Proposal for the experimental environment for Network Worm infection

Masato Terada†, Shingo Takada
Graduate School of Science and Technology,
Keio University.
3-14-1 Hiyoshi, Kohoku, Yokohama, Kanagawa 223-8522, Japan

And

Norihisa Doi ‡
Graduate School of Science and Engineering,
Chuo University.
1-13-27 Kasuga, Bunkyo-ku, Tokyo 112-8551, Japan

Abstract

Code analysis and simulation of network worm infection are useful methods to evaluate how it spreads and its effects [1]. But a bug in infection algorithm or the way of implementing a random number generator etc. affects the retrieval behavior of network worm infection. It is important to evaluate the retrieval behavior of network worm infection in an experimental environment for complementing code analysis. This paper describes a prototype of experimental environment for network worm infection and actual data on network worm infection. The purpose of experimental environment is to investigate retrieval behavior and infection mechanisms in network worm behavior. For example, there are a mapping of retrieved IP addresses and a ratio of IP addresses retrieved and port numbers used by network worms. Also we implemented a prototype system to show the validity of our approach.

Keywords: Network Security, Worm, Infection, Retrieval behavior

1. Introduction

In 2004, Sasser.A, Sasser.B and Sasser.C were discovered on April 30, May 1 and May 2, respectively. The Sasser worm selects retrieval IP addresses and tries to infect selected targets, like Code Red, Nimda and Blaster. The address block ratio of IP addresses retrieved by Sasser, according to the code analysis, is shown in Table 1 [2]. A bug in IP address selection algorithm or the way of implementing a random number generator etc. affects the retrieval behavior of network worm infection. In other words, these factors affect the trends in infection spread. Then it is important to evaluate the retrieval behavior of network worm infection in experimental environment for complementing the code analysis. But there is no actual, measured data on network worm infection for public use. Moreover; to prevent network worm infection, an organization should not rely solely on the outside information and should have some information gathering method of its own, such as experimental environment to evaluate network worm infection, to plan countermeasures. We propose "an experimental environment for retrieval and infection characteristics in network worm behavior" to improve information gathering concerning network worm activities. We also present actual data on infection activities about network worms such as Code Red, Nimda, Slammer, Blaster and Sasser obtained in our experimental environment.

Table 1 Ratio of IP addresses retrieved by Sasser.

The selection of potential target IP addresses	Ratio
The same first two octets	25%
The same first octet	23%
Others	52%

2. Experimental environment for the Network Worm infection behavior

There are the following requirements to construct an experimental environment.

- It doesn't need to use a special device, and is a small-scale, readily-available system with just sufficient hardware/software.
- It provides information for building countermeasures to prevent network worm infection. In other words, we can gather information on the address block ratio of IP addresses retrieved and the port numbers used by network worms.
- It makes it possible to efficiently verify infection behavior of network worms.

We propose two types of experimental environment -- "the experimental environment for retrieval behavior" and "the experimental environment for infection behavior" to satisfy above requirements.

2.1 Experimental environment for retrieval behavior

The purpose of this environment is to gather the address block ratio of IP addresses retrieved by network worm infection. The experimental environment is shown in Figure 1. The infected PC has a virtual machine environment and malicious software was invoked on this virtual machine. The monitoring PC has some functions for monitoring worm-infected packets. A packet monitoring function captures infected packets and a traffic counter function counts the number of those packets and makes summarization. Our prototype uses "tethereal" as packet

†) Systems Development Laboratory, Hitachi Ltd.
890, Kashimada, Saiwa-ku, Kawasaki, 212-8567 Japan.
‡) Graduate School of Science and Technology, Keio University

monitoring function and a traffic counter is implemented by the script language Perl.

Web interface provides the following information about network worm infection.

(a) The mapping of retrieved IP addresses (upper left in Figure 2)
(b) The distribution of addresses with the same first octet (bottom left in Figure 2)
(c) The distribution of addresses with the same first two octets (bottom right in Figure 2)
(d) The address block ratio of IP addresses retrieved (upper right in Figure 2)

The characteristics of this environment are the following:

- The experimental environment can be built with two PCs.
- In this environment, loss of ability to observe activities is limited to only three IP addresses (2 broadcast addresses and 1 infected PC IP address) at a maximum even when a network worm selects an IP address from the same sub network to which the monitoring PC belongs.
- The infected PC sends TCP SYN packets, but it cannot establish a TCP connection because the monitoring PC discards all TCP SYN packets.

2.2 Experimental environment for infection behavior

The purpose of this environment is to gather information about infection behavior of network worms. Especially, information on the port numbers used in network worm infection is our target. The experimental environment is shown in Figure 3. The global Internet address space currently offers 32-bit worth of unique host addresses, or a theoretical maximum of 2^32=4,294,967,296 hosts. The probability that the IP address of our target PC is chosen by network worms by chance is very low. This environment has an IP address translation function to resolve this problem and make efficient verification possible. Our prototype uses “Linux iptables DNAT (Destination Network Address Translation)” as IP address translation function. The DNAT mechanism is shown in Figure 4. The monitoring PC receives any packets on the eth0 interface and changes the destination IP address of those packets from a random IP address (=nnn.nnn.nnn.nnn) to the target IP address (=y.y.y.1). Then it sends the packets to the targeted PC.

In addition, this environment has a flow analysis function, which extracts the following items from the packets captured by the packet monitoring function to build countermeasures.

- The destination port number used in network worm infection.
- The destination port number series and its frequency.

The flow analysis function has two steps: extraction of the packet of interest that has made the first appearance (“the first appearance packet”) and then determination of the destination port number series.

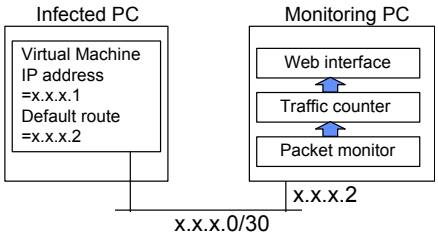


Figure 1 Prototype for retrieval behavior of network worm.

(a) Extraction of the first appearance packet

The flow analysis function divides packets into groups according to the source/destination IP address pairs and the source/destination port number pairs. Next, it extracts the first appearance packet in each group and determines the destination port number used in network worm infection. In case of TCP communications, it usually extracts TCP SYN packets.

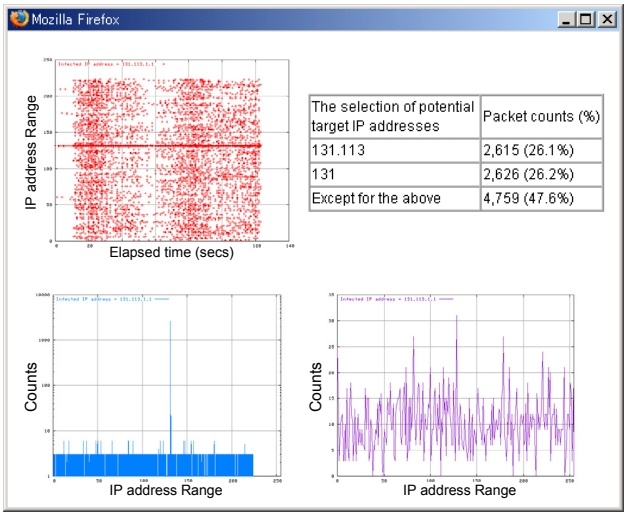


Figure 2 Web interface of prototype for retrieval behavior.

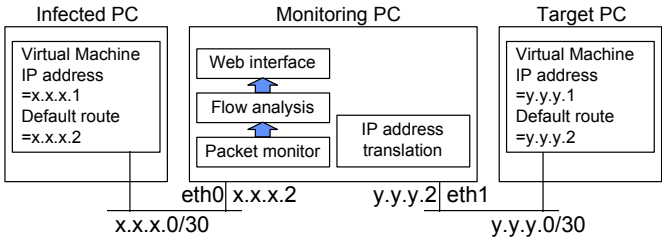


Figure 3 Prototype for infection behavior of network worm

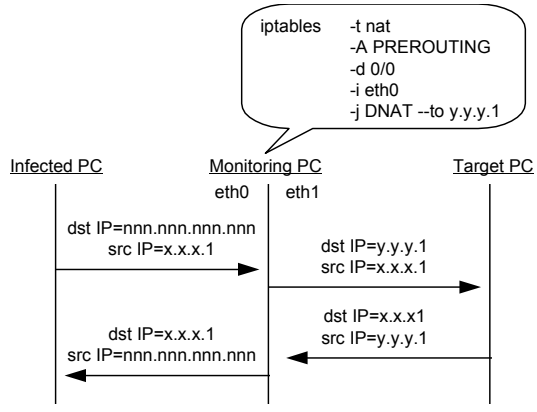


Figure 4 Destination IP address translation by DNAT.

(b) Determination of the destination port number series
The flow analysis divides the first appearance packets extracted in (a) into groups according to the source/destination IP address pairs yet again. Next, it determines the destination port number series in each group. Then it summarizes the extracted destination port number series.

The flow analysis example of Sasser.C is shown in Figure 5. In this case, the IP address of the infected PC is 131.113.1.1 and that of the target PC is 192.168.1.1. In Figure 5, Packet No (see the bottom-right cell in the table on the top) is the serial number of the packet since the capturing started and Elapsed Time indicates how long it took for the first appearance packet in that group made its appearance (0 being the base time when the retrieval activity started). As seen in Figure 5, the TCP connection using the destination port number series (445/TCP, 9996/TCP) is counted 693 times and the first appearance packet of this series is the 1st packet. The TCP connection using the destination port number series (5554/TCP, 1033/TCP) is counted once and the first appearance packet of this series is the 236th packet. The network worm infection process is completed in about 3 seconds. These results mean that the infection behavior of Sasser.C could be classified into two series of port numbers {445/TCP, 9996/TCP} and {5554/TCP, 1033/TCP}.

3. Verification by using the experimental environment

3.1 Retrieval behavior of Network Worm

This section describes the retrieval behavior of Code Red, Nimda, Blaster, Slammer and Sasser verified by using the experimental environment (Figure 1). The infected PC is a Dell PowerEdge1400 PemtiumIII (Memory 256MB) with Microsoft Japanese Windows 2000 Server Service Pack 4. The monitoring PC is an IBM Thinkpad 2609-93J Pentium III (Memory 192MB) with Red Hat Linux 7.3. These PCs are connected via 100Mbps Ethernet. A Virtual Machine on the infected PC is Microsoft Japanese Windows without Service Pack on a VMware Workstation 4.0. Also, Code Red III, Nimda.E and Blaster were verified on Windows 2000 Server. Slammer was verified on Windows 2000 Server with Microsoft SQL Server. Sasser.B and Sasser.C were verified on Windows XP Professional.

(1) Code Red III
In March 2003, Code Red III worm was discovered [3]. Code Red III worm is a new variant of Code Red II and almost identical to Code Red II, with just two bytes being altered. Code Red II stopped spreading at the end of 2002 - the change made in Code Red III changes this and enables it to spread forever (till the end of the year 34952). The retrieval behavior of Code Red III toward the targeted vulnerable Web servers (80/tcp) is shown in Table 2. And the mapping of retrieved IP addresses is shown in Figure 6. The Y-axis represents the IP address range (0.0.0.0 - 255.255.255.255) and the X-axis represents the elapsed time since Code Red III infection was activated. The resulting data of this experiment and data from the code analysis show almost the same value.
(2) Nimda.E
In October 2001, Nimda.E worm was discovered. It is a variant of Nimda that has four modes of propagation: through email, through network shared drives, through vulnerable Web servers, and through file infection. The characteristics of retrieval behavior toward the targeted vulnerable Web servers (80/tcp) are shown in Table 3. And the mapping of retrieved IP addresses is shown in Figure 7. The Y-axis represents the IP address range (0.0.0.0 - 255.255.255.255) and the X-axis represents the elapsed time since the Nimda.E infection was activated. The resulting data of this experiment and data from the code analysis are different because the retrieval behavior of Nimda.E has the periodicity as seen in Figure 7.
(3) Blaster
The Blaster worm was discovered on August 10, 2003 [4]. The number of hosts the worm infected over several weeks is unknown, but it reportedly infected approximately 45,000 hosts in the first four days after its release. Blaster is a network worm and has different characteristics from Code Red, Nimda and Sasser in terms of retrieval behavior. Selection of target IP addresses by Code Red, Nimda and Sasser depends on the IP address of an infected PC at each recursive infection activity.

Mozilla Firefox

Total packet counts	53782
Infection activity start time	1 0.000000 Packet No, Elapsed Time

The series of port numbers	Counts	The logs of first appearance packet Line1: Packet No, Elapsed time, src IP address, dst IP address, Protocol, src PT number, dst PT number Line2: Flag information
445/TCP	2	602 7.882561 131.113.1.1 149.144.49.64 TCP 1096 > 445 [SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=1460
445/TCP 9996/TCP	693	1 0.000000 131.113.1.1 131.113.98.125 TCP 1053 > 445 [SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=1460 169 2.262588 131.113.1.1 131.113.98.125 TCP 1065 > 9996 [SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=1460
5554/TCP 1033/TCP	1	236 2.867330 192.168.1.1 131.113.1.1 TCP 1032 > 5554 [SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=1460 248 2.895472 131.113.1.1 192.168.1.1 TCP 1069 > 1033 [SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=1460

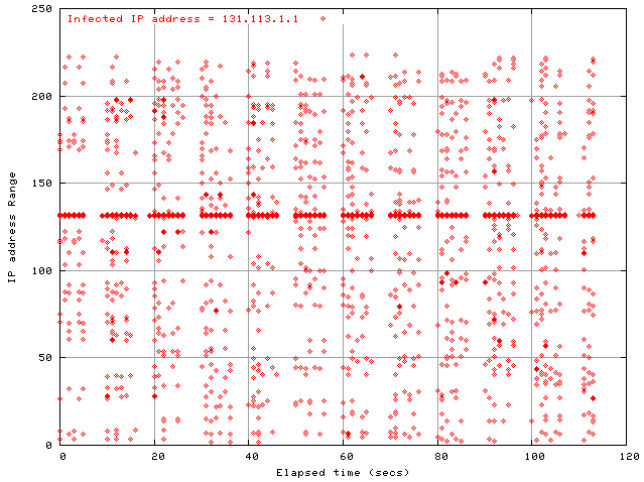
Done

Figure 5 Example of the Flow analysis function.

Table 2 Ratio of IP addresses retrieved by Code Red III.

Selection of IP address	Experiment Data	Code Analysis
The same first two octets	37.7%	37.5%
The same first octet	50.8%	50.0%
Others	11.5%	12.5%

The average of 3 trials (10,000 packets/trial).



Time: from started observation to counted 10,000 pkts.
Figure 6 Map of retrieved IP addresses by Code Red III.

Blaster selects an initial target IP address with which it starts infection activity in a similar way, but then, instead of repeating the process, it increments the retrieval IP address range one by one from there sweeping IP addresses methodically. The mapping of retrieved IP addresses of the targeted vulnerable hosts (135/tcp) is shown in Figure 8. The Y-axis represents the IP address range (153.75.20 - 153.75.50) and the X-axis represents the elapsed time since the Blaster infection was activated.

(4) Slammer

The Slammer worm was discovered on January 25, 2003, and infected approximately 75,000 machines running Microsoft SQL Server. Most vulnerable machines were infected within ten minutes after the release of the worm [5]. According to the code analysis, Slammer uses a single UDP packet (1434/udp) and selects an IP address randomly without any dependence on the host IP address in its attempt to exploit the buffer overflow vulnerability in Microsoft SQL Server. It shows different behavior from other TCP based network worms. And the mapping of retrieved IP addresses is shown in Figure 9. The Y-axis represents the IP address range (0.0.0.0 - 255.255.255.255) and the X-axis represents the elapsed time since the Slammer infection was activated.

(5) Sasser.B

In May 2004, the Sasser.B worm was discovered. The Sasser.B attempts to exploit the Local Security Authority Subsystem Service (LSASS) vulnerability. The retrieval behavior is shown in Table 4. The resulting data of this experiment and data from the code analysis are a little different. And the mapping of retrieved IP addresses of the targeted vulnerable hosts (445/tcp) is shown in Figure 10. The Y-axis represents the IP address range (0.0.0.0 - 255.255.255.255) and the X-axis represents the elapsed time since the Sasser.B infection was activated.

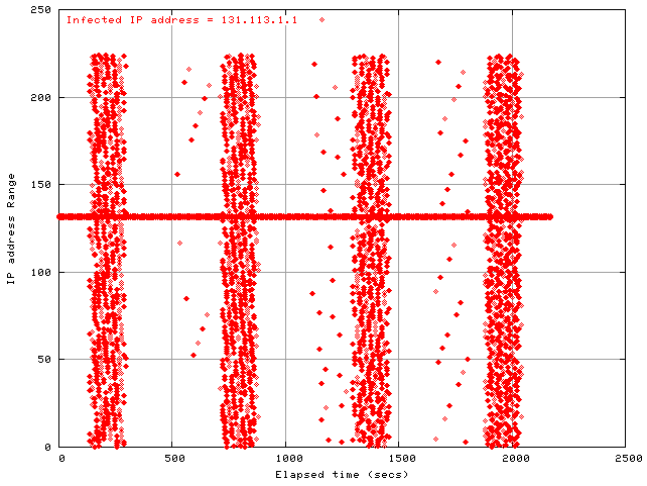
(6) Sasser.C

In May 2004, the Sasser.C worm was discovered. The retrieval behavior is shown in Table 5.

Table 3 Ratio of IP addresses retrieved by Nimda.E.

Selection of IP address	Experiment Data	Code Analysis
The same first two octets	50.9%	50%
The same first octet	38.8%	25%
Others	10.3%	25%

The average of 9 trials (10,000 packets/trial).



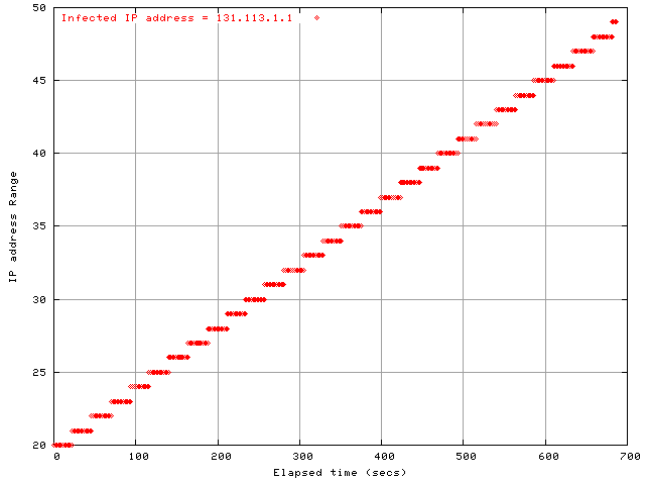
Time: from started obs. to counted 51,261 pkts.

Figure 7 Map of retrieved IP addresses by Nimda.E.

The resulting data of this experiment and data from the code analysis are a little different, just like Sasser.B. And the mapping of retrieved IP addresses to the targeted vulnerable hosts (445/tcp) is shown in Figure 11. The Y-axis represents the IP address range (0.0.0.0 - 255.255.255.255) and the X-axis represents the elapsed time since the Sasser.C infection was activated.

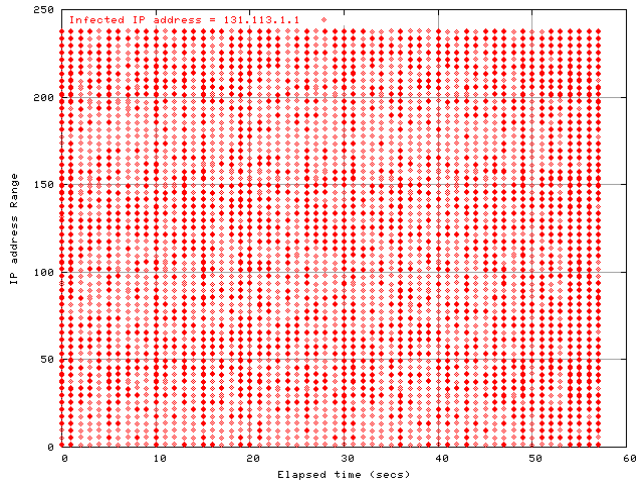
3.2 TCP retransmission behavior of Network Worms

TCP retransmission is one of the factors which affect infection efficiency. In the experimental environment (Figure 1), any packets are dropped at the monitoring PC. Then Code Red, Nimda, Blaster and Sasser will try a procedure of TCP retransmission. The progression of TCP packet transmission is shown in Figure 12. Blaster controls the TCP retransmission and sends TCP packets stably. The Code Red III and Nimda.E send TCP packets more stably and periodically than Sasser. Among the Sasser family, Sasser.B and Sasser.C cannot send TCP packets stably, especially on Japanese Windows XP Professional (Figure 13 and Figure 14). And the expansion of threads from 128 (Sasser.B) to 1024 (Sasser.C) affects the stability of TCP transmission. According to the code analysis, Sasser infects Japanese Windows XP but does not infect Japanese Windows 2000.



Time: from started obs. to counted 7,500 pkts.

Figure 8 Map of retrieved IP addresses by Blaster.



Time: from started obs. to counted 10,000 pkts.

Figure 9 Map of retrieved IP addresses by Slammer.

On Japanese Windows 2000, Sasser.B sends TCP packets more stably than in other environments. The experimental environment can provide the information on the stability of TCP transmission. These verifications on Japanese Windows 2000 environment and the comparison between Sasser.B and Sasser.C are useful to forecast the impact of infection of Sasser.

3.3 Infection behavior of Network Worms

This section describes the infection behavior of Blaster, Sasser and Welchia verified by using the experimental environment (Figure 3). The infected PC and the monitoring PC are the same ones described in the section 3.1. The target PC is a Hitachi FLORA Pentium 4 (Memory 1GB) with Microsoft Japanese Windows XP Professional Service Pack 1. Also, Blaster, Welchia and Sasser were verified on Microsoft Japanese Windows XP Professional without Service Pack on a VMware. The IP address of the infected PC is 131.113.1.1 and that of the target PC is 192.168.1.1.

(1) Blaster

The result of the flow analysis of Blaster is shown in Table 6. The monitoring PC captures 3,317 packets. The TCP connection using the destination port number series (135/TCP, 4444/TCP) is counted 2 times. The first appearance packet of this series is the 7th packet. The UDP using the destination port number series (69/UDP) is counted once . The first appearance packet of this series is the 125th packet . The network worm infection process is completed in about 4.5 seconds.

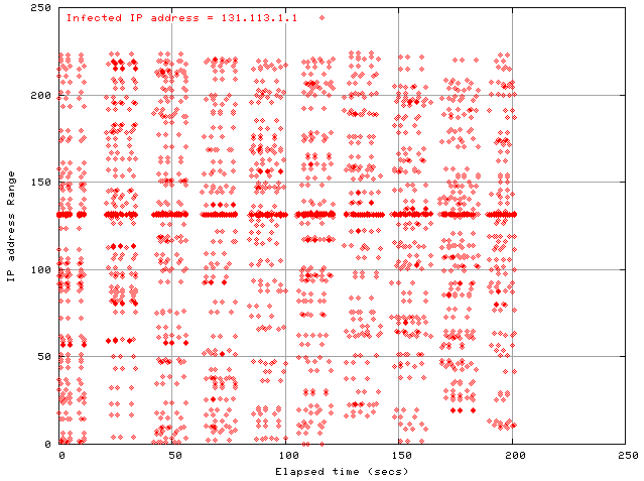
(2) Welchia

On Aug. 18, 2003, the Welchia worm was discovered. The Welchia worm attempts to exploit two vulnerabilities: DCOM RPC and WebDAV. The flow analysis of Welchia which exploits the DCOM RPC vulnerability is shown in Table 7. The monitoring PC captures 77,448 packets. The infection behavior of Welchia follows three steps: (a) via 53/UDP, (b) by ICMP packets and via 135/TCP, and (c) via 707/TCP and 69/UDP.

Table 4 Ratio of IP addresses retrieved by Sasser.B.

Selection of IP address	Experiment Data	Code Analysis
The same first two octets	27.2%	25%
The same first octet	24.6%	23%
Others	48.2%	52%

The average of 5 trials (3,000 packets/trial).



Time: from started obs. to counted 3,757 pkts.

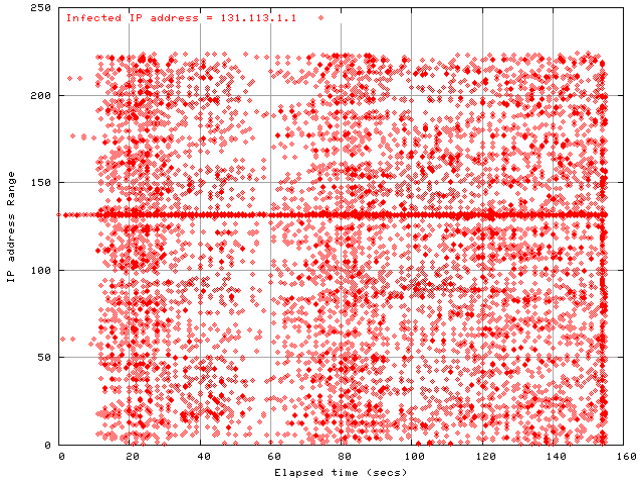
Figure 10 Map of retrieved IP addresses by Sasser.B.

The Final step is confirmed with the 2,618th packet. The network worm infection process is completed in about 3.3 seconds. [a]

Table 5 Ratio of IP addresses retrieved by Sasser.C.

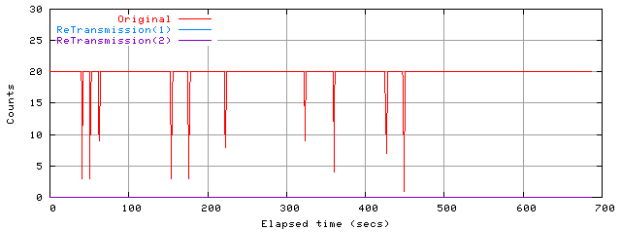
Selection of IP address	Experiment Data	Code Analysis
The same first two octets	27.1%	25%
The same first octet	24.8%	23%
Others	48.1%	52%

The average of 5 trials (10,000 packets/trial).

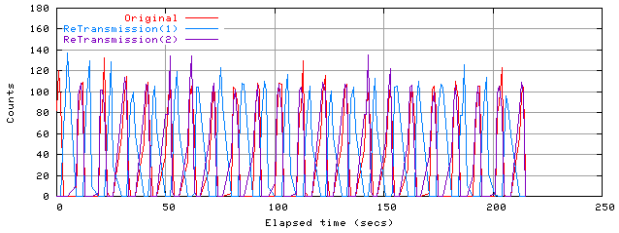


Time: from started obs. to counted 13,602 pkts.

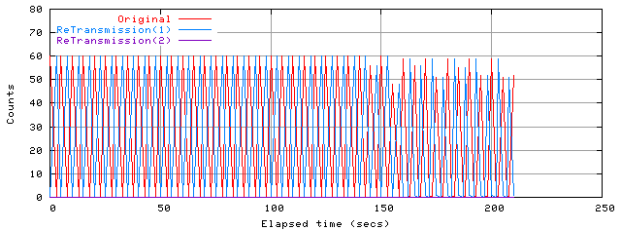
Figure 11 Map of retrieved IP addresses by Sasser.C.



Blaster (Time: from started obs. to counted 7,500 pkts.)



Code Red III(Time: from started obs. to counted 10,000 pkts.)



Nimda.E (Time: from started obs. to counted 5,000 pkts.)

Figure 12 Progression of TCP packet transmission on Japanese Windows 2000 Server.

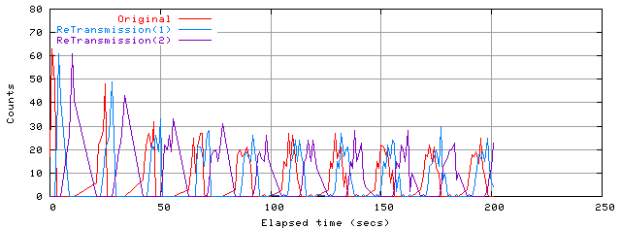
[a]) The default of experimental environment cannot trace the infection behavior of Welchia, because Welchia confirms the existence of microsoft.com domain via DNS. Then we customized the prototype system to trace the infection behavior.

(3) Sasser.B

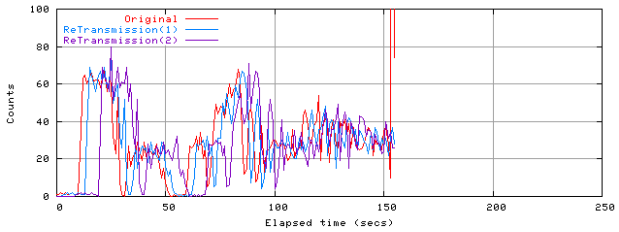
The flow analysis of Sasser.B is shown in Table 8. The monitoring PC captures 44,509 packets. According to Table 8, the infection behavior of Sasser.B appears to follow two steps: (a) via 445/TCP and 9996/TCP and (b) via 5554/TCP and 1033/TCP. The Final step is confirmed with the 367th packet. The network worm infection process is completed in about 4.1 seconds.

4. Conclusion

We proposed two types of experimental environment – "the experimental environment for retrieval behavior" and "the experimental environment for infection behavior" to satisfy the requirements set in the section 2. We described the retrieval behavior of Code Red, Nimda, Slammer, Blaster and Sasser verified in the experimental environment. In addition, we described the infection behavior of Blaster, Welchia and Sasser. The experimental environment sometimes cannot trace the whole infection activities of network worms.

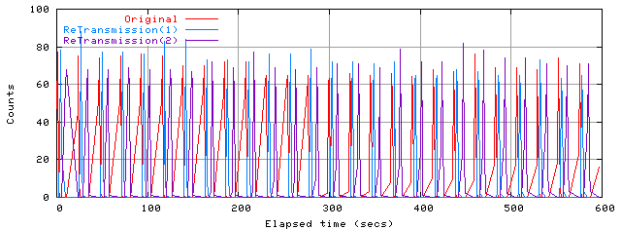


Sasser.B (Time: from started obs. to counted 3,757 pkts.)

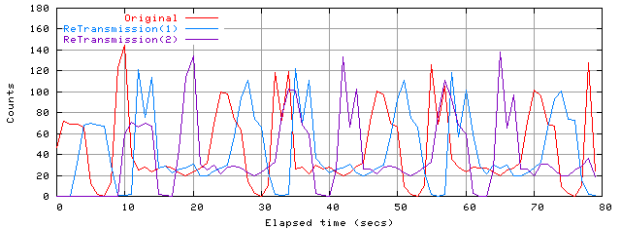


Sasser.C (Time: from started obs. to counted 13,602 pkts.)

Figure 13 Progression of TCP packets transmission on Japanese Windows XP Professional.



Sasser.B (Time: from started obs. to counted 10,000 pkts.)



Sasser.C (Time: from started obs. to counted 10,000 pkts.)

Figure 14 Progression of TCP packets transmission on Japanese Windows 2000 Server.

But we think this approach is important to complement the code analysis and it is one of the methods that can support organizations in planning countermeasures. The reason is that the experimental environment doesn't need to use a special device, and is a small-scale, readily-available system with just sufficient hardware/software.

For our future work, we will enhance peripheral functions such as DNS server for the experimental environment and develop the experimental environment for advanced malware which detects "non-real" environment thus cannot be invoked on the virtual machine.

Table 6 Result of Blaster by Flow analysis.

The series of port numbers	Counts	The logs of first appearance packet Line1 : Packet No, Elapsed time, src IP address, dst IP address, Protocol, src PT number, dDst PT number Line2 : Flag information
135/TCP	540	3 0.000000 131.113.1.1 115.11.58.1 TCP 1032 > 135 [SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=1460
135/TCP 4444/TCP	2	7 0.005741 131.113.1.1 115.11.58.5 TCP 1036 > 135 [SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=1460 103 2.276719 131.113.1.1 115.11.58.5 TCP 1052 > 4444 [SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=1460
69/UDP	1	125 4.517823 192.168.1.1 131.113.1.1 UDP Source port: 1031 Destination port: 69

Table 7 Result of Welchia by Flow analysis.

The series of port numbers	Counts	The logs of first appearance packet Line1 : Packet No, Elapsed time, src IP address, dst IP address, Protocol, src PT number, dDst PT number Line2 : Flag information
53/UDP	1	1 -4.080282 131.113.1.1 144.144.144.144 UDP Source port: 1031 Destination port: 53
ICMP 135/TCP	4434	3 0.000000 131.113.1.1 131.113.0.0 ICMP Echo (ping) request 5 0.007983 131.113.1.1 131.113.0.0 TCP 1032 > 135 [SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=1460
707/TCP 69/UDP	1	29 0.050722 192.168.1.1 131.113.1.1 TCP 3011 > 707 [SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=1460 2618 3.345150 192.168.1.1 131.113.1.1 UDP Source port: 3060 Destination port: 69

Table 8 Result of Sasser.B by Flow analysis.

The series of port numbers	Counts	The logs of first appearance packet Line1 : Packet No, Elapsed time, src IP address, dst IP address, Protocol, src PT number, dDst PT number Line2 : Flag information
445/TCP	1254	9 0.000000 131.113.1.1 131.113.202.138 TCP 1054 > 445 [SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=1460
445/TCP 9996/TCP	586	10 0.003998 131.113.1.1 131.225.169.253 TCP 1055 > 445 [SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=1460 231 2.748501 131.113.1.1 131.225.169.253 TCP 1075 > 9996 [SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=1460
5554/TCP 1033/TCP	1	353 4.024249 192.168.1.1 131.113.1.1 TCP 1032 > 5554 [SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=1460 367 4.135242 131.113.1.1 192.168.1.1 TCP 1084 > 1033 [SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=1460

Reference

1) Symantec Worm Simulator
http://enterprisesecurity.symantec.com/content.cfm?articleid=5479
2) ANALYSIS: Sasser Worm
http://www.eeye.com/html/research/advisories/AD20040501.html
3) ANALYSIS: CodeRed II Worm
http://www.eeye.com/html/research/advisories/AL20010804.html
4) ANALYSIS: Blaster Worm
http://www.eeye.com/html/research/advisories/AL20030811.html
5) ANALYSIS: Microsoft SQL Server Sapphire Worm
http://www.eeye.com/html/research/advisories/AL20030124.html