A Survey of an Adaptive Selective Verification: An efficient Adaptive countermeasure to Thwart DOS Attacks

S.Abarna[1] and A.Muthulakshmi[2]

[1]Department of Computer Science and Engineering, PG Scholar, Dr.Sivanthi Aditanar College of Engineering, Tiruchendur-628215, Tamil Nadu, India

[2]Department of Information Technology, PG Scholar, Dr.Sivanthi Aditanar College of Engineering, Tiruchendur-628215, Tamil Nadu,India

ABSTRACT

In recent years, we have seen the arrival of Distributed Denial-of-Service (DDOS) open source bot based attack tools facilitating easy code enhancement, and so resulting in attack tools becoming more powerful. Developing new techniques for detecting and responding to the latest DDOS attacks often entails using attack traces to determine attack signatures and to test the techniques. However, obtaining actual attack traces is difficult, because the high-profile organizations are typically attacked will not release monitored data as it may contain sensitive information. As opposed to the conventional DOS attacks, Low rate DOS attacker injects a short burst of traffic periodically to fill up the bottleneck buffers right before the expiration of the sender's RTO. This forces the sender's TCP connections to timeout with very low throughput. These attacks are hard to detect and prevent, as most of the DOS attack detection systems are triggered by high- rate traffic. This paper presents the survey of techniques available for detecting Low rate DOS attacks and compares them using various parameters.

1.INTRODUCTION

A denial-of-service attack (DOS attack) is an attempt to make a computer resource unavailable to its intended users. Typically the targets are high-profile web servers, the attack aiming to cause the hosted web pages to be unavailable on the Internet. Denial of service attack programs, root kits, and network sniffers have been around for a very long time. Yet this point-to-point denial of service attacks can be countered by improved tracking capabilities to shut down the source of the problem. However, with the growth of the Internet, the increasingly large number of vulnerable systems are available to the attackers. Rather than relying on a single server, attackers could now take advantage of some hundred, thousand, even tens of thousands or more victim machines to launch the distributed version of the DOS attack. A distributed denial of service attack (DDOS attack) is a large-scale, coordinated attack on the availability of services of a victim system or network resource, launched indirectly through many compromised computerson the Internet .DDOS attacks can seriously impair the Internet service. The first DDOS attack was seen in late June and early July of 1999. The first well-documented DDOS attack appears to have occurred in August 1999, when a DDOS tool called Trinoo was deployed in at least 227 systems, of which at least 114 were on Internet, to flood a single University of Minnesota computer; this system was knocked off the air for more than two days. The first well-publicized DDOS attack in the public press was in February 2000. On February 7, Yahoo! Was the victim of a DDOS during which its Internet portal was inaccessible for three hours. On February 8, Amazon, Buy.com, CNN, and eBay were all hit by DDOS attacks that caused them to either stop functioning completely or slowed them down significantly. Analysts estimated that during the three hours Yahoo was down, it suffered a loss of e-commerce and advertising revenue that amountedto about $500,000. According to book seller Amazon.com, its widely publicized attack resulted in a loss of $600,000 during the 10 hours it was down. During the DDOS attacks, Buy.com went from 100%

availability to 9.4%, while CNN.com's users went down to below 5% of normal volume. The downtime loss was huge.More recently on June 2004, the websites of Google, Yahoo and Microsoft disappeared for hours when their servers were swamped with hundreds of thousands of simultaneous webpage requests that they could not possibly service.These webpage requests are from botnet which consists of thousands of zombie machines. Before the attack, the attacker tried to scan the Internet to find out the vulnerable machines and plant the bot on those machine. Internet relay chat room are used for communication between the attacker and those zombie machines. After the attacker issued the assault command in an Internet relay chat room, the botnet start to generate plenty of web requests which bring down the victim websites .These kinds of botnets are fuelling a growing crime wave against e-commerce in which they are increasingly being offered for hire by hacking groups.Earlier in year 2006, VeriSign experienced attacks on its systems that were larger than anything it had ever seen before.The assaults weren't coming from commandeered "bot" computers, as is common. Instead, its machines were under attack by DNS (domain name system) servers. In this new kind of attack,an assailant would typically use a botnetto send a large number of queries to open DNS servers. These queries will be "spoofed" to look like they come from the target of the flooding, and the DNS server will reply to that networkaddress. Using DNS servers to do their dirty work offers key benefits to attackers. It hides their systems, making it harder for the victim to find the original source of the attack. But more important, reflecting an attack through a DNS server also allows the assault to be amplified, delivering a larger amount of malicious traffic to the target. A single DNS query could trigger a response that is as much as 73 times larger than the request. To protect the DNS servers from abuse and prevent this kind of attack, DNS servers should be configure to only provide DNS services to machines within a trusted domain. Restricting recursion and disabling the ability to send additional delegation information can help prevent DNS-based DDOS attacks. There have been a number of proposals and solutions to the DDOS attacks. However there is still no comprehensive solution which can protect against all known forms of DDOS attacks. This paper tries to analyze and classify the current solutions to the DDOS attack. By examining the pros and cons of each solution,we can know about the effectiveness of the solutions.

2. What makes DDOS attacks possible?

Internet was designed with functionality and not security in mind. The TCP/IP protocol suite, the most widely used protocol suite for data communication assumes that all the hosts participating in the communication have no malicious intent. There is no security built into the internet infrastructure to protect hosts from other hosts not regulating their own behavior. For example, the TCP protocol assumes that hosts will reduce the rate of packet transmission on detecting packet losses due to congestion. If a particular host instead does not respond to the congestion conditions, it can easily overwhelm the intermediate links to the destination. Such design opens up the internet to many opportunities for denial of service attacks. Some features of the internet that make DOS attacks possible are :

2.1 Internet Security is highly dependent

DDOS attacks are launched from hostswhose security has been subverted. No matter how secure a particular host is, it opens itself to the possibility of a DDOS attacks if there are other insecure hosts in the internet which can be used to launch such attacks.

2.2 Difficulty in tracing back the attack to the source

Most of the internet runs on top of the TCP/IP protocol. Theunderlyingprotocol (IP) isbasically connectionless in nature. At each intermediate step from the source tothe destination, the decision about the next host to forward the packet is made. Allsuch routing decisions are made on the basis of the destination address. It is thus possible to generate packets with incorrect source IP addresses and use them to launch Denial of Service attacks. This technique is known as IP spoofing. Users with sufficient privileges on a system have the ability to fabricate such fake packets. Eg in Linux , raw sockets can be created which enable users construct all the packet contents and headers for a given packet.This makes the task of determining the true source of attack very difficult. Apart from the source IP address, the attackers nowadays even randomly change all the headers in an IP datagram, keeping just the destination address constant. This makes dropping of packets based on certain characteristics very difficult, as distinguishing attack packets from legitimate packets becomes difficult. There are also some attacks that rely on illegitimate source addresses to launch a denial of service attack on the hosts whose IP address was used. The Smurf attack is onesuch example. If on detection of an attack, packets are dropped solely on the basis of the IP source addresses, then the hosts whose IP addresses were used for thespoofing will suffer from a denial of service.

2.3 Limited Resources

The infrastructure of the interconnected hosts and networks is comprised of limited resources. Bandwidth processing power and storage capacities are all targets of Denial of Service attacks. If these resources are increased by substantial investments, it just raises the bar on the degree an attack must reach to be effective. Even if the attack is not able to shut down the victim completely, it may waste its resources, reducing the level of quality as seen by theend users, and making the service provider incur heavy financial losses.

2.4 A target rich environment

If the people in the military were to describe the internet today, they would describe it as a target rich environment. There are thousands and thousands of hosts and networks in the internet with vulnerabilities that can be exploited to get access to the machines there. It is therefore easy to gain control of a large number of hosts that can be then used as a spring board to launch DDOS attacks.

2.5 Easier to break systems than to make them

Just as it is easier to destroy a car ,than to make a good car, it is easier to break the networking infrastructure / protocols than to develop them. All the hosts in the internet, including the intermediate routers expect certain packet formats and traffic behavior. Since, at the time of the design of these software, no one foresaw the system being used for malicious purposes. This can lead to unexpected behavior of the network systems as response to unexpected packets.

3.DDOS Bots

3.1 Agobot

Agobot is one of the most popular bots with the Anti-Virus vendor, Sophos [24], listing over 600 different versions. Variants of Agobot include Gaobot, Nortonbot, Phatbot and Polybot. The source code that we studied is the widely available “current” version of Phatbot, written in C++ and provides cross platform capabilities. The bot is structured in a modular way and allows new attacks to be easily added.

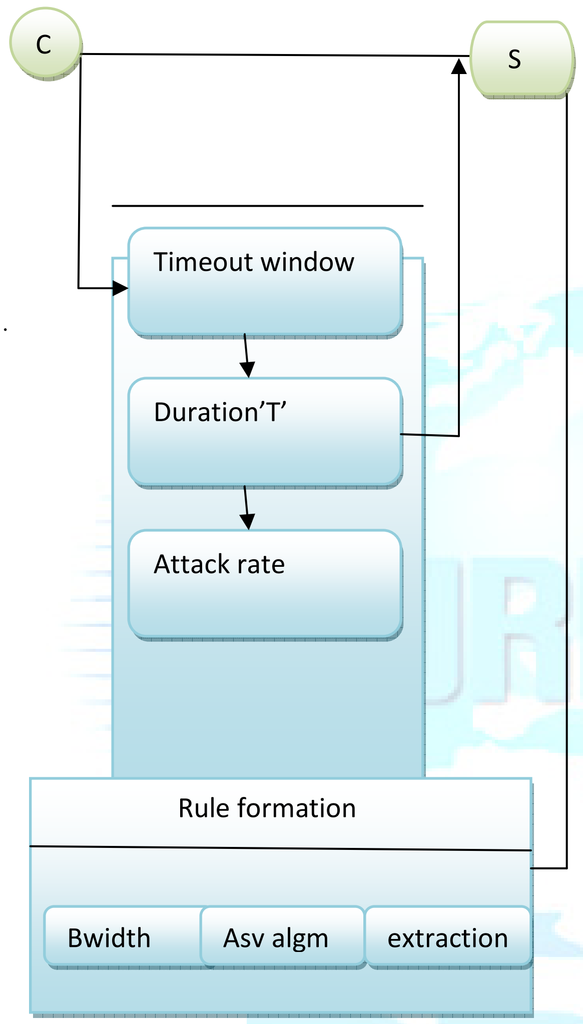
3.2 SDBot

SDBot is another popular bot with over 1800 variants. The widely available version is 0.5b, but only comes with ping and udp flooding tools, whereas the “SYN Flood Edition” includes TCP SYN flooding attacks. SDBot is written in C++ and targets Windows systems.

3.3 RBot

RBot has over 1600 variants. It is also written in C++ and targets Windows systems and masterpassword for scanning and compromising Optix servers.

4.FLOW CHART



.

5.Literature survey

The authorStefan Savage, David Wetherall,AnnaKarlin and Tom Anderson[1]in the year 2000 work is motivated by the increased frequency and sophistication of denial-of-service attacks and by the difficulty in tracing packets with incorrect, or "spoofed", source addresses. In this paper we describe a general purpose traceback mechanism based on probabilistic packet marking in the network. Our approach allows a victim to identify the network path(s) traversed by attack traffic without requiring interactive operational support from Internet ServiceProviders (ISPs).Unfortunately, mechanisms for dealing with denial-of-service have not advanced at the same pace. Most work in this area has focused on *tolerating* attacks by mitigating their effects on the victim This approach can provide an effective stop-gap measure, but does not eliminate the problem nor does it discourage attackers.

Ingress filtering

One way to address the problem of anonymous attacks is to eliminate the ability to forge source addresses. One such approach, frequently called *ingress filtering*, is to configure routers to block packets that arrive with illegitimate source addresses.This requires a router with sufficient power to examine the source address of every packet and sufficient knowledge to distinguish between legitimate and illegitimate addresses. Consequently, ingress filtering is most feasible in customer networks or at the border of Internet Service Providers (ISP) where address ownership is relatively unambiguous and traffic load is low.

The author Cynthia Dwork, Andrew Goldberg[2] in the year 2003introduced an Unsolicited commercial e-mail, or spam, is more than just an annoyance. At two to three *billion* daily spam worldwide, or close to 50% of all e-mail, spam incurs huge infrastructure costs, interferes with worker productivity, devalues the internet, and is ruining e-mail. CPU-bound pricing functions suffer from a possible mismatch in processing speeds among different types of machines (desktops *vs.* servers), and in particular between old machines and the presumed new, top of the line, machines that could be used by a high-tech spam service techniques.

Abstract Algorithm

The algorithm used a modifiable array A, initialized for each trial, of size j, Aj ,w > b bits (recall that b is the number of bits in a memory block, or cache line)Before we present the abstract algorithm, we introduce a few hash functions H0;H1;H2;H3, of varying domains and ranges, that we model as idealized random functions (random oracles). The function H0 is only used during initialization of a path. It takes as input a message m, sender's name (or address) S, receiver's name (or address) R, and date d, together with a trial number k, and produces an array A. The function H1 takes an array A as input and produces an index c into the table .

The author Abraham Yaar,Adrian Perrig, Dawn Song[3] in the year 2004, presented SIFF, a novel design that addresses the DDoS flooding problem in a future Internet setting, without relying on any of the above assumptions. Using this design as a basis, we also present a countermeasure that may be deployed in the current Internet, assuming that client and server software is updated. SIFF does not require any of the above assumptions of previous countermeasures.

Privileged Packet Flooding

SIFF mitigates the impact of flooding (or bandwidth starvation) DOS attacks by isolating and protecting established privileged communication from unprivileged communication and enabling the receiver to downgrade privilege. In this section, we analyze the robustness of our scheme against floods of privileged packets with forged capabilities.We seek to answer a simple question: "How prevalent are denial-of-service attacks in the Internet today?".Our motivation is to understand quantitatively the nature of the current threat as well as to enable longerterm analyses of trends and recurring patterns of attacks. We present a new technique, called "backscatter analysis", that provides an estimate of worldwide denial-of-service activity. We use this approach on three week-long datasets to assess the number, duration and focus of attacks, and to characterize their behavior. During this period, we observe more than 12,000 attacks against more than 5,000 distinct targets, ranging from well known ecommerce companies such as Amazon and Hotmail to small foreign ISPs and dial-up connections.

Backscatter analysis

In this paper they have presented a new technique,"backscatter analysis," for estimating denial-of-service attack activity in the Internet. Using this technique, we have observed widespread DOS attacks in the Internet, distributed among many different domains and ISPs.

The author D. K. Y. Yau, J. C. S. Lui, F. Liang, and Y. Yam,[4] VOL. 13, NO. 1,in the year FEB2005 introduced a server-centric approach to protecting a server system under DDOS attacks. The approach limits the rate at which an upstream router can forward packets to the server, so that the server exposes no more than its designed capacity to the global network. In allocating the server capacity among the upstream routers, we studied a notion of levelmax-min fairness, which is policy-free and hence easy to deploy and manage.

BaselineAlgorithm

The author first presented a baseline algorithm in which each router throttles traffic for by forwarding only a fraction of the traffic. The fraction is taken to be one when no throttle for is in effect. In adjusting according to current server congestion, the algorithm mimics TCP congestion control.Specifically, is reduced by a multiplicative factor when is congested and sends the router a *rate reduction signal*.

Fair Throttle Algorithm

The baseline algorithm is not fair because it penalizes all routers equally, irrespective of whether they are greedy or well behaving.We nowpresent a fair throttle algorithm that installs at each router in a *uniform* leaky bucket rate (i.e., the throttle rate) at which the router can forward traffic.

The author MARTIN ABADI,[5] in the year 2005 had suggested that such abuse may be discouraged by introducing an artificial cost in the form of a moderately expensive computation. Thus, the sender of an e-mail might be required to pay by computing for a few seconds before the e-mail is accepted. Unfortunately, because of sharp disparities across computer systems, this approach may be ineffective against malicious users with high-end systems, prohibitively slow for legitimate users with low-end systems, or both. Starting from this observation,we research moderately hard functions that most recent systems will evaluate at about the same speed. For this purpose, we rely on memory-bound computations.Making problems even harder In addition to grouping problems, other techniques may contribute to making challenges harder. We briefly sketch and speculate on two such techniques in this section. *Omitting bits from problems.* One can often make problems harder by omitting some bits from them. In particular, R could omit some bits of the challenge xk, of the description of the function F(), or both, and S would need to guess or reconstruct the missing bits in finding x0. For instance, R could present the full xk and a checksum of the path from xk to x0, and R could tell S that F() has a definition of the form.

The author David Moore *CAIDA[6],in the year 2006* motivation is to understand quantitatively the nature of the current threat as well as to enable longerterm analyses of trends and recurring patterns of attacks. We present a new technique, called "backscatter analysis", that provides an estimate of *worldwide* denial-of-service activity. We use this approach on three week-long datasets to assess the number, duration and focus of attacks, and to characterize their behavior. During this period, we observe more than 12,000 attacks against more than 5,000 distinct targets, ranging from well known ecommerce companies such as Amazon and Hotmail to small foreign ISPs and dial-up connections.

Ingress and egress filtering

It gives the practical difficulty of ensuring that all networks are filtered, other work has focused on developing tools and mechanisms for tracing flows of packets through the network independent of their ostensibly claimed source address.

The author Xin Liu, Xiaowei Yang, Yanbin Lu[7] in the year 2008 presented the design and implementation of a filter based DOS defense system (StopIt) and a comparison study on the effectiveness of filters and capabilities. Central to the StopIt design is a novel *closed-control*, *open-service* architecture: any receiver can use Stop It to block the undesired traffic.It receives, yet the design is robust to various strategic attacks from millions of bots, including filter exhaustion attacks and bandwidth flooding attacks that aim to disrupt the timely installation of filters.Our evaluation shows that StopIt can block the attack traffic from a few millions of attackers within tens of minutes with bounded router memory. We compare StopIt with existing filter-based and capability based DOS defense systems under simulated DOS attacks of various types and scales. Our results show that StopIt outperforms existing filterbased systems, and can prevent legitimate communications from being disrupted by various DOS Flooding attacks.

circular buffer of bloom filters

A flow cache can be implemented using a circular buffer of bloom filters, a technique also used in . A bloom filter has no false negatives. A router can always catch an attack flow in its flow cache as long as the round trip delay is less than TF.We are not concerned with the small percentage of a false positive, because it occurs rarely and randomly, and can only happen when a malicious host Hd wants to block its own traffic.

The author Xiaowei Yang, *Member,* DavidWetherall[8], in the year 2008,motivated the capability approach to network denial-of-service (DOS) attacks, and evaluate the TVA architecture which builds on capabilities. With our approach, rather than send packets to any destination at any time, senders must first obtain "permission to send" from the receiver, which provides the permission in the formof capabilities to those senders whose traffic it agrees to accept. The senders then include these capabilities in packets. This enables verification points distributed around the network to check that traffic has been authorized by the receiver and the path in between, and hence to cleanly discard unauthorized traffic. to new applications over the past few years. Perversely, thishas happened as a rational response of network and systemadministrators needing to cope with the consequences of the Internet's openness. The Internet architecture is vulnerable to denial-of-service (DOS) attacks, where any collection of hostswith enough bandwidth (e.g., using machines taken over by a virus attack) can disrupt legitimate communication between any pair of other parties, simply by flooding one end or the otherwith unwanted traffic.TopologiesSimulations of TVA require knowing the path identifier distribution of legitimate users and attackers seen at a bottleneck. Unfortunately, this information is not readily available. Parameters. For each sub-topology, we randomly mark d% of edge Aas attackers, with d ranging from 10, 20, 40,to 80.

The authorMusab AlTurki, Jos_e Meseguer, Carl A. Gunter, SecRet[9] in the year 2009 evaluatedAdaptive Selective Verification (ASV) protocol was recently proposed as an effective and e client DOScountermeasure within the shared channel model, in which clients and attackers probabilistically share communicationbandwidth with the server. ASV has been manually shown to satisfy some desirable availabilityand bandwidth consumption properties. Due to the probabilistic nature of the protocol and its underlyingattacker model, it is intrinsically difficult to build a faithful model of the protocol with which one mayautomatically verify its properties. This paper fills the gap between manual analysis and simulation-basedexperimental analysis of ASV, through automated formal analysis. We describe a formal model of ASVusing probabilistic rewrite theories, implemented in a probabilistic extension of Maude, and show how it canbe used to formally verify various characteristics of ASV through automated statistical quantitative modelchecking analysis techniques. In particular, we formally verify ASV's connection concede theorem and aslightly more general bandwidth consumption theorem of ASV.Formal verification techniques now need to deal with real-time, probabilities, and quantitative properties. This means that the standard methods used forverifying secrecy, authenticity and integrity properties, such as invariants and reachability analysis, standard temporal logic and model checking, or various specialized theorem proving schemes may not be directly usable in the realm of availability properties.

5. CONCLUSION

DDOS attacks are quite advanced and powerful methods to attack a network system to make it either unusable to the legitimate users or downgrade its performance.They are increasingly mounted by professional hacks in exchange for money and benefits. This survey examines the possible solutions to this problem, provides a taxonomies to classify those solutions and analyzes the feasibility of those approaches. It compares various techniques available for detection of LDOS attack. It also highlights the issues present in currently available DOS detection mechanisms.

6. REFERENCES

1. S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Practical network support for IP traceback," in *Proc. ACM SIGCOMM*, 2000, pp.295–306.
2. C. Dwork, A. Goldberg, and M. Naor, "On memory-bound functions for fighting spam," in *Proc. CRYPTO*, 2003, pp. 426–444.
3. A. Yaar, A. Perrig, and D. X. Song, "SIFF: A stateless internet flow filter to mitigate DDOS flooding attacks," in *Proc. IEEE Symp. SecurityPrivacy*, 2004, pp. 130–143.4. .D. K. Y. Yau, J. C. S. Lui, F. Liang, and Y. Yam, "Defending against distributed denial-of-service attacks with max-min fair server-centric router throttles, "*IEEE/ ACMTrans. Netw.*, vol. 13, no. 1, pp. 29–42,Feb. 2005.5. M. Abadi, M. Burrows, M. Manasse, and T.Wobber, "Moderately hard, memory-bound functions," *Trans. Internet Technol.*, vol. 5, no. 2, pp.299–327, 2005.6. D. Moore, C. Shannon, D. J. Brown, G. M. Voelker, and S. Savage, "Inferring internet denial-of-service activity," *Trans. Comput. Syst.*, vol.24, no. 2, pp. 115–139, 20067.. X. Liu, X. Yang, and Y. Lu, "To filter or to authorize: Network-layer DoS defense against multimillion-node botnets," *Comput. Commun.Rev.*, vol. 38, no. 4, pp. 195–206, 2008.
8. X. Yang, D. Wetherall, and T. Anderson, "TVA: A DoS-limiting network architecture," *IEEE/ACM Trans. Netw.*, vol. 16, no. 6, pp.1267–1280, Dec. 2008.
9. M. AlTurki, J. Meseguer, and C. A. Gunter, "Probabilstic modelingand analysis of DoS protection for the ASV protocol," *Electron. NotesTheoret. Comput. Sci.*, vol. 234, pp. 3–18, 2009.