A SEMINAR

ON

DISTRIBUTED DENIAL OF SERVICE ATTACK

BY

NAME NAME NAME

MATRIC NO: 00/00/0000

A SEMINAR SUBMITTED TO THE
DEPARTMENT OF COMPUTER SCIENCE
MOSHOOD ABIOLA POLYTECHNIC, ABEOKUTA.
IN PARTIAL FULFILLMENT OF THE AWARD OF
HIGHER NATIONAL DIPLOMA IN COMPUTER SCIENCE

SUPERVISOR:
TITLE NAME

MONTH, 0000

ABSTRACT

With the dramatic evolution in networks nowadays, an equivalent growth of challenges has been depicted toward implementing and deployment of such networks. One of the serious challenges is the security where wide range of attacks would threat these networks. Denial-of-Service (DoS) is one of the common attacks that targets several types of networks in which a huge amount of information is being flooded into a specific server for the purpose of turning of such server. Many research studies have examined the simulation of networks in order to observe the behavior of DoS. However, the variety of its types hinders the process of configuring the DoS attacks. In particular, the Distributed DoS (DDoS) is considered to be the most challenging threat to various networks. Hence, this paper aims to accommodate a comprehensive simulation in order to figure out and detect DDoS attacks. Using the well-known simulator technique of NS-2, the experiments showed that different types of DDoS have been characterized, examined and detected. This implies the efficacy of the comprehensive simulation proposed by this study.

TABLE OF CONTENT

CHAPTER ONE
INTRODUCTION

INTRODUCTION

Distributed denial of service (DDoS) is one of the common attacks within wide range of networks where the recognition and prevention of such attack has always been a hot issue in network security research. DDoS detection and defense systems have many shortcomings such as high false positive rate, low execution efficiency, and lack of linkage between detection and defense. Therefore, eliminating false positives, improving execution efficiency, and enhancing the linkage between detection and defense processes have always been the focuses of research. With the diversity and different characteristics of DoS, the process of detecting such attack is still facing obstacles. Şimşek & Şentürk have proposed method that utilize the pre-congestion in order to analyze the flow of data during this period. The authors had an assumption that low-rate distributed DoS is one of the hardest to be detected due to their similarity to the normal behavior. Therefore, the authors have focused on the periods have no congestions in order to diagnose the features. The features extracted from such periods have been incorporated to form a new filtering approach for detecting DDoS attacks. Results of simulation showed fair progress on characterizing DDoS attacks. Bukharov et al. have proposed a game-based method for simulating DoS attacks. The proposed method has utilized a scenario where the intruder would be attracted in order to gain information regarding his real intentions. Results of simulation showed that the proposed method has the ability to detect wide range of DoS attacks. Wang et al. have proposed a DoS detection method based on honeynet technology. The proposed method was intended to observe and analyze the characteristics of every behavior in order to detect specific pattern. Finally, the proposed method aimed at detecting such patterns which might correspond to DoS attacks. Results of simulation showed progress on detecting DoS attacks. Mohd et al. have examined the distributed DoS that might occur on Internet of Things (IoT) networks. The authors have utilized OMNET++ in order to create a virtual environment that simulate the IoT networks. During such simulation, the authors have characterized several DDoS attacks. As depicted from the literature, it is obvious that the examination of DoS attacks is still a challenging task where wide range of such attack would be encountered especially with the variety of networks nowadays. Therefore, this paper aims to accommodate a comprehensive simulation to examine the types of DoS, as well as, attempting to detect these attacks.

INTRODUCTION OF DENIAL-OF-SERVICE (DoS) ATTACK

A **denial-of-service attack (DoS attack)** is an attempt to make a computer resource unavailable to its intended users. Although the means to carry out, motives for, and targets of a DoS attack may vary, it generally consists of the concerted efforts of a person, or multiple people to prevent an Internet site or service from functioning efficiently or at all, temporarily or indefinitely. Perpetrators of DoS attacks typically target sites or services hosted on high-profile web servers such as banks, credit card payment gateways, and even root name servers. The term is generally used relating to computer networks, but is not limited to this field; for example, it is also used in reference to CPU resource management.

One common method of attack involves saturating the target machine with external communications requests, such that it cannot respond to legitimate traffic, or responds so slowly as to be rendered effectively unavailable. Such attacks usually lead to a server overload. In general terms, DoS attacks are implemented by either forcing the targeted computer(s) to reset, or consuming its resources so that it can no longer provide its intended service or obstructing the communication media between the intended users and the victim so that they can no longer communicate adequately.

Denial-of-service attacks are considered violations of the IAB's Internet proper use policy, and also violate the acceptable use policies of virtually all Internet service providers.

INTRODUCTION OF DISTRIBUTED DENIAL-OF-SERVICE (DDoS) ATTACK

A **distributed denial-of-service (DDoS) attack** is one in which a multitude of compromised systems attack a single target, thereby causing denial of service for users of the targeted system. The flood of incoming messages to the target system essentially forces it to shut down, thereby denying service to the system to legitimate users.

In a typical DDoS attack, a hacker (or, if you prefer, cracker) begins by exploiting a vulnerability in one computer system and making it the DDoS master. It is from the master system that the intruder identifies and communicates with other systems that can be compromised. The intruder loads cracking tools available on the Internet on multiple sometimes thousands of compromised systems. With a single command, the intruder instructs the controlled machines to launch one of many flood attacks against a specified target. The inundation of packets to the target causes a denial of service.

While the press tends to focus on the target of DDoS attacks as the victim, in reality there are many victims in a DDoS attack -- the final target and as well the systems controlled by the intruder. Although the owners of co-opted computers are typically unaware that their computers have been compromised, they are nevertheless likely to suffer degradation of service and malfunction. Both owners and users of targeted sites are affected by a denial of service. Yahoo, Buy.com, RIAA and the United States Copyright Office are among the victims of DDoS attacks. DDoS attacks can also create more widespread disruption. In October 2010, for example, a massive DDoS attack took the entire country of Myanmar offline.

A computer under the control of an intruder is known as a zombie or bot. A group of co- opted computers is known as a botnet or a zombie army. Both Kaspersky Labs and Symantec have identified botnets -- not spam, viruses, or worms -- as the biggest threat to Internet security.

RESEARCH METHODOLOGY

One of the most serious problems is DDoS, and many defenses have been proposed to address this threat. In order to compare and evaluate these solutions, a common evaluation platform is required. The methodology of this paper consists of three parts: a typical attack scenario consisting of the dimensions of legitimate traffic and target network resources, testing the methodological criteria that capture performance metrics and are affected by the effectiveness of attacks and defenses it is composed. In order to do so, the following steps have been applied: Detect and filter one-way legitimate traffic from traffic identified as a possible attack. Detect attack using multiple detection criteria. It is legitimate from attack traffic. Finally, Attack samples from attack traffic, summarize attack functions in a readable format and machine-readable form, and facilitate the application of clustering methods. This makes it easy to collect attack samples from many public traces. All of these pastes are automated by a series of tools. Figure 1 shows applying the simulation process to attack cases requiring more attackers and usage scenarios. In our simulation methodology we follow these steps: First Step: In first step is to create a network topology with an NS-2 tell script for each attack. Second Step: In second step is to attach the legitimate traffic records to perform legitimate traffic on topology nodes. After that, real-time attack tracks are linked to topologies to generate attack traffic. These attack records are analyzed.

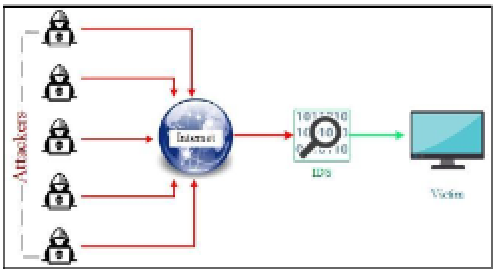


Figure 1. Framework of the proposed method

Then simulation is again performed, all traffic is monitored, and an offline analysis is performed. The output trace file is then used to measure the attack. The simulation topology used for this experiment contains a legitimate client pool containing various nodes that are used to generate legitimate traffic. To generate legitimate traffic, real-time tracks are used. With these traces, the nodes generate TCP traffic. An attacker used UDP traffic to launch an attack. The purpose of the attack is to consume the bandwidth of the bottleneck link so that legitimate traffic could not send the packets. Each simulation time is 2 seconds. Legitimate traffic is based on TCP, so it goes through the slow boot phase. The total number of legitimate clients in the legitimate client pool is 8. The total traffic load and bottleneck bandwidth represent the scenario of a busy connection. In our experiments, legitimate traffic is generated using real time tracks. The legitimate traffic is based on TCP. Here we have considered 13 legitimate clients that want to communicate with the TCP Sink node. Real-time data sets are again used to generate DDoS attacks. The amount and complexity of traffic in records is very high and very difficult to understand. The tracks used to create an attack are stored in tr format. Some results are simulated by gnu plot and other extracted information and then passed to excel to produce the graphical results.

HISTORY OF DISTRIBUTED DENIAL-OF-SERVICE (DDoS) ATTACK

Since the first DoS attack was launched in 1974, DDoS attacks and other DoS attacks have remained among the most persistent and damaging cyber-attacks. These attacks reflect hackers' frustratingly high levels of tenacity and creativity—and create complex and dynamic challenges for anyone responsible for cyber security.

THE EARLY DAYS

In 2014, the DoS attack celebrated its 40th birthday. Born as the handiwork of a teenaged "computer geek," these attacks have since exploded in quantity—and sophistication.

The first-ever DoS attack occurred in 1974 courtesy of David Dennis—a 13-year-old student at University High School, located across the street from the Computer-Based Education Research Laboratory (CERL) at the University of Illinois Urbana-Champaign. David recently learned about a new command that could be run on CERL's PLATO terminals. PLATO was one of the first computerized shared learning systems, and a forerunner of many future multi-user computing systems. Called "external" or "ext," the command was meant to allow for interaction with external devices connected to the terminals. However, when run on a terminal with no external devices attached it would cause the terminal to lock up—requiring a shutdown and power-on to regain functionality.

Curious to see what it would be like for a room full of users to be locked out at once, he wrote a program that would send the "ext" command to many PLATO terminals at the same time. Dennis went over to CERL and tested his program—, which succeeded in forcing all 31 users to power off at once. Eventually the acceptance of a remote "ext" command was switched off by default, fixing the problem.

During the mid to late 1990s, when Internet Relay Chat (IRC) was first becoming popular, some users fought for control of non-registered chat channels, where an administrative user would lose his or her powers if he or she logged off. This behavior led hackers to attempt to force users within a channel to all log out, so they could enter the channel alone and gain administrator privileges as the only user present. These "king of the hill" battles—in which users would attempt to take control of an IRC channel and hold it in the face of attacks from other hackers—were fought using very simple bandwidth-based DoS attacks and IRC chat floods.

DDOS ATTACKS SPREAD

One of the first large-scale DDoS attacks occurred in August 1999, when a hacker used a tool called “Trinoo” to disable the University of Minnesota’s computer network for more than two days. Trinoo consisted of a network of compromised machines called “Masters” and “Daemons,” allowing an attacker to send a DoS instruction to a few Masters, which then forwarded instructions to the hundreds of Daemons to commence a UDP flood against the target IP address. The tool made no effort to hide the Daemons’ IP addresses, so the owners of the attacking systems were contacted and had no idea that their systems had been compromised and were being used in a DDoS attack.

Other early tools include “Stacheldraht” (German for barbed wire), which could be remotely updated and support IP spoofing, along with “Shaft” and “Omega”, tools that could collect attack statistics from victims. Because hackers were able to get information about their DDoS attacks, they could better understand the effect of certain types of attacks, as well as receive notification when a DDoS attack was detected and stopped.

Once hackers began to focus on DDoS attacks, DoS attacks attracted public attention. The distributed nature of a DDoS attack makes it significantly more powerful, as well as harder to identify and block its source. With such a formidable weapon in their arsenals, hackers began to take on larger, more prominent targets using improved tools and methods.

By the new millennium, DDoS captured the public’s attention. In the year 2000, various businesses, financial institutions and government agencies were all brought down by DDoS attacks. Shortly after, DNS attacks began with all 13 of the Internet’s root domain name service (DNS) servers being attacked in 2002. DNS is an essential Internet service, as it translates host names in the form of uniform resource locators (URLs) into IP addresses. In effect, DNS is a phonebook maintaining a master list of all Internet addresses and their corresponding URLs. Without DNS, users would not be able to efficiently navigate the Internet, as visiting a website or contacting a specific device would require knowledge of its IP address.

FROM SCRIPT KIDDIES TO GEO-POLITICAL EVENTS

As attack technology has evolved, so, to have motivations and participants. Today, we no longer face teenage “computer geeks” or “script kiddies” testing the limits of what they can do. While they still exist, they are no longer alone. Recent years have brought a continuous increase in the number of DDoS attacks—fueled by changing, and increasingly complex, motivations.

ANATOMY OF TODAY’S HACKERS

Hacking used to require a distinct set of skills and capabilities. These days, DDoS attack services are bought and sold via marketplaces on the Clearnet and Darknet—a phenomenon that is closing the gap between skilled and amateur hackers and fueling an exponential increase in threats.

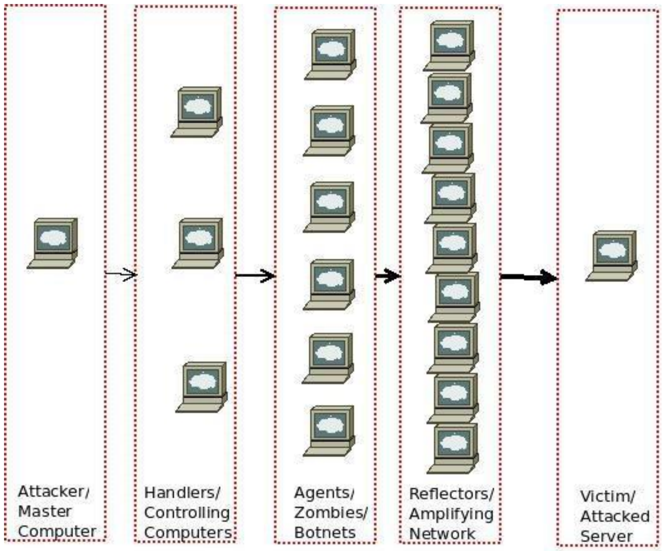
Thanks to the growing array of online marketplaces, it is now possible to wreak havoc even if you know virtually nothing about computer programming or networks. As attack tools and services become increasingly easy to access, the pool of possible attackers—and possible targets—is larger than ever. While many hacktivists still prefer to enlist their own digital “armies,” some are discovering that it is faster and easier to pay for DDoS-as-a-Service than to recruit members or build their own botnet. Highly skilled, financially motivated hackers can be invaluable resources to hacktivists seeking to take down a target.

By commoditizing hacktivist activities, hacking marketplaces have also kicked off a dangerous business trend. Vendors are now researching new methods of attack and incorporating more efficient and powerful vectors into their offerings. Already some of the marketplaces offer a rating system so users can provide feedback on the tools. Ultimately, this new economic system will reach a steady state—with quality and expertise rewarded with a premium.

CHAPTER THREE

DISCUSSION

COMPONENT AND ARCHITECTURE DIAGRAM OF DDoS ATTACK


Attacker/ Master Computer
Handlers/ Controlling Computers
Agents/ Zombies/ Botnets
Reflectors/ Amplifying Network
Victim/ Attacked Server

Distributed Denial Of Service Attacks (DDoS) - Architecture Diagram

As you can see in the above architecture diagram representing Distributed Denial of Service (DDoS) attacks, there may be up to five components. Two of them are aways there – The attacker/ master computer from where the attacks are initiated and the Victim/ Attacked server which comes under the attack. Presence of just these two components makes it a Denial-of-Service attack (DOS).

The three components in the middle, make it a Distributed Denial of Service attack! Zombies / botnets are the computers from which the DDoS attacks are carried out. Classification of DDoS

CLASSIFICATION OF DDoS ATTACKS

Classification by exploited vulnerability DDoS attacks according to the exploited vulnerability can be divided in the following categories flood attacks, amplification attacks, protocol exploit attacks and malformed packet attacks.

1. **Flood Attacks:** In a flood attack, the zombies send large volumes of IP traffic to a victim system in order to congest the victim systems bandwidth. The impact of packet streams sent

by the zombies to the victim system varies from slowing it down or crashing the system to saturation of the network bandwidth. Some of the well-known flood attacks are UDP flood attacks and ICMP flood attacks.

a) **UDP Flood Attacks:** A UDP Flood attack is possible when a large number of UDP packets is sent to a victim system. This has as a result the saturation of the network and the depletion of available bandwidth for legitimate service requests to the victim system. In a DDoS UDP Flood attack, the UDP packets are sent to either random or specified ports on the victim system. Typically, UDP flood attacks are designed to attack random victim ports. 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 of the victim, the system will go down. By the use of a DDoS tool the source IP address of the attacking packets can be spoofed and this way the true identity of the secondary victims is prevented from exposure and the return packets from the victim system are not sent back to the zombies.

b) **ICMP Flood Attacks:** ICMP Flood attacks exploit the Internet Control Message Protocol (ICMP), which enables users to send an echo packet to a remote host to check whether it’s alive. More specifically during a DDoS ICMP flood attack the agents send large volumes of ICMP_ECHO_ REPLY packets (‘‘ping’’) to the victim. These packets request reply from the victim and this has as a result the saturation of the bandwidth of the victim’s network connection. During an ICMP flood attack the source IP address may be spoofed.

2. **Amplification Attacks:** In amplification attacks the attacker or the agents exploit the broadcast IP address feature found on most routers to amplify and reflect the attack and send messages to a broadcast IP address. This instructs the routers servicing the packets within the network to send them to all the IP addresses within the broadcast address range. This way the malicious traffic that is produced reduces the victim systems bandwidth. In this type of DDoS attack, the attacker can send the broadcast message directly, or by the use of agents to send the broadcast message in order to increase the volume of attacking traffic. If the broadcast message is sent directly, the attacker can use the systems within the broadcast network as agents without

needing to infiltrate them or install any agent software. Some well-known amplification attacks, are Smurf and Fraggle attacks.

a) **Smurf Attacks:** Smurf attacks send ICMP echo request traffic with a spoofed source address of the target victim to a number of IP broadcast addresses. Most hosts on an IP network will accept ICMP echo requests and reply to the source address, in this case, the target victim. On a broadcast network, there could potentially be hundreds of machines to reply to each ICMP packet. The use of a network in order to elicit many responses to a single packet has been labeled as ‘‘amplifier’’. In this type of attack the party that is hurt is not only the spoofed source address target (the victim) but also, he intermediate broadcast devices (amplifiers).

b) **Fraggle Attacks:** The Fraggle attacks are a similar attack to the Smurf except that they use UDP echo packets instead of ICMP echoes. Fraggle attacks generate worse traffic and can create even more damaging effects than just a Smurf attack.

3. **Prototype Exploit Attacks:** Protocol exploit attacks exploit a specific feature or implementation bug of some protocol installed at the victim in order to consume excess amounts of its resources. A representative example of protocol exploit attacks is TCP SYN attacks.

a) **TCP SYN Attacks:** TCP SYN attacks exploit the inherent weakness of the three-way handshake involved in the TCP connection setup. A server, upon receiving an initial SYN (synchronize/start) request from a client, sends back a SYN/ACK (synchronize/acknowledge) packet and waits for the client to send the final ACK (acknowledge). An attacker initiates an SYN flooding attack by sending a large number of SYN packets and never acknowledges any of the replies, essentially leaving the server waiting for the nonexistent ACKs. Considering that the server only has a limited buffer queue for new connections, SYN Flood results in the server being unable to process other incoming connections as the queue gets overloaded.

4. **Malformed Packet Attacks:** Malformed packet attacks rely on incorrectly formed IP packets that are sent from agents to the victim in order to crash the victim system. The malformed packet attacks can be divided in two types of attacks: IP address attack and IP packet options attack. In an IP address `attack, the packet contains the same source and destination IP addresses. This has as a result the confusion of the operating system of the victim system and the crash of the victim system. In an IP packet options attack, a malformed packet may randomize the optional fields within an IP packet and set all quality-of-service bits to one. This

would have as a result the use of additional processing time by the victim in order to analyze the traffic. If this attack is combined with the use of multiple agents, it could lead to the crash of the victim system.

CLASSIFICATION OF DDoS DEFENSE

We may classify DDoS defense mechanisms using two different criteria. The first classification categorizes the DDoS defense mechanisms according to the activity deployed. Thus, we have the following four categories:

1. Intrusion Prevention

The best mitigation strategy against any attack is to completely prevent the attack. In this stage we try to stop DDoS attacks from being launched in the first place. There are many DDoS defense mechanisms that try to prevent systems from attacks Using globally coordinated filters, attacking packets can be stopped, before they aggregate to lethal proportions. Filtering mechanisms can be divided into the following categories: Ingress filtering is an approach to set up a router such that to disallow incoming packets with illegitimate source addresses into the network. Ingress filtering, proposed by Ferguson and Senie, is a restrictive mechanism to drop traffic with IP address that does not match a domain prefix connected to the ingress router. This mechanism can drastically reduce the DoS attack by IP spoofing if all domains use it.

Sometimes legitimate traffic can be discarded by an ingress filtering when Mobile IP is used to attach a mobile node to a foreign network Egress filtering is an outbound filter, which ensures that only assigned or allocated IP address space leaves the network. Egress filters do not help to save resource wastage of the domain where the packet is originated but it protects other domains from possible attacks. Besides the placement issue, both ingress and egress filters have similar behavior. Route-based distributed packet filtering has been proposed by Park and Lee. This approach is capable of filtering out a large portion of spoofed IP packets and preventing attack packets from reaching their targets as well as to help in IP traceback. Route-based filters use the route information to filter out spoofed IP packets, making this their main difference from ingress filtering. If route- based filters are partially deployed, a synergistic filtering effect is possible, so that spoofed IP flows are prevented from reaching other Autonomous Systems.

The main disadvantage of this approach is that it requires global knowledge of the network topology leading to scalability issues. History-based IP filtering (HIP) is another filtering mechanism that has been proposed by Peng et al. in order to prevent DDoS attacks. According to this approach the edge router admits the incoming packets according to a pre-built IP address database. The IP address database is based on the edge routers previous connection history. This scheme is robust, does not need the cooperation of the whole Internet community, is applicable to a wide variety of traffic types and requires little configuration. On the other hand, if the attackers know that the IP packet filter is based on previous connections, they could mislead the server to be included in the IP address database. This can be prevented by increasing the period over which IP addresses must appear in order to be considered frequent. Secure Overlay Services (SOS) is an architecture in which only packets coming from a small number of nodes, called servlets, are assumed to be legitimate client traffic that can reach the servlets through hash-based routing inside an overlay network. All other requests are filtered by the overlay. In order to gain access to the overlay network, a client has to authenticate itself with one of the replicated access points (SOAPs). SOS is a distributed system that offers excellent protection to the specified target at the cost of modifying client systems, thus it is not suitable for protection of public servers.

2. Intrusion Detection

Intrusion detection has been a very active research area. By performing intrusion detection, a host computer and a network can guard themselves against being a source of network attack as well as being a victim of a DDoS attack. Intrusion detection systems detect DDoS attacks either by using the database of known signatures or by recognizing anomalies in system behaviors. Anomaly detection relies on detecting behaviors that are abnormal with respect to some normal standard. Many anomaly detection systems and approaches have been developed to detect the faint signs of DDoS attacks.

A scalable network monitoring system called NOMAD has been designed by Talpade et al. This system is able to detect network anomalies by making statistical analysis of IP packet header information. It can be used for detecting the anomalies of the local network traffic and does not support a method for creating the classifier for the high-bandwidth traffic aggregate from distributed sources.

Another detection method of DDoS attacks uses the Management Information Base (MIB) data from routers. The MIB data from a router includes parameters that indicate different packet and routing statistics. Cabrera et al. has focused on identifying statistical patterns in different parameters, in order to achieve the early detection of DDoS attacks. It looks promising for possibly mapping ICMP, UDP and TCP packet statistical abnormalities to specific DDoS attacks. Although, this approach can be effective for controlled traffic loads, it needs to be further evaluated in a real network environment. This research area could provide important information and methods that can be used in the identification and filtering of DDoS attacks. A mechanism called congestion triggered packet sampling and filtering has been proposed by Huang and Pullen. According to this approach, a subset of dropped packets due to congestion is selected for statistical analysis. If an anomaly is indicated by the statistical results, a signal is sent to the router to filter the malicious packets.

Mirkovic et al. proposed a system called DWARD that does DDoS attack detection at the source based on the idea that DDoS attacks should be stopped as close to the sources as possible. D-WARD is installed at the edge routers of a network and monitors the traffic being sent to and from the hosts in its interior. If an asymmetry in the packet rates generated by an internal host is noticed, D-WARD rate limits the packet rate. The drawback of this approach is that there is a possibility of numerous false positives while detecting DDoS conditions near the source, because of the symmetry that there might be in the packet rates for a short duration. Furthermore, some legitimate flows like real time UDP flows do exhibit asymmetry.

3. Intrusion Tolerance and Mitigation

Research on intrusion tolerance accepts that it’s impossible to prevent or stop DDoS completely and focuses on minimizing the attack impact and on maximizing the quality of its services. Intrusion tolerance can be divided in two categories: fault tolerance and quality of service (QoS).

Fault tolerance is a well-developed research area whose designs are built-in in most critical infrastructures and applied in three levels: hardware, software and system. The idea of fault tolerance is that by duplicating the networks services and diversifying its access points, the network can continue offering its services when flooding traffic congests one network link.

Quality of service (QoS) describes the assurance of the ability of a network to deliver the predictable results for certain types of applications or traffic. Many Intrusions Tolerant QoS Techniques and Intrusion Tolerant QoS systems have been developed in order to mitigate DDoS attacks. A similar approach to VIP nets was adopted by Khattab et al.   and they propose an approach called proactive server roaming in order to mitigate DoS attacks. According to this approach the active server proactively changes its location within a pool of servers to defend against unpredictable and undetectable attacks. Only legitimate clients can track the moving server. This roaming scheme has insignificant overhead in attack-free situations and can provide good response time in case of attacks.

4. Intrusion Response

Once an attack is identified, the immediate response is to identify the attack source and block its traffic accordingly. The blocking part is usually performed under manual control (e.g., by contacting the administrators of upstream routers and enabling access control lists) since an automated response system might cause further service degradation in response to a false alarm. Automated intrusion response systems do exist, but they are deployed only after a period of self-learning (for the ones that employ neural computation in order to discover the DDoS traffic) or testing (for the ones that operate on static rules). Improving attack source identification, techniques can expedite the capture of attackers and deter other attack attempts. There are many approaches that target the tracing and identifying of the real attack source. IP traceback traces the attacks back towards their origin, so one can find out the true identity of the attacker and achieve detection of asymmetric routes, as well as path characterization. Some factors that render IP traceback difficult is the stateless nature of Internet routing and the lack of source accountability in the TCP/IP protocol. For efficient IP traceback it is necessary to compute and construct the attack path. It is also necessary to have a low router overhead and low false positive rate. Furthermore, a large number of packets is required to reconstruct the attack path. It is also important the robustness against multiple attacks, the reduction of the privacy of IP communication, the incremental deployment and the backward compatibility. At a very basic level, you can think of this as a manual process in which the administrator of the network under attack places a call to his Internet Service Provider (ISP) asking for the direction from which the packets are coming. Since the manual traceback is very tedious there have been various proposals in the recent past to automate this process.

ICMP traceback has been proposed by Bellovin. According to this mechanism every router sample the forwarding packets with a low probability (1 out of 20,000) and sends an ICMP traceback message to the destination. If enough traceback messages are gathered at the victim, the source of traffic can be found by constructing a chain of traceback messages. A major issue of this approach is the validation of the traceback packets. Although the PKI requirement prevents attackers from generating false ICMP traceback messages, it is unlikely that every router will implement a certificate-based scheme. Furthermore, ICMP traffic generates additional traffic and an upstream router map is required to construct an attack path since the IP addresses of the routers are encoded in the ICMP traceback message.

An alternative, which introduces an intention-bit in the routing and forwarding table, is called Intention- Driven ICMP Traceback. In order to face DDoS attacks by reflectors, Barros proposed a modification of ICMP traceback messages. In this approach, routers send ICMP messages to the source of the currently being processed packet rather than its destination. This reverse trace enables the victim to identify the attacking agent(s) from these packets.

DDoS DEFENSE PROBLEMS

DDoS attacks are a hard problem to solve. First, there are no common characteristics of DDoS streams that can be used for their detection. Furthermore, the distributed nature of DDoS attacks makes them extremely difficult to combat or trace back. Moreover, the automated tools that make the deployment of a DDoS attack possible can be easily downloaded. Attackers may also use IP spoofing in order to hide their true identity, and this makes the traceback of DDoS attacks even more difficult. Finally, there is no sufficient security level on all machines in the Internet, while there are persistent security holes in Internet hosts.

CHAPTER FOUR
CONCLUSION

CONCLUSION

The solution will arise from combining both network and individual countermeasures. DDoS attack tools are readily available and any internet host is targetable as either a zombie or the ultimate DDoS focus. These attacks can be costly and frustrating and are difficult, if not impossible to eradicate. The best defense is to hinder attackers through vigilant system administration.

Applying patches, updating anti-malicious software programs, system monitoring, and reporting incidents go further than retarding DDoS attacks – these defenses also protect against other attacks. The Internet is not stable—it reforms itself rapidly. This means that DDoS countermeasures quickly become obsolete.

New services are offered through the Internet, and new attacks are deployed to prevent clients from accessing these services. However, the basic issue is whether DDoS attacks represent a network problem or an individual problem—or both. If attacks are mainly a network problem, a solution could derive from alterations in Internet protocols. Specifically, routers could filter malicious traffic, attackers could not spoof IP addresses, and there would be no drawback in routing protocols.

If attacks are mostly the result of individual system weaknesses, the solution could derive from an effective IDS system, from an antivirus, or from an invulnerable firewall. Attackers then could not compromise systems in order to create a "zombies" army. Obviously, it appears that both network and individual hosts constitute the problem. Consequently, countermeasures should be taken from both sides.

Because attackers cooperate in order to build the perfect attack methods, legitimate users and security developers should also cooperate against the threat.

REFERENCES

- CERT Coordination Center, Denial of Service attacks, Available from <http://www.cert.org/tech_tips/denial_of_ service.html>.
- Computer Security Institute and Federal Bureau of Investigation, CSI/FBI Computer crime and security survey 2001, CSI, March 2001, Available from <http://www.gocsi. com>.
- D. Moore, G. Voelker, S. Savage, Inferring Internet Denial of Service activity, in: Proceedings of the USENIX Security Symposium, Washington, DC, USA, 2001, pp. 9 22.
- L.D. Stein, J.N. Stewart, The World Wide Web Security FAQ, version 3.1.2, February 4, 2002, Available from <http://www.w3.org/Security/Faq>.
- D. Karig, R. Lee, Remote Denial of Service Attacks and countermeasures, Department of Electrical Engineering, Princeton University, Technical Report CE- L2001-002, October 2001.
- CIAC, Information Bulletin, I-020: Cisco 7xx password buffer overflow, Available from <http://ciac.llnl.gov/ciac/ bulletins/i-020.shtml>.
- Kenney, Malachi, Ping of Death, January 1997, Available from <http://www.insecure.org/sploits/ping-o-death. html>. 662 C. Douligeris, A. Mitrokotsa / Computer Networks 44 (2004) 643–666
- Finger bomb recursive request, Available from <http:// xforce.iss.net/static/47.php>.
- D. Davidowicz, Domain Name System (DNS) Security, 1999, Available from <http://compsec101.antibozo.net/ papers/dnssec/dnssec.html>.
- CERT Coordination Center, Trends in Denial-of-Service attack technology, October 2001, Available from <http:// www.cert.org/archive/pdf/DoS_trends.pdf>.
- DDoS Attacks History (radware.com)