MEDWELL PUBLISHING
International Journal of Soft Computing 4 (2): 68-75, 2009
ISSN: 1816-9503
© Medwell Journals, 2009

Analyzing Interaction Between Denial of Service (DoS) Attacks and Threats

[1]S. Karthik, [1]V.P. Arunachalam and [2]T. Ravichandran
[1]Department of Computer Science Engineering, SNS College of Technology,
Sathy Main Road, Coimbatore 641035, Tamilnadu, India
[2]Hindustan Institute of Technology, Pollachi Main Road, Coimbatore 641032, Tamilnadu, India

Abstract: Denial of Service (DoS) attacks constitutes one of the major threats and among the hardest security problems in today's internet. Of particular concern are Distributed Denial of Service (DDoS) attacks, whose impact can be proportionally severe. With little or no advance warning, a DDoS attack can easily exhaust the computing and communication resources of its victim within a short period of time. Because of the seriousness of the problem, many defense mechanisms have been proposed to combat these attacks. This study aims to provide an understanding of the existing attack methods, tools and defense mechanisms, so that a better understanding of DDoS attacks can be achieved. The goal of the study is to simulate an environment by extending NS2, setting attacking topology and traffic, which can be used to evaluate and compare the methods of DDoS attacks and tools. Based on the simulation and evaluation results, more efficient and effective algorithms, techniques and procedures to combat these attacks may be developed.

Key words: DDoS, attack methods, tools, defenses and simulation

INTRODUCTION

Internet has become the infrastructure of the modern society. The internet architecture focuses on functionality and not the security. Inexperienced users leave their systems vulnerable to compromise. For example, using the vendor supplied default passwords, leaving auto-configure features in default settings, turning off firewalls, etc., makes it easy to gain root or administrator access. The CERT program is part of the Software Engineering Institute (SEI), a federally funded research and development center at Carnegie Mellon University in Pittsburgh, Pennsylvania. The CERT coordinate center, the center of Internet security expertise, has identified 831 key vulnerabilities in the Internet architecture and suggests that automated tools are being used to exploit these security holes (Kevin and George, 1999). The magnitude of DoS attacks against major websites suggests that this is true. Attackers are also, constantly modifying their tools to bypass the security systems and this increases the frequency, severity and sophistication of the attacks (Kevin and George, 1997, 2001). The researchers, in turn modify their approaches to handle new attacks. Many defense systems have been designed in the research and commercial community to counter DDoS attacks, yet the problem remains largely unsolved.

DoS attack is an incident, in which a user or organization is deprived of the services of a resource they would normally expect to have. DoS attacks are, in essence, resource overloading attacks and are capable of either, crashing the host such that it cannot communicate properly with the rest of the network or disrupting/ degrading the host's service and rendering it unavailable for legitimate users (Kevin and George, 1998, 1996). The attack overloads the servers or networks with useless traffic such that the server spends so much time handling the attack traffic such that it cannot attend to its real work. This results in a loss or interruption of all network connectivity and services of the host server.

The goal of the attack is to inflict damage on the victim. The primary resources targeted in an attack are the bandwidth, processing capacity and storage capacity of the victim. The impact on the victim can either be disruptive or degrading. In a degrading attack, the attack consumes some portion of a victim's resources and in a disruptive attack the victim's service is totally denied to its clients. In both types of attack normal operations may resume as soon as the influx of attack packets is stopped. But in some cases, like a server crash, may require human intervention to resume operation. DoS attacks usually cost in terms of money and time and do not normally result in theft of information, damage to databases or security losses.

Corresponding Author: S. Karthik, Thirumaalagam, 160 Kongu Nagar, Perundurai 638052, Erode Dt. Tamilnadu, India

DoS attacks pose a serious threat to the availability of evolving internet services. They have adversely affected service to individual machines, major internet commerce sites and even the core internet infrastructure services. Large scale attacks can cripple the internet-wide communication for hours and can cost victim sites millions of dollars in lost revenue. The attack software is powerful and does not require extensive knowledge to deploy them. The tools for disrupting the services are readily available in the internet. The stateless nature of the internet, dilution of locality in the flooding stream and spoofed source address undermines the effectiveness of traceback techniques for locating the sources. Hence, DoS attacks are becoming simple to implement, harder to detect and more difficult to trace Kevin and George (1997).

MATERIALS AND METHODS

Distributed denial of service: Distributed Denial of Service (DDoS) uses DoS as the basic building block. The main difference between DoS and DDoS attacks lies in its scale of attack and operation mode. DoS attack uses a single attack machine to generate the malicious traffic while, DDoS attacks use large number of attack machines. DoS use sophisticated attack strategies, but DDoS relies on brute force to overload the victim's resources. Both DoS and DDoS attacks are seemingly simple in design and operate without requiring any special skill or resource for their perpetration. The key feature of DDoS includes distributing the attack across hundreds or thousands of compromised hosts (often residing on different networks) and coordinating the attack among the hosts (Ferguson and Senie, 2000).

As shown in Fig. 1, the DDoS attack involves 4 major components-an attacker, handler nodes, daemon/agent nodes and a victim and is carried out in several phases. The attacker orchestrates the attack using a single source machine. The attacker does not directly communicate with the victim, but compromises a number of hosts (handlers and agents/daemons) and develops a network of handler nodes and agent nodes. The attacker controls one or more handler nodes, which in turn controls a number of agent nodes. The attack software is installed on the agents/daemon hosts. The time of the onset of the attack, attack type, duration of the attack and victim address are preprogrammed into the attack code. The victim is flooded with various types of packets from the engaged agents/daemons. The ensuing massive stream of data overwhelms the victim host or routers, rendering them incapable of providing services.

The attacker controls one or more handler nodes, which in turn controls a number of agent nodes. DDoS uses this distributed nature of the attack (dilution of locality in the flooding stream), spoofed source addresses and the stateless nature of the internet to thwart all attempts at discovering the origin of the attack. A successful DDoS attack is one in which the victim is fully overwhelmed and the attacker identity eludes detection.

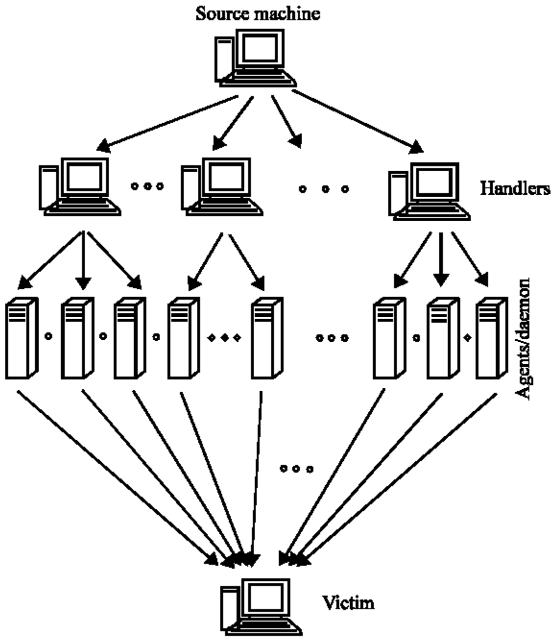

Fig. 1: Distributed denial of service attack scenario

The advantage of the DDoS network structure, which make it extremely difficult to eliminate or stop an attack are:

- Simple attacks. Does not need sophisticated mechanisms or complicated and covert actions like viruses, worms etc
- A single hacker can command hundreds of systems to attack a victim
- Use of IP source spoofing makes it extremely difficult to trace the agent machines
- The attack hosts are replicated and are controlled from a central location. Even if one station is traced and shutdown, the others can continue the attack
- Multi tiered structure makes it difficult to trace the true origin of the attack, which is the client behind the source machine and not the handler or demons
- Attack streams are generated from numerous attack machines spread all over the internet and they converge only in the proximity of the victim. This attack traffic is highly similar to legitimate traffic. This makes it difficult to differentiate the legitimate packets from the high volume of attack traffic and forward them on a packet by packet basis

In order to identify vulnerable machines in the Internet and compromise them a bot software is used. A bot is a program that operates automatically as an agent for a user or another program. The three primary characteristics of a bot are a remote control mechanism, the implementation of commands and a spreading mechanism to propagate it further. These bots are forwarded to handler and agent nodes by scanning either based on host or vulnerability.

Once, a vulnerable computer is identified the attack software automatically infects the vulnerable computers. The bots enable a remote control mechanism in the agents/daemon systems that lets the handler manipulate the infected systems without the user's knowledge. The agent/daemons then waits for commands from the handlers. Typically, 2 types of commands are implemented over the remote control network: DDoS attacks and updates. The bots automatically scan whole network ranges for vulnerabilities, primarily in the operating system. Complexity and various problems in the source code make it easy to exploit and install applications. Once, the vulnerable computers are identified they are quickly infected with the bot software and process repeats itself. The 4 phases of installation are:

Scanning: The installed DDoS attack software (bots) scans a large number of computers for security flaws.

Exploitation: Susceptible hosts are identified and a list of compromised hosts is recorded.

Deployment: The handler software is installed in the compromised hosts. It is a special program, capable of controlling multiple agents.

Propagation: The handler in turn scans for vulnerable hosts and compromises them. An agent or daemon is a compromised host that is running a special program which generates a stream of packets that is directed towards the intended victim. There are 3 common methods of software propagation central source propagation, back chaining propagation and autonomous propagation.

Scanning and exploitation phases are handled by malicious bot software. The bots can be installed on multiple computers to set up botnets. Botnets are a number of computers that although, their owners are unaware of it, have been set up to forward transmissions to other computers on the network. Botnets can be used for the massive distributed denial of service attacks. These installations typically take about 4 sec and allow a large number of systems to be compromised quickly.

DDoS attack methods are categorized as Smurf, ICMP, TCP SYN, UDP, TDP floods and combinations thereof CERT (Kevin and George, 1997). Popular DDoS programs/software include floodnet, Tribal Flood Network (TFN), Trin00, stacheldraht and TFN2K. These programs use a client/server architecture to allow a single attacker to simultaneously direct the attacks by many machines. These attack tools are readily available in the Internet and do not need extensive knowledge to deploy them. Additionally, the software hides the break-in and subsequent activities and erases all the evidence. It is also possible to configure the software to disable and uninstall itself when certain conditions are met. This makes traceback and identification extremely difficult.

DDoS attack methods: DDoS attack may exhaust a key resource by misusing some vulnerability in the software running at the victim (vulnerability attacks) or by simply sending a higher volume of traffic than the victim is provisioned to handle (flooding attacks) (Kevin and George, 1998, 1996).

Vulnerability attacks (e.g., TCP SYN attack) usually contain packets of a special type or content to perform the exploit. As vulnerabilities can frequently be exploited by a few packets, vulnerability attacks are of a low-volume. Special type packets and low volume features simplify handling of vulnerability attacks the victim can either patch its vulnerability or detect the special-type packets and handle them separately.

Flooding attacks (e.g., UDP floods) overwhelm the victim's resource by sheer volume. This strategy is more difficult to counter, as malicious packets can be of any type or content and the high volume hinders detailed traffic analysis.

DDoS attack methods commonly deployed are Smurf, ICMP, TCP SYN, UDP, TCP floods and combinations thereof (Mirkovic and Reiher, 2002).

Smurf floods: Smurf is a reflector attack. Attacker floods the network with ICMP ECHO requests to broadcast address. The source address is set to the address of the target system, which is flooded with the ping response packets overwhelming its network (Kevin and George, 1998).

The attack can be countered by deploying ingress filtering at source network or filtering broadcasted ICMP ECHO requests at intermediate networks.

ICMP floods: The attacker generates a flood of ICMP ECHO packets directed at the victim. The victim replies to each ICMP request, consuming its CPU resources and network resources. The attack is simple to deploy and defend also.

Defense against ICMP flooding attack is done by deploying a simple rate-limiting rule at a high bandwidth point. But this will also, result in dropping a few legitimate requests in the process.

UDP floods: UDP floods send a large number of UDP packets to the target system, effectively tying up the available network bandwidth. Packets usually are of large size and the attack is simple to perpetrate.

Since, many victim sites do not regularly receive large volume of incoming UDP traffic, the attack can be successfully handled by discarding packets at a high bandwidth point using simple filtering rules.

TCP floods: TCP floods are similar to UDP floods. Attackers use TCP packets instead of UDP packets.

TCP SYN: TCP SYN attack is a vulnerability attack which uses the TCP protocol design to perpetrate a TCP SYN flooding attack (Kevin and George, 1996). It research by exploiting the way the servers handle the setup of a TCP connection (the 3-way handshake). Each server has some limit on the number of clients to which it can provide connection/service.

When a session is initiated between the Transport Control Program (TCP) client and server in a network, a very small buffer space exists to handle the usually, rapid hand shaking exchange of messages that sets up the session. The success of the TCP/SYN attack is based on consuming this resource.

The session-establishing packets include a SYN field that identifies the sequence in the message exchange. Upon receiving a SYN packet, the server allocates a connection buffer record, storing information about the client. The server then replies with a SYN/ACK informing the client that its service request will be granted, acknowledging the client's sequence number and sends information about the server's initial sequence number. The client upon receipt of the SYN/ACK packet allocates a connection buffer record. The client then replies with an ACK to the server which completes the opening of the connection. This message exchange is called the 3 way handshake and is depicted in Fig. 2.

An attacker can send a number of connection requests very rapidly and then fail to respond to the reply. To maximize the effectiveness of the attack, SYN attackers usually spoof the IP source address resulting in the server sending the SYN/ACK to spoofed address and hence, no response (ACK) will be received. This results in a half open connection leaving the first packet in the buffer so that other, legitimate connection requests can't be accommodated. Although, the packet in the buffer is dropped after a certain period of time without a reply, the effect of many of these bogus connection requests is to make it difficult for legitimate requests for a session to get established.

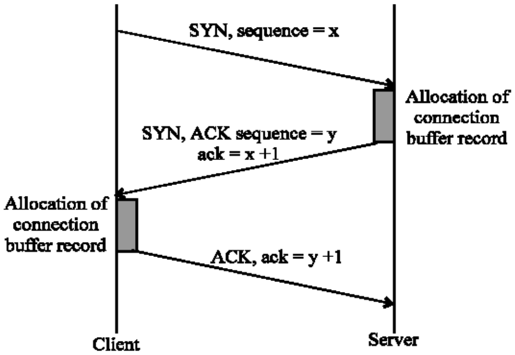
SYN, sequence = x
Allocation of connection buffer record
SYN, ACK sequence = y ack = x +1
Allocation of connection buffer record
ACK, ack = y +1
Client
Server

Fig. 2: TCP Connection opening: Three -way handshake

This is a vicious attack, as servers expect to see large number of legitimate TCP connection requests. Hence, it becomes difficult to distinguish between legitimate SYN requests from attack traffic.

DDoS attack tools: Numerous scripts are available to scan, compromise and infect vulnerable machines, but only a few DDoS attack tools are used to carry out the engagement phase. Popular DDoS programs/software/ tools include floodnet, Tribal Flood Network (TFN), Trin00, stacheldraht and TFN2K. These tools mostly differ in the communication mechanism deployed between masters and slaves and in the customization of attack traffic generation.

These attack tools are readily available in the Internet and do not need extensive knowledge to deploy them. Additionally the software hides the break-in and subsequent activities and erases all the evidence. It is also possible to configure the software to disable and uninstall itself when certain conditions are met. This makes traceback and identification extremely difficult.

Floodnet: It is a java application that inundates the target with request for nonexistent pages and queries. It uses a form of TCP/IP flooding that attacks inbound and outbound data and saturates the processing capability of the target host and the bandwidth of the network.

Trin00: First and simplest of the DDoS software. It uses a master/slave architecture and implements UDP flood package. The target and date of the attack is controlled by the handler. The attacker uses the handler to send commands that control the agents. The attacker authenticates to the handler and sends commands to all the agents to launch the UDP flooding attack targeted at the victim and the attack lasts up to a predefined time. The source address of Trin00 packets is not spoofed. Trin00 supports commands that can change the size of packets sent, stop an attack, check the status of an agent and change the length of the attack.

Tribal Flow Network (TFN): This also uses a master/slave architecture. It comprises of multiple DDoS attacks -Smurf, TCP SYN flood, UDP flood and ICMP flood at specified or random victim ports. TFN uses ICMP ECHO/REPLY to communicate between the agents, handlers and attacker. TFN runs as a root. Hence, source address may be spoofed, making it harder to trace. TFN also, employs coded ICMP packets and not clear text, which hinders detection.

Stacheldraht: Combines the features of TFN and Trin00. It incorporates multiple DDoS attacks-Smurf, TCP SYN flood UDP flood and ICMP ECHO flood. Like TFN, it can spoof the source address. Stacheldraht has an update feature that makes it possible to automatically replace the agents with new versions and start them. It uses encrypted TCP packet to communicate between attacker and handler and encrypted ICMP packets to talk to the agents or daemons.

TFN2K: Improved version of TFN attack tool. But attacks are much harder to detect since communications are encrypted and there is no default key. It can run on both Unix and Windows NT systems and executes as the root or administrator permitting the attacker to verify that the client is running as well as update the client software. It includes special design features:

- To make TFN2K traffic difficult to recognize and filter
- To remotely execute commands
- To conceal the true source of the traffic
- To transport TFN2K traffic over multiple transport protocols including UDP, TCP and ICMP
- To confuse attempts to locate other nodes in a TFN2K network by sending decoy packets

Shaft: Similar to Trin00, TFN and stacheldraht. Shaft uses UDP for communication between masters and agents. Remote control is achieved via a simple telnet connection from the attacker to the master. Agents can generate a UDP flood, TCPSYN flood, ICMP flood, or all 3 attack types. The flooding occurs in bursts, with randomized source port and source address.

Shaft uses tickets for keeping track of its individual agents. Each command sent to the agent contains a password and a ticket. Both passwords and ticket numbers have to match for the agent to execute the request. Additional feature includes

- The ability to switch master servers and master ports on the fly to prevent detection by intrusion detection systems
- Special commands issued by Mater to agents to obtain statistics on malicious traffic generated by each agent

Mstream: Deploys TCP flood with the ACK bit set. Handlers can be controlled remotely by on eor more attackers using a password protected interactive login. The source address in the attack packets are spoofed at random and the communication between attacker and handlers are configured at compile time. The TCP ACK also creates outgoing bandwidth consumption at the victim and along with the incoming flood can exhaust the network resources very quickly.

Trinity: First DDoS tool that is controlled via IRC and eliminates the need for handler machines between attacker and agents. Trinity is capable of launching several types of flooding attacks on the victim site including UDP, IP fragment, TCP SYN, TCP ACK, TCP RST and other floods.

DDoS attack dynamics: During an attack, each participating agent sends a stream of packets towards the victim. The rate of attack packets sent may be constant or variable. In a variable rate attack, the attack packet rate is varied in order to avoid detection. The change in the rate of packets can either be a gradual increase or may be fluctuating. In a fluctuating attack, the attack rate is adjusted based on the victim's behavior or a preprogrammed timing. The agent may also periodically abort the attack and resume it later. This makes extremely difficult to determine the sources of attack.

Attacks are not always aimed at a particular host machine. Depending on the type of victim, attacks can be classified as application, protocol, operating system, host, network and infrastructure attacks.

Application attacks exploit some feature of a specific application on the victim host, disabling the legitimate clients from using that application and also ties up the resources of the host machine.

Protocol attacks misuse vulnerability in a specific version of the protocol on the target machine to consume some of its critical resource.

Operating system attacks misuse vulnerability in a specific version of an OS installed at the target machine. OS attacks are capable of slowing down or completely freezing the operation of the target machine.

Host attacks disable access to the target machine completely by overloading or disabling its communication mechanism.

Network attacks consume the incoming bandwidth of a target machine with attack packets whose destination address can be chosen from the target network's address space.

Infrastructure attacks target some distributed service that is crucial for global Internet operation or operation of a sub-network.

DDoS defenses: In recent years, DDoS attacks have increased in frequency, sophistication and strength. Though many solutions have been proposed, the problem has not been solved yet. The primary objective of the defense strategy is to ensure that legitimate clients receive the expected service even during an ongoing attack with any degradation in the service (Ferguson and Senie, 2000; Mirkovic *et al*., 2002).

The defense approaches can be classified as preventive, survival and responsive (Ferguson and Senie, 2000). A preventive approach tries to avoid the incidence of attack by eliminating the vulnerabilities in the existing system and preventing the attacker from gaining control over a number of zombie machines. A survival approach tries to reduce the effect of the attack by increasing its resources during an attack in order to serve the legitimate clients. A responsive approach tries to detect the occurrence of the attack and responds by controlling the attack streams and/or source machines.

The defense system can either be deployed as an autonomous system or as a distributed system (Savage *et al*., 2000; Cisco, 2008). In an autonomous defense a single node is used to detect and respond to the attack. A distributed defense employs a number of interconnected systems, which may be deployed anywhere but are organized into a network. The nodes communicate and coordinate their actions to achieve a better overall defense. The point of defense can be categorized as victim end defense (deployed at a point close to the victim), intermediate network defense (deployed in core routers capable of detecting and handling the attack) or source end defense (deployed close to the sources of attack usually the ingress routers).

The defense can further be classified as system security defense or protocol security defense (Cisco, 2008). System security mechanism can increase the overall security of the Internet hosts and routers, guarding against illegitimate accesses to a machine, removing application bugs and updating protocol installation to prevent intrusion and misuse of the system. Protocol security defense addresses the problems created by bad protocol design.

Attack detection is primarily of 2 types-pattern detection and anomaly detection (Lippmann and Robert, 1999). In pattern detection signatures of known attacks are stored in a database. Each communication is monitored and compared with the entries in the database to detect DDoS attack occurrences. Anomaly detection have a model of normal system behavior such as normal traffic dynamics or expected system performance. The current state of the system is periodically compared with the models to detect anomalies. Advantage of anomaly detection over pattern detection is that it helps to detect previously unknown attacks also.

The defense strategy involves 4 primary factors-agent identification, rate limiting, filtering and reconfiguration (Savage *et al*., 2000).

Agent identification: Mechanisms provide the victim with information about the identity of the machines that are performing the attack.

Rate limiting: Mechanisms impose a rate limit on a stream that has been characterized as malicious by the detection mechanism.

Filtering: Mechanisms use the characterization provided by detection mechanisms to filter out the attack streams completely.

Reconfiguration: Mechanisms change the topology of the victim of the intermediate networks to either add more resources to the victim or to isolate the attack machines.

An effective defense approach should be capable of accurately detecting a wide range of attacks. It should also relieve the victim from the impact of the attack and ensure minimal damage to legitimate clients.

RESULTS AND DISCUSSION

Simulation of DDoS: The 3 primary components of a DDoS attack scenario that jointly determine all the elements of an attack scenario that influence its impact on a network's infrastructure and a defense's effectiveness are the legitimate traffic, the attack traffic and the topology. Hence, an effective simulation of the DDoS attack requires a collection of traffic generation tools, topology and defense library, experiment control scripts and tools for statistics collection, analysis and visualization (Meadows, 1999). NS2 allows security researchers to replicate threats of interest in a secure environment and to develop, deploy and evaluate potential solutions.

The DDoS simulation considers these elements along 3 dimensions.

DDoS attack traffic: A malicious packet mix arriving at the victim and the distribution and activities of machines involved in the attack.

Legitimate traffic: Communication patterns of the target network.

Network topology and resources: The target network architecture.

Attack traffic generated by the listed attacks interacts with legitimate traffic by creating real or perceived contention at some critical resource. The level of service denial depends on the following traffic and topology features: attack rate, attack traffic on and off periods in case of pulsing attacks, the rate of legitimate traffic, amount of critical resource-size of connection buffers, fragment tables, link bandwidths, CPU speeds, path sharing between the legitimate and the attack traffic prior to the critical resource, legitimate traffic mix at the TCP level-connection duration, connection traffic volume and sending dynamics, protocol versions at end hosts, legitimate traffic mix at the application level- since different applications have different quality of service requirements, they may or may not be affected by a certain level of packet loss, delay or jitter.

The simulation of DDoS attack was performed with a range of routing, bandwidth and delay configurations. Figure 3-6 show the visualization of legitimate traffic and attack traffic at a chosen interface. The incoming and outgoing traffic for the various nodes were modeled for well known port numbers.

Within the selected traffic mix, the distributions of the number and length of service requests, the reply length and the request inter-arrival time are configured. For example, the outgoing traffic from the node consists of traffic to port 53, with the following characteristics (Table 1).

The following are some proposed countermeasures that could apply to packet floods and congestion control exploits to detect, prevent, or counter attacks; a single defense could embody several mechanisms.

Path isolation: Routers mark, sample or record packets to isolate traffic paths. Path information can be used to deploy filters on the path, or to perform fair sharing of resources.

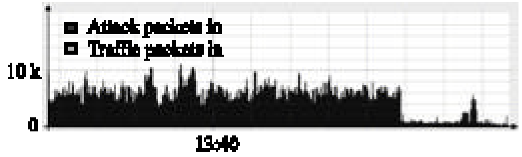

Fig. 3: Incoming traffic at node

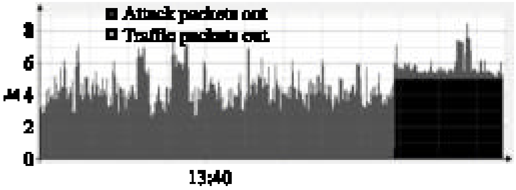

Fig. 4: Outgoing traffic at node

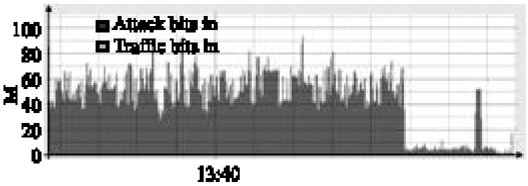

Fig. 5: Incoming bit rate at node

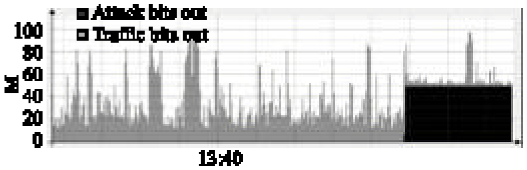

Fig. 6: Outgoing bit rate at node

Table 1: Traffic model at a model

Port	Traffic feature	Distribution
53	Requests per second	Poisson (1.828)
	Requests per host	Pareto (1.1, 2.17)
	Requests size	Pareto (32.74, 2.5)
	Reply size	Pareto (117.5, 3.1)

Privileged customer: Some customers obtain passes that allow privileged access to the critical resource, in form of capabilities, authorization to enter a dedicated overlay, knowledge of the server's identity, good classification, etc. A defense prioritizes traffic with passes.

Traffic base lining: Many traffic parameters are observed over time to learn their valid value ranges. During attacks, some parameter values will exceed their predicted range, which can be used to devise fine-grain filters or to isolate attack packets.

Resource multiplication: Distributed resources are deployed (statically or dynamically) to sustain large attacks.

Legitimate traffic inflation: Legitimate traffic is multiplied to enhance its chances to win in the fight for the limited resource.

CONCLUSION

Internet has not significantly changed in recent years. Network resources remain limited and susceptible to consumption attacks and systems still contain vulnerabilities, new and old, that either remain un-patched or are patched in a less than timely manner. Internet management is distributed and each network is run according to local policies defined by its owners. The implications of this are many. There is no way to enforce global deployment of a particular security mechanism or security policy and due to privacy concerns, it is often impossible to investigate cross-network traffic behavior. Each Internet entity (host, network, service) has limited resources that can be consumed by too many users. Regardless of how well secured the victim system may be, its susceptibility to DDoS attacks depends on the state of security in the rest of the global internet. The end result is there are still plenty of vulnerable systems on the internet that can be used as launch points for DDoS attacks.

Evolution in intruder tools is a long-standing trend and it will continue (Lippmann and Cunningham, 1999). And, DDoS attacks by their very nature are difficult to defend against and will continue to be an attractive and effective form of attack. Automation of attack tool deployment and ease of management will continue to be areas of focused evolution for DDoS tools. It is also likely, at least in the short term, that advancements in DDoS attack technology will take shape in the form of protocol-specific attacks, such as attacks on routing protocols, rather than as significant innovations in basic characteristics of packet flooding streams. While, we do not propose solutions for the issues discussed in this study, it is important to recognize and understand trends in attack technology in order to effectively and appropriately evolve defense and response strategies to help evaluate how security policies, procedures and technologies may need to change to address the current trends in DDoS attack technology.

ACKNOWLEDGEMENT

The author would like to thank Dr. S.N. Subbramanian, Director cum Secretary, Dr. S. Rajalakshmi Correspondent of SNS College of Technology, Coimbatore for their motivation and constant encouragement. The author would like to thank Dr. V.P. Arunachalam, Principal cum Research Supervisor for critical review of this manuscript for his valuable input and fruitful discussions. Also, I extend thank my family members and friends who supported us throughout this research.

REFERENCES

Cisco, 2008. Strategies to protect against distributed denial of service attacks. http://www.cisco.com/warp/public/707/newsflash.html.

Ferguson, P. and D. Senie, 2000. Network ingress filtering: Defeating denial of service attacks which employ. IP Source Address Spoofing. RFC 2827.

Kevin, J.H. and M.W. George, 1999. CERT Coordination Center, Denial of Service Tools http://www.cert.org/advisories/CA-1999-17.html.

Kevin, J.H. and M.W. George, 1997. CERT Coordination Center, IP Denial-of-Service Attacks. http://www.cert.org/advisories/CA-1997-28.html.

Kevin, J.H. and M.W. George, 2001. CERT Coordination Center, Trends in Denial of Service Attack Technology. http://www.cert.org/archive/pdf/DoS trends.pdf.

Kevin, J.H. and M.W. George, 1998. CERT Coordination Center, Smurf attack. http://www.cert.org/advisories/CA-1998-01.html.

Kevin, J.H. and M.W. George, 1996. CERT Coordination Center, TCP SYN flooding and IP spoofing attacks. http://www.cert.org/advisories/CA-1996-21.html.

Lippmann, R.P. and K. Robert, 1999. Computer Security. Technology Center. Cunningham Improving Intrusion Detection Performance Using Keyword Selection and Neural Networks. www.raid-symposium.org/raid99/PAPERS /Lippmann1.pdf.

Mirkovic, J., G. Prier and P. Reiher, 2002. Attacking DDoS at the source. In: Proceedings of the ICNP.

Mirkovic, J. and P. Reiher, 2002. A taxonomy of DDoS attack and DDoS defense mechanisms. In: Proceedings of the 2nd ACM SIGCOMM Internet Measurement Workshop.

Meadows, C., 1999. A formal framework and evaluation method for network denial of service. In: Proceedings of the 12th IEEE Computer Security Foundations Workshop.

Savage, S., D. Wetherall, A. Karlin and T. Anderson, 2000. Practical network support for IP traceback. In: Proceedings of ACM SIGCOMM.