การพัฒนาโปรแกรมจัดการไอพีไฟร์วอลล์บนระบบปฏิบัติการฟรีบีเอสดีผ่านทางเว็บ

The Development of Web-based IP Firewall manager on FreeBSD

โดย

นายวรกุล เมืองสุวรรณ

รหัส 42067032

อาจารย์ที่ปรึกษา

อาจารย์อัครินทร์ คุณกิตติ

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงงานการพัฒนาระบบงาน

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 2 ปีการศึกษา 2543

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อหัวข้อ	การพัฒนาโปรแกรมจัดการไอพีไฟร์วอลล์บนระบบปฏิบัติการฟรีบีเอสดีผ่านทางเว็บ
นักศึกษา	นายวรกุล เมืองสุวรรณ
อาจารย์ที่ปรึกษา	อาจารย์อัครินทร์ คุณกิตติ
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2543

บทคัดย่อ

ไอพีไฟร์วอลล์บนระบบปฏิบัติการฟรีบีเอสดีเป็นไฟร์วอลล์แบบกรองแพ็คเก็ต โดยจะอาศัยกฎหรือนโยบายในการควบคุมและป้องกันการเข้าออกของแพ็คเก็ต ไอพีไฟร์วอลล์มีการกระทำหลายแบบได้แก่ การอนุญาตหรือปฏิเสธแพ็คเก็ตที่เข้ามาสู่ระบบโดยจะพิจารณาองค์ประกอบสำคัญอันได้แก่ แอดเดรสต้นทาง แอดเดรสปลายทาง โปรโตคอล และพอร์ต มาใช้ในการตัดสินใจ เพื่อกำหนดนโยบายหรือกฎ ในโครงงานนี้ได้มีการใช้รูปแบบภาษาพอนเดอร์เพื่อกำหนดนโยบายต่าง ๆ โดยจะอาศัยจุดเด่นของตัวภาษานี้มาพิจารณาแก้ปัญหาต่างๆที่เกิดขึ้นในการกำหนดนโยบาย เช่นการซ้ำกันของกฎและการขัดแย้งกัน โดยจะได้มีการพัฒนาโปรแกรมเพื่อมาตีความภาษาพอนเดอร์นี้ให้อยู่ในรูปแบบที่ไอพีไฟร์วอลล์เข้าใจ และมีการสร้างฟังก์ชันการทำงานต่าง ๆ ทั้งฟังก์ชันทั่วไปที่มีอยู่แล้วในไอพีไฟร์วอลล์และฟังก์ชันที่เพิ่มเติมขึ้นมา โดยจะพัฒนาโปรแกรมที่จะติดต่อกับผู้ใช้ผ่านทางเว็บ ซึ่งจะสร้างความสะดวกให้กับผู้ใช้ที่ไม่ต้องมาทำที่เครื่องที่ให้บริการโดยตรง และยังมีความคุ้นเคยการใช้งานผ่านทางเว็บทำให้งานต่อการเข้าใจ

Title	The Development of Web-based IP Firewall manager on FreeBSD
Student	Mr.Woragoon Muangsuwan
Advisor	Mr.Akharin Khunkitti
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2000

ABSTRACT

IP Firewall or IPFW, the software supplied with FreeBSD, is a packet filtering system which resides in the kernel. It can decide to allow or deny these packets that matched the policies or rules that administrator has defined, to consider with source address, destination address, protocol and port. In the project, used the ponder language to manage and define the policy that used the advantage of it: duplicate proving and conflict solving. The project has compiler engine to compile the ponder syntax and transform it to IPFW syntax and construct common function and extended function to increase capability of using IPFW. It use web browser to interact with user that he/she is intimately acquainted.

กิตติกรรมประกาศ

ในการพัฒนาโปรแกรมจัดการไอพีไฟร์วอลล์บนระบบปฏิบัติการฟรีบีเอสดีผ่านทางเว็บนี้ ต้องอาศัยแหล่งความรู้ต่าง ๆ คำแนะนำและที่ปรึกษาทั้งในภาคทฤษฎีและภาคปฏิบัติ อุปกรณ์ฮาร์ดแวร์และซอฟร์ตแวร์ที่จำเป็นทั้งหลาย ตลอดจนกำลังใจและแรงที่ได้จากบุคคลต่าง ๆ ที่สมควรได้รับความขอบคุณเป็นพิเศษดังนี้

1. คุณพ่อ คุณแม่ ผู้ให้กำเนิด เลี้ยงดู เอาใจใส่เป็นอย่างดี ตลอดจนส่งเสริมด้านการศึกษาอย่างดีที่สุด
2. อาจารย์อัครินทร์ คุณกิตติ อาจารย์ที่ปรึกษาผู้ให้คำแนะนำตลอดการทำโครงงานนี้ และจัดหาทรัพยากรในการดำเนินงานให้แก่ข้าพเจ้า
3. พี่กิตติ รุ่นพี่ IS 3 พี่เจี๊ยบ IS 5 และเพื่อน ๆ IS 7 ทุกคนที่คอยเป็นกำลังใจและให้คำปรึกษาที่ดีมาตลอด

ทั้งหมดนี้ถูกรวบรวมได้จากหลายฝ่าย เพื่อให้การศึกษาและพัฒนาโปรแกรมเป็นไปโดยสำเร็จ

นายวรกูล เมืองสุวรรณ
ผู้จัดทำ

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญภาพ

หน้า

ภาพที่

สารบัญภาพ (ต่อ)

หน้า

ภาพที่

สารบัญภาพ (ต่อ)

หน้า

ภาพที่

บทที่1
บทนำ

การเติบโตอย่างต่อเนื่องในโลกของอินเตอร์เน็ตในปัจจุบันนี้ การติดต่อสื่อสารทางอินเตอร์เน็ตนั้นมีการใช้เว็บบราวเซอร์เป็นเครื่องมือในการที่จะให้ผู้ใช้สามารถใช้ในการติดต่อสื่อสารถึงกันหรือใช้บริการต่างๆที่มีอยู่เป็นจำนวนมากบนอินเตอร์เน็ตได้อย่างสะดวก ซึ่งเป็นที่นิยมใช้กันในหมู่ผู้ใช้เป็นจำนวนมาก เนื่องจากมีการเปิดให้ผู้ใช้เข้ามาใช้บริการได้เป็นจำนวนมากและจากทุกๆทิศทางแล้ว จึงต้องมีระบบรักษาความปลอดภัยเครือข่ายเพื่อให้เกิดความปลอดภัยต่อทรัพยากรที่อยู่ในระบบ ในโครงงานนี้จึงได้มีการการพัฒนาโปรแกรมจัดการไอพีไฟร์วอลล์บนระบบปฏิบัติการฟรีบีเอสดีผ่านทางเว็บ ซึ่งเป็นระบบรักษาความปลอดภัยอย่างหนึ่งที่ระบบปฏิบัติการฟรีบีเอสดีได้ให้บริการไว้แก่ผู้ใช้ (เวอร์ชันของฟรีบีเอสดีที่ใช้ในโครงงานนี้คือ เวอร์ชัน 4.0) สำหรับป้องกันการรุกรานทรัพยากรในเครือข่ายของตน ดังนั้นรายงานนี้จะกล่าวถึงความเป็นมาและความสำคัญของปัญหา วัตถุประสงค์ของการพัฒนาระบบงาน เป้าหมายของการพัฒนาระบบงาน ประโยชน์ที่คาดว่าจะได้รับ ขอบเขตของการพัฒนาระบบงาน ขั้นตอนการทำงาน ทฤษฎีที่นำมาใช้ในการพัฒนาระบบงาน และรายละเอียดในบทต่างๆ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันโลกของอินเตอร์เน็ตได้มีการใช้อย่างกว้างขวางซึ่งทำให้มีการใช้ข้อมูลต่างๆร่วมกันอย่างมากมายซึ่งก็จะเป็นประโยชน์อย่างมากสำหรับผู้ที่เข้าไปใช้ แต่เมื่อมีการใช้งานร่วมกันหรือมีการเปิดเผยข้อมูลสู่สาธารณะ ก็ทำให้เกิดปัญหาตามมา ปัญหาหนึ่งที่เป็นอันตรายต่อองค์กรคือมีการบุกรุกจากเครือข่ายภายนอกที่มีการรักษาความปลอดภัยน้อยเข้ามายังเครือข่ายขององค์กรที่ต้องการรักษาความปลอดภัย ดังนั้นจึงมีการใช้ไฟร์วอลล์ (Firewall)มาใช้ในการแก้ไขปัญหานี้ โดยจะนำมาใช้ในการรักษาความปลอดภัยภายในองค์กรเอง ซึ่งไฟร์วอลล์ที่นิยมใช้ในปัจจุบันมี 2 ประเภท คือ ไฟร์วอลล์แบบกรองแพ็คเก็ต (Packet filtering Firewall) และไฟร์วอลล์แบบบริการพร็อกซี่ (Proxy service Firewall) ซึ่งไฟร์วอลล์ทั้งสองจะได้อธิบายในรายงานต่อไป และได้มีการพัฒนาไฟร์วอลล์ในระบบปฏิบัติการต่างๆ เช่นบนระบบปฏิบัติการยูนิกซ์ (UNIX) ระบบปฏิบัติการวินโดว์เอ็นที (Windows NT)

(FreeBSD) เรียกว่า "ไอพีไฟร์วอลล์" (IP Firewall หรือ IPFW) โดยจะอาศัยหลักการในการกรองแพ็คเก็ต (Packet filtering) เอาไว้ โดยที่ไอพีไฟร์วอลล์จะอยู่ในเคอร์เนล (Kernel) ซึ่งผู้ดูแลสามารถที่จะเข้าไปจัดการในส่วนของกฎ เช่นการนิยามกฎ การสอบถามถึงกฎที่ใช้อยู่ในปัจจุบัน โดยจะมีรายการของกฎ(Rule List) เป็นตังควบคุมการเข้าออกของแพ็คเก็ตที่ผ่านเครือข่ายภายในกับเครือข่ายภายนอก โดยจะระบุการยอมรับหรือการปฏิเสธแพ็คเก็ตนั้นๆ และแต่ละแพ็คเก็ตจะถูกกรองโดยอาศัยคำสำคัญ(Keyword) ซึ่งเป็นฟิลด์(Feild) ที่อยู่ในแพ็คเก็ตเช่น แอดเดรสต้นทาง (Source Address) แอดเดรสปลายทาง(Destination Address) พอร์ตต้นทาง(Source Port) และพอร์ตปลายทาง(Destination Port) เป็นต้น ผู้ใช้สามารถเพิ่ม ลบ และแก้ไข โดยทำคำสั่งตามรูปแบบที่กำหนดไว้ ดังนั้นผู้ใช้จะต้องรู้รูปแบบของคำสั่งต่างๆ จึงจะสามารถที่จะเข้าไปกระทำกับกฎต่างๆ เหล่านั้นได้ และในระบบปฏิบัติการฟรีบีเอสดียังมีเครื่องมือในการรักษาความปลอดภัยอื่นๆ แต่ไม่ได้ใช้หลักการในการรักษาความปลอดภัยเหมือนไฟร์วอลล์ จะเห็นว่าเป็นการกระทำที่ยุ่งยากและเป็นคำสั่งเฉพาะซึ่งการทำงาน จะเป็นการทำงานแบบเท็กซ์โหมด(Text Mode) ที่จะต้องกระทำกับเครื่องที่มีระบบปฏิบัติการฟรีบีเอสดีอยู่ จึงได้มีแนวความคิดในการพัฒนาโปรแกรมจัดการไอพีไฟร์วอลล์บนระบบปฏิบัติการฟรีบีเอสดีผ่านทางเว็บให้มีความสะดวกสบายมากขึ้นกับผู้ใช้ และยังได้มีการนำเอาภาษาพอนเดอร์(ซึ่งจะได้กล่าวในภายหลัง)มาใช้ในการกำหนดนโยบายในการรักษาความปลอดภัยและได้มีการเพิ่มเติมรูปแบบภาษาบางส่วนให้มีความสอดคล้องกับไอพีไฟร์วอล์ เพื่อจะได้เป็นมาตราฐานในการกำหนดนโยบายหรือกฎในการรักษาความปลอดภัยต่อไป รวมทั้งมีเครื่องมืออื่นๆที่ประกอบในการทำงานของไอพีไฟร์วอลล์ให้มีประสิทธิภาพมากยิ่งขึ้นด้วยเช่น การตั้งเวลาในการกรองแพ็คเก็ต(Scheduling) การวิเคราะห์กฎขั้นพื้นฐานเพื่อไม่ให้กฎมีการซ้ำกันและพิจารณาถึงความซ้ำซ้อนของกฎ(Basic Rule Analysis) การตรวจสอบแพ็คเก็ตที่เข้ามา(Monitoring) และการจัดการนโยบาย โดยจะมีการสร้างนโยบายแล้วกระจายออกเป็นกฎต่างๆ รวมทั้งการเพิ่มและลดสมาชิกภายในกลุ่มด้วย(Policy Management)

1.2 วัตถุประสงค์ในการพัฒนาระบบงาน

- เพื่อศึกษาฟังก์ชันการทำงานของไอพีไฟร์วอลล์
- เพื่อวิเคราะห์และออกแบบ รวมถึงการพัฒนาโปรแกรมที่ควบคุมไอพีไฟร์วอลล์บนระบบปฏิบัติการฟรีบีเอสดีผ่านทางเว็บ
- เพื่อลดความยุ่งยากในการจัดการกับคำสั่งไอพีไฟร์วอลล์
- เพื่อให้มีภาษาที่ใช้ในการกำหนดนโยบายและกฎที่เป็นมาตรฐานสามารถใช้ได้กับทุกระบบเพียงแต่เปลี่ยนตัวแปลภาษาให้เหมาะสมกับระบบเท่านั้น

- เพื่อให้สามารถเข้าไปจัดการกับไอพีไฟร์วอลล์ได้จากทุกที่ไม่จำเป็นว่าจะต้องอยู่ที่เครื่องที่เป็นผู้ให้บริการ

1.3 เป้าหมายของการพัฒนาระบบงาน

- โปรแกรมจัดการไอพีไฟร์วอลล์ที่สร้างขึ้นจะเป็นสคริพท์(Script) ที่รับอินพุท/เอาท์พุตจากเว็บบราวเซอร์ที่ทำงานอยู่ในเว็บเซิร์ฟเวอร์และจะติดต่อกับระบบปฏิบัติการฟรีบีเอสดีที่เป็นระบบเดียวกัน
- โปรแกรมไอพีไฟร์วอลล์สามารถกรองแพ็คเก็ตได้ตามเงื่อนไขของนโยบายหรือกฎที่กำหนดให้อย่างถูกต้อง
- ผู้ใช้สามารถตั้งเวลาในการกรองแพ็คเก็ตและสามารถตรวจสอบดูแพ็คเก็ตที่เข้ามาในระบบได้
- ผู้ใช้สามารถกำหนดความสำคัญ(Priority)ของนโยบายหรือกฎได้
- โปรแกรมไอพีไฟร์วอลล์สามารถบันทึกนโยบายหรือกฎที่ผู้ใช้ใส่เข้าไปและแสดงเหตุการณ์ที่เกี่ยวข้องกับการกรองแพ็คเก็ตไว้ในไฟล์(File)

1.4 ประโยชน์ที่คาดว่าจะได้รับ

ผู้ใช้มีความสะดวกสะบายในการใช้งานมากขึ้น เนื่องจากสามารถที่จะไปจัดการไอพีไฟร์วอลล์ที่เครื่องใดก็ได้ อีกทั้งยังสามารถใช้รูปแบบของภาษาในการจัดการทางด้านความปลอดภัยให้สามารถใช้ได้กับทุกระบบเพียงแต่เปลี่ยนตัวแปลภาษาเท่านั้น รวมไปถึงมีฟังก์ชันการทำงานต่างๆที่เพื่อเติมขึ้นมาเพื่อให้การจัดการไอพีไฟร์วอลล์มีประสิทธิภาพและถูกนำไปใช้ประโยชน์อย่างแพร่หลายมากขึ้น

1.5 ขอบเขตของการพัฒนาระบบงาน

ไอพีไฟร์วอลล์โปรแกรมได้แบ่งการทำงานออกเป็น 2 ฟังก์ชันหลักใหญ่คือ ฟังก์ชันการทำงานทั่วไปกับฟังก์ชันในการทำงานที่เพิ่มเติม

1) ฟังก์ชันการทำงานทั่วไป
   - การเพิ่มกฎใหม่เข้าไป
   - การลบกฎที่ไม่ต้องการออก
   - การลบกฎทั้งหมดออก

- การแสดงกฎที่มีทั้งหมด
- การลบค่าตัวนับแพ็คเก็ต

2) ฟังก์ชันการทำงานที่เพิ่มเติม

- การตั้งเวลาการกรองแพ็คเก็ตในช่วงเวลาที่สามารถกำหนดเองได้
- การมอนิเตอร์(Monitor)จำวนวนแพ็คเก็ต
- การวิเคราะห์กฎขั้นพื้นฐาน เพื่อไม่ให้เกิดความซ้ำกันและความขัดแย้งกัน
- การจัดกลุ่มที่เป็นนโยบายแล้วกระจายออกเป็นกฎต่างๆ

โดยฟังก์ชันการทำงานต่างๆเหล่านี้จะถูกกระทำโดยผ่านรูปแบบของภาษา(Syntax) โดยในโครงงานนี้เราจะใช้รูปแบบภาษาพอนเดอร์(Ponder)เป็นภาษาในการกำหนดนโยบายและกฎรวมทั้งมีการออกแบบรูปแบบภาษาเพิ่มเติมเข้าไปเพื่อให้สามารถใช้ได้กับโครงงานนี้ ส่วนรายละเอียดของภาษาพอนเดอร์(Ponder)จะได้อธิบายในบทต่อไป

ขอบเขตของรูปแบบของภาษาพอนเดอร์(Ponder)ที่นำมาใช้ในโครงงานนี้แยกได้ดังนี้

- การกำหนดนโยบายของสิทธิ(Authorisation policies)
- การกำหนดนโยบายในการกรองสารสนเทศ(information Filtering Policy)
- การจัดการกับกลุ่มของโดเมน(Domain Scope Expression)
- การกำหนดลำดับความสำคัญของนโยบายหรือกฎและการจัดการนโยบายหรือกฎ

1.6 ทฤษฎีที่ใช้ในการพัฒนาระบบงาน

- หลักการของโปรโตคอลทีซีพี/ไอพี(TCP/IP) โดยเน้นไปที่โปรโตคอลไอพี(IP) ทีซีพี (TCP) และยูดีพี(UDP)
- การทำงานของไฟร์วอลล์โดยทั่วไป และลักษณะการทำงานของไฟร์วอลล์แบบกรองแพ็คเก็ตรวมถึงพร็อกซี่บริการ
- ลักษณะของฟรีบีเอสดี ซึ่งเป็นระบบปฏิบัติการยูนิกต์แบบหนึ่งบนเครื่องพีซี
- การทำงานของไอพีไฟร์วอลล์ โดยที่ไอพีไฟร์วอลล์เป็นฟังก์ชันหนึ่งที่ระบบปฏิบัติการฟรีบีเอสดีได้จัดหาไว้สำหรับการรักษาความปลอดภัย
- หลักการของภาษาในการกำหนดนโยบายแบบพอนเดอร์(The Ponder Policy Specification Language)

- นโยบายในการจัดการและภาษาในการกำหนดความปลอดภัยของพอนเดอร์สำหรับระบบเครือข่ายแบบกระจาย(Ponder :A Language for Specifying Security and Management Policies for Distributed Systems)

1.7 ขั้นตอนในการพัฒนาระบบงาน

- ศึกษาการทำงานของไอพีไฟร์วอลล์
- มองถึงปัญหาที่เกิดขึ้นและข้อจำกัดต่างๆที่เกิดขึ้นในการใช้งานของฟังก์ชันไอพีไฟร์วอลล์
- ศึกษาการทำงานของไฟล์วอลล์แบบกรองแพ็คเก็ตและระบบปฏิบัติการฟรีบีเอสดี
- หาแนวทางในการแก้ไขปัญหาที่เกิดขึ้น
- ศึกษาถึงหลักการหรือทฤษฎีที่เราจะนำมาแก้ไข(ในที่นี้คือ การศึกษาถึงรูปแบบของภาษาพอนเดอร์(Ponder))ปัญหาและหาข้อจำกัดในทฤษฎีที่เราจะนำมาใช้แก้ขปัญหาเพื่อออกแบบภาษาบางส่วนให้เหมาะสมกับงานของเรา
- ออกแบบโปรแกรมจัดการไอพีไฟร์วอลล์ให้มีความสามารถตามฟังก์ชันที่เราได้กำหนดไว้
- ศึกษาถึงเครื่องมือที่เราจะนำมาใช้ในการพัฒนาระบบงาน
- พัฒนาโปรแกรมจัดการไอพีไฟร์วอลล์ พร้อมทั้งสร้างเอกสารประกอบการพัฒนาโปรแกรม
- ทดสอบการใช้งานของโปรแกรมที่พัฒนาแล้ว
- สรุปผลการทดสอบจากการใช้งานที่เกิดขึ้น

1.8 รายละเอียดของแต่ละบท

<u>บทที่2</u> เป็นบทที่รวบรวมทฤษฎีต่างๆ ซึ่งเกี่ยวข้องกับไอพีไฟร์วอลล์ โดยจะกล่าวถึงโปรโตคอลทีซีพี/ไอพี ไฟร์วอลล์แบบกรองแพ็คเก็ต ระบบปฏิบัติการฟรีบีเอสดี ไอพีไฟร์วอลล์ และทฤษฎีพอนเดอร์ที่เป็นรูปแบบของภาษาที่มาจัดการรักษาความปลอดภัยในระบบเครือข่าย

<u>บทที่3</u> สำหรับบทนี้จะกล่าวถึงการออกแบบระบบทั้งหมด แสดงผังการไหลของข้อมูล ฐานข้อมูล และขั้นตอนในการตีความของภาษาพอนเดอร์

<u>บทที่4</u> การพัฒนาโปรแกรม จะแสดงความสัมพันธ์ขององค์ประกอบต่างๆ ภายในโปรแกรมได้แก่ ฟังก์ชันการทำงานต่างๆ โฟลชาร์ทแสดงการทำงานในฟังก์ชันต่าง ๆ

<u>บทที่5</u> กล่าวถึงการสรุปผลการทดลองการพัฒนาโปรแกรม โดยจะมีการกำหนดสถานการณ์ต่างๆ และแสดงผลการทดลอง และข้อเสนอแนะ

บทที่2
ไอพีไฟร์วอลล์

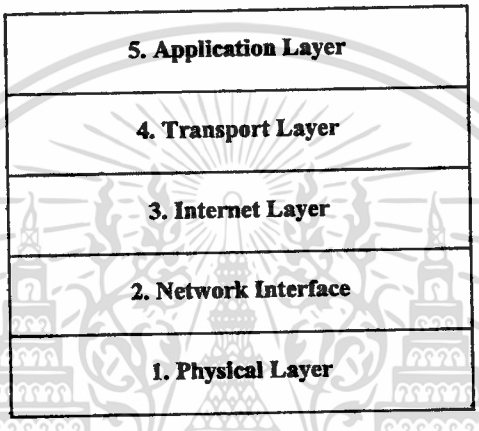
ในการสร้างระบบงานขึ้นมานั้น สิ่งแรกที่จะต้องทำความเข้าใจอันดับแรกคือทำการศึกษาให้เข้าใจถึงการทำงานและทฤษฎีต่างๆที่เกี่ยวข้องของระบบที่จะพัฒนา เพื่อศึกษาถึงข้อดีและข้อเสียขององค์ประกอบต่างๆให้เข้าใจและจะได้นำสิ่งต่างๆที่ได้ศึกษารวบรวมเหล่านั้นมาออกแบบและทำการสร้างระบบงานให้ตรงตามกับความต้องการ ในโครงงานนี้ได้นำหลักการและทฤษฎีต่างๆที่มีอยู่แล้วและเป็นผู้คิดค้นขึ้นมาเพื่อให้เหมาะสมกับโครงงานที่จะพัฒนาได้แก่ การทำงานของโปรโตคอลทีซีพี/ไอพี โดยเฉพาะชั้นอินเตอร์เน็ตและชั้นทรานสปอร์ต เนื่องจากชั้นทั้งสองนี้เกี่ยวข้องกับโปรโตคอลไอพี ทีซีพี และยูดีพี และนอกจากโปรโตคอลเหล่านี้แล้วก็ยังมีอีกหลายโปรโตคอลที่ให้บริการอยู่ ในส่วนของไฟร์วอลล์จะเน้นไฟร์วอลล์ในระดับพื้นฐาน นั้นคือไฟร์วอลล์แบบกรองแพ็คเก็ต ซึ่งเป็นไฟร์วอลล์ที่อาศัยกฎหรือนโยบายเป็นตัวกำหนดการทำงาน หัวข้อถัดไปจะอธิบายระบบปฏิบัติการฟรีบีเอสดีซึ่งได้มีการจัดหาฟังก์ชันที่ให้บริการด้านการรักษาความปลอดภัยที่เรียกว่า "ไอพีไฟร์วอลล์" และจากที่กล่าวมาว่าไฟร์วอลล์ที่จะพิจารณาจะเป็นไฟร์วอลล์แบบกรองแพ็คเก็ตที่มีกฎและนโยบายเป็นตัวกำหนดการทำงานของไฟร์วอลล์ ซึ่งในส่วนนี้จะมีการใช้ภาษาพอนเดอร์(Ponder)ที่กล่าวมาแล้วในตอนต้นมาใช้เป็นทฤษฎีในการกำหนดรูปแบบนโยบายหรือกฎที่ใช้ ดังนั้นไอพีไฟร์วอลล์จึงเกี่ยวข้องกับหลักการต่างๆที่ได้กล่าวมาในเบื้องต้น

2.1 โปรโตคอลทีซีพี/ไอพี(TCP/IP protocol)

แรกเริ่มโปรโตคอลทีซีพี/ไอพีคือระเบียบวิธีหรือโปรโตคอลในการสื่อสารกันระหว่างเครื่องคอมพิวเตอร์ที่เชื่อมต่อกันในระบบยูนิกซ์ (Unix) สำหรับปัจจุบันนี้โปรโตคอลทีซีพี/ไอพีมีใช้งานในเครื่องคอมพิวเตอร์แทบทุกแบบ ทำให้เครื่องคอมพิวเตอร์แบบใดๆก็ตามที่ทำงานกับซอฟต์แวร์โปรโตคอลทีซีพี/ไอพีก็สามารถเชื่อมเข้าในเครือข่ายโปรโตคอลทีซีพี/ไอพีได้ และแต่ละเครือข่ายก็สามารถเชื่อมโยงกันทำให้กลายเป็นเครือข่ายอินเตอร์เน็ตในที่สุด

2.1.1 โครงสร้างของโปรโตคอลทีซีพี/ไอพี

โปรโตคอลทีซีพี/ไอพีเป็นชุดโปรโตคอลที่ประกอบไปด้วยไอพี (IP: Internet Protocol), ทีซีพี (TCP: Transmission Control Protocol), ยูดีพี (UDP: User Datagram Protocol) ฯลฯ โครงสร้างของโปรโตคอลทีซีพี/ไอพีแสดงอยู่ในภาพที่ 2.1 มีจำนวน 5 ชั้น สอดคล้องกับชั้นต่างๆ ของแบบจำลองอ้างอิงสำหรับการเชื่อมต่อระหว่างระบบเปิด (OSI Reference Model) ดังนี้

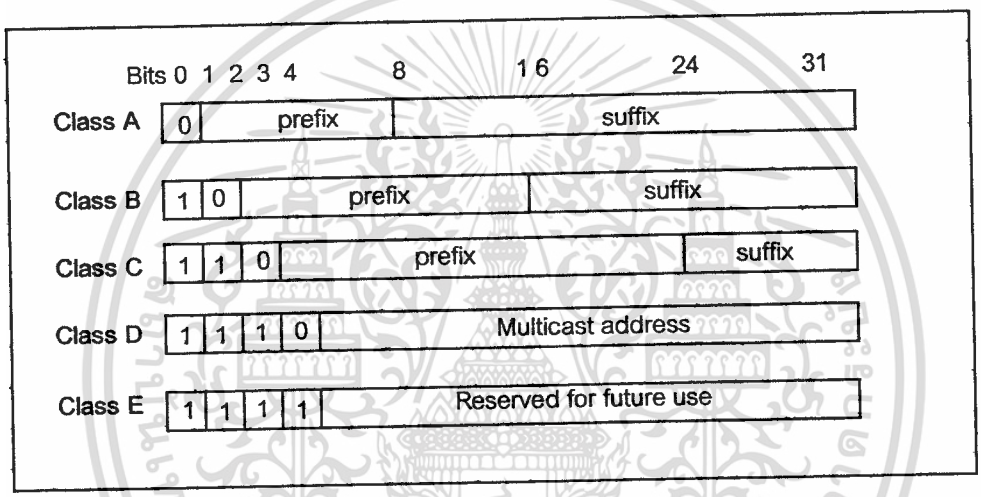


ภาพที่2.1 โครงสร้างโปรโตคอลทีซีพี/ไอพี

- **Layer 1: Physical** ในชั้นที่ 1 จะเกี่ยวกับอุปกรณ์เครือข่ายพื้นฐานอย่างเดียว ซึ่งสอดคล้องกับชั้นที่ 1 ของโมเดลการลำดับชั้นมาตรฐาน
- **Layer 2 :Network Interface** โปรโตคอลขอชั้นที่ 2 จะเกี่ยวข้องกับการจัดข้อมูลลงในเฟรม(Frame) และการส่งเฟรมต่างๆ ข้ามเครือข่ายซึ่งคล้ายกับการทำงานในชั้นที่ 2 ของโมเดลการลำดับชั้นที่เป็นมาตรฐาน
- **Layer 3 :Internet** โปรโตคอลต่างๆในชั้นที่ 3 จะกำหนดรูปแบบของแพ็คเก็ตที่จะส่งข้ามเครือข่ายอินเตอร์เน็ต การทำงานจะคล้ายกับกลไกที่ใช้ในการส่งแพ็คเก็ตจากคอมพิวเตอร์เครื่องหนึ่งไปยังเครื่องปลายทางผ่านเราท์เตอร์(router)
- **Layer 4 : Transport** การทำงานของโปรโตคอลต่างๆในชั้นที่ 4 จะคล้ายกับการทำงานของชั้นที่ 4 ของโมเดลการลำดับชั้นที่เป็นมาตรฐาน คือ ได้มีการกำหนดถึงการรับรองความไว้ใจในการส่งผ่านข้อมูล

- **Layer 5 :Application** ในลำดับชั้นที่ 5 จะสอดคล้องกับชั้นที่ 6 และชั้นที่ 7 ของโมเดลการลำดับชั้นมาตรฐานโดยโปรโตคอลแต่ละตัวจะมีการระบุถึงแอปพลิเคชั่นต่างๆ (Applications) ที่จะใช้ในอินเตอร์เน็ต

โปรโตคอลที่อยู่ในชั้นอินเตอร์เน็ตจะประกอบไปด้วยโปรโตคอลหลายตัวเช่น ไอพี ไอซีเอ็มพี(ICMP: Internet Control Message Protocol) โปรโตคอลเกตเวย์ (Gateway Protocol) ฯลฯ แต่ในเนื้อหาจะกล่าวถึงไอพีเพียงอย่างเดียวเท่านั้น



ภาพที่ 2.2 หมายเลขเครื่องอินเตอร์เน็ตทั้ง 5 ประเภท

2.1.2 ไอพี

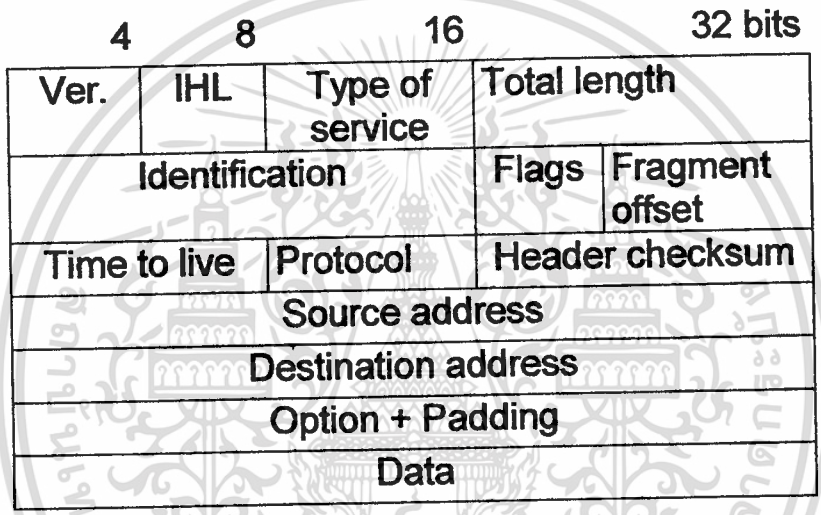
ไอพีเป็นโปรโตคอลระหว่างเครือข่าย (Internetworking Protocol) ที่ถูกพัฒนาโดยกระทรวงกลาโหมของสหรัฐอเมริกา ระบบต่างๆ ได้ถูกพัฒนาให้เป็นส่วนหนึ่งของโครงการดาร์ปาอินเตอร์เน็ตเวิร์คโปรโตคอล (DAPRA internetwork protocol) ซึ่งเป็นระบบที่มีการใช้ทั่วโลก

- **การทำงานของส่วนไอพี**

เนื่องจากไอพีคือขั้นตอนของการส่งข้อมูลระหว่างเครือในข่าย จะมีหมายเลขเครื่องในระบบอินเตอร์เน็ต (Internet Address) ซึ่งเป็นตัวกำหนดว่าจะส่งข้อมูลไปที่ส่วนใดในเครือข่ายลักษณะของหมายเลขเครื่องในระบบอินเตอร์เน็ตแต่ละหมายเลขมีขนาด 32 บิต และได้มีการแบ่งหมายเลขออกเป็น 2 ส่วนคือส่วนที่เป็นหมายเลขเครือข่าย (Prefix) และส่วนที่เป็นหมายเลขประจำเครื่อง (Suffix)

ส่วนของหมายเลขเครือข่ายจะถูกกำหนดโดยหน่วยงาน interNIC เพื่อไม่ให้มีการซ้ำซ้อนกัน ส่วนหมายเลขประจำเครื่องเป็นเลขที่กำหนดโดยผู้บริหารเครือข่ายให้กับเครื่องคอมพิวเตอร์ที่อยู่ภายในเครือข่าย

แต่หมายเลขที่ปรากฏก็จะเป็นอย่างใดอย่างหนึ่งใน 5 ประเภท ซึ่งแต่ละประเภทก็จะต่างกันที่ขนาดของหมายเลขเครื่องและหมายเลขประจำเครื่องนั้นเอง ในภาพที่ 2.2 จะแสดงหมายเลขเครื่องในระบบอินเตอร์เน็ตทั้ง 5 ประเภท และไอพีได้กำหนดไอพีดาต้าแกรม ซึ่งหน่วยพื้นฐานสำหรับการส่งผ่านข้ามไปในโปรโตคอลทีซีพี/ไอพี ดังแสดงในภาพที่ 2.3



ภาพที่ 2.3 รูปแบบของไอพีดาต้าแกรม

โปรโตคอลในชั้นทรานสปอร์ตมีหลายโปรโตคอล โปรโตคอลที่เราสนใจคือ ทีซีพีและยูดีพี โดยที่ทีซีพีเป็นการติดต่อสื่อสารแบบมีการเชื่อมต่อ (Connection-oriented) และยูดีพีเป็นการติดต่อสื่อสารแบบไม่มีการเชื่อมต่อ (Connectionless)

2.1.3 ทีซีพี

โปรโตคอลทีซีพีจะกำหนดช่วงเวลาสำหรับการติดต่อเพื่อยืนยันการส่ง-รับข้อมูลระหว่างคอมพิวเตอร์ทั้ง 2 เครื่อง ทำให้โปรโตคอลทีซีพีเป็นโปรโตคอลที่มีความน่าเชื่อถือ (Reliable) เพราะให้ความแน่นอนว่าแพ็คเก็ตข้อมูลที่ถูกส่งออกไปจากต้นทางจะไปถึงยังปลายทางอย่างเป็นลำดับ และไม่มีความผิดพลาด หรือความสูญหายของข้อมูล ดังนั้นโปรโตคอลทีซีพีจึงเป็นโปรโตคอลสำหรับควบคุมการสื่อสาร กำหนดตำแหน่งต้นทางและปลายทาง และอื่นๆกับข้อมูล

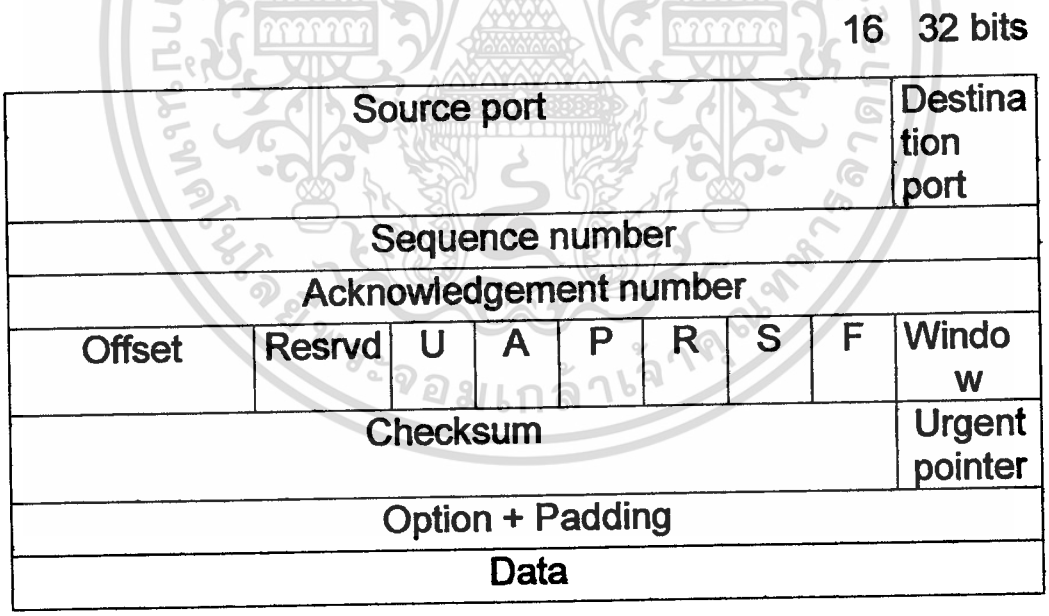
- **การทำงานของทีซีพี**

ขั้นตอนของทีซีพีจะเป็นส่วนการทำงานภายในคอมพิวเตอร์แต่ละเครื่อง มีหน้าที่ทำให้แน่ใจว่าไม่มีความผิดพลาดของข้อมูลที่ได้รับ ลำดับของข้อมูลที่ถูกต้อง ครบถ้วนไม่ซ้ำ

แอปพลิเคชั่นที่ต้องการส่งข้อมูล จะส่งข้อมูลมาให้ทีซีพีเพื่อตัดแบ่งข้อมูลออกเป็นส่วนๆเรียกว่าเป็น "เซ็กเม็นต์" (Segment) เพื่อไม่ให้ข้อมูลมีขนาดยาวเกินไป จากนั้นก็จะส่งข้อมูลแต่ละส่วนให้กับไอพีสร้างชุดข้อมูลที่จะส่งให้กับเครือข่าย

ทีซีพีจะต้องทำกระบวนการรับ-ส่งข้อมูลพร้อมกัน ในขั้นตอนของการรับข้อมูลของทีซีพีจะต้องมีการส่งการตอบรับ (Acknowledge) แจ้งให้ผู้ส่งข้อมูลทราบว่าได้รับข้อมูลที่ถูกต้องมากี่ส่วนแล้ว ถ้าเครื่องที่ส่งข้อมูลยังไม่ได้รับการตอบรับภายในเวลาที่กำหนดทีซีพีก็จะส่งข้อมูลซ้ำออกไปอีก ทำให้มีบางครั้งข้อมูลชุดเดียวกันถูกส่งออกไปมากกว่าหนึ่งครั้ง

เนื่องจากมีการแบ่งข้อมูลออกเป็นส่วนๆ ข้อมูลแต่ละส่วนอาจจะใช้เวลาเดินทางไม่เท่ากัน ข้อมูลที่มาถึงจึงไม่เรียงลำดับ ทีซีพีจะทำการจัดลำดับข้อมูลให้ถูกต้อง ตัดข้อมูลที่ซ้ำซ้อน รวบรวมข้อมูลจนได้ครบทุกส่วนแล้วจึงส่งให้แอปพลิเคชั่นที่จะใช้ ข้อมูลนั้นอีกทีหนึ่ง

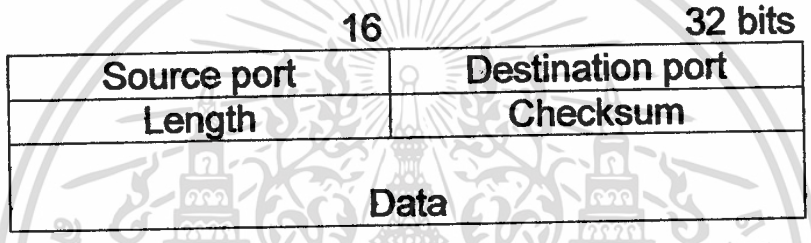


ภาพที่ 2.4 รูปแบบของทีซีพี

2.1.4 ยูดีพี

โปรโตคอลยูดีพีคือโปรโตคอลที่ทำหน้าที่ควบคุมการรับ-ส่งข้อมูลโดยไม่มีการรอคอยการยืนยันการตอบรับจากข้อมูลปลายทาง ทำให้การบริการแบบนี้มีความเชื่อถือน้อยกว่า แต่ทำให้การสื่อสารข้อมูลรวดเร็วยิ่งขึ้นถ้าไม่มีความผิดพลาดเกิดขึ้นในการรับ-ส่งข้อมูล

ในลำดับชั้นตามภาพที่ 2.1 ยูดีพีจะอยู่เหนือไอพีเนื่องจากยูดีพีเป็นการสื่อสารแบบไม่มีการเชื่อมต่อ และยูดีพีจะเพิ่มความสามารถในการหาที่อยู่ของพอร์ต (Port) ให้กับไอพีโดยจะพิจารณาที่ส่วนหัวข้องยูดีพีดังแสดงในภาพที่ 2.5



ภาพที่ 2.5 รูปแบบของฟิลด์ในยูดีพีดาต้าแกรม

2.2 ไฟร์วอลล์

ไฟร์วอลล์ได้ถูกออกแบบมาเพื่อจัดการกับความไม่ต้องการและความไม่มีสิทธิในการจราจรที่จะเข้ามาหรือไปจากเครือข่ายที่ไม่มีการจัดการเรื่องความปลอดภัย เช่น อินเตอร์เน็ต มายังเครือข่ายของคุณ และยังสามารถอนุญาตให้คุณหรือผู้อื่นที่อยู่ในเครือข่ายที่ปลอดภัยสามารถที่จะติดต่อไปยังเครือข่ายภายนอกได้

ไฟร์วอลล์ส่วนใหญ่จะทำหน้าที่เป็น เราท์เตอร์ และจะทำการกรองข้อมูลที่ผ่านเข้ามาหรือส่งออกไปโดยจะอาศัยนโยบายในการรักษาความปลอดภัย หรือ การตัดสินใจของผู้จัดการในการดูแลระบบเครือข่าย

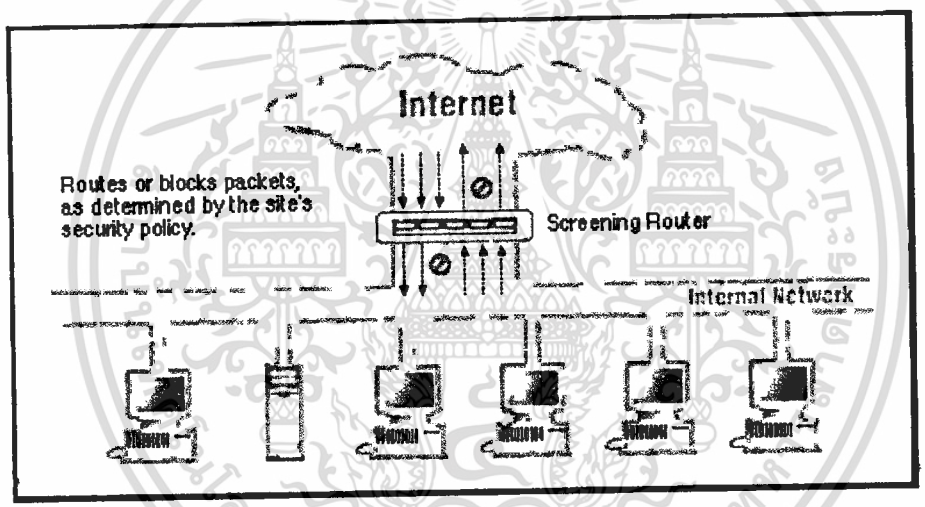
2.2.1 ประเภทของไฟร์วอลล์

ในปัจจุบันมีไฟร์วอลล์ 2 ประเภทที่นิยมใช้กันมากบนอินเตอร์เน็ตคือ การกรองแพ็คเก็ต (Packet Filtering) และ พร็อกซี่บริการ (Proxy Service)

ก. การกรองแพ็คเก็ต (Packet Filtering)

เป็นพฤติกรรมของเครื่องมือที่ใช้ในการควบคุมการไหลของข้อมูลซึ่ง การกรองแพ็คเก็ตจะอนุญาตหรือปฏิเสธแพ็คเก็ตเหล่านั้น ในขณะที่กำลังจัดการการไหลของข้อมูลจากเครือข่ายหนึ่งไปยังอีกเครือข่ายหนึ่ง(ส่วนมากแล้วจะเป็นการส่งข้อมูลระหว่างเครือข่ายอินเตอร์เน็ตกับเครือข่ายภายใน) จะมีการกรองแพ็คเก็ตควบคู่ไปด้วย ผู้ดูแลสามารถที่จะตั้งกฎต่างๆที่กำหนดชนิดของแพ็คเก็ต (เช่น มาจากไอพีแอดเดรสไหน หรือ พอร์ตไหน) ซึ่งสามารถที่จะอนุญาตหรือปฏิเสธแพ็คเก็ตนั้นได้ การกรองแพ็คเก็ตอาจจะพบในเราท์เตอร์ บริดจ์ หรือบนโฮสท์ ซึ่งอาจจะเรียกวิธีนี้ว่า "Screening"

ระบบการกรองแพ็คเก็ตจะทำการส่งแพ็คเก็ต ระหว่างเครือข่ายภายในและเครือข่ายภายนอก ถ้าแพ็คเก็ตนั้นถูกอนุญาตหรือจะถูกบล็อกโดยอาศัยนโยบายขององค์กร ซึ่งภาพที่ 2.6 แสดงชนิดของเราท์เตอร์ที่ทำหน้าที่เป็นไฟร์วอลล์แบบกรองแพ็คเก็ต หรือเรียกว่า "Screening Router"



ภาพที่ 2.6 การใช้ Screening Router เพื่อทำการกรองแพ็คเก็ต

ทุกแพ็คเก็ตจะประกอบด้วยรายละเอียดในส่วนหัว ซึ่งจะมีข้อมูลที่สำคัญดังนี้

- ไอพีแอดเดรสของผู้ส่ง (IP Source Address)
- ไอพีแอดเดรสของผู้รับ (IP Destination Address)
- โปรโตคอล (เช่น ทีซีพี ยูดีพี ไอซีเอ็มพี)
- พอร์ต ทีซีพี หรือ ยูดีพีของผู้ส่ง
- พอร์ต ทีซีพี หรือ ยูดีพีของผู้รับ
- ชนิดข้อมูลของไอซีเอ็มพี (ICMP message type)

รวมไปถึงส่วนต่างๆที่ไม่ได้บอกในส่วนหัวของแพ็คเก็ต

- การเชื่อมต่อของแพ็คเก็ตที่มาถึง
- การเชื่อมต่อของแพ็คเก็ตที่จะถูกส่งออกไป

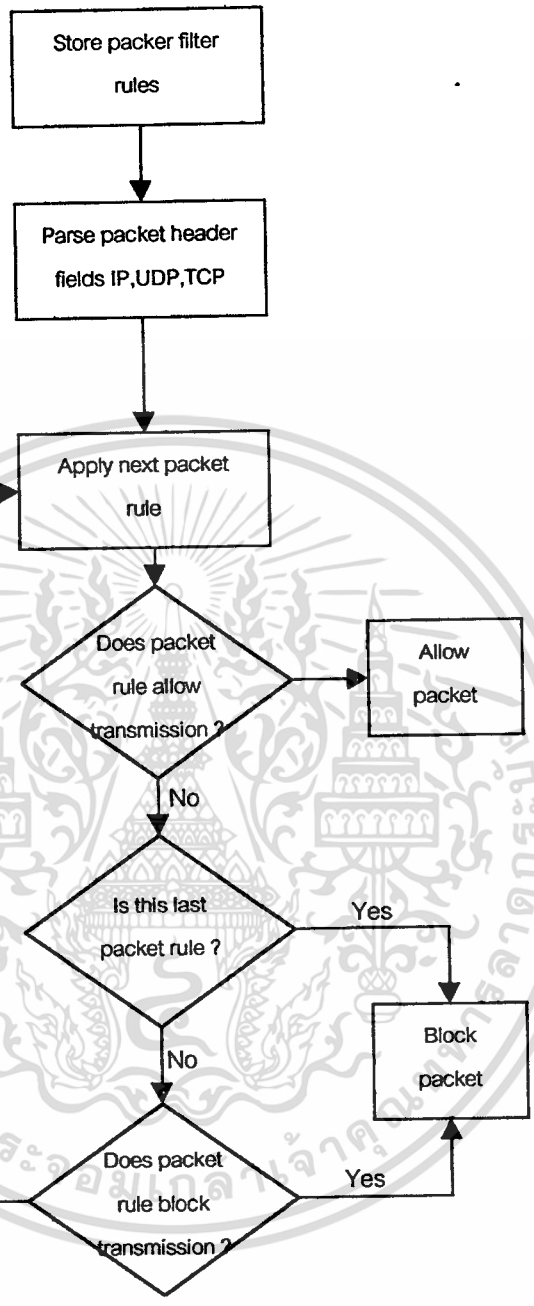
- **ขั้นตอนการทำงานของการกรองแพ็คเก็ตในอุปกรณ์เราท์เตอร์มีดังนี้**

1. เงื่อนไขของการกรองแพ็คเก็ตสำหรับพอร์ตของอุปกรณ์เราท์เตอร์จะถูกเก็บไว้ซึ่งเงื่อนไขการกรองแพ็คเก็ตจะเรียกว่า "กฎการกรองแพ็คเก็ต" (Packet filter rules)
2. เมื่อแพ็คเก็ตเข้ามายังพอร์ต ส่วนหัวของแพ็คเก็ตจะถูกกระจายออก ซึ่งอุปกรณ์เราท์เตอร์ส่วนใหญ่ที่กรองแพ็คเก็ตจะพิจารณาฟิล์ดเฉพาะในส่วนหัวของไอพี ทีซีพี และยูดีพีเท่านั้น
3. กฎการกรองแพ็คเก็ตที่ถูกเก็บไว้ จะมีการระบุถึงลำดับ ทำให้แต่ละกฎตรวจสอบแต่ละแพ็คเก็ตอย่างเป็นลำดับด้วย
4. ถ้ากฎเป็นการบล็อกการส่งผ่านหรือเป็นการบล็อกการรับแพ็คเก็ตเข้ามา แสดงว่าแพ็คเก็ตนั้นไม่ได้รับอนุญาต
5. ถ้ากฎเป็นการอนุญาตให้ส่งผ่านหรืออนุญาตให้รับแพ็คเก็ตเข้ามา แสดงว่าแพ็คเก็ตนั้นได้รับอนุญาตให้กระทำต่อไปได้
6. ถ้าแพ็คเก็ตไม่เข้าคู่กับกฎใดๆ เลย แพ็คเก็ตนั้นจะถูกบล็อก

จากขั้นตอนการทำงานของการกรองแพ็คเก็ตที่ผ่านมาสามารถแสดงได้ด้วยโฟล์วชาร์ทดังภาพที่ 2.7

ช. พร็อกซี่บริการ

พร็อกซี่บริการเป็นแอปพลิเคชั่นที่ทำขึ้นมาเฉพาะหรือเป็นโปรแกรมของเซิร์ฟเวอร์ (Server Program) ที่ทำงานอยู่บนโฮสต์ที่เป็นไฟร์วอลล์ (Firewall Host) พร็อกซี่บริการจะตั้งอยู่ระหว่างผู้ใช้ที่อยู่ในเครือข่ายภายในและการบริการอินเตอร์ที่อยู่ภายนอกและแทนที่ผู้ใช้กับการ

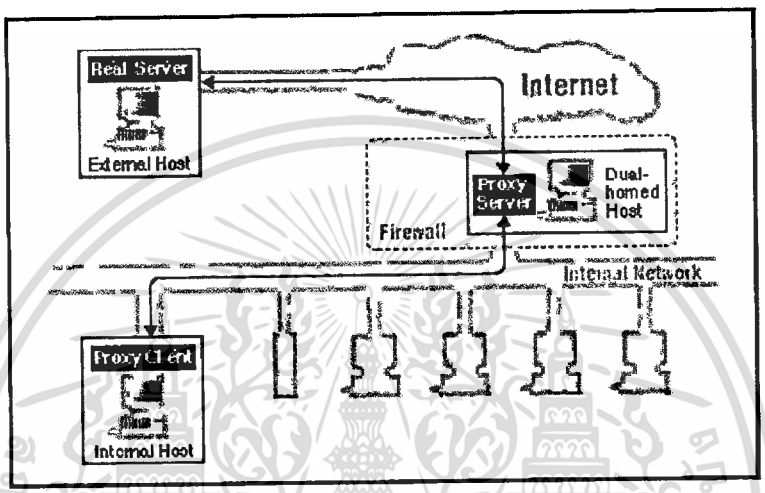


ภาพที่ 2.7 โฟล์วชาร์ทของการทำงานของการกรองแพ็คเก็ต

บริการดังกล่าวจะติดต่อกัน โดยตรงก็จะติดต่อกับพร็อกซี่แทน ดังนั้นพร็อกซี่จะทำหน้าที่ดูแลการติดต่อสื่อสารทั้งหมดระหว่างผู้ใช้กับการบริการอินเตอร์เน็ตเช่น FTP หรือ Telnet

จากภาพที่ 2.8 แสดงการทำงานของพร็อกซี่บริการบนเครื่องโฮสท์ที่เป็นไฟร์วอลล์ ซึ่งจะเรียกเครื่องดังกล่าวว่าเป็น "Dual-homed Host" โดยที่พร็อกซี่เซิร์ฟเวอร์จะเป็นตัวแทนของเครื่อง

เซิร์ฟเวอร์จริง สามารถที่จะติดต่อกับผู้ใช้ผ่านตัวแทนโฮสท์ที่เป็นเครื่องไคล์เอนต์ซึ่งเรียกว่า "พร็อกซี่ไคล์เอนต์" โดยพร็อกซี่ทั้งสองนี้ต่างก็เป็นตัวแทนของเครื่องไคล์เอนต์และเครื่องเซิร์ฟเวอร์ สามารถที่จะติดต่อกันได้ แต่ว่าการที่จะติดต่อออกไปภายนอก พร็อกซี่เซิร์ฟเวอร์จะทำหน้าที่แทนพร็อกซี่ไคล์เอนต์อีกทีหนึ่ง ดังนั้นพร็อกซี่เซิร์ฟเวอร์สามารถที่จะตัดสินใจในการอนุญาตหรือปฏิเสธการร้องขอที่มาจากเครือข่ายภายนอกได้



ภาพที่ 2.8 การใช้พร็อกซี่บริการกับโฮสท์ที่เป็น Dual-homed

2.3 ระบบปฏิบัติการฟรีบีเอสดี

ฟรีบีเอสดีเป็นระบบปฏิบัติการยูนิกส์ที่มีการพัฒนาต่อมาจาก AT&T 's UNIX ซึ่งสามารถที่จะทำงานได้บนแพลท์ฟอร์มดังนี้

- คอมพิวเตอร์ส่วนบุคคลที่อยู่บนพื้นฐานสถาปัตยกรรมการออกแบบของอินเทล i386 รวมไปถึงโพรเซสเซอร์ 386 486 และตระกูลเพนเทียม และยังทำงานได้กับซีพียูค่าย AMD และ Cyrix
- ทำงานได้บนโพรเซสเซอร์ของ Compaq/Digital Alpha
- รวมไปถึงกำลังมีการพัฒนา ฟรีบีเอสดีไปอยู่บนฮาร์ดแวร์อื่นๆ เช่น MIPSR4000 และ SunSparc

ฟรีบีเอสดีถูกพัฒนาจากกลุ่มคนที่ทำงานวิจัยของคณะวิทยาศาสตร์คอมพิวเตอร์ของมหาวิทยาลัยเบอร์กเลย์ที่แคลิฟอร์เนีย (University of California, Berkeley UCB)

ด้วยความสามารถของฟรีบีเอสดีที่ผู้ใช้จะได้รับคือ การเพิ่มการเข้าถึงเคอร์เนลของตัวมันเอง การอนุญาตให้ผู้ใช้เปลี่ยนแปลงพฤติกรรมแรกๆ ได้และอนุญาตให้ดูแลการติดต่อการสื่อสารเองได้ และสิ่งที่สำคัญมากคือ มีการพัฒนาไฟร์วอลล์อยู่ในฟรีบีเอสดีด้วย [3]

ซึ่งไฟร์วอลล์ที่เราเพิ่มเข้าไปในเคอร์เนลนั้นจะเรียกว่า ไอพีไฟร์วอลล์ ซึ่งใช้หลักการทำงานของไฟร์วอลล์แบบกรองแพ็คเก็ต(Packet filtering)

2.4 ไอพีไฟร์วอลล์ (IP Firewall)

เป็นยูทิลิทีที่ใช้ในการควบคุมแพ็คเก็ตและจัดการระเบียบการจราจร ไอพีไฟร์วอลล์เป็นไฟร์วอลล์ที่จะจัดการควบคุมการเข้าถึงในระดับไอพีและระดับทรานสปอร์ตในโปรโตคอลทีซีพี/ไอพี ดังภาพที่ 2.1 และอาจจะมีการอนุญาต(Allow) หรือการปฏิเสธ (Deny) แพ็คเก็ตได้ ขึ้นอยู่กับแอดเดรสต้นทาง แอดเดรสปลายทาง ชนิดของแอปพลิเคชั่นเป็นหลัก และอื่นๆ โดยที่ไฟร์วอลล์แบบกรองแพ็คเก็ตนี้ได้จัดหาความปลอดภัยในระดับง่ายๆ และมีประสิทธิภาพในการทำงาน

- **ความสามารถของไอพีไฟร์วอลล์**

ไอพีไฟร์วอลล์เป็นปริการอย่างหนึ่งที่ซ่อนอยู่ในฟรีบีเอสดีและถูกนำไปรวมอยู่ในเคอร์เนล ทำให้สามารถดูแลและควบคุมการเปลี่ยนแปลงพฤติกรรมโดยรวมและเปลี่ยนแปลงนโยบายทางด้านความปลอดภัยของการติดต่อสื่อสารที่ระดับไอพีได้ตลอดเวลา ซึ่งสามารถใช้ไอพีไฟร์วอลล์บนเครื่องที่ไม่ใช่เราท์เตอร์ได้

ถึงแม้ว่าไอพีไฟร์วอลล์เป็นเวอร์ชันที่ง่ายมากและเป็นพื้นฐานของไฟร์วอลล์แบบกรองแพ็คเก็ตมันสามารถใช้แก้ไขกลุ่มของกฎก่อนที่จะจำกัดการเข้ามาได้ และสามารถเพิ่มกฎเข้ามาได้ แต่ว่ามันไม่สามารถแสดงรายละเอียดของเหตุการณ์ของการล็อคและไม่สามารถวิเคราะห์การติดต่อสื่อสารที่เลยช่วงเวลานั้นมาแล้วได้ เนื่องจากการควบคุมที่นอกเหนือจากไอพีไฟร์วอลล์จะอนุญาตให้เพียงผู้ใช้พิเศษเท่านั้นที่ทำได้

เป็นที่ทราบว่าส่วนหลัก ๆ ของไอพีไฟร์วอลล์จะอยู่ในเคอร์เนล ดังนั้นถ้าผู้ใช้ต้องการเพิ่มตัวเลือก (Option) ในไฟล์โครงร่างของเคอร์เนลของผู้ใช้ หลังจากนั้นจะต้องทำการคอมไพล์เคอร์เนลอีกครั้ง

ตัวอย่างเมื่อมีการขยายกลุ่มของกฎออกไป ทำให้การจำกัดแพ็คเก็ตถูกระบุบมากขึ้น และทำให้การบันทึกแพ็คเก็ตบางส่วนถูกปิดลง เนื่องจากมีการเปลี่ยนแปลงกฎ และถ้าต้องการบันทึกต่อไป จะต้องตั้งค่าตัวนับแพ็คเก็ตที่เกี่ยวข้องด้วยอีกครั้ง โดยใช้คำสั่งของไอพีไฟร์วอลล์ที่เรียกว่า "ifpw clear [rule no.]"

แต่ละแพ็คเก็ตที่ถูกส่งออกไปและรับเข้ามาจะต้องมาจากการผ่านกฎของไอพีไฟร์วอลล์แต่ละแพ็คเก็ตสามารถที่จะกรองได้ จะต้องสอดคล้องกับข้อมูลดังต่อไปนี้

- Receive Interface เป็นอินเตอร์เฟสที่จะรับแพ็คเก็ตเข้ามา

- Transmit Interface เป็นอินเตอร์เฟสที่จะส่งแพ็คเก็ตออกไป
- Incoming เป็นแพ็คเก็ตที่เพิ่งได้รับมา
- Outgoing เป็นแพ็คเก็ตที่ต้องถูกส่งออกไป
- Source IP Address เป็นไอพีแอดเดรสของผู้ส่ง
- Destination IP Address เป็นไอพีแอดเดรสของผู้รับ
- Protocol เป็นไอพีโปรโตคอลที่ประกอบไปด้วยไอพี(IP) ยูดีพี (UDP) ทีซีพี (TCP) และไอซีเอ็มพี (ICMP)
- Source Port เป็นพอร์ตของยูดีพีหรือทีซีพีของผู้ส่ง
- Destination Port เป็นพอร์ตของยูดีพีหรือทีซีพีของผู้รับ
- Connection Setup Flag แพ็คเก็ตนี้เป็นการร้องขอให้มีการจัดตั้ง TCP Connection
- Connection Established Flag แพ็คเก็ตนี้เป็นส่วนของการสร้าง TCP Connection
- All TCP Flags แพ็คเก็ตนี้เป็น TCP Flag ต่างๆเช่นการปิดการติดต่อ(fin) การเปิดการติดต่อ(syn) การตั้งการติดต่อใหม่(rst) ฯลฯ
- Fragment Flag แพ็คเก็ตนี้เป็น Fragment ของไอพีแพ็คเก็ต
- IP Option แพ็คเก็ตนี้เป็น Option ต่าง ๆ ของไอพีเช่น Strict source route(ssrr) หรือ Loose source route (lsrr) เป็นต้น
- ICMP Types แพ็คเก็ตนี้เป็นชนิดต่าง ๆ ของไอซีเอ็มพีเช่น Echo reply(0) หรือ Destination unreachable(3) เป็นต้น

การกรองแพ็คเก็ตเป็นส่วนหนึ่งของไฟร์วอลล์ โดยหลักการนี้ถูกนำมาใช้ในไอพีไฟร์วอลล์ มีความสามารถคือ มีการระบุกฎหรือนโยบายของเครือข่าย โดยที่ในกฎจะมีการระบุถึงฟิลด์ที่สามารถกรองได้ (กรองแล้วมีประโยชน์) ประกอบด้วยแอดเดรสต้นทางและปลายทางของไอพี ชนิดของโปรโตคอล และพอร์ตต้นทางและปลายทางของทีซีพีและยูดีพี และอื่น ๆ

- **โครงร่างของไอพีไฟร์วอลล์**

ไอพีไฟร์วอลล์จะถูกกระทำโดยรายการของกฎ ซึ่งจะตรวจหาในรายการจนกระทั่งแต่ละแพ็คเก็ตเข้าคู่กับกฎ ซึ่งจะขึ้นอยู่กับว่าการกระทำนั้นระบบได้ตั้งไว้ให้ดำเนินการอย่างไร (อนุญาต หรือ ปฏิเสธ) ซึ่งกฎจะมีการเรียงลำดับของกฎ (Line Number) คือมีค่าตั้งแต่ 1 จนถึง 65534 ซึ่งเราสามารถที่จะเข้าไปแก้ไขกฎต่างๆได้ และจะมีการกำหนดกฎเริ่มแรก(Default rule) คือกฎลำดับที่ 65535 ซึ่งเราไม่สามารถแก้ไขได้โดยตรง ซึ่งโดยปกติจะเป็นการปฏิเสธแพ็คเก็ตทั้งหมด นอกจาก

จะไปกำหนดตัวเลือก(option) ของเคอร์เนลด้วย IPFIREWALL_DEFAULT_TO_ACCEPT เพื่อจะเป็นการอนุญาตทุกแพ็คเก็ต ซึ่งกฎทั้งหมดจะมีความสัมพันธ์กับตัวนับ(Counter) คือ ตัวนับแพ็คเก็ต (Packet counter) ตัวนับไบต์ (Byte counter) โดยที่ตัวนับเหล่านี้จะถูกเปลี่ยนแปลงเมื่อแพ็คเก็ตที่เข้ามาเข้าคู่กับกฎ

กฎขั้นพื้นฐานของไอพีไฟร์วอลล์มีดังต่อไปนี้

- Addition เป็นการเพิ่มกฎเข้าไป
- Delete เป็นการลบกฎออก
- Flush เป็นการลบกฎออกทั้งหมด (คล้ายกับการ Delete)
- Show/List เป็นการแสดงรายละเอียดของกฎและตัวนับต่างๆ
- Zero/Resetlog เป็นการปรับค่าตัวนับต่างๆ

ซึ่งมีรูปแบบทั่วไปของการใช้คำสั่งของไอพีไฟร์วอลล์(ipfw)ดังนี้

```
ipfw [prob match_probability] action [log[logamount number] proto from scr to dst
[interface_spec] [options]
```

โดยที่

prob คือ ความน่าจะเป็นที่จะสุ่มแพ็คเก็ตที่เข้าคู่กับกฎเพื่อให้เป็นไปตาม action ที่กำหนด

action คือ กิจกรรมที่กำหนดว่าให้กระทำกับแพ็คเก็ตนั้นๆ เช่น อนุญาต หรือ ปฏิเสธ

log คือ ให้แสดงค่าที่อยู่ใน log ที่กำหนดไว้ทางหน้าจอ

proto คือ โปรโตคอลที่ต้องการจะทำการกรอง

scr คือ ไอพีแอดเดรสต้นทาง

dsr คือ ไอพีแอดเดรสปลายทาง

interface_spec คือ เป็นการระบุทิศทางการเข้ามาหรือออกไปของแพ็คเก็ต

options คือ ตัวเลือกที่สามารถเพิ่มเข้าไปได้

2.5 การสร้างโครงร่างของไอพีไฟร์วอลล์(The Configuration IP Firewall)

ไอพีไฟร์วอลล์โค้ดจะทำงานโดยที่แต่ละแพ็คเก็ตจะต้องผ่านรายการของกฎจนกระทั่งพบกฎที่เข้าคู่ โดยที่กฎจะมีการเรียงลำดับด้วยเลขบันทัด(Line-number) คือตั้งแต่ตัวเลข 1 จนถึง 65534 และกฎทั้งหมดจะมีความสัมพันธ์กับตัวนับ 2 ตัวคือ ตัวนับแพ็คเก็ต(Packet counter) กับตัวนับไบต์

(Byte counter) โดยที่ตัวนับทั้ง 2 ตัวนี้จะถูกเปลี่ยนแปลงเมื่อแพ็คเก็ตนั้นเข้าคู่กับกฎ และเนื่องจากว่าไอพีไฟร์วอลล์เป็นไฟร์วอลล์แบบกรองแพ็คเก็ตที่ได้มีการสร้างกฎต่างๆ เอาไว้ในการจับคู่กับแพ็คเก็ตที่เข้ามายังเครือข่าย เพื่อตัดสินใจว่าแพ็คเก็ตใดสามารถผ่านเข้ามาได้หรือผ่านเข้ามาไม่ได้ ดังนั้นการสร้างกฎจึงมีความสำคัญอย่างยิ่ง สิ่งที่จะต้องพิจารณาต่อไปจะเกี่ยวข้องกับการสร้างกฎ ซึ่งมี 4 อย่างคือ

- Addition/Deletion การเพิ่ม/ลบออกจะใช้ในการสร้างกฎเพื่อควบคุมแพ็คเก็ตที่เข้ามาว่าจะยอมรับหรือไม่ยอมรับแพ็คเก็ตนั้นไว้
- Listing การทำบัญชีของกฎจะใช้ในการพิจารณาถึงสารบัญของกลุ่มของกฎ(หรือที่เรียกว่า Chain) และตัวนับแพ็คเก็ต (Packet Counter)
- Flushing การเคลื่อนย้ายกฎที่เข้ามาทั้งหมดออกจากกลุ่ม
- Clearing การลบค่าที่ตัวนับแพ็คเก็ตให้เป็นศูนย์

หมายเหตุ: การทำงานทั้ง4อย่างที่กล่าวมานี้จะอยู่ในโปรแกรมไอพีไฟร์วอลล์ที่มากับระบบฟรีบีเอสดี

2.5.1 รูปแบบคำสั่งของ Addition/Deletion

```
ipfw [-N] command [index] action [log] protocol address [options]
```

โดยที่

- -N ใช้แยกส่วนของแอดเดรสและชื่อบริการต่างๆใน Output
- command เป็นคำสั่งซึ่งอยู่ในรูปแบบที่สั้นและไม่ซ้ำ ได้แก่
  - add เป็นการเพิ่มกฎในกลุ่มของกฎ
  - delete เป็นการลบกฎในกลุ่มของกฎ
- index เป็นการระบุตำแหน่งของกฎในกลุ่มของกฎ
- action ประกอบด้วย
  - allow เป็นการอนุญาตให้แพ็คเก็ตผ่านไปโดยปกติ
  - deny เป็นการดร็อปแพ็คเก็ต ดั้งนั้นแพ็คเก็ตจะไม่สามารถส่งไปยังต้นทางได้
  - reject เป็นการลบแพ็คเก็ตที่เข้าคู่กับกฎและจะส่งข้อความไปที่โฮสต์ไอซีเอ็มพีที่ไม่สามารถขยายต่อไปได้(Unreachable)
  - unreach code เป็นการลบแพ็คเก็ตที่เข้าคู่กับกฎและจะส่งข้อความไปที่โฮสต์ไอซีเอ็มพี ที่ไม่สามารถขยายต่อไปได้ด้วยโค้ดซึ่งมีค่าตั้งแต่ 0-255

- ◆ reset จะใช้กับแพ็คเก็ตที่เป็นทีซีพีเท่านั้น โดยจะลบแพ็คเก็ตที่เข้าคู่กับกฎและจะส่งข้อความที่เป็นทีซีพีรีเซต (TCP reset Message)
- ◆ count เป็นการนับค่าตัวนับแพ็คเก็ต
- ◆ divert port จะทำการเบี่ยง(Divert) แพ็คเก็ตที่เข้าคู่กับกฎไปยุงพอร์ตที่ระบุ
- ◆ tee port จะส่งการคัดลอกแพ็คเก็ตที่เข้าคู่กับกฎไปยังพอร์ตที่ระบุ

- log ทำให้เกิดการเข้าคู่กฎเพื่อไปแสดงผลที่คอนโซลระบบ (System console) ถ้าเคอร์เนลถูกคอมไพล์ด้วย options IPFIREWALL_VERBOSE
- protocol โปรโตคอลสามารถระบุได้ดังนี้
  - ◆ all เป็นการจับคู่กับทุกไอพีแพ็คเก็ต
  - ◆ icmp เป็นการจับคู่กับไอซีเอ็มพีแพ็คเก็ต
  - ◆ tcp เป็นการจับคู่กับทีซีพีแพ็คเก็ต
  - ◆ udp เป็นการจับคู่กับยูดีพีแพ็คเก็ต
- address แอดเดรสจะมีการระบุดังนี้

from address/mark[port] to address/mark [port] [via interface]

- สามารถระบุพอร์ตที่เชื่อมกับโปรโตคอลที่สนับสนุนพอร์ตเช่น ทีซีพี ยูดีพี
- "via" เป็นสิ่งที่เลือกได้และอาจจะมีการระบุไอพีแอดเดรสหรือชื่อโดเมนของอินเตอร์เฟสของไอพีอินเตอร์เฟสภายในหรือชื่ออินเตอร์เฟส เช่น ed0 เพื่อจับคู่เฉพาะแพ็คเก็ตที่ผ่านเข้ามาในอินเตอร์เฟสนี้
- รูปแบบที่ใช้ในการระบุ address/mark มีดังนี้
  1) address ไอพีแอดเดรส
  2) address/mark-bits ไอพีแอดเดรสที่มีสับเน็ตมาร์ก(Sub-netmark) แบบย่อ
  3) address:mark-pattern ไอพีแอดเดรสที่มีสับเน็ตมาร์ก(Sub-netmark) แบบยาว
- ตัวเลขของพอร์ตที่ถูกบล็อกสามารถระบุได้ดังนี้
  1) port[,port[,port[…]]] เป็นการระบุถึงพอร์ตเพียงพอร์ตเดียวหรือระบุเป็นรายการของพอร์ต
  2) port-port เป็นการระบุช่วงของพอร์ต
- options ตัวเลือกที่สามารถใส่เข้าไปได้มีดังนี้
  - ◆ frag จะจับคู่ถ้าแพ็คเก็ตไม่ได้อยู่ใน fragment แรกของดาต้าแกรม

- in จะจับคู่ถ้าแพ็คเก็ตนั้นเป็นแพ็คเก็ตที่เข้ามา
- out จะจับคู่ถ้าแพ็คเก็ตนั้นเป็นแพ็คเก็ตที่ออกไป
- ipoption spec จะจับคู่ถ้าส่วนหัวของไอพีมี comma separated list ของตัวเลือกที่ถูกระบุใน spec เช่น ssrr (strict source route)
- established จะจับคู่ถ้าสแพ็คเก็ตนั้นเป็นส่วนของการจัดสร้างการเชื่อมต่อของทีซีพี
- setup จะจับคู่ถ้าแพ็คเก็ตนั้นเป็นแพ็คเก็ตที่พยายามจะจัดสร้างการเชื่อมต่อของทีซีพี
- tcpflags flags จะจับคู่ถ้าส่วนหัวของทีซีพีมี comma separate list ของ flags เช่น fin syn rst psh ack และ urg
- icmptypes types จะจับคู่ถ้าชนิดของไอซีเอ็มพีปรากฏอยู่ในรายการของ types

2.5.2 รูปแบบคำสั่งของ Listing

```
ipfw [-a] [-t] [-N]
```

ในการทำรายการของกฎประกอบด้วย 3 ส่วนที่ใช้ในคำสั่งนี้

-a ขณะที่ทำการแสดงรายการของกฎจะแสดงค่าของตัวนับ

-t แสดงการจับคู่กฎกับแพ็คเก็ตในครั้งสุดท้ายในแต่ละกฎ

-N การพยายามแยกส่วนของแอดเดรสและชื่อบริการต่างๆ

2.5.3 รูปแบบคำสั่งของ Flushing

```
ipfw flush
```

สิ่งต้องระวังเมื่อมีการใช้คำสั่งนี้คือ ค่าเริ่มต้นของนโยบายจะเป็นการปฏิเสธการบริการทุกอย่างในเครือข่ายจนกว่าจะมีการเพิ่มกฎเข้าไป

2.5.4 รูปแบบคำสั่งของ Clearing

```
ipfw zero [index]
```

เมื่อการใช้คำสั่งนี้ไม่มีส่วนของ index จะทำให้ตัวนับแพ็คเก็ตทั้งหมดถูกลบค่าออกไปด้วย แต่ถ้ามีการระบุค่าใน index การลบค่าตัวนับแพ็คเก็ตจะกระทำเฉพาะกฎนั้น ๆ

2.6 ตัวอย่างการใช้คำสั่งของไอพีไฟร์วอลล์ในฟรีบีเอสดี

- การปฏิเสธแพ็คเก็ตทั้งหมดจากโฮสต์ที่ชื่อว่า evil.crackers.org เพื่อเทลเน็ตมาที่พอร์ตของโฮสต์ nice.people.org

  # ipfw add deny tcp from evil.crackers.org to nice.people.org 23

- การปฏิเสธและล็อกทุกการจราจรของทีซีพีจากเครือข่าย crackers.org (Class C) เพื่อที่จะเข้ามายังเครื่อง nice.people.org ในทุกพอร์ต

  # ipfw add deny log tp from evil.crackers.org/24 to nice.people.org

- การดูเรคคอร์ดของรายการของกฎ

  # ipfw –a list หรือ #ipfw –al

- การดูกฎที่เข้าคู่กับแพ็คเก็ตได้ในครั้งสุดท้าย

  # ipfw –at l

2.7 ภาษาพอนเดอร์

องค์กรขนาดใหญ่ที่จะต้องเกี่ยวข้องกับข้อมูลสารสนเทศจำเป็นจะต้องมีการเชื่อมต่อกับเครือข่ายขององค์กรอื่นๆและบริการอินเตอร์เน็ต สิ่งเหล่านี้มีความท้าทายเป็นอย่างมากที่จะให้ผู้ดูแลระบบมาจัดการรักษาความปลอดภัยกับองค์กร ภาษาพอนเดอร์เป็นเครื่องมือที่จะใช้ช่วยในการกำหนดนโยบายขององค์กร ซึ่งภาษาพอนเดอร์ได้มีการสนับสนุนความต้องการของภาษาที่ใช้กำหนดนโยบายดังนี้

- สนับสนุนสำหรับนโยบายที่ใช้รักษาความปลอดภัยที่ใช้ในรายการควบคุม(Control list)
- สนับสนุนเทคนิคแบบโครงสร้างที่ใช้ในการกำหนดนโยบายต่างๆที่มีขนาดใหญ่
- สามารถนำนโยบายมาวิเคราะห์หาความขัดแย้งของนโยบายและวิเคราะห์นโยบายที่ไม่สมบรูณ์ได้
- ส่วนที่เพิ่มเติมคือสามารถรองรับงานที่เป็นออปเจ็คโอเรียนเท็ด(Object-Oriented)ในอนาคตได้
- ภาษานี้ผู้ใช้จะต้องสามารถเข้าใจได้ง่ายและง่ายต่อการใช้

2.8 โครงสร้างของภาษาพอนเดอร์

ภาษาพอนเดอร์เป็นภาษาที่สนับสนุนการควบคุมการเข้าถึงโดยมีการจัดเตรียมรูปแบบของนโยบายดังนี้

- นโยบายของสิทธิ (authorisation policy)
- นโยบายของตัวแทน (delegation policy)
- นโยบายการกรองข้อมูลสารสนเทศ (information filtering)
- นโยบายการละเว้น (refrain policy)

ซึ่งทุกๆนโยบายจะเกี่ยวข้องกับออปเจ็คต่างๆดังนี้

- subject หมายถึง หลักการต่างๆ หรือ องค์ประกอบต่างๆที่เป็นตัวจัดการที่ทำงานโดยอัตโนมัติ(automated manager components)
- target หมายถึง ทรัพยากรต่างๆหรือ ตัวที่คอยให้บริการ(Service provider)
- Domain หมายถึง กลุ่มของออปเจ็คไม่ว่าจะเป็น subject หรือ target

2.8.1 รูปแบบของนโยบายของสิทธิ

```
inst (auth+|auth-) policyName "{"
    subject [<type>]    domain-Scope-Expression;
    target  [<type>]    domain-Scope-Expression;
    action              action-list;
    [when               domain-Scope-Expression;]
"}"
```

นโยบายของสิทธิคือการกระทำใดก็ตามที่สมาชิกในโดเมนของซับเจ็ค (Subject) สามารถที่จะกระทำกลับกลุ่มของออปเจ็ค (Objects) ในโดเมนที่กำหนด โดยที่นโยบายของสิทธิที่เป็นบวกคือการกระทำใดที่ซับเจ็คต่างๆถูกอนุญาตให้กระทำต่อกลุ่มออปเจ็คที่เป็นเป้าหมายได้ ส่วนนโยบายของสิทธิที่เป็นลบคือการกระทำใดที่ซับเจ็คต่างๆถูกห้ามไม่ให้กระทำต่อกลุ่มออปเจ็ค

ตัวอย่าง

```
inst auth+ switchPolicyOps {
subject         /NetworkAdmin;
action          load(), remove(), enable(), disable();
```

target /Nregion/switches;

}

คือ สมาชิกของ NetworkAdmin Domain มีสิทธิที่จะโหลด, ย้าย, สามรถกระทำ, ไม่สามารถกระทำต่อออปเจ็คใน Nregion/switches

2.8.2 รูปแบบของนโยบายการกรองข้อมูลสารสนเทศ

ActionName { filter}
Filter = [**it** *condition*] "{"
{ (**in** *parameterName* = expression; |
out *parameterName* = expression; |
result = *expression;*
)
}
"}"

นโยบายการกรองจำเป็นที่จะใช้ในการแปลงค่าพารามเตอร์อินพุท/เอาท์พุทในส่วนของแอ็คชั่น ในนโยบายของสิทธิแบบบวกสามารถที่จะรวบเอานโยบายการกรองข้อมูลเท่านั้น เพื่อเข้ามาแปลงค่าพารามิเตอร์อินพุท/เอาท์พุทที่เกี่ยวข้องกับการกระทำที่ขึ้นอยู่กับแอทริบิวต์ของซับเจ็คและออปเจ็คเช่น พารามิเตอร์ที่เกี่ยวกับเวลา

ในการกรองแต่ละครั้งถ้าเงื่อมไขถูกต้องก็จะเข้ามาในส่วนที่ทำการแปลงค่าพารามิเตอร์ซึ่งจะมีคำสำคัญ in/out ที่ใช้เป็นตัวกำหนดพารามิเตอร์ อินพุท/เอาท์พุท ส่วน result จะใช้ในการคืนค่ากลับไปให้การกระทำนั้นๆ

ตัวอย่าง

inst auth+ *filter1* {
subject /Agroup + /Bgroup;
target USAStaff – Nygroup;
action VideoConf(BW, Priority)

{ **in** BW = 2; **in** Priority =3} //default filter

if((time.after("1900")) {**in** BW=3; **in** Priority = 1;}

คือสมาชิกของ Agroup รวบกับ Bgroup สามารถที่จะกำหนดค่าของวิดีโอคอนเฟอร์เรนซ์ไปที่ USAStaff ยกเว้น New York Group ถ้าเวลาหลังจาก 7:00pm สามารถกำหนดค่าของวิดีโอคอนเฟอร์เรนซ์โดยให้มี แบนวิดท์เท่ากับ 3 Mb/s, ความสำคัญเท่ากับ 1แต่ถ้าไม่ใช้ค่าแบนวิดท์เท่ากับ 2 Mb/s, ความสำคัญเท่ากับ 3

2.8.3 รูปแบบของนโยบายของตัวแทน

Inst oblig *policyName* "{"	
On	*event-specification;*
Subject	*domain-Scope-Expression;*
[target	*domain-Scope-Expression;*
Do	*obligation-action-list;*
[catch	*exception-specification;*]
[when	*constrain-Expression;*]
"}"	

การแต่งตั้งตัวแทนจะใช้ในระบบการควบคุมการเข้าถึงเพื่อให้สอดคล้องสำหรับเคลื่อนย้ายชั่วคราวของการเข้าถึงที่ถูกต้อง อย่างไรก็ตามการที่จะมีการตั้งตัวแทนเข้ามาสำหรับผู้ใช้นั้นจะต้องอยู่ภายใต้กฎหรือนโยบายในการรักษาความปลอดภัย นโยบายของตัวแทนจะอนุญาตให้ซับเจ็คสามารถที่จะมอบสิทธิ์ซึ่งซับเจ็คเหล่านั้นควบคุมหรือกระทำต่อนโยบายของสิทธิ์ที่มีเรียบร้อยแล้วไปยังผู้ที่มีสิทธิ์นี้เพื่อที่สามารถจะกระทำกิจกรรมในนามตัวแทน

นโยบายของตัวแทนจะมีความสัมพันธ์กับนโยบายของสิทธิ์ซึ่งจะเป็นการกำหนดความสามารถในการเข้าถึงที่ถูกต้องซึ่งจะถูกมอบให้กับตัวแทนแต่ไม่สามารถใช้ได้กับนโยบายของสิทธิ์ที่เป็นลบ

ตัวอย่าง

Inst deleg+ (switchPolicyOps) elegSwitchOps {

guarantee /DomainAdmin;

Target /Nregion/switches/typeA;

action enable(), disable();

valid time.duration(24);

}

จากตัวอย่างข้างบน นโยบายของตัวแทนจะได้รับนโยบายของสิทธิที่เป็นบวกคือ switchPolicyOps ซึ่งมาจากตัวอย่างก่อนหน้าซึ่งจะรับมาในลักษณะพารามิเตอร์ โดยอธิบายว่าซับเจ็คของนโยบายของสิทธิ (NetworkAdmin) สามารถมอบอำนาจให้สามารถกระทำได้ (enable) และไม่สามารถกระทำได้ (disable) ที่โดเมนของ /Nregion/switches/typeA ไปยังผู้ที่รับอำนาจในโดเมน /DomainAdmin

2.8.4 นโยบายการละเว้น

นโยบายการละเว้นจะเป็นนโยบายในการกำหนดการกระทำซึ่งซับเจ็คจะต้องละเว้นการกระทำหรือต้องไม่กระทำต่อกลุ่มเป้าหมาย ถึงแม้ว่านโยบายต่างๆ เหล่านั้นถูกกำหนดให้สามารถกระทำได้ นโยบายการละเว้นมีรูปแบบของภาษาที่คล้ายกับนโยบายของสิทธิแบบลบแต่จะเข้าไปมีผลโดยตรงกับซับเจ็คมากกว่าเป็นตัวควบคุมการเข้าถึงในส่วนของเป้าหมาย

ตัวอย่าง

Inst refrain *testingRes* {

Subject s=/test-engineer;

action discloseTestResults();

Target /analysis + /developer;

when s.test_sequence = “in-progress”

}

จากตัวอย่างข้างบนนโยบายการละเว้นจะกำหนดให้ test engineer ต้องไม่เปิดเผยผลที่ได้จากการทดสอบไปยังนักวิเคราะห์และนักพัฒนาเมื่อกระบวนการการทดสอบกำลังดำเนิน สังเกตว่าการกำหนดเงื่อนไขจะไปกระทำต่อซับเจ็คโดยตรง

จากที่ยกตัวอย่างของนโยบายต่างๆ นั้นเราจะเห็นว่านโยบายเหล่านั้นมีความสัมพันธ์กับคำสั่งของไอพีไฟร์วอลล์ เราจึงได้เลือกเอามาเป็นรูปแบบของภาษาที่จะใช้ในการพัฒนาโครงงานนี้ แต่ยังมีส่วนที่ต้องออกแบบเพิ่มเติมให้เหมาะสมกับฟังก์ชันการทำงานของเราคือ ในการพิจารณาความขัด

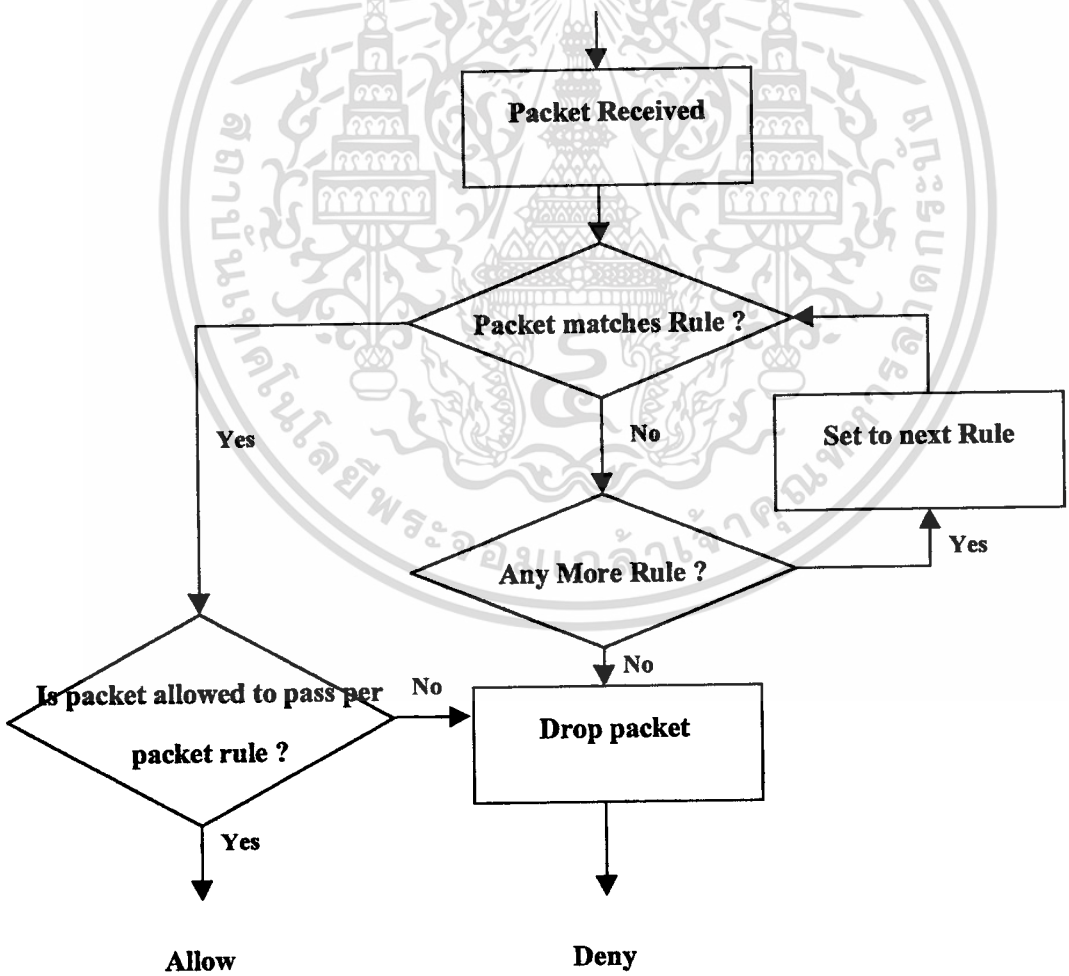
แย้งที่เกิดจากนโยบายหรือกฎเพราะรูปแบบของภาษาพอนเดอร์มีความสามารถเพียงแค่การตรวจสอบเมื่อเกิดความขัดแย้งแต่ไม่สามารถแก้ไขได้เมื่อเกิดปัญหา ดังนั้นจึงได้ออกแบบโดยจะให้ผู้ใช้สามารถใส่ลำดับความสำคัญของนโยบาย (Priority) คือจะเพิ่มฟิลด์ Priority ให้ใส่หมายเลขตั้งแต่ 0 เป็นต้นไปเพื่อไปจัดลำดับของกฎหรือนโยบาย (การให้โปรแกรมเป็นตัวกำหนดหมายเลขของกฎ Rule number โดยอัตโนมัติ)

บทที่ 3

การออกแบบระบบงาน

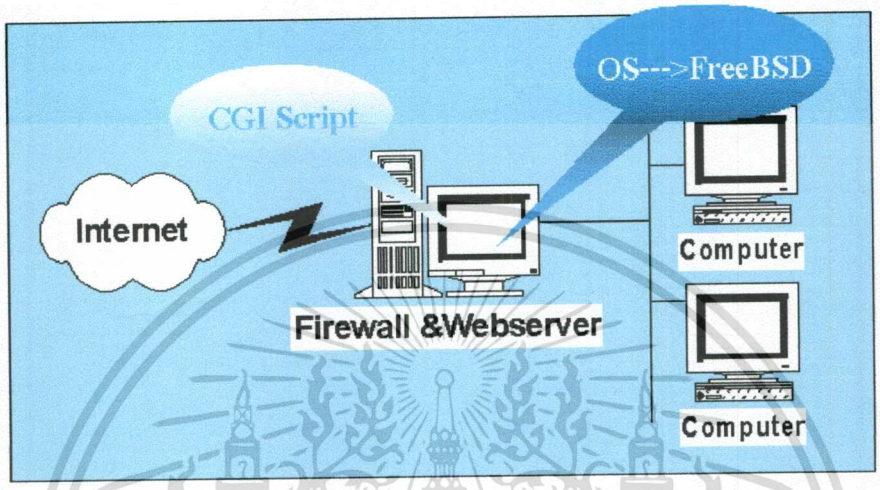
บทนี้จะกล่าวถึงขั้นตอนทำงานของระบบและการออกแบบการทำงานของโปรแกรมจัดการไอพีไฟร์วอลล์บนระบบปฏิบัติการฟรีบีเอสดี โดยจะแสดงในรูปของผังการไหลของข้อมูล (Data Flow Diagram) ดังต่อไปนี้

3.1 การทำงานของไอพีไฟร์วอลล์ในโครงงาน



ภาพที่ 3.1 โฟลชาร์ทการกรองแพ็คเก็ตในระบบของโครงงาน

การทำงานไอพีไฟร์วอลล์อยู่บนพื้นฐานของไฟร์วอลล์แบบกรองแพ็คเก็ต และการทำงานของโมดูลการกรองแพ็คเก็ตสามารถแสดงได้ดังภาพที่ 3.1 สำหรับภาพรวมการทำงานของโปรแกรมว่าจะนำไปใช้ ณ จุดใดสามารถแสดงได้ดังภาพที่ 3.2

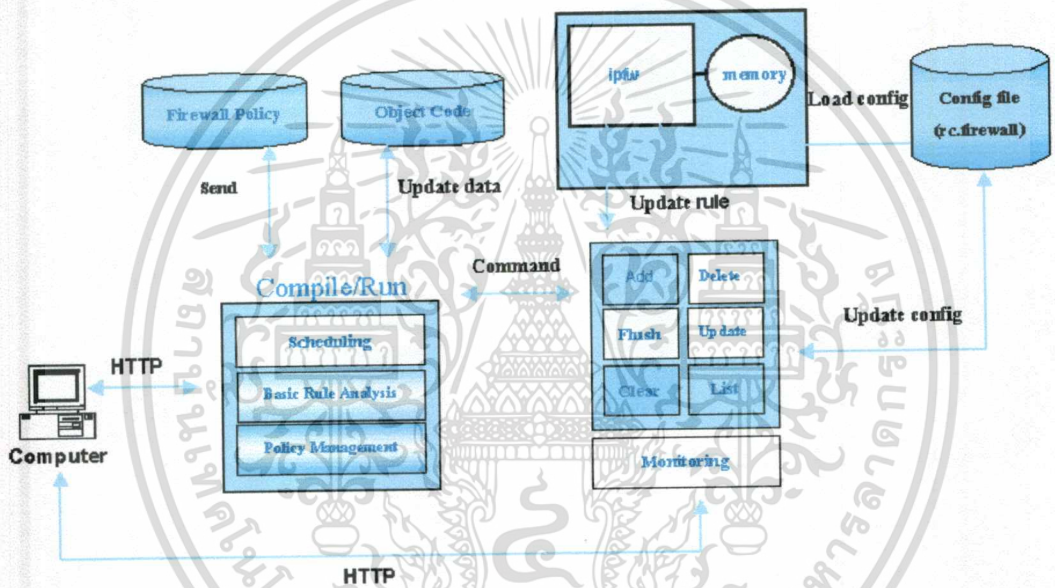


ภาพที่3.2 การกรองแพ็คเก็ตในระบบของโครงงาน

จากภาพที่ 3.2 จะเห็นว่าโปรแกรมที่จัดการไอพีไฟร์วอลล์ในจะติดต่อกับผู้ใช้ผ่านเว็บบราวเซอร์ในสภาวะแวดล้อมที่เป็นเอ็กซ์วินโดร์ และภายในโปรแกรมจะประกอบด้วยฟังก์ชัน 2 ฟังก์ชันหลักโดยที่ฟังก์ชันต่างๆเหล่านั้น ผู้ใช้จะสามารถกระทำได้โดยการป้อนคำสั่งที่อยู่ในรูปแบบภาษาพอนเดอร์ที่ได้ออกแบบไว้ก่อนหน้านี้ ซึ่งกฎต่างๆจะถูกเก็บอยู่ในฐานข้อมูลของกฎ ทำให้ทุกครั้งที่ทำการปรับปรุงกฎให้เป็นปัจจุบันเราก็จะต้องปรับปรุงฐานข้อมูลกฎนี้ด้วย จากรูปที่ 3.3 เราจะเห็นว่าในโปรแกรมของเราจะติดต่อกับฐานข้อมูล ถึง 3 ฐานข้อมูลอันได้แก่ ฐานข้อมูลที่เป็นกฎในรูปแบบของภาษาพอนเดอร์ ฐานข้อมูลที่เก็บกฎที่อยู่ในรูปแบบของไอพีไฟร์วอลล์ และฐานข้อมูลที่กำหนดกฎสำหรับติดตั้งระบบตอนเริ่มทำงานครั้งแรก (Configuration file) สำหรับการปรับรายการของกฎให้เป็นปัจจุบันก็จะมีการปรับปรุงทุกครั้งที่มีการเปลี่ยนแปลงกฎ ได้แก่ การเพิ่มและการลบกฎ เป็นต้น

ดังนั้นเมื่อมีแพ็คเก็ตเข้ามาในโมดูลของการกรองแพ็คเก็ต แพ็คเก็ตเหล่านั้นก็จะถูกตรวจสอบด้วยกฎที่มีทั้งหมด และถ้าแพ็คเก็ตเหล่านั้นเข้าคู่กับกฎ โดยกฎจะแบ่งออกเป็นการอนุญาตกับการปฏิเสธ ถ้าแพ็คเก็ตนั้นเข้าคู่กับกฎการปฏิเสธแพ็คเก็ตนั้นก็จะถูกลบออกไป ถ้าแพ็คเก็ตนั้นเข้าคู่กับกฎการอนุญาตแพ็ตเก็ตนั้นก็จะผ่านเข้ามาได้ การทำงานต่างๆเหล่านี้จะเป็นไปตามโฟลว์ชาร์ทในภาพที่ 3.1

ฟังก์ชันที่อาจมีเพิ่มเติมเช่น การตั้งเวลาในการกรองแพ็คเก็ตในช่วงเวลาที่สามารถกำหนดเองได้ โดยจัดทำเป็นรูปแบบหนึ่งของภาษา อีกทั้งยังมีการวิเคราะห์และจัดการคำสั่งหรือกฎเมื่อเกิดปัญหาเช่นการซ้ำกันของกฎหรือการขัดแย้งของกฎโดยอาศัยข้อดีและวิธีการแก้ปัญหาต่างๆเหล่านี้จากรูปแบบภาษาพอนเดอร์ มีการเก็บข้อมูลเพื่อมาทำวิเคราะห์ทางสถิติของแพ็คเก็ตที่มาเข้าคู่กับกฎและนำเสนอภาพเว็ปเพจ ซึ่งโครงสร้างการทำงานของระบบจะสร้างขึ้นซึ่งแสดงในภาพที่ 3.3

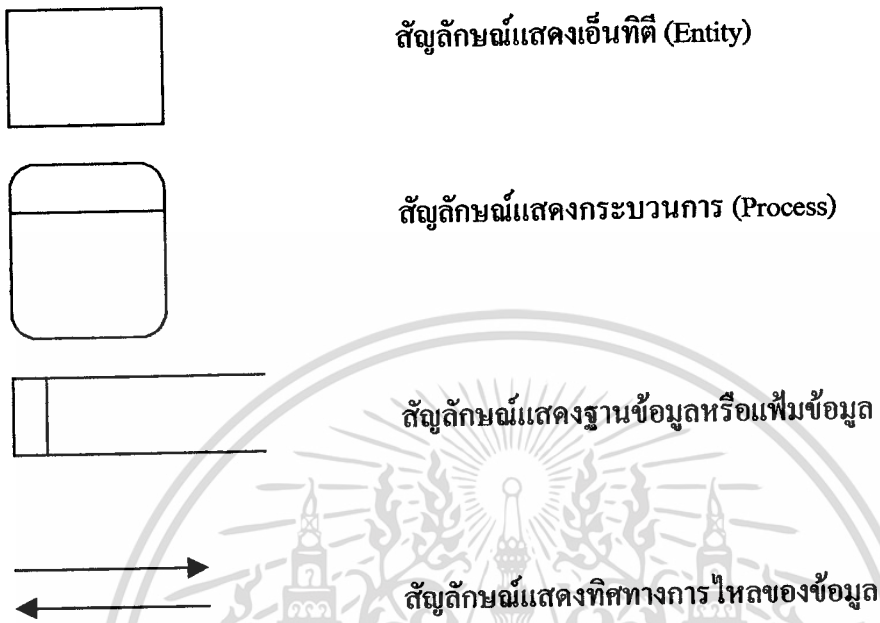


ภาพที่ 3.3 โครงสร้างภายในโปรแกรม

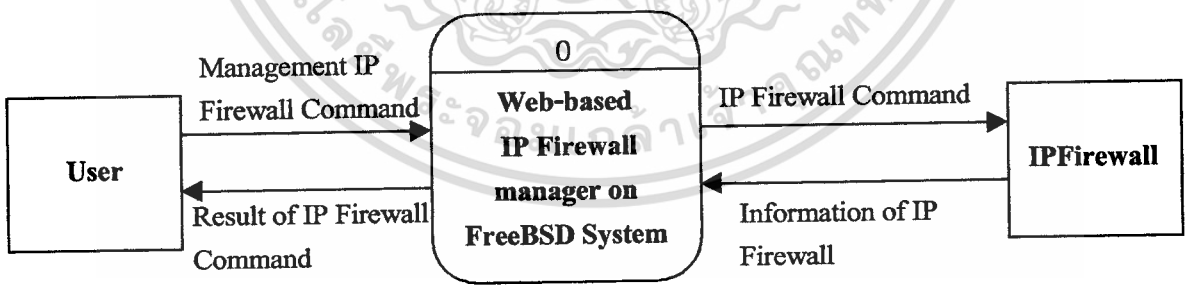
จากภาพที่ 3.3 จะเห็นว่าโปรแกรมจัดการไอพีไฟร์วอลล์ผู้ใช้สามารถติดต่อกับโปรแกรมผ่านทางเว็บบราวเซอร์ซึ่งก็เป็นในลักษณะแบบกราฟฟิกอย่างหนึ่งที่ทำหน้าที่อยู่ส่วนหน้า(Front-end) ส่วนการประมวลผลตามฟังก์ชันต่างๆที่ระบบสามารถรองรับได้จะทำงานอยู่ด้านหลัง(Back-end) คือเป็นสคริ๊ปทำงานตามที่ผู้ใช้ป้อนคำสั่งเข้าไป ในบทต่อไปเราจะอธิบายถึงการออกแบบการทำงานในโมดูลต่างๆที่สำคัญ

เมื่อเข้าใจภาพรวมของระบบในส่วนต่อไปจะแสดงการไหลของข้อมูลที่เกิดขึ้นทั้งหมดในระบบ

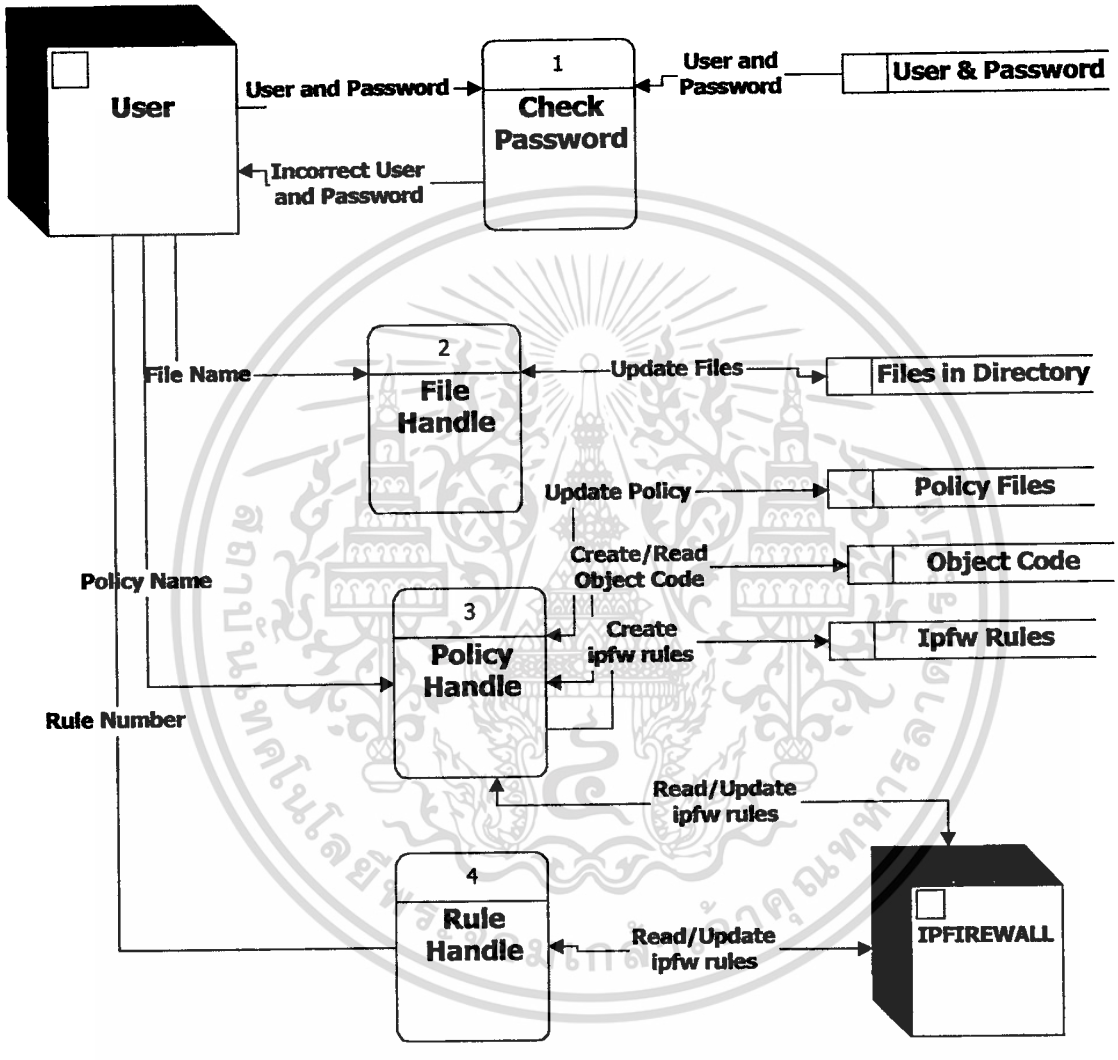
3.2 ผังการไหลของข้อมูลโปรแกรมจัดการไอพีไฟร์วอลล์บนระบบปฏิบัติการฟรีบีเอสดี



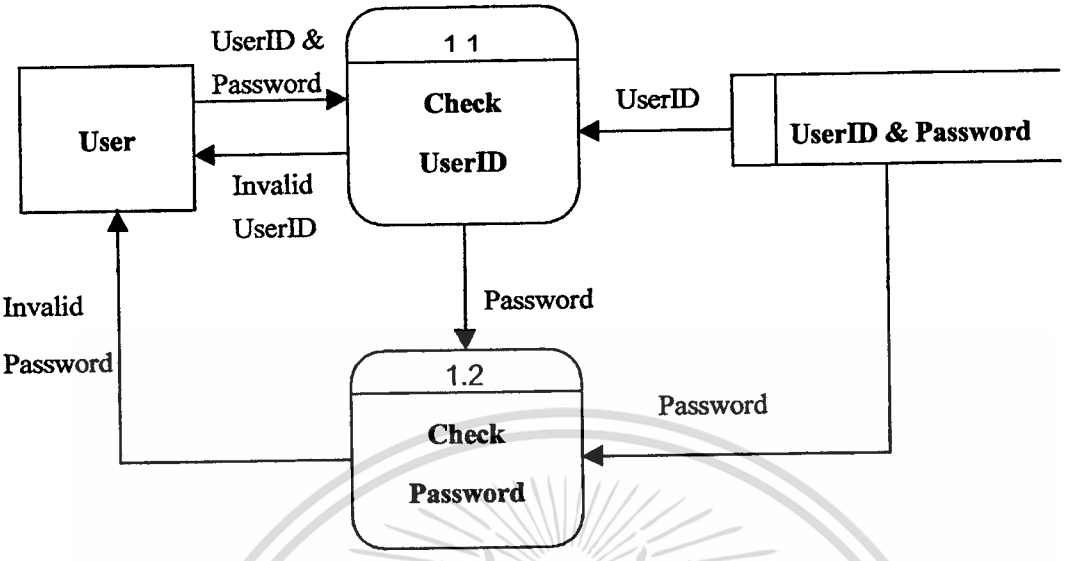
ภาพที่ 3.4 แสดงสัญลักษณ์ของผังการไหลของข้อมูล



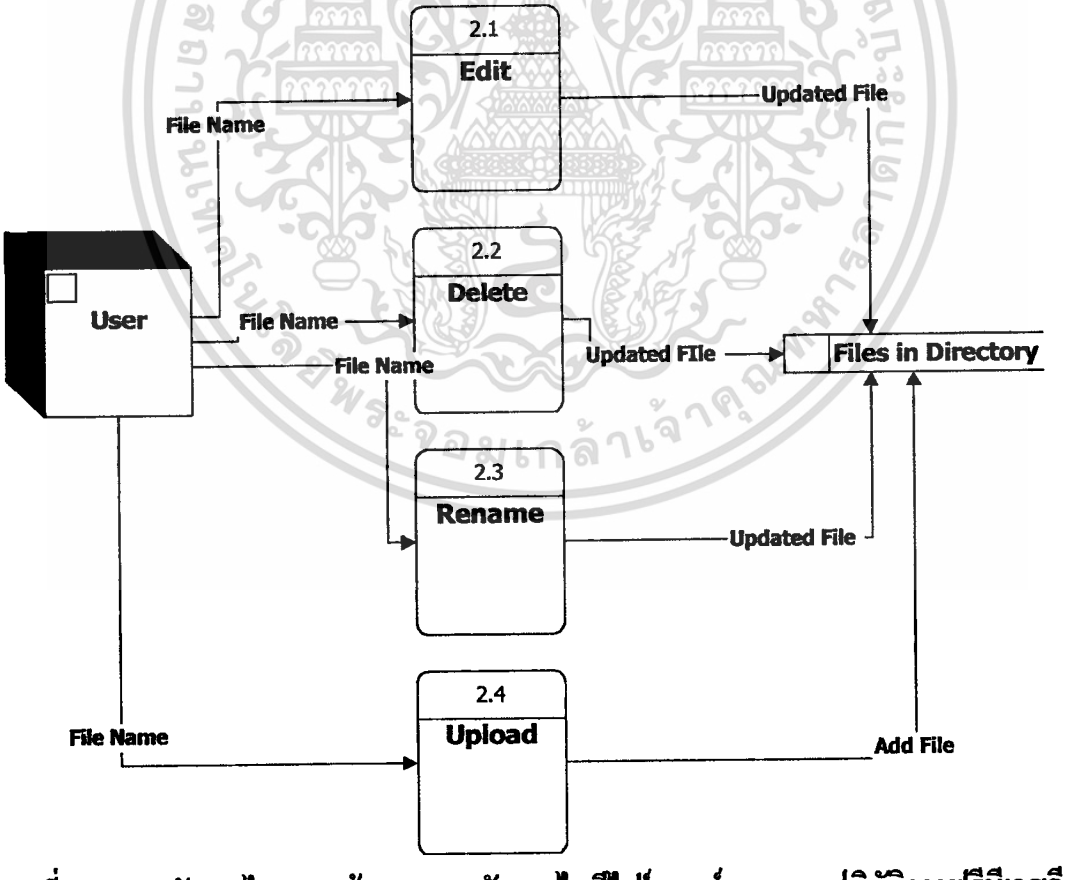
ภาพที่ 3.5 แสดง Context Diagram ของระบบจัดการไอพีไฟร์วอลล์บนระบบปฏิบัติการฟรีบีเอสดี



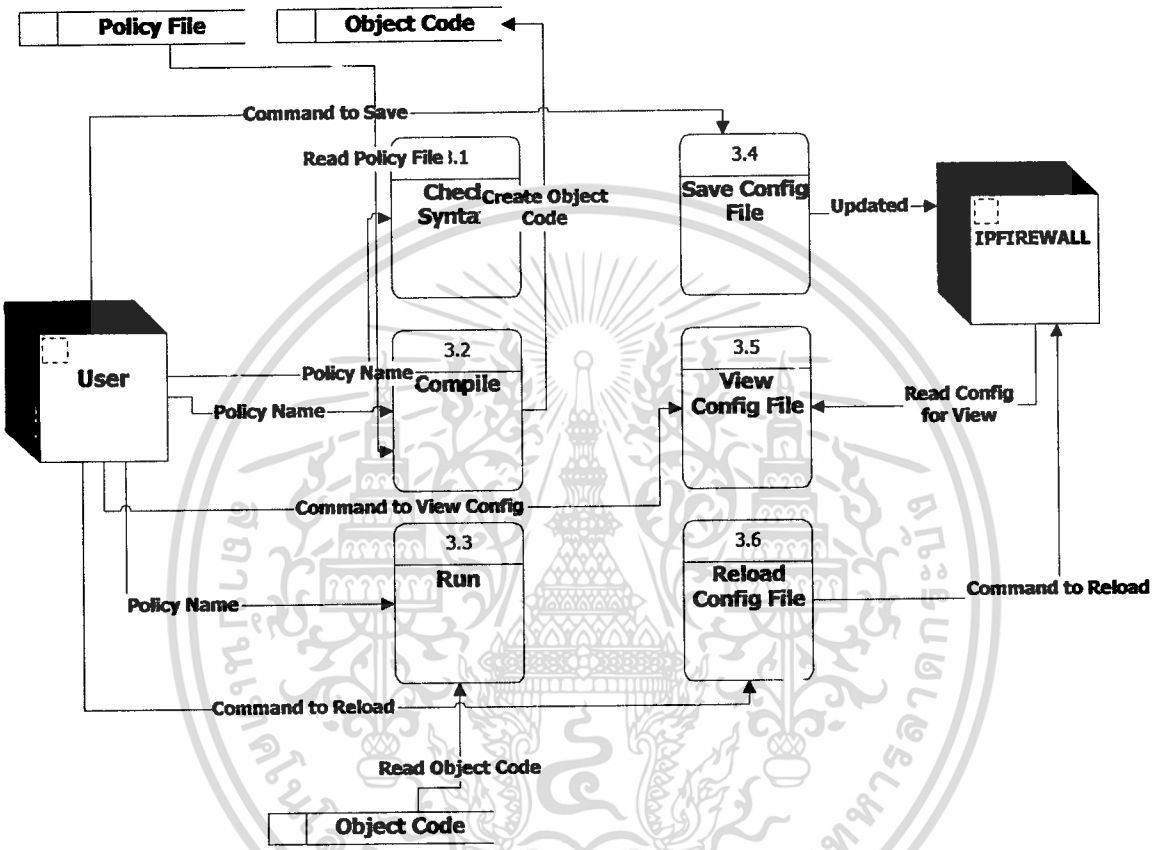
ภาพที่ 3.6 แสดงผังการไหลของข้อมูลระบบจัดการไอพีไฟร์วอลล์บนระบบปฏิบัติการฟรีบีเอสดี ระดับที่ 0 (DFD level 0)



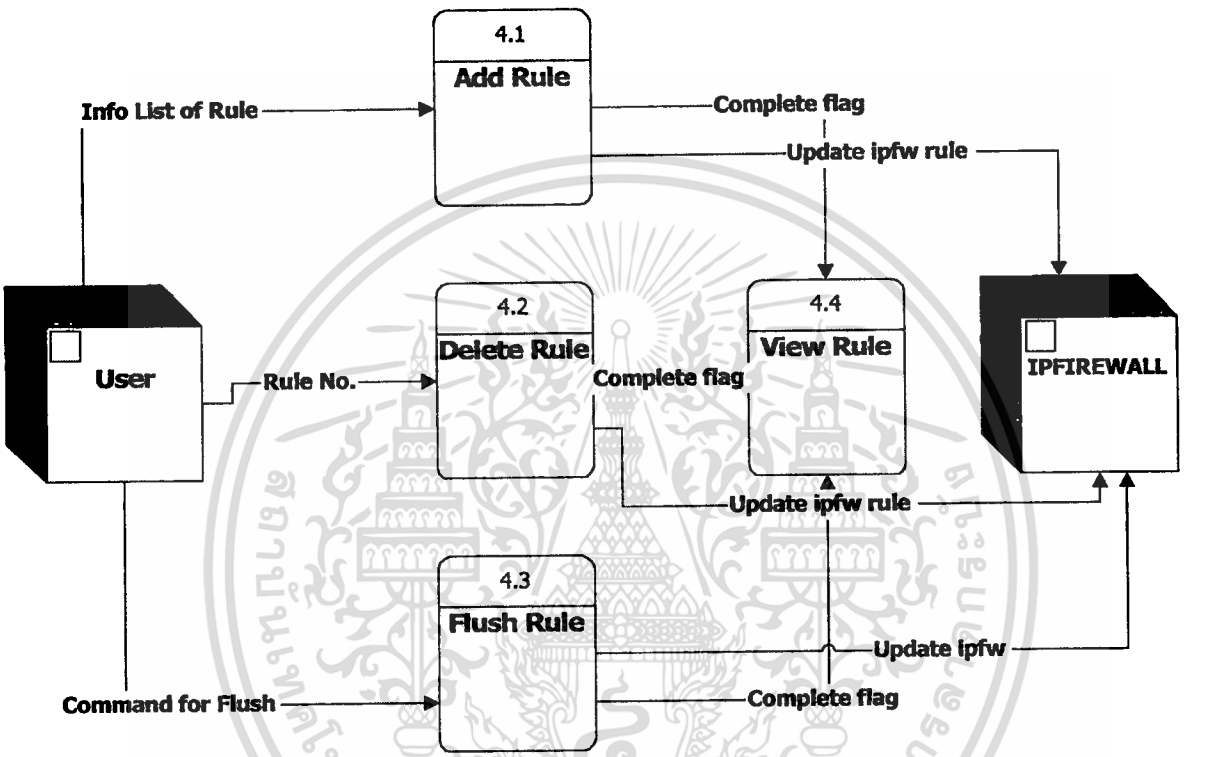
ภาพที่ 3.7 แสดงผังการไหลของข้อมูลระบบจัดการไอพีไฟร์วอลล์บนระบบปฏิบัติการฟรีบีเอสดี ระดับที่ 1 กระบวนการที่ 1 (DFD level 1 Process 1)



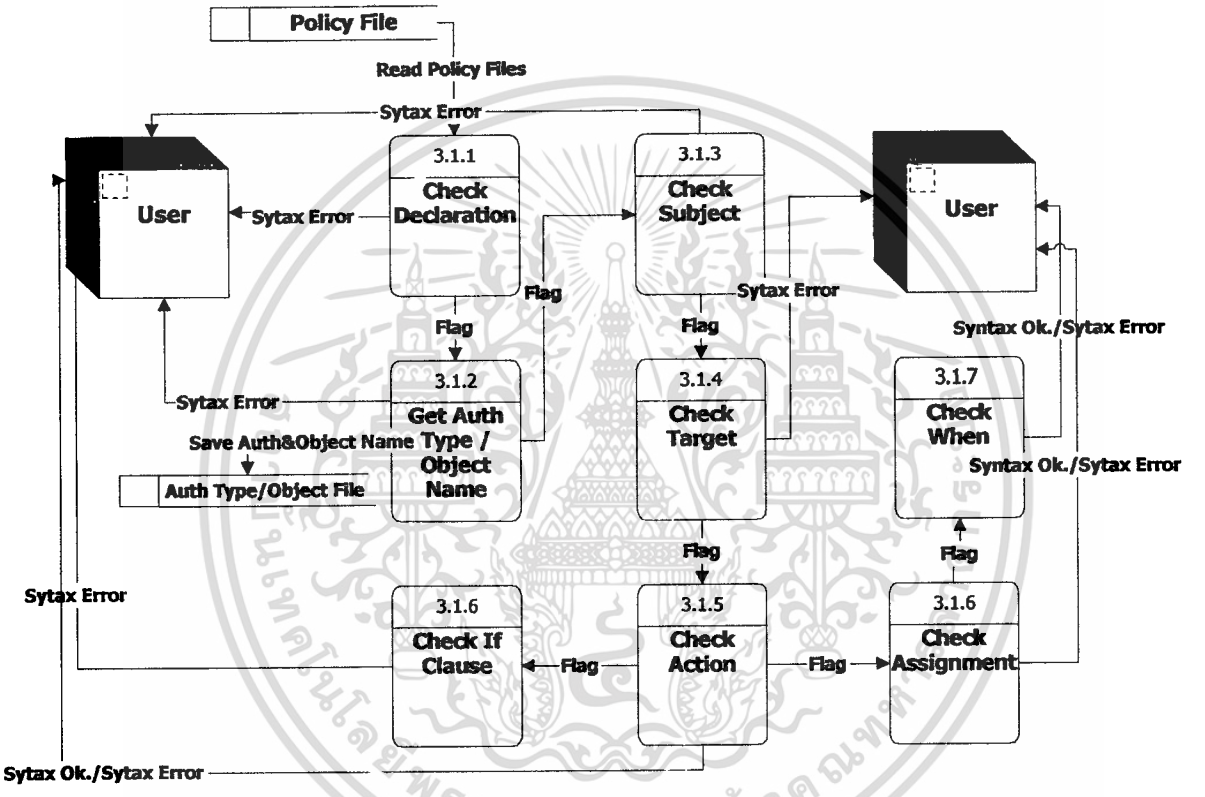
ภาพที่ 3.8 แสดงผังการไหลของข้อมูลระบบจัดการไอพีไฟร์วอลล์บนระบบปฏิบัติการฟรีบีเอสดี ระดับที่ 1 กระบวนการที่ 2 (DFD level 1 Process 2)



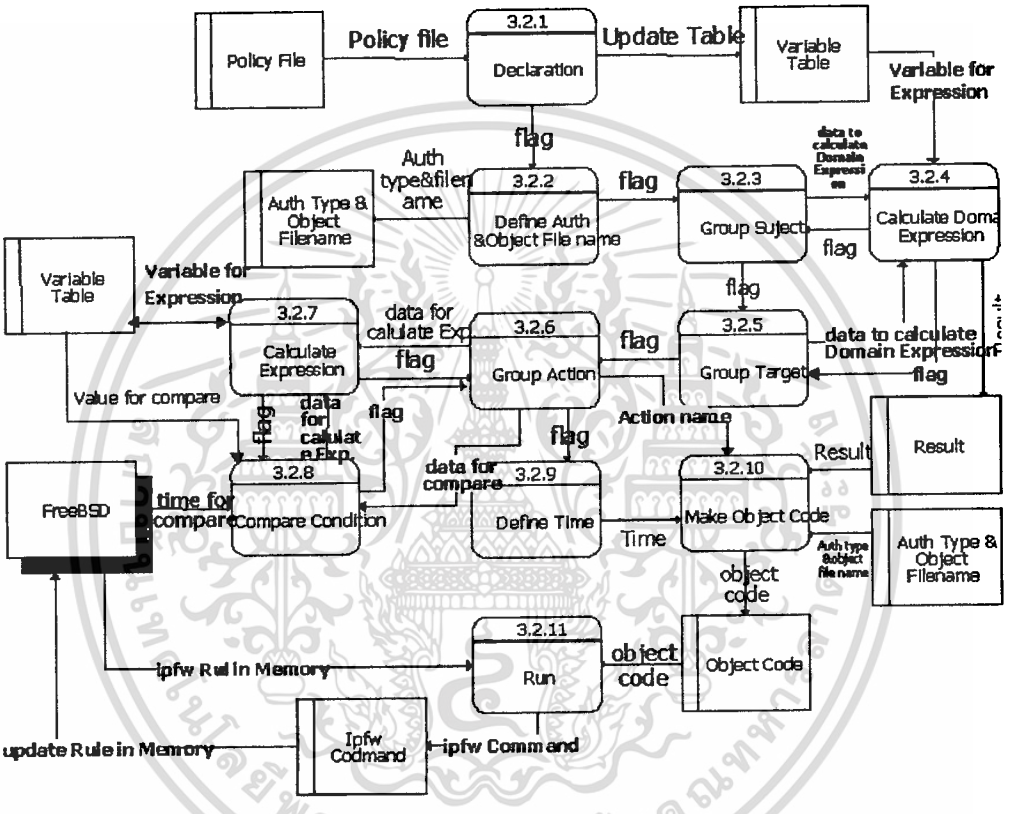
ภาพที่ 3.9 แสดงผังการไหลของข้อมูลระบบจัดการไอพีไฟร์วอลล์บนระบบปฏิบัติการฟรีบีเอสดี ระดับที่ 1 กระบวนการที่ 3 (DFD level 1 Process 3)



ภาพที่ 3.10 แสดงผังการไหลของข้อมูลระบบจัดการไอพีไฟร์วอลล์บนระบบปฏิบัติการฟรีบีเอสดี ระดับที่ 1 กระบวนการที่ 4 (DFD level 1 Process 4)



ภาพที่ 3.11 แสดงผังการไหลของข้อมูลระบบจัดการไอพีไฟร์วอลล์บนระบบปฏิบัติการฟรีบีเอสดี ระดับที่ 2 กระบวนการที่ 3.1 (DFD level 2 Process 3.1)



ภาพที่ 3.12 แสดงผังการไหลของข้อมูลระบบจัดการไอพีไฟร์วอลล์บนระบบปฏิบัติการฟรีบีเอสดี ระดับที่ 2 กระบวนการที่ 3.2 (DFD level 2 Process 3.2)

จากผังการไหลของข้อมูลจะเห็นว่ามีการเข้าไปจัดการฐานข้อมูลถึง 3 ตัว คือ IPFirewall Policy, IPFirewall Rule และ IPFirewall Configuration (rc.firewall) ซึ่งในที่นี้จะขออธิบายในแต่ละฐานข้อมูลดังต่อไปนี้

- IPFirewall Policy เป็นฐานข้อมูลที่เก็บนโยบายต่างๆ ที่ผู้ใช้ทำการสร้างขึ้นมา โดยจะเก็บอยู่ในรูปของเท็กซ์ไฟล์ (text file) โดยจะเก็บชื่อของไฟล์ที่เป็นชื่อของนโยบาย (policy name) ซึ่งผู้ใช้ได้ตั้งไว้ตอนสร้างนโยบาย โดยจะมีนามสกุลเป็น .pol ซึ่งรายละเอียดภายในของไฟล์นี้จะเป็นไปตามรูปแบบของภาษาพอนเดอร์
- IPFirewall Object Code เป็นฐานข้อมูลที่เก็บกฎต่างๆ ที่ผู้ใช้ทำการสร้างขึ้น โดยจะเก็บในรูปแบบของเท็กซ์ไฟล์ ซึ่งจะอยู่ในรูปแบบที่ใกล้เคียงกับคำสั่งของไอพีไฟร์วอลล์แต่จะมีการเพิ่มเติมฟิลด์ขึ้นมาคือฟิลด์ที่ใช้ในการกำหนดลำดับความสำคัญซึ่งจะขออธิบายรายละเอียดภายหลัง ต่อไปจะเป็นการแสดงตัวอย่างคำสั่งของไอพีไฟร์วอลล์ในรูปแบบทั่ว ๆ ไปก่อนที่จะแสดงในรูปแบบของออปเจ็คโค้ด

1) การใช้โปรโตคอลไอพี (IP) เป็นคีย์สำหรับการกรอง

```
Ipfw add [rule no.] allow ip from $SourceAddr to $DestAddr [options]
Ipfw add [rule no.] deny  ip from $SourceAddr to $DestAddr [options]
Ipfw add [rule no.] divert [port] ip from $SourceAddr to $DestAddr [options]
```

ภาพที่ 3.13 รูปแบบการสร้างกฎโดยใช้โปรโตคอลไอพี

ตัวอย่าง

```
ipfw add 00001 allow ip from 192.168.0/24 to any via ed1 out
ipfw add 00002 deny ip from any to any
ipfw add 00003 divert natd ip from any to any via tun0
```

2) การใช้โปรโตคอลทีซีพี (TCP) เป็นคีย์สำหรับการกรอง

```
Ipfw add [rule no.] allow tcp from $SourceAddr $SPort to $DestAddr $DPort [options]
Ipfw add [rule no.] deny  tcp from $SourceAddr $SPort to $DestAddr $DPort [options]
Ipfw add [rule no.] reset  tcp from $SourceAddr $ SPort to $DestAddr $DPort [options]
Ipfw add [rule no.] unreach [code] tcp from $SourceAddr $ SPort to $DestAddr $DPort [options]
```

ภาพที่ 3.14 รูปแบบการสร้างกฎโดยใช้โปรโตคอลทีซีพี

ตัวอย่าง

ipfw add 10 allow tcp from any to any established

ipfw add 11 deny tcp from any to any in via tun0 setup

ipfw add 12 reset tcp from any to any 133 in via tun0

3) การใช้โปรโตคอลยูดีพี (UDP) เป็นคีย์สำหรับการกรอง

```
Ipfw add [rule no.] allow udp from $SourceAddr $Sport to $DestAddr $DPort [options]
Ipfw add [rule no.] deny  udp from $SourceAddr $Sport to $DestAddr $DPort [options]
Ipfw add [rule no.] reject udp from $SourceAddr $Sport to $DestAddr $DPort [options]
Ipfw add [rule no.] unreach [code] udp from $SourceAddr $Sport to $DestAddr $DPort [options]
```

ภาพที่ 3.15 รูปแบบการสร้างกฎโดยใช้โปรโตคอลยูดีพี

ตัวอย่าง

ipfw add 20 allow udp from x.x.x.x 53 to any 1024-65535 in recv tun0

ipfw add 21 reject udp from any to any 137 via tun0

ipfw add 22 unreach net udp from any to any 33400-33499 in recv tun0

4) การใช้โปรโตคอลไอซีเอ็มพี (ICMP) เป็นคีย์สำหรับการกรอง

```
Ipfw add [rule no.] allow icmp from $SourceAddr to $DestAddr [options]
Ipfw add [rule no.] unreach [code] icmp from $SourceAddr to $DestAddr [options]
```

ภาพที่ 3.16 รูปแบบการสร้างกฎโดยใช้โปรโตคอลไอซีเอ็มพี

ตัวอย่าง

ipfw add 30 allow icmp from any to any

ipfw add 31 unreach 13 icmp from any to any via ed1 in

5) การใช้โปรโตคอลทั้งหมด (all) เป็นคีย์สำหรับการกรอง

```
Ipfw add [rule no.] allow all from $SoureAddr to $DestAddr [options]
Ipfw add [rule no.] deny  all from $SoureAddr to $DestAddr [options]
```

ภาพที่ 3.17 รูปแบบการสร้างกฎโดยใช้โปรโตคอลทั้งหมด

ตัวอย่าง

ipfw add 40 allow all from 10.0.0.0:255.0.0.0 to any via ppp0

ipfw add 41 deny all from any to any

ต่อไปจะเป็นการแสดงผลลัพธ์ที่ได้จากการตีความไฟล์นโยบายให้เป็นออปเจ็คโค้ดแสดงได้โดยตัวอย่างต่อไปนี้

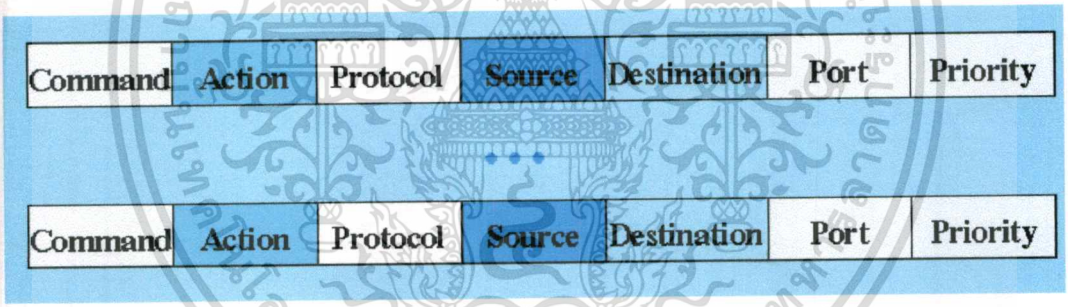
ipfw add allow tcp 161.246.10.1 to 161.246.10.3 23 with 10

ipfw add allow tcp 161.246.10.1 to 161.246.10.4 23 with 10

ipfw add allow tcp 161.246.10.2 to 161.246.10.3 23 with 10

ipfw add allow tcp 161.246.10.2 to 161.246.10.4 23 with 10

จากตังอย่างที่เห็นเมื่อนำมาพิจารณาออกแบบสร้างเป็นฐานข้อมูลจะประกอบด้วยส่วนต่างๆ ดังต่อไปนี้ คำสั่งของไอพีไฟร์วอลล์ การกระทำของกฎ ชนิดของโปรโตคอล แอดเดรสต้นทาง แอดเดรสปลายทาง พอร์ตปลายทาง และตัวเลขระบุลำดับความสำคัญ ซึ่งสามารถแสดงได้ดังนี้



ภาพที่ 3.18 รูปแบบโครงสร้างฐานข้อมูลของออปเจ็คโค้ด

- IPFirewall Config เป็นฐานข้อมูลที่เก็บรายละเอียดของโครงสร้างไอพีไฟร์วอลล์ ซึ่งจะใช้เมื่อไอพีไฟร์วอลล์ได้เริ่มทำงานเป็นครั้งแรก (เมื่อเปิดเครื่อง และให้ไฟร์วอลล์ทำงาน) จะเก็บในรูปแบบเท็กซ์ไฟล์ โดยเราจะเข้าไปจัดการในส่วนของ against people from outside your own network ซึ่งมีรายละเอียดดังนี้

  1. เราจะต้องกำหนดในส่วนของ net, mask และ ip ในตอนเริ่มต้นเป็นการประกาศค่าต่างๆ ให้ทราบก่อนที่เราจะอ้างอิงค่าต่างๆ เหล่านี้ด้วยตัวแปรแทน เช่น

     net = “192.0.2.0”

     mask = “255.255.255.0”

ip = “192.0.2.1”

2. หลังจากผ่านขั้นตอนแรก ก็จะมากำหนดรายละเอียดของกฎเพื่อให้ระบบเข้าใจ ซึ่งจะมีรูปแบบดังนี้

${fwcmd} add pass all from ${ip} to ${net} : $mask

เป็นการอนุญาตให้มีการจราจรของข้อมูลผ่านเข้ามาในเครือข่ายของผู้ใช้เอง หรือ

${fwcmd} add pass tcp from any to any established

เป็นการอนุญาตให้ tcp ผ่านเข้ามาได้เมื่อมีการสร้างช่องทางสำเร็จ เป็นต้น

จากที่ได้กล่าวมานั้นได้อธิบายการไหลของข้อมูลทั้งระบบและการออกแบบฐานข้อมูลซึ่งในโครงงานนี้จะเป็นแบบเท็กซ์ไฟล์ (text file) ในส่วนต่อไปจะขออธิบายหลักการทำงานโดยรวมในส่วนของการตีความหรือคอมไพล์ภาษาพอนเดอร์เพื่อให้เข้าใจในรายละเอียดมากขึ้น

3.3 หลักการในการพิจารณาและตีความภาษาพอนเดอร์

ในการเพิ่มนโยบายเข้าไปนั้นเราจะใส่นโยบายเข้าไป โดยเป็นไปตามรูปแบบภาษาพอนเดอร์ และเมื่อระบบพบคำเฉพาะ (keyword) ก็จะนำกฎคำสั่งที่ตามมาไปวิเคราะห์และดำเนินการตามลำดับ ดังตัวอย่างต่อไปนี้

3.3.1 การประกาศตัวแปรต่างๆที่ใช้ในโปรแกรม (Declaration)

```
int      port = 23;
int      priority = 1;
string   protocol = “tcp”;
domain A = (161.246.10.21, 161.246.10.22)
```

เมื่อพบคำเฉพาะเหล่านี้ คือ int, char, string, domain ก็จะนำตัวแปรเหล่านี้มาจัดทำเป็น ตารางเก็บค่าต่างๆ ตามชนิดของมัน โดยเราได้ออกแบบตารางดังนี้

NAME	TYPE	VALUE

เช่น

NAME	TYPE	VALUE
port	int	23
protocol	string	tcp

โดยเราจะนำค่าที่เก็บเหล่านี้ไปใช้ในการอ้างอิงถึงค่าต่างๆ และเปลี่ยนแปลงค่าเหล่านี้ เมื่อมีการกระทำทางคณิตศาสตร์เกิดขึ้น

3.3.2 คำเฉพาะ auth+, auth-

เป็นคำเฉพาะที่บอกให้ระบบรู้ว่าเป็นนโยบายทางด้านบวกหรือลบ หรือจะพิจารณาใช้เข้าใจง่ายขึ้นก็คือ จะเป็นลักษณะที่เป็นไปตาม action ที่กำหนด หรือจะเป็นในลักษณะตรงกันข้าม เช่น

ถ้าขึ้นต้นด้วย auth+ และในส่วน actionlist เป็น allow หรือ deny ความหมายของ action ก็จะเป็นไปตามนั้น แต่ถ้าขึ้นต้นด้วย auth- และในส่วนของ actionlist เป็น allow ผลที่ได้จะเป็น deny แต่ถ้าเป็น deny ผลที่ได้ก็จะเป็น allow

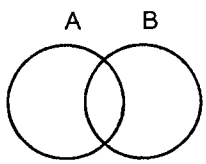
3.3.3 คำเฉพาะ Subject/Target

จะเป็นการกำหนดกลุ่มของสมาชิกต้นทางและปลายทาง โดยในการกำหนดนี้จะอยู่ในส่วนการทำงานของฟังก์ชันการจัดการนโยบาย (policy management)

โดยที่กลุ่มสมาชิกเหล่านี้สามารถนำมากระทำการทางคณิตศาสตร์ ซึ่งเรียกว่า Domain Scope Expression ซึ่งจะมีการกระทำ 3 รูปแบบ คือ

+	=	ยูเนียน
-	=	การหาผลต่าง (differential)
^	=	อินเตอร์เซ็ก

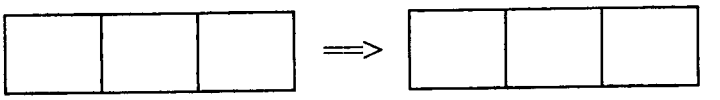
เช่น Subject A + B;



สมาชิก A รวมกับสมาชิก B โดยที่ A และ B ต้องเป็น type ประเภท domain

target 161.246.10.21;

เมื่อระบบทำการตีความ ก็จะกระจายคำสั่งไปยังสมาชิกต่างๆ ตามที่ได้กำหนด ดังรูป



∴ จำนวนกฎที่ได้ออกมาจะเท่ากับ MxN

3.3.4 คำเฉพาะ action

จะมีการกำหนดคำสั่งต่างๆ ในโครงงานนี้เราได้กำหนดคำสั่งพื้นฐานไว้ 2 คำสั่ง คือ allow และ deny โดยทั้ง 2 ฟังก์ชันต้องมีการรับพารามิเตอร์ 3 ตัวเสมอ คือ

ตัวที่ 1	เป็น service มีชนิดเป็น int
ตัวที่ 2	เป็นการกำหนดค่าความสำคัญ มีชนิดเป็น int
ตัวที่ 3	เป็นการกำหนดโปรโตคอล มีชนิดเป็น string โดยเราสามารถรับได้ คือ tcp, udp, ip, all นอกนั้นระบบจะไม่รู้จัก

เช่น allow (s, pri, prot)

นอกจากนี้ในส่วนของ action สามารถมีการกำหนดคำสั่งในเชิงในเชิงเปรียบเทียบเงื่อนไข if เพื่อนำมาใช้ในการกำหนดค่าให้กับตัวแปรได้ เช่น

```
allow (s, pri, prot)
if (time() < 1600)
{
pri = 100;
}
```

ในโครงงานนี้ได้มีการกำหนดฟังก์ชัน time() ขึ้นมาเป็นฟังก์ชันภายในด้วย ซึ่งจะได้อธิบายในรายละเอียดในหัวข้อต่อไป และจากตัวอย่างด้านบนถ้าเงื่อนไขเป็นจริงแล้ว จะมีการเปลี่ยนแปลงค่า pri ให้เท่ากับ 100

3.3.5 คำเฉพาะ when

เป็นการใช้เพื่อให้ระบบเข้าใจว่าจะเป็นการกำหนดให้กฎต่างๆ เมื่อถูกตีความแล้วทำงานเมื่อมีเวลาเป็นเท่าไร โดยที่ เราจะใช้ฟังก์ชัน time() เป็นตัวกำหนดระยะเวลา

ฟังก์ชัน time() จะรับพารามิเตอร์ ด้วยกัน 2 แบบ คือ

1. เวลาในรูปแบบ HHMM เช่น 1644 คือ เวลา 16:44 น.
2. วันเดือนปี ในรูปแบบ DDMMYYYY เช่น 01022001 คือ วันที่ 1 กุมภาพันธ์ 2001

จะเห็นว่า when นี้ จะนำไปใช้ในส่วนของฟังก์ชันเพิ่มเติมในการทำ scheduling ซึ่งจะมีรูปแบบดังต่อไปนี้

```
when        time (0944);
```

หมายถึงในกฎ ต่าง ๆ ที่ระบบตีความได้เริ่มทำงาน ณ. เวลา 9:44 น.

เมื่อเข้าใจในหลักการตีความหรือคอมไฟล์แล้วต่อไปจะแสดงให้เห็นว่าเมื่อผู้ใช้ป้อนนโยบายในรูปแบบของภาษาพอนเดอร์แล้วระบบจะตีความออกมาอย่างไร โดยพิจารณาตามตัวอย่างต่อไปนี้

รูปแบบภาษาพอนเดอร์ในรูปทั่วไป

```
Declaration {
        int         Variable
        string      Variable
        auth+/-     Rule Name {
                    Subject         Domain Scope Expression ;
                    Target          Domain Scope Expression ;
                    action          actionlist ;
                    when            time-to-active ;
        }
```

ตัวอย่างที่ 1

```
int         priority = 1
int         service = 23;
string      protocol = 'tcp';
domain A = (161.246.10.21, 161.246.10.22)
auth+       testpolicy1 {
```

```
        Subject     A;
        Target      161.246.10.23;
        action      allow (service, priority, protocol);
    }
```

เมื่อทำการคอมไพล์และรันแล้วจะได้รูปแบบคำสั่งดังต่อไปนี้

ipfw add 1 allow tcp from 161.246.10.21 to 161.246.10.23 23

ipfw add 1 allow tcp from 161.246.10.22 to 161.246.10.23 23

จากนั้นระบบจะนำไปจัดลำดับตัวเลข (Rule Number) ที่ซ้ำกันอีกครั้งเพื่อความถูกต้อง

ตัวอย่างที่ 2

```
int     priority = 1
int     service = 80;
string  protocol = 'udp';
auth-   testpolicy2 {
        Subject 161.246.10.21 + 161.246.10.22
        Target  161.246.10.23
        action  allow (service, priority, protocol)
                {
                    priority = 100;
                }
        when    (time(0914));
}
```

ผลลัพธ์ที่ได้ดังนี้

ipfw add 100 deny udp from 161.246.10.21 to 161.246.10.23 80

ipfw add 100 deny udp from 161.246.10.22 to 161.246.10.23 80

และกฎจะทำงานเมื่อเวลา 9:14 น.

ตัวอย่างที่ 3

```
int     priority = 5;
int     service = 22;
string  protocol = 'tcp';
```

```
domain A = (161.246.10.21, 161.246.10.22)
domain B = (161.246.10.22, 161.246.10.24)
auth+   testpolicy3 {
        Subject         A ^ B
        Target          A – B
        Action          allow (service, priority, protocol)
                        if (time() <= 1200)
                        {
                                priority = 300;
                        }
}
```

ผลลัพธ์ที่ได้ ถ้าเวลาตอนนี้เป็นเวลาก่อน 12:00

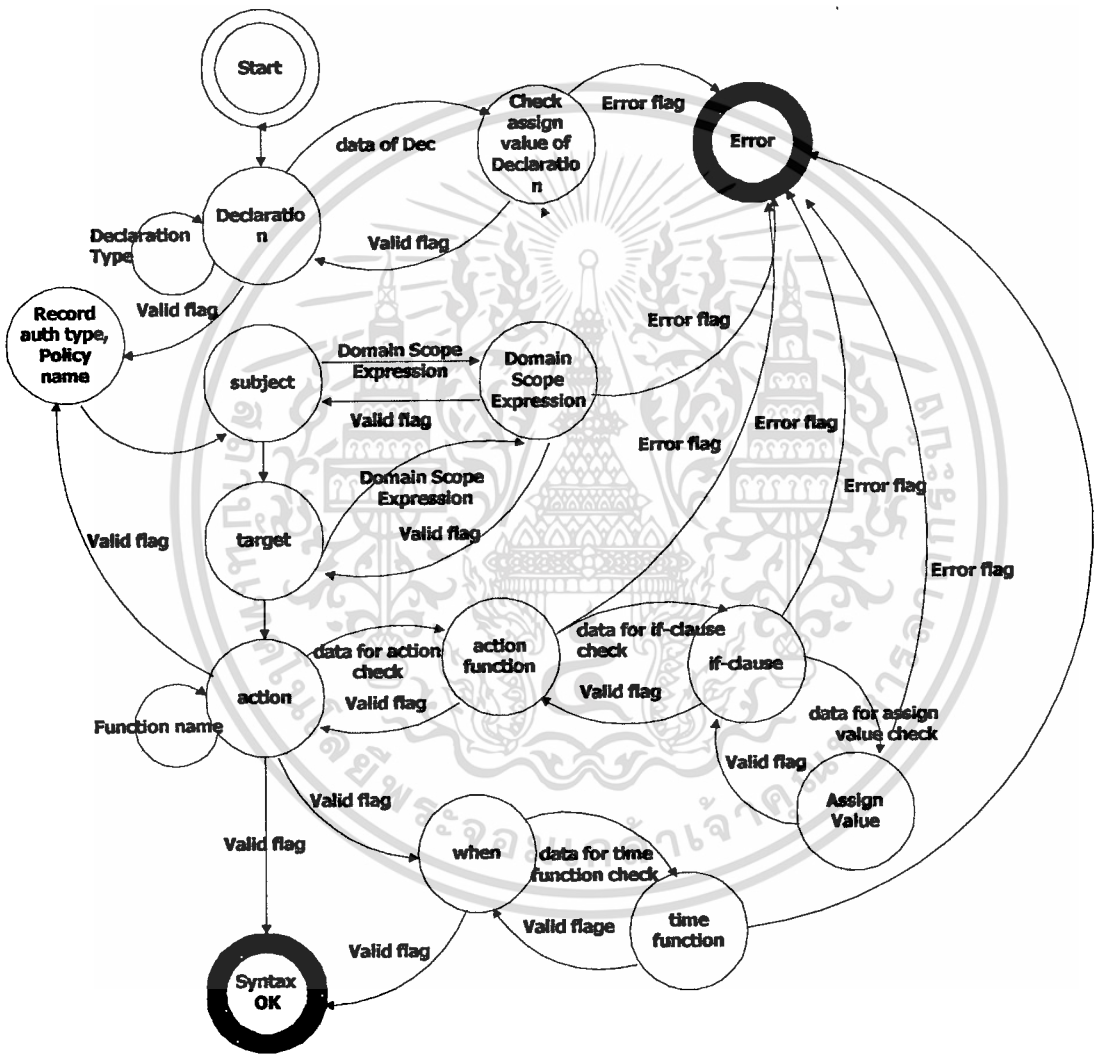
ipfw add 300 allow tcp from 161.246.10.22 to 161.246.10.21 22

ถ้าเวลาตอนนี้หลังเวลา 12:00

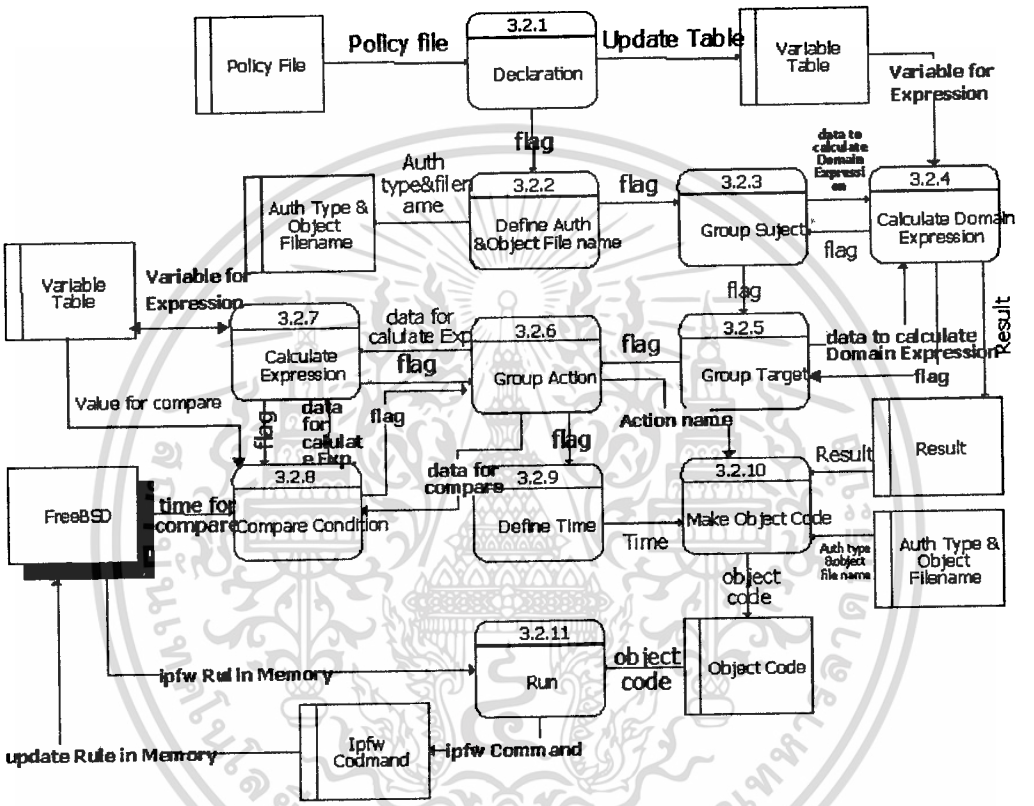
ipfw add 5 allow tcp from 161.246.10.22 to 161.246.10.21 22

จากที่ได้กล่าวมาข้างต้นจะเป็นการอธิบายหลักการพิจารณาการตรวจสอบและตีความนโยบายในแบบกว้าง ๆ ซึ่งต่อไปเราจะแสดงไดอะแกรมขั้นตอนการตรวจสอบความถูกต้องของภาษาในการกำหนดนโยบายดังภาพที่ 3.20และการตีความดังภาพที่3.21

ในบทนี้ได้แสดงให้เห็นถึงการไหลของข้อมูลในระบบทั้งหมด รวมถึงแสดงฐานข้อมูลในแต่ละประเภท และการตีความของภาษาพอนเดอร์ในบทต่อไปเราจะพิจารณาในการทำงานในฟังก์ชันต่าง ๆ เพื่อแสดงให้เห็นถึงขั้นตอนการทำงาน



ภาพที่ 3.19 ขั้นตอนในการตรวจสอบความถูกต้องของการกำหนดนโยบาย



ภาพที่ 3.20 ขั้นตอนในการตีความนโยบาย

บทที่ 4
การพัฒนาระบบงาน

บทนี้จะได้กล่าวถึงเครื่องมือที่จะนำมาใช้ในการพัฒนาโครงงาน ขั้นตอนในการทำให้สามารถใช้บริการของไอพีไฟร์วอลล์บนระบบปฏิบัติการฟรีบีเอสดี และขั้นตอนในการทำงานต่างๆในแต่ละฟังก์ชันการทำงานโดยละเอียด

4.1 เครื่องมือที่ใช้ในการพัฒนาโปรแกรม

- Microsoft Word ใช้ในการผลิตเอกสารประกอบการพัฒนาโครงงาน
- Perl เป็นภาษาที่จะนำมาพัฒนาโครงงาน
- Apache Webserver ใช้เป็นเว็บเซิร์ฟเวอร์บนระบบปฏิบัติการฟรีบีเอสดี
- EditPlus เป็นอิดิเตอร์ที่ใช้ในการเขียนโครงงาน
- Internet Explorer ใช้เป็นบราวเซอร์ที่ใช้ทดสอบโครงงาน
- Visio และ Adobe Photoshop ใช้ในการวาดภาพประกอบเช่น โฟลว์ชาร์ท (Flow Chart) รูปภาพและไดอะแกรม (Diagram)

4.2 การเลือกโครงร่างของไอพีไฟร์วอลล์

ส่วนหลักของระบบไอพีไฟร์วอลล์อยู่ในเคอร์เนล ถ้าต้องการเพิ่มตัวเลือก 1 ตัวหรือมากกว่านั้นในไฟล์โครงร่างภายในเคอร์เนล จำเป็นต้องคอมไพล์เคอร์เนลอีกครั้ง เพื่อให้การทำงานต่างๆถูกต้อง

ในปัจจุบันมี 3 ตัวเลือกที่ใช้ในไอพีไฟร์วอลล์สำหรับจัดการโครงร่างเคอร์เนล

1) Options IPFIREWALL เป็นการคอมไพล์ตัวเคอร์เนลให้สร้างโค้ดสำหรับการกรองแพ็คเก็ต
2) Options IPFIREWALL_VERBOSE เป็นโค้ดที่อนุญาตให้มรการบันทึก(Log)แพ็คเก็ตผ่านคำสั่ง syslog(8) ถ้าไม่มีการใช้ตัวเลือกนี้ แล้วมีการระบุแพ็คเก็ตที่ควรจะถูกบันทึกในกฎการกรอง จะไม่มีการเปลี่ยนแปลง

3) Options IPFIREWALL_VERBOSE_LIMIT = 10 เป็นการจำกัดจำนวนแพ็คเก็ตที่ถูกล็อกผ่านคำสั่ง syslog(8) ต่อหนึ่งกฎ และอาจต้องใช้ตัวเลือกนี้ในสภาพแวดล้อมที่มีผู้รุกราน

4.3 การออกแบบฟังก์ชันการทำงานต่างๆ

4.3.1 ฟังก์ชันการทำงานทั่วไป

4.3.1.1 การเพิ่ม (Add)

ในการ Add เราจะสามารถทำได้ 2 กรณี คือ

1. การ Add เป็นนโยบาย โดยที่จะอยู่ในรูปแบบของภาษาพอนเดอร์ แล้วบันทึกเป็นไฟล์นโยบายเก็บไว้
2. การ Add เป็นกฎ คือจะให้ใส่กฎไปในรูปแบบของไอพีไฟร์วอลล์ แล้วบันทึกเป็นไฟล์ของกฎเก็บไว้ได้

ซึ่งแต่ละวิธีมีขั้นตอนดังต่อไปนี้

4.3.1.1.1 การเพิ่มนโยบาย

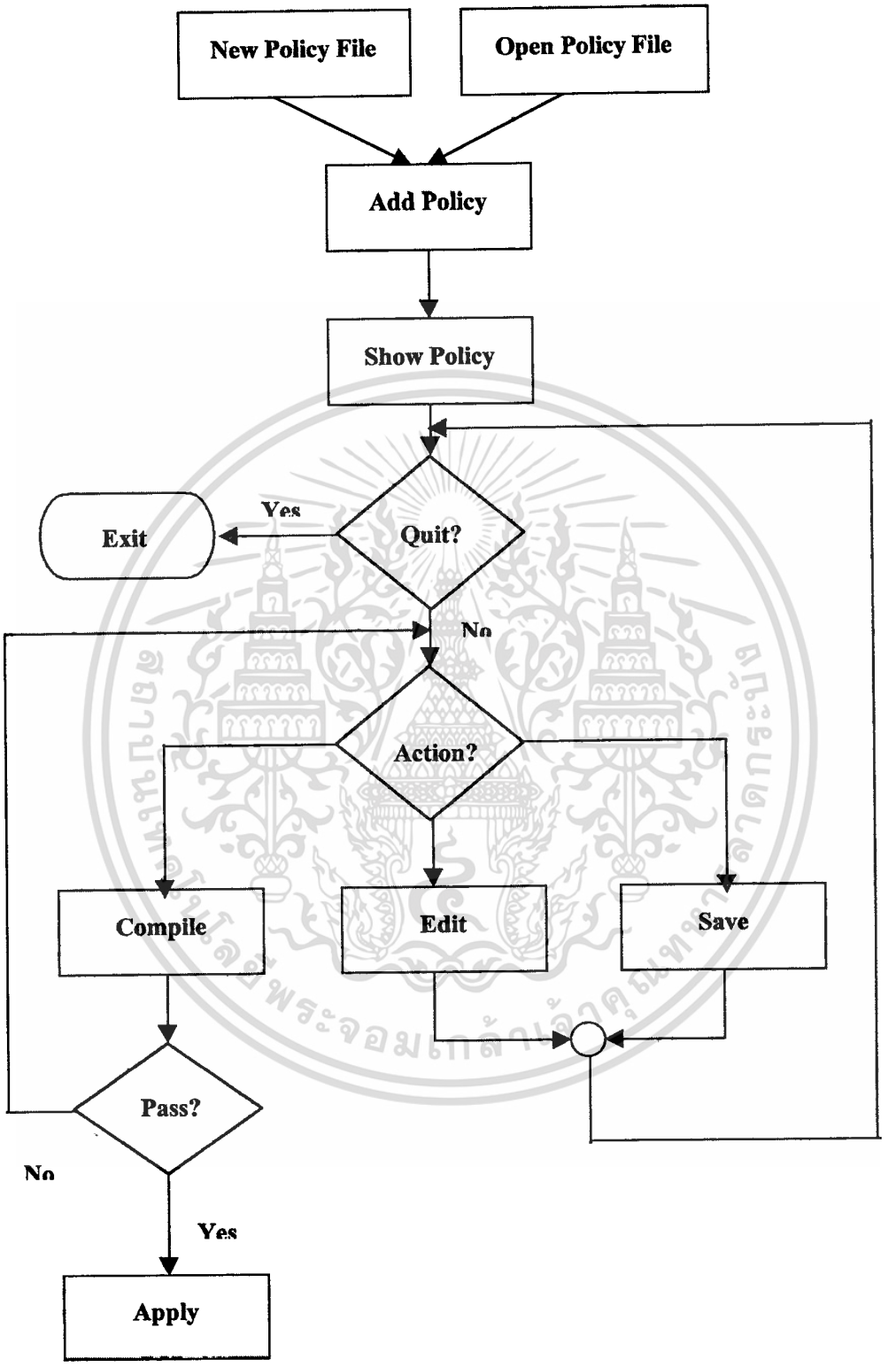
1. เลือกหัวข้อ Add Policy
2. เลือกว่าจะ Add Policy ไฟล์ใหม่ หรือไฟล์เดิมที่มีอยู่แล้ว
3. ทำการสร้างนโยบายให้อยู่ในรูปแบบของภาษาพอนเดอร์
4. บันทึก (save)
5. ทำการคอมไพล์ (compile) จะเป็นการตรวจสอบปัญหาทุกอย่างที่ระบบรองรับ
6. ทำการสั่งให้นโยบายที่ทำไว้เริ่มทำงาน (Apply)

ขั้นตอนเหล่านี้สามารถแสดงเป็นโฟร์วชาร์ทดังภาพที่ 4.1

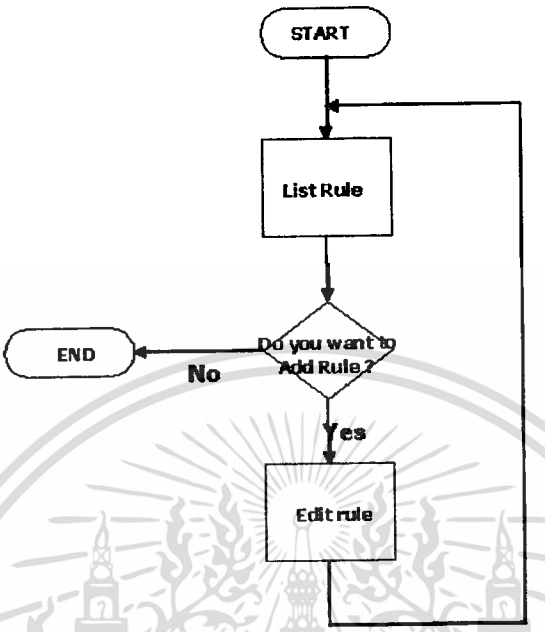
4.3.1.1.2 การเพิ่มกฎ

1. ทำการเปิดไฟล์กฎใหม่หรือเรียกใช้จากไฟล์เดิม
2. ทำการเพิ่มกฎตามรูปแบบของไอพีไฟล์วอลล์
3. ตรวจสอบความซ้ำซ้อนของกฎ
4. บันทึก
5. สั่งให้กฎที่ทำการเพิ่มนั้นทำงาน (Apply)

ขั้นตอนเหล่านี้สามารถแสดงเป็นโฟร์วชาร์ทดังภาพที่ 4.2



ภาพที่ 4.1 โฟลชาร์ทการเพิ่มนโยบาย

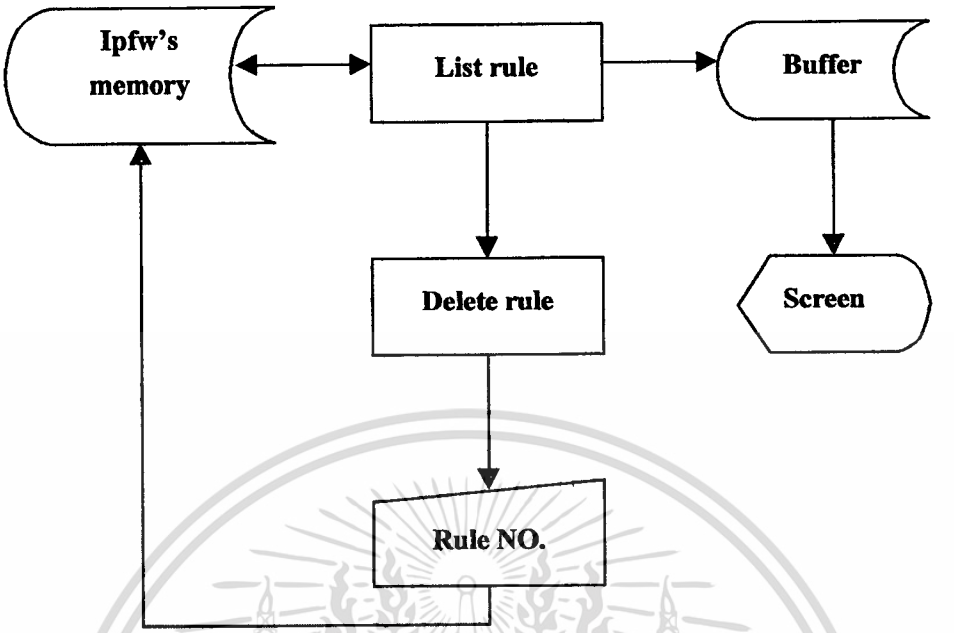


ภาพที่ 4.2 โฟลชาร์ทการเพิ่มกฎ

4.3.1.2 การลบ (Delete)

การลบกฎที่ใช้ ณ ปัจจุบันออกทีละกฎ โดยกำหนดหมายเลขกำกับลงไปด้วย การใช้คำสั่ง delete นี้จะไม่เกี่ยวข้องกับไฟล์กฎแต่อย่างใด เพราะจะกระทำเฉพาะกับกฎที่ถูกเรียกใช้อยู่เท่านั้น (ในหน่วยความจำของไอพีไฟร์วอลล์) ทำให้การเรียกใช้คำสั่งนี้สามารถกระทำได้โดยตรง ซึ่งจะมีขั้นตอนดังนี้

1) ทำการตรวจสอบรายการของกฎก่อน โดยใช้คำสั่ง List Rule ในเมนูของการแก้ไข และคำสั่งที่ใช้ในการแสดงรายการของกฎคือ system("ipfw –a list")
2) เมื่อเรียกใช้คำสั่ง Delete จะปรากฏหน้าต่างของการลบกฎขึ้นมา ให้ทำการใส่หมายเลขของกฎที่ต้องการลบลงไป และคำสั่งที่ใช้ในการลบคือ system("ipfw delete [rule no.]")
3) กดปุ่มตกลง เพื่อยืนยันการลบกฎ หลังจากนั้นกฎก็จะถูกลบออกไป



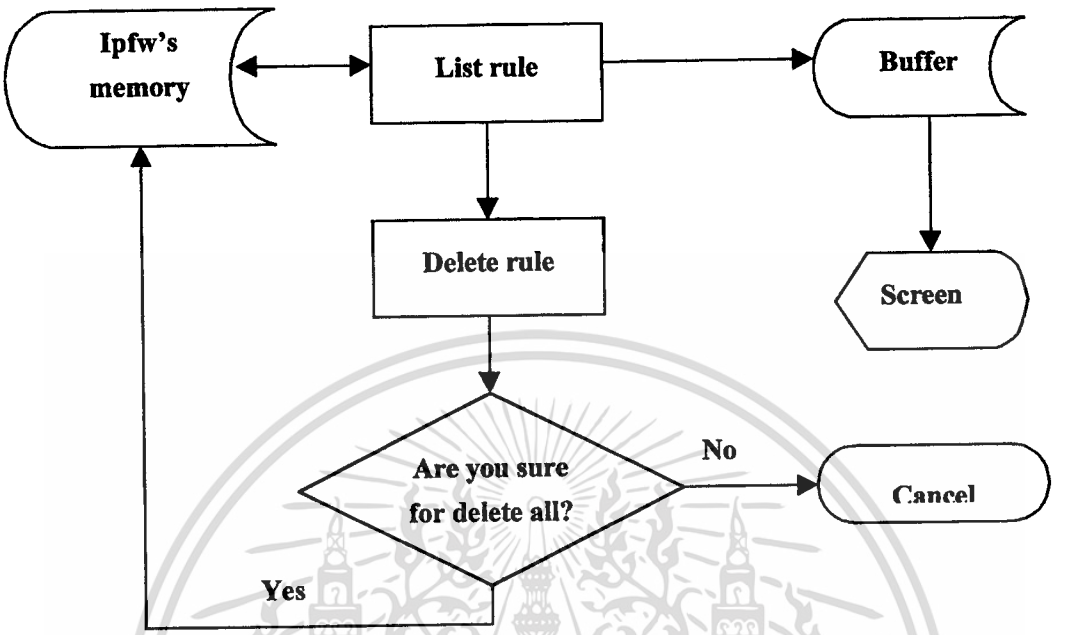
ภาพที่ 4.3 การทำงานของการลบกฎ

4.3.1.3 การลบทั้งหมด (Flush)

การลบกฎที่ใช้ ณ ปัจจุบันออกไปทั้งหมด การใช้คำสั่งนี้สามารถกระทำได้โดยตรงกับกฎที่ถูกเรียกใช้ เหมือนกับการเรียกใช้คำสั่ง delete มีขั้นตอนดังนี้

1) ทำการตรวจสอบรายการของกฎก่อน โดยเรียกใช้คำสั่ง List Rule ในเมนูของการแก้ไข
2) หลังจากนั้นจะทำคำสั่ง Flush
3) เมื่อคำสั่ง Flush ถูกเรียกใช้จะมีหน้าต่างปรากฏขึ้นมาเพื่อถามความแน่ใจในการลบกฎทั้งหมด โดยที่
   - ถ้าตอบ Yes กฎจะถูกลบออกไปทั้งหมด
   - ถ้าตอบ No ยกเลิกการลบกฎ

   และคำสั่งที่ใช้ในการลบกฎทั้งหมดคือ system("ipfw –q flush")

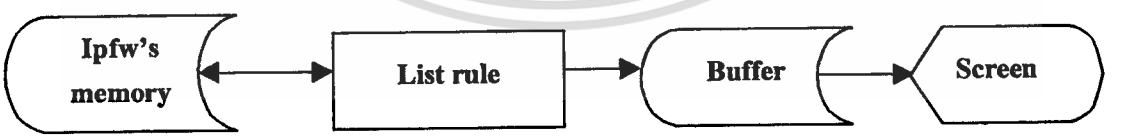


ภาพที่ 4.4 การทำงานของการลบกฎทั้งหมด

4.3.1.4 การแสดงบัญชี (List)

การแสดงกฎที่มีใช้ทั้งหมดในปัจจุบันมีขั้นตอนดังนี้

1) เลือกคำสั่ง List Rule ในเมนูของการแก้ไข
2) เมื่อคำสั่งนี้ถูกเรียกใช้ รายการของกฎจะปรากฏยังที่ว่างที่ได้เตรียมไว้ และคำสั่งที่ใช้ในการแสดงรายการของกฎ คือ system(ipfw –a list)



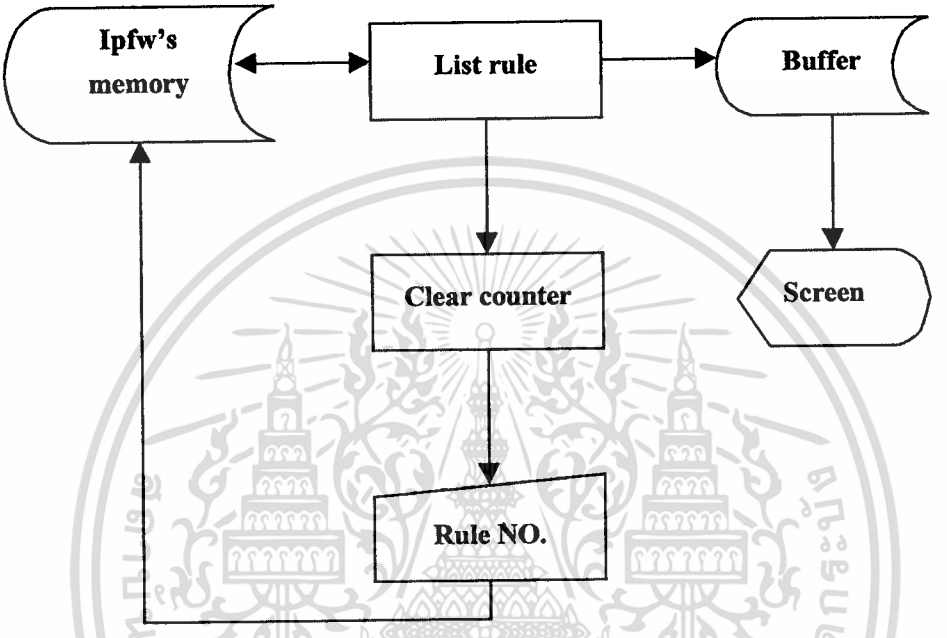
ภาพที่ 4.5 การทำงานของการแสดงรายการของกฎ

4.3.1.5 การลบค่าเคาท์เตอร์ (Clear counter)

การลบค่าตัวนับแพ็คเก็ตให้เป็นศูนย์ มีขั้นตอนดังนี้

1) ทำการตรวจสอบรายการของกฎก่อน โดยใช้คำสั่ง List Rule ในเมนูของการแก้ไข และคำสั่งที่ใช้ในการแสดงรายการของกฎ คือ system("ipfw –a list") โดยที่ –a จะมีการแสดงค่าของตัวนับแพ็คเก็ตด้วย

2) เมื่อเรียกใช้คำสั่ง Clear counter จะปรากฏหน้าต่างของการลบค่าตัวนับแพ็คเก็ตขึ้นมา ให้ทำการใส่หมายเลขของกฎที่ต้องการลบค่าลงไป และคำสั่งที่ใช้ในการลบกฎคือ system("ipfw –q clear [rule no.]")
3) กดปุ่มตกลงเพื่อยืนยันการลบกฎ หลังจากนั้นค่าของตัวนับแพ็คเก็ตนั้นก็จะถูกลบออกไป



ภาพที่ 4.6 การทำงานของการลบค่าตัวนับแพ็คเก็ต

4.3.1.6 การปรับปรุงกฎ (Update rule)

การนำกฎที่แสดงอยู่บนหน้าจอ ใส่เข้าไปในหน่วยความจำของไอพีไฟร์วอลล์ เพื่อให้กฎนั้นทำงาน (ดูในรูปการเพิ่มกฎ) การกระทำนี้จะกระทำบ่อยครั้งแค่ไหน ขึ้นอยู่กับความต้องการของผู้ใช้ และคำสั่งที่ใช้คือ system("ipfw add [rule description]") โดยที่ [rule description] คือ รายละเอียดของกฎที่เราต้องการจะให้กฎนั้นทำงาน (มาจากหน้าจอ)

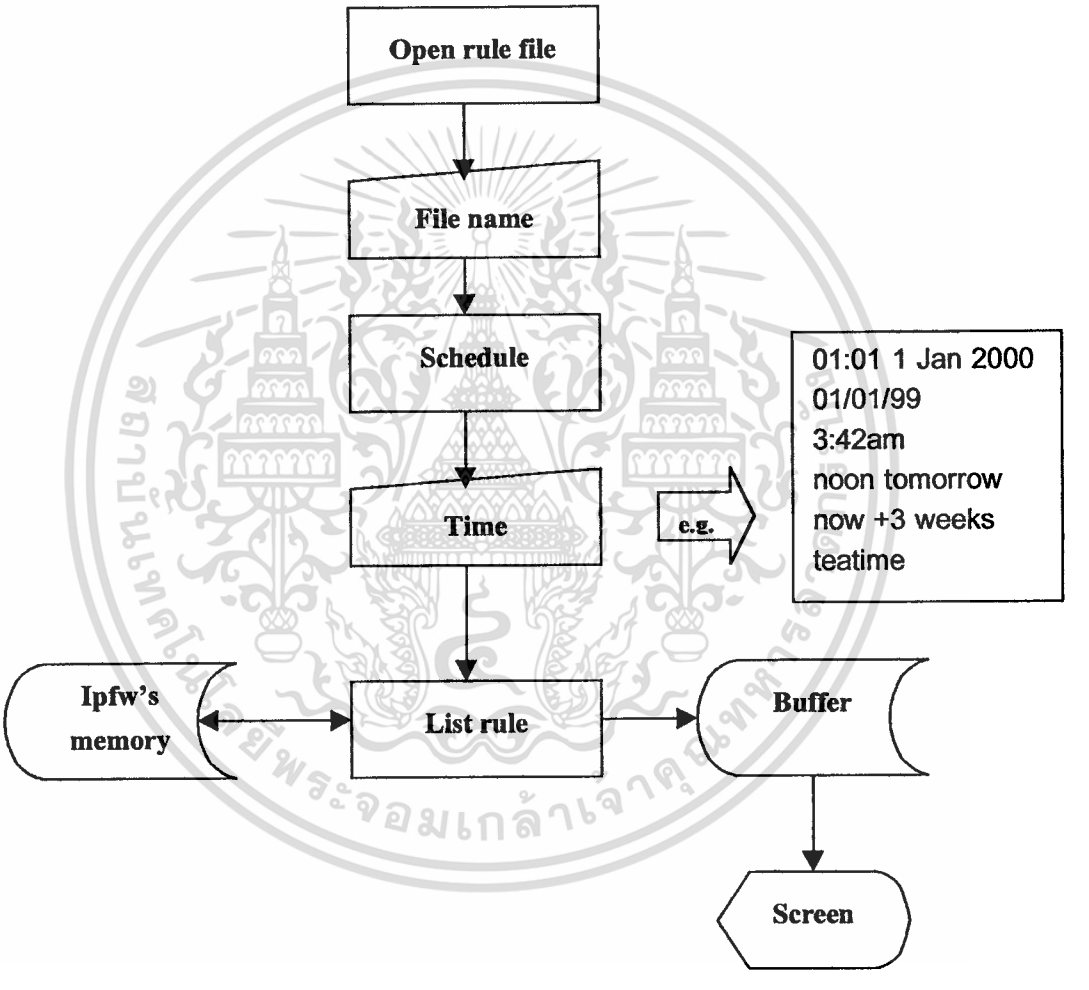
4.3.2 ฟังก์ชันการทำงานเพิ่มเติม

4.3.2.1 การตั้งเวลาการทำงาน (Scheduling)

การตั้งเวลาการกรอบแพ็คเก็ตในช่วงเวลาที่สามารถกำหนดเองได้ คำสั่งที่ใช้คือ "at" ซึ่งเป็นคำสั่งหนึ่งบนยูนิกต์ (ฟรีบีเอสดี) และการใช้คำสั่งจะทำดังนี้คือ system("at –f [file] [time]") โดยที่

file คือ ไฟล์กฎที่เราต้องการจะให้กฎนั้นทำงาน และ time คือ เวลาที่เรากำหนด ตัวอย่างของการบันทึกเวลาดูได้จากภาพที่ 4.11 และเมื่อถึงเวลาที่บันทึก กฎนั้นจึงจะเริ่มทำงาน มีขั้นตอนดังนี้

1) ทำการเปิดไฟล์กฎที่ต้องการตั้งเวลาการกรอง แล้วเลือกคำสั่ง Scheduling
2) บันทึกเวลาที่ต้องการให้กฎนี้ทำงาน เช่น
   - ต้องการเวลา 4 โมงเย็น --> 16:00 หรือ 4:00pm
   - ต้องการเวลา 4 โมงเย็นพรุ่งนี้ --> 16:00 tomorrow



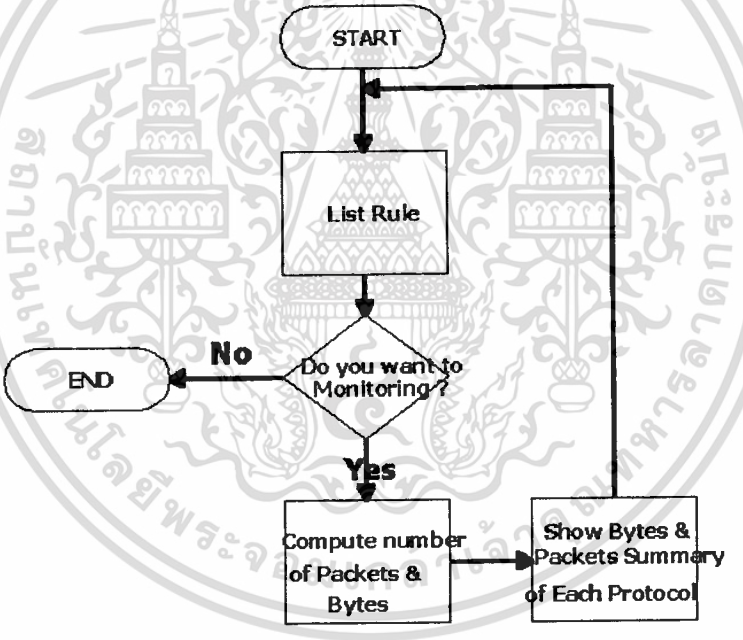
ภาพที่ 4.7 การทำงานของการตั้งเวลาการกรอง

4.3.2.2 การตรวจสอบ (Monitoring)

การตรวจสอบจำนวนแพ็คเก็ตที่ผ่านเข้ามา การจัดเก็บจำนวนแพ็คเก็ตจะใช้คำสั่ง system ("ipfw –a list") ซึ่ง –a จะมีการแสดงค่าแพ็คเก็ตที่เข้าคู่กับกฎ ตามโปรโตคอลที่ระบุ แล้วนำค่าดังกล่าวไปเก็บไว้ในไฟล์อันหนึ่ง (อาจเรียกว่าเป็นล็อคไฟล์ก็ได้) ไฟล์ดังกล่าวประกอบด้วยฟิลด์ดังนี้

คือ เลขที่กฎ จำนวนแพ็คเก็ต จำนวนไบต์ การกระทำของกฎ โปรโตคอล แอดเดรส/พอร์ตต้นทาง แอดเดรส/พอร์ตปลายทาง และตัวเลือกอื่นๆ หลังจากนั้นก็จะทำการเลือกฟิลด์ที่ต้องการ ได้แก่ ฟิลด์จำนวนแพ็คเก็ต และฟิลด์โปรโตคอล โดยใช้คำสั่ง "awk" ในยูนิกต์ และจะมีการเรียกใช้คำสั่งดังนี้คือ system("awk –f sumpacket.awk file(log file)") โดยที่ไฟล์ sumpacket.awk ทำหน้าที่ในการเลือกฟิลด์ที่เป็นไอพีแพ็คเก็ต ทีซีพีแพ็คเก็ต และยูดีพีแพ็คเก็ต แล้วทำการรวมค่าแพ็คเก็ตของแต่ละโปรโตคอล

ฟังก์ชันของไอพีไฟร์วอลล์จะเป็นการทำการตรวจสอบจากการจราจรของเครือข่าย โปรโตคอลที่จะแสดงได้แก่ โปรโตคอลไอพี ทีซีพี และยูดีพี และการแสดงจำนวนแพ็คเก็ตเหล่านี้จะแสดงออกมาโดยจะนำเสนอโดยใช้ตาราง

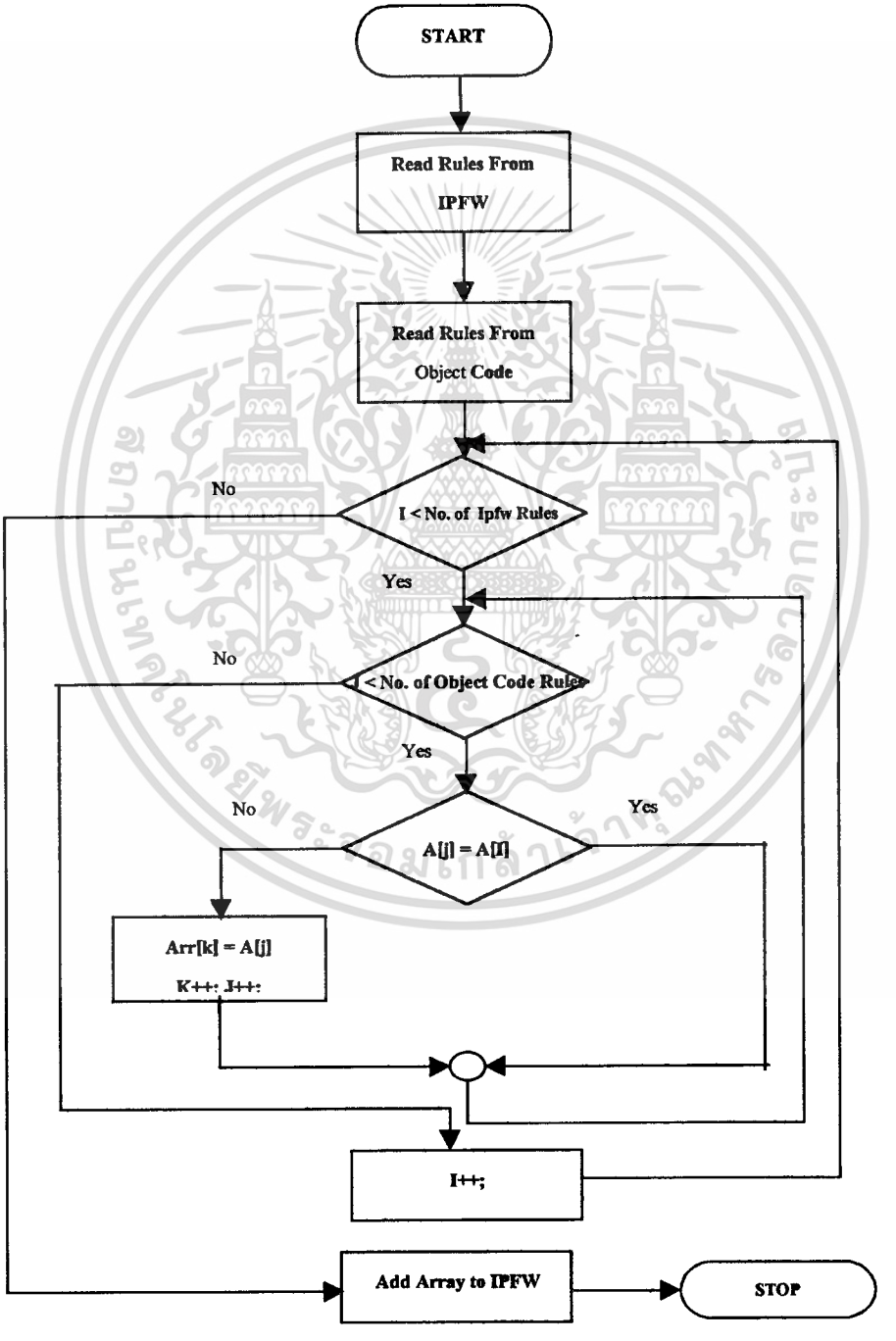


ภาพที่ 4.8 การทำงานของการตรวจสอบจำนวนแพ็คเก็ต

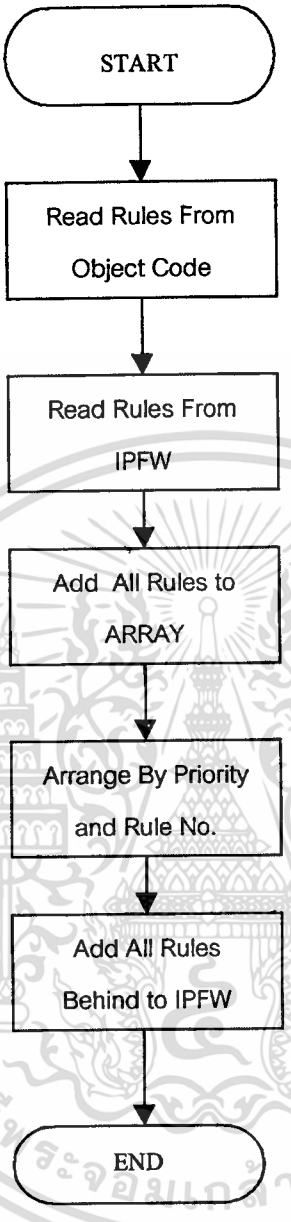
4.3.2.3 การวิเคราะห์กฎขั้นพื้นฐาน (Basic rule analysis)

เพื่อไม่ให้กฎมีการซ้ำกัน เนื่องจากโดยปกติการเพิ่มกฎเข้าไปมักจะเพิ่มเข้าไปเรื่อยๆ ทำให้สิ้นเปลืองเนื้อที่สำหรับเก็บไฟล์กฎ การป้องกันการซ้ำกันของกฎจะกระทำในช่วงของการสร้างกฎและเพิ่มกฎ ถ้ากฎที่ถูกสร้างขึ้นมาใหม่เกิดซ้ำกับกฎที่มีอยู่เดิม กฎนั้นก็จะไม่ถูกเพิ่มเข้าไป ดูในภาพที่ 4.9 ในการพิจารณาว่ากฎมีคาวซ้ำกันเราจะมีการเก็บข้อมูลในส่วนของ Action, Source, Destination, Port เพื่อมาพิจารณาว่ามีการซ้ำกันของกฎรึเปล่า ส่วนในการพิจารณาว่ามีการขัดแย้ง

กันของกฎเราก็จะใช้ข้อมูลชุดเดียวกันมาพิจารณาแต่จะต่างกันที่ถ้า Source, Destination, Port เหมือนกันแต่ Action ต่างกันก็แสดงว่าเกิดความขัดแย้งกันระบบจะอาศัยข้อมูลที่ผู้ใช้ป้อนให้ ในที่นี้คือ การกำหนดลำดับความสำคัญ(Priority) เพื่อมาจัดลำดับกฎในการที่จะใส่เข้าไปทั้งในการทำงาน ณ. ขณะนั้นหรือใส่ลงในไฟล์โครงร่างของไอพีไฟร์วอลล์ซึ่งแล้วแต่ความต้องการของผู้ใช้ ดังภาพที่ 4.10



ภาพที่ 4.9 การทำงานของการตรวจสอบความซ้ำซ้อนของกฎ



ภาพที่ 4.10 การทำงานของการตรวจสอบความขัดแย้งของกฎ

บทที่ 5

การสรุปผลการการทำงานของระบบงาน

ในบทนี้จะมีการสร้างสถานการณ์ เพื่อมาใช้ทดสอบการทำงานของโปรแกรมจัดการไอพีไฟร์วอลล์ โดยจะพิจารณาตามฟังก์ชันการทำงานต่างๆ ที่ระบบรองรับ แล้วสรุปผลออกมา

5.1 การทดสอบระบบงาน

การทดสอบระบบงาน กระทำโดยให้ผู้ใช้กรองนโยบายตามรูปแบบของพอนเดอร์ จากนั้นจะทำการคอมไพล์ เพื่อตรวจเช็ครูปแบบ (syntax) ของภาษาพอนเดอร์ เมื่อทำการตรวจสอบเสร็จแล้ว ระบบจะทำการตีความ โดยเราจะพิจารณาควบคู่กันทั้งการทำ policy management และ การ scheduling โดยเราจะตรวจผลลัพธ์ที่ได้จากการทำคำสั่งโดยตรง (command line) ในกรณีของการทำ scheduling เมื่อระบบได้ทำตามคำสั่งที่ผู้ใช้ได้ระบุลงไปแล้ว ระบบเองจะส่งการติดต่อกลับมายังรูท (Root) เพื่อบอกว่าระบบได้ทำงานตามขั้นตอนเสร็จสิ้นหมดแล้ว โดยผ่านทางระบบอิเล็กทรอนิกส์เมล์ (Electronic mail system)

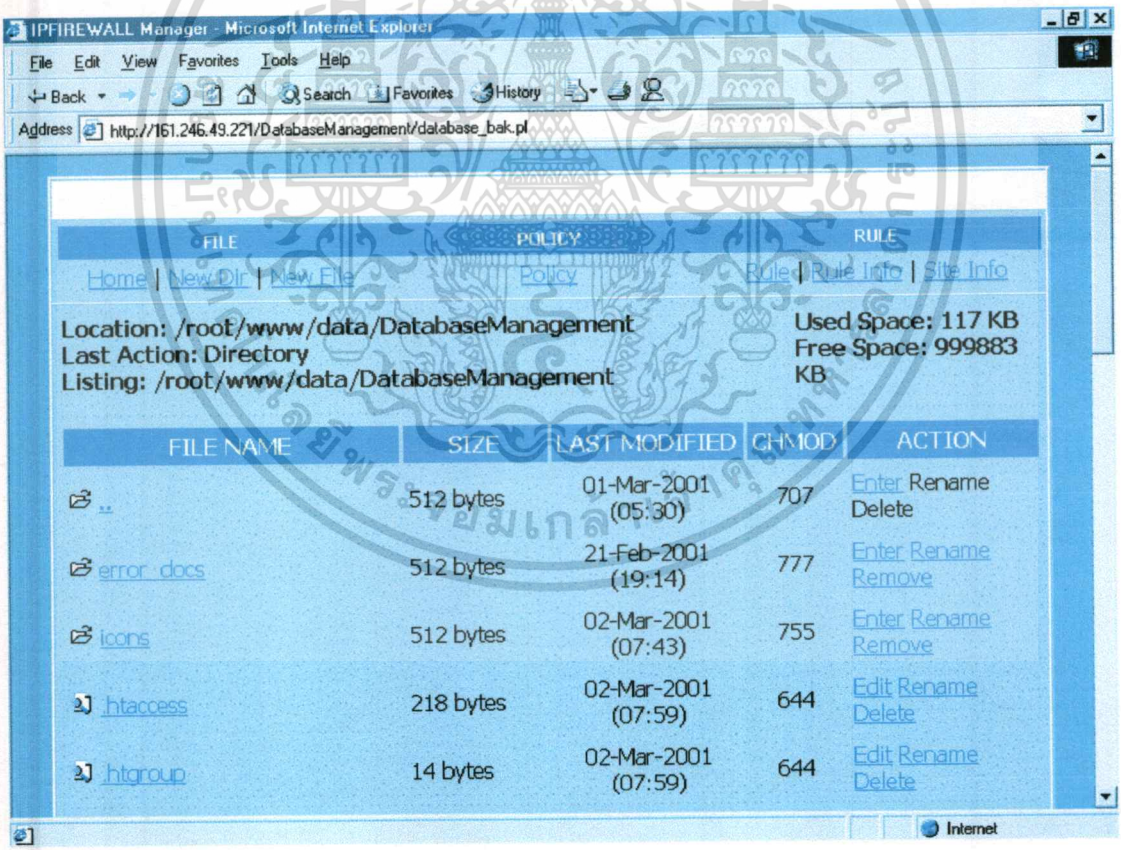
ตัวอย่างของกรณีที่ 1

1) เมื่อต้องการสร้างไฟล์นโยบายใหม่ขึ้นมา ให้เลือกที่ File
2) จากนั้นทำการเลือกที่เมนู New Fileแล้วตั้งชื่อนโยบายตามที่ต้องการ
3) เลือกคำสั่ง Ok โดยจะได้ชื่อไฟล์ว่า Policyrule.pol
4) เลือก Edit เพื่อสร้างนโยบายตามที่แสดลไว้ด้านล่างนี้
5) ไปเลือกที่เมนู Policy เพื่อทำงานในขั้นต่อไป
6) เลือกเช็คเพื่อเลือกไฟล์ที่ต้องการกระทำในที่นี้จะเลือกไฟล์ Policyrule.pol
7) เลือกคำสั่ง Check Syntax เพื่อตรวจเช็คความถูกต้อง
8) เลือกคำสั่ง Compile & Run
9) หลังจากนั้นให้เลือกคำสั่ง List rule เพื่อแสดงกฎ
10) เมื่อต้องการตรวจสอบจำนวนแพ็คเก็ตที่เข้ามาให้เลือกคำสั่ง Monitoring

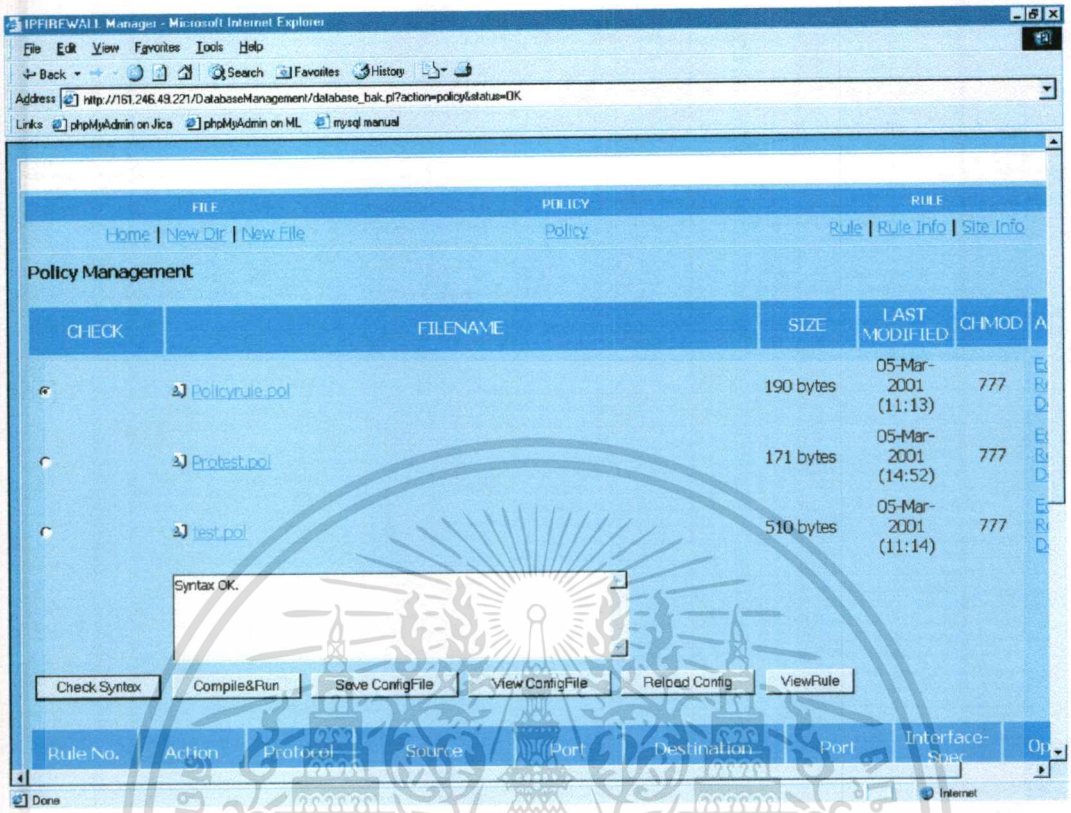
ตัวอย่างนโยบายที่ 1 Rolicyrule.pol

```
int service = 80;
string protocol = "udp";
int priority = 10;
auth+  test{
    subject 161.246.49.21;
     target 161.246.49.221;
   action allow(protocol,priority,service);
}
```

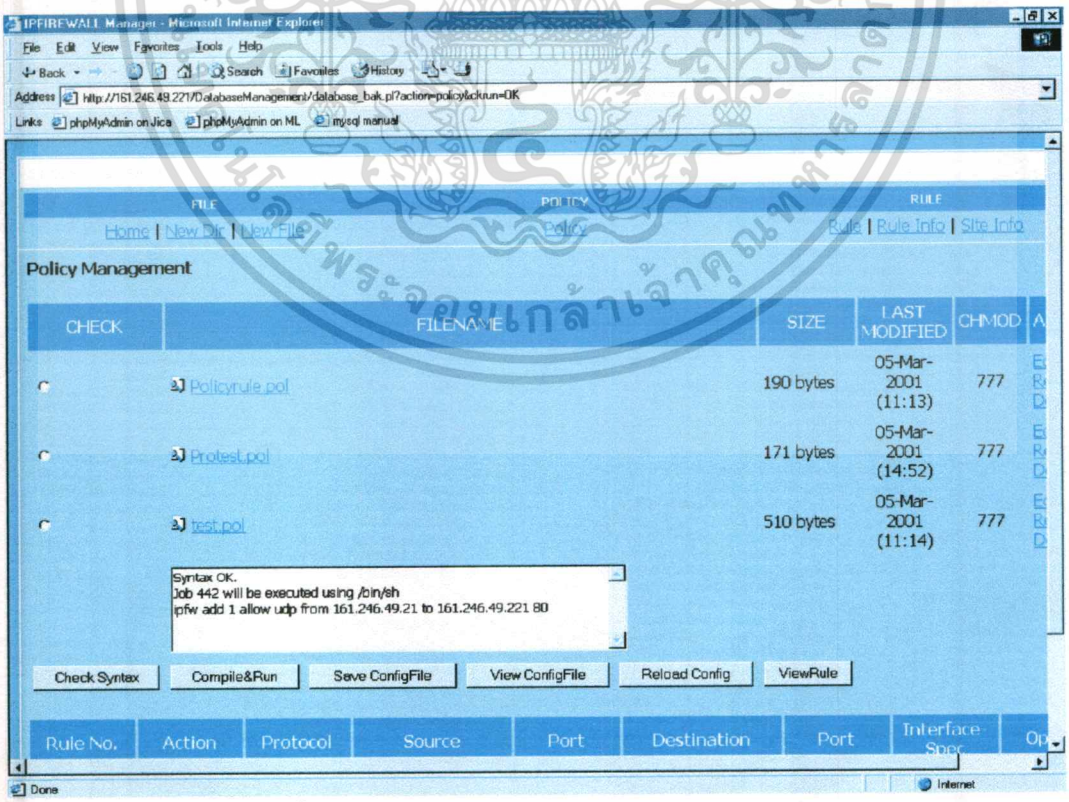
แสดงขั้นตอนการทำงานดังนี้



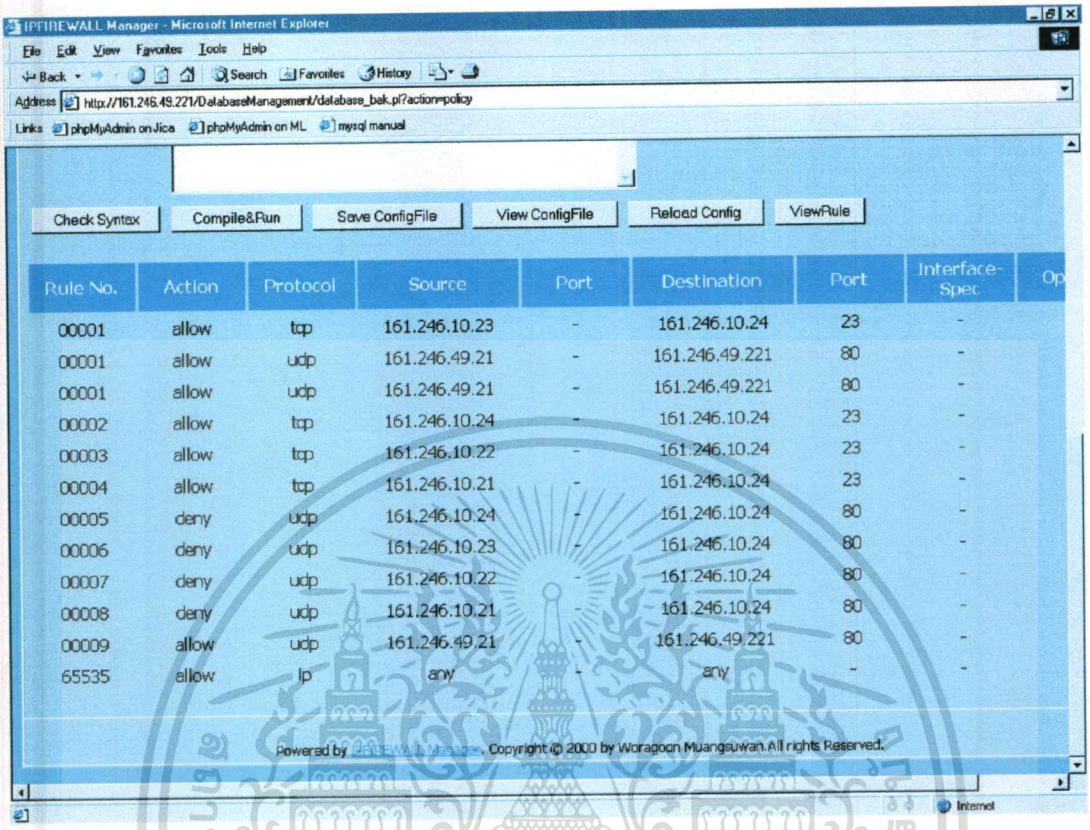
ภาพที่ 5.1 แสดงหน้าต่างแรกของระบบ



ภาพที่ 5.2 แสดงการทำงานของการตรวจสอบความถูกต้องของนโยบาย



ภาพที่ 5.3 แสดงการทำงานของ Compile และ Run



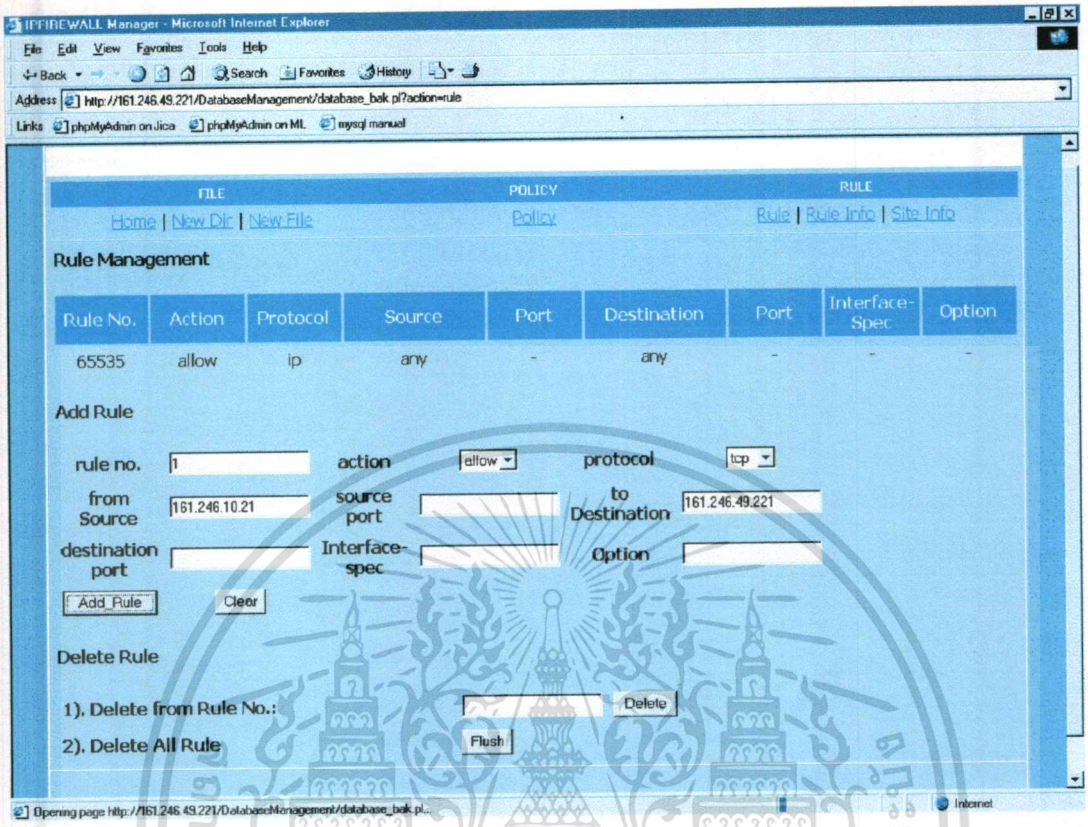
ภาพที่ 5.4 แสดงผลลัพธ์ที่ได้จากการ Run นโยบายที่เราได้กำหนดขึ้นมา

ตัวอย่างที่ 2

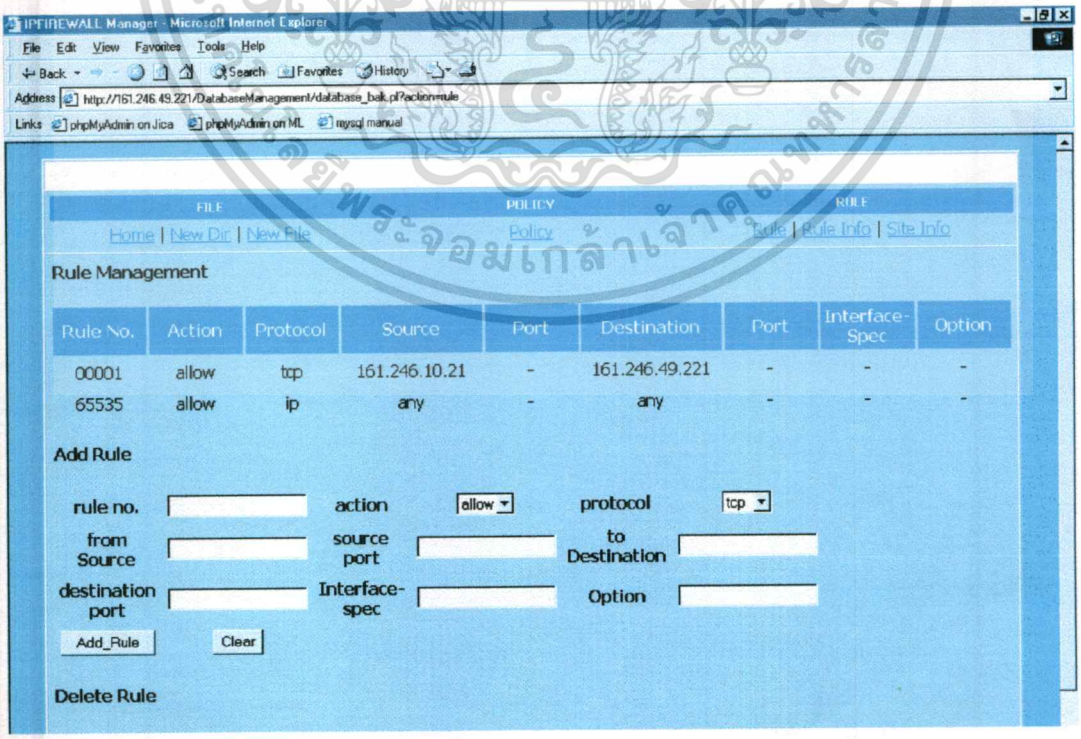
จะเป็นการสร้างกฎขึ้นมาทีละกฎ โดยผู้ใช้สามารถกำหนดรายละเอียดของกฎได้

1) **เลือกเมนู Rule**
2) **ใส่รายละเอียดของกฎในแต่ละช่องตามข้อกำหนดของการสร้างกฎดังที่ได้อธิบายในบทก่อนหน้า**
3) **เลือกคำสั่ง Add หรือ Clear เมื่อไม่พอใจ**

จากนั้นกฎที่ผู้ใช้สร้างขึ้นจะถูกส่งต่อไปยังคำสั่ง ipfw add (ตามด้วยรายละเอียดของกฎที่สร้างขึ้น) ซึ่งจะมีอยู่ในระบบปฏิบัติการฟรีบีเอสดีแล้ว



ภาพที่ 5.5 แสดงการเพิ่มกฎ



ภาพที่ 5.6 แสดงผลลัพธ์ที่ได้จากการเพิ่มกฎที่เราได้กำหนดขึ้นมา

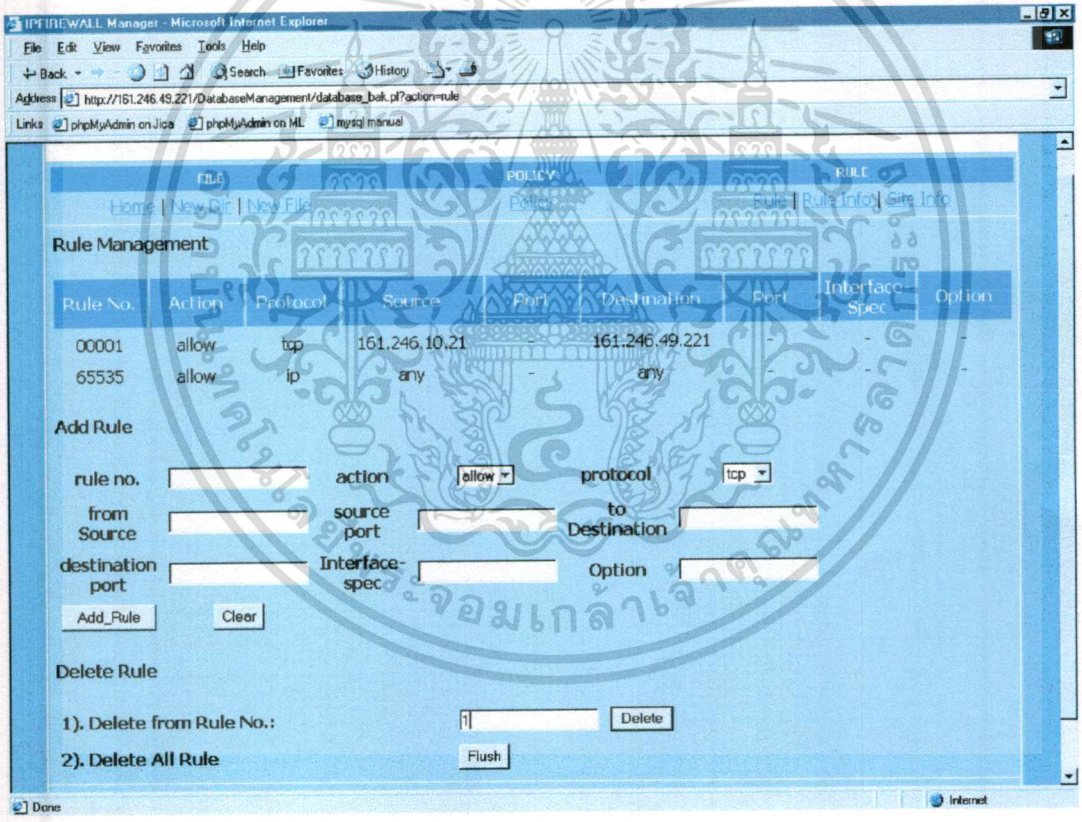
ตัวอย่างที่ 3

เป็นการลบกฎที่มีอยู่ในระบบโดยจะให้ผู้ใช้กรอกหมายเลขกฎที่ต้องการจะลบ

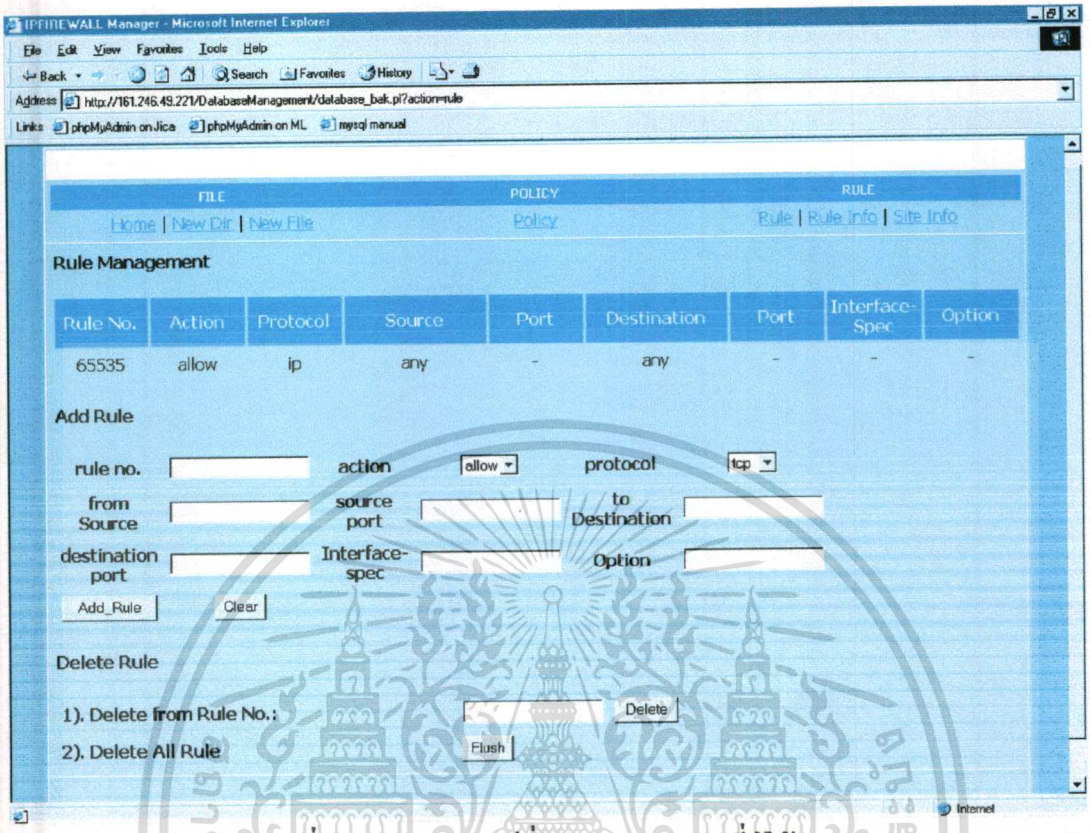
1) เลือกเมนู Rule

2) ใส่หมายเลขของกฎที่ต้องการลบ

จากนั้นระบบจะรับเอาหมายเลขที่ผู้ใช้ระบุไปผ่านคำสั่ง ipfw delete [หมายเลขกฎที่ต้องการลบ]



ภาพที่ 5.7 แสดงการลบกฎโดยให้ผู้ใช้ระบุหมายเลข



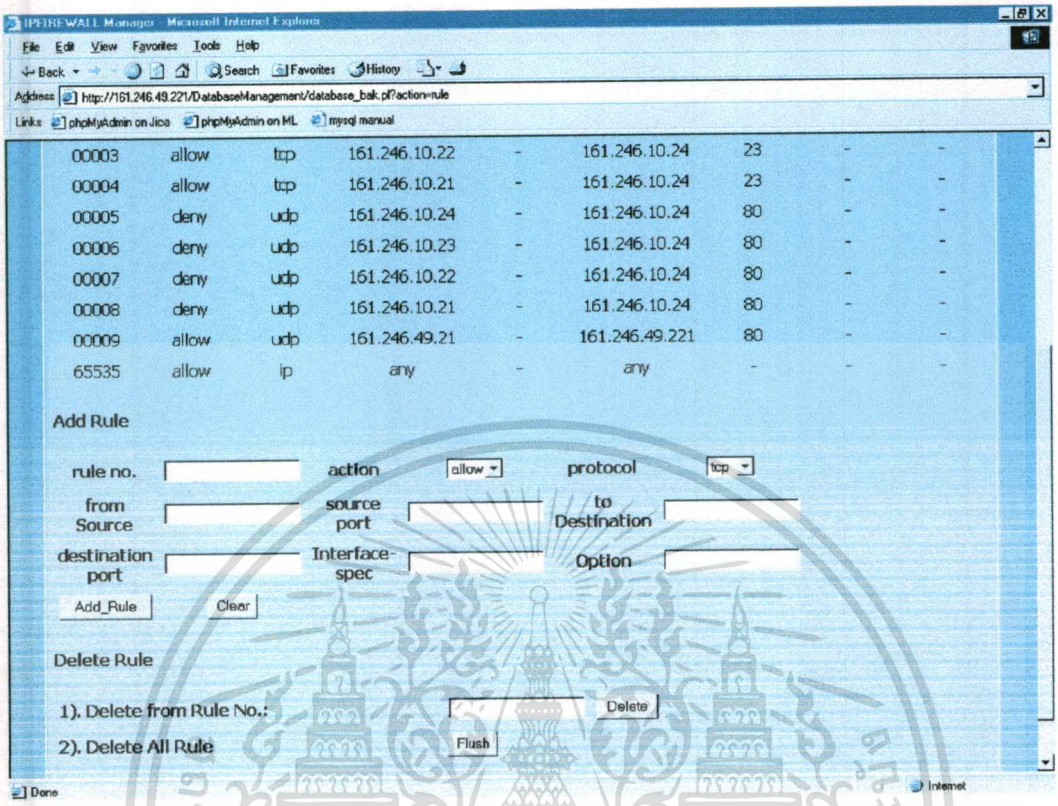
ภาพที่ 5.8 แสดงผลลัพธ์ที่ได้จากการลบกฎที่ผู้ใช้ระบุ

ตัวอย่างที่ 4

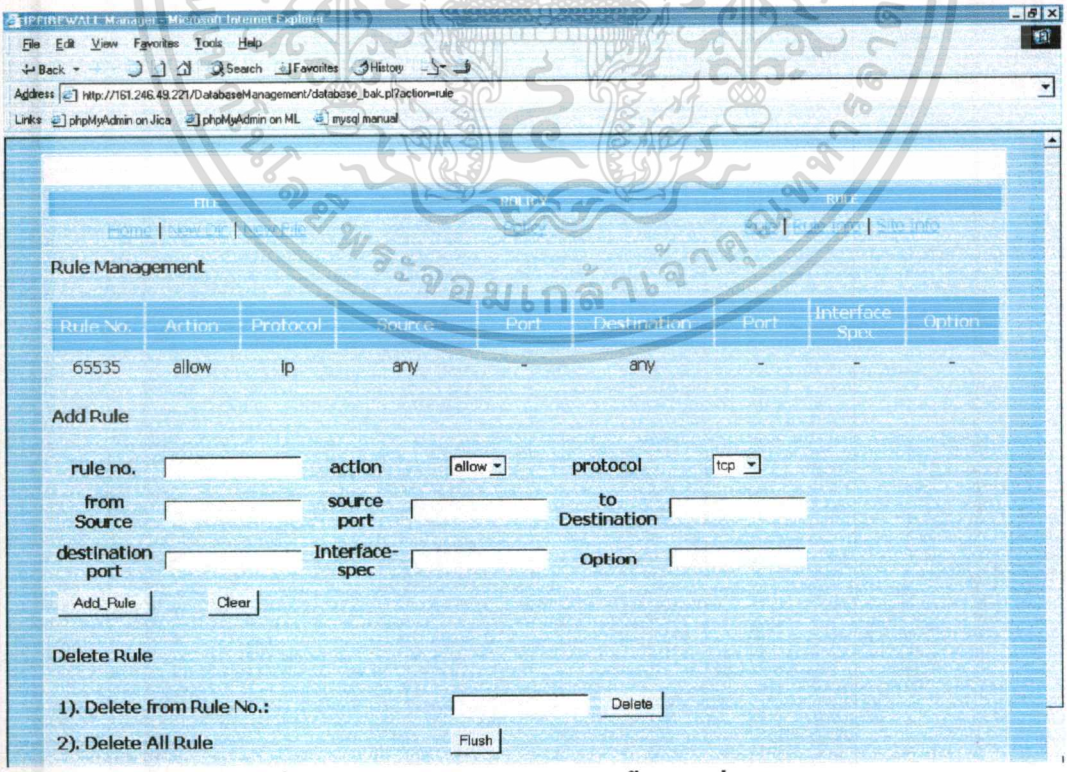
เป็นการลบกฎทั้งหมดที่มีอยู่ในระบบ(ยกเว้นกฎ Default : หมายเลข 65535)โดยจะให้ผู้ใช้กรอกหมายเลขกฎที่ต้องการจะลบ

1) เลือกเมนู Rule
2) เลือกที่ปุ่ม Flush

จากนั้นระบบจะใช้คำสั่ง ipfw –q flush ซึ่งจะทำการลบกฎทั้งหนดในระบบโดยจะไม่ถามการยืนยันของผู้ใช้ (เพราะใช้ –q)



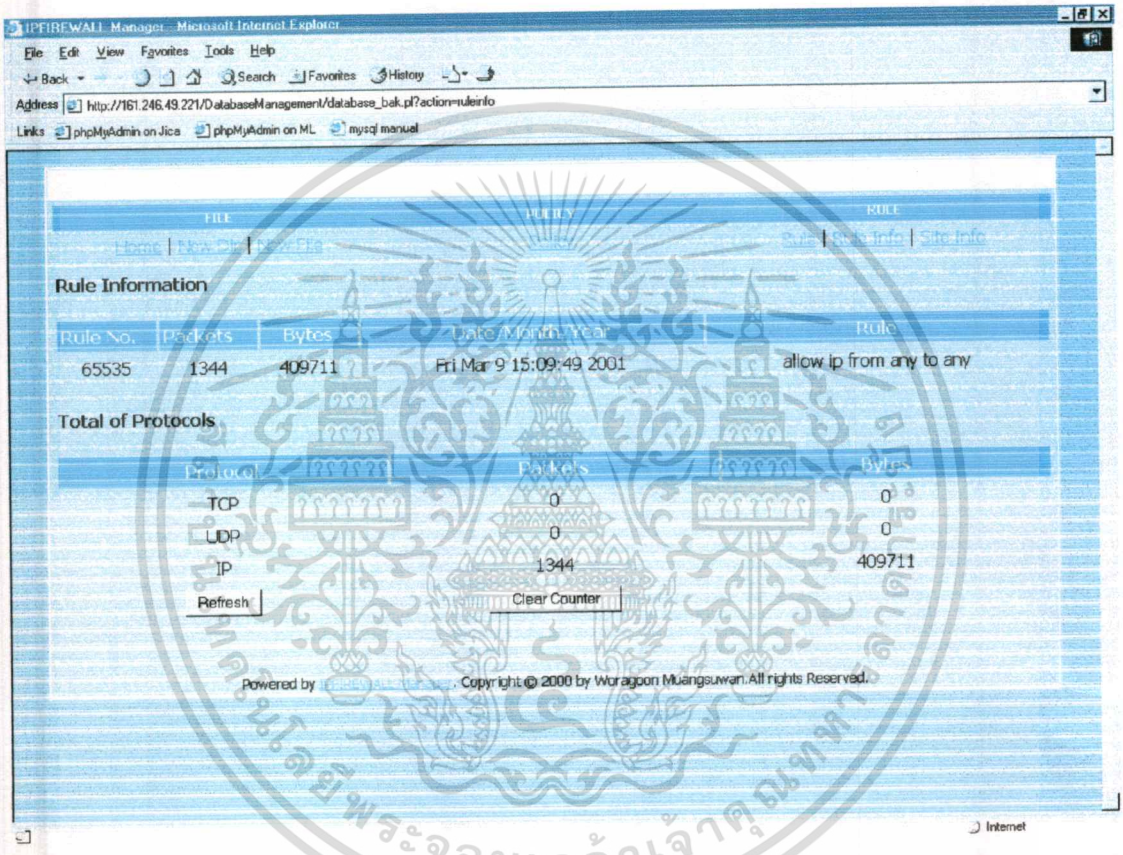
ภาพที่ 5.9 แสดงการลบกฎทั้งหมดที่อยู่ในระบบ



ภาพที่ 5.10 แสดงผลลัพธ์การลบกฎทั้งหมดที่อยู่ในระบบ

ตัวอย่างที่ 5

เป็นการแสดงผลลัพธ์ของจำนวนไบต์และจำนวนแพ็คเก็ตในแต่ละกฎที่แพ็คเก็ตผ่านระบบเข้ามานั้นมีรายละเอียดตรงตามกับกฎที่ได้ออกแบบไว้ซึ่งจะแสดงเป็นผลรวมในแต่ละกฎและผลรวมของแต่ละโปรโตคอล โดยที่ในการแสดงในแต่ละกฎที่เข้าคู่นั้นเราจะแสดงรายละเอียดโดยใช้คำสั่ง ipfw –a list เป็นการแสดงรายละเอียดที่กล่าวมาแล้าข้างบน



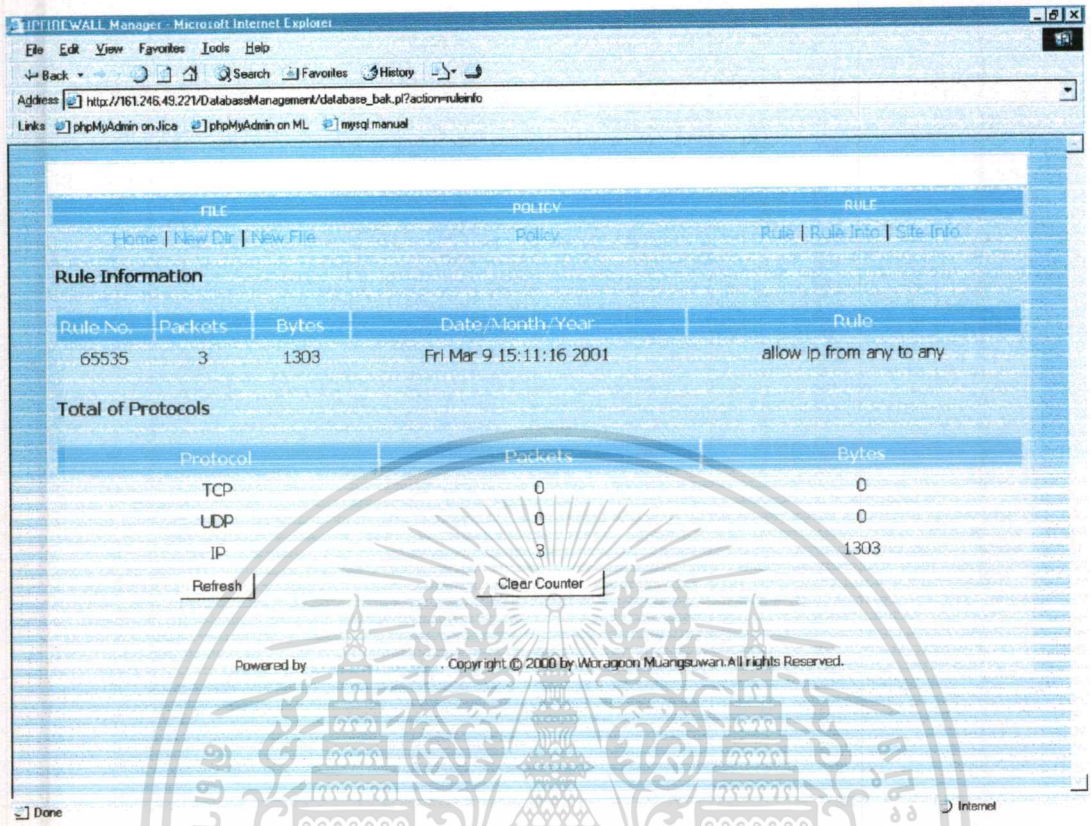
ภาพที่ 5.11 แสดงจำนวนแพ็คเก็ตและจำนวนไบต์ที่เข้าคู่กับกฎและผลรวมในแต่ละโปรโตคอล

ตัวอย่างที่ 5

เป็นการทำการรีเซ็ตค่าในตัวนับต่างๆทั้ง ตัวนับแพ็คเก็ตและตัวนับไบต์ให้มีค่าเป็นศูนย์ โดยทำตามวิธีการดังนี้

1) เลือกที่เมนู Rule info
2) เลือกที่ปุ่ม Clear

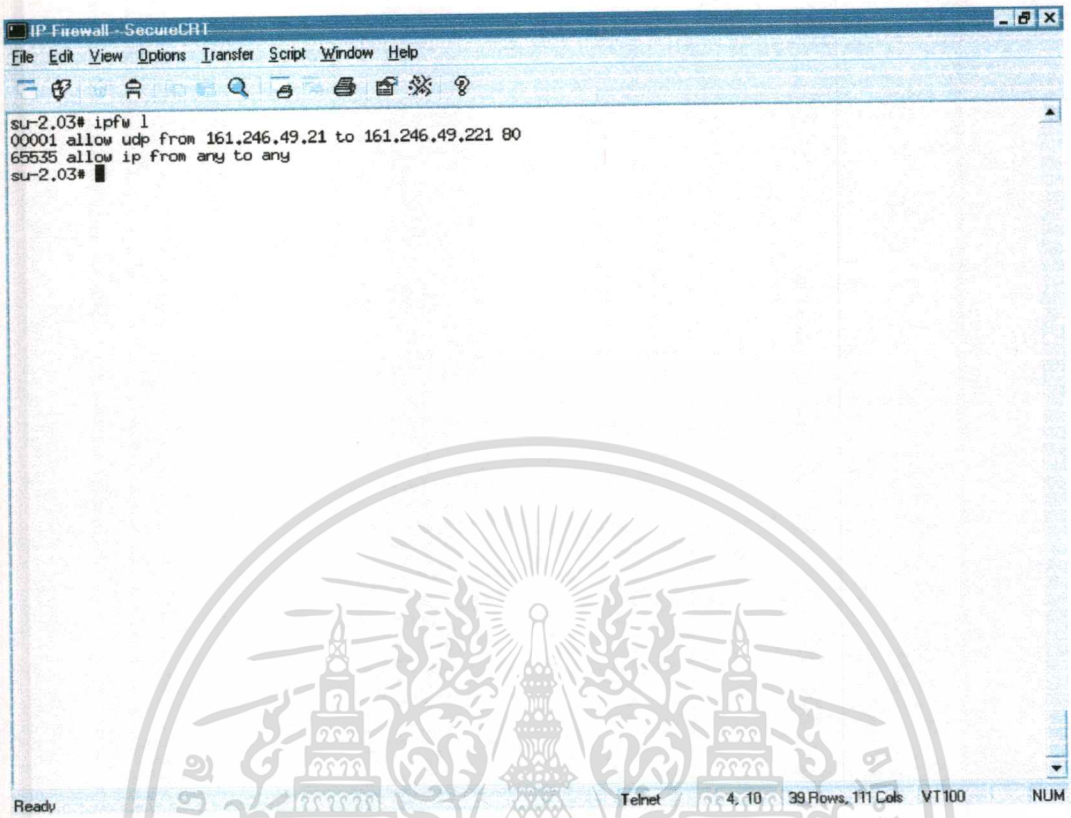
หลังจากนั้นระบบจะรับคำสั่งไปทำงานโดยใช้คำสั่งดังนี้คือ ipfw zero เป็นการลบกฎออกทั้งหนดโดยจะไม่ถามการยืนยันต่อผู้ใช้



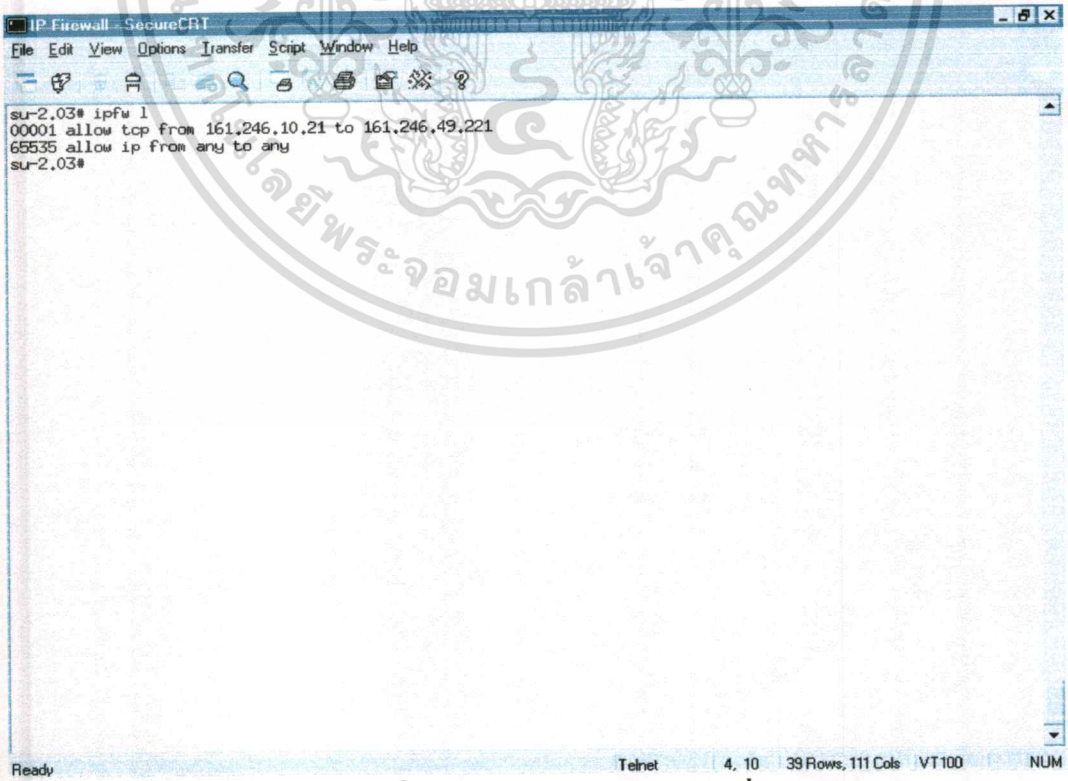
ภาพที่ 5.12 แสดงการลบค่าเคาท์เตอร์ทั้งหมดของระบบ

5.2 ผลสรุปที่ได้จากการทดลอง

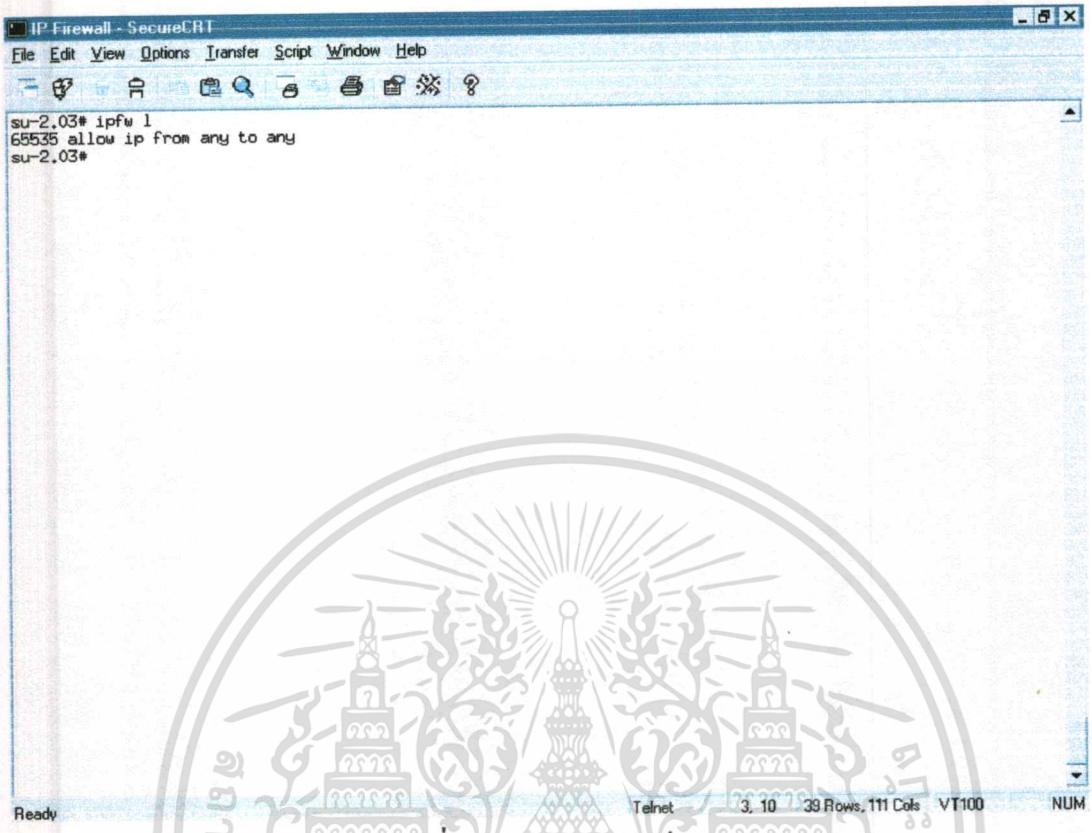
จากตัวอย่างที่แสดงมาทั้งหมดนอกจากผู้ใช้สามารถเห็นถึงผลลัพธ์การทำงานจากโปรแกรมแล้วผู้ใช้ยังสามารถเข้าไปตรวจสอบความถูกต้องของการทำงานทาง Command Line (ซึ่งผู้ใช้จะต้องเป็นผู้จัดการระบบ :Administration) ซึ่งในส่วนต่อไปจะแสดงผลลัพธ์ทาง Command Line ในทุกตัวอย่างและผลลัพธ์ที่แจ้งกลับมาเมื่อผู้ใช้ได้กำหนดเวลาในส่วน Scheduling (when tag) ในรูปของจดหมายอิเล็กทรอนิก (E-mail) โดยใช้ผ่านคำสั่ง at ของยูนิกซ์



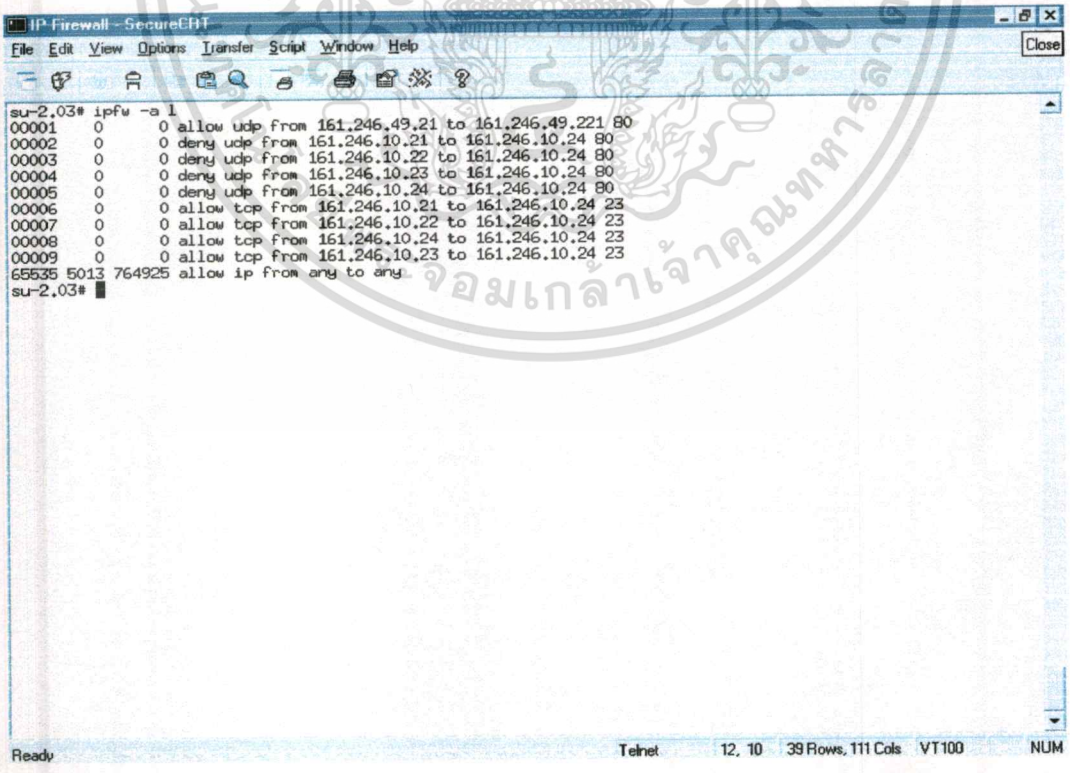
ภาพที่ 5.13 แสดงผลลัพธ์ของการแปลจากนโยบาย Policyrule.pol มาเป็นกฎ



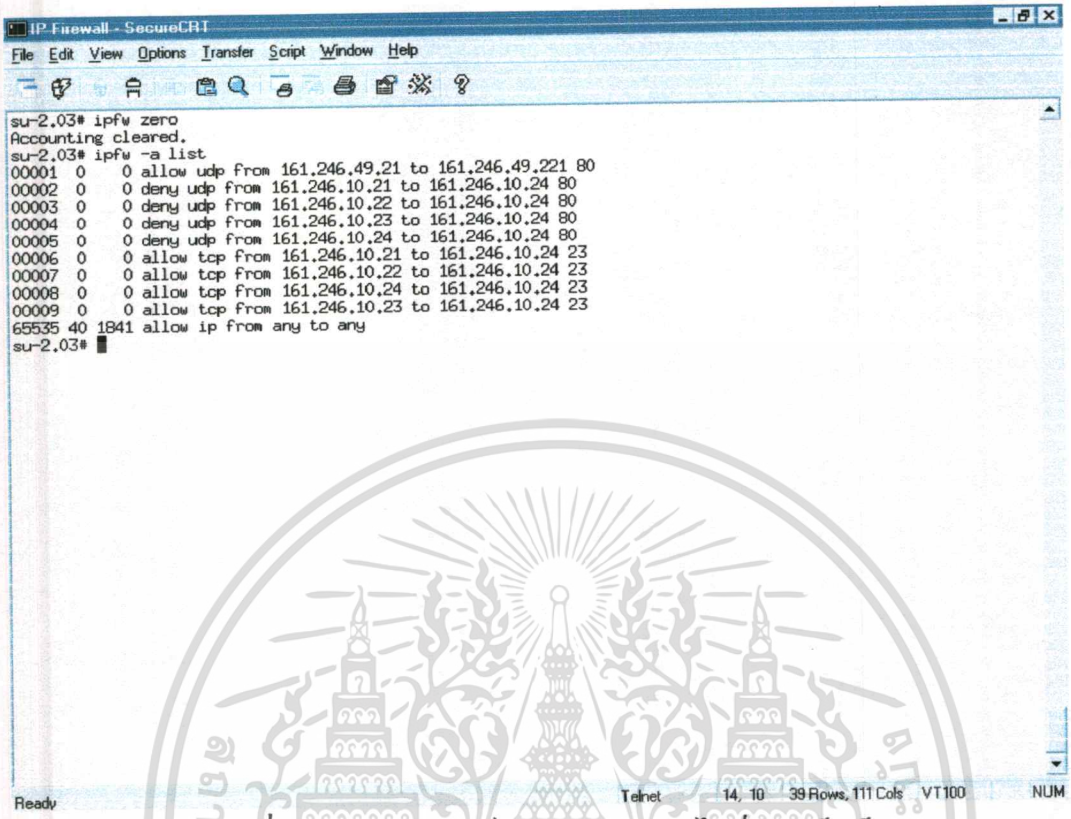
ภาพที่ 5.14 แสดงผลลัพธ์ของการเพิ่มกฎ



ภาพที่ 5.15 แสดงผลลัพธ์ของการฉบกฎ



ภาพที่ 5.16 แสดงผลลัพธ์ของการแสดงตัวนับไบต์และแพ็คเก็ต



ภาพที่ 5.17 แสดงผลลัพธ์ของการลบตัวนับไบต์และแพ็คเก็ต

จากรูปที่ 5.17 จะเห็นว่ากฎลำดับที่ 65535 จำนวนแพ็คเก็ตไม่เป็นศูนย์เพราะว่าเมื่อเราทำการลบค่าตัวนับแล้วใช้คำสั่งแสดงจำนวนแพ็คเก็ตนั้นปรากฏว่าเป็นการส่งข้อมูลที่แพ็คเก็ตไปเข้าคู่กับกฎลำดับที่ 65535 พอดีทำให้ไม่เป็นศูนย์

5.3 สรุปผลการพัฒนาโปรแกรม

จากการพัฒนาโปรแกรมทำให้เข้าใจถึงความสามารถของฟังก์ชันไอพีไฟร์วอลล์และการนำฟังก์ชันการทงานนี้มาใช้ประโยชน์ รวมถึงได้มีการพัฒนาฟังก์ชันการทำงานเพิ่มเติมแก่ไอพีไฟร์วอลล์ด้วย

ผลที่ได้จากการทดลองทั้ง 5 กรณีดังกล่าว เป็นเครื่องพิสูจน์ได้ว่าการทำงานของโปรแกรมสามารถควบคุมการทำงานของไอพีไฟร์วอลล์ได้จริง เช่นการเพิ่มกฎ การแสดงรายการของกฎ การลบกฎ การลบค่าตัวนับแพ็คเก็ต การตั้งเวลาการกรอง ณ เวลาที่เรากำหนดให้ทำงาน การตรวจสอบจำนวนแพ็คเก็ต การวิเคราะห์กฎขั้นพื้นฐานเพื่อไม่ให้กฎที่เพิ่มเข้ามาซ้ำกับกฎที่มีอยู่ และการสร้างนโยบาย

ความสามารถของระบบในการทำการตีความภาษา

- สามารถหาข้อผิดพลาดของรูปแบบภาษาได้ในระดับหนึ่ง

- สามารถบอกถึงการใช้ตัวแปรที่ผิดประเภทและการประกาศตัวแปรที่ซ้ำกัน
- มีฟังก์ชันการทำงานภายใน คือ allow และ deny
- สามารถทำ domain scope expression ในการทำ policy management
- สามารถทำตามฟังก์ชันการทำงานของ scheduling ได้
- สามารถทำตามฟังก์ชันการทำงานที่มีอยู่เดิมจากโครงงานเดิมได้ทั้งหมด

จะเห็นได้ว่าขีดความสามารถของระบบในการจัดการรูปแบบภาษา มีดังนี้

- ในการใช้เงื่อนไข (if) สามารถเปรียบเทียบความถูกต้องที่มีความซับซ้อนได้ไม่มาก เช่น สามารถเปรียบเทียบโดยใช้ and or และ xor ได้ แต่ไม่สามารถทำการเปรียบเทียบฟังก์ชันการทำงานได้
- ในส่วนของการ Assign ค่า สามารถทำความเข้าใจได้เพียง +, -, * และ / ไม่สามารถทำการยกกำลังหรือให้ค่าโดยได้ค่ามาจากฟังก์ชันได้
- สามารถรองรับรูปแบบของภาษาตามลำดับของ Keyword ดังที่ได้แสดงผ่านมา ไม่สามารถสลับ Keyword ได้
- ในการทำ expression เพื่อ assign ทำได้ในระดับไม่ซับซ้อน เช่น การนำค่าของ function มา assign ให้กับตัวแปรไม่ได้
- รองรับ action คือ allow และ deny มีฟังก์ชัน time เป็นฟังก์ชันเพียงอันเดียว
- ในการเพิ่มกฎไม่มีการตรวจสอบค่า ipaddress เมื่อใส่ค่าที่ผิดระบบจะไม่เตือน

จากการทดลองใช้งานของโปรแกรมจะเห็นได้ว่ามันสามารถทำงานได้ตามที่ต้องการ ตามฟังก์ชันการทำงานทั่วไป และฟังก์ชันที่เพิ่มเติมอย่างที่กล่าวไว้ในบทแรกๆ อย่างไรก็ตามโปรแกรมนี้สามารถนำไปพัฒนาต่อได้เพื่อให้ได้โปรแกรมที่สมบูรณ์มากขึ้นและเกิดประโยชน์มากที่สุด

5.4 ข้อเสนอแนะ

การพัฒนาโปรแกรมจัดการไอพีไฟร์วอลล์สามารถที่จะพัฒนาให้มีความสามารถมากยิ่งขึ้นไปได้ โดยการเพิ่มฟังก์ชันการทำงานต่างๆ ให้มีความสะดวกมากยิ่งขึ้น เช่น

1. ในการเช็คความถูกต้องของนโยบาย เมื่อมีความผิดพลาด สามารถที่จะบอกความผิดพลาดให้มีความละเอียดมากกว่าที่เป็นอยู่
2. สามารถนำผลที่ได้จากการ monitoring มาทำการวิเคราะห์ได้ตามที่ผู้ใช้ต้องการได้มากขึ้น

3. มีการกำหนดความสามารถในการสร้างนโยบาย (syntax) ที่มีความยืดหยุ่นและมีความสามารถมากขึ้น เช่น ความสามารถในการพิจารณาเงื่อนไข (if) สามารถมีความซับซ้อนมากขึ้น
4. สร้างฟังก์ชันเพิ่มเติมใน actionlist ได้มากขึ้น
5. สามารถทำ operation ของ Assign Expression ได้มากขึ้น
6. สามารถรองรับรูปแบบของภาษาเชิงวัตถุได้ (object-oriented)
7. ขยายขีดความสามารถของการกำหนดและตีความของภาษาให้มากขึ้น
8. สามารถใช้ประสิทธิภาพของไอพีไฟร์วอลล์ได้มากกว่านี้ เช่น กำหนด Bandwidth

บรรณานุกรม

Chris Hare and Karanjit Siyan. 1996. "Internet Firewalls and Network Security". 2nd Edition. Indianapolis. :New Riders Publishing.

D.Brent Chapman and Elizabeth D. Zwichy. 1995. "Buliding Internet Firewalls".Sebastopol, CA. : O'Reilly & Associate, Inc.

Gray Palmer and Alex Nash. August 2000. "Chapter 8 Security – Firewall – FreeBSD HandBook.".[Online]. Available : http://www.freebsd.org/handbook/index.html

Guy Helmer. August 2000."Security Tool in FreeBSD".[Online]. Available : http://www.freebsd.org/security/.

Marcus Gonclaves. 1998. "Firewalls Complete", New York. : McGrow-Hill.

Nicodemos Damianou, Naranker Dulay, Emil Lupu, Morris Sloman. August 2000. "The Ponder Policy Specification Language". [Online]. Available : http://www-dse.doc.ic.ac.uk/policies/.

Nicodemos Damianou, Naranker Dulay, Emil Lupu, Morris Sloman. 2000/1."Ponder: A Language for Specifying Security and Management Policies for Distributed Systems ". The Language Specification Version 2.2, Imperial College Research Report DoC.

ภาคผนวก

ก. โครงสร้างของภาษาในการสร้างนโยบาย

Declaration Type *variable* = *value*;
(auth+ | auth-) *Namepolicy* {
subject *Domain Scope Expression*;
target *Domain Scope Expression*;
action *action list* [*if-clause*]; [when *function time*];
}

1) ชนิดของตัวแปร (Declaration Type) ภาษาที่ได้ออกแบบสามารถรับชนิดข้อมูลได้ดังต่อไปนี้
   - เลขจำนวนเต็ม (Integer : int)
   - ตัวอักษร (Character : char)
   - สายของอักขระ (String : string)
   - บูลีน (Boolean : boolean)
   - โดเมน (Domain : domain)
2) นิพจน์ของโดเมน (Domain Scope Expression) นิพจน์ของโดเมนถูกออกแบบให้สามารถทำงานกับตัวกระทำได้ดังต่อไปนี้
   - ตัวกระทำยูเนียน (Union : +)
   - ตัวกระทำลบ (Differential : -)
   - ตัวกระทำอินเตอร์เซ็ค (Intersection : ^)
3) นิพจน์ (Expression) นิพจน์ในภาษาที่ออกแบบรองรับการทำงานกับตัวกระทำดังต่อไปนี้
   - ตัวกระทำบวก (+)
   - ตัวกระทำลบ (-)
   - ตัวกระทำคูณ (*)
   - ตัวกระทำหาร (/)

4) รายการของการกระทำ (action list) ยอมรับฟังก์ชัน allow และ deny โดยมีรูปแบบดังนี้

allow(*protocol, priority, service*)

deny(*protocol, priority, service*)

- ฟังก์ชัน allow หมายถึง อนุญาตให้ทำการกระทำที่กำหนด
- ฟังก์ชัน deny หมายถึง ไม่อนุญาตให้ทำการกระทำที่กำหนด
- *protocol* หมายถึง โปรโตคอล เป็นตัวแปรที่มีชนิดเป็นสายของอักขระ (String)
- *priority* หมายถึง ลำดับความสำคัญของการกระทำ เป็นตัวแปรที่มีชนิดเป็นตัวเลขจำนวนเต็ม (Integer)
- *service* หมายถึง บริการของ destinationที่ source จะเข้าไปใช้ เป็นตัวแปรที่มีชนิดเป็นตัวเลขจำนวนเต็ม (Integer)

5) ส่วนของเงื่อนไข (if-clause) ภาษานี้สามารถสร้างโครงสร้างเงื่อนไขได้ โดยมีรูปแบบดังนี้

if (*condition*) {
  (in|out) *Expression*;
}

- เงื่อนไข (condition) สามารถทำงานกับตัวกระทำ <, >, <=, >=, ==, <>, and, or และ xor ได้และสามารถเปรียบเทียบค่าที่ได้รับมาจากฟังก์ชัน time ได้
- in เป็นการประกาศว่าในการกำหนดค่าให้ตัวแปร ค่าที่กำหนดให้จะมีผลกับนโยบายที่ได้ทำการกำหนดให้เท่านั้น (Local)
- out เป็นการประกาศว่าในการกำหนดค่าให้ตัวแปร ค่าที่กำหนดให้จะมีผลกับนโยบายทั้งหมดที่อยู่ในแฟ้มข้อมูลเดียวกัน (Global)

6) รูปแบบของฟังก์ชัน time (function time) ฟังก์ชัน time เป็นฟังก์ชันที่กำหนดขึ้นเพื่อทำงานเกี่ยวกับเวลา สามารถใช้ได้ 2 รูปแบบ คือ

- time() หมายถึง การดึงเวลาปัจจุบันจากระบบ ซึ่งจะใช้สำหรับการเปรียบเทียบในส่วนของเงื่อนไขของ if-clause

- time(*parameter*) สามารถรับพารามิเตอร์ได้ 2 รูปแบบ คือ

  - วันที่ (date) รับตัวเลข 8 หลัก (ddmmyyyy) เช่น time(05022001)
    - dd หมายถึง วันที่ มีค่าตั้งแต่ 01-31
    - mm หมายถึง เดือน มีค่าตั้งแต่ 01-12
    - yyyy หมายถึงปี มีค่าตั้งแต่ 0000-9999
  - เวลา (time) รับตัวเลข 4 หลัก (hhmm) เช่น time(1530)
    - hh หมายถึงเวลาในหน่วยนาฬิกา มีค่าตั้งแต่ 00-23
    - mm หมายถึงเวลาในหน่วยนาที มีค่าตั้งแต่ 00-59

ข. ความต้องการของระบบ

1) ระบบปฏิบัติการ FreeBSD Version 4.0
2) ต้องมีการเซ็ตโหมดการทำงานของไฟร์วอลล์ในฟรีบีเอสดี สามารถดูได้จากคู่มือฟรีบีเอสดี
3) ตัวแปลภาษา perl
4) Apache Web Server

ค. การใช้งานโปรแกรม

1) การทำงานเกี่ยวกับแฟ้มข้อมูล
   - **Make New Directory** การสร้างไดเร็กทอรี่ใหม่
   - **Make New File** การสร้างแฟ้มข้อมูลใหม่
   - **Upload File** การทำสำเนาแฟ้มข้อมูลข้ามระบบ
2) การทำงานเกี่ยวกับนโยบาย
   - **Check Syntax** ตรวจสอบความถูกต้องของนโยบาย
   - **Compile & Run** การตีความนโยบายและสั่งให้ทำงาน
   - **Save to Config File** การนำกฎที่ได้บันทึกลงบนไฟล์โครงสร้าง (rc.firewall)
   - **View Config File** การนำข้อมูลกฎในไฟล์โครงสร้างมาแสดง
   - **Reload Config File** การรีโหลดไฟล์โครงสร้าง
3) การทำงานเกี่ยวกับกฎ
   - **Add Rule** การเพิ่มกฎ

- **Delete Rule** การลบกฎโดยระบุหมายเลขกฎ
- **Flush Rule** การลบกฎออกทั้งหมด
- **View Rule** การแสดงกฎที่มีอยู่ทั้งหมด

4) การทำงานเกี่ยวกับข้อมูลของกฎ

- **Rule Information** การแสดงจำนวนแพ็คเก็ตและจำนวนไบต์ที่มีแพ็คเก็ตมาเข้าคู่กับกฎ
- **Total of Protocol** การแสดงผลลัพธ์ทั้งหมดของจำนวนแพ็คเก็ตและจำนวนไบต์ในแต่ละโปรโตคอล
- **Refresh** การรีเฟรชข้อมูลของจำนวนแพ็คเก็ตและจำนวนไบต์
- **Clear Counter** การเซ็ตค่าตัวนับจำนวนแพ็คเก็ตและจำนวนไบต์ให้เป็นศูนย์

ประวัติผู้เขียน

ชื่อผู้เขียน	นายวรกุล เมืองสุวรรณ
วันเดือนปีเกิด	5 กุมภาพันธ์ 2520
สถานที่เกิด	จ.ชลบุรี
วุฒิการศึกษาระดับปริญญาตรี	วท.บ. (วิทยาการคอมพิวเตอร์)
สถานที่สำเร็จการศึกษา	คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีที่สำเร็จการศึกษา	ปีการศึกษา 2541