International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Analysis of Various Distributed Denial of Service Networks Attacks Detection and Prevention Techniques: An Overview

Mangesh Jaideorao Parate[1], Dr. Vaishali Dinesh Khairnar[2]

[1]Research Student, Department of Information Technology, Terna Engineering College
Nerul, Navi Mumbai, India
[2]Professor, Department of Information Technology, Terna Engineering College
Nerul, Navi Mumbai India

Abstract

Attacks known as Distributed Denial of Service (DDoS) aim to prevent the use of Internet resources and services. To create DDoS attack networks, DDoS attackers infect vast numbers of machines. Thus, a coordinated, large-scale assault on one or more victim systems is launched. Aside from improving current DDoS attack methods, attackers create new and derivative DDoS assault tools. Develop thorough DDoS solutions that prevent known and future DDoS attack variations rather than reactive defenses. Nevertheless, this needs a thorough comprehension of the scope and strategy of DDoS attacks. This investigation attempts a full DDoS scope. We suggest a new taxonomy of DDoS attack networks, DDoS attack methods, and software tools used to build DDoS attack networks. These categories help you understand the scope of DDoS attacks, tools, and issues. We propose a type of mitigation that addresses DDoS before, during, and after an attack. This initiative aims to promote research into innovative DDoS defenses and detection systems and to help develop complete solutions that defeat both known and derivative DDoS assaults.

Keywords: Intrusion detection, DDOS attack, Network connection, network connections, malicious users, probing, user to root attack, probing attacks, root to login attack.

1. Introduction

To prohibit legitimate users from accessing a victim's computer or network resources [1]. DDoS is a coordinated, wide-scale assault on a target system or network resource through the Internet by a large number of compromised devices. The compromised systems utilized to conduct the assault are termed "primary victims" rather than "secondary victims

"If the initial attack targets more people, it may launch more destructive and harder-to-track DDoS attacks. The WWW (“World Wide Web”) Security FAQ states: A DDoS attack is a coordinated DoS attack that uses client-server technology by exploiting the capabilities of many unwitting accomplice computers as an attack platform. [2] As per CIAC, the 1stDDoS attack took place in the summer of 1999. [3] The first large-scale DDoS attack was against Yahoo.com in February 2000. was. The attack lasted almost two hours and caused a significant loss of advertising revenue [4]. On October 20, 2002, a new DDoS attack targeted the DNS Root Servers (DNS). Logical addresses are translated into physical IP addresses, so users can use names instead of numbers to connect to their websites. Having 13 of her

servers severely hinders internet access. His one-hour attack, which had minimal impact on ordinary Internet users, brought down seven of his 13 critical root servers, highlighting the vulnerability of the Internet [5]. [7] If left unchecked, a massive DDoS attack can disable or destroy critical Internet services in minutes. The study categorized DDoS attack networks, tools, attacks, and defense mechanisms. DDoS attacks are a recent problem that many people are unaware of. For example, this study is the first to distinguish between techniques for building and installing active and passive DDoS attack architectures.

The scale of DDoS attacks can be better understood by showing various DDoS attack networks, describing DDoS attack strategies, and describing DDoS software tools. Useful for DDoS detection, prevention, and mitigation systems. Expect more as solutions to known and undetected threats become more and more complete. Based on our insights in creating these taxonomies, we have created a taxonomy for DDoS mitigation. DDoS attacks can be mitigated, mitigated, and forensically investigated. Section 2 provides a selection of DDoS attack networks. Section 3 classifies DDoS attacks. Section 4 discusses DDoS attack tools. Examine how DDoS attack networks interact with software on secondary target PCs. Section 5 details the procedures for DDoS attack tools. Section 6 summarizes general DDoS attack techniques. Section 7 presents a taxonomy of DDoS mitigation. Further research to develop a complete DDoS solution using these taxonomies is recommended in Section 8.

2. DDoS Attack Network

Figure 1 below shows the breakdown structure and protocol of the DDoS attack and handler model.

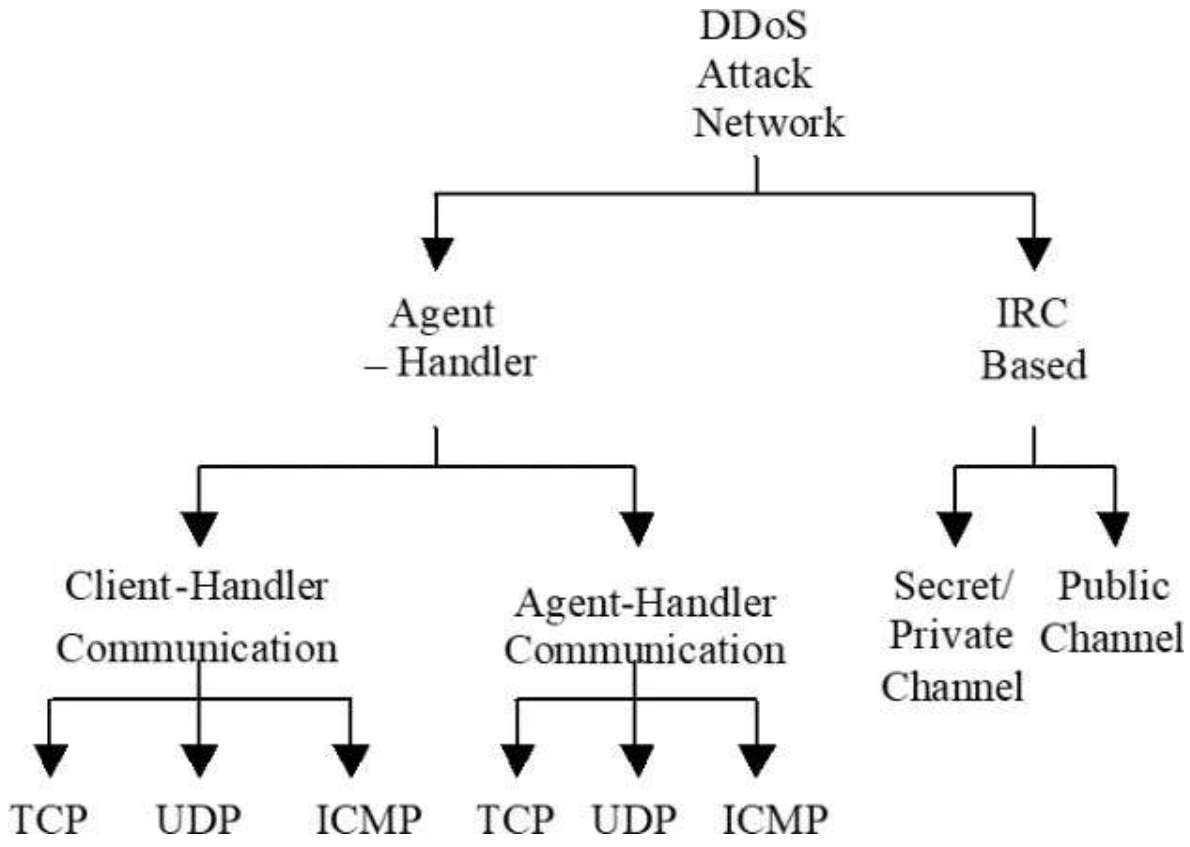

Figure 1: DDoS Attack Networks

2.1 Agent-Handler Model

Agent Handlers are connections between two groups. Derived from the term "distributed denial of service," attackers connect to the remainder of the DDoS assault network through the customer layer. The attacker installs her Wardens on her PC via the Internet to covertly communicate with the experts. Expert software is installed on the infected PC and finally attacks the targeted system. Attackers work with multiple controllers to determine which specialists are active, when to plan attacks, and when specialists need to be updated. Experts can connect to a single controller or multiple controllers, depending on the attacker's DDoS attack configuration. Attackers usually aim to install her Overser software on a heavily loaded switch or organizational server. This makes it difficult to distinguish between client and controller, and manager and expert. TCP, UDP, or ICMP can be used for communication between attackers and controllers, and between controllers and experts. Professional system owners and customers are often unaware that their systems are compromised and vulnerable to DDoS attacks. During a DDoS attack, each specialized program consumes a small number of resources (memory & data transfer) and causes minor performance issues for PC clients. In the depiction of DDoS equipment, supervisors and experts are often replaced by experts and daemons. Hacked PCs running specialized software are known as help victims, while DDoS attack targets are known as primary victims.

2.2 IRC-Based DDoS Attack Model

Internet Relay Chat is an online communication platform for multiple clients.

It allows PC users to form and communicate in two-party or multi-party networks [6]. The geography of the IRC network consists of IRC servers scattered over the Internet connected by channels. IRC clients can create open, private, and secret channels. The term "public channel" refers to a channel where some people can share recordings and messages. Public channel clients can see their IRC ID and messages of other clients [7]. People create covert channels to communicate with specific customers. The identities and communications of registered customers are hidden from unregistered clients in both private and confidential channels [8]. Even if private channel content is hidden, certain channel localization rules allow clients who do not have channel members to discover it, but private channels are more difficult to discover unless the client is a channel member. An IRC communication channel connects the client to her IRC-based DDoS assault team instead of utilizing a controller application that is installed on a corporate server. Using an IRC channel can help attackers use this DDoS attack strategy. For example, an attacker could use a "legal" IRC port [9]. This makes it more difficult to track the delivery of DDoS orders. Also, IRC servers have a lot of traffic, making it easy for bullies to hide from their bosses. You no longer need to monitor specialists as you can get a list of all open specialists just by joining the IRC channel [9]. Once the IRC network's special programming is fully operational, it will periodically communicate with its IRC channel and alert attackers.

A fourth feature of the IRC network is the rapid exchange of documents. Section 4 discusses dataset sharing as an impartial strategy for expert code authoring. Thus, the attacker builds a few simple memories that bind more victims together and help them complete their misdeeds. In his IRC-based DDoS attack, an expert called “bots” or “zombie bots”. Both his IRC-based DDoS attacks and his DDoS attacks on agent handlers refer to agents as optional zombies or victims.

3. Classification of DDoS Attacks

DDoS attacks come in many forms. A scientific ranking of the most common DDoS attack methods. These attacks are classified as either transmission rate exhaustion or resource consumption. A capacity exhaustion attack overwhelms the target organization and prevents authenticated traffic from reaching the victim (main) system. Asset consumption attacks aim to choke the assets of the objective framework. This attack disables the victim's system's server or interface, making it unable to process a large number of support requests.

3.1 Bandwidth Exhaustion Attacks

There are 2 types of DDoS data transfer exhaustion attacks. Flood attacks use zombies to disrupt the flow of data from emergency call systems. An attacker or zombie broadcasts a message to a broadcast IP address that affects all PCs on the subnet accessible through the broadcast address. This strategy increases false exchanges and slows down machine transmission speeds.

3.1.1 Flood Attacks

The zombie floods the target computer with IP traffic. Zombies bombard the target system with packets to slow it down, crash it, or overload the network. Legitimate users cannot contact the victim. Agent Handler and her IRC-based attack network.

Flood-based UDP Attacks: Flood-based UDP attack: User Datagram Protocol (UDP) does not require a connection. With UDP, the sender and receiver do not have to perform a handshake, the receiving system simply receives the data packets it needs to process. The network might get overloaded and have less bandwidth available for valid service requests if it receives a lot of UDP packets. UDP packets are sent to targeted or random ports on the victim's system during a DDoS UDP flood attack. UDP flood assaults frequently target unknown victim ports. So the processes of the victim's system the incoming data to see which program requested the data. "Destination port unreachable" is sent by the victim's system when no app is running on the destination port [3]. Typically, the attacker's DDoS software spoofs her IP address in the attacker's packets. This hides the secondary victim identity and guarantees that return packets from the system of the victim are sent to a bogus address and not to zombies. A UDP flood attack can impact the bandwidth of nearby links (depending on network architecture and line speed). Any system connected to the network close to the victim's system may experience connectivity issues.

ICMP Flood Attack ICMP (“Internet Control Message Protocol”) packets are used for network management operations like discovering network devices and examining hop counts or round-trip times. For example, a ping packet (ICMP ECHO REPLY) can be used to query the runtime on the target system. A DDoS ICMP flood assault arises when a zombie floods a target system with her “ICMP ECHO REPLY” packets. The resulting data traffic overloads the victim's network connection [3]. The source IP address of a UDP flood attack can be spoofed.

3.1.2 Amplification Attacks

The broadcast IP address option on most routers is used by DDoS amplification operations to amplify and mirror assaults. This feature allows transmission systems to use broadcast IP addresses instead of predefined addresses. This tells network routers to deliver the packet to the broadcast address range. In a DDoS attack, an attacker may either send broadcast messages directly or use an agent to send them. To

break into workstations on broadcast networks and use them as undead without installing agent software, an attacker would have to send broadcast messages directly. Enhanced attacks include Smurf & Fraggle attacks.

Smurf Attack: An attacker transmits data to the network expander (which allows broadcast addresses) to launch a DDoS Smurf assault. The attack packet is generally an ICMP ECHO REQUEST expecting a response (such as "ping") from the recipient [10]. In this case, the repeater will send her ICMP echo packets to every machine in the broadcast address range and respond to her target victim's IP address with her ICMP ECHO REPLY.

Such an attack amplifies the initial packet dozens or dozens of times.

Fraggle Attacks: The attacker transmits packets to the network's repeaters comparable to the Smurf assault. UDP ECHO packets are used by Fraggle in place of ICMP ECHO packets [12]. Fraggle attacks can be customized because the sender address mimics the victim's echo service [13]. Broadcast address Destination for UDP Fraggle packets on the access system. Each of these systems generates characters to send echo packets to the character generator. More harm may be done by this assault than by a smurf attack, and it also creates more unwanted traffic.

3.2 Resource Depletion Attacks

Attackers use DDoS attacks to exhaust network capacity by exploiting Internet Protocol connections or sending malformed packets.

3.2.1 Protocol Exploit Attack

TCP SYN Attack: Before sending any data packets, TCP needs a complete handshake between the sender and the recipient. She receives a SYN request from the starting system. A response from the receiving system is an ACK. The sending system then responds with an ACK, providing two-way communication. After some time, a receiver who receives a SYNX packet but does not get her ACKY+1 to her SYNY back will send another HER ACK + SYNY [14]. The recipient system's processing and memory resources are used by this TCP SYN request until it times out. This inhibits the target server from responding to genuine requests by overloading its CPU resources.

The exploit relies on a 3-way handshake between the sending & receiving systems by flooding the target system with TCP SYN packets from spoofed source IP addresses. If the server receives a lot of SYN requests but no ACK+SYN response, it starts to run out of CPU and memory. Affected systems quickly run out of resources and become unresponsive to real users.

PUSH and ACK Attack: The TCP protocol queues packets for the destination and forwards them to the receiving system when the stack is full. In a PUSH packet, the sender asks the receiver to clear the buffer before it becomes full. The TCP header has a PUSH 1-bit flag [15]. To reduce the processing cost needed on the receiving system every time a non-empty buffer is unloaded, TCP keeps incoming data in huge chunks for transmission. PUSH and ACK attacks, like the TCP SYN attack, try to use the resources of the target platform. The PUSH and ACK flags on the TCP packets sent by the attacker's agent are both set to 1. These packets provide the victim's system to clear all data from TCP buffers (full or not) and confirm completion. Performing this operation on multiple agents crashes the receiving node.

3.2.2 Malformed Packet Attacks

Malformed IP packets are sent to the target system by a zombie under the attacker's control in a malformed packet attack. He may attack with two different forms of faulty packets. Attacks using IP addresses have the same source and destination IP address. This may crash the OS of the infected system. An IP packet options attack allows a malformed packet to randomize options fields and set all "quality-of-service" bits to 1, slowing data processing on the target system. If enough agents are used, the victim's system can become unusable.

4. Software Characteristics of DDoS Attack Tools

All DDoS attack tools have common software characteristics. The way the agents are developed, the attackers, the handlers, the way the agents communicate, and the operating systems supported are all identical.

4.1 DDoS Agent Setup

Malware is actively and passively loaded on the second victim system to build her agent handler or IRC-based DDoS assault network. Attackers actively scan networks for known vulnerabilities. Attackers who find a vulnerable system use programs to hack the system. Once inside the system, the attacker can covertly install her DDoS agent application. Because of this, computers can be used as zombies in DDoS attacks. The attacker uploads a corrupted file or creates a website for her that exploits recognized vulnerabilities in the web browser of the second victim. Accessing the website or her file with the DDoS agent inserted compromises the second victim's system.

4.1.1 Active DDoS Installation

Port scan: Attackers must first establish a DDoS attack network. They often check for potential secondary victim systems. An attacker's favorite tool for searching for ports is Nmap. Nmapmay be obtained from many online sources. This application allows attackers to scan IP ranges. The application will then search the Internet for all of these IP addresses. An IP address communicates information like open UDP and TCP ports and the scanning system's OS [16]. An attacker may then search this list for more victim systems. Another penetration testing tool finds vulnerable IP addresses. This offers the attacker a list of susceptible systems. One such tool is Nessus [17]. Following are three examples of active DDoS agent exploited vulnerabilities.

Software Vulnerability: To install the DDoS agent malware on the secondary target PC, the attacker must first search for vulnerable machines, for example, the public list of all known system vulnerabilities from the Common Flaws and Exposures (CVE) group. CVE has classified more than 2,000 vulnerabilities, and 2,000 more will be evaluated [18]. This survey data not only helps network administrators defend their networks but also informs attackers about vulnerabilities. CERT first reported one of these bugs in November 2001. This bug was identified when Kaiten's known default password was used to deploy his IRC-based DDoS agent on Microsoft SQL Server. An attacker could use TCP port 1433 (MS SQL Server port) to locate the host and use the default administrator password to continue. An attacker could use her xp-cmd shell method [19] to download the MS SQL Server Agent software. You are at risk if your MS SQL Server is not patched to prevent the use of default administrator passwords.

Buffer Overflows: Another problem is buffer overflows. A buffer is a limited-capacity section of memory that a computer uses to temporarily store data. A buffer overflow occurs when more data is pushed into a buffer than it could hold. This overwrites data on the memory stack next to the buffer [21], like the return address of the procedure. This could allow the computer to return malicious code in the data from the procedure call and overwrite the buffer. Either the attacker launches software of their choice (like a DDoS agent) or gains access to the victim's computer to install her DDoS agent her malware.

4.1.2 Passive DDoS Installation

Adding Malicious Content to Your Website: An attacker could take advantage of a vulnerability in her web browser to passively compromise a secondary target in her computer system.

Attackers can use this method to create websites that contain code or instructions to catch victims. When the victim's web browser accesses or attempts to access her website, it installs or downloads malicious software (such as a DDoS agent). This instance is a bug in Microsoft's Internet Explorer 5.5 & 6.0. ActiveX allows her IE to control certain plugin programs in the website code. When you use her ActiveX component on your website, the Internet Explorer web browser can automatically download the client's binary code [22]. Her web page in an ActiveX control may contain malicious code. Instead of installing her software on the client, the attacker may use her ActiveX to download her DDoS agent. They normally post ActiveX-encoded malicious websites over the Internet to lure victims. Used to describe ActiveX installation packages that contain malicious code. B. DDoS agent software or malware that allows an attacker to penetrate your system. This tells Internet Explorer to use the ActiveX control with GUID "xxxx". The attacker injects the download URL of the malicious software into the download URL.

If the Internet Explorer software flaw isn't fixed, Internet Explorer will download malicious code instead of legitimate code. The attacker has now installed malware on the victim's PC. If this malware is a DDoS agent, the attacker has built a backup victim system.

Corrupted File: Passive attacks frequently include modifying files and inserting malicious code into them. Attempting to read or execute these files infects the target system. Infected files can be created in many ways. Most attackers can embed DDoS attack agents and other malicious software in legitimate files. The attackers change the desktop icons of these files using long filenames interspersed with the actual extension, so even a partial display of filenames looks real. An attacker can create a text file containing executable binary code for a DDoS agent. Renames the text file to a long name that includes the extension ".txt", but the actual extension is ".exe". For example, newfile.txt is ddos agent.exe.

If the user only sees the first few characters, the file looks like a text file rather than an executable file. In this situation, the new filename should be approximately 150 characters long [24]. When the user runs the file, the DDoS agent will be installed. Some attackers add an opening text box to trick victims into believing that the file is genuine and the DDoS agent is unaware of it.

Corrupted files can spread in several ways. Gnutella and IRC networks are 2 popular file-sharing systems that make it easy to transmit a corrupted file to numerous people. They may even send victims emails with faulty files, hoping they would open them and infect them with DDoS agent code.

.**Rootkits**

After installing handler or agent software, an attacker utilizes rootkits to erase log files and other evidence of exploiting the system [25]. Attackers could also employ rootkit tools to add "back-doors" to the systems of secondary victims [26]. Since the operation of DDoS defense, the network requires a handler, and receiver programs are commonly deployed within ISPs or data centers, rootkit tools are often used when the handler software is loaded. However, using rootkits to remove all agents is cumbersome and ineffective.

4.2 Attacks Network Communication

Attacks Network communication using the protocol:DDoS controllers and handlers can communicate over TCP, UDP, or ICMP. Clients and DDoS defenders can share protocols.

Encrypted Communications: Some DDoS attack techniques offer encrypted transmissions in the DDoS attack network.

Agent Handler DDoS attacks involve encrypted connections between clients and handlers or operators and agents. The DDoS tool's communication protocol determines her Agent Handler's DDoS encryption technique. IRC-based DDoS operations use public, personal, or secret channels to communicate between handlers and agents. Both secret and private IRC channels offer encryption, but only private channels appear in his IRC client's channel list.

Third-Party Agent Activation: The DDoS Agent can be activated in two ways. Some DDoS systems actively look for instructions in handlers or IRC channels, while others just lie around and wait.

4.3 OS Support

DDoS attack tools typically support many operating systems (OS). You can develop any DDoS agent or handler code. Handler code is often written to support operating systems running on corporate or ISP servers or workstations. This typically selects Unix, Solaris, or Linux as the operating system. It is also frequent for agent code to run on Solaris or Linux in addition to Windows. Attackers often use Windows to target their DSL and cable users (to increase the throughput of their attacks).

5. Examples of DDoS Attack Tools

Several DDoS attack techniques are currently available. Here are some of the most common DDoS attack tools: Attackers' use and customization of these tools have resulted in several derivative DDoS attacks based on previous tools and underlying strategies.

5.1 Agent-Handler Attack Tools

[3, 25] was the first widely used DDoS attack software. To exhaust the bandwidth, Trin00 uses a UDP flood attack and an intelligence officer attack architecture. Early trin00 versions do not seem to allow source IP address spoofing. The trin00 agent is typically used on systems vulnerable to remote buffer overflow attacks [25]. It is possible to remotely design and execute agent deployments on the primary victim's system buffers. Early trin00 versions were observed in “Red Hat Linux 6.0” and Solaris 2.5.1 [25]. We typically use TCP between the attacker's browser and the handler computer, andhere we use UDP between the manager and agent computers [25]. The trin00 application uses encrypted symmetric key relationships between users and handlers.

TheTribe Flood Network is a DDoS attack technique that attackers can use to consume bandwidth and resources. TFN supports ICMP and UDP flooding and smurf and TCP SYN attacks [27]. A buffer overflow was detected in the TFN setup [28]. TFN attack handlers and agents use "ICMP ECHO REPLY" packets

These packets can sneak firewalls and are more difficult to recognize than UDP traffic [27]. A TFN attack tool was found in Red Hat Linux 6.0 and Solaris 2. x. TFN does not encrypt communications between agents and managers or between dealers and customers [28]. The TFN2K DDoS attack tool is based on the design of the Tribal Flood Network. The TFN2K attack tool contains encrypted messages [27]. Handlers and agents can communicate via UDP, ICMP, or TCP [27]. There is also a randomized protocol selection option.

A DDoS attack tool based on his previous TFN version is called Barbed Wire. Similar to TFN [29], it supports ICMP, UDP, and TCP SYN attacks. You can also update agents automatically [29]. As a result, an attacker could publish installation files to an anonymous server, and the agent would automatically check for and install updates on each agent's system. Barbed wire also secures the Telnet connection between the attacker & handler computer [29]. System administrators cannot intercept or detect this transmission

A shaft is a tool developed from trin00. Communicate with handlers and agents using UDP. Telnet is used by the attacker to connect to the handler. The shaft can flood UDP, TCP, and ICMP. Attacks can be performed alone or in combination with UDP/TCP/ICMP flooding. Schacht describes a flood. An attacker can use these statistics to determine when a target system has been completely shut down and how to stop zombies from using her machine. [30].

5.2 IRC-based DDoS Attack Tools

Next came IRC-based DDoS attack tools. Therefore, many IRC-based attack tools are more complex than Agent Handler attack tools. An IRC-based DDoS assault tool is called Trinity. Trinity can send ACK, TCP SYN, UDP, and NUL packet floods, TCP fragments, random flags, RST, and established floods. It can generate random 32-bit IP addresses [31]. Additionally, Triune can generate random TCP flood packets. Trinity now has access to additional TCP-based attacks. A DDoS assault tool called Knight was initially discovered in July 2001 [31]. It uses IRC. The Knight DDoS attack tool [32] includes UDP floods, SYN attacks, and an urgent pointer flooder. Back Orifice [31] is often used to install the Knight tool. Knight is a Windows game. Kaiten is another DDoS technique that uses IRC. According to Knight, it was first launched in August 2001[19]. Kaiten may attack in a variety of ways. A PUSH + ACK attack is provided [33]. Kaiten also randomly assigns the source location' 32 bits.

6. Taxonomy of DDoS Countermeasures

There are now several methods and means of short for reducing the impact of a DDoS attack. That most of these ideas and methods may help prevent DDoS attacks. But no one method exists to protect against all known DDoS attacks. Moreover, attackers continuously invent new DDoS tactics to circumvent current countermeasures. More research is required to generate more efficient and complete treatments

The purpose of this research is to better understandDDoS attack networks, attack methods, and malware attack tools. DDoS defense consists of three parts. To combat a DDoS attack, you need to identify and

stop handlers and avoid subsequent victims. DDoS attack detection or prevention, mitigation or termination, and redirection are all components. Finally, there is post-attack network forensics.

6.1 Secondary Victim Prevention

End Users: One of the best strategies for preventing DDoS attacks is to deny attacks to secondary victim systems. It's time to educate everyone about security threats and how to prevent them. Without connecting to and using secondary target systems, attackers have no "DDoS attack infrastructure" in which they can launch DDoS attacks. Users of these technologies should regularly monitor their security to avoid becoming secondary victims of DDoS attacks. Additionally, you must ensure that no traffic is delivered to your network by a DDoS attacker. The decentralization of the Internet and the variety of hardware and software components make it difficult for ordinary users to take the necessary precautions. This usually includes updating antivirus and anti-trojan software. Furthermore, all software updates for identified vulnerabilities should be installed. Recent research suggests that there are mechanisms built into PC hardware and software that can protect against the introduction of exploit code such as:

This makes the system less likely to be used later as a target for DDoS attack networks.

Network Service Providers: Providers and network administrators may apply a fee structure to network use to encourage additional victims to aggressively preventDDoS attacks. Providers can charge for specific services within their existing network if they want to charge different rates for different resources. This allows service providers to restrict network access to legitimate users. This method can be used to keep enemies away from the system [35]. If additional victims are paid to use the internet, they will become more aware of the frequencies they are using and better protect themselves to avoid being part of a DDoS attack.

6.2 Handler Identification and Neutralization

Handlers must be identified and neutralized to prevent DDoS attacks. Identifying and removing handlers is an easy way to stop a DDoS attack infrastructure because they act as an attacker's man-in-the-middle. Examine communications and vehicle traffic between handlers and customers or between controllers and agents to confirm handler infection. Also, since there are more agents than DDoS handlers, neutralizing a single handler can disable multiple agents and stop a DDoS attack.

6.3 Identifying Potential Attacks

Egress Filtering: Egress filtering is an approach for identifying potential attacks. Egress filtering is the process of examining the packet headers of IP packets leaving a network (egress packets) to ensure that they satisfy specific requirements. Packets are routed outside the subnet of origin if the requirements are met

The packet will not be sent to its designated location if the filter requirements are not satisfied. Since spoofed IP addresses are one component of DDoS attacks, likely, the spoofed source address of a DDoS attack packet is not the legitimate source address of a particular subnet. Numerous DDoS packets with fake IP source addresses are disregarded and neutralized when a network administrator sets up a packet sniffer or firewall on a subnet to filter out all traffic without an initiating IP address from that subnet.

MIB Statistics: Also examines Information Management Base data from routers to identify DDoS attacks. His MIB data on the router shows various packet and routing information. DDoS attacks should

be continuously investigated to identify statistical patterns in various indicators [36]. It seems to correlate statistical anomalies in TCP dump, UDP, and HTTPS packets with certain DDoS attacks. The ability of an accurate statistical approach to monitor her DDoS defense traffic and predict attacks relying on her MIB capabilities in routers is currently being tested. This research could help identify DDoS attacks and change prepared networks.

6.4 Mitigate or Stop the Impact of DDoS Attacks

Load Balancing: Network providers use many techniques to mitigate DDoS attacks. In the event of an attack, suppliers can increase bandwidth on critical lines to avoid outages. If some servers go down during a DDoS attack, replicating servers can provide resilient protection. In a multiple-server setup, balancing the demands of each computer can improve normal effectiveness and reduce DDoS attacks.

Throttling: A Max-Min Implementation Fair server-centric router throttling [37] is a proposed method. This method configures the router to scale (throttle) incoming traffic to the server's capacity. This will prevent server overload. This strategy can be improved by limiting the bandwidth of DDoS attacks to legitimate user traffic. While this method is still in its infancy, network providers are implementing similar throttling tactics. Throttling becomes difficult because there is no way to distinguish between legitimate and illegitimate traffic. Legitimate traffic could be delayed or dropped, while malicious traffic may be allowed through.

Drop Requests: Requests can be dropped as soon as the load increases. Don't worry. Alternatively, you can force the requesting system to give up by asking for difficult tasks that require significant computing power or memory to be solved first. Users of zombie computers may notice a decrease in productivity and choose to stop sending DDoS attack packets.

6.5 Repel Attacks

Honeypots: Honeypots are another domain of research. Honeypots are insecure systems that lure attackers to attack the honeypot rather than the primary system. Honeypots are effective in gathering information about attackers, recording their activity, and identifying attacks and the software tools they use. Honeypots that impersonate all network components (web servers, mail servers, clients, etc.) are increasingly being used to lure DDoS attackers [38]. The purpose of this honeypot is to trick DDoS attackers into installing agent or handler code. This allows the honeypot owner to observe the behavior of her agent or handler and learn how to protect against her future DDoS assaults.

6.6 Post-Attack Forensics

Analyzing Network Data Patterns: Additional methods help detect malicious connections using packet traces [39] Tracking is a term that refers to tracing internet traffic back to its source. This allows you to trace network traffic and identify attackers. Additionally, if the attacker is broadcasting a variety of attack traffic, this strategy can help send information to the victim's system that can be used to build filters to stop the attack. [40] presented a methodology that can detect high network volumes and monitor users' traffic in the network. This method is especially useful in controlled network environments such as B. A corporate network where a central system administrator can track the activities of individual end-her users. This strategy tends to fail as the network spreads [40]. Tracing traffic on the Internet or a huge extranet is difficult. It might be difficult to ascertain who is in charge of

traffic monitoring since different network administrators assess various aspects of the Internet. Also, most internet users react negatively to the loss of privacy on the internet.

Event Log. Network administrators can keep records of DDoS attack data for forensic investigations and assist law enforcement agencies when attackers cause significant economic damage. Vendors can use botnets and other network equipment like packet sniffers, server logs, and to firewalls store events from attack configuration and implementation. This allows network administrators to determine which DDoS attacks have been deployed.

7. Conclusion and Future Work

The detection of DDoS attacks and the observation of several self-protection approaches that are fundamentally being researched or deployed can lead to different conclusions. A DDoS attack is a sophisticated means of attacking Internet systems to make them inaccessible to legitimate users.

These assaults, which target essential systems, are at the very least inconvenient and have the potential to be devastating. Loss of system assets and money can cause operational delays and prevent network users from communicating with each other. Distributed DoS processes have detrimental consequences, so it is imperative to have solutions and security measures in place to avoid this type of attack. Hacking and jamming of network traffic have become more common over the past decade as networks have become more accessible and used by more people, businesses, and government agencies. As a result, attack tools have advanced but have become simpler to utilize."[41]. Attackers and script kiddies can easily be DDoS- attacked, and like the current attack against 13 root servers, the potential for additional attacks is huge. Understanding how to avoid and stop DDoS attacks is critical to national security. The first step in this way and the primary contribution of this whitepaper is understandingDDoS assaults and tools.

Cyberattacks such as DDoS are also investigating new legal issues. Victims of attacks are rarely linked to perpetrators, so it is unclear who else could be held liable for contributory negligence. The cost of solutions and preventive measures is among the most significant factors impacting the implementation of DDoS mitigation. If his DDoS protection technology is too expensive for businesses and individuals, these systems will not be quickly or widely deployed. Industry &govt agencies are slow to buy new items. Moreover, attackers develop ways to circumvent certain security measures.

As a result, new security mechanisms are implemented and new attacks are developed in cycles. Attacks are a more complete solution that may protect against both known & unknown versions. This study attempted to define the problem of DDoS by defining DDoS attack networks, attack methods, and attack tools. This helps us consider a more complete, multi-layered approach to countermeasures rather than building countermeasures that are specific to a particular attack. For future work, we would like to create a simulator that allows parametrizable and accurate modeling of DDoS attacks. I also want to create a DDoS behavioral analysis model. These simulation and modeling methods may be utilized to develop new countermeasures and more complete solutions.

8. References

1. David Karig and Ruby Lee, “Remote Denial of Service Attacks and Countermeasures,” Princeton University Department of Electrical Engineering Technical Report CE-L2001-002, October 2001.

2. Lincoln Stein and John N. Stuart. "The World Wide Web Security FAQ", Version 3.1.2, February 4, 2002. http://www.w3.org/security/faq/ (8 April 2003).
3. Paul J. Criscuolo. "Distributed Denial of Service Trin00, Tribe Flood Network, Tribe Flood Network 2000, AndStacheldraht CIAC-2319". Department of Energy Computer Incident Advisory Capability (CIAC), UCRL-ID-136939, Rev. 1., Lawrence Livermore National Laboratory, February 14, 2000.
4. "Yahoo on Trail of Site Hackers", Wired.com, February 8, 2000. http://www.wired.com/news/business/0,1367,34221,00.html (15 May 2003).
5. "Powerful Attack Cripples Internet". Associated Press for Fox News 23 October 2002. http://www.foxnews.com/story/0,2933,66438,00.html. (9 April 2003).
6. Joseph Lo and Others. "An IRC Tutorial", irchelp.com. 1997. http://www.irchelp.org/irchelp/irctutorial.html#part1. (8 April 2003).
7. Nicolas Pioch. "A Short IRC Primer". Edition 1.2, January 1997. http://www.irchelp.org/irchelp/ircprimer.html#DDC. (21 April 2003).
8. Kleinpaste, Karl, Mauri Haikola, and Carlo Kid. "The Original IRC Manual". March 18, 1997. http://www.user-com.undernet.org/documents/irc-manual.html#seen (21 April 2003).
9. Kevin J. Houle. "Trends in Denial of Service Attack Technology". CERT Coordination Center, Carnegie Mellon Software Engineering Institute. October 2001. www.nanog.org/mtg-0110/ppt/houle.ppt. (14 March 2003).
10. TFreak. "smurf.c", www.phreak.org. October 1997. http://www.phreak.org/archives/exploits/denial/smurf.c (6 May 2003).
11. Federal Computer Incident Response Center (FedCIRC), "Defense Tactics for Distributed Denial of Service Attacks". Federal Computer Incident Response Center. Washington, DC, 2000.
12. TFreak. "fraggle.c", www.phreak.org. http://www.phreak.org/archives/exploits/denial/fraggle.c (6 May 2003).
13. Martin, Michael J., "Router Expert: Smurf/Fraggle Attack Defense Using SACLS", Networking Tips and Newsletters, www.searchnetwork.techtarget.com. October 2002. http://searchnetworking.techtarget.com/tip/1,289483,sid7_gci856112,00.html (6 May 2003).
14. Chen, Y. W. "Study on the Prevention of SYN Flooding by Using Traffic Policing", Network Operations and Management Symposium, 2000. NOMS 2000. 2000 IEEE/IFIP, pp. 593-604. 2000.
15. RFC 793, "Transmission Control Protocol DARPA Internet Program Protocol Specification". Arlington, Virginia. September 1981.
16. "Nmap Stealth Port Scanner Introduction", Insecure.org. August 2002. http://www.insecure.org/nmap/. (8 April 2003).
17. "Nessus Documentation", Nessus. 2002. http://www.nessus.org/. (8 April 2003).
18. "CVE (version 20020625)", Common Vulnerabilities and Exposures. March 27, 2002. http://cve.mitre.org/cve/. (9 April 2003).
19. "CERT® Incident Note IN-2001-13". CERT Coordination Center, Carnegie Mellon Software Engineering Institute. November 27, 2001. http://www.cert.org/advisories/CA-2001-20.html. (14 March 2003).
20. Colon E. Pelaez and John Bowles, "Computer Viruses", System Theory, 1991, Twenty-Third Southeastern Symposium, pp. 513-517, Mar 1999.

21. Crispin Cowan, Perry Wagle, Calton Pu, Steve Beattie, and Jonathan Walpole, "Buffer Overflows: Attacks and Defenses for the Vulnerability of the Decade", DARPA Information Survivability Conference and Exposition, 2000. Vol. 2, pp. 119-129, 2000.
22. Microsoft. "How to Write Active X Controls for Microsoft Windows CE2.1", Microsoft Corporation. June 1999. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnce21/html/activexce.asp. (5 April 2003).
23. "Executing arbitrary commands using ActiveX "codebase=" parameter", EdenSoft™, 2 April 2002. http://www.edensoft.com/exploit.html. (9 April 2003).
24. DanchoDanchev. "The Complete Windows Trojans Paper", BCVG Network Security. October 22, 2002. http://www.ebcvg.com/articles.php?id=91. (9 April 2003).
25. David Dittrich. "The DoS Project's "trinoo" Distributed Denial of Service Attack Tool". University of Washington, October 21, 1999. http://staff.washington.edu/dittrich/misc/trinoo.analysis.txt (8 April 2003).
26. Alex Noordergraff. "How Hackers Do It: Tricks, Tools, and Techniques", Sun BluePrints™ OnLine. Part No.: 816-4816-10, Revision 1.0. May 2002. http://www.sun.com/solutions/blueprints/0502/816-4816-10.pdf. (8 April 2003).
27. Frank Kargl, Joern Maier, and Michael Weber, "Protecting Web Servers from Distributed Denial of Service Attacks", Proceedings of the Tenth International Conference on World Wide Web, April 2001.
28. David Dittrich. "The "Tribe Flood Network" Distributed Denial of Service Attack Tool". University of Washington, October 21, 1999. http://staff.washington.edu/dittrich/misc/trinoo.analysis.txt (8 April 2003).
29. David Dittrich. "The "stacheldraht" Distributed Denial of Service Attack Tool". University of Washington, December 31, 1999. http://staff.washington.edu/dittrich/misc/stacheldraht.analysis.txt (8 April 2003).
30. Sven Dietrich, Neil Long, and David Dittrich, "Analyzing Distributed Denial of Service Tools: The Shaft Case", USENIX Association, Proceedings of the 14th Systems Administration Conference (LISA 2000), New Orleans, Louisiana, December 2000.
31. "CERT® Advisory CA-2001-20 Continuing Threats to Home Users", CERT Coordination Center, Carnegie Mellon Software Engineering Institute. July 23, 2001. http://www.cert.org/advisories/CA-2001-20.html. (14 March 2003).
32. Bysin. "Knight.cSourcecode", PacketStormSecurity.nl. July 11, 2001. http://packetstormsecurity.nl/distributed/knight.c. (18 March 2003).
33. Contem. "kaiten.cSourcecode", PacketStormSecurity.nl. December 2001. http://packetstormsecurity.nl/irc/indexsize.shtml. (8 April 2003).
34. Ruby Lee, David Karig, Patrick McGregor and Zhijie Shi, "Enlisting Hardware Architecture to Thwart Malicious Code Injection", Proceedings of the International Conference on Security in Pervasive Computing (SPC-2003), pp. N/A, March 2003.
35. David Mankins, Rajesh Krishnan, Ceilyn Boyd, John Zao, and Michael Frentz, "Mitigating Distributed Denial of Service Attacks with Dynamic Resource Pricing", Computer Security Applications Conference, 2001. ACSAC 2001. Proceedings 17th Annual, pp. 411-421, 2001.
36. Joao B. D. Cabrera, Lundy Lewis, Xinzhou Qin, Wenke Lee, Ravi K. Prasanth, B. Ravichandran, and Ramon K. Mehra, "Proactive Detection of Distributed Denial of Service Attacks Using MIB

Traffic Variables – A Feasibility Study", Integrated Network Management Proceedings, pp. 609-622, 2001.

37. David K. Yau, John C. S. Lui, and Feng Liang, "Defending Against Distributed Denial of Service Attacks with Max-min Fair Server-centric Router Throttles", Quality of Service, 2002 Tenth IEEE International Workshop, pp. 35-44, 2002.
38. Nathalie Weiler. "Honeypots for Distributed Denial of Service", Enabling Technologies: Infrastructure for Collaborative Enterprises, 2002. WET ICE 2002. Proceedings. Eleventh IEEE International Workshops, 2002. pp. 109-114. 2002.
39. Vern Paxon, "An Analysis of Using Reflectors for Distributed Denial of Service Attacks", ACM SIGCOMM Computer Communication Review, Vol. 31, Iss. 3, July 2001.
40. Thomas E. Daniels and Eugene H. Spafford, "Network Traffic Tracking Systems: Folly in the Large?", Proceedings of the 2000 Workshop on New Security Paradigms, February 2001.
41. Freeh, Louis J. "Statement for the Record of Louis J. Freeh, Director Federal Bureau of Investigation on Cybercrime Before the Senate Committee on Judiciary Subcommittee for the Technology, Terrorism, and Government Information." Washington, D.C., March 28, 2000.