A Survey of IP Traceback Mechanisms to overcome Denial-of-Service Attacks

SHWETA VINCENT, J. IMMANUEL JOHN RAJA
Department of Computer Science and Engineering, School of Computer Science and Technology
Karunya University
Karunya Nagar, Coimbatore – 641 114
INDIA
shwetavincent@karunya.edu.in http://www.karunya.edu

Abstract: - The technique of IP traceback is used to overcome Denial-of-Service attacks. This paper deals with explaining the two types of IP traceback techniques namely, Packet Marking and Packet Logging which have been proposed earlier. The paper further explains about a hybrid IP traceback technique which uses both packet marking and logging. The hybrid technique claims to have a better performance level in terms of reducing the storage overhead at the routers by half and the access time overhead by the number of neighboring routers. Future enhancements have been proposed in the domain of security for the entire system.

Key-Words: - Denial-of-Service Attacks (DoS), IP Traceback, Packet Marking, Packet Logging

1 Introduction

Denial-of-Service (DoS) attacks have been threatening the security of the Internet [2]. A DoS attack is an attempt to make a computer 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 concerted efforts of a person or persons to prevent an Internet site or service from functioning efficiently or at all, temporarily or indefinitely. 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 name servers.

DoS attacks can be classified into flooding attacks and software exploits [2]. Flooding attacks work by flooding a victim with large amounts of packets, while software exploits attack a victim by sending as few as a single packet.

Tracing the paths of IP packets back to their origin, known as IP traceback, is an important step in defending against DoS attacks employing IP spoofing. IP traceback facilitates holding attackers accountable and improving the efficacy of mitigation measures.

The existing approaches for IP traceback can be grouped into two orthogonal dimensions: packet marking [3] and packet logging [5]. The main idea behind packet marking is to record network path information in packets. In mark based IP traceback, routers write their identification information (e.g., IP addresses) into a header field of forwarded packets. The destination node then retrieves the marking information from the received packets and determines the network path.

Due to the limited space of the marking field, routers probabilistically decide to mark packets so that each marked packet carries only partial path information. The network path can be constructed by combining the marking information collected from a number of received packets. This approach is also known as probabilistic packet marking (PPM) [3]. PPM incurs little overhead at routers. However, it requires a flow of marked packets to construct the network path toward their origin.

The basic idea in packet logging is to record the path information at routers. In log-based IP traceback, packets are logged by the routers on the path toward the destination. The network path is then derived based on the logged information at the routers. Compared to mark based IP traceback, the log-based approach is more powerful as it can trace attacks that use a single packet, i.e., software exploit attacks, along with flooding attacks. Historically, packet logging was thought impractical due to the enormous storage space required for packet logs. Snoeren et al. [5] [8] proposed a hash-based IP traceback approach, called Source Path Isolation Engine (SPIE), to realize log-based IP traceback in practice. Their approach reduces the storage overhead significantly through recording packet digests in a space-efficient data structure, a Bloom filter [7]. SPIE has made a significant improvement on the practicality of log-based IP traceback. However, its deployment at high-speed networks has still been a challenging task due to the high storage overhead and access time requirement for recording packet digests.

In this paper, we present a comparative survey of the novel packet marking and logging approaches and a proposed hybrid IP traceback approach based on both packet logging and packet marking, proposed by Chao Gong and Kamil Sarac [1]. The main design goal in the hybrid approach is to maintain the single packet traceback ability of the hash-based approach and, at the same time, alleviate the storage overhead and access time requirement for recording packet digests at routers.

2 Problem Formulation

D Moore et al. in their paper [2] present a statistical view for inferring the effect of denial-of-service attacks on the internet. The technique proposed here termed as 'Backscatter Analysis' provides an estimate of world-wide DoS activity. The only public data available for the survey is obtained from the CSI/ FBI survey.

The following sub-sections provide an overview of the various classifications of attacks in general and the further classification of attacks according to the proposed backscatter analysis technique.

2.1 Attack Types

There are two principal categories of attacks; logical attacks and flooding attacks.

Attacks in the first class, such as the "Ping-of-Death", exploit existing software flaws to cause remote servers to crash or substantially degrade in performance. The second class, flooding attacks, overwhelm the victim's CPU, memory or network resources by sending large numbers of spurious requests. There are two related consequences to a flooding attack- the network load induced and the impact on the victim's CPU.

2.2 Backscatter Analysis Technique

Attackers commonly spoof the source IP address field to conceal the location of the attacking host. When a spoofed packet arrives at the victim, the victim usually sends what it believes to be an appropriate response to the faked IP address. Occasionally an intermediate network device may issue its own reply to the attack via an ICMP message. Again these ICMP messages are sent to the randomly spoofed addresses. Because the attacker's source address is selected at random, the victim's responses are equi-probably distributed across the entire Internet address space, an inadvertent effect called as "backscatter".

Assume that an attacker sends SYN packets to the victim from various spoofed addresses. The victim in turn sends SYN/ACK packets back to the host addresses. This is called as Backscatter. By observing a large enough address range we can effectively "sample' all such DoS activity on the Internet. Contained in these samples are the identity of the victim, information about the kind of attack, and a timestamp from which we can estimate the attack duration.

2.3 Classification of Attacks using Backscatter Analysis

2.3.1 Flow-based classification

A flow is defined as a series of consecutive packets sharing the same IP address and IP protocol. In this classification, the first packet seen for a target creates a new flow and any additional packets from the target are counted as belonging to that flow if the packets are received within five minutes of the most recent packet of this flow.

2.3.2 Event-based classification

Here, the trace is divided into one minute periods and each attack event is recorded during this period. An attack event can be defined by a victim emitting at least ten backscatter packets during one minute period.

2.4 Results of Backscatter Analysis

2.4.1 Response Protocols

The Backscatter analysis was used to decompose the data according to the protocols of responses returned by the victim or an intermediate host. For example, 1837 attacks were derived from TCP backscatter with the RST and ACK flag set. It was observed that 50% of the attacks and 20% of the backscatter packets are TCP packets with the RST flag set.

The next largest protocol category is ICMP host unreachable, comprising roughly 15% of the attacks. It was also seen that a number of SYN/ACK backscatter packets and an equivalent number of assorted ICMP messages, including ICMP echo reply, ICMP protocol unreachable, ICMP fragmentation needed.

2.4.2 Attack Protocols

An attack protocol is defined here as the protocol which must have been used by the attacker to produce the backscatter monitored at the test network. It is seen that more than 90% of the attacks use TCP, but a smaller number of ICMP based attacks produce a disproportionate number of the backscatter packets.

2.4.3 Attack Rate

The attack event rate is calculated by multiplying the average arrival rate of the backscatter packets by 256, assuming that an attack represents a random sampling across the entire address space of which 1/256 of it is monitored. Comparing the distributions it is noted that the uniform random attacks have a lower rate than the distribution of all attacks. Half the uniform, random attack events have a packet rate greater than 250, whereas half of all the attack events have a packet rate greater than 350.

2.4.4 Attack Duration

A cumulative distribution graph is plotted and it is seen that most attacks are relatively short: 50% of attacks are less than 10 minutes in duration, 80% are less than 30 minutes, and 90% last less than an hour. 2% of attacks are greater than 5 hours, 1% are greater than 10 hours and dozens span multiple days.

3 Problem Solution

This section of our paper focuses on the various IP Traceback mechanisms based on packet marking, logging and a hybrid approach using both marking and logging together.

3.1 Network Support for IP Traceback (Packet Marking Approach)

S. Savage et al. [3] present a probabilistic marking scheme with partial path information stored in each packet. While each marked packet represents only a "sample" of the path it has traversed, by combining a modest number of such packets a victim can reconstruct the entire path.

3.1.1 Overview of Marking Techniques

All marking algorithms have two components: a marking procedure executed by routers in the network and *a* path reconstruction procedure implemented by the victim. A router "marks" one or more packets by augmenting them with additional information about the path they are traveling. The victim attempts to reconstruct the attack path using only the information in these marked packets.

The various marking techniques proposed in this by S. Savage et al. in their paper are, Node Append, Node Sampling and Edge Sampling.

Node Append similar to the Record Route option consists of appending each node's address to the end of a packet as it traverses through the network from attacker to victim.

Node Sampling is used to reduce both the router overhead and the per-packet space requirement, by sampling the path one node at a time instead of recording the entire path. A single static "node" field is reserved in the packet header-large enough to hold a single router address (i.e., 32 bits for IPv4). Upon receiving a packet, each router chooses to write its address in the node field with some probability p. After enough packets have been sent, the victim will have received at least one sample for every router in the attack path.

The Edge Sampling technique is used to explicitly encode edges in the attack path rather than simply individual nodes. To do this, reserve two static address sized fields, start and end, in each packet to represent the routers at each end of a link, as well as an additional small field to represent the distance of an edge sample from the victim. When a router decides to mark a packet, it writes its own address into the start field and writes a zero into the distance field. Otherwise, if the distance field is already zero this indicates that the packet was marked by the previous router. In this case, the router writes its own address into the end field--thereby representing the edge between itself and the previous router—and increments the distance field to one. Finally, if the router does not mark the packet, then it always increments the distance field.

3.1.2 Encoding Issues in Edge Sampling

The edge-sampling algorithm requires 72 bits of space in every IP packet (two 32-bit IP addresses and 8 bits for distance to represent the theoretical maximum number of hops allowed using IP). This paper has developed a modified version of edge sampling that dramatically reduces the space requirement in return for a modest increase in convergence time and a reduction in robustness to multiple attackers.

Three techniques to reduce per-packet storage requirements while preserving robustness are used. First, each edge in half the space is encoded by representing it as the exclusive-or (XOR) of the two IP addresses making up the edge.

The second modification further reduces the per-packet space requirements by subdividing each edge-id into some number k of smaller non-overlapping fragments.

Finally, unlike full IP addresses, edge-id fragments are not unique and multiple fragments from different edge-ids may have the same value. If there are multiple attackers, a victim may receive multiple edge fragments with the same offset and distance. To reduce the probability of accidentally reconstructing a "false" edge-id by combining fragments from different paths, the size of each router address is increased, and hence each edge-id, by bit-interleaving its IP address with a random hash of itself. The victim constructs candidate edge-ids by combining all combinations of fragments at each distance with disjoint offset values.

3.2 Hash-based IP Traceback Approach (Source Path Isolation Engine)

A. Snoeren et al. in their papers on Source Path Isolation Engine, [5] [8], speak about the ability to identify the source of a particular IP packet given a copy of the packet to be traced, its destination and an approximate time of receipt.

3.2.1 Packet Digesting

SPIE uses auditing techniques to support the traceback of individual packets. Traffic auditing is accomplished by computing and storing packet digests rather than storing the packet themselves. SPIE computes digests over the invariant portion of the IP header and the first 8 bytes of the payload.

Constructing a digest table is accomplished using a space-efficient data structure known as Bloom filter. A Bloom filter computers k distinct packet digest for each packet using independent uniform hash functions, and uses the n-bit results to index into a 2^n-sized bit array. The array is initialized to all zeros and bits are se to one as packets are received.

3.2.2 Source Path Isolation Engine (SPIE)

SPIE enhanced routers maintain a cache of packet digests for recently forwarded traffic. If a packet is determined to be offensive by some intrusion detection system, a query is dispatched to SPIE which in turn queries routers for packet digests of the relevant time periods. The results of this query are used in a simulated reverse-path flooding algorithm to build an attack graph that indicates the packet's source.

Fig.1 shows the three major architectural components of the SPIE system. Each SPIE–enhanced router has a Data Generation Agent (DGA) associated with it. The DGA can be implemented and deployed as a software agent, an interface card plug to the switching background bus, or a separate auxiliary box connected to the router through some auxiliary interface. SPIE Collection and Reduction Agents (SCARs) are responsible for a particular region of the network, serving as data concentration points for several routers and facilitating traceback of any packets that traverse the region. Upon request each SCAR produces an attack graph for its particular region. The SPIE Traceback Manager (STM) controls the whole SPIE system. The STM is in the interface to the intrusion detection system or other entity requesting a packet trace. Upon completion of the traceback process the STM replies to the intrusion detection system with the final attack graph.

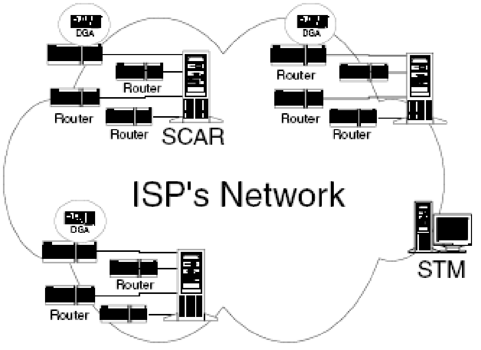

Fig. 1: SPIE Architecture (From [5] [8])

3.3 Novel Hybrid Schemes Employing Packet Marking and Logging (DLLT and PPPM)

B. Al-Duwairi and G. Manimaran in their paper titled Novel Hybrid Schemes Employing Packet Marking and Logging for IP Traceback [6], explain two techniques namely, Distribute Linked List Traceback and Probabilistic Pipelined Packet Marking.

3.3.1 Distributed Linked List Traceback (DLLT)

DLLT is based on the 'store, mark and forward' approach with a fixed size marking field for each packet. Any router that marks a packet, stores the content of the marking field in a 'Marking table' maintained at the router or else it forwards it to the next router. A linked list is used because the marking field servers as a pointer to the last router that did the marking for a given packet and the marking table of that router contains a pointer i.e. an IP address to the previous marking router and so on.

When a router receives a packet, it marks the packet with a probability 'q'. If it has been marked previously, the router stores this information before remarking it. Here only a fraction of traffic is logged at each router without putting a heavy burden on the routers. For storage Bloom filters are used. Each router has a Digest Array (Bloom filter) and a Marking Information Table. Each packet has a 32-bit field which contains the IP address of the marking router.

In a Marking Information table (MIT) the information contained is, the IP address of the previous router that marked a given packet which serves as a pointer to that router and the hash function number (hfn) found from the marked packet i.e. the number used to index the MIT.

3.3.2 Probabilistic Pipelined Packet Marking (PPPM)

Pipelining is used to allow more than one instruction to be in some stage of execution at the same time. A router that marks a packet represents a pipeline stage, the marking process represents the instruction, execution and the propagation of marking information from one marking router to another represents the flow of instructions in a pipelined system. The objective of PPPM is to let the destination know about all routers that were involved in marking a certain packet, P, using a constant space in the IP packet header without incurring long term storage overhead at the intermediate routers.

Each marking information field in PPPM at each packet has an IP address of the marking router (MR) and an ID used to link the marking done for a given packet by different routers. The fields required in each packet for marking are a 32 bit IP address, an 8 bit TTL and a c bit ID.

For each destination, the most recent marking information is buffered. Bloom filters are used here too.

Here it is sufficient to obtain only one mark per router to conduct a traceback process. A Bloom filter is used in PPPM to indicate whether a certain destination has buffered marking information or not. Even in PPPM, probabilistic edge marking can be realized in the proposed marking and buffering procedure by adding a 1 bit marking flag. Once a DoS attack is detected, the victim starts the source identification process using K attack packets as input.

3.4 A Hybrid Approach for Single-Packet IP Traceback

Chao Gong and Kamil Sarac in their paper titled A More Practical Approach for Single-Packet IP Traceback using Packet Logging and Marking, present a Hybrid Single-Packet IP Traceback (HIT) Approach [1]. This approach ascertains that, in comparison to SPIE, HIT has the ability to trace a single IP packet while reducing the storage overhead by half and the access time overhead by the number of neighboring routers.

3.4.1 Hybrid Single Packet IP Traceback (HIT)

In HIT, each traceback enabled router could commit both packet marking and packet logging operations. When forwarding a packet, routers decide to mark or log the packet depending on whether there is free space available in the marking field of the packet. in this way, logging a packet at a router records not only the current router but also the k upstream routers on the network path. While a packet is traversing the network, logging the packet at every (k+1) the router is enough to record the complete network path. In HIT, the marking field of a packet accommodates the identification information of a single router. While a packet is traversing the network, the routers on the path mark the packet deterministically but log the packets alternately.

Router Operation: In HIT, each traceback-enabled router is assigned a 15 bit ID number. The marking values are encoded in the 16 bit identification field of the IP header. The leftmost bit is termed the logging flag bit. It is set to 1 if the current router commits a logging operation on the packet; otherwise, it is set to 0. The remaining 15 bits are used to store a router ID number. HIT computes packet digests similar to SPIE. A 20 byte IP header excluding three variant fields (TTL, TOS, and checksum) plus the first 8 bytes of the payload are used. If the logging flag is 0, the router chooses to commit both logging and marking, if it is 1, the router chooses to commit only a marking operation.

Digest Table: Similar to SPIE, HIT stores packet digest in digest tables that are implemented with Bloom filters. However, in HIT, routers may maintain multiple digest tables to record, multiple packet digests at the same time. Each digest table is associated with one or more router ID numbers. Each digest table is associated with the ID number of one neighbouring router. When the router decides to log a packet, it examines the router ID number carried by the packet and then stores the packet digest into the corresponding digest table. When a digest table gets saturated, it is paged out and archived for some period of time. The length of the time period depends on the resource constraints of routers and the requirements of the IP traceback scheme.

Traceback Process: Similar to SPIE, the traceback process in HIT is managed by traceback servers equipped with the network topology information. The victim under DoS attack dispatches a traceback request to the traceback server, providing an attack packet and the time of receiving the attack packet. From the value of the logging flag bit in the packet, the traceback server can further determine whether the last-hop router logged the packet. When a router receives a query from the traceback server, the router examines all digest tables of the relevant time period for the attack packet. If an entry exists for the packet, the current router is believed to be on the attack path, and the router indicated by the router ID number is considered as the upstream router on the attack path.

Compatibility and Transformation: HIT is able to achieve backward compatibility and trace packets undergoing transformation. The main idea of the improvements is that, routers do not mark but log IP fragments and routers both mark and log packets undergoing transformations. Each router maintains a special digest called the Fragmentation and Transformation Digest (FTDT). FTDT is only for storing the digests of IP fragment and the digests of IP packets that have been transformed at the current router. Each router also maintains a Transform Lookup table (TLT) corresponding to a FTDT. TLT records packet transformation information and is indexed by packet digests.

Fig. 2 shows the HIT approach as implemented in [1].

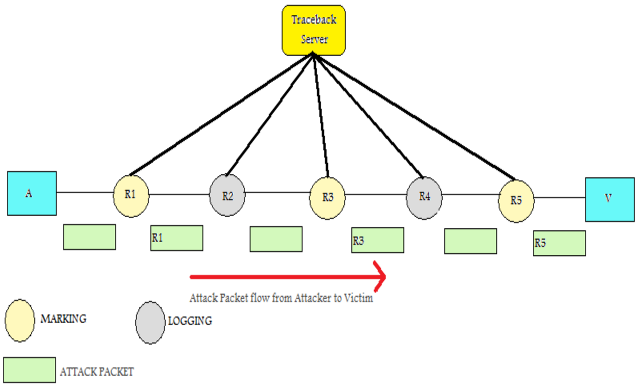

Fig 2. HIT Approach

The Table 1 is a comparative study of the two hybrid approaches dealt with earlier in this paper i.e. DLLT and HIT.

Table 1: Comparison of DLLT and HIT

METRIC	DLLT	HIT
Number of packets required for traceback	Multiple packets	One packet
Marking Overhead on packets	Marking field is 34 bits	Marking field is 16 bits
Router Processing Overhead while creating audit trails	Denoted by marking/logging probability q = 0.05	Router marks 100% and logs 50% of the traffic

A clear explanation is obtained from Table 1 which shows that the DLLT approach as proposed in [6] has performance issues in comparison to HIT as proposed in [1].

4 Conclusion

Further enhancements to the HIT approach [1] could be made in the domain of security. As the authors have mentioned, attackers may write a forged marking value into attack packets. This only helps the attacker to prefix a false router to the attack path. Since a packet is marked by each and logged by every other router, the attacker cannot introduce an arbitrary attack path. In order to successfully exploit this vulnerability, 1) attackers have to know the ID numbers of the neighboring routers of the first router on the path, and 2) attack packets need to enter the network at a router port that is not (or cannot be) upgraded to mark packets. But it could be possible to make the system vulnerable to further attacks if a Distributed Denial of Service (DDoS) attack is used where a group of attackers (probably neighbours) could attack a single victim server. In such a case, the attackers would know the ID numbers of the neighbouring routers which would breach the security concept as just mentioned above by the HIT proposal. Hence, future enhancements to HIT could include, finding out whether an IP address has been spoofed and then going on to trace the packet back to its source.

References:

[1] Chao Gong, Kamil Sarac, “A More Practical Approach for Single-Packet IP Traceback Using Packet Logging and Marking”, IEEE Transactions on Parallel and Distributed Sysems, Vol. 19, No.10, October 2008.

[2] D. Moore, G.Voelker and S.Savage, “Inferring Internet Denial of Service Activity,” Proc. 10th Usenix Security Symp., Aug 2001.

[3] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, “Network Support for IP Traceback,” IEEE/ACM Trans. Networking, vol. 9, no. 3, pp. 226-237, 2001.

[4] T. Doeppner, P. Klein, and A. Koyfman, “Using Router Stamping to Identify the Source of IP Packets,” Proc. Seventh ACM Conf. Computer and Comm. Security (CCS ’00), Nov. 2000.

[5] A. Snoeren, C. Partridge, L. Sanchez, C. Jones, F. Tchakountio, B. Schwartz, S. Kent, and W. Strayer, “Single-Packet IP Traceback,” IEEE/ACM Trans. Networking, vol. 10, no. 6, pp. 721-734, 2002.

[6] B. Al-Duwairi and G. Manimaran, “Novel Hybrid Schemes Employing Packet Marking and Logging for IP Traceback,” IEEE Trans. Parallel and Distributed Systems, vol. 17, no. 5, pp. 403- 418, May 2006.

[7] B. Bloom, “Space/Time Trade-Offs in Hash Coding with Allowable Errors,” Comm. ACM, vol. 13, no. 7, pp. 422-426, 1970.

[8] A. Snoeren, C. Partridge, L. Sanchez, C. Jones, F. Tchakountio, S. Kent, and W. Strayer, “Hash-Based IP Traceback,” Proc. ACM SIGCOMM ’01, Aug. 2001.