Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 3, March 2014, pg.79 – 88

RESEARCH ARTICLE

A NEW IP TRACEBACK SCHEME TO AVOID LAUNCH ATTACKS

E.JANSI*[1]
M.Tech Student
Department of Computer Science and Engineering
PRIST University Pondicherry, India.
jansibtech15@gmail.com

BHARATHI.R*[2]
Assistant professor
Department of Computer Science and Engineering
PRIST University Pondicherry, India.
prist2009cse@gmail.com

E.PUSHPARAJ*[3]
Assistant Professor
Ranganathan Engineering College,
Coimbatore

ABSTRACT

The Internet has been widely applied in various fields; more and more network security issues emerge and catch people's attention. However, adversaries often hide themselves by spoofing their own IP addresses and then launch attacks. For this reason, researchers have proposed a lot of trace back schemes to trace the source of these attacks. Some use only one packet in their packet logging schemes to achieve IP tracking. Others combine packet marking with packet logging and therefore create hybrid IP trace back schemes demanding less storage but requiring a longer search. In this paper, we propose a new hybrid IP trace back scheme with efficient packet logging aiming to have a fixed storage requirement for each router (under 320 KB, according to CAIDA's skitter data set) in packet logging without the need to refresh the logged tracking information and to achieve zero false positive and false negative rates in attack-path reconstruction. In addition, we use a packet's marking field to censor attack traffic on its upstream routers. Lastly, we simulate and analyze our scheme, in comparison with other related research, in the following aspects: storage requirement, computation, and accuracy.

KEYWORDS: New hybrid IP trace back, CAIDA's, packet logging, packet marking

I. INTRODUCTION

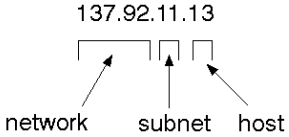
Most of current single packets trace back schemes tend to log packets' information on routers. Most current tracing schemes that are designed for software exploits can be categorized into three groups: single packet, packet logging and hybrid IP

trace back. The basic idea of packet logging is to log a packet's information on routers. The methods used in the existing systems include Huffman Code, modulo/ Reverse modulo Technique (MRT) and modulo/Reverse modulo (MORE). These methods use interface numbers of routers, instead of partial IP or link information, to mark a packet's route information. Each of these methods marks routers' interface numbers on a packet's IP header along a route. However, a packet's IP header has rather limited space for marking and therefore cannot always afford to record the full route information. So, they integrate packet logging into their marking schemes by allowing a packet's marking field temporarily logged on routers. From this, it is found that these tracing methods still require high storage on logged routers. Apart from this, also found that, exhaustive searching is quite inefficient in path reconstruction. Adversaries often hide themselves by spoofing their own IP addresses and then launch attacks. There is a lot of trace back schemes to trace the source of these attacks. Some use only one packet in their packet logging schemes to achieve IP tracking. Others combine packet marking with packet logging and therefore create hybrid IP trace back schemes demanding less storage but requiring a longer search. a new hybrid IP trace back scheme with efficient packet logging aiming to have a fixed storage requirement for each router (under 320 KB, according to CAIDA's skitter data set) in packet logging without the need to refresh the logged tracking information and to achieve zero false positive and false negative rates in attack-path reconstruction. In packet logging, the IP packet is logged at each router through which it passes.

Historically, packet logging was thought to be impractical because of enormous storage space for packet logs. Hash-based IP trace back approach [4] records packet digests in a space-efficient data structure, bloom filter [5], to reduce the storage overhead significantly. Routers are queried in order to reconstruction the network path. the information required to achieve trace back is either stored at different points (mostly on routers) along the path that a packet traverses or that path and usually other incidental paths are analyzed to gain information that will be used in trace back. It is this distinction that we employ to further divide network based schemes into packet logging schemes and network analysis schemes. In this paper, we propose a new hybrid IP trace back scheme with efficient packet logging aiming to have a fixed storage requirement for each router in packet logging without the need to refresh the logged tracking information. In addition, we use a packet's marking field to censor attack traffic on its upstream routers. Like MRT and MORE, RIHT marks interface numbers of routers on packets so as to trace the path of packets. Since the marking field on each packet is limited, our packet-marking scheme may need to log the marking field into a hash table and store the table index on the packet. We repeat this marking/logging process until the packet reaches its destination. After that, we can reverse such process to trace back to the origin of attack packet

1.1Packet Logging Schemes:

In packet logging schemes, trace back is accomplished by requiring nodes that a packet Traverses to log information about that packet in such a format that can later be queried to discover if the packet has been seen at that node. The big drawback here is the huge amount of storage that may be needed at each node especially at nodes connected to high speed links that witness a lot of traffic. One of the most definitive works in IP Traceback is the Hashbased IP Traceback scheme [3] which computes and stores a Bloom filter digest for every packet at each node.



The scheme comprises a Source Path Isolation Engine (SPIE) which instead of storing whole packets, saves space by storing only computed hashes for each packet based on the invariant parts of the IP header along with the first 8 bytes of the payload. The

use of a Bloom filter was innovative in addressing the storage space problems of packet logging schemes but despite this, the requirements in terms of computation and space are still formidable.

As noted in [4] "assuming a packet size of 1000 bits, a duplex OC- 192 link requires 60 million hash operations to be performed every second, resulting in the use of SRAM (50ns DRAM is too slow for this) and 44GB of storage per hour, with the parameters suggested in" [3]. To resolve this, [4] went ahead to propose a packet logging based traceback scheme that is scalable to such high link speeds by sampling and logging only a small percentage of packets and then using more sophisticated techniques to achieve traceback. We note however that the approach taken by [4] loses the single packet traceback ability originally possessed by [3].

1.2 Packet Marking Schemes

Distinct from the ICMP messaging scheme, packet marking schemes encode information about the path a packet traverses in the packet itself, usually in rarely-used fields within the IP header. Apart from the well-known issue of finding enough space in the current IP header in which to place traceback information, another common problem with packet marking schemes arises from the additional computational tasks placed on routers during the marking process.

II. LITERATURE SURVEY

Packet logging scheme makes routers record the state information of packets, and it has the capability to trace a single packet. Therefore it can provide the straightforward evidence for traceback. It becomes practical when SPIE appeared, but it still needs great storage space. Packet marking approach makes routers write ID information into packet IP header, and reconstruct the complete path in the victim node. It does not increase storage burden on routers. Several researchers combine the features of the two approaches to propose hybrid IP traceback approaches [16-18]. The approaches are outstanding for their low storage overhead, and single packet traceback capability. In hybrid IP traceback, each traceback-enabled router will conduct logging or marking. However, hybrid IP traceback approaches should not be the simple combination of the two methods, and we must pay much attention to some key issues. Firstly, in normal condition, a marking router has no more than one neighboring logging router, which logged the attack packet. However, in some certain situations, packets will follow some special routes, and a marking router may have more neighboring logging routers.

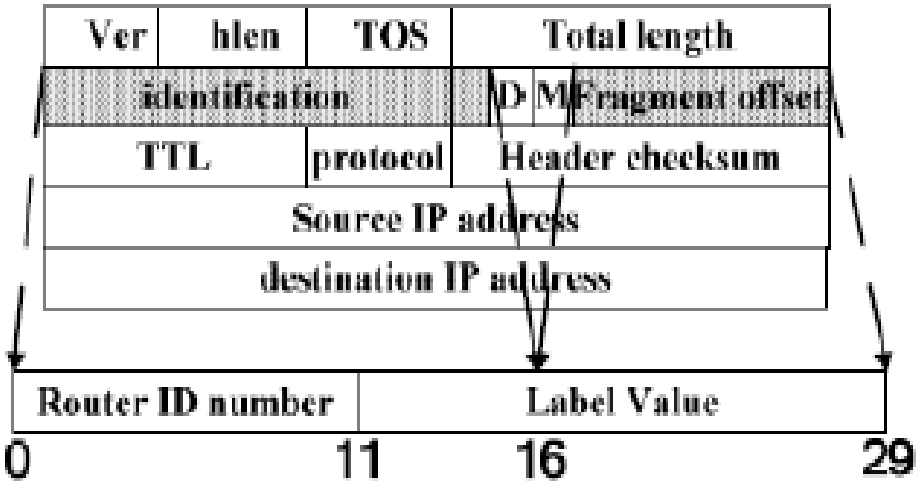


Fig. 1. IP Header

Hybrid IP traceback may return false paths, which could lead to the failure of traceback. Secondly, in the marking process, marking routers insert ID into IP header without logging, which can reduce storage overhead. The marking space in IP header is limited, and full utilization of the space can hold more ID information, which can reduce the percentage of logging routers in all traceback routers. In this way, the IP traceback approach is more practical. The IPv6 implementation of this traceback scheme needs more research because IPv6 has built-in security mechanisms such as authentication headers to provide origin authentication. Therefore, we leave it as our future work. This message can show the state information instead of logging in the router, which can reduce the storage overhead. Like packet marking approaches, this make use of some fields in the IP header when conduct marking. The utilization of IP header for marking. Three fields in IP header are used for marking: they are Identification, Reserved flag and fragment offset. Similar to other PPM, this traceback reuses the Identification field, which is for IP fragmentation. Measurement studies show that less than 0.25% of packets are fragmented in the Internet [19]. Savage *et al*. [8] had some discussion about the backward compatibility issues of utilization of this field and confirmed its utilization. This method reuses Reserved Flag (RF) field for marking as suggested in [20]. The fragment offset field will become meaningless if the IP Identification filed is reused for other purposes. Therefore, some traceback approaches utilize this field [21, 22], PPIT follows this way. Someone may argue that the value in the fragment offset field could cause some compatible problems. In order to avoid such issues in PPIT, the marking information will be removed before the packet arrives at the destination. In a certain area, IP address is too long to be taken in the IP header marking space and is unnecessary. In PPIT, each marking router is assigned a 14-bit ID number, which is to differentiate it to other neighboring routers. Muthuprasanna *et al*. [23] studied the unique ID number assignment problem for Internet routers and proposed an approach for internet colouring.

Table. 1. Packet operating procedure at router

Rid	Router ID
UIi	Upstream Interface
P	Received Packets
H()	Hash Function
M	Size of Hash Table
c1, c2	Constant
HT	m entries in hash table.
HT[index]	Entry of the hash table With the address index. HT[index].mark: mark field. HT[index].UI:UI field.
%	The modulo operation.

They report that 14 bits are enough for a unique ID number assignment within a three-hop neighborhood and 12 bits for two-hop neighborhood, based on the analysis of several Internet topology data sets. Furthermore, the same ID number can be assigned to routers more than one in different three-hop neighborhood, if these neighborhoods do not have a common router. Therefore, we can use such short ID information to replace IP address. PPIT makes use of the Identification field to store routers ID in 14 bits, which is called router ID field. The ID is generated for each traceback-enabled router, which is set by the network administrators. For each arriving packet, the current router first examines the router ID number marked in the packet header to check whether it is *valid*. The router ID number carried by a packet p is valid at a router r if it equals to the ID number of some neighbor router of router r. That is, the packet p was forwarded from some neighbor router to router r. If the router ID number is valid, based on the logging flag bit in the packet, the router may choose to commit (1) only marking operation, or (2) both

marking and logging operations. If the upstream router logged the packet (logging flag is 1), the current router chooses to only mark the packet; if the upstream router didn't log the packet (logging flag is 0), the current router chooses to both mark and log the packet. If the router ID number is not valid, that means the arriving packet came directly from the sender host or an attacker which sends packets with forged mark. In this case, the router chooses to commit only marking operation.

III. PROPOSED SCHEME

In this paper, proposed IP traceback scheme 16-bit marking and logging field in IP packet which solves the packet fragmentation problem. Also RIHT uses router degree for the calculation of marking value where as E-RIHT make use of router id for the calculation of marking value. If attack packet is received by the victim then it automatically discard the packet and send the path reconstruction request to upstream router.

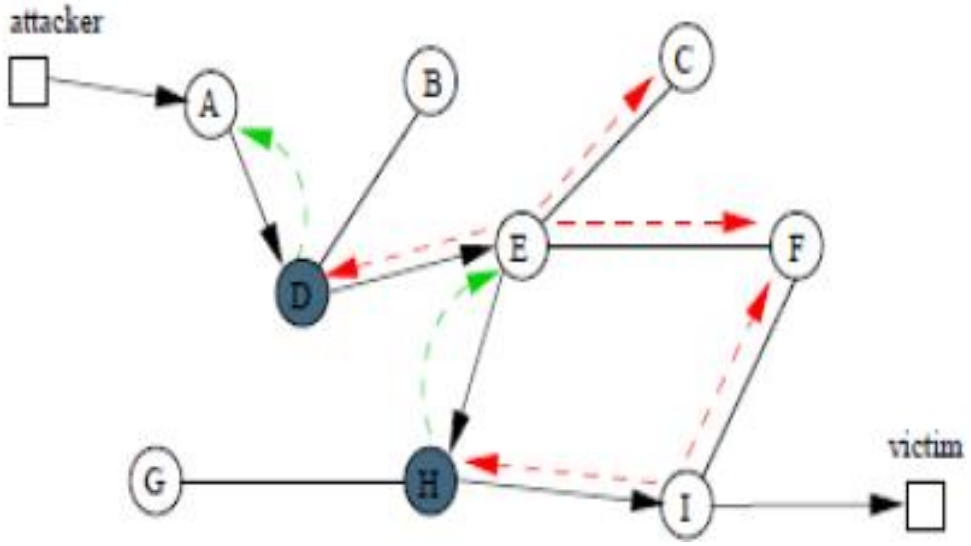


Fig. 2. Attack path construction. (i) Solid arrows represent the attack path; (ii) dashed curves represent the first method; dashed arrow represents the second method. (iii) Router D and H logged the attack packet For path reconstruction, identification field is used and path towards the attacker is constructed. 16-bit hash table is used for logging the 16-bit marking field. Information from the hash table can be retrieved very easily by using the marking field because of the index in the marking field. It is easy to search the particular information and trace back the path.

Here a new hybrid IP traceback scheme with efficient packet logging aiming to have a fixed storage requirement for each router (under 320 KB, according to CAIDA's skitter data set) in packet logging without the need to refresh the logged tracking information and to achieve zero false positive and false negative rates in attack-path reconstruction.

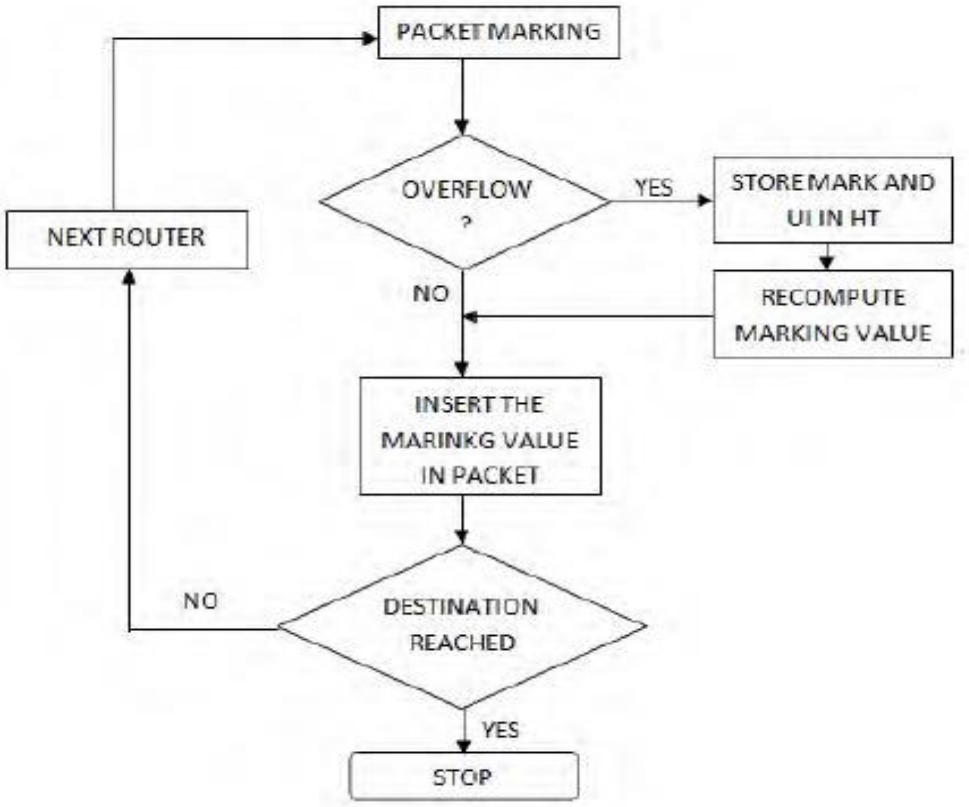


Fig.3. Flowchart for packet marking and packet logging.

Algorithm 1:

Packet marking and logging
Input: P, UIi
Begin
1. marknew = P.mark × (Rid + 1) + UIi +1
2. if marknew is overflow then
3. index = h = H(P.mark)
4. probe = 0
5. while not (HT[index] is empty or HT[index] is equal to (P.mark, UIi))
6. probe++
7. index = (h + C1 × probe + C2 × probe2) % m
8. endwhile
9. if HT[index] is empty then
10. HT[index]. Mark = P.mark
11. HT[index].UI = UIi
12. endif
13. marknew = index × (Rid + 1)
14. endif
15. P.mark = marknew
16. Forward the packet to the next router End

We propose a new hybrid IP traceback scheme with efficient packet logging aiming to have a fixed storage requirement for each router in packet logging without the need to refresh the logged tracking information. In addition, we use a packet's marking field to censor attack traffic on its upstream routers. Propose a new hybrid IP traceback scheme with efficient packet logging aiming to have a fixed storage requirement for each router in packet logging without the need to refresh the logged tracking information. In addition, we use a packet's marking field to censor attack traffic on its upstream routers.

Like MRT and MORE, RIHT marks interface numbers of routers on packets so as to trace the path of packets. Since the marking field on each packet is limited, our packet-marking scheme may need to log the marking field into a hash table and store the table index on the packet.

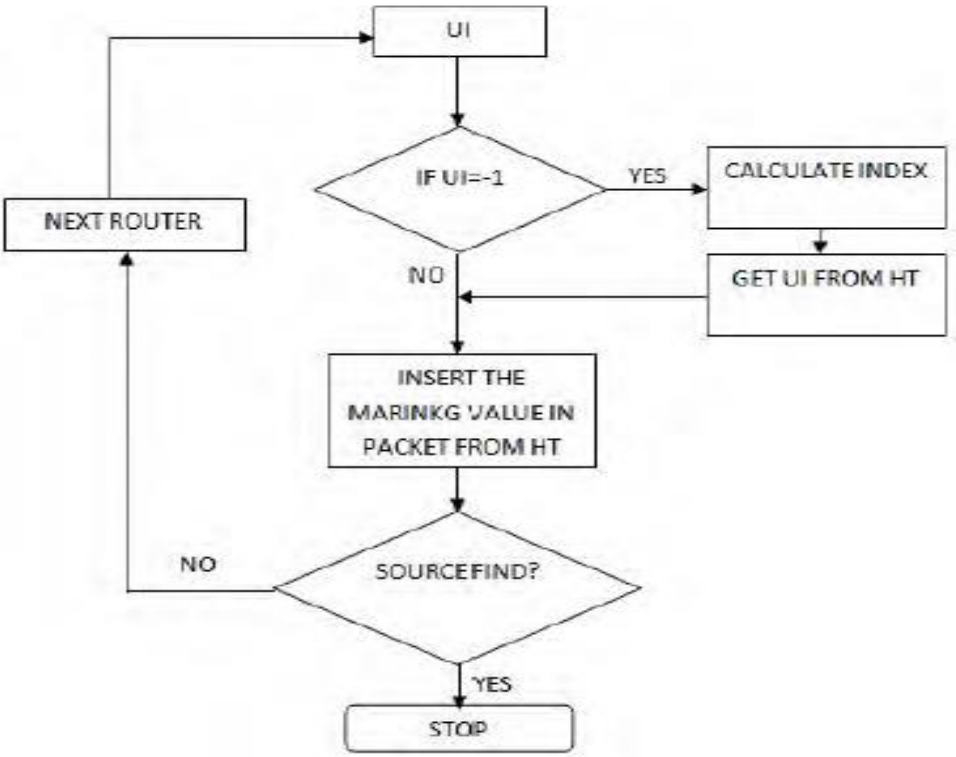


Fig.4. Flowchart for path reconstruction.

In the following algorithm P and UIi are taken as input in the beginning marknew is calculated using the formula P.mark × (Rid + 1) + UIi +1. Here Rid is the router id which is basically the IP address of the router that will be converted from 32-bit to 8-bit . Conversion is done by breaking an IP address into four 8-bit segments say S1, S2, S3 and S4. OR function is applied to the segments and 8-bit resultant is generated. Now incase marknew is overflow then index = h = H(P.mark) and probe = 0. After that checking HT[index] is empty or HT[index] is equal to (P.mark, UIi) then set probe++ and index = (h + C1 × probe + C2 × probe2) % m. Now incase HT[index] is empty then set HT[index]. Mark = P.mark and HT[index].UI = UIi. Then set marknew = index × (Rid + 1). At last set P.mark = marknew and Forward the packet to the next router.

Algorithm 2:

Path Reconstruction
Begin
1. UIi = markreq %(Rid + 1) – 1
2. if UIi = -1 then
3. index = markreq / (Rid+ 1)
4. If not index = 0 then
5. UIi = HT[index].UI
6. markold = HT[index].mark
7. Send reconstruction request with markold to upstream router by UIi
8. else
9. This router is the nearest border router to the attacker
10. Endif
11. else
12. markold = markreq / (Rid + 1)

13. Send reconstruction request with markold to upstream router by UIi
14. endif
end

We repeat this marking/logging process until the packet reaches its destination. After that, we can reverse such process to trace back to the origin of attack packets. Combine packet marking with packet logging and therefore create hybrid IP traceback schemes demanding less storage but requiring a longer search. In this paper, we propose a new hybrid IP traceback scheme with efficient packet logging aiming to have a fixed storage requirement for each router (under 320 KB, according to CAIDA's skitter data set) in packet logging without the need to refresh the logged tracking information and to achieve zero false positive and false negative rates in attack-path reconstruction.

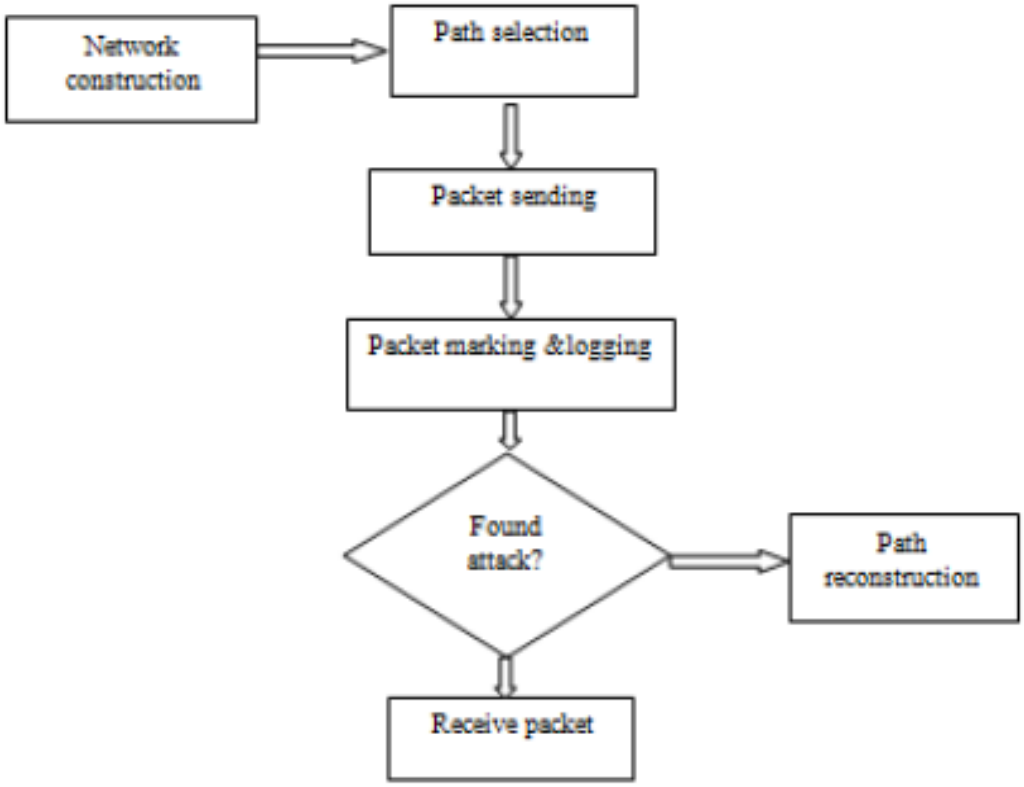


Fig.5. Flowchart for packet marking, logging and path reconstruction.

Algorithm 3:

Packet operating procedure at router
For each packet p
IF router ID number i carried by p is valid
IF the logging flag bit in p is 0
Compute the digest of p
Store the digest in the digest table corresponding to i
Mark p with R's ID number
Set the logging flag bit in p to be 1
ELSE
Mark p with R's ID number
Set the logging flag bit in p to be 0
ELSE
Mark p with R's ID number
Set the logging flag bit in p to be 0

Trace back Process:

In the trace back process, the total number of the digest tables examined is an index reflecting the overhead on the trace back server and the speed of the trace back process. Suppose time synchronization is maintained between adjacent routers, and each router has n neighbors on average. Suppose the attack path is m hops long from the attacker to the victim. Let the average link latency between routers be l.

If

$$t_c = t_h \times n \quad - \quad (1)$$

the average link latency between routers is larger than the average time interval covered by one digest table, multiple digest tables covering continuous time periods at one router or one interface will be examined during the traceback process. Then, during the traceback process, the ratio of the number of digest tables examined in the hybrid approach to that in the hash-based approach is between n/2 and 1 /2 , depending on the average link latency between routers. The mathematical deduction below is based on average values of parameters and omits small value constants. Suppose each router has n (n >= 2) neighbor routers on average, and the traffic load at the router is from each neighbors equally.

Then

$$N_h = m \times n \times \left[\frac{l}{t_h}\right] = m \times n \times \left[\frac{l \times n}{t_c}\right] \text{ - (2)}$$

In Summary,

$$\frac{1}{2} \leq r \leq \frac{n}{2}. \quad - \quad (3)$$

Let the average time interval covered by one digest table in the hash-base approach be th, and the average time interval covered by one digest table in the hybrid approach be tc. Suppose the average time interval covered by one digest table is t, then d l t e tables need to be examined in order to locate

the digest of attack packet. In the hash-based approach, in order to move one hop upstream along the attack path from the current router during the traceback process, the digest tables at n neighbor routers need to be examined (actually n -1, we omit that constant for simplicity). The number of digest tables examined is

IV. CONCLUSION

In this paper, we proposed a new attack path reconstruction approach which is based on both packet marking and packet logging. Our approach has the ability to track packet back to its origin. Compared to hash-based IP traceback approach, it reduces the storage overhead to roughly one half and improves accuracy on the access time by a factor of the number of neighbour routers.

REFERENCES

[1] C. Gong and K. Sarac, "A more practical approach for single-packet IP traceback using packet logging and marking," *IEEE Trans. Parallel Distributed Syst.*, vol. 19, no. 10, pp. 1310–1324, Oct. 2008.

[2] A. Hussain, J. Heidemann, and C. Papadopoulos, "A framework for classifying denial of service attacks," in *Proc. ACM SIGCOMM '03*, Karlsruhe, Germany, Aug. 2003, pp. 99–110.

[3] B.Al-Duwari andM. Govindarasu, "Novel hybrid schemes employing packet marking and logging for IP traceback," *IEEE Trans.*

Parallel Distributed Syst., vol. 17, no. 5, pp. 403–418, May 2006.

[4] H. Burch and B. Cheswick, "Tracing anonymous packets to their approximate source," in *Proc. of the 14th USENIX Systems Administration Conferrence*, December 2000.

[5] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Network support for IP traceback," *IEEE/ACM Transactions on Networking*, Vol. 9, 2001, pp. 226-237.

[6] D. X. Song and A. Perrig, "Advanced and authenticated marking schemes for IP traceback," in *Proceedings of the IEEE INFOCOM*, 2001, pp. 878-886.

[7] C. Gong and K. Sarac, "Toward a practical packet marking approach for IP traceback," Int. J. Network Security , vol. 8, no. 3, pp. 271–281, Mar. 2009.

[8] C. Gong and K. Sarac, "A more practical approach for single-packet IP traceback using packet logging and marking," IEEE Trans. Parallel Distributed Syst. , vol. 19, no. 10, pp. 1310–1324, Oct. 2008.

[9] C. Gong and K. Sarac, "IP traceback based on packet marking and logging," Proc. of IEEE International Conference on Communications, Seoul, Korea, May 2005.

AUTHORS PROFILE

E.Jansi, B.Tech is pursuing her M.tech in Prist University, Pondy, India

Ms. BHARATHI.R Received The M.Tech In Computer Science And Engineering. Presently she is A Working Assistant Professor in Computer Science and Engineering at PRIST University, Puducherry Campus, and Puducherry, India.

Mr. E.Pushparaj received the Bachelor of Engineering in Electronics and Communication Engineering from the Anna University, Chennai, Tamil Nadu, India, in 2009, and the Master of Engineering in Applied Electronics from the Anna University of Technology, Coimbatore (University Department), Tamil Nadu, India, in 2011, and he is a member of ISTE. Where, He is currently working towards the Ph.D degree at the Department of Electronics and Communication Engineering. He is presently working as an Assistant Professor in the Department of Electronics and Communication Engineering at Ranganathan Engineering College, Coimbatore, TN. He has published many research papers in reputed National and International Conferences in India. He has conducted many Staff Development Programs, workshops, Symposiums and conferences. His current research interests include Communication and Networking, VLSI design, Digital Image Processing and So on.

