ISSN 2278-7763

Design and Development of an effective DDoS Shield to implement a well secured Defense System against vulnerable attacks

Ayaz Mohiuddin[1] , Mohd Ayaz Uddin[2], Prof.Dr.G.Manoj Someswar[3]

1. **Master of Science in Telecommunication and Computer Networks. (M.S), Working as Lecturer at Salalah College of Technology. PO Box 608 PC 211, Salalah, Sultanate of Oman. E-mail: ayazmohiuddin@gmail.com, ayaz.m@sct.edu.om**
2. **M.Tech (S.E), Associate Professor, Department of IT, Nawab Shah Alam Khan College of Engineering and Technology, Affiliated to JNTUH, Malakpet, Hyderabad – 500024, A.P. India. E-mail: ayazuddin1227@yahoo.com**
3. **B.Tech., M.S.(USA), M.C.A., Ph.D., Professor & HOD, Department of CSE & IT, Nawab Shah Alam Khan College of Engineering and Technology, Affiliated to JNTUH, Malakpet, Hyderabad – 500024, A.P. India. E-Mail: manojgelli@gmail.com (*Corresponding Author*)**

***ABSTRACT*: Denial of Service and the Distributed Denial of Service Attacks have recently merged as one of the most newsworthy, if not the greatest, weaknesses of the Internet. This research paper attempts to explain how they work, why they are hard to combat today, and what will need to happen if they are to be brought under control. It is divided into parts for the purpose of easy analysis and understanding. The first is an overview of the current situation and also brief explanatory of the rest of the chapters being covered. The second is a detailed description of exactly how this attack works, and why it is hard to cope with today; of necessity it includes a description of how the Internet works today. The third section is totally about the different attacks in recent years and how they affected the people or the big organizations. The fourth section describes the short-term prospects, the tools which are used to rectify these attacks. The fifth is problems being faced with an explanatory of the percentage of attack in recent years and comparing the problems. The sixth is what can be done today to help alleviate this problem. The seventh section describes the legal actions and also legal actions that can be followed against the attack by the victim; and the eighth section describes the long-term picture, what will change to bring this class of problem under control, if not eliminate it entirely.**

***Keywords:** SYN Flood, SYN Packets, Least Suspicion First, Resilient Scheduler, Ingress Filtering, Spoofing, Distributed Defense*

INTRODUCTION

Distributed Denial of Service (DDoS) attacks pose an ever greater challenge to the Internet with increasing resources at the hands of the attackers. Recent studies estimate that farms of compromised hosts, popularly known as "botnets," are as large as 60,000 machines. Moreover, the SYN flood attack, the most popular DDoS attack to date, is giving way to sophisticated application-layer (layer-7) attacks. In one instance, an online merchant employed the "DDoS mafia" to launch an HTTP flood towards his competitors' web sites by downloading large image files when a regular SYN flood failed to bring the site down.

Many prior attacks have targeted network bandwidth around Internet subsystems such as routers, Domain Name Servers, or web clusters. However, with increasing computational complexity in Internet applications as well as larger network bandwidths in the systems hosting these applications, server resources such as CPU or I/O bandwidth can become

Copyright © 2014 SciResPub.

ISSN 2278-7763

the bottleneck much before the network.[1] Anticipating a future shift in DDoS attacks from network to server resources, we explore the vulnerability of Internet applications to sophisticated layer-7 attacks and develop counter-attack mechanisms. In particular, our contributions are

(i) classification and experimentation with new application-layer attacks,

(ii) development of a mechanism to assign suspicion measures to sessions for scenarios with a potentially small and variable number of requests per session, and

(iii) design and experimental evaluation of *DDoS Shield*, a technique that provides DDoS resilience by using suspicion measures and server load to determine if and when to schedule requests to a server. In studying new classes of attacks, we consider a well secured system that has defenses against both

   i. intrusion attacks, i.e., attacks which exploit software vulnerabilities such as buffer overflows and

   ii. protocol attacks, i.e., attacks that exploit protocol inconsistencies to render servers inaccessible (e.g., hijacking DNS entries or changing routing).

In such a scenario, the only way to launch a successful attack is for attackers to evade detection by being non-intrusive and protocol-compliant, and yet overwhelm the system resources while posing as legitimate clients of the application service. Hence, the only system attributes available for the attacker to exploit are those for the application workload.

Research Question

SYN (synchronize) is a type of packet used by the Transmission Control Protocol (TCP) when initiating a new connection to synchronize the sequence numbers on two connecting computers. The SYN is acknowledged by a SYN/ACK by the responding computer.

A type of denial of service attack known as a SYN flood involves sending large numbers of SYN packets and ignoring the return, thereby forcing the server to keep track of a large number of half-open connections.

The half-open connections data structure on the victim server system will eventually fill; then the system will be unable to accept any new incoming connections until the table is emptied out. Normally there is a timeout associated with a pending connection, so the half-open connections will eventually expire and the victim server system will recover. However, the attacking system can simply continue sending IP spoofed packets requesting new connections faster than the victim system can expire the pending connections.

In most cases, the victim of such an attack will have difficulty in accepting any new incoming network connection. In these cases, the attack does not affect existing incoming connections or the ability to originate outgoing network connections. However, in

IJOART

Copyright © 2014 SciResPub.

ISSN 2278-7763

some cases, the system may exhaust memory, crash, or be rendered otherwise inoperative.

Our research focusses on a class of attacks in the first category, namely attacks mounted at the application layer (layer-7) with attackers posing as legitimate clients of the service. The attack classes we consider overwhelm server resources in the web cluster and hence are distinct from earlier attacks that have primarily targeted network connectivity.

We design a counter-mechanism, DDoS-Shield that uses the suspicion assignment mechanism as an input to a scheduler designed to thwart attack sessions before they overwhelm system resources.

The DDoS-resilient scheduler incorporates the suspicion assigned to a session and the current system workload to decide when and if a session is allowed to forward requests. We develop scheduling policies Least Suspicion First (LSF) and Proportional to Suspicion Share (PSS) that incorporate suspicion into the scheduling decision. As a baseline for comparison, we implement and study suspicion-agnostic policies such as per session Round Robin and First Come First Serve among all requests.

We apply two approaches to develop relevant and comprehensive test scenarios for our benchmark suite:

1. we use a set of automated tools to harvest typical attack, legitimate traffic, and topology samples from the Internet, and
2. we study the effect that select features of the attack, legitimate traffic and topology/resources have on the attack impact and the defense effectiveness, and use this knowledge to automatically generate a comprehensive testing strategy for a given defense.

The Structure of the Report

The following is the structure of the report:

In the first place, we provide a background and a brief introduction and an explanation of what the attack is.

This research paper gives an understanding about the details of the denial-of-service phenomena, and various common attacks, the DoS attack scenarios and also the effects of DoS and DDoS attacks.

In this research paper, we discuss about the DDoS defense community faces technical and social challenges that hinder the design of effective and widely deployed defenses.

The research paper covers about the Taxonomy of DDoS Attacks, Defenses, usage of Taxonomies in addition to that we cover Source-End Defense like Source-End Detection, Source-End Response, and Deployment Incentive

The research paper discusses about the different types of models like Attacker Model, Victim Model, Defense Model and Vulnerability to Attacks, and also Quantifying Attack Suspicion in addition to that schedule design for DDoS-Shield and lastly working on Detecting DDoS attacks and Counter-DDoS Mechanisms

IJOART

Copyright © 2014 SciResPub.

ISSN 2278-7763

Finally, this research paper deals with the solutions for the problem in various modules. The modules are as follows.

- Develop an attack strategy to overload a generic system Attacker Model
- Victim Model or experimentally validate attacks on an example system
- Resilient Scheduler
- Defense Model
- Develop a generic framework for defense against such attacks

And also this research paper covers what can be done to improve things, the general views, and suggestions to improve the present situation. It is also about what can be really done to protect our network from the attack, the research results given by various past researchers on this attack, and the taxonomy of the DDoS defense mechanism.

Denial-of-service (DoS) and distributed-denial-of-service (DDoS) attacks pose a grave danger to Internet operation. They are, in essence, resource overloading attacks. The goal of the attacker is to tie up a chosen key resource at the vic- tim, usually by sending a high volume of seemingly legitimate traffic requesting some service from the victim. The over consumption of the resource leads to degradation or denial of the victim's service to its legitimate clients.

In the absence of effective defense mechanisms, the denial-of-service effect lasts for the entire duration of the attack (i.e., as long as key resources are being tied with malicious traffic), and vanishes quickly once the attack is aborted. Since machine resources are usually shared among many applications, the DoS effect inflicts significant damage — not only on client transactions with the victim, but on the victim's total operation. The victim experiences a significant slowdown in all applications sharing the targeted resource, and frequently also connectivity disruption.

Both DoS and DDoS attacks are seemingly simple in design and operate with-out requiring any special skill or resource for their perpetration. The attack tools can be obtained easily online and the attack goal (resource exhaustion) is attained whenever a sufficiently large amount of malicious traffic is generated. The targeted resource dictates the type and contents of attack packets, e.g. exhaustion of CPU resources requires computation-intensive packets such as CGI or authentication requests, while network resources can be exhausted by any high-volume traffic.

The main difference between DoS and DDoS attacks is in scale — DoS attacks use one attack machine (to generate malicious traffic) while DDoS attacks use large numbers of attack machines. The scale difference also invokes differences in operation modes. The large number of attack machines allows DDoS perpetrators a certain recklessness — they frequently trade sophistication for brute force, using simple attack strategies and packet contents to overload victim resources. However, the simplicity in both attack types arises from convenience, not necessity. The lack of effective defense mechanisms, even for simple attacks, offers no motivation for perpetrators to design more sophisticated ones. Once defenses successfully counter one attack class (e.g., like ingress filtering [FS00] has countered random IP source spoofing), attackers quickly deploy slight modifications in their attacks to bypass defensive actions.

IJOART

Copyright © 2014 SciResPub.

ISSN 2278-7763

There are many attack variations and many dimensions in which attacks can still evolve while preserving the ability to inflict damage on the victim. This feature makes it very challenging to design successful defenses. Due to attack variety, defense systems must maintain a volume of statistical data in order to detect attacks and sieve legitimate from attack traffic. This incurs high operation costs.

On the other hand, attackers can easily bypass or trick defenses with slight modifications to their attacks. Any such modifications require added complexity in defense mechanisms (in order to handle the new attack class), thus skyrocketing the

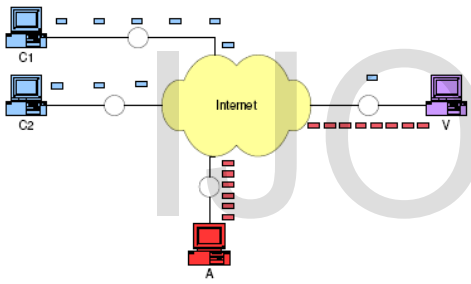

Figure 1: Denial-of-service attack scenario

Denial-of-Service Attacks

A denial-of-service (DoS) attack occurs when the victim receives a malicious stream of packets that exhausts some key resource; this results in denial-of-service to the victim's legitimate clients. Figure 1 depicts a typical denial-of-service attack scenario in which an attacking machine A sends a stream of malicious packets to victim V, denying its service to legitimate clients C1 and C2. Attackers rarely use their own machines to perform attacks, so machine A is, in fact, an agent machine, an unwitting participant subverted by the attacker.

The attack may exhaust a key resource by misusing some vulnerability in the software running at the victim (vulnerability attacks) or by simply sending a higher volume of traffic than the victim is provisioned to handle (flooding attacks). [2] Vulnerability attacks usually contain packets of a special type or content to perform the exploit. As vulnerabilities can frequently be exploited by a few packets, vulnerability attacks are of a low-volume. Both of these features (special type packets and low volume) simplify handling of vulnerability attacks the victim can

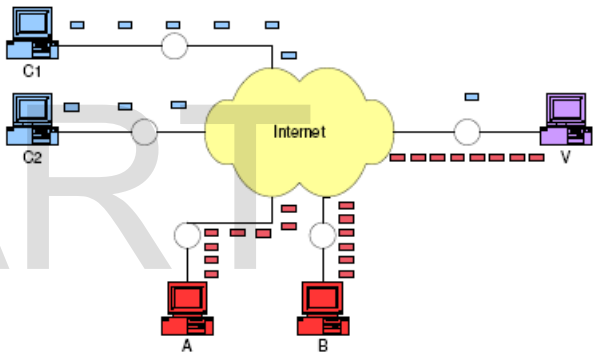

Figure 2: Distributed denial-of-service Attack Scenario

either patch its vulnerability or detect the special-type packets and handle them separately. Flooding attacks overwhelm the victim's resource by sheer volume. This strategy is more difficult to counter, as malicious packets can be of any type or content and the high volume hinders detailed traffic analysis. As DoS attacks involve only one attacking machine, a common approach to de- fending against flooding attacks is to equip the victim with abundant resources. The attacker then needs to find and subvert a better-provisioned machine to per- form a

IJOART

Copyright © 2014 SciResPub.

ISSN 2278-7763

successful attack. The difficulty of the attacker's task to find the adequate agent machine increases with the amount resources allocated to the victim.

Distributed Denial-of-Service Attacks

Distributed denial-of-service (DDoS) attacks are simply denial-of-service attacks performed from multiple subverted machines (agents). In the strawman and most frequently used scenario, all machines are engaged simultaneously and start generating as many packets as they can toward the victim.[3] A large number of 10 participating agents enable the attacker to overload resources of very highly provisioned victims, with modest capabilities of agent machines. Figure 2 depicts a simple distributed denial-of-service attack scenario in which attacking machines A and B send streams of malicious packets to victim V, denying its service to legitimate clients C1 and C2.

IJOART

There are several features of DDoS attacks that severely challenge the design of successful defenses:

- Use of IP source spoofing.

Attackers frequently use source address spoofing during the attack they fake information in the IP source ad- dress field in attack packet headers. One benefit attackers receive from IP spoofing is that it is extremely difficult to trace the agent machines. This, in turn, brings several dire consequences. Since agent machines run a very low risk of being traced, information stored on them (i.e., access logs) cannot help to locate the attacker himself. This greatly encourages DDoS incidents. Furthermore, hiding the address of agent machines enables the attacker to reuse them for future attacks. Last, as attack packets carry a wide variety of addresses, they appear as if they come from many disparate sources; this defeats fair-sharing techniques that are a straight- forward solution to resource overloading problems. The other advantage that IP spoofing offers to the attackers is the ability to perform reflector attacks [Pax01]. The attacker requests (in the victim's name) a public service that generates large replies to specific small-size requests (amplification effect). The attacker generates as many requests for service as his resources permit, faking the victim's source address, and sends them to public servers. These servers direct a many fold volume of replies to the victim (thus reflecting and multiplying the attack force) and overload its 11 resources. A common case of reflector attack is described in [CERe]. The attacker sends a large number of UDP-based DNS requests to a name server using a spoofed source IP address of a victim. Any nameserver response is sent back to the spoofed IP address as the destination.[4] Because name server responses can be significantly larger than DNS requests, there is potential for bandwidth amplification. Even if the traceback problem1 were solved, it would not help to address reflector attacks. The public servers are unwitting participants whose legitimate service is misused in the attack. They possess no information about the attacker. Also, their service cannot be disabled (i.e., to stop the attack) as this would inflict damage on numerous other clients. Depending on these servers' resources and the request volume, they could prevent reflector attacks by limiting the number of replies they are willing to generate to a particular IP address. This approach would require servers to cache requesting addresses, thus potentially consuming significant memory resources.

Copyright © 2014 SciResPub.

ISSN 2278-7763

- Large number of agent machines

Even if traceback could be suc- cessfully performed in the face of IP spoofing, it is difficult to say what actions could be taken against hundreds or thousands of agent machines. Such a large number prevents any but crude automated responses aimed at stopping attack flows close to the sources.

- Similarity of attack to legitimate traffic

Any type of traffic can be used to perform a successful denial-of-service attack. Some traffic types require a higher attack volume for success than others, and attack packets of different types and contents target different resources. However, if the goal is simply to cripple the victim's operation, it can be met by sending sufficiently large volumes of any traffic and clogging the victim's network. Attackers tend to generate legitimate-like packets to perform the attack, obscuring the malicious flow within legitimate traffic. Since malicious packets do not stand out from legitimate ones, it is impossible to sieve legitimate from attack traffic based purely on examination of individual packets. A defense system must keep a volume of statistical data in order to extract transaction semantics from packet flows and thus differentiate some legiti- mate traffic (e.g. belonging to lengthy well-behaved transactions) from the attack traffic.

Origin of Denial-of-Service Phenomenon

Denial-of-service is not simply another weak spot in the Internet, a slip that can be mended with slight protocol changes or by deployment of sophisticated defenses at potential target sites. The origin of denial-of-service lies in the very core of the Internet architecture. Design decisions reached several decades ago, that brought us connectivity and information wealth beyond our wildest dreams, carry within their key concepts the root of the DDoS threat.

The Internet was designed with functionality, not security, in mind, and it has been very successful in reaching its goal. It offers participants fast, simple and cheap communication mechanisms at the network level that provide "best effort" service to a variety of protocols. The only claim made is that the Internet will make a best attempt to move packets from a sender to a destination. Packet loss, reorder or corruption, sharing of Internet resources, different service levels for different traffic types and similar performance issues are handled by higher-level 13 transport protocols deployed at the end hosts — the sender and the receiver.[5] These two principles, best-effort service and the end-to-end paradigm are the cornerstones upon which the Internet was built. Simple basic service provided by the IP protocol and the "best effort" principle enabled the building of numerous transport protocols on top of the IP to provide various performance guarantees: TCP for reliable delivery, RTP, RTCP and RTSP for streaming media, ICMP for control, etc. The end-to-end paradigm enabled end users to manage their communication any way they desired, adding complexities such as encryption and authentication, while the intermediate network remained simple and efficient.

Problems arise when one of the parties in the end-to-end model becomes malicious and acts to damage the other party. In that scenario, end-to-end protocols are violated and provide no more guarantees. At the same

IJOART

Copyright © 2014 SciResPub.

ISSN 2278-7763

time, the end-to- end paradigm prevents the intermediate network from stepping in and policing the violator's traffic. Instead, it continues passively forwarding packets to their destination, where they overwhelm the victim's resources.

This problem first became evident in October 1986 when the Internet suffered a series of congestion collapses. Although the problem was quickly addressed by the design and deployment of several TCP congestion control pro- tocols [Flo00], end-to-end flow management was unable to ensure a fair allocation of resources in the presence of aggressive flows (i.e., those that would not deploy congestion control). This problem was recognized and finally handled by enlisting the help of intermediate routers to monitor and police bandwidth allocation among flows to ensure fairness. There are two major mechanisms deployed in today's routers for congestion avoidance purposes — active queue management and fair scheduling algorithms. A similar approach that engages intermediate routers in flow management may be needed to completely solve the 14 DDoS problem.

The following list summarizes several features of Internet design that open security issues and create opportunities for denial-of-service attacks:

- Internet security is highly interdependent.

DDoS attacks are commonly launched from systems that are subverted through security-related compromises. Regardless of how well secured the victim system may be, its susceptibility to DDoS attacks depends on the state of security in the rest of the global Internet.

- Internet control is distributed.

Internet management is distributed, and each network is run according to local policies defined by its owners. The implications of this are many. There is no way to enforce global deployment of a particular security mechanism or security policy, and due to privacy concerns, it is often impossible to investigate cross-network traffic behavior.

- Internet resources are limited.

Each Internet entity (host, network, service) has limited resources that can be consumed by too many users. This means that every DDoS attempt will be successful (in absence of defenses) if it acquires a sufficiently large pool of agent machines.

- The power of many is greater than the power of few.

Coordinated and simultaneous malicious actions by some participants will always be detrimental to others if the resources of the attackers are greater than the resources of the victims.

- Intelligence and resources are not collocated.

An end-to-end communication paradigm led to storing most of the intelligence needed for service 15 guarantees with end hosts, limiting the amount of processing in the intermediate network so that packets could be forwarded quickly and at minimal cost. At the same time, a desire for large throughput led to the design of high bandwidth pathways in the

IJOART

Copyright © 2014 SciResPub.

ISSN 2278-7763

intermediate network, while the end networks invested in only as much bandwidth as they thought they might need. Thus, malicious clients can misuse the abundant resources of the unwitting intermediate network for delivery of numerous messages to a less provisioned victim.

- Accountability is not enforced.

The source address field in an IP packet is assumed to carry the IP address of the machine that originates the packet. This assumption is not generally validated or enforced at any point on route from the source to the destination. This creates the opportunity for source address spoofing — the forging of source address fields in packets. Source address spoofing gives attackers a powerful mechanism to escape accountability for their actions, and sometimes even the means to perpetrate attacks (reflector attacks, such as the Smurf [CERj] attack).

IJOART

Attacker Goals

The goal of a DDoS attack is to inflict damage on the victim. Frequently the ulterior motives are personal reasons (a significant number of DDoS attacks are perpetrated against home computers, presumably for purposes of revenge), or prestige (successful attacks on popular Web servers gain the respect of the hacker community). However, it is not unlikely that some DDoS attacks are performed for material gain (damaging competitor's resources, such as the recent case of Linux fans attacking SCO [Sha03] because of its lawsuit against IBM) or for 16 political reasons (a country at war could perpetrate attacks against its enemy's critical resources, potentially enlisting a significant portion of the entire country's computing power for this action). In some cases, the true victim of the attack might not be the actual target of the attack packets, but others who rely on the target's correct operation. For example, in September 2002 there was an onset of attacks that overloaded the Internet infrastructure rather than targeting specific victims [Nar02].[6]

It also frequently happens that a DDoS attack is perpetrated accidentally, as a byproduct of another malicious activity, such as worm spread [Moo, Sym]. Inefficient worm-spreading strategies create massive traffic that congests the Internet and creates a denial-of-service effect to numerous clients.

While ordinary home users are less likely to become victims of DDoS attacks than large corporate networks, no one is free from the DDoS threat. The next attack may target AOL servers, denying service to many home users, or the next worm may congest the Internet so severely that no one can receive service. DDoS is an Internet-wide problem and all parties should cooperate to find a suitable solution.

Modus Operandi

A distributed denial-of-service is carried out in several phases. The attacker first recruits multiple agent (slave) machines. This process is usually performed automatically: the attacker downloads a scanning tool and deploys it from other compromised machines under its command (masters). The tool scans remote machines, probing for security holes that will enable subversion. Vulnerable machines are then exploited—broken into using the discovered vulnerability. They are subsequently infected with the attack code. The exploit/infect phase is also

Copyright © 2014 SciResPub.

ISSN 2278-7763

automated, and the infected machines can be used for further recruitment of new agents.

Attackers attempt to cover the fact that agent machines have been compromised. They erase all logs showing malicious activity to destroy evidence that could incriminate them. They also hide attack scripts under system directories and give them obscure, non-suspicious names so they will not attract a user's attention and be erased. Sometimes they patch the vulnerability used for the exploit, to prevent other hackers from taking over the machine. Current exploit/infection scripts contain automated tools for covering tracks, so even inexperienced attackers do not leave much evidence of the subversion.[7]

During a DDoS attack, agent machines are engaged to send the attack pack- ets to the victim. The attacker orchestrates the onset of the attack, and scenario details such as the desired type and duration and the target address from the master to the agent machines. Agent machines usually fire out the packets at a maximum possible rate to increase the attack's chances of success. However, there have been attacks where agents were generating packets at a small rate (to prevent agent discovery) or where agent machines were periodically pausing the attack to avoid detection (pulsing attacks). Attackers usually hide the identity of subverted machines during the attack through spoofing of the source address field in attack packets.[8] Note, however, that spoofing is not always required for a successful DDoS attack. With the exception of reflector attacks that use spoofing as an attack tool, all other attack types use spoofing only to hinder detection and discovery of agent machines.

Figure 3 illustrates the recruitment, exploitation, infection and engagement phases, depicting also the master/slave architecture of compromised machines.

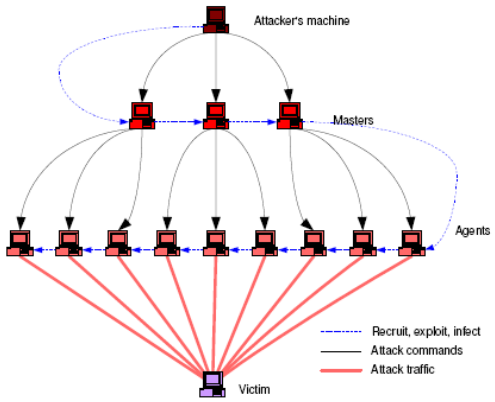

Figure 3: Distributed denial-of-service attack: modus operandi

Commonly Observed Attacks

While there are many ways to create the denial-of-service effect, there are a handful of attacks that have been commonly observed in the majority of DDoS incidents.

- UDP flooding attack.

During this attack the victim is flooded by numerous UDP packets that overwhelm its network bandwidth. To fully exploit bandwidth resources, packets usually have a large size. This attack is very simple to perpetrate, as the attacker need not discover (and take advantage of) any vulnerability at the victim. Simply by deploying a large number of agents, he can ensure the attack's success. On the other hand, many victim sites do not regularly receive incoming UDP traffic and can discard attack packets by deploying simple filtering rules. If filters are deployed at a high

IJOART

Copyright © 2014 SciResPub.

ISSN 2278-7763

bandwidth point (e.g., an upstream router), this attack can be handled successfully.

- TCP SYN flooding attack (open port).

An attacker takes advantage of a vulnerability in the TCP protocol design to perpetrate a TCP SYN flooding attack. A TCP session starts with negotiation of session parameters between a requesting party — a client and a server. The client sends a TCP SYN packet to the server, requesting some service. In the SYN packet header, the client provides his initial sequence number, a unique per-connection number that will be used to keep count of data sent to the server (so the server can recognize and handle missing, reordered or repeated data). [9] Upon SYN packet receipt, the server allocates a connection buffer record, storing information about the client. He then replies with a SYN-ACK, informing the client that its service request will be granted, acknowledging the client's sequence number and sending information about the server's initial sequence number. The client, upon receipt of the SYN- ACK packet, allocates a connection buffer record. The client then replies with an ACK to the server which completes the opening of the connection. This message exchange is called a three-way handshake and is depicted in Figure 4.

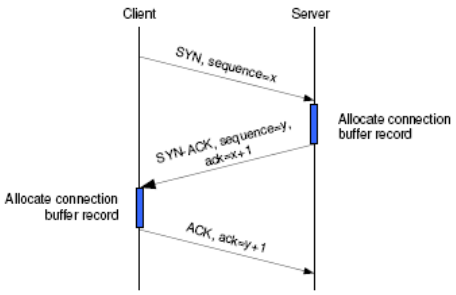

Figure 4: Opening of TCP connection: three-way handshake

The potential for abuse lies in the early allocation of the server's resources. When the server allocates his connection buffer space and replies with a SYN-ACK, the connection is said to be half-open. The server's allocated resources will be tied up until the client sends an ACK packet, closes the connection (by sending an RST packet) or until a timeout expires and the server closes the connection, releasing the buffer space. During a TCP SYN flooding attack, the attacker generates a multitude of half-open connections by using IP source spoofing. These requests quickly exhaust the server's connection buffer space, and the server can accept no more incoming connection requests. Established TCP connections usually experience no degradation in service. In rare cases, the server machine crashes, exhausts its memory or is otherwise rendered inoperative. In order to keep buffer space occupied for the desired time, the attacker needs to generate a steady stream of SYN packets toward the victim (to reserve again those resources that have been freed by timeouts).[10]

- The TCP SYN flooding attack is described in detail in [CERk, SKK97].

This is an especially vicious attack, as servers expect to see large numbers of legitimate SYN packets and cannot easily tell apart the legitimate from the attack traffic. No simple filtering rule can handle the TCP SYN flooding attack because legitimate traffic will suffer collateral damage.

IJOART

Copyright © 2014 SciResPub.

ISSN 2278-7763

In order to perform a successful TCP SYN flooding attack, the attacker needs to locate an open TCP port at the victim. Then he generates a relatively small volume packet stream — as few as ten packets per minute[SKK97] can effectively tie up victim's resources. Another version of the TCP SYN flooding attack — random port TCP SYN flooding — is much less common. In it, the attacker generates a large volume of TCP SYN packets targeting random ports at the victim, with the goal of overwhelming the victim's network resources. Because TCP SYN packets are small, this is a very inefficient way of exhausting bandwidth, and is thus an unlikely attack.

- **ICMP flooding attack**

During an ICMP flooding attack, the attacker generates a flood of ICMP ECHO packets directed at the victim. The victim replies to each ICMP request, consuming its CPU resources (for reply generation) and network resources. This attack is as simple to perpetrate as a UDP flooding attack. As machines usually receive a very low volume of incoming ICMP packets, they can substantially defend against ICMP flooding attacks by deploying a simple rate-limiting rule at a high-bandwidth point (e.g., an upstream router), at the cost of dropping a few legitimate ICMP requests in the process.

- **Smurf attack**

The Smurf attack[CERj] is a reflector attack. The attacker directs a stream of ICMP ECHO requests to broadcast addresses in intermediary networks, spoofing the victim's IP address in their source address fields. A multi- tude of machines then reply to the victim, overwhelming its network. This attack is easily countered either at the source network (generating forged ICMP ECHO requests) by deploying ingress filtering [FS00] or at the intermediary network by ignoring/filtering out ICMP ECHO requests targeting broadcast addresses.[11]

- **Domain Name Service (DNS) reflector attack**

This attack sends a stream of DNS requests to multiple nameservers, spoofing the victim's address in their source address fields [CERe]. Because nameserver responses can be significantly larger than DNS requests, there is potential for bandwidth amplification. Attackers usually request the same valid DNS record from multiple nameservers. If the target nameserver allows the query and is configured to be recursive or to provide referrals, the response could contain significantly more data than the original DNS request, resulting in a higher degree of bandwidth amplification.[12] A target nameserver configured without restrictions on DNS query sources may not log malicious queries at all. An available defense at the source side (the network generating spoofed DNS requests) is to deploy ingress filtering. As intermediary servers receive legitimate-like requests, they cannot detect and prevent the attack (unless they exchange statistics on requesting addresses, or limit the number of responses to a given address).

Commonly Used Attack Tools

While there are numerous scripts that are used for scanning, compromise and infection of vulnerable machines, there are only a handful of DDoS attack tools that have been used to carry out the engagement

IJOART

Copyright © 2014 SciResPub.

ISSN 2278-7763

phase. DDoS attack tools mostly differ in the communication mechanism deployed between masters and slaves, and in the customizations they provide for attack traffic generation. The following paragraphs provide a brief overview of these popular tools. The reader should bear in mind that features discussed in this overview are those that have been observed in instances of attack code detected on some infected machines. Many variations may (and will) exist that have not yet been discovered and analyzed.

Trinoo[Dita] deploys a master/slave architecture, where an attacker sends commands to the master via TCP and masters and slaves communicate via UDP. Both master and slaves are password protected to prevent them from being taken over by another attacker. Trinoo generates UDP packets of a given size to random ports on one or multiple target addresses, during a specified attack interval.

Tribe Flood Network (TFN) [Ditc] also deploys a master/slave architecture. Agents can wage a UDP flood, TCP SYN flood, ICMP ECHO flood and Smurf attacks at specified or random victim ports. The attacker communicates with masters using any of a number of connection methods (e.g., remote shell bound to a TCP port, UDP based client/server remote shells, ICMP-based client/server shells such as LOKI[rou97], SSH terminal sessions, or normal ”telnet” TCP terminal sessions.) Remote control of TFN agents is accomplished via ICMP ECHOREPLY packets. All commands sent from master to slaves through ICMP packets are coded, not clear text, which hinders detection.

Stacheldraht[Ditb] (German for “barbed wire”) combines features of Trinoo and TFN tools and adds encrypted communication between the attacker and the masters. Stacheldraht uses TCP for encrypted communication between the attacker and the masters, and TCP or ICMP for communication between master and agents. Another added feature is the ability to perform automatic updates of agent code. Available attacks are UDP flood, TCP SYN flood, ICMP ECHO flood and Smurf attacks.[13]

Shaft[SD00] is a DDoS tool similar to Trinoo, TFN and Stacheldraht. Added features are the ability to switch master servers and master ports on the fly (thus hindering detection by intrusion detection systems), a ”ticket” mechanism to link transactions, and a particular interest in packet statistics. Shaft uses UDP for communication between masters and agents. Remote control is achieved via a simple telnet connection from the attacker to the master. Shaft uses ”tickets” for keeping track of its individual agents. Each command sent to the agent contains a password and a ticket. Both passwords and ticket numbers have to match for the agent to execute the request. A simple letter-shifting (Caesar cipher) is used to obscure passwords in sent commands. Agents can generate a UDP flood, TCP SYN flood, ICMP flood, or all three attack types. The flooding occurs in bursts of 100 packets per host (this number is hard-coded), with the source port and source address randomized. Masters can issue a special command to agents to obtain statistics on malicious traffic generated by each agent. It is suspected that this is used to calculate the yield of a DDoS network.

Tribe Flood Network 2000 (TFN2K) [CERd] is an improved version of the TFN attack tool. It includes

IJOART

Copyright © 2014 SciResPub.

ISSN 2278-7763

several features designed specifically to make TFN2K traffic difficult to recognize and filter, to remotely execute commands, to obfuscate the true source of the traffic, to transport TFN2K traffic over multiple transport protocols including UDP, TCP, and ICMP, and features to confuse attempts to locate other nodes in a TFN2K network by sending "decoy" packets. TFN2K obfuscates the true traffic source by spoofing source addresses. Attackers can choose between random spoofing and spoofing within a specified range of addresses (to defeat ingress filtering [FS00]). In addition to flooding, TFN2K Can also performs some vulnerability attacks by sending malformed or invalid packets, as described in [CERl, CERg].

mstream[DWD] generates a flood of TCP packets with the ACK bit set. Masters can be controlled remotely by one or more attackers using a password-protected interactive login. The communications between attacker and masters, and a master and agents, are configurable at compile time and have varied significantly from incident to incident. Source addresses in attack packets are spoofed at random. The TCP ACK attack exhausts network resources and will likely cause a TCP RST to be sent to the spoofed source address (potentially also creating outgoing bandwidth consumption at the victim).[14]

Trinity is the first DDoS tool that is controlled via IRC or ICQ. Upon com- promise and infection by Trinity, each machine joins a specified IRC channel and waits for commands. Use of legitimate (IRC or ICQ) service for communication between attacker and agents eliminates the need for a master machine and ele- vates the level of the threat, as explained in Section 4.1. Trinity is capable of launching several types of flooding attacks on a victim site, including UDP, IP fragment, TCP SYN, TCP RST, TCP ACK, and other floods.

Research Analysis

Distributed denial-of-service are simple attacks. They rarely use any sophisticated mechanism or complicated and covert actions (like viruses, worms or intrusion tools do). Instead they attack with brute force, gathering resources of numerous agents to overwhelm the victim. The difficulty in handling DDoS at- tacks lies exactly in their simplicity. Because they misuse legitimate protocols to perform denial-of-service, it is extremely difficult to separate attack traffic from legitimate traffic; this hinders both detection and response. IP spoofing additionally complicates the problem.

This research paper has provided an overview of attack methods, most frequently seen incidents and popular attack tools. However, DDoS attacks are adversarial and constantly evolving. Once a particular kind of attack is successfully countered, a slight variation is designed that bypasses the defense and still performs an effective attack. DDoS attacks can afford to vary many of their features, such as IP header values, deployed protocols, agent sending rate, etc. As long as the victim receives a flood of packets that overwhelms its resources, the attack succeeds. This high variability needs to be sufficiently understood to design effective defenses.

The seriousness of the DDoS problem and the increased frequency, sophistication and strength of attacks have led to the advent of numerous defense mechanisms. Yet, although it has been several years since the first distributed attacks were perpetrated,1 and many solutions have been developed since then,

IJOART

Copyright © 2014 SciResPub.

ISSN 2278-7763

the problem is hardly dented, let alone solved. Why is this so?

Defense Challenges

The challenges to designing DDoS defense systems fall roughly into two categories: technical challenges and social challenges. Technical challenges encompass problems associated with the current Internet protocols and characteristics of the DDoS threat. Social challenges, on the other hand, largely pertain to the manner in which a successful technical solution will be introduced to Internet users, and accepted and widely deployed by these users.

The main problem that permeates both technical and social issues is the problem of large scale. DDoS is a distributed threat that requires a distributed solution. Attacking machines may be spread all over the Internet. Clearly, attack streams can only be controlled if there is a point of defense between the agents and the victims. One approach is to place one defense system close to the victim so that it monitors and controls all of the incoming traffic. This approach has many deficiencies (see Section 3.4), the main one being that the system must be able to efficiently handle and process huge traffic volumes. The other approach is to divide this workload by deploying distributed defenses. Defense systems must then be deployed in a widespread manner to ensure effective action for any combination of agent and victim machines. As widespread deployment cannot be guaranteed, the technical challenge lies in designing effective defenses that can provide reasonable performance even if they are sparsely deployed.

The social challenge lies in designing an economic model of a defense system in a manner that facilitates large-scale deployment in the Internet.

Technical Challenges

The distributed nature of DDoS attacks and use of legitimate traffic models and IP spoofing represent the main technical challenges to designing effective DDoS defense systems. In addition to that, the advance of DDoS defense research is hindered by the lack of attack information and absence of standardized evaluation and testing approaches. The following list summarizes and discusses technical challenges for DDoS defense:

- Need for a distributed response at many points on the Internet.

It elaborates on the fact that there are many possible DDoS attacks, very few of which can be handled only by the victim. Thus it is necessary to have a distributed, possibly coordinated, response system. It is also crucial that the response be deployed at many points on the Internet to cover diverse choices of agents and victims. Since the Internet is administered in a distributed manner, wide deployment of any defense system (or even various systems that could cooperate) cannot be enforced or guaranteed. This discourages many researchers from even designing distributed solutions.

- Lack of detailed attack information.

It is widely believed that reporting occurrences of attacks damages the business reputation of the victim network. Therefore, very limited information exists

IJOART

Copyright © 2014 SciResPub.

ISSN 2278-7763

about various attacks, and incidents are reported only to government organizations under obligation to keep them secret. It is difficult to design imaginative solutions to the problem if one cannot become familiar with it. Note that the attack information should not be confused with attack tool information, which is publicly available at many Internet sites. Attack information would include the attack type, time and duration of the attack, number of agents involved (if this information is known), attempted response and its effectiveness, damages suffered, etc.

- Lack of defense system benchmarks.

Many vendors make bold claims that their solution completely handles the DDoS problem. There is currently no standardized approach for testing DDoS defense systems that would enable their comparison and characterization. This has two detrimental influences on DDoS research:

1. since there is no attack benchmark, defense designers are allowed to present those tests that are most advantageous to their system, and
2. researchers cannot compare actual performances of their solutions to the existing defenses; instead they can only comment on design issues.

- Difficulty of large-scale testing.

DDoS defenses need to be tested in a realistic environment. This is currently impossible due to the lack of large scale testbeds, safe ways to perform live distributed experiments across the Internet, or detailed and realistic simulation tools that can support several thousands of nodes. Claims about defense system performance are thus made based on small-scale experiments and simulations, and are not credible.

Social Challenges

Many DDoS defense systems require certain deployment patterns to be effective. Those patterns fall into several categories:

1. Complete deployment

2. Contiguous deployment

3. Large-scale, widespread deployment

4. Complete deployment at specified points in the Internet

5. Modification of widely deployed Internet protocols, such as TCP, IP or HTTP

6. All (legitimate) clients of the protected target deploy defenses None of the above requirements are practical for general purposes (although they may work well to protect an important server or application that communicates with a selected set of clients).

The Internet is extremely large and is managed in a distributed manner. No solution, no matter how effective, can be deployed simultaneously in hundreds of millions of disparate places. On the other hand, there have been quite a few cases of an Internet product (a protocol, an application or a system) that has become so popular after release that it was very

IJOART

Copyright © 2014 SciResPub.

ISSN 2278-7763

widely deployed within a short time.[15] Examples include Kazaa, SSH (Secure Shell) protocol, Internet Explorer, Windows OS, etc. The following factors determine a product's chances for wide deployment:

- Good performance. A product must meet the needs of customers.
- Good economic model. A customer must gain direct economic benefit, or at least reduce the risk of economic loss, by deploying the product. Alternately, the customer must be able to charge others for improved services resulting from deployment.
- Incremental performance. As the degree of deployment increases, customers might experience increased benefits. However a product must offer considerable benefit to its customers even under sparse partial deployment.

Defense Goals

The primary goal of DDoS defense is to provide good service to a victim's legitimate clients during the attack, thus canceling the denial-of-service effect. Ideally, clients should perceive little or no service degradation while the attack is ongoing. The secondary goal is to alleviate the effect of the attack on the victim so that its resources can be dedicated to legitimate clients or preserved. Last, attack attribution (locating with high accuracy agent machines and perpetrators of the attack) will serve as a strong deterrent to DDoS incidents, as attackers could face the risk of discovery and punishment.

Defense Approaches

DDoS defense approaches can roughly be divided into three categories: preventive, survival and responsive approaches.

Preventive approaches introduce changes into Internet protocols, applications and hosts, in order to patch existing vulnerabilities and reduce the incidence of intrusions and exploits. Their goal is to prevent vulnerability attacks, and to impede the attacker's attempts to gain a large agent army. While preventive approaches are necessary for improving Internet security, they need to be deployed widely to constrain the DDoS threat. As long as large numbers of machines are insecure, attackers can still wage large-scale attacks. There is no reason to believe that preventive approaches will successfully undermine the power of the DDoS threat in the foreseeable future.

Survival approaches enlarge a victim's resources, enabling it to serve both legitimate and malicious requests during the attack, thus cancelling the denial of- service effect. The enlargement is achieved either statically — by purchasing more resources, or dynamically — by acquiring resources at the sign of possible attack from a set of distributed public servers and replicating the target service. Enlargement approaches can significantly enhance a target's resistance to DoS. Replication approaches offer successful DDoS protection (and also load balancing) to static Internet content. The disadvantage is that not all public services (those that may be subject to the DDoS attacks) are replicable. For instance, dynamic Web pages, databases, remote login services, etc., can be replicated only with a great effort invested into synchronization and

IJOART

Copyright © 2014 SciResPub.

ISSN 2278-7763

emulation. The effectiveness of survival approaches is limited to cases in which enlarged resources are greater than the attack volume. As an attacker can easily gather hundreds of thousands of agent machines, survival approaches are not likely to offer a complete solution to DDoS problem.

Responsive approaches detect the occurrence of the attack and respond to it ("fight back") either by controlling attack streams, or by attempting to locate agent machines and invoking human action. In order to be successful, response approaches must meet following requirements:

1. Accurate detection. The system must be able to detect all attacks that inflict damage at the victim.

2. Effective response. The system must stop the attack flows, regardless of their volume or distribution. Alternately, in the case of response by agent identification, the system must be able to accurately identify the majority of attack machines regardless of their distribution. This identification must be prompt so that the action can be taken while the attack is on-going. Ideally, identification responses should identify not only the agent machines, but also the master and the attacker machines.

3. Selective response. The system must differentiate between legitimate and attack packets, and ensure good service to legitimate traffic during the attack. Collateral damage due to the response must be lower than the damage suffered by legitimate clients in the absence of response. This requirement does not pertain to agent identification approaches. We call these requirements responsive defense requirements.

The remainder of the research work will discuss only responsive approaches that react to incidents by controlling attack streams.

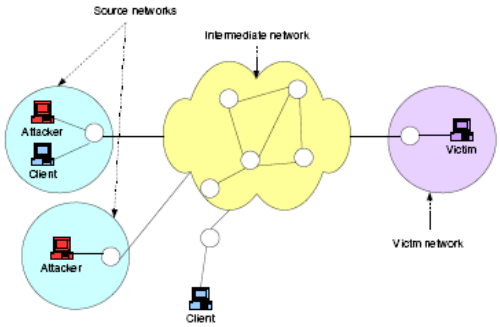

Figure 5: Points of defense

A DDoS defense system can either be deployed as an autonomous (single-point) system or as a distributed system. Autonomous systems consist of a single defense node that observes the attack and applies the response. Distributed systems consist of multiple defense nodes (frequently with same functionality) that are deployed at various locations and organized into a network. Nodes communicate through the network and coordinate their actions to achieve a better overall defense.

Autonomous Defense

DDoS attack streams originate from distributed attack machines, are forwarded by core routers and converge at the victim network or some nearby core router. We observe this process as an interaction of three types of networks: source networks that unwittingly host attack machines, several intermediate networks that forward attack traffic to the victim, and the victim network that hosts the

IJOART

Copyright © 2014 SciResPub.

ISSN 2278-7763

target. Figure 4.1 depicts this interaction. Each of the involved networks (source, intermediate, or victim) can host DDoS defense systems. We now observe how responsive defense requirements are met by an autonomous system deployed at only one of these points.

Victim-End Defense

Historically, the majority of DDoS defense systems have been designed for victimend deployment. This is understandable since the victim suffers the largest damage from a DDoS attack and is therefore motivated to invest in a defense system. A victim-end DDoS defense system facilitates easy detection because it can closely observe the victim, model its behavior and notice any anomalies. However, the range of response is limited. The defense system is on the path of the full-force attack, and may be overwhelmed by a large traffic volume. The point of failure is then simply moved from the target to the DDoS defense system. Alternately, the attacker may send enough traffic to overwhelm the victim's network connection in front of the defense system. The point of DoS is then beyond the defense system's scope. The other consequence of a large traffic volume is the limited amount of processing and storage that defense system can commit to. The differentiation of legitimate streams from attack streams is complex at this point, since they have been heavily aggregated by the time they reach the victim network. To perform sophisticated traffic profiling a system needs a large amount of storage and computational power to store and examine statistics on each stream. In the presence of IP spoofing, this is infeasible as each packet will appear to come from a different source. Poor traffic separation, in turn, leads to large collateral damage during a response.

Intermediate-Network Defense

The danger of a DDoS attack on network resources that is still present in victimend defense was addressed by moving the defense further upstream, into the intermediate network. An intermediate-network defense system, usually installed at a core router, detects the attack through anomalies observed at this router. As core routers handle large-volume, highly aggregated traffic, they are likely to overlook all but large-scale attacks.[14] Victim resources are frequently severely depleted by attacks that look like small glitches in the busy buffer of a core router. Detected attacks can be quickly suppressed, thanks to abundant network resources. However, response is likely to inflict collateral damage as core routers can only accommodate simple rate-limiting requests and cannot dedicate memory or processor cycles to traffic profiling.

Source-End Defense

As DDoS defense is pushed further from the victim to the source, detection capability diminishes. A source-end defense system can no longer easily observe the effect of incoming traffic on the victim. Further, as it may monitor only a small portion of the attack, the defense system has difficulties in detecting anomalies. On the other hand, response effectiveness increases with proximity to the sources. A small attack volume enables an effective response as it is unlikely to overwhelm the defense system. The small volume and degree of aggregation also facilitates

IJOART

Copyright © 2014 SciResPub.

ISSN 2278-7763

complex profiling that, in turn, minimizes collateral damage.

Distributed Defense

Distributed systems for DDoS defense combine actions of victim-end, source-end and sometimes of intermediate-network defense systems. Victim-end defenses detect the attack and deliver the alert to other participants, who then cooperate to suppress attack streams. The goal is to install responses as close to the sources as possible, thus minimizing collateral damage. Distributed defenses are likely to be the proper solution for handling the DDoS threat. However, they are infrastructural solutions—they span multiple networks and administrative domains and represent major undertakings of many Internet participants. Such systems are difficult to deploy and maintain. Further, the required cooperation of defenses is hard to achieve due to distributed Internet management and strictly autonomous operation of administrative domains. Securing and authenticating the communication channels also incurs a high cost if the number of participants is large.

Conclusion

The DDoS defense community faces technical and social challenges that hinder the design of effective and widely deployed defenses. Technical challenges lie in the design of defenses that detect a wide range of attacks, inflict small collateral damage to legitimate traffic and are effective in sparse deployment. The social challenge lies in the design of a good economic model of DDoS defense so that it can be widely deployed.

Defense approaches aim to prevent denial-of-service attacks (preventive approaches), to enable the victim to survive the attack without denying service to legitimate clients (survival approaches) or to detect and respond to the attack by selectively dropping attack traffic (responsive approaches). Preventive and survival approaches improve the security and raise the bar for DDoS attack success, but they cannot completely handle DDoS attacks. Responsive approaches show promise for complementing the action of preventive and survival approaches and completely addressing the DDoS problem.

To provide an effective defense, responsive approaches must accurately detect a wide range of attacks, effectively respond to detected attacks by stopping a large portion of the attack traffic, and apply selective response — thus inflicting low collateral damage to legitimate traffic. Attack detection is easiest at the victim network: a high-volume of incoming traffic or disturbed operation can be readily used as a sign of DDoS attack. Effective response, however, depends on the attack volume and victim network resources. No victim-end defense is possibl against sufficiently high-volume attacks — they overwhelm network resources even before they reach the defense system, leaving legitimate clients without service. Additionally, their high level of traffic aggregation hinders differentiation between legitimate and attack flows, leading to a non-selective response. Thus, while protecting the victim, the response penalizes some legitimate traffic, still leading to denial-of-service.

The selectiveness and effectiveness of response improve as the defense system is moved from the victim closer to the sources of the attack, but the

IJOART

Copyright © 2014 SciResPub.

ISSN 2278-7763

detection accuracy deteriorates. Response is most effective at the source-end network, as attack streams can be stopped before they enter the Internet. Also, sophisticated profiling can be done to facilitate selectiveness of the response, since the attack traffic at the source is not highly aggregated. However, attack machines can be distributed among many source networks, thus each source network only observes a small amount of attack traffic that may appear legitimate, hindering detection. Because none of the individual defense points can meet all three requirements of an effective defense, a distributed defense system presents itself as a likely solution.

References

[1] TPC-W: Transaction Processing Council . http://www.tpc.org.

[2] C. Amza, A. Cox, and W. Zwaenepoel. Conflict-aware scheduling for dynamic content applications. In *Proceedings of the 4th USENIX Symposium on Internet Technologies and Systems (USITS'03)*, Seattle,WA, March 2003.

[3] Service provider infrastructure security: detecting, tracing, and mitigating network-wide anomalies. http://www.arbornetworks.com, 2005.

[4] K. Argyraki and D.R. Cheriton. Active internet traffic filtering: Realtime response to denial-of-service attacks. In *Proc. of USENIX Annual Technical Conference*, April 2005.

[5] M. Aron, D. Sanders, P. Druschel, and W. Zwaenepoel. Scalable content-aware request distribution in cluster-based network servers. In *Proceedings of the USENIX 2000 Annual Technical Conference*, June 2000.

[6] N. Bhatti, A. Bouch, and A. Kuchinsky. Integrating user-perceived quality into web server design. In *Proceedings of the 9th International World Wide Web Conference*, Amsterdam, Netherlands, May 2000.

[7] T. Calinski and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics*, 3:1–27, 1974.

[8] http://www.cert.org, 2005.

[9] President's Information Technology Advisory Committee. Cyber security: A crisis of prioritization. www.hpcc.gov/pitac/reports/20050301 cybersecurity/cybersecurity.pdf.

[10] Cover and Thomas. *"Elements of Information Theory"*. Wiley, 1991.

[11] California Central District. United states vs jay echouafni et al. (operation cyberslam). www.usdoj.gov/criminal/fraud/websnare.pdf.

[12] A. Garg and A. L. N. Reddy. Mitigating denial of service attacks using qos regulation. In *Proceedings of International Workshop on Quality of Service (IWQoS)*, 2002.

[13] J. Jung, B. Krishnamurthy, and M. Rabinovich. Flash crowds and denial of service attacks: Characterization and implications for CDNs and web sites. In *Proceedings of the International World Wide Web Conference*, pages 252–262. IEEE, May 2002.

[14] J. Jung, V. Paxson, A. W. Berger, and H. Balakrishnan. Fast portscan detection using

IJOART

Copyright © 2014 SciResPub.

ISSN 2278-7763

sequential hypothesis testing. In *Proceedings of IEEE Symposium on Security and Privacy*, Oakland, May 2004.

[15] S. Kandula, D. Katabi, M. Jacob, and A. W. Berger. Botz-4-sale: Surviving organized ddos attacks that mimic flash crowds. In *Proceedings of Symposium on Networked Systems Design and Implementation (NSDI)*, Boston, May 2005.

IJOART

Copyright © 2014 SciResPub.