Authenticated Autonomous System Traceback

Vamsi Paruchuri, Arjan Durresi, Rajgopal Kannan and S. Sitharama Iyengar
Department of Computer Science
Louisiana State University
{vamsip, durresi, rkannan, iyengar} @csc.lsu.edu

Abstract

The design of the IP protocol makes it difficult to reliably identify the originator of an IP packet making the defense against Distributed Denial of Service attacks one of the hardest problems on the Internet today. Previous solutions for this problem try to traceback to the exact origin of the attack by requiring every router's participation. For many reasons this requirement is impractical and the victim ends up with an approximate location of the attacker. Reconstruction of the whole path is also very difficult owing to the sheer size of the Internet.

This paper presents lightweight schemes for tracing back to the attack-originating AS instead to the exact origin itself. Once the attack-originating AS is determined, all further routers in the path to the attacker are within that AS and under the control of a single entity; which can presumably monitor local traffic in a more direct way than a generalized, Internet scale, packet marking scheme can. We also provide a scheme to prevent compromised routers from forging markings.

Keywords: *traceback, DDoS, network security.*

1 Introduction

Distributed denial-of-service attacks (DDoS) pose an immense threat to the Internet, and consequently many defense mechanisms have been proposed to combat them. Attackers constantly modify their tools to bypass these security systems, and researchers in turn modify their approaches to handle new attacks. DDoS field is evolving quickly, and it is becoming increasingly hard to grasp a global view of the problem.

DDoS attacks are so difficult to trace because the only hint a victim has as to the source of a given packet is the source address, which can be easily forged. Also, many attacks are launched from compromised systems so finding the source of the attacker's packets may not lead to the attacker. If a victim is able to determine the path of the attacking packets in near real-time, it would be much easier to quickly stop the attack and facilitate the identification of the attacker. Even finding out partial path information would be useful because attacks could be throttled at far routers.

A number of recent studies have been carried to solve the IP traceback problem. All of these studies have aimed at identifying the exact origin of the attacks and for this purpose, they require that every router in the Internet to participate in the traceback algorithm. We believe that this requirement might not be realistic. For technical and political reasons or due to lack of clear incentives, many routers might not participate in the traceback mechanisms. Thus, the algorithms proposed so far either fail completely in identifying the origin or provide an approximate location of the origin.

In this paper, we present a new approach to the traceback problem that addresses the needs of both victims and network operators. Our solution is to probabilistically mark packets with AS numbers rather than with IP addresses. Marking with AS number information greatly improves the efficiency of our solution when compared to other similar packet marking approaches in terms of speed of path reconstruction and number of packets needed to reconstruct attack path. Though our scheme does not succeed in tracing back to the exact origin of the attack, we traceback to the attack originating AS in real time. Once the attack-originating AS is determined, all further routers in the path to the attacker are within that AS and under the control of a single entity; which can presumably monitor local traffic in a more direct way than a generalized, Internet scale, packet marking scheme can.

We also present a lightweight algorithm to mitigate the problem of compromised routers. This prevents compromised routers from changing the contents of a packet and forging the markings of uncompromised routers. For this purpose we assume that it is hard to compromise Autonomous System Border Routers (ASBRs). We think this assumption is valid especially because, once an ASBR is compromised much worse attacks than DOS attacks can be possible [21, 22, 23].

Our traceback algorithms add little or no overhead to the router's critical forwarding path. In fact, the only invariant that we can depend on is that a packet from the attacker must traverse all of the routers between it and the victim.

Proceedings of the 18th International Conference on Advanced Information Networking and Application (AINA'04)
0-7695-2051-0/04 $ 20.00 © 2004 IEEE

The rest of this paper is organized as follows: Section 2 discusses related work, Section 3 deals with motivation and background, Section 4 presents Autonomous System based Traceback, Section 5 presents the Authenticated Traceback scheme and Section 6 concludes.

2 Related Work

Researchers have proposed various schemes to address the IP traceback problem. Unfortunately they are mostly inefficient or ineffective and not robust against DDoS. The most obvious countermeasure DDoS attacks certainly is ingress filtering [2] based on source address. History seems to show that it is quite difficult to convince ISPs to install, configure, maintain, and support new protocols that cannot be sold as part of a service. As ingress filtering mostly protects users at other ISPs, the paying customers of an ISP implementing ingress filtering would not directly have a benefit, but instead might run into problems caused by the above-mentioned protocols. As this decreases customer happiness and increases customer service calls, ISPs thus seem unlikely to implement ingress filtering in the near future. The next step is victim pushback, where a site that believes to be under attack can send back messages installing filters at upstream routers [14, 13, 15]. Due to the current lack of incentives for ISPs to provide such a service, it is not expected to become widely deployed anytime soon.

Several approaches have been proposed with respect to traceback and identification of the attackers. One promising solution is to let routers probabilistically mark packets with partial path information during packet forwarding [6]. The victim then reconstructs the complete paths after receiving a modest number of packets that contain the marking. But, as shown in [7], this approach has a very high computation overhead for the victim to reconstruct the attack paths, and gives a large number of false positives when the denial-of-service attack originates from multiple attackers. This approach is vulnerable to compromised routers. A router compromised can forge markings from other uncompromised routers and hence lead the reconstruction to wrong results. Even worse, the victim will not be able to tell a router is compromised just from the information in the packets it receives.

Song et al. [7] improve on the Savage scheme by predetermining the network topology. This solution is limited to cases when the topology is static (at least locally to the potential victim) and the victim is immobile. The probing of the topology can be very taxing to the network, especially if a large proportion of Internet sites would start doing it. This map also allows for a more efficient encoding of edges and thus resulting in fewer chunks to reconstruct paths and in greatly improving the efficiency and accuracy of the protocol. However, given its high pre-attack overhead and need for continuous topology updates, it might be infeasible for large-scale deployment.

Dean et al. [17] provide another avenue to improve CEFS. Instead of using a hash function as a verifier, the routers algebraically encode the path or edge information iteratively using Horner's rule. The resulting (x, y) coordinate tuples allow the reconstruction of the contributing polynomial coefficients, which encode the complete path. This scheme is susceptible to a GOSSIB attack [19]. Also, the number of packets required to reconstruct path is high and this approach is vulnerable to compromised routers.

The Internet Engineering Task Force (IETF) also has a working group dedicated to establishing a standard traceback mechanism for *very big flows*. The working group proposed that each router would periodically (every few hundred or thousand packets) select a packet and "append" authenticated traceback information to this packet [18]. This information would convey that the packet was seen by this router. *Appending* would not be done by modifying the packet, but by creating a second packet tailgating the original packet. The working group has been largely dead since about a year, and its Internet-Drafts have all expired since, so the approach seems to have been abandoned.

A different approach for traceback is shown by Snoeren et al. [12]. They propose storing a hash of each packet along with information about where it arrived from in a memory efficient fashion. Given ubiquitous deployment of such a service, a network node can immediately ask for a traceback of an individual packet it just received. This approach needs complete (or at least very dense) deployment and the overhead on the routers is too huge.

In this paper, we present two new IP marking techniques to solve the IP traceback problem: AS Marking Scheme and an Authenticated AS Marking Scheme. Our approach has the low network and router overhead and our approach is much more efficient and accurate for the attacker path reconstruction under DDoS. Our approach can trace the origin AS of the attack unlike the earlier schemes that try to trace the attack originating router(s). Our approach can reconstruct the attacker path within seconds and has almost zero false positive rate. Furthermore, our Authenticated Marking Scheme supports efficient authentication of routers' markings. This prevents a compromised router from forging other uncompromised routers markings.

3 Motivation and Background

In this section initially we present two basic marking schemes [6] that our mechanism is based on and present

their limitations. We then present the motivation behind AS traceback and how we intend to overload the IP header.

3.1 Basic Node Marking Algorithms

3.1.1 Node Append

This is the simplest marking algorithm and is conceptually similar to the IP Record Route option. Each node appends its address to the end of the packet as it travels through the network from attacker to victim. Consequently, every packet received by the victim arrives with a complete ordered list of the routers it traversed – a built-in attack path.

The node append algorithm is both robust and extremely quick to converge (a single packet); however it has several serious limitations – principal ones being the unfeasibly high router overhead incurred by appending data to packets in flight and since the length of the path is not known *a priori*, it is impossible to ensure that there is sufficient unused space in the packet for the complete list. This can lead to unnecessary fragmentation and bad interactions with services such as MTU discovery [5]. This problem cannot be solved by reserving *enough* space, as the attacker can completely fill any such space with false, or misleading, path information.

3.1.2 Node sampling

To reduce both the router overhead and the per-packet space requirement, one can sample the path one node at a time instead of recording the entire path. A single static "node" field is reserved in the packet header – large enough to hold a single router address (i.e. 32 bits for IPv4). Upon receiving a packet, each router chooses to write its address in the node field with some probability p. After enough packets have been sent, the victim will have received at least one sample for every router in the attack path. If it is assumed that the attacker sends enough packets and the route is stable enough, this sampling can converge.

Although it might seem impossible to reconstruct an ordered path given only an unordered collection of node samples, it turns out that with a sufficient number of trials, the order can be deduced from the relative number of samples per node. Since, routers are arranged serially, the probability that a packet will be marked by a router and then left unmolested by all downstream routers is a strictly decreasing function of the distance to the victim. If we constrain p to be identical at each router, then the probability of receiving a marked packet from a router d hops away is $p(1- p)^{d-1}$. Since this function is monotonic in the distance from the victim, ranking each router by the number of samples it contributes will tend to produce the accurate attack path.

Putting aside for the moment the difficulty in changing the IP header to add a 32-bit node field, this algorithm is efficient to implement because it only requires the addition of a write and checksum update to the forwarding path. Current high-speed routers already must perform these operations efficiently to update the time-to-live field on each hop. Moreover, if $p > 0.5$ then this algorithm is robust against a single attacker because there is no way for an attacker to insert a "false" router into the path's valid suffix by contributing more samples than a downstream router, nor to reorder valid routers in the path by contributing more samples than the difference between any two downstream routers [6].

However, there are also two serious limitations. First, inferring the total router order from the distribution of samples is a slow process. Routers far away from the victim contribute relatively few samples (especially since p must be large) and random variability can easily lead to misordering unless a very large number of samples are observed. For instance, if $d = 15$ and $p = 0.51$, the receiver must receive more than 42,000 packets on average before it receives a *single* sample from the furthest router. To guarantee that the order is correct with 95% certainty requires more than seven times that number. The order can also be obtained by allocating a separate field for *hop count*. A marking router sets this field to zero and each subsequent router just increments it. Thus, when a packet is received the distance to the router that has marked the packet is given by the *hop count* field and thus ordering is made simple.

Second, if there are multiple attackers then multiple routers may exist at the same distance – and hence be sampled with the sample probability. Therefore, this technique is not robust against multiple attackers. But, if the victim has access to the global Internet map, then by seeing the connectivity between the routers, the victim can construct the attack graph. Again, the current size of the Internet being greater than 1.6e+08 hosts [24, 25], obtaining the global Internet map is a non-trivial issue.

3.2 Design Motivation

An Autonomous System (AS) is a group of IP networks operated by one or more network operator(s), which has a single and clearly defined external routing policy. The classic definition of an Autonomous System is a set of routers under a single technical administration, using an IGP (Interior Gateway Protocol) and common metrics to route packets within the AS, and using an exterior gateway protocol to route packets to other ASes.

An Autonomous System Number (ASN) is a globally unique number in the Internet to identify an AS. An AS number is used in both the exchange of exterior routing information between neighboring ASes, and as an

identifier of the AS itself in the global Internet. AS numbers are 16-bit integers, assigned and managed by Internet Assigned Numbers Authority (IANA).

Autonomous System Border Routers (ASBRs) are connected to more than one AS, and exchange routing information with routers in another AS. ASBRs advertise the exchanged external routing information throughout their AS. The traffic to/from an ASBR is controlled by its ASBRs. Any traffic originating from (to) an AS to (from) a node outside the AS has to pass through an ASBR of the AS.

We propose marking packets with AS numbers rather than IP address of the routers. Marking with AS numbers has following advantages:

1. Number of ASes is far lesser than the number of routers in the Internet. The Internet consists around 14,000 ASes as compared to 1.6e+08 hosts [25]. Hence, obtaining an AS map of the Internet is feasible while obtaining Internet map itself is very difficult if not impossible.
2. In more than 99.5% cases, a packet passes through less than seven ASes before reaching its destination [24, 25].
3. Private owned ASes may not always like to disclose their network details. If each router in the AS participates in the marking scheme, then one can easily infer the network architecture by observing the markings.
4. AS number is 16 bits in length while IP v4 address is 32-bit length (IPv6 address length is 128 bits). Thus, encoding AS number need lesser header space than encoding IP address and thus with the same mechanism, AS path construction needs far lesser packets than whole network path.
5. There is no scope for false positives[1] as when packet is marked, it carries the whole AS number of the router and the victim can reconstruct the attack path without any uncertainty.

It is not practical to assume that all the routers in the Internet participate in the marking scheme. In the case when not all routers are participating in the marking scheme, most of the schemes in the literature succeed in tracing back to a participating router that is closest to the attacker and that is in the path traversed by the packets. Thus, when some routers are not participating, there is always a concern if the last router in the path constructed by the victim is a true origin of the attack. If all the neighboring routers of the router in consideration are participating then one can be sure that the router is a true origin of the attack. But, if at least one neighbor is not participating, one cannot be sure about the origin of the attack. We believe that making all routers participate in marking scheme is very difficult if not impossible either due to technical reasons or administrative reasons.

Furthermore, as stated in [6], the marking probability needs to be greater than 0.5 so that the algorithm is robust against a single attacker so that an attacker cannot insert a "false" router into the path's valid suffix or reorder valid routers in the path. But, when $p > 0.5$, the number of packets that the victim has to obtain to reconstruct the attack path is very high. When an authentication mechanism is in place, the marking probability can be much lesser than 0.5, enabling fast and real time reconstruction of attack graph.

3.3 Header Overloading

We overload IP header to store router markings. Our mechanism uses 25 bits for marking. Even though it is out of the scope of this paper to precisely identify these bits, a possible set of fields is shown below.

The TOS Field: The type of service field is an 8 bit field in the IP header that is currently used to allow hosts a way to give hints to routers as to what kind of route is important for packets. This field has been little used in the past, and, in some limited experiments, we have found that setting this field arbitrarily makes no measurable difference in packet delivery.

The ID Field: The ID field is a 16-bit field used by IP to permit reconstruction of fragments. Naive tampering with this field breaks fragment reassembly. Since less than 0.25% of all Internet traffic is fragments [22], we think that overloading this field is appropriate. A more in-depth discussion of the issues related to its overloading can be found in Savage's work [19].

The Unused Fragment Flag: There is an unused bit in the fragment flags field that current Internet standards require to be zero. Setting this bit to one has no effect on current implementations; with the exception that when receiving the packet, some systems will think it is a fragment. The packet is still successfully delivered however, because it looks to those systems as though it is fragment 1 of 1.

4 Autonomous System based IP Traceback

In this section we present a lightweight traceback mechanism to traceback to the attack originating Autonomous System. For this purpose we use 19-bits of header space.

We use a similar marking scheme as node sampling scheme, but instead of marking a packet with the IP address of a router Ri, we mark the packet with its AS number. In this scheme, we reserve two fields – a 16-bit

[1] We call an AS *false positive* if it is in the reconstructed attack graph but not in the real attack graph.

ASN field and a 3-bit AS_distance field – in the packet header. Note that 3 bits can represent up to a distance of 8 hops in terms of ASes traversed which is sufficient for almost all Internet paths [24, 25, 11].

AS marking scheme is performed only at Autonomous System Border Routers (ASBRs). To be more specific an ASBR marks a packet with its AS number (ASN) only if the packet is forwarded to a router belonging to another AS. Thus, a packet might get marked only when it exits an AS. Upon receiving a packet that is being forwarded to another AS, each ASBR chooses to write its ASN in the node field with some probability *p* and set the *distance* field to zero. If the ASBR chooses not to write the outgoing packet, it just increments the distance field by one. The full algorithm is shown in Figure 1.

After enough packets have been sent the victim will have received at least one marking from every router. The victim can reconstruct the ordered path with the help of the distance field. Assuming the marking probability p is identical at each router, the probability of receiving a marked packet from an ASBR that is at a distance of d_{AS} (in terms of ASes) is $p(1-p)^{d_{AS}-1}$. For instance, if $d_{AS} = 7$ and $p = 0.51$, the receiver must receive more than 141 packets on average before it receives a single sample from the furthest ASBR.

Once the attack originating AS is obtained through the marking scheme, all further routers in the path to the attacker are within that AS and under the control of a single entity; which can presumably monitor local traffic in a more direct way than a generalized, Internet scale, packet marking scheme can. The important contribution of this improvement is that it reduces the overhead imposed on the routers and also drastically simplifies the work need to be done at the victim to reconstruct the attack path. For attack path reconstruction the AS map of the Internet is needed. As the Internet consists around 14,000 ASes as compared to 1.6e+08 hosts [25], AS path reconstruction is definitely lot easier and lighter than complete route reconstruction.

Figure 2 shows the mean number of packets required to reconstruct paths of varying AS path lengths. In [8], it is shown that the optimal marking probability is given as $1/D_{AS}$. The graph plots the number of packets needed for different marking probabilities. As the more than 99% of paths encountered in the Internet have an AS path length less than or equal to six, we propose to set the marking probability to 1/6. For p = 1/6, the victim needs to receive less than 25 packets in order to reconstruct the AS attack graph.

To make the algorithm is robust against a single attacker so that an attacker cannot insert a "false" router into the path's valid suffix or reorder valid routers in the path, p needs to be greater than 0.5 [6]. But, as seen from Figure 2, the number of packets the victim needs to reconstruct the attack graph with p = 0.5 is very high when compared to when $p \leq 0.1667$. But at low *p* values, without any authentication mechanism, it is hard to prevent compromised routers from misleading the victim. In the next section we present an authentication mechanism to overcome these problems.

Marking procedure at router R with AS Number R_{AS}*:*

```
for each packet w
   let x be a random number from [0, 1)
   if x < p then,
      write R_AS into w.AS
      set w.AS_distance=0
   else
      increment w.AS_distance
```

Figure 1. Autonomous System Marking algorithm.

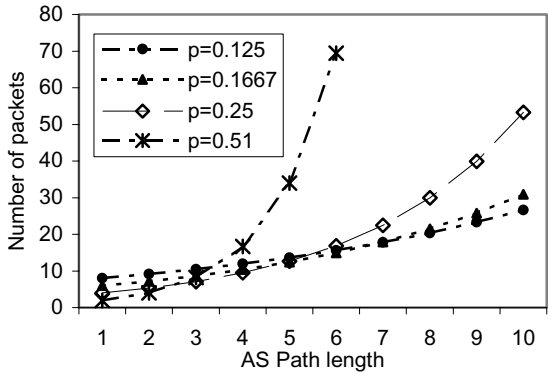

Figure 2. Number of packets needed to reconstruct paths of varying lengths for different marking probabilities.

5 Authenticated Marking Scheme

In this section we present a simple but effective authentication mechanism to prevent compromised routers from forging the ASBR markings and thus mislead the victim. We assume that ASBRs can be trusted and that ASBRs cannot be compromised.

5.1 Notation and Definitions

We assume the presence of a Symmetric Key Infrastructure within in each Autonomous System with each ASBR that either belongs to the AS or connected to the AS knowing the secret key. We denote the secret key of an AS *i* as K_i. We use $E(M, k_i)$ and $D(M, k_i)$ to denote the encryption and decryption of message M with key k_i.

5.2 One-way Hash chains

A *one-way hash chain* is built on a one-way hash function. Like a normal hash function, a one-way hash function, *H*, maps an input of any length to a fixed-length bit string. Thus, $H:\{0,1\}^* \rightarrow \{0,1\}^{\rho}$, where ρ is the length in bits of the output of the hash function. The function *H* should be simple to compute yet must be computationally infeasible in general to invert. A more formal definition of one-way hash functions is provided by Goldwasser and Bellare [9], and a number of such functions have been proposed, including MD5 and SHA1.

To create a one-way hash chain, a node chooses a random initial value $x \in \{0, 1\}$ and computes the list of values $h_0, h_1, h_2, \ldots, h_n$; where $h_0 = x$, and $h_i = H(h_{i-1})$ for $0 < i \leq n$, for some n.

Initially, all ASBRs that belong to a particular AS share a secret key h_0, which is also revealed to all the ASBRs that are directly connected to that AS. Then, each ASBR computes the one-way hash chain as illustrated above. Each key in this key chain starting from right to left (in the order of decreasing subscript *i*) is used as the symmetric key for the AS for one session. Each session is for a time span of *T* seconds, *T* depending upon the strength of the encryption algorithm and number of packets being encrypted per second. If all the ASBRs are time synchronized (at least loosely), then there is need for an explicit advertisement to indicate the start of a new session. If not, a pre-designated ASBR advertises the start of a new session and hence the usage of a new key to all the ASBRs in the AS and to those directly connected to the AS.

5.3 Algorithm

Our Authentication scheme is based on symmetric key cryptography and one-way hash chains. Each AS is assigned a secret hash value. This hash is known to all ASBRs that belong to the AS and to the ASBRs that are connected directly to the AS. Using the hash, all the ASBRs will be able to derive the corresponding one-way hash chain. The 25-bit *AS Marking* field is assigned to a cipher text generated as follows:

$$E(ASN \,||\, RP\,,\, K_{AS})$$

ASN is the16-bit Autonomous System Number of the AS to which the marking ASBR belongs to and RP is the 9-bit Redundancy Predicate that has to be fulfilled so that the marking can be verified. RP can be simply set to *all 1s*, but this mechanism does not prevent a compromised router from copying the marking of one packet to another. Thus to prevent these, RP should be packet-dependent. One method of computing Redundancy Predicate is to set RP to a hash of source-destination address pair.

Marking

When an ASBR receives a packet, it first decrypts the AS Marking carried by the packet with K_{AS}, the symmetric key of the AS it belongs, thus computing $D(E(ASN \,||\, RP\,,\, K_{AS}), K_{AS})$ and verifies the Redundancy Predicate. If the Redundancy Predicate is fulfilled, then with some probability, *p*, it marks the packet with $E(ASN \,||\, RP\,,\, K_{AS})$, where ASN is the AS number of the ASBR and K_{AS} is the symmetric key of the AS to which the next ASBR belongs.

Marking procedure at router R with ASN R_{AS}:

K_{AS} is the symmetric key of R_{AS}
K'_{AS} is the symmetric key of the next AS in the path.

```
for each packet w
        Compute D(AS Marking, K_AS)
    if (Redundancy Predicate is not fulfilled )
        Set AS Marking to E(ASN || RP , K'_AS)
    else
       let x be a random number from [0, 1)
       if x < p then,
         Set AS marking to E(ASN || RP , K'_AS)
       else
         Set AS marking to E(D(AS Marking, K_AS), K'_AS)
```

Figure 3. Authenticated AS marking algorithm

For instance, if the packet is entering the AS, then the ASBR uses the symmetric key of its AS. But, if the packet is being forwarded to another AS, then the ASBR uses the symmetric key of the AS to which the packet is being forwarded[2]. If a packet's RP is not fulfilled, then the ASBR definitely writes into the packet. The marking algorithm is shown in figure 3. For the same reasons explained in section IV, we set the marking probability to $p = 1/6$.

The victim obtains the AS symmetric key of the current session from an ASBR and thus computes all the AS markings. The attack path reconstruction is now pretty same as the reconstruction algorithm explained in Section IV. As we employ hash chains to compute the symmetric key of each session, when a symmetric key of a session is revealed to the victim, he can just use the key to compute the keys of all previous sessions and not any future session. Thus, even if the victim is compromised, the security of the mechanism is not affected.

[2] A simple mechanism to implement this would be to have an ASBR use the key of its AS if the destination IP address of that packet matches a route obtained through an Interior Gateway Protocol (IGP).

A compromised router, which sets the 25-bit AS marking field in a packet, can succeed from preventing the packet being marked overwritten by the next ASBR with a probability of $1/2^9$ that is less than 0.002. Thus, approximately one out of every 500 packets that reach the victim is not marked by an ASBR. A false positive is generated only if the victim receives markings from all intermediate ASes to some AS that is not a origin of an attack. The attacker could not spoof markings of any AS, as the attacker does not have any information of the symmetric keys.

6 Conclusion

In this paper, we present two schemes, the Autonomous System based Traceback and the Authenticated Marking scheme, which allow the victim to traceback to the originating AS of the spoofed IP packets. In contrast to the previous works, we aim at tracing back to the attack originating AS rather than the exact router. This is because the latter requires each and every router's participation, which we consider as an impractical assumption. Our techniques have very low network and router overhead and enables to reconstruct the AS attack graph in real time. Our marking schemes have little or no positive rate and very low computation overhead for the victim to reconstruct the attack graph. Furthermore our Authenticated Marking scheme prevents a compromised router from forging the markings of an ASBR.

7 References

1. Computer Emergency Response Team, CERT Advisory CA-97.28, *IP Denial-of-Service Attacks*, www.cert.org/advisories/CA-97.28.smurf.html, Dec'1997.
2. P. Ferguson and D. Senie, *Network Ingress Filtering: Defeating Denial of Service Attacks Which Employ IP Source Address Spoofing*, RFC 2267, Jan. 1998.
3. Thomas Doeppner, Philip Klein, and Andrew Koyfman. *Using router stamping to identify the source of IP packets.* In Proceedings of the 7th ACM Conference on Computer and Communications Security, pages 184--189, Nov' 2000.
4. A. Mankin, D. Massey, C.L. Wu, S.F. Wu, and L. Zhang, *On Design and Evaluation of Intention-Driven ICMP Traceback*, Proceedings of the IEEE ICCCN, Oct' 2001.
5. J. Mogul and S. Deering. *Path MTU Discovery*. RFC 1191, Nov. 1990.
6. Stefan Savage, David Wetherall, Anna Karlin and Tom Anderson, *Practical Network Support for IP Traceback,* Proceedings of the 2000 ACM SIGCOMM Conference.
7. Dawn Xiadong Song and Adrian Perrig, Advanced and *Authenticated Marking Schemes for IP trace back*, IEEE INFOCOM 2001.
8. T. Peng, C. Leckie and R. Kotagiri. *Adjusted Probabilistic Packet marking*. In Proceedings of the Second IFIP Networking Conference, May 2002.
9. Shafi Goldwasser and Mihir Bellare. *Lecture Notes on Cryptography.* Summer Course "Cryptography and Computer Security" at MIT, 1996–1999, August 1999.
10. HMAC: Keyed-Hashing for Message Authentication, RFC 2104, www.ietf.org/rfc/rfc2104.txt
11. The Cooperative Association for Internet Data Analysis (CAIDA) MapNet tool, http://www.caida.org/tools/visualization/mapnet/.
12. Alex C. Snoeren, Craig Partridge, Luis A. Sanchez, Christine E. Jones, Fabrice Tchakountio, Stephen T Kent and Timothy Strayer. *Hash-Based IP Traceback*, Proc. of ACM SIGCOMM 2001.
13. Ratul Mahajan, Steven M. Bellovin, Sally Floyd, John Ioannidis, Vern Paxson, and Scott Shenker, *Controlling high bandwidth aggregates in the network,* Technical report, AT&T Center for Internet research, ICSI, July 2001.
14. Kihong Park and Heejo Lee, *A proactive approach to distributed dos attack prevention using route-based distributed filtering*. Technical Report CSD-00-017, Department of Computer Sciences, Purdue University, 2000.
15. John Ioannidis and Steven M. Bellovin. *Implementing pushback: Router-based defense against DDoS attacks*. In Proc. of Network and Distributed System Security Symposium, Feb'2002.
16. H. Tangmunarunkit, J. Doyle, R. Govindan, S. Jamin, S. Shenker, and W. Willinger. *Does AS Size Determine Degree in AS Topology?* ACM Computer Communication Review, Vol. 31 No. 5, p. 7-10, October 2001.
17. Drew Dean, Matt Franklin, and Adam Stubblefield. *An algebraic approach to IP traceback*. In Proc of the Network and Distributed System Security Symposium, February 2001.
18. Stefan Savage, David Wetherall, Anna R. Karlin, and Tom Anderson. *Network support for IP traceback*. ACM/IEEE Transactions on Networking, 9(3): 226–237, June 2001.
19. Marcel Waldvogel, *GOSSIB vs. IP Traceback Rumors*, 18th Annual Computer Security Applications Conference (ACSAC 2002) December 2002.
20. W. Theilmann and K. Rothermel, *Dynamic Distance Maps of the Internet*. In Proc. of IEEE INFOCOM 2000.
21. B.R. Smith and J.J. Garcia-Luna-Aceves. *Securing the Border Gateway Routing Protocol*. In Proc of Global Internet '96.

22. S. Kent, C. Lynn, J. Mikkelson, and K. Seo. *Secure Border Gateway Protocol (Secure-BGP) - Real World Performance and Deployment Issues*. Proc. of Symposium on Network and Distributed System Security, Feb'2000.
23. S. Kent, C. Lynn, and K. Seo. *Secure Border Gateway Protocol (Secure-BGP).* IEEE Journal on Selected Areas in Communications, April 2000.I.
24. D. Magoni and J. Pansiot. *Analysis of the Autonomous System Network Topology*. ACM Computer Communication Review, v.31 n.3 July 2001, p. 26-37.
25. M. Fayed, P. Krapivsky, J. Byers, M. Crovella, D. Finkel, S. Redner. *On the Size Distribution of Autonomous Systems*. Technical Report, Boston University, Jan'2003.