การพัฒนายูสเซอร์อินเตอร์เฟสแบบเว็บ สำหรับไอพีไฟร์วอลล์ ของ FreeBSD

Web-based User Interfacing for IPFIREWALL of FreeBSD

โดย

ธัญลักษณ์ ผังชัยมงคล

รหัสประจำตัว 46066725

อาจารย์ที่ปรึกษา

ผศ.อัครินทร์ คุณกิตติ

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 2 ปีการศึกษา 2548

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อหัวข้อ	การพัฒนายูสเซอร์อินเตอร์เฟสแบบเว็บสำหรับไอพีไฟร์วอลล์ของ FreeBSD
นักศึกษา	นางสาว ธัญลักษณ์ ผังชัยมงคล
อาจารย์ที่ปรึกษา	ผศ. อัครินทร์ คุณกิตติ
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2548

บทคัดย่อ

เนื่องจากการใช้งานไอพีไฟร์วอลล์ของ FreeBSD มีการใช้งานในลักษณะ command line จึงทำให้การใช้งานมีความยุ่งยากซับซ้อนไม่สะดวกต่อการใช้งานและการแก้ไขคำสั่งหรือกฎต่างๆ ดังนั้นโครงการนี้จึงมีวัตถุประสงค์ เพื่อพัฒนาระบบการใช้งานคำสั่งในการสร้างกฎของไฟร์วอลล์ผ่านส่วนของการติดต่อกับผู้ใช้งานแบบเว็บ เพื่อให้เกิดความสะดวกต่อการใช้งาน และศึกษาถึงเทคโนโลยีการใช้ไฟร์วอลล์ IPFW โดยในการพัฒนาโครงการนี้ ได้เลือกใช้โปรแกรมภาษา PHP เพื่อพัฒนาในส่วนของการติดต่อกับผู้ใช้งานผ่านเว็บและใช้ Apache เป็นเว็บเซิร์ฟเวอร์ สำหรับฐานข้อมูลของระบบได้เลือกใช้ MySQL ในการวิเคราะห์แบบจำลองเชิงแนวคิดของระบบได้อาศัย OMG-Unified Modeling Language (UML) มาใช้โดยแบ่งออกเป็น 3 มุมมองดังนี้คือ Use Case Model นำมาใช้ในการอธิบายระบบงานทั้งหมด และ Class Diagram นำมาใช้ในการวิเคราะห์โครงสร้างข้อมูลของระบบ สำหรับ Sequence Diagram นำมาใช้ในการวิเคราะห์กลไกของระบบในเชิงลักษณะพฤติกรรมของระบบ ซึ่งได้มีการออกแบบให้ระบบสามารถเพิ่มหรือลบกฎที่ทำการสร้าง รวมถึงสามารถสำรองข้อมูลกฎและเรียกกฎที่มีการใช้งานอยู่เดิมก่อนที่จะทำการแก้ไขเปลี่ยนแปลงขึ้นมาทำงาน และแสดงผลการเก็บล็อกข้อมูลการทำงานของไฟร์วอลล์ได้

ผลการศึกษาในครั้งนี้ ได้พัฒนาโปรแกรมให้เป็นไปตามความต้องการโดยได้มีการออกแบบให้โปรแกรมมีส่วนช่วยในการลดข้อผิดพลาดจากการใช้คำสั่งในการสร้างกฎด้วยการใช้การเลือกค่าข้อมูลมากกว่าการป้อนค่าข้อมูล จากผลการทดสอบพบว่าการทำงานของโปรแกรมการกำหนดกฎไฟร์วอลล์สามารถทำการเปิดและปิด อนุญาตและไม่อนุญาตให้ Packet ต่างๆ ได้ตามกฎที่ได้สร้างขึ้นผ่านเว็บโดยผู้ใช้สามารถควบคุมการใช้งานโดยใช้คำสั่งต่างๆ ตามต้องการ

Title	Web-based User Interfacing for IPFIREWALL of FreeBSD
Student	Miss. Tunyaluck Phangchaimongkol
Advisor	Asst. Prof. Akharin Khunkitti
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2005

ABSTRACT

In the present, because of the IPFW uses command line to manage the rules of firewall. It was complicated and more difficult for the user to apply and change the process. This reason was considered to develop this project. So the purposes of the project were to study to develop IPFIREWALL web-based user interfacing of FreeBSD and to learn the using of IPFW technology as a result of PHP programming was used to create a web-based user interfacing. The project chooses Apache as web server and uses MySQL as the database management system. The system analysis design model of IP firewall user interface program was analyzed by OMG-Unified Modeling Language (UML). The model separate to three views, the Conceptual Models was explained by the use case model for the system process. The sequence diagrams explain behavioral process models. The class diagrams explain the structure models. By this analysis, the system requirement has to support increasing and decreasing rules, the backup rules and rollback rules with keeping the log of firewall to show on web.

This study was follow in the requirement of boundary project. The program was designed to help and reduce the mistake point from the input error data by selecting the data more than input the data. The user can detect the log that use in this by web. The summary of the project, the program can use rules to open and close the other packets that were connected network with the permission and use the rules to control the process of IP Firewall.

กิตติกรรมประกาศ

โครงการพัฒนาระบบงานนี้เกิดขึ้น และสำเร็จลุล่วงไปได้ด้วยดี ผู้จัดทำโครงงานขอกราบขอบพระคุณ ผู้ช่วยศาสตราจารย์ อัครินทร์ คุณกิตติ ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงาน ที่ได้กรุณาเสียสละเวลาในการให้คำแนะนำและแนวคิดในการจัดทำโครงงาน และให้คำปรึกษาด้านวิชาการที่เป็นประโยชน์ในการทำโครงงาน และให้ความช่วยเหลือในการแก้ไขเอกสาร เรียบเรียงเอกสาร รวมทั้งได้รับการดูแลเอาใจใส่ ให้ความเมตตา และให้กำลังใจแก่ผู้จัดทำด้วยดีเสมอมา ผู้จัดทำมีความซาบซึ้งในความกรุณาเป็นอย่างยิ่ง จึงขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ.โอกาสนี้

ขอกราบขอบพระคุณอาจารย์ ดร. ธนารัตน์ ชลิดาพงศ์ ที่ได้กรุณาเสียสละเวลาในการให้คำแนะนำ และคำปรึกษาในการวิเคราะห์และออกแบบระบบงานซึ่งช่วยให้ โครงงานนี้สำเร็จลงได้

ขอกราบขอบพระคุณคณาจารย์ทุกๆ ท่านในคณะเทคโนโลยีสารสนเทศที่ได้ให้ความรู้ซึ่งเป็นประโยชน์อย่างยิ่งแก่ข้าพเจ้า

ขอขอบคุณ คุณอานนท์ ทองเต็ม คุณกฤษดา เอกษรัย์ คุณปนัดดา ฉันทมิตรโอภาส คุณจิราภรณ์ ประยูรพีรพุฒิ ที่ได้ให้ข้อมูลอันเป็นประโยชน์ต่อโครงงาน และคุณศราวุธ สีหอม ที่ได้เอื้อเฟื้ออุปกรณ์เพื่อนำมาใช้ในการพัฒนาโครงงานในครั้งนี้ และเพื่อนๆ ทุกคนที่ได้ให้ความช่วยเหลือและให้กำลังใจด้วยดีเสมอมา และบุคคลอีกหลายท่านที่ให้ความช่วยเหลือที่มิได้กล่าวนามไว้ ณ ที่นี้

และสุดท้ายขอกราบขอบพระคุณ คุณพ่อ คุณแม่ ที่ให้กำเนิด ให้การศึกษา น้องสาว น้องชาย น้องเขยทุกคน รวมทั้งน้าชาย ที่เป็นกำลังใจ และเป็นแรงผลักดันให้ผู้จัดทำมีกำลังใจที่จะมุ่งมั่นในการศึกษาครั้งนี้จนเป็นผลสำเร็จลุล่วงด้วยดี

ประโยชน์อันใดที่เกิดจากการศึกษาครั้งนี้ ย่อมเป็นผลมาจากความกรุณาของท่านทั้งหลายดังกล่าวข้างต้นผู้จัดทำรู้สึกทราบซึ้งเป็นอย่างยิ่งจึงใคร่ขอขอบพระคุณเป็นอย่างสูงไว้ ณ.โอกาสนี้

ธัญลักษณ์ ผังชัยมงคล

กุมภาพันธ์ 2549

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

หน้า

ตารางที่

สารบัญรูป

หน้า

รูปที่

สารบัญรูป (ต่อ)

หน้า

รูปที่

บทที่ 1
บทนำ

1. ความเป็นมาของโครงการ

ในปัจจุบันนั้นไฟร์วอลล์ได้มีการใช้งานกันอย่างแพร่หลายในระบบเครือข่ายทั่วไป มีทั้งไฟร์วอลล์ที่เป็นฮาร์ดแวร์และเป็นซอฟแวร์ ซึ่งแต่ละแบบก็มีทั้งข้อดีและข้อเสียแตกต่างกันไป ทั้งนี้ก็ยังจำแนกเป็นไฟร์วอลล์ที่มีลิขสิทธิ์ และ ไฟร์วอลล์ที่เป็นของฟรีได้อีก โดยในแบบที่มีลิขสิทธิ์นี้ทางบริษัทเจ้าของผลิตภัณฑ์ไฟร์วอลล์เหล่านี้ได้มีการออกแบบระบบติดต่อกับผู้ใช้ และระบบตัวช่วยเหลือในการปรับแต่งกฎต่างๆ ให้ผู้ใช้สามารถใช้งานได้ง่ายอยู่แล้ว เช่น Firewall-1 ซึ่งเป็นผลิตภัณฑ์ไฟร์วอลล์ของบริษัท Checkpoint ได้ถูกออกแบบมาให้ง่ายต่อการใช้งานโดยมีลักษณะการทำงานที่เป็นรูปภาพ เป็นต้น แต่ไฟร์วอลล์ที่เป็นของฟรีนั้น โดยส่วนมากแล้วจะมีระบบในการติดต่อกับผู้ใช้และคำสั่งที่ใช้ในการปรับแต่งกฎของไฟร์วอลล์ที่ยุ่งยากซับซ้อน เช่น IPFW ที่เป็นไฟร์วอลล์ที่อยู่ในระบบปฏิบัติการ FreeBSD จะมีการใช้งานไฟร์วอลล์ในลักษณะ command line ซึ่งก่อให้เกิดความยุ่งยากในการสร้างกฎต่างๆ ขึ้นมาเพื่อให้ระบบเครือข่ายมีความปลอดภัย และในขณะเดียวกันก็สามารถใช้งานเครือข่ายได้อย่างปกติ และนอกจากนี้ผู้ใช้งานจะต้องมีความรู้ความชำนาญในเรื่องระบบเครือข่ายจึงเป็นส่วนหนึ่งที่สร้างความลำบากให้กับผู้ใช้งานโดยทั่วไปเป็นอย่างมาก

เพื่อที่จะสามารถสร้างกฎของไฟร์วอลล์ได้มีประสิทธิภาพและง่ายในการติดตั้งเพื่อใช้งาน จึงได้มีแนวคิดที่จะพัฒนายูสเซอร์อินเตอร์เฟสแบบเว็บสำหรับไอพีไฟร์วอลล์ของ FreeBSD เพื่อช่วยในการควบคุมการทำงานและการสร้างกฎของไฟร์วอลล์ให้มีความสะดวกและง่ายต่อการเข้าใจ รวมถึงการควบคุม โดยที่ผู้ใช้งานสามารถเลือกคำสั่งและกำหนดกฎที่จะใช้ในไฟร์วอลล์ได้โดยผ่านทางส่วนติดต่อกับผู้ใช้งานแบบเว็บ และยังสามารถเรียกดูกฎทั้งหมดที่ได้ทำการติดตั้งไว้ในระบบผ่านทางส่วนการติดต่อผู้ใช้งานแบบเว็บเพื่อให้ง่ายต่อการใช้งานได้ อีกทั้งสามารถทำการสำรองข้อมูลกฎก่อนที่จะมีการเปลี่ยนแปลงแก้ไขและสามารถเรียกกฎที่ได้มีการติดตั้งไว้ใช้งานอยู่เดิมก่อนมีการแก้ไขกลับขึ้นมาใช้งานใหม่ได้

1.1 เป้าหมายในการพัฒนาระบบ

เนื่องจากการใช้งานระบบการติดตั้งกฎของไอพีไฟร์วอลล์ที่มีอยู่เดิมเป็นการทำงานโดยการใช้การป้อนคำสั่งในลักษณะ Command line จึงทำให้เกิดความซับซ้อนและยุ่งยากต่อการเรียกใช้งาน ดังนั้นระบบที่จะมีการพัฒนาขึ้นมาใหม่จะเป็นระบบที่มีการรองรับการใช้งานโดยที่ผู้ใช้สามารถที่จะเรียกใช้ระบบการติดตั้งกฎผ่านเว็บและทำการติดตั้งคำสั่งโดยการป้อนเพียงข้อมูลที่จำเป็นต่อระบบโดยในการทำงานผู้ใช้ไม่จำเป็นต้องจดจำคำสั่งในการเรียกใช้หรือการติดตั้งกฎของไฟร์วอลล์ และสามารถเรียกดูกฎทั้งหมดที่ได้ทำการติดตั้งไว้แล้วได้ รวมทั้งยังสามารถทำการสำรองข้อมูลกฎก่อนที่จะมีการเปลี่ยนแปลงแก้ไขและสามารถเรียกกฎที่ได้มีการติดตั้งไว้ใช้งานอยู่เดิมก่อนมีการแก้ไขกลับขึ้นมาใช้งานใหม่ได้

1.2 ขอบเขตในการพัฒนาระบบ

1. ติดตั้งระบบปฏิบัติการ FreeBSD เพื่อเป็นระบบปฏิบัติการของระบบไฟร์วอลล์
2. ติดตั้งไฟร์วอลล์ IPFW บน FreeBSD เพื่อทำหน้าที่เป็นระบบไฟร์วอลล์
3. ติดตั้ง Web Server เพื่อเป็นส่วนติดต่อกับผู้ใช้ผ่านเว็บ
4. ติดตั้ง Database Server เพื่อเก็บข้อมูลที่จำเป็นของระบบ
5. พัฒนาโปรแกรมระบบการกำหนดกฎไฟร์วอลล์ โดยที่โปรแกรมมีความสามารถในการทำงานดังนี้
   - สามารถป้อนข้อมูลที่จำเป็นสำหรับการสร้างกฎของไฟร์วอลล์ ซึ่งระบุได้ถึง IP Source, Port Source, IP Destination, Port Destination, Protocol, Direction, Interface
   - สามารถเพิ่มหรือลบกฎที่ทำการสร้างขึ้นได้
   - สามารถแสดงกฎที่สร้างขึ้นใหม่และกฎที่มีอยู่เดิมได้
   - สามารถสำรองข้อมูลกฎและเรียกกฎที่มีการใช้งานอยู่เดิมก่อนที่จะทำการแก้ไขเปลี่ยนแปลงขึ้นมาทำงานได้
   - สามารถแสดงผลการทำงานของไฟร์วอลล์ได้
   - สามารถแสดงผลของล๊อกข้อมูลการทำงานของกฎ (Log file) เพื่อใช้วิเคราะห์การทำงานของระบบ

1.3 องค์ประกอบของระบบงาน

ระบบงานประกอบด้วยองค์ประกอบต่างๆ ดังต่อไปนี้

1.3.1 **เครื่องคอมพิวเตอร์เซิร์ฟเวอร์** ระบบปฏิบัติการ FreeBSD ทำหน้าที่ตรวจสอบข้อมูลที่ผ่านเข้าออก ซึ่งได้รับการเตรียมความพร้อมดังนี้

- ติดตั้งระบบปฏิบัติการ FreeBSD version 5 เพื่อรองรับการตรวจสอบข้อมูลที่ผ่านเข้าออก
- ติดตั้งซอฟต์แวร์ไฟร์วอลล์ โดยเลือกใช้ โปรแกรม IPFW
- ติดตั้งส่วนให้บริการเว็บเซิร์ฟเวอร์สำหรับเป็นส่วนติดต่อกับผู้ใช้เพื่อใช้ในการควบคุมระบบ โดยเลือกใช้ Apache เวอร์ชัน2.0.53
- ติดตั้งซอฟต์แวร์ระบบฐานข้อมูล โดยเลือกใช้ MySQL เวอร์ชัน 4
- ติดตั้งซอต์แวร์ภาษา โดยเลือกใช้ PHP เวอร์ชัน 5
- ติดตั้งโปรแกรมระบบการกำหนดกฎไฟร์วอลล์

1.3.2 **เครื่องคอมพิวเตอร์ไคลเอ็นท์** ที่ใช้ควบคุมระบบการกำหนดกฎไฟร์วอลล์โดยผ่านทางเว็บ ซึ่งได้รับการเตรียมความพร้อมดังนี้

- ติดตั้งระบบปฏิบัติการ Windows XP Professional
- ติดตั้งโปรแกรมเว็บบราวเซอร์ สำหรับติดต่อกับเว็บเซิร์ฟเวอร์ เพื่อควบคุมโปรแกรมการกำหนดกฎไฟร์วอลล์

1.4 ขั้นตอนในการพัฒนาระบบ

ประกอบไปด้วยขั้นตอนต่างๆ ดังนี้

1.4.1 **ศึกษาความเป็นไปได้ในการพัฒนาระบบการกำหนดกฎไฟร์วอลล์ของ FreeBSD**

เพื่อกำหนดขอบเขตของปัญหาและวางแผนวิธีการพัฒนาโปรแกรม รวมถึงกำหนดเป้าหมายในการพัฒนาโครงการ โดยศึกษา ดังนี้

- ศึกษาวิธีการติดตั้งและการใช้งานซอฟต์แวร์ต่างๆ ที่ใช้สำหรับการสร้างระบบไฟร์วอลล์และระบบเว็บเซิร์ฟเวอร์ เพื่อรองรับการติดต่อจากภายนอก ได้แก่ FreeBSD, IPFW, Apache Web Server, PHP Extension, MySQL, Perl
- ศึกษาเทคโนโลยีไฟร์วอลล์ IPFW บนระบบปฏิบัติการ FreeBSD
- ศึกษาเทคโนโลยีการรักษาความปลอดภัยเครือข่ายโดยใช้ไฟร์วอลล์
- ศึกษาวิธีการสร้างกฎสำหรับไฟร์วอลล์ให้มีประสิทธิภาพ
- ศึกษาจากเอกสารคู่มือ OMG Unified Modeling Language Specification V1.4
- ศึกษาการใช้งานโปรแกรมฐานข้อมูล MySQL เพื่อเก็บข้อมูลของระบบ

- ศึกษาการใช้งานโปรแกรมภาษา PHP Extension เพื่อพัฒนาโปรแกรมประยุกต์ ในส่วนของการติดต่อกับผู้ใช้งาน

1.4.2 การวิเคราะห์และออกแบบ

ทำการวิเคราะห์และออกแบบรวมถึงกำหนดความต้องการของโครงการพัฒนาระบบ โดยได้ทำการออกแบบให้ระบบสามารถเพิ่มหรือลบกฎที่ทำการสร้างและป้อนข้อมูลที่จำเป็นสำหรับการสร้างกฎของไฟร์วอลล์ ซึ่งระบุได้ถึง IP Source, Port Source, IP Destination, Port Destination, Protocol, Direction, Interface รวมถึงสามารถสำรองข้อมูลกฎและเรียกกฎที่มีการใช้งานอยู่เดิมก่อนที่จะทำการแก้ไขเปลี่ยนแปลงขึ้นมาทำงานได้ และแสดงผลการตรวจสอบการทำงานของไฟร์วอลล์

1.4.3 การพัฒนาและทดสอบ

- ทำการติดตั้งระบบไฟร์วอลล์และทดสอบการทำงานของไอพีไฟร์วอลล์
- ทำการพัฒนาโปรแกรมและทดสอบการทำงานของโปรแกรมในฟังก์ชันต่างๆ

1.4.4 การทดลองใช้งานและปรับปรุงแก้ไข

นำโปรแกรมมาทดลองใช้งานและปรับปรุงแก้ไขเพื่อให้สามารถใช้งานได้ถูกต้องและง่ายยิ่งขึ้น

1.5 ประโยชน์ที่คาดว่าจะได้รับ

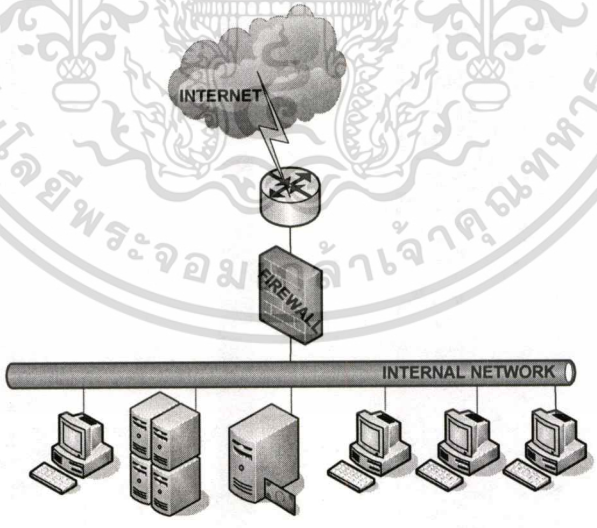
1. ได้พัฒนาความรู้ความเข้าใจเรื่องการทำงานพื้นฐานของไฟร์วอลล์
2. ได้พัฒนาความรู้ความเข้าใจเรื่องการทำงานของไฟร์วอลล์ IPFW
3. ได้พัฒนาความรู้ความสามารถในการวิเคราะห์ ออกแบบและพัฒนาระบบงานและสามารถนำไปใช้ประโยชน์ต่อการทำงานในอนาคตได้
4. ได้โปรแกรมประยุกต์ที่ผู้ดูแลระบบหรือผู้ใช้งานทั่วไป สามารถเรียกใช้งานและทำการแก้ไขการทำงานกฎต่างๆของระบบไอพีไฟร์วอลล์ได้โดยสะดวก รวดเร็ว และ ง่ายต่อการใช้งานยิ่งขึ้นในลักษณะการทำงานแบบเว็บ
5. ช่วยส่งเสริมให้เกิดการใช้งานไฟร์วอลล์แพร่หลายมากยิ่งขึ้น
6. เป็นอีกทางเลือกหนึ่งในการเลือกใช้โปรแกรมประยุกต์ที่ช่วยให้การทำงานกับไฟร์วอลล์เป็นเรื่องง่าย

บทที่ 2
ไอพีไฟร์วอลล์

เนื่องจากในปัจจุบันการทำงานได้มีการใช้เทคโนโลยีระบบเครือข่ายอินเทอร์เน็ตเข้ามาช่วยในการติดต่อสื่อสารแลกเปลี่ยนข้อมูลระหว่างภายในและภายนอกองค์กร ซึ่งมีทั้งข้อมูลที่เป็นความลับและมีความสำคัญต่อองค์กร และข้อมูลที่ไม่สำคัญ เช่น สื่อโฆษณาอิเลคทรอนิคต่างๆ เป็นต้น ซึ่งข้อมูลที่สำคัญเหล่านี้เป็นข้อมูลที่จำเป็นต้องดูแลไม่ให้ข้อมูลดังกล่าวถูกนำไปใช้งานโดยไม่ถูกต้อง ดังนั้นสิ่งหนึ่งที่ทำให้ผู้ดูแลระบบเครือข่ายส่วนใหญ่ให้ความสำคัญจึงเป็นไปในเรื่องของการรักษาความปลอดภัยให้กับระบบเครือข่ายและอุปกรณ์หนึ่งที่ถูกพิจารณาก็คือ ไฟร์วอลล์

2.1 ทำความรู้จักกับไฟร์วอลล์

ไฟร์วอลล์เป็นระบบที่เอาไว้ป้องกันอันตรายจากอินเทอร์เน็ต หรือเครือข่ายภายนอกโดยการควบคุมการเข้า-ออกของข้อมูล ดังนั้นไฟร์วอลล์จึงเป็นคอมโพเน็นต์หรือกลุ่มของคอมโพเน็นต์



รูปที่ 2.1 ไฟร์วอลล์กั้นระหว่างอินเทอร์เน็ตกับเครือข่ายภายใน

ที่ทำหน้าที่ในการควบคุมการเข้าถึงระหว่างเครือข่ายภายนอก (เครือข่ายที่คิดว่าไม่ปลอดภัย) กับเครือข่ายภายใน(เครือข่ายที่ต้องการจะป้องกัน) โดยที่คอมโพเน็นต์นั้นอาจจะเป็น เราเตอร์

คอมพิวเตอร์ หรือเครือข่าย ประกอบกันก็ได้ขึ้นอยู่กับวิธีการหรือสถาปัตยกรรมของไฟร์วอลล์ (Firewall Architecture) ที่ใช้การควบคุมการเข้าถึงของไฟร์วอลล์นั้น สามารถทำได้ในหลายระดับ และหลายรูปแบบขึ้นอยู่กับชนิดหรือเทคโนโลยีของไฟร์วอลล์ที่นำมาใช้ โดยสามารถกำหนดได้ว่าจะให้มีการเข้ามาใช้บริการอะไรได้บ้าง จากที่ไหน เป็นต้น

ไฟร์วอลล์สามารถช่วยเพิ่มความปลอดภัยให้กับระบบได้โดย

- บังคับใช้นโยบายด้านความปลอดภัย โดยการกำหนดกฎให้กับไฟร์วอลล์ว่าจะอนุญาตหรือไม่ให้ใช้บริการชนิดใด
- ทำให้การพิจารณาดูแลและการตัดสินใจด้านความปลอดภัยของระบบเป็นไปได้ง่ายขึ้น เนื่องจากการติดต่อทุกชนิดกับเครือข่ายภายนอกจะต้องผ่านไฟร์วอลล์ การดูแลที่จุดนี้เป็นการดูแลความปลอดภัยในระดับของเครือข่าย (Network-based Security)
- บันทึกข้อมูล กิจกรรมต่างๆ ที่ผ่านเข้าออกเครือข่ายได้อย่างมีประสิทธิภาพ
- ป้องกันเครือข่ายบางส่วนจากการเข้าถึงของเครือข่ายภายนอก เช่น ถ้าหากมีบางส่วนที่ต้องการให้เครือข่ายภายนอกเข้ามาใช้บริการ (เช่น ถ้ามีเว็บเซิร์ฟเวอร์) แต่ส่วนที่เหลือไม่ต้องการให้เครือข่ายภายนอกเข้ามากรณี เช่นนี้จะสามารถใช้ไฟร์วอลล์ช่วยได้
- ไฟร์วอลล์บางชนิดสามารถป้องกันไวรัสได้โดยจะทำการตรวจสอบไฟล์ที่โอนย้ายผ่านทางโปรโตคอล HTTP, FTP และ SMTP

ถึงแม้ว่าไฟร์วอลล์จะสามารถช่วยเพิ่มความปลอดภัให้กับเครือข่ายได้เป็นอย่างมากโดยการตรวจดูข้อมูลที่ผ่านเข้าออก แต่ก็มี**ข้อจำกัดบางประการที่ไฟร์วอลล์ไม่สามารถป้องกันได้** ได้แก่

- อันตรายที่เกิดจากเครือข่ายภายใน ไม่สามารถป้องกันได้เนื่องจากอยู่ภายในเครือข่ายเองไม่ได้ผ่านไฟร์วอลล์เข้ามา
- อันตรายจากภายนอกที่ไม่ได้ผ่านเข้ามาทางไฟร์วอลล์ เช่นการ Dial-up เข้ามายังเครือข่ายภายในโดยตรงโดยไม่ได้ผ่านไฟร์วอลล์
- อันตรายจากวิธีใหม่ๆ ที่เกิดขึ้น ซึ่งทุกวันนี้มีการพบช่องโหว่ใหม่ๆ เกิดขึ้นทุกวัน จึงไม่สามารถไว้ใจไฟร์วอลล์โดยการติดตั้งเพียงครั้งเดียวแล้วคาดหวังให้ปลอดภัยตลอดไป ดังนั้นจึงต้องมีการดูแลรักษาอย่างต่อเนื่องสม่ำเสมอ
- ไวรัส ถึงแม้จะมีไฟร์วอลล์บางชนิดที่สามารถป้องกันไวรัสได้ แต่ก็ยังไม่มีไฟร์วอลล์ชนิดใดที่สามารถตรวจสอบไวรัสได้ในทุกๆ โปรโตคอล

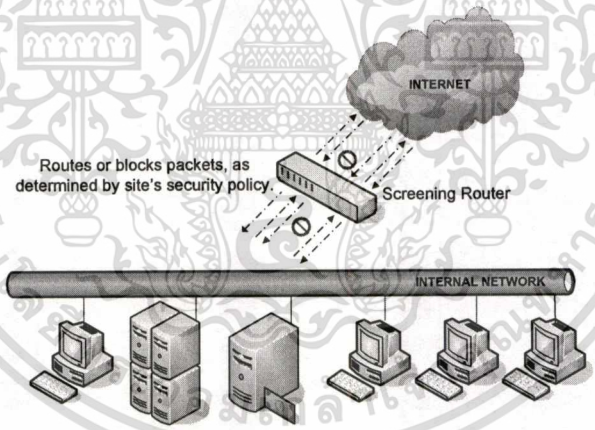
2.2 ชนิดของไฟร์วอลล์

ชนิดของไฟร์วอลล์แบ่งตามเทคโนโลยีที่ใช้ในการตรวจสอบและควบคุม แบ่งได้เป็น 3 ชนิด คือ

- ไฟร์วอลล์ชนิดกรองแพ็คเก็ต(Packet Filtering)
- Proxy Service
- Stateful Multilayer Inspection Firewall

2.2.1 ไฟร์วอลล์ชนิดกรองแพ็คเก็ต (Packet Filtering)

ไฟร์วอลล์ชนิดกรองแพ็คเก็ต (Packet Filtering) คือ ไฟร์วอลล์ที่ทำการหาเส้นทางและส่งต่อ (route) อย่างมีเงื่อนไข โดยจะพิจารณาจากข้อมูลส่วนที่อยู่ในเฮดเดอร์ (header) ของแพ็กเก็ตที่ผ่านเข้ามา เทียบกับกฎ (rules) ที่กำหนดไว้และตัดสินว่าควรจะทิ้ง (drop) แพ็กเก็ตนั้นไปหรือว่าจะยอม (accept) ให้แพ็กเก็ตนั้นผ่านไปได้



รูปที่ 2.2 ใช้ Screening Router ทำหน้าที่ Packet Filtering

ในการพิจารณาเฮดเดอร์ Packet Filter จะตรวจสอบในระดับของอินเทอร์เน็ตเลเยอร์ (Internet Layer) และทรานสปอร์ตเลเยอร์ (Transport Layer) ในอินเทอร์เน็ตโมเดล ซึ่งในอินเทอร์เน็ตเลเยอร์จะมีแอตทริบิวต์ที่สำคัญต่อ Packet Filtering ดังนี้

- ไอพีแอดเดรสต้นทาง
- ไอพีแอดเดรสปลายทาง
- ชนิดของโปรโตคอล (TCP UDP และ ICMP)

และในระดับของทรานสปอร์ตเลเยอร์ มีแอตทริบิวต์ที่สำคัญคือ

- พอร์ตต้นทาง (TCP/UDP)
- พอร์ตปลายทาง (TCP/UDP)
- แฟล็ก (Flag ซึ่งจะมีเฉพาะในเฮดเดอร์ของแพ็กเก็ต TCP)
- ชนิดของ ICMP message (ในแพ็กเก็ต ICMP)

ซึ่งพอร์ตของทรานสปอร์ตเลเยอร์ คือทั้ง TCP และ UDP นั้นจะเป็นสิ่งที่บอกถึงแอพพลิเคชันที่แพ็กเก็ตนั้นต้องการติดต่อด้วยเช่น พอร์ต 80 หมายถึง HTTP, พอร์ต 21 หมายถึง FTP เป็นต้น ดังนั้นเมื่อ Packet Filter พิจารณาเฮดเดอร์ จึงทำให้สามารถควบคุมแพ็กเก็ตที่มาจากที่ต่างๆ และมีลักษณะต่างๆ (ดูได้จากแฟล็กของแพ็กเก็ต หรือ ชนิดของ ICMP ในแพ็กเก็ต ICMP) ได้ เช่น ห้ามแพ็กเก็ตทุกชนิดจาก crack.cracker.net เข้ามายังเครือข่าย 203.154.207.0/24 , ห้ามแพ็กเก็ตที่มีไอพีแอดเดรสต้นทางอยู่ในเครือข่าย 203.154.207.0/24 ผ่านเราเตอร์เข้ามา(ในกรณีนี้เพื่อเป็นการป้องกัน ip spoofing) เป็นต้น

Packet Filtering สามารถอิมพลีเมนต์ได้จาก 2 แพล็ตฟอร์ม คือ

- เราเตอร์ที่มีความสามารถในการทำ Packet Filtering (ซึ่งมีในเราเตอร์ส่วนใหญ่อยู่แล้ว)
- คอมพิวเตอร์ที่ทำหน้าที่เป็นเราเตอร์

ซึ่งจะมีข้อได้เปรียบเสียเปรียบกันดังนี้

ตารางที่ 2.1 เปรียบเทียบข้อดีข้อเสียในการเลือกอุปกรณ์มาทำหน้าที่ Packet Filtering

ชนิด	ข้อดี	ข้อเสีย
เราเตอร์	ประสิทธิภาพสูงมีจำนวนอินเตอร์-เฟสมาก	เพิ่มเติมฟังก์ชันการทำงานได้ยาก, อาจต้องการหน่วยความจำมาก
คอมพิวเตอร์ที่ทำหน้าที่เป็นเราเตอร์	เพิ่มฟังก์ชันการทำงานได้ไม่จำกัด	ประสิทธิภาพปานกลาง,จำนวนอินเตอร์เฟสน้อย,อาจมีความเสี่ยงจากระบบปฏิบัติการที่ใช้

ข้อดี-ข้อเสียของ Packet Filtering

ข้อดี

- ไม่ขึ้นกับแอพพลิเคชัน
- มีความเร็วสูง
- รองรับการขยายตัวได้ดี

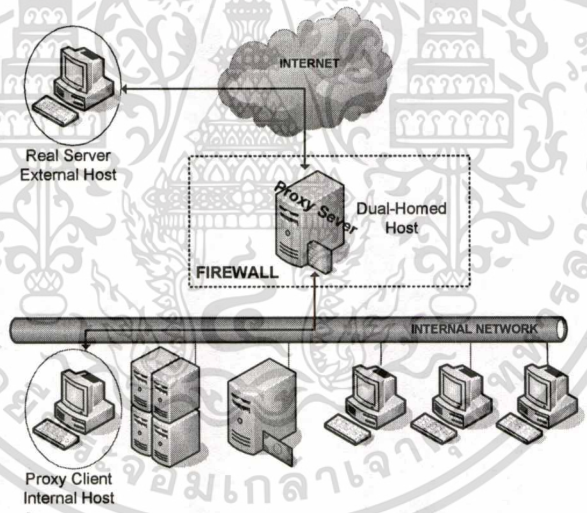
ข้อเสีย

- บางโปรโตคอลไม่เหมาะสมกับการใช้ Packet Filtering เช่น FTP, ICQ เป็นต้น

2.2.2 ไฟร์วอลล์ชนิด Proxy

Proxy หรือ Application Gateway เป็นแอพพลิเคชันโปรแกรมที่ทำงานอยู่บนไฟร์วอลล์ที่ตั้งอยู่ระหว่างเครือข่าย 2 เครือข่าย ทำหน้าที่เพิ่มความปลอดภัยของระบบเครือข่ายโดยการควบคุมการเชื่อมต่อระหว่างเครือข่ายภายในและภายนอก Proxy จะช่วยเพิ่มความปลอดภัยได้มากเนื่องจากมีการตรวจสอบข้อมูลถึงในระดับของแอพพลิเคชันเลเยอร์ (Application Layer)

เมื่อไคลเอนต์ต้องการใช้บริการภายนอก 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 จะส่งต่อแพ็กเก็ตให้หรือไม่



รูปที่ 2.3 ใช้ Dual-homed Host* เป็น Proxy Server

ข้อดี-ข้อเสียของ Proxy

<u>ข้อดี</u>

- มีความปลอดภัยสูง
- รู้จักข้อมูลในระดับแอพพลิเคชัน

<u>ข้อเสีย</u>

- ประสิทธิภาพต่ำแต่ละบริการมักจะต้องการโปรเซสของตนเอง
- สามารถขยายตัวได้ยาก

2.2.3 ไฟร์วอลล์ชนิด Stateful Multilayer Inspection Technology

โดยปกติแล้ว Packet Filtering แบบธรรมดา (ที่เป็น Stateless แบบที่มีอยู่ในเราเตอร์ทั่วไป) จะควบคุมการเข้าออกของแพ็กเก็ตโดยพิจารณาข้อมูลจากเฮดเดอร์ของแต่ละแพ็กเก็ต นำมาเทียบกับกฎที่มีอยู่ ซึ่งกฎที่มีอยู่ก็จะเป็นกฎที่สร้างจากข้อมูลส่วนที่อยู่ในเฮดเดอร์เท่านั้น ดังนั้น Packet Filtering แบบธรรมดาจึงไม่สามารถทราบได้ว่า แพ็กเก็ตนี้อยู่ส่วนใดของการเชื่อมต่อ เป็นแพ็กเก็ตที่เข้ามาติดต่อใหม่หรือไม่ หรือว่าเป็นแพ็กเก็ตที่เป็นส่วนของการเชื่อมต่อที่เกิดขึ้นแล้ว เป็นต้น

Stateful Multilayer Inspection เป็นเทคโนโลยีที่เพิ่มเข้าไปใน Packet Filtering โดยในการพิจารณาว่าจะยอมให้แพ็กเก็ตใดผ่านไปนั้น แทนที่จะดูข้อมูลจากเฮดเดอร์เพียงอย่างเดียว Stateful Multilayer Inspection จะนำเอาส่วนข้อมูลของแพ็กเก็ต (message content) และข้อมูลที่ได้จากแพ็กเก็ตก่อนหน้านี้ที่ได้ทำการบันทึกเอาไว้ นำมาพิจารณาด้วย จึงทำให้สามารถระบุได้ว่าแพ็กเก็ตใดเป็นแพ็กเก็ตที่ติดต่อเข้ามาใหม่ หรือว่าเป็นส่วนหนึ่งของการเชื่อมต่อที่มีอยู่แล้ว

ตัวอย่าง ผลิตภัณฑ์ทางการค้าที่ใช้ Stateful Multilayer Inspection Technology ได้แก่

- Check Point Firewall-1
- Cisco Secure Pix Firewall
- SunScreen Secure Net

และส่วนที่เป็น Open source แจกฟรี ได้แก่

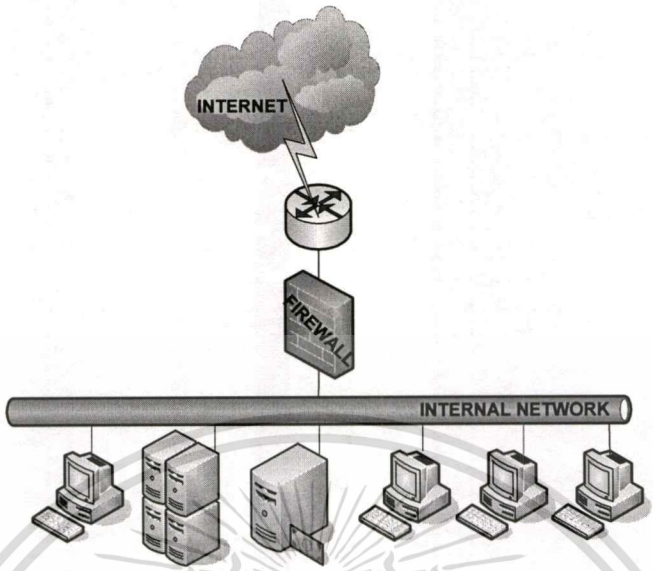
- NetFilter ใน Linux (มี iptables ในลีนุกซ์เคอร์เนล 2.3)

2.3 สถาปัตยกรรมไฟร์วอลล์ (Firewall Architecture)

ในส่วนของสถาปัตยกรรมไฟร์วอลล์ จะกล่าวถึงการจัดวางไฟร์วอลล์คอมโพเน็นต์ในแบบต่างๆ เพื่อทำให้เกิดเป็นระบบไฟร์วอลล์ขึ้น โดยแบ่งรูปแบบสถาปัตยกรรมได้ดังนี้

2.3.1 Single Box Architecture

Single Box Architecture เป็นสถาปัตยกรรมแบบง่ายๆ ที่มีคอมโพเน็นต์ทำหน้าที่เป็นไฟร์วอลล์เพียงอันเดียวตั้งอยู่ระหว่างเครือข่ายภายในกับเครือข่ายภายนอก ข้อดีของวิธีนี้ก็คือ การที่มีเพียงจุดเดียวที่หน้าที่ไฟร์วอลล์ทั้งหมด ควบคุมการเข้าออกของข้อมูล ทำให้ดูแลได้ง่าย เป็นจุดสนใจในการดูแลความปลอดภัยเครือข่าย ในทางกลับกันข้อเสียของวิธีนี้ก็คือ การที่มีเพียงจุดเดียวนี้ทำให้มีความเสี่ยงสูง หากมีการคอนฟิกูเรชันผิดพลาดหรือมีช่องโหว่เพียงเล็กน้อย การผิดพลาดเพียงจุดเดียวอาจทำให้ระบบถูกเจาะได้



รูปที่ 2.4 Firewall Architecture แบบชั้นเดียว

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

1) Screening Router Architecture สามารถใช้เราเตอร์ทำ Packet Filtering ซึ่งวิธีดังกล่าวนี้จะทำให้ประหยัดค่าใช้จ่ายเนื่องจากส่วนใหญ่จะใช้เราเตอร์ต่อกับเครือข่ายภายนอกอยู่แล้ว แต่วิธีนี้อาจไม่ยืดหยุ่นมากนักในการคอนฟิกกูเรชัน

สถาปัตยกรรม นี้เหมาะสำหรับ

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

2) Dual-Homed Host Architecture สามารถใช้ Dual-Homed Host (คอมพิวเตอร์ที่มีเครือข่ายอินเตอร์เฟสอย่างน้อย 2 อินเตอร์เฟส) เพื่อให้การบริการเป็น Proxy ให้กับเครื่องภายในเครือข่าย

สถาปัตยกรรม แบบนี้เหมาะสำหรับ

- เครือข่ายที่มีการใช้งานอินเทอร์เน็ตค่อนข้างน้อย
- เครือข่ายที่ไม่ได้มีข้อมูลสำคัญๆ

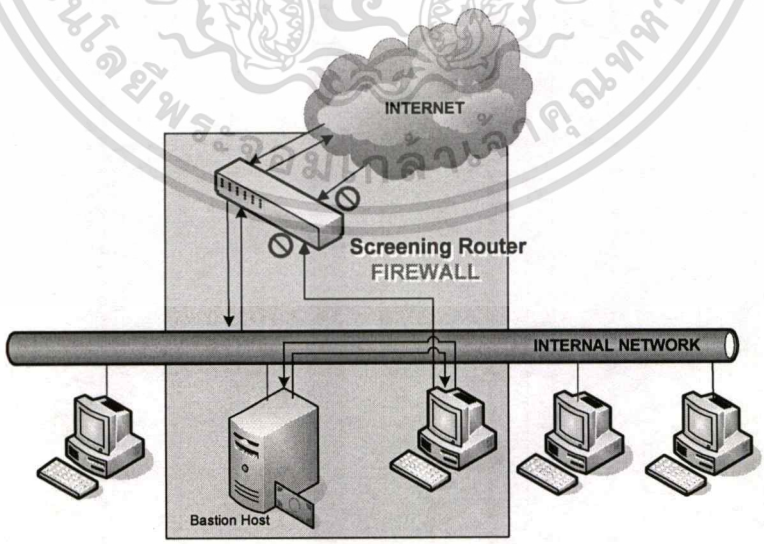
3) Multi-purposed Firewall Box เป็นผลิตภัณฑ์หลายชนิดที่ผลิตออกมาเป็นกล่องๆ เดียว และสามารถทำหน้าที่ได้หลายอย่างเช่น เป็นPacket Filtering, Proxy แต่เนื่องด้วยเป็น สถาปัตยกรรมแบบชั้นเดียว ซึ่งถ้าหากเกิดข้อผิดพลาดแล้วก็จะสร้างความเสียหายให้ทั้งเครือข่ายได้

2.3.2 Screened Host Architecture

Screened Host Architecture จะมีโฮสต์ซึ่งให้บริการ Proxy เหมือนกับใน Single Box Architecture ที่เป็น Dual-homed Host แต่จะต่างกันตรงที่ว่า โฮสต์นั้นจะอยู่ภายในเครือข่าย ไม่ต่ออยู่กับเครือข่ายภายนอกอื่นๆ (ดังนั้นก็ไม่จำเป็นที่จะต้องใช้ Dual Homed Host) และจะมี เราเตอร์ที่ทำหน้าที่ Packet Filtering ช่วยบังคับให้เครื่องภายในเครือข่ายต้องติดต่อบริการผ่าน Proxy โดยไม่ยอมให้ติดต่อใช้บริการจากภายนอกโดยตรง และก็ให้ภายนอกเข้าถึงได้เฉพาะ Bastion host (คือโฮสต์ที่มีความเสี่ยงสูงต่อการถูกโจมตี มักจะเป็นโฮสต์ที่เปิดให้บริการกับอินเทอร์เน็ต ดังนั้นโฮสต์นี้ต้องมีการดูแลเป็นพิเศษ) เท่านั้น จากรูปที่ 2.5 ใน สถาปัตยกรรมแบบนี้จะประกอบไปด้วยเราเตอร์ทำหน้าที่ Packet Filtering และภายในเครือข่ายจะมี Bastion Host ให้บริการ Proxy อยู่ โดยที่เราเตอร์นั้นอาจจะถูกกำหนดได้ดังนี้

- อาจจะอนุญาตให้เครื่องภายในใช้บริการบางอย่างได้โดยตรง
- ส่วนบริการอื่นๆ จะไม่ยอมให้เครื่องภายในติดต่อผ่านออกไปโดยตรง ยกเว้น Bastion Host เท่านั้นที่สามารถติดต่อกับเครือข่ายภายนอกได้ทั้งนี้เพื่อเป็นการบังคับให้ใช้บริการ Proxy ผ่านทาง Bastion Host เท่านั้น

หรืออาจจะกำหนดให้บริการส่วนใหญ่ผ่านเราเตอร์ออกไปได้โดยตรงแล้ว ให้บางส่วนต้องใช้บริการผ่าน Proxy ทั้งนี้ก็ขึ้นอยู่กับนโยบายและความเหมาะสมขององค์กร



รูปที่ 2.5 Screened Host Architecture

วิธีนี้ถึงแม้ว่าจะมีทั้ง Proxy และเราเตอร์ทำหน้าที่ Packet Filtering แต่ก็ยังคงอันตรายอยู่ เพราะว่าเราเตอร์ต้องยอมให้ภายนอกสามารถติดต่อกับ Bastion Host ได้ ดั้งหากแฮกเกอร์สามารถเจาะเข้ามายัง Bastion Host ได้ก็จะทำให้เกิดปัญหาขึ้นได้

สถาปัตยกรรม นี้เหมาะสำหรับ

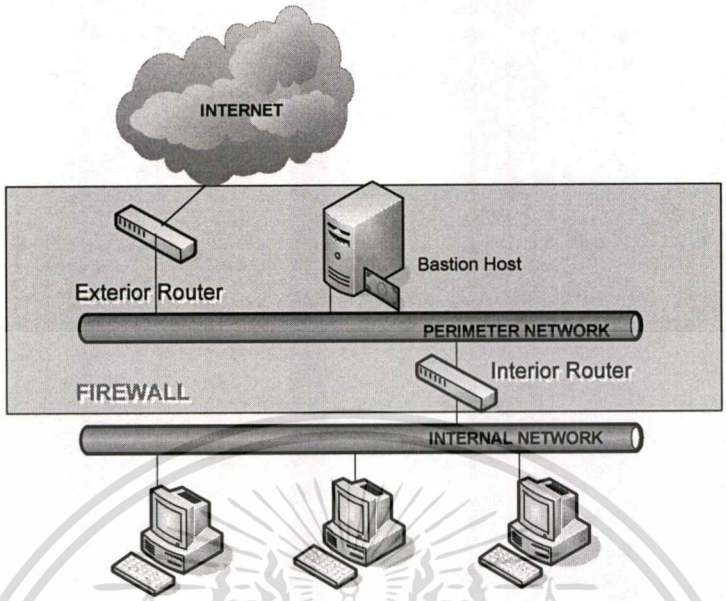
- เครือข่ายที่มีการติดต่อกับเครือข่ายภายนอกน้อย
- เครือข่ายที่มีการป้องกันความปลอดภัยในระดับของโฮสต์เป็นอย่างดีแล้ว

2.3.3 Multi Layer Architecture

ในสถาปัตยกรรมแบบหลายชั้น ไฟร์วอลล์จะเกิดขึ้นจากคอมโพเน็นต์หลายๆส่วนทำหน้าที่ประกอบกันขึ้นเป็นระบบ วิธีการนี้สามารถเพิ่มความปลอดภัยได้มาก เนื่องจากเป็นการลดความเสี่ยงต่อความผิดพลาดที่อาจเกิดขึ้น ถ้าหากมีไฟร์วอลล์เพียงจุดเดียวแล้วมีเกิดความผิดพลาดเกิดขึ้นระบบทั้งหมดก็จะเป็นอันตราย แต่ถ้ามีการป้องกันหลายชั้น หากในชั้นแรกถูกเจาะ ก็อาจจะมีความเสียหายเพียงบางส่วน ส่วนที่เหลือระบบก็ยังคงมีชั้นอื่นๆ ในการป้องกันอันตราย และยังลดความเสี่ยงได้โดยการที่แต่ละชั้นนั้นมีการใช้เทคโนโลยีที่แตกต่างกัน เพื่อให้เกิดความหลากหลาย เป็นการหลีกเลี่ยงการโจมตีหรือ ช่องโหว่ที่อาจมีในเทคโนโลยีชนิดใดชนิดหนึ่ง โดยทั่วไปแล้วสถาปัตยกรรมแบบหลายชั้นจะเป็นการต่อกันเป็นลำดับ โดยมี Perimeter Network (หรือบางทีเรียกว่า DMZ Network) อยู่ตรงกลาง เรียกว่า Screened Subnet Architecture

2.3.4 Screened Subnet Architecture

Screened Subnet Architecture เป็นสถาปัตยกรรมที่มีการเพิ่ม Perimeter Network เข้าไปกั้นระหว่างอินเทอร์เน็ตกับเครือข่ายภายในไม่ให้เชื่อมต่อกันโดยตรง ทำให้เครือข่ายภายในมีความปลอดภัยมากขึ้นดังในรูปที่ 2.6 แสดง Screened Subnet Architecture อย่างง่าย ประกอบไปด้วย เราเตอร์ 2 ตัว ตัวนึงอยู่ระหว่างอินเทอร์เน็ตกับ Perimeter Network ส่วนอีกตัวหนึ่งอยู่ระหว่าง Perimeter Network กับเครือข่ายภายใน ถ้าหากแฮกเกอร์จะเจาะเครือข่ายภายในต้องผ่านเราเตอร์เข้ามาถึง 2 ตัวด้วยกัน ถึงแม้ว่าจะเจาะชั้นแรกเข้ามายัง Bastion host ได้ แต่ก็ยังต้องผ่านเราเตอร์ตัวในอีก ถึงจะเข้ามายังเครือข่ายภายในได้



รูปที่ 2.6 Screened Subnet Architecture

คอมโพเน็นต์ของ Screened Subnet Architecture ในรูปที่ 2.6

- Perimeter Network เป็นเครือข่ายที่เพิ่มเข้ามาเพื่อความปลอดภัย อยู่ระหว่างเครือข่ายภายนอกกับเครือข่ายภายใน ประโยชน์ของ Perimeter Network ที่เห็นได้ชัดก็คือ การแบ่งเครือข่ายออกเป็นส่วนๆ ทำให้การไหลของข้อมูลถูกแบ่งออกเป็นส่วนๆตามเครือข่ายด้วย เนื่องจากโดยทั่วไปแล้ว เครือข่ายที่เป็นแลนนั้น จะเป็นแบบ Ethernet ซึ่งจะมีการส่งข้อมูลแบบ Broadcast ดังนั้นถ้ามีใครลอบดักจับข้อมูลอยู่ในเครือข่ายนั้น ก็จะได้พาสเวิร์ด ข้อมูลต่างๆ ไปหมด ดังนั้นหากไฟร์วอลล์มีชั้นเดียวและแฮกเกอร์สามารถเข้ามาได้ และโดนดักจับก็จะสูญเสียข้อมูลทั้งหมด แต่ถ้ามี Perimeter Network ถึงจะดักจับข้อมูลได้แต่ก็จะได้เพียงที่อยู่บน Perimeter Network เท่านั้น
- Bastion Host ตั้งอยู่บน Perimeter Network ทำหน้าที่ให้บริการ Proxy กับเครือข่ายภายในและให้บริการต่างๆ กับผู้ใช้บนอินเทอร์เน็ต Bastion Host นั้นจะมีความเสี่ยงต่อการโจมตีสูง จึงต้องมีการดูแลความปลอดภัยเป็นพิเศษ
- Interior Router ตั้งอยู่ระหว่าง Perimeter Network กับเครือข่ายภายใน ทำหน้าที่ Packet Filtering ป้องกันเครือข่ายภายในจาก Perimeter Network ในการกำหนดค่า configuration ระหว่าง เครือข่ายภายในกับ Perimeter Network ควรกำหนดอย่างรอบคอบ อนุญาตเฉพาะบริการที่จำเป็นเท่านั้นอย่างเช่น DNS, SMTP เป็นต้น

- Exterior Router ตั้งอยู่ระหว่างเครือข่ายภายนอกกับ Perimeter Network เนื่องจาก Exterior Router นี้เป็นจุดที่ต่ออยู่กับเครือข่ายภายนอก จึงมีหน้าที่ที่สำคัญอย่างหนึ่งคือ การป้องกันแพ็กเก็ตที่มีการ Forged IP Address เข้ามา โดยอ้างว่ามาจากเครือข่ายภายในทั้งๆ ที่จริงๆ แล้วมาจากเครือข่ายภายนอก

2.4 การใช้งานไอพี ไฟร์วอลล์ (IP FIREWALL : IPFW)

IPFirewall (IPFW) เป็นไฟร์วอลล์ชนิดกรองแพ็คเก็ตและมีระบบการบันทึกจำนวน (accounting system) ซึ่งเป็นซอฟแวร์ที่มาพร้อมกับFreeBSD และ อยู่ภายในเคอเนลของ FreeBSD โดย IPFW อนุญาตให้สามารถกำหนดกฎแล้วนำข้อมูลมาเปรียบเทียบกับกฎที่กำหนดไว้เพื่ออนุญาตให้แพ็คเก็ตใดผ่านได้บ้างและแพ็คเก็ตใดควรจะทิ้งไป

IPFW แบ่งออกเป็น 2 ส่วนคือ ส่วนแรกเป็นส่วนของไฟร์วอลล์ (the firewall section) จะทำหน้าที่ในการกรองแพ็คเก็ตและอีกส่วนหนึ่งเป็นส่วนของการบันทึกข้อมูล IP (the IP accounting section) ซึ่งจะทำหน้าที่ตรวจดูการทำงานของเราเตอร์ โดยดูจากจำนวนการจราจรที่เราเตอร์ได้รับภายใต้เงื่อนไขหรือกฎที่คล้ายกันกับในส่วนของไฟร์วอลล์ ด้วยเหตุนี้จากคุณสมบัติของ IPFW จึงทำให้สามารถใช้ IPFW บนอุปกรณ์ชนิดที่เป็น non-router machines เพื่อใช้ในการกรองแพ็คเก็ตที่มีการติดต่อเข้ามา (incoming connections)และติดต่อออกไป (outgoing connections) ซึ่งกรณีนี้ถือเป็นกรณีพิเศษจากกรณีการใช้ IPFW ทั่วไป แต่อย่างไรก็ตามเทคนิคการใช้งานและการใช้คำสั่งก็ยังคงเหมือนกัน

2.4.1 IPFW บน FreeBSD

การเรียกใช้งาน IPFW บนเคอเนลของ FreeBSD ขึ้นอยู่กับสิ่งที่ต้องการให้ IPFW ทำ แต่สิ่งที่สำคัญคือ การกำหนดค่าบนไฟล์การติดตั้งจำเป็นจะต้องเพิ่มทางเลือก (option) อย่างน้อย 1 ทางเลือกหรือมากกว่านั้น พร้อมทั้งคอมไพล์คอนฟิกไฟล์นั้นด้วย เนื่องจาก IPFW มีนโยบายที่เป็นค่าพื้นฐานของเงื่อนไขอยู่ที่ว่า จะทำการ DENY IP ซึ่งมาจากที่ใดก็ตามและจะไปยังที่ใดก็ตามทั้งหมด (Deny IP from any to any) ดังนั้นถ้าไม่ได้ทำการติดตั้งกฎอื่นเข้าไปก็จะทำให้เกิดกรณีการล็อกตัวเองขึ้นในเคอเนลของ FreeBSD ข้อเสนอแนะในการติดตั้งระบบก็คือ

ควรกำหนดค่าของ firewall_type = open ในไฟล์ /etc/rc.conf

ค่าที่กำหนดนี้จะเป็นการอนุญาตให้ IP ใดก็ตามสามารถเข้ามาได้ ซึ่งเมื่อคุณสมบัตินี้เริ่มทำงานก็จะทำให้ script ของกฎที่อยู่ภายใน /etc/rc.firewall ถูกเรียกเพื่อติดตั้งค่ากลุ่มของกฎหลักๆ ของไฟร์วอลล์ อีกทางเลือกหนึ่งของการอนุญาต (allow) ให้ IP จากที่ใดก็ตามถึงที่ใดก็ตามได้ คือ การกำหนดค่าภายในเคอเนลคอนฟิกไฟล์ของไฟร์วอลล์ โดยการใช้คำสั่งระหว่าง IPFIREWALL

และ **IPFIREWALL _DEFAULT_ TO_ ACCEPT** ซึ่งทำให้สามารถหลีกเลี่ยงปัญหาการล็อกตัวเองได้ นอกจากคำสั่งเหล่านี้ยังมีคำสั่งที่สามารถใช้ภายในเคอเนลคอนฟิกไฟล์ได้อีก คือ

options IPFIREWALL

ใช้สำหรับคอมไพล์เพื่อแปลงเป็นเคอเนลโค๊ดสำหรับกรองแพ็คเก็ต

options IPFIREWALL_VERBOSE

เปิดให้สามารถเก็บ log ของแพ็คเก็ตผ่าน syslogd(8) ซึ่งถ้าไม่มี option นี้แม้ว่าจะระบุว่าให้เก็บ log ไว้ใน filter rules ก็จะไม่สามารถเก็บlogได้

options IPFIREWALL_VERBOSE_LIMIT = 10

จำกัดจำนวน log ของแพ็คเก็ต ผ่าน syslogd(8) ต่อแต่ละครั้งของการเข้ามา ซึ่งอาจจะคิดที่จะใช้ option นี้ในการเก็บ log ที่อยู่ในสภาวะที่มีการโจมตีแต่ไม่ต้องการเปิดการใช้งานในส่วนของ DoS attack via syslog flooding ได้

เมื่อการเข้าออกแพ็คเก็ต (chain entry) ถึงลิมิตที่มีการระบุไว้ log จะทำการ turn off เพื่อที่จะทำให้สามารถกลับมาใช้งาน ดังนั้นจะต้องทำการ reset การนับโดยการใช้คำสั่ง ipfw(8)

```
# ipfw zero 4500
```

หมายเหตุ 4500 เป็นค่าการเข้าออกที่ต้องการใน log ทำต่อไป

options IPFIREWALL_DEFAULT_TO_ACCEPT

เป็นการเปลี่ยนค่ากฎพื้นฐานจาก **"deny"** เป็น **"allow"** เพื่อหลีกเลี่ยงโอกาสที่จะเกิดการล็อกตัวเองเมื่อมีการบูท(boot) เคอเนลด้วย IPFIREWALL แต่ไม่มีการกำหนดกฎหรือเงื่อนไขไฟร์วอลล์

ข้อสังเกต เวอร์ชันก่อนหน้านี้ของ FreeBSD อาจจะยังคงมีการใช้คำสั่ง **IPFIREWALL_ACCT** แต่ในเวอร์ชันปัจจุบัน คำสั่งนี้ได้เลิกใช้แล้วเพราะ firewall code จะ Automatic ที่จะเพิ่มในส่วนคุณสมบัติ accounting จึงได้ยกเลิกคำสั่งนี้

2.4.2 การสร้างกฎสำหรับไฟร์วอลล์

เนื่องจากไฟร์วอลล์มีหน้าที่หลักในการกรอง (filter) ข้อมูลเฉพาะส่วนที่ได้รับอนุญาตเท่านั้น ดังนั้นการเขียนกฎหรือ rule สำหรับไฟร์วอลล์จึงเป็นเรื่องที่สำคัญอย่างยิ่ง การสร้างกฎของไฟร์วอลล์ที่ผิดพลาดจะทำให้ไฟร์วอลล์ (ทั้งราคาแพงและใช้งานฟรี) ทั้งหลายไม่สามารถช่วยป้องกันเครือข่ายให้รอดพ้นจากการถูกบุกรุกหรือโจมตีได้อย่างแน่นอน แต่ก่อนอื่นผู้ดูแลไฟร์วอลล์จะต้องมั่นใจว่าเครื่องไฟร์วอลล์นั้นมีความปลอดภัยในระดับโฮสต์อยู่แล้ว (host based security)

เพราะถึงแม้ว่า กฎที่สร้างขึ้นจะสามารถป้องกันเครื่องอื่นๆภายในเครือข่ายได้แต่ถ้าเครื่องไฟร์วอลล์เองไม่สามารถทนต่อการบุกรุกได้ก็เป็นจุดที่อันตรายไม่ยิ่งหย่อนไปกว่า การตั้งกฎที่ผิดพลาดไว้แต่อย่างใด

คำแนะนำเบื้องต้นสำหรับเครื่องที่ทำหน้าที่เป็นไฟร์วอลล์

- ปิด TCP/UDP service ที่ไม่ได้ใช้งาน เช่น bootps, finger ยิ่งเปิด service น้อยก็ยิ่งลดโอกาสในการโจมตีของผู้บุกรุก และยังเป็นการลดการใช้งาน CPU และหน่วยความจำของระบบอีกด้วย
- ในกรณีที่จำเป็นต้องเปิดบริการบนเครื่องไฟร์วอลล์ จะต้องจำกัดการเข้าถึงให้ใช้งานได้เฉพาะผู้ดูแลระบบเท่านั้น
- ปิด บริการที่ไม่จำเป็นอื่นๆ บนเครื่องไฟร์วอลล์ เช่น การทำ remote configuration
- ยกเลิก interface ที่ไม่ได้ใช้งานในเครื่องไฟร์วอลล์(หรือ router)
- ในกรณีที่ใช้ฮาร์ดแวร์ไฟร์วอลล์หรือ router จะต้องป้องกันการเข้าถึง port ที่ใช้ในการควบคุม เช่น console port
- แก้ไขค่า default password โดยให้มีความยาวอย่างต่ำ 8 ตัวอักษรและไม่เป็นคำที่อยู่ในดิกชันนารีรวมทั้งต้องไม่ขึ้นต้นด้วยตัวเลข และมีตัวเลขรวมทั้งตัวอักษรพิเศษรวมอยู่ด้วย ตัวอย่างรูปแบบตัวอักษรพิเศษ เช่น , . / < > ; ' : " [ ] { } \ | ~ ! @ # $ % ^ & * ( ) _ + - = เป็นต้น และควรใช้รหัสผ่านที่แตกต่างกันในแต่ละเครื่อง ทั้งนี้ควรเปลี่ยนรหัสผ่านทุกๆ 90 วัน

หลักการในการสร้างกฎสำหรับไฟร์วอลล์

หลักการในการสร้างกฎสำหรับไฟร์วอลล์ที่ดีคือ ความง่าย (Simplicity) ซึ่งความง่ายในที่นี้หมายถึงการสร้างกฎที่สั้นๆ อ่านง่าย ได้ใจความ ไฟร์วอลล์ที่ดีไม่ควรมีกฎมากกว่า 30 กฎ เพราะถ้ามากกว่านี้จะทำให้เกิดความสับสนได้ง่าย และอาจจะทำให้เกิดความผิดพลาดขึ้นได้โดยง่ายนอกจากนี้ยังมีข้อดีในส่วนที่ทำให้เครื่องทำงานน้อยลงอีกด้วย

การสร้างกฎของไฟร์วอลล์ถือได้ว่าเป็นการนำนโยบายด้านความปลอดภัย ขององค์กรมาบังคับใช้งานในทางเทคนิค โดยใช้ไฟร์วอลล์เป็นเครื่องมือให้เกิดผลตามที่ต้องการ นอกจากนี้ยังมีกฎบางส่วนที่ถือได้ว่า ผู้ดูแลระบบควรเพิ่มเข้าไปในกฎของไฟร์วอลล์เช่น การป้องกัน ip spoofing, การป้องกันการโจมตีแบบ land attack เป็นต้น

ลำดับของกฎของไฟร์วอลล์

การเรียงลำดับของกฎก็มีความสำคัญเช่นเดียวกัน เพราะไฟร์วอลล์โดยส่วนใหญ่ทำงานแบบลำดับ (Sequence) คือ ตรวจสอบ packet กับ กฎตามลำดับของกฎที่สร้างไว้

คำแนะนำในการวางลำดับของกฎคือ ให้วางกฎที่เป็นกฎทั่วไปไว้ด้านล่าง และให้นำ กฎที่มีความเฉพาะเจาะจงมาไว้ด้านบน เพื่อป้องกันไม่ให้ packet match กับ กฎที่เป็น กฎทั่วไปก่อนยกตัวอย่างเช่น ให้นำ กฎที่ทำหน้าที่ block ไอพีแอดเดรสไปไว้ด้านบนเพื่อให้มั่นใจว่า ถ้ามี packet ที่มีไอพีแอดเดรสตรงตามที่ระบุไว้ packet นั้นจะถูก drop ทิ้งไปก่อนที่จะมีการจับคู่ (match) กับกฎอื่น

การกรอง TCP/IP

ผู้ดูแลไฟร์วอลล์สามารถกำหนด default policy ได้ 2 รูปแบบคือ

- default ACCEPT : ผู้ดูแลไฟร์วอลล์จะต้องสร้างกฎเพื่อกำหนดว่าจะปิดบริการ(service) และโฮสต์ใดบ้าง โดยบริการและโฮสต์อื่นๆ ที่ไม่ถูกกำหนดไว้ จะมีค่าเป็นเปิดให้ผ่านได้
- default DROP : ผู้ดูแลไฟร์วอลล์จะต้องสร้าง กฎเพื่อกำหนดว่าจะเปิดบริการ(service) และโฮสต์ใดบ้าง โดยบริการและโฮสต์อื่นๆ ที่ไม่ถูกกำหนดไว้ จะมีค่าเป็นปิดไม่ให้ผ่าน

อย่างไรก็ตาม ไม่ว่าจะกำหนดนโยบายเริ่มต้นในรูปแบบใด ผู้ดูแลระบบไฟร์วอลล์ก็ควรทราบ TCP/IP service ที่เป็นจุดอ่อนต่างๆ ในระบบ ดังนี้

ตารางที่ 2.2 แสดง TCP/UDP service ที่ควรปิดกั้นที่ไฟร์วอลล์ โดยไม่ให้ใช้ทั้งจากภายในและภายนอกเครือข่าย

Port(s) (Transport)	**Server**	**Port(s) (Transport)**	**Server**
1 (TCP & UDP)	tcpmux	1981 (TCP)	Shockrave
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

ตารางที่ 2.2 แสดง TCP/UDP service ที่ควรปิดกั้นที่ไฟร์วอลล์ โดยไม่ให้ใช้ทั้งจากภายในและภายนอกเครือข่าย (ต่อ)

Port(s) (Transport)	**Server**	**Port(s) (Transport)**	**Server**
19 (TCP & UDP)	chargen	2222 (TCP)	Subseven21
37 (TCP & UDP)	time	2301 (TCP & UDP)	compaqdiag
43 (TCP & UDP)	whois	2565 (TCP)	Striker
67 (TCP & UDP)	bootps	2583 (TCP)	WinCrash
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.
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

ตารางที่ 2.2 แสดง TCP/UDP service ที่ควรปิดกั้นที่ไฟร์วอลล์ โดยไม่ให้ใช้ทั้งจากภายในและภายนอกเครือข่าย (ต่อ)

Port(s) (Transport)	Server	Port(s) (Transport)	Server
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
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 2.3 แสดง TCP/UDP service 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 2.4 แสดง ICMP message ที่ควรอนุญาตให้ออกไปจากเครือข่ายภายในได้

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

ตารางที่ 2.5 แสดง ICMP message ที่ควรอนุญาตให้เข้ามายังเครือข่ายภายในได้

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

ตารางที่ 2.6 แสดง TCP/UDP service ที่อาจเปิดให้บริการใน DMZ (ในทางปฏิบัติให้เปิดเฉพาะ service ที่มีการให้บริการจริงเท่านั้น)

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

คำแนะนำอื่นๆ สำหรับการสร้างกฎของไฟร์วอลล์

- ควรมีการบันทึกข้อมูลลงล็อกสำหรับกฎที่ใช้ block การเข้าถึง ซึ่งข้อมูลนี้จะเป็นประโยชน์ในการตรวจสอบการบุกรุก รายละเอียดของล็อกจะพบใน Logging and Debugging
- 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 - 172.31.255.255 : 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 ที่มี source ip address ตรงกันกับ destination ip address รวมทั้งค่า source port และ destination port ที่ตรงกัน ซึ่งก่อให้เกิดการโจมตีแบบ Denial of Service ได้ ซึ่งป้องกันได้โดย block ไม่ให้ข้อมูลขาเข้าที่มี source ip address ตรงกันกับไอพีของเครือข่ายภายในเข้ามาในระบบ
- ป้องกันการโจมตีแบบ SYN flood ที่เครื่องไฟร์วอลล์ ซึ่งผู้บุกรุกจะส่ง SYN packet จำนวนมากมายังเครื่องปลายทาง ทำให้คิวของการรับ connection ใน service ดังกล่าวเต็ม ทำให้ไม่สามารถให้บริการแก่เครื่องอื่นๆ ได้
- เครื่องไฟร์วอลล์และเครื่องอื่นๆ ภายในเครือข่ายควรได้รับป้องกันจาก ICMP message บางชนิด เช่น ป้องกันการรับ ICMP Echo request ซึ่งสามารถส่งมาเพื่อรวบรวมข้อมูลสำหรับการโจมตีครั้งต่อไป หรือการส่ง ICMP Echo request packet ที่มีขนาดใหญ่ (Ping flood) ซึ่งถือว่าเป็นรูปแบบหนึ่งในการโจมตี นอกจากนี้ redirect packet ที่ส่งมาจากภายนอกยังสามารถเปลี่ยน routing table ใน host ได้อีกด้วย ซึ่งเป็นเรื่องที่อันตรายอย่างยิ่ง

สำหรับข้อมูลขาออกนั้น ควรอนุญาตให้ข้อมูล ICMP ดังต่อไปนี้เท่านั้นที่สามารถออกไปได้

- Echo request
- Parameter Problem
- Source Quench

สำหรับข้อมูลขาเข้านั้น ควรอนุญาตให้ข้อมูล ICMP ดังต่อไปนี้เท่านั้นที่สามารถเข้ามาภายในได้

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

- ป้องกันไฟร์วอลล์และเครื่องอื่นๆ ภายในเครือข่ายจาก traceroute เพราะ traceroute เป็นโปรแกรมที่ช่วยให้ทราบถึงไอพีแอดเดรสของ router ที่รับส่งต่อ packet ไปทีละ hop จนกระทั่งถึงปลายทางที่ต้องการ โดยใช้คุณสมบัติของ IP Time To Live (TTL) ในการทำงาน โดยมันจะกำหนดค่า TTL counter ที่ทำให้ router ที่ packet ผ่านไปนั้นต้องสร้าง ICMP message กลับมาเสมอ สำหรับคำสั่ง tracert ใน Windows นั้น จะใช้ ping (ICMP

Echo) เป็นตัวส่ง packet ออกไป ในขณะที่ traceroute ใน Unix นั้น จะใช้ 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 ดังนั้นหากต้องการหยุดยั้งการใช้ traceroute รวมทั้ง Firewalk แล้ว จะต้องใช้วิธี drop TTL Exceeded ใน Transit packet ที่ขาออกไปสู่อินเทอร์เน็ต

- จำกัดการเข้าถึงเครื่องไฟร์วอลล์ โดยให้ใช้งานในบริการที่จำเป็นเท่านั้น(สำหรับผู้ดูแลระบบเท่านั้น) และให้บันทึกข้อมูลล็อกสำหรับทั้งการติดต่อ (connection) ที่สำเร็จและไม่สำเร็จ
- ถ้าหากมี SNMP server ทำงานอยู่บนเครื่องไฟร์วอลล์ จะต้องจำกัดการใช้งานให้ใช้เฉพาะผู้ดูแลระบบเท่านั้น และให้บันทึกข้อมูลล็อกสำหรับทั้งการติดต่อ (connection) ที่สำเร็จและไม่สำเร็จ

การเก็บข้อมูลล็อกและตรวจสอบจุดบกพร่อง

การเก็บข้อมูลล็อกของเครื่องไฟร์วอลล์เป็นเรื่องที่จำเป็นอย่างยิ่ง โดยเฉพาะในกรณีที่เครื่องโดน Compromise ไปแล้ว จะถือว่าเป็นหลักฐานที่แสดงให้เห็นถึงรูปแบบการโจมตีได้ มีคำแนะนำสำหรับการบันทึกข้อมูลล็อกดังนี้

- ให้ส่งข้อมูลล็อกที่มีความสำคัญไปยัง console ของเครื่องไฟร์วอลล์
- ส่งข้อมูลล็อกไปยังเครื่องที่ทำหน้าที่เก็บล็อกโดยเฉพาะ ซึ่งเครื่องนี้ได้รับการควบคุมการเข้าถึงอย่างเคร่งครัด และไม่ได้เปิดให้บริการอื่นใดยกเว้น syslog
- ตั้งเวลาเครื่องไฟร์วอลล์และเครื่องอื่นๆ ในเครือข่ายให้ใช้เวลาที่ตรงกันทั้งหมด โดยใช้ NTP (network time protocol) เพื่อรับข้อมูลเวลาจาก clock server เดียวกัน
- ป้องกันการโจมตีแบบ log flooding ซึ่งจะทำให้ฮาร์ดดิสก์เต็มอย่างรวดเร็ว
- ไม่ควรส่งข้อมูลล็อกออกไปยังเครื่องพิมพ์โดยตรง เพราะอาจจะเสี่ยงต่อการสูญเสียข้อมูลในกรณีที่เครื่องพิมพ์มีปัญหา

2.4.3 การกำหนดค่า IPFW

การกำหนดค่าของ IPFW จะทำผ่านคุณสมบัติ ipfw(8) ซึ่งโครงสร้างของคำสั่งดูยุ่งยากซับซ้อนและจำแนกประเภทได้ 4 ประเภทด้วยกันคือ

1. **addition/deletion** ใช้เพื่อสร้างกฎที่ทำหน้าที่ควบคุมจำนวนแพ็คเก็ตที่จะยอมรับและไม่ยอมรับและล็อกเท่าไร
2. **listing** ใช้เพื่อตรวจสอบรายละเอียดของกฎที่สร้างและจำนวนของแพ็คเก็ตที่นับ
3. **flushing** ใช้เพื่อเอาข้อมูลรายละเอียดการเข้าออก ออกจากระบบ
4. **clearing** ใช้เพื่อลบรายละเอียดการนับทั้งหมดและกลับไปเริ่มที่ค่าศูนย์ใหม่อีกครั้ง

2.4.3.1 รูปแบบคำสั่งของ IPFW

รูปแบบคำสั่งของ IPFW คือ

ipfw [-N] command [index] action [log] protocol addresses [options]

ซึ่งมีความหมายแต่ละส่วนดังนี้

ในส่วนของ flag

-N หมายถึงให้ใช้ค่า address และ service name ใน output

ในส่วนของ command จะใช้เป็นคำสั่งสั้นๆ เช่น

Add หมายถึง การเพิ่มกฎที่เกี่ยวกับfirewallและ accounting เข้าไปใน list*

Delete หมายถึง การลบกฎที่เกี่ยวกับfirewall และ accounting ออกจากใน list*

*เวอร์ชันก่อนหน้านี้จะแยกการทำงานระหว่าง firewall และ accounting entry ออกจากกัน แต่เวอร์ชันในปัจจุบันแพ็คเก็ต accounting ได้รวมกับ firewall entry แล้ว

index ใช้เพื่อระบุตำแหน่งใน chain entry

ในส่วนของ action จะใช้คำสั่งดังนี้

reject เป็นการ drop แพ็คเก็ต และส่ง ICMP host หรือ port unreachable packet ถึง source

allow อนุญาตให้แพ็คเก็ตปกติผ่าน คำสั่งที่ใกล้เคียงเช่น pass, permit และ accept

deny เป็นการสั่งให้ drop แพ็คเก็ต เนื่องจาก source ใน ICMP Message ไม่มีความชัดเจน (ดังนั้นจึงไม่เคยพบว่าแพ็คเก็ตชนิดนี้ส่งถึง destination)

count ใช้ในการ update การนับจำนวนแพ็คเก็ต แต่ไม่สามารถ allow/deny แพ็คเก็ตได้

ในส่วนของ protocol สามารถระบุเป็น

all เพื่อใช้กับแพ็คเก็ตที่เป็น IP ทุกชนิด

icmp เพื่อใช้กับแพ็คเก็ตที่เป็น ICMP

tcp เพื่อใช้กับแพ็คเก็ตที่เป็น TCP

udp เพื่อใช้กับแพ็คเก็ตที่เป็น UDP

ในส่วนของ address ให้ระบุดังนี้

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

หมายเหตุ โดยสามารถระบุเพียง port ร่วมกับ protocols ซึ่งสนับสนุน port ทั้ง UDP และ TCP

การระบุ address/mask สามารถระบุเป็น

address

หรือ

address/mask-bits

หรือ

address:mask-pattern

เช่น 192.216.222.1/24

การระบุเบอร์ port ที่จะ block ระบุดังนี้

port [,port [,port [...]]]

และเพื่อระบุ single port หรือ list ของ port

port-port

ในส่วนของ option จะมีคำสั่งให้เลือกได้ดังนี้คือ

frag จะทำคำสั่งได้ถ้าแพ็คเก็ตไม่เป็น fragment แรกของ datagram

in จะทำคำสั่งถ้าแพ็คเก็ตอยู่ในเส้นทางขาเข้า

out จะทำคำสั่งได้ถ้าแพ็คเก็ตอยู่ในเส้นทางขาออก

ipoptions spec จะทำคำสั่งถ้า IP header มี comma แยก list ของ option ที่ระบุใน spec ซึ่ง IP option ที่สนับสนุนได้แก่ ssrr (strict source route), lsrr (loose source route), rr (record packet route), และ ts (time stamp) ถ้าไม่มีส่วนของ option สามารถระบุเป็น ! นำหน้าได้

established จะทำคำสั่งถ้าแพ็คเก็ตเป็นส่วนของ TCP Connection ที่เกิดขึ้น เช่น มีในส่วนของ RST หรือ ACK bit set

setup จะทำคำสั่งนี้เมื่อมีแพ็คเก็ตที่เกิดจากการพยายามทำ TCP Connection (กำหนด SYN bit แต่ ACK bit ไม่กำหนด)

tcpflags flags จะทำคำสั่งนี้เมื่อ TCP header มี comma แยกรายการของ flags ตัวอย่างของ flags ที่สนับสนุนคือ fin, syn, rst, psh, ack, และ urg สำหรับกรณีที่ไม่มีค่าในส่วนของ flag ให้ระบุ ! ขึ้นต้น

2.4.3.2 การแสดงรายการของกฎ (Listing the IPFW Rules)

รูปแบบของคำสั่งคือ

ipfw [-a] [-c] [-d] [-e] [-t] [-N] [-S] list

ซึ่งมีค่า flags ที่สามารถใช้ได้ดังนี้

-a ใช้สำหรับแสดงค่าของ accounting counters ขณะที่กำลัง listing

-c List rules ในรูปแบบที่กระชับ.

-d แสดง dynamic rules ที่นอกจาก static rules.

-e เป็นการให้ระบุ expired dynamic rules สำหรับกรณีที่มีการระบุ flag เป็น -d

-t แสดงเวลาสุดท้ายสำหรับแต่ละ chain entry. ซึ่ง list ของเวลานี้ไม่ตรงกันกับ input syntax ที่ใช้ใน ipfw(8) .

-N ให้ใช้ค่า addresses และ service names ที่กำหนด

-S แสดงค่าที่กำหนดของแต่ละกฎ ถ้าflag ไม่ได้ระบุ กฎที่ไม่ใช้จะไม่อยู่ใน list

2.4.3.3 คำสั่งที่ใช้ในการฟลัชค่าของ IPFW (Flushing the IPFW Rules)

รูปแบบของคำสั่งคือ

ipfw flush

คำสั่งนี้จะใช้ในการฟลัชค่าของ firewall chain ทั้งหมดทิ้งยกเว้นในส่วนของที่เป็นค่า default policy ของเคอเนล (index 65535) สิ่งที่ควรระมัดระวังคือ ค่า default deny policy ยังคงอยู่จนกว่าจะมีการเพิ่มค่า firewall chain เข้าไปในระบบ

2.4.3.4 การลบค่าการนับ IPFW Packet (IPFW Packet Counters)

รูปแบบคำสั่งที่ใช้ คือ

ipfw zero [*index*]

packet counters จะถูกลบทิ้งทั้งหมดเมื่อไม่มีการระบุค่า index แต่ถ้าได้รับการระบุค่าindex จะทำการลบเฉพาะที่ตำแหน่ง index นั้น

2.4.4 ตัวอย่างวิธีการใช้งานคำสั่ง

การใช้คำสั่งในการ Deny ทุก packets จาก evil.crackers.org ถึง port ที่ใช้ telnet ของ host nice.people.org:

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

ตัวอย่างถัดมาจะเป็นการ denies และ logs ทุกๆ traffic ของ TCP ที่มาจาก crackers.org network (a class C) ถึง เครื่อง

nice.people.org (ทุกๆ port).

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

ถ้าไม่ต้องการให้ใครส่ง X sessions ถึง เน็ตเวริค์ภายใน

(subnet of a class C) คำสั่งที่จำเป็นในการใช้คือ

 # ipfw add deny tcp from any to my.org/28 6000 setup

เพื่อใช้ในการดู accounting records:

 # ipfw -a list

หรือในรูปคำสั่งสั้นๆ

 # ipfw -a l

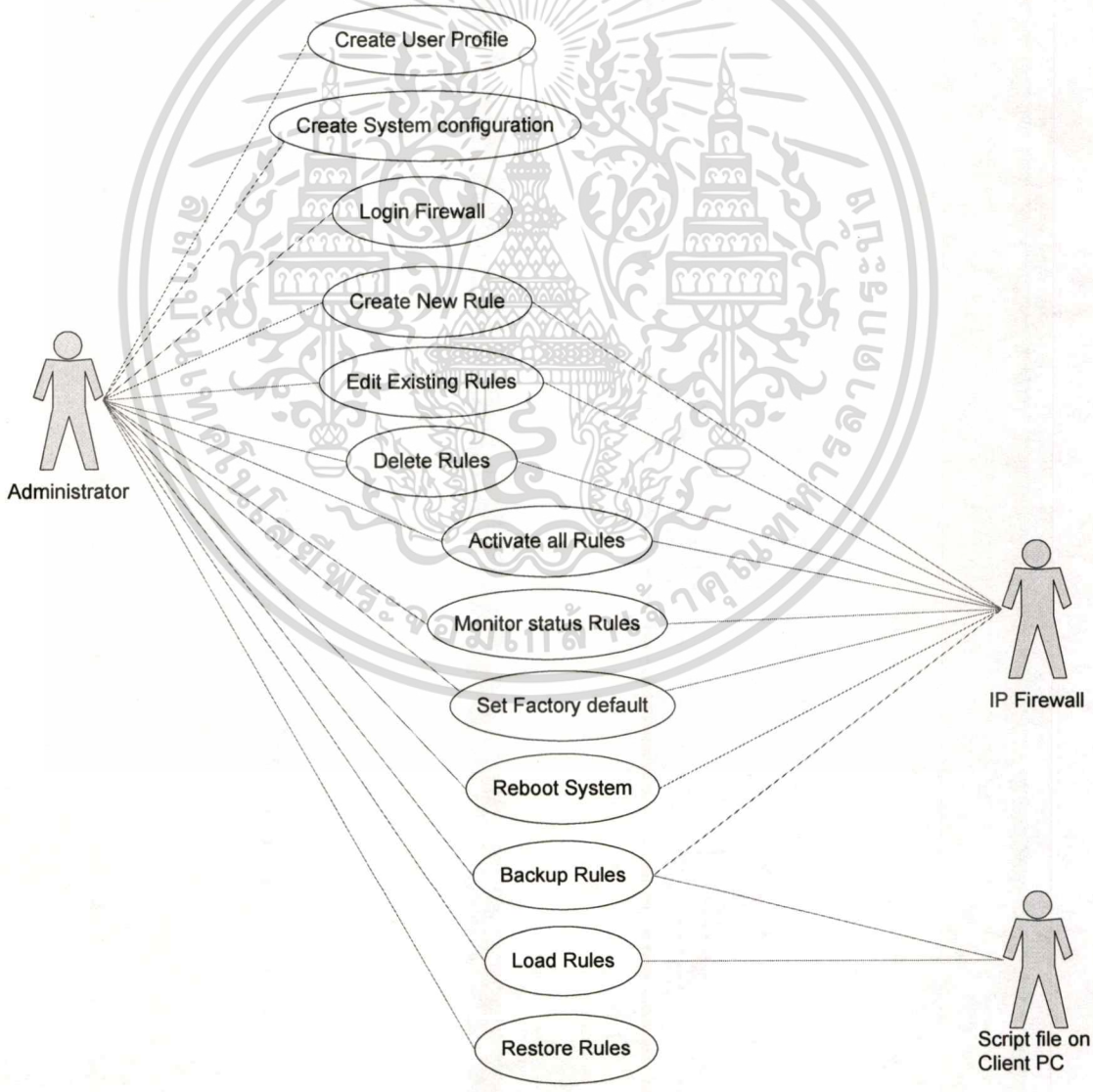
บทที่ 3

วิเคราะห์และออกแบบโปรแกรมการกำหนดกฎของไฟร์วอลล์

3.1 แบบจำลองเชิงแนวคิดของระบบ (Conceptual Models)

การวิเคราะห์ระบบการกำหนดกฎของไฟร์วอลล์ วิเคราะห์โดยอาศัย OMG-Unified Modeling Language (UML) ซึ่งมีแบบจำลองเชิงแนวคิดของระบบแบ่งออกเป็น 3 มุมมอง ดังนี้

3.1.1 Use Case Model ใช้ในการอธิบายระบบงานทั้งหมด



รูปที่ 3.1 แสดง Use Case Diagram ระบบการติดตั้งกฎไอพีไฟร์วอลล์

USE CASE DESCRIPTION

Use Case: Create System Configuration

Brief Description: จะเป็นการกำหนดค่าการใช้งานของระบบ โดยจะทำการบันทึกค่า Hostname, Domain, DNS Servers

Actor: Administrator

Precondition: ระบบได้มีการผ่านการ Login โดยใช้ default login ของระบบซึ่งสิทธิ์การใช้เป็น Admin

Basic Flows:

1. Administrator เรียก Web page ขึ้นเพื่อทำการป้อนข้อมูล Hostname, Domain, DNS Servers
2. Person ทำการบันทึกข้อมูลเข้าสู่ระบบ

Alternative Flow: -

Postcondition: -

Use Case: Create User Profile

Brief Description: จะทำการสร้าง Username และ Password รวมถึงรายละเอียดเกี่ยวกับผู้ใช้

Actor: Administrator

Precondition: ผู้ใช้งานจะต้องทำการ Login เข้าสู่ระบบและได้รับสิทธิ์การใช้งานเป็น Admin

Basic Flows:

1. Administrator เรียก Web page ขึ้นเพื่อทำการสร้างข้อมูล รายละเอียดเกี่ยวกับ Username และ Password ของผู้ที่จะมีสิทธิ์เข้าใช้ระบบ
2. Person ทำการบันทึกข้อมูลเข้าสู่ระบบ

Alternative Flow: -

Postcondition: -

Use Case: Login Firewall

Brief Description: จะทำการตรวจสอบ Username และ Password ว่ามีอยู่ในระบบหรือไม่

Actor: Administrator

Precondition: ผู้ใช้งานจะต้องได้รับการเพิ่มชื่อให้เป็นผู้ใช้งานระบบและได้รับสิทธิ์การใช้เป็น Admin

Basic Flows:

1. Administrator เรียก Web page ขึ้นเพื่อทำการ Login เพื่อเข้าใช้งานระบบ
2. ทำการป้อน Username และ Password ของผู้จะเข้าใช้ระบบ
3. ระบบทำการตรวจสอบความถูกต้องของข้อมูล
4. ข้อมูลถูกต้องทำการเข้าใช้งานส่วนต่างๆของระบบได้

Alternative Flow:

3a. ระบบทำการตรวจสอบแล้วหากไม่พบข้อมูลจะทำการแสดงข้อความผิดพลาดขึ้นและกลับเข้าสู่การ Login ใหม่อีกครั้ง

Postcondition: -

Use Case: Create New Rule

Brief Description: จะทำการสร้างกฎต่างๆที่จะใช้ในการทำงานของไฟร์วอลล์

Actor: Administrator

Precondition: ผู้ใช้งานจะต้องทำการ Login เข้าสู่ระบบและได้รับสิทธิ์การใช้งานเป็น Admin

Basic Flows:

1. Administrator เรียก Web page ขึ้นเพื่อทำการสร้างกฎของไฟร์วอลล์โดยป้อนข้อมูลที่จำเป็น เช่น IP Source, Port Source, IP Destination, Port Destination, Protocol, Direction, Interface
2. ตรวจสอบกฎที่จะทำการเพิ่มว่าถูกต้องตามหลักการสร้างกฎ
3. ทำการเพิ่มกฎและบันทึกกฎที่เพิ่มเข้าสู่ระบบ
4. ระบบแจ้งผลการเพิ่มกฎ

Alternative Flow:

2a. หากกฎที่ทำการสร้างไม่ถูกต้อง ระบบจะแสดงข้อความผิดพลาดขึ้นและไม่ทำการบันทึกผล

Postcondition:

1. กฎที่ทำการสร้างจะถูกบันทึกไว้ในระบบเพื่อรอการ Activate การทำงานจึงจะเกิดการทำงานตามกฎที่ได้สร้าง

Use Case: Edit Existing Rules

Brief Description: จะทำการแก้ไขกฎต่างๆที่ได้มีการสร้างไว้ทั้งที่สร้างใหม่และที่มีอยู่เดิม

Actor: Administrator

Precondition: ผู้ใช้งานจะต้องทำการ Login เข้าสู่ระบบและได้รับสิทธิ์การใช้งานเป็น Admin

Basic Flows:

1. Administrator เรียก Web page ขึ้นเพื่อทำการเรียกดูกฎทั้งหมดที่ถูกสร้าง
2. เลือกกฎที่จะทำการปรับปรุง
3. ทำการปรับปรุงและแก้ไขกฎไฟร์วอลล์
4. ตรวจสอบกฎที่ทำการแก้ไขว่าถูกต้องตามหลักการสร้างกฎ
5. ทำการบันทึกกฎที่ได้รับการแก้ไขเข้าสู่ระบบ
6. ระบบแจ้งผลการบันทึกกฎที่ได้รับการแก้ไขใหม่

Alternative Flow:

4a. หากกฎที่ทำการสร้างไม่ถูกต้อง ระบบจะแสดงข้อความผิดพลาดขึ้นและไม่ทำการบันทึกผล

Postcondition:

1. กฎที่ทำการแก้จะถูกบันทึกไว้ในระบบเพื่อรอการ Activate การทำงานจึงจะเกิดการทำงาน

Use Case: Delete Rules

Brief Description: จะทำการลบกฎต่างๆที่ได้มีการสร้างไว้ทั้งที่สร้างใหม่และที่มีอยู่เดิม

Actor: Administrator

Precondition: ผู้ใช้งานจะต้องทำการ Login เข้าสู่ระบบและได้รับสิทธิ์การใช้งานเป็น Admin

Basic Flows:

1. Administrator เรียก Web page ขึ้นเพื่อทำการเรียกดูกฎทั้งหมดที่ถูกสร้าง
2. เลือกกฎที่จะทำการลบ
3. ตรวจสอบกฎที่จะทำการลบว่าสามารถทำการลบได้
4. ทำการลบกฎและบันทึกผลการลบเข้าสู่ระบบ
5. ระบบแจ้งผลการลบ

Alternative Flow:

3a. หากกฎที่ทำการลบเป็นกฎพื้นฐานของระบบซึ่งไม่สามารถูกลบได้ ระบบจะแสดงข้อความผิดพลาดขึ้นและไม่ทำการบันทึกผลการลบ

Postcondition:

1. กฎที่ทำการลบจะถูกบันทึกไว้ในระบบเพื่อรอการ Activate การทำงานจึงจะเกิดการทำงาน

Use Case: Activate all Rules

Brief Description: จะเปิดการทำงานของระบบไฟร์วอลล์ให้ทำงานตามกฎที่ได้ถูกสร้างหรือแก้ไขไว้

Actor: Administrator

Precondition: ผู้ใช้งานจะต้องทำการ Login เข้าสู่ระบบและได้รับสิทธิ์การใช้งานเป็น Admin

Basic Flows:

1. Administrator เรียก Web page ขึ้นเพื่อทำการเปิดการใช้งานไฟร์วอลล์(Activate)
2. ระบบแจ้งผลการเปิดการใช้งานระบบไฟร์วอลล์

Alternative Flow: -

Postcondition:

1. หลังจากได้มีการเลือก Activate เพื่อเปิดการทำงานระบบจะทำการสร้าง script rules และสร้างฐานข้อมูลกฎที่ถูกใช้งานจริงขึ้น

Use Case: Monitor status Rules

Brief Description: จะทำการเรียกดูการทำงานของกฎพร้อมเวลาล่าสุดที่กฎนั้นทำงาน โดยจะแสดงจำนวน packets เข้า-ออก ที่ตรงกับกฎ และสามารถ reset ค่าการนับจำนวนpacket ทั้งหมดที่ผ่านเข้าออกตามกฎ หรือ reset ค่าการนับจำนวนpacket เฉพาะกฎบางกฎ

Actor: Administrator

Precondition: ผู้ใช้งานจะต้องทำการ Login เข้าสู่ระบบและได้รับสิทธิ์การใช้งานเป็น Admin

Basic Flows:

1. Administrator เรียก Web page ขึ้นเพื่อทำการเรียกดูการทำงานของกฎทั้งหมด

Alternative Flow:

1a. สามารถ reset ค่าการนับจำนวนpacket ทั้งหมดที่ผ่านเข้าออกตามกฎ

1b. สามารถ reset ค่าการนับจำนวนpacket เฉพาะกฎบางกฎ

Postcondition: -

Use Case: Set Factory default

Brief Description: จะทำการลบข้อมูลทั้งหมดและกลับสู่ค่าเริ่มต้นของระบบเพื่อรอการกำหนดค่าเริ่มต้นของระบบใหม่

Actor: Administrator

Precondition: ผู้ใช้งานจะต้องทำการ Login เข้าสู่ระบบและได้รับสิทธิ์การใช้งานเป็น Admin

Basic Flows:

1. Administrator เรียก Web page ขึ้นเพื่อทำการเลือกการทำงานในโหมด Set Factory default
2. ระบบแสดงหน้าจอยืนยันการเลือกใช้งาน

3. ระบบทำการลบค่าการทำงานทั้งหมดและกลับสู่ค่าเริ่มต้นของระบบ

Alternative Flow: -

Postcondition:

1. ระบบจะกลับเข้าสู่ค่า default เริ่มต้นของระบบ โดยข้อมูลที่เคยได้รับการกำหนดค่าไว้จะถูกทำการลบค่าออกจากระบบทั้งหมด

Use Case: Reboot System

Brief Description: จะทำการ reboot ไฟร์วอลล์

Actor: Administrator

Precondition: ผู้ใช้งานจะต้องทำการ Login เข้าสู่ระบบและได้รับสิทธิ์การใช้งานเป็น Admin

Basic Flows:

2 Administrator เรียก Web page ขึ้นเพื่อทำการเลือกการทำงานในโหมด Reboot System

3 ระบบแสดงหน้าจอยืนยันการเลือกใช้งาน

4 ระบบทำการ Reboot System

Alternative Flow: -

Postcondition: -

Use Case: Backup Rules

Brief Description: จะทำการ Backup กฎของไฟร์วอลล์ทั้งหมดออกมาเป็น text

Actor: Administrator

Precondition: ผู้ใช้งานจะต้องทำการ Login เข้าสู่ระบบและได้รับสิทธิ์การใช้งานเป็น Admin

Basic Flows:

1. Administrator เรียก Web page ขึ้นเพื่อทำการเลือกการทำงานในโหมด Backup Rules
2. เลือกไดเรกทอรี่ที่จะบันทึกผลการ backup ไฟล์
3. ระบบแสดงหน้าจอยืนยันการเลือกใช้งาน
4. ระบบทำการบันทึกข้อมูลลงไฟล์

Alternative Flow:

4a. หากระบบไม่สามารถบันทึกข้อมูลลงได้จะแสดงข้อความผิดพลาดขึ้น

Postcondition: -

Use Case: Load Rules

Brief Description: จะทำการ Load ไฟล์กฎของไฟร์วอลล์โดยมีรูปแบบตามที่กำหนดไว้

Actor: Administrator

Precondition: ผู้ใช้งานจะต้องทำการ Login เข้าสู่ระบบและได้รับสิทธิ์การใช้งานเป็น Admin

Basic Flows:

1. Administrator เรียก Web page ขึ้นเพื่อทำการเลือกการทำงานในโหมด Load Rules
2. เลือกไดเรกทอรี่ที่จะทำการ load ไฟล์
3. ระบบทำการ load ไฟล์และบันทึกลงฐานข้อมูลพร้อมทั้งลบค่ากฎที่เคยบันทึกไว้ก่อนหน้าการ load ทั้งหมด
4. ระบบแสดงผลการทำงาน

Alternative Flow:

4a. หากระบบไม่สามารถบันทึกข้อมูลลงได้จะแสดงข้อความผิดพลาดขึ้น

Postcondition:

1. กฎที่ได้ทำการ Load ขึ้นมาจะยังไม่สามารถทำงานได้ตามที่กฎระบุไว้ซึ่งจะต้องรอการเปิดการใช้งานไฟร์วอลล์(Activate) ก่อนกฎดังกล่าวจึงจะสามารถใช้ได้

Use Case: Restore Rules

Brief Description: จะเป็นการดึงข้อมูลกฎครั้งหลังสุดที่มีการทำงานก่อนที่จะมีการเปลี่ยนแปลงแก้ไขกลับขึ้นมาทำงานเหมือนเดิม

Actor: Administrator

Precondition: ผู้ใช้งานจะต้องทำการ Login เข้าสู่ระบบและได้รับสิทธิ์การใช้งานเป็น Admin

Basic Flows:

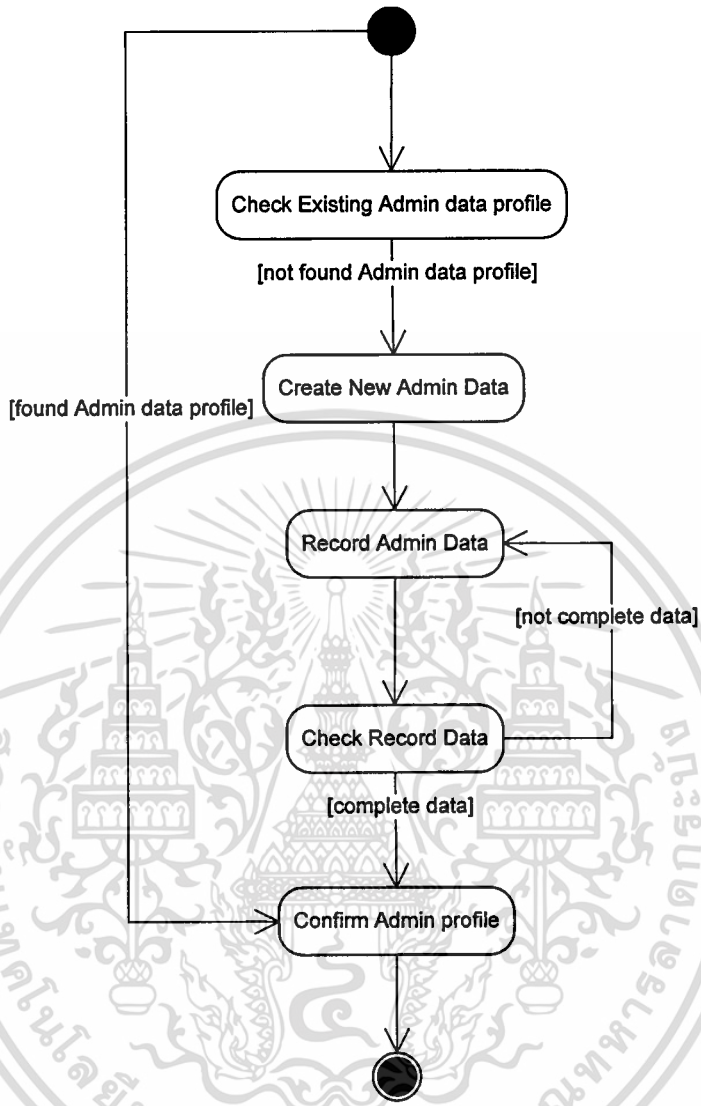
1. Administrator เรียก Web page ขึ้นเพื่อทำการเลือกการทำงานในโหมด Restore Rules
2. ยืนยันผลการเลือกใช้งานในโหมด Restore Rules
3. ระบบทำการรีเซ็ตค่ากฎการใช้งานกลับไปยังค่าก่อนหน้าที่จะมีการเปลี่ยนแปลงแก้ไขกลับขึ้นมาทำงานเหมือนเดิม

Alternative Flow:

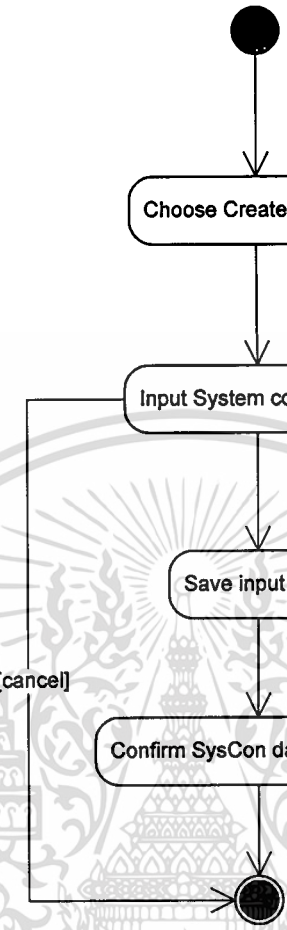
3a. หากระบบไม่สามารถดึงผลข้อมูลก่อนหน้าที่จะมีการเปลี่ยนแปลงแก้ไขกลับขึ้นมาทำงานได้ ระบบจะแสดงข้อความผิดพลาดขึ้น

Postcondition:

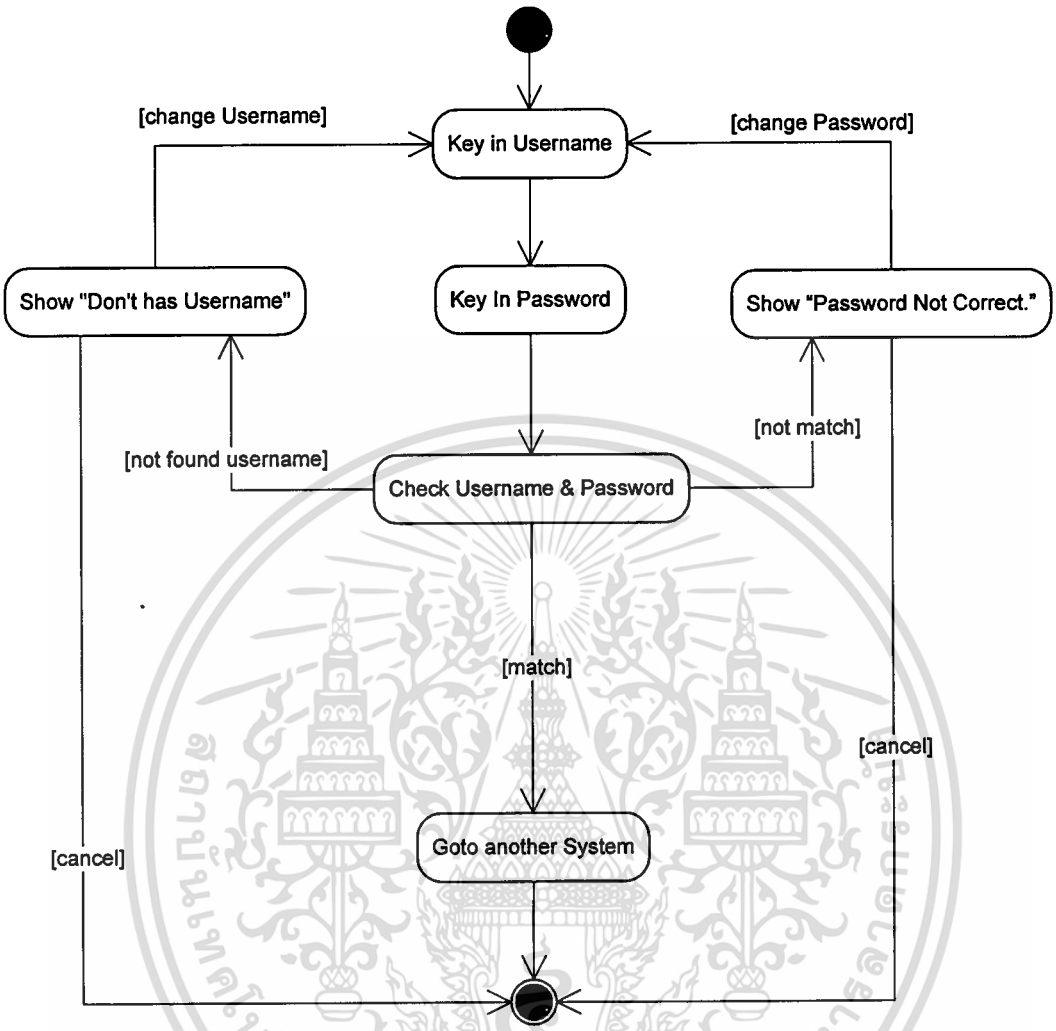
1. กฎที่ได้รับการเรียกกลับมาใช้งานจะเป็นกฎที่เคยมีการใช้งานมาแล้ว



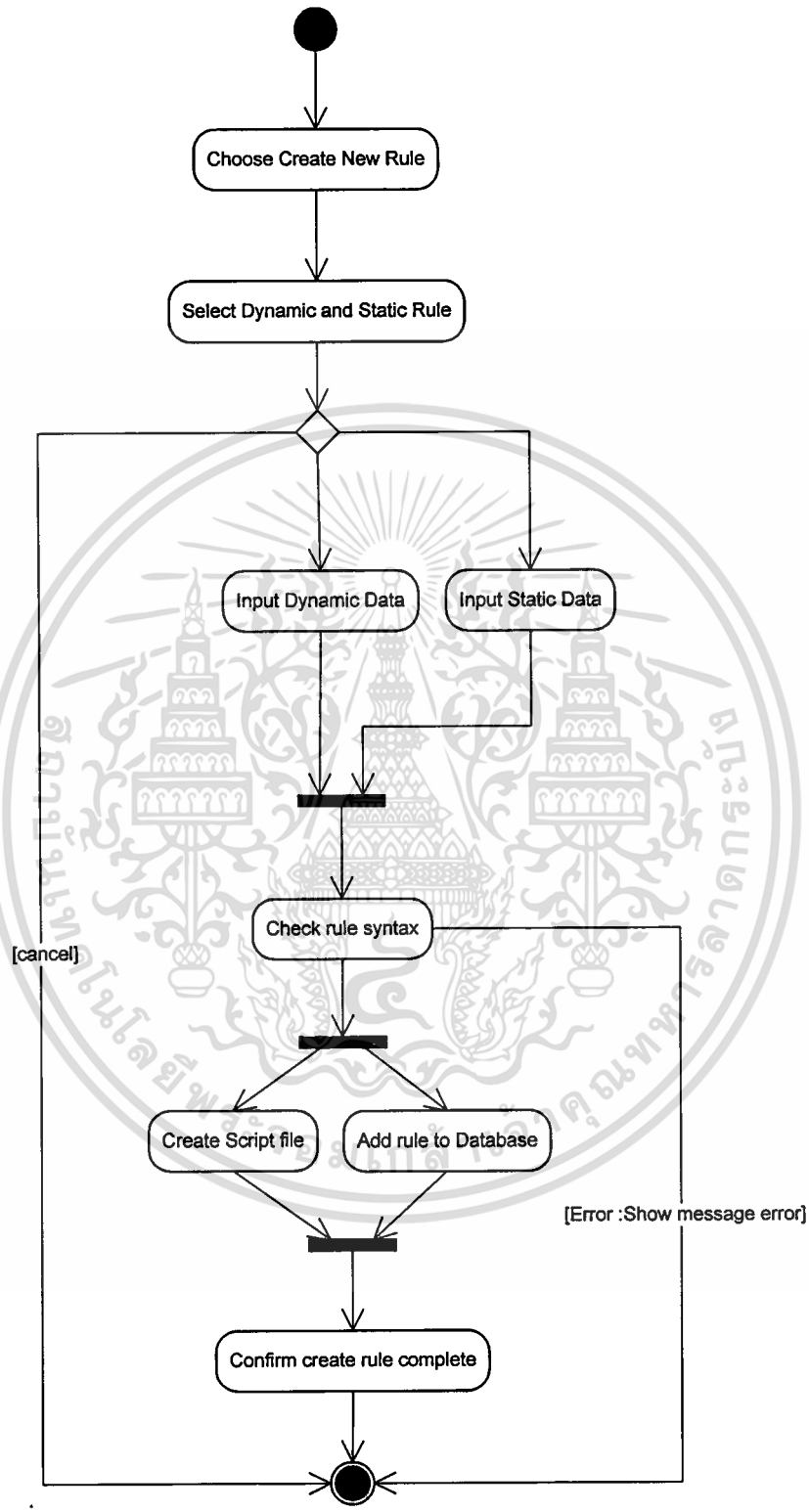
รูปที่ 3.2 แสดง Activity Diagram ของ Create user profile



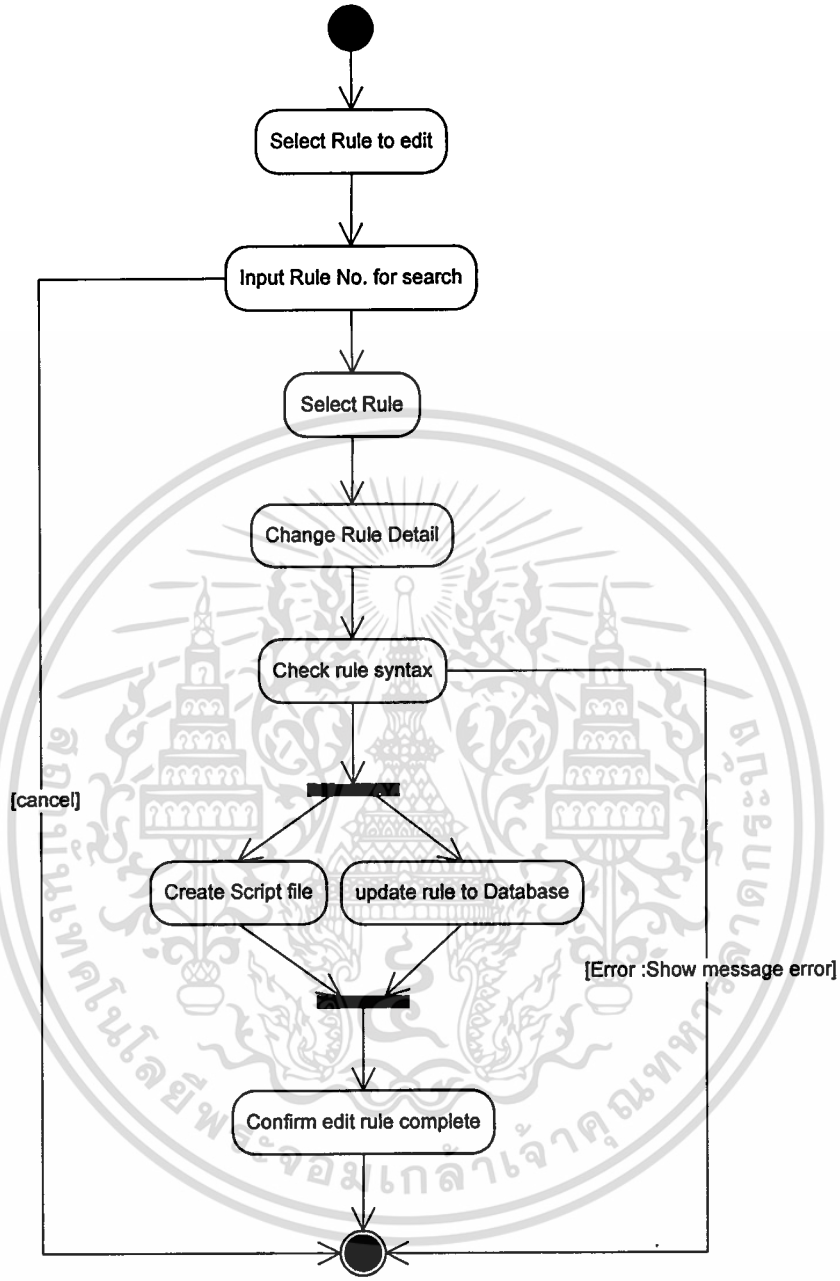
รูปที่ 3.3 แสดง Activity Diagram ของ Create System Configuration



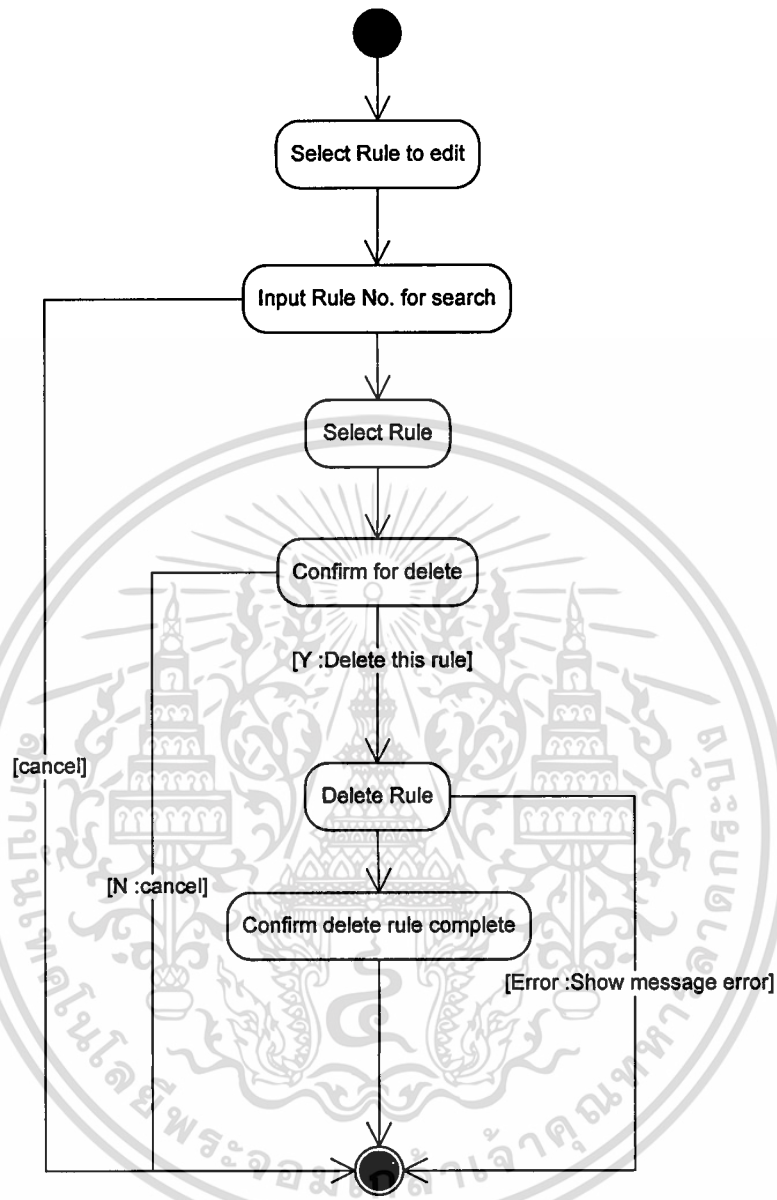
รูปที่ 3.4 แสดง Activity Diagram ของ Login Firewall



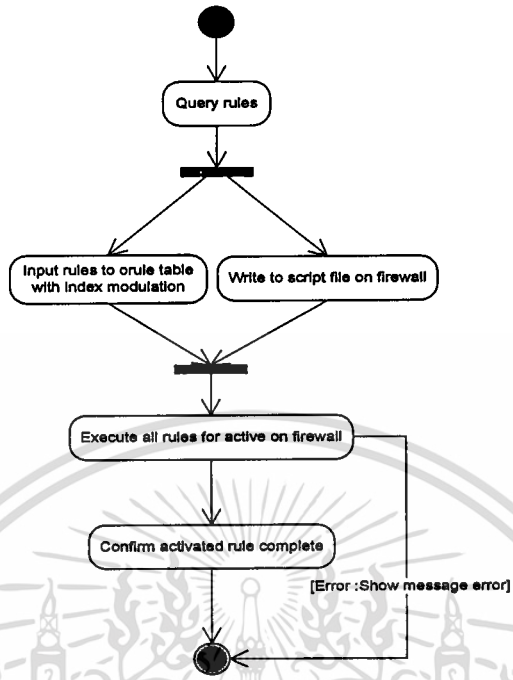
รูปที่ 3.5 แสดง Activity Diagram ของ Create New Rule



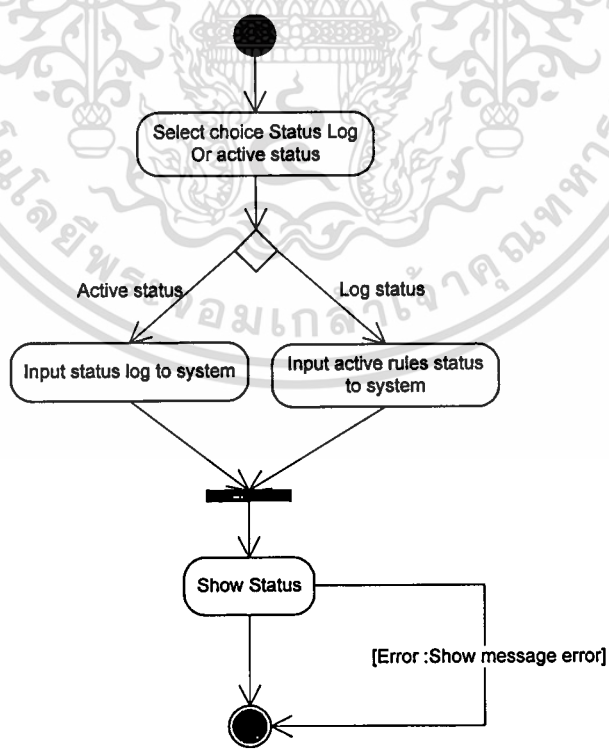
รูปที่ 3.6 แสดง Activity Diagram ของ Edit Existing Rule



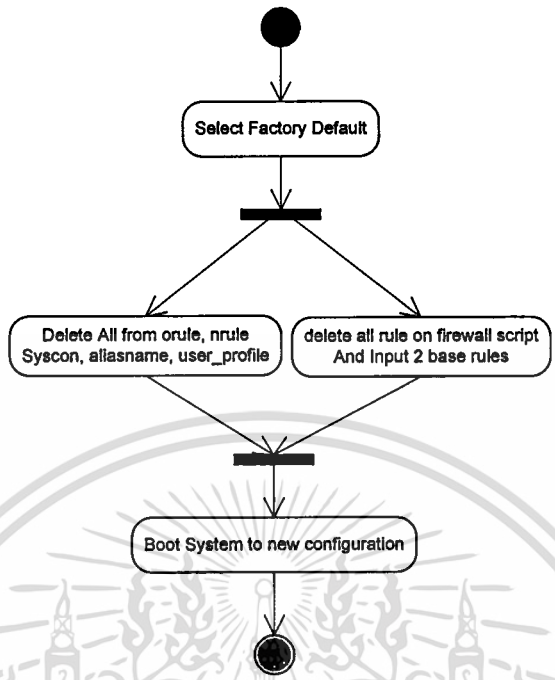
รูปที่ 3.7 แสดง Activity Diagram ของ Delete Rule



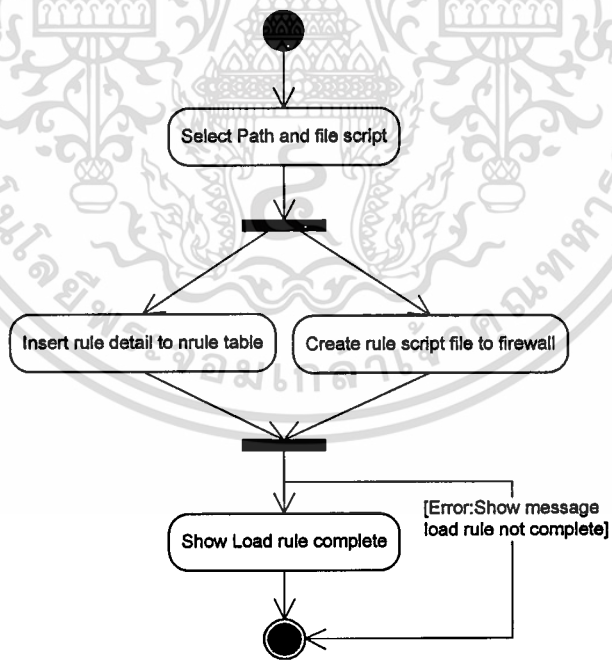
รูปที่ 3.8 แสดง Activity Diagram ของ Activate all Rules



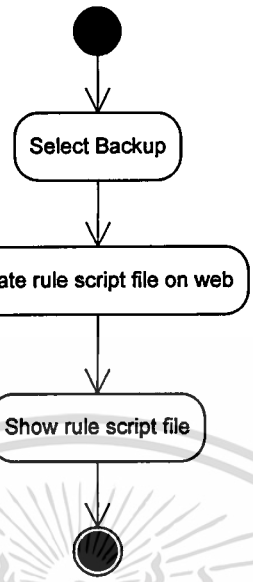
รูปที่ 3.9 แสดง Activity Diagram ของ Monitor Status Rules



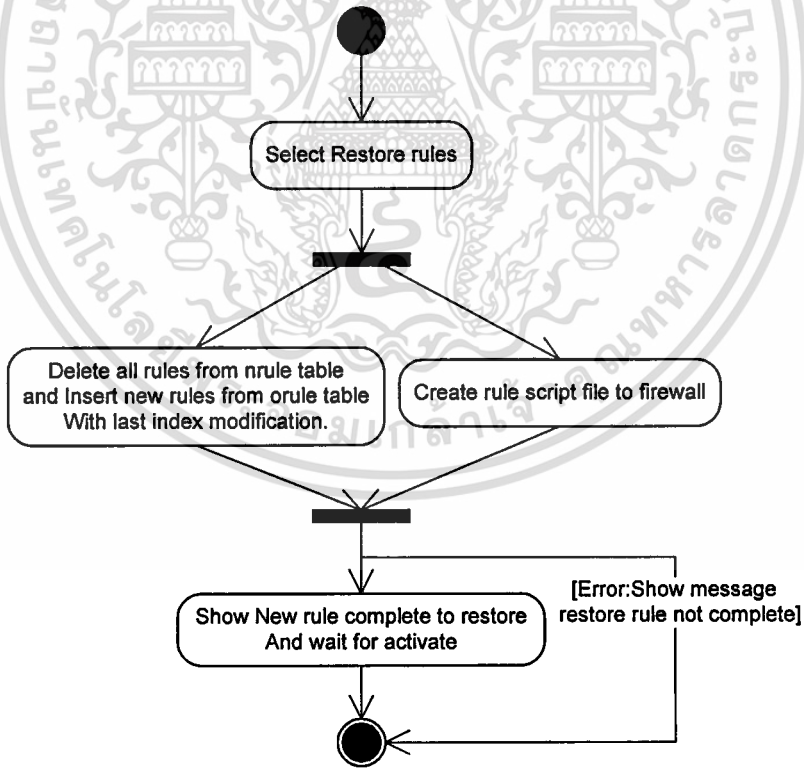
รูปที่ 3.10 แสดง Activity Diagram ของ Set factory default



รูปที่ 3.11 แสดง Activity Diagram ของ Load rules



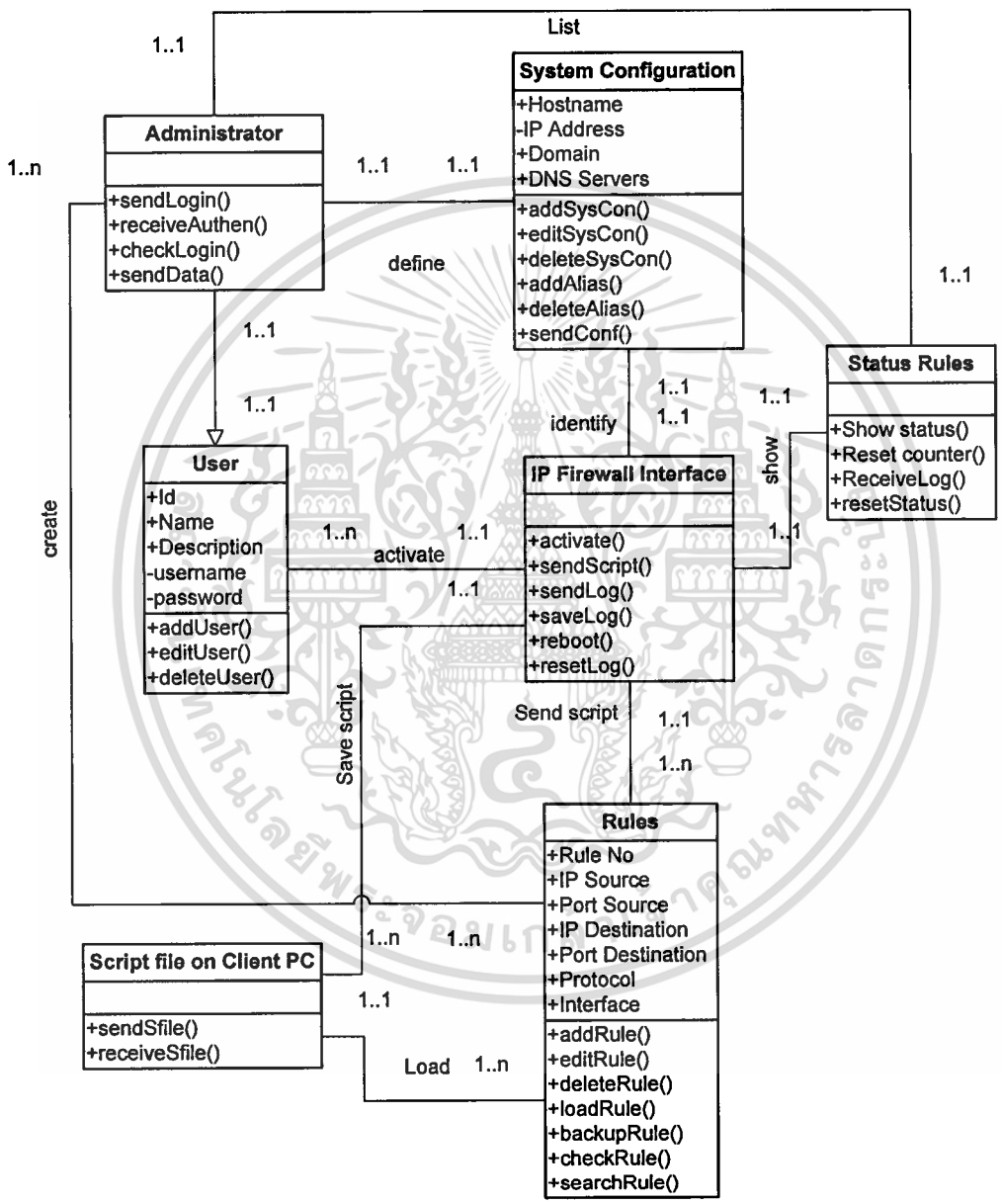
รูปที่ 3.12 แสดง Activity Diagram ของ Backup rules



รูปที่ 3.13 แสดง Activity Diagram ของ Restore rules

3.1.2 Structural Models

เป็นการมองโครงสร้างข้อมูลของระบบ ซึ่งในที่นี้ใช้ Class Diagram เพื่อแสดงโครงสร้างข้อมูลของระบบการติดตั้งกฎไฟร์วอลล์



รูปที่ 3.14 แสดง Class Diagram ระบบการติดตั้งกฎไอพีไฟร์วอลล์

CRC CARD (Class-Responsibility-Collaboration Card)

Class Name: Administrator

Description: ผู้ใช้ระบบ

Responsibilities: ป้อนชื่อผู้ใช้ และรหัสผ่าน และรอการตรวจสอบสิทธิ์การใช้งาน

Collaborators: User Profile

Attributes: UserName, Password

Operations: sendUserpass(), receiveAuthen(),checkLogin()

Class Name: User

Description: ข้อมูลผู้ใช้ระบบ

Responsibilities: เก็บรหัส id no, ชื่อ และรายละเอียดผู้ใช้รวมทั้ง username และ password

Collaborators: Administrator, IP Firewall

Attributes: Id, Name, Description, username, password

Operations: addUser(), editUser(), deleteUser()

Class Name: Client PC

Description: เครื่องที่ใช้ในการส่ง script rule file

Responsibilities: เก็บไฟล์ script rule และส่งไฟล์

Collaborators: IP Firewall

Attributes: -

Operations: sendSfile(), receiveSfile()

Class Name: Rules

Description: ข้อมูลกฎไอพีไฟร์วอลล์

Responsibilities: เก็บกฎไฟร์วอลล์ที่ติดตั้ง

Collaborators: IP Firewall

Attributes: ruleNo, ipSource, portSource, ipDestination, portDestination, protocol, interface

Operations: addRule(), editRule(), deleteRule(), checkRule(), loadRule(), backupRule()

Class Name: IP Firewall Interface

Description: ไอพีไฟร์วอลล์

Responsibilities: เปิด-ปิดการทำงานของกฎต่างๆในไฟร์วอลล์และจัดส่ง-เก็บ script rule file ไปยังฐานข้อมูลกฎและเครื่องไคลเอนต์รวมทั้งจัดเก็บค่า Log ต่างๆ

Collaborators: Rule, Client PC

Attributes: -

Operations: activate(), sendScript(), saveLog(), reboot()

Class Name: System Configuration (SysCon)

Description: เก็บค่าข้อมูลของระบบ

Responsibilities: เก็บรายละเอียดของ host name, Domain, DNS Server, Time zone, NTP Server

Collaborators: IP Firewall

Attributes: host name, Domain, DNS Server, Time zone, NTP Server

Operations: addSysCon(), editSysCon(), sendSysCon()

Class Name: Status Rules

Description: แสดงผลการทำงานของกฎต่างๆ

Responsibilities: แสดงผลการทำงานของกฎต่างๆ

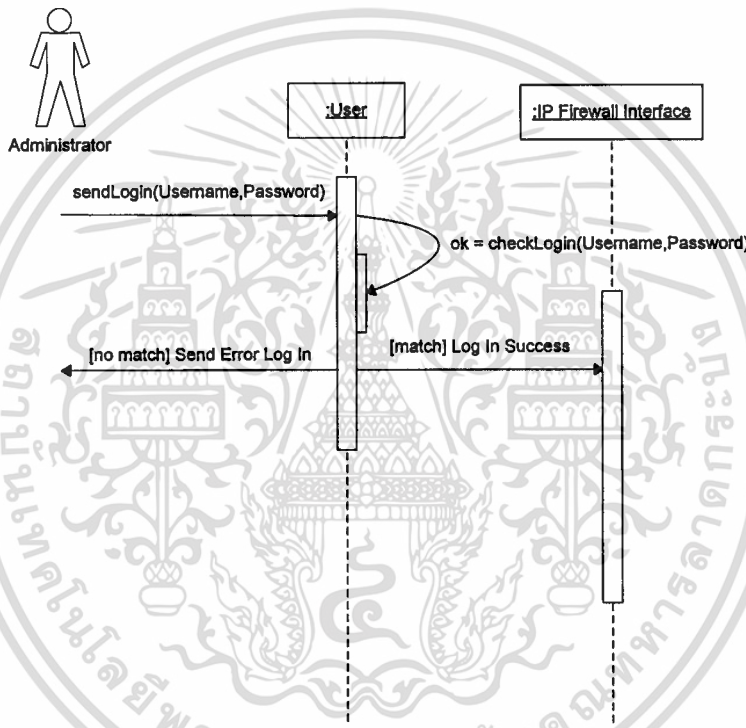
Collaborators: IP Firewall

Attributes: -

Operations: receiveLog(), showStatus(), resetCounter()

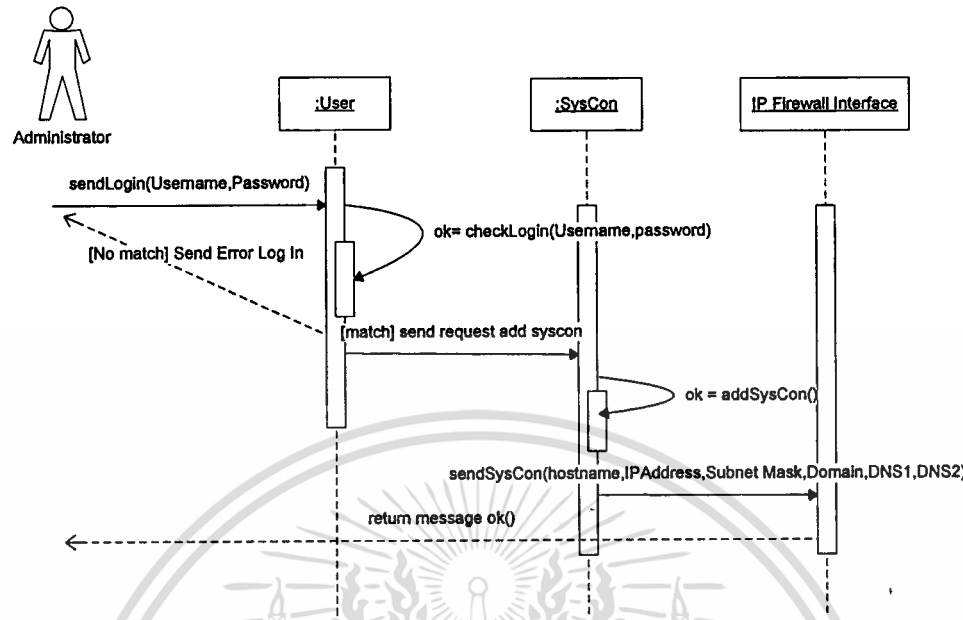
3.1.3 Behavioral Models

เป็นการมองกระบวนการของระบบหรือกลไกของระบบ โดยมองในลักษณะพฤติกรรมของระบบว่าระบบทำงานอย่างไร ซึ่งในที่นี้ใช้ Sequence Diagram เพื่ออธิบายกลไกของระบบในลักษณะพฤติกรรมของระบบ



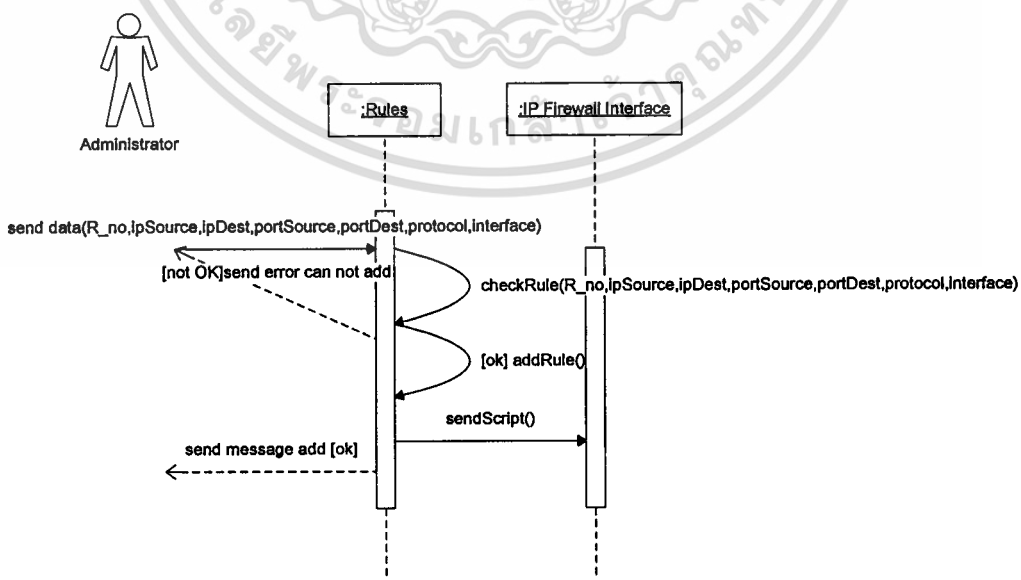
รูปที่ 3.15 แสดง Sequence Diagram ของ Login Firewall Use case

จากรูปที่ 3.15 จะเป็นการแสดงถึงการทำงานของ Login Firewall Use case โดยเมื่อผู้ใช้งานระบบเข้ามาทำการใช้งานโดยการป้อนข้อมูลในส่วนของ username และ password โมดูลในส่วนของ user ก็จะทำการตรวจเช็คข้อมูลและถ้าหากข้อมูลถูกต้องก็จะอนุญาตให้ผู้ใช้งานสามารถใช้งานระบบในส่วนต่างๆได้แต่ถ้าไม่ถูกต้องก็จะทำการแจ้งข้อความผิดพลาดขึ้น



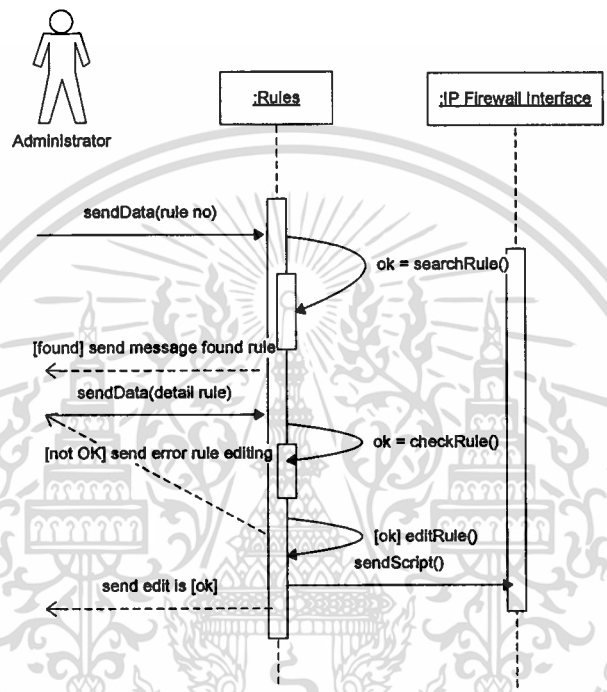
รูปที่ 3.16 แสดง Sequence Diagram ของ Create System configuration Use case

จากรูปที่ 3.16 จะแสดงการทำงานในส่วนของ Create System configuration Use case โดยในการทำงานจะต้องทำการป้อนข้อมูลในส่วนของ username และ password เพื่อทำการตรวจเช็คก่อนว่ามีสิทธิ์ใช้งานระบบหรือไม่ ถ้ามีสิทธิ์ใช้งานระบบก็จะทำการเพิ่มข้อมูลในส่วนของ hostname, ip address, subnet mask, domain, dns1, dns2 เป็นต้น



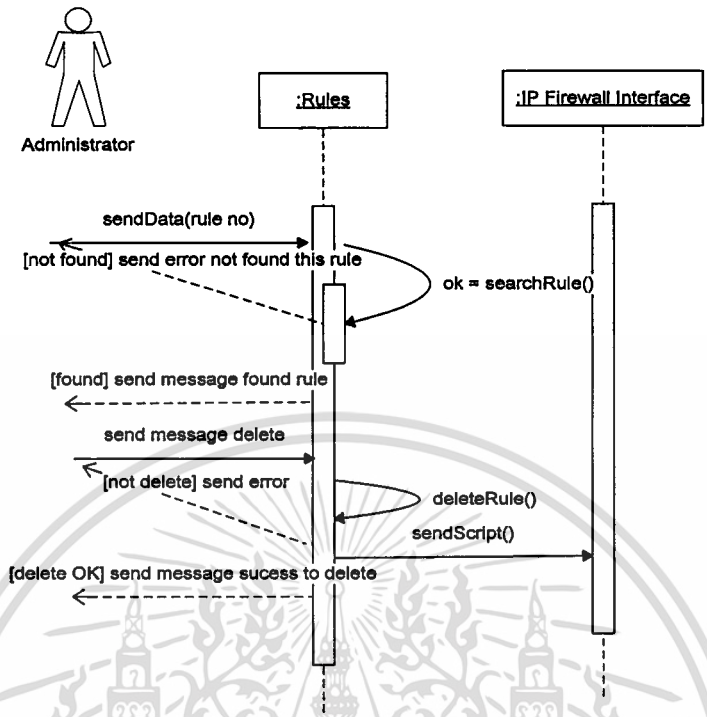
รูปที่ 3.17 แสดง Sequence Diagram ของ Create New Rule Use case

จากรูปที่ 3.17 จะแสดงการทำงานในส่วนของ Create New Rule Use case โดยหลังจากผู้ใช้ได้ทำการล๊อกอินเข้าสู่ระบบเรียบร้อยแล้วจะทำการเพิ่มกฎของไฟร์วอลล์โดยป้อนข้อมูลในส่วนของ ip source ,ip destination, port source, port destination, protocol, interface ฯลฯ เข้าสู่ระบบและระบบจะทำการตรวจเช็คข้อมูลที่ส่งเข้ามาสู่ระบบว่าถูกต้องหรือไม่และแจ้งผลการสร้างกฎเพิ่มให้กับผู้ใช้



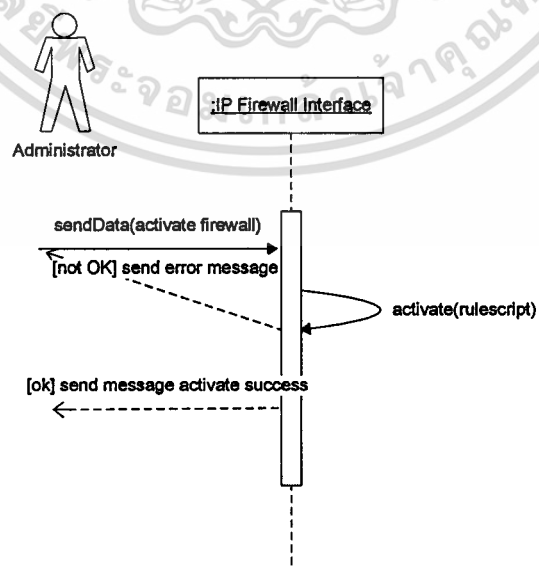
รูปที่ 3.18 แสดง Sequence Diagram ของ Edit Existing Rules Use case

จากรูปที่ 3.18 แสดงการทำงานในส่วนของการแก้ไขกฎที่ได้มีการสร้างไว้แล้วโดยผู้ใช้จะส่งเบอร์ของกฎที่จะทำการแก้ไขและระบบจะทำการตรวจสอบว่ามีหรือไม่ ถ้าพบก็จะให้ทำการแก้ไขโดยผู้ใช้ทำการป้อนข้อมูลเพิ่มหรือแก้ไขข้อมูลจากนั้นนำส่งเข้าสู่ระบบเพื่อตรวจเช็คความถูกต้องและระบบจะแจ้งผลการตรวจสอบ



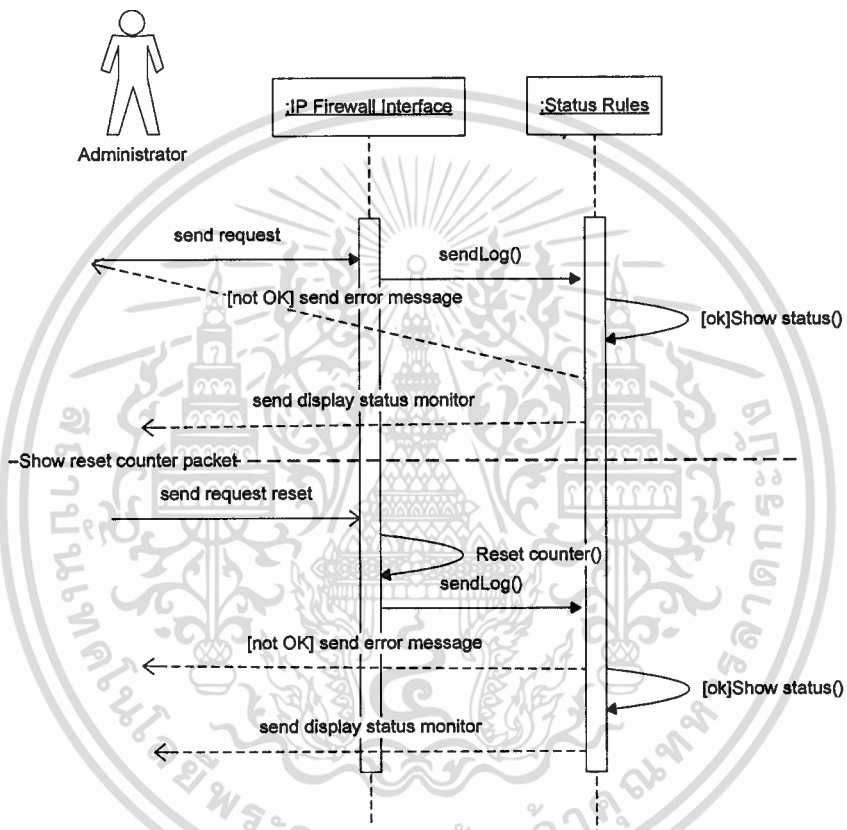
รูปที่ 3.19 แสดง Sequence Diagram ของ Delete Rules Use case

จากรูปที่ 3.19 แสดงการทำงานในส่วนของการลบกฎที่ได้มีการสร้างไว้แล้วโดยผู้ใช้จะส่งเบอร์ของกฎที่จะทำการลบและระบบจะทำการตรวจสอบว่ามีหรือไม่ ถ้าพบก็จะให้ทำการลบกฎได้ จากนั้นระบบจะแจ้งผลการทำงาน



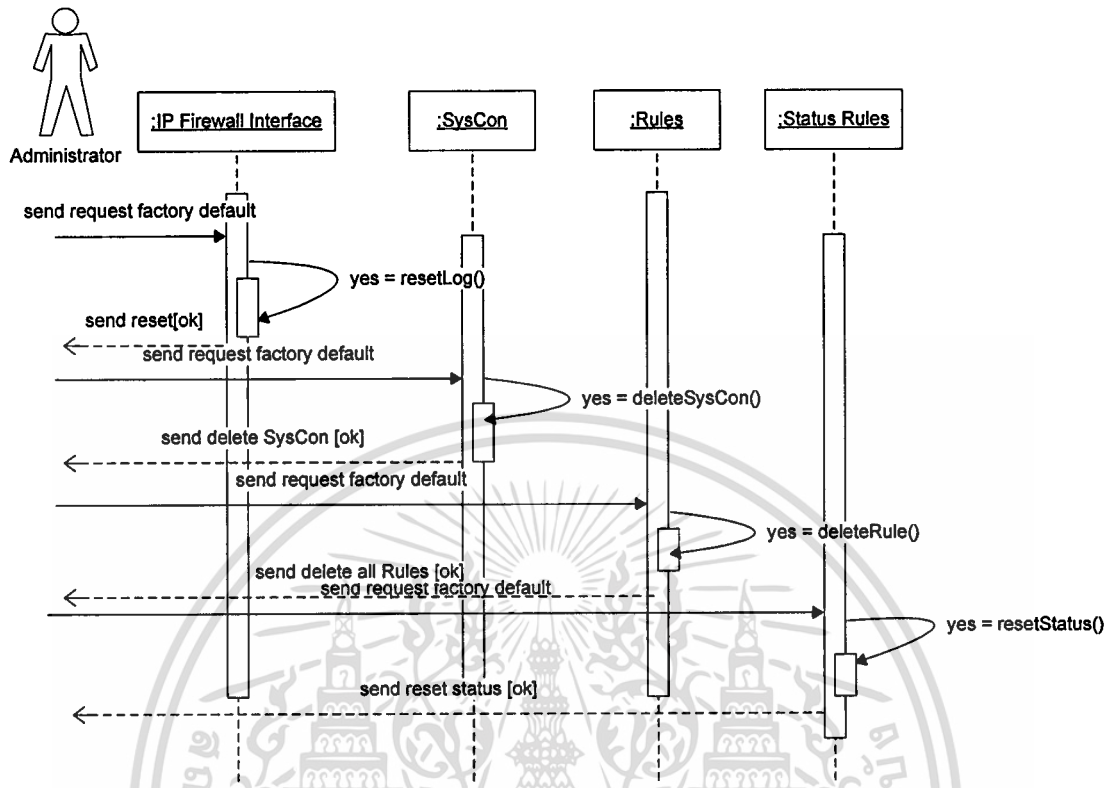
รูปที่ 3.20 แสดง Sequence Diagram ของ Activate All Rules Use case

จากรูปที่ 3.20 แสดงการทำงานในส่วนของ Activate All Rules Use case ซึ่งเป็นส่วนที่จะใช้ในการกำหนดให้กฎที่ได้มีการสร้างไว้แล้วทำงาน โดยผู้ใช้จะส่งกฎทั้งหมดที่ได้ทำการสร้างไว้เข้าสู่ไฟร์วอลล์และทำการ active การทำงานของไฟร์วอลล์กับกฎถ้าการทำงานถูกต้องระบบจะแจ้งผลการทำงาน



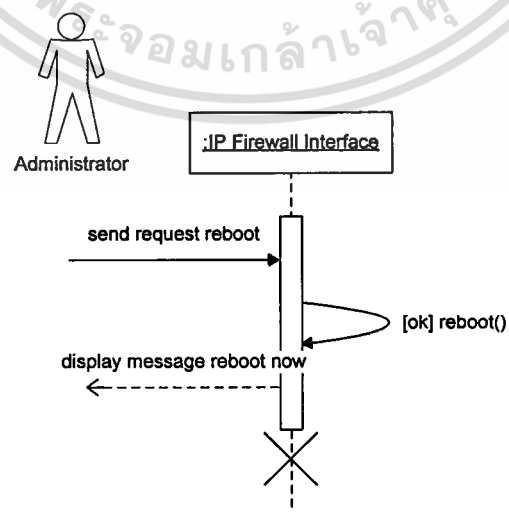
รูปที่ 3.21 แสดง Sequence Diagram ของ Monitor Status Rules Use case

จากรูปที่ 3.21 แสดงการทำงานในส่วนของ Monitor Status Rules Use case โดยผู้ใช้จะทำการ request log จากไฟร์วอลล์จากนั้นระบบก็จะทำการดึงล๊อกไฟล์และทำการแสดงผลของล๊อกไฟล์ถ้ากรณีที่มีล๊อกไฟล์จำนวนมากระบบสามารถทำการลบค่าการเก็บและเริ่มทำการเก็บค่าใหม่อีกครั้งพร้อมทั้งแสดงค่าสถานะการทำงาน



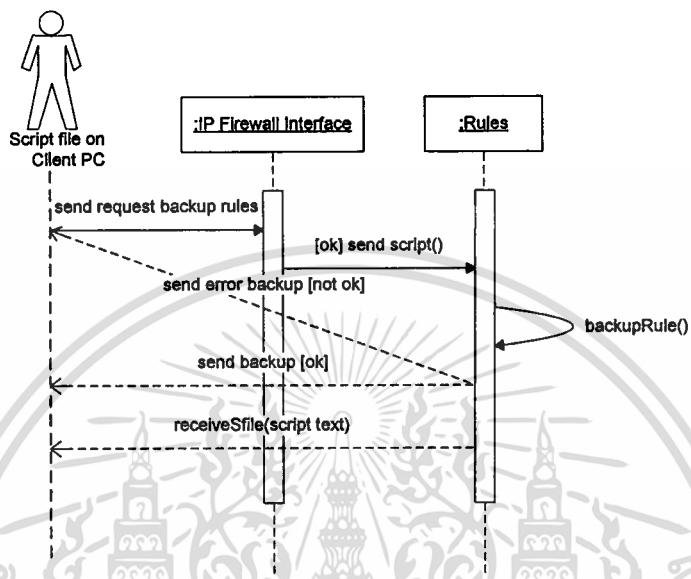
รูปที่ 3.22 แสดง Sequence Diagram ของ Set Factory Default Use case

จากรูปที่ 3.22 แสดงการทำงานในส่วนของ Set Factory Default Use case โดยผู้ใช้จะทำการ request การทำงานจากระบบจากนั้นระบบจะทำการเคลียร์ค่าต่างของระบบออกพร้อมทั้งแจ้งสถานะการทำงาน



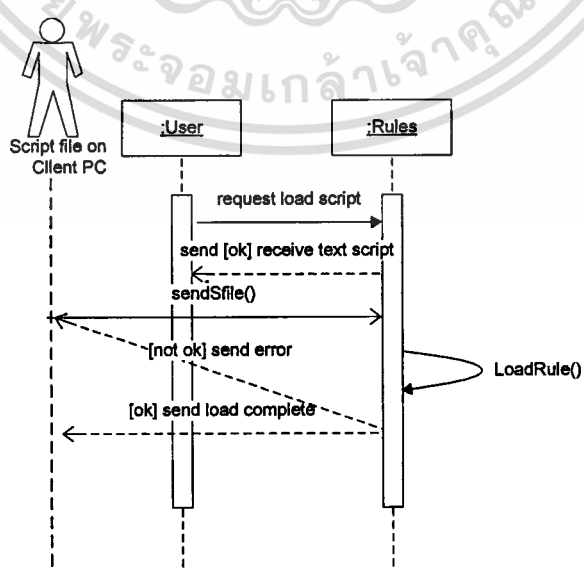
รูปที่ 3.23 แสดง Sequence Diagram ของ Reboot System Use case

จากรูปที่ 3.23 แสดงการทำงานในส่วนของ Reboot System Use case โดยผู้ใช้จะทำการ request การทำงานจากระบบจากนั้นระบบจะทำการติดต่อกับไฟร์วอลล์เพื่อทำการ reboot ระบบ



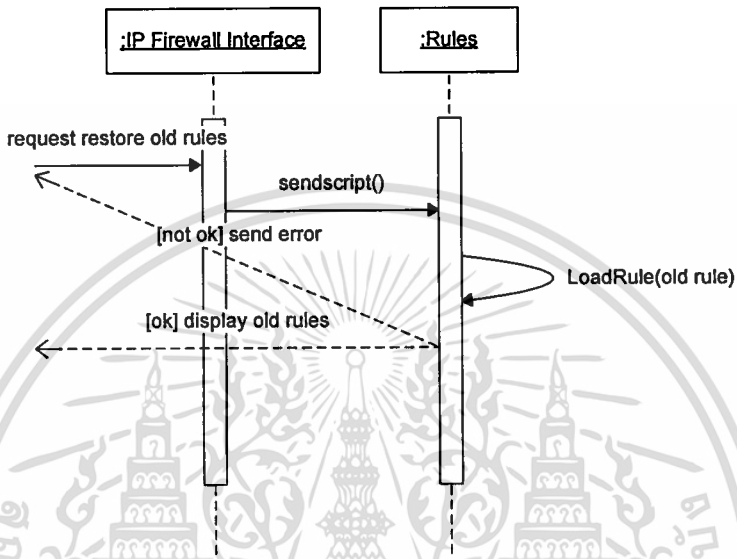
รูปที่ 3.24 แสดง Sequence Diagram ของ Backup Rules Use case

จากรูปที่ 3.24 แสดงการทำงานในส่วนของ Backup Rules Use case โดยผู้ใช้จะทำการ request การทำงานจากระบบจากนั้นระบบจะทำการสร้างไฟล์สคริปเพื่อสำรองข้อมูลกฎทั้งหมดและแจ้งผลการสำรองข้อมูล



รูปที่ 3.25 แสดง Sequence Diagram ของ Load Rules Use case

จากรูปที่ 3.25 แสดงการทำงานในส่วนของ Load Rules Use case โดยผู้ใช้จะทำการ request การ load สคริปไฟล์จากนั้นระบบจะทำการ load สคริปไฟล์จากเครื่อง client เข้าสู่ระบบและแจ้งผลการทำงาน



รูปที่ 3.26 แสดง Sequence Diagram ของ Restore Rules Use case

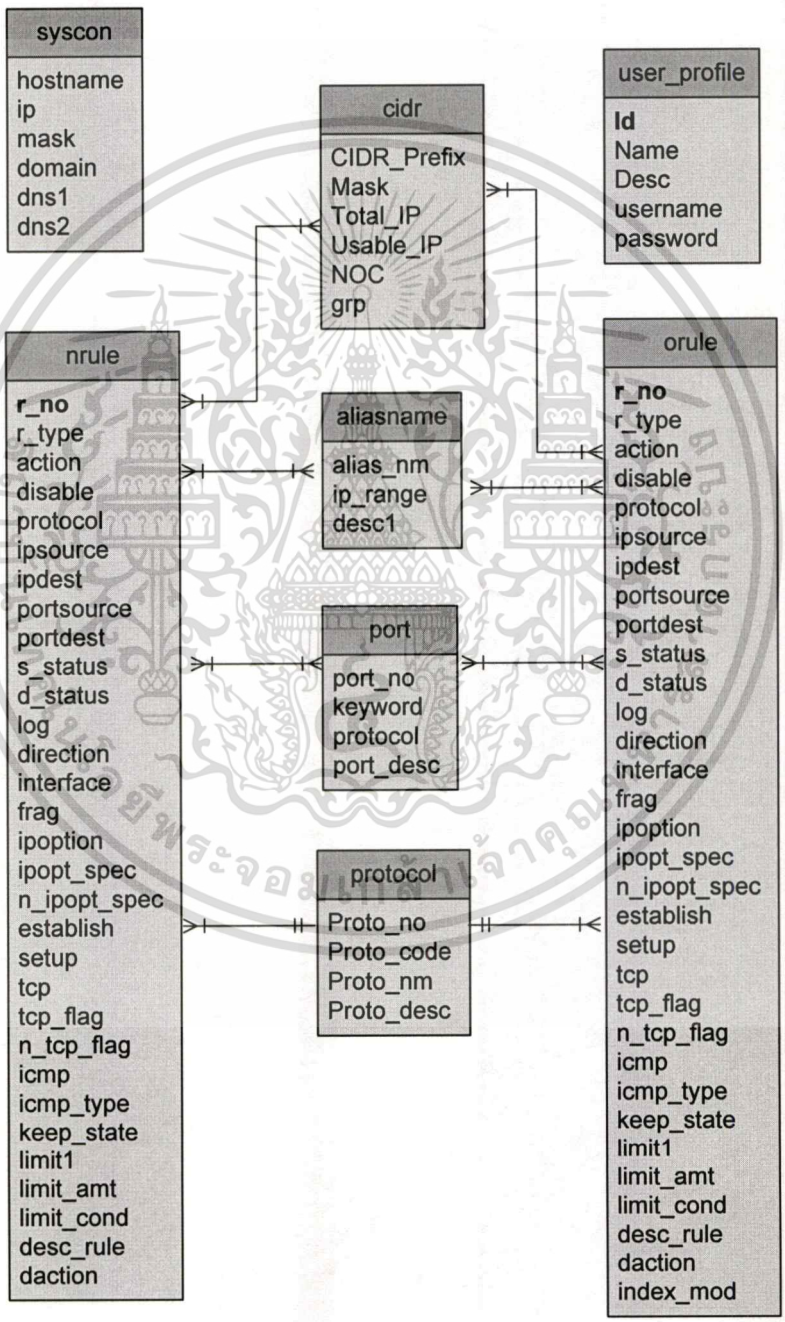
จากรูปที่ 3.26 แสดงการทำงานในส่วนของ Restore Rules Use case โดยผู้ใช้จะทำการ request การทำงานจากระบบ และระบบจะทำการ load กฎจากฐานข้อมูลกฎเก่าเพื่อนำขึ้นมาใช้งานพร้อมทั้งแจ้งผลการทำงาน

3.2 โครงสร้างฐานข้อมูล

เนื่องจากในระบบการกำหนดกฎไฟร์วอลล์มีการเก็บข้อมูลอยู่ 2 รูปแบบด้วยกันคือ

1 การเก็บในรูปฐานข้อมูล

2 การเก็บข้อมูลในรูปไฟล์คอนฟิกหรือไฟล์สคริป



รูปที่ 3.27 แสดง Entity/Relationship Diagram (E/R Diagram)

3.2.1 การออกแบบโครงสร้างฐานข้อมูล

การออกแบบฐานข้อมูลได้แสดงความสัมพันธ์ของตารางต่างๆ ไว้ใน E/R Diagram ตามรูปที่ 3.21 ซึ่งประกอบด้วยตารางต่างๆ ดังนี้

3.2.1.1 ตาราง user_profile

ตารางที่ 3.1 แสดงฟิลด์ของตาราง user_profile

ชื่อข้อมูล	ชื่อเขตข้อมูล	ชนิดข้อมูล	คำอธิบาย
เลขที่ผู้ใช้	ID	Char(3)	รหัสเลขที่ผู้ใช้งานระบบ
ชื่อผู้ใช้	Name	Char(60)	ชื่อผู้ใช้งาน
รายละเอียด	Desc	Char(80)	รายละเอียดของผู้ใช้งาน
ชื่อล๊อกอิน	Username	Char(20)	ชื่อที่ใช้ในการล๊อกอินเข้าสู่ระบบ
รหัสผ่าน	Password	Char(10)	รหัสผ่านเข้าสู่การใช้งานระบบ

3.2.1.2 ตาราง syscon

ตารางที่ 3.2 แสดงฟิลด์ของตาราง syscon

ชื่อข้อมูล	ชื่อเขตข้อมูล	ชนิดข้อมูล	คำอธิบาย
ชื่อโฮสต์	Hostname	Char(20)	ชื่อโฮสต์
ค่าโฮสต์ไอพีแอดเดรส	Ip	Char(15)	ค่า Host IP Address
ค่า Subnet Mask	Mask	Char(15)	ค่า Subnet Mask
ชื่อโดเมน	Domain	Char(60)	ชื่อโดเมน
ค่า DNSเซิร์ฟเวอร์ 1	DNS1	Char(15)	ค่าเซิร์ฟเวอร์ที่ให้บริการ DNS1
ค่า DNSเซิร์ฟเวอร์ 2	DNS2	Char(15)	ค่าเซิร์ฟเวอร์ที่ให้บริการ DNS2

3.2.1.3 ตาราง nrule

ตารางที่ 3.3 แสดงฟิลด์ของตาราง nrule

ชื่อข้อมูล	ชื่อเขตข้อมูล	ชนิดข้อมูล	คำอธิบาย
เลขที่กฎ	R_no	Char(5)	รหัสเลขที่กฎ
ชนิดของกฎไฟร์วอลล์	R_type	Char(10)	ชนิดของกฎระบุเป็น static หรือ dynamic
คำสั่งร่วม	Action	Char(15)	คำสั่งที่ใช้ทำงานร่วมกับคำสั่งหลักเช่น allow, deny เป็นต้น
กำหนดให้กฎไม่ทำงาน	Disable	Char(1)	กำหนดให้กฎที่กำหนดไม่สามารถทำงานได้โดยไม่ต้องลบกฎออกจากรายการกฎ
โปรโตคอล	Protocol	Char(4)	แสดงค่าโปรโตคอลที่จะทำการตรวจสอบ
ค่า IP ต้นทาง	ipSource	Char(40)	แสดงค่า IP Address ต้นทาง
ค่า IP ปลายทาง	ipDest	Char(40)	แสดงค่า IP Address ปลายทาง
ค่าพอร์ตต้นทาง	portSource	Char(120)	แสดงค่าพอร์ตต้นทาง
ค่าพอร์ตปลายทาง	portDest	Char(120)	แสดงค่าพอร์ตปลายทาง
สถานะ ของ Source	S_status	Char(1)	แสดงค่าสถานะ not ของ Source
สถานะ ของ Destination	D_status	Char(1)	แสดงค่าสถานะ not ของ Destination
คำสั่งLog	Log	Char(1)	แสดงค่าการเก็บ Log
ทิศทาง	Direction	Char(3)	แสดงทิศทางของแพ็คเก็ตที่ทำการตรวจสอบมีค่าเป็น in, out
อินเตอร์เฟส	Interface	Char(4)	ค่าอินเตอร์เฟส
ค่า Fragment	Frag	Char(1)	อนุญาตให้กฎจับคู่ได้ถ้า packet ไม่ใช่ Fragment แรกของ diagram
ค่า ipoption	Ipoption	Char(1)	อนุญาตให้กฎจับคู่ได้ถ้า IP header ตรงกับค่า ipoption spec

ตารางที่ 3.3 แสดงฟิลด์ของตาราง nrule (ต่อ)

ชื่อข้อมูล	ชื่อเขตข้อมูล	ชนิดข้อมูล	คำอธิบาย
ค่า ipopt_spec	Ipopt_spec	Char(16)	ค่า ipopt_spec เช่น ssrr, lsrr, rr, ts
สถานะ ipopt_spec	N_ipopt_spec	Char(1)	ค่าสถานะการไม่ตรวจสอบตามเงื่อนไขของ ipopt_spec
ค่า establish	Establish	Char(1)	อนุญาตให้กฎจับคู่ได้ถ้า packet เป็นส่วนหนึ่งของการติดต่อแบบ established TCP
ค่า setup	Setup	Char(1)	อนุญาตให้กฎจับคู่ TCP packet ที่มี SYN bit แต่ไม่มี ACK bit
ค่า tcpflags	Tcp	Char(1)	อนุญาตให้กฎจับคู่ได้ถ้า TCP header ตรงกับค่า flag
ค่า flag	Tcp_flag	Char(23)	ค่า flag ของ tcpflags เช่น fin, syn, rst, psh, ack และ urg
ค่าสถานะ tcp_flag	N_tcp_flag	Char(1)	ค่าสถานะการไม่ตรวจสอบตามเงื่อนไขของ tcp_flag
ค่า Icmptypes	Icmp	Char(1)	ค่า Icmptypes
ค่า Icmp_type	Icmp_type	Char(38)	ค่า type ของ Icmptypes
ค่า Keep_state	Keep_state	Char(1)	ค่า Keep_state
ค่า Limit	Limit1	Char(1)	ค่า Limit
ค่า Limit_amt	Limit_amt	Char(2)	ค่า Limit amount
ค่า Limit_cond	Limit_cond	Char(1)	ค่าเงื่อนไขของ Limit
คำอธิบายเกี่ยวกับกฎ	Desc_rule	Char(80)	แสดงคำอธิบายเกี่ยวกับกฎที่กำหนดขึ้น
ค่าเงื่อนไขตามaction	Daction	Char(40)	แสดงค่าเงื่อนไขตามaction

3.2.1.4 ตาราง orule

ตารางที่ 3.4 แสดงฟิลด์ของตาราง orule

ชื่อข้อมูล	ชื่อเขตข้อมูล	ชนิดข้อมูล	คำอธิบาย
เลขที่กฎ	R_no	Char(5)	รหัสเลขที่กฎ
ชนิดของกฎไฟร์วอลล์	R_type	Char(10)	ชนิดของกฎระบุเป็น static หรือ dynamic
คำสั่งร่วม	Action	Char(15)	คำสั่งที่ใช้ทำงานร่วมกับคำสั่งหลักเช่น allow, deny เป็นต้น
กำหนดให้กฎไม่ทำงาน	Disable	Char(1)	กำหนดให้กฎที่กำหนดไม่สามารถทำงานได้โดยไม่ต้องลบกฎออกจากรายการกฎ
โปรโตคอล	Protocol	Char(4)	แสดงค่าโปรโตคอลที่จะทำการตรวจสอบ
ค่า IP ต้นทาง	ipSource	Char(40)	แสดงค่า IP Address ต้นทาง
ค่า IP ปลายทาง	ipDest	Char(40)	แสดงค่า IP Address ปลายทาง
ค่าพอร์ตต้นทาง	portSource	Char(120)	แสดงค่าพอร์ตต้นทาง
ค่าพอร์ตปลายทาง	portDest	Char(120)	แสดงค่าพอร์ตปลายทาง
สถานะ ของ Source	S_status	Char(1)	แสดงค่าสถานะ not ของ Source
สถานะ ของ Destination	D_status	Char(1)	แสดงค่าสถานะ not ของ Destination
คำสั่งLog	Log	Char(1)	แสดงค่าการเก็บ Log
ทิศทาง	Direction	Char(3)	แสดงทิศทางของแพ็คเก็ตที่ทำการตรวจสอบมีค่าเป็น in, out
อินเตอร์เฟส	Interface	Char(4)	ค่าอินเตอร์เฟส
ค่า Fragment	Frag	Char(1)	อนุญาตให้กฎจับคู่ได้ถ้า packet ไม่ใช่ Fragment แรกของ diagram
ค่า ipoption	Ipoption	Char(1)	อนุญาตให้กฎจับคู่ได้ถ้า IP header ตรงกับค่า ipoption spec

ตารางที่ 3.4 แสดงฟิลด์ของตาราง orule (ต่อ)

ชื่อข้อมูล	ชื่อเขตข้อมูล	ชนิดข้อมูล	คำอธิบาย
ค่า ipopt_spec	Ipopt_spec	Char(16)	ค่า ipopt_spec เช่น ssrr, lsrr, rr, ts
สถานะ ipopt_spec	N_ipopt_spec	Char(1)	ค่าสถานะการไม่ตรวจสอบตามเงื่อนไขของ ipopt_spec
ค่า establish	Establish	Char(1)	อนุญาตให้กฎจับคู่ได้ถ้า packet เป็นส่วนหนึ่งของการติดต่อแบบ established TCP
ค่า setup	Setup	Char(1)	อนุญาตให้กฎจับคู่ TCP packet ที่มี SYN bit แต่ไม่มี ACK bit
ค่า tcpflags	Tcp	Char(1)	อนุญาตให้กฎจับคู่ได้ถ้า TCP header ตรงกับค่า flag
ค่า flag	Tcp_flag	Char(23)	ค่า flag ของ tcpflags เช่น fin, syn, rst, psh, ack และ urg
ค่าสถานะ tcp_flag	N_tcp_flag	Char(1)	ค่าสถานะการไม่ตรวจสอบตามเงื่อนไขของ tcp_flag
ค่า Icmptypes	Icmp	Char(1)	ค่า Icmptypes
ค่า Icmp_type	Icmp_type	Char(38)	ค่า type ของ Icmptypes
ค่า Keep_state	Keep_state	Char(1)	ค่า Keep_state
ค่า Limit	Limit1	Char(1)	ค่า Limit
ค่า Limit_amt	Limit_amt	Char(2)	ค่า Limit amount
ค่า Limit_cond	Limit_cond	Char(1)	ค่าเงื่อนไขของ Limit
คำอธิบายเกี่ยวกับกฎ	Desc_rule	Char(80)	แสดงคำอธิบายเกี่ยวกับกฎที่กำหนดขึ้น
ค่าเงื่อนไขตามaction	Daction	Char(40)	แสดงค่าเงื่อนไขตามaction
ค่า Index modification	Index_mod	Char(2)	ค่า index modification

3.2.1.5 ตาราง port

ตารางที่ 3.5 แสดงฟิลด์ของตาราง port

ชื่อข้อมูล	ชื่อเขตข้อมูล	ชนิดข้อมูล	คำอธิบาย
รหัสพอร์ต	Port_no	Char(5)	รหัสพอร์ต
ชื่อพอร์ต	Keyword	Char(20)	ชื่อพอร์ต
Protocol	Protocol	Char(11)	ค่าโปรโตคอล
คำอธิบายพอร์ต	Port_desc	Char(120)	คำอธิบายพอร์ต

3.2.1.6 ตาราง aliasname

ตารางที่ 3.6 แสดงฟิลด์ของตาราง aliasname

ชื่อข้อมูล	ชื่อเขตข้อมูล	ชนิดข้อมูล	คำอธิบาย
ชื่อเรียก IP_range	Alias_nm	Char(20)	ชื่อเรียก IP_range
ค่า IP_range	IP_range	Char(40)	ค่า IP_range
คำอธิบาย IP_range	Desc1	Char(80)	คำอธิบาย IP_range

3.2.1.7 ตาราง protocol

ตารางที่ 3.7 แสดงฟิลด์ของตาราง protocol

ชื่อข้อมูล	ชื่อเขตข้อมูล	ชนิดข้อมูล	คำอธิบาย
เลขที่โปรโตคอล	Proto_no	Integer(3)	เลขที่โปรโตคอล
รหัสโปรโตคอล	Proto_code	Char(11)	รหัสโปรโตคอล
ชื่อโปรโตคอล	Proto_nm	Char(11)	ชื่อโปรโตคอล
รายละเอียดโปรโตคอล	Proto_desc	Char(80)	รายละเอียดโปรโตคอล

3.2.1.8 ตาราง CIDR ใช้บันทึกค่า Class Inter Domain Routing

ตารางที่ 3.8 แสดงฟิลด์ของตาราง CIDR

ชื่อข้อมูล	ชื่อเขตข้อมูล	ชนิดข้อมูล	คำอธิบาย
รหัส CIDR	CIDR_Prefix	Char(3)	รหัส CIDR
ค่า Mask	Mask	Char(15)	ค่า Subnet Mask
ค่า Total IP's	Total_IP	Interger	ค่าจำนวน IP Address ทั้งหมด
ค่า Usable IP's	Usable_IP	Interger	ค่าจำนวน IP Address ที่สามารถใช้ได้
ค่า Number Of Class	NOC	Interger	ค่าจำนวนของ Class
ค่ากลุ่มข้อมูล	grp	Char(1)	ค่ากลุ่มข้อมูล

3.2.2 การออกแบบโครงสร้างไฟล์คอนฟิกหรือไฟล์สคริป

ตารางที่ 3.9 แสดงฟิลด์ของโครงสร้างไฟล์คอนฟิกหรือไฟล์สคริป (ipfw.rules)

ชื่อข้อมูล	ชื่อเขตข้อมูล	ชนิดข้อมูล	คำอธิบาย
คำสั่งเพิ่มกฎไฟร์วอลล์	Add	Char(3)	คำสั่งที่ใช้ในการเพิ่มกฎไฟร์วอลล์ "add"
เลขที่กฎ	R_no	Char(5)	รหัสเลขที่กฎ
คำสั่งร่วม	Action	Char(15)	คำสั่งที่ใช้ทำงานร่วมกับคำสั่งหลักเช่น allow, deny เป็นต้น
คำสั่งLog	Log	Char(3)	แสดงค่าการเก็บ Log
โปรโตคอล	Protocol	Char(11)	แสดงค่าโปรโตคอลที่จะทำการตรวจสอบ
สถานะ ของ Source	S_status	Char(3)	แสดงค่าสถานะ not ของ Source
ค่า IP ต้นทาง	ipSource	Char(40)	แสดงค่า IP Address ต้นทาง
ค่าพอร์ตต้นทาง	portSource	Char(120)	แสดงค่าพอร์ตต้นทาง
สถานะ ของ Destination	D_status	Char(3)	แสดงค่าสถานะ not ของ Destination
ค่า IP ปลายทาง	ipDest	Char(40)	แสดงค่า IP Address ปลายทาง
ค่าพอร์ตปลายทาง	portDest	Char(120)	แสดงค่าพอร์ตปลายทาง
ทิศทาง	Direction	Char(3)	แสดงทิศทางของแพ็คเก็ตที่ทำการตรวจสอบมีค่าเป็น in, out
อินเตอร์เฟส	Interface	Char(4)	ค่าอินเตอร์เฟส
ค่า Fragment	Frag	Char(4)	อนุญาตให้กฎจับคู่ได้ถ้า packet ไม่ใช่ Fragment แรกของ diagram
ค่า ipoption	Ipoption	Char(8)	อนุญาตให้กฎจับคู่ได้ถ้า IP header ตรงกับค่า ipoption spec
ค่า ipopt_spec	Ipopt_spec	Char(16)	ค่า ipopt_spec เช่น ssrr, lsrr, rr, ts

ตารางที่ 3.9 แสดงฟิลด์ของโครงสร้างไฟล์คอนฟิกหรือไฟล์สคริป (ipfw.rules) (ต่อ)

ชื่อข้อมูล	ชื่อเขตข้อมูล	ชนิดข้อมูล	คำอธิบาย
สถานะ ipopt_spec	N_ipopt_spec	Char(1)	ค่าสถานะการไม่ตรวจสอบตามเงื่อนไขของ ipopt_spec
ค่า establish	Establish	Char(11)	อนุญาตให้กฎจับคู่ได้ถ้า packet เป็นส่วนหนึ่งของการติดต่อแบบ established TCP
ค่า setup	Setup	Char(5)	อนุญาตให้กฎจับคู่ TCP packet ที่มี SYN bit แต่ไม่มี ACK bit
ค่า tcpflags	Tcp	Char(8)	อนุญาตให้กฎจับคู่ได้ถ้า TCP header ตรงกับค่า flag
ค่า flag	Tcp_flag	Char(23)	ค่า flag ของ tcpflags เช่น fin, syn, rst, psh, ack และ urg
ค่าสถานะ tcp_flag	N_tcp_flag	Char(1)	ค่าสถานะการไม่ตรวจสอบตามเงื่อนไขของ tcp_flag
ค่า Icmptypes	Icmp	Char(9)	ค่า Icmptypes
ค่า Icmp_type	Icmp_type	Char(38)	ค่า type ของ Icmptypes
ค่า Keep_state	Keep_state	Char(10)	ค่า Keep_state
ค่า Limit	Limit1	Char(5)	ค่า Limit
ค่า Limit_amt	Limit_amt	Char(2)	ค่า Limit amount
ค่า Limit_cond	Limit_cond	Char(1)	ค่าเงื่อนไขของ Limit
ค่าเงื่อนไขตามaction	Daction	Char(40)	แสดงค่าเงื่อนไขตามaction

ตารางที่ 3.10 แสดงฟิลด์ของโครงสร้างไฟล์คอนฟิกหรือไฟล์สคริป (rule_backup.txt)

ชื่อข้อมูล	ชื่อเขตข้อมูล	ชนิดข้อมูล	คำอธิบาย
เลขที่กฎ	R_no	Char(5)	รหัสเลขที่กฎ
ชนิดของกฎไฟร์วอลล์	R_type	Char(10)	ชนิดของกฎระบุเป็น static หรือ dynamic
คำสั่งร่วม	Action	Char(15)	คำสั่งที่ใช้ทำงานร่วมกับคำสั่งหลักเช่น allow, deny เป็นต้น
กำหนดให้กฎไม่ทำงาน	Disable	Char(1)	กำหนดให้กฎที่กำหนดไม่สามารถทำงานได้โดยไม่ต้องลบกฎออกจากรายการกฎ
โปรโตคอล	Protocol	Char(4)	แสดงค่าโปรโตคอลที่จะทำการตรวจสอบ
ค่า IP ต้นทาง	ipSource	Char(40)	แสดงค่า IP Address ต้นทาง
ค่า IP ปลายทาง	ipDest	Char(40)	แสดงค่า IP Address ปลายทาง
ค่าพอร์ตต้นทาง	portSource	Char(120)	แสดงค่าพอร์ตต้นทาง
ค่าพอร์ตปลายทาง	portDest	Char(120)	แสดงค่าพอร์ตปลายทาง
สถานะ ของ Source	S_status	Char(1)	แสดงค่าสถานะ not ของ Source
สถานะ ของ Destination	D_status	Char(1)	แสดงค่าสถานะ not ของ Destination
คำสั่งLog	Log	Char(1)	แสดงค่าการเก็บ Log
ทิศทาง	Direction	Char(3)	แสดงทิศทางของแพ็คเก็ตที่ทำการตรวจสอบมีค่าเป็น in, out
อินเตอร์เฟส	Interface	Char(4)	ค่าอินเตอร์เฟส
ค่า Fragment	Frag	Char(1)	อนุญาตให้กฎจับคู่ได้ถ้า packet ไม่ใช่ Fragment แรกของ diagram
ค่า ipoption	Ipoption	Char(1)	อนุญาตให้กฎจับคู่ได้ถ้า IP header ตรงกับค่า ipoption spec

ตารางที่ 3.10 แสดงฟิลด์ของโครงสร้างไฟล์คอนฟิกหรือไฟล์สคริป (rule_backup.txt) (ต่อ)

ชื่อข้อมูล	ชื่อเขตข้อมูล	ชนิดข้อมูล	คำอธิบาย
ค่า ipopt_spec	Ipopt_spec	Char(16)	ค่า ipopt_spec เช่น ssrr, lsrr, rr, ts
สถานะ ipopt_spec	N_ipopt_spec	Char(1)	ค่าสถานะการไม่ตรวจสอบตามเงื่อนไขของ ipopt_spec
ค่า establish	Establish	Char(1)	อนุญาตให้กฎจับคู่ได้ถ้า packet เป็นส่วนหนึ่งของการติดต่อแบบ established TCP
ค่า setup	Setup	Char(1)	อนุญาตให้กฎจับคู่ TCP packet ที่มี SYN bit แต่ไม่มี ACK bit
ค่า tcpflags	Tcp	Char(1)	อนุญาตให้กฎจับคู่ได้ถ้า TCP header ตรงกับค่า flag
ค่า flag	Tcp_flag	Char(23)	ค่า flag ของ tcpflags เช่น fin, syn, rst, psh, ack และ urg
ค่าสถานะ tcp_flag	N_tcp_flag	Char(1)	ค่าสถานะการไม่ตรวจสอบตามเงื่อนไขของ tcp_flag
ค่า Icmptypes	Icmp	Char(1)	ค่า Icmptypes
ค่า Icmp_type	Icmp_type	Char(38)	ค่า type ของ Icmptypes
ค่า Keep_state	Keep_state	Char(1)	ค่า Keep_state
ค่า Limit	Limit1	Char(1)	ค่า Limit
ค่า Limit_amt	Limit_amt	Char(2)	ค่า Limit amount
ค่า Limit_cond	Limit_cond	Char(1)	ค่าเงื่อนไขของ Limit
คำอธิบายเกี่ยวกับกฎ	Desc_rule	Char(80)	แสดงคำอธิบายเกี่ยวกับกฎที่กำหนดขึ้น
ค่าเงื่อนไขตามaction	Daction	Char(40)	แสดงค่าเงื่อนไขตามaction

หมายเหตุ rule_backup.txt ภายในค่าข้อมูลแต่ละค่าจะคั่นข้อมูลด้วย "&#" และจบข้อมูลแต่ละกฎด้วย "$$" สำหรับตัวอย่างของ rule_backup.txt ดูได้จาก **ภาคผนวก จ**

บทที่ 4
การพัฒนาโปรแกรมการกำหนดกฎของไฟร์วอลล์

ในการพัฒนาโปรแกรมระบบการกำหนดกฎของไฟร์วอลล์ ได้มีการกำหนดแผนการพัฒนา ดังนี้

4.1 การวางแผนปฏิบัติงาน

ได้เลือกใช้ระบบปฏิบัติการและซอฟแวร์ดังต่อไปนี้ (สามารถอ่านรายละเอียดลำดับขั้นตอนในการติดตั้งและแหล่งที่มาของโปรแกรม ได้จากภาคผนวก ก)

1. **ระบบปฏิบัติการเลือกใช้ FreeBSD 5.4**

   เป็นระบบปฏิบัติการแบบยูนิกซ์ที่ได้รับความนิยมอย่างแพร่หลาย และไม่ต้องเสียค่าใช้จ่าย โดยระบบปฏิบัติการ FreeBSD นั้นเป็นระบบปฏิบัติการที่มีความปลอดภัยสูงกว่าระบบยูนิกซ์แบบ Open Source ทั่วๆไป

2. **ซอฟต์แวร์ไฟร์วอลล์เลือกใช้ IPFW**

   เป็นซอฟต์แวร์ไฟร์วอลล์ที่สามารถทำงานกับระบบปฏิบัติการ FreeBSD โดยที่ IPFW จะทำงานอยู่ใน เคอร์เนลของระบบปฏิบัติการ ซึ่งมีส่วนที่ติดต่อกับผู้ใช้งานแบบ command line โดยผู้ใช้งานจะต้องทำการเพิ่มหรือลบกฎด้วยการใช้คำสั่ง

3. **เว็บเซิร์ฟเวอร์ที่รองรับการทำงานของ PHP และ MySQL เลือกใช้ Apache เวอร์ชัน 2.0.53**

   Apache เป็นซอฟแวร์สำหรับให้บริการเว็บเซิร์ฟเวอร์ (HTTP/Web Server) ผ่านทางโปรโตคอล HTTP โดยเป็นซอฟแวร์แบบ Open Source สามารถนำมาใช้งานได้โดยไม่มีค่าใช้จ่าย

4. **ซอต์แวร์ภาษา เลือกใช้ PHP เวอร์ชัน 5**

   PHP Extension เป็น scripting language ที่ทำงานร่วมกับ HTML โดยสามารถเขียนแทรงเข้าไปใน HTML โดย PHP เป็น Open Source ที่สามารถใช้ได้กับหลายๆ ระบบปฏิบัติการไม่ว่าจะเป็น windows, Unix, Linux และยังสามารถรองรับการติดต่อกับฐานข้อมูลได้หลายชนิด

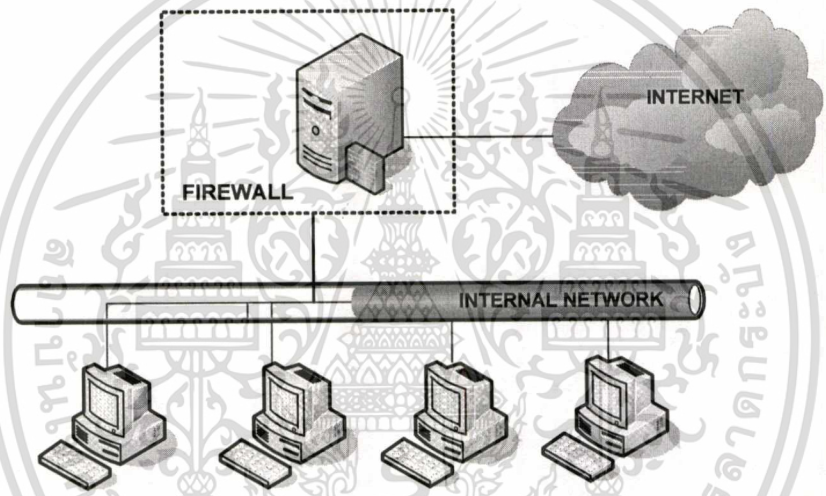
5. **ซอฟต์แวร์ระบบฐานข้อมูล เลือกใช้ MySQL เวอร์ชัน 4**

   MySQL เป็นโปรแกรมฐานข้อมูลที่นิยมใช้กันอย่างแพร่หลาย และมีประสิทธิภาพสามารถนำมาใช้ทดแทนโปรแกรมฐานข้อมูลที่มีจำหน่ายในเชิงพาณิชย์ได้

4.2 การวางแผนโครงสร้างของระบบ

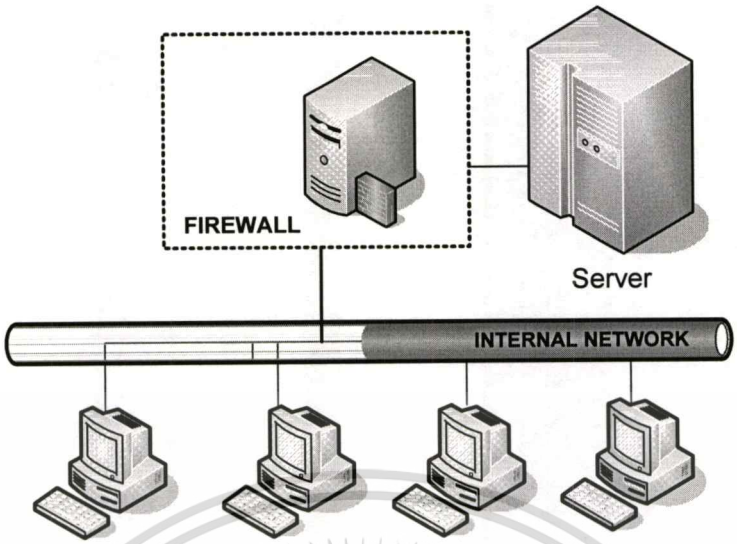
โครงสร้างของระบบที่จะนำเอาไฟร์วอลล์มาใช้งานได้มีการออกแบบไว้ 2 ลักษณะ คือ

- **รูปแบบที่ 1** จะทำการติดตั้งไฟร์วอลล์บนเครื่องที่ทำหน้าที่เป็น Gateway โดยที่ไฟร์วอลล์นี้จะทำหน้าที่เป็นตัวแทนที่จะนำข้อมูลจากภายนอกที่เครื่องลูกข่ายต้องการมาให้โดยที่ไฟร์วอลล์จะทำหน้าที่กรอง packet ที่จะผ่านเข้าออก และดูว่า packet ใดผ่านได้หรือไม่ได้ (แสดงในรูปที่ 4.1) จะเห็นได้ว่าเครื่องลูกข่ายจะออกไปยังอินเทอร์เน็ตได้จะต้องผ่านออกไปทางไฟร์วอลล์เท่านั้น



รูปที่ 4.1 แสดงโครงสร้างการใช้งานไฟร์วอลล์ในลักษณะที่ 1

- **รูปแบบที่ 2** จะทำการติดตั้งไฟร์วอลล์บนเครื่องคอมพิวเตอร์ที่ให้บริการ ที่ต้องการความปลอดภัยสูง โดยเครื่องลูกข่ายที่จะเข้าถึงบริการนั้นๆ จะเรียกใช้บริการผ่านไฟร์วอลล์ ซึ่งไฟร์วอลล์จะทำหน้าที่กรอง packet ที่จะผ่านเข้าออก และดูว่าจะอนุญาตให้ packet ใดผ่านได้หรือไม่ได้ (แสดงในรูปที่ 4.2)



รูปที่ 4.2 แสดงโครงสร้างการใช้งานไฟร์วอลล์ในลักษณะที่ 2

4.3 การเริ่มต้นการใช้งานไฟร์วอลล์ IPFW

ก่อนที่จะเริ่มใช้งานซอฟต์แวร์ IPFW จะต้องทำการเปิดการใช้งาน IPFW ซึ่งอยู่ในเคอร์เนลของระบบปฏิบัติการ FreeBSD ก่อน โดยแฟ้มคอนฟิกของเคอร์แนลของ FreeBSD ปกติจะเก็บอยู่ที่แฟ้ม **/usr/src/sys/i386/conf/GENERIC** ซึ่งในแฟ้มนี้จะเก็บค่าดีฟอลต์ของเคอร์เนลของ FreeBSD ซึ่งไม่สามารถใช้งาน FreeBSD ในบางลักษณะ เช่น การทำไฟร์วอลล์ (Fire Wall) การตั้งเวลาด้วยเน็ตเวิร์กไทมโปรโตคอล(ntp) หรือการทำเน็ตเวิร์กแอดเซสทรานเลชั่น(nat) เป็นต้น

การคอมไพล์เคอร์แนลตัวใหม่ต้องจึงต้องทำการแก้ค่าในแฟ้ม /usr/src/sys/i386/conf /GENERIC และเพื่อเป็นการป้องกันความผิดพลาดจึงควรทำการสำรองแฟ้มนี้ไว้โดยใช้วิธีดังนี้

1. ที่ /root ทำการสร้างไดเรคทอรีชื่อว่า kernels ด้วยคำสั่ง

   #mkdir /root/kernels

2. ทำการคัดลอกแฟ้ม /usr/src/sys/i386/conf/GENERIC มาไว้ที่ไดเรคทอรี /root/kernels ด้วยชื่อใหม่ว่า MYKERNAL ด้วยคำสั่ง

   #cp /usr/src/sys/i386/conf/GENERIC /root/kernels/MYKERNEL

3. สร้างแฟ้มลิ้งที่ไดเรคทอรี /usr/src/sys/i386/conf/ ให้ชี้มาที่ /root/kernels/MYKERNEL ด้วยคำสั่ง

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

   #ln -s /root/kernels/MYKERNEL

   ซึ่ง ณ. ขณะนี้สามารถจะทำการแก้ไขแฟ้ม /root/kernels/MYKERNEL เพื่อทำการ

คอนฟิกและปรับปรุงเคอร์แนลให้รองรับกับการใช้งานตามที่ต้องการได้

4. การคอนฟิกเคอร์เนลในแฟ้มคอนฟิกของเคอร์เนลด้วยออฟชันต่อไปนี้

ตารางที่ 4.1 แสดงความหมายของตัวเลือกต่างๆในไฟล์เคอร์เนลที่เกี่ยวกับไฟร์วอลล์

ตัวเลือก	ความหมาย
options IPFIREWALL	เปิดให้มีการใช้งาน IPFW
options IPFIREWALL_FORWARD	ให้ไฟร์วอลล์ทำการส่งต่อ packet
options IPFIREWALL_VERBOSE	ให้เก็บล็อกของไฟร์วอลล์ลง syslog
options IPFIREWALL_VERBOSE_LIMIT=100	กำหนดจำนวนของล็อกที่จะทำการส่งล็อกไปยัง syslog ของระบบ
options IPFIREWALL_DEFAULT_TO_ACCEPT	ใช้ในกรณีต้องการ deny เฉพาะ packets, ports และ protocols ที่กำหนดไว้
Options IPDIVERT	ใช้ในกรณีต้องการให้เครื่องทำหน้าที่เป็นNAT

5. การคอมไพคอร์เนลของ FreeBSD สามารถคอมไพ ด้วยคำสั่งต่อไปนี้

- เข้าไปที่ไปได้เรคทอรี /usr/src/sys/i386/conf ก่อนโดยใช้คำสั่ง

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

- เรียกโปรแกรม config เพื่อทำการคอนฟิกเคอร์แนล

  #/usr/sbin/config MYKERNEL

- เข้าไปที่ไปได้เรคทอรี /usr/src/sys/compile/MYKERNEL

  #cd ../../compile/MYKERNEL

- คอมไพเคอร์เนล

  # make depend

  # make

- ติดตั้งเคอร์เนล

  # make install

6. ทำการแก้ไขแฟ้ม **/etc/rc.conf** ดังนี้

```
firewall_enable="YES"          # Set to YES to enable firewall functionality
firewall_script="/etc/rc.firewall"      # Which script to run to set up the firewall
firewall_type="OPEN"          # Firewall type (see /etc/rc.firewall)
firewall_quiet="NO"           # Set to YES to suppress rule display
```

```
firewall_logging="NO"       # Set to YES to enable events logging
firewall_flags=""           # Flags passed to ipfw when type is a file
```

7. สิ้นสุดขั้นตอนการติดตั้งให้ทำการบูตเครื่องเพื่อใช้งานไฟร์วอลล์ IPFW

หมายเหตุ หลังจากที่คอมไพเคอร์เนลเสร็จแล้วก็ต้องทำการบูตเครื่องเพื่อใช้งานเคอร์เนลที่เราคอมไพใหม่ ซึ่งบางครั้งอาจเกิดความผิดพลาดจากการคอมไพเคอร์เนลทำให้ไม่สามารถใช้งานระบบได้ ซึ่ง FreeBSD ได้สำรองแฟ้ม /kernel.GENERIC ซึ่งเป็นแฟ้มเคอร์เนลค่าเริ่มต้น กับแฟ้ม /kernel.old ซึ่งเป็นแฟ้มเคอร์เนลตัวเก่าที่เราใช้งานอยู่ ดังนั้นหากเกิดความผิดพลาดของระบบและต้องการเรียกใช้งาน เคอร์เนลตัวเก่าหรือว่าค่าดีฟอลเดิมก็จะสามารถทำได้

4.4 การใช้งานไฟร์วอลล์ IPFW

การใช้งาน IPFW เมื่อเข้าระบบได้แล้วควรลองตรวจสอบเน็ตเวิคร์ก่อนว่าสมบูรณ์ดีหรือไม่ โดยลอง ping เครื่องต่างๆในระบบว่าพบหรือไม่ ถ้าทุกอย่างปกติดีก็สามารถใช้คำสั่งต่างๆเพื่อทำงานต่อไป แต่ถ้าหากพบว่าไม่สามารถ ping ได้แสดงว่าไฟร์วอลล์ที่ทำการติดตั้งยังไม่ได้อนุญาตให้ packet ข้อมูลผ่านเข้าออกได้ ซึ่งสามารถแก้ไขโดย

- คำสั่งที่จะทำให้ไฟร์วอลล์ยอมให้ทุก packet ของข้อมูลผ่านได้

  #ipfw add 65000 pass all from any to any

- คำสั่งที่ใช้ในการเรียกดูกฎของไฟร์วอลล์ทั้งหมดสามารถทำได้ 2 วิธีคือใช้คำสั่ง list และคำสั่ง show ดังนี้

  #ipfw show

```
00100  4174     551308      allow ip from any to any via lo0
00200  0        0           deny ip from any to 127.0.0.0/8
00300  0        0           deny ip from 127.0.0.0/8 to any
00400  367703   255889440   allow ip from 192.168.0.0/24 to 192.168.0.0/24
00500  224956   89416080    allow tcp from any to any via tun0
65000  160037   70649800    allow ip from any to any
65535  0        0           deny ip from any to any
```

รูปที่ 4.3 แสดงผลการทำงานคำสั่ง ipfw show

#ipfw list

```
00100     allow ip from any to any via lo0
00200     deny ip from any to 127.0.0.0/8
00300     deny ip from 127.0.0.0/8 to any
00400     allow ip from 192.168.0.0/24 to 192.168.0.0/24
00500     allow tcp from any to any via tun0
65000     allow ip from any to any
65535     deny ip from any to any
```

รูปที่ 4.4 แสดงผลการทำงานคำสั่ง ipfw list

- คำสั่งที่ใช้ในการเพิ่มกฎของไฟวอล์ลโดยมีรูปแบบคือ

  Ipfw add ***หมายเลขกฎ*** [allow หรือ deny] ***โปรโตคอล*** from ***ไอพีแอดเดรสต้นทาง*** (พอร์ตต้นทาง) to ***ไอพีแอดเดรสปลายทาง***(พอร์ตปลายทาง) ตัวอย่างเช่น

  #ipfw add 65000 pass all from any to any

  คำสั่งนี้เป็นการเพิ่มกฎของไฟวอล์ลที่ 65000 คือยอมให้เพคเกจทุกอย่างผ่านไปได้ซึ่งถ้าดูด้วยคำสั่ง ipfw show ก็จะมีกฎหมายเลข 65000 แสดงขึ้นมาดังรูปที่ 4.3
- สำหรับกฎพื้นฐานของไฟวอล์ลของ FreeBSD จะมีกฎหมายเลข 65535 ดังนี้

  65535 0 0 deny ip from any to any ซึ่งก็คือไม่ยอมให้เพคเกจใดๆผ่าน
- คำสั่งที่ใช้ในการเคลียร์ค่าจำนวนของเพคเกจที่วิ่งเข้าออก(ตัวเลขที่อยู่หลังลำดับของกฎ) สามารถเคลียร์ให้เป็น 0 ได้ด้วยคำสั่ง

  # ipfw zero 65000

  หมายถึงการเคลียร์การนับของเพคเกจของกฎหมายเลข 65000 ให้เป็นศูนย์
- คำสั่งที่ใช้ในการลบกฎของไฟร์วอลล์ใช้คำสั่ง delete เช่น

  # ipfw delete 65000

  หมายถึงการลบกฎหมายเลข 65000 ซึ่งถ้าตาม ipfw show นี้คือ

  65000 160037 70649800 allow ip from any to any

4.5 การพัฒนาโปรแกรมกำหนดกฎไฟร์วอลล์

การพัฒนาโปรแกรมกำหนดกฎไฟร์วอลล์ พัฒนาโดยโปรแกรมภาษา PHP โดยให้โปรแกรมที่ได้พัฒนามานั้นทำงานผ่านเว็บบราวเซอร์ โดยโปรแกรมจะมีหน้าที่ดังนี้

1 สามารถป้อนข้อมูลที่จำเป็นสำหรับการสร้างกฎของไฟร์วอลล์ ซึ่งระบุได้ถึง IP Source, Port Source, IP Destination, Port Destination, Protocol, Direction, Interface ฯลฯ

2 สามารถเพิ่มหรือลบกฎที่ทำการสร้างขึ้นได้

3 สามารถแสดงกฎที่สร้างขึ้นใหม่และกฎที่มีอยู่เดิมได้

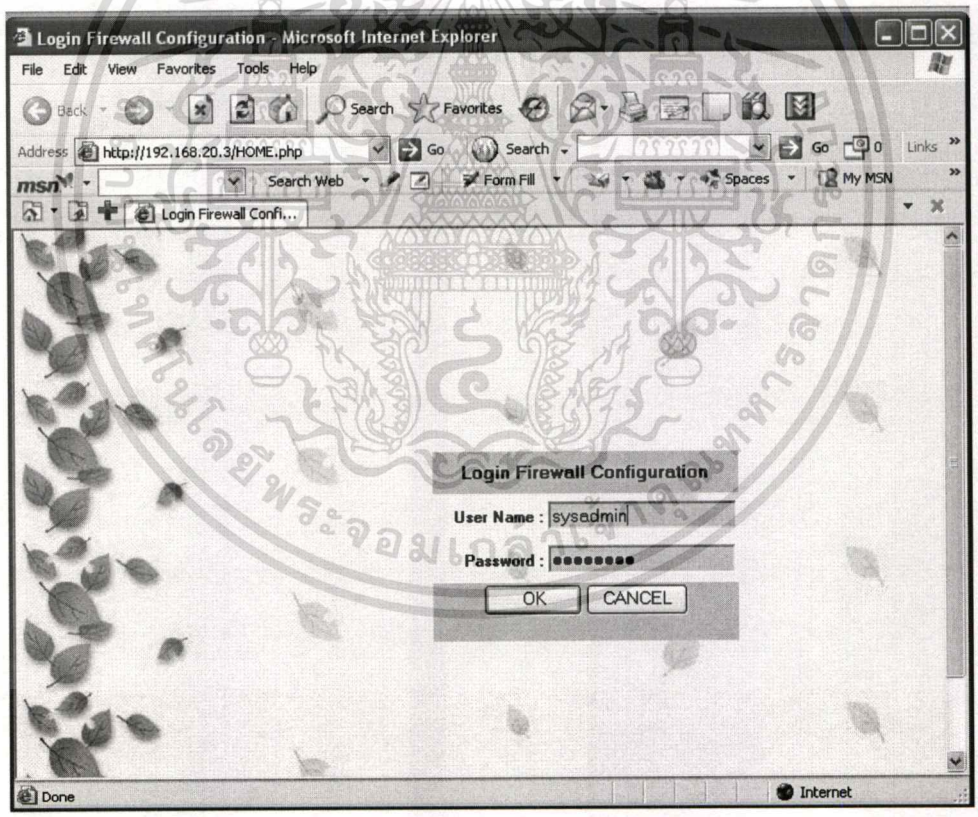
4 สามารถสำรองข้อมูลกฎและเรียกกฎที่มีการใช้งานอยู่เดิมก่อนที่จะทำการแก้ไขเปลี่ยนแปลงขึ้นมาทำงานได้

5 สามารถแสดงผลการทำงานของไฟร์วอลล์ได้

6 สามารถแสดงผลการทำงานของกฎ (Log file) เพื่อใช้วิเคราะห์การทำงานของระบบ

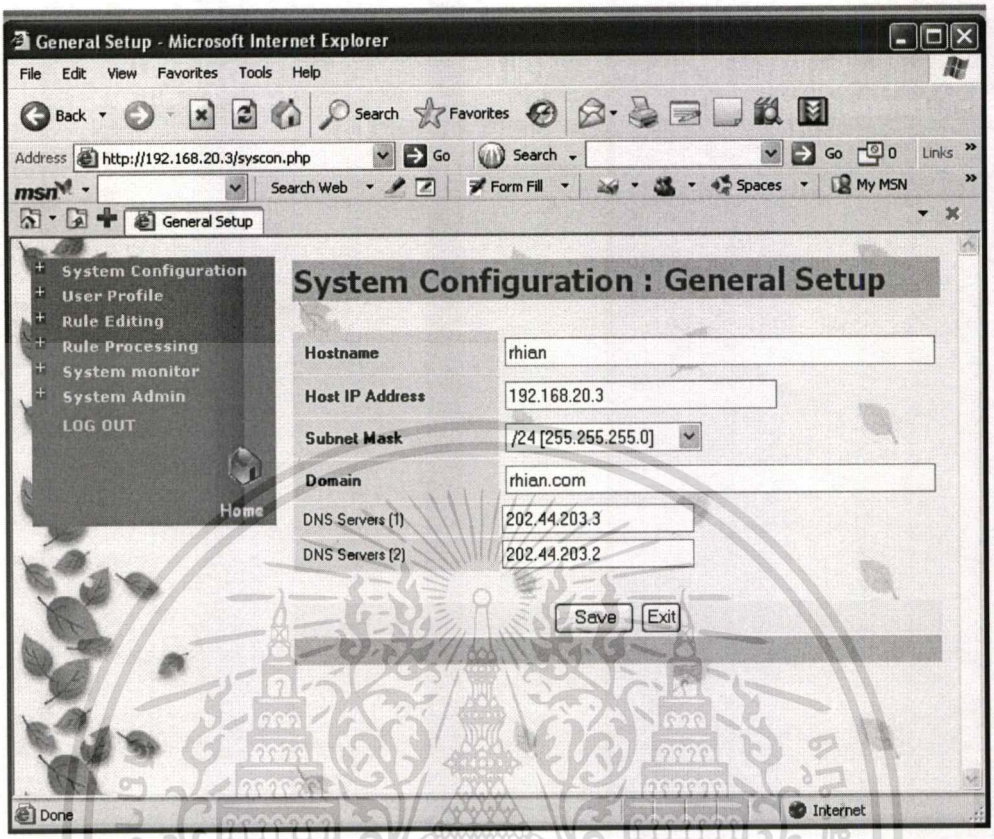
โดยมีรูปแบบการทำงานดังนี้

- ทำการล๊อกอินเข้าสู่ระบบ(จะอยู่ในส่วนของ Login usecase)โดยในครั้งแรกใช้ชื่อรหัสล๊อกอินของระบบเป็น SYSADMIN และรหัสผ่านเป็น “sysadmin” จากนั้นระบบจะตรวจเช็คข้อมูลผู้ใช้(จะอยู่ในส่วนของ create user profile usecase)



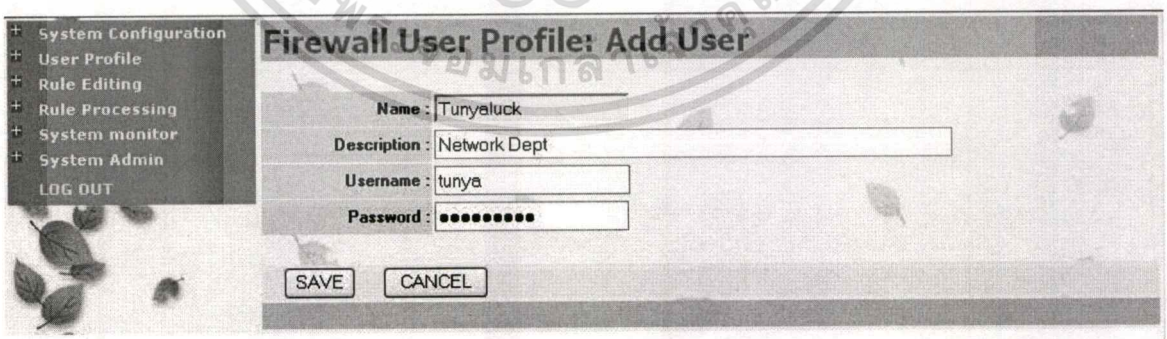
รูปที่ 4.5 แสดงหน้าจอการเรียกใช้งานโปรแกรมในส่วน Login

- จากนั้นทำการป้อนข้อมูลระบบ(ซึ่งจะอยู่ในส่วนของ Cureate system configuration usecase)จากเมนู System Configuration > General Setup



รูปที่ 4.6 แสดงหน้าจอการเรียกใช้งานโปรแกรมในส่วน General Setup

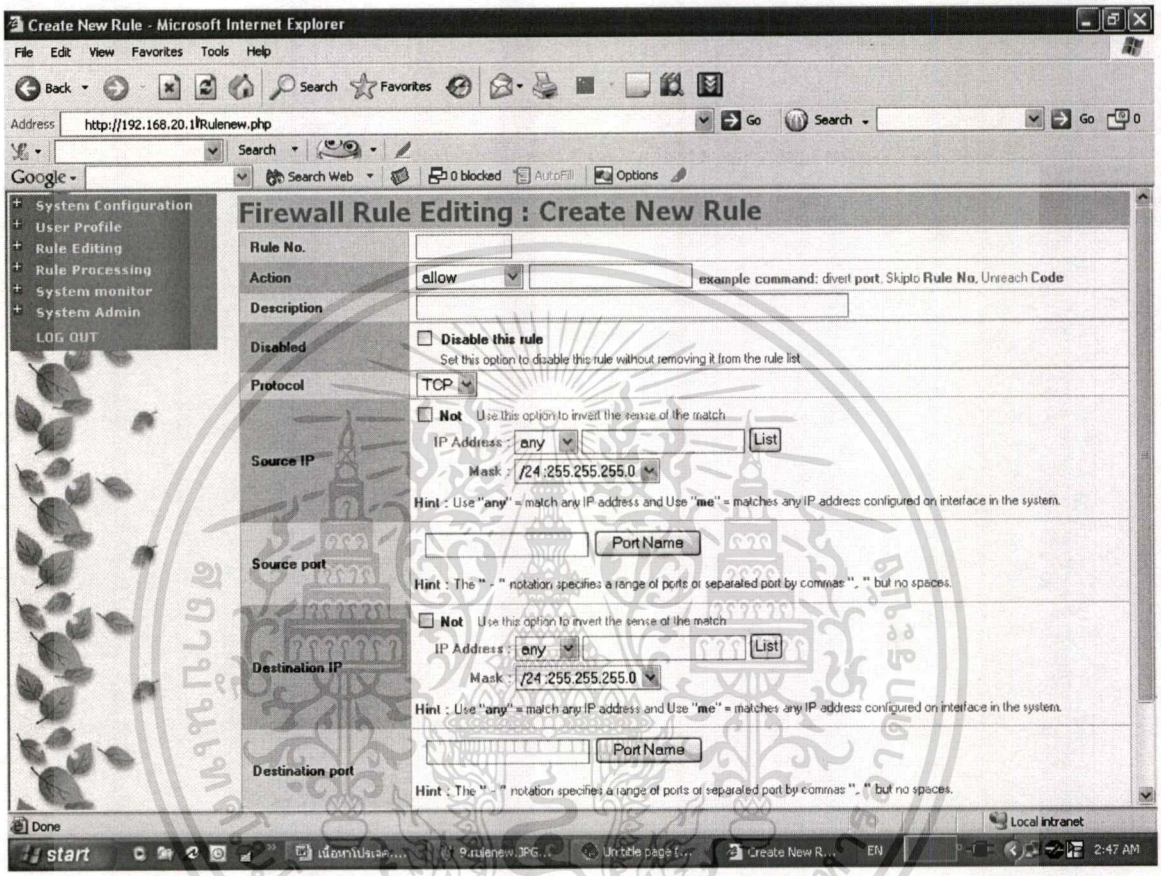
- ผู้ใช้สามารถทำการสร้างผู้ใช้งานระบบ(ซึ่งจะอยู่ในส่วนของ create user profile usecase) ได้จากเมนู User Profile > Add User



รูปที่ 4.7 แสดงหน้าจอการเรียกใช้งานโปรแกรมในส่วนAdd User

- ในการสร้างกฎของไฟร์วอลล์ผู้ใช้สามารถทำการสร้างได้โดยเรียกจากเมนู Rule Editing > Create New Rule ซึ่งจะเป็นการใช้งานในส่วนของ Create New rule usecase และในกรณีที่จะทำการแก้ไขข้อมูลหรือลบข้อมูลกฎ สามารถเรียกได้จากเมนู Rule Editing > List เพื่อทำ

การดูกฎและคลิกเลือกกฎที่จะทำการลบหรือแก้ไขได้ ซึ่งระบบจะทำการเรียกใช้งาน usecase โมดูลในส่วนของ Edit Existing rules และ delete rule usecase



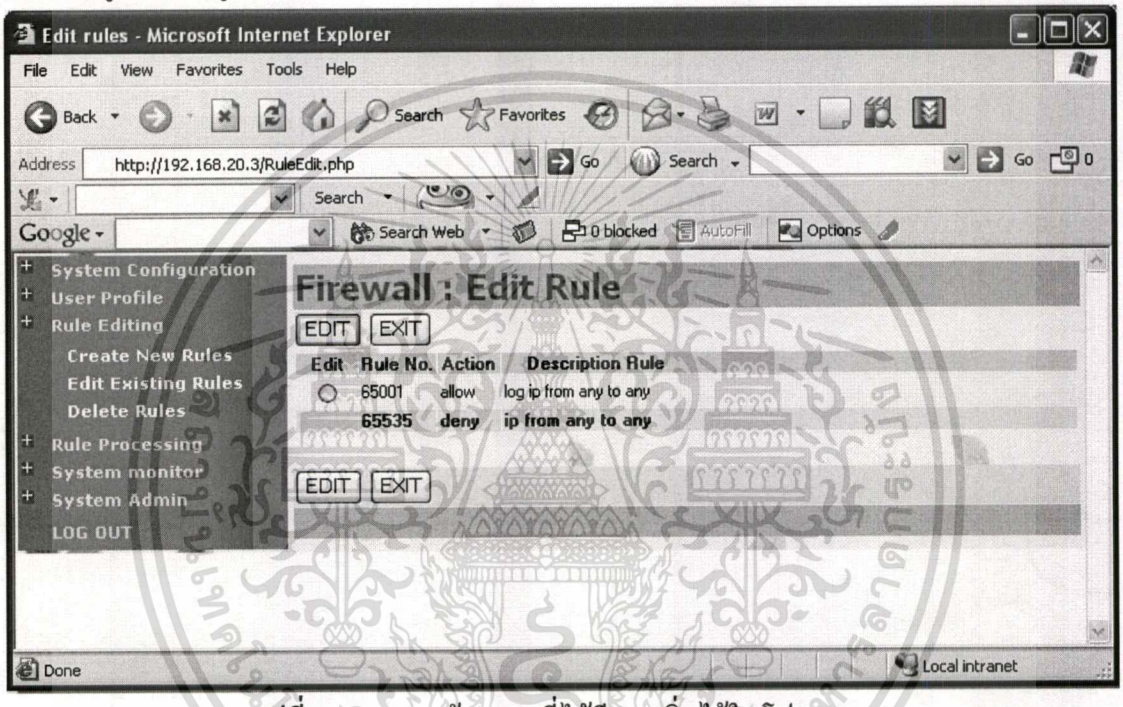
รูปที่ 4.8 แสดงหน้าจอการเรียกใช้งานโปรแกรมในส่วน Create New Rule

- หลังจากทำการสร้างกฎเรียบร้อยแล้วทั้งหมดต้องทำการ Activate การทำงานของกฎทั้งหมดโดยเลือกเมนู Rule Processing > Activate All Rules ซึ่งการทำงานดังกล่าวจะเป็นการเรียกใช้ Activate all rules usecase ขึ้นมาเพื่อใช้งาน
- การสำรองข้อมูลกฎและการโหลดสคริปกฎจะถูกเรียกใช้ได้จากเมนู System Admin โดยจะเป็นการเรียกใช้งาน Backup rules usecase และ Load rules usecase
- สำหรับเมนูอื่นๆจะเป็นในส่วนของการแสดงสถานะการทำงานของไฟร์วอลล์ซึ่งเกิดจากการทำงานในส่วนของ Monitor status rules และการทำงานในส่วนของการดึงกฎเก่ากลับขึ้นมาใช้ซึ่งจะเป็นในส่วนของ Restore rules usecase เป็นต้น

หมายเหตุ รายละเอียดและวิธีการใช้งานของโปรแกรมการกำหนดกฎได้จัดไว้ใน **ภาคผนวก ง**

4.6 ผลการทดสอบการใช้งานโปรแกรมการกำหนดกฎไฟร์วอลล์

การทดสอบการทำงานของโปรแกรมการกำหนดกฎไฟร์วอลล์ จะเริ่มทำการทดสอบโดยวิธีการ ping จากเครื่องลูกข่ายไปยังไฟร์วอลล์ และในระหว่างนั้นจะทำการสร้างกฎเพิ่มเพื่อทดสอบว่า กฎที่ได้ทำการเพิ่มจะสามารถทำการเปิดหรืออนุญาตให้ Packet ต่างๆทำงานได้ โดยจะทำการตรวจสอบเปรียบเทียบการทำงานกับ คำสั่ง ipfw list หรือ ipfw show เพื่อดูผลของกฎที่ได้ทำการป้อนเข้าสู่ระบบดังรูป



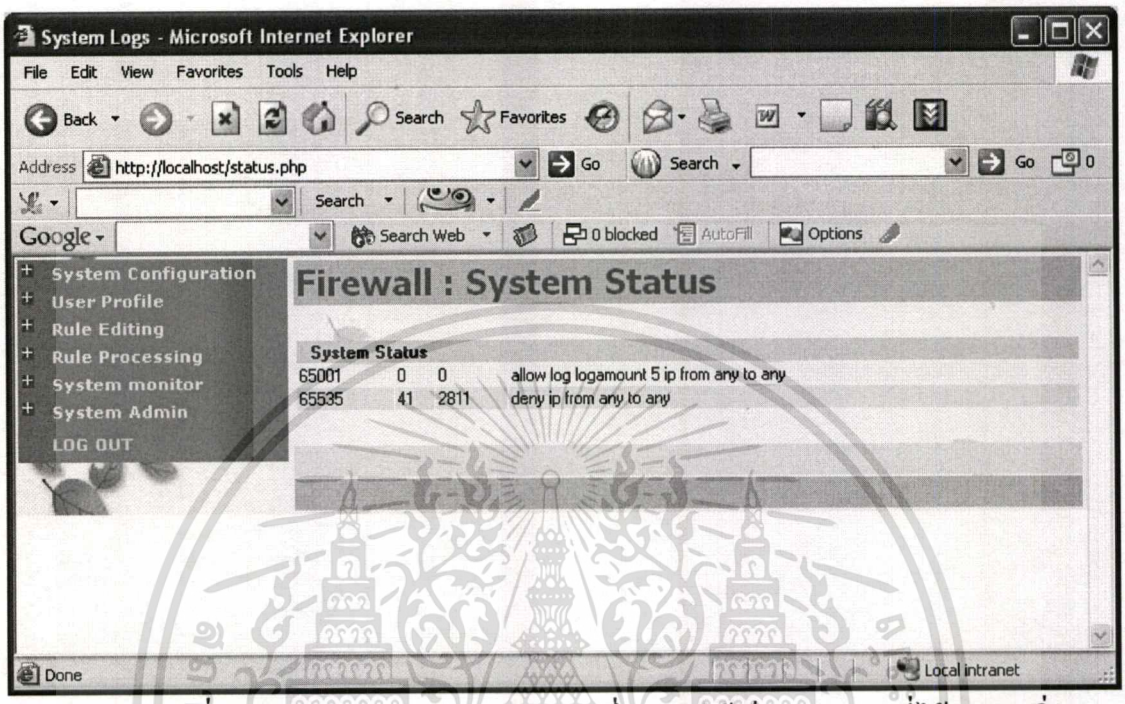
รูปที่ 4.9 แสดงหน้าจอกฎที่ได้มีการเพิ่มไว้ในโปรแกรม

จากการเพิ่มกฎดังนี้คือ รหัสกฎที่ 65001 action “allow” ให้มีการเก็บ Log file โดยอนุญาตให้ packet ทุกชนิดผ่านได้ หลังจากกำหนดโดยการใช้โปรแกรมได้ทำการตรวจสอบว่าได้มีการเพิ่มกฎในไฟล์วอลล์จริงโดยใช้คำสั่ง ipfw show ซึ่งแสดงไว้ดังรูปที่ 4.10

```
tunya# ping 192.168.20.1
PING 192.168.20.1 (192.168.20.1): 56 data bytes
ping: sendto: Permission denied
ping: sendto: Permission denied
ping: sendto: Permission denied
ping: sendto: Permission denied
ping: sendto: Permission denied
^C
--- 192.168.20.1 ping statistics ---
5 packets transmitted, 0 packets received, 100% packet loss
tunya# ipfw show
65001  0     0 allow log logamount 5 ip from any to any
65535 41 2811 deny ip from any to any
```

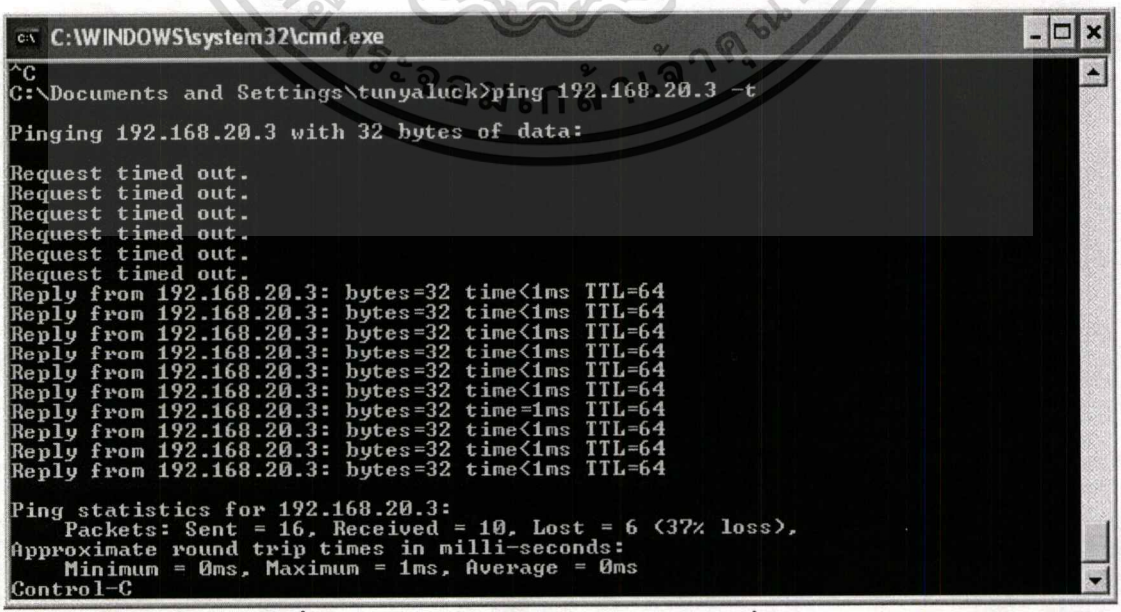
รูปที่ 4.10 แสดงหน้าจอผลคำสั่งกฎที่ได้ทำการเพิ่มไว้ในโปรแกรม

และได้ทำการเปรียบเทียบกับผลการทำงานของระบบได้ผลดังรูปที่ 4.11



รูปที่ 4.11 แสดงหน้าจอ System Status ซึ่งแสดงให้เห็นผลของกฎที่ได้ทำการเพิ่ม

สำหรับการทดสอบการทำงานของกฎ ได้ทำการทดสอบพร้อมกับการเพิ่มกฎเข้าสู่ไฟร์วอลล์ และได้ทดสอบโดยการ ping จากเครื่องลูกข่ายได้ผลดังนี้



รูปที่ 4.12 แสดงหน้าจอผลการทำงานจากเครื่องลูกข่าย

จากผลการทำงานพบว่าก่อนที่จะทำการเพิ่มกฎที่ 65001 เครื่องลูกข่ายไม่สามารถ ping ไปยังไฟร์วอลล์ได้โดยถูกไฟร์วอลล์บล๊อกการติดต่อ แต่หลังจากได้มีการเพิ่มกฎให้อนุญาตให้เครื่องลูกข่ายติดต่อไปยังไฟร์วอลล์ได้จึงเกิดการตอบรับการติดต่อ

สรุปการทำงานของโปรแกรมการกำหนดกฎไฟร์วอลล์สามารถทำการเปิดและปิด อนุญาตและไม่อนุญาตให้ Packet ต่างๆ ได้ตามกฎที่ได้สร้างขึ้นโดยผู้ใช้สามารถควบคุมการใช้งานโดยใช้คำสั่งต่างๆ ตามต้องการ อย่างไรก็ตามผู้ใช้งานควรจะต้องคำนึงถึงนโยบายด้านความปลอดภัยขององค์กรเป็นหลัก เพื่อให้ไฟร์วอลล์นั้นสามารถนำไปใช้งานได้อย่างมีประสิทธิภาพสูงสุด

บทที่ 5
บทสรุปและแนวทางพัฒนาในอนาคต

5.1 ข้อจำกัดของระบบ

เนื่องจากระบบโปรแกรมการกำหนดกฎนี้ได้พัฒนาขึ้นบนพื้นฐานการทำงานผ่านเว็บ เพื่อช่วยให้การทำงานของผู้ดูแลระบบง่ายในการทำงานแต่ทั้งนี้ก็ยังมีข้อจำกัดในการใช้งานอันเนื่องมาจากถ้าผู้กำหนดกฎได้มีการกำหนดกฎให้บล๊อกการทำงานผ่านพอร์ต 80 ซึ่งเป็น port ที่อนุญาตให้การใช้งานโปรแกรมผ่านเว็บสามารถทำงานได้ ระบบโปรแกรมการกำหนดกฎก็จะไม่สามารถทำงานได้

โปรแกรมการกำหนดกฎที่พัฒนาขึ้นนี้ได้มีการกำหนดกฎพื้นฐานของระบบอยู่ 2 ข้อด้วยกันคือ 65000 allow ip any to any และ 65535 deny ip any to any ซึ่งเป็นกฎที่ห้ามทำการลบออกจากระบบการทำงานและจะไม่แสดงบนโปรแกรมการกำหนดกฎยกเว้นเมื่อทำการดูสถานะและล็อกไฟล์ของระบบหรือกรณีที่มีการ activate การทำงานของกฎเท่านั้นจึงจะมีการปรากฏการทำงานแต่ทั้งนี้ระบบก็ได้ทำการล็อกการลบกฎพื้นฐานทั้งสองแล้ว แต่ทั้งนี้เว้นแต่ว่าหากผู้ดูแลระบบใช้ command line ในการ flush กฎทุกๆกฎทิ้งและไม่ได้ทำการกำหนดให้การทำงานของกฎพื้นฐานหลักของระบบทำงานก็จะมีผลต่อโปรแกรมการกำหนดกฎเช่นกัน

5.2 สรุปแนวทางในการพัฒนาในอนาคต

ในการติดต่อสื่อสารผ่านเครือข่ายอินเทอร์เน็ตอาจจะมีเครื่องคอมพิวเตอร์จำนวนมากที่ติดต่อเข้ามาสู่เครือข่ายเพื่อใช้งานในทางที่ดีและถูกต้อง แต่ก็ยังคงมีเครื่องคอมพิวเตอร์และผู้ใช้งานจำนวนไม่น้อยที่มุ่งประสงค์ร้ายต่อเครื่อข่ายและเครื่องคอมพิวเตอร์ที่ต่อเข้าสู่ระบบอินเทอร์เน็ตอยู่ด้วยเช่นกัน ดังนั้นจึงไม่เป็นการปลอดภัยถ้าหากว่าเราได้เชื่อมต่อเครื่องคอมพิวเตอร์ให้บริการเข้าสู่อินเทอร์เน็ตโดยไม่มีการป้องกันใดๆให้กับเครื่องคอมพิวเตอร์ภายในเครือข่ายเหล่านี้ ระบบไฟร์วอลล์จึงเป็นหนึ่งในทางเลือกที่นำมาใช้ในการเพิ่มความปลอดภัยให้กับทรัพย์สินและข้อมูลที่สำคัญภายในระบบเครือข่ายที่ต่อเชื่อมเข้าใช้งานอินเทอร์เน็ต

ระบบไฟร์วอลล์จึงเข้ามามีบทบาทอย่างมากในเรื่องการรักษาความปลอดภัยของเครือข่ายคอมพิวเตอร์โดยมีหน้าที่หลักในการกรอง (filter) ข้อมูลเฉพาะส่วนที่ได้รับอนุญาตเท่านั้น ดังนั้นการเขียนกฎหรือ rule สำหรับไฟร์วอลล์จึงเป็นเรื่องที่สำคัญอย่างยิ่ง การสร้าง rule ของไฟร์วอลล์ที่ผิดพลาดจะทำให้ไฟร์วอลล์ ที่ทั้งราคาแพงและใช้งานฟรี ทั้งหลายไม่สามารถช่วยป้องกันเครือข่ายให้รอดพ้นจากการถูกบุกรุกหรือโจมตีได้อย่างแน่นอน แต่สำหรับไฟร์วอลล์ที่จำหน่ายในเชิง

พาณิชย์ส่วนใหญ่จะมีโปรแกรมสำหรับช่วยในการกำหนดกฎไฟร์วอลล์ซึ่งช่วยในการลดความผิดพลาดอันเนื่องมาจากการสร้างกฎไฟร์วอลล์อยู่แล้ว แต่สำหรับไฟร์วอลล์ที่เป็นชนิด Freeware ส่วนใหญ่มักจะยังไม่มีโปรแกรมสำหรับการคอนฟิกกฎของไฟร์วอลล์จึงทำให้อาจเกิดความผิดพลาดของการกำหนดกฎของไฟร์วอลล์ได้

ดังนั้นการพัฒนาระบบการกำหนดกฎหรือการสร้างกฎของไฟร์วอลล์จึงมีส่วนช่วยให้ไฟร์วอลล์ที่เป็น Freeware สามารถสร้าง rule ที่ถูกต้อง เพื่อเป็นการลดความผิดพลาดอันเนื่องจากการกำหนดกฎของไฟร์วอลล์ลงได้ แต่ทั้งนี้ ผู้ดูแลไฟร์วอลล์จะต้องมั่นใจว่าเครื่องไฟร์วอลล์นั้นมีความปลอดภัยในระดับโฮสต์อยู่แล้ว (host based security) เพราะถึงแม้ว่า rule ที่สร้างขึ้นจะสามารถป้องกันเครื่องอื่นๆ ภายในเครือข่ายได้ แต่ถ้าเครื่องไฟร์วอลล์เองไม่สามารถทนต่อการบุกรุกได้ก็เป็นจุดที่อันตรายไม่ยิ่งหย่อนไปกว่าการสร้างกฎที่ผิดพลาดแต่อย่างใดสำหรับแนวทางที่จะพัฒนาระบบการกำหนดกฎของไฟร์วอลล์นี้ เนื่องจากระบบยังไม่ได้ติดตั้งโปรแกรมเพิ่มเติมในเรื่องของการเข้ารหัสผ่านเว็บจึงอาจจะมีการพัฒนาให้มีกลไกในการป้องกันการใช้งานผ่านเว็บและอาจจะมีการพัฒนาเพิ่มในส่วนของการใช้งานแบบ NAT และมีการใช้งานที่เร็วขึ้นและง่ายพร้อมทั้งกระทัดรัดขึ้นมากกว่านี้ และอาจมีการปรับให้สามารถใช้งานได้ในกรณีที่ระบบทำการสร้างกฎที่ทำการบล็อกตัวเองโดยอาจทำให้ระบบทำการหน่วงเวลาการทำงาน โดยถ้าหากระบบไม่ได้ทำการติดต่อกับเซิร์ฟเวอร์เป็นเวลาหนึ่งก็ทำการเชื่อมต่อระบบอีกครั้งหนึ่งหรืออาจพัฒนาให้มีการทำงานกับซอฟต์แวร์ไฟร์วอลล์ได้หลายชนิดมากยิ่งขึ้น

บรรณานุกรม

กิตติ ภักดีวัฒนะกุล และ กิตติพงษ์ กลมกล่อม "UML- วิเคราะห์และออกแบบระบบเชิงวัตถุ" เคทีพี คอมพ์ แอนด์ คอนซัลท์ 2544.

กิตติพงษ์ สุวรรณราช "การบริหารและจัดการเครือข่ายอินเทอร์เน็ตด้วยระบบปฏิบัติการ FreeBSD" ออฟเซ็ทเพรส ตุลาคม 2547

Booch, G., Jacobson, I., and Rumbaugh, J. 1997. "The UML specification documents." Santa Clara. CA.: Rational Software Corp.

Chris, H. and Karanjit, S. 1996. "Internet firewall and Network Security." **2nd Edition Newriders.**

Daniel Boulet. 2003. "IPFIREWALL(4)." **FreeBSD Kernel Interfaces Manual.**

Fortinet Inc. 2004. "Comprehensive Solutions for Real Time Network Protection." **Fortinet Brochure.**

Fortinet Inc. 2004, 30 March. "Forticlient user guide." **User Guide Manual Version 1.0.**

Fortinet Inc. 2004, 30 March. "Forticlient Host Security." **FortiClientVPN datasheet.**

Packeteer Inc. 2003, 11 March. "Packeteer's PacketShaper." **Packet Shaper datasheet.** 1107.J.

Packeteer Inc. 2003. "Four Steps to Application Performance Across the Network." **Packeteer Technical Product Overview.**

Packeteer Inc. 2004, May. "PacketShaper and PacketShaper/ISP." **PacketWise v6.2.0/6.2.1** P/N 20-0153-05 Revision A.

The FreeBSD Documentation Project. 1999, February. "FreeBSD Handbook." **FreeBSD 4.10 RELEASE and FreeBSD 5.2.1 RELEASE.**

The Object Management Group, Inc., "OMG Unified Modeling Language Specification." OMG-UML Manual Version 1.4, September 2001.

Ugen J. S. Antsilevich. 2001. "IPFW(8)." **FreeBSD System Manager's Manual.**

Unified Modeling Language Resource Center. http://www.rational.com/uml.

ภาคผนวก ก

1. รายละเอียดลำดับขั้นตอนในการติดตั้ง

ลำดับ	ขั้นตอนการทำงาน	หมายเหตุ
1	ติดตั้งระบบปฏิบัติการ FreeBSD 5.4	
2	ติดตั้ง Apache Web Server version 2.0.53	ติดตั้งให้สามารถทำงานร่วมกับ PHP และ MySQL
3	ติดตั้ง MySQL version 4.0	
4.	ติดตั้ง PHP version 5	ติดตั้งให้สามารถทำงานร่วมกับ MySQL ได้
5.	สร้างไดเรคทอรี่และไฟล์ที่จำเป็นสำหรับระบบ	

2. แหล่งที่มาของโปรแกรม

ลำดับ	โปรแกรม	สถานที่ดาวน์โหลด
1	ระบบปฏิบัติการ FreeBSD	http://www.freebsd.com
2	MySQL Server	http://www.mysql.com
3	PHP Extension	http://www.php.net
4	Apache Web Server	http://www.apache.org

ภาคผนวก ข

ตารางแสดงค่า CIDR: - Class Inter Domain Routing

Subnet Mask	CIDR_ Prefix3	Total_IP's	Usable_IP's	Number of Class C networks
255.255.255.255	/32	1	1	1/256th
255.255.255.254	/31	2	0	1/128th
255.255.255.252	/30	4	2	1/64th
255.255.255.248	/29	8	6	1/32nd
255.255.255.240	/28	16	14	1/16th
255.255.255.224	/27	32	30	1/8th
255.255.255.192	/26	64	62	1/4th
255.255.255.128	/25	128	126	1 half
255.255.255.0	/24	256	254	1
255.255.254.0	/23	512	510	2
255.255.252.0	/22	1024	1022	4
255.255.248.0	/21	2048	2046	8
255.255.240.0	/20	4096	4094	16
255.255.224.0	/19	8192	8190	32
255.255.192.0	/18	16,384	16,382	64
255.255.128.0	/17	32,768	32,766	128
255.255.0.0	/16	65,536	65,534	256
255.254.0.0	/15	131,072	131,070	512
255.252.0.0	/14	262,144	262,142	1024
255.248.0.0	/13	524,288	524,286	2048
255.240.0.0	/12	1,048,576	1,048,574	4096
255.224.0.0	/11	2,097,152	2,097,150	8192
255.192.0.0	/10	4,194,304	4,194,302	16,384
255.128.0.0	/9	8,388,608	8,388,606	32,768
255.0.0.0	/8	16,777,216	16,777,214	65,536
254.0.0.0	/7	33,554,432	33,554,430	131,072
252.0.0.0	/6	67,108,864	67,108,862	262,144
248.0.0.0	/5	134,217,728	134,217,726	1,048,576
240.0.0.0	/4	268,435,456	268,435,454	2,097,152
224.0.0.0	/3	536,870,912	536,870,910	4,194,304
192.0.0.0	/2	1,073,741,824	1,073,741,822	8,388,608
128.0.0.0	/1	2,147,483,648	2,147,483,646	16,777,216
0.0.0.0	/0	4,294,967,296	4,294,967,294	33,554,432

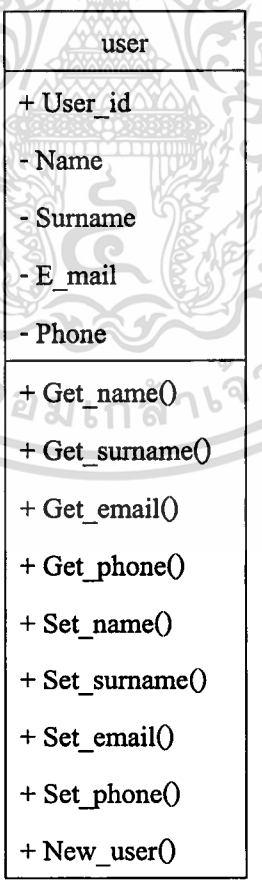
ภาคผนวก ค

หลักการแปลงคลาสไปเป็นตาราง

การแปลงคลาสไปเป็นตาราง คือ การวิเคราะห์จากความสัมพันธ์ เช่น แอสโซซิเอชั่น, เจเนอรัลไรเซชั่นและคอมโพสซิชัน เพื่อแปลงเป็นตาราง

- **หลักในการแปลงคลาสให้เป็นตาราง**

1. กำหนดให้แอตทิบิวต์ตัวใดตัวหนึ่งหรือกลุ่มใดกลุ่มหนึ่งเป็นคีย์หลัก
2. สร้างตารางที่มีทุกแอตทิบิวต์ของคลาสนั้นและมีคีย์หลักตามที่กำหนดแล้ว
3. แอตทิบิวต์หรือกลุ่มของแอตทิบิวต์ที่เป็นคีย์หลักต้องถูกกำหนดเป็น Not Null เสมอ
4. สำหรับแอตทิบิวต์อื่นๆ ที่ไม่ได้ถูกเลือกให้เป็นคีย์หลัก ให้พิจารณาว่าแอตทิบิวต์ใดเป็นนัล(Null) ได้และแอตทิบิวต์ใดเป็นนัลไม่ได้
5. ในการออกแบบตารางไม่ต้องสนใจในส่วนของฟังก์ชั่น ให้สนใจที่แอตทิบิวต์เท่านั้น



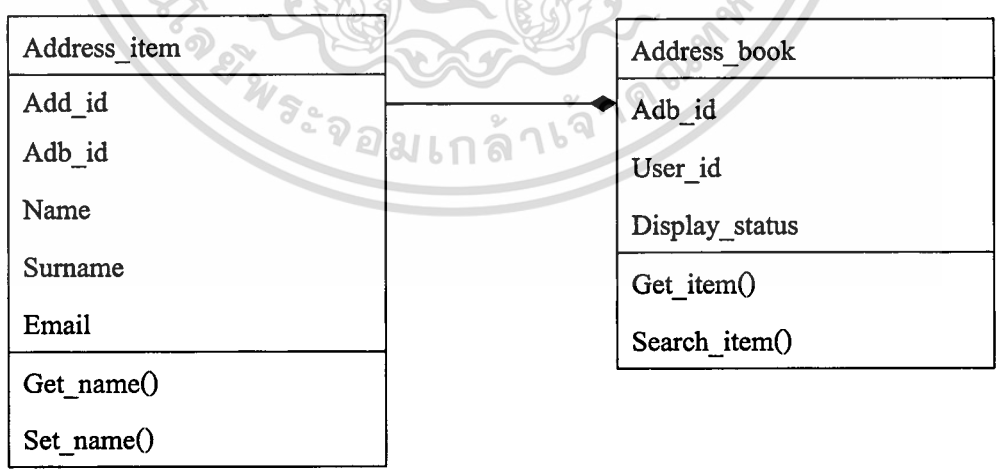
รูปที่ 1 แสดงคลาสตัวอย่าง 1

จากรูป 1 สามารถสร้างตารางของคลาสได้ดังนี้

```
Create Table User
( User_id        Interger(4) Not Null,
  Name           Varchar(50) Not Null,
  Surname        Varchar(50) Not Null,
  E_mail         Varchar(50) Not Null,
  Phone          Varchar(20),
  Primary Key  User_id
)
```

- **หลักการแปลงคลาสที่มีความสัมพันธ์แบบแอกรีเกรชันให้เป็นตารางที่สัมพันธ์กัน**

1. ออกแบบตารางจากคลาสทั้งสองของเครื่องหมายแอกกรีเกรชัน
2. การแสดงความสัมพันธ์ของตารางนั้น ให้นำเอาคีย์หลัก(อาจเป็นฟิลด์เดียวหรือกลุ่มของฟิลด์เดียวก็ได้) ของคลาสหลักมาเป็นคีย์รองของคอมโพสซิสคลาส (Composite Class) ซึ่งก็เหมือนกับการเอาชื่อพ่อแม่ไปเก็บไว้ที่ลูก
3. ในการใส่คีย์รองเข้าไปยังคอมโพซิสคลาสนั้น ต้องพิจารณาด้วยว่าคีย์รองนั้น เป็นค่า Null หรือไม่



รูปที่ 2 แสดงคลาสตัวอย่าง 2

จากรูป 2 สามารถสร้างตารางได้ ดังนี้

```
Create Table Adress_item
(Adb_id                 Int(4) Not Null,
 Add_id                 Int(4) Not Null,
 Name                   Varchar(50) Not Null,
 Surname                Varchar(50) Not Null,
 Email                  Varchar(50) Not Null,
 Primary Key    Add_id,
 Foreign Key    Adb_id References Address_book(Adb_id)
)
```

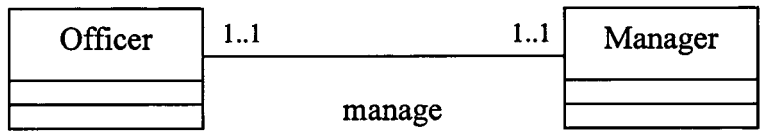
- **หลักการแปลงคลาสที่มีความสัมพันธ์แบบแอสโซซิเอชันให้เป็นตารางที่สัมพันธ์กัน**

1. **แอสโซซิเอชันแบบ 1:1**
   - ออกแบบตารางของคลาสทั้งสองข้างของเครื่องหมายแอสโซซิเอชัน
   - ให้เลือกเอาคีย์หลักของตารางตัวใดก็ได้เป็นคีย์รองของอีกตารางหนึ่ง

รูปที่ 3 แสดงความสัมพันธ์แบบแอสโซซิชัน

จากรูป 3 สามารถสร้างตาราง Man และ Woman ที่สัมพันธ์กันดังนี้

```
Create Table Man
( ManId         Char(10)        Not null,
  WomanId       Char(10),
  Primary Key   ManId,
  Foreign Key    Woman_Id References Woman(Woman_Id))

Create Table Woman
( WomanId       Char(10)        Not null
  Primary Key   WomanId)
```



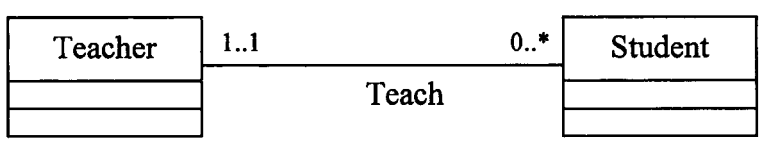
รูปที่ 4 แสดงความสัมพันธ์แบบแอสโซซิชัน แบบ 1:1

จากรูป 4 สามารถสร้างตาราง Manager และ Office ที่สัมพันธ์กันดังนี้

```
Create Table Manager
( ManGid        Char(10)        Not null,
  OfficeId      Char(10)        Not null
  Primary Key   ManGid
  Foreign Key   OfficeId References Office(OfficeId))

Create Table Office
( OfficeId      Char(10)        Not null
  Primary Key   OfficeId)
```

2. แอสโซซิเอชันแบบ 1:N

หลักในการสร้างตารางจากแอสโซซิเอชัน 1:N นั้นมีหลักการเดียวกับการสร้างตารางจากแอกกรีเกรชันโดยตารางในด้าน 1 จะเหมือนกับตารางของคลาสหลักของแอกกรีเกรชัน นั่นคือ ให้เอาคีย์หลักของตารางในด้าน 1 ไปเป็นคีย์รองของตารางในด้าน N ดังรูปที่ 5



รูปที่ 5 แสดงความสัมพันธ์แบบแอสโซซิชัน แบบ 1:N

จากรูป 5 สามารถสร้างตาราง Teacher และ Student ได้ดังนี้

```
Create Table Student
```

```
( StdId          Char(10)        Not null,
  TchId          Char(10)        Not null
  Primary Key    StdId
  Foreign Key    TchId References Teacher(TchId))

Create Table Teacher
( TchId          Char(10)        Not null
  Primary Key    TchId)
```

3. แอสโซซิเอชันแบบ N:N

หลักในการสร้างตารางจากคลาสที่มีความสัมพันธ์แบบ N:N มีหลักการดังนี้

1. สร้างตารางของคลาสทั้งสองข้างของแอสโซซิเอชัน
2. สร้างตารางอีกหนึ่งตาราง ที่มีอย่างน้อย 2 คอลัมน์ ซึ่งก็คือคีย์หลักของตารางทั้งสองและให้คอลัมน์ทั้งหมดเป็นคีย์ของตาราง ดังกล่าว ซึ่งจะเรียกว่าเป็นตารางแอสโซซิเอชัน
3. ให้ส่วนหนึ่งของคีย์หลักที่เป็นคีย์หลักของตาราง ข้างใดข้างหนึ่งเป็นคีย์รองอ้างอิงไปยังตารางนั้นๆ ดังรูปที่ 6

รูปที่ 6 แสดงความสัมพันธ์แบบแอสโซซิชัน แบบ N:N

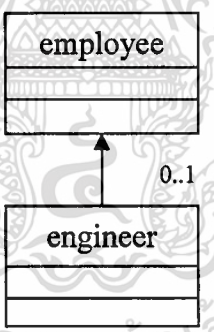
จากรูป 6 สามารถสร้างตาราง Student และ Subject ได้ดังนี้

```
Create Table Student
( StdId          Char(10)        Not null,
  Primary Key    StdId)

Create Table Subject
( SubId          Char(10)        Not null,
  Primary Key    StdId)
```

```
Create Table Std_Sub
( StdId          Char(10),
  SubId          Char(10),
  Primary Key    (StdId, SubId),
 Foreign Key     StdId References Student
 Foreign Key     SubId References Subject)
```

- **หลักการแปลงคลาสที่มีความสัมพันธ์กันแบบเจเนอรัลไรเซชันให้เป็นตารางที่สัมพันธ์กัน**

1. ในกรณีที่เจเนอรัลไรเซชันที่เกิดขึ้นเป็นแบบโททัล-โอเวอร์แลปปิ้ง, พาร์เชียลโอเวอร์แลปปิ้ง (Total-Overlapping, Partial-Overlapping) ให้สร้างตารางของซุปเปอร์คลาสและของทุกๆ คลาสย่อย (ใช้หลักการแอสโซซิเอชันแบบ 1:1) โดยให้สร้างคีย์รองไว้ที่ตารางของคลาสย่อยและคีย์รองนั้น ต้องกำหนดให้เป็น Not null และเป็นคีย์หลักในตารางย่อยด้วย ดังรูปที่ 7



รูปที่ 7 แสดงความสัมพันธ์แบบเจเนอรัลไรเซชันชนิดพาร์เชียลโอเวอร์แลปปิ้ง

จากรูป 7 สามารถสร้างตาราง employee และ engineer ที่สัมพันธ์กันดังนี้

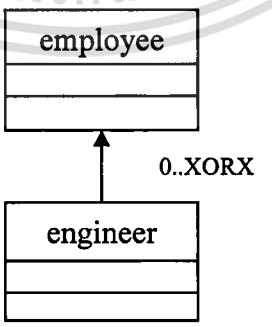
```
Create Table employee
( EmpId          Char(10)        Not null,
  Name           Char(30)        Not null,
  Surname        Char(30)        Not null,
  Primary Key    ManId)
```

```
Create Table engineer
( EngId        Char(10)        Not null,
  MemId        Char(10)        Not null,
  EmpId        Char(10)        Not null,
  Engtype      Char(10)        Not null,
  Foreign Key  EmpId References employee(EmpId))
  Primary Key  (EngId,MemId))
```

2. ในกรณีที่เจเนอรัลไรเซชันที่เกิดขึ้น เป็นแบบโททัลเอ็กซ์คลูซีฟ (Total-Exclusive), พาเชียลเอ็กซ์คลูซีฟ (Partial Exclusive) ให้สร้างตารางของซุปเปอร์คลาสและของทุกๆ คลาสย่อย (ใช้หลักการแอสโซซิเอชันแบบ 1:1) หลังจากนั้นให้สร้าง ตารางแอสโซซิเอชัน ซึ่งมีฟิลด์เดียว คือ คีย์หลักและดำเนินการ ดังนี้

- ในกรณีที่เป็นโททัลเอ็กซ์คลูซีฟ ให้เอาคีย์รองของตารางของคลาสย่อยซึ่งเป็นตัวเดียวกันกับคีย์หลักอ้างอิงมาที่คีย์หลักของตารางแอสโซซิเอชันที่สร้างขึ้น แต่ถ้าเป็นกรณีพาเชียลเอ็กซ์ คลูซีฟให้เพิ่มคีย์รองซึ่งจะมีค่าเดียวกันกับคีย์หลักเสมอ หรือมีค่าเป็นนัลก็ได้ ให้อ้างอิงไปยังตารางแอสโซซิเอชัน

- ให้สร้างคีย์รองเพิ่มเข้าไปยังตารางของซุปเปอร์คลาสอ้างอิงมายังตารางแอสโซซิเอชัน โดยมีเงื่อนไขว่า คีย์รองที่มีในตารางแอสโซซิเอชันนั้น จะสามารถมีค่าเป็นนัลได้ ในกรณีที่เป็นพาเชียลเอ็กซ์ คลูซีฟและถูกตั้งเป็น Not Null ในกรณีเป็นโททัลเอ็กซ์คลูซีฟ ดังรูปที่ 8



รูปที่ 8 แสดงความสัมพันธ์แบบเจเนอรัลไรเซชันชนิดพาร์เชียลเอ็กซ์คลูซีฟ

จากรูป 8 สามารถสร้างตาราง employee และ engineer ที่สัมพันธ์กันดังนี้

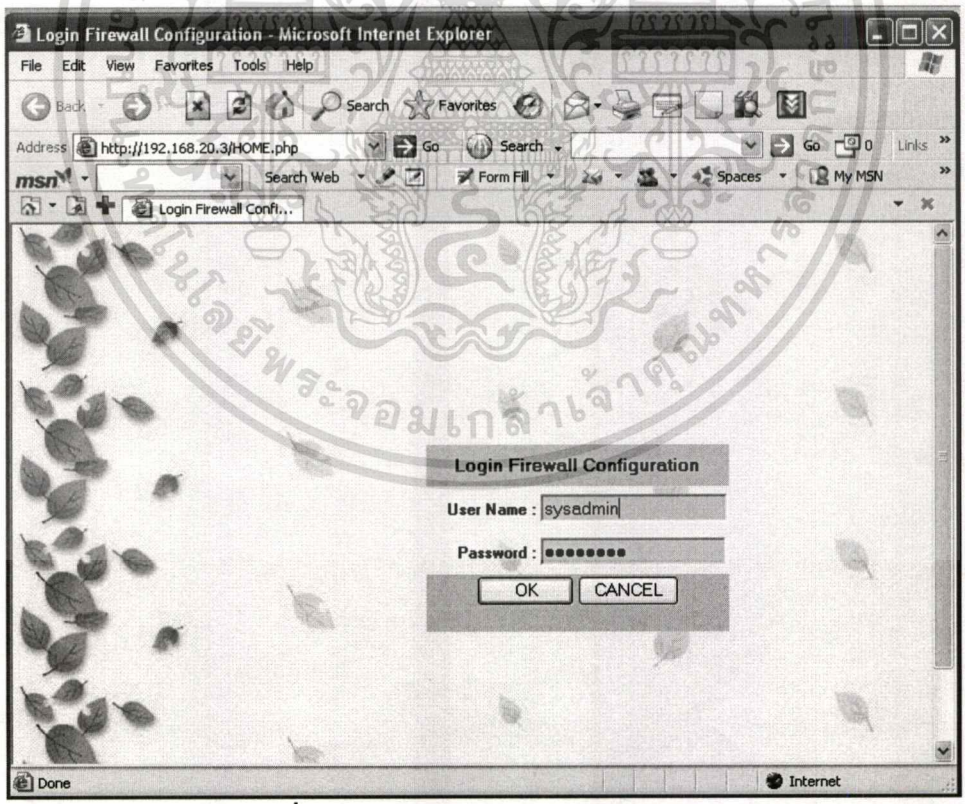
```
Create Table employee
( EmpId        Char(10)        Not null,
  AltId        Char(10)        Not null,
  Name         Char(30)        Not null,
  Surname      Char(30)        Not null,
  Primary Key  ManId)

Create Table Sub_employee
( SubId        Char(10)        Not null,
  EngId        Char(10)        Not null,
  AltId        Char(10)        Null,
  Foreign Key  EngId References engineer(EngId))
  Foreign Key  AltId  References employee(AltId))
  Primary Key  SubId)

Create Table engineer
( EngId        Char(10)        Not null,
  EmpId        Char(10)        Not null,
  Engtype      Char(10)        Not null,
  Foreign Key  EmpId References employee(EmpId))
  Primary Key  EngId)
```

ภาคผนวก ง

คู่มือ การใช้งานโปรแกรมการกำหนดกฎไอพีไฟร์วอลล์

วิธีการติดตั้งโปรแกรม

1. ติดตั้งโปรแกรมกำหนดกฎโดยทำการสร้างโฟรเดอร์ fireweb ไว้ในไดเรกทอรี่ /usr/local/apache/htdocs
2. ทำการสำเนาไฟล์ที่ใช้งานทั้งหมดไปไว้ในโฟรเดอร์ที่สร้างขึ้น
3. การเรียกการใช้งานโปรแกรมให้เรียกใช้งานโดยพิมพ์ url โปรแกรม ดังนี้ “http://local_ip/home.php” โดยค่า local_ip เป็นค่าของ ip address ของเซิร์ฟเวอร์ หลังจากพิมพ์ url และ Enter จะเข้าสู่ระบบการใช้งานดังรูปที่ 1
4. พิมพ์ชื่อล๊อกอิน “SYSADMIN”และรหัสผ่าน “sysadmin” ดังรูปที่ 1



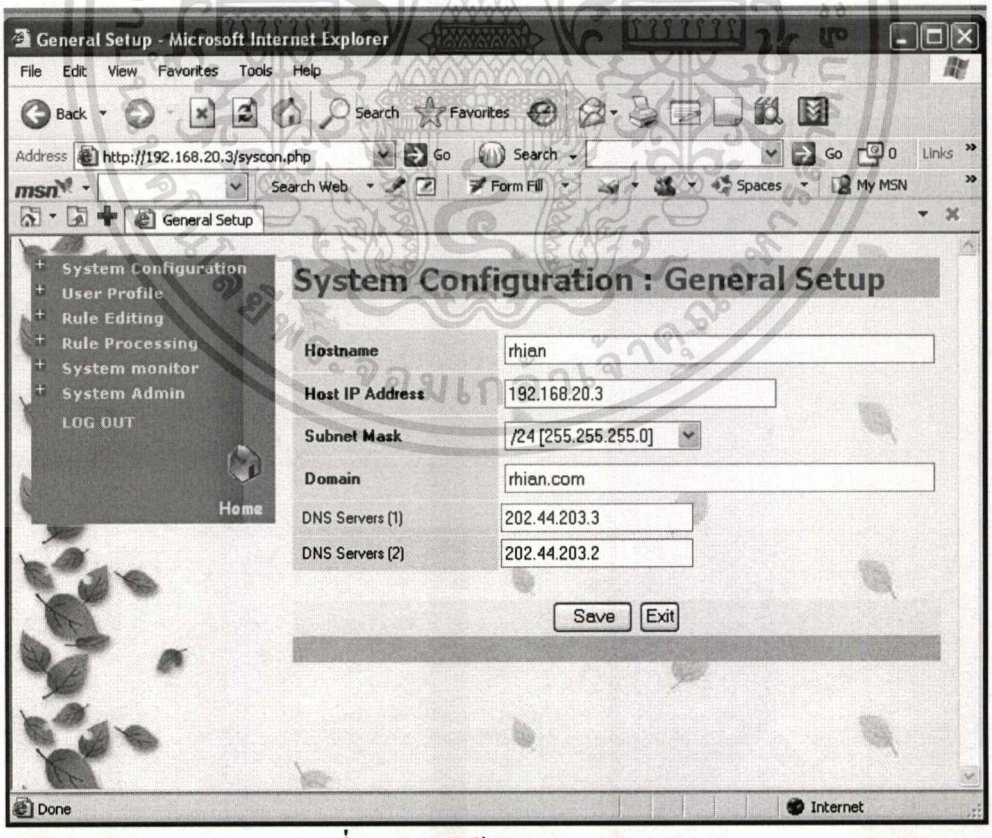
รูปที่ 1 แสดงหน้าจอการเรียกใช้โปรแกรม

5. โปรแกรมกำหนดกฎจะแบ่งการใช้งานหลักๆ ออกเป็น 6 เมนู คือ
    1. เมนูกำหนดค่าของระบบ (System Configuration)
    2. เมนูการเพิ่มชื่อผู้ใช้งานระบบ (User Profile)
    3. เมนูการสร้างกฎของไฟร์วอลล์ (Rule Editing)
    4. เมนูการกำหนดให้กฎทั้งหมดทำงาน (Rule Processing)
    5. เมนูแสดงการทำงานของกฎ (System Monitor)
    6. เมนูคำสั่งพิเศษ (System Admin)

การกำหนดค่าของระบบ (System Configuration)

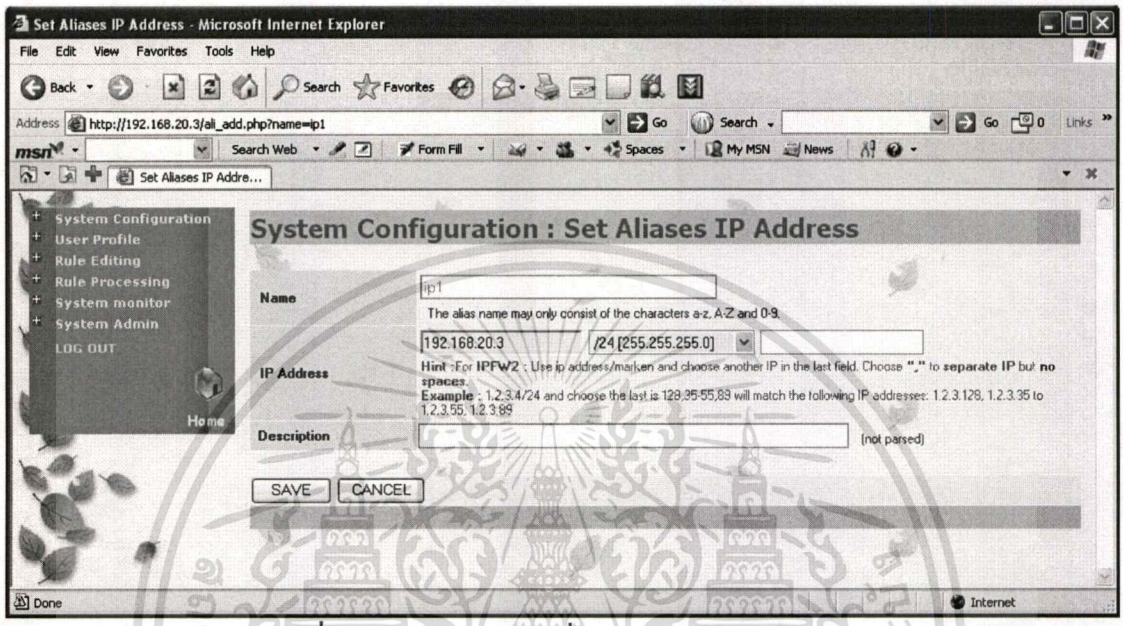
โดยสามารถกำหนดค่าของระบบได้จากเมนู System Configuration โดยสามารถเลือกการใช้งานได้ดังนี้

- **General Setup** จะเป็นการกำหนดค่าพื้นฐานของระบบ ได้แก่ ชื่อโฮสต์ ไอพีแอดเดรสและ Subnet Mask ชื่อโดเมน และชื่อ DNS เซิร์ฟเวอร์ ดังรูป

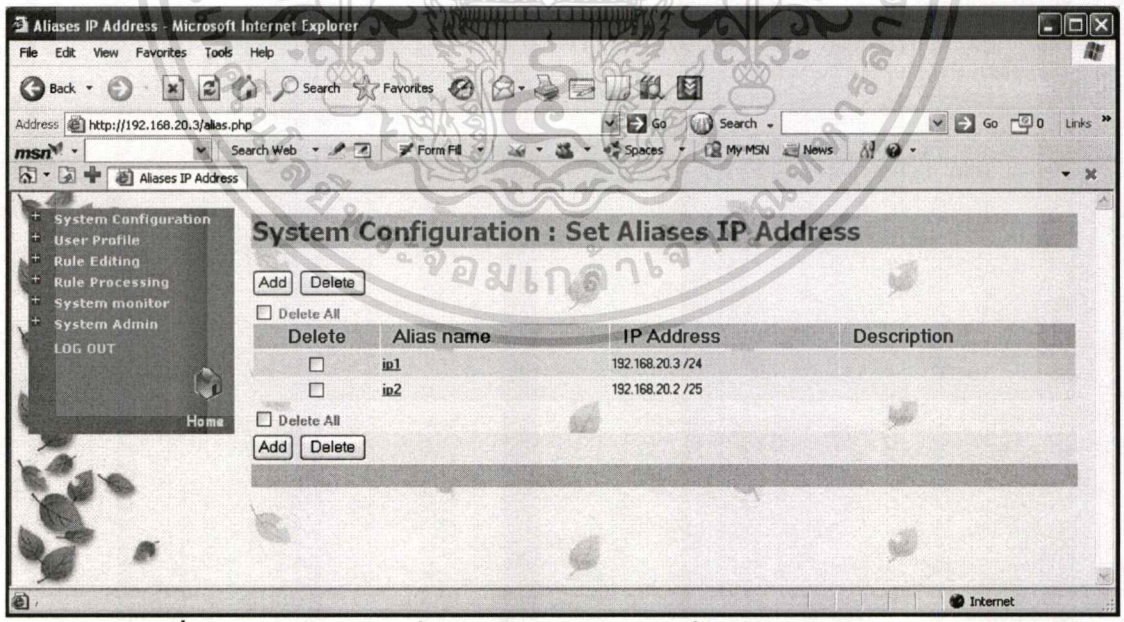


รูปที่ 2 แสดงหน้าจอ General Setup

○ **Setup Aliases IP** จะเป็นการกำหนดชื่อของกลุ่มของ IP Address ที่จะให้กฎทำการตรวจสอบดังรูป



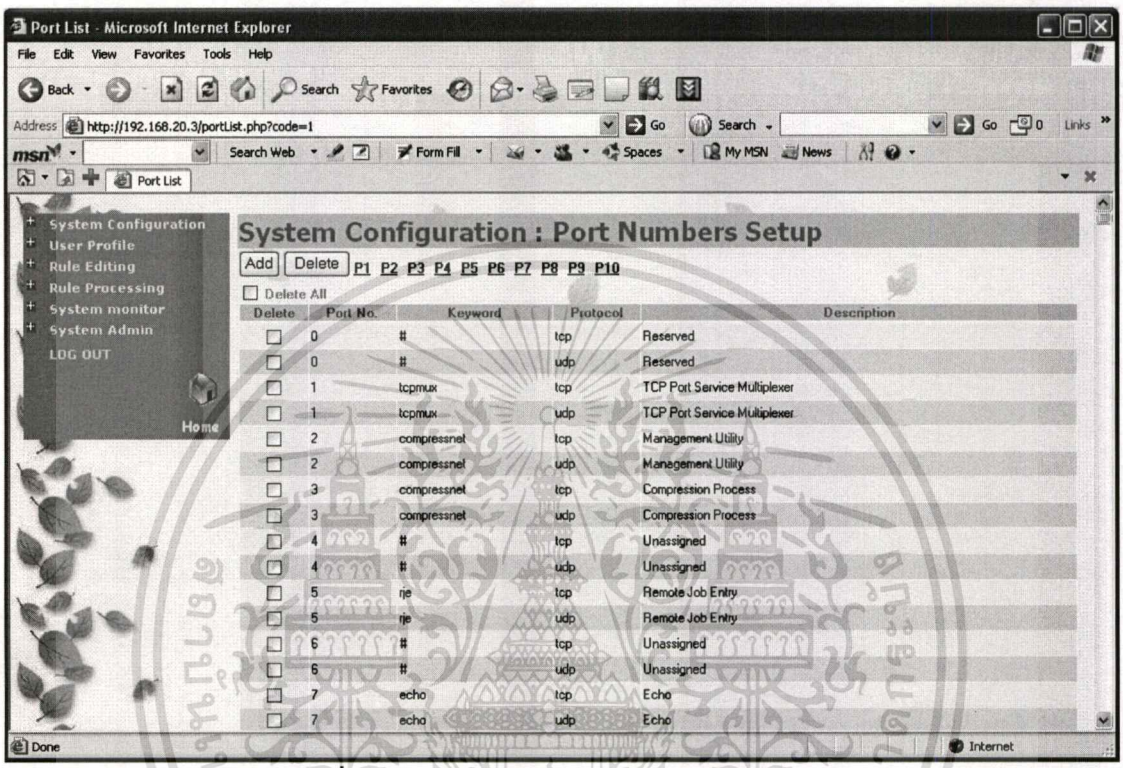
รูปที่ 3 แสดงหน้าจอการเพิ่มค่ากลุ่มของ IP Address



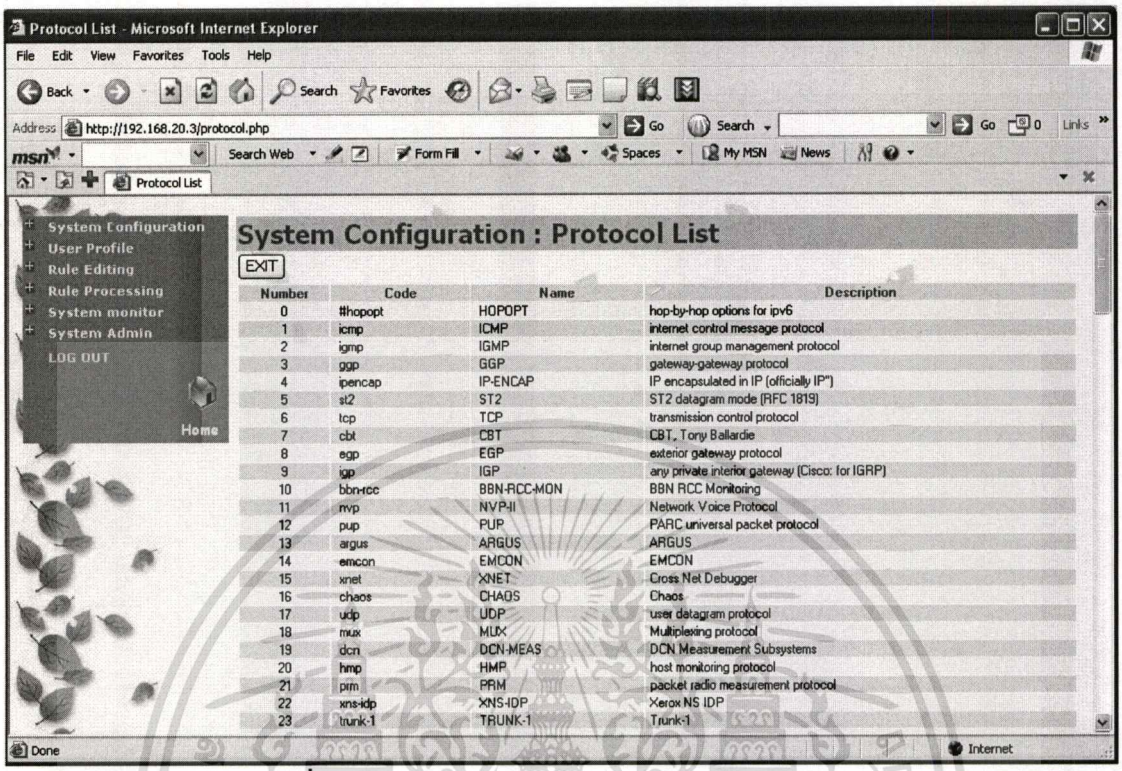
รูปที่ 4 แสดงหน้าจอรายชื่อกลุ่มของ IP Address ที่จะให้กฎทำการตรวจสอบ

○ **Port List** จะเป็นรายละเอียดของพอร์ตต่างๆ ที่มีการใช้งานได้ภายในระบบซึ่งระบบได้มีการกำหนดค่ามาตราฐานมาแล้ว

○ **Protocol List** จะเป็นรายละเอียดของโปรโตคอลต่างๆ ที่มีการใช้งานได้ภายในระบบซึ่งระบบได้มีการกำหนดค่ามาตราฐานมาแล้ว



รูปที่ 5 แสดงหน้าจอรายละเอียดของพอร์ตต่างๆ



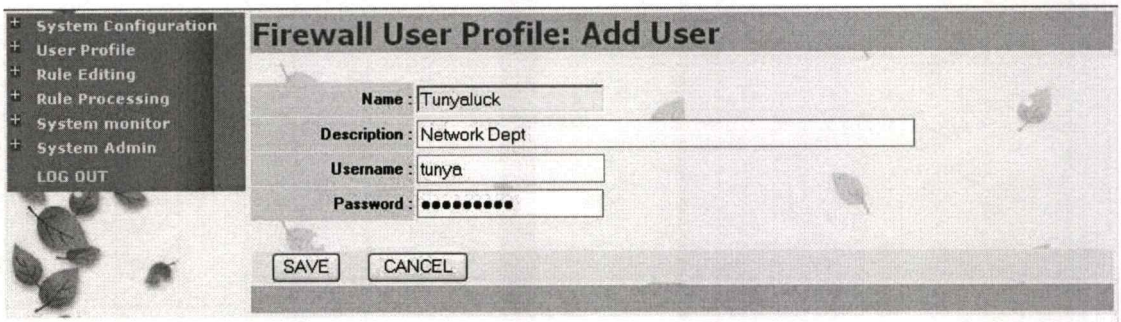
รูปที่ 6 แสดงหน้าจอรายละเอียดของโปรโตคอลต่างๆ

การเพิ่มชื่อผู้ใช้งานระบบ (User Profile)

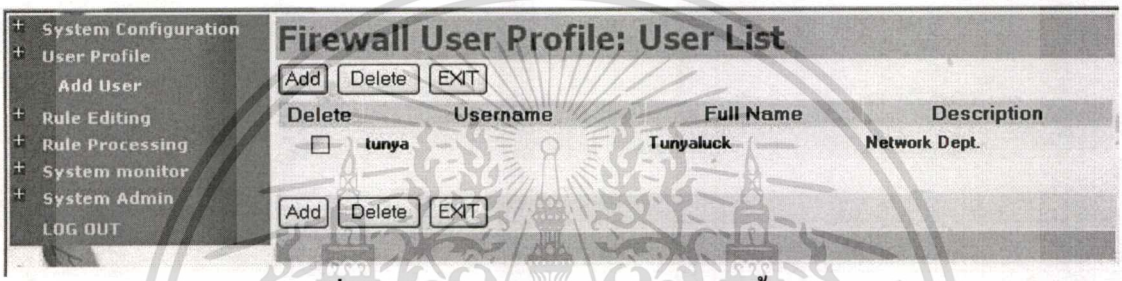
โดยสามารถกำหนดชื่อผู้ใช้งานได้จากเมนู User Profile

<u>วิธีการใช้งาน</u>

1. เลือกเมนูย่อย Add User และทำการป้อนข้อมูลต่อไปนี้
   - ชื่อผู้ใช้งาน (Name)
   - รายละเอียดเกี่ยวกับผู้ใช้งาน (Description)
   - ชื่อที่ใช้ในการล๊อกอินเข้าสู่โปรแกรม (Username)
   - รหัสที่ใช้ในการล๊อกอินเข้าสู่โปรแกรม (Password)
2. จากนั้นทำการบันทึกข้อมูลผู้ใช้งาน ดังรูปที่ 6-7



รูปที่ 7 แสดงหน้าจอการเพิ่มชื่อผู้ใช้งาน



รูปที่ 8 แสดงหน้าจอรายละเอียดผู้ใช้งานทั้งหมด

การสร้างกฎของไฟร์วอลล์ (Rule Editing)

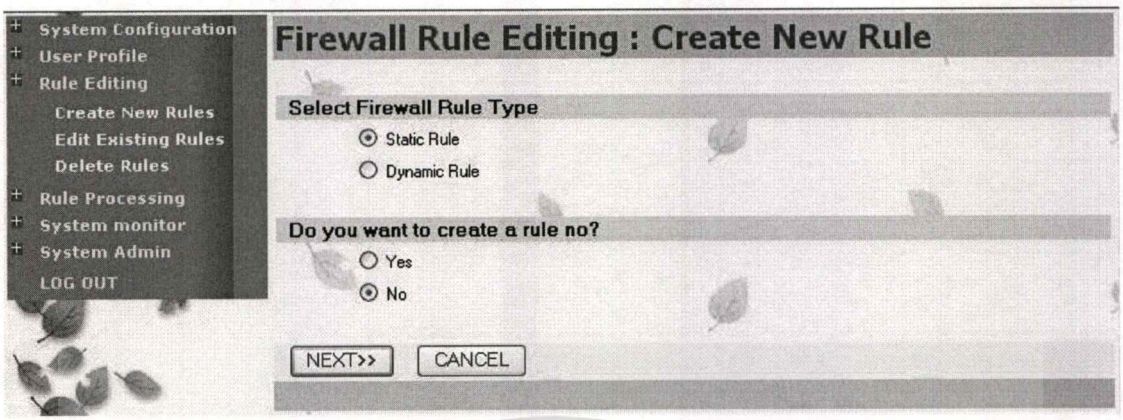
สามารถสร้างได้จากเมนู Rule Editing โดยภายในเมนู Rule Editing สามารถเลือกการทำงานได้ 3 ลักษณะ ดังนี้

1. Create New Rule เป็นเมนูที่ใช้ในการสร้างกฎขึ้นมาใหม่
2. Edit Existing Rules เป็นเมนูที่ใช้ในการเรียกกฎที่ได้ทำการสร้างไว้ขึ้นมาเพื่อทำการแก้ไข
3. Delete Rules เป็นเมนูที่ใช้ในการลบกฎที่ได้ทำการสร้างไว้

เมนู Create New Rule

<u>วิธีการใช้งาน</u>

1. เลือกเมนู Create New Rule จากนั้นระบบจะให้ทำการเลือกกฎที่จะทำการสร้างว่าเป็นชนิด Static หรือ Dynamic และจะให้เลือกว่าจะกำหนดรหัสเลขที่กฎเองหรือให้ระบบเป็นผู้กำหนด หลังจากเลือกเรียบร้อยแล้วคลิก Next>>

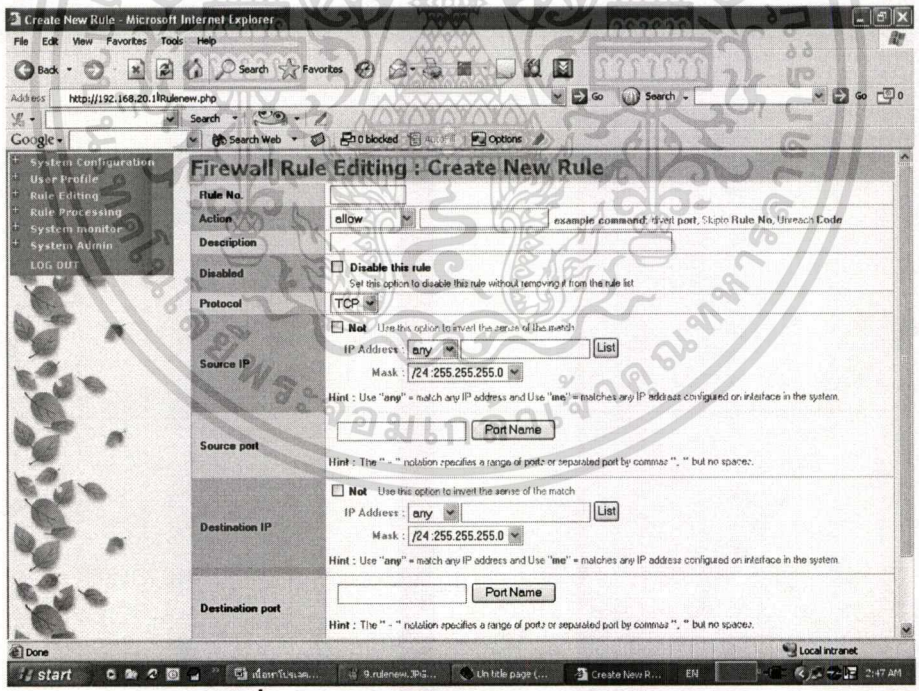


รูปที่ 9 แสดงหน้าจอการเลือกชนิดของกฎที่จะทำการสร้าง

2. ทำการป้อนรายละเอียดของข้อมูลดังนี้
   - Rule No จะเป็นค่าของรหัสเลขที่กฎที่ได้ทำการสร้าง
   - Action จะเป็นคำสั่งที่กำหนดให้กฎที่สร้างทำงานดังนี้
     a. Allow อนุญาตให้ packets ที่ตรงกับกฎ ผ่านออกจากการตรวจสอบของ firewall ได้ และการตรวจสอบจะสิ้นสุดลง ที่กฎตัวนี้
     b. Deny ปฏิเสธ packet ที่ตรงกับกฎ และการตรวจสอบยุติทันที
     c. Check-state เป็นการสั่งให้ตรวจสอบ packet กับ ตารางกฎแบบ dynamic คือถ้าตรวจสอบว่าตรงกัน ให้ดำเนินการ ตามคำสั่งที่เกี่ยวข้องกับกฎ ซึ่งทำงานร่วมกับกฎ แบบ dynamic ข้อนั้น แต่ถ้าไม่ตรงกับกฎ ให้ผ่านไปยังกฎข้อต่อไป กฎแบบ check-state จะไม่มีขอบเขตสำหรับการตรวจสอบ และถ้าไม่มีการใช้กฎแบบ check-state ใน rule set ตารางกฎแบบ dynamic จะถูกตรวจสอบตั้งแต่กฎที่ระบุเป็น keep-state ตัวที่หนึ่ง
     d. Count จะทำการปรับปรุงการนับสำหรับทุก packet ที่ตรงกับกฎ การค้นหาจะทำอย่างต่อเนื่องหลังคำสั่ง count
     e. Divert ***Port*** จะทำการ divert packet ที่ตรงกับกฎไป divert socket bound ยัง Port ที่ระบุ และการค้นหายุติ
     f. Reset ทิ้ง packet ที่ตรงกับกฎนี้และถ้า packet เป็นชนิด TCP ให้ลองส่งค่า TCP reset (RST) กลับ และการค้นหายุติ
     g. Skipto ***Number*** จะทำการข้ามลำดับของกฎที่น้อยกว่าจำนวนที่กำหนดและการค้นหาจะทำอย่างต่อเนื่องหลังคำสั่ง skipto

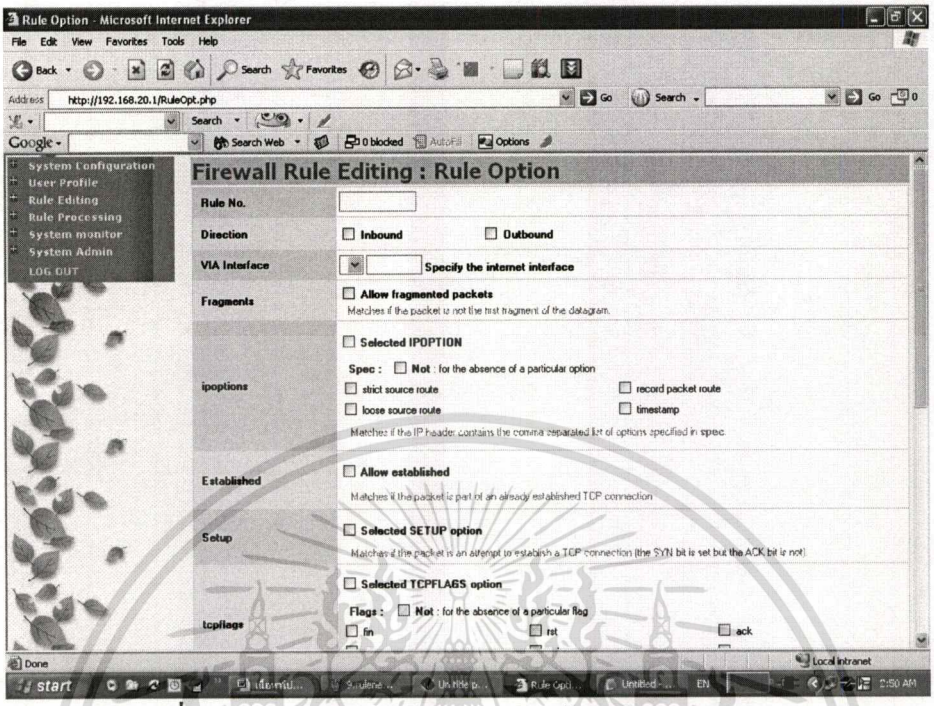
h. Unreach ***Code*** จะเป็นการทิ้ง packet ที่ตรงกับกฎและลองส่ง ICMP unreachable ด้วย Code โดย code เป็นค่าจาก 0 ถึง 255 หรือหนึ่งในค่าต่อไปนี้ ได้แก่ net, host, protocol และ port จากนั้นการตรวจสอบยุติทันที

- Description คำอธิบายรายละเอียดเกี่ยวกับกฎนี้
- Disabled เป็นการกำหนดให้กฎที่ได้ทำการสร้างไม่ทำงานโดยไม่ต้องลบออกจากรายการกฎ
- Protocol จะเป็นค่าของโปรโตคอลที่จะทำการตรวจสอบ
- Source IP แสดงค่า IP Address ต้นทาง
- Source Port แสดงค่าพอร์ตต้นทาง
- Destination IP แสดงค่า IP Address ปลายทาง
- Destination Port แสดงค่าพอร์ตปลายทาง
- Log แสดงค่าการอนุญาตให้กฎนี้เก็บ Log



รูปที่ 10 แสดงหน้าจอการสร้างกฎไฟร์วอลล์

3. หลังจากป้อนข้อมูลที่ต้องการแล้วให้กดปุ่ม Next>> เพื่อทำการป้อนข้อมูลในส่วนของ Rule Option โดยรายละเอียดของ Rule Option มีดังนี้



รูปที่ 11 แสดงหน้าจอการสร้างกฎไฟร์วอลล์ในส่วน Option

- Direction แสดงทิศทางของ packet ที่ทำการตรวจสอบมีค่าเป็น I = in , O = out
- Via Interface ระบุค่า internet interface ที่จะตรวจสอบ
- Fragments อนุญาตให้กฎจับคู่ได้ถ้า packet ไม่ใช่ Fragment แรกของ diagram
- Ipoptions ถ้าเลือก option นี้ให้คลิก Selected IPOPTION และระบุค่า Spec เช่น ssrr – strict source route, lsrr – loose source route, rr – record packet route, ts – timestamp โดยทางเลือกนี้จะอนุญาตให้กฎจับคู่ได้ถ้า IP header ตรงกับค่า Spec
- Established อนุญาตให้กฎจับคู่ได้ถ้า packet เป็นส่วนหนึ่งของการติดต่อแบบ established TCP
- Setup อนุญาตให้กฎจับคู่ TCP packet ที่มี SYN bit แต่ไม่มี ACK bit
- Tcpflags อนุญาตให้กฎจับคู่ได้ถ้า TCP header ตรงกับค่า flag เช่น fin, syn, rst, psh, ack และ urg เป็นต้น
- Icmptypes อนุญาตให้กฎจับคู่กับ ICMP packet ซึ่งมีค่า icmp type ตามที่ระบุ
- Keep-state เป็นคำบังคับ โดยขึ้นอยู่กับการตรวจสอบความถูกต้อง โดย firewall จะสร้างกฎ dynamic ขึ้นมา ซึ่งจะตรวจสอบกฎทั้ง 2 ทาง ระหว่าง ต้นทางและปลายทาง IP/port ที่ใช้ protocol เดียวกัน

- Limit *N* ไฟร์วอลล์จะอนุญาตให้การเชื่อมต่อจำนวนหนึ่ง (N) เท่านั้น ที่จะใช้กฎเดียวกัน จำนวนของต้นทางและปลายทางต้องระบุ ให้ครบ

4. หลังจากป้อนข้อมูลในส่วนของ Option ตามที่ต้องการเรียบร้อยแล้วให้คลิก Save Rule เพื่อทำการบันทึกกฎลงบนฐานข้อมูลกฎ

หมายเหตุ 1. คำว่า "limit" และ "keep-state" ไม่สามารถใช้ร่วมกันในกฎเดียวกัน โดย limit จะมี keep-state เตรียมไว้อยู่แล้ว

2. กฎทั้งหมดที่ทำการสร้างจะยังไม่สามารถใช้งานได้จนกว่าจะได้รับการ Activate จึงจะสามารถใช้งานได้ (ให้ดูในหัวข้อ Rule Processing)

เมนู Edit Existing Rules

เป็นเมนูที่ใช้ในการเรียกกฎที่ได้ทำการสร้างไว้ขึ้นมาเพื่อทำการแก้ไขโดยมีรายละเอียดส่วนต่างๆเหมือนกับเมนู Create New Rule

<u>วิธีการใช้งาน</u>

1. เลือกเมนู Edit Existing Rule จะปรากฏหน้าจอแสดงกฎทั้งหมดที่จะสามารถแก้ไขได้
2. เลือกรหัสเลขที่กฎ และคลิก Edit จะปรากฏหน้าจอให้เลือกว่าจะแก้ไขกฎแบบ Static หรือ Dynamic และจะกำหนดเลขที่กฎเองหรือไม่
3. เมื่อทำการแก้ไขกฎทั้งหมดเรียบร้อยแล้วให้คลิกปุ่ม Save Rule เพื่อบันทึกลงบนฐานข้อมูลกฎ

หมายเหตุ กฎทั้งหมดที่ทำการแก้ไขจะยังไม่สามารถใช้งานได้จนกว่าจะได้รับการ Activate (ให้ดูในหัวข้อ Rule Processing)

เมนู Delete Rules

<u>วิธีการใช้งาน</u>

1. เลือกเมนู Delete Rules ระบบจะแสดงกฎทั้งหมดที่มีอยู่ ให้คลิกเลือกกฎที่ต้องการลบ
2. จากนั้นกดปุ่ม Delete เพื่อทำการลบกฎระบบจะมีการสอบถามว่าจะลบหรือไม่ ถ้าต้องการลบให้ตอบ Yes แต่ถ้าไม่ให้เลือก No
3. หากตอบ Yes หลังจากระบบลบกฎทั้งหมดแล้ว ระบบจะแจ้งผลการลบเสร็จเรียบร้อย

หมายเหตุ กฎทั้งหมดที่ทำการลบจะทำงานถูกต้องสมบูรณ์เมื่อได้รับการ Activate การทำงานเป็นที่เรียบร้อยแล้ว (ให้ดูในหัวข้อ Rule Processing)

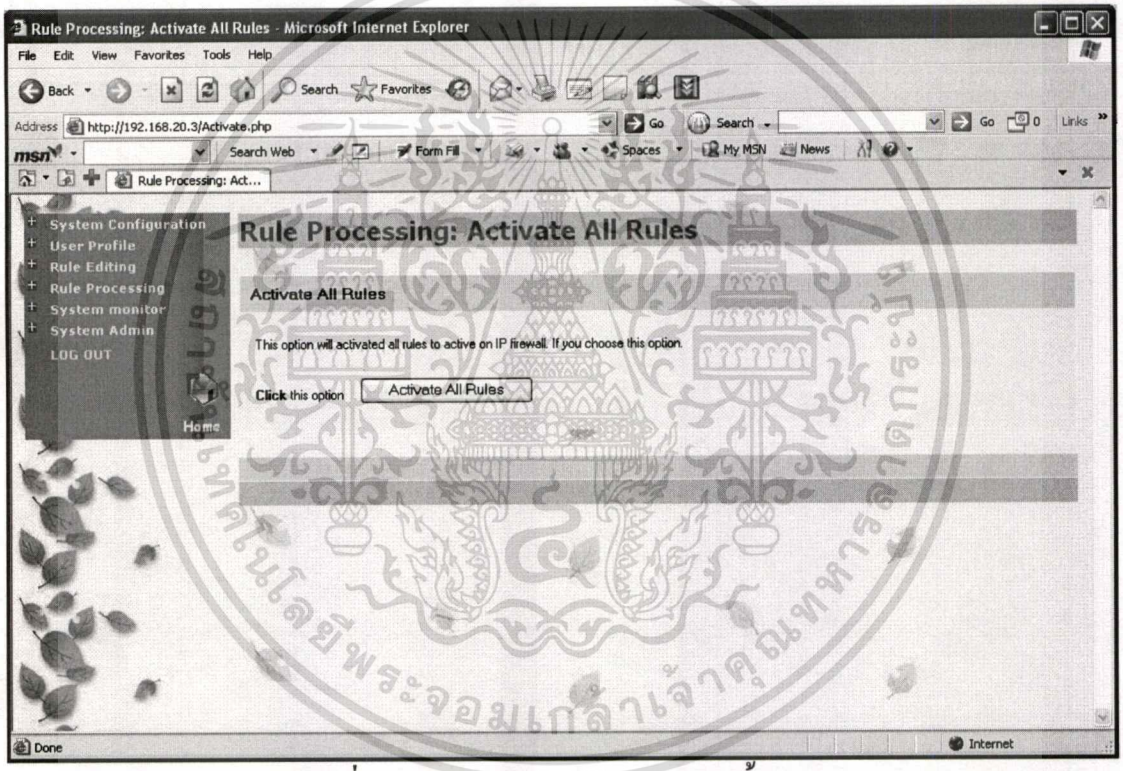
การกำหนดให้กฎทั้งหมดทำงาน (Rule Processing)

กฎทั้งหมดที่ได้ทำการสร้างหรือแก้ไขหรือทำการลบ จะสามารถทำงานได้ถูกต้องสมบูรณ์ต่อเมื่อได้รับการ Activate การทำงานเป็นที่เรียบร้อยแล้วเท่านั้น โดยในการActivate การทำงานนั้นสามารถจะกระทำได้พร้อมกันทีเดียวหลังจากที่ได้มีการสร้างกฎใหม่ หรือแก้ไขกฎเดิม หรือได้มีการลบกฎที่สร้างไว้ออกไปแล้วได้

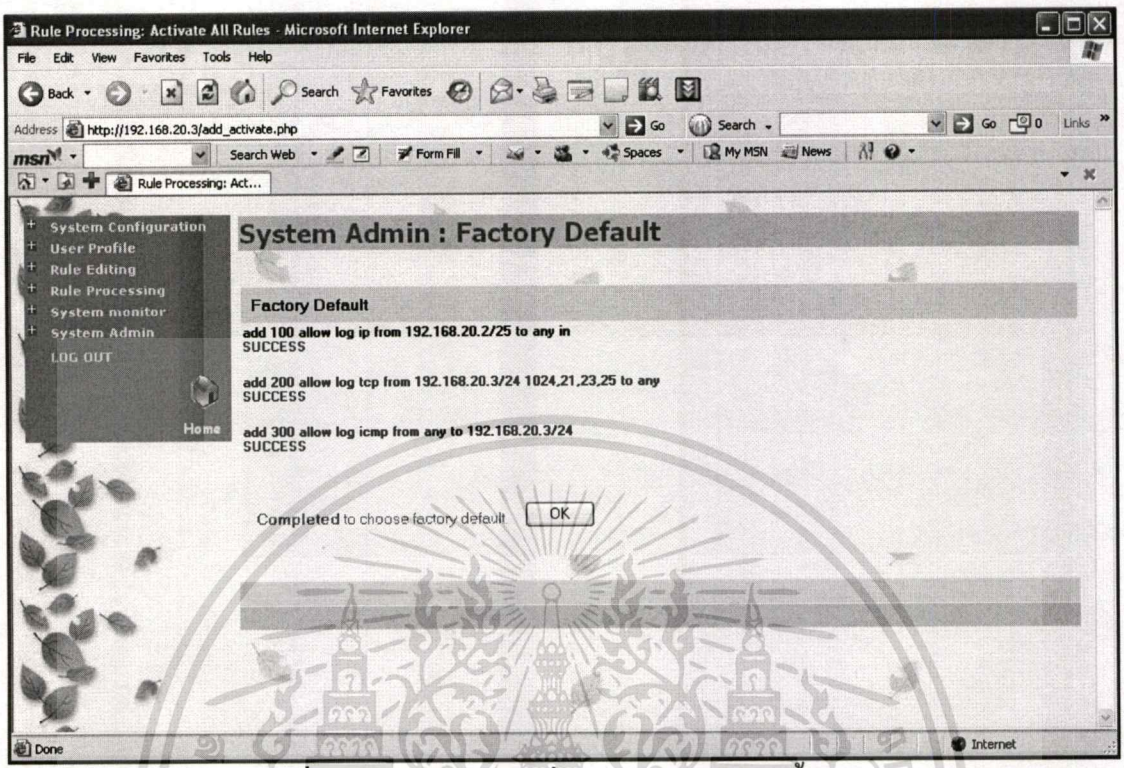
<u>วิธีการใช้งาน</u>

1 เลือกเมนู Rule Processiong > Activate All Rules

2 ระบบจะแสดงหน้าจอการ Activate All Rules ขึ้นให้ตอบ OK เพื่อทำการ Activate Rules



รูปที่ 12 แสดงหน้าจอการActivate กฎทั้งหมด



รูปที่ 13 แสดงหน้าจอกฎที่ได้ถูก Activate แล้วทั้งหมด

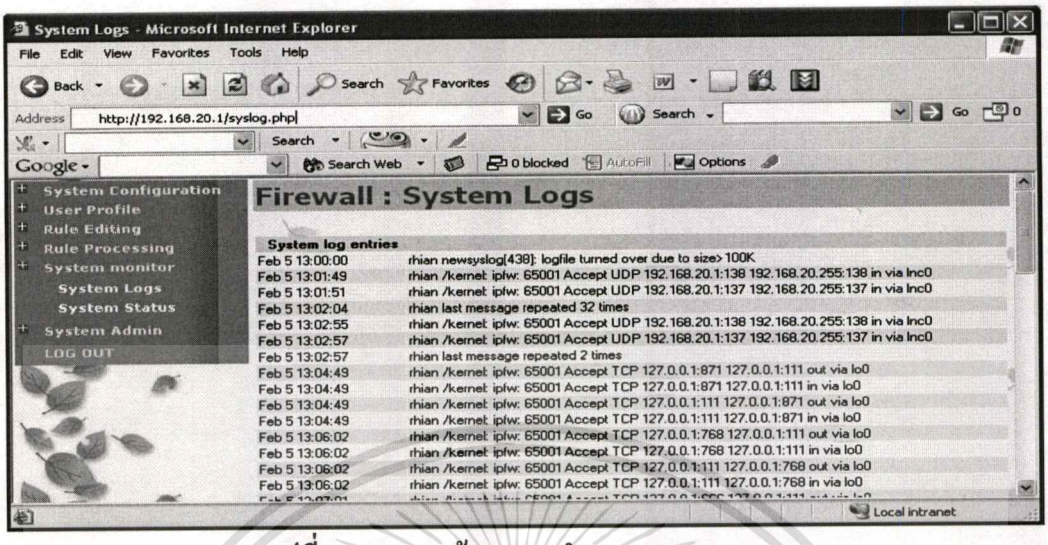
การแสดงการทำงานของกฎที่สร้างไว้ทั้งหมด (System Monitor)

เป็นเมนูที่ให้ในการแสดงสถานะการทำงานของกฎต่างๆที่ได้ทำการสร้างขึ้นโดยแบ่งการทำงานออกเป็น 2 เมนู คือ

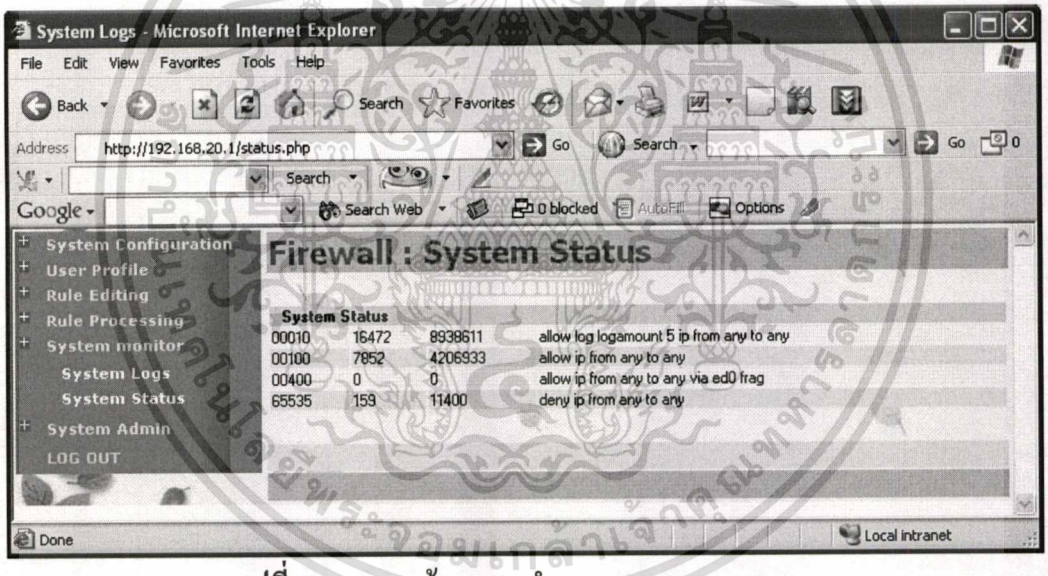
1. เมนู System Logs
2. เมนู System Status

<u>วิธีการใช้งาน</u>

1. เลือกเมนู System Monitor > System Status or System Logs
2. ระบบจะแสดงหน้าจอของระบบขึ้น ดังรูป



รูปที่ 14 แสดงหน้าจอการทำงานของ System Logs



รูปที่ 15 แสดงหน้าจอการทำงานของ System Status

คำสั่งพิเศษ (System Admin)

เป็นคำสั่งที่ใช้ในการทำงานกับระบบโปรแกรมการกำหนดกฎโดยแบ่งการทำงานออกเป็น 5 เมนู คือ

1. เมนู Factory default

<u>วิธีการใช้งาน</u>

1. เลือกเมนู System Admin > Factory default
2. ระบบจะแสดงหน้าจอ Factory default ขึ้นและถามว่าจะทำ Factory default หรือไม่ ถ้าต้องการให้ตอบ Yes ถ้าไม่ให้ตอบ No

2. เมนู Reboot system

<u>วิธีการใช้งาน</u>

1. เลือกเมนู System Admin > Reboot system
2. ระบบจะแสดงหน้าจอ Reboot system ขึ้นและถามว่าจะทำ Reboot system หรือไม่ ถ้าต้องการให้ตอบ Yes ถ้าไม่ให้ตอบ No

3. เมนู Backup configuration

<u>วิธีการใช้งาน</u>

1. เลือกเมนู System Admin > Backup configuration
2. ระบบจะแสดงหน้าจอ Backup configuration ขึ้น คลิกปุ่ม "Backup configuration" ระบบจะทำการ backup file ไปไว้ที่ /tmp/backup_rule.txt
3. หลักจากเลือกเสร็จแล้วให้ตอบ OK เพื่อทำการ Backup file และ CANCEL ถ้าไม่ต้องการ

4. เมนู Load configuration

<u>วิธีการใช้งาน</u>

1. เลือกเมนู System Admin > Load configuration
2. ระบบจะแสดงหน้าจอ Load configuration ขึ้น จากนั้นเลือกว่าจะทำการ Load script config file จากที่ไดเรกทอรี่ใดและไฟล์ชื่ออะไรโดยชนิดของไฟล์จะต้องเป็นชนิด textfile นามสกุล (*.txt)
3. หลักจากเลือกเสร็จแล้วให้ตอบ OK เพื่อทำการ Load file และ CANCEL ถ้าไม่ต้องการ

หมายเหตุ ไฟล์ที่ได้ทำการโหลดจะยังไม่สามารถใช้งานได้จนกว่าจะได้รับการ Activate การทำงานเรียบร้อยแล้ว

5. เมนู Restore configuration

<u>วิธีการใช้งาน</u>

1. เลือกเมนู System Admin > Restore configuration
2. ระบบจะแสดงหน้าจอ Restore configuration ขึ้นและถามว่าจะทำ Restore configuration หรือไม่ ถ้าต้องการให้ตอบ Yes ถ้าไม่ให้ตอบ No

หมายเหตุ การ Restore configuration เป็นการ Rollback กฎที่ได้มีการใช้งานอยู่จริงในก่อนหน้าที่จะทำการแก้ไข ขึ้นมาทำงานแทนกฎที่ได้รับการแก้ไข

ภาคผนวก จ

ตัวอย่างไฟล์ rule_backup.txt

ซึ่งประกอบด้วยกฎชนิด STATIC 2 กฎดังนี้

100 allow log ip from 192.168.20.2/25 to any in

200 allow log tcp from 192.168.20.3/24 to any 1024,21,23,25

100&#static&#allow&#0&#ip&#192.168.20.2/25&#any&#&#&#0&#0&#1&#I&#&#0&#0&#&
#0&#0&#0&#0&#&#0&#0&#&#0&#0&#&#&#&#$$
200&#static&#allow&#0&#tcp&#192.168.20.3/24&#any&#1024,21,23,25&#&#0&#0&#1&#&
#&#0&#0&#&#0&#0&#0&#&#0&#0&#&#0&#0&#&#&#&#$$

ประวัติผู้เขียน

ชื่อสกุล : นางสาว ธัญลักษณ์ ผังชัยมงคล

วัน เดือน ปี เกิด : 26 พฤศจิกายน 2516

สถานที่เกิด : จังหวัดกรุงเทพมหานคร

ประวัติการศึกษา : สำเร็จการศึกษาวิทยาศาสตรบัณฑิต จากมหาวิทยาลัยมหิดล ในปี พ.ศ. 2538 และเข้ารับการศึกษาในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ประวัติการทำงาน : ปัจจุบันทำงานในบริษัทเอกชน ตำแหน่งผู้จัดการฝ่ายพัฒนาธุรกิจ บริษัท BAY COMPUTING CO., LTD.

2003 – 2005	Presale Network Engineer and Marketing	IT DISTRIBUTION CO., LTD.
2002 – 2003	Presale Network Engineer and Marketing	P.T. INFO TECHNOLOGY CO., LTD.
1999 – 2002	EDI System Engineer	NYK TRANSPORT SERVICE (THAILAND)CO., LTD.
1996 – 1999	System Engineer	CDG GROUP (SOFTWARE CITY CO., LTD.)
1996 – 1996	Programmer	SOON HUA SENG GROUP.
1994 – 1996	Analytical Marketing	MARKET FEEDBACK CO., LTD.