Protecting Web Servers from DoS/DDoS Flooding Attacks
A Technical Overview

Noureldien A. Noureldien
College of Technological Sciences
Omdurman, Sudan

Email: noureldien@hotmail.com

Abstract

Recently many prominent web sites face a new type of denial of service attack known as Distributed Denial of Service attack (DDoS). Organizations deploying security measures such as firewalls, and intrusion detection systems could face the traditional DoS attack. Yet there is no complete solution neither for protection from DDoS attack, nor for preserving network hosts from participating in such an attack.

This paper explains how DoS/DDoS attacks are launched and discuss different proposed solutions that aimed to protect Web Servers from the attack or to minimize its effect. These solutions spreads over the organization's entire Internet infrastructure, that includes boarder routers, firewalls, active monitors, load balancer, and the target host/server.

1. Introduction

A service is any aspect of a computer system's functioning that provides benefits to a user. Any intervention that reduces or eliminates the availability of that service is called a Denial of Service, often abbreviated DoS [1]. DoS attacks are as old as the Internet itself. In fact the first connection between computers in the ARPAnet resulted in a crash of the receiving system due to some bugs in the communication software, a classical DoS attack [2]. Another prominent attack is the Internet Worm [3].

It was at the beginning of 2000 when a complete new quality of DoS attack started to be used widely. The so-called Distributed Denial of Service attack (DDoS) stroke a huge number of prominent web sites including Yahoo, Ebay, Amazon and Buy.com [4]. Statistical analysis confirms that vulnerabilities in computer systems have increased, and that

denial of service exploitation of these vulnerabilities, as measured by web site defacements, has also increased [5].

2. DDoS Attacks

A DDoS is a type of an attack technique that saturates the victim system with enormous network traffic to the point of unresponsiveness to the legitimate users. A DDoS attack system has a complicated mechanism and entails an extreme coordination between systems to maximize its attacking effectiveness. The attack systems involved three system components: handlers, agents and a victim respectively.

A DDoS attack is possible by the coordination of many systems. To clog up the victim's network with enormous network traffic, the attacker needs to use a number of systems as handlers and agents. The attacker commands handlers and the handlers control a troop of agents to generate network traffic.

To make a successful attack, an attacker first needs to have a number of systems to secure a bridgehead, usually large systems with high-speed network connection. To compromise such systems as much as possible and install DDoS tools on each of them, an attacker must find those systems with various techniques such as network port scanning, and other known infiltrating techniques. Also, to hide those DDoS tool's presence after installation, the attacker may use other techniques such as IP address spoofing. The installed DDoS tools turn the compromised systems into attack zombies. Once the DDoS tools are installed on many compromised systems, the attacker finds it easy to launch an attack by controlling agents through handlers via commands. Once an attack begins, the target is not able to handle the tremendous volume of the bogus traffic [5].

2.1 DoS/DDoS Flood Attack Methods

Numerous DoS flood attack methods have been documented.

Smurf Attack

An attacker sends forged ICMP echo packets to broadcast addresses of vulnerable

networks. All the systems on these networks reply to the victim with ICMP echo replies. This rapidly exhausts the bandwidth available to the target, effectively denying its services to legitimate users.

TCP SYN Attack

Taking advantage of the flaw of TCP three-way handshaking behavior, an attacker makes connection requests aimed at the victim server with packets with unreachable source addresses. The server is not able to complete the connection requests and, as a result, the victim wastes all of its network resources, resulting in shutting down a server.

UDP Attack

A UDP Flood Attack is possible when an attacker sends a UDP packet to a random port on the victim system. When the victim system receives a UDP packet, it will determine what application is waiting on the destination port. When it realizes that there is no application that is waiting on the port, it will generate an ICMP packet of destination unreachable to the forged source address. If enough UDP packets are delivered to ports on victim, the system will go down.

TCP Attack

TCP floods are similar to UDP floods, except the attacker uses TCP packets instead of UDP packets.

ICMP Attack

An attacker sends a huge number of ICMP echo request packets to victim and, as a result, the victim cannot respond promptly since the volume of request packets is high and have difficulty in processing all requests and responses rapidly. The attack will cause the performance degradation or system down.

2.2 DDoS Tools and Their Attack Methods

While this paper focuses on defensive measures against DoS/DDoS floods, it is important to know the major and best-known tools used to launch this attack [6]:

Trinoo

Trinoo, was the first known DDoS tool, Trinoo is a distributed SYN DoS attack, where masters and daemons communicate using the ports shown in the table below.

The Tribe Flood Network (TFN)

TFN is used to launch a number of attacks, such as ICMP flood, SYN flood, UDP flood, and SMURF attacks. TFN is noticeably different than trinoo in that all communication between the client (attacker), handlers, and agents use ICMP ECHO and ECHO REPLY packets. Communication from the TFN client to daemons is accomplished via ICMP ECHO REPLY packets. The absence of TCP and UDP traffic sometimes makes these packets difficult to detect because many protocol monitoring tools are not even configured to capture and display the ICMP traffic.

Stacheldraht

Stacheldraht (German for "barbed wire") is a DDoS tool that combines features of trinoo and TFN. It also contains some advanced features, such as encrypted attacker-master communication and automated agent updates. The possible attacks are similar to those of TFN; namely, ICMP flood, SYN flood, UDP flood, and SMURF attacks.

Trinity

Trinity is DDoS tool that can be used to launch several types of flooding attacks on a victim site. Communication from the handler or intruder to the agent, is accomplished via Internet Relay Chat (IRC) or AOL's ICQ; Trinity appears to use primarily port 6667 and also has a backdoor program that listens on TCP port 33270.

Shaft

Shaft is another DDoS tool that looks conceptually similar to a trinoo; it is a packet flooding attack and the client controls the size of the flooding packets and duration of the attack.

Tribe Flood Network 2K

Tribe Flood Network 2K (TFN2K) is a complex variant of the original TFN with features designed specifically to make TFN2K traffic difficult to recognize and filter, remotely execute commands, hide the true source of the attack using IP address spoofing.

The tools listed above are the best known and most widely sued, but they are not the only ones and more tools are becoming available. An analysis reports for the above toots is found in [7][8][9][10][11][12].

2.3 DoS/DDoS Exploited Vulnerability

DoS/DDoS attacks exploit different vulnerabilities to deny the service of the victim Web server to its clients. Based on the vulnerability that

is targeted during an attack, two different exploits can be identified, namely, protocol attacks and brute-force attacks [13].

Protocol Attacks

Protocol attacks exploit a specific feature or implementation bug of some protocol installed at the victim in order to consume excess amounts of its resources. Examples include the TCP SYN attack, the CGI request attack and the authentication server attack.

Brute-force Attacks

Brute-force attacks are performed by initiating a vast amount of seemingly legitimate transactions. Since an upstream network can usually deliver higher traffic volume than the victim network can handle, this exhausts the victim's resources.

Further brute-force attacks can be divided into *filterable* and *non-filterable* attacks, based on the relation of packet contents with victim services.

Filterable Attacks

Filterable attacks use bogus packets or packets for non-critical services of the victim's operation, and thus can be filtered by a firewall. Examples of such attacks are a UDP flood attack or an ICMP request flood attack on a Web server.

Non-filterable Attacks

Non-filterable attacks use packets that request legitimate services from the victim. Thus, filtering all packets that match the attack signature would lead to an immediate denial of the specified service to both attackers and the legitimate clients. Examples are a HTTP request flood targeting a Web server or a DNS request flood targeting a name server.

3. DoS/DDoS Defense Mechanisms

Different proposed solutions that aimed to protect from the DoS/DDoS attacks or to minimize its effect are exists. These solutions spreads over the organization's entire Internet infrastructure, that includes boarder routers, firewalls, active monitors, load balancer, and the target host/server.

We classify proposed solutions into three broad categories: system level mechanisms, network level mechanisms and global mechanisms. As the name suggests, system solutions can be implemented on the machine (Web Server) we want to protect, network mechanisms are implemented on a network perimeter. Global solutions, by their very nature, require the cooperation of Internet community.

3.1 System Level Mechanisms

System mechanisms guard against illegitimate accesses, scanning system for malicious DDoS software, and removing of application bugs to prevent intrusions and misuse of the system. Examples of system security mechanisms include scanning tools, monitor access to the machine [14], virus scanners, moving target defense [15], client bottlenecks [16], and access lists for critical resources [17].

3.1.1 Scanning Tools

A system scanning tools should be implemented on the protected Web Server to determine if any of the known DDoS tools are present on the server file system. Since DDoS tools become obsolete as new DDoS exploits are invented or existing ones are modified to evade detection. Scanning tools has to be recently updated to handle the latest DDoS attack methods. An example for DDoS scanning tool is "find_ddos" [18].

The protected server must also be scanned for open ports on a regular basis using tools such as nmap or saint. Ports used by well know DDoS tools that are typically used to remotely control compromised machines must be blocked.

3.1.2 Client Bottlenecks

The objective behind this approach is to create bottleneck processes on the zombie computers, to limit their attacking ability. The methods used require the attacking computer to correctly solve a small puzzle before establishing a connection. Solving the puzzle consumes some computational power, limiting the attacker in the number of connection requests it can make at the same time.

3.1.3 Moving Target Defense

A system protection technique is the so-called moving target defense. Here the host under attack changes its IP address to avoid being attacked. The problem here is that, the legitimate users of the system need to be informed that the IP address has changed, which is usually done by updating the DNS system. Due to caching it can take up to a number of days until all clients are informed of the update. This is clearly unacceptable, as the effects are as severe as any DoS attack can be. Furthermore, attackers only need to incorporate DNS lookups into their tools in order to evade this protection [15].

3.2 Network Level Mechanisms

Network mechanisms can either be deployed at the victim-network, Intermediate-network or at the source-network. DDoS defense mechanisms deployed at the victim network protect this network from DDoS attacks and respond to detected attacks by alleviating the impact on the victim. DDoS defense mechanisms deployed at the intermediate network such as ISP's provide infrastructural service to a large number of Internet hosts. DDoS defense mechanisms deployed at the source network is to prevent customers using this network from generating DDoS attacks.

3. 2.1 Boarder Routers

Many mechanisms for defeating DDoS attacks at routers have been proposed, examples include Ingress, Egress Filtering, and MULTOPS.

3.2.1.1 Ingress Filtering

Ingress Filtering is an Intermediate-network mechanism. Internet Service Providers (ISPs) can take actions against DoS/DDoS that include: eliminating routing of spoofed packets by discarding any packet that contains any RFC 1918 or reserved IP address in the IP source address or destination address. Also they should perform Ingress filtering [19] on their routers to drop packets with IP addresses outside the range of a customer's network, so that they can prevent attackers from using forged source addresses to launch a DoS attack. The weaknesses of applying ingress filtering technique is that, it does nothing to address flooding attacks that originate from valid IP addresses, and may negatively affect mobile IP services.

3.2.1.2 Egress Filtering

Egress Filtering is a source-network mechanism. SANS institute urged network administrators to adopt egress filtering which prevents one's network from being the source of forged communications used in DoS attacks [20]. An egress filter is designed for implementation in the routers at the edge of a network. These filters analyze packets as they are forwarded to their intended destination, looking for forged (spoofed) IP addresses. Since any particular network is assigned a specific subset of IP addresses, any packet containing an invalid IP address is assumed to be spoofed, and the filter drops such packets. This ensures that only IP packets with valid source IP addresses leave the network and thus protects the outside from spoofed packets.

Egress filtering has two severe shortcomings. Firstly, there is little incentive for an ISPs to provide egress filtering since it does not protect from the attack, it only keeps an attacker from using the network for a DDoS attack. If egress filtering is not employed by a significant number of networks, it will not be a viable solution to DDoS attacks. Secondly, egress filtering will not detect internally spoofed IP addresses.

3.2.1.3 MULTOPS Bandwidth Attack Detection

MULTOPS is a source-network defense mechanism. This solution postulates that if a network administrator (a victim) were able to detect an IP addresses that participate in a DDoS attack, then measures could be taken to block only these particular addresses. The solution is a heuristic one, and defines a data-structure that network devices (such as routers) can use to detect (and eliminate) DoS attacks. The Multi-Level Tree for Online Packet Statistics (MULTOPS) is a tree of nodes that contains packet rate statistics for subnet prefixes at different aggregation levels [21]. MULTOPS uses disproportional rates to or from hosts and subnets as a heuristic to detect (and potentially stop) attacks.

3.2.2 Firewalls

Firewalls are victim-network mechanisms. Most firewalls built today are designed to enable a form of protection against SYN floods. Firewalls are better suited to fight the attack because they tend to be designed to examine packets and maintain connection and state information of session traffic. As a countermeasure to DoS attacks, firewalls can be configured as a relay, as a semi-transparent gateway [15], or combine other techniques.

3.2.3 Active Monitoring

This category is of solutions is a victim-network mechanisms, it consists of using software agents to continuously monitor TCP/IP traffic in a network at a given place (Router/Firewall). An agent can collect communication control information to generate a view of all connections that can be observed on a monitored network. Furthermore, active monitors can watch for certain conditions to arise and react appropriately. Examples for active monitors are synkill from COAST Laboratory [22], The Nozzle [23], and The SYNDEF [24].

3.2.4 Load Balancing

The last victim-network defense against DoS floods is to distribute the flood against as many hosts or network devices as possible. In the case of commercial web sites or corporate sites that are well known and have considerable throughput the load balancing is probably already in place.

A solution that is based on Class Based Routing mechanisms in the Linux kernel is proposed [15]. The solution is oriented to suite big sites that

used load-balancing server and aimed to keep the web servers under attack responding to normal requests. The solution uses a number of configurable input queues on the load balancer and output queues on the web servers. If the traffic monitor in the load balancer detects a possible DoS attack, it slows the traffic from the origination IP address by assigning it to a slower queue or block it at the firewall.

3.3 Global Mechanisms

Clearly, Brute-force DDoS floods threaten the Internet as a whole, local solution to the problem become futile. Global solutions are seems better from a technological point of view. Global proposed solutions are:

3.3.1 Improving the security of the entire Internet.

Improving the security of all computers linked to the Internet would prevent attackers from finding enough vulnerable computers to break into and plant daemon programs that would turn them into zombies.

3.3.2 Using globally coordinated filters.

The strategy here is to prevent the accumulation of a critical mass of attacking packets in time. Once filters are installed throughout the Internet, a victim can send information that it has detected an attack, and the filters can stop attacking packets earlier along the attacking path, before they aggregate to lethal proportions. This method is effective even if the attacker has already seized enough zombie computers to pose a threat.

3.3.3 Tracing the source IP address.

The goal of this approach is to trace the intruders' path back to the zombie computers and stop their attacks or, even better, to find the original attacker and take legal actions. If tracing is done promptly enough, it can help to abort the DDoS attack. Catching the attacker would deter repeat attacks. Examples of traceback mechanisms are [25], and [26]. However, two attacker techniques hinder tracing: IP spoofing that uses forged source IP addresses, and the hierarchical attacking structure that detaches the control traffic from the attacking traffic.

4. Conclusion

Distributed denials of service attacks are a complex and serious problem, and consequently, numerous approaches have been proposed to counter them. The multitude of current attack and defense mechanisms obscures the global view of the DDoS problem.

The ultimate solution for preventing DDoS/DoS is to detect and block floods at source-networks. This cuts the problem off before it can ever manifest. Thus many experts suggest that we "pull together as a community" to secure our Internet computers from becoming unwitting accomplices to such malicious intruders.

References

[1] Denial of service Attack on the Rice. http://www.safesite.com/WhitePapers/DoSandDDoS.asp .

[2] K. Hafner, "Where Wizards Stay Up Late", Simson & Schuster, New York 1996.

[3] E.H. Spafford, "The Internet Worm Program: An Analysis". Purdue Technical Report CSD-TR- 823, Purdue University, West Lafayette, IN. 1988.

[4] M. Williams, "Ebay, Amazon, Buy.com hit by attacks", IDG News Service, 02/09/00 http://www.nwfusion.com/news/2000/0209attack.html –

[5] Understanding DDoS Attacks, http://www.sans.org/infosecFAQ/threats/understanding_DDoS.htm

[6] Gary C. Kessler, "Defenses against Distributed Denial of Service Attacks", November 29, 2000 http://www.garykessler.net/library/

[7] CERT/CC. 'CERT Advisory CA-1999-17 Denial of Service Tools.'' 3 March 2000. http://www.cert.org/advisories/CA-1999-17.html

[8] CERT/CC. 'Results of the Distributed-Systems Intruder Tools Workshop.' 2-4 November 1999. http://www.cert.org/reports/dsit_workshop.pdf

[9] Dittrich, S. D. Dittrich, and N. Long. ' An analysis of the 'Shaft' Distributed denial of Service Tool'. 13 March 2000. http://www.sans.org/y2k/shaft.htm

[10] Dittrich, David. ' The DoS Project's 'Trinoo' distributed denial of service attack tool'. 21 October 1999. http://staff.washington.edu/dittrich/misc/trinoo.analysis

[11] Dittrich, David. ' The stacheldraht' distributed denial of service attack tool'. 31 December 1999. http://staff.washington.edu/dittrich/misc/stacheldraht.analysis

[12] Dittrich, David. ' The Tribe Flood Network' distributed denial of service attack tool'. 21 October 1999. http://staff.washington.edu/dittrich/misc/ftn.analysis

[13] Jelena Mirkovic, Janice Martin and Peter Reiher. "A Taxonomy of DDoS Attacks and DDoS

Defense Mechanisms".
http://www.icir.org/pushback/
[14] Tripwire, "Tripwire for servers,"
http://www.tripwire.com/products/servers/
[15] Frank Kargl, et. al. Protecting Web Servers from Distributed Denial of Service Attacks. May 2001.
http://www10.org/cdrom/papers/409/protectingwebservers_ddos.htm
[16] A. Juels and J. Brainard, "Client puzzles: A cryptographic countermeasure against connection depletion attacks," In *Proceedings of the 1999 Networks and distributed system security symposium (NDSS'99)*, Mar 1999.
[17] Cisco, "Strategies to protect against distributed denial of service attacks,"
http://www.cisco.com/warp/public/707/newsflash.html
[18]"find-ddos" http://www.fbi.gov/nipc/trinoo.htm
[19] P. Ferguson, D. Senie. "Network Ingress Filtering: Defeating Denial of Service Attacks Which Employ IP Source Spoofing". RFC 2827 May 2000.
[20] Engress Filtering. V 0.2. GIAC Special Notice, SANS Institute Resources,
[21] T. M Gil and M. Poleto, "MULTOPS: a data-structure for bandwidth attack detection," In *Proceedings of 10th Usenix Security Symposium*, August 2001.
[22] C. Schulba, I. Krsul, M. Kuhn, E. Spafford, A. Sundram, D. Zamboni, "Analysis of a Denial of Service Attack on TCP", In *proceedings of the 1997 IEEE Symposium on Security and Privacy*.
[23] Elizabeth Strother. ' Denial of Service Protection, The Nozzle'. *In Proceeding of 16th Annual Computer Security Applications Conference, December 2000*.
[24] Noureldien A.N, Izzeldin M. Osman, 'The SYNDEF: A method for defeating DoS/DDoS TCP SYN Flooding Attacks'. In Proceedings of the 14th Annual FIRST Computer Security Incident Handling Conference, Hawaii, U.S.A June 2002.
[25] D. Dean, M. Franklin and A. Stubblefield, "An algebraic approach to IP Traceback," In *Proceedings of the 2001 Network and Distributed System Security Symposiu*m, February 2001.
[26] S. M. Bellovin, "ICMP traceback messages," Internet draft, http://search.ietf.org /internet-drafts/draft-ietf-itrace-01.txt, Oct. 2001.