Information-Sharing System for Vulnerability Information Dissemination in Large-Scale Organization

Tohru Sato and Jumpei Watase

NTT Information Sharing Platform Laboratories
3-9-11, Midori-cho, Musashino-shi, Tokyo 180-8585, Japan
sato.tohru@lab.ntt.co.jp
watase.jumpei@lab.ntt.co.jp

Abstract: An organization has to deploy the capability to respond to vulnerabilities in products for which the organization is responsible. For a proper vulnerability response practice, disseminating vulnerability information to appropriate departments in an organization is critical. Some technologies for vulnerability information exchange have been presented. However, vulnerability information dissemination in a large-scale organization still has critical problems, mainly in management and maintenance of contact point information and confidentiality. To address these issues, we developed an information sharing system that utilizes a real-world social network in an organization and centralized information storage by means of a web application server. Our developed system makes information dissemination and organizational risk management more efficient and appropriate.

1 Introduction

To maintain the security of computer networks, dealing with vulnerabilities related to information technology products such as servers, routers, application software, and terminal devices, for example, is critical.

For proper vulnerability handling in an organization, deploying a vulnerability-information-sharing mechanism in an organization is necessary. The mechanism should include the capability to send vulnerability information in a timely manner to the appropriate product owners who are in charge of products/services.
The goal of our work is to integrate a system that comprehensively manages vulnerabilities. Our developed system is named “Security Information eXchange Infrastructure (SIXI)”. SIXI enables an organization to distribute vulnerability information in a timely and secure manner to product owners in an organization who are responsible for their products, systems, or services. Furthermore, SIXI also provides a vulnerability response status tracking function. This function collects information about the status of the ability to respond to vulnerabilities and shows security managers such as the Chief Security Officer (CSO) how many vulnerabilities have been fixed and how many vulnerabilities still exist in the organization.

In this paper, we discuss some problems in the vulnerability handling process in a large-scale organization. We also present a method and a conceptual diagram of our developed system. Target users of this system are security managers, system developers, and system operators in an organization.

1.1 Types of vulnerability information

In consideration of the confidentiality level of vulnerability information, we classify vulnerabilities into the following two types.

(Type A) Public

The word “Public” is defined by the Vendor Special Interest Group (Vendor SIG) of the Forum of Incident Response and Security Teams (FIRST) as “entities outside of a vendor” [1]. They also defined “vendor” as “private or public corporations, partnerships or other for-profit business entities”.

This type of vulnerability information has been disclosed to the public. For instance, some security information websites or mailing lists disclose vulnerability information of this type. This type of information does not require a high-confidentiality level because the information has already become available to everyone.

However, if program codes that exploit certain vulnerabilities are published, the vulnerability information should be shared among developers and operators who are in charge of affected systems as soon as possible to facilitate early responses and minimize the potential damage.

(Type B) Undisclosed

This type of vulnerability information is secretly shared among one or more specific entities, such as vendors and coordinators, before being disclosed to the public. Vendors who supply affected products are required to respond, which includes developing and releasing security updates. They have to maintain the confidentiality of the undisclosed vulnerability information during their response because the inadvertent public disclosure of such information puts all vendors’ customers and product users at risk of a potential exploitation.
In this context, we discuss the system requirements that deal with the two above-mentioned types of vulnerability.

2 Related Work

The efficient exchange of vulnerability information using the standardized description format is one of the major problems in responding to vulnerabilities.

The Extended Incident Handling Working Group (INCH WG) [2] in the Internet Engineering Task Force (IETF) [3] defined the essential vulnerability information descriptions as the following [4].

v1) The affected product

v2) The nature of the problem

v3) The possible impact if the vulnerability and/or exploit were, accidentally or maliciously, triggered

v4) Available means of remediation

v5) Disclosure restrictions

Some description formats for efficient exchange of vulnerability information using the above description have been developed among security professionals.

Another major format is Mitre's Common Vulnerabilities and Exposures (CVE) [5]. CVE is the de facto standard for storage of vulnerability information. Mitre also introduced another format named the Open Vulnerability Assessment Language (OVAL) [6]. OVAL is the common language for security experts to discuss and agree upon technical details about how to check for the presence of vulnerabilities on a computer system.

The Organization for the Advancement of Structured Information Standards (OASIS) [7] developed another format named the Application Vulnerability Description Language (AVDL) [8]. AVDL was a candidate of a security interoperability standard to create a uniform manner of describing application security vulnerabilities using XML, but this effort has been deactivated at present.

The Open Security Project (Opensec) [9] provides another format named the Advisory and Notification Markup Language (ANML) [10]. ANML is an XML-based specification for describing advisories and other types of notifications. ANML intends to resolve the inconsistent use of terminology by software vendors in their advisories and enable applications to read these advisories easily.

In addition, a lot of applications to operate these formats have also been proposed.
While various vulnerability description formats and applications have been proposed by many security professionals, some critical issues remain in vulnerability handling in an organization. In the next section, we discuss these issues.

3 Vulnerability Handling Issues

In this section, we discuss the following four issues. These issues should be resolved to handle vulnerabilities properly, in a timely manner, and securely in an organization.

Issue 1. Notification to everyone who needs vulnerability information

To distribute vulnerability information among all developers and administrators who are responsible for the affected product in the organization, it is necessary to manage the list of the product developed/operated in the organization and the person in charge of these products. When vulnerability information is reported, the capability to search for appropriate people quickly is significant.

However, management and maintenance of contact point information including the components of products and their product owners is difficult work in a large-scale organization. One of the reasons is that a large organization owns a lot of products and there are many employees.

Another reason is that the components of products and their owners are frequently changed by the startup of a new development project, changes in product specifications, or reassignment of their owner/section. Even if the information has been collected successfully at some point in time, the information dynamically changes and becomes obsolete along with company activities.

Therefore, effective management methods that can maintain components of products and their owners are required.

Issue 2. Accurate understanding of product composition

Accurate understanding of product composition is also difficult work because products consist of a complex combination of operating system, middleware, application software, library, and protocol for example. Products are also supplied in various forms such as commercial, or OEM. In addition, in the case of outsourcing product development or failure to change owners of products, owners cannot understand the entire composition of their products.

An effective method of extracting product component information is required.

Issue 3. Observation of vulnerability information distribution and response status

To observe that vulnerability information is appropriately shared with the right person, a method to monitor the information distribution status in an organization is required.
Furthermore, in the case of dealing with undisclosed vulnerability information, distribution control is also needed to control confidentiality of the information. If the information were exposed to outsiders such as a hacker community, the risk of a potential exploitation would increase.

Similarly, existing vulnerabilities of products in an organization and the status of responding to those vulnerabilities should be monitored. However, the conventional distribution method by means of E-mail makes response status tracking difficult.

A monitoring method that enables a security management section to track the response status effectively is required.

**Issue 4. Centralized control of vulnerability information**

Distributing undisclosed vulnerability information by E-mail has another problem. The E-mails are stored in many unspecified computers by repeated forwarding. In this case, the risk of leaking undisclosed vulnerability information becomes higher.

Centralized control of vulnerability information in an organization is also required to reduce the risk of information leakage.

**3.1 Conventional Practice of Vulnerability Information Dissemination**

We analyze some vulnerability handling cases that we experienced and examine our conceptual design.

In one case, we dealt with undisclosed vulnerability information. In maintaining the confidentiality of undisclosed vulnerability information, we were careful to distribute only to product owners who were thought to be responsible for affected products.

At the beginning, we selected people who were eligible to receive the information according to the following criteria.

(A) Product owners in charge of affected products.

(B) People who are in charge of managing the point of contact of development sections and operations sections.

(C) People who have wide human networks in related fields.

(D) People who have extensive knowledge of related technologies.

However, we could not distribute vulnerability information to all people who needed the information in the first stage. Then, we asked the above selected people to introduce us to other people who should be notified. The vulnerability information was distributed in our organization more widely and properly due to the introductions or information forwarding by the selected people.
In this case, to confirm that the introduced people are truly eligible to receive the information, we checked who received the information. Furthermore, we also checked their response status. This work took a huge amount of time and human resources.

We assume that efficient organizational vulnerability handling requires a system that has the following capabilities. These are 1) to identify “what types of vulnerability information each product owner needs”, 2) to forward vulnerability information to individual users securely, and 3) to track distribution and response status.

The next section discusses our approaches to solve these problems.

4 SIXI System

4.1 Overview

The objective of our developed system is to provide effective functions for organizational vulnerability handling. The main features of SIXI are an efficient management of contact point information and dissemination of vulnerability information based on a social network in an organization.

SIXI consists of three subsystems. The first subsystem is the Contact Point Management (CPM) subsystem. The function of this subsystem is to collect contact information of individual product owners such as developers/administrators. In addition, CPM also provides a function to search their contact information.

The second subsystem is the Vulnerability Information Management (VIM) subsystem. One function of this subsystem is to collect vulnerability information and information about the distribution and status of a response by product owners. Other functions are to summarize that the distribution and response status information, and to provide reports on various aspects of vulnerability handling.

The third is the Contact Point Information Update (CPIU) subsystem. This subsystem provides a mechanism that autonomously updates the Contact Point Information according to the vulnerability information distribution process. This mechanism enables more efficient maintenance of the Contact Point Information.

The major components in SIXI are depicted in Figure 4.1. In the following sections of this paper, the details and features of the two subsystems are discussed.
4.2 CPM Subsystem

The purpose of the CPM subsystem is to provide a capability to search for product owners quickly and accurately. This capability is a solution to the above-mentioned problems of section 3.

The subsystem deals with two types of information. One is Contact Point Information. The other is the Keyword Template. The components in the CPM are depicted in Figure 4.2. The definitions of the two types of information are as follows.
4.2.1 Contact Point Information

Contact Point Information is a user’s profile to be used to search and determine the product owners who should receive certain vulnerability information.

The following components are described:

  c1) Product owner’s name
  c2) Product owner’s contact information
  c3) Product name that he/she is responsible for
  c4) Product composition described with technical keywords
  c5) Other technical keywords that the owner has an interest in or to which he/she pays attention

Technical keywords describe product composition such as operating system, middleware, application software, libraries, and protocols.
Contact Point Information will be maintained by product owners themselves. For example, if an owner is responsible for a certain product, the owner will store his/her product name and composition with his/her name and contact. The owner also stores other technical keywords in which he/she is interested.

4.2.2 Keyword Template

The Keyword Template is a set of technical keywords to offer choices of technical keywords that can be registered to product owners. In other words, keywords included in Contact Point Information about a certain product owner are a subset of the Keyword Template.

The purpose of the Keyword Template is to store technical keywords easily.

4.2.3 Keyword Storage Assistant

The Keyword Storage Assistant is a function that extracts product composition information as a list of keywords. The function extracts technical keywords from various resources by comparing them with the Keyword Template.

One of the simple ways to extract keywords is to compare the Keyword Template with specifications of a product document. If some software agents or devices that scan hardware to investigate composition of a certain product were available, keywords can also be extracted by comparing a scan result with the Keyword Templates.

4.2.4 Keyword Template Maintenance

A keyword that should be listed in the Keyword Template will change as a new protocol, middleware, and product appear. Therefore, the Keyword Template requires continuous maintenance.

We provide two methods to update the Keyword Template for maintenance. One method is a vulnerability information collector like a security web site crawler. The method observes vulnerability information including new technology keywords in some security information resources such as CVE. The crawler can observe new technology keywords and add them to the Keyword Template because security information resources provide vulnerability information described in a specific format.

Another method is manual editing of the Keyword Template by any product owner who noticed that he/she needs to register more keywords. To prevent an improper keyword from being added in this way, an approval process performed by a security manager is introduced.
4.3 Vulnerability Information Management Subsystem

The VIM subsystem provides services to handle vulnerability in the centralized management manner. This is for centralized management of vulnerability information distribution and the status of responding to vulnerabilities. In addition, information is prevented from proliferating toward many unspecified PCs. The components in the VIM are depicted in Figure 4.3.

The subsystem has the following features.

![Figure 4.3: Internal Structure of Vulnerability Information Management Subsystem](image)

4.3.1 Vulnerability Information Collection

Public vulnerability information can be collected by the Vulnerability Information Collection Module from external sources such as a security information web site and/or mailing lists. The module periodically observes new vulnerability information.
If new vulnerability information is published by a specific format such as CVE, the module will automatically parse the information and store it in an internal vulnerability information database.

Other vulnerability information without a specific format is stored by SIXI operators manually.

4.3.2 Vulnerability Information Notification

According to matching “description of affected products included in a vulnerability information (v1)” and “product composition keywords included in Contact Point information (c4, c5)”, vulnerability information is only sent to the person in charge of affected products.

SIXI provides a view using a web application interface instead of E-mail of the vulnerability information sent to the person in charge of affected products. Product owners can view the vulnerability information using generic web browsers without saving the information to their hosts.

4.3.3 Information Distribution Status Tracking

Vulnerability information distribution status can be confirmed by extracting an access history of the vulnerability information database. The status would be disclosed to all SIXI users. Knowing which information has/has not been read by each product owner is possible.

4.3.4 Vulnerability Information Forwarding

The information forwarding function that enables any person to forward certain information to other product owner(s) is introduced. A person who has noticed that another person who should be notified has not been informed could transfer the information to that person. In this manner, we expect the information would be delivered to the appropriate people more comprehensively.

However, the forwarding of undisclosed vulnerability information should be restricted to people in charge of affected products. To hide undisclosed information from the public, we introduce a label, which represents confidentiality as metadata of vulnerability information. We also implement a process to audit and approve the forwarding of undisclosed vulnerability information.

When the forwarding of undisclosed vulnerability information with high confidentiality is required in the SIXI system, the forwarding will be executed after approval by the security manager in the organization.
4.3.5 Response Status Management

SIXI has a Response Status Database, Response Status Input Module, and Response Status Tracking Module. The Response Input Interface accepts status information about responses to certain vulnerability information from each product owner. The status information is stored in the Response Status Database.

The Response Status Tracking Module shows how many vulnerabilities remain in an individual product and summarizes how much risk is still in the organization in performing organizational risk management.

4.4 CPIU Subsystem

The CPIU subsystem is responsible for only one function. The subsystem helps the interaction between CPM and VIM. This subsystem updates Contact Point Information of a product owner regarding vulnerability information forwarding action as a trigger of updating. The object to be updated is the keyword set included in Contact Point Information about each product owner.

When a product owner receives the vulnerability information, he can confirm the distribution status of the information in his organization. If he noticed the information was not reported to colleagues of his acquaintances who should know that information, he can forward or transfer that to them. People who are notified by the forwarding can input their evaluation of the usefulness of the information to SIXI. Then, SIXI interprets the input and autonomously adds the related keywords to the product owners’ Contact Point Information.

By means of the above mechanism, SIXI can maintain Contact Point Information effectively. The function to edit Contact Point Information manually is also provided.

5 Discussion

Correlation between contact point information and response action/status by the Contact Point Information Update Subsystem enables an organization to manage and maintain information that indicates who needs what kind of information in the organization. As a result, security managers can easily disseminate the critical vulnerability information to appropriate parts of the organization more comprehensively. This can resolve the issue 1 in chapter 3.

The introduction of Keyword Template and Keyword Registration Assistant capability confers an advantage in understanding products composition easily. This is a solution to issue 2.
The centralized management of vulnerability information and response status by means of the Vulnerability Information Management subsystem enables security manager to observe the progress of the response and the total amount of the risk remaining in the organization. This could be a solution of issues 3 and 4.

6 Conclusion and Future Work

In this paper, we have discussed some issues of organizational vulnerability handling. To handle vulnerability information appropriately, managing Contact Point information efficiently and reducing the risk of leaking undisclosed vulnerability information is significant.

We presented a vulnerability information management system named Security SIXI as an approach to resolving these problems.

By using the social network in an organization, SIXI effectively maintains Contact Point Information. Furthermore, SIXI makes vulnerability information notifications more secure by centralized control of vulnerability information. The centralization of individual response status information enables the security risk of the organization to become more visible.

However, to make organizational vulnerability handling better and more appropriate, a mature organizational culture and climate for vulnerability handling is necessary besides vulnerability management system like SIXI.

In the future, we will execute a field test of this system for evaluating the effectiveness of the approach. We will also examine the effect of various kinds of peer-recommendation using social networks such as “a person needs to know about this particular vulnerability” and “a person manages (owns) this type of system”. Furthermore, we have a plan to evaluate the interoperability between SIXI and other current vulnerability information management technologies.

Bibliography