

The Design and Implementation of a Practical Meta-Heuristic for the Detection and Identification of Denial-of-Service Attack Using Hybrid Approach

Hsia-Hsiang Chen

Department of Computer Science and Information
Engineering
National Chiao Tung University
Hsinchu, 300 Taiwan
allen.chen736@msa.hinet.net

Wuu Yang

Department of Computer Science and Information
Engineering
National Chiao Tung University
Hsinchu, 300 Taiwan
wuuyang@cis.nctu.edu.tw

Abstract—Network attacks are occurring continuously day after day. The researchers are expected to find the solution by identifying the address of source. We propose the IP traceback ant colony system (ITACS) algorithm to solve the IP traceback of denial of service (DoS) problem. The ITACS is novel attempted to apply in solving the problem. It is a meta-heuristic algorithm, which is a technique applies so that attack detection and attack identification can be implemented. The proposed algorithm has improved by the previous one to conquer this problem successfully. We obtained the data set of topology from one of famous research organizations for the experiment. The parameters of algorithm are considered by packet contents in topology. In the meanwhile, we discussed the increment of traffic condition. By the experiment, the examples of increment of traffic are above average 70%. The results show that the performance of ITACS algorithm is efficient and accurate. Furthermore, the proposed algorithm has also nature of robust for the problem. Future work may even be extended to study the other behaviors of organisms from derivations of meta-heuristic algorithm.

Keywords—IP traceback ant colony system; denial of service; meta-heuristic algorithm

I. INTRODUCTION

The network security is more and more important and is subjected to household, organization or government [1]. There are many famous attacking events that were demonstrated through examples such as eBay, Yahoo, Amazon.com and FBI Web sites. Such attacks cause financial damage of enterprises [2]. Most behaviors of attack belong to the denial of service (DoS) [5]. Many researchers have researched about the IP traceback problem few years ago [6]. Therefore, some methods had been proposed to solve the problem. For instance, the link testing, marking and ICMP-based had been applied to store IP or reduce memory resource in order to trace the IP address of attacker [7-10].

Nevertheless, few heuristic algorithms are studied in IP traceback. The heuristic algorithm has the congenital advantage to search the combinatorial optimal problem. The nature of an organism is to search food for a living, similar to IP traceback [11-12]. A few researchers simulated and designed the topology by using their arbitrary manners [13]. Similarly, the most of literatures were lack of the experiment design for increment of traffic.

For these reasons, we proposed IP traceback ant colony system (ITACS) algorithm to solve the problem. The implemented autocatalysis mechanism in the algorithm can be to search for attack path. We take the traffic flow and time into consideration. As a result, the experiment reveals that the algorithm can find attack path accurately and efficiently. The rest of this paper is organized as followed: we describe the IP traceback problem in Section 2 and the application of the ITACS algorithm in IP traceback problem in Section 3. The experimental design and analysis are discussed in details in Section 4. Finally in Section 5, we discuss the conclusion and future work regarding ITACS in IP traceback problem.

II. IP TRACEBACK

A. Objective

When the attack on the network occurs, victim hopes to find the attacker's real address. The traceability of the traceback mechanism would be applied. Moreover, the attacker will spoof the address so that finding the source address would be difficult. There are many mechanisms of IP traceback that aims to search correct attack path with applying certain technique. Nevertheless, we propose an algorithm to find the attack path accurately and efficiently via hop by hop. The proposed algorithm will be verified the performance index that includes the robustness, accuracy, efficiency, effectiveness and convergence. The results of experiment can be demonstrated the contention.

B. Filtration

We analyze the topological data sets and code program from Internet databanks. After parsing the relative parameters of topology, it obtains the traffic time, traffic flow, as well as the numbers of nodes and edges to be input parameters into algorithm. Simultaneously, the algorithm runs to search behavior by the parameters. The method of traffic filter is also applied to limit the threshold of traffic for stopping and improving the algorithm.

C. Hybrid Form

The proposed algorithm incorporated all three methods, which are ingress filtering, link testing and logging. It obtains the traceback information by packet log and filters traffic flow to decide termination condition. When the algorithm is launched to trace the path, it would search the

upstream router recursively until the real source is found. The ITACS algorithm would construct the path gradually. The abnormal traffic is also found initially as ingress filtering and the accumulative pheromone quantities as candidate of attack signature. Moreover, the map in topology is crucial for providing the parameters of the algorithm as controlled flooding. Hence, the proposed algorithm is a hybrid form in order to fit the IP traceback problem.

III. ITACS ALGORITHM

A. Background

Many researchers studied biological behaviors issues. Ants search for food from nest to target is called the shortest path problem. Dorigo etc. addressed the framework of ant system. Furthermore, it applied in a variety of different fields, for example, biology, scheduling of practical factories and TSP problem etc. A few papers illustrate the concept. We consider the IP traceback ant colony system (ITACS) algorithm as the solution to DoS problem in the paper. The algorithm improves several features to be better than traditional one. The paper also addresses the ITACS algorithm to solve IP traceback problem and verifies the efficiency and effectiveness in phase.

B. State Transition Rule

When an ant wants to go from router r to router u , it can search the next router by state transition rule. The formulation of state transition rule consists of two parts that involve the exploitation and exploration manners. First, we must set the limitation value to parameter q_0 . If the random variable q is less than q_0 , the exploitation will search candidate router step by step. We choose $\tau(r, u)$ that represents edge of pheromone quantities from router r to router u and $\eta(r, u)$ represents information flow from router r to router u . The maximum of multiplication is chosen to decide next router. Otherwise, we apply the exploration manner to avoid falling into local optimal value, and the exploration bases on the random proportional rule. To begin, we get a random number, and determine whether its value is greater than any candidate router value of $[\tau(r, s)][\eta(r, s)]$ in the random proportional rule. If it was found, we pick up a certain candidate router that conforms to the rule given by equation (1.1) and (1.2).

$$s = \begin{cases} \arg \max_{u \in J_k(r)} \{[\tau(r, u)][\eta(r, u)]^\beta\} & \text{if } q \leq q_0 \quad (\text{exploitation}) \\ S & \text{otherwise} \quad (\text{biased exploration}) \end{cases} \quad (1)$$

$$p_k(r, s) = \begin{cases} \frac{[\tau(r, s)][\eta(r, s)]^\beta}{\sum_{u \in J_k(r)} [\tau(r, u)][\eta(r, u)]^\beta} & \text{if } s \in J_k(r) \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

C. Global Updating Rule

When searching for a certain path by an ant, it would reinforce the pheromone quantity by all of edges and increases the probability to choose the same path for the next iteration. If the attack path found a real source router, the nature of algorithm would converge quickly to find the attack path. We deposited the pheromone quantity on edge last iteration and decided the pheromone quantity on edge by changing the pheromone quantity next iteration. The parameter α is also considered to be weight of pheromone quantity. The change of pheromone quantity was decided by traffic flow and time. The formulation is described as follows:

$$\tau_{ij}(t+1) = (1-\alpha) \cdot \tau_{ij}(t) + \alpha \cdot \Delta \tau_{ij}(t, t+1), \quad \Delta \tau_{ij}^k = \frac{Q_k}{L_k}, \quad (3)$$

$$\Delta \tau_{ij}(t, t+1) = \sum_{k=1}^m \Delta \tau_{ij}^k(t, t+1)$$

D. Local Updating Rule

When an ant went from router i to router j step by step, it would update pheromone locally on edge. The pheromone would be evaporated by time the volatility rate is crucial enough to affect it. The $\Delta \tau_{ij}(t)$ was decided by the Ant-Q that has $\Delta \tau_{ij}(t)$ and $\Delta \tau_{ij}(t) = \gamma \cdot \max_{z \in J_k(s)} \tau(s, z)$, $0 \leq \gamma \leq 1$ respectively. After updating the pheromone on edge, the formulation is described as follows:

$$\tau_{ij}(t) = (1-\rho) \cdot \tau_{ij}(t) + \rho \cdot \Delta \tau_{ij}(t) \quad (4)$$

It decreases the edge of pheromone quantities from router i to router j during iteration t . $\Delta \tau_{ij}(t)$ has two values for Ant-Q and $\Delta \tau_{ij}(t) = \tau_0$. The Ant-Q represents $\Delta \tau_{ij}(t) = \gamma \cdot \max_{z \in J_k(s)} \tau(s, z)$, $0 \leq \gamma \leq 1$. A $J_k(S)$ is router set and we choose the maximum value of all of the router set. ρ is a parameter and $(1-\rho)$ represents the evaporation of pheromone by local updating rule.

E. Terminal Condition

We considered the stopping situation for the problem. The outgoing flow had to be large than the incoming flow within the threshold, which is the information flow of a certain router for terminal condition. The formulation is as follows:

$$\sum_{h \in J_k(i)} \tau_{hi} \eta_{hi} - \sum_{j \in J_k(i)} \tau_{ij} \eta_{ij} < I_{stop} \quad (5)$$

IV. EXPERIMENTAL DESIGN AND ANALYSIS

We simulated the algorithm to verify performance on DoS attack. The topology had constructed by our test environment. The proposed algorithm tested the performance in IP traceback problem. In our case, there are two parameters, one is information flow and the other is pheromone, therefore the two parameters that represent $\tau(\mathbf{r}, \mathbf{u})$ and $\eta(\mathbf{r}, \mathbf{u})$ respectively in state transition rule.

A. Parameters for Meta-Heuristic Algorithm

We define the relative parameters of proposed algorithm as follows:

I_MAX: number of iteration

K_MAX: number of ant

R_MAX: number of router

q_0 : The parameter decides the exploitation or exploration by state transition rule

τ_0 : Initial value of pheromone

β : The parameter could scale the information flow

ρ : The evaporation of pheromone rate

α : The parameter decides to reinforce pheromone

r_i : Router i

u_k : Choose router j from the set u_k

τ_{ij} : The pheromone from r_i to r_j

η_{ij} : The traffic flow from r_i to r_j

RP: A random probability that generates by randomization

T: The threshold that is difference of incoming and outgoing

f : Traffic flow

B. Measurements

When the increment of traffic (IF) situation would be discussed, we should design the measurement indices by new traffic (NT) and old traffic (OT) for the increment of traffic experiment. The accuracy is important measurement to analyze the result of the experiment. Therefore, we considered calculating the ratio by true positive (TP) and total sample size. The detailed description are shown in formulation 8 and 9 as followed:

$$I.F.(%) = [(NT - OT) / OT] * 100 \% \quad (6)$$

$$Accuracy(\%) = \frac{TP}{Total_samp \quad le_size} \quad (7)$$

C. Topology and General Experiment

The data of experiment design is obtained from DARPA 2000 data set at the Lincoln Laboratory [14]. Topology includes the traffic flow, the connection relation from upstream router to downstream router and transmission time. The total number of routers is 372. Each problem is DoS that owns arbitrary attack path. We set traffic data and

topology information into our algorithm. The general experiment considered the significant traffic flow that has alternative value and no exceed the maximum. Two levels, number of ants and iterations, are categorized for each problem. The accuracy has two trails and each trial runs twenty times. After algorithm ended, we found the AVG time of total problems is less than 1 minute. Furthermore, all of them are greater than 95% for both small and large problem. The results of the experiment are shown in Table 1 as followed:

Example	Number of Ants and Number of Iterations	Accuracy rate	AVG time
Attack1	S: (10, 5)	97.5%	1 sec
	L: (500, 20)	95%	2 sec
Attack 2	S: (10, 5)	100%	1 sec
	L: (500, 20)	100%	34 sec
Attack 3	S: (10, 5)	100%	1 sec
	L: (500, 20)	95%	1 sec
Attack 4	S: (10, 5)	97.5%	1 sec
	L: (500, 20)	95%	1 sec
Attack 5	S: (10, 5)	97.5%	1 sec
	L: (500, 20)	100%	32 sec
Attack 6	S: (10, 5)	95%	1 sec
	L: (500, 20)	97.5%	40 sec

TABLE I. General Experiment

D. Increment of Traffic Experiment

The six different attacks consist of four levels of traffic increment, which are 1%, 5%, 10% and 30% for each trial. Moreover, the three levels of traffic threshold are 100, 50 and 0 for each trial. Each experiment runs 10 times with three trials each time in order to obtain accuracy and average. Table V shows the accuracy of more than 90% and more than 70% for other examples. Furthermore, Table III, Table VI and Table VII show the failure to search for the attack path when 1% increment of traffic is increased. On the contrary, another examples show successful attempts in searching for the attack path. The variance of accuracy is still stable even when the increment of traffic changes. Therefore, we find the proposed algorithm owns a robust behavior. Similarly, in most cases, the best traffic threshold set is 100 as shown in Table V, Table VI and Table VII. Other cases are 0 as shown in Table II and Table III. Therefore, the traffic threshold variation would affect accuracy significantly. In particularly, Table VI and Table VII show the different traffic threshold levels compared from Table II to Table V throughout the experiment. Figure 1 shows the majority of accuracy falls between 70% and 80% except for Table III, Table VI and Table VII, which the traffic of attack in all percentage levels cannot be found. When the 30% increment of traffic is reached, the experiment results in unit accuracy close to or exceeds average accuracy. Moreover, each curve appears smaller variance as linear.

TABLE II. Increment of Traffic Experiment for Example 1

	1%	5%	10%	30%	Detection AVG
100	63.3%	63.3%	80%	80%	71.67%
50	70%	76.67%	53.33%	73.33%	68.33%
0	80%	80%	76.67%	60%	74.17%
AVG	71.1%	73.32%	70%	71.11%	71.38%

TABLE III. Increment of Traffic Experiment for Example 2

	1%	5%	10%	30%	Detection AVG
100	0%	70%	70%	56.67%	65.56%
50	0%	80%	70%	76.67%	75.56%
0	0%	63.33%	86.67%	76.67%	75.56%
AVG	No detection	71.11%	75.56%	70%	70.23%

TABLE IV. Increment of Traffic Experiment for Example 3

	1%	5%	10%	30%	Detection AVG
100	70%	93.33%	63.33%	80%	76.67%
50	80%	83.33%	93.33%	90%	86.67%
0	60%	80%	63.33%	76.67%	70%
AVG	70%	85.55%	73.33%	82.22%	77.78%

TABLE V. Increment of Traffic Experiment for Example 4

	1%	5%	10%	30%	Detection AVG
100	100%	83.33%	96.67%	96.67%	94.17%
50	80%	86.67%	96.67%	83.33%	86.67%
0	90%	96.67%	90%	96.67%	93.34%
AVG	90%	88.89%	94.45%	92.22%	91.39%

TABLE VI. Increment of Traffic Experiment for Example 5

	1%	5%	10%	30%	Detection AVG
100	0%	73.33%	86.67%	83.33%	81.11%
50	0%	60%	66.67%	96.67%	74.45%
1	0%	80%	80%	73.33%	77.78%
AVG	No Detection	71.11%	77.78%	84.44%	77.78%

TABLE VII. Increment of Traffic Experiment for Example 6

	1%	5%	10%	30%	Detection AVG
10	0%	86.67%	63.33%	76.67%	75.56%
5	0%	70%	56.67%	66.67%	64.45%
0	0%	73.33%	73.33%	73.33%	73.33%
AVG	No Detection	76.67%	64.44%	72.22%	71.11%

V. CONCLUSION AND FUTURE WORK

We considered the major facts, traffic flow and time stamp for finding the target in the proposed algorithm. Furthermore, the meta-heuristic algorithm provides two advantages in solving the attack detection and attack identification. The experimental environment of DoS topology is considered by proposed algorithm. In these cases, this paper discusses the increment of traffic. The results show the accuracy above average 70% in different examples. It is to maintain the efficacy and effectiveness. We verify the proposed algorithm by the experiment. As a result, the algorithm is robust and precise to find attack path. By the results of experiment of simulation, the ITACS found the attack path in a short time and converged fast correspond to lower bound. The future work might extend to study other nature of organism to derive the meta-heuristic algorithm to solve network security problem.

ACKNOWLEDGEMENTS

Many thanks to Professor Wu Yang and my laboratory members for providing valuable thesis guidance and advice.

REFERENCES

- [1] J. A. Rochlis, and M. W. Eichin, "With microscope and tweezers: The worm from MIT's perspective," ACM Communication, vol. 32, 1989, pp. 689-698.
- [2] L. Garber, "Denial-of-service attacks rip the Internet," IEEE Computing, vol. 33, 2000, pp. 12-17.
- [3] J. Mirkovic, and P. Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," ACM SIGCOMM Computer Communications Review(CCR), vol. 34, 2004, pp. 39-54.
- [4] K. C. Chang, "Defending against flooding-based distributed denial-of-service attacks: A tutorial," IEEE Communication Magazine, vol. 40, 2002, pp. 42-51.
- [5] V. D. Gligor, "A note on denial-of-service in operating systems," IEEE Transactions on Software Engineering, vol. 10, 1984, pp. 320-324.
- [6] Belenky and N. Ansari, "On IP traceback," IEEE Communication Magazine, vol. 41, 2003, pp. 142-153.
- [7] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Network support for IP traceback," IEEE/ACM Transactions on Networking, vol. 9, 2001, pp. 226-237.
- [8] Network Group, "ICMP traceback messages," AT&T Labs Research <http://www.lasr.cs.ucla.edu/save/rfc/draft-bellovin-itrace-00.txt>, 2000.
- [9] Pacific Institute for Computer Security, "Security Fun with OCxmon and cflowd," San Diego Supercomputer Center, <http://www.caixa.org/funding/ngi1998/content/security/1198/>, 1998.
- [10] R. Stone, "Centertrack: An IP overlay network for tracking DoS floods," Proc. USENIX Security Symp, 2000, pp. 199-212.

- [11] M. Dorigo, V. Maniezzo, and A. Colomi, "Positive feedback as a search strategy," Milan, Italy: Politecnico di Milano, Dipartimento di Elettronica, Tech. Rep. 1991, 91-016.
- [12] G. H. Lai, C. M. Chen, B. C. Jeng, and W. Chao, "Ant-based IP traceback," Expert System Systems with Application, vol. 34, 2008, pp. 3071-3080.
- [13] M. Goodrich, "Probabilistic packet marking for large-scale IP traceback," IEEE/ACM Transactions on Networking, vol. 16, 2008, pp. 15-24.
- [14] Massachusetts Institute Technology, "MIT 2000 DARPA intrusion detection evaluation data set," Lincoln Laboratory, <http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/index.html>, 2000.

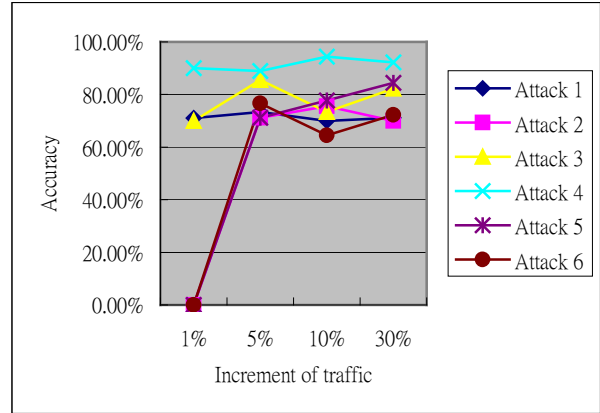


Figure 1. Increment of traffic to accuracy.