



# Fuzzy Intrusion Detection System

By

Ms. Piyakul Tillapart

Submitted in Partial Fulfillment of  
the Requirements for the Degree of  
Master of Science  
in Telecommunications Science  
Assumption University

November, 2001

# Fuzzy Intrusion Detection System

By

**Ms. Piyakul Tillapart**



**Submitted in Partial Fulfillment of the Requirements for  
the Degree of Master of Science  
In Telecommunications Science  
Assumption University  
November, 2001**

## The Faculty of Science and Technology


### Thesis Approval

Thesis Title                      Fuzzy Intrusion Detection System

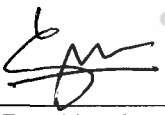
By                                      Ms. Piyakul Tillapart  
Thesis Advisor                      Asst.Prof.Dr. Pratit Santiprabhob  
Academic Year                      1/2001

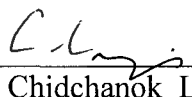
The Department of Telecommunications Science, Faculty of Science and Technology of Assumption University has approved this final report of the **twelve** credits course. **TS7000 Master Thesis**, submitted in partial fulfillment of the requirements for the degree of Master of Science in Telecommunications Science.

Approval Committee:

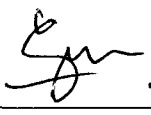
  
(Asst.Prof.Dr. Pratit Santiprabhob)  
Advisor

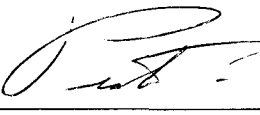
  
(Asst.Prof.Dr. Dobri Batovski)  
Committee Member

  
(Asst.Prof.Dr. Chanintorn J. Nukoon)  
Committee Member

  
(Professor Dr. Chidchanok Lursinsap)  
Representative of Ministry of  
University Affairs

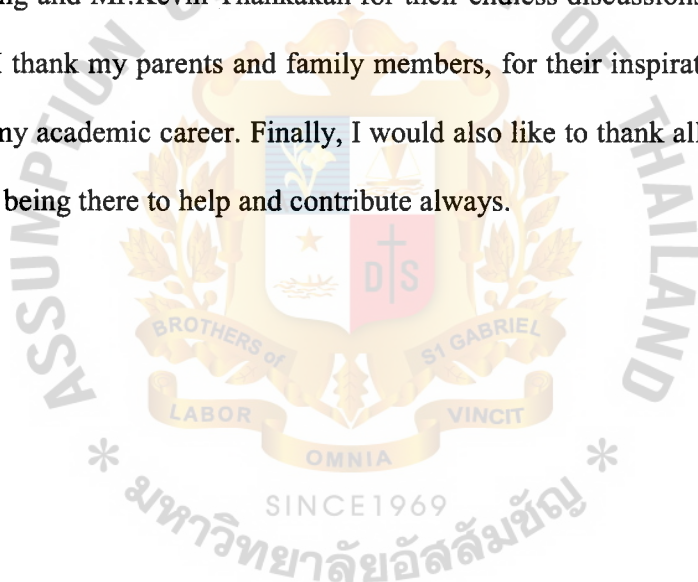
Faculty Approval:

  
(Asst.Prof.Dr. Chanintorn J. Nukoon)  
Director

  
(Asst.Prof.Dr. Pratit Santiprabhob)  
Dean

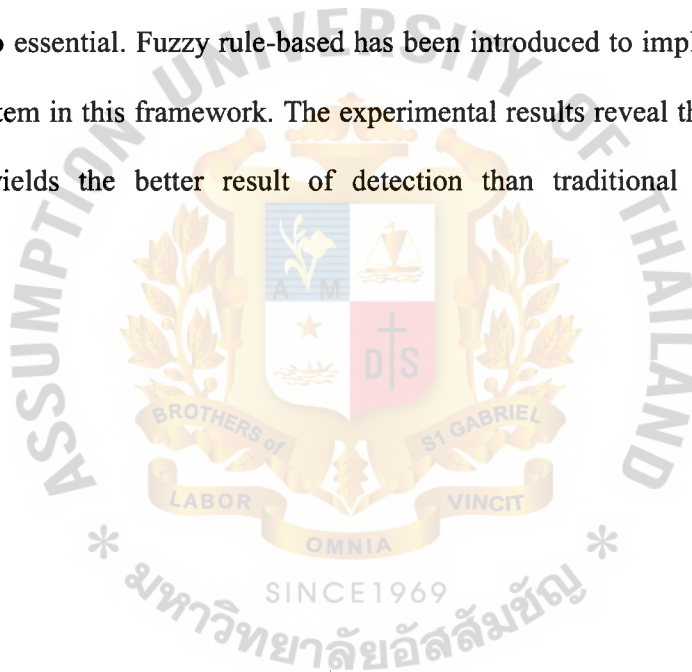
## ACKNOWLEDGMENTS

I would like to express my gratitude to my advisor Asst. Prof. Dr. Pratit Santiprabhob for his constant encouragement and support and the opportunity to work on this research project. He has been a great source of inspiration. I would also like to thank Mr. Thanachai Thamthawatworn for being my co-advisor and my mentor. I would like to thank Asst. Prof. Dr. Chanintorn Jittawiriyankoon and Asst. Prof. Dr. Dobri Batovski for being willing to serve on my committee and for giving their valuable suggestions and feedback. I express my gratitude to colleagues Mr. Sahadol Busyapanpong and Mr. Kevin Thankakan for their endless discussions on the subject. Especially, I thank my parents and family members, for their inspiration and support throughout my academic career. Finally, I would also like to thank all my friends, far and near for being there to help and contribute always.



## ABSTRACT

In this thesis, I propose a framework for intrusion detection system over TCP/IP network. The key idea is to use soft computing for detecting intrusive behaviors and Denial of Service attacks (DoS). The basic intent of a DoS attack either overwhelms the resources allocated by a networked device to a particular service in order to prevent its use, or to crash a target device or system. This will cause disaster in network environment. To protect the most valuable possession from these malicious attempts is so essential. Fuzzy rule-based has been introduced to implement intrusion detection system in this framework. The experimental results reveal that the proposed framework yields the better result of detection than traditional threshold-based detection.



## TABLE OF CONTENTS

ACKNOWLEDGEMENTS	i
ABSTRACT	ii
LIST OF FIGURES	v
LIST OF TABLES	ix
CHAPTER	
CHAPTER 1 INTRODUCTION	1
1.1 BACKGROUND OF INTRUSION DETECTION SYSTEM	1
1.2 MOTIVATION	2
1.3 OBJECTIVES	4
1.4 PROBLEM STATEMENTS	4
CHAPTER 2 SURVEY OF RELATED WORKS	6
2.1 BACKGROUND OF TECHNOLOGY USED	6
2.2 EXISTING SYSTEM AND LITERATURE REVIEW	8
CHAPTER 3 FUZZY INTRUSION DETECTION SYSTEM FRAMEWORK	15
3.1 SCOPE OF WORK	15
3.2 FUZZY INTRUSION DETECTION SYSTEM OVERVIEW	18
3.3 FUZZY RULE-BASE DETECTOR	24
CHAPTER 4 EXPERIMENTAL RESULT AND ANALYSIS	84
4.1 EVENT GENERATION AND GATHERING	84
4.2 TESTING, EXPERIMENTAL RESULTS AND ANALYSIS	84

CHAPTER 5 CONCLUSIONS AND RECOMMENDATION FOR FURTHER REARCH	117
5.1 CONCLUSIOND	117
5.2 RECOMMENDATION FOR FURTHER RESEARCH	118
REFERENCES	120





## LIST OF FIGURES

Figure 1	Pre-processing of the network tcpdump data	10
Figure 2	The organization of intrusion detection system	12
Figure 3	The Fuzzy Intrusion Detection System framework	18
Figure 4	The 1 <sup>st</sup> alternative Fuzzy Intrusion Detection implementation location	20
Figure 5	The 2 <sup>nd</sup> alternative Fuzzy Intrusion Detection implementation location	20
Figure 6	The 3 <sup>rd</sup> alternative Fuzzy Intrusion Detection implementation location	21
Figure 7	The generic detector framework	25
Figure 8	The SYN-Flood detector framework	38
Figure 9	The SYN packet membership function	38
Figure 10	The SYN traffic level membership function	39
Figure 11	The SYN traffic level membership function of SYN-FLOOD DETECTOR BOX	42
Figure 12	The weighted accumulative number membership function of SYN-FLOOD DETECTOR BOX	42
Figure 13	The attack possibility membership function of SYN-FLOOD DETECTOR BOX	43
Figure 14	The UDP-Flood detector framework	47
Figure 15	The membership function of UDP packet frequency	47
Figure 16	The membership function of UDP traffic level	48
Figure 17	The UDP traffic level membership function of UDP-FLOOD DETECTOR BOX	50
Figure 18	The weighted accumulative number membership function of UDP-FLOOD DETECTOR BOX	50
Figure 19	The attack possibility membership function of UDP-FLOOD DETECTOR BOX	51



Figure 20	The Ping-of-Death detector framework	54
Figure 21	The membership function of ICMP reply fragment frequency	54
Figure 22	The membership function of ICMP reply level	55
Figure 23	The ICMP reply level membership function of ICMP REPLY DETECTOR BOX	56
Figure 24	The weighted accumulative number membership function of ICMP REPLY DETECTOR BOX	56
Figure 25	The attack possibility membership function of ICMP REPLY DETECTOR BOX	57
Figure 26	The email bomb detector framework	60
Figure 27	The membership function of mail frequency	60
Figure 28	The membership function of mail level	61
Figure 29	The mail level membership function of MAIL DETECTOR BOX	62
Figure 30	The weighted accumulative number membership function of MAIL DETECTOR BOX	63
Figure 31	The attack possibility membership function of MAIL DETECTOR BOX	64
Figure 32	The FTP password guessing detector framework	68
Figure 33	The membership function of FTP login incorrect packet frequency	68
Figure 34	The membership function of FTP login incorrect level	69
Figure 35	The FTP login incorrect level membership function of FTP DETECTOR BOX	70
Figure 36	The weighted accumulative number membership function of FTP DETECTOR BOX	70
Figure 37	The attack possibility membership function of FTP DETECTOR BOX	71
Figure 38	The Telnet password guessing detector framework	73

Figure 39	The Telnet login incorrect level membership function of TELNET DETECTOR BOX	75
Figure 40	The weighted accumulative number membership function of TELNET DETECTOR BOX	75
Figure 41	The attack possibility membership function of TELNET DETECTOR BOX	76
Figure 42	The port scanning detector framework	80
Figure 43	The SYN-RESET pair frequency membership function	80
Figure 44	The attack possibility membership function of PORT SCAN DETECTOR BOX	81
Figure 45a	The detection result of 1 <sup>st</sup> SYN-Flood testing, FIDS vs. Threshold 1	87
Figure 45b	The detection result of 1 <sup>st</sup> SYN-Flood testing, FIDS vs. Threshold 2	87
Figure 45c	The detection result of 1 <sup>st</sup> SYN-Flood testing, FIDS vs. Threshold 3	88
Figure 46a	The detection result of 2 <sup>nd</sup> SYN-Flood testing, FIDS vs. Threshold 1	90
Figure 46b	The detection result of 2 <sup>nd</sup> SYN-Flood testing, FIDS vs. Threshold 2	91
Figure 46c	The detection result of 2 <sup>nd</sup> SYN-Flood testing, FIDS vs. Threshold 3	91
Figure 47a	The detection result of 3 <sup>rd</sup> SYN-Flood testing, FIDS vs. Threshold 1	92
Figure 47b	The detection result of 3 <sup>rd</sup> SYN-Flood testing, FIDS vs. Threshold 2	93
Figure 47c	The detection result of 3 <sup>rd</sup> SYN-Flood testing, FIDS vs. Threshold 3	93
Figure 48a	The detection result of UDP-Flood testing, FIDS vs. Threshold 1	96
Figure 48b	The detection result of UDP-Flood testing, FIDS vs. Threshold 2	96
Figure 48c	The detection result of UDP-Flood testing, FIDS vs. Threshold 3	97

Figure 49a	The detection result of Ping-of-Death testing, FIDS vs. Threshold 1	100
Figure 49b	The detection result of Ping-of-Death testing, FIDS vs. Threshold 2	100
Figure 49c	The detection result of Ping-of-Death testing, FIDS vs. Threshold 3	101
Figure 50a	The detection result of 1 <sup>st</sup> email bomb testing, FIDS vs. Threshold 1	104
Figure 50b	The detection result of 1 <sup>st</sup> email bomb testing, FIDS vs. Threshold 2	104
Figure 50c	The detection result of 1 <sup>st</sup> email bomb testing, FIDS vs. Threshold 3	105
Figure 51a	The detection result of 2 <sup>nd</sup> email bomb testing, FIDS vs. Threshold 1	106
Figure 51b	The detection result of 2 <sup>nd</sup> email bomb testing, FIDS vs. Threshold 2	106
Figure 51c	The detection result of 2 <sup>nd</sup> email bomb testing, FIDS vs. Threshold 3	107
Figure 52	The detection result of 2 <sup>nd</sup> FTP password guessing testing	110
Figure 53a	The detection result of 1 <sup>st</sup> Telnet password guessing testing, FIDS vs. Threshold 1	112
Figure 53b	The detection result of 1 <sup>st</sup> Telnet password guessing testing, FIDS vs. Threshold 2	112
Figure 54	The detection result of 2 <sup>nd</sup> Telnet password guessing testing	113
Figure 55a	The detection result of port scanning testing, FIDS vs. Threshold 1	115
Figure 55b	The detection result of port scanning testing, FIDS vs. Threshold 2	116

## LIST OF TABLES

Table 1	The weighted accumulative number example	26
Table 2	Example of preprocessed SYN-packet data within 1 <sup>st</sup> second	37
Table 3	Example of SYN traffic rate and weighted accumulative in 10 <sup>th</sup> second	41
Table 4	Example of preprocessed UDP packet in 1 <sup>st</sup> second	46
Table 5	Example of SYN traffic rate and weighted accumulative in 10 <sup>th</sup> second	50
Table 6	Example of preprocessed ICMP reply fragment in 1 <sup>st</sup> second	53
Table 7	Example of preprocessed incoming emails in 3 seconds	59
Table 8	Example of preprocessed FTP <i>login incorrect</i> connection in 1 <sup>st</sup> second	67
Table 9	Threshold level of SYN-Flood Threshold 1	86
Table 10	Threshold level of SYN-Flood Threshold 2	86
Table 11	Threshold level of SYN-Flood Threshold 3	86
Table 12	Threshold level of UDP-Flood Threshold 1	95
Table 13	Threshold level of UDP-Flood Threshold 2	95
Table 14	Threshold level of UDP-Flood Threshold 3	95
Table 15	Threshold level of Ping-of-Death Threshold 1	98
Table 16	Threshold level of Ping-of-Death Threshold 2	99
Table 17	Threshold level of Ping-of-Death Threshold 3	99
Table 18	Threshold level of email bomb Threshold 1	102
Table 19	Threshold level of email bomb Threshold 2	103
Table 20	Threshold level of email bomb Threshold 3	103
Table 21	Threshold level of FTP Password Guessing Threshold 1	108

Table 22	Threshold level of FTP Password Guessing Threshold 2	109
Table 23	Threshold level of FTP Password Guessing Threshold 3	109
Table 24	Threshold level of Telnet Password Guessing Threshold 1	111
Table 25	Threshold level of Telnet Password Guessing Threshold 2	111
Table 26	Threshold level of port scanning Threshold 1	114
Table 27	Threshold level of port scanning Threshold 2	115



# CHAPTER 1: INTRODUCTION

## 1.1 Background of Intrusion Detection System

Network grows very fast in its size and many networks are tied together to form the Internetwork. Then the network resources, for example hosts, network bandwidth or information, are the most valuable possession for all the organizations today. The more the value, the more the danger. Consequently these network resources become the targets for both the investors and attackers.

Nowadays, the system penetrations that come from both inside and outside the network are very terrible. There are not only organizations' web pages that were attacked by the hackers but also the servers, which provide services to the customers or subscribers, were compromised by the intruders. Therefore these organizations could not provide services for some moment of time.

To protect these network resources from the intruders, the intrusion detection systems (IDSs) have been developed. The IDS is used for detecting the intrusions that are defined to be unauthorized uses, misuses, or abuse of computer system by authorized users or external perpetrator. There are two basic techniques of intrusion detection system: 1) anomaly detection and 2) misuse detection.

### **Anomaly Detection**

Anomaly detection technique establishes the normal patterns (normal activity) of computer usage, called profile, such as the CPU usage, login/logout time, and

application used by a particular user. If the anomaly detection system notices the differences between the incoming traffic from the normal pattern, this event is considered to be the intrusion.

### **Misuse Detection**

Misuse detection technique uses a collection of intrusion signatures, which are specific and precisely representative techniques of computer system abuse. It tries to match the incoming traffic to the known signatures. If the incoming traffic is coincident with the signature, it is considered to be intrusion or attack.

Moreover, the intrusion detection systems can be categorized into 1) host-based IDSs and 2) network-based IDSs.

Host-based IDSs are used to secure critical network servers of other systems containing sensitive information. Network-based IDSs monitor activity on a specific network segment. Unlike host-based agents, network-based systems are dedicated platform, which analyzes the network traffic and display alarm information.

## **1.2 Motivation**

Since all the network resources are very essential, all organizations must maintain these resources for providing the services to the customers 24 hours a day 7 days a week. Then these resources are attracting the attackers.



According to the report in [4], it has reported that DDoS attacks hits the most popular web sites. Yahoo, for example, is hit on Monday 7. Next day, Buy.com is also attacked. Next nine hours, eBay is hammered by DDoS attacks. During this period, DDoS attacks breaks many web sites and service servers, for example CNN.com, Amazon.com, ZDNet, E\*Trade, and Exite.

To do attacking to these servers and network resources, such as hang it up, slow its services down, and overwhelm the processes, will cause a disaster. These explained attacks are called "Denial of Service." Denial of Service (DoS) can be accomplished by both the single machine and several machines. DoS that is accomplished by several machines is called Distributed Denial of Service attacks (DDoS).

So, many organizations try to develop and employ the intrusion detection systems that can detect these attacks. Even though there are a lot of available intrusion detection systems, however there are no one that can detect completely all types of attacks. Most of them are not up-to-date concerning with the new techniques of attack.

All above reasons motivate me to study about not only the proper technique for detecting the intrusions but also technique that will be used to update the system due to the new types of attack. Hence a proposal for my thesis is introduced.

### 1.3 Objectives

There are many intrusion detection systems available today. Some of them can do only the detection function if they are misuse detection. These misuse detection systems can only detect the intrusion based on known attack signatures. They do not contain the mechanism that can update the new attack types. The software must be changed when the new attack signatures are discovered.

The main objective for this thesis is to build up the real-time intrusion detection system framework, employ fuzzy rule-base technology in order to detect the intrusive behaviors and Denial of Service attacks. Moreover, the data mining technology is learned and it is used for discovering the patterns from the large data [5]. Therefore, the data mining technique is used as a tool for discovering the attack patterns from the network traffic.

### 1.4 Problem Statements

The proposed IDS framework is developed for detecting only specific types of attack (SYN-Flood attack, UDP-Flood attack, Ping-of-Death attack, email bomb attack, password guessing on FTP and Telnet server and port scanning.) The capability of detecting depends on the attack signatures in the database. Due to the fast development of computer technology, the attack patterns or the suspicious behaviors also change very rapidly. Then the proposed IDS cannot adapt to these changes.

Therefore, the performance improving for the proposed IDS, I need to update the attack signatures and update the rule-base sometime.



## CHAPTER 2: SURVEY OF RELATED WORKS

### 2.1 Background of Technology Used

#### Fuzzy Technology

Many intrusion detection schemes, which are currently available, are not flexible. Since the IDSs, which use the attack signatures for detecting the penetration, are based on the expert knowledge. To getting the attack patterns, the experts must examine the huge audit data (audit log file), which is the most difficult task and time consuming. Not only the difficulty of getting the attack signatures but also the techniques used to classify which traffic is intrusion and which is not clearly defined. They are too ambiguous and not very concise about the detecting techniques of the experts [13].

Consequently, to deal with these ambiguous detecting techniques, fuzzy technology is used for implementing this intrusion detection system.

#### Data Mining Technology

In addition to the fuzzy technology, the data mining technology is also used for discovering the attack pattern from the large audit data.

Data mining is currently the popular concept. This concept is made used in many application areas. Data mining can discover the understandable patterns, which are unknown or hidden patterns, from the large data. There are many data mining techniques, for example decision tree, neural networks, classification, association

rules, frequent episodes and others. From these mining techniques, the association rule is used in my system. The algorithm is described in [6].

### Association Rules

Association models are models that examine the extent to which values of one field depend on or are predicted by values of another field within the same record. Association discovery finds rules, which show the relation between the items, when the items appear together in an event. Association discovers things that go together.

From [5][6], an association rule is the expression

$$X \rightarrow Y, c, s$$

Where X and Y are items, and  $(X \cap Y) = \emptyset$ .  $s = \text{support}(X \cup Y)$  is the support of the rule, and  $c = \frac{\text{support}(X \cup Y)}{\text{support}(X)}$  is confidence. Define support (X) as the percentage of transactions (records) in the database that contain itemset X. Here is the example of association rule:

$$\text{Introduction to Unix} \rightarrow \text{Programming in C} (0.84, 0.34)$$

Which states that 84% of the students that take Introduction to Unix, also take Programming C, and 43% of all the students actually have taken both courses.

This technique is used to find out the parameters that are related to each other. For example:

$$\text{Flag} = \text{SYN} \rightarrow \text{Destination address is same}$$

If this event consecutively occurs within the specified period of time, which identifies that this is SYN-flood attack. These parameters are used to build up the fuzzy rule-base.

I use the data mining software called DM II - CBA from National University of Singapore as a tool for discovering the important parameters.

## 2.2 Existing Technologies and Literature Review

The intrusion detection systems are very attractive and these systems cannot completely detect all the intrusions because the intruders always change their techniques when entering into the protected system. As the result, many researchers and developers have exploited their ideas in employing many techniques for detecting the new intrusion behaviors.

### **Literature Review for *A Sense of Self for Unix Process***

Here the correlation technique is employed for detecting the normal and abnormal patterns of the system calls as described in [3]. Forrest has introduced the method for anomaly detection in which "normal" is defined by the short-range correlation in a process's system calls. This system is employed for standard UNIX programs and it is able to detect several common intrusions involving *sendmail* and *plp*. This system is based on the way natural immune systems distinguish self from other, whereby self is statistical of legitimate activities. The system can distinguish between self and dangerous foreign activities.

Forrest proposes two stages of the algorithm. First stage is scanning traces of normal behavior and building up a database of characteristic normal patterns (observing

sequences of system calls). Second stage is scanning traces that might contain abnormal behavior, looking for patterns not present in the normal database.

To build up the database, a widow of size  $k + 1$  is slid across the trace of system calls and record which calls follow which within the sliding window. For example when  $k = 3$  and the sequence of normal system call behavior is

open, read, mmap, mmap, open, getrlimit, mmap, close

Then the following database is produced:

call	position 1	Position 2	position 3
open	read,	Mmap	mmap,
	getrlimit		close
read	mmap	Mmap	open
mmap	mmap,	open,	getrlimit,
	open,	Getrlimit,	mmap
	close		
getrlimit	mmap	Close	
close			

After generating the normal database from the trace of normal *sendmail*, which should cover the full spectrum of normal, the system measures new behavior and determines whether it is normal or abnormal. The system simply counts the number of mismatches between a new trace and the database. The system uses a threshold value to classify the normal and abnormal, below which a behavior is said to be normal and above which it is deemed anomalous. The algorithm used to find out the mismatch is defined in [3]



As this system is anomaly detection system, which builds the database of normal patterns, then the success of the system depends on the complete trace of normal system calls. If the trace of normal system does not complete all normal behavior, the detecting result might not be correct.

**Literature Review for Algorithms for Mining System Audit Data**

This paper shows the data mining techniques that are used for constructing intrusion models [6]. The idea of data mining is discovering the consistent and useful patterns of program and user behavior. Two data mining techniques, association rules and frequent episodes, are used to compute the relevant (useful) patterns. The algorithm of these two are shown in [6].

The input data is the network *tcpdump* data, where this data is preprocessed to be the connection records. Then these connection records pass through the mining system.

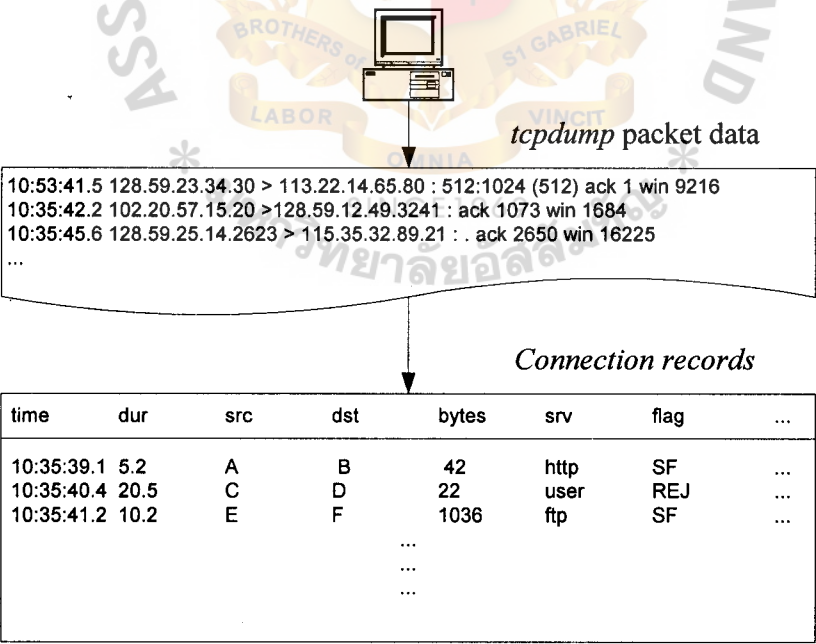


Figure 1: Pre-processing of the network *tcpdump* data

There are two mining techniques that are used in this work. Association rule that discover the patterns, which shows the relevance of the attributes within the record (intra-record). For example, an association rule from the shell command history file of a user is

$trn \rightarrow rec.humer, 0.3, 0.1$

which indicates that 30% of the time when the user invokes *trn*, he or she is reading the news in *rec.humor*, and reading this newsgroup account for 10% of the activities recorded in his or her common history file.

Next is frequent episode that discovers the patterns, which shows the relation between records (inter-record) within the interval of time. For example, the frequent episode generates the following rule from the network tcpdump data,

(service = smtp, src\_bytes = 200, dst\_bytes = 300, flag = SF),

(service = telnet, flag = SF)  $\rightarrow$  (service = http, src\_bytes = 200)

Note that each itemset of the episode rule, such as (service = smtp, src\_bytes = 200, dst\_bytes = 300, flag = SF) is an association.

Using the association rule and frequent episode, the system can discover the essential attributes and the relation among records for creating the rules (patterns). This paper uses them for creating the pattern of abnormal behavior.

### **Literature Review for Neural Network Applied in Intrusion Detection**

Neural networks are applied in the intrusion detection systems as described in [2]. This work presents a prototype of an intrusion detection system for TCP/IP networks. The system works by capturing packets and using a neural network to identify an

intrusion behavior within the analyzed data stream. The identification is based on previous well known intrusion profiles. The system is adaptive, since the new profiles can be added to the database. This model uses Neural Networks to find out intruders. The model is considered to be network-base system, which uses audit trails, analyzes the traffic of packets within the network to detect intrusion behavior. The model comprises a security agent able to detect intrusive behavior in established connection. This agent acts by capturing and deciphering packets, which are transmitted through the network under monitoring. In connections, the agent will employ an Expert System and a Neural Network, which can give an idea about the severity of attack or the degree of suspicion of the activities in that connection. The system is based on an intrusion that can be detected from an analysis of predetermined models, which are anomalous compared with normal actions. In the system, the agent is placed in a safe machine, which is placed at sensitive points of the network system. The agent is organized in a four-layer model.

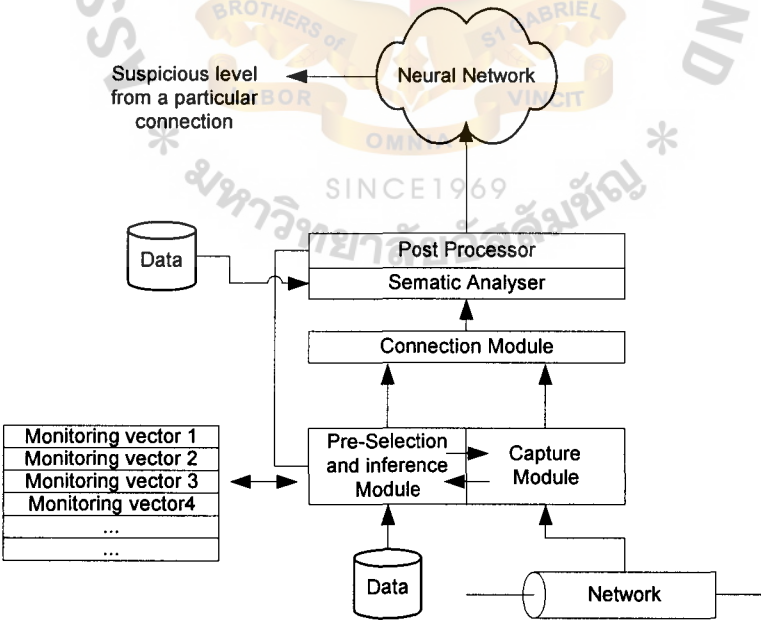


Figure 2: The organization of intrusion detection system

1) The lowest level captures a flow of data in the network and passes the ordered packets to the second layer. 2) There are two modules in second layer: Packet Pre-selection module and Expert module. The packet pre-selection module makes an initial packet filtering that may represent interesting events. Then the filtered packets pass through the expert system where the analysis function is performed. The expert system uses information of the expected paths of connection source and destination, source and destination ports involved, sensibility of the machines and reliability of domains, when making decision. The expert system uses the *security value* for each connection for determining the intrusion behavior. When the security value reaches a pre-established threshold, the expert system *locks in*, by using vector connection or alternatively, all connections coming from that domain. Then the system sends these abnormal connections to the next level. 3) In the third module, the vectors are passed through the semantic analyzer, which acts upon connection vectors, searching for attack profiles that would appear the data. The profiles, which are attack signatures containing information on how a suspicious session behaves, are stored in a database and updated according to need. 4) The last module receives the vector that matches to the profiles, then the *stimulus vector*, which contains values and information about the connection, is formed. The neural system analyses the stimulus vector with its respective weights, representing the importance of the occurrence of events, and tries to attribute a suspicious degree, representing the suspicious state of a particular connection.

As this system is implemented by using neural network technology, before the neural network system can identify potential attacks, it must be trained with a meaningful and large amount of stimulus vector, which represents the behavior of the suspicious

connections and the legitimate ones. In addition, if there are any new patterns or new types of suspicious connections added to the system, the neural system is needed to retrain in order to learn to recognize the new pattern appropriately.



# CHAPTER 3: FUZZY INTRUSION DETECTION

## SYSTEM FRAMEWORK

### 3.1 Scope of Work

The purpose of this thesis is to develop a real-time, Fuzzy Intrusion Detection System (FIDS) that is able to detect and suspect the Denial of Service attacks (DoS) and other types of malicious behaviors.

FIDS is designed to deal with several kinds of attacks: malicious behaviors such as port scanning and password guessing on FTP and Telnet server and DoS attacks such as SYN-flood attack, UDP-flood attack, Ping-of-Death attack and email bomb.

#### Types of Attack

##### 1. Denial of Services

DoS can be used to attack the victim host or the network by both single attacking machine (tradition technique) and numbers of attacking machines, called Distributed Denial of Service attacks (DDoS). Due to DDoS attacks, the hacker's system sends the packets or deposits zombie program to intermediary victims for sending the specified packets to the ultimate victim.

##### **SYN-Flood Attack**

It is the continual creation of the "half-open" connections to the target system (victim). The attack machine sends the victim the SYN packets for opening the

connections but it does not response to the SYN-ACK packet sent from the victim. The attack machine might be spoofed the return address, it then sends the victim a high number of the SYN packets within a short period of time (thousands within seconds by using sophisticate attack program such as TFN2K or Trinoo). This causes the half-open connections, which are unusual and should be evident in the analysis. In DDoS, the intermediaries with zombie program send the victim a SYN packet simultaneously, this is demanded by the zombie control program in the hacker's machine. Then these SYN packets from different sources should be detected.

### **UDP-Flood Attack**

Attacking machine uses UDP technology by sending a victim a lot of UDP packets. The recipient must respond or process these UDP packets. This causes the services provided by the victim are slow down. In DDoS, hackers have used UDP technology to launch DDoS attacks. For example, by sending UDP packets with spoofed return address, a hacker links one system's UDP character-generating (chargen) service to another system's UDP echo service. As the chargen-service keeps generating and sending characters to the other system, whose echo service keeps responding, UDP traffic bounces back and forth, preventing the systems from providing services.

### **Ping of Death Attack**

Ping of Death is a large ICMP request packet sent by attacking machine to a victim. The target receives the ping in fragments and starts reassembling the packet. A fragmented packet size is larger than 65536 bytes in length. However, due to the size of packet once it is reassembled it is too big for the buffer and overflows it. This causes unpredictable results, such as reboots or system hangs.



## Email Bomb

Email bomb is the intentional delivering of very large volumes of the email messages to particular mailbox and mail server with the intention of flooding a persons mailbox of mail server or degrading the server's performance. Many servers and particular mailbox can't handle large volumes of email and this will cause legal users to be denied service or may lose email messages for legal users.

## 2. FTP and Telnet Password Guessing

This could be considered as data attack (since the password is part of the data field) however the repeated connection attempts of incorrect password for either the *ftp* and *telnet* port might be detectable. The issue is whether the analysis would reveal a high rate of connections from one node to another accompany with the incorrect password. A normal set of *ftp* and *telnet* connections might be a connection every few minutes on average. Automated password guessing attacks could generate hundreds per minute.

## 3. Port Scanning

There are many network scanners available. The pattern analysis should reveal an outside node that is attempting to scan a network to see which services are on the network and what ports (i.e. network applications) they advertise Scanning Attack Identifiers. The attribute, which shows the attack, is the number of nodes and ports being accessed from one source system. Typically, individual systems access few others per day. These accesses are typically limited set of ports. Scanning creates connections to hundreds of systems and thousands of ports.

### 3.2 FIDS Overview

The main purpose of this system is to detect the intrusive traffics by employing fuzzy rule-based system and to alter the system administrator (SA) about these attacks. The Fuzzy Intrusion Detection System (FIDS) framework is shown in Figure 3. Using fuzzy rule-based system, FIDS can make decision of penetration more flexible and can overcome the sharp boundary in determining between normal and abnormal network traffic.

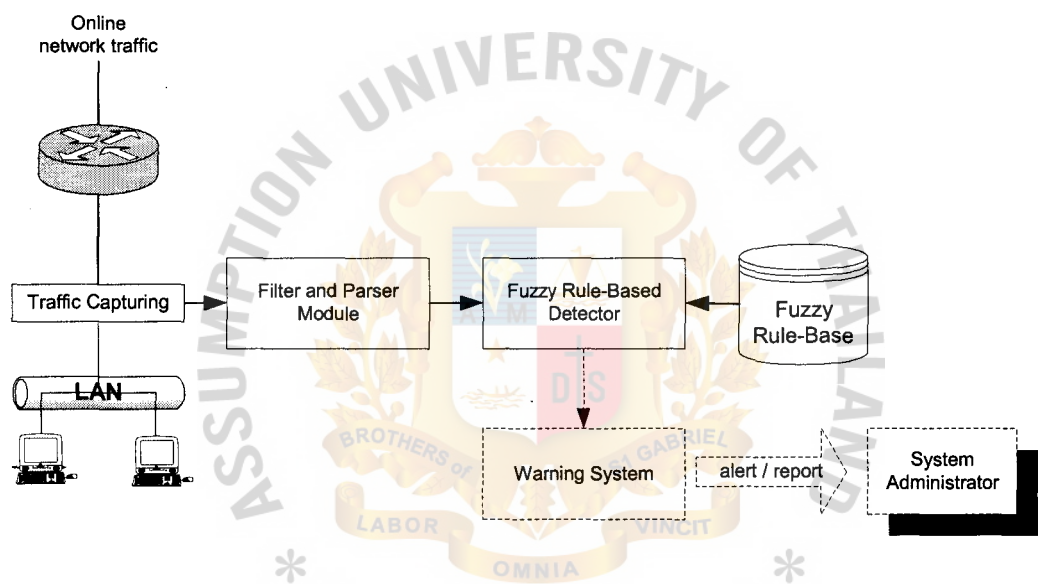


Figure 3: The Fuzzy Intrusion Detection System framework

Rather than using crisp value (threshold-based detection) to distinguish between the normal and abnormal network traffic, FIDS uses fuzzy rule-based system. If the crisp value is used, the quantity of intrusive packets lower than the threshold are considered to be normal traffic, even though this traffic may degrade the performance of network or server. Therefore, using fuzzy rule-based system, certain amount of abnormal

traffic that are in between normal and attack can be considered as abnormal (with some low degree of suspicious).

The FIDS framework comprises of three main components. The first component is *Filter and Parser Module* (FPM). The second component is *Fuzzy Rule-Based Detector* (FD) and the last one is *Warning System* (WS). FPM, the captured packets are filtered and collected according to the pre-defined attack signatures. FD analyzes the attack severity (attack possibility) of the filtered traffic. The last component, if the attacks are detected, WS displays the detected attacks' information and creates attack report for administrator.

Notwithstanding, this thesis focuses on both Filter and Parser Module and Fuzzy Rule-Based Detector.

#### FIDS Implementation Locations

There are several locations that the Fuzzy Intrusion Detection System can be implemented.

- 1) It can be implemented at the critical point of the network (the point of interconnection between internal network and external network) as shown in Figure 4.
- 2) It can be implemented after the gateway or the router as a firewall as shown in Figure 5.
- 3) It can be built into the router as shown in Figure 6.

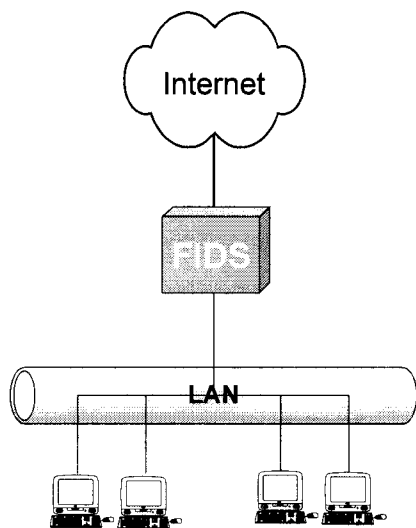


Figure 4: The 1<sup>st</sup> alternative Fuzzy Intrusion Detection System implementation location

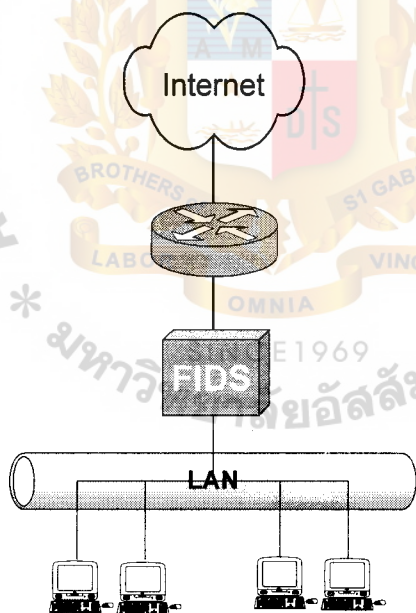


Figure 5: The 2<sup>nd</sup> alternative Fuzzy Intrusion Detection System implementation location

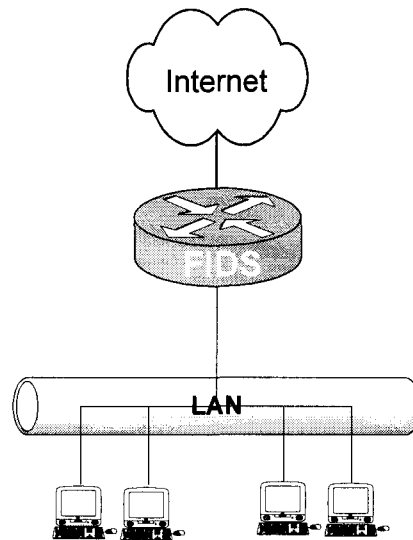


Figure 6: The 3<sup>rd</sup> alternative Fuzzy Intrusion Detection System  
implementation location

Therefore implementing FIDS at any of these locations, most of inbound and outbound traffic can be analyzed.

#### Filter and Parser Module (FPM)

This is the important module. The network traffic is usually enormous and a possible attack is hidden in it. Hence it is essential to filter the necessary information. Therefore an appropriate filter is necessary. Two main functions of this module are to filter and to collect the necessary information. To filter the traffics, the FPM captures and maps both inbound and outbound network traffic with the pre-defined intrusive patterns (attack signatures). The captured packets that match with the pre-defined signatures are collected.

To obtain the attack signatures, expertise observations and data mining technique [1], [2] have been employed. The data mining is used to discover the unknown patterns

from large data set obtained from the network traffic [5], [6]. The followings present the intrusive patterns obtained by using observing and data mining technique.

### SYN-Flood Signature

According to the nature of SYN-Flood attack (half-open connections), the attacking machines send a flood of SYN TCP/IP package to a destination IP number and destination port and the source IP numbers of those packets are spoofed. Therefore, the signature of SYN-Flood attack can be derived as:

$$flag = S, dst\_host = victim (same), dst\_service = vulnerable port (same)$$

The pattern indicates the TCP packet contains SYN, which has been sent to the same destination IP number and the same destination port.

### UDP-Flood Signature

UDP-Flood attack, it sends a flood of UDP packets to a destination IP number but it can send to a single destination port or several destination ports and the source IP number of those UDP packets are also spoofed. Consequently, the signature of UDP-Flood attack is set up as:

$$dst\_host = victim (same), dst\_service = vulnerable port/random port$$

The pattern indicates the UDP packets that have been sent to the same destination.

### Ping of Death Signature

Ping of death attack (packet carries ICMP reply packet):

$$src\_host = victim (same), fragment\_identification = same$$

The pattern indicates that the packet that contains ICMP reply sent from the same IP number and has same fragment identification number.

### Email Bomb Signature

Due to the nature of email bomb the victim, mail server, has received a flood of mails from the same source. Therefore the signature can be derived as:

*src\_host = bombing machine (same), dst\_host = victim (same),*

*recipient = email-address (same), dst\_port = smtp*

The pattern indicates that the packet contains the same recipient email address from the same source IP number (sending mail server) and to the same destination IP number (receiving mail server).

### Password Guessing Signature

FTP password guessing:

*src\_host = victim (same), src\_service = ftp, dst\_host = guessing machine (same),*

*ftp\_data = "login incorrect"*

The pattern indicates the FTP packet contains data of "login incorrect" sent from victim to guessing machine.

Telnet password guessing:

*src\_host = victim (same), src\_service = telnet, dst\_host = guessing machine*

*(same), ftp\_data = "login incorrect"*

The pattern indicates the Telnet packet contains data of "login incorrect" sent from victim to guessing machine.

### Port Scanning Signature

Port scanning pattern (packet carries TCP packet):

*(flag = S, src\_host = attacking machine, dst\_service = vulnerable port) =>*



*(flag = R, dest\_host = attacking machine, src\_service = dst\_vulnerable port)*

The pattern indicates the pairs of SYN packets and RESET packets that are sent from/to the same source host consequently.

### Fuzzy Rule-Based Detector

This component is the engine of FIDS. It composes of seven detectors:

1. SYN-Flood detector
2. UDP-Flood detector
3. Ping of death detector
4. Email bomb detector
5. FTP password guessing detector
6. Telnet password guessing detector
7. Port scanning detector

### **3.3 Fuzzy Rule-Based Detector**

In this section, the detection mechanisms using to detect several types of attack, mentioned above, are discussed in more detail.

#### Generic detector framework

Most of detectors comprise of two fuzzy rule boxes, LEVEL BOX and DETECTOR BOX (except Port Scan Detector has only DETECTOR BOX). The first fuzzy rules box, LEVEL BOX, receives the frequency of packets, matched with the attack signature, from the FPM and then normalizes this input to become a traffic level.

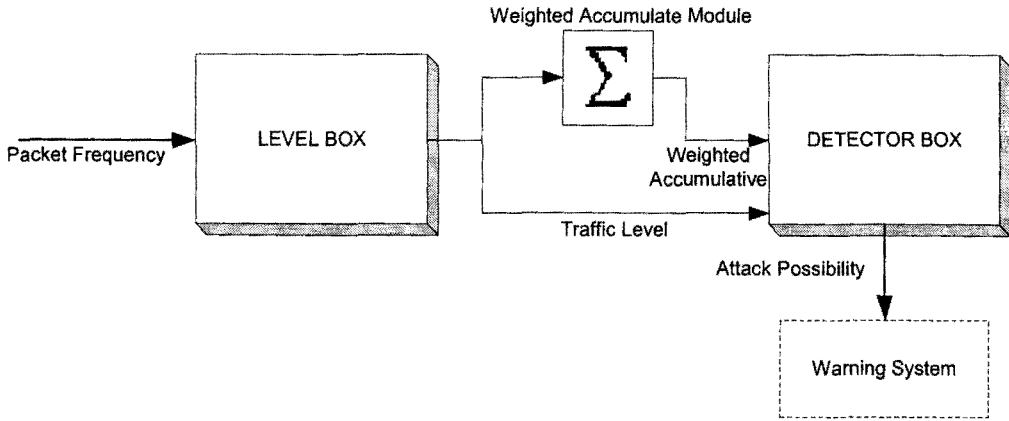


Figure 7: The generic detector framework

Therefore, this normalized number is used as the first input of the second fuzzy rule box, DETECTOR BOX. The normalized number indicates the level of the malicious traffic at current second. It is also used by the additional module, called Weighted Accumulate Module. Weighted Accumulate Module determines the amount of the malicious traffic in previous seconds/minutes. Consequently, the output is weighted accumulative number.

#### How to figure out the weighted accumulative number

To detect the intrusion, the second fuzzy rule box, DETECTOR BOX, uses the traffic level in current second and the amount of malicious traffic in the past seconds for determining the attack possibility. Which the amount of malicious traffics in the past consecutive second should affect the attack possibility of current second much more than other past seconds. Therefore the following formula is used by Weighted Accumulate Module to find out the weighted accumulative number of current time ( $t$ ).

$$\text{Weighted accumulative number } (t) = \sum_{i=0}^a (1 - 0.1i) \text{Traffic\_level}(t - i)$$

The traffic level and weighted accumulated number examples are shown in below.

Table 1: The weighted accumulative number example

Time (second)	Frequency	Traffic Level	Weighted Accumulative number
1	4	0.0004	0.0004
2	122	1.6229	1.62326
3	144	1.9042	3.36513
4	117	1.5696	4.58198
5	96	1.377	5.44927
6	126	1.6677	6.46956
7	127	1.6795	7.33488
8	139	1.832	8.18475
9	138	1.818	8.83742
10	116	1.5595	9.04979
11	104	1.4447	8.99141
12	143	1.8889	9.2328
13	99	1.4018	8.96049
14	118	1.5798	8.91642
15	106	1.4622	8.75373
16	127	1.6795	8.79982
17	97	1.3853	8.55053
18	130	1.7144	8.65976
19	152	2	9.06635
27	119	1.5907	8.72131
28	123	1.6336	8.69536

$$\begin{aligned}
 \text{Weighted Accumulative number (10)} &= \text{Traffic level [10]} + (0.9 \times \text{Traffic level [9]}) \\
 &\quad + (0.8 \times \text{Traffic level [8]}) + (0.7 \times \text{Traffic level [7]}) \\
 &\quad + (0.6 \times \text{Traffic level [6]}) + (0.5 \times \text{Traffic level [5]}) \\
 &\quad + (0.4 \times \text{Traffic level [4]}) + (0.3 \times \text{Traffic level [3]}) \\
 &\quad + (0.2 \times \text{Traffic level [2]}) + (0.1 \times \text{Traffic level [1]}) \\
 &= 1.5595 + (0.9 \times 1.818) + (0.8 \times 1.832) \\
 &\quad + (0.7 \times 1.6795) + (0.6 \times 1.677) + (0.5 \times 1.377) \\
 &\quad + (0.4 \times 1.5696) + (0.3 \times 1.9042) + (0.2 \times 1.6229) \\
 &\quad + (0.1 \times 0.0004) = 9.04979
 \end{aligned}$$

This formula is used with SYN-Flood, UDP-Flood, Ping-of-death, FTP password guessing, and Telnet password guessing detector. In case of Email bomb detector, the number of incoming mails is accumulated every 3 minutes rather than 1 second.

#### Fuzzy rule setting up criteria

To set up the LEVEL BOX fuzzy rules, the rules are set to normalize the input variable. Hence, the heuristic rules are set based on the following knowledge.

1. If the traffic frequency is *low* then the level is “0.”
2. If the traffic frequency is *medium* then the level is “1.”
3. If the traffic frequency is *high* then the level is “2.”
4. If the traffic frequency is *very high* then the level is “3.”
5. If the traffic frequency is *extremely high* then the level is “4.”

Notwithstanding the number of rules in LEVEL BOX depends on types of detector because the characteristics of each attack are different. For instance, SYN LEVEL BOX fuzzy rules contain all these rules, while ICMP REPLY LEVEL BOX contains only 3 rules because number of abnormal packets (in one second) in SYN-Flood attack is very high while it is not so high in case of Ping-of-Death attack. Due to the experiences, the traffic frequency membership function (input variable) of each detector can be adjusted to yield appropriate result of traffic level then the LEVEL BOX can derive the most suitable traffic level as an output variable.

To set up the DETECTOR BOX fuzzy rules, the heuristic rules are set, based on the expert knowledge. The rules are also set based on two variables the number of traffic level in current second and the amount of traffic during past seconds. These two

parameters are kept in mind during rule-setting process. Due to the system administrators' experience, they figure out that the system has been flooded when the victim has continuously received high number of open-connection packets (in case of SYN-Flood attack) for long period of time. If the victim has been flooded just one or two second, they discover out that the victim can deal with these packets. According to this heuristic knowledge, the following heuristic rules can be extended and applied with SYN-FLOOD and UDP-FLOOD DETECTOR BOX.

1. If the current traffic level is *very high* and the past traffic levels are also *very high* then this event is considered to be *severe attack*.
2. If the current traffic level is *very high* and the past traffic levels are also *high* then this event is considered to be *severe attack*.
3. If the current traffic level is *very high* and the past traffic levels are also *medium* then this event is considered to be *attack*.
4. If the current traffic level is *very high* and the past traffic levels are also *low* then this event is considered to be *abnormal*.
5. If the current traffic level is *high* and the past traffic levels are also *very high* then this event is considered to be *severe attack*.
6. If the current traffic level is *high* and the past traffic levels are also *high* then this event is considered to be *attack*.
7. If the current traffic level is *high* and the past traffic levels are also *medium* then this event is considered to be *abnormal*.
8. If the current traffic level is *high* and the past traffic levels are also *low* then this event is considered to be *abnormal*.

9. If the current traffic level is *medium* and the past traffic levels are also *very high* then this event is considered to be *severe attack*.
10. If the current traffic level is *medium* and the past traffic levels are also *high* then this event is considered to be *attack*.
11. If the current traffic level is *medium* and the past traffic levels are also *medium* then this event is considered to be *abnormal*.
12. If the current traffic level is *medium* and the past traffic levels are also *low* then this event is considered to be *normal*.
13. If the current traffic level is *low* and the past traffic levels are also *very high* then this event is considered to be *severe attack*.
14. If the current traffic level is *low* and the past traffic levels are also *high* then this event is considered to be *abnormal*.
15. If the current traffic level is *low* and the past traffic levels is also *medium* then this event is considered to be *abnormal*.
16. If the current traffic level is *low* and the past traffic levels is also *low* then this event is considered to be *normal*.

In fuzzy rules of both, SYN-FLOOD and UDP-FLOOD DETECTOR BOX, the traffic level is the normalized input for each detector box. Therefore adjusting these rule boxes, the second input membership function, weighted accumulative number, is tuned according to experiences and types of attack to yield the correct detection result.

In case of Ping-of-Death detector, the heuristic rules can be derived from the expert knowledge. The rules are also set based on two variables the number of ICMP reply level in current second and the amount of ICMP reply fragments during past seconds.



Due to the system administrators' experience, they figure out that the system has been attacked when the victim has continuously transmitted high number of ICMP reply fragments (with the same fragment identification number) for long period of time. If the victim has just one or two ICMP reply fragment in one second, they discover out that this event is not an attack. According to this heuristic knowledge, the following heuristic rules can be extended and applied with Ping-of-Death DETECTOR BOX.

1. If number of ICMP reply fragment is low and number of past ICMP reply fragment is also low then this event is considered as normal.
2. If number of ICMP reply fragment is low and number of past ICMP reply fragment is medium then this event is also considered as normal.
3. If number of ICMP reply fragment is low and number of past ICMP reply fragment is high then this event is considered as abnormal.
4. If number of ICMP reply fragment is medium and number of past ICMP reply fragment is low then this event is considered as normal.
5. If number of ICMP reply fragment is medium and number of past ICMP reply fragment is also medium then this event is considered as abnormal.
6. If number of ICMP reply fragment is medium and number of past ICMP reply fragment is high then this event is considered as attack.
7. If number of ICMP reply fragment is high and number of past ICMP reply fragment is low then this event is considered as abnormal.
8. If number of ICMP reply fragment is high and number of past ICMP reply fragment is medium then this event is considered as abnormal.



9. If number of ICMP reply fragment is high and number of past ICMP reply fragment is also high then this event is considered as attack.

In this detector, the traffic level is also the normalized input for the detector box. Like other detectors, to adjust these rule boxes, the second input membership function, weighted accumulative number, is tuned according to experiences and types of attack to yield the correct detection result.

In case of MAIL DETECTOR BOX, the rule setting criteria is mentioned as following.

*“If the mail server has received the mail directed to the same recipient more than 50 mails in 30 minutes, then this event is considered to be mail bomb attack.”*

According to the criteria, the average of mail in very 3 minutes equal to 5 mails. For example if the mail server has gotten 5 mails every 3 minutes for 30 minutes, this means that the server has been bombed. Hence, the weighted accumulative number should be derived from mail levels in past 30 minutes. The rule setting criteria is also based on the attack in past 30 minutes and the current mail level with in 3 minutes. Like other DETECTOR BOXes, the possible events should be considered. The following shows the heuristic rules applied with MAIL DETECTOR BOX.

1. If the mail level in 3 minutes is *very low* and the number of mails during past 30 minutes is also *low* then this event is *normal*.

2. If the mail level in 3 minutes is *very low* and the number of mails during past 30 minutes is also *medium* then this event is *normal*.
3. If the mail level in 3 minutes is *very low* and the number of mails during past 30 minutes is also *high* then this event is *abnormal*.
4. If the mail level in 3 minutes is *very low* and the number of mails during past 30 minutes is also *very high* then this event is *attack*.
5. If the mail level in 3 minutes is *low* and the number of mails during past 30 minutes is also *low* then this event is *normal*.
6. If the mail level in 3 minutes is *low* and the number of mails during past 30 minutes is also *medium* then this event is *abnormal*.
7. If the mail level in 3 minutes is *low* and the number of mails during past 30 minutes is also *high* then this event is *attack*.
8. If the mail level in 3 minutes is *low* and the number of mails during past 30 minutes is also *very high* then this event is *severe attack*.
9. If the mail level in 3 minutes is *medium* and the number of mails during past 30 minutes is also *low* then this event is *abnormal*.
10. If the mail level in 3 minutes is *medium* and the number of mails during past 30 minutes is also *medium* then this event is *attack*.
11. If the mail level in 3 minutes is *medium* and the number of mails during past 30 minutes is also *high* then this event is *attack*.
12. If the mail level in 3 minutes is *medium* and the number of mails during past 30 minutes is also *very high* then this event is *severe attack*.
13. If the mail level in 3 minutes is *high* and the number of mails during past 30 minutes is also *low* then this event is *attack*.

14. If the mail level in 3 minutes is *high* and the number of mails during past 30 minutes is also *medium* then this event is *attack*.
15. If the mail level in 3 minutes is *high* and the number of mails during past 30 minutes is also *high* then this event is *severe attack*.
16. If the mail level in 3 minutes is *high* and the number of mails during past 30 minutes is also *very high* then this event is *severe attack*.
17. If the mail level in 3 minutes is *very high* and the number of mails during past 30 minutes is also *low* then this event is *attack*.
18. If the mail level in 3 minutes is *very high* and the number of mails during past 30 minutes is also *medium* then this event is *severe attack*.
19. If the mail level in 3 minutes is *very high* and the number of mails during past 30 minutes is also *high* then this event is *severe attack*.
20. If the mail level in 3 minutes is *very high* and the number of mails during past 30 minutes is also *very high* then this event is *severe attack*.
21. If the mail level in 3 minutes is *extremely high* then this event is *severe attack*.

Like other detector boxes, the mail level is the normalized input for the detector box. Therefore adjusting these rule boxes, the second input membership function, weighted accumulative number, is tuned according to experiences to yield the correct detection result.

In case of Password guessing detector, the heuristic rules are also derived from the expert knowledge. The rules are also set based on two variables, the number of packet containing *login incorrect* as data in current second and the amount of packet containing *login incorrect* during past seconds. Due to the system administrators'

experience, they figure out that the system has been password-guessed when the victim has continuously received high number of packet containing *login incorrect* as data for long period of time. If the victim has been guessed just one or two times within several seconds, they discover that this event is not an attack. According to this heuristic knowledge, the following heuristic rules can be extended and applied with Password guessing DETECTOR BOX.

1. If the number of current password guessing attempts is low and the number of past password guessing attempts is also low then this event is considered as normal.
2. If the number of current password guessing attempts is low and the number of past password guessing attempts is medium then this event is considered as abnormal.
3. If the number of current password guessing attempts is low and the number of past password guessing attempts is high then this event is considered as attack.
4. If the number of current password guessing attempts is medium and the number of past password guessing attempts is low then this event is considered as abnormal.
5. If the number of current password guessing attempts is medium and the number of past password guessing attempts is also medium then this event is considered as abnormal.
6. If the number of current password guessing attempts is medium and the number of past password guessing attempts is high then this event is considered as attack.
7. If the number of current password guessing attempts is high and the number of past password guessing attempts is low then this event is considered as abnormal.
8. If the number of current password guessing attempts is high and the number of past password guessing attempts is medium then this event is considered as abnormal.

9. If the number of current password guessing attempts is high and the number of past password guessing attempts is also high then this event is considered as attack.

Like other detector boxes, the first variable, current password guessing attempts, is the normalized input for the detector box. Therefore adjusting these rule boxes, the second input membership function, weighted accumulative number, is tuned according to experiences to yield the correct detection result.

In case of Port scanning detector, the heuristic rules are also derived from the expert knowledge. Unlike others, there is just one variable, SYN-RESET pair in each second, used to determine the attack possibility, because the hacker can guess the victim for the password at anytime. Due to the system administrators' experience, they figure out that the system has been port-scanned when the victim has received high number of open-connection packets (SYN packets) and the victim also responds these SYN packets by transmitting RESET packets. According to this heuristic knowledge, the following heuristic rules can be extended and applied with Port scanning DETECTOR BOX.

Therefore the heuristic rules of PORT SCAN DETECTOR BOX can be derived as shown follow.

1. If the number of scanned ports is low then this event is considered as normal.
2. If the number of scanned ports is medium then this event is considered as abnormal.

3. If the number of scanned port is high then this event is considered as attack.

According to these heuristic rules, the variable, scanned port, should be tuned according to experiences to yield the correct detection result.

#### Defuzzification Method

Both fuzzy rules boxes, LEVEL BOX and DETECTOR BOX, employ *Centroid* as a defuzzification method. Because using *Centroid*, the FIDS, that employs fuzzy sets and fuzzy rule-based system, can determine all characteristics of attacks including the hidden attack's characteristic, that tries to hide itself from threshold-based detection. If the other defuzzification methods are employed rather than *Centroid* then the hidden attack's characteristic cannot be discovered. For instance, the traffic level gradually increases when the traffic frequency gradually increases and the traffic level also gradually decreases when the traffic frequency gradually decreases when using *Centroid*. Moreover using *Centroid*, the DETECTOR BOX can give the continuous detection result ranged from 0 to 100. For instance, if there is no any attacking or intrusive traffic, the FIDS detection result is almost "0" when using *Centroid* as defuzzification method. Therefore employing *Centroid* as the defuzzification method, FIDS yields the most effective and reasonable detection results.

#### SYN-Flood Detector

After the Filter and Parser Module (FPM) filter network traffic by comparing the captured network traffic, if any packet matches with the SYN-Flood signature then only SYN packets are maintained and collected. FPM processes the filtered SYN



traffic by counting the number of SYN packets occurrence directed to the same destination host and same destination port every second. Here is the list of parameters collected by FPM and example of filtered information is shown in Table 2.

- Time stamp
- Destination host (destination IP number)
- Destination service (destination port)

Table 2: Example of preprocessed SYN-packet data within 1<sup>st</sup> second

Timestamp	Flag	Dst_host	Dst_service	Frequency
1	S	Victim1	80	2510
1	S	Victim2	80	1999
1	S	Victim3	23	2
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
1	S	Victim <i>n</i>	80	1

Actually, a SYN packet contains more details than the considered parameters such as source IP number, source port, packet sequence number and etc [12]. However, some information are not necessary. For example source IP number and source port cannot be used as considered parameters when counting the frequency because the source IP number (return IP number) can be faked.

Then at the end of every second, FPM forwards the SYN-packets frequencies (amount of packet in one second) to SYN-Flood Detector.

The SYN-Flood Detector consists of two fuzzy boxes as shown in Figure 8. The first box is SYN LEVEL BOX used to normalize the SYN packet frequency to become the SYN traffic level.



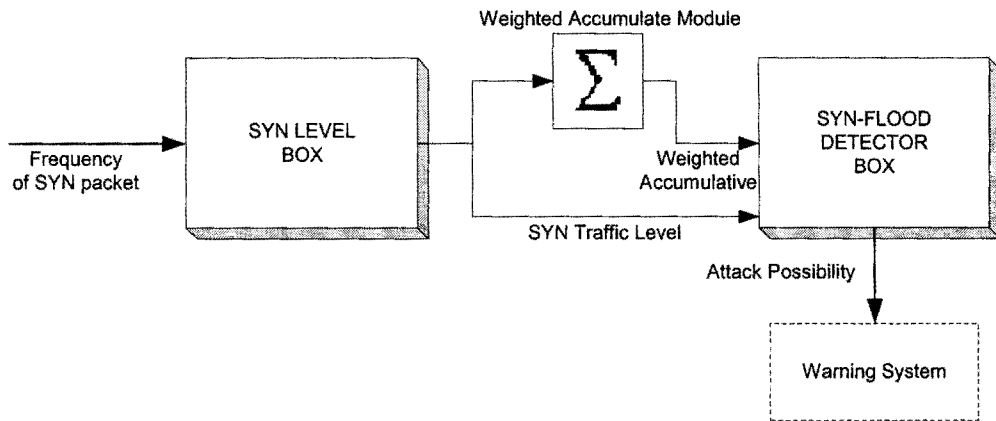


Figure 8: The SYN-Flood detector framework

### SYN LEVEL BOX

The SYN LEVEL BOX input variable is the SYN packet frequency in specific time period (current second). This input variable comprises of five fuzzy sets (low, medium, high, very high, and extremely high) as shown in Figure 9.

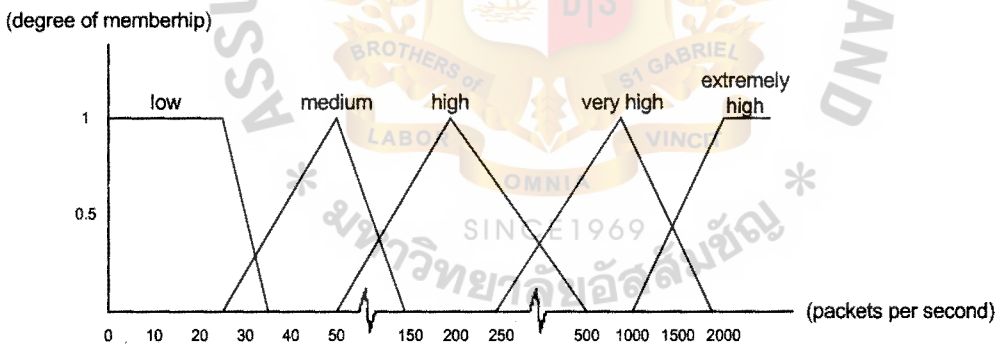


Figure 9: The SYN packet membership function

### **SYN packet frequency membership function**

- *Low* is the set of low number of SYN packets per second on live traffic (This indicates normal SYN packet traffic).

- *Medium* is the set of medium number of SYN packets per second (This might contain low degree of abnormal SYN packets).
- *High* is the set of high number of SYN packets per second (This might contain high degree of abnormal SYN packets).
- *Very high* is the set of very high number of SYN packets per second (This might contain the attack SYN packets).
- *Extremely high* is the set of extremely high number of SYN packets per second (This might contain the severe attack traffic).

The SYN LEVEL BOX output variable is the SYN traffic level. The SYN traffic level also comprises of five fuzzy sets (0, 1, 2, 3, and 4) as show figure below.

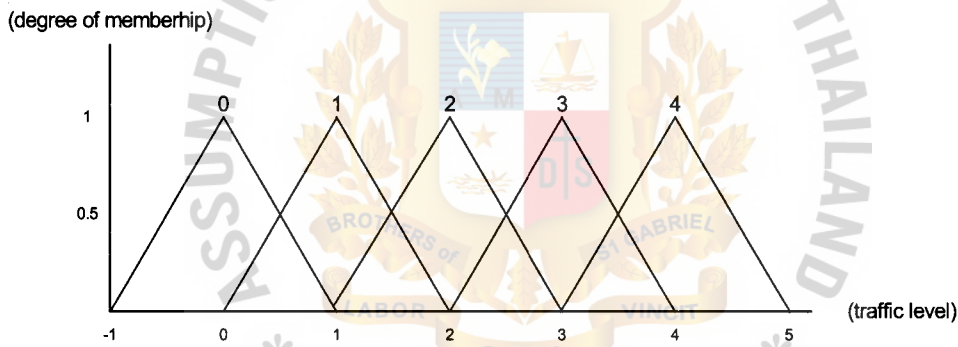


Figure 10: The SYN traffic level membership function

### SYN traffic level membership function

These sets of SYN traffic level are set of normalized SYN packet frequency. Therefore, the SYN traffic level also comprises of five fuzzy sets (level 0, 1, 2, 3, and level 4) as mentioned above, shown in Figure 10.

### SYN LEVEL BOX fuzzy rules

These fuzzy rules are used to normalize the input value, SYN packet frequency, to become SYN traffic level.

1. If (SYN packet frequency is low) then (SYN traffic level is 0)
2. If (SYN packet frequency is medium) then (SYN traffic level is 1)
3. If (SYN packet frequency is high) then (SYN traffic level is 2)
4. If (SYN packet frequency is very high) then (SYN traffic level is 3)
5. If (SYN packet frequency is extremely high) then (SYN traffic level is 4)

Thereafter the frequencies are forwarded to the SYN LEVEL BOX, this box produces the SYN traffic level of each individual input according to the defined fuzzy rules. Only the SYN traffic levels may not enough for a decision making of attack behavior. Therefore the *weighted accumulative number* is also taken into account. The SYN traffic level passes to the additional module called *Weighted Accumulate Module*. Weighted Accumulate Module is used to determine the weighted accumulative number of SYN packets in past seconds according to the defined formula, mentioned above.

Now, there are two new variables produced by SYN LEVEL BOX and Weighted Accumulate Module, SYN traffic level and weighted accumulative number. The example is shown in Table 3. These two values are used as input variables of second fuzzy box, SYN-FLOOD DETECTOR BOX.

Table 3: Example of SYN traffic level and weighted accumulative number  
in 10<sup>th</sup> second

Time stamp	Flag	Dst_host	Dst_service	Frequency	SYN level	Weighted accumulative number
10	S	Victim1	80	2500	4	21.78
10	S	Victim2	80	1750	3.91	17.36
10	S	Victim3	23	0	0.0004	0.00408
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
10	S	Victim <i>n</i>	80	1	0.0004	0.00408

SYN-FLOOD DETECTOR BOX receives both SYN traffic level and weighted accumulative number, and then produces the *attack possibility* based on the fuzzy rules. Attack possibility defines the severity of attack ranged between 0 to 100. Very high attack possibility indicates that the victim is being attack severely.

#### Attack Possibility States

The attack possibility range of 0 to 100 is divided into five severity states.

Attack Possibility	Severity State
0	Normal
20	Abnormal
40	Warning
60	Attack
80	Critical attack
100	Critical attack

#### SYN-FLOOD DETECTOR BOX

The SYN-FLOOD DETECTOR BOX input variables are the SYN traffic level (normalized SYN packet frequency) in current time period and the weighted accumulative number. As mentioned above, the first input variable is derived from SYN LEVEL BOX therefore this variable composes of four fuzzy sets, 1, 2, 3, and 4, as shown in Figure 11. The second input variable, weighted accumulative number,

also composes of four fuzzy sets (low, medium, high, and very high) as shown Figure 12.

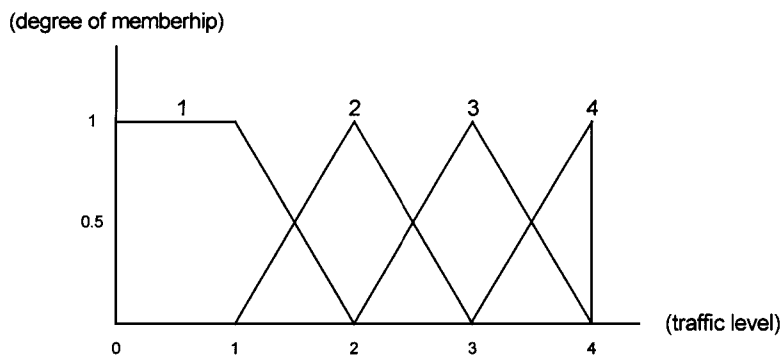


Figure 11: The SYN traffic level membership function of  
SYN-FLOOD DETECTOR BOX

**SYN traffic level membership function**

The set of “0” is missing from the SYN traffic level membership function of SYN-FLOOD DETECTOR BOX because this set is combined with the set of “1” and these two sets yield the same detection results.

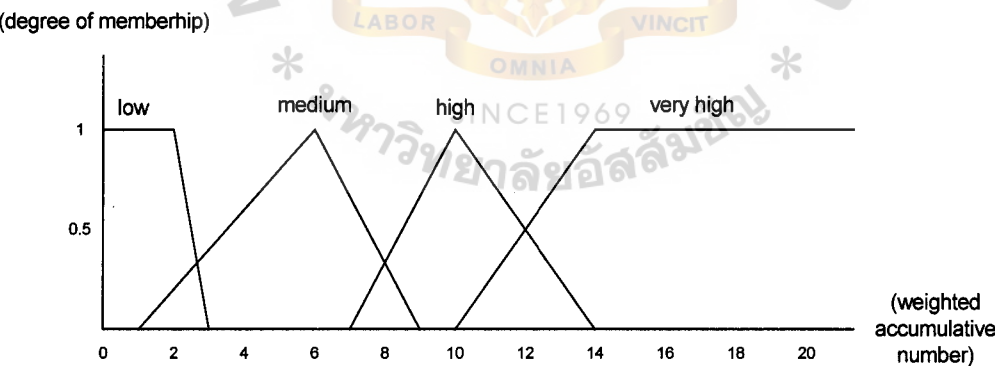


Figure 12: The weighted accumulative number membership function of  
SYN-FLOOD DETECTOR BOX

### Weighted accumulative number membership function

- *Low* is the set of low number of SYN packets in past seconds (This indicates that there may not be the attack during these seconds).
- *Medium* is the set of medium number of SYN packets in past seconds (This indicates that there may be the attack in few past seconds).
- *High* is the set of high number of SYN packets in past seconds (This indicates that there may be the attack in many past seconds).
- *Very high* is the set of very high number of SYN packets in past seconds (This indicates that there may be the attack almost every past seconds).

The output of SYN DETECTOR BOX is the attack possibility. The attack possibility composes of four fuzzy sets (normal, abnormal, attack, and server attack). Its' membership function are shown in Figure 13.

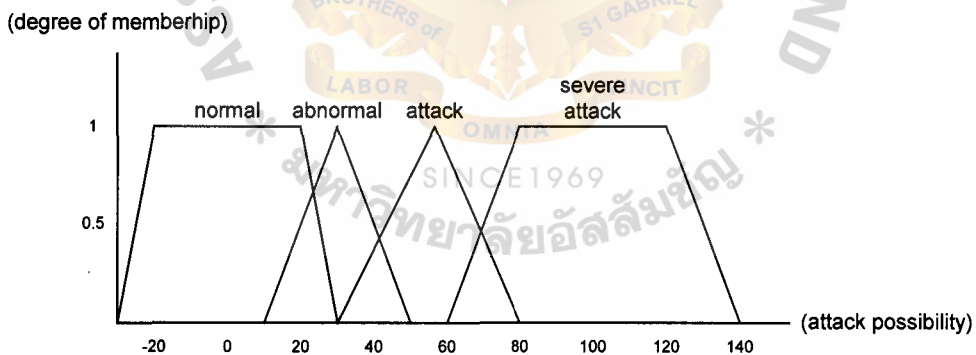


Figure 13: The attack possibility membership function of SYN-FLOOD DETECTOR

### Attack Possibility membership function

- *Normal* is the set of low attack possibility in current second (This indicates that there is a normal to abnormal state).
- *Abnormal* is the set of medium attack possibility in current second (This indicates that there is an abnormal to warning state).
- *Attack* is the set of high attack possibility in current second (This indicates that there is a warning to attack state).
- *Severe attack* is the set of very high attack possibility in current second (This indicates that there is an attack to critical attack state).

### SYN-FLOOD DETECTOR BOX fuzzy rules

The following fuzzy rules are derived from the heuristic rules, that are set based on the expert knowledge, as explained above in *Fuzzy rule setting up criteria* section. These rules are employed to figure out the attack possibility.

1. If (SYN traffic level is 4) and (Weighted accumulative number is very high)  
then (Attack possibility is severe attack)
2. If (SYN traffic level is 4) and (Weighted accumulative number is high)  
then (Attack possibility is severe attack)
3. If (SYN traffic level is 4) and (Weighted accumulative number is medium)  
then (Attack possibility is attack)
4. If (SYN traffic level is 4) and (Weighted accumulative number is low)  
then (Attack possibility is abnormal)
5. If (SYN traffic level is 3) and (Weighted accumulative number is very high)  
then (Attack possibility is severe attack)



6. If (SYN traffic level is 3) and (Weighted accumulative number is high)  
then (Attack possibility is attack)
7. If (SYN traffic level is 3) and (Weighted accumulative number is medium)  
then (Attack possibility is abnormal)
8. If (SYN traffic level is 3) and (Weighted accumulative number is low)  
then (Attack possibility is abnormal)
9. If (SYN traffic level is 2) and (Weighted accumulative number is very high)  
then (Attack possibility is sever attack)
10. If (SYN traffic level is 2) and (Weighted accumulative number is high)  
then (Attack possibility is attack)
11. If (SYN traffic level is 2) and (Weighted accumulative number is medium)  
then (Attack possibility is abnormal)
12. If (SYN traffic level is 2) and (Weighted accumulative number is low)  
then (Attack possibility is normal)
13. If (SYN traffic level is 1) and (Weighted accumulative number is very high)  
then (Attack possibility is severe attack)
14. If (SYN traffic level is 1) and (Weighted accumulative number is high)  
then (Attack possibility is abnormal)
15. If (SYN traffic level is 1) and (Weighted accumulative number is medium)  
then (Attack possibility is abnormal)
16. If (SYN traffic level is 1) and (Weighted accumulative number is low)  
then (Attack possibility is normal)

**UDP-Flood Detector**

In this section, the mechanism used to detect UDP-Flood attack is discussed, which is almost similar to SYN-Flood detector's mechanism. Start with, the network traffic is captured according to the defined UDP-Flood signature. Then UDP packets are only maintained and collected. The FPM processes the filtered UDP packets by counting the number of UDP packet occurrence directed to the same destination IP number within every second. Here is the list of parameters kept by FPM and example of preprocessed data shown in Table 4.

- Time stamp
- Destination host (destination IP number)

Table 4: Example of preprocessed UDP packet in 1<sup>st</sup> second

Timestamp	Dst_host	Frequency
1	Victim1	20
1	Victim2	2900
1	Victim3	10
.	.	.
.	.	.
1	Victim <i>n</i>	5

According to data in the above table, all UDP packet directed to the same victim are accumulated together even though the destination ports vary. The reason is that the attacker can attack the victim by flooding UDP packets to either specific destination port or random port. Like SYN-Flood attack, the source IP number is not collected, because the source IP number can be faked. At the end of every second, FPM forwards the frequencies of UDP packets to UDP-Flood detector.

The UDP-Flood detector also employs two fuzzy boxes as shown in Figure 14. First box is UDP LEVEL BOX, used to normalize the UDP packet frequency.

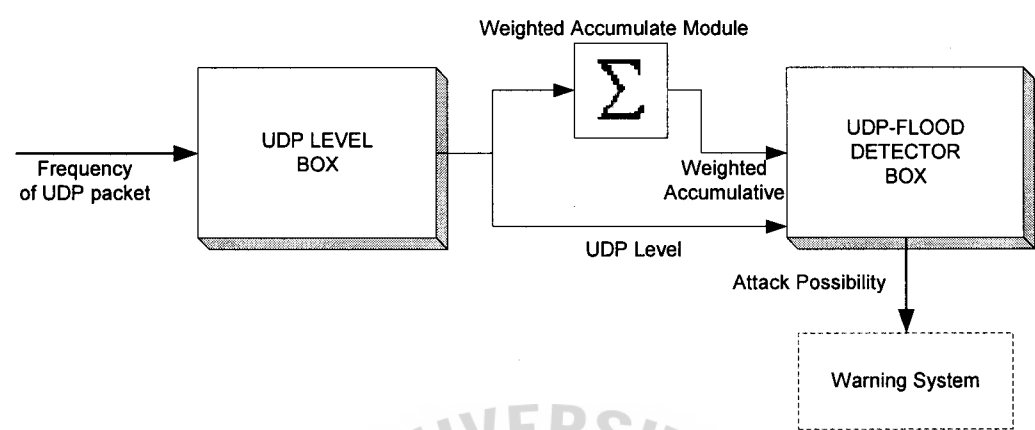


Figure 14: The UDP-Flood detector framework

UDP LEVEL BOX

Like SYN LEVEL BOX, the UDP packet frequency and UDP traffic level also compose of five fuzzy sets as shown in Figure 15 and 16.

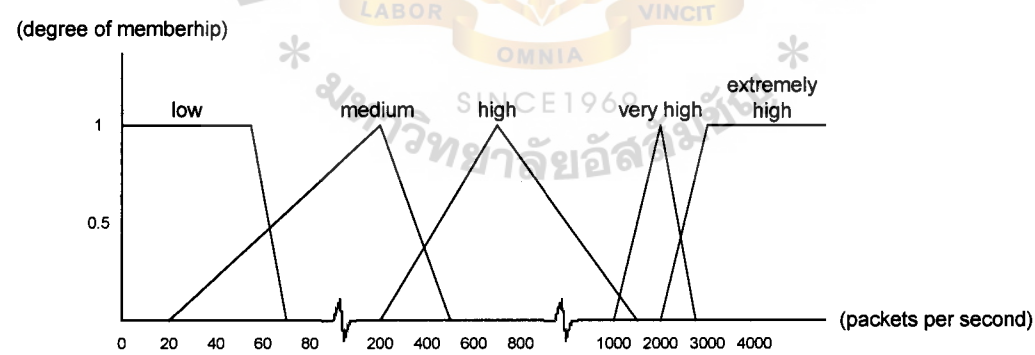


Figure 15: The membership function of UDP packet frequency

### UDP packet frequency membership function

- *Low* is the set of low number of UDP packets per second on live traffic (This indicates normal UDP packet traffic).
- *Medium* is the set of medium number of UDP packets per second (This might contain low degree of abnormal UDP packets).
- *High* is the set of high number of UDP packets per second (This might contain high degree of abnormal UDP packets).
- *Very high* is the set of very high number of UDP packets per second (This might contain the attack UDP packets).
- *Extremely high* is the set of extremely high number of UDP packets per second (This might contain the severe attack traffic).

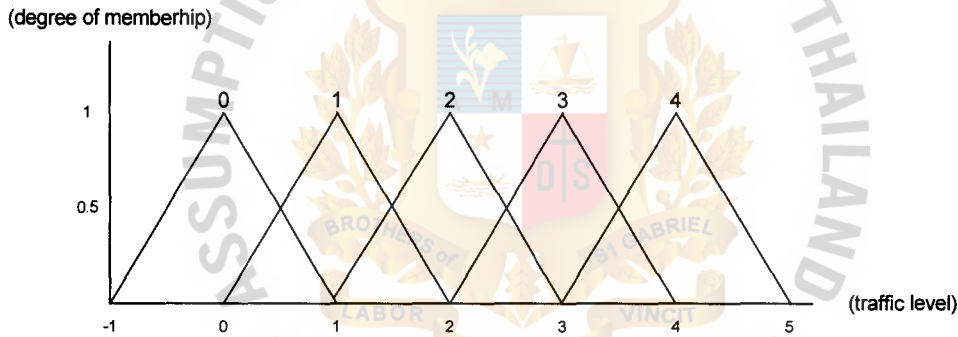


Figure 16: The membership function of UDP traffic level

### UDP traffic level membership function

Like SYN traffic level, UDP traffic level are set of normalized UDP packet frequency. Therefore, the UDP traffic level also comprises of five fuzzy sets (level 0, 1, 2, 3, and level 4) as mentioned above, shown in Figure 16.

### UDP LEVEL BOX fuzzy rule

These rules are used normalize the UDP packet frequency to become UDP traffic level.

1. If (UDP packet frequency is low) then (UDP traffic level is 0)
2. If (UDP packet frequency is medium) then (UDP traffic level is 1)
3. If (UDP packet frequency is high) then (UDP traffic level is 2)
4. If (UDP packet frequency is very high) then (UDP traffic level is 3)
5. If (UDP packet frequency is extremely high) then (UDP traffic level is 4)

After the frequency is forwarded to the UDP LEVEL BOX, this box produces the UDP traffic level of each individual input according to the defined fuzzy rules. Like the SYN-Flood detector, not only the UDP traffic level is employed but weighted accumulative number is also used by UDP-FLOOD DETECTOR BOX. The UDP traffic level passes Weighted Accumulate Module. Then the Weighted Accumulate Module determines the weighted accumulative number of the abnormal behavior during past seconds. The following table shows the example of data after the frequencies of UDP packet pass UDP LEVEL BOX and weighted accumulate module. Therefore two new values are used as input variables of second fuzzy box, UDP-FLOOD DETECTOR BOX, as shown in Table 5.

Next, UDP-FLOOD DETECTOR BOX receives both UDP traffic level and weighted accumulative number, and produces the attack possibility based on the defined fuzzy rules.

Table 5: Example of SYN traffic level and weighted accumulative number  
in 10<sup>th</sup> second

Time stamp	Dst_host	Frequency	UDP level	Weighted accumulative number
10	Victim1	20	0.000408	0.00408
10	Victim2	2900	4	20.74
10	Victim3	10	0.000408	0.00408
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
10	Victim <i>n</i>	1200	2.35	27.5

UDP-FLOOD DETECTOR BOX

The two input variables are the UDP traffic level and weighted accumulative number as shown in Figure 17 and 18.

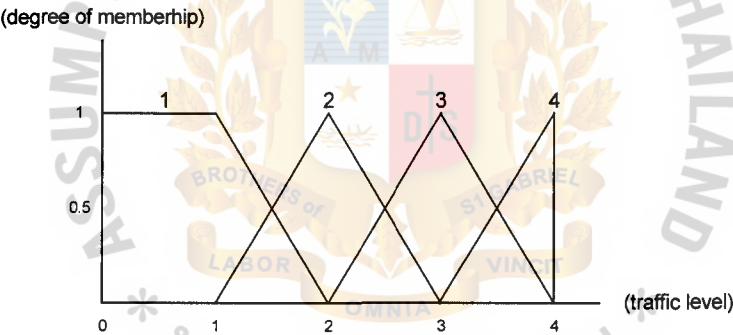


Figure 17: The UDP traffic level membership function of  
UDP-FLOOD DETECTOR BOX

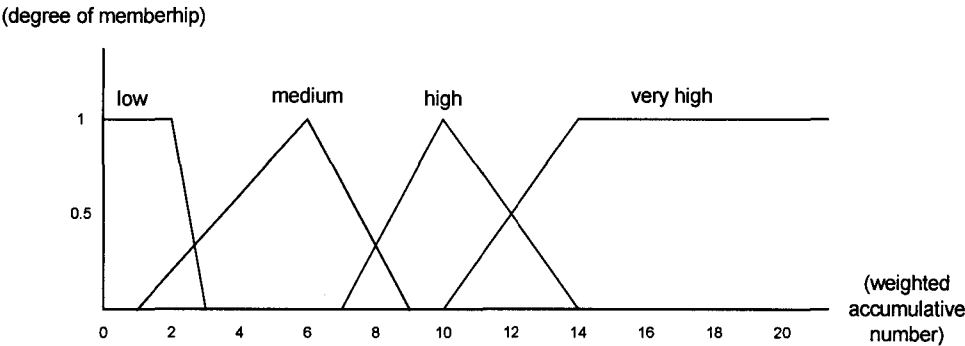


Figure 18: The weighted accumulative number of UDP-FLOOD DETECTOR BOX

The output of UDP-FLOOD DETECTOR BOX is the attack possibility. The attack possibility composes of four fuzzy sets (normal, abnormal, attack, and server attack). Its' membership function are shown in Figure 19.

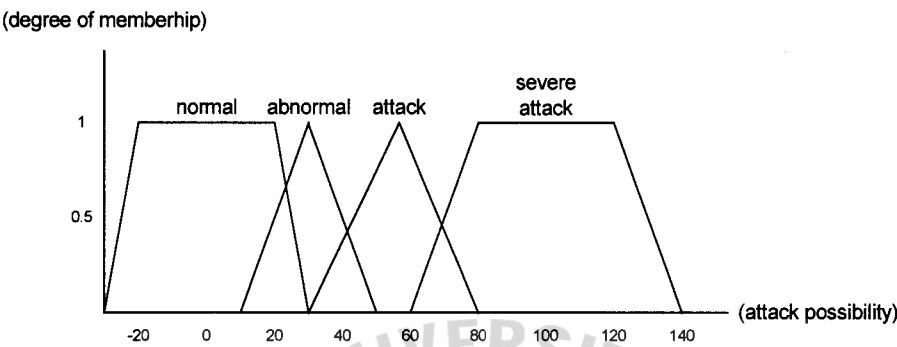


Figure 19: The attack possibility membership function of UDP-FLOOD DETECTOR

**UDP-FLOOD DETECTOR BOX fuzzy rules**

The following fuzzy rules are derived form the heuristic rules, that are set based on the expert knowledge, as explained above in *Fuzzy rule setting up criteria* section. These rules are employed to figure out the attack possibility.

- 1. If (UDP traffic level is 4) and (Weighted accumulative number is very high)  
then (Attack possibility is severe attack)
- 2. If (UDP traffic level is 4) and (Weighted accumulative number is high)  
then (Attack possibility is severe attack)
- 3. If (UDP traffic level is 4) and (Weighted accumulative number is medium)  
then (Attack possibility is attack)
- 4. If (UDP traffic level is 4) and (Weighted accumulative number is low)  
then (Attack possibility is abnormal)



5. If (UDP traffic level is 3) and (Weighted accumulative number is very high)  
then (Attack possibility is severe attack)
6. If (UDP traffic level is 3) and (Weighted accumulative number is high)  
then (Attack possibility is attack)
7. If (UDP traffic level is 3) and (Weighted accumulative number is medium)  
then (Attack possibility is abnormal)
8. If (UDP traffic level is 3) and (Weighted accumulative number is low)  
then (Attack possibility is abnormal)
9. If (UDP traffic level is 2) and (Weighted accumulative number is very high)  
then (Attack possibility is attack)
10. If (UDP traffic level is 2) and (Weighted accumulative number is high)  
then (Attack possibility is attack)
11. If (UDP traffic level is 2) and (Weighted accumulative number is medium)  
then (Attack possibility is abnormal)
12. If (UDP traffic level is 2) and (Weighted accumulative number is low)  
then (Attack possibility is normal)
13. If (UDP traffic level is 1) and (Weighted accumulative number is very high)  
then (Attack possibility is severe attack)
14. If (UDP traffic level is 1) and (Weighted accumulative number is high)  
then (Attack possibility is abnormal)
15. If (UDP traffic level is 1) and (Weighted accumulative number is medium)  
then (Attack possibility is abnormal)
16. If (UDP traffic level is 1) and (Weighted accumulative number is low)  
then (Attack possibility is normal)

**Ping-of-Death Detector**

In this section, the mechanism used to detect Ping-of-Death attack is discussed. The network traffics are captured according to the predefined Ping of Death signature. Then only ICMP reply fragments (with the same fragment identification number) are maintained and collected. The FPM processes these fragments by counting the number of occurrence, in which the fragments are sent from the same source IP number in every second. Here is the list of parameters kept by FPM and example of preprocessed data shown in Table 6.

- Time stamp
- Source host (source IP address)
- Fragment identification number

The detector detects by considering the number of ICMP reply fragments. Because any hosts send a lot of ICMP reply fragments mean that those hosts may vulnerable to Ping-of-Death attack. Therefore ICMP REPLY LEVEL BOX uses the ICMP reply fragment frequency to determine the normalized number, the ICMP reply fragment level.

Table 6: Example of preprocessed ICMP reply fragment in 1<sup>st</sup> second

Timestamp	Scr_host	Fragment identification number	Frequency
1	Host1	1234	150
1	Host 2	4872	2
1	Host 3	1475	1
.	.	.	.
.	.	.	.
.	.	.	.
1	Host <i>n</i>	1245	1

Then, the ICMP reply fragment level, got from first fuzzy box, is used to determine the weighted accumulative number during past seconds by Weighted Accumulate Module. Consequently, the attack possibility can be derived by ICMP REPLY DETECTOR BOX. ICMP REPLY DETECT BOX, uses both the ICMP reply level and the weighted accumulative number as input variables. The framework of Ping-of-Death detector is shown in Figure 20.

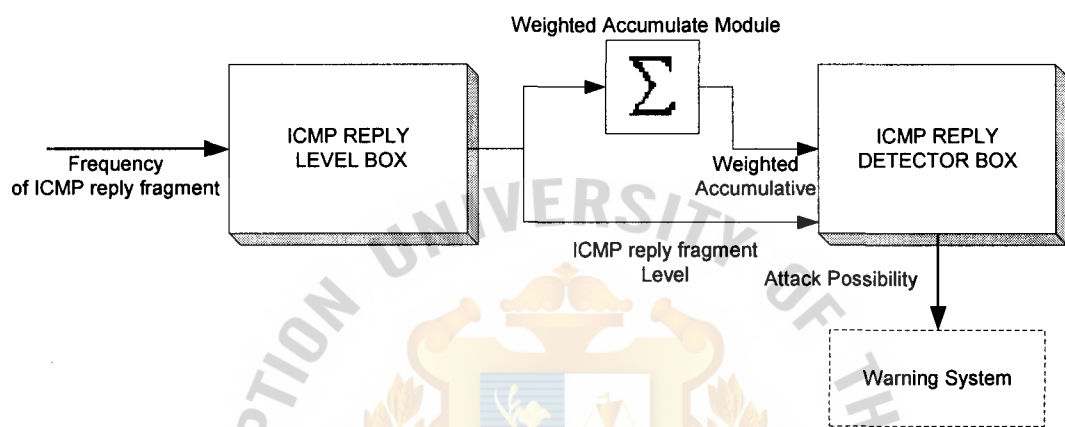


Figure 20: The Ping-of-Death detector framework

### ICMP REPLY LEVEL BOX

The input variable of this fuzzy box is ICMP reply fragment frequency. This variable comprises of three fuzzy sets (low, medium, and high) as shown in Figure 21.

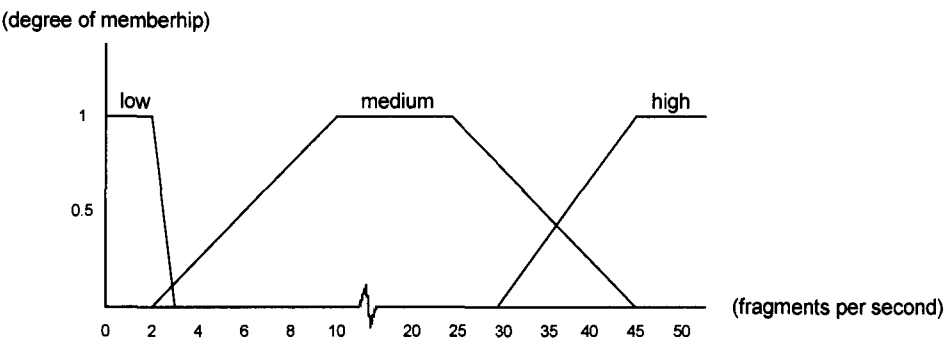


Figure 21: The membership function of ICMP reply fragment frequency

**ICMP reply fragment frequency membership function**

- *Low* is the set of normal number of ICMP reply fragments per second (This indicates the normal amount of ICMP reply fragments on the live traffic).
- *Medium* is the set of medium number of ICMP reply fragments per second.
- *High* is the set of high number of ICMP reply fragments per second (This indicated the abnormal amount of ICMP reply fragments).

The ICMP REPLY LEVEL BOX output variable, ICMP reply level, also comprises of three fuzzy sets (0, 1, and 2) and show in Figure 22.

**ICMP reply level membership function**

These sets of ICMP reply level are set of normalized ICMP reply fragment frequency. Therefore, the ICMP reply level also comprises of three fuzzy sets (level 0, 1, and level 2) as mentioned above, shown in figure below.

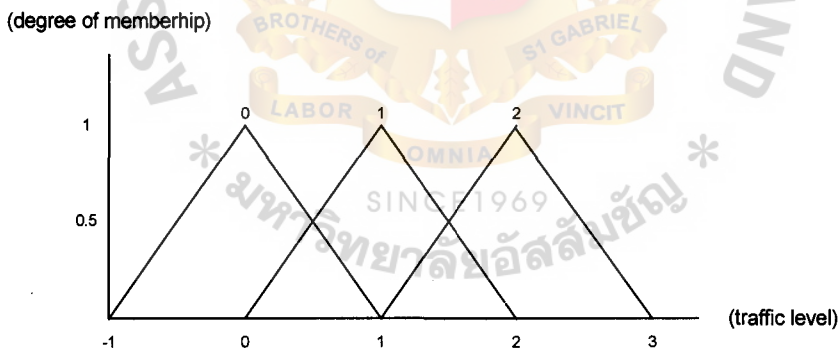


Figure 22: The membership function of ICMP reply level

**ICMP REPLY LEVEL BOX fuzzy rules**

These rules are used to normalize the ICMP reply fragment frequency to become the ICMP reply fragment level.

- 1. If (ICMP reply fragment frequency is low)  
then (the ICMP reply fragment level is 0)
- 2. If (ICMP reply fragment frequency is medium)  
then (the ICMP reply fragment level is 1)
- 3. If (ICMP reply fragment frequency is high)  
then (the ICMP reply fragment level is 2)

ICMP REPLY DETECTOR BOX

The two input variables are the ICMP reply fragment level and the weighted accumulative number as shown in Figure 23 and 24.

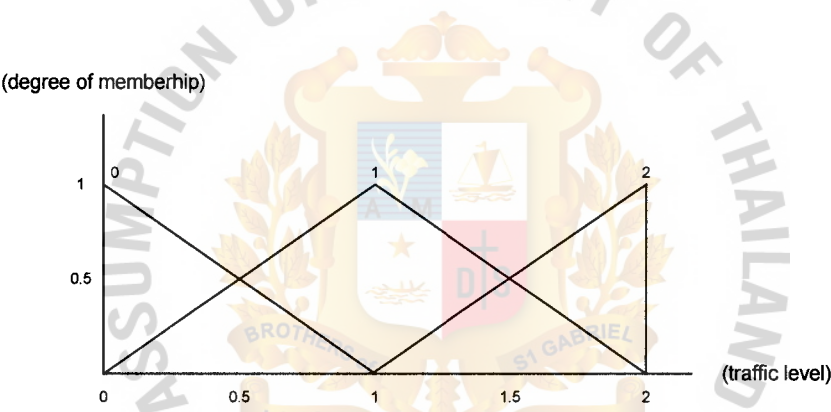


Figure 23: The ICMP reply level membership function of  
ICMP REPLY DETECTOR BOX

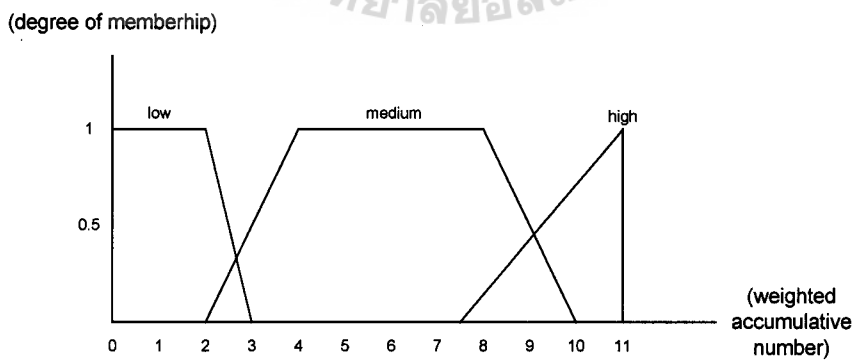


Figure 24: The weighted accumulative number of ICMP REPLY DETECTOR BOX

**Weighted accumulative number membership function**

- *Low* is the set of low number of ICMP reply fragments in past seconds (This indicates that there may not be the attack during these seconds).
- *Medium* is the set of medium number of ICMP reply fragments in past seconds (This indicates that there may be the attack in few past seconds).
- *High* is the set of high number of ICMP reply fragments in past seconds (This indicates that there may be the attack in many past seconds).

The output of ICMP REPLY DETECTOR BOX is the attack possibility. The attack possibility composes of three fuzzy sets (normal, abnormal, and attack). Its membership function are shown in Figure 25.

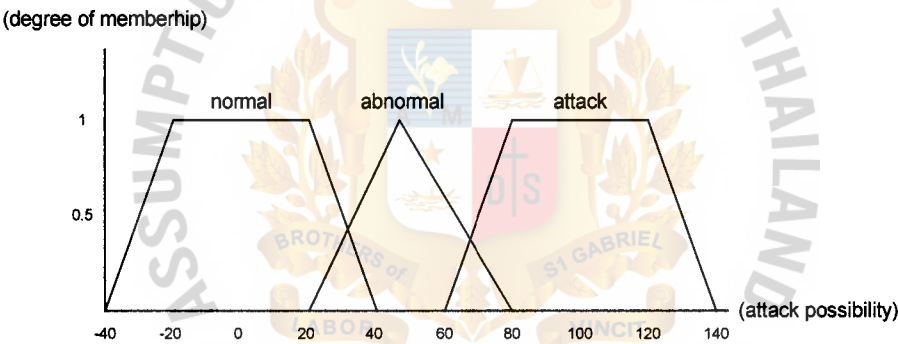


Figure 25: The attack possibility membership function of  
ICMP REPLY DETECTOR BOX

**Attack Possibility membership function**

- *Normal* is the set of low attack possibility in current second (This indicates that there is a normal to abnormal state).
- *Abnormal* is the set of medium attack possibility in current second (This indicates that there is an abnormal, warning to attack state).

- *Attack* is the set of very high attack possibility in current second (This indicates that there is a attack to critical attack state).

### ICMP REPLY DETECTOR BOX fuzzy rules

The following fuzzy rules are derived form the heuristic rules, that are set based on the expert knowledge, as explained above in *Fuzzy rule setting up criteria* section.

These rules are employed to figure out the attack possibility.

1. If (ICMP reply level is 0) and (Weighted accumulative number is low)  
then (Attack possibility is normal)
2. If (ICMP reply level is 0) and (Weighted accumulative number is medium)  
then (Attack possibility is normal)
3. If (ICMP reply level is 0) and (Weighted accumulative number is high)  
then (Attack possibility is abnormal)
4. If (ICMP reply level is 1) and (Weighted accumulative number is low)  
then (Attack possibility is normal)
5. If (ICMP reply level is 1) and (Weighted accumulative number is medium)  
then (Attack possibility is abnormal)
6. If (ICMP reply level is 1) and (Weighted accumulative number is high)  
then (Attack possibility is attack)
7. If (ICMP reply level is 2) and (Weighted accumulative number is low)  
then (Attack possibility is abnormal)
8. If (ICMP reply level is 2) and (Weighted accumulative number is medium)  
then (Attack possibility is abnormal)
9. If (ICMP reply level is 2) and (Weighted accumulative number is high)  
then (Attack possibility is attack)



**Email Bomb Detector**

In this section discusses the mechanism used to detect Email Bomb attack. First, the FPM captures network traffic according to the predefined Email Bomb signature. FPM counts for the number of mail directed to the same recipient from the same source IP number in every 3 minutes. Here is the list of parameters kept by FPM and example of preprocessed data shown in Table 7.

- Source IP number (sending mail server)
- Destination IP number (receiving mail server)
- Recipient email address

Table 7: Example of preprocessed incoming emails in 3 seconds

Scr_host	Dst_host	Recipient email address	Frequency
Host1	Victim 1	AAA	2
Host 2	Victim 1	BBB	50
Host 3	Victim 1	CCC	100
.	.	.	.
.	.	.	.
.	.	.	.
Host n	Victim 1	ZZZ	* 1

The Email Bomb detector framework is shown in Figure 26. The mail frequency is normalized to become mail level by MAIL LEVEL BOX. Unlike other detectors, the mail levels are used to figure out the weighted accumulative number during past 30 minutes. Consequently, the attack possibility can be derived by MAIL DETECTOR BOX. MAIL DETECTOR BOX, uses both email level and the weighted accumulative number as input variables for making decision.

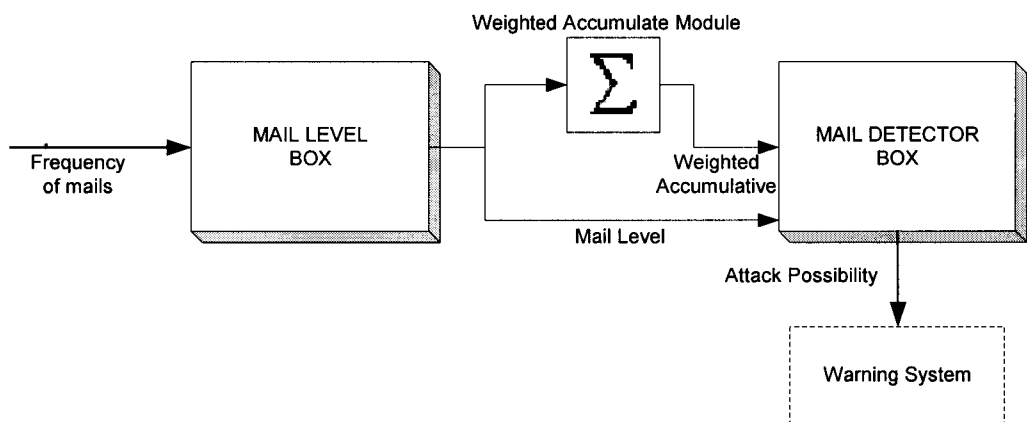


Figure 26: The email bomb detector framework

### MAIL LEVEL BOX

The input variable of this fuzzy box is mail frequency in every 3 minutes. This variable comprises of six fuzzy sets (very low, low, medium, high, very high, and extremely high) as shown in Figure 27.

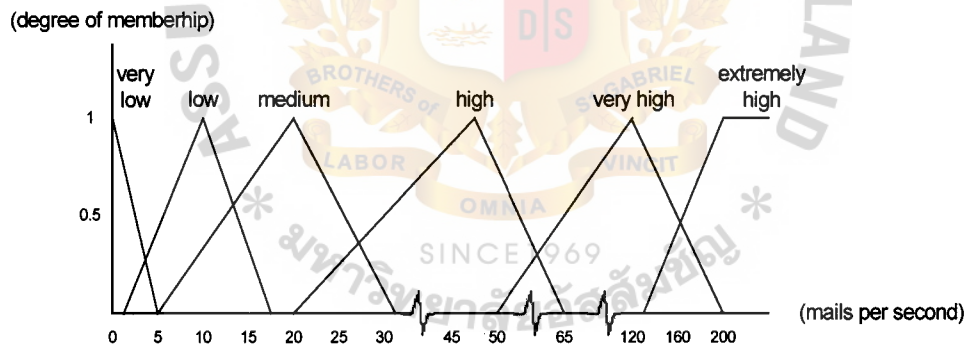


Figure 27: The membership function of the mail frequency

### Mail frequency membership function

- *Very low* is the set of very low number of mails per 3 minutes. (This indicates the normal amount of mails on the live traffic).
- *Low* is the set of low number of mails per 3 minutes.
- *Medium* is the set of medium number of mails per 3 minutes.
- *High* is the set of high number of mails per 3 minutes.
- *Very high* is the set of very high number of mails per 3 minutes.
- *Extremely high* is the set of extremely high number of mails per 3 minutes. (This indicates the recipient is suddenly email-bombed).

The MAIL LEVEL BOX output variable, mail level, also comprises of six fuzzy sets as show in Figure 28.

### Mail level membership function

These sets of Mail level are set of normalized mail frequency. Therefore, the ICMP reply level also comprises of three fuzzy sets (level 0, 1, and level 2) as shown in figure below.

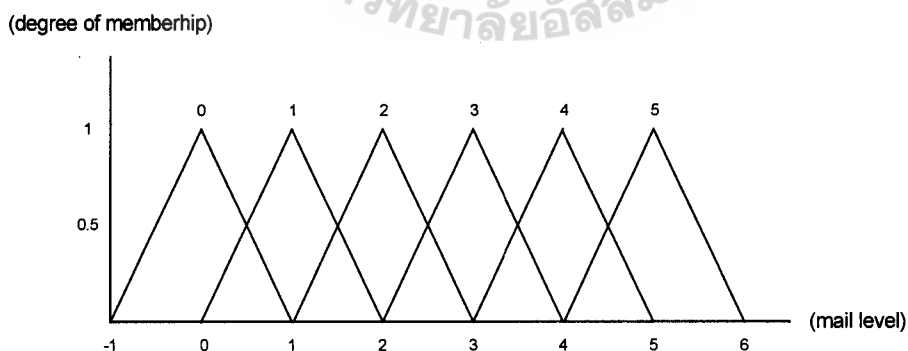


Figure 28: The membership function of mail level

### MAIL LEVEL BOX fuzzy rules

These rules are used to normalize the mail frequency to become the mail level.

1. If (Mail frequency is very low) then (Mail level is 0)
2. If (Mail frequency is low) then (Mail level is 1)
3. If (Mail frequency is medium) then (Mail level is 2)
4. If (Mail frequency is high) then (Mail level is 3)
5. If (Mail frequency is very high) then (Mail level is 4)
6. If (Mail frequency is extremely high) then (Mail level is 5)

### MAIL DETECTOR BOX

This fuzzy box employs mail level and weighted accumulative number to determine the attack possibility of the occurring event. These two input variables' membership functions are shown in Figure 29 and 30.

The first input variable, mail level, composes of 5 fuzzy sets (0, 1, 2, 3, 4, and 5). These sets are the normalized number of mail frequency.

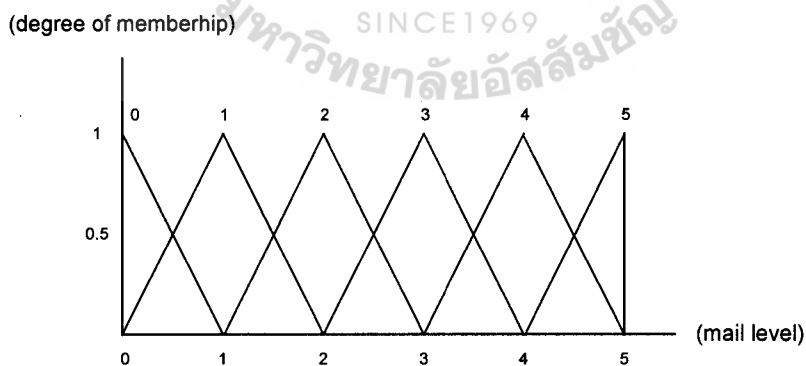


Figure 29: The mail level membership function of MAIL DETECTOR BOX

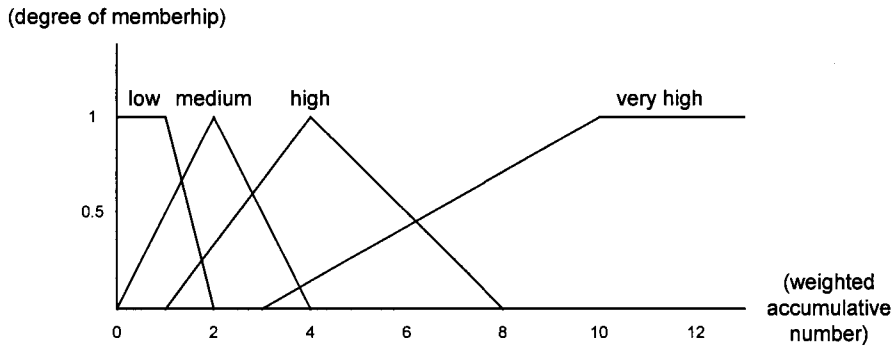


Figure 30: The weighted accumulative number of MAIL DETECTOR BOX

### Weighted accumulative number membership function

- *Low* is the set of low number of mails directed to the same recipient in past 30 minutes. (This indicates that there may not be the attack during these minutes).
- *Medium* is the set of medium number of mails directed to the same recipient in past 30 minutes. (This indicates that there may be the attack in few past seconds).
- *High* is the set of high number of mails directed to the same recipient in past 30 minutes. (This indicates that there may be the attack in many past seconds).
- *Very high* is the set of very high number of mails directed to the same recipient in past 30 minutes. (This indicates that there may be the severe attack in all past seconds).

The output of MAIL DETECTOR BOX is the attack possibility. The attack possibility composes of four fuzzy sets (normal, abnormal, attack, and server attack). Its membership function are shown in Figure 31.

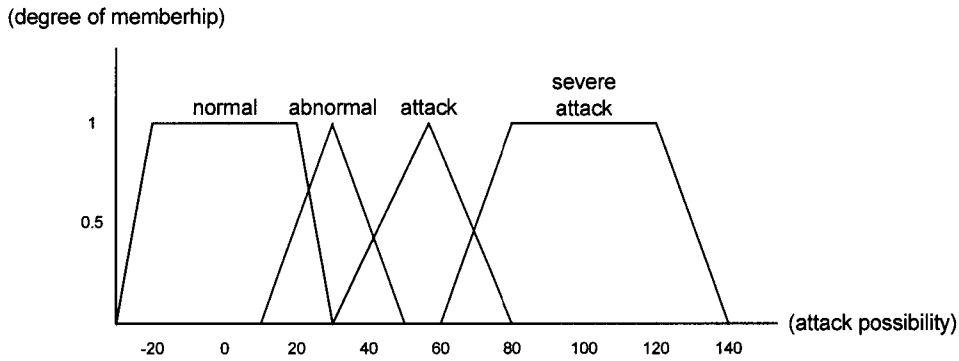


Figure 31: The attack possibility membership function of MAIL DETECTOR BOX

#### Attack Possibility membership function

- *Normal* is the set of low attack possibility in current second (This indicates that there is a normal to abnormal state).
- *Abnormal* is the set of medium attack possibility in current second (This indicates that there is an abnormal to warning state).
- *Attack* is the set of high attack possibility in current second (This indicates that there is a warning to attack state).
- *Severe attack* is the set of very high attack possibility in current second (This indicates that there is an attack to critical attack state).

#### MAIL DETECTOR BOX fuzzy rules

The following fuzzy rules are derived form the heuristic rules, that are set based on the expert knowledge, as explained above in *Fuzzy rule setting up criteria* section. These rules are employed to figure out the attack possibility.

1. If (Mail level is 0) and (Weighted accumulative number is low)  
then (Attack possibility is normal)
2. If (Mail level is 1) and (Weighted Accumulative number is low)  
then (Attack possibility is normal)
3. If (Mail level is 2) and (Weighted Accumulative number is low)  
then (Attack possibility is abnormal)
4. If (Mail level is 3) and (Weighted Accumulative number is low)  
then (Attack possibility is attack)
5. If (Mail Level is 4) and (Weighted Accumulative number is low)  
then (Attack possibility is attack)
6. If (Mail level is 5) and (Weighted Accumulative number is low)  
then (Attack possibility is severe attack)
7. If (Mail lvel is 0) and (Weighted Accumulative number is medium)  
then (Attack possibility is normal)
8. If (Mail level is 1) and (Weighted Accumulative number is medium)  
then (Attack possibility is abnormal)
9. If (Mail level is 2) and (Weighted Accumulative number is medium)  
then (Attack possibility is attack)
10. If (Mail level is 3) and (Weighted Accumulative number is medium)  
then (Attack possibility is attack)
11. If (Mail level is 4) and (Weighted Accumulative number is medium)  
then (Attack possibility is severe attack)
12. If (Mail level is 5) and (Weighted Accumulative number is medium)  
then (Attack possibility is severe attack)
13. If (Mail level is 0) and (Weighted Accumulative number is high)



- then (Attack possibility is abnormal)
14. If (Mail level is 1) and (Weighted Accumulative number is high)  
then (Attack possibility is attack)
15. If (Mail level is 2) and (Weighted Accumulative number is high)  
then (Attack possibility is attack)
16. If (Mail level is 3) and (Weighted Accumulative number is high)  
then (Attack possibility is severe attack)
17. If (Mail level is 4) and (Weighted Accumulative number is high)  
then (Attack possibility is severe attack)
18. If (Mail level is 5) and (Weighted Accumulative number is high)  
then (Attack possibility is severe attack)
19. If (Mail level is 0) and (Weighted Accumulative number is very high)  
then (Attack possibility is attack)
20. If (Mail level is 1) and (Weighted Accumulative number is very high)  
then (Attack possibility is severe attack)
21. If (Mail level is 2) and (Weighted Accumulative number is very high)  
then (Attack possibility is severe attack)
22. If (Mail level is 3) and (Weighted Accumulative number is very high)  
then (Attack possibility is severe attack)
23. If (Mail level is 4) and (Weighted Accumulative number is very high)  
then (Attack possibility is severe attack)
24. If (Mail Level is 5) and (Weighted Accumulative number is very high)  
then (Attack possibility is severe attack)

FTP Password Guessing Detector

This section describes the mechanism for detection FTP password guessing. The network traffic is captured and examined for the packets matched the FTP password guessing signature, described above. The captured traffic is also filtered and collected by FPM. FPM counts for the number of “*login incorrect*” packets in every second and PM also keeps only the following information of these matched packets. Example of preprocessed data shown in Table 8.

- Timestamp
- Source IP number (victim)
- Destination IP number (hacking machine)

Table 8: Example of preprocessed FTP *login incorrect* connection in 1<sup>st</sup> second

Timestamp	Src host	Dst host	Frequency
1	Victim1	Attacker1	14
1	Victim2	Attacker2	2
.	.	.	.
.	.	.	.
.	.	.	.
1	Victim <i>n</i>	Attacker <i>m</i>	1

Then at the end of every second, FPM forwards the frequencies of FTP *login incorrect* packets to FD. The FTP password guessing detector also consists of two fuzzy boxes, FTP LEVEL BOX and FTP DETECTOR BOX, as shown in Figure 32. However, the functions of these two boxes and Weighted Accumulate Module are very similar to other detectors.

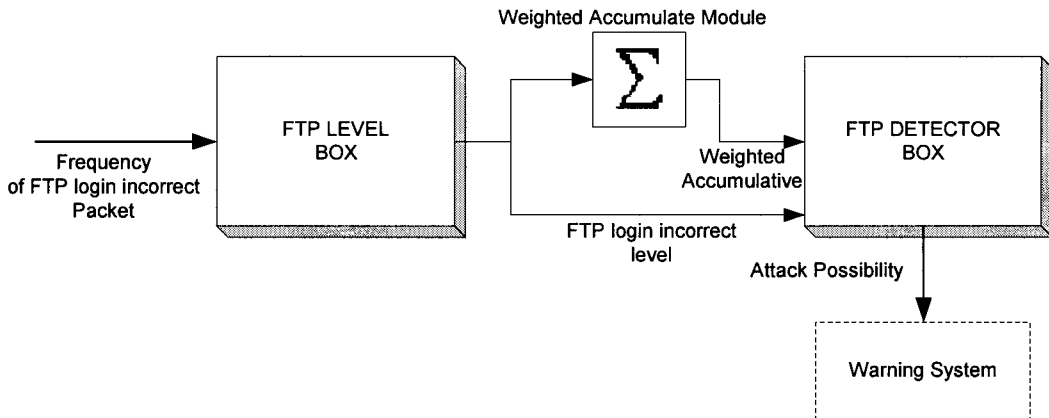


Figure 32: The FTP password guessing detector framework

### FTP LEVEL BOX

The function of the first fuzzy box is to normalize the input variable, FTP login incorrect frequency, to become the FTP login incorrect level. This variable comprises of three fuzzy sets (low, medium, and high) as shown in Figure 33.

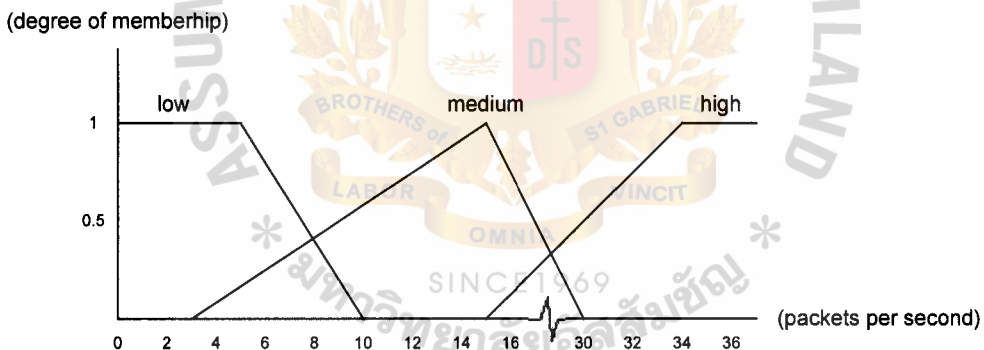


Figure 33: The membership function of the of FTP login incorrect packet frequency

### **FTP login incorrect frequency membership function**

- *Low* is the set of low number of FTP login incorrect packets per second. (This indicates the normal number of FTP login incorrect packets on live traffic).
- *Medium* is the set of medium number of FTP login incorrect packets per second.

- *High* is the set of high number of FTP login incorrect packets per second. (This indicates the FTP server might be guessed for a password).

The FTP LEVEL BOX output variable, FTP login incorrect level, also comprises of three fuzzy sets (0, 1, and 2) and show in Figure 34. The sets are the normalized number of FTP login incorrect frequency.

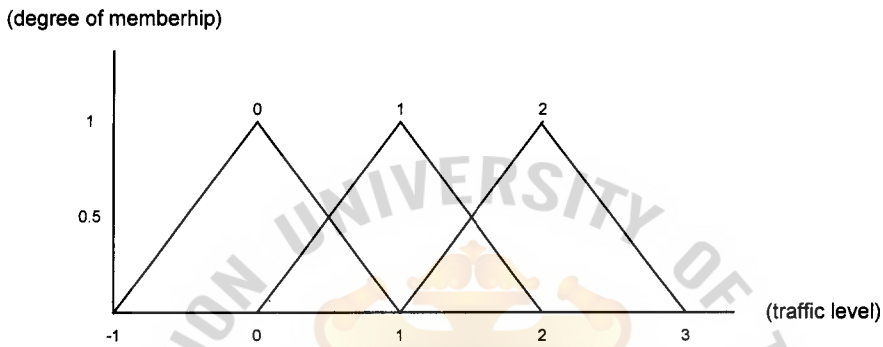


Figure 34: The membership function of FTP login incorrect level

#### FTP LEVEL BOX fuzzy rules

These rules are used to normalize FTP login incorrect frequency to become FTP login incorrect level.

1. If (FTP login incorrect frequency is low) then (FTP login incorrect level is 0)
2. If (FTP login incorrect frequency is medium) then (FTP login incorrect level is 1)
3. If (FTP login incorrect frequency is high) then (FTP login incorrect level is 2)

Next, the FTP login incorrect levels during past 10 seconds are accumulated together to form the weighted accumulative number by Weighted Accumulate Module. Which this number provide the past traffic behavior.

Then these two inputs, FTP login incorrect level of current second and weighted accumulative number, are employed by FTP DETECTOR BOX for determining the attack possibility of the event.

FTP DETECTOR BOX

The two input variables are the FTP login incorrect level and weighted accumulative number as shown in Figure 35 and 36.

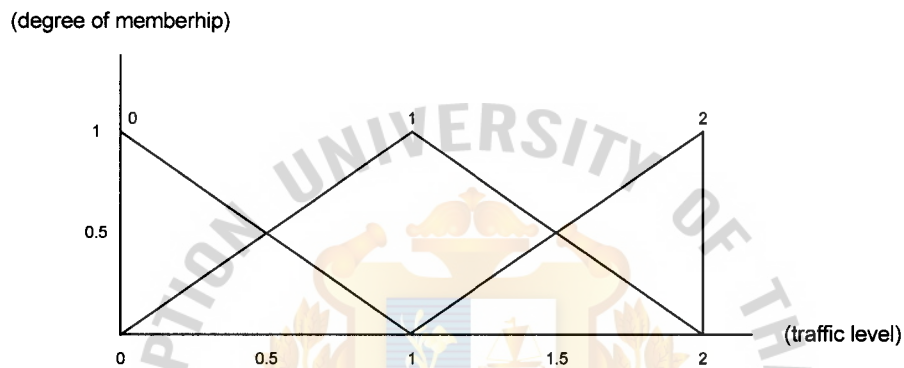


Figure 35: The FTP login incorrect level membership function of FTP DETECTOR BOX

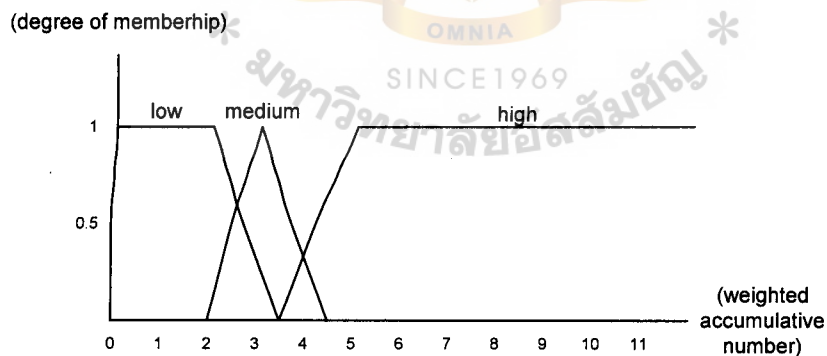


Figure 36: The weighted accumulative number of FTP DETECTOR BOX

### Weighted accumulative number membership function

- *Low* is the set of low number of FTP login incorrect packets in past 10 second. (This indicates that there may not be the password guessing during these minutes).
- *Medium* is the set of medium number of FTP login incorrect packets in past 10 second. (This indicates that there may be the password guessing in few past seconds).
- *High* is the set of high number of FTP login incorrect packets in past 10 second. (This indicates that there may be the password guessing in many past seconds).

The output of FTP DETECTOR BOX is the attack possibility. The attack possibility composes of three fuzzy sets (normal, abnormal, and attack). Its membership function are shown in Figure 37.

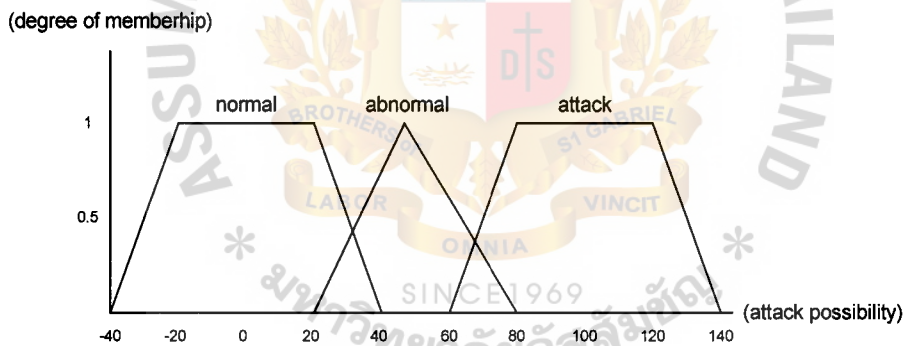


Figure 37: The attack possibility membership function of FTP DETECTOR BOX

### Attack Possibility membership function

- *Normal* is the set of low attack possibility in current second (This indicates that there is a normal to abnormal state).

- *Abnormal* is the set of medium attack possibility in current second (This indicates that there is an abnormal, warning to attack state).
- *Attack* is the set of very high attack possibility in current second (This indicates that there is an attack to critical attack state).

### **FTP DETECTOR BOX fuzzy rules**

The following fuzzy rules are derived from the heuristic rules, that are set based on the expert knowledge, as explained above in *Fuzzy rule setting up criteria* section. These rules are employed to figure out the attack possibility.

1. IF (FTP login incorrect Level is 0) and (Weighted accumulative number is low) then (Attack possibility is normal)
2. IF (FTP login incorrect Level is 0) and (Weighted accumulative number is medium) then (Attack possibility is abnormal)
3. IF (FTP login incorrect Level is 0) and (Weighted accumulative number is high) then (Attack possibility is Attack)
4. IF (FTP login incorrect Level is 1) and (Weighted accumulative number is low) then (Attack possibility is abnormal)
5. IF (FTP login incorrect Level is 1) and (Weighted accumulative number is medium) then (Attack possibility is abnormal)
6. IF (FTP login incorrect Level is 1) and (Weighted accumulative number is high) then (Attack possibility is Attack)
7. IF (FTP login incorrect Level is 2) and (Weighted accumulative number is short) then (Attack possibility is abnormal)



8. IF (FTP login incorrect Level is 2) and (Weighted accumulative number is medium) then (Attack possibility is abnormal)
9. IF (FTP login incorrect Level is 2) and (Weighted accumulative number is high) then (Attack possibility is Attack)

### **Telnet Password Guessing Detector**

The detection mechanism of Telnet password guessing is almost the same as FTP password guessing detector. The difference of these two is that the hacker tries to guess the passwords of root, administrator, or other on Telnet server rather than FTP server. Therefore, the server should be synchronized many times per second and the server keeps sending the hacking machine, “Login incorrect”. Therefore FPM collected the network traffics that match the Telnet Password Guessing signature.

The following figure shows the framework of Telnet password guessing detector. As shown in Figure 38, this framework composes of two fuzzy boxes and one additional Weighted Accumulate Module.

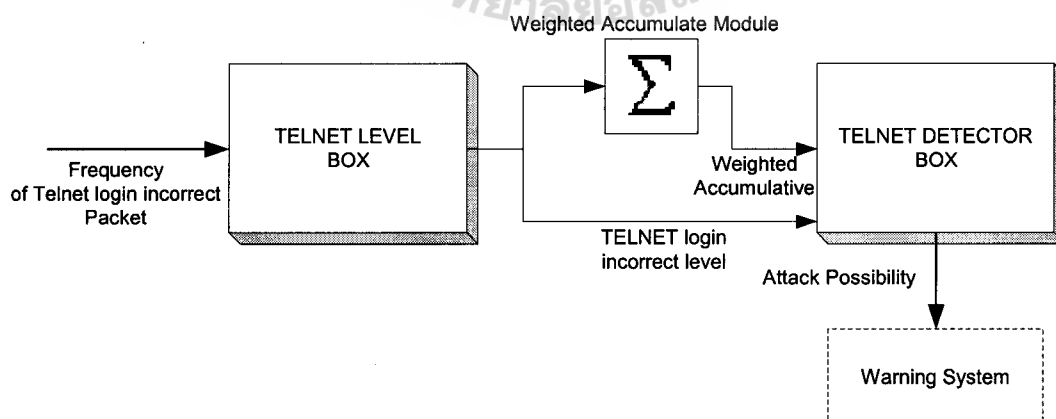


Figure 38: The Telnet password guessing detector framework

Like other detectors, first fuzzy box (TELNET LEVEL BOX) is used to normalize the input variable, Telnet login incorrect frequency. This frequency number is discovered by FPM. FPM captures the network traffic according the predefined Telnet password guessing signature and counts for the number of occurrences in every second.

### TELNET LEVEL BOX

There is time delay during authentication when logging in to the server via using Telnet. Thus the number of Telnet login incorrect packets per second is not too high. Therefore, the following rules are used to normalize the input variable, Telnet login incorrect frequency, to become the output variable, Telnet login incorrect level.

#### **TELNET LEVEL BOX rules**

These rules used to normalize the input variable:

1. If (Telnet login incorrect frequency is 0)  
then (Telnet login incorrect level is 0)
2. If (Telnet login incorrect frequency is 1)  
then (Telnet login incorrect level is 1)
3. If (Telnet login incorrect frequency is 2)  
then (Telnet login incorrect level is 2)
4. If (Telnet login incorrect frequency is 3)  
then (Telnet login incorrect level is 3)
5. If (Telnet login incorrect frequency is greater than 3)  
then (Telnet login incorrect level is 4)

Next, the Telnet login incorrect levels during past 10 seconds are accumulated together to form the weighted accumulative number by Weighted Accumulate Module. This number provides the FIDS the past traffic behavior.

Then these two inputs, Telnet login incorrect level of current second and the weighted accumulative number, are employed by TELNET DETECTOR BOX for determining the attack possibility of the event.

### TELNET DETECTOR BOX

The two input variables of this box are the Telnet login incorrect level and weighted accumulative number as shown in Figure 39 and 40.

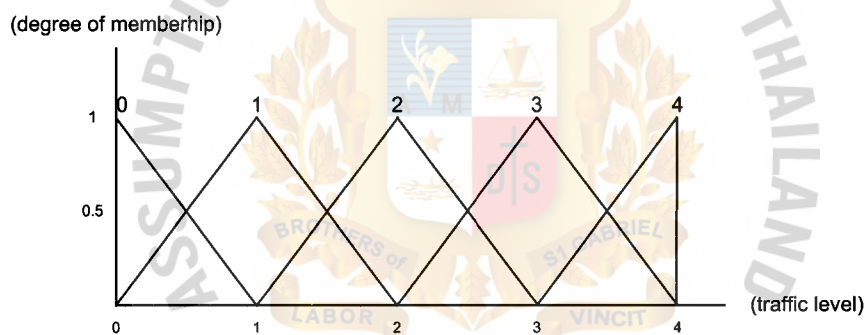


Figure 39: The Telnet login incorrect level membership function of TELNET DETECTOR BOX

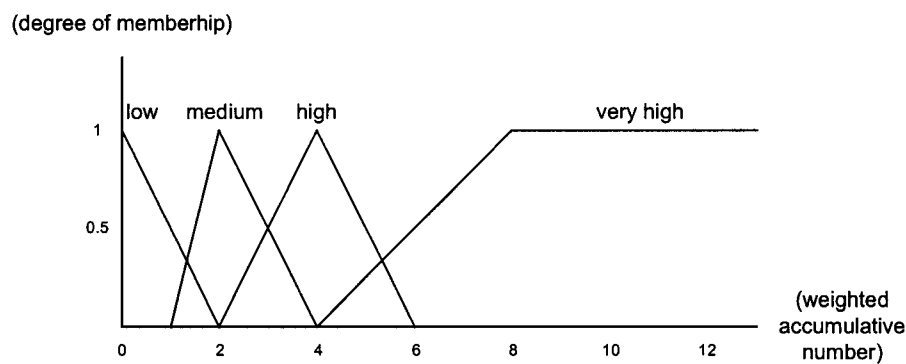


Figure 40: The weighted accumulative number of TELNET DETECTOR BOX

### Weighted accumulative number membership function

- *Low* is the set of low number of Telnet login incorrect packets in past 10 seconds. (This indicates that there may not be the password guessing during these minutes).
- *Medium* is the set of medium number of Telnet login incorrect packets in past 10 seconds. (This indicates that there may be the password guessing in few past seconds).
- *High* is the set of high number of Telnet login incorrect packets in past 10 seconds. (This indicates that there may be the password guessing in many past seconds).
- *Very high* is the set of very high number of Telnet login incorrect packets in past 10 seconds. (This indicates that there may be the password guessing in all past seconds).

The output of TELNET DETECTOR BOX is the attack possibility. The attack possibility composes of four fuzzy sets (normal, abnormal, attack and server attack). Its membership function are shown in Figure 41.

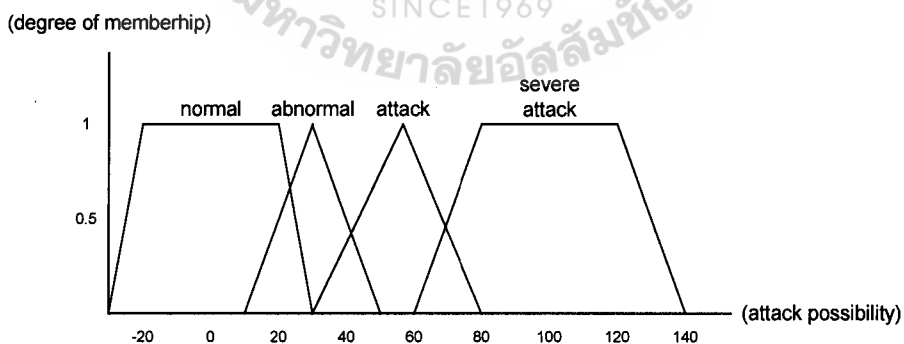


Figure 41: The attack possibility membership function of  
TELNET DETECTOR BOX

### **Attack Possibility membership function**

- *Normal* is the set of low attack possibility in current second (This indicates that there is a normal to abnormal state).
- *Abnormal* is the set of medium attack possibility in current second (This indicates that there is an abnormal to warning state).
- *Attack* is the set of high attack possibility in current second (This indicates that there is a warning to attack state).
- *Severe attack* is the set of very high attack possibility in current second (This indicates that there is an attack to critical attack state).

### **TELNET DETECTOR BOX fuzzy rules**

The following fuzzy rules are derived from the heuristic rules, that are set based on the expert knowledge, as explained above in *Fuzzy rule setting up criteria* section. These rules are employed to figure out the attack possibility.

1. If (Telnet login incorrect level is 0) and (Weighted accumulative number is small) then (Attack possibility is normal)
2. If (Telnet login incorrect level is 0) and (Weighted accumulative number is medium) then (Attack possibility is abnormal)
3. If (Telnet login incorrect level is 0) and (Weighted accumulative number is high) then (Attack possibility is attack)
4. If (Telnet login incorrect level is 0) and (Weighted accumulative number is very high) then (Attack possibility is severe attack)

5. If (Telnet login incorrect level is 1) and (Weighted accumulative number is small) then (Attack possibility is normal)
6. If (Telnet login incorrect level is 1) and (Weighted accumulative number is medium) then (Attack possibility is abnormal)
7. If (Telnet login incorrect level is 1) and (Weighted accumulative number is high) then (Attack possibility is attack)
8. If (Telnet login incorrect level is 1) and (Weighted accumulative number is very high) then (Attack possibility is sever attack)
9. If (Telnet login incorrect level is 2) and (Weighted accumulative number is small) then (Attack possibility is abnormal)
10. If (Telnet login incorrect level is 2) and (Weighted accumulative number is medium) then (Attack possibility is abnormal)
11. If (Telnet login incorrect level is 2) and (Weighted accumulative number is high) then (Attack possibility is attack)
12. If (Telnet login incorrect level is 2) and (Weighted accumulative number is very high) then (Attack possibility is sever attack)
13. If (Telnet login incorrect level is 3) and (Weighted accumulative number is small) then (Attack possibility is normal)
14. If (Telnet login incorrect level is 3) and (Weighted accumulative number is medium) then (Attack possibility is attack)
15. If (Telnet login incorrect level is 3) and (Weighted accumulative number is high) then (Attack possibility is attack)
16. If (Telnet login incorrect level is 3) and (Weighted accumulative number is very high) then (Attack possibility is sever attack)

17. If (Telnet login incorrect level is 4) and (Weighted accumulative number is small) then (Attack possibility is attack)
18. If (Telnet login incorrect level is 4) and (Weighted accumulative number is medium) then (Attack possibility is attack)
19. If (Telnet login incorrect level is 4) and (Weighted accumulative number is high) then (Attack possibility is sever attack)
20. If (Telnet login incorrect level is 4) and (Weighted accumulative number is very high) then (Attack possibility is sever attack)

### **Port Scanning Detection**

In case of Port scanning detector, FPM collects the traffic of SYN packets and RESET packets that match to the defined Port Scan signature. Therefore, if any host has received many SYN packets for variety of services and it always response back with RESET packets, FPM will count the number of the SYN and RESET packet pairs in every second. The PFM keeps only the necessary information including:

- source IP number (attacking machine)
- destination IP number (victim)
- destination service (scanned port)

Unlike other detector, the Port Scan detector has only one fuzzy box as shown in Figure 42. The frequencies of SYN and RESET pairs, which are matched with the defined signatures, are sent through PORT SCAN DETECTOR BOX. This rules box



produces the severity of being scanned (attack possibility) within short interval of time.

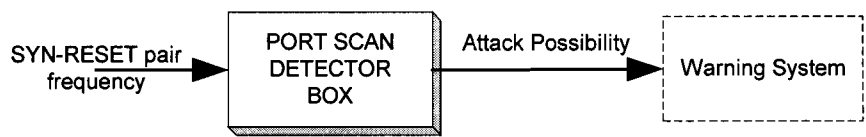


Figure 42: The port scanning detector framework

PORT SCAN DETECTOR BOX

The input variable is the SYN-RESET pair frequency as shown in Figure 43. However, the setting-up rationales are the same as SYN-FLOOD DETECTOR.

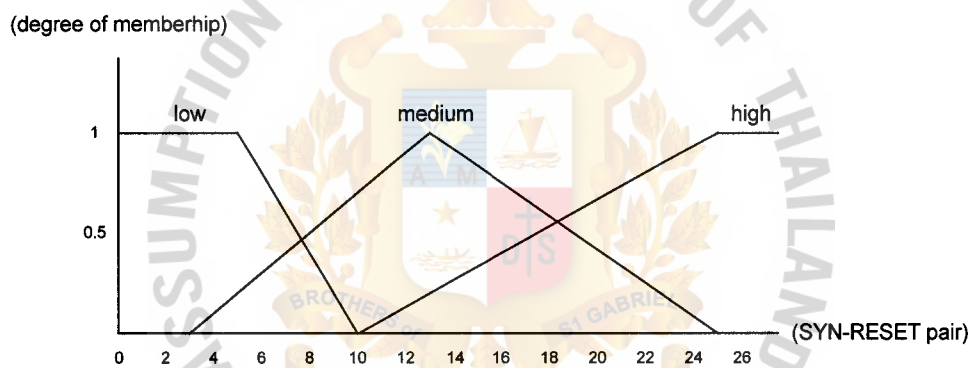


Figure 43: The SYN-RESET pair frequency membership function

**SYN-RESET pair frequency membership function**

- *Low* is the set of low amount of SYN-RESET pairs per second on live traffic (This indicates normal SYN-RESET pair traffic).
- *Medium* is the set of moderate amount of SYN-RESET pairs per second on live traffic (This indicates abnormal SYN-RESET pair traffic).
- *High* is the set of high amount of SYN-RESET pairs per second on live traffic (This indicates low suspicious degree of port scanning attack).

- *Very high* is the set of very high amount of SYN-RESET pairs per second on live traffic (This indicates high suspicious degree of port scanning attack).

The output variable is the attack possibility. The attack possibility composes of three fuzzy sets (normal, abnormal, and attack). Its' membership function are shown in Figure 44.

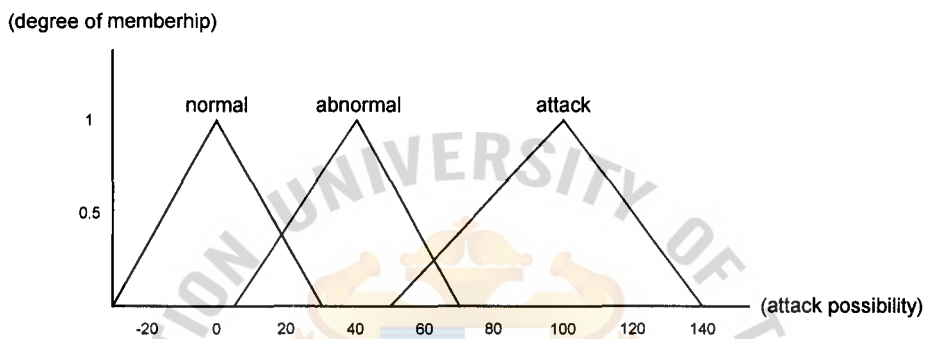


Figure 44: The attack possibility membership function of  
PORT SCAN DETECTOR BOX

#### Attack Possibility membership function

- *Normal* is the set of low attack possibility in current second (This indicates that there is the normal to abnormal state).
- *Abnormal* is the set of medium attack possibility in current second (This indicates that there is the normal to attack state).
- *Attack* is the set of very high attack possibility in current second (This indicates that there is the warning to critical attack state).

### PORT SCAN DETECTOR BOX fuzzy rules

The following fuzzy rules are derived from the heuristic rules, that are set based on the expert knowledge, as explained above in *Fuzzy rule setting up criteria* section. These rules are employed to figure out the attack possibility.

1. If (SYN-RESET pair frequency is low) then (Attack possibility is normal)
2. If (SYN-RESET pair frequency is medium) then (Attack possibility is abnormal)
3. If (SYN-RESET pair frequency is high) then (Attack possibility is attack)

### Membership Function Setting and Adjusting Mechanism

The traffic frequency, traffic level, weighted accumulative number, and attack possibility (fuzzy variables) are the tunable parameters of FIDS. To adjust membership functions, the both normal and abnormal traffic must be collected, then the network traffic datasets are collected.

- 1) Sets of normal network traffics
- 2) Sets of abnormal/misuse network traffics with:
  - SYN-Flood attack
  - UDP-Flood attack
  - Ping-of-Death attack
  - Email bomb
  - Password guessing (ftp and telnet password guessing)
  - Port scan

These datasets are learned and used to set and adjust the membership functions of each detector. In case of SYN LEVEL BOX, the set of *low* is set and adjusted base on the average number of normal SYN packets per second, learned from the normal-traffic datasets. Also the average number of abnormal datasets, contains high degree of SYN-Flood attack, is used to set and tune the set of *extremely high*. Thereafter the least fuzzy sets, *medium*, *high*, and *very high*, are set and adjusted by using statistical measures. Moreover this mechanism is also used for setting and adjusting other traffic frequency membership functions. However tuning the membership functions depend on the environment of a unique network. Therefore to apply the FIDS with the different network environment the datasets should be recollected and then the membership functions should be readjusted again.



# CHAPTER 4: EXPERIMENTAL RESULT AND ANALYSIS

## 4.1 Event Generation and Gathering

The experiment has been conducted by capturing the network traffic. These sets of network traffic comprise of

- 1) Sets of normal network traffic
- 2) Sets of abnormal/misuse network traffic with:
  - SYN-Flood attack (generated by using TFN2K)
  - UDP-Flood attack (generated by using TFN2K)
  - Ping-of-Death attack
  - Email bomb (generated by using Quikfyre)
  - FTP and Telnet Password guessing (generated by using BrutusA2)
  - Port scan (generated by using SupperScan)

## 4.2 Testing, Experimental Results and Analysis

In order test FIDS, it is necessary to develop some Denial-of-Service attacks, and other intrusive behaviors, mentioned above. Its detection is to be tested. The testing results of FIDS are compared with the result of threshold-based detections, because the threshold-based detection is widely used to defend the servers or network resources against the attacks. As mentioned in [9], when the target computer was filled a very small queue of half open port connections (in case of SYN-Flood attack),

the computer stops answering requests on the attacked port once the threshold is reached. Moreover, the FIDS testing results are also compared with the rule-setting criteria.

### Threshold-based detection

The threshold-based detection is used with  $x$  threshold levels (in this case, there are five threshold levels). Depending on where the value falls in the threshold level range, a severity state is assigned. Five severity states (normal, abnormal, warning, attack and critical attack) are used. Using statistical measures derived from the datasets, both normal and abnormal network-traffics, may be used to set the thresholds. In addition, the threshold level setting also relates to the packet frequency membership function (input variable) of each detector.

#### **1. SYN-Flood Attack Testing and Analysis**

To test SYN-Flood detector, TFN2K was used to develop SYN-Flood attack by flooding a victim the number of SYN packets. Several testing have been conducted under different amount of SYN packets per second. The following figures show the detection result of FIDS versus three threshold-based detectors (Threshold 1, Threshold 2, and Threshold 3) and rule-setting criteria. According to the rule-setting criteria, the FIDS should be able to detect SYN-Flood attack when the victim has continuously received more than 1000 half-open connections in one second. Moreover, if the severe attack occurs then the FIDS should be able to detect this attack within 10 seconds.

The threshold levels are set based on the number of SYN-packet per second and these threshold levels also relate to the way that SYN packet frequency membership function has been adjusted. The following tables show the threshold levels of each threshold-based detector.

Table 9: Threshold level of SYN-Flood Threshold 1

Severity State	Threshold Level (packets/second)	Attack Possibility
Normal	0-25	0
Abnormal	26-50	20
Warning	51-200	40
Attack	201-950	60
Critical attack	951-2000	80
Critical attack	>2000	100

Table 10: Threshold level of SYN-Flood Threshold 2

Severity State	Threshold Level (packets/second)	Attack Possibility
Normal	0-35	0
Abnormal	36-150	20
Warning	151-500	40
Attack	501-1800	60
Critical attack	1801-2000	80
Critical attack	>2000	100

Table 11: Threshold level of SYN-Flood Threshold 3

Severity State	Threshold Level (packets/second)	Attack Possibility
Normal	0-25	0
Abnormal	26-50	20
Warning	51-250	40
Attack	251-950	60
Critical attack	951-1000	80
Critical attack	>1000	100



1<sup>st</sup> SYN-Flood Testing

The first testing, the attacker attacks a victim by flooding the victim approximately 2,000 half-open connections per second. The detection results are shown in the figures below.

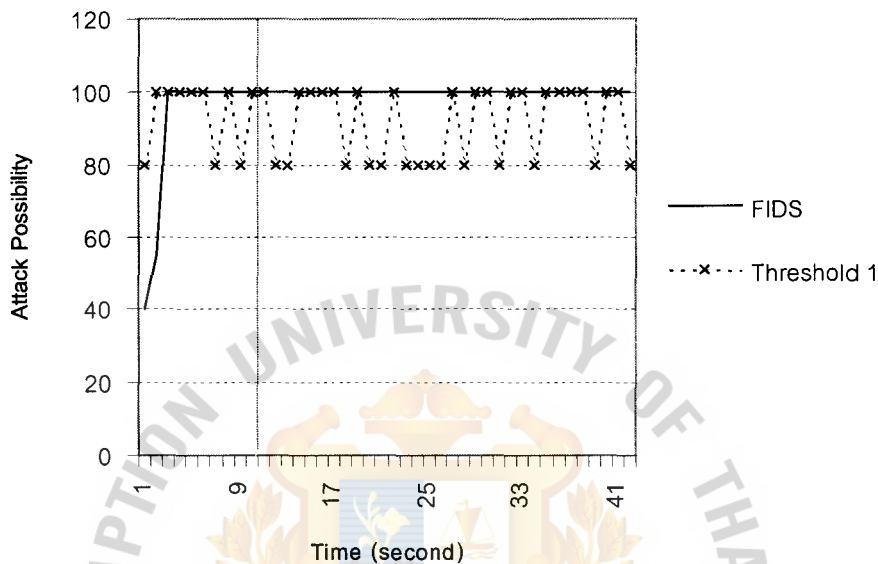


Figure 45a: The detection result of 1<sup>st</sup> SYN-Flood testing, FIDS vs. Threshold 1

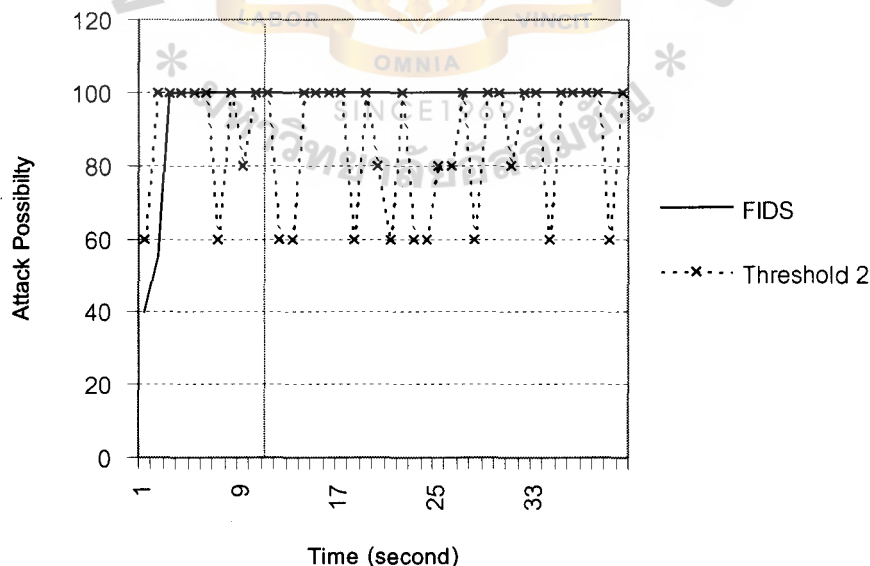


Figure 45b: The detection result of 1<sup>st</sup> SYN-Flood testing, FIDS vs. Threshold 2

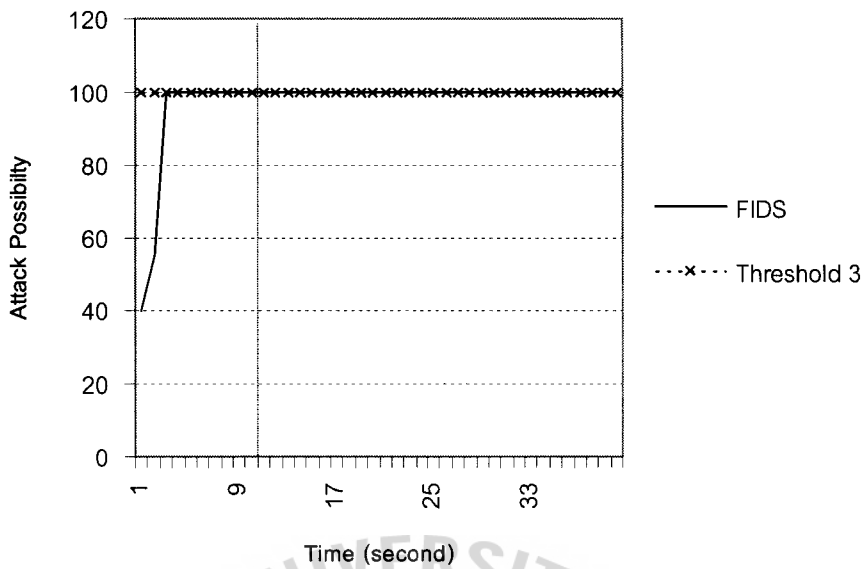


Figure 45c: The detection result of 1<sup>st</sup> SYN-Flood testing, FIDS vs. Threshold 3

As the detection result shown in Figure 45c, FIDS and Threshold 3 gives the similar result since the amount of SYN-packets in each second is extremely high (approximately 2,000). All of them can detect the attack within 10 seconds when compared with the rule-setting criteria because the attacking machine keeps flooding the victim with very high number of half-open connections per second (greater than 1,000 connections per second).

The difference is only the detection result at the beginning of the attack (first few seconds). Because to detect the attack behavior, the FIDS does not consider only the amount of packets but it also considers the weighted accumulative number. If the victim was attacked just few seconds (1 or 2 seconds) by sending the victim approximately 2,000 half-open connections per second. However, this event does not effect the victim's performance much.

Therefore, it may not necessary to detect this event as critical attack. In contrast, if this even occurs then Threshold 1 and Threshold 2 will detect this event as critical attack with the attack severity of 100 whereas the result of FIDS is not the same as threshold detectors. Since to detect the malicious behavior, the FIDS considers both amount of packet in current second and the weighted accumulative number while threshold detectors consider only the amount of packet in particular second. Therefore, FIDS does not detect this event as critical attack but it detects the event as warning state (attack possibility is in between 40 and 60).

Figure 45a and 45b show the result of detection of FIDS versus Threshold 1 and Threshold 2. The results of threshold detectors continue changing between 100 and 80 of attack possibility because the amounts of packet are not always extremely high in every second. When, the amount of SYN packet drops from 2,000 to 1,000 in some second, the detection result of the threshold-based detector also drops. While, the detection result of FIDS does not drop because the amount of packet in that second is still high and the weighted accumulative of attack remains high.

## 2<sup>nd</sup> SYN-Flood Testing

The second testing, the intruder attacks a victim by sending the victim approximately 1,000 half-open connections per second. The amount of SYN-packet is very high traffic. The detection results are shown in the figures below.

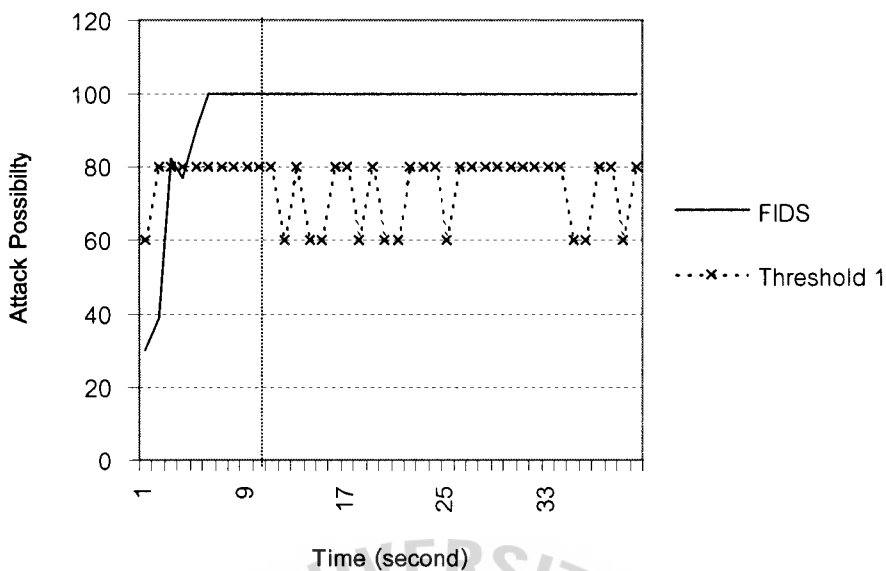


Figure 46a: The detection result of 2<sup>nd</sup> SYN-Flood testing, FIDS vs. Threshold 1

In this case, the results of FIDS, compared with three threshold-based detectors, are better as shown in Figure 46a, 46b, and 46c. The FIDS can detect the attack as a critical attack state within 10 seconds when compared with the rule-setting criteria because the attacking machine is flooding the victim with approximately 1000 half-open connections per second.

Since the amount of packet per second is not extremely high like the first case, the result detected by Threshold 1 and Threshold 2 can detect this event as attack state with the attack possibility varies between 60 and 80. However, the result of threshold-based detector is better when Threshold 3 is used. Threshold 3 gives the better result than Threshold 1 and Threshold 2. Threshold3 detects this event as critical attack state in some seconds and as attack state in other seconds. The attack possibility of the result varies between 60 and 100.

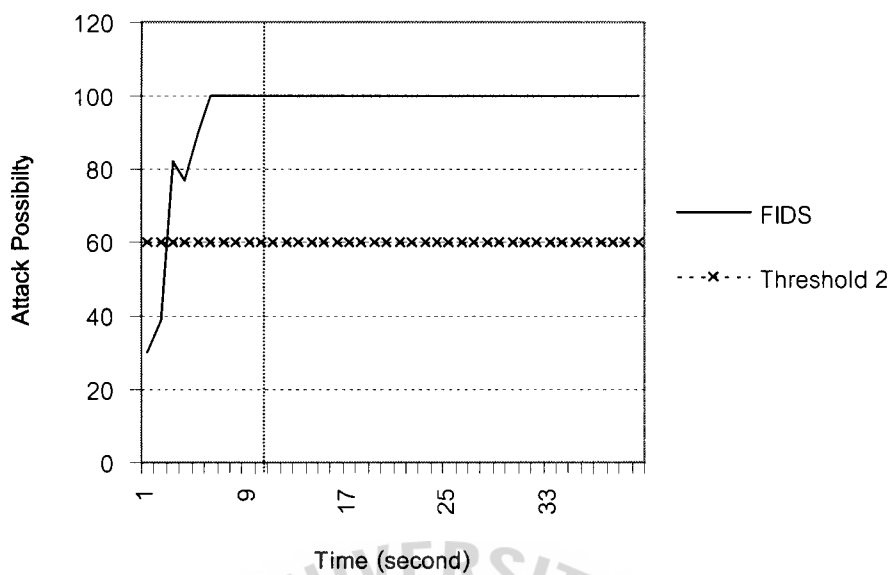


Figure 46b: The detection result of 2<sup>nd</sup> SYN-Flood testing, FIDS vs. Threshold 2

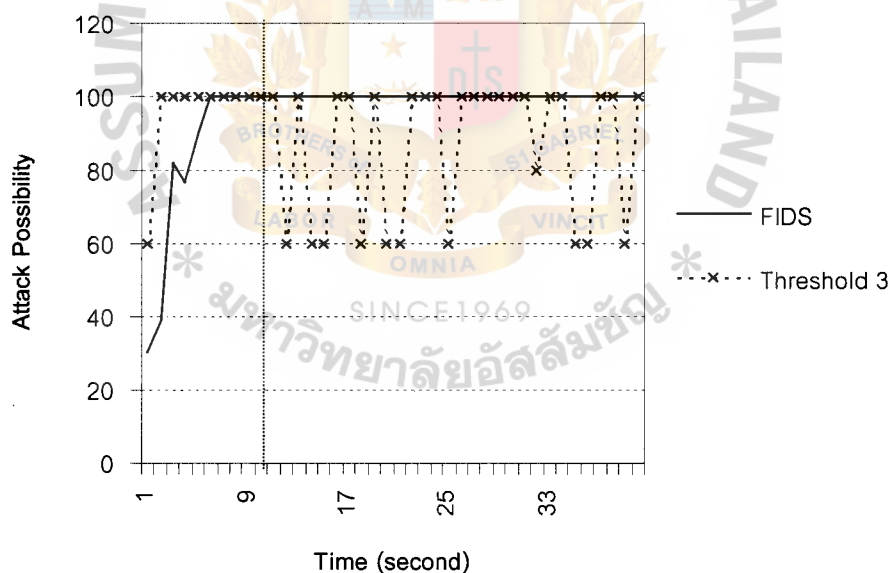


Figure 46c: The detection result of 2<sup>nd</sup> SYN-Flood testing, FIDS vs. Threshold 3

FIDS can detect this event better than other threshold-based detectors, since the intruder continuously attacks by sending very high amount of SYN packet. See Figure 46a, Threshold 3 detects that the packet amount drops in 12<sup>th</sup> second therefore the attack severity also drop. In contrast, the FIDS takes the packet amount into account accompanies with the weighted accumulative during past seconds when it detects the attack.

3<sup>rd</sup> SYN-Flood Testing

This testing, the intruder attacks a victim by sending the victim approximately 120 half-open connections per second. The amount of SYN-packet is high traffic. The results of detection are shown in the figures below.

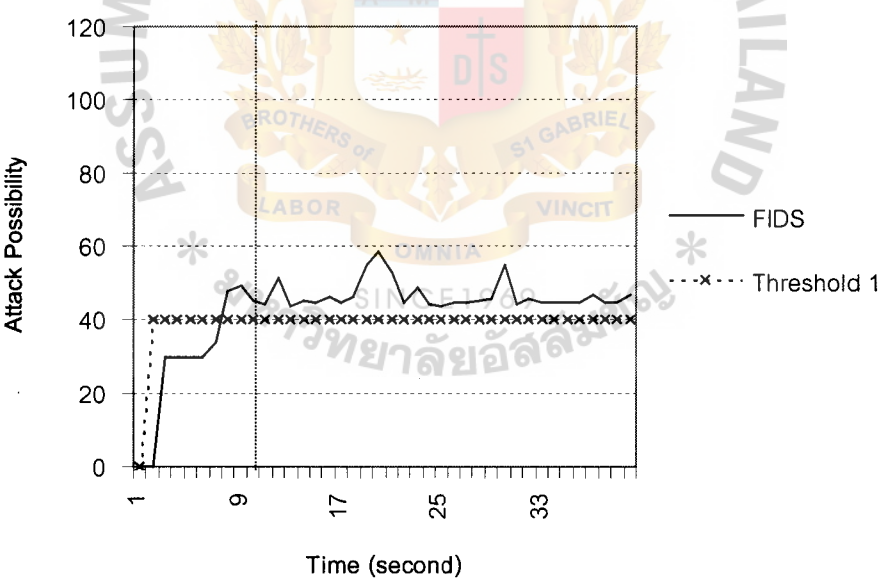


Figure 47a: The detection result of 3<sup>rd</sup> SYN-Flood testing, FIDS vs. Threshold 1

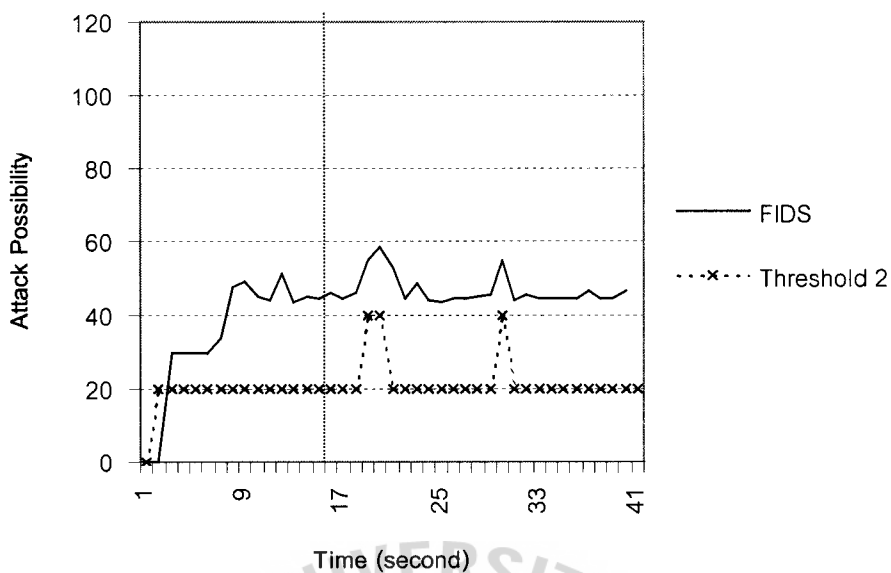


Figure 47b: The detection result of 3<sup>rd</sup> SYN-Flood testing, FIDS vs. Threshold 2

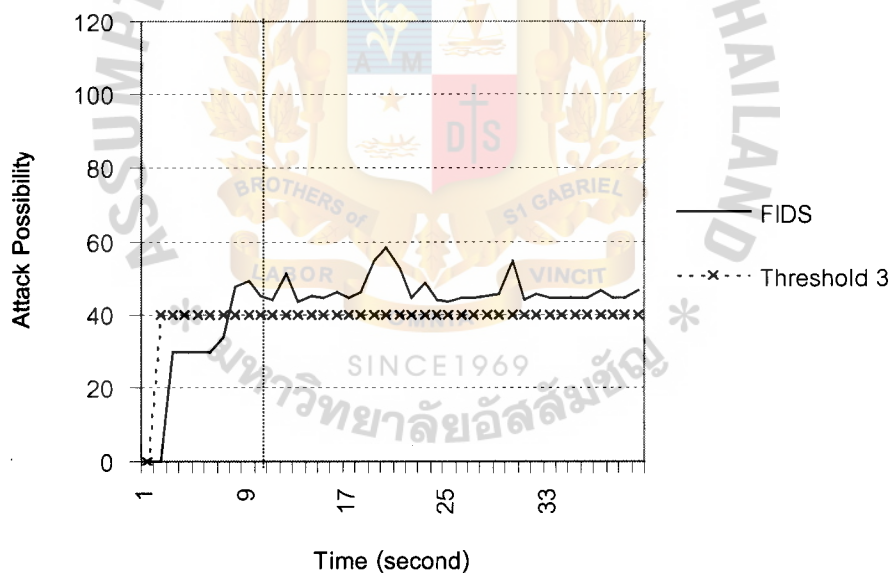


Figure 47c: The detection result of 3<sup>rd</sup> SYN-Flood testing, FIDS vs. Threshold 3

Using FIDS, the detection result gradually increases until it reaches the warning state (attack possibility is in between 40 and 60) because the number of half-open



connections per second in this case is not so high. The FIDS can detect this attack as warning state within 10 seconds when compare with the rule-setting criteria because the attacking machine continuously floods the victim even though the number of half-open connections per second is not reach 1000. Moreover, if the attacker sends the packets continuously then the weighted accumulative is also increasing. Therefore the graph, represented FIDS result, is moving within the warning state. Meanwhile the results produced by threshold-based detector are worse than the FIDS detection result.

## 2. UDP-Flood Attack Testing and Analysis

To test UDP-Flood detector, TFN2K is also used to create UDP-Flood attack by flooding a victim the number of UDP datagrams. Like SYN-Flood detector testing, several tests have been conducted under different amount of UDP datagrams per second. The following figures show the detection results of FIDS versus three threshold-based detectors and rule-setting criteria. According to the rule-setting criteria, the FIDS should be able to detect UDP-Flood attack when the victim has continuously received more than 2000 UDP packets in one second. Moreover, if the severe attack occurs then the FIDS should detect this attack within 10 seconds.

The threshold levels are set on the basis of number of UDP datagrams per second and these threshold levels also relate to the way that UDP packet frequency membership function has been adjusted. The following tables show the threshold levels of each UDP-Flood threshold detector.

Table 12: Threshold level of UDP-Flood Threshold 1

Severity State	Threshold Level (packets/second)	Attack Possibility
Normal	0-56	0
Abnormal	57-240	20
Warning	241-700	40
Attack	701-2000	60
Critical attack	2001-3000	80
Critical attack	>3000	100

Table 13: Threshold level of UDP-Flood Threshold 2

Severity State	Threshold Level (packets/second)	Attack Possibility
Normal	0-70	0
Abnormal	71-500	20
Warning	501-1500	40
Attack	1501-2850	60
Critical attack	2851-3000	80
Critical attack	>3000	100

Table 14: Threshold level of UDP-Flood Threshold 3

Severity State	Threshold Level (packets/second)	Attack Possibility
Normal	0-56	0
Abnormal	57-200	20
Warning	201-500	40
Attack	501-1000	60
Critical attack	1001-2000	80
Critical attack	>2000	100

UDP-Flood Testing

The first UDP-Flood testing, the attacker attacks a victim by sending the victim approximately 4,000 UDP packets per second. This amount of UDP packets is extremely high. The detection results are shown in the figures below.

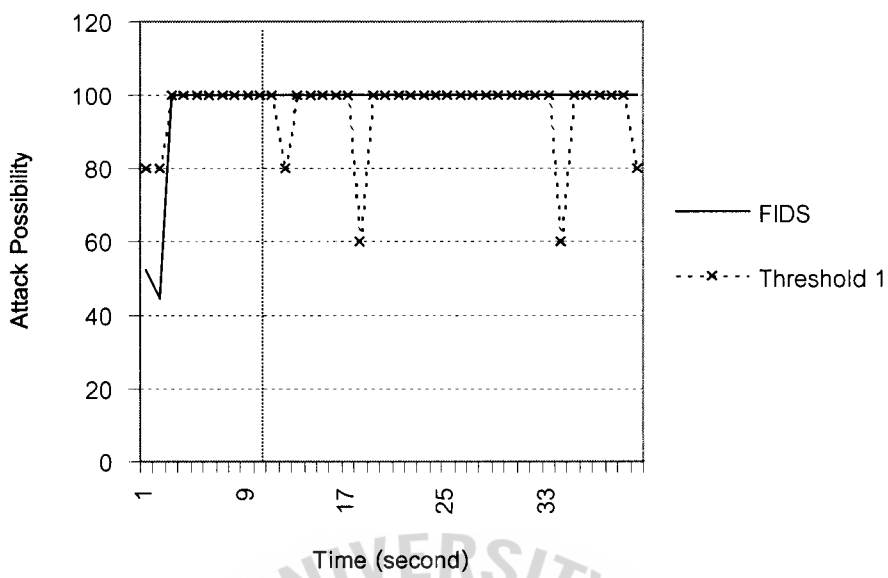


Figure 48a: The detection result of UDP-Flood testing, FIDS vs. Threshold 1

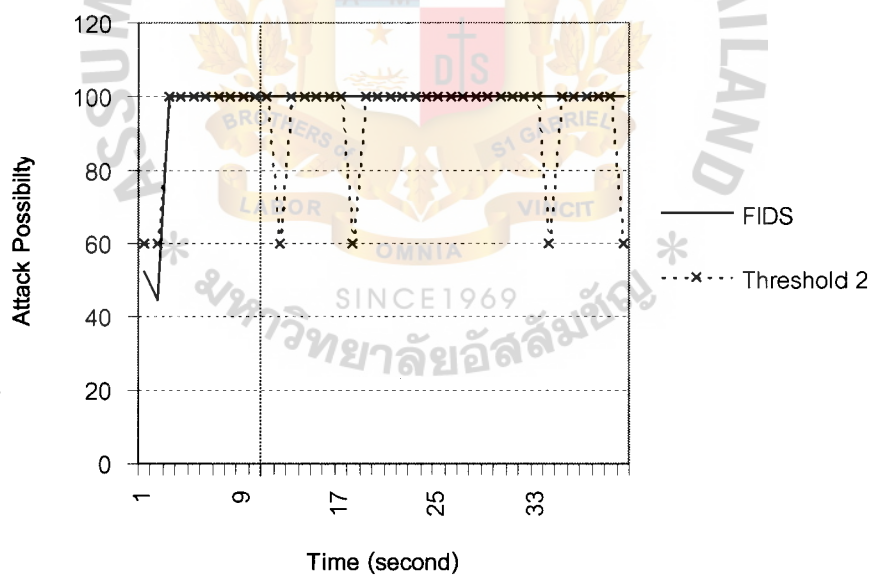


Figure 48b: The detection result of UDP-Flood testing, FIDS vs. Threshold 2

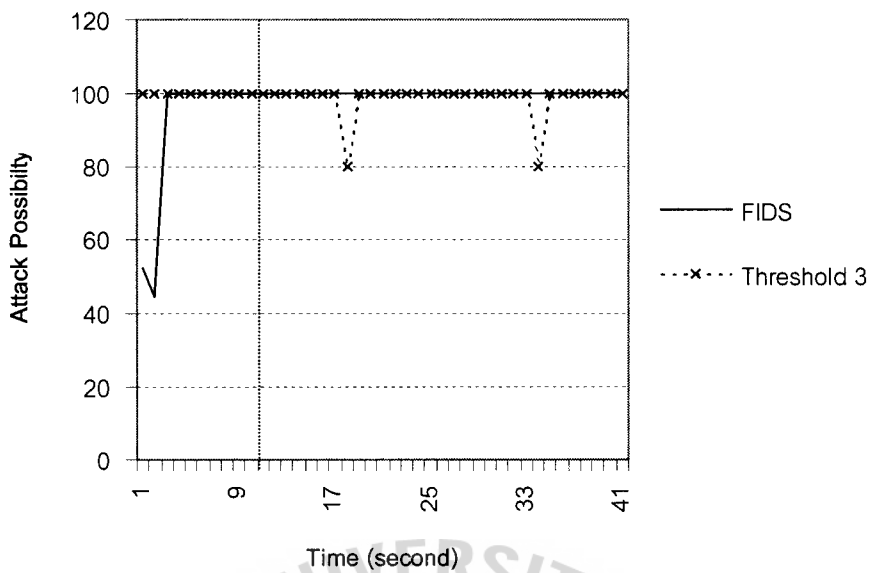


Figure 48c: The detection result of UDP-Flood testing, FIDS vs. Threshold 3

According to the result shown in Figure 48a and 48b, all of them can detect the attack within 10 seconds as compare with the rule-setting criteria because the victim has been flooded with UDP packets more than 2000 packets per second continuously. However, FIDS yields better result than Threshold 1 and Threshold 2 while the FIDS result is similar to the result of Threshold 3 as shown in Figure 48c.

Notwithstanding, the FIDS result does not drop. Thereafter the FIDS's attack severity reaches the critical attack state (with attack severity of 100), even though the number of UDP packet in that second drops from approximate 4,000 to 2,000 UDP packets. Unlike the threshold-based detectors, the attack severity of result drops, when the number of UDP datagram drops in each second.

3. Ping of Death Attack Testing and Analysis

To test Ping-of-Death detector, the victim receives the large size of ICMP echo request packet (ping packet). These large of ICMP echo request packet is fragmented into many ping fragments. Therefore the victim receives many ping fragments and the victim replies these ping fragments with the ICMP reply packet. The number of reply packet is also the same number as request packet. Several testing have been conducted under different sizes of ping. The following figures show the detection result of FIDS versus three threshold-based detectors (Threshold 1, Threshold 2, and Threshold 3) and rules-setting criteria. Due to the rule-setting criteria, the FIDS should be able to detect Ping-of-Death attack when the victim has sent more than 45 ICMP reply fragment with the same identification number in one second. Moreover, if the severe attack occurs then the FIDS should detect this attack within 10 seconds.

The threshold levels are set based on the number of ICMP reply fragments per second and these threshold levels also relate to the way that ICMP reply fragment frequency membership function has been adjusted. The following tables show the threshold levels of each threshold detector

Table 15: Threshold level of Ping-of-Death Threshold 1

Severity State	Threshold Level (packets/second)	Attack Possibility
Normal	0-1	0
Abnormal	2-3	20
Warning	4-10	40
Attack	11-25	60
Critical attack	26-44	80
Critical attack	>45	100

Table 16: Threshold level of Ping-of-Death Threshold 2

Severity State	Threshold Level (packets/second)	Attack Possibility
Normal	0-3	0
Abnormal	4-10	20
Warning	11-25	40
Attack	26-30	60
Critical attack	31-44	80
Critical attack	>45	100

Table 17: Threshold level of Ping-of-Death Threshold 3

Severity State	Threshold Level (packets/second)	Attack Possibility
Normal	0-1	0
Abnormal	2-3	20
Warning	4-10	40
Attack	11-25	60
Critical attack	26-29	80
Critical attack	>30	100

Ping-of-Death Testing

This first testing, the victim receives ping packets size of 65500 bytes continuously. The victim also keeps replying these pings by sending ICMP reply packet back with the same size. The results of detection are shown in the figures below.

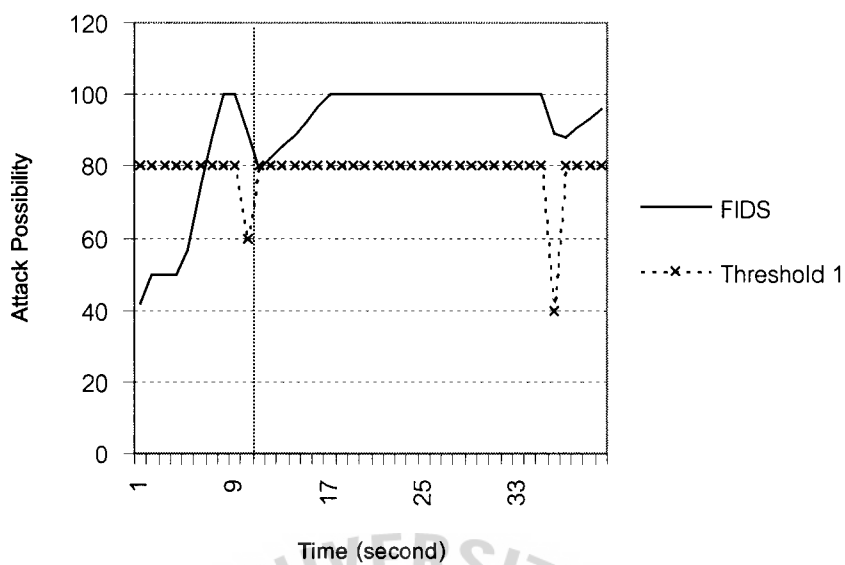


Figure 49a: The detection result of Ping-of-Death testing, FIDS vs. Threshold 1

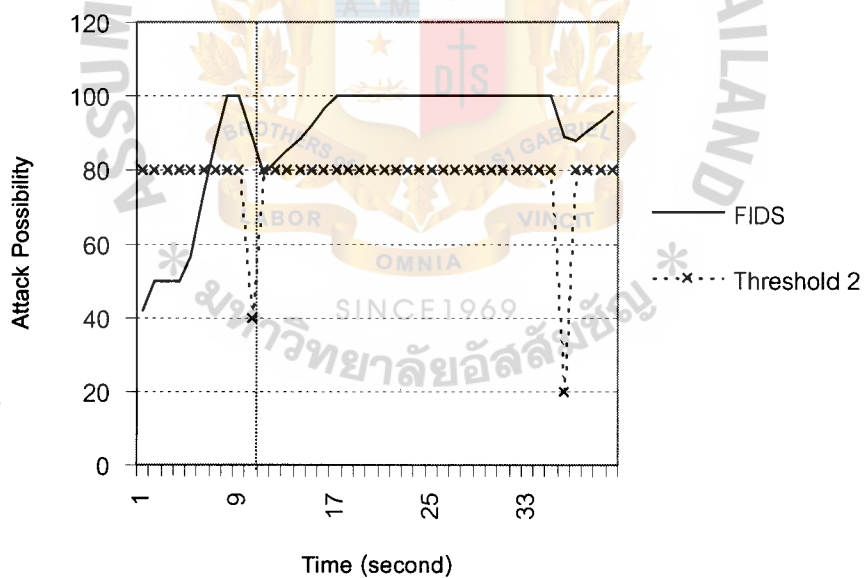


Figure 49b: The detection result of Ping-of-Death testing, FIDS vs. Threshold 2



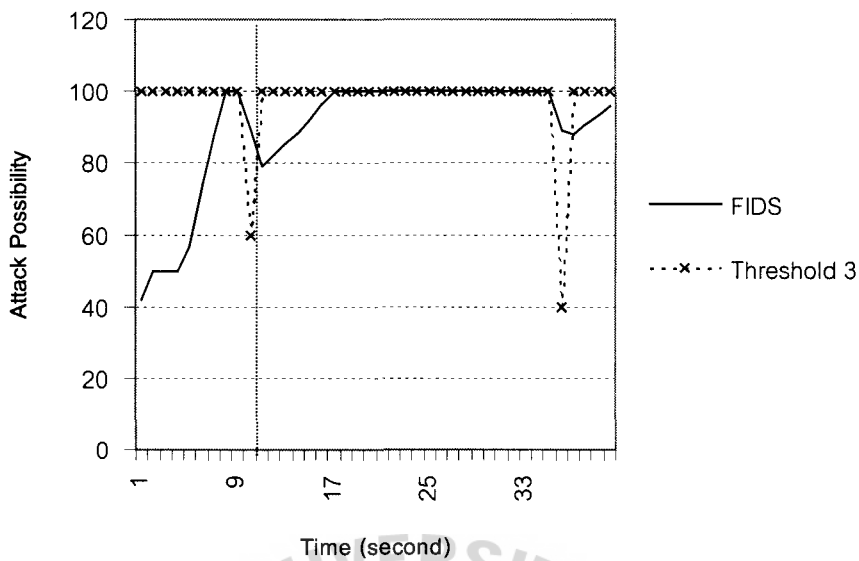


Figure 49c: The detection result of Ping-of-Death testing, FIDS vs. Threshold 3

Since the IP packet is able to contain 1,500 bytes of data, the 65500 bytes of ping will be fragmented into approximately 44 fragments. Thereafter the victim receives these fragments, the victim also reply 44 fragments of ICMP reply packet in one second. Due to the characteristics of this attack, FIDS gives better result than Threshold 1 and Threshold 2. The FIDS can detect the attack as critical attack state within 10 seconds when compared with the rule-setting criteria because the victim has continuously sent ICMP reply fragment almost 45 fragments every second. However, IFDS and Threshold 3 also give the similar results.

Even though the ICMP reply fragments drop in some seconds, as shown Figure 49c, the FIDS result is not effected much like Threshold 3 result. The dropping of ICMP reply fragments may cause by the loss of packet or delay.

4. Email Bomb Testing and Analysis

To test Email Bomb detector, the automatic bomber, called Quikfyre, was used to develop email bomb attack by flooding a victim (mail server) with multiple emails to the same person’s mailbox over and over again. Several tests have been conducted under different amount of email per second. The following figures show the detection result of FIDS versus three threshold-based detectors and the rule-setting criteria. Due to the rule-setting criteria, the FIDS should be able to detect Email Bomb attack when the victim has received more than 50 mails directed to the same recipient within 30 minutes. Moreover, if the severe attack occurs then the FIDS should detect this attack within 30 minutes.

The threshold levels are set based on the number of incoming email (directed to the same person’s mailbox) in 3 minutes and these threshold levels also relate to the way that email frequency membership function has been adjusted. The following tables show the threshold levels of each threshold-based detector.

Table 18: Threshold level of email bomb Threshold 1

Severity State	Threshold Level (packets/second)	Attack Possibility
Normal	0-5	0
Abnormal	6-18	20
Warning	19-33	40
Attack	34-65	60
Critical attack	66-200	80
Critical attack	>200	100

Table 19: Threshold level of email bomb Threshold 2

Severity State	Threshold Level (packets/second)	Attack Possibility
Normal	0-3	0
Abnormal	4-10	20
Warning	11-19	40
Attack	20-47	60
Critical attack	48-120	80
Critical attack	>120	100

Table 20: Threshold level of email bomb Threshold 3

Severity State	Threshold Level (packets/second)	Attack Possibility
Normal	0-2	0
Abnormal	3-5	20
Warning	6-20	40
Attack	21-50	60
Critical attack	51-150	80
Critical attack	>150	100

1<sup>st</sup> Email bomb Testing

The first email bomb testing, the attacker floods the victim (mail server) a burst of emails directed to the same person, every 30 minutes. Then the results of detection are shown in the following figures.

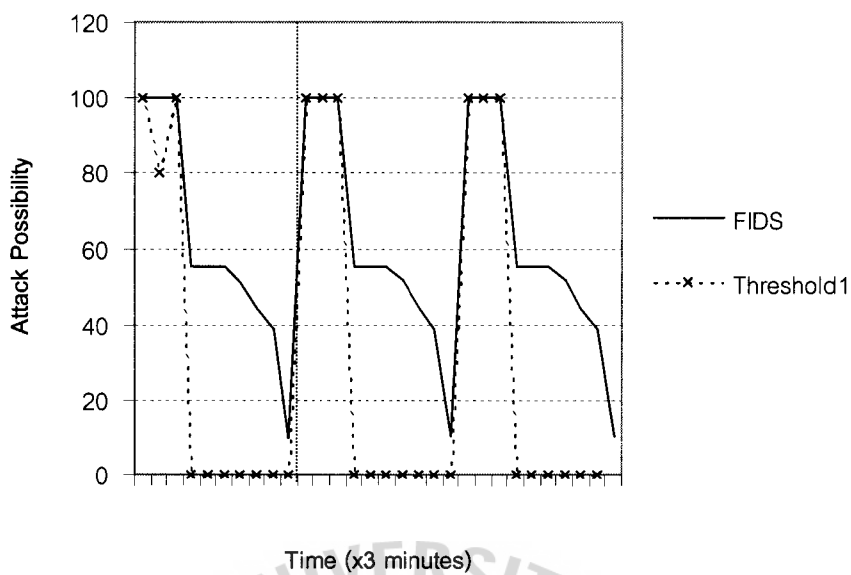


Figure 50a: The detection result of 1<sup>st</sup> email bomb testing, FIDS vs. Threshold 1

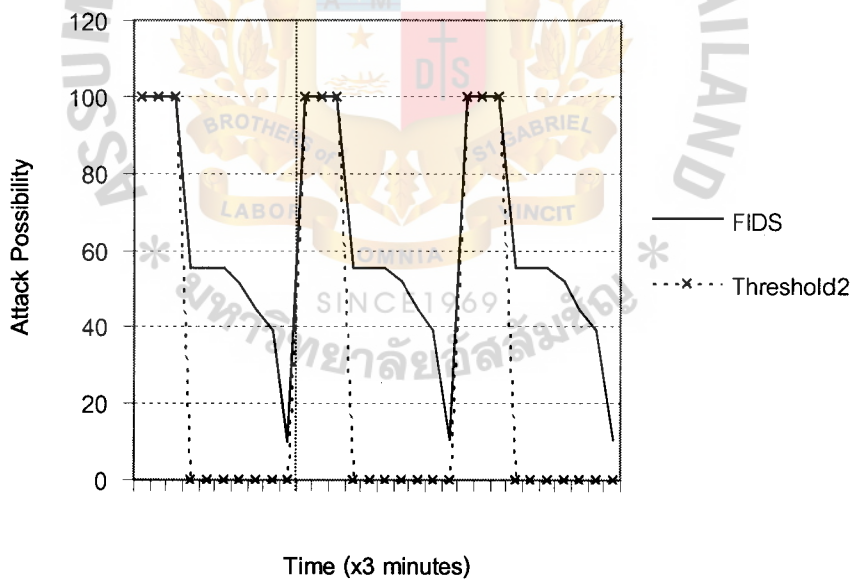


Figure 50b: The detection result of 1<sup>st</sup> email bomb testing, FIDS vs. Threshold 2

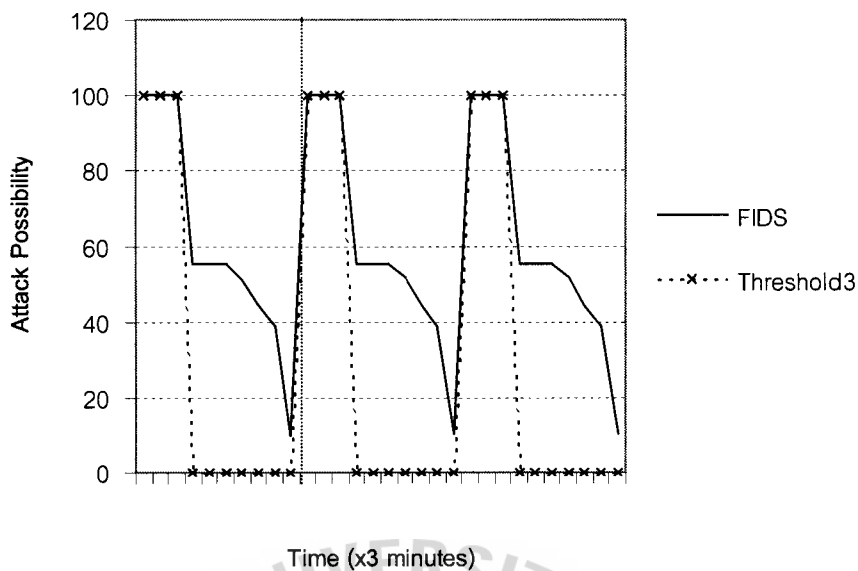


Figure 50c: The detection result of 1<sup>st</sup> email bomb testing, FIDS vs. Threshold 3

The results of detection are almost the same as shown in Figure 50a, 50b and 50c. All of them can detect the attack within 30 minutes as compared with the rule-setting criteria because the number of incoming mails is extremely high (more than 50 mails in 30 minutes). However, the FIDS result is gradually decreased.

This event is very dangerous for mail server since the flood of multiple mails may hang up the server or degrade the performance of server to perform mailing activities.

## 2<sup>nd</sup> Email Bomb Testing

The second email bomb testing, the attacker continuously sends the victim (mail server) small amount of email directed to the same person, 10 mails every 3 minutes. Then the results of detection are shown in the following figures.

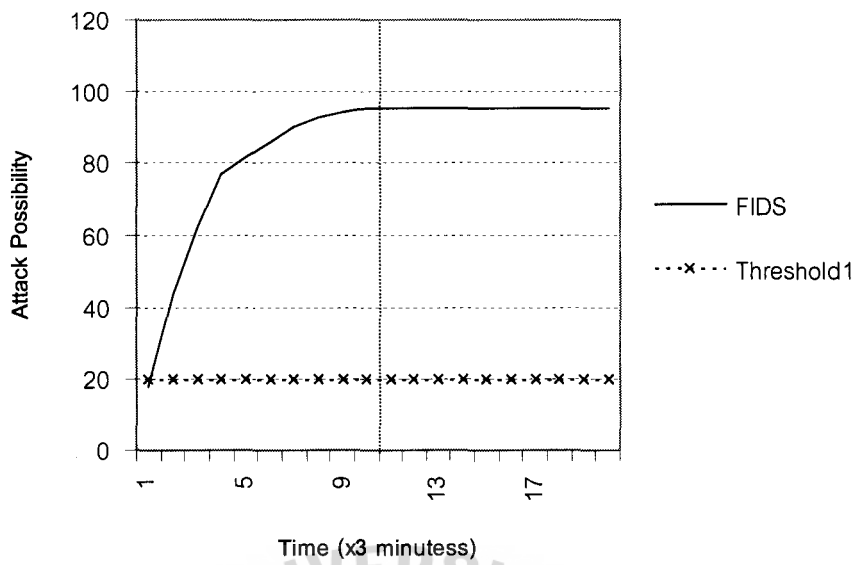


Figure 51a: The detection result of 2<sup>nd</sup> email bomb testing, FIDS vs. Threshold 1

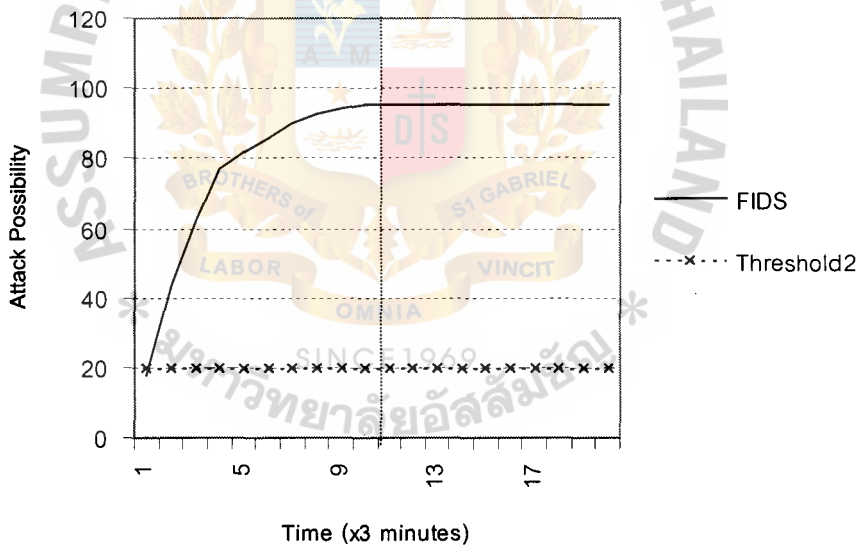


Figure 51b: The detection result of 2<sup>nd</sup> email bomb testing, FIDS vs. Threshold 2

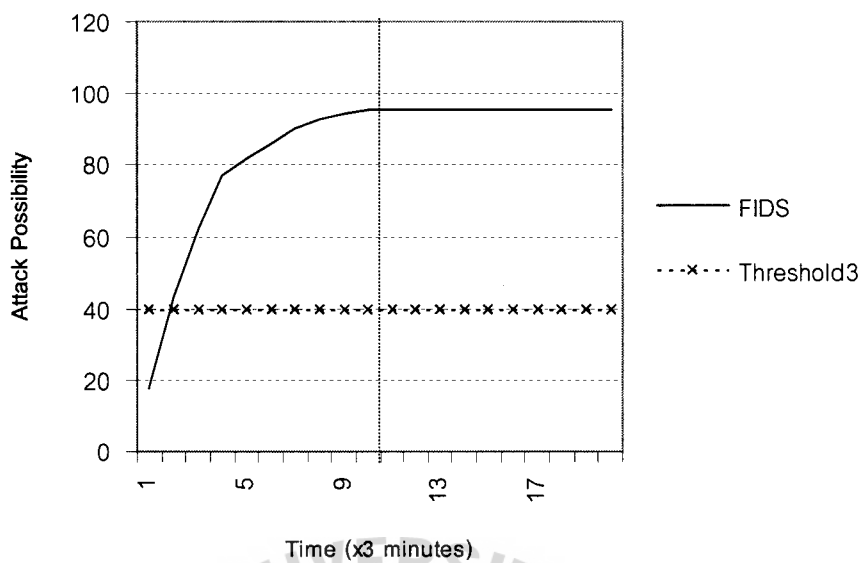


Figure 51c: The detection result of 2<sup>nd</sup> email bomb testing, FIDS vs. Threshold 3

Most mail servers can be configured to protect themselves from email bomb by preventing to receive the large amount of mails from same sender/same source mail server within short period of time. However the prevention might not be able to detect the email bomb that the bomber continuously sends few numbers of mails to fill up someone's mailbox.

Notwithstanding, FIDS could be able to detect this attack event as the result is shown in Figure 51a, 51b and 51c. The FIDS attack severity keeps increasing every period from normal state to critical attack state while the results of Threshold 1 and Threshold 2 are always in abnormal state (with 20 of attack severity). Therefore the FIDS can detect the attack as severe attack within 30 minutes when compared with the rule-setting criteria because the victim has received 50 mails directed to the same recipient in 30 minutes.



5. FTP and Telnet Password Guessing Analysis

To test the FTP and Telnet Password guessing detector, automatic guessing program, named BrutusA2, is used to guess the password. By using automatic password guessing programs, there are several attempts of a making connection with a server at port 21 (FTP) and port 23 (Telnet). Tests have been conducted with different number of guessing connections simultaneously.

FTP password guessing

The following figures show the detection result of FIDS versus three threshold-based detectors and the rule-setting criteria. Due to the rule-setting criteria, the FIDS should be able to detect FTP password guessing attack when the victim has been guessed more than 15 time per second for the password. Moreover, if the severe attack occurs then the FIDS should detect this attack within 10 seconds. The threshold levels are set based on the number of attempts (number of packet contains login incorrect data) per second and these threshold levels also relate to the way that FTP login incorrect packet frequency membership function has been adjusted. The following tables show the threshold levels of each threshold detector.

Table 21: Threshold level of FTP Password Guessing Threshold 1

Severity State	Threshold Level (packets/second)	Attack Possibility
Normal	0-3	0
Abnormal	4-6	20
Warning	7-9	40
Attack	10-12	60
Critical attack	13-15	80
Critical attack	>15	100

Table 22: Threshold level of FTP Password Guessing Threshold 2

Severity State	Threshold Level (packets/second)	Attack Possibility
Normal	0-3	0
Abnormal	4-10	20
Warning	11-15	40
Attack	16-24	60
Critical attack	25-33	80
Critical attack	> 33	100

Table 23: Threshold level of FTP Password Guessing Threshold 3

Severity State	Threshold Level (packets/second)	Attack Possibility
Normal	0-10	0
Abnormal	11-15	20
Warning	16-21	40
Attack	22-27	60
Critical attack	28-33	80
Critical attack	>33	100

The following figure shows the testing result of FTP Password Guessing Detector, when the victim has been guessed by using one connection. Because there are several attempts of guessing per second and the attacker should perform guessing over long period of time, therefore the FIDS gives the better result of attack severity. The FIDS can detect the attack as critical attack state within 10 seconds when compared with the rule-setting criteria because the victim has been continuously guessed for the password, even though there are approximately 10 attempts per second.

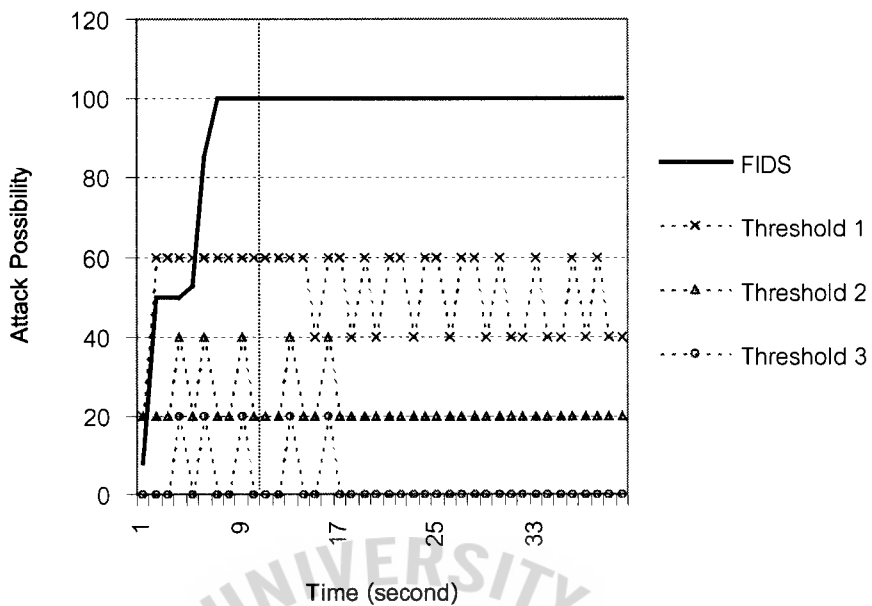


Figure 52: The detection result of 2<sup>nd</sup> FTP password guessing testing

#### Telnet password guessing

Next, the following figures also show the detection results of FIDS versus two threshold-based detectors when the victim has been guessed Telnet password. Due to the rule-setting criteria, the FIDS should be able to detect Telnet password guessing attack when the victim has been guessed more than 3 times per second for the password. Moreover, if the severe attack occurs then the FIDS should detect this attack within 10 seconds.

The threshold levels are set based on the number of attempts (number of packet contains login incorrect data) per second and these threshold levels also relate to the way that Telnet login incorrect packet frequency membership function has been adjusted. The following tables show the threshold levels of each threshold detector.

Table 24: Threshold level of Telnet Password Guessing Threshold 1

Severity State	Threshold Level (packets/second)	Attack Possibility
Normal	0	0
Abnormal	1	20
Warning	2	40
Attack	3	60
Critical attack	4	80
Critical attack	>4	100

Table 25: Threshold level of Telnet Password Guessing Threshold 2

Severity State	Threshold Level (packets/second)	Attack Possibility
Normal	0	0
Abnormal	-	20
Warning	1	40
Attack	2	60
Critical attack	3	80
Critical attack	> 3	100

1<sup>st</sup> Telnet Password Guessing Testing

The first Telnet Password Guessing testing, the attacker attempts to guess Telnet password with one connection. The detection results are shown in the figures below.

According to the results in Figure 53a, Threshold 1 and FIDS give the similar result. Meanwhile FIDS gives the worse result when compared to Threshold 2 because the number of attempts is too small, see Figure 53b. However the reason behind this result is that the attacker cannot figure out the password within a century. Therefore the result remains in normal state.

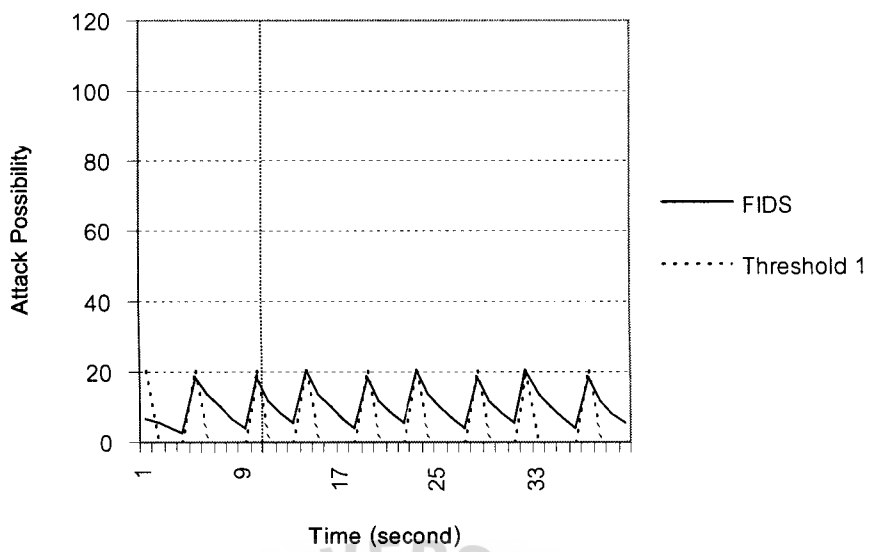


Figure 53a: The detection result of 1<sup>st</sup> Telnet password guessing testing,  
FIDS vs. Threshold 1

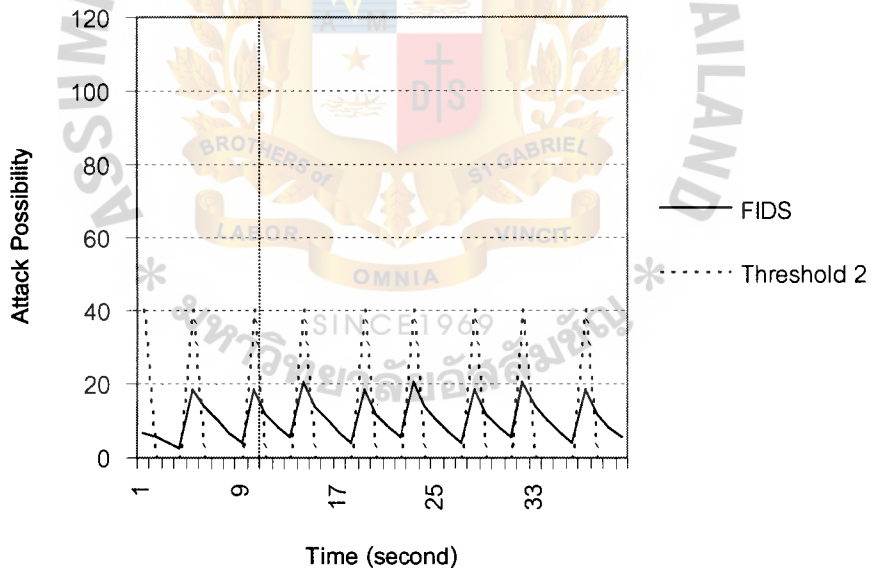


Figure 53b: The detection result of 1<sup>st</sup> Telnet password guessing testing,  
FIDS vs. Threshold 2

2<sup>nd</sup> Telnet Password Guessing Testing

This Telnet Password Guessing testing, the attacker attempts to guess Telnet password with three connections simultaneously. The detection results are shown below.

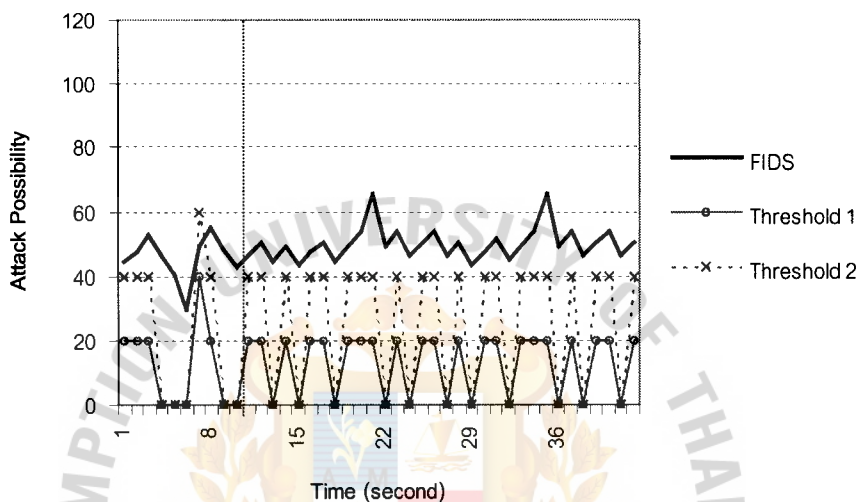


Figure 54: The detection result of 2<sup>nd</sup> Telnet password guessing testing

This case of guessing, FIDS yields better result compared to Threshold 1 and Threshold 2 as shown in Figure 54. Because the number of attempts per second increases, the attacker may have a few chances to get to password. Using FIDS can detect this event as warning state within 10 seconds when compared with the rule-setting criteria because the victim is continuously guessed for the password, even though the number of attempts just 1 or 2 times per seconds. Meanwhile using Threshold 1 and Threshold 2, the detection results are in normal and abnormal state.

6. Port Scanning Testing and Analysis

Actually, Port Scanning is not an attack but it is the starting port of attacking. Since the hacker tries to check the victims’ services. Thereafter, the attacker may try to compromise these services. Therefore, it is so difficult to check that the victim had been scanned. To test Port Scanning Detector, the scanner program, named SuperScan 2.06, is used to scan the victim’s services with the maximum speed of scanning. The results as shown in Figure 26a and 26b. The results are compared with the threshold-based detectors’ result and rule-setting criteria. Due to the rule-setting criteria, the FIDS should be able to Port Scanning attack when the victim has been port-scanned more than 24 times per second.

The following tables show the threshold levels of two threshold-based detectors. The threshold levels are set on the basis of the number of SYN-RESET pair per second and these threshold levels also relate to the way that SYN-RESET pair frequency membership function has been adjusted.

Table 26: Threshold level of port scanning Threshold 1

Severity State	Threshold Level (packets/second)	Attack Possibility
Normal	0-5	0
Abnormal	6-13	20
Warning	14-17	40
Attack	18-21	60
Critical attack	22-24	80
Critical attack	>24	100



Table 27: Threshold level of port scanning Threshold 2

Severity State	Threshold Level (packets/second)	Attack Possibility
Normal	0-5	0
Abnormal	6-10	20
Warning	11-15	40
Attack	16-20	60
Critical attack	21-25	80
Critical attack	> 25	100

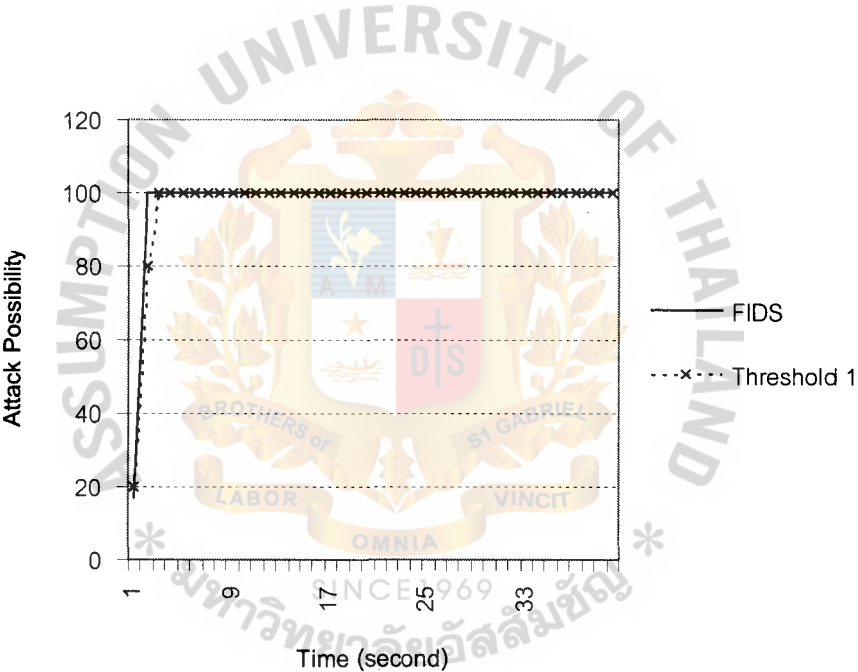


Figure 55a: The detection result of port scanning testing, FIDS vs. Threshold 1

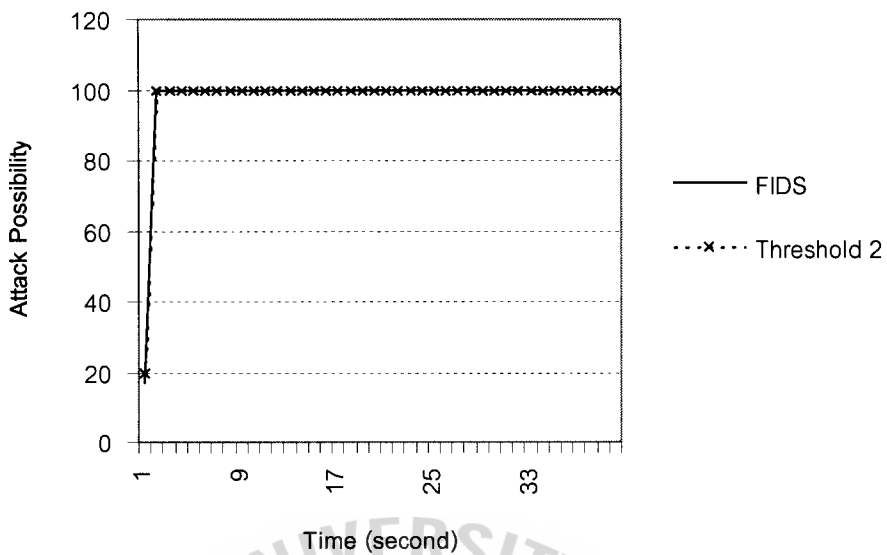


Figure 55b: The detection result of port scanning testing, FIDS vs. Threshold 2

According to the results of port scanning attack shown in Figure 55a and 55b, FIDS, Threshold 1 and Threshold 2 give the similar result because Port Scanning detector has only one fuzzy box. It does not take the weighted accumulative into account, since victim can be scanned at any time. It is not necessary to scan continuously.

### Summary

To detect to Denial of Services (SYN Flood, UDP flood, Ping-of-Death and Email Bomb attack) and password guessing by using FIDS yields the better results than Threshold-based detection in many cases and the FIDS can detect the attacks with the specified period when compared to the rule-setting criteria. In addition, the FIDS take two important parameters (amount of packets and the weight accumulative) into account, therefore the FIDS provides more accuracy when detecting the intrusive behaviors.

## **CHAPTER 5: CONCLUSIONS AND RECOMMENDATION FOR FURTHER RESEARCH**

### **5.1 Conclusions**

This thesis presents a framework for a Fuzzy Intrusion Detection System (FIDS) utilizing a fuzzy rule-based system in order to detect intrusive network traffic. The use of the fuzzy rule-based system has made FIDS be able to detect intrusive traffic more flexibly than the one that uses threshold-based detection. Rather than using sharp boundary between normal and intrusive traffic for the decision, the FIDS considers both current traffic level and the weighted accumulative number, number of intrusive traffic during past seconds. Consequently, the FIDS can provide better detection results than other threshold-based detection. FIDS yields a better result than the threshold-based detector even though the number of intrusive packets in some seconds drops. Even if this number is lower than attack-state or critical-state threshold level (attack possibility is lower than 60 and 80) in case of threshold-based detector, the FIDS may detect this kind of intrusive pattern as attack or critical attack when there are also intrusive traffic during past seconds.

Even if the FIDS framework is designed based on the specific environment, the detection rules of FIDS are more flexible to be applied on other network environments. Since this FIDS variables' membership functions are tuned based on normal and abnormal datasets of specific network, to apply this framework with another network, the normal and

abnormal datasets of new network environment should be collected, mined and studied for determining the traffic pattern. Then only FIDS variables' membership function, packet frequency of each detector, should be readjusted again for dealing with another network environment. This means that the LEVEL BOX input variables of each detector, packet frequency, should be tuned once again to yield the appropriate normalized number, traffic level. Thereafter, this traffic level can work with FIDS DETECTOR BOX.

## **5.2 Recommendation for further research**

This thesis can be further extended to provide firstly a self-adjustable FIDS. As mentioned above, to be applied with another network, LEVEL BOX input variable (packet frequency) should be readjusted again manually by the system administrator. Hence, this manual tuning may take some time and be inconvenient. Therefore, to become the self-adjustable FIDS will be the efficient detector. When the self-adjustable FIDS is implemented on another network, the traffic frequency's membership function of each detector will be automatically tuned to yield the suitable normalized number, traffic level. Then this traffic level will be further used by DETECTOR BOX.

Secondly, this thesis can be extended to provide a warning and response system. Then the FIDS will be a complete suit of intrusion detection and prevention system. Thereafter the Fuzzy Rule-Based Detector module detects the intrusion, then the warning and response

system will alert the system administrator about these attacks and it will also issue the appropriate responses such as disconnecting the connection or blocking the traffic.



## REFERENCES

- [1] Wenke Lee, Salvatore J. Stolfo, and Kui W. Mok, "Data Mining Approaches for Intrusion Detection," Proc. 7th USENIX security symposium, 1998.
- [2] Jose M. Bonifacio Jr., Adriano M. Cansian, Andre C. P. L. F. de Carvalho, Edson S. Moreira, "Neural Networks Applied in Intrusion Detection Systems," IEEE WCCI'98, pg. 205-210, 1998.
- [3] S. Forrest, S. A. Hofmeyr, A. Somayaji, T. A. Longstaff, "A Sense of Self for Unix Processes," Proc. IEEE Symposium, pg. 120-128, 1996.
- [4] Garber Lee, "Computer Innovative Technology for Computer Professionals," *In magazine IEEE Computer Security*, April 2000, volume 33 number 4, pages 12 – 17, 2000.
- [5] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," Proc. ACM SIGMOD, pages 207-216, 1993.
- [6] Wenke Lee, Salvatore J. Stolfo, Kui W. Mok. "Algorithms for Mining System Audit Data," 1998.
- [7] Steven R. Snapp, James Brentano, Gihan V. Dias, Terrance L. Goan, Tim Grance, L. Todd Heberlein, Che-Lin Ho, Karl N. Levitt, Biswanath Mukherjee, Douglass L. Mansur, Kenneth L. Pon, and Stephen E. Smaha. "A system for distributed intrusion detection," In *COMPCOM Spring '91 Digest of Papers*, pages 170-176, February/March 1991.
- [8] Teresa F. Lunt, R. Jagannathan, Rosanna Lee, Alan Whitehurst, and Sherry Listgarten. "Knowledge based Intrusion Detection," In Proceedings of the Annual AI Systems in Government Conference, Washington, DC, March 1989.
- [9] Adam L. Rice, "Defending Network from SYN Flooding Attack," [http://www.sans.org/infosecFAQ/threats/SYN\\_flood.htm](http://www.sans.org/infosecFAQ/threats/SYN_flood.htm).
- [10] T. Lunt, A. Tamaru, F. Gilham, R. Jagannathan, P. Neumann, H. Javitz, A. Valdes, and T. Garvey, "A Real-Time Intrusion Detection Expert System (IDES) – final technical report," Technical report, Computer Science Laboratory, SRI International, California, February 1992.
- [11] Biswanath Mukherjee, L. Todd Heberlein, Karl N. Levitt, "Network Intrusion Detection," IEEE Network, May/June 1994.

- [12] Karanjit S. Siyan, Inside TCP/IP, 3<sup>rd</sup> Edition. New Riders Publishing, 1997.
- [13] Cox Earl, The fuzzy system handbook: a practitioner's guide to building, using, and maintaining fuzzy system. AP Professional, 1994.





