December 2004

An Intrusion Detection and Response Cooperation Model Based on XML Message Exchange

Xiaomei Dong
Northeastern University

Guang Xiang
Northeastern University

Ge Yu
Northeastern University

An Intrusion Detection and Response Cooperation Model Based on XML Message Exchange

Xiaomei Dong
School of Information Science and Engineering, Northeastern University, Shenyang 110004, P.R.China
xmdong@mail.neu.edu.cn

Guang Xiang
School of Information Science and Engineering, Northeastern University, Shenyang 110004, P.R.China
mmysunshine@sina.com

Ge Yu
School of Information Science and Engineering, Northeastern University, Shenyang 110004, P.R.China
yuge@mail.neu.edu.cn

Abstract

In a distributed intrusion detection system, multiple detection components are usually applied to monitor different hosts and network resources. The components sometimes need to cooperate with one another to perform complex detection tasks. However, the heterogeneity of the detection components greatly increases the complexity of the cooperation among the components. Therefore, a more general and efficient intrusion detection cooperation mechanism is required. Considering XML's advantages in data representation and platform independence, we proposed in this paper a distributed intrusion detection and response cooperation model based on XML message exchange. In our model, cooperation agents correlate the detection results from the detection agents and cooperation agents of other domains to detect complex intrusions. To facilitate the communication between different components, the Intrusion Detection Message Exchange Format (IDMEF) is extended and applied to represent the messages exchanged among the intrusion detection components. In addition, cooperation agents cooperate with one another by exchanging XML messages. In the model, a schema is defined to constrain the XML documents. A new concept of suspect is proposed, which indicates the suspected degree of an activity. And all the suspected activities and intrusions detected are reported to the monitors for isolation and monitoring.

Keywords: Intrusion detection, Cooperation, XML, XSL, DOM

1. Introduction

With the rapid development of the Internet, the number of the security problems has been increasing daily. All kinds of intrusion events have posed a great threat to the security of computer and network systems. As a result, the research on the intrusion detection（ID） techniques and the protection of the system has become hot topics nowadays (Debar et al. 1999; Joo et al. 2003; Kim et al. 2002). However, since most early intrusion detection systems（IDS）are centralized while most current network resources are generally distributed and intrusion activities tend to be distributed cooperative ones, designing and building distributed intrusion detection systems are extremely necessary.

In a distributed intrusion detection system, many detecting components monitor different host and network resources. Sometimes these components need to cooperate with one another to complete relatively

[1] This work is funded by CIMS Subject of Chinese 863 Plan (2003AA414210), Chinese National Natural Science Funds (60173051), Chinese National Doctoral Program Fund (20030145029) and the Excellent Young Teacher Encouragement Program of Ministry of Education of China.

complex detection tasks (Spafford et al. 2000). However, since the components may employ different detection methods, run on different platforms and have different data formats, cooperation among them is very complicated. Hence, a universal and efficient intrusion detection cooperation mechanism is needed in a distributed intrusion detection system.

XML (Extensible Markup Language), a standard markup language created by the World Wide Web Consortium (W3C), has been an open standard of data representation (W3C 2003). With the appearance of XML, many relevant technological standards followed (W3C 2001; W3C 2003). Now XML has been applied extensively to the fields such as data and information management, data exchange, Web applications, electronic commerce and application integration.

Considering XML's advantage in data representation and platform-independence, we propose a distributed intrusion detection and response cooperation mechanism based on XML message exchange. In our model, cooperation agents correlate the detection results from the detection agents and cooperation agents of other domains to detect complex intrusions. XML documents are adopted to represent the messages exchanged among the intrusion detection components and cooperation is achieved by means of XML message exchange. In addition, the paper also includes the thinking of intrusion tolerance and comes up with the definition of *suspect*.

2. Related Works

As research on distributed intrusion detection systems goes on, people have come to realize the necessity of the generality of the intrusion detection systems and the cooperation between them and as a result have been researching the general architecture and cooperation mechanism of the intrusion detection system.

2.1 Common Intrusion Detection Framework（CIDF）

The architecture and thinking of the intrusion detection system have undergone great changes throughout these years, from the host-based intrusion detection at the beginning to the network-based intrusion detection later and then to the distributed intrusion detection recently. However, these intrusion detection systems, host-based or network-based ones, both include at least several basic components.

CIDF is a set of general languages, protocols and APIs developed by the CIDF working group under the direction of USA DARPA (Defense Advanced Research Projects Agency). It enables different intrusion detection components to interoperate and share information. CIDF divides an intrusion detection system into many fixed components, which, except for the database component that stores a large amount of data, consist of codes and configuration information and exchange data by means of Generalized Intrusion Detection Object（GIDO）. A hierarchical architecture including the GIDO layer, the message layer and negotiation and transmission layer makes the communication between these components possible. CIDF defines a common model of an intrusion detection system and divides it into four components:

(1) Event generators: CIDF regards all data to be analyzed by the intrusion detection system as events. In a network-based intrusion detection system, these data are packets; in a host-based intrusion detection system they are the information generated by system logs.

Event generators（"E-boxes"）obtain event information from outside the intrusion detection system and send them to other parts of the entire system in the format of GIDO.
(2) Event analyzers: Event analyzers（"A-boxes"）receive GIDO from other components, analyze it and then return a new GIDO.
(3) Response units: Response units（"R-boxes"）execute corresponding actions according to the GIDO received. They may cut off the connection, change the file properties and even beat back the attackers or just give a simple alarm.
(4) Event databases: Event databases（"D-boxes"）store GIDO and receive queries. Event databases, which store mid or final data, can be complex databases or simple text files.

GIDO is represented by a standard common format defined by Common Intrusion Specification Language (CISL). A GIDO consists of two parts: a fixed header and a body with variable length. The header includes the copyright information, the timestamp, the body length, etc. The header format of GIDO is shown in Figure 1.

Version ID (2 bytes)	Class ID (2 bytes)	Length (4 bytes)	TimeStamp (4 bytes)	Tread ID (4 bytes)	Source ID (16 bytes)	Flag (1 byte)

Fig. 1 Header format of GIDO

The logical format of GIDO S-expression is similar to the Lisp expression. The parsing tree of the expression is grouped by parentheses. The GIDO body contains data and semantic identifier SID. Due to the low efficiency of the expressions in ASCII format, CISL defines a new type of coding which uses byte sequences to represent the information in the GIDO to save space. This coding is widely used when GIDF components exchange GIDO.

The IDIAN project supported by DARPA adopted the CIDF architecture and developed a negotiation protocol to enable distributed heterogeneous intrusion detection components to interoperate and reach agreements on each component's information processing capabilities and needs (Feiertag et al. 2000). Ning P. proposed a request model framework among cooperative intrusion detection systems (Ning et al. 2000), which harnessed the formal approaches in CIDF and extended the CIDF components to include a query function. However, the CIDF working group did not continue its work.

2.2 IDWG's Intrusion Detection Information Exchange Standard Drafts

The intrusion detection working group（IDWG）of IETF proposed a series of intrusion detection information exchange standard drafts on the basis of CIDF including Intrusion Detection Message Exchange Format (IDMEF) (IETF 2003) and Intrusion Alert Protocol (IAP) (IETF 2001). Further more, it is considering constructing an Intrusion Detection Exchange Protocol (IDXP)(IETF 2002) using the BEEP（Blocks Extensible Exchange Protocol）to replace IAP to perform the information exchange among intrusion detection components. Information can be exchanged in text, binary or IDMEF format. In the IDMEF draft, data models are specified by UML, the messages exchanged in intrusion detection systems such as alarms are defined by XML and corresponding DTD is also defined to constrain the XML documents of these messages.

However, IDWG did not define the overall architecture of the distributed intrusion

detection system. In addition, since the corresponding standards of the XML Schema were not available when IDWG sketched these drafts, IDWG drafts use DTD other than Schema to constrain XML documents. IDWG is still revising these drafts and there are no formal standards of the architecture and cooperation mechanism of the distributed intrusion detection system up to now.

3. The Distributed Intrusion Detection and Response Cooperation Model Based on XML Message Exchange

In distributed intrusion detection systems, the components may run on different platforms, use different detection algorithms and have different data representation formats, all of which hinder the cooperation among them. Hence, it is necessary to construct a general and platform-independent data description language. XML is a good solution meeting these requirements. In addition, since there are many tools and technical standards available for the processing of XML documents, the generation and processing of messages will be greatly simplified. Considering all the facts above, we propose a distributed intrusion detection and response cooperation model based on XML message exchange.

3.1 Overall Architecture

To meet the requirements of a large, distributed and heterogeneous network environment, a distributed cooperative intrusion detection system should be composed of several manageable components that are distributed among different domains of the network. A domain is a relatively independent part of the network and different domains communicate with one another by TCP/IP.

The architecture of the distributed cooperative intrusion detection system is defined as a quintuple.

<Distributed Cooperative IDS> = (<Data_collector>, <Detection_agent>, <Monitor>, <Cooperation_agent>, <Intrusion_event_database>)

<Data_collector>：There can be one or more data collectors distributed among the network;

<Detection_agent>：There can be one or more detection agents using different detection algorithms distributed among the network;

<Monitor>：Only one monitor exists in each domain;

<Cooperation_agent>：Only one cooperation agent exists in each domain;

<Intrusion_event_database>：Only one intrusion event database exists in each domain;

Here, an agent can be defined as (Bradshaw 1997): "… a software entity which functions continuously and autonomously in a particular environment … able to carry out activities in a flexible and intelligent manner that is responsive to changes in the environment… Ideally, an agent that functions continuously … would be able to learn from its experience...."

Figure 2 is the UML description of the overall architecture of a distributed cooperative intrusion detection system.

3.2 The Work Flow of the Distributed IDS

In a distributed intrusion detection system, different components in the same domain cooperate with one another to complete the detection tasks. Different domains cooperate by message exchange. The functions of each component are like this:

(1) The data collectors collect data from system logs or network packets and then forward them corresponding detection agents.

(2) The detection agents analyze the data to capture intrusion or suspected activities in a host or network and then send the results after analysis to cooperation agents. The cooperation agents integrate the data received, store the results after integration into the intrusion event database or send them to cooperation agents of other domains.

(3) Intrusion event databases record detected intrusion or anomalous activities.

(4) Cooperation agents analyze and combine the reports from the detection agents and, if an intrusion is detected, store the analysis results in the intrusion detection database, automatically generate alert messages in the format of XML document and then send them to monitors for display and to cooperation agents on other hosts via the network. Taking into consideration the detection results from the detection agents of the same domain, cooperation agents can parse and analyze the alert messages to detect more complex attacks.

(5) Monitors watch each agent, control the system responses and report the system status to administrators.

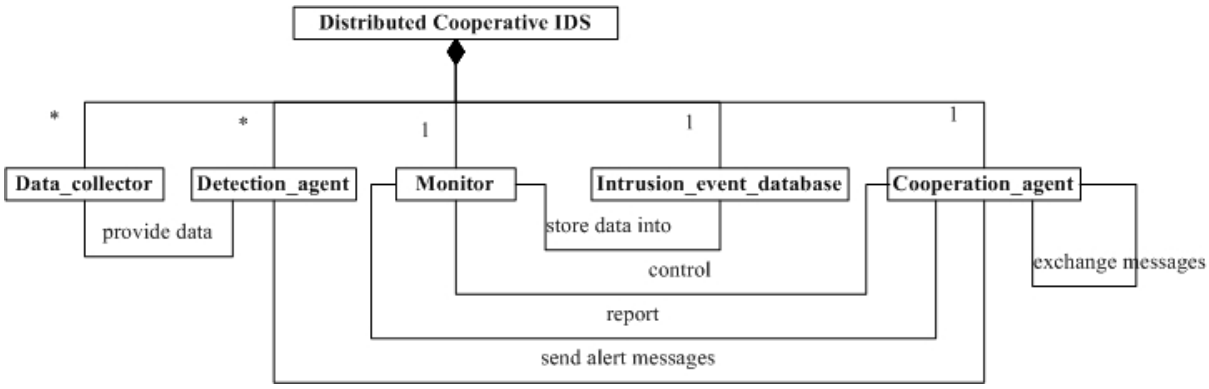

Fig. 2. Architecture of a distributed cooperative intrusion detection system

There can be one or more data collectors and detection agents, but only one cooperation agent, intrusion detection database and monitor in each domain.

The work flow of the system is described by Figure 3 as below.

4. The Cooperation Mechanism Based on XML Message Exchange

4.1 The Definition and Constraint of XML Messages

Nowadays, scholars generally believe intrusion detection technology alone is not enough. The accuracy of the intrusion detection system has always left much to be desired and new attack methods frequently emerge. Therefore, tolerating some minor attacks is very important to a system (Dutertre et al. 2001; Gong et al. 2000; Liu 2002; Liu et al. 2001; Luenam et al. 2002; Malkin et al. 1999; Pal et al. 2001; Pal et al. 2000; Wu et al. 1999). In our intrusion detection and response cooperation model, IDMEF is extended and a new concept named *suspect* is proposed. All detected suspected intrusion activities are reported to cooperation components to be isolated and monitored as early as possible.

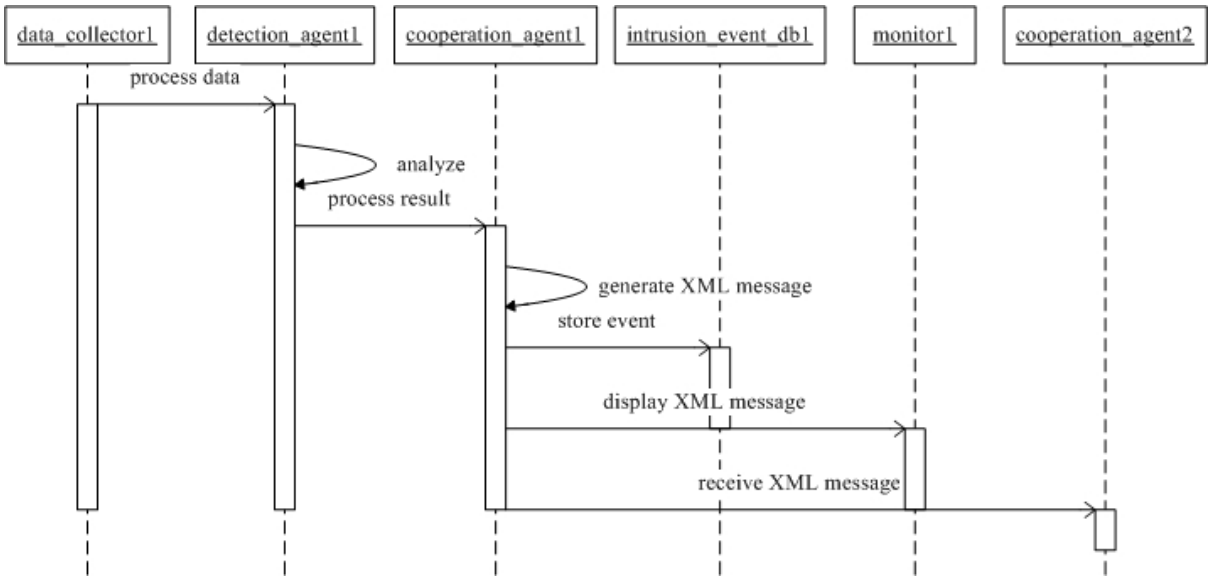

Fig. 3. The work flow of the system

We modify IDMEF and add a new property called *suspect* into class alert to indicate the suspected degree of an alert message. The value of this new property *suspect*, expressed in percentage, ranges from 0 to 100%. If an activity is identified as intrusion for sure, *suspect* is set to 100%. When detecting, detection agents can appropriately broaden the threshold of the activities regarded as anomaly and report the suspected but not definite intrusion activities to cooperation agents to detect potential intrusion activities as early as possible. The bigger the value of *suspect*, the higher the suspected degree of an activity and the more dangerous it is to the system.

To correctly create and process alert messages in XML format, we propose a Schema to define and constrain XML messages. Since this format is the extension to IDMEF, standard IDMEF can be constrained by our Schema and is able to interoperate with other IDS products that conform to IDMEF standard. According to the IDMEF drafts of IDWG, the XML message Alert consists of Analyzer, CreateTime, DetectTime, AnalyzerTime, Source, Target, Classification, AdditionalData, etc.

The XML Schema document of a message is like this:

```
<xs: Schema xmlns: xs = “http://www.w3.org/2001/XMLSchema”>
    … …
    <xs: element name = “Alert” type = “AlertType”/>
    <xs: ComplexType name = “AlertType”>
    <xs: sequence>
    <xs: element name = “Analyzer” type = “AnalyzerType”/>
    <xs: element name = “CreateTime” type = “CRTime”/>
    … …
</xs: Schema>
```

All messages transmitted by cooperation agents can be transformed into XML format for further transmission. For example, an alert message may be like below:

```
<IDMEF>
    <Alert id="123" impact="unknown" suspect="6">
```

```
        <Analyzer id="4461">
            <Node Id="568" Category="unknown">
              <Location>202.112.231.122</Location>
              <name>badguy.hacker.net</name>
              <Address Id="232" Category="vm" Vlan-Name="instruction"
Vlan-Num="4333">
                    <Address>http://www.neu.edu.cn</Address>
                    <netmask>202.118.1.255</netmask>
              </Address>
              <Address Id="2643" Category="e_mail" Vlan-Name="detect"
Vlan-Num="4939">
                    <Address>http://www.jlu.edu.cn</Address>
                    <netmask>202.112.3.23</netmask>
              </Address>
            </Node>
            <Process Process="1">
                <pid>1020</pid>
                <Path>H://ourtest/</Path>
            </Process>
        </Analyzer>
        ……
    </Alert>
</IDMEF>
```

4.2 Correlation Analysis of the XML Messages

When a detection agent detects an intrusion or an anomalous activity whose *suspect* exceeds the alert threshold, it sends an alert message to corresponding cooperation agents. Similarly, if the *suspect* of the detected anomalous activity is below the alert threshold but a detection agent regards it as a suspected activity, the detection agent sends a suspected activity message to cooperation agents. Otherwise, detection agents periodically send normal messages to cooperation agents. Cooperation agents receive detection results from detection agents of the same domain and other cooperation agents and then correlate and analyze these results. The process is like this:

(1) If an alert message with a *suspect* of 100% is received, an intrusion is detected and alert messages are sent to other monitors of the same domain and cooperation agents of other domains.

(2) Correlating and analyzing all the alert messages received with an alert correlation algorithm based on frequent closed pattern mining to find out the frequent close patterns whose *suspect* or support exceeds corresponding thresholds. Then alert messages are sent to other monitors of the same domain and the cooperation agents of other domains.

4.3 Message Exchange

When a cooperation agent detects an intrusion, it automatically generates a XML alert message, sends it to other monitors of the same domain and relevant cooperation agents of

other domains and at the same time stores the intrusion event into the intrusion event database. When a monitor receives a alert message, it displays the message in appropriate style on the screen and determines whether a response action should be taken according to the content and the danger level of the message. By analyzing the alert messages from other domains, cooperation agents can detect complex intrusions. The cooperation process based on XML message exchange can be described by Figure 4 below.

Due to the insecure transmission in TCP/IP network, messages should be encrypted beforehand. The IDWG drafts include IAP (IETF 2001) and IDXP (IETF 2002) both of which implement the transmission of XML messages. However, no formal standards and relevant products are available. A new distributed object computing protocol — Simple Object Access Protocol（SOAP）(World 2001) proposed by W3C defines some simple rules for service request and message format and transfers method parameters through XML documents. A SOAP message is a XML document. Its element is <Envelop> which includes <Header>、<Body> and <Fault>. We can utilize SOAP and transmit messages by putting encrypted XML alert messages in <Body>.

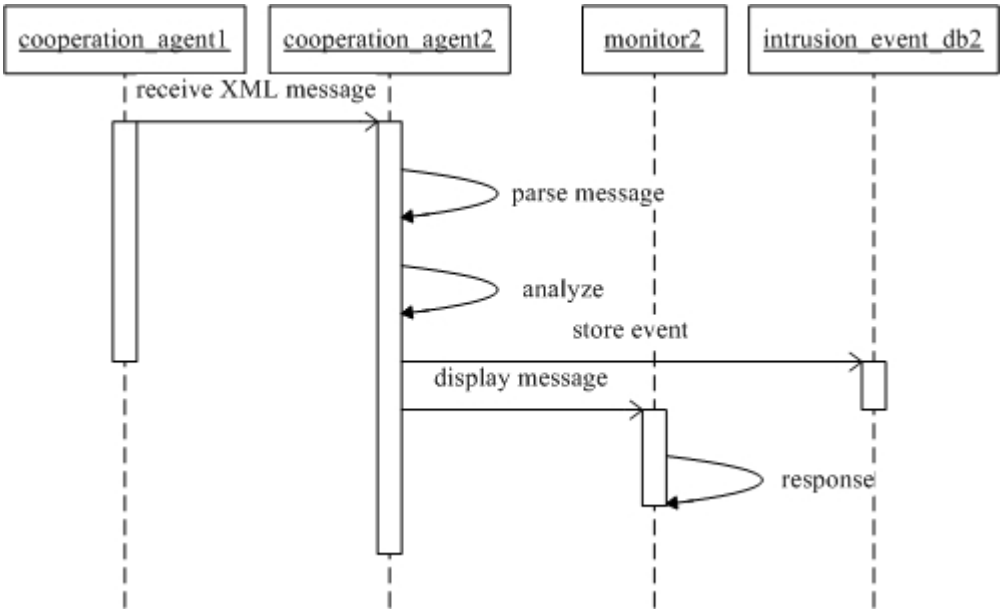

Fig.4 Cooperation process of XML-based message exchange

4.4 Data Exchange Between XML Messages and Database

Cooperation agents store XML alert messages that conform to standard IDMEF into the database and can also generate XML alert messages from the database. Most of the work here is the parsing and generation of XML documents. XML documents in standard IDMEF are parsed and stored in the database; on the other hand, data in the database are extracted to generate XML documents in IDMEF that will be transmitted in the network to facilitate the data exchange between different domains.

Since most mature business database systems nowadays are relational ones while special XML database systems are still under research and no practical products are available, we adopt relational database to store XML alert messages. Basically there are two ways to store XML data. The first one disassembles XML data according to its structure into different fields and then stores these fields; the second one stores the original XML document as it is

into the database. But because relational databases cannot handle very well large structural information and text data, the application of the latter method above is subject to more restriction. Naturally, it is possible to break up the structural text into portions as small as possible and then store them in the database as fields. However, this will bring more additional work in the retrieval and indexing of the database.

In the paper, the first method is adopted. XML data are disassembled into fields according to the IDMEF data model and then stored in the database.

The rules mapping the XML schema to relation schema are:

(1) Mapping the property of a XML element into a column in the corresponding table;

(2) Mapping those elements that have properties or sub-elements into a table;

(3) Mapping repeated elements (brothers) into a table and setting a foreign key referring to the key of their parents;

(4) Mapping those elements with no properties、sub-elements or repeated brothers into a column;

(5) Mapping those elements with multi-parents into a table and setting a foreign key for each parent referring to their keys;

We construct table structures as indicated by Table 1 below according to the IDMEF data models and mapping rules.

Table 1 Table structures of IDMEF message database

Table name	Properties
IDMEF_Message	Id_pk, version
Alert	id_pk, IDMEF_Message_fk, Analyzer_fk, CreateTime_fk, AnalyzerTime_fk, ident,impact
Heartbeat	id_pk, IDMEF_Message_fk, Analyzer_fk, CreateTime_fk, AnalyzerTime_fk, ident
Analyzer	Id_pk, Node_fk, Process_fk, analyzerid
CreateTime	Id_pk, pcdata, ntpstamp
DetectTime	Id_pk, Alert_pk, pcdata, ntpstamp
AnalyzerTime	Id_pk, pcdata, ntpstamp
Source	id_pk, Alert_fk, Node_fk, User_fk, Process_fk, Service_fk, ident, spoofed, interfaces
Target	id_pk, Alert_fk, Node_fk, User_fk, Process_fk, Service_fk, ident, decoy, interfaces
Classification	Id_pk, Alert_fk, name, url, origin
ToolAlert	Id_pk, Alert_fk, name, command
OverflowAlert	Id_pk, Alert_fk, program, size, buffer
CorrelationAlert	Id_pk, Alert_fk, name
alertident	Id_pk, ToolAlert_fk, CorrelationAlert_fk, pcdata, analyzerid
AdditionData	Id_pk, Alert_fk, Heartbeat_fk, pcdata, type, meaning
Node	Id_pk, location, name, ident, category

Address	id_pk, Node_fk, address, netmask, ident, category, vlan_name, vlan_num
Process	Id_pk, name, pid, path, ident
env	Id_pk, Process_fk, pcdata
Service	Id_pk, name, port, portlist, protocol, ident
SNMPService	Id_pk, Service_fk, oid, community, command
WebService	Id_pk, Service_fk, url, cgi, method
arg	Id_pk, Process_fk, SNMPServie_fk, pcdata
User	Id_pk, ident,category
UserId	Id_pk, User_fk, name, number, ident, type

In each table, id_pk represents the key, **_fk represents the foreign key, ** represents the table referred. For example, in table Alert, column IDMEF_Message_fk refers to the key of table IDMEF_Message.

There are three relations in the IDMEF data model:

(1) 1-to-many: For example in class Alert and Source, there are many repeated Source elements in Alert. To map this kind of relation into the database, we add a foreign key Alert_fk in table Source referring to the key of table Alert.

(2) Many-to-1: Such as class Source, Target and Node. Both Source and Target include Node elements, i.e. Node has two parents — Source and Target. To address this problem, we add a foreign key Node_fk in table Source and Target referring to the key id_pk of table Node.

(3) Many-to-many: Appropriate examples are class ToolAlert, Alertident and CorrelationAlert. Both ToolAlert and CorrelationAlert include many Alertident elements. The solution in this case is to add foreign keys ToolAlert_fk and CorrelationAlert_fk in table Alertident referring to the key id_pk of table ToolAlert and the key id_pk of table Correlation respectively.

The table structures of other classes can be built similarly.

Due to the high complexity of the document modes, DOM (document object mode) is adopted to parse XML documents. In implementation, we use class DOM in Microsoft's .NET framework to access XML documents. We use the LoadXml method in the instance of XmlDocument or an override method of Load method to open an existed document. After being analyzed by a XmlDocument object, the list ChildNodes (a set of XmlNode objects describing the contents of the XML documents) of the XML document will be filled according to the contents of the document and can be used for navigation when necessary. Recursive methods are used to traverse this XmlDocument. Each node in the document correlates with a set of sub-nodes, namely set ChildNodes, and includes a pointer to its parent node saved in the property of ParentNode. The object type of each node determines the type of its content. The content can then be extracted from the document.

When each node is being parsed, method GetAttribute of class XmlNode is used to obtain the value of the property, property HasChildNodes of class XmlNode is used to determine whether there are more sub-nodes and property InnerText of class XmlNode is

used to obtain the text value of the node. After parsing, the content of the document can be stored in the database.

The generation of the XML document is quite contrary to parsing in that the bottom-up recursive method is used in parsing while an opposite method is employed in generating. The root element IDMEF-Message is generated first and then each of its sub-elements. Other elements such as Alert and Heartbeat are generated similarly.

5. Conclusion

Because distributed cooperative intrusions will be the trend of intrusions in the future, it is very significant to study the distributed cooperative intrusion detection and the response cooperation mechanism. In the distributed cooperative intrusion detection mechanism based on XML message exchange proposed in the paper, XML documents are adopted to exchange information via cooperation agents among different domains. A corresponding XML Schema is designed for this purpose. Existing XML techniques can be applied to generate, parse, present and transmit the XML documents. With the cooperation between the intrusion detection components, more complex cooperative attacks can be detected and the system can be better defended.

References

Bradshaw J. M. (ed.). *Software Agents*, AAAI Press/The MIT Press, 1997.

Debar H., Dacier M., and Wespi A. "Towards a taxonomy of intrusion-detection systems", Computer Networks (31:8), 1999, pp. 805-822

Dutertre B., Saïdi H., and Stavridou V. "Intrusion-Tolerant Group Management in Enclaves", International Conference on Dependable Systems and Networks (DSN'01), 2001, pp. 203–212.

Feiertag R., Rho S., Benzinger L., et al. "Intrusion detection inter-component adaptive negotiation", Computer Networks(34:4), 2000, pp. 605-621

Gong F., Goseva-Popstojanova K., Wang F., et al. "Characterizing Intrusion Tolerant Systems Using A State Transition Model", http://www.anr.mcnc.org/projects/SITAR/papers/darpa00.pdf. 2000

IETF. "IAP: Intrusion Alert Protocol", http://www.ietf.org/internet-drafts/draft-ietf-idwg-iap-04.txt, 2001-8-20

IETF. "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, 2003-1-10

IETF. "The Intrusion Detection Exchange Protocol (IDXP)", http://www.ietf.org/internet-drafts/draft-ietf-idwg-beep-idxp-07.txt, 2002-10-23

Joo D., Hong T., and Han I. "The neural network models for IDS based on the asymmetric costs of false negative errors and false positive errors", Expert Systems with Applications(25:1), 2003, pp. 69-75

Kim J., and Bentley P. J. "Towards an Artificial Immune System for Network Intrusion Detection: An Investigation of Dynamic Clonal Selection", the Congress on Evolutionary Computation (CEC-2002), Honolulu, 2002, pp.1015-1020

Liu P. "Architectures for Intrusion Tolerant Database Systems", http://ist.psu.edu/s2/paper/itdb-archs.pdf, 2002

Liu P., and Jajodia S. "Multi-Phase Damage Confinement in Database Systems for Intrusion Tolerance", Proceedings of 14th IEEE Computer Security Foundations Workshop, 2001, pp.191-205

Luenam P., and Liu P. “The Design of an Adaptive Intrusion Tolerant Database System”, http://ist.psu.edu/s2/paper/AITDB-dsn02.pdf, 2002

Malkin M, Wu T, Boneh D. Experimenting with Shared Generation of RSA keys. Proceedings of the Internet Society's 1999 Symposium on Network and Distributed System Security (SNDSS), 1999, pp. 43-56.

Ning P., Wang X., and Jajodia S. “Modeling requests among cooperating intrusion detection systems”, Computer Communications(23:17), 2000, pp.1702-1715

Pal P., Webber F., Schantz R. E., et al. “Survival by Defense-Enabling”, Proceedings of the New Security Paradigms Workshop (NSPW 2001), New York: ACM Press, 2001, pp. 71-78

Pal P., Webber F., Schantz R.E., et al. “Intrusion Tolerant Systems”, Proceedings of the IEEE Information Survivability Workshop (ISW-2000), 2000, pp. 24-26.

Spafford E. H., and Zamboni D. “Intrusion detection using autonomous agents”, Computer Networks(34:4), 2000, pp. 547-570

W3C. “XML Schema Part 2: Datatypes”, http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/, 2001

W3C. “Document Object Model (DOM) Level 3 Core Specification”, http://www.w3.org/TR/2003/CR-DOM-Level-3-Core-20031107, 2003

W3C. “Extensible Markup Language (XML)”, http://www.w3.org/XML/，2003-8-20

W3C. “The Extensible Stylesheet Language Family (XSL)”, http://www.w3.org/Style/XSL, 2003-10-9

W3C. “XML Schema Part 0: Primer”, http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/ ,2001

W3C. “XML Schema Part 1: Structures”, http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/, 2001

World Wide Web Consortium. “SOAP 1.1”, http://www.w3.org/TR/#Notes. 2001.1

Wu T., Malkin M., and Boneh. D. “Building Intrusion Tolerant Applications”, Proceedings of the 8th USENIX Security Symposium, Washington, D.C., USENIX Association, 1999, pp. 79-91