การพัฒนาระบบป้องกันการบุกรุกเครือข่ายโดยใช้โปรแกรม SNORT

DEVELOPMENT OF INTRUSION PREVENTION SYSTEM USING SNORT PROGRAM

โดย

ดุษกร อรรจน์อังกูร

DUSAKORN AT-ANGKUL

อาจารย์ที่ปรึกษา

ผศ.อัครินทร์ คุณกิตติ

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ภาคเรียนที่ 2 ปีการศึกษา 2551

DEVELOPMENT OF INTRUSION PREVENTION SYSTEM USING SNORT PROGRAM

DUSAKORN AT-ANGKUL

A SYSTEM DEVELOPMENT PROJECT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY

FACULTY OF INFORMATION TECNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2/ 2008

หัวข้อ	การพัฒนาระบบป้องกันการบุกรุกเครือข่ายโดยใช้โปรแกรม Snort
นักศึกษา	นายดุษกร อรรจน์อังกูร
รหัสประจำตัว	49066813
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2551
อาจารย์ที่ปรึกษา	ผศ.อัครินทร์ คุณกิตติ

บทคัดย่อ

ปัจจุบันการรักษาความปลอดภัยของสารสนเทศภายในองค์กรนั้นมีความสำคัญมาก ซึ่งสามารถกระทำได้ตั้งแต่ การติดตั้งไฟร์วอล หรือการติดตั้งระบบตรวจจับการบุกรุกเครือข่าย ซึ่งการใช้งานเพียงแค่ระบบตรวจจับการบุกรุกนั้นก็ยังไม่มีความปลอดภัยเพียงพอ จึงทำให้เกิดแนวคิดในการสร้างระบบป้องกันการบุกรุกเครือข่ายขึ้น โดยในโครงการพัฒนาระบบงานนี้ จะเป็นการพัฒนาโปรแกรมบนระบบปฏิบัติการ FreeBSD และทำงานอยู่บนเครื่องเดียวกับระบบตรวจจับการบุกรุกเครือข่าย (SNORT) โดย SNORT จะทำการบันทึกข้อมูลการบุกรุกที่ตรวจสอบพบลงในฐานข้อมูล (MYSQL) จากนั้นโปรแกรมที่พัฒนาขึ้นจะไปอ่านข้อมูลจากฐานข้อมูล เพื่อนำเอาข้อมูลการบุกรุกนั้นมาสร้างเป็นคำสั่งเพิ่มกฎในการป้องกันไปยังไฟร์วอล (IPFW) จากนั้นต้องสามารถยกเลิกกฎการป้องกันได้เมื่อครบกำหนดเวลาตามที่กำหนด และระบบที่พัฒนาสามารถออกรายงานแสดงข้อมูลเกี่ยวกับกฎที่กำลังใช้งานอยู่หรือกฎที่ได้ถูกใช้งานไปแล้วได้และยังมีช่องทางในการปรับแต่งค่าของตัว SNORT ได้ง่าย ผ่านทางหน้า Webpage ได้

ผลที่ได้จากการพัฒนาระบบคือ โปรแกรมที่พัฒนาสามารถอ่านข้อมูลการบุกรุกจากฐานข้อมูลได้อย่างถูกต้อง และสามารถสั่งการไฟร์วอลให้ทำการป้องกันได้ เป็นการพัฒนาโปรแกรมเพื่อการประยุกต์ใช้งานโปรแกรม SNORT และ ไฟร์วอล (IPFW) ให้ทำงานร่วมกันได้ โดยระบบที่พัฒนาเป็นช่องทางทำให้ประสิทธิภาพในการรักษาความปลอดภัยบนระบบเครือข่ายเพิ่มขึ้นอีกทางหนึ่ง

Title	Development of Intrusion Prevention System Using Snort Program
Student	Mr. Dusakorn At-angkul
Student ID	49066813
Degree	Master of Science
Programme	Information Technology
Academic Year	2008
Advisor	Asst.Prof. Akharin Khunkitti

ABSTRACT

At present internal information security is very important. Firewall or network intrusion detection system is not sufficient for safety system nowadays. Thus, network intrusion prevention system is developed. It is the development of FreeBSD operation system. Working on the same unit of SNORT which will record intrusion information onto MYSQL, the developed program will read information on database in order to create the order in increasing preventive rule to IPFW accordingly. Besides, the program developed should have ability in canceling intrusion prevention system on time set. In addition, it can report rule which is exercising or the rule which is performed. The adjustment of SNORT value can be done easily through webpage.

The results benefiting from system development are that the developed program can read intrusion information from database correctly and can command firewall to working properly. It is developed with the purpose of the working application of SNORT and IPFW program together. This safety program is another channel in enhancing the efficiency of safety system on network.

กิตติกรรมประกาศ

โครงการนี้สำเร็จลุล่วงได้ด้วยดี ด้วยคำแนะนำและคำปรึกษา ตลอดจนการตรวจสอบแก้ไข เพื่อให้โครงการนี้เสร็จสมบูรณ์ จาก ผศ. อัครินทร์ คุณกิตติ ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงาน

ขอขอบคุณ คณาจารย์คณะเทคโนโลยีสารสนเทศทุกๆท่าน ที่ให้ความรู้

ขอขอบคุณเพื่อนๆพระจอมเกล้าพระนครเหนือและพระจอมเกล้าลาดกระบังที่สนับสนุน และคอยแบ่งปันความรู้ให้กันและกันมาโดยตลอด

สุดท้ายนี้ขอขอบพระคุณ บิดา มารดา อันเป็นที่รักที่เลี้ยงดูและให้การศึกษาและขอขอบคุณท่านอื่นๆที่คอยสนับสนุนทำให้โครงการนี้สำเร็จซึ่งมิได้กล่าวนะที่นี้

ดุษกร อรรจน์อังกูร

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

สารบัญรูป

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากปัจจุบันเครือข่ายคอมพิวเตอร์มีการใช้งานอย่างแพร่หลายในทุกๆที่ และเครือข่ายในแต่ละที่จะมีการเชื่อมต่อถึงกันเพื่อแลกเปลี่ยนข้อมูลข่าวสารเพื่อความสะดวกในการ ติดต่อสื่อสาร ซึ่งเมื่อเครือข่ายมีการเชื่อมต่อเข้าหากันมากขึ้น ทำให้เกิดความไม่ปลอดภัยของเครือข่ายเพิ่มมากขึ้นจากการเข้าถึงเครือข่ายจากผู้ไม่ประสงค์ดี ทำให้ในปัจจุบันทุกหน่วยงานให้ความสำคัญในการลงทุนในเรื่องระบบรักษาความปลอดภัยในเครือข่าย เช่น มีการนำไฟร์วอลล์เข้ามาใช้งานภายในองค์กรเพื่อช่วยในการป้องกันระบบเครือข่ายภายในองค์กร ซึ่งไฟร์วอลล์นั้นมีหน้าที่ในการควบคุมการเข้าออกของข้อมูลที่ผ่านตัวไฟร์วอลล์ โดยที่ไฟร์วอลล์จะมีลักษณะการทำงานโดยการอ้างอิงจากกฎ ที่มีการสร้างไว้เพื่อบอกให้ไฟร์วอลล์ทำตามกฎที่ตั้งไว้ ว่าจะให้ข้อมูลหรือการกระทำใด ผ่านเข้าหรือออกไฟร์วอลล์ได้ ซึ่งก็จะทำให้ระบบเครือข่ายมีความปลอดภัยในระดับหนึ่ง แต่จะเห็นว่าการมีไฟร์วอลล์ใช้งานเพียงอย่างเดียวนั้น ยังไม่สามารถช่วยให้ระบบเครือข่ายมีความปลอดภัยได้ เนื่องจากจะเห็นว่าไฟร์วอลล์นั้นจะทำงานตามกฎที่ถูกกำหนดไว้แล้ว ดังนั้นถ้าผู้ดูแลระบบกำหนดกฎไม่ครบถ้วนก็อาจจะเกิดความไม่ปลอดภัยแก่ระบบเครือข่ายได้

ดังนั้นในเวลาต่อมาจึงเกิดการคิดค้น ระบบตรวจจับการบุกรุกเครือข่ายขึ้น ซึ่งระบบตรวจจับการบุกรุกเครือข่ายที่ได้รับความนิยมอย่างมากและเป็นซอฟต์แวร์ Open source คือ โปรแกรม SNORT เนื่องจากเป็นซอฟต์แวร์ที่สามารถดาวน์โหลดได้จากอินเตอร์เน็ต และตัว SNORT ก็มีกฎมากมายในการตรวจจับการบุกรุกจึงเป็นอีกสาเหตุหนึ่งที่ทำให้เป็นที่นิยม แต่จะเห็นว่า SNORT นั้นเป็นเพียงระบบตรวจจับการบุกรุกเครือข่ายเท่านั้น กล่าวคือ เมื่อมีการบุกรุกเครือข่ายขึ้น เมื่อ SNORT ทำการตรวจพบก็จะทำการแจ้งเตือนเท่านั้น แต่ไม่สามารถป้องกันการบุกรุกนั้นได้ จึงต้องเป็นหน้าที่ของผู้ดูแลระบบ ในการที่จะต้องแก้ไข กฎของตัวไฟร์วอลล์ให้ทำการป้องกันการบุกรุกตามที่ตัว SNORT ตรวจจับได้ซึ่งอาจจะช้าเกินไปในการป้องกัน เพราะผู้บุกรุกอาจจะเข้าถึงระบบและสร้างความเสียหายแก่ระบบแล้ว

จึงทำให้เกิดแนวความคิดในการพัฒนาระบบให้สามารถเชื่อมการทำงานระหว่าง ตัว ไฟร์วอลล์และ ตัว SNORT ให้สามารถทำการป้องกันการบุกรุกได้ทันทีโดยไม่ต้องอาศัยผู้ดูแลระบบคอยสั่งไฟร์วอลล์ให้ทำการป้องกันการบุกรุกนั้น ซึ่งการทำงานของระบบคือ ระบบจะตรวจสอบการบุกรุกผ่านฐานข้อมูลการบุรุกที่ SNORT ใช้บันทึกข้อมูลการบุกรุก เมื่อตรวจพบว่ามีการบุกรุก

โปรแกรมก็จะทำการไปสั่งไฟร์วอลล์ให้ป้องกันการบุกรุกทันที และไฟร์วอลล์ที่ใช้ คือ IPFW ซึ่งเป็นไฟร์วอลล์ที่มีอยู่แล้วในระบบปฏิบัติการ FreeBSD เป็นต้น

1.2 ขอบเขตของโครงการพัฒนาระบบงาน

ระบบป้องกันการบุกรุกเครือข่ายโดยใช้โปรแกรม SNORT ที่จะพัฒนาขึ้น มีขอบเขตดังนี้

1.2.1 ใช้ภาษา PHP ในการเขียนโปรแกรมเพื่อเป็นตัวกลางในการติดต่อสื่อสารกัน ระหว่าง SNORT กับ IPFW ซึ่งภาษา PHP นั้นสามารถใช้ได้บนระบบปฏิบัติการยูนิกส์ เช่น FreeBSD, Red hat

1.2.2 ระบบจะทำการวิเคราะห์ข้อมูลการบุกรุกที่โปรแกรม SNORT ตรวจจับได้จากฐานข้อมูล ซึ่งข้อมูลการบุกรุกที่ต้องการจากฐานข้อมูล ได้แก่ Protocol, IP Address, Port ของต้นทางและปลายทาง

1.2.3 ซึ่งระบบต้องสามารถทำการป้องกันได้ทั้ง 2 กรณีคือ

1) ถ้าข้อมูลในฐานข้อมูลที่ตรวจจับได้มีแต่ค่า IP Address ต้นทาง, ปลายทาง ระบบต้อง สามารถป้องกันการบุกรุกที่เกิดจาก IP Address ต้นทาง ที่ไปยังปลายทางได้

2) ถ้าข้อมูลในฐานข้อมูลที่ตรวจจับได้มีค่า IP Address ต้นทาง, ปลายทาง Port ต้นทาง, ปลายทาง ระบบต้องสามารถป้องกันการบุกรุกที่เกิดจาก IP Address ต้นทาง Port ต้นทางไปยัง IP Address ปลายทาง Port ปลายทางได้

1.2.4 เมื่อตรวจพบการบุกรุกระบบต้องสามารถ Run Script เพื่อสั่งไฟร์วอลล์ให้ทำการป้องกันการบุกรุกนั้นได้

1.2.5 เมื่อระบบได้ทำการป้องกันการบุกรุกนั้นแล้ว จะมีการกำหนดช่วงเวลาที่จะทำการยกเลิก การป้องกันการบุกรุกนั้น ระบบต้องสามารถทำการยกเลิกการป้องกันได้ทันที เมื่อถึงกำหนดเวลา

1.2.6 ระบบต้องสามารถให้ผู้ดูแลระบบ สามารถระบุค่าที่จำเป็นในการใช้งานลงในไฟล์ Snort.conf ผ่านทางหน้าเว็บเพจได้ เพื่อให้สะดวกในการใช้งานแก่ผู้ดูแลระบบ

1.2.7 ระบบต้องสามารถแสดงรายงานผ่านทางเว็บเพจเพื่อให้ผู้ดูแลระบบสามารถตรวจสอบได้ว่าขณะนี้มีกฎใดบ้างที่ยังทำงานอยู่และกฎใดบ้างที่ยกเลิกไปแล้ว

1.3 ขั้นตอนการดำเนินงาน

ในการพัฒนาโครงการนี้ได้แบ่งขั้นตอนในการศึกษาและพัฒนาโปรแกรม ดังนี้

ระยะที่ 1 ศึกษาการใช้งานคำสั่งที่ใช้ควบคุมและกฎต่างๆของไฟร์วอลล์ IPFW และศึกษาการทำงานของ SNORT การแจ้งเตือนในรูปแบบต่างๆ

ระยะที่ 2 ศึกษาการเขียนโปรแกรมด้วยภาษา PHP และเครื่องมือที่ใช้ในการพัฒนาโปรแกรม

ระยะที่ 3 วิเคราะห์และออกแบบระบบตามขอบเขตการพัฒนา

ระยะที่4 พัฒนาตัวโปรแกรม และทดสอบระบบ

ระยะที่ 5 สรุปผลการพัฒนา

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้พัฒนาความรู้ความเข้าใจหลักการทำงานของ Firewall (IPFW) และระบบป้องกันการบุกรุกเครือข่าย (SNORT) และสามารถนำมาประยุกต์ใช้งานในการรักษาความปลอดภัย ในครือข่ายได้
2. ได้พัฒนาความรู้ในการเขียนโปรแกรมด้วย ภาษา PHP บนระบบปฏิบัติการ FreeBSD
3. ได้พัฒนาความรู้ในการวิเคราะห์และออกแบบระบบและสามารถนำไปใช้ประโยชน์ในการทำงานได้
4. ส่งเสริมให้เกิดการใช้งานซอฟต์แวร์ Open source ให้แพร่หลายมากขึ้น
5. ระบบที่พัฒนาขึ้นสมารถช่วยให้เกิดการป้องกันการบุกรุกเครือข่ายได้

บทที่ 2
การป้องกันการบุกรุกเครือข่ายโดยใช้โปรแกรม SNORT

2.1 ภัยคุกคามระบบคอมพิวเตอร์ (Threat)

ภัยคุกคามระบบคอมพิวเตอร์ในที่นี้หมายถึงการกระทำอันก่อให้เกิดความเสียหายแก่ระบบคอมพิวเตอร์ ซึ่งไม่ว่าจะเป็นการขโมยข้อมูล หารทำลายข้อมูล ไปจนกระทั่งทำให้ระบบไม่สามารถใช้งานได้ ภัยคุกคามอาจจะเกิดขึ้นโดยอุบัติเหตุ เช่น ไฟไหม้ น้ำท่วม หรือเกิดขึ้นโดยเจตนาของบุคคลที่ต้องการก่อให้เกิดความเสียหายแก่ระบบโดยการโจมตี การกระทำที่ก่อให้เกิดภัยคุกคามเหล่านี้เป็นการกระทำที่ทำให้ระบบขาดความปลอดภัย ซึ่งระบบที่มีความปลอดภัยจะหมายถึงระบบที่มีการให้บริการในลักษณะอย่างน้อย 3 ประการคือ

1. การรักษาความลับ (Confidentiality) คือการรับรองว่าจะมีการเก็บข้อมูลไว้เป็นความลับ และผู้มีสิทธิเท่านั้นจึงจะเข้าถึงข้อมูลนั้นได้
2. การรักษาความสมบูรณ์ (Integrity) คือการรับรองว่าข้อมูลจะไม่ถูกเปลี่ยนแปลงหรือถูกทำลายโดยผู้ไม่มีสิทธิ
3. ความพร้อมใช้ (Availability) คือการรับรองว่าข้อมูลและบริการพร้อมที่จะใช้ในเวลาที่ต้องการใช้

ดังนั้นหากมีพฤติกรรมใดที่เป็นผลให้ขาดข้อใดข้อหนึ่งข้างต้นไป เราจะเรียกพฤติกรรมเหล่านั้นว่า “การบุกรุก” (intrusion) และเรียกผู้ที่กระทำพฤติกรรมดังกล่าวว่า “ผู้บุกรุก” (intruder)

2.2 รูปแบบการโจมตีระบบเครือข่าย

การที่เราจะสามารถป้องกันระบบเครือข่ายได้นั้น สิ่งหนึ่งที่จะต้องทราบในเบื้องต้นก่อนก็คือ รูปแบบต่างๆที่ผู้บุกรุกจะสามารถใช้ในการโจมตีระบบของเรา เพื่อที่เราจะสามารถเตรียมการป้องกันได้อย่างมีประสิทธิภาพและทำให้ครอบคลุมการโจมตีทุกรูปแบบ ซึ่งวิธีการโจมตีนั้นมีมากมาย แต่ที่จะยกตัวอย่างนั้นได้แก่

1. DoS (Denial of service) เป็นการโจมตีให้ปิดบริการโดยเป็นการโจมตีไปเครื่องเป้าหมาย โดยใช้เครื่องที่ถูก compromise แล้วจากที่อื่นเป็นฐานในการโจมตีเครื่องเป้าหมายพร้อมกันเพื่อให้ไม่สามารถให้บริการได้ ทำให้เกิดความคับคั่งในเครือข่ายและเครื่องที่ถูกโจมตีไม่สามารถทำงานได้ตามปกติ
2. IP spoofing การปลอม ไอพีแอดเดรสเพื่อให้สามารถเข้าสู่ระบบได้ และทำให้หาแหล่งที่มาของการโจมตีไม่ได้

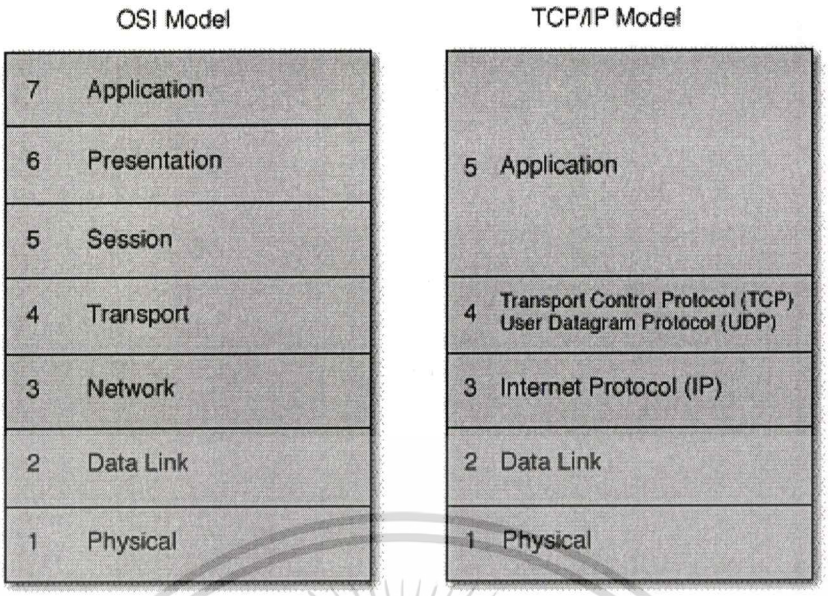
3. Buffer Overflows พยายามส่งข้อมูลให้เกิด Buffer Overflows ซึ่งจะลดความสามารถในการป้องกันได้ โดยที่บางระบบเมื่อเกิด Buffer Overflows อาจจะทำให้ระบบหยุดทำงานและผู้บุกรุกสามารถเข้าไปในเครือข่ายได้
4. Scan port เป็นการทำการค้นหา Port ที่เปิดอยู่บนเครื่องเป้าหมาย เพื่ออาศัย Port นั้นใช้ในการโจมตี
5. Ping sweeps เป็นการพยายามหาเครื่องที่เปิดอยู่โดยการส่ง ICMP Packet ไป ถ้าเครื่องตอบกลับมาแสดงว่าเครื่องนั้น Online อยู่ ผู้บุกรุกก็สามารถทำการโจมตีได้
6. Software Bug คือการที่ผู้บุกรุกพยายามหาช่องโหว่ของซอฟต์แวร์ ซึ่งโดยส่วนใหญ่ Bug จะมีการอุดช่องโหว่ไปก็ต่อเมื่อ พบว่ามีการโจมตีไปแล้วระยะหนึ่ง
7. Ping of death เป็นการพยายามใช้คำสั่ง ping ไปยังเครื่องเป้าหมาย เพื่อให้เครื่องเป้าหมายตอบกลับมาอย่างต่อเนื่อง โดยทำให้เครื่องเป้าหมายช้าลง และทำให้เครือข่ายมีดับดั่ง
8. SYN flood คือการส่ง TCP SYN Packet จำนวนมากและรวดเร็ว ไปที่เครื่องเป้าหมายเพื่อให้เป้าหมายตอบกลับมา โดยเป็นการหวังผลเพื่อลดความสามารถการทำงานของเครื่องในขณะนั้น
9. Password Cracking เป็นการที่ผู้บุกรุกพยายามเข้าระบบโดยการสุ่มรหัสผ่านเพื่อเข้าใช้งานในระบบ โดยมี 2 วิธีการคือ Dictionary Attacks , Brute force attacks
10. Web server attacks โจมตีผ่านทาง web server เช่น พยายาม execute file พวก script บน Web server เพื่อจะเข้าควบคุมเครื่องนั้น
11. Web browser attacks คือการโจมตีที่แฝงมากับ script ต่างๆบนหน้าเว็บ

2.3 ไฟร์วอลล์ (Firewall)

ไฟร์วอลล์ คือ ระบบป้องกันภัยทาง Network ป้องมิให้ผู้ที่ไม่ได้รับอนุญาต เข้ามา หรือ ส่ง packets เข้ามาจารกรรมข้อมูล หรือสอดแนม หรือ ทำลาย ความมั่นคง ในระบบเครือข่ายของคุณ ไม่ว่าคุณจะมีคอมพิวเตอร์เครื่องเดียว หรือ หลายเครื่อง เมื่อใดก็ตามที่ คุณเชื่อมต่อเข้าสู่อินเตอร์เน็ต นั่นหมายความว่าใครก็ตามในโลกที่รู้วิธีก็จะทำการเชื่อมต่อ เพื่อดูข้อมูล, สอดแนม, ทำลายระบบบนเครื่องของคุณ หรือ ระบบของคุณได้ทันที

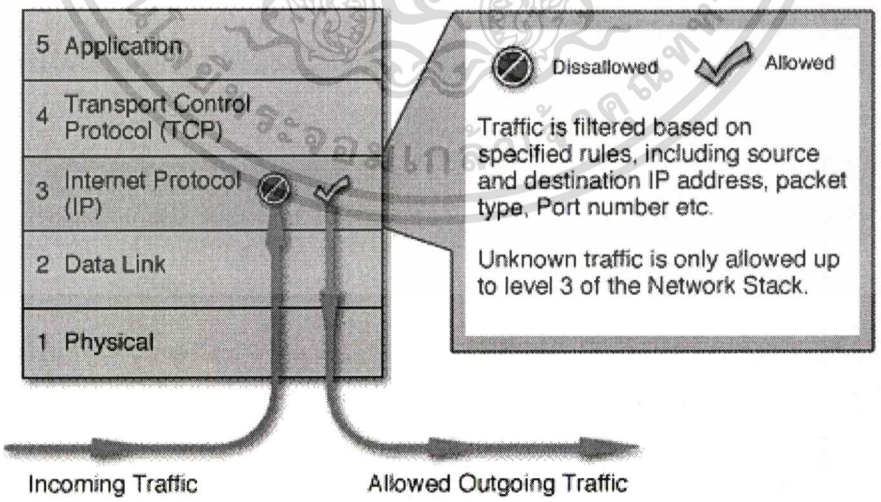
2.3.1 ประเภทของไฟร์วอลล์

ปัจจุบันมีการจำแนกไฟร์วอลล์ ตามความสามารถในการตรวจสอบ Packet ต่างๆ ในระดับของ OSI Model และ TCP/IP Model โดยจำแนกได้ 4 ประเภท คือ



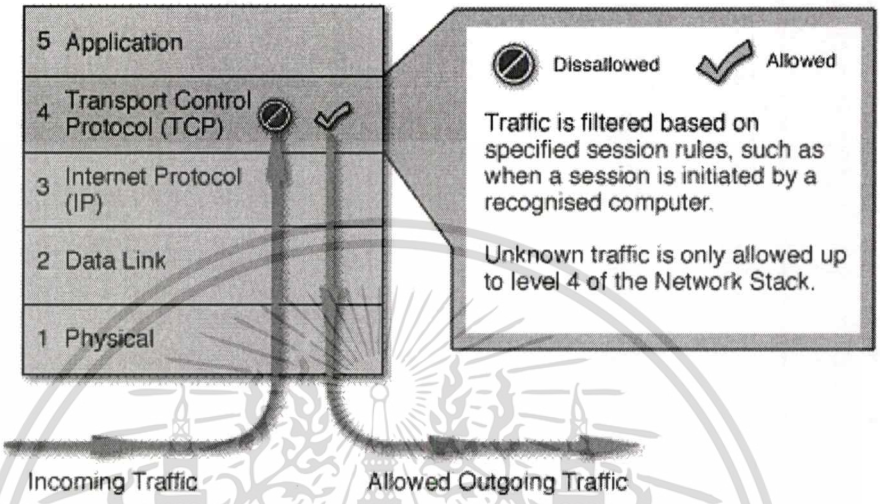
รูปที่ 2.1 OSI Model และ TCP/IP Model

1) **Packet Filter** เป็นไฟร์วอลล์ที่มีการตรวจสอบ packet ในระดับ network ซึ่งตรงกับ layer ที่ 3 ของ OSI model หรือ ระดับ Internet Protocol(IP) ของ TCP/IP Model มีการทำงานคล้ายกับ Router คือ จะรับ Packets มาแล้ว Forward ไปยังเป้าหมายปลายทาง การตรวจสอบ Packet ทำโดยเปรียบเทียบกับกฎ หรือ เงื่อนไขที่ตั้งเอาไว้ว่าตรงกันหรือไม่ ถ้าไม่ตรงกันก็จะไม่ให้ Packet นั้นๆ ผ่านไป โดยจะทำการ Drop หรือ forward หรือ แจ้งข่าวสารกลับไปยังแหล่งที่มาของ Packet นั้นๆ



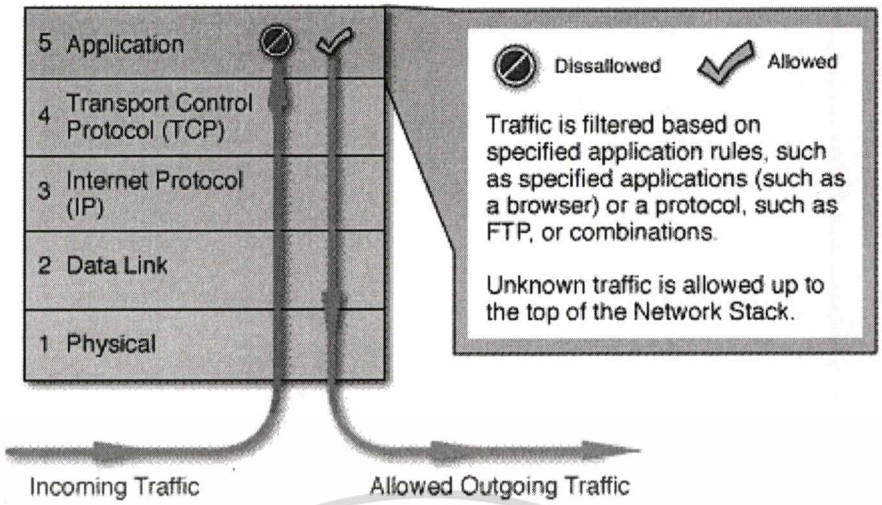
รูปที่ 2.2 รูปแบบการทำงานแบบ Packet Filter

2) **Circuit level Gateway** มีการทำงานในระดับ TCP ของ TCP/IP ตรวจสอบ Packet ทั้งสองด้านว่ามีการร้องขอมาถูกต้องหรือไม่ ข้อมูลข่าวสารจากต้นทางเมื่อผ่านไฟร์วอลล์ไปแล้วจะเปลี่ยนข้อมูลใหม่ โดยจะระบุต้นทางใหม่ว่ามาจากไฟร์วอลล์แทน มีประโยชน์ทำให้สามารถปกปิดแหล่งที่มาหรือต้นทางที่แท้จริง



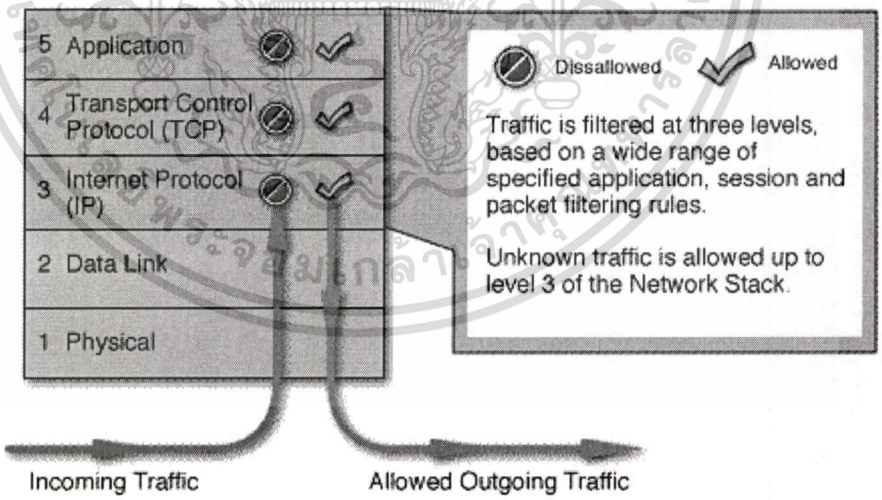
รูปที่ 2.3 รูปแบบการทำงานแบบ Circuit level Gateway

3) **Application level Gateway** ไฟร์วอลล์แบบนี้บางที่ก็เรียกว่า Proxy Firewall เป็นแอพพลิเคชั่นโปรแกรมบนไฟร์วอลล์ที่ตั้งอยู่ระหว่างเครือข่าย 2 เครือข่าย ทำหน้าที่ควบคุมการเชื่อมต่อระหว่างเครือข่ายภายในและภายนอก มีการตรวจสอบข้อมูลถึงในระดับแอพพลิเคชั่นเลเยอร์ เมื่อเครื่องในเครือข่ายภายในต้องการเชื่อมต่อไปยังภายนอก ก็จะร้องขอไปที่ Proxy ก่อนแล้ว Proxy จะติดต่อไปยังเครื่องปลายทางให้ ซึ่งจะมี 2 การเชื่อมต่อ คือ เครื่องภายในกับ Proxy และ Proxy กับเครื่องปลายทาง โดยที่ Proxy จะทำหน้าที่รับข้อมูลและส่งต่อข้อมูล



รูปที่ 2.4 รูปแบบการทำงานแบบ Application Level Gateway

4) **Stateful Multilayer Inspection Firewall** เป็นการรวมการทำงานของไฟร์วอลล์ทั้ง 3 แบบ ที่กล่าวมา มีการทำงานทั้ง 3 layer ตาม TCP/IP Model ไฟร์วอลล์ประเภทนี้มีความน่าเชื่อถือได้มากที่สุด มีความซับซ้อนในการสร้างกฎสำหรับตรวจสอบ Packet จึงตรวจสอบ Packet ได้อย่างละเอียด



รูปที่ 2.5 รูปแบบการทำงานแบบ Stateful Multilayer Inspection Firewall

ซึ่งไฟร์วอลล์ที่เป็นแบบ Stateful Inspection ที่นิยมใช้กันยกตัวอย่างเช่น Cisco Pix Firewall, Checkpoint Firewall-1 เป็นสินค้าที่มีขายอยู่ในท้องตลาด แต่ก็ยังมีไฟร์วอลล์ที่เป็นแบบ

Open source เช่น Netfilter ซึ่งมีอยู่ใน Linux หรือ IPFW ที่อยู่ใน FreeBSD ซึ่งแต่ละแบบก็มีลักษณะของกฎและคำสั่งที่แตกต่างกันไป ผู้ใช้จึงต้องทำการศึกษาให้เข้าใจ

2.4 เตรียมตัวก่อนการใช้งาน IPFW

การใช้งาน IPFW นั้นจะมีส่วนประกอบที่อยู่ภายในเคอร์เนลของระบบปฏิบัติการ FreeBSD ดังนั้นเราจะต้องทำการปรับแต่งเคอร์เนลของระบบ โดยไฟล์เคอร์เนลของ FreeBSD จะอยู่ที่ /usr/src/sys/i386/conf/GENERIC และทำการคัดลอกไฟล์ GENERIC เป็นชื่อไฟล์อื่นแล้วใส่ตัวเลือกเพิ่ม ดังนี้

Options IPFIREWALL การบอกให้เคอร์เนลรองรับการทำงานของIPFW

Options IPFIREWALL_VERBOSE ตัวเลือกที่ทำให้เคอร์เนลสามารถเก็บกิจกรรมที่เกิดขึ้นกับไฟร์วอลล์ลง Log file ในรูปแบบ syslog ได้

Options IPFIREWALL_VERBOSE_LIMIT=X จะเป็นการป้องกัน Log file เต็ม X จะแทนตัวเลขที่จะเก็บจำนวนมากสุดคือ X

ด้วยการเพิ่มตัวเลือก ทั้งสามเข้าไปในเคอร์เนลจะทำให้ระบบสามารถใช้งาน IPFW ได้และเมื่อทำการเพิ่มตัวเลือกเข้าไปแล้วจะต้องทำให้ระบบรับรู้ถึงเคอร์เนลใหม่โดยมีขั้นตอนดังนี้

1. config KERNEL โดยที่ KERNEL คือชื่อของไฟล์ เคอร์เนลที่ได้เพิ่มตัวเลือกเข้าไปแล้ว
2. ทำการเปลี่ยน Directory ไปยัง /usr/src/sys/compile/KERNEL
3. พิมพ์ make depend; make; make install เป็นการคอมไพล์และติดตั้งเคอร์เนลใหม่
4. แล้วทำการ Reboot ระบบใหม่ก็เป็นอันเสร็จสิ้น

หลังจากที่ได้เคอร์เนลที่พร้อมใช้กับ IPFW แล้ว เราก็มาศึกษาเกี่ยวกับการใช้งานไฟร์วอลล์ที่ชื่อ IPFW บนระบบปฏิบัติการ FreeBSD

2.4.1 การใช้งาน IPFW

คำสั่ง ipfw เป็นการสั่งด้วยมือ เพื่อเพิ่มหรือลด กฎการทำงานของ firewall ในขณะที่กำลังใช้งานอยู่ ซึ่ง ปัญหาของการใช้วิธีนี้คือ เมื่อปิดเครื่องลง กฎทั้งหมดที่เราสร้างขึ้น หรือ เปลี่ยนแปลงจะหายไปหมด แต่คำสั่ง ipfw ในลักษณะนี้ ยังมีประโยชน์ เพื่อแสดงให้เห็นกฎ ของ firewall ที่กำลังทำงานอยู่ผ่านทาง console การคำนวณของ IPFW จะแสดงค่าจำนวนนับของ packet ที่ตรงกับกฎแต่ละกฎ ในระหว่างการทดสอบกฎนั้น เมื่อเราดูที่ จำนวนนับเหล่านี้ เป็นการยืนยันว่า กฎที่ตั้งไว้ทำงานได้

คำสั่งสำหรับดูกฎทั้งหมดที่ตั้งไว้ตามลำดับ

ipfw list

คำสั่งสำหรับดูกฎทั้งหมด พร้อมเวลาล่าสุดที่กฎนั้นทำงาน

ipfw -t list

คำสั่งดูข้อมูลของกฎทั้งหมด แถวแรกคือ หมายเลขประจำกฎ ตามด้วยจำนวน packets ขาออกที่ตรงกับกฎ ตามด้วยจำนวน packets ขาเข้าที่ตรงกับกฎ และ ต่อมาเป็นตัวกฎ ที่เราตั้งไว้

ipfw -a list

คำสั่งดูกฎที่ dynamic

ipfw -d list

คำสั่งดูกฎ dynamic ที่หมดอายุ

ipfw -d -e list

คำสั่งกำหนดให้ จำนวนนับเป็น 0 เพื่อเริ่มนับใหม่

ipfw zero

คำสั่งกำหนดให้ จำนวนนับเป็น 0 เพื่อเริ่มนับใหม่ เฉพาะกฎข้อใดข้อหนึ่ง (NUM)

ipfw zero NUM

IPFWRuleSets

rule set คือกลุ่มของกฎ ipfw ที่เขียนขึ้นมาเพื่ออนุญาต หรือปฏิเสธ packet ตามค่าที่อยู่ใน packet นั้นๆ การแลกเปลี่ยนทั้ง 2 ทางคือไป-กลับ ระหว่างการสื่อสารของคอมพิวเตอร์ โดยปกติ rule set ของ firewall จะตรวจสอบ packet 2 รอบเสมอ ครั้งแรกตอนมาจากเครื่องบน Internet ที่ติดต่อมา และ ตอนกลับออกไปยัง Internet ยังเครื่องที่เรียกมา การบริการต่างๆ (เช่น telnet, www , mail และอื่นๆ) ถูกกำหนดโดย protocol ประจำตัว และ เลขประจำ port สิ่งเหล่านี้เป็นขอบเขตการเลือกขั้นต้น ที่ใช้สร้างกฎ สำหรับการอนุญาต หรือ ปฏิเสธ งานบริการเหล่านั้น เมื่อ packet เข้ามายัง firewall มันจะถูกเปรียบเทียบกับกฎข้อที่ 1 ใน rule set และจะดำเนินตามกฎแต่ละข้อ ตั้งแต่บนจนถึง กฎล่างสุด ตามลำดับหมายเลขประจำกฎ และเมื่อ packet ตรงกับกฎข้อใด กฎจะทำงานทันทีกับ packet นั้น ด้วยวิธีการนี้ถูกเรียกว่า การค้นหาแบบ "the first match wins" และถ้า packet นั้นไม่ตรงกับกฎข้อใดๆ เลย มันจะถูกดักจับด้วยกฎข้อหลัก ของ ipfw คือกฎหมายเลข 65535 ซึ่งจะแล้วแต่ผู้ดูแลระบบออกแบบไว้ว่าจะให้ Allow หรือ DENY

Rule Syntax

รูปแบบการเขียนกฎ ที่นำเสนอต่อไปนี้ เป็นแบบธรรมดา เพื่อให้เข้าใจการสร้าง rule set ของ firewall กฎ ประกอบด้วย keywords โดยที่ keywords เหล่านี้ต้องเขียนตามลำดับ จาก ซ้ายไปขวา ในแต่ละบรรทัด keywords แสดงให้เห็นด้วยตัวพิมพ์ใหญ่ บาง keyword มีคำสั่งย่อย ซึ่งอาจจะ

มี keyword สำหรับ ตัวมันเอง และ อาจจะมีคำสั่งย่อยด้วยก็ได้ เครื่องหมาย "#" ใช้เป็นตัวเริ่มต้นประโยคหมายเหตุ และ อาจจะระบุไว้ตอนท้ายบรรทัดของกฎ ขณะที่ บรรทัดว่างจะไม่ถูกตีความ

CMD RULE NUMBER ACTION LOGGING SELECTION STATEFUL

CMD กฎใหม่แต่ละกฎ ต้องมีคำนำหน้าด้วยคำว่า "add" เพื่อสั่งเพิ่มกฎเข้าไปในตารางเก็บข้อมูลภายใน

RULE_NUMBER กฎแต่ละกฎ ต้องมีหมายเลขประจำกฎ ไว้สำหรับทำงานตามหมายเลขนั้น

ACTION กฎจะทำงานร่วมกับคำสั่งต่อไปนี้ ซึ่งจะทำงานทันทีที่ packet ตรงกับข้อกำหนดของกฎ

allow | accept | pass | permit คำสั่งทั้งหมดนี้มีความหมายเหมือนกัน คือ อนุญาตให้ packets ที่ตรงกับกฎ ผ่านออกจาก ตรวจสอบของ firewall ได้ และการตรวจสอบจะสิ้นสุดลง ที่กฎตัวนี้

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

deny | drop ทั้งสองคำมีความหมายเดียวกัน คือ ปฏิเสธ packet ที่ตรงกับกฎการตรวจสอบยุติทันที

Logging log หรือ log amount เมื่อ packet ใดตรงกับกฎ และมีการใช้ log จะมีการบันทึกข้อความไปที่ syslogd ด้วยข้อความว่า SECURITY การบันทึก log จะทำงานต่อเมื่อ จำนวน packet ที่ถูกบันทึก ไม่เกินจำนวนที่ระบุไว้ใน logamount แต่ถ้าไม่มีการระบุ logamount ไว้ การบันทึก log จะตรวจสอบข้อจำกัดของการบันทึก จากค่าที่ให้ไว้ในตัวแปร net.inet.ip.fw.verbose_limit และในทั้งสองกรณีนั้น หากมีค่าเป็น 0 จะเป็นการยกเลิก ข้อจำกัดในการบันทึก log ในกรณีที่การบันทึก log เต็มตามข้อจำกัดที่กำหนดไว้ เราสามารถกำหนดค่าในการจำกัด การบันทึก log ได้ ทั้งนี้ให้ดูคำสั่ง reset log ของ ipfw เพิ่มเติม

Selection คำต่อไปนี้ ใช้เพื่อเป็นการบอกถึงรายละเอียดของ packet เพื่อใช้ในการตรวจสอบว่า packet นั้นตรงกับกฎหรือไม่ และต้องเรียงลำดับตามนี้

udp | tcp | icmp ทั้งนี้ยังสามารถใช้ ชื่อ protocol ที่อยู่ใน /etc/protocols

from src to dst คำว่า from ใช้เพื่อเป็นการตรวจสอบกับ IP address การตั้งกฎต้องระบุทั้งต้นทางและปลายทาง สำหรับคำว่า "any" ใช้ระบุเพื่อให้ตรงกับทุกๆ IP address ส่วนคำว่า "me"

กำหนดเพื่อให้ตรงกับ IP address ที่ใช้ใน FreeBSD ในเครื่องของเราที่มี firewall กำลังทำงานอยู่ ตัวอย่างเช่น "from me to any" หรือ "from any to me" หรือ "from 0.0.0.0/0 to any" หรือ "from any to 0.0.0.0/0" หรือ "from 0.0.0.0 to any" หรือ "from any to 0.0.0.0" หรือ "from me to 0.0.0.0" โดย IP address ถูกระบุตัวเลข IP address และ จุด ตามด้วย /mask-length หรือ สามารถใช้แบบ IP address และจุด เท่านั้น ก็ได้

port number สำหรับ protocols ที่รองรับหมายเลข ports (เช่น TCP และ UDP) มันเป็นข้อบังคับที่เราต้องระบุ หมายเลข ports สำหรับ ข้อมูลที่เราต้องการ ตรวจสอบให้ถูกต้อง ทั้งนี้ชื่อบริการต่างๆ ที่อยู่ใน /etc/services อาจนำมาใช้แทนค่าของหมายเลข port ได้

in | out ใช้แทน packets ที่ตรงกับ ด้านขาเข้า หรือ ด้านขาออก คำว่า in และ out เป็นข้อบังคับที่เราต้องระบุไว้ในกฎ

via IF packets ที่ตรงต้องเข้ามาทาง lan card ที่เราระบุ คำว่า via จะทำให้ lan card ที่ระบุไว้ถูกตรวจสอบเสมอ

keep-state เป็นคำบังคับ โดยอยู่กับการตรวจสอบความถูกต้อง firewall จะสร้างกฎ dynamic ขึ้นมา ซึ่งจะ ตรวจสอบกฎทั้ง 2 ทาง ระหว่าง ต้นทางและปลายทาง IP/port ใช้ protocol เดียวกัน

2.5 ระบบตรวจจับผู้บุกรุก

ในปัจจุบันเมื่อระบบเครือข่ายมีการเชื่อมต่อเข้าหากันมากขึ้น สิ่งที่ตามมาคือผู้ไม่ประสงค์ดีที่อาจสร้างความเสียหายให้กับระบบและข้อมูลขององค์กร ดังนั้นระบบรักษาความปลอดภัยจะต้องมีความมั่นคงพร้อมรับมือต่อการบุกรุกที่อาจจะเกิดขึ้น ซึ่งการใช้งานไฟร์วอลล์อย่างเดียวมิได้ช่วยทำให้ระบบปลอดภัยร้อยเปอร์เซ็นต์ แต่ระบบรักษาความปลอดภัยที่ดีจะต้องป้องกันการโจมตีในทุกระดับ ดังนั้นการทำงานร่วมกับระบบตรวจจับการบุกรุก (Intrusion Detection System) จะช่วยเพิ่มความปลอดภัยขึ้นอีกระดับหนึ่ง ซึ่งระบบตรวจจับการบุกรุกจะทำการเฝ้าดู วิเคราะห์เหตุการณ์ต่างๆว่าเป็นการบุกรุกหรือมีความพยายามในการบุกรุกหรือไม่ โดยอาศัยข้อมูลต่างๆเช่น CPU Utilization, I/O Utilization, Network traffic และทำการเตือนภัยต่างๆที่เกิดขึ้นบนเครือข่ายคอมพิวเตอร์

2.5.1 ความหมายของระบบตรวจจับการบุกรุก

ระบบตรวจจับการบุกรุก (Intrusion Detection System) คือ ระบบที่ใช้ในการเฝ้าดูวิเคราะห์ตรวจจับเหตุการณ์และเตือนภัยเมื่อมีผู้บุกรุกหรือมีสิ่งผิดปกติที่เข้ามาในระบบ ซึ่งความพยายามในการใช้งานระบบของผู้บุกรุกนั้นขัดต่อข้อบังคับหรือเจตุจำนงการใช้งานซึ่งส่งผลต่อ

ความปลอดภัยของระบบคอมพิวเตอร์หรือเครือข่ายคอมพิวเตอร์ 3 ประการคือ Integrity, Confidentiality, Availability

ประเภทของระบบตรวจจับการบุกรุก

- Host base Intrusion Detection System(HIDS)

  HIDS เป็นระบบตรวจจับการบุกรุกประเภทแรกที่มีการพัฒนาและใช้งาน ซึ่งระบบจะทำการตรวจสอบผู้บุกรุกเฉพาะเครื่องที่ลงระบบ HIDS เอาไว้ โดยการตรวจสอบจะทำการวิเคราะห์ข้อมูลการใช้งานภายในเครื่อง เช่นวิเคราะห์ Log file, CPU Utilization, การ Log in เข้าใช้งาน, การ Access file

- Network based Intrusion Detection System(NIDS)

  NIDS นั้นจะทำการเฝ้าดูข้อมูลบนเครือข่ายที่ติดตั้งระบบนี้ไว้ โดยทำการรับข้อมูลทั้งหมดบนเครือข่ายที่รับผิดชอบ โดยจะมองเห็นเฉพาะ packet ที่ผ่านส่วนของเครือข่ายที่ sensor นั้นติดอยู่ ซึ่ง Packetต่างๆ จะเป็นที่สนใจของ sensor ก็ต่อเมื่อ packet นั้นเข้ากับ signature ที่กำหนด ซึ่งปกติแล้ว signature มี 3 ประเภทคือ

  1) String signatures จะมองหา text string ซึ่งอาจบ่งบอกถึงการโจมตี
  2) Port signatures จะเฝ้าดูการพยายามติดต่อเข้ามาทาง Port ที่รู้จักกันดีและมักจะถูกโจมตี เช่น telnet จะใช้ TCP port 23, FTP จะใช้ TCP port 21/20, IMAP จะใช้ TCP port 143 ซึ่งถ้าระบบไม่ได้เปิด port ดังกล่าว แต่มีการพยายามเชื่อมต่อเข้ามาแสดงว่า packet ดังกล่าวอาจจะประสงค์ร้าย
  3) Header condition signatures พยายามมองหา combination ที่อันตรายและผิดกฎของ packet header เช่น TCP packet ซึ่งมีทั้ง SYN และ FIN Flags มาในคราวเดียวกัน

2.5.2 กลวิธีในการตรวจจับการบุกรุก

- **การตรวจจับพฤติกรรมที่ผิดปกติ(Anomaly Detection)**

  ใช้ในการพิจารณาจากสถิติต่างๆ เช่น CPU Utilization การเข้าใช้ระบบของผู้ใช้ โดยค่าสถิติเปล่านี้จะมีค่า Threshold เพื่อเป็นตัวเปรียบเทียบว่าเหตุการณ์นั้นมีพฤติกรรมที่ผิดปกติหรือมีการบุกรุกหรือไม่ ประโยชน์ของวิธีนี้คือ สามารถจับความผิดปกติต่างๆ โดยที่ไม่ต้องรู้สาเหตุของความผิดปกตินั้น

  - การตรวจจับพฤติกรรมที่ผิดปกติมีข้อดีดังนี้

สามารถตรวจจับพฤติกรรมที่ผิดปกติที่ไม่เคยเกิดขึ้นมาก่อนได้
สถิติในการจัดเก็บง่ายต่อการเข้าใจ

- การตรวจจับพฤติกรรมที่ผิดปกติมีข้อเสียดังนี้
  การกำหนดค่า Threshold ที่ถูกต้องทำได้ยาก อาจจะทำให้ระบบตัดสินใจผิดพลาด อาจทำให้ผู้ใช้ที่เจตนาดีถูกมองว่าบุกรุกได้

- **การตรวจจับรูปแบบของการโจมตี (Signature Detection)**

เรียกอีกอย่างหนึ่งว่า การตรวจจับการใช้งานที่ผิด (Misuse Detection) ใช้วิธีตรวจสอบหรือศึกษารูปแบบการโจมตี หมายความว่าเทคนิคต่างๆที่ผู้บุกรุกใช้ก็จะถูกบันทึกเป็นกฎเข้าสู่ระบบเพื่อทำการตรวจจับการบุกรุกต่อไป

- การตรวจจับพฤติกรรมการใช้งานที่ผิดมีข้อดีดังนี้
  มีประสิทธิภาพมากต่อการตรวจจับโดยไม่มีการเตือนภัยที่ผิดพลาด
  การทำงานไม่จำเป็นต้องใช้การคำนวณที่ซับซ้อนเพื่อเก็บข้อมูล

- การตรวจจับพฤติกรรมการใช้งานที่ผิดมีข้อเสียดังนี้
  การทำงานตามกฎนั้น จำเป็นต้องนำเหตุการณ์มาเปรียบเทียบกับกฎทุกกฎ ซึ่งถ้ามีกฎมากทำให้การทำงานช้าได้

2.6 โปรแกรมตรวจจับการบุกรุก SNORT

SNORT เป็นระบบตรวจจับการบุกรุก (Intrusion Detection System) นั้นมีมาตั้งแต่ปี 1998 และในปัจจุบัน มีการพัฒนาเพื่อนำมาประยุกต์ใช้งานอยู่เสมอซึ่ง SNORT เป็น Open source พัฒนาโดย ภาษา C และสามารถติดตั้งใช้งานได้หลายระบบปฏิบัติการ SNORT เป็นเครื่องมือที่ใช้ในการตรวจจับการบุกรุกทางเครือข่าย โดยการทำงานของ SNORT จะใช้ ไลบรารี พื้นฐานชื่อ WinPCAP ซึ่งใช้กันทั่วไปในบรรดา network sniffer และ network analyzer ทั้งหลาย โดยสามารถทำการวิเคราะห์ข้อมูลที่รับส่งกันบนเครือข่ายแบบ Real Time, สามารถบันทึก Packet, วิเคราะห์ Protocol, ตรวจสอบข้อมูลใน Packet, ตรวจจับการโจมตีในรูปแบบต่างๆได้ เช่น Buffer Overflows, Port Scan, DOS, Ping of death, IP Spoofing และอื่นๆ เมื่อทำการตรวจพบการโจมตีก็จะทำการแจ้งเตือนได้

2.6.1 โหมดการทำงานของ SNORT

SNORT มีการทำงานอยู่ด้วยกัน 3 โหมด คือ

1. Sniffer Mode การดัก Packet บนเครือข่าย โดยเป็นการเปิดโหมดของ Network Interface Card เป็นแบบ Promiscuous Mode

2. Packet Logger Mode การบันทึกข้อมูลบนเครือข่ายเก็บลงดิสก์ไว้

3. Network Intrusion System Mode จะทำการวิเคราะห์ตรวจสอบข้อมูลที่ทำการเก็บมาว่าเป็นลักษณะ Packet ที่ตรงกับรูปแบบของการโจมตีตามกฎหรือไม่ โดยกฎที่ใช้ในการตัดสินใจว่าเป็นการโจมตีหรือไม่นั้นจะเก็บไว้ที่ snort.conf ซึ่งกฎที่มีอยู่นั้นจะมีผลโดยตรงต่อการตรวจจับการบุกรุกของ SNORT เช่น ถ้าไม่มีกฎที่ตรงกับรูปแบบการโจมตี SNORT ก็จะไม่สามารถตรวจจับการโจมตีแบบนั้นได้ หรือถ้ามีกฎที่มากเกินไปก็อาจจะเกิดการเข้าใจผิดคิดว่า Packet ที่ปกติเป็นการโจมตี (False Alert) ก็จะทำให้ส่งผลต่อประสิทธิภาพการทำงาน

2.6.2 ลักษณะกฎของโปรแกรม SNORT

กฎของ SNORTแบ่งเป็น 2 ส่วน คือ Rule Header และ Rule Option

Rule Header	Rule Options

รูปที่ 2.6 รูปแบบกฎของ SNORT

Rule Header ประกอบด้วย

Action	Protocol	Address	Port	Direction	Address	Port

รูปที่ 2.7 รายละเอียดของ Rule Header

Action เป็นส่วนที่บอกว่าเมื่อพบ Packet ที่ตรงกับกฎนี้จะให้ทำอะไรมีหลายการกระทำ เช่น

1. Alert ให้ทำการแจ้งเตือนว่าพบการบุกรุก หลังจากนั้นจะบันทึกข้อมูลการบุกรุกไว้ที่ไฟล์

   /var/log/snort/alert

2. Log จะทำการบันทึกว่ามีการโจมตีไว้แต่ไม่มีการแจ้งเตือนว่ามีการโจมตีเกิดขึ้น
3. Pass จะทำการละทิ้ง Packet นั้นไป
4. Activate จะทำการแจ้งเตือนและไปทำ Dynamic rule
5. Dynamic จะไม่ทำงานจนกว่าจะได้รับการกระตุ้นจาก Activate

Protocol เป็นการบ่งบอกประเภทของ Protocol ซึ่งได้แก่ IP, ICMP, TCP, UDP

Source IP Address and Net Mask คือ ค่า IP address และ Net Mask ของเครื่องต้นทาง เช่น 92.168.1.0/24 หรืออาจจะระบุเป็น any จะหมายถึงหมายเลขอะไรก็ได้

Destination IP Address and Net Mask คือ ค่า IP address และ Net Mask ของเครื่องปลายทาง เช่น 192.168.1.0/24 หรืออาจจะระบุเป็น any จะหมายถึงหมายเลขอะไรก็ได้

Source Port คือหมายเลข Port ของเครื่องต้นทางที่ใช้ในการสื่อสาร เช่น 23 (Telnet) หรือระบุเป็น any เพื่อบอกว่าเป็นหมายเลขอะไรก็ได้

Destination Port คือหมายเลข Port ของเครื่องปลายทางที่ใช้ในการสื่อสาร เช่น 80 (HTTP) หรือระบุเป็น any เพื่อบอกว่าเป็นหมายเลขอะไรก็ได้

Direction คือ การบอกทิศทางการติดต่อ โดย ฝั่งซ้ายของเครื่องหมายคือ ต้นทาง ส่วนด้านขวาของเครื่องหมายคือ ปลายทาง เช่น

1. -> ลักษณะการเดินทางของ Packet โดยเดินทางจาก IP address and Port ทางซ้ายไปทางขวา
2. <- ลักษณะการเดินทางของ Packet โดยเดินทางจาก IP address and Port ทางขวาไปทางซ้าย
3. <> ลักษณะการพิจารณาทั้งสองฝั่ง

Rule Option

มีรูปแบบดังนี้ (Keyword$_1$: Argument$_1$; Keyword$_2$: Argument$_2$; …… Keyword$_n$: Argument$_n$;)

เช่น msg: "Detected confidential"; (msg คือKeyword ส่วน Detected confidential คือ Argument)

ซึ่ง Keyword ที่ใช้ใน Snort มีหลายแบบ เช่น

1. msg: “<message text>”;

   ทำการระบุข้อความที่ต้องการให้ถูกแสดง ตอนแจ้งเตือนและตอนบันทึกลง Log
2. logto :”<filename>”;

   ทำการบันทึก Packet ไปยังไฟล์ที่ระบุ แทนที่จะบันทึกรวมกับ log ไฟล์อื่นๆ
3. ttl:”<number>”;

   ตรวจสอบค่า TTL ของ Packet ว่าตรงกับที่ระบุหรือไม่
4. tos:”<number>”;

   ตรวจสอบค่า TOS ของ Packet ว่าตรงกับที่ระบุหรือไม่
5. ack:”<number>”;

   ตรวจสอบค่าใน ACK field ว่าตรงกับที่ระบุไว้หรือไม่

2.6.3 Output Modules

ใช้สำหรับการควบคุมผลลัพธ์ที่ได้หลังจากทำการตรวจพบการบุกรุก โดยปกติเมื่อตรวจพบการบุกรุกจะทำการแจ้งเตือนและทำการบันทึก log ไว้ที่ /var/log/snort โดยรูปแบบของคำสั่งที่

จะระบุว่าจะใช้ Output Modules แบบใดมีรูปแบบดังนี้ output <module_name>[: arguments] โดยจะทำการระบุไว้ในไฟล์ snort.conf ตัวอย่างเช่น

output database: log, mysql, user=rr password=rr \ dbname=snort host=localhost เป็นการระบุว่าให้ทำการบันทึก log ลงในฐานข้อมูล โดยรูปแบบของ Output เราสามารถกำหนดไว้ในไฟล์ snort.conf มีดังต่อไปนี้

1. **Alert_syslog** ระบบจะส่ง Alert ไปยัง syslog ของระบบ

   รูปแบบคำสั่ง

   output alert_syslog: <facility> <priority> <options>

2. **Alert_full** จะทำการเก็บรายละเอียดของPacket Header ทั้งหมด

   รูปแบบคำสั่ง

   output alert_full: alert_detailed

3. **Alert_fast** แต่ละ Alert Message จะทำการบันทึกเพียง 1 บรรทัด เพื่อความรวดเร็ว

   รูปแบบคำสั่ง

   output alert_fast: alert_quick

4. **Alert_smb** ระบบจะส่ง Alert Message ไปที่ WinPopUp ผ่านทาง Protocol SMB ซึ่งเครื่องที่ได้รับจะมีPopup ข้อความปรากฏขึ้นมา

   รูปแบบคำสั่ง

   output alert_smb: workstation.list

5. **Log_tcpdump** เก็บข้อมูลให้อยู่ในรูปของ tcpdump ซึ่งสามารถใช้โปรแกรมที่เปิดไฟล์รูปแบบ tcpdump มาทำการเปิดดูภายหลังได้

   รูปแบบคำสั่ง

   output log_tcpdump: <filename>

6. **XML** เก็บข้อมูลให้อยู่ในรูปแบบของ XML

   รูปแบบคำสั่ง

   output xml: [log | alert], [parameter list]

7. **Database** ทำการบันทึก log ลงฐานข้อมูลได้

   รูปแบบคำสั่ง

   output database: <log | alert>, <database_type>, <parameter_list>

8. **CVS** เมื่อ CVS Plug in จะบันทึก alert Message อยู่แบบที่ง่ายต่อการ Import ไปยังฐานข้อมูล

   รูปแบบคำสั่ง

   output csv: <filename> <formatting_options>

9. **Unified Logging** ถูกออกแบบให้มีความเร็วมากที่สุดในการบันทึก log ของ Snort ซึ่งแบ่งเป็น 2 ส่วนคือ ไฟล์ Alert และ Log แยกจากกัน โดยเก็บอยู่ในรูปแบบของไบนารี่ไฟล์

รูปแบบคำสั่ง

output alert_unified: filename <alert_file>, limit <max_size>

output log_unified: filename <log_file>, limit <max_size>

จากข้อมูลข้างต้นจะเห็นว่าไฟร์วอลล์จะมีหน้าที่คอยตรวจสอบว่า Packet ใดมีสิทธิในการเข้าหรือออกจากเครือข่าย ส่วนระบบตรวจจับการบุกรุกนั้นจะมีหน้าที่คอยตรวจสอบการบุกรุกเครือข่ายและทำการส่งข้อมูลการบุกรุกนั้นเก็บลงในฐานข้อมูล ทำให้สามารถนำคุณสมบัติทั้งสองอย่างนี้มาประยุกต์ใช้งานกับโปรแกรมที่กำลังจะพัฒนาขึ้นได้ โดยจะกล่าวถึงการออกแบบระบบในบทถัดไป

บทที่ 3
การวิเคราะห์และออกแบบ

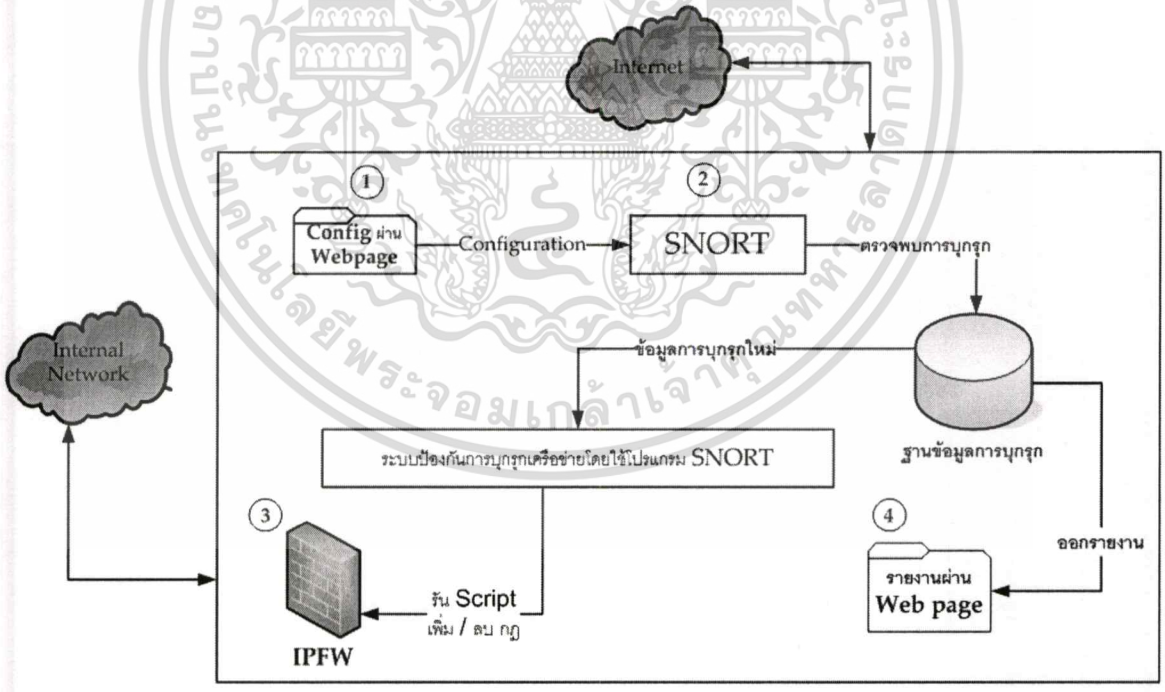
3.1 ความต้องการของระบบ

ก่อนที่จะพัฒนาระบบนั้นต้องเตรียมความพร้อมก่อน โดยระบบมีความต้องการพื้นฐานดังนี้

- โปรแกรมตรวจจับการบุกรุก SNORT เพื่อใช้ตรวจจับการบุกรุก
- โปรแกรมฐานข้อมูล Mysql เพื่อใช้เป็นที่เก็บข้อมูลการบุกรุก
- ไฟร์วอลล์ IPFW เพื่อใช้ในการป้องกันการบุกรุก
- โปรแกรมภาษา PHP ที่ทำการเขียนขึ้น เพื่ออ่านข้อมูลการบุกรุกจากฐานข้อมูลแล้วนำไปสร้างกฎป้องกันการบุกรุกที่ไฟร์วอลล์

3.2 ลักษณะการทำงานของระบบ

หลังจากการทำการศึกษาการทำงานของโปรแกรม SNORT และ ไฟร์วอลล์ IPFW แล้วจึงได้ทำการออกแบบโครงสร้างและลักษณะการทำงานของระบบ โดยมีภาพรวมการทำงานดังนี้



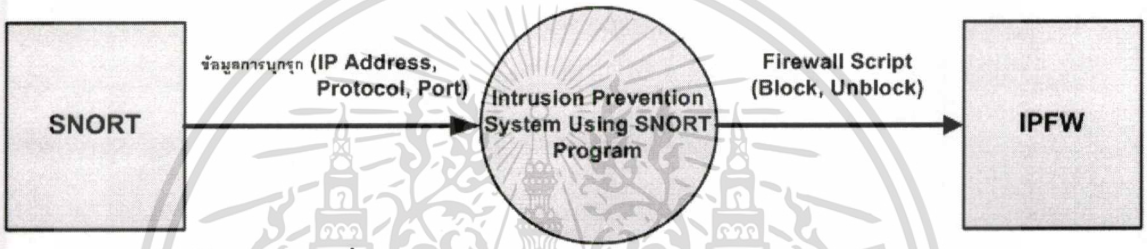
รูปที่ 3.1 แสดงภาพลักษณะการทำงานเมื่อมีการใช้งานโปรแกรม SNORT และ IPFW

จากรูปจะเห็นว่าเมื่อทำการติดตั้งโปรแกรม SNORT และ IPFW แล้ว ระบบมีช่องทางให้สามารถทำการปรับแต่งค่าโปรแกรม SNORT ผ่านทาง Webpage ได้ โดยเมื่อโปรแกรม SNORT ทำการตรวจพบการบุกรุกก็จะทำการบันทึกข้อมูลการบุกรุกนั้นลงในฐานข้อมูล และโปรแกรมที่

พัฒนาก็จะไปอ่านข้อมูลจากฐานข้อมูลเป็นช่วงเวลา ถ้าพบการบุกรุกใหม่ก็จะทำการอ่านข้อมูลที่จำเป็นในการสั่งให้ IPFW ทำการป้องกันการบุกรุกได้แก่ IP Address ต้นทาง ปลายทาง, Port ต้นทาง ปลายทาง, Protocol เมื่อได้ข้อมูลดังกล่าว ก็จะทำการรันสคริปต์คำสั่งๆให้ไฟร์วอลล์ป้องกันการบุกรุกที่มี IP Address, Protocol, Port ต้นทางและปลายทางที่ตรงกับข้อมูลที่ตรวจพบนั้น ซึ่งเมื่อครบกำหนดเวลาก็สามารถลบกฎป้องกันการบุกรุกออกจากไฟร์วอลล์ IPFW ได้ทันที และยังสามารถมอนิเตอร์สถานะของกฎที่ใช้งานอยู่หรือถูกลบไปแล้วผ่านทาง Webpage ได้

3.3 แผนภาพแสดงการไหลของข้อมูล

Context Diagram

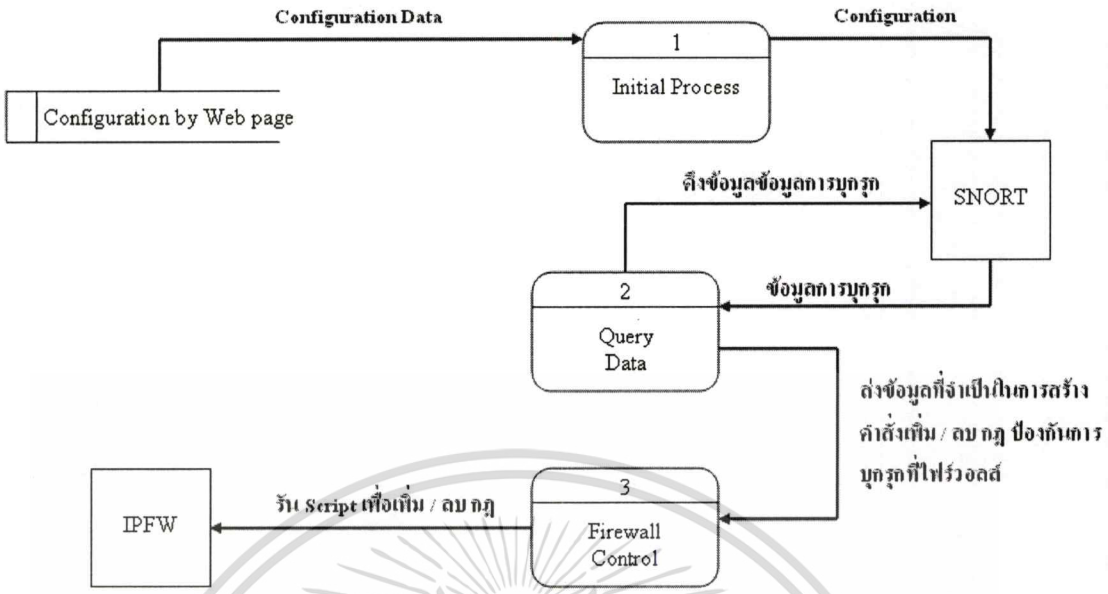


รูปที่ 3.2 แสดงContext diagram ของระบบ

ระบบป้องกันการบุกรุกเครือข่ายที่พัฒนาขึ้นนี้จะทำงานร่วมกับ 2 อินเตอร์เฟส คือ

1. SNORT ระบบที่พัฒนาจะทำการอ่านค่าข้อมูลการบุกรุกที่เข้ามาในเครือข่ายจากโปรแกรม SNORT โดยข้อมูลที่จำเป็นได้แก่ IP Address, Protocol, Port ต้นทาง ปลายทาง
2. Firewall (IPFW) ระบบที่พัฒนาจะทำการส่งคำสั่งไปยังไฟร์วอลล์ให้ทำการ Block หรือ Unblock การบุกรุกนั้น

Data Flow Diagram Level 1



รูปที่ 3.3 แสดง Data Flow Diagram Level 1

จากรูป จะเห็นว่ามีส่วนการทำงานดังนี้

- **Initial Process (1)**

  เป็นการปรับแต่งค่าของโปรแกรม SNORT ผ่านทางหน้า Webpage ได้

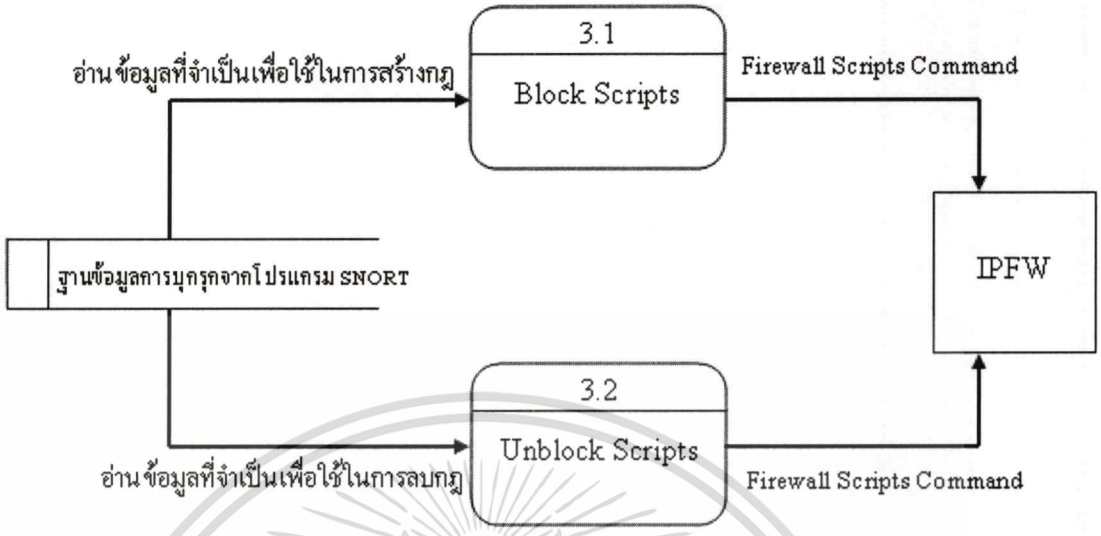
- **Query Data (2)**

  เป็นการดึงข้อมูลการบุกรุกจากฐานข้อมูลการบุกรุกที่โปรแกรม SNORT ตรวจจับได้

- **Firewall Control (3)**

  เป็นนำข้อมูลที่ได้รับมาสร้างกฎ เพื่อสั่งให้ไฟร์วอลล์ป้องกันการบุกรุก ข้อมูลที่ต้องการได้แก่ IP Address, Protocol, Port ต้นทางและปลายทาง แล้วทำการสั่งไฟร์วอลล์ให้ทำการป้องกันการบุกรุกนั้น จากนั้นเมื่อถึงช่วงเวลาที่กำหนดก็จะทำการลบกฎออกจากไฟร์วอลล์ โดยทำการอ่านหมายเลขกฎที่ครบกำหนดเวลาแล้วจากฐานข้อมูลการบุกรุก ข้อมูลที่ต้องการได้แก่ หมายเลยกฎที่ครบกำหนดเวลาแล้ว จากนั้นทำการสั่งไฟร์วอลล์ให้ทำการลบกฎนั้นออกจากระบบ

Data Flow Diagram Level 2 Process 3



รูปที่ 3.4 แสดง Data Flow Diagram Level 2 Process 3

จากรูป จะเห็นว่ามีส่วนการทำงานดังนี้

- **Block Scripts (3.1)**

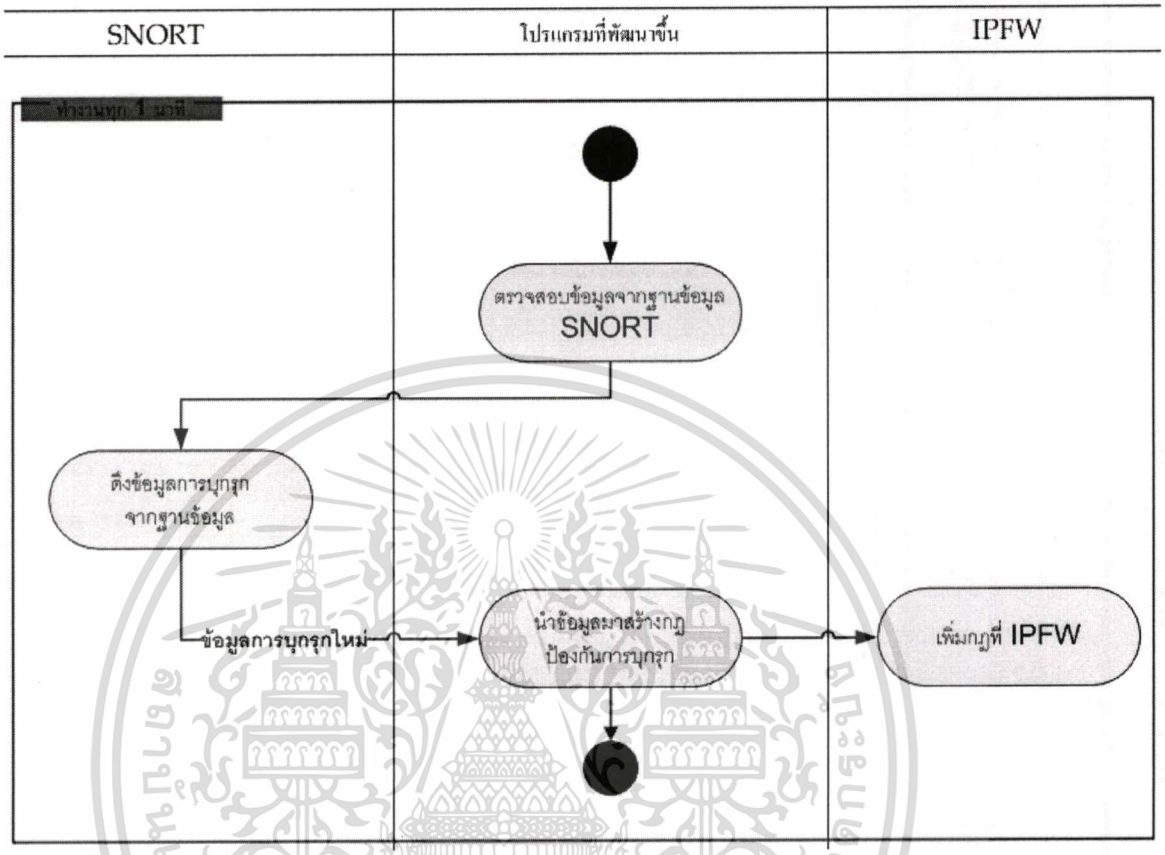
เป็นการอ่านข้อมูลการบุกรุกใหม่โดยทำการอ่านข้อมูลได้แก่ IP Address, Protocol, Port ต้นทาง ปลายทาง จากนั้นจะไปทำการสั่งรันสคริปต์ไฟร์วอลล์โดยทำการเพิ่มกฎเพื่อป้องกันการบุกรุกนั้น

- **Unblock Scripts (3.2)**

เป็นการอ่านค่าหมายเลขกฎที่ครบกำหนดเวลาแล้ว โดยจะทำการตรวจสอบเลขกฎจากฐานข้อมูลว่ามีเลขกฎใดบ้างครบกำหนดเวลาแล้ว จากนั้นจะทำการสั่งรันสคริปต์ไฟร์วอลล์เพื่อทำการลบเลขกฎนั้นออกจากไฟร์วอลล์

Flow การเพิ่มกฎ

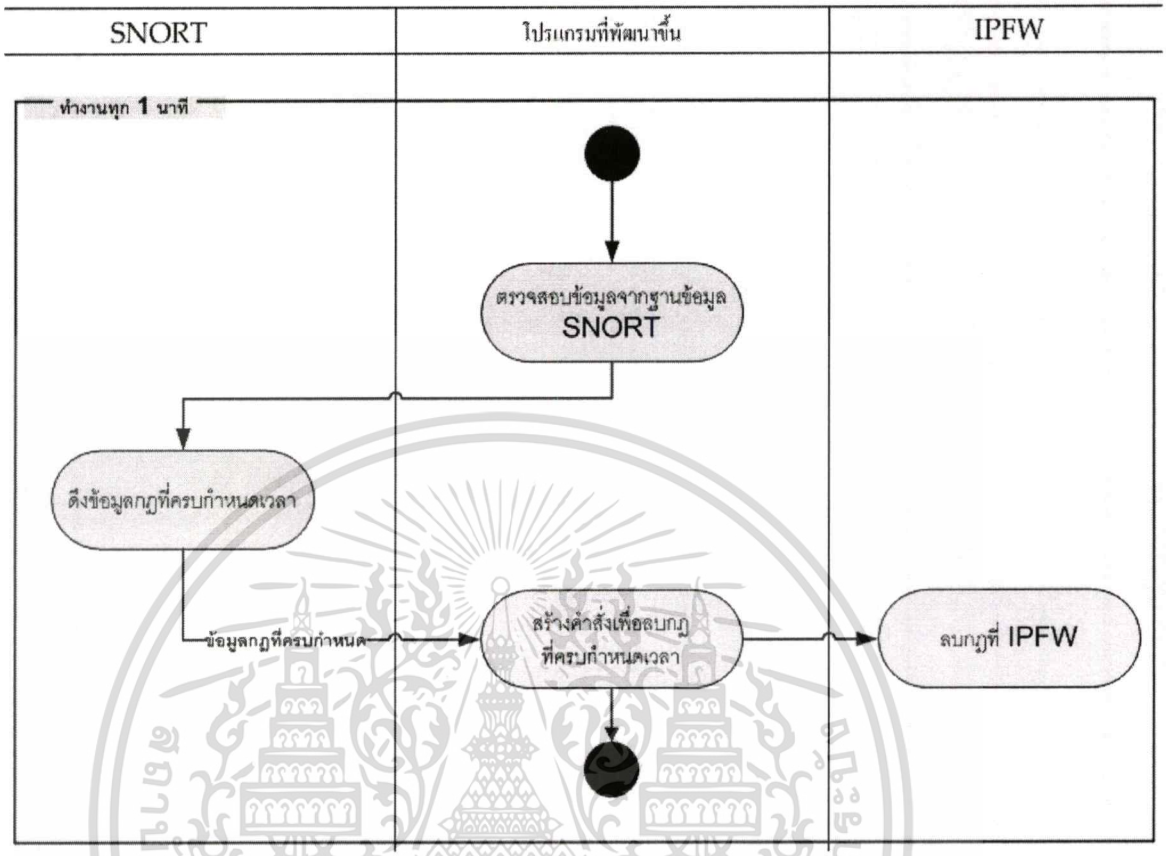
แสดงรายละเอียดเกี่ยวกับการเพิ่มกฎป้องกันการบุกรุกที่ไฟร์วอลล์ IPFW



รูปที่ 3.5 แสดง Flow การเพิ่มกฎป้องกันการบุกรุก

Flow การลบกฎ

แสดงรายละเอียดเกี่ยวกับการลบกฎป้องกันการบุกรุกที่ไฟร์วอลล์ IPFW



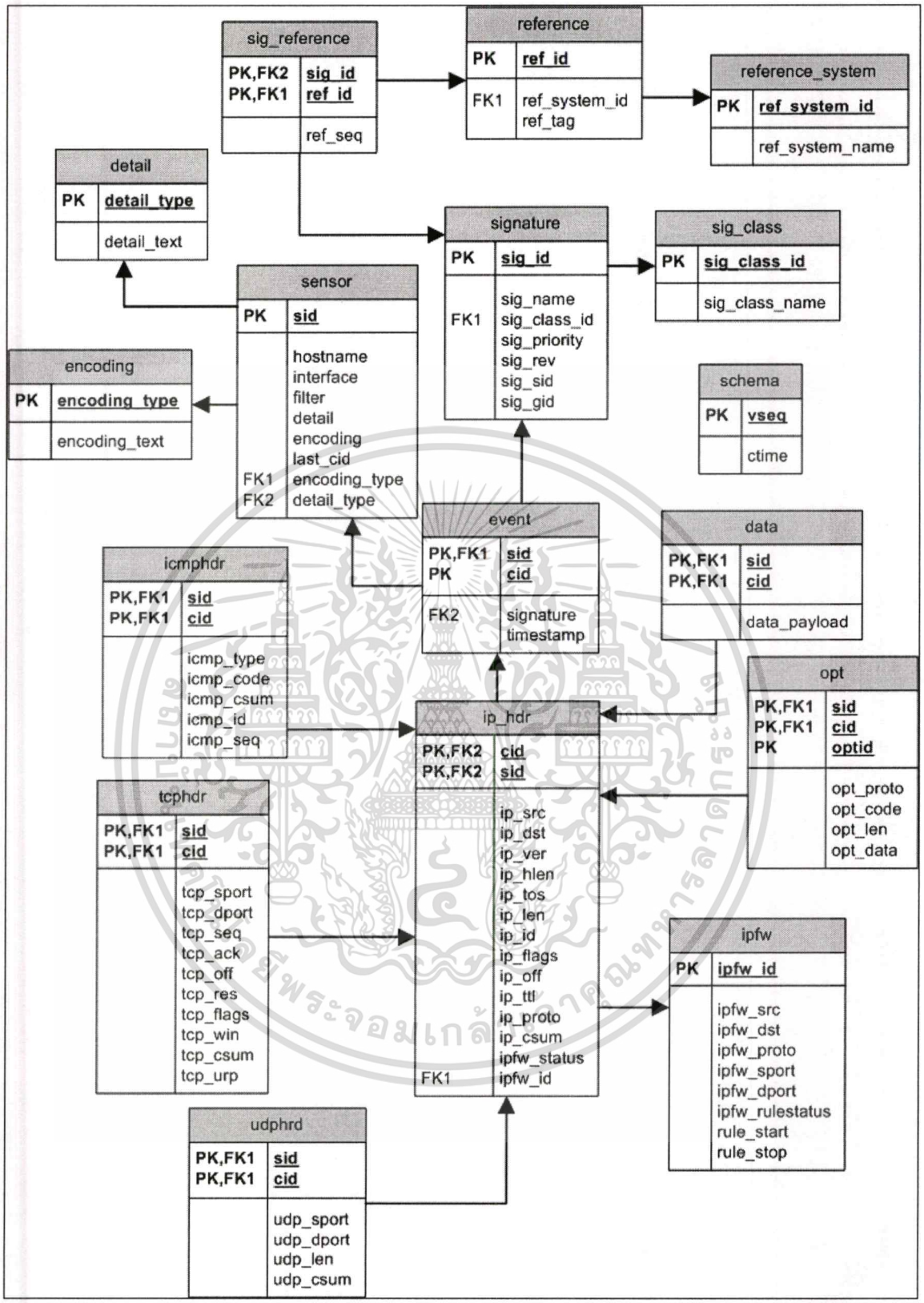
รูปที่ 3.6 แสดง Flow การลบกฎป้องกันการบุกรุก

3.4 การออกแบบการทำงานของระบบ

โครงสร้างการทำงานของระบบแบ่งเป็นสองส่วนหลักๆ คือ

1) ส่วนการแจ้งเตือนการบุกรุกจาก SNORT

ส่วนนี้เกี่ยวข้องกับการเก็บข้อมูลการบุกรุกเมื่อโปรแกรม SNORT ตรวจพบการบุกรุกจะทำการเก็บข้อมูลการบุกรุกลงในฐานข้อมูล ซึ่งในฐานข้อมูลจะเก็บรายละเอียดต่างๆของการบุกรุกไว้ โดยโปรแกรมที่พัฒนาขึ้นก็จะไปดึงเอาข้อมูลในฐานข้อมูลนี้มาใช้ประโยชน์ในการสร้างกฎในการป้องกันการบุกรุก โดยฐานข้อมูลของ SNORT มีรูปแบบดังนี้



รูปที่ 3.7 แสดงฐานข้อมูล SNORT ที่ใช้เก็บข้อมูลการบุกรุก

จากรูปข้างต้น เป็นฐานข้อมูลที่ใช้ในระบบป้องกันการบุกรุกเครือข่ายโดยเมื่อโปรแกรม SNORT ตรวจจับพบการบุกรุกจะทำการบันทึกข้อมูลการบุกรุกลงในฐานข้อมูลของ SNORT โดยมีตารางทั้งหมด 17 ตารางดังนี้

1. data เป็นตารางที่เก็บข้อมูลเกี่ยวกับ Contents of packet payload
2. detail เป็นตารางที่เก็บข้อมูลเกี่ยวกับ (lookup table) Level of detail with which a sensor is logging
3. encoding เป็นตารางที่เก็บข้อมูลเกี่ยวกับ (lookup table) Type of encoding used for the packet payload
4. event เป็นตารางที่เก็บข้อมูลเกี่ยวกับ Meta-data about the detected alert
5. icmphdr เป็นตารางที่เก็บข้อมูลเกี่ยวกับ ICMP protocol fields
6. iphdr เป็นตารางที่เก็บข้อมูลเกี่ยวกับ IP protocol fields
7. ipfw เป็นตารางที่เก็บข้อมูลเกี่ยวกับ หมายเลขกฎ, IP Address, Protocol, Port, Status ของกฎ, เวลา
8. opt เป็นตารางที่เก็บข้อมูลเกี่ยวกับ IP and TCP options
9. reference เป็นตารางที่เก็บข้อมูลเกี่ยวกับ Reference IDs for a signature
10. reference_system เป็นตารางที่เก็บข้อมูลเกี่ยวกับ (lookup table) Reference system list
11. schema เป็นตารางที่เก็บข้อมูลเกี่ยวกับ Self-documented information about the database
12. sensor เป็นตารางที่เก็บข้อมูลเกี่ยวกับ Sensor name
13. sig_class เป็นตารางที่เก็บข้อมูลเกี่ยวกับ Normalized listing of alert/signature classifications
14. sig_reference เป็นตารางที่เก็บข้อมูลเกี่ยวกับ Reference information for a signature
15. signature เป็นตารางที่เก็บข้อมูลเกี่ยวกับ Normalized listing of alert/signature names, priorities, and revision IDs
16. tcphdr เป็นตารางที่เก็บข้อมูลเกี่ยวกับ TCP protocol fields
17. udphdr เป็นตารางที่เก็บข้อมูลเกี่ยวกับ UDP protocol fields

ตารางที่ 3.1 พจนานุกรมข้อมูลตาราง data

Attribute Name	Description	Type	Key	Reference Table
sid	Sensor ID	NUMERIC(10,0)	PK,FK	event
cid	Event ID	NUMERIC(10,0)	PK	event
data_payload	Packet payload encoded according to sensor. encoding	VARCHAR(8000)		

ตารางที่ 3.2 พจนานุกรมข้อมูลตาราง detail

Attribute Name	Description	Type	Key	Reference Table
detail_type		TINYINT	PK	
detail_text		VARCHAR(50)		

ตารางที่ 3.3 พจนานุกรมข้อมูลตาราง encoding

Attribute Name	Description	Type	Key	Reference Table
encoding_type		TINYINT	PK	
encoding_text		VARCHAR(50)		

ตารางที่ 3.4 พจนานุกรมข้อมูลตาราง event

Attribute Name	Description	Type	Key	Reference Table
sid	Sensor ID	NUMERIC(10,0)	PK,FK	sensor
cid	Event ID	NUMERIC(10,0)	PK	
signature	Signature ID	NUMERIC(10,)	FK	signature
timestamp	Timestamp of when the event was logged	DATETIME		

ตารางที่ 3.5 พจนานุกรมข้อมูลตาราง icmphdr

Attribute Name	Description	Type	Key	Reference Table
sid	Sensor ID	NUMERIC(10,0)	PK,FK	event
cid	Event ID	NUMERIC(10,0)	PK	event
icmp_type	ICMP type	TINYINT		
icmp_code	ICMP code	TINYINT		
icmp_csum	ICMP code	INT		
icmp_id	ICMP ID	INT		
icmp_seq	ICMP sequence number	INT		

ตารางที่ 3.6 พจนานุกรมข้อมูลตาราง iphdr

Attribute Name	Description	Type	Key	Reference Table
sid	Sensor ID	NUMERIC(10,0)	PK,FK	event
cid	Event ID	NUMERIC(10,0)	PK	event
ip_src	Source IP address (32-bit unsigned int)	NUMERIC(10,0)		
ip_dst	Destination IP address (32-bit unsigned int)	NUMERIC(10,0)		
ip_ver	IP version	TINYINT		
ip_hlen	IP Header length	TINYINT		
ip_tos	IP type-of-service	TINYINT		
ip_len	IP datagram length	INT		
ip_id	IP ID	INT		
ip_flags	IP flags	TINYINT		

ตารางที่ 3.6 (ต่อ)

ip_off	IP fragment offset	INT		
ip_ttl	IP time-to-live	TINYINT		
ip_proto	IP protocol	TINYINT		
ip_csum	IP checksum	INT		

ตารางที่ 3.7 พจนานุกรมข้อมูลตาราง ipfw

Attribute Name	Description	Type	Key	Reference Table
ipfw_id		INT	PK	
ipfw_src		INT		
ipfw_dst		INT		
ipfw_proto		VARCHAR(10)		
ipfw_sport		INT		
ipfw_dport		INT		
ipfw_rulestatus		INT		
ipfw_rulestart		TIMESTAMP		
ipfw_rulestop		TIMESTAMP		

ตารางที่ 3.8 พจนานุกรมข้อมูลตาราง opt

Attribute Name	Description	Type	Key	Reference Table
sid	Sensor ID	NUMERIC(10,0)	PK,FK	iphdr
cid	Event ID	NUMERIC(10,0)	PK,FK	tcphdr
optid	Option ID (multiple options per alert)	NUMERIC(10,0)	PK	
opt_proto	Option protocol	TINYINT		
opt_code	Option code	TINYINT		

ตารางที่ 3.8 (ต่อ)

opt_len	Option length	INT		
opt_data	Option data	VARCHAR(8000)		

ตารางที่ 3.9 พจนานุกรมข้อมูลตาราง reference

Attribute Name	Description	Type	Key	Reference Table
ref_id	Reference ID	NUMERIC(10,0)	PK	
ref_system	Reference system ID	NUMERIC(10,0)	FK	reference_system
ref_tag	Reference tag (e.g. CVE-CAN-2001-01)	VARCHAR(8000)		

ตารางที่ 3.10 พจนานุกรมข้อมูลตาราง reference_system

Attribute Name	Description	Type	Key	Reference Table
ref_system_id	Reference system ID	NUMERIC(10,0)	PK	
ref_system_name	Reference system name (e.g. CVE	VARCHAR(20)		

ตารางที่ 3.11 พจนานุกรมข้อมูลตาราง schema

Attribute Name	Description	Type	Key	Reference Table
vseq	Database schema ID number (e.g. '102')	NUMERIC(10,0)	PK	
ctime	Timestamp of database creation time	DATETIME		

ตารางที่ 3.12 พจนานุกรมข้อมูลตาราง sensor

Attribute Name	Description	Type	Key	Reference Table
sid	Sensor ID	NUMERIC(10,0)	PK	
hostname	Hostname of the sensor (IP if can't qualify)	VARCHAR(100)		
Interface	Network interface (e.g. eth0)	VARCHAR(100)		
filter	BPF filter	VARCHAR(100)		
detail	Detail level of the logging	INT	FK	detail
encoding	Encoding format of the payload	INT	FK	detail
last_cid		NUMERIC(10,0)		

ตารางที่ 3.13 พจนานุกรมข้อมูลตาราง sig_class

Attribute Name	Description	Type	Key	Reference Table
sig_class_id	Signature classification ID	NUMERIC(10,0)	PK	
sig_class_name	Classification name (e.g. recon)	VARCHAR(60)		

ตารางที่ 3.14 พจนานุกรมข้อมูลตาราง sig_reference

Attribute Name	Description	Type	Key	Reference Table
sig_id	Signature ID	NUMERIC(10,0)	PK,FK	signature
ref_seq	Reference sequence number (multiple references)	NUMERIC(10,0)	PK	
ref_id	Reference ID	NUMERIC(10,0)	FK	reference

ตารางที่ 3.15 พจนานุกรมข้อมูลตาราง signature

Attribute Name	Description	Type	Key	Reference Table
sig_id	Signature ID	NUMERIC(10,0)	PK	
sig_name	Signature Name	VARCHAR(255)		
sig_class_id	Classification ID	NUMERIC(10,0)	FK	sig_class
sig_priority	Priority	NUMERIC(10,0)		
sig_rev	Revision number	NUMERIC(10,)		
sig_sid	Internal signature ID	NUMERIC(10,0)		
sig_gid		NUMERIC(10,0)		

ตารางที่ 3.16 พจนานุกรมข้อมูลตาราง tcphdr

Attribute Name	Description	Type	Key	Reference Table
sid	Sensor ID	NUMERIC(10,0)	PK,FK	Event
cid	Event ID	NUMERIC(10,0)	PK	
tcp_sport	TCP source port	INT		
tcp_dport	TCP destination port	INT		

ตารางที่ 3.16 (ต่อ)

tcp_seq	TCP sequence number	NUMERIC(10,0)		
tcp_ack	TCP ACK number	NUMERIC(10,0)		
tcp_off	TCP offset	TINYINT		
tcp_res	TCP reserved	TINYINT		
tcp_flags	TCP flags	TINYINT		
tcp_win	TCP window	INT		
tcp_csum	TCP checksum	INT		
tcp_urp	TCP urgent pointer	INT		

ตารางที่ 3.17 พจนานุกรมข้อมูลตาราง udphdr

Attribute Name	**Description**	**Type**	**Key**	**Reference Table**
sid	Sensor ID	NUMERIC(10,0)	PK,FK	event
cid	Event ID	NUMERIC(10,0)	PK	
udp_sport	UDP source port	INT		
udp_dport	UDP destination port	INT		
udp_len	UDP length	INT		
udp_csum	UDP checksum	INT		

2) ส่วนของการสั่งงานไฟร์วอลล์

มีหน้าที่สั่งการไฟร์วอลล์ให้ทำการป้องกัน Packet จาก IP Address ของผู้บุกรุกและสามารถทำการยกเลิกการป้องกันได้แบบอัตโนมัติ เมื่อครบกำหนดเวลา โดยรูปแบบในการสั่งการไฟร์วอลล์ให้ทำการป้องกันและยกเลิกการป้องกันเป็นดังนี้

- คำสั่งในการเพิ่มกฎให้กับไฟร์วอลล์เพื่อทำการป้องกัน

  ipfw add Rule_number Action Protocol from IP_Address Port to IP_Address Port

  เช่น ipfw add 100 deny tcp from 172.16.23.41 1456 to 172.16.199.3 80

  ตัวอย่างข้างต้นเป็นการสั่งให้ไฟร์วอลล์ทำการป้องกัน Packet ที่มาจาก IP Address 172.16.23.41 ที่ทำการโจมตีมายังIP Address 172.16.199.3
- คำสั่งในการลบกฎในไฟร์วอลล์เพื่อยกเลิกการป้องกัน

  ipfw del Rule_number

  เช่น ipfw del 100

  ตัวอย่างข้างต้นเป็นคำสั่งในการลบกฎออกจากไฟร์วอลล์เมื่อครบกำหนดเวลาที่กำหนด

จากการออกแบบข้างต้นทำให้มีความเข้าใจในกระบวนการทำงานของระบบอย่างชัดเจน ซึ่งต่อจากนี้จะเป็นการนำสิ่งเหล่านี้ไปใช้ในการเขียนโปรแกรมเพื่อให้ระบบที่ทำการออกแบบไว้สามารถทำงานได้จริง ซึ่งการพัฒนาระบบจะอยู่ในบทต่อไป

บทที่ 4
การพัฒนาระบบป้องกันการบุกรุกเครือข่าย โดยใช้โปรแกรมSNORT

ในการพัฒนาระบบป้องกันการบุกรุกเครือข่ายโดยใช้โปรแกรม SNORT นั้นจะเป็นการพัฒนาบนระบบปฏิบัติการ FreeBSD ซึ่งเป็นแบบ Open source ไม่เสียค่าใช้จ่าย ส่วนภาษาที่ใช้ในการเขียนโปรแกรมนั้นคือ PHP ซึ่งเป็นแบบ Open source อีกเช่นกัน

4.1 เครื่องมือที่ใช้ในการพัฒนา

ได้เลือกใช้ระบบปฏิบัติการและซอฟต์แวร์ดังต่อไปนี้ สามารถอ่านวิธีการติดตั้งและขั้นตอนได้จากภาคผนวก (ก)

1. ระบบปฏิบัติการเลือกใช้ FreeBSD 7.0

เป็นระบบปฏิบัติการแบบยูนิกซ์ที่ได้รับความนิยมอย่างแพร่หลาย ไม่ต้องเสียค่าใช้จ่ายเนื่องจากเป็นแบบ Open source โดยระบบปฏิบัติการ FreeBSD นั้นเป็นระบบปฏิบัติการที่มีความปลอดภัยสูงกว่าระบบปฏิบัติการแบบ Open source ทั่วไป

2. ไฟร์วอลล์ IPFW

เป็นไฟร์วอลล์ที่มีอยู่บนระบบปฏิบัติการ FreeBSD มีหน้าที่ในการป้องกันการบุกรุกตามกฎที่มีการกำหนดค่าไว้ในไฟร์วอลล์

3. โปรแกรม SNORT เวอร์ชั่น 2.4

เป็นโปรแกรมตรวจจับกรบุกรุกบนเครือข่าย โดยเมื่อพบการบุกรุกก็จะทำการบันทึกการบุกรุกลงในฐานข้อมูล เพื่อให้โปรแกรมที่พัฒนาทำการอ่านค่าจากฐานข้อมูลและนำค่าที่ได้ไปสร้างกฎบนไฟร์วอลล์เพื่อทำการป้องกันการบุกรุกนั้น

4. โปรแกรมฐานข้อมูล เลือกใช้ MySQL เวอร์ชั่น

MySQL เป็นโปรแกรมฐานข้อมูลแบบ Open source ที่นิยมใช้กันอย่างแพร่หลาย เนื่องจากตัวโปรแกรมเป็นโปรแกรมที่ไม่ต้องเสียค่าใช้จ่ายและสามารถทำงานได้อย่างมีประสิทธิภาพ

5. เว็บเซิร์ฟเวอร์เลือกใช้ Apache ซึ่งรองรับการทำงานของภาษา PHP

Apache เป็นซอร์ฟแวร์ให้บริการเว็บเซิร์ฟเวอร์ เพื่อติดต่อกับผู้ใช้งานผ่านโปรโตคอล HTTP ซึ่งในกรณีใช้งานแบบมีการเข้ารหัสข้อมูล จะใช้งานผ่านโปรโตคอล HTTPS

6. ซอฟต์แวร์ภาษาเลือกใช้ PHP

PHP เป็น Scripts language ที่ทำงานร่วมกับ HTML โดยสามารถเขียนแทรกเข้าไปใน HTML โดย PHP เป็น Open source ที่สามารถใช้งานได้กับหลายๆระบบปฏิบัติการ

7. โปรแกรม Macromedia Dreamweaver เวอร์ชั่น 8

เป็นโปรแกรมสำหรับการพัฒนา ซึ่งสนับสนุนหลายภาษาได้แก่ HTML, PHP, ASP, C/C++, VBScript, JavaScript, XML เป็นต้น ซึ่งโปรแกรมที่จะทำการพัฒนานั้นจะใช้ภาษา PHP ในการพัฒนา

8. โปรแกรมที่ใช้สร้างการบุกรุกบนเครือข่าย เช่น โปรแกรม Radware Security

เนื่องจากในการพัฒนาจำเป็นจะต้องมีการบุกรุกเกิดขึ้น เพื่อที่จะเขียนโปรแกรมได้อย่างถูกต้องและสามารถตรวจสอบได้ จึงใช้โปรแกรมเหล่านี้ในการสร้าง Packet ที่มีรูปแบบการบุกรุกเช่น Ping Flood, Scan port, TCP attack, UDP attack เป็นต้น โดยโปรแกรม SNORT เมื่อตรวจพบการบุกรุกจะทำการบันทึกข้อมูลเหล่านี้ลงในฐานข้อมูล

4.2 การพัฒนาระบบป้องกันการบุกรุกเครือข่ายโดยใช้โปรแกรม SNORT

การพัฒนาระบบป้องกันการบุกรุกเครือข่ายโดยใช้โปรแกรม SNORT ทำการพัฒนาโดยใช้โปรแกรมภาษา PHP โดยโปรแกรมที่พัฒนาขึ้นจะมีความสามารถดังนี้

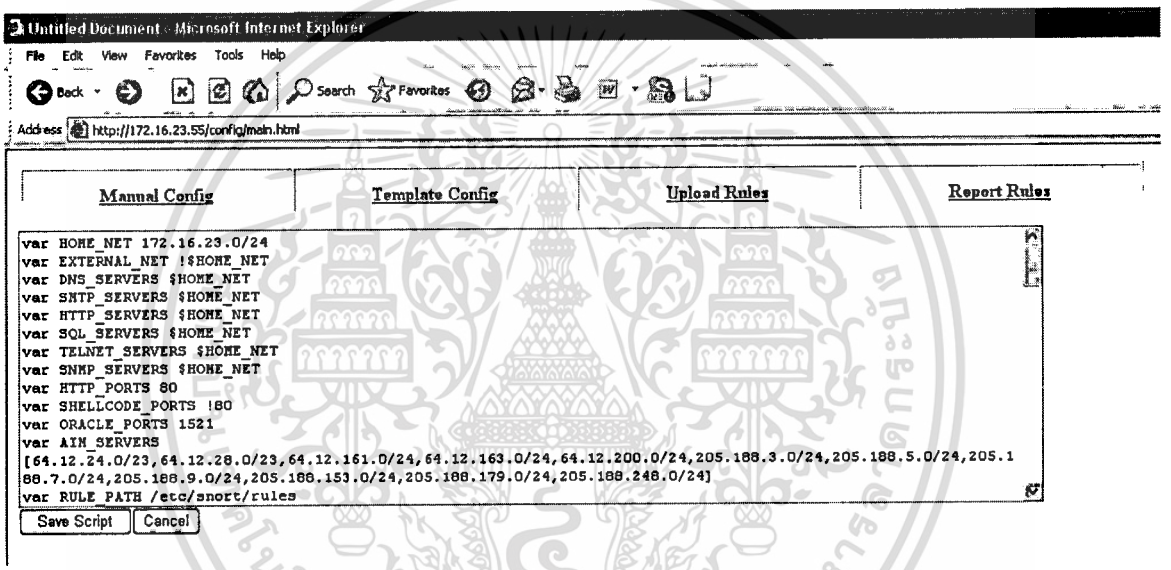
1. สามารถดึงข้อมูลจากฐานข้อมูลการบุกรุกเมื่อโปรแกรมSNORT ตรวจพบการบุกรุกได้ โดยข้อมูลที่ต้องทำการดึงจากฐานข้อมูลคือ IP Address, Protocol, port เพื่อใช้เป็นข้อมูลในการสร้างกฎที่ไฟร์วอลล์
2. สามารถนำข้อมูลการบุกรุกที่ได้จากฐานข้อมูลไปสร้างกฎที่ไฟร์วอลล์ IPFW ได้
3. สามารถลบกฎที่ครบกำหนดเวลาแล้วออกจากไฟร์วอลล์ IPFW ได้
4. สามารถแก้ไขไฟล์ Snort.conf ซึ่งเป็นไฟล์ที่มีไว้ในการปรับแต่งค่าการตรวจจับการบุกของโปรแกรม SNORT ผ่านทางหน้า Web page ได้และ Upload กฎของ SNORT ได้
5. สามารถแสดงรายงานสถานะของกฎต่างๆว่ากฎใดที่ถูกใช้งานอยู่และกฎใดที่ไม่ได้ใช้แล้วที่ถูกสร้างขึ้นบนไฟร์วอลล์ IPFW ผ่านทาง Web page ได้

โดยมีรูปแบบการทำงานดังนี้

4.2.1 ไฟล์ SNORT

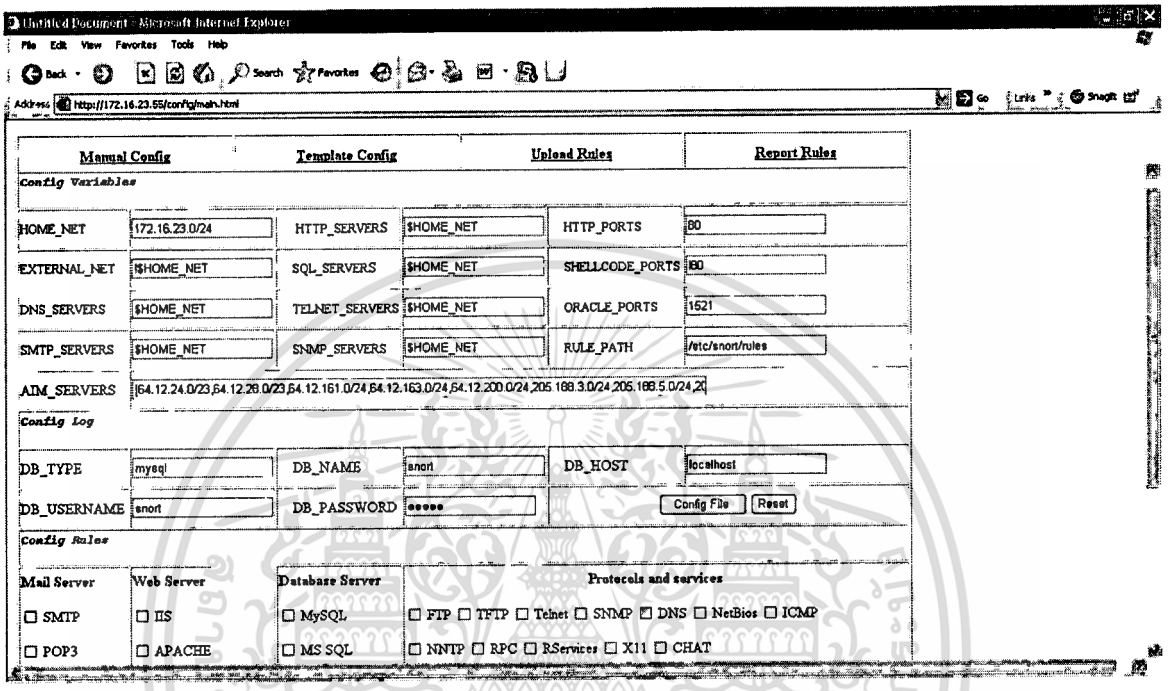
เนื่องจากก่อนการใช้งานระบบต้องมีการกำหนดค่าพื้นฐานของไฟล์ Snort.conf ให้มีความเหมาะสมกับระบบที่ใช้งาน ดังนั้นเพื่อความสะดวกจึงทำการออกแบบให้สามารถทำการกำหนดค่าพื้นฐานของโปรแกรม SNORT ได้ผ่านทาง Web browser ได้ 2 วิธีดังนี้

- สามารถปรับแก้ไฟล์ Snort.conf ผ่านทาง Web browser โดยรูปแบบการแก้ไขคือเป็นลักษณะ text editor สามารถกำหนดค่าให้กับโปรแกรม SNORT และทำการใส่ค่าที่ต้องการลงไปและทำการบันทึกค่าลงในไฟล์ Snort.conf ผ่านทาง Web browser ได้



รูปที่ 4.1 แสดงการกำหนดค่าไฟล์ snort แบบ text editor

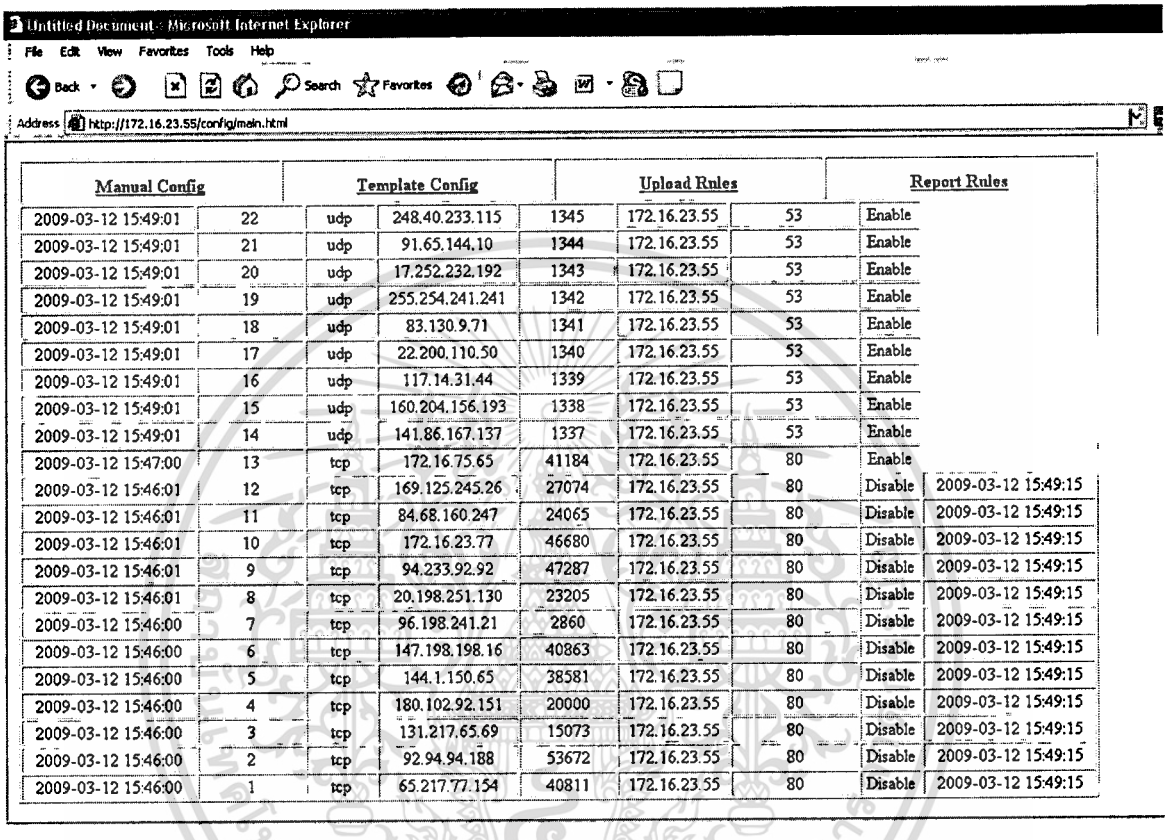
- สามารถปรับแก้ไฟล์ Snort.conf ผ่านทาง Web browser โดยรูปแบบการแก้ไขจะเป็นลักษณะ text box และ Check box โดยสามารถเลือกและกำหนดค่าได้ง่ายผ่านทาง menu ที่กำหนดไว้ เมื่อทำการกำหนดรูปแบบที่ต้องการเรียบร้อยก็ ทำการบันทึกข้อมูลลงในไฟล์ Snort.conf ผ่านทาง Web browser ได้



รูปที่ 4.2 แสดงการกำหนดค่าไฟล์ snort แบบ Wizard

4.2.2 ไฟล์ Report

เนื่องจากระบบที่พัฒนาขึ้นจะต้องสามารถแสดงรายงานสถานะของกฎที่ได้ถูกใช้งานอยู่หรือใช้งานไปแล้วที่ไฟร์วอลล์ได้ ดังนั้นเพื่อความสะดวกในการใช้งานและการจัดการของผู้ดูแลระบบ จึงทำการออกแบบให้สามารถดูรายงานสถานะผ่านทาง Web browser ได้



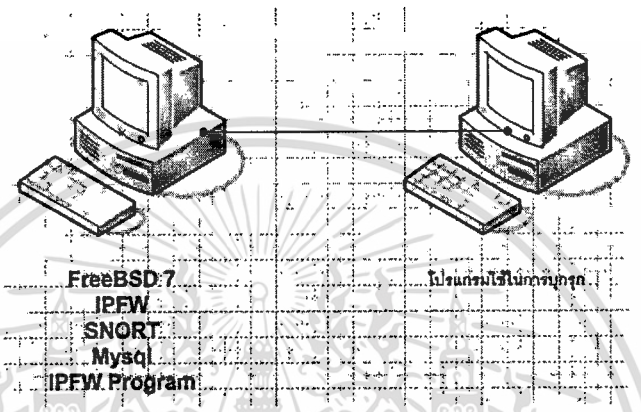
Manual Config		Template Config			Upload Rules		Report Rules	
2009-03-12 15:49:01	22	udp	248.40.233.115	1345	172.16.23.55	53	Enable	
2009-03-12 15:49:01	21	udp	91.65.144.10	1344	172.16.23.55	53	Enable	
2009-03-12 15:49:01	20	udp	17.252.232.192	1343	172.16.23.55	53	Enable	
2009-03-12 15:49:01	19	udp	255.254.241.241	1342	172.16.23.55	53	Enable	
2009-03-12 15:49:01	18	udp	83.130.9.71	1341	172.16.23.55	53	Enable	
2009-03-12 15:49:01	17	udp	22.200.110.50	1340	172.16.23.55	53	Enable	
2009-03-12 15:49:01	16	udp	117.14.31.44	1339	172.16.23.55	53	Enable	
2009-03-12 15:49:01	15	udp	160.204.156.193	1338	172.16.23.55	53	Enable	
2009-03-12 15:49:01	14	udp	141.86.167.137	1337	172.16.23.55	53	Enable	
2009-03-12 15:47:00	13	tcp	172.16.75.65	41184	172.16.23.55	80	Enable	
2009-03-12 15:46:01	12	tcp	169.125.245.26	27074	172.16.23.55	80	Disable	2009-03-12 15:49:15
2009-03-12 15:46:01	11	tcp	84.68.160.247	24065	172.16.23.55	80	Disable	2009-03-12 15:49:15
2009-03-12 15:46:01	10	tcp	172.16.23.77	46680	172.16.23.55	80	Disable	2009-03-12 15:49:15
2009-03-12 15:46:01	9	tcp	94.233.92.92	47287	172.16.23.55	80	Disable	2009-03-12 15:49:15
2009-03-12 15:46:01	8	tcp	20.198.251.130	23205	172.16.23.55	80	Disable	2009-03-12 15:49:15
2009-03-12 15:46:00	7	tcp	96.198.241.21	2860	172.16.23.55	80	Disable	2009-03-12 15:49:15
2009-03-12 15:46:00	6	tcp	147.198.198.16	40863	172.16.23.55	80	Disable	2009-03-12 15:49:15
2009-03-12 15:46:00	5	tcp	144.1.150.65	38581	172.16.23.55	80	Disable	2009-03-12 15:49:15
2009-03-12 15:46:00	4	tcp	180.102.92.151	20000	172.16.23.55	80	Disable	2009-03-12 15:49:15
2009-03-12 15:46:00	3	tcp	131.217.65.69	15073	172.16.23.55	80	Disable	2009-03-12 15:49:15
2009-03-12 15:46:00	2	tcp	92.94.94.188	53672	172.16.23.55	80	Disable	2009-03-12 15:49:15
2009-03-12 15:46:00	1	tcp	65.217.77.154	40811	172.16.23.55	80	Disable	2009-03-12 15:49:15

รูปที่ 4.3 แสดงรายงานสถานะของกฎที่อยู่บนไฟร์วอลล์

4.3 การทดสอบการทำงานของระบบ

การทำสอบการทำงานของระบบจะทำการทดสอบโดยต้องใช้เครื่อง 2 เครื่อง คือ

- เครื่องหลักติดตั้งระบบปฏิบัติการ FreeBSD, ไฟร์วอลล์ IPFW, โปรแกรม SNORT, โปรแกรมที่พัฒนาขึ้นเอง
- เครื่องสร้างการบุกรุกโดยทำการติดตั้งโปรแกรมสร้างการบุกรุก (Radware Security Demonstration Toolkit) เป็น Tools ที่ใช้ในการจำลองการโจมตีเพื่อทดสอบระบบ



รูปที่ 4.4 รูปแบบการทดสอบ

4.3.1 ขั้นตอนการทดสอบ

สามารถทดสอบการทำงานของระบบดังนี้

1. ทำการกำหนดค่าพื้นฐานของโปรแกรม SNORT โดยระบุค่าผ่านทาง Web browser
2. ทำการรันโปรแกรมสร้างการบุกรุกเพื่อให้ตัวโปรแกรมที่พัฒนาตรวจจับการบุกรุกและทำการดึงข้อมูลการบุกรุกไปใช้งานและทำการสร้างกฎเพื่อป้องกันการบุกรุกนั้น
3. เมื่อสร้างการโจมตีเข้าไปในระบบแล้วสามารถเข้าไปดูสถานะของกฎที่มีอยู่บนไฟร์วอลล์ IPFW ได้โดยดูผ่านทาง Web browser ได้
4. จากนั้นเมื่อครบกำหนดเวลาที่ตั้งไว้ กฎบนไฟร์วอลล์ IPFW จะถูกลบออกจากระบบโดยอัตโนมัติ โดยสามารถดูสถานะของกฎในไฟร์วอลล์ผ่านทาง Web browser ได้

4.3.2 รายละเอียดการทดสอบและผลการโจมตี

ลำดับ	รูปแบบการโจมตี	ผลการโจมตี
1.	IPS Attack Replay	
	- Worms	ตรวจจับไม่พบ
	- Trojans / Backdoors	ตรวจจับไม่พบ
	- Buffer – Overflows	ตรวจจับพบ
	- Mail (SMTP / POP3 / IMAP)	ตรวจจับไม่พบ
	- FTP	ตรวจจับพบ
	- Web (Apache / IIS / CGI)	ตรวจจับพบ
	- NetBIOS / MS-RPC	ตรวจจับไม่พบ
	- SIP Attacks	ตรวจจับไม่พบ
2.	DOS Attacks	
	- SYN Flood	ตรวจจับพบ
	- TCP Flood	ตรวจจับพบ
	- UDP Flood	ตรวจจับพบ
3.	DOS WWW / HTTP	ตรวจจับพบ
4.	IP Fragmentation	ตรวจจับพบ

ตารางที่ 4.1 แสดงรายละเอียดการทดสอบและผลการโจมตี

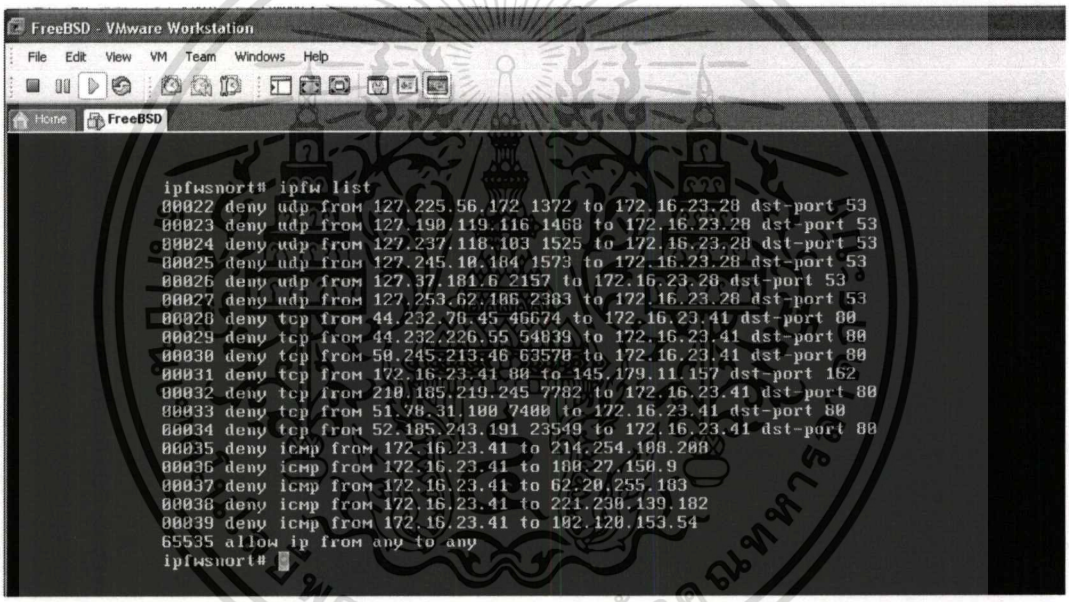
จากการทดสอบการโจมตีข้างต้นทั้งหมด 13 รูปแบบ พบว่าระบบสามารถตรวจจับการบุกรุกได้ 8 รูปแบบ และโปรแกรมที่พัฒนาขึ้นสามารถนำข้อมูลการบุกรุกที่ตรวจจับได้ไปทำการเพิ่มกฎป้องกันการบุกรุกที่ไฟร์วอลล์ได้อย่างอัตโนมัติ ส่วนสาเหตุที่บางรูปแบบการโจมตีนั้นไม่สามารถตรวจจับได้ เนื่องจากในฐานข้อมูลการบุกรุกอาจจะไม่มีรูปแบบการบุกรุกนั้น จึงทำให้ไม่สามารถตรวจจับการบุกรุกนั้นได้ ดังนั้นจึงควรอัพเดตฐานข้อมูลการบุกรุกให้เป็นปัจจุบันที่สุด เพื่อให้สามารถป้องกันการบุกรุกได้อย่างมีประสิทธิภาพสูงสุด

4.3.3 ผลการทดสอบระบบ

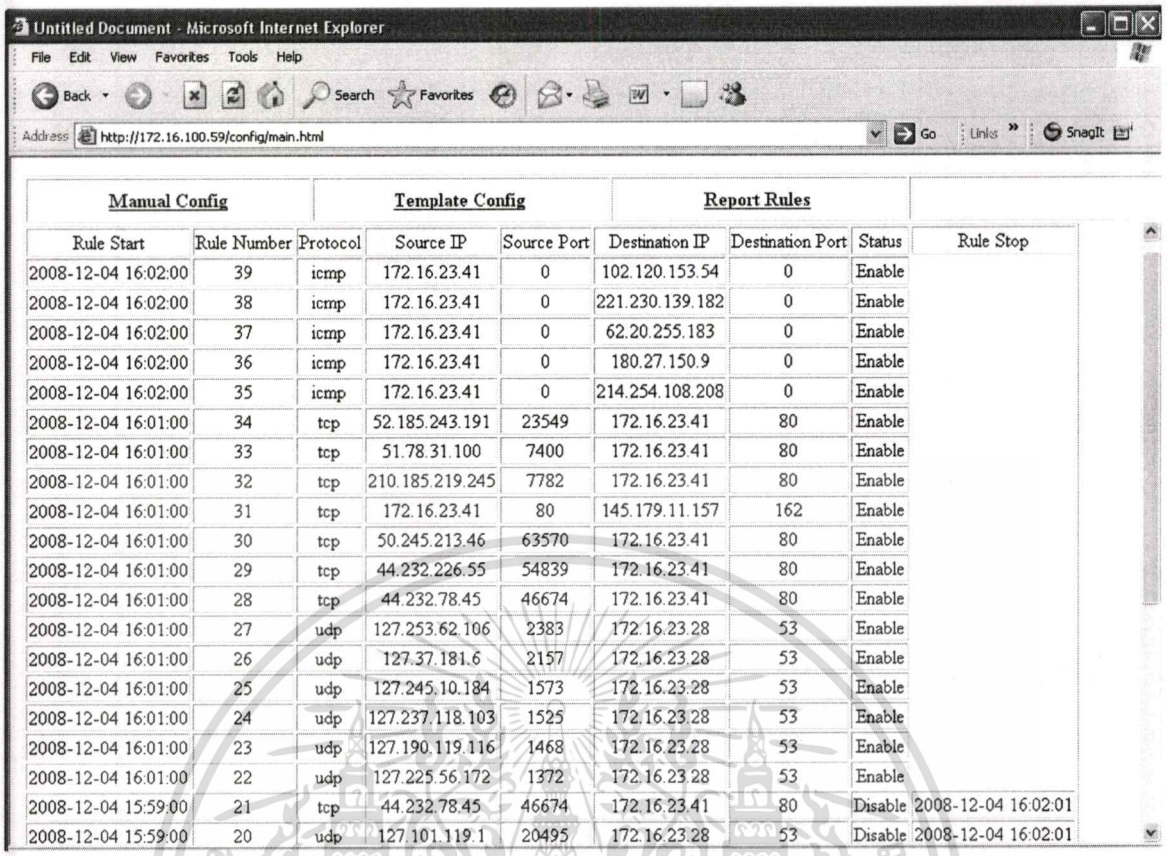
พบว่าโปรแกรมที่พัฒนาขึ้นสามารถใช้งานได้ตามต้องการคือ

1. โปรแกรมที่พัฒนาขึ้นสามารถกำหนดค่าพื้นฐานของโปรแกรม SNORT ผ่านทาง Web browser ได้
2. โปรแกรมที่พัฒนาขึ้นสามารถดึงข้อมูลจากฐานข้อมูลการบุกรุก โดยนำค่าที่ได้ไปสร้างกฎป้องกันการบุกรุกที่ไฟร์วอลล์ IPFW ได้
3. โปรแกรมที่พัฒนาขึ้นสามารถลบกฎที่ครบกำหนดแล้วออกจากไฟร์วอลล์ IPFW ได้
4. โปรแกรมที่พัฒนาขึ้นสามารถแสดงรายงานเกี่ยวกับสถานะและรายละเอียดต่างๆของกฎที่สร้างขึ้นบนไฟร์วอลล์ IPFW ผ่านทาง Web browser ได้

ดังรูปที่ 4.5



รูปที่ 4.5 แสดงกฎที่สร้างขึ้นบนไฟร์วอลล์ IPFW



Manual Config | Template Config | Report Rules

Rule Start	Rule Number	Protocol	Source IP	Source Port	Destination IP	Destination Port	Status	Rule Stop
2008-12-04 16:02:00	39	icmp	172.16.23.41	0	102.120.153.54	0	Enable	
2008-12-04 16:02:00	38	icmp	172.16.23.41	0	221.230.139.182	0	Enable	
2008-12-04 16:02:00	37	icmp	172.16.23.41	0	62.20.255.183	0	Enable	
2008-12-04 16:02:00	36	icmp	172.16.23.41	0	180.27.150.9	0	Enable	
2008-12-04 16:02:00	35	icmp	172.16.23.41	0	214.254.108.208	0	Enable	
2008-12-04 16:01:00	34	tcp	52.185.243.191	23549	172.16.23.41	80	Enable	
2008-12-04 16:01:00	33	tcp	51.78.31.100	7400	172.16.23.41	80	Enable	
2008-12-04 16:01:00	32	tcp	210.185.219.245	7782	172.16.23.41	80	Enable	
2008-12-04 16:01:00	31	tcp	172.16.23.41	80	145.179.11.157	162	Enable	
2008-12-04 16:01:00	30	tcp	50.245.213.46	63570	172.16.23.41	80	Enable	
2008-12-04 16:01:00	29	tcp	44.232.226.55	54839	172.16.23.41	80	Enable	
2008-12-04 16:01:00	28	tcp	44.232.78.45	46674	172.16.23.41	80	Enable	
2008-12-04 16:01:00	27	udp	127.253.62.106	2383	172.16.23.28	53	Enable	
2008-12-04 16:01:00	26	udp	127.37.181.6	2157	172.16.23.28	53	Enable	
2008-12-04 16:01:00	25	udp	127.245.10.184	1573	172.16.23.28	53	Enable	
2008-12-04 16:01:00	24	udp	127.237.118.103	1525	172.16.23.28	53	Enable	
2008-12-04 16:01:00	23	udp	127.190.119.116	1468	172.16.23.28	53	Enable	
2008-12-04 16:01:00	22	udp	127.225.56.172	1372	172.16.23.28	53	Enable	
2008-12-04 15:59:00	21	tcp	44.232.78.45	46674	172.16.23.41	80	Disable	2008-12-04 16:02:01
2008-12-04 15:59:00	20	udp	127.101.119.1	20495	172.16.23.28	53	Disable	2008-12-04 16:02:01

รูปที่ 4.6 แสดงรายละเอียดสถานะของกฎที่สร้างขึ้นบนไฟร์วอลล์ผ่านทาง Web browser

สรุปผลการทดสอบเป็นไปตามความต้องการที่ได้ทำการออกแบบไว้ โดยสามารถให้โปรแกรมที่พัฒนาขึ้นเป็นตัวกลางในการเชื่อมโยงระหว่างโปรแกรมตรวจจับการบุกรุกรุก SNORT และ ไฟร์วอลล์ IPFW โดยโปรแกรมที่พัฒนาขึ้นสามารถดึงข้อมูลการบุกรุกที่โปรแกรม SNORT ตรวจจับได้จากฐานข้อมูลการบุกรุก จากนั้นนำข้อมูลที่ได้ไปสร้างกฎป้องกันการบุกรุกที่ไฟร์วอลล์ IPFW และสามารถลบกฎที่ครบกำหนดเวลาออกจากไฟร์วอลล์ได้โดยอัตโนมัติและมีเครื่องมือช่วยในการปรับแต่งค่าพื้นฐานของโปรแกรม SNORT และสามารถมอนิเตอร์สถานะของกฎที่ใช้งานผ่านทาง Web page ได้

บทที่ 5
บทสรุปและแนวทางในการพัฒนาในอนาคต

5.1 สรุปผลการพัฒนาระบบ

ผลที่ได้จากการพัฒนาระบบป้องกันการบุกรุกเครือข่ายโดยใช้โปรแกรม SNORT นั้น สามารถพัฒนาโปรแกรมให้สามารถเชื่อมโยงการทำงานระหว่างโปรแกรมตรวจจับการบุกรุก SNORT และไฟร์วอลล์ IPFW ให้สามารถทำงานร่วมกันได้อย่างสอดคล้องตามวัตถุประสงค์ การพัฒนาคือ โปรแกรมที่เราพัฒนาขึ้นจะคอยทำการอ่านข้อมูลการบุกรุกและนำข้อมูลที่ได้มาสร้างกฎที่ไฟร์วอลล์เพื่อทำการป้องกันการบุกรุกโดยเมื่อครบกำหนดเวลาสามารถลบกฎที่ครบกำหนดเวลาแล้วออกจากไฟร์วอลล์ได้อีกทั้งมีเครื่องมือช่วยในการปรับแต่งค่าพื้นฐานของโปรแกรม SNORT เพื่อให้ผู้ดูแลระบบสามารถปรับแต่งค่าได้ง่ายขึ้นและสามารถมอนิเตอร์สถานะได้อย่างสะดวก

5.2 ประโยชน์ที่ได้รับ

โปรแกรมที่พัฒนาขึ้นสามารถทำการป้องกันการบุกรุกได้อย่างอัตโนมัติ โดยเมื่อมีการตรวจพบการบุกรุกเข้ามาในเครือข่าย โปรแกรมที่พัฒนาขึ้นสามารถสั่งให้ไฟร์วอลล์ทำการป้องกันได้ในทันที ซึ่งรวดเร็วกว่าการให้ผู้ดูแลระบบมาทำการปรับแต่งค่าที่ไฟร์วอลล์ด้วยตนเองเนื่องจากมีความสะดวกรวดเร็วและลดความเสียหายที่เกิดขึ้นจากการโจมตีได้

5.3 ข้อจำกัดของระบบ

จากการที่โปรแกรมที่พัฒนาขึ้นต้องทำการนำข้อมูลจากโปรแกรมตรวจจับการบุกรุกมาทำการประมวลผลต่อเพื่อนำมาสร้างกฎป้องกันการบุกรุกที่ไฟร์วอลล์จึงทำให้เกิดข้อจำกัดของโปรแกรมที่พัฒนาขึ้น เนื่องจากถ้ามีการบุกรุกเกิดขึ้นในระบบแต่ตัวโปรแกรม SNORT ตรวจไม่พบโปรแกรมที่เราพัฒนาขึ้นก็จะไม่ทำงาน หรือในกรณีที่โปรแกรม SNORT ตรวจจับการบุกรุกผิดพลาดก็จะทำให้โปรแกรมที่พัฒนาขึ้นทำงานผิดพลาดไปด้วยจึงเป็นข้อจำกัดของระบบในการทำงาน

5.4 แนวทางในการพัฒนาต่อ

เนื่องจากโปรแกรมที่พัฒนาขึ้นนี้เป็นลักษณะโปรแกรมที่เป็นตัวกลางในการเชื่อมโยงกันระหว่างสองโปรแกรมให้สามารถทำงานร่วมกันได้คือระหว่างโปรแกรมตรวจจับการบุกรุก SNORT

และไฟร์วอลล์ IPFW แต่เนื่องจากในปัจจุบันมีไฟร์วอลล์อื่นๆจึงควรทำการพัฒนาโปรแกรมเพิ่มเติมให้สามารถรองรับการใช้งานกับไฟร์วอลล์อื่นๆได้หรือนำไปประยุกต์ บนระบบปฏิบัติการอื่นๆด้วย เพื่อให้สามารถพัฒนาระบบให้สามารถรองรับได้กับทุกระบบปฏิบัติการหรือไฟร์วอลล์หลายๆแบบ และเนื่องจากโปรแกรมที่พัฒนามีข้อจำกัดคือ ทำงานทุก 1 นาที ซึ่งอาจจะทำให้ไม่สามารถป้องกันการบุกรุกได้ดีเท่าที่ควร ดังนั้นอาจจะพัฒนาต่อโดยให้สามารถทำงานได้เร็วกว่านี้ ซึ่งจะทำให้ระบบมีประสิทธิภาพยิ่งขึ้น

บรรณานุกรม

Arie D.Jones and Ron Plew. 2006 **Sams Teach Yourself SQL in 24 Hours**. Sams Publishing

Brian Tiemann. 2003. **Sams Teach Yourself FreeBSD in 24 Hours**. Sams Publishing

Charlie Scott and Bert Hayes, Paul Wolfe. 2004. **Snort for Dummies.** Wiley Publishing

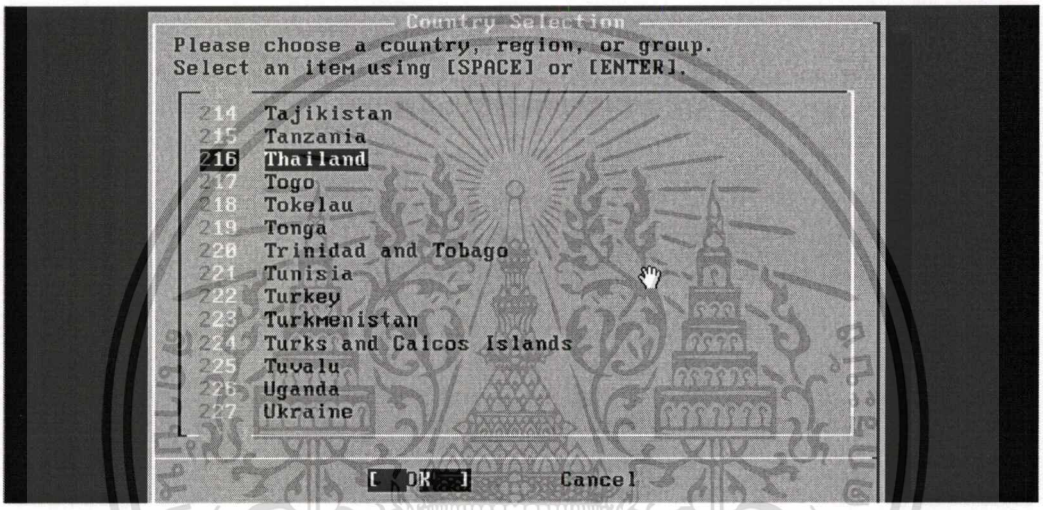
Jay Beale and James C. Foster. 2003. **Snort 2.0 Intrusion Detection.** Syngress Publishing

Stephen Northcutt and Toby Kohlenberg. 2007. **Snort IDS and IPS Toolkit.** Syngress Publishing

ภาคผนวก ก
คู่มือการติดตั้งระบบ

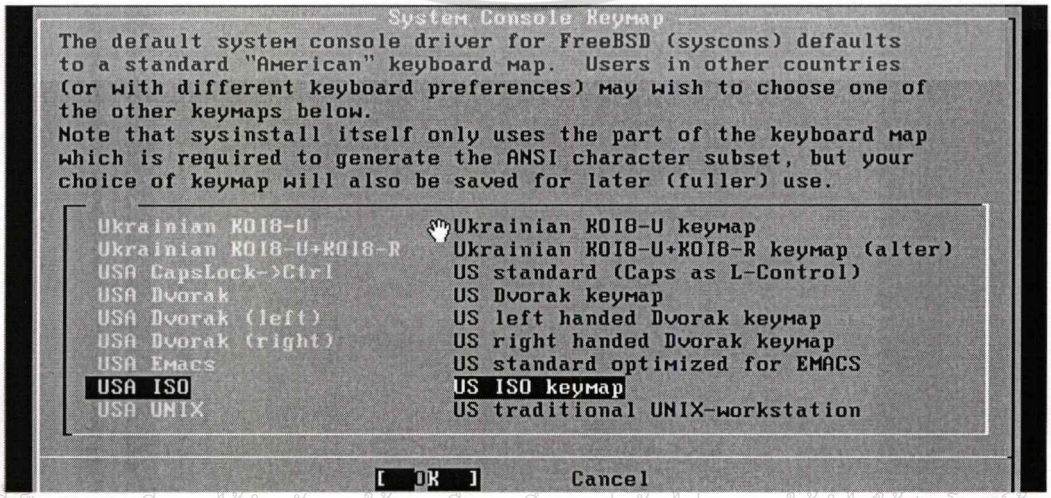
<u>การติดตั้งระบบปฏิบัติการฟรีบีเอสดี</u>

1. เริ่มต้นด้วยการบูทเครื่องเซิรฟ์เวอร์จากแผ่น Installation Disc รอจนปรากฏหน้าจอภาพให้เลือกประเทศ (ลำดับที่ 216 Thailand) แล้วกดปุ่ม Enter เพื่อทำงานต่อ ดังรูปที่ 1



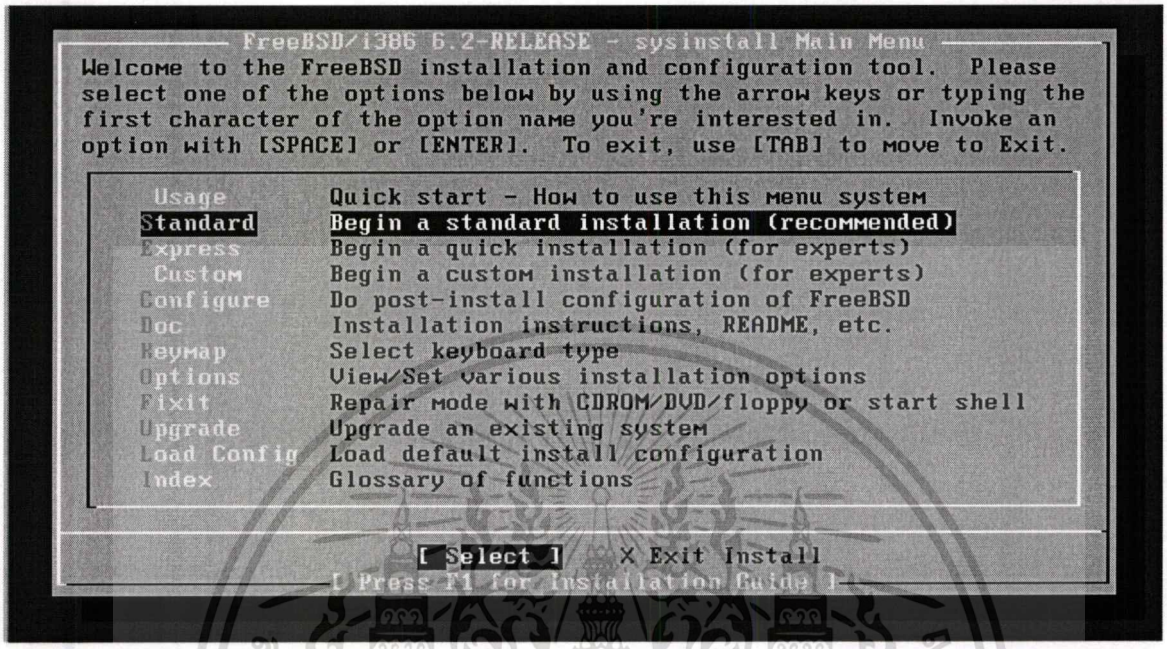
รูปที่ 1 เลือกประเทศ ลำดับที่ 216 Thailand

2. เลือกคีย์บอร์ดเพื่อใช้งาน ในที่นี้จะเลือกเป็น USA ISO ซึ่งถือว่าเป็น Keyboard มาตรฐานที่เราสามารถพิมพ์สั่งงานระบบได้ ดังรูปที่ 2



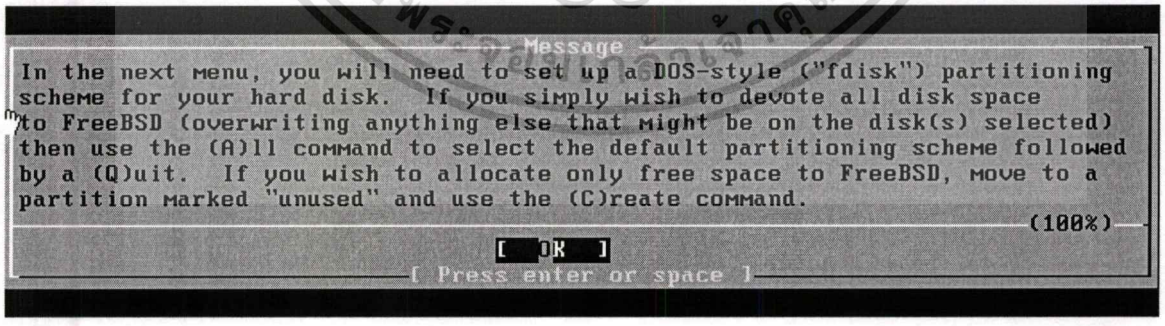
รูปที่ 2 เลือกคีย์บอร์ดที่ใช้งาน

3. เลือก Standard และกดปุ่ม Enter เพื่อเริ่มการติดตั้งระบบปฏิบัติการ FreeBSD ดังรูปที่ 3



รูปที่ 3 เลือก Standard Installation

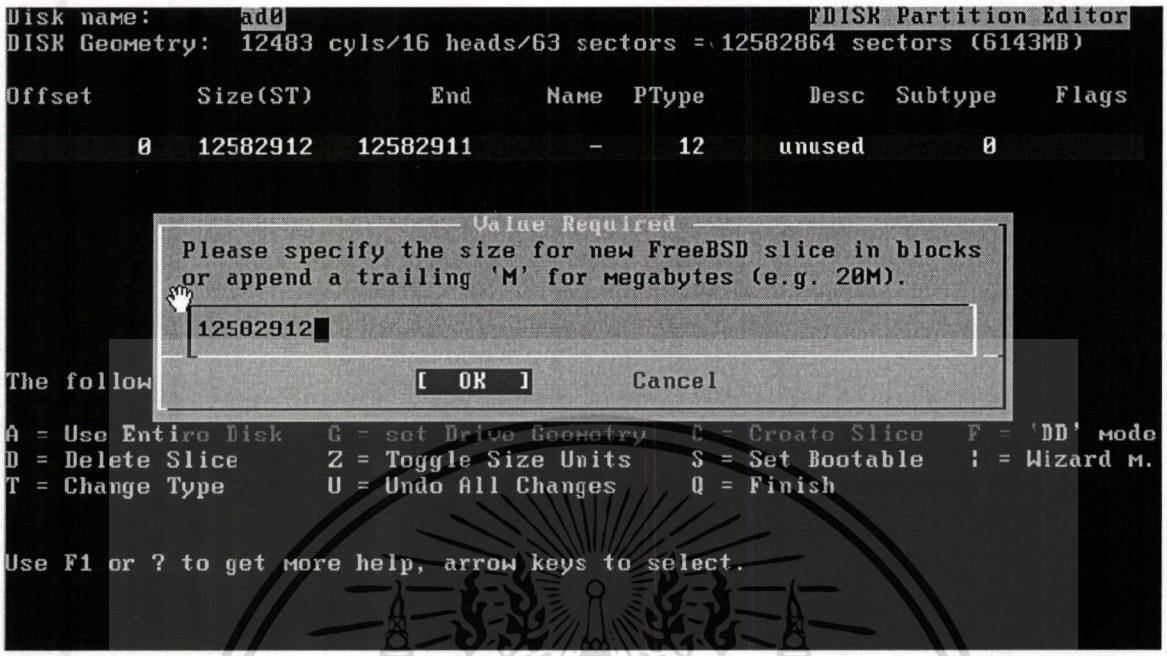
4. โปรแกรมติดตั้งระบบปฏิบัติการ FreeBSD จะแจ้งให้เราทราบว่าจะจัดการกับโครงสร้างของ Partition ของ Hard disk ของเรา เมื่ออ่านคำแนะนำเสร็จแล้วก็สามารถกดปุ่ม Enter เพื่อเริ่มการจัดการกับ Partition ดังรูปที่ 4



รูปที่ 4 แนะนำเกี่ยวกับการจัดการ Partition

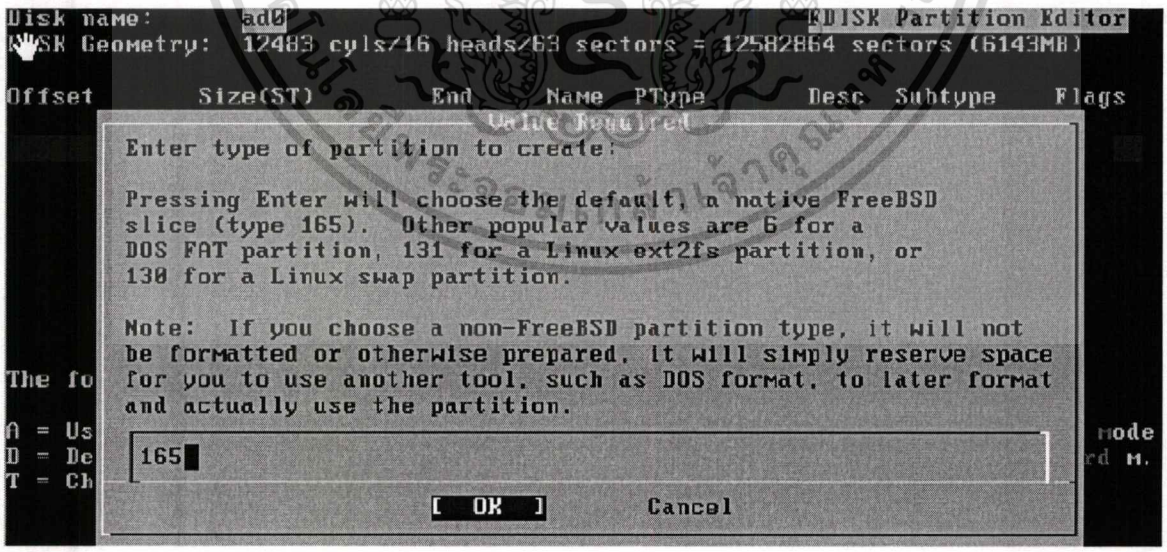
5. เราสามารถสร้าง Partition ได้โดยการกดปุ่มตัวอักษร C หรือมีความหมายเป็น Create เพื่อสร้าง Partition ใหม่ จากนั้นระบบจะแสดงขนาดของ Block Size ออกมาให้เราทราบ ซึ่งก็ขอให้ยอมรับว่า

Block Size ทั้งหมด เนื่องจากเราต้องการสร้าง Partition ใหม่นี้เป็นโครงสร้างที่ใช้กับระบบปฏิบัติการ FreeBSD ให้เรากดปุ่ม Enter ดังรูปที่ 5



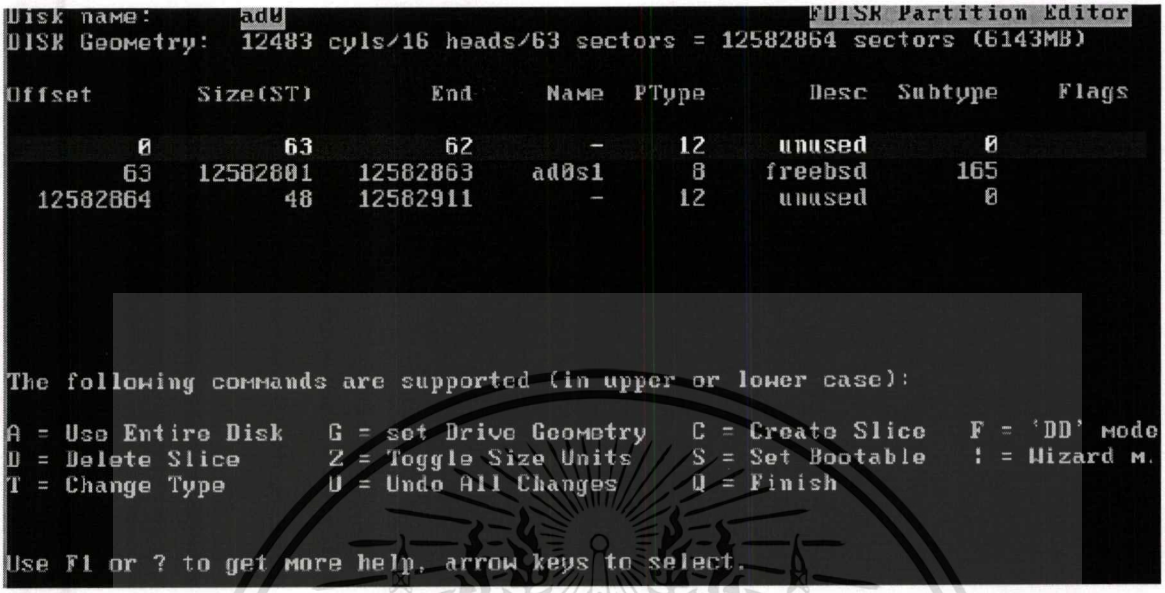
รูปที่ 5 เลือกขนาดของ Partition ทั้งหมด

6. ให้กำหนดค่าชนิดของ Partition ให้กำหนดเป็น 165 ซึ่งเป็นมาตรฐานที่ใช้ในระบบปฏิบัติการ FreeBSD ให้กดปุ่ม Enter ดังรูปที่ 6



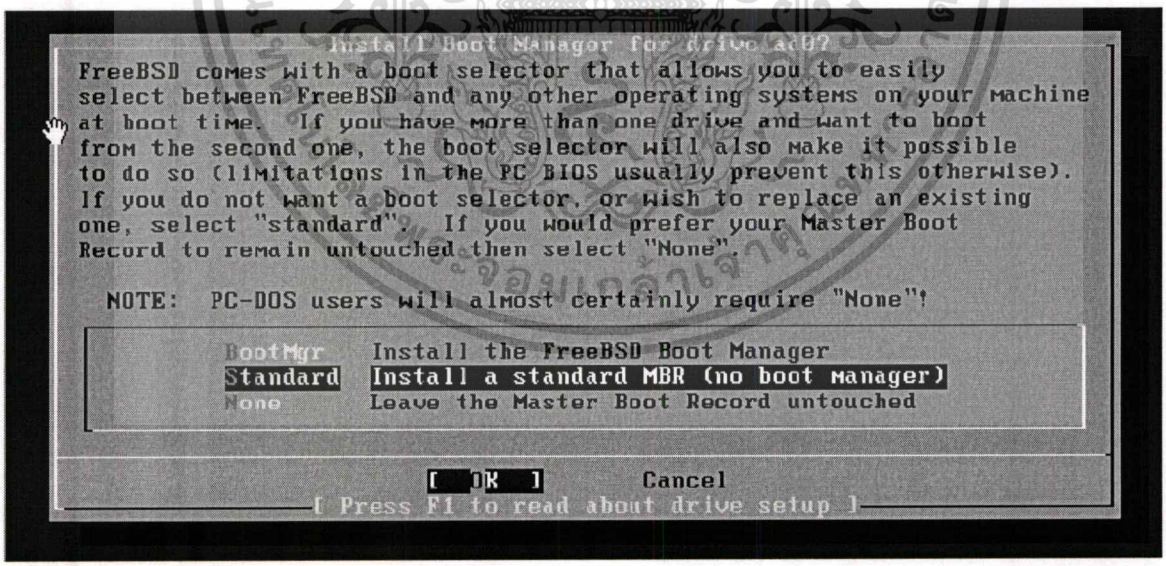
รูปที่ 6 เลือก Partition Type

7. โปรแกรมจะแสดงโครงสร้างของ Hard Disk ที่เราได้กำหนดและแบ่ง Partition ให้เราทราบ ซึ่งดูรายละเอียดต่างๆ เรียบร้อยแล้วให้กดปุ่ม Q ดังรูปที่ 7



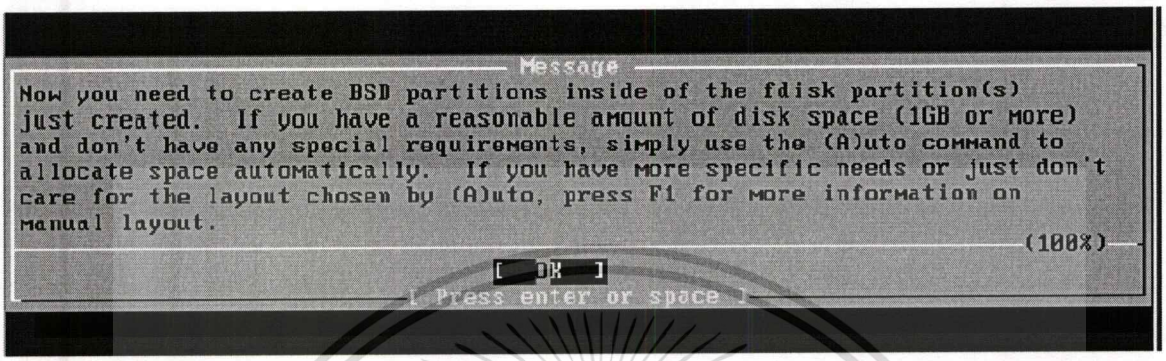
รูปที่ 7 แสดงโครงสร้างของ Partition

8. เลือก Boot Manage ของ Hard Disk (ad0) เป็น Standard ดังรูปที่ 8



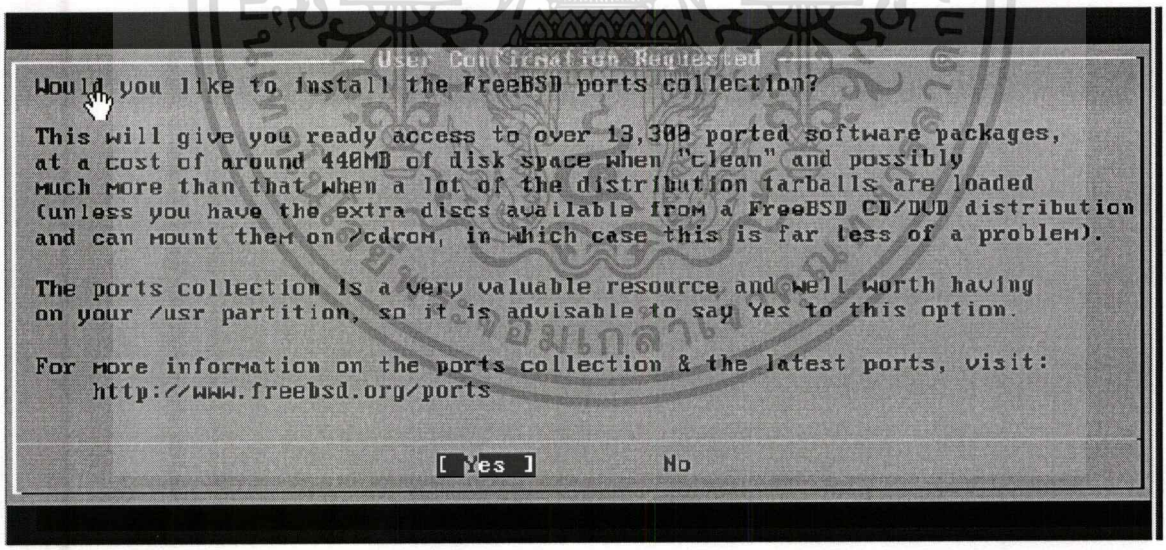
รูปที่ 8 เลือก Boot Manager

9. โปรแกรมติดตั้งจะแสดงคำแนะนำเพื่อบอกให้เราทราบเกี่ยวกับการแบ่ง Partition ออกเป็นส่วนต่างๆ โปรแกรมก็ได้แนะนำให้เราแบ่ง Partition ง่าย ๆ โดยการกดปุ่ม A หรือ Automatic ก็จะเป็นอันเสร็จกระบวนตรงนี้ หากดำเนินการเกี่ยวกับ Partition เรียบร้อยแล้วก็สามารถกดปุ่ม Q เพื่อเสร็จสิ้นการแบ่ง Partition ในส่วนนี้ ดังรูปที่ 9



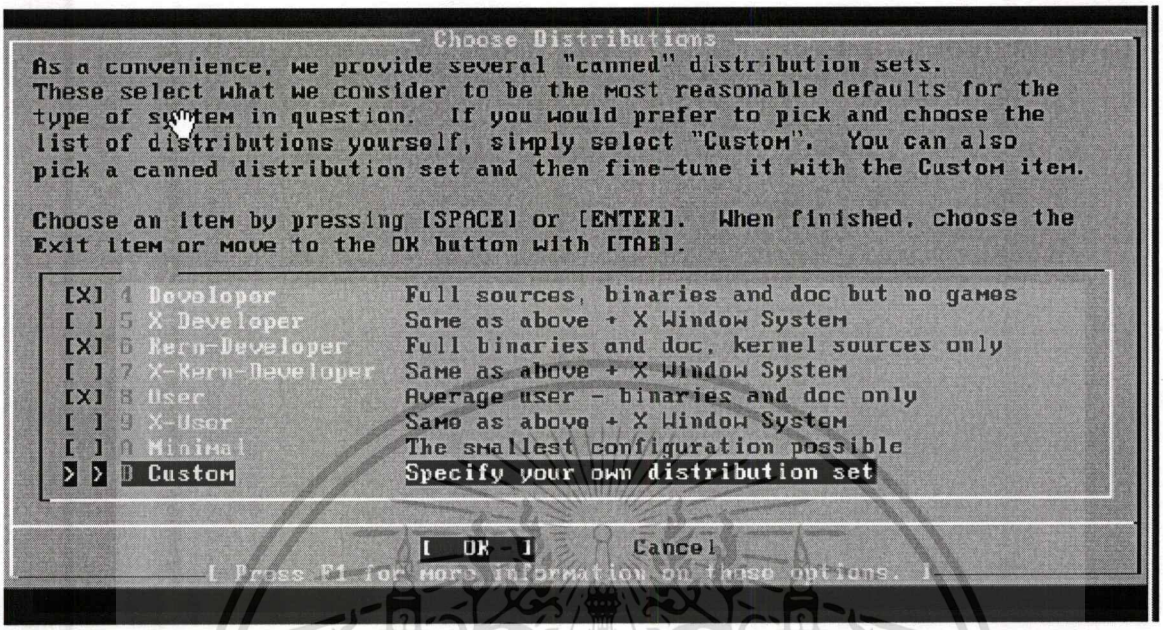
รูปที่ 9 แสดงคำแนะนำในการแบ่ง Partition

10. โปรแกรมติดตั้งจะแสดงรายละเอียดให้เราทราบว่า ในระบบปฏิบัติการ FreeBSD นั้นมีโปรแกรมหรือ Packages ต่าง ๆ ให้เราได้เลือกติดตั้ง เมื่ออ่านคำแนะนำเสร็จแล้วก็กดปุ่ม Enter ดังรูปที่ 10



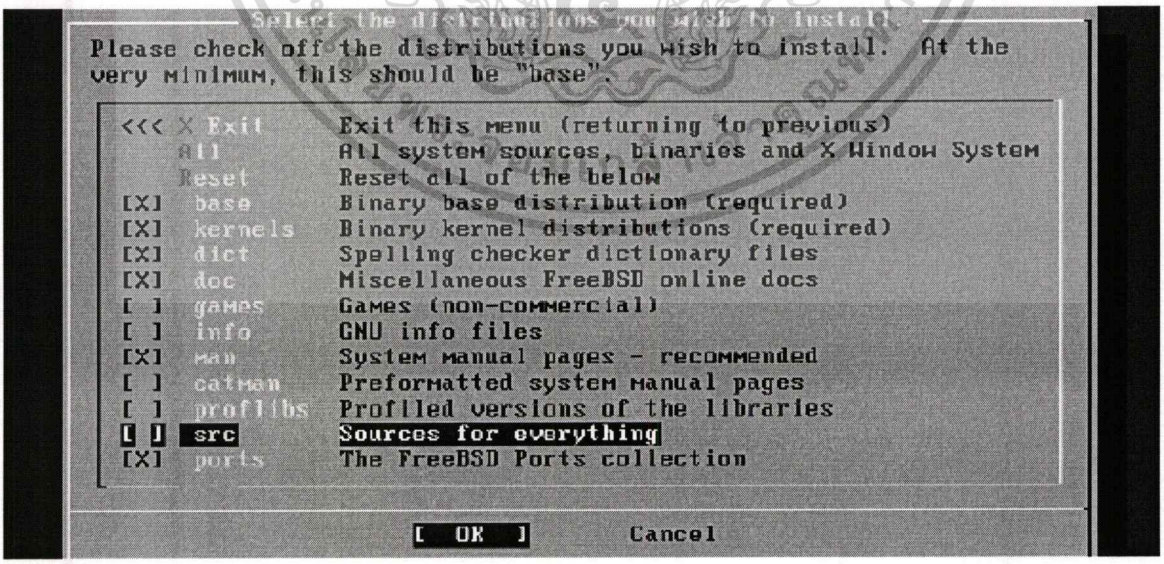
รูปที่ 10 คำแนะนำเกี่ยวกับ Packages ต่าง ๆ

11. โปรแกรมจะให้เราเลือกประเภทของ Distributions ที่เราต้องการติดตั้ง ได้แก่ตัวเลือกที่ 4, 6, 8 และ B Custom เท่านั้น ดังรูปที่ 11



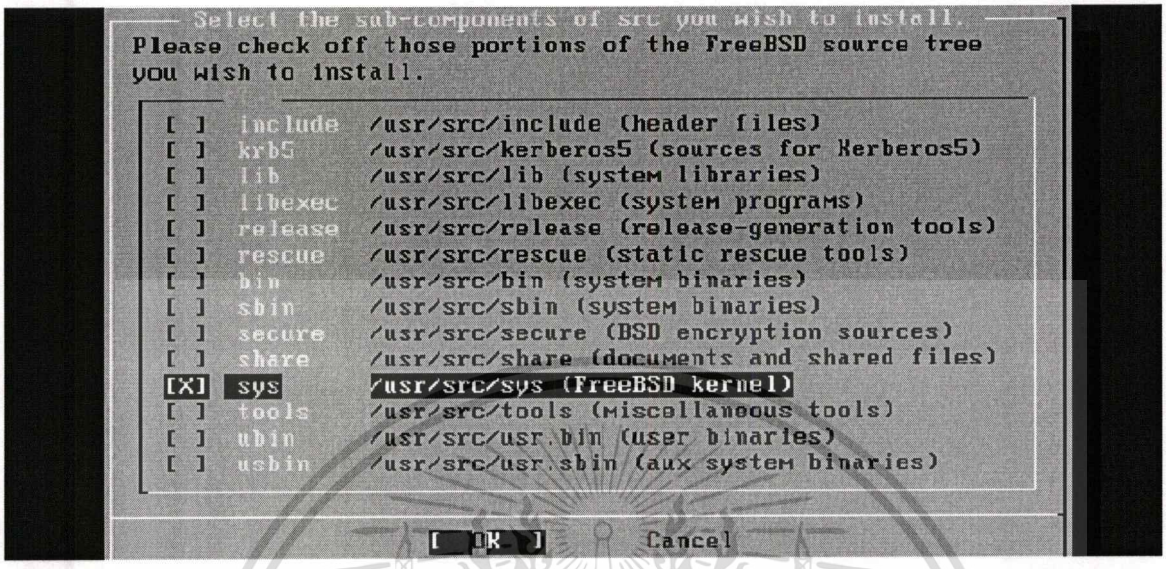
รูปที่ 11 เลือก Distributions เพื่อใช้งาน

12. เมื่อเราเลือกตัวเลือก B Custom แล้ว ก็จะปรากฏหน้าจอให้เราเลือกตัวเลือกย่อยอีก ในที่นี้จะเลือกตัวเลือก [ ] src โดยการกดปุ่ม Space Bar เพื่อเป็นการเลือก ดังรูปที่ 12



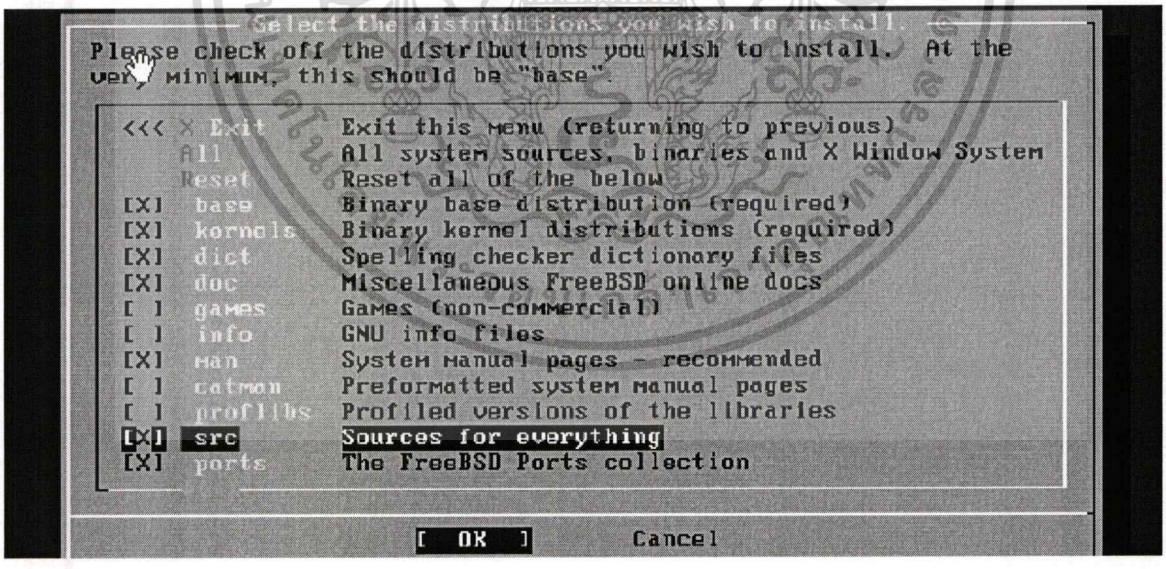
รูปที่ 12 เลือกตัวเลือก src ที่อยู่ใน B Custom

13. ให้เลือกตัวเลือก [ ] sys โดยการกดปุ่ม Space Bar เพื่อเลือก ซึ่งตัวเลือกนี้จะเป็น FreeBSD Kernel จากนั้นให้กดปุ่ม TAB มาที่ OK ดังรูปที่ 13



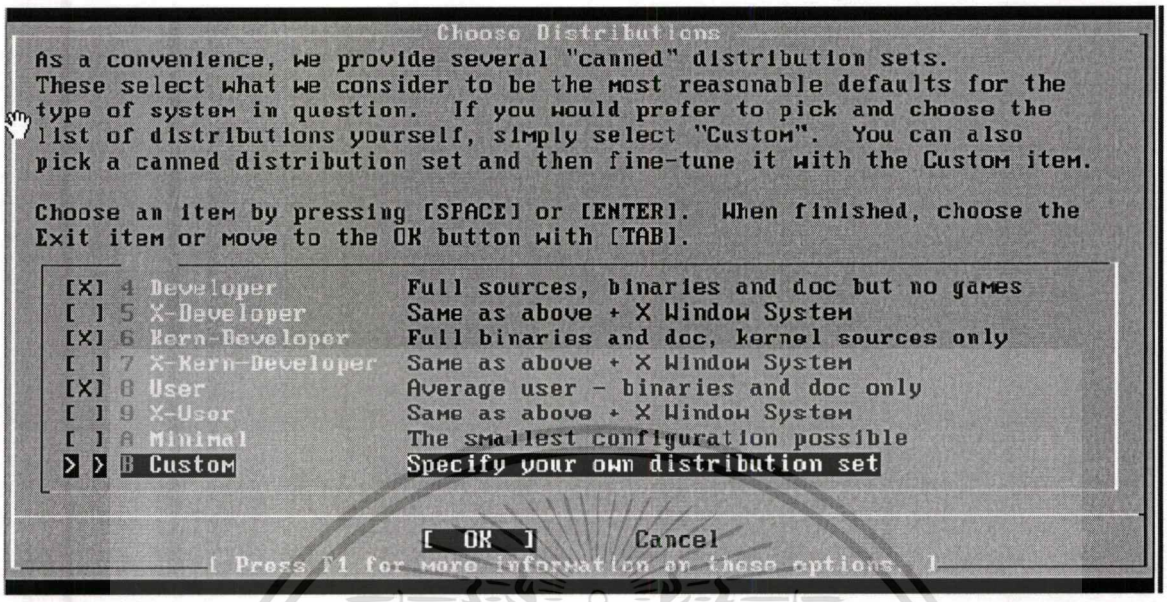
รูปที่ 13 เลือกตัวเลือก sys (FreeBSD Kernel)

14. กดปุ่ม TAB มาที่ OK อีกครั้งเพื่อเป็นการย้อนกลับไปเมนูก่อนหน้านี้ ดังรูปที่ 14



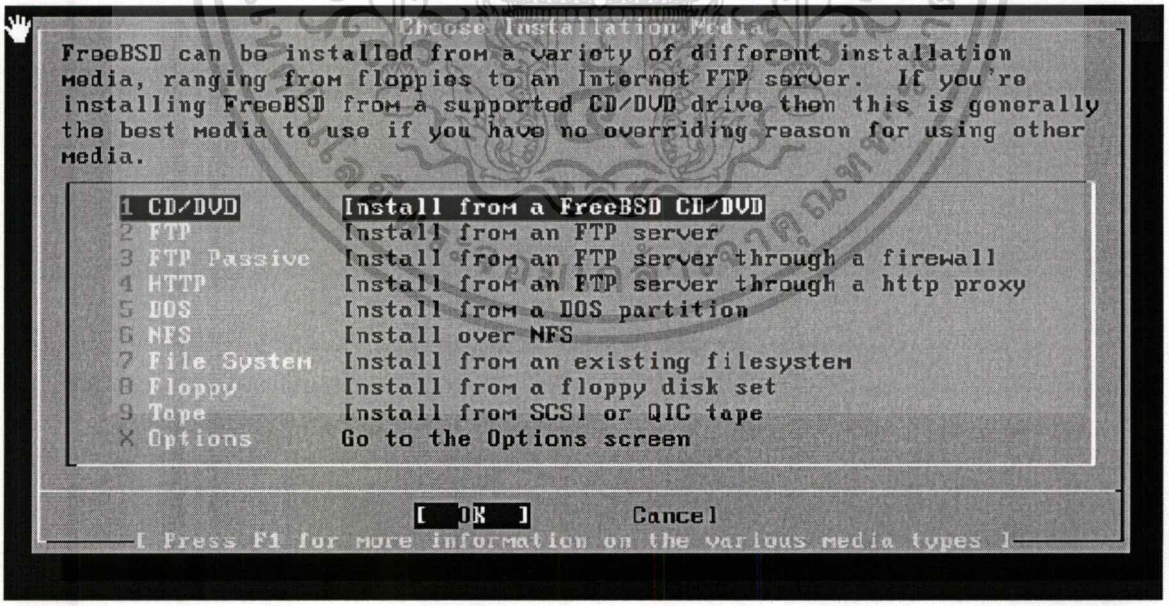
รูปที่ 14 การกดปุ่ม TAB เพื่อย้อนกลับเมนู

15. กดปุ่ม TAB มาที่ OK เพื่อย้อนกลับไปเมนูก่อนหน้านี้



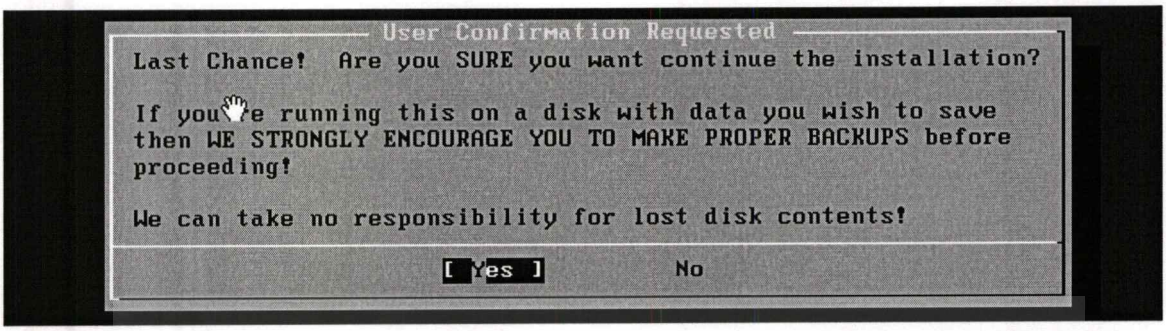
รูปที่ 15 กดปุ่ม TAB เพื่อย้อนกลับไปเมนูก่อนหน้านี้

16. โปรแกรมติดตั้งจะให้เราเลือกว่าต้องการติดตั้งระบบปฏิบัติการ FreeBSD จากสื่อประเภทใด ในที่นี้แนะนำเป็นตัวเลือกที่ 1 คือ CD/DVD แล้วกดปุ่ม Enter ดังรูปที่ 16



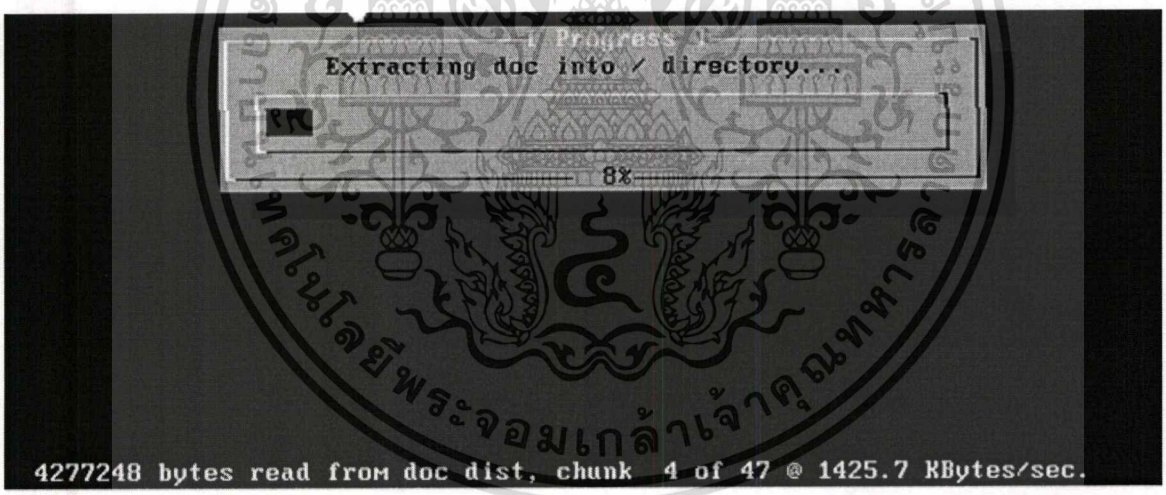
รูปที่ 16 เลือกสื่อที่ใช้ในการติดตั้ง FreeBSD

17. ตอบ Yes เพื่อทำการติดตั้งติดตั้งต่อไป ดังรูปที่ 17



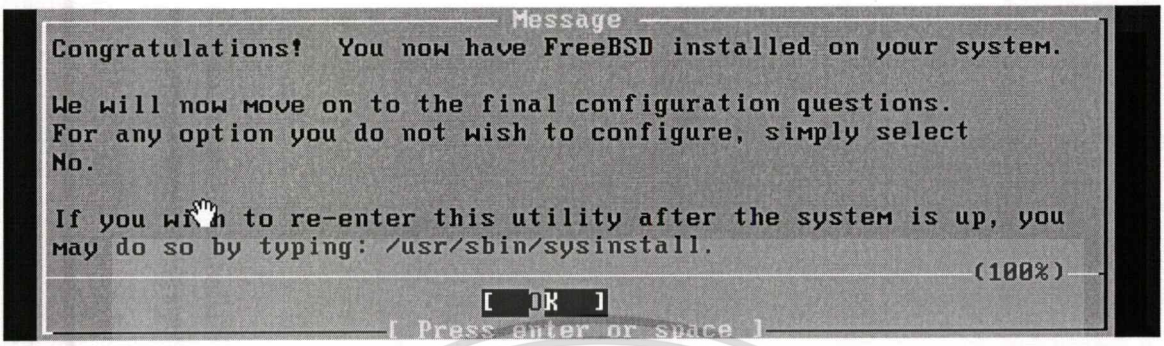
รูปที่ 17 โปรแกรมถามความแน่ใจในการติดตั้ง

18. หลังจากที่ยืนยันการติดตั้งระบบปฏิบัติการ FreeBSD แล้วโปรแกรมจะเริ่มทำการ Format Hard Disk ของเครื่องเซิร์ฟเวอร์แล้วติดตั้งโปรแกรมต่าง ๆ ไปตามโครงสร้างที่ได้กำหนดไว้ ดังรูปที่ 18



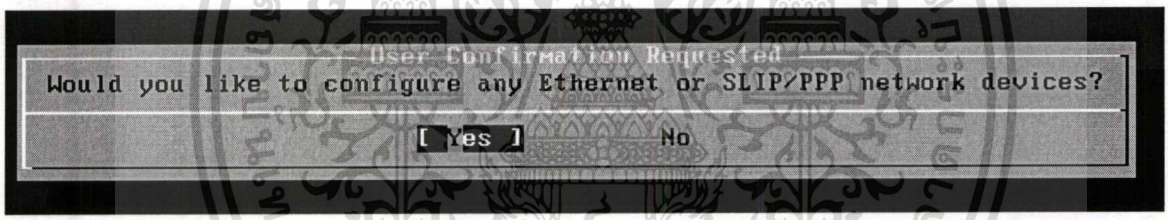
รูปที่ 18 โปรแกรมเริ่ม Format และ ติดตั้ง FreeBSD

19. เมื่อโปรแกรมได้ติดตั้ง Source ต่าง ๆ ลงในโครงสร้างต่าง ๆ เรียบร้อยแล้ว ก็จะแสดงหน้าจอแสดงความยินดีให้เราทราบว่า ได้ทำการติดตั้งระบบปฏิบัติการ FreeBSD ให้เราเรียบร้อยแล้ว ดังรูปที่ 19



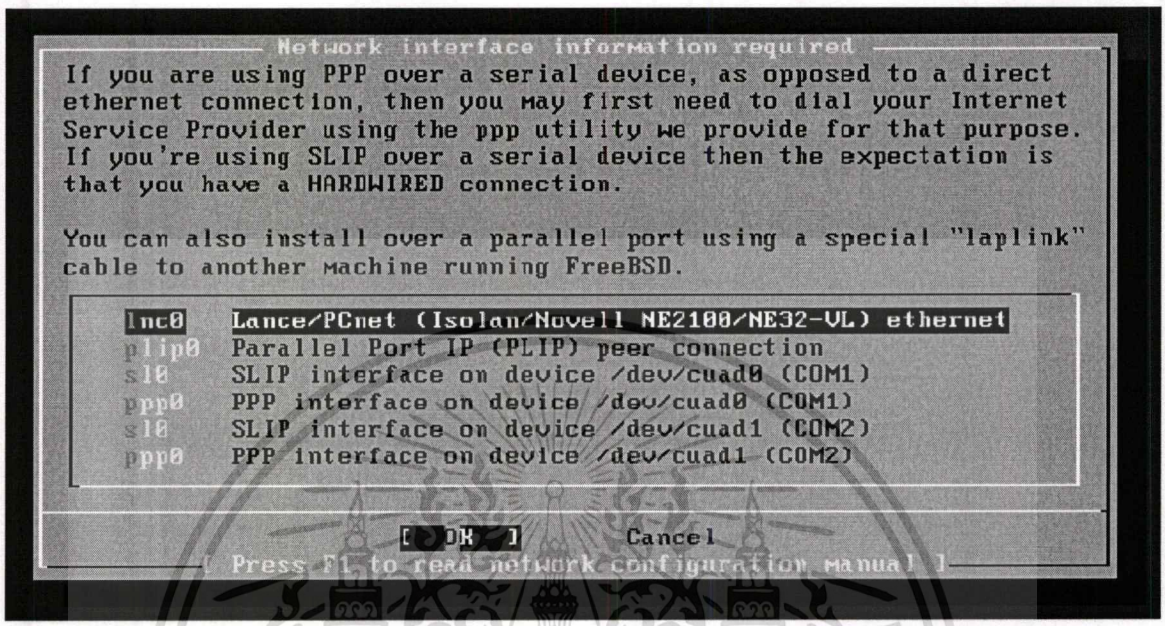
รูปที่ 19 หน้าจอแสดงความยินดีเมื่อติดตั้งเสร็จ

20. โปรแกรมจะให้เรากำหนดค่าของเครือข่าย หรือที่เราเรียกว่า Network Devices ในที่นี้เราจะต้องกำหนดค่าให้กับอุปกรณ์เครือข่ายของเราโดยการตอบ Yes ดังรูปที่ 20



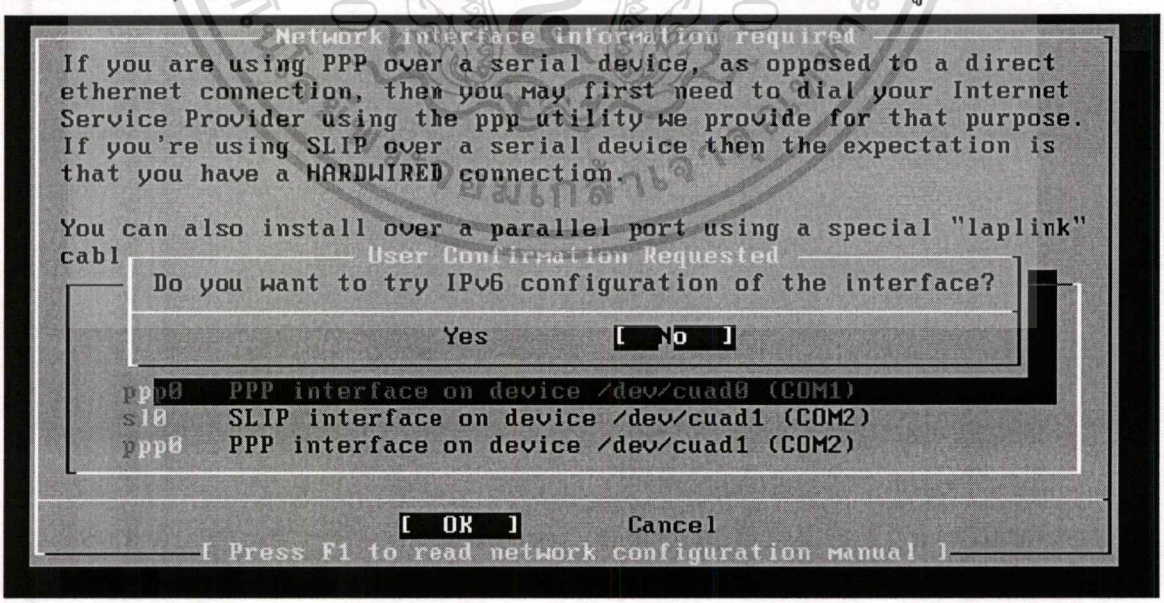
รูปที่ 20 กำหนดค่าเครือข่าย Network Devices

21. เลือกอุปกรณ์เครือข่าย Network Devices ในที่นี้เราจะเลือก lnc0 (เป็น LAN Card ที่เราใช้ในการติดตั้งนี้ แต่ในความเป็นจริง เราอาจจะพบเป็นชื่ออุปกรณ์อื่น ๆ ก็ได้ขึ้นอยู่กับยี่ห้อเช่น rl0, fxp0) กดปุ่ม Enter เพื่อทำงานต่อไป ดังรูปที่ 21



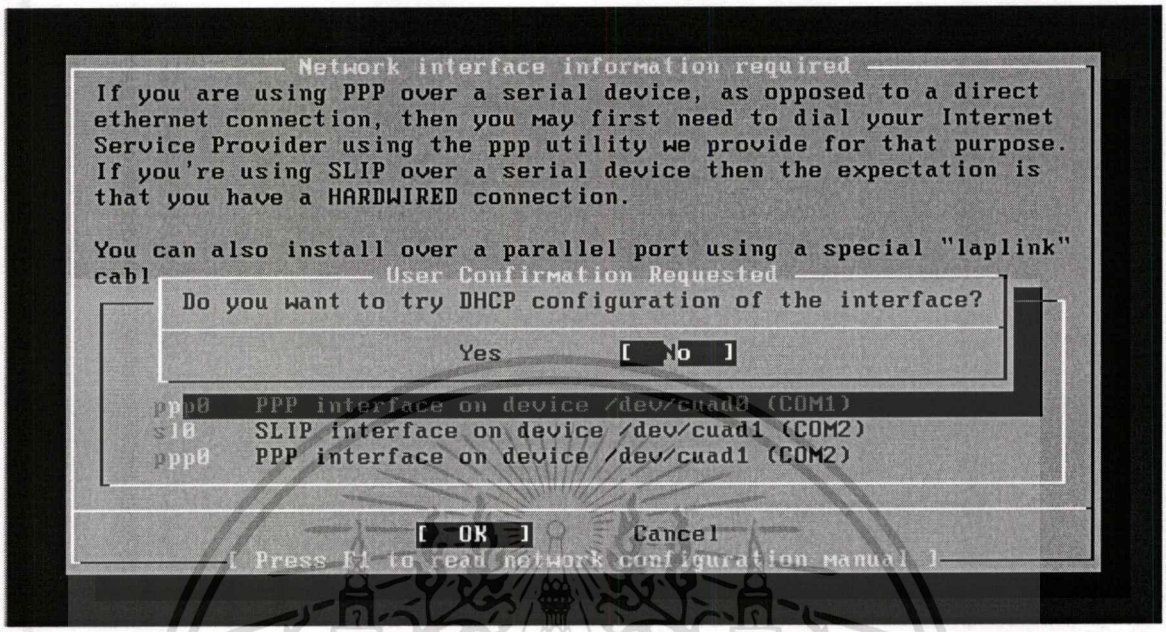
รูปที่ 21 เลือกอุปกรณ์ Network Devices

22. เราต้องการใช้งาน IPv6 (Internet Address Version 6 คือ มาตรฐานใหม่ของ IP address ที่มีขนาด 128 Bits) ในที่นี้เราจะตอบ No เนื่องจากยังไม่ได้ใช้ IPv6 ในการทำงาน ดังรูปที่ 22



รูปที่ 22 ไม่เลือก IPv6 Interface

23. ต้องการปรับแต่งหรือใช้บริการ DHCP หรือไม่ ในที่นี้ตอบ No เพื่อเราจะทำการกำหนดค่า IP address ให้กับอุปกรณ์เครือข่ายของเราเอง ดังรูปที่ 23

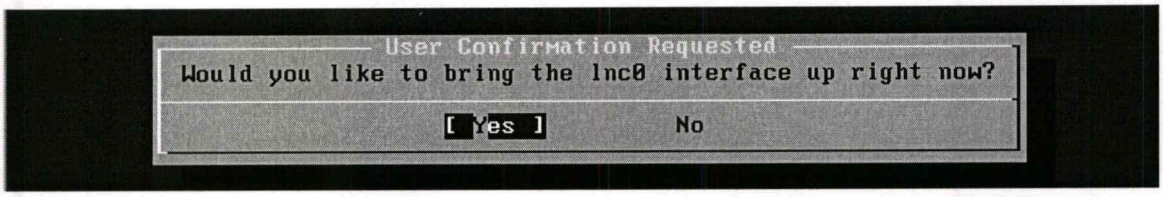


รูปที่ 23 ไม่ได้ปรับแต่งหรือใช้บริการ DHCP

24. กำหนดค่าให้กับอุปกรณ์เครือข่าย จะต้องกำหนดให้สอดคล้องกับเครือข่ายที่ใช้งานอยู่สามารถอธิบายความหมายของค่าต่าง ๆ ได้ดังนี้

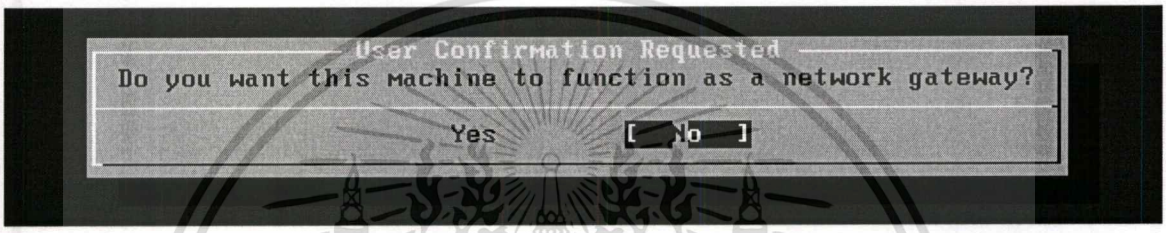
- Host : ชื่อเซิร์ฟเวอร์
- Domain : ชื่อ Domain ของเครื่องเซิร์ฟเวอร์
- IPv4 Gateway: หมายเลข IP Address ของ Gateway ที่ออกสู่ Internet
- Name Server : หมายเลข IP Address ของ DNS Server ขององค์กร หากไม่มีก็ให้กำหนดเป็นค่าของ DNS ของผู้ให้บริการหรือ ISP (Internet Service Providers)
- IPv4 address : หมายเลข IP Address ของเครื่องเซิร์ฟเวอร์
- Netmask : หมายเลข Netmask ของเครื่องเซิร์ฟเวอร์

25. ตอบ Yes เพื่อกำหนดให้อุปกรณ์เครือข่ายทำงาน ดังรูปที่ 25



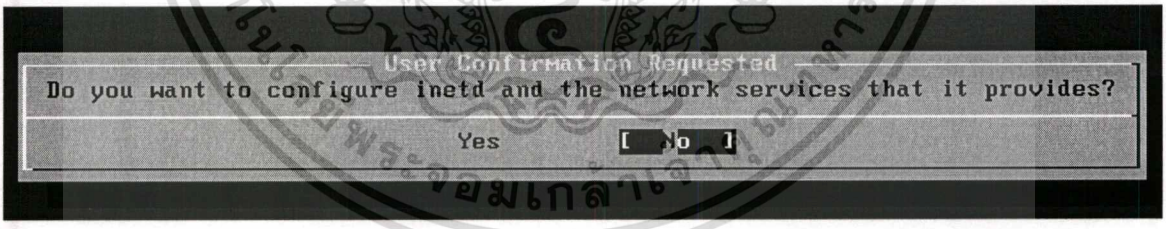
รูปที่ 25 กำหนดให้อุปกรณ์เครือข่ายทำงาน

26. ตอบ Yes เพื่อให้เครื่องเซิร์ฟเวอร์นี้เป็น Network Gateway ดังรูปที่ 26



รูปที่ 26 กำหนดค่า Network Gateway

27. ตอบ No เพราะไม่ต้องการที่จะแก้ไขและปรับแต่งค่าของ inetd และ Network Service ดังรูปที่ 27

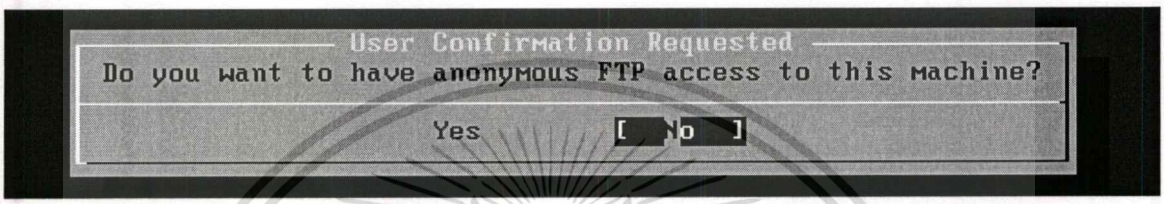


รูปที่ 27 การปรับแต่งค่า inetd (Internet Daemon)

28. ตอบ Yes เพื่อเปิดบริการ SSH Loginเนื่องจากเมื่อเราติดตั้งระบบปฏิบัติการ FreeBSD เรียบร้อยแล้ว เราสามารถที่จะ Remote เข้ามาได้จากภายนอก คล้าย ๆ กับบริการ Telnet แต่ SSH นั้นจะมีการเข้า รหัสข้อมูลจากเครื่องคอมพิวเตอร์ที่ Remote เข้ามายังเซิร์ฟเวอร์ ดังรูปที่ 28

รูปที่ 28 เปิดบริการ SSH Login

29. ตอบ No เพื่อการปิดบริการ Anonymous FTP Service ดังรูปที่ 29

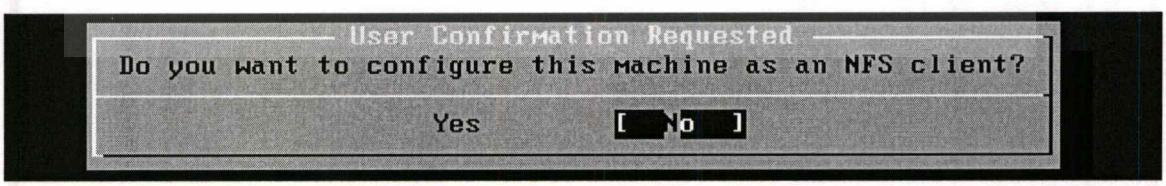


รูปที่ 29 การปิดบริการ Anonymous FTP Service

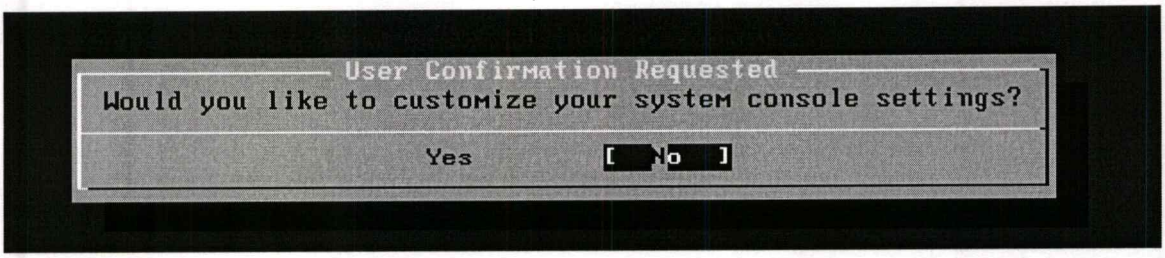
30. ตอบ No เพื่อการปิดบริการ NFS Server ดังรูปที่ 30

รูปที่ 30 ปิดบริการ NFS Server

31. ตอบ No เพื่อการปิดบริการ NFS Client ดังรูปที่ 31

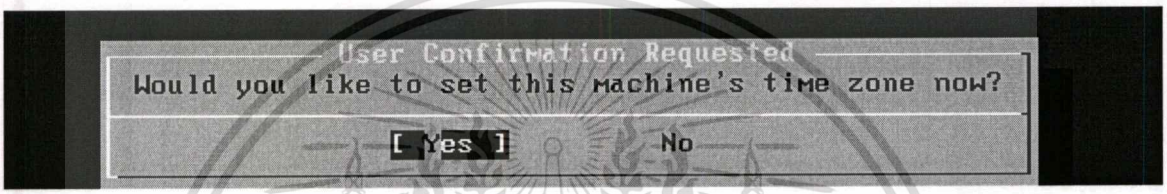


รูปที่ 31 ปิดบริการ NFS Client32. ตอบ No เพื่อไม่ปรับแต่งค่าของ Console Setting ดังรูปที่ 32



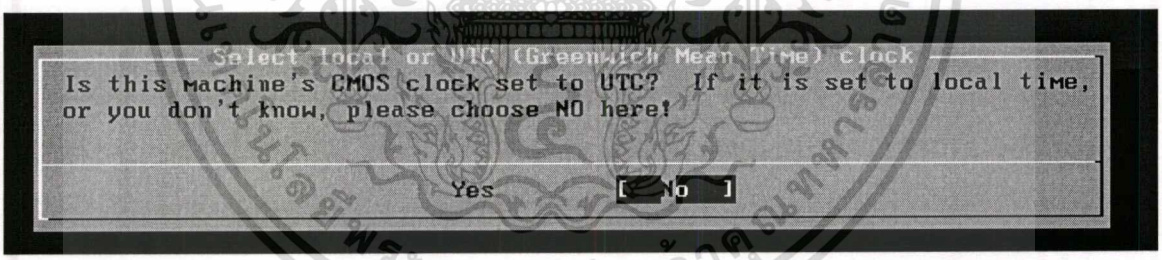
รูปที่ 32 ไม่ปรับแต่งค่าของ Console Setting

33. ระบบจะถามว่าต้องการปรับแต่งค่าของ Time Zone หรือไม่ ในที่นี้ให้ตอบ Yes เพื่อปรับแต่งค่าของ Time Zone ดังรูปที่ 33

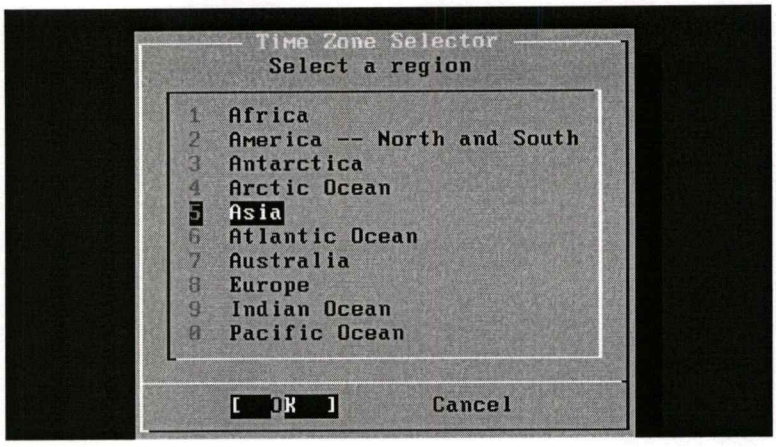


รูปที่ 33 ปรับแต่งค่าของ Time Zone

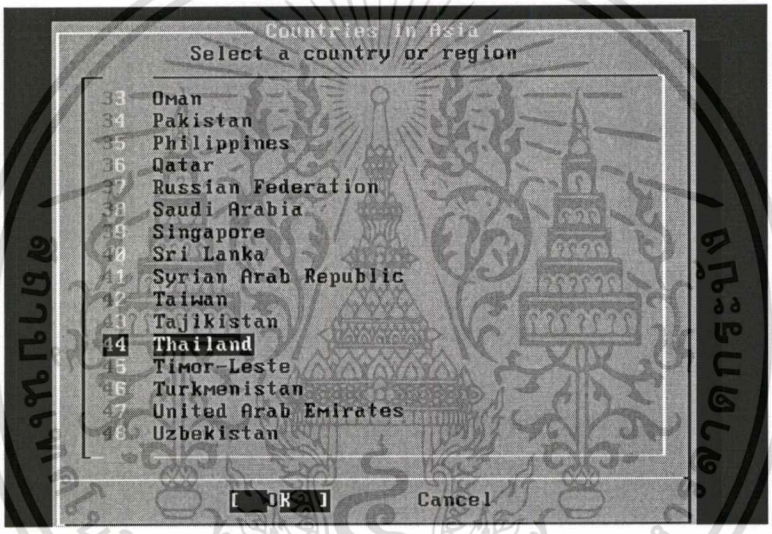
34. ตอบ No เพื่อจะได้ปรากฏตัวช่วยในการเลือก Time Zone ดังรูปที่ 34.1, 34.2, 34.3 และ 34.4



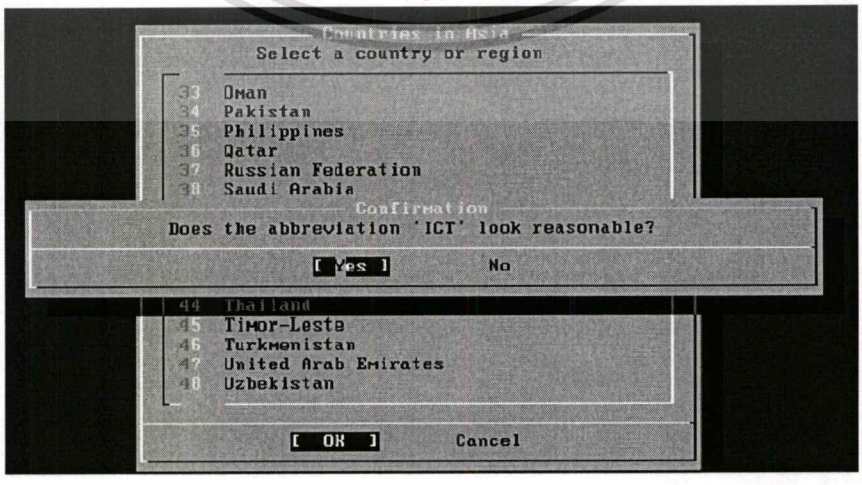
รูปที่ 34.1 ปรับเวลามาตรฐาน UTC



รูปที่ 34.2 ปรับเวลามาตรฐาน UTC

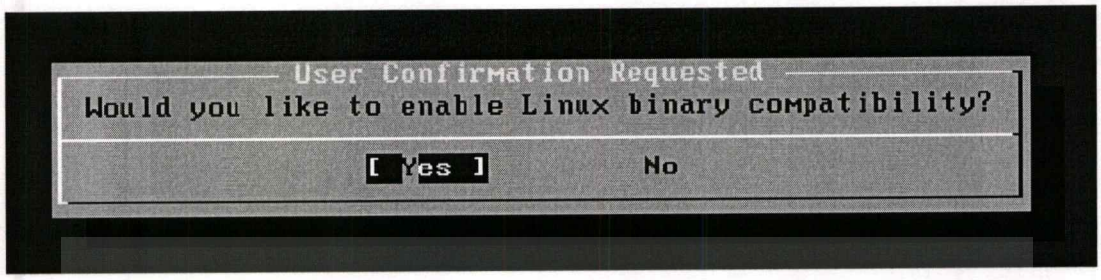


รูปที่ 34.3 ปรับเวลามาตรฐาน UTC



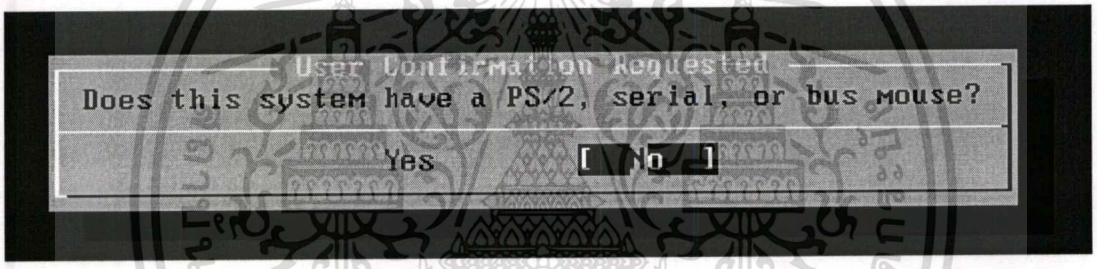
รูปที่ 34.4 ปรับเวลามาตรฐาน UTC

35. ระบบถามว่าต้องการอนุญาตให้ โปรแกรมของระบบปฏิบัติการ Linux สามารถนำมาใช้กับ FreeBSD ได้หรือไม่ ให้เลือก Yes เพราะเราจะได้นำโปรแกรมของ Linux มาใช้งานร่วมกับ FreeBSD ได้ ดังรูปที่ 35



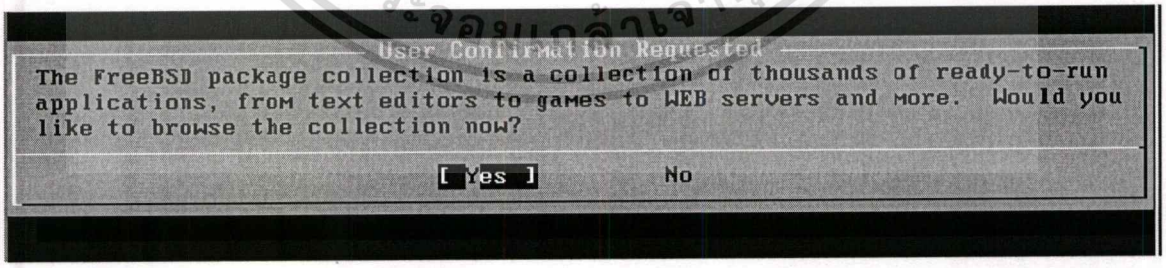
รูปที่ 35 การยอมรับให้โปรแกรม Linux ทำงานบน FreeBSD

36. ตอบ No เนื่องจากการใช้งานเซิร์ฟเวอร์นี้ไม่ได้ใช้ Mouse ดังรูปที่ 36



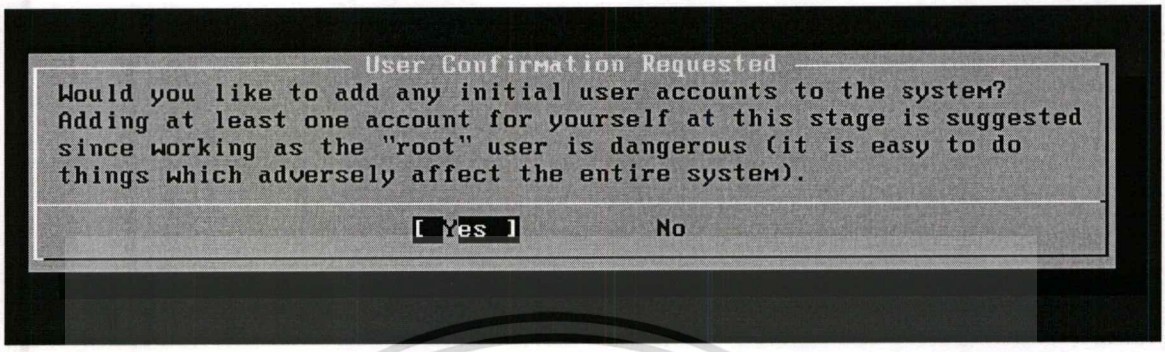
รูปที่ 36 ไม่มีการใช้ Mouse ในระบบ

37. ตอบ No เพราะไม่มีการติดตั้ง FreeBSD Packages เพิ่มเติม ดังรูปที่ 37



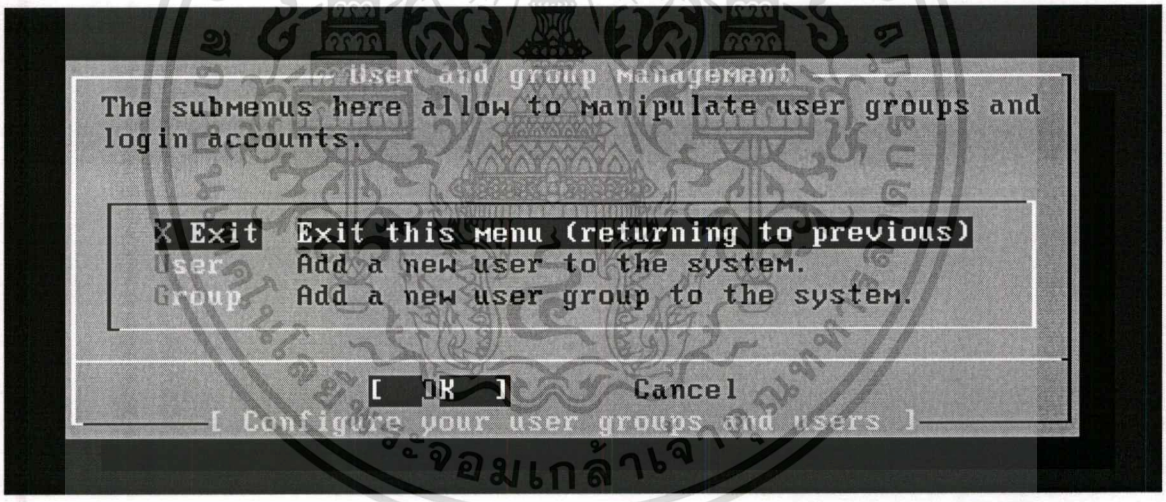
รูปที่ 37 การติดตั้ง FreeBSD Packages เพิ่มเติม

38. ระบบจะให้เราจัดการกับ Username และ Group ร่วมถึงการจัดการกับ root account (เป็น Account ที่เป็นผู้บริหารระบบ) ซึ่งเราสามารถที่จะจัดการได้จากเมนูนี้ ให้เลือกตอบ Yes ดังรูปที่ 38.1



รูปที่ 38.1 เลือกจัดการกับ User และ Group

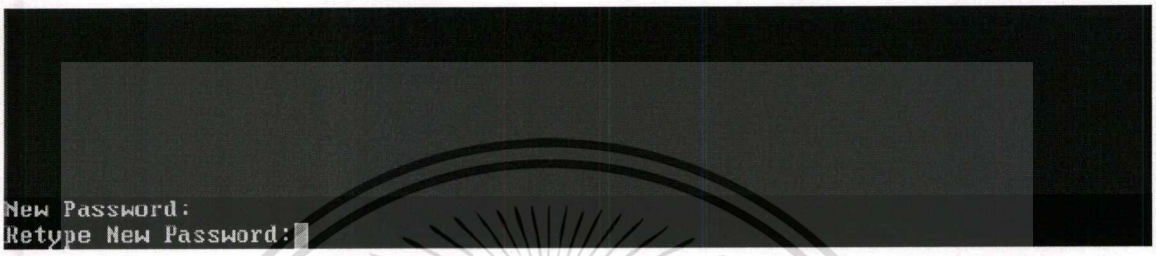
ให้เลือกคำสั่ง Exit เพื่อไม่เพิ่ม User Name และ Group ดังรูปที่ 38.2



รูปที่ 38.2 เลือก Exit เพื่อออกจากการทำงาน

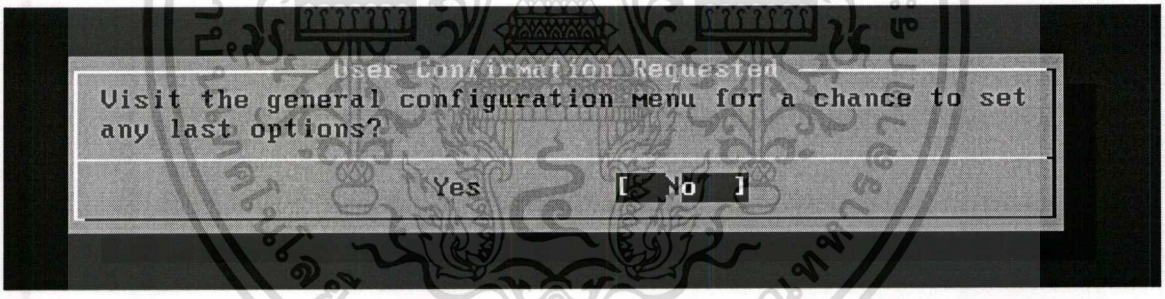
จากนั้นจะปรากฏหน้าจอแนะนำให้เรากำหนดรหัสผ่านให้กับผู้ดูแลระบบที่เราเรียกว่า root โดยจะให้เราพิมพ์รหัสผ่านเข้าไปให้เหมือนกันสองครั้ง ซึ่งการพิมพ์รหัสผ่านนี้จะไม่มีการแสดงตัวอักษรใด ๆ บนหน้าจอภาพ ขอให้พิมพ์ให้ถูกต้องแล้วกดปุ่ม Enter ผ่านได้เลย ดังรูปที่ 38.3 และ 38.4

รูปที่ 38.3 คำแนะนำในการกรอกรหัสผ่านของ root



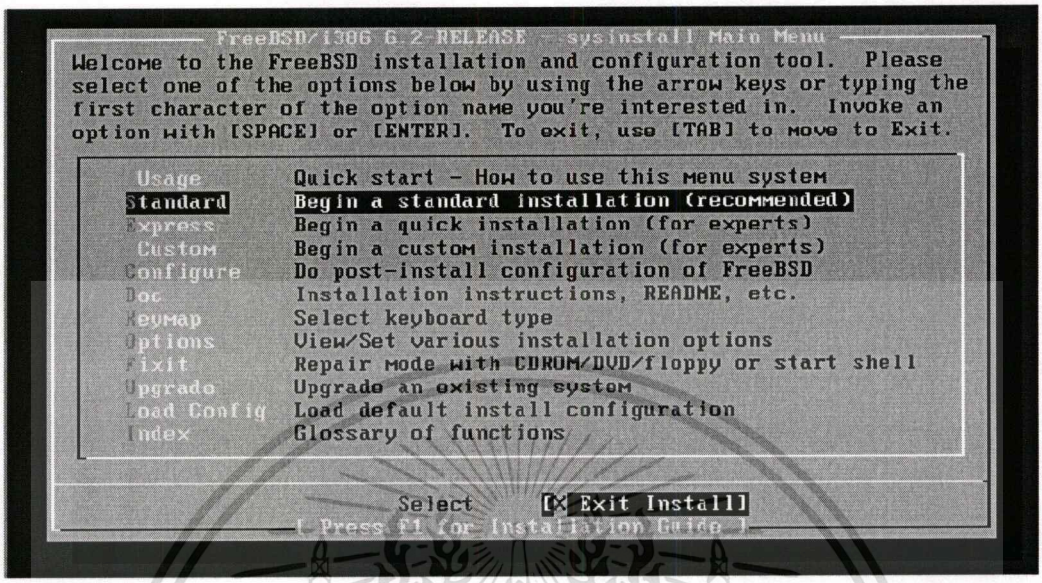
รูปที่ 38.4 กำหนดรหัสผ่านให้เหมือนกัน 2 ครั้ง

39. เลือก No เพราะไม่ต้องการกลับไปปรับแต่งค่าใด ๆ ดังรูปที่ 39



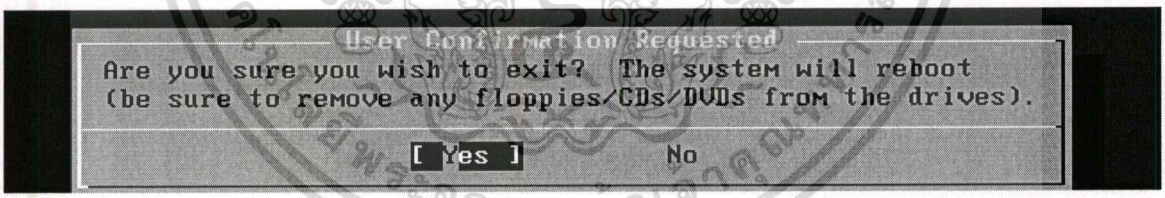
รูปที่ 39 ไม่เลือกปรับแต่งค่าต่าง ๆ ของระบบ

40. จากนั้นจะกลับมาที่เมนูหน้าแรกของการติดตั้งระบบปฏิบัติการ FreeBSD ให้เราเลือกปุ่ม Exit Install แล้ว Enter เพื่อเสร็จสิ้นการติดตั้ง



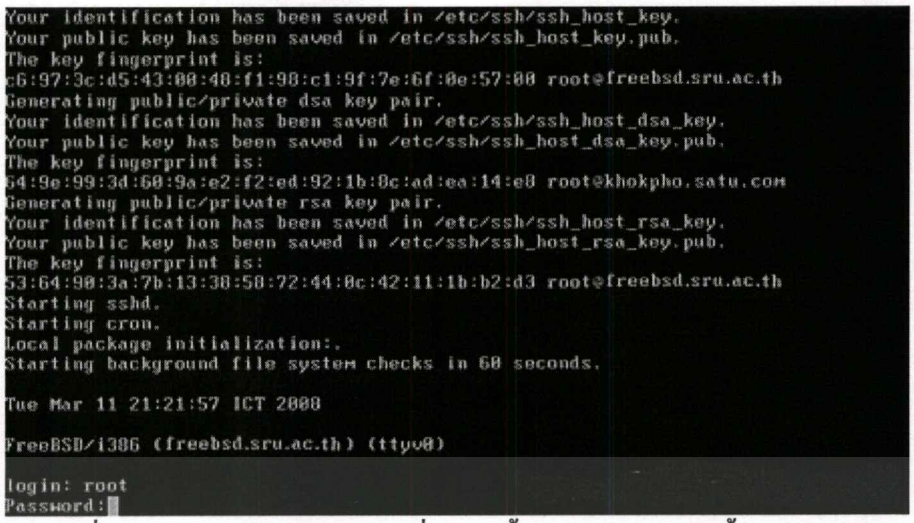
รูปที่ 41 เลือกเมนู Exit Install

41. โปรแกรมจะแนะนำให้เราเอาแผ่นติดตั้งระบบปฏิบัติการ FreeBSD ออกจาก CD-Rom Drive หรือ DVD Drive แล้วเลือกตัวเลือก Yes



รูปที่ 41 นำแผ่นติดตั้งออกจาก CD-Rom Drive

เมื่อนำแผ่นติดตั้งระบบปฏิบัติการ FreeBSD ออกจาก CD-Rom Drive แล้ว ระบบจะทำการรีบูทเครื่องเซิร์ฟเวอร์ จากนั้นก็เป็นการสิ้นสุดการติดตั้งระบบปฏิบัติการ FreeBSD หลังจากที่ระบบปฏิบัติการ FreeBSD ได้ทำการบูทเสร็จเรียบร้อยแล้ว เราจะสามารถ Login เข้าระบบโดยใช้ root account แล้วใส่รหัสผ่านที่ได้กำหนดไว้ในขั้นตอนติดตั้งระบบ ดังรูปที่ 42



รูปที่ 42 การ Login เข้าสู่ระบบ เมื่อเสร็จสิ้นกระบวนการติดตั้ง FreeBSD

<u>การติดตั้ง Apache Web Server</u>

เว็บเซิร์ฟเวอร์ หมายถึง Application ที่ทำหน้าที่รับและประมวลผลเอกสารที่ถูกร้องขอจากผู้ใช้บริการทางอินเทอร์เน็ต ซึ่งเว็บเซิร์ฟเวอร์จะส่งเอกสารกลับไปแสดงผลให้ผู้ใช้บริการผ่านบราวเซอร์นอกจากเว็บเซิร์ฟเวอร์ จะถูกนำมาให้บริการในอินเทอร์เน็ตแล้วแต่อาจมีการประยุกต์ให้นำมาใช้กับเครือข่ายภายในองค์กรหรืออินเทอร์เน็ตได้เช่นกัน

โปรแกรมที่สามารถทำหน้าที่หรือให้บริการเว็บเซิร์ฟเวอร์บนระบบ FreeBSD มีหลายโปรแกรม เช่น Apache เวอร์ชั่น 2.2.8 โดยมีการใช้งานร่วมกับ PHP เวอร์ชั่น 5.2.5 และ MySQL เวอร์ชั่น 5 ในส่วนของ Apache สามารถดาวน์โหลด Source Code ได้ที่ http://www.apache.org/ ขั้นตอนการติดตั้งมีรายละเอียด ดังต่อไปนี้

1. ให้เราเข้าไปยังไดเร็กทอรี /usr/ports/www/apache22 แล้วทำการติดตั้ง Apache Web Server ด้วยคำสั่งดังนี้

```
# cd /usr/ports/www/apache22
# make install
```

2. โปรแกรมจะเริ่มติดตั้งแล้วจะปรากฏหน้าจอภาพ ถามเกี่ยวกับ Options เสริมของ gettext ในที่นี้ให้เรากดปุ่ม Tab มาที่ OK แล้วกดปุ่ม Enter เพื่อทำงานต่อไป

3. ให้รอจนกว่าระบบปฏิบัติการ FreeBSD จะทำการคอมไฟล์ (Compile) Apache จนเรียบร้อยก็จะปรากฏเครื่องหมาย Prompt (#) อีกครั้ง จากนั้นให้เราทำการปรับแต่ง Apache ดังนี้

```
# vi /usr/local/etc/apache22/httpd.conf
```

เพิ่มบรรทัดคำสั่งต่อไปนี้ลงไป

DirectoryIndex index.php index.html

AddDefaultCharset tis-620

AddType application/x-httpd-php .php

AddType application/x-httpd-php-source .phps

4. การที่เราจะสั่งให้ Apache Web Server ทำงานทุกครั้งที่มีการบูทเครื่องก็สามรถทำได้โดยการเพิ่มคำสั่ง apache22_enable="YES" ในไฟล์ /etc/rc.conf ดังนี้

vi /etc/rc.conf

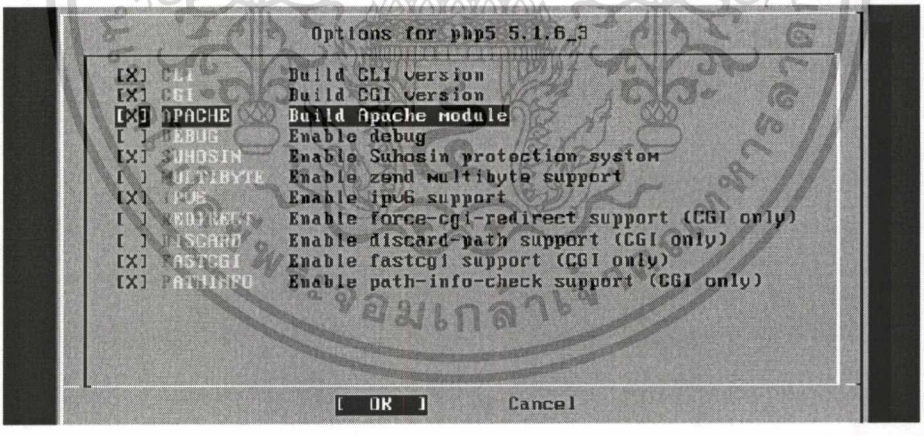
<u>การติดตั้งภาษา PHP 5</u>

1. เริ่มการติดตั้งโดยการเข้าไปยังไดเร็กทอรี /usr/ports/lang/php5 ซึ่งจะเป็นภาษา PHP เวอร์ชั่น 5 ซึ่งจำเป็นต่อการเขียนโปรแกรมภาษา PHP อย่างมากในปัจจุบัน เราสามารถติดตั้งได้โดยการใช้คำสั่งด้านล่าง

cd /usr/ports/lang/php5

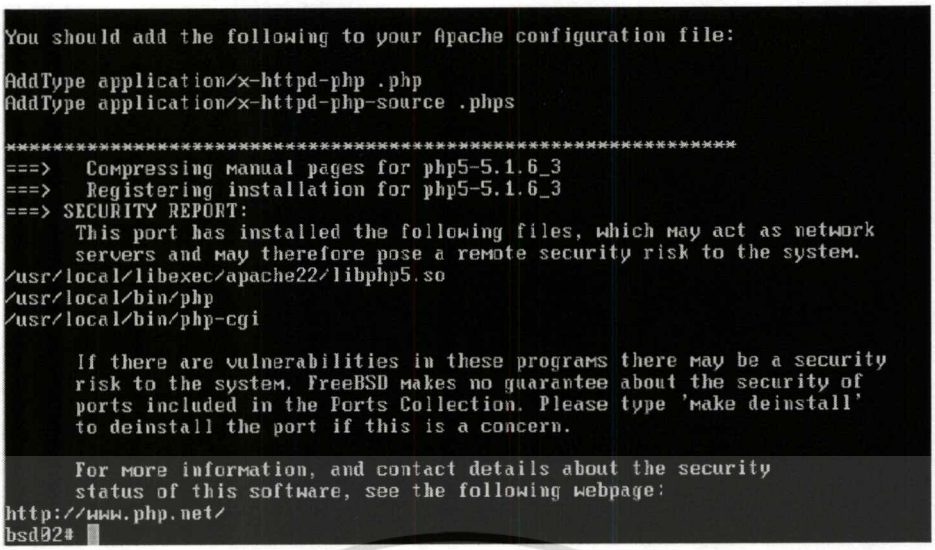
make install

2. เลือก Options ของ Apache ซึ่งจะเป็นโมดูล (Modules) ที่จำเป็นต้องใช้งานเพื่อให้ Apache Web Server รับรู้ภาษา PHP ใหม่นี้ด้วย จากนั้นให้กดปุ่ม TAB มาที่ OK แล้ว Enter ดังรูปที่ 1



รูปที่ 1 การเลือก Apache Modules ของ PHP เพิ่มเติม

3. เมื่อโปรแกรมภาษา PHP ได้ทำการติดตั้งและคอมไฟล์ (Compile) เสร็จเรียบร้อยแล้ว จะปรากฏหน้า จอ ดังรูปที่ 2 เพื่อให้ตรวจสอบว่าการมีการเพิ่มบรรทัดคำสั่ง AddType ทั้ง 2 บรรทัดในไฟล์ httpd.conf



รูปที่ 2 ภาษา PHP ถูกติดตั้งเรียบร้อยแล้ว

4. หากเราต้องการทดสอบนั้นก็สามารถทำได้โดยการเขียนโปรแกรมภาษา PHP เพื่อทดสอบ ตามขั้นตอนนี้

cd /usr/local/www/data

vi test.php

เขียนคำสั่งเพิ่มหนึ่งบรรทัด ดังนี้

<? phpinfo(); ?>

หลังจากนั้นให้เปิดโปรแกรม Internet Explorer เรียก URL ตาม IP address ของเครื่อง เช่น http://192.168.2.1/test.php เป็นต้น หากทำได้ถูกต้องก็จะเขียนรายละเอียดของภาษา PHP ให้เราได้ทราบ เป็นการเสร็จสิ้นการติดตั้งภาษา PHP

<u>การติดตั้งฐานข้อมูล MySQL</u>

การติดตั้งฐานข้อมูล MySQL นั้น มีหลายวิธี หากเลือกใช้การติดตั้งด้วย Source Code ก็สามารถทำได้โดยการดาวน์โหลดไฟล์จากเว็บไซต์ http://www.mysql.com/ เลือกเวอร์ชั่นที่ หรือเลือกติดตั้งผ่าน Ports โดยมีขั้นตอนการติดตั้ง ดังต่อไปนี้

tar zvfx mysql-5.1.xxxx.tar (จะได้ Directory ตามชื่อ)

mv mysql-5.1.xxxx.tar /usr/local/mysql

การติดตั้งโปรแกรม SNORT

ดาวน์โหลด Snort เวอร์ชั่นล่าสุดจาก http://www.snort.org/ จากนั้นให้ขยายไฟล์ออก ดังนี้

```
#tar xzf snort-x.x.x.tar.gz -C /usr/local
```

ให้ compile โดยเพิ่ม option ดังนี้ --with-mysql-includes=DIR และ --with-mysql-libraries=DIR เช่น

```
#cd /usr/local/snort
#./configure --with-mysql-includes=/usr/include/mysql --with-mysql-
libraries=/usr/lib/mysql
#make
#make install
```

ทั้งนี้ในขณะที่โปรแกรมกำลังคอมไพล์อยู่นั้น ท่านควรจะสังเกตเห็นคำว่า **checking for mysql... yes** ปรากฏขึ้น จากนั้นให้ก๊อปปี้ข้อมูล configuration และ rules files จาก source ของ Snort ไปยัง /etc/snort เพื่อความเป็นระเบียบ

```
#mkdir /etc/snort
#cd /usr/local/src/snort
#cp snort.conf /etc/snort
#cp *.rules /etc.snort
#cp classification.config /etc/snort
```

สร้างไดเร็คทอรีเพื่อเก็บล็อกไฟล์ของ Snort ทั้งหมดแยกต่างหาก และควรป้องกันไม่ให้บุคคลอื่น access เข้ามาที่ไดเร็คทอรีนั้นๆ โดยปกติแล้วจะสร้างไว้ที่ /var/log/snort

```
#mkdir /var/log/snort
#chmod 700 /var/log/snort
```

สร้าง Database structure สำหรับ Snort

ให้ล็อกอินเข้าไปยัง MySQL และสร้าง database ชื่อ snort ขึ้นมา

#mysql -uroot -p

.....

mysql>CREATE DATABASE snort;

จากนั้นให้สร้าง MySQL account ขึ้นมาเพื่อให้มีสิทธิในการจัดการกับฐานข้อมูล

mysql>grant insert,delete,select,create,update on snort.* to snort@localhost;

mysql>flush privileges;

จากนั้นก็ต้องสร้าง database structure ตามที่ Snort กำหนดไว้

#cd /usr/local/snort

#vi contrib/create_mysql แล้วเพิ่มคำว่า USE snort; ไว้ที่บรรทัดบนสุด

#mysql < ./contrib/create_mysql -uroot –p

การปรับแต่งค่าไฟล์ snort.conf

เราจะทำการแก้ไขไฟล์ /etc/snort/snort.conf เนื่องจากเราใช้ snort.conf เป็นไฟล์หลักในการรัน Snort จึงจำเป็นต้องแก้ไขข้อมูลในบางส่วนดังต่อไปนี้

แก้ไข HOME_NET ให้เป็น network address ของเครือข่ายที่ต้องการมอนิเตอร์ เช่น var HOME_NET 172.16.23.0/24

แก้ไขค่า network ip address อื่นให้ตรงกับความต้องการ เช่น SMTP , SQL_SERVERS

สำหรับ parameter อื่นๆ หรือกฎต่างๆนั้นสามารถทำการเขียนเพิ่มเติมได้เอง โดยสามารถเลือกได้ว่าจะใช้งานกฎใดบ้าง ตามความเหมาะสมของการใช้งานในระบบนั้นๆ ด้านล่างจะเป็นตัวอย่างไฟล์ Snort.conf ที่ใช้งานในระบบ

ตัวอย่างไฟล์ snort.conf

var HOME_NET 172.16.23.0/24

var EXTERNAL_NET !$HOME_NET

var DNS_SERVERS $HOME_NET

var SMTP_SERVERS $HOME_NET

var HTTP_SERVERS $HOME_NET

```
var SQL_SERVERS $HOME_NET

var TELNET_SERVERS $HOME_NET

var SNMP_SERVERS $HOME_NET

var HTTP_PORTS 80

var SHELLCODE_PORTS !80

var ORACLE_PORTS 1521

var AIM_SERVERS
[64.12.24.0/23,64.12.28.0/23,64.12.161.0/24,64.12.163.0/24,64.12.200.0/24,205.188.3.0/24,205.1
88.5.0/24,205.188.7.0/24,205.188.9.0/24,205.188.153.0/24,205.188.179.0/24,205.188.248.0/24]

var RULE_PATH /etc/snort/rules

config disable_decode_alerts

preprocessor flow: stats_interval 0 hash 2

preprocessor frag3_global: max_frags 65536

preprocessor frag3_engine: policy first detect_anomalies

preprocessor stream4: disable_evasion_alerts

preprocessor stream4_reassemble

preprocessor http_inspect: global \

   iis_unicode_map unicode.map 1252

preprocessor http_inspect_server: server default \

  profile all ports { 80 8080 8180 } oversize_dir_length 500
```

```
preprocessor rpc_decode: 111 32771

preprocessor bo

preprocessor telnet_decode

preprocessor sfportscan: proto  { all } \

    memcap { 10000000 } \

    sense_level { low }

preprocessor arpspoof

preprocessor xlink2state: ports { 25 691 }

output database: log, mysql, user=snort password=12345 dbname=snort host=localhost

include classification.config

include reference.config

include $RULE_PATH/attack-responses.rules

include $RULE_PATH/backdoor.rules

include $RULE_PATH/bad-traffic.rules

include $RULE_PATH/ddos.rules

include $RULE_PATH/deleted.rules

include $RULE_PATH/dos.rules

include $RULE_PATH/experimental.rules

include $RULE_PATH/exploit.rules
```

include $RULE_PATH/finger.rules

include $RULE_PATH/info.rules

include $RULE_PATH/local.rules

include $RULE_PATH/misc.rules

include $RULE_PATH/multimedia.rules

include $RULE_PATH/other-ids.rules

include $RULE_PATH/p2p.rules

include $RULE_PATH/pop2.rules

include $RULE_PATH/policy.rules

include $RULE_PATH/porn.rules

include $RULE_PATH/scan.rules

include $RULE_PATH/shellcode.rules

include $RULE_PATH/virus.rules

include $RULE_PATH/smtp.rules

include $RULE_PATH/pop3.rules

include $RULE_PATH/imap.rules

include $RULE_PATH/web-iis.rules

include $RULE_PATH/web-php.rules

include $RULE_PATH/web-attacks.rules

include $RULE_PATH/web-cgi.rules

include $RULE_PATH/web-client.rules

include $RULE_PATH/web-coldfusion.rules

include $RULE_PATH/web-frontpage.rules

include $RULE_PATH/web-misc.rules

include $RULE_PATH/mysql.rules

include $RULE_PATH/sql.rules

include $RULE_PATH/oracle.rules

include $RULE_PATH/ftp.rules

include $RULE_PATH/tftp.rules

include $RULE_PATH/telnet.rules

include $RULE_PATH/snmp.rules

include $RULE_PATH/dns.rules

include $RULE_PATH/netbios.rules

include $RULE_PATH/icmp.rules

include $RULE_PATH/icmp-info.rules

include $RULE_PATH/nntp.rules

include $RULE_PATH/rpc.rules

include $RULE_PATH/rservices.rules

include $RULE_PATH/x11.rules

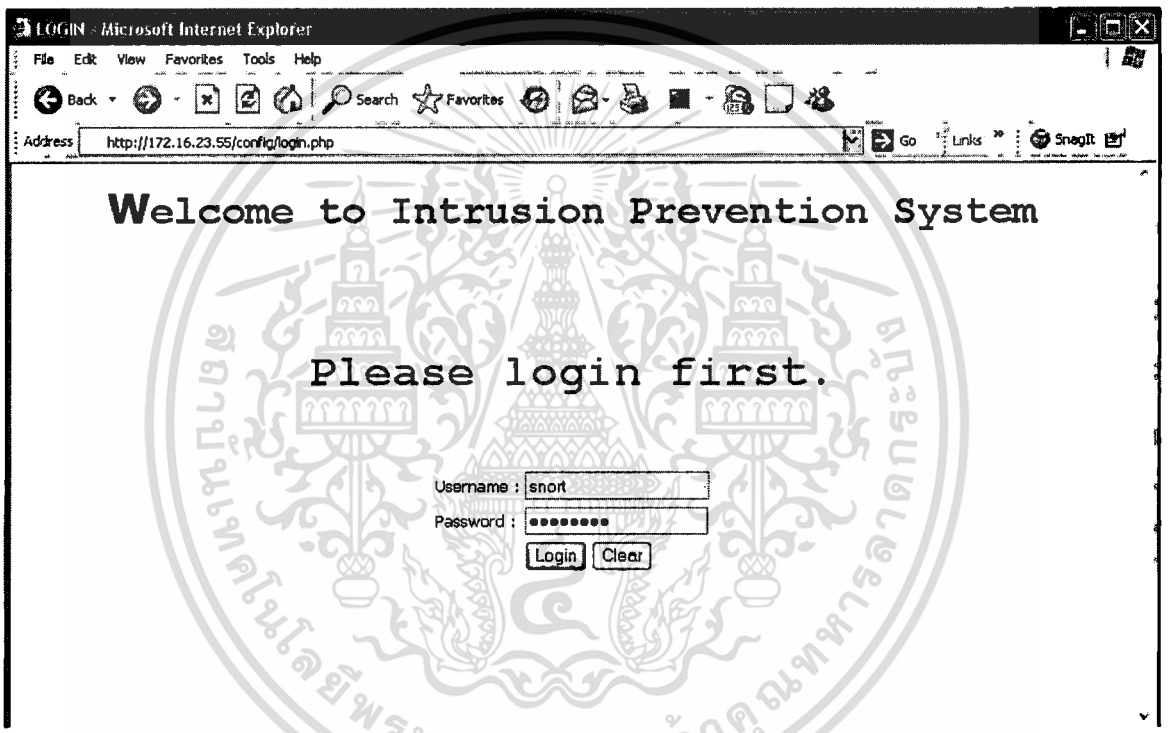
include $RULE_PATH/chat.rules

ภาคผนวก ข

คู่มือการใช้งานระบบ

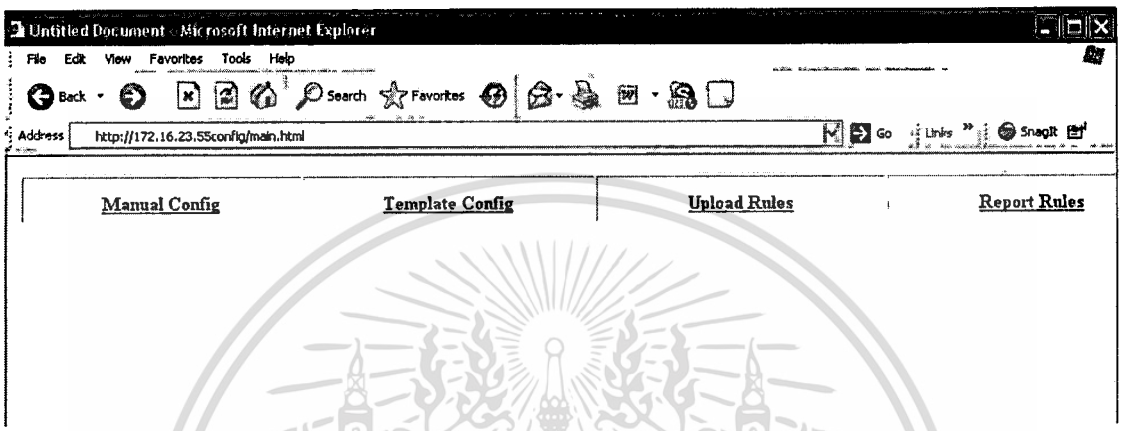
1. การใช้งานระบบผ่านทาง Web browser

- เปิด Internet Explorer แล้วพิมพ์ http://<Server IP Address>/config/login.php
- ป้อน Username:<snort> และ Password:<p@ssw0rd>
- ทำการ Login เข้าสู่ระบบ ดังภาพด้านล่าง



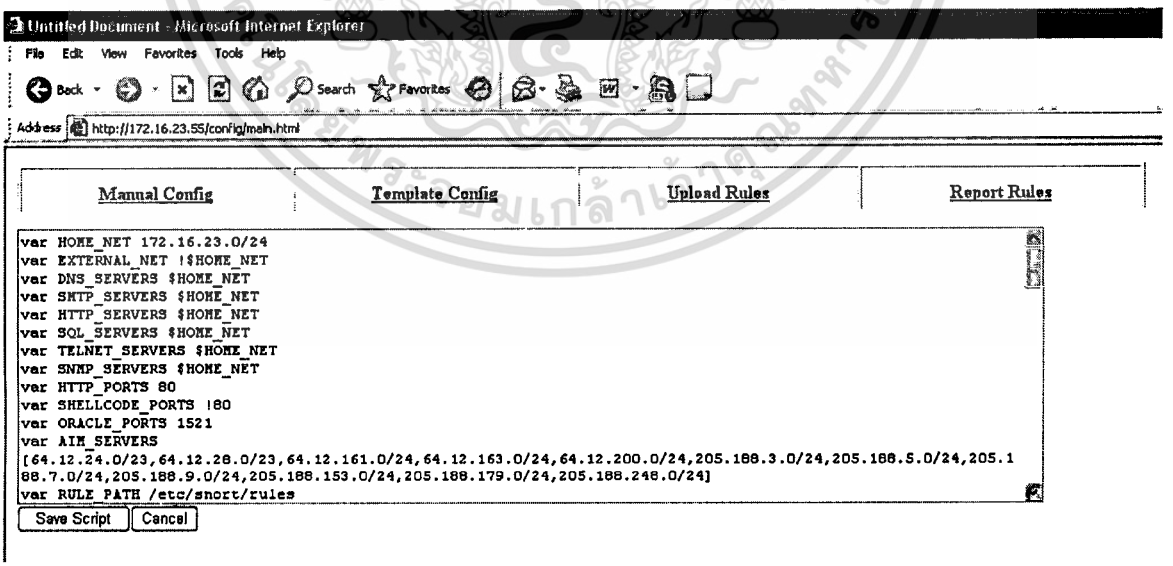
2. เมนูการทำงานในระบบ

- Manual Config เป็นเมนูที่ไว้ทำการปรับแต่งค่าไฟล์ Snort.conf
- Template Config เป็นเมนูที่ไว้ทำการปรับแต่งค่าไฟล์ Snort.conf
- Upload Rules เป็นเมนูที่ไว้สำหรับ Upload กฎของ SNORT
- Report Rules เป็นเมนูไว้สำหรับดูรายงานสถานะของกฎที่สร้างขึ้นที่ไฟร์วอลล์



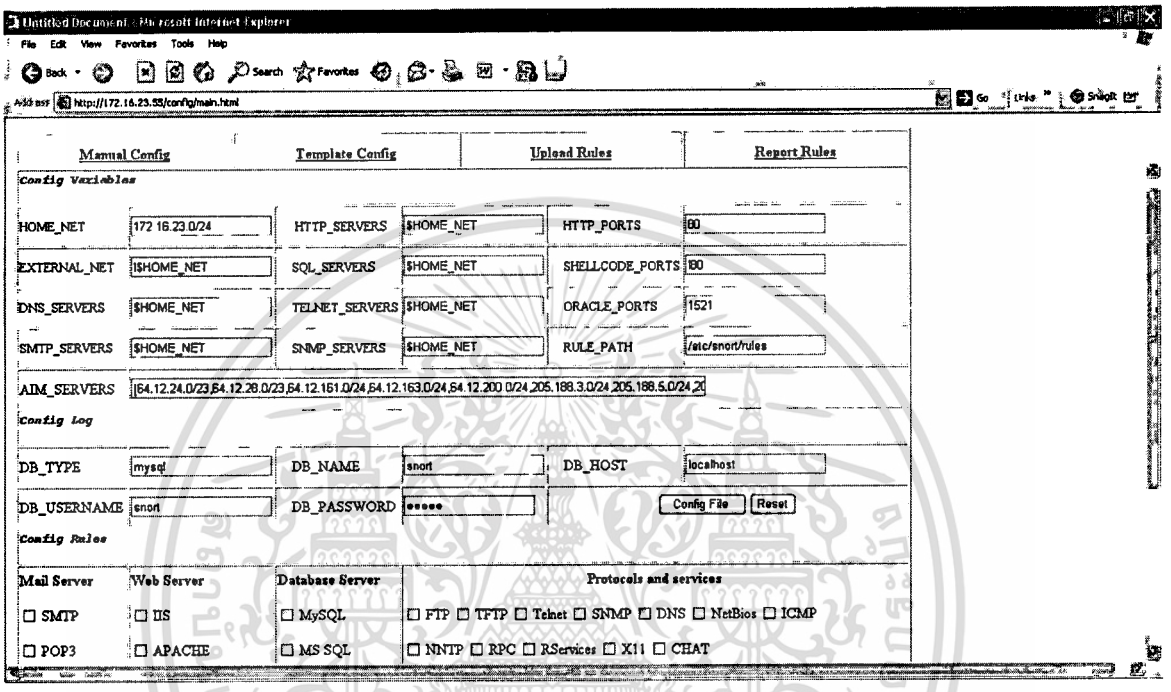
3. เมนู Manual Config

- สามารถทำการปรับแต่งค่าไฟล์ Snort.conf ผ่านทางเว็บเพจได้ โดยจะเป็นการแก้ไขปรับแต่งค่าไฟล์เอง เมื่อแก้ไขเสร็จก็ทำการบันทึกได้



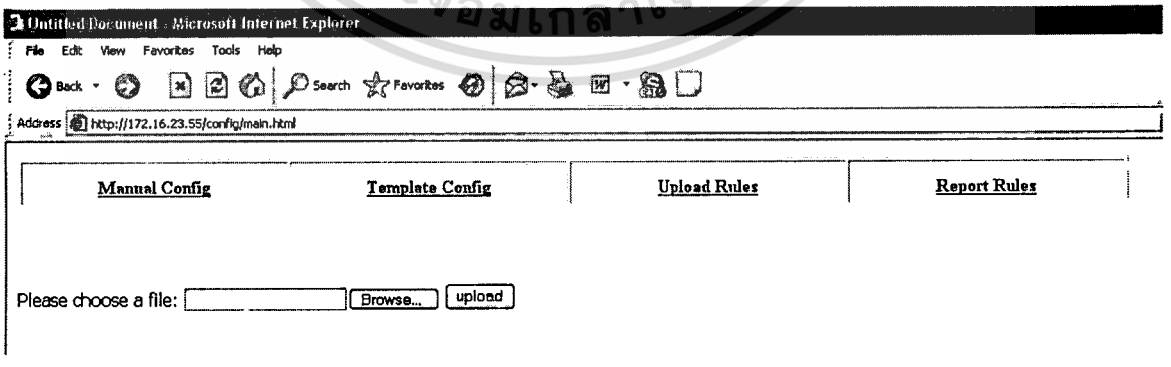
4. เมนู Template Config

- สามารถทำการปรับแต่งค่าไฟล์ Snort.conf ได้ผ่านทางหน้าเว็บเพจ โดยมีความง่ายกว่าการปรับแต่งค่าแบบ Manual Config เนื่องจากมีการทำเป็นตัวเลือกง่ายๆให้ทำการระบุค่าที่ต้องการลงไปหรือทำการเลือกเองได้ โดยเมื่อทำการบันทึกระบบจะไปทำการปรับแต่งค่าไฟล์ Snort.conf ให้เอง



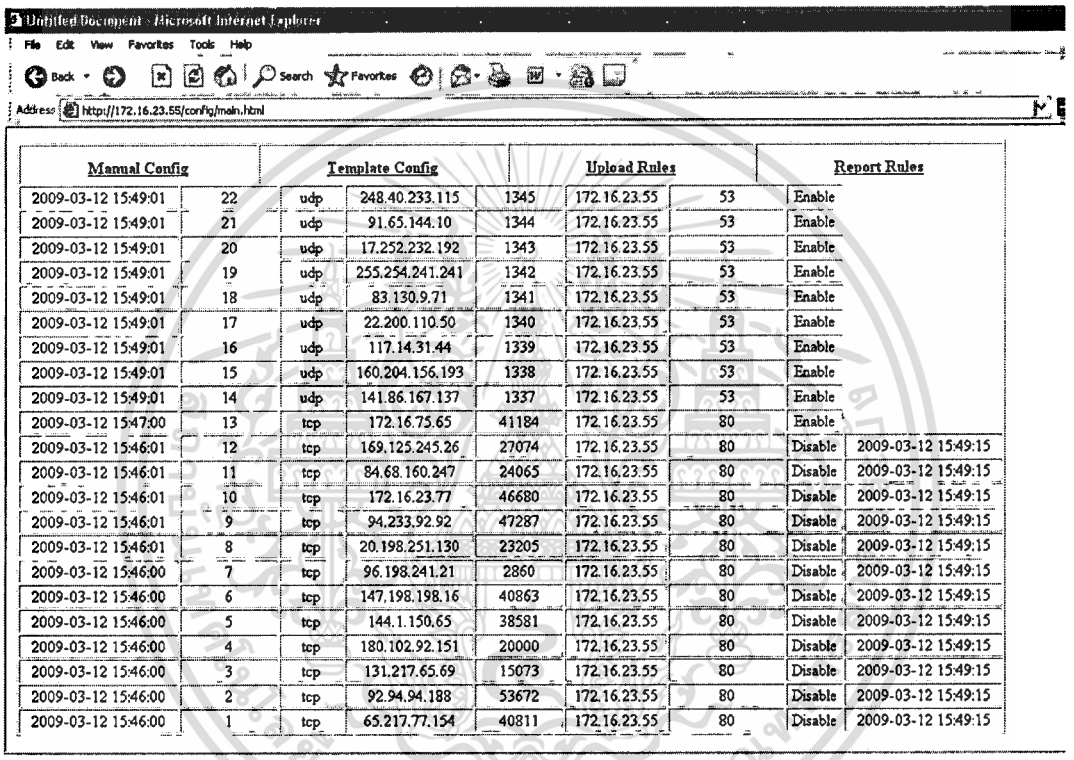
5. เมนู Upload Rules

- สามารถทำการ Upload กฎให้กับโปรแกรม SNORT ได้ผ่านทางช่องทางนี้



6. เมนู Report Rules

- เป็นเมนูที่แสดงรายการของกฎที่ได้มีการสร้างไว้ที่ไฟร์วอลล์ (IPFW) และกฎที่หมดอายุและได้ถูกลบออกจากไฟร์วอลล์ไปแล้ว โดยสามารถดูรายละเอียดได้เช่น สถานะ Enable เป็นสถานะที่กฎนี้ได้ถูกสร้างและยังใช้งานอยู่ที่ไฟร์วอลล์ (IPFW) ส่วนสถานะ Disable เป็นสถานะที่กฎนี้ได้ถูกลบออกจากไฟร์วอลล์ (IPFW)แล้ว เนื่องจากครบกำหนดเวลา ซึ่งการมีเมนูนี้ทำให้ง่ายในการมอนิเตอร์สถานะได้ง่ายขึ้น



Manual Config		Template Config			Upload Rules		Report Rules	
2009-03-12 15:49:01	22	udp	248.40.233.115	1345	172.16.23.55	53	Enable	
2009-03-12 15:49:01	21	udp	91.65.144.10	1344	172.16.23.55	53	Enable	
2009-03-12 15:49:01	20	udp	17.252.232.192	1343	172.16.23.55	53	Enable	
2009-03-12 15:49:01	19	udp	255.254.241.241	1342	172.16.23.55	53	Enable	
2009-03-12 15:49:01	18	udp	83.130.9.71	1341	172.16.23.55	53	Enable	
2009-03-12 15:49:01	17	udp	22.200.110.50	1340	172.16.23.55	53	Enable	
2009-03-12 15:49:01	16	udp	117.14.31.44	1339	172.16.23.55	53	Enable	
2009-03-12 15:49:01	15	udp	160.204.156.193	1338	172.16.23.55	53	Enable	
2009-03-12 15:49:01	14	udp	141.86.167.137	1337	172.16.23.55	53	Enable	
2009-03-12 15:47:00	13	tcp	172.16.75.65	41184	172.16.23.55	80	Enable	
2009-03-12 15:46:01	12	tcp	169.125.245.26	27074	172.16.23.55	80	Disable	2009-03-12 15:49:15
2009-03-12 15:46:01	11	tcp	84.68.160.247	24065	172.16.23.55	80	Disable	2009-03-12 15:49:15
2009-03-12 15:46:01	10	tcp	172.16.23.77	46680	172.16.23.55	80	Disable	2009-03-12 15:49:15
2009-03-12 15:46:01	9	tcp	94.233.92.92	47287	172.16.23.55	80	Disable	2009-03-12 15:49:15
2009-03-12 15:46:01	8	tcp	20.198.251.130	23205	172.16.23.55	80	Disable	2009-03-12 15:49:15
2009-03-12 15:46:00	7	tcp	96.198.241.21	2860	172.16.23.55	80	Disable	2009-03-12 15:49:15
2009-03-12 15:46:00	6	tcp	147.198.198.16	40863	172.16.23.55	80	Disable	2009-03-12 15:49:15
2009-03-12 15:46:00	5	tcp	144.1.150.65	38581	172.16.23.55	80	Disable	2009-03-12 15:49:15
2009-03-12 15:46:00	4	tcp	180.102.92.151	20000	172.16.23.55	80	Disable	2009-03-12 15:49:15
2009-03-12 15:46:00	3	tcp	131.217.65.69	15073	172.16.23.55	80	Disable	2009-03-12 15:49:15
2009-03-12 15:46:00	2	tcp	92.94.94.188	53672	172.16.23.55	80	Disable	2009-03-12 15:49:15
2009-03-12 15:46:00	1	tcp	65.217.77.154	40811	172.16.23.55	80	Disable	2009-03-12 15:49:15

ประวัติผู้เขียน

ชื่อผู้เขียน	ดุษกร อรรจน์อังกูร
วัน เดือน ปีเกิด	30 พฤศจิกายน 2525
สถานที่เกิด	กรุงเทพมหานคร
วุฒิการศึกษาระดับปริญญาตรี	ทลบ.(เทคโนโลยีสารสนเทศ)
สถานที่สำเร็จการศึกษา	คณะเทคโนโลยีและการจัดการอุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ ปราจีนบุรี
ปีการศึกษาที่สำเร็จการศึกษา	2547