A Low Traceback and Zero Logging Overhead IP Traceback Approach for Communication Networks

S. Malliga
Department of CSE
Kongu Engineering College
Erode, India
mallinishanth72@gmail.com

S.V. Kogilavani
Department of CSE
Kongu Engineering College
Erode, India
kogilavani@kongu.ac.in

P.S.Nandhini
Department of CSE
Kongu Engineering College
Erode, India
nandhini.cse@kongu.ac.in

Abstract—In an IP address spoofing attack, attackers send IP packets from a forged source address in order to camouflage themselves. Denial of Service attacks quite often employ IP spoofing to overwhelm a target with packets that appear to have come from legitimate IP addresses. Such attacks may be prevented by tracing these attacks back to their origin. IP traceback is a technique which plays a vital role in finding the source of spoofed packets. This paper reviews an ICMP traceback method, SPITRI and suggests a few changes in the way the packets are marked and tracked back. The proposed marking scheme reduces the number of clock cycles needed for marking and tracking back. Also, it does not require logging at any of the routers. The simulation results demonstrate that the refinements reduce the time for marking and tracing back with 100% accuracy.

Keywords- IP Spoofing, DoS/DDoS, IP Traceback, Packet Marking, Packet Logging, Traceback Accuracy

I. INTRODUCTION

Spoofing is the creation of TCP/IP packets with forged source IP address. Routers use the destination IP address in order to forward packets through the Internet, but ignore the source address and it is never authenticated. This motivates attackers to exploit spoofing for launching Denial of Service (DoS) or Distributed DoS (DDoS) attack. A DoS/DDoS attack is characterized by an explicit attempt by attackers to thwart legitimate users of a service from using that service. Since DoS/DDoS attackers employ spoofing, it is very difficult to find the source of such attacks and defend against them.

CERT Coordination Center defines three basic motivations of DoS/DDoS attacks [1] namely: consumption of resources, destruction of configuration information and physical destruction or alteration of network components. Of these forms, common intention of the DoS/DDoS attackers is to consume scarce resources like CPU, memory, disk space etc.

There are two classes of DoS/DDoS attacks namely: flooding attacks and software exploits [2]. These attacks need not to always flood a victim. Yet, a single well focused attack packet can dethrone a target system [3]. Tremendous efforts have been paid by researchers to address these attacks. One such effort is IP traceback technique. It is a technique for identifying the true origin of packet and setting up protection mechanisms to prevent spoofing attacks. IP traceback is used for identifying both flooding and single packet attacks. IP traceback techniques can be broadly classified into two types: in-band and out-of-band approaches [4]. In-band approaches use IP packets to enable traceback and out-of-band approaches use a separate trace packet like ICMP packet to do tracing. Packet marking and logging exploits [2,5,6] are two sub-types of in-band approaches. Packet marking requires the routers along the path to mark their identification information in the packets they forward. These markings are done either probabilistically (PPM) [7,8,9,10] or deterministically (DPM) [11,12]. Such marked packets are then used to reconstruct the path traversed by the packets. Packet logging let the routers store the packets they forward and these logged packets are used for reconstruction of the path during traceback [16].Hybridizing packet marking and logging provides the benefits of both. Several hybrid methods have also been recommended [5,13,14,15,17,18,19,20,21]. 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. Out-of-band approaches [22,23,24,25] like iTrace, Intension-driven ICMP, iCaddie, and SPITRI generate separate ICMP packets to record the path information and these ICMP packets are then used to traceback the origin of the attack packets.This paper reviews the present methods for IP traceback and identifies the overhead incurred in marking and tracking back by latest ICMP traceback approach, SPITRI. In our work, a few modifications are suggested in SPITRI to improve its performance and it is extended for IP traceback.

The rest of the article is orchestrated as follows: Section 2 presents the requirements of an ideal traceback system and reviews the traditional and state-of-the art techniques for traceback. SPITRI is briefly explored in Section 3. The processing overhead of SPITRI is also discussed in this section. Section 4 proposes a new system, ESPITRI (Enhanced SPITRI) that extends SPITRI with a few modifications for IP traceback. The marking and traceback process is also demonstrated with a numerical example in the section. The performance of the proposed system is evaluated with different metrics and the results are presented in Section 5. Finally, in Section 6 we present the conclusion of the proposed work and further research directions.

II. SINGLE PACKET ICMP TRACEBACK TECHNIQUE USING ROUTER INTERFACE (SPITRI)

SPITRI [24] is an ICMP traceback scheme. At first, in SPITRI, the egress router generates an ICMP packet with the probability ‘P’ and sends it towards the destination. While forwarding the packets, the intermediate routers update the path information in the received ICMP traceback

packet using the ID assigned to the incoming link. Equation (1) is used to update the path information.

$$Pathinfo_{new} = \frac{1}{Pathinfo + 2} + ID + 1 \quad (1)$$

In Equation (1), *'$Pathinfo_{new}$'* is the path information to be recorded at a router, 'P*athinfo*' is the path information received from the upstream router. *'ID'* is number assigned to the incoming link. This formula was derived in such a way that path information value has a cumulative effect of all the incoming interfaces through which the packet traversed to reach the destination. Hence, during traceback, the complete list of upstream Interface IDs to reach the attacker could be retrieved from the '*Pathinfo*'. During traceback, every router uses Equation (2) and Equation (3) to find the incoming link and the path information it received from its upstream router.

$$ID = floor(Pathinfo) - 1 \quad (2)$$

$$Pathinfo_{old} = \frac{1}{Pathinfo - floor(Pathinfo)} - 2 \quad (3)$$

In Equation (2), *'$Pathinfo_{old}$'* is the path information retrieved from the upstream router during the marking process. These equations are repeatedly applied by routers to find their upstream routers. To stop the traceback process, TTL is used. From Equations (1), (2) and (3), we can understand that these equations perform a few arithmetic operations like adding and subtracting two, adding and subtracting one. While analyzing SPITRI with some numerical examples, it is found that these arithmetic operations are highly redundant. The number of clock cycles required for integer addition and subtraction depends on the processor. For instance, Intel core 2 duo processor expends one clock cycle for addition and subtraction. Even though, addition and subtraction require a one clock cycle, when a router marks more number of packets, it has to spend huge clock cycles. This causes additional overhead on the routers. Also, the size of path information element in the ICMP traceback packet is at least 512 bits. The traced packet content element of the traceback packet generally holds the content of the traced (original) packet and contains at least the 20 byte IP header of the traced packet. This is used to correlate the ICMP packet with the original packet being traced. It is found that the path information of size 512 bits is also redundant. A maximum of 16 bits path information is sufficient to enable traceback. Another limitation is that it can trace only the flooding attacks. Since the ICMP packet is sent with a probability, it could not trace a single attack packet. These limitations are addressed in the proposed approach.

III. ENHANCED SPITRI (ESPITRI)

This article proposes a few modifications in Equations (1), (2) and (3). These modifications definitely eliminate the additional overhead incurred by the redundant arithmetic operations. Instead of marking ICMP packet, the proposed work marks IP packets. And, it marks only 16 bit path information, which greatly reduces the marking overhead. ESPITRI also enables single packet traceback.

A. Marking Procedure

ESPITRI lets the router mark each packet that flows through it, Here again, the markings made by the routers give the cumulative effect of the complete path traversed by the packets. To mark each packet, the routers use the incoming link or interface through which the packets enter into the routers. Every router keeps a table that maps the hardware address of each its link to a unique *ID* from 0 onwards. When a router marks, it finds the ID associated with the incoming link and marks the packet. As, it is understood that the addition operations in the marking procedure of SPITRI are redundant, ESPITRI eliminates these operations. Albeit, this addition requires one clock cycles on most of the processor, when large number of packets are marked, the number of clock cycles would be considerably very high. The modification in Equation (1) to reduce the clock cycle is given Equation (4).

$$Markinfo_{new} = \frac{1}{Markinfo} + ID \quad (4)$$

In Equation (4), *'Markinfo'* is the marking information that arrives at a router, say R. *'$Markinfo_{new}$'* is the newly computed marking information by R using *'Markinfo'*. For marking, ESPITRI uses 16 bits IP ID field in the IP packet, which is used to identify the frame and any associated fragments for reassembly. While this field is used for marking, having non-zero values in flags and offset fields causes mystification at the destination. For this reason, only IP ID field is used for marking. As only 0.25% of internet traffic is fragmented [26], many packet marking schemes uses this field.

The IP packet header containing the marking field is shown in Fig. 1.

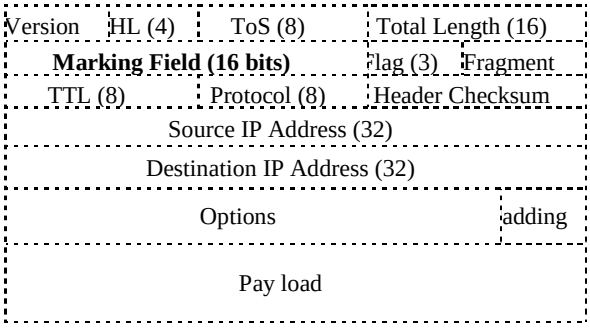


Fig. 1. Format of IP packet with marking field

Path Reconstruction Algorithm

The traceback or path reconstruction process is brought into play when the victim detects that the received packet is an attack packet. The victim now uses the marking or path information in the packet to trace the path travelled by the packet. Since the markings are made in such a way that they create a cumulative effect of the path, they are used to enable traceback. First, the victim sends the packet to its first hop router. Upon receiving the traceback requested packet, the router finds the incoming interface that lets the packet that is being tracked, into it using the marking

information received from the victim and also finds marking it received from its upstream router when the packet was forwarded through it earlier. After identifying these two details, the router forwards the traceback request to its upstream router using the identified interface. To determine the incoming link and marking information from the upstream router, every router uses Equations (5) and (6).

$$ID = floor(Markinfo) \quad (5)$$

$$Markinfo_{new} = \frac{1}{Markinfo - ID} \quad (6)$$

Every router first applies Equation (4) on the path information in the traceback requested packet to find the incoming interface of the packet that caused traceback and then Equation (5) to compute the marking received from its upstream router during the marking phase.

C. Realizing the power of Marking and Traceback algorithm

To show that marking and traceback processes are reversible, we have taken CAIDA's dataset IDTK [27]. The details of the dataset is outlined in Section 5.1. The dataset contains enumerous number of links. For the purpose of demonstraiton, we have taken a link which connects ten routers and each router with variable number of interfaces. Table 1 exemplifies the marking information computed by the routers along a link in the dataset.

TABLE I. ILLUSTRATION OF MARKING PROCESS

Router ID	Number of interfaces in the router	Interface through the packets come in (average case)	Marking information	
			Received	Computed
N169668	175	87	87	87.01
N10994	84	42	87.01	42.01
N105646	44	22	42.01	22.02
N172198	568	284	22.02	284.05
N166846	38	19	284.05	19.01
N168844	140	70	19.01	70.05
N165929	19	9	70.05	9.01
N171697	173	86	9.01	86.11
N169207	154	77	86.11	77.01
N245531	234	117	77.01	117.01

The traceback process for the above marking is depicted in Table 2.

TABLE II. ILLUSTRATION OF TRACEBACK PROCESS

Router ID	No. of interfaces in the router	Interface through the packets come in (average case)	Marking information		Interface ID identified
			Received	Computed	
N245531	234	117	117.01	77.01	117
N169207	154	77	77.01	86.11	77
N171697	173	86	86.11	9.01	86
N165929	19	9	9.01	70.05	9
N168844	140	70	70.05	19.01	70
N166846	38	19	19.01	284.05	19
N172198	568	284	284.05	22.02	284
N105646	44	22	22.02	42.01	22
N10994	84	42	42.01	87.01	42
N169668	175	87	87.01	87	87

IV. EXPERIMENTS AND RESULTS

To understand the performance, we test ESPITRI on a simulated network and CAIDA's dataset. Also, we compare the proposed ESPITRI with Revised MORE [15], HIT [20] and SPITRI [24]. As it has been demonstrated that Revised MORE outperforms PPIT [21], MORE [14] and HAHIT [19], we take Revised More alone for comparison. And, since ESPITRI is extended from SPITRI, we also intend to compare ESPITRI with SPITRI.

A. Details of Experiments

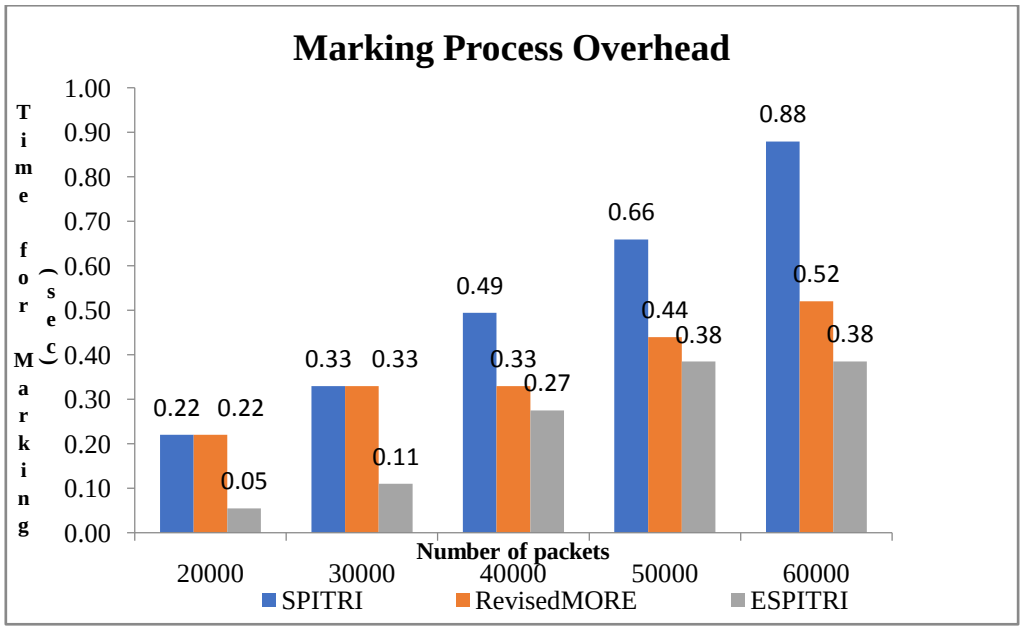
To evaluate and compare the performance of ESPITRI with other trace back schemes, we use a network simulated using NS2.

B. Marking Overhead on Routers

This metric helps in finding the amount of time a router is expected to sacrifice for marking. As the router's principal responsibility is to forward the packets, marking process would increase the burden on the router in case of marking. We have evaluated the time taken for marking by the routers in the simulated network. The network has gradually increased number of packets sent by the source node and these packets were flowing through eleven routers in the simulated network. The time taken by these eleven routers for marking the packets has been presented in Fig. 2.

From Fig. 2, it is clear that the marking overhead incurred by the proposed ESPITRI scheme is less when compared to SPITRI and RevisedMORE. The performance of ESPITRI is also evaluated using CAIDA's IDTK dataset. For experimental purposes, we have taken a set of links from CAIDA's IDTK and packets were allowed to pass through these links. The number hops along the paths were varying. The packets passing through the links were marked by the routers along the path using ESPITRI, SPITRI and

RevisedMORE. The marking information recorded by these approaches is presented in Table 3.



.

Fig. 2. Marking Process Overhead

TABLE III. SIZE OF MARKING INFORMATION

Links	Nodes	No. of interfaces	Interface passing the packets	Marking Information		
				ESPITRI	SPITRI	RevisedMORE
L1	N169668	175	87	87.01	88.50	87
	N10994	84	42	42.01	43.01	7350
	N105646	44	22	22.02	23.02	22*
	N172198	568	284	284.05	285.04	12780
	N166846	38	19	19.00	20.00	19*
	N168844	140	70	70.05	71.05	2730
	N165929	19	9	9.01	10.01	51879
	N171697	173	86	86.11	87.08	86*
	N169207	154	77	77.01	78.01	13321
	N245531	234	117	117.01	118.01	117*
L5	N169668	175	87	87.01	88.50	87
	N108036	74	37	37.01	38.01	6475
	N245531	234	117	117.03	118.02	117*

*- logging is done

From Table 3, it is clear that ESPITRI and SPITRI require no logging. Whereas, RevisedMORE requires logging at four routers for link 1 and at one router for link 5.

C. Logging Overhead

It is the time taken for storing the digest of the packets when a router finds no space for marking in the packet. Since ESPITRI marks in such a way that it always accommodates the marking information in the 16-bit field, it never demands logging at the routers. SPITRI embeds its marking information of size 512 bytes in ICMP packet. SPITRI avoids logging by copying the first few bytes of the packet being marked in the Traced Packet Content field of SPITRI message. The number of routers required logging has been calculated for ITDK dataset and presented in Fig. 3.

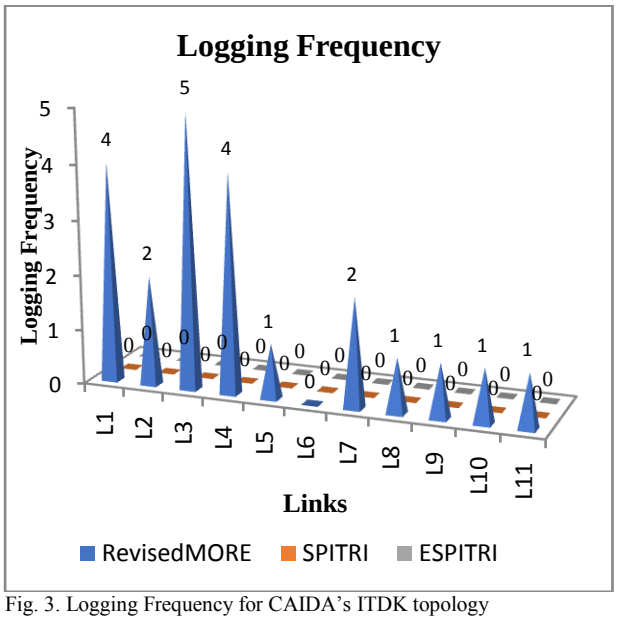


Fig. 3. Logging Frequency for CAIDA's ITDK topology

From Fig. 3, it is apparent that only RevisedMORE demands logging, the other two schemes do not log the packets at any of the intermediate routers. Apart from logging and marking overhead, experiments were also conducted to find the overhead in traceback process, storage overhead, traceback accuracy and the results are presented in Table 5.

D. ESPITRI vs SPITRI

As EPSITRI is an enhancement over SPITRI, we specifically intend to compare ESPITRI and SPITRI in this section. The enhancements of ESPITRI over SPITRI are already discussed. To sum up, Table 5 provides a comparison of ESPITRI and SPITRI.

TABLE V. COMPARISON OF ESPITRI AND SPITRI

Parameters/Metrics	ESPITRI	SPITRI
Traceback technique	In-band approach	Out-of-band approach
Packet used for Marking	IP	ICMP
Marking overhead	Very low	Low
Size of the marking information	16 bits	512 bits
Traceback process overhead	Very low	Low
Traceback accuracy	100%	100%
Storage overhead	Nil	Nil
Number of packets for traceback	1	1
Reaction to an attack packet	Immediate	Deferred till the arrival of marked ICMP packet (This may lead to flooding at the victim)

From Table 5, we can understand that the extensions introduced in ESPITRI leads to an IP traceback system with very low marking and traceback overhead. Also, ESPITRI demands every router to mark 16-bit information, whereas the size of the marking information recorded by SPITRI is 512 bits. Another significant benefit of the ESPITRI is its immediate attention to the attack. Once a packet is detected as an attack packet, the marking information in the attack packet is used to kick off the traceback process. In SPITRI, Traced Packet Content field of the ICMP packet contains the 20 byte of the IP header of the IP packet to be traced. This is done to associate the ICMP to the IP packet being traced. If an IP packet is detected as an attack packet, then the corresponding ICMP packet has to arrive at the destination and is used for traceback. The ICMP packets used for marking are generated with some probability. This means that not every IP packet could be traced with the help of an ICMP packet. Even if it is argued that one ICMP packet is enough to trace the all packets coming from same source, only after the arrival of the marked ICMP packet, the traceback process can start. In the intervening time, more number of packets may arrive and flood the victim.

V. CONCLUSION AND FUTURE WORK

This article has given a careful thought to the problems of DoS/DDoS attacks and presented a solution that detects the source of these attacks. The present hybrid packet marking and logging approaches are reviewed and their limitations are also addressed in this paper. SPITRI, an ICMP traceback system, forms the source of the proposed ESPITRI. SPITRI has been carefully analyzed and its issues have been identified. The solutions to these issues have led to the development of ESPITRI.

ESPITRI eliminates the redundant arithmetical operations performed by SPITRI. This reduces the overhead incurred by ESPITRI during marking and traceback process. Since every IP packet is marked, once an IP packet is detected as an attack packet, the traceback process is initiated by ESPITRI. Simulation results demonstrate that none of the intermediate routers require logging of the marked packet. This completely relieves ESPITRI from the logging overhead and thus eliminating storage requirements at the router. As no router is consulted during traceback, ESPITRI provides 100% traceback accuracy.

Once the first hop router of the source is identified, source-end defensive schemes [29] may be employed to prevent the DoS/DDoS attacks at the source itself. Hence, we plan to integrate a source-end defense in the proposed system as a future research work. In addition to that, even though the marking and traceback overhead is very low compared to other schemes, we look forward to reduce it further.

REFERENCES

[1] M. Abliz. “Internet Denial of Service Attacks and Defense Mechanisms”, Department of Computer Science, University of Pittsburgh, Technical Report, No. TR-11-178, 2011.

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

[3] http://support.microsoft.com/support/kb/articles/Q143/4/78.asp.

[4] L. Santhanam, Anup Kumar and D.P. Agrawal. “Taxonomy of IP Traceback”. Journal of Information Assurance and Security, 2006, No. 1, pp. 79-94.

[5] K. Singh, S. Koppu, and V. Madhu Viswanathan. “ E-RIHT: Enhanced Hybrid IP Traceback Scheme with 16-bit marking field”. International Journal of Engineering and Technology, 2013,Vol. 5, No. 3, pp. 2594-2600.

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

[7] T. Anderson, A. Karlin and S. Savage and D. Wetherall. “Practical network for IP traceback”. IEEE/ACM Transactions on Networking, 2001,Vol. 9, No. 3, pp. 226-237.

[8] A. Perrig and D.X. Song. “Advanced and Authenticated marking scheme for IP traceback”. In 20th Annual Conf. IEEE Communications and computer Societies, Ancharage, Alaska, 2001, pp. 878–886.

[9] Dean, M. Franklin and A. Stubblefield. “An algebraic approach to IP traceback”. ACM Transaction on Information and System Security, 2002, Vol. 5, No. 2, pp. 119–137.

[10] Perrig, D. Song and A. Yaar. “FIT: Fast Internet traceback”. In IEEE INFOCOM, Miami, FL, USA, 2005, pp. 13-17.

[11] N. Ansari and A. Belenky. “IP traceback with Deterministic Packet Marking”. IEEE Communications Letter, 2003, Vol. 7, No. 4, pp. 162–164.

[12] K.H. Choi and K.H. Dai. “A Marking Scheme using Huffman Codes for IP Traceback”. In 7th Int. Symp. on Parallel Architectures, Algorithms and Networks, Hong Kong, 2004, May 10-12, pp. 421-428.

[13] S. Malliga and A. Tamilarasi. “A proposal for new marking scheme with its performance evaluation for IP traceback”. WSEAS Transactions on Computer Research, 2008, Vol. 3, No. 4, pp. 259-27.

[14] S. Malliga and A. Tamilarasi. “A hybrid scheme using packet marking and logging for IP traceback”. International Journal of Internet Protocol Technology, 2010, Vol. 5, No. 1/2, pp. 81-91.

[15] S. Malliga, C.S. Kanimozhi Selvi and S.V. Kogilavani.” A low storage and traceback overhead system for IP traceback”. Accepted for publication in Journal of Information Science and Engineering, published by the Institute of Information Science, Academia Sinica, Taipei, Taiwan

[16] A.C. Snoren, C. Partridge, L.A. Sanchez, C.E. Jones, F. Tchakountio, B. Schwartz, S.T. Kent and W.T. Strayer. „Single-packet IP Traceback“. IEEE/ACM Transactions on Networking, 2002, Vol. 10, No. 6, pp. 721-734.
[17] B. Al-Duwari amd M. Govindarasu. “Novel hybrid schemes employing packet marking and logging for IP traceback”. IEEE Transactions on Parallel and Distributed Systems, 2006, Vol. 17, No. 5, pp. 403-418.
[18] M.H. Yang and M.C. Yang. “RIHT: a novel hybrid IP traceback scheme”. IEEE Transactions on Information Forensics and Security,
[19] 2012, Vol. 7, No. 2, pp. 789–797.
[20] M.H. Yang. “Hybrid Single-Packet IP Traceback with Low Storage and High Accuracy”. The scientific world journal, 2014, 12:Article ID 239280, pp. 1-12.
[21] M.H. Yang. “Storage-Efficient 16-Bit Hybrid IP Traceback with Single Packet”. The Scientific World Journal, 2014, 12: Article ID 659894, pp. 1 -11.
[22] D. Yan, Y. Wang, S. Su and F. Yang. “A Precise and Practical IP Traceback Technique Based on Packet Marking and Logging”. Journal of Information Science and Engineering, 2012, Vol. 28, No. 3, pp. 453-470.
[23] Mankin, D. Massey, C. Wu, S. Felix Wu and L. Zhang. “On Design and Evaluation of Intention-Driven ICMP Traceback”. In IEEE
[24] International Conference on Computer Communication and Networks, Scottsdale, AZ , October 15- 17,2001, pp. 159-165.
[25] B. Wang and H. Schulzrinne. “A Denial-of-Service-Resistant IP Traceback Approach”. In 9th IEEE International Symposium on
[26] Computers and Communication, Alexandria, Egypt, 2004, pp. 351-356.
[27] M. Vijayalakshmi, and M. Shalinie. „Single Packet ICMP Traceback Technique using Router Interface“. Journal of Information Science and Engineering, 2014, Vol. 30, No. 6, pp. 1673-1694.
[28] S.M. Bellovin. „ICMP Traceback Messages“. Network Working Group Internet Draft, 2000.
[29] K. Claffy and S. McCreary. „Trends in wide area IP traffic patterns: A view from Ames Internet exchange“. In ITC Specialist Seminar on IP
[30] Traffic Modeling, Measurement and Management, Monterey, CA, USA, 2000.
[31] “CAIDA: CAIDA’s skitter project,” https://topodata.caida.org/ITDK /ITDK-2013-07/
[32] “CAIDA : The Macroscopic Internet Topology Data Kit (ITDK),” http://caida.org/tools/measurement/skitter/idkdata.xml, 2003
[33] S. Malliga and A. Tamilarasi. « Autonomous Framework for Early Detection of Spoofed Flooding Attacks”. International Journal of Network Security, 2008, Vol. 10, No. 1, pp. 39-50.