IMPACT: International Journal of Research in Engineering & Technology (IMPACT: IJRET)
ISSN(E): 2321-8843; ISSN(P): 2347-4599
Vol. 4, Issue 1, Jan 2016, 29-36

SECURE NETWORK USING RECTIFIED PROBABILISTIC PACKET MARKING BASED ON TRACE BACK DEFENSE AGAINST DDOS FLOODING ATTACKS

ANIL V TURUKMANE[1] & S. K. YADAV[2]

[1]Research Scholar, JJTU, Rajasthan, India

[2]JJTU, Rajasthan, India

ABSTRACT

The most significant merit of the RPPM algorithm is that when the algorithm terminates, the algorithm guarantees that the constructed attack graph is correct with a specified level of confidence. We carry out simulations on the RPPM algorithm and show that the RPPM algorithm can guarantee 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 is a promising mean to enhance the reliability of the PPM algorithm. As attackers use automated methods to inflict widespread damage on vulnerable systems connected to the network, it has become painfully clear that traditional manual methods of protection do not suffice. This paper discusses an intrusion prevention approach, intrusion detection, response based on active networks that helps to provide rapid response to vulnerability advisories.

KEYWORDS: Structured Network, Secure Data Sharing, Secure Packets

INTRODUCTION

We design a new probabilistic packet marking technology -- P3M in this article. Comparing with the traditional PPM technologies, our first contribution is a new payload (called as P3M payload below) carrying router address and path identification to avoid influencing the normal running of recombining packets and QoS mechanism. Our second contribution is a new path identification scheme based on router addresses and hash algorithm. The use of path identification makes our probabilistic packet marking technology P3M simple when victim computes DDoS attack paths. And path identification also could be used by other network security equipment.

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 the distance fields (wherein Savage ET alohas 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 is given in Figure 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 cannot be applied under this multiple-attacker environment.

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**

Figure 1: The Pseudo Code of the Packet Marking Procedure of the PPM Algorithm

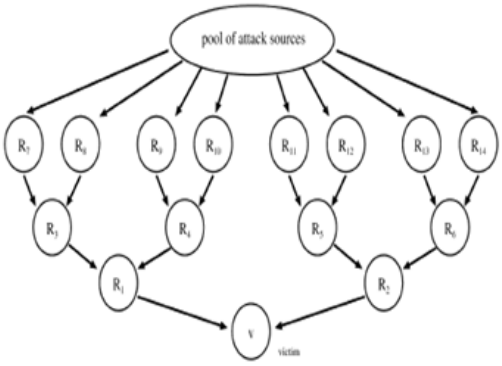

Figure 2: A 14-Router Binary-Tree Network the Upper-Bound Equation

ROUTER MAINTENANCE

The assumption that every router has only one outgoing route toward the victim This change may cause the attack packets to take more than one path toward to the victim, and the routers in the onstructed graph may have more than one outgoing edge.

Problem of Multiple Victim Routes

Originally, without considering routers that have multiple victim routes, the arrival of a new encoded edge will add only a new node and a new edge to the constructed graph (note that it is the worst-case execution scenario). However, when we allow a router to have multiple victim routes, the arrival of a marked packet that encodes a new edge can result in two different scenarios:

- A new node is added that is, one node plus one edge and
- No new node is added, which means that the new edge connects two existing nodes.

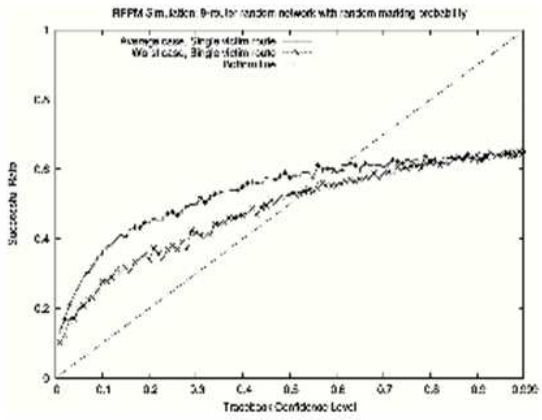

Figure 3: When the Routers Have More than One Victim Route, the RPPM Algorithm Cannot Guarantee the Correctness of the Constructed Graph when the Confidence Level is Larger than 0.59

The Simulation Environment

The testing network is a random-tree network with 10 nodes: one victim plus inner outers. However, this time, we allow the routers in the testing network to have more than one victim route. Again, the marking probability is set to a random numbering [0.1:0.9], a ditch value see the same for all routers.

THE SIMULATION RESULTS

Figure 3 shows the simulation results for both the average-case and the worst-case executions. For small values of the trace back confidence level, the successful rates of both execution modes are still over the bottom line. However, the successful rate of the worst-case execution falls below the bottom line when the trace back confidence level goes beyond 0.54, where a she successful rate of the average-case execution falls below the bottom line when the trace back confidence level goes beyond 0.59. One can conclude that the PPM algorithm cannot provide a guarantee of the successful rate in reconstructing the attack graph when the routers have multiple outgoing routes toward the victim.

Formulating an Extra Set of Extended Graphs

The new set of extended graphs is defined as follows

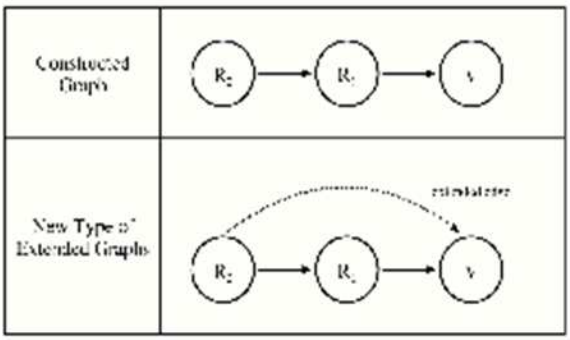

Figure 4: An Illustration of the Extended Graph with the Support of Multiple Victim Routes

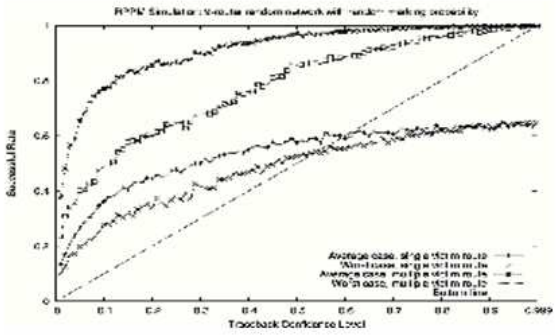

Figure 5: With the Support for Multiple Victim Routes, the RPPM Algorithm Can Provide the Guarantee of the Correctness of the Constructed Graph

Simulation: Support for Multiple Victim Routes

Shown in Figure 5. In this figure, the PPM algorithm can guarantee the correctness of the constructed graph, again, with the support of multiple victim routes. Technically speaking, the introduction of the extra set of extended graphs actually increases the value of the TPN. As the TPN increases, the successful rate therefore increases

TPN GENERATION

We $P_{\tau_i}(C_i \rightarrow C_{i+1})$ as the probability that the rectified graph reconstruction procedure proceeds from state *C*i to state *C*i+1 , with the TPN set to *T*i , and we name this probability the state-change probability from *C*i to Ciþ1 . In other words, it is the probability that the victim receives a new edge before the number of collected marked packets is larger than the TPN *T*i. Since the probability that the PPM algorithm that returns a correct constructed graph is equivalent to the probability that the RPPM algorithm makes a transition of n - 1 steps from states *C*1 to *C*n , mathematically, we have the following:

$$P(\text{constructed graph is correct}) = \prod_{j=1}^{n-1} P_{\tau_j}(C_j \rightarrow C_{j+1}).$$

Then, our claim is correct, given that the product of the state-change probabilities from states *C*1 to *C*n should be greater than P Ã and is given by

$$\prod_{j=1}^{i} P_{\tau_j}(C_j \rightarrow C_{j+1}) > P^*.$$

For the sake of further presentation, we transform the above equation as follows:

$$P_{\tau_i}(C_i \rightarrow C_{i+1}) > \frac{P^*}{X_{i-1}}, \quad \text{where } X_{i-1} = \prod_{j=1}^{i-1} P_{\tau_j}(C_j \rightarrow C_{j+1}). \tag{8}$$

*X*i-1 in (8) is the product of the state-change probabilities of the past states of the rectified graph reconstruction procedure, and we named it the accumulated state-change probability at state *C*i .

Termination Packet Number Derivation

According to the previous section, we know that the TPN at each connected state can be found by (8), which is expressed in terms of the state-change probability. In this section, we derive the TPN by deriving the state-change probability with the following steps:

- To recall, the state-change probability is the probability that the constructed graph of state *C*i evolves into the constructed graph of state *C*i+1. Hence, the first step in calculating the state-change probability is to find all the graphs that could possibly be the next constructed graph, and we name this set of graphs the extended graphs.
- In the second step, for each extended graph *G*e, we find the probability that the current constructed graph becomes the extended graph *G*e . As a matter of fact, the above probability is the state-change probability from *C*i to *C*i+1, conditioned that the extended graph *G*e is the next constructed graph, and we name this the conditional state-change probability.
- From the conditional state-change probability, one can find the state-change probability (and, thus, the TPN) through the definition of the condition probability. Nevertheless, because the calculation of the exact TPN violates the basic assumptions of the traceback problem, the upper-bounded TPN would alternatively be derived, and the relationship between the exact TPN and the upper-bounded TPN will be presented.

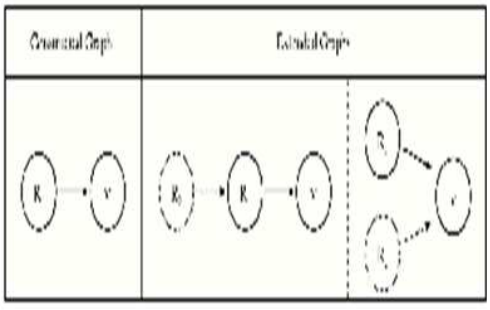

Figure 6: An Illustration of the Concept of the Extended Graph

RE-CONSTRUCTION PATH

The computational burden lies mainly on the procedure of path reconstruction. Reducing the total number of marked packets required for path reconstruction is therefore critical. First attempt to find the optimal marking probability,

then to enhance the marking mechanism, and finally to study the possibility of ‘‘reducing’’ the path length.

Denote k as the number of attack paths to the victim v. For path *j* (1 <= j <= k), the number of routers between the attack source and v is dj. Let p_j^i (m) be the marking probability of router *i* (1 <= i <= d*j*) along path *j*, and p_j^i (v) be the marking probability of router i along path j perceived by v. p_j^i (v) may be different from p_j^i (m), e.g., for PPM p_j^i (m) = p and p_j^i (v) = p(1- p)$^{d(j-i)}$. Denote Nj as the number of packets traversing along path j, and M_j^i as the number of packets marked by the *i*th router along path j and received by v. In other words, those packets initially marked by the *i*th router but are remarked by any subsequent router are not counted into M_j^i (v) Denote M*j* as the number of packets marked by any router along path j and received by v. Clearly, the expectations of M_j^i and M*j* are respectively.

$$E[M_j^i] = N_j p_j^i(v), \tag{1}$$

and

$$E[M_j] = E\left[\sum_{i=1}^{d_j} M_j^i\right] = \sum_{i=1}^{d_j} E[M_j^i] = N_j \sum_{i=1}^{d_j} p_j^i(v), \tag{2}$$

The Number of Marked Packets for Path Reconstruction

- The expected values of the total number of marked packets along path j

In PPM, $p_j^i(v) = p(1-p)^{d_j - i}$. From (2) we obtain

$$E[M_j] = N_j \sum_{i=1}^{d_j} p_j^i(v) = N_j(1-(1-p)^{d_j}). \tag{3}$$

- Probability of receiving at least one marked packet from each routerIn PPM, since each router conducts marking independently, therefore

$$P\{M_j^1 \geqslant 1; M_j^2 \geqslant 1; \cdots; M_j^{d_j} \geqslant 1\} = P\{M_j^1 \geqslant 1\} P\{M_j^2 \geqslant 1\} \cdots P\{M_j^{d_j} \geqslant 1\}. \tag{4}$$

That is,

$$P\{M_j^1 \geqslant 1; M_j^2 \geqslant 1; \cdots; M_j^{d_j} \geqslant 1\} = \prod_{i=1}^{d_j}(1 - P\{M_j^i = 0\}) = \prod_{i=1}^{d_j}(1 - [1 - p_j^i(v)]^{N_j}). \quad (5)$$

$$\text{Since } p_j^1(v) < p_j^2(v) < \cdots < p_i^{d_j-1}(v), \quad 1 - [1 - p_j^1(v)]^{N_j} < 1 - [1 - p_j^2(v)]^{N_j} < \cdots < 1 - [1 - p_j^{d_j}(v)]^{N_j}. \quad (6)$$

Combining with (5), we obtain

$$P\{M_j^1 \geqslant 1; M_j^2 \geqslant 1; \cdots; M_j^{d_j} \geqslant 1\} < (1 - [1 - p_j^{d_j}(v)]^{N_j})^{d_j} = (1 - [1 - p]^{N_j})^{d_j}. \quad (7)$$

Inequality (7) holds for any p (0 < p < 1). On the other hand, the maximum value of (5) can be obtained by taking the derivative of (5) with respect

to p, resulting in

$$p = \frac{1}{dj} \quad (8)$$

Thus, the maximum value of (5) can be reached if (8) is satisfied.

CONCLUSIONS

In this work, we have shown that there are some problems in PPM algorithm: the overwritten problem, limited marking field, low accuracy and so on. Dynamic probabilistic packet marking has solved these problems by using dynamic probability and fragment-reassembly. Meanwhile, using the expected number of required marked packets E½X as the termination condition is not sufficient Path reconstruction is the fundamental goal of packet marking. Reduced false positives. High false positives are actively suppressed due to the above improvements. Effectiveness to handle large-scale DDoS attacks which is dominant in today's Internet.

ACKNOWLEDGEMENTS

The author is extremely thankful for the respected guide for their encouragement.

REFERENCES

1. Software Engineering – A Practitioners Approach, 7th Edition by Pressman UML User Guide, By Grady Booch, James Rumbaugh and Ivar JacobsanF. Baker. Requirements for IP Version 4 Routers. RFC 1812, June 1995.
2. S. Savage, D. Wetherill, A. Karl in, and T. Anderson, "Network Support for IP Trace back," IEEE/ACM Trans. Networking, vol. 9, pp.226-237, Jun. 2001.

3. D. Song and A. Perrig, “Advanced and Authenticated Marking Schemes for IP trace back,” IEEE INFOCOM 2001, Anchorage, AK.
4. A. Year, A. Perrig, and D. Song, “Fast Internet Trace back,” IEEE INFOCOM 2005, in press.
5. T. Peng, C. Leckie and R. Kotagiri, “Adjusted Probabilistic Packet Marking for IP trace back,” Proc. Of Networking, 2002, Pisa, Italy, May 2002.
6. J. Liu, Z. Lee, and Y. Chung, “Efficient dynamic probabilistic packet marking for IP trace back,” the 11th International Conf. Networks (ICON 2003), Sydney, Australia, Sep. 2003..
7. B. Rizvi and E. Fernandez-Gaucher and, “Analysis of adjusted Probabilistic Packet Marking,” IP Operations & Management (IPOM2003), Kansas City, MO, Oct. 2003..
8. M. Adler, “Tradeoffs in Probabilistic Packet Marking for IPTraceback,” Annual ACM Symp. Theory of Computing’02, Quebec, Canada, 2002.
9. H. Aljifri, M. Smets and A. Pons, “IP trace back using header compression,” Computer & Security, vol.22, pp.