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. 2, Issue. 12, December 2013, pg.145 – 152

RESEARCH ARTICLE

A hybrid IP Trace Back Scheme Using Integrate Packet logging with hash Table under Fixed Storage

SSVR Kumar Addagarla[1], Usha Nag S[2]

[1] Dept of Comp Science, Prasiddha College of Engineering & Technology, Amalapuram, India

[2]Dept of Comp Science, Prasiddha College of Engineering & Technology, Amalapuram, India

[1] ssvrkumar.research@gmail.com; [2] ushanag1686@gmail.com

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 traceback 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 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) 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: Spoofing; DDos; hybrid IP traceback; packet logging; packet marking

I. 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 [8]. 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. Therefore, IP spoofing makes hosts hard to defend against a DDoS attack.

For these reasons, developing a mechanism to locate the real source of impersonation attacks has become an important issue nowadays.

For tracing the real source of flooding-based attack packets, we propose a traceback scheme that marks routers' interface numbers and integrates packet logging with a hash table 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. However, 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 digest to assemble the different segments. On the other hand, Savage *et al.* [8] propose a PPM scheme with edge sampling which is called FMS. Song and Perrig [20] propose the AMS scheme. Yaar *et al.* [2] propose the FIT scheme. Al-Duwari and Govindarasu [1] propose the probabilistic pipelined packet marking (PPPM) scheme. Gong and Sarac [6] 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 say, 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 [7], [8], and hybrid IP traceback [1], [8], [6], [5]. The basic idea of packet logging is to log a packet's information on routers. Huffman codes [8], Modulo/ Reverse modulo Technique (MRT) [5] and Modulo/Reverse modulo (MORE) [6] 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. We find these tracing methods still require high storage on logged routers. Also, 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 negative problem when routers refresh logged data.Apart from these,we find their exhaustive searching quite inefficient in path reconstruction. For these reasons, we propose a traceback scheme that marks routers' interface numbers and integrates packet logging with a hash table to deal with these logging and marking issues in IP traceback.

Our hybrid IP traceback scheme designed to achieve the following properties:

- Our storage requirement for an arbitrary router is bounded above by the number of paths to the router, and thus every router does not need to refresh logged tracking information.
- Our scheme achieves zero false positive and false negative rates in attack-path reconstruction.
- We have higher efficiency in path reconstruction.
- Our scheme can censor attack traffic.
- Our propose packet sending scheme can be shown below

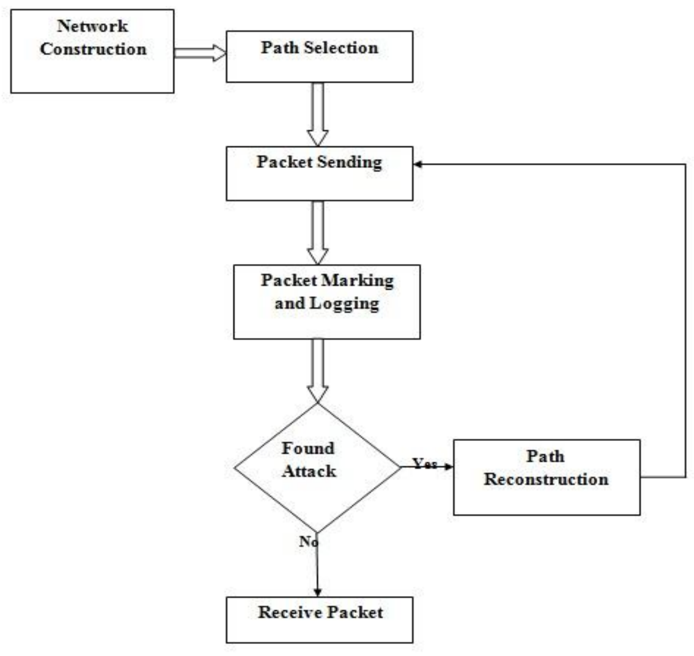

Figure 1. Packet sending Construction Scheme

Routers' interface numbers and integrates packet logging with a hash table 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 packets[6]. The entire work of this paper is divided into five different modules. They are:

- Network topology Construction
- Path Selection
- Packet Sending
- Packet Marking and Logging
- Path Reconstruction

II. Network topology Construction

A Network Topology may consist of the number of routers that are connected with local area networks. Thus, a router can either receive data from the nearer router or from the local area network. A border router receives packets from its local network [5]. A core router receives packets from other routers. The number of routers connected to a single router is called as the degree of a router. This is calculated and stored in a table. The Upstream interfaces of each router also have to be found and stored in the interface table.

As the network topology shows in Fig. 2, a router can be connected to a local network or other routers, or even both. A border router receives packets from its local network. A core router receives packets from other routers [4]. For example, serves as a border router when it receives packets from Host. However, it becomes a core router when receiving packets from the assumptions of our scheme are as follows.

1) A router creates an interface table and numbers the up-stream interfaces in advance.
2) A router knows whether a packet comes from a router or a local network.
3) Such a traceback scheme is viable on every router.
4) The traffic route and network topology may be changed, but not often.

If we use the identification field to mark a packet, it can lead to identification number collision in the reassembling process.

III. Path Selection

The path is said to be the way in which the selected packet or file has to be sent from the source to the destination [3][2]**.** The Upstream interfaces of each router have to be found and it is stored in the interface table. With the help of that interface table, the desired path between the selected source and destination can be defined.

IV. Packet Sending

One of the Packet or file is to be selected for the transformation process. The packet is sent along the defined path from the source LAN to destination LAN[1][5]. The destination LAN receives the packet and checks whether that it has been sent along the defined path or not.

V. Packet Marking and Logging

Packet marking is the phase, where the efficient Packet Marking algorithm is applied at each router along the defined path. It calculates the Pmark value and stores in the hash table. If the Pmark is not overflow than the capacity of the router, then it is sent to the next router. Otherwise it refers the hash table and again applies the algorithm. The packet marking and logging scheme and Notations are shown below.

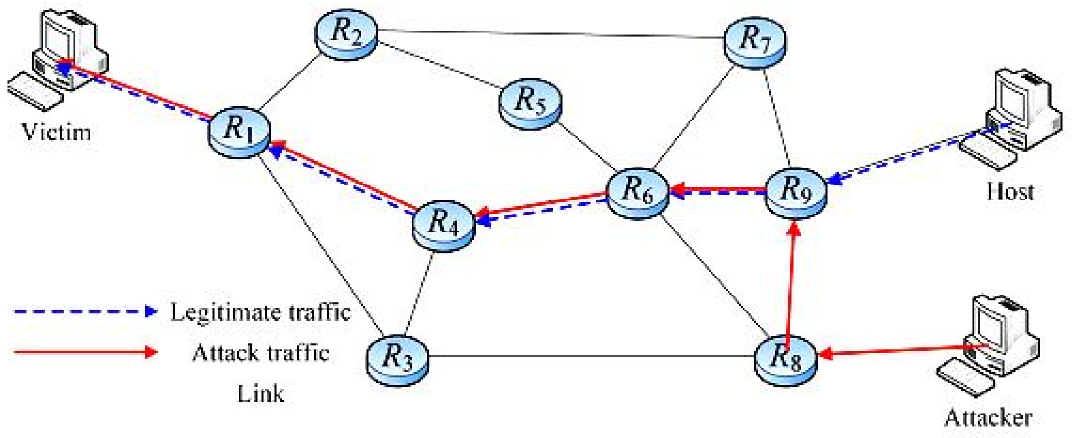

Figure 2. Network Topology for path construction

Notations:

R_i	{R1,R2,R3……Ri,…..Rx} Routers in the internet
$D(R_i)$	The degree of R_i
UI_i	The upstream interface number of the router R_i in the route r
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)
C1,c2	Constant
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 operations

Packet Marking and Logging Scheme

```
Input: P, $UI_i$
begin
1.  $mark_{new}$=P.mark X (D($R_i$)+1) + $UI_i$+1
2.  if $mark_{new}$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, $UI_i$))
6.  probe++
7.  index=(h+c1 x probe + c2 x $probe^2$)%m
8.  end while
9.  if HT[index] is empty then
10. HT[index].mark=P.mark
11. HT[index].UI=UIi
12. Endif
13. $mark_{new}$=index x (D($R_i$)+1)
14. endif
15. P.mark=$mark_{new}$
16. Forward the packet to the next router
End
```

VI. Path Reconstruction

Once the Packet has reached the destination after applying the Algorithm, there it checks whether it has sent from the correct upstream interfaces. If any of the attack is found, it request for the Path Reconstruction. Path Reconstruction is the Process of finding the new path for the same source and the destination in which no attack can be made. The below shows the path reconstruction scheme.

Path Reconstruction Scheme

```
    Begin
1.  $UI_i$=$mark_{req}$%($D(R_i)$+1)-1
2.  If $UI_i$ = -1 then
3.  Index = $mark_{reg}$/($D(R_i)$+1)
4.  If not index = 0 then
5.  $UI_i$ =HT[index].UI
6.  $mark_{old}$ = HT[index].mark
7.  send reconstruction request with $mark_{old}$ to upstream router by $UI_i$
8.  else
9.  this router is the nearest border router to the attacker
10. endif
11. else
12. $mark_{old}$ =$mark_{req}$/($D(R_i)$+1)
13. send reconstruction request with $mark_{old}$ to upstream router by $UI_i$
14. endif
    End
```

VII. CONCLUSION

In this paper, we propose a new hybrid IP traceback scheme for efficient packet logging aiming to have a fixed storage requirement in packet logging without the need to refresh the logged tracking information. Also, the proposed scheme has zero false positive and false negative rates in an attack-path reconstruction. Apart from these properties, our scheme can also deploy a marking field as a packet identity to filter malicious traffic and secure against DoS/DDoS attacks. Consequently, with high accuracy, a low storage requirement, and fast computation, our scheme can serve as an efficient and secure for hybrid IP traceback. As for our future work, we would like to come up with another version of the scheme which uses a 16-bit marking field to avoid the problem caused by packet fragmentation.

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 packetmarking (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.