4 Applied Virtualization Technology

4-1 Design and Implementation of an Isolated Sandbox Used to Analyze Malware

MIWA Shinsuke, KADOBAYASHI Youki, and SHINODA Yoichi

Recent viruses, worms, and bots, called malwares, have to be analyzed their behaviors for drawing out countermeasures against them. To avoid any impacts to/from the Internet, analyzing environments should be isolated from the Internet.

In this paper, we defined the levels of containing malwares, also estimated our developed isolated sandboxes according to definition of the levels. Furthermore, we propose new isolated sandbox, which designed according to the estimation.

Keywords
Malware, Live analysis, Isolation, Containment

1 Introduction

Malware such as viruses, worms, and bots[1] are continually evolving. To combat these threats, their mechanisms of operation must be understood and their adverse effects must be correctly identified. To carry out such analysis, isolated environments are effective in preventing further infection and mitigating the effects of these attacks into external areas.

Research and development have been applied to techniques that employ swapping between actual nodes and node renewal, potentially offering the same level of recoverability as virtualization technologies. Techniques have also been investigated using a mimetic Internet that tricks the malware entities, rendering them unaware that they are being analyzed in an isolated environment[2]. This paper will first define the levels of isolation and summarize the methods of bypass and circumvention against isolation employed on the malware side, followed by an introduction to the design and implementation of the isolated sandbox for malware, which will enable isolation experiments for a wider variety of malware.

2 Background

First, I will describe why malware analysis and isolation techniques are necessary. There are basically two major strategies for analyzing malware. First, there is static analysis, in which the mechanism of the malware behavior is analyzed based on its program code, without executing the program instance; second, there is live analysis, in which analysis is performed through observation of malware behavior during execution of the program instance.

If the effects of the malware were to be determined, it would then be possible to design and implement countermeasures. Moreover, since many of the malware countermeasures involve the detection of and

response to malware by tracking transmission logs and file access history, information on the effects of malware behavior will assist in the detection of and response to new types of malware. Therefore, live analysis is widely used as a technique of analysis in which the effects of malware behavior are measured.

However, in live analysis, the malware program instance must be run in an analysis environment to observe the effects of its behavior, and since the malware will actually make attempts to infect and attack, its adverse effects may spread to external areas when the environment is connected to the Internet. Thus, measures must be taken to prevent such infection or attacks.

In addition, the number of malware currently proliferating on the Internet is quite large[3], and any analysis environment having direct access to the Internet may be affected by malware other than the target malware. Thus, measures must be taken to eliminate these external factors.

In order to eliminate effects migrating to or from the external areas, the analysis environment must feature a physical or logical barrier to isolate the executing environment of the malware from the outside. Environments having such barriers will be defined as "isolated" in the present paper. The environment in which the live analysis of malware or other experiments are performed will be referred to as the "sandbox."

3 Isolation

As described above, some form of isolation strategy must be employed when carrying out live analysis of malware. This chapter will summarize the activities of the malware that will serve as the target of isolation, potential effects from external areas, techniques of isolation, and problems in isolation.

3.1 Targets of isolation

Isolation is carried out for two purposes: to contain malware activity and to eliminate effects of the external area on the sandbox. Malware activity crosses the boundary between the sandbox and the external area through one of two paths: direct, physical access to the external area, and indirect access via network. It must be noted that in the latter case, there is also an element of direct access through physical connection of the sandbox, as well as the indirect connection. Based on these observations, this section will discuss the first purpose of isolation, the containment of malware activity within the sandbox.

Malware activity basically consists of infecting, spying, and attacking.

Infecting refers to activities that attempt to copy the malware program instance, or a portion of it, onto another host or media. Malware known as Trojan horses make the user run a program without his knowledge or consent to initiate the infection process. Some malware takes advantage of a vulnerability to initiate the infecting process immediately and automatically after infection. Malware known as bots perform infecting activities on command from external bot herders. Infection magnifies the extent of damage, and is the main activity of many malware. Without this activity, malware serves simply as an attack or spyware tool. Thus, containing the infection is the most important goal of isolation.

Spying involves the attempt to collect accessible information from the host on which the malware is being executed and to transfer it to another host or media. Spying may include activities such as the extraction of a specific file from the host or the acquisition of IDs or passwords via keylogger. In particular, when the intention of the malware author is to gather information, transmission of information tends to be directed toward a specific host. In the case of bots, most are transmitted to a C&C (Command and Control) network. Spying is normally a preparatory step for future attacks or a part of a larger plan of criminal activity, and so must be contained to prevent the leakage of important, often confidential, information.

Attacking is directed against the infected host or against another host from the host on

which the malware is being executed. Examples include DoS attacks on a specific host or the destruction of files in the host storage. When the purpose of a malware is to attack, it will tend to attempt to infect other hosts as well as to target a specific host for the attack. In the case of bots, commands from the bot herder will initiate various forms of attacks. If among the various types of attack the target of a given attack is the infected host, there is no need for containment. When the attack is targeted toward a different host from the one infected, then the malware must be contained to prevent damage.

3.2 Techniques and levels of isolation

As stated above, the spread of malware from the sandbox or the effects of malware to the sandbox from the outside requires either a direct physical connection or an indirect connection via a network. In both cases, a medium is involved. In the case of physical connection, the media will consist of both the physical storage medium and the network, and in the case of indirect connection, the media will consist of the network.

The techniques for isolation will thus involve the blocking of the physical connection and preventing the network from functioning as the medium. When the network acts as the medium, different strategies may be required to contain the malware within the sandbox, and to eliminate the effects of malware penetration from the outside.

Since the purpose of the isolated sandbox is to execute the malware specimen and to acquire experimental data, some method must be devised to allow the injection of the specimen and acquisition of experimental data while keeping the sandbox isolated. The malware must also be allowed to operate freely even while its activity is contained.

Various practical isolation techniques may be designed. In this paper, the levels of isolation achieved by the various techniques will be defined from the perspective outlined above. The degree to which physical connection is prevented will be referred to as the Physical Security Level (PSL), the degree of containment of malware activity via networks will be referred to as the Malware Containment Level (MCL), and the extent to which effects originating from the exterior are eliminated will be referred to as the Environment Isolation Level (EIL). Each of these values will be represented numerically according to several levels of isolation. Level 0 is the lowest security level, and higher values will represent higher security levels. The present paper will deal mainly with the containment of malware activity, and so focus will be placed on MCL; PSL and EIL will not be discussed herein.

3.2.1 Malware Containment Level (MCL)

The sandbox for isolation in our study is assumed to consist of an executing environment for executing the malware, peripheral instruments for analysis and observation, management instruments to manage and control the executing environment and these peripheral instruments, and networks and other wiring to connect the instruments. The management instruments are assumed to be equipped with sufficient security countermeasures relative to known vulnerabilities. The sandbox network consists of three sub-networks: an experimental network on which the malware actually operates and experimental data are collected; a management network, which oversees malware execution and manages the component instruments; and an external network outside the sandbox, such as the Internet. The scheme of the sandbox environment is presented in Fig. 1.

When malware activity is carried out via networks, the form of connection between the experimental network, management network, and the external network will be an important factor. The connection may be a physical one, involving, for example, physical switches or link layers such as VLAN; or it may be a logical connection — e.g., via routers and stepping-stones. Different situations must be taken into consideration when determining how restrictions are imposed on communications,

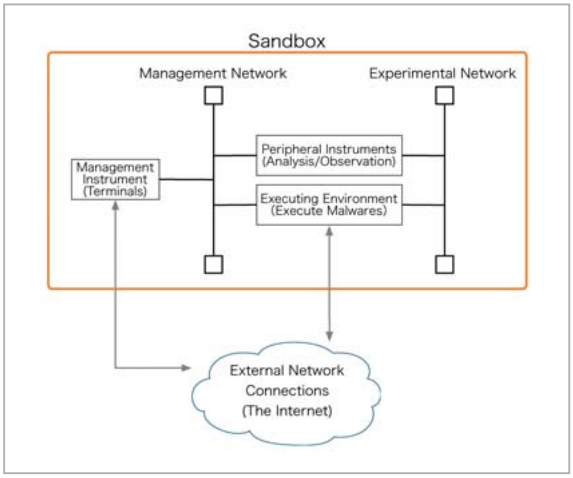

Fig.1 *Scheme of the sandbox*

such as uniform restriction by firewalls, partial restriction allowing specific types of communication, or elimination of direct communication through the installation of proxy and/or application gateways. Another point that must be taken into consideration is the possibility of secondary infection from the management network to the external network through the management terminals in the event malware activity spreads to the management instruments.

Based on these factors, six MCL levels are defined, as shown in Table 1. The levels in the table correspond to the relative degrees of containment. Each level entails a different method of connecting the networks and a different method of restricting communication. The "Target malware" column gives the types of malware that may be safely subject to experiment at each MCL level without negative impact on the external network.

"Direct connection" refers to a configuration in which no restrictions are placed on the connection method, where all networks share the same network switches and are installed in identical network segments. "Logically isolated" refers to a configuration in which the networks will share physical network switches, but will be installed in different network segments using VLAN or the like. "Physically isolated" refers to a configuration in which network switches are not shared and the networks are installed in physically separate network segments. With "Restriction by firewall, etc." permission for communication is judged in a port-level inspection, and "Allows only specific types of communication" means that communication following specific protocols is identified and allowed. "Direct communication not permitted; allowed only via dedicated proxy, etc." refers to a configuration in which communication is first received by proxy or

Table 1 *Definition of MCL (Malware Containment Level)*

Levels	Containment method		Target malware
MCL–0	Executing environment → external Management → external Connection	No restriction No restriction Direct connection	Known, and determined to have no adverse effects on external network
MCL–1	Executing environment → external Management → external Connection	Restriction by firewall, etc. Restriction by firewall, etc. Each logically isolated	Known, with known communication activities, and easy extermination
MCL–2	Executing environment → external Management → external Connection	Allows only specific types of communication Restriction by firewall, etc. Each logically isolated	Specifics unknown, but utilizes only known vulnerabilities
MCL–3	Executing environment → external Management → external Connection	Direct communication not permitted; allowed only via dedicated proxy, etc. Allows only specific types of communication Experimental network connection physically isolated: others logically isolated	Unknown and attempts attacks on unknown vulnerabilities
MCL–4	Executing environment → external Management → external Connection	No communication permitted Direct communication not permitted; allowed only via dedicated proxy, etc. Experimental network connection physically isolated: others logically isolated	Unknown and may possibly involve extremely hazardous activities
MCL–5	Executing environment → external Management → external Connection	No communication permitted No communication permitted Each physically isolated	May not be prevented except by physical isolation

application gateways and only those judged to be normal are permitted.

When a sandbox satisfies all of the containment methods for a specific level given above, it is defined as reflecting MCL on that level. A sandbox will not be judged by any range of superior features, but simply by the degree to which it satisfies all of the conditions pertaining to a given level. Note that here, containment of the penetration of malware from the management instruments into the external network is represented by the "Management → external" condition; however, when restrictions on communication from the executing environment to the management network are stricter than those from the management network to the external network, then the same restrictiveness may be deemed satisfied by connection between the management instruments to the external network. In other words, the stronger of the two restrictions — "Management → external" (given in the table) and "Executing environment → management" (not shown in the table) will be used as the criteria for judging whether the sandbox satisfies the "Management → external" condition.

3.3 Problems in isolation

In live analysis of malware, isolation is an effective technique for containing malware activity. However, increasing the degree of isolation may result in the following problems.

1. Easy detection of the executing environment by malware
2. Blocking of communication required for malware activity

Overviews for these problems will be given below, as well as some solutions.

3.3.1 Detection of the executing environment by the malware

Malware can detect the executing environment of the malware by inspecting to determine whether the malware is running in the environment we have established for analysis. Live analysis will be inhibited when the malware suppresses its activity or conceals itself when it detects an isolated sandbox.

Isolated sandbox detection by malware normally involves the confirmation of user IP addresses and scanning for connectivity. The malware confirms user IP addresses by checking to confirm whether it is operating in a private address space, often used for isolated sandboxes. In addition, the malware checks for connectivity to specific hosts or services on the Internet to confirm whether it is confined to an isolated sandbox. Since these techniques are extremely straightforward, they are widely employed.

To counter the checking of IP addresses, it will suffice to deploy addresses besides the private ones in the sandbox, after eliminating the risk of effects on the external network. In contrast, the checking of connectivity must be countered by allowing connections to specific hosts or services at the cost of the containment level, by providing a mechanism that can take the place of the Internet[4], or by constructing a mimetic Internet[2].

3.3.2 Communication required for malware activity

Malware that require communication with an external network will shut down or discontinue normal operation when such communication is blocked. Thus, it will not be possible to observe the behavior of malware entities that download a main program upon execution, or those that participate in C&C networks to receive commands (such as bots) when unexpected communications are initiated at isolation levels above MCL-2 or in a sandbox above MCL-4.

Measures implemented to resolve this problem at the cost of the containment level include permitting communications to specific hosts, such as the source host of the malware[5].

4 Design and implementation of the malware isolation sandbox

Thus far we have defined MCL, which represents the level of isolation, followed by a presentation of problems encountered in attempts at isolation. In this chapter, malware

sandboxes developed by our group will be re-evaluated in light of the foregoing discussion, and a summary will be provided on the design and implementation of a new malware isolation sandbox.

4.1 Evaluation of existing sandboxes

The VM Nebula and isolated sandbox used to analyze malware with mimetic Internet, created through the R&D efforts of our group, will be evaluated based on the respective degrees of isolation and in consideration of the problems mentioned above.

4.1.1 VM nebula

The VM Nebula[6] is a sandbox for experiments not only on malware, but also on overall Internet security. The sandbox makes use of virtualization technologies to virtualize four servers as a maximum of 256 PC hosts, allowing for the possibility of large-scale experiments. This configuration also enables easy recovery of the environment following destructive experiments. A schematic rendition of the architecture of the sandbox is presented in Fig. 2.

A multiple PC environment is mimicked using the VMware Server of VMware, Inc. on a server simulating attackers, a server simulating victims, and servers simulating networks. Communication during the experiment is performed via LANs for the experimental system, and each of the servers is controlled via a control LAN. Access to external networks such as the Internet is not provided. The experimental LAN and the control LAN are physically isolated using separate network switches. Based on these characteristics, the VM Nebula may be classified as an MCL-5 isolated sandbox.

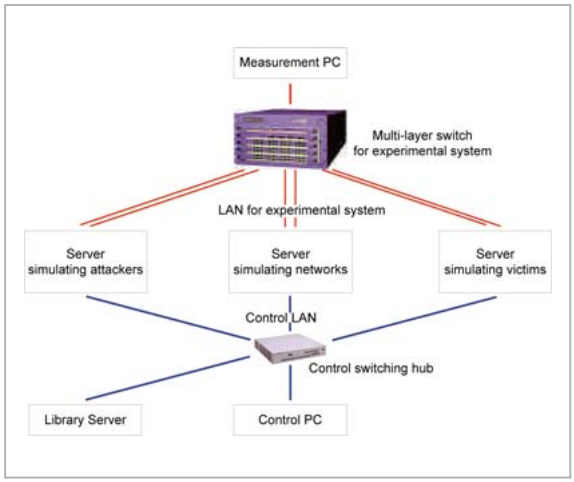

Fig.2 *Architecture of the VM Nebula*

Since the VM Nebula is an MCL-5 isolated sandbox, experiments with dangerous malware specimens may be carried out safely. However, it is not possible to perform precise observation of the behavior of malware entities that download their program bodies from an external network or that those that check connectivity with the Internet. The VM Nebula is also only minimally user-friendly, since it requires the introduction from outside of physical media required to update the OS of the executing environment and the management instruments.

4.1.2 Isolated sandbox used to analyze malware with mimetic Internet

The isolated sandbox used to analyze malware with mimetic Internet[2][7][8] is an isolated sandbox for safe live analysis of malware equipped with analysis-evading functions such as the recognition of virtualized and isolated environments. Such functions have been countered by combining a technique of swapping between and renewal of actual nodes that offers recoverability equal to virtualization technologies, and a technique for tricking the malware using a mimetic Internet such that the malware fails to detect that it is being analyzed in an isolated environment. The architecture of the scheme is shown in Fig. 3.

This structure consists mainly of a malware incubator, which serves as the malware executing environment through its provision of a renewable actual node, a mimetic Internet that mimics the Internet, a controller node group that controls the incubator and mimetic Internet (hereafter referred to collectively as the controller node group), and a management terminal. All sections, with the exception of the management terminal, are present within the isolated environment. The malware is run on the malware incubator, and the mimetic

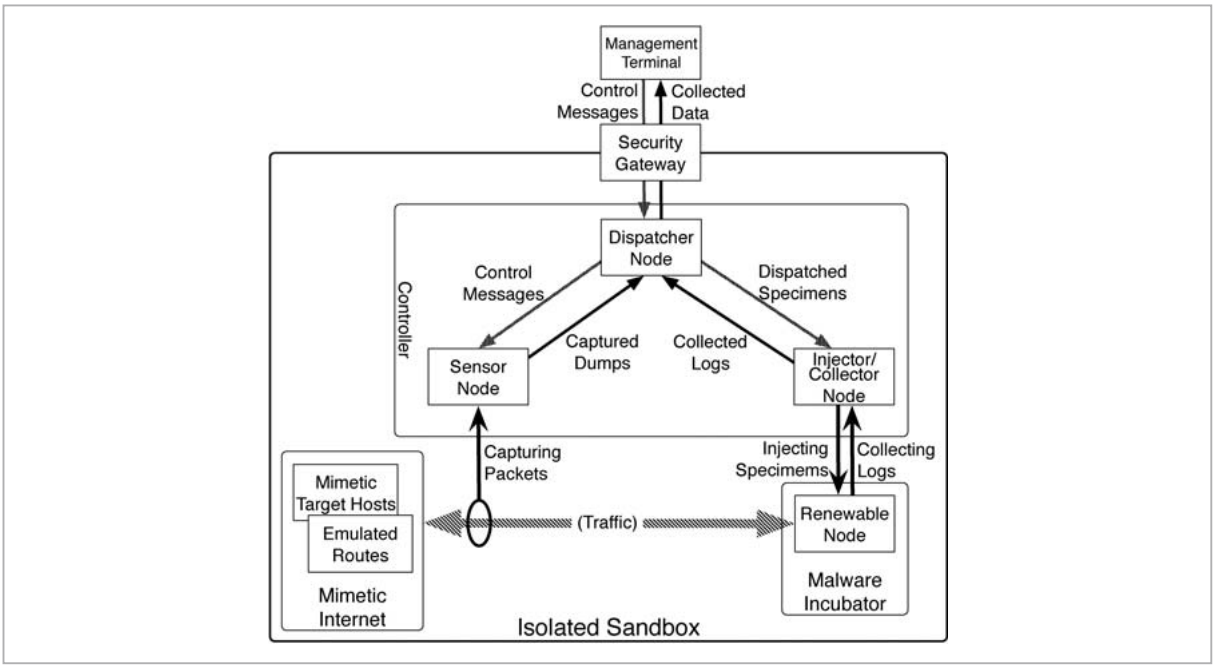

Fig.3 *Architecture of the malware isolation and analysis sandbox with mimetic Internet*

Internet mimics access to the Internet in order to trick the malware's connectivity-checking function. The experimental network and the management network are physically and logically isolated, respectively. Communication from the malware incubator to the management network is completely blocked during malware execution. During data collection or the injection of a specimen, the malware incubator is shut down and restarted with a different network boot OS. Thus, the malware activity does not spread into the control node group. A security gateway that permits only a specific type of communication is installed between the controller node group and the management terminal, providing a double isolation barrier. In light of the foregoing structure, the present isolated sandbox used to analyze malware with mimetic Internet may be regarded as an MCL-4 equivalent isolated sandbox.

Although the mimetic Internet is capable of tricking the connectivity-checking function of the malware, it will not allow the proper execution of malware requiring downloads from a site and commands from a C&C network. Moreover, the present mimetic Internet is a fixed environment with a limited number of servers and networks, and so malware that checks connectivity to hosts and services that are not provided will not be tricked.

4.2 Proposed method

We propose a method for safe live analysis, similar to those described in existing studies, in an isolated sandbox with a security level above MCL-4. We will also aim to develop a system that will enable isolated experimentation on a wider variety of malware by devising measures to handle malware that require communication with the external network (including downloading, C&C network communication, etc.) and more complex connectivity-checking schemes.

4.2.1 Design

First, the basic architecture will be that of an isolated sandbox used to analyze malware with mimetic Internet, and an isolated sandbox capable of handling various malware types will be constructed by expanding this architecture. The presently identified problems may be solved if the mimetic Internet can be designed to allow downloading of data and program instances and C&C network participation and to counter more complex connectivity-checking functions. Thus the following two func-

tions were added to the mimetic Internet.

- High-fidelity mimetic Internet (HF-mimetic Internet)
- External information gathering agent (download agent)

The HF-mimetic Internet will have two additional major functions relative to the current mimetic Internet.

- Collection and recording of access information from malware (access collector)
- Dynamic mimetic services, mimetic hosts, and mimetic networks (dynamic constructor)

The former function will allow for the collection and recording of information on the accessed hosts, services, and protocols actually used by the malware. Based on these logs, the latter function will be used to newly incorporate the mimetic services and hosts required to run the malware on the mimetic Internet. Since in this scheme, the mimetic Internet is constructed based on the access information used for connectivity checking by the malware, the malware may be tricked no matter what scheme it uses for checking. In addition, by collaborating with the download agent to acquire data and the program instance downloaded by the malware or the communication from the C&C network, it should be possible to observe the behavior of malware requiring such communication with the external network.

The download agent extracts the download requests and participation messages to C&C networks from the access information collected by the access collector on the HF-mimetic Internet. The download agent then executes the requests and participation messages in place of the malware on the external network and returns the downloaded contents or command messages to the HF-mimetic networks. This strategy prevents the malware from carrying out harmful communication directly, ensuring safety while adding the information required by the malware into the isolated sandbox.

4.2.2 Implementation

The scheme of the present malware isolation sandbox with high fidelity mimetic Internet is shown in Fig. 4. The proposed method is

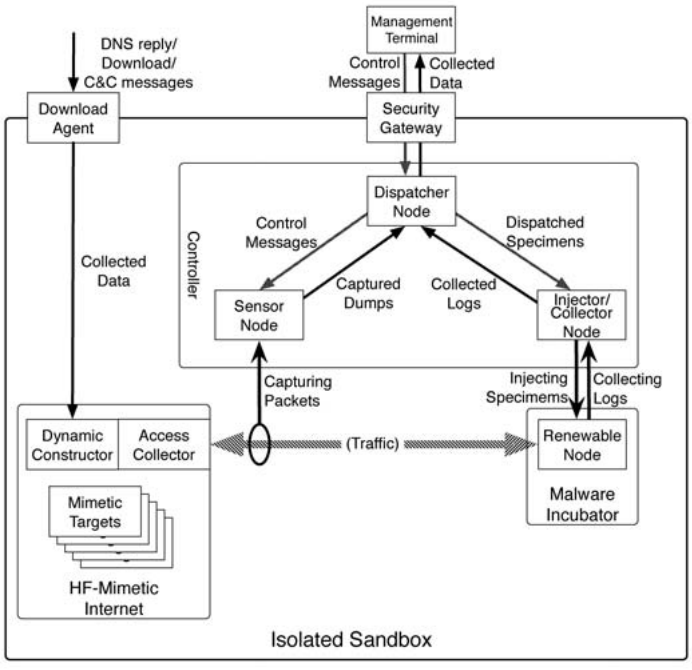

Fig.4 *Scheme of the malware isolation sandbox with high-fidelity mimetic Internet*

currently being implemented. The implementation is summarized as follows.

The access collector simply performs packet capture at the entry point to the mimetic Internet and lists the host names, IP addresses, and protocols accessed. The currently implemented access collector is capable of handling DNS queries and HTTP access.

The dynamic constructor acquires the list of host name and IP addresses from the access collector and injects the corresponding DNS records and service servers. Implementation of a simple DNS record generator and HTTP server setting generator is now underway.

The mimetic targets consist of the mimetic DNS server and mimetic service servers, mimetic clients, and mimetic networks, all dynamically generated based on the settings supplied by the dynamic constructor. The method of implementation is currently being examined.

The download agent acquires the list of host names, IP addresses, and access protocols from the access collector, and attempts proxy access to the Internet. Implementation of DNS query proxy and HTTP proxy is underway.

A mechanism is also being implemented that will automatically select between the virtual-machine malware incubator and the actual-node malware incubator.

5 Discussion and prospects

Implementation of the proposed method is currently underway, and after completion, a validation experiment will be performed using an actual malware specimen to examine the method's effectiveness. Here, we will touch on discussion and prospects of the isolated sandbox.

5.1 Discussion

When using an isolated sandbox such as the one proposed, problems similar to those described in Section **3.3** will be encountered; namely, the easy detection of the executing environment by the malware and the blocking of communication required for malware activity. This implies an inevitable trade-off between the level of isolation and the types and quantities of malware suited to analysis.

For unknown and harmful malware, the level of isolation must be increased. However, malware created using the latest technology may have sophisticated executing environment-detection functions, requiring extensive coordination with the external network during operation. Therefore, the trade-off is anticipated to be more costly for unknown and harmful malware that entail the deployment of such isolated sandboxes.

The proposed method attempts to overcome the trade-off, but its ability to trick the malware or to acquire and provide sufficient information to allow for malware activity depends on the capability of the HF-mimetic Internet. Since the capability of the HF-mimetic Internet may be improved by adding mimetic mechanisms that correspond to the required elements, this approach may lead to an endless game of cat-and-mouse between the development of new counter-isolation techniques on the malware side and the incorporation of new mimetic mechanisms in the sandbox to counter these techniques. Thus, one of the themes for future studies is to find an architecture that will be capable of handling new malware activity without requiring new functions.

5.2 Prospects

A malware isolation sandbox such as the one proposed may be applied to purposes outside malware analysis, such as penetration tests for security products and education and training in the security field. R&D is now being applied to the creation of a malware-analysis training field using the proposed technique, for training of security specialists.

Section **3.2** of this paper presented an overview on the level of isolation. By providing this measure of evaluation for security sandboxes, we hope to promote comparative capability studies between different sandboxes and to assist in the circulation of the results between different sandboxes.

6 Conclusions

Malware technology is advancing day by day, and it is essential that we keep up with analysis and testing of malware. The present paper has presented a scheme for measuring the degrees of isolation of isolated sandboxes used to perform safe live analysis of malware behavior and effects, and we have isolated sandboxes created by our group based on the developed scheme. We then proposed a high-fidelity mimetic Internet that will allow a sandbox to maintain its current level of isolation while allowing application to a wider variety of malware isolation experiments.

Our proposed methods will allow downloading from and participation in control and command networks and will counter more complex connectivity-checking schemes, tasks previously difficult to execute in existing sandboxes. These methods will therefore allow us to examine the behavior and effects of the latest malware by carrying out live analysis in a safely isolated environment.

Implementation of the proposed method is currently underway; once implementation is complete, validation tests will be carried out using actual malware specimens.

References

1 E. Skoudis with L. Zeltser, "MALWARE – Fighting Malicious Code –", Prentice Hall PTR, ISBN 0-13-101405-6, Pearson Education Inc., 2004.

2 S. MIWA, T. MIYACHI, M. ETO, M. YOSHIZUMI, and Y. SHINODA, "Design and Implementation of an Isolated Sandbox with Mimetic Internet used to Analyze Malwares", DETER Community Workshop on Cyber Security Experimentation and Test 2007 (DETER07), Aug. 2007.

3 Information-technology Promotion Agency, Japan (IPA), "Measures Against Computer Viruses and Unauthorized Computer Accesses", http://www.ipa.go.jp/security/english/first.html, 2008.

4 T. SUDOH and K. FUJIHARA, "The evaluation of the botnet analysis system based on the virtual internet environment", Computer Security Symposium 2006 (CSS2006), Oct. 2006. (in Japanese).

5 S. BABA, Kouei SUZUKI, and Kazuya SUZUKI, "Event profiling using honey-farm", The 2007 Symposium on Cryptography and Information Security (SCIS2007), Jan. 2007. (in Japanese).

6 S. MIWA and H. OHNO, "A Development of Experimental Environments "SIOS" and "VM Nebula" for Reproducing Internet Security Incidents", Journal of the National Institute of Information and Communications Technology, Vol.52 Nos.1/2 (pp.23-34) 2005, ISSN 1349-3205, Oct. 2005.

***MIWA Shinsuke**, Ph.D.*

Researcher, Traceable Secure Network Group, Information Security Research Center

Networks Security

***KADOBAYASHI Youki**, Ph.D.*

Guest Expert Researcher, Traceable Secure Network Group, Information Security Research Center

Network Security

***SHINODA Yoichi**, Dr. Eng.*

Executive Director of Information Security Research Center

Distributed and Parallel Computing, Networking Systems, Operating Systems, Information Environment