

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ชุดโปรแกรมเน็ตเวิร์กไฟร์วอลล์

NETWORK FIREWALL PROGRAM SUITE



ปฏิญานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง พ.ศ.2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุดโปรแกรมเน็ตเวิร์กไฟร์วอลล์  
NETWORK FIREWALL PROGRAM SUITE

นายทวีศักดิ์	เมฆศิขริน	รหัส 45010295
นายพิพัฒน์	ประภาพรรณพงศ์	รหัส 45010535
นายพีรพงศ์	วงศ์วิวัฒน์กิจ	รหัส 45010554



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
พ.ศ.2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโททางการศึกษา 2548

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ชุดโปรแกรมเน็ตเวิร์กไฟร์วอลล์

Network Firewall Program Suite

ผู้จัดทำ

- |                              |              |          |
|------------------------------|--------------|----------|
| 1. นายทวิศักดิ์ เมฆศิขริน    | รหัสประจำตัว | 45010295 |
| 2. นายพิพัฒน์ ประภาพรรณพงศ์  | รหัสประจำตัว | 45010535 |
| 3. นายพีรพงศ์ วงศ์วิวัฒน์กิจ | รหัสประจำตัว | 45010554 |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ชุดโปรแกรมเน็ตเวิร์กไฟร์วอลล์

นายทวีศักดิ์ เมฆศิขริน	45010295
นายพิพัฒน์ ประภาพรรณพงศ์	45010535
นายพีรพงศ์ วงศ์วิวัฒน์กิจ	45010554
อ. อัครเดช วัชรระภูพงษ์	อาจารย์ที่ปรึกษา
ผศ. ธนา หงษ์สุวรรณ	อาจารย์ที่ปรึกษาร่วม
อ. ธนัญชัย ศรีภาค	อาจารย์ที่ปรึกษาร่วม
ปีการศึกษา 2548	

### บทคัดย่อ

ในปัจจุบันนี้ระบบรักษาความปลอดภัยของคอมพิวเตอร์และระบบเครือข่ายมีส่วนสำคัญขึ้นเรื่อยๆ เป็นลำดับ โดยเฉพาะอย่างยิ่งกับองค์กรทางธุรกิจทั้งหลาย สิ่งแรกๆ ที่นึกถึงคงจะหนีไม่พ้นไฟร์วอลล์ ด้วยเป็นอุปกรณ์รักษาความปลอดภัยทางเครือข่ายที่มีความจำเป็นเป็นอันดับต้นๆ อย่างไรก็ตาม ไฟร์วอลล์ยังคงแบ่งออกเป็นหลายประเภทและมีคุณสมบัตินี้ความสามารถแตกต่างกันออกไป ไฟร์วอลล์ชนิดใดชนิดหนึ่งไม่สามารถทำหน้าที่ครอบคลุมได้ทุกด้านอย่างแน่นอน

วิทยานิพนธ์ฉบับนี้เสนอการพัฒนาชุด โปรแกรมเน็ตเวิร์กไฟร์วอลล์ขึ้นเพื่อใช้กับระบบเครือข่ายขนาดกลาง ซึ่งโครงการนี้ได้ทำการรวบรวมและปรับปรุงความสามารถของไฟร์วอลล์ชนิดต่างให้สามารถใช้งานได้เหมาะสมและเต็มประสิทธิภาพ โครงการนี้มุ่งเน้นไปที่การทำงานร่วมกัน และสามารถควบคุมไฟร์วอลล์บนส่วนต่างๆ ทั้งหมดได้จากศูนย์กลาง คือสามารถกำหนดกฎให้กับไฟร์วอลล์ส่วนต่างๆ และรวมถึงโปรแกรมสำหรับคู่มือที่ส่งกลับมาจากไฟร์วอลล์ทุกๆ ส่วน อันได้แก่ไฟร์วอลล์ส่วนบุคคลที่มีลักษณะเป็นแบบอิงผู้ใช้พร้อมระบบตรวจจับผู้บุกรุก เกดเวย์ไฟร์วอลล์ซึ่งทำงานอยู่บนเกตเวย์ และสุดท้ายคือฟร็อกซีไฟร์วอลล์ซึ่งคือซีเคียวริตี้เว็บฟร็อกซี

## Network Firewall Program Suite

Mr. Taweesak Meksikarin	45010295
Mr. Pipat Prapapanpong	45010535
Mr. Pirapong Wongvivatkit	45010554
Mr. Akkradach Watcharapupong	Advisor
Asst.Prof. Thana Hongsuwan	Co-Advisor
Mr. Thanunchai Threepak	Co-Advisor

Academic Year 2005

### ABSTRACT

Nowadays, the computer and network security has become increasingly more and more important. Especially with the organizations that their information sensitive so that they have to keep those safe all the time. The most popular security system everyone first thinking of is "Firewall" as it is indeed one of the must-have security system out there. However, there are various types of firewalls that have different abilities. Therefore, in order to get the highest efficiency, we have to combine those all to work together. Only one type of firewall cannot do the job.

This thesis proposes the development of the "Network Firewall Program Suite" suitable for a medium-size organization. The project has integrated different types of firewalls to make the most secure system out of their various abilities. The project is mainly aiming at the centralization of the system. The entire system can be managed from the single central server. The management mentioned consists of *defining firewall rules for every firewall in the system and receiving logs from those as well*. The different types of firewalls in this system include a user-based personal firewall with an intrusion detection system, a gateway firewall and a secure reverse web proxy. *These should be put together to ensure the real security.*

## กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้จะไม่สามารถสำเร็จสมบูรณ์ได้ถ้าไม่ได้รับความช่วยเหลือ และความ  
ร่วมมือของบุคคลหลาย ๆ ฝ่ายด้วยกัน โดยเฉพาะอย่างยิ่งบุคคลผู้ซึ่งเป็นผู้จุดประกายความคิดให้เกิด  
หัวข้อโครงการนี้ขึ้น นั่นคือท่าน อาจารย์อิศเรศ วัชรภพพงษ์ ซึ่งเป็นอาจารย์ที่ปรึกษา รวมทั้งท่าน  
อาจารย์ทุกท่านในภาควิชา วิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระ  
จอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้ให้การอบรมสั่งสอน และให้วิชาความรู้ต่าง ๆ ที่ดีแก่คณะ  
ผู้จัดทำเสมอมา

ขอขอบพระคุณท่านที่สำคัญที่สุดในชีวิตนี้ ที่ได้ให้กำเนิด ให้การอบรมดูแล และเอาใจใส่  
ทั้งด้านการศึกษา ด้านการดำเนินชีวิต และด้านอื่นทุกด้าน ที่คงไม่อาจมีใครอีกแล้ว ที่เสมอเหมือน  
ท่านทั้งสองนี้ นั่นคือ บิดา และ มารดา ผู้ซึ่งเป็นที่เคารพรักอย่างยิ่ง ผู้ซึ่งคอยให้กำลังใจในยามที่ไม่มี  
ใครเหลือ ผู้ซึ่งคอยชี้แนวทางที่ถูกเสมอ ผู้ที่ให้การสนับสนุนในการทำสิ่งที่ถูก และให้กำลังใจหากสิ่ง  
ที่เราทำไม่ใช่สิ่งที่ดี ไม่ใช่สิ่งที่ควร จึงขอกราบขอบพระคุณมา ณ ที่นี้

สุดท้ายนี้ ขอขอบคุณภาควิชาวิศวกรรมคอมพิวเตอร์ โดยเฉพาะห้องวิจัย และพัฒนาการ  
รักษาความปลอดภัยข้อมูล (ISAG) และ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่  
ได้เอื้อเฟื้อสถานที่ ให้คณะผู้จัดทำได้ทำการวิจัย และช่วยอำนวยความสะดวกต่าง ๆ ขอขอบคุณ  
เพื่อน ๆ พี่ ๆ น้อง ๆ ชาว ISAG ที่คอยให้ความช่วยเหลือ ในการทำงาน ตลอดเวลา ทางคณะผู้จัดทำ  
ขอขอบพระคุณมา ณ ที่นี้ด้วย

นายทวีศักดิ์ เหมศิริกริน  
นายพิพัฒน์ ประภาพรรณพงษ์  
นายพีรพงศ์ วงศ์วิวัฒน์กิจ

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูป.....	VIII
บทที่ 1 บทนำ.....	1
1.1 ความสำคัญและที่มาของ โครงการ.....	1
1.2 วัตถุประสงค์ของ โครงการ.....	1
1.3 ขอบเขตของ โครงการ.....	1
1.4 วิธีการดำเนินการ.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
1.6 ส่วนประกอบของปริณยานิพนธ์.....	3
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....	4
2.1 ไฟร์วอลล์.....	4
2.2 เทคโนโลยีแอคทีฟไดเรกทอรี.....	6
2.3 Netfilter/IPTABLES.....	10
2.4 Proxy.....	15
2.5 ระบบตรวจจับผู้บุกรุกทางเครือข่าย(Intrusion Detection System).....	18
บทที่ 3 การออกแบบ โครงสร้างและการพัฒนาระบบ.....	24
3.1 แนวคิด.....	24
3.2 โครงสร้าง.....	26
บทที่ 4 การพัฒนาไฟร์สแตชัน.....	27
4.1 แนวคิด.....	27
4.2 ขอบเขตและความสามารถ.....	27
4.3 การพัฒนาโปรแกรม.....	27
4.4 การทำงานของโปรแกรม.....	30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

	หน้า
บทที่ 5 การพัฒนาโปรแกรมไพร์ออลาร์ม .....	31
5.1 แนวคิด .....	31
5.2 ขอบเขตและความสามารถ .....	31
5.3 การพัฒนาโปรแกรม .....	31
5.4 การทำงานของโปรแกรม .....	32
บทที่ 6 การพัฒนาโปรแกรมไพร์สกรีน .....	36
6.1 แนวคิด .....	36
6.2 ขอบเขตและความสามารถ .....	36
6.3 การพัฒนาโปรแกรม .....	36
6.3.1 การทำงานร่วมกับไพร์สเดชั่น .....	36
6.3.2 การติดต่อกับแอ็คทีฟไดเรกทอรี .....	37
6.3.3 การค้นหาข้อมูลในแอ็คทีฟไดเรกทอรี .....	39
6.3.4 การเปลี่ยนแปลงแก้ไขข้อมูลบนแอ็คทีฟไดเรกทอรี .....	42
6.3.5 การนำกฎจากไพร์สเดชั่นมาใช้บนไพร์สกรีน .....	44
6.3.5.1 การดึงกฎจากไพร์สเดชั่น .....	44
6.3.5.2 การนำกฎมาใช้กับไอพีเทเบิลส์ .....	45
6.3.6 การส่งสื่อกลับไปยังไพร์สเดชั่น .....	46
6.4 การทำงานของโปรแกรม .....	48
บทที่ 7 การพัฒนาโปรแกรมไพร์เบรก .....	52
7.1 แนวคิด .....	52
7.2 ขอบเขตและความสามารถ .....	52
7.3 การพัฒนาโปรแกรม .....	52
7.3.1 โปรแกรม Squid .....	53
7.3.2 โปรแกรม Snort .....	54
7.4 การทำงานของโปรแกรม .....	55

## สารบัญ (ต่อ)

	หน้า
บทที่ 8 การพัฒนาโปรแกรมสื่อคอมพิวเตอร์ .....	57
8.1 แนวคิด .....	57
8.2 ขอบเขตและความสามารถ .....	57
8.3 การพัฒนาโปรแกรม .....	57
8.3.1 สื่อคอมพิวเตอร์ของไฟร์ลาร์ม .....	57
8.3.2 สื่อคอมพิวเตอร์ของไฟร์สกรีน .....	58
8.3.3 สื่อคอมพิวเตอร์ของไฟร์เบรก .....	59
8.4 การทำงานของโปรแกรม .....	59
บทที่ 9 การทดสอบการทำงาน .....	61
9.1 ระบบที่ใช้ในการทดสอบ .....	61
9.2 โครงสร้างของระบบที่ใช้ในการทดสอบ .....	62
9.3 การทดสอบการทำงาน .....	62
9.3.1 ทดสอบการทำงานของโปรแกรมไฟร์สเดชั่น .....	63
9.3.2 ทดสอบการทำงานของโปรแกรมไฟร์ลาร์ม .....	65
9.3.3 ทดสอบการทำงานของโปรแกรมไฟร์สกรีน .....	67
9.3.4 ทดสอบการทำงานของโปรแกรมไฟร์เบรก .....	70
9.3.5 ทดสอบการทำงานของโปรแกรมสื่อคอมพิวเตอร์ไฟร์ลาร์ม .....	72
9.3.6 ทดสอบการทำงานของโปรแกรมสื่อคอมพิวเตอร์ไฟร์สกรีน .....	72
9.3.7 ทดสอบการทำงานของโปรแกรมสื่อคอมพิวเตอร์ไฟร์เบรก .....	74
บทที่ 10 บทสรุปและวิจารณ์ .....	75
10.1 บทสรุป .....	75
10.2 วิจารณ์สิ่งที่ได้จากโครงงาน .....	75
10.3 ปัญหาอุปสรรคในการพัฒนา .....	76
10.4 แนวทางการพัฒนาต่อ .....	76
บรรณานุกรม .....	77

# สารบัญตาราง

ตารางที่	หน้า
2.1 แสดงรูปแบบของแอคเซสรูทเบื้องต้น.....	4
2.2 ตัวอย่างของการตั้งค่าแอคเซสรูทของไฟร์วอลล์.....	5
2.3 แสดงโครงสร้างการเก็บข้อมูลของ Fragment Buffer.....	20
2.4 แสดงโครงสร้างการเก็บข้อมูลของ Fragment.....	21
4.1 รายละเอียดของแอตทริบิวต์ที่เพิ่มในฐานข้อมูลของแอคทีฟไดเรกทอรี.....	28
4.2 รายละเอียดคลาสที่เพิ่มในฐานข้อมูลของแอคทีฟไดเรกทอรี.....	28
4.3 รายละเอียดการทำงานของฟังก์ชันในคลาส ADConnect.....	30
6.1 พารามิเตอร์ของฟังก์ชัน ldap_init().....	37
6.2 พารามิเตอร์ของฟังก์ชัน ldap_bind_s().....	37
6.3 พารามิเตอร์ของฟังก์ชัน ldap_search_s().....	39
6.4 พารามิเตอร์ของฟังก์ชัน ldap_first_entry().....	40
6.5 พารามิเตอร์ของฟังก์ชัน ldap_first_attribute().....	40
6.6 พารามิเตอร์ของฟังก์ชัน ldap_get_values().....	41
6.7 พารามิเตอร์ของฟังก์ชัน ldap_modify_s().....	42
6.8 แสดงความหมายของแต่ละค่า.....	45
8.1 แสดงโครงสร้างตารางที่ใช้ในการเก็บล็อกของเครื่องลูกข่าย.....	58
8.2 แสดงโครงสร้างตารางที่ใช้ในการเก็บล็อกของเครื่องเกตเวย์.....	58
8.3 แสดงเทเบิลที่สำคัญของดาต้าเบส Soort.....	59
9.1 รายละเอียดของเครื่องคอมพิวเตอร์ต่างๆภายในระบบ.....	61
9.2 แสดงการกำหนดกฎตัวอย่าง.....	63

# สารบัญรูป

รูปที่	หน้า
2.1 แสดงการใช้ Dual-homed Host เป็น พร็อกซีเซิร์ฟเวอร์ (Proxy Server).....	6
2.2 โครงสร้างโดยรวมของแอคทีฟไดเรกทอรีดาต้าเบส (Active Directory Database) .....	7
2.3 การเดินทางของแพ็คเก็ตใน Filter Table .....	11
2.4 แสดงตำแหน่งของเว็บพร็อกซีในเครือข่าย .....	16
2.5 การทำงานของรีเวิร์สเว็บพร็อกซี .....	17
2.6 การทำงานของรีเวิร์สเว็บพร็อกซีแบบปลอดภัย .....	18
2.7 แสดงการเก็บข้อมูลของตัวแปร Tuple .....	20
2.8 แสดงการตรวจสอบความผิดปกติในการทำแฟล็กแมนเดชั่น .....	21
2.9 แสดงการตรวจสอบแพ็คเก็ตที่ส่งแบบวนลู .....	22
2.10 แสดงแผนภูมิแสดงประเภทของการโจมตีเพื่อให้ปีบริการแบบผสม .....	22
3.1 ระบบเครือข่ายที่มีการติดตั้งชุดโปรแกรมเน็ตเวิร์กไฟร์วอลล์ .....	26
4.1 แสดงรูปแบบการเข้าถึง Active Directory Service.....	28
4.2 รายละเอียดของคลาส ADConnect .....	29
4.3 แสดงโครงสร้างการทำงานของไฟร์สเตชัน(FireStation).....	30
5.1 แสดงส่วนประกอบเพอร์ซันนอลไฟร์วอลล์.....	32
5.2 แสดงขั้นตอนการทำงานของไฟร์วอลล์ .....	32
5.3 แสดงระดับชั้นการทำงานของ WinPCap.....	33
5.4 โครงสร้างของระบบตรวจจับผู้บุกรุกทำงานร่วมกับไฟร์วอลล์.....	34
5.5 โครงสร้างของระบบตรวจจับผู้บุกรุกทำงานร่วมกับไฟร์วอลล์.....	35
6.1 แสดงแพ็คเกจที่จำเป็น.....	36
6.2 กฎในเชน LOG-CHAIN .....	45
6.3 แสดงล็อกที่ถูกเก็บไว้โดย syslogd .....	47
6.4 แสดงการรับกฎของไฟร์สกรีน .....	48
6.5 แสดงการส่งล็อกของไฟร์สกรีน .....	49
7.1 แสดงการทำงานของโปรแกรมไฟร์เบรก.....	56
8.1 แสดงโครงสร้างการทำงานระหว่างล็อกมอนิเตอร์กับแอคทีฟไดเรกทอรี .....	60
9.1 โครงสร้างทางเครือข่ายของระบบที่ใช้ในการทดสอบ.....	62
9.2 โปรแกรมไฟร์สเตชัน.....	63
9.3 โปรแกรมไฟร์สเตชันส่วนของการกำหนดกฎให้กับไฟร์สกรีน(FireScreen) .....	64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

## สารบัญรูป (ต่อ)

รูปที่	หน้า
9.4 โปรแกรมไฟร์วอลล์	65
9.5 แสดงกฎการฟิเตอร์ที่รับมาจากเซิร์ฟเวอร์	65
9.6 แสดงสถานะของโปรเซสที่ใช้งานเครือข่าย	66
9.7 แสดงป๊อปอัพ(Popup) แสดงโปรเซสที่ใช้งานเครือข่าย	66
9.8 การแจ้งเตือนการบุกรุกหรือถูกโจมตี	67
9.9 แสดงกฎเริ่มต้นของไฟร์สกรีน	68
9.10 กฎจากไฟร์สแตชันที่เพิ่มเข้าไปในเซิร์ฟเวอร์	69
9.11 กฎจากไฟร์สแตชันที่เพิ่มเข้าไปในเซิร์ฟเวอร์	69
9.12 ล็อกที่แสดงขึ้นที่หน้าจอของไฟร์สกรีน	70
9.13 แสดงข้อมูลใน /var/log/firescreen.log	70
9.14 แสดงการทำงานของไฟร์เบรกขณะเปิดโปรแกรม snort_inline	71
9.15 โปรแกรมล็อกมอนิเตอร์ไฟร์วอลล์	72
9.16 โปรแกรมล็อกมอนิเตอร์ไฟร์สกรีน	73
9.17 โปรแกรมล็อกมอนิเตอร์ไฟร์เบรก	74

# บทที่ 1

## บทนำ

### 1.1 ความสำคัญและที่มาของโครงการ

การใช้งานคอมพิวเตอร์ผ่านทางเครือข่ายนั้น มีความเสี่ยงต่อการถูกบุกรุกทั้งจากภายในและภายนอกองค์กร จึงจำเป็นที่จะต้องมีการป้องกันที่เหมาะสมเพื่อความปลอดภัยในการใช้งาน โดยการใช้ไฟร์วอลล์ซึ่งเป็นเครื่องมือที่ใช้ในการตรวจสอบและป้องกันการใช้งานคอมพิวเตอร์ผ่านทางเครือข่าย และไฟร์วอลล์นั้นก็ยังมีหลายประเภทที่มีความสามารถแตกต่างกันออกไป สำหรับการใช้งานในระดับเครือข่ายซึ่งเป็นระดับที่มีขอบเขตในการใช้งานขนาดใหญ่ ทำให้การใช้ความสามารถของไฟร์วอลล์เพียงชนิดเดียวอาจมีความปลอดภัยไม่เพียงพอ จึงได้มีการนำไฟร์วอลล์ประเภทต่างๆ มาใช้งานร่วมกันเพื่อให้เกิดประสิทธิภาพมากขึ้น

### 1.2 วัตถุประสงค์ของโครงการ

1. เพื่อศึกษาโครงสร้าง และลักษณะการทำงานของไฟร์วอลล์แบบต่างๆ
2. เพื่อปรับปรุงต้นแบบไฟร์วอลล์บนระบบปฏิบัติการยูนิกซ์และไมโครซอฟท์วินโดวส์
3. เพื่อศึกษาหลักการและโครงสร้างของระบบไดเรกทอรีเซอร์วิส
4. เพื่อเขียนโปรแกรมติดต่อระหว่างคอมพิวเตอร์ผ่านเครือข่าย บนระบบปฏิบัติการไมโครซอฟท์วินโดวส์และยูนิกซ์เพื่อใช้งานร่วมกับ แอคทีฟไดเรกทอรี

### 1.3 ขอบเขตของโครงการ

1. เกรดเวย์ไฟร์วอลล์มีความสามารถรองรับการเชื่อมต่อที่จะผ่านเข้ามาภายในเครือข่ายตอบสนองได้เหมาะสม และรายงานเหตุผิดปกติให้กับส่วนกลางได้
2. ไฟร์วอลล์ส่วนบุคคลมีความสามารถควบคุมพฤติกรรมที่น่าสงสัย และรายงานเหตุผิดปกติให้กับส่วนกลางได้
3. ซีเคียวริตี้เว็บพริอ็อกซ์ทำหน้าที่เพิ่มความปลอดภัยให้กับเว็บเซิร์ฟเวอร์ และรายงานเหตุผิดปกติให้กับส่วนกลางได้
4. ส่วนควบคุมไฟร์วอลล์ส่วนกลางสามารถกำหนดกฎให้กับไฟร์วอลล์ส่วนบุคคลและเกรดเวย์ไฟร์วอลล์ได้ สามารถรับรายงานเหตุผิดปกติที่ถูกส่งมาจากไฟร์วอลล์ในส่วนต่างๆ และนำมาแสดงผลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ส่วนควบคุมไฟร์วอลล์ส่วนกลางมีการเก็บรายงานเหตุผิดปกติที่รับมาจากส่วนต่างๆ ไว้ในฐานข้อมูลเพื่อที่จะสามารถนำมาวิเคราะห์สาเหตุของปัญหาในภายหลังได้
6. การติดต่อระหว่างส่วนควบคุมและไฟร์วอลล์จะผ่าน SSL เพื่อรับรองความปลอดภัยและความถูกต้องของข้อมูล

#### 1.4 วิธีการดำเนินงาน

1. ศึกษารายละเอียดของโปรโตคอลที่ซีพี/ไอพี ยูดีพี และโปรโตคอลอื่นๆที่เกี่ยวข้อง
2. ศึกษาโครงสร้างและหลักการทำงานของไฟร์วอลล์แบบต่างๆ
3. ศึกษาหลักการ โครงสร้างการทำงานของแอคทีฟไดเรกทอรี และการเขียนโปรแกรมเพื่อประยุกต์ใช้งานร่วมกับแอคทีฟไดเรกทอรี
4. ศึกษาการทำงานของระบบตรวจจับผู้บุกรุกเพื่อนำมาใช้งานร่วมกับไฟร์วอลล์ส่วนบุคคล และวีเวิร์สเว็บพริ็อกซ์
5. พัฒนาโปรแกรมไฟร์วอลล์ส่วนบุคคล พร้อมทั้งระบบตรวจจับผู้บุกรุกทางเครือข่ายที่มีการทำงานแบบอิงผู้ใช้ (Directory base firewall) มีความสามารถในการรับกฎจากแอคทีฟไดเรกทอรีของส่วนกลาง สามารถจัดการกับแอปพลิเคชันบนเครื่องที่ทำงานอยู่ได้ และสามารถส่งรายงานเหตุผิดปกติกลับไปยังส่วนกลางได้
6. พัฒนาวีเวิร์สเว็บพริ็อกซ์ที่สามารถตรวจสอบและป้องกันให้เว็บเซิร์ฟเวอร์มีความปลอดภัยในการใช้งานได้ และสามารถส่งรายงานเหตุผิดปกติกลับไปยังส่วนกลางได้
7. พัฒนาเกตเวย์ไฟร์วอลล์ที่สามารถกรองการเชื่อมต่อที่ผ่านเข้าออกเครือข่ายได้ โดยรับกฎมาจากแอคทีฟไดเรกทอรีของส่วนกลาง และสามารถส่งรายงานเหตุผิดปกติกลับไปยังส่วนกลางได้

#### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถรู้และเข้าใจถึงหลักการทำงานของไฟร์วอลล์แบบต่างๆ
2. เข้าใจหลักการทำงานของบริการไดเรกทอรีและการติดตั้งปรับตั้งค่าของไมโครซอฟท์แอคทีฟไดเรกทอรี รวมถึงการเขียนโปรแกรมติดต่อกับบริการไดเรกทอรี
3. ชุดโปรแกรมไฟร์วอลล์ที่พัฒนาขึ้นมาสามารถใช้งานป้องกันระบบเครือข่ายได้อย่างมีประสิทธิภาพตามเป้าหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.6 ส่วนประกอบของปฏิญญานิพนธ์

ปฏิญญานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 10 บทด้วยกันคือ

บทที่ 1 กล่าวถึงความสำคัญและที่มาของโครงการ วัตถุประสงค์ของโครงการ ขอบเขตของโครงการ วิธีการดำเนินการ ประโยชน์ที่คาดว่าจะได้รับ และส่วนประกอบของปฏิญญานิพนธ์

บทที่ 2 กล่าวถึงทฤษฎีพื้นฐานที่ใช้ในโครงการ ซึ่งประกอบด้วยไฟร์วอลล์ แอ็คทีฟไดเรกทอรี เน็ตฟิลเตอร์ พร็อกซี ระบบตรวจจับผู้บุกรุกทางเครือข่าย

บทที่ 3 กล่าวถึงการออกแบบการทำงานของระบบที่ใช้กับชุดโปรแกรมเน็ตเวิร์กไฟร์วอลล์ ซึ่งประกอบด้วย รูปโครงสร้างของระบบและแนวคิดในการออกแบบเป็นโปรแกรมในส่วนต่างๆ

บทที่ 4 กล่าวถึงการพัฒนาโปรแกรมไฟร์สเตชัน ซึ่งประกอบด้วยแนวคิด ขอบเขตและความสามารถ การพัฒนาโปรแกรม และการทำงานของโปรแกรม

บทที่ 5 กล่าวถึงการพัฒนาโปรแกรมไฟร์สตาร์ม ซึ่งประกอบด้วยแนวคิด ขอบเขตและความสามารถ การพัฒนาโปรแกรม และการทำงานของโปรแกรม

บทที่ 6 กล่าวถึงการพัฒนาโปรแกรมไฟร์สกรีน ซึ่งประกอบด้วยแนวคิด ขอบเขตและความสามารถ การพัฒนาโปรแกรม และการทำงานของโปรแกรม

บทที่ 7 กล่าวถึงการพัฒนาโปรแกรมไฟร์เบรก ซึ่งประกอบด้วยแนวคิด ขอบเขตและความสามารถ การพัฒนาโปรแกรม และการทำงานของโปรแกรม

บทที่ 8 กล่าวถึงการพัฒนาโปรแกรมล็อกมอนิเตอร์ ซึ่งประกอบด้วยแนวคิด ขอบเขตและความสามารถ การพัฒนาโปรแกรม และการทำงานของโปรแกรม

บทที่ 9 กล่าวถึงการทดลองและผลการทดลองของ โปรแกรมส่วนต่างๆ

บทที่ 10 เป็นบทวิจารณ์และสรุป ซึ่งกล่าวถึงบทสรุปของโครงการ วิจารณ์สิ่งที่ได้รับจากโครงการ และข้อเสนอแนะสำหรับเป็นแนวทางในการพัฒนาต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### ทฤษฎีที่เกี่ยวข้อง

โครงการชุดโปรแกรมเน็ตเวิร์กไฟร์วอลล์มีการพัฒนาโปรแกรมส่วนต่างๆ คือ ไฟร์สเตชัน (FireStation), ไฟร์ออลาร์ม (FireAlarm), ไฟร์สกรีน (FireScreen), ไฟร์เบรก (FireBreak) และ ล็อกมอนิเตอร์ (LogMonitor) ซึ่งในแต่ละส่วนนี้จะมีทฤษฎีที่เกี่ยวข้องอยู่หลายอย่างด้วยกันทั้งทฤษฎีในการพัฒนาโปรแกรมและการนำโปรแกรมมาใช้งานร่วมกันดังที่จะกล่าวถึงต่อไปนี้

#### 2.1 ไฟร์วอลล์

ไฟร์วอลล์เป็นเครื่องมือรักษาความปลอดภัยที่ทำงานในเชิงป้องกันโดยทำหน้าที่ควบคุมการเข้าถึงเน็ตเวิร์ก ซึ่งแพ็คเก็ตที่สามารถผ่านเข้า-ออกเน็ตเวิร์กได้นั้น จะต้องเป็นแพ็คเก็ตที่ไฟร์วอลล์เห็นว่ามีความปลอดภัย โดยอาศัยการเปรียบเทียบคุณสมบัติของแพ็คเก็ตที่จะผ่านไฟร์วอลล์กับกฎที่กำหนดไว้เป็นพื้นฐาน

ไฟร์วอลล์สามารถแบ่งได้หลายชนิดและหลายประเภทแต่ถ้าคำนึงถึงลักษณะการทำงานของไฟร์วอลล์ที่ใช้ในการตรวจสอบ และควบคุมสามารถแบ่งไฟร์วอลล์ได้เป็น 3 ประเภท ดังต่อไปนี้คือ

1. แพ็คเก็ตฟิลเตอร์ (Packet Filtering)
2. แอปพลิเคชันพร็อกซี (Application Proxy)
3. สเตตฟูลอินสเปกชัน (Stateful Inspection)

##### 2.1.1 แพ็คเก็ตฟิลเตอร์ (Packet Filtering)

เป็นลักษณะการทำงานโดยทั่วไปของไฟร์วอลล์ที่ใช้ควบคุมทราฟฟิกโดยอาศัยการตรวจสอบข้อมูลที่ปรากฏอยู่ในแพ็คเก็ตเฮดเดอร์ เช่น แอดเดรสต้นทาง (Source Address) แอดเดรสปลายทาง (Destination Address) พอร์ต (Port) โพรโตคอล (Protocol) เป็นต้น ซึ่งจากข้อมูลเหล่านี้จะสามารถนำมาใช้เป็นเงื่อนไขสำหรับควบคุมการเข้าออกของข้อมูลได้ โดยการพิจารณาข้อมูลทั้งหมดให้เป็นไปตามกฎที่ระบุไว้ ซึ่งเรียกว่า แอคเซสรูล (Access Rules) หรือ กฎของการควบคุมการผ่านการเข้าออกของแพ็คเก็ต โดยทั่วไปรูปแบบของแอคเซสรูลเบื้องต้นจะเป็นดังนี้

ตารางที่ 2.1 แสดงรูปแบบของแอคเซสรูลเบื้องต้น

Source Address	Destination Address	Protocol	Service (Dst. Port)	Action
----------------	---------------------	----------	---------------------	--------

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยข้อมูลทั้งหมดจะเป็นเสมือนตัวแปรที่จะนำมาเปรียบเทียบกับค่าที่ระบุไว้ในแอสเซสรูลทีละค่าเงื่อนไขในการเปรียบเทียบของแต่ละตัวจะเป็นตรรกะ และในส่วนของฟิลด์สุดท้าย "Action" หมายถึงสิ่งที่ไฟร์วอลล์จะกระทำเมื่อค่าในแพ็คเก็ตนั้นตรงกับเงื่อนไข ตัวอย่างเช่น

ตารางที่ 2.2 ตัวอย่างของการตั้งค่าแอสเซสรูลของไฟร์วอลล์

Source Address	Destination Address	Protocol	Service (Dst. Port)	Action
161.246.5.50	ANY	TCP	25	ACCEPT

จากการตั้งแอสเซสรูลเช่นนี้จะหมายความว่าอนุญาตให้แพ็คเก็ตที่มีต้นทาง IP Address 161.246.5.50 และปลายทางใดๆ ที่ใช้โปรโตคอล TCP หมายเลขพอร์ตปลายทาง 25 ผ่านไฟร์วอลล์ได้หากไฟร์วอลล์มีแอสเซสรูลนี้เพียงข้อเดียวก็เท่ากับอนุญาตให้โฮสต์เพียงโฮสต์เดียวคือโฮสต์ที่มี IP Address 161.246.5.50 เท่านั้นที่สามารถใช้บริการ SMTP (TCP พอร์ต 25) ไปยังโฮสต์อื่นที่อยู่อีกฟากหนึ่งของไฟร์วอลล์ได้

### 2.1.2 สเตตฟูลอินสเปกชัน (Stateful Inspection)

สเตตฟูลอินสเปกชันไฟร์วอลล์ เป็นไฟร์วอลล์ที่ทำการควบคุมทราฟฟิกโดยใช้หลักการของแพ็คเก็ตฟิลเตอร์ และการกำหนดแอสเซสรูลเช่นเดียวกัน แต่สเตตฟูลอินสเปกชัน มีความสามารถที่เพิ่มจากแพ็คเก็ตฟิลเตอร์ คือสามารถฟิลเตอร์แบบมองสถานะและภาพรวมโดยในการพิจารณาว่าจะยอมให้แพ็คเก็ตผ่านไปได้หรือไม่ นั้น แทนที่จะดูข้อมูลจากเฮดเดอร์เพียงอย่างเดียว ไฟร์วอลล์แบบสเตตฟูลจะมีการบันทึกข้อมูลเกี่ยวกับการเชื่อมต่อที่เกิดขึ้นลงในตารางเก็บสถานะ (state table) ก่อนที่จะส่งแพ็คเก็ตนั้นไปยัง IP stack ตารางเก็บสถานะนี้จะเป็นส่วนที่ใช้บันทึกข้อมูลสำหรับแต่ละการเชื่อมต่อที่ถูกต้อง โดยปกติจะเก็บข้อมูล ไอพีของต้นทางและปลายทาง, โปรโตคอล, พอร์ต และแฟล็ก แต่มีไฟร์วอลล์บางยี่ห้อที่เก็บข้อมูลลำดับการส่งข้อมูล (sequence number) เพิ่มด้วย เพื่อใช้ในการตรวจสอบแพ็คเก็ตที่กำลังจะเข้ามาและป้องกันการจacking (session hijacking)

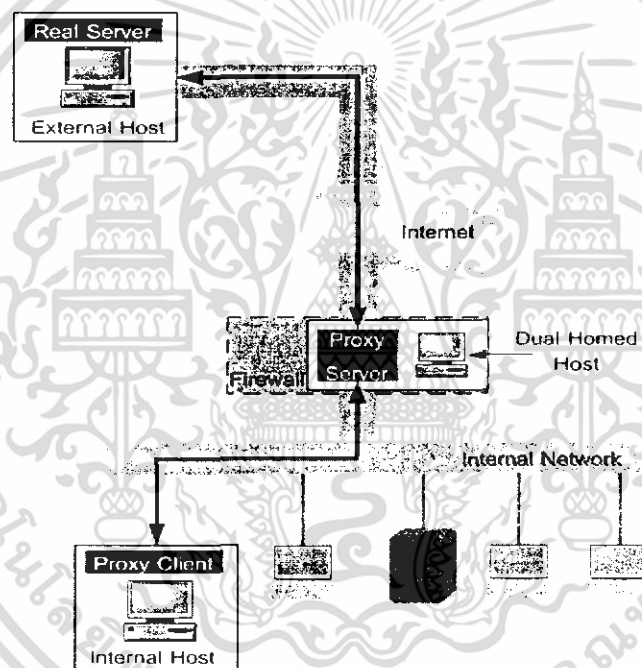
และเมื่อได้รับแพ็คเก็ตใหม่เข้ามา ไฟร์วอลล์แบบสเตตฟูลจะทำการตรวจสอบข้อมูลกับตารางเก็บสถานะว่าเป็นส่วนของการเชื่อมต่อที่สร้างไว้แล้วหรือไม่ โดยจะพิจารณาจากข้อมูล ไอพี และพอร์ตของต้นทางและปลายทาง จะต้องสอดคล้องกับตารางเก็บสถานะ ซึ่งถ้าเป็นส่วนหนึ่งของการเชื่อมต่อจริงก็ไม่มีปัญหาใดๆ ที่ต้องตรวจสอบซ้ำอีก

อย่างไรก็ตาม สเตตฟูลไฟร์วอลล์นี้ยังไม่สามารถป้องกันการโจมตีที่แทรกซึมมาจากการเชื่อมต่อตามปกติได้ เช่น โพรโตคอล FTP นั้น แม้ว่าไฟร์วอลล์แบบสเตตฟูลจะมีการติดตามให้มีการเปิดพอร์ต 20 และ 21 อย่างถูกต้อง แต่ถ้าหากในเนื้อหาที่ส่งมาเป็นสิ่งที่ประหลาดก็ไม่สามารถรับรู้และป้องกันการโจมตีแบบนี้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.3 แอปพลิเคชันพร็อกซี (Application Proxy)

พร็อกซี (Proxy) หรือ แอปพลิเคชันเกตเวย์ (Application Gateway) เป็นแอปพลิเคชันโปรแกรมที่ทำงานอยู่บนไฟร์วอลล์ ที่ตั้งอยู่ระหว่างเน็ตเวิร์ค 2 เน็ตเวิร์ค ทำหน้าที่เพิ่มความปลอดภัยของระบบเครือข่ายโดยการควบคุมการเชื่อมต่อระหว่างเครือข่ายภายในกับภายนอก พร็อกซี จะช่วยเพิ่มความปลอดภัยได้มากเนื่องจากมีการตรวจสอบข้อมูลถึงในระดับของแอปพลิเคชันเลเยอร์ (Application Layer) เมื่อไคลเอนต์ (Client) ต้องการใช้บริการจากภายนอก ไคลเอนต์จะทำการติดต่อไปยังพร็อกซีก่อน แล้วพร็อกซีจึงจะติดต่อไปยังเครื่องปลายทางให้แล้วจะมีการเชื่อมต่อ (connection) 2 การเชื่อมต่อ คือ ไคลเอนต์ กับ พร็อกซี และ พร็อกซี กับเครื่องปลายทาง โดยที่พร็อกซีจะทำหน้าที่รับข้อมูลและส่งข้อมูลให้ใน 2 ทิศทาง ทั้งนี้พร็อกซีจะทำหน้าที่ในการตัดสินใจว่าจะให้มีการเชื่อมต่อกันหรือไม่ หรือจะส่งต่อเพื่อให้เกิดให้หรือไม่



รูปที่ 2.1 แสดงการใช้ Dual-homed Host เป็น พร็อกซีเซิร์ฟเวอร์ (Proxy Server)

## 2.2 เทคโนโลยีแอคทีฟไดเรกทอรี (Active Directory)

แอคทีฟไดเรกทอรีเป็นเทคโนโลยีบริการไดเรกทอรีของไมโครซอฟท์ เป็นบริการที่เกี่ยวข้องกับการจัดเก็บข้อมูลของทรัพยากรที่มีอยู่ในเครือข่ายลงในไดเรกทอรี โดยแอคทีฟไดเรกทอรีจะมีฐานข้อมูลในตัวเองสำหรับจัดเก็บรายละเอียดของทรัพยากรนั้น

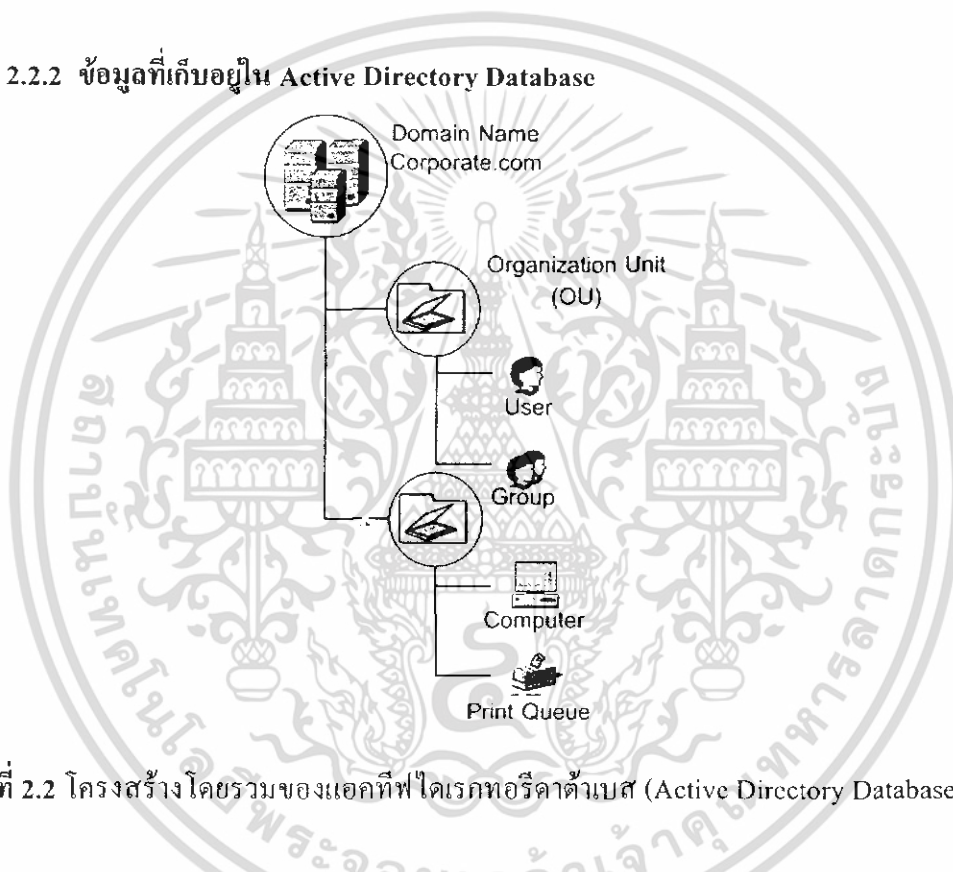
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.2.1 ส่วนประกอบของแอคทีฟไดเรกทอรี

แอคทีฟไดเรกทอรีจะแบ่งการทำงานออกเป็น 2 ส่วน คือ

- Active Directory Service เป็นเครื่องมืออำนวยความสะดวกในการเรียกค้น จัดการกับ บัญชีรายชื่อของผู้ใช้และรายชื่อกลุ่ม และสามารถค้นหารายชื่อทรัพยากรในระบบที่มี คุณสมบัติตามที่ผู้ใช้ต้องการ ซึ่งแอคทีฟไดเรกทอรีเซอร์วิสจะเข้าไปค้นหาข้อมูลจากแอคทีฟไดเรกทอรีดาต้าเบสแล้วแสดงรายละเอียดต่างๆขึ้นมา
- Active Directory Database ทำหน้าที่ในการจัดเก็บข้อมูลในระบบไดเรกทอรี โดยการเข้าถึงดาต้าเบสจะต้องผ่าน การล็อกอินจึงจะมีสิทธิในการทำงาน

### 2.2.2 ข้อมูลที่เก็บอยู่ใน Active Directory Database



รูปที่ 2.2 โครงสร้างโดยรวมของแอคทีฟไดเรกทอรีดาต้าเบส (Active Directory Database)

- ออบเจกต์ต่างๆที่ใช้เป็นตัวแทนของทรัพยากรในระบบเครือข่าย เช่น ออบเจกต์ที่เป็นตัวแทนของแชร์โฟลเดอร์(Shared Folder Object) ออบเจกต์ที่เป็นตัวแทนของเครื่องพิมพ์
- คอนเทนเนอร์ต่างๆที่เป็นของระบบตั้งแต่เริ่มต้นเช่นคอนเทนเนอร์ที่ชื่อUsers, Computers และ Built-in
- คอนเทนเนอร์พิเศษที่เรียกว่า Organizational Unit (OU)
- System Configuration ต่างๆ ของโครงสร้าง Active Directory ทั้งหมด เช่น Schema , Global Catalog

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เป็นที่เก็บ Group Policy Object (GPO) เพื่อใช้ในการควบคุม และจัดการเครื่องคอมพิวเตอร์ของผู้ใช้
- อื่นๆอีกมากมายแล้วแต่จะขยายเพิ่มเติมในอนาคตเช่นออบเจกต์ที่เป็นตัวแทนของ อุปกรณ์สื่อสารในระบบเน็ตเวิร์ก

### 2.2.3 สคีมา (Schema)

เป็นฐานข้อมูลส่วนย่อยที่เก็บอยู่ในแอคทีฟไดเรกทอรีดาต้าเบส ซึ่งทำหน้าที่เก็บรวบรวมข้อกำหนดเกี่ยวกับคลาสของออบเจกต์ทุกคลาส เช่น ออบเจกต์ที่อยู่ในแต่ละคลาสจะต้องมีแอตทริบิวต์อะไร ค่าของแต่ละแอตทริบิวต์มีคุณสมบัติเป็นอะไร Schema สามารถเพิ่มขยายได้โดยผู้บริหารระบบเครือข่าย ได้แก่ การเพิ่มคลาสประเภทใหม่ๆ เข้าไป หรือการเพิ่ม แอตทริบิวต์ใหม่ให้กับคลาสเดิมที่มีอยู่แล้ว

### 2.2.4 การติดต่อกับ Active Directory ( LDAP Protocol)

LDAP (Lightweight Directory Access Protocol) เป็นโพรโตคอลที่พัฒนามาจาก X.500 ซึ่งใช้ในการเข้าถึงและอัปเดตข้อมูลของไดเรกทอรี และเป็นโพรโตคอลมาตรฐานสำหรับการสืบค้นหรือเปลี่ยนแปลงข้อมูลบนบริการไดเรกทอรี โดย LDAP จะทำงานอยู่บนโพรโตคอล TCP/IP อีกที่หนึ่ง ซึ่งไดเรกทอรีจะมีรูปแบบเป็นโครงสร้างต้นไม้ที่แต่ละ entry จะประกอบไปด้วยชุดของ attribute และค่าของ attribute นั้นๆ ซึ่งบริการไดเรกทอรีของไมโครซอฟท์ (Microsoft Active Directory Service) ก็ทำงานอยู่บนโพรโตคอล LDAP เช่นเดียวกัน

ในส่วนการติดต่อระหว่างไคลเอนต์ที่ต้องการใช้บริการของเซิร์ฟเวอร์ LDAP มีหลักการเบื้องต้นดังนี้ LDAP ไคลเอนต์ทำการเริ่มต้น LDAP session โดยเชื่อมต่อเข้ามายัง LDAP เซิร์ฟเวอร์ ซึ่งปกติจะใช้งานพอร์ตหมายเลข 389 (636 สำหรับ SSL/TLS) หลังจากนั้นไคลเอนต์ก็จะส่งคำสั่งต่างๆ ไปยังเซิร์ฟเวอร์ซึ่งจะส่งคืนค่าที่ต้องการให้กับทางไคลเอนต์โดยมีข้อนำส่งเกณฑ์ทางไคลเอนต์ไม่มีความจำเป็นที่จะต้องรอจนกว่าเซิร์ฟเวอร์จะส่งคืนค่ามาแล้วจึงส่งคำสั่งถัดไปสามารถส่งคำสั่งต่อเนื่องได้เรื่อยๆ เช่นเดียวกันทางเซิร์ฟเวอร์อาจส่งค่ากลับมาในลำดับใดๆก็ได้

### Directory structure ของ แอคทีฟไดเรกทอรี

'directory' เป็นโครงสร้างต้นไม้ (tree) ของ entry ในไดเรกทอรี

'entry' ประกอบไปด้วยชุดของ attribute

'attribute' จะต้องมีชื่อและมี value 1 ค่าหรือมากกว่า และถูกกำหนดไว้ใน schema

โดยแต่ละ entry ต้องมีชื่อที่สามารถระบุได้ชัดเจนถึงแต่ละ entry เรียกว่า Distinguished Name (DN)

โดยแต่ละ entry จะมีลักษณะดังนี้เมื่อแสดงในรูปแบบของไฟล์ LDIF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

dn: cn=John Doe,dc=example,dc=com
cn: John Doe
givenName: John
sn: Doe
telephoneNumber: +1 555 6789
telephoneNumber: +1 555 1234
mail: john@example.com
manager: cn=Barbara Doc, dc=example, dc=com
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: top

```

dn เป็นชื่อของ entry เอง ไม่ได้เป็น attribute ใด attribute หนึ่ง ใน entry นั้น จากตัวอย่างข้างต้นจะมี "cn=John Doe" เป็น RDN ของ entry และ "dc=example, dc=com" เป็น dn ของ parent entry ส่วนที่เหลือจะแสดง attribute และ value ที่เก็บไว้ เช่น "cn" สำหรับ common name และ "dc" สำหรับ domain component ซึ่ง value ก็เป็นเพียงชุดของ string ธรรมดาๆ

#### คำสั่งพื้นฐานมีดังนี้

- Bind : พิสูจน์สิทธิ์และระบุเวอร์ชันของ LDAP protocol
- Search : ค้นหาข้อมูลที่ต้องการ ในไดเรกทอรี
- Compare : ทดสอบว่า entry ดังกล่าวมี attribute value ที่ต้องการหาหรือไม่
- Add a new entry : เพิ่มค่า entry ใหม่
- Delete an entry : ลบ entry
- Modify an entry : แก้ไขค่า entry
- Modify DN : ย้ายที่หรือเปลี่ยนชื่อ entry
- Start TLS : ใช้งานการเชื่อมต่อแบบ TLS
- Abandon : ยกเลิกการร้องขอที่ผ่านมา
- Extended Operation : คำสั่งต่างๆ ไปใช้ร่วมกับคำสั่งอื่นๆ
- Unbind : ยกเลิกการเชื่อมต่อ (ไม่ได้เป็นคำสั่งตรงข้ามกับ Bind แต่อย่างใด)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการเข้าถึง organizational unit

```
Set ou = GetObject("LDAP://host1/OU=Sales, DC=ArcadiayBay,DC=COM")
```

```
For each obj in ou
```

```
    Debug.Print obj.Name
```

```
Next
```

ตัวอย่างนี้เป็นการแสดงรายชื่อของ Object ที่อยู่ใน Organizational unit ที่ชื่อว่า Sales ในโดเมน ArcadiayBay.com

## 2.3 netfilter/iptables

netfilter/iptables เป็นโมดูลสำหรับกรองแพ็คเก็ตซึ่งมีอยู่ในระบบปฏิบัติการ GNU/Linux เคอร์เนลตั้งแต่เวอร์ชัน 2.4.x จนถึงปัจจุบันที่เวอร์ชัน 2.6.x โดยจัดเป็นยุคที่ 3 ของไฟร์วอลล์บนลินุกซ์นับตั้งแต่ ipfwadm ซึ่งมีในลินุกซ์เคอร์เนล 2.0.x (ipfwadm พัฒนามาจาก ipfw ของ BSD) จากนั้นจึงกลายมาเป็น ipchains ในลินุกซ์เคอร์เนล 2.2.x โดยมี Rusty Russel เป็นแกนหลักในทีมพัฒนา และได้พัฒนาต่อมาจากกลายเป็น netfilter/iptables ในที่สุด

netfilter นั้นเป็นชื่อของโค้ดในส่วนที่ทำหน้าที่กรองแพ็คเก็ต โดยต่อไปนี้จะเรียกรวมว่าเป็น iptables ซึ่งเป็นคำสั่งในการเรียกใช้งาน netfilter/iptables

iptables เป็นไฟร์วอลล์ที่มีความสามารถในการกรองแพ็คเก็ตแบบสเตทฟูล อินสเปคชันสามารถทำ Network Address Translation (NAT) ได้ และสามารถทำ packet mangling เช่นการแก้ไขค่า ToS หรือ Mark ได้ เป็นต้น

ในโครงการนี้จะให้ความสำคัญไปที่ความสามารถในการกรองแพ็คเก็ตของ iptables เป็นสำคัญซึ่งเป็นความสามารถสำคัญของ iptables ในการใช้งานเป็นไฟร์วอลล์

### 2.3.1 การทำงานเบื้องต้นของ iptables

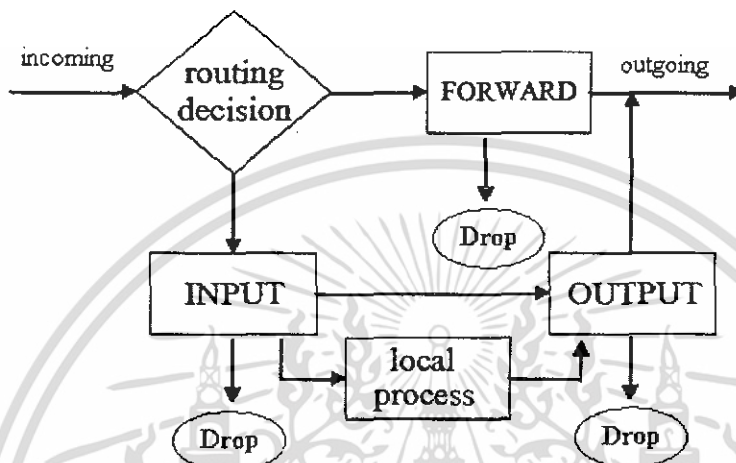
แพ็คเก็ตที่ผ่านเข้ามายังเครื่องไฟร์วอลล์จะผ่านเข้าไปยังเทเบิลต่างๆของ iptables ซึ่งแต่ละเทเบิลก็จะมีหน้าที่ต่างกันไป โดยแต่ละเทเบิลจะมีหลาย chain สำหรับแพ็คเก็ตรูปแบบต่างๆ โดยมีรายละเอียดดังต่อไปนี้

- **Tables**

iptables มีเทเบิลที่สร้างไว้อยู่แล้วทั้งหมด 3 เทเบิล คือ

## 1. Filter Table

ใช้สำหรับกรองแพ็คเก็ตมี chain ที่ถูกสร้างไว้แล้วอยู่ด้วยกัน 3 chain คือ INPUT, OUTPUT, FORWARD เป็นตารางที่มีส่วนร่วมในการใช้งานมากที่สุดของ iptables เป็นจุดที่ใช้ในการตรวจสอบควบคุมการผ่านเข้าออกของแพ็คเก็ต โดยการไหลเวียนของแพ็คเก็ตเฉพาะในส่วน of filter table แสดงให้เห็นได้ดังรูปที่ 2.2 เมื่อแพ็คเก็ตผ่านเข้ามาในระบบจะเข้าไปยัง routing decision เพื่อตัดสินใจว่าแพ็คเก็ตจะถูกส่งไปที่ใด



รูปที่ 2.3 การเดินทางของแพ็คเก็ตใน Filter Table

กรณีที่แพ็คเก็ตถูกส่งผ่านไปยังเครื่องอื่นแพ็คเก็ตนั้นจะผ่านเข้าไปใน FORWARD chain ก่อนที่จะส่งผ่านไปยังปลายทางที่ระบุหรืออาจจะถูก drop ทั้งตามแต่กฎที่ตั้งไว้

ถ้าแพ็คเก็ตนั้นมีเป้าหมายเป็นเครื่องปัจจุบัน แพ็คเก็ตนั้นจะผ่านเข้าไปใน INPUT chain โดยที่จะถูกนำไปประมวลผลใน userspace หรืออาจจะถูก drop ทั้งตามแต่กฎที่ตั้งไว้

กรณีที่แพ็คเก็ตถูกสร้างจากเครื่องปัจจุบัน แพ็คเก็ตจะผ่านเข้าไปใน OUTPUT chain ก่อนที่จะถูกส่งออกไปหรืออาจจะถูก drop ทั้งตามแต่กฎที่ตั้งไว้

## 2. Nat Table

ใช้สำหรับการแปลงแอดเดรส (Network Address Translation), คู่มือในส่วนที่เกี่ยวข้องกับการแปลงแอดเดรสทั้ง SNAT และ DNAT รวมถึงการทำ MASQUERADE และการทำ REDIRECT มี 3 built-in chain

- 1) PREROUTING ทำการแปลงแพ็คเก็ตที่เข้ามาในส่วน of DNAT และดูแลเรื่อง REDIRECT
- 2) POSTROUTING ทำการแปลงแพ็คเก็ตในส่วน of SNAT และดูแลเรื่อง REDIRECT
- 3) OUTPUT ทำการแปลงแพ็คเก็ตที่จะส่งออกไปในส่วน of DNAT และดูแลเรื่อง REDIRECT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3. Mangle Table

เป็นตารางที่ใช้ดูแลการเปลี่ยนแปลงหรือแก้ไขแพ็คเก็ตเช่น เปลี่ยนค่า TTL, ToS, MARK ซึ่งปกติจะใช้ในการทำ routing และ forwarding ที่มีความซับซ้อนสูง มี 2 built-in chain คือ

- 1) PREROUTING chain ทำการเปลี่ยนแปลงแพ็คเก็ตที่เข้ามาก่อนที่จะทำการ routing decision
- 2) OUTPUT chain ทำการเปลี่ยนแปลงแพ็คเก็ตที่จะส่งออกไปจากภายในเครือข่าย

#### 2.3.2 การสร้างกฎของ iptables

รูปแบบของคำสั่งในการใช้งาน iptables

**iptables [table] <command> <match> <target/jump>**

iptables มีรูปแบบการใช้งานดังข้างต้น โดยกฎที่เขียนขึ้นจะเป็นตัวบอกเตอร์เน็ตว่าให้กระทำอย่างไร (action) ในกรณีที่พบแพ็คเก็ตที่ตรงตามกฎที่ระบุไว้

##### ■ Command

- A เพิ่มกฎใหม่ต่อท้าย chain (Append rule)
- D ลบกฎ (Delete rule)
- I เพิ่มกฎใหม่ใน chain (Insert rule)
- R แทนที่กฎเดิมด้วยกฎใหม่ (Replace rule)
- L แสดงกฎใน chain (ถ้าไม่ระบุ chain จะแสดงกฎใน filter table ทั้งสาม built-in chain)
- F ลบกฎทั้งหมดใน chain ทิ้ง
- Z ใช้ reset byte counter สำหรับทุกกฎใน chain ที่กำหนด
- N ใช้สร้าง chain ใหม่
- X ลบ chain ใช้กับ user-defined chain ที่ไม่มีกฎ แต่ไม่สามารถลบ built-in chain ได้
- P เปลี่ยน default policy ของ chain ค่าที่ใช้กับ command นี้คือ ACCEPT, DROP
- E ใช้เปลี่ยนชื่อ chain ใหม่

ซึ่งการใช้ command ข้างต้นนั้นสามารถใช้ร่วมกับอปชันบางอย่างได้ คือ

- v (--verbose) ใช้ร่วมกับ -L, -A, -I, -D, -R เพื่อให้แสดงจำนวน byte ที่ match กับ กฎออกมา (หน่วยที่แสดงออกมานั้นอาจจะเป็น K, M, G)
- x (--exact) ใช้ร่วมกับ -L และ -v เพื่อให้เห็นจำนวนแพ็คเก็ตและจำนวนของ byte ข้อมูลที่ match โดยไม่ให้แสดงผลในหน่วยของ K, M, G
- n (--numeric) ใช้ร่วมกับ -L เพื่อสั่งให้ iptables แสดงข้อมูล ip address และ พอร์ต เป็นตัวเลขเท่านั้น
- line-numbers ใช้ร่วมกับ -L เพื่อแสดงเลขบรรทัดของกฎซึ่งตัวเลขที่แสดงนี้จะสามารถใช้ได้กับคำสั่งแทรกกฎ ที่ระบุเป็นลำดับที่ของกฎ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Match

- การระบุ source, destination IP address

สามารถระบุ source ip address ของแพ็คเก็ตโดยใช้ -s หรือ --source หรือ --src และ destination ip address ใช้ -d หรือ --destination หรือ --dst ในการระบุไอพีแอดเดรสนั้นสามารถทำได้ 3 แบบด้วยกันคือ

1. ใช้ชื่อเต็มแทน เช่น localhost หรือ [www.ce.kmitl.ac.th](http://www.ce.kmitl.ac.th)
2. ระบุไอพีแอดเดรสโดยตรง เช่น 127.0.0.1 หรือ 161.246.4.7
3. ระบุเป็นกลุ่มของไอพีแอดเดรส เช่น 161.246.5.0/24 ซึ่งหมายถึงไอพีแอดเดรสตั้งแต่ 161.246.5.0 – 161.246.5.255 หรือ 161.246.5.0/255.255.255.0 แทน 161.246.5.0/24 ได้

- การทำ Inversion

ทำได้โดยใช้เครื่องหมาย ! นำหน้า argument ที่ต้องการ (! หมายถึง NOT) เช่น -p ! TCP

- การระบุโปรโตคอล

สามารถระบุโปรโตคอลที่ต้องการได้ดังนี้คือ TCP, UDP, ICMP และสามารถใช้ได้ทั้งตัวอักษรเล็กหรือใหญ่ (ใช้ได้ทั้ง tcp และ TCP) เช่น -p TCP หรือ -p ! tcp

- การระบุ interface

-i หรือ --in-interface ตามด้วยชื่อ interface ใช้เพื่อระบุ incoming interface ซึ่งหมายถึงว่าแพ็คเก็ตที่จะ match กับ rule นี้ต้องเข้ามาจาก interface ที่กำหนด เช่น -i eth0 หมายความว่า ทุกแพ็คเก็ตที่เข้ามาทาง eth0 จะ match กับกฎนี้

-o หรือ --out-interface ตามด้วยชื่อของ interface ใช้เพื่อระบุ outgoing interface ซึ่งหมายถึงว่าแพ็คเก็ตที่จะ match กับ rule นี้ กำลังจะเดินทางผ่าน interface ที่ระบุไว้ เช่น -o eth1 หรือ -o ! eth1 จะ match กับกฎนี้

- fragment packet

ในการส่งข้อมูลผ่านเครือข่ายนั้นเป็นเรื่องปกติที่จะเกิดการ fragment ของแพ็คเก็ตเนื่องจากขนาดของแพ็คเก็ตมีขนาดใหญ่เกินไปที่จะส่งไปในครั้งเดียว จำเป็นต้องมีการแบ่งแพ็คเก็ตออกเป็นหลายๆชิ้นทยอยส่งไป ซึ่งเรียกกันว่าการทำ fragment โดยเครื่องปลายทางจะทำหน้าที่ประกอบ fragment แพ็คเก็ตรวมกันเป็นแพ็คเก็ตที่สมบูรณ์ดั้งเดิม ข้อมูลที่เป็น fragment แพ็คเก็ตจะมีเฮดเดอร์ที่สมบูรณ์แค่แพ็คเก็ตแรกเท่านั้นแพ็คเก็ตที่ตามมาจะมีเฮดเดอร์แค่บางส่วนคือ ไอพีแอดเดรส ไม่มีข้อมูลของโปรโตคอลแนบมาด้วย ดังนั้นการตรวจสอบข้อมูลเฮดเดอร์ของ TCP, UDP, ICMP จึงไม่สามารถทำได้ในแพ็คเก็ตที่สองเป็นต้นมา

โดยปกติแล้วมักจะปล่อยให้ fragment แพ็คเก็ตผ่านไป เนื่องจากถ้าสามารถ DROP ตัว fragment แพ็คเก็ต ตัวแรกได้แล้ว มันก็ไม่สามารถถูกประกอบที่เครื่องปลายทางได้ แต่ทั้งนี้ fragment แพ็คเก็ต ที่ถูกปล่อยให้ดังกล่าวอาจจะทำให้เครื่องที่ได้รับไม่สามารถทำงานต่อได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- TCP extension

ถ้ามีการเรียกใช้ `-p tcp` ตัว TCP extension ก็จะถูกโหลดมาใช้งานโดยอัตโนมัติ โดยมี  
 ออปชั่นให้เลือกใช้งานดังนี้

`--tcp-flags mask flags : mask` นั้นหมายถึง flag ที่ต้องการตรวจสอบ และ flag เป็นตัวที่  
 บ่งชี้ว่า flag ใดต้องถูก set บ้าง

`--source-port` หรือ `--sport` สามารถใช้ได้ทั้งตัวเลขและตัวอักษร ระบุเป็น port เดียว หรือ  
 ช่วงของ port ได้

`--destination-port` หรือ `--dport` มีรูปแบบการใช้งานเช่นเดียวกันกับ `--sport`

- UDP extension

เช่นเดียวกับ TCP ตัว UDP extension มีออปชั่นให้เลือกใช้เพียงแค่ 2 อย่างเท่านั้นคือ `--  
 source-port (--sport)` และ `--destination-port (--dport)` โดยต้องระบุ `-p udp` ด้วย

- ICMP extension

โดยการระบุ `-p icmp` ก็สามารถใช้งาน ICMP extension ได้ โดยมีออปชั่นให้เลือกคือ `--  
 icmp-type` เช่น `--icmp-type host-unreachable` (หรือใช้เลข 3 แทนได้) นอกจากนี้ยังสามารถระบุ  
 type/code ได้ เช่น 3/3 ซึ่งหมายถึง port unreachable

- Match Extension

เป็น netfilter package อื่นๆรูปแบบการใช้งานให้ใช้ `-m` แล้วตามด้วย match ที่ต้องการมี  
 ออปชั่นให้เลือกใช้งานดังต่อไปนี้

mac รูปแบบการใช้งาน: `-m mac` ใช้ตรวจสอบ source MAC address ว่าตรงกับค่าที่ระบุไว้  
 หรือไม่ มีประโยชน์สำหรับ *PREROUTING* และ *INPUT chain*

limit รูปแบบการใช้งาน: `-m limit` ใช้เพื่อจำกัดจำนวนของการ match ที่อาจจะมากเกินไป  
 เป็นประโยชน์สำหรับ rule ที่วางไว้ตอนท้ายสุดของ chain (ใช้รวมกันกับ DROP policy) ซึ่งส่วน  
 ใหญ่เป็น rule ที่ใช้เก็บข้อมูลลงล็อกไฟล์ ซึ่งถ้าผู้บุกรุกส่งแพ็คเก็ตที่ไม่เข้าข่าย rule ใดๆ ใน chain  
 จนกระทั่งมาถึง rule ที่ทำหน้าที่เก็บล็อกนี้ถ้าแพ็คเก็ต ที่เข้ามาจำนวนมากก็อาจจะทำให้ฮาร์ดดิสก์  
 เต็มได้ ดังนั้นจึงต้องใช้จำกัดจำนวนในการเก็บข้อมูลลงล็อก ซึ่งมีออปชั่นให้ใช้งานดังนี้คือ

State Match รูปแบบการใช้งานคือ `-m state` หรือ `--match state` เป็น โมดูลที่ใช้ประโยชน์  
 ได้เป็นอย่างดี มีออปชั่นให้ใช้งานดังนี้

NEW – แพ็คเก็ตที่เป็นตัวสร้างการเชื่อมต่อใหม่

ESTABLISHED – แพ็คเก็ตที่เกี่ยวข้องกับการเชื่อมต่อที่สร้างไว้แล้ว

RELATED - แพ็คเก็ตที่เกี่ยวข้องกับการเชื่อมต่อที่สร้างไว้แล้ว แต่ไม่ใช่ส่วนหนึ่งส่วนใด  
 ของการเชื่อมต่อ

INVALID – เป็นแพ็คเก็ตที่ไม่เกี่ยวข้องกับส่วนอื่นเลย

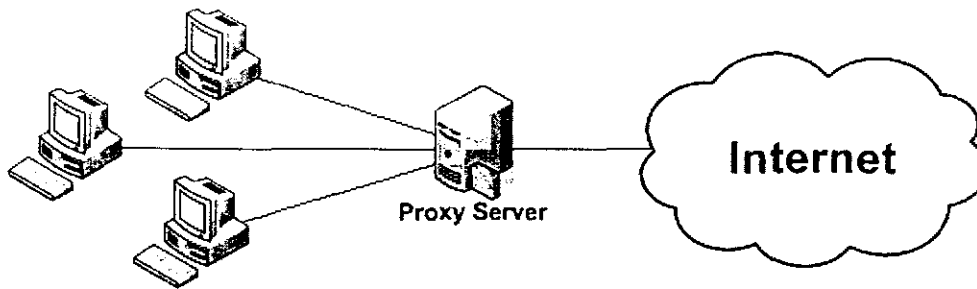
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Target
- เป็นส่วนที่ระบุ action กับแพ็คเก็ตต่างๆที่ match กับกฎที่ได้ตั้งขึ้นซึ่งมี target ในแบบต่างๆ ดังนี้
- ACCEPT อนุญาตให้แพ็คเก็ตที่ match กับกฎนี้สามารถผ่านไป
  - DROP แพ็คเก็ตที่ match กับกฎนี้ให้ทำการ Drop ทิ้งหรือไม่ให้ผ่าน
  - LOG เป็นโมดูลที่มีความสามารถในการเก็บข้อมูลลงล็อก
  - REJECT คล้ายกับ DROP แต่จะมีการส่ง ICMP port unreachable กลับไปยังผู้ส่ง
  - RETURN ออกจาก chain ปัจจุบันกลับไปยัง chain ที่ทำการเรียกมา
  - QUEUE เป็น chain พิเศษใช้สำหรับส่งต่อแพ็คเก็ตไปยังแอปพลิเคชันที่เขียนขึ้นมารองรับ โดยเฉพาะ โดยจะต้องมี queue handler และแอปพลิเคชันเป็นส่วนประกอบที่จะทำงานร่วมกัน
  - User-defined chain ผู้ใช้สามารถสร้าง chain ใหม่ได้โดยต้องใช้ตัวอักษรตัวเล็กทั้งหมดสำหรับ chain ที่สร้างขึ้นใหม่ เมื่อแพ็คเก็ต match กับกฎที่เป็น user-defined chain แพ็คเก็ตจะถูกนำไปตรวจสอบใหม่โดย user-defined chain นั้นๆและถ้าใน chain ไม่มีการตัดสินใจใดๆตัวแพ็คเก็ตก็สามารถย้อนกลับมายังกฎถัดไปใน chain เริ่มต้นได้

## 2.4 พร็อกซี

### 2.4.1 เว็บพร็อกซี (Regular Webproxy)

เว็บพร็อกซีมีลักษณะการใช้งานคือ การอนุญาตให้เครื่องไคลเอ็นต์ภายในเครือข่ายสามารถติดต่อกับอินเทอร์เน็ตได้โดยผ่านเว็บพร็อกซี โดยเว็บพร็อกซีจะรอรับการร้องขอจากเครื่องไคลเอ็นต์ภายในเครือข่าย แล้วทำการส่งต่อการร้องขอนี้ไปยังเครื่องเซิร์ฟเวอร์ จากนั้นก็รอรับการตอบกลับของเซิร์ฟเวอร์แล้วทำการส่งต่อคำตอบรับนั้นให้กับไคลเอ็นต์ ซึ่งเว็บพร็อกซีนี้สามารถทำการแคชข้อมูล คือการเก็บสำเนาข้อมูลจากอินเทอร์เน็ตมาไว้ที่เว็บพร็อกซี ทำให้เว็บพร็อกซีนั้น ไม่ต้องทำการร้องขอข้อมูลที่เคยทำการการร้องขอไปแล้วซ้ำ สามารถที่จะส่งข้อมูลเหล่านี้ให้กับเครื่องไคลเอ็นต์ที่ทำการร้องขอข้อมูลได้ทันที ทำให้การใช้งานอินเทอร์เน็ตนั้นทำได้รวดเร็วมากขึ้น



รูปที่ 2.4 แสดงตำแหน่งของเว็บพ็อกซี่ในเครือข่าย

#### ความสามารถและการใช้งานเว็บพ็อกซี่

- อนุญาตและจำกัดการใช้งานอินเทอร์เน็ตของเครื่องไคลเอนต์โดยวิธีการตรวจสอบไอพีแอดเดรส
- แคมข้อมูลจากภายนอกให้เครื่องภายในเครือข่ายที่ต้องการข้อมูลที่เหมือนกัน เรียกใช้งานได้เร็วขึ้น
- สามารถควบคุมให้การเข้าใช้งานอินเทอร์เน็ตและซบเน็ตที่ต้องการได้โดยกำหนด URL เป็นหลัก
- อนุญาตหรือปฏิเสธการร้องขอจากไคลเอนต์โดยมีรูปแบบเงื่อนไขในการตัดสินใจ
- กำหนด URL ที่จะใช้ในการกรองหรืออนุญาตให้เรียกใช้ผ่านเครื่องเซิร์ฟเวอร์ได้

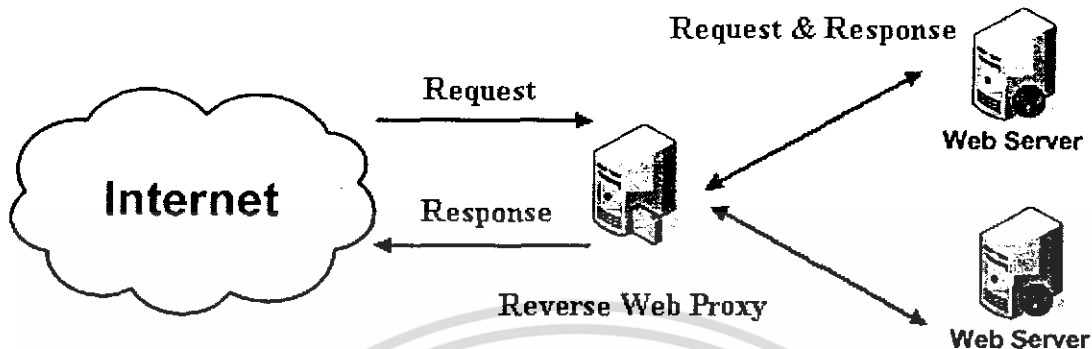
#### 2.4.2 รีเวิร์สเว็บพ็อกซี่ (Reverse Webproxy)

รีเวิร์สเว็บพ็อกซี่ เป็นพ็อกซี่ที่ทำงานบนฝั่งเซิร์ฟเวอร์แทนที่จะทำงานบนฝั่งไคลเอนต์จึงเรียกว่าเป็นการทำงานแบบรีเวิร์ส รีเวิร์สเว็บพ็อกซี่เป็นแอปพลิเคชันพ็อกซี่สำหรับเซิร์ฟเวอร์ที่ใช้โปรโตคอล HTTP รีเวิร์สเว็บพ็อกซี่จะทำงานอยู่ระหว่างไคลเอนต์กับเว็บเซิร์ฟเวอร์ทำหน้าที่เป็นเกตเวย์ของเว็บเซิร์ฟเวอร์ หรือ ไอพีแอดเดรสที่ใช้สำหรับรับการร้องขอจากภายนอก ทำให้ไคลเอนต์เห็นรีเวิร์สเว็บพ็อกซี่เป็นเครื่องเว็บเซิร์ฟเวอร์ และดูแลทราฟฟิกก่อนที่จะเข้าถึงเว็บเซิร์ฟเวอร์ โดยตรวจสอบดูการร้องขอที่ส่งเข้ามานั้นเคยมีการแคชไว้หรือไม่ถ้ามีอยู่ในแคชก็จะทำการตอบกลับข้อมูลในแคชให้กับไคลเอนต์ ทำให้เว็บเซิร์ฟเวอร์ไม่ต้องสร้างข้อมูลตอบกลับหรือผลลัพธ์ทุกครั้งที่ได้รับการร้องขอ เว้นแต่ข้อมูลประเภท Dynamic ที่จะต้องมีการส่งการร้องขอไปยังเว็บเซิร์ฟเวอร์เพื่อประมวลผลทุกครั้ง โดยก่อนใช้งานรีเวิร์สเว็บพ็อกซี่ต้องมีการกำหนด prefix mapping ให้แกรีเวิร์สเว็บพ็อกซี่ก่อนซึ่ง prefix mapping มี 2 ชนิด คือ

- 1) regular mapping ใช้สำหรับบอกรีเวิร์สเว็บพ็อกซี่ว่า URL prefix ไหนถูกแทนที่และบอก URL ปลายทางจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2) reverse mapping เป็นส่วนแปลง URL prefix ไปเป็น URL ของรีเวิร์สเว็บพร็อกซี่ซึ่งเป็นชื่อเว็บเซิร์ฟเวอร์จริงที่ใช้ในการติดต่อสื่อสารกับไคลเอนต์



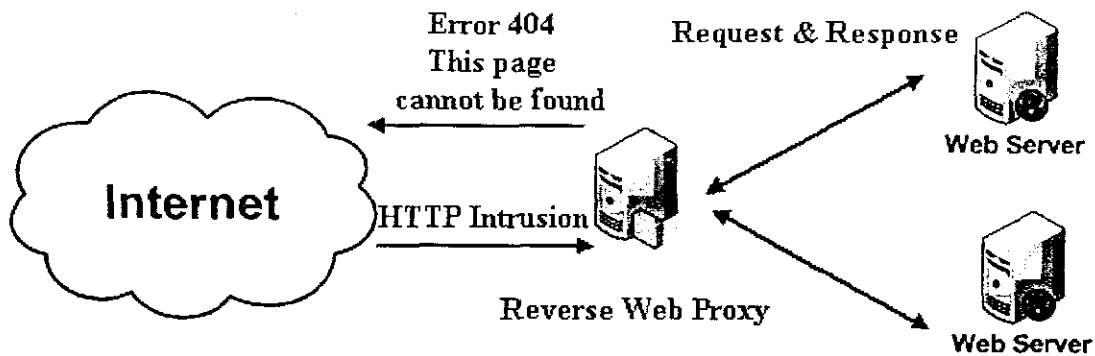
รูปที่ 2.5 การทำงานของรีเวิร์สเว็บพร็อกซี่

ในการใช้งานรีเวิร์สเว็บพร็อกซี่นั้นจะมีการใช้งานร่วมกับไฟร์วอลล์โดยมีการกำหนดกฎให้มีเพียงแต่ไอพีแอดเดรสของรีเวิร์สเว็บพร็อกซี่เท่านั้นที่สามารถติดต่อกับเว็บเซิร์ฟเวอร์ได้ ซึ่งหากมีการติดต่อสื่อสารเป็นการร้องขอจากไคลเอนต์มายัง URL ของรีเวิร์สเว็บพร็อกซี่ซึ่งเป็นทางที่ใช้ในการติดต่อกับเว็บเซิร์ฟเวอร์ รีเวิร์สเว็บพร็อกซี่ก็จะทำการตรวจสอบการร้องขอนั้นกับ prefix mapping ว่าจะต้องส่งการร้องขอนั้นให้กับเว็บเซิร์ฟเวอร์ใด จากนั้นจึงจะส่งการร้องขอนั้นไปยังไฟร์วอลล์เพื่อทำการส่งการร้องขอนั้นให้กับเว็บเซิร์ฟเวอร์

ทางฝั่งของเว็บเซิร์ฟเวอร์เมื่อประมวลผลของการร้องขอที่ได้รับจากไคลเอนต์เสร็จแล้วจะทำการส่งผลลัพธ์กลับไปยังไคลเอนต์ ก็จะต้องทำการส่งผ่านไปยังรีเวิร์สเว็บพร็อกซี่ก่อนเช่นกัน ซึ่งเมื่อรีเวิร์สเว็บพร็อกซี่ได้รับผลลัพธ์ที่จะส่งกลับไปยังไคลเอนต์ ก็จะนำไปตรวจสอบกับ reverse mapping และทำการเปลี่ยน URL และ HTTP header กลับเป็น URL ที่ใช้ในการติดต่อกับไคลเอนต์ ก่อนที่จะส่งผลลัพธ์นั้นกลับไปยังไคลเอนต์ ซึ่งไคลเอนต์ก็จะมองเห็นว่าการร้องขอนั้นถูกตอบรับโดยรีเวิร์สเว็บพร็อกซี่ แต่แท้จริงแล้วถูกตอบโดยเว็บเซิร์ฟเวอร์

#### 2.4.3 รีเวิร์สเว็บพร็อกซี่แบบปลอดภัย (Secure Reverse Web Proxy)

เป็นการใช้งานรีเวิร์สเว็บพร็อกซี่ให้มีความปลอดภัยในการใช้งาน โดยทำการตรวจสอบการร้องขอที่เข้ามาก่อนที่จะส่งต่อให้กับเว็บเซิร์ฟเวอร์ ว่าการร้องขอนั้นตรงกับรูปแบบการโจมตีหรือไม่ หากพบว่าการร้องขอที่ตรงกับรูปแบบการโจมตีก็จะทำการสกัดการร้องขอนั้นไว้ โดยอาจจะทำการเก็บบันทึกลงล็อกไฟล์และทำการส่งค่า page ที่เหมือน page ปกติเช่น *page Error 404 This Page can not be found* เป็นการตอบสนองที่เรียกว่า Silent Killer Operation



รูปที่ 2.6 การทำงานของรีเวิร์สเว็บพร็อกซีแบบปลอดภัย

## 2.5 ระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ (Intrusion Detection System)

ระบบตรวจจับผู้บุกรุกทางเครือข่ายทางคอมพิวเตอร์นี้จะมีการเก็บข้อมูลของแพ็คเก็ตต่างๆ ที่เข้ามาสู่ระบบเครือข่ายแล้วนำมาวิเคราะห์เปรียบเทียบกับกฎต่างๆที่กำหนดไว้ รวมถึงนโยบายขององค์กร เพื่อตรวจสอบว่ามีสิ่งผิดปกติเกิดขึ้นกับระบบหรือไม่ หากเกิดสิ่งผิดปกติก็จะแจ้งเตือนไปยังผู้ดูแลระบบหรือเก็บไว้ในล็อกไฟล์ต่อไป

การตรวจจับผู้บุกรุกทางคอมพิวเตอร์สามารถแบ่งตามลักษณะของการโจมตีได้ 5 ประเภท

1. การพยายามเจาะเข้าไปทำการถ่ายเครือข่าย (Attempted break-ins)
2. การปลอมแปลงเพื่อเข้ามาโจมตีเครือข่าย (Masquerade attacks)
3. การอาศัยจุดบกพร่องของระบบรักษาความปลอดภัยเพื่อเจาะเข้าสู่เครือข่าย (Penetration of the security control system)
4. การโจมตีเพื่อปิดบริการ (Denial of service)
5. การสำรวจระบบ (System survey)

### 2.5.1 ระบบตรวจจับผู้บุกรุกที่ใช้งานในไฟร์วอลล์

สำหรับระบบตรวจจับผู้บุกรุกทางเครือข่ายที่สร้างขึ้นเพื่อใช้งานร่วมกับเพอร์ซันนอลไฟร์วอลล์นั้นมุ่งเน้นศึกษาในเรื่องของการออกแบบพัฒนาระบบตรวจจับการสำรวจระบบ และการโจมตีเพื่อให้ปิดบริการ ซึ่งเป็นลักษณะของการโจมตีระบบที่สำคัญและมีแนวโน้มเพิ่มขึ้นมากในปัจจุบัน มีรายละเอียดดังต่อไปนี้

#### 1. การสำรวจระบบ

การบุกรุกเพื่อทำการสำรวจระบบเป็นเก็บข้อมูลของระบบ เพื่อใช้ในการโจมตีโดยข้อมูลที่ผู้โจมตีมักต้องการได้แก่ หมายเลขไอพีแอดเดรส ชื่อเครื่อง โครงสร้างทางเครือข่ายของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป้าหมาย พอร์ตหรือบริการที่เครื่องเป้าหมายเปิด และระบบปฏิบัติการรวมทั้งเวอร์ชันที่ติดตั้งบนเครื่องเป้าหมาย สำหรับการสำรวจระบบที่สามารถทำการตรวจจับได้มีอยู่ 3 วิธีการสำรวจดังนี้

### 1.1 ปิงสวீป (Ping sweep)

การตรวจจับการปิงสวี่ปทำได้โดยการตรวจสอบดูแพ็คเก็ต ICMP Request (Type8) ที่เข้ามายังระบบ หากมีแพ็คเก็ตลักษณะนี้จำนวนมากเกินกว่าที่กำหนดและมีปลายทางแตกต่างกันจะสามารถสรุปได้ว่าในเครือข่ายกำลังถูกสำรวจโดยการปิงสวี่ป

### 1.2 การสแกนพอร์ต

การตรวจสอบการสแกนพอร์ตทำได้โดยการตรวจดูแพ็คเก็ตที่มีหมายเลขพอร์ตปลายทางในลักษณะกระจาย คือแพ็คเก็ตมีการส่งไปยังเรื่อยๆ เดียวแต่มีการส่งไปยังพอร์ตต่างๆ กันเป็นจำนวนมากเกินกว่าที่กำหนดจะสามารถสรุปได้ว่ากำลังถูกสำรวจโดยการสแกนพอร์ต

### 1.3 การตรวจสอบระบบปฏิบัติการ

การตรวจจับการตรวจสอบระบบปฏิบัติการสามารถตรวจสอบได้จากการพิจารณาแพ็กเก็ตที่ถูกส่งไปในชั้นที่ซีพีของทุกๆพอร์ตว่าเป็นแพ็คเก็ตแบบผัดปกติหรือไม่ กล่าวคือในแต่ละสเตทของทีซีพีโปรโตคอลนั้นจะมีรูปแบบแพ็กเก็ตตายตัวอยู่ ตามสถานะปัจจุบันของสเตทของทีซีพี เช่น หากต้องการเริ่มต้นการเชื่อมต่อโปรโตคอลทีซีพี จะต้องกำหนดให้แพ็กเก็ต ACK เป็น 1 ส่วนแพ็กเก็ตอื่นต้องเป็น 0 หรือหากต้องการยกเลิกการเชื่อมต่อโปรโตคอลทีซีพี จะต้องกำหนดให้แพ็กเก็ต FIN เป็น 1 ส่วนแพ็กเก็ตอื่นเป็น 0 เป็นต้น

วิธีการตรวจจับที่ใช้ในโปรแกรมนี้ ได้ทำการตรวจจับโดยหากแพ็คเก็ตมีแพ็กเก็ต ACK และ FIN เป็น 1 พร้อมกันในแพ็คเก็ตเดียวกัน จะระบุว่าเป็นการบุกรุกโดยการสำรวจเพื่อระบุระบบปฏิบัติการเนื่องจากการตรวจจับวิธีนี้เป็นกรณีมาตรฐานที่โปรแกรมที่ทำการระบุระบบปฏิบัติการทุกโปรแกรมจะนำมาตรวจสอบ และเป็นแพ็กเก็ตที่ไม่สามารถเกิดได้จริงเมื่อมีการใช้งานโปรโตคอลทีซีพี

## 2. การโจมตีเพื่อปิดบริการ

การโจมตีเพื่อปิดบริการคือการกระทำใดๆ ที่ทำให้ระบบเป้าหมายไม่สามารถให้บริการบางอย่างได้ หรือไม่สามารให้บริการต่อไปได้ การโจมตีระบบเครือข่ายคือการสร้างภาระให้กับเครือข่าย แบ่งได้ 3 ประเภทดังนี้

### 2.1 การส่งแพ็คเก็ตปริมาณมาก

การตรวจจับแพ็คเก็ตที่เข้ามาในลักษณะนี้ทำได้โดยใช้การนับจำนวนแพ็คเก็ตที่เข้ามาสู่ระบบโดยพิจารณาจากแอดเดรสปลายทาง (Destination Address) ในแพ็คเก็ตเฮดเดอร์ของไอพี หากเป็นค่าเดียวกันให้นับจำนวนแพ็คเก็ตที่เข้ามาในช่วงเวลาหนึ่ง แล้วนำค่าที่ได้มาเปรียบเทียบกับค่าที่ยอมรับได้ หากค่าที่นับได้มากกว่าค่าที่ยอมรับได้ ก็ให้แจ้งเตือนแก่ผู้ดูแลระบบ หรือเก็บไว้ในล็อกไฟล์ ความยากของการวิเคราะห์แบบนี้อยู่ที่การหาค่าที่ระบบยอมรับได้ เพราะขึ้นอยู่กับปัจจัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลายประการ เช่น ความเร็วของเครือข่าย ความเร็วของหน่วยประมวลผลเครื่อง ปริมาณหน่วยความจำในเครื่อง เป็นต้น

การหาค่าที่ระบบยอมรับได้นี้ สามารถทำได้โดยการเชื่อมต่อกับระบบที่วิเคราะห์ จากนั้นหาจำนวนแพ็คเกจที่เข้ามาในระบบในลักษณะการใช้งานปกติของแต่ละช่วงเวลา จากนั้นนำค่าสูงสุดที่ได้มาเป็นค่าที่ระบบยอมรับได้ ค่าที่ผ่านการวิเคราะห์และยอมรับได้โดยปกติมีค่าประมาณประมาณ 20,000 - 30,000 แพ็คเกจต่อวินาที

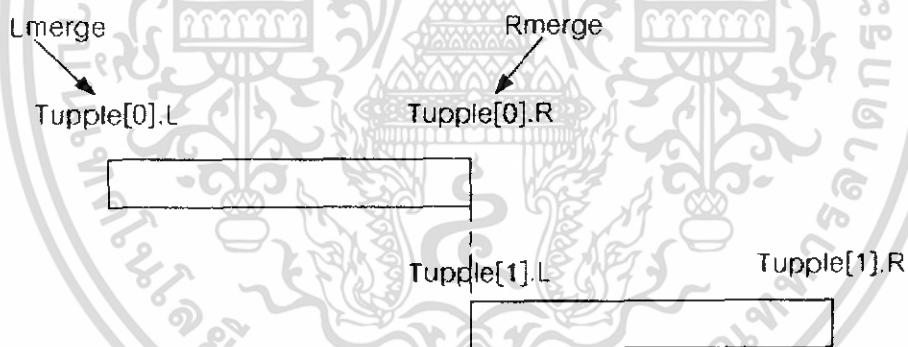
## 2.2 ความผิดปกติของแฟล็กเมนต์

การตรวจสอบความผิดปกติ ของแฟล็กเมนต์มีขั้นตอนก่อนข้างซับซ้อน ซึ่งแยกอธิบายตามประเภทของความผิดปกติได้ ดังต่อไปนี้

### 2.2.1 การส่งแพ็คเกจที่มีลำดับผิดปกติ และ แพ็คเกจที่มีขนาดหลั่มล้ากัน

การวิเคราะห์ความผิดปกติของแพ็คเกจในลักษณะนี้ต้องวิเคราะห์หลังจากกระบวนการรีแฮสเซนเบิ้ลไปแล้ว ดังนั้นจึงนำบัพเฟอร์เข้ามาช่วยในการเก็บข้อมูล เพื่อนำมาวิเคราะห์ ดังนี้

- Fragment Buffer คือ บัพเฟอร์ที่เก็บข้อมูลในการวิเคราะห์ของแพ็คเกจ ไอพีและข้อมูลที่จำเป็นอื่นๆไว้ ได้แก่ Source\_IP, Destination\_IP, Identification, Protocol, Sec, PointArray และ Array\_Fragment การเก็บข้อมูลจะเก็บในลักษณะของโครงสร้างข้อมูลแบบลิงคิสต์



รูปที่ 2.7 แสดงการเก็บข้อมูลของตัวแปร Tuple

การเก็บข้อมูลใน Fragment Buffer มีตัวแปรต่างๆ ที่จัดเก็บดังตารางที่ 2.3 และการเก็บข้อมูลส่วน Fragment โดยจะเก็บเป็น Array มีข้อมูลตามตารางที่ 2.4

### ตารางที่ 2.3 แสดง โครงสร้างการเก็บข้อมูลของ Fragment Buffer

IP_Src	IP_Dst	Identification	Protocol	Sec	Pointarray	Arrau_Fragment
--------	--------	----------------	----------	-----	------------	----------------

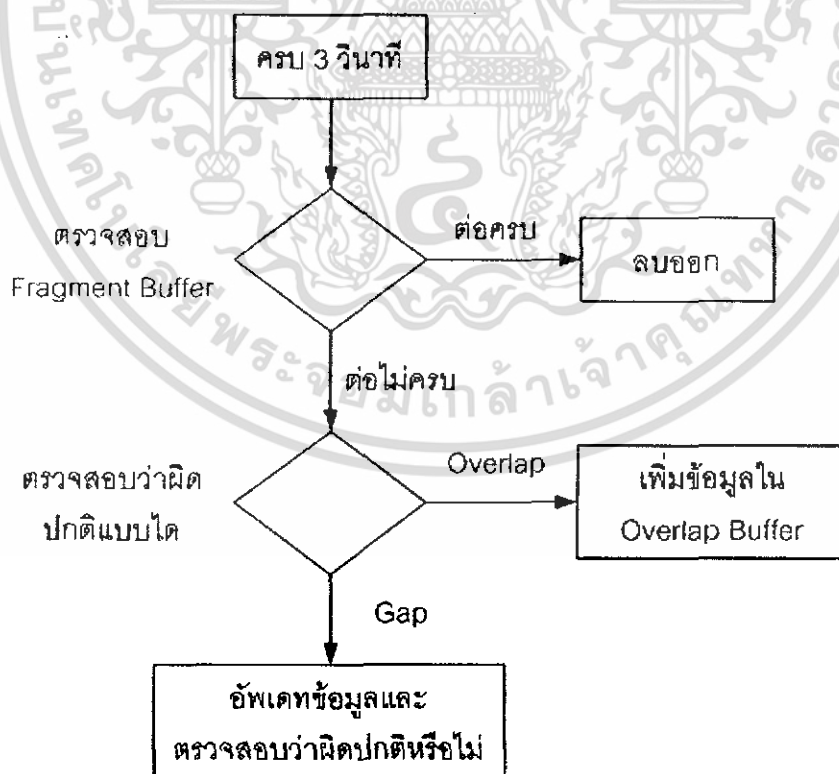
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4 แสดงโครงสร้างการเก็บข้อมูลของ Fragment

Flag_U	Flag_D	Flag_M	Offset	Size_Data
--------	--------	--------	--------	-----------

- Overlap Buffer เป็นบัฟเฟอร์ที่เก็บข้อมูลเมื่อตรวจพบว่าการเชื่อมต่อของแพ็คเก็ต มีการเก็บในลักษณะของลิงค์ลิสต์
- Gap Frame Buffer คือ บัฟเฟอร์ที่เก็บข้อมูล เมื่อตรวจพบว่าการประกอบเฟรมไม่ได้ในลักษณะมีช่องว่างระหว่างแพ็คเก็ต มีการเก็บในลักษณะของลิงค์ลิสต์

ในการวิเคราะห์จะใช้บัฟเฟอร์เหล่านี้ร่วมกัน โดยเก็บข้อมูลแพ็คเก็ตที่เข้ามาทั้งหมดลงใน Fragment Buffer หากแพ็คเก็ตที่ส่งมาสามารถรวมกันได้ก็รวมกันเป็นแพ็คเก็ตเดี่ยวที่ต่อเนื่องกัน แต่หากรวมกันแล้วเกิดความผิดปกติ ให้แจ้งมายัง Overlap Buffer หรือ Gap Frame Buffer แล้วแต่ความผิดปกติที่เกิดขึ้น หากไม่มีความผิดปกติเกิดขึ้นเมื่อครบ 3 วินาที โปรแกรมจะตรวจสอบ Fragment Buffer ว่าหากมีแพ็คเก็ตใดยังไม่ได้ประกอบหรือประกอบไม่ครบให้เก็บไว้ใน Overlap Buffer หรือ Gap Frame Buffer และเมื่อครบ 3 วินาที ข้อมูลใน Overlap Buffer หรือ Gap Frame Buffer นี้จะออกมาที่หน้าจอเพื่อแจ้งให้ผู้ดูแลระบบทราบหรือเก็บไว้ในล็อกไฟล์เพื่อบันทึกความผิดปกติที่เกิดขึ้น หากไม่มีความผิดปกติใดๆเกิดขึ้นและแพ็คเก็ตเหล่านั้นสามารถประกอบเป็นเฟรมได้อย่างถูกต้อง ให้ลบเฟรมเหล่านั้นออกจากบัฟเฟอร์ทันที เพื่อไม่ให้สิ้นเปลืองเนื้อที่ในการจัดเก็บ

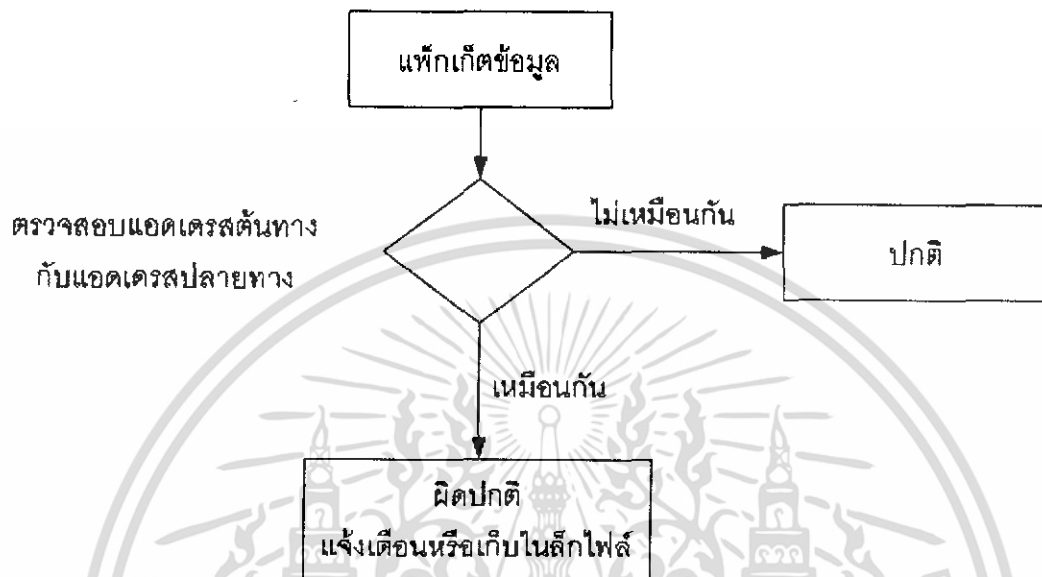


รูปที่ 2.8 แสดงการตรวจสอบความผิดปกติในการทำแฟร็กเมนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.2.2 การส่งแพ็คเกจแบบวนลูป

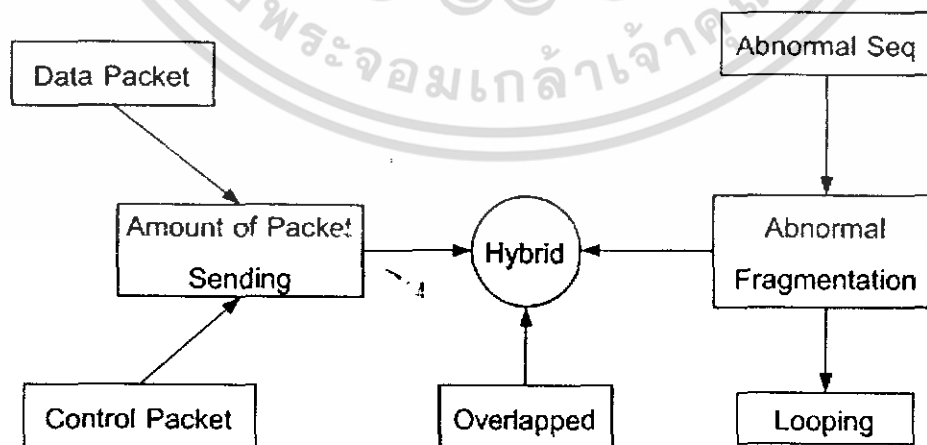
การส่งแพ็คเกจแบบนี้สามารถทำได้โดยการเปรียบเทียบค่าไอพีแอดเดรสต้นทาง และ ไอพีแอดเดรสปลายทางของเฮดเดอร์แพ็คเกจ หากค่าไอพีต้นทางและไอพีปลายทางเป็นค่าเดียวกัน แสดงว่ามีความผิดปกติเกิดขึ้นเพราะทำให้เกิดการส่งในลักษณะวนลูป ซึ่งมีขั้นตอนดังรูปที่ 2.9



รูปที่ 2.9 แสดงการตรวจสอบแพ็คเกจที่ส่งแบบวนลูป

### 2.3 การโจมตีแบบผสม (Hybrid)

การวิเคราะห์แพ็คเกจประเภทนี้ให้นำวิธีการวิเคราะห์ที่กล่าวข้างต้นมาใช้ร่วมกัน เนื่องจากเกิดวิธีการที่ผสมผสานกันระหว่างวิธีต่าง ๆ ที่ได้กล่าวมาแล้ว ซึ่งสามารถแยกวิเคราะห์ห่อออกเป็นแต่ละแบบหรือ วิเคราะห์รวมกันก็ได้ดังขั้นตอนในรูปที่ 2.10



รูปที่ 2.10 แสดงแผนภูมิแสดงประเภทของการโจมตีเพื่อให้ปิดบริการแบบผสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.5.2 ระบบตรวจจับผู้บุกรุกที่ใช้งานในไฟร์เบรก

สำหรับไฟร์เบรกซึ่งเป็นรีเวิร์สเว็บพรีอ็อกชั่นนั้นจะมีการใช้งานของระบบตรวจจับผู้บุกรุกเพื่อตรวจสอบในระดับของแอปพลิเคชันเลเยอร์ที่เป็นการติดต่อกับเว็บเซิร์ฟเวอร์ที่อยู่ข้างหลัง รีเวิร์สเว็บพรีอ็อกชันสำหรับในส่วนนี้ได้มีการใช้โปรแกรม Snort\_inline ซึ่งเป็นโปรแกรมระบบตรวจจับและป้องกันผู้บุกรุก (Intrusion Prevention System IPS) คือโปรแกรมระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ที่สามารถป้องกันการโจมตีได้ทันทีที่ตรวจพบ เป็นการเพิ่มความสามารถในการรับมือเพื่อให้เกิดให้กับ โปรแกรมระบบตรวจจับผู้บุกรุกเดิมนั่นเอง

การติดตั้ง การตั้งค่าการใช้งาน การทดสอบการใช้งานจะกล่าวถึงในบทของการออกแบบ การพัฒนา และการทดสอบการใช้งานต่อไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 3

### การออกแบบโครงสร้างและการพัฒนาระบบ

#### 3.1 แนวคิด

ในระบบเครือข่ายทั่วไปนั้นจะมีการแบ่งโซนในการใช้งานออกเป็น 3 โซนคือ เครือข่ายภายใน เครือข่ายภายนอก และ เครือข่าย DMZ เพื่อใช้ในการแบ่งแยกลักษณะของโฮสต์และความน่าเชื่อถือของโฮสต์นั้นๆ ซึ่งในแต่ละโซนก็จะมีการใช้งานไฟร์วอลล์ในแต่ละประเภทที่แตกต่างกันเพื่อความเหมาะสมในการใช้งานสำหรับโฮสต์แต่ละประเภท เช่น เครือข่ายภายในโฮสต์ที่ทำงานอยู่ในส่วนนี้จะมีมากมายเพราะอาจเป็นเครื่องของพนักงานแต่ละคนทำให้ความต้องการในการใช้งานไฟร์วอลล์นั้นต่างกันไฟร์วอลล์ที่ใช้งานจึงควรที่จะเป็นเพอร์ซันนอลไฟร์วอลล์ ในส่วนของเกตเวย์หรือทางเชื่อมต่อระหว่างโซนนั้นก็ควรที่จะมีเกตเวย์ไฟร์วอลล์ติดตั้งอยู่เพื่อทำการกรองแพ็กเก็ตที่ผ่านเข้าออกในแต่ละโซน และในส่วนของเครือข่าย DMZ ซึ่งเป็นที่อยู่ของเครื่องเว็บเซิร์ฟเวอร์ก็อาจจะมีการติดตั้งซีเคียวริตีวีรส์เว็บพริคซ์เพื่อใช้ในการป้องกันการเข้าถึงหรือการโจมตีที่ไม่หวังดีต่อเครื่องเว็บเซิร์ฟเวอร์

จะเห็นได้ว่าในระบบเครือข่ายหนึ่งๆมีการใช้งานไฟร์วอลล์หลายประเภทที่มีหน้าที่แตกต่างกันทำให้การจัดการกับกฎที่จะใช้กับไฟร์วอลล์ในส่วนต่างๆนั้นทำได้ยาก เพราะต้องมีการคำนึงถึงกฎของไฟร์วอลล์ในส่วนอื่นๆที่มีความเกี่ยวข้องกัน ซึ่งถ้ามีการติดตั้งไฟร์วอลล์ในหลายตำแหน่งเช่นพวกเพอร์ซันไฟร์วอลล์ก็จะต้องทำการกำหนดกฎใหม่ทุกครั้งที่ทำารติดตั้งใหม่ จึงได้มีความคิดที่จะเพิ่มความสามารถในการกำหนดกฎให้กับเพอร์ซันนอลไฟร์วอลล์ และเกตเวย์ไฟร์วอลล์ จากส่วนกลางเพียงจุดเดียว เพื่อความสะดวกในการควบคุมและจัดการกับระบบเครือข่าย

ในปัจจุบันนี้มีการนำระบบไดเรกทอรีเซอร์วิสมาใช้ในองค์กรมากยิ่งขึ้นเพราะมีความง่ายในการควบคุม จัดการ และเก็บข้อมูลของผู้ใช้ อีกทั้งยังมีการพิสูจน์ตน (Authentication) ก่อนที่จะเข้าใช้งานทรัพยากรของระบบ และมีมาตรการ การรักษาความปลอดภัยข้อมูลของระบบที่มีประสิทธิภาพ จึงได้นำคุณสมบัติของ แอ็คทีฟไดเรกทอรีในวินโดวส์เซิร์ฟเวอร์ 2003 มาเพิ่มความสามารถเพื่อใช้ในการควบคุมกฎของไฟร์วอลล์ และ ใช้ในการเก็บล็อกลงในฐานข้อมูล จากเครื่องลูกข่ายที่อยู่ภายในโดเมนคอนโทรลเลอร์เดียวกัน ทำให้สามารถควบคุมการทำงานของไฟร์วอลล์ได้ที่จุดศูนย์กลางเพียงจุดเดียว ซึ่งหากมีการโจมตีเกิดขึ้นนั้นสามารถเก็บผลจากการโจมตีได้เพื่อรวบรวมเข้าด้วยกันให้สามารถนำมาวิเคราะห์กฎที่ จะเพิ่มเข้าไปได้อย่างเหมาะสมทำให้ระบบมีความปลอดภัยยิ่งขึ้น

จากที่ได้กล่าวมาข้างต้นทำให้เกิดแนวความคิดสร้างชุดโปรแกรมเน็ตเวิร์กไฟร์วอลล์ที่มีความสามารถดังนี้

- เพอร์ซันนอลไฟร์วอลล์ที่สามารถป้องกันการโจมตีได้ตามกฎที่ได้รับจากส่วนกลาง พร้อมทั้งมีระบบตรวจจับผู้บุกรุกทำงาน และสามารถจัดเก็บล็อกไฟล์ที่ผิดปกติไว้ที่ฐานข้อมูลของเครื่องส่วนกลางได้
- เกตเวย์ไฟร์วอลล์ที่สามารถรองแพ็กเก็ตที่ผ่านเข้าออกเครือข่ายได้ตามกฎที่ได้รับจากส่วนกลาง และสามารถจัดเก็บล็อกไฟล์ที่ผิดปกติไว้ที่ฐานข้อมูลของเครื่องส่วนกลางได้
- รีเวิร์สเว็บพร็อกซี่ที่สามารถป้องกันการโจมตีและการเข้าถึงเว็บเซิร์ฟเวอร์ได้โดยตรง และสามารถจัดเก็บล็อกไฟล์ที่ผิดปกติไว้ที่ฐานข้อมูลส่วนกลางได้เช่นเดียวกัน

จากแนวคิดนี้ทำให้สามารถแบ่งโปรแกรมการทำงานออกเป็น 5 ส่วนดังนี้คือ

#### 1. ไฟร์สเตชัน (FireStation)

ออกแบบให้ทำหน้าที่ในการกำหนดกฎของเกตเวย์ไฟร์วอลล์และเพอร์ซันนอลไฟร์วอลล์ ซึ่งกฎของเพอร์ซันนอลไฟร์วอลล์สามารถที่จะกำหนดกฎให้กับกลุ่มผู้ใช้และผู้ใช้แต่ละคนได้

#### 2. ไฟร์ออลาร์ม (FireAlarm)

ออกแบบให้เป็นเพอร์ซันนอลไฟร์วอลล์ทำหน้าที่ป้องกันการโจมตีได้ตามกฎที่รับมาจากไฟร์สเตชันได้อย่างถูกต้องและเมื่อมีการบุกรุกจะส่งข้อมูลการบุกรุกกลับไปเครื่องแม่ข่าย

#### 3. ไฟร์สกรีน (FireScreen)

ออกแบบให้เป็นเกตเวย์ไฟร์วอลล์ทำหน้าที่ป้องกันการโจมตีได้ตามกฎที่รับมาจากเครื่องไฟร์สเตชันได้อย่างถูกต้องและเมื่อมีการบุกรุกจะส่งข้อมูลการบุกรุกกลับไปเครื่องแม่ข่าย

#### 4. ไฟร์เบรก (FireBreak)

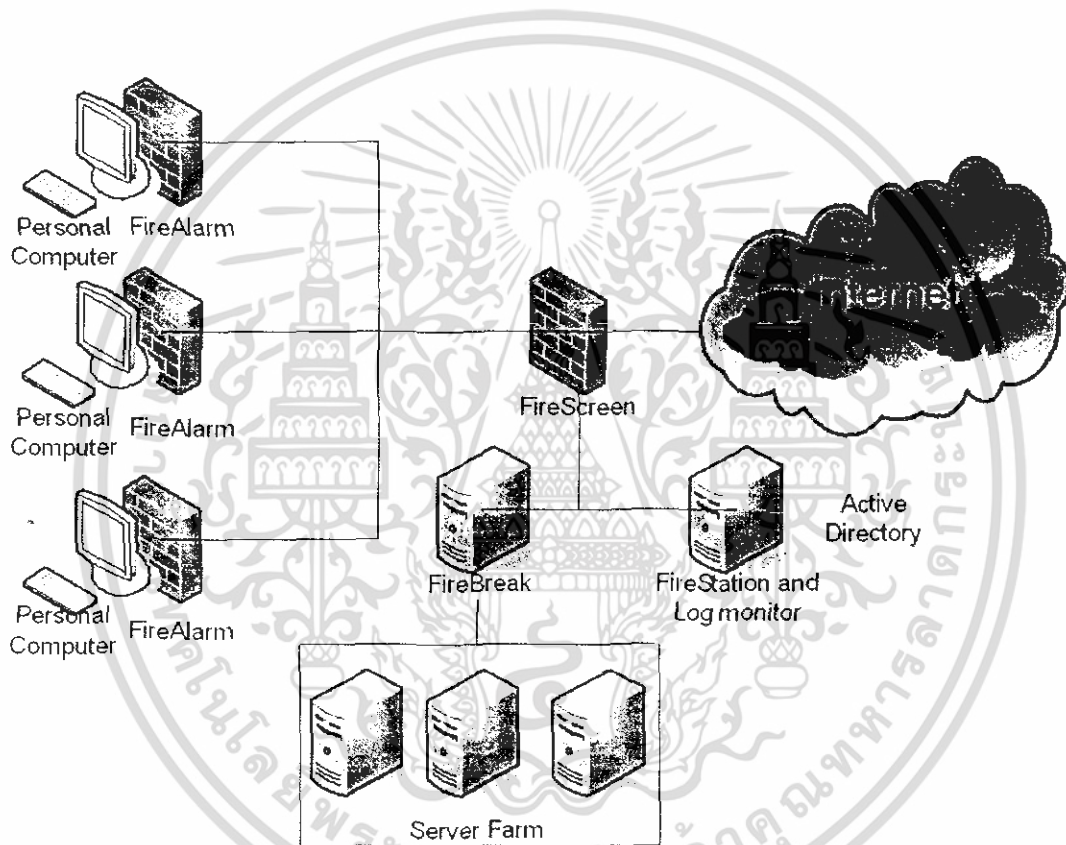
ออกแบบให้เป็นโปรแกรมซีเคียวริตี้เว็บพร็อกซี่สำหรับเพิ่มความปลอดภัยให้กับเว็บเซิร์ฟเวอร์ซึ่งอยู่ภายในโซน DMZ มีความสามารถตรวจจับผู้บุกรุกในระดับแอปพลิเคชัน และเมื่อมีการบุกรุกจะส่งข้อมูลการบุกรุกกลับไปเครื่องแม่ข่าย

#### 5. ล็อกมอนิเตอร์ (LogMonitor)

ออกแบบให้รับรายละเอียดการโจมตีจากเครื่องลูกข่ายที่อยู่ในระบบ และจัดการเกี่ยวกับการนำล็อกไฟล์จากฐานข้อมูลขึ้นมาแสดงผล

### 3.2 โครงสร้างระบบ

จากแนวคิดข้างต้นสามารถนำชุดโปรแกรมเน็ตเวิร์กไฟร์วอลล์มาใช้กับระบบเครือข่ายได้ ดังรูปที่ 3.1 คือโปรแกรมไฟร์สเคชชั่น, ไฟร์เบรก และสื่อกมอนิเตอร์จะถูกวางอยู่ในโซน DMZ โปรแกรมไฟร์อลาร์มจะถูกติดตั้งอยู่ในโซนของเครือข่ายภายในที่เครื่องลูกข่ายต่างๆ ที่อยู่ในโดเมนคอนโทรลเลอร์ของเครื่องไฟร์สเคชชั่น และ โปรแกรมไฟร์สกรีนจะถูกติดตั้งอยู่ที่เครื่องที่ทำหน้าที่เป็นเกตเวย์ของระบบเครือข่าย



รูปที่ 3.1 ระบบเครือข่ายที่มีการติดตั้งชุดโปรแกรมเน็ตเวิร์กไฟร์วอลล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

# การพัฒนาโปรแกรมไฟร์สเทชั่น

### 4.1 แนวคิด

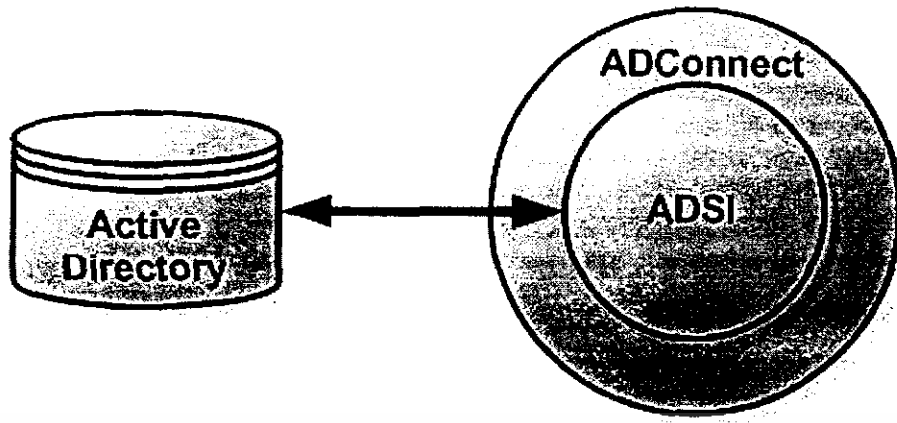
เป็นโปรแกรมที่ใช้ในการกำหนดคกฏของเพอร์ซันนอลไฟร์วอลล์ และเกตเวย์ไฟร์วอลล์ โดยใช้เทคโนโลยีของแอ็คทีฟไดเรกทอรีเซอร์วิส (Active Directory Service) ในการจัดเก็บคกฏของเพอร์ซันนอลไฟร์วอลล์แต่ละเครื่องและคกฏของเกตเวย์ไฟร์วอลล์ เพื่อให้เครื่องลูกข่ายและเครื่องที่ทำหน้าที่เป็นเกตเวย์ไฟร์วอลล์ที่มีสิทธิในการใช้งานสามารถเข้าถึงแอ็คทีฟไดเรกทอรีเพื่อนำคกฏไปใช้ในการป้องกันการบุกรุกได้ โดยข้อมูลคกฏจะเก็บอยู่ในฐานข้อมูลของแอ็คทีฟไดเรกทอรี ทำให้การวางตำแหน่งของเครื่องที่ควรจะต้องติดตั้งโปรแกรมนี้คือ เครื่องที่ทำงานเป็นเครื่องแม่ข่ายหรือเครื่องเซิร์ฟเวอร์ที่เปิดใช้งานโดเมนคอนโทรลเลอร์ (Domain Controller)

### 4.2 ขอบเขตและความสามารถ

- สามารถพิสูจน์ผู้ใช้ที่มาขอใช้ทรัพยากรได้อย่างถูกต้องโดยใช้ความสามารถของแอ็คทีฟไดเรกทอรี
- สามารถกระจายคกฏและจัดเก็บได้อย่างมีระเบียบ

### 4.3 การพัฒนาโปรแกรม

การติดต่อระหว่างไฟร์สเทชั่น ไฟร์วอลล์และสื่อคอมอนิเตอร์ของไฟร์วอลล์จะทำงานอยู่บนความสามารถของแอ็คทีฟไดเรกทอรี (Active Directory) ของวินโดวส์ 2003 เซิร์ฟเวอร์โดยข้อมูลจะเก็บอยู่ในฐานข้อมูลของแอ็คทีฟไดเรกทอรี (Active Directory Database) และเข้าถึงข้อมูลผ่าน ADSI (Active Directory Service Interfaces) ไปยังแอ็คทีฟไดเรกทอรีเซอร์วิส (Active Directory Service) เพื่อเข้าถึงที่จัดเก็บ แสดงดังรูปที่ 4.1 ดังนั้นจึงต้องพัฒนาโปรแกรมไฟร์สเทชั่น ไฟร์วอลล์ และสื่อคอมอนิเตอร์ของไฟร์วอลล์ไปพร้อมๆกันเพราะต้องมีการติดต่อกับแอ็คทีฟไดเรกทอรีในกรณีเดียวกัน



รูปที่ 4.1 แสดงรูปแบบการเข้าถึง Active Directory Service

การพัฒนาโปรแกรมไฟร์วอลล์ ฟร็อลาร์ม และล็อกมอนิเตอร์ของไฟร์วอลล์จะมีการสร้างสกีมา(Schema) ซึ่งเป็นโครงสร้างฐานข้อมูลของแอ็กทีฟไดเร็กทอรีประกอบไปด้วย คอนเทนเนอร์ คลาส และ แอตทริบิวต์ โดยการเข้าถึงข้อมูลเหล่านั้นจะมีการกำหนดหน้าที่การทำงานมาแล้ว ดังนั้นถ้าเราต้องการนำข้อมูลที่เราต้องการจัดเก็บหรือเข้าถึงรูปแบบใหม่เข้าไปจึงต้องมีการสร้างขึ้นมาใหม่เพื่อให้เหมาะสมกับการทำงานของระบบ ไคเรกทอรีเบสไฟร์วอลล์

ข้อมูลที่จัดเก็บประกอบด้วย ข้อมูลของกฎ ข้อมูลรายละเอียดการโจมตีและข้อมูลที่บอกการเปลี่ยนแปลงกฎโดยเพิ่มสกีมาที่เหมาะสมลงไป ในฐานข้อมูลของแอ็กทีฟไดเร็กทอรีดังนี้

1. แอตทริบิวต์ รายละเอียดการเพิ่มจะเป็นดังตาราง ที่ 4.1

ตารางที่ 4.1 รายละเอียดของแอตทริบิวต์ที่เพิ่มในฐานข้อมูลของแอ็กทีฟไดเร็กทอรี

Common Name	OID	Syntax
firewallRule	1.2.840.113556.1.4.7000.142	Case Insensitive String
firewallLog	1.2.840.113556.1.4.7000.144	Case Insensitive String
update	1.2.840.113556.1.4.7000.146	Case Insensitive String

2. คลาส รายละเอียดการเพิ่มจะเป็นดังตารางที่ 4.2

ตารางที่ 4.2 รายละเอียดคลาสที่เพิ่มในฐานข้อมูลของแอ็กทีฟไดเร็กทอรี

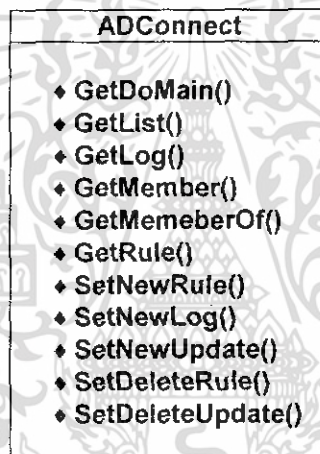
Common Name	OID	Syntax
ISAGFW	1.2.840.113556.1.4.7000.143	Auxiliary

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยกำหนดให้แอตทริบิวต์ firewallRule เป็น Mandatory และ FirewallLog จะเป็น Auxiliary ใน คลาส ISAGFW

### 3. ผนวกสกีมาใหม่ เข้ากับ คลาส groups คลาส users และคลาส computers

ในการเข้าถึงข้อมูลที่อยู่ในยังแอ็คทีฟไดเรกทอรีเซอรัวิส นั้นต้องใช้ ADSI ดังนั้นจึงมีการพัฒนา function ที่ใช้เข้าถึงและกำหนดการทำงานกับแอ็คทีฟไดเรกทอรีได้อย่างง่ายขึ้น โดยการสร้างคลาส ADConnect ที่มีรายละเอียด Function ดังรูปที่ 4.2 โดยคลาส ADConnect จะใช้ในโปรแกรมไฟร์สแตชัน ไฟร์วอลล์ และสื่อกอนโทรลเลอร์ของไฟร์วอลล์ที่พัฒนาขึ้นเพื่อติดต่อยังแอ็คทีฟไดเรกทอรีเซอรัวิส



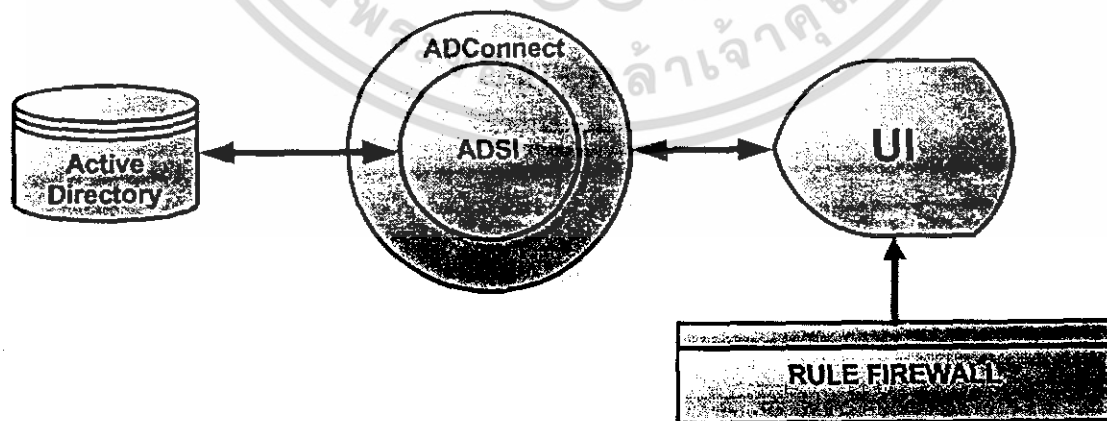
รูปที่ 4.2 รายละเอียดของคลาส ADConnect

ตารางที่ 4.3 รายละเอียดการทำงานของฟังก์ชันในคลาส ADConnect

Function	การทำงาน
GetDomain	หาชื่อเครื่อง โดเมนที่ผู้ใช้สังกัดอยู่
GetList	อ่านรายละเอียดที่เป็นจุดอ้างอิง
GetLog	อ่านรายละเอียด Log
GetMember	อ่านรายชื่อ User จาก Group
GetMemberOf	อ่านรายชื่อ Group จาก User
GetRule	อ่าน กฎ ที่มีอยู่จาก
SetNewRule	สร้างกฎ
SetNewLog	สร้าง Log
SetNewUpdate	บอกให้ Gateway ทราบว่ากฎมีการเปลี่ยนแปลงโดย set flag เป็น YES
SetDeleteRule	ลบ กฎ ที่ต้องการลบ
SetDeleteUpdate	ลบ flag ที่ใช้ในการบอกการเปลี่ยนแปลงของกฎ

#### 4.4 การทำงานของโปรแกรม

เมื่อเปิดโปรแกรมไฟร์สเตชัน(FireStation) จะทำการอ่านข้อมูลผ่านคลาส ADConnect เพื่อเข้าถึงข้อมูลที่อยู่ในกับแอ็คทีฟไดเรกทอรีเซอร์วิส ( Active Directory Service) โดยจะเลือกแสดงตามกลุ่มผู้ใช้ที่ถูกเลือก และเมื่อมีการ เพิ่ม ลบ และแก้ไข ข้อมูลกฎที่อยู่ในฐานข้อมูล ดังรูปที่ 3.4 จะมีส่วนแสดงการทำงานของโปรแกรมไฟร์สเตชัน(FireStation)



รูปที่ 4.3 แสดงโครงสร้างการทำงานของไฟร์สเตชัน(FireStation)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

# การพัฒนาโปรแกรมไฟร์วอลล์

### 5.1 แนวคิด

โปรแกรมไฟร์วอลล์เป็นโปรแกรมเพอร์ซันนอลไฟร์วอลล์ ที่มีระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ที่พัฒนาขึ้นเองโดยภาษา C++ เพื่อใช้งานร่วมกับไฟร์สแตชันในการรับกฎของเพอร์ซันนอลไฟร์วอลล์สำหรับแต่ละเครื่องลูกข่าย โดยเป็นการใช้งานไฟร์วอลล์แบบอิงผู้ใช้คือเมื่อผู้ใช้งานมีการล็อกอินใช้งานเครื่องที่อยู่ในระบบเครือข่ายภายในเครื่องใดก็ตาม ก็จะมีกฎสำหรับผู้ใช้คนนั้นในการใช้งานเพอร์ซันนอลไฟร์วอลล์เสมอ

### 5.2 ขอบเขตและความสามารถ

ในระบบจะมีเครื่องลูกข่ายหลายเครื่อง โดยทุกเครื่องจะมีการติดตั้ง เพอร์ซันนอลไฟร์วอลล์ ซึ่งก็คือเอเจนต์ที่ฝังตัวทำงานอยู่แบบอัตโนมัติ โดยจะมีการควบคุมดูแลจากส่วนกลางโดยผู้ใช้งาน ไม่รับรู้ถึงการทำงานของเอเจนต์ รวมไปถึงผู้ดูแลระบบสามารถกำหนดกฎการป้องกันการบุกรุกให้ผู้ใช้ในแต่ละกลุ่มตามความเหมาะสม

อีกทั้งยังมีส่วนของการทำงานที่เป็นระบบตรวจจับผู้บุกรุกที่คอยทำหน้าที่ตรวจสอบว่ามี การบุกรุกเข้ามาที่เครื่องลูกข่ายนั้นๆหรือไม่ ถ้ามีการแจ้งเตือนไปยังเครื่องเซิร์ฟเวอร์กลางเพื่อนำผลที่ได้ไปวิเคราะห์หาแนวทางป้องกันต่อไป และส่วนตรวจการเชื่อมต่อเครือข่ายของโปรเซสต่างๆเพื่อให้ผู้ใช้งานได้รับรู้ว่าโปรเซสอะไรที่เชื่อมต่อกับเครือข่ายอยู่บ้าง และสามารถกำจัดโปรเซสที่น่าสงสัยได้ด้วยตนเอง ทำให้ผู้ใช้สามารถป้องกันตนเองได้ในระดับหนึ่ง

### 5.3 การพัฒนาโปรแกรม

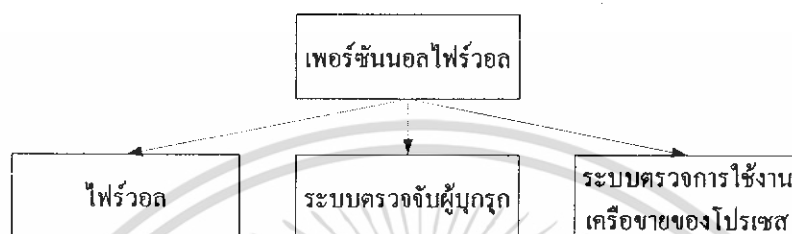
การพัฒนาโปรแกรมไฟร์วอลล์จะใช้ Firewall-Hook Driver ซึ่งเป็น DDK (Driver Development Kit) ของทางบริษัทไมโครซอฟต์ ซึ่งไดร์เวอร์ตัวนี้ใช้ในการฟิลเตอร์แพ็คเก็ต การเขียนโปรแกรมนี้จะใช้เฮดเดอร์(Header) ที่ชื่อว่า IpDrvFw.h โดยภายในเฮดเดอร์(Header) นี้จะมีโครงสร้างต่างๆที่ใช้ในการฟิลเตอร์ และใช้ WinPCap ช่วยในการดักจับแพ็คเก็ตเพื่อมาวิเคราะห์หาว่ามีกรบุกรุกระบบเกิดขึ้นหรือไม่ โดยกฎที่ใช้ในการฟิลเตอร์นั้นจะรับมาจากฐานข้อมูลของแอ็คทีฟไดเรกทอรี (Active Directory Database)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5.4 การทำงานของโปรแกรม

ไฟร์วอลล์เป็นเพอร์ซันนอลไฟร์วอลล์ซึ่งจะแบ่งการทำงานออกเป็น 3 ส่วน คือ

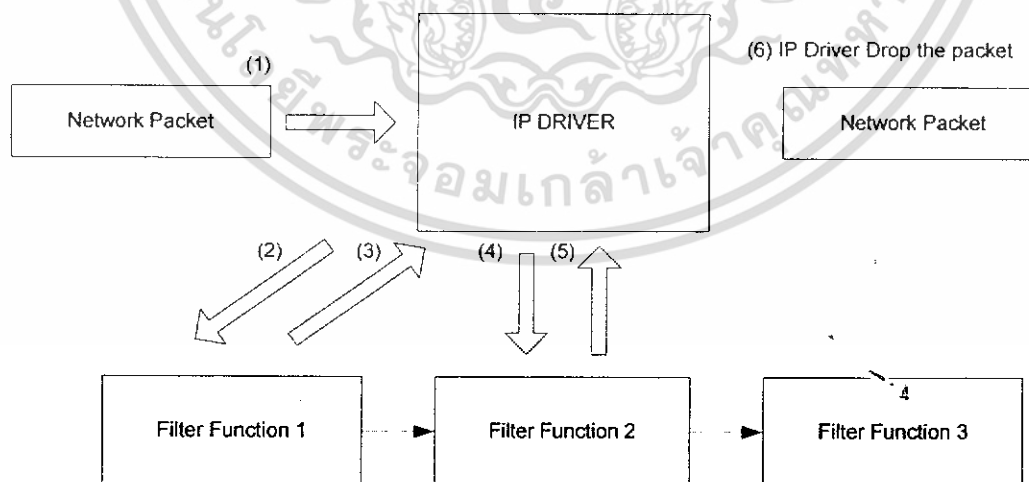
1. ไฟร์วอลล์
2. ระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์
3. ระบบตรวจการใช้งานเครือข่ายของโปรเซส



รูปที่ 5.1 แสดงส่วนประกอบเพอร์ซันนอลไฟร์วอลล์

### 5.3.1 แพ็คเก็ตฟิลเตอร์ริงไฟร์วอลล์บนวินโดวส์ XP

เพอร์ซันนอลไฟร์วอลล์ที่พัฒนามีลักษณะการทำงานแบบแพ็คเก็ตฟิลเตอร์ริง(Packet Filtering) โดยจะมีการตรวจสอบแพ็คเก็ตด้วยฟังก์ชันฟิลเตอร์ว่าจะอนุญาตหรือไม่อนุญาตให้แพ็คเก็ตนั้นผ่านไปได้ โดยที่ฟังก์ชันฟิลเตอร์จะมีการกำหนดลำดับไว้แล้วให้ระบบเรียกฟังก์ชันฟิลเตอร์เข้าไปทำงานทีละฟังก์ชันตามลำดับ จนกว่าจะมีฟังก์ชันใดฟังก์ชันหนึ่งส่งค่ากลับเป็น “DROP PACKET” ถ้าฟังก์ชันทั้งหมดส่งค่ากลับเป็น “ALLOW PACKET” แพ็คเก็ตเหล่านั้นจะสามารถผ่านไปได้



รูปที่ 5.2 แสดงขั้นตอนการทำงานของไฟร์วอลล์

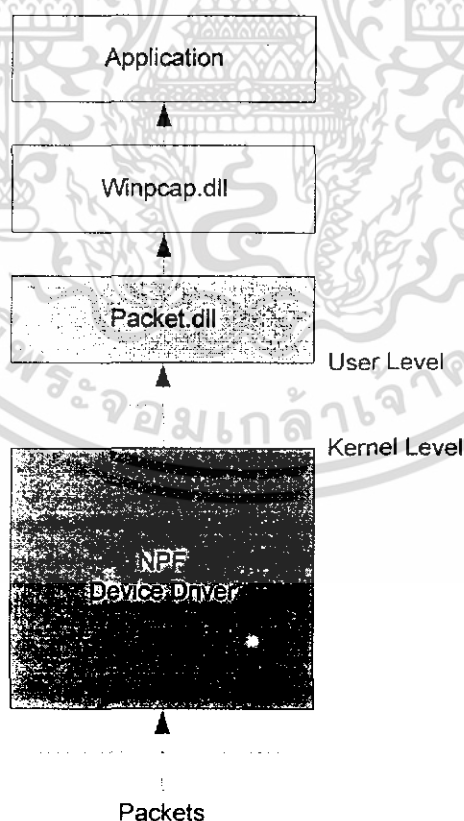
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการทำงานของไฟร์วอลล์ตามรูปที่ 5.2

- 1) เมื่อเครื่องได้รับแพ็คเกจเข้ามา โดยที่ IP Driver มีฟังก์ชันฟิลเตอร์อยู่ตามที่กำหนดไว้
- 2) IP Driver จะส่งแพ็คเกจนั้นเข้าไปโดย ผ่านเข้าไปยังฟังก์ชันฟิลเตอร์ตามลำดับไปเรื่อยๆ โดยรอค่าที่ส่งกลับออกมา
- 3) สมมุติฟังก์ชันแรกส่งค่ากลับเป็น “ALLOW PACKET” เมื่อ IP DRIVER ได้รับค่าส่งกลับจากฟังก์ชันแรกเป็น “ALLOW PACKET”
- 4) ดังนั้น IP DRIVER จะส่งแพ็คเกจนั้นไปที่ฟังก์ชันที่สองต่อไปในกรณีนี้
- 5) สมมุติให้ฟังก์ชันที่สองนี้ส่งค่ากลับเป็น “DROP PACKET” เมื่อ IP DRIVER ได้รับค่าส่งกลับจากฟังก์ชันที่สองเป็น “DROP PACKET” ดังนั้น IP DRIVER จะไม่ส่งแพ็คเกจนี้ต่อไปยังระบบ และจะไม่ส่งไปยังฟังก์ชันต่อไปอีก

### 5.3.2 ระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

ระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์จะอาศัยคุณสมบัติ และ ความสามารถของ WinPCap ซึ่งเป็นไลบรารีที่ทำการติดต่อกับการ์ดแลน เพื่อทำการควบคุมการทำงานของ การ์ดแลน และตรวจจับแพ็คเกจ เพื่อนำมาวิเคราะห์ผลว่ามีการ โจมตีเกิดขึ้นหรือไม่

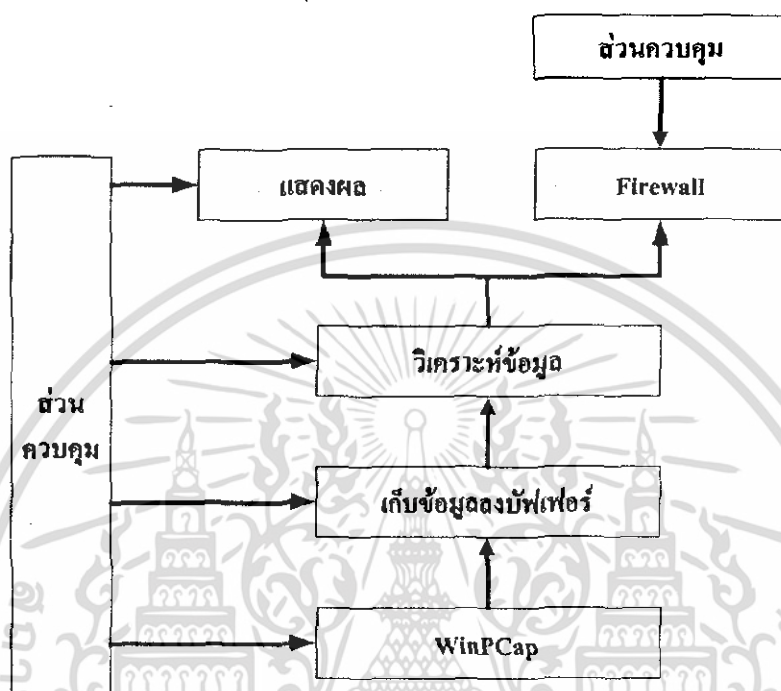


รูปที่ 5.3 แสดงระดับชั้นการทำงานของ WinPCap

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของระบบตรวจจับผู้บุกรุกนั้นจะแบ่งการทำงานออกเป็น 2 ส่วนหลักๆด้วยกัน คือ ส่วนการดักจับแพ็กเก็ตเก็บลงบัฟเฟอร์ และส่วนวิเคราะห์ผลแพ็กเก็ตที่ได้ดักจับมา เมื่อวิเคราะห์ผลเสร็จก็จะแสดงผล พร้อมกับส่งแพ็กเก็ตไปฟิลเตอร์ตามกฎที่กำหนดเอาไว้ดังรูปที่ 5.3

โดยการทำงานทั้ง 2 ส่วนนี้จะมีการแบ่งเชรด (Thread) การทำงานเพื่อที่จะสามารถดักจับแพ็กเก็ตและวิเคราะห์ผลไปได้พร้อมๆกัน



รูปที่ 5.4 โครงสร้างของระบบตรวจจับผู้บุกรุกทำงานร่วมกับไฟร์วอลล์

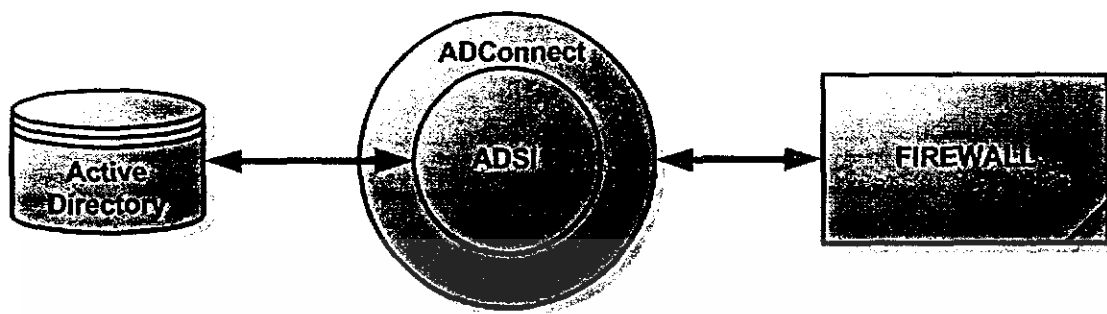
### 5.3.2 ระบบตรวจการใช้งานเครือข่ายของโปรเซส

ระบบตรวจการใช้งานเครือข่ายของโปรเซสจะทำงานคล้ายกับคำสั่ง `netstat -bn` โดยจะแสดง โปรเซสต่างๆที่กำลังใช้งานเครือข่ายอยู่ในขณะนั้น พร้อมทั้งบอกโปรเซสไอดี(PID) โปรโตคอล(Protocol) โลคอลแอดเดรส (Local Address) รีโมตแอดเดรส(Remote Address) และสถานะของการเชื่อมต่อ (State) โดยที่จะมีเวลาการอัปเดตข้อมูลตามเวลาที่ผู้ใช้ได้ตั้งเอาไว้

ระบบตรวจการใช้งานเครือข่ายของโปรเซสนี้มีความสามารถที่จะจบการทำงานของโปรเซส (Kill Connection) และ ปิดการเชื่อมต่อเครือข่ายของโปรเซสนั้นได้ (Close Connection) ถ้าหากพบว่ามีโปรเซสใดที่มีการใช้งานเครือข่ายที่น่าสงสัย ผู้ใช้ก็สามารถจบการทำงานของโปรเซสหรือปิดการเชื่อมต่อได้ทันที เพื่อให้ผู้ใช้สามารถป้องกันตนเองได้อีกทางหนึ่ง

เนื่องจากเพอร์ซันนอลไฟร์วอลล์ (Personal Firewall) จะมีความสามารถในการที่จะรับกฎเพื่อเริ่มการทำงานของไฟร์วอลล์จากเครื่องคอมพิวเตอร์ที่ได้ติดตั้งแอคทีฟไคเร็กทอรีไว้และจะมีการส่งล็อกไฟร์ไปเก็บไว้ยังแอคทีฟไคเร็กทอรีเมื่อมีการตรวจพบการโจมตี หรือ มีสิ่งผิดปกติ

เกิดขึ้น ซึ่งโครงสร้างการทำงานของส่วนที่ใช้ในการติดต่อกับเครื่องเซิร์ฟเวอร์ แสดงดังรูปที่ 6-6 โดยเพอร์ซันนอลไฟร์วอลล์จะมีการติดต่อ ไปยังเซิร์ฟเวอร์โดยผ่านกลาส ADConnect ที่สร้างขึ้น



รูปที่ 5.5 แสดงโครงสร้างส่วนติดต่อกับเซิร์ฟเวอร์

เพอร์ซันนอลไฟร์วอลล์จะติดตั้งอยู่ที่เครื่องลูกข่ายทุกเครื่องในระบบ โดยมีหน้าที่การทำงานดังนี้

- เป็นไฟร์วอลล์ชนิดแพ็คเกจฟิลเตอร์ริง (Packet Filtering Firewall)
- ตรวจสอบผู้บุกรุกทางเครือข่ายที่เข้ามาที่เครื่องลูกข่าย รวมทั้งสิ่งผิดปกติที่ไม่เป็นไปตามมาตรฐานของ โปรโตคอลทีซีพี (TCP) , ยูดีพี(UDP) , ไอพี(IP) , ไอซีเอ็มพี(ICMP)
- ตรวจสอบการเชื่อมต่อเครือข่ายของโปรเซสต่างๆ โดยผู้ใช้สามารถจบการทำงานของโปรเซสและปิดการเชื่อมต่อเครือข่ายของโปรเซสใดๆได้
- ส่งการแจ้งเตือนกลับไปยังล็อกมอนิเตอร์เมื่อตรวจจับ ได้ว่ามีการบุกรุกเกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 6

# การพัฒนาโปรแกรมไฟร์สกรีน

### 6.1 แนวคิด

โปรแกรมไฟร์สกรีนเป็นส่วนของไฟร์วอลล์ที่อยู่บนเกตเวย์ ซึ่งใช้งานระบบปฏิบัติการลินุกซ์ (Gnu/Linux) โดยสำหรับการเข้าใช้บริการของแอคทีฟไดเรกทอรีบนไฟร์สแตชันนั้นจะใช้ OpenLDAP ซึ่งเป็นมาตรฐานเปิดสำหรับใช้งานโปรโตคอล LDAP ซึ่งสามารถใช้งานได้ดีกับทุกแพลตฟอร์ม และในส่วนของการทำงานเชื่อมต่อซึ่งนับเป็นส่วนสำคัญของไฟร์วอลล์แล้วจะให้ความสามารถของเน็ตฟิลเตอร์/ไอทีเทเบิลส์ (Netfilter/Iptables) ซึ่งทำงานอยู่ในระดับเคอร์เนลและติดตั้งมากับระบบปฏิบัติการลินุกซ์อยู่แล้วตั้งแต่เคอร์เนลเวอร์ชัน 2.4.x ขึ้นมา

### 6.2 ขอบเขตและความสามารถ

ไฟร์สกรีนทำงานอยู่บนเกตเวย์ ซึ่งการเชื่อมต่อในทุกๆ ส่วนจะต้องผ่านทางเกตเวย์นี้ นั่นคือไฟร์สกรีนจะสามารถควบคุมการเชื่อมต่อเกือบทั้งหมดในระบบได้ โดยปกติแล้วเกตเวย์ไฟร์วอลล์จะมีลักษณะเป็นแบบ allow-based คือยอมให้แพ็คเก็ตผ่านไปได้อย่างหมด เว้นแต่ส่วนที่ถูกระบุมาตรงกับกฎที่ตั้งไว้ว่าจะไม่ให้ผ่านไป ส่วนการจะตั้งกฎในการที่จะปฏิเสธการเชื่อมต่อหรือแพ็คเก็ตใดบ้างขึ้นขึ้นกับการตั้งค่าของผู้ดูแลระบบผ่านทางไฟร์สแตชัน

### 6.3 การพัฒนาโปรแกรม

#### 6.3.1 การทำงานร่วมกับไฟร์สแตชัน

ไฟร์สกรีนมีระบบปฏิบัติการเป็นลินุกซ์ ซึ่งในที่นี้ใช้ดิสโทรเดเบียน (debian) โดยการทำงานร่วมกับแอคทีฟไดเรกทอรีของไมโครซอฟท์บนไฟร์สแตชันนั้นจะใช้แพ็คเกจของ OpenLDAP โดยตัวโปรแกรมไฟร์สกรีนจะต้องใช้ไลบรารีของ OpenLDAP ในการเขียนโปรแกรมทำงานร่วมกับแอคทีฟไดเรกทอรี บนระบบปฏิบัติการเดเบียนนั้นใช้แพ็คเกจชื่อ libldap2-dev ซึ่งสนับสนุนโปรโตคอล LDAP เวอร์ชัน 3 และทำงานร่วมกันได้ดีกับแอคทีฟไดเรกทอรี

/ Name	Version	Description
ii libldap2	2.1.30-12	OpenLDAP libraries
ii libldap2-dev	2.1.30-12	OpenLDAP development libraries

#### รูปที่ 6.1 แสดงแพ็คเกจที่จำเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 6.3.2 การติดต่อกับแอคทีฟไดเรกทอรี

**LDAP \*ldap\_init(host, port)**

char \*host;

int port;

ตารางที่ 6.1 พารามิเตอร์ของฟังก์ชัน ldap\_init()

host	ชื่อโฮสต์หรือไอพีแอดเดรสของเซิร์ฟเวอร์ (สามารถระบุหลายค่าได้โดยกันด้วยช่องว่าง) และสามารถระบุพอร์ตได้ดังนี้ <i>hostname:portnumber</i> ซึ่งจะถือว่าหมายเลขพอร์ตนี้มีความสำคัญกว่า <i>port</i> ซึ่งเป็นพารามิเตอร์ตัวถัดไป
port	หมายเลขพอร์ตที่ใช้ (ค่าปกติคือ 389 หรืออาจใช้ LDAP_PORT แทน)

หลังจากใช้ฟังก์ชัน ldap\_init() แล้วการเชื่อมต่อระหว่างไคลเอนต์และเซิร์ฟเวอร์ยังไม่ได้เริ่มขึ้นจริงๆ เพียงคืนค่ากลับมาเป็นแอสแตริชของการเชื่อมต่อนั้นๆเก็บไว้ในตัวชี้ไปยัง LDAP structure หากไม่มีข้อผิดพลาดเกิดขึ้นเท่านั้น ส่วนการเชื่อมต่อจริงๆจะเริ่มต้นขึ้นในขั้นตอนต่อไป และหากเกิดข้อผิดพลาดขึ้นจะคืนค่ามาเป็น NULL

**int ldap\_bind\_s(LDAP \*ld, const char \*who, const char \*cred, int method);**

ตารางที่ 6.2 พารามิเตอร์ของฟังก์ชัน ldap\_bind\_s()

ld	ตัวชี้ไปยัง LDAP structure ที่ทำงานด้วย
who	dn ของผู้ใช้ที่จะทำงานพิสูจน์ตนเข้าใช้งานแอคทีฟไดเรกทอรี
cred	รหัสผ่านสำหรับผู้ใช้คนนั้นๆ
method	รูปแบบสำหรับระบบการพิสูจน์ตน ปกติใช้เป็น LDAP_AUTH_SIMPLE

หลังจาก ldap\_init() แล้วก่อนที่จะกระทำการใดๆกับเซิร์ฟเวอร์ได้จะต้องทำการเริ่มการเชื่อมต่อและพิสูจน์ตนกับเซิร์ฟเวอร์เสียก่อน โดยผ่านทางฟังก์ชัน ldap\_bind\_s() ซึ่งจะคืนค่า 1 หากเกิดความผิดพลาดขึ้น

\_s ที่ต่อท้ายชื่อฟังก์ชันจะหมายถึงเป็นแบบ synchronous คือต้องรอให้ฟังก์ชันทำงานเรียบร้อยได้รับการตอบรับจากเซิร์ฟเวอร์จึงจะสามารถทำคำสั่งต่อไปได้

```
int ldap_unbind(LDAP *ld);
```

```
int ldap_unbind_s(LDAP *ld);
```

ldap\_unbind และ ldap\_unbind\_s ใช้สำหรับจบการเชื่อมต่อกับเซิร์ฟเวอร์ และจะลบค่าใน ld ทั้งหมด โดยทั้ง 2 ฟังก์ชันไม่ได้มีความแตกต่างกันแต่อย่างใด สามารถใช้ได้ไม่ว่าขณะสร้างการเชื่อมต่อได้ใช้ ldap\_bind หรือ ldap\_bind\_s ก็ตาม

*ตัวอย่างการเขียน โปรแกรมเชื่อมต่อกับแอคทีฟไดเรกทอรี*

```
include <ldap.h>

.....

char *ldap_host = "192.168.1.100";
char *root_dn = "cn=Administrator, cn=Users, dc=hephaestus, dc=com";
char *root_pw = "secret";

.....

LDAP *ld;

.....

/* ldap_init โดยระบุชื่อ โฮสต์และพอร์ต */
if((ld = ldap_init(ldap_host, LDAP_PORT)) == NULL ) {
    perror( "ldap_init failed!!!" );
    exit( EXIT_FAILURE );
}

/* เริ่มการเชื่อมต่อ โดยมี credential เป็น root_dn และ root_pw */
if(ldap_bind_s(ld, root_dn, root_pw, LDAP_AUTH_SIMPLE) != LDAP_SUCCESS ) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        ldap_perror( ld, "ldap_bind failed!!!" );
        exit( EXIT_FAILURE );
    }

    .....

    /* ยกเลิกการเชื่อมต่อ */
    ldap_unbind_s(ld);

    .....

```

### 6.3.3 การค้นหาข้อมูลในแอคทีฟไดเรกทอรี

การสืบค้นข้อมูลที่ต้องการบนแอคทีฟไดเรกทอรีเซิร์ฟเวอร์มี 3 ขั้นตอนด้วยกันคือ

1. ได้แต่ละเอนทรีจากผลลัพธ์การค้นหา
2. ได้แอตทริบิวต์ที่ต้องการจากแต่ละเอนทรี
3. ได้ค่าที่เก็บไว้ในแต่ละแอตทริบิวต์

โดยมีฟังก์ชันหลักๆที่ใช้งานดังนี้

```
int ldap_search_s(ld, base, scope, filter, attrs, attrsonly, res)
```

```
DAP *ld;
```

```
char *base;
```

```
int scope;
```

```
char *filter, *attrs[];
```

```
int attrsonly;
```

```
LDAPMessage **res;
```

ตารางที่ 6.3 พารามิเตอร์ของฟังก์ชัน ldap\_search\_s()

ld	ตัวชี้ไปยัง LDAP structure ที่ทำงานด้วย
base	DN ของเอนทรีที่จะเป็นจุดเริ่มต้นของการค้นหา
scope	ขอบเขตของการค้นหา <ul style="list-style-type: none"> <li>• LDAP_SCOPE_BASE ค้นหาเอนทรีทั้งหมดบน base</li> </ul>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	<ul style="list-style-type: none"> <li>LDAP_SCOPE_ONELEVEL ค้นหาเอนทรีทั้งหมดบน base และต่ำลงไป 1 ชั้น</li> <li>LDAP_SCOPE_SUBTREE ค้นหาเอนทรีทั้งหมดบน base และโคเรกทอรีภายใต้ทั้งหมด</li> </ul>
filter	ฟิลเตอร์ที่จะใช้กรองผลลัพธ์จากการค้นหา อยู่ในรูปแบบ attributetype=attributevalue
attrs	อาเรย์ของชนิดแอดทรีบิวต์ที่จะส่งคืนมาสำหรับเอนทรีที่เข้าข่าย
attrsonly	ระบุว่าค่าในแอดทรีบิวต์จะถูกส่งกลับมาด้วยกันหรือไม่ <ul style="list-style-type: none"> <li>0 หมายความว่าทั้งแอดทรีบิวต์และค่าของแอดทรีบิวต์นั้นถูกส่งกลับ</li> <li>1 หมายความว่าเพียงแอดทรีบิวต์เท่านั้นที่ถูกส่งกลับมา</li> </ul>
res	ผลลัพธ์จากการค้นหา

```
LDAPMessage *ldap_first_entry( LDAP *ld, LDAPMessage *result );
```

ตารางที่ 6.4 พารามิเตอร์ของฟังก์ชัน ldap\_first\_entry()

ld	ตัวชี้ไปยัง LDAP structure ที่ทำงานด้วย
result	ชุดของผลลัพธ์ในรูปแบบของ LDAPMessage structure

```
char *ldap_first_attribute( LDAP *ld, LDAPMessage entry, BerElement **berptr );
```

ตารางที่ 6.5 พารามิเตอร์ของฟังก์ชัน ldap\_first\_attribute()

ld	ตัวชี้ไปยัง LDAP structure ที่ทำงานด้วย
entry	ตัวชี้ไปยังเอนทรีที่คืนค่ามาจาก ldap_first_entry() โดยอยู่ในรูปแบบของ LDAPMessgae structure
berptr	ตัวชี้ไปยัง BerElement ใช้สำหรับการจดจำตำแหน่งของแอดทรีบิวต์ต่างๆในเอนทรีที่ได้มา

```
char **ldap_get_values(ld, entry, attr)
```

```
LDAP *ld;
```

```
LDAPMessage *entry;
```

```
char *attr
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.6 พารามิเตอร์ของฟังก์ชัน ldap\_get\_values()

ld	ตัวชี้ไปยัง LDAP structure ที่ทำงานด้วย
entry	ผลลัพธ์จาก ldap_search_s()
attr	ค่า (value) ในแอตทริบิวต์จาก ldap_first_attribute()

ตัวอย่างการค้นหาข้อมูลที่ต้องการบนแอคทีฟไดเรกทอรี

```

include <ldap.h>
include <lber.h>

.....

LDAP *ld;
char *base = "cn=computers, dc=hephaestus, dc=com";
char *filter = "firescreen";
char* attr = "firewallRule";
char** vals;
BerElement* ber;
LDAPMessage* msg;
LDAPMessage* entry;

.....

/* ldap_search_s จะคืนเอนทรีที่เข้าข่ายสำหรับขอบเขตการค้นหาและฟิลเตอร์ที่ระบุเป็น
ตัวชี้ในตัวแปร msg */
if (ldap_search_s(ld, base, LDAP_SCOPE_SUBTREE, filter, &attr, 0, &msg) !=
LDAP_SUCCESS) {
    ldap_perror( ld, "ldap_search_s failed!!!" );
    exit(EXIT_FAILURE);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* ลูป for สำหรับวนจนครบทุกๆเอนทรีในส่งกลับมาจาก ldap_search_s */
for (entry = ldap_first_entry(ld, msg); entry != NULL; entry = ldap_next_entry(ld,
entry)) {
    /* ลูป for สำหรับวนจนครบทุกแอตทริบิวต์ในแต่ละเอนทรี */
    for( attr = ldap_first_attribute(ld, entry, &ber);
        attr != NULL; attr = ldap_next_attribute(ld, entry, ber)) {
        /* ลูป for สำหรับวนจนครบทุกๆค่าในแต่ละแอตทริบิวต์ */

        if((vals = ldap_get_values(ld, entry, attr)) != NULL) {
            /* ค่าที่ต้องการจะอยู่ใน vals[i] */
        }
    }
}
.....

```

### 6.3.4 การเปลี่ยนแปลงแก้ไขข้อมูลบนแอคทีฟไดเรกทอรี

การเปลี่ยนแปลงแก้ไขข้อมูลที่ต้องการในแอคทีฟไดเรกทอรีเซิร์ฟเวอร์นั้นสามารถทำได้ผ่านฟังก์ชัน ldap\_modify\_s()

```
int ldap_modify_s(ld, dn, mods)
```

```
LDAP *ld;
```

```
char *dn;
```

```
LDAPMod *mods[];
```

ตารางที่ 6.7 พารามิเตอร์ของฟังก์ชัน ldap\_modify\_s()

ld	ตัวชี้ไปยัง LDAP structure ที่ทำงานด้วย
dn	DN ของเอนทรีที่จะทำการแก้ไขเปลี่ยนแปลง
mods	ตัวชี้ไปยังอาร์เรย์ของตัวชี้ไปยัง LDAPMod structure สำหรับระบุว่า จะทำการเปลี่ยนแปลงอย่างไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ตัวอย่างการเปลี่ยนแปลงแก้ไขข้อมูลบนแอดที่ไฟโดเรททอรี

```

include <ldap.h>

.....

LDAP *ld;
LDAPMod *list_of_attr[4];
LDAPMod attribute1, attribute2, attribute3;

char *dn = "cn=firescreen, cn=computers, dc=hephaestus, dc=com";

/* ค่าที่จะทำการแก้ไขเปลี่ยนแปลง */
char *homePhone_values[] = { "555-1212", NULL };
char *telephoneNumber_values[] = { "869-5309", NULL };

.....

/* ระบุรูปแบบการเปลี่ยนแปลงแก้ไขในแต่ละแอดทริบิวต์ */
attribute1.mod_type = "homePhone";
attribute1.mod_op = LDAP_MOD_ADD;
attribute1.mod_values = homePhone_values;
attribute2.mod_type = "telephoneNumber";
attribute2.mod_op = LDAP_MOD_REPLACE;
attribute2.mod_values = telephoneNumber_values;
attribute3.mod_type = "facsimileTelephoneNumber";
attribute3.mod_op = LDAP_MOD_DELETE;
attribute3.mod_values = NULL;

/* เก็บแต่ละค่าที่จะเปลี่ยนแปลงแก้ไขในอาร์เรย์ */
list_of_attr[0] = &attribute1;
list_of_attr[1] = &attribute2;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

list_of_attrs[2] = &attribute3;

list_of_attrs[3] = NULL;

.....

/* เปลี่ยนแปลงแก้ไขในแต่ละแอตทริบิวต์ */

if ( ldap_modify_s( ld, dn, list_of_attrs ) != LDAP_SUCCESS ) {
    ldap_perror( ld, "ldap_modify_s" );
    return( 1 );
}

.....

```

mod\_type, mod\_op และ mod\_values ถูกนิยามไว้ใน LDAPMod structure โดย mod\_type บ่งบอกถึงแอตทริบิวต์ที่ต้องการทำงานด้วยและ mod\_op คือรูปแบบการเปลี่ยนแปลงแก้ไข ประกอบไปด้วย LDAP\_MOD\_ADD, LDAP\_MOD\_REPLACE และ LDAP\_MOD\_DELETE ส่วน mod\_values ก็คือค่าที่จะทำการแก้ไขเปลี่ยนแปลง หาก mod\_op เป็น LDAP\_MOD\_DELETE นั้นก็จะต้องให้ค่า mod\_values เป็น NULL

### 6.3.5 การนำกฎจากไฟร์สแตนด์มาใช้งานไฟร์สกรีน

#### 6.3.5.1 การดึงกฎจากไฟร์สแตนด์

สำหรับกฎของไฟร์วอลล์ที่เก็บไว้ในแอตทริบิวต์ไฟร์สแตนด์นั้นถูกเก็บไว้ได้ dn คือ "cn=firescreen, cn=computers, dc=hephaestus, dc=com" ภายในแอตทริบิวต์ชื่อ firewallRule ดังนั้นจึงสามารถเข้าถึงได้โดยใช้ฟังก์ชัน ldap\_search\_s() กำหนดขอบเขตของการค้นหาเป็น "cn=firescreen, cn=computers, dc=hephaestus, dc=com" จากนั้นจึงดึงค่าออกจากแอตทริบิวต์ firewallRule จนครบทุกค่าและเก็บค่าไว้ใช้งานในอาเรย์

จากรูปแบบการทำงานของไฟร์สกรีนที่กล่าวถึงในหัวข้อก่อนหน้านี้ ไฟร์สกรีนจะคอยตรวจสอบที่ไฟร์สแตนด์ทุกระยะเวลาหนึ่ง เพื่อตรวจสอบว่ามีกฎถูกแก้ไขหรือไม่ ซึ่งหลังจากกฎถูกแก้ไขบนไฟร์สแตนด์จะทำการอัปเดตค่าในแอตทริบิวต์ชื่อ update เป็น "YES" ดังนั้นก่อนหน้านี้ที่จะรับกฎมาจากไฟร์สแตนด์จึงต้องทำการตรวจสอบแอตทริบิวต์ update ก่อน หากมีค่าเป็น "NO" ก็ จะจบการทำงานและรอเวลาวนกลับไปตรวจสอบอีกครั้ง และหากไฟร์สกรีนพบว่าค่าในแอตทริ

บิวต์ update เป็น "YES" ก็จะทำการดึงกฎมาจากแอตทริบิวต์ firewallRule จากนั้นจึงทำการแก้ไขค่าในแอตทริบิวต์กลับเป็น "NO" อีกครั้งผ่านทางฟังก์ชัน ldap\_modify\_s()

### 6.3.5.2 การนำมาใช้งานกับไอพีเทเบิลส์

กฎที่ดึงมาจากทางไฟร์สเตรนนั้นจะอยู่ในรูปแบบดังต่อไปนี้

SRC:MASK:DST:MASK:PROTOCOL:SPORT:DPORT:ICMP-TYPE:ZONE:TARGET

ตารางที่ 6.8 แสดงความหมายของแต่ละค่า

SRC	ไอพีแอดเดรสต้นทาง
MASK	เน็ตมาสก์ของไอพีแอดเดรสต้นทาง
DST	ไอพีแอดเดรสปลายทาง
MASK	เน็ตมาสก์ของไอพีแอดเดรสปลายทาง
PROTOCOL	โปรโตคอล ( TCP, UDP, ICMP หรือ any)
SPORT	พอร์ตต้นทาง (เมื่อระบุโปรโตคอลเป็น TCP หรือ UDP เท่านั้น)
DPORT	พอร์ตปลายทาง (เมื่อระบุโปรโตคอลเป็น TCP หรือ UDP เท่านั้น)
ICMP-TYPE	หมายเลข ICMP-TYPE (เมื่อระบุโปรโตคอลเป็น ICMP เท่านั้น)
ZONE	โซนที่กฎนี้จะมีผลในการใช้งาน
TARGET	ปลายทางของแพ็คเก็ตที่เมซท์กับกฎนี้

โดยก่อนที่จะสามารถนำมาใช้งานกับไฟร์สเตรนได้จะต้องแปลงกฎที่รับมาจากไฟร์สเตรนที่มีรูปแบบดังกล่าวให้อยู่ในรูปแบบของไอพีเทเบิลส์เสียก่อน โดยผ่านฟังก์ชันแปลง (parser) ซึ่งจะแยกแต่ละส่วนออกจากกัน โดยมีเครื่องหมายโคลอน ( : ) เป็นตัวแบ่ง โดยนำเอาแต่ละส่วนมาต่อเข้าด้วยกันตามรูปแบบคำสั่งของไอพีเทเบิลส์โดยขึ้นต้นด้วย iptables -A (กฎจะถูกเพิ่มต่อท้ายเรื่อยๆ ตามลำดับ) ส่วนกฎจะถูกเพิ่มเข้าเชน (CHAIN) โดยของไอพีเทเบิลส์จะพิจารณาจาก ZONE และ TARGET ที่ระบุเป็น DROP ทั้งหมดจะถูกระบุ TARGET ในคำสั่งไอพีเทเบิลส์เป็น -j LOG-CHAIN ซึ่งจะส่งแพ็คเก็ตที่ตรงกับกฎเข้าไปยังเชนชื่อ LOG-CHAIN เพื่อที่หากแพ็คเก็ตที่ถูกละทิ้ง (DROP) จะมีการเก็บข้อมูลลงล็อกไฟล์

```
Chain LOG-CHAIN (4 references)
num target prot opt source destination
1 LOG all -- anywhere anywhere limit: avg 1/sec
burst 5 LOG level info prefix 'FIRESCREEN '
2 DROP _ all -- anywhere anywhere
```

รูปที่ 6.2 กฎในเชน LOG-CHAIN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กฎแรกจะทำการเก็บข้อมูลทุกแพ็คเกจที่ผ่านเข้ามาในเซสนี้ลงล็อกไฟล์ในอัตรา 1 แพ็คเกจต่อวินาที (limit: avg 1/sec burst 5) โดยทุกๆค่าที่ถูกเก็บลงล็อกไฟล์จะมีคำว่า "FIRESCREEN " (prefix 'FIRESCREEN ') ขึ้นต้น เพื่อประโยชน์ในการคัดแยกล็อกในภายหลัง เนื่องจากกระบวนการเก็บเหตุการณ์ล็อกไฟล์นั้นถูกจัดการโดย syslogd ซึ่งไม่ได้ถูกจัดการโดยตัวไอฟีเทเบิลส์เองทำให้ในล็อกไฟล์จะเก็บเหตุการณ์ที่มีระดับของการล็อกเหมือนกันเอาไว้ โดยในเซส LOG-CHAIN นี้ระดับของการล็อกเป็น info (LOG level info)

กฎที่สองจะทำการละทิ้งหมดทุกอย่างแพ็คเกจ เนื่องด้วยแพ็คเกจที่ถูกส่งต่อมายังเซสนี้เพื่อเก็บข้อมูลล็อกไฟล์คือแพ็คเกจที่ถูกกำหนด TARGET จากไฟร์สเตชันไว้เป็น DROP นั่นเอง

ตัวอย่างการแปลงกฎจากไฟร์สเตชันมาเป็นคำสั่งของไอฟีเทเบิลส์

กฎจากไฟร์สเตชัน

```
192.168.2.10:255.255.255.255:192.168.1.2:255.255.255.255:TCP:ANY:22:-:DMZ-in:DROP
```

คำสั่งไอฟีเทเบิลส์

```
iptables -A FORWARD -s 192.168.2.10 -d 192.168.1.2 -p tcp -dport 22 -j LOG-CHAIN
```

กฎจากไฟร์สเตชัน

```
192.168.2.7:255.255.255.255:-:-:ANY:-:-:LOCAL-in:ACCEPT
```

คำสั่งไอฟีเทเบิลส์

```
iptables -A INPUT -s 192.168.2.7 -j ACCEPT
```

สุดท้ายเมื่อได้กฎจากไฟร์สเตชันอยู่ในรูปแบบคำสั่งของไอฟีเทเบิลส์ทั้งหมดแล้วก็จะทำการเขียนทั้งหมดลงบนไฟล์เชลล์สคริปต์ เพื่อทำการรันเพื่อใช้งานตามคำสั่งไอฟีเทเบิลส์ดังกล่าว รวมทั้งเก็บไฟล์นี้ไว้ใช้ครั้งต่อไปหากโปรแกรมถูกปิดหรือเครื่องไฟร์สกรีนเริ่มทำงานใหม่เพื่อที่จะสามารถคงกฎเดิมไว้ได้

### 6.3.6 การส่งล็อกกลับไปยังไฟร์สเตชัน

ไฟร์สเตชันจะมีการเก็บล็อกไฟล์สำหรับแพ็คเกจที่ถูกละทิ้งไปตามกฎที่ได้กำหนดไว้เอาไว้ เพื่อใช้ประกอบการพิจารณาป้องกันการบุกรุกที่เกิดขึ้นหรือความผิดปกติที่อาจเกิดขึ้น ซึ่งล็อกไฟล์จะถูกเก็บลงไฟล์บนเครื่องไฟร์สกรีนเองก่อน จากนั้นจึงค่อยส่งต่อไปเก็บยังศูนย์กลางที่ไฟร์สเตชันต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ล็อกไฟล์ในส่วนของไอพีเทเบิลส์นั้นไม่ได้ถูกเก็บไว้โดยตัวไอพีเทเบิลส์เองโดยตรง แต่ผ่านโปรแกรมเก็บล็อกมาตรฐานบนระบบปฏิบัติการยูนิกซ์อย่าง syslogd ล็อกไฟล์ในประเภทเดียวกันก็จะถูกเก็บไว้ในไฟล์เดียวกัน โดยปกติล็อกในส่วนของไอพีเทเบิลส์จะถูก syslogd เก็บไว้ที่ /var/log/kern.log แต่สำหรับบนตัวไฟร์สกรีนเองนั้นได้แยกล็อกไฟล์ในส่วนนี้เก็บแยกออกไปต่างหากที่ /var/log/firescreen.log อีกส่วนด้วย ซึ่งแยกออกตามระดับการล็อก โดยได้กำหนดระดับของล็อกไว้ที่ info อย่างไรก็ตามล็อกในส่วนของเคอร์เนลที่มีระดับเป็น info เช่นกันก็จะยังถูกเก็บลงยังไฟล์ firescreen.log นี้ด้วยเช่นกัน ดังนั้นแล้วขณะที่ทำการล็อกจะกำหนดคำนำหน้า (prefix) ในส่วนของล็อกจากไอพีเทเบิลส์เป็น "FIRESCREEN " เพื่อความง่ายในการแยกล็อกของไอพีเทเบิลส์ออกจากล็อกจากส่วนอื่นๆ

```
Jan 10 16:00:45 firescreen kernel: NET: Registered protocol family 10
Jan 10 16:00:45 firescreen kernel: Disabled Privacy Extensions on device c02cc960(lo)
Jan 10 16:00:45 firescreen kernel: IPv6 over IPv4 tunneling driver
Jan 10 16:01:53 firescreen kernel: FIRESCREEN IN=eth0 OUT=eth1 SRC=161.246.5.11
DST=192.168.1.100 LEN=40 TOS=0x00 PREC=0x00 TTL=1 ID=38473 PROTO=UDP SPT=38469
DPT=33438 LEN=20
Jan 10 16:01:58 firescreen kernel: FIRESCREEN IN=eth0 OUT=eth1 SRC=161.246.5.11
DST=192.168.1.100 LEN=40 TOS=0x00 PREC=0x00 TTL=1 ID=38474 PROTO=UDP SPT=38469
DPT=33439 LEN=20
```

### รูปที่ 6.3 แสดงล็อกที่ถูกเก็บไว้โดย syslogd

รูปที่ 6.3 แสดงให้เห็นข้อมูลในล็อกไฟล์ /var/log/firescreen.log ซึ่งกำหนดไว้ในไฟล์การตั้งค่าของ syslogd เป็น kern.info -/var/log/firescreen.log จะพบว่าไม่เฉพาะเพียงแต่ล็อกในส่วนของไอพีเทเบิลส์เท่านั้นที่ถูกเก็บไว้ในไฟล์นี้แต่จะมีล็อกจากเหตุการณ์อื่นๆที่อยู่ในระดับ info รวมอยู่ด้วยเช่นกัน ล็อกของไอพีเทเบิลส์เองสามารถสังเกตได้จากคำนำหน้า "FIRESCREEN " หลังส่วนต้นที่แสดงเวลาและชื่อโฮสต์

โปรแกรมจะคอยอ่านล็อกไฟล์ดังกล่าวเป็นระยะและส่งกลับไปเก็บยังแอคทีฟไดเรกทอรีบนไฟร์สแตชัน จากนั้นจะทำการลบข้อมูลทั้งไฟล์ทิ้งเพื่อตัดปัญหาที่จะส่งล็อกเดิมไปอีกครั้งหนึ่ง โดยภายใต้ dn "cn=firescreen, cn=computers, dc=hephaestus, dc=com" ในแอคทีฟไดเรกทอรีบนไฟร์สแตชันนั้นจะมีแอคทีฟไดเรกทอรีชื่อ firewallLog อยู่สำหรับเก็บล็อกที่ส่งมาจากไฟร์สกรีน อนึ่ง การเพิ่มล็อกเข้าไปยังแอคทีฟไดเรกทอรีดังกล่าวสามารถทำได้ผ่านฟังก์ชัน ldap\_modify\_s() ดังกล่าวถึงในหัวข้อก่อนหน้านี้

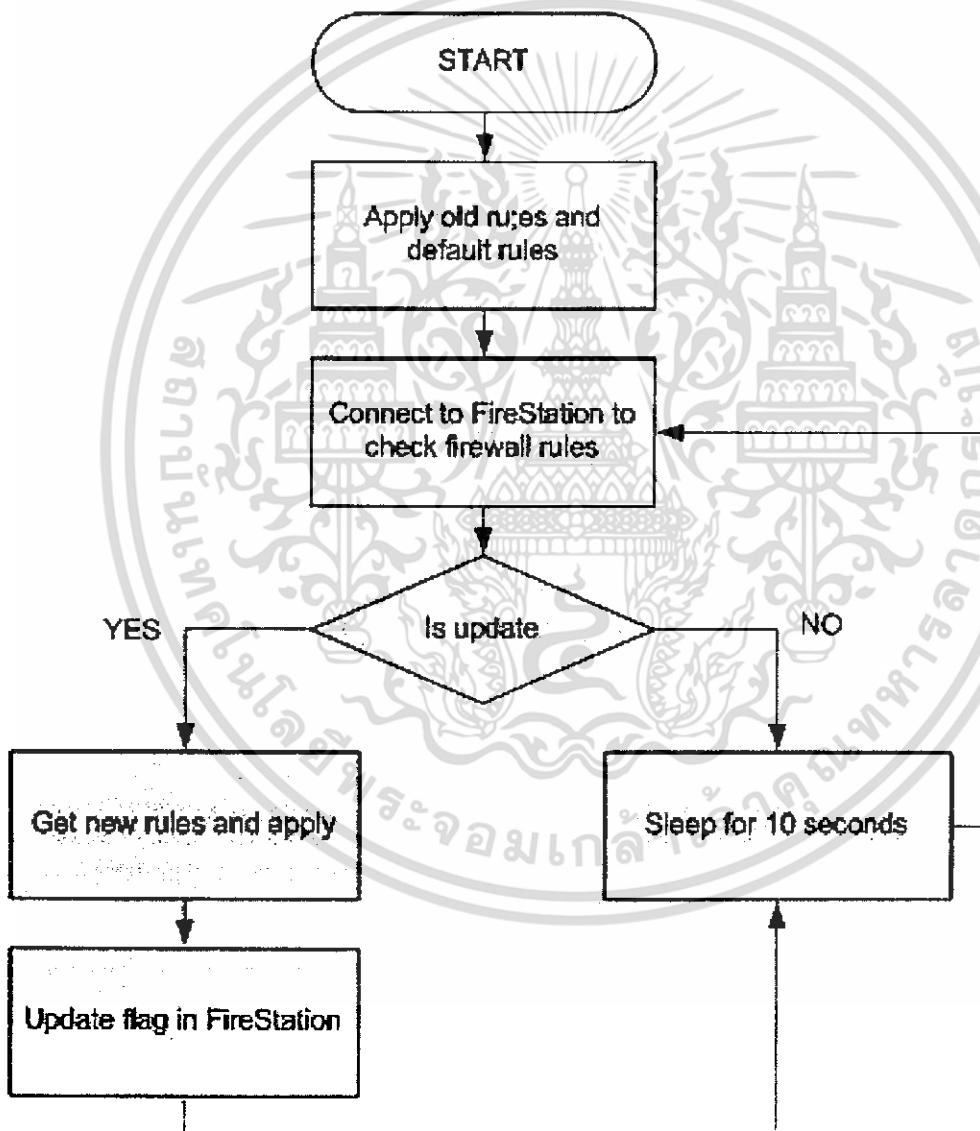
## 6.4 การทำงานของโปรแกรม

### 6.4.1 ลักษณะการทำงานของไฟร์สกรีน

การทำงานของไฟร์สกรีนจะเป็นในลักษณะของโปรแกรมเดมอน (daemon) คือทำงานอยู่เบื้องหลังตลอดเวลาตั้งแต่เครื่องไฟร์สกรีน (เกตเวย์) เริ่มทำงานขึ้นมา โดยการทำงานนั้นจะแบ่งออกเป็น 2 ส่วนหลักคือ ส่วนที่ทำหน้าที่รับกฎจากไฟร์สเตชันมาใช้ และส่วนของการส่งสื่อกลับไปยังไฟร์สเตชัน

#### 1. ส่วนที่ทำหน้าที่รับกฎจากไฟร์สเตชัน

โปรแกรมในส่วนนี้จะทำการนำเอาที่ถูกกำหนดไว้ที่ไฟร์สเตชันซึ่งถูกเก็บไว้ในแอดที่ไฟโดเรกทอรีมาใช้งานกับไอพีเทเบิลส์ สามารถแสดงลักษณะการทำงานเป็นแผนภาพดังนี้



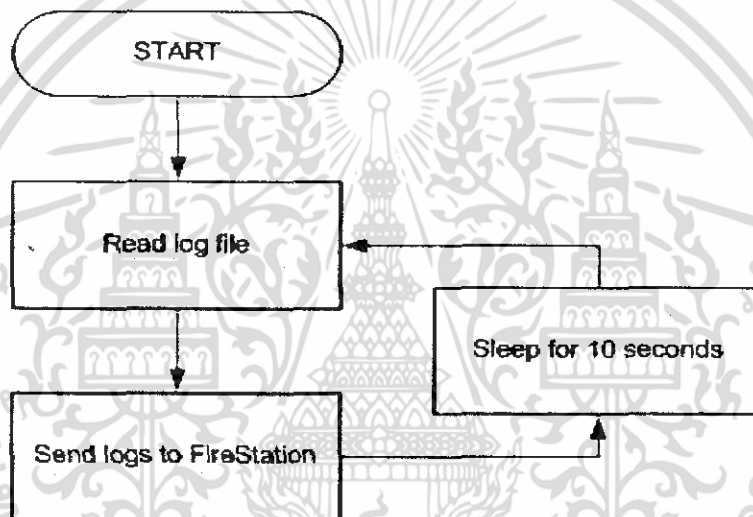
รูปที่ 6.4 แสดงการรับกฎของไฟร์สกรีน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเริ่มทำงานขึ้นมาครั้งแรกไฟร์สกรีนจะทำการตั้งกฎพื้นฐานที่ถูกกำหนดไว้แล้วให้กับ ไอพีเทเบิลส์ก่อน และจากนั้นจึงตรวจสอบคูกฎที่ถูกเก็บไว้เดิมและนำมาใช้งานกับไอพีเทเบิลส์เป็นที่เรียบร้อยเสียก่อน จากนั้นจึงตรวจสอบที่ไฟร์สเดชั่นว่าจากที่รับกฎมาครั้งล่าสุดได้มีการเปลี่ยนแปลงแก้ไขหรือไม่ หากมีก็จะนำกฎที่กำหนดใหม่มาใช้ หากไม่มีก็จะหยุดระยะเวลาหนึ่งแล้วจึงคอยตรวจสอบที่ไฟร์สเดชั่นอีกเรื่อยๆ (polling)

## 2. ส่วนที่ทำหน้าที่ส่งล็อกไปยังไฟร์สเดชั่น

โปรแกรมในส่วนนี้จะทำการอ่านล็อกไฟล์จากเครื่องเกตเวย์และส่งไปเก็บยังแอดที่ไฟโดเรทอริบนไฟร์สเดชั่น สามารถแสดงลักษณะการทำงานเป็นแผนภาพดังนี้



รูปที่ 6.5 แสดงการส่งล็อกของไฟร์สกรีน

เมื่อเริ่มการทำงานขึ้นมาโปรแกรมจะคอยตรวจสอบล็อกไฟล์เป็นระยะ และส่งล็อกที่เกี่ยวข้องในส่วนของไอพีเทเบิลส์กลับไปยังไฟร์สเดชั่น

#### 6.4.2 กฎพื้นฐานบนไฟร์สกรีน

ไฟร์สกรีนเองจะต้องมีการทำงานร่วมกันกับไฟร์สแตชันในการรับกฎและส่งสื่อกลับไปคั้งนั้นกฎที่ใช้งานบนไฟร์สกรีนเมื่รับมาจากตัวไฟร์สแตชัน แต่บนตัวไฟร์สกรีนเองโปรแกรมจะต้องมีการตั้งกฎพื้นฐานไว้ส่วนหนึ่งก่อนหน้ากฎอื่นๆเพื่อเป็นการป้องกันปัญหาที่อาจเกิดขึ้นหากมีการตั้งกฎที่เป็นปัญหาการทำงานขึ้นในภายหลัง เช่นอาจทำให้ไฟร์สกรีนไม่สามารถทำการติดต่อไปยังไฟร์สแตชันได้ เป็นต้น นอกเหนือไปจากนั้นยังเป็นเพื่อรับประกันความปลอดภัยพื้นฐานให้กับตัวไฟร์สกรีนและตัวระบบเองด้วยเช่นกัน

รองรับการสามารถติดต่อเข้าไปยังไฟร์สแตชันจากเครื่องไคลเอนต์ภายใน

```
iptables -A FORWARD -s $LOCAL -d $FIRESTATION -p tcp --dport 686 -i $LOCAL_INTERFACE -o $DMZ_INTERFACE -j ACCEPT
iptables -A FORWARD -s $FIRESTATION -d $LOCAL -p tcp -m state --state ESTABLISHED,RELATED -i $DMZ_INTERFACE -o $LOCAL_INTERFACE -j ACCEPT
```

รองรับการติดต่อกลับจากไฟร์สแตชัน

```
iptables -A FORWARD -s $DMZ -m state --state ESTABLISHED,RELATED -i $DMZ_INTERFACE -j ACCEPT
```

ป้องกันการโจมตีจาก syn flood โดยจำกัดปริมาณการเชื่อมต่อที่มี TCP flag เป็น syn ไว้ที่ 5 ครั้งต่อ 1 วินาที

```
iptables -A FORWARD -p tcp --syn -m limit --limit 5/second -j ACCEPT
```

ละทิ้ง (DROP) แพ้ก็เกิดที่พบว่า invalid ทั้งหมด

```
iptables -A FORWARD -s 224.0.0.0/4 -j DROP
iptables -A FORWARD -s 240.0.0.0/5 -j DROP
iptables -A FORWARD -s 127.0.0.0/8 -j DROP
iptables -A FORWARD -s 169.254.0.0/16 -j DROP
iptables -A FORWARD -d 224.0.0.0/4 -j DROP
iptables -A FORWARD -d 240.0.0.0/5 -j DROP
iptables -A FORWARD -d 127.0.0.0/8 -j DROP
iptables -A FORWARD -d 169.254.0.0/16 -j DROP
iptables -A FORWARD -d 255.255.255.255 -j DROP
iptables -A FORWARD -m state --state INVALID -j DROP
```

รองรับการเชื่อมต่อเข้ามายัง localhost ทั้งหมด

```
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
```

กฎที่กำหนดให้กับเซน FORWARD ทั้งหมดจะต้องใส่ให้กับเซน INPUT และ OUTPUT เพื่อรองรับการติดต่อระหว่างไฟร์สกรีนเองกับไฟร์สเตรชั่นและป้องกันความปลอดภัยของตัวไฟร์สกรีนเองด้วยเช่นกัน

## บทที่ 7

# การพัฒนาโปรแกรมไฟร์เบรก

### 7.1 แนวคิด

ไฟร์เบรก (FireBreak) เป็นโปรแกรมซีเคียวริตีเว็บพรีอ็อกซี่สำหรับเพิ่มความปลอดภัยให้กับเว็บเซิร์ฟเวอร์ซึ่งอยู่ในโซน DMZ ใช้ความสามารถของโปรแกรม squid-cache ในการทำงานเป็นรีเวิร์สเว็บพรีอ็อกซี่ทั่วไปและเพิ่มความสามารถในส่วนของระบบการตรวจจับการบุกรุกในระดับของแอปพลิเคชันของโปรแกรม snort\_inline ซึ่งสามารถเก็บล็อกไฟล์ได้จะถูกส่งไปยังดาต้าเบสของเครื่องแม่ข่าย

### 7.2 ขอบเขตและความสามารถ

โปรแกรมนี้สามารถตรวจสอบแพ็คเก็ตที่จะเข้าสู่เว็บเซิร์ฟเวอร์ที่อยู่ข้างหลังได้ทำให้ผู้ที่ต้องการติดต่อกับเว็บเซิร์ฟเวอร์ไม่สามารถติดต่อได้โดยตรง ดังนั้นการที่จะโจมตีเว็บเซิร์ฟเวอร์โดยตรงจึงไม่สามารถทำได้ ทำให้การโจมตีที่เกิดขึ้นนั้นเกิดขึ้นกับตัวเครื่องที่ทำหน้าเป็นซีเคียวริตีเว็บพรีอ็อกซี่นี้แทน ซึ่งโปรแกรมนี้เป็นการรวมความสามารถของโปรแกรม Squid และ Snort\_inline เพื่อเพิ่มความสามารถในการป้องกันการโจมตีหรือการบุกรุกต่างๆที่อาจเกิดขึ้นได้

แต่ความสามารถในการตรวจสอบของโปรแกรมนี้ยังมีข้อจำกัดอยู่บ้างเนื่องจากการที่จะหวังพึ่งแต่ความสามารถของ Snort\_inline ในการตรวจสอบการบุกรุกทางแอปพลิเคชันหรือทางเว็บนั้นแต่เพียงอย่างเดียว คงไม่อาจทำให้เว็บเซิร์ฟเวอร์ หรือตัวเครื่องที่รัน โปรแกรมนี้มีความปลอดภัยได้ทั้งหมด เนื่องจากการบุกรุกทางแอปพลิเคชันนั้นก็ได้มีการพัฒนาและมีรูปแบบต่างๆมากมาย ทำให้ต้องมีการอัปเดตตัว rule ของ Snort\_inline อยู่เสมอ และขึ้นอยู่กับความเร็วของเว็บเซิร์ฟเวอร์เองด้วยว่ามีการเขียนป้องกันการบุกรุกไว้ด้วยหรือไม่

### 7.3 การพัฒนาโปรแกรม

โปรแกรมไฟร์เบรกเป็นสามารถแบ่งการทำงานได้ออกเป็น 2 ส่วนคือ

1. โปรแกรม Squid ที่ทำเป็นรีเวิร์สเว็บพรีอ็อกซี่
2. โปรแกรม Snort\_inline ที่ช่วยเพิ่มความสามารถในการป้องกันให้กับโปรแกรมนี้

### 7.3.1 โปรแกรม Squid

สำหรับความสามารถในการเป็นรีเวิร์สเว็บพรีอกซ์นี้ จะใช้ Squid-cache ซึ่งเป็นโปรแกรมแบบเปิดเผยแพร่โค้ดสำหรับใช้งานเป็นเว็บพรีอกซ์ที่แพร่หลายโปรแกรมหนึ่ง ที่สามารถนำมาประยุกต์ใช้เป็นรีเวิร์สเว็บพรีอกซ์สำหรับเว็บเซิร์ฟเวอร์ได้

สามารถดาวน์โหลดซอร์สโค้ดเวอร์ชันล่าสุดของ Squid-cache ได้จากเว็บ <http://www.squid-cache.org/> ซึ่งเวอร์ชันที่นำมาใช้นี้คือ เวอร์ชัน 2.5

Squid-cache มีลักษณะการตั้งค่าการใช้งาน โดยต้องทำการแก้ไขในไฟล์ squid.conf ซึ่งอยู่ที่ `/etc/squid/squid.conf` ซึ่งในไฟล์มีคอมเมนต์อธิบายการใช้งานคำสั่งที่เกี่ยวข้องกับการตั้งค่าของ squid-cache ไว้เรียบร้อยแล้ว สำหรับการตั้งค่าของ squid-cache ในไฟล์ squid.conf ที่ใช้งานในโครงการนี้มีดังนี้

```
http_port 80 # reverse webproxy port
visible_hostname 192.168.1.1 # IP Address of webserver
httpd_accel_host 192.168.1.1 # IP Address of webserver
httpd_accel_port 80 # webserver port
http_access allow all # permit any IP to connection
```

คำอธิบายการใช้งานในส่วนต่างๆของไฟล์ squid.conf

- http\_port
 

ใช้กำหนดพอร์ตที่จะติดต่อเข้ามาซึ่งพรีอกซ์มีค่าเริ่มต้นเป็น 3128 สำหรับรีเวิร์สเว็บพรีอกซ์ http\_port ก็ต้องกำหนดค่าเป็น 80 ซึ่งเป็นพอร์ตทั่วไปในการใช้งาน HTTP
- visible\_hostname
 

ใช้สำหรับระบุโฮสต์ที่เป็นเว็บเซิร์ฟเวอร์เมื่อเกิด error\_message ในการทำงานของ squid กรณีที่ไม่สามารถระบุได้ว่า โฮสต์ที่เป็นเว็บเซิร์ฟเวอร์คือเครื่องใด
- httpd\_accel\_host
 

ใช้ระบุโฮสต์ที่เป็นเว็บเซิร์ฟเวอร์จริงๆ ในการทำเป็น Transparent Proxy
- httpd\_accel\_port
 

ใช้ระบุพอร์ตที่ใช้งานของเว็บเซิร์ฟเวอร์ ในการทำเป็น Transparent Proxy
- http\_access
 

ใช้ระบุสิทธิ์ผู้ที่สามารถใช้งานรีเวิร์สเว็บพรีอกซ์นี้ได้ โดยค่าเริ่มต้นจะเป็น deny all แต่การ ใช้งานเป็นรีเวิร์สเว็บพรีอกซ์จะรองรับบริการทั้งจากภายในและจากอินเทอร์เน็ตเองจะสามารถ เข้าถึงได้ จึงต้องเปลี่ยนเป็น allow all

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนความสามารถในการ cache ของ squid นั้นไม่ได้นำมาใช้ในโครงการนี้เนื่องจากเว็บเซิร์ฟเวอร์ที่อยู่หลังรีเวิร์สเว็บพรีอ็อกซีนั้นมีเพียงเครื่องเดียวจึงขอไม่กล่าวถึงการตั้งค่าในส่วนนี้

### 7.3.2 โปรแกรม Snort\_inline

สำหรับความสามารถของระบบการตรวจจับผู้บุกรุกนั้นจะใช้ snort\_inline ซึ่งเป็น Intrusion Prevention System (IPS) ที่มีความสามารถในการตรวจจับแพ็กเก็ตที่เป็นอันตรายต่อเว็บเซิร์ฟเวอร์และสามารถรีอปแพ็กเก็ตนั้นทิ้งได้ทันที โดย snort\_inline นี้เป็น โปรแกรมแบบเปิดเผยแพร่ฟรี

สามารถดาวน์โหลดซอร์สโค้ดเวอร์ชันล่าสุดของ Squid-cache ได้จากเว็บ

[http://prdownloads.sourceforge.net/snort-inline/snort\\_inline-2.3.0-RC1.tar.gz?download](http://prdownloads.sourceforge.net/snort-inline/snort_inline-2.3.0-RC1.tar.gz?download) ซึ่งโครงการนี้ใช้ snort\_inline เวอร์ชัน 2.3.0-RC1

โดยการติดตั้ง snort\_inline สามารถดูคู่มือที่ช่วยในการติดตั้งและทดสอบการทำงานเบื้องต้นของ snort\_inline ได้ที่ <http://linuxgazette.net/117/savage.html>

หลังจากการติดตั้ง snort\_inline แล้วก็จะต้องทำการตั้งค่าการใช้งานที่เหมาะสมให้กับ snort\_inline ได้ที่ไฟล์ snort\_inline.conf ซึ่งอยู่ที่ /etc/snort\_inline/snort\_inline.conf โดยมีการตั้งค่าที่ต้องแก้ไขเพิ่มเติมดังต่อไปนี้

ข้อความเดิมที่มีอยู่ในไฟล์ snort\_inline.conf ในบรรทัดเหล่านี้

```
var RULE_PATH /etc/snort_inline/drop_rules
output alert_full : snort_inline-full
output alert_fast : snort_inline-fsat
```

แก้ไขเป็นข้อความใหม่ดังนี้

```
var RULE_PATH /etc/snort_inline/rules
#output alert_full : snort_inline-full
#output alert_fast : snort_inline-fast
output database : log,mysql, dbname=snort user=snort password=123 host 192.168.1.100
```

โดยในเครื่องแม่ข่ายจะต้องมีการลง mysql เวอร์ชัน 4.1 ที่มีการสร้าง database ชื่อ snort ในดาต้าเบสนี้สร้าง table ต่างๆโดยใช้ไฟล์ create\_mysql ของ snort\_inline และมีการสร้าง user ใหม่ที่รองรับการใช้งานของดาต้าเบส snort ชื่อ snort ที่กำหนด password ในการใช้งานคือ 123 โดยต้องกำหนดสิทธิ์ในการใช้งานของ user snort ให้สามารถทำการ select, insert ดาต้าเบส snort ได้

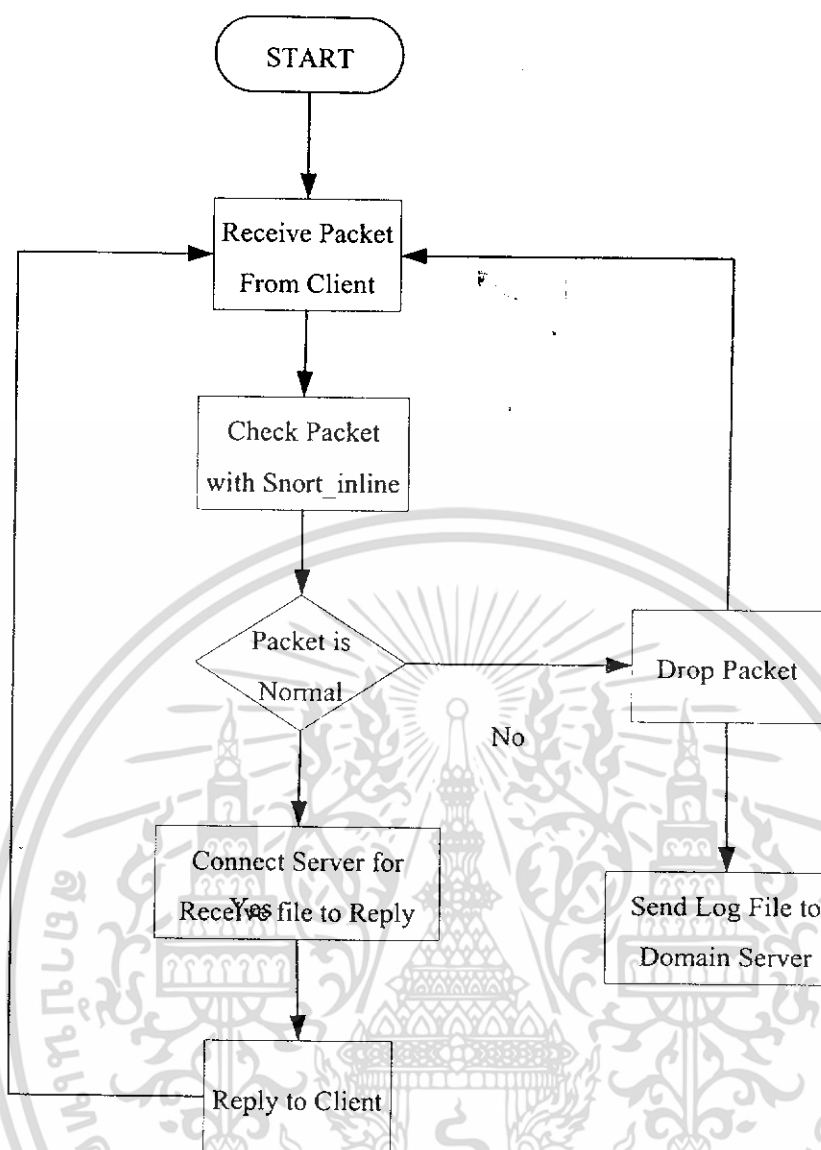
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นทำการอัปเดต signature หรือ rule ของ snort\_inline ให้มีความสามารถในการป้องกันเว็บเซิร์ฟเวอร์ได้ โดยทำการโหลด rule ใหม่ ๆ สำหรับ snort\_inline ได้ที่ <http://www.bleedingsnort.com/> สำหรับ ไฟล์ rule ที่สำคัญในการป้องกันเว็บเซิร์ฟเวอร์ คือ bleeding-web.rules, bleeding-exploit.rules, bleeding-scan.rules, bleeding-attack response.rules ซึ่งเมื่อทำการโหลดไฟล์เหล่านี้มาแล้วต้องมีการแก้ไขไฟล์ snort\_inline.conf ให้รองรับการทำงานของ rules ใหม่เหล่านี้โดยเพิ่มการตั้งค่าดังนี้ในส่วนของ drop rules

```
include $RULE_PATH/bleeding-web.rules
include $RULE_PATH/bleeding-exploit.rules
include $RULE_PATH/bleeding-scan.rules
include $RULE_PATH/bleeding-attack response.rules
```

#### 7.4 การทำงานของโปรแกรม

สำหรับการทำงานของโปรแกรมไฟร์วอลล์นี้เป็น ไปดังรูปที่ 7.1 คือ หลังจากทำการตั้งค่าการใช้งานและทำการเปิดเซอวิสเริ่มการใช้งานของ Squid และ Snort\_inline แล้วก็จะเท่ากับอยู่ในสถานะ START ในรูปที่ 7.2 เพื่อรอรับแพ็คเก็ตที่มาจากไคลเอ็นต์ที่ต้องการจะติดต่อกับเว็บเซิร์ฟเวอร์ที่อยู่หลังจากเครื่องรีเวิร์สเว็บพร็อกซีนี้ เมื่อได้รับแพ็คเก็ต Snort\_inline จะทำการตรวจสอบแพ็คเก็ตนั้นกับ rule ที่มีอยู่ว่าเข้าข่ายพฤติกรรมที่ผิดปกติหรือไม่ ถ้าแพ็คเก็ตนั้นผิดปกติ Snort\_inline จะทำการดรอปแพ็คเก็ตนั้นทิ้งและทำการบันทึกงล็อกไฟล์ส่งกลับไปยังเครื่องแม่ข่าย แต่ถ้าแพ็คเก็ตนั้นเป็นแพ็คเก็ตปกติจะทำการติดต่อกับเว็บเซิร์ฟเวอร์เพื่อรับไฟล์ที่จะต้องทำการตอบกลับให้กับไคลเอ็นต์ถ้าต้องมีการตอบกลับ แล้วจึงจะทำการส่งไฟล์ที่ได้รับจากเว็บเซิร์ฟเวอร์นั้นให้กับไคลเอ็นต์ต่อไป



รูปที่ 7.1 แสดงการทำงานของโปรแกรมไฟร์เบรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 8

# การพัฒนาโปรแกรมล็อกมอนิเตอร์

### 8.1 แนวคิด

โปรแกรมล็อกมอนิเตอร์นั้นจะถูกติดตั้งอยู่ในเครื่องแม่ข่ายที่เป็นโคเมนคอนโทรลเลอร์ และเนื่องจากการที่มีไฟร์วอลล์ในส่วนต่างๆของระบบเครือข่ายนั้นจึงได้มีการแบ่งแยกล็อกมอนิเตอร์สำหรับแต่ละโปรแกรม คือ ล็อกมอนิเตอร์ของไฟร์วอลล์ ล็อกมอนิเตอร์ของไฟร์สรีน ที่สามารถเก็บล็อกลงในแอ็คทีฟไดเรกทอรีของเครื่องแม่ข่ายได้ จึงพัฒนาโปรแกรมล็อกมอนิเตอร์ด้วยภาษา C ที่มีการใช้คลาส ADCONNECT ในการพัฒนาโปรแกรม และโปรแกรมล็อกมอนิเตอร์ของไฟร์เบรก ที่ต้องเก็บล็อกในดาต้าเบส MySQL เนื่องจากล็อกของ Snort\_inline นั้นไม่สามารถเก็บลงในแอ็คทีฟไดเรกทอรีได้ จึงพัฒนาโปรแกรมล็อกมอนิเตอร์ด้วยภาษา C ที่มีการใช้คลาส MySqlConnection ในการติดต่อกับ MySQL

### 8.2 ขอบเขตและความสามารถ

ล็อกมอนิเตอร์ของไฟร์วอลล์และไฟร์สรีนนั้นสามารถแสดงผลล็อกไฟล์ที่ถูกส่งกลับมายังเครื่องแม่ข่ายได้ และสามารถลบล็อกไฟล์ที่ต้องการได้ ส่วนล็อกมอนิเตอร์ของไฟร์เบรคนั้นสามารถแสดงผลล็อกไฟล์ที่ถูกส่งกลับมายังเครื่องแม่ข่ายได้เท่านั้น ในการลบล็อกไฟล์ที่ต้องการนั้นจะต้องทำการลบผ่าน โปรแกรมที่ใช้ในการติดต่อกับ MySQL โดยผู้ที่มีสิทธิในการใช้งานดาต้าเบสนี้เท่านั้น

### 8.3 การพัฒนาโปรแกรม

#### 8.3.1 ล็อกมอนิเตอร์ของไฟร์วอลล์

โปรแกรมนี้จะทำการพัฒนาโดยภาษา C โดยการใช้คลาส ADCONNECT ที่ใช้ในการติดต่อกับแอ็คทีฟไดเรกทอรี เพื่อเข้าถึงค่าของล็อกไฟล์ที่ถูกส่งมาเก็บยังแอ็คทีฟไดเรกทอรี จากนั้นจะนำค่าที่ได้นี้มาลงไฟล์ดาต้าเบสที่สร้างจาก Microsoft Access ที่มีชื่อว่า LogfireAlarm โดยมีโครงสร้างตารางที่ใช้ในการเก็บล็อกของเครื่องลูกข่ายดังตารางที่ 8.1 และแสดงผลผ่านทางโปรแกรม

ตารางที่ 8.1 แสดงโครงสร้างตารางที่ใช้ในการเก็บล็อกของเครื่องลูกข่าย

ชื่อเขตข้อมูล	ชนิดข้อมูล	คำอธิบาย
LogNo (คีย์หลัก)	NUMBER	หมายเลขลำดับที่ของล็อก
AttackDst	TEXT	หมายเลขไอพีปลายทาง
AttackSrc	TEXT	หมายเลขไอพีต้นทาง
AttackType	TEXT	ชนิดของการโจมตี
AttackDate	TEXT	วันที่
AttackTime	TEXT	เวลา
User	TEXT	ชื่อผู้ใช้
Group	TEXT	กลุ่มผู้ใช้

### 8.3.2 ล็อกมอโนิเตอร์ของไฟร์สกรีน

โปรแกรมนี้จะทำการพัฒนาโดยภาษา C โดยการใช้คลาส ADCONNECT ที่ใช้ในการติดต่อกับแอ็คทีฟไดเรกทอรี เพื่อเข้าถึงค่าของล็อกไฟล์ที่ถูกส่งมาเก็บยังแอ็คทีฟไดเรกทอรี จากนั้นจะนำค่าที่ได้นี้มาลงไฟล์ดาต้าเบสที่สร้างจาก Microsoft Access ที่มีชื่อว่า LogfireScreen โดยมี

โครงสร้างตารางที่ใช้ในการเก็บล็อกของเครื่องลูกข่ายดังตารางที่ 8.2 และแสดงผลผ่าน โปรแกรม

ชื่อเขตข้อมูล	ชนิดข้อมูล	คำอธิบาย
LogNo (คีย์หลัก)	NUMBER	หมายเลขลำดับที่ของล็อก
Date	TEXT	วันที่
Time	TEXT	เวลา
In	TEXT	อินเตอร์เฟซขาเข้า
Out	TEXT	อินเตอร์เฟซขาออก
Mac	TEXT	หมายเลขแม็คแอดเดรส
Source	TEXT	หมายเลขไอพีต้นทาง
Destination	TEXT	หมายเลขไอพีปลายทาง
Protocol	TEXT	โปรโตคอล
Source Port	TEXT	หมายเลขพอร์ตต้นทาง
Destination Port	TEXT	หมายเลขพอร์ตปลายทาง
Type	TEXT	รูปแบบของไอซีเอ็มพี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 8.3.3 ล็อกมอนิเตอร์ของไฟร์เบรก

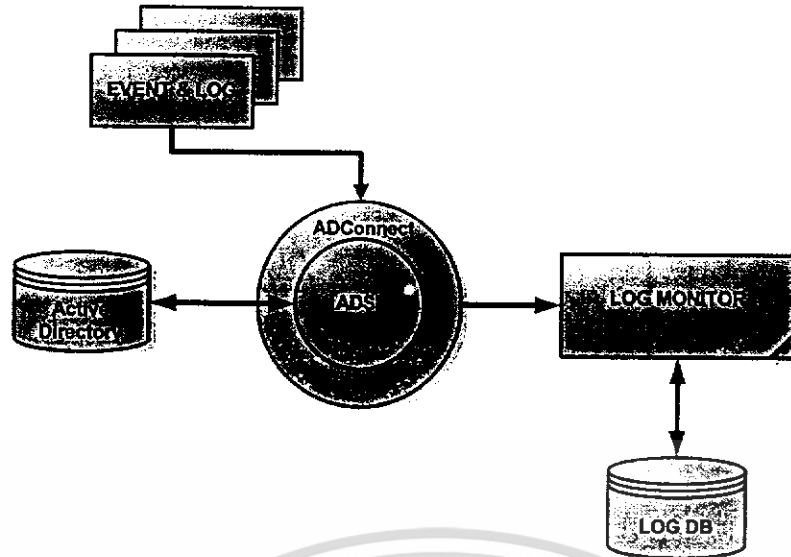
โปรแกรมนี้จะทำการพัฒนาโดยภาษา C โดยการใช้คลาส MySQLConnect ที่สร้างขึ้นเพื่อใช้ในการติดต่อกับ MySQL เพื่อเข้าถึงค่าล็อกไฟล์ที่ถูกส่งมาเก็บยังดาต้าเบสที่ชื่อ Snort ของ MySQL จากนั้นจะนำค่าที่ได้แสดงผลผ่านโปรแกรม โดยมีโครงสร้างของดาต้าเบส Snort และเทเบิลต่างๆ จากไฟล์ Create\_mysql ของโปรแกรม Snort\_inline โดยมีเทเบิลที่สำคัญๆดังตารางที่ 8.3

ตารางที่ 8.3 แสดงเทเบิลที่สำคัญของดาต้าเบส Snort

ชื่อเทเบิลในดาต้าเบส Snort	ชื่อฟิลด์ต่างๆในเทเบิล
detail	detail_type, detail_text
encoding	encoding_type, encoding_text
event	sid, cid, signature, timestamp
iphdr	sid, cid, ip_scr, ip_dst, ip_ver, ip_hlen, ip_tos, ip_len, ip_id, ip_flags, ip_off, ip_ttl, ip_proto, ip_csum
schema	vseq, ctime
sensor	sid, hostname, interface, filter, detail, encode, last_cid
signature	sig_id, sig_name, sig_class_id, sig_priority, sig_rev, sig_sid
tcphdr	sid, cid, tcp_sport, tcp_dport, tcp_seq, tcp_ack, tcp_off, tcp_res, tcp_flags, tcp_win, tcp_csum, tcp_urp

### 8.4 การทำงานของโปรแกรม

โปรแกรมล็อกมอนิเตอร์ของไฟร์วอลล์และล็อกมอนิเตอร์ของไฟร์สกรีน จะมีการทำงานอยู่บนความสามารถของบริการแอ็คทีฟไดเร็กทอรี(Active directory) ของ windows 2003 โดยข้อมูลจะเก็บอยู่ในฐานข้อมูลของแอ็คทีฟไดเร็กทอรี (Active Directory Database) และเข้าถึงข้อมูลผ่าน ADSI ไปยังที่ไฟร์วอลล์เซอร์วิส (Active Directory Services) เพื่อเข้าถึงข้อมูลที่จัดเก็บ ซึ่งการทำงานของล็อกมอนิเตอร์ที่ทำงานร่วมกันกับ แอ็คทีฟไดเร็กทอรี (Active Directory) แสดงโครงสร้างการทำงาน ดังรูปที่ 8.1



รูปที่ 8.1 แสดงโครงสร้างการทำงานระหว่างล็อกมอนิเตอร์กับแอ็กทีฟไดเรคทอรี

ส่วนโปรแกรมล็อกมอนิเตอร์ของไฟร์เบรคนั้นจะมีความทำงาน โดยการติดต่อกับดาต้าเบส MySQL โดยตรงด้วยการส่งคำสั่ง SQL ผ่านทางโปรแกรมเพื่อทำการ Select ROW ที่ต้องการมาแสดงผลคือ Row ที่อยู่ในตาราง Event, Signature, Sensor โดยคำสั่ง SQL ที่ส่งเข้าไปนั้นจะเป็นการ Select ดังต่อไปนี้

1. ค่า NO. คือค่าของ Cid ในตาราง Event
2. ค่า Source IP คือค่าของ ip\_src ในตาราง iphdr
3. ค่า Acctack Type คือค่าของ Sig\_name ในตาราง Signature
4. ค่า Acctack Time คือค่าของ TimeStamp ในตาราง Event

## บทที่ 9

### การทดสอบการทำงาน

#### 9.1 ระบบที่ใช้ในการทดสอบ

การทดสอบการทำงานของระบบชุดโปรแกรมเน็ตเวิร์กไฟร์วอลล์ได้ทำการทดสอบภายใต้โปรแกรมจำลองเครื่องคอมพิวเตอร์มีรายละเอียดดังตารางที่ 9.1

ตารางที่ 9.1 รายละเอียดของเครื่องคอมพิวเตอร์ต่างๆภายในระบบ

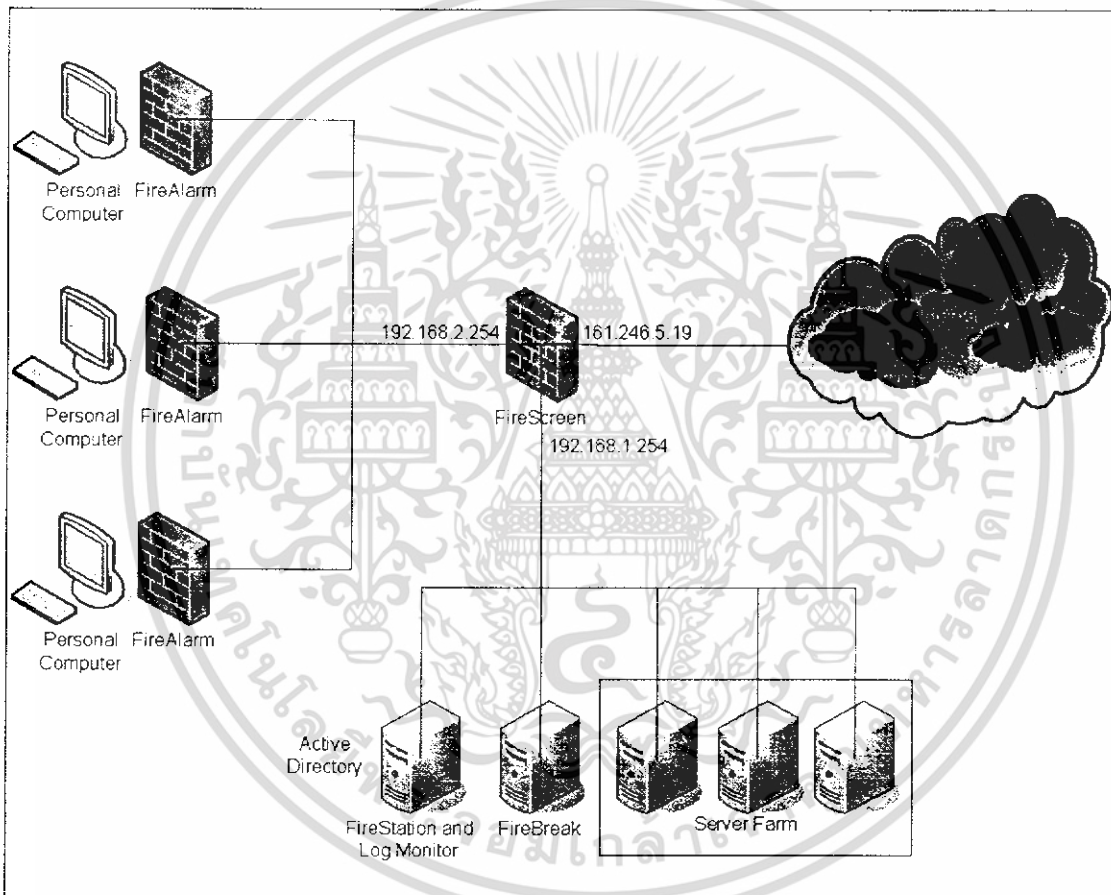
	ไฟร์สเตชัน (FireStation)	ไฟร์อลาร์ม (FireAlarm)	ไฟร์สกรีน (FireScreen)	ไฟร์เบรก (FireBreak)	อินทรูเดอร์ (Intruder)
ระบบปฏิบัติการ	Windows Server 2003 Enterprise Edition	Windows XP Professional SP2	Debian GNU/Linux	Debian GNU/Linux	Debian GNU/Linux
RAM	256 MB	256 MB	128 MB	128 MB	256 MB
IP Address	192.168.1.100 /24	192.168.2.2 /24	192.168.1.254 /24 และ 192.168.2.254/ 24	192.168.1.1 /24	192.168.2.10 /24
Default Gateway	192.168.1.254	192.168.2.254	-	192.168.1.254	192.168.2.254
DNS	192.168.1.254	192.168.2.254	-	192.168.1.254	192.168.2.254
หน้าที่	เป็นศูนย์กลาง การกำหนดคกฎ	เป็นเครื่อง ลูกข่ายภายใน ซึ่งติดตั้ง ไฟร์อลาร์มไว้	เป็นเกตเวย์ ซึ่งติดตั้ง ไฟร์สกรีนไว้	เป็นรีเวิร์ส เว็บพร็อกซี	เป็นเครื่อง ผู้บุกรุก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่เครื่องไฟร์สแตชันนั้นจะมีการติดตั้งแอคทีฟไดเรกทอรีเซอร์วิส(Active Directory) ลงไปและเพิ่มแอตทริบิว (Attribute) เพื่อความสามารถในการใช้เป็นศูนย์กลางการควบคุมไฟร์วอลล์ส่วนต่างๆ ส่วนที่เครื่องไฟร์ลาร์มจะมีการลงWinPCap เพื่อใช้ในการดักจับแพ็คเก็ตของโปรแกรมไฟร์ลาร์มในส่วนของการตรวจจับผู้บุกรุก

## 9.2 โครงสร้างของระบบที่ใช้ในการทดสอบ

โครงสร้างของระบบที่สามารถทำงานได้เต็มประสิทธิภาพนั้น ควรติดตั้งให้ครบทุกส่วน คือ ส่วนไฟร์สแตชัน ไฟร์ลาร์ม ไฟร์เบรก และไฟร์สกรีน ดังรูปที่ 9.1



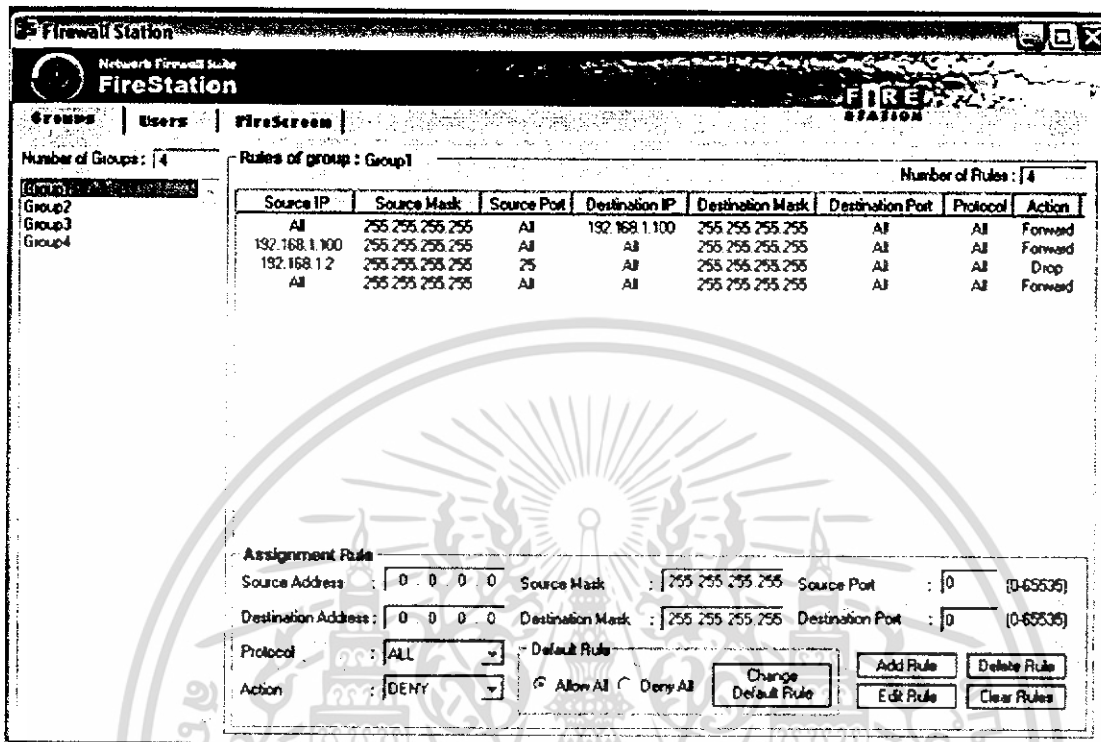
รูปที่ 9.1 โครงสร้างทางเครือข่ายของระบบที่ใช้ในการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 9.3 การทดสอบการทำงาน

#### 9.3.1 ทดสอบการทำงานของไฟร์สแตชัน(FireStation)

เมื่อ โปรแกรมไฟร์สแตชันเริ่มดำเนินการทำงานแล้วจะมีลักษณะดังต่อไปนี้



รูปที่ 9.2 โปรแกรมไฟร์สแตชัน

โปรแกรมไฟร์สแตชันจะแบ่งออกเป็น 3 ส่วนด้วยกันคือ

- ส่วนการกำหนดกฎให้กับกลุ่มผู้ใช้
- ส่วนการกำหนดกฎให้กับผู้ใช้เป็นรายบุคคล
- ส่วนการกำหนดกฎให้กับไฟร์วอลล์บนเกตเวย์หรือไฟร์สกรีน(FireScreen)

การทำงานของโปรแกรมนี้จะเป็นการกำหนดกฎให้กับไฟร์วอลล์ส่วนต่างๆในระบบ สามารถทำการเพิ่ม แก้ไข และลบ กฎได้โดยการนำค่าที่กำหนดไปใส่ในแอดดรีสของอ็อบเจ็กต์นั้นๆ โดยกฎพื้นฐานจะเป็น Deny All คือไม่อนุญาตให้การติดต่ออื่นๆผ่านเข้ามาได้

ตัวอย่างการกำหนดกฎให้กับกลุ่มและผู้ใช้รายบุคคล ถ้าต้องการบล็อกทุกอย่างทุกไอพีแต่ให้สามารถใช้ ICMP ได้ของเครื่อง 192.168.2.2 ให้ใส่กฎเพิ่มกฎดังตารางที่ 9.2 นี้

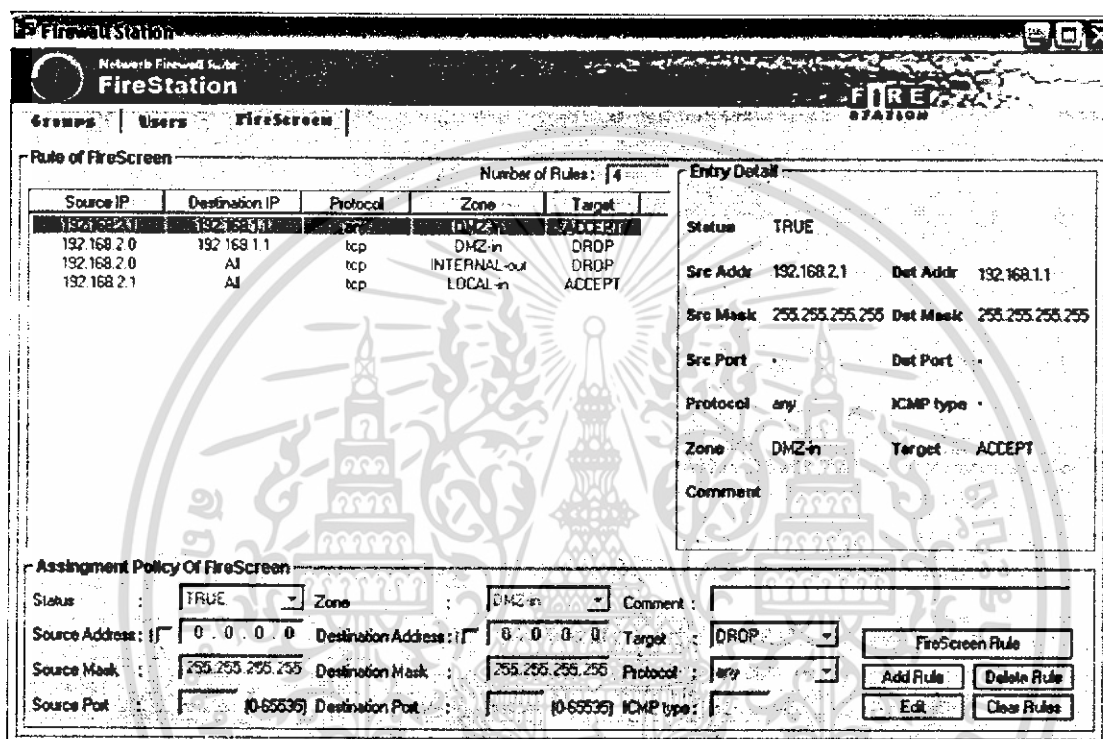
ตารางที่ 9.2 แสดงการกำหนดกฎตัวอย่าง

Src.IP	Src.Mask	Src.Port	Dst. IP	Dst.Mask	Dst.Port	Protocol	Action
192.168.2	255.255.255.255	0	0.0.0.0	255.255.255.255	0	ICMP	ALLOW

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0.0.0.0	255.255.255. 255	0	192.168.2 .2	255.255.255. 255	0	ICMP	ALLO W
---------	---------------------	---	-----------------	---------------------	---	------	-----------

กฎแรกเป็นกฎที่อนุญาตให้ส่ง ICMP ออกไปได้ยังทุกเครื่อง กฎที่สองให้อนุญาตให้ส่ง ICMP เข้ามาได้ การกำหนดกฎให้กับกลุ่มและผู้ใช้รายบุคคลนั้นจะมีการกำหนดที่เหมือนกัน แต่กฎของผู้ใช้รายบุคคลจะมีลำดับความสำคัญที่สูงกว่า ดังนั้นเมื่อมีแพ็คเก็ตเข้ามาที่เครื่องของผู้ใช้คนนั้นแพ็คเก็ตก็จะถูกนำไปเทียบกับกฎที่เป็นของผู้ใช้รายบุคคลก่อนที่จะไปเปรียบเทียบกับกฎที่กำหนดไว้ให้กับกลุ่ม



รูปที่ 9.3 โปรแกรมไฟร์สแตชันส่วนของการกำหนดกฎให้กับไฟร์สกรีน(FireScreen)

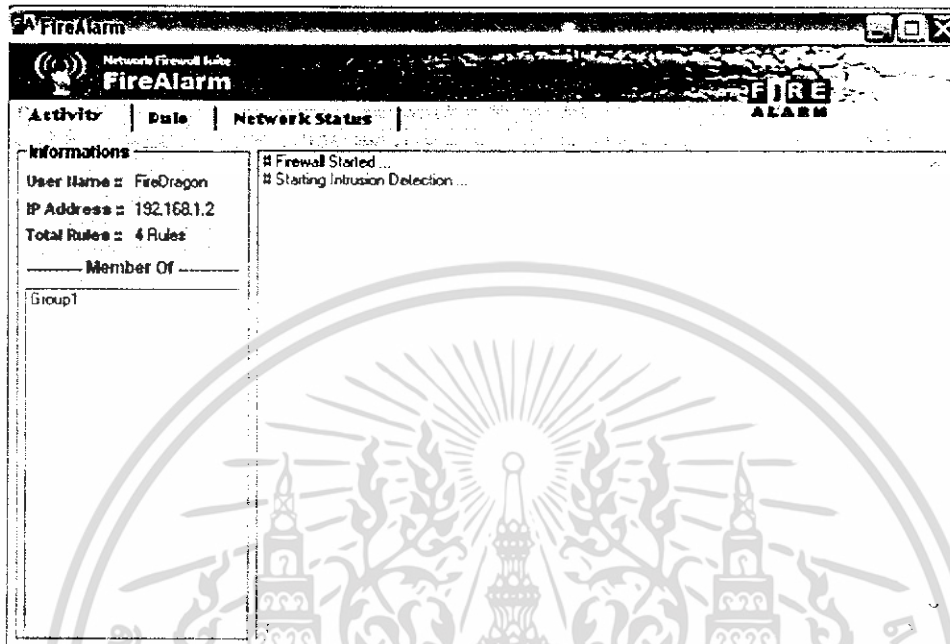
กฎที่กำหนดให้กับแต่ละส่วนจะถูกเก็บไว้ในแอคทีฟไดเรกทอรีในตำแหน่งที่ถูกต้องเพื่อรอโคลเอนต์ที่เป็นเจ้าของกฎนั้นมานำไปใช้ต่อไป ในส่วนของไฟร์สกรีนนั้นกฎที่ใส่ไว้ทั้งหมดนี้ จะถูกนำไปเก็บในแอคทีฟไดเรกทอรีในแอดทริบิวต์ firewallLog ของ DN คือ "cn=firescreen, cn=computers, dc=hephaestus, dc=com" และหลังจากกฎถูกเปลี่ยนแปลงแก้ไขอย่างไรก็ตามจะไปเปลี่ยนค่าในแอดทริบิวต์ update ของ DN เดียวกันนี้เป็น "YES" ด้วย

เช่นเดียวกันกฎสำหรับไฟร์สกรีนจะถูกตีความหมายจากบนลงล่างเช่นกัน สำหรับตัวอย่างข้างต้น 2 กฎแรกสำหรับแพ็คเก็ตที่จะผ่านเข้าไปยังโซน DMZ กฎที่ 3 สำหรับแพ็คเก็ตที่ผ่านออกมาจากเครือข่ายภายใน และสุดท้ายจะเป็นสำหรับแพ็คเก็ตที่เข้ามายังเครื่องเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 9.3.3 ทดสอบการทำงานของไฟร์วอลล์(FireAlarm)

เมื่อ โปรแกรมเทอร์มินอลไฟร์วอลล์เริ่มทำงานแล้ว จะมีลักษณะดังต่อไปนี้



รูปที่ 9.4 โปรแกรมไฟร์วอลล์

เมื่อโปรแกรมเริ่มทำงานจะทำการรับเอากฎการฟิเตอร์ของกลุ่มและผู้ใช้ที่ล็อกอิน(Login) ตามที่กำหนดไว้ที่ส่วนกลางมาเป็นกฎการฟิเตอร์ของตัวโปรแกรมไฟร์วอลล์ก่อนจะเริ่มทำงาน ด้วยการฟิเตอร์ต่างๆ ตามกฎการฟิเตอร์ที่รับมาจากส่วนกลาง และจะเริ่มการทำงานของส่วนตรวจจับผู้บุกรุกด้วยทันที โดยกฎที่รับมาจะแสดงดังรูป 9.5

Source Address	Source Mask	Source Port	Destination Address	Destination Mask	Destination Port	Protocol	Action
All	255.255.255.255	All	192.168.1.100	255.255.255.255	All	All	Allow
192.168.1.100	255.255.255.255	All	All	255.255.255.255	All	All	Allow
192.168.1.2	255.255.255.255	25	All	255.255.255.255	All	All	Deny
All	255.255.255.255	All	All	255.255.255.255	All	All	Allow

รูปที่ 9.5 แสดงกฎการฟิเตอร์ที่รับมาจากเซิร์ฟเวอร์

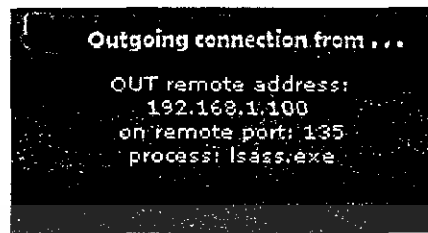
โปรแกรมไฟร์วอลล์ในส่วนสถานะทางเครือข่าย (Network Status) จะแสดงรายการของโปรแกรมที่ใช้งานเครือข่ายอยู่ในขณะนั้น โดยจะอัปเดตรายการตามเวลาที่ได้ตั้งเอาไว้ และในส่วนนี้สามารถเลือกจบโปรแกรมการทำงาน (Kill Process) หรือปิดการเชื่อมต่อของโปรเซส (Close Connection) ได้ดังแสดงในรูป 7-4

Protocol	PID	Process Name	Local Address	Local Port	Remote Address	Remote Port	State
<input type="checkbox"/> TCP	1036	svchost.exe	0.0.0.0	135	0.0.0.0	-	LISTENING
<input type="checkbox"/> TCP	4	System	0.0.0.0	445	0.0.0.0	-	LISTENING
<input type="checkbox"/> TCP	1956	alg.exe	127.0.0.1	1028	0.0.0.0	-	LISTENING
<input type="checkbox"/> TCP	4	System	192.168.1.2	139	0.0.0.0	-	LISTENING
<input type="checkbox"/> TCP	4	System	192.168.1.2	1411	192.168.1.100	445	ESTABLISH...
<input type="checkbox"/> UDP	4	System	0.0.0.0	445	-	-	-
<input type="checkbox"/> UDP	748	lsass.exe	0.0.0.0	500	-	-	-
<input type="checkbox"/> UDP	1212	svchost.exe	0.0.0.0	1025	-	-	-
<input type="checkbox"/> UDP	1212	svchost.exe	0.0.0.0	1026	-	-	-
<input type="checkbox"/> UDP	602	svchost.exe	0.0.0.0	1027	-	-	-
<input type="checkbox"/> UC	-	Kill Process	0.0.0.0	4500	-	-	-
<input type="checkbox"/> UC	-	Close Connection	127.0.0.1	123	-	-	-
<input type="checkbox"/> UDP	748	lsass.exe	127.0.0.1	1035	-	-	-
<input type="checkbox"/> UDP	684	winkgon.exe	127.0.0.1	1236	-	-	-
<input checked="" type="checkbox"/> UDP	880	FireAlarm.exe	127.0.0.1	1294	-	-	-
<input type="checkbox"/> UDP	1240	svchost.exe	127.0.0.1	1900	-	-	-

รูปที่ 9.6 แสดงสถานะของโปรเซสที่ใช้งานเครือข่าย

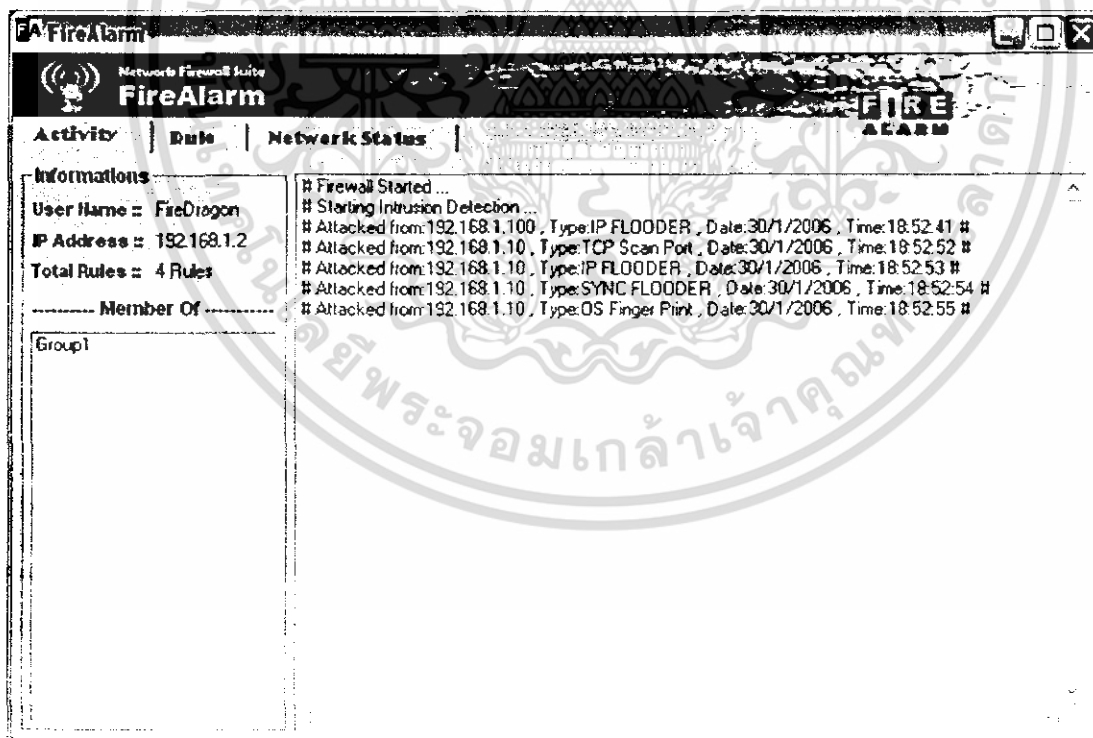
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อมีโปรเซสใดที่เชื่อมต่อเครือข่ายจะป๊อปอัพ (Popup) ขึ้นมาบอกทิศทางและไอพีแอดเดรสที่โปรเซสนั้นเชื่อมต่อ ดังแสดงในรูปที่ 9.7



รูปที่ 9.7 แสดงป๊อปอัพ(Popup) แสดงโปรเซสที่ใช้งานเครือข่าย

และเมื่อมีการโจมตีเกิดหรือมีการบุกรุกทางเครือข่ายเกิดขึ้น โปรแกรมก็จะแสดงรายละเอียดดังรูปที่ 9.8 ทำการส่งรายละเอียดต่างๆ ของการโจมตีหรือการบุกรุกนั้นๆ ไปเก็บไว้ที่แอดทริบิวต์ firewallLog ของกลุ่มของผู้ใช้ในแอ็กทีฟไดเรกทอรี (Active Directory)



รูปที่ 9.8 การแจ้งเตือนการบุกรุกหรือถูกโจมตี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 9.3.3 ทดสอบการทำงานของไฟร์สกรีน(FireScreen)

หลังจากเครื่องเกตเวย์ซึ่งติดตั้งโปรแกรมไฟร์สกรีนไว้เริ่มต้นทำงานขึ้นมาขั้นแรก โปรแกรมจะทำการใส่กฎเริ่มต้นและกฎเดิมเข้ากับเกตเวย์เสียก่อน โดยในที่นี้จะเป็นการทำงานครั้งแรกของโปรแกรมไฟร์สกรีนจะไม่มีกฎเก่าก่อนอยู่ เหลือเพียงแค่กฎเริ่มต้นที่ตั้งไว้แล้วเท่านั้น

ใช้คำสั่ง iptables -L --line-number เพื่อตรวจสอบกฎในไอพีเทเบิลส์ว่าถูกต้องตามต้องการหรือไม่

```

firescreen:~# iptables -L --line-number
Chain INPUT (policy ACCEPT)
num target prot opt source destination
1 ACCEPT all -- anywhere anywhere
2 ACCEPT tcp -- anywhere anywhere tcp flags:SYN,RST,ACK/SYN
limit: avg 5/sec burst 5
3 DROP all -- BASE-ADDRESS.MCAST.NET/4 anywhere
4 DROP all -- 240.0.0.0/5 anywhere
5 DROP all -- 127.0.0.0/8 anywhere
6 DROP all -- 0.0.0.0/8 anywhere
7 DROP all -- 169.254.0.0/16 anywhere
8 DROP all -- anywhere anywhere state INVALID
9 ACCEPT all -- firestation.hephaestus.com firescreen.hephaestus.com state RELATED,ESTABLISHED
Chain FORWARD (policy ACCEPT)
num target prot opt source destination
1 ACCEPT tcp -- 192.168.2.0/24 firestation.hephaestus.com tcp dpt:886
2 ACCEPT tcp -- firestation.hephaestus.com 192.168.2.0/24 state RELATED,ESTABLISHED
3 ACCEPT all -- 192.168.1.0/24 anywhere state RELATED,ESTABLISHED
4 ACCEPT tcp -- anywhere anywhere tcp flags:SYN,RST,ACK/SYN
limit: avg 5/sec burst 5
5 DROP all -- BASE-ADDRESS.MCAST.NET/4 anywhere
6 DROP all -- 240.0.0.0/5 anywhere
7 DROP all -- 127.0.0.0/8 anywhere
8 DROP all -- 169.254.0.0/16 anywhere
9 DROP all -- anywhere BASE-ADDRESS.MCAST.NET/4
10 DROP all -- anywhere 240.0.0.0/5
11 DROP all -- anywhere 127.0.0.0/8
12 DROP all -- anywhere 169.254.0.0/16
13 DROP all -- anywhere 255.255.255.255
14 DROP all -- anywhere anywhere state INVALID
Chain OUTPUT (policy ACCEPT)
num target prot opt source destination
1 ACCEPT all -- anywhere anywhere
2 DROP all -- anywhere 255.255.255.255
3 DROP all -- anywhere BASE-ADDRESS.MCAST.NET/4
4 ACCEPT tcp -- firescreen.hephaestus.com firestation.hephaestus.com tcp dpt:ldap
Chain LOG-CHAIN (0 references)
num target prot opt source destination
1 LOG all -- anywhere anywhere limit: avg 1/sec burst 5
LOG level info prefix "FIRESCREEN"
2 DROP all -- anywhere anywhere

```

รูปที่ 9.9 แสดงกฎเริ่มต้นของไฟร์สกรีน

จะพบว่ากฎพื้นฐานที่กำหนดไว้ในเชลล์สคริปต์ `init.sh` ถูกกำหนดให้กับไอพีเทเบิลส์เรียบร้อยแล้ว รวมถึงการสร้างเชน LOG-CHAIN ซึ่งใช้สำหรับเก็บลิสต์การรายละเอียดของแพ็กเก็ตที่จะละทิ้งไป (DROP) อนึ่งสำหรับแพ็กเก็ตที่ละทิ้งไปในกฎพื้นฐานนี้จะไม่มีการเก็บลิสต์ไว้แต่อย่างใด เฉพาะส่วนที่ละทิ้งไปเนื่องจากกฎที่กำหนดไว้ที่ไฟร์สแตชันเท่านั้นที่จะมีการเก็บรายละเอียดลงลิสต์ไฟล์

ขั้นตอนต่อไปจะเป็นการนำกฎจากไฟร์สแตชันที่ได้กำหนดไว้ก่อนหน้ามาใช้ ไฟร์สกรีนจะคอยตรวจสอบไปยังไฟร์สแตชันเป็นระยะเพื่อดูว่ามีการเปลี่ยนแปลงแก้ไขหรือไม่ กรณีนี้เพิ่งได้ทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเพิ่มกฎเข้าไปใหม่ ดังนั้นไฟร์สกรีนจะทำการนำกฎทั้งหมดมาจากไฟร์สเดชั่นต่อท้ายกฎพื้นฐานก่อนหน้า

ตรวจสอบดูด้วยคำสั่ง iptables -L --line-number เพื่อดูว่าไอพีเทเบิลส์ได้รับกฎตามที่กำหนดไว้โดยไฟร์สเดชั่นถูกต้องหรือไม่

```
Chain FORWARD (policy ACCEPT)
num target prot opt source destination
15 ACCEPT tcp -- 192.168.2.0/24 www.hephaestus.com tcp dpt:ssh
16 ACCEPT tcp -- firestation.hephaestus.com 192.168.2.0/24 state RELATED,ESTAB
LISTED
13 ACCEPT all -- 192.168.1.0/24 anywhere state RELATED,ESTABLISHED
14 ACCEPT tcp -- anywhere anywhere tcp flags SYN,RST,ACK,SYN
limit avg 5/sec burst 5
15 DROP all -- BASE-ADDRESS: MCAST: NET//4 anywhere
16 DROP all -- 240.0.0.0/5 anywhere
17 DROP all -- 127.0.0.0/8 anywhere
18 DROP all -- 0.0.0.0/8 anywhere
19 DROP all -- 169.254.0.0/16 anywhere
20 DROP all -- anywhere anywhere state INVALID
15 ACCEPT all -- hep01.hephaestus.com www.hephaestus.com
16 LOG-CHAIN tcp -- 192.168.2.0/24 www.hephaestus.com tcp dpt:ssh
17 LOG-CHAIN tcp -- 192.168.2.0/24 anywhere tcp dpt:1863
```

รูปที่ 9.10 กฎจากไฟร์สเดชั่นที่เพิ่มเข้าไปในเซิร์ฟเวอร์

```
Chain INPUT (policy ACCEPT)
num target prot opt source destination
1 ACCEPT all -- anywhere anywhere
2 ACCEPT tcp -- anywhere anywhere tcp flags SYN,RST,ACK,SYN
limit avg 5/sec burst 5
3 DROP all -- BASE-ADDRESS: MCAST: NET//4 anywhere
4 DROP all -- 240.0.0.0/5 anywhere
5 DROP all -- 127.0.0.0/8 anywhere
6 DROP all -- 0.0.0.0/8 anywhere
7 DROP all -- 169.254.0.0/16 anywhere
8 DROP all -- anywhere anywhere state INVALID
9 ACCEPT all -- firestation.hephaestus.com firestation.hephaestus.com state RELATED,ESTABLISHED
10 ACCEPT tcp -- hep01.hephaestus.com anywhere tcp dpt:ssh
```

รูปที่ 9.11 กฎจากไฟร์สเดชั่นที่เพิ่มเข้าไปในเซิร์ฟเวอร์

จะสังเกตเห็นว่าในเซิร์ฟเวอร์จะพบว่ามีกฎเพิ่มมาอีก 3 กฎเพิ่มขึ้นมาจากกฎพื้นฐานจาก init.sh คือกฎที่ 15, 16 และ 17 ซึ่งเป็น 3 กฎแรกที่กำหนดไว้ในไฟร์สเดชั่นที่ได้แสดงให้ดูในหัว 9.3.1 นั้นเอง (รูปที่ 9.3)

ส่วนในเซิร์ฟเวอร์ก็มีเพิ่มขึ้น 1 กฎเช่นกัน ก็คือกฎสุดท้ายที่ตั้งไว้ในไฟร์สเดชั่น (รูปที่ 9.3) อย่างไรก็ตามในส่วนของเซิร์ฟเวอร์และเซิร์ฟเวอร์ LOG-CHAIN นั้นจะไม่ถูกทำการเปลี่ยนแปลงแก้ไขแต่อย่างใด

สำหรับส่วนของล็อกไฟล์ หากมีแพ็คเกจที่ตรงกับกฎที่ระบุทาร์เก็ตเป็นเซิร์ฟเวอร์ LOG-CHAIN (ก่อนที่จะถูกละทิ้งไปที่ตอนท้ายของเซิร์ฟเวอร์ LOG-CHAIN) จะถูกเก็บรายละเอียดลงยังล็อกไฟล์ซึ่งเก็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไว้ที่ /var/log/iptables.log และจะมีรายละเอียดของล็อกแสดงขึ้นมาบนหน้าจอด้วย

รายละเอียดของล็อกที่แสดงขึ้นมาเป็นดังรูปที่ 9.12

```

FIRESCREEN IN=eth2 OUT=eth1 SRC=192.168.2.7 DST=192.168.1.1 LEN=64 TOS=0x00 PREC
0x00 TTL=63 ID=30207 DF PROTO=TCP SPT=2524 DPT=22 WINDOW=1460 RES=0x00 ACK URGI
0
FIRESCREEN IN=eth2 OUT=eth1 SRC=192.168.2.7 DST=192.168.1.1 LEN=64 TOS=0x00 PREC
0x00 TTL=63 ID=30209 DF PROTO=TCP SPT=2524 DPT=22 WINDOW=1460 RES=0x00 ACK URGI
0
FIRESCREEN IN=eth2 OUT=eth1 SRC=192.168.2.7 DST=192.168.1.1 LEN=64 TOS=0x00 PREC
0x00 TTL=63 ID=30291 DF PROTO=TCP SPT=2524 DPT=22 WINDOW=1460 RES=0x00 ACK URGI
0
FIRESCREEN IN=eth2 OUT=eth1 SRC=192.168.2.7 DST=192.168.1.1 LEN=64 TOS=0x00 PREC
0x00 TTL=63 ID=30293 DF PROTO=TCP SPT=2524 DPT=22 WINDOW=1460 RES=0x00 ACK URGI
0

```

รูปที่ 9.12 ล็อกที่แสดงขึ้นมาที่หน้าจอของไฟร์สกรีน

ล็อกดังกล่าวเป็นลอกจากเครื่องภายในเครือข่ายในซึ่งมีไอพีแอดเดรสเป็น 192.168.2.7 พยายามใช้ซีเคียวเชลล์ (ssh) เข้าไปยังเครื่องเว็บเซิร์ฟเวอร์ซึ่งมีไอพีแอดเดรสเป็น 192.168.1.1 แต่ถูกปฏิเสธโดยตัวไฟร์สกรีนจากกฎที่ 16 ในเซิร์ฟเวอร์ (รูปที่ x)

ล็อกไฟล์จริงๆนั้นเก็บไว้ที่ /var/log/iptables.log จะมีรูปแบบเช่นเดียวกันกับล็อกที่แสดงบนหน้าจอ ดังรูปที่ x เพียงแต่เพิ่มวันเวลาและชื่อโฮสต์ด้านหน้า ไฟร์สกรีนจะอ่านไฟล์นี้เป็นระยะเพื่อส่งล็อกในส่วนของไอพีเทเบิลส์ไปเก็บยังแอกทิฟไดเรกทอรีบนไฟร์สแตชัน (ล็อกใน iptables.log ไม่ได้มีเฉพาะลอกจากไอพีเทเบิลส์แต่ประกอบด้วยล็อกอื่นๆในระดับเดียวกันด้วยเช่นกัน)

ข้อมูลภายใน iptables.log เป็นดังแสดงในรูปที่ 9.13

```

Jan 11 02:43:33 iptables kernel: FIRESCREEN IN=eth2 OUT=eth1 SRC=192.168.2.7
DST=192.168.1.1 LEN=64 TOS=0x00 PREC=0x00 TTL=63 ID=30289 DF PROTO=TCP SPT=2524
DPT=22 WINDOW=1460 RES=0x00 ACK URGP=0
Jan 11 02:43:57 iptables kernel: FIRESCREEN IN=eth2 OUT=eth1 SRC=192.168.2.7
DST=192.168.1.1 LEN=64 TOS=0x00 PREC=0x00 TTL=63 ID=30291 DF PROTO=TCP SPT=2524
DPT=22 WINDOW=1460 RES=0x00 ACK URGP=0
Jan 11 02:44:45 iptables kernel: FIRESCREEN IN=eth2 OUT=eth1 SRC=192.168.2.7
DST=192.168.1.1 LEN=64 TOS=0x00 PREC=0x00 TTL=63 ID=30293 DF PROTO=TCP SPT=2524
DPT=22 WINDOW=1460 RES=0x00 ACK URGP=0

```

รูปที่ 9.13 แสดงข้อมูลใน /var/log/iptables.log

ล็อกจะถูกส่งไปเก็บยังแอกทิฟไดเรกทอรีบนไฟร์สแตชัน ในแอกทริบิวต์ชื่อ firewallLog ภายใต้ DN = "cn=iptables, cn=computers, dc=hephaestus, dc=com" ส่วนการนำล็อกมาแสดงบนไฟร์สแตชันจะกล่าวถึงในหัวข้อถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 9.3.4 การทดสอบการทำงานของไฟร์เบรก

โปรแกรมนี้เป็นการทำงานร่วมกันของ Squid และ Snort\_inline ดังนั้นการเริ่มทำงานของโปรแกรมก็คือการเปิดเซอร์วิสของ Squid และ Snort\_inline ให้เริ่มทำงานบนเครื่องที่ทำหน้าที่เป็นซีเคียวริตี้เว็บพริคซ์ ซึ่งการเปิดเซอร์วิสของ squid ให้เริ่มใช้งานด้วยคำสั่ง

```
# /etc/init.d/squid start หรือ # /etc/init.d/squid restart
```

และการสั่งให้ Snort\_inline เริ่มทำงาน โดยต้องมีการตั้งค่าการใช้งานของ iptables ก่อนและต้องทำการโหลดโมดูลของ ip\_queue มาใช้งานดังนี้

```
# iptables -A INPUT -p tcp --dport 80 -j QUEUE
```

```
# modprobe ip_queue
```

```
# lsmod | grep ip_queue
```

หลังจากตั้งค่าการใช้งานของ iptables แล้วให้เริ่มการทำงานของ snort\_inline ในโหมด verbose ด้วยคำสั่งต่อไปนี้

```
# snort_inline -c /etc/snort_inline/snort_inline.conf -Q -v -i eth0
```

จากการเปิดใช้งาน โปรแกรมทั้งสองนี้ทำให้สามารถเชื่อมต่อกับเครื่องแม่ข่ายได้และสามารถใช้งานไฟร์เบรกได้โดยมีความสามารถในการเป็นซีเคียวริตี้เว็บพริคซ์แล้ว ดังรูปที่ 9.14 ที่แสดงถึงการเปิดใช้งานโปรแกรม snort\_inline ในโหมด verbose

```
i gen-id=1 sig-id=2275 type=Threshold tracking=dst count=5 seconds=
60
i gen-id=1 sig-id=2001073 type=Limit tracking=src count=1 seconds=
60
i gen-id=1 sig-id=2001579 type=Both tracking=src count=200 seconds=
60
i gen-id=1 sig-id=2002664 type=Limit tracking=src count=1 seconds=
60
i gen-id=1 sig-id=2496 type=Both tracking=dst count=20 seconds=
60
i gen-id=1 sig-id=2523 type=Both tracking=dst count=10 seconds=
10
i gen-id=1 sig-id=2495 type=Both tracking=dst count=20 seconds=
60
-----[suppression]-----
i none
-----
Rule application order: ->activation->dynamic->drop->sdrop->reject->alert->pass-
->log
Log directory = /var/log/snort

==== Initialization Complete ====

--> Snort_Inline! <*-
o" )" Version 2.3.0 (Build 10)
****
By Martin Roesch & The Snort Team: http://www.snort.org/team.html
Snort_Inline Mod by William Metcalf, Victor Julien, Rob McMillen, Jed Haile
(C) Copyright 1998-2004 Sourcefire Inc., et al.
```

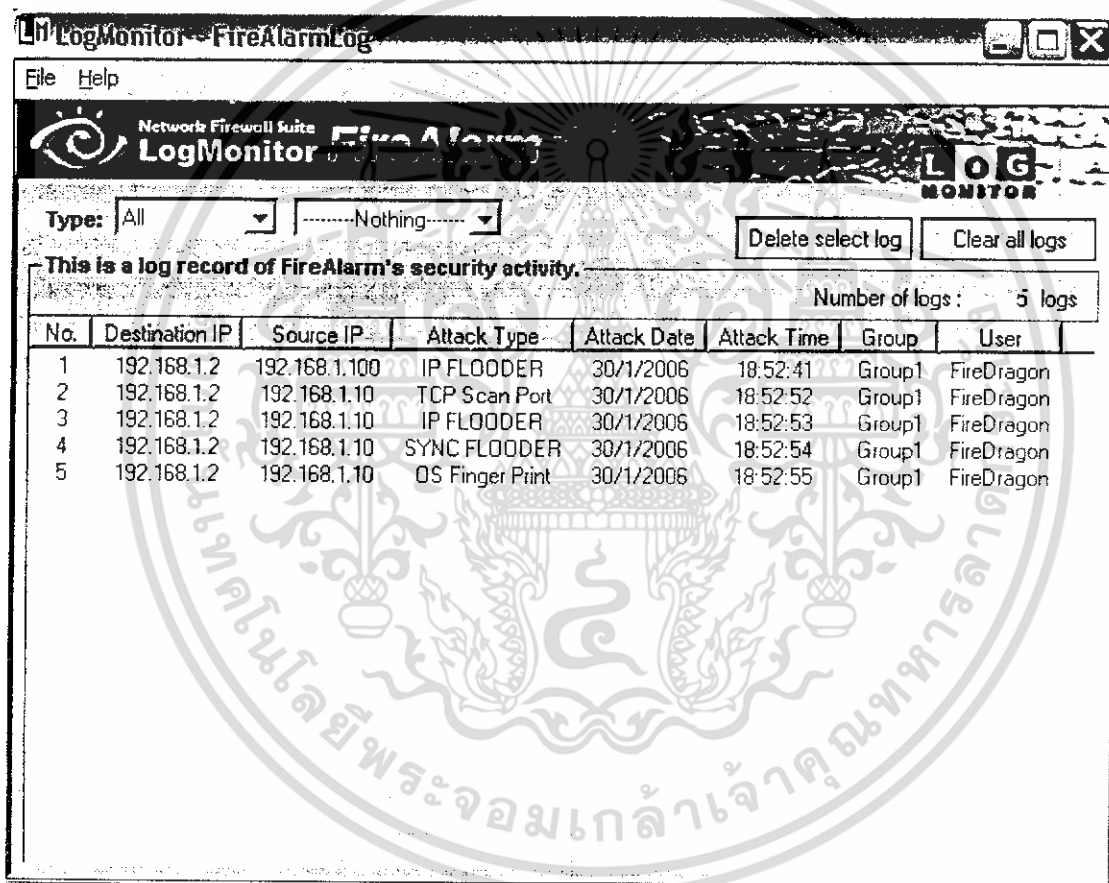
รูปที่ 9.14 แสดงการทำงานของไฟร์เบรกขณะเปิดโปรแกรม snort\_inline

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งเมื่อมีการร้องขอการใช้งานเว็บเซิร์ฟเวอร์ Snort\_inline จะทำการตรวจสอบแพ็คเก็ตที่ผ่านเข้ามาถ้าไม่มีการผิดปกติก็จะทำการติดต่อไปยังเว็บเซิร์ฟเวอร์ที่อยู่ข้างหลังเครื่องรีเวิร์สเว็บพร็อกซี่นี้ เพื่อรับไฟล์มาตอบกลับให้กับไคลเอนต์ที่ทำการร้องขอการใช้งานเข้ามา แต่ถ้าแพ็คเก็ตที่ผ่านเข้ามานั้นมีความผิดปกติแล้ว Snort\_inline สามารถตรวจสอบได้ก็จะมีการส่งล็อกไฟล์กลับมาแจ้งเตือนยังเครื่องแม่ข่ายและนำเก็บลงดาต้าเบส Snort ของ MySQL และนำไปแสดงผลยังโปรแกรมลือกมอนิเตอร์ของไฟร์เบรกต่อไป

### 9.3.5 ทดสอบการทำงานของลือกมอนิเตอร์ไฟร์อลาร์ม (Log Monitor FireAlarm)

เมื่อโปรแกรมลือกมอนิเตอร์ไฟร์อลาร์มเริ่มทำงานแล้ว จะมีลักษณะดังต่อไปนี้



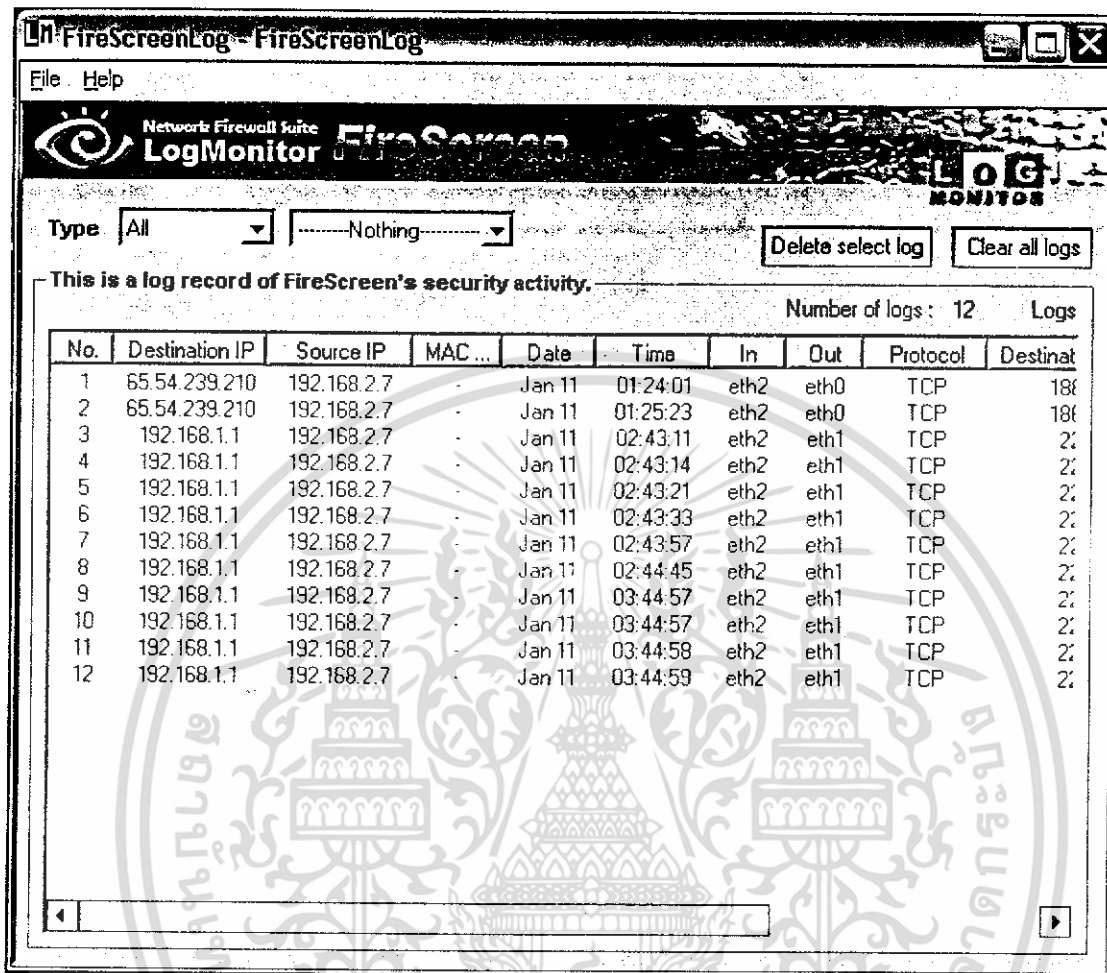
รูปที่ 9.15 โปรแกรมลือกมอนิเตอร์ไฟร์อลาร์ม

จะสามารถเลือกดูบันทึกข้อมูลจากโปรแกรมไฟร์อลาร์มบนเครื่องไคลเอนต์ (Client) ที่ส่งมาทั้งหมดได้ ทั้งแบบเลือกดูแต่ละประเภทได้ สามารถลบได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 9.3.6 การทดสอบโปรแกรมล็อกมอเนอริไฟร์สกรีน (Log Monitor FireScreen)

เมื่อโปรแกรมล็อกมอเนอริไฟร์สกรีนเริ่มทำงานแล้ว จะมีลักษณะดังต่อไปนี้



The screenshot shows the FireScreen Log Monitor window. At the top, there is a menu bar with 'File' and 'Help'. Below the menu bar is the FireScreen logo and the text 'Network Firewall Suite LogMonitor'. There are two dropdown menus for 'Type' (set to 'All') and a filter set to 'Nothing'. There are buttons for 'Delete select log' and 'Clear all logs'. Below this is a section titled 'This is a log record of FireScreen's security activity.' with a 'Number of logs: 12' and a 'Logs' button. The main area contains a table with 12 rows of log entries.

No.	Destination IP	Source IP	MAC ...	Date	Time	In	Out	Protocol	Destinat
1	65.54.239.210	192.168.2.7	-	Jan 11	01:24:01	eth2	eth0	TCP	186
2	65.54.239.210	192.168.2.7	-	Jan 11	01:25:23	eth2	eth0	TCP	186
3	192.168.1.1	192.168.2.7	-	Jan 11	02:43:11	eth2	eth1	TCP	2%
4	192.168.1.1	192.168.2.7	-	Jan 11	02:43:14	eth2	eth1	TCP	2%
5	192.168.1.1	192.168.2.7	-	Jan 11	02:43:21	eth2	eth1	TCP	2%
6	192.168.1.1	192.168.2.7	-	Jan 11	02:43:33	eth2	eth1	TCP	2%
7	192.168.1.1	192.168.2.7	-	Jan 11	02:43:57	eth2	eth1	TCP	2%
8	192.168.1.1	192.168.2.7	-	Jan 11	02:44:45	eth2	eth1	TCP	2%
9	192.168.1.1	192.168.2.7	-	Jan 11	03:44:57	eth2	eth1	TCP	2%
10	192.168.1.1	192.168.2.7	-	Jan 11	03:44:57	eth2	eth1	TCP	2%
11	192.168.1.1	192.168.2.7	-	Jan 11	03:44:58	eth2	eth1	TCP	2%
12	192.168.1.1	192.168.2.7	-	Jan 11	03:44:59	eth2	eth1	TCP	2%

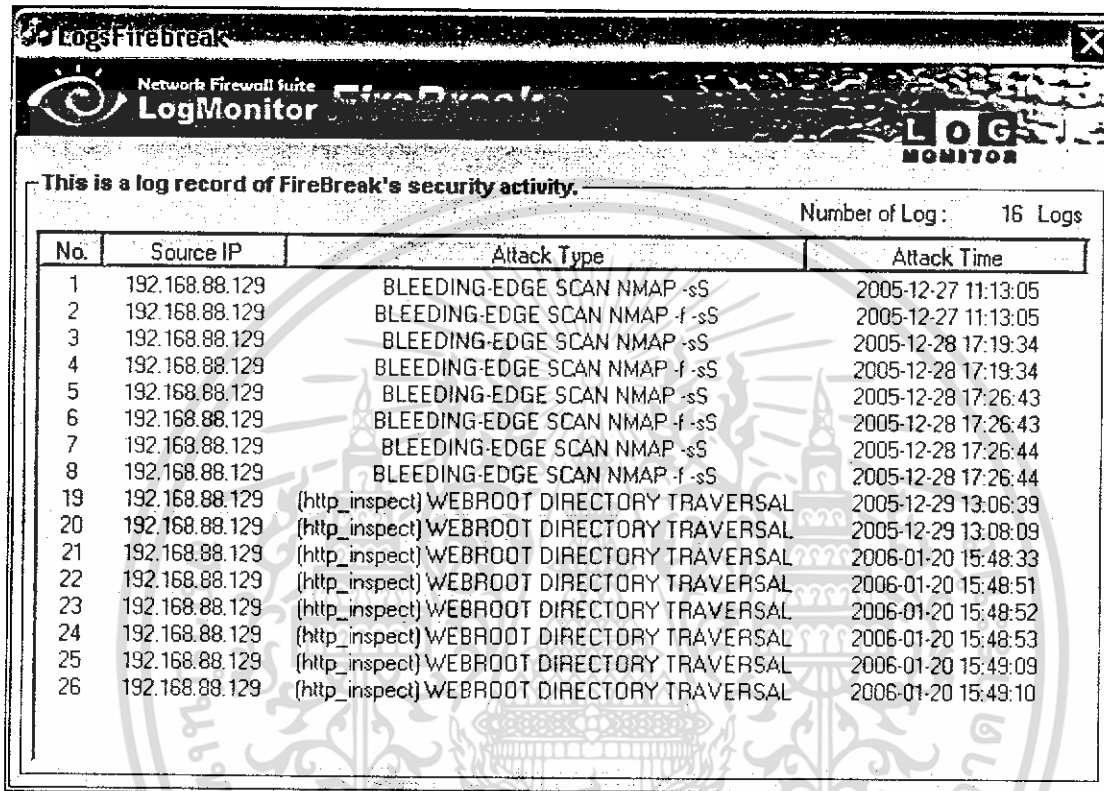
รูปที่ 9.16 โปรแกรมล็อกมอเนอริไฟร์สกรีน

จะสามารถเลือกดูบันทึกข้อมูลจาก โปรแกรมไฟร์สกรีนบนเครื่องเกตเวย์ ที่ส่งมาทั้งหมด ได้ ทั้งแบบเลือกดูแต่ละประเภทได้ สามารถลบได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 9.3.7 การทดสอบโปรแกรมล็อกมอนิเตอร์ไฟร์เบรก (Log Monitor FireBreak)

เมื่อ โปรแกรมล็อกมอนิเตอร์ไฟร์เบรกเริ่มทำงานแล้ว จะมีลักษณะดังต่อไปนี้



This is a log record of FireBreak's security activity. Number of Log : 16 Logs

No.	Source IP	Attack Type	Attack Time
1	192.168.88.129	BLEEDING-EDGE SCAN NMAP -sS	2005-12-27 11:13:05
2	192.168.88.129	BLEEDING-EDGE SCAN NMAP -f-sS	2005-12-27 11:13:05
3	192.168.88.129	BLEEDING-EDGE SCAN NMAP -sS	2005-12-28 17:19:34
4	192.168.88.129	BLEEDING-EDGE SCAN NMAP -f-sS	2005-12-28 17:19:34
5	192.168.88.129	BLEEDING-EDGE SCAN NMAP -sS	2005-12-28 17:26:43
6	192.168.88.129	BLEEDING-EDGE SCAN NMAP -f-sS	2005-12-28 17:26:43
7	192.168.88.129	BLEEDING-EDGE SCAN NMAP -sS	2005-12-28 17:26:44
8	192.168.88.129	BLEEDING-EDGE SCAN NMAP -f-sS	2005-12-28 17:26:44
19	192.168.88.129	(http_inspect) WEBROOT DIRECTORY TRAVERSAL	2005-12-29 13:06:39
20	192.168.88.129	(http_inspect) WEBROOT DIRECTORY TRAVERSAL	2005-12-29 13:08:09
21	192.168.88.129	(http_inspect) WEBROOT DIRECTORY TRAVERSAL	2006-01-20 15:48:33
22	192.168.88.129	(http_inspect) WEBROOT DIRECTORY TRAVERSAL	2006-01-20 15:48:51
23	192.168.88.129	(http_inspect) WEBROOT DIRECTORY TRAVERSAL	2006-01-20 15:48:52
24	192.168.88.129	(http_inspect) WEBROOT DIRECTORY TRAVERSAL	2006-01-20 15:48:53
25	192.168.88.129	(http_inspect) WEBROOT DIRECTORY TRAVERSAL	2006-01-20 15:49:09
26	192.168.88.129	(http_inspect) WEBROOT DIRECTORY TRAVERSAL	2006-01-20 15:49:10

รูปที่ 9.17 โปรแกรมล็อกมอนิเตอร์ไฟร์เบรก

จะสามารถเลือกดูบันทึกข้อมูลจากโปรแกรมไฟร์เบรกบนเครื่องซีเคียวริตี้เว็บพริ๊ออกซ์ที่ส่งมาทั้งหมดได้ แต่ไม่สามารถทำการลบได้ ถ้าต้องการที่จะลบล็อกไฟล์ใด ก็จะต้องทำการผ่านโปรแกรมที่สามารถติดต่อกับ MySQL ค่าด้านลบที่ชื่อ Snort ได้ โดยการสั่งด้วยคำสั่ง SQL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 10

### บทสรุปและวิจารณ์

#### 10.1 บทสรุป

ในปัจจุบันนี้ระบบเครือข่ายคอมพิวเตอร์ มีการใช้งานกันอย่างแพร่หลายและมีจำนวนเพิ่มขึ้นมากทำให้เกิดความยุ่งยากในการจัดการดูแล ดังนั้นจึงต้องหาวิธีการในการควบคุมระบบทั้งหมดจากเพียงจุดเดียวเพื่อทำให้การทำงานมีความสะดวกมากขึ้น

การพัฒนาชุด โปรแกรมไฟร์วอลล์บนเครือข่าย โดยใช้แอ็คทีฟไดเรกทอรี (Active Directory) เป็นศูนย์กลางการควบคุมนั้นช่วยทำให้การทำงานสะดวกขึ้นมาก โดยที่สามารถทำการ เพิ่ม ลด แก้ไข เปลี่ยนแปลง กฎการป้องกันการบุกรุกของไฟร์วอลล์ส่วนต่างๆที่อยู่ในไดเรกทอรีดาต้าเบสได้ ทำให้สามารถป้องกันการโจมตีได้ตามการกำหนดกฎการป้องกันที่มีอยู่ในไดเรกทอรีดาต้าเบส และสามารถนำข้อมูลการโจมตีที่ตรวจจับได้จากเครื่องเกตเวย์และเครื่องลูกข่ายมาเก็บลงฐานข้อมูล โดยล็อกมินิเตอร์ โดยสามารถนำข้อมูลที่ได้นำมาแสดงผลและวิเคราะห์เพื่อกำหนดกฎการป้องกันได้ตามการโจมตีที่เกิดขึ้นได้จริง

#### 10.2 วิจารณ์สิ่งที่ได้จากโครงการ

1. ไฟร์สเตชันทำงานเป็นศูนย์กลางของชุดโปรแกรมเน็ตเวิร์กไฟร์วอลล์ มีความสามารถดังนี้
  - สามารถกำหนดกฎให้กับไฟร์วอลล์ซึ่งเป็นไฟร์วอลล์ส่วนบุคคลพร้อมระดับตรวจจับผู้บุกรุกบนเครื่องไคลเอนต์ในเครือข่ายภายในได้และสามารถรับล็อกกลับเก็บไว้ที่ศูนย์กลาง
  - สามารถกำหนดกฎให้กับไฟร์สกรีนซึ่งทำหน้าที่เป็นเกตเวย์ไฟร์วอลล์ และรับล็อกกลับมาจากไฟร์สกรีนได้
  - สามารถรับล็อกกลับมาจากไฟร์เบรกซึ่งทำหน้าที่เป็นรีเวิร์สเว็บพร็อกซี
2. ไฟร์วอลล์เป็นแบบอิงผู้ใช้และมีความสามารถพิเศษช่วยให้ผู้ใช้เองสามารถเลือกดูโปรแกรมที่กำลังใช้งานเครือข่ายอยู่และเลือกที่จะปิดโปรแกรมเหล่านั้นได้เอง
3. การติดต่อระหว่างไฟร์สเตชันและไฟร์วอลล์มีความปลอดภัยโดยอยู่บนโปรโตคอล LDAPS ซึ่งใช้ TLS1.0 ในการเข้ารหัส
4. การพัฒนาชุดโปรแกรมพัฒนาขึ้นอย่างสอดคล้องกันในทุกๆส่วน เพื่อให้ไม่มีปัญหาที่อาจทำงานขัดแย้งกันเองจนก่อให้เกิดความเสียหายขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 10.3 ปัญหาอุปสรรคในการพัฒนา

1. การศึกษาโครงการเดิมที่มีอยู่แล้ว ซึ่งจำเป็นที่จะต้องศึกษาโดยละเอียดเพื่อที่จะให้สามารถนำมาพัฒนาต่อได้อย่างถูกต้อง และมีประสิทธิภาพ แต่เนื่องจากโครงการเดิมนั้นมีข้อความอธิบายโปรแกรม (Comment) น้อยมากทำให้การศึกษาโค้ดโปรแกรม และ โครงสร้างการทำงาน ของโปรแกรมทำได้ยาก และ ช้าซึ่งจะต้องเสียเวลานาน
2. การพัฒนาบางส่วนอยู่บนคนละแพลตฟอร์ม ทำให้มีปัญหาติดขัดและข้อจำกัดในการใช้เครื่องมือหรือวิธีการในการพัฒนาอยู่บ้าง
3. ปัญหาการรวมโครงการแต่ละส่วนเข้าด้วยกันเพื่อที่จะให้ทำงานร่วมกันได้อย่างถูกต้องนั้น จะต้องมีการทำความเข้าใจหลักการทำงาน ของโปรแกรมแต่ละส่วนเป็นอย่างดีเพื่อที่จะให้สามารถนำมารวมเข้าด้วยกันแล้ว มีประสิทธิภาพมากที่สุด

### 10.4 แนวทางการพัฒนาต่อ

#### ไฟร์ออลาร์ม

1. เพิ่มความสามารถในการทำงานของโปรแกรมไฟร์ออลาร์ม(FireAlarm) ให้สามารถใช้งานภายใต้สิทธิของผู้ใช้ได้ทุกกลุ่ม
2. พัฒนาโปรแกรมไฟร์ออลาร์ม (FireAlarm) ให้มีการทำงานเป็นแบบวินโดว์เซอร์วิส (Windows Services) ซึ่งสามารถทำให้ป้องกันเครื่องของผู้ใช้ได้แม้ว่าผู้ใช้นั้นยังไม่ได้ล็อกอินเข้ามาในระบบ
3. พัฒนาระบบตรวจจับผู้บุกรุกให้มีความสามารถในการตรวจจับการบุกรุกรูปแบบใหม่ๆ ได้

#### ไฟร์สกรีน

1. เปลี่ยนรูปแบบการรับกฎมาจากไฟร์สเตชัน โดยในโครงการนี้ใช้การตรวจสอบเป็นระยะ (polling) ไปเป็นการส่งสัญญาณมาจากไฟร์สเตชันเมื่อมีการอัปเดตแทน ซึ่งจะช่วยลดปริมาณการใช้งานเครือข่ายลงส่วนหนึ่ง
2. พัฒนารูปแบบการแปลงกฎจากไฟร์สเตชันมาเป็นกฎของโอเพิเทเบิลส์เพื่อที่จะสามารถกำหนดกฎให้กับไฟร์สกรีนได้ยืดหยุ่นขึ้น
3. เพิ่มความปลอดภัยในการติดต่อระหว่างไฟร์สกรีนและไฟร์สเตชัน อาจใช้โปรโตคอล LDAPS ใช้ซีเคียวเชลล์หรือออคัสตัวครอบ (wrapper) อย่างเช่น stunnel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ไฟร์เบรก

1. พัฒนาเพิ่มเติมในส่วนการอัปเดตกฎในการตรวจสอบของซีเคียวริตี้เว็บพรีออกซ์ผ่านไฟร์สแตนด์ได้
2. เพิ่มความปลอดภัยในการติดต่อระหว่างไฟร์เบรกและไฟร์สแตนด์ อาจใช้ซีเคียวเชลล์หรืออากซ์ตัวครอบ (wrapper) อย่างเช่น stunnel



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

เรื่องไกร รังสิพล 2544. เาะระบบ TCP/IP จุดอ่อนโปรโตคอลและวิธีป้องกัน . พิมพ์ครั้งที่ 1

กรุงเทพฯ : บริษัท โปรวิชั่น จำกัด

เรื่องไกร รังสิพล 2545. เปิดโลก Firewall และ Internet Security . พิมพ์ครั้งที่ 1

กรุงเทพฯ : บริษัท โปรวิชั่น จำกัด

สุรวัดน์ ปุณณชัยยะ 2543. เปิดโลกของ TCP/IP และโปรโตคอลของอินเทอร์เน็ต. ครั้งที่ 1

กรุงเทพฯ : บริษัท โปรวิชั่น จำกัด

สุรศักดิ์ สงวนพงษ์ 2545. สถาปัตยกรรมและโปรโตคอลที่ซีพี/ไอที . พิมพ์ครั้งที่ 2 กรุงเทพฯ :

บริษัท ซีเอ็ดยูเคชั่น จำกัด

Anthohn Jones. 2002. **Network Programming for Microsoft Windows Second Edition.**

Canada : Microsoft Press.

Charlie Kaufman. 2002. **Network Security Private Communication in a PUBLIC World.**

New Jersey : Prentice Hall.

Joel Scambray, Sturat McClure. 2001. **Hacking Exposed: Network Security Secrets &**

**Solution Second Edition.** New York : McGraw-Hill Companies.

Douglas E. Comer. 2000. **Internetworking With TCP/IP Principles, Protocols, and**

**Architecture Forth Edition.** New Jersey : Prentice Hall.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้