TOWARDS AN ADAPTIVE PACKET MARKING SCHEME FOR IP TRACEBACK

Ping Yan, Moon Chuen Lee
Department of Computer Science and Engineering, The Chinese University of Hong Kong, CUHK, Hong Kong

Keywords: DoS attacks, IP traceback, probabilistic packet marking, inter-domain marking, source router marking, attack graph reconstruction.

Abstract: Denial of Service attacks have become one of the most serious threats to the Internet community. An effective means to defend against such attacks is to locate the attack source(s) and to isolate it from the rest of the network. This paper proposes an adaptive packet marking scheme for IP traceback, which supports two types of marking, namely source router *id* marking and domain *id* marking. For each packet traversing, we let the border routers perform probabilistic router *id* marking if this packet enters the network for the first time, or perform probabilistic domain *id* marking if the packet is forwarded from another domain. After collecting sufficient packets, the victim reconstructs the attack graph, by which we keep track of the intermediate domains traversed by attack packets instead of individual routers within a domain; however, the source routers serving as ingress points of attack traffic are identified at the same time. Simulation results show that the proposed marking scheme outperforms other IP traceback methods as it requires fewer packets for attack paths reconstruction, and can handle large number of attack sources effectively; and the false positives produced are significantly low. Further, it does not generate additional traffic.

1 INTRODUCTION

The intent of *Denial of Service* (DoS) attacks is to prevent or impair the legitimate use of computer or network resources. Internet connected systems face a consistent and real threat from DoS attacks, because the Internet fundamentally composed of limited and consumable resources like bandwidth, processing power, and storage capacities is rather vulnerable to some level of service disruption (Kevin J. Houle et al., 2001). In case of *Distributed Denial of Service* (DDoS), an attacker first compromises a bunch of hosts weakly secured or possessing vulnerable network service programs, and he then uses these compromised computers to launch coordinated attacks on victim machines.

The primary difficulty of dealing with (D)DoS attack is *IP Spoofing*, which is almost always present in such attacks. In order to prolong the effectiveness of the attack, the attackers spoof the source IP addresses in their attacking packets to avoid being traced. Therefore, in a *traceback* problem, our task is to find out the actual source(s) of the attack, where we define the *source* as the router directly connected to the system from which the flow of packets, constituting the attack, was initiate (Hal Burch, 1999) (Steven H. Bass, 2001). Upon identifying the attack source(s), the victim or the network operators can conduct efficient defenses against DoS or DDoS attacks, either by blocking the traffic from the identified sources or filtering out the malicious packets on their way to the victim.

1.1 Problem Model and Performance Metrics

We would model the attack with a number of coordinated attackers attacking a single victim as an undirected graph with each node representing a domain. Domain is a logical subnet[1] on the Internet; a campus or internal corporate network is an example of a domain. Data exchange between campus and corporate domains is facilitated by one or more ISP domains, which offer, as a service, transmission and switching facilities for data exchange between their customers. The IP packets thus flow though differ-

[1] Subnet is a portion of a network, which may be a physically independent network segment, which shares a network address with other portions of the network and is distinguished by a subnet number (Tanenbaum, 2002).

Yan P. and Chuen Lee M. (2004).
TOWARDS AN ADAPTIVE PACKET MARKING SCHEME FOR IP TRACEBACK .
In *Proceedings of the First International Conference on E-Business and Telecommunication Networks*, pages 119-126
DOI: 10.5220/0001394701190126

ent network domains, from regional ISP network to international ISP network and finally get to the destination. In general, to model the attack, a network domain can be thought of as a cloud, which connects to other domains at the peering points, with clients attaching on border routers. In our solution, the reconstructed *attack graph* would incorporate attack paths and the source router(s) identified, with each node on the paths can be viewed as a domain.

In the literature, the performance of IP traceback approaches is commonly measured by several parameters. *Minimum number of packets* is the number of packets required for attack graph reconstruction; it is desired to be minimized to achieve a fast response to an attack and diminish the damage (Savage et al., 2000) (Kuznetsov et al., 2002). A *false positive* is a router that is actually not on an attack path but is reconstructed to an attack graph by a traceback mechanism, and a *false negative* is a router that is missed in the reconstructed attack graph (Savage et al., 2000) (Kuznetsov et al., 2002). Furthermore, an efficient traceback approach should feature a relatively low *computation complexity* and incremental deployment into the current Internet structure, at low cost (Kuznetsov et al., 2002).

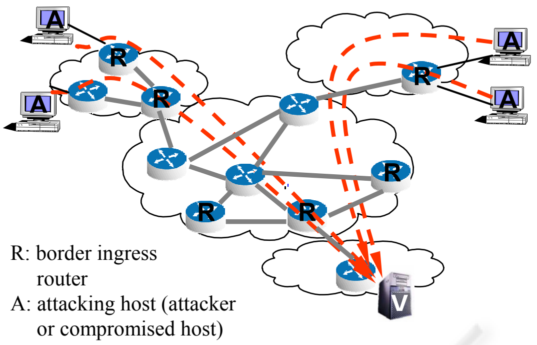
Our proposed method will not incur network traffic overload or storage overhead on the participating routers, though certain memory is required at the victim site. Therefore, we would assess the proposed method mainly based on the above parameters and the simulation results are demonstrated in section 4.

1.2 Overview Of the Proposed Method and Contributions

In this paper, we present a practical IP traceback approach. It addresses the issues concerned by both the victims and network operators such as per-packet marking space limitation, network overload and computation overhead.

In our proposed marking algorithms, we employ 25 bits space in the IPv4 packet header as marking fields. Probabilistically, each participating router adaptively inscribes onto a traversing packet with its local partial path information. There are two types of markings: router identification (*rid*) marking and domain identification (*did*) marking. The *rid* marking is executed if the packet enters the network for the first time; in contrast, *did* marking is performed when the packet traverses along the following domains towards the victim.

The victim under a DDoS attack reconstructs the attack graph in two phases. First, it identifies all the intermediate domains taking part in forwarding the attack packets, and recovers the inter-domain attack paths by inspecting the domain *ids* marked in the received packets. Second, from the router *id* markings in the received packets, the victim can identify the source routers. In general, the inter-domain attack paths reconstruction leads to the identification of the *source domains* (where the source router may reside), and then the identified source domains are associated with the router *id* markings to uncover the source routers as we wish.



R: border ingress router
A: attacking host (attacker or compromised host)
V: victim host

Figure 1: Adaptive traceback mechanism equipped network

This work presents a novel design of probabilistic packet marking scheme at the granularity of domain, while at the same time, the attack sources can be identified. And this inter-domain IP traceback design has been proved to possess the following advantages. First, we use much less number of packets to identify the attack source(s). In particular, it requires only two uniquely marked attack packets to identify a source router. Second, this approach generates quite low false rates which will be demonstrated by the simulation results. Third, our traceback mechanism ensures incremental deployment and requires fewer routers to participate. Actually, only the border routers[2] need to take part in this traceback mechanism. Figure 1 shows the example network with the implementation of our traceback mechanism.

Furthermore, keeping track of the domains traversed instead of intermediate routers on the attack paths and using *ids* instead of full 32-bit IP addresses have some underlying advantages. First, a domain is an administrative unit on the Internet, which has the capability to conduct defenses against the attacks when it is identified to be involved in an attack and notified by the victim. Second, it still makes sense to identify the source domains even the individual source routers are not identified correctly. We observe that systems of some domains are more likely to be compromised (to launch a DDoS attack,

[2] A router that sits on the border of a network connecting it to an end-host or a router in another network.

attackers usually compromise a bunch of vulnerable systems as attack agents) due to the domains' poor security features such as weak intrusion defense mechanisms and flawed security policies.

1.3 Organization Of the Paper

In the second section, we survey the previous work on traceback problem. In section 3, we present our proposed traceback scheme in depth and articulate the operation of our traceback mechanism. At last, we give the simulation results in section 4. Section 5 concludes the paper.

2 RELATED WORK

2.1 Tracing Hop-by-hop

J. Ioannidis and S. M. Bellovin (Ioannidis et al., 2002) proposed a *Pushback mechanism*. In this approach, a congested router nearest to the victim uses statistics and pattern analysis to determine from which most adjacent upstream routers the unexpected traffic volume are coming, and then send signals to notify the traffic contributors to rate-limit the suspect traffic. The approach is then repeated at the upstream routers in a chain to identify and rate-limit the traffic contributors. This scheme therefore requires immediate action during the attack, and requires considerable coordination between network operators. The main drawback with this method is that, in large-scale DDoS attacks, they have limited capabilities to separate the legitimate packets from attack packets in a pattern-based way.

2.2 ICMP Traceback Messaging (iTrace)

S. M. Bellovin (Bellovin, 2000) proposed an alternative approach, *ICMP traceback messaging* (or simply iTrace). With some probability q (typically, q = 1/20000 is proposed), each router sends an additional ICMP message packet to the destination for each packet it received. The message contains information of the local router traversed and its adjacent hop. With sufficient ICMP traceback messages from routers along the path, the attack source(s) and paths can be determined at the victim site. The main drawback of this approach is that it causes additional network traffic even when no attack is present. Consequently, q should be small enough to imply a relatively low network traffic overload. However, using a small q, this approach is inefficient in terms of the number of ICMP traceback packets required. For example, if the maximum path length is 20 and there are about 1000 nodes on the reconstructed attack graph, the expected number of attack packets required to arrive at the victim to reconstruct the attack graph is 7.5 million (Goodrich, 2002).

2.3 Logging & Querying

In a logging solution, we let the routers log the packets they process, and a victim then actively queries the routers to see whether they sent suspect attack packets. In general, this approach is infeasible because of the huge storage requirement at the routers (Ioannidis et al., 2002). However, *Source Path Isolation Engine* (SPIE) has the capability of identifying the source of a particular IP packet given a copy of the packet to be traced, its destination, and an approximate time or receipt (Snoeren et al., 2002). Most notably, with the use of an innovative logging technique, collecting only the hashes of the packets, this approach reduces the memory requirement down to 0.5% of link bandwidth per unit time (Snoeren et al., 2002). However, though the storage requirement has been significantly reduced, the overhead is still considerable.

2.4 Probabilistic Packet Marking

To avoid the network overloading, some researchers propose to embed traceback information in the IP packets, which is commonly referred to as *probabilistic packet marking* (or simply PPM) method. Savage *et al.* (2000) proposed to let each router mark each packet it forwards with a piece of partial path information at a set probability p (e.g., p = 1/20). A message "edge" recording the identities of a router and its previous hop would be inscribed onto certain bits employed as marking fields in the IP header. However, the edge message has to be made to fit in the limited reserved bits; so they break it into fragments sent by separate packets. To reconstruct the attack paths, every possible fragments combination is tried to form a valid edge, and then the edge is used to recover the sequence of intermediate routers hop by hop at the victim site. Unfortunately, for even small-scale distributed DoS attacks, this method is not practical due to the tremendous combinatorial trials and the high false rates. Even worse, it would introduce many false positives because the previous mis-reconstructed messages lead to more false combinatorial trials, which can be described as "explosion effect".

In Advanced and Authenticated Marking Schemes, Song and Perrig (2001) proposed the use of hash chains for authenticating routers to improve the performance of probabilistic packet marking. They do not fragment router messages. Instead, they assume the victim knows the map of its upstream routers, so the full IP address is encoded into 11 bits hash values by two sets of universal random hash functions in the packet marking. To reconstruct the attack graph, the victim uses the upstream router map as a road-map and performs a breadth-first search from the victim to identify the corresponding router which was hashed and written into the marking fields.

3 ADAPTIVE PACKET MARKING SCHEME

Our adaptive packet marking scheme is based on the probabilistic packet marking technique, but a novel IP packet marking scheme is proposed, which is motivated by the below issues.

3.1 Design Motivation

The IP traceback approaches, such as iTrace or the proposed probabilistic packet marking schemes, rely on observing a high volume of spoofed traffic comprised of thousands or millions of packets, so the attacker can undermine the traceback by spreading the attack traffic across many attacking hosts (also referred to as agents, slaves, or reflectors in a reflector DDoS attack (Chang, 2002)), greatly increasing the amount of time required by the traceback scheme to gather sufficient packets to analyze. Therefore, an effective traceback scheme should use as few packets as possible to reveal an attack path. Using a relatively short *id* instead of a full IP address, we do not need to spread a mark across multiple packets, and we thus feature a relatively small number of packets to fulfill the traceback.

In addition, some people are challenging the necessity of the full-path traceback solution (Belenky et al., 2003); identifying all the intermediate routers that the attack packets traversed, may be unattractive to the victims and ineffective for DoS (DDoS) countermeasures. First, the full-path traceback is as good as the address of an ingress point in terms of identifying the attacker. Second, each packet in a datagram network is individually routed so packets may take different routes even if their source and destination are identical. Third, the addressing within ISPs' networks is not necessarily understandable to the public since ISP may use private addressing plans within their own networks (Belenky et al., 2003). Therefore, we propose a domain based IP packet marking scheme to identify the intermediate domains instead of the individual routers, except the one serving as the attack source. In the following paragraphs, we will describe the proposed scheme in depth and state how this method addresses the problems with the existing solutions.

3.2 Using ID for Marking

The proposed marking scheme overloads 25 bits space in IPv4 header; the 25 bits space consists of the 16-bit Fragment Identification field, 1-bit fragmentation flag and 8-bit Type of Service (ToS) field. Employing the 25 bits in the IP header for marking was first advocated by Dean *et al*. (2001). The ToS field is currently not set except for extreme unusual cases. The Fragment ID field is a 16-bit field used by IP to permit reconstruction of fragments; this field is commonly used as a marking field and the backward compatibility is fully discussed in Savage's paper (Savage et al., 2000). The fragmentation flag is an unused bit that current Internet standards require to be zero. We also see there are some proposals on marking in the IPv6 header; however, it is not to be discussed in this work.

As every host or router on the Internet is identified using a 32-bit IP address (Tanenbaum, 2002), it is a challenging issue to overload the 25-bit marking space in the IP header with a 32-bit IP address. In our proposal, since we only aim to identify the intermediate domains taking part in the attack and the source routers, there is no need to use full IP addresses, as long as we can uniquely identify each domain with a different identification. If we assign a 16-bit domain *id* to each domain, we can uniquely identify up to 2^{16} (65536) domains. If we assume there are at most 2^{10}(1024) border routers within a domain, a 10-bit value is sufficient to be assigned as a router *id* to identify the source routers within a source domain. However, to defend against the attack, the victim may demand to block the malicious traffic at the source routers, so the victim needs to retrieve the IP addresses from the *ids*. This could be implemented as an ID-to-IP mapping table published on websites, or it could be maintained at the victims individually.

Two types of markings, either the router *id* marking or the domain *id* marking, are to be performed by a router adaptively by checking whether the router concerned is the ingress point of the to-be-marked packet or not. At first, however, the border routers, with implementation of our marking scheme should be capable of determining which type of marking to perform. Physically, these routers are connected to end-hosts or other routers through different interfaces; a router therefore checks through which interface it receives a packet to see

flag = 1	dist	edge
1 bit	5 bits	16 bits

(a) domain *id* marking

flag = 0	cord	rid
1 bit	14 bits	10 bits

(b) router *id* marking

Figure 2: Marking fields

whether a packet is forwarded by another router from outside the domain concerned or sent by an end-host at the customer side. The domain *id* marking would be performed if the router receives a packet routed from outside the current domain, and the router *id* marking would be performed when the packet comes from an end-host. Figure 2 shows the marking fields for domain *id* marking and router *id* marking respectively.

3.3 Domain id Marking

The domain *id* marking algorithm allows the victim to infer the inter-domain attack paths by inspecting the domain *ids* in the received packets. As shown by Figure 2 (a), "edge" field stores one encoded edge on an attack path, and the 5-bit distance field represents the number of hops traversed since the edge it contains is sampled. A flag is used to indicate whether this is a domain *id* marking or a router *id* marking. Basically, the domain *ids* of two neighboring domains are encoded by *exclusive-or* (XOR) to make up the edge, and it can be decoded back during reconstruction in virtue of XOR's property that $\alpha \oplus \beta \oplus \alpha = \beta$. This XOR encoding technique is used to reduce per-packet storage requirement.

Figure 3 shows the domain *id* marking scheme. Marking probability *p* determines whether to mark a packet or not. To mark the packet, router *R* sets the distance to be zero and writes the domain *id* into the edge field. Otherwise, if the distance is zero, router *R* overwrites the edge field with XOR of edge value present in the to-be-marked packet and its own *id*. The distance field is used as hop counts, and is always incremented, which is critical to minimize the spoofing of the markings by an attacker, so that a single attacker is unable to forge an edge between itself and the victim (Savage et al., 2000). Repeatedly, this procedure takes place for the following domains as the packets traverse along the path. We also remark that incremental deployment is ensured, because we can identify a domain even if only one border router within that domain sees the attack packet and marks it by our marking scheme.

Algorithm 1 Domain id marking by border router R

```
for each packet $pkt$ from an upstream domain do
  generate a random number $x$ within [0..1]
  if $x < p$ then
    $pkt.edge = did$
    $pkt.dist = 0$
    $pkt.flag = 1$
  else
    if $pkt.dist$ is 0 then
      $pkt.edge = pkt.edge \oplus did$
    increment $pkt.dist$
```

Figure 3: Domain id marking algorithm

3.4 Router id Marking

Figure 4 outlines the algorithm for router *id* marking by router *R*. The router *id* marking algorithm is used to identify the source routers that serve as ingress points of attack packets. A router performs router *id* marking with certain marking probability if it receives a packet from the customer side. Recall that we refer to the domain where a source router resides as source domain. To complete the inter-domain attack path, the source domain *id* should also be conveyed to the victim; so we make it equally likely to mark a packet with the router *id* or the domain *id* at the source router. In practice, we use a larger marking probability *q* in router *id* marking procedure, which is double of the probability *p* that we use in domain *id* marking. It's like flipping a coin to decide to mark with domain *id* or router *id*. To mark a packet with a domain *id*, we set flag to be one and write the domain *id* into the edge field; and to mark the packet with a router *id*, we set flag to be zero and write the 10-bit router *id* into the rid field.

We also note that a 10-bit router *id* can be used to identify a router uniquely only within a domain; so we need to combine the router *id* and the corresponding source domain *id* to uniquely identify the source routers universally. We therefore write a 14-bit checksum *cord* side by side with the router *id* into the marking fields to associate a router *id* with the corresponding source domain. The checksum can be hashed from the 16-bit domain *id*, and it is sufficient for distinguishing 2^{14}(16384) source domains possibly involved in an attack. Therefore, we can place the identified source routers in their corresponding source domains according to the checksums to complete the attack graph reconstruction.

```
Algorithm 2 Router id marking by router R

for each packet pkt passing through R do
    generate a random number x within [0..1]
    generate a random number r within [0..1]
    if $x < q$ then
        if r < 0.5 then
            pkt.edge = did
            pkt.distance = 0
            pkt.flag = 1
        else
            pkt.rid = rid
            pkt.cord = hash(did)
            pkt.flag = 0
```

Figure 4: Source router id marking algorithm

3.5 Attack Graph Reconstruction

Figure 5 describes the reconstruction procedure; it's a two-phase procedure. The first one is inter-domain attack graph reconstruction using packets marked with domain *ids* and then by the end of first stage, the source domains are identified. In the second phase, the algorithm relies on the packets with router *id* markings to identify the source routers. The property of XOR that $\alpha \oplus \beta \oplus \alpha = \beta$ allows us to decode the domain *ids* hop-by-hop during inter-domain attack path reconstruction. The algorithm starts from recovering the domains one hop away from the victim. Let the victim's domain *id* be α and a domain closest to the victim have an *id* β. By $\alpha \oplus \beta \oplus \alpha = \beta$, the domain *id* β can be decoded, given the attack packet marked with the value: $\alpha \oplus \beta$, and distance equal to 1. Likewise, for all domain-*id*-marked packets at distance d, pkt_i, a number of candidate domain *ids* can be generated by XORing the edge it contains with the *ids* of previously reconstructed domains at distance d−1. We denote a candidate as D_{ij}, which is decoded from pkt_i and the known endpoint of the edge is node D_j. Then the victim checks the upstream domain topology, *M* as a road-map to verify candidate D_{ij} by checking if an edge does exist between the candidate and node D_j on *M*. Node D_j and the verified candidate D_{ij} would therefore make up an edge on the reconstructed attack path. Hop by hop, the inter-domain attack graph is thus reconstructed by the repeated process.

To locate the source domain associated with checksum C_1 (let the checksum marked in an attack packet be C_1), the victim performs a breadth-first search from the victim on the reconstructed inter-domain attack graph, level by level until it gets to the node with a checksum equal to C_1. Suppose we denote a source domain node as *SD* and denote the set of source routers that may belong to domain *SD* as *SD.Rset*. The victim first sorts the router-*id*-marked packets by their cord field, finds out the packets with a matched checksum of *SD*, and then adds their router *ids* into *SD.Rset* (it is initially empty and it does not include any duplicates). The victim thus reconstructs the attack graph that incorporates the identified intermediate domains and the source routers.

```
Algorithm 3 Path reconstruction at victim V

let max_d be maximum attack path length
let G be reconstructed attack graph, initialized with the
vertex V
let M be the upstream inter-domain Internet map
Let SD.Rset be the set of source routers in domain SD

//Inter-domain attack graph reconstruction
for d = 1 to max_d do
  for each node D at distance $d - 1$ in G do
    for each packet pkt with distance d, flag 1 do
      candidate = $pkt.edge \oplus D.did$
      if candidate is equal to D'.did and D' is one of
      the upstream nodes of D on M then
        insert a new edge (D , D') into G

//Source router identification
for each attack packet pkt with flag 0 do
  for each node SD in G do
    if hash(SD.did) is equal to pkt.cord then
        insert pkt.rid into SD.Rset

output all the attack paths in G
```

Figure 5: Attack path reconstruction algorithm

4 PERFORMANCE EVALUATION

To test the performance of our proposed marking scheme, we conduct a number of experiments using an Internet map based on the traceroute dataset of the real Internet from CAIDA's Skitter Internet mapping project (CAIDA, 2004); the dataset contains 178,207 distinct traceroute paths widely distributed over the entire Internet. For the experiments, we use the first two bytes of an IP address as a domain *id* used in the domain *id* marking. On the other hand, the last two bytes of an IP address would be processed to be used as router *id* which would be used in the router *id* marking. The experiments are performed to assess the performance of our marking scheme characterized by a number of parameters, namely the minimum number of packets for the reconstruction of an attack path of a certain length, false positives, number of attack sources, and attack path reconstruction time. The results are as presented in Figures 6 to 8. Each data point in Figures 6 to 8 corresponds to an average value based on around 1000 experiment runs.

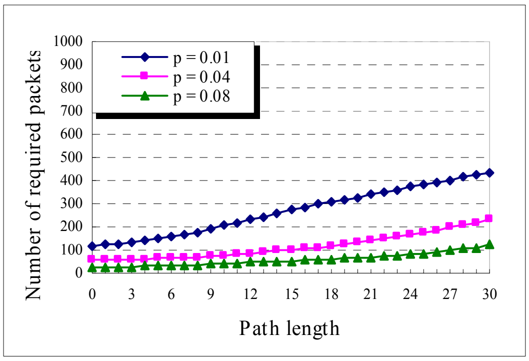


Figure 6: Number of packets required for attack paths reconstruction for different path lengths and different marking probabilities

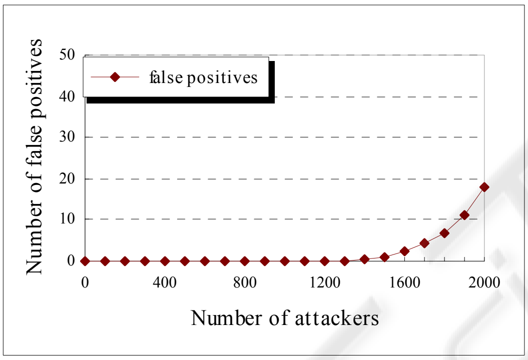


Figure 7: False positives generated for different number of attackers

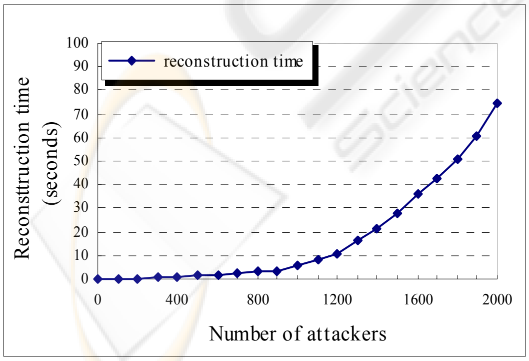


Figure 8: Reconstruction time for different number of attackers

Figure 6 shows the minimum number of packets required for the reconstruction of attack paths of different lengths and different marking probabilities. Since a packet will normally traverse no more than 30 routers in the Internet to reach its destination, the attack path lengths considered in the experiments range from 0 to 30. In general, for each marking probability, the number of required packets for path reconstruction increases linearly with the path length. For the case with marking probability 1%, and path length 30, the required number of packets would be around 400; if the marking probability is 4%, roughly 200 packets would be required. When compared with other IP traceback methods, our proposed marking scheme requires fewer packets for reconstruction. For instance, for a marking probability of 4% and path length 30, scheme 2 with m>5 and scheme 2 with m>6 of the Advanced Marking Schemes (AMS) (Song et al., 2001), around 1000 and 4000 packets respectively are required for reconstruction, where m is the number of hash functions used to encode the router identification.

If we conservatively assume each domain contains two routers, a maximum domain based path length would be half of its equivalent router based path length. This implies that our marking scheme needs to handle attack paths with average path length equal to one half the path lengths of those handled by other marking schemes. Moreover, we need only one marked packet to identify a domain; whereas other marking schemes normally employ full *IP* address markings for full-path reconstruction, and several packets are usually required to identify each router on an attack path. For instance, eight marked packets are employed in both Savage's method (Savage et al., 2000) and Song and Perrig's method (Song et al., 2001) to encode each router's identity. So our marking scheme needs substantially fewer packets for attack graph reconstruction.

Figure 7 presents the number of false positives for different number of attackers in the range 100 to 2000. It shows that our marking scheme is free of any false positives even in presence of 2000 attackers. Though hash is used to encode the checksum which is used to locate source routers, 2^{14} (16384) different values are sufficient for the number of do mains possibly involved as source domains, that is , there is sufficient mapping space for the hashed values so that a collision-free hash function should be able to generate a near-zero result. For comparison, scheme 2 of Song and Perrig's method (Song et al., 2001) with m >7 produced around 20 false positives in presence of 2000 highly distributed attackers. While our marking scheme has a computation complexity of around O(dn^2), the method of Savage, *et al.*(2000) and the method of Song and Perrig (Song et al., 2001) have a complexity of around O(dn^8) and O(dn^2) respectively, where d is the maximum path length and n is the number of attacking hosts. Since our domain based marking scheme involves a smaller distance *d*, its complexity is relatively small.

Figure 8 presents the reconstruction times of our reconstruction algorithm for different number of attackers, measured on a 1500MHz Pentium IV PC platform. The results show that in general the attack graph reconstruction could be completed quite rapidly. Even for the case of 2000 attackers, it takes only about 50 seconds to reconstruct the attack graph, which is considered quite a fast response to a highly distributed large-scale DDoS attack.

5 CONCLUSION

This paper proposes an innovative marking scheme which supports two kinds of packet marking: inter-domain marking and source router marking. Based on the markings in the received packets, the victims can reveal the inter-domain attack paths and identify the source routers serving as the ingress points of attack traffic.

The advantages of the proposed IP traceback method include: (1) As the marking algorithms involve only the border routers, it ensures a practical implementation without a universal deployment on all the routers. (2) We keep track of domains traversed by attack packets other than all individual routers; as a result, attack paths reconstruction could be carried out more rapidly by our marking scheme. (3) Using the relatively short *id* instead of a full IP address, we do not need to split the markings for each domain into a number of fragments and the whole marking can be written into a single IP header. The number of packets required to identify each domain can thus be kept to a minimum.

Through the simulation experiments on the proposed marking scheme, we observe the following: (i) It requires a much smaller number of packets for attack paths reconstruction than other methods such as AMS (Song et al., 2001); (ii) It can handle multiple attack sources effectively in very large scale; (iii) The number of false positives generated even in the presence of 2000 attack sources is relatively small; (iv) It performs attack paths reconstruction quite rapidly and takes only around 50 seconds to reconstruct as many as 2000 attack paths on a Pentium IV PC platform. Thus it could be used to locate attack sources in real time, which is one of the critical steps in defending against DDoS attacks.

REFERENCES

Alezxx C. Snoeren, Craig Partridge, Luis A. Sanchez, Christine E. Jones, Fabrice Tchakountio, 2002. Single-Packet IP Traceback. *IEEE/ACM transactions on NETWORKING, Vol. 10, No. 6, December.*

Andrew S. Tanenbaum, Aug 9, 2002. *Computer Networks*, 4rd edition. Published by Prentice Hall PTR.

Andrey Belenky, Nirwan Ansari, 2003. IP Traceback with Deterministic Packet Marking, *IEEE COMMUNICATIONS LETTERS, VOL. 7, NO. 4, APRIL.*

The Cooperative Association for Internet Data Analysis, 2004. Available: *http://www.caida.org/tools/measurement/skitter*

Dawn X. Song and Adrian. Perrig, 2001. Advanced and Authenticated Marking Schemes for IP Traceback. *Proc. of the IEEE Infocom conference, April.*

Hal Burch and Bill Cheswick, 1999. Tracing Anonymous Packets to Their Approximate Source. *Unpublished paper, December.*

J. Ioannidis and S. M. Bellovin. 2002. Implementing Pushback: Router-based Defense against DDoS Attacks. *Proc. in Network and Distributed System Security Symposium, the Internet Society.*

Kevin J. Houle, George M. Weaver, 2001. Trends in Denial of Service Attack Technology. *Technical report from CERT Coordination Center.* October.

Michael T. Goodrich, 2002. Efficient Packet Marking for Large-Scale IP Traceback. *CCS'02, November, Washington, DC, USA.*

Rocky K. C. Chang, 2002. Defending against Flooding-based Distributed Denial-of-service Attacks: a Tutorial, *IEEE Communications Magazine, October.*

S. M. Bellovin, 2000. ICMP Traceback Messages.
Internet Draft: *http://www.research.att.com/~smb/papers/draft-bellovin-itrace-00.txt* (June 20, 2004)

Stefan Savage, David Wetherall, Anna Karlin and Tom Anderson, 2000. Practical Network Support for IP Traceback. *Proc. of the ACM SIGCOMM conference, August.*

Steven H. Bass, 2001. Spoofed IP Address Distributed Denial of Service Attacks: Defense-in-Depth. Available: *http://www.sans.org/rr/papers/60/469.pdf* (July 30, 2004)

Vadim Kuznetsov, Andrei Simkin, Helena Sandström, 2002. An Evaluation of Different IP Traceback Approaches. Available: *http://www.sm.luth.se/csee/csn/publications/ip_traceback.pdf*