A new approach for solving the IP traceback problem for Web security

Ping Wang, Hui-Tang Lin, Tzu Chia Wang, Pu-Tsun Kuo

Department of Information Management, Kun Shan University, Taiwan,
pingwang@mail.ksu.edu.tw bluesky.kuo@gmail.com

Institute of Computer and Communication Engineering National Cheng Kung University, Taiwan,
htlin@mail.ncku.edu.tw tzuchia@crypto.ee.ncku.edu.tw

Abstract

To effectively counter DDOS attacks from bot herder, Web defenders have developed some approaches to successfully detect and traceback the command and control (C&C) server of botnet for Web security. Yet, available botnet detection schemes assume that all of the ISPs cooperate in providing the routing information required reconstructing the attack path. However, in most practical cases, this assumption cannot be guaranteed. Accordingly, the present study proposes a new approach for solving the IP traceback problem in botnets by means of ant colony optimization (ACO) algorithm. In the proposed approach, ant-inspired collective intelligence is used to predict the most possible attack path based on a consideration of both the support degree and the confidence degree. The validation of model uses NS2 (Network Simulator, version2) compiled by dark IP map, to simulate the scenario of spoofed IP attacks. Finally, the robustness of the proposed scheme toward spoofed IP attacks in investigated. Overall, the results confirm that the proposed method provides an effective means of reconstructing the path between the attacker and the victim in the absence of full routing information.

Keywords: Web security, attack path, ant colony optimization

1. Introduction

The botnet refers to a group of computers, manipulated by herder via malicious codes inserted into Web page. After malware has been successfully installed in a victim via network connections, it becomes a compromised host and will accept the remote commands from a malicious user, referred to as a bot herder. The compromised systems are referred to as “zombies” or “bots”, and collectively form a botnet. While the motivation of most herders in establishing a botnet is that of financial gain, or simply attracting recognition amongst his or her peers, the botnet may also be hired out to third parties to conduct a range of nefarious activities, including sending out spam messages, distributing viruses, installing spyware, and so on.

According to survey results released by the CSI / FBI in the USA [4], the average losses incurred per respondent as a result of security incidents amounted to $234,244 in 2009. Moreover, 22% of the respondents reported the theft of personal information as a result of downloading malicious bots. 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 [5,6,8]. According to statistic reports of TrendLabs [16], bots infected 340 million hosts all over the world from Jan to Oct. in 2008; there were over one million infected hosts per month. Web defenders have developed some approaches to detect the attack origin of botnet using IP traceback techniques via analyzing the network flow information of firewall, honeypot and IDS. [6] IRC botnet forms a centralized architecture which sends the commands to a C&C server using http, ssh and telnet protocol for remote controlling zombies. It could be turned down if the defender captured the C&C server [5]. IRC botnets have a severe weak point because they can be disabled by a shutdown command to disconnect with...
C&C control server. Based on this weakness, a distributed-based P2P botnet had recently emerged and became a major threat to Web security in recent years [7, 14]. The features of advance P2P bot are that some bots adopts an encrypted channel between C&C server and victim, such as SSH or WASTE, such that network sniffer can not effectively crack the control messages in C&C channels. In further, three important discoveries in Peacom bots (P2P) in 2007 so that defenders are hard to track them: (i) anti-detection: herder may design countermeasures against defender (ii) self-destroyed or hided when defender successfully traced back to C&C server or cracked the communication channels of botnet (iii) communicate through encrypted channel [11,13-14].

Though available IP traceback approaches are capable of detecting the zombies via accumulating attacking routing packets, they are all based on an idealistic assumption –the full cooperation of all the servers between the victim and the C&C server in providing the routing information required to reconstruct the attack path. Therefore the constraint, require cooperation between the victim and the C&C server, must be relaxed. A new IP traceback scheme based on ant colony optimization (ACO) approach extends our work [15] to discover the most possible attack path based on consideration of both the support degree and the confidence degree for locating the true IP of C&C server, even if the path reconstruction process is in an incomplete routing environment.

The rest of the paper is organized as follows. Section II provides the background knowledge of the IP traceback techniques; Section III describes the ACO optimization scheme proposed in this study for solving the IP traceback problem in botnet. Simulations on the results are presented in Section VI. Section V performs a sensitivity analysis. Conclusions and future studies follow in Section VI.

2. Related works

From 1990s, a number of IP traceback techniques [1, 10, 12] have been developed and applied in DDOS network attacks. IP traceback techniques can be simply classified as two main categories based on attack scenarios: (i) passive (post mortem) traceback (ii) pro-active (on-going) traceback. Notably, passive traceback 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 invariably assume the full cooperation of all the servers between the victim and the C&C server in providing the routing information required to reconstruct the attack path.

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 and overlay network. 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 may be erased or overlapped, because extra routing information being stored.(ii) Require cooperation between ISPs.

Although existing IP traceback methods can be employed to botnet detection and classification purposes, they invariably assume the full cooperation of all the servers between the victim and the C&C server in providing the routing information required to reconstruct the attack path. In practice, some service providers may be unwilling to provide this information. As a result, it is necessary to reconstruct the path given only a limited knowledge of the routing information. 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. Accordingly, the present study proposes a new approach for solving the IP traceback problem in botnets by means of ACO algorithm as described as the following section.

3. An analysis model of attack paths in botnets

This section is to develop an analysis model for discovering the possible attack paths of botnet, estimating the support and confidence of each attack path, and appropriately pointing out the most possible attack path.
3.1 Basic idea

Basically, the attack path reconstruction process involves interrogating the routing packets received at the victim(s) in order to find the immediate upstream node(s), and then systematically repeating the interrogation process at each intermediate upstream node until the attack source is reached (see Fig. 1). In other words, the path reconstruction problem can be regarded as a special form of graph optimization problem. In using an ACO scheme to solve the IP traceback problem in botnets, the ants lay a pheromone trail along the route they select between the victim (the food source) and the attacker (the nest) (e.g., paths $P_1 - P_2 - P_3$ and $P_4 - P_5 - P_6 - P_3$, in Fig. 1), and the relative probability of each path being the actual attack path is given by the intensity of the pheromone along the corresponding trail. As in nature, the isolated ants in the ACO scheme move essentially at random. However, upon encountering a previously laid trail, the ants decide with a high probability to trace it. As a result, the pheromone intensity of this path progressively increases, and thus the likelihood of the path representing the actual attack path also increases.

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

Assume that a group of ants, $A_q$, $q=1,...,m$, are assigned to find food and travel at a random walk. Furthermore, assume that $n$ ants traverse trail $P_i$ in finding their way back to the nest. The support degree of path $P_i$ ($\text{sup} P_i$) is therefore given as

$$\text{sup} P_i = \frac{n}{m}$$

(1)

Furthermore, let the confidence degree of path $P_i$ ($\text{conf} P_i$) be defined as the ratio of the number of ants which traverse path $P_i$ to the number of ants which traverse all possible attack paths with a support greater than the minimal support degree.

$$\text{conf}_{P_i} = \frac{\text{sup} P_i \geq \text{min} P}{\sum_{i=1}^{m} \text{sup} P_i \geq \text{min} P}$$

(2)

where $\text{min} P$ represents the minimal support degree. Thus, a high confidence degree indicates that the corresponding path has a higher probability of being the actual attack path. In the following, we are interested in two issues of reconstruction of attack path: (i) find all directed paths between two arbitrary (end) nodes, (ii) exploit portable paths via trait pheromone density in order to identify the most possible attack path.

3.2 Application of ACO to Solution of Botnet Detection Problem

Let network topology be a directed graph, $G=(V,E)$, where $V$ represent a set of nodes, $V=\{v_1, v_2, ..., v_n\}$, $V_s$ is a set of source nodes (i.e., attack sources), $V_d$ 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: Construct network topology
The present study focuses on the security management aspect of Web services. Accordingly, a service-oriented network topology is simulated and established for model analysis.

Step 2: Determine all possible paths between two network nodes

Basically, a path in a digraph comprises a sequence of edges leading from one vertex to another. In the IPTBK context, the attack path is therefore represented by a set of contiguous edges. According to Skvarcius and Robinson [15], the number of non-cyclic paths between any two network nodes can be determined as follows:

Let the length of an edge be represented by $E$, and let $E^2, E^3, \ldots$, denote edges with a power of two and three length units, respectively. In accordance with the closure transition theorem of graph theory, the following transition relation can therefore be defined:

$$E^k = E^{k-1} \circ E$$

where $E^k$ represent the power $k$ of edge $E$.

Theorem 1. Given some connections from node $v_i$ to $v_j$ in graph $G$, and assuming that the length unit of the edge between the two nodes is equal to $k$ (i.e., $E^k$), the connection relation between the two nodes is denoted by $v_i E^k v_j$, and $E^* = E^1 \cup E^2 \cup E^3 \cup \ldots$, where $E^*$ is the union set of $E^1, E^2, \ldots$. Let the adjacent matrix, $M = \{v_i | i = 1, \ldots, m; j = 1, \ldots, n\}$, represent the edge E between nodes $v_i$ and $v_j$ in graph $G$.

Furthermore, let $M^*$ represent edge $E^*$. It therefore follows that:

$$M^* = M^1 \lor M^2 \lor M^3 \lor \ldots \lor M^p$$

where $M^*$ is the reachability matrix of a directed graph, $M^1 = M \times M$, $M^2 = M^1 \times M$, etc$. v_{12} = 0$, it means that there exist no path between node $v_1$ and node $v_2$. In contrast, $v_{24} = 1$ a path connects between $v_2$ and $v_4$.

$$M^* = \begin{bmatrix}
1 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 \\
1 & 1 & 0 & 1
\end{bmatrix}$$

Let $N(i, j)$ be a counting matrix for indicating the number of connection edges between nodes $v_i$ and $v_j$ with a length unit equal to one, i.e.,

$$N(i, j) = [n_{ij}]_{m \times n}, \quad n_{ij} = \begin{cases}
0, & \text{if} (v_i, v_j) \notin E \\
1, & \text{if} (v_i, v_j) \in E
\end{cases}$$

Similarly, let matrices $N_2(i, j), N_3(i, j)$ be counting matrices indicating the number of connection edges between nodes $v_i$ and $v_j$ with length units equal to two and three, respectively. Note that similar to Eq.(4), the elements of a counting matrix are given by the cross product of the two adjacent counting matrices. For example, if $N_1$ is an $m \times p$ matrix whose elements are defined by Eq. (6), and $N_2$ is a $p \times n$ matrix, then matrix $N_3$ is expressed by

$$N_3(i, j) = \sum_{k=2}^{\infty} N_2(i, k) N_3(k, j)$$

According to the induction rule, the counting matrix whose length unit is equal to $k$, $N^k$, is derived as

$$N^1 = N$$

$$N^k = N \times N^{k-1} \quad \forall k \geq 2.$$
\[ N \rho(i, j) = \sum_{k=1}^{g} N^k(i, j) \]  

(9)

Step 3: Reconstruction of attack paths

In solving the IP traceback problem in botnets using ACO, each ant builds a tour (i.e., a feasible path between the victim and the attacker) by repeatedly applying the state transition rule.

Step 3.1 State transition rule

In the ACO algorithm, the path searching process is accomplished using a state transition rule comprising two policies, namely exploitation and biased exploration [9]. The exploitation policy (see Eq. 10(a)) chooses the arc with the greatest pheromone intensity and visibility, while the exploration policy (see Eq. 10(b)) is a random decision rule. Thus, an ant located at node \( i \) chooses the next node \( j \) in accordance with the following rule:

\[
j = \begin{cases} 
\arg \max_{j \in \text{tabu}} \left[ \tau^\alpha_{ij}(t) \right] \left[ \eta^\beta_{ij}(t) \right] & \text{if } q \leq q^a \\
S & \text{otherwise}
\end{cases} \tag{10}
\]

where \( q^a \) is a user-defined parameter which specifies the distribution ratio of the two policies, and \( q \) is a random number, \( 0 \leq q \leq 1 \). Here \( p_i(t) \) determine the probability where an ant chooses a path from node \( i \) to node \( j \) following the probability density function as Eq.(10c). \( \tau^\alpha_{ij}(t) \) is the pheromone intensity of trail between router \( i \) and router \( j \) at time \( t \), \( \eta^\beta_{ij}(t) \) is calculated as the number of routing packets between router \( i \) and router \( j \) between time \((t-1)\) and time \((t)\), and is used to simulate the visibility of the ants on arc \((i, j)\). \( \alpha \) 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.

While constructing its path, each ant increases the amount of pheromone on the traversed trail by applying a ‘local updating rule’. Once all of the ants have completed their tour, the intensity of the pheromone on the path is updated once again by means of a ‘global updating rule’.

Step 3.2 Local update rule

As described above, to search a global optimal solution, two search strategies of ACO are given to reconstruct all possible attack paths, when ants find their way back to nest. (i) exploitation: follow by the higher intensity of pheromone over a trail, however, this strategy might lead to the result that solution might converge to a local optimal, (ii) exploration: add a sight capability to each ant that it can react to direction searching by online judging trail intensity of pheromone using Eq.(10). This strategy let ant’s behavior become more flexible than that of the former. Based on strategy (ii), path search of adjacent routers for each ant (local update rule) is given by

\[
\tau^\alpha_{ij}(t+1) = (1-w) \times \tau^\alpha_{ij}(t) + w \Delta \tau^\alpha_{ij}(t) \tag{11}
\]

where \( w \) represents the evaporation or decay rate of local pheromone, residing in \([0,1]\), higher value hints that pheromone fast-decay, \( \tau_{ij}^\alpha \) is the initial value of \( \Delta \tau_{ij}(t) \) which can be rationally determined by ANT-quantity, Ant-density or ANT-cycle [2,18].
Step 3.3 Global update rule

Once all of the ants have completed their tour, the intensity of the pheromone on each node of the best path is updated once again by Eq. (12)

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

\[
\Delta \tau_{ij}(t) = \begin{cases} 
\frac{C}{L_k} & \text{if } k\text{th route is the best path} \\
0 & \text{otherwise}
\end{cases}
\]

where \( \rho \) represents the decay rate of global pheromone \( 0 < \rho < 1 \), \( C \) is a constant and \( L_k \) is the number of nodes on the optimal path. A higher value of \( C \) results in a more rapid convergence time. Conversely, a longer distance between C&C server and victim leads to a slower converge.

Step 4: Against Spoofed IP attacks

If bot herders want to masquerade by hiding their attack location using fake IP, then true origin might be lost. Thus, attack scenarios with fake source addresses which need to be simulated and verified whether ACO algorithm can discover the correct attack paths or not. The validation of our model is applied using Network Simulator 2 (NS2) tool. 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 attacks. Inspired by PPM [12], 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

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

Let \( Y \) be the number of packets required to detect a spoofed IP attacks. Suppose the analysis cost \( c \) is a specific function of routing distant \( d \), then the probability of receiving \( Y \) routing packets to ensure a spoofed IP attack incurred via successive packets from \( R \) is

\[
E(Y) = \frac{f(d)}{q(1-q)^{d-1}}
\]

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

\[
E(Y) = \frac{\ln(d)}{q(1-q)^{d-1}}
\]

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 Web defender or network administrator to effectively observe and locate the infected hosts by reading information of dark IP, such as nation, bot type and its attack signatures.

4. Testing and Validation

A series of NS2 simulations was performed to investigate the potential threats to a typical Web service network by botnet attacks and to explore the effectiveness of the ACO algorithm via the IP
traceback technique. The simulations were performed using a PC with an Intel Dual core CPU 3.0G, DDR2 1G of RAM and the MS Windows XP operating system.

Step 1: Construct network topology
To give an explanation of our scheme, the simulations considered a Web service network (32 nodes) with the topology referred to real-world network connections in U.S.A, as shown in Fig. 2.

![Figure 2. Network topology](image)

Step 2: Determine number of paths between two nodes
End nodes 0, 1 and 2 were assumed to represent attack sources, while end nodes 29, 30 and 31 were assumed to be victims. The NP matrix, i.e., the matrix giving the number of paths between the attack source $v_i$ and the victim $v_j$, was derived from Eqs. (3)–(9). The matrix is shown in Table 1 where the network topology is shown in Fig. 2.

<table>
<thead>
<tr>
<th></th>
<th>N_0</th>
<th>N_1</th>
<th>N_2</th>
<th>N_3</th>
<th>...</th>
<th>N_29</th>
<th>N_30</th>
<th>N_31</th>
</tr>
</thead>
<tbody>
<tr>
<td>N_0</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>N_1</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>N_2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>N_3</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>
<tr>
<td>N_4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>N_5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
<td>10</td>
<td>6</td>
<td>11</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>N_31</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Step 3: Reconstruction of attack paths
The process of attack paths were conducted by the following two sub-steps:

Step 3.1: Attack on victim
200 random attacks were simulated using a Monte Carlo model and 30 ants in order to generate routing information. Table 2 presents some illustrative results for the attack paths generated between node 0 and node 29.
Table 2. Records of passing nodes on attack path

<table>
<thead>
<tr>
<th>Attack source</th>
<th>Victim</th>
<th>Nodes on attack path</th>
</tr>
</thead>
<tbody>
<tr>
<td>N0</td>
<td>N29</td>
<td>0,4,9,15,22,28,25,29</td>
</tr>
<tr>
<td>N0</td>
<td>N29</td>
<td>0,4,5,10,9,14,21,20,27,29</td>
</tr>
<tr>
<td>N0</td>
<td>N29</td>
<td>0,4,9,14,21,25,20,27,29</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N0</td>
<td>N29</td>
<td>0,3,4,9,15,22,28,29</td>
</tr>
<tr>
<td>N0</td>
<td>N29</td>
<td>0,4,9,14,21,25,29</td>
</tr>
<tr>
<td>N0</td>
<td>N29</td>
<td>0,4,9,15,21,25,29</td>
</tr>
</tbody>
</table>

Step 3.2: Traceback to attack paths

The routing information generated in Step 3.1 was used as the input dataset. The ants performed tours using the local and global pheromone updating rules given in Eqs. (10)–(12). Table 3 presents a subset of the 15 possible attack paths identified by the ACO model between Node 0 and Node 29.

Table 3. Possible attack paths

<table>
<thead>
<tr>
<th>Attack source</th>
<th>Victim</th>
<th>Possible Attack paths</th>
<th>Path No</th>
</tr>
</thead>
<tbody>
<tr>
<td>N0</td>
<td>N29</td>
<td>N0→N4→N9→N15→N22→N25→N28→N29</td>
<td>Path1</td>
</tr>
<tr>
<td>N0</td>
<td>N29</td>
<td>N0→N4→N9→N15→N21→N25→N29</td>
<td>Path2</td>
</tr>
<tr>
<td>N0</td>
<td>N29</td>
<td>N0→N4→N9→N15→N21→N22→N28→N29</td>
<td>Path3</td>
</tr>
<tr>
<td>N0</td>
<td>N29</td>
<td>N0→N4→N9→N15→N22→N28→N29</td>
<td>Path4</td>
</tr>
<tr>
<td>N0</td>
<td>N29</td>
<td>N0→N4→N9→N15→N22→N25→N29</td>
<td>Path5</td>
</tr>
<tr>
<td>N0</td>
<td>N29</td>
<td>N0→N3→N4→N9→N15→N25→N28→N29</td>
<td>Path6</td>
</tr>
<tr>
<td>N0</td>
<td>N29</td>
<td>N0→N4→N9→N15→N21→N25→N29</td>
<td>Path7</td>
</tr>
<tr>
<td>N0</td>
<td>N29</td>
<td>N0→N4→N9→N15→N21→N22→N25→N29</td>
<td>Path8</td>
</tr>
<tr>
<td>N0</td>
<td>N29</td>
<td>N0→N4→N9→N14→N21→N25→N29</td>
<td>Path9</td>
</tr>
<tr>
<td>N0</td>
<td>N29</td>
<td>N0→N4→N9→N14→N21→N22→N25→N29</td>
<td>Path10</td>
</tr>
<tr>
<td>N0</td>
<td>N29</td>
<td>N0→N4→N9→N14→N20→N21→N25→N29</td>
<td>Path11</td>
</tr>
<tr>
<td>N0</td>
<td>N29</td>
<td>N0→N4→N9→N15→N21→N22→N25→N28→N29</td>
<td>Path12</td>
</tr>
<tr>
<td>N0</td>
<td>N29</td>
<td>N0→N4→N9→N15→N21→N22→N25→N28→N29</td>
<td>Path13</td>
</tr>
<tr>
<td>N0</td>
<td>N29</td>
<td>N0→N4→N9→N14→N15→N22→N25→N28→N29</td>
<td>Path14</td>
</tr>
<tr>
<td>N0</td>
<td>N29</td>
<td>N0→N3→N4→N9→N15→N22→N28→N29</td>
<td>Path15</td>
</tr>
</tbody>
</table>

Note that the fourth and fifth columns of the Table 4 show the support degree and confidence degree of each path, as calculated using Eqs. (1)–(2) with a minimal support degree of 10%.

Table 4. Attack paths with corresponding degrees of support and confidence

<table>
<thead>
<tr>
<th>Attack source</th>
<th>Victim</th>
<th>Possible Attack paths</th>
<th>Support degree</th>
<th>Confidence degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>N0</td>
<td>N29</td>
<td>N0→N4→N9→N15→N21→N25→N28→N29</td>
<td>13.3%</td>
<td>19.05%</td>
</tr>
<tr>
<td>N0</td>
<td>N29</td>
<td>N0→N4→N9→N15→N31→N22→N28→N29</td>
<td>3.3%</td>
<td>0%</td>
</tr>
<tr>
<td>N0</td>
<td>N29</td>
<td>N0→N4→N9→N15→N32→N25→N29</td>
<td>56.7%</td>
<td>80.95%</td>
</tr>
<tr>
<td>N0</td>
<td>N29</td>
<td>N0→N4→N9→N15→N22→N28→N29</td>
<td>6.7%</td>
<td>0%</td>
</tr>
<tr>
<td>N0</td>
<td>N29</td>
<td>N0→N4→N9→N14→N15→N25→N29</td>
<td>6.7%</td>
<td>0%</td>
</tr>
<tr>
<td>N0</td>
<td>N29</td>
<td>N0→N4→N9→N14→N20→N21→N25→N29</td>
<td>3.3%</td>
<td>0%</td>
</tr>
<tr>
<td>N0</td>
<td>N29</td>
<td>N0→N4→N9→N15→N21→N22→N25→N28→N29</td>
<td>3.3%</td>
<td>0%</td>
</tr>
</tbody>
</table>

Figure 3 illustrates the number of ants which traverse each of the 15 possible attack paths in each iteration of the solution procedure. In other words, Fig. 3 confirms that path #5 (N0→N4→N0
\( \rightarrow N_{15} \rightarrow N_{22} \rightarrow N_{25} \rightarrow N_{29} \) is the most possible attack path where the path with the highest support and confidence degrees, i.e., this path is traversed by the largest number of ants.

![Figure 3. Simulation of attack path reconstruction](image)

In a real-world scenario, having successfully detected the attack source, the defender can use GIS technology to map the zombie IPs to the physical world. In our work, a dark IP map is located and displayed in a Google map-based GIS and connect the attack paths marking with confidence degree in accordance with the infection sequence as shown Figure 4. The resulting Dark IP map facilitates the effective monitoring of the infected hosts by examining various items of the IP information, including the IP nationality, the bot type and its attack signatures. Figure 4 shows the Dark IP map for the example considered above.

![Figure 4. Dark IP map showing attack path between Node 0 and Node 29](image)

For real-world IP traceback problems in botnet, some of the ISPs (denoted as ‘grey nodes’, for example, node 4, 9, 15 and node 22) are unwilling to provide the routing information required to reconstruct the attack path. To reflect this, Steps 3.1 and 3.2 were repeated with the pheromone intensity of the grey nodes set to zero. The results showed the ants also selected the right attack path as a tour (nodes 9, 15 and 25). However, when grey nodes had changed by alternative pair, the ants found
other alternative path. For example, an alternative pair, (3,4) in Fig.2, node 4 will be visited in the tour instead of an alternative node 3 in this situation.

Step 4: Spoofed IP attack

Let node 2 be a spoofed IP as shown in Fig.5, and then reset its routing information to zero. Accordingly, this step evaluates the robustness of the proposed algorithm toward spoofed IP attacks.

We conducted 100 times of algorithm with the maximum number is 200 iterations and population of colony is set to 30. After test runs executed, the results showed that partial ants searched the false paths in the beginning; however, later most ants would not attract by the fake 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.6.

![Figure 5. Attack paths derived by fake IP](image)

![Figure 6. Spoofed IP attacks](image)

(a) Percentage of correct path and iteration cycle  
(b) Percentage of the ants on the best route

The diagrams (a)–(b) in Fig.6 stand for simulation results of capability against the spoofed IP attack, respectively. Figure (a) reveals the relationship between iteration cycle 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 cycle and the percentage of ants walked on the best path. It is clear that the ants find the correct path and over 50% ants cleverly found the best path, when the iterations were approximately over 9. In summary, most ants can resist the spoofed IP attacks via path research process after over 9 iterations in our case.
5. Discussion

Once finished the traceback processes, sensitivity analysis is further investigated for realizing how the variation in the output of our model. 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. Two weighting factor $\alpha$ and $\beta$, affecting the search results of attack paths, are primary selected.

In the IP trackback process, good solutions of ACO model require high numbers of packets to converge on the attack path(s) that can help defender identify the most possible attack path [12]. In our work, the trait pheromone, a convergence metric, is indirectly represented by the total number of routing packets pass through each node on attack paths. After executed 200 test runs, we averaged the total number of packets pass through each node on attack path to evaluate the solution quality, as illustrated in Table 5.

From Table 5, two parameters affect the output results. The average number of routing packets converged on attack paths for each node is near 160, it hinted that the attack paths covered about 80% attack test runs, if $\alpha$ residues in [0.9, 1.2] and $\beta$ in [1.5, 1.8]. In the following, defender may concern that the decay rate of pheromone how to affect the speed of convergence. From Table 6, we realized that it will have an explicit effect on the proceeding speed of packets converged on the attack paths and lowering the average number of routing packets converged on the attack paths, when $\rho$ is too low for intensity of pheromone trail $\tau_{ij}(t)$. Consequently, the ants will keep slowing down to search a new path. In contrast, it might increase the convergence speed so as to gain a local optimal solution, if $\rho$ is too high. Using the trade-off analysis, we observed that the best-fit value of decay rate of pheromone is 0.5 in the Table 6 which cover over 80% routing packets converged on the attack paths. From Table 7, when the iteration numbers (generations) is over 10, satisfactory results would be attained.

<table>
<thead>
<tr>
<th>Table 5. Effect of parameter $\alpha$, $\beta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta$</td>
</tr>
<tr>
<td>$\beta=0.1$</td>
</tr>
<tr>
<td>$\beta=0.3$</td>
</tr>
<tr>
<td>$\beta=0.5$</td>
</tr>
<tr>
<td>$\beta=0.7$</td>
</tr>
<tr>
<td>$\beta=0.9$</td>
</tr>
<tr>
<td>$\beta=1.2$</td>
</tr>
<tr>
<td>$\beta=1.5$</td>
</tr>
<tr>
<td>$\beta=1.8$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 6. Effect of decay rate factor, $\rho$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho$</td>
</tr>
<tr>
<td>Average number of routing packets converged</td>
</tr>
<tr>
<td>$\rho$</td>
</tr>
<tr>
<td>Average number of routing packets converged</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 7. Effect of iteration number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iteration cycle</td>
</tr>
<tr>
<td>Average number of routing packets converged</td>
</tr>
<tr>
<td>Percentage of routing packets converged</td>
</tr>
</tbody>
</table>
6. Conclusion

A new IP traceback analysis model in botnets based on an extended ACO algorithm is proposed. To avoid attacks from spoofed IP attacks, our scheme can assist defender to discover all possible attack paths with support and confidence degree. In addition, sensitivity analysis of ACO control variables be examined to increase user’s confidence by verifying the solution quality on the iteration process; Computation time (iteration cycle) is examined and the number of packets required for path reconstruction as function of routing distance is also discussed. The numerical examples presented herein show that the proposed approach effectively discovers the most possible attack path and detects the attack origin or C&C servers of botnet.

Acknowledges
This work was supported partly by TWISC@ NCKU, National Science Council under the Grants No: NSC98-2219-E-006-001 and NSC 99-2219-E-168-001.

References