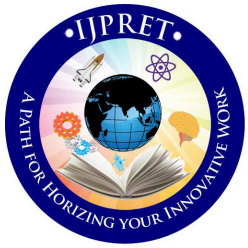
Research Article *ISSN: 2319-507X*
Gayatri Chavan, IJPRET, 2013; Volume 1(8): 832-841 *IJPRET*



INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

RECTIFIED PROBABILISTIC PACKET MARKING ALGORITHM TO OBTAIN A CORRECT CONSTRUCTED GRAPH

GAYATRI CHAVAN[1], PROF. P. P. TIJARE[2]

1. M.E. Student, Department of Information Technology, Sipana's Collage of Engineering & Technology.

2. Associate Professor, Department of Information Technology, Sipana's Collage of Engineering & Technology.

Accepted Date:

27/02/2013

Publish Date:

01/04/2013

Keywords

PPM

Algorithm

Corresponding Author

Ms. Gayatri Chavan

Abstract

A majority of attacks on computer systems result from a combination of vulnerabilities exploited by an intruder to break into the system. An Attack Graph is a general formalism used to model security vulnerabilities of a system and all possible sequences of exploits which an intruder can use to achieve a specific goal. Attack Graphs can be constructed automatically using off-the-shelf model-checking tools. We present two algorithms for generation of an Attack Graph based on the probability of an attacker reaching those states. The first algorithm is the probabilistic packet marking (PPM) algorithm is a promising way to discover the Internet map or an attack graph that the attack packets traversed during a distributed denial-of-service attack. However, the PPM algorithm is not perfect, as its termination condition is not well defined. More importantly, without a proper termination condition, the attack graph constructed by the PPM algorithm would be wrong. The second algorithm rectified PPM (RPPM) algorithm is that when the algorithm terminates, the algorithm guarantees that the constructed attack graph is correct, with a specified level of confidence. The experimental results of the RPPM algorithm to show that the RPPM algorithm can gives the correctness of the constructed attack graph under 1) different probabilities that a router marks the attack packets and 2) different structures of the network graph. The RPPM algorithm provides an autonomous way for the original PPM algorithm to determine its termination, and it will be a promising means of enhancing the reliability of the PPM algorithm.

1 Introduction

Distributed denial-of-service (DDoS) attacks have become a major threat to the internet. DoS defense research has blossomed into one of the main streams in network security. Various techniques such as the pushback message, ICMP trace back, and the packet filtering techniques are the results from this active field of research. The probabilistic packet marking (PPM) algorithm has attracted the most attention in contributing the idea of IP trace back. 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). 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. 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 Overview of PPM algorithm

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. 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 the distance fields.

Packet Marking Procedure(Packet w**)**

1. Let x be a random number in $[0 \ldots 1)$
2. **If** $x < p_m$, **then**
3. write router's address into $w.start$ and 0 into $w.distance$
4. **else**
5. **If** $w.distance = 0$ **then**
6. write router's address into $w.end$
7. **end If**
8. increment $w.distance$ by one
9. **end If**

Fig.1 shows the pseudo code of the marking procedure.

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 distance field by one so that the victim will know the distance of the encoded edge.

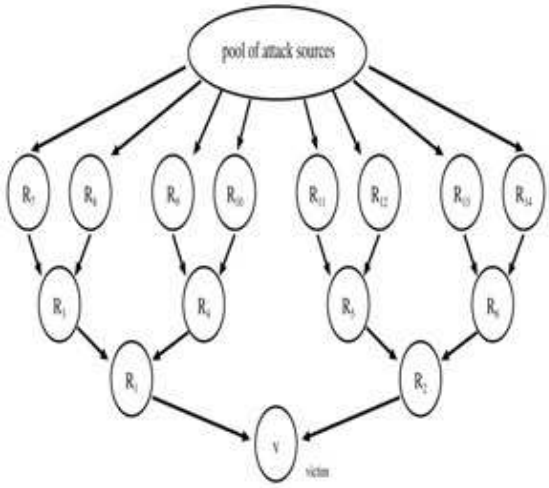


Fig.2 A 14-router binary-tree network

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. Also one cannot apply one cannot apply the termination condition to complex networks such that the reconstruction of one path is dependent on another. This scenario can be explained in Fig. 2, which is a binary-tree network with 14 routers. The leaf routers from R7 to R14 are connected to a pool of attackers. These attackers send out attack traffic toward the victim v, and this presents a multiple-attacker environment. In this graph, the attack packets traversed through eight paths that are identical in structure. However, there are "shared" edges among these paths. This implies that the reconstruction of one path is dependent on another. Therefore, one cannot treat (1) as the termination condition under this scenario and this restricts the application of the PPM algorithm. Hence, a proper termination condition can also help in speeding up the trace back process. Therefore, the new algorithm the rectified PPM (RPPM) algorithm is a way to obtain a correct constructed graph with a specified level of guarantee.

2 Literatures Review & Related Work

The denial-of-service (DoS) attack has been a pressing problem in recent years [2]. Pushback message [3], ICMP trace back [4], rate limiting, packet filtering [5], [7], [8], in some cases, help limit the impact of Denial-of-service attacks, but usually only at points where the Denial-of-service attack is consuming fewer resources than that are available. In many cases, the only defense is a reactive one, where the source or sources of an ongoing attack are identified and prevented from continuing the attack. One major difficulty is to defend against Distributed Denial-of-service attack is that attackers often use fake, or spoofed IP addresses as the IP source address. Therefore, attackers can easily disguise themselves as some other hosts on the Internet. Because of the stateless nature of the Internet, it is a difficult task to determine or trace the source of these

attacker's packets and there by locate the potential locations of these attackers. This is known as the IP trace back problem. Many IP trace back techniques [9], [11], [12], [13] have been proposed, they all have short comings that limit their usability in practice. Some of them are Ingress filtering[6] requires edge routers to have sufficient processing power, to inspect the packet's destination IP address for normal packet forwarding service. It also needs to inspect the source address and determine whether it is a legitimate or illegitimate address. Another major problem with ingress filtering is that this technique is only effective if there is a widespread deployment in the networking community such that many ISPs are willing to deploy this service. Moreover, even with the enabling of ingress filtering service, attackers can still forge the source IP addresses as other hosts within their network domain. Alternative approach to DDoS trace back includes input debugging approach, which requires cooperation between system administrators of different ISPs. Therefore, it may not be able to trace the attackers in real-time or in the midst of a DDoS attack. Other approaches such as controlled flooding, which either generates many additional packets to the network (which can be viewed as another form of DDoS attack), or network logging [12], which requires additional storage and computational overhead of the participating routers. All, the above approaches have performance problems and significant deployment difficulties. One promising solution, proposed by savage et al [10], is to let routers probabilistically mark packets with partial path information during packet forwarding. The victim then reconstructs the complete path after receiving a modest number of packets that contain the marking. This approach has a low overhead for routers and the network and supports incremental deployment.

3 Analysis of Problem

The PPM algorithm is not perfect, as its termination condition is not well defined. The algorithm requires prior knowledge about the network topology. In PPM algorithm the Termination Packet Number (TPN) calculation is not well defined and it only supports the single attacker environment. On the other hand, the RPPM

[1] algorithm does not require any prior knowledge about the network topology and it determines the certainty that the constructed graph is the attack graph when the algorithm terminates. To accomplish this goal, the graph reconstruction procedure of the original PPM algorithm is replaced and new technique is called rectified graph reconstruction procedure. And the packet marking procedure of PPM algorithm is not required to change.

4 Objectives & Proposed Work

RPPM [1] is the scheme which introduces the new termination condition of the PPM algorithm. Through the new termination condition, the user of the new algorithm is free to determine the correctness of the constructed graph. The constructed graph will reach the marking probability and the structure of the underlying network graph.

In the proposed work, the path will be constructed which the data packets should traverse. This path should dynamically change in case of traffic and failure in router. In packet marking procedure each packet will be marked with random values. These values are encoded edges' information on the packet arriving at the routers. After that, the router availability will be checked, depends upon the router availability the path will be constructed. The encoded values in the packet are retrieved and it decoded and checked with the generated code. The path will be re-constructed with the received packets, it validated with the constructed path.

5 Rectified Probabilistic Packet Marking Algorithms

The goal of this algorithm is that guarantees that the constructed graph is the same as the attack graph with probability greater than P* where P* is the *trace back confidence level*. To accomplish this goal, the graph reconstruction procedure of the original PPM algorithm is completely replaced, and the new procedure called rectified graph reconstruction procedure. On the other hand, the packet marking procedure put as it is so that every router deployed with the PPM algorithm is not required to change.

5.1 Rectified Graph Reconstruction Procedure

The pseudo code of the rectified graph reconstruction procedure is shown in Fig.3 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 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.

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.3 The pseudocode of the rectified graph reconstruction procedure of the RPPM algorithm.

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.

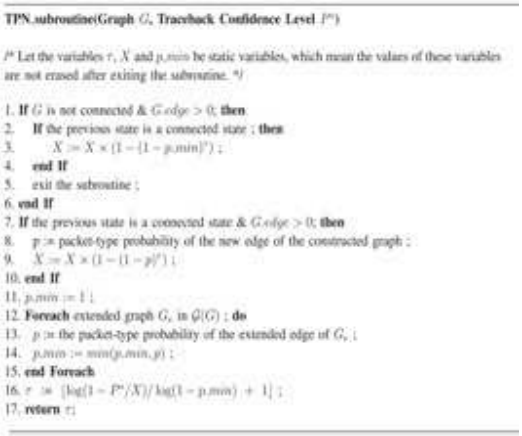
As suggested by the pseudo code, the termination condition of the RPPM algorithm is that "the counter for the incoming packets is larger than the TPN," and this implies that the calculation of the TPN during each update of the constructed graph is the core of the RPPM algorithm

6 TPN Generations

This section presents the calculation of the TPN at each connected state so that the RPPM algorithm returns a correct constructed graph, with probability larger than P*.

6.1 Termination Packet Number Calculation Pseudo code

Fig.4 shows the subroutine that calculates the TPN, and it is executed whenever the rectified graph reconstruction procedure enters a new state. When the routine is visited for the first time, the variable "X" that is used to store the accumulated state-change probability is initialized to one.


TPN.subroutine(Graph G, Traceback Confidence Level P^*)

/* Let the variables τ, X and p_min be static variables, which mean the values of these variables are not erased after exiting the subroutine. */

1. If G is not connected & $G.edge > 0$; then
2. If the previous state is a connected state ; then
3. $X := X \times (1 - (1 - p\_min)^{\tau})$;
4. end If
5. exit the subroutine ;
6. end If
7. If the previous state is a connected state & $G.edge > 0$; then
8. $p :=$ packet-type probability of the new edge of the constructed graph ;
9. $X := X \times (1 - (1 - p)^{\tau})$;
10. end If
11. $p\_min := 1$;
12. Foreach extended graph G_e in $\mathcal{G}(G)$; do
13. $p :=$ the packet-type probability of the extended edge of G_e ;
14. $p\_min := min(p\_min, p)$;
15. end Foreach
16. $\tau := \lfloor \log(1 - P^*/X) / \log(1 - p\_min) + 1 \rfloor$;
17. return τ;

Fig.4 The pseudo code of the TPN calculation subroutine.

Next, based on the connectivity of the current constructed graph, the variable "X" is updated in different ways: 1) if the current constructed graph is connected, the subroutine calculates the packet-type probability of the new edge and then updates the variable "X," and 2) if the current constructed graph is disconnected, the subroutine uses the minimum packet-type probability of the extended edge that was chosen from the extended graphs of the previous constructed graph, that is, "p min" in the pseudo code in Fig.4 Next, if the current constructed graph is disconnected, the TPN subroutine will not calculate the TPN, and one should exit the subroutine. Otherwise, the subroutine calculates the TPN. Finally, the subroutine returns the calculated TPN.

7 Applications

An attack graph is a visual aid used to document the known security risks of a particular architecture; in short, it captures the paths attackers could use to reach their goals. The graph's purpose is to document the risks known at the time the system is designed, which helps architects and analysts understand the system and find good trade-offs that mitigate these risks. RPPM has natural applications in solving the IP trace back problem which is a potential countermeasure against distributed denial-of-service (DDoS) attacks. In this problem, the internal network information at each

router is the IP address of its (incoming) interface and the goal of a RPPM scheme for IP trace back is to convey the entire IP-level path from the source to the destination.

8 Conclusions

The denial-of-service attacks motivate the development of improved trace back capabilities. There are various trace back algorithms based on packet marking and the PPM is one of them which is based on overloading existing IP header fields and its implementation is capable of fully tracing an attack after having received only a few thousand packets. But PPM algorithm lacks a proper definition of the termination condition. Which leads to an undesirable outcome: there is no guarantee of the correctness of the constructed graph produced by the PPM algorithm. So the RPPM 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.

9 References

1. "A Precise Termination Condition of the Probabilistic Packet Marking Algorithm" Tsz-Yeung Wong, Man-Hon Wong, and Chi-Shing (John) Lui, Senior Member, IEEE.

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

3. 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.

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

5. 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.

6. 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.

7. 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 Distributed Systems, July 2007.

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

10. S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Practical Network Support for IP Trace back", Proc. ACM SIGCOMM, pp. 295-306, 2000.

11. D.X. Song and A. Perrig, "Advanced and Authenticated Marking Schemes for IP Traceback", Proc. IEEE INFOCOM '01, Apr. 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 Trace back", Proc. ACM SIGCOMM '01, Aug. 2001.

13. K. Park and H. Lee, "On the Effectiveness of Probabilistic Packet Marking for IP Trace back under Denial-of-Service Attacks", Proc. IEEE INFOCOM '01, 2001.