Submit to Workshop on Cryptology and Information Security

Two Layer Network Intrusion Detection System

Yao-Tsung Lin Shian-Shyong Tseng, and T. T. Kuo*

Department of Computer and Information Science
National Chiao Tung University
Hsinchu 300, Taiwan, R.O.C.

Tel: +886-3-5712121 ext. 56658
Fax: +886-3-5721490
E-mail: gis88801@cis.nctu.edu.tw

Abstract

Due to the rapid growth of networked computer resources and the increasing importance of related applications, intrusions which threaten the infrastructure of these applications become critical problems today. In recent years, intrusion detection systems (IDS) are proposed to find and prevent the intrusion behaviors on network. However, as the intrusions become more and more complicated, general intrusion detection system based on monitoring network connections seems not useful for all kinds of intrusions, and Distributed IDSs are proposed to detect intrusions by cooperating different kinds of intrusion detection mechanism. In this work, a Two Layer Network Intrusion Detection System, which combines a general intrusion detection system and an inference engine to detect network intrusions, is designed. With these two mechanisms, both general network packet based intrusions and complicated intrusions can be detected. A prototype of this system together with the detection models about several complicated intrusions are designed and implemented to show the detection capability of the prototype.

Keywords: Intrusion Detection, Distributed IDS, Rule Base, Network Security

1. Introduction

The intrusion behavior today is more and more complicated [2], and the traditional network security mechanism seems not adaptive to current intrusions due to following reasons:

1. The variety of behavior models of intrusions: Traditional network intrusions are usually based on one to one network connection model, which means the intruder and victim host can be easily identified by monitoring each connection. However, due to the growth of computer computation power and network usage, back door programs and agent softwares make it more possible for intruder to attack hosts by the cooperation of several different hosts [2][3][4][8][9][23]. For example, typical DDoS intrusions are used to crash the service of victim host by launching lots of service requests from many cooperative hosts. Also, by taking advantage of vulnerability of network protocols, the intruders are even more hard to identify since they may faking themselves by modifying the information about network connections [4][8][9][23]. On the other hand, lots of victims may be affected in a single intrusion activity by taking advantage of the higher network bandwidth and computer resources. Intrusions are changing day by day.

2. The complexity of intrusions: As more and more different network applications are developed, many different kind of intrusions are also used to attack system for different purposes [2][3]. Many of these intrusions may use very complicated approaches to achieve and may be not easy to be detected from few network behavior features. Most of traditional intrusions are taking advantage of system vulnerabilities, which can be fixed by patches or newer versions of the software. However, the intrusions today attack the systems from the application or society aspect of view; for example, lots of E-commerce systems are intruded not due to the bugs of program or the vulnerabilities of system, but intruded with some social

approaches or cheats. Mechanisms to help monitoring users' behaviors will be needed for network security systems today.

3. The lack of features of intrusions: In order to detect intrusions between normal behaviors, features of behavior must be extracted from network usage. For many of different network applications, there exist lots of different features to be extracted from corresponding behaviors using different approaches. To extract complicated network behavior features becomes a great challenge to detect network intrusions.

According to the issues mentioned above, a more advanced Intrusion Detection System model is required. Researches and products [5][6][13][14][16][17][22][25][29][36] are proposed to solve some of these issues, including Distributed IDS model. In Distributed IDS model, different machines are coordinated to detect complex intrusion behavior. The issues of lower performance when detecting complex intrusion behavior can be solved in this model, and more complicated detection process can be done. However, most of these proposed systems use specific detecting model or connection model to cooperate; i.e., means these systems may be not adaptively helpful to detect unknown intrusions.

In this work, a DIDS model is proposed to solve these problems by cooperating a high performance lightweight IDS [17] and a rule base inference mechanism [34]. For none complicated intrusion behaviors, Online Network Analyzer/Detector (ONAD) of the proposed model is used for signature based intrusion detection process, which is widely used for online intrusion detection. On the other hand, in order to detect more complicated intrusion behaviors, the alarm events and abstract network information generated from ONAD are further analyzed by Meta Detection Engine (MDE) using Rule Base technology, and logical expressions are used to express intrusions with complicated behavior pattern. Based on the

concept, a Two Layer Network Intrusion Detection System, including Fundamental Network Connection Layer and Customized Application Layer, is proposed and implemented. In the fist Layer, ONAD is responsible for real-time detect intrusions in huge amount of network connection. And in the second Layer, MDE receives and analyzes events reported from ONAD and other applications to discover possible complicated intrusions using Rule Base inference technology. With these two layers, not only ordinary intrusions can be detected, but also the intrusions with complicated and varied behavior patterns can be identified.

2. Related Work

In this section, the existing intrusion categories and several previous researches about intrusion detection system will be firstly introduced. Several issues about the design of intrusion detection system will be next examined. Since the expression of intrusion pattern is very important for an intrusion detection system, the expressions of intrusion pattern in current intrusion detection systems are also summarized in this section. Finally, some advanced Distributed Intrusion Detection systems are also introduced.

2.1 Intrusion Detection System

To prevent network environment from intrusions, lots of products, e.g., firewall products [7][12][20][32], can be purchased on the market. Although the different systems may provide different functions and mechanisms for intrusion detection, the main purpose of them is to detect, filter, or prevent intrusions properly.

In designing an intrusion detection system, several issues, including the representation of intrusion patterns, the tradeoff between complexity of detection process and system resources

required, and the maintenance of expert knowledge, must be considered.

In traditional firewall systems [1][7][12][20][32], each intrusion pattern can be merely represented in single and simple rules, the system administrators should set rules about what kind of packet information should be filtered or noticed, and the system will match the information of each single packet with these rules. Although the dramatical improvement of hardware system improves the processing ability of these firewall systems, it is still very hard to deal with the increasingly huge amount of rules.

Besides, some researches about intrusion detection system are focusing on the design of efficient and practical representations of intrusion patterns to represent complex situations. Some specific data structures including rules and Goal Tree [13] are used in these researches, which may be robust enough to represent more advanced knowledge about intrusions but still face the problems of knowledge maintenance. Also, the performance of these systems may not meet the on-line performance requirements of an intrusion detection system, and the execution of intrusion detection using these mechanisms can not be efficiently evaluated.

2.2 Distributed Intrusion Detection System

Since the scale of intrusion behavior increases, the range to be protected for an IDS also becomes larger and larger. For an enterprise or organization which has departments distributed in different areas, a distributed management mechanism to monitor and manage network behaviors in different locations at the same time is needed for these organizations to protect whole organization. Many commercial IDS systems [5][10][13][[22][36] or network security systems have been proposed with similar architecture, including NetRanger [6] proposed by Cisco, EMERALD [22][29] proposed by SRI International, NetStat [15][25][36]

proposed by UCSB, and these systems will be introduced as follows.

NetRanger

NetRanger [6] is a network based intrusion detection system developed by Cisco [6], which is designed for large scale network environment for an enterprise. There are three major components in NetRanger, including Sensor, Director, and a communication system named Post-Office. Each Sensor is a Cisco hardware device, which is used to monitor network information by analyzing network packets and extracting the information contained in the packet header and content. Sensor also has the ability to relate packet information with the similar features, and detect possible intrusions. Director is used to collect the information sent from Sensors and analyze the information at remote site, it provides a centralized management mechanism for NetRanger.

EMERALD

EMERALD [22][29] is a distributed intrusion detection system developed by SRI [29]. Both pattern based and signature based intrusion detection model are used in EMERALD, and normal behavior of users will also be modeled. There are three tiers in EMERALD, including Service monitor, Domain monitor, and Enterprise monitor. Service monitor is used to monitor an independent system or service in a network domain. Domain monitor integrates the information from service monitors, and provides some security analysis about a network domain. Enterprise monitor provides a larger scale analysis to network activities, all the information of different network domains is considered to evaluate and detect some large-scale intrusions which may cross different network domain.

NetStat

NetSTAT [15][25][36] was developed by University California Santa Barbara (UCSB), which is the newest research result in a series of “STAT” [15][25] intrusion detection researches. From early 1990, “STAT” project tried to develop intrusion detection system based on State transition analysis to provide real time intrusion detection ability. NetSTAT consists of a set of Probes, which is used to detect and evaluate intrusion behavior in each network domain. Different probes may be used to detect different kind of behavior. Once a suspected behavior is detected, the information will be transferred to other probes which may use the information for further detection. In order to manage and configure the probes, Analyzer is designed. Analyzer is an independent tool to generate and manage the probes, and the configuration for each probe will be generated. And then Probes will detect network behaviors according to the configurations generated, including the setting for filter, state transition information, and a decision table for probe to follow.

However, network intrusion behaviors today become more and more complicated, traditional signature or pattern based intrusion detection is not adaptively useful for detecting new intrusions. More complete abilities for intrusion detection engine to model intrusion behaviors are needed. Expert needs to express their knowledge about intrusion behaviors in more complicated model, and refine the knowledge easily. Based on these requirements, a Two Layer Network Intrusion Detection System is proposed. In Two Layer Network Intrusion Detection System, intrusions are detected with two different mechanisms, including state-transition based intrusion detection and rule base expert system analysis. Intrusions not only can be modeled as traditional state-transition behavior, but also can be modeled as logic expressions. Both performance and modeling ability are considered in the system to construct a more complete network security system.

3. The architecture of Two Layer Network Intrusion Detection System

According to the issues mentioned above, a Two Layer Network Intrusion Detection System is proposed, which combines the high efficiency for network activity monitoring of general IDS and the accuracy for knowledge expression of rule base system. In this system, network behaviors will be monitored and detected in different levels, including fundamental network connection layer and customized application layer. The following figure shows the architecture of Two Layer Network Intrusion Detection System.

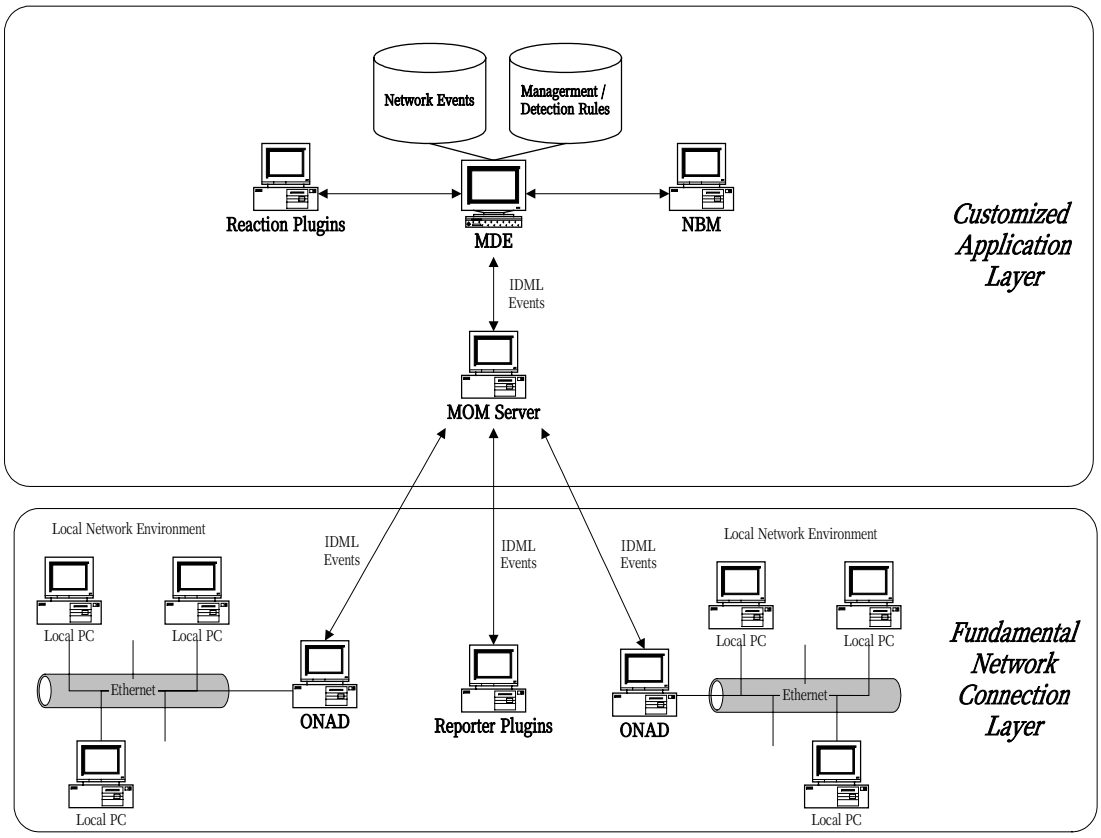

Figure 1: The architecture of Two Layer Intrusion Detection System

The fundamental network connection layer consists of two main components: Online Network Analyzer/Detector (ONAD) and Reporter Plugins. The ONAD is responsible for

detecting intrusions in network packet level at each local network area, and sending collected network information to Meta Detection Engine for detecting complicated network intrusions. The Reporter Plugins are other components that can send information or events to the Meta Detection Engine, where the messages sent by ONAD and Reporter Plugins are represented by the IDML event format [17][18].

The customized application layer consists of four components: Meta Detection Engine (MDE), Message-Oriented Middleware (MOM), Network Behavior Miner (NBM), and Reaction Plugins. Meta Detection Engine, which is one of important components in this system, receives the information from ONADs, and detects higher level intrusions according to these events. The rule inference engine of the MDE is used to detect intrusions based upon the rules given by experts. In other words, experts can represent their expertise about intrusions in complete logical expression, which makes the system more flexible to detect many kinds of intrusions. The Message-Oriented Middleware, a message system using multicasting and message queuing technologies, provides an efficient communication mechanism for bridging ONAD and MDE. For unknown intrusion patterns, Network Behavior Miner (NBM) is responsible for offline discovering these patterns from network behaviors reported from ONADs. NBM uses data mining technologies to find possible network behavior patterns, and discovered patterns can then be feedbacked to ONAD or MDE for further network intrusion detection and management. The Reaction Plugins are other reaction components, including firewall, Short-Message Service (SMS), e-mail service, and any other systems. In the following sections, the six components in two layers of the Two Layer Network Intrusion Detection System will be detailedly introduced.

3.2 Online Network Analyzer/Detector (ONAD)

ONAD in Two Layer Intrusion Detection System is used as the first layer security in a local network area, and reports the network events collected. The architecture of ONAD can be shown in following figure:

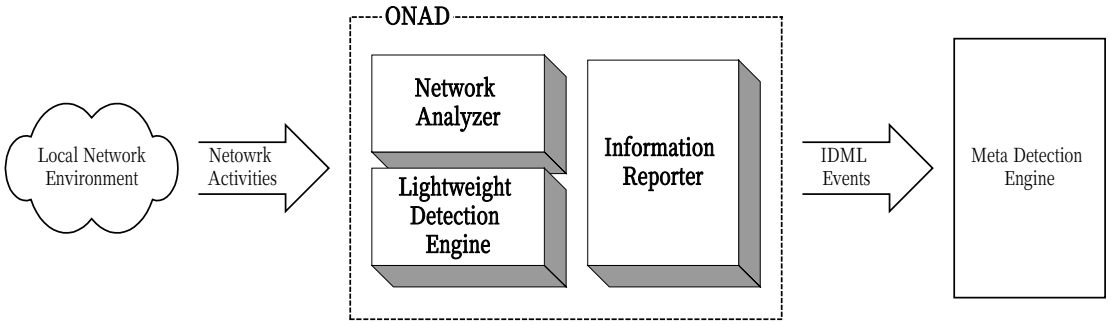

Figure 2: The architecture of ONAD

ONAD consists of three components, including Network Analyzer, Lightweight Detection Engine, and Information Reporter. The Lightweight Detection Engine in ONAD is used for detecting network intrusions similar to traditional IDS. Basic intrusions that can be identified by contents or flags in a single packet can be detected by this component. In the design of LDE, the IDML Based Intrusion Detection Model proposed in [17] is used. In the IDML Based Intrusion Detection Engine, an Intrusion Detection Markup Language (IDML) is designed to represent intrusion patterns. On the other hand, the detection engine is designed via state transition, which means intrusion patterns represented in IDML will be transformed as state transition diagram, and corresponding intrusion detection process can then be executed efficiently.

When LDE detects intrusions from basic network information, some network properties of the current state of the environment including general network traffic, general network behaves, and specific network behaves of network users, can be also analyzed and reported to MDE to monitor and manage the network areas of the system. The Network Analyzer of ONAD is

responsible for collecting and reporting network information, and statistical information about each target host or destination host including the traffic, number of packets, number of TCP packets, etc, will be collected.

Information Reporter of ONAD is responsible for reporting events to MDE in the format of MDE Events, which is used by MDE inference engine to detect intrusions and monitor network environment. The format of MDE Events follows the Event format defined in IDML [17], and information about corresponding event will be included. Both LDE and Network Analyzer will use this component to report intrusion event and network event.

3.3 Reporter Plugins

Since IDML format is used to represent events, the event types which can be detected by this system will not be limited to those reported from ONAD. Any event reported in IDML format will be able to be received and detected by MDE. The IDML parser and Fact Manager in MDE system are used to extract information from IDML event, and the inference engine of MDE can then retrieve facts to inference.

3.4 Meta Detection Engine

On each secured network area of this system, ONAD is responsible for detecting local events and intrusions, and necessary events will then be sent to MDE. MDE detects possible intrusions from the received events. The architecture of MDE is shown in following figure:

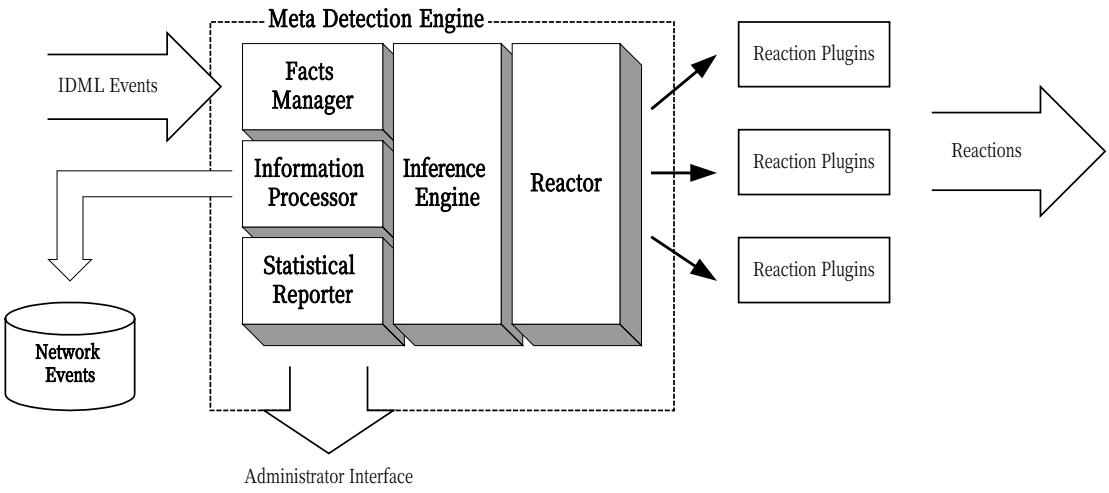

Figure 3: The architecture of Meta Detection Engine

MDE has five major components, including Facts Manager, Information Processor, Statistical Reporter, Inference Engine, and Reactor. Facts Manager is responsible for collecting events from ONAD and Network Analyzer, and the information extracted from Facts Manager will also be processed and stored by Information Processor for further mining and analyzing. Statistical Reporter visualizes statistical information for manager of the system to understand basic online information about the environment. The Inference Engine of MDE detects and manages the operation environment according to the intrusion detection rules or management rules defined by experts. Actors are designed for performing reaction according to the intrusion or management events detected

3.4.1 Facts Manager

Facts Manager of MDE is responsible for parsing the IDML data and managing extracted information. When Facts Manager receives connection from ONAD, the Receiver first receives and extracts the data. The IDML Parser [17] is used to extract the data.. For all the information extracted by IDML Parser, Data Management will store the information in Facts

Pool, and index the facts for efficiently retrieving necessary information. A timestamp is given to every fact in the Fact pool, and Facts Pool will clean up the expired facts periodically.

3.4.2 Information Processor

When the information reported from ONAD is extracted by Facts Manager, the information must be concurrently saved for network behavior analyzing and mining. Information Processor of MDE will save the information to database according to the information content type. The information stored will be used in Network Behavior Miner (NBM) to analyze network behaviors and discover possible network behavior patterns.

3.4.3 Statistical Reporter

According to the information extracted and stored by Facts Manager and Information Processor, Statistical Reporter will provide online statistical analysis about the network activities in network areas. Information of entire system can be provided by this component, and graphical interface is used for system manager to easily retrieve the information and help understanding network behaviors efficiently.

3.4.4 Inference Engine

In order to support intrusion detection and management monitoring, an Inference Engine is used in MDE. According to the intrusion detection rules and the management rules given by system manager or experts, the inference engine will use the facts from Fact Manager to trigger the rule inference and detect possible intrusions or management events. Since a rule

based inference engine is used, more complicated rule chains are supported for experts to represent more complicated intrusion behaviors. Since logical expressions can be used to represent intrusion patterns or behavior patterns in MDE, more complicated intrusions and behaviors can be detected by MDE.

The OORBMS (Object-Oriented Rule Based Management System) inference engine proposed in [34] is used as the Inference Engine of MDE. OORBMS, an object oriented rule based inference engine, can efficiently support forward chaining inference with Object-based inference mechanism. Since OORBMS supports data driven inference process and fact encapsulation, MDE can efficiently manage the inference process and provide the IDML formatted Facts to the inference engine.

For intrusions detection system as MDE, performance is one of the important properties to be concerned. OORBMS is designed to have good performance in rule inference, which means MDE will be able to infer the rules and detect intrusions efficiently. Some modifications are made on OORBMS inference engine, including Facts retrieving and managing mechanism. On the other hand, instead of managing the facts using original fact management mechanism, Facts Manager in this system is used to index and retrieve necessary facts for the modified OORBMS inference engine.

When an intrusion is detected or some specific rules is triggered, corresponding reactions should be taken. The reactions supported in original OORBMS include only fact assignment and rule class triggering. In order to use the inference engine for intrusion detection and network management in MDE, some reaction mechanisms are extended, such as Log, Alert, and Reactor triggering, to customize reaction process.

3.4.5 Reactor

The component is used for MDE to react when some management or intrusion detection condition triggered. The modified OORBMS inference engine will trigger Reactor to process specified action when some rules are triggered, and the parameters needed by the Reactor will be also passed by the inference engine. The types of processes can be done by Reactor including system log, mail alert, and all the customized Reactions set by manager.

3.5 Message-Oriented Middleware

Message-Oriented Middleware (MOM) is a client/server infrastructure, which allows communication among distributed applications on heterogeneous platforms [Rao95]. MOM allows applications to send message to each other efficiently and flexibly. It is situable for event-driven applications, such as intrusion detection systems. MOM is widely used in asynchronized information transferring, which means information sender and receiver will not directly connect and handshake with each other, but ask local agent to send the information to specific destination with lower connection delay. Also, MOM may use some technologies to reduce the bandwidth requirement, e.g., IP multicasting.

3.6 Network Behavior Miner

In order to make our system adaptively useful to detect modified or new intrusion behaviors, Network Behavior Miner is designed to discover possible behavior patterns from network information. Network Behavior Miner (NBM) is designed by Data Mining technology to find patterns included in the target data. The patterns found by NBM can be feedbacked to components for network behavior detection.

In order to collect the information for NBM, the Fact Manager in MDE stores the events received according to the type of the event. And NBM will be launched to discover possible intrusions or behavior patterns from collected data. At the beginning of the mining process, NBM will retrieve necessary information including data type and some abstraction information about the data, and the information gained will be used as fundamental information for mining. Then mining process [18] will be applied to the information and find user behavior pattern, where some outlier behaviors found may be treated as intrusions.

With NBM, not only known intrusions can be modeled and detected by the proposed system, but also some frequent patterns can be found and provided to system manager to enhance the detection ability of the system by filtering possible threats or intrusions from the users.

3.7 Reaction Plugins

For different management and detection events, different reactions should be done. In the MDE system, customized reactions are implemented as plug-ins of this system, and applications can be integrated using this feature. The concept of abstraction class inheritance in Object Oriented Programming is used to design such a mechanism. By implementing plug-ins according to the definition of abstract class, the system kernel does not need to be modified to provide new reaction functionality.

4. Implementation

The prototype of Two Layer Network Intrusion Detection System is implemented to verify the usability of this mode. Experiments about the prototype are introduced in this section.

4.1 Prototype Architecture

According to the framework proposed, a prototype of Two Layer Network Intrusion Detection System is implemented. In the prototype, the Meta Detection Engine is implemented as a centralized server for receiving events sent from client ONAD agents. The ONAD is an enhanced version of IDML detection engine [17], to provide basic detection capability of detecting possible intrusions in a local area network based on the packet level information extracted from network data. When ONAD performs detection process on each local area network, the alerts, and extracted information of ONAD will be sent to MDE server.

However, when the ONAD sending information to MDE server, network handshaking and delaying may degrade the performance of this system. In order to enhance the performance of our prototype, MOM is used here for delivering information between MDE and ONAD. In this prototype, OpenJMS, which follows the JMS standard [30], is used as the MOM system. With OpenJMS, network traffic usage and system performance can be obviously enhanced for our Two-Layer Intrusion Detection System.

The MDE server of this system, implemented using Java Language, is designed to receive IMDL events from UDP and JMS channels, i.e., our MDE server can not only receive information from JMS server, but also receive information from other applications which report IDML events using UDP datagram. OORB inference engine, which is also implemented in Java, is used in MDE for detect complicated intrusions. Also, several kinds of charts to show the network information are also implemented..

As we mentioned before, any application with the ability of sending IDML events to MDE

server can be treated as the Reporter plugin for our system. In our implemented system, an SNMP information collector is designed and implemented to poll information from SNMP hardware or software and send to MDE server. According to the network address settings and OID information, SNMP collector will retrieve corresponding information from SNMP device and send the information in IDML event format to MDE server. With this SNMP collector, SNMP information can be used as the source of events for MDE to detect, and enhance the detection ability of this prototype.

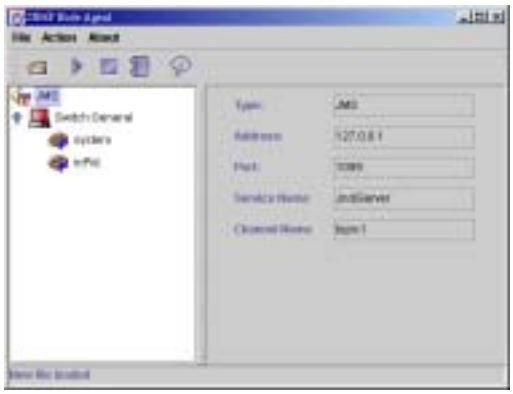

Figure 8: MDE Server. The left part shows the events received. The right part shows the configuration of the server.

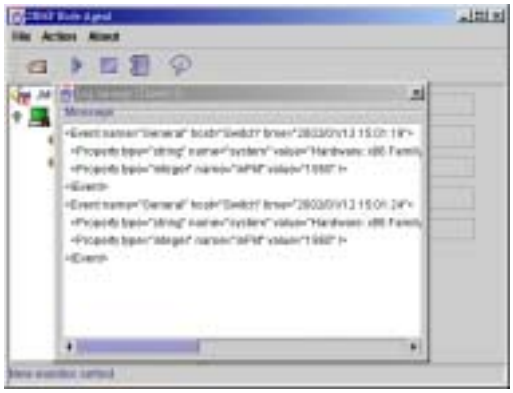

Figure 9: Received IDML event.

4.2 Experiments

Some local network behaviors, which are presumed normal, may be intrusions after some signatures are detected. An example is IP spoofing attacks, which first denies the service of a client A and then spoofs A to connect server B. In this sections, detection model for TFN is described as follows.

TFN (Tribal Flood Network), a distributed denial of service attack, consists of an intrusion master (server) and zombie ants (clients). Unlike some specific intrusion detection tool for TFN, in our system, this intrusion can be modeled without modifying system kernel. In the experiment, the TFN attacking master is at 210.1.2.3, the TFN clients are at 140.113.87.101~105, and the victim is at 140.113.87.25.

The detection model consists of three steps. First, ONAD detects local signatures, which may be a TFN attack. Second, ONAD reports to MDE via MOM. Finally, MDE collects information from all local area network, confirms the intrusion and identifies the source IP of the TFN attacking master. The ONAD patterns, IDML events and MDE rules of each step are described as follows:

1. Patterns of ONAD to detect the local signatures about TFN (four rules only):

```
(Probe)
Pattern   IcmpTypeValue=8
              and   ContentInclude=”1234”
Alert     ”DdosTfnProbe”

(BE)
Pattern   IcmpTypeValue=0
              and   IcmpEchoId=456
              and   IcmpEchoSeq=0
Alert     ”DdosTfnClientCommandBE”

(LE)
Pattern   IcmpTypeValue=0
              and   IcmpEchoId=51021
              and   IcmpEchoSeq=0
Alert     ”DdosTfnClientCommandLE”

(SR)
Pattern   IcmpTypeValue=0
```

```
                and   IcmpEchoId=123
                and   IcmpEchoSeq=0
                and   ContentInclude="73 68 65 6C 6C 20 62 6F 75 6E 64 20 74
                        6F 20 70 6F 72 74"
Alert     “DdosTfnServerResponse”
```

2. IDML event message sent to MDE via MOM (one example only):

```
<?xml version="1.0"?>
<Event>
        <Time>20020701125634</Time>
        <Name>DdosTfnProbe</Name>
        <Attribute>
                <Name>SourceIp</Name>
                <Value>140.113.87.101</Value>
        </Attribute>
        <Attribute>
                <Name>DestinationIp</Name>
                <Value>140.113.87.25</Value>
        </Attribute>
</Event>
```

3. Rules of MDE to detect and identify the TFN attacking master:

```
If              ∃x, y, z, LargerThen(ProbeCount(x, y), 1000)
                and   BE(x, z)
                and   LE(x, z)
                and   SR(z, x)
Then  Response(“TFN Attack, master IP:” + z)
```

Where x, y, z denote the sources of network behaviors. ProbeCount(a, b) is the number of the “Probe” events with the source a and destination b. The relations of BE, LE, and SR are detected by the information sent from ONAD via MOM.

5. Conclusion

In this work, a Two Layer Network Intrusion Detection System is proposed. By combining the ability of effective network activity monitoring and the ability of accurate intrusion pattern expressing and detecting, this system have following properties:

1. Efficiency: Our proposed system overcomes the performance problem in the real world intrusion detection tasks with mass connections along with lots of noises. By transferring the messages via MOM, ONAD and Reporter Plugins do not need to wait for

handshaking and sending message to the MDE, and MDE is free from the risk of the burst networking traffic problems.

2. Accuracy: In the Two Layer Intrusion Detection System, we combine two different types of analysis mechanisms: the misuse expert-system-based MDE and the anomaly data-mining-based NBM. Therefore both current known and unknown intrusion patterns can be modeled and detected, and higher accuracy can be achieved. Mined patterns can be further feedbacked to enhance the detecting capability of the ONAD and MDE.

3. Flexibility: In our system, either the input detectors (Reporter Plugings) or the output actors (Reaction Plugings) are loosely coupled to the kernel of the system. Any external plugins could be added to improve the detecting ability of our system. On the other hand, all components in our system are well-modularized, and can be upgraded easily and rapidly.

A prototype of the Two Layer Network Intrusion Detection System is implemented, and some software technologies are also applied in the prototype to enhance the performance and flexibility of the system. We also describe some intrusion detection models based on our prototype, and these intrusions are usually hard to be modeled and detected in general IDSs. Logical expressions used in this system are able to express complicated intrusions. Thus, this system is expected to be more useful for expressing and detecting unknown advanced intrusion behaviors.

Reference

[1] ADCOM Technology Inc, “Sonic wall”, http://www.adcom.com.tw/product/sonicw/index.htm, 2002

[2] Australian Computer Emergency Response Team, “Distributed Denial of Service Attacks,” http://www.auscert.org.au/Information/Auscert_info/Papers/ddos.html, 2002
[3] CERT Coordination Center, http://www.cert.org, 2002
[4] CERT, “tribe flood network,” http://www.cert.org/incident_notes/IN-99-07.html#tfn, 2002
[5] You-Luen Cheng, Chi-Sung Laih, “The Design and Implementaion of A Distributed Network Intrusion Detection System with The Reconnaissance Ability,” Master’s thesis, Department of Electrical Engineering, National Cheng Kung University, Taiwan, June 2000.
[6] Cisco, “Cisco IDS (Formerly NetRanger),” http://www.cisco.com/univercd/cc/td/doc/pcat/nerg.htm, 2002
[7] CLDP, “CLDP Firewall How to,” http://freebsd.ntu.edu.tw/cldp/Firewall-HOWTO.html, 2000
[8] DDoS World, http://www.ddosworld.com/, 2002
[9] Dave Dittrich, “Distributed Denial of Service (DDoS) Attacks/tools,” http://staff.washington.edu/dittrich/misc/ddos/, 2002
[10] DIDS(Distributed Intrusion Detection System), Motivation,Architecture, and An Early Prototype
[11] Exolab.org, “The OpenJMS Project,” http://openjms.exolab.org/, 2002
[12] FEYA TECHNOLOGIES CO., “Border Ware 6.0,” http://www.feya.com.tw/security/borderware.html, 2002
[13] Ming-Yuh Huang, Robert J. Jasper, Thomas M. Wicks, “A large scale distributed intrusion detection framework based on attack strategy analysis,” Computer Network 31, pp. 2465-2475, 1999
[14] K. Ilgun, R.A. Kemmerer, P. A. Porras “State Transition Analysis: A Rule-Based Intrusion Detection System,” IEEE Transactions on Software Engineering, 21(3), March 1995
[15] K. Ilgun “USTAT: A Real-Time Intrusion Detection system for UNIX,” Proceedings of the IEEE Symposium on Research on Security and Privacy, Oakland, CA, May 1993
[16] R. Kemmerer “NSTAT: A Model-based Real-time Network Intrusion Detection System,” Technical Report TRCS-97-18, Department of Computer Science, University of California, Santa Barbara, November 1997
[17] Yao Tsung Lin, Shian Shyong Tseng, and Shun Chieh Lin, “An Intrusion Detection Model Based Upon Intrusion Detection Markup Language (IDML),” Journal of Information Science and Engineering 17, pp. 899-919, 2002
[18] Shun-Chieh Lin, Shian-Shyong Tseng, and Yao-Tsung Lin, “A New Mechanism of Mining Network Behavior,” Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining 2002, May 2002
[19] Stuart McClure, Joel Scambray, ”InfoWorld Security Sweet 16 (ISS16),” http://www.infoworld.com/cgi-bin/displayNew.pl?/security/links/security_iwss16.htm, 2002
[20] Megasoft Corporation, http://www.taipeisoft.com/Products/WinR/winr.html, 2002
[21] David Newman, Tadesse Giorgis, Farhad Yavari-Issalou, “Intrusion Detection Systems: Suspicious,” http://www.data.com/lab_tests/intrusion.html, August 1998
[22] Neumann, P. Porras ”Experience with EMERALD to data,” 1st USENIX Workshop on Intrusion Detection and Network Monitoring, Santa Clara, California, April 1999, pp.73-80
[23] National Infrastructure Protection Center (NIPC), “Overview of Scans and DDoS Attacks,” http://www.nipc.gov/ddos.pdf, 2002
[24] P. Porras, Phillip A. Neumann, Peter G., “EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances,” http://www2.csl.sri.com/emerald/concepts.html, 1999
[25] P. Porras “STAT – A State Transition Analysis Tool for Intrusion Detection,” Master’s

thesis, Computer Science Department, University of California, Santa Barbara, June 1992

[26] Rao, B.R., “Making the Most of Middleware,” Data Communications International 24, 12 (September 1995): 89-96.

[27] Shiuh-Pyng Shieh, Gligor, V.D., “On a pattern-oriented model for intrusion detection,” IEEE Transactions on Knowledge and Data Engineering, Volume 9, pp 661 -667, July-Aug. 1997.

[28] Sonic Software, “SonicMQ,” http://www.sonicsoftware.com/, 2002

[29] SRI International, “EMERALD,” http://www.sdl.sri.com/projects/emerald/, 2002

[30] Sun Microsystem, “JMS Standard,” http://java.sun.com/products/jms/, 2002

[31] Synscan, http://www.habets.pp.se/synscan/index.php, 2002

[32] SYSWARE CORPORATION, “CheckPoint 2000,” http://firewall.sysware.com.tw/, 2002

[33] Shieh, S. W., Gligor, V. D., “A pattern-oriented intrusion-detection model and its applications,” Proceedings of IEEE Computer Society Symposium on Research in Security and Privacy, pp 327 –342, 1991.

[34] Chi-Feng Tsai, Shian-Shyong Tseng, “Design and Implementation of New Object-Oriented Rule Base Management System,”Master’s thesis, Institute of Computer and Information Science, National Chiao Tung University, , Taiwan, June 2002.

[35] Lindqvist U., Porras, P.A., “Detecting computer and network misuse through the production-based expert system toolset (P-BEST),” Proceedings of the 1999 IEEE Symposium on Security and Privacy, pp 146 – 161, 1999.

[36] Giovanni Vigna, Richard A. Kemmereer “NetSTAT: A Network-based Intrusion Detection Approach,” Computer Security Applications Conference 1998 IEEE, pp. 25-34, 1998