A novel approach for IP trace back evaluation with specified probability in Network Model

S Babu[1], A Raghuvira Pratap[2], J V D Prasad [3]
[1,2,3]V.R.Siddhartha Engineering College , India

***Abstract*—**the issue of recognizing DDoS (Distributed Denial of Service) Attack is one of the common dangers in the field of Internet security. The probabilistic parcel stamping (PPM) calculation is a promising method to find the Internet outline an assault chart that the assault bundles crossed amid a dispersed disavowal of-benefit assault. In any case, the PPM calculation isn't flawless, as its end condition isn't very much characterized in the writing. All the more vitally, without an appropriate end condition, the assault chart developed by the PPM calculation would not be right. In this work, we give an exact end condition to the PPM calculation and name the new calculation the amended PPM (RPPM) calculation. The most noteworthy value of the RPPM calculation is that when the calculation ends, the calculation ensures that the developed assault chart is right, with a predetermined level of certainty. We complete reproductions on the RPPM calculation and demonstrate that the RPPM calculation can ensure the rightness of the built assault diagram under 1) distinct probabilities that a switch denotes the assault parcels and 2) unique structures of the system chart. The RPPM calculation gives a self-sufficient route to the first PPM calculation to decide its end, and it is a promising method for improving the dependability of the PPM calculation.

***Keywords*—** *PPM and RPPM algorithm, denial of service, internet control message protocol, routing function*

I. INTRODUCTION

The denial-of-service (DoS) attack has been a pressing Problem over the last few years DoS defense research has blossomed into one of the main streams in network security. Different ways of doing things such as the pushback message, ICMP trace back, and the packet filtering ways of doing things are the results from this active field of research. The (related to the study of how likely or unlikely things are to happen) packet marking (PPM) set of computer instructions has attracted the most attention in (related to one thing being directly responsible for another) the idea of IP trace back.

In this paper, we study how an assailant could proficiently spoof the packets in order to deceive the victim and consequently conceal his identity, even without the learning of system topology. We at first spotlight on a straightforward technique, in which the assailant sets the packet stamping field haphazardly, yet have the capacity to accomplish his objective. We additionally demonstrate that by using the way length circulation, the aggressor can accomplish higher namelessness. [4] We additionally contemplate the remaking procedure and the quantity of packets expected to remake in reasonable situations. We appear that choosing how to gather the assault packets and to what extent the recreation process ought to be executed is a non-insignificant challenge that has not been beforehand tended to.

The most interesting point of this IP trace back 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).

1.1 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 et al. [6] 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.

1.1.1 A Brief Review of the Packet Marking Procedure

The packet marking procedure aims at every edge of the attack graph, and the routers the information in three marking fields of an attack packet: the start, the end, and the distance fields Animal-like et al. [7] has discussed the 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 pseudo code of the procedure in [7] is given in Fig. 1 for reference. When a packet arrives at a router, the router decides/figures out how the packet can be processed based on a random number x. If x is smaller than the predefined marking chance pm, the router chooses to start 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,[1] 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 another edge. For example, for this time, the router chooses not to start 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.

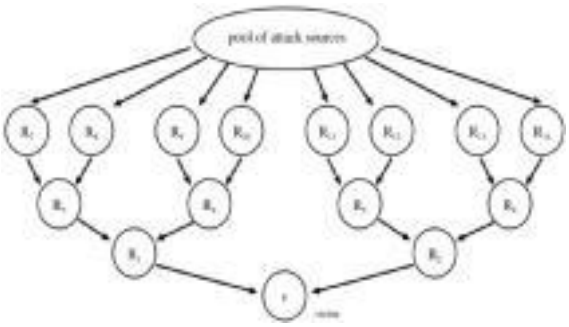

Fig.1. the pseudo code of the packet marking procedure of the PPM algorithm.

Anyway, the router small steps forward the distance field of the packet by one to point the end. Now, the start and the end field together an edge of the attack graph. For this edge to be received by the victim, routers should choose not to start an edge, that is, the case x > pm in the Pseudo code, because a packet can only one edge. What's more, every router will be the distance field by one so that the victim will know the distance of the edge.

1.2 Termination of the PPM Algorithm

According to the above description of the packet marking procedure, although a packet has already encoded an edge, successive routers may choose to start encoding another edge randomly. As a result, when a packet arrives at the victim, the packet may encode any of the edges of the attack graph, or a packet may not encode any edges. Therefore, if the victim can collect a sufficiently large number of marked packets, the victim can successfully construct all the paths in the attack graph by using the graph reconstruction procedure. When the graph reconstruction procedure returns a constructed graph, it implies the termination of the PPM algorithm.

However, the termination condition has not thoroughly been investigated in the literature. It turns out that the termination condition is important, because it determines the correctness of the constructed graph: If it stops too early, the constructed graph will not contain enough edges of the attack graph and, thus, fails to fulfill the trace back purpose. In addition, it is also not a proper way to allow the victim to collect marked packets for a long period before the victim starts the graph reconstruction procedure, because the victim would never know how much time is long enough. Hence, a proper termination condition can also help in speeding up the trace back process. In [8], Savage et al. have provided an estimation of the number of marked packets required before the victim can have a constructed graph that is the same as the attack graph under a single-attacker environment. Let X be the number of marked packets required for the victim to reconstruct a path.

II. RELATED WORK

2. 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 trace back 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. [10] 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. For example, in Fig. 6, the failure of the router R1 forces the routing table to completely change. Under such a scenario, the constructed attack graph may become the one shown in Fig. 6c. We argue that this result is not an undesirable one, as long as the definition of a correct attack graph construction still holds (because the new attack graph is indeed composed of all the edges traversed by the packets). In the remainder of this paper, we stay with this assumption, and we will discuss the scenario when this assumption is relaxed in Section 6.

2.1 Assumptions about the Victim

On the victim side, we assume that by the time that the victim starts collecting marked packets, all routers in the network have already invoked the packet marking procedure. In addition, we assume that the victim does not have any knowledge about the real network or the attack graph. However, the victim knows the marking probability that the routers are using.

2.2 Flow of the Rectified Graph Reconstruction Procedure

[9] The pseudo code of the fixed graph reconstruction procedure is shown in Fig.1, 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 a new edge. An execution diagram of the fixed graph reconstruction procedure of the RPPM set of computer instructions that constructs a graph with n edges procedure in the same way/in that way updates the built graph Gc. Next, if the built graph is connected, where connected means that every router can reach the victim, the procedure calculates the number of incoming packets needed/demanded before the set of computer instructions 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. [14] If so, the procedure claims that the built graph Gc is the attack graph, with chance P_. Otherwise, the victim receives a packet that a new edge. Then, the procedure updates the built graph, revisits the TPN calculation subroutine, resets the counter for incoming packets, and waits until a packet that a new edge arrives or the number of incoming packets is larger than the new TPN. As suggested by the pseudo code, the end/ending/firing condition of the RPPM set of computer instructions is that "the counter for the incoming packets is larger than the TPN," and this hints that the calculation of the TPN during each update of the built graph is the core of the RPPM set of computer instructions. In the next step, we provide a deeper understanding of the RPPM set of computer instructions through the introduction of the execution diagram.

2.3 Execution Diagram of the Rectified Probabilistic Packet Marking Algorithm

According to the previous section, it is observed that the TPN, the constructed graph, and the execution of the rectified graph reconstruction procedure are closely related. Such a relationship can be visualized by the construction of the execution diagram, as shown in Fig. 1. The execution diagram presents the dynamics of the execution of the rectified graph reconstruction procedure.

2.3.1 Types of States

There are two sorts of states in the graph: the execution state and the end state. At the point when the strategy is running, we say that "the redressed diagram remaking method is in an execution state." Otherwise, we say that "the corrected chart recreation system is in the end express." The execution state additionally discloses to us the condition of the built diagram:

1) When the strategy is in the begin state, named by "0," it implies that the strategy has begun running, and there are no edges in the developed chart.

2) When the methodology is in an associated state, it implies that the developed chart is associated. An associated state, marked by Ci, implies that the developed chart is associated and contains I edges.

3) When the strategy is in a detached express, the developed diagram is disengaged. A disengaged state, named by Di, implies that the developed diagram is detached and contains I edges. Note that both the associated and disengaged states, say, Ci and Di, separately, allude to all the conceivable diagrams that have I edges. Last, when the method is in the end state, it implies that the methodology has ceased.

2.3.2 Types of Transitions

There are two sorts of changes in the execution outline. At the point when the strategy takes a development progress, it implies that another edge is added to the built chart. At the point when the system takes an end change, it implies that the strategy will quit running. The progress structure in Fig. 1 is gotten from the pseudo code of the redressed chart reproduction system in Fig. 1. We quickly depict the progress structure as takes after: 1) If a parcel that encodes another edge lands before the quantity of got bundles is bigger than the TPN, at that point the methodology takes a development change and continues to either an associated state or a separated state, contingent upon the availability of the refreshed built chart. 2) If the quantity of got parcels is bigger than the TPN, at that point the methodology takes the end progress and continues to the end state. 3) If the system is in one of the disengaged states, at that point it is aimless to return such a diagram as the right built chart, and there is no change that associates the detached states to the end state. The system at that point keeps on gathering bundles until the point that it continues to an associated state.

2.4 Role of the Execution Diagram

The execution outline gives an exhaustive comprehension of the relationship among the execution of the corrected diagram recreation technique, the developed chart, and the TPN. Through the investigation of the execution outline, it can be watched that diverse execution situations of the method would influence the likelihood that the methodology restores a right developed diagram. It is watched that the direst outcome imaginable would be the hardest case for the redressed chart remaking methodology to restore a right diagram. Consequently, it is a perfect point for us to infer the figuring of the TPN. Assuming that one could effectively give a certification of the rightness of the developed chart under the direst outcome imaginable, at that point such an assurance can likewise be given in the normal case situation. Additionally, it is normal that the normal case situation ought to beat the most dire outcome imaginable regarding the effective rate of restoring a right developed chart. Next, we will proceed onward to the displaying of the parcel stamping procedure of the bundle checking technique.

III. METHODOLOGY

3. PACKET-TYPES PROBABILITY

The packet marking procedure is the source of different kinds of marked packets, and the total number of possible marked packets is the number of edges of the attack graph. However, it will be shown in the next section that the chance for every kind of marked packets that arrive at the victim plays a very important part in the derivation of the end/ending/firing packet number. In this section, we present the definition and the derivation of such a set of chances, and we name them the packet-type chances.

3.1 Encoded Edge Random Variable

By definition, an incoming packet may encode one of the edges of the attack graph, or the incoming packet does not encode any edges of the attack graph. We use a random variable called the encoded edge random variable to represent all possible encodings on an incoming packet. We formally define the encoded edge random variable as follows:

3.2 Procedure aims at encoding every edge

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 the distance fields (wherein Savage et al. [8] has discussed the 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 pseudo code of the procedure in [8] is given in Fig. 1 for reference.

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 pseudo code). 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 pseudo code, because a packet can encode only one edge. Furthermore, every successive router will increment the 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 pseudo code, because a packet can encode only one edge. Furthermore, every successive router will increment the variable.

The PPM algorithm is made up of two separated procedures:

The packet marking procedure: Which is executed on the router side?

The graph reconstruction procedure: Which is executed on the victim side?

3.3 Rectified problem the user use RPPM algorithm

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.

IV. PROPOSED APPROACH

4.1 DEPLOYMENT ISSUES OF THE RECTIFIED PROBABILISTIC PACKET MARKING ALGORITHM

1. We introduce the termination condition of the PPM algorithm, which is missing or is not explicitly defined in the literature.

2. Through the new termination condition, the user of the new algorithm is free to determine the correctness of the constructed graph.

3. The user to construct overall structure of the network graph.

In this section, we discuss several issues in deploying the RPPM algorithm. We first discuss the choice in the marking probability. Then, we cover the trade-off of the RPPM algorithm over the PPM algorithm. Last, we address the scalability problem in the PPM and the RPPM algorithms.

Choice of the Marking Probability

It is not desirable to have a high value of the marking probability. First, a high value of the marking probability means a low value for the packet-type probabilities for the majority of the types of packets. Hence, this implies that a large number of marked packets are needed before the RPPM algorithm stops. This also implies a long execution time of the RPPM algorithm. Let us take a linear network with three routers and one victim (as shown in Fig. 1) as an example to illustrate the relationship between the marking probability and the number of packets required. Fig. 2 shows the result of a simulation that aims at counting the average number of marked packets required for a correct graph reconstruction with different values of the marking probability. The result shows that for small values of marking probability, the number of required packets is small. Nevertheless, the number of required packets dramatically increases for large values of the marking probability. Despite the above reason, according to Section 5, a high value of the marking probability implies the presence of the worst-case scenario of the RPPM algorithm. Although the worst-case scenario can still guarantee the successful rate, it would be more beneficial to set the value of the marking probability to a

lower value so as to gain a larger successful rate than what is expected.

In conclusion, one should choose a small value for the marking probability for a faster and more reliable graph reconstruction. Note that there would be a large number of unmarked packets if one chooses a too-small value of the marking probability.

4.2 Execution Time Comparison between the PPM and the RPPM Algorithms

In order to guarantee the correctness of the constructed graph, the RPPM algorithm has to collect extra packets so as to attain such a guarantee. Technically speaking, before the moment that the constructed graph becomes the same as the attack graph, the number of marked packets collected should be the same for both the PPM and RPPM algorithms.

After the constructed graph has become the attack graph, the RPPM algorithm has to wait until the number of collected packets is larger than the TPN. In other words, that extra sum of packets is the trade-off in deploying the RPPM algorithm than the PPM algorithm. However, it is difficult to determine a theoretical value or bound of the TPN, because the TPN calculation depends on the construction process of the constructed graph.

The construction process, in turn, depends on the sequence of the arrivals of the marked packets, which is randomized. Alternatively, we conduct an empirical study on the trade-off of the RPPM algorithm. we present the number of increased marked packets when one compares the number of packets collected by the RPPM algorithm to those collected by the PPM algorithm (which is instructed to stop when the constructed graph becomes the attack graph). Such a set of simulations is performed using a marking probability of 0.1 (as suggested in Section 7.1) with increasing network scales: from a 15-node Random-tree network to a 1,000-node one. The RPPM Algorithm is operated under the average-case scenario.

Three main observations can be concluded from this set of Simulations. First, when the trace back confidence level Increases, the trade-off of the RPPM algorithm increases .Second, the number of collected packets by the RPPM algorithm is larger than those collected by the PPM algorithm by several times for the small range of the trace back Confidence level (two to five times for the trace back The Average Number of Packets and the Time Required to Reconstruct a Correct Constructed Graph in a 100BaseT Ethernet10 times for high values of the trace back confidence level. Last, an interesting observation is that the trade-offs for small networks are more significant than those for large networks. This can be explained by the probability of forming a disconnected graph. For a large network, such a probability is much higher than that of a small network. When a disconnected graph is formed, the TPN calculation is skipped until the graph becomes connected. Hence, this keeps the value of the TPN small during the ending states of the RPPM algorithm.

On the other hand, according to Table 1, one can observe that the time for the PPM algorithm to collect enough packets is in the order of a few seconds in a 100BaseT Ethernet.1 Therefore, although the trade-off of the RPPM algorithm could reach a multiple of 10, such a trade-off is acceptable.

4.3 Scalability

Scalability is one of the weaknesses of the PPM algorithm. One can observe that as the path length between the victim and the leaf router becomes longer, it becomes more difficult to collect a complete set of the marked packets. The case is that not only the path length affects the trace back time but the size of the attack graph also matters. one can observe that the number of marked packets required to build the constructed graph increases with the size of the graph, and the trend does not subside. Therefore, the PPM algorithm itself has a scalability problem. Nonetheless, as the RPPM algorithm inherits the packet marking procedure from the PPM algorithm, the RPPM algorithm also has the scalability problem. As suggested in Section , for small networks, the trace back process takes only a few seconds to complete. However, for networks as large as the one in [19] (with nearly 200,000 routers and more than 600,000 directed links), the trace back process may take days to finish.

V. CONCLUSION

In this work, we have pinpointed that the PPM algorithm lacks a proper definition of the termination condition. Meanwhile, using the expected number of required marked packets E½X_ as the termination condition is not sufficient. The above two outstanding problems only lead to an undesirable outcome: there is no guarantee of the correctness of the constructed graph produced by the PPM algorithm. We have devised the rectified graph reconstruction procedure to solve the above two problems, and we name the new trace back approach the RPPM algorithm. The RPPM algorithm, on one hand, does not require any previous knowledge about the network graph. On the other hand, 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. We have carried out a series of simulations to show the correctness and the robustness of the RPPM algorithm. The simulation results show that the RPPM algorithm can always satisfy our claim that the constructed graph is correct with a given probability. In addition, the algorithms robust under different values of the marking probability and different structures of the attack graphs. To conclude, the RPPM algorithm is an effective means of improving the reliability of the original PPM algorithm. Since the RPPM algorithm is an extension of the PPM algorithm, the RPPM algorithm inherits defects of the PPM algorithm. Problems such as scalability and different attack patterns will be future research directions.

ACKNOWLEDGMENTS

The authors would like to thank the editor and supporting staff for coordinating the review process. They also thank the anonymous reviewers for their insightful comments and

constructive suggestions. The work of M.H. Wong was partially supported by the RGC Grant 4208/04E. The work of John Lui was supported in part by the RGC Grant2150347.

VI. REFERENCES

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

[2] J. Ioannidis and S.M. Bellovin, "Implementing Pushback: Router-Based Defense against DDoS Attacks," Proc. Network and Distributed System Security Symp., pp. 100-108, Feb.2002.

[3] S. Bellovin, M. Leech, and T. Taylor, ICMP Traceback Messages, Internet Draft Draft-Bellovin-Itrace-04.txt, Feb. 2003.

[4] K. Park and H. Lee, "On the Effectiveness of Route-Based Packet Filtering for Distributed DoS Attack Prevention in Power-Law Internets," Proc. ACM SIGCOMM '01, pp. 15-26, 2001.

[7] P. Ferguson and D. Senie, "RFC 2267: Network Ingress Filtering: Defeating Denial of Service Attacks Which Employ IP Source Address Spoofing," The Internet Soc., Jan. 1998.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.

[8] C.W. Tan, D.M. Chiu, J.C. Lui, and D.K.Y. Yau, "A Distributed Throttling Approach for Handling High-Bandwidth Aggregates,"IEEE Trans. Parallel and DistributSystems, vol. 18, no. 7, pp. 983-995, July 2007.

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

[10] 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] 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.

[13] K.T. Law, J.C.S. Lui, and D.K.Y. Yau, "You Can Run, But You Can't Hide: An Effective Methodology to Traceback DDoS Attackers," IEEE Trans. Parallel and Distributed Systems, vol. 15,no. 9, pp. 799-813, Sept. 2005.

[14] M. Adler, "Trade-Offs in Probabilistic PacketMarking for IP Traceback," J. ACM, vol. 52, pp. 217-244, Mar. 2005.

[15] H. von Schelling, "Coupon Collecting for Unequal Probabilities,"Am. Math. Monthly, vol. 61, pp. 306-311, 1954.