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. 9, September 2014, pg.324 – 331

RESEARCH ARTICLE

RIHT: A Novel Hybrid IP Traceback Scheme

[1]Chinthaparthy Charan Kumar Reddy, [2]Mrs. K.Srilakshmi

[1]M.Tech, Mallareddy College of Engineering & Technology, Hyderabad, INDIA

E-mail: c.charankumarreddy@gmail.com

[2]Assistant Professor, Mallareddy College of Engineering & Technology, Hyderabad, INDIA

E-mail: ksrilakshmi.d@gmail.com

ABSTRACT- Because the Internet has been widely applied in various fields, more and more network protection issues emerge and catch people's attention. Though, adversaries often hide themselves by spoofing their own IP addresses and then launch attacks. For this reason, researchers have planned a lot of traceback schemes to trace the source of these attacks. Various use only one packet in their packet logging schemes to achieve IP tracking. Others unite packet marking with packet logging and therefore create hybrid IP traceback schemes demanding less storage but requiring a longer search. In this paper, we suggest 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. In addition, we utilize a packet's marking field to censor attack traffic on its upstream routers. Finally, we simulate and analyze our scheme, in evaluation with other related research, in the following aspects: computation, storage requirement, and accuracy.

1. INTRODUCTION

With the rapid growth of the Internet, various internet applications are developed for different kinds of users. Due to the decreasing cost of Internet access and its increasing availability from a plethora of devices and applications, the impact of attacks becomes more significant. To disrupt the service of a server, the sophisticated attackers may launch a distributed denial of service (DDoS) attack. Based on the number of packets to deny the service of a server, we can categorize DDoS attacks into flooding-based attacks and software exploit attacks [10].

The major signature of flooding-based attacks is a huge amount of forged source packets to exhaust a victim's limited resources. Another type of DoS attack, software exploit attacks, attacks a host using the host's vulnerabilities with few packets

(e.g., Teardrop attack and LAND attack). Since most edge routers do not check the origin's address of a packet, core routers have difficulties in recognizing the source of packets. The source IP address in a packet can be spoofed when an attacker wants to hide himself from tracing. So, IP spoofing makes hosts hard to defend against a DDoS attack. For these reasons, increasing a mechanism to locate the real source of impersonation attacks has become an important issue nowadays.

For tracing the actual source of flooding-based attack packets, we propose a traceback scheme that marks routers' interface numbers and integrates packet logging with a hash table (RIHT) to deal with these logging and marking issues in IP traceback. Packet marking can be put into two categories,

deterministic packet marking (DPM) and probabilistic packet marking (PPM). Belenky and Ansari [3], [4] propose DPM traceback schemes to mark a border routers' IP address on the passing packets. Though, IP header's identification field is not enough to store the full IP address. For this reason, the border router divides its IP into several segments and computes the digest of its IP. Then it randomly chooses a segment and the digest to mark on its passing packets. When the destination host receives enough packets, it can use the process to assemble the different segments. On the other hand, Savage et al. [18] propose a PPM scheme with edge sampling which is called FMS. Song and Perrig [20] propose the AMS scheme. Yaar et al. [24] propose the FIT scheme. Al-Duwari and Govindarasu [1] propose the probabilistic pipelined packet marking (PPPM) scheme. Gong and Sarac [26] propose a practical packet marking scheme.

These probability-based schemes require routers to mark partial path information on the packets which pass through them with a probability. That is to declare, if a victim collects enough marked packets, it can reconstruct the full attack path. Since flooding-based traceback schemes need to collect a large amount of attack packets to find the origin of attacks, these schemes are not suitable for tracing the origins of software exploit attacks Most current tracing schemes that are designed for software exploits can be categorized into three groups: single packet, packet logging [9], [19], and hybrid IP traceback [1], [8], [9], [15], [16], [25]. The basic idea of packet logging is to log a packet's information on routers. Huffman codes [8], Modulo/ Reverse modulo Technique (MRT) [15] and MOdulo/Reverse modulo (MORE) [16] use interface numbers of routers, in its place of partial IP or link information, toward mark a packet's route information. Each of these methods marks routers' border 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. Hence, they integrate packet logging into their marking schemes by allowing a packet's marking field temporarily logged on routers. We get these tracing methods still require high storage on logged routers. Moreover, their schemes cannot avoid a false positive problem because their packet digests in each log table may have collision, and their schemes even have false depressing problem when routers refresh logged data. Separately from these, we find their exhaustive searching quite inefficient in path reconstruction. For these reasons, we suggest a traceback scheme that marks routers' interface numbers and integrates packet logging with a hash table (RIHT) to deal with these logging and marking issues in IP traceback. RIHT be a hybrid IP traceback scheme designed to achieve the following properties: 1) Our storage requirement for an arbitrary router is bounded above by the number of paths to the router, and hence every router does not need to refresh logged tracking information. 2) Our scheme achieves nothing false positive and false negative rates in attack-path reconstruction. 3) We have superior efficiency in path reconstruction. 4) Our scheme can edit attack traffic.

2. RELATED WORK

Most of current single packet traceback schemes tend to log packets' information on routers. For instance, Snoeren *et al.*[19] propose a system SPIE to digest the unchanged parts of a packet and used bloom filter to log the digest. However, this scheme requires large storage space and has a false positive problem in the bloom filter. For this reason, Zhang and Guan [25] propose TOPO to improve the efficiency and precision of SPIE, but TOPO still needs large storage capacity and inevitably has a false positive problem because of the bloom filter. The hybrid IP traceback schemes are introduced to mitigate the storage problem of logging-based traceback schemes. Gong and Sarac [9] propose a hybrid IP traceback scheme called Hybrid IP Traceback (HIT) combining packet marking and packet logging. HIT uses packet marking to reduce the number of routers required for logging. Further researchers have proposed new schemes to other reduce the storage requirement for router logging and to decrease the number of routers required for logging, Modulo/Reverse modulo Technique (MRT) [15] and MOdulo/REverse modulo (MORE) [16].

Since these schemes use interface numbers of routers for marking, they assume a router set R={R1,R2,….,Ri,….Rl} comprising l routers in a network and require all the routers support the

respective traceback schemes. And, they use the degree of a router as a parameter in their marking schemes where the degree is the number of interfaces of the router, not including ports connected to local networks. Here we use $D(R_i)$ to denote the degree of a router R_i. Besides, these schemes need to maintain an interface table on each router in advance. This table maps a unique number to each interface of a router along which the router is connected to another router. The interface numbers of a router R_i are between 0 and $D(R_i)$ -1 . For discussion, we denote by UI^r_i (or UI_i if there is no ambiguity) the upstream interface number of a R_i router in a route r. In what follows, we use routes and paths interchangeably. In the marking process, each router puts UI_i into the marking field. Perhaps the simplest way to encode UI_i is by fixed-length coding [8]. However, such an approach does not use a packet's marking field efficiently $D(R_i)$ if is not a power of two. Choi and Dai [8] propose a marking scheme using Huffman coding to reduce the bits required for marking on a packet. It encodes UI_i by Huffman coding according to the traffic of each interface. Their analysis shows their scheme has superior performance when the traffic distribution for each interface is unequal.

Malliga and Tamilarasi propose two traceback schemes, namely MRT [15] and MORE [16]. While MRT uses a 32-bit marking field, MORE uses a 16-bit marking field and separates a log table into $D(R_i)$ parts. They use mathematical methods to mark the marking fields. In their marking schemes, the new marking field =marking field * $D(R_i)$+ UI_i is computed by the routers to which a packet is forwarded. In their path rebuilding, the old marking field= marking field/ $D(R_i)$ is computed by the routers to which a packet is traced back; the upstream interface number UI_i =marking field% $D(R_i)$is also computed where % is the modulo operation, also the packet is sent back to the upstream router along the obtained upstream interface. According to the test results in MRT and MORE, the average bits used for marking are fewer than those in Huffman coding.

3. RIHT

Like MRT and MORE, RIHT marks interface numbers of routers on packets so as to trace the path of packets. While the marking field on each packet is limited, our packet-marking scheme can need to log the marking field into a hash table and store the table index on the packet. We replacement this marking/logging process until the packet reaches its destination. After that, we can reverse such procedure to trace back to the origin of attack packets.

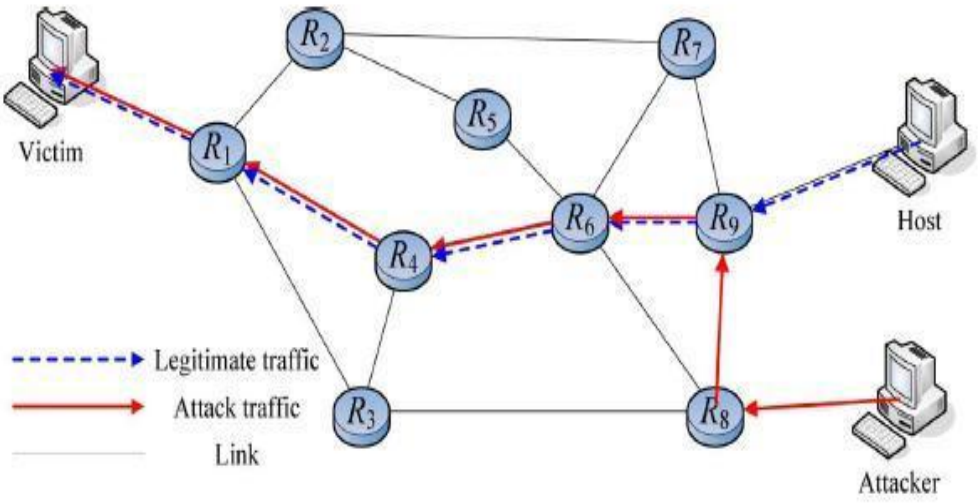

Fig. 1. Network topology

Network Topology and Preliminaries

As the network topology shows in Fig. 1, a router can be connected to a local network or other routers, or even both. A frame router receives packets from its local network. A core router receives packets from further routers. For example, serves as a border router when it receives packets from Host. Though, it becomes a core router when receiving packets from. The assumptions of our methods are as follows.

1) A router creates an interface table and numbers the upstream interfaces from 0 to $D(R_i)$-1 in advance.
2) A router knows whether a packet comes from a router or a local network.
3) Such a traceback method is viable on every router.
4) The traffic route and network topology can be changed, but not often.

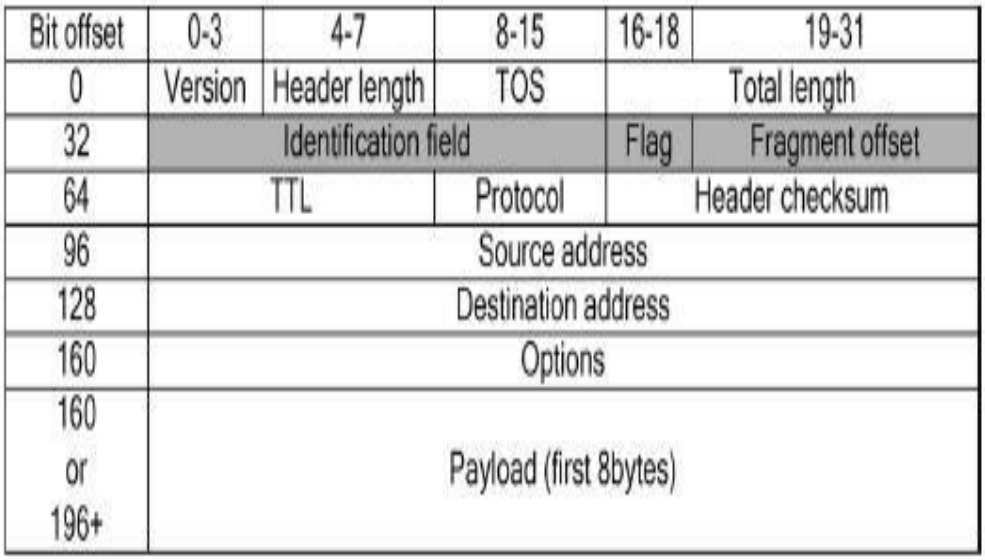

Fig. 2. Fields of an IP packet. We use the gray fields as marking field in RIHT.

If we use the identification field to mark a packet, it can lead to identification number collision in the reassembling process. In fact, the chance of segmented packets has been getting lower and lower from 0.25% to 0.06%, according to Stocia *et al.* [21] and John *et al.* [11]. The two pieces of research also help support our argument about the effect of MTU usage in TCP. By using MTU, there is no big fragment problem raised in OpSec [27]. John *et al.* [12] also point out that over 60% of fragmented packets are attacking packets. Therefore, if attackers try to use Encapsulating Security Payload (ESP) packets to evade IDS, their randomly generated ESP packets can never be decrypted at a victim's site because of the lack of proper shared keys. In such a case, the adversaries can only generate a large volume of forged ESP packets to attack a host, overriding the victim's bandwidth and computation resources.

TABLE I
NOTATIONS

R_i	$\{R_1, R_2, \ldots, R_i, \ldots, R_x\}$, routers in the internet
$D(R_i)$	the degree of R_i
UI_i^r (or UI_i)	the upstream interface number of the router R_i in the route r (or UI_i if there is no ambiguity)
P	the received packet
$H()$	a hash function
m	the size of a hash table (i.e. the number of slots in a hash table)
c_1, c_2	constants
HT	an m entries hash table
$HT[index]$	$HT[index]$: the entry of the hash table HT with the address $index$ ($HT[0]$ is reserved) $HT[index].mark$: $HT[index]$'s $mark$ field $HT[index].UI$: $HT[index]$'s UI field
%	the modulo operation

If we scratch the ESP packets with a low probability, the marked packets are enough for us to trace the attackers' source; also the unmarked segmented ESP packets are still able to assemble at the destination host. In some cases, adversaries can compromise a node in the target network first. Then they can use ESP packets in their software exploit, e.g., Teardrop assault and LAND attack, which lean to consume destination hosts' buffer and computation resources. But we overwrite the fragment field; the attackers are not bright to launch Teardrop attacks to deny the service at a victim's site. If we overwrite together the fragment field and the fragment "flag", then adversaries can no longer keep a victim waiting for the last fragmented segment.

As mentioned above, the use of the fragment and the identification fields will not affect most legitimate packets. Further, fragmentation is commonly used for IDS evasion. Hence, when we overwrite these two fields in our traceback design, we avoid attackers using fragmented packets to evade IDS. For this cause, we use an IP header's recognition field, flag field, and fragment offset field as a 32-bit marking field, shown in Fig. 2. Notations in this paper are shown in Table I.

4. CONCLUSION

In this paper, we propose a new hybrid IP traceback scheme (RIHT) for efficient packet logging aiming to have a fixed storage requirement in packet logging without the need to refresh the logged tracking information. And, the proposed scheme has zero false positive and false negative rates in an attack-path reconstruction. Separately from these properties, our scheme can also install a marking field as a packet identity to filter malicious traffic and secure against DoS/DDoS attacks. Accordingly, with high accuracy, a low storage condition, and fast computation, RIHT can provide as an efficient and secure scheme for hybrid IP traceback.

REFERENCES

[1] 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.

[2] A. Appleby, Murmurhash 2010 [Online].
Available: http://sites.google.com/site/murmurhash

[3] A. Belenky and N. Ansari, "IP traceback with deterministic packet marking," IEEE Commun. Lett., vol. 7, no. 4, pp. 162–164, Apr. 2003.

[4] A. Belenky and N. Ansari, "Tracing multiple attackers with deterministic packet marking (DPM)," in Proc. IEEE PACRIM'03, Victoria, BC, Canada, Aug. 2003, pp. 49–52.

[5] S. M. Bellovin, M. D. Leech, and T. Taylor, "ICMP traceback messages," Internet Draft: Draft-Ietf-Itrace-04.Txt, Feb. 2003.

[6] H. Burch and B. Cheswick, "Tracing anonymous packets to their approximate source," in Proc. USENIX LISA 2000, New Orleans, LA, Dec. 2000, pp. 319–327.

[7] CAIDA's Skitter Project CAIDA, 2010 [Online].
Available: http://www.caida.org/tools/skitter/

[8] K. H. Choi and H. K. Dai, "A marking scheme using Huffman codes for IP traceback," in Proc. 7th Int. Symp. Parallel Architectures, Algorithms Networks (SPAN'04), Hong Kong, China, May 2004, pp. 421–428.

[9] 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.

[10] 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.

[11] W. John and S. Tafvelin, "Analysis of internet backbone traffic and header anomalies observed," in Proc. IMC '07: 7th ACM SIGCOMM Conf. Internet Measurement, San Diego, CA, Oct. 2007, pp. 111–116.

[12] W. John and T. Olovsson, "Detection of malicious traffic on backbone links via packet

header analysis," Campus-Wide Inform. Syst., vol. 25, no. 5, pp. 342–358, 2008.

[13] D. E. Knuth, The Art of Computer Programming, 2nd ed. Redwood City, CA: Addison Wesley Longman, 1998, vol. 3, pp. 513–558.

[14] T. Korkmaz, C. Gong, K. Sarac, and S. G. Dykes, "Single packet IP traceback in AS-level partial deployment scenario," Int. J. Security Networks, vol. 2, no. 1/2, p. 95–108, 2007.

[15] S. Malliga and A. Tamilarasi, "A proposal for new marking scheme with its performance evaluation for IP traceback,"WSEAS Trans. Computer Res., vol. 3, no. 4, pp. 259–272, Apr. 2008.

[16] S. Malliga and A. Tamilarasi, "A hybrid scheme using packet marking and logging for IP traceback," Int. J. Internet Protocol Technol., vol. 5, no. 1/2, pp. 81–91, Apr. 2010.

[17] L. C. Noll, FNV Hash 2010 [Online]. Available: http://www.isthe.com/ chongo/ tech/comp/fnv/index.html

[18] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Practical network support for IP traceback," in Proc. ACM SIGCOMM2000, Stockholm, Sweden, Aug. 2000, pp. 295–306.

[19] A. C. Snoeren, 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 Trans. Networking, vol. 10, no. 6, pp. 721– 734, Dec. 2002.

[20] D. X. Song and A. Perrig, "Advanced and authenticated marking schemes for IP traceback," in Proc. IEEE INFOCOM2001, Anchorage, AK, Apr. 2001, pp. 78–886.

[21] I. Stocia and H. Zhang, "Providing guaranteed services without peer flow management," in Proc. ACM SIGCOMM'99, Boston, MA, Sep. 1999, pp. 81–94.

[22] The MD5 Message-Digest Algorithm. : IEFT RFC 1321, 1992.

[23] X. J.Wang and Y. L. Xiao, "IP traceback based on deterministic packet marking and logging," in Proc. SCALCOM-EMBEDDEDCOM'09, Dalian, China, Sep. 2009, pp. 178–182.

[24] A. Yaar, A. Perrig, and D. Song, "FIT: Fast internet traceback," in Proc. IEEE INFOCOM2005, Miami, FL, Mar. 2005, pp. 1395–1406.

[25] L. Zhang and Y. Guan, "TOPO: A topology-aware single packet attack traceback scheme," in Proc. IEEE In. Conf. Security Privacy Communication Networks (SecureComm 2006), Baltimore, MD, Aug. 2006, pp. 1–10.

[26] 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.

[27] F. Gont, "Security assessment of the internet protocol version 4," Internet Draft: Draft-Ietf-Opsec-Ip-Security-07.Txt, Apr. 2011

Author's Profile

First Author1

Chinthaparthy Charan Kumar Reddy,
M.Tech in CS&E, Mallareddy College of Engineering & Technology, Hyderabad

Second Author

M Sreelakshmi,
Assistant Professor, Dept. of CS&E,
Mallareddy College of Engineering & Technology, Hyderabad

