Linking Cybersecurity Knowledge: Cybersecurity Information Discovery Mechanism

Takeshi Takahashi, Youki Kadobayashi, Yuuki Takano
* National Institute of Information and Communications Technology, Tokyo, Japan
Email: takeshi_takahashi@nict.go.jp

I. Introduction

To cope with increasing amount of cyber threats, organizations need to share cybersecurity information beyond the borders of organizations, countries, and even languages. Assorted organizations built repositories that store and provide XML-based cybersecurity information on the Internet. Among them are NVD [1], OSVDB [2], and JVN [3], and more cybersecurity information from various organizations from various countries will be available in the Internet. However, users are unaware of all of them. To advance information sharing, users need to be aware of them and be capable of identifying and locating cybersecurity information across such repositories by the parties who need that, and then obtaining the information over networks.

This paper proposes a discovery mechanism, which identifies and locates sources and types of cybersecurity information and exchanges the information over networks. The mechanism uses the ontology of cybersecurity information [4] to incorporate assorted format of such information so that it can maintain future extensibility. It generates RDF-based metadata from XML-based cybersecurity information through the use of XSLT. This paper also introduces an implementation of the proposed mechanism and discusses extensibility and usability of the proposed mechanism.

II. Proposed Mechanism

A. Roles

The proposed mechanism introduces four distinct roles. **Discovery Client** retrieves cybersecurity information by communicating with one or more arbitrary Discovery Servers. **Discovery Server** provides assistances to find proper Information Source to Discovery Clients by communicating with multiple Registries, aggregating information from them, and then delivering that to the Discovery Client. **Registry** manages an internal registry that contains the metadata of Information Sources by communicating with them. **Information Source** provides cybersecurity information that is described in XML format by communicating with Registries.

B. Information Structure

A Registry uses an RDF-based internal repository to maintain the metadata list of cybersecurity information residing in Information Sources. The metadata is generated by accessing Information Sources and extracting needed information from them, as described in Section II-C. Note that the level of details of the metadata depends on implementation, but URI that can uniquely identify an Information Source is needed The repository uses the information structure described in Table I, which separates information category and content description format. The ontology of cybersecurity operational information proposed in [4] is used for the category, whereas various industry specifications are used for the content description format, so that it can maintain future extensibility and compatibility with future such specifications.

TABLE I
Incorporating assorted cybersecurity information formats

Components of the ontology		Specifications
User Resource DB		AI, ARF, CRF
Provider Resource DB		-
Incident DB		CEE, IODEF
Warning DB		IODEF
Cyber Risk KB	Vulnerability KB	CVE, CVRF, CWE
	Threat KB	CAPEC, MAEC
Countermeasure KB	Assessment KB	CVSS, CWSS
	Detection/Protection KB	OCIL, OVAL, XCCDF
Product & Service KB	Version KB	CPE
	Configuration KB	CCE

DB: Database, KB: Knowledge Base

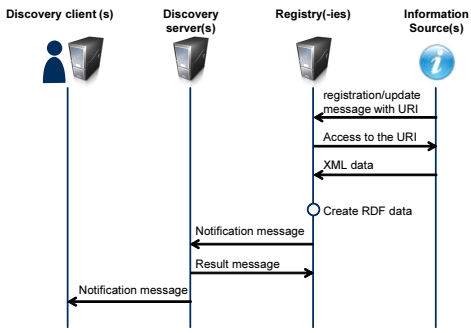

Fig. 1. Information publishing

C. Protocol

Information Publishing, described in Figure 1, is a procedure for an Information Source to publish its XML-based cybersecurity information. An Information Source sends registration message, which contains the information's URI, category, and allowed access method (e.g., http), to a Registry. The Registry then accesses to the URI by using one of the methods, receives the information, and converts it into RDF-based metadata by running XSLT. It then generates and sends Notification message to its Discovery Servers, which may also send the message to Discovery Clients so that they can receive any security information updates as soon as possible.

Server Registration and cancellation are procedures for a Discovery Client to use a Discovery Server. A Discovery Client sends join message to a Discovery Server it wants to use. The Discovery Server then sends result message with the category and supporting format information. Though this paper proposes a single category following the ontology proposed in [4], the procedure allows to use different categories by embedding different category information in the Result message, so that the proposed mechanism can provide future extensibility. When the Discovery Client wishes to stop using the server,

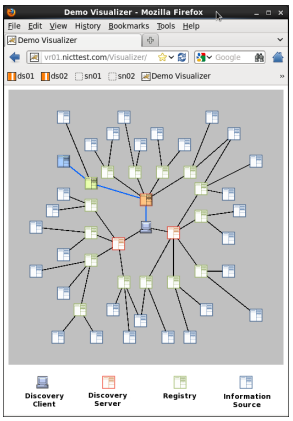

Fig. 2. Network view

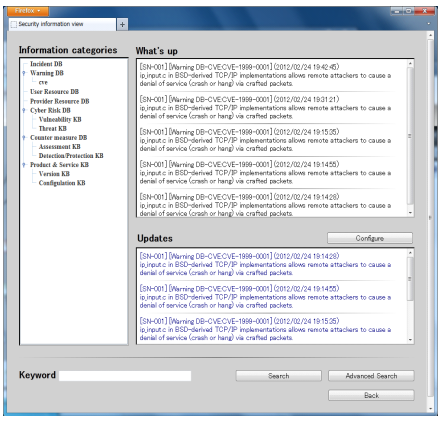

Fig. 3. Search view

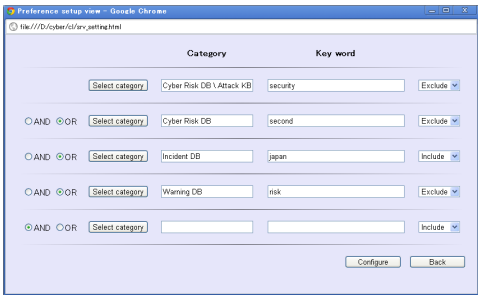

Fig. 4. Advanced search view

the client may send leave message to the server.

Information Retrieval is a procedure for a Discovery Client to retrieve and obtain cybersecurity information. A Discovery Client sends query message to a Discovery Server, which forwards the message to all of the Registries it communicates with. Each of them then retrieves its internal repository and creates and sends a Result message. The Discovery Server receives the messages from all of the Registries, aggregates them into one, ranks and reorders the candidate Information Sources, and then embeds the information into a new Result message, which is sent back to the Discovery Client. The Discovery Client chooses one Information Source among the candidate Information Sources that is listed in the message. Then it accesses to the Information Source's URI using the allowed access method and obtains the XML information stored inside the Information Source.

III. Prototype

A. Implementation

A prototype of the proposed mechanism is implemented with Java on Linux CentOS. It uses a certificate provided by Jetty [5] to certify the Information Source. Its Registry simply converts all the tags of the Information Sources' XML information into RDF-based metadata by using XSLT, though meticulous metadata extraction mechanism could be implemented, if needed. Sesame [6], an implementation of SPARQL engine, is also used. The proposed mechanism allows Information Sources to support arbitrary transport protocol for accessing itself, but this implementation supports only HTTP, HTTPS, and WebSocket.

During the retrieval procedure, the Registry needs to rank candidates of Information Sources. Though the ranking algorithm is outside the scope of the paper, the implementation adopted a simple algorithm as follows. The algorithm counts the number of keywords available in a tag, and then divide the number by the total number of the words in the tag. Then it assigns high rank on the entry that has higher resultant value. If the same value could be found, the one with older registration date gets higher rank.

B. Demonstration

Figure 2 provides a network view, which demonstrates a case of an Information Source publishing its cybersecurity information. This demonstration is conducted over a network consisting of 1 Discovery Client, 3 Discovery Servers, 15 Registries, and 30 Information Sources, all of which are running over different virtual machines.

Figure 3 depicts the search view of Discovery Clients. It provides category-based search, keyword search, and security information update. The **keyword search** is in the bottom part of Figure 3. Users can enter arbitrary keyword in the bottom text box and run search by clicking on the "Search" button. Users may enjoy more sophisticated searches by clicking on the "Advance Search" button in the figure and moving to the advanced search as can be seen in Figure 4, where they may specify the target tags of the retrieval. When specifying the tags, the users may lookup the available tags by clicking on the "Select category" button. The Discovery Client can provide the category information since it went through the server registration procedure as described in Section II-C, where it received the information from its Discovery Server. Users can simply select the tag, then identify the keyword in the advanced search.

IV. Discussion and Future Works

The proposed mechanism incorporates various formats defined by assorted industry specifications, which are yet to be developed further. Its metadata structure is designed so that it can maintain extensibility. In case current information format becomes obsoleted, any new specification could be introduced as a means to describe information of the types defined by the ontology. In this way, the changes are kept minimal. Even more, the ontology itself could be extended though the ontology is designed so that these won't happen in the near future. In addition to the extensibility, this mechanism needs to be scalable to accommodate large volume of cybersecurity information. This evaluation must be done as our future work.

The proposed mechanism enables users to search cybersecurity information across assorted repositories including NVD and JVN. The current implementation is, however, run on the Intranet that is isolated from the Internet since it does not consider security of the system. For instance, it may suffer from impersonation or man-in-the-middle attacks, which may cause severe security incidents. Though this paper excluded the security issue from this paper, our future work considers this issue and integrates with assorted security techniques to reinforce the mechanism's security level.

References

[1] National Institute of Standards and Technology, "National Vulnerability Database (NVD)," http://nvd.nist.gov/, Mar. 2011.
[2] The Open Source Vulnerability Database (OSVDB), http://osvdb.org/, June 2011.
[3] JPCERT/CC and IPA, "Japan Vulnerability Notes," http://jvn.jp/, Mar. 2012.
[4] T. Takahashi, Y. Kadobayashi, and H. Fujiwara, "Ontological approach toward cybersecurity in cloud computing," in *SIN*, 2010.
[5] M. B. Consulting, "Jetty," http://jetty.codehaus.org/jetty/, Mar. 2012.
[6] "openRDF.org," http://www.openrdf.org/, Mar. 2012.