IP Traceback based on Packet Marking and Logging

Chao Gong
Department of Computer Science
University of Texas at Dallas, USA
Email: gong@student.utdallas.edu

Kamil Sarac
Department of Computer Science
University of Texas at Dallas, USA
Email: ksarac@utdallas.edu

***Abstract*— Two main kinds of IP traceback techniques have been proposed in two dimensions: packet marking and packet logging. IP traceback based on packet marking is often referred to as probabilistic packet marking (PPM) approach where packets are probabilistically marked with partial path information as they are forwarded by routers. This approach incurs little overhead at routers. But due to its probabilistic nature, it can only determine the source of the traffic composed of a number of packets. IP traceback based on packet logging is often referred to as hash-based approach where routers compute and store digest for each forwarded packet. This approach can trace an individual packet to its source. However, the storage space requirement for packet digests and the access time requirement for recording packets commensurate with their arriving rate are prohibitive at routers with high speed links. We propose an IP traceback approach based on both packet marking and packet logging. Compared with the PPM approach, our approach is able to track individual packets. Compared with the hash-based approach, our approach incurs less storage overhead and less access time overhead at routers. Specifically, the storage overhead is reduced to roughly one half, and the access time requirement is decreased by a factor of the number of neighbor routers.**

I. Introduction

The goal of IP traceback is to trace the path of an IP packet to its origin. The most important usage of IP traceback is to deal with certain denial-of-service (DoS) attacks, where the source IP address is spoofed by attackers. Identifying the sources of attack packets is a significant step in making attackers accountable. In addition, figuring out the network path which the attack traffic follows can improve the efficacy of defense measures such as packet filtering as they can be applied further from the victim and closer to the source.

Two main kinds of IP traceback techniques have been proposed in two orthogonal dimensions: packet marking [1] and packet logging [2]. In packet marking, the router marks forwarded IP packets with its identification information. Because of the limited space in packet header, routers probabilistically decide to mark packets so that each marked packet carries only partial path information. The network path can be reconstructed by combining a modest number of packets containing mark. This approach is known as probabilistic packet marking (PPM) [3]. The PPM approach incurs little overhead at routers. But it can only trace the traffic composed of a number of packets because of its probabilistic nature.

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 traceback 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 reconstruct the network path. This approach can track a single IP packet. However, the requirements for digest table storage and access time to record packets commensurate with their arrival are prohibitive at routers with high speed links.

We propose to develop a hybrid IP traceback approach based on both packet marking and packet logging. The motivation is to develop an IP traceback approach that has advantages of both packet marking and packet logging. Our goal is to remain the ability to track a single packet as in hash-based IP traceback approach, but at the same time reduce the storage and access time overheads at routers with the help of packet marking. In our approach, each traceback-enabled router could commit both marking and logging operations on packets. The marking operation is to mark the packet with router identification information. The logging operation is to record the packet digest and the mark carried by the packet. The logging operation on a packet records not only the current router but also the upstream routers on the network path followed by the packet. In order to record the path of a packet, not all routers traversed by the packet need log the packet, only part of them need log the packet as long as the whole path can be derived from the logging information at those routers.

The contribution of our approach is (1) to reduce the storage overhead at routers to roughly one half, and (2) to reduce the access time requirement for recording packets by a factor of the number of neighbor routers. For each arriving packet, routers always commit marking operation, but commit logging operation when needed (generally alternately). The packet digest is stored in the same fashion as the hash-based approach. But the mark is stored in a space-efficient fashion so that the storage requirement for marks is negligible. Thus the storage overhead at routers is reduced to one half. Each router maintains a different digest table for each of its neighbor routers. Packets coming from different neighbor routers (with different marks) can be recorded in corresponding digest tables simultaneously. That reduces the access time requirement by a factor of the number of neighbor routers.

The rest of this paper is organized as follows. Section II puts our approach in the context of the related work. Section III describes our IP traceback approach in detail. Section IV analyzes the resource requirements and performance of our approach. Finally, we summarize our work in Section V.

II. Background and Related Work

Based on the vulnerability that is exploited, DoS attacks can be divided into *brute-force* and *semantic* attacks. Brute-force attacks work by flooding some limited resource with large amounts of traffic, thereby preventing legitimate users from accessing that resource. Semantic attacks exploit some specific feature or implementation bug of operating systems or routers to disable the services with one single or a few packets. An IP traceback approach that can track an individual packet is a must for defending against semantic DoS attacks.

In order to keep consistent with the literature, we term a packet of interest an *attack packet*. Similarly, the destination of an attack packet is a *victim*, the network path traversed by an attack packet is an *attack path*, and the output of IP traceback process is an *attack graph* composed of one or more possible attack paths for an attack packet.

The basic idea of IP traceback approach based on packet marking is that the router marks packets with its identification information as they pass through that router. The mark overloads a rarely used field in IP packet header, i.e., 16-bit IP identification field. The identification of a router could be 32-bit IP address [3], hash value of IP address [6], or uniquely assigned number [7]. In the last two cases, the length of identification information is variable and could be less than 16 bits. Since the marking space in packet header is too small to record the entire path, routers mark packets with some probability so that each marked packet carries the information of one node in the path. In addition, based on the length of router identification and the implementation of marking procedure, the router may only write part of its identification information into the marking space. While each marked packet represents only a small portion of the path it has traversed, the whole network path can be reconstructed by combining a modest number of such packets. This kind of approach is referred to as probabilistic packet marking (PPM) [3]. The PPM approach does not incur any storage overhead at routers and the marking procedure (a write and checksum update) can be easily and efficiently executed at current routers. But due to its probabilistic nature, it can only trace the traffic that consists of a large volume of packets.

The main idea of IP traceback approach based on packet logging is to log packets at each router through which they pass. In order to decrease the required storage space, logging should be done in an intelligent way. Hash-based IP traceback [4] stores packet digests, instead of packets themselves, in a space-efficient data structure, bloom filter. In this way, the storage overhead is reduced significantly. For each arriving packet, the router uses the first 24 invariant byte of the packet (20-byte IP header with 4 bytes masked out plus the first 8 bytes of payload) as input to the digesting function. The 32-bit packet digest is stored into the time-stamped digest table which is realized with bloom filter. The digest table is paged out before it becomes saturated, preventing unacceptable false-positive rates. Digest tables are archived for one minute for potential traceback operation. Each digest table is annotated with the time interval which the table covers, and hash functions used to compute packet digests over that interval. During the traceback process, routers are queried in the reverse-path flooding (RPF) manner and the digest tables at queried routers are examined to reconstruct the network path. This approach could track a single IP packet and therefore is considered to be more powerful compared to the PPM approach. Although the hash-based approach requires about 0.5% of the total link capacity in digest table storage, the storage requirement is prohibitive at routers with high speed links. In addition, packets must be recorded into the digest table at a rate commensurate with their arrival. At routers with high speed links, the access time requirement places limits on the type of memory used for digest table and the size of digest table. A small digest table can increase the chance of false positives in attack graph.

Two approaches have been recently proposed for addressing the deficiencies of hash-based IP traceback. T. Lee *et al.* [8] proposed to digest packet aggregation units (flow or source-destination set) instead of individual packets. Recording the digests of packet aggregation units reduces the digest table storage. However, tracing an individual packet back toward its source is actually accomplished by tracking the packet aggregation to which the packet belongs. That increases the false positives in constructing the attack graph. Moreover, depending on implementation, either the writing or reading rate of digest tables needs to be commensurate with packet arriving rate. The access time requirement is not alleviated. J. Li *et al.* [9] proposed to probabilistically select a small percentage of packets and record the digests of the selected packets. This method reduces both storage and access time overheads at routers. But the tradeoff is the loss of the ability to track individual packets since the probability that two adjacent routers on an attack path both record a specific packet is tiny. Our approach also addresses the deficiencies of hash-based IP traceback. But our approach not only reduces storage and access time overheads at routers but also remains the ability to track individual packets.

III. Hybrid IP Traceback

The hybrid IP traceback approach has a similar architecture with the hash-based approach [4]. Traceback-enabled routers audit traffic, and a traceback server (or multiple servers in hierarchy) which has the network topology information constructs attack graph by querying routers. The differences are at router operations on packets and the procedure of attack graph construction.

A. Router Operation

Each traceback-enabled router could commit both packet marking and packet logging operations. The marking operation on a packet is to mark the packet with router identification information. The logging operation on a packet is to record the packet digest and the mark (router identification) carried by the packet.

Every router is assigned an ID number of 15 bits in length. In hybrid IP traceback approach, the router ID number is used to differentiate neighbor routers of a router, instead of all routers within an ISP network. So the same ID number

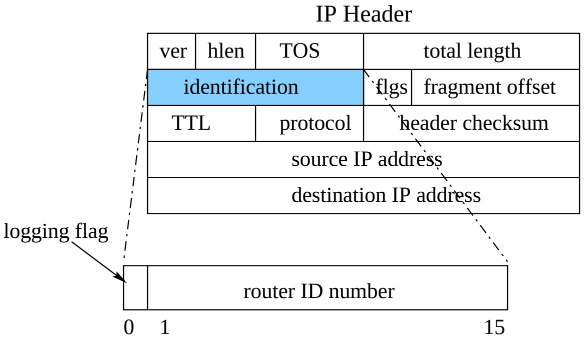

Fig. 1. Encoding mark into IP header

can be assigned to any two routers as long as they are more than 2 hops away.

The mark is stamped on packets through overloading the 16-bit identification field in IP header. The leftmost bit is a flag termed logging flag bit. It is set to 1 if the current router commits logging operation on the packet, otherwise set to 0. The remaining 15 bits is used to represent router identification. Figure 1 depicts the encoding scheme.

In the hybrid approach, routers record the router ID numbers carried by packets besides packet digests. The hybrid approach computes and stores packet digests using the same method as the hash-based approach. The storage of router ID numbers is implemented in a space-efficient fashion. Each router maintains a different digest table for each of its neighbor routers. When a router decides to commit logging operation on a packet, it examines the router ID number carried by the packet to get to know from which neighbor router the packet came, then stores the packet digest in the digest table corresponding to that neighbor. The digest table is paged out before being saturated. Each digest table is annotated with the time interval which the table covers, hash functions used to compute packet digests over that interval, and the neighbor router's ID number. Each digest table stores the digests of the packets which are forwarded by the same router and carry the same router ID number. The router ID number carried by packets is recorded as the annotation of digest table. In this way, the storage overhead for router ID numbers is negligible.

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. Figure 2 describes the full algorithm.

```
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
```

Fig. 2. Packet operating procedure at router R

The effectiveness of IP traceback increases greatly with widespread deployment of traceback-enabled routers in the network. However, it is likely that hybrid IP traceback approach does not require all routers to be traceback-enabled. All traceback-enabled routers form an overlay network. If the traceback server has the topology knowledge of that overlay network and each traceback-enabled router knows its neighboring traceback-enabled routers, hybrid IP traceback approach still works.

B. *Attack Graph Construction*

If a router commits logging operation on an attack packet, examining digest tables at that router will not only confirm that router is in the attack path, but also find out its upstream router in the attack path since each digest table is annotated with an upstream router's ID number. Given an attack packet and victim, the traceback server could infer the last hop router and whether the last hop router committed logging operation based on the logging flag bit carried by the attack packet.

1) If the traceback server infers a router logged the attack packet, examining the digest tables at that router would identify its upstream router in the attack path.
2) If the traceback server infers a router didn't log but marked the attack packet, querying the neighbor routers of that router in the RPF manner and examining the digest tables on these neighbor routers would identify the upstream router.

The attack graph can be constructed using those two methods alternately. Figure 3 shows how to construct attack graph.

C. *Transformation and Compatibility*

The lethal drawback of any packet marking based IP traceback is backwards compatibility [3]. Because the IP identification field designated for fragmentation is used for overloading marks, packet marking collides with fragmented IP traffic. Moreover, IP packets may undergo valid transformation while traversing the network [4].

With enhancements, hybrid IP traceback approach is able to trace packets that underwent transformation and avoid the backwards compatibility problem.

The router operations on packets are enhanced as below. For every arriving packet,

1) If the packet undergoes transformation at the current router, commit both marking and logging operations on

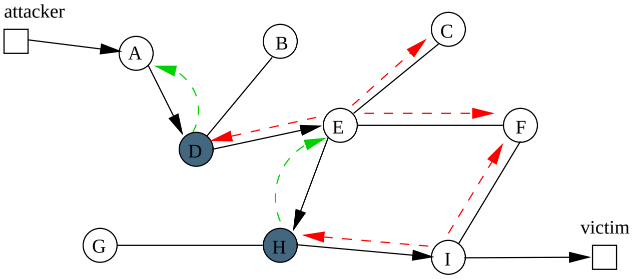

Fig. 3. Attack graph construction. Solid arrows represent the attack path; dashed curves represent the first method; dashed arrow represent the second method. Router D and H logged the attack packet.

the packet, and record the transformation information in the *transform lookup table*. Given a packet, consulting the transform lookup table can get to know whether the packet was transformed and the original packet can be reconstructed. The implementation of the transform lookup table is described in [4].

2) If the packet is a fragmented packet, compute and store the packet digest in a particular digest table which is only for fragmented packets and is managed in the same way as the hash-based approach.
3) Otherwise, follow the algorithm in Section III-A.

Attack graph construction is also improved accordingly. When the traceback server examines the digest tables at a router, it also consults the transform lookup table at that router and reconstruct the attack packet to its original form if possible.

1) If the attack packet provided by victim is not a fragmented packet, the traceback server employs the procedure to construct attack graph which is similar to the one presented in Section III-B. The only difference is that it is not any longer true that routers in an attack path log an attack packet alternately. It is possible that two adjacent routers, say m and n, both log a same packet p because the upstream router m commits logging operation on packet p, and p undergoes transformation at the downstream router n. During traceback process, when moving to the upstream router m from router n which logged and transformed packet p, the traceback server can not assume router m did not log packet p. The traceback server needs to examine the digest tables at m to figure out whether m marked p only or both marked and logged p, then takes proper action accordingly.
2) If the attack packet is a fragmented packet, starting from the last hop router, the traceback server queries routers in the RPF manner and examines the digest tables recording digests for fragmented packets to construct the attack graph.

IV. Analysis

Hybrid IP traceback approach introduces packet marking into hash-based IP traceback for the purpose of reducing the storage and access time overheads at routers. This section compares hybrid IP traceback approach and hash-based approach regarding overhead and performance.

A. Overheads on Routers

In the hybrid approach, when a packet is traversing the network, each router on the route commits marking operation on the packet, every other router commits logging operation. Keeping different digest table for each neighbor router makes the storage overhead for router ID numbers negligible. So the overall storage overhead at routers is reduced to roughly one half. In addition, since the router keeps separate table for each neighbor, packets coming from different neighbor routers can be recorded in corresponding digest tables simultaneously as long as each digest table has it own read/write hardware support. Thereby the access time requirement for recording packet digests is reduced by a factor of the number of neighbor routers. When taking into account packet transformation and backwards compatibility, the hybrid approach can handle those two issues in return for a modest increase in storage and access time requirements. But the percentage of IP traffic undergoing transformation and the percentage of IP fragmented traffic are small (3% [10] and 0.25% [11]), hence the increases of storage and access time requirements should be trivial.

B. Traceback Process Overhead

During the traceback process, the total number of the digest tables examined is an index reflecting the overhead on the traceback server and the speed of the traceback process.

Suppose time synchronization is maintained between adjacent routers, and each router has n neighbors on average. 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 $\frac{n}{2}$ and $\frac{1}{2}$, 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 ($n \geq 2$) neighbor routers on average, and the traffic load at the router is from each neighbors equally. Let the average time interval covered by one digest table in the hash-base approach be t_h, and the average time interval covered by one digest table in the hybrid approach be t_c. Then,

$$t_c = t_h \times n \,. \tag{1}$$

Suppose the attack path is m hops long from the attacker to the victim. Let the average link latency between routers be l. If 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. Suppose the average time interval covered by one digest table is t, then $\lceil \frac{l}{t} \rceil$ 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

$$N_h = m \times n \times \left\lceil \frac{l}{t_h} \right\rceil = m \times n \times \left\lceil \frac{l \times n}{t_c} \right\rceil \,. \tag{2}$$

In the hybrid approach, in order to move from the current router which marked attack packet to the upstream marking router which is 2 hops away, the digest tables at all interface of n neighbor routers need to be examined, there are n^2 interfaces totally (actually $(n-1)^2$). The number of digest tables examined is

$$N_c = \frac{m}{2} \times n^2 \times \left\lceil \frac{l}{t_c} \right\rceil . \tag{3}$$

(we assume m is even. When m is odd, $N_c = (\frac{m+1}{2} \times n^2) \times \lceil \frac{l}{t_c} \rceil$. We omit the odd case for simplicity.)

Hence the ratio of the number of digest tables examined in the hybrid approach to that in the hash-based approach during the traceback process is

$$r = \frac{N_c}{N_h} = \frac{n}{2} \times \frac{\left\lceil \frac{l}{t_c} \right\rceil}{\left\lceil \frac{l}{t_h} \right\rceil} . \tag{4}$$

When $l \le t_h$, we have $\lceil \frac{l}{t_c} \rceil = \lceil \frac{t}{t_h} \rceil = 1$. Then, $r = \frac{n}{2}$.

When $t_h < l < t_c$, we have $\lceil \frac{l}{t_c} \rceil = 1$ and $2 \le \lceil \frac{l}{t_h} \rceil \le n$. Then, $\frac{1}{2} \le r \le \frac{n}{4}$.

When $l = a \times t_c$ $(a = 1, 2, 3 \ldots)$, we have $\lceil \frac{l}{t_c} \rceil = a$ and $\lceil \frac{l}{t_h} \rceil = a \times n$. Then, $r = \frac{n}{2} \times \frac{1}{n} = \frac{1}{2}$.

When $l = a \times t_c + r$ $(a = 1, 2, 3 \ldots,\ 0 < r < t_c)$, we have $\lceil \frac{l}{t_c} \rceil = \lceil a + \frac{r}{t_c} \rceil = a + 1$ and $\lceil \frac{l}{t_h} \rceil = \lceil \frac{l \times n}{t_c} \rceil = a \times n + \lceil \frac{r \times n}{t_c} \rceil$. Because $1 \le \lceil \frac{r \times n}{t_c} \rceil \le n$, so $a \times n + 1 \le \lceil \frac{l}{t_h} \rceil \le (a+1) \times n$. Then,

$$\frac{1}{n} = \frac{a+1}{(a+1) \times n} \le \frac{\left\lceil \frac{l}{t_c} \right\rceil}{\left\lceil \frac{l}{t_h} \right\rceil} \le \frac{a+1}{a \times n + 1} < \frac{1}{n} + \frac{1}{a \times n + 1} < \frac{2}{n} .$$

So $\frac{1}{2} \le r < 1$.

In summary,

$$\frac{1}{2} \le r \le \frac{n}{2} . \tag{5}$$

According to the deduction above, when $l \ge t_c$, $r < 1$. That is, when the link latency between routers is large enough, less digest tables are examined in hybrid IP traceback approach than hash-based IP traceback during the traceback process.

If time synchronization is not maintained between adjacent routers, more digest tables need to be examined at each router. That is equivalent to the prior case with an increased link latency between routers. We could get a similar conclusion.

C. False Positive

The accuracy of the attack graph can be measured in the number of false positives in the graph. In the hash-based approach [4], the upper bound of the average number of additional spurious nodes in an attack graph is deduced according to

$$n \cdot \frac{dP}{(1 - dP)} ,$$

where n is the number node in the actual attack path, d represents the average degree of routers, and P denotes the false positive rate in examining digest tables at an individual router.

In the hybrid approach, using the same mathematical model, it is easy to show that the upper bound of the average number of false positives in an attack graph is

$$\frac{n}{2} \cdot \frac{2dP}{(1 - 2dP)} .$$

If we set $P = 1\%$, $n = 25$ (few paths in the Internet exceed this length [12]), and $d = 5$ (less than 5% of Internet routers have degrees of more than 5 [13]), then the hybrid approach results in no more than 2 additional nodes in expectation than the hash-based approach. The hybrid approach increases the false positives in attack graph, but the increase is small and acceptable.

V. Conclusion

In this paper, we proposed a new IP traceback approach which is based on both packet marking and packet logging. Our approach has the ability to track a single packet back to its origin. Compared to hash-based IP traceback approach, it reduces the storage overhead to roughly one half and improves on the access time by a factor of the number of neighbor routers. We also presented mathematical analysis comparing our hybrid IP traceback approach with hash-based IP traceback approach.

References

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

[2] G. Sager, “Security fun with OCxmon and cflowd,” in *Internet 2 working group meeting*, November 1998.

[3] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, “Network support for IP traceback,” *IEEE/ACM Transactions on Networking*, vol. 9, June 2001.

[4] A. Snoeren, C. Partridge, L. Sanchez, C. Jones, F. Tchakountio, B. Schwartz, S. Kent, and W. Strayer, “Single-packet IP traceback,” *IEEE/ACM Transactions on Networking*, vol. 10, December 2002.

[5] B. Bloom, “Space/time trade-offs in hash coding with allowable errors,” *Communications of ACM*, vol. 13, July 1970.

[6] D. Song and A. Perrig, “Advanced and authenticated marking schemes for IP traceback,” in *Proc. of IEEE INFOCOM*, April 2001.

[7] U. Tupakula and V. Varadharajan, “A practical method to counteract denial of service attacks,” in *Proc. of the 26th Australasian Computer Science Conference*, February 2003.

[8] T. Lee, W. Wu, and W. Huang, “Scalable packet digesting schemes for IP traceback,” in *Proc. of IEEE International Conference on Communications (ICC)*, June 2004.

[9] J. Li, M. Sung, J. Xu, L. Li, and Q. Zhao, “Large-scale IP traceback in high-speed Internet: Practical techniques and theoretical foundation,” in *Proc. of IEEE Symposium on Security and Privacy*, May 2004.

[10] S. McCreary and K. Claffy, “Trends in wide area IP traffic patterns: A view from Ames Internet exchange,” in *ITC Specialist Seminar on IP Traffic Modeling, Measurement and Management*, September 2000.

[11] I. Stoica and H. Zhang, “Providing guaranteed services without per flow management,” in *Proc. of ACM SIGCOMM*, August 1999.

[12] W. Theilmann and K. Rothermel, “Dynamic distance maps of the Internet,” in *Proc. of IEEE INFOCOM*, March 2000.

[13] CAIDA. http://www.caida.org/tools/measurement/iffinder.