A COMBINATORICS PROLIFERATION MODEL TO DETERMINE THE TIMING FOR BLOCKING SCANNING MALWARE

Kazumasa Omote
University of Tsukuba, Tokyo, Japan

Takeshi Shimoyama, Satoru Torii
Fujitsu Laboratories, Ltd., Kawasaki, Kanagawa, Japan

Keywords: Malware countermeasure, enterprise network, threshold, computer simulation.

Abstract: One of the worst threats present in an enterprise network is the propagation of "scanning malware" (e.g., scanning worms and bots). It is important to prevent such scanning malware from spreading within an enterprise network. It is especially important to suppress scanning malware infection to less than a few infected hosts. We estimated the timing of containment software to block "scanning malware" in a homogeneous enterprise network. The "combinatorics proliferation model", based on discrete mathematics, developed in this study derives a threshold that gives the number of the packets sent by a victim that must not be exceeded in order to suppress the number of infected hosts to less than a few. This model can appropriately express the early state under which an infection started. The result from our model fits very well to the result of computer simulation using a typical existing scanning malware and an actual network.

1 INTRODUCTION

We aim at achieving a countermeasure against "malware" within an enterprise network. In such a network, security software such as the anti-virus detection and containment is mostly installed in every host. There are mainly two kinds of security software: signature-based detection software and anomaly-based software. In this study we target anomaly-based detection 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. On top of that, there are recently a lot of variants of malware (Barford et al., 2006).

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 recent scanning malware can select local addresses. Once a new malware is infected within an enterprise network, it propagates rapidly and puts a heavy financial burden on the enterprise. We therefore consider that it is important to prevent such a malware from spreading within an enterprise network. It is especially important to suppress their occurrence to less than a few infected hosts in order to decrease the financial loss to an enterprise as much as possible.

Mainly, two kinds of evaluation model 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 are either a continuous time model (e.g., SIR model (Nikoloski et al., 2006)) or a discrete time model (e.g., AAWP model (Chen et al., 2003)).

Another example is the evaluation model of an enterprise network such as the Staniford model (Staniford, 2004). This model evaluates the number of infected hosts by considering the timing for blocking the infection packets sent by a victim. This timing is measured using a threshold, namely, the number of packets that can be checked until detection and containment of an infection and that a malware scanner can send out to deal with the infection. If a scanning malware is contained in quick reaction time after minor infection, the

Omote K., Shimoyama T. and Torii S. (2007).
A COMBINATORICS PROLIFERATION MODEL TO DETERMINE THE TIMING FOR BLOCKING SCANNING MALWARE.
In *Proceedings of the Second International Conference on Security and Cryptography*, pages 16-24
DOI: 10.5220/0002119300160024

infection damage can be considerably suppressed (Moore et al., 2003).

The Staniford model assumes anomaly-based detection. It is thus necessary to discriminate between normal traffic and new malware scans. An anomaly-based scheme generally has a threshold that discriminates between normal traffic and malware communications. If the threshold is too high, the scanning malware can scan and spread through the enterprise network. On the contrary, if the threshold is too low, the scheme frequently mistakes normal traffic for a communication of the scanning malware (A false positive alert is frequently generated.) It is therefore important to choose the appropriate threshold in the case of an anomaly-based detection.

The Staniford model can estimate the threshold according to the number of infected hosts. However, it is suitable as long as the number of infected hosts is comparatively large. We therefore need to derive the threshold that suppresses the number of infected hosts to less than a few.

1.1 Our Contribution

In an enterprise network, it is important not only to prevent the scanning malware from spreading but also to suppress the number of infected hosts as much as possible. Our model is suitable for situations in which there are a small number of infected hosts. It uses discrete mathematics known as combinatorics. It can also 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 threshold by using a computer simulation. As a result, we confirmed that the result obtained with our model precisely corresponds to the result of a computer simulation when the number of infected hosts can be suppressed to a small number.

Moreover, we clarified the relation between the number of subnets and the upper bound of the threshold when the number of hosts is evenly distributed within the enterprise network. We conclude from this result that the more the number of subnets increases, the higher the upper bound of threshold can be set.

2 RELATED WORK

Various Internet evaluation models of 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 (Nikoloski et al., 2006) is an “epidemic” continuous time model. In this model, an infected host can be removed at a certain rate. It can also be used to study the effect of software patching and traffic blocking. The AAWP (analytical active worm propagation) model (Chen et al., 2003) 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.

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

The importance of evaluation in an early stage of infection is described in (Zou et al., 2003). 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 does not evaluate the threshold in the early stage of infection.

Various scan-detection schemes for observing packets behavior have been proposed. The scheme described in (Williamson, 2002) for rate limiting counts the number of connections of a new destination address and restricts that number. And the DNS-based scheme in (Whyte et al., 2005a) looks for non-DNS-based connections that use numeric IP addresses. The ARP-based scheme in (Whyte et al., 2005b) 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 (Bakos et al., 2002) looks for ICMP destination-unreachable (ICMP-T3) messages. These scan-detection schemes check the amount and the behavior of plural packets.

2.1 Staniford Model

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 evaluates the number of infected hosts by considering the infection packets sent unwillingly by a victim. This timing is measured using threshold *T*.

In this section, we outline the Staniford model and state its limitations. It is assumed that a containment mechanism is installed by 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 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 Equation (1) below of the Staniford model using threshold *T*, vulnerable density *v*, and probability P_N of targeting a host.

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

The value of α is constant if TvP_N is the same because α is determined by TvP_N in Equation (1).

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

In Equation (1), the value of TvP_N means the expected number of hosts that a single victim infected. If $TvP_N > 1$ is satisfied, the infection keeps growing rapidly for a while and the scanning malware spreads. On the other hand, if $TvP_N < 1$ is satisfied, this means that chances are the first infection will infect less than one other site. The Staniford model can therefore only estimate the value of α on the condition that the scanning malware spreads.

2.2 Threshold

Two kinds of threshold are introduced in (Weaver et al., 2004): an "epidemic threshold" and a "sustained scanning threshold" (SST). An epidemic threshold is the upper bound for preventing the scanning malware from spreading in an enterprise network. Staniford discusses the importance of this epidemic threshold from the viewpoint of the worm-containment problem. The containment software for scanning malware necessarily allows some scans before the number of scans exceeds the threshold. Until the number of scans exceeds the threshold, the scanning malware may find one or more vulnerable hosts and spread within the enterprise network. The more the threshold increases, the higher the propagation risk becomes. It is thus important to derive such an epidemic threshold.

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

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

It is accurately derived under the condition $T > 1/vP_N$ ($TvP_N > 1$).

In addition to the epidemic threshold, a sustained scanning threshold (SST) such as the adaptive threshold (Threshold Random Walk) (Weaver et al., 2004; Jung et al., 2004; Schechter et al., 2004) is known well. However, we do not target a SST because it does not consider preventing a scanning malware from spreading.

3 CONTAINMENT OF SCANNING MALWARE

Our goal is to detect a scanning malware and prevent it from spreading. This study is limited to the containment of random scanning malware and does not deal with the issue of containing flash and topological worms. Furthermore, we target an enterprise network in which the containment software is widely installed or in which a containment device is widely deployed.

3.1 Characteristics of Scanning Malware

A scanning malware (e.g., a scanning worm or bot) performs a random scanning in a network. Since the scanning malware tries to communicate with a lot of other destination addresses (including non-existent addresses) and finds new vulnerable hosts, it communicates with a host whom a correct user rarely does. 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.

We experimentally verified that it is difficult to distinguish between the scanning malware and the normal traffic by counting the number of packets, because the malware scanning is buried in the normal traffic (Omote et al., 2003). On the other hand, we also experimentally verified that it is possible to distinguish between the scanning malware and the normal traffic by counting the number of destination addresses, because the normal traffic is limited to several destinations.

3.2 A Detection and Containment Method for Scanning Malware

Our scanning malware-detection scheme directly observes a scanning malware's connection. It is different from related works such as (Whyte et al., 2005a; Whyte et al., 2005b; Bakos et al., 2002), which do not directly observe the behavior of scanning packets. Since our scheme directly observes outgoing scanning packets, it can grasp the header information of IP packets in the network traffic and determine its amount. Note that it is assumed that the detection and containment software is installed in a host or is deployed in a network device (e.g., a router or switch) within the enterprise network.

The scheme counts the number of scans sent by a single host. More concretely, the scheme counts at a short interval the number of destination IP addresses in the first outbound connection packet, such as a SYN packet, that has the same source IP address, destination port, and protocol. The scheme is also very simple for observation of packets because it only refers to the header information of IP packets.

A detection alert is generated when the count value of scans becomes more than the threshold. This threshold is corresponding to Staniford's threshold. Although the count value of scans sent by a host is cleared after a certain time interval, a detection alert is generated as soon as the count value reaches the threshold before such an interval expires.

Threshold *T* used in our previous scheme has the following meaning. The scheme blocks the connection of worm after the number of outgoing packets exceeds *T*. This means that the scanning malware is permitted to scan *T* times from a victim until the alert is generated. If the threshold is too high, the scanning malware can scan through the enterprise network and propagate. On the contrary, if the threshold is too low, the scheme frequently mistakes normal traffic for the scanning malware (and a false-positive alert is generated). It is therefore important to determine the appropriate threshold for our scheme.

4 COMBINATORICS PROLIFERATION MODEL

In an enterprise network, it is important not only to prevent scanning malware from spreading but also to suppress 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 threshold for suppressing the number of infected hosts.

In the previous section, we stated the importance of determining the appropriate threshold to prevent both the spreading and mistaking of normal traffic. However, we cannot determine the threshold for the number of infected hosts to be suppressed from previous known works. In regards to the Staniford model (Staniford, 2004), although it can derive the threshold for preventing a scanning malware from spreading, it cannot derive the threshold for suppressing the number of infected hosts.

4.1 Outline

Our model derives the threshold for suppressing the number of infected hosts by using discrete mathematics (i.e., combinatorics). This threshold means the number of scans that go out from an infected host before the host is contained. The details about this model are described in the remainder of this section. First the model's preparation is described, then the design of the model is explained, and, lastly, the method for deriving the upper bound of the threshold is described.

4.2 Premise

The premise of combinatorics proliferation model is as follows.

1. To start with, a single node is already infected within the enterprise network.
2. Whenever the infection packet reaches a vulnerable node, the node is infected.
3. A vulnerable node is uniformly distributed within the enterprise network.
4. An infected node sends out the infection packet at regular intervals.
5. Containment software with the same 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.
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 host is infected by one packet, though several packets (SYN packet, data packet, and so on) are actually necessary for infection. In regards to premise 6, refer to section 4.4. In regards to premise 7, we introduce a time parameter into our model.

4.3 Parameters

The parameters used in our model are as follows:

- Number of vulnerable nodes (*N*): the number of vulnerable nodes (hosts) within the enterprise network.
- Vulnerable node density (*D*): the density of vulnerable nodes in the enterprise network. For example, when the number of hosts in the subnet of class-B is *N*b, *D* is expressed as $(N_b - 1)/2^{16}$.
- Probability of targeting a host (*p*): the probability that a scanning malware picks a vulnerable address. Probability *p* is constant in the premise. And *p* corresponds to the value of vP_N in the Staniford model.
- Threshold (*T*): the number of scans that is sent out from an infected node before the node is contained. *T* is the epidemic threshold, and the value of *T* is one or more.
- Tick (*k*): A time unit. For example, the time unit of 1-tick advances when one infection packet is sent out from an infected node.
- Generation (*n*): the infection distance from the infection source ($n \le N$). For example, the number in “3-generation” means the number of grandchildren is three.
- $E_n(k,p,T)$: the expected number of infected *n*-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 enough time passes.

It is very rare that all vulnerable addresses are used within the enterprise network. The address space in such a network is usually only partly used. Moreover, since the scanning malware selects a targeting node probabilistically, an infection packet that is sent out from an infected node does not always reach the vulnerable node within the enterprise network. 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 within the enterprise network.

4.4 Probability of Targeting

An existing scanning worm mainly uses two kinds of target-selection algorithm: (1) the worm selects a target node completely randomly or (2) the worm selects a target node probabilistically according to the local subnet in which the infected node exists. For example, a Sasser worm 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.

$$p = \frac{2}{4}\cdot\left(\frac{N-1}{2^{32}}\right) + \frac{1}{4}\cdot\left(\frac{Na-1}{2^{24}}\right) + \frac{1}{4}\cdot\left(\frac{Nb-1}{2^{16}}\right) \qquad (3)$$

Probability *p* of targeting is obtained according to the composition of the vulnerable host. Thus, p is constant regardless of the number of infections. However, in practice, the more the number of infected nodes is, the fewer the number of nodes that can be infected. Actually, it is thought that the probability of targeting gradually becomes small. We therefore think that the probability of targeting in our model takes the upper bound because it does not become low. We also think that a constant probability *p* is acceptable as long as the number of infected nodes within the enterprise network is fewer than that in the whole address space.

4.5 Expected Number of Infected Nodes

Probability *p* of targeting is used to define the number of 0-generation infected nodes (the infection source) *i* as $E_0(k,p,T)=1$ regardless of the parameters *k*, *p* and *T*.

The 1-generation infected node is the target that the infection source will infect directly. The number of 1-generation infected nodes after *k*-tick is the sum of nodes that the infection source directly infects until *k*-tick. The expected number of 1-generation infected nodes after *k*-tick is therefore calculated as follows.

$$E_1(k,p,T)=\begin{cases}\sum_{i=1}^{k} i\cdot {}_k C_i\cdot p^i\cdot(1-p)^{k-i} & (k<T)\\ \sum_{i=1}^{T} i\cdot {}_T C_i\cdot p^i\cdot(1-p)^{T-i} & (k\geq T)\end{cases}\qquad(4)$$

The number of 2-generation infected nodes does not include the number of the 1-generation infected nodes because the 1-generation infected nodes can not be infected twice. When *k* is larger than *T*, the number of each-generation infected node is calculated from the approximation

$$E_n(k,p,T)=E_1(T,p,T)^n\cdot$$

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

$$\begin{aligned}E(k,p,T)&=\sum_{i=0}^{k}E_i(k,p,T)\\&=\sum_{i=0}^{k}E_1(T,p,T)^i\qquad(5)\end{aligned}$$

After enough time passes (*k* is close to infinity), the total expected number of infected nodes under both probability *p* of targeting and threshold *T* is calculated as follows.

$$\begin{aligned}I(p,T)&=\lim_{k\to\infty}\sum_{i=0}^{k}E_1(T,p,T)^i\\&=\frac{1}{1-E_1(T,p,T)}\quad(\text{if } E_1(T,p,T)<1)\qquad(6)\end{aligned}$$

Note that the value of *I*(*p*,*T*) in the above equation diverges when threshold *T* is much larger. In an actual network, the number of infected nodes finally approaches *N* even if threshold *T* is much larger.

4.6 Upper Bound of *T*

We can get the upper bound of *T* by using Equation (6) in the following steps.

1. Plural values of *I*(*p*,*T*) are calculated from increments of *T* under probability *p*.
2. The upper bound of *T* to satisfy the following equation is obtained.

$$I(p,T)\leq u\qquad(7)$$

Parameter *u* is the upper bound of the expected number of infected nodes that is a maximum finite value or a definite value. Though a candidate for one or more threshold *T* values is derived, we use the upper bound of *T* that satisfies Equation (7).

We can obtain the threshold according to the expected number of infected nodes by changing parameter *u*. For example, the setting *u* = 2 in Equation (7) 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).

5 COMPUTER SIMULATION

We evaluated the expected number of infected nodes under a certain threshold by using a computer simulation. We confirmed that the result from our model precisely corresponds to the result of the computer simulation under the condition $E_1(T,p,T)$ < 1.

Our goal is to evaluate by computer simulation whether an actual worm can be prevented from spreading in a practical enterprise network. In our evaluation, we thus assume the subnet of class-B and that an actual scanning worm has damaged the enterprise network.

We simulate the malware spreading by using a simple Monte Carlo simulator under the condition that the containment software with the threshold is installed in each host. 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 the 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 host is always infected. Also, if the selected address is infected or invulnerable, the state of the host is unchanged even if it receives an infected packet.

Figure 1 shows that the value of *I*(*p*,*T*) in our model fits the result of computer simulation when *T* is less than 70. In the computer simulation, the infection of a Sasser worm in the subnet of class-B of an enterprise network is considered. The first set of experiments we did involved the following selected parameters: the size of the subnet of class-B: 2^{16} , p

(= vP_N) = 0.0125, *N* = 3277 (vulnerable node density: $3277/2^{16}$ = 0.05). The value of *p* is calculated from Equation (3) with *N* = *N*a = *N*b = 3277. We simulated 10,000 runs by varying *T* from 1 to 79 in steps of 1, and plotted the average values.

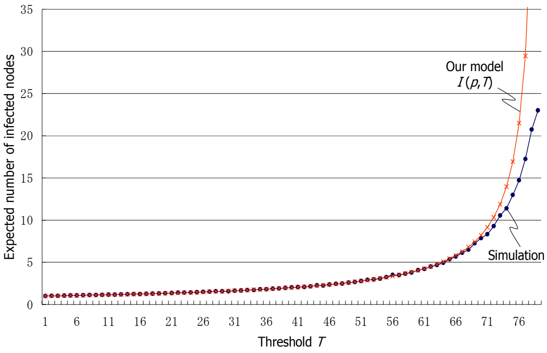

Figure 1: The relation between the result of computer simulation and the result of $I(p,T)$ in our model using a Sasser worm in the subnet of class-B.

In Figure 1, the value of $I(p,T)$ in our model becomes larger than the result of computer simulation when the number of infected nodes becomes large. The probability of targeting in our model is constant for simplicity (refer to section 4.2). On the contrary, in our computer simulation, the probability of targeting decreases as the number of infected hosts increases.

6 DISCUSSION

As mentioned in section 2.2, Staniford's threshold is derived under the condition $T > 1/vP_N$ (T > 1/p). On the other hand, our threshold is derived under the condition $T < 1/vP_N$ (T < 1/p). Note that $T = 1/vP_N$ is a singularity point in both models. In this section, we confirm 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 2 extends the x-axis of Figure 1 and also includes the results from the Staniford model. While Staniford's result is calculated using Equation (2), our threshold is calculated using Equation (7). For the expected number of infected nodes, the Staniford model uses $\alpha \cdot N$ but our model uses $I(p,T)$.

As regards the range for fitting the computer-simulation results in Figure 2, our model is different from Staniford model. In short, while the coverage of the Staniford model is T > 80 (1/0.0125), the coverage of our model is T < 80. In the Staniford model, threshold T can not be calculated when T < 80. The boundary point between the Staniford model and our model is T = 80. The target range of threshold T is clearly divided between the Staniford model and our model. As shown in Figure 1, therefore, our model is suitable for the evaluation of the expected number of infected hosts that suppresses the number of infected nodes to low below the threshold.

Here, we discuss Equation (6) for the explanation about the approximation calculation. Since our model considers generation infection, it must calculate the number of infected hosts up to the number of k-generation infections after k-tick. However, our model is approximated to the calculation of the 1-generation infection like Equation (5). We can therefore easily calculate the expected number of whole infected hosts from only the number of one-generation infections.

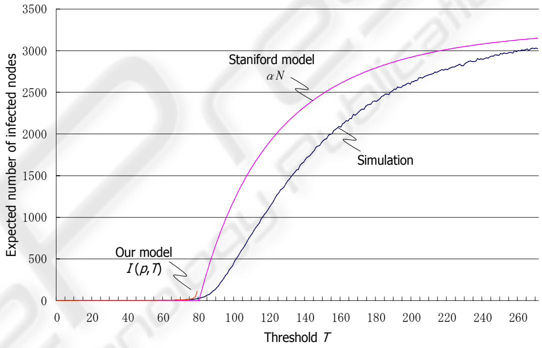

Figure 2: The relation between the result of computer simulation and the results of both αN in the Staniford model and $I(p,T)$ in our model.

Although the condition that $I(p,T)$ is finite becomes $E_1(T,p,T)$ < 1, we have not gotten the condition that satisfies $I(p,T)$ < 2. Here we want to get the condition for $E_1(T,p,T)$ that satisfies $I(p,T)$ < 2 (i.e., the number of new infected nodes is less than one). As a result, the condition $E_1(T,p,T)$ < 1/2 is obtained from Equation (6). This means that the expected number of infected nodes from a single victim must become less than 1/2 in order that the number of infected nodes is suppressed to less than two.

7 CASE STUDY

The vulnerable-node density changes within an enterprise network. It is said that the higher the vulnerable-node density in the network becomes, the easier infection spreading becomes. Our model can determine the relation between the number of subnets, the upper bound of T, and $I(p,T)$. Note that

the same number of vulnerable hosts is distributed within the enterprise network. The relation is described concretely as follows.
Table 1 gives the relation between the number of subnets of class-B and the upper bound of T to prevent a Sasser worm from spreading when $I(\mathrm{p},T) < 2$ is satisfied. We assume that the number of new infected nodes is a constant value like 3277 (as explained in Section 5). The more the number of subnets of class-B increases, the lower the vulnerable node density becomes. As a result, the upper bound of threshold can be set higher. For example, when 3277 hosts with T are distributed within the enterprise network, we set $T = 39$ if there is one subnet of class-B, and we set $T = 79$ if there are two subnets of class-B.

Table 1: The relation between the number of subnets of class-B (the number of vulnerable nodes is constant at 3277) and the upper bound of T to prevent a Sasser worm when $I(p,T) < 2$.

Number of subnets of class-B	N	Number of vulnerable hosts in each subnet of class-B	Upper bound of T
1	3277	3277	39
2	3277	1638	79
3	3277	1092	118
4	3277	819	157
5	3277	655	196

From the viewpoint of preventing infection from spreading, it is important to expand the number of subnets and to lower the density of the host when the same host is put in an enterprise network. We quantitatively show how much threshold we should be set according to the number of class-B subnets. If the upper bound of the threshold can be raised while suppressing worm spreading, the times for both detection and containment can be increased. Accordingly, the accuracy of detection can be expected to be improved. It is a big contribution to the countermeasure of scanning malware in the enterprise network to know how much high the threshold we should be set by according to the number of subnets.

8 SUMMARY

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 result of computer simulation of malware spreading when $E_1(T,p,T) < 1$ is satisfied.

Moreover, we clarified the relation between the number of subnets in an enterprise network and the upper bound of the threshold when the same number of hosts is distributed within the network. For example, when 3277 hosts are distributed within the network, we set $T = 39$ if there is one class-B subnet, and we set $T = 79$ if there is two class-B subnets.

In a practical enterprise network, it is important that a suitable countermeasure is executed in the early stages of infection. 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

Barford, P., Yegneswaran, V., 2006. An Inside Look at Botnets. *Special Workshop on Malware Detection, Advances in Information Security*.

Nikoloski, Z., Deo, N., Kucera, L., 2006. Correlation Model of Worm Propagation on Scale-Free Networks. *Complexus*, 2006(3):169-182.

Chen, Z., Gao, L., Kwiat, K., 2003. Modeling the Spread of Active Worms. In *Proceedings of IEEE INFOCOM*.

Staniford, S., 2004. Containment of Scanning Worms in Enterprise Networks. *Journal of Computer Security*.

Moore, D., Shannon, C., Voelker, G. M., Savage, S., 2003. Internet Quarantine: Requirements for Containing Self-Propagating Code, In *Proceedings of IEEE INFOCOM*.

Zou, C. C., Gao, L., Gong, W., Towsley, D., 2003. Monitoring and Early Warning for Internet Worms. In *Proceedings of the 10th ACM Conference on Computer and Communication Security*, pages 190-199. ACM Press.

Williamson, M. M., 2002. Throttling Viruses: Restricting propagation to defeat malicious mobile code. In *Proceedings of the 18th Annual Computer Security Applications Conference*.

Whyte, D., Kranakis, E., Oorschot, P. C., 2005. DNS-based Detection of Scanning Worms in an Enterprise Network. In *Proceedings of the 12th Annual Network and Distributed System Security Symposium*.

Whyte, D., Oorschot P. C., Kranakis, E., 2005. Detecting Intra-enterprise Scanning Worms based on Address Resolution. In *Proceedings of the 21st Annual Computer Security Applications Conference*.

Bakos, G., Berk, V. H., 2002. Early detection of Internet worm activity by metering ICMP destination unreachable messages. In *Proceedings of the SPIE Aerosense*.

Weaver, N., Staniford, S., Paxson, V., 2004. Very Fast Containment of Scanning Worms. In *Proceedings of the13th USENIX Security Symposium*.

Jung, J. Paxson, V., Berger, A. W., Balakrishnan, H., 2004. Fast portscan detection using sequential hypothesis testing. In *Proceedings of the IEEE Symposium on Security and Privacy*.

Schechter, S., Jung, J., Berger, A. W., 2004. Fast Detection of Scanning Worm Infections. In *Proceedings of the 7th International Symposium on Recent Advances in Intrusion Detection*.

Omote, K., Torii, S., 2003. A Detection Method of Worms's Random Scanning. In *Proceedings of the CSS2003*. (Japanese).