

A Survey of DDoS Defense Mechanisms

Antonio Challita Mona El Hassan Sabine Maalouf Adel Zouheiry

Department of Electrical and Computer Engineering
American University of Beirut

{asc04,mhe03,sem05,atz00}@aub.edu.lb

Abstract – In this paper we overview the different types of DDoS attacks, present recent DDoS defense methods as published in technical papers, and propose a novel approach to counter DDoS. Based on common defense principles and taking into account the different types of DDoS attacks, we survey defense methods and classify them according to several criteria. We propose a simple-to-integrate DDoS victim based defense method, Packet Funneling, which aims at mitigating an attack’s effect on the victim. In this approach, heavy traffic is “funneled” before being passed to its destination node, thus preventing congestion at the node’s access link and keeping the node on-line. This method is simple to integrate, requires no collaboration between nodes, introduces no overhead, and adds slight delays only in case of heavy network loads.

I. INTRODUCTION

By targeting victims such as Yahoo, Amazon, and CNN, Distributed Denial of Service (DDoS) attacks have succeeded in interrupting Internet services and causing huge financial losses [1]. A DDoS attack fills the network pipe of the victim server by an overwhelming amount of packets, thus consuming all of the network’s available bandwidth and obstructing normal traffic flow. To launch the attack, the attacker exploits a number of unaware intermediate hosts and implements certain easy-to-use techniques [1]. DDoS attacks are easy to launch, while most detection and response approaches in use today remain incapable of eradicating the problem. In this paper we overview the different types of attacks, list common principles used in defense mechanisms, and survey the most recent proposed methods which tackle DDoS attacks. We also classify the defense methods and propose our own defense method, commenting on its strengths and weaknesses.

II. DDOS ATTACKS

DDoS attacks can take several approaches. Five common techniques are used to implement DDoS attacks [7]. Smurf Attacks send an ICMP Echo Request to the victim’s network address with the victim’s address as the

source address. This causes all the computers on the network to reply with ICMP Echo Reply messages to the victim, thus overloading it. TCP SYN Attacks repeatedly send connection requests to the victim’s server using an unreachable network address as the source IP address. The victim then replies to the invalid user with an ACK and SYN according to the three way handshake mechanism of TCP and awaits an ACK from the unreachable host. This results in several pending connections that drain the server’s memory resources. UDP, TCP and ICMP Attacks flood the victim with packets continuously and at a high rate, requesting replies and thus causing congestion in the network. All of the above attacks use IP spoofing to conceal the identity of the attacker or direct traffic to a certain destination. In order to classify and evaluate the different defense mechanisms studied, it is important to first dissect the various attack methods. As can be observed from the mentioned DDoS attacks, they are mainly distinguished according to which system vulnerability they exploit: attacks taking advantage of some of the protocols’ particularities such as Smurf and TCP SYN attacks are called protocol attacks, and those targeting the victim directly such as UDP, TCP and ICMP attacks are called brute force attacks. This classification is useful to distinguish on which attack types the defense methods are most effective.

III. DDOS DEFENSE PRINCIPLES

Many challenges are involved in designing an effective DDoS defense mechanism, and thus it is recommended to follow certain principles in order to build a consistent solution.

Five principles have been proposed [2] to set the guidelines of a proper design. DDoS attacks are carried out through three main network elements: the source, the intermediate network, and the victim. At the victim’s end, the attack is simple to detect, due to the high volume of traffic generated, but rather complex to contain. On the other hand, it is difficult to detect an attack at the source, but the response gets more effective as heavy

traffic is contained before entering the internet core. Thus the first principle is to implement a distributed defense that collaborates between the victim and source ends. Second, it is primordial that a defense system conserves legitimate traffic while in action, thus preventing collateral damage. Third, a DDoS defense method should provide secure communication channels as well as authentication and control mechanisms between defense nodes. Fourth, it is beneficial to adopt a practical defense strategy consisting of autonomous components to be implemented partially and incrementally without disturbing the general network flow. Fifth, a defense system must take into account future compatibility issues such as interfacing with other systems and negotiating different defense policies.

Considering those five proposed design rules for a defense system and taking into account the various attack methods described in the previous section, we will base our classification on four criteria. The first is the site of action, which may include the victim, the victim's surrounding network, or the global intermediate network. The second is the ability to distinguish and preserve legitimate traffic, as some methods drop packets according to probabilistic measures, thus filtering out some legitimate traffic, while others radically drop all traffic at a certain point. The third is the ease of deployment and integration, where some methods can be implemented independently on a single point in the network while others may require wide collaboration between several nodes, making their integration a decisive factor. The fourth is the effectiveness of the defense mechanism on the different attack methods previously mentioned.

IV. DEFENSE METHODS

In what follows, we will overview several defense techniques and evaluate them according to the four proposed criteria.

A. Denying Denial-of-Service Attacks: A Router Based Solution

A general method based initially on more secure packet forwarding among routers is proposed [3] as a solution to prevent DDoS attacks. The routers are modified to provide encryption, digital signatures, and authentication, enabling the tracing of a packet back to its origin and thus stopping further traffic at the closest intelligent router point. Every group of collaborating routers is called a "hardened network". The hardened routers should be implemented at the border and access point of an Autonomous System. When arriving at the first hardened router, the packet's payload is encrypted together with one byte of its IP address and the last hardened router before the host will decrypt it. This way the packet can be traced back to the first hardened router, and an attack can be stopped at that point.

Even though this system provides more secure and private communication between the routers involved, a tremendous amount of complexity is introduced,

increasing cost, delay, and bandwidth parameters. In addition, knowledge of the last router is critical as it decrypts the initial packet, thus a single point of failure and consequently a less reliable information system is created.

B. Hop-Count Filtering: An Effective Defense against Spoofed DDoS Traffic

Hop-count filtering [4] is a victim based solution relying on the fact that the number of hops between source and destination is indirectly indicated by the TTL field in an IP packet. Linking the source IP with the statistical number of hops to reach the destination can be used as a reference to assess the authenticity of the claimed IP source.

In normal operation, the hop count is computed from the TTL value in the IP header and stored in a table with its respective IP address. When an attack is detected, a received IP packet is discarded if major discrepancies exist between its hop count and the value stored in the previously built table. This process depends heavily on assumptions and probabilistic methods, rendering the method inaccurate. Sophisticated hackers may also use IP addresses with relevant hop count, making this defense strategy altogether ineffective.

C. Implementing Pushback: Router-Based Defense Against DDoS Attacks

A network-based solution, Pushback [5], tries to solve the problem of DDoS attacks from within the network using the congestion level between different routers. When a link's congestion level reaches a certain threshold, the sending router starts dropping packets and tries to identify illegitimate traffic by counting the number of times packets are dropped for a certain destination IP address, since the attacker constantly changes the source IP address. The router then sends a pushback message to the routers connecting it to other congested links, asking them to limit the traffic arriving to this destination.

D. Protection against DDoS Attacks Based on Traffic Level Measurements

In another defense method based on measuring traffic levels, a DDoS module is attached to a given server making it a virtual server [6]. The module relies on a buffer through which all incoming traffic enters. Traffic level is continuously monitored and when it shoots to high levels, most incoming packets will be dropped. The module thus attempts to isolate the server from the attack. It first aims to detect the beginning of an attack at time t when the buffer becomes congested. Then the module, which can detect all active sources, attempts to identify the source of the attack by using statistical properties of traffic flow between t and $t+\delta$. Illegitimate traffic is recognized by its higher mean of traffic level and can thus be effectively suppressed.

E. StackPi: A Path Identification Mechanism against IP Spoofing and DDoS Attacks

Unlike other defense methods which use a probabilistic approach to detect illegitimate traffic, StackPi is a method that acts to mitigate illegitimate traffic by marking packets deterministically to detect IP address spoofing [8]. StackPi comprises two parts: Marking and Filtering. The former consists of concatenating the MD5 hash of the next node's IP address with the current node's IP address. The result is computed on each router and placed in the IP identification field of the IP header, with newer values replacing older ones when the field's 16 bits are entirely used. This gives a unique marking for each <source, destination> pair, which is stored in a table at the end-host. Meanwhile, the filtering scheme is responsible for detecting illegitimate traffic based on the marking scheme, where access is allowed if the marking matches the database entry and is denied otherwise.

F. Secure Overlay Services: An Architecture for Mitigating DDoS Attacks

Secure Overlay Services (SOS) [9] utilizes an architecture in which a site installs a filter in its immediate vicinity and selects a number of SOS nodes to be its "secret servlets". Secret servlets, which could be as few as two or three nodes, are the only nodes through which traffic can pass to the site. The attacker's task is complicated because of their secrecy to nodes outside the SOS. All other traffic whose source addresses are not approved, are aggressively blocked by the filter. A secret servlet node computes keys for well-known hash functions based on the site's IP address. Each of these keys will identify a number of SOS nodes known as "beacons" for that site, which verify the validity of the received information and forward traffic in a random manner inside the SOS, further toughening the attacker's task. For an outer source to communicate with the target site, it contacts a secure overlay access point (SOAP), which verifies that the source point has a legitimate communication for the target, meaning that the target has given prior permission to this user. SOAP routes packets to an appropriate beacon, which eventually forwards them to the secret servlets before they reach the target site. SOS therefore uses a combination of secure overlay tunneling, routing via consistent hashing, and filtering.

G. Differential Packet Filtering Against DDoS Flood Attacks

A cost-effective intrusion detection and response (IDR) solution to the flooding problem is differential packet filtering [1]. In normal traffic conditions each incoming packet is considered safe, normal, or risky according to a continuously updated history table of trusted IP sources. When an attack is detected, some risky IP packets are dropped while some safe and normal IP addresses are downgraded. This method relies on probabilistic means to determine risky packets, which results in the possible dropping of some legitimate traffic, but it is adaptive to traffic changes and attempts to sustain quality of service.

V. CLASSIFICATION

The region of action plays an important role in the selection of the method to be used. The reviewed methods can be implemented on a single node or on multiple collaborating nodes that can be part of the victim, the victim's network, or the intermediate network. Table 1 shows the defense strategies according to this last parameter.

Table 1. Region of action.

	Single node	Multiple collaborating nodes
Victim's end	Hop-count filtering Traffic level meas. Diff. packet filtering	StackPi
Victim's network		Pushback StackPi SOS
Intermediate network		Hardened network StackPi SOS

By referring to Table 1, we observe that none of the methods discussed involves a single node on the intermediate or the victim's network. We also note that as a method gets more complex or involves more nodes, its integration time on the Internet increases. Even a very efficient method might not be adopted if it requires large amounts of resources and numerous network modifications (Figure 1).

We remark that a certain relationship exists between a system's complexity and efficiency, which is logical since a more complex system is expected to be more powerful.

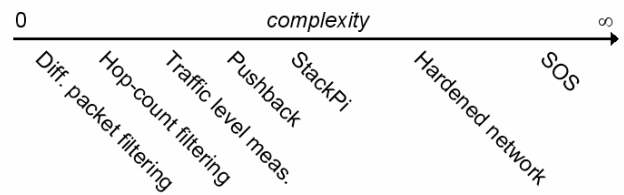


Figure 1. Relative method complexity.

In addition, defense methods involve dropping some legitimate traffic. This acts a good indicator of how well a method performs. Figure 2 gives a relative appreciation of the amount of legitimate traffic dropped.

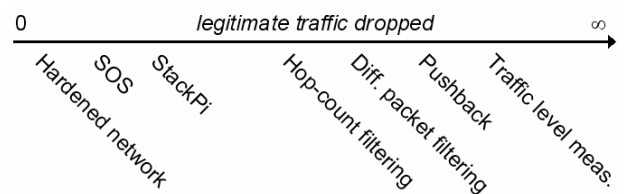


Figure 2. Legitimate traffic dropping.

As for their effectiveness on the attacks, the defense methods act either on the authenticity of the IP datagram, thus preventing IP spoofing, or on the reduction of the traffic volume. Since the five presented attack methods use spoofed IP addresses even in the case of a protocol attack, all five are mitigated by the presented defense methods.

VI. OBSERVATIONS

DDoS defense methods have been proposed for some time, but most of them remain theoretical with no actual implementation. From the study carried out in the previous sections, we can point out several observations.

Many of the methods need to be implemented simultaneously and collaboratively on several nodes, making them difficult to implement, especially on nodes that need to maintain round-the-clock Internet connectivity. Moreover, an adopted solution introduces relative complexity, overhead, and delay to the network, by adding features such as digital signatures and encryption.

The defense methods rely on random or probabilistic means to detect illegitimate traffic and discard it, which necessitates that a certain percentage of legitimate packets be dropped in the process. While this might not be a serious downside, it reduces the overall Quality of Service.

In addition, not all companies would be willing to implement solutions developed by competitors or to share their Internet resources for contributing in the implementation of a global standardized solution.

With these observations and concerns in mind, implementing an effective defense method becomes a critical investment that requires serious consideration to reach a balance between benefits and costs: the location, simplicity, performance, and cost of a defense system are correlated and an efficient system is one which optimizes these factors.

VII. PROPOSED DEFENSE METHOD: PACKET FUNNELING

After describing and evaluating the defense mechanisms, we propose a method that aims to mitigate the effects of a DDoS attack. We will first discuss our motivations for coming up with this method and then discuss its features, describe its modules, and attempt to predict its behavior.

A. Motivation

We are not seeking a method that would totally prevent a DDoS attack but more of a method that would mitigate an attack's effects on the victim's side to a point where it would no longer serve its purpose: denial of service.

Based on our previous conclusions, the method should be simple to integrate and should not drop any legitimate packets. As a result, we propose packet

funneling as a load balancing solution that would delay heavy traffic on the server and the server's link to avoid any denial of service, whether legitimate or malicious. It is vital to note that modern DDoS attacks employ a large number of zombies and thus attacking hosts to congest a certain link. They rely heavily on spoofed IP generation to mask the source of the attack. The result is a stream of IP packets with different IP addresses. As opposed to this, the IP pattern of a normal user will have repetitive occurrences.

Additionally, it is very difficult to differentiate legitimate traffic from DDoS traffic, and consequently to determine which packets should be preserved and which can be dropped, especially that there are cases of "legitimate" denial of service where the server cannot cope with the amount of incoming traffic. Based on these facts, our aim is to effectively handle DDoS attacks without dropping any legitimate traffic, exploiting the fact that the attacker uses continuously spoofed IP addresses. Moreover, our method should be easy to implement and should dissuade attackers from initiating further DDoS attacks.

B. Features

Based on the preceding motivations, we propose a solution that would limit the number of active IP addresses flowing on an access link in order to limit congestion on this link (Figure 3). Packets with IP addresses recurring at small intervals are favored over IP addresses with flash occurrences by delaying the passage of the latter packets until bandwidth is available on the link.

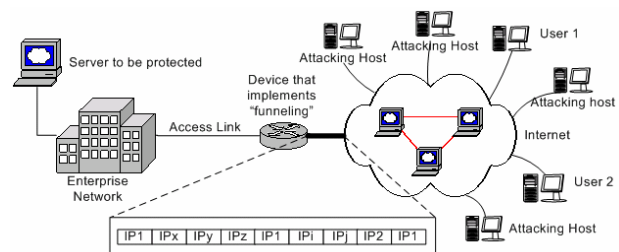


Figure 3. Network diagram.

We define the user as a distinct IP address and the link capacity as the average number of users the access link can handle. This capacity takes into account the power of the server, the bandwidth of the link, and the average bandwidth consumed per user. This variable is particular to every server and is set up at the administrative level. We also define a timeout value, which is the time for which a single occurrence of an IP address is considered a flash appearance and not a regularly connected user.

C. Description

Our proposed packet funneling method consists of a single device that connects the global network to the designated access link (Figure 3). An Active IP (AIP) table is used to specify the active users (IP addresses) and their relative timeout values. In normal cases, when the

packet of a new user is received, the user is entered in the AIP table, its timeout value is set, and the packet is forwarded to its destination.

The size of the AIP table is a parameter set by the administrator according to the average number of expected users. When there are a huge number of users, as in the case of a DDoS attack, the AIP table becomes full. In this case, newly arriving packets of active users continue to be forwarded, whereas new users need to be delayed and are transferred to a Waiting Matrix instead.

The Waiting Matrix stores the arriving packets of each delayed user until one of the active users times out and is thus removed from the AIP table. It should have a dynamic structure with its size limited to the available storage space. Only when the memory is entirely consumed will the packets be dropped instead of delayed. Figure 4 shows the module that is triggered whenever a new packet is received.

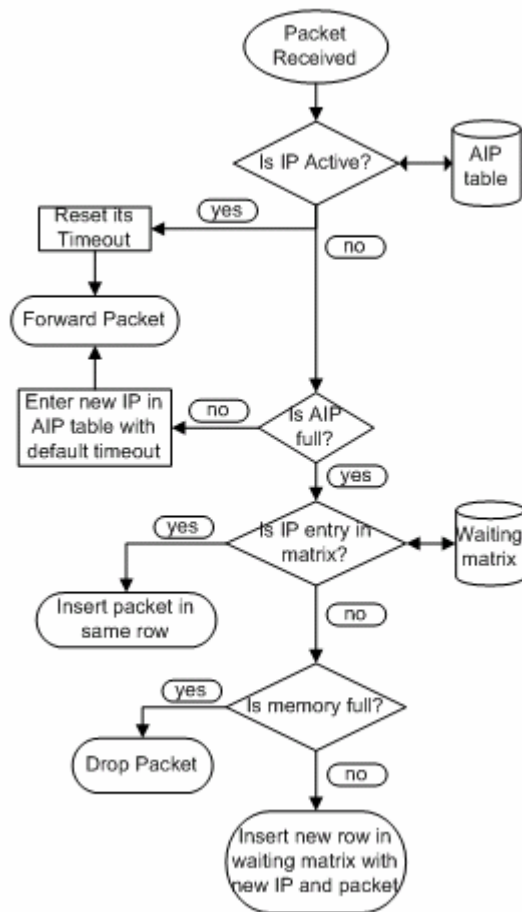


Figure 4. Module for packet reception.

When an active user is no longer sending packets, its timeout value will soon expire and the user will be dropped from the AIP table, as shown in Figure 5. If there are delayed users in the AIP table, one of those users can now become active. There are several factors that can be used to determine which user to activate, such as:

- First in, first out: the user with the longest delay is activated and its delayed packets forwarded.
- The user with the largest number or size of waiting packets is activated.
- The user with highest priority is activated, where priority is determined according to historical data, by analyzing server logs for frequency and levels of user activity.
- Trusted users are favored over unknown users or users with known malicious intent. This information can be stored in a separate table.

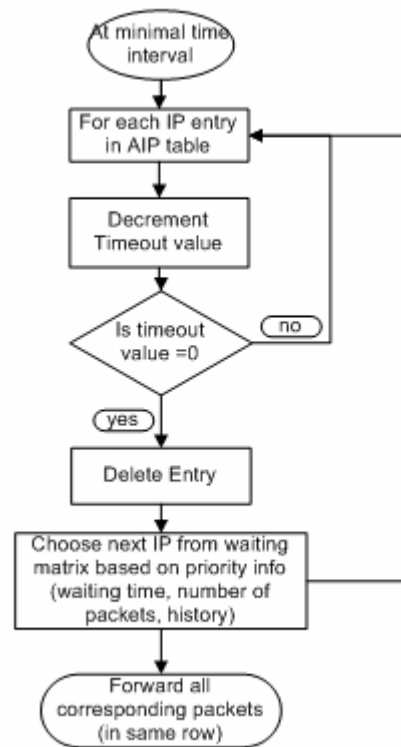


Figure 5. AIP table maintenance module.

D. Expected Behavior

The effectiveness of packet funneling in coping with DDoS attacks is largely dependent on the involved parameters. The parameters should be carefully chosen and balanced so as to achieve a valid solution with minimal side effects.

To begin with, the size of the AIP table should be small enough to be able to service all active users without experiencing link congestion, taking into account the case where each single user is sending a large number of packets. On the other hand, the table size should be large enough to allow as many users as possible to be connected without being delayed.

The timeout value of AIP entries is also critical. It should guard against the multiple flash occurrences of DDoS traffic without keeping the entry idle for a long time while other users are being delayed. It also faces additional challenges, as we need to keep in mind that delaying packets could cause the sender to retransmit them, thus unnecessarily increasing the traffic load.

Therefore, the AIP timeout value and the selection of the next delayed user to be activated should be balanced, for example, with the retransmission timeout of TCP.

These and other implementation details are open to further scrutiny and testing, where several possible improvements may be suggested, such as the method to choose the next IP to be retrieved from the Waiting Matrix which may employ some historical data to favor an IP over another.

With these concerns in mind, packet funneling as a method of traffic management on the last link is expected to be effective against distinct IP addresses arriving in large quantities and heavy traffic problems in general, making it possible to implement at servers with high peaks at some times of the day. A main feature of packet funneling is easy integration, giving it a big advantage over other methods that need modifications at different locations throughout the network. However, if implemented on a slow device, packet funneling may cause extra delay even at normal times, which reduces Quality of Service and thus discourages legitimate users from connecting to the server. As a result, it is advised to implement this method on dedicated hardware like a programmable network processor.

VIII. CONCLUSION

By presenting different types of DDoS defense mechanisms, we are able to evaluate them based on the four criteria we proposed. While the methods differ in their region of action, the amount of legitimate traffic they drop, their ease of implementation, and the type of attack they are effective against, each method has certain features that make it more suitable to implement in one situation than another. The proposed packet funneling approach promises to be a suitable means of coping with DDoS traffic, with easy integration at minimal cost.

REFERENCES

- [1] Tanachaiwiwat, S. and Hwang, K. "Differential packet filtering against DDoS flood attacks." ACM Conference on Computer and Communications Security (CCS). Washington, DC, October 2003.
- [2] M. Robinson, J. Mirkovic, M. Schnaider, S. Michel, and P. Reiher. "Challenges and principles of DDoS defense." SIGCOMM 2003.
- [3] Zhang, S. and Dasgupta, P. "Denying denial-of-service attacks: a router based solution." International Conference on Internet Computing, June 2003.
- [4] Jin, G., Wang, H., and Shin, K. G. "Hop-count filtering: an effective defense against spoofed DDoS traffic". In *Proceedings of the 10th ACM conference on Computer and communication security*. Washington D.C., USA, 2003.
- [5] Ioannidis, J. and Bellovin, S. M. "Implementing pushback: Router-based defense against DDoS attacks". NDSS Conference Proceedings, 2002.
- [6] Bencsath, B. and Vajda, I. "Protection against DDoS attacks based on traffic level measurements." Western Simulation MultiConference. San Diego, California, USA, January 2004.
- [7] Noureldien, N. "Protecting web servers from DoS/DDoS flooding attacks: a technical overview." International Conference on Web-Management for International Organisations. Geneva, October 2002.
- [8] Perrig A., Song D., Yaar A. "StackPi: a new defense mechanism against IP spoofing and DDoS attacks." CMU technical report. December 2002. Updated February 2003.
- [9] Keromytis, A.D., Misra, V., and Rubenstein, D. "SOS: an architecture for mitigating DDoS attacks". Selected Areas in Communications, IEEE Journal volume: 22, Issue: 1, January 2004.