Consideration of a Countermeasure Model against Self-Evolving Botnets

Kouji Hirata[*], Koki Hongyo[*], Takanori Kudo[†], Yoshiaki Inoue[‡], and Tomotaka Kimura[§],
[*] Faculty of Engineering, Kansai University, Osaka 564-8680, Japan, Email: {hirata, k896955}@kansai-u.ac.jp
[†] Faculty of Science and Engineering, Setsunan University, Osaka 572-8508, Japan, Email: t-kudo@ele.setsunan.ac.jp
[‡] Graduate School of Engineering, Osaka University, Osaka 565-0871, Japan, Email: yoshiaki@comm.eng.osaka-u.ac.jp
[§] Faculty of Science and Engineering, Doshisha University, Kyoto 610-0321, Japan, Email: tomkimur@mail.doshisha.ac.jp

***Abstract*—The literature has suggested the appearance of self-evolving botnets, which autonomously discover vulnerabilities by performing machine learning with computing resources of zombie computers and evolve accordingly. The infectablity of the self-evolving botnets is too strong compared with conventional botnets. This paper introduces a countermeasure model against the self-evolving botnets. This model aims at preventing the self-evolving botnets from spreading by discovering vulnerabilities with computing resources of volunteer hosts before the self-evolving botnets discover them. Through simulation experiments based on a continuous-time Markov chain, we evaluate the performance of the countermeasure model.**

Keywords–Botnet; machine learning; epidemic model; continuous-time Markov chain; countermeasure.

I. Introduction

Recently, machine learning techniques, such as deep learning [1][2], have been widely used and achieved significant results in various research areas. In addition, some researchers have been proposed vulnerability discovery methods that discover bugs and vulnerabilities with static code analysis and machine learning techniques [7][8]. Of course, the main purpose of these methods is to protect software. However, these methods can be used for discovering unknown security holes and exploited for illegal attacks by malicious attackers.

To perform illegal attacks, malicious attackers often infect hosts with malware. A botnet is a set of hosts infected by the botnet malware [6]. The zombie computers are controlled by a malicious attacker and perform illegal attacks. In the past, there have been some botnets that consist of more than a million zombie computers. The authors in [4][5] have introduced a new concept named self-evolving botnets, based on these facts. The self-evolving botnets discover vulnerabilities by performing distributed machine learning with computing resources of zombie computers and evolves autonomously exploiting the discovered vulnerabilities. Accordingly, they infect other hosts and make themselves bigger. The authors in [4][5] have provided an epidemic model of the self-evolving botnets, which formulates the infection dynamics of the self-evolving botnets as a continuous-time Markov chain. The authors have shown that the infectivity of self-evolving botnets is very high, compared with conventional botnets, through numerical experiments. In response, in [3], the authors have proposed basic ideas of countermeasures against self-evolving botnets and shown their effectiveness.

In this paper, we propose a countermeasure model against self-evolving botnets, which extends the basic ideas discussed in [3]. This model aims to counter the self-evolving botnets by discovering and repairing unknown vulnerabilities by utilizing computing resources of volunteer hosts before the self-evolving botnets discover them. Therefore, we call this model volunteer model hereafter. We represent the infection dynamics of self-evolving botnets with a continuous-time Markov chain under the situation where the volunteer model works. Through simulation experiments, we examine the behavior of the volunteer model.

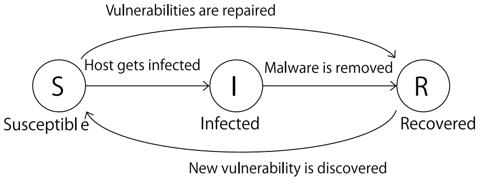

Figure 1. SIRS model.

The rest of this paper is organized as follows. Section II discusses the epidemic model for self-evolving botnets. In Section III, we explain the volunteer model. In Section IV, we discuss the behavior of the volunteer model with the results of simulation experiments. We state the conclusion of this paper in Section V.

II. Basic epidemic model for self-evolving botnets

In [4][5], in order to reveal threats of self-evolving botnets, the authors assumed situations where there is a self-evolving botnet in a network and proposed an epidemic model representing the infection dynamics of the self-evolving botnet. In this epidemic model, the state of each host in the network is represented by a Susceptible-Infected-Recovered-Susceptible (SIRS) model shown in Figure 1. In the SIRS model, "S" indicates that the host has vulnerabilities, "I" indicates that the host is infected, and "R" indicates that the host has no known vulnerabilities. Each host belongs to one of the states. We assume that hosts belonging to the recovered state R can get infected by unknown vulnerabilities which are discovered by the self-evolving botnet.

Hosts belonging to the susceptible state S transition to the infected state I when they get infected by attacks of the self-evolving botnet. Then the hosts are embedded in the self-evolving botnet. Hosts belonging to the susceptible state S and the infected state I transition to the recovered state R when known vulnerabilities and the botnet malware, respectively, removed from the hosts by, e.g., OS updates and anti-virus software. Note that we assume that all known vulnerabilities are simultaneously removed in these cases. When the self-evolving botnet discovers a new vulnerability by means of distributed machine learning using known vulnerabilities, all hosts belonging to the recovered state R transition to the susceptible state S because the botnet can infect the hosts by using the discovered vulnerability. The summary of the events in the SIRS model is as follows.

1) When a new vulnerability is discovered by the self-evolving botnet, all hosts belonging to the recovered state R transition to the susceptible state S.
2) When a host belonging to the susceptible state S removes its known vulnerabilities, it transitions to the recovered state R.
3) When a host belonging to the infected state I infects a host belonging to the susceptible state S and embeds it in the self-evolving botnet, the host getting infected transitions to the infected state I.
4) When a host belonging to the infected state I removes the botnet malware from itself, it transitions to the recovered state R.

In [5], the authors have formulated the infection process of a self-evolving botnet as a continuous-time Markov chain and evaluated its characteristic. In the Markov chain, the occurrence of each event 1)-4) described above, which is based on the SIRS model, follows a Poisson process.

III. Volunteer model

A. Modeling

Self-evolving botnets discover unknown vulnerabilities by utilizing the computing resources of zombie computers and attack susceptible hosts based on the discovered vulnerabilities. It is very difficult for each host to individually protect itself from such attacks. To overcome this difficulty, the volunteer model counters the self-evolving botnets by repairing vulnerabilities that are found with use of the computing resources of volunteer hosts before the self-evolving botnets discover them. In this paper, we represent the infection dynamics of the volunteer model under the following assumptions.

1) There is one volunteer group, to which all volunteer hosts belong, in a given network.
2) Each host in the susceptible state S or the recovered state R can become a volunteer host (i.e., join the volunteer group). The probability that a host becomes a volunteer host is proportional to the number of volunteer hosts. This assumption indicates that the effect of vulnerability discovery and protection increases with the number of volunteer hosts, so that the participation of new hosts to the volunteer group is encouraged.
3) Volunteer hosts share the information on vulnerability discovery each other and can repair the vulnerability. This is an incentive reward for participating the volunteer group. Therefore, the information is not shared with non-volunteer hosts.
4) Volunteer hosts can leave the volunteer group freely.

Figure 2 represents the state transition diagram of each host in the volunteer model, which follows these assumptions and is based on the SIRS model shown in Figure 1. In the volunteer model, the susceptible state S and the recovered state R are divided into two states "S_1", "S_2", "R_1", and "R_2", respectively. S_1 (resp. R_1) indicates that the host belongs to the susceptible state (resp. the recovered state) but does not belong to the volunteer group. On the other hand, S_2 (resp. R_2) indicates the host belongs to both the susceptible state (resp. the recovered state) and the volunteer group. In the volunteer model, the state of each host transitions according to the following event.

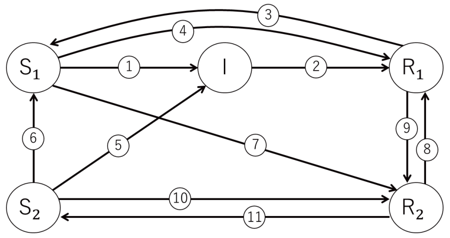

Figure 2. Volunteer model.

a) The host gets infected by an attack of an infected host (①, ⑤).
b) The host removes the botnet malware from itself (②).
c) The host removes known vulnerabilities from itself (④, ⑩).
d) The host leaves the volunteer group (⑥, ⑧).
e) The host join the volunteer group (⑦, ⑨).
f) The self-evolving botnet discovers a new vulnerability (③, ⑪).

In the event e), susceptible hosts transition to the recovered state R_2 immediately after they join the volunteer group because we assume that hosts belonging to the volunteer group share the information on vulnerabilities discovered by the volunteer group. We also assume that the volunteer group can discover unknown vulnerabilities with use of distributed machine learning, so that the probability that the transition ⑪ occurs is smaller than the probability that the transition ③ occurs in event f).

B. Continuous Markov chain

In this paper, we consider a continuous time Markov chain that represents the infection dynamics of the volunteer model, where each event occurs according to a Poisson process. Let $U_1(t)$, $U_2(t)$, $V(t)$, $W_1(t)$, and $W_2(t)$ denote the numbers of hosts belonging to the states S_1, S_2, I, R_1, and R_2, respectively at time t. The system state is represented by $(U_1(t), U_2(t), V(t), W_1(t), W_2(t))$. When the system state is $(U_1(t), U_2(t), V(t), W_1(t), W_2(t)) = (u_1, u_2, v, w_1, w_2) = \boldsymbol{\tau}$, the occurrence rate of each event is defined as follows.

a) When a host belonging to the state S_1 (resp. S_2) gets infected, the system state $\boldsymbol{\tau}$ transitions to $(u_1 - 1, u_2, v+1, w_1, w_2)$ (resp. $(u_1, u_2-1, v+1, w_1, w_2)$) (①, ⑤). The occurrence rates $\lambda_{\boldsymbol{\tau}}^{[1]}$ and $\lambda_{\boldsymbol{\tau}}^{[2]}$ of the respective events are given by

$$\lambda_{\boldsymbol{\tau}}^{[1]} = \alpha u_1 v, \tag{1}$$

$$\lambda_{\boldsymbol{\tau}}^{[2]} = \alpha u_2 v, \tag{2}$$

where α denotes the infection rate per host.

b) When a host belonging to the state I removes the botnet malware from itself, the system state $\boldsymbol{\tau}$ transitions to $(u_1, u_2, v-1, w_1+1, w_2)$ (②). The occurrence rate of this event is given by

$$\mu_{\boldsymbol{\tau}} = \delta_i v, \tag{3}$$

where δ_i denote the removal rate per host.

c) When a host belonging to the state S_1 (resp. S_2) repairs its own vulnerabilities, the system state $\boldsymbol{\tau}$ transitions to $(u_1 - 1, u_2, v, w_1 + 1, w_2)$ (resp. $(u_1, u_2 -$

$1, v, w_1, w_2 + 1)$) (④, ⑩). The occurrence rates $\psi_{\boldsymbol{\tau}}^{[1]}$ and $\psi_{\boldsymbol{\tau}}^{[2]}$ of the respective events are given by

$$\psi_{\boldsymbol{\tau}}^{[1]} = \delta_s u_1, \tag{4}$$

$$\psi_{\boldsymbol{\tau}}^{[2]} = \delta_s u_2, \tag{5}$$

where δ_s denote the repair rate per host.

d) When a host belonging to the state S_2 (resp. R_2) leaves the volunteer group, the system state $\boldsymbol{\tau}$ transitions to $(u_1 + 1, u_2 - 1, v, w_1, w_2)$ (resp. $(u_1, u_2, v, w_1 + 1, w_2 - 1)$) (⑥, ⑧). The occurrence rates $\zeta_{\boldsymbol{\tau}}^{[s]}$ and $\zeta_{\boldsymbol{\tau}}^{[r]}$ of the respective events are given by

$$\zeta_{\boldsymbol{\tau}}^{[s]} = \phi u_2, \tag{6}$$

$$\zeta_{\boldsymbol{\tau}}^{[r]} = \phi w_2, \tag{7}$$

where ϕ denotes the leave rate per host.

e) When a host belonging to the state S_1 (resp. R_1) joins the volunteer group, the system state $\boldsymbol{\tau}$ transitions to $(u_1 - 1, u_2, v, w_1, w_2 + 1)$ (resp. $(u_1, u_2, v, w_1 - 1, w_2 + 1)$) (⑦, ⑨). The occurrence rates $\epsilon_{\boldsymbol{\tau}}^{[s]}$ and $\epsilon_{\boldsymbol{\tau}}^{[s]}$ of the respective events are given by

$$\epsilon_{\boldsymbol{\tau}}^{[s]} = \theta(u_2 + w_2 + 1)u_1, \tag{8}$$

$$\epsilon_{\boldsymbol{\tau}}^{[r]} = \theta(u_2 + w_2 + 1)w_1, \tag{9}$$

where θ denotes the join rate per host. We assume that the probability that hosts join the volunteer group increases with the current size of the volunteer group.

f) When the self-evolving botnet discovers a new vulnerability, one of the following two event occurs. If the discovered vulnerability has been already repaired by the volunteer group, hosts belonging to the volunteer group do not transitions to the susceptible state. In this case, the system state $\boldsymbol{\tau}$ transitions to $(u_1 + w_1, u_2, v, 0, w_2)$ (③). The occurrence rate $\gamma_{\boldsymbol{\tau}}^{[1]}$ of this event is given by

$$\gamma_{\boldsymbol{\tau}}^{[1]} = \eta v \frac{\sigma(u_2 + w_2)}{\sigma(u_2 + w_2) + \eta v}, \tag{10}$$

where η and σ denote the vulnerability discovery rate per infected host and per volunteer host, respectively. If the discovered vulnerability has not been repaired by the volunteer group yet, hosts belonging to the volunteer group also transitions to the susceptible state. Therefore, the system state $\boldsymbol{\tau}$ transitions to $(u_1 + w_1, u_2 + w_2, v, 0, 0)$ (③, ⑪). The occurrence rate $\gamma_{\boldsymbol{\tau}}^{[2]}$ of this event is given by

$$\gamma_{\boldsymbol{\tau}}^{[2]} = \eta v \frac{\eta v}{\sigma(u_2 + w_2) + \eta v}. \tag{11}$$

We assume that the discovery capability of vulnerabilities of the self-evolving botnet (i.e., $\gamma_{\boldsymbol{\tau}}^{[1]} + \gamma_{\boldsymbol{\tau}}^{[2]} = \eta v$) is weakened according to the discovery capability of the volunteer group.

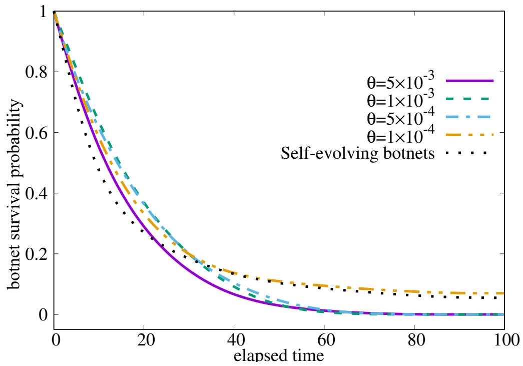

Figure 3. Botnet survival probability ($\eta = \sigma = 0.01$).

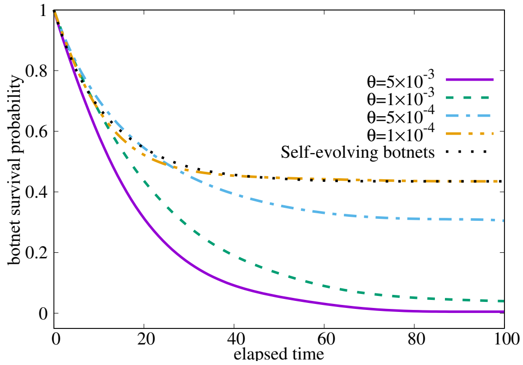

Figure 4. Botnet survival probability ($\eta = \sigma = 0.05$).

IV. Evaluation

A. Model

In this paper, we examine the infection dynamics of the volunteer model through simulation experiments. The total number of hosts in a network is equal to 1,000. The initial state of the system is assumed to be $(U_1(t), U_2(t), V(t), W_1(t), W_2(t)) = (999, 0, 1, 0, 0)$. Specifically, there is one infected host and the other hosts have vulnerabilities, which do not belong to the volunteer group. The system parameters in (1)-(11) are set to be $\alpha = 0.001$, $\delta_i = 0.1$, $\delta_s = 1$, and $\phi = 0.1$. For each experiment, we collect 200 independent samples.

B. Results

We examine the infectivity of the self-evolving botnets under infection control environments. Figures 3 and 4 show the botnet survival probability as a function of the elapsed time t, where $\eta = \sigma = 0.01$ and 0.05, respectively. The botnet survival probability means the ratio of the number of samples where one or more infected hosts still exist at time t to the total number of samples. For the sake of comparison, we plot the results for the self-evolving botnet without the volunteer model in these figures. As shown in these figures, the botnet survival probability is very large when the volunteer model is not applied to the self-evolving botnet. We also observe that when the join rate θ to the volunteer group is low (i.e., $\theta = 1 \times 10^{-4}$), the botnet survival probability is almost the same as the self-evolving botnet without the volunteer model. On the other hand, the botnet survival probability decreases with the increase in the value of θ.

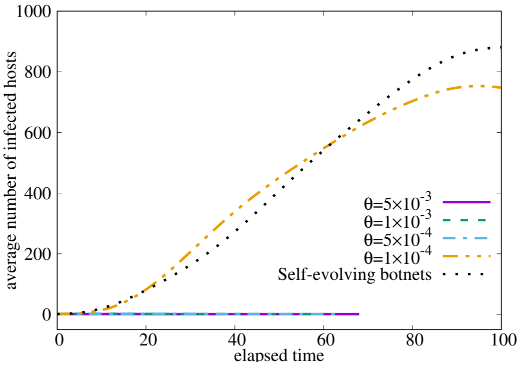

Figure 5. Average number of infected hosts ($\eta = \sigma = 0.01$).

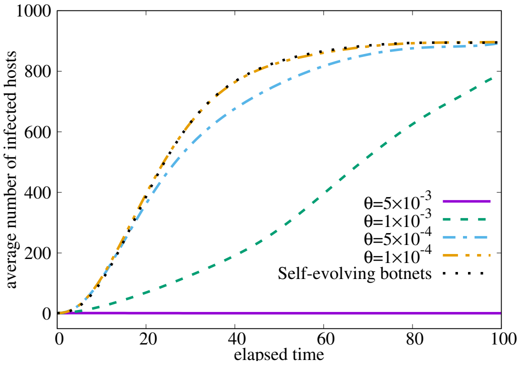

Figure 6. Average number of infected hosts ($\eta = \sigma = 0.05$).

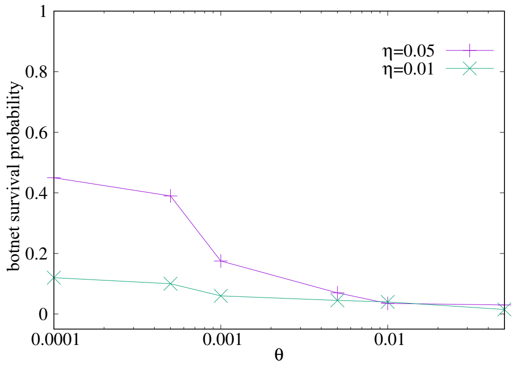

Figure 7. Botnet survival probability ($\eta = \sigma$).

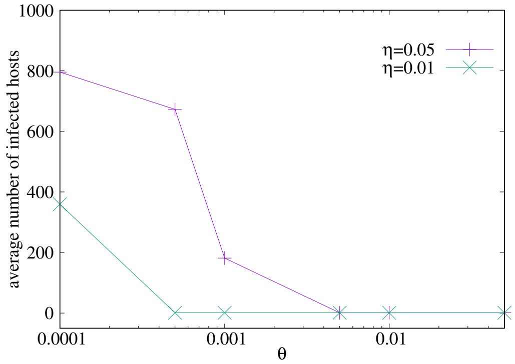

Figure 8. Average number of infected hosts ($\eta = \sigma$).

Figures 5 and 6 show the average number of infected hosts of samples in which infected hosts still exist at time t as a function of the elapsed time t, where $\eta = \sigma = 0.01$ and 0.05, respectively. From these figures, we observe that the average number of infected hosts rapidly increases when the volunteer model is not used or θ is low. Meanwhile, the volunteer model with large θ efficiently reduces the average number of infected host.

Figures 7 and 8 show the botnet survival probability and the average number of infected hosts, respectively, as a function of the value of θ, where $t = 40$ and $\eta = \sigma$. As we can see from these figures, the botnet survival probability and the average number of infected hosts decrease with the increase in the value of θ. These results mean that the volunteer model is effective for suppressing the spread of the self-evolving botnet.

V. Conclusion

This paper introduced a volunteer model to countermeasure self-evolving botnets. Through simulation experiments, we showed that the volunteer model efficiently reduces botnet survival probability and the average number of infected hosts. As future work, we will consider how hosts are encouraged to join the volunteer model. In this paper, we assume that the probability that a host becomes a volunteer host is proportional to the number of volunteer hosts. This is because the effect of vulnerability discovery and protection increases with the number of volunteer hosts. Volunteer hosts share the information on vulnerability discovery each other and can repair the vulnerability, which is an incentive reward for participating the volunteer group. However, the volunteer hosts should provide a certain amount of their computing resources, which degrade their performance. Therefore, we should consider this trade-off, using concepts such as the game theory.

Acknowledgement This is a product of research which was financially supported by the Kansai University Fund for Supporting Young Scholars, 2018, "Design of anti-malware systems against future malware evolution".

References

[1] J. Dean et al., "Large scale distributed deep networks," in *Proc. Neural Information Processing Systems*, Lake Tahoe, NV, Dec. 2012, pp. 1–11.

[2] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[3] K. Hongyo, T. Kimura, T. Kudo, Y. Inoue, and K. Hirata, "Modeling of countermeasure against self-evolving botnets," in Proc. *IEEE International Conference on Consumer Electronics - Taiwan (IEEE ICCE-TW 2017)*, Taipei, Taiwan, Jun. 2017, pp. 1–2.

[4] T. Kudo, T. Kimura, Y. Inoue, H. Aman, and K. Hirata, "Behavior analysis of self-evolving botnets," in *Proc. the 2016 International Conference on Computer, Information, and Telecommunication Systems (CITS 2016)*, Kunming, China, Jul. 2016, pp. 1–6.

[5] T. Kudo, T. Kimura, Y. Inoue, H. Aman, and K. Hirata, "Stochastic modeling of self-evolving botnets with vulnerability discovery," *Computer Communications*, vol. 124, pp. 101–110, 2018.

[6] M. A. Rajab, J. Zarfoss, F. Monrose, and A. Terzis, "A multifaceted approach to understanding the botnet phenomenon," in *Proc. ACM SIGCOMM Conference on Internet measurement*, Rio de janeiro, Brazil, Oct. 2006, pp. 1–12.

[7] R. Scandariato, J. Walden, A. Hovsepyan, and W. Joosen, "Predicting vulnerable software components via text mining," *IEEE Transactions on Software Engineering*, vol. 40, no. 10, pp. 993–1006, 2014.

[8] F. Yamaguchi, F. Lindner, and K. Rieck, "Vulnerability extrapolation: assisted discovery of vulnerabilities using machine learning," in *Proc. USENIX conference on Offensive Technologies*, San Francisco, CA, Aug. 2011, pp. 1–10.