Single Packet AS Traceback against DoS Attacks

Abdullah Yasin Nur
Department of Computer Science
University of New Orleans
New Orleans, LA, USA 70148
Email: ayn@cs.uno.edu

Mehmet Engin Tozal
School of Computing and Informatics
University of Louisiana at Lafayette
Lafayette, LA, USA 70504
Email: metozal@louisiana.edu

***Abstract*—The Internet is every facet of our daily lives and becomes more pervasive every day. It is designed to forward packets with minimal intervention, including malicious packets. This design enables different attack types including Denial of Service (DoS), which is one of the most harmful cyber-attack types in the Internet. In this work, we propose an Autonomous System (AS) traceback packet marking scheme to infer AS level forward paths from attackers towards a victim site. We utilize the Record Route option of the IP protocol to implement our packet marking scheme. Traceback on the AS level has many advantages, including a significant reduction in the number of required packets to construct forward-paths from attackers toward a victim site, reduction in the number of routers that involves in the packet marking process, and lower packet size overhead to routers, comparing to interface level traceback. Our results show that a victim site can construct the AS level forward path from an attacker site after receiving a single packet. In our marking algorithm, we provide an encoding method to reduce the bandwidth usage. The proposed technique uses 96.91 bits on the average in the Record Route options field, whereas the unencoded version uses 153.96 bits on the average.**

***Index Terms*—Denial of Service attack, DoS, DDoS, AS Traceback, IP Traceback, Record Route**

I. Introduction

The Internet is a decentralized interconnected network of networks serving billions of people worldwide. The Internet is every facet of our daily lives and becomes more pervasive every day. It provides the primary communication medium for critical infrastructures including electricity, financial services, and transportation. It consists of tens of thousands of Autonomous Systems (ASes) connected to each other. An Autonomous System is defined as a group of networks that are operated by single or more operators with a well defined routing policy [19].

The Internet is designed to forward packets with minimal intervention, including malicious packets. This freedom in the architecture provides an excellent opportunity for attackers to deploy cyber-attacks towards various targets. The economical, social, and political impacts of the cyber-attacks have significantly increased. Denial-of-Service (DoS) attacks and its variants, such as Distributed Denial-of-Service attacks (DDoS), are one of the most perilous attack types in the Internet. DoS attacks aim to deplete the resources of a system until the system fails to provide services to intended users. Typically, the attack involves flooding a target by legitimate-looking traffic or sending an excessive amount of requests to overload the system to disrupt its services. A more severe version of the DoS attack is the DDoS attack, where a large number of hosts simultaneously attack a target. In DDoS, the perpetrators plan their attack by compromising multiple hosts through a common vulnerability and use all compromised hosts to flood the victim site. In October 2016, a large-scale DDoS attack targeted the DNS infrastructure of Dyn which caused major Internet platforms unaccessible including Amazon, Netflix, Twitter, PayPal, and the New York Times. In April 2018, attackers targeted GitHub with 1.35 Tbps which corresponds to 126.9 million packets per second. In April 2019, Imperva reported that one of its clients experienced DDoS attack with 580 million packets per second. Most recently, Amazon reported a world record DDoS attack sustaining a 2.3 Tbps to their Amazon Web Services (AWS).

Criticality of an AS is defined as the amount of potential traffic that it carries between the pairs of other ASes [3]. Cyber-attacks targeting critical ASes can cause larger-scale traffic disruptions in the Internet. Therefore, critical ASes must deploy several defense mechanisms covering different types of attacks. Filtering the attack traffic within the AS network is not a valid solution because it creates congestion in the incoming links. In case the AS deploys a filtering mechanism within its border, the attack traffic will still use the bandwidth of the incoming links to that AS. Moreover, the rogue traffic also affects intermediate ASes which carry the attack traffic. These ASes are unintentional victims. Hence, the main goal of a DDoS defense mechanism is to stop the attackers as close as possible to their sources to prevent malicious traffic in the Internet.

Inferring the path between two hosts from the destination site in the Internet is called IP traceback. The IP protocol does not directly support the traceback, which makes it a challenging problem. Packet marking [9] is suggested by researchers to solve the IP traceback problem. Packet marking is blemishing IP packets with routers' IP address information while the packet traverses the routers from the source toward the destination. One of the variations of the IP traceback is AS traceback which is tracing the AS path between attackers and the victims instead of the Interface level hop-by-hop path.

In this paper, we propose a single packet AS level traceback method. We modify the Record Route feature of the IP protocol. Instead of tracing all IP addresses between an attacker and a victim site, our method traces only ASes between them.

Traceback on the AS level has many advantages, including a significant reduction in the number of required packets to construct forward-paths from attackers toward a victim site, reduction in the number of routers that involves in the packet marking process, and lower packet size overhead to routers, comparing to Interface level traceback. Each AS in the Internet is assigned a 32-bit unique Autonomous System Number (ASN). In our method, the border router which is the first router belongs to a different AS in the path appends its corresponding ASN into the Record Route options field of a packet. After receiving a packet, the destination site aligns the sequence of the ASNs in the Record Route options field and determines the AS path from the attacker. We also present an encoding technique for storing the ASN to reduce bandwidth usage significantly.

We conducted several experiments using a real world dataset to demonstrate the efficiency of our approach under DoS attacks. Our results show that a victim site can construct the AS level forward path from an attacker site with a single packet. Our encoding technique utilizes 96.91 bits, on the average, in the Record Route options field. On the other hand, the unencoded version requires 153.96 bits, on the average. In addition, we compared our method with previously suggested methods [1, 14, 27, 28, 29]. To achieve a fair comparison, we divided the methods into three categories; (i) IP traceback, (ii) Probabilistic Packet Marking (PPM) based AS traceback, and (iii) Record Route option based AS traceback. We compare our method with IP traceback methods, RRIPTrace [1] and Fast Internet Traceback (FIT) [14], in terms of the average number of packets needed to construct a forward-paths graph. RRIPTrace needs 1 to 229 packets and FIT needs between 244 and 858 packets on the average to construct forward paths for varying hop distances. The overall averages are 1, 20.23, and 457.18 for our approach, RRIPTrace, and FIT, respectively, regardless of the hop distance. Note that, our approach constructs AS level forward-paths whereas RRIP-Trace and FIT constructs Interface level forward-paths. The comparison clearly shows that discovering AS level forward paths (AS traceback) requires less many packets than Interface level forward paths (IP traceback). Next, we compare our work with the Probabilistic Packet Marking (PPM) based AS traceback methods [27, 29] in terms of the average number of packets needed to construct a forward-paths graph. The first method uses PPM fixed probability (PPM-fixed) [29] and the second method uses dynamic probability (PPM-dynamic) [28] for packet marking. Our results show that PPM-dynamic needs between 3.55 and 64.24 packets and PPM-fixed needs 23.02 to 96.08 packets on the average to construct AS level forward paths for varying hop distances, whereas our method requires only 1 packet. The overall averages are 1, 27.26, and 46.04, for our approach, PPM-dynamic, and PPM-fixed, respectively, regardless of the hop distance. Finally, we compare our method with SRRT [28] which modifies the Record Route field to trace AS paths like our proposed method. Since the Record Route option allows us to trace most of the paths with a single packet, we compare our method with SRRT in terms of the size of packets. Our results show that SRRT inserts 140 to 332 bits on the average to the packet header to construct AS level forward paths for varying hop distances. On the other hand, our approach utilizes 41 to 148.48 bits on the average for the same path traces. The overall averages are 96.91 and 236.95 for our approach and SRRT, respectively, regardless of the hop distance.

The rest of the paper is organized as follows. In Section II, we present the related work. We explain the details of our approach in Section III. Section IV demonstrates our experimental results. Finally, we conclude the paper in Section VI.

II. Related Work

DoS defense mechanisms can be classified into four categories: attack detection, attack reaction, attack prevention, and attack source identification [1].

Attack detection mechanisms aim to detect the DoS attacks by monitoring the incoming traffic. Gil et al. [4] propose a data-structure to help routers and network monitors to detect volumetric DDoS attacks. They assume that the packet rate from Host A to Host B should be proportional to the packet rate from Host B to Host A during normal operations. Therefore, in case of a dramatic change in the packet rate from one side of the flow indicates a volumetric DDoS attack. Peng et al. [5] propose an attack detection technique by monitoring the source IP addresses. They assume that an extreme increase in the set of new source IP addresses indicates an attack.

Attack reaction techniques involve resource management to mitigate the impact of DoS attacks in a timely fashion. These type of methods are usually short term, but immediately effective solutions which require deploying redundant network service resources to distribute the attack traffic during an attack. High profile service providers, such as Microsoft and Yahoo, dynamically increase service and network resources during attacks [6]. The increasing popularity of cloud services brought new approaches to DoS defense [23, 24]. Cloud-based security companies such as Cloudflare and Imperva provide a cloud layer between their customers, which allows them to monitor and analyze traffic patterns in real-time. When a DDoS attack is detected by monitoring systems, they apply a filtering technique and drop the malicious traffic without forwarding to their customers.

Attack prevention techniques aim to control targeted attacks before they reach to the victims. Ingress Filtering [8] is a filtering technique which requires each AS checking the source IP addresses of the outgoing packets and filtering them if the source IP addresses do not belong to their IP address spaces to prevent IP spoofing. Kalkan and Alagoz propose a statistical packet filtering mechanism to defend a victim site against DDoS attacks [7]. The proposed method calculates each packet's score based on its attributes including IP address, port number, packet length, TTL value and TCP flags. Packets having a score below than a threshold value are filtered.

Finally, the last category is attack source identification. Due to the free nature of IP protocol, attackers can forge the attack by using IP address spoofing, which is creating

the packets with a false source IP address. If the victim site blocks the attack IP address, it may accidentally block one of its legitimate users. Therefore, it is necessary to be able to infer the forward paths from attack sites to the victim site. By finding the path, the victim site can know the exact point of the attacker even if the attacker spoofs its IP address. Inferring the path between two hosts from the destination site in the Internet is called IP traceback [9].

The IP traceback problem has been extensively studied in the last two decades [2, 10, 11, 13, 15]. One of the earliest work can be credited to Savage et al. [9]. They propose the Fragment Marking Scheme (FMS) which uses 16-bit IP ID field in the IP packet header to probabilistically mark the partial path information. The victim site uses marked packets to construct forward paths. Song and Perrig [12] proved that FMS approach requires many packets to construct the forward paths and introduces many false positives for DDoS attacks. They propose Advanced and Authenticated Marking Schemes (AMS) to reduce the number of required packets for constructing the forward paths as well as to decrease the false positive rate. In a later study, Yaar et al. [14] improve AMS by using more space for encoding and decreased the number of required packets to build forward paths from attackers towards the victim. In our previous work [1], we propose a novel probabilistic packet marking scheme to infer forward paths from attackers to a victim. We exploit the Record Route feature of the IP protocol. In the method, a router inserts its IP address in the Record Route options field of a packet with probability 1 as long as there is room. On the other hand, if there is no room in the Record Route options field, the router rewrites the field with probability p or skips rewriting with probability $1 - p$. The victim site starts from an empty forward-paths graph and gradually builds up the graph by incorporating the sub-paths from the received packets.

The popularity of IP traceback inspired a different approach which is called AS traceback [25, 26]. Different from the IP traceback, AS traceback schemes store the Autonomous System Numbers (ASN) instead of IP addresses. ASN used to be an unsigned 16-bit integer which has a range between 0 to 65,535. The drastic increase in the number of Autonomous Systems in the Internet required a change in ASNs. RFC 4893 [17] introduced 32-bit AS numbers in 2007 and RFC 6793 [18] updated the standard in 2012.

Parachuri et al. propose one of the earliest works in AS traceback [25]. The proposed method uses 16-bit IP-ID field to hold the ASN and 3-bit to mark the AS distance in the IP packet header. Each AS'es border routers on the path between an attacker and the victim site probabilistically mark the packet with the ASN information. Gao and Ansari propose AS-based Edge Marking (ASEM) [26] which applies the conventional Probabilistic Packet Marking (PPM) [9] method at the AS level. Different from the PPM, only ingress edge routers of each AS mark the packet and routers cannot mark packets already marked by an upstream router. Okada et al. [29] apply the PPM method for 32-bit ASN by using a fixed packet marking probability. Alenezi and Reed [27] propose a method that utilizes 25 bits in the IP header. In their method, each border router that receiving packet check its current Border Gateway Protocol (BGP) table and obtain AS hop distance between the destination and itself. The difference from the previous works is that they calculate the rewrite probability dynamically based on the AS Path distance from BGP. In their other work [28], they use the Record Route options field in IP protocol. The first ingress router closest to the attacker marks the packet by its IP address and AS path between the attacker and the victim site which is obtained from BGP. After the first router records the route, it sets a guard bit to prevent overwrite issues for other routers on the path. The method can trace at most 7 ASes by using 40 bytes in the IP option field and an additional 20-bits in the packet's header.

In this work, we propose a new AS-level packet marking scheme by utilizing the Record Route options field in the IP protocol. Different from the previous works, we propose an encoding method for ASN which reduces the bandwidth usage significantly, increases the maximum number of traceable ASes, and reduces the router overhead.

III. Methodology

In this section, we demonstrate the packet marking and encoding scheme to track the forward AS paths from attackers towards a victim site. We modify the Record Route feature of the IP protocol. Record Route option in IP protocol records the route of a packet. Each router relaying the packet to the next router appends its IP address into the options field. Therefore, the destination host gets the list of IP addresses of the routers that appeared on the path. Note that, the Record Route options field can store at most nine IP addresses [16]. Therefore, the destination host receives the IP addresses of the first nine hops, even if the path between the two hosts is more than nine.

IP traceback is useful to obtain the entire Interface Level path. However, the ASes are responsible for taking the filtering decisions within their internal network. According to our experiments, there are 15.43 hops on the average between two end hosts in the Internet (see Section IV for more details). On the other hand, those IP interfaces belong to 4.16 ASes on the average. Most of the interfaces belong to the ASes' backbone routers. Instead of keeping the entire IP path, keeping the AS path would be sufficient to find the sources of attackers. In this work, we develop an AS traceback mechanism to discover the AS path between attackers and a victim site. The proposed method will modify the Record Route options field in IPv4 protocol, yet it can be extended to IPv6 as well.

A router is a device that forwards a packet toward its destination in packet switching networks. In IP traceback methods, routers insert one of their interface IP address information to the packet header. In our approach, routers insert their AS information instead of IP addresses. Each AS in the Internet is assigned a unique number which is called Autonomous System Number (ASN). ASN is defined as a 32-bit unsigned integer. The method takes advantage of the unique specifier numbers where the routers insert ASN into the IP packets headers, instead of one of their IP addresses. Different from

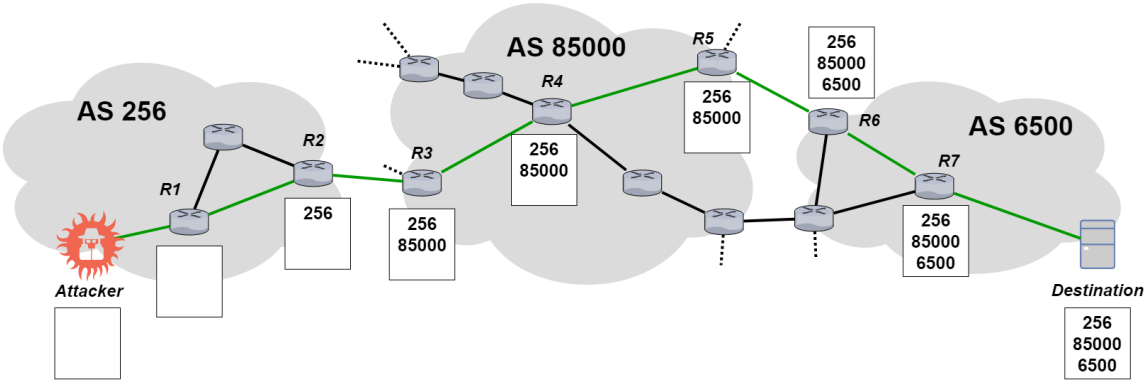


Fig. 1: An example attack case from a source host in $AS256$ to a destination host in $AS6500$

Ctr	ASN-256	Ctr	ASN-85000		Ctr	ASN-6500
0	0000 0001 0000 0000	1	0000 0000 0000 0001	0100 1100 0000 1000	0	0001 1001 0110 0100

Fig. 2: Encoded RROption field received by the victim site for ASPath =>256, 85000, 6500

the IP traceback schemes, only the ingress-routers of an AS inserts the ASN and the backbone routers skip-inserting the ASN value. Ingress-routers are the first routers that receive the traffic from the previous AS. To put in other words, the first router belonging to a different AS on a path appends its corresponding ASN into the Record Route options field of the packet. The destination site gradually joins the recorded ASNs in the options field to construct an AS level forward-paths graph from attackers toward the destination. Basically, the destination site starts with an empty forward-paths graph and fills the graph with paths that are reported via the Record Route option field of the received packets. After receiving a packet, the destination site aligns the sequence of the ASNs in the Record Route with respect to the current snapshot of the graph and implants the new route in the graph. The final graph contains all AS paths from source ASes toward the destination site.

Figure 1 presents an overly simplified topology and a path trace from a source host in AS256 to a destination host in AS6500. The ASes are illustrated as clouds and the route taken by a packet is shown in green. In accordance with our scheme, R1 inserts ASN 256 to the Record Route options field of the packet and forwards the packet to the next router. R2 skips inserting because its AS number is already inserted. When R3 receives the packet, it inserts ASN 85000. R4 and R5 skip inserting and transmit the packet to R6. R6 inserts 6500 in the Record Route options field. The procedure repeats itself until the packet reaches to the destination site. In the end, the victim site gets a packet that contains ASN 256, 85000, and 6500 in the Record Route options field.

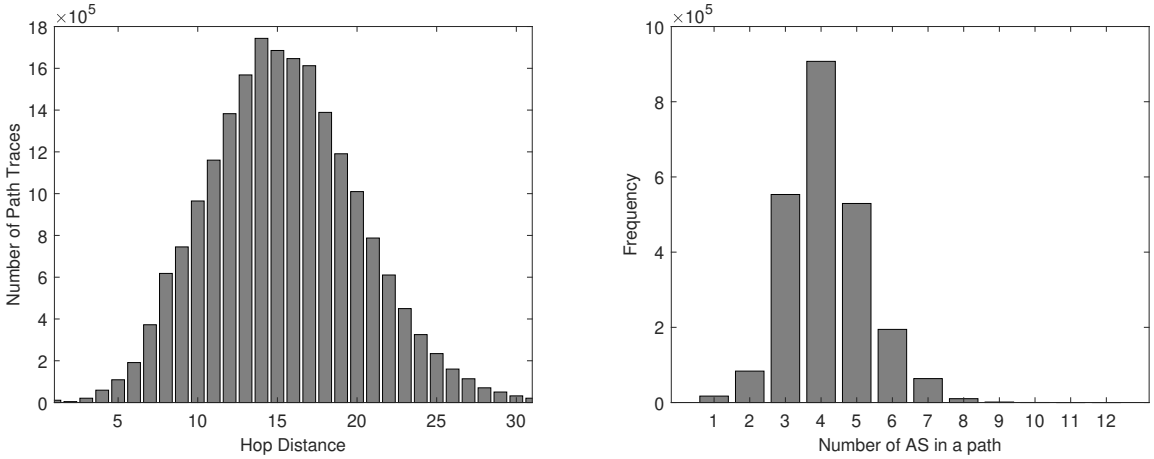
The Record Route feature in the IPv4 protocol can utilize up to 40-byte in the options field of a packet. Record Route uses 3 bytes overhead for controlling the process. The remaining 37 bytes can be used to record ASNs. Since each ASN is 4 bytes, the Record Route options field can hold at most 9 ASNs. According to our experiments, the majority of the paths involve 3 to 7 ASes in a path and 4.16 AS on the average in the Internet (see Section IV for more details). It shows that the victim site can find the forward path between an attacker to itself with a single packet for 99.5% of the cases.

A. *Encoding ASNs*

It is reported that 85% of the ASes are stub ASes which are virtually at the edge of the Internet without having any customer ASes [3]. Therefore, stub ASes are those ASes which do not transit any traffic belonging to other ASes. The remaining 15% of the ASes are ISPs that provide Internet access to other ASes. When we check the intermediate ASes' ASNs, we observe that 16-bit is enough to cover the majority of the ASes in the Internet. For example, the ASN of AT&T (tier-1) is 7018, CenturyLink (tier-1) is 209, and Hurricane Electric (tier-2) is 6939.

The main reason for this observation is that ASN used to be an unsigned 16-bit integer which has a range between 0 to 65,535. The drastic increase in the number of Autonomous Systems in the Internet required a change in ASNs. RFC 4893 [17] introduced 32-bit AS numbers in 2007, and RFC 6793 [18] updated the standard in 2012. Therefore, earlier ISPs keep their 16-bit ASN by extending it by padding zeros and making the ASN 32-bit. The gradual transition from 16 bits to 32 bits is proposed by Internet Engineering Task Force (IETF) in 2007. Right now, all ASNs are considered 32 bits [20]. Note that, the reason behind the 32-bit ASN assignment is to cover all possible future AS involvements in the Internet.

In our encoding scheme, we divide the ASN into 16-bit chunks. If 16-bit can cover the ASN, the first 16 bit will be all 0, which is not required to store. However, if the ASN is greater than 16-bit, we need to use both chunks. In order to decode the encoded number, we keep one control bit. The control bit is 0 if the ASN is represented by 16-bits, the control bit is 1 if the ASN is represented by 32-bits.



(a) Interface Level Hop Distance Distribution (b) AS Level Hop Distance Distribution

Fig. 3: Hop distance distribution of the Internet

Algorithm 1 Encoding ASN

Input: ASN ▷ AS Number of the current AS
Output: $RROptions$ ▷ updated Record Route options field
1: **if** ASN is smaller than 65536 **then** ▷ ASN is in 16-bit range
2: $controlBit = 0$
3: $insertToRROption(controlBit, lastTwoBytesOfASN)$
4: **else** ▷ ASN is in 32-bit range
5: $controlBit = 1$
6: $insertToRROption(controlBit, fourBytesOfASN)$
7: **end if**

Algorithm 2 Decoding AS Path

Input: $RROptions$ ▷ Final Record Route options field
Output: $ASPath$ ▷ An array holds consecutive ASNs in the path
1: **while** index is smaller than RROptions length **do**
2: $controlBit = RROptions[index]$
3: **if** controlBit equals to '0' **then** ▷ ASN is the next 16 bits
4: $ASNV = getNext16bits()$
5: $index = index + 17$ ▷ 1 control bit + 16-bit ASN
6: **else** ▷ ASN is the next 32 bits
7: $ASNV = getNext32bits()$
8: $index = index + 33$ ▷ 1 control bit + 32-bit ASN
9: **end if**
10: $ASPath.insert(ASNV)$
11: **end while**

Figure 2 shows an example of the encoding scheme for the attack case in Figure 1. The first router in AS256, R1, inserts the encoded ASN to the RROption field. Since ASN 256 can be represented with 16-bit, it inserts 0 as a control bit and 16-bit binary number representing 256. Once the first router in AS85000, R3, receives the packet, it inserts 1 as a control bit because 85000 cannot be represented by 16-bit. Finally, the first router receives the packet in AS6500, R6, inserts 0 and its 16-bit AS number into the Record Route options field. The victim site receives 67-bit field instead of getting 96-bit unencoded version.

Algorithm 1 presents the encoding pseudocode executed by ingress routers of the ASes. The algorithm expects the ASN to be reported by the router. Line 1 checks the size of the ASN. Remember that, the range of 16-bit is from 0 to 65535. Therefore, if an ASN is less than 65535, the algorithm executes line 2, which is the 16-bit ASN case. On the other hand, if the ASN is greater than 65535, line 5 updates the control bit, which represents the 32-bit case. *"insertToRROption"* is a function that appends the binary sequence to the Record Route options field of the packet.

Algorithm 2 presents the decoding pseudocode executed by the victim site. The algorithm expects an RROption field of the packet. We assume that the RROption field is an array where each element represents a bit. The algorithm returns the ASPath, which contains the consecutive ASNs. Line 1 starts the index from the first control bit of the RROption and loops until the end. Line 2 assigns the control bit from the RROptions. Line 3 checks if it is a 16-bit case or 32-bit case. In line 4, the algorithm uses a simple method "getNext16bits()" and assign those 16 bits to the variable ASNV. Line 5 makes the current index plus 17 because of 1 bit for control and 16 bits for ASN. The algorithm executes the lines 6 to 9 if the ASN is 32-bit. Finally, the algorithm inserts AS Numbers to ASPath array in line 10.

IV. Experimental results

In this section, we empirically demonstrate the efficiency of our algorithm using a real world dataset. We used the CAIDA IPv4 Prefix-Probing Traceroute Dataset [21] consisting of more than 20 million (20,377,233) path traces. The dataset consists of $899,916$ different IP addresses. Note that we only included the loop-free path traces that reach to their specified destinations. The minimum and maximum Interface level hop lengths in our dataset are 1 and 31, respectively. The average hop length is 15.43 and the hop length distribution is symmetric-like as shown in Figure 3a.

We used RouteViews prefix to AS mapping dataset obtained from CAIDA [22]. In order to generate an AS Level Internet

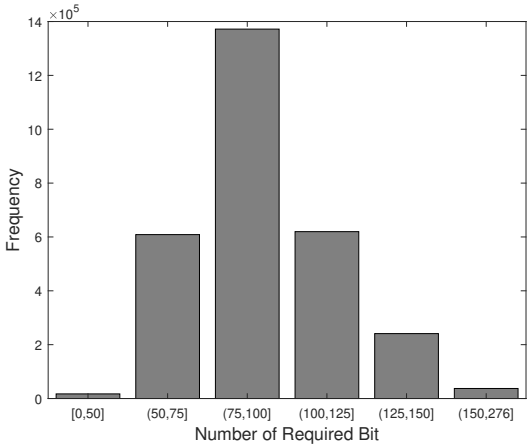


Fig. 4: Average Record Route Options Field Size Distribution

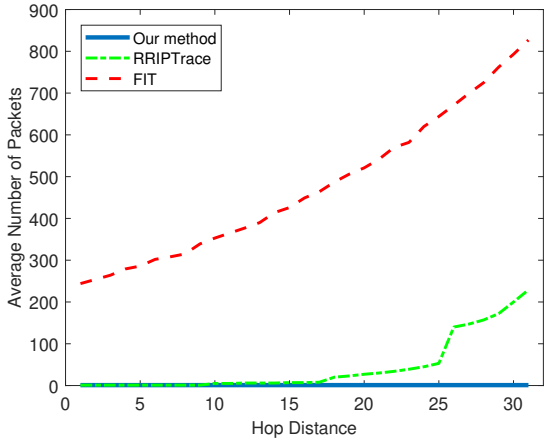


Fig. 5: Our approach compared to IP Traceback methods

topology, we mapped IP addresses reported in the traceroute dataset to their corresponding ASes. The dataset consists of 39,148 different ASes. The minimum and maximum AS level hop lengths in our dataset are 1 and 12, respectively. The average AS level hop length is 4.16 and the distribution is shown in Figure 3b.

In our experiments, we checked each AS path reported by the traceroute. Our results show that our method can trace all reported routes with a single packet. Figure 4 shows the distribution of the average Record Route options field size for all traces. The x-axis presents the average number of bits utilized in the Record Route options field and the y-axis presents the frequency in all traces. Our method is able to trace the attacker with a minimum of 41 bits and a maximum of 276 bits. On the average, we are able to trace the attackers with 96.91 bits. Note that, all number of bits represented here contains 24 bits Record Route overhead. Since the proposed method traces the attacker path with a packet, the method does not produce any false positive or false negative. If we do not use the encoding scheme and insert the 32-bit AS numbers in the Record Route options field, we are unable to trace 0.5% of the paths in our dataset. For the unencoded version, the minimum packet size is 56 bits and the maximum packet size is 312 bits. The average number of bits required to find an attacker is 153.96. Therefore, our encoding method reduces the bandwidth usage by 1.59 times on the average compared to the unencoded Record Route method.

We compare our method with previously suggested methods [1, 14, 27, 28, 29]. To achieve a fair comparison, we divided the methods into three categories; (i) IP traceback, (ii) Probabilistic Packet Marking (PPM) based AS traceback, and (iii) Record Route option based AS traceback.

A. Comparison with IP Traceback Methods

In the following, we show the efficiency of AS traceback over IP traceback methods, as we discussed in Section III. We compare our results to IP traceback packet marking schemes: Record Route IP Traceback (RRIPTrace) [1] and Fast Internet Traceback (FIT) [14]. In our previous work, RRIPTrace exploits the Record Route feature of the IP protocol to implement a probabilistic packet marking scheme for IP traceback. Also, we implemented FIT's 4/3 (n/n_{map}) scheme with 95% probability of reconstruction.

We repeated the experiment presented in our previous work RRIPTrace [1]. We randomly picked a single attacker and a victim IP addresses to mimic a DoS attack. Next, we ran our Matlab procedure which emulates the DoS attack. We gradually constructed the forward-paths graph from the attacker IP address toward the victim IP address. To achieve a fair comparison, we started the emulation at time zero for all methods where the victim site does not have any information about the topology. For each hop distance in the traceroute dataset, we selected 5000 random path traces where the source of the trace is an attacker and the destination is the victim site. Then, we computed the average number of required packets to build forward paths by repeating the experiment 1000 times. To calculate our method results, we mapped the IP addresses reported in path traces to their corresponding ASes. After that, we applied our method presented in Section III and trace the AS level paths.

Figure 5 shows that RRIPTrace needs 1 to 229 packets and FIT needs between 244 and 858 packets on the average to construct forward paths for varying hop distances. The overall averages are 1, 20.23, and 457.18 for our approach, RRIPTrace, and FIT, respectively, regardless of the hop distance. IP traceback methods discover the Interface level paths, whereas our new AS traceback method discovers the AS level paths between attackers and a victim site. Compared to collecting the IP addresses via IP Traceback and mapping them to their AS numbers, AS Traceback exhibits significant gain in terms of the number of required packets to construct the AS Level forward-paths graph from attackers towards a victim site.

B. Comparison with PPM based AS Traceback Methods

In the following, we compare our work with the Probabilistic Packet Marking (PPM) based AS traceback meth-

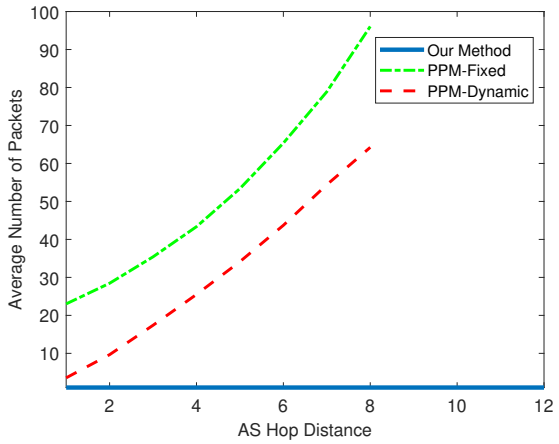


Fig. 6: Comparison with PPM based AS traceback methods

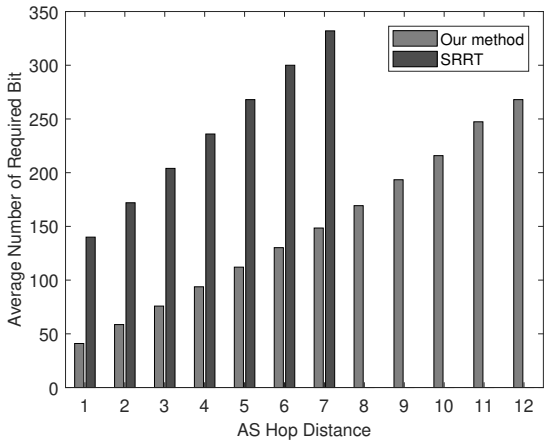


Fig. 7: Comparison with SRRT in terms of packet overhead

ods [27, 29]. These type of methods apply the traditional PPM IP traceback [9] technique to AS level. They utilize the 16-bit IP-ID field and additional fields in the packet's header. Therefore, they use less bandwidth compared to Record Route based methods. However, they require many packets to trace the AS path between attackers and a victim site.

We compare our method with PPM fixed probability [29] and dynamic probability [28] for packet marking. We implemented the method in [29] using a fixed probability $p = 0.092$, as suggested in the paper. Also, we implemented the method in [28] where the marking probability is calculated dynamically based on the distance between attackers and the victim. The distance field is represented by 3 bits which gives the maximum number of traceable AS count as 8.

Figure 6 shows that PPM-dynamic needs 3.55 to 64.24 packets and PPM-fixed needs 23.02 to 96.08 packets on the average to construct AS level forward paths for varying hop distances, whereas our method requires only 1 packet. The overall averages are 1, 27.26, and 46.04 for our approach, PPM-dynamic, and PPM-fixed, respectively, regardless of the hop distance. On the other hand, the PPM-fixed method uses 16-bit IP ID field, PPM-dynamic uses 25-bit in the IP packet header, and we use 96.91 bits on the average. Therefore, our approach consumes more bandwidth compared to them.

C. Comparison with Record Route Option based AS Traceback Methods

Similar to our proposed method, SRRT [28] aims to trace AS paths by modifying the Record Route options field. Record Route options field provides a 40-byte field which allows tracing most of the paths with a single packet. Therefore, we compare our method with SRRT in terms of the size of the packets.

In SRRT, the first ingress router closest to the attacker marks the packet by its IP address and AS path between the attacker and the victim site which is obtained from the router's current BGP table (AS-PATH field). After the first router records the route, it sets a guard bit to prevent overwrite issues for other routers on the path. The method can trace at most 7 ASes by using 40 bytes in the Record Route options field along with an additional 20 bits in the packet's header. SRRT cannot trace if the AS path is more than 7 hops. However, this is not a crucial drawback since the majority of the AS paths in the Internet is less than 8 hops as shown in Figure 3b.

Figure 7 shows that SRRT uses 140 to 332 bits, on the average in the packet header to construct AS level forward paths for varying hop distances. Note that, the numbers include 24 bits Record Route overhead and 20 bits additional overhead in the packet header that SRRT uses. Our approach uses 41 to 148.48 bits on the average for the same path traces. Moreover, the maximum number of utilized bits is 276 for AS level hop distance 12. The overall averages are 96.91 and 236.95 for our approach and SRRT, respectively, regardless of the hop distance. Therefore, our encoding method reduces the bandwidth usage by a factor of 2.45 on the average compared to SRRT.

V. Discussions

A. DDoS Attack Case

In this work, we show our results by mimicking a DoS attack. Since the average number of required packets is 1, DDoS attacks can be considered as individual DoS attacks from several attackers at different ASes. Therefore, the numbers shown in the experimental section is linearly correlated with the attacker size. For example, finding n-attackers requires n packets overall since the victim size can construct an AS level forward paths graph by receiving 1 packet from each attacker.

B. Multiple Origin Autonomous System Case

Border Gateway Protocol (BGP) is an exterior gateway protocol for exchanging IP prefix reachability information among the ASes in the Internet. An AS willing to deliver traffic to an IP address prefix originates a BGP advertisement declaring the prefix. A Multiple Origin Autonomous System (MOAS) conflict occurs if more than one ASes advertise the same prefix [30]. We used RouteViews prefix to AS

mapping dataset obtained from CAIDA [22] which addresses this problem by showing both of the ASes, e.g., ASX_ASY. In our experiments, we only considered the first ASN, e.g., ASX.

VI. Conclusions

A broad spectrum of attacks target the Internet's communication infrastructure to disable or disrupt the network connectivity and traffic flow until recovery processes take place. Denial of Service (DoS) attack and the variants such as Distributed DoS (DDoS) are one of the most harmful cyber-attack types in the Internet. In this work, we proposed an AS traceback scheme to infer AS level forward paths from attacker sites to a victim site. We exploited the Record Route option of the IP protocol. In our method, only the ingress routers of ASes insert their AS numbers. The packet delivered to the destination holds the AS path between the source and the destination. Our experimental results show that our method can trace all AS paths with a single packet in the Internet due to the low Internet diameter at the AS level. Moreover, we introduce an encoding technique which reduces the bandwidth usage significantly. Our approach uses 41 to 276 bits, on the average, for varying AS level hop distances. The overall average is 96.91 bits, regardless of the hop distance.

References

[1] A. Y. Nur and M. E. Tozal, "Record Route IP Traceback: Combating DoS Attacks and the Variants", Computers & Security 72 (2018): 13-25

[2] A. Y. Nur and M. E. Tozal, "Defending Cyber-Physical Systems Against DoS Attacks", IEEE SMARTCOMP, 2016

[3] A. Y. Nur and M. E. Tozal. "Identifying Critical Autonomous Systems in the Internet", The Journal of Supercomputing 74.10 (2018): 4965-4985

[4] M. T. Gil and M. Poletto, "MULTOPS: A Data-Structure for Bandwidth Attack Detection", USENIX Security Symposium, 2001

[5] T. Peng, C. Leckie, and K. Ramamohanarao, "Proactively Detecting Distributed Denial of Service Attacks Using Source IP Address Monitoring", International Conference on Research in Networking, Springer, 2004

[6] T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems", ACM Computing Surveys, 2007

[7] K. Kalkan and F. Alagoz, "A Distributed Filtering Mechanism Against DDoS Attacks: ScoreForCore", Computer Networks 108 (2016): 199-209

[8] F. Baker and P. Savola, "Ingress Filtering for Multihomed Networks", RFC 3704, 2004

[9] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Network Support for IP Traceback", IEEE/ACM Transactions on Networking 9.3 (2001): 226-237

[10] B. Al-Duwairi and G. Manimaran, "Novel Hybrid Schemes Employing Packet Marking and Logging for IP Traceback", IEEE Trans. Parallel Distrib. Syst., vol. 17, no. 5, pp. 403- 418, May 2006

[11] M.T. Goodrich, "Probabilistic Packet Marking for Large-Scale IP Traceback", IEEE/ACM Trans. Networking, vol. 16, no. 1, pp. 15-24, Feb. 2008

[12] D. Song and A. Perrig, "Advanced and Authenticated Marking Schemes for IP traceback", IEEE INFOCOM, April 2001.

[13] V. A. Foroushani and A. N. Zincir-Heywood, "TDFA: Traceback-based Defense against DDoS Flooding Attacks," IEEE 28th International Conference on Advanced Information Networking and Applications, pp.597-604, May 2014.

[14] A. Yaar, A. Perrig, and D. Song, "FIT: Fast Internet Traceback," IEEE INFOCOM, March 2005.

[15] A. Belenky and N. Ansari, "On Deterministic Packet Marking", Computer Networks, vol. 51, no. 10, pp. 2677-2700, 2007

[16] RFC 791 - https://tools.ietf.org/html/rfc791

[17] RFC 4893 - https://tools.ietf.org/html/rfc4893

[18] RFC 6793 - https://tools.ietf.org/html/rfc6793

[19] RFC 1930 - https://tools.ietf.org/html/rfc1930

[20] ARIN - Autonomous System Numbers - https://www.arin.net/resources/guide/asn/

[21] The CAIDA IPv4 Prefix-Probing Traceroute Dataset - 2020/02/02, https://www.ImpactCyberTrust.org, DOI 10.23721/107/1354205

[22] Routeviews Prefix to AS mappings Dataset for IPv4 and IPv6 - 2020/02/02, https://www.caida.org/data/routing/routeviews-prefix2as.xml

[23] O. Osanaiye, K. K. R. Choo, and M. Dlodlo, "Distributed Denial of Service (DDoS) Resilience in Cloud: Review and Conceptual Cloud DDoS Mitigation Framework", Journal of Network and Computer Applications 67 (2016): 147-165.

[24] Q. Jia, H. Wang, D. Fleck, F. Li, A. Stavrou, and W. Powell, "Catch Me If You Can: A Cloud-Enabled DDoS Defense", IEEE/IFIP International Conference on Dependable Systems and Networks, 2014

[25] V. Paruchuri, A. Durresi, R. Kannan, and S. S. Iyengar, "Authenticated Autonomous System Traceback", 18th International Conference on Advanced Information Networking and Applications, IEEE, 2004

[26] Z. Gao and N. Ansari, "A Practical and Robust Inter-domain Marking Scheme for IP Traceback", Computer Networks 51.3 (2007): 732-750

[27] M. Alenezi and M. J. Reed, "Traceback of DoS over Autonomous Systems", International Journal of Network Security and Its Applications 5.2 (2013): 131

[28] M. Alenezi and M. J. Reed, "Selective Record Route DoS Traceback", International Conference on Risks and Security of Internet and Systems (CRiSIS), IEEE, 2013

[29] M. Okada, Y. Katsuno, A. Kanaoka, and E. Okamoto, "32-bit AS Number Based IP Traceback", Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IEEE, 2011

[30] X. Zhao,D. Pei, L. Wang, D. Massey, A. Mankin, S. F. Wu, and L. Zhang, "An Analysis of BGP Multiple Origin AS (MOAS) Conflicts", Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement. 2001.