ห้องสมุดคณะเทคโนโลยีสารสนเทศ พระจอมเกล้าลาดกระบัง

# เครื่องมือบริหารระบบป้องการบุกรุกเครือข่ายผ่านเว็บ

# WEB-BASED IPS ADMINISTRATION TOOL

โดย

ธเนศ ไพรินทราภา

THANEATH PIRINTRAPHA

H005995

อาจารย์ที่ปรึกษา

ผศ.ดร.จันทร์บูรณ์ สถิตวิริยวงศ์

ว.ท.
ธ285ธ
2551

เลขหมู่.....
เลขทะเบียน 05995
วัน,เดือน,ปี 15 ก.พ. 2553

b.12172832
i.....

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ภาคเรียนที่ 2 ปีการศึกษา 2551

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# WEB-BASED IPS ADMINISTRATION TOOL

THANEATH PIRINTRAPHA

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**COPYRIGHT 2009**

**FACULTY OF INFORMATION TECHNOLOGY**

**KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|---|---|
| **หัวข้อ** | เครื่องมือบริหารระบบป้องการบุกรุกเครือข่ายผ่านเว็บ |
| **นักศึกษา** | นายธเนศ ไพรินทราภา |
| **รหัสนักศึกษา** | 48066710 |
| **ปริญญา** | วิทยาศาสตร์มหาบัณฑิต |
| **สาขาวิชา** | เทคโนโลยีสารสนเทศ |
| **แขนงวิชา** | วิทยาการสารสนเทศ |
| **ปีการศึกษา** | 2551 |
| **อาจารย์ที่ปรึกษา** | ผศ.ดร.จันทร์บูรณ์ สถิตวิริยวงศ์ |

## บทคัดย่อ

ระบบป้องกันการบุกรุก (Intrusion Prevention System : IPS) เป็นสิ่งที่ช่วยเพิ่มความสามารถในการป้องกันและตรวจจับภัยคุกคามและผู้บุกรุกในรูปแบบต่างๆ ทั้งจากภายในและภายนอกองค์กร ซึ่งก็มีทั้งซอฟต์แวร์และฮาร์ดแวร์ที่ราคาค่อนข้างสูง แต่ก็มีซอฟต์แวร์ที่เป็น Open Source อยู่ที่ทำงานในลักษณะนี้และเป็นที่นิยมมากตัวหนึ่งนั่นก็คือโปรแกรม Snort (Intrusion detection system : IDS) และแต่เดิมในการทำงานของมันเพียงสามารถแจ้งเตือนได้อย่างเดียวแต่ไม่สามารถป้องกันได้และในวิทยานิพนธ์ฉบับนี้จะนำเสนอรูปแบบของการตรวจจับการบุกรุกพร้อมกับสามารถป้องกันการบุกรุกนั้นได้โดยใช้ซอฟต์แวร์ตัวเดิมเพียงแต่เปลี่ยนโหมดการทำงานของมันที่เรียกว่า Snort inline ร่วมกับใช้ซอฟต์แวร์อีกตัวเพื่อนำเสนอรูปแบบของการป้องกันการบุกรุกนั่นก็คือโปรแกรม Iptable นั่นเองและทั้ง 2 โปรแกรมนี้มันทำงานอยู่บน linux platform ซึ่งในการติดตั้งหรือปรับแต่งนั้นยังไม่ค่อยมีความสะดวกและเกี่ยวกับกฎการตรวจจับและป้องกันซึ่งก็ถือเป็นหัวใจหลักของ snort (snort rules) อีกอย่างหนึ่งซึ่งจะทำงานแบบ command line จึงเกิดแนวคิดในการพัฒนาเครื่องมือเพื่อช่วยอำนวยความสะดวกให้กับผู้ดูแลระบบโดยให้สามารถปรับแต่งโปรแกรมหรือจัดการกฎที่ใช้ในการควบคุมผ่านทางหน้าเว็บเพจได้ในระดับหนึ่งทำให้สามารถเข้าถึงได้ง่ายกว่าแบบเดิม ที่เป็นอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|---|---|
| **Title** | Web-based IPS Administration Tool |
| **Student** | Mr. Thaneath Pirintrapha |
| **Student ID.** | 48066710 |
| **Degree** | Master of Science in Information Technology |
| **Programme** | Information Science |
| **Academic Year** | 2008 |
| **Advisor** | Asst. Prof. Chanboon Sathiwiriyawong, Ph.D. |

# ABSTRACT

Intrusion Prevention System, or IPS, is the capability supporter for prevention and detection of threaten danger and intruder in many aspects from inside or outside the organization, but the drawback is the high cost of software and hardware. Currently, there is Open Source software that popular called Snort program (Intrusion detection system : IDS). Formerly, this program was designed for warning however it lacks the ability of prevention. This project will be presented the configuration of intrusion detection including prevention performance by using the old program. By changing mode of operation called Snort inline, this program works with Iptable program in order to add function of intrusion prevention. Both programs are able to work on linux platform. The obstacles are inconvenience of the installation and modification, moreover, there is the problem related to detection and prevention rules which are another key role of snort rules working by means of command line. That point leads to the development of the system to support the administrators by offering the abilities of program modification or handling the rules by controlling via web page in some levels. This method makes things more convenient than ever.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# กิตติกรรมประกาศ

ผู้จัดทำขอขอบพระคุณ ผศ.ดร.จันทร์บูรณ์ สถิตวิริยวงศ์ ซึ่งได้ให้คำปรึกษาทั้งแนวคิดและข้อเสนอแนะต่างๆ และขอขอบใจเพื่อนๆที่คอยกระตุ้นและให้กำลังใจและความช่วยเหลือมาโดยตลอด

สุดท้ายนี้ขอกราบขอบพระคุณบิดา มารดา ผู้ที่ให้การสนับสนุนอย่างเต็มที่ตลอดมาและขอบคุณพระเจ้าผู้ทรงมอบพลัง สติปัญญาในการทำงาน และชี้ทางสว่างในการนำทางชีวิต

ธเนศ ไพรินทราภา

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

หน้า




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ (ต่อ)

หน้า




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ (ต่อ)

หน้า






เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญรูป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญรูป (ต่อ)

รูปที่ หน้า



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1
# บทนำ

## 1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันเทคโนโลยีสารสนเทศที่คนรู้จักและใช้งานมากที่สุดคืออินเทอร์เน็ต(Internet) เนื่องจากสามารถค้นคว้าหาความรู้ได้มากมายจากทั่วทุกมุมโลก ไม่ว่าจะเป็นหน่วยงานหรือองค์กร หรือพวกเราที่เป็นนักศึกษาต่างก็ต้องอาศัยเทคโนโลยีสารสนเทศเข้ามาเป็นส่วนหนึ่งเพื่อเพิ่มทักษะและพัฒนาศักยภาพในการศึกษาของตัวเรา รวมถึงการค้นคว้าหาข้อมูล ซึ่งเหล่านี้ทำให้เกิดความสะดวกในการแลกเปลี่ยนและสืบค้นได้อย่างอิสระ อย่างไรก็ตามสิ่งเหล่านี้เปรียบเสมือนประตูที่เปิดให้ผู้ไม่พึงประสงค์รุกล้ำเข้าไปยังคอมพิวเตอร์ต่างๆ ที่เชื่อมต่อมกับอินเทอร์เน็ตได้แม้จะมาได้รับอนุญาตก็ตาม และเมื่อผู้บุกรุกเหลานี้เข้ามาในระบบได้แล้ว ก็สามารถที่จะ ทำลาย ขโมยหรือเปลี่ยนแปลงข้อมูลสำคัญทั้งที่ตั้งใจและไม่ตั้งใจจึงเป็นสาเหตุให้ผู้ดูแลระบบต้องเสียเวลาอย่างมากในการค้นหาหนทางเพื่อป้องกันการบุกรุกของผู้บุกรุกเหล่านี้

และเนื่องจากโครงสร้างหลักของเครือข่ายมีขนาดใหญ่ ปัญหาที่พบมากที่สุดคือการแพร่ระบาดของไวรัสอย่างรวดเร็วและการโดนผู้ไม่ประสงค์ดีทำการโจมตีระบบเพื่อให้หยุดการทำงานอยู่เสมอ ทำให้การจัดการกับปัญหาเป็นไปด้วยความลำบากอย่างยิ่ง

จากการศึกษาข้อมูลเพิ่มเติมพบว่าองค์กรขนาดใหญ่มักมีระบบ ป้องกันและตรวจจับการบุกรุกเครือข่ายโดยมากจะเป็นฮาร์ดแวร์ที่มีราคาแพงมาก ซึ่งทางหน่วยงานหรือองค์กรอาจไม่มีงบประมาณเพียงพอในการจัดซื้อ ดังนั้นจึงได้เกิดแนวคิดในการนำระบบป้องการบุกรุกเครือข่ายโดยใช้ open source (Free Software) นั่นคือ สนอร์ทอินไลน์ (IPS) เป็นซอฟต์แวร์ระบบป้องกันการบุกรุกที่เป็นที่ยอมรับและใช้กันอย่างแพร่หลาย มีกฎมากมายในการป้องกันและตรวจจับผู้บุกรุกรวมถึงลักษณะและสภาพของการใช้งานยังไม่ค่อยสะดวกเนื่องจากเป็นซอฟต์แวร์ที่อยู่บน linux platform ซึ่งมีคำสั่งเป็นแบบ command line (text mode) และในปัจจุบันกฎของ snort นั้นเป็นเท็กซ์ไฟล์ทำให้การเพิ่ม ลด กฎทำได้ไม่สะดวก โครงการพัฒนาระบบนี้จึงสร้างเครื่องมือเพื่อช่วยผู้ดูแลระบบให้สามารถใช้งานได้สะดวกขึ้นในรูปแบบ Web base ทำให้สามารถจัดการได้จากทุกที่โดยผ่านทาง Internet หรือ Intranet ภายในองค์กร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

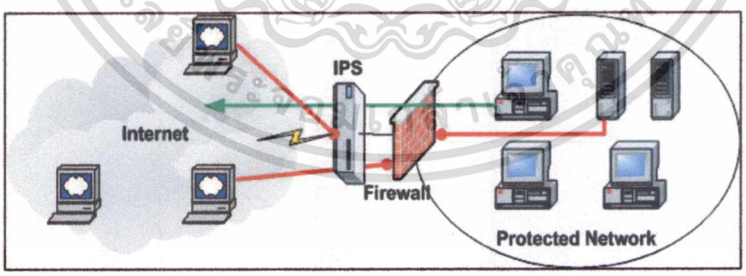
## 1.2 ขอบเขตของโครงการพัฒนาระบบงาน

ในโครงการพัฒนาระบบจะเป็นการพัฒนาเว็บเบสบนระบบปฏิบัติการลินุกซ์ โดยใช้ภาษา PHP และ PERL-CGI ในการเขียนเว็บแอพพลิเคชั่น ซึ่งระบบที่พัฒนาขึ้นจะทำงานร่วมกับ Snort และ Iptable เพื่อนำเสนอการทำงานแบบ IPS โดยจะแก้ไขคอนฟิกไฟล์เฉพาะโปรแกรม Snort เท่านั้นและอ่าน rules fileทั้งหมดของโปรแกรม Snort มาเก็บไว้ในฐานข้อมูลของเว็บแอพลิเคชั่นโดยที่จะนำเสนอในรูปแบบเว็บเบสและระบบที่จะพัฒนานั้นเป็นเว็บแอพลิเคชั่นที่ทำขึ้นแยกต่างหาก ซึ่งไม่มีการปรับแต่งซอร์สโค้ดของ Snort และ IPtables แต่อย่างใด

## 1.3 ทฤษฎีหรือแนวคิดที่ใช้

**- ระบบป้องกันการบุกรุกเครือข่าย (Intrusion Prevention System)**

ระบบป้องกันการบุกรุกเครือข่ายเป็นสิ่งที่ออกแบบเพื่อใช้จับการโจมตีที่ซ่อนแฝงมาและมีมาตรการต่อต้านที่อิสระเพื่อยับยั้งการโจมตีเหล่านั้น โดยทั่วไปแล้ว IPS จะอยู่ตรงในตำแหน่งที่เส้นทางที่แพ็กเกตทำการเดินทางในเครือข่าย (inline เน็ตเวิร์ค) และ เฝ้าดู เมื่อมีเหตุการณ์เกิดขึ้นมันจะทำการอ้างถึงกฎเงื่อนไขที่เราตั้งไว้ สิ่งนี้ไม่เหมือนกับ IDS ซึ่งมันจะไม่ได้ทำงานอยู่ในตำแหน่ง Inline ดังรูปที่ 1 แต่จะเป็นในลักษณะแบบ Passive ซึ่งแบ่งเป็น 2 ประเภทคือ HIPS , NIPS (ในรายงานฉบับนี้สนใจเฉพาะ NIPS)




**รูปที่ 1.1** Inline Network IPS

**Network Intrusion prevention systems (NIPS)**

เป็นระบบที่จะวิเคราะห์ ดักฟังข้อมูลบนเครือข่ายทั้งหมดและเฝ้าดูว่ามันมีเหตุการณ์หรือการเคลื่อนไหวอะไรที่น่าสงสัย ตรวจจับและรายงานโดยสามารถ drop ทราฟฟิกที่ปองร้ายและในตำแหน่งที่มันอยู่นั้นจะเรียกว่า inline network มันจึงทำการป้องกันการโจมตีได้อย่างทันท่วงทีในส่วนที่เพิ่มเติมคือจะสามารถมองหา (ถอดรหัส) ในเลเยอร์ที่ 7 (โพรโทคอล) ได้เช่น HTTP,FTP,SMTP และ ตรวจสอบการบุกด้วยเครื่องบ่งชี้ (scan intrusion signature) ค้นหาความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผิดปกติของโพรโทคอล (search protocol anomalies) และตรวจจับ command ที่ไม่ปกติที่จะ executed บนเครือข่าย

## 1.4 ประโยชน์ที่คาดว่าจะได้รับ

- เข้าใจถึงหลักการทำงานของโปรแกรมตรวจจับการบุกรุก Snort และ Iptables ทำให้สามารถประยุกต์นำมาใช้งานรักษาความปลอดภัยในเครือข่ายได้
- มีความเข้าใจในการโปรแกรมด้วยภาษา PHP และ PERL-CGI บนระบบลินุกซ์ และสามารถนำความรู้ไปใช้พัฒนาโปรแกรมอื่นๆ ได้
- ระบบที่สร้างสามารถเข้าถึงได้โดยง่ายที่ผ่านทางหน้าเว็บแอพพลิเคชั่น เป็นทางเลือกให้สำหรับผู้ดูแลระบบในการนำไปใช้งานที่หน่วยงานหรือองค์กรนั้นๆ ให้มีระบบป้องกันการบุกรุกได้โดยที่ไม่ต้องเสียเงินค่าลิขสิทธิ์ในการจัดซื้ออุปกรณ์ฮาร์ดแวร์ราคาแพง

## 1.5 การเปรียบเทียบระหว่างวิธีการที่นำเสนอกับวิธีการแบบพื้นฐาน

ก่อนที่จะมาเป็นระบบป้องกันการบุกรุกนั้น(IPS) แต่เดิมจะเป็นระบบตรวจจับการบุกรุก (IDS) หรือถ้ากล่าวอีกนัยหนึ่งก็คือ IPS เป็นขั้นถัดไปของระบบตรวจจับการบุกรุกนั่นเอง จากIDS ที่เคยตรวจจับหรือแจ้งเตือนให้ผู้ดูแลระบบได้รับทราบเท่านั้นก็สามารถป้องกันการบุกรุกจากการตรวจจับดังกล่าวได้ Snort (IDS,Snort-inline) + firewall (iptable) = IPS ซึ่งแสดงการเปรียบเทียบได้ดังตารางที่ 1.1

**ตารางที่ 1.1** การเปรียบเทียบระหว่าง IDS กับ IPS

| IDS | IPS |
|---|---|
| ติดตั้งในส่วนต่างๆ ของเครือข่าย | ติดตั้งในส่วนต่างๆ ของเครือข่าย |
| อยู่ตำแหน่งที่เป็นทางผ่านของเครือข่าย | อยู่ในตำแหน่งที่เป็นเส้นทางการเดินทางของข้อมูลในเครือข่าย |
| ไม่สามารถแยก traffic ที่เข้ารหัสได้ | มีการป้องกันได้ในระดับแอพลิเคชั่นที่ดีกว่า |
| Central Management control | Central Management control |
| มีการตรวจจับผู้โจมตีที่ดี | มีเป้าหมายเพื่อหยุด web defacement |
| Alert product (reactive) | Blocking product (proactive) |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.6 ขั้นตอนการดำเนินงาน

ในโครงการนี้ได้แบ่งขั้นตอนในการศึกษาและพัฒนาเว็บแอพพลิเคชั่นดังนี้

ขั้นตอนที่ 1 ศึกษาหลักการทำงานในเชิงเปรียบเทียบของ Intrusion Detection Systems กับ Intrusion Prevention Systems (next generation IDS) รวมถึงซอฟต์แวร์ที่ทำงานในทั้ง 2 ลักษณะคือโปรแกรม Snort เกี่ยวกับรูปแบบคำสั่งและการทำงานต่างๆ และในส่วนการทำงานของคำสั่งที่ใช้ควบคุม IPtables

ขั้นตอนที่ 2 ศึกษาการเขียนเว็บด้วยภาษา PHP และ PERL-CGI และเครื่องมือที่ใช้ในการพัฒนาเว็บ

ขั้นตอนที่ 3 วิเคราะห์และออกแบบระบบตามขอบเขตของโครงการ

ขั้นตอนที่ 4 พัฒนาตังเว็บแอพพลิเคชั่นและทดสอบเพื่อปรับปรุง

ขั้นตอนที่ 5 สรุปผลการใช้งานระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 2

# หลักการระบบป้องกันการบุกรุก

ในยุคที่เครือข่ายอินเทอร์เน็ตเป็นเสมือนปัจจัยพื้นฐานในการสื่อสารข้อมูลขององค์กรกับภายนอกองค์กร ซึ่งในเครือข่ายขนาดใหญ่นี้ มีการเชื่อมโยงระบบเครือข่ายทั่วโลกเข้าหากันจนเป็นเครือข่ายคอมพิวเตอร์ขนาดใหญ่ที่สุดในโลกมีคนใช้บริการนับล้านคนเพื่อติดต่อสื่อสารกัน โอกาสของผู้ไม่ประสงค์ดีที่เข้ามารบกวนหรือทำลายระบบเครือข่ายภายในขององค์กรมีอยู่ตลอดเวลา ซึ่งนับเป็นปัญหาที่มีผลกระทบกับการทำงานของหลายๆองค์กรในปัจจุบัน

## 2.1 ความสำคัญของการรักษาความปลอดภัยสำหรับระบบเครือข่ายคอมพิวเตอร์

แนวโน้มปัญหาเรื่องความปลอดภัยในระบบเครือข่ายขององค์กรที่ต่อเชื่อมกับเครือข่ายอินเทอร์เน็ตปัจจุบันมีความรุนแรงและขยายตัวไปอย่างรวดเร็ว ซึ่งองค์กรใดก็ตามที่ไม่เตรียมเครื่องมือหรือวิธีการไว้รองรับจะเผชิญกับความเสียหายอย่างหลีกเลี่ยงไม่ได้ สาเหตุสำคัญประการหนึ่งของการความเสียหายที่แผ่ขยายออกไปเป็นวงกว้างในเวลาที่รวดเร็วขึ้นเนื่องจากปัจจุบันมีผู้นิยมเชื่อมต่อกับเครือข่ายอินเทอร์เน็ตแบบความเร็วสูง (Broadband) ทำให้ปริมาณข้อมูลที่วิ่งบนเครือข่ายอินเทอร์เน็ตมีจำนวนเพิ่มมากขึ้น ตัวอย่างความเสียหายที่รู้จักกันดีคือไวรัสคอมพิวเตอร์ประเภทหนอนอินเทอร์เน็ต (Worm) ที่สามารถแพร่กระจายผ่านจดหมายอิเล็กทรอนิกส์ (E-mail) ได้อย่างรวดเร็วอาทิเช่น Code Red, NIMDA, SQL Slammer, MSBlaster, และ Sasser worm ซึ่งได้สร้างความเสียหายต่อองค์กรหลายแห่งอย่างมากมาย

## 2.2. การแบ่งประเภทของผู้บุกรุก

Hacker และ Cracker เป็นคำที่ใช้แทนผู้บุกรุกซึ่ง Hacker คือบุคคลที่ชอบเจาะเข้าสู่ระบบต่างๆ Hacker ที่ดีคือบุคคลที่จะเข้าสู่ระบบคอมพิวเตอร์ของตัวเอง เพื่อที่จะเข้าใจการทำงานต่างๆ ของระบบแต่ Hacker ที่เป็นผู้ร้ายหรือ Cracker คือบุคคลที่พยายามเจาะเข้าสู่ระบบของผู้อื่นโดยมีเจตนาเพื่อทำรายระบบให้เกิดความเสียหายสำหรับบุคคลหรือสิ่งอื่นใดก็ตามที่เข้าสู่ระบบโดยไม่ได้รับอนุญาต แล้วกระทำการใดๆที่อาจก่อให้เกิดความเสียหายแก่ระบบ เราจะเรียกว่า Intruder (ผู้บุกรุก) ซึ่งผู้บุกรุกแบ่งออกเป็น 2 ประเภทคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.2.1 Outsides** คือ ผู้บุกรุกจากภายนอกเครือข่ายและบุคคลที่อาจจะโจมตีมาจากภายนอก เช่นการเปลี่ยนแปลงหน้า web server หรือการส่งต่อเมล์ผ่านทาง e-mail server ซึ่งการบุกรุกจากภายนอกอาจมาจากอินเทอร์เน็ต, การ Diaul-modem , walk in เป็นต้น

**2.2.2 Insider** คือ ผู้บุกรุกที่มีสิทธิ์ในการใช้เครือข่ายภายใน รวมทั้งผู้ใช้สิทธิ์ในทางที่ผิดหรือลักลอบใช้สิทธิ์ของผู้อื่นๆ ที่มีสิทธิ์เหนือกว่า

## 2.3 ประเด็นที่เกี่ยวข้องกับเรื่องของความปลอดภัยและประเภทของการละเมิดบนเครือข่าย

เหตุการณ์ที่เกี่ยวข้องกับเรื่องของความปลอดภัยในระบบเครือข่ายมักจะเกี่ยวข้องกับกิจกรรมที่ก่อให้เกิดผลในทางลบต่อระบบ โดยทั่วไปแล้วมักจะเป็นเรื่องของการละเมิดนโยบายในด้านการรักษาความปลอดภัยที่ได้กำหนดไว้แล้วขององค์กร ซึ่งอาจเกิดจากคนในองค์กร หรือมาจากเครือข่ายภายนอกองค์กรอย่างเช่น อินเทอร์เน็ต การโจมตีบางครั้งมุ่งไปที่ระบบบางระบบโดยเฉพาะ และบางครั้งก็ต้องอาศัยบัญชีรายชื่อพิเศษจากระบบ เช่น ของผู้ดูแลระบบนั้นๆ

ซึ่งรูปแบบในการโจมตีบางครั้ง มุ่งไปที่บัญชีรายชื่อบางอันของระบบ หรือของผู้ดูแลระบบเพื่อให้ได้สิทธิ์ในการจัดการกับระบบ หรืออาจใช้ระบบที่ได้ตกเป็นเหยื่อแล้วเพื่อมุ่งโจมตีไปยังระบบอื่นๆ โดยทั่วไปสามารถทำได้ในเวลาแค่ 45 วินาที ซึ่งในอนาคตอาจทำได้เร็วขึ้นกว่านี้อีก

แหล่งที่มักจะเข้าโจมตีระบบ บางครั้งการจะระบุว่าเป็นใครที่เข้ามาบุกรุกระบบค่อนข้างทำได้ยาก พวกเค้าเหล่านั้นอาจเป็นนักศึกษาที่อยากรู้อยากเห็น โดยใช้อินเทอร์เน็ตเป็นเครื่องมือหรืออาจเป็นบุคคลที่ต้องการข้อมูลเพื่อเป็นประโยชน์ต่อการแข่งขันในด้านธุรกิจ หรืออาจเป็นพนักงานภายในองค์กรนั้นๆ เอง โดยส่วนใหญ่แล้วมักจะโจมตีจากช่องโหว่หรือช่องโหว่จากการปรับแต่งระบบที่ผิดพลาด ผู้บุกรุกที่ประสบความสำเร็จในการโจมตีระบบหลายๆ ครั้งจะยิ่งสร้างความเสียหายเพิ่มมากขึ้นเรื่อยๆในครั้งถัดไป

เหตุการณ์ที่ใช้ในการโจมตี สามารถแบ่งเป็นประเภทต่างๆได้ดังต่อไปนี้ probe, scan, account compromise, root compromise, packet sniffer, denial of service, exploitation of trust, malicious code, และ Internet infrastructure attacks.

**Probe (โพรบ)** เป็นลักษณะของการทดลองหรือการเดาวิธีการหรือแนวทางเพื่อหาทางเข้าสู่ระบบ ตัวอย่างเช่นมีการ พยายามเข้าสู่ระบบ จากบัญชีรายชื่อที่ไม่ได้ใช้ หรือ จากการเดาบัญชีรายชื่อที่มีอยู่ในระบบ ดังนั้นหากบัญชีผู้ใช้ในระบบมีการการกำหนดรหัสผ่านที่ง่ายต่อการเดาด้วยวิธีการ probe จะทำให้ผู้ที่โจมตีสามารถใช้ช่องโหว่นี้เข้าทำลายระบบได้ง่าย

**Scan (สแกน)** เป็นลักษณะของวิธีการ โพรบด้วยจำนวนความถี่หรือจำนวนครั้งมากๆ ซึ่งบางครั้งจะได้ผลลัพธ์ที่เป็นประโยชน์ในการที่จะใช้ในการโจมตีระบบต่อไป ยกตัวอย่างเช่น การ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทดลองสุ่มใช้รหัสผ่านจำนวนหนึ่งกับบัญชีรายชื่อผู้ใช้ในระบบ โดยรหัสผ่านที่ใช้จะเป็นรหัสผ่านที่มักเป็นที่นิยมสำหรับผู้ใช้ทั่วไปที่ไม่ได้ใส่ใจกับเรื่องของความปลอดภัยของระบบ

**Account Compromise** (การแอบเข้าไปใช้งานระบบจากบัญชีผู้ใช้คนอื่น) เป็นลักษณะเข้าไปใช้งานจากระบบโดยที่ใช้บัญชีรายชื่อของบุคคลอื่นที่อยู่ในระบบ เพื่อทำการขโมยข้อมูลหรือทำลายข้อมูล และหากผู้ใช้ที่เป็นผู้ดูแลระบบขาดความใส่ใจในเรื่องของความปลอดภัย อาจนำไปสู่ปัญหาเรื่องความปลอดภัยในระดับที่รุนแรงขึ้น

**Root Compromise** (การแอบเข้าไปใช้งานระบบจากบัญชีผู้ใช้ในระดับผู้ดูแลระบบ) มีลักษณะเหมือนกับ Account compromise แต่ระดับสิทธิ์ที่ได้มีมากกว่ากล่าวคือ ผู้บุกรุกสามารถทำได้ทุกอย่างที่ผู้ดูแลระบบปกติทำได้ ซึ่งสามารถสร้างความเสียหายได้อย่างร้ายมาก

**Packet Sniffer** (การดักจับข้อมูล) เป็นลักษณะของการบุกรุกโดยอาศัยโปรแกรมที่มีความสามารถในการดักจับข้อมูลที่ถูกส่งผ่านไปมาในระบบเครือข่าย ซึ่งข้อมูลเหล่านั้นอาจประกอบไปด้วย บัญชีรายชื่อที่อยู่ในระบบและรหัสผ่าน ซึ่งโดยมากข้อมูลในระบบมักจะเป็นข้อความธรรมดาที่ไม่ได้ทำการเข้ารหัสไว้

**Denial of Service** (การทำให้หยุดบริการ) วัตถุประสงค์ของการโจมตีในลักษณะของการให้ระบบเป้าหมายหยุดให้บริการ ต่างจากการจู่โจมแบบอื่นคือ ไม่ได้มุ่งหวังในการเข้าไปใช้งานระบบ แต่ต้องการให้ระบบนั้นหยุดทำงาน วิธีการโจมตีเพื่อให้หยุดการให้บริการสามารถทำได้หลายรูปแบบ เช่น การทำ "flood" เพื่อทำให้เครื่องเป้าหมายได้รับข้อมูลเป็นปริมาณมากๆ จนไม่สามารถตอบสนองต่อการให้บริการได้ทันจนในที่สุดหยุดให้บริการไป ความเสียหายจากการทำให้เกิดการหยุดให้บริการนั้นขึ้นอยู่กับหน้าที่ของระบบนั้นๆ ว่ามีความสำคัญอย่างไร เช่นถ้าระบบนั้นต้องรองรับการให้บริการการทำธุรกรรมผ่านทางอินเทอร์เน็ต มูลค่าความเสียหายก็จะมากกว่าระบบที่ให้บริการข่าวสารทั่วๆไป

**Exploitation of Trust** (การอาศัยช่องโหว่จากความเชื่อถือจากเครื่องที่ใช้งานร่วมกัน) การสื่อสารระหว่างคอมพิวเตอร์ที่อยู่ในเครือข่ายที่อาศัยความเชื่อถือกันว่าถ้าเป็นเครื่องที่ตกลงกันไว้เข้ามาขอใช้งานระบบ ก็จะอนุญาตให้ดำเนินการได้ แต่ผู้บุกรุกสามารถปลอมแปลงเครื่องให้มีคุณลักษณะที่เป็นเครื่องที่ได้สร้างข้อตกลงกันไว้เพื่อแอบเข้าไปใช้งานในระบบได้

**Malicious Code** (โปรแกรมที่ประสงค์ร้าย) มักจะเป็นโปรแกรมที่เราไม่อาจคาดเดาผลลัพธ์ที่ได้จากการใช้งานบนระบบ ผู้ใช้งานโดยทั่วไปแล้วไม่ได้มีความระมัดระวังเกี่ยวกับการใช้งานโปรแกรมเท่าที่ควรบางครั้งสั่งให้โปรแกรมที่ไม่รู้จักทำงานและกว่าจะพบว่าเป็นโปรแกรมที่มุ่งประสงค์ร้ายก็มักจะเกิดความเสียหายขึ้นแล้ว โปรแกรมที่ประสงค์ร้ายประกอบไปด้วยโปรแกรมแบบม้าโทรจัน (Trojan horses) โปรแกรมไวรัส (Viruses) โปรแกรมหนอน (Worms) โปรแกรมแบบม้าโทรจันและไวรัสส่วนมากจะทำการซ่อนตัวเองอยู่ในโปรแกรมอื่นๆ อีกทีหนึ่ง หรืออาจเป็นไฟล์ที่ผู้โจมตีได้เปลี่ยนแปลงแก้ไขการทำงานภายในเรียบร้อยแล้ว โปรแกรมหนอนเป็นโปรแกรมที่สามารถทำงานได้ด้วยตัวเองจากเครื่องหนึ่งไปยังอีกเครื่องหนึ่งในระบบเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้นำไปดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

โดยไม่ต้องอาศัยให้ผู้ใช้ระบบเป็นคนสั่งเหมือน โปรแกรมไวรัสหรือม้าโทรจัน โปรแกรมทั้งหมดเหล่านี้สามารถสร้างความเสียหายให้เกิดขึ้นเป็นอย่างมาก เช่นมีการสูญเสียข้อมูล เกิดการหยุดให้บริการ เป็นต้น

**Internet Infrastructure Attacks** (การโจมตีโครงสร้างพื้นฐานที่ต้องใช้ในระบบเครือข่าย) เป็นการมุ่งโจมตีไปยังระบบหลักที่เครือข่ายต้องใช้งานเช่น Name Server (เครื่องแม่ข่ายที่ทำหน้าที่เปลี่ยนชื่อเป็นไอพีแอดเดรส) เพื่อทำให้เครื่องในเครือข่ายไม่สามารถใช้งานอินเทอร์เน็ต หรือมุ่งไปที่เครื่องที่ทำหน้าที่เป็นเกตเวย์ (Gateway) เพื่อหยุดการให้บริการ การโจมตีลักษณะนี้เกิดได้ค่อนข้างยาก แต่เมื่อเกิดแล้วจะทำให้ระบบหยุดทำงานเป็นเวลานาน

## 2.4 ระบบป้องกันการบุกรุกเครือข่าย

ระบบป้องกันการบุกรุกเครือข่ายเป็นสิ่งที่ออกแบบเพื่อใช้จับการโจมตีที่ซ่อนแฝงมา และ มีมาตรการต่อต้านที่อิสระเพื่อยับยั้งการโจมตีเหล่านั้น โดยทั่วไปแล้ว IPS จะอยู่ตรงในตำแหน่งที่เส้นทางที่แพ็กเกตทำการเดินทางในเครือข่าย (inline เน็ตเวิร์ค) และ เฝ้าดู เมื่อมีเหตุการณ์เกิดขึ้น มันจะทำการอ้างถึงกฎเงื่อนไขที่เราตั้งไว้ สิ่งนี้ไม่เหมือนกับ IDS ซึ่งมันจะไม่ได้ทำงานอยู่ในตำแหน่ง Inline แต่จะเป็นในลักษณะแบบ Passive

การกระทำของผู้ใช้จะมีลักษณะเช่นเดียวกันกับการกระทำในที่ได้กำหนดเอาไว้ล่วงหน้าในฐานความรู้แล้ว ถ้าการกระทำไม่เป็นไปตามที่อยู่บนรายการที่ยอมรับนั้น IPS จะป้องกัน การกระทำนั้น ซึ่งจะไม่เหมือนกับ IDS ด้วยเหตุผลคือ ใน IPS จะทำการบอกกล่าวก่อนที่การกระทำนั้นจะทำสำเร็จในหน่วยความจำ อีกด้านหนึ่งของมัน จะมีการเปรียบเทียบ ไฟล์ checksum กับ รายการ checksum ที่ถูกต้องก่อนอนุญาตให้ ไฟล์ นั้นจะ ทำสำเร็จ และ ทำงานโดยการสกัดการเรียกใช้งานระบบ ดังนั้นการตรวจจับการโจมตีภายใน IPS จะสำเร็จได้ ประกอบด้วย Signature Detection , Protocol Analysis , Anomaly Detection

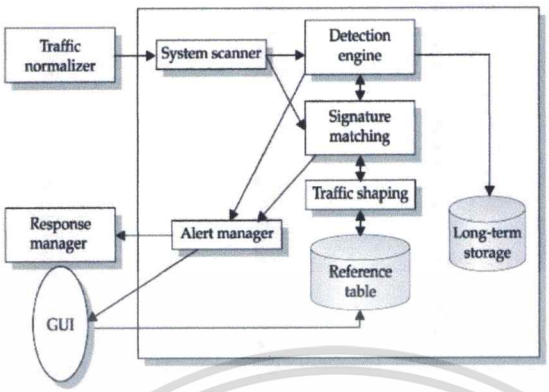
**2.4.1 ส่วนประกอบของ IPS** ประกอบด้วย 4 ส่วนหลักประกอบด้วย

1. Traffic normalizer
2. Service scanner
3. Detection engine
4. Traffic shaper

Traffic normalizer จะแปลข้อมูลที่อยู่ในการจราจรเครือข่ายและทำการวิเคราะห์และทำแพ็กเกต reassembly เช่นเดียวกับการบล็อกทราฟฟิกที่เป็นอยู่ตอนนั้นถูกป้อนเข้าสู่ detection engine และ service scanner service scanner จะร่างตารางที่เกี่ยวข้องว่าแบ่งแยกประเภทของ ข้อมูลที่ผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประมวลผลมาแล้ว และ ช่วยให้ควบคุมการบีบการจราจรของกระแสข้อมูล Detection engine จะทำรูปแบบจับคู่เทียบกับ reference table และถือเอาการตอบรับในการตัดสินใจดังรูปที่ 1




**รูปที่2.1** Standard IPS Systems

**2.4.2 ส่วนประกอบของกระบวนการวิเคราะห์การบุกรุก**

กระบวนการวิเคราะห์การบุกรุกสามารถแตกออกเป็น 4 ขั้นตอน

1. Preprocessing
2. Analysis
3. Response
4. Refinement

Preprocessing เป็นหน้าที่สำคัญครั้งหนึ่งของข้อมูลที่เป็นการรวบรวมจาก IPS sensor ในขั้นตอนนี้ข้อมูลจะถูกรวบรวมขึ้นในบางแบบเพื่อแบ่งแยกออกเป็นประเภทpreprocessing จะช่วยตัดสินรูปแบบของข้อมูลที่จะเข้าไป ซึ่งโดยปกติบางรูปแบบได้มาตรฐาน หรือ สามารถเป็นตามโครงสร้างของฐานข้อมูล เมื่อไรที่ข้อมูลเป็นตามรูปแบบก็จะแตกแยกออกเพื่อช่วยในการแบ่งแยกออกเป็นแต่ละประเภท

ความสามารถของการแบ่งแยกออกเป็นประเภทต่าง ๆ เหล่านี้ขึ้นอยู่กับแผนการวิเคราะห์ที่นำมาใช้ สำหรับตัวอย่าง ถ้า นำ rule-based detection มาใช้ การแยกประเภทก็จะเกี่ยวกับ กฎ และรูปแบบตัวบอกลักษณะ ถ้าการตรวจจับที่ใช้ผิด โดยปกติเราจะมีสถิติของ profile เป็นพื้นฐานบนความแตกต่างของ algorithms ซึ่งใช้พฤติกรรมของผู้ใช้เป็นเส้นฐานทุกครั้ง และ พฤติกรรมใดๆ ที่เกิดขึ้นอย่างไม่คาดคิดการแบ่งออกเป็นประเภทต่างๆนั้นทำได้ช้าลงตามสิ่งที่ผิดจากปกติ

เหนือจากการทำให้สมบูรณ์ของกระบวนการแบ่งแยกประเภทต่างๆข้อมูลเป็นข้อมูลที่เชื่อมเข้าด้วยกันและใส่เข้าไปข้างใน คำบอกที่ชัดเจนหรือแบบการตรวจจับของบางสิ่งที่เราสนใจโดยการแทนที่ค่าของตัวแปร(สิ่งที่เปลี่ยนแปลงได้) แบบการตรวจจับเหล่านี้ตั้งอยู่ในฐานความรู้ซึ่งสะสมอยู่ใน analysis engine

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอขั้น preprocessing สำเร็จขั้นของการวิเคราะห์ (Analysis) ก็เริ่มขึ้น ข้อมูลที่บันทึกจะเปรียบเทียบกับฐานความรู้และข้อมูลที่บันทึกจะเป็นอย่างใดอย่างหนึ่งคือ event log หรือ ถูกละทิ้งไป จากนั้นข้อมูลถัดไปก็จะถูก analyzed

ขั้นต่อไปคือ respond อย่างหนึ่งที่แสดงให้เห็นถึงเหตุของความแตกต่างระหว่าง IPS กับ IDS ซึ่ง IDS มีข้อจำกัดของความสามารถการป้องกันคุณได้รับข้อมูลโดยไม่แสดงกิริยาใดๆ หลังจากรู้ข้อเท็จจริงแล้วดังนั้นคุณจะได้รับการเตือนให้ระวังหลังจากรู้ข้อเท็จจริงแล้ว เมื่อไหร่ที่ข้อมูลได้กลายเป็นข้อมูลที่บันทึกเหตุการณ์การณ์ ( logged )ของการบุกรุก การตอบสนองก็เริ่มขึ้นซึ่ง IPSนั้น sensor จะอยู่ที่ inline และมันสามารถเตรียมการป้องกัน แบบ real-timeไตร่ตรองและตอบรับอัตโนมัติ

ในขั้นสุดท้ายนี้คือ refinement ซึ่งเป็นที่ปรับแต่งให้ดีขึ้นของระบบ IPS ให้สามารถทำได้บนพื้นฐานของการใช้งานในครั้งก่อนๆ และตรวจจับการบุกรุก ให้ความปลอดภัยที่เป็นไปได้ รวบรวมความผิดพลาดและเป็นเครื่องมือรักษาความปลอดภัยที่แม่นยำมากขึ้น เป็นขั้นที่อันตรายมากๆ สำหรับสิ่งที่ได้รับมาจาก ระบบ IPS ระบบจะปรับให้เข้ากับสภาพแวดล้อมของท่านเพื่อรับค่าที่แท้จริงจากมัน

## 2.5 ประเภทของระบบป้องกันบุกรุก

ระบบป้องกันการบุกรุกเครือข่ายเป็นระบบป้องกันผู้บุกรุกชนิดหนึ่ง ซึ่งระบบป้องกันผู้บุกรุกจะสามารถแบ่งได้เป็นดังนี้

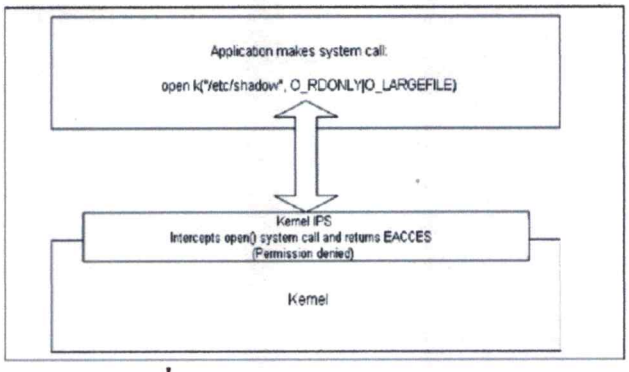
**A: แบ่งตามตำแหน่งในการตรวจจับการบุกรุก**

**- Host Intrusion prevention systems (HIPS)**

เป็นระบบที่คอยตรวจจับและป้องกันการบุกรุกในเครื่องคอมพิวเตอร์(ทั้งเซิร์พเวอร์ และเวิร์คสเตชั่น) ซึ่งซอฟแวร์นั้นจะทำงานระหว่างแอพพลิเคชั่นของระบบและ Os Kernel HIPS จะจับร่องรอยของการเคลื่อนไหวบนระบบต่อจากนั้นจะขึ้นอยู่กับกฎที่ได้กำหนดเอาไว้ว่ามันจะ block หรือ allow ให้เหตุการณ์นั้นเกิดขึ้น และ จะคอยเฝ้าดูวิเคราะห์การใช้งานภายในเครื่องหรือโพรเซสที่น่าสงสัย

เมื่อไหร่แอพพลิเคชั่นที่เป็นอันตรายตามรูปที่ 2 มันจะมีท่าทีว่าจะพยายามทำให้สำเร็จ (execute) การทำนั้นร้องขอ system call interface ที่เตรียมการโดย kernel กลไกที่ป้องกันสร้างขึ้นโดยตรงข้างใน kernel คือข้างในจะมีตำแหน่งที่ไม่ซ้ำกันที่จะ อนุญาตหรือปฏิเสธแอพพลิเคชั่นใดๆให้สามารถ เรียกใช้ระบบได้สำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 2.2.** System Call Interception

แสดงให้เห็นว่า kernel จะส่ง error code EACCESS กลับไปถึงแอพพลิเคชั่นนั้นที่พยายามจะเปิดไฟล์ /etc/shadow


**รูปที่ 2.3.** แอพพลิเคชั่น Shim

รูปที่ 2.3 จะแสดงถึงการบล็อกข้อมูลที่ปองร้ายโดย แอพพลิเคชั่น shim และทำการสร้าง application-specific error code แอพพลิเคชั่น shim เป็นส่วนที่มีความเชี่ยวชาญ ที่กะทัดรัดรวมอยู่กับแอพพลิเคชั่น (เป็นการทำงานภายในที่เกี่ยวกับ process address space) และการทำการตรวจตราข้อมูลที่เข้ามาและ ตรวจสอบคำสั่งเพื่อให้แน่ใจว่าจะทำงานโดยไม่มีความผิดพลาด การตรวจตราของข้อมูลที่เข้ามาโดยแอพพลิเคชั่นshim จะทำที่ภายในแอพพลิเคชั่นถ้าข้อมูลมีการปองร้ายโดยปกติ shim สามารถส่งคืนมันไป ข้อมูลที่ปองร้ายคือแต่ละอันจะถูกทิ้ง(drop)โดย shim ดังนั้นที่มันไม่เคยทำถึงระดับที่สูงกว่าภายในแอพพลิเคชั่นหรือ เป็นการเปลี่ยนแปลงเข้ามาในแบบที่ไม่เป็นภัย

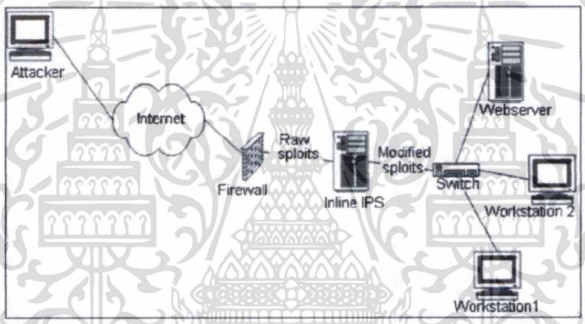
**- Network Intrusion prevention systems (NIPS)**

NIPS คือ อุปกรณ์ใด ๆ (ไม่ว่าจะเป็นซอฟท์แวร์หรือฮาร์ดแวร์ แพลทฟอร์ม) เพื่อออกแบบการวิเคราะห์ ดักฟังข้อมูลบนเครือข่ายทั้งหมดและเฝ้าดูว่ามันมีเหตุการณ์หรือการเคลื่อนไหวอะไรที่น่าสงสัย ตรวจจับและรายงานบนความปลอดภัยที่เกี่ยวข้องกับเหตุการณ์ NIPS ออกแบบมาเพื่อตรวจตรา ทราฟฟิก และเป็นพื้นฐาน บนการปรับแต่งสิ่งเหล่านั้น หรือ นโยบายความปลอดภัย พวกเราสามารถทิ้ง(drop) ทราฟฟิกที่ปองร้าย ซึ่งมันจะอยู่ในตำแหน่งที่มีการไหลของทราฟฟิกเครือข่ายและ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ป้องกันการโจมตีได้ทันท่วงที ในส่วนที่เพิ่มเติมของมันคือจะสามารถมองหา(ถอดรหัส) ในเลเยอร์ที่ 7 (โพรโทคอล) ได้เช่น HTTP,FTP,SMTP และ ตรวจสอบการบุรุกด้วยเครื่องบ่งชี้(scan intrusion signature) ค้นหาความผิดปกติของโพรโทคอล(search protocol anomalies) และตรวจจับ command ที่ไม่ปกติที่จะ executed บนเครือข่าย

ยกตัวอย่าง อุปกรณ์ที่ทำงานแบบ inline เช่นเราท์เตอร์ (router) ซึ่งมันจะทำการส่งผ่านไอพีแพ็กเกตระหว่างเครือข่ายที่เชื่อมต่อและก็บริดท์(Bridge) ซึ่งเชื่อมต่ออีเทอร์เน็ท 2 ส่วนเข้าด้วยกันเป็นต้นและอีกอย่างหนึ่งของมาตรการต่อต้านที่สำคัญที่เหมาะที่จะใช้กับ IPS คือ การแก้ไขข้อมูลของแอพพลิเคชั่นเลเยอร์(Application Layer Data) เทคนิคนี้จะอนุญาตให้ IPS ดัดแปลงข้อมูลในแพ็กเกต (pack payload data) ดังนั้นการโจมตีที่ แอพพลิเคชั่นเลเยอร์กลายเป็นไม่ได้ผลก่อนที่มันจะไปถึงเป้าหมายที่มันต้องการ




**รูปที่ 2.4.** Inline Network IPS

ตัวอย่างของการนำเทคโนโลยี Inline IPS มาใช้ก็คือ Snort_Inline ซึ่งจะทำการกล่าวถึงในหัวข้อ โปรแกรมป้องกันการบุกรุกต่อไป

**B: แบ่งตามวิธีการตรวจจับการบุกรุก**

**- Anomaly Detection หรือ Profile-based Detection**

การตรวจจับโดยวิธีนี้ต้องอาศัยการสร้างแฟ้มข้อมูล (Profile) ของผู้ใช้หรือกลุ่มของผู้ใช้งานในระบบขึ้นมา เพื่อใช้เก็บพฤติกรรมการใช้งานที่เป็นปกติจากกิจกรรมหรืองานที่ต้องทำ อยู่เป็นประจำ แฟ้มข้อมูลเหล่านี้จึงเปรียบเสมือนบรรทัดฐานที่ใช้ในการตรวจจับ พฤติกรรมที่ผิดปกติไปจากการทำงานตามปกติของผู้ใช้ในระบบด้วยวิธีนี้ระบบตรวจสอบผู้บุกรุกจะสามารถตรวจจับการบุกรุกได้โดยดูจากพฤติกรรมการที่ผิดไปจากการใช้งานปกติของระบบ ซึ่งจะไม่มีรูปแบบที่แน่นอนสำหรับทุกระบบขึ้นอยู่กับพฤติกรรมการใช้งานปกติของระบบนั้นๆเอง ฉะนั้นการกำหนดบรรทัดฐานของพฤติกรรมปกติในระบบจึงเป็นเรื่องที่สำคัญและละเอียดอ่อนมาก ถ้าระบบสามารถกำหนด บรรทัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฐานที่ครอบคลุมพฤติกรรมการใช้งานปกติของระบบได้หมด การเกิด False Positive จากระบบตรวจสอบผู้บุกรุกประเภทนี้ก็จะมีโอกาสเกิดขึ้นน้อย ในขณะที่การเกิด False Negative ก็มีโอกาสเกิดขึ้นได้จากการบุกรุกที่มีพฤติกรรมเหมือนการใช้งานตามปกติในระบบ ซึ่งในกรณีนี้แทบจะเป็นไปไม่ได้เลยที่ระบบตรวจสอบผู้บุกรุกประเภทนี้จะสามารถแยกแยะได้ว่าพฤติกรรมใดเป็นการใช้งานตามปกติหรือว่าเป็นบุกรุกสามารถแบ่งได้เป็น

1. **Behavioral analysis** มองหาสิ่งที่ผิดในชนิดของพฤติกรรมที่มีอยู่ในสถิติ
2. **Traffic-pattern analysis** มองหาลักษณะของแพทเทิรน(pattern) ใน network traffic
3. **Protocol analysis** มองหาเน็ตเวิร์คโพรโทคอลที่ฝ่าฝืนหรือการนำไปใช้ในทางที่ผิด

**- Misuse Detction หรือ Signature-based Detection**

เป็นการตรวจจับพฤติกรรมการบุกรุกโดยเปรียบเทียบจากข้อมูลลักษณะเฉพาะ (Signature) ที่ใช้อ้างอิง ซึ่งลักษณะเฉพาะเหล่านี้จะเป็นกลุ่มของกฎต่างๆที่เป็น รูปแบบหรือพฤติกรรมของผู้บุกรุกที่ใช้ในการบุกรุกเข้าสู่ระบบ

ฉะนั้นการกำหนดรูปแบบลักษณะเฉพาะที่ใช้อ้างอิงเปรียบเทียบที่ดี จะสามารถลดโอกาสในการเกิด False Positive ได้ ในขณะที่การป้องกันการเกิด False Negative จะขึ้นอยู่กับการหมั่นปรับปรุงฐานข้อมูลลักษณะเฉพาะเหล่านี้ให้ทันสมัยต่อรูปแบบการบุกรุกที่เกิดขึ้นใหม่ๆอยู่ตลอดเวลา จะอ้างอิงถึง signature detection , pattern matching และ misuse detection จะทำการวิเคราะห์เหตุการณ์ที่เกิดขึ้นและค้นหารูปแบบที่เหมือนกับรูปแบบการโจมตีซึ่งการตรวจสอบจะใช้เทคนิค pattern-matching ทำให้รู้ลักษณะของการโจมตีได้ วิธีการโดยทั่วไปคือจะตรวจและพิจารณา ดู contentใน packet โดยจะดู payload และ header ของ packet

## 2.6. ระบบป้องกันการบุกรุกโดยใช้ Snort inline

ในช่วงที่ผ่านมาได้มีการพยายามในการพัฒนาซอฟต์แวร์ซึ่งทำการตรวจจับการบุกรุก (IDS) แต่เนื่องจากในปัจจุบันการโจมตีหรือการบุกรุกนั้นนับวันยิ่งมีเทคนิคใหม่ๆเพิ่มมากขึ้น จึงได้เกิดเทคโนโลยีขั้นต่อมานั่นคือ ระบบป้องกันการบุกรุก (IPS) โดยมีทั้งที่เป็นฮาร์ดแวร์ และซอฟต์แวร์แต่ก็ยังมีราคาค่อนข้างสูงดังนั้นทางเลือกอีกทางหนึ่งคือการใช้ซอฟต์แวร์ที่เป็น open source นั้นก็คือ โปรแกรม snort การใช้ snort เป็นระบบตรวจจับการบุกรุกนั้นมีมาตั้งแต่ปี 1998 และเป็นที่นิยมมากในปัจจุบันซึ่งเขียนโดยภาษา C และต่อมา snort ได้เปลี่ยน Code name project เป็น Snort inline เพื่อที่จะนำเสนอการทำงานในรูปแบบของ IPS โดยเพิ่มความสามารถในการป้องกันภายใต้แนวความคิด IDS + Firewall = IPS (Active Respond)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.6.1 โปรแกรมป้องกันการบุกรุก**

Snort ซึ่งเป็นโครงการพัฒนา IDS นั้นได้มีการพัฒนาเป็นโครงการที่เกี่ยวข้องกับ IPS โดยตรงซึ่งใช้ชื่อโครงการว่า Hogwashโดยมีการพยายามนำ Snort Engine มาแก้ไขโดยให้ความสามารถป้องกัน(drop)การโจมตีของการบุกรุกได้โดยไม่เพียงแต่แจ้งเตือน(alert)อย่างเดียวซึ่งปัญหาของ Hogwash ก็คือไม่ได้ใช้ Intrusion Signature ตัวเดียวกับ Snort ดังนั้นการพัฒนาการจึงมีลักษณะที่ไม่สอดคล้องกัน จึงมีแนวความคิดที่จะทำให้ Snort มีความสามารถในลักษณะที่เป็น IPS โดยนำ Snort Engine , Iptables มาทำงานร่วมกันโดยดักจับแพ็กเกตที่ Layer2(Data-Link Layer)ซึ่งมีลักษณะการทำงานแบบบริดจ์และตั้งชื่อโครงการใหม่นี้ว่า "Snort Inline" ซึ่งเป็น IPS ที่สามารถป้องกันการโจมตีของ Hacker อย่างได้ผล

**2.6.2 โหมดการทำงานของ Snort**

1.Sniffer Mode เป็นการดักแพ็กเกตบนเครือข่ายแล้วนำมาแสดงบนหน้าจอว่ามีข้อมูลอะไรบ้างที่วิ่งอยู่บนเครือข่าย

2. Packet Logger Mode เป็นการบันทึกข้อมูลจากการดักแพ็กเกตที่วิ่งบนเครือข่ายเก็บลงบนดิสก์ไว้

3. Network Intrusion System Mode เป็นการวิเคราะห์ข้อมูลบนเครือข่ายเพื่อตรวจสอบ แพ็กเกตซึ่งใช้กฎในการตัดสินใจว่าเป็นการโจมตีหรือไม่(snort.conf)

4. Inline Mode ซึ่งจะเอาแพ็กเกตมาจาก Iptables แทน libcab ทำให้ Iptable ทำการหยุดหรือปล่อยแพ็กเกตบนกฎของ Snort ที่ใช้การกำหนดกฎแบบ Inline

และสำหรับการทำงานของ Snort_Inline นั้นอย่างที่กล่าวไว้คือมันต้องทำงานร่วมกับ Iptable และในการติดตั้ง Iptable การ Compile จะต้องใช้คำสั่ง "make install-devel" เพราะมันจะทำการติดตั้ง libipq library ที่จะอนุญาตให้ Snort_Inline รับแพ็กเกตจาก iptable ได้ซึ่งในรายงานจะเน้นการทำงาน mode inlineเป็นหลัก

**2.6.3 ลักษณะของกฎใน Snort_Inline**

กฎ 3 ประเภทที่เราสามารถจะใช้เมื่อเวลาใช้งาน Snortใน mode Inline คือ **Drop, reject , sdrop** และใน snort rule ถูกแบ่งออกเป็น 2 ส่วนคือ Rule header , Rule action ดังนี้

**- Rule Header**

Rule Header จะเป็นรายการแรกใน Rule Header ซึ่งจะบอก Snort ว่าต้องทำอะไรเมื่อตรวจพบแพ็กเกตที่ตรงกับเงื่อนไขที่กฎระบุไว้โดยมีการตั้งค่าการกระทำอยู่ 5 รูปแบบ

**1. Rule Actions**

**1.1 alert สร้างและทำการแจ้งเตือนว่าพบการบุกรุกและบันทึกแพ็กเกตนั้นไว้**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**1.2 log** บันทึกแพ็กเกตนั้นอย่างเดียว

**1.3 pass** ละทิ้งแพ็กเกตนั้นไป

**1.4 activate** ทำการแจ้งเตือนแล้วทำ Dynamic rule ที่กำหนด

**1.5 dynamic** จะไม่ทำอะไรจนกว่าจะมีการ activate โดย activatw rule ซึ่งทำงานเหมือนกับ Log rule

**1.6 drop** สั่งให้ Iptable หยุดแพ็กเกตและบันทึกแพ็กเกตนั้นไว้

**1.7 reject** สั่งให้ Iptable หยุดแพ็กเกตและบันทึกแพ็กเกตไว้ต่อจากนั้นก็จะส่ง TCP ถ้าใช้โพรโตคอล TCP หรือ ส่ง Icmp port unreachable ถ้าใช้โพรโตคอล Udp

**1.8 sdrop** เพื่อจะบอก Iptable ว่าให้ drop แพ็กเกตและไม่ต้อง log ไว้

**2. Protocol** เป็นฟิลด์ถัดมาของ Rule โดยในปัจจุบัน Snort สามารถวิเคราะห์ได้ 4 โพรโตคอลคือ TCP , UDP , ICMP , IP และในอนาคตจะมีการเพิ่ม ARP , IGRP , GRE , OSPF , RIP , IPX เป็นต้น

**3. Ip Address** เป็นข้อมูล IP Address ที่ให้กับกฎโดยคำว่า any หมายถึง IP address ใดๆ

**4. Port Number** เบอร์ port ของเครื่องต้นทางและปลายทางที่ใช้ในการติดต่อ (Source ,dest , lange) โดยสามารถระบุแบบเฉพาะเจาะจงหรือช่วงของหมายเลขพอร์ตเช่น 23 สำหรับ telnet เป็นต้น

**5. Direction** เป็นส่วนที่บอกทิศทางการติดต่อ

- (->) คือจะพิจารณาเพียงทิศทางเดียวและฝั่งซ้ายของเครื่องหมายคือต้นทางฝั่งขวาคือปลายทาง **[ข้อสังเกต : จะไม่มีการใช้เครื่องหมาย ( <-)]**

- (<>) คือพิจารณาทั้ง 2 ทาง

**- Rule Option** ช่วยเพิ่มความสามารถของกฎซึ่งมีหลักอยู่ 4 ประเภทในส่วนนี้ดังนี้

**1. Meta-data** ทางเลือกนี้เป็นการจัดหาข้อมูลเกี่ยวกับกฎแต่ไม่ส่งผลกระทบกับการตรวจจับ

**1.1 Msg ( msg: "<message text>"; )**

ระบุข้อความที่จะถูกแสดงในตอนแจ้งเตือน(alert) และตอนบันทึกแพ็กเกต

**1.2 Sid (sid: <snort rules id>;)**

ใช้ในการกำหนดหมายเลขของกฎในsnort โดยที่จะต้องไม่ซ้ำกันและในไฟล์ sid-msg.map จะบรรจุการจับคู่ของ alert message ไปถึง Id ของกฎ snort

**1.4 rev** (rev: <revision integer>)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้ระบุการปรับปรุงแก้ไขและต้องไม่ซ้ำกันกฎของ snort การปรับปรุงแก้ไขจะต้องใช้ร่วมด้วยกับ snort id ที่อนุญาตให้มีการแก้ไขและเปลี่ยนแปลงกับการปรับปรุง signature และ description ให้เป็นข้อมูลที่ทันสมัย

**1.5 Classtype** (classtype: <class name>;)

คือการกำหนด class ให้กับกฎจะกำหนดในไฟล์ classification.config

**1.6 Priority (priority: <priority integer>;)**

เป็นการกำหนดระดับความเข้มงวดของกฎซึ่งถ้าน้อยยิ่งสำคัญมาก และค่านี้สามารถเอาไปแทนที่ค่า default priority ที่มาจาก classtype ของกฎได้

**2. Payload** ทางเลือกนี้จะมองหาข้อมูลภายใน payload ทั้งหมด

**2.1 content: [!] "<content string>";**

เป็นการค้นหาในส่วนของแพ็กเกต payload ว่ามีข้อมูลตรงกับที่ระบุไว้หรือไม่ ซึ่งในส่วน content string สามารถระบุค่าที่เป็น text รวมกับ Binary ได้โดยค่าที่เป็น Binary จะปิดหัวปิดท้ายด้วยเครื่องหมาย "|" และการใช้เครื่องหมาย "!" อยู่หลัง ":" หมายถึงการค้นหาว่าไม่มีข้อความนี้ในส่วน Payload

**2.2 Depth:<number>;**

เป็น Keyword ที่ใช้ร่วมกับ content ในการบอกถึงจำนวน byte ที่จะค้นหาใน payload ซึ่งทำให้ไม่ต้องเสียเวลาในการทำ pattern matching

**2.3 offset:<number>;**

เป็น Keyword ที่ใช้ร่วมกับ content ในการบอกตำแหน่งถัดจากจุดเริ่มต้นของ payload ในการค้นหาคำมีประโยชน์สำหรับ CGI scan ซึ่งจะไม่ค้นใน 4 byte แรก

**2.4 byte test**

สามารถทดสอบค่าไบนารี่หรือเป็นตัวแทนในการแปลง byte string เป็นค่าไบนารี่ที่มีค่าเท่ากันและทดสอบมัน

**2.5 content list:"<file_name>";**

เป็นการค้นหาในส่วนของแพ็กเกต payload โดยใช้ข้อมูลในไฟล์ซึ่งบรรจุคำที่ใช้ในการค้นหา

นอกจากนี้ยังมี keyword อื่นๆอีกเช่น distance,within,uricontent,isdata, pcre,byte_jume ,Ftpbounce, regex ซึ่งช่วยให้การใช้งานมีความยืดหยุ่นมากขึ้น

**3. Non-payload** มองหาข้อมูลที่ไม่ใช่ส่วนของ payload

**3.1 ttl:"<number>";**

ตรวจสอบค่า time –to –live ของแพ็กเกตว่าตรงกับที่ระบุหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**3.2 Tos:"number";**

**ตรวจสอบค่าใน ToS field ของแพ็กเกตว่าตรงกับที่ระบุหรือไม่**

**3.3 id:"<number>";**

**ตรวจสอบค่าใน fragment id field ของ Ip header ว่าตรงกับที่ระบุหรือไม่**

**3.4 ipopts: <option>;**

**ดูค่าใน Ip option field ว่าตรงกับที่ระบุไว้หรือไม่ซึ่งใน 1 กฎจะระบุค่า Ip option ได้ 1 ค่าเท่านั้น**

**3.5 fragbit:<bit values>;**

**ตรวจ Fragment และ Reserve bit ซึ่งมี bit value 3 ค่าได้แก่ R แทน Reserve bit (RB) , M แทน More fragments bit (MF) และ D แทน Don't fragments bit (DF)**

**3.6 dsize:[]>|<]<number>[<><number>];**

**ระบุขนาดpayloadของแพ็กเกตเช่น dsize:100<>500 คือขนาดระหว่าง 100 ถึง 500**

**3.7 flags:<flag values>[,mask value];**

**ตรวจค่า TCP Flag ซึ่งมีค่า Flag 9 ค่าที่ใช้ได้ใน Snort ได้แก่**

F= FIN(LSB in TCP Flag byte)

S=SYN

R=RST

P=PSH

A=ACK

U=URG

2= Reserved bit 2

1=Reserved bit 1 (MSB in TCP Flag byte)

0=No TCP Flag Set

**3.8 flow**

**ใช้ร่วมกันกับ tcp stram มันเป็นกฎที่ใช้กับทิศทางที่แน่นอนของ traffic flow เท่านั้น**

**3.9 seq:<number>;**

**ดูค่า Tcp sequence number ซึ่งจะระบุเห็นค่าคงที่ดังนั้นปกติจะไม่ค่อยได้ใช้เพราะค่า sequence number จะเปลี่ยนอยู่ตลอด**

**3.10 ack:<number>;**

**ตรวจค่าใน acknowledgement field (ACK) ว่าตรงกับที่ระบุหรือไม่**

**3.11 Window:[!]<number>;**

**ใช้ตรวจสอบเพื่อกำหนด ขนาด window**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

05995

**3.12 itype:<number>;**

ตรวจดูค่าใน ICMP type field ซึ่งในค่าในส่วนนี้มักใช้กับการโจมตีแบบ Denial of Service และ flooding attack

**3.13 Icode:<number>;**

การตรวจดูค่าใน ICMP Code field ซึ่งค่าใช้ประโยชน์ได้เหมือนกับ itype

**3.14 icmp_id:<number>;**

ดูค่าหมายเลข ICMP ID ของ ICMP Echo แพ็กเกต

**3.15 icmp_seq:<number>;**

ดูค่า ICMP Sequence number ของ ICMP Echo แพ็กเกต

นอกจากนี้ยังมี keyword อื่นๆ อีกเช่น rpc , ip_proto sameip, flagoffset ,flowbits ซึ่งช่วยให้การใช้งานมีความยืดหยุ่นมากขึ้น

**4. Post-detection** จะเป็นกฎที่กำหนดการกระตุ้นว่าจะให้เป็นยังไงหลังจากกฎ has fired

**4.1 logto:"<filename>"**

จะบอก snort ว่าให้บันทึกแพ็กเกตไปยังไฟล์ที่ระบุแทนที่จะบันทึกร่วมกับ log อื่นๆ

**4.2 session:[printable|all];**

ใช้ในการบันทึกข้อมูลจาก Tcp session ซึ่งจะเหมาะสมเมื่อต้องการจะรู้ว่ามีการพิมพ์ข้อความหรือเห็นอะไรบ้างในการใช้งาน เช่น telnet , ftp หรือ web ซึ่งเลือกได้ว่าจะเก็บเฉพาะข้อมูลที่แสดงผลออกมาได้(printable)หรือข้อมูลทั้งหมด

**4.3 resp** ใช้ในการพยายามที่จะปิด session เมื่อเวลาการแจ้งเตือน(alert) ถูกกระตุ้น

นอกจากนี้ยังมี keyword อื่นๆ อีกเช่น react,Tag ซึ่งช่วยให้การใช้งานมีความยืดหยุ่นมากขึ้น

**- การแจ้งเตือน(Output Module)**

เราสามารถทำการกำหนดรูปแบบของการแจ้งเตือนใน snort และในส่วนนี้จะทำงานเพื่อรับการ alert หรือ log

**1. XML** จะให้ Xml plug-in เก็บ log ในรูปแบบ SNML(Simple network markup language )ซึ่งสามารถส่ง report ไปยัง database หรือ snort ตัวอื่นได้

**2. Database** สามารถบันทึก log ไปยังฐานข้อมูลได้ ซึ่งที่สามารถใช้ตอนนี้ก็มี Mysql , Oracle เป็นต้น

**3. CVS** เป็น plugin ซึ่งจะส่ง alert message ให้อยู่ในรูปแบบที่ง่ายต่อการอิมพอร์ทไปยังฐานข้อมูล

**4. Alert_syslog** ระบบจะส่ง alert ไปยัง syslog ของระบบ

เอกสารนี้เป็นที่ [illegible] ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**5. Alert_fast** alert message แต่ละอันจะถูกบันทึกใน 1 บรรทัดซึ่งจะให้ความเร็วกว่า alert_full

**6. Alert_full** alert message จะเก็บรายละเอียดของแพ็กเกต header ทั้งหมด

**7. Alert_smb** ระบบจะส่ง alert message ไปยัง WinPopUpผ่านโพรโตคอล smb

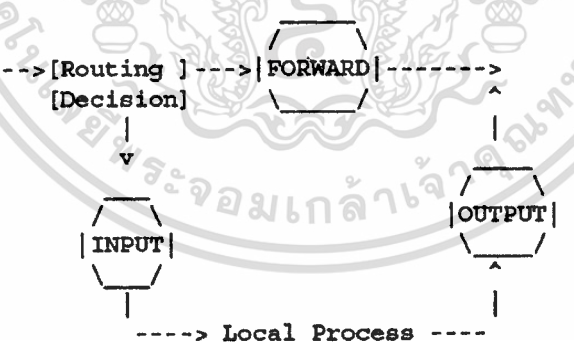
**8. Alert_unixsock** จะส่ง alert message ไปยัง unix socket ซึ่งจะต้องทำการติดตั้งระบบโดเมน unix socket และส่ง alert message ไปหามัน

**9. Unifield Unifield** output plugin ถูกออกแบบให้มีความเร็วมากที่สุดในการให้บันทึก log ของ snort ซึ่งจะบันทึกไว้ 2 ส่วนคือ alert ไฟล์ จะเก็บรายละเอียดของเหตุการณ์ใน snort ส่วน log ไฟล์ จะเก็บข้อมูลเกี่ยวกับแพ็กเกตที่เกี่ยวกับเหตุการณ์และทั้ง 2 ไฟล์จะเก็บอยู่ในรูปแบบของไบนารี่ไฟล์

**10. Log_tcpdump** จะเก็บบันทึกแพ็กเกตบนเครือข่ายในรูปแบบของ tcpdump

## 2.7 ไฟล์วอล IPTABLES

Iptables เป็นโปรแกรมที่พัฒนาบน Linux Kernel 2.4 มีความสามารถในการตรวจสอบสถานะการทำงานของทราฟิกส์บนแอพลิเคชั่นต่างๆ ได้อย่างดี สนับสนุนการทำงานแบบ SPI (Stateful Inspection) โดยทำการวิเคราะห์และตรวจสอบพฤติกรรมการทำงานของตัวแอพลิเคชั่นว่ามรอะไรผิดปกติหรือไม่




**รูปที่ 2.5 เส้นทางการเดินทางของแพ็กเกตใน filter table**

### 2.7.1 การทำงานของ IPTABLES

ไฟล์วอล iptables นั้นมีความสามารถหลักๆ เหมือนกับไฟล์วอลทั่วไปคือ กรองแพ็กเกตที่ผ่านเข้าออกระหว่างเครือข่ายซึ่งปกติจะต้องวางขั้นระหว่าง 2 เครือข่าย iptables สามารถทำงานได้กับ 3 ส่วนหรือตาราง 3 ตาราง คือ

**1. Filter table** ใช้สำหรับกรองแพ็กเกตมี 3 chain คือ INPUT, OUTPUT, FORWARD ดังรูปที่ 9. ซึ่งจะเป็นส่วนหลักในการป้องกันระบบเครือข่ายและหากพิจารณาการเดินทางของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แพ็กเกตเฉพาะในส่วนของ Filter tableตามรูปโดยเมื่อแพ็กเกตเข้ามาในระบบจะเข้าไปยัง routing decision เพื่อตัดสินใจว่าแพ็กเกตจะถูกส่งไปที่ใด

- ในกรณีที่แพ็กเกตถูกส่งผ่านไปยังเครื่องอื่นแพ็กเกตนั้นจะต้องถูกตรวจสอบโดย rule ใน FORWARD Chain

- ถ้าแพ็กเกตนั้นมีเป้าหมายเป็นเครื่องนี้(เครื่องที่ Run Iptables) ตัวแพ็กเกตจะถูกตรวจสอบโดย Rule ใน INPUT Chain

- และในกรณีที่แพ็กเกตถูกสร้างจากเครื่องนี้ตัวแพ็กเกตจะถูกตรวจสอบจาก Rule ใน OUTPUT Chain ก่อนที่จะถูกส่งออกไป

**2. NAT table** ใช้สำหรับการแปลงแอดเดรส (Network Address Translation) มี 3 Chain คือ PEROUTING , POSTROUTING , OUTPUT

**3. Mangle table** เป็นตารางที่ใช้เปลี่ยนแปลงหรือแก้ไข packet เช่น เปลี่ยนค่า TTL, MARK ซึ่งปกติจะใช้ในการทำ routing ที่มีความซับซ้อนสูง มี 2 built-in chain คือ PREROUTING chain (ใช้แก้ไข packet ก่อนที่จะเข้าสู่ไฟร์วอลล์และก่อนเข้าสู่ routing decision) และ OUTPUT chain (ใช้แก้ไข packet ที่ถูกสร้างโดยไฟร์วอลล์ก่อนที่มันจะถูกส่งไปยัง routing decision) ทั้งนี้ไม่สามารถทำ network address translation หรือ masquerading ที่ table นี้ได้ และในเอกสารฉบับนี้จะไม่กล่าวถึง mangle อีก เนื่องจากเป็นส่วนที่ไม่นิยมนำไปใช้

### 2.7.2 รูปแบบการใช้งาน iptables

iptables จะมีรูปแบบการใช้งานดังนี้คือ

```
iptables [table] <command> <match> <target/jump>
```

โดย rule ที่เขียนขึ้นจะเป็นเป็นตัวบอกเคอร์เนลว่าให้กระทำ action อย่างไร ในกรณีที่พบ packet ตรงตามที่ระบุไว้

- **[table]** หมายถึง ตารางหรือ table ที่ต้องการระบุ เช่น iptables -t nat หมายถึงให้ทำงานกับ nat table ในกรณีที่ไม่ได้ระบุตาราง iptables จะถือว่าคำสั่งดังกล่าวระบุถึง filter table โดยอัตโนมัติ

- **<command>** จะเป็นตัวสั่งให้ iptables ทำในสิ่งที่ต้องการ เช่น iptables -A INPUT ซึ่งหมายถึงให้สร้าง rule ต่อท้าย INPUT chain ใน filter table

- **<match>** เป็นส่วนที่ใช้ตรวจสอบว่า packet มีข้อมูลตรง (match) กับที่ระบุไว้หรือไม่ เช่น มี source ip address เป็น 1.2.3.4

- **<target/jump>** เป็นตัวระบุว่าเมื่อเจอ packet ที่ match ก็จะกระทำ (action) ตามที่ระบุไว้ เช่น ถ้า packet ใดมี source ip address เป็น 1.2.3.4 ให้ DROP packet นั้นทิ้งไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

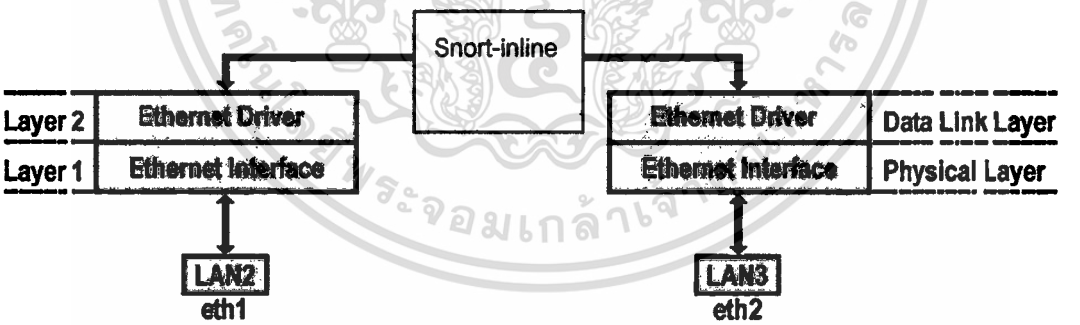
แต่ในกรณีของ IPS นั้นเราจะใช้ iptables เพื่อให้ทำงานในลักษณะ Data Control เพราะเนื่องจากมันจะทำงานในรูปแบบที่ต่างจากรูปแบบเดิมอยู่เล็กน้อยคือการเขียน chain ต่างๆ ยังเหมือนเดิมแต่ในกรณีนี้จะ jump มาใส่ ใน queue ของ iptable เพื่อเป็นที่พักให้ snort สามารถมาหยิบ แพ็คเกตเหล่านั้นมาวิเคราะห์ต่อได้นั่นเอง

```
iptables -I INPUT -p tcp --dport 80 -j QUEUE
```

คือแพ็คเกตที่เข้ามาที่เครื่องนี้ที่มี โพรโตคอล TCP มีหมายเลขพอรต์ เป็น 80 ให้กระโดดไปที่ queue

## 2.8 Bridge

ในการนำเสนอรูปแบบการทำงานของ IPS นั้น Snort เองมีความสามารถในลักษณะที่เป็น IPS โดยนำ Snort Engine , iptables มาทำงานร่วมกันโดยดักจับ Packet ที่ Layer2 (Data-Link Layer) ซึ่งมีลักษณะการทำงานแบบ Bridge ซึ่งเราจำเป็นต้องใช้ซอฟต์แวร์อีกตัวและที่ผมนำมาใช้บนระบบปฏิบัติการลินุกซ์ที่เรียกว่า Brctl มันจะทำให้เครื่องลินุกซ์ของเราสามารถทำงานในลักษณะนี้ได้




**รูปที่ 2.6.** Bridge and Snort inline

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 3
# การศึกษาและออกแบบระบบป้องกันการบุกรุกระบบเครือข่าย ด้วย Web

## 3.1. ความต้องการของระบบ

ความต้องการของระบบป้องกันการบุกรุกเครือข่ายด้วยเว็บนั้นที่จะพัฒนาขึ้นมีดังต่อไปนี้

- เป็นระบบที่สามารถทำงานบนระบบปฏิบัติการลินุกซ์ ซึ่งเว็บที่เขียนขึ้นมาใช้ภาษา php ในการพัฒนา และติดต่อฐานข้อมูล MySQL
- ระบบจะมีฐานข้อมูลเป็นของตัวเองซึ่งจะสร้างภายในฐานข้อมูลของ snort ซึ่งตรงนี้ได้พัฒนา script โดยใช้ภาษา perl เพื่อทำการนำกฎของสนอร์ทซึ่งเป็นเพียง text file ธรรมดาเข้าฐานข้อมูลซึ่งที่ใช้คือ MySQL ระบบจะทำการ load rule เข้ามาเก็บในฐานข้อมูลซึ่งเราจะสามารถ enable หรือ disable กฎต่างๆ ได้ว่าเซ็นเซอร์ ตัวไหนให้ใช้กฎไหนเป็นต้น
- เราสามารถทำการปรับแต่งค่าคอนฟิกเบื้องต้นของสนอร์ทได้โดยผ่านทางหน้า web front end
- เมื่อเราได้มีการ update rule ใหม่เข้าไปเซ็นเซอร์จะตรวจสอบ time stamp ของ rule นั้นๆ และจะดึง rule ใหม่ที่ได้จากการตรวจสอบมาเก็บไว้ในแต่ละเซ็นเซอร์
- Rule ที่ได้จากการ filter จาก text file มานั้นจะมีการแบ่งกลุ่มของการโจมตีด้วย

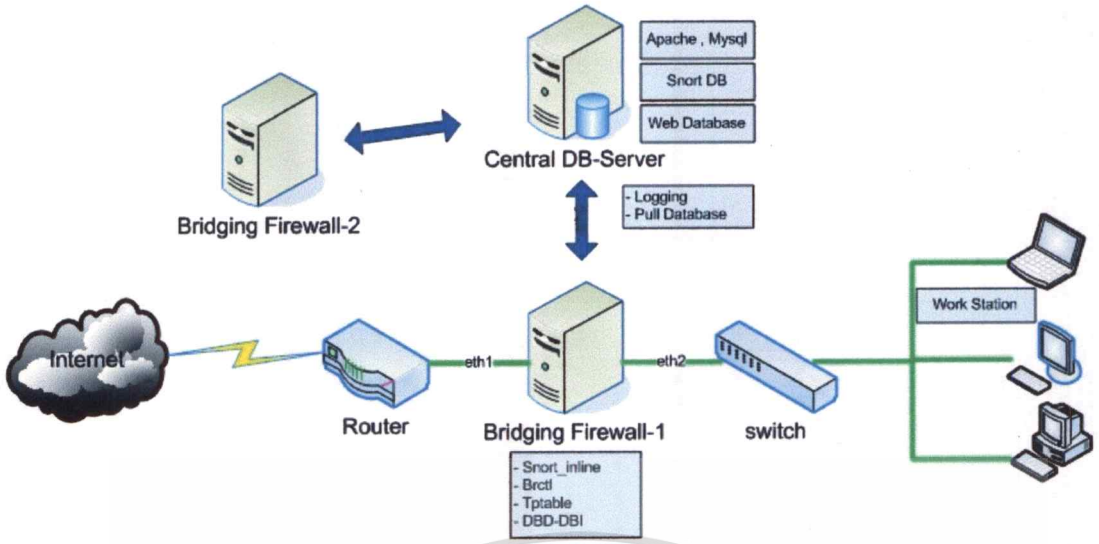
## 3.2. การออกแบบการทำงานของระบบ

หลังจากที่ได้ทำการศึกษาวิเคราะห์การทำงานของสนอร์ทและซอฟต์แวร์ที่ใช้ในการ implement ระบบทั้งหมดแล้วพบว่าสนอร์ทนั้นโดยปกติไม่ว่าจะ rule , config files ต่างๆนั้นล้วนแล้วแต่เป็น text file ธรรมดาทั้งสิ้นดังนั้นแนวความคิดที่จะจัดการกฎทั้งหมดของสนอร์ทจากศูนย์กลางนั้นอาจมีความเป็นไปได้ โดยมีแนวคิดที่ว่าเราทำฐานข้อมูลกฎของสนอร์ททั้งหมดไว้ และถ้า เซ็นเซอร์สนอร์ทตัวไหนต้องการกฎอันไหนก็มาดึงจากคาต้าเบสเซิร์ฟเวอร์ไปเก็บไว้ที่ตัวมัน (text) โดยผ่านทางหน้า เว็บแอพพลิเคชั่นทำให้เซ็นเซอร์สนอร์ทแต่ละตัวนั้นมีกฎตามที่ได้เลือกไว้เหมือนปกติของมัน

จึงทำการออกแบบโครงสร้างการทำงานของ Web-based IPS Tool และภาพรวมของระบบโดยแสดงผ่าน Use case diagram ในรูปที่ 3.1

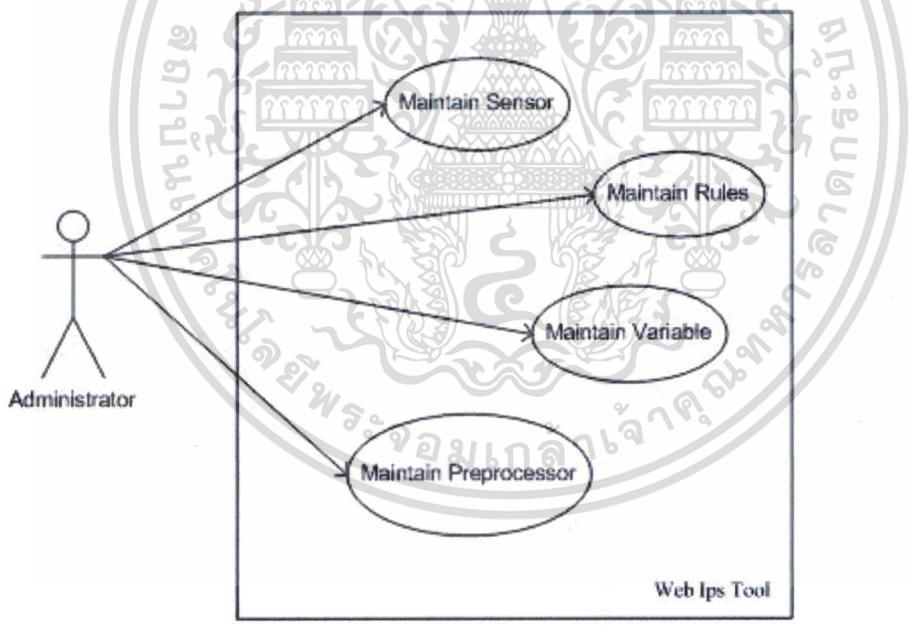
โดยภาพรวมของ Web-based IPS Tool จะประกอบด้วยกระบวนการทำงานที่เกี่ยวข้องกัน 4 โพรเซสคือ Maintain Sensor , Maintain Rules ,Maintain Variable และ Maintain Preprocessor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 3.1 ภาพโดยรวมของการทำงานของ Web-based IPS Administration Tool

### 3.2.1 Usecase Diagram




**รูปที่ 3.2.** Use Case Diagram ของ Web-based IPS Administration Tool

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 3.1** คำอธิบายยูสเคสไดอะแกรม Maintain Sensor

| ยูสเคส | Maintain Sensor |
|---|---|
| วัตถุประสงค์ | เพื่อทำการเพิ่มเซ็นเซอร์ของสนอร์ทให้กับระบบรวมถึงแอคติเวทเซ็นเซอร์ให้พร้อมใช้งานและควบคุมการใช้กลุ่มของกฎของสนอร์ท |
| เงื่อนไขตอนเริ่มต้น | ต้องมีข้อมูลของกฎของสนอร์ท |
| เมื่อทำงานเสร็จ | แอคมินสามารถใช้งานได้ตามปกติ |
| เมื่อทำงานไม่เสร็จ | ไม่สามารถเพิ่มเซ็นเซอร์หรือจัดการกลุ่มของกฎของสนอร์ทได้ซึ่งอาจจะมีข้อผิดผลาดแจ้ง |
| แอกเตอร์ที่เกี่ยวข้อง | Administrator |
| สิ่งกระตุ้นการทำงาน | ข้อมูลการริจิสเตอของเซ็นเซอร์ของสนอร์ท |
| อินพุต | ข้อมูลเซ็นเซอร์ของสนอร์ท |
| เอาท์พุต | จัดการและควบคุมกฎของแต่ละเซ็นเซอร์ได้ |
| รายละเอียด | - เลือกที่ เมนู Maintain Sensor<br>- เพิ่มหรือเซ็ตแอคติเวทเซ็นเซอร์<br>- จัดการกลุ่มของกฎของสำหรับแต่ละเซ็นเซอร์ |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 3.2** คำอธิบายยูสเคสไดอะแกรม Maintain Rule

| ยูสเคส | Maintain Rule |
|---|---|
| วัตถุประสงค์ | เพื่อจัดการกลุ่มของกฎของสนอร์ทรวมทั้งกฎแต่ละกฎซึ่งโดยการแก้ไขเปลี่ยนแปลงข้อมูลของกฎ , เพิ่มกฎหรือกลุ่มของกฎ , ลบกฎหรือกลุ่มของกฎ , ย้ายกลุ่มของกฎใหม่และการ enable/disable กฎเป็นต้น |
| เงื่อนไขตอนเริ่มต้น | ต้องมีข้อมูลของกฎของสนอร์ท |
| เมื่อทำงานเสร็จ | แอดมินสามารถจัดการกฎหรือกลุ่มของกฎของสนอร์ทได้ตามปกติ |
| เมื่อทำงานไม่เสร็จ | ไม่สามารถจัดการกฎหรือกลุ่มของกฎของสนอร์ทได้(ต้องพิจารณาในส่วนของ permission databases) |
| แอกเตอร์ที่เกี่ยวข้อง | Administrator |
| สิ่งกระตุ้นการทำงาน | ข้อมูลของกฎของสนอร์ท |
| อินพุต | ข้อมูลของกฎของสนอร์ท |
| เอาท์พุต | จัดการและควบคุมกฎหรือกลุ่มของกฎของสนอร์ทได้ |
| รายละเอียด | - เลือกที่ เมนู Maintain Rule<br>- จัดการกฎตามความต้องการ<br>- อัพเดทกฎเพื่อให้แต่ละเซ็นเซอร์ได้ดึงไปใช้งาน |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 3.3 คำอธิบายยูสเคสไดอะแกรม Maintain Variable**

| ยูสเคส | Maintain Variable |
|---|---|
| วัตถุประสงค์ | เพื่อจัดการตัวแปลของสนอร์ทซึ่งถูกอ้างอิงมาจากกฎต่างๆเพื่อกำหนดให้กับแต่ละเซ็นเซอร์ (เช่นHOME_NET,EXTERNAL_NET ) รวทถึงการสร้างตัวแปลใดๆขึ้นมาใช้เองก็เป็นได้ |
| เงื่อนไขตอนเริ่มต้น | ข้อมูลของตัวแปลของสนอร์ท |
| เมื่อทำงานเสร็จ | แอดมินสามารถจัดการตัวแปลของสนอร์ทได้ |
| เมื่อทำงานไม่เสร็จ | ไม่สามารถจัดการตัวแปลของกฎของสนอร์ทได้(ต้องพิจารณาในส่วนของ permission databases) |
| แอกเตอร์ที่เกี่ยวข้อง | Administrator |
| สิ่งกระตุ้นการทำงาน | ข้อมูลของตัวแปลของสนอร์ท |
| อินพุต | ข้อมูลของตัวแปลของสนอร์ท |
| เอาท์พุต | จัดการและควบคุมตัวแปลของกฎของสนอร์ทได้ |
| รายละเอียด | - เลือกที่ เมนู Maintain Variable<br>- จัดการตัวแปลตามความต้องการ<br>- อัพเดทตัวแปลเพื่อให้แต่ละเซ็นเซอร์ได้ดึงไปใช้งาน |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

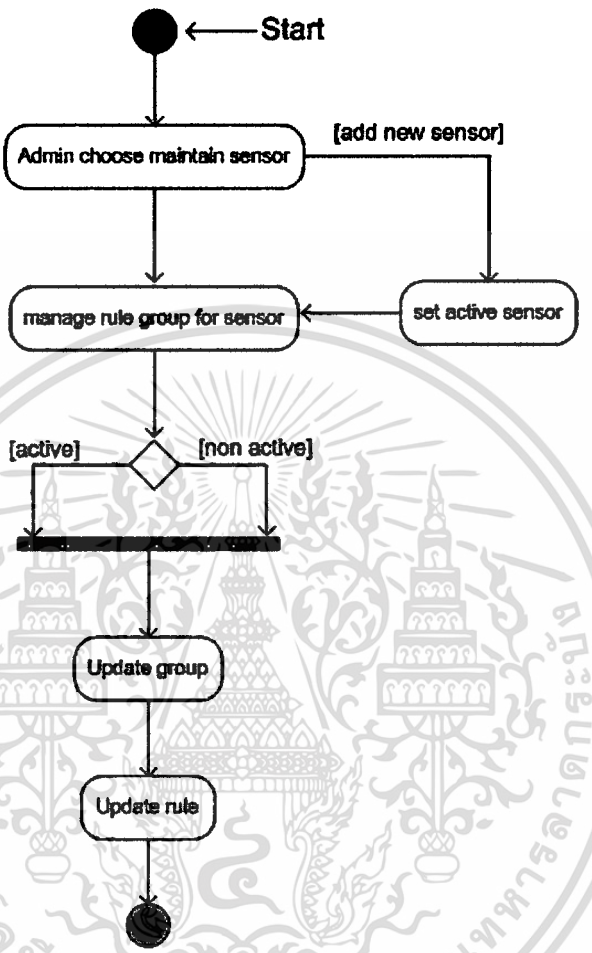
**ตารางที่ 3.4** คำอธิบายยูสเคสไดอะแกรม Maintain Preprocessor

| ยูสเคส | Maintain Preprocessor |
|---|---|
| วัตถุประสงค์ | เพื่อจัดการตัวแปลเกี่ยวกับ pre-processor ของสนอร์ทไม่ว่าจะเป็นการเพิ่มหรือลบรวมถึงการแก้ข้อข้อมูลต่างๆ เพื่อกำหนดให้กับแต่บละเซ็นเซอร์ |
| เงื่อนไขตอนเริ่มต้น | ข้อมูลของตัวแปล pre-processor ของสนอร์ท |
| เมื่อทำงานเสร็จ | แอดมินสามารถจัดการตัวแปล pre-processor ของสนอร์ทได้ |
| เมื่อทำงานไม่เสร็จ | ไม่สามารถจัดการตัวแปล pre-processor ของกฎของสนอร์ทได้ (ต้องพิจารณาในส่วนของ permission databases) |
| แอกเตอร์ที่เกี่ยวข้อง | Administrator |
| สิ่งกระตุ้นการทำงาน | ข้อมูลของตัวแปล pre-processor ของสนอร์ท |
| อินพุต | ข้อมูลของตัวแปล pre-processor ของสนอร์ท |
| เอาท์พุต | จัดการและควบคุมตัวแปล pre-processor ได้ |
| รายละเอียด | - เลือกที่ เมนู Maintain Preprocessor<br>- จัดการตัวแปลตาม pre-processor ความต้องการ<br>- อัพเดทตัวแปลเพื่อให้แต่ละเซ็นเซอร์ได้ดึงไปใช้งาน |

เมื่อรู้ถึงภาพรวมของโปรแกรมแล้ว ขั้นตอนต่อมาคือการศึกษาขั้นตอนการทำงานของแต่ละ Use Case และออกแบบขั้นตอนการทำงานของแต่ละ Use Case โดยสามารถแสดงขั้นตอนการทำงานผ่านทาง Activity Diagram โดยมีรายละเอียดดังต่อไปนี้

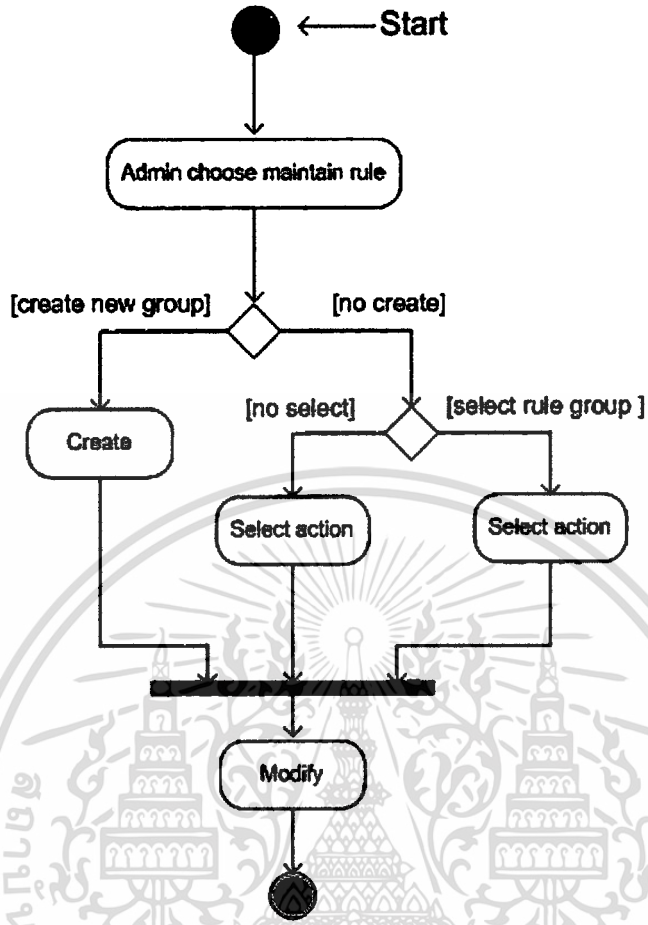
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.2 Activity Diagram

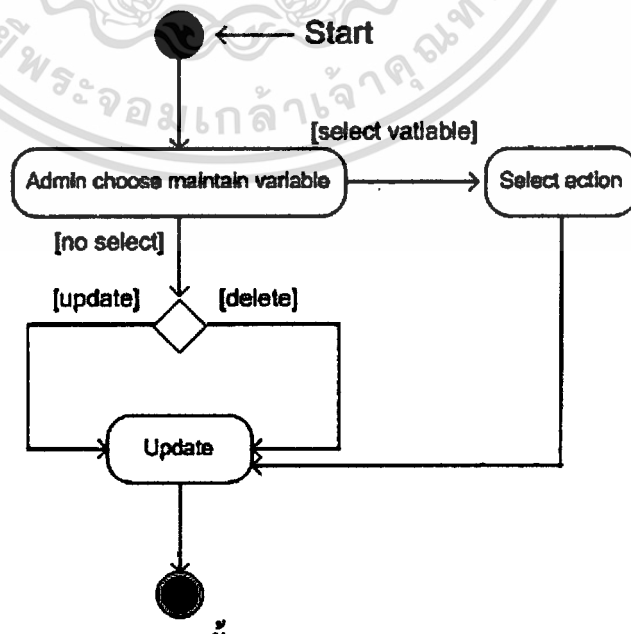



**รูปที่ 3.3** Activity Diagram **ของขั้นตอน** Maintain Sensor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

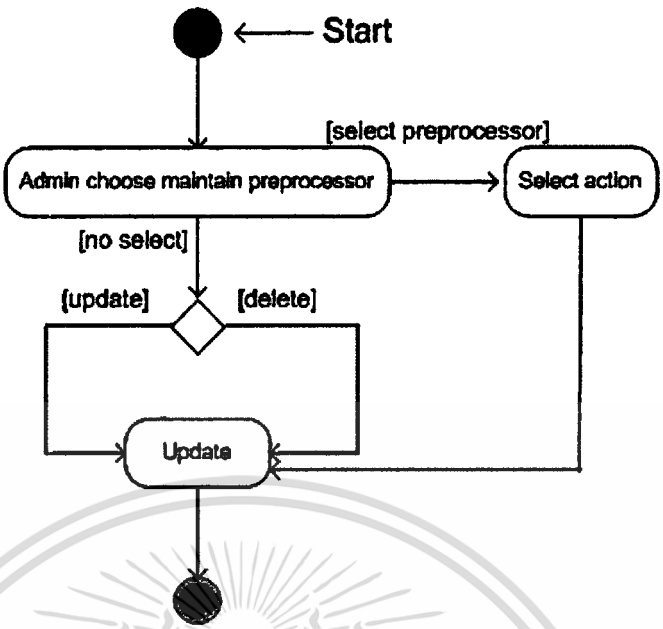



**รูปที่** 3.4 Activity Diagram **ของขั้นตอน** Maintain Rule




**รูปที่** 3.5 Activity Diagram **ของขั้นตอน** Maintain Variable

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

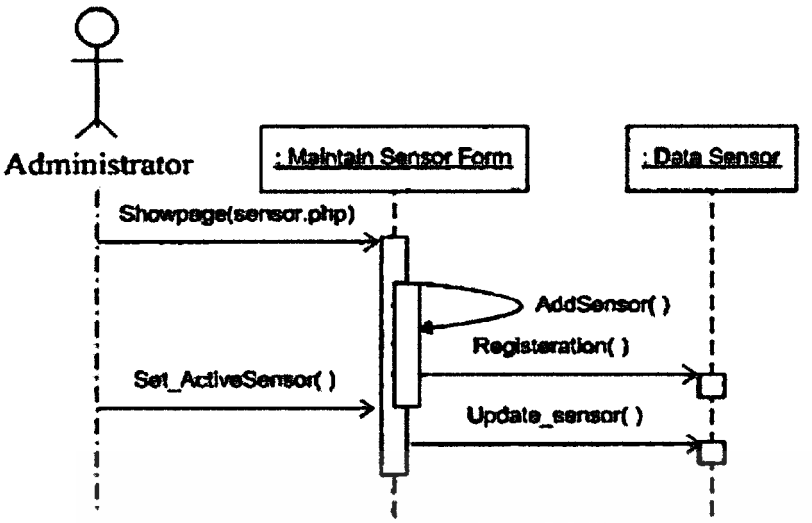



**รูปที่** 3.6 Activity Diagram **ของขั้นตอน** Maintain Preprocessor

### 3.2.3 Sequence Diagram

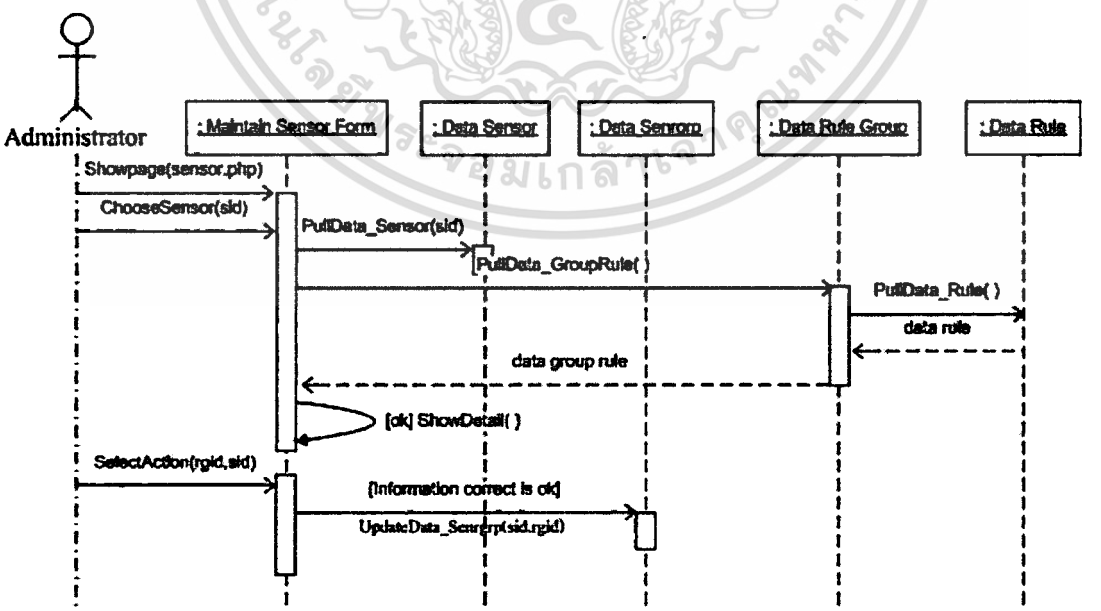
**ต่อไปจะอธิบายถึงลำดับขั้นตอนการทำงานของยูสเคสต่างๆ โดยใช้ซีเควนซ์ไดอะแกรม ดังนี้คือ** Maintain Sensor , Maintain Rule , Maintain Variable , Maintain Preprocessor **ซึ่งจะแสดงตามรูปที่** 3.6 **ถึง** 3.15 **ตามลำดับดังนี้**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.7** ซีเควนซ์ไดอะแกรมของยูสเคส Maintain Sensor [Add Sensor]

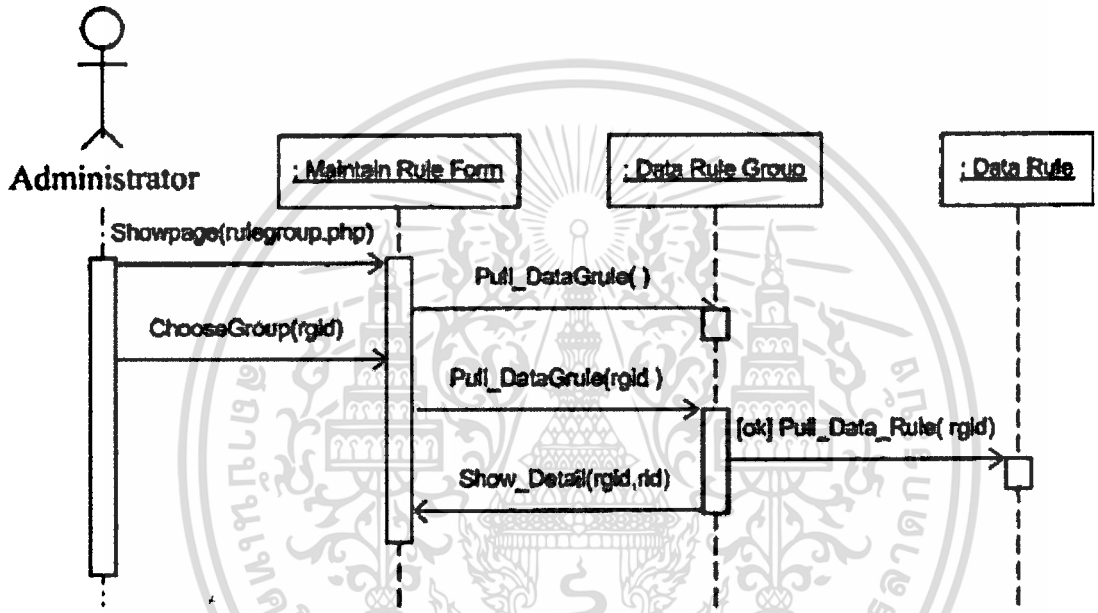
ซีเควนซ์ไดอะแกรมของยูสเคส Maintain Sensor ในเหตุการณ์ Add Sensor มีขั้นตอนคือ ผู้ดูแลระบบเลือกที่เมนู Maintain Sensor (ติดตั้งใหม่ยังไม่มีเซ็นเซอร์เลย) โดยดีฟอร์ระบบจะลิสเซ็นเซอร์ที่ได้รีจิสเตอร์ไว้แต่ตอนนี้ไม่มีเนื่องติดตั้งระบบใหม่ดังนั้นผู้ดูแลระบบจะต้องกดปุ่ม Add Sensor ที่อยู่ในหน้านี้และเลือกเช็คบ๊อกที่เซ็นเซอร์นั้นเพื่อแอดติเวทเซ็นเซอร์นั้นเพื่อจะได้กำหนดกฎหรือตัวแปลอื่นๆต่อไป




**รูปที่ 3.8** ซีเควนซ์ไดอะแกรมของยูสเคส Maintain Sensor [specify rule]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

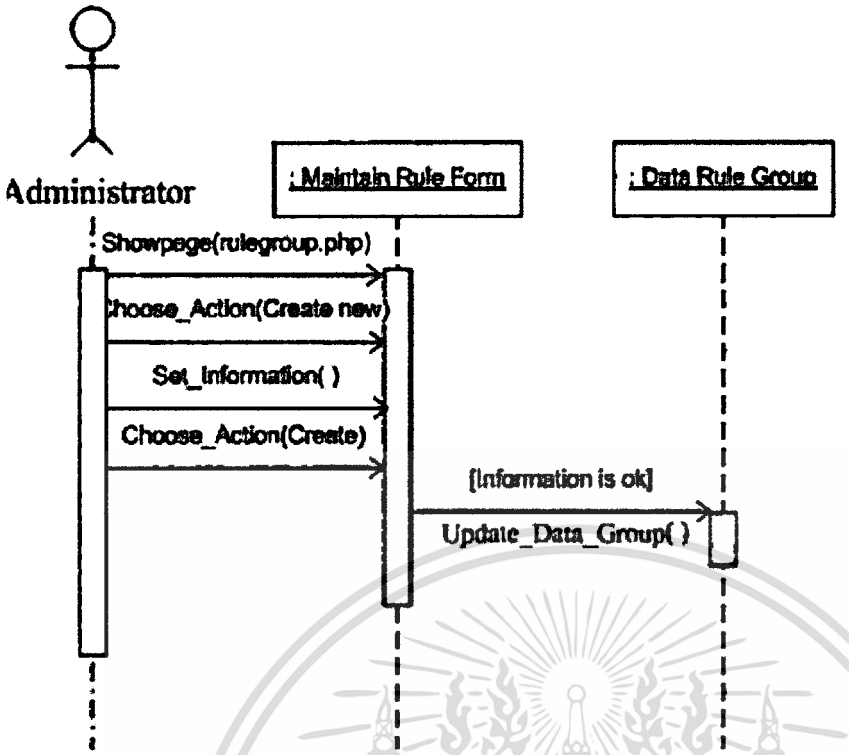
ซีเควนซ์ไดอะแกรมของยูสเคส Maintain Sensor ในเหตุการณ์การกำหนดกลุ่มของกฎของสนอร์ทให้กับแต่ละเซ็นเซอร์มีขั้นตอนการทำงานดังนี้ เมื่อผู้ดูแลระบบเลือกที่เมนู Maintain Sensor ในหน้านั้นโดยดีฟอร์จะลิสเซ็นเซอร์ทั้งหมดที่รีจิสเตอร์ไว้ขึ้นมา (จากยูสเคสที่แล้วเราได้เพิ่มเซ็นเซอร์ไปแล้ว) ผู้ดูแลระบบเลือกที่ชื่อเซ็นเซอร์ที่ต้องการ จากนั้นระบบจะลิสกลุ่มของกฎขึ้นมาทั้งหมด ต่อจากนั้นผู้ดูแลระบบก็เลือกเช็คบ๊อกที่กลุ่มของกฎนั้นเพื่อกำหนดให้กับเซ็นเซอร์นั้น




**รูปที่ 3.9 ซีเควนซ์ไดอะแกรมของยูสเคส Maintain Rule**

ซีเควนซ์ไดอะแกรมของยูสเคส Maintain Rule ในเหตุการณ์ที่ดูกฎต่างๆที่อยู่กลุ่มของกฎนั้นๆ มีขั้นตอนการทำงานดังนี้ ผู้ดูแลระบบเลือกที่เมนู Maintain Rule ระบบจะลิสรายการกลุ่มของกฎทั้งหมดขึ้นมา ต่อจากนั้นผู้ดูแลระบบก็เลือกที่ชื่อกลุ่มของกฎของสนอร์ท ระบบก็จะลิสรายการกฎทั้งหมดที่อยู่ในกลุ่มของกฎที่เลือกขึ้นมา

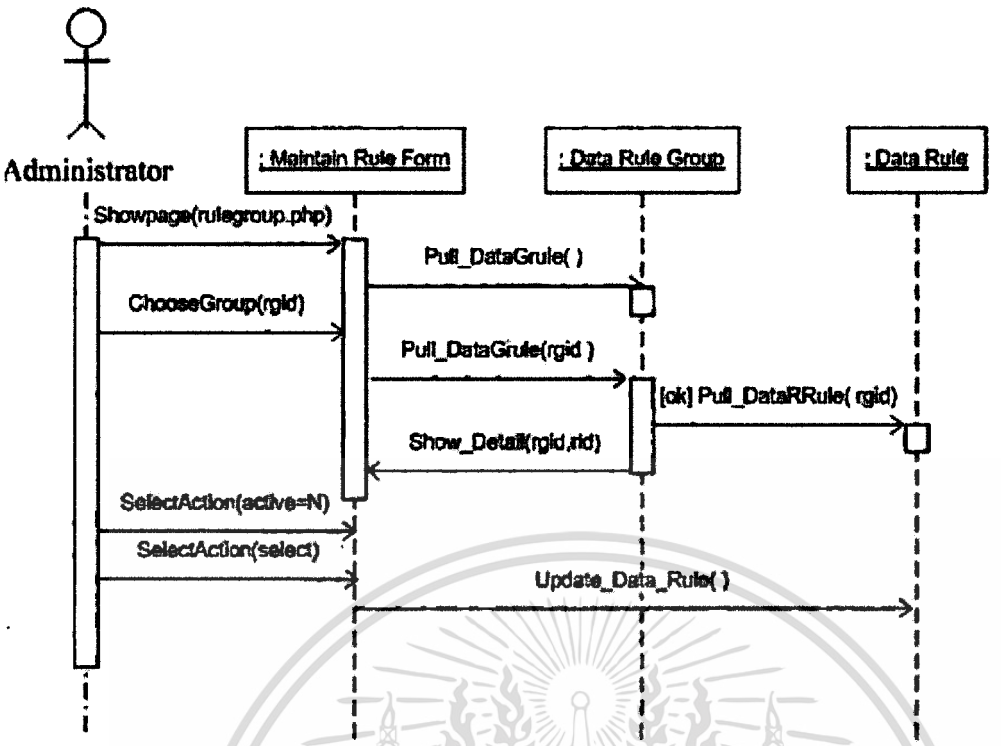
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.10** ซีเควนซ์ไดอะแกรมของยูสเคส Maintain Rule [Add group]

ซีเควนซ์ไดอะแกรมของยูสเคส Maintain Rule ในเหตุการณ์ที่ต้องการตั้งกลุ่มของกฎขึ้นมาใหม่ซึ่งมีขั้นตอนดังนี้ ผู้ดูแลระบบเลือกที่เมนู Maintain Rule ในหน้านั้นเองก็เลือกที่ปุ่ม Create New จากนั้นระบบจะให้ผู้ดูแลระบบใส่ข้อมูลที่จำเป็นและเมื่อใส่ข้อมูลเรียบร้อยแล้วก็เลือกที่ปุ่ม Create เพื่อยืนยันการสร้างกลุ่มของกฎ

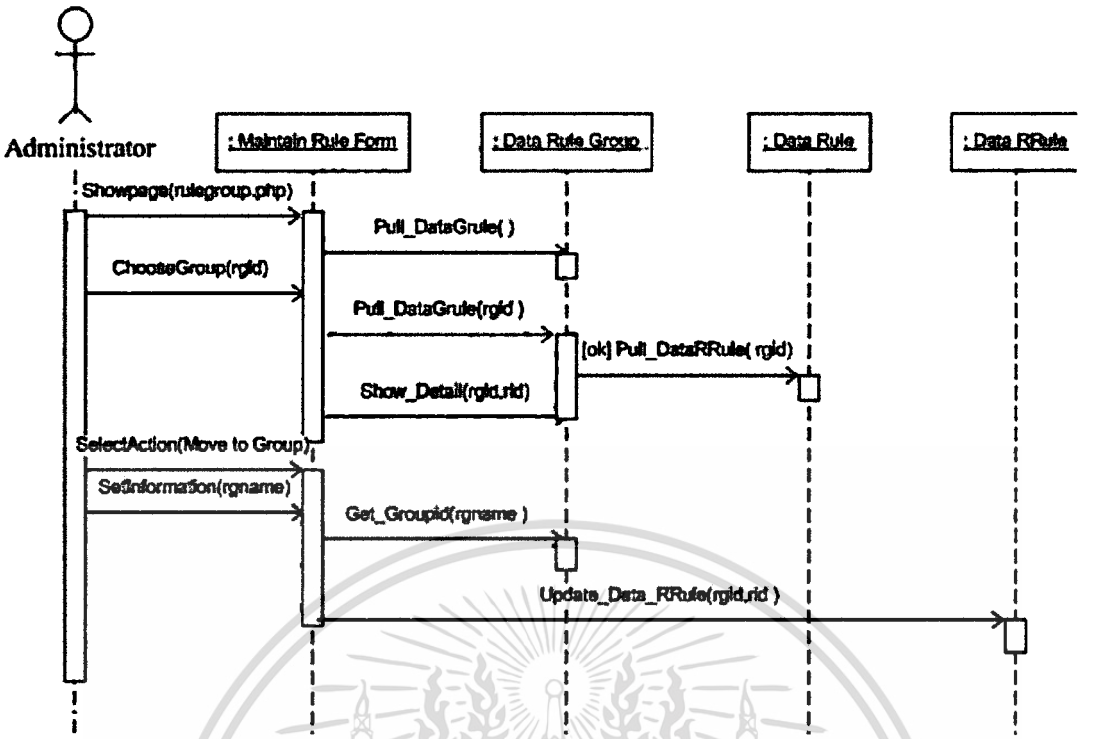
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.11 ซีเควนซ์ไดอะแกรมของยูสเคส Maintain Rule[disable rule]**

ซีเควนซ์ไดอะแกรมของยูสเคส Maintain Rule ในเหตุการณ์ที่จะดิสเอเบ็ลกฎนั้นๆ ซึ่งมีขั้นตอนดังนี้ ผู้ดูแลระบบเลือกที่เมนู Maintain Rule และเลือกที่ชื่อกลุ่มของกฎที่ต้องการ ต่อจากนั้นก็เลือกเช็คบ๊อกที่กฎนั้น และก็เลือกคำสั่งในลิสบ๊อกคือ Deactivate เพื่อดิสเอเบ็ลไม่ให้กฎนั้นถูกนำไปใช้ (ขั้นตอนเหมือนกันในส่วนของ Activate )

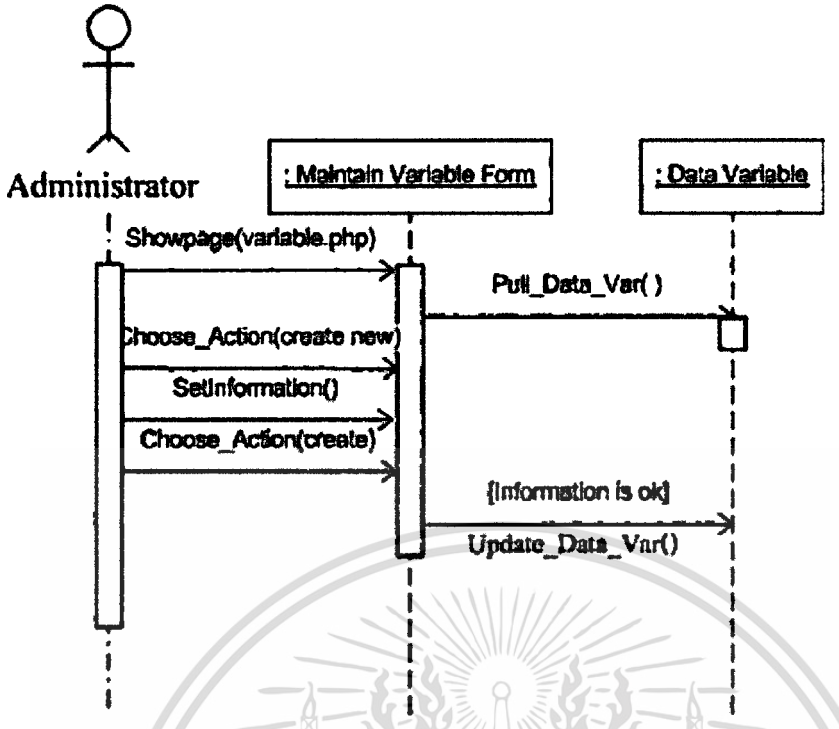
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.12 ซีเควนซ์ไดอะแกรมของยูสเคส Maintain Rule[move rule]**

ซีเควนซ์ไดอะแกรมของยูสเคส Maintain Rule ในเหตุการณ์ที่ย้ายกฎนั้นๆ ไปอยู่กลุ่มอื่น ซึ่งมีขั้นตอนดังนี้ ผู้ดูแลระบบเลือกที่เมนู Maintain Rule และเลือกที่ชื่อกลุ่มของกฎที่ต้องการ ต่อจากนั้นก็เลือกเช็คบ๊อกที่กฎนั้น และก็เลือกคำสั่งในลิสบ๊อกคือ Move to Group และผู้ดูแลระบบต้องระบุชื่อของกลุ่มของกฎใน text box ด้วยเพื่อย้ายกฎไปในกลุ่มนั้น

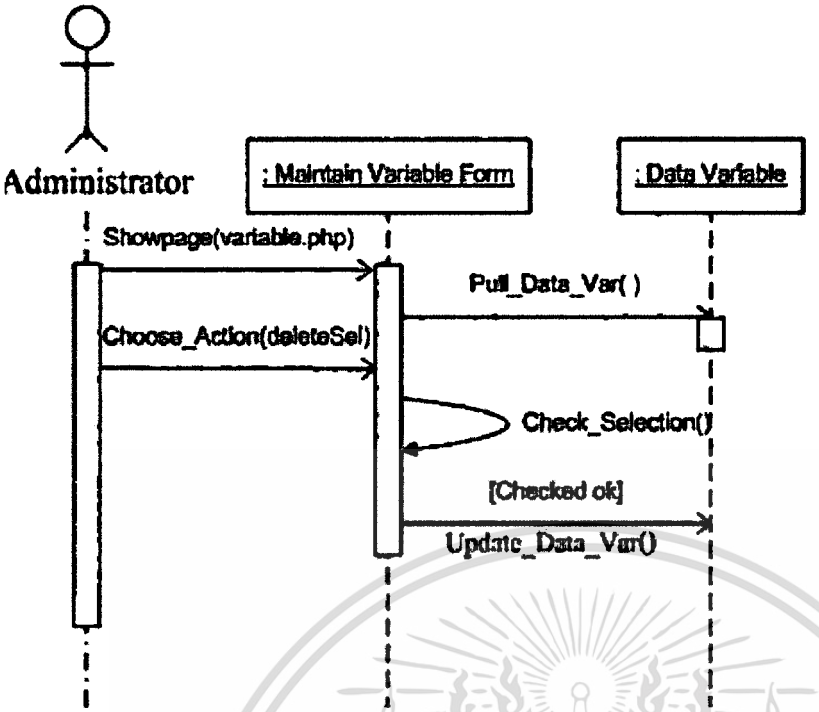
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.13 ซีเควนซ์ไดอะแกรมของยูสเคส Maintain Variable[add var]**

ซีเควนซ์ไดอะแกรมของยูสเคส Maintain Variable ในเหตุการณ์ที่เพิ่มตัวแปลใหม่มีขั้นตอนการทำงานดังนี้ ผู้ดูแลระบบเลือกที่เมนู Maintain Variable ในหน้านั้นเองผู้ดูแลระบบเลือกที่ปุ่ม Create New จากนั้นก็ระบุข้อมูลที่จำเป็น ต่อจากนั้นก็เลือกที่ปุ่ม Create ระบบก็จะเพิ่มตัวแปลนั้นในฐานข้อมูล

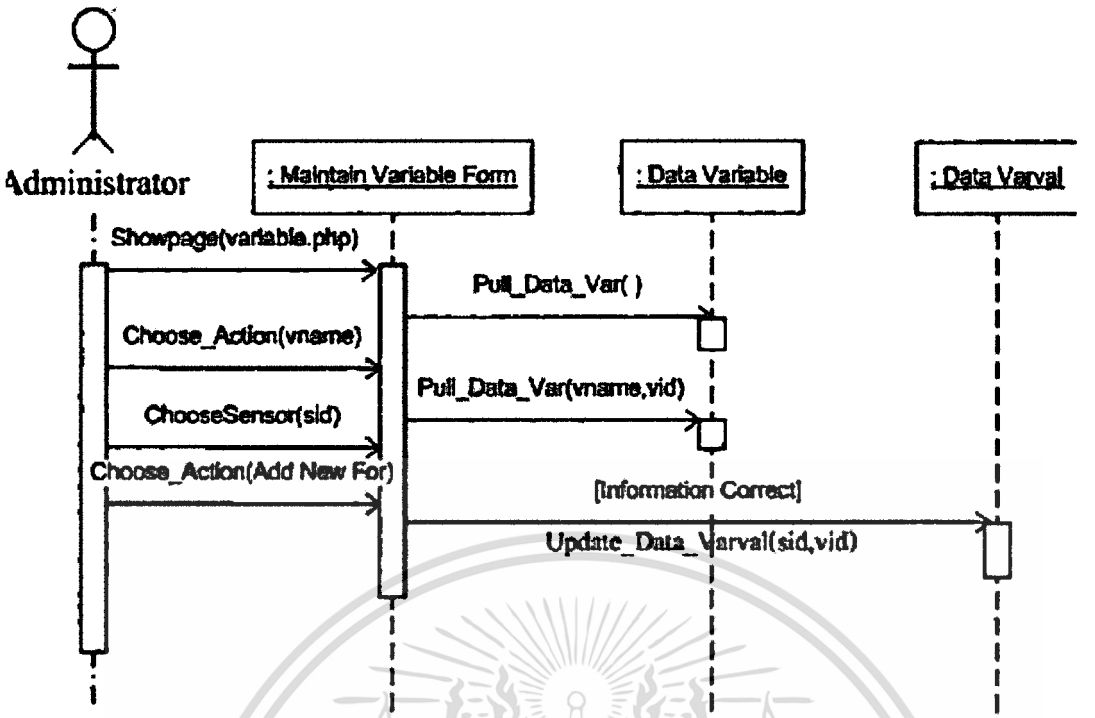
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.14 ซีเควนซ์ไดอะแกรมของยูสเคส Maintain Variable[del var]**

**ซีเควนซ์ไดอะแกรมของยูสเคส Maintain Variable ในเหตุการณ์ที่ลบตัวแปลนั้น ซึ่งมีขั้นตอนการทำงานดังนี้ ผู้ดูแลระบบเลือกที่เมนู Maintain Variable ในหน้านั้นเองผู้ดูแลระบบเลือกเช็คบ๊อกที่ตัวแปลนั้น จากนั้นผู้ดูแลระบบก็เลือกที่ปุ่ม Delete ระบบก็จะลบตัวแปลนั้นออกจากฐานข้อมูล**

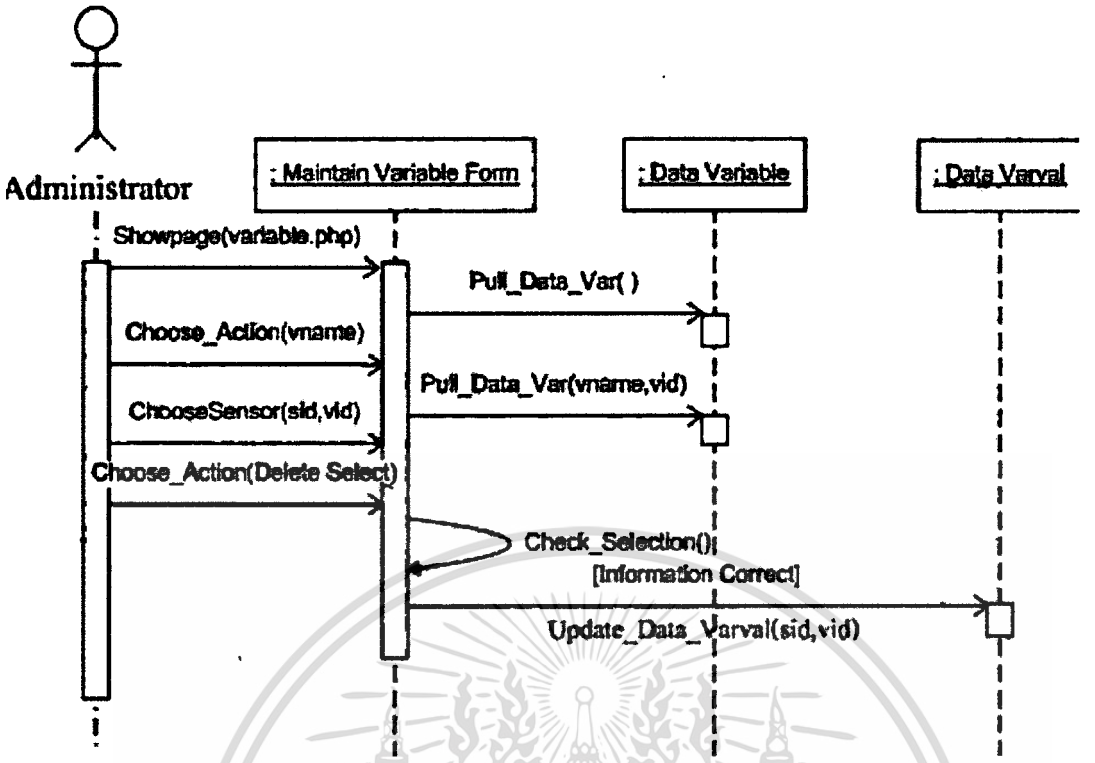
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.15 ซีเควนซ์ไดอะแกรมของยูสเคส Maintain Variable[use for sensor]**

ซีเควนซ์ไดอะแกรมของยูสเคส Maintain Variable ในเหตุการณ์ที่กำหนดความสัมพันธ์ระหว่างตัวแปลกับเซ็นเซอร์นั้น ซึ่งมีขั้นตอนการทำงานดังนี้ ผู้ดูแลระบบเลือกที่เมนู Maintain Variable ในหน้านั้นเองผู้ดูแลระบบเลือกเซ็นเซอร์ที่ลิสบ๊อก ต่อจากนั้นผู้ดูแลระบบก็เลือกที่ปุ่ม Add New For เพื่อเพิ่มความสัมพันธ์ของตัวแปลนั้น ระบบก็จะเพิ่มความสัมพันธ์ของตัวแปลกับเซ็นเซอร์นั้นในฐานข้อมูล

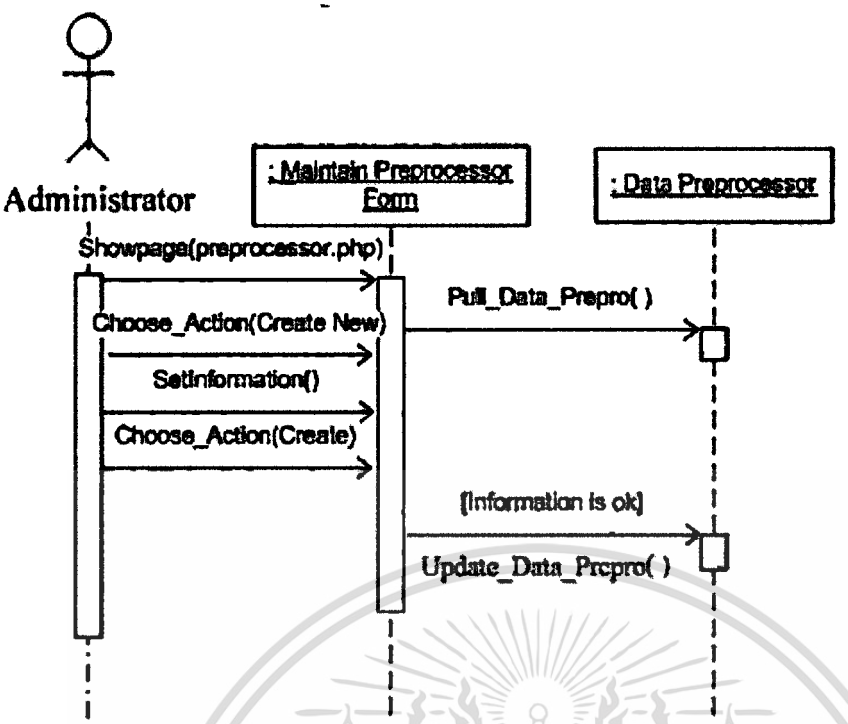
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.16** ซีเควนซ์ไดอะแกรมของยูสเคส Maintain Variable [del var for sensor]

ซีเควนซ์ไดอะแกรมของยูสเคส Maintain Variable ในเหตุการณ์ที่ลบความสัมพันธ์ระหว่างตัวแปลกับเซ็นเซอร์นั้น ซึ่งมีขั้นตอนการทำงานดังนี้ ผู้ดูแลระบบเลือกที่เมนู Maintain Variable ในหน้านั้นเองผู้ดูแลระบบเลือกเช็คบ๊อกที่ความสัมพันธ์ของตัวแปลนั้น จากนั้นผู้ดูแลระบบก็เลือกที่ปุ่ม Delete Select ระบบก็จะลบความสัมพันธ์ของตัวแปลกับเซ็นเซอร์นั้นออกจากฐานข้อมูล

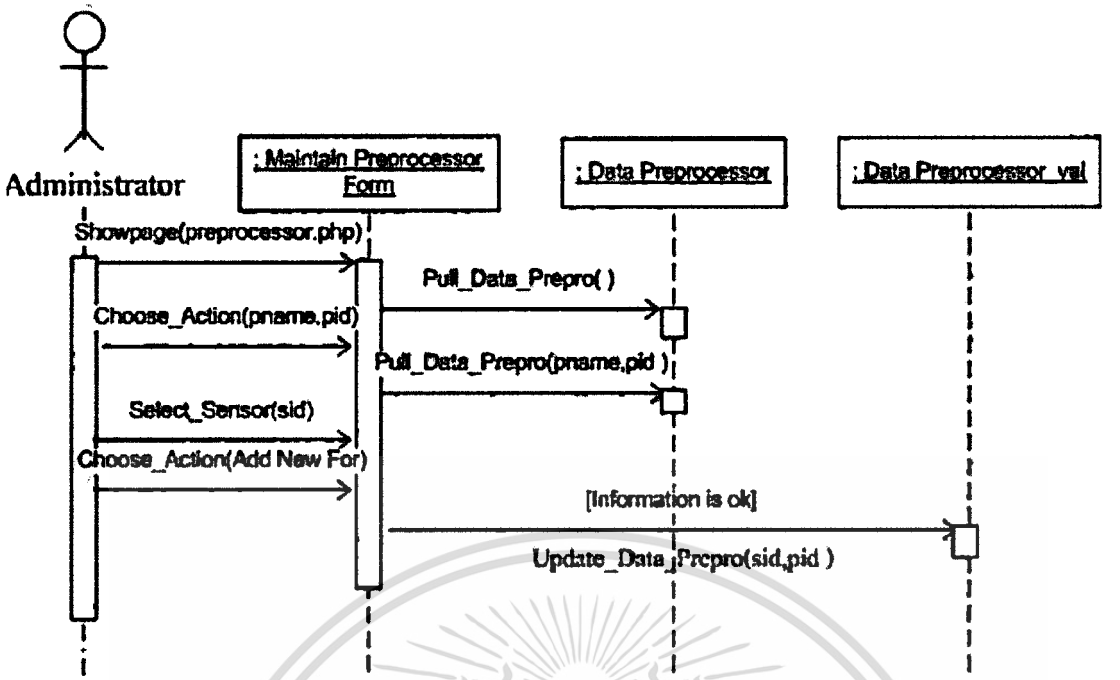
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.17** ซีเควนซ์ไดอะแกรมของยูสเคส Maintain Preprocessor

ซีเควนซ์ไดอะแกรมของยูสเคส Maintain Preprocessor ในเหตุการณ์ที่สร้างตัวแปลของ Preprocessor ใหม่ซึ่งมีขั้นตอนการทำงานดังนี้ ผู้ดูแลระบบเลือกที่เมนู Maintain Preprocessor ในหน้านั้นเองผู้ดูแลระบบเลือกที่ปุ่ม Create New จากนั้นผู้ดูแลระบบก็ระบุข้อมูลที่จำเป็นเข้าไป จากนั้นผู้ดูแลระบบก็เลือกที่ปุ่ม Create ระบบก็จะเพิ่มตัวแปล Preprocessor ในฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



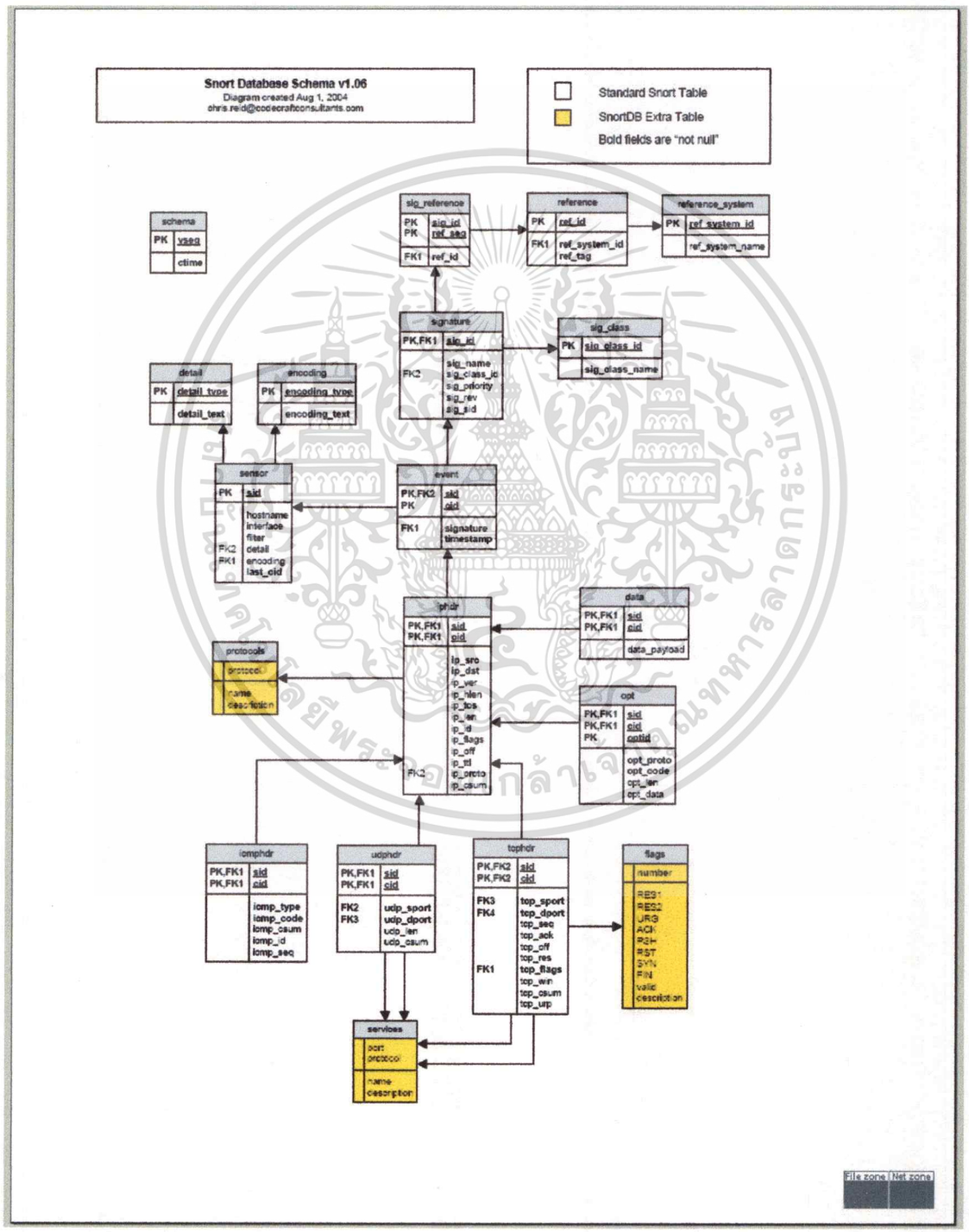

**รูปที่ 3.18 ซีเควนซ์ไดอะแกรมของยูสเคส Maintain Preprocessor**

ซีเควนซ์ไดอะแกรมของยูสเคส Maintain Variable ในเหตุการณ์ที่กำหนดความสัมพันธ์ระหว่างตัวแปล Preprocessor กับเซ็นเซอร์นั้น ซึ่งมีขั้นตอนการทำงานดังนี้ ผู้ดูแลระบบเลือกที่เมนู Maintain Preprocessor ในหน้านั้นเองผู้ดูแลระบบเลือกเซ็นเซอร์ที่ลิสบ๊อก ต่อจากนั้นผู้ดูแลระบบก็เลือกที่ปุ่ม Add New For เพื่อเพิ่มความสัมพันธ์ของตัวแปลนั้น ระบบก็จะเพิ่มความสัมพันธ์ของตัวแปล Preprocessor กับเซ็นเซอร์นั้นในฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3. ส่วนของ IPS Administration Tool

จะเพิ่ม table เข้าไปในฐานข้อมูลเดิมของ snort อีก 7 table โดยเป็นของ เว็บแอ็พพลิเคชั่น ซึ่งไม่ได้มีส่วนเกี่ยวข้องกับการทำงานของ snort เลยและจากรูปด้านล่างนี้คือ class diagram ของ สนอร์ท ซึ่งจากการศึกษาทั้งสนอร์ทและสนอร์ทอินไลน์ นั้นใช้ schema ในการจัดการเดียวกัน ดังนี้




**รูปที่ 3.19 แสดงโครงสร้างของฐานข้อมูลที่เซ็นเซอร์ใช้จัดเก็บ**

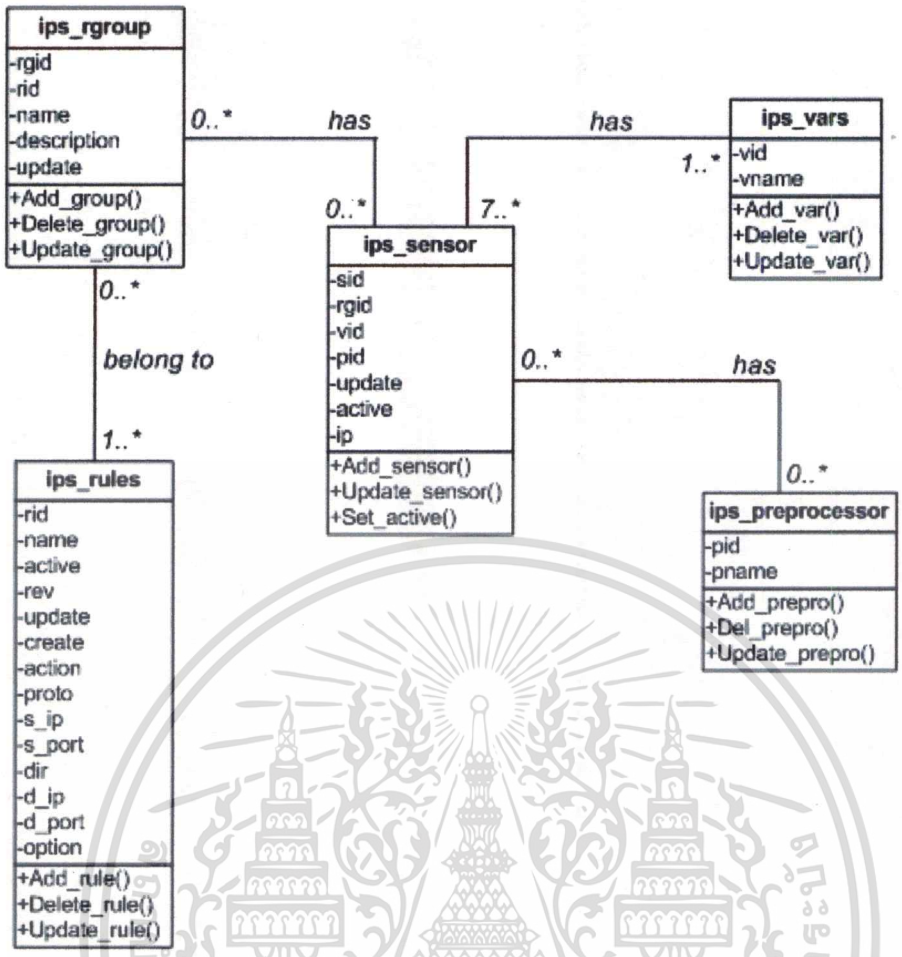
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.7 นั้นจะเป็นโครงสร้างของฐานข้อมูลที่เซ็นเซอร์ใช้ ซึ่งประกอบไปด้วยตาราง ดังต่อไปนี้

**ตารางที่ 3.5** แสดงตารางของสนอร์ทในการจัดเก็บข้อมูลการบุกรุก

| ตาราง | รายละเอียด |
|---|---|
| data | เก็บข้อมูล payload ของการบุกรุก |
| detail | ข้อมูลโหมดการจัดเก็บ (Full , Fast) |
| encoding | เก็บประเภทการจัดเก็บ (hex,base64,ascii) |
| event | เก็บจัดเก็บข้อมูลที่เกี่ยวข้องกับเหตุการณ์ (ช่วงเวลาที่เกิด,ประเภทของเหตุการณ์) |
| icmphdr | เก็บข้อมูลส่วนหัวของโปรโตคอล icmp |
| iphdr | เก็บข้อมูลส่วนหัวของโปรโตคอล ip |
| opt | เก็บข้อมูลออปชั่นเพิ่มเติม |
| reference | เก็บข้อมูลการอ้างอิงไปยังเวบไซต์ที่เกี่ยวข้อง |
| reference_system | เก็บข้อมูลประเภทการอ้างอิง (url,cve,bugtraq,nessus) |
| schema | เก็บข้อมูลเวอร์ชันของโครงสร้างฐานข้อมูล |
| sensor | เก็บข้อมูลของเซ็นเซอร์ |
| sig_class | เก็บข้อมูลประเภทของรูปแบบการบุกรุก |
| sig_reference | เก็บข้อมูลการอ้างอิงที่เกี่ยวข้องกับประเภทของการบุกรุก |
| signature | เก็บข้อมูลรูปแบบการบุกรุกที่ตรวจพบ |
| tcphdr | เก็บข้อมูลส่วนหัวของโปรโตคอล tcp |
| udphdr | เก็บข้อมูลส่วนหัวของโปรโตคอล udp |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

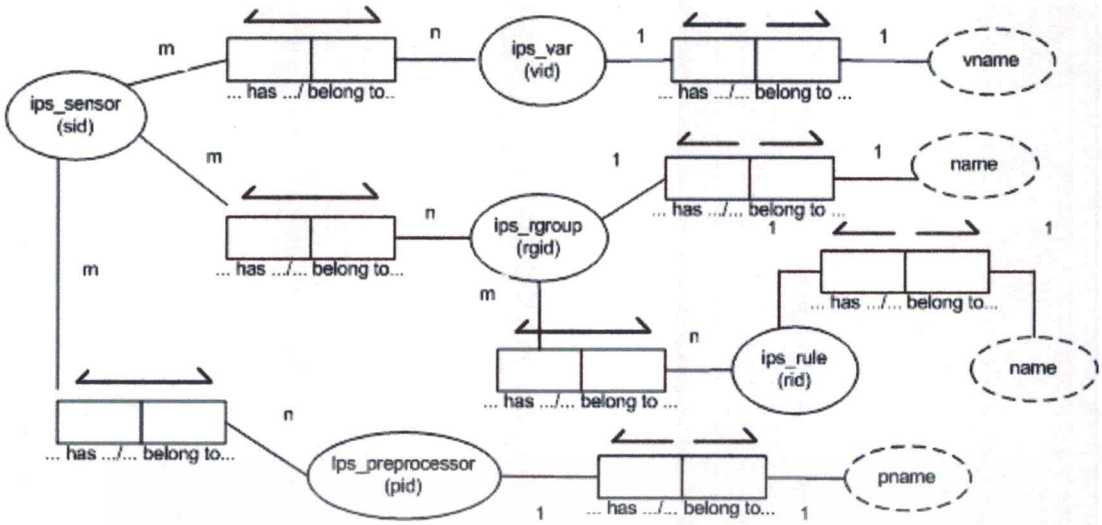



**รูปที่ 3.20** class diagram ของ web ips tool

**การออกแบบฐานข้อมูล**

ตามที่ได้ทำการออกแบบซึ่งได้เพิ่มเข้าไปในฐานข้อมูลของ snort_inline ซึ่งไม่ได้เปลี่ยนแปลงหรือแก้ไดๆทั้งสิ้นในฐานข้อมูลของ snort_inline เดิมเพียงแต่จะใช้อ้างอิงจากตาราง sensor ของ snort inline เพื่ออ้างอิงในส่วนของ sensor โดยอาศัยแบบจำลองข้อมูลเพื่อแสดงรายละเอียดต่างๆที่เกี่ยวข้องกันในฐานข้อมูลโดยใช้แบบจำลอง ORM ซึ่งแสดงได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.21** Object Role Model

จากการออกแบบด้วยแบบจำลอง Object Role Model จะสามารถแม็ปออกมาได้ 5 NF เลยซึ่งจะต่างจาก Er-Diagram ที่แม็ปออกมาได้เพียงแค่ 1NF เท่านั้นต่อจากนั้นเราจะแม็ปออกมาเป็นแต่ละ table (database schema) ดังนี้

**Database Schema**

1. **ips_sensor** (sid , updated , active , ip )
2. **ips_senrgrp** ( sid , rgid )
3. **ips_preprocessor** (pid , pname)
4. **ip_preprocessorvals** (pid , sid , options , comment , updated )
5. **ips_rules** (rid , name , active , rev , updated , created , action , proto , s_ip , s_port , dir , d_ip , d_port , option )
6. **ips_rrgid** (rid , rgid)
7. **ip_rgroup** (rgid , name , description , updated)
8. **ips_var** (vid , vname)
9. **ips_varval** (vid , sid , value , comment , updated)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อจากนั้นจะแสดงรายละเอียดของแต่ละ schema ดังตาราง

**ตารางที่ 3.6** รายละเอียดของ schema ips_sensor

| ชื่อ | คำอธิบาย | ชนิดข้อมูล | คีย์ | ตารางอ้างถึง |
|---|---|---|---|---|
| sid | รหัสเซ็นเซอร์ | int(11) | PK | ips_senrgrp, ip_preprocessorvals , ips_varval |
| updated | เวลาที่รีจิสเข้ามา | timestamp | | |
| active | สถานะเซ็นเซอร์ | Enum | | |
| ip | ไอพีเซ็นเซอร์ | varchar(15) | | |

**ตารางที่ 3.7** รายละเอียดของ schema ips_senrgrp

| ชื่อ | คำอธิบาย | ชนิดข้อมูล | คีย์ | ตารางอ้างถึง |
|---|---|---|---|---|
| sid | รหัสเซ็นเซอร์ | int(11) | PK | ips_senrgrp, ip_preprocessorvals , ips_varval |
| rgid | รหัสกลุ่มกฎ | int(11) | PK | ips_rrgid , ip_rgroup |

**ตารางที่ 3.8** รายละเอียดของ schema ips_preprocessor

| ชื่อ | คำอธิบาย | ชนิดข้อมูล | คีย์ | ตารางอ้างถึง |
|---|---|---|---|---|
| pid | รหัสพลีโพรเซสเซอร์ | int(11) | PK | ips_senrgrp, ip_preprocessorvals , ips_varval |
| pname | ชื่อพลีโพรเซสเซอร์ | Varchar(30) | | ip_preprocessorvals |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 3.9** รายละเอียดของ schema ip_preprocessorvals

| ชื่อ | คำอธิบาย | ชนิดข้อมูล | คีย์ | ตารางอ้างถึง |
|---|---|---|---|---|
| pid | รหัสพลีโพรเซสเซอร์ | int(11) | PK | ips_senrgrp, ip_preprocessorvals , ips_varval |
| sid | รหัสเซ็นเซอร์ | int(11) | PK | ips_senrgrp, ip_preprocessorvals , ips_varval |
| options | ข้อมูลออปชั่น | Varchar(255) | | |
| comment | ข้อมูลคอมเม็น | Varchar(255) | | |
| updated | เวลาการเปลี่ยนแปลง | timestamp | | |

**ตารางที่ 3.10** รายละเอียดของ schema ips_rules

| ชื่อ | คำอธิบาย | ชนิดข้อมูล | คีย์ | ตารางอ้างถึง |
|---|---|---|---|---|
| rid | รหัสกฎ | int(11) | PK | ips_rrgid |
| name | ชื่อกฎ | Varchar(255) | | |
| active | สถานะกฎ | enum | | |
| rev | การแก้ไข(ครั้ง) | int(11) | | |
| updated | เวลาเปลี่ยนแปลง | timestamp | | |
| created | เวลาสร้างกฎ | timestamp | | |
| action | เงื่อนไข | Varchar(30) | | |
| proto | ชนิดโพรโตคอล | Varchar(30) | | |
| s_ip | ไอพีต้นทาง | Varchar(255) | | |
| s_port | พอร์ทต้นทาง | Varchar(30) | | |
| dir | ทิศทางเงื่อนไข | enum | | |
| d_ip | ไอพีปลายทาง | Varchar(255) | | |
| d_port | พอร์ทปลายทาง | Varchar(30) | | |
| option | ข้อมูลกฎ | blob | | |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 3.11** รายละเอียดของ schema ips_rrgid

| ชื่อ | คำอธิบาย | ชนิดข้อมูล | คีย์ | ตารางอ้างถึง |
|---|---|---|---|---|
| rid | รหัสกฎ | int(11) | PK | ips_rules |
| rgid | รหัสกกลุ่มกฎ | int(11) | PK | ip_rgroup |

**ตารางที่ 3.12** รายละเอียดของ schema ips_rgroup

| ชื่อ | คำอธิบาย | ชนิดข้อมูล | คีย์ | ตารางอ้างถึง |
|---|---|---|---|---|
| rgid | รหัสกกลุ่มกฎ | int(11) | PK | ips_rrgid , ips_senrgrp |
| name | ชื่อกลุ่มกฎ | Varchar(30) | | |
| description | ข้อมูลกลุ่มกฎ | Varchar(255) | | |
| updated | เวลาการเปลี่ยนแปลง | timestamp | | |

**ตารางที่ 3.13** รายละเอียดของ schema ips_var

| ชื่อ | คำอธิบาย | ชนิดข้อมูล | คีย์ | ตารางอ้างถึง |
|---|---|---|---|---|
| vid | รหัสตัวแปลระบบ | int(11) | PK | ips_varval |
| vname | ชื่อตัวแปลระบบ | Varchar(30) | | |

**ตารางที่ 3.14** รายละเอียดของ schema ips_varval

| ชื่อ | คำอธิบาย | ชนิดข้อมูล | คีย์ | ตารางอ้างถึง |
|---|---|---|---|---|
| vid | รหัสตัวแปลระบบ | int(11) | PK | ips_var |
| sid | รหัสเซ็นเซอร์ | int(11) | PK | ips_senrgrp, ip_preprocessorvals , ips_varval |
| vlue | ข้อมูลกฎ | Varchar(255) | | |
| comment | ข้อมูลตัวแปล | Varchar(255) | | |
| updated | เวลาการเปลี่ยนแปลง | timestamp | | |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.2. ส่วนของ File Config ของ Snort_inline

กล่าวคือการปรับแต่ง config file เดิมเพื่อให้ทำงานตามที่เราต้องการซึ่งจะมี 3 ไฟล์ ดังนี้

- **db.config** เป็นไฟล์ ที่ใช้ในการบอกให้ สนอร์ทอินไลน์ชี้ไปที่เครื่องดาต้าเบส เซิร์ฟเวอร์ไหนดาต้าเบสชื่ออะไรเซ็นเซอร์ชื่ออะไร เป็นต้น มีรูปแบบดังนี้

```
#
## dbserv: 192.168.0.4
## dbname: snort
## sensor_name: pae_smart
## senintf: Br0
## dbuser: snort
## dbpasswd: snort
#
output database: log, mysql, user=snort password=snort dbname=snort host=192.168.0.4
sensor_name=pae_smart
```

**รูปที่ 3.22** รายละเอียด db.config

- **db.rules** เป็น text file ธรรมดาที่ดึงมาจากดาต้าเบสเซิร์ฟเวอร์ตามกฎที่เราได้เลือกไว้
- **db.vars** เป็น text file ที่ดึงมาจากดาต้าเบสเซิร์ฟเวอร์ที่เราได้เลือก variable แต่ละตัวไว้
- **db.preprocessor vars** เป็น text file ที่ดึงมาจากดาต้าเบสเซิร์ฟเวอร์ตามที่เราได้เลือก variable ที่เกี่ยวกับ preprocessor แต่ละตัวไว้

ซึ่งไฟล์ทั้งหมดนี้จะถูกเรียกใช้งานจากสคริปต์ที่ชื่อว่า extractrule.pl

### 3.2.3. ส่วนของ script perl ที่ควบคุมการทำงานของ snort_inline

Script perl ที่ได้พัฒนาขึ้นมานั้นมี 2 สคริปต์

- **loadrule.pl** เป็นสคริปต์ perl ที่พัฒนาขึ้นมาเพื่อใช้ปั้มกฎของสนอร์ททั้งหมดลงดาต้าเบสโดยที่แยกประเภทของกฎด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

alert | tcp | $EXTERNAL_NET | any | -> | $TELNET_SERVERS | 23 | (msg:"TELNET Solaris memory mismanagement exploit attempt"; flow:to_server,established; content:"|A0 23 A0 10 AE 23 80 10 EE 23 BF EC 82 05 E0 D6 90|%|E0|"; classtype:shellcode-detect; sid:1430; rev:7;)

**รูปที่ 3.23 รายละเอียด telnet.rule**

ให้พิจารณาตรงส่วนที่ผมได้มีการแบ่งด้วยเครื่องหมาย ( | ) และที่ขีดเส้นใต้ในแต่ละส่วนใน ruleเพราะฉะนั้นในที่ผมได้แบ่งเป็น 8 ส่วนด้วยกันคือ

**ส่วนที่ 1.** เก็บค่า action ของกฎ

**ส่วนที่ 2.** เก็บค่า protocol ของกฎ

**ส่วนที่ 3.** เก็บค่า source ip ของ packet ที่ต้องการ

**ส่วนที่ 4.** เก็บค่า source port ของ packet ที่ต้องการ

**ส่วนที่ 5.** เก็บค่า direction

**ส่วนที่ 6.** เก็บค่า destination ip (server)

**ส่วนที่ 7.** เก็บค่า destination port

**ส่วนที่ 8.** เก็บค่า option ต่างๆ

โดยที่ในการใช้งานจริงนั้นตอนที่เราได้ download snort_inline มาจากทางเว็บของ snort นั้น rule ต่างๆของมันยังใช้เป็นแบบเดิมอยู่คือเป็นแบบ snort (ids) ดังนั้นเราจึงต้องทำการเปลี่ยน rule ให้เป็นแบบ ips เสียก่อนกล่าวคือความสามารถของ snort inline ในการ drop , reject , replace และในที่นี้ผมได้เปลี่ยน rule ทุก rule ของ snort_inline ในส่วนของ action ให้เป็น drop ทั้งหมดทุก rule เพื่อนำเสนอการทำงานในลักษณะ ips ครับจะได้ดังนี้

**drop** tcp $EXTERNAL_NET any -> $TELNET_SERVERS 23 (msg:"TELNET Solaris memory mismanagement exploit attempt"; flow:to_server,established; content:"|A0 23 A0 10 AE 23 80 10 EE 23 BF EC 82 05 E0 D6 90|%|E0|"; classtype:shellcode-detect; sid:1430; rev:7;)

**Telnet.rule (action=drop) ของ snort**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **extractrule.pl** เป็น script perl ที่พัฒนาขึ้นเพื่อควบคุมการทำงานของ snort_inline เช่นสั่ง start หรือ restart snort_inline , ตรวจสอบการทำงานของ snort , ดึงกฎของ snort_inline มาจาก database มาเก็บไว้เป็นต้น

ซึ่ง file ที่อธิบายไว้ทั้งในหัวข้อที่ 3.3.2 และ 3.2.3 จะต้องอ้างอิงเอาไว้ใน file config ของ snort_inline ด้วยคือ snort_inline.conf ซึ่งมีรูปแบบดังนี้

**** ตัดรายละเอียดในส่วนบนออกเพื่อความกระชับ ***

```
# Honeynet snort_inline configuration file
# Version 0.5
# Last modified 01 January, 2004
# Standard Snort configuration file modified for inline
# use.  Most preprocessors currently do not work in inline
# mode, as such they are not included.
### Network variables
##var HOME_NET 10.0.0.0
var HONEYNET any
var EXTERNAL_NET any
var SMTP_SERVERS any
var TELNET_SERVERS any
var HTTP_SERVERS any
var SQL_SERVERS any
# Ports you run web servers on
# Please note:  [80,8080] does not work.
# If you wish to define multiple HTTP ports,
## var HTTP_PORTS 80
## include somefile.rules
## var HTTP_PORTS 8080
## include somefile.rules
var HTTP_PORTS 80
# Ports you want to look for SHELLCODE on.
var SHELLCODE_PORTS !80
### Logging alerts of outbound attacks
```

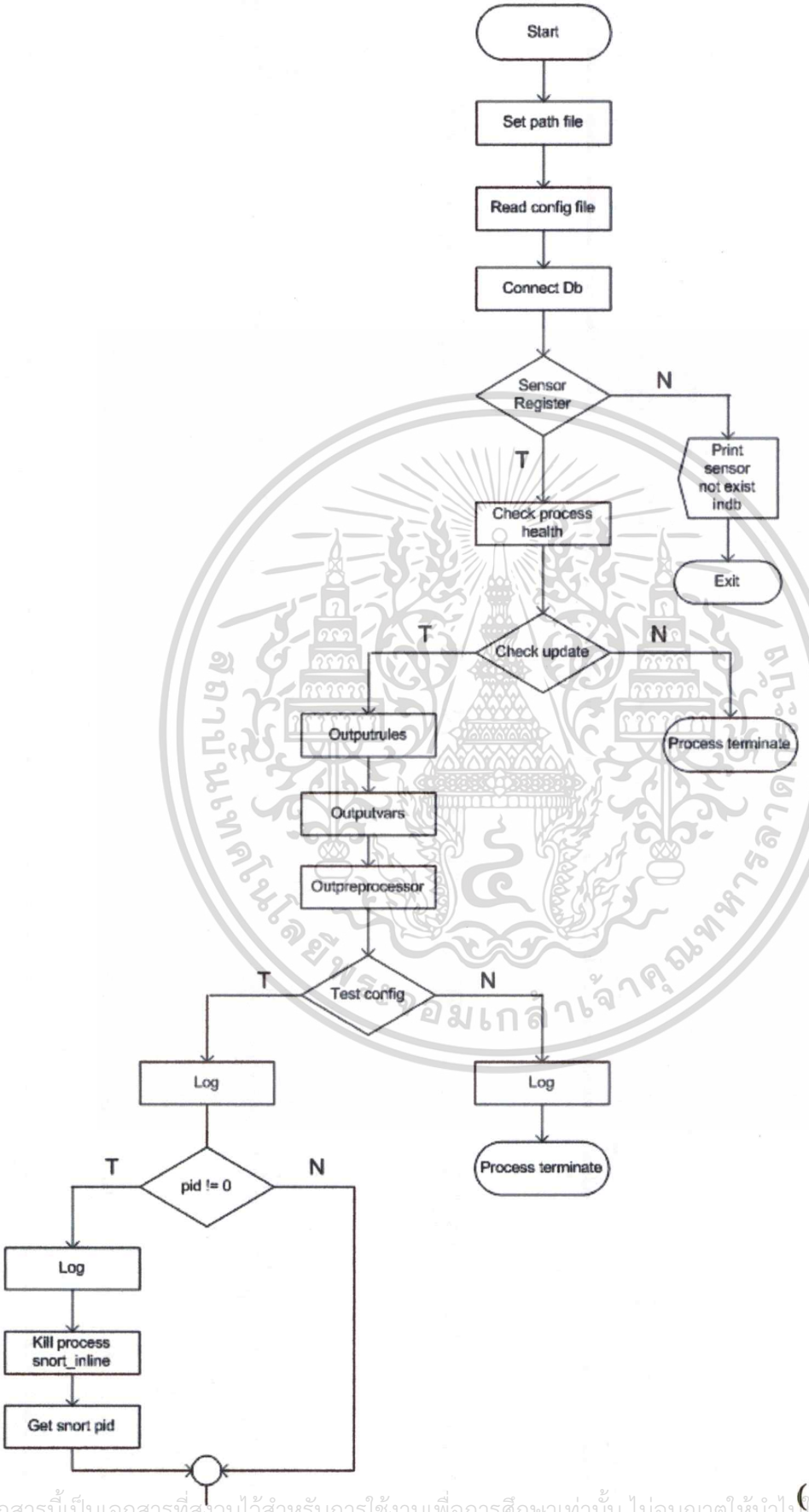
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
output alert_full: snort_inline-full
output alert_fast: snort_inline-fast
include db.vars
include db.config
#output database: alert, mysql, user=snort password=snort dbname=snort host=localhost
### If you want to log the contents of the dropped packets, remove comment
#output log_tcpdump: tcpdump.log
# Include classification & priority settings
include $RULE_PATH/classification.config
include $RULE_PATH/reference.config
include db.rules
### The Drop Rules
# Enabled
#include $RULE_PATH/exploit.rules
#include $RULE_PATH/finger.rules
#include $RULE_PATH/ftp.rules
#include $RULE_PATH/telnet.rules
#include $RULE_PATH/rpc.rules
#include $RULE_PATH/rservices.rules
#include $RULE_PATH/dos.rules
#include $RULE_PATH/ddos.rules
#include $RULE_PATH/dns.rules
#include $RULE_PATH/tftp.rules
#include $RULE_PATH/web-cgi.rules
#include $RULE_PATH/web-coldfusion.rules
#include $RULE_PATH/web-iis.rules
#include $RULE_PATH/web-frontpage.rules
#include $RULE_PATH/web-misc.rules
#include $RULE_PATH/web-client.rules
#include $RULE_PATH/web-php.rules
**** ตัดรายละเอียดออกเพื่อความกระชับ ***
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

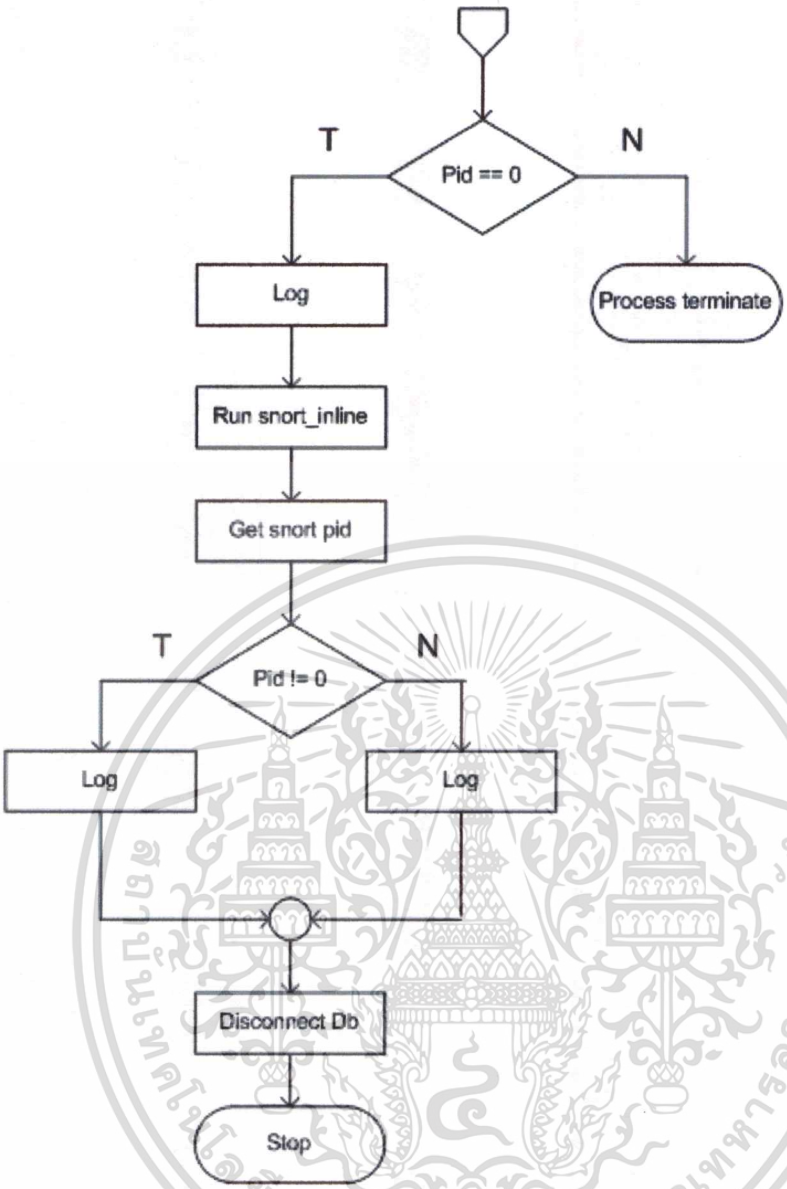
**รูปที่ 3.24 ตัวอย่างไฟล์คอนฟิกของสนอรท์**

## 3.4. Flow Chart ของทั้ง 2 script ของ Process



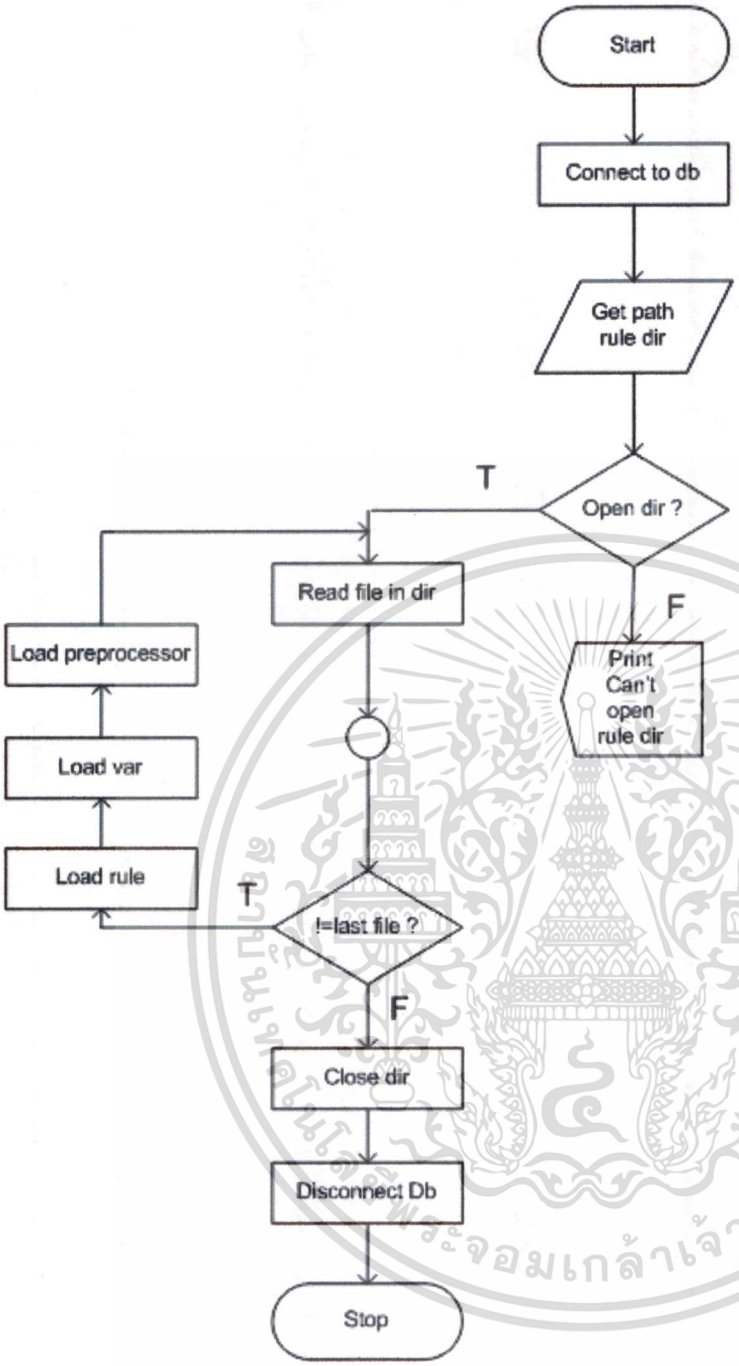

( ต่อ )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.25** Main Flow Chart Script Extractrule.pl

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

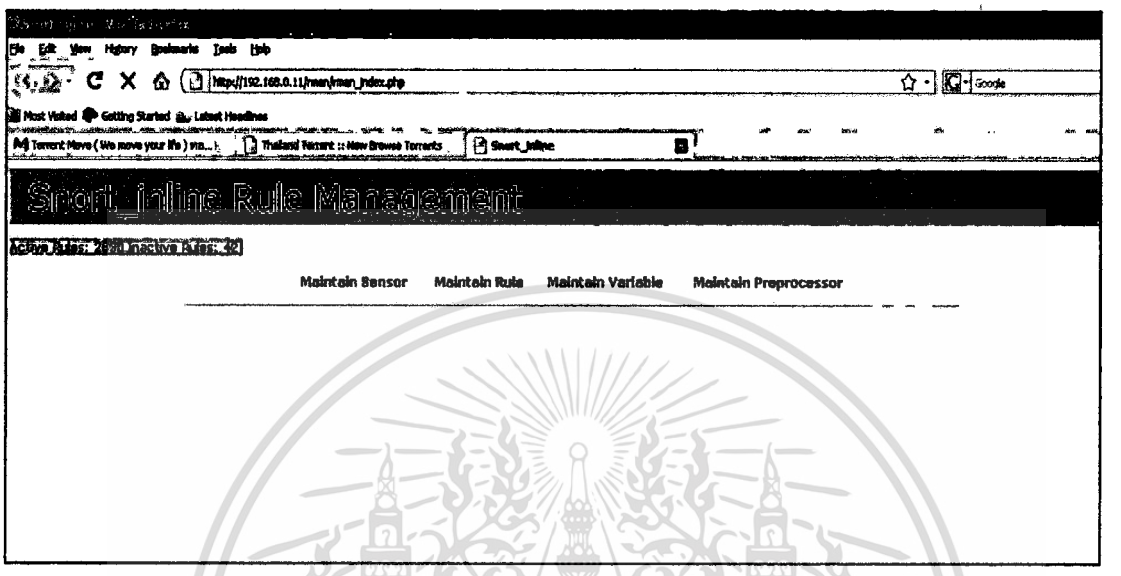



**รูปที่ 3.26** Main Flow Chart Script Loadrules.pl

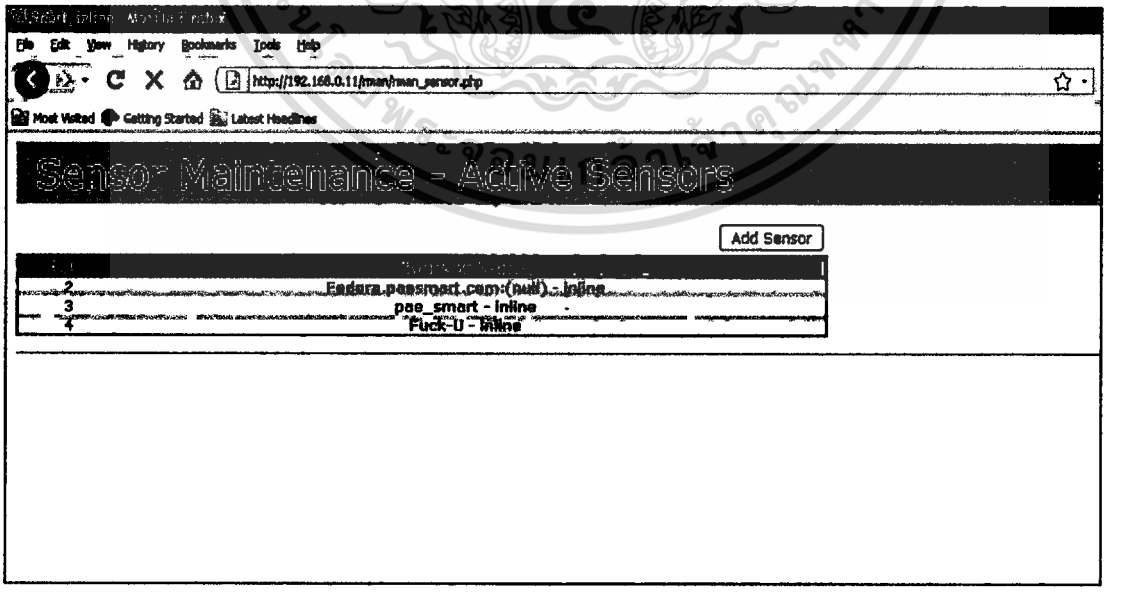
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.5. การออกแบบหน้าจอการทำงานของโครงการพัฒนาระบบงาน

- **หน้าจอหลักของโครงการพัฒนาระบบงาน**

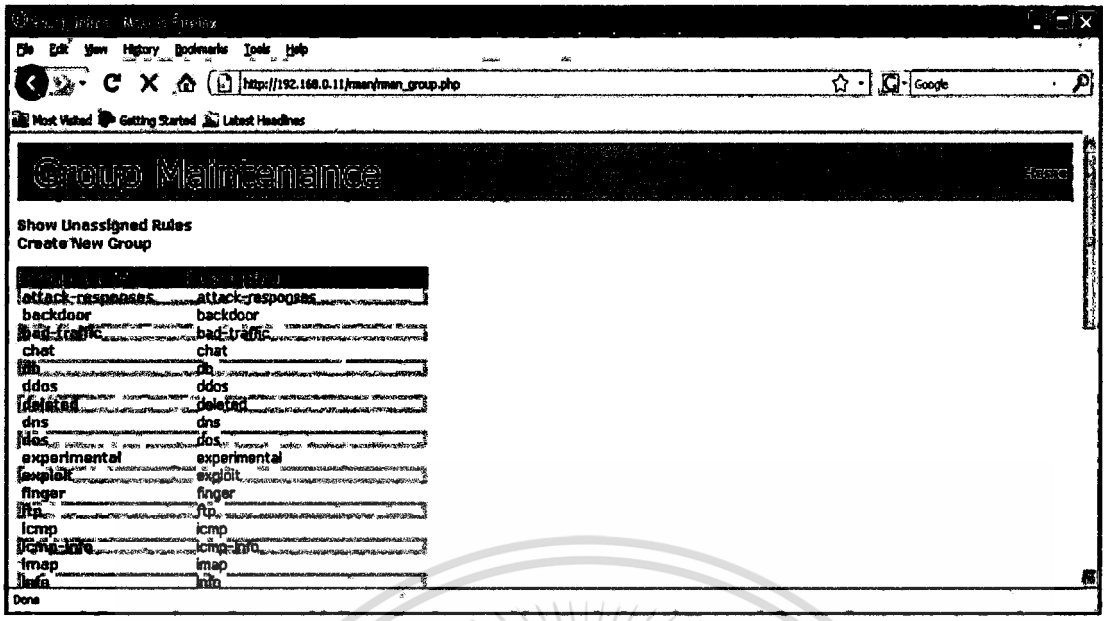



**รูปที่ 3.27.** หน้าแรกของ Web-based IPS Administration Tool

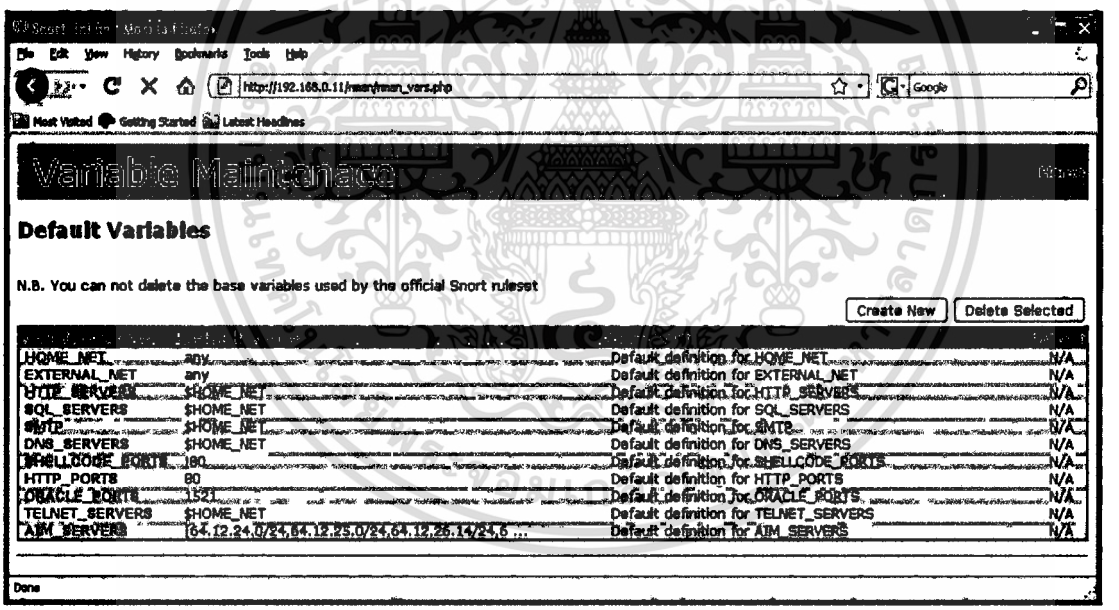



**รูปที่ 3.28.** หน้าของ Maintain Sensor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

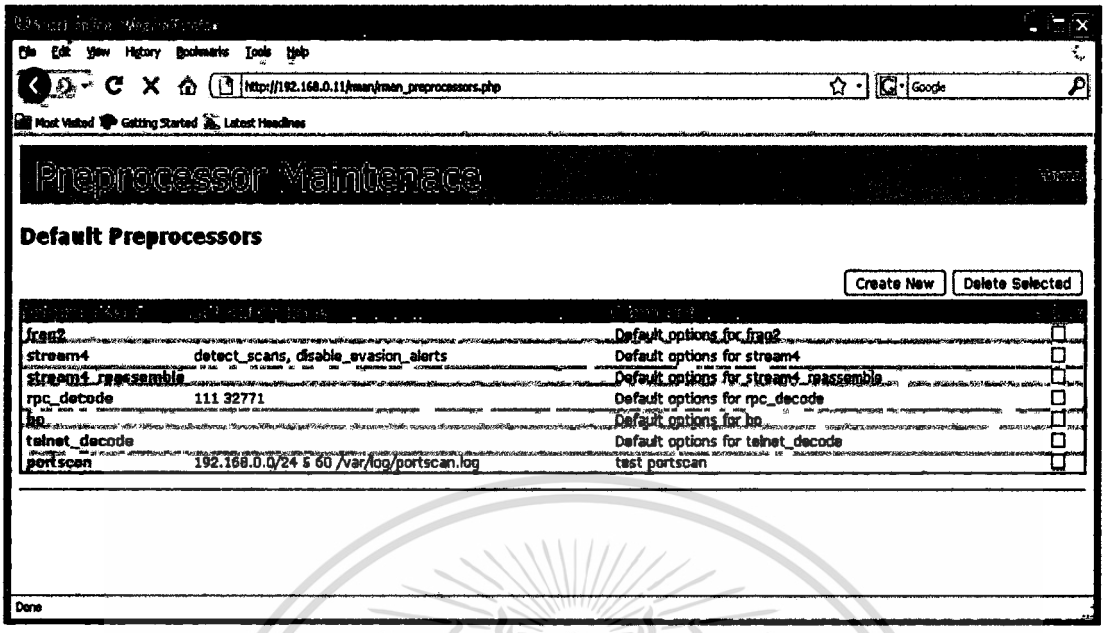



**รูปที่ 3.29. หน้าของ Maintain Rule**




**รูปที่ 3.30. หน้าของ Maintain Variable**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.31. หน้าของ Maintain Preprocessor**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 4

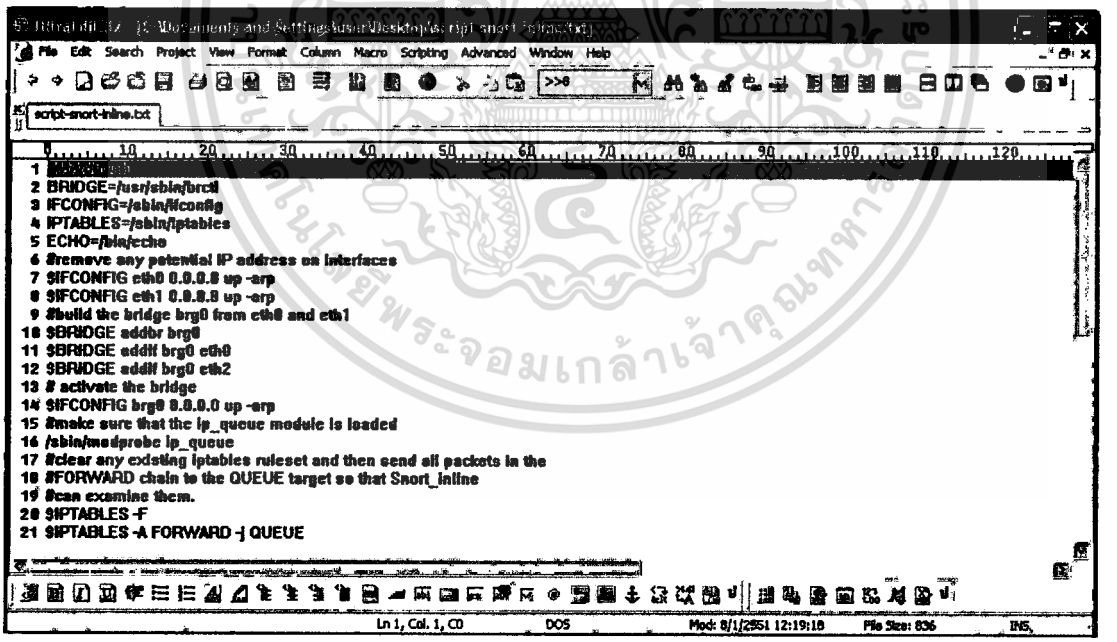
# การพัฒนาระบบ

ระบบที่พัฒนาขึ้นนั้นมีจุดมุ่งหมายที่ทำงานบนระบบปฏิบัติการลินุกซ์ ซึ่งในครั้งนี้ได้ใช้ Fedora Core 4 เป็นระบบปฏิบัติการในการพัฒนาโปรแกรม ซึ่งรันด้วยเคอร์เนลเวอร์ชั่น 2.6 ส่วนภาษาที่ใช้ในการเขียนโปรแกรมนั่นคือ Perl และ PHP ซึ่งระบบนี้จะมีทั้งสคริปต์(perl) และ หน้าเว็บเพจ (php) ที่เป็นส่วนประกอบของโปรแกรม

## 4.1 เครื่องมือที่ใช้ในการพัฒนา

**1) โปรแกรม UltraEdit-32 Version 13.20a**

โปรแกรม UltraEdit เป็นโปรแกรม TextEditor ที่มี color coding ที่ทำให้อ่านง่ายสบายตา และยังแก้ไขไฟล์ได้หลายฟอร์แมตเช่น Pascal , C++ , PHP , Perl Script , ASP เป็นต้น และเป็นโปรแกรมที่ใช้ ram ในการทำงานน้อยมากอีกด้วย




**รูปที่ 4.1 หน้าจอของโปรแกรม UltraEdit**

**2) pcre-6.1**

เป็น library ที่มีกลุ่มของฟังก์ชันเพื่อใช้สำหรับ regular expression pattern matching ใน ภาษา perl

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**3) iptables-1.3.1**

เป็นเหมือนตัวที่ทำ data control ของ แพ็คเกตที่เราจะทำการตรวจสอบ iptables ประกอบไปด้วย built-in chain จำนวน 3 chain ซึ่งไม่สามารถลบได้คือ INPUT, OUTPUT, FORWARD และสำหรับ FORWARD chain

**4) libnet-1.0.2a**

เป็นเครื่องมือในลักษณะของ API คือจะต้องเขียนโปรแกรมเพื่อเรียกใช้อีกทีหนึ่ง โดย Libnet จะทำให้เราสามารถเขียนโปรแกรม เพื่อสร้างแพ็กเกตส่งไปในเครือข่ายได้

**5) snort_inline-2.3.0-RC1**

เป็นตัวที่ใช้สำหรับตรวจสอบและวิเคราะห์ packet ที่ไหลเข้าและออกระหว่างเน็ตเวิคก์

**6) Brctl**

เป็นโปรแกรมทำบริดเสมือนโดยจะต้องใช้ card lan 2 interface ในการทำงาน

**7) MySQL**

MySQL นั้นน่าจะเป็นที่รู้จักกันอยู่แล้วจึงไม่ขอพูดถึงรายละเอียดมากแต่จะพูดถึงส่วนที่ศึกษาเพิ่มขึ้นมาดังนี้

- DBI (Database Independent Interface) คือ เราสามารถเขียนโปรแกรมติดต่อกับฐานข้อมูลของระบบจัดการฐานข้อมูลแต่ละประเภทโดยใช้โปรแกรมหรือ Script เดียวกัน เพียงแต่ต้องบอกว่าต้องการติดต่อกับฐานข้อมูลประเภทใด
- DBD(Database Driver) คือ ไลบรารีของฐานข้อมูลแต่ละประเภท

**8) Apache (Web server)**

**9) PHP**

เป็นภาษาที่เรียกว่า server-side หรือ HTML-embedded scripting language เป็นเครื่องมือที่สำคัญชนิดหนึ่ง ที่ช่วยให้เราสามารถสร้างเอกสารแบบ Dynamic HTML ได้อย่างมีประสิทธิภาพและมีลูกเล่นมากขึ้น

**10) Perl**

Perl ถูกสร้างขึ้นมาบนระบบปฏิบัติการ Unix เพื่อใช้ในงานทั่วไปเนื่องจากตัวของภาษาเป็นภาษาสคริปต์ ซึ่งสามารถนำไปใช้ได้โดยไม่ต้องคอมไพล์ จึงสะดวกในการนำไปใช้บนระบบปฏิบัติการอื่นๆ เมื่อภายหลังเกิดหลักการของ CGI ที่เป็นช่องทางให้ผู้ใช้เว็ปสามารถส่งข้อมูลไปให้ Server ได้ จึงได้นำเอาภาษา Perl มาเขียนเป็น CGI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.2 ขั้นตอนในการพัฒนาระบบ

หลังจากที่ออกแบบระบบแล้ว ก็เริ่มการเขียนโปรแกรม แต่เนื่องจากไม่มีประสบการณ์ในการเขียน Perl Script และ PHP อย่างจริงจังมาก่อน จึงต้องทำการศึกษาการใช้ภาษา Perl และ PHP ได้แก่ ไวยกรณ์ของภาษา ลักษณะการเขียน การเรียกใช้ไลบรารี่ต่างๆ บนระบบ Linux ทำความเข้าใจพื้นฐานของ process บนระบบ Linux และได้ตั้งชื่อโปรแกรมขึ้นว่า IPS-Rule Administration tool

ในการเขียนโปรแกรมนั้นๆได้เขียนฟังก์ชั่นหลักๆดังนี้ (script perl)

**- หาค่า pid ของ snort inline**

ซึ่งการทำงานของมันจะเป็นฟังค์ชั้นเริ่มแรกที่ script ต้องเรียกใช้เลยเพราะจะต้องตรวจสอบก่อนว่ามี snort inline run อยู่หรือเปล่าถ้ามี snort inline run อยู่ก็ return ค่า pid ของ snort inlie นั้นออกมาแต่ถ้าไม่มีการ run snort inline อยู่ก็จะ return ค่า 0 ไปให้ main program

**Note :** การที่เราจะได้ pid ของ snort inline มานั้นเราจะต้อง run snort inline ให้เป็น background mode กล่าวคือเราต้อง run snort inline ในโหมด daemon mode นั่นเองครับ (option –D เวลา run snort inline)

**- Load Rule**

หน้าที่ของฟังก์ชั่นนี้คือจะอ่านไฟล์ที่มีนามสกุล .rule ของสนอร์ททั้งหมดทุกไฟล์และปั้มลงใน database โดยการนำข้อมูลเข้านั้นจะมีกระบวนการแบ่งกฎออกเป็นหมวดๆตามกฎที่มีทั้งหมดรวมถึงการกำหนดหมายเลขของแต่กลุ่มและแต่ละกฎด้วย

**- Update**

หน้าที่หลักคือตรวจสอบ timestamp ใน database กับ virtual file (text file) ว่าตรงกันหรือไม่ซึ่งถ้าตรงกันคือไม่มีอะไร update แต่ถ้าไม่ตรงกันแสดงว่ามีการเปลี่ยนแปลงเกิดขึ้น กล่าวคือในการที่เรากระทำการใดๆกับ sensor ของ snort inline โดยผ่านทางหน้า web page นั้นไม่ว่าจะเป็นการเพิ่ม หรือ ลด rule ก็แล้วแต่เวลาที่เราทำการแก้ไขใดๆนั้นมันก็จะปั้มลงใน database ด้วยครับ ต่อจากนั้น timestamp ที่เป็น virtual file นั้นมันจะเก็บเวลาครั้งสุดท้ายในการแก้ไขและ run script นี้

**- Output Var**

หน้าที่หลักคือทำการสร้าง file โดยมันจะดึงมาจาก databases ซึ่งใน ข้อมูล variable เหล่านี้ที่อยู่ใน database จะได้จากการ run script perl (loadrule) เพื่อที่จะให้ snort inline อ่านได้นั่นเอง (text file ธรรมดา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Web Application (php)**

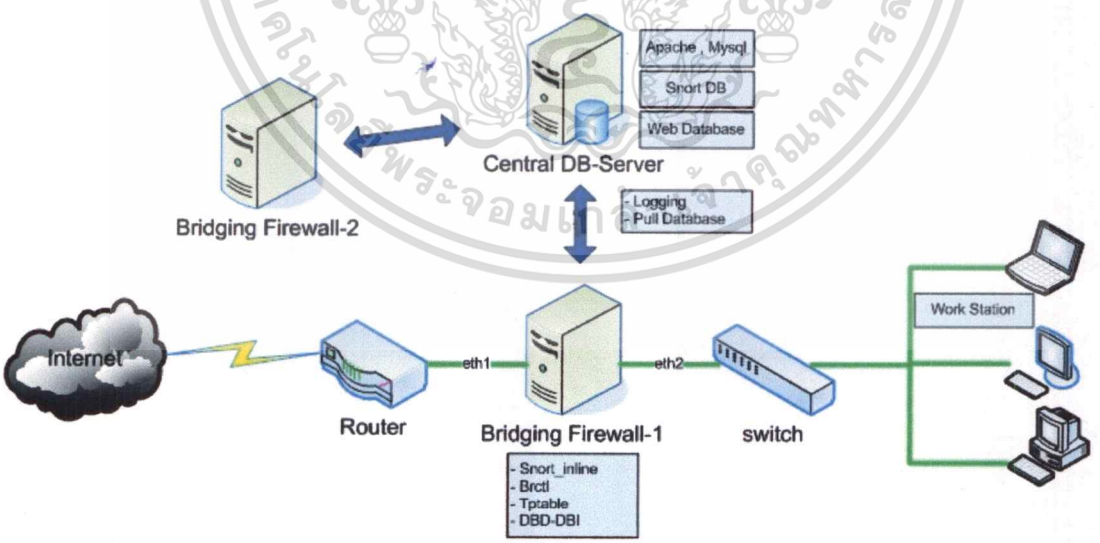
ใช้ภาษา php ในการพัฒนาเขียนติดต่อกับดาต้าเบสเซิรฟ์เวอร์ เพื่อนำเสนอการจัดการ rule ของสนอร์ทอินไลน์ผ่านทางหน้า เว็บแอพพลิเคชั่นโดยได้นำซอฟต์แวร์โอเพ่นซอร์ส มาเป็นต้นแบบกล่าวคือ ACID จะเป็นเว็บที่ใช้ในการ monitor log ของสนอร์ทรวมถึง log anlysis ด้วยโดยมีรูปแบบนำเสนอเป็นกราฟซึ่งช่วยให้การวิเคราะห์ง่ายขึ้น ซึ่งได้นำมาปรับและประยุกต์ใช้ในโปรเจคด้วยบางส่วน

## 4.3 การทดสอบการทำงาน

เนื่องจากการทดสอบนั้นต้องใช้เครื่องอย่างน้อย 4 เครื่อง คือ

- เครื่องที่ทำเป็น Brigde , database server + Web server + sensor (snort inline) 1 เครื่อง
- เครื่องที่ทำเป็น sensor (snort_inline) 1 เครื่อง
- เครื่อง client 1 เครื่อง
- เครื่องที่ทำหน้าเป็น web server **(ทดสอบกฎที่สร้างว่าใช้งานได้จริง)**

ซึ่งที่ต้องใช้เครื่องที่เป็น sensor 2 เครื่องก็เพราะจะแสดงให้เห็นถึงการที่ sensor แต่ละตัวติดต่อเข้ามาเพื่อรับกฎของ snort inline ไปใช้ที่แต่ละ sensor ของแต่ละเน็ทเวิคร์ ซึ่งระบบทั้งหมดรวมถึงเครื่องเซิรฟ์เวอร์ทั้งหมดผมทำใน Vmware version 5.5 ทั้งหมด




**รูปที่ 4.2** เครื่องคอมพิวเตอร์ในการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**คุณสมบัติของเครื่องที่ใช้ทดสอบ (sensor)**

- Ram 94 Mb
- Os Linux Fedora kernel 2.4.9 or more than
- Network interface card 3
- snort_inline,iptable,mysql_client

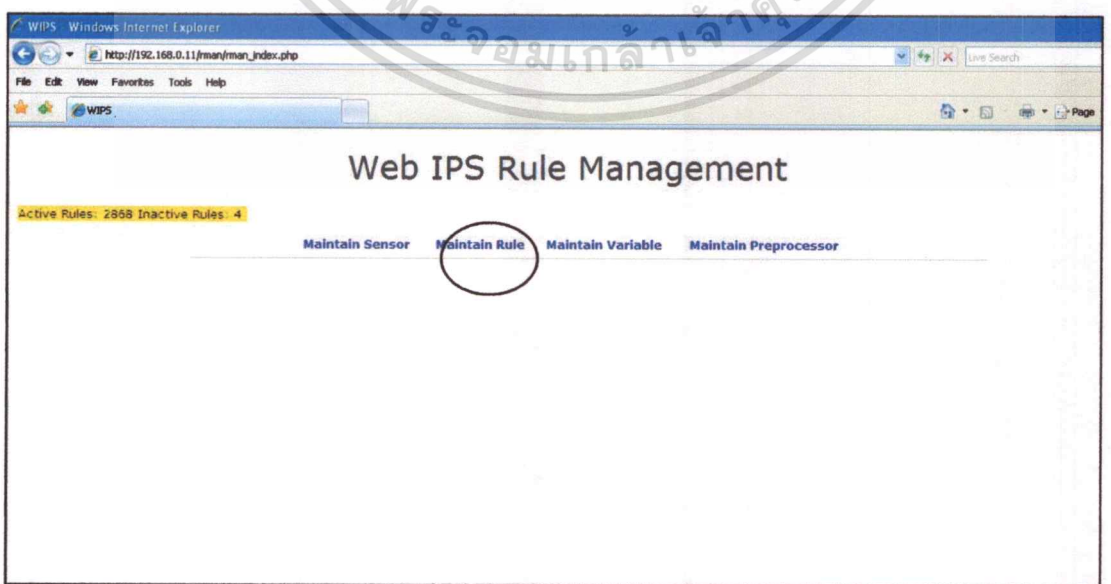
**คุณสมบัติของเครื่องที่ใช้ทดสอบ (Databases server + Web server)**

- Ram 94 Mb
- Os Linux Fedora kernel 2.4.9 or more than
- Network interface card 2
- Apache,Mysql Server,Dbd-Dbi-mysql-module,php

**4.3.1 ขั้นตอนการทดสอบ**

**ทดสอบการใช้งานปกติดังนี้**

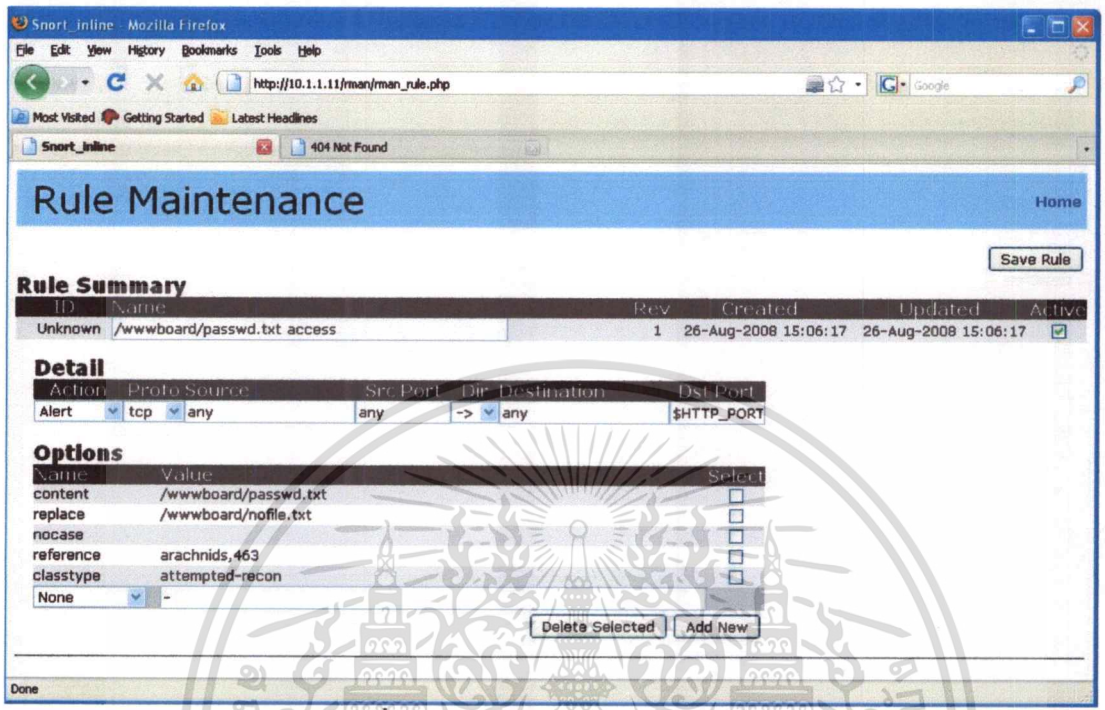
1) เข้าหน้าเว็บเพื่อเลือกกลุ่มของกฎแต่ในที่จะทำการสร้างกฎขึ้นมาเพื่อทดสอบว่าสามารถใช้งานร่วมกับโปรแกมสนอร์ทได้จริงโดย 1. เลือกเมนู Maintain Rule -> 2. เลือกกลุ่มของกฎ -> 3. เลือกกฎไหนก็ได้และกด copy rule -> save rules จากนั้นกฎที่เราสร้างจะอยู่ในกลุ่ม Unassigned เราจะต้องเข้าไปที่กลุ่มนั้นและเลือกว่าจะให้กฎที่เราสร้างใหม่นั้นไปอยู่ที่กลุ่มไหน




**รูปที่ 4.3 หน้าแรกของระบบ**

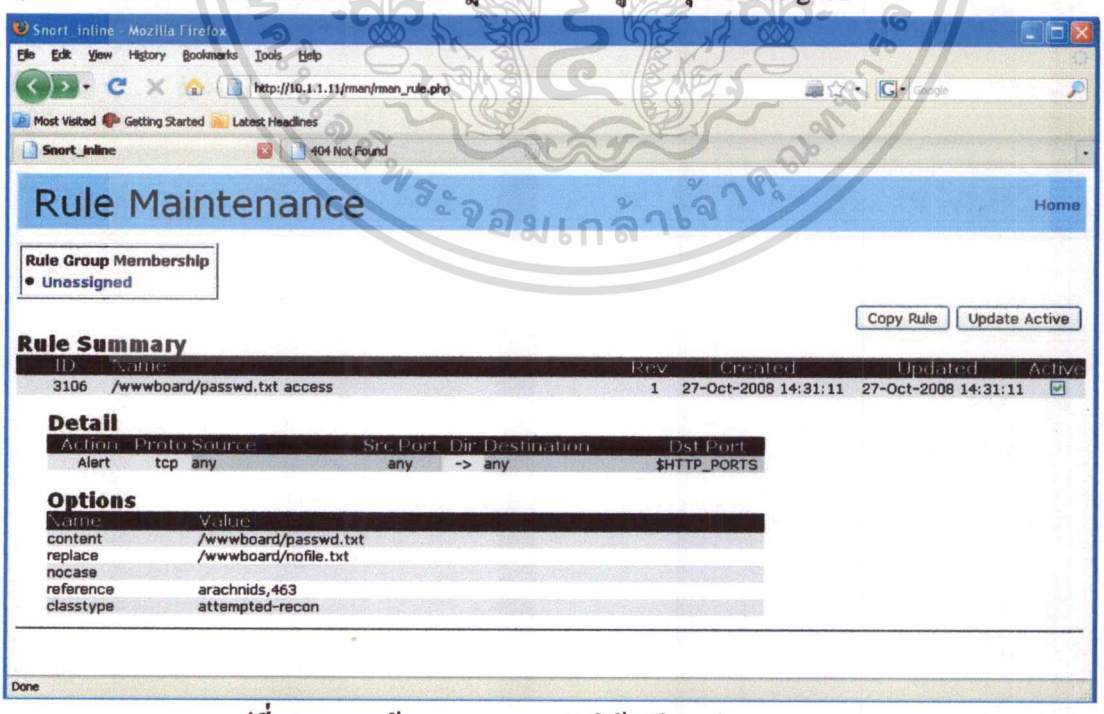
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ตั้งกฎชื่อ /wwwboard/passwd.txt access ใช้สำหรับป้องกันการเข้าถึงในระดับไฟล์ดังรูปที่4.4




**รูปที่ 4.4 หน้าการสร้างกฎของสนอร์ท**

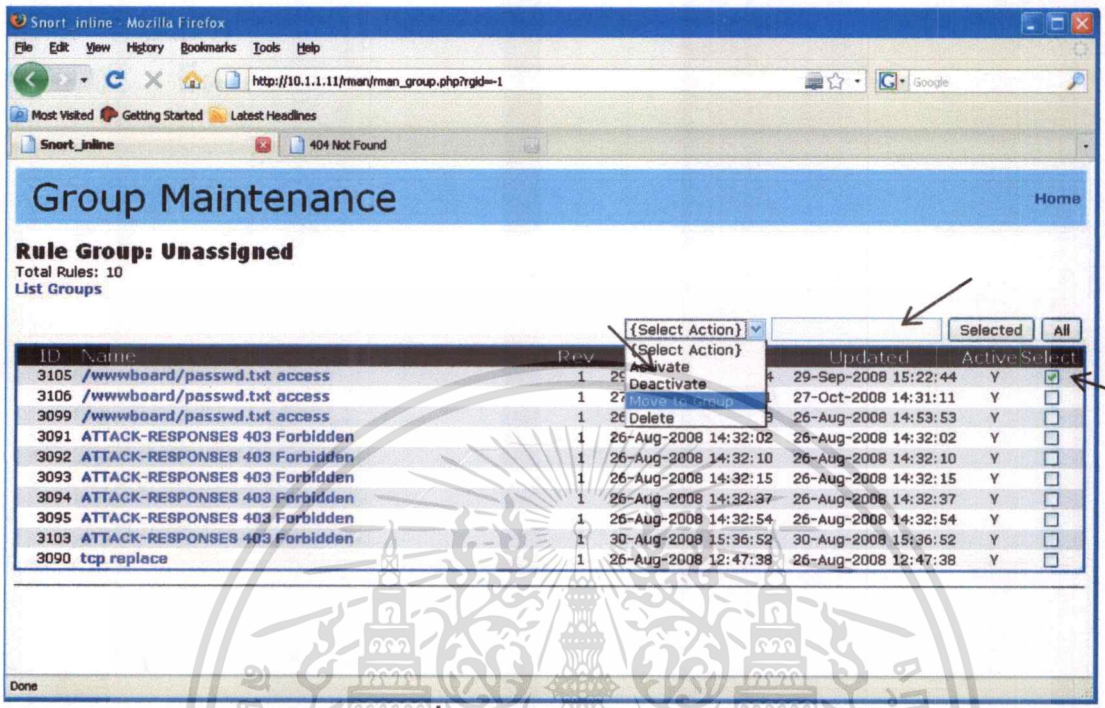
4) หลังจากที่ save rule ที่สร้างใหม่แล้วกฎใหม่จะไปอยู่ในกลุ่ม Unassigned




**รูปที่ 4.5 การสร้างกฎกฎจะถูกจัดให้อยู่ในกลุ่ม Unassigned**

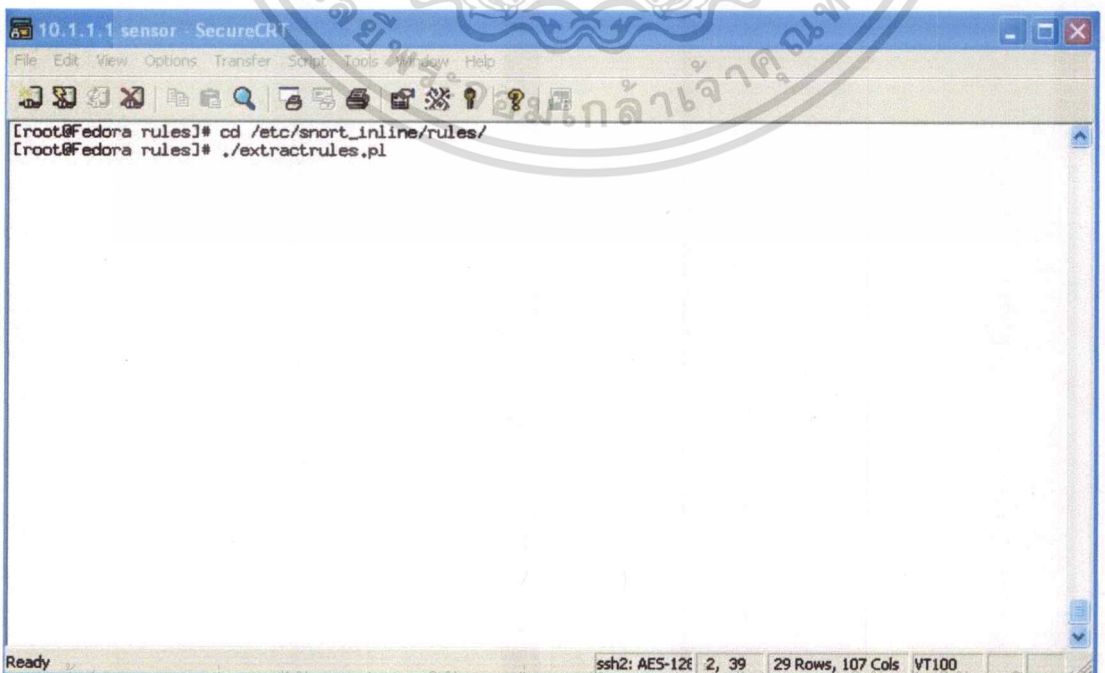
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) เลือกที่ link Unassigned เพื่อเข้าไปในกลุ่มนั้นและเลือกว่าจะให้กฎที่เราสร้างนั้นไปอยู่ในกลุ่มใด(ต้องใส่ชื่อของกลุ่มให้ถูก)




**รูปที่ 4.6** การย้ายกลุ่มของกฎ

6) จากนั้นเข้าไปที่เมนู Maintain sensor -> เลือกกลุ่มของกฎที่ต้องการใช้ -> กด Update Group ต่อไปก็สั่งรันสคริบต์ extractrules.pl



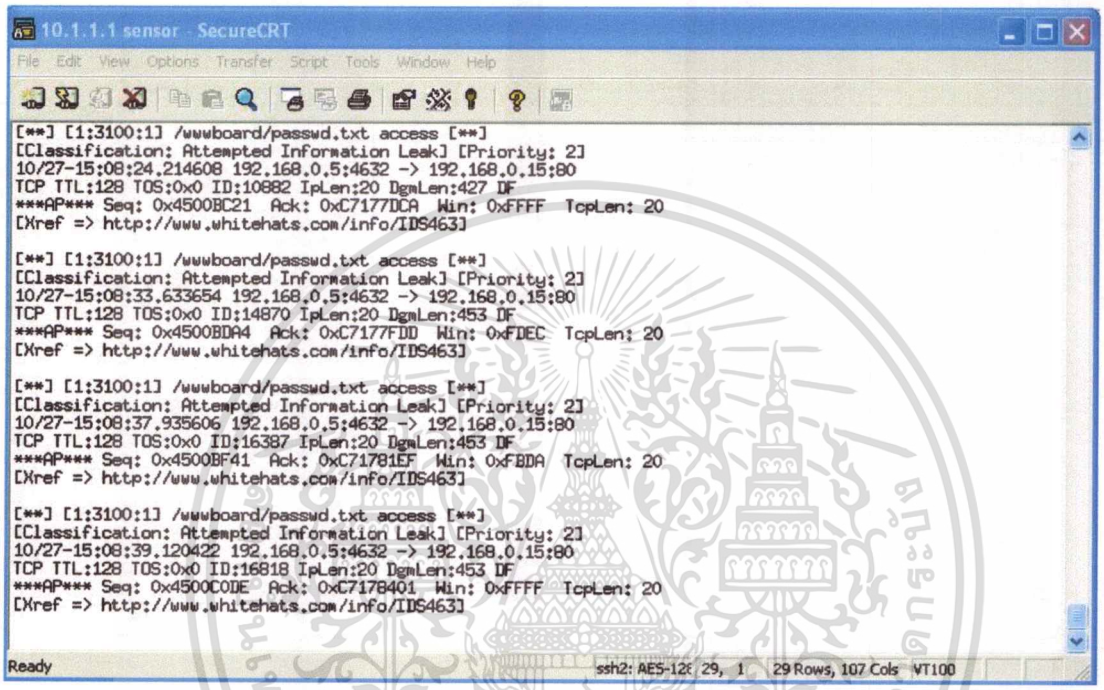

**รูปที่ 4.7** การ run สคริปต์ดึงกฎจากดาต้าเบส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.3.2 ผลการทดสอบ

**โปรแกรมสามารถทำงานได้ตามที่ต้องการคือ**

- สามารถสร้างกฎเพียงครั้งเดียวจากส่วนกลางและนำไปใช้กับแต่ละ sensor ได้
- แต่ละ sensor นั้นสามารถจัดการกฎของตัวเองผ่านทางหน้าเว็บได้จริง
- สามารถสั่งให้ snort inline ทำงานตามกฎที่ได้เลือกไว้ทางหน้าเว็บ




**รูปที่ 4.8** ล็อกไฟล์ของการตรวจจับและป้องกันของไฟล์ passwd.txt บนเครื่อง server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 5
# บทสรุป

## 5.1 ผลจากการพัฒนาระบบ

ผลจากการพัฒนาระบบควบคุมกฎของสนอร์ทจากศูนย์กลางผ่านทางหน้า Web Application นั้น สามารถที่จะทำให้ระบบใช้งานได้ตามความต้องการที่ตั้งไว้ ซึ่งระบบจะประกอบด้วย 2 ส่วน ส่วนแรกจะจัดการกฎ กลุ่มของกฎ และตัวแปลต่างของสนอร์ทของแต่ละ sensor(web) ส่วนที่สองคือ perl script ที่ใช้ในการโหลดกฎของสนอร์ทเข้า database และ perl script ที่ใช้ในการสั่งรวมถึงควบคุมการทำงานของสนอร์ท ซึ่งในส่วนของหน้า web นั้นจะอยู่ที่เครื่องที่ทำหน้าที่เป็น web server แต่ในส่วนของ perl script จะอยู่ที่เครื่องในตำแหน่ง inline หรือเครื่องที่ทำหน้าที่เป็น sensor ของ network นั้นๆเพื่อตรวจสอบข้อมูลที่วิ่งผ่านเหล่านั้น

## 5.2 ประโยชน์ที่ได้รับ

ระบบที่พัฒนาขึ้นสามารถที่จะนำไปใช้ในการป้องกันการบุกรุกได้ โดยเราสามารถกำหนดได้ว่า sensor ไหนใช้กฎไหนได้บ้างรวมถึงเขียนกฎใหม่ขึ้นมาเองก็ได้ซึ่งทำผ่านทางหน้า web ได้เลยทันที ซึ่งรวดเร็วกว่าของเดิมที่เวลาเราจะสั่งให้กฎไหนทำงานหรือจะต้องเปิด text editer ขึ้นมาและค่อยแก้ไขโดยใส่ # ไว้หน้าบรรทัดของกลุ่มของกฎนั้น และของเดิมจะเป็นการจัดการแบบ stand alone ทำที่เครื่องไหนก็เครื่องนั้น ซึ่งระบบนี้ก็จะป้องกันได้ทันทีที่ hit กับกฎที่เลือกไว้และเพื่อให้รบบทำงานได้อย่างมีประสิทธิภาพควรจะปรับแต่งระบบป้องกันการบุกรุกให้เหมาะสมกับการใช้งานของเครือข่ายด้วย

## 5.3 แนวทางการพัฒนาต่อ

เนื่องจากเวลาที่เราได้ติดตั้งสนอร์ทจากที่เราดาวน์โหลดมานั้นกฎก็อาจจะอัพเดพในขณะนั้นแต่เมื่อมีกฎใหม่ออกมาเราก็ต้องโหลดเข้าไปใน database เองกล่าวคืออาจจะให้สนอร์ทอัพเดทกฎเองได้อัตโนมัติเลย หรืออีกอย่างคือระบบนี้สามารถใช้ได้บนระบบปฏิบัติการ Linux เท่านั้นอาจจะนำหลักการแต่บนแพลตฟอร์มอื่นๆ เพื่อเพิ่มความหลากหลายในการทำงานให้มากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บรรณานุกรม


ภูวดล ด่านระหาญ. 2544.**การติดตั้ง Snort ร่วมกับ ACID (+MySQL)**. [Online]. Available: http://www.thaicert.nectec.or.th/paper/ids/snort2.php

ภูวดล ด่านระหาญ. 2544.**Stateful Firewall: IPTABLES**. [Online]. Available: http://www.thaicert.org/paper/firewall/iptables.php

Snort Team. 2006.**Snort User Manual** . [Online].Available: www.snort.org/docs/snort_manual/2.6.1/snort_manual.pdf

Pete Savage. 2005.**Snort Inline Part I** .[Online].Available: http://linuxgazette.net/117/savage.html

Alavoor Vasudevan. 2003.**Compile Kernel** .[Online].Available: http://www.faqs.org/docs/Linux-HOWTO/Kernel-HOWTO.html

Jqme E.Thiel. 2005.**An Introduction to IDS and IPS**.[Online].Available http://www.ece.drexel.edu/telecom/Talks/thiel.pdf

Neil Desai. 2003.**Intrusion Prevention Systems: the Next Step in the Evolution of IDS.** [Online].Available www.securityfocus.com/infocus/1670

Nils Radtke. 2002.**Ethernet Bridge + netfilter Howto**.[Online].Available: http://www.faqs.org/docs/Linux-HOWTO/Ethernet-Bridge-netfilter-HOWTO.html

Michael Rash,Angela Orabaugh ,Becky Pinkard ,Graham Clark ,Jake Babbin and Anne Henmi. 2005.**Intrusion Prevention.And Active Response Deploying Network and Host IPS** , syngress

Carl Endorf ,Eugene Schultz and Jim Mellander. 2003.**Intrusion Detection & Prevention** , McGraw-Hill/Osborne



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก ก
# การติดตั้งระบบ

**ความต้องการของระบบ**

สำหรับเครื่องที่สามารถติดตั้งได้นั้นมีรายละเอียดดังนี้

- ระบบปฏิบัติการลินุกซ์ หรือ FreeBSD (Fedora Core 4 or high)
- MySql 3.23.40 or later (on server)
- MySql Client + Shared Libraries (on sensor)
- Snort Inline + IPtables
- PHP 4.1.x
- Perl 5.6.x
- Perl DBD/Mysql DBI interface ทั้ง central server and sensor

**การติดตั้ง**

**- โปรแกรม Mysql + DBI**

rpm –ivh Mysql-##-###.rpm

rpm –ivh perl-DBI-1.37-0.src.rpm

**- โปรแกรม Apache และ PHP**

ทำการแตกไฟล์ httpd-2.0.53.tar.gz โดยใช้คำสั่ง tar xvfz httpd-2.0.53.tar.gz

1.1 ./configure –prefix=/usr/local/apache \
--enable-so \
--enable-cgi \
--enable-info \
--enable-rewrite \
--enable-speling \
--enable-usertrack \
--enable-deflate \
--enable-ssl \
--enable-mime-magic

1.2 make

1.3 make install

1.4 แตกไฟล์ PHP โดยใช้คำสั่ง tar xvfz php-4.3.10.tar.gz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

./configure \

--with-apxs2=/usr/local/apache/bin/apxs \

--with-mysql \

--prefix=/usr/local/apache/php \

--with-config-file-path=/usr/local/apache/php \

--enable-force-cgi-redirect \

--disable-cgi \

--with-zlib \

--with-gettext \

--with-gd

1.5 make

1.6 make install

1.7 cp -p php.ini-recommended /usr/local/apache/php/php.ini

1.8 เพิ่มบรรทัดนี้ในไฟล์ httpd.conf

LoadModule php4_module modules/libphp4.so

1.9 เพิ่ม index.php ในท้ายบรรทัดดังนี้

DirectoryIndex index.html index.php

1.10 ลบเครื่องหมาย # ในไฟล์httpd.conf บรรทัดดังนี้

AddHandler php5-script php

AddType text/html php

AddType application/x-httpd-php-source phps

<Files *.php>

SetOutputFilter PHP

SetInputFilter PHP

</Files>

1.11 /usr/local/apache/bin/apachectl start

**- IPtables**

- tar xjvf iptables-1.3.1.tar.tar

- cd /usr/src/iptables-1.3.1

- make install-devel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**- Libnet**

- tar xzvf libnet-1.0.2a.tar.gz
- cd /usr/src/Libnet-1.0.2a
- ./configure
- make
- make install

**- Pcre**

- tar xzvf pcre-6.1.tar.gz
- cd /usr/src/pcre-6.1
- ./configure
- make
- make install

**- Snort_inline**

- tar xzvf snort_inline-2.3.0-RC1.tar.gz
- cd snort_inline-2.3.0-RC1
- ./configure --with-mysql
- make
- make install
- cp /usr/src/snort_inline-2.3.0-RC1/etc/classification.config /usr/src/snort_inline-2.3.0-RC1/rules/
- cp /usr/src/snort_inline-2.3.0-RC1/etc/reference.config /usr/src/snort_inline-2.3.0-RC1/rules/
- mkdir /etc/snort_inline
- cp /usr/src/snort_inline-2.3.0-RC1/etc/* /etc/snort_inline/

cp /usr/src/snort_inline-2.3.0-RC1/rules /etc/snort_inline/ -R

**- Brctl**

- tar xvzf bridge-utils-0.9.5.tar.gz
- cd bridge-utils-0.9.5
- make
- cp -vi brctl/brctl /sbin/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากที่ได้ติดตั้งโปรแกรมที่จะเป็นทั้งหมดแล้วเพื่อให้ระบบสามารถทำงานให้ได้ตามความต้องการมีขั้นตอนการปรับแต่งดังนี้

**1. Databases Server**

**2. Population the Databases**

**3. Sensor Configuration**

**4. Web Front End**

**Databases Server**

ก่อนจะติดตั้งบนเครื่อง databases server จะต้องสร้างฐานข้องของ snort ก่อนดังนี้

```
# mysql -u root
mysql> set password for 'root'@'localhost'=password('mypassword');
mysql> create database snort;
mysql> grant CREATE, INSERT, SELECT, DELETE, UPDATE on snort.* to snort;
mysql> grant CREATE, INSERT, SELECT, DELETE, UPDATE on snort.* to
snort@localhost;
mysql> connect mysql;
mysql>set password for 'snort'@'localhost'='password('mypassword');
mysql>set password for 'snort'@'%'='password('mypassword');
mysql> flush privileges;
mysql>exit
/usr/src/snort_inline-2.3.0-RC1/schemas/create_mysql.gz | mysql -p snort
```

ต่อไปให้ใช้ file mysql.dbschema เพื่อสร้าง table เพิ่มเข้าไปในฐานข้อมูลของ snort ดังนี้

```
[pae_smart@snortrm]$ mysql –p snort < mysql.dbschema
Enter password : XXXXXXX
```

หลังจากนั้นควรจะได้ table เพิ่ม 7 table ใน databases ซึ่งมี prefixed ด้วย ips_ มาถึงตอนนี้จะต้องสร้าง user ชื่อ ips_www ที่มีสิทธิที่จะ Select Insert Update และ Delete บน 7 table ที่ได้สร้างขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Population the Databases**

ดาวน์โหลด กฎของสนอร์ทเวอร์ชั่นล่าสุดจาก website ก่อนและเก็บลงโฟล์เดอร์ที่จะปั้มลงใน database ซึ่งตรงนี้จะต้องใช้ perl script ที่ชื่อ loadrules.pl และในการ run ต้องตามด้วยที่อยู่ของกฎด้วย ดังนี้

[pae_smart@snortrm]$ ./loadrules.pl **/etc/snort_inline/rules/**

Ruleset: attack-responses

Rule 1292: new, ATTACK RESPONSES http dir listing

Rule 498: new, ATTACK RESPONSES id check returned root

Rule 494: new, ATTACK RESPONSES command completed

Rule 495: new, ATTACK RESPONSES command error

Rule 497: new, ATTACK RESPONSES file copied ok Ruleset: backdoor Rule 103: new, BACKDOOR subseven 22 Rule 104: new, BACKDOOR - Dagger_1.4.0_client_connect Rule 105: new, BACKDOOR - Dagger_1.4.0

..... ตัดเนื้อหาบางส่วนออกเพื่อความกระชับ

Script จะแสกนไฟล์ทั้งหมดในโฟร์เดอร์และโหลดกฎทั้งหมดตามชื่อกลุ่มของกฎต่างๆ และเมื่อดาวน์โหลดกฎมาใหม่จะต้อง run script ใหม่ทุกครั้งเพื่อเพิ่มลงใน databases

**Sensor Configuration**

ตามปกติเมื่อเรา run snort_inline มันจะ register เข้าไปที่ table sensor ของ snort inline ก่อนในครั้งแรกแต่ไม่ในที่นี้ไม่ต้อง run snort ก่อนเลย แต่มีสิ่งที่ต้องพิจารณาร่วมดังนี้

- ระบุตำแหน่งไดเรกทอรีที่เก็บกฎใน snort_inline.conf

  var RULE_PATH /etc/snort_inline/rules/

- คัดลอกไฟล์ db.config และ db.timestamp ลงในโฟร์เดอร์กฎของสนอร์ท (/etc/snort_inline/rules/)
- แก้ไขข้อมูลเกี่ยวกับ sensor และ database ในไฟล์ db.config
- แก้ไข snort_inline.conf ตามต้องการ
- Disable rule ทั้งหมดรวมถึง variable ต่างๆ ด้วยในไฟล์ snort_inline.conf (var HOME_NET etc.)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เพิ่มบรรทัดที่เป็นตัวหนาในไฟล์ snort_inline.conf ig เหมือนกับที่แสดงด้านล่างเมื่อเรา disable variable ต่างๆดังนี้

```
# or you can specify the variable to be any IP address
# like this:
include db.vars
```

- เพิ่มบรรทัดนี้ในส่วนของ database logging ดังนี้

```
# database: log to a variety of databases
# -----------------------------------------
include db.config
```

- disable rule ทั้งหมดใน snort_inline.conf ig และเพิ่ม db.rules เข้าไปในส่วนของกฎ

```
#==================================================
# Include all relevant rulesets here
#
include db.rules
# all the other include.rules commented out
```

- สร้างไฟล์เปล่าที่ชื่อว่า db.rules , db.vars

```
[pae_smart@snortrm]$ touch db.rules
```

**Web Front End**

- คัดลอกไฟล์ทั้งหมด (.php, .inc, .html ,css) ลงในไดเรกทอรี่ของ web server
- แก้ไขในส่วนของ ips_common.inc ตรงตัวแปร $dbuser , $dbpass
- ลองเข้า web ดู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก ข

# การใช้งานระบบ

## การใช้งาน

เงื่อนไขในการใช้งานได้จะต้องรันโปรแกรมที่เกี่ยวข้องโดยได้เขียนเป็น shell script ดังนี้

```
#!/bin/bash
ifconfig eth0 0.0.0.0 up -arp
ifconfig eth1 0.0.0.0 up -arp
brctl addbr brg0
brctl addif brg0 eth0
brctl addif brg0 eth2
# activate the bridge
ifconfig brg0 0.0.0.0 up -arp
#make sure that the ip_queue module is loaded
modprobe ip_queue
#clear any existing iptables ruleset and then send all packets in the
#FORWARD chain to the QUEUE target so that Snort_inline
#can examine them.
$IPTABLES -F
$IPTABLES -A FORWARD -j QUEUE
# turn forwarding OFF!!!!
$ECHO 0 > /proc/sys/net/ipv4/ip_forward
```

**รูปที่ 1** snort_inline.sh

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**วิธีการใช้งาน**

**พิมพ์คำสั่งดังนี้**

```
[pae_smart@snortrm]$ ./snort_inline
```

**หลังจากนั้นให้รัน perl script extractrule.pl ดังนี้**

```
[pae_smart@snortrm]$ ./extractrule.pl
```

**หลังจากที่รัน perl script extractrule.pl แล้วให้ลองตรวจสอบดูว่ามีโพรเซสที่ snort_inline อยู่หรือไม่ดังนี้(ต้องมี)**

```
[pae_smart@snortrm]$ ps –ef | more
```

**ซึ่งตรงนี้มีข้อควรระวังว่าให้ตรวจสอบที่อยู่ของ snort_inline และที่อยู่ของกฎของสนอร์ทให้ดีด้วยใน script extractrule.pl เพราะไม่อย่างนั้นแล้วจะรันไม่สำเร็จ**

**ผู้ใช้สามารถตรวจสอบ log file ของโปรแกรม เพื่อดูว่าเกิดข้อผิดพลาดอะไรหรือเปล่าได้ที่ /etc/snort_inline/rule/ruletest.log รวมถึง log file ต่างของตัวสนอร์ทเองได้ที่ /var/log/snort_inline**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ประวัติผู้เขียน

| | |
|---|---|
| **ชื่อ – นามสกุล** | **นายธเนศ ไพรินทราภา** |
| **วัน เดือน ปีเกิด** | **27 พฤษภาคม 2524** |
| **ประวัติการศึกษา** | **สำเร็จการศึกษาปริญญาตรี วิทยาศาสตร์บัณฑิต** |
| | **สาขาวิชาวิทยาการคอมพิวเตอร์** |
| | **มหาวิทยาลัยหอการค้าไทย** |
| **ที่ทำงานปัจจุบัน** | System Advisor Group |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้