International Journal on Future Revolution in Computer Science & Communication Engineering
Volume: 4 Issue: 1
ISSN: 2454-4248
229 – 236

A Logarithmic and Exponentiation Based IP Traceback Scheme with Zero Logging and Storage Overhead

Dr. S. Malliga
Professor/CSE
Kongu Engineering College
Erode, Tamil Nadu, India
mallisenthil@kongu.ac.in

Dr. C. S. Kanimozhiselvi
Associate Professor/CSE
Kongu Engineering College
Erode, Tamil Nadu, India
kanimozhi@kongu.ac.in

Dr. S. V. Kogilavani
Assistant Professor(SRG)/CSE
Kongu Engineering College
Erode, Tamil Nadu, India
kogilavani@kongu.ac.in

***Abstract*—** IP spoofing is sending Internet Protocol (IP) packets with a forged source IP address to conceal the identity of the sender. A Denial-of-Service attack is an attempt to make a machine unavailable to the intended users. This attack employs IP Spoofing to flood the victim with overwhelming traffic, thus bringing it down. To prevent such attacks, it is essential to find out the real source of these attacks. IP Traceback is a technique for reliably determining the true origin of a packet. To traceback, a marking and a traceback algorithm are proposed here which use logarithmic and exponentiation respectively. The time required for marking and traceback has been evaluated and compared with state-of-art techniques. The percentage of increase in marking information is found to be very less in the proposed system. It is also demonstrated that the proposed system does not require logging at any of the intermediate routers thus leading to zero logging and storage overhead. The system also provides 100% traceback accuracy.

Keywords*: IP Spoofing, DoS, IP Traceback, Marking and Traceback, Logarithmic and Exponentiation, Traceback Accuracy*

*****

I. INTRODUCTION

The objective of a DoS (Denial of Service) / DDoS (Distributed DoS) attacker is to degrade the resources on a server, so that genuine users are deprived of the services they need. Among all the fields, the header of each IP packet contains the source and destination address of the packet. When a packet goes from a source to a destination, the source address in the packet is never authenticated on its way. This weakness is exploited by DoS/DDoS attackers, which leads to IP spoofing. In IP spoofing, an attacker gains illegal access to a victim by making it appear that a message has come from a trusted machine by spoofing the IP address of that machine. The intention of spoofing is to hide the real identity of the attackers, which makes it difficult to find the real source of the attacks. Mandia et. al. [19] described that the DoS/DDoS attacks are destructive, resource and bandwidth consuming.

According to Gong and Sarac [11], there are two classes of DoS/DDoS attacks, which are flooding attacks and software exploits. Flooding attacks flood a victim by huge amount of packets whereas software exploits employ the vulnerabilities of the TCP/IP suite. The source of these attacks can be identified by tracing the packets. DoS/DDoS attacks need not be always flooding attack. Even a single, well targeted attack packet can depose an entire system [21]. The fact that makes difficult to prevent DoS/DDoS attacks is that the illegitimate packets are indistinguishable from the legitimate packets.

The solutions against the DoS/DDoS attacks are generally classified into proactive and reactive [25]. Traceback, a reactive technique, is used to find the origin of malicious traffic. Two main categories of traceback techniques are in-band and out-of-band approaches [13].

In-band approaches are further classified into packet marking and/or packet logging [11],[10],[12]. In packet marking, intermediate routers mark their identification information, either probabilistically (PPM) [2],[9],[22],[23] or deterministically (DPM) [3],[23] in the packets that they forward. These marked packets are then used to reconstruct the path traversed by the packets. In Packet logging, the routers store the packets that they forward and these logged packets are used for reconstruction of the path during traceback [24]. To take the benefits of both packet marking and logging, hybrid methods have also been proposed [11],[10],[1],[16],[17],[18]. These techniques mark the packets with router identification information. When the marking information overflows beyond the fields used for marking, the routers log the packets, clear the marking information and restart marking.

In out-of-band approach, the trace information is sent in a separate trace packet, namely, ICMP (Internet Control Message Protocol) packets whereas in in-band approach, the trace is done by using IP packet. Such schemes are said to be ICMP traceback schemes. In the approaches like iTrace [4], Intension-driven ICMP [20], iCaddie [27] and SPITRI [26], the path information is collected in a separate ICMP packet. The ICMP traceback schemes send ICMP traceback packets towards destination host of an IP packet. These ICMP packets are then used to traceback the origin.

In this article, we propose a new IP traceback algorithm which consists of packet marking and traceback algorithms. These algorithms use logarithmic and exponentiation operations. The logarithmic and exponentiation are inverse to each other. A significant contribution of the proposed system is that it does not require logging at all in any of the intermediate routers. The proposed system also tracks back to the real source with 0% false positives/negatives.

The rest of the article is orchestrated in the following way: Section 2 reviews the state-of-the art techniques for traceback

 ISSN: 2454-4248
229 – 236

and identifies the overhead incurred in marking, tracking back and logging of packets. Section 3 addresses the design goals of an ideal traceback system. Section 4 proposes a new marking and traceback system. Section 5 attempts to demonstrate the working of the proposed algorithm with a numerical example. The performance of the proposed system is evaluated with different metrics and the results are presented in Section 6. Finally, in Section 7, we present the conclusion of the proposed work.

II. HYBRID PACKET MARKING LOGGING APPROACHES FOR IP TRACEBACK

To date, several traceback techniques have been proposed and evaluated. Having given introduction to the various approaches for IP traceback, this section reviews the recent hybrid approaches for IP traceback.

A MOdulo REverse Modulo (MORE)

MORE [17] has two algorithms. The first algorithm is for packet marking along with logging. The second algorithm involves traceback of packets to their real origin. The objective of the marking algorithm is to keep track of all routers that contribute for marking the packets. Each router marks the packet using modulo operation. For marking, IP ID field in the IP header is used. For Traceback, MORE uses reverse modulo to find the inbound interface of the traceback requested packet using the marking information present in the packet.

Both MORE and its earlier version, MRT [16] use modulo and reverse module techniques for traceback. But, it is found that MORE and MRT consume more space while the packets are logged at the intermediate routers. Also, every packet is logged when its ID field is insufficient, which increases storage requirements enormously at the routers.

B Logging and Storage based Hybrid IP Traceback

Gong and Sarac [11] has proposed a hybrid traceback method which uses both packet marking and logging. The main idea of this approach is to record the path information partially at routers and partially in packets. Every router marks the packets whereas every alternative router logs the packets. Thus, this approach reduces the storage overhead of packet digest to one half and access time requirement for recording packets by a factor of degree of the router. The authors have extended their work and proposed that, instead of logging at the alternative routers, the routers decide to mark if free space available in the marking field. This means that logging is done at the every k^{th} router. The storage overhead and access time requirement for recording digests reduce by a factor of k+1. In [17], it was demonstrated that MORE outperforms this approach.

C Precise and Practical IP Traceback (PPIT)

In PPIT [28], the intermediate routers do marking or logging alternatively in a certain manner (i.e.) routers, at every three hops, compute packet digests and record the digests of the packets. The main idea of PPIT is to add a path authentication to the packet digest, which can eliminate the incorrect path and make full use of the marking space to reduce storage overhead further. As in SPIE [24], Bloom filter is used by PPIT for storing the packet digests. Since every three hop routers store the packet digests, the logging overhead is high.

D RIHT

RIHT [29] marks interface number of routers in the packets so as to trace the path of packets. Since the marking field on each packet is limited, RIHT needs to log the marking information into a hash table and stores the table index on the packet. This marking/logging process is repeated until the packet reaches its destination. After that, the process is reversed to trace back to the origin of attack packets. RIHT uses IP ID and offset fields to mark packets, which are used by fragmented packets. So RIHT certainly suffers from fragment and drop issues for its packet reassembly.

E Hybrid IP Traceback (HIT)

M.H. Yang proposed [30],[31] two hybrid traceback algorithms named HAHIT (HIT with High Accuracy) and HIT. In HAHIT, a router receiving a packet, computes new marking information and stores it in the packet. If the marking details cannot be placed in IP ID field, HAHIT computes the hash value on marking in the packet and stores the marking information along with the incoming interface in a log table. Multitables are used to store the packets' digest when log tables overflow along with a timestamp. As log tables to be used are based on the timestamp, the gap between the times at which the packet's digest is recorded and the traceback request for the packet is initialized to be within a reasonable limit. Otherwise, redundant search of log tables may incur. HIT calculates the marking details similar to HAHIT. It differs in the way that it follows for logging. Whenever no space is found in the packet for marking, a router compares it's degree with a predefined threshold. Based on this, either both the marking information from the upstream router and the incoming interface or marking information is alone logged. HIT claims that it definitely reduces storage requirements at the routers compared to HAHIT.

F RevisedMORE

In this proposal [18], each packet is marked in its ID field. When a packet does not have space to embed the marking information, it will be logged. Instead of logging the packets based on source IP addresses, Revised-MORE logs them based on the subnet addresses. To enable traceback, the marking information in the packet is used. To retrieve the logged information at a router, the hash table is used. The subnet address is used to find the entry in the hash table. Once the marking information and the inbound interface are found at a given router, the packet is forwarded to the upstream router using the interface. Every router repeats this process till the first hop router of the packet is located. Since Revised-MORE logs packets based on subnet addresses, the storage has been reduced considerably.

 ISSN: 2454-4248
229 – 236

III. BACKGROUND AND MOTIVATIONS

A *Traceback Problem and Terminology*

IP traceback is a technique that helps to detect the real origin of a packet. Let R_1, R_2 .... R_n be the ordered list of routers which forward the packets from a source to a destination. The process of reconstructing the path of the traffic is known as traceback process. IP traceback process reconstructs the path consists of R_n, R_{n-1} .... R_1.

The following glossary of terms is intended to assist in better understanding commonly used terms and concepts in this article: an attack packet is the one whose source is to be traced. A victim is the destination of attack traffic. Edge or outbound router or first hop router is a system that attaches a local area network with the Internet. *d* is the number of inbound interfaces of a router. These interfaces are assigned with IDs *0, 1,…d-1*. *d* is also referred to as degree of the router. *Intf* is the interface through which the packets go into a router. *False positive* occurs when a router, which is not along the traceback path, is identified as present along the traceback path. *False negative* occurs when a router, which is along the traceback path, is not identified as present along the traceback path.

B *Design Objectives*

An ideal IP traceback mechanism should exhibit the following features:

- Providing the accurate and complete information about the routers along the path
- Ability to carry out single packet IP traceback
- Ability to locate the real source of attacking traffic or at least the outbound router.
- No or low packet logging and storage overhead:
- Low traceback process overhead.
- High traceback accuracy

Designing a traceback method having the above these design goals is a challenging task. This paper makes an attempt to propose an IP traceback system having these features.

IV. LOGARITHMIC AND EXPONENTIATION BASED IP TRACEBACK (LEIP)

The proposed system, Logarithmic and Exponentiation based IP traceback (LEIP), consists of two algorithms namely: marking and traceback algorithms. LEIP system uses two mathematical operations, namely logarithmic and exponentiation. The proposed marking algorithm cumulatively collects identification information of the routers along the path the packets travel. To collect this path information, the marking algorithm uses simple mathematical operations like applying log and division. The path information gathered from all the routers will be used to traceback from last hop router to the first hop router. So, each router applies exponentiation on the path information to find the upstream router. Since logarithm and exponentiation are inverse operations, it is possible to revert back to previous values. The details of these algorithms are presented below.

A *Logarithmic based Marking Algorithm*

The marking algorithm is based on the mathematical operations: logarithmic and division. Each router along the path uses these operations to record its identification information and to generate complete path information. The information is the interface through which the packet enters into the routers. The IP addresses and ID assigned to these interfaces are maintained by each router in a table. Hence, every router records the interface (I_{id}) through which the packets arrive at the router. The routers use the formula in Equation (1) for recording the path information.

$$P_{mark} = (1/\log_{I_{id}}(d * P_{markold})) + I_{id} \quad (1)$$

Here $P_{markold}$, at any router, is the marking information received from its upstream router. It is set to 1 for the first hop router. The subsequent routers along the path use the same formula for recording their identification information. Next, it is important to locate a place in the IP packet for marking. All the existing marking algorithms use the part of the IP header which are used for handling the fragmented packets. This part includes 16-bit Identification, 3-bit flags and 13-bit offset fields. Most of the algorithms [11][10][17][18][29][30] use only 16-bit IP ID field for recording the path information. There are a few algorithms [16, 29] which use all the fields of fragmented packets. The motivation behind using these fields for marking is that only 0.25% of the packets over the internet are fragmented [8] and these fields are needed only when a packet is fragmented into segments. When IP ID field is used for marking, the flag and offset fields used during fragmentation is of no use. But, using the offset bits would cause confusion at the destination as non zero values in the fragmentation offset field would be misinterpreted as IP fragments. Also, 32-bit marking has been employed at the cost of backward compatibility. To address this issue, in LEIP, only 16-bit IP ID field is used for marking. Figure 1 shows the field in the IP packet where the path information is recorded.

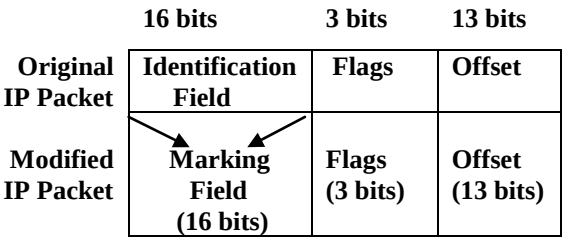

Figure. 1. IP ID field for embedding path information

Figure 2 shows the format of modified IP header containing the path identification information.

Version	IHL	ToS (8)	Total Length (16)
Marking Field (16 bits)		Flag	Fragment
TTL (8)		Protocol	Header Checksum
Source IP Address (32)			

 ISSN: 2454-4248
229 – 236

Destination IP Address (32)	
Options	Padding
Pay load	

Figure. 2. Format of IP packet embedded with path information

The details of marking algorithm are depicted in Figure 3.

1. For each packet *p* entering into a router *R* via *intf*
2. {
3. If *R* is the outbound router
4. Initialize $P_{markold}$ to 1
5. Else {
6. Let *d* be the number of interfaces at *R*
7. Find the ID (I_{id}) assigned to the *intf* through which the packet enters into *R*
8. Calculate the new marking information $P_{mark} = (1/\log_{I_{id}}(d * P_{markold})) + I_{id}$
9. }
10. }

Figure. 3. Marking algorithm

The marking procedure used by a router is simple. The router has to just find the interface ID and calculates the path information as given in Figure. 3. The path information recorded by a router is forwarded to the downstream routers for recording their path information. The reason for doing so is to get the cumulative information at the destination. That is, the Equation (1) is derived in such a way that the path information has cumulative effect of the incoming interface at each router. When the packet arrives at the destination, the marking in the packet carries the details of the complete path traversed by the packet.

B Exponentiation based Traceback Algorithm

To detect that a system is under DoS/DDoS attacks, it is assumed that the system is equipped with some attack detection mechanisms. Once the attack is detected, the source of the attack is to be located to prevent the attacks from further crippling into the destination. To locate the source, the path taken by the packets to reach the destination has to be identified. That is, path reconstruction procedure has to be invoked. To reconstruct the path, the marking (i.e.) the path information recorded in the packet is used. Since the marking is done using logarithmic function, exponentiation, the reverse function of logarithmic, is used to reconstruct the path. The reconstruction process starts at the last hop router. This router applies exponentiation on the path information to find the path information received from the upstream routers and also the interface through which the packet has entered into this router. Using this interface, the traceback request is forwarded to the upstream router. Each upstream router applies exponentiation successively on the path information it received from the downstream router to find the interface and the previous path information. This process is repeated till the first hop router is reached. From the first hop router, it is quite easy to find the origin of the packet. The formulae used for finding the interfaces ID and marking from the upstream routers are given in Equation (2) and Equation (3).

$$I_{id} = \lfloor P_{mark} \rfloor \quad (2)$$

$$P\_mark = exp(I\_id, 1/(P\_mark - \lfloor P\_mark \rfloor))/d \quad (3)$$

In the above equations, $\lfloor x \rfloor$ represents the largest integer that is smaller than or equal to x. The steps in traceback algorithm are enumerated in Figure. 4.

1. For each packet *p* at router R_c.
2. {
3. Let *d* be the number of interfaces of R_c
4. Calculate $I_{id} = \lfloor P_{mark} \rfloor$
5. Find $P_{marktemp} = 1/(P_{mark} - \lfloor P_{mark} \rfloor)$
6. Calculate $P_{mark} = exp(I_{id}, P_{marktemp}) / d$
7. Forward the packet with P_{mark} to the upstream router connected via I_{id}
8. }

Figure. 4. Traceback algorithm

The formulae used for path reconstruction was designed in such a way that the cumulative results produced by marking algorithm can be reverted back successively by the upstream routers. This means, each router would find the path information what is has received from the upstream routers while applying exponentiation during path reconstruction. Successive execution of the traceback algorithm stops when the P_{mark} reaches 1. This indicates that the traceback process has arrived at the first hop router.

V. UNDERSTANDING THE WORKING OF LEIP

This section uses a numerical example to illustrate the working of marking and traceback process. Exponents and Logarithms work well together because they undo each other. The logarithmic function with base *b* is the function.

$$y = log_b\ x \quad (4)$$

An exponential function is the inverse of a logarithm function. Corresponding to every logarithm function with base *b*, there is an exponential function with base *b*:

$$y = b^x$$

For illustration, we have considered a path that consists of 11 routers along it. The number of interfaces and the interface through the packet enter into the routers is given in Table 1. The marking information calculated by each router using Equation (1) is also presented in the table.

Table 1. Illustration of recording of marking information by marking algorithm

(R)	(d)	(I_{id})	($P_{markold}$)	(P_{mark})
1	8	3	1	3.528320834
2	4	2	3.528320834	2.261849903
3	7	6	2.261849903	6.648696246
4	9	4	6.648696246	4.338810977
5	25	21	4.338810977	21.64964001
6	8	7	21.64964001	7.377521862

 ISSN: 2454-4248
229 – 236

7	7	5	7.377521862	5.408036496
8	18	9	5.408036496	9.479925913
9	4	2	9.479925913	2.190662285
10	11	4	2.190662285	4.435654039
11	9	7	4.435654039	7.527790374

As can be seen from Table 1, the marking recorded by a router is used by the downstream router to calculate its marking. This is done to create a cumulative effect of markings at each router.

Next, we demonstrate how the marking information can be reverted back to the original marking made at each router.

Table 2. Traceback Algorithm – an illustration

(R)	(d)	(I_{id})	$(P_{mark})^1$	$(I_{id})^1$	$(P_{mark})^2$
11	9	7	7.527790374	7	4.435654039
10	11	4	4.435654039	4	2.190662285
9	4	2	2.190662285	2	9.479925913
8	18	9	9.479925913	9	5.408036496
7	7	5	5.408036496	5	7.377521888
6	8	7	7.377521888	7	21.64963236
5	25	21	21.64963236	21	4.339050631
4	9	4	4.339050631	4	6.648686241
3	7	6	6.648696246	6	2.261849902
2	4	2	2.261849902	2	3.528320868
1	8	3	3.528320868	3	0.999999863 (~ 1)
$(P_{mark})^1$ – Marking at (R) , $(P_{mark})^2$ -Marking sent to upstream router, $(I_{id})^1$ – Interface ID retrieved					

Table 2 shows how the present marking information is reverted back to the original marking which has come into a router. From Table 1 and Table 2, it is clear that LEIP marking and traceback algorithms are reversible.

VI. Simulation Results and Performance Evaluation

To measure the effectiveness of the proposed LEIP algorithm, its performance is measured and compared with recent IP traceback schemes. We have considered Revised-MORE [30], HAHIT [26] and HIT [31] for comparison.

A Metrics for Performance Evaluation

In this section, we enumerate the performance metrics used for evaluation. These metrics help to appraise any traceback scheme against the design objectives presented in Section 3.2.

- Marking overhead
- Memory overhead
- Logging Overhead
- Number of packets needed to reconstruct the path (i.e) convergence time
- Traceback process overhead.
- Traceback process accuracy

To measure the performance of LEIP, we have simulated a network using BRITE topology generator under NS2 environment. Also, to simulate the internet topology, the topology distributed by CAIDA Skitter project [5] has been used as sample data set.

B Marking Overhead at Routers

Marking overhead is defined as the time taken by each router to calculate the marking information and time taken to embed the marking in IP ID field. This time also includes the time needed to search log/hash tables before marking. When there is an increase in size of marking information, the number of bits to be marked would increase and subsequently, the time for recording the marking information would also increase. The time for calculating the marking information has been calculated for both simulated environment and CAIDA data set. For the simulation, 3000 packets were sent from different sources which pass through six routers. For CAIDA, the number of routers considered was 14. This is because the average hop count of paths in the data set is 14.42. The time taken by LEIP, RevisedMORE, HAHIT and HIT for the simulated environment is shown in Table 3.

Table 3. Marking Overhead (Simulation Environment)

Time for marking information (ms)			
LEIP	Revised-MORE	HAHIT	HIT
50	73	110	84

As can be seen from Table 3, the time for marking in LEIP is least among all. RevisedMORE, HAHIT and HIT took large time marking due to the logging. As the routers in the simulated network had high degree, these schemes were supposed to do logging often. This increased their marking time.

Table 4. Marking Overhead (CAIDA Data set)

Time for marking information (s)			
LEIP	Revised-MORE	HAHIT	HIT
0.989	0.109	0.1	0.1

LEIP and all the schemes considered for comparison use 16-bits for marking. The marking overhead for the CAIDA data set has been calculated and shown in Table 4. The time LEIP takes for marking is found to be higher than other schemes. As the average degree of CAIDA data set is 2.63, no logging may be needed in all the other schemes. This reduced the time for marking in RevisedMORE, HAHIT and HIT.

C Memory Overhead

Memory overhead determines the memory required at every logging router to store the packets. A packet needs to be stored in a digest/hash or log table, when there is no space in it for recording the calculated marking at a router. Most of the traceback algorithms require logging of packets at the intermediate routers. In LEIP, the marking information to be recorded can be very well accommodated in the IP ID field and there is no need for logging the packets at all in any of the intermediate routers. Hence, LEIP causes no logging and storage overhead. To determine the amount of memory required at the logging routers in HAHIT, HIT and

 ISSN: 2454-4248
229 – 236

RevisedMORE, the logging analysis was conducted using CAIDA data set. With the average degree 2.63 (i.e. 3), it is found that the interlogging distance is 10 for Revised-MORE, which means every 11th router needs to log the packet. Both HAHIT and HIT require logging at 9th router. Table 5 shows the amount of memory required for a few number of packets at the logging routers for the topology took for investigation.

Table 5. Memory requirement at logging routers

Marking and Traceback Schemes	Memory required (in KB) (In terms of number of packets)		
	5000	3,75,000	20,00,000
LEIP	**Absolutely Nil**		
RevisedMORE	0.02	0.03	0.05
HAHIT	0.23	1.13	2.53
HIT (degree < threshold)	0.20	1.00	2.25
HIT (degree > threshold)	0.23	1.13	2.53

When logging is to be performed, a router may need to find the appropriate table or whether an entry for the packet has been already logged. LEIP relieves the routers greatly from these operations. Hence, marked packet can reach the destination quickly. When routers along the path have higher degree, the packets need to be logged quite often which increases the amount of memory requirements.

D Packet Logging Overhead

When a packet enters into a router through one of its interface, the router calculates the new marking information using the marking information in the incoming packet and the incoming interface. The router, then checks whether the calculated marking details can be placed in the IP ID field. If so, the router records the mark; otherwise it has to log the packet. For logging, Revised-MORE uses hash tables which are indexed with subnet addresses. The details of logging can be referred from [18]. But, the proposed system LEIP does not require doing logging as the calculated information would be very much less than the size of IP ID field.

The packet logging overhead of other hybrid traceback schemes have also been evaluated for a router-level topology from CAIDA [5]. For this topology, it is assumed that every packet flows through the interface whose ID is the highest among all the other interfaces. Under this scenario, RevisedMORE logs its packets at the 11th router, where as HAHIT and HIT logs at 9th routers. Even though, for the average hop count of 14, the logging is required at only one router, the amount of memory required at this router would very enormous, when a huge amount of packets move through it.

Simulation experiments using a router level topology ITDK0304 provided by CAIDA [6] has also been conducted. The topology has 192244 nodes and 636643 directed links. The average and maximum node degrees of this topology are 6.34 and 1071 respectively. For finding the logging frequency, we have considered two scenarios namely average case, wherein the packets are assumed that they flow through the interfaces with average ID and worst case, wherein the packets pass through the interfaces having largest ID among all the interfaces in each router. That is, the interfaces are assigned with maximum (*d-1*) and average (*(d-1)/2*) ID.

For the worst case scenario, LEIP requires no logging, whereas RevisedMORE and HAHIT require logging at 21.8%, and 24% of the routers respectively. The logging frequency of HIT is slightly lesser than HAHIT. For the packets assumed to come through (d-1)/2th interface, the logging frequency is 20.1%, 21.8% and 21.2% for RevisedMORE, HAHIT and HIT respectively. In this case too, a LEIP enabled router does not require logging.

E Traceback Process Overhead at Routers

IP Traceback involves querying the routers from the destination to the source. The number of routers queried determines the overhead of traceback process. To reconstruct the attack path from the destination to the source, the traceback process iteratively queries the routers starting last hop router.

As LEIP solely depends on the cumulative path information embedded in the marking field, a router has to apply only exponentiation operations as given in Equation (2) and Equation (3). Applying Equation (2) and Equation (3) helps to find the incoming interface and the marking from the upstream router at any router. It does not require consulting any log or hash tables as in other traceback schemes. This significantly reduces the time for traceback process.

RevisedMORE, HAHIT and HIT require finding appropriate log tables to retrieve the incoming interface and the marking information from the upstream routers. All the routers, which logged the packets during marking, will have to be queried to find this information. A RevisedMORE enabled router examines a packet and if it finds logging field is set to 1, it understands that it has logged packets. Then, it uses the subset address and TTL in the traceback requested packet to find the hashed entry to find the incoming interface and the marking information from the upstream router. In HAHIT and HIT, the time difference between the process of marking and tracking back is used to find the appropriate hash table. If not done carefully, the search may lead to querying wrong tables.

The time involved in the traceback process has been calculated for the simulated environment with 3000 packets and presented in Table 6.

Table 6. Traceback Overhead (Simulation Environment)

Time for marking information (ms)			
LEIP	Revised-MORE	HAHIT	HIT
79	53	110	74

The time taken for traceback process in case of RevisedMORE is lesser when compared to all the other schemes. This is so because it maintains a log table for each of the router's interface. So, there is no need to search all the tables to find the appropriate table which has the required entry. LEIP takes time higher than RevisedMORE and HIT. This is due to the exponentiation operation involved in the process of traceback.

 ISSN: 2454-4248
229 – 236

F Traceback Process Accuracy

A traceback scheme is said to be robust when it has high traceback accuracy. Traceback process accuracy depends on the number of correct/false nodes grafted on the attack path. A system is said to be robust only when it yields low false positives/negatives. Traceback accuracy increases when the false positives/negatives decrease. During traceback process, the queried router may return false nodes, thus leading to wrong path construction. Wrong path construction is mainly due the usage of log/hash tables. The log/hash tables have to keep the logged details for a long time for path reconstruction. Refreshing of log /hash tables should be constrained. Otherwise, refreshing leads to construct wrong path. Traceback schemes that use log/hash tables have the prospect of introducing false positives/negatives.

In LEIP, all the essential information needed for path reconstruction at any router is available in the marking information that it receives from the downstream router itself. LEIP never consults log tables. LEIP is designed in such a way that cumulative effect made by marking algorithm can be relapsed back to the initial value by traceback algorithm. As no requirement is posed for the use of log tables during marking, no tables are consulted during traceback. Hence, LEIP guarantees zero false positives/negatives.

The simulation experiments have been conducted to find the traceback accuracy of RevisedMORE, HAHIT and HIT. HIT and HAHIT claim that they have low storage requirements and routers can keep the path information for a long time and therefore do not need to refresh their log tables under flood attacks, hence zero false negatives [27]. But this is true only when the elapsed time between the marking of a packet and its traceback is well within the reasonable limit. Otherwise, unnecessary search may be caused. RevisedMORE uses TTL for traceback; hence there is no false positive.

G Convergence Time

Convergence time is the measure that evaluates how many packets are needed at the victim to trace an attacker. The more the number of packets, higher will be the time for the reconstruction of path. Generally, the traceback methods depending on more number of packets are time consuming. LEIP and the schemes considered for comparison namely, RevisedMORE, HAHIT and HIT require just one packet for path reconstruction.

H Accomplishment of Design Goals

The list of features for an IP traceback schemes has been highlighted in Section 3.2. Here, we analyze how the metrics used for performance of evaluation help LEIP accomplishing the design goals. The logarithmic based marking algorithm helps to embed the complete path information in a packet. Thus, one packet is enough to build the traversed path. This enables to determine both flooding and single packet attacks. Through a numerical example, it was demonstrated how the traceback process identifies the first hop router in Section 5. As the marking information is very well fit into the IP ID field, no packet logging and memory overhead is incurred by LEIP. Since the use of log tables has been completely avoided, there are no false positives/negatives in LEIP, thus leading to 100% traceback accuracy. The only issue with LEIP is that it requires slightly higher time for marking and tracking back than some of the traceback approaches. This increase in time is meager when frequency of logging is considered in those approaches. Table 7 provides a detailed qualitative comparison of LEIP, Revised-MORE, HAHIT and HIT.

7 Conclusion and Future Work

In this article, we address the problem of identifying the true origin of these attacks packets. The article has reviewed the recent works on the traceback problem and identified their limitations. Based on the understanding from the recent works, the article proposed a new marking and traceback scheme. The proposed scheme, LEIP, has used logarithmic operation for marking and exponentiation for traceback. As one operation complements the other, the traceback has been successful. The markings are made to create a collective effect of the all the marking information at the routers. The collective marking information at the destination can be reverted back to the initial value of the marking information using exponentiation operations. LEIP demands no logging at any of the routers.

As no logging is performed, the overhead involved in the traceback process is completely avoided. This leads to zero false positives/negatives. But, all the recent works on the traceback problem demand a considerable amount of memory at the intermediate routers for logging. This distinguishes LEIP from all the other traceback systems.

Source-end defenses [14],[15] may be instituted to prevent the packets from the identified source from further entering into the network. This would further increase the strength of a detection system.

It is found from experiments that LEIP consumes time little bit higher than other schemes while marking and tracking back. This is because of the logarithmic and exponentiation operations performed by LEIP. LEIP may be further analyzed to optimize these operations, so that the marking and trace back time can be reduced. We are working on this aspect as a next step towards finding a scheme with low marking and tracking time.

References

[1] Al-Duwari B., and Govindarasu M., " Novel hybrid schemes employing packet marking and logging for IP traceback", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 17, pp. 403-418, 2006.

[2] Anderson T. et.al. "Practical network for IP traceback", *IEEE/ACM Transactions on Networking*, Vol. 9, pp. 226-237, 2001.

[3] Ansari N., and Belenky A.," IP traceback with Deterministic Packet Marking", *IEEE Communications Letter*, Vol.7, pp.162–164, 2003

[4] Bellovin S.M., "ICMP Traceback Messages", Network Working Group Internet Draft, 2000.

 ISSN: 2454-4248
229 – 236

[5] CAIDA: CAIDA's skitter project, http://www.caida.org/tools/skitter/.

[6] CAIDA : Internet Topology Data Kit (ITDK), http://caida.org/tools/measurement/skitter/idkdata.xml, 2003

[7] Choi K.H., and Dai K.H., " A Marking Scheme using Huffman Codes for IP Traceback" Proc. of the 7th International Symposium on Parallel Architectures, Algorithms and Networks, Hong Kong, 2004.

[8] Claffy K., and McCreary S., "Trends in wide area IP traffic patterns: A view from Ames Internet exchange", Proc. of ITC Specialist Seminar on IP Traffic Modeling, Measurement and Management, 2000

[9] Dean D. et.al.,"An algebraic approach to IP traceback", *ACM Transaction on Information and System Security*, Vol.5, pp.119–137, 2002

[10] Gong C., and Sarac K., "IP Traceback based on Packet Marking and Logging", Proc. of IEEE Conference on Communications (ICC), Seoul, Korea, 2005.

[11] Gong C., and Sarac K**., "**A More Practical Approach for Single-Packet IP Traceback using Packet Logging and Marking", *IEEE Transactions on Parallel and Distributed Systems*, Vol.**19,** No. 10, pp. 1310-1324,2008**.**

[12] Jain R., and Meshram A., " A Survey on Packet Marking and Logging", *International Journal of Computer Science and Information Technologies*, Vol. 4, No. 3, pp. 426-429, 2013.

[13] Lakshmi S., Anup Kumar, and Agrawal D. P., " Taxonomy of IP Traceback", *Journal of Information Assurance and Security*, Vol. 1, pp. 79-94, 2006.

[14] Malliga S., and Tamilarasi A., " An Autonomous Framework for Early Detection of Spoofed Flooding Attacks", *International Journal of Network Security*. Vol. 10, pp. 39-50, 2008.

[15] Malliga S., Tamilarasi, A., and Janani, M., "Filtering spoofed traffic at source end for defending against DoS/DDoS attacks" , Proc. of International Conference on Computing, Communication Networking", pp. 1-5, 2008.

[16] Malliga S., and Tamilarasi A., " A proposal for new marking scheme with its performance evaluation for IP traceback", *WSEAS Transactions on Computer Research*, Vol. 3, No. 4, pp. 259-272, 2008.

[17] Malliga S., and Tamilarasi A., " A hybrid scheme using packet marking and logging for IP traceback" International Journal of Internet Protocol Technology, Vol. 5, pp. 81-91, 2010.

[18] Malliga S., Kanimozhi Selvi C.S., and Kogilavani S.V., "A low storage and traceback overhead system for IP traceback", Accepted for publication in *Journal of Information Science and Engineering*, 2015.

[19] Mandia K., P and rosise C., *Incident Response: Investigating Computer Crime Berkeley: Osborne*, McGraw-Hill. 2001.

[20] Mankin A. et.al., "On Design and Evaluation of Intention-Driven ICMP Traceback", Proc. of the IEEE International Conference on Computer Communication and Networks", pp. 159-165,2001.

[21] Microsoft Corporation, *Stop 0A in tcpip.sys when receiving out of band (OOB) data*. http://support.microsoft.com/support/kb/articles/Q143/4/78.asp, 2000

[22] Perrig A., and Song D.X., " Advanced and Authenticated marking scheme for IP traceback ", Proc. of 20th Annual Conference of IEEE Communications and Computer Societies, 2001.

[23] Dean D. et.al., "An algebraic approach to IP traceback", *ACM Transaction on Information and System Security*, Vol. 5, pp. 119–137, 2002

[24] Snoren A.C. et.al., " Single-packet IP Traceback", *IEEE/ACM Transactions on Networking*, Vol. 10, pp. 721-734, 2002.

[25] Vijayalakshmi M., Shalinie M., and Nithya N.," A Brief Survey of IP Traceback Methodologies", *Acta Polytechnica Hungarica,* Vol. 11, No. 9, pp. 197 –216, 2014.

[26] Vijayalakshmi M., and Shalinie M., " Single Packet ICMP Traceback Technique using Router Interface", *Journal of Information Science and Engineering*, Vol. 30, pp. 1673-1694, 2014.

[27] Wang B., and Schulzrinne H., "A Denial-of-Service-Resistant IP Traceback Approach", Proc. of the IEEE 9th International symposium on Computers and Communication", Vol.1, pp. 351- 356, 2004.

[28] Yan D. et.al., " A Precise and Practical IP Traceback Technique Based on Packet Marking and Logging", *Journal of Information Science and Engineering*, Vol. 28, pp. 453-470, 2012.

[29] Yang M.H., and Yang M.C., "RIHT: a novel hybrid IP traceback scheme", *IEEE Transactions on Information Forensics and Security*, Vol. 7, pp. 789–797, 2012.

[30] Yang M.H., "Hybrid Single-Packet IP Traceback with Low Storage and High Accuracy", *The scientific world journal*, Article ID 239280, 2014.

[31] Yang M.H., "Storage-Efficient 16-Bit Hybrid IP Traceback with Single Packet", *The Scientific World Journal*, Article ID 659894, 2014.