3-7 Reproduction and Emulation Technologies for Researches on Secure Networking

MIWA Shinsuke

Mechanisms of various attacks must be analyzed in detail for clarifying and defining targets of research and development on secure networking. Moreover, new technologies concerning secure networking must be verified on the realistic network environment.

In this paper, we describe our reproduction and emulation technologies for researches on secure networking, and its applications.

Keywords
Reproduction, Emulation, Inter-AS network, Malware, Containment

1 Introduction

In research and development on secure networking, it is necessary to analyze the mechanisms of various types of malware[1] and attacks such as viruses, worms, and bots in order to clarify problems. These analyses require a collection of datasets that include contents of communication related to malware itself or attacks, as well as experiment environments that can reproduce the actual attacks based on the datasets. Moreover, when a new secure networking technology is developed, its effectiveness and properties need to be verified in a realistic network environment.

For this reason, we have been conducting research and development on reproduction and emulation technologies for secure networking, including reproduction of environments from the wide area Internet network to an organizational network environment, and emulation of service systems that can fool malware and attack tools. This paper overviews the background and technologies of our research, and describes the outcomes and applications.

2 Technologies of reproduction/emulation of the wide area Internet

First of all, we describe the reproduction/emulation technologies of the wide area Internet that we researched and developed in order to experiment with the security of the wide area Internet.

2.1 Background — the structure of the Internet

When considering the network security of the current ICT environments, the main concern is the security in the Internet. From a viewpoint of connectivity, it can be said that the Internet is a single network; however, in fact, it is divided by a unit called Autonomous System (AS) to which each client's network is connected to. In short, it has a multi-level structure that consists of the following (Fig. 1):

1) Network of each organization
2) Network within AS (inner-AS network) of providers
3) Network between AS (inter-AS network)

Following this, the location to implement the Internet security measures is roughly classified into the following three types:

1) End users' devices such as PCs

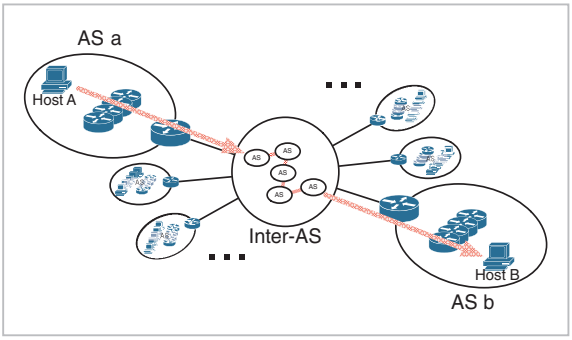

Fig.1 *Structure of the Internet*

2) Gateway of each organizational network
3) Gateway of AS

For example, anti-virus software will be implemented on 1), firewall and IDS will be on 2), and route filters will be on 3).

In each case, when a new technology has been developed, some kind of verification experiments will be required before the technology can be deployed. However, the wider the deployment area is, the more difficult it is to conduct such experiments. This means that it is more difficult to experiment with a technology that is used for the inter-AS network than one for users' devices or organizational networks. This is because there are issues in terms of the construction cost to prepare an environment for the actual experiment, and the operation cost caused by the requirement for cooperation of various organizations in order to cover the wider area.

Therefore, we have been conducting research and development on reconstruction/emulation of the inter-AS network in order to make it easy to experiment with technologies especially for the wide area Internet.

2.2 Method of reproduction/emulation of the wide area Internet

Our purpose is to verify a new technology in an environment as close as the real world, when the technology has been actually implemented in software or hardware. Therefore, the major task for reproduction/emulation technology was to consider how to construct an experiment environment for the inter-AS network that is as close as possible to the real Internet environment. In addition, in order to experiment with the implementation, the environment had to be built on a testbed that is capable of testing the implementation rather than on a network simulator.

On the other hand, there are more than 37,500 ASs on the Internet at present (as of May 2011, Fig. 2). There were already 25,000 at the time when our research started in 2006. Due to this scale, it had been considered that it was difficult to reproduce the whole inter-AS network, and therefore it had not been tried before. However, NICT has a testbed called StarBED[2] for research and development on networking which consists of more than 1,000 PCs, and in conjunction with the fact that virtualization technology was beginning to make progress, we started a bold attempt to reproduce/emulate the whole inter-AS network by combining virtualization technology and the large scale cluster-type testbed.

The inter-AS network uses Border Gateway Protocol (BGP) routing. Therefore, it is necessary to place BGP routers as required in order to emulate the inter-AS network. In addition, an inter-AS network is a network that connects voluntarily operated ASs based on the determined connection method, and its whole structure is not managed. For this reason, in order to reproduce the structure of an inter-AS

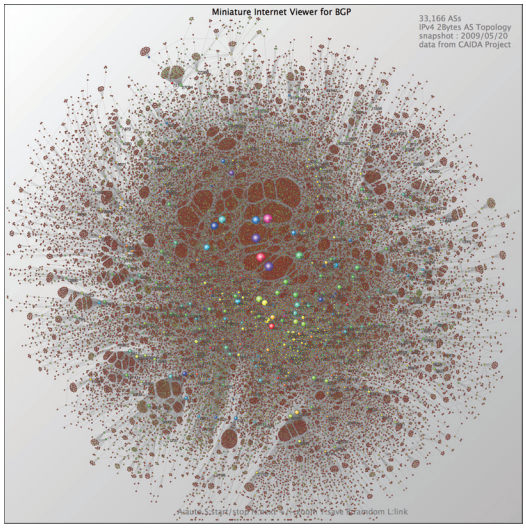

Fig.2 *Inter-AS network*

network, it is necessary to obtain information related to inter-AS network structure in an observational way, and to construct the structure based on that information.

We have developed a toolset (XENebula[4]) that distributes a large number of BGP routers on a testbed such as StarBED, by the use of virtualization technology, and another toolset (AnyBed[3]) that automatically generates configuration files for each BGP router from inter-AS network structure (topology) data and these configuration files. In the current status, 1 AS is emulated as 1 BGP router to simplify things, and a large number of Virtual Machines (VM) that can run BGP routing software are launched. The inter-AS network can be reproduced/emulated by connecting these VMs.

As for the inter-AS network structure, we use AS Rank annotated inferred relationship Dataset provided by CAIDA, and 1) cut out the required area of topology, 2) estimate the routes between BGP routers, and 3) generate configuration files with AnyBed. After that, in XENebula, based on the resource information of the configuration files and testbed (StarBED), 1) compute the allocation of VMs where BGP routing software is installed to each PC server, 2) distribute the OS image of the VM that runs on each PC server and the configuration file of BGP router, and 3) actually launch the BGP router as a VM on each server (Fig. 3).

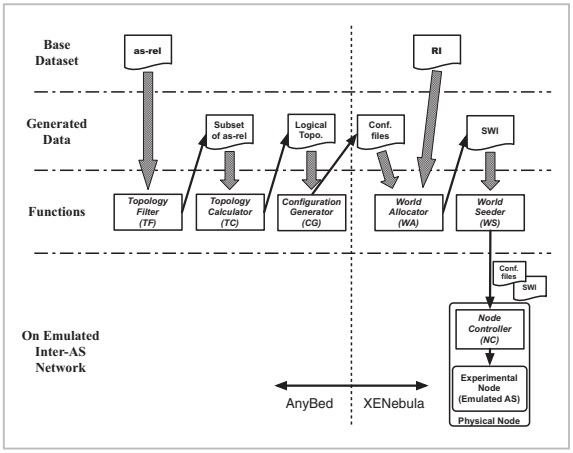

Fig.3 Architecture of AnyBed/XENebula

2.3 The outcomes and applications of reproduction/emulation technologies for the wide area Internet

By this method we have succeeded in reproduction and emulation of an inter-AS network that consists of 10,000 ASs, which is about one-third of the scale of the actual inter-AS network. The scale of this environment is unprecedented in the world in terms of reproduction/emulation of the inter-AS network.

The actual use cases of the reproduction/emulation of the inter-AS network environment based on this technology include preliminary experiments[5] for demonstration experiments of inter-AS IP-traceback technology.

Inter-AS IP-traceback technology is exactly the technology we assumed to be widely implemented on the Internet. The cost of demonstration experiments was high because it required building an experiment environment and operation across multiple providers. Therefore, the use of the environment for reproduction/emulation of the inter-AS network developed by us enabled the implementation of multiple preliminary experiments and minimized the number of demonstration experiments and organizations involved. We consider that we have achieved the results that we expected.

Research and development on reproduction/emulation of inter-AS network technology is continuing, and is now shifting from a phase where reproduction/emulation is simply carried out from a viewpoint of scale and structure to a phase that aims for faster construction and more accurate reproduction. Such large-scale reproduction/emulation technology is not limited to the field of network security, and is globally forming a research field of testbeds for network technology, in accordance with the progress of research and development to construct the next/new generation of network. We expect it to continue to make further progress in the future.

3 Reproduction/emulation technologies to fool malware/attack tools

Next, we describe the reproduction/emulation technologies to fool malware/attack tools.

3.1 Background — conflict between isolation and reproducibility

There are mainly the following two methods to analyze behaviors and mechanisms of malware such as viruses/worms/bots, and various attack tools:

1) Static analysis that analyzes mechanisms of behaviors from program codes.
2) Dynamic analysis that actually performs these behaviors and monitors/analyzes them.

Most countermeasures technologies detect and handle malware and attack tools by the traces left when they were triggered, such as the content of communication and file access history, and therefore, if the effect of those behaviors is known it is possible to handle new types of malware and attack tools. For this reason, dynamic analysis is widely used for measuring the effect of behaviors.

In dynamic analysis, it is necessary to actually execute malware or attack tools by certain execution methods and in a certain sandbox to observe the effect of their behaviors. This is to simulate the actual activities such as infection and attacks of malware and attack tools. Therefore, when the sandbox is connected to the external environment, some kind of countermeasures are required, as the effects such as infection may spread outside. In addition, as numerous malware already exist on the actual Internet, if the sandbox is directly connected to the Internet, it may be affected by malware and attacks which are not subject to analysis. For this reason, some countermeasures to eliminate external effects are required.

In order to eliminate such effects from or to the external environment, it is necessary to implement a certain barrier physically or on the network to separate the executing environment of the malware and attack tools from the external environment. This is called isolation[6].

However, a lot of malware and attack tools feature functions that make their analysis difficult so that they can avoid being analyzed in detail in an isolated environment. These features include checking connectivity to specific hosts or servers on the Internet. In addition, when malware and attack tools need to download certain information from the Internet during execution, or when bots require receiving a command via the network, they may not be executed successfully. To summarize, the following problems can be caused in an isolated environment.

1) It makes it easier for malware and attack tools to detect their executing environment.
2) The communication required for the activities of malware and attack tools may be blocked.

Isolation is an effective method to avoid effects from or to the external environment, but on the contrary, it may cause inaccuracy in behaviors of malware and attack tools, and as a result, reproducibility of the actual behaviors may be reduced.

For this reason, we have been conducting research and development on an isolated sandbox with a mimetic Internet that aims to fool malware and attack tools to analyze them accurately, while isolating and emulating services and hosts on the Internet.

3.2 Mimetic Internet

Malware and attack tools detect their executing environment in order to check whether or not they are executed in an environment for analysis purposes. Based on the check results, the malware and attack tools may suppress execution or conceal themselves in order to make dynamic analysis difficult.

As countermeasures for isolation, they check used IP addresses or connectivity. In the method to check used IP addresses, they verify whether or not they are executed in private address space that is often used in an isolated environment. The other method is to test connectivity to specific hosts or services on the

Internet in order to determine whether they are executed in an isolated environment or not. This method is widely used as it is very simple.

The method to check IP addresses is not a big problem, as it can be overcome by simply using non-private addresses for the experiment environment after eliminating effects on the external environment. On the other hand, issues with connectivity checks cannot be solved very easily.

In light of this, we decided to develop a mimetic Internet[7][8] that can mimic services and hosts targeted for connectivity checks, and cause false recognition of connectivity.

Malware and attack tools basically run as software on a certain executing environment such as PC. Therefore, any mechanism that operates on a PC and its OS which provides the executing environment could be detected by malware and attack tools easily. On the other hand, when mechanisms operate on a PC or network that are not in the executing environment, malware and attack tools can only check them through external observation. The mimetic Internet takes advantages of this relation, and places a VM that mimics services and hosts targeted for the connectivity check onto a network. The basic structure of the mimetic Internet is as follows:

1) Emulated well known sites
2) Emulated global services
3) Emulated local services
4) Emulated neighboring hosts
5) Emulated routes

3.3 Malware analysis testbed with mimetic Internet

The malware analysis testbed with mimetic Internet enables secure dynamic analysis of malware that feature functions to detect virtual or isolated environments in order to make it difficult to be analyzed. To combat these malware functions, it combines a method that secures the similar level of usability as virtualization technology by switching or renewing actual nodes, and a method that fools malware by using the mimetic Internet to prevent them from detecting that they are being analyzed in an isolated environment. The outline of the structure is shown in Fig. 4.

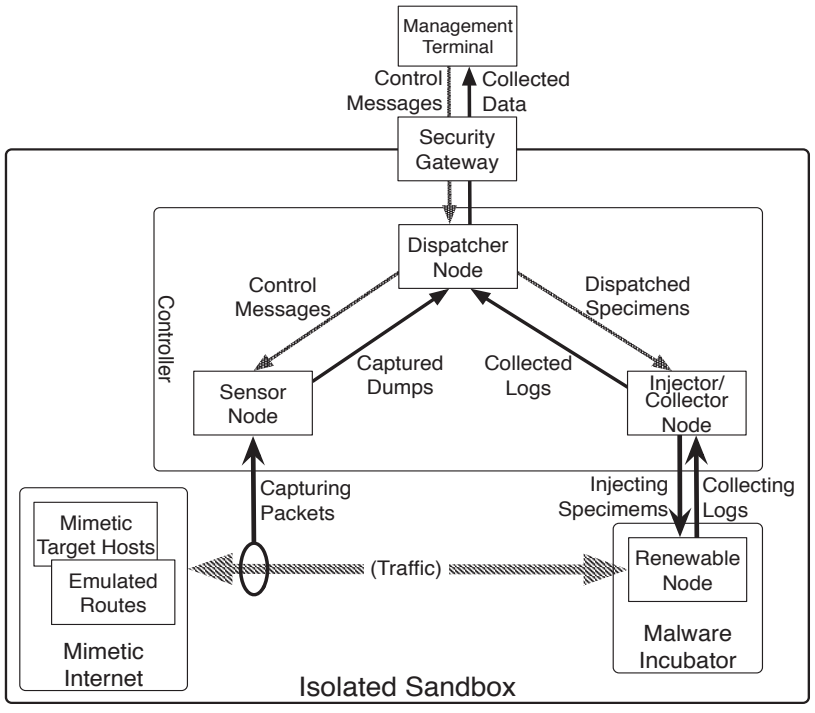

Fig.4 *Components of malware analysis testbed with mimetic Internet*

It consists of malware incubator which is an executing environment of malware based on renewable actual nodes, a mimetic Internet that functions as a mimicked Internet, controller nodes that control the malware incubator and mimetic Internet (hereafter, control node group), and a management terminal. All of these except the management terminal are in the isolated sandbox. Malware can be executed on the malware incubator, and the access to the Internet from the malware can be mimicked by the mimetic Internet to fool connectivity tests by the malware. The network for the experiment is physically separated, and the one for management is logically separated. The communication from the executing environment to the management network is completely disconnected when malware are executed. When it is required to collect experimental data or dispatch specimens, the executing environment of malware will be stopped first, and then restarted by the OS in a separate network boot beforehand. Therefore, the activities of the malware will not affect the control node group. Moreover, a security gateway that allows only the specific communication is installed between the control node group and management terminal to provide double isolation.

3.4 The outcome and application of the mimetic Internet

The isolated sandbox with mimetic Internet enables the achievement of both security and accuracy. The following activities have been implemented as an application of this sandbox.

1) Generation of datasets by sequential automatic analysis
2) Hands-on training

In terms of the generation of datasets by sequential automatic analysis, when malware and attack tools are deployed, they are executed in the sandbox with the mimetic Internet. The content of communications and executing memory, and various access logs at the time of the execution are obtained so that they can be retrieved as a dataset. The series of these processes are automated in this method (Fig. 5). The datasets generated by this method have

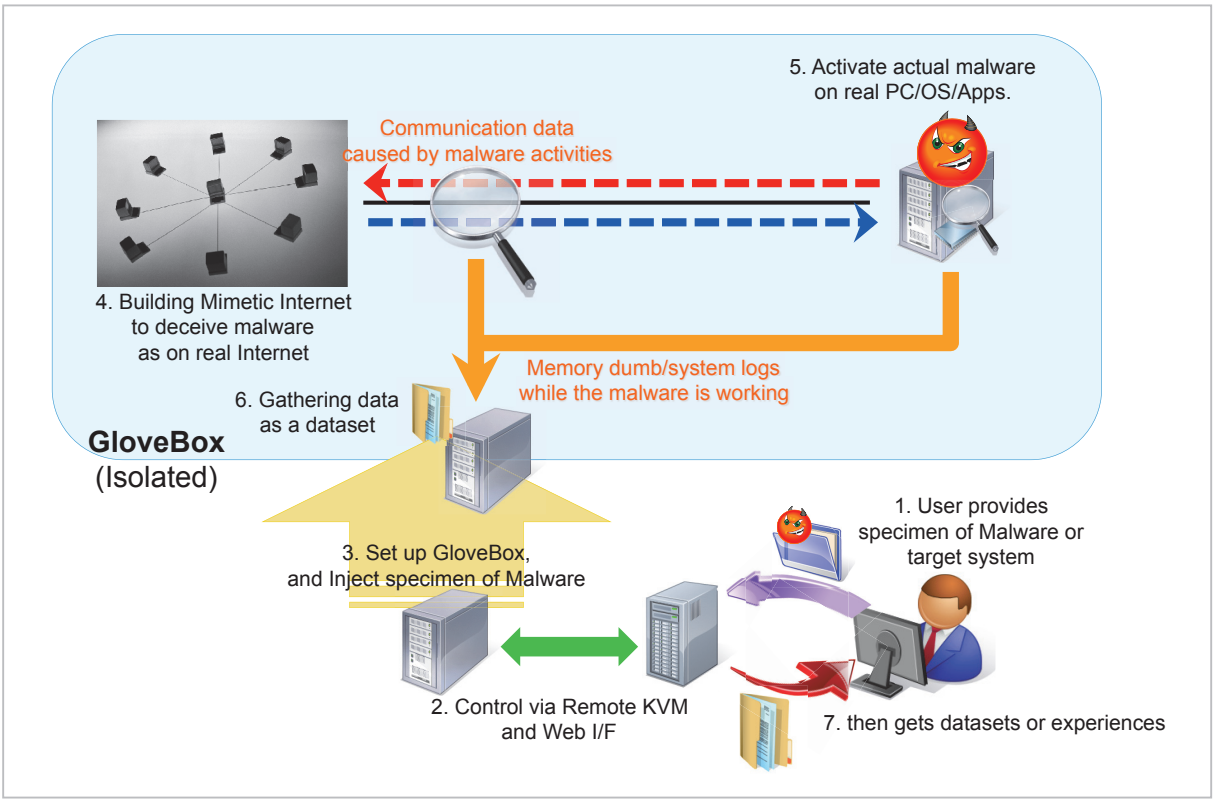

Fig.5 *Dataset generation using the sequencer*

been actually adopted as common datasets for research by some academic conferences.

An isolated environment is suitable for hands-on training, as it enables experience to be gained of risky activities safely. Based on this concept, "hands-on training for learning cyber security incidents"[9] has been conducted. In this training, multiple isolated environments were constructed on StarBED by using the same software used for the isolated sandbox with mimetic Internet, and the students actually experienced and reported various malware and attacks in the environment. With the use of about 15 pairs of isolated environments including spares, when specific content was selected from the provided contents, malware and attack tools would be dispatched to the isolated environment and executed in a similar way to sequential automatic analysis. The students would monitor the situation via the monitoring server within the isolated environment, and report the types of attacks and required countermeasures. This training for learning incidents was conducted for three years from 2008.

Research and development on mimetic Internet technology is continuing, now aiming to enable emulation of stereotypical services and hosts, construction of a more flexible environment for targeted attacks, and accurate emulation of the target services including a wide area service network. The mimetic Internet technology is a technology for projecting services, and therefore, it is expected to be used not only to fool malware and attack tools but also to enable accurate experiments in the future.

4 Issues and future prospects

It cannot be said that reproduction/emulation technologies of the wide area Internet or the mimetic internet have been completed yet. For example, although reproduction/emulation technologies require accurate observation of the Internet and accurate projection based on the observations in order for accurate emulation of inter-AS networks, it is not easy to achieve this in the current situation as it is based on so many presumptions. In addition, although it is possible to fool connectivity checks in the mimetic Internet, when downloads or commands from the external environment are required, it is necessary to fake these contents as well in order to realize accurate reproduction/emulation. Moreover, even though the aim of reproduction/emulation is to pursue "how close it can get to the real Internet," if it causes the cost of experiments and risks like the real Internet, there would be no point in the reproduction/emulation.

In order to solve these issues, we consider it is necessary to conduct software science research that examines how to project a distributed system to another distributed system, what capabilities are needed to define accurate projection (reproduction/emulation), what can be omitted, and so on. The accumulation of such research is considered to widen the usage from not only simple sandboxes or experiment environments but also for frameworks for evaluation of the whole ICT environment and scientific construction method, and therefore, further progress is expected.

5 Summary

In accordance with the progress of new ICT environments, subjects in the field of network security are becoming more complex. In these circumstances, it is important to have technologies that conduct experiments and verification by using the same software and hardware that are used in practice. Our reproduction/emulation technologies for the wide area Internet and to fool malware/attack tools are essential for such experiments and verifications, and they have achieved results in actual use cases.

We hope that the combination of practical verification/experiment that we aimed for and scientific verification/experiment will realize "security by design" that can ensure there is no network security issue in design and development phases in the future.

References

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

2 Toshiyuki Miyachi, Ken-ichi Chinen, and Yoichi Shinoda, ”StarBED and SpringOS: Large-scale General Purpose Network Testbed and Supporting Software,” Valuetools 2006, Pisa, Italy, ISBN 1-59593-504-5, Oct. 2006.

3 Mio SUZUKI, Hiroaki HAZEYAMA, Daisuke MIYAMOTO, Shinsuke MIWA, and Youki KADOBAYASHI, “Expediting experiments across testbeds with AnyBed: a testbed-independent topology configuration system and its tool set,” IEICE Trans. of Information and System, Vol. E92-D, Num. 10, pp. 1877–1887, Oct. 2009.

4 Shinsuke Miwa, Mio Suzuki, Hiroaki Hazeyama, Satoshi Uda, Toshiyuki Miyachi, Youki Kadobayashi, and Yoichi Shinoda, “Experiences in Emulating 10K AS Topology with Massive VM Multiplexing,” The First ACM SIGCOMM Workshop on Virtualized Infastructure Systems and Architectures (VISA'09), Aug. 2009.

5 Hiroaki Hazeyama, Ken Wakasa, and Youki Kadobayashi, “A consideration on deployment scenarios on a filed test of IP traceback in Japan,” IEICE Communication Society Technical Committee on Internet Architecture, Jul. 2008. (in Japanese)

6 MIWA Shinsuke, KADOBAYASHI Youki, and SHINODA Yoichi, “Design and Implementation of an Isolated Sandbox Used to Analyze Malware,” Journal of the National Institute of Information and Communications Technology, Vol. 55, Nos. 2/3, pp. 17–26, 2008, ISSN 1349-3205, Nov. 2008.

7 Shinsuke MIWA, Toshiyuki MIYACHI, Masashi ETO, Masashi YOSHIZUMI, and Yoichi SHINODA, “Design Issues of an Isolated Sandbox used to Analyze Malwares,” proceedings of Second International Workshop on Security (IWSEC2007), LNCS 4752 Advances in Information and Computer Security, ISBN 978-3-540-75650-7, pp. 13–27, Oct. 2007.

8 Shinsuke MIWA, Daisuke MIYAMOTO, Hiroaki HAZEYAMA, Daisuke INOUE, and Youki KADOBAYASHI, “Improving Isolated Sandbox using Fake DNS Server,” IPSJ, anti-Malware engineering Workshop 2008 (MWS2008), pp. 19–24, Okinawa, Oct. 2008. (in Japanese)

9 Shinsuke MIWA, Daisuke MIYAMOTO, Hiroaki HAZEYAMA, Shigeru KASHIHARA, Youki KADOBAYASHI, and Yoichi SHINODA, “Design and Implementation of Hands-on Training Environment for Learning Cyber Security Incidents,” IPSJ, Computer Security Symposium 2008 (CSS2008), pp. 929–934, Okinawa, Oct. 2008. (in Japanese)

(Accepted June 15, 2011)

MIWA Shinsuke, *Ph.D.*

Associate Director, Network Testbed Research and Development Laboratory, Network Testbed Research and Development Promotion Center/Senior Researcher, Security Architecture Laboratory, Network Security Research Institute

Networks Security