75   192.168.1.30   TCP      mosaicsys
3 30.377960   192.168.1.75   192.168.1.30   TCP      mosaicsys
9 30.377980   192.168.1.75   192.168.1.30   TCP      mosaicsys
1 30.378854   192.168.1.75   192.168.1.30   TCP      mosaicsys
3 30.379061   192.168.1.75   192.168.1.30   TCP      mosaicsys
4 30.379085   192.168.1.75   192.168.1.30   TCP      mosaicsys
5 30.379106   192.168.1.75   192.168.1.30   TCP      mosaicsys
7 30.388991   192.168.1.75   4.3.2.1        TCP      0 > 65477
9 30.389007   192.168.1.75   4.3.2.1        TCP      0 > 65478
1 30.389011   192.168.1.75   4.3.2.1        TCP      0 > 65479
3 30.389014   192.168.1.75   4.3.2.1        TCP      0 > 65480

e 149815: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits
rnet II, Src: Vmware_08:3a:a8 (00:0c:29:08:3a:a8), Dst: Vmware_ad:b9:0b (0
rnet Protocol, Src: 192.168.1.75 (192.168.1.75), Dst: 192.168.1.30 (192.16
smission Control Protocol, Src Port: mosaicsyssvc1 (1235), Dst Port: 50944

0 0c 29 ad b9 0b 00 0c  29 08 3a a8 08 00 45 00   ..)..... ).:...E.
5 dc 73 73 40 00 40 06  3d ef c0 a8 01 4b c0 a8   ..ss@.@. =....K..
1 1e 04 d3 c7 00 e7 0b  7e 3b 22 b1 10 a3 80 10   ........ ~;".....
0 e3 77 c0 00 00 01 01  08 0a 00 17 12 a2 00 19   ..w..... ........

ets: 1035970 Displaye...   Profile: Default

ontal:            1.000
tical:            1.000
rizontal step:    1.2
/ertical step:    1.2
```

Fig 28: TCP Slow Start



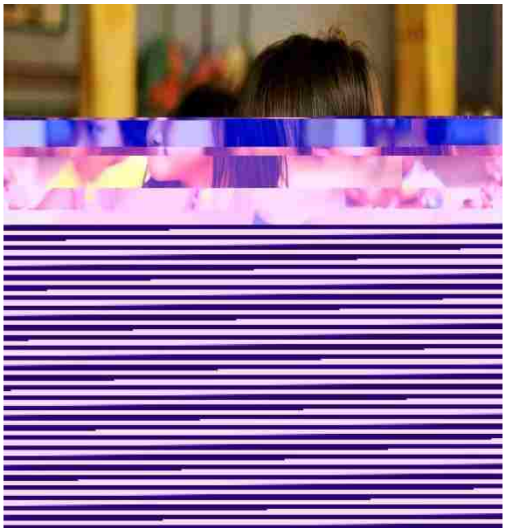


Fig 29: Image Sent

Fig 30: Image Received

CHAPTER 5

DDoS ATTACKS IN APPLICATION LAYER

According to Arbor networks [4], application layer DDoS attacks are classified into four types. The below is the description of each attack.

5.1 Types of Attacks in Application Layer

5.1.1 Request Flooding Attacks

In this kind of attack, the attacker sends huge number of legitimate requests to the server and overwhelms the session resources of the server.

5.1.2 Asymmetric Attacks

In this kind of attack, the attacker sends requests at normal rate which has high work load. The goal of this attack is to consume resources like CPU, memory of the server and degrade it.

5.1.3 Repeated One Shot Attacks

These kinds of attacks are stealthier when compared to the request flooding and asymmetric attacks. But the goal of this attack is the same, to degrade the server. In this attack, high workload request are sent over multiple TCP sessions.

5.1.4 Application - Exploit Attacks

These attacks targets the applications vulnerabilities and thus gaining control of application and network. Examples of these kinds of attacks include, buffer overflows, cookie poisoning, SQL injection etc.

The next section discusses about various approaches proposed by various researchers to detect and defend against DDoS attacks at Application layer.

5.2 Approaches for Application Layer DDoS Attack Defense and Detection

This section gives an insight about various detection and defense mechanisms proposed by various researchers. Each approach is explained in brief. It covers the mechanism followed by each approach, advantages and disadvantages of each approach etc.

5.2.1 A Novel Framework to Detect and Block DDoS Attack at Application Layer

[6] Introduced new algorithms that are capable to detecting and blocking various DDoS attacks which allows the legitimate users including flash crowds. Its main goal is to design algorithms at application layer that detects the attack traffic and allows legitimate traffic to receive web services. It implements user signature calibration using CAPTCHA or AYAH.

5.2.1.1 CAPTCHA

Use of CAPTCHA (Completely automated public Turing test to tell computers and humans apart) to detect DDoS includes Kandula et al [7] and Boyd et al [8] which is implemented as a puzzle authentication mechanism. A signature is generated for each user that determines whether a user is suspicious or not. According to David Pogue [9], CAPTCHA really stands for “Computer annoying people with time-wasting challenges”.

5.2.1.2 AYAH

It is similar to that of CAPTCHA. It allows dynamic determination of whether a signature really represents an attack or non-human user like robots or a legitimate human user. AYAH is implemented on a tiny fraction of traffic.

5.2.1.3 System Model

It contains signatures and web requests. Each user makes a web request and is named as USER 1, USER 2 so on. Each user’s web request is assigned a signature by signature generator. Once the signature is generated, signature database is updated. A threshold value is set for the server load. This model considered two thresholds as Low Load Threshold (LLT) and High Load Threshold (HLT). If the threshold value is above LLT, then suspicious users are detected and delayed. If threshold value is above HLT then the suspicious users are blocked. It detects the suspicious users based on blocking methods like AYAH and existing signature detection. In this system AYAH page is implemented on very small amount of traffic.

5.2.1.4 Advantages

This model differentiates flash crowd from attack traffic.

5.2.1.5 Disadvantages

Use of AYAH occasionally causes some delay and it is implemented on very small amount of traffic.

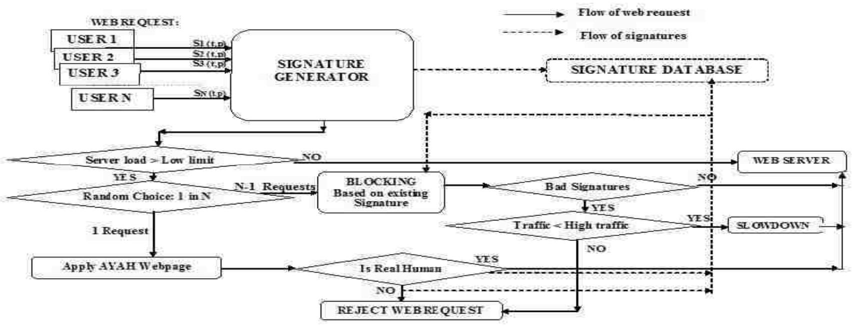


Fig 31: System Model

5.2.2 IP Trace Back System for Network and Application Layer Attacks

IP Trace back System [10] detects both network layer and application layer attacks. This system considered HTTP Flood attack and worms where attackers evade detection by posing as legitimate clients. This method also employs SNORT during the creation of normal profiles [11].

This paper proposed a *hybrid technique*, which detects an attack and generates an alert file and sends it to IP address reconstruction module. In this module, the IP Address of ingress router of the attacker can be reconstructed.

5.2.2.1 Attack Detection

Initially, packet headers are analyzed by generating histograms and various behaviors are saved as baselines. Later, the payload information is analyzed. The online traffic payload is compared with header and statistical models are developed which are used to determine the deviation. More the deviation, more anomalous the payload is. Under feature selection and histogram creation various features can be captured from the traffic and can be used for detecting the anomalies. Features like IP address are used to detect DDoS flooding attacks. For non-flooding attacks, payload is processed to extract the model. MAHALANOBIS distance is used to classify the non-flooding application layer attacks. Higher the distance, greater is the chance of payload to be abnormal.

5.2.2.2 Hybrid IP Trace Back

Packet marking reduces the overload of the router. It consists of three components,

First one is Packet marking. Router’s IP address is fragmented into four parts and marked. In order to avoid errors while grouping the fragments, checksum is used.

Second is IP Address Reconstruction. Once the malicious packets are detected, then reconstruction is done to detect the ingress router. Address identification and Address recovery are the two phases involved in IP address reconstruction. Third is attacker's source identification using the entropy. Entropy variation is calculated for certain amount of traffic in particular time interval. Each router has various interfaces. The interface with large deviation is considered as suspicious and added to the list. This suspicious list is referred to track back the suspected host.

5.2.2.3 Advantages

1. Detection system detects both flooding and non-flooding bad payload attacks.
2. Checksum is used instead of hash function calculations, reduces time and byte consumption of IP header fields.
3. The interface from which the attacker enters the network is found.
4. Proactive traffic shaping pushes flooding packets to lower priority queue even before detecting the attack.
5. Medium number of false positives.
6. Proactive shaping will allocate lesser bandwidth to suspicious flows.

5.2.2.4 Disadvantages

It has the problem of false positives.

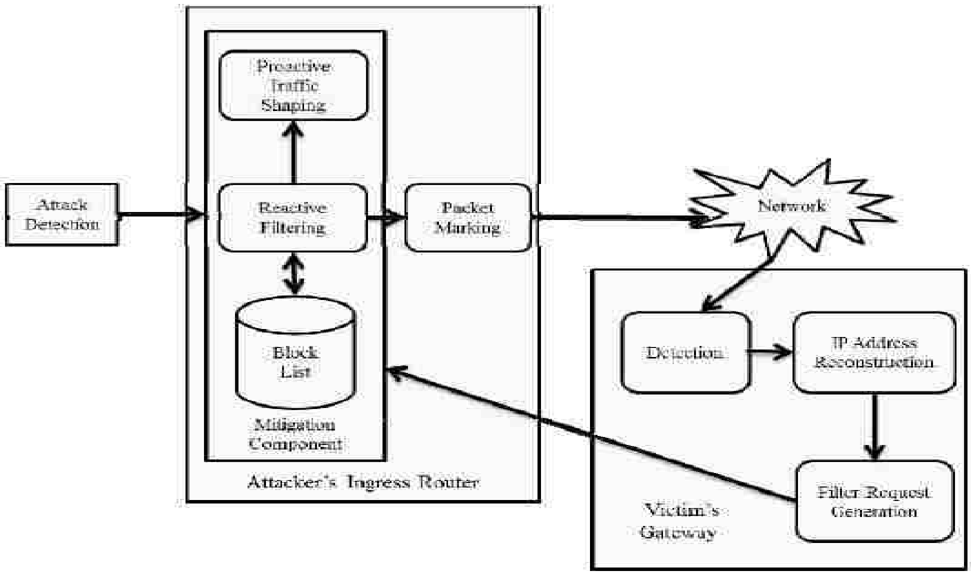


Fig 32: Overview of Proposed IP Trace Back System

5.2.3 Application Layer DDoS Detection using Clustering Analysis

[12] Introduced a clustering method to analyze application layer DDoS attacks. User's sessions are clustered to capture the browsing behavior. Various features like Session, Request rate, Average Popularity, Average transition probability are extracted to cluster user sessions.

[12] Uses Cluster analysis method to analyze browsing behavior of user and to detect application layer DDoS attacks. Its main goal is to detect application layer DDoS attacks.

5.2.3.1 Proposed Method

Initially, the user sessions are clustered. To detect the application layer DDoS attacks, deviation between sessions and normal clusters need to be calculated.

5.2.3.2 Clustering Analysis

Using the features extracted above, user sessions are clustered and these clusters are used to group browsing behaviors. When there is a DDoS attack, the attack sessions can be separated from normal ones. There are various methods to implement clustering. This model uses hierarchical clustering method to cluster the sessions. Finally, the number of clusters has to be determined. This model used Hierarchical clustering method [13].

5.2.3.3 Summary

A clustering model is proposed to determine the web user browsing behavior. Based on this behavior, a counter mechanism to detect application layer attacks is built. Simulated the attack for number of times and results prove that this model is efficient and effective.

5.2.3.4 Advantages

This model uses various features to calculate the browsing behavior. It finds out the number of sessions, anomalies, detection rate.

5.2.3.5 Disadvantages

This model cannot distinguish attack traffic from flash crowds.

5.2.4 An Effective Approach to Counter Application Layer DDoS Attacks

[14] Proposed a scheme to defend against DDoS attacks in application layer and schedule the flash crowd during these attacks.

An access matrix is used to capture access patterns of legitimate clients and normal flash crowd. Its main goal is to drop the suspicious traffic and to provide services to legitimate users.

5.2.4.1 Proposed Work

This method is based on the behavior of the web user. It uses access matrix to capture the access patterns of the legitimate users as well as the flash crowd. Various parameters like HTTP request rate, HTTP session rate, Server documents, access duration are store in access matrix. DDoS counter mechanism examines the request; parse this request URL to identify the request type. It maintains the work-load and arrival-history of these requests. This counter mechanism uses suspicion assignment and scheduler. Suspicion mechanism assigns score to each client. If the deviation is more, then it is considered to be suspicious. Scheduler decides whether to forward session requests or not.

5.2.4.2 Detection Principle

It has three steps namely, Data collection, data abstraction and detection.

5.2.4.3 Summary

Using the system log, compute an access matrix. This access matrix is decomposed into singular value. Now, each independent component is analyzed. For each element, suspicions score is assigned and based on the score the suspicious attacks are detected. Then normal flows are scheduled.

5.2.4.4 Advantages

This model detects DDoS attacks during normal flow as well as during flash crowds. Schedules traffic even on attack based on the system workload and scheduling policy.

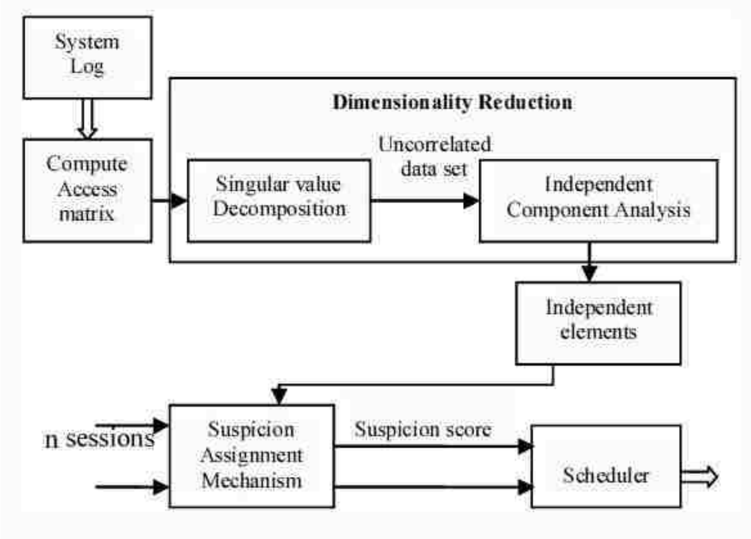


Fig 33: System Architecture

5.2.5 Detection of Application Layer Distributed Denial of Service

A simple and effective approach is introduced to detect application layer DDoS attacks. [15] proposed an http request transition matrix in order to describe users browsing behavior. This paper considered a scenario where a bot keeps sending requests to the web server which have a very small transition probability. Using the likelihood interval, the bots can be easily recognized. Its main goal is to differentiate between humans and botns request sequences even when the attack occurs in low volume or at low rate.

[15] involves four steps namely, Data preprocessing, Threshold, Generating DDoS traces, Detecting DDoS. Each of them are explained below.

5.2.5.1 Dataset Preprocessing

Datasets are required to train an algorithm. Generated datasets from Internet Traffic archives sponsored by ACM SIGCOMM. The dataset contains various information like, the host making the request, hostname, IP address, date, HTTP reply code, bytes in reply, document_ID etc.

5.2.5.2 Threshold

Three parameters have to be determined from the dataset collected. First is, Frequency vector which defines the “popularity of all the objects”. Second is, transition probability matrix which defines the transition probability from one page to another. Third is host request sequence probability which gives the average probability of transition probability of the request sequence. Later, run a detecting algorithm for a particular interval called sampling rate.

5.2.5.3 Generate DDoS Traces

An attack which establishes large number of open connectionsand utilize the disk space is used in this experiment. This kind of attack is detected through this experiment. 100 DDoS attacker hosts are injected to generate the attack. DDoS attack is simulated using “DDosim tool”.

5.2.5.4 Detecting DDoS

Transition probability matrix, frequency vector, sampling rate are determined. A detection algorithm is carried out using these parameters. Using this method, random request DDoS attack can be detected very easily.

5.2.5.5 Summary

Initially, the datasets are preprocessed and various parameters are determined in order to set a threshold. Later, transition probability and request sequence probability are calculated using an algorithm. Now, DDoS attacks are simulated using DDosim tool in linux environment. Later detection algorithm is used to detect the random request DDoS attacks on a web server.

5.2.6 Timeslot Monitoring Model for Application Layer DDoS Attack Detection

A new model for detecting application layer DDoS attacks is proposed in [16]. This model generates the profiles for the traffic patterns of legitimate user and the attacker.

Timeslot Monitoring Model (TMM) generates service request traffic profiles of legitimate users and attackers. Its main goal is to extract IP address of the attacker, to determine whether the traffic is attack traffic or legitimate traffic.

5.2.6.1 Support Vector Machine (SVM)

TMM utilizes a pattern classification algorithm called SVM [17]. It is one of the most accurate classification methods. It monitors the traffic in a period called “Monitoring Period (MP)”. One HTTP GET request is managed at a time under monitoring period. Once the monitoring period is passed, then key features are extracted. Using these features, SVM detects whether it is attack traffic or normal traffic.

5.2.6.2 Summary

TMM monitors the traffic and processes one request at a time. This period is Monitoring period. Once it passes, key features are extracted. Using these key features, SVM detects whether it is attack traffic or normal traffic.

5.2.6.3 Advantages

It requires small amount of memory and CPU resources. It extracts the IP address of the attacker with very high detection rates.

5.2.6.4 Disadvantages

It can be used to detect low amount of application layer DDoS attacks.

5.2.7 CALD: Surviving Various Application Layer DDoS Attacks that Mimic Flash Crowd

Application layer attacks utilize HTTP requests to overwhelm server. These kinds of attacks are more undetectable. It is even more difficult to detect these attacks when they occur during flash crowd event. CALD [18] filters legitimate traffic and blocks the attack traffic. This model is concerned with three types of attacks namely, Repeated request DDoS, Recursive request DDoS, Repeated Workload DDoS. MyDoom [19], Code Red [20] belongs to these kinds of DDoS attacks.

CALD [18] is an architectural extension that protects web servers against various DDoS attacks that mimic flash crowds. It has three major functions, abnormal traffic detection, and DDoS attack detection, filter. The main goal of CALD is to let legitimate traffic and stop attack traffic. It has three main functions namely, Abnormal traffic detection, DDoS attack detection, Filter.

5.2.7.1 Front-end Sensor

Initially, it monitors the traffic to find out if it contains any DDoS attack traffic or flash crowds. Intense pulse in traffic means possible existence of abnormality because it is the basic property of DDoS attacks and flash crowds. If the sensor identifies abnormal traffic, it sends ATTENTION signal and activates the attack detection module. It sends DISMISS signal when it finds that the traffic is normal. Secondly, records average frequency of source IP address and check the total mess extent. Then set a threshold value, malicious IP's are detected. It uses parameters from detection module to filter legitimate traffic and stop attack traffic.

5.2.7.2 Abnormal Traffic Detection

It is a real time series analyzer. This is deployed in front-end sensor. This system is aimed to detect any abrupt changes in the HTTP Get request traffic. The difference between observed behavior and output of the model gives anomalous signature. These signatures are reported as a signal to DDoS attack detection component and identify whether flash crowd or DDoS really happens. A lot of applications having such idea on network traffic analysis have been observed in [21], [22].

5.2.7.3 DDoS Attack Detection

When the sensor at front-end sends an ATTENTION signal, this component is activated. This component traces the incoming source IP address, each visiting webpage, and records the average frequencies in a vector. Based on vector, entropy is calculated. Entropy describes the distribution of incoming sources and target Webpages.

Incoming source IP address = A

Extent of target Webpages =B

Rate between A and B = R.

The value of R is smaller in flash crowds when compared to DDoS attacks in application layer. Thus, threshold values are set and anomalous source IP addresses are detected.

5.2.7.4 Filter

The anomalous Source IP addresses are sent to filter so that it can release the flooding. This model used around 20,000 compromised computers [23] to create DDoS attack. This paper adopted Bloom filter [24]. This model uses Kalman filter to calibrate the prediction results.

5.2.7.5 Summary

First, front-end sensor detects the abnormal traffic, sends ATTENTION signal to Attack detection module. It traces the incoming source IP address, each visiting webpage, and records the average frequencies in a vector. A threshold value is set and malicious IP addresses are found. These addresses are sent to a filter to perform flooding and these IP addresses are blocked and flash crowd is continued.

5.2.7.6 Advantages

Runs attack detection component only when it detects some anomalies. Filters abnormal traffic and leaves the web site safe. It overcomes disadvantage of DDoS-Shield.

5.2.7.7 Disadvantages

Sensitive to slowly increasing DDoS attack traffic.

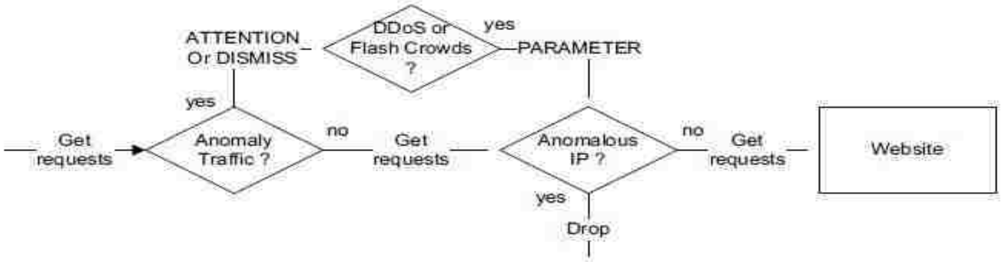


Fig 34: CALD Overview

5.2.8 DDoS–Shield: DDoS Resilient Scheduling to Counter Application Layer Attacks

DDoS-Shield [25] considered sophisticated attacks which are protocol-compliant, non-intrusive, and which utilize legitimate application-layer requests to overwhelm system resources. In [25], the application layer attacks are characterized into three classes namely, request flooding, asymmetric or repeated one-shot on the basis of workload that they exhibit.

DDoS-Shield [25] contains two functions namely, suspicion assignment mechanism and DDoS-Resilient Scheduler. The main goal of DDoS-Shield is to protect web servers from above mentioned application layer attacks.

5.2.8.1 Attacker Model

The goal of the attacker is to degrade the capacity of server from providing services to legitimate users. Through monitoring or profiling, the attacker obtains the information related to server resources that are consumed by different legitimate users. As said before, the attacks at application layer are classified into three classes as Request flooding attack, Asymmetric flooding attack, repeated one-shot attack. Attacker model does not make any assumptions about the set of IP addresses that can be accessed by the attacker. In this model, it is assumed that the system scales its capacity based on the client's demand using Content Distribution Network [26] or a server on-demand infrastructure [27].

5.2.8.2 Victim Model

In victim model, the main focus is on e-commerce applications, which consists of multiple-tiers for processing requests. According to load-balancing policy, once when a request is received, the reverse proxy server parses the request's URL and routes the request to a web server. Victim model assume that all tiers continuously monitor the resources and generates resource utilization reports as well as overall system statistics such as throughput and response time at the application layer. Each e-commerce application is served by various scripts like PHP, JSP etc. Each query originating from the dynamic requests are then redirected to the database server using load-balancing strategy [28], [29].

5.2.8.3 Defense Model

The defense model consists of a *DDoS-Shield. This* is integrated with the reverse-proxy. It schedules or drops attack requests before they reach the web-cluster tier. The *DDoS-Shield* verifies the requests belonging to each session, parses them to get the request type and maintains the request's workload and arrival- history.

5.2.8.4 DDoS-Shield

Suspicion assignment mechanism uses session history to assign a suspicion measure to every client session. *DDoS-resilient scheduler* that decides which sessions are allowed to forward requests and when, depending on the scheduling policy and the scheduler service rate.

5.2.8.5 Summary

This model explores the vulnerability of systems to sophisticated application layer DDoS-attacks which are both protocol-compliant and non-intrusive. A framework is developed to classify these resource attacks as one of request flooding, asymmetric workload, repeated one-shot attacks or combinations thereof, on the basis of the application workload exhibit. Since these resource attacks are un-detectable via application layer techniques, they developed DDoS-Shield, a counter-mechanism which assigns a suspicion measure to a session in proportion to its deviation from legitimate behavior and uses a DDoS-resilient scheduler to decide whether and when the session is serviced. Using a web application hosted on an experimental test bed, they demonstrated the potency of these attacks as well as the efficacy of DDoS-Shield in mitigating them.

5.2.8.6 Advantages

This model detects session arrival misbehavior as well as session workload misbehavior.

5.2.8.7 Disadvantages

This model cannot distinguish flash crowd traffic from the attack traffic. It monitors only abnormal traffic.

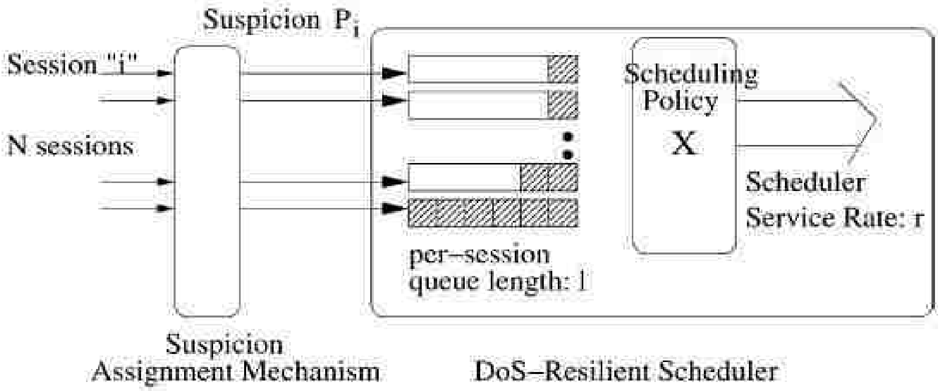

Suspicion P_i
Session "i"
N sessions
Scheduling Policy
X
Scheduler Service Rate: r
per–session queue length: l
Suspicion Assignment Mechanism
DoS–Resilient Scheduler

Fig 35: Defense System Model: DDoS-Shield

5.2.9 Monitoring the Application-Layer DDoS Attacks for Popular Websites

[30] Introduced a scheme to capture the patterns of normal flash crowd and to implement application layer DDoS attacks detection.

It uses access matrix to capture patterns of normal flash crowd, anomaly detector based on HsMM to detect attacks. Its main goal is to identify whether the surge is due to application layer DDoS attacks or due to normal flash crowd which is generated due to high access rate.

5.2.9.1 Detection Principle

[30] Considered application layer DDoS attacks as anomaly browsing behavior. Various results which are significant to this work showed that user’s access behavior can be used to detect anomalous users. This paper used the same concept used by [31]-[32] where the document popularity is used to determine the user behavior.

5.2.9.2 Detection Architecture

Overall detection process is divided into three steps namely, data preparation, training and monitoring. In practical, initially the model is trained by low workload whose normality can be easily detected by anomaly detection systems. Later, this workload is monitored and it is used in anomaly detection.

5.2.9.3 Summary

[30] Proposed detection architecture at monitoring Web traffic in order to detect the dynamic shifts in normal flash crowd. This method is based on PCA, ICA and HsMM. The result shows that, the detection system is able to capture shift of traffic due to normal traffic and traffic due to attacks.

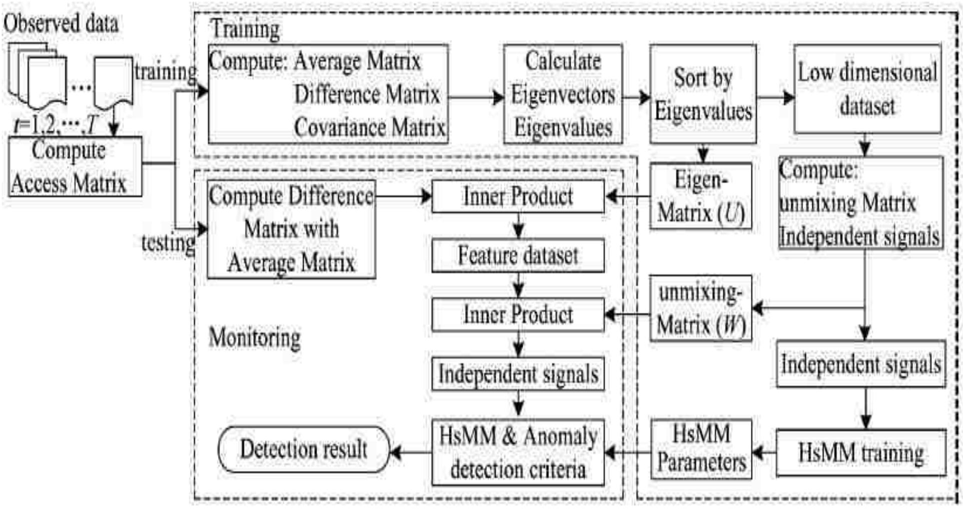


Fig 36: Monitoring Architecture

5.2.10 Detection and Offense Mechanism to Defend Against Application Layer DDoS Attacks

Similar to [25], this paper characterized application layer attacks into three classes namely, session flooding, Request flooding, Asymmetric attacks. It uses a combination of Detection and Currency technology to defend against application layer DDoS attacks.

This paper proposed DOW (Defense and offense wall) mechanism [33] which uses Detection technology (Anomaly detection model), Currency technology (Encouragement model). The main goal of DOW is to minimize delay, maximize service rate.

5.2.10.1 Anomaly Detection Method

It is used to reduce attack request rate and fraction of workload requests. It defends Request flooding and asymmetric attacks. It drops suspicious sessions using anomaly filter. This method has three phases namely, Training, Detection, Filtering.

In training phase, it uses K-means clustering method to build normal client behavior profile. In detection phase, the attacks are detected by a cluster distance based method. In Filtering phase, based on the trust value on each session, the filters drop the suspicious sessions. It filters flooding, asymmetric attacks.

5.2.10.2 Encouragement Model

The sessions dropped through anomaly filter in anomaly detection method are used by encouragement model. Encouragement model encourages the client to retry using the same session. This method defends session flooding. It uses client's session rate or some kind of puzzle as currency. This method encourages more legitimate sessions.

5.2.10.3 Advantages

It offers another chance for legitimate users whose sessions are dropped by anomaly detection model to get service eventually reducing false-positive rate.

5.2.10.4 Disadvantages

It is annoying for legitimate clients to enter the puzzle and it is also causes some delay. Network bandwidth affects the functionality. It is very complicated to train a model and computation is very complicated.

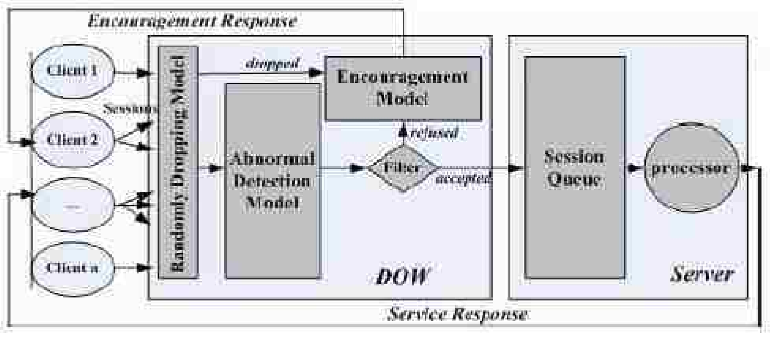


Fig 37: Detection and Offense Mechanism (DOM)

5.2.11 A Three Layer Defense Mechanism Based on Web Servers Against DDoS Attacks

[34] proposed a novel three-layered security mechanism which protects web servers. It filters the illegitimate traffic using statistical filtering and traffic limiting. Traffic limit is used on application layer for DDoS attacks using legitimate IP.

This model performs fair bandwidth allocation among all clients and attackers that are using legitimate IP address. It enforces a law to enforce quota each client may send. After an IP address sends more than Q packets, it will be given a share of 1/10 of its fair share. This bandwidth allocation limits the amount of bandwidth attackers can use. Its main goal is to sustain web server from DDoS attacks and ensure the availability of web services.

5.2.11.1 Summary

Distinguish packets using genuine IP for attack and prevent them from consuming system resources. Thus, allowing legitimate users to pass through.

5.2.11.2 Advantages

Uses a law to enforce quota for limiting amount of bandwidth the attackers can consume.

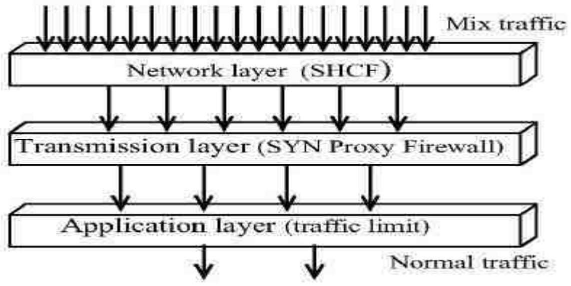

Mix traffic
Network layer (SHCF)
Transmission layer (SYN Proxy Firewall)
Application layer (traffic limit)
Normal traffic

Fig 38: Three-Layer Defense Mechanism

5.2.12 A Novel Model for Detecting Application Layer DDoS Attacks

[35] Considered attacks that utilize HTTP requests and overwhelm the web server at application layer. Hidden semi-Markov Model is applied to measure browsing behaviors and to implement anomaly detection for application layer DDoS attacks. Its functionality is to detect DDoS attacks based on web user browsing behavior.

5.2.12.1 Hidden Semi-Markov Model

It can be used to describe web user browsing behaviors and in implementing anomaly detection. When compared to HMM (Hidden Markov Model), HsMM (Hidden semi-Markov Model) is better in describing second order self-similarity and long range dependence of which might change with time.

5.2.12.2 HsMM for Web Browsing Behaviors

A web user can browse a website by entering the URL or just by clicking on the hyperlinks. It means, web user can log into a single page using different ways. Browsing behaviors can be described as follows, each clicked page is a Markov state (Hidden state), URLs and Embedded objects as observations on the state, Number of requests as duration of the state. Here, Hidden semi-Markov Model (HsMM) [36]-[38] is used to capture browsing behavior of web users. Many researches have been done on capturing web user behaviors in past ten years [39]-[44]. Yu et al in [45] proved that HsMM is better than HMM in anomaly detection.

5.2.12.3 Algorithm for the Model

Consider parameters of new HsMM as $\lambda = (\{a_{mn}\}, \{b_{m(}(v_k)\}, \{p_m(d)\})$ where,

$\{a_{mn}\}$ is transition state probability

$\{b_{m(}(v_k)\}$ is observation probability

$\{p_m(d)\}$ is probability of state duration

Derive backward variable, backward formulae and forward variables and forward formulae. Using these variables, joint probability variables are defined.

5.2.12.4 Detection of DDoS Attacks

Hidden semi-Markov model computes the likelihood of normal user's browsing sequences. This is called Original Likelihood Distribution (OLD). Deviation from this OLD is defined as abnormality in observed request sequence. Usually, HTTP requests are used by the attackers to mimic as legitimate user and overwhelm the server. It results in large deviation from OLD and thus we can easily detect the DDoS attack.

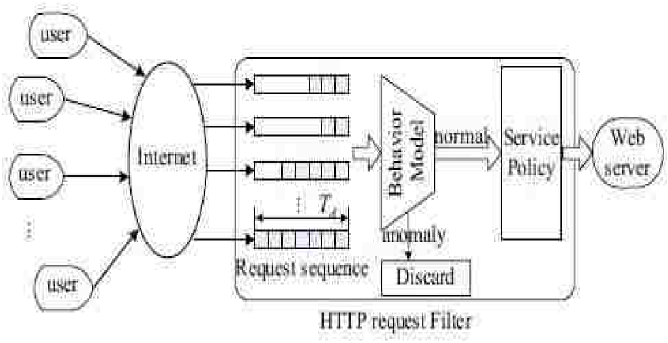


Fig 39: Filter Based on Behavioral Model

5.2.12.5 Summary

Initially, set training data, construct HsMM and OLD. Apply this model to detect DDoS attacks. A filter between internet and victim takes HTTP request sequence and decides whether to accept or reject the request. All the requests that are unaccepted are discarded. The requests that are accepted are passes through the filter and reach the service module.

5.2.12.6 Advantages

This model can distinguish normal browsing data from the bad ones. This model can be integrated with many applications.

5.2.13 Defense Mechanism Proposed by Cisco

Apart from these approaches, Cisco Systems proposed an innovative technology and architecture that delivers protection from DDoS attacks. [46] Considered various key points to build an efficient DDoS protection. They are as follows.

- The approach should not only detect the attack it should also mitigate the attack.
- The approach should easily differentiate between good traffic and bad traffic other than detecting the presence of attack.
- The approach should be reliable and cost-efficient.

5.2.13.1 Cisco Systems DDoS Protection Solution

Cisco provides DDoS protection solution based on principles of detection, diversion, verification, and forwarding to help ensure total protection. When DDoS attack is launched, business continuity is maintained by:

- Detecting the DDoS attack
- Diverting the data traffic
- Analyzing and filtering the bad traffic from good traffic without having any impact on the performance while allowing legitimate users to complete
- Forwarding good traffic to maintain business continuity

5.2.13.2 The Cisco Solution Set

Cisco solution delivers a very rapid response to DDoS attacks which is measured in seconds, not hours. The solution set uses two components.

Cisco Anomaly Detector (TAD) XT which acts as a warning system. It monitors the traffic and detects if there is any deviation from normal behavior. If the deviation is present, then it alerts the Cisco Guard XT. *Cisco Guard XT* which acts as a DDoS-mitigation device. Here, the traffic is subjected to five-stage analysis and filtering process.

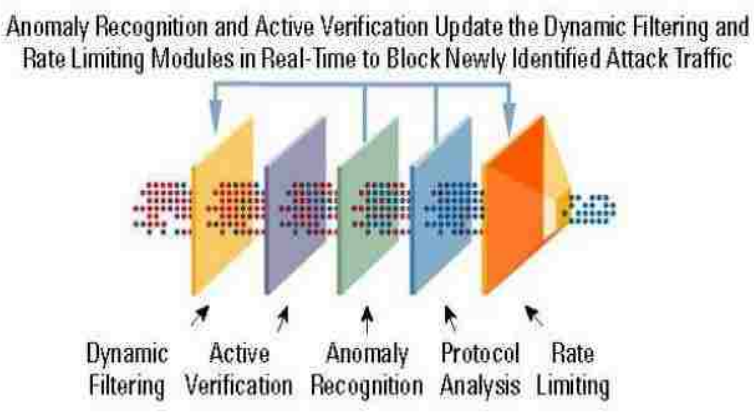

Anomaly Recognition and Active Verification Update the Dynamic Filtering and Rate Limiting Modules in Real-Time to Block Newly Identified Attack Traffic
Dynamic Filtering
Active Verification
Anomaly Recognition
Protocol Analysis
Rate Limiting

Fig 40: Cisco Systems MVP Architecture

This approach scrutinizes the traffic in detail and ensures that DDoS attacks fail to achieve in degrading the target machine. Apart from filtering, Cisco solution cleans malicious data and allows legitimate packets to pass through, thus maintaining the business integrity.

The table below gives an insight about list of approaches and detection categories.

DETECTION CATEGORY	APPROACH
Session History	“DDoS- shield” uses session history to detect the attack
Traffic Monitoring/ Web User Behavior	“CALD”, “A novel method for detecting application layer DDoS attacks”, “An effective approach to counter application layer DDoS attacks”, “Cisco Systems Defeating DDoS Attacks “and “Application layer DDoS detection using Clustering analysis” uses Traffic monitoring or Web user behavior.
Clustered User Sessions	“Detection and offense mechanism to defend against application layer DDoS attacks” uses K-means clustering method to detect attacks. “Application layer DDoS detection using Clustering analysis” uses clustered user sessions.
Pattern recognition	“An effective approach to counter application layer DDoS attacks” and “Timeslot monitoring model for application layer DDoS attack detection” uses pattern recognition to detect an attack.
IP address	“A three layer defense mechanism based on web servers against DDoS attacks” uses IP address to detect the attack traffic.
Signature	“A novel framework to detect and block DDoS attack at application layer” uses signature to determine whether the user is suspicious or not.
Packet Marking	“IP Trace back system for network and application layer attacks” uses packet marking method.

Table 3: Classification Based on Detection Categories

The table below gives an insight about list of approaches and attack category they come under.

ATTACK CATEGORY	APPROACH
Request flooding, Asymmetric or Repeated one-shot, Session Flooding	“DDoS-Shield”, “Detection and offense mechanism to defend against application layer DDoS attacks”
HTTP Request Flooding Attacks	“CALD”, “IP Trace back System for network and application layer attacks”, “A novel method for detecting application layer DDoS attacks”

Table 4: Classification Based on Attack Categories

CHAPTER 6

CONCLUSION AND FUTURE WORK

It is clear that one of the major hazardous security threats today comes from DDoS attacks. Detection and prevention of DDoS attacks is still an ongoing research. From this research, we can see that it is a tedious task to distinguish legitimate traffic from that of the bad traffic. It is even more difficult to block the attack traffic without having any impact on the performance of server in providing services to the legitimate users. In this thesis, we also studied about various approaches to detect and defend against DDoS attacks in application layer, as mentioned in Chapter 5. [47] - [60] also proposed various approaches to detect and mitigate DDoS attacks. Each paper proposed a new method to detect and defend DDOS attacks.

Most of the approaches used user session history or user behavior to detect the anomalies. All the approaches proposed are efficient in their own way, but when it comes to huge amount of attack traffic it becomes difficult to overcome these attacks completely. Lots of approaches have been proposed by various researchers and many papers have been published relating to this problem. Hence, our future direction towards DDoS attack defense would be to collect different data sets from the proposed approaches, compare the results and come up with various mechanisms that can handle and mitigate the DDoS attacks more effectively.

BIBLIOGRAPHY

[1] http://en.wikipedia.org/wiki/Attack_(computing).

[2] James F Kurose, Keith W Ross, *Computer Networking Top-Down Approach,* Fifth Edition, Chapter 1

[3] Xueping Chen, *Distributed Denial of Service Attack and Defense*, International Conference on Educational and Information Technology, 2010.

[4] Arbor Application Brief, *The Growing Threat of Application-layer DDoS attacks.*

[5]Corero White Paper, *Protecting the Enterprise against Today's Distributed Denial of Service (DDoS) Attacks.*

[6] Sujatha Sivabalan, Dr P J Radcliffe, *A Novel Framework to Detect and Block DDoS Attack at the Application Layer,* TENCON Spring Conference, IEEE 2013

[7] S. Kandula, *et al., Botz-4-sale: Surviving organized DDoS attacks that mimic flash crowds,* presented at the Proceedings of the 2nd conference on Symposium on Networked Systems Design \& Implementation - Volume 2, 2005.

[8] J. Rangasamy, *et al., "An integrated approach to cryptographic mitigation of denial-of-service attacks,"* presented at the Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security, Hong Kong, China, 2011.

[9] D. Pogue, *Time to kill off Captchas,* vol. ScientificAmerican.com, p. 15, 2012.

[10] M.Vijayalakshmi, Dr.S.Mercy Shaline, A.Pragash, *IP Traceback System for Network and Application Layer Attacks,* 2012

[11] www.snort.org

[12] Chenhxu Ye, Kesong Zheng, Chuyu She, *Application Layer DDoS Detection using Clustering Analysis,* IEEE 2012

[13] M. Kantardzic, *Data Mining Concepts, Models, Methods and Algorithms,* New York: IEEE Press, 2002.

[14] S.Renuka Devi, P.Yogesh, *An Effective Approach to Counter Application Layer DDoS Attacks,* ICCCNT 2012

[15] Chengxu Ye, Kesong Zheng, *Detection of Application Layer Distributed Denial of Service,* IEEE 2011

[16] Y.S. Choi, J.T. Oh, J.S.Jang, I.K. Kim, *Timeslot Monitoring Model for Application Layer DDoS Attack Detection,* 6th International Conference, 2011

[17] C.M. Bishop, *Pattern Recognition and Machine Learning,* Springer, 2006

[18] Sheng Wen, Wanlei Zhou, Chuan Xu, *CALD: Surviving Various Application-Layer DDoS Attacks That Mimic Flash Crowd,* Fourth International Conference on Network and System Security, 2010.

[19] CERT: Incident Note IN-2004-01 W32/Novarg, *A.Virus,* 2004.

[20] CERT: Incident Note IN-2001-10 *Code Red*, Worm crashes IIS 4.0 Servers with URL Redirection Enables, 2001.

[21] Jouni Viinikka, Herve Debar, Ludovic Me et al. *Processing Intrusion Detection Alerts Aggregates with Time Series Modeling,* Information Fusion, Elsevier, 2009.

[22] Wei Lu and Ali A. Ghorbani. *Network Anomaly Detection Based on Wavelet Analysis*, EURASIP Journal on Advances in Signal Processing, Hindawi Publishing Corporation, 2009.

[23] Georgios Oikonomou and Jelena Mirkovic, *Modeling Human Behaviour for Defense against Flash-Crowd Attacks,* IEEE International Conference on Communications, 2009.

[24] A.Broder and M.Mitzenmacher, *Network Applications of Bloom Filters: A Survey. Internet Math,* Volume 1, Number 4, 2003.

[25] S Ranjan, R Swaminathan, M Uysal, A Nucci, E Knightly, *DDoS-Shield: DDoS-Resilient Scheduling to Counter Application Layer Attacks, INFOCOM'06, 2006*

[26] Akamai. [Online]. Available: http://www.akamai.com

[27] S. Ranjan, J.Rolia, H. Fu, E. Knightly, *QoS-Driven server migration for internet data centers,* presented at the IWQoS, Miami Beach, FL, 2002.

[28] C. Amza, A. Cox, W. Zwaenepoel, *Conflict-aware scheduling for dynamic content applications*, presented at the 4th USENIX Symp. Internet Technologies and Systems (USITS), Seatle, WA, Mar, 2003.

[29] I. Csiszar, *The method of types,* IEEE Trans. Inf. Theory. New York: Wiley, 1991.

[30] Yi Xie, Shung-Zheng Yu, *Monitoring the Application Layer DDoS Attacks for Popular Websites,* IEEE/ACM Transactions on Networking, Vol.17, No. 1, 2009

[31] S. Burklen, P. J. Marron, S. Fritsch, and K.Rothermel, *User Centric Walk: An Integrated Approach for Modeling the Browsing Behavior of Users on the Web,* in proc. 38th Ann. Simulation Symp., Apr. 4-6, 2005, pp.149-159.

[32] C.Roadknight, I.Marshall, and D.Vearer, *File Popularity characterization,* ACM SIGMETRICS Performance Eval. Rev., vol 23, no. 4, pp.45-50, Mar, 2000.

[33] Jie Yu, Zhoujun Li, Huowang Chen, Xiaoming Chen, *A Detection and Offense Mechanism to Defend Against Application Layer DDoS Attacks,* Third International Conference on Networking and Services, 2007

[34] Zhijun Wu, Zhifeng Chen, *A Three-Layer Defense Mechanism Based on Web Services Against Distributed Denial of Service Attacks*, 2006

[35] Yi Xie and Shun-Zheng Yu, *A Novel Model for Detecting Application Layer DDoS Attacks*, Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences, IEEE, 2006.

[36] L. R. Rabiner, *A Tutorial on Hidden Markov Models and selected applications in speech recognition,* in *Proc. of IEEE,* February 1989, Vol. 77, No. 2, pp. 257-286.

[37] J. D. Ferguson, *Variable duration models for speech, in Symp.* Application of Hidden Markov Models to Text and Speech, Oct. 1980, pp.143-179.

[38] S. Z. Yu, and H. Kobayashi, *An efficient forward backward algorithm for an explicit duration hidden markov model, in IEEE Signal Processing Letters*, Vol. 10, No. 1, January 2003, pp. 11-14.

[39] P. Chatterjee, D. Joffman, and T. Novak. *Modeling the clickstream: Implications for Web-based advertising efforts,* Marketing Science. 22 (2003), pp. 520-541.

[40] S. Bürklen, P. J. Marrón, S. Fritsch, et al, *User centric walk: An integrated approach for modeling the browsing behavior of users on the Web,* in the Proceedings of the 38th Annual Simulation Symposium (ANSS'05), 4-6 April 2005, pp. 149 – 159.

[41] S. Dill, R. Kumar, K. S. Mccurley, *Self-Similarity in the Web,* ACM Transactions on Internet Technology, Vol. 2, No.3, August 2002, pp. 205–223.

[42] D. Dhyani, S. S. Bhowmick, and W. K. Ng, *Modeling and predicting Web page accesses using Markov processes*, in Proceedings of the 14th International workshop on the Database and Expert Systems Applications (DEXA'03). 2003. pp. 332-336.

[43] X. D. Hoang, J. Hu, and P. Bertok, *A Multi-layer model for anomaly intrusion detection using program sequences of system calls,* in the Proceeding of the 11th IEEE International Conference on Networks (ICON2003), 28 Sept.-1 Oct. 2003, pp. 531–536.

[44] T. J. Alexander, *Biometrics on smart cards: An approach to keyboard behavior signature,* Future Generation Computer System, 1997, Vol. 13. pp. 19-26.

[45] S. Z. Yu, Z. Liu, M. S. Squillante, C. Xia, and L. Zhang, *A hidden semi-Markov model for Web workload self-similarity,* in Proc. of the 21st IEEE International Performance, Computing, and Communications Conference (IPCCC 2002).

[46] Cisco Systems, Inc. White Paper *Defeating DDoS Attacks*, 2004.

[47] J Yu, C. Fang, L. LU, Z.Li, *Mitigating application layer distributed denial of service attacks via effective trust management*, IET Communications, December 2009.

[48] Jin Wang, Xianolong Yang, Keping Long, *Web DDoS Detection Schemes Based on Measuring User's Access Behavior with Large Deviation*, IEEE Globecom 2011 proceedings.

[49] Arun Raj Kumar, S. Selvakumar, *Distributed Denial-of-Service Threat in Collaborative Environment- A Survey on DDoS Attack Tools and Traceback Mechanisms*, IACC 2009.

[50] Veronika Durcekova, Ladislav Schwartz and Nahid Shahmehri, *Sophisticated Denial of Service Attacks Aimed at Application Layer*, IEEE 2012.

[51] S. B. Ankali, D. V. Ashoka, *Detection Architecture of Application Layer DDoS Attacks for Internet*, Int. J. Advanced Networking and Applications, 2011.

[52] S. Kumar, G. Varalakshmi, *Detection of Application Layer DDoS Attack for a Popular Website Using Delay of Transmission*, IJAEST, 2011.

[53] S. Prabha, R. Anitha, *Mitigation of Application Traffic DDoS Attacks with Trust and AM Based HMM Models*, International Journal of Computer Applications, 2010.

[55] Yu Xie, S. Z. Yu, *A Large-Scale Hidden Semi-Markov Model for Anomaly Detection on User Browsing Behaviors.* Networking, IEEE/ACM Transactions, 2009.

[56] Mudhakay Srivatsa, Arun Iyengar, et. Al. *Mitigating Application-level Denial of Service Attacks on Web Servers: A Client- Transparent Approach*, ACM Transcations on Web, July 2008.

[57] Morein W.G., Stavrou A., Cook D.L., Keromytis A.D., Misra., *Using Graphical Turing Tests to Counter Automated DDoS attacks against web servers*, ACM, 2003.

[58] Yu J, Fang C, Lu L, Li Z, *A Light-Weight Mechanism to Mitigate Application Layer DDoS Attacks*, Proc. Infoscale, 2009.

[59] Y Xie, S. Yu, *A Dynamic Anomaly Detection /model for Web User Behavior Based on HsMM*, in Proc. 10th Int. Conf. Supported Cooperative work in Design, 2006.

[60] Esraa Alomari, Selvakumar, B. B Gupta, *Botnet-Based Distributed Denial of Service Attacks on Web Servers: Classification and Art, International Journal of Computer Applications, July 2012.*

VITA

Graduate College

University of Nevada, Las Vegas

Naga Shalini Vadlamani

Degrees:

Bachelor of Technology in Information Technology, 2011

Jawaharlal Nehru Technological University

Master of Science in Computer Science, 2013

University of Nevada Las Vegas

Thesis Title**:** A Survey on Detection and Defense of Application Layer DDoS Attacks

Thesis Examination Committee:

Chair Person, Dr. Ju-Yeon Jo, Ph.D.

Committee Member, Dr. Yoohwaan Kim, Ph.D.

Committee Member, Dr. Laxmi Gewali, Ph.D

Graduate College Representative, Dr. Venkatesan Muthukumar, Ph.D.