Volume 3, Issue 1, January 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Protecting Against Distributed Denial of Service Attacks and its Classification: An Network Security Issue

[1]Saurabh Ratnaparkhi
M.tech Schlor
CMJ, University
India

[2]Anup Bhange
Asst.Prof
KDKCE,Nagpur
India

Abstract: DoS/DDoS attacks are a strong, comparatively new type of Internet attacks, they have basis some Biggest web sites on the world -- owned by the mainly famous E-Commerce companies such as Yahoo, eBay, Amazon -- became unreachable to customers, partners, and users; the financial losses are very huge. While former security threats could be faced by a tight security policy and active measures like using recalls, vendor patches etc. these DDoS are novel in such way that there is no totally pleasing protection yet. In this paper we classify diverse Forms of attacks and give an indication over the most common DDoS tools. The goal of this paper to is present the idea behind various protecting technique against the DDOS attack.

Keywords: DoS , DDoS, Tribal Flood Network

I. Introduction

Distributed Denial of Service (DDoS) attacks are becoming more abundant, threatening companies that rely on having their websites available online with 100% uptime to visitors, users, and customers. There are a lot of companies operating online businesses including ecommerce, electronic banking, financial services, and other web based companies that depend on security in order to maintain 100% uptime. But most companies that depend on maximum uptime are not aware of the serious damage DDoS attacks can cause, nearly ruining or completely stopping a business in its tracks. When companies that rely on 100% uptime have even .09% downtime, they could lose thousands, even millions of dollars in revenue. Companies count on the fact that their services are “always up” for business success. Service disruption often means substantial financial loss as well as damage to business reputation. In order to maintain 100% uptime, companies spend a lot of money to build and maintain a reliable network infrastructure or they purchase web hosting and data storage services from a reliable hosting company. However, in recent times the companies that provide their customers online services face a new threat. Security threats are as old as the Internet itself. In fact the first connection between computers in the ARPA net between SRI and UCLA resulted in a crash of receive in system due to some bugs in the communication software a classical Denial-of-Service attack [1].

Another prominent story which 'DoS' hundreds of machines is the Internet Worm [2][3][4]. But it was at the beginning of 2000 when a complete new quality of DoS attacks started to be used widely. So called Distributed Denial of Service attacks stroke a huge number of prominent web sites including E-bay, Amazon or Buy.com.

Attack scenario:

Before going more into details on DoS attacks we will first give an overview over the systems that are typically involved in DoS attacks. There are three generic DoS attack methods stand out as particularly dangerous:

Smurf or Fraggle:

Smurf attacks are one of the most devastating DoS attacks. See the Figure 1, in the Smurf (ICMP Packet Magnification) attack, the attacker sends an ICMP echo request (ping) to a broadcast address. The source address of the echo request is the IP address of the victim (uses the IP address of the victim as the return address). After receiving the echo request, all the machines in the broadcast domain send echo replies (responses) to the victim’s IP address (see the Figure 2). Victim will be crash or freeze when receiving larger-sized packet flood from many machines.

Smurf attack uses bandwidth consumption to disable a victim system’s network resources. It accomplishes the consumption using amplification of the attacker’s bandwidth. If the amplifying network has 100 machines, the signal can be amplified 100 times, so the attacker with relatively low. Bandwidth (such as the 56K modem) can flood and disable a victim system with much higher bandwidth (such as the T1 connection)

II. SYN Flood

The SYN flood attack was considered to be the most devastating DoS attack method before the Smurf was discovered. This method uses resource starvation to achieve the DoS attack. See the figure on below, during a normal TCP handshake, a client sends a SYN request to the server; then the server responds with a ACK/SYN to the client, finally the client sends a final ACK back to the server. But in a SYN flood attack, the attacker sends multiple SYN requests to the victim server with spoofed source addresses for the return address. The spoofed addresses are nonexistent on network.

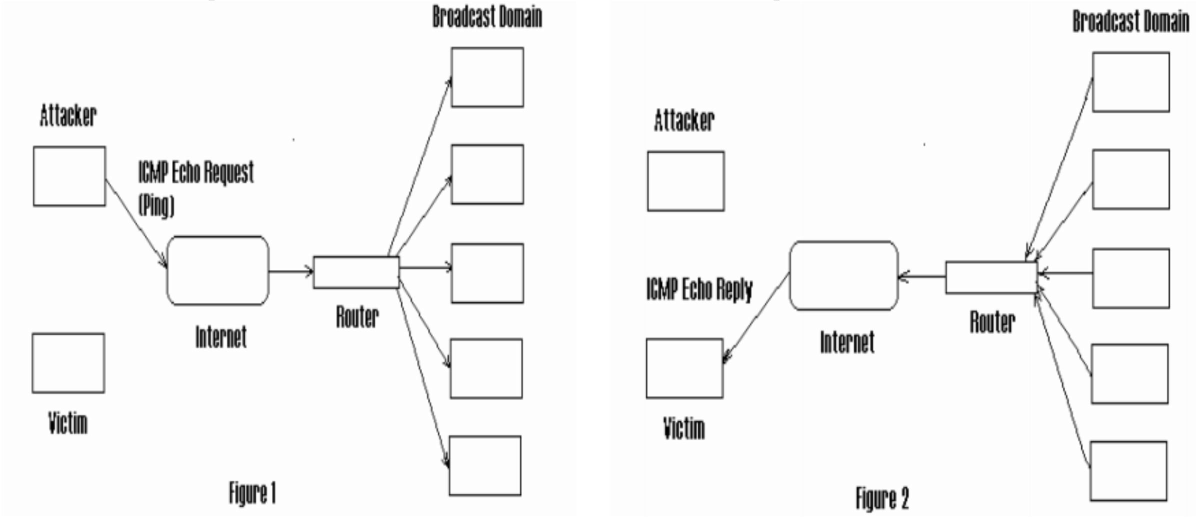

Fig.1 Smurf attacks

The victim server then responds with an ACK/ SYN back to the nonexistent address. Because no address receives this ACK/SYN, the victim server just waits for the ACK from the client. The ACK never arrives, and the victim server eventually times out. If the attacker sends SYN requests often enough, the victim server's available resources for setting up a connection will be consumed waiting for these bogus ACKs. These resources are usually low in number, so relatively few bogus SYN requests can create a DoS event.

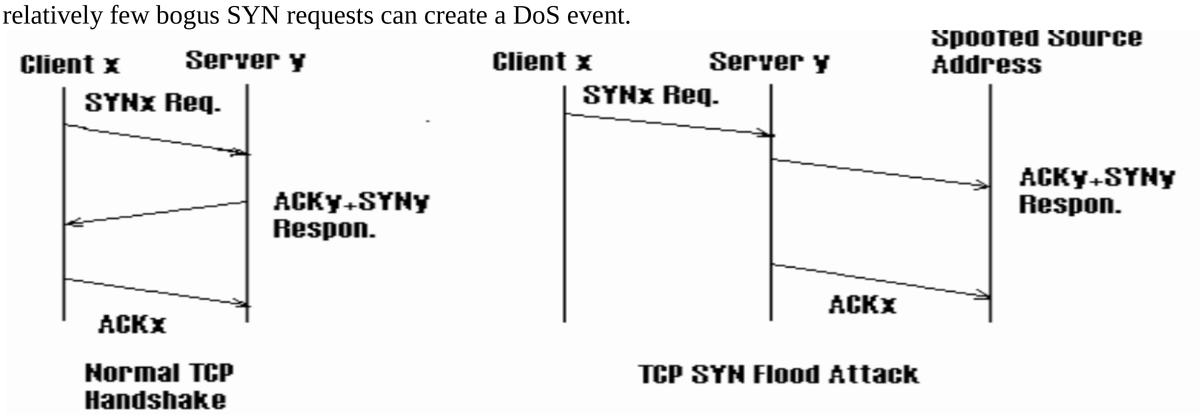

Fig.2 SYN flood attack

III. Definitions for DoS and DDoS

In the rest of the paper we will focus on attack forms which are motivated by the Denial-of-Service intention. The WWW Security FAQ [5] describes a DoS attack as:

... an attack designed to render a computer or network incapable of providing normal services.The most common DoS attacks will target the computer' network bandwidth or connectivity. Bandwidth attacks flood the network with such a high volume of traffic that all available net-work resources are consumed and legitimate use requests cannot get through. Connectivity at-tacks flood a computer with such a high volume of connection requests, that all available operating system resources are consumed and the computer can no longer process legitimate user requests.

J.D. Howard denes DoS as [6]

A Distributed Denial of Service (DDoS) attack uses many computers to launch a coordinated DoS attack against one or more targets. Using client/server technology, the perpetrate at or is able to multiply the effectiveness of the Denial of Service significantly by harnessing the re-sources of multiple unwitting accomplice computers which serve as attack platforms. Typically a DDoS master program is installed on one computer using a stolen account. The master program, at a design at end time, then communicates to any number of" agent" programs, in-stalled on computers anywhere on the Internet. The agents, when they receive the command, initiate the attack. Using client/server technology, the master program can initiate hundreds or even thousands of agent programs within seconds.

The Types of DDoS Attacks:

Generally, DDoS attacks are a combination of four types: Trinoo, TFN, TFN2K, and Stecheldraht.

Trinoo

Trinoo is essentially a master/slave (called Masters and Daemons) programs that coordinate with each other to launch a UDP DoS flood against a victim machine. See the figure, in a typical scenario, the following steps take place as the Trinoo DoS network is set up:

Step 1The attacker, using a compromised host, compiles a list of machines that can be compromised. Most of this process is done automatically from the compromised host, because the host stores amount of information including how to find other hosts to compromise.

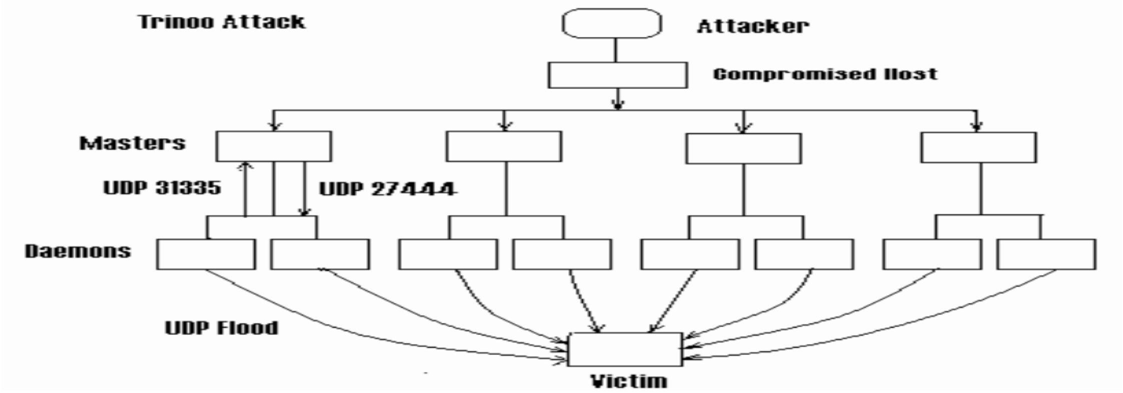

Fig.3 Trinoo Attack

Step 2 As soon as the list of machines that can be compromised has been compiled, scripts are run to compromise them and convert them into the Trinoo Masters or Daemons. One Master can control multiple Daemons. The Daemons are the compromised hosts that launch the actual UDP floods against the victim machine.

Step 3 The DDoS attack is launched when the attacker issues a command on the Master hosts. The Masters instruct every Daemon to start a DoS attack against the IP address specified in the command, many DoS compromise the DDoS attack.

TFN/TFN2K:

TFN (Tribal Flood Network), like Trio, is essentially a master/slave (called Clients and Daemons) programs that coordinate with each other to launch a SYN flood against a victim

Machine, see the figure. The TFN Daemons, however, are capable of a larger variety of attacks, including ICMP flooding, SYN flooding, and Smurf attacks, so TFN attack is more complicated Than the Trinoo attack.

TFN2K introduces some enhancements to the original TFN tool. TFN2K attacks are launched using spoofed IP addresses, making detecting the source of the attacks more difficult. TFN2K attacks are not just simple floods like those in TFN. They also include attacks exploiting the operating system’s vulnerabilities to malformed or invalid packets, which can cause the victim machines to crash. The TFN2K attackers no longer need to execute commands by logging into the Client machine, they can execute these commands remotely. The communication between the Clients and the Daemons is no longer limited to simply ICMP echo replies; it can take place over a larger variety of mediums, such as TCP and UDP. So TFN2K attacks are more dangerous and also more difficult to detect.

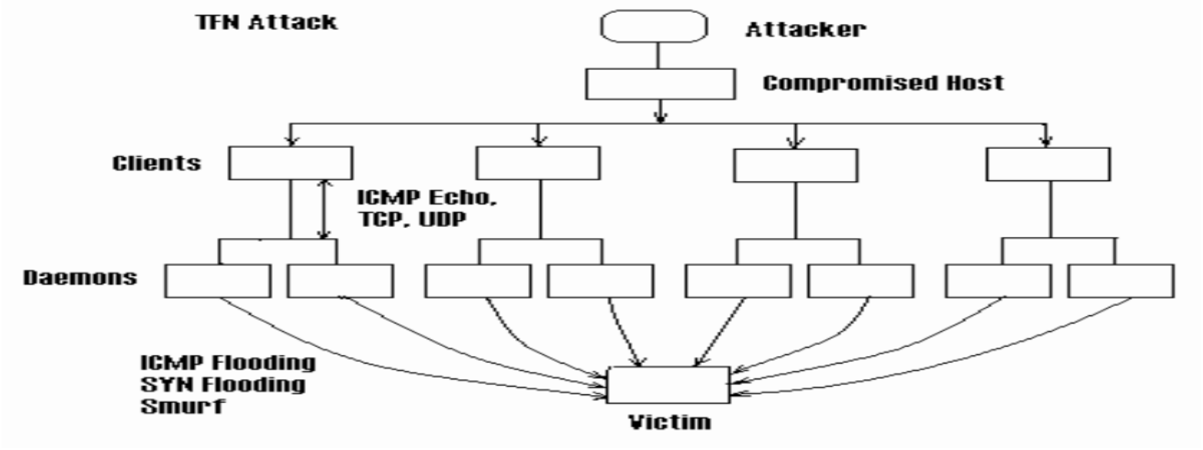

Fig.4 TFN Attack

Stacheldraht:

Stacheldraht code is very similar to the Trinoo and TFN, but Stacheldraht allows the communication between the attacker and the Masters (called Handlers, see the figure) to be encrypted; the Agents can upgrade their code automatically; can launch different types of attacks such as ICMP floods, UDP floods and SYN floods.

From DoS to DDoS:

Major Internet websites like amazon or Yahoo end to have Internet connections with very large bandwidth and server farms with lots of components. Furthermore they are typically protected by firewall systems that block the Known attacks that are based on malformed packets like jolt2 does. In the second half of 1999 DDoS tools matured to a point where a widespread use was foreseeable. In November the CERT/CC invited a number of accredited security experts to a workshop on distributed-systems intruder tools [7]. Their fears about large-scale attacks were proved soon later in February 2000 when major Internet sites where

Under attack [8].There are currently a few popular DDoS at-tack tools, which we will describe in the following sections: Trinoo, Tribe Flood Network (TFN), it's successor TFN2K and a tool called 'stacheldraht'. The architecture of these tools is very similar and in fact some tools are medications of others. The actual attack is carried out by so called daemons. A number of daemons is controlled by a handler and naturally these handlers are activated by the attacker using client tools The intrusions into computers onto which handlers and Daemons are to be installed usually follow a simple pattern [9].

1.As token account is set up as a repository for pre-compiled versions of scanning tools, attack(i.e.bueroverrun exploit) tools, root kits and sneers, DDoS handler and daemon programs, lists of vulnerable and previously compromised hosts, etc.

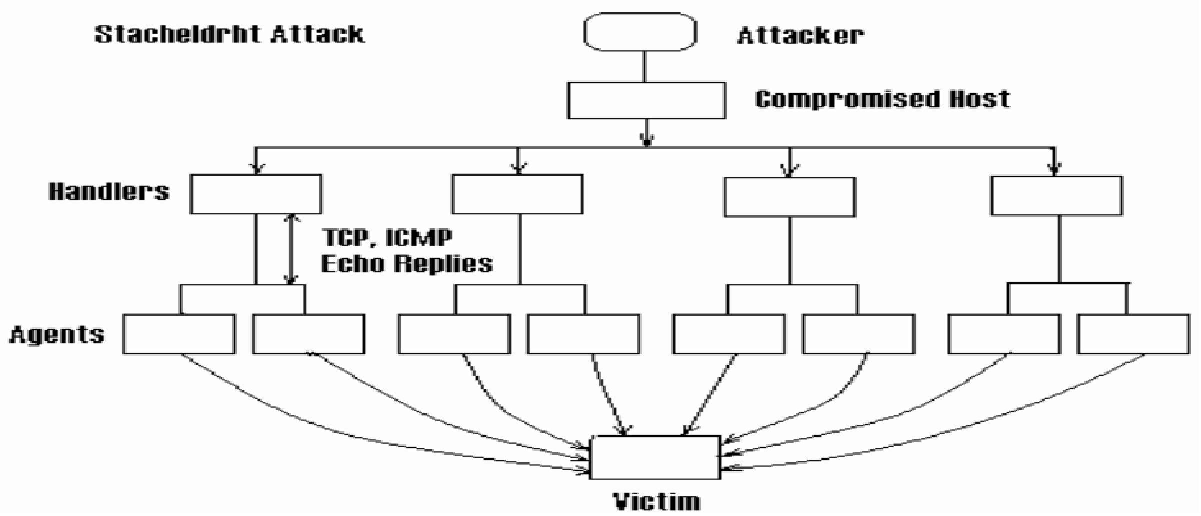

Fig.5 Stacheldraht Attack

2.A scan is performed on large ranges of network blocks to identify potential targets. Targets would include systems running various services known to have remotely exploitable buffer overflow security bugs, such as wu-ftpd, RPC services for "cmsd", "statd", "ttdb-serverd", "amd", etc.

3. A list of vulnerable systems is then used to create a script that performs the exploit, sets up a command shell running under the root account that listens on a TCP port and connects to this port to confirm the success of the exploit.

4. From this list of compromised systems, subsets with the desired architecture are chosen for the Trinoo net-work. Pre-compiled binaries of the DDoS daemons and handlers programs are created and stored on a stolen account somewhere on the Internet.

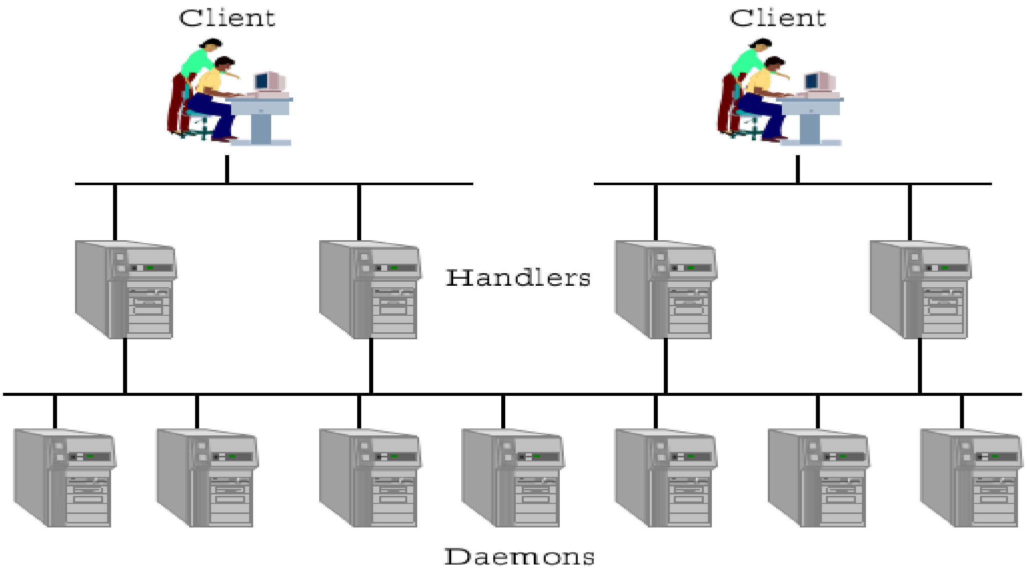

Fig.6 DDOS attack

Our DDoS Protection Environment

Our system consists of several web servers that are accessed via a load balancing tool. All systems are running Linux and all other software is either freely available or self-written. An over view of the components. We use a number of readily available security options in the kernel to provide basic protection. The Linux Virtual Server is used for high speed load balancing. Ip chains re-wall mechanisms protect the load balancer from unwanted traffic. The web servers are running the popular apache server software. Our Traffic Shaping Monitor scans the net-work and reports unusual activity to the central monitor on the load balancer. This component can manipulate traffic using either the Class Based Queuing or ip chains filtering of the kernel.

IV. Linux Kernel

We have carefully chosen Linux Kernel Version 2.2.16 as base for all our systems as this is known to be immune to most poisoned traffic attacks like tear drop or TARGA. The backlog queue of the system defaults to 128 entries and Tcp syn cookies is enabled. This makes the system very robust against SYN flood attacks.

Linux Virtual Server

The load balancer we use is the Linux Virtual Server (LVS)[10].LVS inserts itself directly into the kernel which provides a maximum performance again stabilizing the sys-tem against overload attacks. LVS has two load balancing algorithms: round robin and least connection. We are using 'least connection' as this provides generally a fairer load distribution between the web servers. There are three different modes to access the web servers.

Network address translation (NAT) transcripts every in-coming packet and changes the destination IP from the load balancer's to the web server's IP. All outgoing traffic is transcript alike. As all in- and outgoing traffic has to pass the load balancer, this is not an ideal solution for our purposes, as the load balancer may easily become a bottle neck. Direct Routing Request Dispatching (DR) changes the layer 2MAC addresses of incoming packets to the MAC address of the web server and forwards the packets. Web servers may answer directly by passing the load balancer. All Web-servers must reside in one IP subnet for this to work.

IP Tunneling (IPIP)is a solution where incoming packets are wrapped in an IPIP encapsulation and are tunneled to the web server. At the tunnel end packets get unwrapped and are delivered to the web-server application. We have chosen to use IPIP mode as this has no restrictions on subnets and showed to work very efficiently.

ip chains Firewall

All systems protect themselves from unauthorized access by filtering incoming packets according to a number of security rules. In brief the rules state that only port 80is

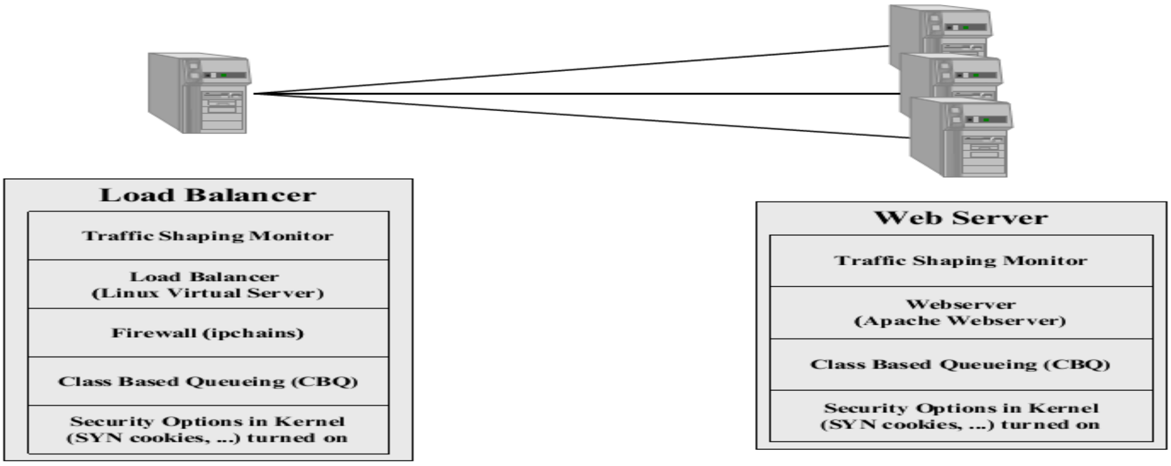
Load Balancer
Traffic Shaping Monitor
Load Balancer (Linux Virtual Server)
Firewall (ipchains)
Class Based Queueing (CBQ)
Security Options in Kernel (SYN cookies, ...) turned on
Web Server
Traffic Shaping Monitor
Webserver (Apache Webserver)
Class Based Queueing (CBQ)
Security Options in Kernel (SYN cookies, ...) turned on

Fig.7 IP Chains

Reachable directly and only ICMP host unreachable messages are accepted. Another set of rules allows communication between the load balancer and the web servers as well as Access to some important local services (DNS server etc.). This conjuration may later be modified dynamically by the Traffic Shaping Monitor to totally block all traffic from attacking hosts. All these measures provide a pretty stable environment which should block all common attacks to the Systems and leave only the web server reachable. The only two potential security holes that are not covered here are bugs in the web-server (or CGI scripts etc.) and overload attacks which generate a large amount of HTTP traffic.

Experimental Setup:

This Research work design and implemented in NS2. NS (version 2) is an object-oriented, discrete event driven network simulator developed at UC Berkelywritten in C++ and OTcl. NS is primarily useful for simulating local and wide area networks. Tcl is a general purpose scripting language. While it can do anything other languages could

Possibly do, its integration with other languages has proven even more powerful. In this section we present the experimental setup of our research work with complete result. As mentioned we use the NS2 to calculate the result. Basically we focus on to detecting and preventing flood and flash crowd anomaly in network. Here we consider the 10 nodes in network and sending the packet at regular interval of time and providing the proper threshold to calculate the anomaly in network. The generalized ratio test can be used to divide the anomalous network. And draw the result through graph.

Flash Crowd Anomaly:

A flash crowd occurs when there is a surge in demand for a service and is typically manifested by a large number of clients trying to access network resources. Flash-crowd anomalies encompass traffic patterns which are caused by a net growth of (usually human) users trying to access a network resource. Typical flash-crowd anomalies are related to overwhelming web server usage patterns.

Flood anomaly:

Flood anomalies include attacks, or any other circumstances, which result in a net growth of Instantaneous traffic. One can think of flood anomalies as having one or more relatively constant traffic sources added to otherwise normal traffic. DDoS attacks typically give rise to Anomalies of this kind. Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks are one of the most common malefic actions over the Internet. This type of attacks consumes the resources of a remote host or network that would otherwise be used to serve legitimate users. Nowadays a diversity of tools is available to accomplish DoS and DDoS.

Design of Network:

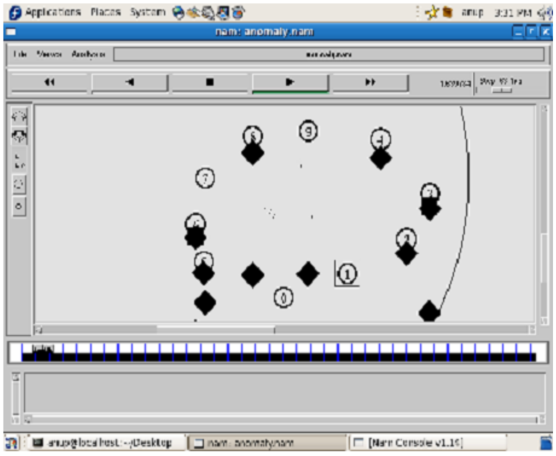

Fig. Design of Transmission of Packet

V. Conclusion

This Paper discusses the DOS and DDOS attack in network and its prevention approach. This paper also explains the various techniques which are help to prevent such attack in the network. And also discuss the various types of DDOS attack and its scenario.

References:

[1] K. Hafner, M.Lyon.Where Wizards Stay Up Late.Simon & Schuster, New York, 1996.

[2] E.H. Spaord. The internet worm program: Ananalysis. Purdue Technical Report CSD-TR-823, Department of Computer Sciences Purdue University,West Lafayette, IN. 1988.

[3] D. Seeley. A tour of the worm. Department of Computer Science, University of Utah, 1988.

[4] M. Eichin, J. Rochlis.With microscope and tweezers: An analysis of the internet virus of november 1988.Massachusetts Institute of Technology, 1988.

[5] L. Stein. The world wide web security faq,version2.0.1.http://www.w3.org/Security/Faq/ - visited04.10.2000

[6] Dr. J.D. Howard.An analysis of security incidents on the internet 1989 - 1995. Carnegie Mellon University, Carnegie Institute of Technology, http://www.cert.org/research/JHThesis/ - visited 02.11.2000.

[7] Results of the Distributed-Systems Intruder ToolsWorkshopPittsburgh, Pensilvania USA, November 2-41999, CERT Coordination Center, SoftwareEngineering Institute, Carnegie Mellon University,Pittsburgh, http://www.cert.org/reports/dsit workshop.pdf -visited 12.11.2000.

[8] M. Williams.Ebay, amazon, buy.com hit by attacks,02/09/00. IDG News Service,02/09/00,http://www.nwfusion.com/news/2000/0209attack.html- visited 18.10.2000

[9] D. Dittrich.The DoS Project's "trinoo" distributed denial of service attack tool. October 21, 1999, http://sta .washington.edu/dittrich/misc/trinoo.analysis.txt - visited 13.11.2000

[10] Linux Virtual Server. http://www.linuxvirtualserver.org/ visisted13.11.2000

[11] Anup Bhange, Sumit Utareja "nomaly Detection and Prevention in Network Traffic based on Statistical approach and α-Stable Model " IJARCET Volume 1, Issue 4, June 2012 ISSN: 2278 – 1323

[12] Anup Bhange, Amber Syad, Satyendra Singh Thakur "DDoS Attacks Impact on Network Traffic and its Detection Approach" IJCA Volume 40– No.11, February 2012