IPSJ Journal Vol. 51 No. 3 705–715 (Mar. 2010)

Regular Paper

A Combinatorics Proliferation Model with Threshold for Malware Countermeasure ⋆1

Kazumasa Omote,†1 Takeshi Shimoyama†2
and Satoru Torii†2

Security software such as anti-virus software and personal firewall are usually installed in every host within an enterprise network. There are mainly two kinds of security software: signature-based software and anomaly-based software. Anomaly-based software generally has a "threshold" that discriminates between normal traffic and malware communications in network traffic observation. Such a threshold involves the number of packets used for behavior checking by the anomaly-based software. Also, it indicates the number of packets sent from an infected host before the infected host is contained. In this paper, we propose a mathematical model that uses discrete mathematics known as combinatorics, which is suitable for situations in which there are a small number of infected hosts. Our model can estimate the threshold at which the number of infected hosts can be suppressed to a small number. The result from our model fits very well with the result of computer simulation using typical existing scanning malware and a typical network.

1. Introduction

We aim to achieve a countermeasure against "malware" within an enterprise network. In such a network, security software such as anti-virus software and personal firewall are usually installed in every host. There are mainly two kinds of security software: signature-based software and anomaly-based software. In this study we target anomaly-based software without pattern files. This is because the signature-based scheme might not be able to detect a new malware for a few hours, because it takes time to make pattern files for each variant. Recently, there are a lot of variants of malware [2].

Infection damage by malware has been widely reported in the popular press. One of the most serious threats in an enterprise network is propagation of scanning malware (e.g., scanning worms and bots). A scanning malware scans network to find vulnerable hosts. Some scanning malwares can also select local addresses. Once a new malware has infected an enterprise network, it propagates rapidly and puts a heavy financial burden on the enterprise. We therefore consider that it is important not only to prevent the scanning malware from infection but also to prevent the scanning malware from spreading in an enterprise network. It is especially important to suppress occurrence to less than a few infected hosts, in order to reduce the financial loss to an enterprise as much as possible.

Mainly, two kinds of evaluation models for preventing scanning malware from spreading have been proposed. One is evaluation models of the Internet. These models estimate the number of infected hosts and the speed of infection. They can be either a continuous time model (e.g., SIR model [3]) or a discrete time model (e.g., AAWP model [4]). The other main type is the evaluation model of an enterprise network, such as the Staniford model [5].

The Staniford model assumes anomaly-based detection, as does our model. An anomaly-based scheme generally has a threshold that discriminates between normal traffic and malware communications in network traffic observation. This model estimates the number of infected hosts by considering the timing of blocking the infection packets sent by a victim. Such blocking is done by countermeasure software, such as a personal firewall. This timing is measured using a threshold, namely, the number of packets that can be checked by software until blocking and that can be sent out through the countermeasure software (See **Fig. 1**). If the threshold is too high, the scheme can hardly detect the scanning malware (A false-negative is frequently found). On the contrary, if the threshold is too low, the scheme frequently may mistake normal traffic for communication of the scanning malware, because the scheme cannot check enough packets (A false-positive alert is frequently generated). It is therefore important to choose an appropriate threshold in the case of anomaly-based detection. If a scanning malware is contained in quick reaction time after minor infection, infection damage can be kept to a minimum [6]. We therefore need to derive an appropriate threshold to suppress the number of infected hosts to less than a few. Note that

†1 Japan Advanced Institute of Science and Technology (JAIST)
†2 Fujitsu Laboratories, Ltd.
⋆1 The preliminary version of this paper was presented at SECRYPT 2007 [1].

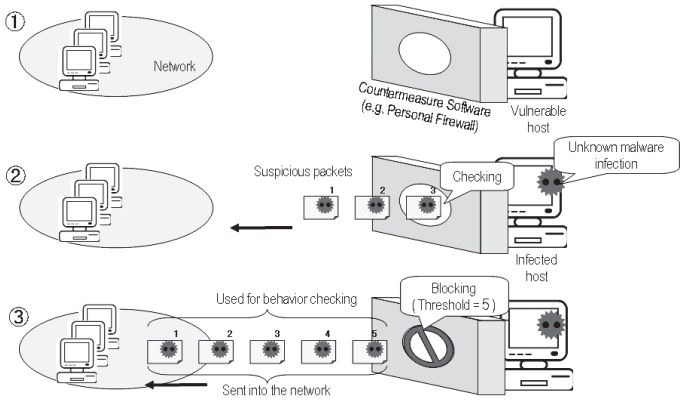

Fig. 1 Difficulty to set the timing for blocking.

we do not mention the concrete anomaly-based detection algorithm.

Our contribution. In an enterprise network, it is important not only to prevent the scanning malware from infecting a host but also to prevent the scanning malware from spreading. It is especially important to suppress the number of infected hosts as much as possible. The Staniford model can estimate an appropriate threshold according to the number of infected hosts. However, it is only suitable when the number of infected hosts is comparatively large. We therefore propose a mathematical model that uses discrete mathematics known as combinatorics, which is suitable for situations in which there are a small number of infected hosts. Our model can estimate the threshold at which the number of infected hosts can be suppressed to a small number.

We evaluated the expected number of infected hosts under a certain probability of targeting and a certain threshold by using a computer simulation, in which we developed a simulation program with the same scanning strategy as the Sasser worm (one of the most strategic scanning malwares). Note that this program was developed independently of our model. As a result, we confirmed that the result obtained with our model precisely corresponded to the result of a computer simulation when the number of infected hosts was able to be suppressed to a small number (See Section 4).

Moreover, we demonstrated that the derived threshold has a reasonable value, when we used the strategic malware in a typical network. It is important that the threshold not be too low, because the threshold also represents the number of packets that the countermeasure software can check.

2. Related Work

Various Internet evaluation models for preventing a scanning malware from spreading have been proposed. These models estimate the number of infected hosts and the rate of infection. Such Internet evaluation models include the continuous time model and the discrete time model. The SIR (susceptible-infectious-removed) model [3] is an "epidemic" continuous time model. In this model, malware can be removed at a certain rate. This model can also be used to study the effects of software patching and traffic blocking. The AAWP (analytical active worm propagation) model [4] is a discrete time model of worm propagation. This model considers the patching rate, that is, the reasonable rate at which a user can patch the vulnerability on their computer. When an infected or vulnerable host is patched, it becomes an invulnerable host.

Among the evaluation models for preventing the scanning malware from spreading within an enterprise network, the Staniford model [5] is the most famous. It can calculate the final infection density under the condition that detection and containment software is installed in every host, or is deployed in a network device (e.g., a router or switch) within the enterprise network. We describe the Staniford model in Section 2.1.

The importance of evaluation in an early stage of infection is described in Ref. 7). That work presents a non threshold-based worm-early-detection system that uses the idea of detecting the trend, that is, not the rate, of monitored scan traffic. However, this scheme cannot evaluate the threshold in the early stage of infection.

Various scan-detection schemes for observing packets behavior have been proposed. The scheme described in Ref. 8) for rate limiting counts the number of connections of a new destination address, and restricts that number. And the DNS-based scheme in Ref. 9) looks for non-DNS-based connections that use numeric IP addresses. The ARP-based scheme in Ref. 10) calculates and checks the total anomaly score from three kinds of ARP activity in order to detect

the scanning malware. The ICMP-based scheme in Ref. 11) looks for ICMP destination-unreachable (ICMP-T3) messages. These scan-detection schemes check the amount and the behavior of multiple packets.

2.1 Staniford Model

We outline the Staniford model and state its limitations. The Staniford model is composed of either the basic model (non-cell model) or the extended model (cell model). We treat the non-cell model in the present study. The Staniford model estimates the number of infected hosts by considering the infection packets sent unwillingly by a victim. This timing is measured using threshold T. It is assumed that a containment mechanism is installed in every host. Since the containment mechanism with threshold T blocks the infection packets after detection, a malware can send only T infection packets from an infected host. The threshold thus means the number of packets that can be checked until detection and containment of the malware, and the number that the scanning malware can send from an infected host. This model can calculate the final infection density under a certain threshold.

Final infection density α $(0 < \alpha < 1)$ is derived by solving Eq. (1) below of the Staniford model using threshold T, vulnerable density v, and probability P_N of targeting a host.

$$\alpha + \frac{1}{TvP_N}\ln(1-\alpha) = 0 \tag{1}$$

The value of α is constant if TvP_N is fixed because α is determined by TvP_N in Eq. (1).

The value of TvP_N, however, is the limitation factor in Eq. (1) for having solution α. If $TvP_N \leq 1$, α does not have a solution except $\alpha = 0$. α has a solution except $\alpha = 0$ as long as $TvP_N > 1$. This model can therefore accurately estimate the value of α as long as $TvP_N > 1$.

In Eq. (1), the value of TvP_N means the expected number of hosts infected by a single victim. If $TvP_N > 1$ then the infection keeps growing rapidly for a while. On the other hand, if $TvP_N < 1$, then this means that a victim will infect less than one host as an expected value. The Staniford model can, however, only estimate the value of α on the condition that the scanning malware spreads $(TvP_N > 1)$.

2.2 Threshold

Two kinds of thresholds are introduced in Ref. 12): an "epidemic threshold" and a "sustained scanning threshold" (SST). An epidemic threshold is the upper bound for preventing the scanning malware from spreading in a network. Staniford discusses the importance of this epidemic threshold from the viewpoint of the malware-containment problem. A Staniford's epidemic threshold is $1/vP_N$. In addition to the epidemic threshold, a sustained scanning threshold (SST) such as the adaptive threshold (Threshold Random Walk) [12)–14)] is well known. However, we do not target SST, because it does not consider preventing a scanning malware from spreading.

We can obtain Eq. (2) by transforming Eq. (1). Staniford's threshold is calculated as follows.

$$T = -\frac{\ln(1-\alpha)}{\alpha \cdot vP_N} \tag{2}$$

It is accurately derived under the condition $T > 1/vP_N$ $(TvP_N > 1)$. The containment software for scanning malware necessarily allows some infection packets before the number of these packets exceeds the threshold. Until the number of infection packets exceeds the threshold, the scanning malware may find one or more vulnerable hosts, and spread within the network. The more the threshold increases, the higher the propagation risk becomes.

3. Combinatorics Proliferation Model

In an enterprise network, it is important not only to prevent the scanning malware from infecting a host, but also to prevent the scanning malware from spreading. Especially, it is important to reduce the number of infected hosts as much as possible. It is thus necessary to strictly evaluate the infected hosts in the early stages of infection. We therefore propose a mathematical model that uses combinatorics. This model is suitable for the early stages of infection. It can also derive the appropriate threshold for reducing the number of infected hosts. We were not able to determine the appropriate threshold for reducing the number of infected hosts from previous known works. Although the Staniford model [5)] can derive the threshold for preventing a scanning malware from spreading, it cannot also derive the threshold for suppressing the number of infected hosts.

Our model derives the threshold for suppressing the number of infected hosts by using discrete mathematics (i.e., combinatorics). This threshold indicates the number of infection packets sent out from an infected host before the infected host is contained. The details about this model are described in the remainder of this section. Note that we do not mention the concrete anomaly-based detection algorithm. First we briefly explain the behavior of scanning malware used in our model.

3.1 Scanning Malware

A scanning malware (e.g., a scanning worm or bot) performs random scanning in a network, more specifically, it tries to communicate with a lot of other destination addresses (including non-existent addresses) and finds new vulnerable hosts. The scanning malware chooses a random IP address according to several scanning rules, and then attempts to infect it. Such scanning rules include binary search, sequential search, and universal random search.

A personal firewall is usually installed in most hosts within an enterprise network, and shuts down unnecessary ports. Some malwares, however, infect a host through the port that a firewall does not shut down. For instance, a Sasser worm tries to infect a host through the 445/TCP port which is usually opened in the host such as Windows client. We target such a terrible malware which infects a host through a personal firewall.

3.2 Premise

The premise of the combinatorics proliferation model is as follows.

(1) A single node (host) is already infected within the enterprise network.
(2) Whenever an infection packet reaches a vulnerable node, the node is infected.
(3) Vulnerable nodes are uniformly distributed within the enterprise network.
(4) An infected node sends out infection packets at regular intervals.
(5) Containment software with a threshold T is installed in every node.
(6) Probability p of targeting is constant.
(7) The time unit (1-tick) advances when one infection packet is sent out from an infected node. It is assumed that all nodes are synchronized.
(8) The processing time from receiving infection to the next infection activity is disregarded.

In premise (2), for simplicity, it is assumed that a vulnerable node is infected by one packet, although several packets (SYN packet, data packets, and so on) are actually necessary for infection. In short, we regard a data stream as one packet. Regarding premise (4), actually, most malwares based on TCP do not always send out packets at regular intervals, because the malware waits for response packets. This premise, however, would be valid in the early stages of infection. For example, we confirmed in our observation that some worms sent out the first few hundreds of SYN packets at regular intervals. Regarding premise (6), in practice, the more nodes that are already infected, the fewer the number of nodes that can be infected in the future. The probability of targeting gradually becomes smaller. Therefore, the probability of targeting in our model takes the upper bound. That is why a constant probability p is acceptable.

3.3 Notations

The notations used in our model are as follows:

- N: The number of vulnerable nodes within the enterprise network.
- p: The probability that a scanning malware picks a vulnerable address. Probability p is constant in the premise. This value of p corresponds to the value of vP_N in the Staniford model.
- T: The threshold of containment software, namely, the number of infection packets sent out from an infected node before the infected node is contained. For example, if $T = 5$ then each node can send out only 5 infection packets (see **Fig. 2**).
- k-tick: A time unit. For example, the time unit of 1-tick advances when one infection packet is sent out from an infected node.
- nth-generation: The infection distance from the infection source ($n < N$). For example, the number in "2nd-generation" means the number of grandchildren (see Fig. 2).
- $E_n(k, p, T)$: the expected number of infected nth-generation nodes after k-tick under both probability p of targeting and threshold T.
- $E(k, p, T)$: the expected number of all infected nodes after k-tick under probability p of targeting and threshold T.
- $I(p, T)$: the total expected number of infected nodes under probability p of targeting and threshold T after k is close to infinity.

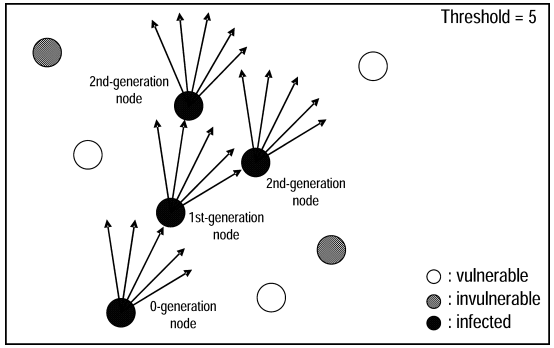

Fig. 2 An example of generation.

3.4 Probability of Targeting

The probability p of targeting is the probability that a scanning malware picks a vulnerable address described in Section 3.3. It is very rare that all vulnerable addresses are used within an enterprise network. The address space in such a network is usually only partly used. Moreover, since the scanning malware selects nodes to target probabilistically, an infection packet that is sent out from an infected node does not always reach a vulnerable node. We therefore get probability p of targeting by using both the number of vulnerable nodes and the target-selection algorithm of the scanning malware operating.

For instance, p is calculated in IPv4 network topology as follows. An existing scanning malware mainly uses two kinds of target-selection algorithms: (1) the malware selects a target node completely randomly or (2) the malware selects a target node probabilistically according to the local subnet. The Sasser worm, which is one of the most strategic malwares, chooses an address from the same /8 subnet (the number of vulnerable nodes is N_a) with probability 1/4, chooses a random address from the same /16 subnet (the number of vulnerable nodes is N_b) with probability 1/4, and chooses a random Internet address with probability 2/4. Hence probability p of targeting is calculated as follows.

$$p = \frac{2}{4}\left(\frac{N-1}{2^{32}}\right) + \frac{1}{4}\left(\frac{N_a-1}{2^{24}}\right) + \frac{1}{4}\left(\frac{N_b-1}{2^{16}}\right) \tag{3}$$

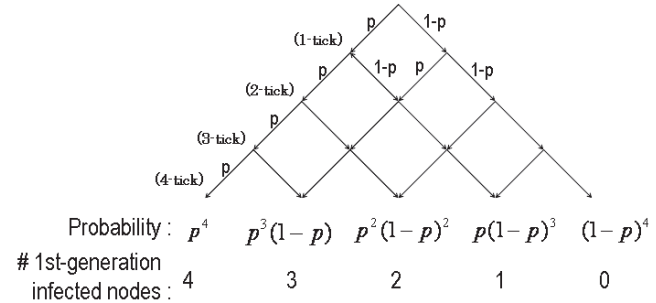

Fig. 3 Number of 1st-generation infected nodes after 4-tick.

3.5 Expected Number of Infected Nodes

Expected number of infected nodes is calculated by the probability p and the threshold T in our model. We define the number of 0-generation infected nodes (the infection source) as $E_0(k, p, T) = 1$ regardless of k, p and T. The 1st-generation infected node is the target that the infection source will infect directly. The number of 1st-generation infected nodes after k-tick is the sum of nodes that the infection source directly infects until k-tick. For example, the number of 1st-generation infected nodes after 4-tick changes from 0 to 4 under each infection probability in **Fig. 3**. The expected number of 1st-generation infected nodes after 4-tick is calculated as follows.

$$\begin{aligned} E_1(4,p,T) &= 1 \cdot {}_4C_1 \cdot p(1-p)^3 \\ &\quad + 2 \cdot {}_4C_2 \cdot p^2(1-p)^2 \\ &\quad + 3 \cdot {}_4C_3 \cdot p^3(1-p) \\ &\quad + 4 \cdot {}_4C_4 \cdot p^4 \\ &= \sum_{i=1}^{4} i \cdot {}_4C_i \cdot p^i(1-p)^{4-i} \quad (T > 4) \end{aligned}$$

The expected number of 1st-generation infected nodes after k-tick is therefore calculated as follows.

$$E_1(k,p,T) = \begin{cases} \displaystyle\sum_{i=1}^{k} i \cdot {}_kC_i \cdot p^i(1-p)^{k-i} & (k < T) \\ \displaystyle\sum_{i=1}^{T} i \cdot {}_TC_i \cdot p^i(1-p)^{T-i} & (k \geq T) \end{cases} \quad (4)$$

Since the infection source sends out only T packets, $E_1(k,p,T)$ satisfies as follows:

$$E_1(k,p,T) = E_1(T,p,T), \quad \text{if } k \geq T. \quad (5)$$

The number of 2nd-generation infected nodes after k-tick is as follows using the number of 1st-generation infected nodes.

$$\begin{aligned} E_2(k,p,T) = {} & E_1(1,p,T) \cdot E_1(k-1,p,T) \\ & + (E_1(2,p,T) - E_1(1,p,T)) \cdot E_1(k-2,p,T) \\ & + (E_1(3,p,T) - E_1(2,p,T)) \cdot E_1(k-3,p,T) \\ & + \cdots \\ & + (E_1(T,p,T) - E_1(T-1,p,T)) \cdot E_1(k-T,p,T) \end{aligned} \quad (6)$$

The number of 2nd-generation infected nodes does not include the number of 1st-generation infected nodes because the 1st-generation infected nodes can not be infected twice. The value of $(E_1(T,p,T) - E_1(T-1,p,T))$ means the number of 1st-generation infected nodes which are infected at T-tick for the first time. The number of nth-generation infected nodes similarly does not include the sum from 1st-generation infected nodes to $(n-1)$th-generation ones. We have the following theorem to derive $E_n(k,p,T)$.

Theorem 1 Let n be a positive integer. If $k \gg T$ then the following approximation holds: $E_n(k,p,T) \simeq E_1(T,p,T)^n$.

Proof. We prove this by complete induction. We find the approximation $E_1(k-1,p,T) \simeq E_1(k-2,p,T) \simeq \cdots \simeq E_1(k-T,p,T) \simeq E_1(T,p,T)$ from Eq. (5) because of $k \gg T$. The case of $n = 1$ is trivial. The assertion is true when $n = 2$ since we get $E_2(k,p,T) \simeq E_1(T,p,T)^2$ from Eq. (6). Suppose that $E_n(k,p,T) \simeq E_1(T,p,T)^n$ $(n \geq 3)$. We have the number of $(n+1)$th-generation infected nodes after k-tick as follows.

$$\begin{aligned} E_{n+1}(k,p,T) = {} & E_n(1,p,T) \cdot E_1(k-1,p,T) \\ & + (E_n(2,p,T) - E_n(1,p,T)) \cdot E_1(k-2,p,T) \\ & + \cdots \\ & + (E_n(T,p,T) - E_n(T-1,p,T)) \cdot E_1(k-T,p,T) \\ \simeq {} & E_n(T,p,T) \cdot E_1(k-T,p,T) \\ \simeq {} & E_1(T,p,T)^{n+1} \end{aligned}$$

The proof is done because $E_n(k,p,T) \simeq E_1(T,p,T)^n$ $(n \geq 1)$ holds. ∎

The total expected number of infected nodes after k-tick is subsequently the sum of victims from the infection source (0-generation) to the kth-generation, calculated as follows.

$$\begin{aligned} E(k,p,T) &= \sum_{i=0}^{k} E_i(k,p,T) \\ &= \sum_{i=0}^{k} E_1(T,p,T)^i \end{aligned} \quad (7)$$

After k is close to infinity, the total expected number of infected nodes under probability p of targeting and threshold T is calculated as follows.

$$\begin{aligned} I(p,T) &= \lim_{k\to\infty} \sum_{i=0}^{k} E_1(T,p,T)^i \\ &= \frac{1}{1 - E_1(T,p,T)}, \quad \text{if } E_1(T,p,T) < 1 \end{aligned} \quad (8)$$

The above equation derives the number of infected nodes when each victim sends out T infection packets with probability p. Fortunately, our model is approximated to the calculation of the 1st-generation infection. We have the following theorem about $E_1(k,p,T)$.

Theorem 2 If $k \gg T$ then $E_1(k,p,T) = pT$.

Proof. Transform $E_1(k,p,T)$ using Eq. (4) as follows:

$$\begin{aligned} E_1(k,p,T) &= \sum_{i=1}^{T} i \cdot {}_TC_i \cdot p^i(1-p)^{T-i} \\ &= p \cdot \sum_{i=1}^{T} i \cdot {}_TC_{T-i} \cdot p^{i-1}(1-p)^{T-i} \\ &= p \cdot \sum_{i=1}^{T} T \cdot {}_{T-1}C_{T-i} \cdot p^{i-1}(1-p)^{T-i} \end{aligned}$$

$$= pT \cdot \sum_{i=1}^{T} {}_{T-1}C_{i-1} \cdot p^{i-1}(1-p)^{T-i}$$
$$= pT \cdot ((1-p)+p)^{T-1}$$
$$= pT. \quad \blacksquare$$

If $k \geq T$ then $E_1(T,p,T) = pT$ since $E_1(k,p,T) = E_1(T,p,T)$. If $E_1(T,p,T) < 1$ then the value of $I(p,T)$ is finite in Eq. (8). Therefore, the coverage of our threshold is as follows:

$$T < \frac{1}{p}. \tag{9}$$

As a result, we have $I(p,T)$ as follows:

$$I(p,T) = \frac{1}{1-pT}, \quad \text{if } T < \frac{1}{p}. \tag{10}$$

3.6 Upper Bound of T

We can get the upper bound of T using Eq. (10) in the following steps.

(1) Plural values of $I(p,T)$ are calculated from increments of T under a probability p.

(2) The upper bound of T to satisfy the following equation is obtained.

$$I(p,T) < u \tag{11}$$

u is the upper bound of the expected number of infected nodes. We can obtain the threshold according to the expected number of infected nodes by changing u. For example, the setting $u = 2$ in Eq. (11) means that the total expected number of infected nodes is finally less than two (the expected number of new infected nodes is less than one only).

4. Computer Simulation with Sasser Worm

Our goal is to evaluate the expected number of infected nodes under probability p and threshold T by using a computer simulation. We confirmed that the result from our model precisely corresponds to the result of computer simulation under the condition $T < 1/p$. In our evaluation, we assume that an actual scanning malware has damaged the enterprise network.

We simulate the malware spreading by using a simple Monte Carlo simulator, under the condition that containment software with the threshold is installed in each node. We developed a simulation program with the same scanning strategy as the Sasser worm (one of the most strategic scanning malwares). Note that this program was developed independently of our model. Every address is modeled to determine whether it is invulnerable, vulnerable or infected. A malware selects only T addresses for scanning, and then stops its activity. To establish reliable statistics on malware behavior, the computer simulation is repeatedly run with different seeds. Since malware spreading is randomized differently on each run, the result of one simulation will be different from the next. If the selected address is vulnerable, the node is always infected. Also, if the selected address is infected or invulnerable, the state of the node will be unchanged even if it receives an infected packet. The difference between the computer simulation and our model is that the probability of targeting a node can be changed in the computer simulation.

Figure 4 shows that three kinds of values of $I(p,T)$ in our model fit the results of computer simulation when the expected number of infected nodes is less than about 10. In the computer simulation, the infection by a Sasser worm in the

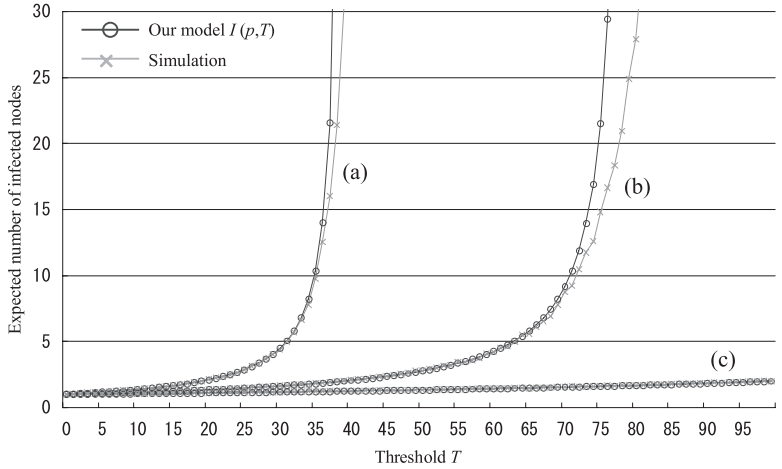

Fig. 4 The relation between the result of computer simulation and the result of $I(p,T)$ in our model using a Sasser worm in the subnet of class-B ((a) $d = 0.1$, (b) $d = 0.05$, (c) $d = 0.02$).

subnet of class-B of an enterprise network is considered. The set of experiments we did involved the selected parameters: the size of the subnet of class-B: 2^{16}, three kinds of vulnerable-node density (d): 0.1, 0.05 and 0.02, the correspond number of vulnerable nodes ($N = 2^{16} \cdot d$): 6,554, 3,277 and 1,311, and the corresponding probability (p): 0.0251, 0.0125 and 0.00502. The value of p is calculated from Eq. (3) with $N = N_a = N_b$. We consider the worst case that all vulnerable nodes are included in N_b. We simulated 10,000 runs by varying T in increments of 1, and plotted the average values.

In Fig. 4, the value of $I(p, T)$ in our model becomes larger than the result of the computer simulation when the number of infected nodes becomes large, because the probability of targeting in our model is constant for simplicity (see Section 3.2) and our computer simulation does not have such a premise. Therefore, the probability of targeting might decrease as the number of infected nodes increase.

5. Discussion

5.1 Coverage of Threshold

As mentioned in Section 2.2, Staniford's threshold is derived under the condition $T > 1/vP_N$ ($T > 1/p$). However, our threshold is derived under the condition $T < 1/p$ in Eq. (9). Note that $T = 1/p$ ($1/vP_N$) is a singularity in both models. In this section, we confirm that the coverage of the two above-described thresholds is different.

We compare the results from both the Staniford model and our model with the computer-simulation results under the same condition as stated in the previous section. **Figure 5** extends the x-axis of Fig. 4, and also includes the results from the Staniford model. While Staniford's result was calculated using Eq. (2), our threshold is calculated using Eq. (11). For the expected number of infected nodes, the Staniford model uses $\alpha \cdot N$ but our model uses $I(p, T)$.

Regarding the range for fitting the computer-simulation results in Fig. 5, our model is different from the Staniford model. Concretely, while the coverage of the Staniford model is (a) $T > 39.8$ (1/0.0251) (b) $T > 80.0$ (1/0.0125) and (c) $T > 199$ (1/0.00502), the coverage of our model is (a) $T < 39.8$, (b) $T < 80.0$ and (c) $T < 199$, respectively. In the Staniford model, threshold T cannot be

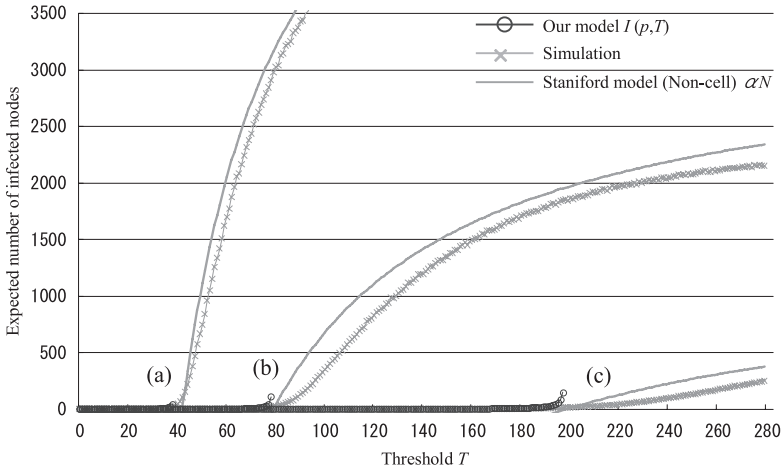

Fig. 5 The relation between the result of computer simulation and the results of both αN in the Staniford model and $I(p, T)$ in our model using a Sasser worm in the subnet of class-B ((a) $d = 0.1$, (b) $d = 0.05$, (c) $d = 0.02$).

calculated when (a) $T < 39.8$ (b) $T < 80.0$ and (c) $T < 199$. The boundary points between the Staniford model and our model are (a) $T = 39.8$ (b) $T = 80.0$ and (c) $T = 199$. The target range of threshold T is clearly divided between the Staniford model and our model. As shown in Fig. 4, therefore, our model is suitable for evaluation of the expected number of infected nodes, to reduce the number of infected nodes to below the threshold.

5.2 Approximation Calculation

We discuss Eq. (8) to explain about the approximation calculation. Since our model considers generation infection, it must calculate the number of infected nodes up to the number of kth-generation infections after k-tick. Fortunately, our model is approximated to the calculation of the 1st-generation infections like Eq. (7). We can therefore easily calculate the total expected number of infected nodes from only the number of 1st-generation infections.

5.3 Effective Threshold

Although we can obtain the epidemic threshold (i.e., the boundary point in Fig. 5) from Staniford model, this threshold is just the upper bound for preventing the scanning malware from spreading in a network. We want to find the effective

Table 1 The relation between the vulnerable-node density d in the subnet of class-B and the upper bound of T to prevent a Sasser worm from spreading when $I(p,T) < 2$.

	d	p	#vulnerable nodes	upper bound of T
(a)	0.1	0.0251	6,554	19
(b)	0.05	0.0125	3,277	39
(c)	0.02	0.00502	1,311	99

threshold which can estimate the number of infected nodes in the early stages of infection. However, we did not obtain such a threshold T that satisfies $I(p,T) < u$ (u is a small number) from the existing models. In our model, we can find T such that $I(p,T) < u$ (i.e., the number of new infected nodes is less than $u-1$). Hence the condition $pT < 1-1/u$ is obtained from Eq. (10). This means that the expected number of infected nodes from a single victim must become less than $1-1/u$ in order to suppress the number of infected nodes to less than u.

6. Case Study

Intuitively, the higher the vulnerable-node density in the network becomes, the easier spreading infection becomes. **Table 1** gives the relation between the node distribution of the network and the upper bound of T for a Sasser worm in the subnet of class-B when $I(p,T) < 2$. We use three kinds of density values as used in Section 4 and 5. If we can lower the vulnerable-node density d, then we can decrease the probability p. Hence we can reduce the threat of malware by distributing nodes sparsely. As a result, the upper bound of the threshold can be set higher. For example, we can set $T = 19$ when $d = 0.1$ and we can set $T = 99$ when $d = 0.02$, in the subnet of class-B.

From the viewpoint of suppressing the number of infected nodes, it is important to expand the number of subnets and to lower the density of nodes in each subnet. If the upper bound of the threshold can be raised, it implies that we can increase the number of packets used for behavior checking of network communication by an infected node.

7. Applying Our Model to a Real Network

7.1 Propagation Parameters

There are many propagation parameters on a real network, i.e., installation ratio of security measures (personal firewall, anti-virus software, and patches), node operation ratio, OS type distribution, and communication speed. The vulnerable-node density in our model is determined by the non-installation ratio of anti-virus software and patches, the OS type distribution, and node operation ratio. If a node installs anti-virus software and patches or its OS type is different from malware's target, the node is excluded from vulnerable nodes. A non-operated node is not also included in vulnerable nodes. However, we assume that the installation ratio of personal firewall is 100% as a premise. If a node without the personal firewall is infected in a real network, it will keep propagating malwares until detected. Our model does not treat such a situation. Furthermore, in our model, we assume that the communication speed is constant in a network. Our model does not treat the network with non-uniform communication bandwidth.

7.2 Decreasing of False Detection

There are two kinds of false detection (false positives and false negatives) in the anomaly detection software. When our model is applied to the anomaly detection software, we also have to consider how to decrease false detection.

Assume that the countermeasure software counts the destination of communication and then blocks the communication by the threshold T. The value of T should be set small in order to suppress the number of infections. However, the false positive will frequently occur if T is set small, because a legitimate node may access many other nodes (e.g., HTTP client, SMTP server, and P2P services). Thus, we consider how to suppress the false positives when our model is applied to a real network. At first, we obtain the following information.

- p: The estimated probability of targeting. This implies the threat of envisioned malwares.
- V: The maximum number of destination of normal communication by a legitimate node. This is obtained by observation in a real network.
- d: The estimated vulnerable-node density.
- u: The upper bound of the expected number of infected nodes.

Then, T should be set higher than V in order to suppress the false positives. Of course, we have to choose T such that $I(p,T) < u$. However, if both $I(p,T) < u$ and $T > V$ are not satisfied, then we can increase the upper bound of T by reducing d. For instance, we can increase the upper bound of T from 19 to 99

by reducing the density d from 0.1 to 0.02 in Table 1. As a result, we would be able to set the threshold which suppresses the false positives in a real network.

Of course, it is also important to decrease the false negatives. However, it is not so important to minimize the false negatives from viewpoints of preventing the malware from spreading. We assume that our model admits several infections. As long as $I(p,T) < u$ is satisfied, it can suppress the number of infected nodes to less than u even if some false negatives occur.

Therefore, we can use the threshold T such that $T > V$ and $I(p,T) < u$. Note that we cannot always find such a threshold T because of a legitimate node access in a network.

8. Conclusion

We proposed a "combinatorics proliferation model" based on discrete mathematics (combinatorics) and derived the threshold T for satisfying $I(p,T) < u$ (u is a small number), where $I(p,T)$ is the expected number of infected hosts. We confirmed that the results from this model precisely correspond to the results of computer simulation of malware spreading when $T < 1/p$ is satisfied.

We demonstrated that the derived threshold has a reasonable value when we used the strategic malware (Sasser worm) in a typical class-B network. Our model can appropriately express the number of infected hosts in the early stages of infection, and can derive the effective threshold to contain the scanning malware in the enterprise network to a few infections only.

References

1) Omote, K., Shimoyama, T. and Torii, S.: A Combinatorics Proliferation Model to Determine the Timing for Blocking Scanning Malware, *Proc. International Conference on Security and Cryptography – SECRYPT*, pp.16–24 (2007).
2) Barford, P. and Yegneswaran, V.: An inside look at botnets, *Special Workshop on Malware Detection, Advances in Information Security*, Springer-Verlag (2006).
3) Nikoloski, Z. and Kucera, L.: Correlation model of worm propagation on scale-free networks, *Complexus 2006*, Vol.3, pp.169–182 (2006).
4) Chen, Z., Gao, L. and Kwiat, K.: Modeling the spread of active worms, *Proc. INFOCOM 2003*, pp.1890–1900, IEEE (2003).
5) Staniford, S.: Containment of scanning worms in enterprise networks, *Journal of Computer Security* (2004).
6) Moore, D., Shannon, C., Voelker, G.M. and Savage, S.: Internet quarantine: Requirements for containing self-propagating code, *Proc. INFOCOM 2003*, pp.1901–1910, IEEE (2003).
7) Zou, C.C., Gao, L., Gong, W. and Towsley, D.: Monitoring and early warning for Internet worms, *Proc. 10th ACM Conference on Computer and Communication Security – CCS'03*, pp.190–199, ACM (2003).
8) Williamson, M.M.: Throttling viruses: Restricting propagation to defeat malicious mobile code, *Proc. 18th Annual Computer Security Applications Conference – ACSAC'02*, pp.61–68, IEEE (2002).
9) Whyte, D., Kranakis, E. and Oorschot, P.: DNS-based detection of scanning worms in an enterprise network, *Proc. 12th Annual Network and Distributed System Security Symposium – NDSS'05*, Internet Society (2005).
10) Whyte, D., Oorschot, P. and Kranakis, E.: Detecting Intra-enterprise scanning worms based on address resolution, *Proc. 21st Annual Computer Security Applications Conference – ACSAC'05*, pp.371–380, IEEE (2005).
11) Bakos, G. and Berk, V.H.: Early detection of Internet worm activity by metering ICMP destination unreachable messages, *Proc. SPIE Conference on Command, Control, Communications and Intelligence*, pp.33–42, SPIE Press (2002).
12) Weaver, N., Staniford, S. and Paxson, V.: Very fast containment of scanning worms, *Proc. 13th USENIX Security Symposium*, pp.29–44 (2004).
13) Jung, J., Paxson, V., Berger, A.W. and Balakrishnan, H.: Fast portscan detection using sequential hypothesis testing, *Proc. IEEE Symposium on Security and Privacy*, pp.211–225 (2004).
14) Schechter, S.E., Jung, J. and Berger, A.W.: Fast detection of scanning worm infections, *Proc. 7th International Symposium on Recent Advances in Intrusion Detection – RAID'04*, LNCS 3224, pp.59–81, Springer-Verlag (2004).

(Received May 18, 2009)
(Accepted December 17, 2009)
(Original version of this article can be found in the Journal of Information Processing Vol.18, pp.77–87.)

Kazumasa Omote received his M.S. and Ph.D. degrees in information science from Japan Advanced Institute of Science and Technology (JAIST) in 1999 and 2002, respectively. He joined Fujitsu Laboratories Ltd. from 2002 to 2008 and engaged in research and development for network security. He has been a research assistant professor at Japan Advanced Institute of Science and Technology (JAIST) since 2008. His research interests include applied cryptography and network security. He has received the Best Student-Paper Award at CSS 2001 and the Best Paper Award at CSS 2004. He is a member of Information Processing Society of Japan (IPSJ).

Takeshi Shimoyama received his B.S., M.S. degrees in mathematics from Yokohama City University in 1989 and 1991, respectively, and D.E. degree in information and system engineering from Chuo University in 2000. He is a research engineer of Fujitsu Laboratories Ltd. from 1991. He joined the Research Project of Info Communication Security under Telecommunications Advancement Organization of Japan from 1996 to 1998. His current research interests are in cryptanalysis and information security. He was awarded SCIS paper prize in 1997, IWSEC paper prize in 2007, and OHM Technology Award in 2007. He attained the world record of an integer factoring by GNFS in 2006.

Satoru Torii received the B.S. degree in Information Sciences from Tokyo University of Science, Chiba, Japan in 1985. He joined Fujitsu Laboratories Ltd., Kawasaki, Japan in 1985, where he has been engaged in research and development of operating systems, intrusion management systems, and network security. He is a member of Information Processing Society of Japan (IPSJ).