Managing the Denial-of-Service Attacks Using Rectified PPM Algorithm

V. Midhun [#1], T. Nalini[*2], M. Padmavathy[#3]

[#] *Dept of Computer Science and Engineering, Bharath University Chennai India*

[2]Professor,Bharath University
[1,3]Student, Bharath University,
s.niharranjan8899@gmail.com
[3]krishnavathy@gmail.com

Abstract— **The denial-of-service (DoS) attack has been a pressing problem in recent years. DoS defense research has blossomed into one of the main streams in network security. It is a malicious attempt by a single person or a group of people to cause the victim, site, or node to deny service to its customers. When this attempt derives from a single host of the network, it constitutes a DoS attack. On the other hand, it is also possible that a lot of malicious hosts coordinate to flood the victim with an abundance of attack packets, so that the attack takes place simultaneously from multiple points. This type of attack is called a Distributed DoS, or DDoS attack. Whenever this type of attack occurs, the Probabilistic Packet Marking (PPM) algorithm is used to discover the internet map or an attack graph that contains the traverse details of the packets. The PPM algorithm does not perform well, when the termination condition is not well defined in the literature. Also without a proper termination condition, the attack graph constructed by the PPM algorithm would be wrong. So by providing a precise termination condition for the PPM algorithm, the new algorithm is named as the Rectified PPM (RPPM) algorithm. The significant feature of the RPPM algorithm is that when the algorithm terminates, it guarantees that the constructed attack graph is correct, with a specified level of confidence. It also provides an autonomous way for the original PPM algorithm to determine its termination and it is a promising means of enhancing the reliability.**

Keywords— **Network-level security and protection, probabilistic computation, DoS attack, PPM.**

I. INTRODUCTION

The Internet consists of hundreds of millions of computers distributed all around the world. Millions of people use the Internet daily, taking full advantage of the available services at both personal and professional levels. The interconnectivity among computers on which the World Wide Web relies, however, renders its nodes an easy target for malicious users who attempt to exhaust their resources and launch Denial-of-Service (DoS) attacks against them. DoS attacks attempt to exhaust the victim's resources. These resources can be network bandwidth, computing power, or operating system data structures. To launch a distributed DoS attack, malicious users first build a network of computers that they will use to produce the volume of traffic needed to deny services to computer users. To create this attack network, attackers discover vulnerable sites or hosts on the network. Vulnerable hosts are usually those that are either running no antivirus software or out-of-date antivirus software, or those that have not been properly patched. Vulnerable hosts are then exploited by attackers who use their vulnerability to gain access to these hosts. The next step for the intruder is to install new programs (known as attack tools) on the compromised hosts of the attack network. The hosts that are running these attack tools are known as zombies, and they can carry out any attack under the control of the attacker. Many zombies together form what we call an army.

The structure of this paper is organized as follows. In Section II we present the theoretical background of our work. Section III describes the proposed system, Rectified PPM algorithm. Section IV describes the experiments and results. We conclude in Section V.

II. RELATED WORK

The probabilistic packet marking (PPM) algorithm by Savage et al. [8] has attracted the most attention in contributing the idea of IP traceback [9], [10], [11]. The most interesting point of this IP traceback approach is that it allows routers to encode certain information on the attack packets based on a predetermined probability. Upon receiving a sufficient number of marked packets, the victim (or a data collection node) can construct the set of paths that the attack packets traversed and, hence, the victim can obtain the location(s) of the attacker(s).

A. The Probabilistic Packet Marking Algorithm

The goal of the PPM algorithm is to obtain a constructed graph such that the constructed graph is the same as the attack graph, where an attack graph is the set of paths the attack packets traversed, and a constructed graph is a graph returned by the PPM algorithm. To fulfill this goal, Savage Suggested a method for encoding the information of the edges of the attack graph into the attack packets through the cooperation of the routers in the attack graph and the victim site. Specifically, the PPM algorithm is made up of two separated procedures: the packet marking procedure, which is executed on the router side, and the graph reconstruction procedure, which is executed on the victim side. The packet marking procedure is designed to randomly encode edges' information on the

packets arriving at the routers. Then, by using the information, the victim executes the graph reconstruction procedure to construct the attack graph. Let us review the packet marking procedure in the section B.

B. A Brief Review of the Packet Marking Procedure

The packet marking procedure aims at encoding every edge of the attack graph, and the routers encode the information in three marking fields of an attack packet: the start, the end, and design of the marking fields). In the following, we describe how a packet stores the information about an edge in the attack graph, and the pseudocode of the procedure in [8] is given in Fig. 1 for reference

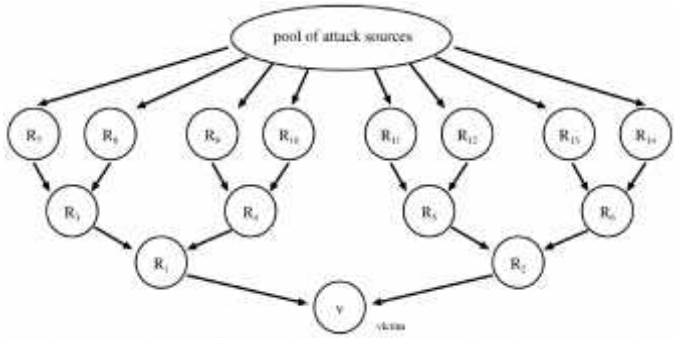

Fig. 1 The pseudocode of the packet marking procedure of the PPM algorithm.

When a packet arrives at a router, the router determines how the packet can be processed based on a random number x (line number 1 in the pseudocode). If x is smaller than the predefined marking probability pm, the router chooses to start encoding an edge. The router sets the start field of the incoming packet to the router's address and resets the distance field of that packet to zero. Then, the router forwards the packet to the next router. When the packet arrives at the next router, the router again chooses if it should start encoding another edge. For example, for this time, the router chooses not to start encoding a new edge. Then, the router will discover that the previous router has started marking an edge, because the distance field of the packet is zero. Eventually, the router sets the end field of the packet to the router's address. Nevertheless, the router increments the distance field of the packet by one so as to indicate the end of the encoding. Now, the start and the end fields together encode an edge of the attack graph. For this encoded edge to be received by the victim, successive routers should choose not to start encoding an edge, that is, the case x > pm in the pseudocode, because a packet can encode only one edge. Furthermore, every successive router will increment the distance field by one so that the victim will know the distance of the encoded edge.

In the existing system, PPM algorithm is not perfect, as its termination condition is not well defined. The algorithm requires prior knowledge about the network topology. In packet marking algorithm the Termination Packet Number (TPN) calculation is not well defined in the literature. At present, it only supports the single attacker environment. Without proper termination condition the attack graph constructed by the PPM algorithm would be wrong. The constructed path and the re-construction will be differed. It won't support the multiple attacker environments.

III. PROSPOSED SYSTEM

To propose termination condition of the PPM algorithm, this is missing or is not explicitly defined in the literature. Through the new termination condition, the user of the new algorithm is free to determine the correctness of the constructed graph. The constructed graph is guaranteed to reach the correctness assigned by the user, independent of the marking probability and the structure of the underlying network graph. In this system we proposed a Rectified Probabilistic Packet Marking Algorithm to encode the packet in the routers to detect the attacked packets. To reduce the a constructed graph such that the constructed graph is the same as the attack graph, where an attack graph is the set of paths the attack packets traversed, To construct a graph, is a graph returned by the PPM algorithm.

In this paper, we modify the PPM algorithm so that the victim can obtain a correct constructed graph with a specified level of guarantee. The contributions of this work are listed as follows:

- We introduce the termination condition of the PPM algorithm, which is missing or is not explicitly defined in the literature.
- Through the new termination condition, the user of the new algorithm is free to determine the correctness of the constructed graph.
- The constructed graph is guaranteed to reach the correctness assigned by the user, independent of the marking probability and the structure of the underlying network graph.

A. Rectified Probabilistic Packet Marking Algorithm

The RPPM algorithm is designed to automatically determine when the algorithm should terminate. We aim at achieving the following properties:

1. The algorithm does not require any prior knowledge about the network topology.
2. The algorithm determines the certainty that the constructed graph is the attack graph when the algorithm terminates.

Our goal is to devise an algorithm that guarantees that the constructed graph is the same as the attack graph with probability greater than P^*, where we name P^* the traceback confidence level (it is analogous to the level of confidence that the algorithm wants to achieve). To accomplish this goal, the graph reconstruction procedure of the original PPM algorithm is completely replaced, and we name the new procedure the rectified graph reconstruction procedure. On the other hand, we preserve the packet marking procedure so that every router deployed with the PPM algorithm is not required to change. In the following section, we list the assumptions of our solution. Then, we describe the flow of the rectified graph reconstruction procedure.

B. Flow of the Rectified Graph Reconstruction Procedure

The pseudocode of the rectified graph reconstruction procedure is shown in Fig. 2, and the procedure is started as soon as the victim starts collecting marked packets. When a marked packet arrives at the victim, the procedure first checks if this packet encodes a new edge. If so, the procedure accordingly updates the constructed graph Gc. Next, if the constructed graph is connected, where connected means that

Rectified Graph Reconstruction Procedure (Traceback Confidence Level P^*)

```
/* Initially, G_c contains the "victim" node only, and pkt_count = 0. */
1. Foreach incoming packet pkt ; do
2.   pkt_count := pkt_count + 1;
3.   If the incoming packet pkt contains an edge e that is not included in G_c; then
4.         Construct the new attack graph G_c by inserting the edge e ;
5.         If G_c is a connected graph ; then
6.             TPN := TPN_subroutine(G_c, P*) ;
7.             pkt_count := 0 ;
8.         end If
9.   end If
10.  If G_c is a connected graph ; then
11.        If pkt_count > TPN ; then
12.            Return G_c as the constructed graph ;
13.        end If
14.  end If
15. end Foreach
```

Fig. 2 The pseudocode of the rectified graph reconstruction procedure of the RPPM algorithm.

every router can reach the victim, the procedure calculates the number of incoming packets required before the algorithm stops, and we name this number the TPN. The procedure then resets the counter for the incoming packets to zero and starts counting the number of incoming packets. In the meantime, the procedure checks if the number of collected packets is larger than the TPN. If so, the procedure claims that the constructed graph Gc is the attack graph, with probability P^*. Otherwise, the victim receives a packet that encodes a new edge. Then, the procedure updates the constructed graph, revisits the TPN calculation subroutine, resets the counter for incoming packets, and waits until a packet that encodes a new edge arrives or the number of incoming packets is larger than the new TPN.

IV. EXPERIMENTS AND RESULTS

Savage et al. [8] suggest probabilistically marking packets as they traverse routers in the Internet. More specifically, they propose that router mark the packet, with low probability (say, 1/20,000), with either the router's IP address or the edges of the path that the packet traversed to reach the router.

For the first alternative, analysis shows that in order to gain the correct attack path with 95% accuracy as many as 294,000 packets are required. The second approach, edge marking, requires that the two nodes that make up an edge mark the path with their IP addresses along with the distance between them (the latter requires 8 bits to represent the maximum hop count allowable in IP). This approach would require more state information in each packet than simple node marking but would converge much faster. They suggest 3 ways to reduce the state information of these approaches into something more manageable.

The first approach is to XOR each node forming an edge in the path with each other. Node a inserts its IP address into the packet and sends it to b. Upon being detected at b (By detecting a 0 in the distance), b XORs it's address with the address of a. This new data entity is called an edge id and reduces the required state for edge sampling by half. Their next approach is to further take this edge id and fragment it into k smaller fragments. Then, randomly select a fragment and encode it, along with the fragment offset so that the correct corresponding fragment is selected from a downstream router for processing.

When enough packets are received, the victim will receive all edges and all fragments so that an attack path can be reconstructed (even in the presence of multiple attackers). The low probability of marking reduces the associated overheads. Moreover, only a fixed space is needed in each packet. The approach results in a large number of false positives. As an example, with only 25 attacking hosts in a DDoS attack the reconstruction process takes days to build and results in thousands of false positives.

Accordingly, Song and Perrig propose the following traceback scheme: instead of encoding the IP address interleaved with a hash, they suggest encoding the IP address into an 11 bit hash and maintain a 5 bit hop count, both stored in the 16-bit fragment ID field. This is based on the observation that a 5-bit hop count (32 max hops) is sufficient for almost all Internet routes. Further, they suggest that two different hashing functions be used so that the order of the routers in the markings can be determined. Next, if any given hop decides to mark it first checks the distance field for a 0, which implies that a previous router has already marked it. If this is the case, it generates an 11-bit hash of it's own IP address and then XORs it with the previous hop. If it finds a non-zero hop count it inserts it's IP-hash, sets the hop count to zero and forwards the packet onThese work can be classified in to various modules and it can be viewed in following figure.

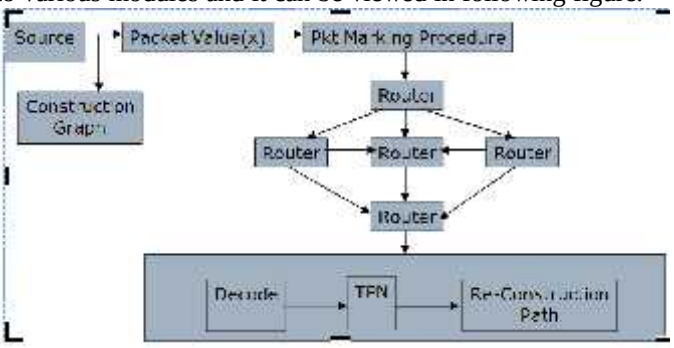

Fig. 3 Overall workflow Diagram

1) Path Construction

The path will be constructed in which the data packets should traverse. This path should be dynamically changed in case of traffic and failure in router.

2) Packet Marking Procedure

Each packet will be marked with random values. These values are encoded and its appended in the start or in the edge of the packets. Also it is checked by the packet marking procedure.

3) Router Maintenance

The router availability will be checked depends upon the router availability the path will be constructed.

4) TPN Generation

The encoded values in the packet are retrieved and it's decoded and checked with the generated code.

5) Re-Construction Path

The path will be re-constructed with the received packets it's validated with the constructed path.

V. CONCLUSION

In this paper, we addressed the problem of securing a communication service on top of the existing IP infrastructure from DoS attacks.To solve this we have the new traceback approach called the RPPM algorithm. The RPPM algorithm does not require any previous knowledge about the network graph. Also, it guarantees that the constructed graph is a correct one, with a specified probability, and such a probability is an input parameter of the algorithm. To conclude, the RPPM algorithm is an effective means of improving the reliability of the original PPM algorithm. The most fundamental lesson to be learned from distributed denial of service is the fact that all sites on the Internet are interdependent, whether they know it or not. Intruders have automated the processes for discovering vulnerable sites, compromising them, installing daemons, and concealing the intrusion. There is some hope for the future in technological and other approaches.

REFERENCES

[1] Tsz-Yeung Wong, Man-Hon Wong, and Chi-Shing (John) Lui, "A Precise Termination Condition of the Probabilistic Packet Marking Algorithm," *IEEE* Transactions on dependable and secure computing, vol. 5, no. 1, January-March 2008.

[2] D.K.Y. Yau, J.C.S. Lui, F. Liang, and Y. Yam, "Defending against Distributed Denial-of-Service Attacks with Max-Min Fair Server-Centric Router Throttles," IEEE/ACM Trans. Networking, no. 1,pp. 29-42, 2005.

[3] The Internet Protocol Journal, Cisco Systems.

[4] "CERT Advisory CA-2000-01: Denial-of-Service Developments," Computer Emergency Response Team, http://www.cert.org/-advisories/-CA-2000-01.html, 2006.

[5] "CAIDA Router-Level Topology Measurements," Cooperative Assoc. Internet Data Analysis,http://-www.caida.org/tools/measurement/skitter/router_topology/, 2006.

[6] W. Hoeffding, "Asymptotically optimal tests for multinomial distributions," *Ann. Math. Statist.*, vol. 36, pp. 369–401, 1965.

[7] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris. Resilient Overlay Networks. In *Proceedings of the 18th Symposium on Operating Systems Principles (SOSP)*, October 2001.

[8] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Practical Network Support for IP Traceback," Proc. ACM SIGCOMM '00, pp. 295-306, 2000.

[9] D. Dean, M. Franklin, and A. Stubblefield, "An Algebraic Approach to IP Traceback," ACM Trans. Information and System Security, vol. 5, no. 2, pp. 119-137, 2002.

[10] D.X. Song and A. Perrig, "Advanced and Authenticated Marking Schemes for IP Traceback," Proc. IEEE INFOCOM '01, pp. 878-886,Apr. 2001.

[11] A.C. Snoeren, C. Partridge, L.A. Sanchez, C.E. Jones, F. Tchakountio,S.T. Kent, and W.T. Strayer, "Hash-Based IP Traceback,"Proc. ACM SIGCOMM '01, pp. 3-14, Aug. 2001.

[12] A.C. Snoeren, C. Partridge, L.A. Sanchez, C.E. Jones, F. Tchakountio,S.T. Kent, and W.T. Strayer, "Hash-Based IP Traceback,"Proc. ACM SIGCOMM '01, pp. 3-14, Aug. 2001.

[13] K. Park and H. Lee, "On the Effectiveness of Probabilistic Packet Marking for IP Traceback under Denial-of-Service Attacks," Proc. IEEE INFOCOM '01, pp. 338-347, 2001.