Investigating Unique Flow Marking for Tracing Back DDoS Attacks

Vahid Aghaei-Foroushani and A. Nur Zincir-Heywood
Faculty of Computer Science
Dalhousie University
Halifax, NS, Canada
Email: {vahid, zincir}@cs.dal.ca

***Abstract*—In this paper, we outline the recent efforts of our research in defense against Distributed Denial of Service (DDoS) attacks. In particular, we present a novel approach to IP traceback, namely Unique Flow Marking (UFM), and we evaluate UFM against other marking schemes. Our results show that the UFM can reduce the number of marked packets compared to the other marking schemes, while achieving a better performance in terms of its ability to trace back the attack.**

Keywords*—*Flow Based IP Traceback; DDoS Attacks; Security

I. Introduction

A DDoS attack is an explicit attempt to make a network or a system unavailable for use by its legitimate users. Defending against DDoS attacks is extremely difficult because the source identity is hidden by filling IP header fields with randomized values, known as IP spoofing or by using a NAT, or a proxy device. In such cases, when the administrator of the victim network tries to block all traffic from the apparent attack source, he is in fact taking action against innocent systems, thereby contributing to the (D)DoS attack. So in such cases, IP traceback is a critical ability for identifying sources of attacks and instituting protection measures for the Internet.

In this paper, we propose a new approach to IP traceback called Unique Flow Marking (UFM) to improve upon our previous work Deterministic Flow Marking (DFM) [1]–[4]. Unlike DFM that marks every flow by embedding the source identity data to the first packets of each flow, UFM marks only the unique flows of the network traffic. The main concept behind the UFM approach is that once the victim finds the origin of a packet in a flow, then the origin of any other packet in that flow, and all packets in any other flow with the same flow identity is also discovered. In comparison to the DFM and the packet based traceback approaches, we aim to minimize the number of marked flows and maximize the traceback rate via UFM. Indeed, there is a trade-off between the marking rate and the traceback rate in such techniques. To this end, we perform a sensitivity analysis on this trade-off of UFM and compare it against the existing techniques in the literature including DFM.

II. Literature review on IP traceback

In this section, we summarize the traceback schemes found in the literature based on the basic principle, the processing mode, and the location these schemes employ [5].

Using the basic principle, most of the existing traceback schemes can be summarized into logging and marking methods. In logging methods, the routers keep track of specific packet information [6]. One of the major problems of the logging method is the requirement for high amount of memory and CPU usage on the routers that are on the attack paths [7]. On the other hand, in marking methods, some or all routers in an attack path send specific information along with traveling packets. This way, the destination node can use this specific information to trace the attacker. This information could be sent by either embedding it in the IP header of the packets or by generating new packets [8]–[10].

Using the processing mode, most of the existing traceback can be summarized into two methods: deterministic and probabilistic. In deterministic methods, regardless of the marking or logging, every packet should be processed at both the source and the destination nodes. These methods require more processing overhead but they are able to provide a higher accuracy [6], [9]. Most of the recent traceback methods are probabilistic. While the required bandwidth and processing time in these methods are less than the ones required by the deterministic methods, the complexity for reconstruction at the destination side is higher. Some well-known examples of probabilistic methods are PPM [8] and its variants [11].

Finally, using the locations information, most of the existing traceback schemes can also be summarized into two groups: those that send traceback information by the edge routers closest to the source (source group) and those that send traceback information by some or all routers on the attack path (network group). Most of the current traceback methods belong to the network group [8].Their objective is to identify the attack path entirely or partially [11]. The drawbacks of these methods are the involvement of many routers along the attack paths and the cost in terms of processing times and memories [6], [10]. On the other hand, the goal of the source group methods is to identify the attack source. However, they do not identify the attack path [9].

III. Unique Flow Marking, UFM

UFM is developed based on our previous proposal, DFM [1]–[3]. DFM marks every flow, instead of every packet. For DFM, we have defined the Flow ID, which can be extracted from the traveling packets, as the five tuples of source IP address, destination IP address, L4 protocol type (TCP/UDP), source port and destination port numbers for TCP and UDP flows. For ICMP flows, it has been defined as the six tuples of

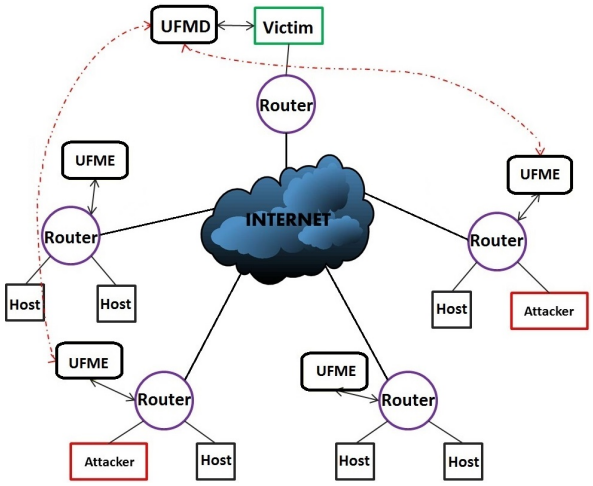

Fig. 1: The locations of UFMD and UFME modules on a sample network.

source IP address, destination IP address, L4 protocol type (ICMP), ICMP type, ICMP code and ICMP ID. Only the ingress interfaces of the edge router marks the flows and the rest of the routers, including the backbone routers, do not involve in the flow marking. All of the previous IP traceback methods in the literature require the participation of all or some routers in the attack path. However, DFM aims to minimize this issue by requiring only one participating router in the attack path.

Although DFM shows good traceback performance such as high traceback rate, low marking rate and low bandwidth usage, this performance is achieved by marking all the flows. To reduce this overhead, we propose the Unique Flow Marking, UFM, IP traceback approach. The main concept that leads us to the UFM approach is that we have seen that usually there are some flows in the traffic traces that share the same flow ID. If one of these flows at the source edge router can be found and marked, then the victim is able to traceback all of the packets in all of the flows that share the same flow ID. In this case, UFM may reduce the processing and memory overheads of IP traceback at both sides of the source and the destination networks. In fact, UFM aims to find and mark only one flow out of all the flows that have the same flow ID. UFM falls into the following category of classification: Basic principle: Marking; Processing modes: Deterministic at flow level; Location: Near the source. In the following, we describe the two modules of UFM: UFM Encoding (UFME) and UFM Decoding (UFMD) modules.

A. UFM encoding module (UFME)

This module is located on the edge routers and its goal is to mark the flows, Figure 1. In UFM, only the edge routers mark the packets, and the others, including the core routers, are not involved in the marking process. The UFME module uses three identifiers to mark the packets in order to trace up to the attacker node. These three identifiers are as the following:

The IP address of the egress interface of the edge router (32 bits): An edge router is the closest router to the attacker node with at least one valid assigned IP address to its egress interface.

The network interface identifier, NI-ID, (12 bits): This is an identifier assigned to each interface of either the MAC address of a network interface on the edge router, or the VLAN ID of a virtual interface if the edge router uses VLAN interfaces. The NI-ID specifies from which subnet a traffic flow comes.

Node-ID (16 bits): An identifier assigned to each source MAC address observed on the incoming traffic from local networks. Each MAC has a unique Node-ID.

The UFM identification data consists of the IP address of the egress interface (32 bits) + NI-ID (12 bits) + Node-ID (16 bits) = 60 bits, to distinguish the traffic of a particular node from the other nodes. The UFME module at the source side marks each unique flow (in terms of its flow ID). In doing so, the identification data of a flow should be divided into K fragments, and the first K packets of the flow should carry these fragments. Once a flow is marked, then any other flow with the same flow ID does not need to be marked. Therefore, each selected packet carries $M = 60/K$ bits of the identification data and $S = log2(K)$ bits required to identify a fragment of the identification data. UFM also embeds one flag bit, F, to each selected packet to distinguish between marked and unmarked packets. Therefore, each identification data fragment, by which each selected packet should be marked, consists of: M bits for the identification data fragment, S offset bits to represent 2^S possible fragments and one bit, flag F, which should be set to 1 for the marked packets and 0 for the rest.

The UFME module maintains a table to keep track of the marked packets, their flow IDs, and their embedded marks. This table is called the FragTable. Once the UFME module observes an outgoing packet, first of all it extracts its flow ID and then looks for the existence of a table record for this flow ID in the FragTable. If this flow ID is not in the FragTable, it means that this flow is new to the UFME module, so the UFME module takes the following steps: (i) Creates a table record for the new Flow ID in the FragTable; (ii) Calculates the flow identification data; (iii) Divides the flow identification data into K fragments and inserts them into the FragTable; (iv) Marks the selected packet by one of the identification data fragments. However, if this flow ID is already in the FragTable, it means that this flow has been already seen by the UFME module. In this case, if all of the identification data fragments of this flow have been already sent, then the UFME module just forwards the packets. Otherwise, the UFME module marks the packet by one of the unsent fragments in the FragTable and then forwards it.

As described before, the identification data (60 bits) should be divided into K fragments, and each fragment contains $M = 60/K$ bits of the identification data, $S = log2(K)$ bits to address each fragment, and 1 bit flag to differentiate between the marked and the unmarked packets. Based on the work in [1] [2], the best value of K is either 2, so that the size of each fragment is 32 bits, or 5, so that the size of each fragment is 16 bits. Moreover, based on several previous studies on the IP traceback [10], it is possible to use the identification and the flag fields of the IP header as a 16-bit marking field, or to use the identification, the flag and the fragment offset fields of the

IP header as a 32-bit marking field. Fortunately, the use of the fragment and the identification fields in the IP header affects only the 0.06% of the legitimate packets [10]. Selecting the value of K between 2 or 5 is a tradeoff between the marking rate and the number of required marking bits in the IP header. Selecting the lower value of K causes lower marking rate (which is desirable) but needs more space in the IP header (which is undesirable). If K=5 is selected, the UFME module uses 16 bits of the IP header to mark the selected packets, but if K=2 is chosen, then the UFME module marks 32 bits of the IP header.

B. UFM decoding module (UFMD)

The UFMD module is located at the destination network and its goal is to infer the origin of the incoming traffic, even if the source IP addresses of the incoming packets are spoofed, Figure 1. This module maintains a table for matching the flow ID and K possible identification data fragments of a flow, called ReconTable in order to reconstructs a valid source identification data. The UFMD module checks for the flag field (which is set by the UFME module) in the packet's IP header of the incoming packets to find the marked packets. Once the UFMD module finds a marked packet, it extracts its flow ID and then checks for the existence of a table record for this flow ID in the ReconTable. If this flow ID is not in the ReconTable, it means that this flow is new to the UFMD module. So the UFMD module creates a table record for the new Flow ID in the ReconTable and inserts the extracted identification data fragment into the ReconTable. However, if this flow ID was already in the ReconTable, it means that this flow has been already seen by the UFMD modules, so the UFMD module only inserts the extracted identification data fragment into the ReconTable. The order of different fractions of the same source identification data is recognizable to the UFMD module by the fragment number field which has S bits (for K=5, S is 3 bits and for K=2, S is 1 bit) in the packet's IP header.

Once all the fractions of the same source identification data of a flow get to the UFMD module, the origin of all the packets in that flow, as well as the origin of all the packets in any other flow with the same flow ID is apparent to the victim, even if the source IP address of those packets are hidden in one shape or form as discussed earlier. Using UFMD, the destination is able to distinguish the traffic of different nodes behind an edge router. As a result, when an abnormal traffic is observed, the victim is able to distinguish between the attack and the legitimate traffic and infer the source of an attack, even if it is behind a NAT or a proxy device.

IV. Experimental Results of UFM

To evaluate our proposed UFM approach and compare it with the DFM [1]–[3] and DPM [9] approaches, we have employed three evaluation network traces. These are the CAIDA anonymized Internet traces March and July 2014 datasets [12], and the CAIDA DDoS attack 2007 data set [13]. We chose DPM to compare with our method because it is the only well-known approach that falls into the same categories of classification as UFM, Section III. We implemented a testbed network in our research lab to replay the aforementioned traces from a local area network and direct them to a destination over the Internet. For this purpose, we used tcpreplay and tcprewrite open source applications [14]. In addition, we implemented two real-time programs using the Winpcap library by C++ [15], one for implementing the UFME module to mark the packets, and the other for implementing the UFMD module to trace back the source of the packets. The UFME module runs at the egress interface of the source edge router and only marks outgoing packets. At the same time, the traceback program runs at the destination network and aims to trace the origin of the marked traffic.

TABLE I: Comparison between DPM [9], DFM and UFM Approaches in Terms of Traceback and Marking Rates

K	Data Set	Method	TR(Packets)	TR(Volume)	MR
Two	CAIDA March 2014	DPM	99.21	99.76	100
		DFM	94.57	97.22	6.90
		UFM	96.38	98.38	5.20
	CAIDA July 2014	DPM	99.14	99.69	100
		DFM	92.70	97.24	9.11
		UFM	95.98	98.60	6.44
	CAIDA DDoS 2007	DPM	99.99	99.99	100
		DFM	77.71	78.79	79.16
		UFM	96.80	97.41	6.00
Five	CAIDA March 2014	DPM	99.27	99.74	100
		DFM	93.48	96.72	8.10
		UFM	94.87	97.53	6.15
	CAIDA July 2014	DPM	99.38	99.65	100
		DFM	91.24	96.73	10.48
		UFM	93.64	97.71	8.12
	CAIDA DDoS 2007	DPM	99.99	99.99	100
		DFM	8.87	9.28	98.39
		UFM	93.82	95.00	6.68

A. Traceback and Marking Rates

For evaluating the UFM approach, we have used the following evaluation metrics:

- TR, Traceback Rate: The ratio of the number or the volume of successfully traced back packets to all packets.
- MR, Marking Rate: The ratio of the marked packets to all packets.

The desired outcome is to have higher values of TR and lower values of MR. As described before in section III-A, the best value of K (the best number of fragments for each flow identification data) is either 2 or 5. We have evaluated our UFM approach by both of these two best values of K. Table I presents the comparison between the UFM and the other approaches, using TR and MR metrics, on the CAIDA July 2014, March 2014 and DDoS 2007 data sets. As can be seen in this table, DPM has higher traceback rate compared to UFM and DFM, but this accuracy is achieved by marking all the packets in the network ($MR = 100\%$), which is not acceptable. In comparison to DFM, UFM increases the traceback rate and decreases the marking rate up to 3% for March and July data sets, but for the DDoS data set we achieve a huge improvement (for $K = 2$, UFM increase the TR by 19% and decrease the MR up to 73%. For $K = 5$, UFM increase the TR up to 85% and decrease the MR up to 91%).

It should be note that one of the advantages of the UFM is that there is a direct relation between the lower marking rate and the higher traceback rate. As can be seen, when we set $K = 2$, we get better results than when we set $K = 5$. However, as described in section III-A, the lower marking rate

needs more bits in the header of the packets to embed the marking. The required bits in the IP header for $K = 2$ is 32 bits. This is 2 times more than when $K = 5$ is set.

B. Memory Usage

The UFME and the UFMD modules have the following memory usage:

1) Memory Usage of the UFME Module at the Source-end: The memory required for running the UFME module on an edge router is equal to the sum of the required memory for three tables namely, the FragTable, the NI-ID table, and the Node-ID table. On our testbed where the aforementioned data sets were run, the total required memory for running the UFME module was about 35MB. Below we explain the details of each of these tables and the corresponding memory usage:

FragTable: As described before, the UFME module maintains a table to keep track of the marked packets, their flow IDs, and the embedded marks into them. Each row in this table stores a flow ID and K possible fragments of the source identification data. If K=2 is selected, then each row of the FragTable is 168 bits including 13 byte flow ID and 64 bits for two data fragments. If K=5 is selected, each row is 184 bits including 13 bytes flow ID and 80 bit for five data fragments. Since the UFME module marks only one flow among all the flows that share the same flow ID, it keeps the records of a flow for a specific period of time in the FragTable to be able to compare the flow IDs of the new incoming packets with the already seen flow IDs.

NI-ID table: For every interface on the edge router and for every VLAN (in case of the existence of VLANs on the edge router), 9 bytes including 12 bits for NI-ID, 48 bits for MAC address, and 12 bits for VLAN ID are stored.

Node-ID table: For every record in the NI-ID table, UFME module stores a separate Node-ID table. For every newly observed source MAC address, a 60-bit record including 12 bits NI-ID and 48 bits MAC address are stored. It should be note that the NI-ID and the Node-ID can be extracted from the ARP table of the edge routers.

2) Memory Usage of the UFMD Module at the Destination-end: As discussed in section III-B, the UFMD module maintains the ReconTable for matching the flow ID and K possible identification data fragments of a flow, in order to reconstruct a valid source identification data. Like the FragTable, for every record in the ReconTable, a flow ID and K possible identification data fragments should be stored. Therefore, the space required for the ReconTable is the same as the FragTable which is 35MB.

V. Conclusion

In this paper, we presented a new IP traceback approach, called Unique Flow Marking, UFM. The main concept behind the UFM is to find and mark only some packets of a flow among all of the flows that have the same flow ID. Once the source of some packets in a flow can be found, the source of any other packets in that flow, as well as the source of any other packets in any other flows with the same flow ID can also be found. UFM utilizes this fact to decrease the marking rate and to increase the trace back rate (compared to other approaches) simultaneously. UFM consists of two modules, the UFM encoding module, UFME, and the UFM decoding module, UFMD. UFME runs at the egress interface of the edge router at the source network and aims to mark some packets of one flow out of all the flows that have the same flow ID. UFMD runs at the destination network and aims to infer the source of traffic by analyzing the marked packets and extracting the source identification data from them. Our results show that the UFM demonstrates better performance in terms of traceback and marking rates in comparison to the DFM. For future work, we will improve our new UFM approach to operate at the conversation level, instead of the flow level, to achieve higher traceback and lower marking rates. In addition, we will perform a sensitivity analysis on the traceback and marking rates with the number and the location (on the network) of the participating routers.

Acknowledgment

This research is supported by the Natural Science and Engineering Research Council of Canada (NSERC) grant, and is conducted as part of the Dalhousie NIMS Lab at: https://projects.cs.dal.ca/projectx/

References

[1] V. Aghaei-Foroushani and A.N. Zincir-Heywood. Ip traceback through (authenticated) deterministic flow marking: an empirical evaluation. *EURASIP Journal on Information Security*, 5:., 2013.

[2] V. Aghaei-Foroushani and A.N. Zincir-Heywood. Deterministic and authenticated flow marking for ip traceback. *The 27th IEEE International Conference on Advanced Information Networking and Applications (AINA), Barcelona*, 5:25–28, March 2013.

[3] V. Aghaei-Foroushani and A.N. Zincir-Heywood. On evaluating ip traceback schemes: a practical perspective. *IEEE International Workshop on Cyber Crime (IWCC 2013), San Francisco*, pages 127–134, May 2013.

[4] V. Aghaei-Foroushani and A.N. Zincir-Heywood. Tdfa: Traceback-based defense against ddos flooding attacks. *The 28th IEEE International Conference on Advanced Information Networking and Applications (AINA), Victoria, Canada*, page ., May 2014.

[5] T. Subbulakshmi, I. A. A. Guru, and S. M. Shalinie. Attack source identification at router level in real time using marking algorithm deployed in programmable routers. *ICRTIT*, pages 79–84, 2011.

[6] AC Snoeren, C Partridge, LA Sanchez, CE Jones, F Tchakountio, B Schwartz, ST Kent, and WT Strayer. Single-packet ip traceback. *IEEE/ACM Transactions on Networking, December*, 10(6):721–734, December 2002.

[7] A. Belenky and N. Ansari. On ip traceback. *IEEE Communications Magazine, July*, 41(7):142–153, Jul 2003.

[8] S Savage, A Karlin, D Wetherall, and T Anderson. Network support for ip traceback. *IEEE/ACM Transactions on Networking, June*, 9(3):226–237, Jun 2001.

[9] A. Belenky and N. Ansari. On deterministic packet marking. *The International Journal of Computer and Telecommunications Networking, July*, 51(10):2677–2700, Jul 2007.

[10] M. Yang. Riht: A novel hybrid ip traceback scheme. *IEEE Transactions on Information Forensics and Security, April*, 7(2):789–797, Apr 2012.

[11] A. Yaar, A. Perrig, and D. Song. Fit: fast internet traceback. *24th Annual Joint Conference of the IEEE Computer and Communications INFOCOM, March*, 2:1395–1406, Mar 2005.

[12] The caida anonymized internet traces 2014 dataset, accessed february 13, 2015.

[13] The caida "ddos attack 2007" dataset, accessed february 13, 2015.

[14] A. turner, tcpreplay suite, accessed february 13, 2015.

[15] Riverbed technology, winpcap, the industry-standard windows packet capture library, accessed february 13, 2015.