ระบบปรับแต่งกฎของไฟร์วอลล์โดยอัตโนมัติ

Firewall Automatic Configuration System

โดย

นายปณิธาน เชินอำนวย

รหัส 44067079

อาจารย์ที่ปรึกษา

ผศ.ดร. จันทร์บูรณ์ สถิตวิริยวงศ์

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
ภาคเรียนที่ 1 ปีการศึกษา 2546
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อหัวข้อ	ระบบปรับแต่งกฎของไฟร์วอลล์โดยอัตโนมัติ
นักศึกษา	นายปณิธาน เขินอำนวย
อาจารย์ที่ปรึกษา	ผศ.ดร. จันทร์บูรณ์ สถิตวิริยวงศ์
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2546

บทคัดย่อ

เนื่องจากไฟร์วอลล์เป็นอุปกรณ์ที่รักษาความปลอดภัยในเครือข่ายที่จำเป็น แต่ไฟร์วอลล์เป็นอุปกรณ์ที่ต้องมีการปรับแต่งระบบเพื่อให้งานตามความต้องการของระบบ และการปรับแต่งระบบต้องอาศัยบุคลากรที่มีความรู้ความสามารถ จึงได้พัฒนาระบบที่สามารถปรับแต่งไฟร์วอลล์ได้โดยอัตโนมัติเพื่อให้การใช้งานระบบไฟร์วอลล์ง่ายและสะดวกมากยิ่งขึ้น โดยจะทำการศึกษาถึงประสิทธิภาพในการปรับแต่งกฎของไฟร์วอลล์ที่ได้จากระบบที่ทำการพัฒนานี้ โดยคาดว่าผลที่ได้จากระบบที่ได้พัฒนาขึ้นมานี้คือระบบจะสามารถปรับแต่งกฎให้ผู้ใช้งานสามารถใช้ระบบเครือข่ายได้ตามความต้องการ โดยยังคงมีความปลอดภัยให้กับระบบเครือข่ายได้อย่างมี ประสิทธิภาพ

Title	Firewall Automatic Configuration System
Student	Mr. Panitan Kern-amnuay
Advisor	Asst. Prof. Dr. Chanboon Sathitwiriyawong
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2003

Abstract

Firewall is one of tools for network security, however it has to be adjusted to suit with system requirement. Since the firewall adjustment has to be proceeded properly by expert people, there have been developing a new automatic system for it. This new system will be able to adjust the firewall automatically to work conveniently and easily with the network system. And then, there will be a study of effectiveness of firewall rule adjustment from this developed system. The result of the new system will be expected to be a rule adjustment system, which users can work on the network conveniently. And also, the new system still has to provide security to the network system effectively.

กิตติกรรมประกาศ

โครงการพัฒนาระบบงานนี้ ผู้เขียนได้ทำการพัฒนาระบบจนเสร็จสมบูรณ์โดยได้รับความช่วยเหลือจากหลายท่าน โดยเฉพาะอย่างยิ่งท่านอาจารย์ที่ปรึกษา ,ท่านอาจารย์หลายๆท่าน รวมถึงหัวหน้างานที่คอยให้คำปรึกษา คำแนะนำ ข้อสังเกตและชี้แนะข้อบกพร่องต่างๆ รวมทั้งเพื่อนร่วมงานที่คอยแบ่งปันความรู้ความเข้าใจในเทคโนโลยีที่ใช้ในโครงการ เพื่อให้โครงการสำเร็จได้อย่างสมบูรณ์

ขอบคุณคุณพ่อ คุณแม่ ที่คอยให้คำปรึกษา กำลังใจ และความรักที่มีให้ ขอบคุณมิ้งค์สำหรับกำลังใจที่ให้และคอยเคียงข้างตลอด

สารบัญ

สารบัญ (ต่อ)

หน้า

สารบัญรูป

หน้า

รูปที่

สารบัญรูป (ต่อ)

หน้า

รูปที่

สารบัญตาราง

หน้า

ตารางที่

บทที่ 1

บทนำ

1.1 ความเป็นมาของโครงงาน

ในปัจจุบันระบบเครือข่ายคอมพิวเตอร์มีการใช้งานอย่างแพร่หลาย ในทุกๆที่ และเครือข่ายในแต่ละที่ก็ได้ทำการเชื่อมต่อถึงกันเพื่อแลกเปลี่ยนข้อมูลข่าวสารซึ่งกันและกัน เพื่อเพิ่มความสะดวกในการติดต่อสื่อสาร เมื่อมีการเชื่อมต่อเครือข่ายเข้าหากันมากขึ้น การควบคุมข้อมูลต่างๆ รวมถึงการเข้าถึงเครือข่ายจากผู้ที่ไม่ประสงค์ดี จากภายนอกจะควบคุมได้ยากขึ้น ดังนั้นจึงต้องมีอุปกรณ์ที่สามารถควมคุมการเข้าออกของข้อมูลในเครือข่ายได้ ซึ่งอุปกรณ์นั้นเรียกว่า ไฟร์วอลล์

ไฟร์วอลล์นั้น มีหน้าที่ควบคุมการเข้าออกของข้อมูลที่ผ่านตัวไฟร์วอลล์ โดยที่ไฟร์วอลล์จะมีการทำงานในลักษณะที่มีพื้นฐานจากกฎ คือมีการสร้างกฎของไฟร์วอลล์ขึ้นมาเพื่อบอกให้ไฟร์วอลล์รู้ว่าจะให้ข้อมูลแบบใด หรือการกระทำแบบใด ผ่านเข้าหรือออกไฟร์วอลล์ได้หรือไม่

โดยในปัจจุบันนั้นไฟร์วอลล์ได้มีการใช้งานกันอย่างแพร่หลายในระบบเครือข่ายทั่วไป มีทั้งไฟร์วอลล์ที่เป็น ฮาร์ดแวร์ และเป็น ซอฟต์แวร์ ซึ่งแต่ละแบบก็มีข้อดีข้อเสียแตกต่างกันออกไป และนอกจากนี้ยังมีทั้งไฟร์วอลล์ที่มีลิขสิทธิ์และไฟร์วอลล์ที่เป็นของฟรี อีกด้วย โดยทั้งสองแบบนี้ข้อแตกต่างหลักๆ ก็น่าจะอยู่ที่ความยากง่ายในการปรับแต่งระบบให้สามารถใช้งานได้ตามความต้องการ คือไฟร์วอลล์ที่มีขายตามท้องตลาดนั้น บริษัทเจ้าของผลิตภัณฑ์ก็ได้มีการออกแบบระบบการติดต่อกับผู้ใช้ และระบบตัวช่วยเหลือในการปรับแต่งกฎต่างๆให้ผู้ใช้สามารถใช้งานได้ง่ายอยู่แล้ว ยกตัวอย่างเช่น ไฟร์วอลล์ของบริษัท Checkpoint ซึ่งออกผลิตภัณฑ์ไฟร์วอลล์ที่ชื่อว่า Firewall-1 ซึ่งมีลักษณะการทำงานที่เป็นรูปภาพ เพื่อง่ายต่อการใช้งาน หรือผลิตภัณฑ์ของบริษัท Norton ที่มีไฟร์วอลล์ในชื่อของ Norton Personal Firewall ซึ่งก็มีระบบติดต่อกับผู้ใช้งานที่เป็นแบบรูปภาพเช่นเดียวกัน แต่ไฟร์วอลล์ที่เป็นของฟรีนั้น โดยส่วนมากแล้วจะมีระบบในการติดต่อกับผู้ใช้และคำสั่งที่ใช้ในการปรับแต่งกฎของไฟร์วอลล์ที่ยุ่งยากซับซ้อน ยกตัวอย่างเช่น IPTables ซึ่งเป็นไฟร์วอลล์ที่ทำงานบนระบบ Linux ซึ่งเป็นไฟร์วอลล์ที่ฟรี แต่ก็จะมีความลำบากและความยุ่งยากในการปรับแต่งกฎของไฟร์วอลล์ หรือ IPFW ที่เป็นไฟร์วอลล์ที่อยู่ในระบบปฏิบัติการ FreeBSD ก็เป็นไฟร์วอลล์ที่ฟรีเช่นเดียวกัน แต่ก็ยังมีความยุ่งยากในการปรับแต่งกฎเช่นเดียวกัน

นอกจากความยุ่งยากในการที่จะสร้างกฎต่างๆขึ้นมาเพื่อให้ระบบเครือข่ายมีความปลอดภัย ในขณะเดียวกับที่ยังสามารถใช้งานระบบเครือข่ายได้อย่างปกตินั้น ก็มีความลำบากและต้องการความรู้ความสามารถในเชิงเครือข่ายและระบบเป็นอย่างมาก เพื่อที่จะสามารถสร้างกฎของไฟร์วอลล์เพื่อให้ไฟร์วอลล์สามารถทำงานได้อย่างมีประสิทธิภาพ จึงได้มีการออกแบบระบบที่สามารถที่จะปรับแต่งไฟร์วอลล์ได้โดยอัตโนมัติ โดยที่ผู้ใช้งานไม่จำเป็นจะต้องมีความรู้ทางด้านเครือข่ายหรือระบบคอมพิวเตอร์มากนัก

ในอนาคตเครือข่ายคอมพิวเตอร์มีแนวโน้มที่จะขยายตัวออกไปอยู่ตามสถานที่ต่างๆไม่เพียงเฉพาะสำนักงานต่างๆเพียงเท่านั้น อาจจะมีการขยายออกไปอยู่ในที่พักอาศัย บ้านเรือนทุกๆหลัง แล้วเมื่อเป็นเช่นนั้น การดูแลความเป็นส่วนตัวของแต่ละเครือข่ายก็จะเป็นเรื่องที่ยากมากขึ้นยิ่งสำหรับเครือข่ายที่ไม่มีผู้ดูแลที่ดีพอ หรือเครือข่ายส่วนตัวที่อยู่ภายในที่พักอาศัยที่ไม่ต้องการดูแลระบบเครือข่ายแต่ยังต้องกาาความปลอดภัยและความเป็นส่วนตัว ก็จะสามารถใช้ระบบนี้เข้ามาช่วยได้เป็นอย่างดี

โครงการพัฒนาระบบงานนี้ แบ่งออกเป็น 2 ส่วนหลัก ส่วนที่ 1 เป็นการปรับแต่งและทดสอบระบบไฟร์วอลล์ด้วยซอฟต์แวร์ IPFW บนระบบปฏิบัติการ FreeBSD ส่วนที่ 2 เป็นการออกแบบและพัฒนาระบบปรับแต่งกฎของไฟร์วอลล์โดยอัตโนมัติโดยใช้โปรแกรมภาษา Perl ทำหน้าที่เป็นตัวควบคุมการเพิ่มและลดกฎของไฟร์วอลล์ โดยก่อนจะทำการเพิ่มหรือลดกฎนั้น จะมีการตรวจสอบ ถึงความถูกต้องของกฎที่จะสร้างด้วย และใช้โปรแกรมภาษา PHP เพื่อเป็นส่วนติดต่อกับผู้ใช้งาน โดยที่ผู้ใช้งานจะสามารถเพิ่มหรือลดเครื่องคอมพิวเตอร์ที่จะอนุญาตให้ผ่านเข้าออกไฟร์วอลล์ได้โดยทั้งหมดจะทำงานผ่านเวป โครงการพัฒนาระบบงานนี้ มีจุดประสงค์เพื่อศึกษาค้นคว้า ทดลองการใช้งานไฟร์วอลล์โดยที่ผู้ใช้ไม่จำเป็นต้องมีความรู้ทางด้านคอมพิวเตอร์มากนักก็จะสามารถใช้งานไฟร์วอลล์ได้อย่างมีประสิทธิภาพ เพื่อเป็นการส่งเสริมให้หน่วยงานที่ไม่มีบุคลากรที่มีความสามารถพอ สามารถที่จะใช้ไฟร์วอลล์ได้โดยไม่ต้องเสียค่าใช้จ่ายที่สูง

1.2 วัตถุประสงค์

1. เพื่อศึกษาค้นคว้าเรื่องพื้นฐานของระบบไฟร์วอลล์
2. เพื่อศึกษาทดลองการใช้งานไฟร์วอลล์ด้วยซอฟต์แวร์ IPFW บนระบบปฏิบัติการ FreeBSD
3. เพื่อเป็นการประหยัดค่าใช้จ่ายของค่าบุคลากรและค่าซอฟต์แวร์ ในการนำระบบไฟร์วอลล์มาใช้งานในระบบเครือข่าย

4. เพื่อพัฒนาระบบที่สามารถปรับแต่งกฎของไฟร์วอลล์ได้เองโดยอัตโนมัติเพื่อให้ผู้ที่ไม่มีความรู้ทางด้านคอมพิวเตอร์สามารถควบคุมไฟร์วอลล์ได้
5. เพื่อพัฒนาระบบที่ให้ผู้ใช้สามารถใช้งานระบบปรับแต่งกฎของไฟร์วอลล์โดยอัตโนมัติผ่านทางโปรโตคอล HTTP
6. เพื่อสนับสนุนการใช้งานระบบไฟร์วอลล์ให้แพร่หลายมากยิ่งขึ้น เนื่องจากปัจจุบันตามหน่วยงานต่างๆยังไม่มีความสามารถพอที่จะดูแลระบบไฟร์วอลล์ได้

1.3 ขอบเขตในการพัฒนาระบบงาน

1. ติดตั้งระบบปฏิบัติการ FreeBSD เพื่อเป็นระบบปฏิบัติการของระบบไฟร์วอลล์
2. ติดตั้งไฟร์วอลล์ IPFW บน FreeBSD เพื่อทำหน้าที่เป็นระบบไฟร์วอลล์
3. ติดตั้ง Web Server เพื่อเป็นส่วนติดต่อกับผู้ใช้ผ่านเวป
4. ติดตั้ง Database Server เพื่อเก็บข้อมูลที่จำเป็นของระบบ
5. พัฒนาโปรแกรมปรับแต่งกฎของไฟร์วอลล์โดยอัตโนมัติ โดยที่โปรแกรมมีความสามารถในการทำงานดังนี้
   - วิเคราะห์ Log file เพื่อนำค่าต่างๆมาสร้างเป็นกฎของไฟร์วอลล์
   - สร้างกฎของไฟร์วอลล์เองได้ ตามความต้องการของระบบ
   - สร้างกฎของไฟร์วอลล์โดยระบุได้ถึง IP Source ,Port Source ,IP Destination ,Port Destination ,Protocol ,Direction ,Interface
   - ลบกฎของไฟร์วอลล์ได้เอง เมื่อกฎๆ นั้น ไม่มีการใช้งานเกินระยะเวลาที่กำหนด
   - กำหนดสิทธิของเครื่องคอมพิวเตอร์ที่จะให้ระบบสร้างกฎให้อัตโนมัติได้
   - กำหนดระยะเวลาที่จะลบกฎที่ไม่ใช้ออกได้
6. พัฒนาโปรแกรมเพื่อควบคุมโปรแกรมปรับแต่งกฎของไฟร์วอลล์อัตโนมัติ โดยโปรแกรมมีความสามารถในการทำงานดังนี้
   - สามารถเพิ่มหรือลบ สิทธิในการสร้างกฎของเครื่องคอมพิวเตอร์ได้
   - สามารถเปลี่ยนระยะเวลาที่จะใช้ลบกฎที่ไม่ใช้ออกได้
   - ทำงานผ่านโพรโตคอล HTTP โดยมีหน้าจอการทำงานเป็นแบบรูปภาพผ่านโปรแกรม Web Browser

1.4 ทฤษฎีที่ใช้ในการพัฒนาระบบงาน

1. ศึกษาวิธีการติดตั้งแล้วการใช้งานซอฟต์แวร์ต่างๆ ที่ใช้สำหรับการสร้างระบบไฟร์วอลล์และระบบ Web Server เพื่อรองรับการติดต่อจากภายนอก ได้แก่ FreeBSD ,IPFW ,Apache Web Server ,PHP Extension ,MySQL ,Perl
2. ศึกษาเทคโนโลยีไฟร์วอลล์ IPFW บนระบบปฏิบัติการ FreeBSD
3. ศึกษาเทคโนโลยีการรักษาความปลอดภัยเครือข่ายโดยใช้ ไฟร์วอลล์
4. ศึกษาวิธีการสร้างกฎสำหรับไฟร์วอลล์ให้มีประสิทธิภาพ
5. ศึกษาการใช้งานโปรแกรมฐานข้อมูล MySQL เพื่อเก็บข้อมูลของระบบ
6. ศึกษาการใช้งานโปรแกรมภาษา Perl เพื่อพัฒนาโปรแกรมประยุกต์ ในส่วนของการทำงานกับระบบไฟร์วอลล์
7. ศึกษาการใช้งานโปรแกรมภาษา PHP Extension เพื่อพัฒนาโปรแกรมประยุกต์ ในส่วนของการติดต่อกับผู้ใช้งาน
8. ศึกษาเทคโนโลยี โพรโตคอลเครือข่าย TCP/IP เพื่อให้เข้าใจถึงรูปแบบของเฮดเดอร์

1.5 องค์ประกอบของระบบงาน

ระบบงานประกอบด้วยองค์ประกอบต่างๆ ดังต่อไปนี้

เครื่องคอมพิวเตอร์เซิร์ฟเวอร์ ระบบปฏิบัติการ FreeBSD Version 5.0 ทำหน้าที่

- ตรวจสอบข้อมูลที่วิ่งผ่านเข้าออกแล้วกระทำการต่อข้อมูลนั้นตามกฎที่กำหนดไว้ (Firewall)
- ติดตั้งโปรแกรมระบบปรับแต่งกฎของไฟร์วอลล์โดยอัตโนมัติเพื่อให้โปรแกรมสามารถปรับแต่งกฎของไฟร์วอลล์ได้โดยอัตโนมัติ
- เครื่องให้บริการเวปสำหรับเป็นส่วนติดต่อกับผู้ใช้งานเพื่อใช้ในการควบคุมระบบ

มีการติดตั้งซอฟต์แวร์เพื่อให้บริการดังนี้

- ซอฟต์แวร์ ไฟร์วอลล์ เลือกใช้ โปรแกรม IPFW
- ซอฟต์แวร์เวปเซิร์ฟเวอร์ที่รองรับมาตรฐาน SSL/TLS เลือกใช้ Apache เวอร์ชัน 2.047
- ซอฟต์แวร์ SSL/TLS เลือกใช้ OpenSSL เวอร์ชัน 0.9.6
- ซอฟต์แวร์ Database เลือกใช้ MySQL เวอร์ชัน 4.014
- ซอฟต์แวร์ภาษา Perl เลือกใช้ Perl เวอร์ชัน 5.8.0
- ซอฟต์แวร์ PHP Extension เลือกใช้ PHP เวอร์ชัน 4.3.2

เครื่องคอมพิวเตอร์ไคลเอ็นท์ ที่ใช้ควบคุมระบบปรับแต่งกฎของไฟร์วอลล์โดยอัตโนมัติ ใช้ระบบปฏิบัติการ Windows XP Professional ทำหน้าที่

- เครื่องสำหรับควบคุมระบบปรับแต่งกฎของไฟร์วอลล์โดยอัตโนมัติโดยผ่านเวปบราวเซอร์
- เครื่องไคลเอ็นท์ที่ใช้งานเครือข่ายโดยทำงานผ่านไฟร์วอลล์

มีการติดตั้งซอฟต์แวร์เพื่อให้บริการดังนี้

- โปรแกรมเวปบราวเซอร์ สำหรับติดต่อกับเวปเซิร์ฟเวอร์ เพื่อควบคุมโปรแกรมปรับแต่งกฎของไฟร์วอลล์อัตโนมัติ

1.6 ขั้นตอนในการพัฒนาระบบ

ประกอบไปด้วยขั้นตอนต่างๆ ดังนี้

1. การศึกษาความเป็นไปได้ในการพัฒนา

   เป็นการศึกษาความเป็นไปได้ในการพัฒนาโครงการ กำหนดขอบเขตของปัญหาและวางแผนวิธีการพัฒนาโปรแกรม รวมถึงกำหนดเป้าหมายในการพัฒนาโครงการ

2. การศึกษาและวิเคราะห์ทฤษฎีที่เลือกใช้

   ศึกษาอัลกอริทึมที่จะสามารถทำให้สามารถสร้างกฎของไฟร์วอลล์ขึ้นมาได้เองจากข้อมูลที่ผู้ใช้ระบบเครือข่ายต้องการ ศึกษาการทำงานของระบบไฟร์วอลล์ IPFW

3. การวิเคราะห์และออกแบบ

   ทำการวิเคราะห์และออกแบบโครงการพัฒนาระบบ

4. การพัฒนาและทดสอบ

   - ทำการติดตั้งระบบไฟร์วอลล์และทดสอบการทำงาน
   - ทำการพัฒนาโปรแกรมและทดสอบการทำงานในฟังก์ชันต่างๆ

5. การทดลองใช้งานและปรับปรุงแก้ไข

   นำโปรแกรมมาทดลองใช้งานและปรับปรุงแก้ไขเพื่อให้สามารถใช้งานได้ถูกต้องและง่ายยิ่งขึ้น

1.7 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้พัฒนาความรู้ความเข้าใจเรื่องการทำงานพื้นฐานของไฟร์วอลล์
2. ได้พัฒนาความรู้ความเข้าใจเรื่องการทำงานของไฟร์วอลล์ IPFW
3. ได้พัฒนาความรู้ ความสามารถในการวิเคราะห์ ออกแบบและพัฒนาระบบงาน และสามารถนำไปใช้ประโยชน์ต่อการทำงานในอนาคตได้
4. ได้โปรแกรมประยุกต์ที่ทำให้การใช้งานไฟร์วอลล์เป็นเรื่องง่ายที่ใครๆก็สามารถใช้งานได้
5. ช่วยส่งเสริมให้เกิดการใช้ไฟร์วอลล์แพร่หลายมากยิ่งขึ้น
6. เป็นอีกทางเลือกหนึ่งในการเลือกใช้โปรแกรมประยุกต์ที่ช่วยให้การทำงานกับไฟร์วอลล์เป็นเรื่องง่าย

เนื้อหาในบทต่างๆ มีดังต่อไปนี้

- บทที่ 2 กล่าวถึงทฤษฎีที่เกี่ยวข้องในการพัฒนาระบบ
- บทที่ 3 กล่าวถึงการติดตั้ง ปรับแต่ง และทดสอบใช้งานระบบไฟร์วอลล์
- บทที่ 4 กล่าวถึงการวิเคราะห์และออกแบบโปรแกรมระบบปรับแต่งกฎของไฟร์วอลล์โดยอัตโนมัติ
- บทที่ 5 กล่าวถึงการพัฒนาโปรแกรมระบบปรับแต่งกฎของไฟร์วอลล์โดยอัตโนมัติ
- บทที่ 6 กล่าวถึงบทสรุปและแนวทางการพัฒนาในอนาคต

บทที่ 2
ทฤษฎีที่เกี่ยวข้องในการพัฒนาระบบ

2.1 ทำความรู้จักกับไฟร์วอลล์

ไฟร์วอลล์นั้นก็จะเหมือนกับอุปกรณ์ต่างๆ ก็คือมีขีดความสามารถในการทำงานเหมือนดังเช่นอุปกรณ์ทั่วไป เราก็จะมาดูว่าไฟร์วอลล์นั้นมีความขีดความสามารถแค่ไหน โดยไฟร์วอลล์นั้นจะสามารถช่วยเพิ่มความปลอดภัยให้กับระบบได้โดย

- บังคับใช้นโยบายด้านความปลอดภัย โดยการกำหนดกฎให้กับไฟร์วอลล์ว่าจะอนุญาตหรือไม่อนุญาต การให้บริการแบบใด
- ทำให้การพิจารณาดูแลและตัดสินใจด้านความปลอดภัยของระบบให้เป็นได้ง่ายขึ้น เนื่องจากการติดต่อกับเครือข่ายภายนอกทุกชนิดจะต้องผ่านไฟร์วอลล์ การดูแลที่จุดนี้เป็นการดูแลความปลอดภัยในระดับเครือข่าย
- บันทึกข้อมูลของกิจกรรมต่างๆที่ผ่านเข้าออกเครือข่ายได้อย่างมีประสิทธิภาพ
- ป้องกันเครือข่ายบางส่วนจากการเข้าถึงของเครือข่ายภายนอก เช่นถ้าหากเรามีเครือข่ายบางส่วน ที่ต้องการให้ภายนอกเข้ามาใช้บริการ แต่บางส่วนไม่ต้องการให้เครือข่ายภายนอกใช้บริการก็จะสามารถให้ไฟร์วอลล์ช่วยได้
- ไฟร์วอลล์บางชนิด สามารถป้องกันไวรัสได้ โดยจะทำการตรวจสอบไฟล์ที่ทำการโอนย้ายผ่านทางโปรโตคอล HTTP, FTP และ SMTP
- ที่ผ่านมาคือความสามารถของโดยรวมของไฟร์วอลล์ว่าไฟร์วอลล์สามารถทำอะไรได้บ้าง ถึงแม้ว่าไฟร์วอลล์จะสามาถช่วยเพิ่มความปลอดภัยให้กับเครือข่ายได้อย่างมากโดยการตรวจดูข้อมูลที่ผ่านเข้าออกเครือข่าย แต่ก็อย่าลืมว่าระบบไฟร์วอลล์ก็มีขีดความสามารถเหมือนกัน ต่อมาเราจะมากล่าวถึงว่า อะไรอยู่นอกเหนือความสามารถของไฟร์วอลล์
- อันตรายที่เกิดจากเครือข่ายภายใน ไฟร์วอลล์จะไม่สามารถป้องกันอันตรายที่เกิดขึ้นจากภายในเครือข่ายเอง เพราะว่าไม่ได้ผ่านไฟร์วอลล์เข้ามา
- อันตรายจากภายนอกที่ไม่ได้ผ่านไฟร์วอลล์เข้ามา

- อันตรายจากวิธีใหม่ๆ ที่เกิดขึ้น ทุกวันนี้มีการพบช่องโหว่ใหม่ๆ เกิดขึ้นทุกวัน เราไม่สามารถจะไว้ใจไฟร์วอลล์โดยการติดตั้งเพียงครั้งเดียวแล้วก็หวังให้มันทำงานได้อย่างปลอดภัยตลอดไป เราจะต้องมีการดูแลรักษาอย่างต่อเนื่องสม่ำเสมอ
- ไวรัส ถึงแม้จะมีไฟร์วอลล์บางชนิดที่สามารถป้องกันได้ แต่ก็ยังไม่มีไฟร์วอลล์ชนิดใดที่สามารถตรวจสอบไวรัสได้ในทุกๆ โปรโตคอล
- เมื่อเราได้รับรู้ถึงขีดความสามารถของไฟร์วอลล์ว่าอะไรที่ไฟร์วอลล์ทำให้เราได้และอะไรที่ไฟร์วอลล์ไม่สามารถทำได้ แล้ว จากนี้เราก็จะมาดูว่าไฟร์วอลล์มีอยู่ด้วยกัน 3 ชนิด
    - PACKET FILTERING
    - PROXY SERVICE
    - STATEFUL INSPECTION

โดยในแต่ละแบบก็จะมีความสามารถต่างกันไปดังนี้

- Packet filtering คือเราเตอร์ที่ทำการค้นหาเส้นทางและส่งต่อ อย่างมีเงื่อนไข โดยจะทำการพิจารณาข้อมูลที่อยู่ในส่วนหัวของ Packet ที่วิ่งผ่านเข้ามา และนำไปเปรียบเทียบกับกฎ ที่กำหนดไว้ และตัดสินว่าควรจะทิ้ง Packet หรือว่าจะยอมให้ Packet นั้นผ่านไปได้ ในการพิจารณาส่วนหัวของ Packet นั้นจะเป็นการตรวจสอบในระดับชั้นของ อินเทอร์เน็ตเลเยอร์ (internet Layer) และในชั้นของ ทรานสปอร์ตเลเยอร์ (transport layer) ในระบบอินเทอร์เน็ตโมเดล ในการพิจารณาส่วนหัวของ Packet นั้นจะตรวจสอบในระดับของอินเทอร์เน็ตเลเยอร์ และมีส่วนประกอบสำคัญที่เกี่ยวข้องต่อ Packet filtering ดังนี้ ไอพีต้นทาง ,ไอพีปลายทาง ,ชนิดของโปรโตคอล และส่วนประกอบสำคัญในชั้นของทรานสปอร์ตเลเยอร์ที่เกี่ยวข้องต่อ packet filtering ดังนี้ พอร์ตต้นทาง พอร์ตปลายทาง แฟล็ก ชนิดของ icmp message โดย ไฟร์วอลล์แบบนี้จะนำส่วนประกอบต่างๆที่จับได้จากข้อมูลที่ผ่านเข้ามาแล้วนำมาพิจารณาเปรียบเทียบกับกฎที่มีอยู่ โดยไฟร์วอลล์แบบ packet filtering นั้นมีข้อดี ข้อเสียดังนี้ ข้อดี ไม่ขึ้นกับแอพพลิเคชั่น มีความเร็วสูง รองรับการขยายตัวได้ดี แต่ก็มีข้อเสียคือไม่เหมาะสมกับการใช้ในบางโปรโตคอลเช่น icq ,ftp
- Proxy Service มาถึงไฟร์วอลล์ในแบบที่สองคือ Proxy หรือ Application Gateway เป็นแอพพลิเคชันโปรแกรมที่ทำงานอยู่บนไฟร์วอลล์ที่ตั้งอยู่ระหว่างเน็ตเวิร์ก 2 เน็ตเวิร์ค ทำหน้าที่เพิ่มความปลอดภัยของระบบเน็ตเวิร์กโดยการควบคุมการเชื่อมต่อระหว่างเน็ตเวิร์กภายในและภายนอก Proxy จะช่วยเพิ่มความปลอดภัยได้มากเนื่องจากมีการตรวจสอบข้อมูลถึงในระดับของแอพพลิเคชันเลเยอร์ (Application Layer) เมื่อไคลเอนต์ต้องการใช้เซอร์วิสภายนอก ไคลเอนต์จะทำการติดต่อไปยัง Proxy ก่อน ไคลเอนต์จะเจรจา (negotiate) กับ

Proxy เพื่อให้ Proxy ติดต่อไปยังเครื่องปลายทางให้ เมื่อ Proxy ติดต่อไปยังเครื่องปลายทางให้แล้วจะมีการเชื่อมต่อ (connection) 2 การเชื่อมต่อ คือ ไคลเอนต์กับ Proxy และ Proxy กับเครื่องปลายทาง โดยที่ Proxy จะทำหน้าที่รับข้อมูลและส่งต่อข้อมูลให้ใน 2 ทิศทาง ทั้งนี้ Proxy จะทำหน้าที่ในการตัดสินใจว่าจะให้มีการเชื่อมต่อกันหรือไม่ จะส่งต่อแพ็กเกจให้หรือไม่ ข้อดี ข้อเสียของไฟร์วอลล์แบบ Proxy มีดังนี้ ข้อดี มีความปลอดภัยสูง , รู้จักข้อมูลในแบบแอพพลิเคชั่น ส่วนข้อเสียของไฟร์วอลล์ในระบบนี้ก็คือ ประสิทธิภาพต่ำ ,แต่ละบริการมักจะต้องการโปรเซสของตนเอง และยังสามารถจะขยายตัวได้ยาก

- Stateful inspection มาถึงไฟร์วอลล์ในแบบที่ 3 โดยปกติแล้ว Packet Filtering แบบธรรมดา (ที่เป็น Stateless แบบที่มีอยู่ในเราเตอร์ทั่วไป) จะควบคุมการเข้าออกของแพ็กเก็ตโดยพิจารณาข้อมูลจากเฮดเดอร์ของแต่ละแพ็กเก็ต นำมาเทียบกับกฎที่มีอยู่ ซึ่งกฎที่มีอยู่ก็จะเป็นกฎที่สร้างจากข้อมูลส่วนที่อยู่ในเฮดเดอร์เท่านั้น ดังนั้น Packet Filtering แบบธรรมดาจึงไม่สามารถทราบได้ว่า แพ็กเก็ตนี้อยู่ส่วนใดของการเชื่อมต่อ เป็นแพ็กเก็ตที่เข้ามาติดต่อใหม่หรือเปล่า หรือว่าเป็นแพ็กเก็ตที่เป็นส่วนของการเชื่อมต่อที่เกิดขึ้นแล้ว เป็นต้นStateful Inspection เป็นเทคโนโลยีที่เพิ่มเข้าไปใน Packet Filtering โดยในการพิจารณาว่าจะยอมให้แพ็กเก็ตผ่านไปนั้น แทนที่จะดูข้อมูลจากเฮดเดอร์เพียงอย่างเดียว Stateful Inspection จะนำเอาส่วนข้อมูลของแพ็กเก็ต (message content) และข้อมูลที่ได้จากแพ็กเก็ตก่อนหน้านี้ที่ได้ทำการบันทึกเอาไว้ นำมาพิจารณาด้วย จึงทำให้สามารถระบุได้ว่าแพ็กเก็ตใดเป็นแพ็กเก็ตที่ติดต่อเข้ามาใหม่ หรือว่าเป็นส่วนหนึ่งของการเชื่อมต่อที่มีอยู่แล้ว โดยทั่วไปแล้วไฟร์วอลล์ที่นิยมใช้กันอย่างแพร่หลายในปัจจุบันก็คือ Stateful Inspection เพราะเป็นเทคโนโลยีที่มีความสามารถที่จะตรวจกรองข้อมูลได้เป็นอย่างนี้ ซึ่งไฟร์วอลล์ ที่เป็นแบบ Stateful Inspection ที่นิยมใช้กันยกตัวอย่างได้เช่น Checkpoint Firewall-1 ,Cisco Secure Pix Firewall ซึ่งเป็น เป็นสินค้าที่มีขายในท้องตลาด และนอกจากนี้ยังมีไฟร์วอลล์ที่เป็น Open Source อีกด้วยอย่างเช่น Netfilter ซึ่งมีอยู่ใน Linux และมี ipfw ที่อยู่ใน FreeBSD โดยที่ไฟร์วอลล์ในแต่ละแบบนั้นก็จะมีรูปแบบในการควบคุมและปรับแต่งกฎของไฟร์วอลล์นั้นแตกต่างกันออกไป ซึ่งจะต้องใช้ผู้เชี่ยวชาญที่มีความสามารถเพื่อทำการปรับแต่งกฎ ยิ่งถ้าเป็นไฟร์วอลล์ในแบบที่เป็น Open Source แล้วก็จะมีความยากในการปรับแต่งกฎมากกว่าไฟร์วอลล์ที่มีขายอยู่ทั่วไป เพราะว่าไฟร์วอลล์ที่เป็น Open Source นั้นส่วนมากจะมี Interface ที่ใช้ติดต่อกับผู้ใช้เป็นแบบ command line คือผู้ใช้จะต้องพิมพ์คำสั่งเข้าไปเพื่อปรับแต่งกฎของไฟร์วอลล์ และหนึ่งในไฟร์วอลล์แบบ

Open Source นั้นก็คือ IPFW ที่อยู่ใน FreeBSD ที่ต้องใช้คำสั่งที่เป็น command line เพื่อปรับแต่งกฎของไฟร์วอลล์ ซึ่งมีรูปแบบและคำสั่งในการใช้งานดังนี้

2.2 เตรียมตัวก่อนใช้งาน IPFW

การใช้งาน ipfw นั้นจะมีส่วนประกอบที่อยู่ภายใน เคอร์เนล ของระบบปฏิบัติการ ดังนั้นแล้วเราจะต้องทำการปรับแต่ง เคอร์เนล ของระบบโดยจะต้องเพิ่มตัวเลือกเข้าไปใน เคอร์เนล โดยที่ไฟล์ เคอร์เนล ของ FreeBSD นั้นจะอยู่ที่ /usr/src/sys/i386/conf/GENERIC และจำต้องทำการคัดลอกไฟล์ GENERIC เป็นไฟล์ชื่ออื่นแล้วใส่ตัวเลือกเพิ่ม ดังนี้

options IPFIREWALL เป็นการบอกให้ เคอร์เนล รองรับการทำงานของไฟร์วอลล์

options IPFIREWALL_VERBOSE จะเป็นตัวเลือกที่ช่วยให้ เคอร์เนล สามารถเก็บกิจกรรมที่เกิดขึ้นกับไฟร์วอลล์ลง log file ในแบบ syslog ได้

options IPFIREWALL_VERBOSE_LIMIT=X จะเป็นการป้องกัน log file เต็มเนื่องจากมีข้อมูลที่ไม่ต้องการมากเกินไปโดยที่ X จะแทนตัวเลขที่เป็นจำนวนของกิจกรรมที่ซ้ำกันโดยจะเก็บแค่จำนวนมากสุดคือ X

ด้วยตัวเลือกทั้งสามตัวที่เพิ่มเข้าไปใน เคอร์เนล จะเป็นในส่วนที่ทำให้ระบบสามารถใช้งาน ipfw ได้และเมื่อทำการเพิ่มตัวเลือกเข้าไปแล้วก็จะต้องทำให้ระบบรับรู้ถึง เคอร์เนล ใหม่โดยมีขั้นตอนดังนี้

1. config KERNEL โดยที่ KERNEL คือชื่อของไฟล์ เคอร์เนล ที่ได้เพิ่มตัวเลือกเข้าไปแล้ว
2. ทำการเปลี่ยน Directory ไปยังที่ /usr/src/sys/compile/KERNEL
3. make depend; make เพื่อทำการ คอมไพล์ เคอร์เนล ใหม่
4. make install เพื่อทำการติดตั้ง เคอร์เนล ใหม่เข้าใจกับระบบ
5. แล้วทำการ reboot ระบบใหม่ก็เป็นอันเสร็จการติดตั้ง เคอร์เนล ใหม่

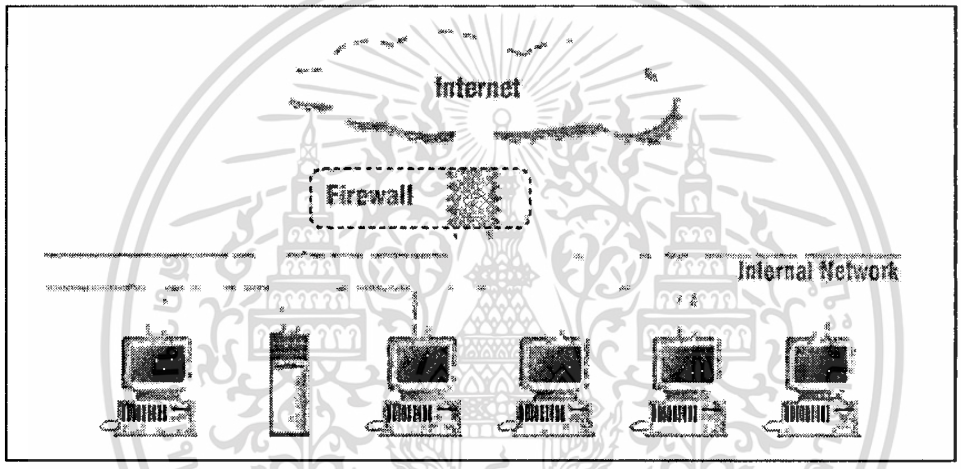
หลังจากที่ได้ เคอร์เนล ที่พร้อมจะใช้กับ ipfw แล้วเราก็จะมาศึกษาเกี่ยวกับ Firewall ที่ชื่อ IPFW บนระบบปฏิบัติการ FreeBSD

2.3 สถาปัตยกรรมไฟร์วอลล์

ในส่วนของ Firewall Architecture นั้น จะพูดถึงการจัดวางไฟร์วอลล์คอมโพเน็นต์ในแบบต่างๆ เพื่อทำให้เกิดเป็นระบบไฟร์วอลล์ขึ้น

Single Box Architecture

Single Box Architecture เป็น Architecture แบบง่ายๆ ที่มีคอมโพเน็นต์ทำหน้าที่เป็นไฟร์วอลล์เพียงอันเดียวตั้งอยู่ระหว่างเน็ตเวิร์กภายในกับเน็ตเวิร์กภายนอก ข้อดีของวิธีนี้ก็คือการที่มีเพียงจุดเดียวที่หน้าที่ไฟร์วอลล์ทั้งหมด ควบคุมการเข้าออกของข้อมูล ทำให้ดูแลได้ง่าย เป็นจุดสนใจในการดูแลความปลอดภัยเน็ตเวิร์ก ในทางกลับกันข้อเสียของวิธีนี้ก็คือ การที่มีเพียงจุดเดียวนี้ ทำให้มีความเสี่ยงสูง หากมีการคอนฟิกูเรชันผิดพลาดหรือมีช่องโหว่เพียงเล็กน้อย การผิดพลาดเพียงจุดเดียวอาจทำให้ระบบถูกเจาะได้



รูปที่ 2.1 Firewall Architecture แบบชั้นเดียว

คอมโพเน็นต์ที่ใช้ใน Architecture นี้อาจเป็น Screening Router , Dual-Homed Host หรือ Multi-purposed Firewall Box ก็ได้

Screening Router

เราสามารถใช้เราเตอร์ทำ Packet Filtering ได้ วิธีนี้จะทำให้ประหยัดค่าใช้จ่ายเนื่องจากส่วนใหญ่จะใช้เราเตอร์ต่อกับเน็ตเวิร์กภายนอกอยู่แล้ว แต่วิธีนี้อาจไม่ยืดหยุ่นมากนักในการคอนฟิกกูเรชัน Architecture แบบนี้เหมาะสำหรับ

- เน็ตเวิร์คที่มีการป้องกันความปลอดภัยในระดับของโฮสต์ (Host security) เป็นอย่างดีแล้ว
- มีการใช้โปรโตคอลไม่มาก และโปรโตคอลที่ใช้ก็เป็นโปรโตคอลที่ไม่ซับซ้อน
- ต้องการไฟร์วอลล์ที่มีความเร็วสูง

Dual-Homed Host

เราสามารถใช้ Dual-Homed Host (คอมพิวเตอร์ที่มีเน็ตเวิร์คอินเตอร์เฟสอย่างน้อย 2 อัน) ใช้การบริการเป็น Proxy ให้กับเครื่องภายในเน็ตเวิร์ก Architecture แบบนี้เหมาะสำหรับ

- เน็ตเวิร์คที่มีการใช้งานอินเตอร์เน็ตค่อนค่างน้อย
- เน็ตเวิร์คที่ไม่ได้มีข้อมูลสำคัญๆ

Multi-purposed Firewall Box

มีผลิตภัณฑ์หลายชนิดที่ผลิตออกมาเป็นกล่องๆ เดียว ซึ่งทำหน้าที่ได้หลายอย่าง ทั้ง Packet Filtering, Proxy Server แต่ก็อย่าลืมว่านี่คือ Architecture แบบชั้นเดียว ซึ่งถ้าพลาดแล้วก็จะเสียหายทั้งเน็ตเวิร์กได้

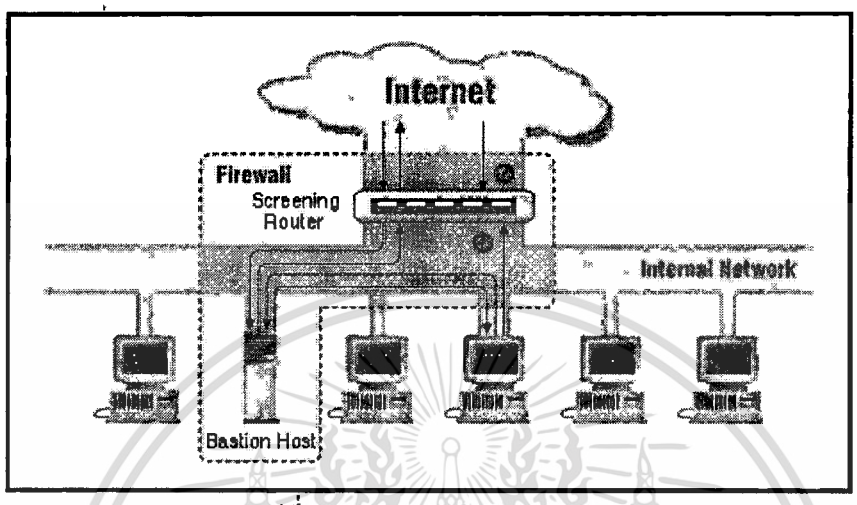
Screened Host Architecture

Screened Host Architecture จะมีโฮสต์ซึ่งให้บริการ Proxy เหมือนกับใน Single Box Architecture ที่เป็น Dual-homed Host แต่จะต่างกันตรงที่ว่า โฮสต์นั้นจะอยู่ภายในเน็ตเวิร์ก ไม่ต่ออยู่กับเน็ตเวิร์กภายนอกอื่นๆ (ดังนั้นก็ไม่จำเป็นที่จะต้องใช้ Dual Homed Host) และจะมี เราเตอร์ที่ทำหน้าที่ Packet Filtering ช่วยบังคับให้เครื่องภายในเน็ตเวิร์คต้องติดต่อเซอร์วิสผ่าน Proxy โดยไม่ยอมให้ติดต่อใช้เซอร์วิสจากภายนอกโดยตรง และก็ให้ภายนอกเข้าถึงได้เฉพาะ Bastion host (คือโฮสต์ที่มีความเสี่ยงสูงต่อการถูกโจมตี มักจะเป็นโฮสต์ที่เปิดให้บริการกับอินเตอร์เน็ต ดังนั้นโฮสต์นี้ต้องมีการดูแลเป็นพิเศษ) เท่านั้น

จากรูปที่ 2.2 ใน Architecture แบบนี้จะประกอบไปด้วยเราเตอร์ทำหน้าที่ Packet Filtering และภายในเน็ตเวิร์กจะมี Bastion Host ให้บริการ Proxy อยู่ โดยที่เราเตอร์นั้นอาจจะถูกเซ็ตดังนี้

- อาจจะอนุญาตให้เครื่องภายในใช้เซอร์วิสบางอย่างได้โดยตรง
- ส่วนเซอร์วิสอื่นๆ จะไม่ยอมให้เครื่องภายในติดต่อผ่านออกไปโดยตรง ยกเว้น Bastion Host เท่านั้นที่สามารถติดต่อกับเน็ตเวิร์กภายนอกได้ทั้งนี้เพื่อเป็นการบังคับให้ใช้บริการ Proxy ผ่านทาง Bastion Host เท่านั้น

หรืออาจจะเซ็ตให้เซอร์วิสส่วนใหญ่ผ่านเราเตอร์ออกไปได้โดยตรงแล้ว ให้บางส่วนต้องใช้เซอร์วิสผ่าน Proxy ก็แล้วแต่นโยบายและความเหมาะสมขององค์กร



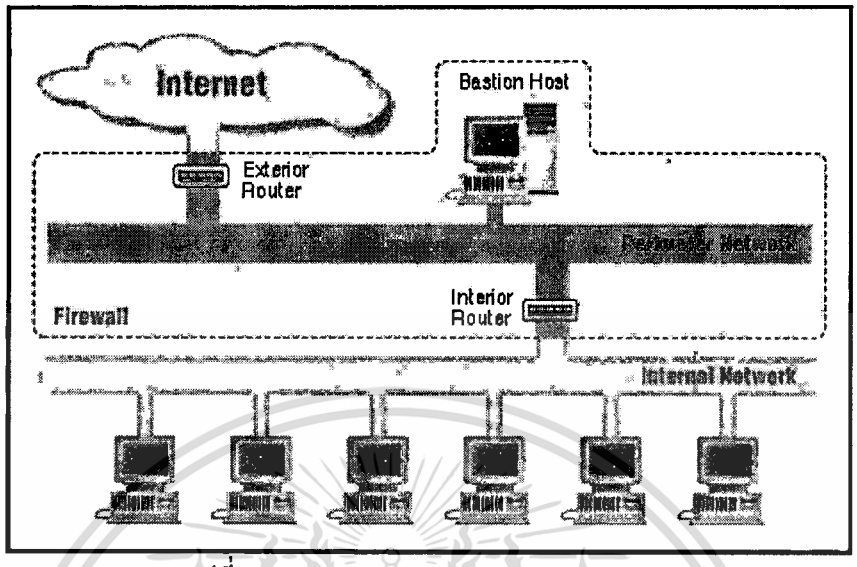
รูปที่ 2.2 Screened Host Architecture

วิธีนี้ถึงแม้ว่าจะมีทั้ง Proxy และเราเตอร์ทำหน้าที่ Packet Filtering แต่ก็ยังคงอันตรายอยู่ เพราะว่าเราเตอร์ต้องยอมให้ภายนอกสามารถติดต่อกับ Bastion Host ได้อยู่แล้ว หากแฮกเกอร์สามารถเจาะเข้ามายัง Bastion Host ได้ก็เสร็จ Architecture นี้เหมาะสำหรับ

- เน็ตเวิร์คที่มีการติดต่อกับเน็ตเวิร์กภายนอกน้อย
- เน็ตเวิร์กที่มีการป้องกันความปลอดภัยในระดับของโฮสต์เป็นอย่างดีแล้ว

Multi Layer Architecture

ในสถาปัตยกรรมแบบหลายชั้น ไฟร์วอลล์จะเกิดขึ้นจากคอมโพเน็นต์หลายๆส่วนทำหน้าที่ประกอบกันขึ้นเป็นระบบ วิธีการนี้สามารถเพิ่มความปลอดภัยได้มาก เนื่องจากเป็นการลดความเสี่ยงต่อความผิดพลาดที่อาจเกิดขึ้น ถ้าหากมีไฟร์วอลล์เพียงจุดเดียวแล้วมีเกิดความผิดพลาดเกิดขึ้นระบบทั้งหมดก็จะเป็นอันตราย แต่ถ้ามีการป้องกันหลายชั้น หากในชั้นแรกถูกเจาะ ก็อาจจะมีความเสียหายเพียงบางส่วน ส่วนที่เหลือระบบก็ยังคงมีชั้นอื่นๆ ในการป้องกันอันตราย และยังลดความเสี่ยงได้โดยการที่แต่ละชั้นนั้นมีการใช้เทคโนโลยีที่แตกต่างกัน เพื่อให้เกิดความหลากหลาย เป็นการหลีกเลี่ยงการโจมตีหรือช่องโหว่ที่อาจมีในเทคโนโลยีชนิดใดชนิดหนึ่ง โดยทั่วไปแล้วสถาปัตยกรรมแบบหลายชั้นจะเป็นการต่อกันเป็นซีรี่โดยมี Perimeter Network (หรือบางทีเรียกว่า DMZ Network) อยู่ตรงกลาง เรียกว่า Screened Subnet Architecture



รูปที่ 2.3 Screened Subnet Architecture

Screened Subnet Architecture

Screened Subnet Architecture เป็นสถาปัตยกรรมที่มีการเพิ่ม Perimeter Network เข้าไปกั้นระหว่างอินเตอร์เน็ตกับเน็ตเวิร์กภายในไม่ให้เชื่อมต่อกันโดยตรง ทำให้เน็ตเวิร์กภายในมีความปลอดภัยมากขึ้น ในรูปที่ 2.3 แสดง Screened Subnet Architecture อย่างง่าย ประกอบไปด้วย เราเตอร์ 2 ตัว ตัวนึงอยู่ระหว่างอินเตอร์เน็ตกับ Perimeter Network ส่วนอีกตัวหนึ่งอยู่ระหว่าง Perimeter Network กับเน็ตเวิร์กภายใน ถ้าหากแฮกเกอร์จะเจาะเน็ตเวิร์กภายในต้องผ่านเราเตอร์เข้ามาถึง 2 ตัวด้วยกัน ถึงแม้ว่าจะเจาะชั้นแรกเข้ามายัง Bastion host ได้ แต่ก็ยังต้องผ่านเราเตอร์ตัวในอีก ถึงจะเข้ามายังเน็ตเวิร์กภายในได้ คอมโพเน็นต์ของ Screened Subnet Architecture ในรูปที่ 2. 3

- Perimeter Network เป็นเน็ตเวิร์คที่เพิ่มเข้ามาเพื่อความปลอดภัย อยู่ระหว่างเน็ตเวิร์กภายนอกกับเน็ตเวิร์กภายใน ประโยชน์ของ Perimeter Network ที่เห็นได้ชัดก็คือการแบ่งเน็ตเวิร์คกออกเป็นส่วนๆ ทำให้การไหลของข้อมูลถูกแบ่งออกเป็นส่วนๆตามเน็ตเวิร์กด้วย เนื่องจากโดยทั่วไปแล้ว เน็ตเวิร์คที่เป็นแลนนั้น จะเป็นแบบ Ethernet ซึ่งจะมีการส่งข้อมูลแบบ Broadcast ดังนั้นถ้ามีใครคอบดักจับข้อมูลอยู่ในเน็ตเวิร์คนั้น ก็จะได้พาสเวิร์ด ข้อมูลต่างๆ ไปหมด ดังนั้นหากไฟร์วอลล์เรามีชั้นเดียวและแฮกเกอร์สามารถเข้ามาได้ โดนดักจับข้อมูลก็เสร็จหมด แต่ถ้าเรามี Perimeter Network ถึงจะดักจับข้อมูลได้แต่ก็จะได้เพียงที่อยู่บน Perimeter Network เท่านั้น

- Bastion Host ตั้งอยู่บน Perimeter Network ทำหน้าที่ให้บริการ Proxy กับเน็ตเวิร์กภายใน และให้บริการต่างๆ กับผู้ใช้บนอินเตอร์เน็ต Bastion Host นั้นจะมีความเสี่ยงต่อการโจมตีสูง จึงต้องมีการดูแลความปลอดภัยเป็นพิเศษ
- Interior Router ตั้งอยู่ระหว่าง Perimeter Network กับเน็ตเวิร์กภายใน ทำหน้าที่ Packet Filtering ป้องกันเน็ตเวิร์กภายในจาก Perimeter Network ในการเซ็ต configuration ระหว่าง เน็ตเวิร์กภายในกับ Perimeter Network ควรกำหนดอย่างรอบคอบ อนุญาตเฉพาะเซอร์วิสที่จำเป็นเท่านั้นอย่างเช่น DNS, SMTP
- Exterior Router ตั้งอยู่ระหว่างเน็ตเวิร์กภายนอกกับ Perimeter Network เนื่องจาก Exterior Router นี้เป็นจุดที่ต่ออยู่กับเน็ตเวิร์กภายนอก จึงมีหน้าที่ที่สำคัญอย่างหนึ่งคือการป้องกันแพ็กเก็ตที่มีการ Forged IP Address เข้ามา โดยอ้างว่ามาจากเน็ตเวิร์กภายในทั้งๆ ที่จริงๆ แล้วมาจากเน็ตเวิร์กภายนอก

2.4 การสร้างกฎสำหรับไฟร์วอลล์

เป็นที่ทราบกันดีอยู่แล้วว่า ไฟร์วอลล์มีหน้าที่หลักในการกรอง (filter) ข้อมูลเฉพาะส่วนที่ได้รับอนุญาตเท่านั้น ดังนั้นการเขียนกฎหรือ rule สำหรับไฟร์วอลล์จึงเป็นเรื่องที่สำคัญอย่างยิ่ง การสร้าง rule ของไฟร์วอลล์ที่ผิดพลาดจะทำให้ไฟร์วอลล์ (ทั้งราคาแพงและใช้งานฟรี) ทั้งหลายไม่สามารถช่วยป้องกันเครือข่ายให้รอดพ้นจากการถูกบุกรุกหรือโจมตีได้อย่างแน่นอน แต่ก่อนอื่น ผู้ดูแลไฟร์วอลล์จะต้องมั่นใจว่าเครื่องไฟร์วอลล์นั้นมีความปลอดภัยในระดับโฮสต์อยู่แล้ว (host based security) เพราะถึงแม้ว่า rule ที่สร้างขึ้นจะสามารถป้องกันเครื่องอื่นๆ ภายในเครือข่ายได้ แต่ถ้าเครื่องไฟร์วอลล์เองไม่สามารถทนต่อการบุกรุกได้ก็เป็นจุดที่อันตรายไม่ยิ่งหย่อนไปกว่า rule ที่ผิดพลาดแต่อย่างใด

คำแนะนำเบื้องต้นสำหรับเครื่องที่ทำหน้าที่เป็นไฟร์วอลล์

- ปิด TCP/UDP service ที่ไม่ได้ใช้งาน เช่น bootps, finger ยิ่งเปิด service น้อยก็ยิ่งลดโอกาสในการโจมตีของผู้บุกรุก และยังเป็นการลดการใช้งาน CPU และหน่วยความจำของระบบอีกด้วย
- ในกรณีที่จำเป็นต้องเปิด service บนเครื่องไฟร์วอลล์ จะต้องจำกัดการเข้าถึงให้ใช้งานได้เฉพาะผู้ดูแลระบบเท่านั้น
- ปิด service ที่ไม่จำเป็นอื่นๆ บนเครื่องไฟร์วอลล์ เช่น การทำ remote configuration
- ยกเลิก interface ที่ไม่ได้ใช้งานในเครื่องไฟร์วอลล์

- ในกรณีที่ใช้ฮาร์ดแวร์ไฟร์วอลล์ จะต้องป้องกันการเข้าถึง port ที่ใช้ในการควบคุม เช่น console port
- แก้ไขค่า default password โดยให้มีความยาวอย่างต่ำ 8 ตัวอักษร, ไม่เป็นคำที่อยู่ในดิกชันนารี, ต้องไม่ขึ้นต้นด้วยตัวเลข, และมีตัวเลขรวมทั้งตัวอักษรพิเศษรวมอยู่ด้วย (เช่น „/<>;':"[]{}\|~!@#$%^&*()_+-=) และควรใช้รหัสผ่านที่แตกต่างกันในแต่ละเครื่อง ทั้งนี้ควรเปลี่ยนรหัสผ่านทุกๆ 90 วัน

หลักการในการสร้างกฎสำหรับไฟร์วอลล์

หลักการง่ายๆ ในการสร้างกฎของไฟร์วอลล์ที่ดีคือ ความง่าย (Simplicity) ซึ่งความง่ายในที่นี้หมายถึงการสร้างกฎที่สั้นๆ อ่านง่ายได้ใจความ ไฟร์วอลล์ที่ดีไม่ควรมี กฎมากกว่า 30 กฎ เพราะถ้ามากกว่านี้จะทำให้เกิดความสับสนได้ง่าย และอาจจะทำให้เกิดความผิดพลาดโดยไม่รู้ตัวขึ้น นอกจากนี้ยังมีข้อดีในส่วนที่ทำให้เครื่องทำงานน้อยลงอีกด้วย

การสร้างกฎของไฟร์วอลล์ถือได้ว่าเป็นการนำนโยบายทางด้านความปลอดภัยขององค์กรมาบังคับใช้ในทางเทคนิค โดยใช้ไฟร์วอลล์เป็นเครื่องมือให้เกิดผลตามที่ต้องการ นอกจากนี้ยังมีกฎบางส่วนที่ถือได้ว่า ผู้ดูแลระบบควรเพิ่มเข้าไปในกฎของไฟร์วอลล์ เช่น การป้องกัน ip spoofing, ป้องกันการโจมตีแบบ land attack

ลำดับของกฎ

การเรียงลำดับของกฎก็มีความสำคัญเช่นเดียวกัน เพราะไฟร์วอลล์โดยส่วนใหญ่ทำงานแบบเรียงตามลำดับคือ ตรวจสอบ packet กับกฎตามลำดับของกฎที่สร้างไว้

คำแนะนำในการจัดลำดับของกฎ คือให้วางกฎที่เป็นกฎทั่วไปไว้ด้านล่าง และให้นำกฎที่มีความเฉพาะเจาะจงมาไว้ด้านบน เพื่อป้องกันไม่ให้ packet ถูกจับคู่กับกฎทั่วไปก่อน ยกตัวอย่างเช่น ให้นำกฎที่ทำหน้าที่กันไอพีแอดเดรสไปไว้ด้านบนเพื่อให้มั่นใจว่า ถ้ามี packet ที่มี ไอพีแอดเดรสตรงตามที่ระบุไว้ packet นั้นจะถูกละทิ้งไปก่อนที่จะถูกจัดคู่กับกฎอื่น

การกรอง TCP/IP

ผู้ดูแลไฟร์วอลล์สามารถกำหนดนโยบายเริ่มต้นได้ 2 รูปแบบคือ

- Default ACCEPT : ผู้ดูแลไฟร์วอลล์จะต้องสร้างกฎ เพื่อกำหนดว่าจะปิดบริการและเครื่องคอมพิวเตอร์ใดบ้าง โดยบริการและเครื่องคอมพิวเตอร์อื่นๆ ที่ไม่ถูกกำหนดไว้จะมีค่าให้ผ่านได้

- Default DROP : ผู้ดูแลไฟร์วอลล์จะต้องสร้างกฎ เพื่อกำหนดว่าจะเปิดบริการและเครื่องคอมพิวเตอร์ใดบ้าง โดยบริการและเครื่องคอมพิวเตอร์อื่นๆที่ไม่ถูกำหนดไว้จะถูกปิดไม่ให้ผ่าน

อย่างไรก็ตาม ไม่ว่าจะกำหนดนโยบายเริ่มต้นในรูปแบบใด ผู้ดูแลไฟร์วอลล์ก็ควรทราบบริการ TCP/IP ที่เป็นจุดอ่อนต่างๆ ในระบบดังนี้

ตารางที่ 2.1 แสดงบริการ TCP/UDP ที่ควรปิดกั้นที่ไฟร์วอลล์ โดยไม่ให้ใช้ทั้งจากภายในและภายนอกเครือข่าย

Port(s) (Transport)	Server	Port(s) (Transport)	Server
1 (TCP & UDP)	tcpmux	1981 (TCP)	Shockwave
7 (TCP & UDP)	echo	1999 (TCP)	BackDoor
9 (TCP & UDP)	discard	2001 (TCP)	Trojan Cow
11 (TCP & UDP)	systat	2023 (TCP)	Ripper
13 (TCP & UDP)	daytime	2049 (TCP & UDP)	nfs
15 (TCP & UDP)	netstat	2115 (TCP)	Bugs
17 (TCP & UDP)	qotd	2140 (TCP)	Deep Throat
19 (TCP & UDP)	chargen	2222 (TCP)	Subseven21
37 (TCP & UDP)	time	2301 (TCP & UDP)	compaqdiag
43 (TCP & UDP)	whose	2565 (TCP)	Striker
67 (TCP & UDP)	bootps	2583 (TCP)	Win Crash
68 (TCP & UDP)	bootpc	2701 (TCP & UDP)	sms-rcinfo
69 (UDP)	tftp	2702 (TCP & UDP)	sms-remctrl
93 (TCP)	supdup	2703 (TCP & UDP)	sms-chat
111 (TCP & UDP)	sunrpc	2704 (TCP & UDP)	sms-xfer
135 (TCP & UDP)	loc-srv	2801 (TCP)	Phineas P.

ตารางที่ 2.1 แสดงบริการ TCP/UDP ที่ควรปิดกั้นที่ไฟร์วอลล์ โดยไม่ให้ใช้ทั้งจากภายในและภายนอกเครือข่าย (ต่อ)

Port(s) (Transport)	Server	Port(s) (Transport)	Server
137 (TCP & UDP)	netbios-ns	4045 (TCP)	lockd
138 (TCP & UDP)	netbios-dgm	5800 - 5899 (TCP)	winvnc web server
139 (TCP & UDP)	netbios-ssn	5900 - 5999 (TCP)	winvnc
177 (TCP & UDP)	xdmcp	6000 - 6063 (TCP)	X11 Window System
445 (TCP & UDP)	microsoft-ds	6665 - 6669 (TCP)	irc
512 (TCP)	rexec	6711 - 6712 (TCP)	Subseven
513 (TCP)	rlogin	6776 (TCP)	Subseven
513 (UDP)	who	7000 (TCP)	Subseven21
514 (TCP)	rsh, rcp, rdist, rdump, rrestore	12345 - 12346 (TCP)	NetBus
515 (TCP)	lpr	16660 (TCP)	Stacheldraht
517 (UCP)	talk	27444 (UCP)	Trinoo
518 (UCP)	ntalk	27666 (TCP)	Trinoo
540 (TCP)	uucp	31335 (UCP)	Trinoo
1024 (TCP)	NetSpy	31337 -31338 (TCP & UDP)	Back Orifice
1045 (TCP)	Rasmin	32700 - 32900 (TCP & UDP)	RPC services
1090 (TCP)	Xtreme	32720 (TCP)	Trinity V3
1170 (TCP)	Psyber S.S	39168 (TCP)	Trinity V3
1234 (TCP)	Ultors Trojan	65000 (TCP)	Stacheldraht

ตารางที่ 2.1 แสดงบริการ TCP/UDP ที่ควรปิดกั้นที่ไฟร์วอลล์ โดยไม่ให้ใช้ทั้งจากภายในและภายนอกเครือข่าย (ต่อ)

Port(s) (Transport)	Server	Port(s) (Transport)	Server
1243 (TCP)	Backdoor-G		
1245 (TCP)	VooDoo Doll		
1349 (UCP)	Back Orifice DLL		
1492 (TCP)	FTP99CMP		
1600 (TCP)	Shivka-Burka		
1761 - 1764 (TCP & UDP)	sms-helpdesk		
1807 (TCP)	SpySender		

ตารางที่ 2.2 แสดงบริการ TCP/UDP ที่ควรปิดกั้นไม่ให้เข้ามาจากภายนอก

Port(s) (Transport)	Server
79 (TCP)	finger
161 (TCP & UDP)	snmp
162 (TCP & UDP)	snmp trap
514 (UDP)	syslog
550 (TCP & UDP)	new who

ตารางที่ 2.3 แสดงบริการ TCP/UDP ที่อาจจะปิดให้บริการใน DMZ โดยเปิดเฉพาะบริการที่ใช้จริง

Port(s) (Transport)	Server
20 (TCP)	ftpdata
21 (TCP)	ftp
22 (TCP)	ssh
23 (TCP)	telnet
25 (TCP)	smtp
Port(s) (Transport)	**Server**
53 (TCP & UDP)	domain
80 (TCP)	http
110 (TCP)	pop3
119 (TCP)	nntp
123 (TCP)	ntp
143 (TCP)	imap
179 (TCP)	bgp
389 (TCP & UDP)	ldap
443 (TCP)	ssl
1080 (TCP)	socks
3128 (TCP)	squid
8000 (TCP)	http (alternate)
8080 (TCP)	http-alt
8888 (TCP)	http (alternate)

ตารางที่ 2.4 แสดงข้อความ ICMP ที่ควรอนุญาตให้ออกไปจากเครือข่ายภายในได้

Message Type	
Number	**Name**
4	source quench
8	echo request (ping)
12	parameter problem

ตารางที่ 2.5 แสดงข้อความ ICMP ที่ควรอนุญาตให้เข้ามายังเครือข่ายภายในได้

Message Type	
Number	**Name**
0	echo reply
3	destination unreachable
4	source quench
11	time exceeded
12	parameter problem

คำแนะนำอื่นๆ สำหรับการสร้างกฎของไฟร์วอลล์

- ควรมีการบันทึกข้อมูลลงล็อกสำหรับกฎที่ใช้ ป้องกัน การเข้าถึงซึ่งข้อมูลนี้จะเป็นประโยชน์ในการตรวจสอบการบุกรุก
- ป้องกันการปลอมไอพี (IP spoof) สำหรับข้อมูลขาเข้ามาจากอินเทอร์เน็ต โดยป้องกันไม่ให้ packet ที่มีไอพีดังต่อไปนี้เข้ามายังเครือข่ายภายใน
- 127.0.0.0 – 127.255.255.255 : local host address
- 10.0.0.0 – 10.255.255.255 : reserved address

- 172.16.0.0 – 172.16.255.255 : reserved address
- 192.168.0.0 – 192.168.255.255 : reserved address
- 224.0.0.0 – 239.255.255.255 : multicast address
- ป้องกันเครื่องไฟร์วอลล์จากการโจมตีแบบ land attack ซึ่งการโจมตีแบบนี้จะใช้วิธีส่ง packet ที่มีไอพีแอดเดรสต้นทางตรงกันกับ ไอพีแอดเดรสปลายทาง รวมทั้งค่า พอร์ตต้นทางและพอร์ตปลายทางที่ตรงกัน ซึ่งก่อให้เกิดการโจมตีแบบ Denial of Service ได้ ซึ่งป้องกันได้โดย block ไม่ให้ข้อมูลขาเข้าที่มีไอพีแอดเดรสต้นทางตรงกันกับไอพีแอดเดรสของเครือข่ายภายในเข้ามาในระบบ
- ป้องกันการโจมตีแบบ Syn flood ที่เครื่องไฟร์วอลล์ ซึ่งผู้บุกรุกจะส่ง SYN packet จำนวนมากมายังเครื่องปลายทาง ทำให้คิวของการรับ connection ในบริการดังกล่าวเต็ม ทำให้ไม่สามารถให้บริการแก่เครื่องอื่นๆได้
- เครื่องไฟร์วอลล์และเครื่องอื่นๆ ภายในเครือข่ายควรได้รับป้องกันจากข้อความ ICMP บางชนิดเช่น ป้องกันการรับ ICMP Echo request ซึ่งสามารถส่งมาเพื่อรวบรวมข้อมูลสำหรับการโจมตีต่อไป หรือการส่ง ICMP Echo request packet ที่ส่งมาจากภายนอกยังสามารถเปลี่ยน routing table ในเครื่องคอมพิวเตอร์ได้อีกด้วย ซึ่งเป็นเรื่องที่อันตรายอย่างยิ่ง

สำหรับข้อมูลขาออกนั้นควรอนุญาตให้ข้อมูล ICMP ดังต่อไปนี้เท่านั้นที่สามารถออกไปได้

- Echo request
- Parameter Problem
- Source Quench

สำหรับข้อมูลขาเข้านั้น ควรอนุญาตให้ข้อมูล ICMP ดังต่อไปนี้เท่านั้นที่สามารถเข้ามาภายในได้

- Echo Reply
- Destination Unreachable
- Source Quench
- Time Exceeded
- Parameter Problem

- ป้องกันไฟร์วอลล์และเครื่องอื่นๆ ภายในเครือข่ายจาก traceroute เพราะ traceroute เป็นโปรแกรมที่ช่วยให้ทราบถึงไอพีแอดเดรสของ router ที่รับส่งต่อ packet ไปทีละ hop จนกระทั่งถึงปลายทางที่ต้องการ โดยใช้คุณสมบัติของ IP Time to Live (TTL) ในการทำงานโดยมันจะกำหนดค่า TTL counter ที่ทำให้ router ที่ packet ผ่านไปนั้นต้องสร้าง ICMP message กลับมาเสมอ สำหรับคำสั่ง tracert ใน windows นั้นจะใช้ ping (ICMP Echo) เป็นตัวส่ง packet ออกไปในขณะที่ traceroute ใน unix นั้นจะใช้ UDP datagram เป็นตัวส่งข้อมูลออกไป datagram ที่ถูกส่งออกไปนั้นจะถูกส่งไปยัง port 33434 โดยดีฟอลต์ และ ค่าหมายเลข port นี้จะถูกเพิ่มขึ้นเมื่อได้รับ packet ที่ตอบกลับมาอย่างถูกต้อง โดยปกติแล้ว traceroute มักจะส่ง datagram ออกไปจำนวน 3 datagram เพื่อป้องกันการสูญหายระหว่างทาง

  ถึงแม้ว่าจะมีการป้องกันการใช้งาน traceroute จากทั้ง Unix และ Windows แล้วก็ตาม ผู้บุกรุกก็ยังสามารถใช้วิธีอื่นในการ trace เข้ามายังเครือข่ายภายใน เช่น การใช้โปรแกรม Firewalk ดังนั้นหากต้องการหยุดการใช้ traceroute รวมทั้ง Firewalk แล้ว จะต้องใช้วิธี drop TTL Exceeded in Transit packet ที่ขาออกไปสู่อินเทอร์เน็ต
- จำกัดการเข้าถึงเครื่องไฟร์วอลล์ โดยให้ใช้งานในบริการที่จำเป็นเท่านั้น (สำหรับผู้ดูแลระบบเท่านั้น) และให้บันทึกข้อมูลล็อกสำหรับทั้ง connection ที่สำเร็จและไม่สำเร็จ
- ถ้าหากมี SNMP server ทำงานอยู่บนเครื่องไฟร์วอลล์จะต้องจำกัดการใช้งานให้ใช้เฉพาะผู้ดูแลระบบเท่านั้น และให้บันทึกข้อมูลล็อกสำหรับทั้ง connection ที่สำเร็จและไม่สำเร็จ

การเก็บข้อมูลล็อก

การเก็บข้อมูลล็อกของเครื่องไฟร์วอลล์เป็นเรื่องที่จำเป็นอย่างยิ่ง โดยเฉพาะในกรณีที่เครื่องโดน compromise ไปแล้วจะถือว่าเป็นหลักฐานที่แสดงให้เห็นถึงรูปแบบการโจมตีได้ มีคำแนะนำสำหรับการบันทึกข้อมูลล็อกดังนี้

- ให้ส่งข้อมูลล็อกที่มีความสำคัญไปยัง console ของเครื่องไฟร์วอลล์
- ส่งข้อมูลล็อกไปยังเครื่องที่ทำหน้าที่เก็บล็อกโดยเฉพาะ ซึ่งเครื่องนี้ได้รับการควบคุมการเข้าถึงอย่างเคร่งครัด และไม่ได้เปิดให้บริการอื่นใดยกเว้น syslog
- ตั้งเวลาเครื่องไฟร์วอลล์และเครื่องอื่นๆ ในเครือข่ายให้ใช้เวลาที่ตรงกันทั้งหมด โดยใช้ NTP
- ป้องกันการโจมตีแบบ log flooding ซึ่งจะทำให้ฮาร์ดดิสก์เต็มอย่างรวดเร็ว

- ไม่ควรส่งข้อมูลล็อกออกไปยังเครื่องพิมพ์โดยตรง เพราะอาจจะเสี่ยงต่อการสูญเสียข้อมูลในกรณีที่เครื่องพิมพ์มีปัญหา

2.5 การใช้งาน IPFW

รูปแบบของไฟร์วอลล์นั้นจะถูกสร้างขึ้นมาจากบัญชีรายการของกฎ ซึ่งระบบจะทำการอ่านข้อมูลทั้งขาเข้าและขาออกทีละ packet จนกระทั่งเจอกฎที่เหมาะสมกับ packet นั้นๆ เมื่อพบกฎที่เหมาะสมกับ packet นั้นแล้วการกระทำต่อมาที่จะกระทำกับ packet ก็จะขึ้นกับกฎของระบบที่ได้ติดตั้งไว้ กฎทั้งหมดจะถูกใช้กับทุก interface ของระบบและแน่นอนเป็นความรับผิดชอบของผู้ดูแลระบบที่จะดูแลและเขียนกฎเพื่อให้ง่ายต่อการควบคุมและตรวจสอบ

การปรับแต่งระบบจะประกอบไปด้วยกฎพื้นฐาน (หมายเลข 65535) ซึ่งจะไม่สามารถเปลี่ยนแปลงได้ และทุก packet จะต้องผ่านการตรวจสอบจากกฎข้อนี้ ส่วนการกระทำที่เกิดจากกฎข้อนี้จะสามารถเป็นได้คือ DENY หรือ ALLOW ซึ่งจะขึ้นอยู่การปรับแต่งใน เคอร์เนล

ถ้าในรายการของกฎนั้นมีกฎใดกฎหนึ่งหรือหลายกฎที่ใช้ตัวเลือก keep-state หรือ limit แล้วล่ะก็ ipfw ก็จะทำหน้าที่ตัวเองให้เป็นแบบ stateful ตัวอย่างเช่น เมื่อมีการจับคู่กันระหว่าง packet และกฎ แล้วก็จะมีการสร้างกฎแบบไม่ตายตัวด้วยค่าต่างๆ (address และ port) ของ packet ที่ถูกจับคู่

ด้วยกฎที่ไม่ตายตัวนี้ซึ่งจะมีการกำหนดระยะเวลา และทำการตรวจสอบที่จุดเริ่มต้นของเหตุการณ์ของกฎที่ระบุถึงตัวเลือกแบบ check-state หรือ keep-state และโดยทั่วๆไปจะใช้เพื่อให้ทำการเปิดไฟร์วอลล์ตามความต้องการ โดยจะเป็นไปตามเฉพาะข้อมูลที่ต้องการ

กฎทุกกฎ (รวมไปถึงกฎที่ไม่ตายตัว) จะมีตัวนับเข้ามาเกี่ยวข้องด้วยซึ่งแต่ละกฎจะมีตัวนับคือ ตัวนับ packet ตัวนับ byte ตัวนับ log และเวลาล่าสุดที่กฎนี้ถูกจับคู่กับ packet ตัวนับจะถูกแสดงและถูกตั้งใหม่ได้ด้วยคำสั่ง ipfw

จะสามารถเพิ่มกฎได้ด้วยคำสั่ง add และจะสามารถลบกฎแต่ละกฎได้ด้วยคำสั่ง delete และการที่จะลบกฎทุกกฎก็จะสามารถทำได้ด้วยคำสั่ง flush การแสดงผลพร้อมกับตัวนับและกฎต่างๆนั้นจะสามารถทำได้ด้วยคำสั่ง show และ list และสุดท้ายการจะตั้งค่าของตัวนับใหม่ด้วยคำสั่ง zero และ resetlog โดยในแต่ละคำสั่งก็จะมีตัวเลือกต่างๆมากมายดังนี้

-a เมื่อใช้ตัวเลือกนี้จะแสดงค่าของตัวนับ ตัวเลือกนี้จะใช้สำหรับคำสั่ง show เท่านั้น

-d เมื่อใช้ตัวเลือกนี้จะเป็นการแสดงกฎแบบไม่ตายตัวขึ้นมาด้วย

-e เมื่อใช้ตัวเลือกนี้พร้อมกับตัวเลือก –d จะเป็นการแสดงว่ากฎแบบไม่ตายตัวนั้นจะหมดอายุเมื่อใด

-f เมื่อใช้ตัวเลือกนี้ระบบจะไม่มีการถามยืนยันเมื่อมีการถาม ตัวอย่างเช่นในการใช้คำสั่ง flush จะมีการถามยืนยันเพื่อป้องกันความผิดพลาด

-q เมื่อมีการใช้ตัวเลือกนี้ระบบจะไม่มีการแสดงผลออกมา เมื่อมีการ เพิ่ม เปลี่ยนค่าตัวนับ เปลี่ยนค่า log หรือทำการลบกฎทั้งหมด ตัวเลือกนี้จะใช้มากในการปรับแต่กฎด้วยการใช้ชุดของคำสั่งของ ipfw

-t เมื่อใช้กับตัวเลือกนี้ จะแสดงถึงเวลาล่าสุดที่กฎนั้นๆได้ถูกจับคู่กับ packet

-N ตัวเลือกนี้จะเป็นการพยายามที่จะแสดง address และ port เป็นชื่อ

-s [field] ตัวเลือกนี้จะใช้สำหรับในการเรียงลำดับตัวนับทั้ง 4

รูปแบบของกฎใน ipfw

ipfw มีรูปแบบในการตั้งกฎดังนี้

ipfw [**prob** ความน่าจะเป็นในการเข้าคู่] การกระทำ [**log** [**logamount** จำนวน]] โปรโตคอล **from** ต้นทาง **to** ปลายทาง [ชื่อเน็ตเวิร์คการ์ด] [ตัวเลือก]

โดยแต่ละ packet จะถูกกรองโดยเกี่ยวข้องกับข้อมูลต่างๆของ packet ดังนี้

- เน็ตเวิร์คการที่ใช้รับและส่ง
- ทิศทางของข้อมูล
- หมายเลข IP ต้นทางและปลายทาง
- โปรโตคอล
- พอร์ตต้นทางและปลายทาง
- TCP flags
- IP fragment flag
- ตัวเลือกของ IP
- รูปแบบของ ICMP
- หมายเลขของผู้ใช้และหมายเลขของกลุ่มของ socket ที่เกี่ยวข้อกับ packet
- prob ความน่าจะเป็นในการเข้าคู่

การจับคู่จะอธิบายเฉพาะความน่าจะเป็นที่ระบุได้ นั้นจะสามารถใช้การสุ่มที่จะเลือกทิ้ง packet นั้นทิ้งไป หรือจะเป็นการจำลองผลจากการที่ packet นั้นหมดอายุในการส่ง

การกระทำ

allow จะอนุญาตให้ packet ผ่านเมื่อ packet นั้นจับคู่ได้กับกฎๆ นี้ ซึ่งจะมีค่าเทียบเท่าได้กับ pass, permit และ accept เมื่อ packet ใดถูกจับคู่กับกฎนี้แล้วก็จะจบการค้นหา

deny จะทำการทิ้ง packet นั้นไปเมื่อ packet นั้นถูกจับคู่ได้กับกฎๆนี้ ซึ่งจะมีค่าเทียบเท่าได้กับ deny เมื่อ packet ใดถูกจับคู่กับกฎนี้แล้วก็จะจบการค้นหา

reject จะทำการทิ้ง packet นั้นไปเมื่อ packet นั้นถูกจับคู่ได้กับกฎๆนี้และจะทำการส่ง ICMP host unreachable กลับไป เมื่อ packet ใดถูกจับคู่กับกฎนี้แล้วก็จะจบการค้นหา

unreach รหัส

จะทำการทิ้ง packet นั้นไปเมื่อ packet นั้นถูกจับคู่ได้กับกฎๆนี้และจะทำการส่ง ICMP unreachable ตามรหัสที่ใส่ไว้กลับไป เมื่อ packet ใดถูกจับคู่กับกฎนี้แล้วก็จะจบการค้นหา

reset ใช้สำหรับ TCP packet เท่านั้น จะทำการทิ้ง packet นั้นไปเมื่อ packet นั้นถูกจับคู่ได้กับกฎๆนี้และจะทำการส่ง TCP reset (RST) กลับไป เมื่อ packet ใดถูกจับคู่กับกฎนี้แล้วก็จะจบการค้นหา

count ปรับปรุงตัวนับสำหรับทุก packet ที่ถูกจับคู่กับกฎ การค้นหาจะดำเนินต่อไปแม้ว่าจะถูกจับคู่กับ กฎนี้

check-state

ตรวจสอบ packet ที่ตรงข้ามกับกฎแบบไม่ตายตัว ถ้าจับคู่กันได้ก็จะหยุดการค้นหา ถ้าไม่ได้ก็หาต่อไป ถ้าไม่พบกฎที่ระบุ check-state การตรวจสอบกฎแบบไม่ตายตัวก็จะเริ่มตรวจสอบกฎที่จุดเริ่มต้น keep-state

divert พอร์ต

เปลี่ยนเส้นทางของ Packet ที่ถูกจับคู่กับกฎนี้ไปยังพอร์ตที่กำหนด เมื่อ packet ใดถูกจับคู่กับกฎนี้แล้วก็จะจบการค้นหา

tee พอร์ต

เมื่อ packet ใดที่ถูกจับคู่กับกฎนี้ packet นั้นจะถูกคัดลอกและส่งไปยังพอร์ตที่กำหนด เมื่อ packet ใดถูกจับคู่กับกฎนี้แล้วก็จะจบการค้นหา

บทที่ 3
การติดตั้ง ปรับแต่ง และทดสอบการใช้งานระบบไฟร์วอลล์

เทคโนโลยีไฟร์วอลล์ เป็นเทคโนโลยีที่ได้รับการยอมรับในวงกว้าง แต่ปัญหาก็คือ ระบบไฟร์วอลล์ที่มีการจัดจำหน่ายในเชิงพาณิชย์ อยู่ในท้องตลาดนั้นจะมีราคาสูง จึงมีบุคคลบางกลุ่มที่ได้พยายามหาทางเลือกอื่นโดยการสร้างไฟร์วอลล์ที่เป็น Freeware ที่แจกจ่ายโดยไม่คิดค่าใช้จ่าย อย่างเช่น IPFW ,IPTables เพื่อให้สามารถใช้ไฟร์วอลล์ได้โดยไม่จำเป็นจะต้องเสียค่าใช้จ่ายมาก ทั้งนี้ แนวโน้มในปัจจุบันหน่วยงานทางด้านเทคโนโลยีสารสนเทศต่างๆ ก็เริ่มหันมาสนับสนุนและให้ความสนใจการใช้งานซอฟต์แวร์ที่เป็น freeware เพื่อเป็นการลดค่าใช้จ่ายในองค์กร

3.1 การวางแผนปฏิบัติงาน

ได้เลือกใช้ระบบปฏิบัติการและซอฟต์แวร์ดังต่อไปนี้ (สามารถอ่านวิธีการติดตั้งได้จากภาคผนวก ก ขั้นตอนการติดตั้งระบบ

1. **ระบบปฏิบัติการ เลือกใช้ FreeBSD 5.0**

   เป็นระบบปฏิบัติการแบบยูนิกซ์ที่ได้รับความนิยมอย่างแพร่หลาย ออกแบบไว้สำหรับการใช้งานบนเครื่องคอมพิวเตอร์ และเครื่องคอมพิวเตอร์ที่ให้บริการ โดยที่ไม่ต้องเสียค่าใช้จ่าย โดยระบบปฏิบัติการ FreeBSD นั้นเป็นระบบปฏิบัติการที่มีความปลอดภัยสูงกว่าระบบยูนิกซ์แบบ Open Source ทั่วๆไป

2. **ซอฟต์แวร์ไฟร์วอลล์เลือกใช้ IPFW**

   เป็นซอฟต์แวร์ไฟร์วอลล์ที่ทำงานกับระบบปฏิบัติการ FreeBSD โดยที่ IPFW จะทำงานอยู่ใน เคอร์เนล ของระบบปฏิบัติการ เพื่อความปลอดภัยของระบบไฟร์วอลล์ ซึ่งมีส่วนที่ติดต่อกับผู้ใช้ที่เป็นแบบ command line คือผู้ใช้งานจะต้องทำการเพิ่มและลบกฎต่างๆ ของไฟร์วอลล์ด้วยคำสั่งซึ่งเป็นการยุ่งยากในการจดจำ

3. **โปรแกรม Script Extension เลือกใช้ PHP Extension เวอร์ชัน 4.3.2**

   PHP Extension เป็น scripting language ที่ทำงานร่วมกับ HTML โดย PHP จะถูกเขียนแทรกเข้าไปใน HTML โดยที่ PHP script จะเพิ่มความสามารถในการทำงานผ่าน

เวปที่ HTML ธรรมดาไม่สามารถทำได้ โดยที่ PHP นั้นเป็น Open Source สามารถนำมาใช้งานได้โดยไม่มีค่าใช้จ่ายซึ่ง PHP ได้พัฒนาโดยกลุ่มบุคคล ซึ่งในปัจจุบันได้พัฒนามาถึงเวอร์ชัน 4.3.3

4. **ซอฟต์แวร์ SSL/TLS เลือกใช้ OpenSSL เวอร์ชัน 0.9.6g**

โครงการ OpenSSL เป็นความพยายามในการพัฒนาทูลคิดแบบ Open Source ที่มีประสิทธิภาพเทียบเท่าผลิตภัณฑ์เชิงพาณิชย์ โดยมีการอิมพลีเมนต์โปรโตคอล Secure Socket Layer (SSL v2/v3) และ Transport Layer Security (TLS v1) รวมไปถึงไลบรารี่ระบบบรหัสต่าง (Cryptography) ซึ่งโครงการนี้ได้รับความร่วมมือจากอาสาสมัครทั่วโลกที่มาจากเครือข่ายอินเทอร์เน็ท ในการสื่อสาร วางแผนและพัฒนาทูลคิด OpenSSL และเอกสารอื่นๆที่เกี่ยวข้อง OpesnSSL มีพื้นฐานมาจากไลบรารี่ SSLeay ที่พัฒนาโดย Eric A. Young และ Tim J. Hudson มีการจดลิขสิทธ์ภายใต้รูปแบบเดียวกับ Apache คือสามารถนำไปใช้ได้ทั้งในเชิงพาณิชย์และไม่ใช่เชิงพาณิชย์

5. **โปรแกรม Script Extension เลือกใช้ PHP Extension เวอร์ชัน 4.3.2**

PHP Extension เป็น scripting language ที่ทำงานร่วมกับ HTML โดย PHP จะถูกเขียนแทรกเข้าไปใน HTML โดยที่ PHP script จะเพิ่มความสามารถในการทำงานผ่านเวปที่ HTML ธรรมดาไม่สามารถทำได้

6. **เว็บเซิร์ฟเวอร์ที่รองรับการทำงาน PHP Extension และ รองรับ SSL/TLS เลือกใช้ Apache เวอร์ชัน 2.0.46**

Apache HTTP Server เป็นซอฟต์แวร์สำหรับให้บริการเว็ปเซิร์ฟเวอร์ (HTTP/Web Server) ผ่านทางโปรโตคอล HTTP โดยเป็นซอฟต์แวร์แบบ Open Source สามารถนำไปใช้งานได้โดยไม่มีค่าใช้จ่าย เนื่องจากกลุ่มอาสาสมัครจากทั่วโลกได้ร่วมกันพัฒนามาจนกระทั่งในปัจจุบันเป็นเวอร์ชันที่ 2.0.47 แล้ว

7. **โปรแกรมภาษาเลือกใช้ Perl เวอร์ชัน 5.8.0**

Perl เป็นโปรแกรมภาษาที่เป็น Open Source ที่สามารถนำไปใช้งานได้โดยไม่มีค่าใช้จ่าย โดยโปรแกรมภาษา Perl นั้นมีความสามารถที่จะเขียนแล้วทำงานบนระบบปฏิบัติการต่างๆได้เป็นอย่างดี

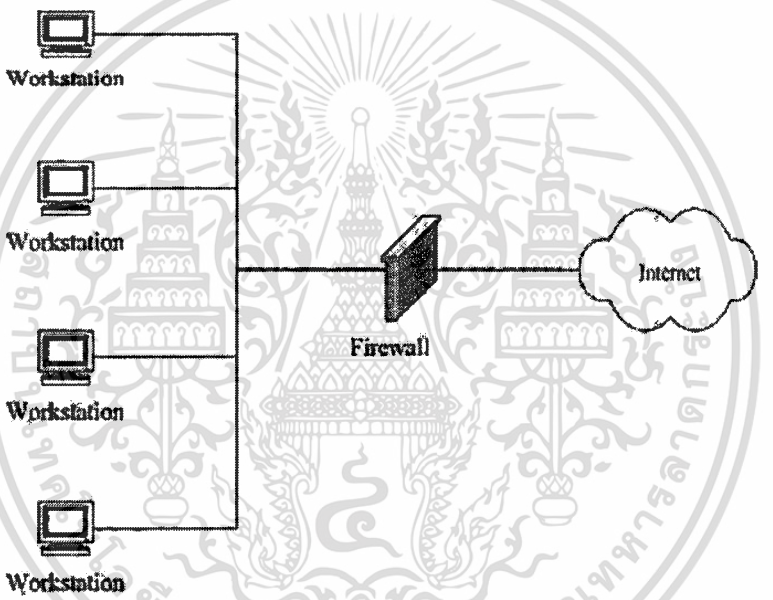
8. **โปรแกรมฐานข้อมูล เลือกใช้ MySQL เวอร์ชัน 3.23.54**

MySQL เป็นโปรแกรมฐานข้อมูลที่เป็น Open Source ที่นิยมใช้กันอย่างแพร่หลาย เนื่องด้วยที่ตัวโปรแกรมเองเป็นโปรแกรมที่มีประสิทธิภาพดี สามารถนำมาใช้ทดแทนกับโปรแกรมฐานข้อมูลที่จำหน่ายในเชิงพานิชย์ได้

3.2 การวางแผนโครงสร้างของระบบ

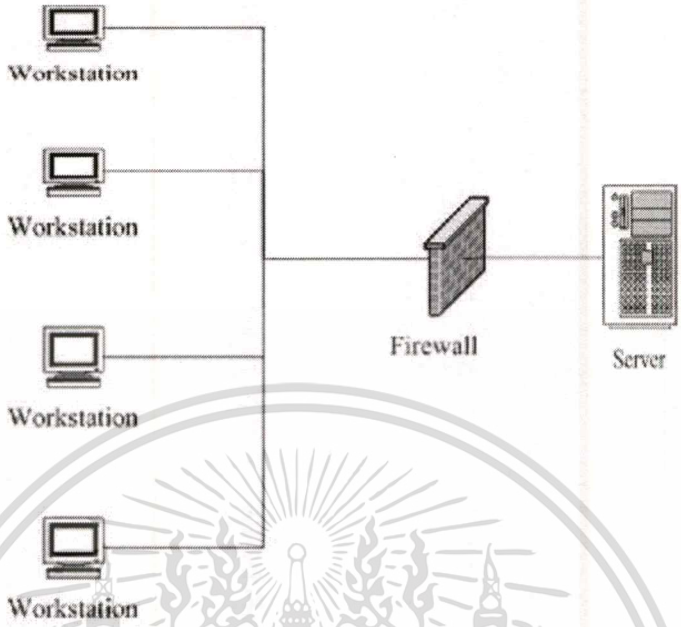
โครงสร้างของระบบที่จะนำเอาไฟร์วอลล์มาใช้งานนั้น ในที่นี้จะแบ่งออกใน 2 ลักษณะคือ

- จากรูปที่ 3.1 จะทำการติดตั้งไฟร์วอลล์บนเครื่องที่ทำหน้าที่เป็น gateway โดยที่ไฟร์วอลล์นี้จะทำหน้าที่เป็น proxy ด้วย คือเครื่องที่ทำหน้าที่เป็นไฟร์วอลล์จะเป็นตัวแทนที่จะนำข้อมูลจากภายนอกที่เครื่องลูกข่ายต้องการมาให้ โดยที่ไฟร์วอลล์จะทำหน้าที่ กรอง packet ที่จะผ่านเข้าออก ดูว่า packet ใดผ่านได้หรือผ่านไม่ได้ โดยจากรูปที่ 3.1 จะเห็นได้ว่าการที่เครื่องลูกข่ายจะออกไปยังอินเทอร์เน็ทได้นั้นจะต้องผ่านออกไปทางไฟร์วอลล์เท่านั้น



รูปที่ 3.1 แสดงโครงสร้างการใช้งานไฟร์วอลล์ในลักษณะที่ 1

- จากรูปที่ 3.2 จะทำการติดตั้งไฟร์วอลล์บนเครื่องคอมพิวเตอร์ที่ให้บริการที่ต้องการความปลอดภัยสูง โดยเครื่องลูกข่ายที่จะเข้าถึงบริการของเครื่องให้บริการนั้นๆ จะต้องใช้บริการผ่านไฟร์วอลล์ ซึ่งไฟร์วอลล์จะทำหน้าที่กรอง packet ที่ผ่านเข้าออกว่าจะอนุญาตให้ packet นั้นผ่านไปได้หรือไม่



รูปที่ 3.2 แสดงโครงสร้างการใช้งานไฟร์วอลล์ในลักษณะที่ 2

3.3 การเริ่มต้นใช้งานซอฟต์แวร์ IPFW

ก่อนที่จะเริ่มใช้งานซอฟต์แวร์ IPFW นั้นเราจะต้องทำการเปิดการใช้งาน IPFW ซึ่งอยู่ในเคอร์เนล ของระบบปฏิบัติการ FreeBSD ขึ้นมาเสียก่อนโดยขั้นตอนในการเปิดการใช้งาน IPFW มีขั้นตอนดังต่อไปนี้

- เข้าไปยังไดเรคทอรี่ที่เก็บไฟล์ เคอร์เนล ของระบบไว้ ดังรูปที่ 3.3

```
test#
test#
test# cd /usr/src/sys/i386/conf/
```

รูปที่ 3.3 แสดงไดเรคทอรี่ที่เก็บไฟล์ เคอร์เนล

- ทำการสร้างไฟล์ค่าเริ่มต้นของ เคอร์เนล ด้วยคำสั่ง “make LINT” จะได้ผลดังรูปที่ 3.4

```
test#
test# make LINT
cat ../../conf/NOTES NOTES | sed -E -n -f ../../conf/makeLINT | sed > LINT
test#
```

รูปที่ 3.4 แสดงคำสั่งและผลของการสร้างไฟล์ เคอร์เนล เริ่มต้น

- นำตัวเลือกที่เกี่ยวข้องกับ IPFW ในไฟล์ LINT ที่ได้สร้างมา นำไปใส่ไว้ในไฟล์เคอร์เนล ที่ต้องการใช้งาน โดยอาจจะทำการ copy มาจากไฟล์ เคอร์เนล GENERIC ก็ได้ โดยในที่นี้คือไฟล์ เคอร์เนล FIREWALL โดยจะใช้ตัวเลือกในไฟล์ LINT ตารางที่ 3.1

ตารางที่ 3.1 แสดงถึงความหมายของตัวเลือกต่างๆในไฟล์ เคอร์เนล ที่เกี่ยวกับไฟร์วอลล์

ตัวเลือก	ความหมาย
options IPFIREWALL	ให้มีการใช้งาน IPFW
options IPFIREWALL_VERBOSE	เก็บล็อกของไฟร์วอลล์ลง syslog
options IPFIREWALL_FORWARD	ให้ไฟร์วอลล์ทำการส่งต่อ packet
options IPFIREWALL_VERBOSE_LIMIT=100	กำหนดจำนวนของล็อกที่จะกระทำการส่งล็อกไปยัง syslog ของระบบ
options IPDIVERT	ใช้ในกรณีต้องการให้เครื่องทำหน้าที่เป็น NAT

- นอกจากตัวเลือกที่เกี่ยวข้องกับไฟร์วอลล์แล้วยังมีตัวเลือกที่ช่วยเพิ่มความปลอดภัยให้กับระบบไฟร์วอลล์อีกด้วยดังตารางที่ 3.2

ตารางที่ 3.2 แสดงถึงความหมายของตัวเลือกต่างๆในไฟล์ เคอร์เนล ที่เกี่ยวกับความปลอดภัย

ตัวเลือก	ความหมาย
options TCP_DROP_SYNFIN	ให้ระบบละทิ้ง packet ที่มีสถานะเป็น SYN+FIN ทิ้งไป

- เมื่อทำการเพิ่มตัวเลือกตามที่ต้องการแล้ว ขั้นตอนต่อมาก็จะเป็นการปรับแต่งเคอร์เนล ใหม่ที่เราจะนำมาใช้งานโดยใช้คำสั่ง make ดังรูปที่ 3.5

```
test# config FIREWALL
Kernel build directory is ../compile/FIREWALL
Don't forget to do a ``make depend''
test#
```

รูปที่ 3.5 แสดงคำสั่งเริ่มต้นในการ คอมไพล์ เคอร์เนล

- เมื่อใช้คำสั่ง config เรียบร้อยแล้วระบบจะแจ้งบอกว่า ได้ทำการสร้าง เคอร์เนล ไว้ที่ไดเรกทอรี่ ../compile/FIREWALL แล้วให้ใช้คำสั่ง "make depend" ดังรูปที่ 3.6

```
test#
test# cd ../compile/FIREWALL/
test# make depend
```

รูปที่ 3.6 แสดงคำสั่ง make depend

- เมื่อใช้คำสั่ง config เรียบร้อยแล้วก็จะตามด้วยคำสั่ง make ทำให้ระบบทำการเริ่มคอมไพล์ เคอร์เนล ดังรูปที่ 3.7

```
test#
test# make
```

รูปที่ 3.7 แสดงคำสั่งที่ใช้ในการ คอมไพล์ เคอร์เนล

- เมื่อทำการ คอมไพล์ เคอร์เนล เสร็จแล้วก็จะต้องทำการติดตั้ง เคอร์เนล ใหม่ที่เราได้คอมไพล์ทำได้โดยใช้คำสั่งตามรูปที่ 3.8

```
test#
test# make install
```

รูปที่ 3.8 แสดงคำสั่งในการติดตั้งเคอร์เนล

- เมื่อทำการคอมไพล์เคอร์เนลและทำการติดตั้งเรียบร้อยแล้ว ก็ให้ทำการรีสตาร์ทเครื่อง 1 ครั้ง เมื่อเครื่องเปิดขึ้นมาทำงานตามปกติแล้ว ใช้คำสั่ง "uanme –v" ควรจะได้ค่าตามรูปที่ 3.9

```
root@test.gits.net.th:/usr/src/sys/i386/compile/FIREWALL
```

รูปที่ 3.9 แสดงถึงผลลัพธ์เมื่อใช้คำสั่งแสดงเคอร์เนลที่ใช้อยู่

- ก็เป็นอันเสร็จสิ้นการเปิดใช้งานซอฟต์แวร์ไฟร์วอลล์ IPFW เมื่อระบบพร้อมทำงาน IPFW แล้วระบบปฏิบัติการ FreeBSD มีไฟล์ที่ใช้สำหรับกำหนดค่าเริ่มต้นของระบบคือไฟล์ "rc.conf" ซึ่งอยู่ในไดเรคทอรี่ "/etc" โดยไฟล์นี้จะเป็นไฟล์ที่ใช้เก็บว่าระบบนี้จะเปิดการทำงานใดบ้าง ในที่นี้ก็จะทำการแก้ไขไฟล์นี้ตามรูปที่ 3.10

```
firewall_enable="YES"            # Set to YES to enable firewall functionality
firewall_quiet="YES"             # Set to YES to suppress rule display
firewall_logging="YES"           # Set to YES to enable events logging
```

รูปที่ 3.10 แสดงถึงส่วนที่ต้องเพิ่มเข้าไปในไฟล์ /etc/rc.conf

- สิ้นสุดขั้นตอนของการเปิดการใช้งานซอฟต์แวร์ไฟร์วอลล์ IPFW

3.4 การใช้งานไฟร์วอลล์ IPFW

การใช้งาน IPFW จำเป็นที่จะต้องจดจำคำสั่งและรูปแบบในการทำงานของ IPFW เนื่องจาก IPFW นั้นจะรับคำสั่งการทำงานจากคอมมานด์ไลน์ โดยจะมีรูปแบบคำสั่งดังนี้

- การเพิ่มกฎของ IFW ทำได้โดยใช้คำสั่ง add โดยมีรูปแบบการใช้งานดังนี้

Ipfw add หมายเลขกฎ [allow หรือ deny] โปรโตคอล from ไอพีต้นทาง(พอร์ตต้นทาง) to ไอพีปลายทาง(พอร์ตปลายทาง)

ยกตัวอย่างเช่นรูปที่ 3.11 เป็นการเพิ่มกฎของ IPFW กฎที่ 1 โดยเป็นการอนุญาตให้ IP packet จากหมายเลขไอพีที่ใดก็ได้วิ่งผ่านไปยังหมายเลขไอพีที่ใดก็ได้

```
test# ipfw add 1 allow ip from any to any
```

รูปที่ 3.11 แสดงตัวอย่างการเพิ่มกฎใน IPFW

- การลบกฎของ IPFW ทำได้โดยใช้คำสั่ง del โดยมีรูปแบบการใช้งานดังนี้

ipfw del หมายเลขกฎ

ยกตัวอย่างเช่นรูปที่ 3.12 เป็นการลบกฎของ IPFW กฎที่ 1 ทิ้ง

```
test# ipfw del 1
```

รูปที่ 3.12 แสดงตัวอย่างการลบกฎใน IPFW

- การลบกฎทั้งหมดที่มีอยู่ของ IPFW ทิ้งทำได้โดยใช้คำสั่ง flush โดยมีรูปแบบการใช้งานดังนี้

ipfw flush

- การแสดงกฎของ IPFW ที่มีการใช้งานอยู่ทำได้ 2 วิธีคือ ใช้คำสั่ง list และคำสั่ง show โดยมีรูปแบบการใช้งานดังนี้

ipfw list
ipfw show

โดยที่ทั้งสองคำสั่งนี้เมื่อสั่งแล้วจะทำการแสดงกฎของไฟร์วอลล์ที่มีอยู่ในขณะนั้นๆ ออกมาแต่จะแตกต่างกันที่คำสั่ง list จะแสดงเฉพาะกฎออกมาแต่ว่าคำสั่ง show จะแสดงตัวนับของข้อมูลที่วิ่งผ่านกฎนั้นๆ ออกมาด้วย ดังรูปที่ 3.13

```
test# ipfw list
00001 allow tcp from 172.17.1.19 to 172.17.1.23 dst-port 22 in via dc0
00002 allow tcp from 172.17.1.23 22 to 172.17.1.19 out via dc0
00005 allow udp from 172.17.1.19 to 172.17.7.255 dst-port 138 in via dc0
00006 allow udp from 172.17.7.255 138 to 172.17.1.19 out via dc0
00007 allow udp from 172.17.1.19 to 172.17.7.255 dst-port 137 in via dc0
00008 allow udp from 172.17.7.255 137 to 172.17.1.19 out via dc0
65534 deny log ip from any to any
65535 deny ip from any to any
test#
test# ipfw show
00001        232       20800 allow tcp from 172.17.1.19 to 172.17.1.23 dst-port 22 in via dc0
00002        155       15016 allow tcp from 172.17.1.23 22 to 172.17.1.19 out via dc0
00005          0           0 allow udp from 172.17.1.19 to 172.17.7.255 dst-port 138 in via dc0
00006          0           0 allow udp from 172.17.7.255 138 to 172.17.1.19 out via dc0
00007          0           0 allow udp from 172.17.1.19 to 172.17.7.255 dst-port 137 in via dc0
00008          0           0 allow udp from 172.17.7.255 137 to 172.17.1.19 out via dc0
65534      16699     1983764 deny log ip from any to any
65535          2         120 deny ip from any to any
test#
```

รูปที่ 3.13 แสดงถึงผลลัพธ์ของคำสั่งที่ใช้ในการแสดงกฎของไฟร์วอลล์

- การรีเซทตัวนับของกฎของ IPFW ทำได้โดยการใช้คำสั่ง zero จะเป็นการรีเซทค่าของตัวนับของกฎทุกกฎให้เป็น 0 โดยมีรูปแบบการใช้งานดังนี้

ipfw zero

โดยเมื่อใช้คำสั่งนี้แล้วก็จะได้ผลลัพธ์ดังรูปที่ 3.14

```
test# ipfw show
00001         24        2080 allow tcp from 172.17.1.19 to 172.17.1.23 dst-port 22 in via dc0
00002         18        1984 allow tcp from 172.17.1.23 22 to 172.17.1.19 out via dc0
00005          0           0 allow udp from 172.17.1.19 to 172.17.7.255 dst-port 138 in via dc0
65534         33        4673 deny log ip from any to any
65535          0           0 deny ip from any to any
test#
test# ipfw zero
Accounting cleared.
test#
test# ipfw show
00001          0           0 allow tcp from 172.17.1.19 to 172.17.1.23 dst-port 22 in via dc0
00002          0           0 allow tcp from 172.17.1.23 22 to 172.17.1.19 out via dc0
00005          0           0 allow udp from 172.17.1.19 to 172.17.7.255 dst-port 138 in via dc0
65534          0           0 deny log ip from any to any
65535          0           0 deny ip from any to any
test#
```

รูปที่ 3.14 จะเป็นการแสดงผลลัพธ์ของการใช้คำสั่ง zero ของ IPFW

3.5 ผลการใช้งานไฟร์วอลล์ IPFW

- **การทำงานของไฟร์วอลล์ IPFW** IPFW สามารถทำการเปิดและปิด อนุญาต และไม่อนุญาต packet ต่างๆได้ตามที่กฎได้สร้างขึ้นมารองรับไว้
- **การสร้างกฎของไฟร์วอลล์ IPFW** ผู้ใช้สามารถสร้างกฎของไฟร์วอลล์ ได้ตามความต้องการ

- **การลบกฎของไฟร์วอลล์ IPFW** ผู้ใช้สามารถทำการลบกฎของไฟร์วอลล์ที่ไม่ต้องการออกไปได้
- **การใช้งานไฟร์วอลล์** ผู้ใช้งานสามารถควบคุมและใช้งานไฟร์วอลล์ได้ตามที่ต้องการ

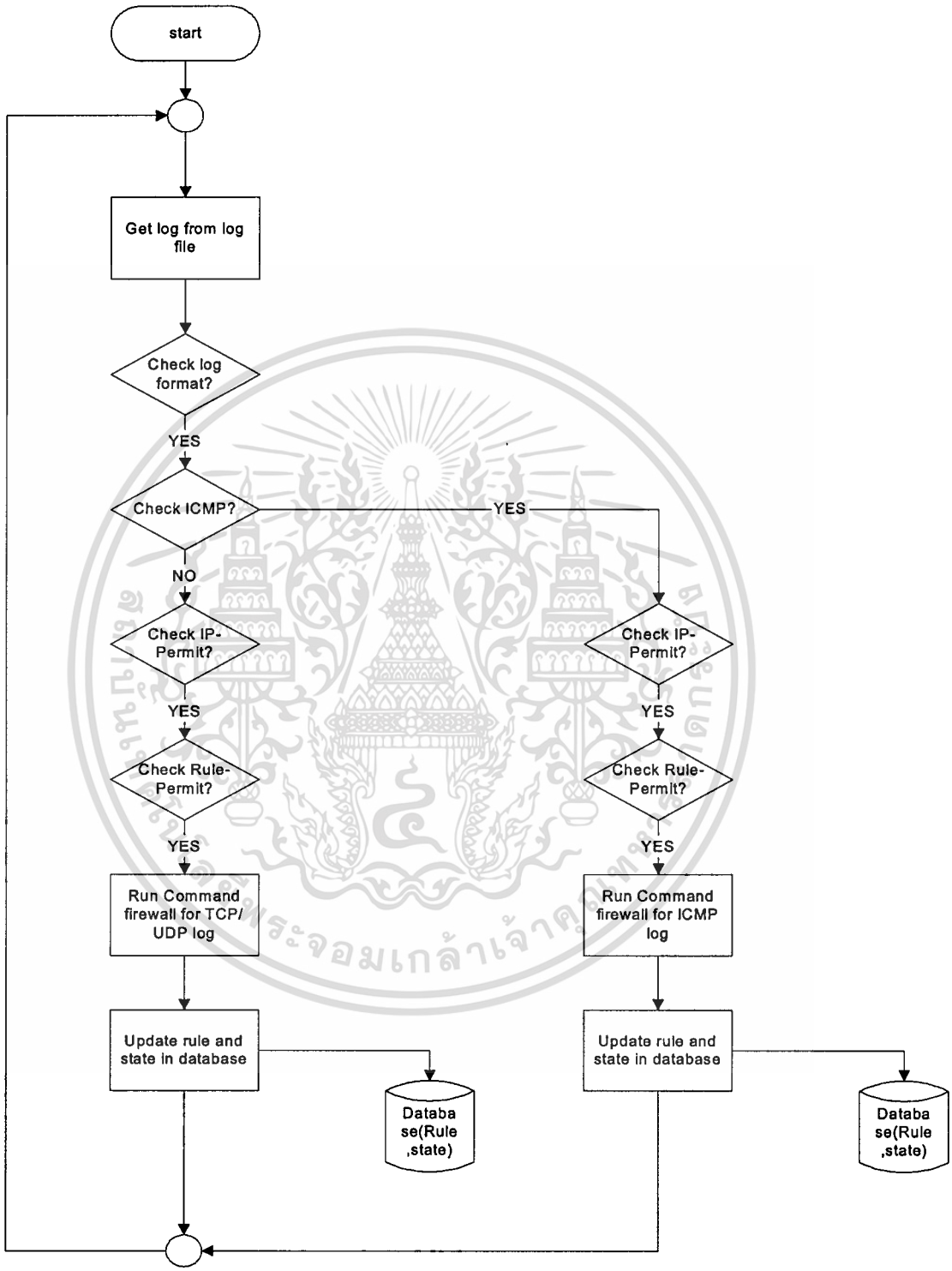
หลังจากติดตั้งไฟร์วอลล์ IPFW แล้ว ผู้พัฒนาสามารถงานไฟร์วอลล์บนระบบปฏิบัติการ FreeBSD ได้ ซึ่งเป็นการเลือกใช้ซอฟต์แวร์ Open Source ทั้งหมดที่ไม่ต้องเสียค่าใช้จ่าย เพียงแต่ต้องอาศัยความเชี่ยวชาญในระบบปฏิบัติการ FreeBSD และไฟร์วอลล์ IPFW อย่างไรก็ตามผู้ใช้งานควรจะต้องคำนึงถึงนโยบายทางด้านความปลอดภัยขององค์กรเป็นหลัก เพื่อให้ไฟร์วอลล์นั้นสามารถนำไปใช้งานได้อย่างมีประสิทธิภาพสูงสุด

บทที่ 4
การวิเคราะห์และออกแบบโปรแกรมปรับแต่งกฎของไฟร์วอลล์โดยอัตโนมัติ

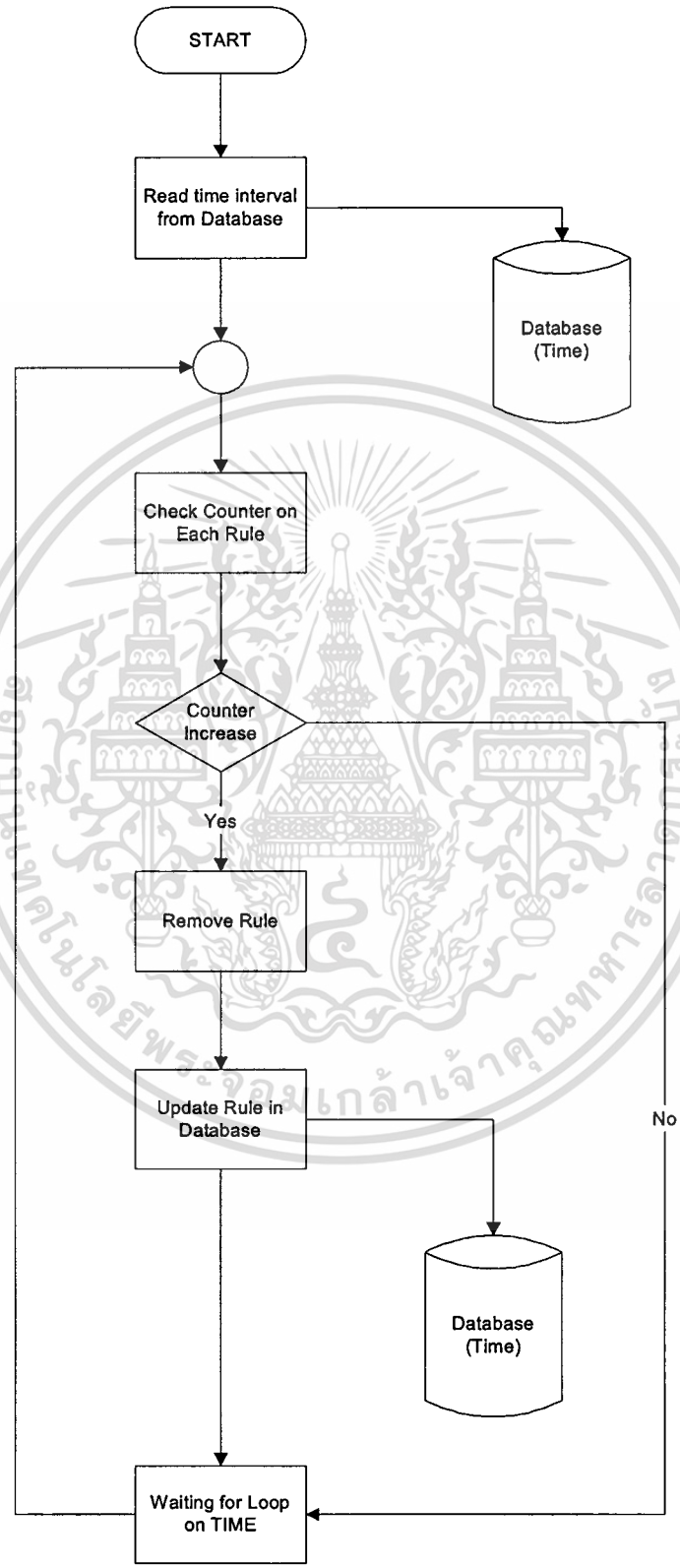
4.1 ขั้นตอนการทำงานของโปรแกรม

ได้แบ่งการทำงานของโปรแกรมออกเป็น 2 ส่วนใหญ่ๆ ดังนี้

1. การเพิ่มกฎของไฟร์วอลล์ โดยอัตโนมัติ มีขั้นตอนในการเพิ่มกฎของไฟร์วอลล์ดังนี้
   - ตรวจสอบล็อกของไฟร์วอลล์ IPFW
   - วิเคราะห์ล็อกของไฟร์วอลล์ IPFW ดูว่ามีการ deny ล็อกของเครื่องคอมพิวเตอร์ที่อยู่ใน Permit list หรือไม่
   - เมื่อพบว่ามีล็อกว่าเครื่องคอมพิวเตอร์ที่อยู่ใน Permit list ถูก deny ระบบก็จะทำการสร้างกฎของไฟร์วอลล์ขึ้นมาเพื่อรองรับกับการเชื่อมต่อนั้นๆ
2. การลบกฎของไฟร์วอลล์โดยอัตโนมัติ มีขั้นตอนในการลบกฎของไฟร์วอลล์ดังนี้
   - ตรวจสอบการใช้งานของกฎนั้นๆ ตามเวลาที่กำหนด
   - ตรวจสอบว่าเมื่อเวลาที่กำหนดแล้วกฎนั้นๆ ไม่ได้มีการใช้งานหรือไม่
   - ถ้ากฎนั้นไม่มีการใช้งานก็จะลบกฎนั้นทิ้งไป



รูปที่ 4.1 แสดงถึง Flow chart การทำงานของการเพิ่มกฎโดยอัตโนมัติ



รูปที่ 4.2 แสดงถึง Flow Chart การทำงานของการลบกฎโดยอัตโนมัติ

4.2 การออกแบบฐานข้อมูล

เนื่องจากระบบปรับแต่งกฎของไฟร์วอลล์โดยอัตโนมัตินั้นจำเป็นต้องมีการเก็บข้อมูลในส่วนของ Permit list เป็นต้น จึงจำเป็นต้องใช้ระบบฐานข้อมูลเพื่อมาเก็บข้อมูลต่างๆ เพื่อเป็นการง่ายและสะดวกในการใช้งาน โดยมีตารางต่างๆ ดังนี้

1. ตาราง ip_permit เก็บข้อมูลของ Permit list โดยมีฟิลด์ต่างๆ ดังที่แสดงในตารางที่ 4.1

ตารางที่ 4.1 แสดงฟิลด์ของตาราง ip_permit

ชื่อฟิลด์	คำอธิบาย
ip	หมายเลขไอพีของเครื่องที่จะอนุญาตให้ใช้งานได้
Desc	คำอธิบายของหมายเลขไอพี

2. ตาราง time เก็บข้อมูลของระยะเวลาที่โปรแกรมจะใช้ตรวจสอบตัวนับของกฎเพื่อทำการลบกฎ โดยมีฟิลด์ต่างๆ ดังที่แสดงในตารางที่ 4.2

ตารางที่ 4.2 แสดงฟิลด์ของตาราง time

ชื่อฟิลด์	คำอธิบาย
time	ระยะที่ให้โปรแกรมหน่วงเวลาในการตรวจสอบกฎ (วินาที)

3. ตาราง rule เก็บข้อมูลของกฎที่ระบบได้ทำการสร้างขึ้นมา โดยมีฟิลด์ต่างๆ ดังที่แสดงในตารางที่ 4.3

ตารางที่ 4.3 แสดงฟิลด์ของตาราง rule

ชื่อฟิลด์	คำอธิบาย
no	หมายเลขกฎ
ip_source	หมายเลขไอพีเครื่องต้นทาง
port_source	หมายเลขพอร์ทต้นทาง
ip_des	หมายเลขไอพีเครื่องปลายทาง
port_des	หมายเลขพอร์ทปลายทาง

ตารางที่ 4.3 แสดงฟิลด์ของตาราง rule (ต่อ)

ชื่อฟิลด์	คำอธิบาย
protocol	ชื่อโปรโตคอล
direction	ทิศทางของข้อมูล
interface	ชื่อของ Network Interface Card
day	วันที่สร้างกฎ
month	เดือนที่สร้างกฎ
year	ปีที่สร้างกฎ
hour	เวลาที่สร้างกฎ (ชั่วโมง)
minute	เวลาที่สร้างกฎ (นาที)
second	เวลาที่สร้างกฎ (วินาที)
in	ตัวนับข้อมูลขาเข้าของกฎ
out	ตัวนับข้อมูลขาออกของกฎ

บทที่ 5
การพัฒนาโปรแกรมระบบปรับแต่งกฎของไฟร์วอลล์โดยอัตโนมัติ

5.1 ซอฟต์แวร์สำหรับพัฒนาโปรแกรม

ในการพัฒนาโปรแกรมระบบปรับแต่งกฎของไฟร์วอลล์โดยอัตโนมัตินั้น แบ่งการทำงานออกเป็น 2 ส่วนดังนี้

- ส่วนทำงานติดต่อกับระบบไฟร์วอลล์ ซึ่งพัฒนาโดย Perl 5.8.0
- ส่วนทำงานควบคุมระบบ ซึ่งพัฒนาโดย PHP 4.3.2

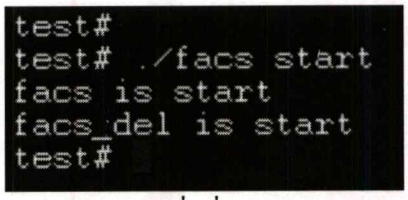
5.2 การพัฒนาโปรแกรมในส่วนการทำงานที่ติดต่อกับระบบไฟร์วอลล์

การพัฒนาโปรแกรมในส่วนการทำงานที่ติดต่อกับระบบไฟร์วอลล์พัฒนาโดยใช้โปรแกรมเพิร์ล โดยมีโมดูลพิเศษเพิ่มเข้ามาคือ โมดูล FileTail ซึ่งจะเป็นโมดูลที่ทำหน้าที่คอยนำข้อความที่เพิ่มขึ้นจากไฟล์ในขณะนั้นๆ ออกมา ในการพัฒนานี้เราก็จะใช้โมดูลนี้ คอยตรวจสอบไฟล์ล็อกของไฟร์วอลล์ โดยเมื่อโมดูล FileTail พบว่าไฟล์ล็อกนั้นมีข้อมูลเพิ่มเข้ามา ก็จะนำล็อกที่มาใหม่นั้นไปตรวจสอบกับฐานข้อมูล Permit List ว่าตรงกันหรือไม่ แล้วก็จะไปทำการเพิ่มกฎของไฟร์วอลล์โดยอัตโนมัติ

โดยที่โปรแกรมในส่วนของการสร้างกฎก็จะทำงานเป็นเดมอนไว้ตลอดเวลา โดยที่ระบบก็จะทำการเพิ่มกฎอย่างเดียวจะไม่มีการลบกฎ โดยการลบกฎก็จะเป็นหน้าที่ของโปรแกรมที่ทำหน้าที่ลบกฎของจากไฟร์วอลล์ โดยที่การลบกฎนั้นทำงานโดยคอยตรวจสอบตัวนับของกฎแต่ละกฎว่ามีการใช้งานกฎๆนั้นหรือไม่ ถ้าไม่มีการใช้งานกฎๆนั้นตามเวลาที่กำหนดก็จะลบกฎนั้นทิ้งไป โดยจะทำงานวนไปเรื่อยๆ เว้นช่วงตามระยะเวลาที่กำหนด

โปรแกรมทั้ง 2 โปรแกรมนั้นจะทำงานผ่าน shell script ซึ่งมีรูปแบบการทำงานดังนี้

- เริ่มต้นการทำงานของโปรแกรมด้วยคำสั่ง facs start ระบบจะทำการตรวจสอบว่ามีโปรแกรมนี้ทำงานอยู่บนระบบปฏิบัติการหรือไม่ ถ้าไม่มี script ก็จะทำการสั่งให้โปรแกรมเริ่มต้นการทำงานดังในรูปที่ 5.1

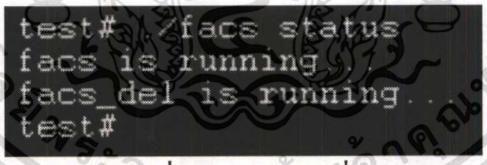


รูปที่ 5.1 แสดงผลจากคำสั่งเริ่มต้นทำงานของโปรแกรม

ในกรณีที่มีโปรแกรมนี้ทำงานอยู่ในระบบปฏิบัติการแล้ว script จะทำการแจ้งกลับมาว่ามีโปรแกรมทำงานอยู่แล้วดังรูปที่ 5.2

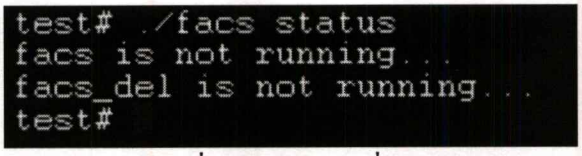
รูปที่ 5.2 แสดงผลของคำสั่งเริ่มต้นทำงานเมื่อมีโปรแกรมทำงานอยู่

- แสดงสถานะการทำงานของโปรแกรมด้วยคำสั่ง facs status เมื่อสั่งแล้ว script จะแสดงสถานะการทำงานของโปรแกรมว่าทำงานอยู่หรือไม่ดังรูปที่ 5.3 ที่จะแสดงว่ามีโปรแกรมทำงานอยู่



รูปที่ 5.3 แสดงผลของคำสั่งแสดงสถานะเมื่อมีโปรแกรมทำงานอยู่

แต่ถ้าใช้คำสั่ง facs status แล้วไม่มีโปรแกรมทำงานอยู่ script ก็จะแจ้งผลออกมาตามรูปที่ 5.4



รูปที่ 5.4 แสดงผลของคำสั่งแสดงสถานะเมื่อไม่มีโปรแกรมทำงานอยู่

- ยกเลิกการทำงานของโปรแกรมด้วยคำสั่ง facs stop เมื่อสั่งแล้ว script จะทำการสั่งให้โปรแกรมหยุดการทำงานแล้วจะแสดงผลออกมาทางหน้าจอดังรูปที่ 5.5

```
test# ./facs stop
facs is stopped
facs_del is stopped
test#
```

รูปที่ 5.5 แสดงผลของคำสั่งหยุดการทำงาน

แต่ในกรณีที่ไม่มีโปรแกรมทำงานอยู่ในระบบปฏิบัติการ script จะแสดงผลออกมาว่าไม่มีโปรแกรมทำงานอยู่ดังที่แสดงในรูปที่ 5.6

```
test# ./facs stop
facs is not running...
facs_del is not running...
test#
```

รูปที่ 5.6 แสดงผลของคำสั่งหยุดการทำงานโดยที่ไม่มีโปรแกรมทำงานอยู่

5.3 การพัฒนาโปรแกรมในส่วนควบคุมระบบ

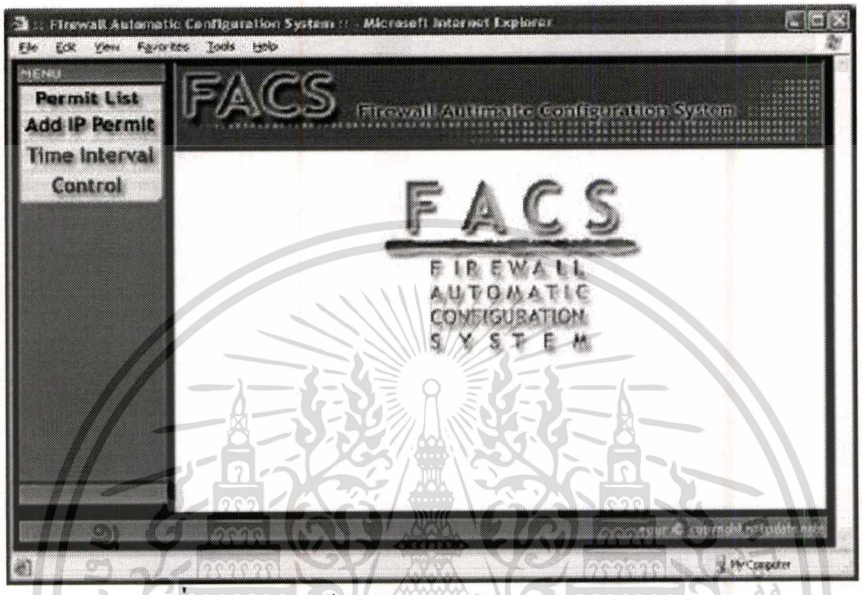
การพัฒนาโปรแกรมในส่วนควบคุมระบบนั้น พัฒนาโดยโปรแกรม PHP โดยให้โปรแกรมที่ได้พัฒนามานั้นทำงานผ่านเวปบราวเซอร์ โดยที่เครื่องลูกข่ายที่ใช้ทำการควบคุมไม่ต้องลงซอฟต์แวร์ใดๆเพิ่มเติม

โดยโปรแกรมส่วนควบคุมระบบนั้นจะทำหน้าที่ดังนี้คือ

1. เพิ่มไอพีแอดเดรสเข้าไปใน Permit list
2. ลบไอพีแอดเดรสใน Permit list
3. ค้นหาไอพีแอดเดรสใน Permit list
4. แก้ไขไอพีแอดเดรสใน Permit list
5. แก้ไขระยะเวลาที่โปแกรมจะตรวจสอบกฎที่ไม่ได้ใช้งาน
6. เริ่มต้นการทำงานระบบ
7. แสดงสถานะการทำงานของระบบ
8. หยุดการทำงานของระบบ

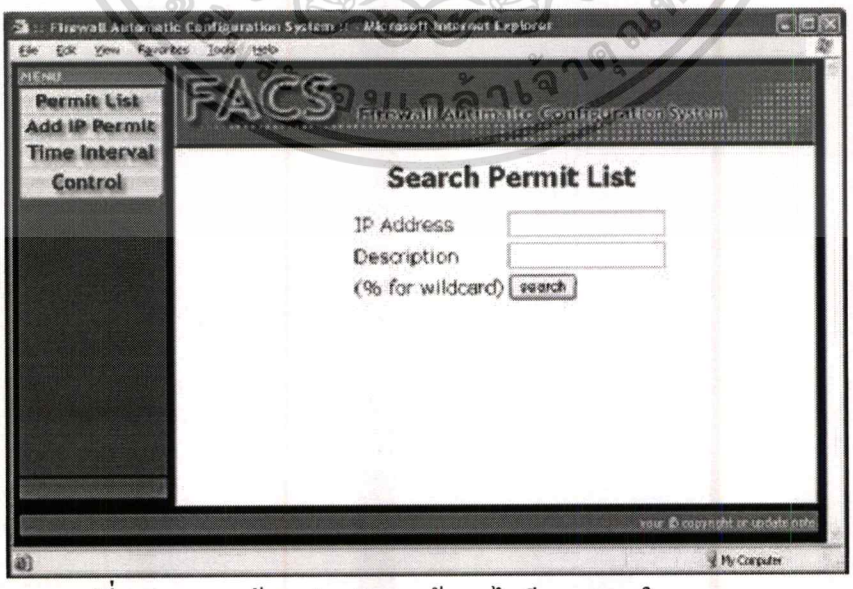
โดยมีรูปแบบการทำงานดังนี้

- โปรแกรมในส่วนควบคุมการทำงานของระบบปรับแต่งกฎของไฟร์วอลล์โดยอัตโนมัติ เมื่อเข้ามาที่ url ของระบบก็จะขึ้นหน้าจอดังรูปที่ 5.7



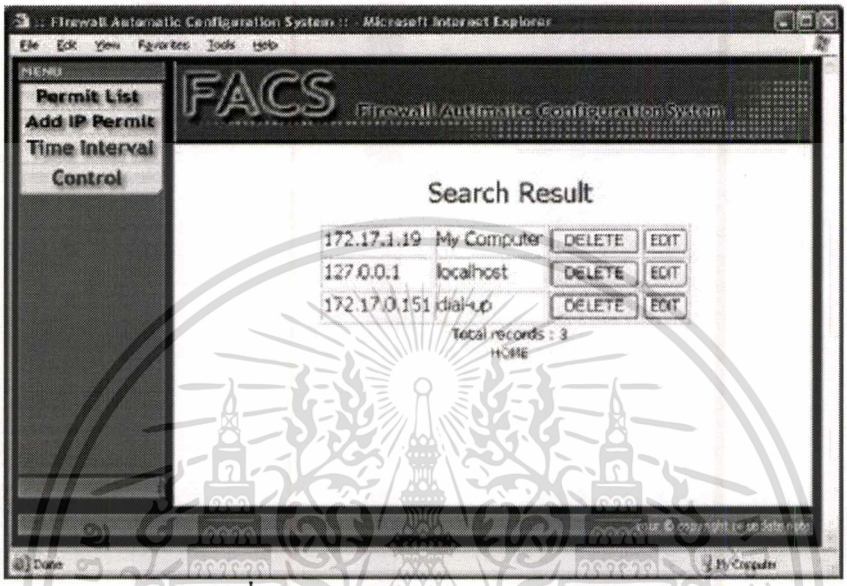
รูปที่ 5.7 แสดงหน้าจอหลักของโปรแกรมควบคุมระบบ

- การค้นหาไอพีแอดเดรสที่อยู่ใน Permit list เพื่อทำการลบ และแก้ไขทำได้โดยกดปุ่ม Permit List ที่อยู่บนเมนูด้านซ้ายมือจะปรากฏหน้าจอดังรูปที่ 5.8



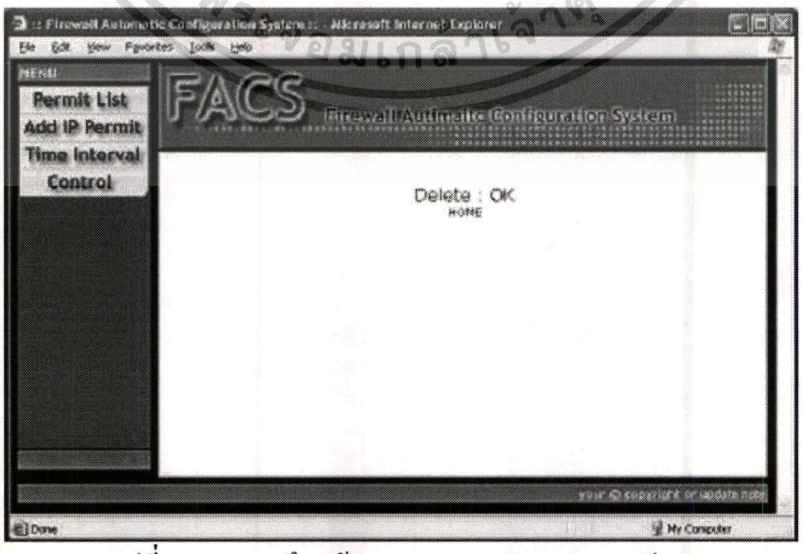
รูปที่ 5.8 แสดงหน้าจอสำหรับการค้นหาไอพีแอดเดรสใน Permit list

- เมื่อทำการค้นหาไอพีแอดเดรสหรือ คำอธิบายใน permit list ซึ่งจะสามารถใช้ % แทนคำใดๆได้ ก็จะปรากฏหน้าจอดังรูปที่ 5.9



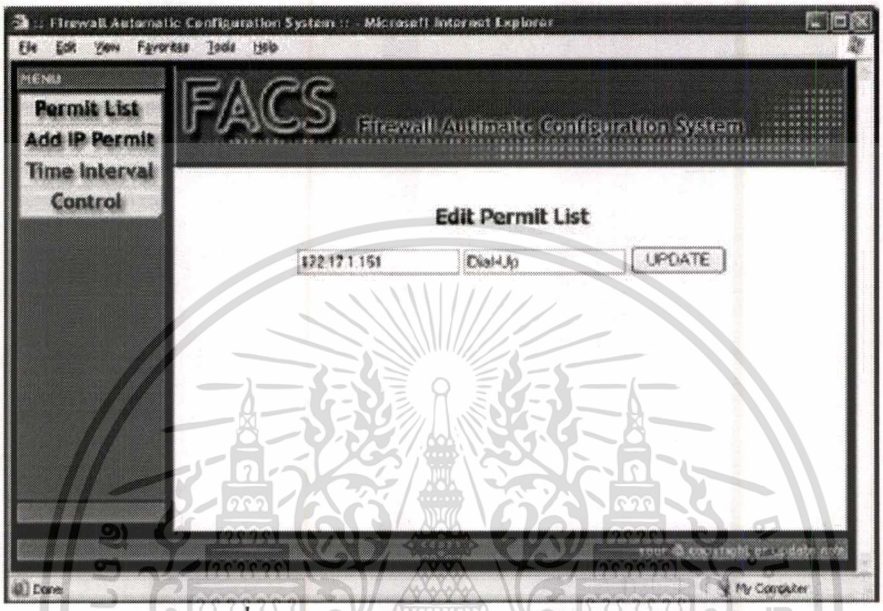
รูปที่ 5.9 แสดงหน้าจอแสดงผลการค้นหา

- หลังจากค้นหาเสร็จแล้วในหน้าจอผลของการค้นหาด้านท้ายของแต่ละระเบียนจะมีปุ่มสำหรับให้ลบ และแก้ไขข้อมูลในระเบียนนั้น ดังในรูปที่ 5.10 จะแสดงหน้าจอของการการลบ ข้อมูลในระเบียนนั้น



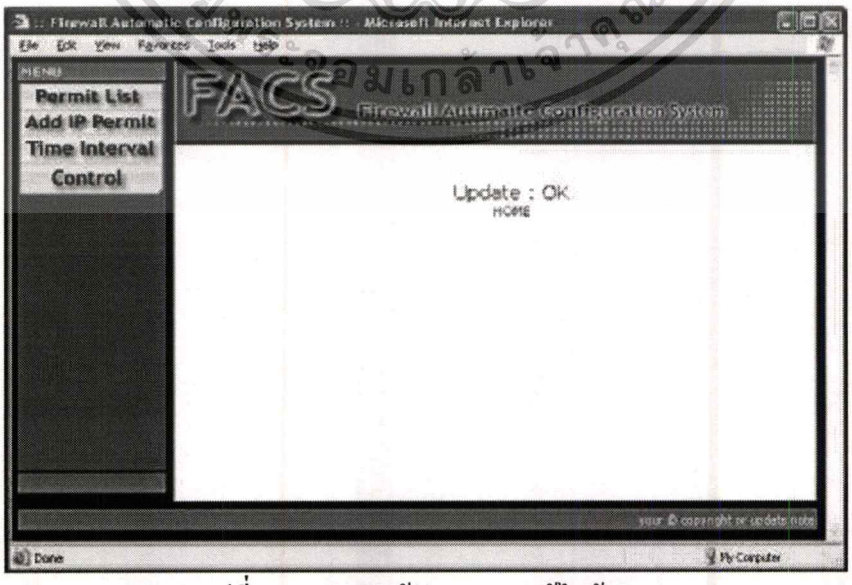
รูปที่ 5.10 แสดงถึงหน้าจอแสดงผลหลังการลบระเบียน

- แล้วถ้ากดปุ่ม Edit ที่อยู่หลังระเบียนนั้นๆ ก็จะเป็นการแก้ไขข้อมูลที่อยู่ในระเบียนนั้นๆ ซึ่งมีหน้าจอดังรูปที่ 5.11



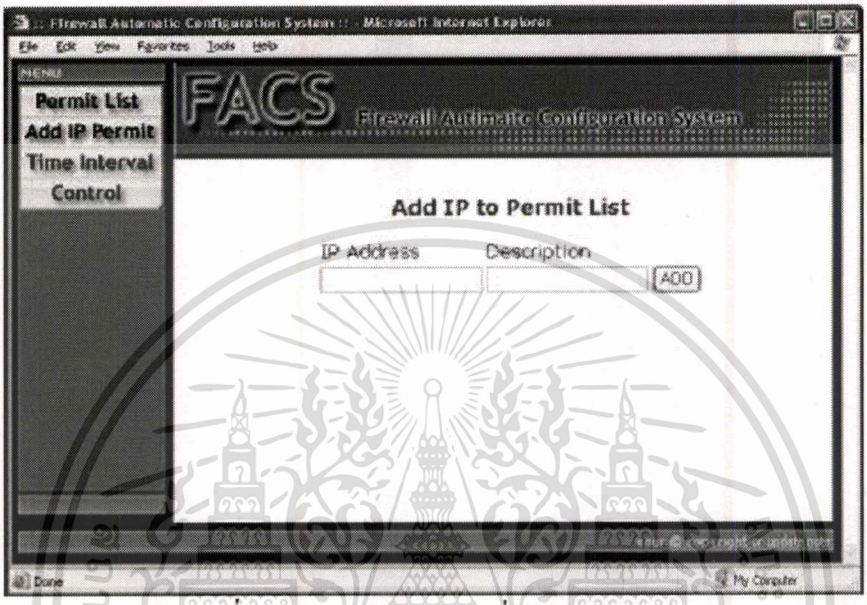
รูปที่ 5.11 แสดงหน้าจอการแก้ไข Permit List

- เมื่อแก้ไขข้อมูลเสร็จแล้วกดปุ่ม UPDATE ระบบก็จะแสดงผลดังรูปที่ 5.12 แสดงว่าการแก้ไขข้อมูลได้เสร็จเรียบร้อย



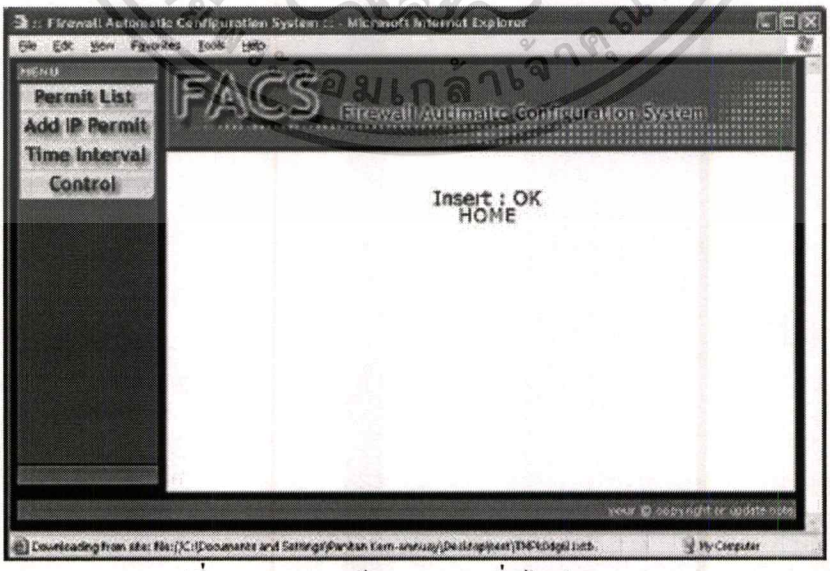
รูปที่ 5.12 แสดงหน้าจอผลการแก้ไขข้อมูล

- การเพิ่มไอพีแอดเดรสเข้าไปใน Permit list ทำได้โดยการกดปุ่ม Add IP Permit ในเมนูที่อยู่ทางซ้ายมือ จะปรากฏหน้าจอดังรูปที่ 5.13



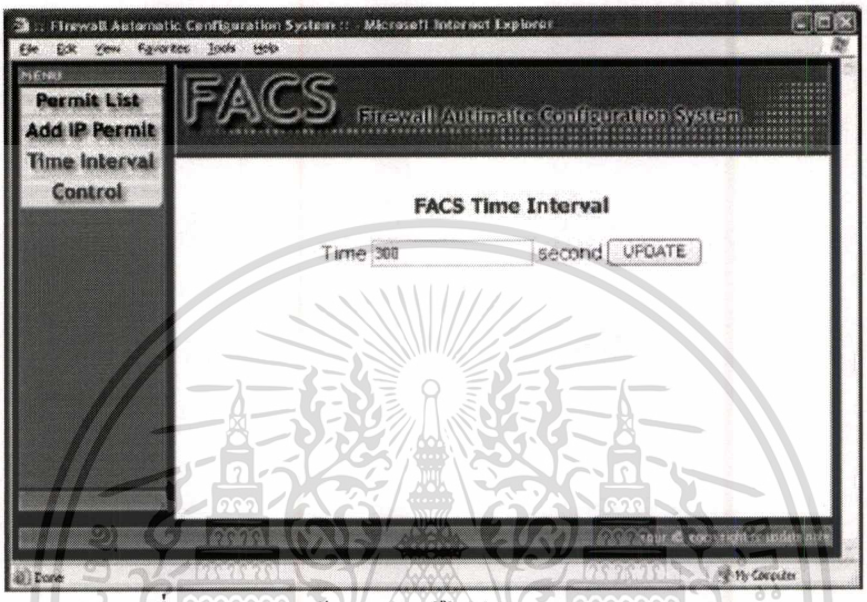
รูปที่ 5.13 แสดงหน้าจอการเพิ่มไอพีแอดเดรส

- เมื่อใส่ข้อมูลตามที่ต้องการแล้วกดปุ่ม ADD ระบบก็จะส่งผลมาว่าทำการเพิ่มข้อมูลเข้าไปเรียบร้อยแล้วดังรูปที่ 5.14



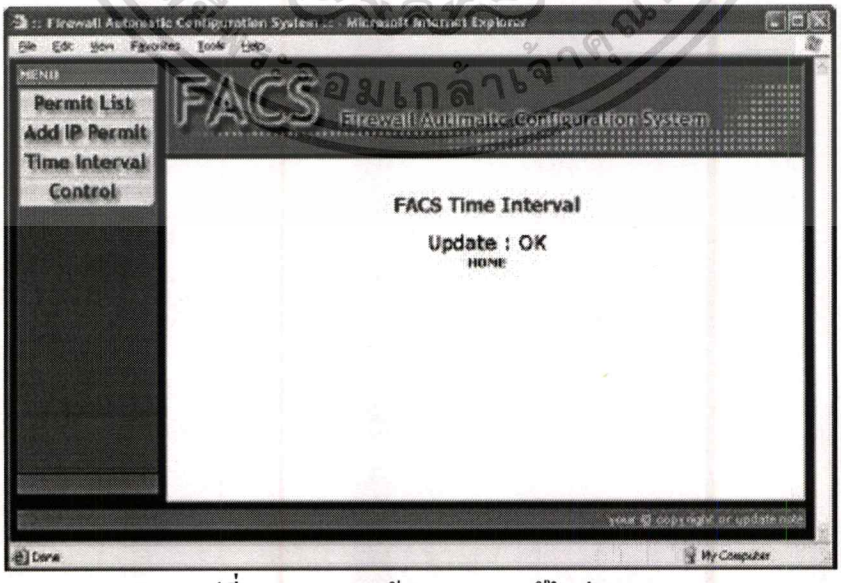
รูปที่ 5.14 แสดงหน้าจอผลการเพิ่มไอพีแอดเดรส

- การแก้ไขค่าระยะเวลาที่ให้ระบบคอยตรวจสอบกฎของไฟร์วอลล์ที่ไม่ได้ใช้งานทำได้โดยกดปุ่ม Time Interval ที่เมนูด้านซ้ายมือจะปรากฏหน้าจอดังรูปที่ 5.15



รูปที่ 5.15 แสดงหน้าจอการแก้ไขค่าระยะเวลาตรวจสอบกฎ

- เมื่อแก้ไขค่าเวลาแล้ว ทำการกดปุ่ม UPDATE ระบบก็จะทำการแก้ไขค่าข้อมูลตามที่ต้องการและแจ้งให้รู้ดังรูปที่ 5.16



รูปที่ 5.16 แสดงหน้าจอผลการแก้ไขค่าเวลา

- การเริ่มต้นการทำงานของระบบทำได้โดยกดปุ่ม CONTROL ที่เมนูด้านซ้ายมือจะปรากฏหน้าจอดังรูปที่ 5.17

รูปที่ 5.17 แสดงหน้าจอการควบคุมการเริ่มต้น ,สถานะและหยุดของระบบ

- และเมื่อจะทำการเริ่มต้นการทำงานของระบบก็กดปุ่ม Enable ก็จะมีผลการทำงานแสดงออกมาเหมือนกับการทำงานบนคอมมานด์ไลน์

บทที่ 6
บทสรุปและแนวทางพัฒนาในอนาคต

การติดต่อสื่อสารผ่านเครือข่ายอินเทอร์เน็ทนั้นเป็นระบบการสื่อสารแบบเปิด และมีเครื่องคอมพิวเตอร์จำนวนมากมายที่ติดต่อเข้ากับเครือข่ายอินเทอร์เน็ท โดยเราไม่มีทางรู้ได้เลยว่าเครื่องคอมพิวเตอร์ที่ติดต่อเข้ามาสู่เครือข่ายอินเทอร์เน็ทนั้น มีจุดประสงค์ใด อาจจะมีเครื่องคอมพิวเตอร์จำนวนมากที่ติดต่อเข้ามาสู่เครือข่ายอินเทอร์เน็ทเพื่อใช้งานในทางที่ดี ที่ถูกต้อง แต่ก็คงมีเครื่องคอมพิวเตอร์และผู้ใช้งานจำนวนไม่น้อยที่มุ่งประสงค์ร้ายต่อเครือข่ายและเครื่องคอมพิวเตอร์ที่ต่อเข้าสู่ระบบเครือข่ายอินเทอร์เน็ทอยู่ด้วยเหมือนกัน ดังนั้นจึงเป็นการไม่ปลอดภัยเลยถ้าเราเชื่อมต่อเครื่องคอมพิวเตอร์ให้บริการเข้าสู่อินเทอร์เน็ทโดยไม่มีการป้องกันใดๆ ระบบไฟร์วอลล์จึงเป็นหนึ่งในเทคโนโลยีทางด้านความปลอดภัยที่ถูกนำมาใช้เพื่อเพิ่มความปลอดภัยให้กับทรัพท์สินและข้อมูลที่สำคัญยิ่ง

ระบบไฟร์วอลล์เข้ามามีบทบาทอย่างมากในเรื่องความปลอดภัยของเครือข่ายคอมพิวเตอร์ที่มีการเชื่อมต่อเข้าสู่ระบบอินเทอร์เน็ท ส่วนมากจะมีการนำเอาไฟร์วอลล์เข้ามาใช้งานเพื่อเพิ่มความปลอดภัยของเครือข่าย แต่สำหรับเครือข่ายขนาดเล็กแล้วมักจะมองข้ามการนำไฟร์วอลล์หรือแม้แต่ในเรื่องของความปลอดภัยไป เพราะสำหรับผู้ดูแลระบบเครือข่ายขนาดเล็กแล้ว เรื่องความปลอดภัยอาจจะเป็นที่ยังไกลตัว และการที่จะนำระบบรักษาความปลอดภัยต่างๆเป็นเรื่องที่มีค่าใช้จ่ายและต้องใช้ความเชี่ยวชาญมาก ผู้ดูแลระบบเครือข่ายขนาดเล็กจึงมองข้ามระบบความปลอดภัยไป

การพัฒนาระบบปรับแต่งกฎของไฟร์วอลล์โดยอัตโนมัติขึ้นมาก็เพื่อให้ผู้ดูแลเครือข่ายขนาดเล็กหรือแม้แต่ผู้ดูแลเครือข่ายขนาดใหญ่ ที่ยังคิดว่าไฟร์วอลล์เป็นเรื่องที่ต้องเสียค่าใช้จ่ายและต้องอาศัยความเชี่ยวชาญมากนั้น จะได้นำระบบนี้ไปใช้ เพราะระบบนี้เป็นระบบที่ใช้ซอฟต์แวร์ที่เป็น Opens Source ทั้งหมดซึ่งไม่เสียค่าใช้จ่ายใดๆ และยังมีระบบที่สามารถปรับแต่งกฎของไฟร์วอลล์ได้โดยอัตโนมัติ ทำให้หมดปัญหาในเรื่องของค่าใช้จ่ายและความเชี่ยวชาญในระบบการทำงานของไฟร์วอลล์

อย่างไรก็ตามถึงแม้ไฟร์วอลล์จะทำงานได้ดีเพียงใด ไฟร์วอลล์เพียงอย่างเดียวก็ไม่สามารถที่จะรับประกันได้ว่ามีไฟร์วอลล์แล้วเครือข่ายของคุณจะปลอดภัยแน่นอน

ในเรื่องของความปลอดภัยยังต้องมีส่วนประกอบอีกมากมายเพื่อทำให้เครือข่ายมีความปลอดภัยมากที่สุด แต่การที่ยิ่งทำให้เครือข่ายความความปลอดภัยสูงมาขึ้นเท่าใดสิ่งที่จำต้องแลกมาด้วยความปลอดภัยนั้นก็คือ ราคาที่ต้องสูงขึ้นไปด้วย ดังนั้นแล้วก็คงจะต้องมองหาจุดที่สมดุลย์ที่สุดระหว่างความปลอดภัยที่เพียงพอและอยู่ในราคาที่ยอมรับได้

สำหรับแนวทางที่จะพัฒนาระบบปรับแต่งกฎของไฟร์วอลล์โดยอัตโนมัตินี้ อาจจะมีการพัฒนาให้มีกลไกในการเรียนรู้ที่เร็วขึ้นให้มากกว่านี้ และอาจจะมีการปรับให้สามารถใช้งานได้หลายระบบปฏิบัติการและใช้งานกับซอฟต์แวร์ไฟร์วอลล์ได้หลายชนิดมากยิ่งขึ้น

บรรณานุกรม

Zwicky, Elizabeth D. and Cooper, Simon. 2000. **Building Internet Firewalls.** 2nd Edition Sebastopol, California: O'Reilly & Associates.

Tiemann, Brian and Urban, Michael. 2003. **FreeBSD Unleashed** 2nd Edition. Indianapolis, Indiana: SAMS Publishing.

Lahey, Greg. 2003. **The Complete FreeBSD.** 4th Edition. Sebastopol, California: O'Reilly & Associates.

Randal, Schwartz L. 1994. **Learning Perl.** Sebastopol, California: O'Reilly & Associates.

Cheswick, William R. et.al. 2003. **Firewall and Internet Security: Repelling the wily hacker,** 2nd Edition. Boston, Massasusette: Addison-Wesley.

Rob, Peter and Coronel, Carlos. **Database Systems.** Florence, Kentucky: Thomson Lerning.

FreeBSD. 2003. **The FreeBSD Project.** [Online]. Available: http://www.freebsd.org

Defcon1. 2003. **IPFW How-To.** [Online] Available: http://www.defcon1.org/html/NATD-config/firewall-setup/ipfw-1.html

ภาคผนวก ก

1. การติดตั้งและปรับแต่งส่วนเซิร์ฟเวอร์ เตรียมสภาพแวดล้อมในการทำงาน

ลำดับ	ขั้นตอนการดำเนินการ	รายละเอียด
1	ติดตั้งระบบปฏิบัติการ FreeBSD 5.0	
2	ทำการปรับแต่งระบบโดยมิให้เปิดบริการใดๆที่ไม่ได้ใช้ในการทำงาน	
3	ติดตั้ง Secure Shell Server SSH 3.5p1	ใช้สำหรับควบคุมเครื่องจากระยะไกล
4	ติดตั้ง MySQL 3.23.54	
5	ติดตั้ง Perl 5.8.0	
6	ติดตั้ง PHP Extension 4.3.2	ติดตั้งให้ทำงานร่วมกับ MySQl ด้วย
7	ติดตั้ง Open SSL 0.9.6g	
8	ติดตั้ง Perl Module Fail-Tail	
9	ติดตั้ง Apache 2.0.46	ติดตั้งให้ apache ทำงานร่วมกับ php extension และ ssl ด้วย
10	สร้างไดเรคทอรี่ในเวปเซิร์ฟเวอร์สำหรับระบบ	

2. แหล่งดาวน์โหลดโปรแกรมที่ใช้งาน

ลำดับ	โปรแกรม	แหล่งดาวน์โหลด
1	ระบบปฏิบัติการ FreeBSD	http://www.freebsd.com
2	Secure Shell Server SSH	http://www.openssh.com
3	MySQL Server	http://www.mysql.com
4	Perl	http://www.perl.org
5	PHP Extension	http://www.php.net
6	OpenSSL	http://www.openssl.org

ลำดับ	ชื่อเรื่อง	ตำแหน่งที่มาของข้อมูล
7	Fail-Tail Perl Module	http://www.cpan.org/modules/01modules.index.html
8	Apache Web Server	http://www.apache.org

ประวัติผู้เขียน

ชื่อผู้เขียน	**นายปณิธาน เขินอำนวย**
วันเดือนปีเกิด	**8 มกราคม 2518**
สถานที่เกิด	**กรุงเทพมหานคร**
วุฒิการศึกษาระดับปริญญาตรี	วท.บ. (วิทยาการคอมพิวเตอร์)
สถานที่สำเร็จการศึกษา	**คณะวิทยาศาสตร์และเทคโนโลยี สถาบันราชภัฏส่วนสุนันทา**
ปีที่สำเร็จการศึกษา	2540