An Analysis Model of Botnet Tracking based on Ant Colony Optimization Algorithm

Ping Wang  Tzu Chia Wang*  Pu-Tsun Kuo  Chin Pin Wang

Department of Information Management
Kun Shan University, Taiwan
TEL: 886-6-205-0545
FAX: 886-6-205-0545
E-mail: tzuchia@mail.ksu.edu.tw

Abstract- To effectively counter DDOS attacks from bot herder, defenders have developed approaches to successfully traceback IPs of botnet C&C via logging the suspicious flow information of routing routers. Yet, available botnet detection schemes all supposed that ISPs would be cooperative to offer the complete routing information for path reconstruction. In practice, ISP’s service constantly is a mutual benefit for intelligence exchange. Therefore the constraint, require cooperation between ISPs, ought to be relaxed. A new IP traceback scheme based on ant colony optimization (ACO) algorithm is proposed for incomplete routing logs are provided. The aim of our work is to develop an analysis model for reconstruction of attack paths to traceback the botnet C&C via ant-inspired collective intelligence by calculating the pheromone to find possible routes with support and confidence degree. The validation of model uses NS2 (Network Simulator, version2) compiled by dark IP map, to simulate the scenario of fake IP attack, to test the effectiveness of model. Furthermore, sensitivity analysis is conducted to investigate significant parameters’ effect on the output of attack paths. Experimental results show that the proposed approach effectively suggests the best attack path of botnet in a dynamic network environment.

Key words: Botnet, bot, zombie, attack path, ant colony optimization

I. INTRODUCTION

The botnet refers to a group of computers, manipulated by herder via malicious codes inserted. After malware has been successfully installed in a victim, it becomes a zombie (compromised hosts) and will accept the remote commands from the bot herder. CSI/FBI reported average losses due to security incidents are $234,244 per respondent in 2009. Noticeably, 22% of respondents stated that they notified individuals whose personal information was breached [3]. According to reports in [4,5,7], botnet has become a major threat to steal victim’s private data, instead of massive connections to suspend network services like DDOS (Distributed Denial of Service) attacks.

Defenders developed some approaches to detect and classify botnet via IDS, honeypot and network flow information. [2, 4-5] In practice, a botnet uses a one-to-many control links between command and control (C&C) and victims. There are two main types of malicious bots in past two decades: IRC and peer-to-peer.

Available IP traceback approaches focused on DDoS network attacks. Two promising solutions to the IP traceback are passive and pro-active traceback to discover the possible attack paths that are the traits when an attacker exploited to achieve his intention. The former can be only used for after an attack, such as PPM, iTrace and SPIE, and its pros are that need no additional router storage requirements. Their cons are supposed that ISPs of attack paths will cooperate and provide the defender the complete path information. If one of four routers has lost the path information, these approaches would be failed, especially when lacking of full administration authorities of routers in the Internet.

The latter is supposed attack remains active until trace complete, so it can be used for an on-going traceback, such as input debugging, link testing, overlay network,...etc. This type of approaches need specify attack network flows from normal network flows immediately by collecting and examining online routing information in the underlying attack environment. In further, it also needs additional router storage to pile up logs for analyzer to investigate the true attack sources. The cons of these schemes are (i) massive messages should be recorded and transported to a repository for further assessing appropriate attack paths, It brings about the situation that considerable management overhead and routing information might be erased, because extra routing information being stored,(ii) Require cooperation between ISPs.

Though available IP traceback approaches are capable of detecting the zombies via accumulating attacking packets, they are all based on an idealistic assumption --all ISPs would be cooperative to offer the entire path information for attack path reconstruction. Generally, ISP’s service is a mutual benefit for intelligence exchange or followed by criminal law & procedure. Therefore the constraint, require cooperation between ISPs, must be relaxed. A new IP traceback scheme based on ant colony optimization (ACO) algorithm is proposed to discover all possible attack paths with support and confidence degree for locating C&C and herder IP even if the path reconstruction is in an incomplete routing environment.

Our work incorporates the ant colony optimization (ACO) algorithm to develop an analysis model to trace the herder IP via path reconstruction, discover all possible attack paths with support and confidence degree for locating C&C and herder IP even if the path reconstruction is in an incomplete routing environment.

Our work incorporates the ant colony optimization (ACO) algorithm to develop an analysis model to trace the herder IP via path reconstruction, discover all possible attack paths with support and confidence degree for locating C&C and herder IP even if the path reconstruction is in an incomplete routing environment.

Our work incorporates the ant colony optimization (ACO) algorithm to develop an analysis model to trace the herder IP via path reconstruction, discover all possible attack paths with support and confidence degree for locating C&C and herder IP even if the path reconstruction is in an incomplete routing environment.
The rest of the paper is organized as follows. Section II provides the background information of the IP traceback techniques, Section III reports our approach. Simulations and analysis on the results are presented in Section VI. Section V performs a sensitivity analysis and discusses the effect of control variables to output attack paths. Conclusions and future studies follow in Section VI.

II. RELATED WORKS

From 1990s, a number of IP traceback techniques have been developed and applied in DDOS network attacks. In theory, to reconstruct routing information in attack paths generates possible attack paths. Basically, defender needs to answer the following critical questions for IP traceback techniques: (i) how much storage of path information can successfully reconstruct an attack path. (ii) How fast can effectively track back to attack origin. The aim of IP traceback technique is to locate the proper attack origin by analyzing information of attack paths, especially when incomplete path information provided. Furthermore, because attacker can easily create source addresses by spoofed IPs to bother the defender; in other words, disguise their location using fake IP addresses; hence the true origin is lost. Thus requirements of our IP traceback scheme at least should involve the requirements: (i) discriminate the attack between real IPs and spoofed IP; (ii) report how much storage of path information (a group of successive routing packets) to reconstruct an attack path with a quantified index, such as probability, support degree or confidence degree.

In the following, we briefly review these approaches as follows: (i) Logging: embed a log function in a router such that can record all traffic packets. Once attack happened, data mining technique is employed to traceback attack source from data repository. It caused heavy overhead on the network and considerable management overhead. Snoeren (2001)[1] improved and proposed a mechanism, SPIE (Source Path Isolation Engine) to reduce management and network overhead via logging the hashing value of network packet, generated by bloom filters storing and forwarding a 32-bit hash digest for each IP data packet. It effectively reduced the transmission periods of resource demands for storing massive network packets, but cause heavy load on server computation.

Savage (2000)[10] designed a packet marking (PM) approach to resolve high demand on computational resources using marks packets, derived by adjacent routers deterministically or probabilistically. A well-known scheme, PPM (Probabilistic Packet Marking) can trace attack source by marked packets, originated from an attacker to victim, without causing heavy overhead to network traffic. It also answers the question that how many packets can successfully reconstruct the attack paths. Basically, there are three varies types: (i) Node append (ii) edge sampling (iii) node sampling. Though PM approaches have many advocates, their drawbacks are clear. First, it cannot trace multiple attackers. Second, shift packet marking overhead to router. Third, attackers can easily create spoof of IP and make a counterfeit of fake path.

To decrease computational loading on router, Bellovin et al. (2000) [9] replaced marked packet by icmp message, named iTrace. This approach assists defender to reconstruct the entire attack paths via router responses icmp messages to destination site on demand (intention). A useful but controversial way to rebuild attack path is link testing approach – defender filters the destination IP and port of upstream router based on attack signature and reversely traceback herder IP step by step, when attack accident detected. A famous method is the controlled flooding that injected massive packets on specific route to discover the congestion phenomenon of router (ex, packet drops) for looking for an attack path.

The aforementioned approaches are supposed that all ISPs are cooperative as well as provide the entire routing information to the victim. When this constraint is violated, then these approaches may fail.

III. AN ANALYSIS MODEL FOR ATTACK PATHS

Our works is to develop an analysis model for attack paths of botnet, estimate the support and confidence of each attack path for appropriately selecting the correct attack path under the circumstance of those only partial ISPs cooperate.

3.1 Basic idea

Basically, attack path reconstruction is started from victim via upstream links and recursively repeat until attack source is located, as illustrated in Fig.1. In other words, receiving each routing packet to form the path information enables the victim to reconstruct the whole attack path. Thus, path reconstruction can be considered as a special graph optimization problem. ACO is applied to attack path reconstruction via ants laying down a trail pheromone for attracting and guiding other ants back to the nest with food. Attack paths, such as two examples of attack paths, P1 – P2 – P3, P4 – P5 – P6 – P7, and also display the possibility to each attack path based on colony pheromone.

![Figure 1. Traceback of possible attack paths](image_url)

Inspired by the wisdom of natural ants, ants use pheromone on the trail to find their ways back to nest from victim to attack.
source. A moving ant lays some pheromone (in varying quantities) on the ground, thus marking the path it follows by a trail of this substance. While an isolated ant moves essentially at random, an ant encountering a previously laid trail can detect it and decide with a high probability path to trace it.

A group of ants, \( A_q, q=1,...,m \), are assigned to get foods and ants travel at a random walk. There are \( n \) ants walked on the high pheromone tail (\( P_i \)) when ants find their way back to nest, then support degree of attack path \( P_i \) (\( sup \ P_i \)) is given by

\[
sup_{P_i} = \frac{n}{m}
\]  

(1)

In the following, confidence degree of attack path (\( conf \ P_i \)) can be derived by the ratio of the ant number on attack path \( P_i \) to ant numbers on all attack paths with the minimal support degree.

\[
conf_{P_i} = \frac{\sup(p_1, p_2, ..., p \geq \min \ p)}{\sum \sup(p_1, p_2, ..., p \geq \min \ p)}
\]  

(2)

where \( \min \ p \) represents the minimum support degree, denoted that \( p \geq \min \ p \). High confidence degree hints the higher probability of attack path is.

### 3.2 IP Traceback Scheme for Botnet

It assumed that attack path be a non-cyclic directed graph to avoid entering an infinite number of loops, if selecting a cyclic graph. We focused on two issues of path reconstruction: (i) Find all directed paths between two arbitrary nodes, (ii) discover possible attack paths to identify the most possible attack path.

Let network topology be a directed graph, \( G=(V,E) \), where \( V \) represent a set of nodes, \( V=\{v_1, v_2, ..., v_n\} \), \( Vs \) is a set of source nodes, \( Vd \) is a set of sink nodes (victims), \( E \) denotes the edge of graph. The analysis model of IP traceback for botnet is stated by the following four steps:

**Step 1: Build up a network topology**

Our works focuses on security management issues of web services; therefore, a service-oriented topology for model analysis will be established based on location of service centers.

**Step 2: Decide the number of paths between two nodes**

Basicallly, a path in a digraph is a sequence of vertices from one vertex to another vertex. Here a set of edges those represent the flow through the whole attack path. According to Skvarcius and Robinson (1986) [11], the number of non-cyclic paths between two network nodes can be derived by the followings:

If the length of an edge may be represented by \( E \), then \( E^1, E^2, ..., \), represents length unit equals to two and three, respectively, which denoted by the power of edge. There exists a transition relation based on closure transition theorem of graph theory.

\[
E^k = E^{k-1} \circ E
\]  

(3)

where \( E^k \) represent the power \( k \) of edge \( E \).

**Theorem 1.** There are some connections from node \( v_i \) to \( v_j \) in graph \( G \), where length unit of a edge equals to \( k \) between two nodes, \( E^k \), then connection relation is denoted by \( v_iE^k v_j \) and \( E^* = E^1 \cup E^2 \cup E^3 \cup ... \), where \( E^* \) is an union set for \( E^1, E^2, E^3 \), ... .

Let adjacent matrix, \( M = [v_i, i=1,...,m; j=1,...,n] \) represent the edge \( E \) between node \( vi \) to \( vj \) in graph \( G \), \( M^* \) represents the edge \( E^* \). From Theorem 1, we have

\[
M^* = M^1 \circ M^2 \circ M^3 \circ ... \circ M^k
\]  

(4)

where \( M^* \) is the reachability matrix of a directed graph, \( M^k = M \times M \), \( M^i = M^i \times M \), etc. Next, let \( N(i,j) \) be a counting matrix for calculating the number of connection edge between node \( vi \) to \( vj \) whose length unit equals to one.

\[
N(i,j)=\begin{cases} 
\eta_i, & \text{if } (v_i,v_j) \notin E \\
1, & \text{if } (v_i,v_j) \in E 
\end{cases}
\]  

(5)

Similarly, \( N_2(i,j), N_3(i,j) \), its length unit is equal to two and three. Similar to Eq.(3), the elements of counting matrix can be derived by the cross product of two adjacent counting matrix. For example, \( N_1 \) is a mp matrix whose element is defined by Eq. (5), \( N_2 \) is a pxn matrix, then matrix \( N_3 \) is expressed by

\[
N_3(i,j)=\sum_{k=1}^{n}N_1(i,k)N_2(k,j)
\]  

(6)

According to induction rule, the counting matrix whose length unit equals to \( k \) is derived by

\[
N^k = N \times N^{k-1} \quad \forall k \geq 2.
\]  

(7)

where \( N^k \) represents a counting matrix whose path length is equal to \( k \). In the following, we define a matrix \( NP \) to stand for the total number of paths between node \( vi \) to \( vj \), where path length is equal to \( k \).

\[
NP(i,j) = \sum_{k=1}^{n}N^k(i,j)
\]  

(8)

**Step 3: Reconstruction of attack paths**

To search the global optimal solution of ACO, two search strategies are given to reconstruct the possible attack paths, when ants find their way back to nest. (i) Trace by the higher intensity of the pheromone over a trail, however, this strategy might lead to algorithm converges to a local optimal solution, (ii) Reinforce
ant’s sight ability for direction searching by examining trail intensity of pheromone. This strategy make ants search more flexible than that of the former. According to [8], each ant in the colony carries out a complete search on the graph abided by the probability density function as

$$p_{ij}(t) = \frac{\tau_{ij}(t)^{\alpha} \eta_{ij}(t)^{\beta}}{\sum_{j \in \text{neighbor}} \tau_{ij}(t)^{\alpha} \eta_{ij}(t)^{\beta}}$$

(9)

where $p_{ij}(t)$ determine the probability where an ant chooses a path from node $i$ to node $j$. $\tau_{ij}(t)$ be the intensity of pheromone trail between router $i$ and router $j$ at time $t$, $\eta_{ij}(t)$ means a heuristic value denotes the number of routing packets passing through between router $i$ and router $j$ at time $t$, $\alpha$ is the weighting factor of pheromone, $\beta$ is the weighting factor of visibility. Ant colony updates the probability density function of feasible attack paths and chooses the right one. The path search of adjacent routers for each ant (local update rule) is given by

$$\tau_{ij}(t+1) = (1- \rho) \times \tau_{ij}(t) + \rho \Delta \tau_{ij}$$

(10)

where $\rho$ represents the evaporation or decay rate of pheromone, residing in $[0,1]$, higher value hints that pheromone fast-decay, $\tau_{ij}$ is the initial value of pheromone. The intensity of pheromone path can be revised after all the ants select their route from the victim to an attack source. The purpose of path update is to avoid ants selecting on the same trail which might be a local optimal solution.

Once ant colony completes one generation, an overall update of pheromone intensity (global update rule) process is preceded, the intensity of pheromone on each node will be recalculated by

$$\tau_{ij}(t+1) = (1- \rho) \times \tau_{ij}(t) + \rho \Delta \tau_{ij}$$

(11)

$$\Delta \tau_{ij}(t) = \begin{cases} C & \text{if } k\text{th route is the optimal path} \\ 0 & \text{otherwise} \end{cases}$$

where $C$ is a constant, $L_q$ is the number of nodes on the optimal path. Global search of feasible paths is designed to that will increase the update speed of pheromone and convergence time of ACO. The difference between local update rule and global update rule is (i) $\tau_{ij}(t)$ update in global update rule only alter pheromone intensity of the optimal path that amplifies the bias between attack path and ordinary path. But $\tau_{ij}(t)$ update in local update rule is used for exploring other possible paths, so local update rule will revise the pheromone intensity of any paths where ant passes through it. Finally, ant traceback process iterates until the tour reaches all ants choose the same path or the preset cycles. Then, the proposed scheme find the optimal attack path, established by colony traveled based on ant-density rule. In summary, the whole process of path reconstruction is stated as illustrated Fig.2.

**Step 1:** Initialize
Construct a route graph based on the topology
For $t=1$ to $k$
 initialize pheromone $\tau_{0}$ for node(i)
 Lay h ants on the starting node
 For $q=1$ to $m$
 Reset the starting node
 **Step 2:** Traceback Process
 For $q=1$ to $m$
 If ant(q) not arrived the edge node (victim)
 Move to the neighbor node j and update the probability $Pij(t)$
 Add the node j into qth route solution
 **Step 3:** Local pheromone update
 for $i,j=1$ to $k$
 if route($i,j$) in the qth Route solution
 Set $\tau_{ij}(t+1) = (1-\rho) \times \tau_{ij}(t) + \rho \Delta \tau_{ij}$(t)
 else
 Set $\tau_{ij}(t+1) = (1-\rho) \times \tau_{ij}(t) + \rho \Delta \tau_{ij}$(t)
 **Step 4:** Global pheromone update
 Compute the most possible route solution and $L_q$
 If node(i) is in the most possible route solution
 Set $\tau_{ij}(t+1) = (1-\rho) \times \tau_{ij}(t) + \rho \frac{C}{L_q}$
 If not satisfied the terminate conditions (200 iterations)
 Empty all the route solution
 For $q=1$ to $m$
 Swap the starting node into qth route solution
 Back to Step 2
 Else output the most possible route solution

**Figure 2.** The process of path reconstruction

**Step 4:** Validation of Spoofed IP source
If bot herders want to masquerade or hide their attack location using spoofed IP, then true origin might be lost. Thus, attack scenarios with spoofed source addresses which need to be simulated to specify the probability successful to detect a fake IP attack and verify whether ACO algorithm can discover the correct attack paths or not. Based on the above statement of attack scenarios, we have to answer the question that how many routing packets required to detect a spoofed IP attack. Inspired by PPM, we suppose the probability successful to detect a fake IP attack at routers is $q$ and an attack path is $d$ hops long and the furthest router in this path is $R$. Let $X$ be the first time to detect a spoofed IP attack from $R$. Obviously, $X$ abides by the geometric distribution. The probability of receiving the some routing packets to discover a fake IP attack incurred via successive packets from $R$ is $q(1-q)^{d-1}$, then we have its expectation value

$$E(X) = \frac{1}{q(1-q)^{d-1}}$$

(12)

Let $Y$ be the number of packets required to detect a spoofed IP attacks. Suppose the cost is a specific function of routing distant
Then the probability of receiving $Y$ routing packets to ensure a spoofed IP attack incurred via successive packets from $R$ is

$$E(Y) = \frac{f(d)}{q(1-q)^{d-1}}$$ (13)

For example, assumed that analysis cost $c$ is the natural logarithm function of routing distant $d$, then we have

$$E(Y) = \frac{\ln(d)}{q(1-q)^{d-1}}$$ (14)

### 3.3 Dark IP Map

Upon successfully tracked back to attack source, defender needs a GIS to map zombie IPs in the real world. In our work, dark IPs will be located in a Google map and the attack paths marking with confidence degree be connected according to the infection sequences. This model can assist the defender to effectively monitor the infected hosts by reading information of dark IP, such as nation, bot type and its attack signatures.

### IV. TESTING AND VALIDATION

Our works simulated a service network in U.S.A to discover the potential threats caused by botnet attacks

**Step.1: Build up a network topology**

By applying Google map, network topology is established as shown in Fig.3 where there are 32 service nodes (0~31).

**Step 2: Decide the number of paths between two nodes**

The end nodes 0, 1 and 2 represents attack sources, end nodes 29, 30 and 31 are victims. The matrix $NP$, the amount of path number between attack source $v_i$ to victim $v_j$ with distant length 6 can be derived by Eq.(3)~(9) as Table 1.

**Table 1 adjacent matrix**

<table>
<thead>
<tr>
<th></th>
<th>N0</th>
<th>N1</th>
<th>N2</th>
<th>N3</th>
<th>...</th>
<th>N29</th>
<th>N30</th>
<th>N31</th>
</tr>
</thead>
<tbody>
<tr>
<td>N0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>N1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>N2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>7</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>N3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>7</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

**Step 3.2 Traceback to attack paths**

The routing information derived by Step 3.1, is input as initial values and required dataset of ACO. Then ants travel around all traits based on the local and global pheromone update rules using Eqs.(9)~(11). Consequently, there are 16 possible attack paths, where minimum support degree is greater than 10% as Fig.4.

**Table 2. Records of passing nodes on attack path**

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Attack source</th>
<th>Victim</th>
<th>Nodes on attack path</th>
</tr>
</thead>
<tbody>
<tr>
<td>N0</td>
<td>N29</td>
<td>0,9,15,22,28,25,29</td>
<td></td>
</tr>
<tr>
<td>N0</td>
<td>N29</td>
<td>0,4,9,10,9,14,21,20,27,29</td>
<td></td>
</tr>
<tr>
<td>N0</td>
<td>N29</td>
<td>0,4,9,14,21,25,29,27,29</td>
<td></td>
</tr>
<tr>
<td>S3.1</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>N0</td>
<td>N29</td>
<td>0,3,4,9,15,22,28,29</td>
<td></td>
</tr>
<tr>
<td>N0</td>
<td>N29</td>
<td>0,4,9,14,21,25,29</td>
<td></td>
</tr>
<tr>
<td>N0</td>
<td>N29</td>
<td>0,4,9,14,21,20,27,29</td>
<td></td>
</tr>
<tr>
<td>N0</td>
<td>N29</td>
<td>0,4,9,15,21,25,29</td>
<td></td>
</tr>
</tbody>
</table>

**Step.3 Reconstruction of attack paths**

To search a global optimal solution, two search strategies are given to reconstruct the possible attack paths, when ants find their way back to nest. (i) to follow by higher the intensity of pheromone over a trail, however, this strategy might lead to the result that solution might converge to a local optimal solution, (ii) add a sight ability to an ant that it can react to direction searching by online judging trail intensity of pheromone. This strategy let ant’s behavior become more flexible than that of the former.

**Step 3.1: Attack to a victim**

We conduct 200 test runs of random attacks using Monte Carlo simulation to collect routing information. For example, the attack from node 0 to node 29 is executed via 30 ants to trace back to their nest, the complete tour routings are recorded in Table 2.

**Figure 3. network topology**

**Figure 4 Simulation of attack path reconstruction**
Good modeling practice requires that the developer evaluate of the support and confidence in the model via assessing the uncertainties associated with the outcome of model itself. Then support and confidence degree of each path is evaluated as the fourth and the fifth column of Table 3 using Eqs. (1)–(2), when the minimal support degree is set to 10%. As a result, top 2 of support and confidence degree of possible attack paths are drawn in red line in Fig. 5 and attack paths with maximum probability in NS2 is shown as Fig. 6.

<table>
<thead>
<tr>
<th>Attack source</th>
<th>Attack paths</th>
<th>Support degree</th>
<th>Confidence degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>N29 N0→N4→N9→N15→N21→N25→N29</td>
<td>13.3%</td>
<td>19.05%</td>
<td></td>
</tr>
<tr>
<td>N29 N0→N4→N9→N15→N22→N28→N29</td>
<td>3.3%</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>N29 N0→N4→N9→N15→N22→N25→N29</td>
<td>56.7%</td>
<td>80.95%</td>
<td></td>
</tr>
<tr>
<td>N29 N0→N3→N4→N9→N15→N21→N25→N28→N29</td>
<td>6.7%</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>N29 N0→N4→N9→N14→N21→N25→N29</td>
<td>6.7%</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>N29 N0→N4→N9→N14→N21→N22→N25→N29</td>
<td>6.7%</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>N29 N0→N4→N9→N14→N20→N21→N25→N29</td>
<td>3.3%</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>N29 N0→N4→N9→N15→N21→N25→N28→N29</td>
<td>3.3%</td>
<td>0%</td>
<td></td>
</tr>
</tbody>
</table>

Once successfully tracked back to attack source, defender needs a GIS to map zombie IPs in the real world. In our work, dark IPs are located and displayed in a Google map-based GIS and connect the attack paths marking with confidence degree according to the infection sequence, as shown Fig.7. Dark IP map can facilitate to effectively monitor the infected hosts by examining dark IPs’ information, including IP’s nationality, bot type and its attack signatures.

Case I: Only Partial Routing Information is provided
Suppose partial ISPs are not willing to offer the complete routing information. These nodes, lacking of necessary touring information, are called grey nodes. Then pheromones of grey nodes set to zero. Repeat the Steps 3.1–3.2, experiment results show that ants will select the right attack paths as a tour, if grey nodes are in the attack paths, for example node 9,15 and 25. But when ants also find other alternative nodes to construct an attack path, then searching path may be altered, such as two alternative pairs, (3,4) and (21, 22) in Fig.7. In this situation, node 3 may be shifted to alternative node 4.

Case II: Preset Nodes in Attack Paths
It assumed that part of network nodes has been judged in advance as the preset nodes in attack paths. For example, preset nodes 13, 26 and 28 are forced to be selected into simulation case. Repeat the Step 3.1, defender finds that node 13 is far away the original attack path, and then new attack path will be generated. However, nodes 26 and 28 are close to the attack path, the results have almost the same attack paths as those of the above example, as shown in Fig.8.
Step 4: Validation of Spoofed IP Source

Let node 2 be a spoofed IP as shown in Fig. 5, and then reset the routing information to zero. The goal of deception test is to check whether our model could resist the spoofed IP attack. After executed the simulations and observed the outcomes, the results showed that partial ants searched the false paths in the beginning; however, later most ants would not attract by the spoof IP and come back to the correct paths progressively after five iterations. The search behavior of colony and resistance capability to the spoofed IP attacks is shown as Fig. 9.

The diagram (a)–(d) in Fig. 9 stands for simulation results of capability against the spoofed ID attack, respectively. We conducted and executed 100 times of algorithm with 20 iterations and the maximum population of colony is set to 30. Figure (a) reveals the relationship between iteration and the accuracy of searching the attack path. It’s approximate above 5 iterations that ants can regularly discover the correct path. In figure (b), it shows that the relationship between iteration and the percentage of ants on the best path. Over 50% ants cleverly found the best path, when the iterations were approximately over 9. From figure (c), we recognized that pheromone of best path would not evaporate and maintain a steady value as ants continually pass through. In the following, we examine the relationship between iteration process and the converge ratio of ants on the best path. In Figure (d), the threshold is set as 40%. When over 40% ants have gathered on the best path, it indicated that our algorithm has converged during these iterations. From figure (d), we observed that the convergent results had attained during iteration 3 to 8. In summary, most ants can resist the spoofed IP attacks via path research process after over 9 iterations in our case.
V. DISCUSSION

Once finished the traceback processes, sensitivity analysis is further investigated for realizing how the variation in the output of our model relating to the control variables of path searching of ACO. The sensitivity analysis is an essential step of quality assurance in model development process. In ACO model, three control variables need be explored - \(\alpha\), the weighting factor of pheromone; \(\beta\), the weighting factor of visibility and \(\rho\), the evaporation or decay rate of pheromone, respectively. Sensitivity analysis can play what-if analysis exploring the impact of varying three input parameters. Investigate these three significant parameters' effect on the output of attack paths. The selection of two weighting factor \(\alpha\) and \(\beta\) may affect the search results of attack paths. It might increase the convergence speed resulting in falling into a local optimal solution, if \(\alpha\)'s value too high (over the threshold). Since the higher \(\alpha\)’s value implies that ants would have a tendency to fast convergence in comparison with lower \(\alpha\)’s value. After executed 200 test runs, accumulate and average the path information on attack paths as illustrated in Table 4. From Table 4, two parameters affect the output results of attack paths. The total number of routing packets collected on attack paths will be over 160 as the threshold that covers about 80% paths. The total number of routing packets collected on attack paths as illustrated in Table 4.

In the following, defender may concern that the decay rate of pheromone how to affect the speed of convergence. From Table 5, we realized that it will have an effect on the proceeding speed of packets and lowering the average number of routing packets handled on the attack paths, when \(\rho\) is too low for intensity of pheromone trail \(\tau_{ij}(t)\). Consequently, ant would keep slowing down to search a new path. In contrast, it might increase the influence of convergence speed to gain a local optimal solution, if \(\rho\) is high. From the trade-off analysis of Table 5, we observe that the best-fit value of decay rate of pheromone is 0.5 in our case. From Table 6, satisfactory results on average number of path information colleted about 80%) would be attained, when the iteration numbers (generations) is over ten.

VI. CONCLUSION

A new IP traceback analysis model based on an extended ACO algorithm is proposed for incomplete flow information being provided. To avoid attacks from IP spoofing, our scheme can assist defender to make an appropriate decision on discovering possible attack paths with support and confidence. In addition, sensitivity analysis of ACO control variables be examined to increase confidence by verifying the effect of the convergence tendency; Computation time is also examined. The numerical examples presented herein show that the proposed approach effectively discovers the most possible attack path and suggest the right location of deployment nodes for promoting the probability of trapping attack sources.

ACKNOWLEDGES

This work was supported partly by TWISC@ NCKU, National Science Council under the Grants No: NSC98-2219-E-006-001 and NSC 98-2219-H-168-001.

REFERENCES


<table>
<thead>
<tr>
<th>Table 4. Effect of parameter (\alpha) and (\beta)</th>
<th>(\alpha)</th>
<th>(\beta)</th>
<th>(\alpha=0.1)</th>
<th>(\alpha=0.3)</th>
<th>(\alpha=0.5)</th>
<th>(\alpha=0.7)</th>
<th>(\alpha=0.9)</th>
<th>(\alpha=1.2)</th>
<th>(\alpha=1.5)</th>
<th>(\alpha=1.8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\beta=0.1)</td>
<td>110.8</td>
<td>111.3</td>
<td>114.0</td>
<td>112.8</td>
<td>114.8</td>
<td>114.9</td>
<td>116.6</td>
<td>108.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\beta=0.3)</td>
<td>109.6</td>
<td>111.5</td>
<td>122.3</td>
<td>125.2</td>
<td>129.8</td>
<td>130.2</td>
<td>131.2</td>
<td>129.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\beta=0.5)</td>
<td>121.2</td>
<td>133.3</td>
<td>134.0</td>
<td>134.6</td>
<td>135.3</td>
<td>135.8</td>
<td>134.2</td>
<td>131.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\beta=0.7)</td>
<td>137.5</td>
<td>139.2</td>
<td>141.9</td>
<td>143.1</td>
<td>146.9</td>
<td>152.7</td>
<td>145.7</td>
<td>136.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\beta=0.9)</td>
<td>141.3</td>
<td>148.6</td>
<td>149.7</td>
<td>150.8</td>
<td>151.1</td>
<td>156.3</td>
<td>150.5</td>
<td>145.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\beta=1.2)</td>
<td>148.9</td>
<td>153.7</td>
<td>155.8</td>
<td>157.7</td>
<td>157.8</td>
<td>157.7</td>
<td>156.3</td>
<td>152.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\beta=1.5)</td>
<td>153.8</td>
<td>158.3</td>
<td>158.3</td>
<td>159.7</td>
<td>160.0</td>
<td>161.8</td>
<td>162.9</td>
<td>154.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\beta=1.8)</td>
<td>159.7</td>
<td>159.5</td>
<td>159.3</td>
<td>162.5</td>
<td>164.8</td>
<td>162.1</td>
<td>158.6</td>
<td>157.4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 5 Effect of decay rate factor, (\rho)</th>
<th>(\rho)</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average number of routing packets handled</td>
<td>155.5</td>
<td>158.4</td>
<td>161.2</td>
<td>162.7</td>
<td>164.7</td>
<td></td>
</tr>
<tr>
<td>Average number of routing packets handled</td>
<td>0.6</td>
<td>0.7</td>
<td>0.8</td>
<td>0.9</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Table 6 Effect of iteration number | \# of generation | 5 | 10 | 20 | 40 | 100 |
|---|---|---|---|---|---|
| Average number of path information collected | 158.2 | 163 | 164.5 | 164.7 | 164.8 |
| Percentage of path information collected | 79.1% | 81.5% | 82.3% | 82.4% | 82.4% |