Managed Security System—A Mechanism that Centrally Reconfigures Associated Hosts Based on New Vulnerability Alerts

Fumiyuki Tanemo†*, Hideo Sakuma, and Masao Tanabe*

Abstract

To keep the network secure, system administrators must constantly be aware of every new vulnerability alert and manually apply the corresponding countermeasures in their systems. This article presents a managed security system that can simplify such tasks by centrally reconfiguring associated hosts when a new alert is received.

1. Increase in security risks

As the Internet becomes prevalent, the damage caused by illegal access and denial-of-service (DoS) attacks is increasing. When security weaknesses exist in the host computers or local networks connected to the Internet, problems such as information leakage, falsification, and system breakdown can be caused by various security attacks. Furthermore, rectifying the damage frequently requires troublesome effort.

2. Problems with current security management

Although security devices such as firewalls*1 and intrusion detection systems (IDS)*2 are somewhat effective against some network threats, most security attacks exploit the weaknesses of host computers such as software vulnerabilities or configuration errors. Therefore, it is necessary to correct the weaknesses on the host computers as soon as possible.

However, new security weaknesses are found every day and made public as vulnerability alerts. It is not easy for administrators to confirm whether there is a problem for each vulnerability alert in the host computers and apply the corresponding countermeasures to them continuously. Moreover, when the security measure is done inadequately, important functions on the host computer may not operate or a more serious security problem might occur. Therefore, sufficient knowledge and experience are required to apply the security countermeasures appropriately. We also believe that many enterprises have too few engineers skilled in the necessary security management, which increases the damage caused by security breaches.

3. Managed Security System (MSS)

Managing the security of the host computer requires inspecting security weaknesses on it and applying necessary countermeasures to it accurately and efficiently. To make such management easier, we are undertaking research and development of a managed security system (MSS) that centrally diagnoses and reconfigures all associated host computers based on vulnerability alerts (Fig. 1). It first confirms whether the host computer has the security weaknesses corresponding to the registered vulnerability alerts. Then, if these weaknesses are found, the system executes the corresponding countermeasures, such as changing the host computer's configuration or the policies of the firewall and IDS. It also aims to select and execute an appropriate action within the

† NTT Information Sharing Platform Laboratories
Musashino-shi, 180-8585 Japan
E-mail: tanemo.fumiyuki@lab.ntt.co.jp

*1 Firewall: a system that defends internal networks against network threats by controlling the reachability of network packets according to a policy decided beforehand. The firewall passes only packets that are permitted by the policy and blocks the packets prohibited by the policy.

*2 Intrusion detection system (IDS): a system that detects security attacks or malicious acts and alerts network manager by continuously observing packets on the network or auditing user behavior on host computers. In general, an IDS has a policy that defines what it should inspect from a large history of attack patterns.

range available that does not inconvenience network users.

4. Vulnerability alert

Figure 2 shows an example of a general vulnerability alert. This section surveys the main aspects of vulnerability alerts and explains their usage in MSS.

4.1 Name, details, effects

These are general descriptions of each vulnerability alert. They mention the target system, the effects of the vulnerability, level of emergency, and some other technical information. The system administrator often judges whether or not his/her system has security weaknesses based on this information. MSS provides this information in plain English, so the system administrator can understand it easily.

4.2 Target software, version

In general, each vulnerability alert lists the version of the target software and the operating systems that will and will not be affected. One can judge whether or not there is a weakness in the software from this version information. MSS registers the software name and version information as searchable items corresponding to the vulnerability alert in the database, which allows the system to check the vulnerability of the host computer.

4.3 Countermeasures

Most countermeasures described in vulnerability alerts can be classified into the following six categories.

1. Upgrade the version
2. Install a patch

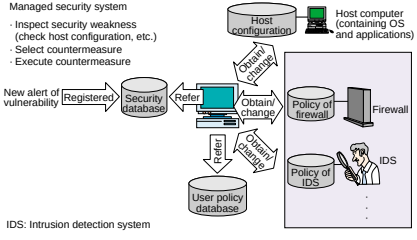

Fig. 1. Conceptual framework of managed security system (MSS).

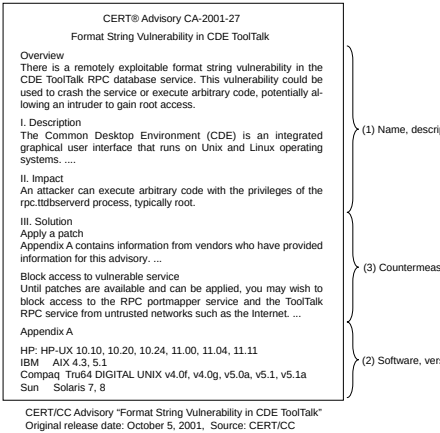

Fig. 2. Example of vulnerability alert.

Table 1. Example of data in security database.

Name	Description	OSs		Versions
Format string vulnerability in CDE ToolTalk	There is a remotely exploitable format string vulnerability in the CDE ToolTalk RPC database service.	HP IBM Compaq Sun	HP-UX AIX Tru64 Solaris	HP-UX 10.10, 10.20, 10.24, 11.00, 11.04, 11.11 AIX 4.3, 5.1 Compaq Tru64

Patches	Countermeasures	Executable script
Solaris5.8 110286-04 Solaris5.7 107893-15 Solaris5.6 105802-16	(1) Apply a patch (2) Block access to vulnerable service	RUN chmod 0 rpc.ttdbserverd

3. Modify the configuration file
4. Modify file permissions
5. Modify program execution options
6. Terminate service

MSS maintains the countermeasures as executable internal control information to automate the actions. Table 1 shows examples of the database items corresponding to some vulnerability alerts in MSS.

5. Security functions

MSS provides two main security functions: setting the initial security level and updating security based on recent vulnerability alerts.

5.1 Initial security setting function

To make the user network secure, the user policy should be defined first and each host computer should be diagnosed and reconfigured based on existing vulnerability alerts. Moreover, firewalls and IDSs should be set according to the user policy.

MSS provides the initial security setting function that does this along with user network registration by the user him/herself. The sequence of the initial security setting function is as follows.

(1) Register the network information regarding the user network (addresses and domains, etc.) and the user policy.
(2) Register each host computer's initial configuration
(3) Set the policy of firewalls and IDSs.
(4) Extract the host configuration from each host computer.
(5) Check the host configuration based on vulnerability alerts.
(6) If the host configuration has security weaknesses, select and apply countermeasures based on the user policy and host computer's initial configuration to the host computer.

The user policy defines services such as WWW, DNS, and mail, which the host computer provides. Firewalls and IDSs are configured based on this policy. Before selecting and applying countermeasures, MSS extracts each host computer's actual configuration, which includes the operating system's configuration files and a list of security patches that have already been applied. The host configuration to be extracted is stored in the database, and the system can diagnose host computers by checking this host configuration upon each vulnerability alert.

5.2 Security update function

To keep the user network secure, the system administrator must respond to all vulnerability alerts immediately. To make this possible, MSS provides a security update function that diagnoses and reconfigures the managed host computers as soon as a new vulnerability alert is registered. Although vulnerability alerts must be registered by the administrator, once one has been registered in the security database, the selection and application of countermeasures based on it is performed automatically on all associated host computers simultaneously. The sequence of the security update function is as follows.

1. Register a new vulnerability alert in the database.
2. Extract the host configuration from each host computer.
3. Check the host configuration based on the vulnerability alert.
4. If the host configuration has security weaknesses, select and apply countermeasures to the host computer.

MSS provides a graphical user interface that lets the system administrator check the history of security updates for managed host computers.

6. Architecture

MSS consists of two main subsystems: the gateway-agent module can keep many host computers secure and host-agent modules are installed on the managed host computers. It can cooperate with exist-

ing firewalls and IDSs. Figure 3 shows the architecture of MSS.

6.1 Gateway-agent module

This module, which is usually installed on a particular management system, diagnoses all managed host computers. If a security weakness is found in any host computer, it directs the host-agent of the host computer to execute the corresponding countermeasures. It also has a security database for storing vulnerability alerts.

The system administrator can set policies for firewalls and IDSs and diagnose and reconfigure the managed host computers by setting the user network information and the user policy on this gateway-agent module. When a new vulnerability alert is published, the administrator registers it in the security database of the gateway-agent module, so the module checks host computers according to the alert.

6.2 Host-agent module

MSS needs a host-agent module installed on each host computer that is managed by the system. The module executes some processes under the instructions of the gateway-agent module on the management system, including sending the host configuration to the gateway-agent module and executing countermeasures distributed by the gateway-agent module.

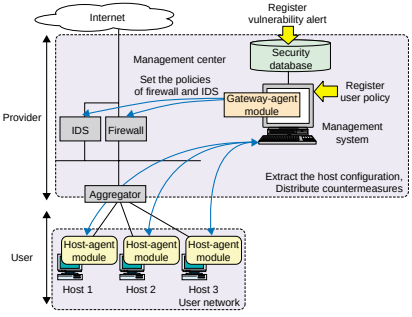

Fig. 3. Architecture of MSS.

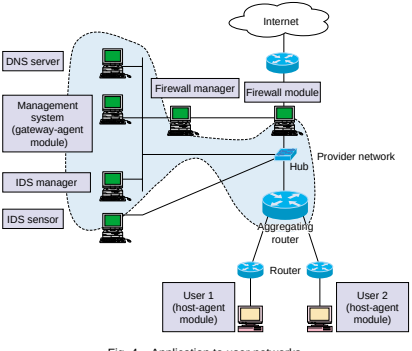

Fig. 4. Application to user networks.

7. Application to user networks

Figure 4 shows a network topology example in which MSS is applied to user networks. It can manage host computers running Windows 2000 Server or Solaris as the managed host computers. After configuring the initial minimum settings of a host computer including network settings, the system administrator introduces the host-agent module into the host computer. Next, the system administrator sets the initial security of the gateway-agent module on the management system. Thus, the policies of firewalls and IDSs are set according to the user policy, and the host computer will be made secure by executing each countermeasure that the gateway-agent module distributes.

When a new vulnerability alert is found and the system administrator registers it in the security database,

the security update function is executed on the system, and the action corresponding to the problem is achieved. This execution makes the security states of all host computers up-to-date.

8. Future development

The current version of MSS can be applied only to a limited number of operating systems, server software types, firewalls, and IDSs, so we plan to expand the target system coverage and generalize the system by making a cooperation protocol between modules that are standardized in the future.

Fumiyuki Tanemo
Research Engineer, NTT Information Sharing Platform Laboratories.
He received the B.S. and M.S. degrees in information engineering from Nagoya University, Nagoya in 1991, 1993, respectively. In 1993, he joined NTT Network and Information Systems Laboratories, Tokyo, Japan. In 1999, he joined NTT Information Sharing Platform Laboratories. He is a member of the Information Processing Society of Japan and IEEE Computer Society.

Hideo Sakuma
NTT Information Sharing Platform Laboratories.
He received the B.S. degree in communication engineering from Tokai University, Tokyo in 1995. In 1988, he joined NTT, Tokyo, Japan. In 2001, he joined NTT Information Sharing Platform Laboratories.

Masao Tanabe
Senior Research Engineer, NTT Information Sharing Platform Laboratories.
He received the B.S. and M.S. degrees in communication engineering from Waseda University, Tokyo in 1985 and 1987, respectively. In 1987, he joined NTT, Tokyo, Japan. In 1999, he joined NTT Information Sharing Platform Laboratories. He is a member of the Institute of Electronics, Information and Communication Engineers of Japan and IEEE Computer Society.