A Stateless Traceback Technique for Identifying the Origin of Attacks from a Single Packet

Marcelo D. D. Moreira[†], Rafael P. Laufer[‡], Natalia C. Fernandes[†], and Otto Carlos M. B. Duarte[†]

[†]Universidade Federal do Rio de Janeiro
Rio de Janeiro, Brazil

[‡]University of California at Los Angeles
Los Angeles, USA

Abstract—**Anonymity is one of the main motivations for conducting denial-of-service attacks. Currently, there is no mechanism to either identify the true source of an IP packet or to prove its authenticity. In this paper we propose a stateless IP traceback technique that identifies the origin network of each individual packet. We show that the proposed traceback system is the only one that scales with the number of attackers and also satisfies practical requirements, such as no state stored at routers and a header overhead (25 bits) that can be allocated in IPv4 header. The proposed system exploits the customer-provider hierarchy of the Internet at autonomous system (AS) level and introduces the idea of checkpoints, which are the two most important nodes in an AS-level path. Simulation results using a real-world topology trace show that the proposed system narrows the source of an attack packet down to less than two candidate ASes on average. In addition, considering a partial deployment scenario, we show that the proposed system is able to successfully trace more than 90% of the attacks if only 8% of the ASes (i.e., just the core ASes) implement the system. The achieved success rate is quite better than using the classical hop-by-hop path reconstruction.**

I. Introduction

Distributed Denial-of-Service (DDoS) attacks are one of the main security challenges of the Internet today [1]. Usually, each attacking machine generates a certain amount of traffic towards a target server and the aggregate traffic of several machines is responsible for depleting the server's resources and, consequently, making the offered service unavailable. DDoS attacks only occurs because it is possible for the attackers to hurt the victims and still remain anonymous and, consequently, unpunished [2]. The IP layer does not provide source authentication and hence packets with spoofed source addresses can be transmitted over the network. A recent study [2] shows that approximately 20% of Internet Service Providers (ISPs) do not filter outgoing packets with spoofed source address. Inside such networks, a host can spoof up to 100% of all Internet addresses. Therefore, it is not possible to *prove* the participation of a particular machine in the attack, even if attacking machines employ real source addresses. This work aims at filling this gap in the Internet architecture in the source accountability area [3]. The scope of this work is the identification of the origin network of each IP packet.

A promising solution to the source identification problem is to make the network capable of tracing the attack packets back to their true source, which is known as the IP traceback problem [4]. The basic idea of IP traceback is to reconstruct the path taken by attack packets using routers markings inserted into forwarded packets or logging information saved at routers memory [4]. Nevertheless, up to now IP traceback has only been thought as a first step for the defense against DDoS attacks, and not as per-packet source identification mechanism. Only logging schemes and the system proposed by Laufer et al. have the ability of tracing the attacker from a single packet [5]. These schemes, however, do not meet all the requirements that a practical solution must satisfy. To the best of our knowledge, this work is the first proposal that allows identifying the attack origin from a single packet and also fulfills practical requirements, such as no state stored at routers and a header overhead (25 bits) sufficiently small to be allocated in IPv4 header. We exploit the hierarchical structure of the Internet at AS-level to accurately locate the attacker using information from the ASes traversed by the attack packet. The novelty of our idea is to strategically choose which ASes should mark the packet. We do not store the whole attack path due to space constraints; instead, only the path information that is essential to locate the attacker is stored in the packet. Given an AS-level attack path, we identify two ASes, called checkpoints, which are the most important for an accurate path reconstruction and propose a novel marking scheme that prioritizes these two critical nodes. This approach allows tracing the origin AS with high accuracy, despite the marking space limitation.

We developed a simulator and compared the performance of the proposed system with three marking schemes [6]–[8]. We used the AS-level Internet topology constructed with the Archipelago (Ark) measurement infrastructure [9] to evaluate all traceback systems in a real-world scenario. Our key finding is that the error rate of the proposed system is independent from the number of attackers, confirming the scalability of the single-packet traceback approach. In comparison, the performance achieved by the compared systems, which rely on multiple received packets to reconstruct the attack path, quickly degrades with the number of attackers. In addition, due to the use of checkpoints, beside of being constant, the error rate of the proposed system is quite low, because we have only 0.8 false positives per traced attacker on average.

This paper is organized as follows. Section II qualitatively compares our proposal with related work. Next, we present in Section III our findings about the AS-level path reconstruction problem that justify our solution, described in Section IV. Finally, we present simulation results in Section V and conclusions in Section VI.

II. Related Work

Several source identification mechanisms have been proposed. The simplest approach is to prevent spoofed packets from traversing the network using filtering techniques [10]. These techniques, however, require a widespread deployment to be effective. Alternatively, Passport [11] and AIP (Accountable Internet Protocol) [3] use cryptographic primitives to validate the source of IP packets. These proposals, however, face practical limitations that prevent their immediate deployment. Passport has a header overhead of 192 bits, which cannot be allocated in IPv4 header, and AIP's self-certifying addresses are also incompatible with the current version of IP. Furthermore, per-packet MAC (Message Authentication Code) computations require a processing overhead that limits the forwarding capacity to, at maximum, few gigabits per second with today's router hardware [11].

In this paper, instead of employing cryptographic primitives to authenticate the source address of IP packets, we propose to use a traceback system. IP traceback has several advantages over cryptography-based approaches. First, the header and processing overhead is significantly lower. Most traceback systems use as few as 16-25 bits to efficiently store marking information into rarely used header fields[1]. The processing overhead of packet marking algorithms is essentially simpler than computing digital signatures or message authentication codes (MACs). Furthermore, traceback systems do not waste routers processing power with the validation of legitimate traffic source, because the path reconstruction algorithm can be started only when necessary.

Although the traceback approach presents several advantages over cryptography-based approaches, existing traceback systems do not serve as a mechanism for identifying the origin of each packet, because they have one or more of the following limitations: (i) requirement of multiple packets to be able of reconstructing the attack path, (ii) use of a header overhead larger than it is possible to allocate in IP header, and (iii) storage of state at routers. Among the existing systems, due to paper space constraints, we describe in the following only those that are evaluated in this work.

Song and Perrig [7] proposed the use of the topology map to assist the traceback process. In this scheme, the attack path is reconstructed by performing a search on the graph that represents the network topology. The marking inserted by each router is the *hash* of the router IP address. This allows compressing the 32 bits of the IP address into an 8-bit identifier. During the path reconstruction, the marking extracted from the received packet is compared with the *hash* of the IP address of each checked router. The graph that contains all the routers detected in the path-reconstruction procedure is called reconstruction graph. During the path-reconstruction procedure a router not traversed by the attack packet may be incorrectly integrated to the reconstruction graph. This kind of error is said to be a false positive.

Belenky and Ansari [8] observed that only the first router needs to mark the packet in order to identify the attacker. They propose that the first router of the attack path inserts its ingress interface IP address into outgoing packets. Because the 32 bits that compose an IP address do not fit into the available marking space, they propose to split the router IP address into k fragments. The marking router randomly chooses one address fragment and inserts it on the packet. After receiving all the k address fragments, the victim can recover the address by correctly reassembling the received fragments. Hereafter, we consider $k = 4$.

Durresi et al. proposed the Fast Autonomous System Traceback (FAST) [6], an inter-domain traceback system in which the first 5 ASes mark the packet. The marking field is divided into 5 subfields to accommodate the 5 marks. A counter is also carried into the packet to inform each AS which subfield it should mark. The marking is made by the border router of each AS. To notify the victim about the presence of its AS in the attack path, the border router inserts into forwarded packets the *hash* of AS number (ASN) of its AS.

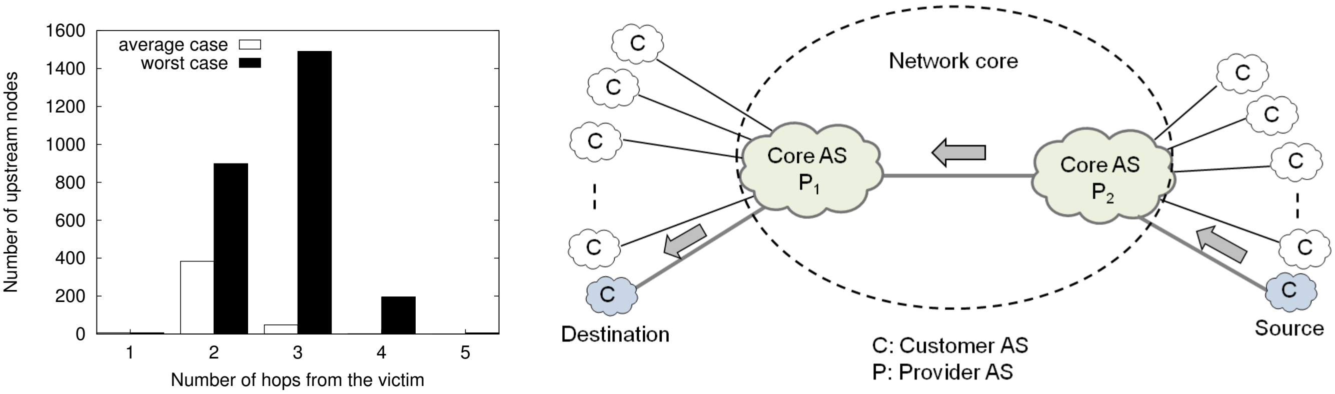
The previously mentioned systems depend on the reception of multiple packets to be able of reconstruction the path taken by the sent packets. Consequently, these systems cannot be used as a mechanism for identifying the origin of each individual packet. In addition, distributing the traceback information among multiple attack packets implies a rapid growth of the false-positive rate as the number of attackers increases. When there are lots of attackers, during the path reconstruction procedure, packets from different origins are believed to belong to the same attack path, leading to severe inaccuracies, as show the results presented in Section V. In comparison, the proposed system traces an unlimited number of attackers and finds less than one false positive per attacker, using only a single received packet.

III. The AS-level Path Reconstruction Problem

Inter-domain traceback [6], [13] has some advantages over the router-level traceback. AS-level paths are about 5x shorter than router-level paths and the number of ASes (about 50 thousand) is far less than the number of routers (on the order of tens of millions). Despite these advantages, inter-domain traceback presents a new challenge: to deal with a highly hierarchical structure. This leads to points of divergence during the path reconstruction procedure that may compromise the accuracy of the traceback system, as shown in the following.

The markings received by the victim do not directly indicate the ASes traversed by the attack flow. Thus, to reconstruct the attack path, the victim has to check if each upstream AS belongs to the attack path using the received markings. Fig. 1(a) depicts the number of ASes (of the Ark topology) tested by the path-reconstruction procedure according to the distance from the victim. We note that depending on the distance to the victim the number of tested ASes varies significantly. This is an expected result since the node degree distribution in AS-level topology follows a power law [14]. Most ASes have low degree and are located at the network edge, whereas minor ASes have a high number of neighbors and are located at the core. Thus, when the path-reconstruction procedure is started at the victim, there few neighbor ASes to be tested, given that we are at the network edge. If we move towards the network core, the number of neighbors that have to be tested increases. Likewise, if we keep increasing the distance,

[1] The marking information could be stored in IP options field, but this would cause packet fragmentation, beside of generating a high processing overhead, because adding options to packets requires a *slow path* forwarding. An alternative is to send the marking information in a separate packet, but this adds even more processing overhead in routers, reduces the network available bandwidth, and there is also the problem of authenticating those additional packets. Therefore, the most efficient solution is to overload rarely used fields of IP header, such as the 16-bit fragment identification (ID) field, the 8-bit type of service (TOS) field, and the 1-bit reserved fragment flag. Some studies show that less than 0,25% of Internet packets suffer fragmentation [12]. Hence, the negative impact of adopting a traceback system is small when compared to the brought benefits. In addition, it is possible to maintain the compatibility with fragmentation if the marking field content is the same for all fragments, so that they can be further reassembled at destination, given that they have the same ID field [8]. As argued by Dean et al. [12], the TOS field was designed to allow special handling of traffic, but setting this field to arbitrary values actually makes no measurable difference in packet delivery. Overloading the reserved fragment flag has also no prejudicial effect on current implementations.



(a) Number of upstream ASes that must be tested at each hop.

(b) The customer-provider AS-level hierarchy and the most common path pattern.

Fig. 1. Consequences of customer-provider hierarchy in AS-level path reconstruction.

we reach the edge of the network again and, therefore, the number of neighbor ASes decreases. This behavior reflects the customer-provider AS-level hierarchy. The customer-provider relationship is a consequence of commercial relations between ASes: a customer AS pays its provider AS to transport traffic from/to the customer AS. Commonly, a provider AS has lots of customer ASes, forming what is known as the customer-provider hierarchy, which can be seen in Fig. 1(b). This hierarchy is also seen in the AS-level paths characteristic: 62% of paths are only 3 hops long [14]. In the most common path pattern, a packet is originated in a customer AS, goes up to two core ASes and then it goes down to the customer AS of the destination. In Fig. 1(b), P_1 and P_2 are called core ASes because they are highly connected ASes.

According to Figs. 1(a) and 1(b), the critical steps during path reconstruction occur at second and third hops, where one must test on the worst case 898 and 1490 upstream ASes, respectively. We have a large number of ASes due to the concentration of paths that pass through the large providers, called core-ASes. Half of the Internet paths traverse the 10 ASes with more than 500 neighbors [14]. Departing from the victim towards the attacker AS, the path reconstruction procedure can easily find the victim's provider and enter the network core. Nevertheless, to go away from the network core requires testing almost one thousand ASes, which may result in a high false-positive rate. Thus, the first critical step in the path reconstruction is the identification of the first core AS of the attack path. The second critical step occurs when the reconstruction procedure reaches the attacker's provider and is trying to locate the attacker AS. In this step, the attacker AS must be identified among about one thousand ASes.

IV. The Proposed System

As Castelucio et al. [13], we propose to use the Border Gateway Protocol (BGP) as a vehicle to distribute the deployment information. Cooperative ASes, i.e., the system-deployed ASes, advertise their support for traceback in a BGP attribute in route advertisements. Hence, an overlay network is formed by cooperative ASes, allowing a given AS to determine whether it is the first cooperative AS of a given path or not[2].

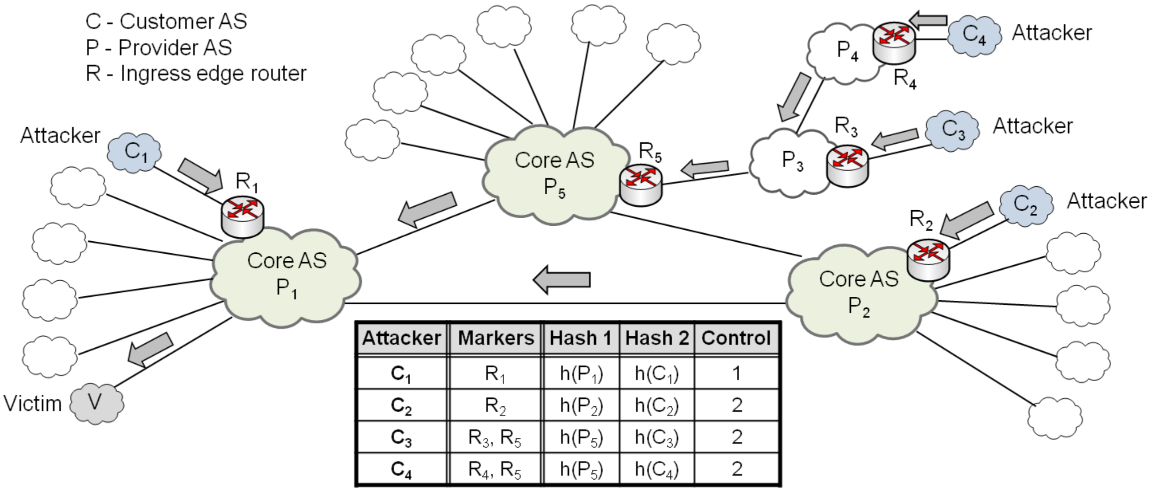
A. Packet Marking Procedure

Observing the existence of two critical steps during the path reconstruction, we propose a novel packet-marking scheme in which only two ASes mark the packet. These two markings are used as checkpoints to guide the path-reconstruction procedure. The checkpoints are strategically placed at the attacker AS and at the first core AS of the attack path. We propose to use only two markings, because in this way we can allocate more bits to the critical steps, guaranteeing that the checkpoints are reached with a low false-positive rate. To store the markings, we propose to overload 25 bits of IPv4 header. We divide the overloaded header fields into 3 subfields. Two fields, called `Hash 1` and `Hash 2`, 11 and 12 bits long, respectively, are used to accommodate the two AS markings. We also use a 2-bit `Control` field, whose purpose is explained in the following.

Fig. 2 shows how the marking procedure works. The figure presents four examples with different attack path lengths. In all the cases, we consider that customer ASes are not cooperative in order to show that the proposed procedure does not need that both the customer AS and its provider are cooperative. The shortest path is (C_1, P_1, V). In this example, the attack packet is originated in the customer AS C_1, goes to the provider AS P_1, which is also a core AS, and then reaches the victim V. The packet-marking procedure determines that, first of all, the ingress edge router of the first cooperative AS, R_1 in this example, erases all the marking fields in order to eliminate the initial condition of the packet, which is determined by the attacker. Then, R_1 inserts the hash of $ASN_{originAS}$, which is $h(C_1)$ in this case, into the `Hash 2` field. The second marking is done by the first core AS of the path, which is also P_1 in this example. For this reason, R_1 inserts $h(P_1)$ into the `Hash 1` field. R_1 also writes into the `Control` field the distance[3] from P_1 to the victim V, which is one hop in this case. Other example of attack path is (C_2, P_2, P_1, V). Again, the first cooperative AS, P_2, is also the first core AS of the attack path. Therefore, the ingress edge router of P_2, R_2, inserts $h(P_2)$ and $h(C_2)$ into `Hash 1` and `Hash 2` fields, respectively. The distance between P_2 and V is 2, and then this value is put into the `Control` field. After leaving P_2, the packet reaches P_1, which is also a core AS. Hence, P_1 checks

[2] Since it is possible to determine the first cooperative AS of a given path, a simple solution to identify the source of a packet is to put the AS number of the origin AS into the packet. But, the size of AS numbers has recently increased from 16 to 32 bits [15], which do not fit into the available space in IPv4 header. In addition, reconstructing the attack path with single 25-bit marking is clearly worse than using checkpoints.

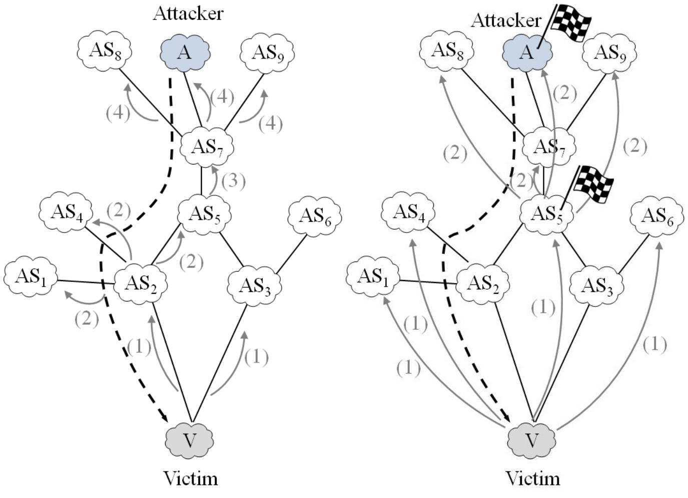
[3] The distance can be calculated by counting the number of elements in the BGP attribute called `AS_PATH`, which provides the list of the ASes needed to traverse before reaching a given destination.



Attacker	Markers	Hash 1	Hash 2	Control
C_1	R_1	$h(P_1)$	$h(C_1)$	1
C_2	R_2	$h(P_2)$	$h(C_2)$	2
C_3	R_3, R_5	$h(P_5)$	$h(C_3)$	2
C_4	R_4, R_5	$h(P_5)$	$h(C_4)$	2

Fig. 2. Example showing the markings received by the victim after the attack from customer ASes C_1, C_2, C_3, and C_4.

the value of the `Control` field. P_1 then verifies that the value is different than 0 and concludes that some core AS has already marked the packet. For this reason, P_1 just forwards the packet to the victim, without making any changes. The third example of attack path is (C_3, P_3, P_5, P_1, V). In this example, the first cooperative AS is P_3. Accordingly, R_3 inserts $h(C_3)$ into the `Hash 2` field. Next, the packet reaches the first core AS, P_5. Thus, the ingress edge router of P_5, R_5, marks the packet with $h(P_5)$ and updates the `Control` field with its distance to V, which is 2 in this case. The packet is then forwarded to the core AS P_1, which does nothing, because the value of the `Control` field is different than 0. Finally, the packet reaches the victim. The last example is the 5-hop path $(C_4, P_4, P_3, P_5, P_1, V)$. In this case, the first cooperative AS is P_4, and thus R_4 inserts $h(C_4)$ into the `Hash 2` field. The packet is then forwarded to P_3, which is neither the first cooperative AS nor the first core AS of this path. Thus, P_3 does not insert any marking. Next, the packet reaches the first core AS, which is P_5. Likewise, the ingress edge router R_5 marks the packet with $h(P_5)$ and updates the `Control` field with its distance to V, which is 2. The packet then goes to P_1, which just forwards it to the victim.



(a) Hop-by-hop procedure.

(b) Proposed procedure.

Fig. 3. Reconstruction of the attack path (A, AS_7, AS_5, AS_2, V).

B. Path Reconstruction Procedure

After receiving the marked packet, the victim can start the path-reconstruction procedure. The proposed procedure differs from the classical hop-by-hop reconstruction procedure, as shown in Fig. 3. The figure shows an example of a path reconstruction, considering the attack path (A, AS_7, AS_5, AS_2, V). In the classical hop-by-hop path-reconstruction procedure, as illustrated by Fig. 3(a), the victim checks first the ASes that one hop away (AS_2 and AS_3). In the example shown in Fig. 3(a), assuming that AS_3 is not a false positive, then only AS_2 is integrated to the reconstruction graph. In the following step of the path-reconstruction procedure, the ASes located at two hops from the victim, AS_1, AS_4, and AS_5, are tested. Only AS_5 is recognized as belonging to the attack path. Next, the ASes at the third hop are tested. In the example, AS_7 is tested and integrated to the reconstruction graph. Although AS_3 is a neighbor of AS_5, AS_3 is not tested, because its distance to the victim (1 hop) is less than or equal to AS_5 distance. We use a breadth-first search for the path reconstruction. Finally, the ASes at the fourth hop, AS_8, A, and AS_9, are tested and the attacker A is then found. On the other hand, the proposed procedure (Fig. 3(b)) skips some steps of the classical procedure and performs tests only on strategic points, where the checkpoints are located at. Hence, the proposed procedure skips the first hop and directly tests the ASes located at d (in the example, $d = 2$) hops from the victim, where d, the distance from the victim to the last AS that has marked, is the content of the field `Control` extracted from packet received by the victim. In the example shown in Fig. 3(b), the ASes tested in the first step are AS_1, AS_4, AS_5, and AS_6. At this moment, the checkpoint, which is the marking extracted from the received packet, is used to identify the correct AS, AS_5, with high accuracy. In the next step of the proposed procedure, all the ascendant ASes of AS_5 (AS_7, AS_8, A, and AS_9) are tested. After using the second checkpoint, the attacker A is finally found. We note that the proposed path-reconstruction procedure always finds the attacking-AS, provided that the attacker marking is included in the received packet. According to the proposed packet-marking procedure, this condition is satisfied if the attacking-AS or its provider is cooperative.

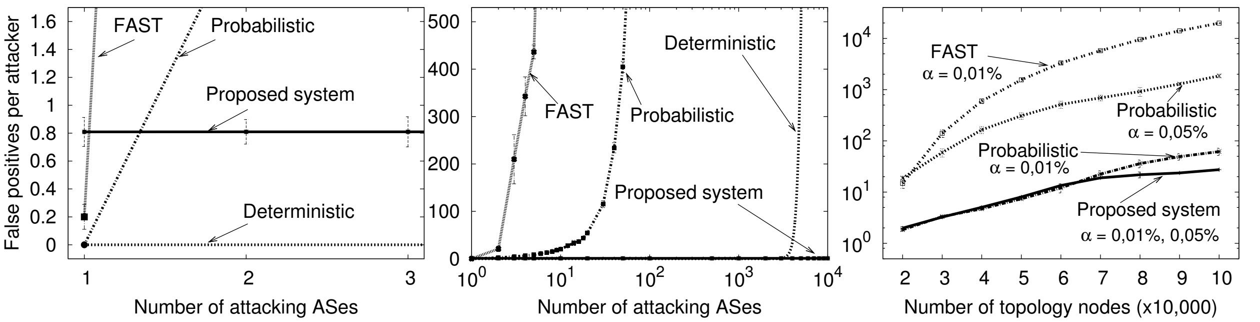


Fig. 4. Accuracy of the evaluated systems, measured by the number of false positives per attacker, as a function of the number of attacking-ASes and the topology size. The behavior of the proposed system is independent from the number and the percentage of attackers (α).

V. Simulation Results

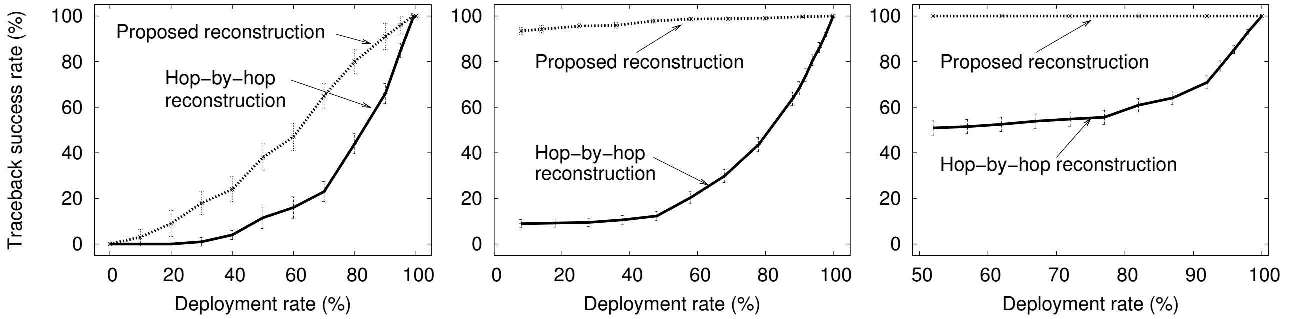
We developed a C++ simulator to analyze the performance of the proposed system. In our simulations, we use a real-world topology trace obtained from measurements data of the Ark (*Archipelago*) project [9]. The used topology is composed of 16,352 nodes and 39,346 links[4]. The DDoS attack, the packet marking, and the path reconstruction are simulated as follows. First, we randomly choose a victim from the set of nodes and then define loop-free attack paths. Next, we simulate the transmission of attack packets by inserting the appropriated markings into the fields, according to the ASes that compose each attack path. Once the packets are properly marked, the path reconstruction procedure starts at the victim.

We implemented in the simulator the FAST inter-domain system [6] and the probabilistic system proposed by Song and Perrig [6]. In addition, in order to compare also with a deterministic system, we present the analytical expression of the expected number of false positives for the system proposed by Belenky and Ansari [8]. Contrary to our system, these systems are not single-packet. Thus, the simulated attacks are composed of a sufficient number of packets for each system to work properly, i.e., 10 packets for FAST, 55 for the deterministic system, and 1,000 for the probabilistic scheme. Therefore, the victim receives all possible marks from each AS that belongs to the attack path. This represents the best possible scenario for the compared systems in terms of number of false positives.

The simulation results for the system accuracy are depicted in Fig. 4. The system accuracy is given by the number of false positives. An ideal traceback system finds the attacker without any false positive during the path reconstruction. Thus, the lower the number of false positives, the better the system accuracy in identifying the attacking-AS. The graph of Fig. 4(a) shows the accuracy as a function of the number of attacking-ASes. We observe that, when there is just one attacking-AS, all the compared systems obtain less false positives than the proposed system. This result was expected, since the FAST system, the probabilistic system, and the deterministic system take advantage of multiple attack packets, whereas the proposed system can rely only on the information of a single attack packet. Nevertheless, even with a simple increment in the number of attacking-ASes, the FAST and the probabilistic systems already exceed the proposed system in the number of false positives. With more than one attacking-AS, those systems begin to have problems with reconstruction errors caused by crossing the markings received from different attack paths. This problem becomes worse with the increase of the number of attacking-ASes, as shown in Fig. 4(b). Even the deterministic system, which maintains a low false-positive rate with few attackers, presents a number of false positives that quickly increases with the number of attacking-ASes from the order of thousands of attacking-ASes on. The proposed system, on the other hand, depends only on a single packet to trace each attacking-AS and, as a consequence, does not has the problem of incorrectly crossing the markings received from different attack paths. This fact is clear in Fig. 4(b), which shows that the proposed system is scalable with the number of attacking-ASes, because the obtained accuracy is constant with respect to the number of attackers. In addition, the number of false positives is indeed small, just 0.8 false positives per attacker. Fig. 4(c) shows the number of false positives as a function of the topology size, measured by the total number of nodes in the topology. We observe that FAST is the system less scalable with respect to the topology size, even considering a percentage of attacking-ASes of only $\alpha = 0.01\%$. For this same percentage, the probabilistic system scales almost as well as the proposed system. Nevertheless, by the curve with 0.05% of attacking-ASes, we realize that the probabilistic system does not scale so well, because the range of attackers changed from [2,10] to [10,50], which already implies a high number of false positives. In comparison, the proposed system is shown to be scalable with the number of node in the topology and this property is independent from the percentage of attacking-ASes. The reason for the scalability with respect to the topology size is that the proposed system exploits the customer-provider hierarchy, which is not done by the other systems in the literature. This property is important, because some studies show an exponential growth in the number of ASes. It is expected to exceed the number of 100 thousand in the next decade [15].

We also evaluate the traceback success rate, defined as the percentage of cases the attacker is found, as a function of the system deployment rate, as shown in Fig. 5. In Fig. 5(a), we consider a random distribution of cooperative ASes. We observe that in this case the proposed path-reconstruction procedure achieves a traceback success rate always better than the hop-by-hop procedure. The fact that the proposed procedure skips some steps of the hop-by-hop reconstruction makes the proposed system able to find the attacker even when some AS of the attack path is not cooperative. On the other

[4] The Ark project measurements are carry out through successive *traceroutes* and, therefore, only the paths actually used are captured, which explains the fact that the number of ASes of our topology is less than the total number of assigned ASNs (about 50 thousand) [15].



(a) Random distribution. (b) Core ASes + random distribution. (c) Non-customer ASes + random distribution.

Fig. 5. Traceback success rate as a function of the percentage of cooperative ASes for different deployment distributions.

hand, the hop-by-hop reconstruction is interrupted soon in the first non-cooperative AS found during the path-reconstruction procedure. We note that the curve of the proposed reconstruction approches the identity function, which shows that the proposed procedure attains to make the most of cooperative ASes, finding almost all of them. In Fig. 5(b) we simulate an scenario where the core ASes are cooperative and the remaining Ases follow a random deployment distribution. As core ASes we consider the ASes that have 7 or more neighbors, which represents about 8% of the ASes that compose the used topology. It is interesting to observe that the proposed reconstruction procedure attains to trace more than 90% of the ASes, with only 8% of deployment rate. Moreover, with 60% of deployment rate, the proposed reconstruction finds 100% of the attackers. The explanation for this is that, once found the first core AS of the attack path, it is enough that the attacking-AS or its provider is cooperative to the reconstruction procedure succeed. On the other hand, the hop-by-hop procedure, for deployment rates from 10 to 50%, finds only about 10% of the attackers, because this procedure only succeeds when the paths are exclusively composed of cooperative ASes. Finally, Fig. 5(c) shows the case where all non-customer ASes are cooperative and the customer-Ases follow a random deployment distribution. We consider as non-customer ASes the nodes that have 2 or more neighbors, which represents about 52% of the ASes that compose the used topology. In this scenario, the proposed reconstruction always finds the attacking-AS, regardless of the deployment rate of the customer ASes, performing again quite better than the hop-by-hop reconstruction, which is the mostly used.

VI. Conclusions

We propose in this work a stateless single-packet traceback system that uses only 25 bits in the packet header to store the attack path. Our system locates the origin of the attack with a constant accuracy regardless of the number of attack packets and attacking machines. Different from most proposals, our system does not rely on multiple received packets to reconstruct the attack path, and hence does not has the problem of incorrectly crossing the markings received from different attack paths. To overcome the challenge of storing the complete path in a small header space, we exploit the hierarchical structure of the Internet at AS-level. We analyze the path-reconstruction problem on the Internet at AS-level and obtain two main findings: (i) we show that the customer-provider hierarchy implies the existence of two critical steps during the path-reconstruction procedure, and (ii) we note that the origin AS can be found, even if some ASes do not participate in the packet marking. We evaluate the performance of our scheme in a real-world topology, comparing our system with three previous proposals, and our simulation results corroborate the accuracy and the scalability of our system. Our scheme individually traces an unlimited number of attackers with less than 1 false positive per attacker. The proposed path-reconstruction procedure traces more than 90% of the ASes if only the core ASes participate in marking, which corresponds to about 8% of system deployment rate in our simulation scenario.

Acknowledgments

This work was supported by CNPq, CAPES, FINEP, FUNTTEL, and FAPERJ.

References

[1] X. Liu, X. Yang, and Y. Xia, “NetFence: Preventing Internet Denial of Service from Inside Out,” in *ACM SIGCOMM*. ACM, 2010.

[2] T. Ehrenkranz and J. Li, “On the State of IP Spoofing Defense,” *ACM Trans. on Internet Tech.*, vol. 9, no. 2, pp. 1–29, 2009.

[3] D. G. Andersen, H. Balakrishnan, N. Feamster, T. Koponen, D. Moon, and S. Shenker, “Accountable Internet Protocol (AIP),” in *ACM SIGCOMM*, 2008, pp. 339–350.

[4] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, “Network Support for IP Traceback,” *IEEE/ACM Trans. Netw.*, vol. 9, no. 3, pp. 226–237, Jun. 2001.

[5] R. P. Laufer, P. B. Velloso, D. de O. Cunha, I. M. Moraes, M. D. D. Bicudo, M. D. D. Moreira, and O. C. M. B. Duarte, “Towards Stateless Single-Packet IP Traceback,” in *IEEE LCN*, 2007, pp. 548–555.

[6] A. Durresi, V. Paruchuri, and L. Barolli, “Fast Autonomous System Traceback,” *Journ. of Netw. and Comp. App.*, vol. 32, no. 2, pp. 448 – 454, 2009.

[7] D. X. Song and A. Perrig, “Advanced and Authenticated Marking Schemes for IP Traceback,” *IEEE INFOCOM*, vol. 2, pp. 878–886, 2001.

[8] A. Belenky and N. Ansari, “On Deterministic Packet Marking,” *Comput. Netw.*, vol. 51, no. 10, pp. 2677–2700, 2007.

[9] Y. Hyun, B. Huffaker, D. Andersen, E. Aben, M. Luckie, K. Claffy, and C. Shannon, “The IPv4 Routed /24 AS Links Dataset - Jul, 2009.” [Online]. Available: http://www.caida.org/data/active/ipv4_routed_topology_aslinks_dataset.xml

[10] P. Ferguson and D. Senie, “RFC 2827: Network Ingress Filtering: Defeating Denial of Service Attacks which Employ IP Source Address Spoofing,” 2000.

[11] X. Liu, A. Li, X. Yang, and D. Wetherall, “Passport: Secure and Adoptable Source Authentication,” in *USENIX NSDI*, April 2008.

[12] D. Dean, M. Franklin, and A. Stubblefield, “An Algebraic Approach to IP Traceback,” *ACM Trans. Inf. Syst. Secur.*, vol. 5, no. 2, pp. 119–137, 2002.

[13] A. Castelucio, A. Ziviani, and R. M. Salles, “An AS-level Overlay Network for IP Traceback,” *IEEE Netw.*, vol. 23, no. 1, pp. 36–41, 2009.

[14] P. Mahadevan, D. Krioukov, M. Fomenkov, X. Dimitropoulos, K. C. Claffy, and A. Vahdat, “The Internet AS-level Topology: Three Data Sources and One Definitive Metric,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 1, pp. 17–26, 2006.

[15] G. Huston, *32-bit Autonomous System Number Report*, Mar. 2009. [Online]. Available: http://www.potaroo.net/tools/asn32/