The Design of XML-Based Internet Security Integrated System Architecture

Kwang H. Kim, Tae-Kyung Kim, Dong S. Kim, and Tai M. Chung

Real-Time Systems Laboratory,
School of Electrical and Computer Engineering,
Sungkyunkwan University,
300 Cheoncheon-dong, Jangan-gu, Suwon-si, Gyeonggi-do, 440-746
{byraven, tkkim, dskim, tmchung}@rtlab.skku.ac.kr

Abstract. To protect networks from intrusions and attacks, many vendors have developed various security systems. Managing these systems individually demands much work and high cost. To decrease these overheads, we have developed an Internet Security Integrated System (ISIS). This system is composed of several security systems and applies the security policy to each security system to establish the security zone. But we have some problems in standard forms of platform interfaces, operating methods, log formats, and message structure required in integrating security systems. So, we propose a message exchange scheme among the security systems using XML. This approach presents significant advantages in terms of increasing simplicity and scalability, supporting ease of maintenance, and providing fast information retrieval.

1 Introduction

Today, the Internet is connected to almost everyone all around the world. Despite the usefulness of the Internet, the security issues are emerging from intrusions and attacks such as computer viruses, denial of services, and hackings to destroy information, which have been considered to be seriously threatening the Internet. And the number of intrusions and incidents is increasing every year[18]. So using firewalls, intrusion detection systems, virtual private networks, and vulnerability scanners in network, we protect the information systems from those threats. But the uses of various security systems require not only highly advanced management knowledge but also many administrators to handle the different user interfaces and functions. The most important problem is that it is very difficult to maintain the consistency of the security policies among different security systems. However, we can raise the security level, using the security events of some security systems. Also, it is necessary to integrate heterogeneous security systems to raise the security level by cooperating with other security systems.

Our Integrated Security Management System (ISMS) is suggested as a solution to these problems[5]. The ISMS integrates user interfaces and methods of parameter

setting, and it is very helpful in maintaining the consistency of security policy. And the ISMS applies a policy globally among the different security systems. But the ISMS has some problems in different interface protocols, variety of factors, and different platforms among the heterogeneous systems. Most of all, we must solve the problem of message exchange among the security systems. The message exchange scheme has an influence on performance of the system. So we suggest a message exchange method to be applied by using XML among the security systems. We use this message exchange method for setting configuration, distributing policy, alerting intrusion detection, and logging.

In this paper, we suggest an Internet Security Integrated System (ISIS) Architecture. The ISIS is one of the Internet Security Management Systems and uses XML-based message exchange scheme. We expect the ISIS to be capable of providing the fundamental methods of integration of different security systems.

This paper is organized as follows. In section 2, we show several security systems that use the XML and the problems of systems. In section 3, we explain the architecture and function of the ISIS. In section 4, we evaluate the performance of the ISIS whether the XML message exchange scheme is useful or not. Conclusions and future works are discussed in section 5.

2 Related Works

XML has been developed successively since W3C had announced XML 1.0 recommendation[1]. XML is applied in various fields of industry, but it is rare for security area or products to use XML. But these days XML is used in the security area because security systems need the characteristics of extendibility, flexibility and formality.

XML has the following characteristics : (i) Simplicity - XML makes it easier to provide metadata, or descriptive data about the information. XML is easily readable and understandable to both people and computer. Also, it is easily processed by computer and is easy to learn, so it can reduce the development time. (ii) Powerful - XML can represent not only complex data structure but also security sensitive data. (iii) Scalability - Even though we use a text-based format, it can be also supported from XML. (iv) Exchangeability - XML allows most computer systems to manipulate XML without major modifications of existing infrastructure. (v) Independency - XML has separated content from design. So XML allows the look of the document to change without changing the document.

Intrusion Detection Working Group (IDWG) of IETF has developed the Intrusion Detection Message Exchange Format (IDMEF) to define the data formats and exchange procedures for sharing the information of interest. The IDMEF describes a data model to represent the information exported by intrusion detection systems, and explains the rationale for using this model. Currently, IDWG has published four internet drafts - Intrusion Detection Message Exchange Requirements, Intrusion Detection Message Exchange Format Data Model, and Extensible Markup Language (XML) Document Type Definition, the TUNNEL Profile, and the Intrusion Detection Exchange Protocol (IDXP)[3]. Although IDMEF provides flexibility of alert data format,

it is insufficient to maintain the policies and the configurations of security systems. But XML provides more suitable solution than SMI in the interactions of systems [3].

DAXFi(Dynamic XML Firewall) is a Python package that helps to configure several different kinds of firewalls in a consistent way. Rules can be described with XML files, XML strings, or generated directly by the code. It comes with a Python package, useful in building other applications aimed at manipulating different firewalls in a uniform way. DAXFi has offered the integrating methods of firewall systems. But its extensibility was restricted to some firewalls. It supports iptables, ipfilter, ipchains and ipfwadm[19].

Snort is a lightweight network intrusion detection system, and it can perform real-time traffic analysis and packet logging on IP networks. Snort uses the flexible rules language to describe traffic that it should collect or pass, as well as a detection engine that utilizes a modular plug-in architecture. The XML plug-in enables snort to log in Simple Network Markup Language(SNML) to a file or over a network. We can use this plug-in with one or more snort sensors to log on to a central database and create highly configurable intrusion detection infrastructures within networks. But Snort is restricted with narrow uses of XML in alert or log.

There are many efforts to use the XML in security systems. But most of them use the XML in the same products at system log or alert message. Therefore, we integrate security systems using XML. And we show the efficiency of the ISIS, which uses the XML, comparing it to the conventional ISMS by simulations.

In the next section, we will suggest the architecture of the ISIS using XML.

3 Architecture of the ISIS

3.1 Overview of the ISIS

The ISIS is composed of several security systems and applies the security policy to each security system to establish the security zone. The ISIS is required to solve the problems in the following manner: Firstly, the more we hire a security system, the more we suffer from managing the security systems. The security systems have many configuration parameters, so management skill is required to manage them. Secondly, individual security policies cause problems of security policy conflict. The lack of consistency in security policies can bring serious problems. Thirdly, distributed management of each security system requires much cost such as manpower, time, and money. Therefore, we designed the ISIS to diminish the Total Cost of Ownership(TCO).

Figure 1 shows the conceptual architecture of the ISIS. The ISIS has integrated the security systems as agents to control accesses, inspect vulnerabilities, and detect intrusions. The ISIS consists of four elements – the ISIS Client, the ISIS Server, the ISIS Agent, and the ISIS database(ISISDB)[5,8]. The ISIS agents installed on each security system are distributed in large networks to collect the logs and runtime events, but also control the systems. The ISIS Server processes the request of the client and collects the information of logs, performances, configurations, and policies. And the collected

data are sent to the ISIS client to manage the ISIS. The ISISDB has stored the user information, system configurations, security policies, performance data, security events, and logs.

Fig. 1. A conceptual architecture of the ISIS.

3.2 Design and Implementation of the ISIS

Architecture of the ISIS follows a 3-tier model(Client, Server, Agent), with the specific packages designed for each tier. Figure 2 shows the main module of the ISIS. The functions of each module are as follows.

CMM(Configuration Management Module) : This module manages the configuration information of the security systems. CMM offers basic information to the others modules. We can obtain or change the configuration properties. Whenever the security system boots up, the agent sends the configuration information to the server. The server maintains configuration information in database and uses as basic information for other modules.

PMM(Policy Management Module) : PMM manages the security policies concerned the security systems of the ISIS. It operates the distribution, modification and deletion of security policies. It also checks the validity of the policies and provides the method of checking an authorization to perform the operations.

PeMM(Performance Management Module) : This module measures the performance of the ISIS. PeMM has the performance configurations of the security systems and the ISIS Server, and performs the response action when the performance gets over the threshold value.

UMM(User Management Module) : UMM manages the user authentication and authorization. UMM maintains Access Control List(ACL)s and checks the authorization of user using ACLs. According to the user access level, the operations are determined whether it is permitted or not.

SOMM(Security Object Management Module) : The network objects concerning the security policies in the ISIS are called the Security Object(SO)s. SOs are composed of security systems, routers, switches, and hosts. SOMM manages the objects, which are the subjects of the operation concerning with the configurations, the policies, and events.

EMM(Event Management Module) : EMM is in charge of alarm to client. It notifies the security events from security systems to the administrator and reports the event logs to compile a statistical report.

Fig. 2. The system Architecture of the ISIS

ErHM(Error Handling Module) : ErHM collects the system errors, and responds to errors, and reports their analysis. It can report the conflict of security policies, the violations of authentication or authorization, and system faults.

SSMM(Security Service Management Module) : SSMM maintains the available service lists in the ISIS. The administrator has to select the service names to use.

These modules are operated using the message exchange with the clients or the agents. And the ISIS must prepare the extension of messages caused by adding new security systems. The message structure has an influence on the scalability of the system. We have guaranteed the system scalability by using XML message exchange scheme. In the next section, we suggest a message exchange scheme in detail.

3.3 Message Definition of the ISIS

We had developed the ISIS MIB, and used it in past. The ISIS MIB had managed objects, policies, services, users, monitoring or performance information and et al. The ISIS MIB had been developed using the BER Encoding rule[20]. These information is similar to current message structure. When we had developed the ISIS, we have faced to unexpected difficulty. The difficulty is that we had problems about maintaining the different message formats of security systems entirely, and we took much time to develop the agent of security systems. So, we suggested the XML based integrated systems to offer the convenience of appending the additional security systems[3].

There are five types of message. Each type has a subclass corresponding to each security system; Firewall, IDS, VPN, Scanner, etc.

Currently, the data model representing the information must be flexible to accommodate different needs[10]. Figure 3 shows the relationship between the principal

Fig. 3. The ISIS Message Architecture.

components of the data model. The top-level class of all the ISIS messages is the ISISMessage; every message is a subclass of this top-level class. Five types of message are defined - Configuration, Policy, Event, Performance, and Security Log. Within the message, subclasses are used to provide the detailed information.

Fig. 4. A Part of XML Schema of interchange Message in the ISIS.

The Communication Module operates by using text-based scheme. Figure 4 shows a part of XML schema of interchange messages in the ISIS. The policy class contains the attributes generated by security systems. We define elements and attributes using element and attribute schema elements, and structures by using the simpleType and the complexType schema elements. The XML schema increases the reusability on the

basis of the object-oriented concepts and customized data types. We can omit the validation codes to delegate validation of data type to the parser applying the XML schema. But XML uses a text format and tags to delimit the data. Therefore, the XML format is larger than binary format[14].

Figure 5 shows an example of setting the security policy. The administrator commands the ISIS server to set the security policies. The ISIS server enables the parser to carry out the task of validating messages. If messages are valid, the PMM receives the next task of conflict checking. The PMM checks whether the policies are in conflict with the existing policies by using the conflict check algorithm [8]. Then, the server transmits the policies to the ISIS agent. The ISIS agent translates the received message to the format of each security system. Each security system may have its own format; the parameters, the tag names, the structures, etc. The XSLT handles the translation process. The XSLTs of the agents are important because they take charge of conversion to the understandable messages. Actually, the development of XSLT takes long time in adding an agent. The Agent:XDMA processes the remaining works by using the translation result. The ISIS agent returns the result of policy set. The Server stores the policies, or logs the error according to the result.

Fig. 5. Sequence Diagram about the Policy Set

4 Comparison of Performance

We have examined the usefulness of XML based scheme. In this simulation, we did not take the propagation latency in network into consideration. And we only estimated the message parsing time of the server and the agent because each security system uses its own particular message format after the message parsing.

The simulation was carried out using the policy set module to both the IDS and the firewall agent. The IDS and the firewall agent were installed in two machines, Heaven and Ixion. In each host, we installed the ISIS server with an agent to decrease the

network delay. Thus, we measured the message processing time – from one message generation to another message generation. We increased the policy count from 1 to 100 and performed 10 thousands times for each system. Figure 6 shows the result of our simulation. Figures 6 (a) and 6 (c) indicate the processing time in the past method using the BER encoding. As the count of policies increased, the processing time increased slowly and stably. Figure 6 (b) and (d) show the result using the XML message format. Compared with the previous system, the elapsed time has taken longer when the XML was used. Though the XML message scheme takes more time than the BER message scheme, the elapsed time was under 100ms. Simulation circumstance was as follows.

Oracle XML Developer's Kit for C 9.2.0.4.0

Heaven(IDS Agent) :

OS : Solaris7

CPU : UltraSparc II 450Mhz

Memory : 512MB

Ixion(Firewall Agent) :

OS : Solaris7

CPU : UltraSparc IIIi 440Mhz

Memory : 512MB

Fig. 6. The Parsing Result of Policy Set Message. (a) Processing Time of ASN.1(BER) Message for IDS Agent. (b) Processing Time of XML Message for IDS Agent. (c) Processing Time of ASN.1(BER) Message for Firewall Agent. (d) Processing Time of XML Message for Firewall Agent

Recently, security systems are being considered as real time systems. Therefore, it is very important to respond quickly to abnormal activities. To overcome the slow Process time of XML, many studies are in progress to develop an accelerator of the XML process. [21].

5 Conclusion and Future Works

Nowadays, security systems run independently. And this independence has brought the problems of difficulty in responding to increasing security threats. We have proposed the ISIS to solve the Internet threats using Integrated Security Management. The ISIS has improved the security intensity through collaborated works among the security systems. The ISIS Architecture has a 3-tier architecture, and the agents of the ISIS are customized to operate security tasks, taking the characteristics and the operation of each security system into consideration. The ISIS uses the XML-based message exchange as a method of collaboration among security systems. Using the XML, we increase the scalability and integrity of the ISIS with a minimum effort.

Comparison of the response time of the security system using BER encoding method and the ISIS using the XML method shows that response time of the ISIS is somewhat slower, but it can be acceptable. Through further study of the XML processing methods, we will continue to enhance the performance of the ISIS.

In the future work of this study, we are planning to optimize the ISIS for highly efficient throughput. We will study the optimization of XDMA, XSLT or DHM, ISISDB, and database query. A more quantitative approach will be needed to solve this issue.

References

1. W3C XML Working Group: Extensible Markup Language (XML) 1.0 (Second Edition), <http://www.w3.org/TR/REC-xml>, 2000. 10. 6
2. W3C XML Schema Working Group, XML Schema Part 0, Structures, <http://www.w3.org/TR/2001/REC-xmlschema-0-20010502>, 2001. 5. 2
3. IETF IDWG, The Intrusion Detection Exchange Protocol (IDXP), <http://www.ietf.org/internet-drafts/draft-ietf-idwg-beep-idxp-07.txt>, 2002. 10. 22
4. IETF IDWG, Intrusion Detection Message Exchange Format Data Model and Extensible Markup Language (XML) Document Type Definition, <http://www.ietf.org/internet-drafts/draft-ietf-idwg-idmef-xml-10.txt>, 2001. 1. 30
5. Dong, Y. L., Dong, S. K., Seung, S. H., Tai, M. C.: A Development of Web-based Integrated Security Management System for Firewalls, The Transactions of KIPS, Vol.7, KIPS, NO10, pp. 3171–3181, 2000
6. Samuel, C. M.: XML and Information Security, Information Security Management Handbook(4th), Vol.3, CRC Press LLC, pp.551–361, 2002
7. Xavier, E.: XML based security for e-commerce applications, Engineering of Computer Based Systems, pp. 10–17, 2001
8. Yoon J. K., Kwang H. K., Hyun H. C., Tai M. C.: Policy Conflict Probing for Integrated Security Management, APNOMS2001, pp. 377–387, 2001

9. Roy, J., Ramanujan, A.: XML schema language: taking XML to the next level, *IT Professional*, Volume: 3 Issue: 2, 2001
10. Widergren, S.; deVos, A.; Jun Z.: XML for data exchange, *Power Engineering Society Summer Meeting*, Vol. 2, pp.840–842, 1999
11. Daniel, E., Robert, L.: *Structured Information Retrieval using XML*, *ACM SIGIR2000 Workshop On XML and Information retrieval*, Athens, Greece, 2000
12. Takeshi, I., Blair, D., Ed, S.: *XML Encryption Syntax and Processing*, W3C, <http://www.w3.org/TR/xmlenc-core>, 2002. 12. 10
13. Mark, B., John, B., Barb, F., Brian, L., Ed, S.: *XML-Signature Syntax and Processing*, W3C, <http://www.w3.org/TR/xmlsig-core>, 2002. 2. 12
14. Takashi, I., Hiroshi, M.: *Mapping between ASN.1 and XML*, *Applications and the Internet Symposium*. 2001
15. Joseph, F., Francis, P., Chris, B.: *Converting Relational Database into XML Document*, *Database and Expert Systems Applications*, *Proceedings*, pp.61–65, 2001
16. Flesca, S., Greco, S., Zumpano, E.: *Modeling and querying XML-data*, *Database Engineering and Applications Symposium*, pp. 275–283, 2000
17. Jason, M., Jennifer, W.: *Query Optimization for XML*. *Proceedings of the Twenty-Fifth International Conference on Very Large Data Bases*, Edinburgh, Scotland, September 1999
18. CERT/CC Statistics 1988-2002, http://www.cert.org/stats/cert_stats.html
19. The Dynamic XML Firewall, <http://daxfi.sourceforge.net>,
20. Steedman, Douglas., *Abstract Syntax Notation One (ASN.1) – The Tutorial and Reference*, *Technology Appraisals*, 1990
21. Larry Seltzer, “XML on a chip?”, <http://techupdate.zdnet.com/techupdate>, 2002. 12. 17