A Robust Mechanism to Mitigate DDOs
Attack Using Entropy Variation

REDDYBATHINI DURGA SIVA PRASAD
M.Tech Student, Dept of CSE
Vasireddy Venkatadri Institute of Technology
Nambur, Guntur (Dt), A.P, India

P.R.KRISHNA PRASAD
Assistant Professor, Dept of CSE
Vasireddy Venkatadri Institute of Technology
Nambur, Guntur (Dt), A.P, India

ABSTRACT: In the scenario of attack of Distributed Denial-of-Service, the flows by means of destination as the victim consist of legitimate flows and a grouping of flows of attack and legitimate flows. To commence an attack of Distributed Denial-of-Service, the attacker initially set up a network of computers that are used to produce the enormous traffic amounts that are essential to reject services to the legitimate users of the victim. The volumes of various flows augment considerably in an extremely small time period in the attack of Distributed Denial-of-Service when compared with the cases of non attack. The use of flow entropy variation was introduced in this paper. Once an attack of Distributed Denial-of-Service has been recognized, the victim commences the succeeding process of pushback to spot the location of zombies. Additionally this process is repetitive in a fashion of parallel and distributed mode until it reaches the source of attack otherwise the limit of discrimination connecting the flows of attack and lawful flows is fulfilled.

Keywords: Distributed Denial-of-Service, Legitimate flow, Flow entropy variation, Push back, Attack Mitigation, NAT, Network Security.

I. INTRODUCTION

Due to the susceptibility of the inventive design of the Internet, we may possibly not be competent to discover the actual hackers at the present time. The schemes of IP trace back are measured flourishing if they can possibly recognize the zombies from which the attack packets of Distributed Denial-of-Service go through the Internet [4]. It is an astonishing challenge to outline back the basis of the attacks of probabilistic packet marking and the in which the attackers produce a vast quantity of requests on the way to victims’ right the way through the computers of compromised through the endeavour of normal service denying [8]. A numeral of approaches of IP trace back have been recommended to make out attackers and there are two most important methods such as the technique of deterministic packet marking and probabilistic packet marking method. The attacks of Distributed Denial-of-Service are under attack at shattering the resources of victim, such as computing power, network bandwidth and the data structures of operating system [1]. To initiate an attack of Distributed Denial-of-Service, the attacker at the start set up a network of computers which construct the massive traffic quantity that are crucial to decline the services. In the attack of Distributed Denial-of-Service, the flows by means of destination as the victim consist of legitimate and a grouping of flows of attack and legitimate flows [12]. To generate this network of attack, attackers find out susceptible hosts on the network. The probabilistic packet marking method can possibly function in a restricted range of the Internet, where the defender has the influence to administer and this type of ISP networks is usually relatively small, and not possible to trace back to the sources of attack situated out of the ISP network [3]. The strategy deterministic packet marking necessitates the entire Internet routers to be modernized for the marking of packet and this mechanism creates an astonishing challenge on storage intended for packet logging for the routers. As a result, it is infeasible in tradition at the present time.

Deterministic packet marking and probabilistic packet marking methods are susceptible to hacking which is termed as packet pollution [7]. The usage of flow entropy variation was introduced and once an attack of Distributed Denial-of-Service has been recognized, the victim commences the succeeding process of push back to spot the location of zombies. We classify packets that are passing all the way through a router into flows, which are definite by the router of upstream where a packet arrives from, and the address of the destination of the packet [13].

II. LITERATURE SURVEY

Over the past few years, the size and frequency of DDoS attacks have grown dramatically as attackers take advantage of botnets and other high-speed
Internet access technologies to overwhelm their target’s network infrastructure.

Servers can directly identify the DDoS attack occurrence situation, but actually, there are not many things that server can do. It is because the main goal of a server is offering a service. There are not many security functionalities for server itself. If lots of network traffic is getting into a server, then the server would be failed. Therefore, in host level layer, DDoS defense techniques have to be optimized minimized for the server. These days, however, server based DDoS attack defense technology is currently being researched. For example SecureNIC project in ETRI. Secure-NIC is a NIC (Network Interface Card) which contain security functionalities such as SYN Flooding prevention, HTTP GET Flooding prevention, and so on. In Secure-NIC, it has its own CPU, so, it minimizes the consumption of main CPU resources.

The purpose of this paper is to overcome the some of the limitations in the existing system work as mention below:

The detection and response techniques should be adaptable to a wide range of network environments, preferably without significant manual tuning.

1. Attack detection should be as accurate as possible. False positives can lead to inappropriate responses that cause denial of service to legitimate users. False negatives result in attacks going unnoticed.

2. Attack response should employ intelligent packet discard mechanisms to reduce the downstream impact of the flood while preserving and routing the non-attack packets.

3. The detection method should be effective against a variety of attack tools available today and also robust against future attempts by attackers to evade detection.

### III. PROPOSED APPROACH

In SYN floods, attacker would send an instant barrage of SYN packets from IP addresses (often spoofed) that will not generate replies to the SYN/ACKs. To remain effective, attacker needs to send new barrages of bogus connection requests frequently. Almost all the SYN flooding packets will not be retransmitted. Then again, Should a legitimate client’s SYN packet is lost, it'd retransmit the SYN packet many times before dropping. Our mitigation scheme utilizes the characteristic of SYN floods and client’s persistence.

#### Pseudo Code

For each packet arrival do
Check the IP Header=20 then
Choose the protocol=TCP or UDP or Http
Check the payload packet
If payload normal
Goto destination
Else distinguish the packet for analysis
IF TCP SYN flood then
Report alert mail to admin
Else analysis for other threats
End if
End if
Other protocol
Goto destination
Endif
End for

### Packet Filtering

So, as to do the packet filtering, many factors will be considered. There are three main factors in this paper:

1. The traffic filtered each packet to each protocol such as TCP, UDP and ICMP. Since TCP packet is the main protocol in this research, other protocol is distinguished.

2. TCP flags SYN, ACK, RST, FIN, PSH, and URG are divided to each group to check the three-way handshake is complete or not. Usually, when there are a lot of SYN packets or RST packets, the analysis is focusing to these packets and assumed there is an attack.

3. IP address is valid and not a spoofed address. Since this experiment is done in local network, unrecognized IP address is considered an attack. These factors are important to detect method to recognize which packets are the infected packets.

#### Algorithm to Packet capture and filtering:

Step 1: open the interface
Step 2: Start capturing packets
For each packet pack
a) set filter='TCP or IP'
b) temp[]= capturesetfilter(filter)
c) if(temp[]=TCP')
d) store pack dest port, seq, src port, syn to DB
else
  e) store identifier(v4.0), dest port, src port, sync to DB.
step 3: Sort DB according to sequence number in the TCP table.
Step 4: Sort the DB according to IP addresses.
Step 5: End

#### Algorithm to capture n/w packets

Step 1: Get list of all network interfaces and store them in NetworkInterface[]
Step 2: Get each Network Interface name and its MAC addresses in the NetworkInterface[]
Step 3: Choose NetworkInterface to capture packets in promiscuous mode.
(In non-promiscuous mode, when a NIC receives a frame, it normally drops it unless the frame is
addressed to that NIC's MAC address or is a broadcast or multicast frame, thus in Promiscuous mode allowing the computer to read frames intended for other machines or network devices.

Step 4: Set no. of Packets to capture. (Infinite -1)
Step 5: Print the packets in the console.
Step 6: End

**Chi-Square Statistic**

Pearson’s chi-square: Test is used for distribution comparison in cases where the measurements involved are discrete values. For example, it could be used to test the distribution of TCP SYN flag values (0 or 1) or protocol numbers. The test works best when the number of possible values is small. In particular, a rule of thumb is that the expected number of packets in a sample having each possible value be at least five. However, this can often be achieved through “binning”, that is combining a set or range of possible values and treating them as one. For example, the chi-square test can be applied to service ports by considering four values: HTTP, FTP, DNS, and “other.” Similarly, packet lengths can be binned into ranges such as 0-64 bytes, 65-128 bytes, 129-255 bytes, etc. For a sample of \( N \) packets, let \( B \) be the number of available bins. Define \( N_i \) as the number of packets whose value falls in the \( i \)th bin and \( n_i \) as the expected number of packets in the \( i \)th bin under the typical distribution. Then the chi-square statistic is computed as follows:

\[
\chi^2 = \sum_{i=1}^{B} \frac{(N_i - n_i)^2}{n_i}
\]

When the \( N_i \) and \( n_i \) values are large and the \( N \) measurements are independent and drawn from the expected distribution, this value follows the well-known chi-square distribution with \( B-1 \) degrees of freedom.

**Pushback Attack Information**

In the attack detection we have used four rules for detecting the attacks and the next step is to decrease the intensity of attack by passing the attack information to upstream routers. The router which receives the attack signature limits the traffic in the outgoing line through which it got the attack information and it further forwards the attack signature to its upstream routers. The attack information is pushed deep enough so that the traffic is reduced towards the victim machine.

The Attack Signature specified in Table 2 is sent by destination router to upstream routers to mitigate the attack. Where Source IP, is the list of source IP Address which are sending the attack packets to the victim machine. The list is used to identify the source of attack and send the attack signature to the upstream routers to discard the packets from these source machines. Transport protocol is used to know which protocol is used by the attacker to launch the attack. Flow Rate and Packet count are the aggregates of number bits arriving per second and number of packets respectively; they are used to decrease the traffic to the destination router.

<table>
<thead>
<tr>
<th>S.No</th>
<th>Attack Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Source IP ( i )</td>
</tr>
<tr>
<td>2.</td>
<td>Source Port</td>
</tr>
<tr>
<td>3.</td>
<td>Transport protocol</td>
</tr>
<tr>
<td>4.</td>
<td>Flow Rate</td>
</tr>
<tr>
<td>5.</td>
<td>Packet Count</td>
</tr>
<tr>
<td>6.</td>
<td>Attack Type</td>
</tr>
</tbody>
</table>

**Table 1: Attack Signature**

The pushback mechanism is described in the Fig 2, router R0 is connected to the victim machine, sends the attack signature to R2 and R3. Upon receiving the attack signature R2 and R3 reduces the traffic to R0 and the attack signature is further forwarded to R4 and R7.

In the next step the traffic is reduced towards the victim machine by rate limiting the traffic towards the victim machine.

**Fig 2. Attack Mitigation**

**Mitigation of Attack**

After detecting the attack, upstream routers rate limit the traffic towards the victim machine. The main objective is to decrease the attack traffic and to forward the legitimate traffic towards the victim machine. As the intensity of the attack increases the outgoing traffic towards the victim machine will be rate-limited according to the following formula:

\[
TRAFFIC_{OUT} (R_i) = TRAFFIC_{IN} (R_i) \times \lambda \times INTEN_{Ti}
\]

where \( TRAFFIC_{OUT} \) is the outgoing traffic from router \( R_i \) towards the victim machine, \( TRAFFIC_{IN} \) is the incoming traffic at router \( R_i \), and \( INTEN_{Ti} \) is the intensity of attack at router \( R_i \) when the value of
INTEN, is 0, then \( \lambda (\text{INTEN}_i) = 1 \) in order to avoid dropping the legitimate traffic, upstream routers limit the outgoing traffic by dropping the packets from the source IP address contained in the attack traffic.

**IV. METHODOLOGY**

To disagree with the lawful users from using the services of server is the most important aim of attack of Distributed Denial-of-Service. The fig1 shown identify the attack, alleviates the attack and load balances the victim machine [2]. In the identification of attack phase, the detection of the Distributed Denial-of-Service attack is detected by means of constantly monitoring of the traffic patterns is detected in the proposed approach by the router that is joined to the victim. The router makes an attempt to carry out the load balancing after the detection of the attack on the machine of the victim by means of replicating servers by means of the network address translator [5] [14]. The outsized volume of the traffic arrives at the site of the victim during the period of attack as a result replicas of servers were maintained to hold the load, simultaneously the server will not crash since the heavy traffic and the users of legitimate obtain the services from the servers. To alleviate the attack, that is joined to the victim machine pushback the information of the attack to the routers of upstream initiates monitoring the traffic and alleviates the attack [9]. The use of flow entropy variation was introduced in this paper. Once an attack of Distributed Denial-of-Service has been recognized, the victim commences the succeeding process of push back to spot the location of zombies. The victim initially recognizes the routers of upstream that are in the attack tree on the basis of the variations of flow entropy it has gathered, and then accepts requests to the connected instantaneous routers of upstream routers which then recognize the flow of attacks coming from, based on their variations of local entropy that they have supervised [6] [10]. On one occasion the instantaneous routers of upstream have recognized the flow of attack, they will possibly forward the needs to their instantaneous routers of upstream, correspondingly, to recognize the sources of attacker. Additionally this process is repetitive in a fashion of parallel and distributed mode until it reaches the source of attack otherwise the limit of discrimination connecting the flows of attack and lawful flows is fulfilled [15]. The entropy variation is constant; additionally, it is autonomous from patterns of definite distribution and it is steady against enormous fluctuations of flow and the number of flows in the cases of non attack as a result, it can be used as a standard to differentiate the attack flows of Distributed Denial-of-Service.

In the scenario of attack of Distributed Denial-of-Service, the flows by means of destination as the victim consist of legitimate flows and a grouping of flows of attack and legitimate flows. The volumes of various flows augment considerably in an extremely small time period in the attack of Distributed Denial-of-Service when compared with the cases of non attack. Observers at routers will become aware of the spectacular changes; on the other hand, the routers who are not present in the paths of attack, will not be competent to intellect the variations [11].

**V. RESULTS**

The efficacy and competence of the mechanism of entropy variation mechanism based on IP trace back was evaluated. The variation of entropy increases effortlessly against the enhancement of the number of flows that are passing all the way through the local router. The resemblance of the parameters of entropy variation for the two distributions, and the disparity of the variation entropies is reasonably restricted. The entropy variation is constant; additionally, it is autonomous from patterns of definite distribution and it is steady against enormous fluctuations of flow and the number of flows in the cases of non attack as a result, it can be used as a standard to differentiate the attack flows of Distributed Denial-of-Service.

**VI. CONCLUSION**

The use of flow entropy variation was introduced in this paper. The entropy variation is constant and it is autonomous from patterns of definite distribution in addition to, it is steady against enormous...
fluctuations of flow and the number of flows in the cases of non attack as a result, it can be used as a standard to differentiate the attack flows of Distributed Denial-of-Service. Once an attack of Distributed Denial-of-Service has been recognized, the victim commences the succeeding process of pushback to spot the location of zombies. The process is repetitive in a fashion of parallel and distributed mode until it reaches the source of attack otherwise the limit of discrimination connecting the flows of attack and lawful flows is fulfilled.

REFERENCES


