International Journal on Future Revolution in Computer Science & Communication Engineering
Volume: 3 Issue: 9
ISSN: 2454-4248
48 – 51

An Enhanced IP Trace Back Mechanism by using Particle Swarm System

N. Venkataramanan
Research Scholar, P.G and Department of Computer Science
Periyar E.V.R. College (Autonomous)
Trichy, Tamilnadu, India.
venkatramanan17.cs@gmail.com venbhu65@yahoo.in

Dr. T. N. Ravi
Assistant Professor of Computer Science
Periyar E.V.R. College (Autonomous)
Trichy, Tamilnadu, India
proftnravi@gmail.com

***Abstract*—**Internet is the most powerful medium as on date, facilitating varied services to numerous users. It has also become the environment for cyber warfare where attacks of many types (financial, ideological, revenge) are being launched. "Network forensics is a sub-branch of digital forensics relating to the monitoring and analysis of computer network traffic for the purposes of information gathering, legal evidence, or intrusion detection." Cloud Storage is a service where data is remotely maintained, managed, and backed up. The service is available to users over a network, which is usually the internet. It allows the user to store files online so that the user can access them from any location via the internet. The provider company makes them available to the user online by keeping the uploaded files on an external server. In this paper, a novel Digital Network Forensic Investigation Method is proposed. This paper will do changes in the analysis and investigation place of the network forensic. The investigation of the case will be based on the previous data collecting framework. The Spoofed IP address are classified by the previous framework and Enhanced IP trace back mechanism by Particle Swarm System is trace the real victim of the case in the network forensic.

Keywords- *Digital Network Forensic, Particle Swarm Optimization, IP Address, Trace Back Mechanism, Genetic Algorithm, Ant Colony Optimization, Map Reduce, Spoofed IP Address*

*****

I. INTRODUCTION

In network forensics, investigators also work to minimize system modification due to forensic activity. However, in these cases investigators often do not have the luxury of an offline copy. Moreover, network-based evidence is often highly volatile and must be collected through active means that inherently modify the system hosting the evidence. Even when investigators are able to sniff traffic using port monitoring or tapping a cable, there is always some impact on the environment, however small. This impact can sometimes be minimized through careful selection of acquisition techniques, but it can never be eliminated entirely.

The difficulty in identifying the origin of an attack over the Internet is termed the IP traceback (IPTBK) problem. Typically, the IPTBK problem involves collecting sufficient routing information to determine all possible paths between the attacker and victim under the constraints of the required number of routing packets and computational time. Solving the IPTBK problem is crucial in information security management for detecting the origin of a malicious attack and bringing the perpetrator to court. Many methods for reconstructing the attack path have been proposed [1][2][3][4][5]. These methods invariably assume the full cooperation of all servers between the victim and the command and control (C&C) server in providing the routing information required to reconstruct the attack path. However, in practice, some service providers may be unwilling (or unable) to provide this information, thus necessitating the reconstruction of the attack path using only partial knowledge of the routing information. Furthermore, the convergence time and the amount of routing information required to reconstruct the attack path must be minimized to ensure a prompt and effective response to perceived or actual attacks. Thus, the optimization scheme used for solving the IPTBK problem must not only be operable with limited routing information but also have low time complexity.

II. EXISTING IP TRACE BACK MECHANISM

Savage et al. [5] presented a packet marking–based approach named probabilistic packet marking (PPM) in which the load on the computational resources was reduced by marking each packet probabilistically using partial path information (i.e., the current router identity or the link between the current router and its downstream neighbour) during the forwarding process. Although PPM approaches have many advocates, they suffer several critical limitations; for example, multiple attackers cannot be traced simultaneously, and attackers can easily create spoofed IP addresses to construct a counterfeit attack path. Song et al. [6] proposed two new schemes, namely advanced and authenticated marking schemes, for tracing the approximate origin of the spoofed IP packets in a computationally efficient and robust manner.

Bellovin et al. [4] proposed a message-based method, iTrace, to assist web defenders in reconstructing the entire attack path. The Internet Control Message Protocol messages generated by the routers are used to forward packets towards the destination. Another useful but controversial means of reconstructing the attack path is through link testing: the defender injects a massive number of packets along a specific route and gradually filters the destination IP and ports of the upstream routers on the basis of the attack signature [8]. Yang and Yang [9] proposed a new hybrid IPTBK scheme, namely RIHT, with efficient packet logging. Each router has a fixed storage requirement (<320 KB) without requiring to refresh the logged tracking information while reconstructing the attack path. Song et al. [10] proposed a new anomaly detection method for malware behaviour analysis that automatically tunes and optimises the values of parameters for an intrusion detection system to filter and cluster processes against network threats such as DDOS. Later, Corona et al. [11] provided a systematic, high-level categorization of attack tactics by using a taxonomy technique through security event reconstruction for protecting against violation of defence mechanisms with the attack profile.

 ISSN: 2454-4248
48 – 51

A useful review of existing methods is available in [12]. In theory, all of the proposed methods are applicable to the IPTBK problem. However, they invariably assume the full cooperation of all service providers between the victim and the C&C servers in providing the routing information required to reconstruct the attack path. In practice, however, some of these service providers may be unwilling (or unable) to provide this information because of different autonomous systems.

III. PROPOSED IP TRACE BACK MECHANISM BY USING PARTICLE SWARM OPTIMIZATION

The following figure 1 represents the proposed Digital Network Forensic Investigation Framework and this framework is based on the OSCAR investigation methodology of network forensic. In the process of IP Address Check by Classification Method, the Hybrid Classification Model for IP Spoofing [13] is considered to classify the IP address as Spoofed and non-Spoofed IP Address.

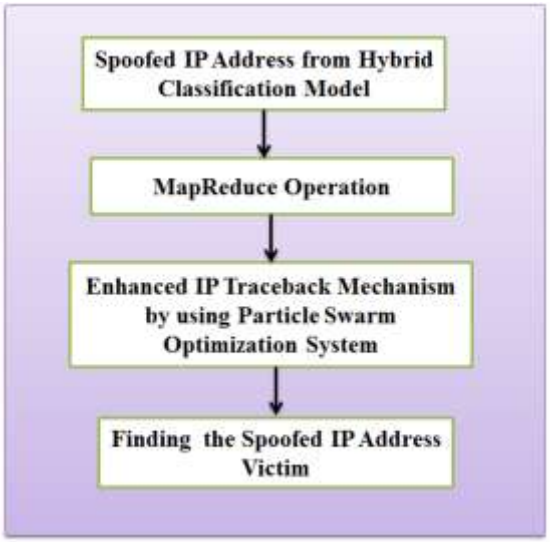

Figure 1. Enhanced IP Trace back Mechanism by using Particle Swarm Optimization.

IV. MAP REDUCE OPERATION FOR SPOOFED IP ADDRESS

MapReduce is suitable to be in cloud computing environment, as an example Google has proved automatically parallelize computations across large scale cluster. The Digital Preservation Coalition (DPC) is an advocate and catalyst for digital preservation, ensuring our members can deliver resilient long-term access to digital content and services. This report was written by Jeremy Leighton John, a specialist in the theory and practice of digital forensics in the context of personal, cultural and scientific archives. The report is published by the DPC in association with Charles Beagrie Ltd. Neil Beagrie, Director of Consultancy at Charles Beagrie Ltd, was commissioned to act as principal investigator for, and managing editor of this Series in 2011. Institutional repositories and professionals with responsibilities for personal archives can benefit from forensics in addressing digital authenticity, accountability and accessibility. Digital personal information must be handled with due sensitivity and security while demonstrably protecting its evidential value. Advancing capabilities promise increasingly effective automation in the handling of ever higher volumes of personal digital information.

MapReduce is a programming model for cloud computing in data processing, as in, which can achieve secure distributed storage and processing on large data sets efficiently, it has good scalability and ease of use, can be run through thousands of commercial machine as in,and provides a convenient programming interface to allocate data intensive work in the cluster. However, the development of MapReduce application in cloud present several challenges considering cloud as open public system. A MapReduce programming model drives in three fundamental phases:

1. Map phase: partition into M Map function (Mapper); each Mapper runs in parallel. The outputs of Map phase are intermediate key and value pairs.
2. Shuffle and Sort phase: the output of each Mapper is partitioned by hashing the output key. Number of partitions is equal to the number of reducers all key and value pairs in shuffle phase share the same key that belongs to the same partition. After partitioning the Map output, each partition is stored by a key to merge all values for that key.
3. Reduce phase: partition into R Reduce function (Reducer) each Reducer also runs in parallel and processes different intermediate keys. The first contribution of this work is to propose a framework for running MapReduce system in a cloud environment based on the captured requirements and to present its implementation on Amazon Web services. The second contribution is to present an experimentation of running the MapReduce system in a cloud environment to validate the proposed framework and to present the evaluation of the experiment based on the criteria such as speed of processing, data-storage us age, response time and cost efficiency

V. PARTICLE SWARM SYSTEM

The PSS algorithm is an extension of the original Particle Swarm Optimization algorithm, in which search efficiency is enhanced using a local updating rule in addition to the aforementioned global updating rule. The PSS algorithm was used for solving the IPTBK problem and identifying the most probable attack path under the constraints of the required number of routing packets (i.e., cost) and convergence time. The attack path was assumed to be a non cyclic directed graph to ensure the convergence of the solution procedure.

Let the network topology be represented as a directed graph, $G = (V, E)$, where V represents a set of nodes, $V = \{v1, v2,\ldots, vn\}$; Vs is a set of source nodes (i.e., attack sources), Vd is a set of sink nodes (i.e., victims), and E denotes the graph edges. To solve the IPTBK problem, two nodes are chosen arbitrarily from Vs and Vd as the attack source and victim, respectively, and the PSS algorithm was employed to determine the most probable attack path between them. The algorithm was implemented using the following three-step approach.

Step 1: Creating the Network Topology

To evaluate the minimum number of packets required by the PSS algorithm to reconstruct the attack path, various experimental network topologies were constructed using a random graph generator according to the Waxman model [8]. The generator randomly placed a total of p nodes at integer coordinates distributed over a rectangular area of size $n \times n$. Adjacent nodes, i v and j v, were then connected to form edges with a probability of

 ISSN: 2454-4248
48 – 51

$$P(i,j) = \eta \exp\left(\frac{-d(i,j)}{L\gamma}\right) ,$$

where $d(i, j)$ is the Euclidean distance between nodes *i v* and *j v* , and L is the maximum possible distance between any two nodes in the topology. In addition, h and g are parameters with values in the interval [0, 1] and are used to vary the graph characteristics. In particular, a higher value of h increases the average degree of the nodes, whereas a higher value of g increases the ratio of the number of long edges to the number of short edges.

Step 2: Reconstructing the Attack Paths

Step 2.1 A series of random attacks was simulated in which the Monte Carlo method was used to generate routing information having placed *m* particles at random starting nodes within the topology.

Step 2.2 Let each particle construct a tour (i.e., a feasible path between the victim and the attacker) by repeatedly applying the following state transition rule.

Step 3: Initialization of the Particle

The initialization phase is used to determine the position of the m particles. The random initialization is one of the most popular methods for this job. There is no assurance that a randomly created particle be a better answer and this will make the initialization more attractive. A good initialization algorithm makes the optimization algorithm more efficient and reliable. For initialization, initial information or knowledge of the problem can help the algorithm to converge in less iterations.

The best-fit particle of the entire swarm [23] influences the position of each particle. Each individual particle j $\in [1 \ldots m]$ where m > 1, has current position in search space s_j, a current velocity u_j and a personal best position $p_{b,j}$ where j is the smallest value determined by objective function o. By using $p_{b,j}$ the global best position G_b is calculated, which is the buck value obtained by comparing all the $p_{b,j}$

The $p_{b,j}$ is calculated by using the formula

$$p_{b,j} = \begin{cases} p_{b,j} \; if \; (y_j) > p_{b,j} \\ y_j \; if \; f(y_j) \leq p_{b,j} \end{cases} \quad (1)$$

The formula used to calculate Global Best Position G_{best} is

$$G_b = \{\min\{p_{b,j}\}, where \; j \in [1, \ldots, m] where \; m > 1 \quad (2)$$

Update the Velocity and Position

In each iteration, each particle updates its velocity and position according to its heretofore best position, its current velocity and some information of its neighbours. Equation (3) is used for updating the velocity:

$$V_i^{(k+1)} = w * V_i^k + C_1 * rand_1 * (pbest_i^k - S_i^k) + C_2 * rand_2 * (gbest^k - S_i^k) \quad (3)$$

The searching point in the solution space may be modified by the following equation:

$$S_i^{(k+1)} = S_i^k + V_i^{(k+1)} \quad (4)$$

The first term of Equation (3) is the previous velocity of the particle vector. The second and third terms are used to change the velocity of the particle vector. Without the second and third terms, the particle vector will keep on ‘‘flying’’ in the same direction until it hits the boundary. Namely, it corresponds to a kind of inertia represented by the inertia constant, *w* and tries to explore new areas.

Table 1: Terms and its abbreviations used in the Particle Swarm Optimization System

Term	Abbreviations
w	Weighting Function
V_i^k	Velocity of the i^{th} particle vector at k^{th} iterations
$C_1 \; and \; C_2$	are the positive weighting factors
$rand_1 \; and \; rand_2$	are the random numbers between 0 and 1;
S_i^k	is the current position of i^{th} particle vector $h(n)$ at k^{th} iteration;
$pbest_i^k$	is the personal best of the i^{th} particle at the k^{th} iteration;
$gbest^k$	is the group best of the group at the k^{th} iteration.

Algorithm: IP Trace Back by Particle Swarm System

Step 1: Start
Step 2: Construct the route graph using Waxman’s scheme
Step 3: Initialize *t*:=0; iteration number:=0;
Step 4: for j, k-1 to i

Step 5: Initialize the swarm for nodes on route (j,k)
Step 6: for t=1 to i
Step 7: Lay s swarm on starting node j
Step 8: for r-1 to n
Step 9: reset the velocity for every j and k
Step 10: for r-1 to n
Step 11: if swarm(r) not reached at the victim node
Step 12: Move the swarm to nearest node k
Step 13: update the $p_{b,j}$ using equation (1)
Step 14: add node k into r^{th} route solution
Step 15: for j, k-1 to i
Step 16: if route (j,k) is in r^{th} route solution
Step 17: then update the local best solution by using (1)
Step 18: Compute most probable route solution
Step 19: update the global best solution G_b by equation (2)
Step 20: If the spoofed IP attack is occurred in the IP address
Step 21: update the velocity of the swarm by using equation (3)
Step 22: if the stopping criteria (100 generations) not statisfied
Step 23: for r-1 to n
Step 24: Swap the starting node into r^{th} route solution.
Step 25: if loop_num<max_loop then again start from step 2.

VI. RESULT AND DISCUSSION

Here the result achieved is compared with other existing optimization techniques like Ant Colony optimization, Genetic Algorithm. The performance of the enhanced IP trace back is compared with the number of iterations used.

 ISSN: 2454-4248
48 – 51

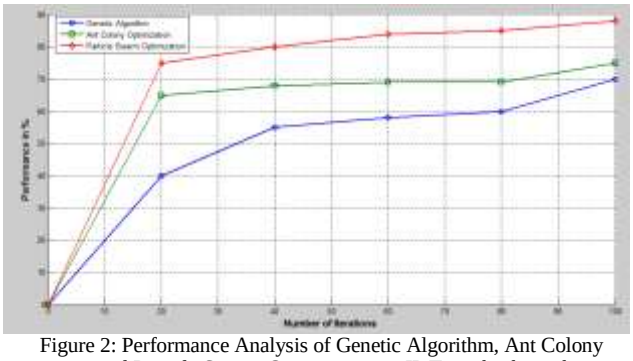

Figure 2: Performance Analysis of Genetic Algorithm, Ant Colony Optimization and Particle Swarm Optimization in IP Trace back mechanism

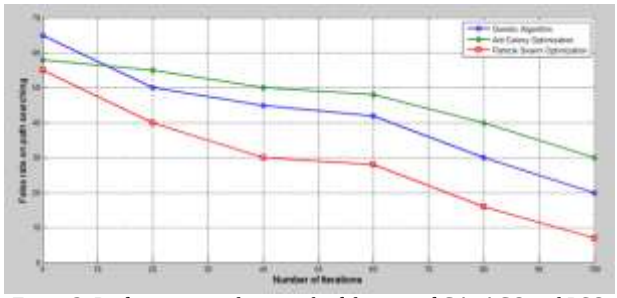

Figure 3: Performance analysis on the false rate of GA, ACO and PSO

In the figure 2, x-axis depicts the number of iterations and y-axis represents the performance of the optimization technique in the IP trace back problems. In figure 2, the Particle Swarm Optimization gives the better performance at the different number of iterations when it is compared with GA and ACO techniques.

In the figure 3, x-axis presents the number of iterations and y-axis depicts the false rate on the path searching. In this figure 3 also, PSO gives the less number of false rate when it is compared with ACO and GA techniques

VII. CONCLUSION

This paper presents a particle swarm optimization based scheme for solving the IP trace back problems in digital network forensics. In the proposed scheme, the efficiency of the particle to search all the feasible attack paths within the solution space was enhanced by partitioning the particle swarm into subgroups, where each subgroup applies a velocity updating rule. The use of a subgroup policy reduced the rate at which the velocity on the most probable attack path was updated, thus improving the global optimality of the final solution. The simulation results have confirmed the ability of the proposed Particle Swarm System – IP Trace Back (PSS-IPTBK) scheme in locating the true attack path even without the entire information or when the identity of the attacker is disguised using a spoofed IP address

REFERENCES

[1] A. Belenky and N. Ansari. On IP traceback, *IEEE Communications Magazine*, 41(7), (2003).

[2] A. Belenky and N. Ansari "IP Traceback with deterministic packet marking," *IEEE Communications Letters*, 7(4), (2003).

[3] A. C. Snoeren, C. Partridge, L. A. Sanchez, and C. E. Jones. Hash-based IP traceback, in *Proc. of Special Interest Group on Data Communication* (SIGCOMM), (2001) 27-31.

[4] A. C. Snoeren, C. Partridge, L. A. Sanchez, and C. E. Jones, S. T. Kent, and W. T. Strayer. Single- Packet IP Traceback, *IEEE/ACM Transactions on Networks*, 10(6), (2002) 721–734.

[5] S. Bellovin, M. Leech, and T. Taylor, ICMP traceback messages, Internet Draft: draft-ietf-itrace- 01.txt, (2001).

[6] S. Savage, D. Wetherall, A. Karlin, et al., Network support for IP traceback, *IEEE/ACM Transaction on Networks*, 9(3), (2001) 226–237.

[7] D. X. Song and A. Perrig. Advanced and authenticated marking schemes for IP traceback, in *Proc. of* the 20th Conference on Computer Communications (INFOCOM'01), 2, (2001) 878–86.

[8] R. Stone, CenterTrack: An IP overlay network for tracking DoS floods, in *Proc. of 9th USENIX Security Symposium*, Berkeley, Calif., (2000).

[9] M. H. Yang and M. C. Yang, RIHT: A novel hybrid IP traceback scheme, *IEEE Transactions on Information Forensics and Security* 7(4), (2012) 789-797.

[10] J. Song, H.Takakura, Y.Okabe, K.Nakao, Toward a more practical unsupervised anomaly detection system, *Information Sciences*, 231, (2013) 4–14.

[11] I. Corona, G. Giacinto, F. Roli, Adversarial attacks against intrusion detection systems: Taxonomy, solutions and open issue, Information Sciences, 239, (2013) 201-225.

[12] D. Martens, M. De Backer, R. Haesen, J. Vanthienen, M. Snoeck, B. Baesens. Classification with ant colony optimization, *IEEE Transactions on Evolutionary Computation*, 11(5), (2007) 651-665s.

[13] N. Venkataramanan, Dr. T.N. Ravi, "A Hybrid Classification Model for IP Spoofing in Network Forensic", I J C T A, 9(27), 2016, pp. 503-511.