Modeling of Computer Virus Spread and Its Application to Defense

Jun Shitozawa s1090109 Supervised by Hiroshi Toyoizumi

Abstract

The purpose of this paper is to model a computer virus spread and evaluate content filtering and IP address blacklisting with a key parameter of the reaction time R. We model the Sasser worm by using the Pure Birth process in this paper. Although our results require a short reaction time, this paper is useful to obviate the outbreak of the new worms having high reproduction rate λ.

1 Introduction

In recent years, new computer worms are being created at a rapid pace with around 5 new computer worms per a day. Furthermore, the speed at which the new computer worms spread is amazing. For example, Symantec [5] received 12041 notifications of an infection by Sasser.B in 7 days.

Computer worms are a kind of computer virus. The worms make a copy of themselves and spread throughout the Internet. New types of worms, including MyDoom, Netsky, Sasser, have appeared. One characteristic of these worms is that they are able to infect PCs with security vulnerabilities. The Sasser worm targets Windows 2000 and Windows XP machines by scanning many hosts and attempting to exploit the LSASS vulnerability [8].

The new computer worms such as Sasser, which have the ability to infect speedily, are able to self-propagate rapidly while security measures against new computer worms are lacking. It is necessary to deal with new computer worms at an early stage to prevent them from increasing rapidly. In this paper, two systems "Content Filtering" and "IP Address Blacklisting" [1], are discussed. Both are systems to avoid computer worm outbreaks at an early stage.

The purpose of this research is to model a computer virus spread and evaluate two systems. We have to examine the effect of two containment systems by using the worms having high reproduction rate such as Sasser because new computer worms can spread worldwide speedily, using random number to generate IP addresses, in a short time. Therefore, Sasser.B is modeled mathematically by using a "Pure Birth Process" [3], which is the process of ecosystems regarding only births.

2 Two Systems

2.1 Content Filtering

Content filtering is a containment system that has a database of content signatures known to represent particular worms. Packets containing one of these signatures are dropped when a containment system member receives the packets. This containment system is able to stop computer worm outbreaks immediately when the systems obtain information of content signatures. However, it takes too much time to create content signatures, and this system has no effect on polymorphic worms [10]. A polymorphic worm is one whose code is transformed regularly, so no single signature identifies it.

2.2 The IP Address Blacklisting

IP address blacklisting is a containment system that has a list of IP addresses that have been identified as being infected. Packets arriving from one of these addresses are dropped when received by a member of the containment system. It does not take long to implement this system because it does not require the worm to be identified. However, it cannot stop computer worm outbreaks completely in its first attempt, and must be updated continuously to reflect newly infected hosts. In addition, it is in danger of denying service to normal nodes because of wrong detection.

Figure 1 shows the two systems.

2.3 The Reaction Time

In using the containment systems, we model reaction time as follows: The first "seed" host is infected at time 0 and begins to probe randomly. If a host is infected at time t, we assume that all susceptible hosts are notified of this fact at time $t+R$, where R specifies the reaction time of the system. When using content filtering, this notification simply includes the signature of the worm, and all worm probes from any host are ignored after time $t+R$. In the case of IP address blacklisting, this notification simply consists of the IP address of the infected host. Although probes from one of the list will be ignored, the infected hosts can infect other hosts for R hours.

Figure 2 shows the influence of the reaction time on each containment system.

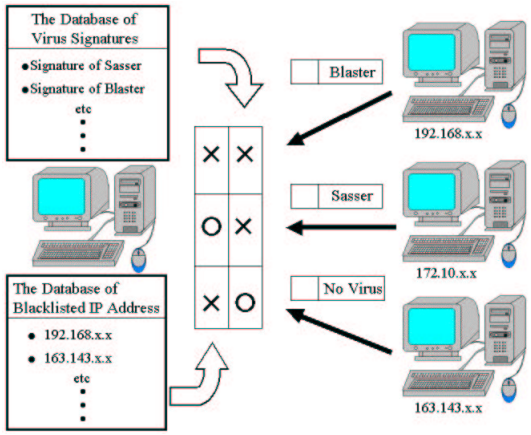

Figure 1: The containment system of content filtering and IP address blacklisting drop the packets from one of the signatures or IP addresses in each database.

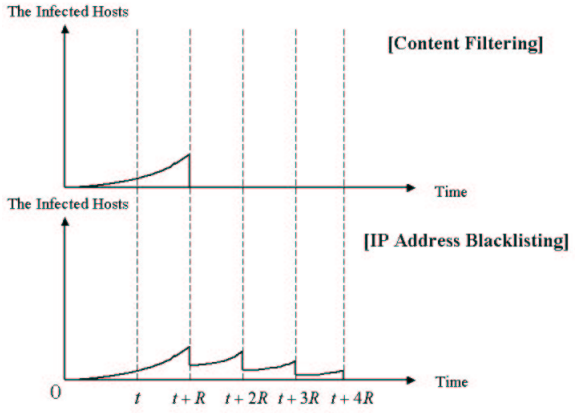

Figure 2: The Reaction Time *R* in Each Case.

3 Sample Virus

3.1 Sasser

The sample virus is called ”Sasser.B”, which was discovered on May 1, 2004. On the Symantec web page, it is called ”W32.Sasser.B.Worm” and a variant of ”W32.Sasser.Worm. It attempts to exploit the LSASS vulnerability described in Microsoft Security Bulletin MS04-011. This worm spreads by scanning randomly selected IP addresses for vulnerable systems. Table 1 expresses other information.

Type	Worm
Infection Length	15,872 bytes
Systems Affected	Windows 2000, Windows XP

Table 1: Information of Sasser.B

3.1.1 Destination of Probe Packet

The Sasser.B worm generates another IP address, based on the IP address retrieved from the infected computer. The destination of generated IP addresses are shown in Table 2. The damage by the virus spreads worldwide because the IP addresses are generated with random number.

The Random Part of IP Addresses	The Probabilities
All Octets	52%
The Last Three Octets	23%
The Last Two Octets	25%

Table 2: The Generated IP Addresses

3.1.2 Penetraion

What the Sasser.B worm does to infect other hosts is as follows:

1. Starts an FTP server on TCP port 5554.
2. Generates the IP addresses according to Table 2.
3. Connects to the host of generated IP address on TCP port 445 to know whether a remote computer is online.
4. Sends shell code, which causes it to open a remote shell on TCP port 9996.
5. Uses the shell on the remote computer to reconnect to the infected computer’s FTP server, running on TCP port 5554, and retrieves a copy of the worm.
6. Repeats from 1.

The Lsass.exe process crashes after the worm exploits the Windows LSASS vulnerability. Windows displays the alert and shut down the system in one minute.

4 The Experimental Data

4.1 The Tested Network

To obtain the data of Sasser.B worm, the worm was executed in an experimental network. The purpose of the experiment is to obtain the mean number of packets the worm throws to the network per hour.

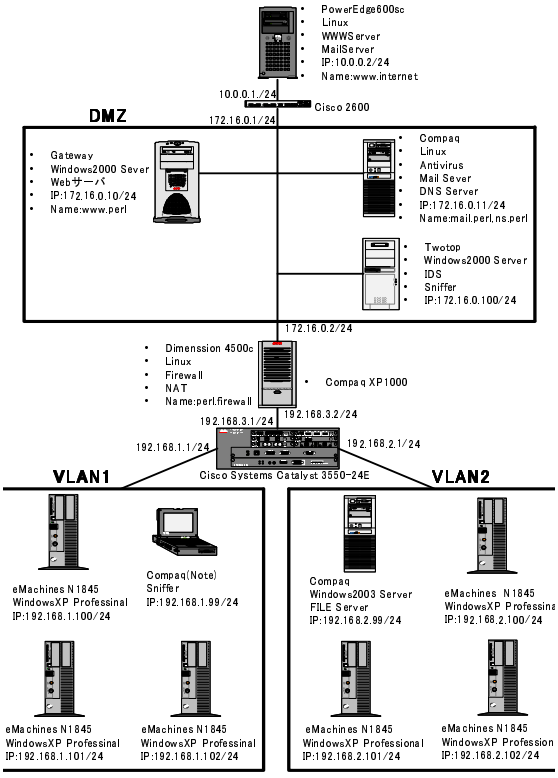

Figure 3: The Experiment Network

4.2 Laboratory Experiment of Sasser.B

In the experiment, one Sasser.B worm was executed on one PC, and then we observed the packets on the neighbor PC connected to the infected PC with hub. To capture the packets, we used a Sniffer [4] software. After capturing, we obtained the mean number of packets per hour from 3000 packets of data. One Sasser.B worm sends about 69,089 packets per hour. Furthermore, we found that Sasser.B worm sends packets to the same IP address three times. A short interval occurs after 128 packets are sent to different IP addresses. Next, the worm sends the 128 packets to the same IP address twice. If we assume that three times sending are one set, the worm sends one set per about 20.8591 second.

Information	Numeric
Average packets per hour	69,089
The number of IP address in one set	128
The time to send one set	20.8591(s)

Table 3: Data of Sasser.B

5 The Modeling of Sasser.B

5.1 The Poisson Process

In this section, to investigate Internet attacks by Sasser worms, we obtained other data with a router connected to the Internet. The router saved information of all transmissions to it. This information included the time, the attacked port numbers and the attacking IP addresses.

In the case of Sasser worms, since they try to connect on the TCP port 445 first (See Section 3.1.2), we would extract only Sasser worms attack data. Although Sasser worm has four varieties of worms, we can deal with those data in common with the Sasser.B worm because they do not have points of difference for infection, and there are many Sasser.B worms in the Internet. Therefore, we assumed that the Sasser worm sends to the same IP address three times (See Section 4.2) and we considered three attacks as one attack. After that, we obtained the interval time of attacking based on information of the attacked time, and consider that the interval time is T.

Here, let T be the interval time of the Poisson process with its rate α. Also, let $P\{T \leq t\}$ be the probability that T is less than or equal to t, and $P_0(t)$ denotes the probability that there is no attack up to time t. Then,

$$P\{T \leq t\} = 1 - P_0(t) = 1 - e^{-\alpha t}.$$

Now, if we let $f(t)$ be the probability that each T exists between $n\Delta t$ and $(n+1)\Delta t$, we obtain the following equation:

$$f(t) = P\{n\Delta t < T \leq (n+1)\Delta t\} = e^{-\alpha t}(1 - e^{-\alpha \Delta t}), \quad (1)$$

where

$$\frac{1}{\alpha} = E[T].$$

Figure 4 indicates the relation of $f(t)$ and T. The numeric of Δt means the interval between one bar and next bar.

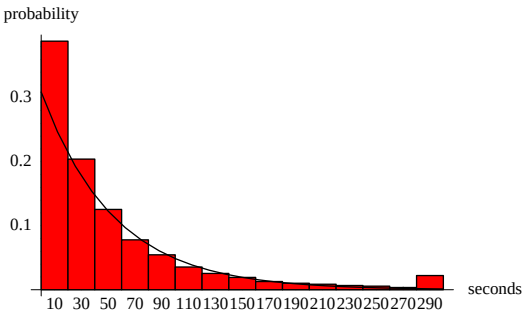

Figure 4: The line and bar chart indicate $f(t)$ and histogram of T each where Δt is 20 in this graph.

Similarly, Figure 5 shows the relation of $f(t)$ and T in the case of "Blaster worms". However, we would extract only Blaster worms attack data because Blaster worms try to connect on the TCP port 135 first [9]. Moreover, we considered two or three attacks as one attack because Blaster worms also sent packets to the same IP addresses twice or three times.

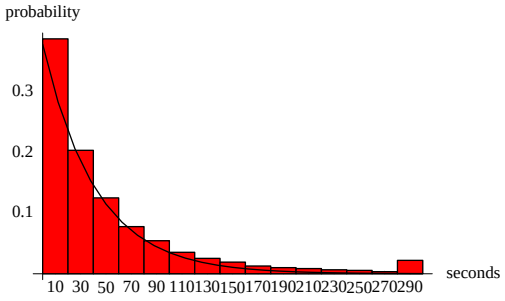

Figure 5: This graph indicates $f(t)$ and histogram of T in the case of Blaster worms.

Figure 4 and Figure 5 indicate that the Internet attacks by computer worms conform to a Poisson process. Therefore, the Internet attacks by Sasser worms occur at a LAN with rate α. Thus, if one LAN has n vulnerable hosts, where n is the counting number, the attacks occur at the LAN with rate $n\alpha$.

5.2 The Pure Birth Model

To model the Sasser worm in a LAN with the stochastic pure birth model, let us make the assumptions that:

- the worms do not die.
- they develop independently.
- the birth rate λ is the same for all worms, and does not change with time.
- the LAN is enough large.

Here, let $N(t)$ be the number of the hosts infected in a LAN at time t, and n_0 means $N(0)$, then, $\{N(t), t \geq 0\}$ is a pure birth process with rate λ. Thus, we obtain the following equation:

$$P\{N(t) = n\} = \begin{pmatrix} n-1 \\ n_0 - 1 \end{pmatrix} e^{-\lambda n_0 t}(1 - e^{-\lambda t})^{n-n_0}. \qquad (2)$$

Also, if we assume that one Sasser worm comes into the LAN at time t, we can consider n_0 as 1. Therefore, we obtain

$$P\{N(t) = n\} = e^{-\lambda t}(1 - e^{-\lambda t})^{n-1}. \qquad (3)$$

Equation (3) denotes the probability that the number of PCs infected by Sasser worms is equal to n, which is a constant number of infected hosts, at time t.

5.2.1 The Infection Rate λ

In this section, we will obtain the infection rate λ to see the relation between the probabilities and time with the equation (3). From the Section 4.2, the Sasser worm sends 128 packets per about 20.8591 seconds. Thus, it sends about 22091 packets per hour, and we let the value be S. Furthermore, to determine the infection rate λ, we have to consider the following parameters:

- P_{usedIP}: The rate that IP addresses are used.
- $P_{XP/2000}$: The rate that the OS of the hosts are Windows XP or Windows 2000.
- P_v: The rate that the users are not quite establishing security measures.

According to CNET Japan [6] and ITmedia [7], the above parameters are as follows:

- P_{usedIP} is 5.8%.
- $P_{XP/2000}$ is 58%.
- P_v is 33.6%.

Then, if we assume that the LAN is enough large such as Class B, all we need to consider the only packets sent to the IP addresses generated at the last two octets random. Thus, from Table 2, the infection rate λ per hour is given by the following equation:

$$\lambda = S \times 0.25 \times P_{usedIP} \times P_{XP/2000} \times P_v. \qquad (4)$$

5.2.2 The Distribution of Sasser

In this section, we can see the graph of the Sasser worm with the equation (3) in Figure 6.

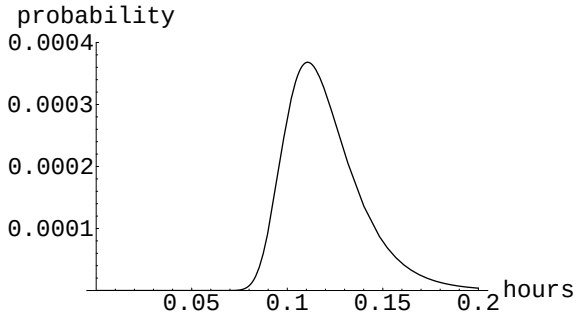

Figure 6: This graph shows $P\{N(t) = 1000\}$.

From the Figure 6, we could model the Sasser worm in the LAN with the pure birth model. Therefore, we can evaluate content filtering and IP address blacklisting by using the model in the next section.

6 Numerical Comparison

In this section, we assume that one Sasser worm comes into a LAN at time t, and Sasser worms grow in the LAN by conforming to a Pure Birth process with rate λ after time t.

6.1 Content Filtering

The containment systems of content filtering stop worms from infecting immediately after time $t + R$. Therefore, all we need to do is to consider that the worms infect other hosts by conforming to the equation (3) for only $t + R$ hours. Then, if we let $m(t)$ be the average number of the hosts infected for time t based on the equation (3), $m(t + R)$ is given by the following equation:

$$m(t + R) = e^{\lambda(t+R)}, \tag{5}$$

The result of the equation (5) can be seen in Figure 7, which shows the effectual reaction time R of the Sasser worm.

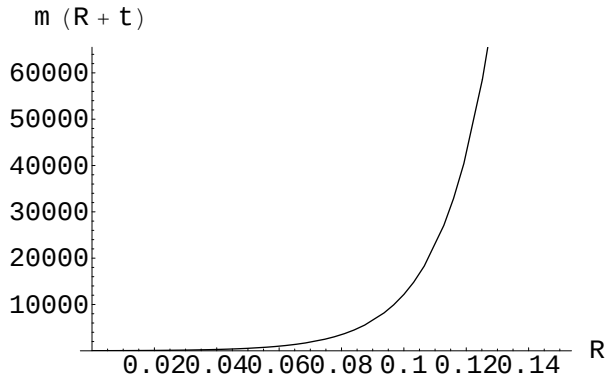

Figure 7: This graph indicates the relation between the average number of the hosts infected by the Sasser worm and reaction time R(hours) where $t = \log 2/n\alpha$.

From Figure 7, we can say that the containment system of content filtering can obviate the Sasser worm outbreak if the reaction time R is not longer than 0.05 hours.

6.2 IP Address Blacklisting

The containment systems of IP address blacklisting can drop the packets from the IP address of the first host infected at time t after $t + R$ hours. Therefore, the worms can infect other hosts for R hours. After an additional R hours, the containment systems add IP addresses of the hosts infected between t and $t + R$ to a list at time $t + 2R$. Similarly, the systems repeat to add IP addresses to a list every R hours after time t, and the number of n_0 is constantly changed at the same time. Then, if we consider adding IP addresses to a list as deaths of the worms, the average number of the infected hosts by the Sasser worm is given by the following equation:

$$m(t + iR) = n_0(i) \times e^{\lambda R} \tag{6}$$

where

$$n_0(i) = (e^{\lambda R} - 1)^{i-1} \qquad (i = 1, 2, 3, \ldots).$$

Figure 8 shows the relation between the number of the hosts infected by Sasser worm and reaction time R with the equation (6) where $t = \log 2/n\alpha$.

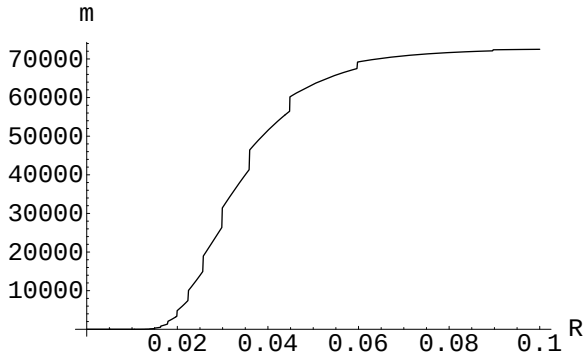

Figure 8: This graph indicates the relation between the average number of the hosts infected after 0.23 hours and the reaction time R(hours).

We need more detail information to obtain the effectual reaction time. Then, the Table 4 shows $m(t + R)$, $m(t + 2R)$ and the reaction time R.

$m(t + R)$	$m(t + 2R)$	R (hours)
12	135	0.04
6	35	0.03
3	8	0.02
2	2	0.01

Table 4: The Detail Information.

In order to obviate the outbreak, the reaction time R should be shorter than the time when $m(t + 2R)$ is not greater than $m(t + R)$. From Table 4, we can say that the containment system of IP address blacklisting can obviate Sasser worm outbreak if the reaction time R is shorter than 0.01 hours.

7 Conclusion and Future Work

From Figure 7 and Table 4, we have to implement content filtering in 2.4 minutes to obviate the Sasser worm outbreak. Also, IP address blacklisting can obviate the Sasser worm outbreak if the reaction time R is not greater than 36 seconds. However, our results require the short reaction time R. Although it will be impossible to implement the containment systems in 2.4 minutes or 36 seconds, content filtering and IP address blacklisting are effective to prevent the worms spreading in the Network. However, our model may be different from the real growth of the Sasser worm because the results in this paper reflect the worst case. Thus, the effectual reaction time will be longer. The strong trigger for the difference would be what we did not consider whether the vulnerability and infected hosts are powered on or not. In reality, there are many hosts powered on and off in the Internet. If we consider this factor, the infection rate λ will be lower because the vulnerability hosts powered off cannot be infected, and worms cannot act for infection if the infected hosts are powered off. Furthermore, when we implement the containment systems in the Internet, the effect of the containment systems will be better if we implement the systems at the effective place, for example, not at the hosts of customers but at the ISPs. Then, the systems would be able to avoid computer worm outbreaks if we could have more time to implement the containment system.

Acknowledgment

I would like to thank Prof. H. Toyoizumi for his advise. I also thank the members of the Performance Evaluation Laboratory for their help in this research.

References

[1] David Moore, Colleen Shannon, Geoffrey M. Voelker, and Stefan Savage, Internet Quarantine: Requirements for Containing Self-Propagating Code,
2003

[2] Sheldon M.Ross, Applied Probability Models with Optimization Applications

[3] Eric Renshaw, Modeling Biological Populations in Space and Time, pages 1-27

[4] Sniffer Technologies,
http://www.toyo.co.jp/sniffer/

[5] Symantec,
http://www.symantec.co.jp/

[6] CNET Japan,
http://japan.cnet.com/

[7] ITmedia,
http://www.itmedia.co.jp/

[8] Microsoft Security Bulletin MS04-011,
http://www.microsoft.com/technet/security/bulletin/MS04-011.mspx/ *August 2004*

[9] Tatehiro Kaiwa, Optimization of Blaster worms by Stochastic Modeling, University of Aizu, Graduation Thesis, *March 2004*

[10] Eugene H. Spafford, Computer Viruses as Artificial Life, Purdue University