ชุดโปรแกรมเน็ตเวิร์กไฟร์วอล

NETWORK FIREWALL PROGRAM SUITE

นายทวีศักดิ์ เมฆสิชริน
นายพิพัฒน์ ประภาพรรณพงศ์
นายพีรพงศ์ วงศ์วิวัฒน์กิจ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.2548

ชุดโปรแกรมเน็ตเวิร์กไฟร์วอล

NETWORK FIREWALL PROGRAM SUITE

นายทวีศักดิ์ เมฆศิขริน รหัส 45010295

นายพิพัฒน์ ประภาพรรณพงศ์ รหัส 45010535

นายพีรพงศ์ วงศ์วิวัฒน์กิจ รหัส 45010554

อาจารย์ที่ปรึกษา

อาจารย์ อัครเดช วัชระภูพงษ์

ผศ. ธนา หงษ์สุวรรณ

อาจารย์ ธนัญชัย ตรีภาค

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2548

ปริญญานิพนธ์ปีการศึกษา 2548

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ชุดโปรแกรมเน็ตเวิร์กไฟร์วอล

Network Firewall Program Suite

ผู้จัดทำ

1. นายทวีศักดิ์ เมฆศิขริน รหัสประจำตัว 45010295
2. นายพิพัฒน์ ประภาพรรณพงศ์ รหัสประจำตัว 45010535
3. นายพีรพงศ์ วงศ์วิวัฒน์กิจ รหัสประจำตัว 45010554

อาจารย์ที่ปรึกษา

(อาจารย์ อัครเดช วัชระภูพงษ์)

อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ธนา หงษ์สุวรรณ)

อาจารย์ที่ปรึกษา

(อาจารย์ ธนัญชัย ตรีภาค)

ชุดโปรแกรมเน็ตเวิร์กไฟร์วอล

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

บทคัดย่อ

ในปัจจุบันนี้ระบบรักษาความปลอดภัยของคอมพิวเตอร์และระบบเครือข่ายมีส่วนสำคัญขึ้นเรื่อยๆเป็นลำดับ โดยเฉพาะอย่างยิ่งกับองค์กรทางธุรกิจทั้งหลาย สิ่งแรกๆที่นึกถึงคงจะหนีไม่ไฟร์วอล ด้วยเป็นอุปกรณ์รักษาความปลอดภัยทางเครือข่ายที่มีความจำเป็นเป็นอันดับต้นๆ อย่างไรก็ตาม ไฟร์วอลยังคงแบ่งออกเป็นหลายประเภทและมีคุณสมบัติมีความสามารถแตกต่างกันออกไป ไฟร์วอลชนิดใดชนิดหนึ่งไม่สามารถทำหน้าที่ครอบคลุมได้ทุกด้านอย่างแน่นอน

วิทยานิพนธ์ฉบับนี้เสนอการพัฒนาชุดโปรแกรมเน็ตเวิร์กไฟร์วอลขึ้นเพื่อใช้กับระบบเครือข่ายขนาดกลาง ซึ่งโครงงานนี้ได้ทำการรวบรวมและปรับปรุงความสามารถของไฟร์วอลชนิดต่างให้สามารถใช้งานได้เหมาะสมและเต็มประสิทธิภาพ โครงงานนี้มุ่งเน้นไปที่การทำงานร่วมกันและสามารถควบคุมไฟร์วอลบนส่วนต่างๆทั้งหมดได้จากศูนย์กลาง คือสามารถกำหนดกฎให้กับไฟร์วอลส่วนต่างๆและรวมถึงโปรแกรมสำหรับดูล็อกที่ส่งกลับมาจากไฟร์วอลทุกๆส่วน อันได้แก่ ไฟร์วอลส่วนบุคคลที่มีลักษณะเป็นแบบอิงผู้ใช้พร้อมระบบตรวจจับผู้บุกรุก เกตเวย์ไฟร์วอลซึ่งทำงานอยู่บนเกตเวย์ และสุดท้ายคือพร็อกซี่ไฟร์วอลซึ่งคือซีเคียวรีเวิร์สเว็บพร็อกซี่

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

ปริญญานิพนธ์ฉบับนี้จะไม่สามารถสำเร็จสมบูรณ์ได้ถ้าไม่ได้รับความช่วยเหลือ และความร่วมมือของบุคคลหลาย ๆ ฝ่ายด้วยกันโดยเฉพาะอย่างยิ่งบุคคลผู้ซึ่งเป็นผู้จุดประกายความคิดให้เกิดหัวข้อโครงการนี้ขึ้น นั้นคือท่าน อาจารย์อัครเดช วัชระภูพงษ์ ซึ่งเป็นอาจารย์ที่ปรึกษา รวมทั้งท่านอาจารย์ทุกท่านในภาควิชา วิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้ให้การอบรมสั่งสอน และให้วิชาความรู้ต่าง ๆ ที่ดีแก่คณะผู้จัดทำเสมอมา

ขอขอบพระคุณท่านที่สำคัญที่สุดในชีวิตนี้ ที่ได้ให้กำเนิด ให้การอบรมดูแล และเอาใจใส่ทั้งด้านการศึกษา ด้านการดำเนินชีวิต และด้านอื่นทุกด้าน ที่คงไม่อาจมีใครอีกแล้ว ที่เสมอเหมือนท่านทั้งสองนี้ นั้นคือ บิดา และ มารดา ผู้ซึ่งเป็นที่เคารพรักอย่างยิ่ง ผู้ซึ่งคอยให้กำลังใจในยามที่ไม่มีใครเหลือ ผู้ซึ่งคอยชี้นำทางที่ถูกเสมอ ผู้ที่ให้การสนับสนุนในการทำสิ่งที่ถูก และให้คำชี้นำหากสิ่งที่เราทำไม่ใช่สิ่งที่ดี ไม่ใช่สิ่งที่ควร จึงขอกราบขอบพระคุณมา ณ ที่นี้

สุดท้ายนี้ ขอขอบคุณภาควิชาวิศวกรรมคอมพิวเตอร์ โดยเฉพาะห้องวิจัย และพัฒนาการรักษาความปลอดภัยข้อมูล (ISAG) และ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้เอื้อเฟื้อสถานที่ ให้คณะผู้จัดทำได้ทำการวิจัย และช่วยอำนวยความสะดวกต่าง ๆ ขอขอบคุณเพื่อน ๆ พี่ ๆ น้อง ๆ ชาว ISAG ที่คอยให้ความช่วยเหลือ ในการทำงาน ตลอดเวลา ทางคณะผู้จัดทำขอขอบพระคุณมา ณ ที่นี้ด้วย

นายทวีศักดิ์ เมฆสิขริน
นายพิพัฒน์ ประภาพรรณเพงศ์
นายพีรพงศ์ วงศ์วิวัฒน์กิจ

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

สารบัญรูป

สารบัญรูป (ต่อ)

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของโครงงาน

การใช้งานคอมพิวเตอร์ผ่านทางเครือข่ายนั้น มีความเสี่ยงต่อการถูกบุกรุกทั้งจากภายในและภายนอกองค์กร จึงจำเป็นที่จะต้องมีการป้องกันที่เหมาะสมเพื่อความปลอดภัยในการใช้งาน โดยการใช้ไฟร์วอลซึ่งเป็นเครื่องมือที่ใช้ในการตรวจสอบและป้องกันการใช้งานคอมพิวเตอร์ผ่านทางเครือข่าย และไฟร์วอลนั้นก็มีหลายประเภทที่มีความสามารถแตกต่างกันออกไป สำหรับการใช้งานในระดับเครือข่ายซึ่งเป็นระดับที่มีขอบเขตในการใช้งานขนาดใหญ่ ทำให้การใช้ความสามารถของไฟร์วอลเพียงชนิดเดียวอาจมีความปลอดภัยไม่เพียงพอ จึงได้มีการนำไฟร์วอลประเภทต่างๆมาใช้งานร่วมกันเพื่อให้เกิดประสิทธิภาพมากขึ้น

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อศึกษาโครงสร้าง และลักษณะการทำงานของไฟร์วอลแบบต่างๆ
2. เพื่อปรับปรุงต้นแบบไฟร์วอลบนระบบปฏิบัติการยูนิกซ์และไมโครซอฟท์วินโดวส์
3. เพื่อศึกษาหลักการและโครงสร้างของระบบไดเรกทอรีเซอร์วิส
4. เพื่อเขียนโปรแกรมติดต่อระหว่างคอมพิวเตอร์ผ่านเครือข่าย บนระบบปฏิบัติการไมโครซอฟท์วินโดวส์และยูนิกซ์เพื่อใช้งานร่วมกับ แอคทีฟไดเรกทอรี

1.3 ขอบเขตของโครงงาน

1. เกตเวย์ไฟร์วอลมีความสามารถกรองการเชื่อมต่อที่จะผ่านเข้ามาภายในเครือข่าย ตอบสนองได้เหมาะสม และรายงานเหตุผิดปกติให้กับส่วนกลางได้
2. ไฟร์วอลส่วนบุคคลมีความสามารถควบคุมพฤติกรรมที่น่าสงสัย และรายงานเหตุผิดปกติให้กับส่วนกลางได้
3. ซีเคียวรีเวิร์สเว็บพร็อกซี่ทำหน้าที่เพิ่มความปลอดภัยให้กับเว็บเซิร์ฟเวอร์ และรายงานเหตุผิดปกติให้กับส่วนกลางได้
4. ส่วนควบคุมไฟร์วอลส่วนกลางสามารถกำหนดกฎให้กับไฟร์วอลส่วนบุคคลและเกตเวย์ไฟร์วอลได้ สามารถรับรายงานเหตุผิดปกติที่ถูกส่งมาจากไฟร์วอลในส่วนต่างๆ และนำมาแสดงผลได้

5. ส่วนควบคุมไฟร์วอลส่วนกลางมีการเก็บรายงานเหตุผิดปกติที่รับมาจากส่วนต่างๆไว้ในฐานข้อมูลเพื่อที่จะสามารถนำมาวิเคราะห์สาเหตุของปัญหาในภายหลังได้
6. การติดต่อระหว่างส่วนควบคุมและไฟร์วอลจะผ่าน SSL เพื่อรับรองความปลอดภัยและความถูกต้องของข้อมูล

1.4 วิธีการดำเนินงาน

1. ศึกษารายละเอียดของโปรโตคอลทีซีพี/ไอพี ยูดีพี และโปรโตคอลอื่นๆที่เกี่ยวข้อง
2. ศึกษาโครงสร้างและหลักการทำงานของไฟร์วอลแบบต่างๆ
3. ศึกษาหลักการ โครงสร้างการทำงานของแอ็คทีฟไดเรคทอรี่ และการเขียนโปรแกรมเพื่อประยุกต์ใช้งานร่วมกับแอ็คทีฟไดเรคทอรี่
4. ศึกษาการทำงานของระบบตรวจจับผู้บุกรุกเพื่อนำมาใช้งานร่วมกับไฟร์วอลส่วนบุคคลและรีเวิร์สเว็บพร็อกซี่
5. พัฒนาโปรแกรมไฟร์วอลส่วนบุคคล พร้อมทั้งระบบตรวจจับผู้บุกรุกทางเครือข่ายที่มีการทำงานแบบอิงผู้ใช้ (Directory base firewall) มีความสามารถในการรับกฎจากแอ็คทีฟไดเรคทอรี่ของส่วนกลาง สามารถจัดการกับแอพพลิเคชั่นบนเครื่องที่ทำงานอยู่ได้ และสามารถส่งรายงานเหตุผิดปกติกลับไปยังส่วนกลางได้
6. พัฒนารีเวิร์สเว็บพร็อกซี่ที่สามารถตรวจสอบและป้องกันให้เว็บเซิร์ฟเวอร์มีความปลอดภัยในการใช้งานได้ และสามารถส่งรายงานเหตุผิดปกติกลับไปยังส่วนกลางได้
7. พัฒนาเกตเวย์ไฟร์วอลที่สามารถกรองการเชื่อมต่อที่ผ่านเข้าออกเครือข่ายได้ โดยรับกฎมาจากแอ็คทีฟไดเรคทอรี่ของส่วนกลาง และสามารถส่งรายงานเหตุผิดปกติกลับไปยังส่วนกลางได้

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถรู้และเข้าใจถึงหลักการทำงานของไฟร์วอลแบบต่างๆ
2. เข้าใจหลักการทำงานของบริการไดเรกทอรีและการติดตั้งปรับตั้งค่าของไมโครซอฟท์แอคทีฟไดเรกทอรี รวมถึงการเขียนโปรแกรมติดต่อกับบริการไดเรกทอรี
3. ชุดโปรแกรมไฟร์วอลที่พัฒนาขึ้นมาสามารถใช้งานป้องกันระบบเครือข่ายได้อย่างมีประสิทธิภาพตามเป้าหมาย

1.6 ส่วนประกอบของปริญญานิพนธ์

ปริญญานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 10 บทด้วยกันคือ

บทที่ 1 กล่าวถึงความสำคัญและที่มาของโครงงาน วัตถุประสงค์ของโครงงาน ขอบเขตของโครงงาน วิธีการดำเนินการ ประโยชน์ที่คาดว่าจะได้รับ และส่วนประกอบของปริญญานิพนธ์

บทที่ 2 กล่าวถึงทฤษฎีพื้นฐานที่ใช้ในโครงงาน ซึ่งประกอบด้วยไฟร์วอล แอ็คทีฟไดเรกทอรี เน็ตฟิลเตอร์ พร็อกซี่ ระบบตรวจจับผู้บุกรุกทางเครือข่าย

บทที่ 3 กล่าวถึงการออกแบบการทำงานของระบบที่ใช้กับชุดโปรแกรมเน็ตเวิร์กไฟร์วอล ซึ่งประกอบด้วย รูปโครงสร้างของระบบและแนวคิดในการออกแบบเป็นโปรแกรมในส่วนต่างๆ

บทที่ 4 กล่าวถึงการพัฒนาโปรแกรมไฟร์สเตชั่น ซึ่งประกอบด้วยแนวคิด ขอบเขตและความสามารถ การพัฒนาโปรแกรม และการทำงานของโปรแกรม

บทที่ 5 กล่าวถึงการพัฒนาโปรแกรมไฟร์อลาร์ม 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 กล่าวถึงการทดลองและผลการทดลองของโปรแกรมส่วนต่างๆ

บทที่ 10 เป็นบทวิจารณ์และสรุป ซึ่งกล่าวถึงบทสรุปของโครงงาน วิจารณ์สิ่งที่ได้รับจากโครงงาน และข้อเสนอแนะสำหรับเป็นแนวทางในการพัฒนาต่อ

บทที่ 10
บทสรุปและวิจารณ์

10.1 บทสรุป

ในปัจจุบันนี้ระบบเครือข่ายคอมพิวเตอร์ มีการใช้งานกันอย่างแพร่หลายและมีจำนวนเพิ่มขึ้นมากทำให้เกิดความยุ่งยากในการจัดการดูแล ดังนั้นจึงต้องหาวิธีการในการควบคุมระบบทั้งหมดจากเพียงจุดเดียวเพื่อทำให้การทำงานมีความสะดวกมากขึ้น

การพัฒนาชุดโปรแกรมไฟร์วอลบนเครือข่าย โดยใช้แอ็คทีฟไดเร็คทอรี (Active Directory) เป็นศูนย์กลางการควบคุมนั้นช่วยทำให้การทำงานสะดวกขึ้นมาก โดยที่สามารถทำการ เพิ่ม ลด แก้ไข เปลี่ยนแปลง กฎการป้องกันการบุกรุกของไฟร์วอลส่วนต่างๆที่อยู่ในไดเรกทอรีดาต้าเบสได้ ทำให้สามารถป้องกันการโจมตีได้ตามการกำหนดกฎการป้องกันที่มีอยู่ในไดเรกทอรีดาต้าเบส และสามารถนำข้อมูลการโจมตีที่ตรวจจับได้จากเครื่องเกต์เวย์และเครื่องลูกข่ายมาเก็บลงฐานข้อมูลโดยล็อกมินิเตอร์ โดยสามารถนำข้อมูลที่ได้มาแสดงผลและวิเคราะห์เพื่อกำหนดกฎการป้องกันได้ตามการโจมตีที่เกิดขึ้นได้จริง

10.2 วิจารณ์สิ่งที่ได้จากโครงงาน

1. ไฟร์สเตชั่นทำงานเป็นศูนย์กลางของชุดโปรแกรมเน็ตเวิร์กไฟร์วอล มีความสามารถดังนี้
   - สามารถกำหนดกฎให้กับไฟร์อลาร์มซึ่งเป็นไฟร์วอลส่วนบุคคลพร้อมระดับตรวจจับผู้บุกรุกบนเครื่องไคลเอนต์ในเครือข่ายภายในได้และสามารถรับล็อกกลับเก็บไว้ที่ศูนย์กลาง
   - สามารถกำหนดกฎให้กับไฟร์สกรีนซึ่งทำหน้าที่เป็นเกตเวย์ไฟร์วอล และรับล็อกกลับมาจากไฟร์สกรีนได้
   - สามารถรับล็อกกลับมาจากไฟร์เบรกซึ่งทำหน้าที่เป็นรีเวิร์สเว็บพร็อกซี่
2. ไฟร์อลาร์มเป็นแบบอิงผู้ใช้และมีความสามารถพิเศษช่วยให้ผู้ใช้เองสามารถเลือกดูโปรแกรมที่กำลังใช้งานเครือข่ายอยู่และเลือกที่จะปิดโปรแกรมเหล่านั้นได้เอง
3. การติดต่อระหว่างไฟร์สเตชั่นและไฟร์อาล์มมีความปลอดภัยโดยอยู่บนโปรโตคอล **LDAPS** ซึ่งใช้ **TLS**1.0 ในการเข้ารหัส
4. การพัฒนาชุดโปรแกรมพัฒนาขึ้นอย่างสอดคล้องกันในทุกๆส่วน เพื่อให้ไม่มีปัญหาที่อาจทำงานขัดแย้งกันเองจนก่อให้เกิดความเสียหายขึ้น

10.3 ปัญหาอุปสรรคในการพัฒนา

1. การศึกษาโครงงานเดิมที่มีอยู่แล้ว ซึ่งจำเป็นที่จะต้องศึกษาโดยละเอียดเพื่อที่จะให้สามารถนำมาพัฒนาต่อได้อย่างถูกต้อง และมีประสิทธิภาพ แต่เนื่องจากโครงงานเดิมนั้นมีข้อความอธิบายโปรแกรม (Comment) น้อยมากทำให้การศึกษาโค้ดโปรแกรม และ โครงสร้างการทำงานของโปรแกรมทำได้ยาก และ ช้าซึ่งจะต้องเสียเวลานาน
2. การพัฒนาบางส่วนอยู่บนคนละแพลตฟอร์ม ทำให้มีปัญหาติดขัดและข้อจำกัดในการใช้เครื่องมือหรือวิธีทางในการพัฒนาอยู่บ้าง
3. ปัญหาการรวมโครงงานแต่ละส่วนเข้าด้วยกันเพื่อที่จะให้ทำงานร่วมกันได้อย่างถูกต้องนั้นจะต้องมีการทำความเข้าใจหลักการทำงาน ของโปรแกรมแต่ละส่วนเป็นอย่างดีเพื่อที่จะให้สามารถนำมารวมเข้าด้วยกันแล้ว มีประสิทธิภาพมากที่สุด

10.4 แนวทางการพัฒนาต่อ

ไฟร์อลาร์ม

1. เพิ่มความสามารถในการทำงานของโปรแกรมไฟร์อลาร์ม(FireAlarm) ให้สามารถใช้งานภายใต้สิทธิของผู้ใช้ได้ทุกกลุ่ม
2. พัฒนาโปรแกรมไฟร์อลาร์ม (FireAlarm) ให้มีการทำงานเป็นแบบวินโดว์เซอร์วิส (Windows Services) ซึ่งสามารถทำให้ป้องกันเครื่องของผู้ใช้ได้แม้ว่าผู้ใช้คนนั้นยังไม่ได้ล็อกอินเข้ามาในระบบ
3. พัฒนาระบบตรวจจับผู้บุกรุกให้มีความสามารถในการตรวจจับการบุกรุกรูปแบบใหม่ๆได้

ไฟร์สกรีน

1. เปลี่ยนรูปแบบการรับกฎมาจากไฟร์สเตชั่น โดยในโครงงานนี้ใช้การวนตรวจสอบเป็นระยะ (polling) ไปเป็นการส่งสัญญาณมาจากไฟร์สเตชั่นเมื่อมีการอัพเดตแทน ซึ่งจะช่วยลดปริมาณการใช้งานเครือข่ายลงส่วนหนึ่ง
2. พัฒนารูปแบบการแปลงกฎจากไฟร์สเตชั่นมาเป็นกฎของไอพีเทเบิ้ลส์เพื่อที่จะสามารถกำหนดกฎให้กับไฟร์สกรีนได้ยืดหยุ่นขึ้น
3. เพิ่มความปลอดภัยในการติดต่อระหว่างไฟร์สกรีนและไฟร์สเตชั่น อาจใช้โปรโตคอล LDAPS ใช้ซีเคียวเชลล์หรืออาศัยตัวครอบ (wrapper) อย่างเช่น stunnel

ไฟร์เบรก

1. พัฒนาเพิ่มเติมในส่วนการอัพเดตกฎในการตรวจสอบของซีเคียวรีเวิร์สเว็บพร็อกซี่ผ่านไฟร์สเตชั่นได้

2. เพิ่มความปลอดภัยในการติดต่อระหว่างไฟร์เบรกและไฟร์สเตชั่น อาจใช้ซีเคียวเชลล์หรืออาศัยตัวครอบ (wrapper) อย่างเช่น stunnel

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

โครงงานชุดโปรแกรมเน็ตเวิร์กไฟร์วอลนี้มีการพัฒนาโปรแกรมส่วนต่างๆ คือ ไฟร์สเตชั่น (FireStation),ไฟร์อลาร์ม(FireAlarm),ไฟร์สกรีน(FireScreen), ไฟร์เบรก(FireBreak) และ ล็อกมอนิเตอร์(LogMonitor) ซึ่งในแต่ละส่วนนี้ก็จะมีทฤษฎีที่เกี่ยวข้องอยู่หลายอย่างด้วยกันทั้งทฤษฎีในการพัฒนาโปรแกรมและการนำโปรแกรมมาใช้งานร่วมกันดังที่จะกล่าวถึงต่อไปนี้

2.1 ไฟร์วอล

ไฟร์วอลเป็นเครื่องมือรักษาความปลอดภัยที่ทำงานในเชิงป้องกันโดยทำหน้าที่ควบคุมการเข้าถึงเน็ตเวิร์ก ซึ่งแพ็คเก็ตที่สามารถผ่านเข้า-ออกเน็ตเวิร์กได้นั้น จะต้องเป็นแพ็คเก็ตที่ไฟร์วอลเห็นว่ามีความปลอดภัยโดยอาศัยการเปรียบเทียบคุณสมบัติของแพ็คเก็ตที่จะผ่านไฟร์วอลล์กับกฎที่ได้กำหนดไว้เป็นพื้นฐาน

ไฟร์วอลสามารถแบ่งได้หลายชนิดและหลายประเภทแต่ถ้าคำนึงถึงลักษณะการทำงานของไฟร์วอลที่ใช้ในการตรวจสอบ และควบคุมสามารถแบ่งไฟร์วอลได้เป็น 3 ประ เภท ดังต่อไปนี้คือ

1. แพ็คเก็ตฟิลเตอริง (Packet Filtering)
2. แอพพลิเคชั่นพร็อกซี่ (Application Proxy)
3. สเตตฟูลอินสเปกชั่น (Stateful Inspection)

2.1.1 แพ็คเก็ตฟิลเตอริง (Packet Filtering)

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

ตารางที่ 2.1 แสดงรูปแบบของแอคเซสรูลเบื้องต้น

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

โดยข้อมูลทั้งหมดจะเป็นเสมือนตัวแปรที่จะนำมาเปรียบเทียบกับค่าที่ระบุไว้ในแอคเซสรูล ทีละค่าเงื่อนไขในการเปรียบเทียบของแต่ละตัวจะเป็นตรรกะ และในส่วนของฟิลด์สุดท้าย "Action" หมายถึงสิ่งที่ไฟร์วอลจะกระทำเมื่อค่าในแพ็คเก็ตนั้นตรงกับเงื่อนไข ตัวอย่างเช่น

ตารางที่ 2.2 ตัวอย่างของการตั้งค่าแอคเซสรูลของไฟร์วอล

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

จากการตั้งแอคเซสรูลเช่นนี้จะหมายความว่าจะอนุญาตให้แพ็คเก็ตที่มีต้นทาง IP Address 161.246.5.50 และปลายทางใดๆ ที่ใช้โปรโตคอล TCP หมายเลขพอร์ตปลายทาง 25 ผ่านไฟร์วอลได้หากไฟร์วอลมีแอคเซสรูลนี้เพียงข้อเดียวก็เท่ากับอนุญาตให้โฮสต์เพียงโฮสต์เดียวคือโฮสต์ที่มี IP Address 161.246.5.50 เท่านั้นที่สามารถใช้บริการ SMTP (TCP พอร์ต 25) ไปยังโฮสต์อื่นที่อยู่อีกฟากหนึ่งของไฟร์วอลได้

2.1.2 สเตทฟูลอินสเป็คชั่น (Stateful Inspection)

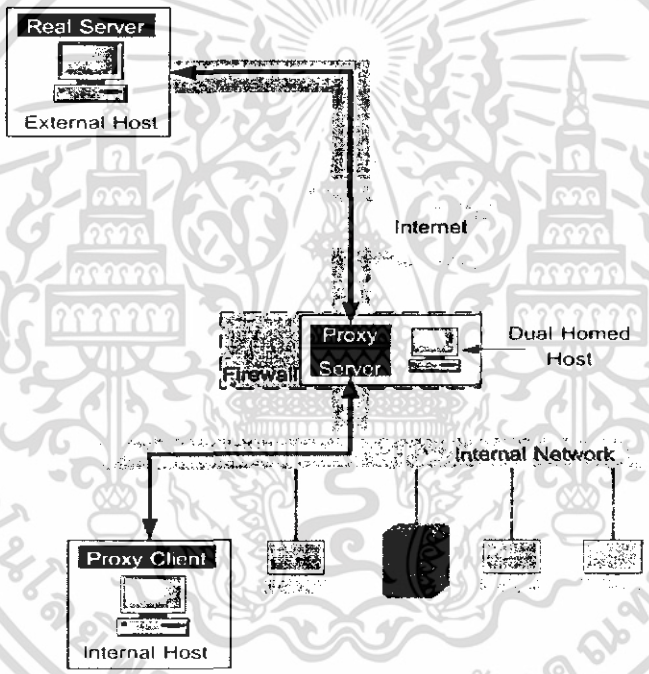
สเตทฟูลอินสเป็คชั่นไฟร์วอล เป็นไฟร์วอลที่ทำการควบคุมทราฟฟิคโดยใช้หลักการของแพ็คเก็ตฟิลเตอริง และการกำหนดแอคเซสรูลเช่นเดียวกัน แต่สเตทฟูลอินสเป็กชั่น มีความสามารถที่เพิ่มจากแพ็คเก็ตฟิลเตอริง คือสามารถฟิลเตอร์แบบมองสถานะและภาพรวมโดยในการพิจารณาว่าจะยอมให้แพ็คเก็ตผ่านไปได้หรือไม่นั้น แทนที่จะดูข้อมูลจากเฮดเดอร์เพียงอย่างเดียว ไฟร์วอลแบบสเตทฟูลจะมีการบันทึกข้อมูลเกี่ยวกับการเชื่อมต่อที่เกิดขึ้นลงในตารางเก็บสถานะ(state table) ก่อนที่จะส่งแพ็คเก็ตนั้นไปยัง IP stack ตารางเก็บสถานะนี้จะเป็นส่วนที่ใช้บันทึกข้อมูลสำหรับแต่ละการเชื่อมต่อที่ถูกต้อง โดยปกติจะเก็บข้อมูล ไอพีของต้นทางและปลายทาง, โปรโตคอล, พอร์ต และแฟล็ก แต่มีไฟร์วอลล์บางยี่ห้อที่เก็บข้อมูลลำดับการส่งข้อมูล(sequence number) เพิ่มด้วย เพื่อใช้ในการตรวจสอบแพ็คเก็ตที่กำลังจะเข้ามาและป้องกันการขโมยเซสชั่น (session hijacking)

และเมื่อได้รับแพ็คเก็ตใหม่เข้ามา ไฟร์วอลแบบสเตทฟูลจะทำการตรวจสอบข้อมูลกับตารางเก็บสถานะว่าเป็นส่วนของการเชื่อมต่อที่สร้างไว้แล้วหรือไม่ โดยจะพิจารณาจากข้อมูล ไอพี และพอร์ตของต้นทางและปลายทาง จะต้องสอดคล้องกับตารางเก็บสถานะ ซึ่งถ้าเป็นส่วนหนึ่งของการเชื่อมต่อจริงก็ไม่มีความจำเป็นใดๆ ที่ต้องตรวจสอบซ้ำอีก

อย่างไรก็ตาม สเตทฟูลไฟร์วอลนี้จะยังไม่สามารถป้องกันการโจมตีที่แทรกซึมมากับการเชื่อมต่อตามปกติได้ เช่น โพรโตคอล FTP นั้น แม้ว่าไฟร์วอลแบบสเตทฟูลจะมีการติดตามให้มีการเปิดพอร์ต 20 และ 21 อย่างถูกต้อง แต่ถ้าหากในเนื้อหาที่ส่งมาเป็นสิ่งที่ประสงค์ร้าย ก็จะไม่สามารถรับรู้และป้องกันการโจมตีแบบนี้ได้

2.1.3 แอพพลิเคชั่นพร็อกซี่ (Application Proxy)

พร็อกซี่ (Proxy) หรือ แอพพลิเคชั่นเกตเวย์ (Application Gateway) เป็นแอพพลิเคชันโปรแกรมที่ทำงานอยู่บนไฟร์วอล ที่ตั้งอยู่ระหว่างเน็ตเวิร์ค 2 เน็ตเวิร์ค ทำหน้าที่เพิ่มความปลอดภัยของระบบเครือข่ายโดยการควบคุมการเชื่อมต่อระหว่างเครือข่ายภายในกับภายนอก พร็อกซี่ จะช่วยเพิ่มความปลอดภัยได้มากเนื่องจากมีการตรวจสอบข้อมูลถึงในระดับของแอพพลิเคชั่นเลเยอร์ (Application Layer) เมื่อไคลเอนต์ (Client) ต้องการใช้บริการจากภายนอก ไคลเอนต์จะทำการติดต่อไปยังพร็อกซี่ก่อน แล้วพร็อกซี่จึงจะติดต่อไปยังเครื่องปลายทางให้แล้วจะมีการเชื่อมต่อ (connection) 2 การเชื่อมต่อ คือ ไคลเอนต์ กับ พร็อกซี่ และ พร็อกซี่ กับเครื่องปลายทาง โดยที่พร็อกซี่จะทำหน้าที่รับข้อมูลและส่งข้อมูลให้ใน 2 ทิศทาง ทั้งนี้พร็อกซี่จะทำหน้าที่ในการตัดสินใจว่าจะให้มีการเชื่อมต่อกันหรือไม่ หรือจะส่งต่อแพ็คเก็ตให้หรือไม่



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

2.2 เทคโนโลยีแอคทีฟไดเรกทอรี (Active Directory)

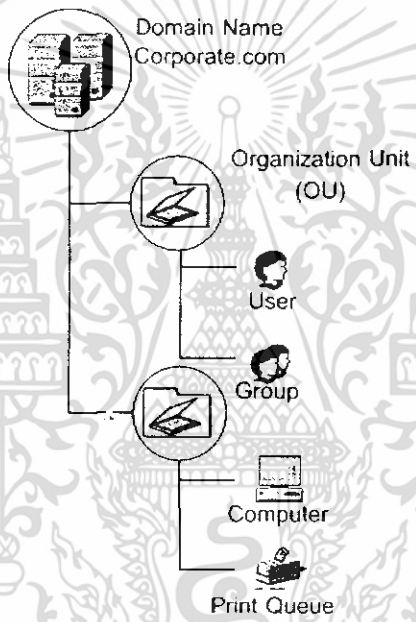
แอคทีฟไดเรกทอรีเป็นเทคโนโลยีบริการไดเรกทอรีของไมโครซอฟท์ เป็นบริการที่เกี่ยวกับการจัดเก็บข้อมูลของทรัพยากรที่มีอยู่ในเครือข่ายลงในไดเรกทอรี โดยแอคทีฟไดเรกทอรีจะมีฐานข้อมูลในตัวเองสำหรับจัดเก็บรายละเอียดของทรัพยากรนั้น

2.2.1 ส่วนประกอบของแอคทีฟไดเรกทอรี

แอคทีฟไดเรกทอรีจะแบ่งการทำงานออกเป็น 2 ส่วน คือ

- Active Directory Service เป็นเครื่องมืออำนวยความสะดวกในการเรียกค้น จัดการกับบัญชีรายชื่อของผู้ใช้และรายชื่อกลุ่ม และสามารถค้นหารายชื่อทรัพยากรในระบบที่มีคุณสมบัติตามที่ผู้ใช้ต้องการ ซึ่งแอคทีฟไดเรกทอรีเซอร์วิสจะเข้าไปค้นหาข้อมูลจากแอคทีฟไดเรกทอรีดาต้าเบสแล้วแสดงรายละเอียดต่างๆขึ้นมา
- Active Directory Database ทำหน้าที่ในการจัดเก็บข้อมูลในระบบไดเรกทอรี โดยการเข้าถึงดาต้าเบสจะต้องผ่าน การล็อกอินจึงจะมีสิทธิในการใช้งาน

2.2.2 ข้อมูลที่เก็บอยู่ใน Active Directory Database



รูปที่ 2.2 โครงสร้างโดยรวมของแอคทีฟไดเรกทอรีดาต้าเบส (Active Directory Database)

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

- เป็นที่เก็บ Group Policy Object (GPO) เพื่อใช้ในการควบคุม และจัดการเครื่องคอมพิวเตอร์ ของผู้ใช้
- อื่นๆอีกมากมายแล้วแต่จะขยายเพิ่มเติมในอนาคตเช่นออบเจ็กต์ที่เป็นตัวแทนของ อุปกรณ์สื่อสารในระบบเน็ตเวิร์ก

2.2.3 สกีมา (Schema)

เป็นฐานข้อมูลส่วนย่อยที่เก็บอยู่ในแอคทีฟไดเรกทอรีคาต้าเบส ซึ่งทำหน้าที่เก็บรวบรวมข้อกำหนดเกี่ยวกับคลาสของออบเจ็กต์ทุกคลาส เช่น ออบเจ็กต์ที่อยู่ในแต่ละคลาสจะต้องมีแอตตริบิวต์อะไร ค่าของแต่ละแอตตริบิวต์มีคุณสมบัติเป็นอะไร Schema สามารถเพิ่มขยายได้โดยผู้บริหารระบบเครือข่าย ได้แก่ การเพิ่มคลาสประเภทใหม่ๆ เข้าไป หรือการเพิ่ม แอตตริบิวต์ใหม่ให้กับคลาสเดิมที่มีอยู่แล้ว

2.2.4 การติดต่อกับ Active Directory (LDAP Protocol)

LDAP (Lightweight Directory Access Protocol) เป็นโปรโตคอลที่พัฒนามาจาก X.500 ซึ่งใช้ในการเข้าถึงและอัพเดตข้อมูลของไดเรกทอรี และเป็นโปรโตคอลมาตรฐานสำหรับการสืบค้นหรือเปลี่ยนแปลงข้อมูลบนบริการไดเรกทอรี โดย LDAP จะทำงานอยู่บนโปรโตคอล TCP/IP อีกทีหนึ่ง ซึ่งไดเรกทอรีจะมีรูปแบบเป็นโครงสร้างต้นไม้ที่แต่ละ entry จะประกอบไปด้วยชุดของ attribute และค่าของ attribute นั้นๆ ซึ่งบริการไดเรกทอรีของไมโครซอฟท์ (Microsoft Active Directory Service) ก็ทำงานอยู่บนโปรโตคอล LDAP เช่นเดียวกัน

ในส่วนการติดต่อระหว่างไคลเอ็นต์ที่ต้องการใช้บริการของเซิร์ฟเวอร์ LDAP มีหลักการเบื้องต้นดังนี้ LDAP ไคลเอ็นต์ทำการเริ่มต้น LDAP session โดยเชื่อมต่อเข้ามายัง LDAP เซิร์ฟเวอร์ ซึ่งปกติจะใช้งานพอร์ตหมายเลข 389 (636 สำหรับ SSL/TLS) หลังจากนั้นไคลเอนต์ก็จะส่งคำสั่งต่างๆไปยังเซิร์ฟเวอร์ซึ่งจะส่งคืนค่าที่ต้องการให้กับทางไคลเอ็นต์โดยมีข้อน่าสังเกตที่ทางไคลเอนต์ไม่มีความจำเป็นที่จะต้องรอจนกว่าเซิร์ฟเวอร์จะส่งคืนค่ามาแล้วจึงส่งคำสั่งถัดไปสามารถส่งคำสั่งต่อเนื่องได้เรื่อยๆ 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

```
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 Doe, dc=example, dc=com
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: top
```

dn เป็นชื่อของ entry เอง ไม่ได้เป็น attribute ใด attribute หนึ่งใน 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	: พิสูจน์สิทธิ์และระบุเวอร์ชั่นของ LDAP protocol
-Search	: ค้นหาข้อมูลที่ต้องการในไดเรกทอรี
-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	: คำสั่งทั่วๆไปใช้ร่วมกับคำสั่งอื่นๆ
-Unbind	: ยกเลิกการเชื่อมต่อ (ไม่ได้เป็นคำสั่งตรงข้ามกับ Bind แต่อย่างใด)

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 เคอร์แนลตั้งแต่เวอร์ชั่น 2.4.x จนถึงปัจจุบันที่เวอร์ชั่น 2.6.x โดยจัดเป็นยุคที่ 3 ของไฟร์วอลบนลินุกซ์นับตั้งแต่ ipfwadm ซึ่งมีในลินุกซ์เคอร์แนล 2.0.x (ipfwadm พัฒนามาจาก ipfw ของ BSD) จากนั้นจึงกลายมาเป็น ipchains ในลินุกซ์เคอร์แนล 2.2.x โดยมี Rusty Russel เป็นแกนหลักในทีมพัฒนา และได้พัฒนาต่อมาจนกลายมาเป็น netfilter/iptables ในที่สุด

netfilter นั้นเป็นชื่อของโค้ดในส่วนที่ทำหน้าที่กรองแพ็คเก็ต โดยต่อไปนี้จะเรียกรวมว่าเป็น iptables ซึ่งเป็นคำสั่งในการเรียกใช้งาน netfilter/iptables

iptables เป็นไฟร์วอลที่มีความสามารถในการกรองแพ็คเก็ตแบบสเตทฟูล อินสเปคชั่น สามารถทำ Network Address Translation (NAT) ได้ และสามารถทำ packet mangling เช่นการแก้ไขค่า ToS หรือ Mark ได้ เป็นต้น

ในโครงงานนี้จะให้ความสำคัญไปที่ความสามารถในการกรองแพ็คเก็ตของ iptables เป็นสำคัญซึ่งเป็นความสามารถสำคัญของ iptables ในการใช้งานเป็นไฟร์วอล

2.3.1 การทำงานเบื้องต้นของ iptables

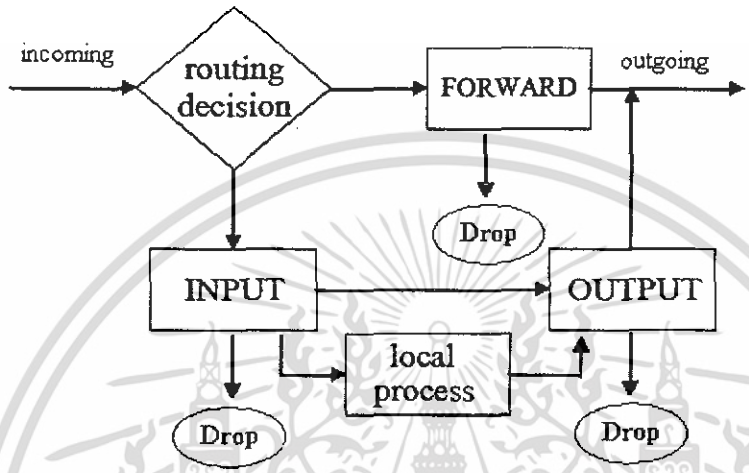
แพ็คเก็ตที่ผ่านเข้ามายังเครื่องไฟร์วอลจะผ่านเข้าไปยังเทเบิลต่างๆของ iptables ซึ่งแต่ละเทเบิลก็จะมีหน้าที่ต่างกันไปโดยแต่ละเทเบิลจะมีหลาย chain สำหรับแพ็คเก็ตรูปแบบต่างๆ โดยมีรายละเอียดดังต่อไปนี้

- **Tables**

  iptables มีเทเบิลที่สร้างไว้อยู่แล้วทั้งหมด 3 เทเบิล คือ

1. Filter Table

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



รูปที่ 2.3 การเดินทางของแพ็คเก็ตใน Filter Table

กรณีที่แพ็คเก็ตถูกส่งผ่านไปยังเครื่องอื่นแพ็คเก็ตนั้นจะผ่านเข้าไปใน FORWARD chain ก่อนที่จะส่งผ่านไปยังปลายทางที่ระบุหรืออาจจะถูก drop ทิ้งตามแต่กฎที่ตั้งไว้

ถ้าแพ็คเก็ตนั้นมีเป้าหมายเป็นเครื่องปัจจุบัน แพ็คเก็ตนั้นจะผ่านเข้าไปใน INPUT chain โดยที่จะถูกนำไปประมวลผลใน userspace หรืออาจจะถูก drop ทิ้งตามแต่กฎที่ตั้งไว้

กรณีที่แพ็คเก็ตถูกสร้างจากเครื่องปัจจุบัน แพ็คเก็ตจะผ่านเข้าไปใน OUTPUT chain ก่อนที่จะถูกส่งออกไปหรืออาจจะถูก drop ทิ้งตามแต่กฎที่ตั้งไว้

2. Nat Table

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

1) PREROUTING ทำการแปลงแพ็คเก็ตที่เข้ามาในส่วนของ DNAT และดูแลเรื่อง REDIRECT
2) POSTROUTING ทำการแปลงแพ็คเก็ตในส่วนของ SNAT และดูแลเรื่อง REDIRECT
3) OUTPUT ทำการแปลงแพ็คเก็ตที่จะส่งออกไปในส่วนของ DNAT และดูแลเรื่อง REDIRECT

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 โดยกฎที่เขียนขึ้นจะเป็นตัวบอกเคอร์เนลว่าให้กระทำอย่างไร (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 ค่าที่ใช้กับcommand นี้คือ ACCCEPT, DROP

-E ใช้เปลี่ยนชื่อ chain ใหม่

ซึ่งการใช้ command ข้างต้นนั้นสามารถใช้ร่วมกับออปชั่นบางอย่างได้ 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 เพื่อให้แสดงจำนวนแพ็คเก็ตและจำนวนของ 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

- 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
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 กับกฎ นี้

-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 แพ็คเก็ตรวมกันเป็นแพ็คเก็ตที่สมบูรณ์ดังเดิม ข้อมูลที่เป็น fragment แพ็คเก็ตจะมีเฮดเดอร์ที่สมบูรณ์แค่แพ็คเก็ตแรกเท่านั้นแพ็คเก็ตที่ตามมาจะมีเฮดเดอร์แค่บางส่วนคือ ไอพีแอดเดรส ไม่มีข้อมูลของโปรโตคอลแนบมาด้วย ดังนั้นการตรวจสอบข้อมูลเฮดเดอร์ของ TCP, UDP, ICMP จึงไม่สามารถทำได้ในแพ็คเก็ตที่สองเป็นต้นมา

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

- TCP extension

ถ้ามีการเรียกใช้ -p tcp ตัว TCP extension ก็จะถูกโหลดมาใช้งานโดยอัตโนมัติ โดยมีออปชั่นให้เลือกใช้งานดังนี้

--tcp-flags mask flags : mask นั้นหมายถึง flag ที่ต้องการตรวจสอบ และ flag เป็นตัวที่บ่งชี้ว่า flag ใดต้องถูก set บ้าง

--source-port หรือ --sport สามารถใช้ได้ทั้งตัวเลขและตัวอักษร ระบุเป็น port เดี่ยว หรือช่วงของ 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 ที่ต้องการมีออปชั่นให้เลือกใช้งานดังต่อไปนี้

mac รูปแบบการใช้งาน: -m mac ใช้ตรวจสอบ source MAC address ว่าตรงกับค่าที่ระบุไว้หรือไม่ มีประโยชน์สำหรับ *PREROUTING และ INPUT chain*

limit รูปแบบการใช้งาน: -m limit ใช้เพื่อจำกัดจำนวนของการ match ที่อาจจะมากเกินไป เป็นประโยชน์สำหรับ rule ที่วางไว้ตอนท้ายสุดของ chain (ใช้ร่วมกันกับ DROP policy) ซึ่งส่วนใหญ่เป็น rule ที่ใช้เก็บข้อมูลลงล็อกไฟล์ ซึ่งถ้าผู้บุกรุกส่งแพ็คเก็ตที่ไม่เข้าข่าย rule ใดๆ ใน chain จนกระทั่งมาถึง rule ที่ทำหน้าที่เก็บล็อกนี้ถ้าแพ็คเก็ต ที่เข้ามามีจำนวนมากก็อาจจะทำให้ฮาร์ดดิสก์เต็มได้ ดังนั้นจึงต้องใช้จำกัดจำนวนในการเก็บข้อมูลลงล็อก ซึ่งมีออปชั่นให้ใช้งานดังนี้คือ

State Match รูปแบบการใช้งานคือ -m state หรือ --match state เป็นโมดูลที่ใช้ประโยชน์ได้เป็นอย่างดี มีออปชั่นให้ใช้งานดังนี้

NEW – แพ็คเก็ตที่เป็นตัวสร้างการเชื่อมต่อใหม่

ESTABLISHED – แพ็คเก็ตที่เกี่ยวข้องกันกับการเชื่อมต่อที่สร้างไว้แล้ว

RELATED - แพ็คเก็ตที่เกี่ยวข้องกับการเชื่อมต่อที่สร้างไว้แล้ว แต่ไม่ใช่ส่วนหนึ่งส่วนใดของการเชื่อมต่อนั้น

INVALID – เป็นแพ็คเก็ตที่ไม่เกี่ยวข้องกับส่วนอื่นเลย

- Target

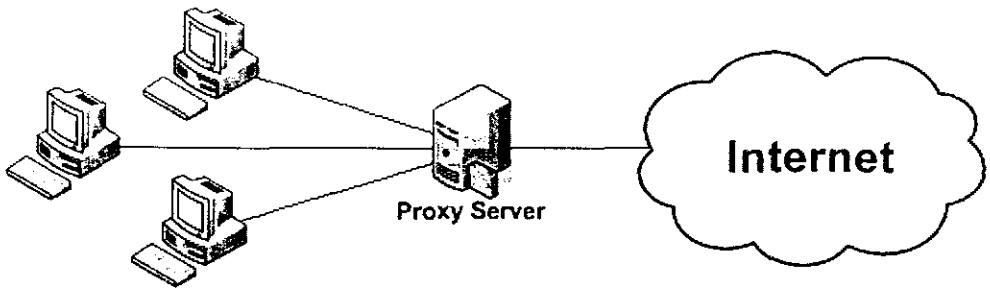
เป็นส่วนที่ระบุ action กับแพ็คเก็ตต่างๆที่ match กับกฎที่ได้ตั้งขึ้นซึ่งมี target ในแบบต่างๆ ดังนี้

- ACCEPT อนุญาตให้แพ็คเก็ตที่ match กับกฎนี้สามารถผ่านไปได้
- 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 พิเศษใช้สำหรับส่งต่อแพ็คเก็ตไปยังแอปพลิเคชั่นที่เขียนขึ้นมารองรับโดยเฉพาะ โดยจะต้องมี queue handler และแอปพลิเคชั่นเป็นส่วนประกอบที่จะทำงานร่วมกัน
- User-defined chain ผู้ใช้สามารถสร้าง chain ใหม่ได้โดยต้องใช้ตัวอักษรตัวเล็กทั้งหมด สำหรับ chain ที่สร้างขึ้นใหม่ เมื่อแพ็คเก็ต match กับกฎที่เป็น user-defined chain แพ็คเก็ตจะถูกนำไปตรวจสอบใหม่โดย user-defined chain นั้นๆและถ้าใน chain ไม่มีการตัดสิน action ใดๆตัวแพ็คเก็ตก็สามารถย้อนกลับมายังกฎถัดไปใน chain เริ่มต้นได้

2.4 พร็อกซี่

2.4.1 เว็บพร็อกซี่ (Regular Webproxy)

เว็บพร็อกซี่มีลักษณะการใช้งานคือ การอนุญาตให้เครื่องไคลเอ็นต์ภายในเครือข่ายสามารถติดต่อกับอินเตอร์เน็ตได้โดยผ่านเว็บพร็อกซี่ โดยเว็บพร็อกซี่จะรอรับการร้องขอจากเครื่องไคลเอ็นต์ภายในเครือข่าย แล้วทำการส่งต่อการร้องขอนี้ไปยังเครื่องเซิร์ฟเวอร์ จากนั้นก็รอรับการตอบกลับมาของเซิร์ฟเวอร์แล้วทำการส่งต่อการตอบรับนั้นให้กับไคลเอ็นต์ ซึ่งเว็บพร็อกซี่นี้สามารถทำการแคชข้อมูล คือการเก็บสำเนาข้อมูลจากอินเตอร์เน็ตมาไว้ที่เว็บพร็อกซี่ ทำให้เว็บพร็อกซี่นั้นไม่ต้องทำการร้องขอข้อมูลที่เคยทำการการร้องขอไปแล้วซ้ำ สามารถที่จะส่งข้อมูลเหล่านี้ให้กับเครื่องไคลเอ็นต์ที่ทำการร้องขอข้อมูลได้ทันที ทำให้การใช้งานอินเตอร์เน็ตนั้นทำได้รวดเร็วมากขึ้น



รูปที่ 2.4 แสดงตำแหน่งของเว็บพ็อกซี่ในเครือข่าย

ความสามารถและการใช้งานเว็บพร็อกซี่

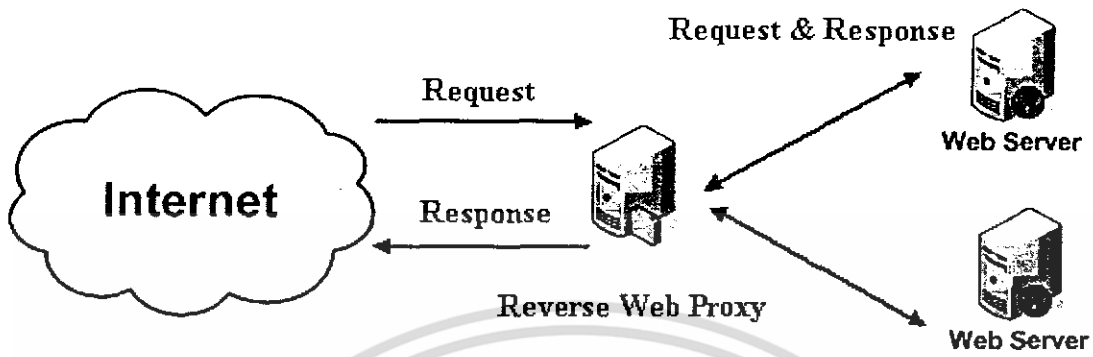
- อนุญาตและจำกัดการใช้งานอินเตอร์เน็ตของเครื่องไคลเอนต์โดยวิธีการตรวจสอบไอพีแอดเดรส
- แคชข้อมูลจากภายนอกให้เครื่องภายในเครือข่ายที่ต้องการข้อมูลที่เหมือนกันเรียกใช้งานได้เร็วขึ้น
- สามารถควบคุมให้การเข้าใช้งานอินเตอร์เน็ตและซับเน็ตที่ต้องการได้โดยกำหนด URL เป็นหลัก
- อนุญาตหรือปฏิเสธการร้องขอจากไคลเอนต์โดยมีรูปแบบเงื่อนไขในการตัดสิน
- กำหนด URL ที่จะใช้ในการกรองหรืออนุญาตให้เรียกใช้ผ่านเครื่องเซิร์ฟเวอร์ได้

2.4.2 รีเวิร์สเว็บพร็อกซี่ (Reverse Webproxy)

รีเวิร์สเว็บพร็อกซี่ เป็นพร็อกซี่ที่ทำงานบนฝั่งเซิร์ฟเวอร์แทนที่จะทำงานบนฝั่งไคลเอ็นต์จึงเรียกว่าเป็นการทำงานแบบรีเวิร์ส รีเวิร์สเว็บพร็อกซี่เป็นแอพพลิเคชั่นพร็อกซี่สำหรับเซิร์ฟเวอร์ที่ใช้โปรโตคอล HTTP รีเวิร์สเว็บพร็อกซี่จะทำงานอยู่ระหว่างไคลเอ็นต์กับเว็บเซิร์ฟเวอร์ทำหน้าที่เป็นเกตเวย์ของเว็บเซิร์ฟเวอร์ หรือ ไอพีแอดเดรสที่ใช้สำหรับรับการร้องขอจากภายนอก ทำให้ไคลเอ็นต์เห็นรีเวิร์สเว็บพร็อกซี่เป็นเครื่องเว็บเซิร์ฟเวอร์ และดูแลทราฟฟิกก่อนที่จะเข้าถึงเว็บเซิร์ฟเวอร์ โดยตรวจสอบดูการร้องขอที่ส่งเข้ามานั้นเคยมีการแคชไว้หรือไม่ถ้ามีอยู่ในแคชก็จะทำการตอบกลับข้อมูลในแคชให้กับไคลเอ็นต์ ทำให้เว็บเซิร์ฟเวอร์ไม่ต้องสร้างข้อมูลตอบกลับหรือผลลัพธ์ทุกครั้งที่ได้รับการร้องขอ เว้นแต่ข้อมูลประเภท Dynamic ที่จะต้องมีการส่งการร้องขอไปยังเว็บเซิร์ฟเวอร์เพื่อประมวลผลทุกครั้ง โดยก่อนใช้งานรีเวิร์สเว็บพร็อกซี่ต้องมีการกำหนด prefix mapping ให้แก่รีเวิร์สเว็บพร็อกซี่ก่อนซึ่ง prefix mapping มี 2 ชนิด คือ

1) regular mapping ใช้สำหรับบอกรีเวิร์สเว็บพร็อกซี่ว่า URL prefix ไหนถูกแทนที่และบอก URL ปลายทางจริง

2) reverse mapping เป็นส่วนแปลง URL prefix ไปเป็น URL ของรีเวิร์สเว็บพร็อกซี่ซึ่งเป็นชื่อเว็บเซิร์ฟเวอร์จริงที่ใช้ในการติดต่อสื่อสารกับไคลเอนต์



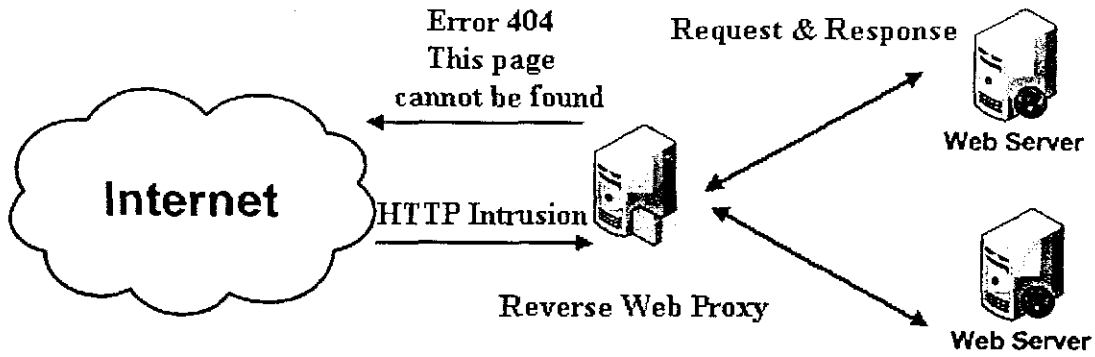
รูปที่ 2.5 การทำงานของรีเวิร์สเว็บพร็อกซี่

ในการใช้งานรีเวิร์สเว็บพร็อกซี่นั้นจะมีการใช้งานร่วมกับไฟร์วอลโดยมีการกำหนดกฎให้มีเพียงแต่ไอพีแอดเดรสของรีเวิร์สเว็บพร็อกซี่เท่านั้นที่สามารถติดต่อกับเว็บเซิร์ฟเวอร์ได้ ซึ่งหากมีการติดต่อสื่อสารเป็นการร้องขอจากไคลเอ็นต์มายัง URL ของรีเวิร์สเว็บพร็อกซี่ซึ่งเป็นทางที่ใช้ในการติดต่อกับเว็บเซิร์ฟเวอร์ รีเวิร์สเว็บพร็อกซี่ก็จะทำการตรวจสอบการร้องขอนั้นกับ prefix mapping ว่าจะต้องส่งการร้องขอนั้นให้กับเว็บเซิร์ฟเวอร์ใด จากนั้นจึงจะส่งการร้องขอนั้นไปยังไฟร์วอลเพื่อทำการส่งการร้องขอนั้นให้กับเว็บเซิร์ฟเวอร์

ทางฝั่งของเว็บเซิร์ฟเวอร์เมื่อประมวลผลของการร้องขอที่ได้รับจากไคลเอ็นต์เสร็จแล้วจะทำการส่งผลลัพธ์กลับไปยังไคลเอ็นต์ ก็จะต้องทำการส่งผ่านไปยังรีเวิร์สเว็บพร็อกซี่ก่อนเช่นกัน ซึ่งเมื่อรีเวิร์สเว็บพร็อกซี่ได้รับผลลัพธ์ที่จะส่งกลับไปยังไคลเอ็นต์ ก็จะนำไปตรวจสอบกับ reverse mapping และทำการเปลี่ยน URL และ HTTP header กลับเป็น URL ที่ใช้ในการติดต่อกับไคลเอ็นต์ก่อนที่จะส่งผลลัพธ์นั้นกลับไปยังไคลเอ็นต์ ซึ่งไคลเอ็นต์ก็จะมองเห็นว่าการร้องขอนั้นถูกตอบรับโดยรีเวิร์สเว็บพร็อกซี่ แต่แท้จริงแล้วถูกตอบโดยเว็บเซิร์ฟเวอร์

2.4.3 รีเวิร์สเว็บพร็อกซี่แบบปลอดภัย (Secure Reverse Web Proxy)

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



รูปที่ 2.6 การทำงานของรีเวิร์สเว็บพร็อกซี่แบบปลอดภัย

2.5 ระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ (Intrusion Detection System)

ระบบตรวจจับผู้บุกรุกทางเครือข่ายทางคอมพิวเตอร์นี้จะมีการเก็บข้อมูลของแพ็คเก็ตต่างๆ ที่เข้ามาสู่ระบบเครือข่ายแล้วนำมาวิเคราะห์เปรียบเทียบกับกฎต่างๆที่กำหนดไว้ รวมถึงนโยบายขององค์กร เพื่อตรวจสอบว่ามีสิ่งผิดปกติเกิดขึ้นกับระบบหรือไม่ หากเกิดสิ่งผิดปกติก็จะแจ้งเตือนไปยังผู้ดูแลระบบหรือเก็บไว้ในล็อกไฟล์ต่อไป

การตรวจจับผู้บุกรุกทางคอมพิวเตอร์สามารถแบ่งตามลักษณะของการโจมตีได้ 5 ประเภท

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

2.5.1 ระบบตรวจจับผู้บุกรุกที่ใช้งานในไฟร์อลาร์ม

สำหรับระบบตรวจจับผู้บุกรุกทางเครือข่ายที่สร้างขึ้นเพื่อใช้งานร่วมกับเพอร์ซันนอลไฟร์วอลนั้นมุ่งเน้นศึกษาในเรื่องของการออกแบบพัฒนาระบบตรวจจับการสำรวจระบบ และการโจมตีเพื่อให้ปิดบริการ ซึ่งเป็นลักษณะของการโจมตีระบบที่สำคัญและมีแนวโน้มเพิ่มขึ้นมากในปัจจุบัน มีรายละเอียดดังต่อไปนี้

1. การสำรวจระบบ

การบุกรุกเพื่อทำการสำรวจระบบเป็นเก็บข้อมูลของระบบ เพื่อใช้ในการโจมตีโดยข้อมูลที่ผู้โจมตีมักต้องการได้แก่ หมายเลขไอพีแอดเดรส ชื่อเครื่อง โครงสร้างทางเครือข่ายของระบบ

เป้าหมาย พอร์ตหรือบริการที่เครื่องเป้าหมายเปิด และระบบปฏิบัติการรวมทั้งเวอร์ชั่นที่ติดตั้งบนเครื่องเป้าหมาย สำหรับการสำรวจระบบที่สามารถทำการตรวจจับได้มีอยู่ 3 วิธีการสำรวจดังนี้

1.1 ปิงสวีป (Ping sweep)

การตรวจจับการปิงสวีปทำได้โดยการตรวจสอบดูแพ็คเก็ต ICMP Request (Type8) ที่เข้ามายังระบบ หากมีแพ็คเก็ตลักษณะนี้จำนวนมากเกินกว่าที่กำหนดและมีปลายทางแตกต่างกันจะสามารถสรุปได้ว่าในเครือข่ายกำลังถูกสำรวจโดยการปิงสวีป

1.2 การสแกนพอร์ต

การตรวจสอบการสแกนพอร์ตทำได้โดยการตรวจดูแพ็คเก็ตที่มีหมายเลขพอร์ตปลายทางในลักษณะกระจาย คือแพ็คเก็ตมีการส่งไปยังเครื่องๆ เดียวแต่มีการส่งไปยังพอร์ตต่างๆ กันเป็นจำนวนมากเกินกว่าที่กำหนดจะสามารถสรุปได้ว่ากำลังถูกสำรวจโดยการสแกนพอร์ต

1.3 การตรวจสอบระบบปฏิบัติการ

การตรวจจับการตรวจสอบระบบปฏิบัติการสามารถตรวจสอบได้จากการพิจารณาแฟล็กที่ถูกส่งไปในชั้นทีซีพีของทุกๆพอร์ตว่าเป็นแพ็คเก็ตแบบผิดปกติหรือไม่ กล่าวคือในแต่ละสเตทของทีซีพีโปรโตคอลนั้นจะมีรูปแบบแฟล็กตายตัวอยู่ ตามสถานะปัจจุบันของสเตทของทีซีพี เช่น หากต้องการเริ่มต้นการเชื่อมต่อโปรโตคอลทีซีพี จะต้องกำหนดให้แฟล็ก ACK เป็น 1 ส่วนแฟล็กอื่นต้องเป็น 0 หรือหากต้องการยกเลิกการเชื่อมต่อโปรโตคอลทีซีพี จะต้องกำหนดให้แฟล็ก FIN เป็น 1 ส่วนแฟล็กอื่นเป็น 0 เป็นต้น

วิธีการตรวจจับที่ใช้ในโปรแกรมนั้นได้ทำการตรวจจับโดยหากแพ็คเก็ตมีแฟล็ก ACK และ FIN เป็น 1 พร้อมกันในแพ็คเก็ตเดียวกัน จะระบุว่าเป็นการบุกรุกโดยการสำรวจเพื่อระบุระบบปฏิบัติการเนื่องจากการตรวจจับวิธีนี้เป็นกรณีมาตรฐานที่โปรแกรมที่ทำการระบุระบบปฏิบัติการทุกโปรแกรมจะนำมาตรวจสอบ และเป็นแฟล็กที่ไม่สามารถเกิดได้จริงเมื่อมีการใช้งานโปรโตคอลทีซีพี

2. การโจมตีเพื่อให้ปิดบริการ

การโจมตีเพื่อให้ปิดบริการคือการกระทำใดๆ ที่ทำให้ระบบเป้าหมายไม่สามารถให้บริการบางอย่างได้ หรือไม่สามารถให้บริการต่อไปได้ การโจมตีระบบเครือข่ายคือการสร้างภาระให้กับเครือข่าย แบ่งได้ 3 ประเภทดังนี้

2.1 การส่งแพ็คเก็ตปริมาณมาก

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

หลายประการ เช่น ความเร็วของเครือข่าย ความเร็วของหน่วยประมวลผลเครื่อง ปริมาณหน่วยความจำในเครื่องเป็นต้น

การหาค่าที่ระบบยอมรับได้นี้ สามารถทำได้โดยการเชื่อมต่อกับระบบที่วิเคราะห์ จากนั้นหาจำนวนแพ็คเก็ตที่เข้ามาในระบบในลักษณะการใช้งานปกติของแต่ละช่วงเวลา จากนั้นนำค่าสูงสุดที่ได้มาเป็นค่าที่ระบบยอมรับได้ ค่าที่ผ่านการวิเคราะห์และยอมรับได้โดยปกติมีค่าประมาณประมาณ 20, 000 - 30,000 แพ็คเก็ตต่อนาที

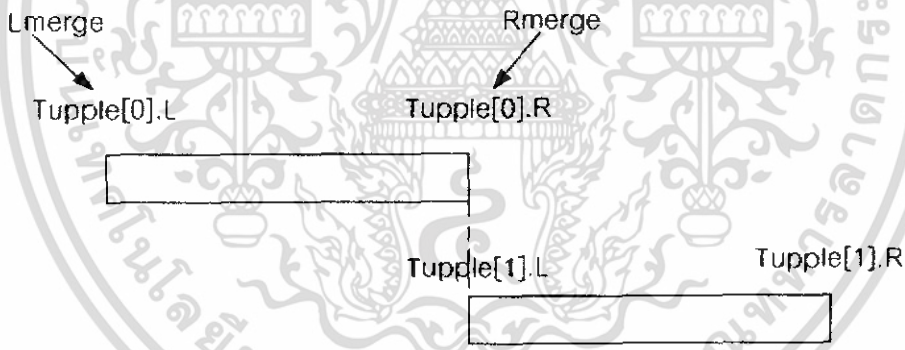
2.2 ความผิดปกติของแฟล็กเมนต์

การตรวจสอบความผิดปกติ ของแฟล็กเมนต์มีขั้นตอนค่อนข้างซับซ้อน ซึ่งแยกอธิบายตามประเภทของความผิดปกติได้ ดังต่อไปนี้

2.2.1 การส่งแพ็คเก็ตที่มีลำดับผิดปกติ และ แพ็คเก็ตที่มีขนาดเหลื่อมล้ำกัน

การวิเคราะห์ความผิดปกติของแพ็คเก็ตในลักษณะนี้ต้องวิเคราะห์หลังจากกระบวนการรีแอสเซมเบิ้ลไปแล้ว ดังนั้นจึงนำบัฟเฟอร์เข้ามาช่วยในการเก็บข้อมูล เพื่อนำมาวิเคราะห์ ดังนี้

- Fragment Buffer คือ บัฟเฟอร์ที่เก็บข้อมูลในการวิเคราะห์ของแพ็คเก็ตไอพีและข้อมูลที่จำเป็นอื่นๆไว้ ได้แก่ Source_IP, Destination_IP, Identification, Protocol, Sec, PointArray และ Array_Fragment การเก็บข้อมูลจะเก็บในลักษณะของโครงสร้างข้อมูลแบบลิงค์ลิสต์



รูปที่ 2.7 แสดงการเก็บข้อมูลของตัวแปร Tupple

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

ตารางที่ 2.3 แสดงโครงสร้างการเก็บข้อมูลของ Fragment Buffer

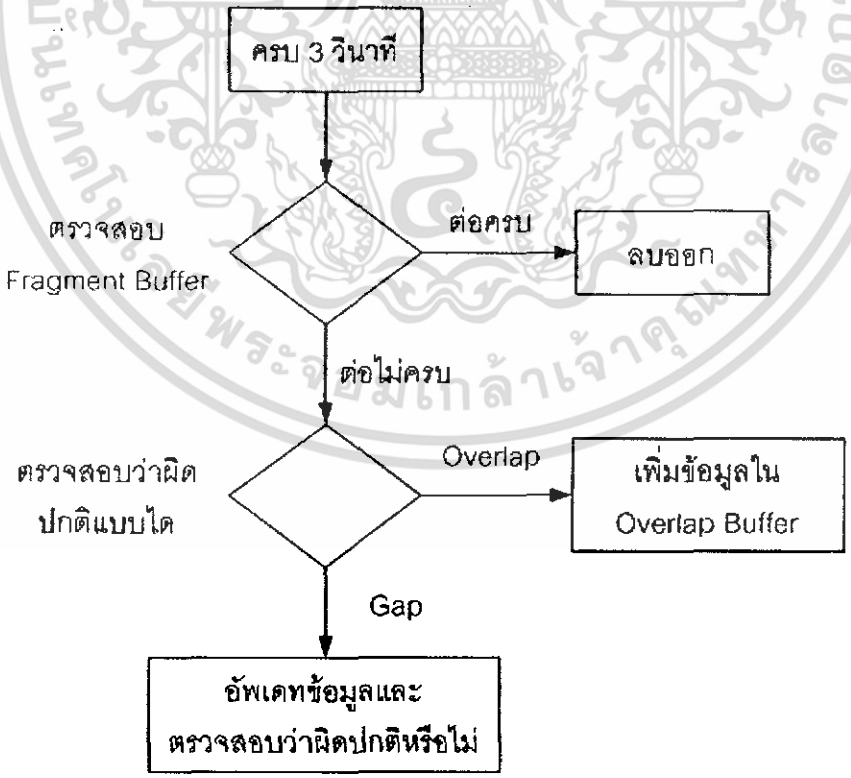
IP_Src	IP_Dst	Identification	Protocal	Sec	Pointarray	Arrau_Fragment

ตารางที่ 2.4 แสดงโครงสร้างการเก็บข้อมูลของ Fragment

Flag_U	Flag_D	Flag_M	Offset	Size_Data

- Overlap Buffer เป็นบัฟเฟอร์ที่เก็บข้อมูลเมื่อตรวจพบว่าเกิดการเหลื่อมล้ำของแพ็คเก็ต มีการเก็บในลักษณะของลิงค์ลิสต์
- Gap Frame Buffer คือ บัฟเฟอร์ที่เก็บข้อมูล เมื่อตรวจพบว่าเกิดการประกอบเฟรมไม่ได้ในลักษณะมีช่องว่างระหว่างแพ็คเก็ต มีการเก็บในลักษณะของลิงค์ลิสต์

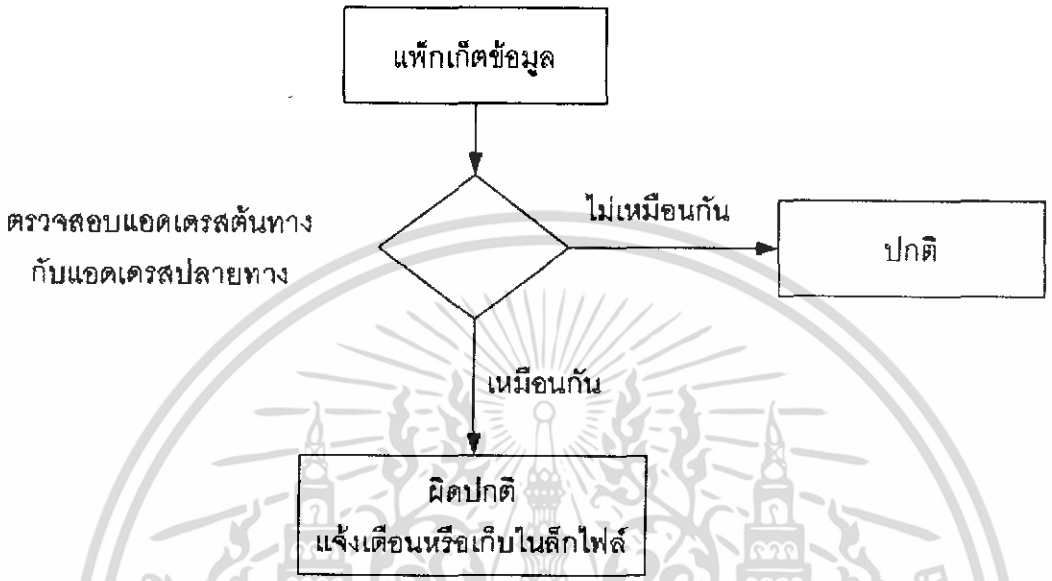
ในการวิเคราะห์จะใช้บัฟเฟอร์เหล่านี้ร่วมกันโดยเก็บข้อมูลแพ็คเก็ตที่เข้ามาทั้งหมดลงใน Fragment Buffer หากแพ็คเก็ตที่ส่งมาสามารถรวมกันได้ก็รวมกันเป็นแพ็คเก็ตเดียวที่ต่อเนื่องกัน แต่หากรวมกันแล้วเกิดความผิดปกติ ให้แจ้งมายัง Overlap Buffer หรือ Gap Frame Buffer แล้วแต่ความผิดปกติที่เกิดขึ้น หากไม่มีความผิดปกติเกิดขึ้นเมื่อครบ 3 วินาที โปรแกรมจะตรวจสอบ Fragment Buffer ว่าหากมีแพ็คเก็ตใดยังไม่ได้ประกอบหรือประกอบไม่ครบให้เก็บไว้ใน Overlap Buffer หรือ Gap Frame Buffer และเมื่อครบ 3 วินาที ข้อมูลใน Overlap Buffer หรือ Gap Frame Buffe นี้จะออกมาที่หน้าจอเพื่อแจ้งให้ผู้ดูแลระบบทราบหรือเก็บไว้ในล็อกไฟล์เพื่อบันทึกความผิดปกติที่เกิดขึ้น หากไม่มีความผิดปกติใดๆเกิดขึ้นและแพ็คเก็ตเหล่านั้นสามารถประกอบเป็นเฟรมได้อย่างถูกต้อง ให้ลบเฟรมเหล่านั้นออกจากบัฟเฟอร์ทันที เพื่อไม่ให้สิ้นเปลืองเนื้อที่ในการจัดเก็บ



รูปที่ 2.8 แสดงการตรวจสอบความผิดปกติในการทำแฟร็กเมนเตชัน

2.2.2 การส่งแพ็คเก็ตแบบวนลูป

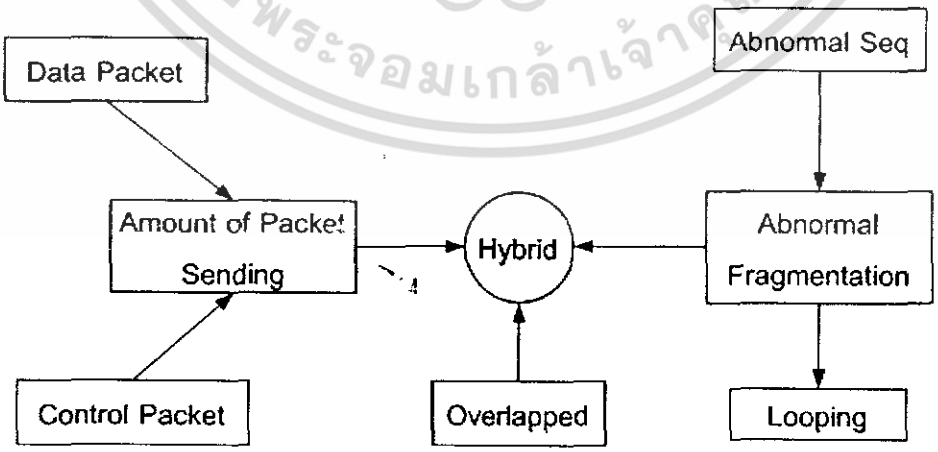
การส่งแพ็คเก็ตแบบนี้สามารถทำได้โดยการเปรียบเทียบค่าไอพีแอดเดรสต้นทาง และ ไอพีแอดเดรสปลายทางของเฮดเดอร์แพ็คเก็ต หากค่าไอพีต้นทางและไอพีปลายทางเป็นค่าเดียวกัน แสดงว่ามีความผิดปกติเกิดขึ้นเพราะทำให้เกิดการส่งในลักษณะวนลูป ซึ่งมีขั้นตอนดังรูปที่ 2.9



รูปที่ 2.9 แสดงการตรวจสอบแพ็คเก็ตที่ส่งแบบวนลูป

2.3 การโจมตีแบบผสม (Hybrid)

การวิเคราะห์แพ็คเก็ตประเภทนี้ให้นำวิธีการวิเคราะห์ที่กล่าวข้างต้นมาใช้ร่วมกัน เนื่องจากเกิดวิธีการที่ผสมผสานกันระหว่างวิธีต่าง ๆ ที่ได้กล่าวมาแล้ว ซึ่งสามารถแยกวิเคราะห์ออกเป็นแต่ละแบบหรือ วิเคราะห์รวมกันก็ได้ดังขั้นตอนในรูที่ 2.10



รูปที่ 2.10 แสดงแผนภูมิแสดงประเภทของการโจมตีเพื่อให้ปิดบริการแบบผสม

2.5.2 ระบบตรวจจับผู้บุกรุกที่ใช้งานในไฟร์เบรก

สำหรับไฟร์เบรกซึ่งเป็นรีเวิร์สเว็บพร็อกซี่นั้นจะมีการใช้งานของระบบตรวจจับผู้บุกรุกเพื่อตรวจสอบในระดับของแอพพลิเคชั่นเลเยอร์ที่เป็นการติดต่อกับเว็บเซิร์ฟเวอร์ที่อยู่ข้างหลัง รีเวิร์สเว็บพร็อกซี่สำหรับในส่วนนี้ได้มีการใช้โปรแกรม Snort_inline ซึ่งเป็นโปรแกรมระบบตรวจจับและป้องกันผู้บุกรุก (Intrusion Prevention System IPS) คือโปรแกรมระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ที่สามารถป้องกันการโจมตีได้ทันทีที่ตรวจพบ เป็นการเพิ่มความสามารถในการดร็อปแพ็คเก็ตให้กับโปรแกรมระบบตรวจจับผู้บุกรุกเดิมนั้นเอง

การติดตั้ง การตั้งค่าการใช้งาน การทดสอบการใช้งานจะกล่าวถึงในบทของการออกแบบการพัฒนา และการทดสอบการใช้งานต่อไป

บทที่ 3
การออกแบบโครงสร้างและการพัฒนาระบบ

3.1 แนวคิด

ในระบบเครือข่ายทั่วไปนั้นจะมีการแบ่งโซนในการใช้งานออกเป็น 3 โซนคือ เครือข่ายภายใน เครือข่ายภายนอก และ เครือข่าย DMZ เพื่อใช้ในการแบ่งแยกลักษณะของโฮสต์และความน่าเชื่อถือของโฮสต์นั้นๆ ซึ่งในแต่ละโซนก็จะมีการใช้งานไฟร์วอลในแต่ละประเภทที่แตกต่างกันเพื่อความเหมาะสมในการใช้งานสำหรับโฮสต์แต่ละประเภท เช่น เครือข่ายภายในโฮสต์ที่ทำงานอยู่ในส่วนนี้จะมีมากมายเพราะอาจเป็นเครื่องของพนักงานแต่ละคนทำให้ความต้องการในการใช้งานไฟร์วอลนั้นต่างกันไฟร์วอลที่ใช้งานจึงควรที่จะเป็นเพอร์ซันนอลไฟร์วอล ในส่วนของเกตเวย์หรือทางเชื่อมต่อระหว่างโซนนั้นก็ควรที่จะมีเกตเวย์ไฟร์วอลติดตั้งอยู่เพื่อทำการกรองแพ็กเก็ตที่ผ่านเข้าออกในแต่ละโซน และในส่วนของเครือข่าย DMZ ซึ่งเป็นที่อยู่ของเครื่องเว็บเซิร์ฟเวอร์ก็อาจจะมีการติดตั้งซีเคียวรีเวิร์สเว็บพร็อกซี่เพื่อใช้ในการป้องกันการเข้าถึงหรือการโจมตีที่ไม่หวังดีต่อเครื่องเว็บเซิร์ฟเวอร์

จะเห็นได้ว่าในระบบเครือข่ายหนึ่งๆมีการใช้งานไฟร์วอลหลายประเภทที่มีหน้าที่แตกต่างกันทำให้การจัดการกับกฎที่จะใช้กับไฟร์วอลในส่วนต่างๆนั้นทำได้ยาก เพราะต้องมีการคำนึงถึงกฎของไฟร์วอลในส่วนอื่นๆที่มีความเกี่ยวข้องกัน ซึ่งถ้ามีการติดตั้งไฟร์วอลในหลายตำแหน่งเช่น พวกเพอร์ซันไฟร์วอลก็จะต้องทำการกำหนดกฎใหม่ทุกครั้งที่ทำการติดตั้งใหม่ จึงได้มีความคิดที่จะเพิ่มความสามารถในการกำหนดกฎให้กับเพอร์ซันนอลไฟร์วอล และเกตเวย์ไฟร์วอล จากส่วนกลางเพียงจุดเดียว เพื่อความสะดวกในการควบคุมและจัดการกับระบบเครือข่าย

ในปัจจุบันนี้มีการนำระบบไดเรกทอรีเซอร์วิสมาใช้ในองค์กรมากยิ่งขึ้นเพราะมีความง่ายในการควบคุม จัดการ และเก็บข้อมูลของผู้ใช้ อีกทั้งยังมีการพิสูจน์ตน (Authentication) ก่อนที่จะเข้าใช้งานทรัพยากรของระบบ และมีมาตรการ การรักษาความปลอดภัยข้อมูลของระบบที่มีประสิทธิภาพ จึงได้นำคุณสมบัติของ แอ็คทีฟไดเร็คทอรี่ในวินโดวส์เซิร์ฟเวอร์ 2003 มาเพิ่มความสามารถเพื่อใช้ในการควบคุมกฎของไฟร์วอล และ ใช้ในการเก็บล็อกลงในฐานข้อมูล จากเครื่องลูกข่ายที่อยู่ภายในโดเมนคอนโทรเลอร์เดียวกัน ทำให้สามารถควบคุมการทำงานของไฟร์วอลได้ที่จุดศูนย์กลางเพียงจุดเดียว ซึ่งหากมีการโจมตีเกิดขึ้นนั้นสามารถเก็บผลจากการโจมตีได้เพื่อรวบรวมเข้าด้วยกันให้สามารถนำมาวิเคราะห์กฎที่ จะเพิ่มเข้าไปได้อย่างเหมาะสมทำให้ระบบมีความปลอดภัยยิ่งขึ้น

จากที่ได้กล่าวมาข้างต้นทำให้เกิดแนวความคิดสร้างชุดโปรแกรมเน็ตเวิร์กไฟร์วอลที่มีความสามารถดังนี้

- เพอร์ซันนอลไฟร์วอลที่สามารถป้องกันการโจมตีได้ตามกฎที่ได้รับจากส่วนกลาง พร้อมทั้งมีระบบตรวจจับผู้บุกรุกทำงาน และสามารถจัดเก็บล็อกไฟล์ที่ผิดปกติไว้ที่ฐานข้อมูลของเครื่องส่วนกลางได้
- เกตเวย์ไฟร์วอลที่สามารถกรองแพ็กเก็ตที่ผ่านเข้าออกเครือข่ายได้ตามกฎที่ได้รับจากส่วนกลาง และสามารถจัดเก็บล็อกไฟล์ที่ผิดปกติไว้ที่ฐานข้อมูลของเครื่องส่วนกลางได้
- รีเวิร์สเว็บพร็อกซี่ที่สามารถป้องกันการโจมตีและการเข้าถึงเว็บเซิร์ฟเวอร์ได้โดยตรง และสามารถจัดเก็บล็อกไฟล์ที่ผิดปกติไว้ที่ฐานข้อมูลส่วนกลางได้เช่นเดียวกัน

จากแนวคิดนี้ทำให้สามารถแบ่งโปรแกรมการทำงานออกเป็น 5 ส่วนดังนี้คือ

1. **ไฟร์สเตชั่น (FireStation)**

ออกแบบให้ทำหน้าที่ในการกำหนดกฎของเกตเวย์ไฟร์วอลและเพอร์ซันนอล ไฟร์วอล ซึ่งกฎของเพอร์ซันนอลไฟร์วอลสามารถที่จะกำหนดกฎให้กับกลุ่มผู้ใช้และผู้ใช้แต่ละคนได้

2. **ไฟร์อลาร์ม (FireAlarm)**

ออกแบบให้เป็นเพอร์ซันนอลไฟร์วอลทำหน้าที่ป้องกันการโจมตีได้ตามกฎที่รับมาจากไฟร์สเตชั่นได้อย่างถูกต้องและเมื่อมีการบุกรุกจะส่งข้อมูลการบุกรุกกลับไปที่เครื่องแม่ข่าย

3. **ไฟร์สกรีน (FireScreen)**

ออกแบบให้เป็นเกตเวย์ไฟร์วอลทำหน้าที่ป้องกันการโจมตีได้ตามกฎที่รับมาจากเครื่องไฟร์สเตชั่นได้อย่างถูกต้องและเมื่อมีการบุกรุกจะส่งข้อมูลการบุกรุกกลับไปที่เครื่องแม่ข่าย

4. **ไฟร์เบรก (FireBreak)**

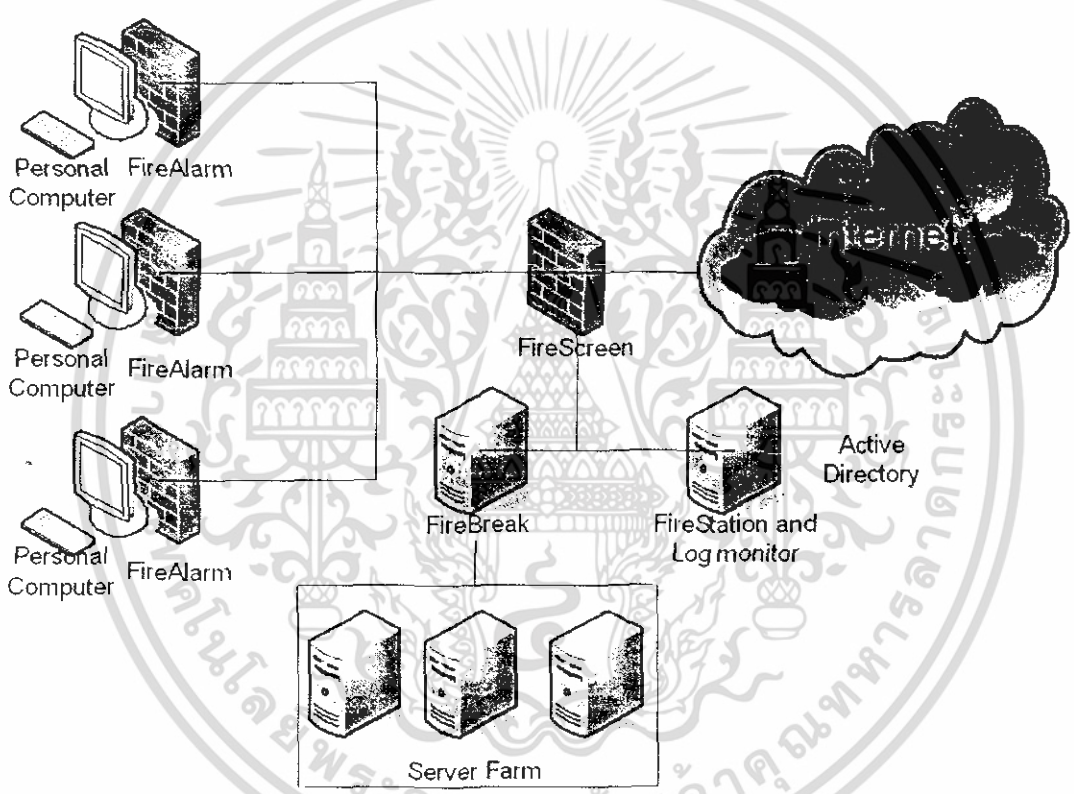
ออกแบบให้เป็นโปรแกรมซีเคียวรีเวิร์สเว็บพร็อกซี่สำหรับเพิ่มความปลอดภัยให้กับเว็บเซิร์ฟเวอร์ซึ่งอยู่ภายในโซน DMZ มีความสามารถตรวจจับผู้บุกรุกในระดับแอพพลิเคชั่น และเมื่อมีการบุกรุกจะส่งข้อมูลการบุกรุกกลับไปที่เครื่องแม่ข่าย

5. **ล็อกมอนิเตอร์ (LogMonitor)**

ออกแบบให้รับรายละเอียดการโจมตีจากเครื่องลูกข่ายที่อยู่ในระบบ และจัดการเกี่ยวกับการนำล็อกไฟล์จากฐานข้อมูลขึ้นมาแสดงผล

3.2 โครงสร้างระบบ

จากแนวคิดข้างต้นสามารถนำชุดโปรแกรมเน็ตเวิร์กไฟร์วอลนี้มาใช้กับระบบเครือข่ายได้ ดังรูปที่ 3.1 คือโปรแกรมไฟร์สเตชั่น, ไฟร์เบรก และล็อกมอนิเตอร์จะถูกวางอยู่ในโซน DMZ โปรแกรมไฟร์อลาร์มจะถูกติดตั้งอยู่ในโซนของเครือข่ายภายในที่เครื่องลูกข่ายต่างๆ ที่อยู่ในโดเมน คอนโทรเลอร์ของเครื่องไฟร์สเตชั่น และโปรแกรมไฟร์สกรีนจะถูกติดตั้งอยู่ที่เครื่องที่ทำหน้าที่เป็น เกตเวย์ของระบบเครือข่าย



รูปที่ 3.1 ระบบเครือข่ายที่มีการติดตั้งชุดโปรแกรมเน็ตเวิร์กไฟร์วอล

บทที่ 4
การพัฒนาโปรแกรมไฟร์สเตชั่น

4.1 แนวคิด

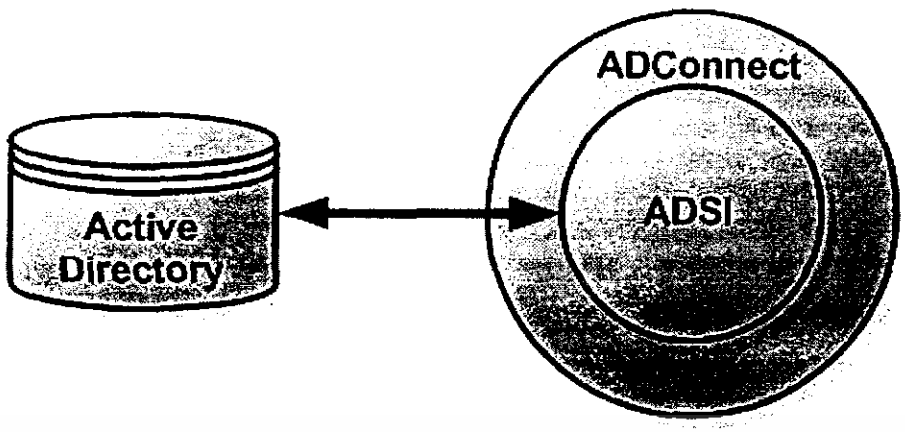
เป็นโปรแกรมที่ใช้ในการกำหนดกฎของเพอร์ซันนอลไฟร์วอล และเกตเวย์ไฟร์วอล โดยใช้เทคโนโลยีของแอ็คทีฟไดเร็กทอรีเซอร์วิส (Active Directory Service) ในการจัดเก็บกฎของเพอร์ซันนอลไฟร์วอลแต่ละเครื่องและกฎของเกตเวย์ไฟร์วอล เพื่อให้เครื่องลูกข่ายและเครื่องที่ทำหน้าที่เป็นเกตเวย์ไฟร์วอลที่มีสิทธิในการใช้งานสามารถเข้าถึงแอ็คทีฟไดเรกทอรีเพื่อนำกฎไปใช้ในการป้องกันการบุกรุกได้ โดยข้อมูลกฎจะเก็บอยู่ในฐานข้อมูลของแอ็คทีฟไดเร็กทอรี ทำให้การวางตำแหน่งของเครื่องที่ควรจะติดตั้งโปรแกรมนี้คือ เครื่องที่ทำงานเป็นเครื่องแม่ข่ายหรือเครื่องเซิร์ฟเวอร์ที่เปิดใช้งานโดเมนคอนโทรลเลอร์ (Domain Controller)

4.2 ขอบเขตและความสามารถ

- สามารถพิสูจน์ผู้ใช้ที่มาขอใช้ทรัพยากรได้อย่างถูกต้องโดยใช้ความสามารถของแอ็คทีฟไดเร็คทอรี
- สามารถกระจายกฎและจัดเก็บได้อย่างมีระเบียบ

4.3 การพัฒนาโปรแกรม

การติดต่อระหว่างไฟร์สเตชั่น ไฟร์อลาร์มและล็อกมอนิเตอร์ของไฟร์อลาร์มจะทำงานอยู่บนความสามารถของแอ็คทีฟไดเร็คทอรี (Active Directory) ของวินโดว์ 2003 เซิร์ฟเวอร์โดยข้อมูลจะเก็บอยู่ในฐานข้อมูลของแอ็คทีฟไดเร็คทอรี (Active Directory Database) และเข้าถึงข้อมูลผ่าน ADSI (Active Directory Service Interfaces) ไปยังแอ็คทีฟไดเรกทอรีเซอร์วิส (Active Directory Service) เพื่อเข้าถึงที่จัดเก็บ แสดงดังรูปที่ 4.1 ดังนั้นจึงต้องพัฒนาโปรแกรมไฟร์สเตชั่น ไฟร์อลาร์ม และล็อกมอนิเตอร์ของไฟร์อลาร์มไปพร้อมๆกันเพราะต้องมีการติดต่อกับแอ็คทีฟไดเรกทอรีในกรณีเดียวกัน



รูปที่ 4.1 แสดงรูปแบบการเข้าถึง Active Directory Service

การพัฒนาโปรแกรมไฟร์สเตชั่น ไฟร์อลาร์ม และล็อกมอนิเตอร์ของไฟร์อลาร์มนี้จะมีการสร้างสกีมา(Schema) ซึ่งเป็นโครงสร้างฐานข้อมูลของแอ็คทีฟไดเร็คทอรีประกอบไปด้วย คอนเทนเนอร์ คลาส และ แอตตริบิวต์ โดยการเข้าถึงข้อมูลเหล่านั้นจะมีการกำหนดหน้าที่การทำงานมาแล้ว ดังนั้นถ้าเราต้องการนำข้อมูลที่เราต้องการจัดเก็บหรือเข้าถึงรูปแบบใหม่เข้าไปจึงต้องมีการสร้างขึ้นมาใหม่ เพื่อให้เหมาะสมกับการทำงานของระบบ ไดเรกทรอรีเบสไฟร์วอล์

ข้อมูลที่จัดเก็บประกอบด้วย ข้อมูลของกฎ ข้อมูลรายละเอียดการโฆมตีและข้อมูลที่บอกการเปลี่ยนแปลงกฎโดยเพิ่มสกีมาที่เหมาะสมลงไปในฐานข้อมูลของแอ็คทีฟไดเร็คทอรีดังนี้

1. แอตตริบิวต์ รายละเอียดการเพิ่มจะเป็นดังตาราง ที่ 4.1

ตารางที่ 4.1 รายละเอียดของแอตตริบิวต์ที่เพิ่มในฐานข้อมูลของแอ็คทีฟไดเร็คทอรี

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 รายละเอียดคลาสที่เพิ่มในฐานข้อมูลของแอ็คทีฟไดเร็คทอรี

Common Name	OID	Syntax
ISAGFW	1.2.840.113556.1.4.7000.143	Auxiliary

โดยกำหนดให้แอตตริบิวต์ firewallRule เป็น Mandatory และ FirewallLog จะเป็น Auxiliary ใน คลาส ISAGFW

3. ผนวกสกีมาใหม่ เข้ากับ คลาส groups คลาส users และคลาส computers

ในการเข้าถึงข้อมูลที่อยู่ในยังแอ็คทีฟไดเรกทอรีเซอร์วิส นั้นต้องใช้ ADSI ดังนั้นจึงมีการพัฒนา function ที่ใช้เข้าถึงและกำหนดการทำงานกับแอ็คทีฟไดเรกทอรีได้อย่างง่ายขึ้น โดยการสร้างคลาส ADConnect ที่มีรายละเอียด Function ดังรูปที่ 4.2 โดยคลาส ADConnect จะใช้ในโปรแกรมไฟร์สเตชั่น ไฟร์อลาร์ม และล็อกมอนิเตอร์ของไฟร์อลาร์มที่พัฒนาขึ้นเพื่อติดต่อไปยังแอ็คทีฟไดเรกทอรีเซอร์วิส

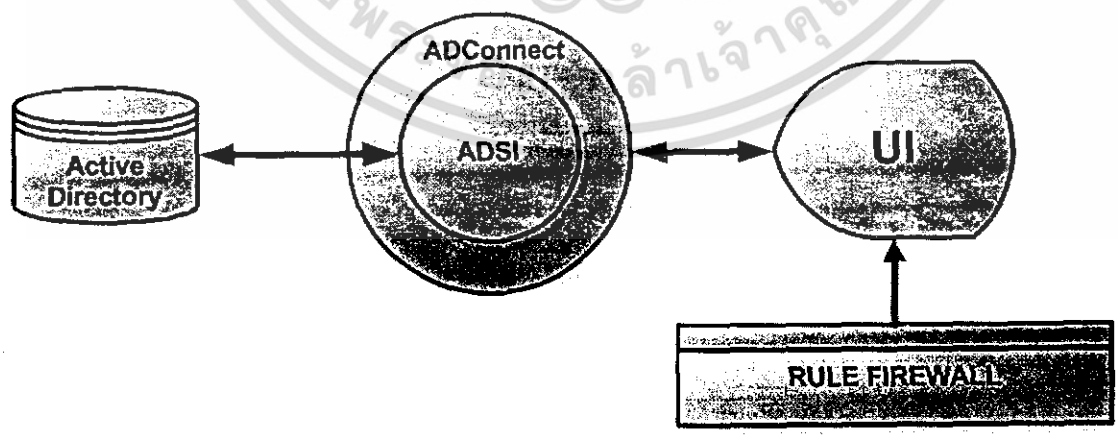
รูปที่ 4.2 รายละเอียดของคลาส ADConnect

ตารางที่ 4.3 รายละเอียดการทำงานของฟังก์ชั่นในคลาส ADConnect

Function	การทำงาน
GetDomain	หาชื่อเครื่องโดเมนที่ผู้ใช้สังกัดอยู่
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

เมื่อเปิดโปรแกรมไฟร์สเตชั่น(FireStation) จะทำการอ่านข้อมูลผ่านคลาส ADConnect เพื่อเข้าถึงข้อมูลที่อยู่ในกับแอ็คทีฟไดเร็กทอรีเซอร์วิส (Active Directory Service) โดยจะเลือกแสดงตามกลุ่มผู้ใช้ที่ถูกเลือก และเมื่อมีการ เพิ่ม ลด และแก้ไข ข้อมูลกฎที่อยู่ในฐานข้อมูล ดังรูปที่ 3.4 จะมีส่วนแสดงการทำงานหลักของโปรแกรมไฟร์สเตชั่น(FireStation)



รูปที่ 4.3 แสดงโครงสร้างการทำงานของไฟร์สเตชั่น(FireStation)

บทที่ 5
การพัฒนาโปรแกรมไฟร์อลาร์ม

5.1 แนวคิด

โปรแกรมไฟร์อลาร์มเป็นโปรแกรมเพอร์ซันนอลไฟร์วอล ที่มีระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ที่พัฒนาขึ้นเองโดยภาษา C++ เพื่อใช้งานร่วมกับไฟร์สเตชั่นในการรับกฎของเพอร์ซันนอลไฟร์วอลสำหรับแต่ละเครื่องลูกข่าย โดยเป็นการใช้งานไฟร์วอลแบบอิงผู้ใช้คือเมื่อผู้ใช้มีการล็อกอินใช้งานเครื่องที่อยู่ในระบบเครือข่ายภายในเครื่องใดก็ตาม ก็จะมีกฎสำหรับผู้ใช้คนนั้นในการใช้งานเพอร์ซันนอลไฟร์วอลเสมอ

5.2 ขอบเขตและความสามารถ

ในระบบจะมีเครื่องลูกข่ายหลายเครื่องโดยทุกเครื่องจะมีการติดตั้ง เพอร์ซันนอลไฟร์วอล ซึ่งก็คือเอเจนต์ที่ฝังตัวทำงานอยู่แบบอัตโนมัติ โดยจะมีการควบคุมดูแลจากส่วนกลางโดยผู้ใช้จะไม่รับรู้ถึงการทำงานของเอเจนต์ รวมไปถึงผู้ดูแลระบบสามารถกำหนดกฎการป้องการบุกรุกให้ผู้ใช้ในแต่ละกลุ่มตามความเหมาะสม

อีกทั้งยังมีส่วนของการทำงานที่เป็นระบบตรวจจับผู้บุกรุกที่คอยทำหน้าที่ตรวจสอบว่ามีการบุกรุกเข้ามาที่เครื่องลูกข่ายนั้นๆหรือไม่ ถ้ามีจะมีการแจ้งเตือนไปยังเครื่องเซิร์ฟเวอร์กลางเพื่อนำผลที่ได้ไปวิเคราะห์หาแนวทางป้องกันต่อไป และส่วนตรวจการเชื่อมต่อเครือข่ายของโปรเซสต่างๆเพื่อให้ผู้ใช้งานได้รับรู้ว่ามีโปรเซสอะไรที่เชื่อมต่อกับเครือข่ายอยู่บ้าง และสามารถกำจัดโปรเซสที่น่าสงสัยได้ด้วยตนเอง ทำให้ผู้ใช้สามารถป้องกันตนเองได้ในระดับหนึ่ง

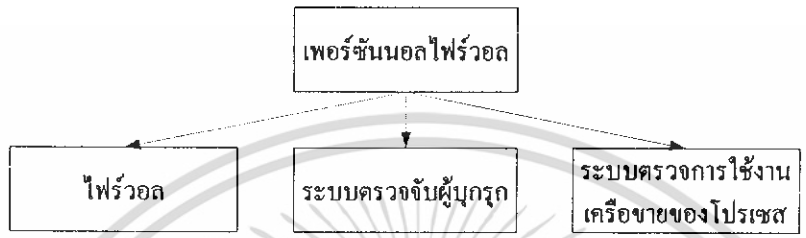
5.3 การพัฒนาโปรแกรม

การพัฒนาโปรแกรมไฟร์อลาร์มจะใช้ Firewall-Hook Driver ซึ่งเป็น DDK (Driver Development Kit) ของทางบริษัทไมโครซอฟต์ ซึ่งไดร์เวอร์ตัวนี้ใช้ในการฟิลเตอร์แพ็คเก็ต การเขียนโปรแกรมนั้นจะใช้เฮดเดอร์(Header) ที่ชื่อว่า IpDrvFw.h โดยภายในเฮดเดอร์(Header) นี้จะมีโครงสร้างต่างๆที่ใช้ในการฟิลเตอร์ และใช้ WinPCap ช่วยในการดักจับแพ็คเก็ตเพื่อมาวิเคราะห์หาว่ามีการบุกรุกระบบเกิดขึ้นหรือไม่ โดยกฎที่ใช้ในการฟิลเตอร์นั้นจะรับมาจากฐานข้อมูลของแอ็คทีฟไดเร็คทอรี (Active Directory Database)

5.4 การทำงานของโปรแกรม

ไฟร์อลาร์มเป็นเพอร์ซันนอลไฟร์วอลซึ่งจะแบ่งการทำงานออกเป็น 3 ส่วน คือ

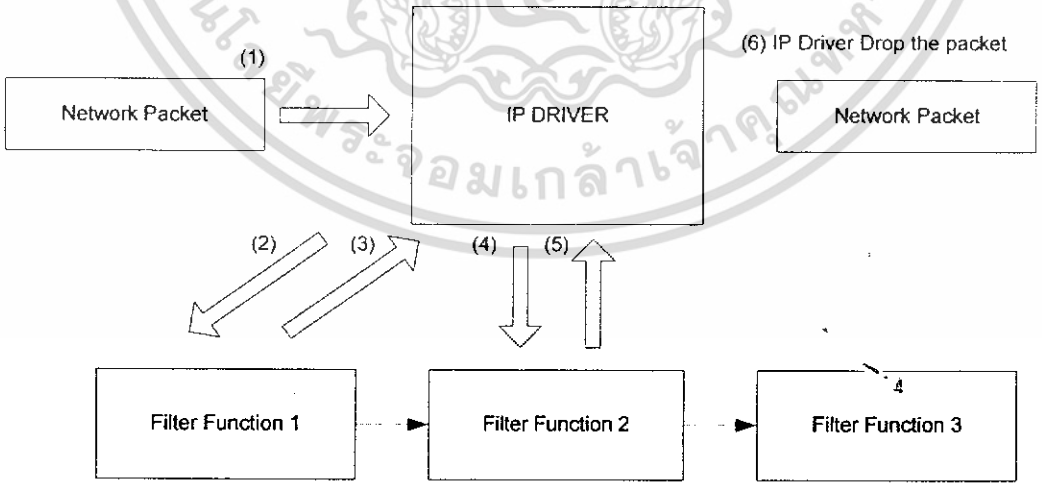
1. ไฟร์วอล
2. ระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์
3. ระบบตรวจการใช้งานเครือข่ายของโปรเซส



รูปที่ 5.1 แสดงส่วนประกอบเพอร์ซันนอลไฟร์วอล

5.3.1 แพ็คเก็ตฟิลเตอร์ริงไฟร์วอลบนวินโดวส์ XP

เพอร์ซันนอลไฟร์วอลที่พัฒนามีลักษณะการทำงานแบบแพ็คเก็ตฟิลเตอร์ริง(Packet Filtering) โดยจะมีการตรวจสอบแพ็คเก็ตด้วยฟังก์ชันฟิลเตอร์ว่าจะอนุญาตหรือไม่อนุญาตให้แพ็คเก็ตนั้นผ่านไปได้ โดยที่ฟังก์ชันฟิลเตอร์จะมีการกำหนดลำดับไว้แล้วให้ระบบเรียกฟังก์ชันฟิลเตอร์เข้าไปทำงานทีละฟังก์ชันตามลำดับ จนกว่าจะมีฟังก์ชันใดฟังก์ชันหนึ่งส่งค่ากลับเป็น "DROP PACKET" ถ้าฟังก์ชันทั้งหมดส่งค่ากลับเป็น "ALLOW PACKET" แพ็คเกจเหล่านั้นจะสามารถผ่านไปได้



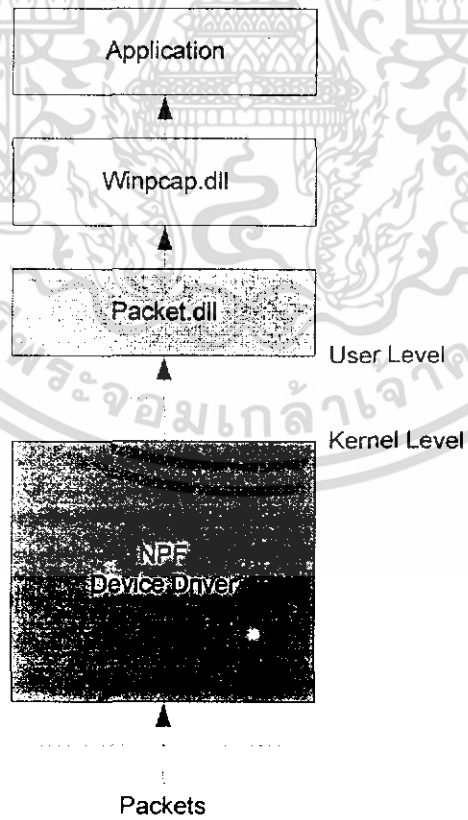
รูปที่ 5.2 แสดงขั้นตอนการทำงานของไฟร์วอล

ขั้นตอนการทำงานของไฟร์วอลตามรูปที่ 5.2

1) เมื่อเครื่องได้รับแพ็กเกตเข้ามา โดยที่ IP Driver มีฟังก์ชันฟิลเตอร์อยู่ตามที่กำหนดไว้
2) IP Driver จะส่งแพ็คเก็ตนั้นเข้าไปโดย ผ่านเข้าไปยังฟังก์ชันฟิลเตอร์ตามลำดับไปเรื่อยๆ โดยรอค่าที่ส่งกลับออกมา
3) สมมุติฟังก์ชันแรกส่งค่ากลับเป็น "ALLOW PACKET" เมื่อ IP DRIVER ได้รับค่าส่งกลับจากฟังก์ชันแรกเป็น "ALLOW PACKET"
4) ดังนั้น IP DRIVER จะส่งแพ็คเก็ตนั้นไปที่ฟังก์ชันที่สองต่อไปในกรณีนี้
5) สมมุติให้ฟังก์ชันที่สองนี้ส่งค่ากลับเป็น "DROP PACKET" เมื่อ IP DRIVER ได้รับค่าส่งกลับจากฟังก์ชันที่สองเป็น "DROP PACKET" ดังนั้น IP DRIVER จะไม่ส่งแพ็กเกจนี้ต่อไปยังระบบ และจะไม่ส่งไปยังฟังก์ชันต่อไปอีก

5.3.2 ระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

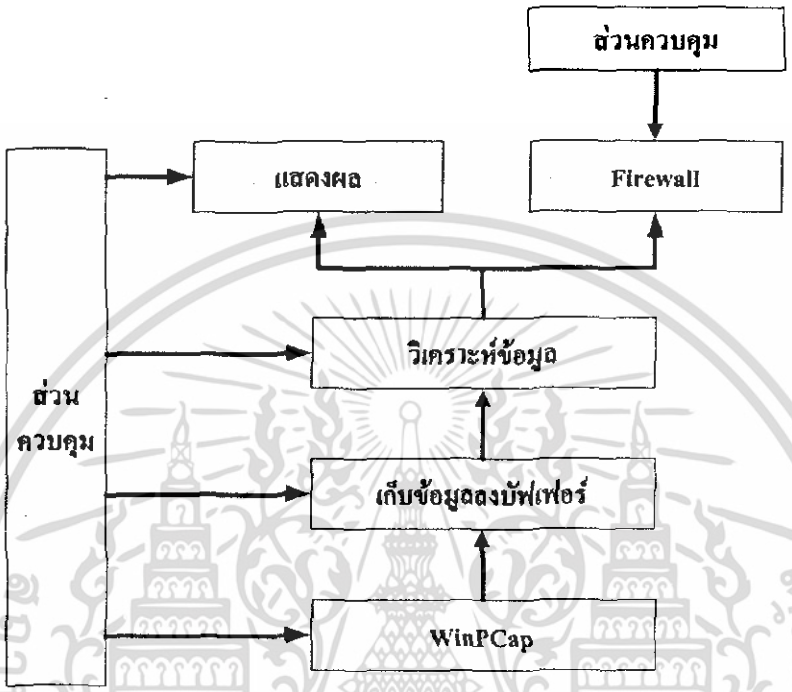
ระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์จะอาศัยคุณสมบัติ และ ความสามารถของ WinPCap ซึ่งเป็นไลบรารีที่ทำการติดต่อกับการ์ดแลน เพื่อทำการควบคุมการทำงานของ การ์ดแลน และตรวจจับแพ็คเก็ต เพื่อนำมาวิเคราะห์ผลว่ามีการโจมตีเกิดขึ้นหรือไม่



รูปที่ 5.3 แสดงระดับชั้นการทำงานของ WinPCap

การทำงานของระบบตรวจจับผู้บุกรุกนั้นจะแบ่งการทำงานออกเป็น 2 ส่วนหลักๆด้วยกัน คือ ส่วนการดักจับแพ็คเก็ตเก็บลงบัฟเฟอร์ และส่วนวิเคราะห์ผลแพ็คเก็ตที่ได้ดักจับมา เมื่อวิเคราะห์ผลเสร็จก็จะแสดงผล พร้อมกับส่งแพ็คเก็ตไปฟิลเตอร์ตามกฎที่กำหนดเอาไว้ดังรูปที่ 5.3

โดยการทำงานทั้ง 2 ส่วนนี้จะมีการแบ่งเธรด (Thread) การทำงานเพื่อที่จะสามารถดักจับแพ็คเก็ตและวิเคราะห์ผลไปได้พร้อมๆกัน



รูปที่ 5.4 โครงสร้างของระบบตรวจจับผู้บุกรุกทำงานร่วมกับไฟร์วอล

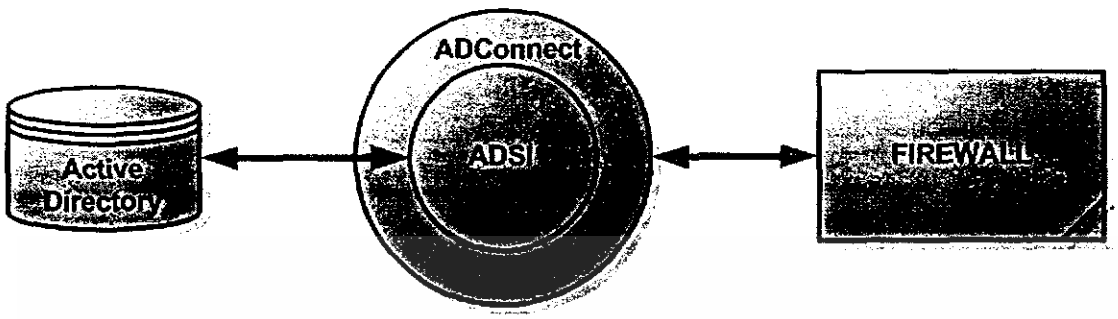
5.3.2 ระบบตรวจการใช้งานเครือข่ายของโปรเซส

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 เพื่อทำให้ผู้ใช้สามารถป้องกันตนเองได้อีกทางหนึ่ง

เนื่องจากเพอร์ซันนอลไฟร์วอล (Personal Firewall) จะมีความสามารถในการที่จะรับกฎเพื่อเริ่มการทำงานของไฟร์วอลจากเครื่องโดเมนคอนโทลเลอร์ที่ได้ติดตั้งแอ็คทีฟไดเร็กทอรีไว้และจะมีการส่งล็อกไฟร์ไปเก็บไว้ยังแอ็คทีฟไดเร็กทอรีเมื่อมีการตรวจพบการโจมตี หรือ มีสิ่งผิดปกติ

เกิดขึ้น ซึ่งโครงสร้างการทำงานของส่วนที่ใช้ในการติดต่อกับเครื่องเซิร์ฟเวอร์ แสดงดังรูปที่ 6-6 โดย เพอร์ซันนอลไฟร์วอลจะมีการติดต่อ ไปยังเซิร์ฟเวอร์โดยผ่านคลาส ADConnect ที่สร้างขึ้น



รูปที่ 5.5 แสดงโครงสร้างส่วนติดต่อกับเซิร์ฟเวอร์

เพอร์ซันนอลไฟร์วอลจะติดตั้งอยู่ที่เครื่องลูกข่ายทุกเครื่องในระบบโดยมีหน้าที่การทำงานดังนี้

- เป็นไฟร์วอลชนิดแพ็คเก็ตฟิลเตอร์ริ่ง (Packet Filtering Firewall)
- ตรวจจับผู้บุกรุกทางเครือข่ายที่เข้ามาที่เครื่องลูกข่าย รวมทั้งสิ่งผิดปกติที่ไม่เป็นไปตามมาตรฐานของ โปรโตคอลทีซีพี (TCP) , ยูดีพี(UDP), ไอพี(IP), ไอซีเอ็มพี(ICMP)
- ตรวจการเชื่อมต่อเครือข่ายของโปรเซสต่างๆโดยผู้ใช้สามารถจบการทำงานของโปรเซสและปิดการเชื่อมต่อเครือข่ายของโปรเซสใดๆได้
- ส่งการแจ้งเตือนกลับไปยังล็อกมอนิเตอร์เมื่อตรวจจับได้ว่ามีการบุกรุกเกิดขึ้น

บทที่ 6
การพัฒนาโปรแกรมไฟร์สกรีน

6.1 แนวคิด

โปรแกรมไฟร์สกรีนเป็นส่วนของไฟร์วอลที่อยู่บนเกตเวย์ ซึ่งใช้งานระบบปฏิบัติการลินุกซ์ (Gnu/Linux) โดยสำหรับการเข้าใช้บริการของแอคทีฟไดเรกทอรีบนไฟร์สเตชั่นนั้นจะใช้ OpenLDAP ซึ่งเป็นมาตรฐานเปิดสำหรับใช้งานโปรโตคอล LDAP ซึ่งสามารถใช้งานได้ดีกับทุกแพลตฟอร์ม และในส่วนของการกรองการเชื่อมต่อซึ่งนับเป็นส่วนสำคัญของไฟร์วอลแล้วจะใช้ความสามารถของเน็ตฟิลเตอร์/ไอพีเทเบิลส์ (Netfilter/Iptables) ซึ่งทำงานอยู่ในระดับเคอร์แนลและติดตั้งมากับระบบปฏิบัติการลินุกซ์อยู่แล้วตั้งแต่เคอร์แนลเวอร์ชั่น 2.4.x ขึ้นมา

6.2 ขอบเขตและความสามารถ

ไฟร์สกรีนทำงานอยู่บนเกตเวย์ ซึ่งการเชื่อมต่อในทุกๆส่วนจะต้องผ่านทางเกตเวย์นี้ นั่นคือไฟร์สกรีนจะสามารถควบคุมการเชื่อมต่อเกือบทั้งหมดในระบบได้ โดยปกติแล้วเกตเวย์ไฟร์วอลจะมีลักษณะเป็นแบบ allow-based คือยอมให้แพ็คเก็ตผ่านไปได้ทั้งหมด เว้นแต่ส่วนที่ถูกระบุมาตรงกับกฎที่ตั้งไว้ว่าจะไม่ให้ผ่านไป ส่วนการจะตั้งกฎในการที่จะปฏิเสธการเชื่อมต่อหรือแพ็คเก็ตใดบ้างขึ้นขึ้นกับการตั้งค่าของผู้ดูแลระบบผ่านทางไฟร์สเตชั่น

6.3 การพัฒนาโปรแกรม

6.3.1 การทำงานร่วมกับไฟร์สเตชั่น

ไฟร์สกรีนมีระบบปฏิบัติการเป็นลินุกซ์ ซึ่งในที่นี้ใช้ดิสโทรเดเบียน (debian) โดยการทำงานร่วมกับแอคทีฟไดเรกทอรีของไมโครซอฟท์บนไฟร์สเตชั่นนั้นจะใช้แพ็กเกจของ OpenLDAP โดยตัวโปรแกรมไฟร์สกรีนจะต้องใช้ไลบรารีของ OpenLDAP ในการเขียนโปรแกรมทำงานร่วมกับแอคทีฟไดเรกทอรี บนระบบปฏิบัติการเดเบียนนั้นใช้แพ็กเกจชื่อ libldap2-dev ซึ่งสนับสนุนโปรโตคอล LDAP เวอร์ชั่น 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 แสดงแพ็กเกจที่จำเป็น

6.3.2 การติดต่อกับแอคทีฟไดเรกทอรี

LDAP *ldap_init(host, port)

char *host;

int port;

ตารางที่ 6.1 พารามิเตอร์ของฟังก์ชั่น ldap_init()

host	ชื่อโฮสต์หรือไอพีแอดเดรสของเซิร์ฟเวอร์ (สามารถระบุหลายค่าได้โดยคั่นด้วยช่องว่าง) และสามารถระบุพอร์ตได้ดังนี้ *hostname:portnumber* ซึ่งจะถือว่าหมายเลขพอร์ตนี้มีความสำคัญกว่า *port* ซึ่งเป็นพารามิเตอร์ตัวถัดไป
port	หมายเลขพอร์ตที่ใช้ (ค่าปกติคือ 389 หรืออาจใช้ LDAP_PORT แทน)

หลังจากใช้ฟังก์ชั่น ldap_init() แล้วการเชื่อมต่อระหว่างไคลเอนต์และเซิร์ฟเวอร์ยังไม่ได้เริ่มขึ้นจริงๆ เพียงคืนค่ากลับมาเป็นแฮนเดิลของการเชื่อมต่อนั้นๆเก็บไว้ในตัวชี้ไปยัง LDAP structure หากไม่มีข้อผิดพลาดเกิดขึ้นเท่านั้น ส่วนการเชื่อมต่อจริงๆจะเริ่มต้นขึ้นในขั้นตอนต่อไป และหากเกิดข้อผิดพลาดขึ้นจะคืนค่ามาเป็น NULL

int ldap_bind_s(LDAP *ld, const char *who, const char *cred, int method);

ตารางที่ 6.2 พารามิเตอร์ของฟังก์ชั่น ldap_bind_s()

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

หลังจาก ldap_init() แล้วก่อนที่จะกระทำการใดๆกับเซิร์ฟเวอร์ได้จะต้องทำการเริ่มการเชื่อมต่อและพิสูจน์ตนกับเซิร์ฟเวอร์เสียก่อน โดยผ่านทางฟังก์ชั่น ldap_bind_s() ซึ่งจะคืนค่า 1 หากเกิดความผิดพลาดขึ้น

_s ที่ต่อท้ายชื่อฟังก์ชั่นจะหมายถึงเป็นแบบ synchronous คือต้องรอให้ฟังก์ชั่นทำงานเรียบร้อยได้รับการตอบรับจากเซิร์ฟเวอร์จึงจะสามารถทำคำสั่งต่อไปได้

int ldap_unbind(LDAP *ld);

int ldap_unbind_s(LDAP *ld);

ldap_unbind และ ldap_unbind_s ใช้สำหรับจบการเชื่อมต่อกับเซิร์ฟเวอร์ และจะลบค่าใน ld ทั้งหมด โดยทั้ง 2 ฟังก์ชั่นไม่ได้มีความแตกต่างกันแต่อย่างใด สามารถใช้ได้ไม่ว่าขณะสร้างการเชื่อมต่อได้ใช้ ldap_bind หรือ ldap_bind_s ก็ตาม

ตัวอย่างการเขียนโปรแกรมเชื่อมต่อกับแอคทีฟไดเรกทอรี

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

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

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

    .....

    /* ยกเลิกการเชื่อมต่อ */
    ldap_unbind_s(ld);

    .....
```

6.3.3 การค้นหาข้อมูลในแอคทีฟไดเรกทอรี

การสืบค้นข้อมูลที่ต้องการบนแอคทีฟไดเรกทอรีเซิร์ฟเวอร์มี 3 ขั้นตอนด้วยกันคือ

1. ได้แต่ละเอนทรี่จากผลลัพธ์การค้นหา
2. ได้แอตทริบิวต์ที่ต้องการจากแต่ละเอนทรี่
3. ได้ค่าที่เก็บไว้ในแต่ละแอตทริบิวต์

โดยมีฟังก์ชั่นหลักๆที่ใช้งานดังนี้

```
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 พารามิเตอร์ของฟังก์ชั่น ldap_search_s()

ld	ตัวชี้ไปยัง LDAP structure ที่ทำงานด้วย
base	DN ของเอนทรีที่จะเป็นจุดเริ่มต้นของการค้นหา
scope	ขอบเขตของการค้นหา • LDAP_SCOPE_BASE ค้นหาเอนทรี่ทั้งหมดบน base

	• LDAP_SCOPE_ONELEVEL ค้นหาเอนทรี่ทั้งหมดบน base และต่ำลงไป 1 ขั้น • LDAP_SCOPE_SUBTREE ค้นหาเอนทรี่ทั้งหมดบน base และไดเรกทอรีภายใต้ทั้งหมด
filter	ฟิลเตอร์ที่จะใช้กรองผลลัพธ์จากการค้นหา อยู่ในรูปแบบ attributetype=attributevalue
attrs	อาเรย์ของชนิดแอตทริบิวต์ที่จะส่งคืนมาสำหรับเอนทรี่ที่เข้าข่าย
attrsonly	ระบุว่าค่าในแอตทริบิวต์จะถูกส่งกลับมาด้วยกันหรือไม่ • 0 หมายความว่าทั้งแอตทริบิวต์และค่าของแอตทริบิวต์นั้นถูกส่งกลับ • 1 หมายความว่าเพียงแอตทริบิวต์เท่านั้นที่ถูกส่งกลับมา
res	ผลลัพธ์จากการค้นหา

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

ตารางที่ 6.4 พารามิเตอร์ของฟังก์ชั่น 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 พารามิเตอร์ของฟังก์ชั่น ldap_first_attribute()

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

char **ldap_get_values(ld, entry, attr)

LDAP *ld;

LDAPMessage *entry;

char *attr

ตารางที่ 6.6 พารามิเตอร์ของฟังก์ชั่น 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 จะคืนเอนทรี่ที่เข้าข่ายสำหรับขอบเขตการค้นหาและฟิลเตอร์ที่ระบุเป็นตัวชี้ในตัวแปร 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);
}
```

```
/* ลูป for สำหรับวนจนครบทุกๆเอนทรีในส่งกลับมาจาก ldap_search_s */
for (entry = ldap_first_entry(ld, msg); entry != NULL; entry = ldap_next_entry(ld,
entry)) {
        /* ลูป for สำหรับวนจนครบทุกแอตทริบิวต์ในแต่ละเอนทรี่ */
        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

การเปลี่ยนแปลงแก้ไขข้อมูลที่ต้องการในแอคทีฟไดเรกทอรีเซิร์ฟเวอร์นั้นสามารถทำได้ผ่านฟังชั่น ldap_modify_s()

```
int ldap_modify_s(ld, dn, mods)
LDAP *ld;
char *dn;
LDAPMod *mods[];
```

ตารางที่ 6.7 พารามิเตอร์ของฟังก์ชั่น ldap_modify_s()

ld	ตัวชี้ไปยัง LDAP structure ที่ทำงานด้วย
dn	DN ของเอนทรี่ที่จะทำการแก้ไขเปลี่ยนแปลง
mods	ตัวชี้ไปยังอาเรย์ของตัวชี้ไปยัง LDAPMod structure สำหรับระบุว่าจะทำการเปลี่ยนแปลงอย่างไร

ตัวอย่างการเปลี่ยนแปลงแก้ไขข้อมูลบนแอคทีฟไดเรกทอรี

```
include <ldap.h>

.....

LDAP *ld;
LDAPMod *list_of_attrs[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 };

.....

/* ระบุรูปแบบการเปลี่ยนแปลงแก้ไขในแต่ละแอตทริบิวต์ */
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;

/* เก็บแต่ละค่าที่จะเปลี่ยนแปลงแก้ไขในอาเรย์ */
list_of_attrs[0] = &attribute1;
list_of_attrs[1] = &attribute2;
```

```
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 การนำกฎจากไฟร์สเตชั่นมาใช้บนไฟร์สกรีน

6.3.5.1 การดึงกฎจากไฟร์สเตชั่น

สำหรับกฎของไฟร์วอลที่เก็บไว้ในแอคทีฟไดเรกทอรีบนไฟร์สเตชั่นนั้นถูกเก็บไว้ใต้ dn คือ "cn=firescreen, cn=computers, dc=hephaestus, dc=com" ภายในแอตทริบิวต์ชื่อ firewallRule ดังนั้นจึงสามารถเข้าถึงได้โดยใช้ฟังก์ชั่น ldap_search_s() กำหนดขอบเขตของการค้นหาเป็น "cn=firescreen, cn=computers, dc=hephaestus, dc=com" จากนั้นจึงดึงค่าออกจากแอตทริบิวต์ firewallRule จนครบทุกค่าและเก็บค่าไว้รองานใช้ในอาเรย์

จากรูปแบบการทำงานของไฟร์สกรีนที่กล่าวถึงในหัวข้อก่อนหน้า ไฟร์สกรีนจะคอยตรวจสอบที่ไฟร์สเตชั่นทุกระยะเวลาหนึ่ง เพื่อตรวจสอบดูว่ามีกฎถูกแก้ไขหรือไม่ ซึ่งหลังจากกฎถูกแก้ไขบนไฟร์สเตชั่นจะทำการอัพเดตค่าในแอตทริบิวต์ชื่อ update เป็น "YES" ดังนั้นก่อนหน้าที่จะรับกฎมาจากไฟร์สเตชั่นจึงต้องทำการตรวจสอบแอตทริบิวต์ update ก่อน หากมีค่าเป็น "NO" ก็จะจบการทำงานและรอเวลาวนกลับไปตรวจสอบอีกครั้ง และหากไฟร์สกรีนพบว่าค่าในแอตทริ

บิวต์ update เป็น "YES" ก็จะทำการดึงกฎมาจากแอตทริบิวต์ firewallRule จากนั้นจึงทำการแก้ไขค่าในแอตทริบิวต์กลับเป็น "NO" อีกครั้งผ่านทางฟังก์ชั่น ldap_modify_s()

6.3.5.2 การนำมาใช้งานกับไอพีเทเบิ้ลส์

กฎที่ดึงมาจากทางไฟร์สเตชั่นนั้นจะอยู่ในรูปแบบดังต่อไปนี้

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	ปลายทางของแพ็กเก็ตที่แมชท์กับกฎนี้

โดยก่อนที่จะสามารถนำมาใช้งานกับไฟร์สกรีนได้จะต้องแปลงกฎที่รับมาจากไฟร์สเตชั่นที่มีรูปแบบดังกล่าวให้อยู่ในรูปแบบของไอพีเทเบิ้ลส์เสียก่อน โดยผ่านฟังก์ชั่นแปลง (parser) ซึ่งจะแยกแต่ละส่วนออกจากกันโดยมีเครื่องหมายโคลอน (:) เป็นตัวแบ่ง โดยนำเอาแต่ละส่วนมาต่อเข้าด้วยกันตามรูปแบบคำสั่งของไอพีเทเบิ้ลส์โดยขึ้นต้นด้วย iptables -A (กฎจะถูกเพิ่มต่อท้ายเรื่อยๆ ตามลำดับ) ส่วนกฎจะถูกเพิ่มเข้าเชน (CHAIN) ใดของไอพีเทเบิ้ลส์จะพิจารณาจาก ZONE และ TARGET ที่ระบุเป็น DROP ทั้งหมดจะถูกระบุ TARGET ในคำสั่งไอพีเทเบิ้ลส์เป็น -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

กฎแรกจะทำการเก็บข้อมูลทุกแพ็คเก็ตที่ผ่านเข้ามาในเชนนี้ลงล็อกไฟล์ในอัตรา 1 แพ็คเก็ตต่อวินาที (limit: avg 1/sec burst 5)โดยทุกๆค่าที่ถูกเก็บลงล็อกไฟล์จะมีคำว่า "FIRESCREEN " (prefix `FIRESCREEN ') ขึ้นต้น เพื่อประโยชน์ในการคัดแยกล็อกในภายหลัง เนื่องจากกระบวนการเก็บเหตุการณ์ลงล็อกไฟล์นั้นถูกจัดการโดย syslogd ซึ่งไม่ได้ถูกจัดการโดยตัวไอพีเทเบิ้ลส์เองทำให้ในล็อกไฟล์จะเก็บเหตุการณ์ที่มีระดับของการล็อกเหมือนกันเอาไว้ โดยในเชน LOG-CHAIN นี้ระบุระดับของการล็อกเป็น info (LOG level info)

กฎที่สองจะทำการละทิ้งหมดทุกๆแพ็คเก็ต เนื่องด้วยแพ็คเก็ตที่ถูกส่งต่อมายังเชนนี้เพื่อเก็บข้อมูลลงล็อกไฟล์คือแพ็คเก็ตที่ถูกกำหนด TARGET จากไฟร์สเตชั่นไว้เป็น DROP นั่นเอง

ตัวอย่างการแปลงกฎจากไฟร์สเตชั่นมาเป็นคำสั่งของไอพีเทเบิ้ลส์

กฎจากไฟร์สเตชั่น
192.168.2.10:255.255.255.255:192.168.1.2:255.255.255.255:TCP:ANY:22:-:DMZ-in:DROP
คำสั่งไอพีเทเบิ้ลส์
iptables -A FORWARD -s 192.168.2.10 -d 192.168.1.2 -p tcp -dport 22 -j LOG-CHAIN

กฎจากไฟร์สเตชั่น
192.168.2.7:255.255.255.255:-:-:ANY:-:-:LOCAL-in:ACCEPT
คำสั่งไอพีเทเบิ้ลส์
iptables -A INPUT -s 192.168.2.7 -j ACCEPT

สุดท้ายเมื่อได้กฎจากไฟร์สเตชั่นอยู่ในรูปแบบคำสั่งของไอพีเทเบิ้ลส์ทั้งหมดแล้วก็จะทำการเขียนทั้งหมดลงบนไฟล์เชลล์สคริปต์ เพื่อทำการรันเพื่อใช้งานตามคำสั่งไอพีเทเบิ้ลส์ดังกล่าว รวมทั้งเก็บไฟล์นี้ไว้ใช้ครั้งต่อไปหากโปรแกรมถูกปิดหรือเครื่องไฟร์สกรีนเริ่มทำงานใหม่เพื่อที่จะสามารถคงกฎเดิมไว้ได้

6.3.6 การส่งล็อกกลับไปยังไฟร์สเตชั่น

ไฟร์สเตชั่นจะมีการเก็บล็อกไฟล์สำหรับแพ็คเก็ตที่ถูกละทิ้งไปตามกฎที่ได้กำหนดไว้เอาไว้เพื่อใช้ประกอบการพิจารณาป้องกันการบุกรุกที่เกิดขึ้นหรือความผิดปกติที่อาจเกิดขึ้น ซึ่งล็อกไฟล์จะถูกเก็บลงไฟล์บนเครื่องไฟร์สกรีนเองก่อน จากนั้นจึงค่อยส่งต่อไปเก็บยังศูนย์กลางที่ไฟร์สเตชั่นต่อไป

ล็อกไฟล์ในส่วนของไอพีเทเบิ้ลส์นั้นไม่ได้ถูกเก็บไว้โดยตัวไอพีเทเบิ้ลส์เองโดยตรง แต่ผ่านโปรแกรมเก็บล็อกมาตรฐานบนระบบปฏิบัติการยูนิกซ์อย่าง syslogd ล็อกไฟล์ในประเภทเดียวกันก็จะถูกเก็บไว้ในไฟล์เดียวกัน โดยปกตินั้นล็อกในส่วนของไอพีเทเบิ้ลส์จะถูก syslogd เก็บไว้ที่ /var/log/kern.log แต่สำหรับบนตัวไฟร์สกรีนเองนั้นได้แยกล็อกไฟล์ในส่วนนี้เก็บแยกออกไปต่างหากที่ /var/log/firescreen.log อีกส่วนด้วย ซึ่งแยกออกตามระดับการล็อก โดยได้กำหนดระดับของล็อกไว้ที่ info อย่างไรก็ตามล็อกในส่วนของเคอร์แนลที่มีระดับเป็น info เช่นกันก็จะยังถูกเก็บลงยังไฟล์ firescreen.log นี้ด้วยเช่นกัน ดังนั้นแล้วขณะทำการล็อกจะกำหนดคำนำหน้า (prefix) ในส่วนของล็อกจากไอพีเทเบิ้ลส์เป็น "FIRESCREEN " เพื่อความง่ายในการแยกล็อกของไอพีเทเบิ้ลส์ออกจากล็อกจากส่วนอื่นๆ

```
Jan 10 16:00:45 firescreen kernel: NET: Registered protocol family 10
Jan 10 16:00:45 firescreen kernel: Disabled Privacy Extensions on device c02cc9
60(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 จะพบว่าไม่เฉพาะเพียงแต่ล็อกในส่วนของไอพีเทเบิ้ลส์เท่านั้นที่ถูกเก็บไว้ในไฟล์นี้แต่จะมีล็อกจากเหตุการณ์อื่นๆที่อยู่ในระดับ info รวมอยู่ด้วยเช่นกัน ล็อกของไอพีเทเบิ้ลส์เองสามารถสังเกตได้จากคำนำหน้า "FIRESCREEN " หลังส่วนต้นที่แสดงเวลาและชื่อโฮสต์

โปรแกรมจะคอยอ่านล็อกไฟล์ดังกล่าวเป็นระยะและส่งกลับไปเก็บยังแอคทีฟไดเรกทอรีบนไฟร์สเตชั่น จากนั้นจะทำการลบข้อมูลทั้งไฟล์ทิ้งเพื่อตัดปัญหาที่จะส่งล็อกเดิมไปอีกครั้งหนึ่ง โดยภายใต้ dn "cn=firescreen, cn=computers, dc=hephaestus, dc=com" ในแอคทีฟไดเรกทอรีบนไฟร์สเตชั่นนั้นจะมีแอตทริบิวต์ชื่อ firewallLog อยู่สำหรับเก็บล็อกที่ส่งมาจากไฟร์สกรีน อนึ่ง การเพิ่มล็อกเข้าไปยังแอตทริบิวต์ดังกล่าวสามารถทำได้ผ่านฟังก์ชั่น ldap_modify_s() ดังกล่าวถึงในหัวข้อก่อนหน้า

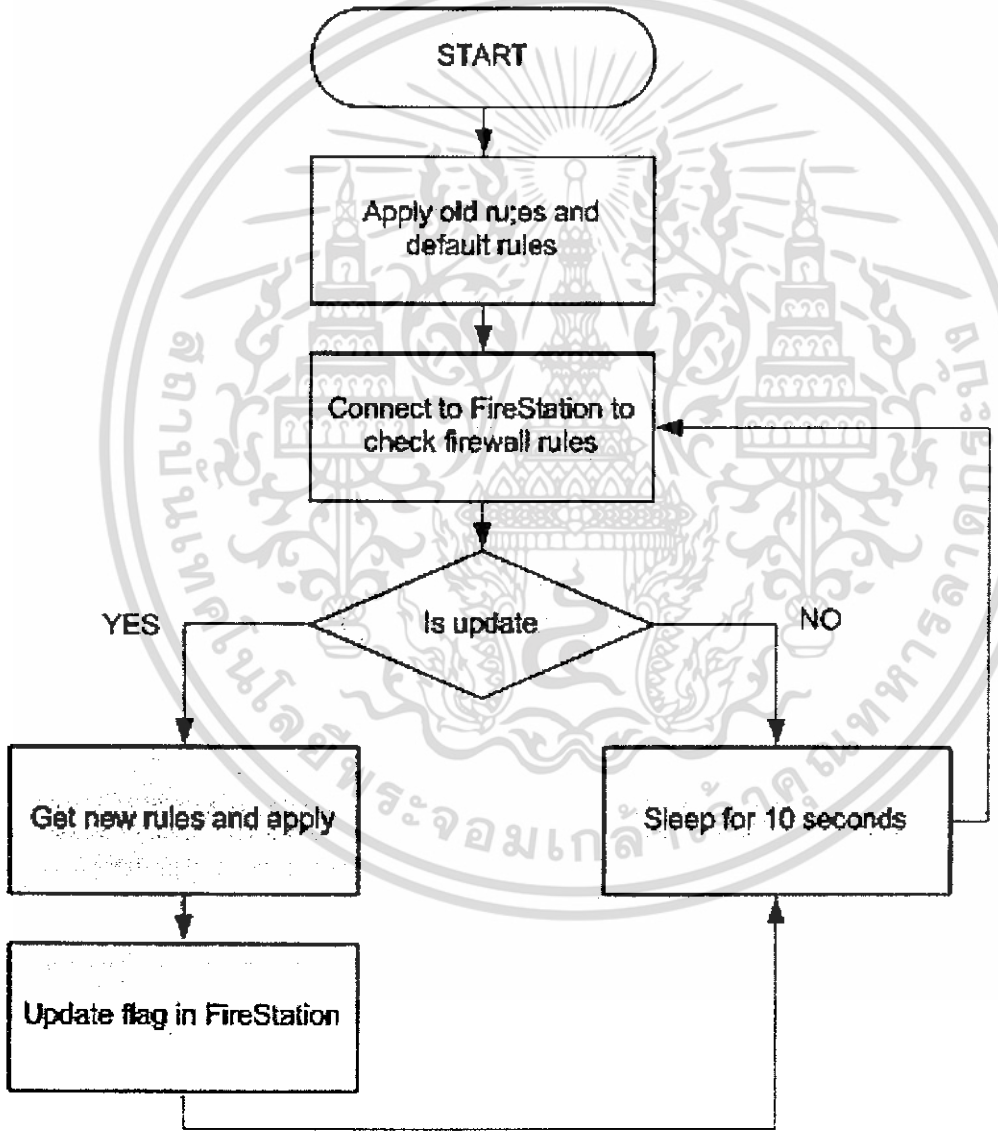
6.4 การทำงานของโปรแกรม

6.4.1 ลักษณะการทำงานของไฟร์สกรีน

การทำงานของไฟร์สกรีนจะเป็นในลักษณะของโปรแกรมเดมอน (daemon) คือทำงานอยู่เบื้องหลังตลอดเวลาตั้งแต่เครื่องไฟร์สกรีน (เกตเวย์) เริ่มทำงานขึ้นมา โดยการทำงานนั้นจะแบ่งออกเป็น 2 ส่วนหลักคือ ส่วนที่ทำหน้าที่รับกฎจากไฟร์สเตชั่นมาใช้ และส่วนของการส่งล็อกกลับไปยังไฟร์สเตชั่น

1. ส่วนที่ทำหน้าที่รับกฎจากไฟร์สเตชั่น

โปรแกรมในส่วนนี้จะทำการนำเอาที่ถูกกำหนดไว้ที่ไฟร์สเตชั่นซึ่งถูกเก็บไว้ในแอคทีฟไดเรกทอรีมาใช้งานกับไอพีเทเบิ้ลส์ สามารถแสดงลักษณะการทำงานเป็นแผนภาพดังนี้

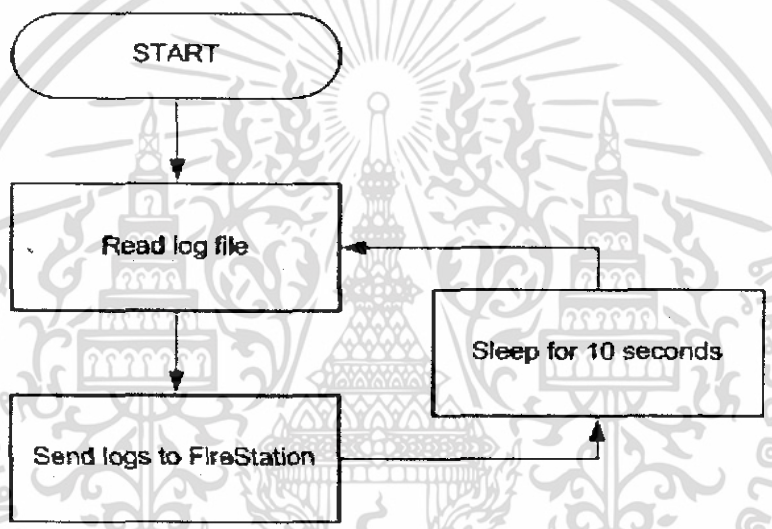


รูปที่ 6.4 แสดงการรับกฎของไฟร์สกรีน

เมื่อเริ่มทำงานขึ้นมาครั้งแรกไฟร์สกรีนจะทำการตั้งกฎพื้นฐานที่ถูกกำหนดไว้แล้วให้กับไอพีเทเบิ้ลส์ก่อน และจากนั้นจึงตรวจสอบดูกฎที่ถูกเก็บไว้เดิมและนำมาใช้งานกับไอพีเทเบิ้ลส์เป็นที่เรียบร้อยเสียก่อน จากนั้นจึงตรวจสอบดูที่ไฟร์สเตชั่นว่าจากที่รับกฎมาครั้งล่าสุดได้มีการเปลี่ยนแปลงแก้ไขหรือไม่ หากมีก็จะนำกฎที่กำหนดใหม่มาใช้ หากไม่มีก็จะหยุดระยะเวลาหนึ่งแล้วจึงคอยตรวจสอบที่ไฟร์สเตชั่นอีกเรื่อยๆ (polling)

2. ส่วนที่ทำหน้าที่ส่งล็อกไปยังไฟร์สเตชั่น

โปรแกรมในส่วนนี้จะทำการอ่านล็อกไฟล์จากเครื่องเกตเวย์และส่งไปเก็บยังแอคทีฟไดเรกทอรีบนไฟร์สเตชั่น สามารถแสดงลักษณะการทำงานเป็นแผนภาพดังนี้



รูปที่ 6.5 แสดงการส่งล็อกของไฟร์สกรีน

เมื่อเริ่มการทำงานขึ้นมาโปรแกรมจะคอยตรวจสอบล็อกไฟล์เป็นระยะ และส่งล็อกที่เกี่ยวข้องในส่วนของไอพีเทเบิ้ลส์กลับไปยังไฟร์สเตชั่น

6.4.2 กฎพื้นฐานบนไฟร์สกรีน

ไฟร์สกรีนเองจะต้องมีการทำงานร่วมกันกับไฟร์สเตชั่นในการรับกฎและส่งล็อกกลับไป ดังนั้นกฎที่ใช้งานบนไฟร์สกรีนแม้รับมาจากตัวไฟร์สเตชั่น แต่บนตัวไฟร์สกรีนเองโปรแกรมจะต้องมีการตั้งกฎพื้นฐานไว้ส่วนหนึ่งก่อนหน้ากฎอื่นๆเพื่อเป็นการป้องกันปัญหาที่อาจเกิดขึ้นหากมีการตั้งกฎที่เป็นปัญหากับการทำงานขึ้นในภาบหลัง เช่นอาจทำให้ไฟร์สกรีนไม่สามารถทำการติดต่อไปยังไฟร์สเตชั่นได้ เป็นต้น นอกเหนือไปจากนั้นยังเป็นเพื่อรับประกันความปลอดภัยพื้นฐานให้กับตัวไฟร์สกรีนและตัวระบบเองด้วยเช่นกัน

รองรับการสามารถติดต่อเข้าไปยังไฟร์สเตชั่นจากเครื่องไคลเอนต์ภายใน

```
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
```

รองรับการติดต่อกลับจากไฟร์สเตชั่น

```
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) แพ็คเก็ตที่พบว่า 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
```

กฎที่กำหนดให้กับเชน FORWARD ทั้งหมดจะต้องใส่ให้กับเชน INPUT และ OUTPUT เพื่อรองรับการติดต่อระหว่างไฟร์สกรีนเองกับไฟร์สเตชั่นและป้องกันความปลอดภัยของตัวไฟร์สกรีนเองด้วยเช่นกัน

บทที่ 7
การพัฒนาโปรแกรมไฟร์เบรก

7.1 แนวคิด

ไฟร์เบรก (FireBreak) เป็นโปรแกรมซีเคียวรีเวิร์สเว็บพร็อกซี่สำหรับเพิ่มความปลอดภัยให้กับเว็บเซิร์ฟเวอร์ซึ่งอยู่ภายในโซน DMZ ใช้ความสามารถของโปรแกรม squid-cache ในการทำงานเป็นรีเวิร์สเว็บพร็อกซี่ทั่วไปและเพิ่มความสามารถในส่วนของระบบการตรวจจับการบุกรุกในระดับของแอพพลิเคชั่นของโปรแกรม snort_inline ซึ่งสามารถเก็บล็อกไฟล์ได้จะถูกส่งไปยังดาต้าเบสของเครื่องแม่ข่าย

7.2 ขอบเขตและความสามารถ

โปรแกรมนี้สามารถตรวจสอบแพ็คเก็ตที่จะเข้าสู่เว็บเซิร์ฟเวอร์ที่อยู่ข้างหลังได้ทำให้ผู้ที่ต้องการติดต่อกับเว็บเซิร์ฟเวอร์ไม่สามารถติดต่อได้โดยตรง ดังนั้นการที่จะโจมตีเว็บเซิร์ฟเวอร์โดยตรงจึงไม่สามารถทำได้ ทำให้การโจมตีที่เกิดขึ้นนั้นเกิดขึ้นกับตัวเครื่องที่ทำหน้าเป็นซีเคียวรีเวิร์สเว็บพร็อกซี่นี้แทน ซึ่งโปรแกรมนี้เป็นการรวมความสามารถของโปรแกรม Squid และ Snort_inline เพื่อเพิ่มความสามารถในการป้องกันการโจมตีหรือการบุกรุกต่างๆที่อาจเกิดขึ้นได้

แต่ความสามารถในการตรวจสอบของโปรแกรมนี้ยังมีข้อจำกัดอยู่บ้างเนื่องจากการที่จะหวังพึ่งแต่ความสามารถของ Snort_inline ในการตรวจสอบการบุกรุกทางแอพพลิเคชั่นหรือทางเว็บนั้นแต่เพียงอย่างเดียว คงไม่อาจทำให้เว็บเซิร์ฟเวอร์ หรือตัวเครื่องที่รันโปรแกรมนี้อยู่มีความปลอดภัยได้ทั้งหมด เนื่องจากการบุกรุกทางแอพพลิเคชั่นนั้นก็ได้มีการพัฒนาและมีรูปแบบต่างๆมากมาย ทำให้ต้องมีการอัพเดตตัว rule ของ Snort_inline อยู่เสมอ และขึ้นอยู่กับการเขียนโค้ดของเว็บเซิร์ฟเวอร์เองด้วยว่ามีการเขียนป้องกันการบุกรุกไว้ด้วยหรือไม่

7.3 การพัฒนาโปรแกรม

โปรแกรมไฟร์เบรกเป็นสามารถแบ่งการทำงานได้ออกเป็น 2 ส่วนคือ

1. โปรแกรม Squid ที่ทำเป็นรีเวิร์สเว็บพร็อกซี่
2. โปรแกรม Snort_inline ที่ช่วยเพิ่มความสามารถในการป้องกันให้กับโปรแกรมนี้

7.3.1 โปรแกรม Squid

สำหรับความสามารถในการเป็นรีเวิร์สเว็บพร็อกซี่นี้ จะใช้ Squid-cache ซึ่งเป็นโปรแกรมแบบเปิดเผยซอร์สโค้ดสำหรับใช้งานเป็นเว็บพร็อกซี่ที่แพร่หลายโปรแกรมหนึ่ง ที่สามารถนำมาประยุกต์ใช้เป็นรีเวิร์สเว็บพร็อกซี่สำหรับเว็บเซิร์ฟเวอร์ได้

สามารถดาวน์โหลดซอร์สโค้ดเวอร์ชั่นล่าสุดของ Squid-cache ได้จากเว็บ http://www.squid-cache.org/ ซึ่งเวอร์ชั่นที่นำมาใช้นี้คือ เวอร์ชั่น 2.5

Squid-cache มีลักษณะการตั้งค่าการใช้งานโดยต้องทำการแก้ไขในไฟล์ squid.conf ซึ่งอยู่ที่ /etc/squid/squid.conf ซึ่งในไฟล์มีคอมเมนต์อธิบายการใช้งานคำสั่งที่เกี่ยวข้องกับการตั้งค่าของ squid-cache ไว้อยู่แล้ว สำหรับการตั้งค่าของ squid-cache ในไฟล์ squid.conf ที่ใช้งานในโครงงานนี้มีดังนี้

```
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

ใช้กำหนดพอร์ตที่จะติดต่อเข้ามายังพร็อกซี่มีค่าเริ่มต้นเป็น 3128 สำหรับรีเวิร์สเว็บพร็อกซี่ http_port ก็ต้องกำหนดค่าเป็น 80 ซึ่งเป็นพอร์ตทั่วไปในการใช้งาน HTTP

- visible_hostname

ใช้สำหรับระบุโฮสต์ที่เป็นเว็บเซิร์ฟเวอร์เมื่อเกิด error_message ในการทำงานของ squid กรณีที่ไม่สามารถระบุได้ว่าโฮสต์ที่เป็นเว็บเซิร์ฟเวอร์คือเครื่องใด

- httpd_accel_host

ใช้ระบุโฮสต์ที่เป็นเว็บเซิร์ฟเวอร์จริงๆ ในการทำเป็น Transparent Proxy

- httpd_accel_port

ใช้ระบุพอร์ตที่ใช้งานของเว็บเซิร์ฟเวอร์ ในการทำเป็น Transparent Proxy

- http_access

ใช้ระบุสิทธิ์ผู้ที่สามารถใช้งานรีเวิร์สเว็บพร็อกซี่นี้ได้ โดยค่าเริ่มต้นจะเป็น deny all แต่การใช้งานเป็นรีเวิร์สเว็บพร็อกซี่จะรองรับบริการทั้งจากภายในและจากอินเตอร์เน็ตเองจะสามารถเข้าถึงได้ จึงต้องเปลี่ยนเป็น allow all

ส่วนความสามารถในการ cache ของ squid นั้นไม่ได้นำมาใช้ใน โครงงานนี้เนื่องจากเว็บเซิร์ฟเวอร์ที่อยู่หลังรีเวิร์สเว็บพร็อกซี่นั้นมีเพียงเครื่องเดียวจึงขอไม่กล่าวถึงการตั้งค่าในส่วนนี้

7.3.2 โปรแกรม Snort_inline

สำหรับความสามารถของระบบการตรวจจับผู้บุกรุกนั้นจะใช้ snort_inline ซึ่งเป็น Intrusion Prevention System (IPS) ที่มีความสามารถในการตรวจจับแพ็กเก็ตที่เป็นอันตรายต่อเว็บเวิร์ฟเวอร์และสามารถดรอปแพ็กเก็ตนั้นทิ้งได้ทันที โดย snort_inline นี้เป็นโปรแกรมแบบเปิดเผยซอร์สโค้ด

สามารถดาวน์โหลดซอร์สโค้ดเวอร์ชั่นล่าสุดของ Squid-cache ได้จากเว็บ http://prdownloads.sourceforge.net/snort-inline/snort_inline-2.3.0-RC1.tar.gz?download ซึ่งโครงงานนี้ใช้ snort_inline เวอร์ชั่น 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
```

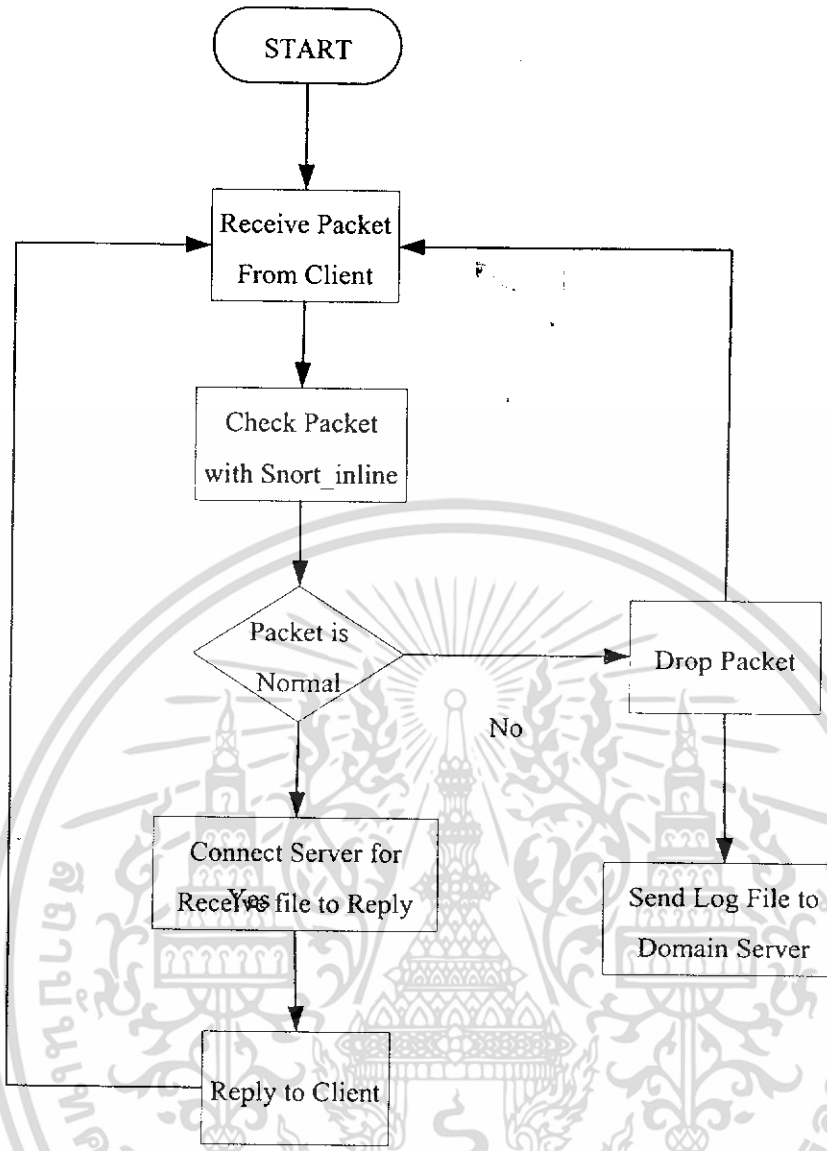
โดยในเครื่องแม่ข่ายจะต้องมีการลง mysql เวอร์ชั่น 4.1 ที่มีการสร้าง database ชื่อ snort ในดาต้าเบสนี้สร้าง table ต่างๆ โดยใช้ไฟล์ create_mysql ของ snort_inline และมีการสร้าง user ใหม่ที่รองรับการใช้งานของดาต้าเบส snort ชื่อ snort ที่กำหนด password ในการใช้งานคือ 123 โดยต้องกำหนดสิทธ์ในการใช้งานของ user snort ให้สามารถทำการ select, insert ดาต้าเบส snort ได้

จากนั้นทำการอัพเดต signature หรือ rule ของ snort_inline ให้มีความสามารถในการป้องกันเว็บเซิร์ฟเวอร์ได้ โดยทำการโหลด ruleใหม่ๆสำหรับ snort_inline ได้ที่ http://www.bleedingsnort.com/ สำหรับไฟล์ rule ที่สำคัญในการป้องกันเว็บเซิร์ฟเวอร์ คือ bleeding-web.rules, bleeding-expliot.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-expliot.rules
include $RULE_PATH/bleeding-scan.rules
include $RULE_PATH/bleeding-attack response.rules
```

7.4 การทำงานของโปรแกรม

สำหรับการทำงานของโปรแกรมไฟร์เบรกนี้เป็นไปดังรูปที่ 7.1 คือ หลังจากที่ทำการตั้งค่าการใช้งานและทำการเปิดเซอร์วิสเริ่มการใช้งานของ Squid และ Snort_inline แล้วก็จะเท่ากับอยู่ในสถานะ START ในรูปที่ 7.2 เพื่อรอรับแพ็คเก็ตที่มาจากไคลเอ็นต์ที่ต้องการจะติดต่อกับเว็บเซิร์ฟเวอร์ที่อยู่หลังจากเครื่องรีเวิร์สเว็บพร็อกซี่นี้ เมื่อได้รับแพ็คเก็ต Snort_inline จะทำการตรวจสอบแพ็คเก็ตนั้นกับ rule ที่มีอยู่ว่าเข้าข่ายพฤติกรรมที่ผิดปกติหรือไม่ ถ้าแพ็คเก็ตนั้นผิดปกติ Snort_inline จะทำการดร็อปแพ็คเก็ตนั้นทิ้งและทำการบันทึกลงล็อกไฟล์ส่งกลับไปยังเครื่องแม่ข่าย แต่ถ้าแพ็คเก็ตนั้นเป็นแพ็คเก็ตปกติจะทำการติดต่อกับเว็บเซิร์ฟเวอร์เพื่อรับไฟล์ที่จะต้องทำการตอบกลับให้กับไคลเอ็นต์ถ้าต้องมีการตอบกลับ แล้วจึงจะทำการส่งไฟล์ที่ได้รับจากเว็บเซิร์ฟเวอร์นั้นให้กับไคลเอ็นต์ต่อไป



รูปที่ 7.1 แสดงการทำงานของโปรแกรมไฟร์เบรก

บทที่ 8
การพัฒนาโปรแกรมล็อกมอนิเตอร์

8.1 แนวคิด

โปรแกรมล็อกมอนิเตอร์นั้นจะถูกติดตั้งอยู่ในเครื่องแม่ข่ายที่เป็นโดเมนคอนโทรลเลอร์ และเนื่องจากการที่มีไฟร์วอลในส่วนต่างๆของระบบเครือข่ายนั้นจึงได้มีการแบ่งแยกล็อกมอนิเตอร์สำหรับแต่ละโปรแกรม คือ ล็อกมอนิเตอร์ของไฟร์อลาร์ม ล็อกมอนิเตอร์ของไฟร์สรีน ที่สามารถเก็บล็อกลงในแอ็คทีฟไดเรกทอรีของเครื่องแม่ข่ายได้ จึงพัฒนาโปรแกรมล็อกมอนิเตอร์ด้วยภาษา C ที่มีการใช้คลาส ADCONNECT ในการพัฒนาโปรแกรม และโปรแกรมล็อกมอนิเตอร์ของไฟร์เบรก ที่ต้องเก็บล็อกในดาต้าเบส MySql เนื่องจากล็อกของ Snort_inline นั้นไม่สามารถเก็บลงแอ็คทีฟไดเรกทอรีได้ จึงพัฒนาโปรแกรมล็อกมอนิเตอร์ด้วยภาษา C ที่มีการใช้คลาส MySqlConnect ในการติดต่อกับ MySql

8.2 ขอบเขตและความสามารถ

ล็อกมอนิเตอร์ของไฟร์อลาร์มและไฟร์สกรีนนั้นสามารถแสดงผลล็อกไฟล์ที่ถูกส่งกลับมายังเครื่องแม่ข่ายได้ และสามารถลบล็อกไฟล์ที่ต้องการได้ ส่วนล็อกมอนิเตอร์ของไฟร์เบรกนั้นสามารถแสดงผลล็อกไฟล์ที่ถูกส่งกลับมายังเครื่องแม่ข่ายได้เท่านั้น ในการลบล็อกไฟล์ที่ต้องการนั้นจะต้องทำการลบผ่านโปรแกรมที่ใช้ในการติดต่อกับ MySql โดยผู้ที่มีสิทธิในการใช้งานดาต้าเบสนี้เท่านั้น

8.3 การพัฒนาโปรแกรม

8.3.1 ล็อกมอนิเตอร์ของไฟร์อลาร์ม

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 ล็อกมอนิเตอร์ของไฟร์สกรีน

โปรแกรมนี้จะทำการพัฒนาโดยภาษา C โดยการใช้คลาส ADCONNECT ที่ใช้ในการติดต่อกับแอ็คทีฟไดเรกทอรี เพื่อเข้าถึงค่าของล็อกไฟล์ที่ถูกส่งมาเก็บยังแอ็คทีฟไดเรกทอรี จากนั้นจะนำค่าที่ได้นี้มาลงไฟล์ดาต้าเบสที่สร้างจาก Microsoft Access ที่มีชื่อว่า LogfireScreen โดยมีโครงสร้างตารางที่ใช้ในการเก็บล็อกของเครื่องลูกข่ายดังตารางที่ 8.2 และแสดงผลผ่านโปรแกรม

ตารางที่ 8.2 แสดงโครงสร้างตารางที่ใช้ในการเก็บล็อกของเครื่องเกต์เวย์

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

8.3.3 ล็อกมอนิเตอร์ของไฟร์เบรก

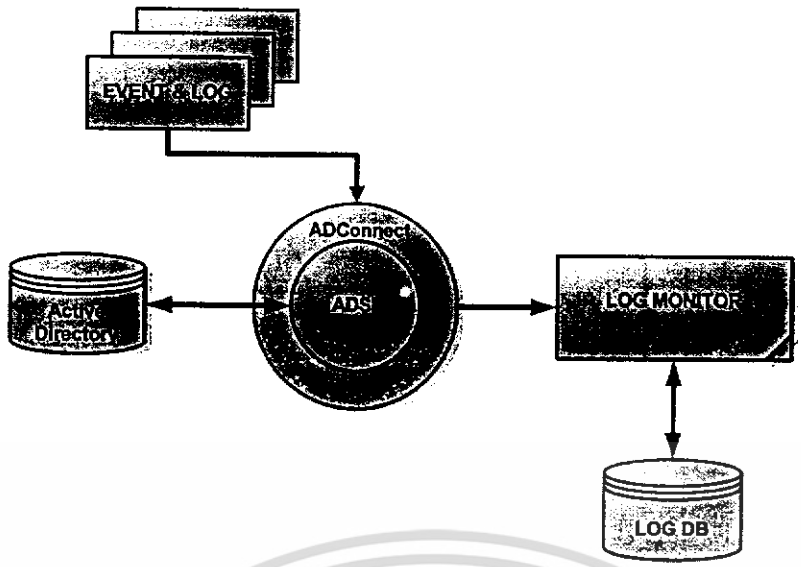
โปรแกรมนี้จะทำการพัฒนาโดยภาษา C โดยการใช้คลาส MySqlConnect ที่สร้างขึ้นเพื่อใช้ในการติดต่อกับ MySql เพื่อเข้าถึงค่าล็อกไฟล์ที่ถูกส่งมาเก็บยังดาต้าเบสที่ชื่อ Snort ของMySql จากนั้นจะนำค่าที่ได้แสดงผลผ่านโปรแกรม โดยมีโครงสร้างของดาต้าเบส Snort และเทเบิลต่างๆ จากไฟล์ Create_mysql ของโปรแกรม Snort_inline 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

โปรแกรมล็อกมอนิเตอร์ของไฟร์อลาร์มและล็อกมอนิเตอร์ของไฟร์สกรีน จะมีการทำงานอยู่บนความสามารถของบริการแอ็คทีฟไดเร็คทอรี(Active directory) ของ windows 2003 โดยข้อมูลจะเก็บอยู่ในฐานข้อมูลของแอ็คทีฟไดเร็คทอรี (Active Directory Database) และเข้าถึงข้อมูลผ่าน ADSI ไปยังทีฟไดเร็กทอรีเซอร์วิส (Active Directory Services) เพื่อเข้าถึงข้อมูลที่จัดเก็บ ซึ่งการทำงานของล็อกมอนิเตอร์ที่ทำงานร่วมกันกับ แอ็คทีฟไดเร็คทอรี (Active Directory) แสดงโครงสร้างการทำงาน ดังรูปที่ 8.1



รูปที่ 8.1 แสดงโครงสร้างการทำงานระหว่างล็อกมอนิเตอร์กับแอ็คทีฟไดเรคทอรี่

ส่วนโปรแกรมล็อกมอนิเตอร์ของไฟร์เบรกนั้นจะมีการทำงานโดยการติดต่อกับดาต้าเบส 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

การทดสอบการทำงานของระบบชุดโปรแกรมเน็ตเวิร์กไฟร์วอลได้ทำการทดสอบภายใต้โปรแกรมจำลองเครื่องคอมพิวเตอร์มีรายละเอียดดังตารางที่ 9.1

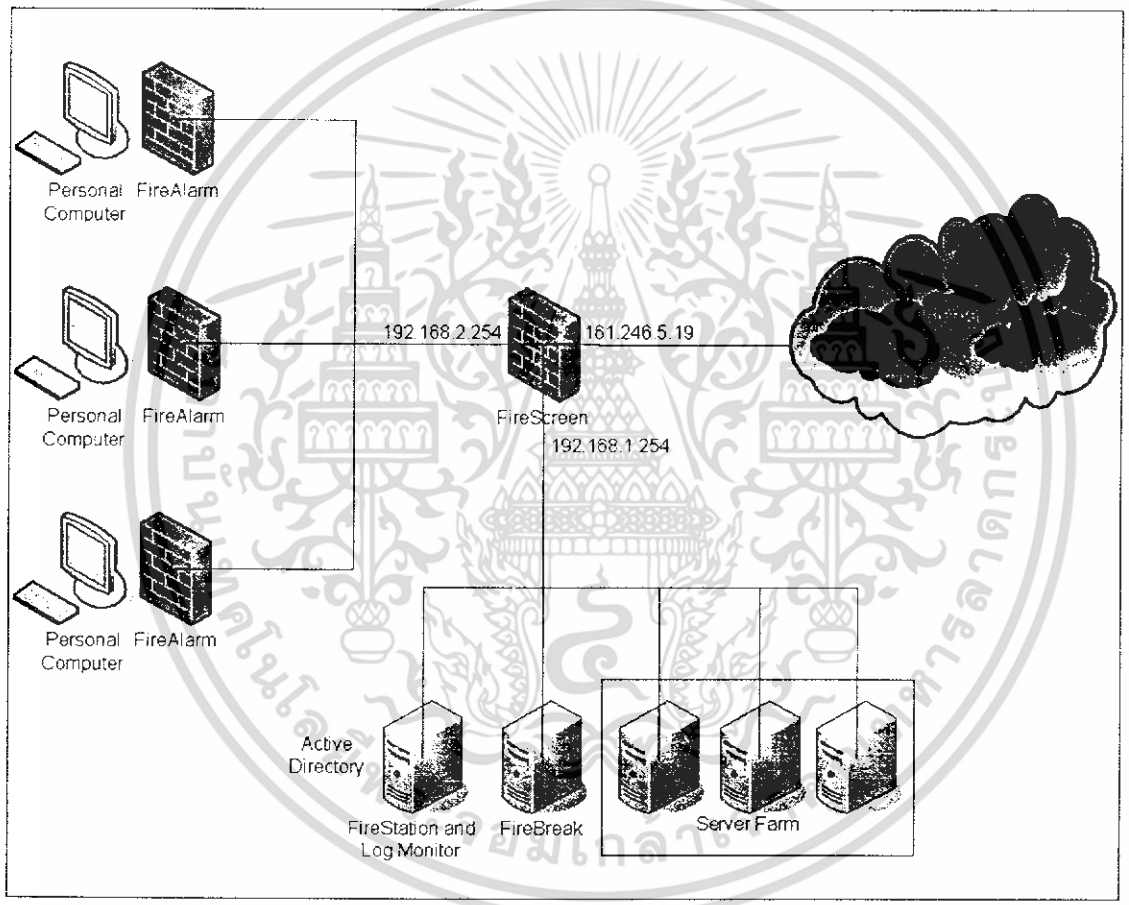
ตารางที่ 9.1 รายละเอียดของเครื่องคอมพิวเตอร์ต่างๆภายในระบบ

	ไฟร์สเตชั่น (FireStation)	ไฟร์อลาร์ม (FireAlarm)	ไฟร์สกรีน (FireScreen)	ไฟร์เบรก (FireBreak)	อินทรูเดอร์ (Intruder)
ระบบปฏิบัติการ	Windows Server 2003 Enterprise Edition	Windows XP Professional SP2	Debain GNU/Linux	Debain GNU/Linux	Debain 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	เป็นศูนย์กลางการกำหนดกฎ	เป็นเครื่องลูกข่ายภายในซึ่งติดตั้งไฟร์อลาร์มไว้	เป็นเกต์เวย์ซึ่งติดตั้งไฟร์สกรีนไว้	เป็นรีเวิร์สเว็บพร็อกซี่	เป็นเครื่องผู้บุกรุก

โดยที่เครื่องไฟร์สเตชั่นนั้นจะมีการติดตั้งแอ็คทีฟไดเร็คทอรีเซอร์วิส(Active Directory) ลงไปและเพิ่มแอตตริบิว (Attribute) เพื่อความสามารถในการใช้เป็นศูนย์กลางการควบคุมไฟร์วอลส่วนต่างๆ ส่วนที่เครื่องไฟร์อลาร์มจะมีการลงWinPCap เพื่อใช้ในการดักจับแพ็คเก็ตของโปรแกรมไฟร์อลาร์มในส่วนของการตรวจจับผู้บุกรุก

9.2 โครงสร้างของระบบที่ใช้ในการทดสอบ

โครงสร้างของระบบที่สามารถทำงานได้เต็มประสิทธิภาพนั้น ควรติดตั้งให้ครบทุกส่วน คือ ส่วนไฟร์สเตชั่น ไฟร์อลาร์ม ไฟร์เบรก และไฟร์สกรีน ดังรูปที่ 9.1

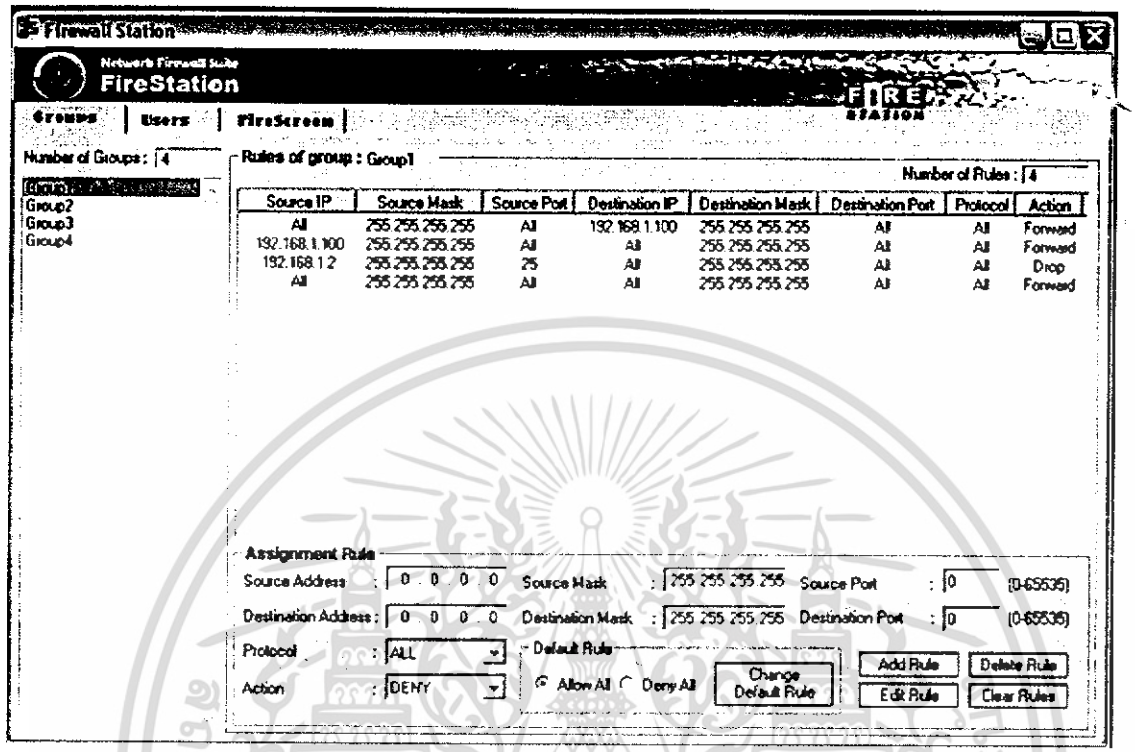


รูปที่ 9.1 โครงสร้างทางเครือข่ายของระบบที่ใช้ในการทดสอบ

9.3 การทดสอบการทำงาน

9.3.1 ทดสอบการทำงานของไฟร์สเตชั่น(FireStation)

เมื่อโปรแกรมไฟร์สเตชั่นเริ่มต้นการทำงานแล้วจะมีลักษณะดังต่อไปนี้



รูปที่ 9.2 โปรแกรมไฟร์สเตชั่น

โปรแกรมไฟร์สเตชั่นจะแบ่งออกเป็น 3 ส่วนด้วยกันคือ

- ส่วนการกำหนดกฎให้กับกลุ่มผู้ใช้
- ส่วนการกำหนดกฎให้กับผู้ใช้เป็นรายบุคคล
- ส่วนการกำหนดกฎให้กับไฟร์วอลบนเกต์เวย์หรือไฟร์สกรีน(FireScreen)

การทำงานของโปรแกรมนี้จะเป็นการกำหนดกฎให้กับไฟร์วอลส่วนต่างๆในระบบ สามารถทำการเพิ่ม แก้ไข และลบ กฎได้โดยการนำค่าที่กำหนดไปใส่ในแอตตริบิวของอีอปเจ็คนั้นๆ โดยกฎพื้นฐานจะเป็น Deny All คือไม่อนุญาตให้การติดต่ออื่นๆผ่านเข้ามาได้

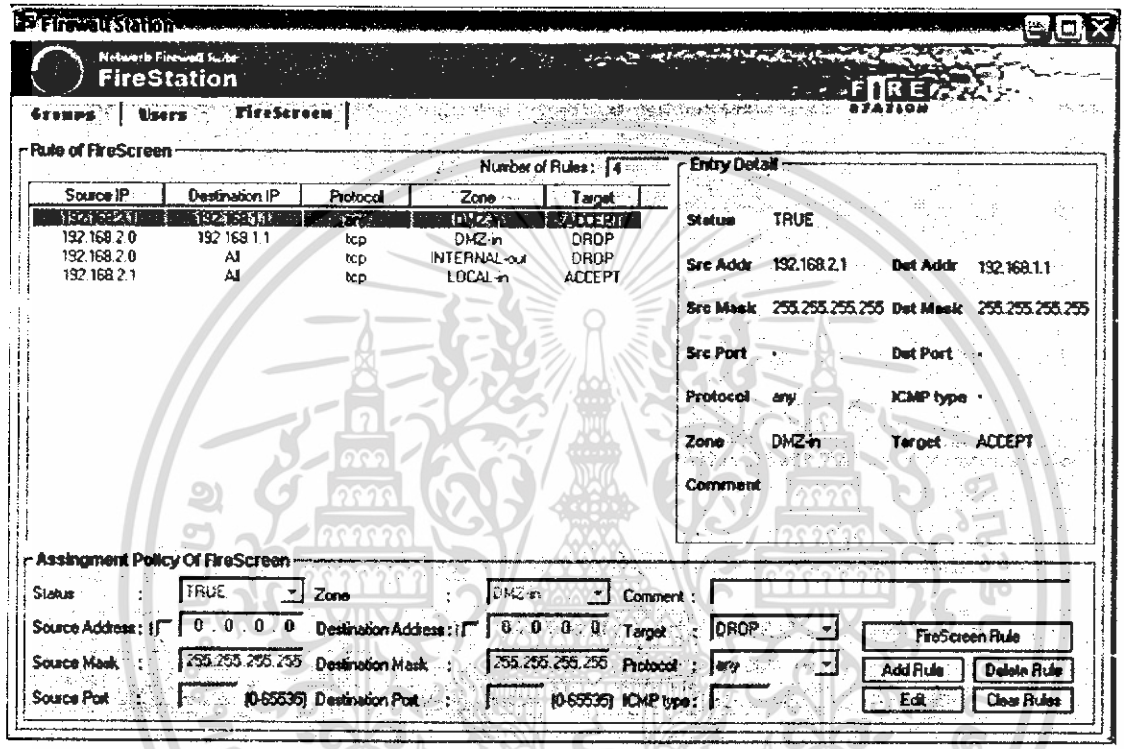
ตัวอย่างการกำหนดกฎให้กับกลุ่มและผู้ใช้รายบุคคล ถ้าต้องการบล็อกทุกอย่างทุกไอพีแต่ให้สามารถใช้ ICMP ได้ของเครื่อง 192.168.2.2 ให้ใส่กฎเพิ่มกฎดังตารางที่ 9.2 นี้

ตารางที่ 9.2 แสดงการกำหนดกฎตัวอย่าง

Src.IP	Src.Mask	Src.Port	Dst. IP	Dst.Mask	Dst.Port	Protocol	Action
192.168.2.2	255.255.255.255	0	0.0.0.0	255.255.255.255	0	ICMP	ALLOW

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



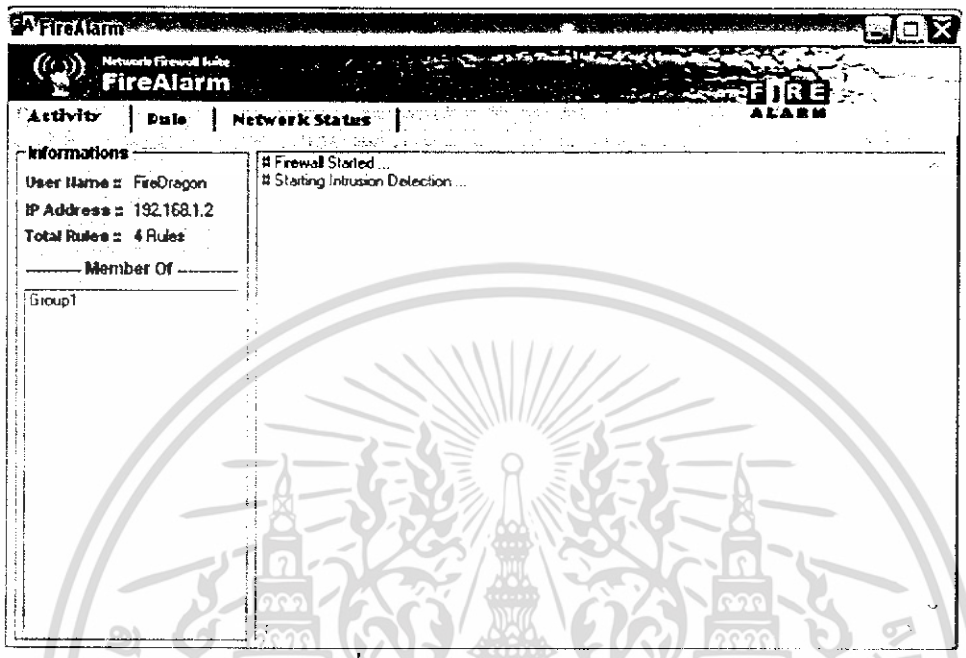
รูปที่ 9.3 โปรแกรมไฟร์สเตชั่นส่วนของการกำหนดกฎให้กับไฟร์สกรีน(FireScreen)

กฎที่กำหนดให้กับแต่ละส่วนจะถูกเก็บไว้ในแอคทีฟไดเรกทอรีในตำแหน่งที่ถูกต้องเพื่อรอไคลเอนต์ที่เป็นเจ้าของกฎนั้นมานำไปใช้ต่อไป ในส่วนของไฟร์สกรีนนั้นกฎที่ใส่ไว้ทั้งหมดนี้จะถูกนำไปเก็บในแอคทีฟไดเรกทอรีในแอตทริบิวต์ firewallLog ของ DN คือ "cn=firescreen, cn=computers, dc=hephaestus, dc=com" และหลังจากกฎถูกเปลี่ยนแปลงแก้ไขอย่างไรก็ตามจะไปเปลี่ยนค่าในแอตทริบิวต์ update ของ DN เดียวกันนี้เป็น "YES" ด้วย

เช่นเดียวกันกฎสำหรับไฟล์สกรีนจะถูกตีความหมายจากบนลงล่างเช่นกัน สำหรับตัวอย่างข้างต้น 2 กฎแรกสำหรับแพ็กเก็ตที่จะผ่านเข้าไปยังโซน DMZ กฎที่ 3 สำหรับแพ็กเก็ตที่ผ่านออกมาจากเครือข่ายภายใน และสุดท้ายจะเป็นสำหรับแพ็กเก็ตที่เข้ามายังเครื่องเกตเวย์เอง

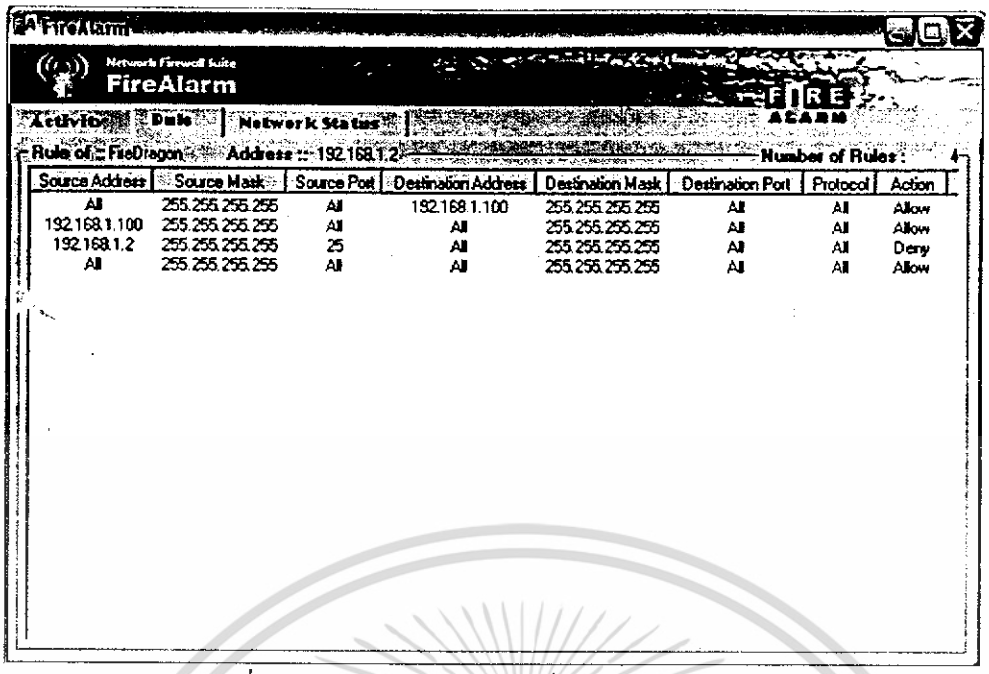
9.3.3 ทดสอบการทำงานของไฟร์อลาร์ม(FireAlarm)

เมื่อโปรแกรมเพอร์ซันนอลไฟร์วอลเริ่มทำงานแล้ว จะมีลักษณะดังต่อไปนี้



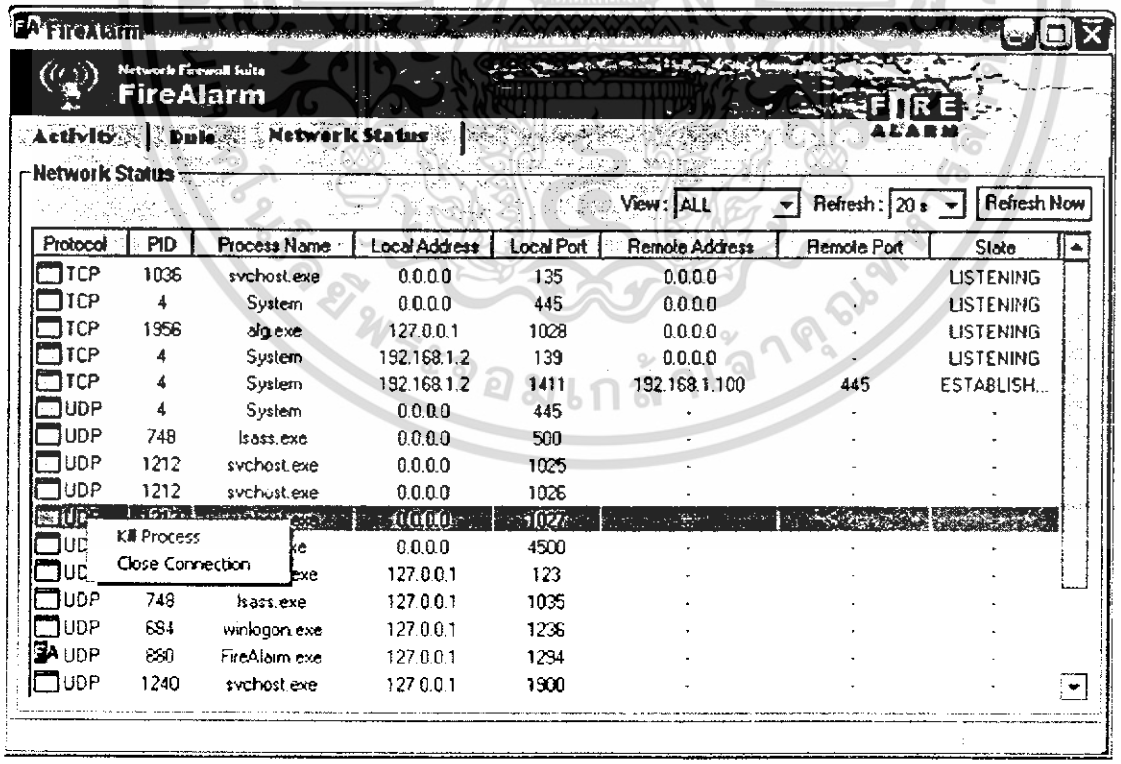
รูปที่ 9.4 โปรแกรมไฟร์อลาร์ม

เมื่อโปรแกรมเริ่มทำงานจะทำการรับเอากฎการฟิลเตอร์ของกลุ่มและผู้ใช้ที่ล็อคอิน(Login) ตามที่กำหนดไว้ที่ส่วนกลางมาเป็นกฎการฟิลเตอร์ของตัวโปรแกรมไฟร์อลาร์มก่อนจะเริ่มทำงาน ด้วยการฟิลเตอร์ต่างๆ ตามกฎการฟิลเตอร์ที่รับมาจากส่วนกลาง และจะเริ่มการทำงานของส่วนตรวจจับผู้บุกรุกด้วยทันที โดยกฎที่รับมาจะแสดงดังรูป 9.5



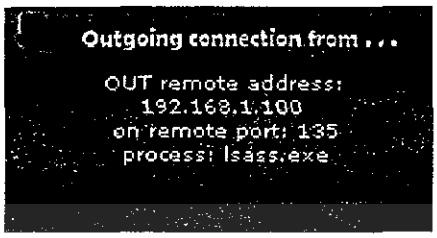
รูปที่ 9.5 แสดงกฎการฟิลเตอร์ที่รับมาจากเซิร์ฟเวอร์

โปรแกรมไฟร์อลาร์มในส่วนของสถานะทางเครือข่าย (Network Status) จะแสดงรายการของโปรแกรมที่ใช้งานเครือข่ายอยู่ในขณะนั้นโดยจะอัปเดตรายการตามเวลาที่ได้ตั้งเอาไว้ และในส่วนนี้สามารถเลือกจบโปรแกรมการทำงาน (Kill Process) หรือปิดการเชื่อมต่อของโปรเซส (Close Connection) ใดๆได้ดังแสดงในรูป 7-4



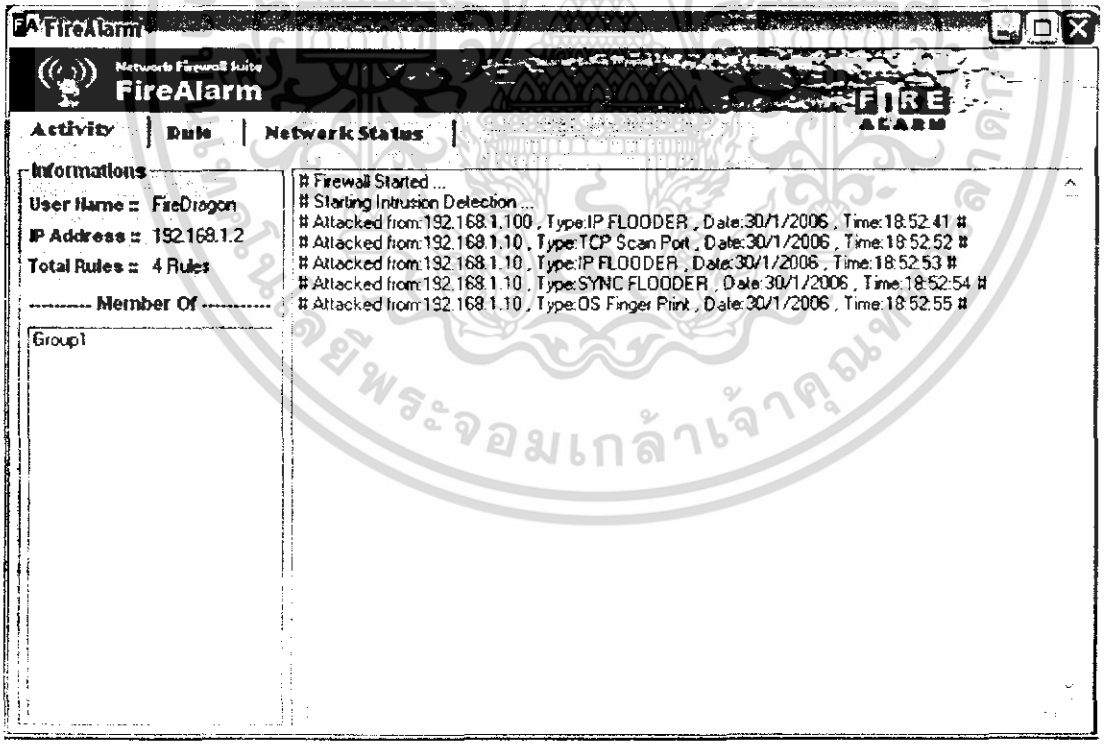
รูปที่ 9.6 แสดงสถานะของโปรเซสที่ใช้งานเครือข่าย

เมื่อมีโปรเซสใดที่เชื่อมต่อเครือข่ายจะป๊อปอัพ (Popup) ขึ้นมาบอกทิศทางและไอพีแอดเดรสที่โปรเซสนั้นเชื่อมต่อ ดังแสดงในรูปที่ 9.7



รูปที่ 9.7 แสดงป๊อปอัพ(Popup) แสดงโปรเซสที่ใช้งานเครือข่าย

และเมื่อมีการโจมตีเกิดหรือมีการบุกรุกทางเครือข่ายเกิดขึ้น โปรแกรมก็จะแสดงรายละเอียดดังรูปที่ 9.8 ทำการส่งรายละเอียดต่างๆ ของการโจมตีหรือการบุกรุกนั้นๆไปเก็บไว้ที่แอตทริบิวต์ firewallLog ของกลุ่มของผู้ใช้ในแอ็คทีฟไดเร็คทอรี (Active Directory)



รูปที่9.8 การแจ้งเตือนการบุกรุกหรือถูกโจมตี

9.3.3 ทดสอบการทำงานของไฟร์สกรีน(Firescreen)

หลังจากเครื่องเกตเวย์ซึ่งติดตั้งโปรแกรมไฟร์สกรีนไว้เริ่มต้นทำงานขึ้นมาขั้นแรก โปรแกรมจะทำการใส่กฎเริ่มต้นและกฎเดิมเข้ากับเกตเวย์เสียก่อน โดยในที่นี้จะเป็นการทำงานครั้งแรกของโปรแกรมไฟร์สกรีนจะไม่มีกฎเก่าก่อนอยู่ เหลือเพียงแค่กฎเริ่มต้นที่ตั้งไว้แล้วเท่านั้น

ใช้คำสั่ง iptables -L --line-number เพื่อตรวจสอบดูกฎในไอพีเทเบิ้ลส์ว่าถูกต้องตามต้องการหรือไม่

```
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 RELATE
D,ESTABLISHED

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     tcp  --  192.168.2.0/24       firestation.hephaestus.com tcp dpt:686
2    ACCEPT     tcp  --  firestation.hephaestus.com  192.168.2.0/24      state RELATED,ESTA
BLISHED
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
```

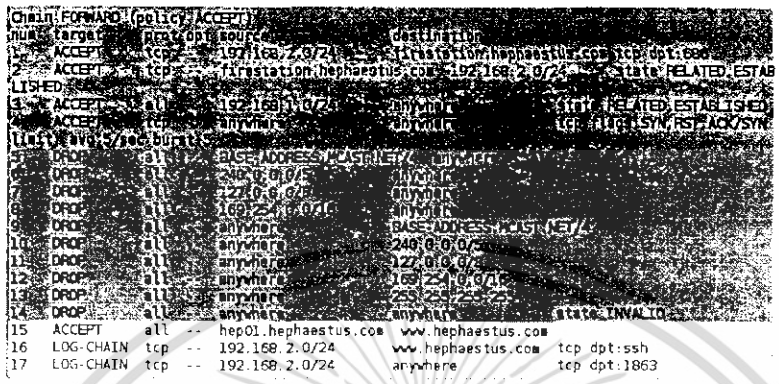
รูปที่ 9.9 แสดงกฎเริ่มต้นของไฟร์สกรีน

จะพบว่ากฎพื้นฐานที่กำหนดไว้ในเชลล์สคริปต์ init.sh ถูกกำหนดให้กับไอพีเทเบิ้ลส์เรียบร้อย รวมถึงการสร้างเชน LOG-CHAIN ซึ่งใช้สำหรับเก็บล็อกรายละเอียดของแพ็กเก็ตที่จะละทิ้งไป (DROP) อนึ่งสำหรับแพ็กเก็ตที่ละทิ้งไปในกฎพื้นฐานนี้จะไม่มีการเก็บล็อกไว้แต่อย่างใด เฉพาะส่วนที่ละทิ้งไปเนื่องจากกฎที่กำหนดไว้ที่ไฟร์สเตชั่นเท่านั้นที่จะมีการเก็บรายละเอียดลงล็อกไฟล์

ขั้นตอนต่อไปจะเป็นการนำกฎจากไฟร์สเตชั่นที่ได้กำหนดไว้ก่อนหน้ามาใช้ ไฟร์สกรีนจะคอยตรวจสอบไปยังไฟร์สเตชั่นเป็นระยะเพื่อดูว่ามีการเปลี่ยนแปลงแก้ไขหรือไม่ กรณีนี้เพิ่งได้ทำ

การเพิ่มกฎเข้าไปใหม่ ดังนั้นไฟร์สกรีนจะทำการนำกฎทั้งหมดมาจากไฟร์สเตชั่นต่อท้ายกฎพื้นฐานก่อนหน้านี้

ตรวจสอบดูด้วยคำสั่ง iptables -L --line-number เพื่อดูว่าไอพีเทเบิ้ลส์ได้รับกฎตามที่กำหนดไว้โดยไฟร์สเตชั่นถูกต้องหรือไม่



รูปที่ 9.10 กฎจากไฟร์สเตชั่นที่เพิ่มเข้าไปในเชนฟอร์เวิร์ด

รูปที่ 9.11 กฎจากไฟร์สเตชั่นที่เพิ่มเข้าไปในเชนอินพุท

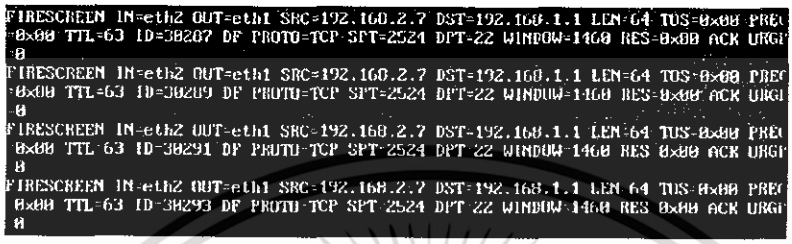
จะสังเกตเห็นว่าในเชนฟอร์เวิร์ดจะพบว่ามีกฎเพิ่มมาอีก 3 กฎเพิ่มขึ้นมาจากกฎพื้นฐานจาก init.sh คือกฎที่ 15, 16 และ 17 ซึ่งเป็น 3 กฎแรกที่กำหนดไว้ในไฟร์สเตชั่นที่ได้แสดงให้ดูในหัว 9.3.1 นั่นเอง (รูปที่ 9.3)

ส่วนในเชนอินพุทก็มีเพิ่มขึ้น 1 กฎเช่นกัน ก็คือกฎสุดท้ายที่ตั้งไว้ที่ไฟร์สเตชั่น (รูปที่ 9.3) อย่างไรก็ตามในส่วนของเชนเอาท์พุทและเชน LOG-CHAIN นั้นจะไม่ถูกทำการเปลี่ยนแปลงแก้ไขแต่อย่างใด

สำหรับส่วนของล็อกไฟล์ หากมีแพ็กเก็ตที่ตรงกับกฎที่ระบุทาร์เก็ตเป็นเชน LOG-CHAIN (ก่อนที่จะถูกละทิ้งไปที่ตอนท้ายของเชน LOG-CHAIN) จะถูกเก็บรายละเอียดลงยังล็อกไฟล์ซึ่งเก็บ

ไว้ที่ /var/log/firescreen.log และจะมีรายละเอียดของล็อกแสดงขึ้นบนหน้าจอด้วย

รายละเอียดของล็อกที่แสดงขึ้นมาเป็นดังรูปที่ 9.12



รูปที่ 9.12 ล็อกที่แสดงขึ้นที่หน้าจอของไฟร์สกรีน

ล็อกดังกล่าวเป็นล็อกจากเครื่องภายในเครือข่ายภายในซึ่งมีไอพีแอดเดรสเป็น 192.168.2.7 พยายามใช้ซีเคียวเชลล์ (ssh) เข้าไปยังเครื่องเว็บเซิร์ฟเวอร์ซึ่งมีไอพีแอดเดรสเป็น 192.168.1.1 แต่ถูกปฏิเสธโดยตัวไฟร์สกรีนจากกฎที่ 16 ในเชนฟอร์เวิร์ด (รูปที่ x)

ล็อกไฟล์จริงๆนั้นเก็บไว้ที่ /var/log/firescreen.log จะมีรูปแบบเช่นเดียวกันกับล็อกที่แสดงบนหน้าจอดังรูปที่ x เพียงแค่เพิ่มวันเวลาและชื่อโฮสต์ด้านหน้า ไฟร์สกรีนจะอ่านไฟล์นี้เป็นระยะเพื่อส่งล็อกในส่วนของไอพีเทเบิ้ลส์ไปเก็บยังแอคทีฟไดเรกทอรีบนไฟร์สเตชั่น (ล็อกใน firescreen.log ไม่ได้มีเฉพาะล็อกจากไอพีเทเบิ้ลส์แต่ประกอบด้วยล็อกอื่นๆในระดับเดียวกันด้วยเช่นกัน)

ข้อมูลภายใน firescreen.log เป็นดังแสดงในรูปที่ 9.13

```
Jan 11 02:43:33 firescreen 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 firescreen 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 firescreen 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/firescreen.log

ล็อกจะถูกส่งไปเก็บยังแอคทีฟไดเรกทอรีบนไฟร์สเตชั่นในแอตทริบิวต์ชื่อ firewallLog ภายใต้ DN = "cn=firescreen, cn=computers, dc=hephaestus, dc=com" ส่วนการนำล็อกมาแสดงบนไฟร์สเตชั่นจะกล่าวถึงในหัวข้อถัดไป

9.3.4 การทดสอบการทำงานของไฟร์เบรก

โปรแกรมนี้เป็นการทำงานร่วมกันของ Squid และ Snort_inline ดังนั้นการเริ่มทำงานของโปรแกรมก็คือการเปิดเซอร์วิสของ Squid และ Snort_inline ให้เริ่มทำงานบนเครื่องที่ทำหน้าที่เป็นซีเคียวรีเวิร์สเว็บพร็อกซี่ ซึ่งการเปิดเซอร์วิสของ squid ให้เริ่มใช้งานด้วยคำสั่ง

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

และการสั่งให้ Snort_inline เริ่มทำงานโดยต้องมีการตั้งค่าการใช้งานของ 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
```

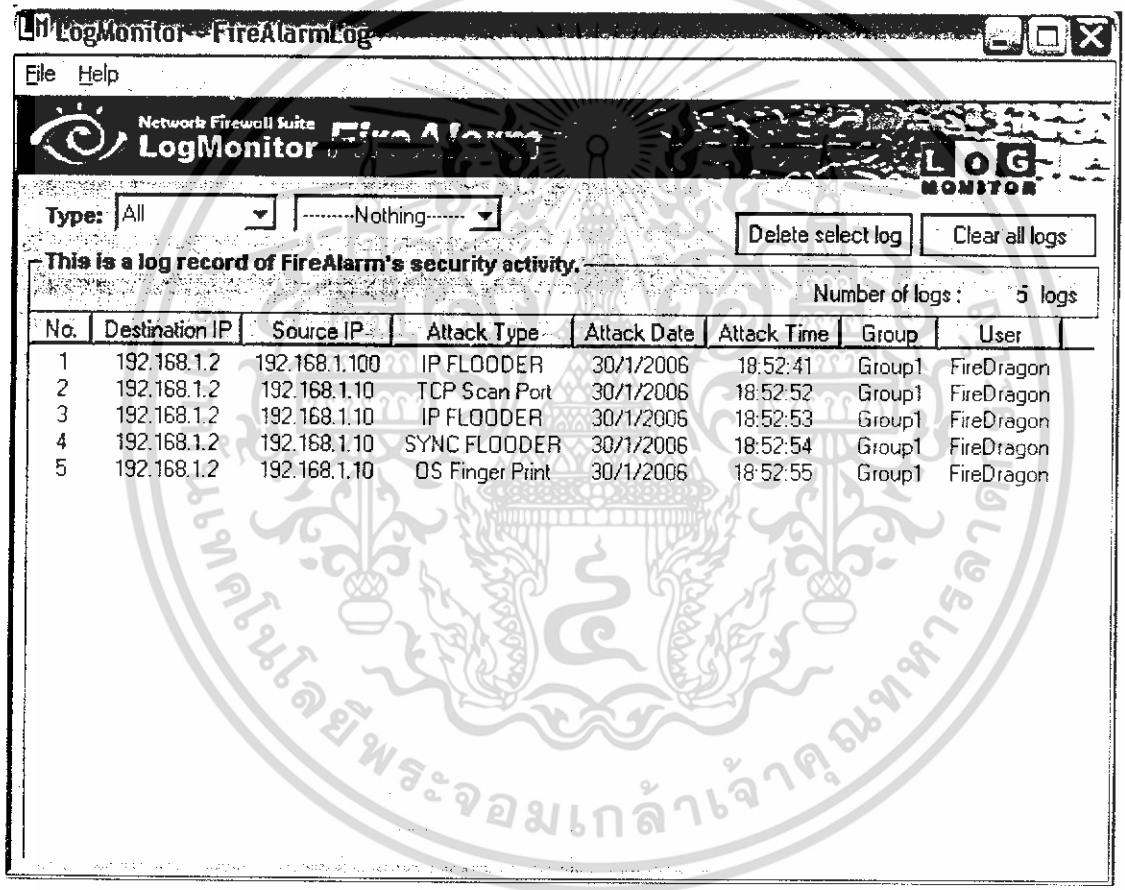
จากการเปิดใช้งานโปรแกรมทั้งสองนี้ทำให้สามารถเชื่อมต่อกับเครื่องแม่ข่ายได้และสามารถใช้งานไฟร์เบรกได้โดยมีความสามารถของการเป็นซีเคียวรีเวิร์สเว็บพร็อกซี่แล้ว ดังรูปที่ 9.14 ที่แสดงถึงการเปิดใช้งานโปรแกรม snort_inline ในโหมด verbose

รูปที่ 9.14 แสดงการทำงานของไฟร์เบรกขณะที่เปิดโปรแกรม snort_inline

ซึ่งเมื่อมีการร้องขอการใช้งานเว็บเซิร์ฟเวอร์ Snort_inline จะทำการตรวจสอบแพ็คเก็ตที่ผ่านเข้ามาถ้าไม่มีการผิดปกติก็จะทำการติดต่อไปยังเว็บเซิร์ฟเวอร์ที่อยู่ข้างหลังเครื่องรีเวิร์สเว็บพร็อกซี่นี้ เพื่อรับไฟล์มาตอบกลับให้กับไคลเอ็นต์ที่ทำการร้องขอการใช้งานเข้ามา แต่ถ้าแพ็คเก็ตที่ผ่านเข้ามานั้นมีความผิดปกติแล้ว Snort_inline สามารถตรวจสอบได้ก็จะมีการส่งล็อกไฟล์กลับมาแจ้งเตือนยังเครื่องแม่ข่ายและนำเก็บลงดาต้าเบส Snort ของ MySql และนำไปแสดงผลยังโปรแกรมล็อกมอนิเตอร์ของไฟร์เบรกต่อไป

9.3.5 ทดสอบการทำงานของล็อกมอนิเตอร์ไฟร์อลาร์ม (Log Monitor FireAlarm)

เมื่อโปรแกรมล็อกมอนิเตอร์ไฟร์อลาร์มเริ่มทำงานแล้ว จะมีลักษณะดังต่อไปนี้

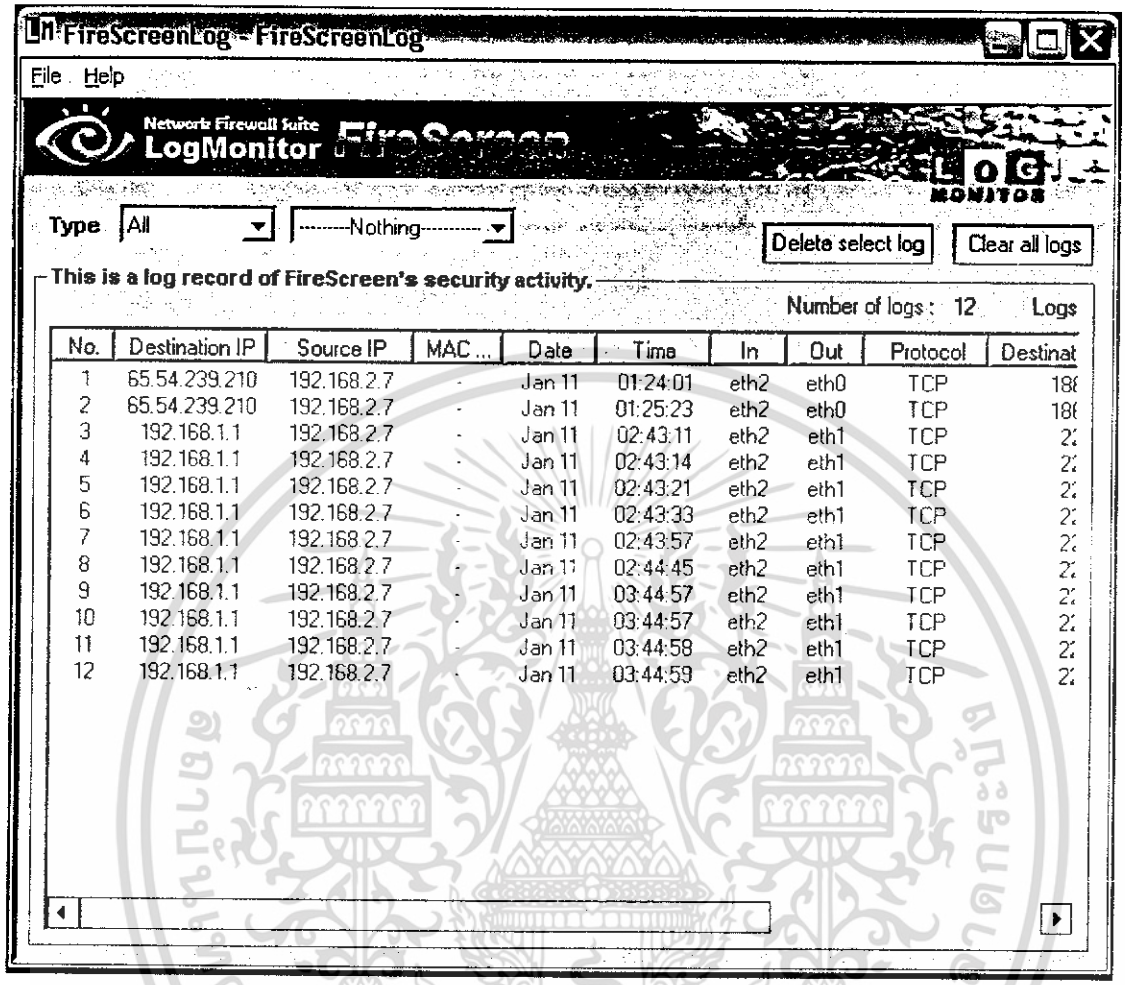


รูปที่ 9.15 โปรแกรมล็อกมอนิเตอร์ไฟร์อลาร์ม

จะสามารถเลือกดูบันทึกข้อมูลจากโปรแกรมไฟร์อลาร์มบนเครื่องไคลเอ็นต์ (Client) ที่ส่งมาทั้งหมดได้ ทั้งแบบเลือกดูแต่ละประเภทได้ สามารถลบได้

9.3.6 การทดสอบโปรแกรมล็อกมอนิเตอร์ไฟร์สกรีน (Log Monitor FireScreen)

เมื่อโปรแกรมล็อกมอนิเตอร์ไฟร์สกรีนเริ่มทำงานแล้ว จะมีลักษณะดังต่อไปนี้

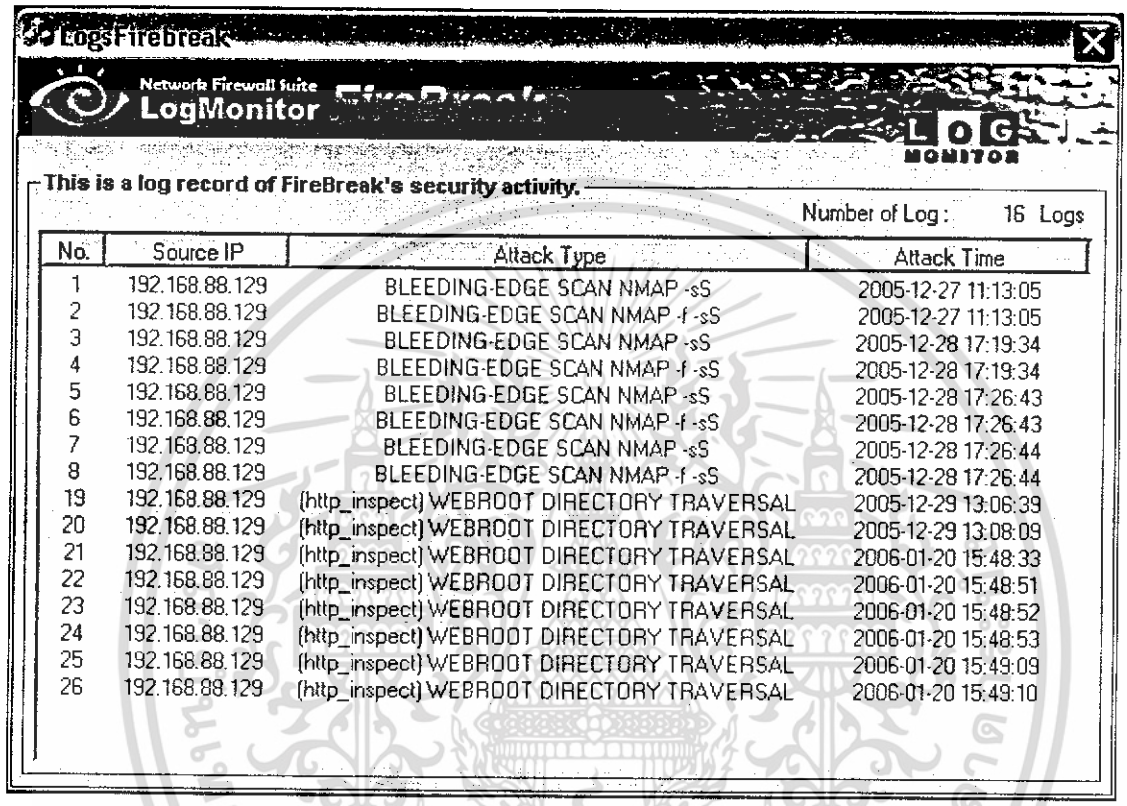


รูปที่ 9.16 โปรแกรมล็อกมอนิเตอร์ไฟร์สกรีน

จะสามารถเลือกดูบันทึกข้อมูลจากโปรแกรมไฟร์สกรีนบนเครื่องเกตเวย์ ที่ส่งมาทั้งหมดได้ ทั้งแบบเลือกดูแต่ละประเภทได้ สามารถลบได้

9.3.7 การทดสอบโปรแกรมล็อกมอนิเตอร์ไฟร์เบรก (Log Monitor FireBreak)

เมื่อโปรแกรมล็อกมอนิเตอร์ไฟร์เบรกเริ่มทำงานแล้ว จะมีลักษณะดังต่อไปนี้



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

จะสามารถเลือกดูบันทึกข้อมูลจากโปรแกรมไฟร์เบรกบนเครื่องซีเคียวรีเวิร์สเว็บพร็อกซี่ที่ส่งมาทั้งหมดได้ แต่ไม่สามารถทำการลบได้ ถ้าต้องการที่จะลบล็อกไฟล์ใด ก็จะต้องทำการผ่านโปรแกรมที่สามารถติดต่อกับ MySql ดาต้าเบสที่ชื่อ Snort ได้ โดยการสั่งด้วยคำสั่ง SQL

บรรณานุกรม

เรืองไกร รังสิพล 2544. **เจาะระบบ TCP/IP จุดอ่อนโปรโตคอลและวิธีป้องกัน** . พิมพ์ครั้งที่ 1 กรุงเทพฯ : บริษัท โปรวิชั่น จำกัด

เรืองไกร รังสิพล 2545. **เปิดโลก Firewall และ Internet Security** . พิมพ์ครั้งที่ 1 กรุงเทพฯ : บริษัท โปรวิชั่น จำกัด

สุรวัฒน์ ปุณณชัยยะ 2543. **เปิดโลกของ TCP/IP และโปรโตคอลของอินเตอร์เน็ต.** ครั้งที่ 1 กรุงเทพฯ : บริษัท โปรวิชั่น จำกัด

สุรศักดิ์ สงวนพงษ์ 2545. **สถาปัตยกรรมและโปรโตคอลทีซีพี/ไอพี** . พิมพ์ครั้งที่ 2 กรุงเทพฯ : บริษัท ซีเอ็ดยุเคชั่น จำกัด

Anthonh 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.