An approach to detect and prevent denial of service attacks and worms using distributed denial-of-service detection mechanism

A.Sudhakar[1],Uma Devi Dava[2]
A.Sudhakar,Post Graduate Student of Sri Mittapally College of Engineering & Technology.

Mrs.Uma Devi Deva,Associate Professor of Sri Mittapally College of Engineering & Technology.

Abstract

In the networking systems the flow of information is the most important service. It is clear that a simple self-propagating worm can quickly spread across the Internet and cause severe damage to our society. Facing this great security threats like Denial-of-Service (DoS) and Worms, we need to build an early detection system that can detect the presence of a worm in the Internet as quickly as possible in order to give people accurate early warning information and possible reaction time for counteractions. To avoid these types of threats more effective approaches are required to counter the threats. This requirement has motivated us to create novel mechanism for effective early detection and prevention of Worms and DoS attacks at the router level within an Internetworking infrastructure. Here our system presents a "trend detection" methodology to detect a worm at its early propagation stage by using Kalman filter. In addition, for uniform-scan worms such as Code Red, we can effectively predict the overall vulnerable population size, and estimate accurately how many computers are really infected in the global Internet based on the biased monitored data. Also in this system we propose a domain-based approach, the mechanism that combines both stateful and stateless signatures to provide early detection of DoS attacks and Worms, therefore, protect the network. In this project we are using the novel Distributed DoS Detection Mechanism (DiDDeM) using the Kalman filter mechanism to detect both Worms and DoS attacks at the early stage.

Keywords: Denial of Service, Detection, Worms.

1. Introduction

A denial-of-service (DoS) attack is a malicious attempt by a single person or a group of people to cripple an online service. The impact of these attacks can vary from minor inconvenience to users of a website, to serious financial losses for companies that rely on their on-line availability to do business. On February 9, 2000, Yahoo, eBay, Amazon. com, E*Trade, ZDnet, Buy.com, the FBI, and several other Web sites fell victim to DoS attacks resulting in millions of dollars in damage and inconvenience [6][5]. As emergency and essential services become more reliant on the Internet as part of their communication infrastructure, the consequences of denial-of-service attacks could even become life-threatening. After the September 11 terrorist attack in the US, there is a growing concern that the Internet may also fall victim to terrorists. There are many indications that since September 11, the number of DoS attacks have greatly increased [7]. Recently the same fate befell the music industry web site www.riaa.org [20]. Sophisticated tools to gain root access to other people's machines are freely available on the Internet. In computing, a denial-of-service attack (DoS attack) or distributed denial-of-service attack (DDoS attack) is an attempt to make a machine or network resource unavailable to its intended users. Although the means to carry out, motives for, and targets of a DoS attack may vary, it generally consists of the efforts of one or more people to temporarily or indefinitely interrupt or suspend services of a host connected to the Internet.

Perpetrators of DoS attacks typically target sites or services hosted on high-profile web servers such as banks, credit card payment gateways, and even root nameservers. The term is generally used relating to computer networks, but is not limited to this field; for example, it is also used in reference to CPU resource management.One common method of attack involves saturating the target machine with external communications requests, such that it cannot respond to legitimate traffic, or responds so slowly as to be rendered effectively unavailable. Such attacks usually lead to a server overload. In general terms, DoS attacks are implemented by either forcing the targeted computer(s) to reset, or consuming its resources so that it can no longer provide its intended service or obstructing the communication media between the intended users and the victim so that they can no longer communicate adequately.

A computer worm is a self-replicating computer program similar to a computer virus. A virus attaches itself to, and becomes part of, another executable program; however, a worm is self-contained and does not need to be part of another program to propagate itself. They are often

designed to exploit the file transmission capabilities found on many computers. A worm uses a network to send copies of it to other systems and it does so without any intervention. In general, worms harm the network and consume bandwidth, whereas viruses infect or corrupt files on a targeted computer. Viruses generally do not affect network performance, as their malicious activities are mostly confined within the target computer itself. Network worms are malicious programs that spread automatically across networks by exploiting vulnerabilities that affect a large number of hosts. Because of the speed at which worms spread to large computer populations, countermeasures based on human reaction time are not feasible. Therefore, recent research has focused on devising new techniques to detect and contain network worms without the need of human supervision. In particular, a number of approaches have been proposed to automatically derive signatures to detect network worms by analyzing a number of worm-related network streams. Most of these techniques, however, assume that the worm code does not change during the infection process. Unfortunately, worms can be polymorphic. That is, they can mutate as they spread across the network. To detect these types of worms, it is necessary to devise new techniques that are able to identify similarities between different mutations of a worm.

To meet this challenge, we have proposed a system for early defense against DoS. At the heart of this system is a novel Distributed DoS Detection Mechanism (DiDDeM) providing the means by which DoS attacks are detected early, beyond the perimeter of the network under attack, so as to enable an early propagated response to block the attack through network routers, particularly those close to the attack sources. This approach has been implemented as a prototype, further details of which are discussed in Section VI of this paper and [4]. This paper is focused mainly on the presentation of the new mechanism design. The DiDDeM presented in this paper has four novel contributions. First, it does not require much extra computational load for DoS detection as only a small proportion of packets compared with the total throughput of network routers are analyzed. Second, no state information about the networks under protection needs to be held, thus requiring little extra storage load for the detection. Third, reports of attacks may relate to several packets rather than “one packet, one alert” techniques employed by traditional countermeasures, so as to further reduce workload on the networks protected.

2. Literature Survey

Flooding DoS attacks are distinct from other attacks, for example, those that execute malicious code on their victim, in that they require a large volume of traffic, and it is this continuing stream of data that prevents the victim from providing services to legitimate users. It is the mass of all attack packets directed at the victim, which poses the threat, rather than the contents of the packets themselves.Flooding DoS attacks are problematic due to their subversion of normal network protocols. As such, it is these attacks that pose the greatest problem in today’s network infrastructures. Subverting the use of protocols, such as transmission control protocol (TCP) or user datagram protocol (UDP), enables an attacker to disrupt on-line services by generating a traffic overload to block links or cause routers near the victim to crash [5]. The packets involved in these attacks are high-volume without being conspicuous or easily traceable. For example, TCP SYN flooding specifically targets weaknesses in the TCP protocol, particularly, the three-way handshake, to achieve its aim (amongst others—[6], [7]). A number of approaches have been posited to counter the DoS problem. For example, [8] proposes stronger authentication between communicating parties across a network. Alternatively, [9] suggests that network resources should be divided into classes of service, where higher prices would attract less traffic and ensure that an attacker could not afford to launch an attack. Alternatively, [10] suggests that the routing infrastructure should be more robust by securing servers in the first place. These approaches do not provide a total solution. For example, authentication, while attempting to prevent DoS, leaves itself open to such an attack due to the computational load required for the defense. Payment approaches assume that a consumer is willing to pay for different levels of service, and also consumers could be forced to pay heavy financial penalties (refusing the payment itself may lead to DoS to these consumers) if their computers are compromised and abused by an attacker. Finally, it is often poor software development practices due to the pressure of getting a product to market, which lead to the release of server applications that could be subverted. These problems have led to the rise of traffic monitoring approaches that fall into two categories: statistical monitoring and adaptation of congestion algorithms. Statistical monitoring of networks, such as [11], [12], observes a network and detects upsurges in traffic of a particular type or for system compromise. An advantage of this approach is that one alert may cover a number of attack packets, thus reducing the network load caused by the reporting of events.

Alternatively, congestion algorithms are adapted for detection of DoS attacks. Approaches such as [7], [13], and [14] use existing congestion techniques, where routers deal with upsurges in traffic to ensure quality of service, to detect DoS. These approaches have the benefit of being able to detect an attack in the routing infrastructure, thus being able to halt the attack before it reaches its intended victim. These two approaches are not ideal solutions. Statistical approaches require human intervention to monitor the networks for upsurges, so they are both labor intensive and inefficient. The congestion adaptation approaches may only apply simplistic signatures so as to not impede on the throughput of traffic. In addition, the approaches such as [7] and [13] require that state information be held on routers. This information is too computationally exhaustive to be effective within the routing infrastructure. Therefore, a new approach is required that can provide early detection of DoS attacks by combining and making use of the advantages of both the statistical and adaptation of congestion algorithm approaches. In this way, the following benefits may be achieved: computational load on networks is reduced by analyzing fewer packets; no state information is required about the systems/services under protection; alerts may span many attack packets; and the defense may be placed within the routing infrastructure.

3.History Of DiDDeM

DiDDeM utilises the two types of signature to meet the requirements for early detection of TCP SYN flood attacks.The three key elements of DiDDeM are: the DiDDeM domain, the pre-filter (PF) detection node, and the command and control (C2) server. A DiDDeM domain comprises a number of PF nodes and a C2 server. A PF is a key element in the detection of denial-of-service attacks. A PF is located on a router and utilises the congestion algorithm to infer stateful information from stateless information for detection. A C2 server manages its DiDDeM domain, communicates with C2 servers in other DiDDeM domains, provides a central station for attack analysis, and co-ordinates a response to an attack.

A. DiDDeM Design Goals

The DiDDeM design goals provide the requirements of the system. The two principal goals are as follows:

• Scalability. Scalability of the system can be measured in two ways [12]. First, in terms of the ability to be deployed within the routing infrastructure, the system must meet the demands of both large LANs and the Internet itself. Second, scalability is measured in terms of the volume of network traffic processed by the system.

• High-speed, large-volume monitoring. A large amount of traffic must be considered within very tight temporal constraints within the routing infrastructure. The filtering of a large volume of traffic is achieved by identifying patterns in the TCP/IP headers, which greatly reduces the processing load of monitoring [13, 14].Other requirements met by DiDDeM include: acceptable performance degradation, inference of stateful information in a stateless environment, and real-time notification and response. These requirements form the focus of the DiDDeM system and are addressed in the remainder of this section.

B. DiDDeM System

If denial-of-service attacks are allowed to reach their intended target, the attack is able to succeed in its objective of denying resources to legitimate users. The objective of the DiDDeM system is to provide early detection and response to denial-of-service attacks before they denigrate the services and resources of the target.The DiDDeM system integrates co-operative DiDDeM domains. The domain is comprised of two types of element, a C2 and a number of PFs. The C2 acts as a server to a number of PFs located within the domain. The services that the C2 provides are attack analysis through collation of PF reported events, management of the domain, attack response, and authentication of PFs within a domain. It is also responsible for intra- and inter-DiDDeM domain communications. A PF is responsible for attack detection through stateful and stateless signatures and is located within the routing infrastructure.

The domain allows a two stage detection and analysis process. The first stage of detection is via traffic analysis.First, PFs detect surges in network traffic that indicate that a possible denial-of-service attack is under way. A selection of packets are inspected to determine whether the packets form part of an attack. If so, a message is sent to the C2 server with details of the attack. Second, the C2 compares this message with its posessed information and that received from C2s in other relevant domains, and then issues a response as required.The first stage is the detection of possible attacks by the PFs. The objective of a PF is to identify the stateful signatures of denial-of-service attacks in stateless way, which indicate the possibility that an attack is under way against a particular host, domain, or network. Thus, they detect rises in traffic passing through the router heading in a particular direction, i.e. the target of an attack. If an attack is detected,the alert is then confirmed or discarded by applying stateless signatures to the packets in question.

The stateful detection of attack traffic flows is achieved by interacting with the

congestion algorithm used by the router.This algorithm already detects and responds to upsurges in traffic. During periods of heavy traffic, traffic is first queued by the router to control the traffic load on the network. By queuing the traffic, the router is able to implement a volume threshold. Once the threshold is surpassed, it drops packets rather than relay them across an already congested link.Rather than merely drop packets when a queue threshold is surpassed, a PF located on a domain ingress router picks packets to be dropped and inspects a statistical sample of those dropped packets to ascertain the direction of traffic flow. If the destination address of all the packets in the statistical sample match, it is likely that they are part of a large flow of traffic towards a destination, providing a stateful signature of a denial-of-service attack. In this way,we can infer unusual rises in traffic against a particular host,domain, or network without holding any state information on those networks.Once a stateful signature of a denial-of-service attack is inferred, stateless signatures are then applied to a sample of packets to confirm the attack. If an attack is confirmed through the stateless signature, an alert is generated and forwarded to the controlling C2. This alert contains details of the reporting PF, the destination of the traffic, and the attack signature matched.Employing a PF approach enables the reduction of computational overheads in three ways. First, the PF utilizes the already existing congestion algorithm for detection. The stateful signature can be inferred by the upsurge in traffic volumes that cause the congestion algorithm to drop packets.Second, only a small number of packets, i.e. those that are dropped, need be inspected by the PF. The number of packets during a denial-of-service attack may be high, and to perform signature analysis on each and every packet adds computational overheads. Therefore, the packets are inspected to ascertain direction of flow. Finally, a message reporting the alert is sent to the C2 rather than redirecting the packets themselves. In traditional IDSs, each packet on the network is compared to a database of signatures, and as many as 300 signatures are applied to each and every packet traversing the network [15]. Every time a signature is matched with that in the IDS database, an alert is generated.By using a statistical sample of packets, the ratio of packets to alerts can be further enhanced to reduce computational load.

The second stage of detection occurs at the C2 server overseeing the DiDDeM domain. The C2 receives the alert message from a subordinate PF. If the attack is confirmed, a response directive, such as blocking all traffic matching a particular signature, is passed to the PF from the C2.To provide holistic security, the C2 passes information about the attack to other DiDDeM domains. The C2 identifies adjacent domains joined via ingress filter PFs. Once an alert is received from the PF, the server analyses the information.If an attack is confirmed, the C2 passes an alert to the C2s of adjacent domains. Each such adjacent C2 is identified by the ingress link of the reporting PFs.

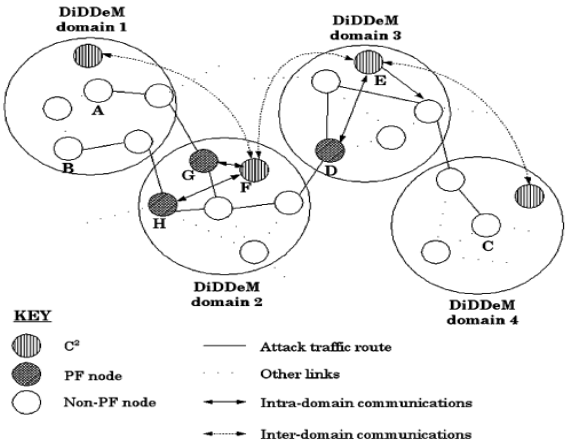

Fig. 1. DiDDeM domain cooperation example.

To assist a better understanding of the DiDDeM-based system described above, Fig. 1 illustrates the cooperative process among DiDDeM domains during a DoS attack, which will be detailed in the subsequent sections. Fig. 1 only highlights the servers together with the PF nodes (i.e., routers with a PF running on each of them) needed to tackle the attack.The explanation of the process is given as follows.

• Nodes and in DiDDeM domain 1 launch a TCP SYN flood as part of a distributed DoS attack against node in DiDDeM domain 4. Due to the network topology, the attack traffic passes through DiDDeM domain 2 undetected as congestion does not occur.

• As the attack traffic enters into DiDDeM domain 3, PF node detects an upsurge in network traffic against one destination, and further signature analysis has confirmed that the upsurge is caused by a DoS attack. The PF issues an alert about the attack to the of the domain . Having confirmed that no similar response has been generated recently against the reported attack, issues a response directive to to block incoming attack packets.Suppose that the arrival rate of attack packets being blocked by exceeds a set threshold. reports this to,while continuing the blockage. then identifies the adjacent DiDDeM domain 2, where attack packets come from, and sends a joint response request to the of domain 2,, for blocking attack packets.

• Upon receipt of the request from , performs the same response actions as has done, to block

incoming attack packets from domain 1 through PF's, and , and to inform the of domain 1 to take the same response actions if necessary. As the attack traffic is being blocked in domain 2, the amount of attack traffic entering into domain 3 reduces significantly. Once the amount is no longer over the set threshold, in domain 3 ends the blockage of attack traffic.The above cooperative process illustrates that the attack can be traced back to its originating domain, DiDDeM domain 1,and contained within the domain.

In summary, the DiDDeM design exhibits a number of advantages. First, the design offers the scalability that enables the DiDDeM system to be widely deployed to effectively protect the network infrastructure. Second, by using the filtering process, signature analysis, and stateful information are achieved statelessly. Furthermore, by utilizing congestion algorithms currently employed by routers, packet inspection only occurs during periods of high traffic volume, which indicates a signature of a possible attack. These help to reduce the computational and storage overheads placed on routers, particularly during a DoS attack. Third, only one message per signature application is issued by a PF covering a number of packets ensuring the efficient use of network resources. Fourth,domains cooperate to provide a holistic approach to ensure the trace-back and containment of attack sources.

4. IMPLEMENTATION

To demonstrate the way in which stateful signature detection is achieved, a first-in, first-out (FIFO) queue is used as the basis of a DiDDeM router case study within the network simulator ns2 [18]. FIFO queues are often used in routing as it is a simple yet robust algorithm for mitigating congestion at a particular router.Within the prototype, the routing algorithm first checks whether an average queue size is less than the maximum threshold of the queue. If it is, the router accepts the new packet, enqueues the packet to the back of the FIFO queue, and then dequeues the packet when it finally reaches the front of the queue. This packet is transmitted to the next hop en route to its destination. This represents normal network load without any congestion occurring at the router.If the average queue size exceeds the maximum threshold,congestion occurs. Prior to dropping packets, as would occur, the packets to be dropped are inspected by the corresponding PF. Within the prototype, this is set to two packets being compared.However, this variable may be set to a different number by a system administrator to ensure effective packet comparisons.A packet to be dropped is drawn from the queue based on the congestion algorithm used by the router. The destination address is compared with the previous inspected packet's destination address, which is stored by the router for this type of comparison. If both destination addresses match, then these packets are passed for stateless signature analysis. If they do not match, the current destination address is stored for comparison with the next packet and the current packet is dropped.In a case study implemented in ns2 and detailed in [4], [15],the above method for the derivation of stateful information was implemented. During the simulation, approximately 19 500 attack packets were directed at a victim node by two attacking nodes. This represents an attack consisting of approximately 1000 packets/s. Once the congestion algorithm was invoked by the router, 798 attacks and legitimate packets were to be dropped. Of this number, 742 packets were actual attack packets while the remainder were legitimate traffic. Therefore,out of a total of 19 500 attack packets, only about 4% of this volume was inspected. The 697 packets detected out of the 742 inspected by the DiDDeM-enhanced router ensures a 94% detection rate. In addition, only two legitimate packets were detected as attack packets, thus providing a false positive rate of 4%. In a simulation of stateless signature detection, stateless TCP SYN flood signatures were applied to 432 packets, of which 50 were actual attack packets. For TCP SYN flood detection, packets were searched for instances of SYN flags within the header, which are indicative of an attack. If a flag is found, it is compared with other packets within the stream to ascertain whether neighboring flags also have the same flag set. For example, the stream was searched for flags within three packets of one another. The 50 attack packets and eight begin packets were all identified, providing a 100% detection rate.

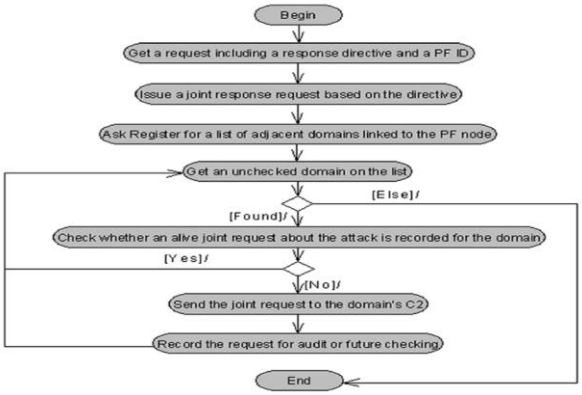

Fig2: Operation process for PF response

5. CONCLUSION AND FUTURE WORK

The flow of information is the most valuable commodity for organizations and users alike, and DoS attacks pose a great threat to this flow. These attacks are highly prevalent despite the widespread deployment of network security countermeasures such as firewalls and intrusion detection systems. Current countermeasures find DoS extremely problematic, therefore,a number of other approaches have been proposed to counter the problem. However, these approaches are not without their problems, so a new approach to more effective detection and prevention of DoS attacks is required.This requirement has motivated us to propose the novel distributed mechanism, DiDDeM, for effective early detection and prevention of DoS attacks. In this paper, we have demonstrated that the DiDDeM makes use of stateful and stateless signatures in conjunction for attack detection, which differs from the other related work that mainly employs one of the two signature approaches. The main benefit from the combination of the two approaches is that not all malicious packets have to be inspected in order to ascertain the presence of an attack,thus improving detection efficiency and making attack detection feasible within the routing infrastructure. Moreover, the DiDDeM offers a novel, distributed and scalable approach to attack responses.utilizing the DiDDeM within an organizational boundary may enhance the detection and prevention of any threat as these programs require a high-volume of traffic during their spreading periods.

References

[1] R. Lippmann, J.W. Haines, D. J. Fried, J. Korba, and K. Das, “The 1999 DARPA off-line intrusion detection evaluation,” Comput. Netw., vol. 34, pp. 579–595, 2000.
[2] D. Moore, G. M. Voelker, and S. Savage, “Inferring Internet denial-ofservice activity,” in Proc. 10th Usenix Security Symp.,Washington, DC, 2001.
[3] R. Richardson, “The eighth annual CSI/FBI computer crime and security survey 2004,” Comput. Security Inst./Federal Bureau of Investigation, Tech. Rep., 2003.
[4] J. Haggerty, T. Berry, Q. Shi, and M. Merabti, “DiDDeM: A system for early detection of TCP SYN flood attacks,” in Proc. GLOBECOM 2004, Dallas, TX, Nov.–Dec. 29–3, 2004, pp. 2037–2042.
[5] T. M. Gil and M. Poletto, “MULTOPS: A data-structure for bandwidth attack detection,” in Proc. USENIX Security Symp., Washington, DC, 2001.
[6] C. Douligeris and A. Mitrokotsa, “DDoS attacks and defense mechanisms: Classification and state of the art,” Comput. Netw., vol. 44, pp. 643–666, 2004.
[7] A. Kazmanovic and E. W. Knightly, “Low-rate TCP-targeted denial of service attacks,” in Proc. Symp. Commun. Arch. Protocols, Karlesruhe, Germany, 2003, pp. 345–350.
[8] C. Meadows, “A cost-based framework for analysis of denial of service in networks,” J. Comput. Security, vol. 9, pp. 143–164, 2001.
[9] J. C. Brustoloni, “Protecting electronic commerce from distributed denial- of-service attacks,” in Proc. WWW2002, Honolulu, HI, 2001, pp. 553–561.
[10] P. Papadimitratos and Z. J. Haas, “Securing the Internet routing infrastructure,” IEEE Commun. Mag., vol. 40, pp. 76–82, Oct. 2002.
[11] S. Shyne, A. Hovak, and J. Riolo, “Using active networking to thwart distributed denial of service attacks,” in Proc. IEEE Aerosp. Conf., Big Sky, MT, 2001, pp. 3/1103–3/1108.
[12] D. Sterne, K. Djahandari, R. Balupari, W. La Cholter, B. Babson, B. Wilson, P. Narasimhan, and A. Purtell, “Active network based DDoS defense,” in Proc. DARPA Active Netw. Conf. Expo., San Francisco, CA, 2002, pp. 193–203.
[13] J. Ioannidis and S. M. Bellovin, “Implementing pushback: Router-based defense against DDoS attacks,” in Proc. Netw. Distrib. Syst. Security Symp., San Diego, CA, 2002.
[14] A. Hussain, J. Heodemann, and C. Papadopoulos, “A framework for classifying denial of service attacks,” in Proc. Symp. Commun. Arch. Protocols, Karlesruhe, Germany, 2003, pp. 99–110.
[15] J. Haggerty, Q. Shi, and M. Merabti, “Statistical signatures for early detection of flooding denial-of-service attacks,” in Proc. IFIP/SEC, Chiba, Japan, May-Jun. 30–1, 2005, pp. 327–342.
[16] C. Krugel and T. Toth, “Distributed pattern detection for intrusion detection,” in Proc. Netw. Distrib. Syst. Security Symp., San Diego, CA, 2002.
[17] Y. Zhang and V. Paxson, “Detecting backdoors,” in Proc. USENIX Security Symp., Denver, CO, 2000.
[18] S. McCanne and S. Floyd. (1998) NS (Network Simulator). [Online]. Available: http: //www-nrg.ee.lbl.gov/ns/