Performance Evaluation of Packet Overloading TTL Based Method using DPM

Mandeep Singh Ramdev[1]
Rimpi Kumari[2]
[1,2]Department of Computer Science, SVIET

Abstract

The ability to traceback an attacker from its origin is a necessary evidence in order to block its packets and to legally prosecute him. The forever changing nature of internet has made it really difficult to locate the culprit. For this various methods and techniques are being used one of which is DPM. This paper will give the performance analysis of DPM with packet overloading combined with TTL recognition.

1. Introduction

Denial of Service (DOS) and Distributed Denial of Service (DDOS) attacks against computer systems result in economic losses for businesses and Public organizations (Paxson, 2001; Mirkovic et al., 2004; Wu et al., 2009). The costs are counted in terms of protecting from events, the loss of assets, the loss of services, the loss of revenue, reputation damages and the recovery costs. Attacks may be launched from anywhere and become difficult to identify the source.

2. Assumption

The assumptions in this paper are borrowed from [1]. Some of them are modified to reflect the fact that the technique is not designed merely for traceback of DoS attacks

- An attacker may generate any random packet.
- Attackers may be aware that they are being tracked.
- Packets may be lost or reordered
- The attack may consist of just a few packets
- The packets may have a different route
- Resources are limited on routers
- Router Security is not compromised
-

3. Denial of Service

DoS attacks can be defined as: An intentional attempt to prevent or degrade the availability of resources. It should be noted that DoS can also result from unintentional human errors, design faults, or software bugs. The prevention of authorized access to resources or the delaying of time-critical operations. Examples of resources in this definition are network bandwidth, processing capacity, disk space, memory and static memory structures. An attack (which doesn't have to be successful) is defined in ANSI's Telecom Glossary 2000 to be an attempt to violate security. This will be used as the basis for defining a DoS attack.

3.1. Direct DDoS Attacks

In direct DDoS attack, the attacker is able to implant zombie software on a number of hosts distributed throughout the internet. Often the DDoS attack involves two levels of zombie machines: master zombies and slave zombies. The hosts of both the machines have been infected with malicious software. The attacker coordinates and triggers the master zombies (handlers), which in turn coordinate and trigger the slave zombies (agents). The use of two levels of zombies makes it more difficult to trace the attack back to its source and provides for a more resilient network of attackers.

3.2. Distributed DDoS Attacks

An attempt to prevent or degrade the availability of resources by using multiple source hosts at the same time to send attack traffic. In the context of DDoS attack, an agent (or daemon or zombie or bot) is defined as a compromised host used to send attack traffic in a DoS attack. A master (or handler) is defined as a compromised host used to control operation of a large set of agents. A DDoS network is defined as a hierarchically structured set of masters and agents to make it easier to control a DDoS attacks by an attacker. DDoS attacks can be classified as either direct or reflector attacks.

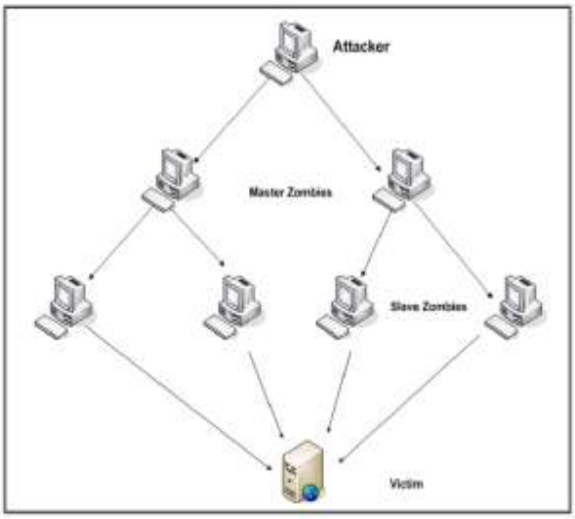

Fig. 1 Distributed DoS

3.3. Reflector DDoS Attacks

In reflector attacks, packets with the victim's address in the source IP address field are sent by slave zombies (agents) to innocent third parties (uninfected machines like web servers, DNS server etc), which in turn will send the reply to the victim (flood the victim). Reflector attacks thus have at least two victims at the same time. Reflector attacks can be more damaging since they involve more machines and thus more traffic and they are more difficult to trace as well.

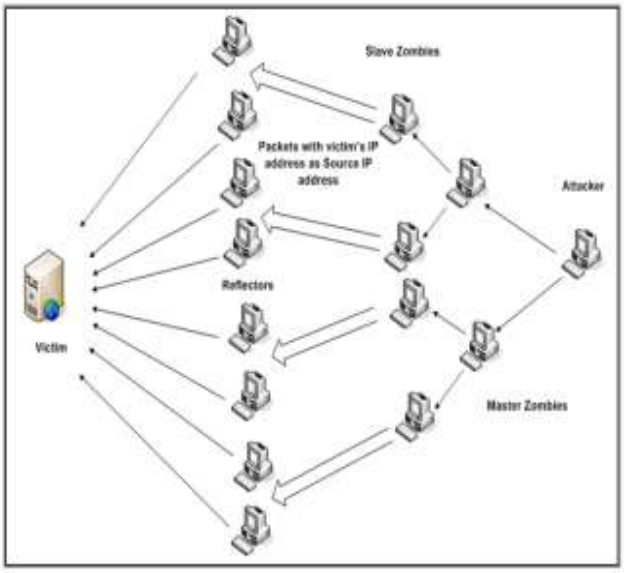

Fig 2 Reflector DoS

4. IP Traceback

IP traceback is a name given to any method for reliably determining the origin of a packet on the Internet. The datagram nature of the Internet makes it difficult to determine the originating host of a packet, the source id supplied in an IP packet can be falsified (IP spoofing) allowing for Denial of Service attacks (DoS) or one-way attacks where the response from the victim host is so well known that return packets need not be received to continue the attack.

In DoS/DDoS attack, attacker uses fake source IP addresses to make tracing and stopping of DoS difficult. This technique is called *IP spoofing*. This technique involves the manipulation of the source IP address in the IP header of a transmitted packet. This gives the attacker a form of anonymity. It is difficult to solve problem of IP Spoofing because of lack of security features in TCP/IP specifications. Ingress filtering, use of cryptographic authentication , IP trace back are some of the approaches used to handle forged IP source addresses. The purpose of IP traceback is to identify the true IP address of a host originating attack packets. IP trace back is vital for quickly restoring normal network functionality and preventing reoccurrences.

4.1. PPM (Probabilistic Packet Marking)

In Probabilistic Packet Marking the mark overloads a rarely used field in IP packet header, i.e., 16-bit IP identification field. The identification of a router could be 32-bit IP address, hash value of IP address or uniquely assigned number. In the last two cases, the length of identification information is variable and could be less than 16 bits. Since the marking space in packet header is too small to record the entire path, routers mark packets with some probability so that each marked packet carries the information of one node in the path. In addition, based on the length of router identification and the implementation of marking procedure, the router may only write part of its identification information into the marking space. While each marked packet represents only a small portion of the path it has traversed, the whole network path can be reconstructed by combining a modest number of such packets. The PPM approach does not incur any storage overhead at routers and the marking procedure (a write and checksum update) can be easily and efficiently executed at current routers. But due to its probabilistic nature, it can only trace the traffic that consists of a large volume of packets. However, this method increases the packet's length at each router hop and can lead to additional fragmentation.

4.2. DPM (Deterministic Packet Marking)

In DPM only ingress edge routers perform the marking. All other routers are exempted from the marking task. Basic DPM uses the 16-bit IP identification field of the IP header and one reserved bit to record the marking information. The IP address of every ingress edge router is split into two segments with 16 bits each. One segment is randomly selected when a packet traverses this router. The idea is that the victim is capable of recovering the whole IP address of an ingress edge router once it obtains both segments from the same router. For the victim to figure out which portion of the IP address the current packet carries, one bit is used as a flag. Therefore, the marking information comprises two parts, the 16-bit partial IP address of the edge router and a 1-bit flag.

Table 1. PPM vs DPM

PPM Scheme	DPM Scheme
Packets are marked with random probability. TTL base marking can also be done using this.	The packets are marked using ingress routers with fixed probability.
The number of packets required for reconstruction is very large.	The number of packets required are very less and hence reconstruction is easier.
The farther is the router, less is the chance that encoding reaches the victim.	The packets are marked by the ingress router only so this problem doesn't arise.
More overhead in network infrastructure.	Less overhead on network infrastructure.

There are two main differences between DPM and PPM. DPM only marks the first ingress edge router, while PPM marks all routers along an attack path. PPM marks probabilistically, while DPM marks every packet at the ingress edge router. The task of ingress address reconstruction in DPM is much simpler than the task of path reconstruction in PPM.

Deterministic Packet Marking uses 17 bit of IP Packet Field (16 bit ID field and 1 bit reserve flag) to mark the packets and these packets remains marked until the packet stays in the network. The marking is done by ingress router closer to the source of attack. As a result it makes sure that egress router does not overwrite the marking in any case. Hence, the scheme makes a distinction between incoming and outgoing packets.

If the attacker tries to spoof the source address of the packet, they would be overwritten with correct marks by the initial router the packet traverses. The code in the ID field assumes that there are almost no IP fragments in the internet. This assumption was made in [5] and is supported by empirical traffic analysis that less than 0.5% of packets are fragmented [6]. In this method an IP address of 32 bits is fragmented into two parts comprising of 16 bits each. The one bit flag will indicate if the fragment is first or second. The advantages of the DPM are enormous-

- It is simple
- It is scalable
- It has no security flaws in its structure

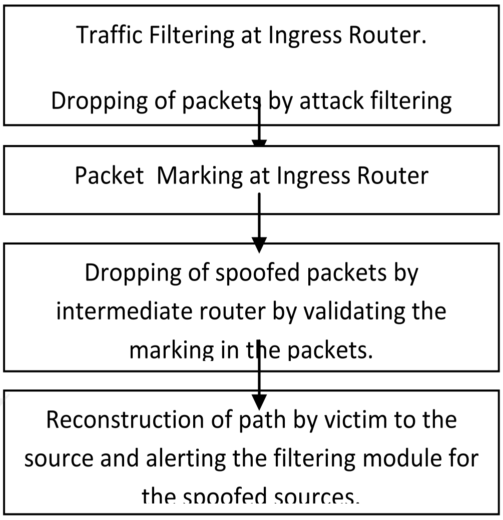
Traffic Filtering at Ingress Router.
Dropping of packets by attack filtering
Packet Marking at Ingress Router
Dropping of spoofed packets by intermediate router by validating the marking in the packets.
Reconstruction of path by victim to the source and alerting the filtering module for the spoofed sources.

Fig 3. IP Traceback Scheme using DPM

5. Pseudo code for DPM Algorithm

sMarking Procedure at router R, edge interface I:

```
 for each incoming packet w
let x be random number from [0,1]
   if x<0.5 then
write $I_{0-15}$ into w.ID_field
write '0' into w.flags[0]
 else
    write $I_{16-31}$ into w.ID_field
    write '0' into w.flags[0]
```

Ingress address reconstruction procedure at victim V:

```
    for each packet w from source Sx
    if IngressTbl[Sx] = = NIL then
 create  IngressTbl[Sx]
    if w.flags[0] = = '0' then
  $IngressTbl[Sx]_{0-15}$ = w.ID_field
    else
```

$\text{IngressTbl[Sx]}_{15\text{-}31}$ = **w.ID_field**

6. Formal Description of algorithm

As noticed above, all edge interfaces on all edge routers will place either the first or the last 16 bits in every incoming packet in the ID field, and set the reserved flag to the appropriate value.

At the victim, I suggest that the table matching the source addresses to the ingress addresses is maintained. The victim would check to see if the table entry for a given source already exists, and create it if it did not. Then, it would write appropriate bits, depending on the value of the reserved flag, into the ingress IP address value.

These algorithms are presented basically for illustration only. DPM code consists of more efficient mechanisms for Ingress Filtering.

7. Analysis

7.1 Performance Analysis-

The performance of the developed method is much better than the rest of algorithms. Due to the different addresses of the ingress and egress the route will be 100% correct all the time in any case. And due to its design, it prevents mark spoofing also.

For this the victim should be receiving a 32bit packet in 2 pieces of 16 bits each. In which last 16 bits is the address of ingress interface. Based upon this if we want to calculate that how many packets it will take for the victim to gather the complete IP address, by careful observation and learning it has been found that it will take approximately 7 packets to generate IP address with a probability more than 99.2% in case of DPM based upon TTL based Packet Overloading. Here Probability comes to approximately 0.9922(1-0.57).

In a similar way, it could be shown that it would take only 10 packets to obtain the ingress IP address with a probability more than 99.96%.

7.2 Topology Based Analysis-

It is prudent to assume that even though a given ISP does not participate in DPM, it will honestly inform other ISPs of this fact. It is, therefore, assumed that an upstream ISP knows whether its client ISP implements DPM.

If all of the clients in fact do implement DPM, then no action is necessary on behalf of the upstream ISP other than to implement DPM on the interfaces facing its own customers if there are any.

If, on the other hand, a client ISP does not implement DPM, it should be treated as a potential attacker by an upstream ISP, and DPM should be implemented on the interface(s) connecting to that client ISP. It should be noted that in most other traceback schemes, if a certain ISP does not wish to participate, traceback through its network will be impossible.

8. Results

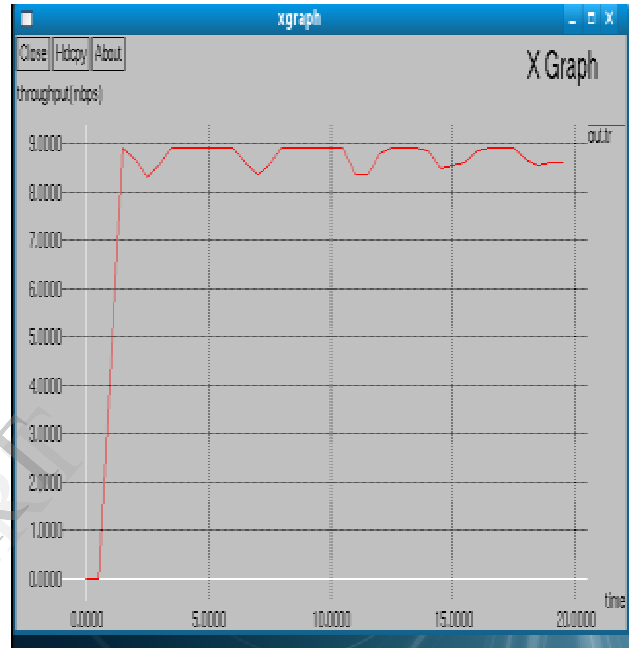

Fig 4. Throughput vs Time Graph

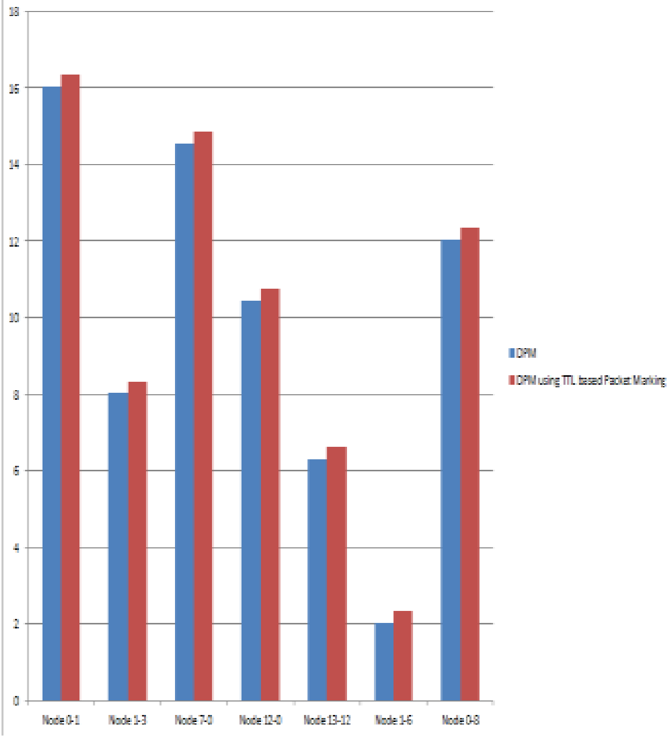

Fig 5. Marked vs Unmarked Traffic Comparison

Table 2. Final Comparison of Traceback Methods

Criteria **Method**	**Mgmt. Overhead**	**Network Overhead**	**Router Overhead**	**Logic Attack Detection**	**Nature**
Input Debugging	High	Low	High	Poor	Reactive
Controlled Flooding	Low	High	Low	Poor	Reactive
Logging	High	Low	High	Good	Proactive
ICMP Traceback	Low	Low	Low	Poor	Proactive
PPM	Low	Low	Low	Poor	Proactive
DPM	Low	Low	Low	Good	Proactive
DPM using TTL based packet marking	Low	Low	Low	Excellent	Proactive

As seen from the table above DPM using TTL based packet marking scheme is the best among all methods used till date. This is only due to fact that DPM considers each interface of a router separate from the other one. Moreover it has several other advantages such as:

- It doesn't reveal the internal topology of the network.
- It is scalable.
- Very easy to implement.
- A single solution for many types of attacks.

9. Conclusion and Future Work

As we have seen that DPM using TTL based packet marking scheme is far more effective than any other scheme due to its structure. But here it is implemented only on IPv4, so in future I want to implement it in IPv6 networks also.

Also seen in Fig.5 there is a little time delay while using this. So some minor changes in the algorithm will improve the overall efficiency. Here the proposed scheme is manual i.e. at some level human intervention is required. In future I want to make it fully automatic so that human intervention will no longer be required and it will work as smart system.

10. References

[1] Stefan Savage, David Wetherall, Member, IEEE, Anna Karlin, and Tom Anderson *"Network Support for IP Traceback*"http://www.ece.cmu.edu/~adrian/630f04/readings/SWKA.pdf

[2] Brian Cusack and Cary Ho *"Tracing Sources of DoS and DDoS attack: Evidential recovery*" , Edith Cowan University Research Online, Auckland, pp. 39-47, Dec 2011.

[3] Ankit Fadia, "*Network Security: A Hacker's Perspective*", pp. 125-127, Course Technology Inc, 2006. ISBN 1598631632.

[4] Vadim Kuznetsov, Andrei Simkin and Helena Standstorm, "*An evaluation of different IP Traceback Approaches*", Proc. of 4th International Conference on Information and Communication Security, Singapore, 2002, pp 37-48.

[5] C. Shannon, D. Moore, and K. Claffy, "Characteristics of fragmented IP traffic on internet links," Proceeding IMW '01 Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement, Pages 83-97,2001.

[6] Andrey Belenky and Nirwan Ansari, "Accommodating Fragmentation in Deterministic Packet Marking for IP Traceback", IEEE GLOBOCOMM 2003