

RIM: Router Interface Marking for IP Traceback

Ruiliang Chen^{*}, Jung-Min Park^{*}, and Randolph Marchany[†]

^{*}Laboratory for Advanced Research in Information Assurance and Security (ARIAS)

^{*}Bradley Department of Electrical and Computer Engineering

[†]Virginia Tech Information Technology Security Office

Virginia Polytechnic Institute and State University

{rlchen, jungmin, marchany}@vt.edu

Abstract—Distributed Denial-of-Service (DDoS) attacks have become a major threat to the Internet. As a countermeasure against DDoS attacks, IP traceback schemes identify the network paths the attack traffic traverses. This paper presents a novel IP traceback scheme called Router Interface Marking (RIM). In RIM, a router probabilistically marks packets with a router interface’s identifier. After collecting the packets marked by each router in an attack path, a victim machine can use the information in the marked packets to trace back to the attack source. Different from most existing IP traceback schemes, RIM marks packets with the information of router interfaces rather than that of router IP addresses. This difference endows RIM with several advantageous features, including fast traceback speed, last-hop traceback capability, small computation overhead, low occurrence of false positives, and enhanced security.

Index Terms—Distributed Denial-of-Service (DDoS) attacks, IP Traceback, Probabilistic Packet Marking.

I. INTRODUCTION

As the Internet’s scale and complexity continue to grow, the lack of security mechanisms during its early deployment years has led to serious problems today. In the past few years, many forms of Denial-of-Service (DoS) and malware attacks have been documented and brought to our attention through the news media. Among such attacks, Distributed DoS (DDoS) attacks are particularly menacing and very difficult to defend against. In a DDoS attack, an adversary gradually gains control over a large number of unsecured hosts, which are called *zombies*, as a prelude to the actual attack. The adversary then uses these zombies to launch a synchronized attack on a victim machine, overwhelming it with a deluge of packets.

DDoS attacks are hard to defend against due to two major reasons. First, in a DDoS attack, the number of zombie machines involved in an attack can reach several thousands or even more. Mitigating the effects of an attack of such a scale is a daunting task. Second, the adversary often forges IP source addresses (i.e., “IP spoofing”) to hide the true origin of an attack. Although ingress/egress filtering [9] is being deployed in many networks to prevent IP spoofing, their effectiveness is limited without wide deployment. Moreover, ingress/egress filtering does not prevent *subnet spoofing* (the IP spoofing of a random address from the address space assigned to a zombie machine’s subnet). For these and other reasons, traceback schemes are necessary to locate the attack sources (i.e., zombies).

To be practical and effective, an IP traceback scheme should possess several properties, including:

- *Fast convergence*: It should be able to execute traceback after

collecting a small number of attack packets;

- *Last-hop traceback*: It should be able to trace back to an actual zombie rather than merely to its edge router;
- *Minimal network and router overhead*: It should incur little increase in communication overhead to the network and impose minimal computation and storage overhead on the routers;
- *Scalability*: It should scale to a large number of attackers while incurring small numbers of false positives and false negatives;
- *Gradual deployment support*: It should work in the presence of legacy routers that do not support traceback; and
- *Marking field security*: It should provide mechanism to thwart marking field forgery.

Unfortunately, none of the existing IP traceback schemes [1, 2, 4, 5, 6, 11, 12, 14, 15, 16] meets all the above requirements.

In this paper, we propose a novel IP traceback scheme—*Router Interface Marking (RIM)*—that meets the above requirements. Unlike most of the existing schemes that treat a router as the atomic unit for traceback, RIM recognizes a *router interface* as the atomic unit for traceback. A RIM-enabled router probabilistically marks each packet with the *interface identifier (IID)* of one of the hardware input interfaces that processed the packet. A victim collects the packets marked by the RIM-enabled routers, and the information gathered from those packets is used to reconstruct their traversed paths. Our analysis and simulation results show that RIM has several advantageous features.

The remainder of this paper is organized as follows. Section II presents an overview of related work. The technical details of RIM are described in Section III. In Section IV, we analyze RIM’s performance in terms of the numbers of false negatives and false positives. Simulation results are presented in Section V, and we conclude the paper in Section VI.

II. RELATED RESEARCH

IP traceback schemes can be roughly divided into four categories: *ICMP traceback* [2], *probabilistic packet marking (PPM)* [4, 5, 12, 15, 16], *packet logging* [11, 14], and a *hybrid* of packet marking and packet logging [1, 6].

In ICMP traceback [2], an Internet router samples packets at a very low probability (e.g., 1/20,000) and sends an ICMP packet to the origin or the destination of a sampled packet. The ICMP packet contains the link information that a victim can collect to reconstruct an attack path. However, ICMP traceback has been criticized for inducing additional communication overhead to networks. Moreover, it needs a global key distribution infrastructure for authentication purposes, which is expensive to deploy and maintain.

In PPM schemes [4, 5, 12, 15, 16], routers mark each packet at a

pre-defined probability (typically 4%) with partial path information. Packets are marked by overloading infrequently used fields in the IP header, so that no additional communication overhead is induced. Typically the *Identification* and/or *ToS* (Type of Service) fields are used. By collecting the markings from a certain number of malicious packets, the victim can reconstruct the attack paths. Existing PPM schemes treat the router itself as the *atomic unit* of traceback. In contrast, RIM uses the router interface as the atomic unit of traceback. When a router is treated as the atomic unit of traceback, its identifier, which is an IP address, cannot be accommodated in the marking fields of a single packet. Therefore, multiple packets are required to encode marking information corresponding to a single router. The communication overhead needed to transmit the multiple packets and the computation cost required to decode the information contained in them are expensive. Because the identifier of a router interface is much shorter and can be held in the marking fields of a single packet, RIM can reduce the communication overhead and computation complexity. The identifier of a router interface can be used to identify the incoming link of a zombie's edge router, while a router's IP address can only represent the edge router itself. This property makes it possible to use RIM for last-hop traceback.

In packet logging schemes [11, 14], routers store packet digests in the form of Bloom filters. By checking neighboring routers iteratively with information from attack packets, the attack path of a flow can be reconstructed. The major problem with this technique is that they induce significant computation and storage overhead to routers. For instance, a router supporting the mechanism of [11] on an OC-192 link needs to compute 660,000 hash functions per second and store 216MB of data per hour, assuming the size of each packet to be 1,500 bytes. Under the same assumption, the mechanism in [14] requires a router to compute 2.5 million hash operations per second and store 1.875GB of data per hour.

The hybrid schemes [1, 6] utilize both packet marking and packet logging. Because both techniques are employed, the collection of packet markings can converge faster compared to pure PPM schemes, and the computation and storage overhead can be decreased compared to pure packet logging schemes. However, these hybrid schemes have their drawbacks, too. For example, the scheme proposed in [1] requires 34 bits in an IP packet for packet marking (which is still not practical [12]) and the mechanism in [6] consumes half of the overhead required by that in [14].

III. RIM: ROUTER INTERFACE MARKING

A. Assumptions and Overview

We assume the following network environment. Every host, whether a client or a server, is connected to its local *edge router*. Edge routers are interconnected by *core routers*. A zombie that is sending attack traffic is called *an attacker*, and the server being attacked is called the *victim*.

Recent studies showed that 95% of the observed routes had fewer than five observable daily changes [8]. Therefore, we make the reasonable assumption that every route from a client to a server has a *stable path* within the timeframe of interest. We will use the term *false negative* to denote an attacker that has escaped identification, and use the term *false positive* to represent a legitimate client that has been incorrectly identified as an attacker.

A RIM-enabled router should allocate an IID to each of its hardware interface, which is assigned in advance and unique to the router. The interface of the RIM-enabled router can mark an incoming packet with its IID and can execute XOR (exclusive OR) and increment operations on certain fields of the packet's IP header.

These functions are well within the capabilities of today's routers, considering the facts that: 1) a typical commercial router maintains a shadow copy of a routing table at each interface [10] (which means that the different interfaces can process packets independently) and 2) a router routinely updates the *TTL* (Time-To-Live) field and recomputes the *checksum* field.

B. Packet Marking Employed by RIM

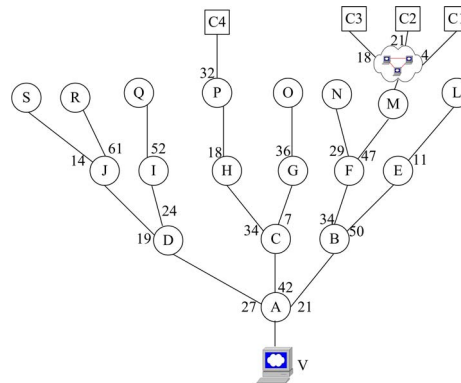


Fig. 1. The upstream tree of the victim V .

The conventional PPM schemes use IP addresses to identify a unique attack path, since an IP address is a globally unique identifier. RIM takes a fundamentally different approach. The basic principle of RIM is straightforward: in the upstream tree of a server, *a string composed of locally-unique router input IIDs is a globally unique identifier of a path*. “Locally-unique” means that an IID is unique within a router, but two interfaces of two different routers may have the same IID. Fig. 1 shows an example. In the figure, an upstream tree of server V is shown. The numbers on the links represent input IIDs. Suppose client $C3$'s path to server V is route $C3-M-F-B-A$, then the string of IIDs 18-47-34-21 (starting from $C3$) can be the unique identifier of $C3$ and its path. If $C3$ launches an attack on V , V can try to acquire the string and use it to trace back to the source of the attack. In practice, like the upstream interface of M , the FE (Fast Ethernet) and GE (Gigabit Ethernet) interfaces of a router can be connected to a switch, and thus to multiple hosts/routers through a LAN. This is the general scenario for most edge networks. To maintain the uniqueness of IIDs in such a case, we propose to map MAC addresses of connected devices to unique virtual IIDs. If a router interface hears frames coming from multiple MAC addresses, the router maps each address to a unique virtual IID and marks packets from different addresses with the respective virtual IIDs. With this approach, RIM is able to distinguish multiple hosts/routers that are connected to the same router interface, thereby supporting last-hop traceback.

RIM uses 17 bits in the IP header for packet marking. Specifically it uses the 16-bit *Identification* field and one more reserved bit immediately following the *Identification* field. There are schemes that also use the 8-bit *ToS* field, thus using a total of 25 bits [5]. In Subsection V.D, we investigate the relation between the number of bits used for packet marking and traceback performance. In RIM, we allocate six bits to an IID so that it can be any number from 0 to 63. Six bits are chosen because the vast majority of Internet routers have less than $2^6 = 64$ working interfaces¹ [13]. Besides the requirement of local uniqueness, an IID should also be

¹ In rare cases, a router may be connected to more than 64 devices. RIM addresses the problem by allowing a router to mark packets twice. For further details, please refer to our technical report [3].

allocated randomly to minimize collisions that may result in false positives. More discussions on this issue are given in Section IV. RIM uses the following marking fields: five bits for *Distance*, six bits for *XOR*, and six bits for *IID*. The *Distance* field is used for recording hop-count related information. Its length is five bits because the current Internet has few routes that span more than 32 hops [8].

In RIM, a router probabilistically marks a given packet with a probability of p . The marking decision is made independently for each packet at each router. A router marks a packet by: (1) resetting the *Distance* field to zero and (2) copying the IID of the packet's incoming interface to both the *IID* and the *XOR* fields. When a router does not mark a packet, which is with the probability of $(1 - p)$, it does the follows: (1) increases the *Distance* field by one and (2) computes the XOR of the packet's incoming IID and the value in the packet's *XOR* field, and writes the result back to the *XOR* field.

The information marked by an upstream router can be overwritten by downstream routers. When a victim receives a packet, its marking fields reveal three bits of information: (1) the hop count from the *nearest* upstream router that marked the packet to the victim; (2) the *IID* field value that was marked by that marking router; and (3) the *XOR* field value that was computed by taking the XOR over all the IIDs, starting from that marking router to V 's edge router. Hereafter, we say a victim's received packet is marked by a specific router, if the router is the nearest upstream router to the victim that marks the packet.

We illustrate our marking scheme with an example. Again, let us refer to $C3$'s path in Fig. 1. Suppose attacker $C3$ sends a packet to victim V and the packet is marked by M , i.e., none of F , B , and A marks the packet. So M will reset the *Distance* field to 0, and copy the packet's incoming interface's IID 18 (010010) to the *XOR* and *IID* fields. On the other hand, F , B , and A will increase the *Distance* field and update the *XOR* field. For example, F will increase the *Distance* field by one and compute the XOR of the packet's incoming interface's IID (47) and the packet's *XOR* field value and write the result in the *XOR* field. When V receives the packet, it reveals the following information: the *Distance* field is 3, the *XOR* field value is 02 ($=18 \oplus 47 \oplus 34 \oplus 21$), and the *IID* field is 18. Hence, V records the *XOR* and *IID* field values associated with hop count $d = 3$. We denote the three-tuple (d, IID, XOR) as a *record*. After a sufficient amount of time, V will eventually receive packets marked by other routers positioned along the attack path (such as F , B or A). Once all such packets are collected, V can organize the records got from these packets into a table that is sorted by the hop count. We call such a table a *trace table*. A trace table showing $C3$'s attack path is shown by the shaded records in Table 1.

A trace table contains records from multiple packets. Hence, V needs to group the records based on the paths they belong to, beginning with hop 0. The grouping is to verify whether any two records of consecutive hop counts satisfy the equation

$$XOR(d+1) \oplus IID(d+1) = XOR(d), \quad (1)$$

where the number inside each $()$ represents the hop count associated with the record. If the above equation is satisfied, then the two rows are grouped together and associated with the same path. This process is repeated for all the rows in a sequential manner. In the example, the four shaded records can be grouped to the same path. This means that the corresponding IID string 18-47-34-21, which represents $C3$'s attack path, can be recovered. The same technique can be used to differentiate multiple paths. Suppose that another attacker $C4$ sends packets to V via $C4-P-H-C-A-V$ in addition to $C3$. The resulting trace table contains

Table 1. A trace table containing the paths of $C3$ and $C4$.

The marking router closest to V^*	Hop Count: d	Interface Identifier: $IID(d)$	XOR: $XOR(d)$
A	0	21 [010101] [#]	21 [010101]
A	0	42 [101010]	42 [101010]
B	1	34 [100010]	55 [110111] ($\oplus 34=21$)
C	1	34 [100010]	08 [001000] ($\oplus 34=42$)
F	2	47 [100111]	16 [010000] ($\oplus 47=55$)
H	2	18 [010010]	26 [011010] ($\oplus 18=08$)
M	3	18 [010010]	02 [000010] ($\oplus 18=16$)
P	3	32 [100000]	58 [111010] ($\oplus 32=26$)

* This column is for illustration purposes. The column's information is not revealed by the packets collected.

The binary strings inside [...] represent the binary equivalents.

all records in Table 1. What V first notices is that there are two records with hop count of 0, which implies that there are at least two attack paths. The two attack paths can be separated by iteratively verifying Eq. (1) starting with $d = 0$ and increasing d by one for each iteration. The verification is done for every row. The string of *IID* values of the records that satisfy (1) is the desired IID string. In Table 1, the four shaded records represent $C3$'s path 18-47-34-21 while the non-shaded records represent $C4$'s path 32-18-34-42.

Once an IID string is obtained, a victim can get its corresponding IP addresses in two ways. It can either query upstream routers iteratively (detailed in [11, 14]) or utilize *a priori* knowledge of an Internet map (discussed in [15, 16]).

C. Security of Marking Fields

In this subsection, we discuss the problem of defending against marking field forgeries, which is a critical security problem shared by all PPM schemes. In such an attack, attackers arbitrarily write forged traceback information to the marking fields. Because each router marks packets probabilistically, some forged packet markings will not be overwritten. They could eventually arrive at the victim and disrupt path reconstruction. Although an attacker cannot forge the marking fields along its path to the victim because the *Distance* field is increased strictly by every non-marking router, the attacker may forge any marking fields with the *Distance* field longer than its hop count from the victim. To counter such an attack, an authentication scheme using time-released key chains is proposed in [15]. However, the scheme requires a global key-distribution infrastructure, which is costly to deploy and maintain.

RIM can solve this problem with an alternative approach. The conventional packet marking schemes mark packets with different destinations in the same manner. Our approach requires that a router dynamically allocate different IIDs to its interfaces for different destination addresses in the marked packets. For example, suppose a router has two interfaces A and B . When it marks a packet, it can use $H(pkt.dest)$ as A 's IID and use $[(H(pkt.dest) + offset) \bmod 64]$ as B 's IID. Here $H(pkt.dest)$ is a hash function of the packet's destination address with a 6-bit output and $offset$ is a number between 1 and 63. What hash function and which offset value to use are decided independently by each router. A more sophisticated version would be to vary the hash function with time so that it is even harder for malicious hosts to guess a valid IID.

Findings from an Internet topology study [13] showed that the average router degree is 6.34. Hence, a 6-bit IID and a 6-bit XOR arbitrarily guessed by an adversary are valid with an average

probability of $0.00155 (= 6.34/2^{12})$. If a path includes m hops that are not part of any attack path, then an attempt to guess a valid IID string is successful with a probability of 0.00155^m , which is a very small value when m is large. Moreover, if the attacker is far away from the victim, the forged or guessed markings will be likely overwritten by intermediate routers.

IV. ANALYSIS OF FALSE NEGATIVES AND FALSE POSITIVES

We first show that RIM incurs *no false negatives* if all of the attack packets have been collected. Suppose that the attacker is d hops away from the victim's edge router and that the string $IID(0), \dots, IID(d-1)$ represents the string of interfaces that processed the attack traffic. Also, suppose that $XOR(i)$ denotes the XOR field of a packet marked by a router i ($i = 0, \dots, d-1$) hops away from the victim's edge router. Then, a trace table should contain the following records:

$$\{0, IID(0), XOR(0)\}, \dots, \{d-1, IID(d-1), XOR(d-1)\},$$

where $XOR(i) = IID(0) \oplus IID(1) \oplus \dots \oplus IID(i)$. Since

$$\begin{aligned} XOR(i) \oplus IID(i) &= IID(0) \oplus \dots \oplus IID(i) \oplus IID(i) \\ &= IID(0) \oplus IID(1) \oplus \dots \oplus IID(i-1) = XOR(i-1) \end{aligned}$$

is equivalent to Eq. (1), which is used to associate records to their corresponding paths, we see that $IID(0), \dots, IID(d-1)$ must be the router IIDs along a single path.

RIM incurs *false positives* because the IID and XOR fields of the packets coming from a legitimate client may coincide with those of the packets coming from a zombie. For example, suppose a legitimate client's path expressed as an IID string is $P_{10}-P_{11}-P_{12}$, it can become a false positive when substring $P_{10}-P_{11}$ is a substring of a zombie's path and there is another zombie's path $P_{z0}-P_{z1}-P_{z2}$, which satisfies: (1) $P_{z2} = P_{12}$ and (2) $P_{10} \oplus P_{11} = P_{z0} \oplus P_{z1}$. The first condition leads to a collision of the IID fields at hop 2 and the second condition results in a collision of the XOR fields at hop 2. Generalizing the above argument, a false positive occurs when each hop of a legitimate client's path (that is not shared by any attack paths) collides with a zombie's attack path. Here, a collision means that two different packets traversing two different paths induce the same XOR and PID values at the considered hop in a trace table. It can be shown that allocating PIDs randomly minimizes the chance of collision. Hence, we assume that each router assigns IID values to its interfaces randomly from 0 to 63. If we assume that N_a zombies are being traced back and assign the constant value $S_b = 2^{12}$, then the probability that a link of a legitimate client's path (that is not on any zombie's path) collides with a zombie's path is:

$$P_r = \frac{\sum_{k=1}^{\min(N_a, S_b)} k \binom{S_b}{k} \left(\frac{1}{S_b}\right)^{N_a} M(N_a, k)}{S_b}, \quad (2)$$

where

$$M(N_a, k) = \begin{cases} 1 & (k=1, N_a \geq k) \\ k^{N_a} - \sum_{j=1}^{k-1} \binom{k}{j} M(j) & (2 \leq k \leq S_b, N_a \geq k) \end{cases}. \quad (3)$$

In (2), P_r can be understood as the expected ratio of the S_b combinations that is covered by N_a independent random 12-bit numbers. If a legitimate client's path has m links that are not on any zombie's path, then its false positive probability is:

$$P_{FP} = P_r^m. \quad (4)$$

V. SIMULATION RESULTS

A. The Topology Model for Simulation

We adopt two real network topologies for our simulations. The first topology is chosen from the Skitter Internet map [13] on April 21st, 2003. We first chose a router with a degree of six as the victim's edge router, and then randomly chose multiple distinct routes originating from it. The second topology is selected from the network topology data of Lumeta's Internet Mapping Project [7]. The provided path data was collected on October 19th, 2003. All paths start from a single router, which is the victim's edge router in our simulation. This edge router has a degree of two.

B. The Performance of RIM

An important performance indicator for IP traceback is the number of packets that need to be collected for reconstructing attack paths. Fig. 2 shows the 95th percentiles for the number of packets required to reconstruct attack paths when marking probability p is set to 0.04. To better evaluate RIM's effectiveness, we repeat the same experiments with FMS [12], AMS [15], and FIT [16]. AMS and FIT can be configured in more than one way. The curves of AMS and FIT in Fig. 2 represent the suggested configurations that require the minimal number of packets. Every point in the plots represents the 95th percentile value out of 1000 independent experiments. The result shows that the number of packets needed by RIM is the smallest among all PPM schemes. This is expected since RIM needs to collect only one marked packet from each router along an attack path, whereas other schemes need multiple marked packets.

We are also concerned about the relation between the false positive ratio and the number of attackers. Figs. 3 and 4 show the false positive ratios versus the number of attackers on two maps. The false positive ratio is the number of legitimate clients that are mistakenly recognized as attackers by RIM divided by the number of legitimate clients. In each simulation, we fix the total number of clients to 5000 and vary the number of randomly chosen attackers from 1 to 4101 in increments of 10. All data shown in the figures are the average of three independent experiments. The results show that RIM incurs relatively low false positive ratios (which could be lowered even further by allocating more marking bits, as will be discussed in Subsection V.D). It is interesting to note that the two maps have quite different false positive ratios. The reason is that in the Skitter map, the victim's edge router has a degree of six, while in the Lumeta's map the degree is two. Because a larger degree means more path possibilities and less path collision probabilities, the Skitter map, compared to the Lumeta's map, induces a larger value of m in Eq. (4) and a smaller P_{FP} value. This result shows that making a victim network multihomed can decrease RIM's false positive ratio.

C. The Impact of Legacy Routers

We evaluated RIM when a portion of the routers are legacy routers which do not support RIM². Figs. 5 and 6 show the simulation results on both maps. In the simulations, we assumed that the victim's edge router is RIM-enabled and other RIM-enabled routers are uniformly randomly distributed in the network. We varied the percentage of RIM-enabled routers from 20% to 100% in increments of 20%. The total number of clients was fixed at 2000, out of which the number of attackers was varied

² It is noted that if *a priori* knowledge of Internet map is used for converting IID strings to IP addresses, there is no additional requirement for supporting gradual deployment. If we choose to query upstream routers, then we need RIM-enabled routers to support neighbor-discovery handshake protocol [3].

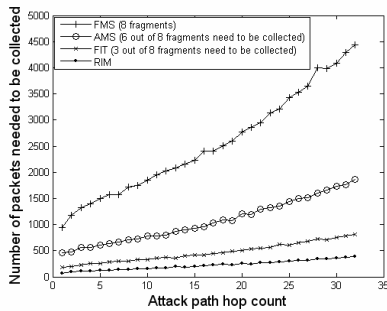


Fig. 2. Number of packets required for path reconstruction ($p = 4\%$).

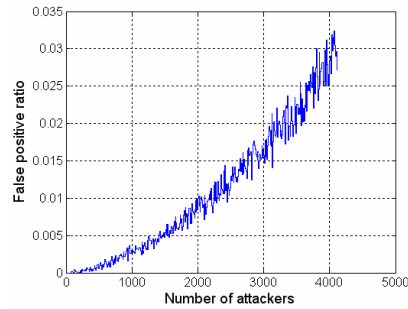


Fig. 3. False positive ratio in Skitter map.

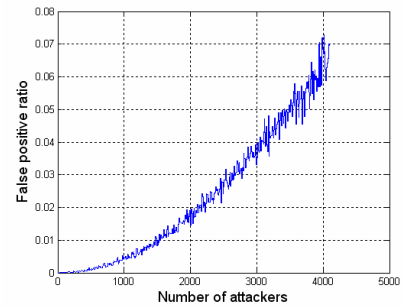


Fig. 4. False positive ratio in Lumeta's map.

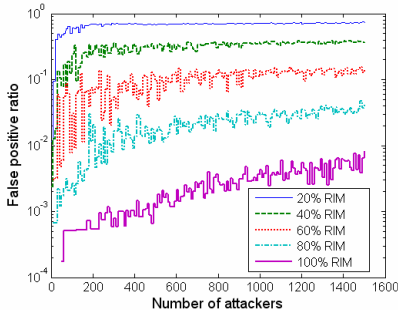


Fig. 5. False positive ratio in Skitter map with partial deployment.

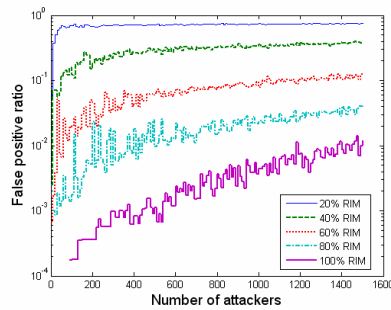


Fig. 6. False positive ratio in Lumeta's map with partial deployment.

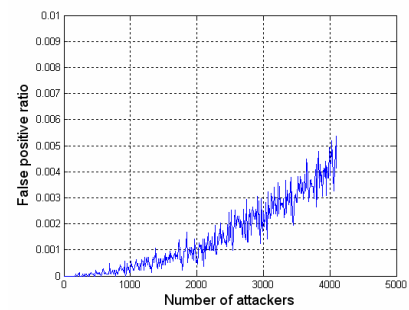


Fig. 7. False positive ratio in Lumeta's map with two more marking bits.

from 1 to 1501 in increments of 10. Each point in the curves is the average of three independent experiments. The results show that RIM performs relatively well even when only a fraction of the routers are RIM-enabled. When the ratio of RIM-enabled routers to legacy routers is 60/40, the false positive ratios are 12% for both maps when there are 1500 attackers.

D. The Impact of More Bits for Packet Marking

We have assumed that RIM assigns 17 bits for packet marking. However, as mentioned in Section III, it is possible to allocate more bits. In this subsection, we assume that RIM uses two more bits (e.g., by using bits from the *ToS* field). We repeated previous simulations on the Lumeta's map with one more bit allocated for the *IID* field and one more bit allocated for the *XOR* field. Fig. 7 shows the simulation result. Compared to Fig. 4, the result shows a significant decrease of the false positive ratio. Based on this fact, we can conclude that the number of false positives can be reduced substantially by allocating a few more bits for packet marking.

VI. CONCLUSION

We presented a novel PPM scheme for IP traceback called RIM. The principal characteristic of RIM that distinguishes it from existing PPM schemes is that it treats a router's interface—rather than the router itself—as the atomic unit for traceback. This characteristic endows RIM with several advantageous features, including: (1) RIM requires to collect a relatively small number of marked packets to execute traceback; (2) RIM supports last-hop traceback; (3) RIM incurs no false negatives (when all packets are collected) and a relatively small number of false positives even when faced with a large number of attackers; and (4) it is possible to incorporate into RIM new mechanisms for countering marking field forgeries.

REFERENCES

[1] D. Basheer and G. Manimaran, "Novel hybrid schemes employing packet

marking and logging for IP traceback," *IEEE Trans. Parallel and Distributed Systems*, Vol. 17(5), May 2006, pp. 403–418.

[2] S. Bellovin, M. Leech, and T. Taylor, *ICMP Traceback Messages*, Internet Draft, draft-ietf-itrace-04.txt, Feb. 2003.

[3] R. Chen, J.-M. Park, and R. Marchany, "TRACK: A Novel Approach for Defending against Distributed Denial-of-Service Attacks," *Technical Report TR-ECE-06-02*, Dept. of ECE, Virginia Tech, Feb. 2006, available at: <http://www.arias.ece.vt.edu/publications/TechReports/Chen-2006-2.pdf>.

[4] D. Dean, M. Franklin, and A. Stubblefield, "An Algebraic Approach to IP Traceback," *ACM Trans. Information and System Security (TISSEC)*, Vol. 5(2), May 2002, pp. 119–137.

[5] M. T. Goodrich, "Efficient Packet Marking for Large Scale IP Traceback," in *Proc. CCS*, Nov. 2002, pp. 117–126.

[6] C. Gong and K. Sarac, "IP traceback Based on Packet Marking and Logging," in *Proc. ICC*, May 2005, pp. 1043–1047.

[7] The Internet Mapping Project, available at: <http://research.lumeta.com/ches/map/>.

[8] C. Jin, H. Wang, and K. G. Shin, "Hop-Count Filtering: An Effective Defense against Spoofed DoS Traffic," in *Proc. CCS*, Oct. 2003, pp. 30–41.

[9] T. Killalea, *Recommended Internet Service Provider Security Services and Procedures*, RFC 3013, Nov. 2000.

[10] J. Kurose and K. Ross, *Computer Networking: a Top-down Approach Featuring the Internet, Third Edition*, Addison Wesley, 2004.

[11] J. Li, M. Sung, J. Xu, and L. Li, "Large-scale IP Traceback in High-speed Internet: Practical Techniques and Theoretical Foundation," in *Proc. IEEE Symposium on Security and Privacy*, May 2004, pp. 115–129.

[12] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Practical Network Support for IP Traceback," in *Proc. SIGCOMM*, Aug. 2000, pp. 295–306.

[13] CAIDA's Skitter project web page, available at: <http://www.caida.org/tools/measurement/skitter/index.xml>.

[14] A. Snoeren, C. Partridge, L. Sanchez, C. Jones, F. Tchakountios, S. Kent, and W. Strayer, "Hash-based IP Traceback," in *Proc. SIGCOMM*, Aug. 2001, pp. 3–14.

[15] D. X. Song and A. Perrig, "Advanced and authenticated marking schemes for IP traceback," in *Proc. INFOCOM*, Apr. 2001, pp. 878–886.

[16] A. Yaar, A. Perrig, and D. Song, "FIT: Fast Internet Traceback," in *Proc. INFOCOM*, Mar. 2005, pp. 1395–1406.