Optimizing Intrusion Detection Pattern Model for Improving Network-based IDS Detection Efficiency

Jai-myong Kim, Kyu-ho Lee, Jong-seob Kim, Kuinam J Kim

ABSTRACT

In this paper, separated and optimized pattern database model is proposed. In order to improve efficiency of Network-based IDS, pattern database is classified by proper basis. Classification basis is decided by the specific intrusions validity on specific target. Using this model, IDS searches only valid patterns in pattern database on each captured packets. In result, IDS can reduce system resources for searching pattern database. So, IDS can analyze more packets on the network. In this paper, proper classification basis is proposed and pattern database classified by that basis is formed. And its performance is verified by experimental results.

1. Introduction

Internet application gets more developed, the harmful activities are increased. For this reason, needs of IDS is increased and varieties of solutions based on misuse detection & anomaly detection method have been presented. As a strong point, the anomaly detection method can detect unknown attacks, but it is difficult to apply to in reality since false alarm rate is too high to detect intrusions only with the extracted normal action models. With the misuse detection method, it is not hard to extract the intrusion detection patterns. But the intrusion pattern is diversified and especially on a network-based intrusion system, the increase of network bandwidth will cause increase in the amount of data to be analyzed instantly. This will effect on the performance limits of the intrusion detection systems process.[1][2]

This paper is focused on to make a specific attack valid-the accordance of the OS or application kind and the version, and it is used for improving the performance of the network based intrusion detection system. If the intrusion detection system is aware of the host information (OS or application kind and version) of the network it detects, it can only match with the intrusion patterns that are valid in the host. In this case, the searching space for intrusion detection pattern gets narrower and can improve the performance of the whole system. For this, we are to present an efficient basis to classify the intrusion pattern database by the host information. And by collecting the real intrusion detection patterns and establishing the databases, we will inspect the functions of the presented models by trial examinations.

2. Intrusion Detection Systems

2.1 Classification of Intrusion Detection Systems

There are several classifications, but two classifications as below are referenced in general. And network-based intrusion detection system that is using misuse detection method is focused on in this paper.

2.1.1 Kumar Classification

In his doctoral thesis "Classification and Detection of Computer Intrusions", S. Kumar, Purdue University, classified intrusions, by the results of the actions, into anomalous intrusions and misuse intrusions[3]. The anomalous intrusion is based on an anomalous action of a computer environment, and misuse intrusion means a well-developed attack pattern. Intrusion detection is classified into anomalous intrusion and misuse intrusion as shown in [Table 1]

Table 1. Kumar Classification

Classification	Intrusion Detection Method
Anomaly Detection	Statistical approaches
	Feature Selection
	Conjunction of anomaly measures
	Predictive Pattern Generation
	Neural Network
Misuse Detection	Conditional Probability
	Production/Expert System
	State Transition Analysis
	Keystroke Monitoring
	Model-based Intrusion Detection

2.1.2 COAST Classification

COAST[4] classifies intrusion detection system by two basis as below

*data source based classification
- Host Based
- Multi-host Based
- Network Based

*intrusion detection model based classification
- Anomaly Detection
- Misuse Detection

2.2 Technical Elements of Intrusion Detection Systems

The intrusion detection system, as shown in [Figure 1], has mainly 4 stages: raw data collection, data reduction and filtering, intrusion analysis and detection, and reporting and response.

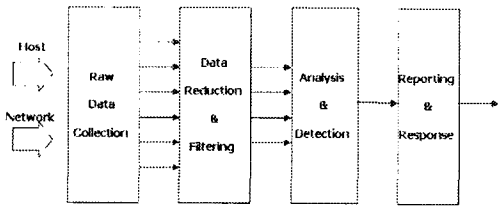


Figure 1. Technical Elements of Intrusion Detection

Raw data collection stage is where the intrusion detection system collects the audit datum- formed by the object being detected such as a network packets or system log files. The host-based itself has a log file which is recorded the specification of the system use so it collects the related data from the files. These collected audit datum then are converted to meaningful information- to make determining of intrusion- in the data reduction and filtering stage- possible.

In the analysis and determination stage, it analyzes and determines whether or not it has been intruded. This stage is the core stage of the intrusion detection system, and it is divided into anomaly detection method and misuse detection method by whether the object is to detect anomalous use of the system or to detect vulnerability of the system or the application program bug.

Further details of this content will be stated in the next chapter. In the reporting and response stage, if the intrusion detection system determines that the system has been intruded, it automatically responses to it or notifies the security administrator to take care of the subject.

Recently the demand for intrusion detection and response is increasing, and moreover there have been researches on intrusion tracing.

System using intrusion pattern model that proposed in this paper is also composed of this basic elements.

3. Models of separated Intrusion Pattern Database

3.1 Intrusion Pattern Database of General IDS

Intrusion pattern database of IDS is classified as following.

First, there is no intrusion pattern database. When Intrusion pattern DB doesn't exist, IDS itself has hard coded intrusion pattern than

having a formalized Intrusion Pattern Database. As a strong point, it can detect attacks that couldn't be presented in formalized patterns and processes very fast because of each detecting routines are optimized. However, IDS, using misuse detection method always have possibilities of facing new patterns and each time new patterns are supplemented programs should be modified.

Second, there is non-separated pattern database. It has formalized intrusion patterns read by IDS. Once the IDS is implemented, new intrusion patterns can be added or updated, allowing great speed against new attacks. But as the number of patterns growing, searching space also increases and this lowers the efficiency. So this model is not used in general.

Third, there is separated pattern database by some basis. Basically its working is similar to former. IDS read the patterns produced by formalized pattern database. Except in this case database is separated by some basis (ex, by port numbers in general; refer to the [Figure 2]). It searches only one pattern database related its port number. In this case, efficiency of IDS declines gradually though the patterns increase. This mechanism is used in many IDS such as SNORT[5], known as public IDS.

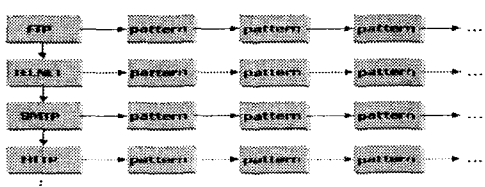


Figure 2. Common Intrusion Pattern DB

3.2 Separated Pattern Database Using Host Information

There are two kinds of intrusion patterns. One is intrusion pattern can be presented in formalized format, and the other is not. The latter is cant be in pattern database, otherwise, it must be implemented as a program. So, in case of this is not mentioned in this paper. Intrusion pattern can be classified as below.

Table 2. Pattern Classification

Classification	Content
Service Ports	- HTTP - SMTP - FTP - TELNET - etc
Operating Systems	- SunOS - HP-UX - AIX - DG/UX - LINUX - Windows - etc

Separated intrusion pattern DB becomes most efficient when the patterns are evenly divided and allocated to the database. Therefore, in this thesis, separated intrusion pattern DB is newly classified to allocate the patterns evenly to the database.

Table 2. Separated intrusion pattern DB

Class	Pattern DB	Explanation	Remarks
Service Ports	HTTP Pattern DB	Etc related to each service port contained	Containing patterns valid to all OS
	SMTP Pattern DB		
	FTP Pattern DB		
	TELNET Pattern DB		
	Etc	Others	
Operating Systems	SunOS Pattern DB	Patterns of attack related to each OS	Containing patterns valid to specific OS
	HP-UX Pattern DB		
	AIX Pattern DB		
	DG/UX Pattern DB		
	Linux Pattern DB		
	Windows Pattern DB		
	Etc	Others	

[Figure 3] shows the effects of intrusion pattern DB that are classified by OS. It is compared with those classified by services.

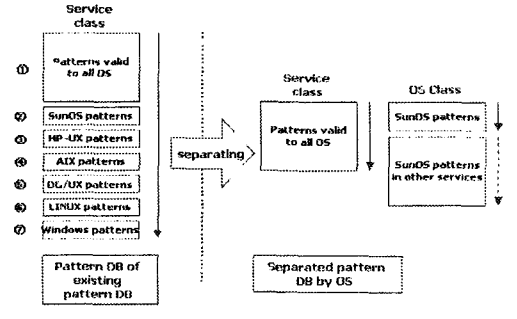


Figure 3. Benefits of Separated Pattern D

If worst, we have to search ①+②+③+④+⑤+⑥+⑦ when using existing pattern DB of IDS. However, if the DB model suggested in this section uses destination host packet, we only need to search ①+②+[SunOS pattern of other services]. [SunOS pattern of other services] is marked with arrows in [Figure 3]. 'Working process of analysis & determination module' compares service port information at the beginning. This means protocol, flag, packet data(data/payload) is not compared, making computing quantity less than searching for ①+②+③+④+⑤+⑥+⑦. Therefore, when Intrusion Pattern DB is used with patterns that are efficient to specific OS, efficiency in detecting intrusions will increase. This is due to decrease in computing quantity that is used in matching the pattern. so IDS can monitor and analyze more packets.

Followings are the facts that could affect the separated intrusion pattern DB.

- Message Queue Transaction Overhead : To transmit reduced audit data to determination module(with separated intrusion DB classified by OS), data duplication is occurred once. This process lowers the efficiency of separated DB.
- The number of intrusion pattern: Many intrusion patterns and equally divided intrusion pattern DB helps to overcome message queue transaction overhead.
- The number of intrusion pattern DB: All the pattern DB, excluding DoS and those classified by Services, is most efficient when it is equally composed with many kinds of pattern DB.

4. Example Network-based IDS using

Separated Intrusion Pattern Database

4.1 Structure of Entire System

Structure of Separated intrusion pattern DB based network IDS is shown in [Figure 4]. Main reason to divide intrusion pattern DB is to reduce the searching space of database. Also, average searching time is reduced since the searching space is balanced at the same time. In this section, we will deal with monitoring host information, audited by IDS. Pattern DB of monitoring host information is classified by OS(Operating system).

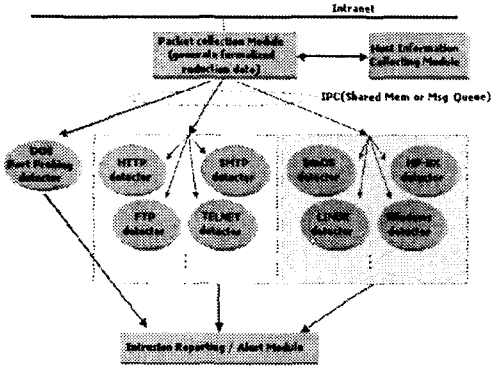


Figure 4. Structure of Entire System

Packet collection module is implemented to pcaplib that is a identified public packet capture library. Pcaplib collects packets of Ethernet level and convert it into formalized reduction audit data. Communications between packet collection module and analysis & determination module is done by message queue which is one of IPC(Inter-Process Communication). Reduced packet data is sent to each intrusion determination module through message queue.

First, host information collection module uses SNMP(Simple Network Management Protocol) to collect the information. Then, this module determines which detecting engine should be started to run the whole system. When the whole system operates, collected packet data and host information is used to appoint detection engine that analyzes and determines the intrusion.

Analysis & determination module that determines each intrusion is classified as following.

- Modules detecting DoS(Denial of Service) and Port Probing patterns.
- Modules detecting patterns common to all OS
- Modules detecting patterns valid to specific OS

Each detection engine is instance of analysis & determination module. Its functions are same, as it runs the same executable file repeatedly. Only difference is the intrusion pattern DB structure.

Intrusion reporting and alert module reports the intrusion to the security administrator. So Security administrator can react properly against intrusion.

5. Experimental Results

Test scenarios are set to run the test accurately; attack sequence process efficiency test for the patterns with the intrusion pattern. All patterns used in this experiment is collected from system & network vulnerabilities in the public web-sites such as SNORT, WHITEHATS, CVE, CERT, etc· [5][6][7][8]

5.1 Objects of Test

Three of pattern DB was loaded in the program to proceed in the test. All the conditions concerning objects are same except for the pattern DB loaded in the memory.

- Linear Model : linear shaped pattern DB without separating pattern DB
- Service Model : Pattern DB divided by service
- OS Model : Pattern DB classified by service and OS

5.2 Attack Sequence Processing Efficiency Test

5.2.1 Purpose

Measure each objects efficiency concerning attack sequence among pattern DB

5.2.2 Methods

- Produce the packet containing attack signature with implemented attack program. Let the packet flow in the network for one minute.
- Measure the number of transferred packet and the number of packet that each object detected as an intrusion
- 112 patterns used in the test are equally divided as pattern DB
- Repeat the test 10 times, calculate average number

Table 2. Pattern DB used in experiment

Model	DB	Number of patterns
Linear Model	Single DB	112
Service Model	HTTP	30
	FTP	27
	SMTP	26
	Etc	29
OS Model	HTTP	15
	FTP	15
	SMTP	16
	Service Etc	19
	UNIX	12
	Linux	12
	Windows	11
	OS Etc	12

5.2.3 Test Environment

Two Workstations and two PCs are connected with Intranet which is a component of Internet. IDS and attack programs are located in each Workstation when packet generator is located in PC. Packet generator will be used for the first scenario and the attack program is for the second scenario.

5.2.4 Results and Analysis

Test results are shown in the [Table 5] below.

Table 3. Experimental Results

Scenario	Total no. of packets transferred (approximate numbers)	Linear Model	Service Model	OS Model
Attack sequence efficiency test	130,000	13,130 detected	41,870 detected	45,220 detected

In attack sequence process efficiency test, OS model detected highest number of packets. OS model is 3.4% times efficient than linear model and 8.0% more efficient than service model. This proves the efficiency of the model. However, the number didn't reach the estimate of 20% that was calculated with separated pattern DB model effect. This is due to the insufficient number of intrusion patterns, expanding overhead of using message queue. So a strong point of suggested model is not performed sufficiently. Accurate analysis in attack patterns and separating pattern DB effectively will bring out more satisfactory results.

6. Conclusions and Future Works

IDS that exists today classifies intrusion pattern database by one database or service. This makes searching space unnecessarily big and patterns exists invalid to target host. To improve this, we have separated intrusion pattern DB by host information (OS type) and reduced the searching space for collected packet data. This has increased the efficiency of IDS. Collected data can be processed faster and more packets can be collected and analyzed. Therefore, in this paper, improved method to detect the intrusion among same hardware platform is presented.

In this paper, patterns are classified only by OS information of target host. But other factors such as variety / different versions of OS, application versions, and patches are also important. Further information on monitoring host and classifying intrusion pattern DB more accurately will bring better outcomes. Moreover, drawing the number of intrusion detection engine process needed in the target network will bring satisfactory results.

Arranging intrusion patterns differently considering network traffic and hacking trends will increase the efficiency of IDS. Continuous researches are expected, related to these parts.

References

[1] B.Mukherjee, T.L. Heberlein, and K.N.Kevitt, "Network intrusion Detection", IEEE Network, 8(3):26-41, May/June 1994

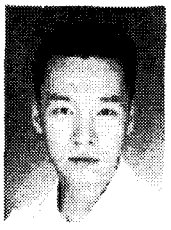
[2] R. Heady, G.Luger, A.Maccabe, and M.Servilla, "The Architecture of a Network Level Intrusion Detection System", Technical Report, Computer Science Department, University of New Mexico, August 1990.

[3] S. Kumar, Classification and Detection of Computer Intrusions, Purdue University, Aug, 1995

[4] http://www.cerias.purdue.edu/coast/intru

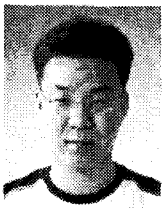
sion-detection/welcome.html
[5] http://www.snort.org/
[6] http://www.whitehats.com/
[7] http://cve.mitre.org/
[8] http://www.cert.org/

이 규 호



1999년 아주대학교 정보및컴퓨터공학부(공학사)
2001년 아주대학교 컴퓨터공학과(공학석사)
2001년 경기대학교 정보보호기술공학과(박사과정)
2001년 ~ 현재 (주)시큐브 선임연구원

김 재 명



1997년 충남대학교 컴퓨터과학과(공학사)
1999년 충남대학교 컴퓨터과학과(공학석사)
2001년 경기대학교 정보보호기술공학과(박사과정)
1996년 ~ 1997년 한국정자통신연구원 위촉연구원
1998년 ~ 2001년 한국정보보호진흥원 연구원
1999년 ~ 2001년 전자서명 공인인증기관 실질심사 위원
2001년 ~ 현재 (주)시큐브 상무이사

김 종 섭

1978년 중앙대학교 법학과(법학사)
2000년 동국대학교 정보보호학과(정보보학석사)
2001년 경기대학교 정보보호학과(박사과정)
1996년 ~ 1999년 한국정보보호진흥원 협력관
1981년 ~ 현재 경찰청 사이버테러대응센터 경감

김 귀 남

미국 캔자스대학 수학과(응용수학사)
미국 콜로라도주립대학 통계학과(통계학석사)
미국 콜로라도주립대학 기계·산업공학과(기계·산업공학박사)
현재 경기대학교 정보보호기술공학과 주임교수