INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

RECTIFIED PROBABILISTIC PACKET MARKING BASED ON TRACE BACK MECHANISM AGAINST MULTIPLE ATTACKER'S

ANIL V TURUKMANE[1], DR. S. K. YADAV[2]

1. Research Scholar, JJTU, Rajasthan, India.
2. , JJTU, Rajasthan, India.

Accepted Date: 15/03/2016; Published Date: 01/05/2016

Abstract: A Traceback-based Defense against DDoS Flooding Attacks (TDFA) approach to counter this problem. TDFA consists of three main components: Detection, Traceback, and Traffic Control. In this approach, the goal is to place the packet filtering as close to the attack source as possible. In doing so, the traffic control component at the victim side aims to set up a limit on the packet forwarding rate to the victim. This mechanism effectively reduces the rate of forwarding the attack packets and therefore improves the throughput of the legitimate traffic. Our results based on real world data sets show that TDFA is effective to reduce the attack traffic and to defend the quality of service for the legitimate traffic.

Keywords: Packet Filtering; Traffic Control; IP Trace back; DDoS Attack, Deterministic Flow Marking.

Corresponding Author: MR. ANIL V TURUKMANE

Access Online On:

www.ijpret.com

How to Cite This Article:

PAPER-QR CODE

Anil V Turukmane, IJPRET, 2016; Volume 4 (9): 14-22

INTRODUCTION

Two features of the DDoS attacks are challenging for the security (defense) systems. Firstly, the DDoS packets tend to appear as legitimate packets, so filtering them with no impact on the legitimate traffic is challenging. Secondly, as the source IP address can be spoofed easily, finding the source of DDoS attacks is very challenging. To find the source of such an attack, several IP traceback mechanisms are proposed. These approaches usually require that some routers along the attack path embed marking information to the packets to recognize the identity of the routers. After collecting an appropriate number of marked packets, the victim is able to find the attack path or the source of the attack. Although the IP traceback approaches let the victim to guess the source of an attack, in general they do not have the ability to decrease the impact of the attack while the attack is in progress.

In this work, our goal is to explore how far we can push a Defense/security system to mitigate the impact of such an attack while the attack is in progress. Our objectives are: (i) to minimize the burden of filtering on the participating routers; (ii) to minimize the number of modifications required on the current protocols in use; and (iii) to maximize the survival rate for the legitimate traffic under an intensive attack. To achieve such a solution, we propose the TDFA approach. The proposed system defends against DDoS attacks by coordinating between the defense systems at the source and the victim ends. This necessitates communication between the victim and the filtering routers so that the filtering can be located as far away

as possible from the victim. TDFA consists of three main components: Detection, Traceback, and Traffic Control. The detection component aims to detect unusual changes of the incoming traffic to identify DDoS attacks. The traceback component employs the deterministic flow marking (DFM) technique [1-3] to find the source of the attack. The traffic control component, which is the focus of this work, aims to minimize the attack traffic. Once a DDoS attack is detected by the detection component, the traceback component finds the source of the attack using the DFM technique. Then the traffic control component sends traffic control messages to the edge router of the attack network. When the edge router of the attack network receives the control messages, it will be triggered to adjust the packet forwarding rate to the victim. TDFA filters attack traffic at the source end to eliminate the consumption of the computing resources and the bandwidth of the victim. This in return improves the performance of the system for the legitimate services and users.

2. RELATED WORK

IP SPOOFING AND DDOS ATTACKS: Internet security is of critical importance to our society, as the government and economy increasingly rely on the Internet to conduct their business, and people use the Internet as a convenient vehicle for simplifying a wide range of tasks, from banking to shopping. Unfortunately, the current Internet infrastructure is vulnerable to a Distributed Denial of Service (DDoS) attack. Because DDoS attacks typically rely on compromising a large number of hosts to generate traffic to a single destination, the severity of DDoS attacks will likely increase as greater numbers of poorly secured hosts are connected to high bandwidth Internet connections. For example, an attacker who could compromise the popular SE distributed computation software, or any popular P2P client, would be able to harness several hundreds of thousands of hosts to generate traffic for an attack. The weakness of the current Internet infrastructure that facilitates DDoS attacks is the inability for a packet recipient to authenticate that packet's claimed source IP address. In other words, an attacker can intentionally modify, or *spoof*, the source address of the packets it sends from a compromised host. Two examples of DDoS attacks that rely on IP address spoofing are:

TCP SYN Flooding: In this attack, an attacker sends TCP SYN packets as if to initiate a TCP connection with its victim. These SYN packets contain spoofed source IP addresses, which cause the victim to waste resources that are allocated to half-open CP connections which will never be completed by the attacker.

Reflector Attack: In this attack described by Paxson, the attacker attempts to overwhelm the victim with traffic, by using intermediate servers to amplify the attacker's bandwidth and/or hide the attacker's origin. The attacker simply sends requests to the intermediate server with a spoofed source IP address matching the victim's IP address. The intermediate server only sees that a number of requests are supposedly coming from the victim, and so sends its responses to the victim. When properly coordinated, a group of attackers can cause a flood of packets to hit the victim, without sending any packets directly to the victim itself. To amplify the traffic, the attacker selects intermediate servers whose responses to the spoofed requests are larger than the requests themselves. For example, in DNS server based reflector attacks, attackers send short DNS lookup requests (50 bytes each), whose replies can be over a thousand bytes long, thus giving the attacker a 20-fold traffic amplification. Other popular reflectors are Internet game servers, where attackers can use similar methods to gain two orders of magnitude of traffic amplification.

THE PROPOSED DEFENSE SYSTEM

When a DDoS attack occurs, most of the traffic drops by the upstream routers, even before reaching to the victim. In this case, when react to the attack at the victim-end. To mitigate the attack, we propose a system model to set up the second defense line at the upstream routers to react to the attack, This system relies on the IP trace back approach based on the DFM [1-3] algorithm. DFM enables the victim to differentiate the traffic coming from different sources. DFM has the ability to infer the source of an attack. What makes DFM more appealing is that unlike several other IP trace back methods, DFM only requires the cooperation of the edge router. The proposed system based on DFM improves the throughput of the legitimate traffic during the attack by filtering the attack traffic at the source end. This approach can be viewed as a distributed algorithm that consists of the following components.

DDoS Detection component: This component is used at the victim-end edge network for recognizing anomalous changes on the network traffic. There are several algorithms and tools for detecting DDoS attacks that can be used for this component such as Snort. Our intention in this paper is not to propose a new DDoS detection algorithm.

IP Traceback component: The IP traceback component employs the DFM traceback technique to identify the source of a DDoS attack. This traceback mechanism consists of a light weight flow marking module running on the edge routers (*DFME*) and a mark decoding module running on the victim-end (*DFMD*) to infer the source of traffic based on the information extracted from the marked packets. A detailed description of the DFM technique could be found in our previous work [1-3].

Traffic Control component: This component consists of two modules: Traffic Adjustment (*TA*) and Packet Filtering (*PF*) modules. The *TA* module runs on the victim or the border gateway device (e.g. firewall) of the victim network. After finding the source of the attack, using IP traceback component, the *TA* module sends a request message, which carries attack traffic information, to the defense system at the source-end edge network. On the other hand, the *PF* module runs on every edge router and filters packets that are directed to the victim based on the instructions issued from the *TA* module.

Traffic Adjustment (TA) and Packet Filtering Modules:

To control the attack traffic, we propose a rate limit algorithm that relies on some concepts of TCP flow and congestion control lgorithms to allocate the bandwidth to all the incoming traffic from the routers that forward the attack traffic. It is not fair to punish all these routers by setting the same rate limit on them. Therefore the traffic histories of individual routers should be considered by the *TA* module before sending a rate limit request to a *PF* module. Our proposed defense algorithm can be divided into the following phases:

Connection Establishment: Connections must be properly established in a 3-way handshake process. The *TA* module at the victim end opens a full duplex session with the *PF* module at the attacker edge router end for subsequent communications.

Congestion Avoidance: The *PF* module performs the congestive avoidance phase once it gets a message indicating that the estination of the outgoing packets is under a DDoS flooding attack. In this case, the forwarding rate of the packets directed to the victim is decreased exponentially in such a manner that the forwarding rate is halved at each round trip time (i.e. after *X* round trip times, the forwarding rate will be decreased by *2X* times). This operation continues until the *PF* module gets a message from the victim indicating that the receiving packet rate is tolerable.

Slow Start: The *PF* module performs this phase once it is informed that the victim is able to tolerate the current packet forwarding rate. In this case, the packet forwarding rate to the victim is increased linearly. After each increase in the rate, the *PF* module sends an update message to the *TA* module indicating that the rate has been changed, and then waits for the confirmation of the new rate from the *TA* module. This way, the *PF* module infers if the victim is able to tolerate the new rate or not. If the new rate is not confirmed, then the algorithm gets back to the congestion avoidance phase; otherwise, it waits for a given period of time and then increases the forwarding rate again. This process continues until the packet drop rate, calculated by the *PF* module, is equal to zero.

Connection Termination: If the drop rate is zero, then the *PF* module removes any rate limit on the forwarding packets destined to the victim and sends an update message to the *TA* module. If the *TA* module confirms the new rate limit, the algorithm enters the connection termination phase to close the established virtual circuit. This process can be done by a 3-way FIN/ACK exchanges between the *PF* and the *TA* modules, which is started by the *PF* module. If the *TA* module does not confirm it, the algorithm gets back to the congestion avoidance phase.

Note that once the *PF* module changes the packet forwarding rate (it may happen in any of the phases, namely congestion avoidance, slow start, and connection termination phases), it sends an update message to the *TA* module and waits for a response. If the *TA* module does not confirm the new rate, then the *PF* module performs the congestion avoidance phase again. Not confirming to the new rate limit is determined by the *PF* module in 3 ways:

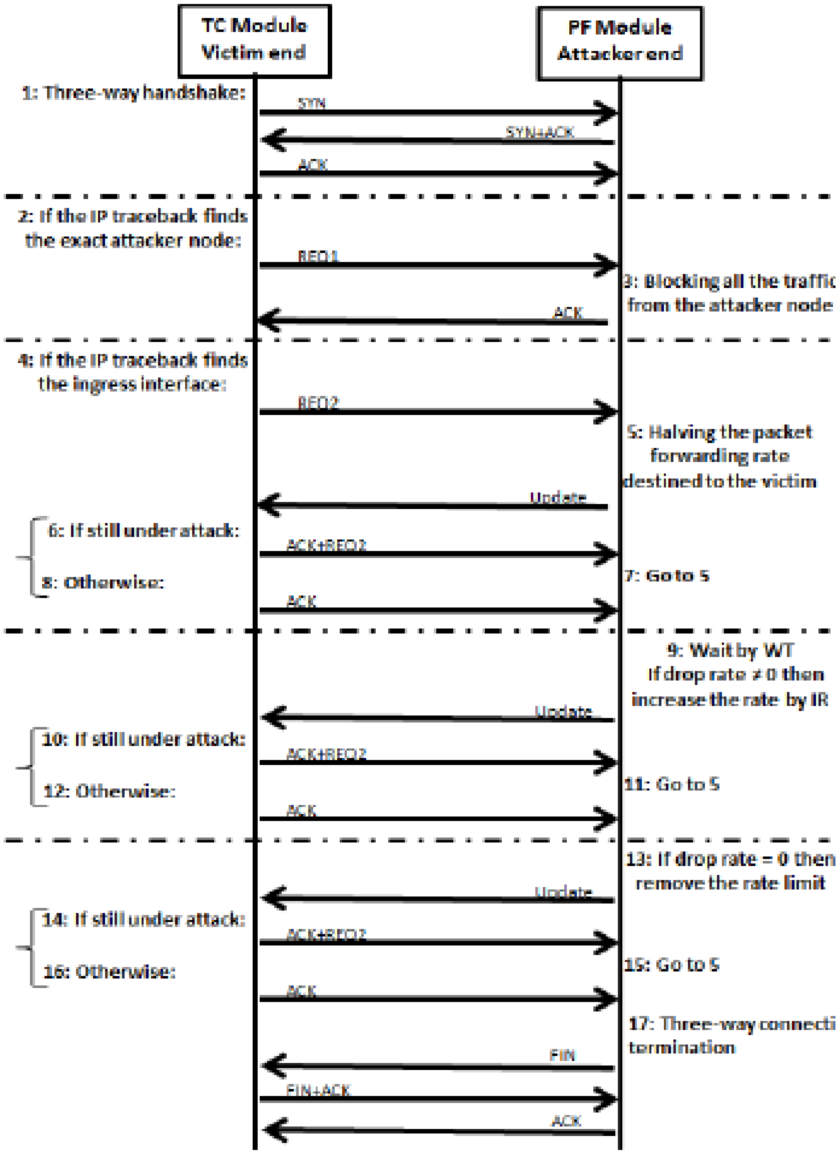
1. Getting an ACK+REQ2 message from the *TA* module: If the *TA* module gets an update message from the PC module while the victim is still under the flooding attack, it sends an ACK+REQ2 message to the PC.

2. Duplicated acknowledgment or request messages: It means that the victim is under a heavy flooding attack, so the update message has not reached to the *TA* module due to the traffic congestion.

3. Lack of acknowledgment before time out: It indicates that because of heavy traffic, the bandwidth is full at the victim-end and the *TA* messages cannot reach to the PC module.

The Proposed System Design

To transmit the control messages between the *TA* and *P*F modules, we propose to use Internet Control Message Protocol (ICMP). ICMP messages are used by network devices, like routers, to send error messages indicating, for example, that a requested service is not available or that a host or a router could not be reached. ICMP can also be used to relay query messages. Fig. 2 presents the proposed ICMP message that is very similar with the ICMP Timestamp (Type 13) and the ICMP address Mask Request (Type 17) messages. For all the ICMP packets, the first 4 bytes of the header will be consistent. The first byte is for the ICMP type. The second byte is for the ICMP code, and the third and fourth bytes are a checksum of the entire ICMP message. The first byte can be used to address 255 different ICMP Type values, but so far only 28 of them are used and the rest are free to employ for other purposes. Types 1 and 2 are both free, so we have employed type 1 for sending messages from the *TA* module to the *PF* module and type 2 for sending messages from the *PF* module to the *TA* module. The next 39 bits consists of 16 bits sequence number, 16 bits acknowledgment number, and 5 control flag bits including SYN, FIN, ACK, REQ1 and REQ2. If the SYN flag is set to (1), then this is the initial sequence number. The sequence number of the actual first data byte and the acknowledgement number in the corresponding ACK are equal to this sequence number plus 1. If the SYN flag is clear (0), then this is the accumulated sequence number of the first data byte of this segment for the current session. If the ACK flag is set then the value of this field is the next sequence number that the receiver is expecting. This acknowledges receipt of all prior bytes (if any). The first ACK sent by each end acknowledges the other end's initial sequence number itself. The next two fields are the Interface-ID and Node-ID of the attacker that are identified by the IP trace back component. Using these 2 fields, the *TA* module sends the attacker identity to the *PF* module.


TC Module
Victim end
PF Module
Attacker end
1: Three-way handshake:
SYN
SYN+ACK
ACK
2: If the IP traceback finds the exact attacker node:
REQ1
3: Blocking all the traffic from the attacker node
ACK
4: If the IP traceback finds the ingress interface:
REQ2
5: Halving the packet forwarding rate destined to the victim
Update
6: If still under attack:
ACK+REQ2
7: Go to 5
8: Otherwise:
ACK
9: Wait by WT If drop rate ≠ 0 then increase the rate by IR
Update
10: If still under attack:
ACK+REQ2
11: Go to 5
12: Otherwise:
ACK
13: If drop rate = 0 then remove the rate limit
Update
14: If still under attack:
ACK+REQ2
15: Go to 5
16: Otherwise:
ACK
17: Three-way connecti termination
FIN
FIN+ACK
ACK

The proposed defense approach is shown in Fig. All the messages exchanged between the *TA* and the *PF* modules and their flags in the ICMP header. To establish a connection between the *TA* module at the victim end, and the *PF* module at the attacker end edge router, the three-way handshake should be set. Establishing a connection is initiated by the *TA* module at the victim-end by sending a SYN message to the *PF* module and the attacker-end edge router. The *TA* module sets the segment's sequence number to a random value *A*. In response, the *PF* module replies with a SYN-ACK message. The acknowledgment number is then set to the received sequence number plus one, *A*+1, and the sequence number that the server chooses for the packet is another random number, *B*. Finally, the *TA* module sends an ACK back to the *PF* module. The sequence number is set to the value of the received acknowledgement, i.e. *A*+1, and the acknowledgement number is set to *B*+1. At this point, both the *TA* and the *PF* modules have received an acknowledgment of the connection. As described in [1, 2], in the cases where the IP traceback component is able to find the exact node (Line 3 - Fig.), the *TA* module sends REQ1 message to ask to the *PF* module to block all the traffic corresponding to the attacker. This can be done by providing the Node-ID and the NI-ID of the attacker node in the REQ1 message. Then, the *PF* module blocks all packets coming from that specific node and sends an ACK message to the victim. As described before, in some cases the IP traceback component can find the ingress interface of the edge router on the attacker side, but not the attacker node behind that edge router. In these cases the proposed algorithm performs the congestion avoidance phase to mitigate the effect of DDoS attack. Lines 4-8 (Fig.) demonstrate the congestion avoidance phase to exponentially decrease the attack forwarding rate. Once the *PF* module receives a REQ2 message (Line 4 - Fig.), it halves the forwarding rate (Line 5 in fig) and sends an update message to the victim. Then, the *TA* module decides if the current rate is still above the load limit (Line 6 - Fig), or not (Line 8 - Fig.). Lines 13-17 (Fig.) present the connection termination phase. This occurs when either the flooding attack has ended or the victim is able to tolerate the current packet forwarding rate. The connection termination is initiated by the *PF* module (Line 17 - Fig.), by sending a FIN message to the *TA* module. In return, the *TA* module replies with a FIN+ACK message and then the *PF* module replies with an ACK message.

CONCLUSION

In this work, have implemented and evaluated the proposed system, TDFA, against the real-world DDoS and normal traffic traces. Our results show that TDFA efficiently drops the attack packets at the source end of the attack, while imposing a low overhead on the edge routers participating in the defense. Moreover, the location of defense modules are fully in line with the location of DFM IP traceback modules, making the scheme more deployable

REFERENCES

1. A. Yaar, A. Perrig and D. Song “SIFF: a stateless Internet flow filter to mitigate DDoS flooding attacks”, IEEE Symp. Security and Privacy, Berkeley, pp. 130–143, May 2004.
2. Y. Chen, S. Das, P. Dhar, A. El Saddik and A. Nayak, “Detecting and preventing IP-spoofed distributed DoS attacks” International Journal of Network Security, vol. 7, no. 1, pp. 69–80, July 2008.

3. M.S. Fallah and N. Kahani, “TDPF: a traceback-based distributed packet filter to mitigate spoofed DDoS attacks”, Security and communication networks, Wiley Online Library, DOI: 10.1002/sec.725, Feb. 2013.

4. R. Mahajan, S. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker, “Controlling high bandwidth aggregates in the network”, Technical report, ACIRI and AT&T Labs Research, Feb. 2001.

5. Snort network intrusion prevention and detection system, Accessed 28 October 2013,

6. Internet Control Message Protocol, Accessed 28 October 2013, http://en.wikipedia.org/wiki/Internet_Control_Message_Protocol/

7. The CAIDA "DDoS Attack 2007" Data set, Accessed 28 October 2013,

8. The CAIDA Anonymized Internet Traces 2012 Data set, Accessed 28 October 2013, http://www.caida.org/data/passive/passive_2012_dataset.xml/