สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การศึกษาพฤติกรรมของไวรัสบนเครือข่าย

THE MONITORING OF VIRUS'S BEHAVIOR ON NETWORK SYSTEM

ณีรนุช มหรัตนวิโรจน์
ทรงธรรม เจริญทอง
กฤษณะพงศ์ จารุวิจิตรรัตนา

วพ.
ว4451ก
2550
เลขหมู่.....
เลขทะเบียน.....82767
วัน,เดือน,ปี.....23 ต.ค. 2551

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2550

b.....11950644.....
i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

THE MONITORING OF VIRUS'S BEHAVIOR ON NETWORK SYSTEM

NEERANUCH MAHARATANAVIROJ

SONGTHAM CHAREONTONG

KRISSANAPONG JARUVIJITRATTANA

A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR DEGREE OF BACHELOR OF SCIENCE

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE

FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2007

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	การศึกษาพฤติกรรมของไวรัสบนเครือข่าย THE MONITORING OF VIRUS'S BEHAVIOR ON NETWORK SYSTEM	
ชื่อนักศึกษา	นางสาวณีรนุช มหรัตนวิโรจน์	47050327
	นายทรงธรรม เจริญทอง	47050330
	นายกฤษณะพงศ์ จารุวิจิตรรัตนา	47050348
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์	
สาขาวิชา	วิทยาการคอมพิวเตอร์	
อาจารย์ที่ปรึกษา	อาจารย์ศังกรศรัณย์ ล่องชูผล	

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์ ปีการศึกษา 2550

คณะกรรมการสอบ	ลายมือชื่อ
รองศาสตราจารย์ไพรบูลย์ พันธรักษ์พงษ์ ประธานกรรมการ	
อาจารย์วิสันต์ ตั้งวงษ์เจริญ กรรมการ	
อาจารย์ศังกรศรัณย์ ล่องชูผล กรรมการและอาจารย์ที่ปรึกษา	

(รองศาสตราจารย์ไพรบูลย์ พันธรักษ์พงษ์)
หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แด่ คุณพ่อ คุณแม่ อาจารย์และเพื่อนๆ ผู้เป็นที่รัก

กฤษณะพงศ์ ทรงธรรม ณีรนุช

ขออุทิศให้ทุกๆคนที่เป็นกำลังใจ และช่วยเหลือมาโดยตลอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	การศึกษาพฤติกรรมของไวรัสบนเครือข่าย	
ชื่อนักศึกษา	นางสาวณีรนุช มหรัตนวิโรจน์	47050327
	นายทรงธรรม เจริญทอง	47050330
	นายกฤษณะพงศ์ จารุวิจิตรรัตนา	47050348
ปริญญา	วิทยาศาสตรบัณฑิต	
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์	
สาขาวิชา	วิทยาการคอมพิวเตอร์	
ปีการศึกษา	2550	
อาจารย์ที่ปรึกษา	อาจารย์ศังกรศรัณย์ ล่องชูผล	

บทคัดย่อ

เนื่องจากในปัจจุบันนี้มีการพัฒนาของโปรแกรมประยุกต์เป็นจำนวนมากรวมไปถึงการพัฒนาของหนอนและไวรัสต่างๆ ซึ่งหนอนและไวรัสจะก่อให้เกิดความเสียหายไม่มากก็น้อยแก่เครื่องคอมพิวเตอร์ของผู้ใช้งาน กลุ่มผู้ทดลองได้ตระหนักถึงอันตรายของหนอนบนเครือข่าย จึงได้ทำการศึกษาแพ็คเกตต่างๆที่อยู่บนเครือข่าย โดยจะใช้ตัวโปรแกรมสนอร์ทในการนำมาช่วยตรวจสอบแพ็คเกต ภายในโปรแกรมจะมีส่วนของกฎที่ไว้ใช้สำหรับการตรวจจับแพ็คเกต กลุ่มผู้ทดลองได้ทำการทดลองเกี่ยวกับการดูพฤติกรรมต่างๆของหนอนบนเครือข่าย ว่าหนอนนั้นทำงานเป็นเช่นไร และเมื่อรู้การทำงานของมัน ก็สามารถทำให้ผู้ใช้สามารถป้องกันการโจมตีของหนอนชนิดนั้นได้ โดยกลุ่มผู้ทดลองใช้โปรแกรมสนอร์ทในการดักจับแพ็คเกตที่มีลักษณะคล้ายว่าจะเป็นหนอน แล้วเมื่อเจอก็จะทำการแจ้งเตือนขึ้นมาทันที และผู้ทดลองยังคิดค้นส่วนของการเพิ่มกฎในโปรแกรมสนอร์ทที่ไว้ใช้ดักจับแพ็คเกตที่ผ่านเข้ามาบนเครือข่ายเพื่อให้ผู้ใช้งานมีความสะดวกสบายมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	THE MONITORING OF VIRUS'S BEHAVIOR ON NETWORK SYSTEM	
Students	Ms.Neeranuch Maharatanaviroj	47050327
	Mr.SongTham Chareontong	47050330
	Mr.Krissanapong Jaruvijitratana	47050348
Degree	Bachelor of Science	
Department	Mathematics and Computer Science, Faculty of Science	
Programme	Computer Science	
Academic Year	2007	
Advisor	Mr.Sungkornsarun Longchupole	

ABSTRACT

Many worms and viruses are spreading all over networks today. Some of them may cause serious damage to network and computer systems. The purpose of this special problem is to propose a solution to the problem by developing a program that can alarm when worm/virus like behavior packets are transmitted to/from a network. The information that the program needs is the differences between normal and suspected packets. In order to gain such information, all packets that travel along a network are needed to be captured and examined to find out the patterns of the infected packets. The rules for identifying such packets are then developed and used by the program to warn users about the possible intrusion of worms/viruses. Snort, an open source network prevention/detection system is used as a tool for packet capturing. The rules developed in this research are also transformed into the form that can be used directly by snort. This allows network administrators to have more tools to filter the malicious packets out of their network systems.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ในการจัดทำปัญหาพิเศษนี้ คณะผู้จัดทำขอขอบพระคุณอาจารย์ที่ปรึกษาที่ได้เสียสละเวลาให้คำแนะนำ คำชี้แจง ความรู้และความเอาใจใส่จาก อาจารย์ศังกรศรัณย์ ล่องชูผลในการปรับปรุงปัญหาพิเศษนี้

ขอขอบพระคุณ คณะกรรมการการสอบปัญหาพิเศษที่กรุณาให้คำแนะนำตลอดจนข้อชี้แนะต่างๆ ทำให้ปัญหาพิเศษฉบับนี้สำเร็จลงได้

ขอขอบพระคุณบิดา-มารดา ที่ได้สนับสนุนด้านการศึกษาในระดับอุดมศึกษา อีกทั้งคอยดูแลและเป็นห่วงในเรื่องต่างๆเป็นอย่างดี

ขอขอบคุณเพื่อนๆที่เกี่ยวข้อง ทั้งที่ได้ให้ความช่วยเหลือในการให้ข้อมูล พื้นที่ที่ให้ทำปัญหาพิเศษ ข้อเสนอแนะในการทำปัญหาพิเศษฉบับนี้

ขอขอบคุณสำหรับคุณงามความดีและประโยชน์อันใดที่เกิดขึ้นจากการทำปัญหาพิเศษนี้ คณะผู้จัดทำขอมอบให้กับบิดา-มารดา อาจารย์ทุกท่านซึ่งเป็นที่เคารพยิ่ง ตลอดจนญาติพี่น้องและเพื่อนๆ ทุกคน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่ **หน้า**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่ **หน้า**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากในปัจจุบันนี้ยุคสมัยมีความเปลี่ยนแปลงไปจากเดิม ทั้งในเรื่องของการเดินทาง การติดต่อสื่อสาร หรือการส่งข้อมูลที่มีการพัฒนาให้มีความรวดเร็วในการส่งมากขึ้นโดยการใช้ระบบเครือข่ายเข้ามาช่วย เมื่อมีการใช้งานส่วนนี้มากขึ้น ระบบการรักษาความปลอดภัยจึงจำเป็นอย่างยิ่งเพราะอาจมีผู้ลักลอบเข้ามาในระบบโดยที่ไม่ได้รับอนุญาต โดยอาจจะทำพฤติกรรมที่ไม่ดี ไม่ว่าจะเป็นการทำลายข้อมูล การลอบนำข้อมูลที่สำคัญไปใช้ หรือใช้เป็นฐานเพื่อการโจมตีเครือข่ายอื่นๆ โดยอาจใช้ไวรัสบนเครือข่ายทำให้ข้อมูลเสียหายและแพร่กระจายไปสู่เครื่องอื่นในเครือข่ายเดียวกัน โดยผู้ใช้งานบางท่านยังไม่เห็นถึงความสำคัญในส่วนประกอบนี้ ทางคณะผู้จัดทำได้เห็นถึงปัญหาในปัจจุบันนี้ จึงเสนอเพื่อทำการวิจัยและศึกษาวิธีการป้องกันแก้ไขของปัญหาที่เกิดขึ้น

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1.2.1 เพื่อพัฒนาและให้ความสำคัญในความปลอดภัยของระบบเครือข่าย

1.2.2 เพื่อให้ระบบเครือข่ายที่ใช้งานมีความปลอดภัยมากยิ่งขึ้น

1.2.3 เพื่อศึกษาการใช้งานและพัฒนาระบบรักษาความปลอดภัยให้เครือข่ายอย่างมีประสิทธิภาพ

1.2.4 เพื่อให้มีการตรวจสอบความปลอดภัยให้ใช้งานได้ดียิ่งขึ้น

1.2.5 ทราบถึงปัญหาและขั้นตอนการทำงานของระบบรักษาความปลอดภัย

1.2.6 เพื่อให้สามารถรับรู้โครงสร้างของไวรัสแต่ละประเภทได้

1.2.7 เพื่อศึกษาการทำงานของไวรัสแต่ละประเภท

1.3 ขอบเขตของปัญหาพิเศษ

ปัญหาพิเศษนี้ จะทำการตรวจสอบข้อมูลที่มีพิษภัยโดยแอบแฝงมากับไฟล์ต่างๆ หรือที่เราเรียกว่า “ไวรัส” เริ่มแรกเราจะทำการทดลองในระบบปิดก่อน โดยจะเริ่มจากเครื่อง 2 เครื่องคือ เครื่องผู้ที่โจมตีกับเครื่องผู้ถูกโจมตี และจะทำการทดลองส่งไวรัสเข้ามาในตัวเครื่องของผู้ที่ถูกโจมตี เราจะมีตัวตรวจสอบข้อมูลที่ส่งเข้ามาทางเครือข่าย โดยการตรวจสอบจะตรวจสอบกับข้อมูลที่เรียกว่า “แพ็คเกต” ถ้าหากแพ็คเกตที่ส่งเข้ามามีลักษณะคล้ายสิ่งที่เป็นข้อมูลไม่หวังดี เราจะส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณเตือนให้ผู้ใช้รู้ว่ามีข้อมูลที่อาจจะเป็นอันตรายต่อคอมพิวเตอร์เข้ามา หลังจากนั้นจะเก็บลักษณะข้อมูลนั้นมาทำการดูโครงสร้างภายในของมันและ การทำงานของมันว่าเป็นอย่างไร

1.4 ขั้นตอนในการดำเนินงาน

1.4.1 ศึกษาชนิดของไวรัส และการทำงานของไวรัสชนิดต่างๆ

1.4.2 ศึกษาการทำงานของโปรแกรมดักจับไวรัส

1.4.3 วิเคราะห์การทำงานและปัญหาของระบบ

1.4.4 ออกแบบการตรวจสอบระบบในชุดเครือข่าย

1.4.5 เขียนกฎและทดสอบการทำงาน

1.4.6 ทำการทดลองปัญหา

1.4.7 เข้าใจปัญหา ศึกษาสิ่งที่ต้องแก้ไข

1.4.8 ประเมินผลงาน

1.4.9 ทำรายงาน

1.4.10 ส่งผลงานและนำเสนอผลงาน

1.5 ประโยชน์ที่คาดว่าจะได้รับ

เนื่องจากกลุ่มวิจัยได้นำโปรแกรมตรวจสอบเครือข่ายมาใช้ทดสอบไวรัสประเภทหนอนที่ทำการบุกรุกต่อระบบของเครื่องผู้ใช้งาน และยังสามารถตรวจสอบถึงแพ็คเกตที่เข้าออกเครื่องผู้ใช้งาน ดังนั้นจึงทำให้ผู้ที่อ่านสามารถใช้งานโปรแกรมตรวจสอบเครือข่ายได้อย่างมีประสิทธิภาพและยังเพิ่มความสะดวกในการดูสิ่งที่เข้ามาภายในเครือข่ายอีกด้วย ทั้งยังเพิ่มความรู้ความเข้าใจให้กับผู้อ่านในเรื่องของโปรแกรมที่โจมตีเครื่องของผู้ใช้งาน โครงสร้างและการทำงานของหนอน ทำให้สามารถป้องกันการบุกรุกและการแพร่กระจายของหนอนที่จะทำการโจมตีเครื่องผู้ใช้งานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

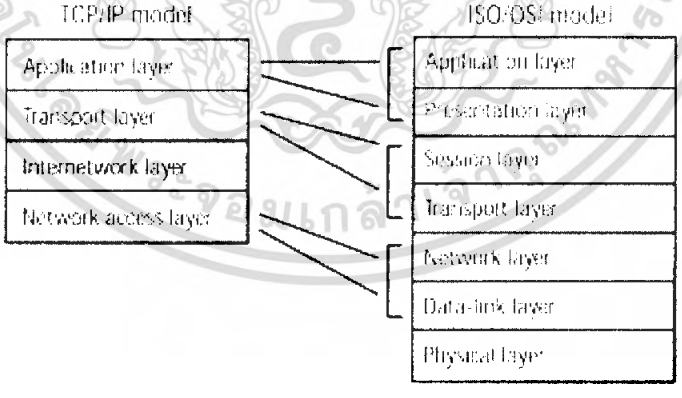
2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 ความรู้เบื้องต้นเกี่ยวกับระบบเครือข่ายบนอินเตอร์เน็ต

การทำงานของระบบเครือข่ายคอมพิวเตอร์จะแบ่งออกเป็นแต่ละชั้นหรือที่เราเรียกว่าเลเยอร์ (Layer) โดยแต่ละเลเยอร์จะทำหน้าที่ไม่เหมือนกันและเป็นอิสระต่อกัน เลเยอร์ทุกตัวจะส่งข้อมูลต่อกันไปเรื่อย โดยแต่ละโพโตคอลจะมีชั้นเลเยอร์ไม่เหมือนกัน ตามมาตรฐานที่ยอมรับจะแบ่งชั้นเลเยอร์ออกเป็น 7 เลเยอร์หรือที่เราเรียกย่อๆว่าชุดโปรโตคอลโอเอสไอ (OSI) ส่วนมากโปรโตคอล (OSI) จะนำมาอธิบายการสื่อสารระหว่างคอมพิวเตอร์ในเครือข่ายระหว่างเครื่องต่อเครื่อง แต่เราจะเลือกโปรโตคอล ทีซีพี/ไอพี (TCP/IP) ที่เป็นโปรโตคอลพื้นฐานของเครือข่ายบนอินเตอร์เน็ตมาใช้และสามารถนำมาใช้ได้จริง

- **การทำงานของทีซีพี/ไอพี (TCP/IP)**

โปรโตคอลทีซีพี/ไอพี จะเป็นที่นิยมมากในเครือข่ายบนอินเตอร์เน็ต การทำงานของมันจะมีการจัดรูปแบบที่แตกต่างกันจากแบบโอเอสไอเพราะเนื่องจากออกแบบมุ่งเน้นไปที่การเชื่อมต่อระหว่างระบบที่ต่างกันมากกว่า โดยโพโตคอลทีซีพี/ไอพี จะแบ่งโครงสร้างการทำงานออกเป็น 4 เลเยอร์ โดยเราจะนำมาเปรียบเทียบกับชุดโปรโตคอลโอเอสไอ

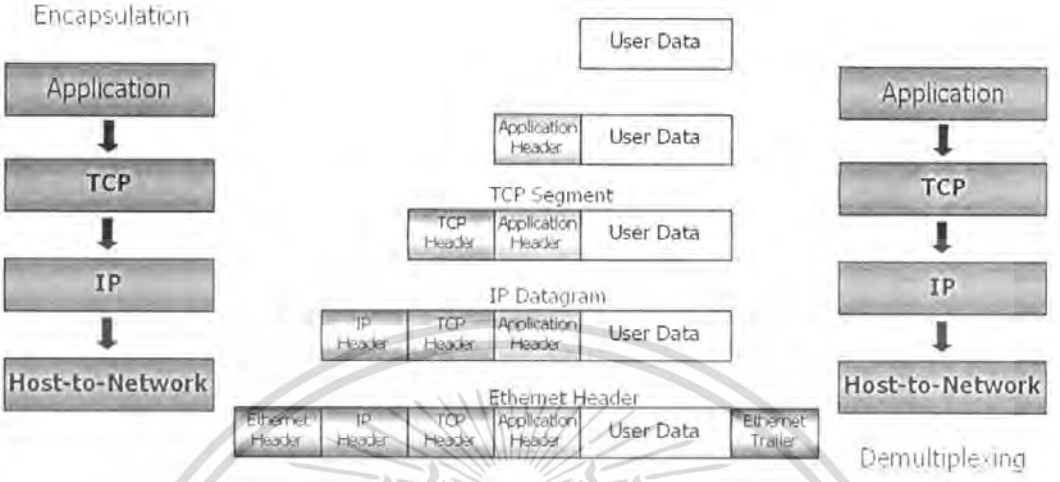


รูปที่ 2.1.1 เปรียบเทียบ ทีซีพี/ไอพี กับ โอเอสไอ

โดยรูป 2.1.1 จะแสดงการแบ่งของเลเยอร์โพโตคอลทีซีพี/ไอพี (TCP/IP) ที่แต่ละเลเยอร์ควรจะอยู่ในเลเยอร์ตามมาตรฐานโอเอสไอ (OSI) บ้าง โดยจะแบ่ง 4 เลเยอร์ออกเป็นดังนี้ ชั้นประยุกต์ใช้งาน (Application layer) ชั้นเชื่อมต่อระหว่างโฮสต์ (Transport layer) ชั้นอินเตอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Internet layer) และชั้นเข้าใช้เครือข่าย (Network access layer) ในการแบ่งแบบนี้บางเลเยอร์ในทีซีพี/ไอพี จะทำงานต่างกับเลเยอร์ในโอเอสไอ แต่จะบอกลักษณะคร่าวๆ



รูปที่2.1.2 ขั้นตอนการแปลข้อมูลและการแปลข้อมูลกลับ

โดยการทำงานใน รูปที่ 2.1.2 เริ่มแรกโปรแกรมประยุกต์ที่ผู้ใช้เป็นคนใช้นั้นจะส่งโปรโตคอลไปให้ชั้นประยุกต์การใช้งาน โดยชั้นนี้จะเพิ่มข้อมูลส่วนหัวเข้าไปซึ่งประกอบด้วยชื่อของคอมพิวเตอร์ที่ต้องการสื่อสารกับหมายเลขพอร์ตของเครื่องนั้น จากนั้นข้อมูลก็จะถูกส่งไปให้กับชั้นเชื่อมต่อระหว่างโฮสต์ โดยชั้นนี้ก็จะทำการเพิ่มข้อมูลส่วนหัวอีกเช่นกันจะมีโปรโตคอลให้เลือกใช้ระหว่างทีซีพี (TCP) หรือ ยูดีพี (UDP) โดยจะทำการแบ่งข้อมูลออกเป็นส่วนย่อยๆ เราเรียกแต่ละตัวว่า "เซ็กเมนต์ (Segment)" หลังจากนั้นก็จะส่งไปที่ชั้นอินเตอร์เน็ต โดยชั้นนี้ก็จะเพิ่มข้อมูลส่วนหัวลงไปอีก โดยข้อมูลที่เพิ่มจะเป็นพวกหมายเลขไอพี (IP) โปรโตคอลที่ใช้และเช็คซัม (checksum) โดยจะทำการตัดแบ่งออกเป็นส่วนๆอีก แต่ละส่วนจะเรียกว่า "แพ็คเกต (Packet)" จากนั้นจะทำการส่งต่อไปที่ชั้นเครือข่ายเพื่อทำการส่งไปที่ช่องสื่อสารต่อไป หลังจากนั้นพอถึงหมายเลขเครื่องที่เราจะส่งแล้ว มันจะทำวิธีการตรงกันข้ามกับที่กล่าวมาเพื่อแปลงข้อมูล การทำงานแต่ละเลเยอร์จะแบ่งการทำงานออกได้เป็นดังนี้

- **ชั้นเข้าใช้เครือข่าย (Network access layer)**

เลเยอร์นี้ทำหน้าที่ช่วยในการจัดส่งข้อมูลโดยผ่านสายสัญญาณแล้วแต่ว่าจะใช้สายสัญญาณแบบไหน โดยแต่ชนิดมีตัวควบคุมโปรโตคอลไม่เหมือนกันและมันจะทำการเพิ่มเฮดเดอร์เข้าไปในดาต้าแกรมเพื่อให้กลายเป็นเฟรมแล้วส่งต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **ชั้นอินเตอร์เน็ต (Internet layer)**

เลเยอร์นี้จะเป็นแบบระบบเครือข่ายแบบสลับช่องสื่อสารระดับแพ็คเกต (packet-switching network) ซึ่งเป็นการติดต่อแบบไม่ต่อเนื่อง (Connectionless) หลักการทำงานคือการปล่อยให้ข้อมูลขนาดเล็กที่เรียกว่า แพ็คเกต (Packet) สามารถไหลจากโหนดผู้ส่งไปตามโหนดต่างๆ ในระบบจนถึงจุดหมายปลายทางได้โดยอิสระ

1) ไอพี (Internet Protocol หรือ IP)

ไอพีเป็นโปรโตคอลในระดับเน็ตเวิร์คเลเยอร์ ทำหน้าที่จัดการเกี่ยวกับแอดเดรสและข้อมูล และควบคุมการส่งข้อมูลบางอย่างที่ใช้ในการหาเส้นทางของแพ็คเกต ซึ่งกลไกในการหาเส้นทางของไอพีจะมีความสามารถในการหาเส้นทางที่ดีที่สุด (Routing) และสามารถเปลี่ยนแปลงเส้นทางได้ในระหว่างการส่งข้อมูล และยังมีระบบการแยกและประกอบดาต้าแกรม (datagram) ดังรูป 2.1.3

การเชื่อมต่อของไอพีเพื่อทำการส่งข้อมูล จะเป็นแบบคอนเน็กชันเลส (Connectionless) หรือเกิดเส้นทางการเชื่อมต่อในทุกๆครั้งของการส่งข้อมูล 1 ดาต้าแกรม โดยโปรโตคอลนี้จะมีความน่าเชื่อถือน้อยเนื่องจากไม่มีการเชื่อมต่อกันระหว่างผู้ส่งกับคนรับ โดยโฮสต์ที่ต้องการส่งจัดทำการส่งไปอย่างเดียวจะไม่มีการตรวจสอบว่าส่งสำเร็จหรือไม่ การแก้ปัญหานี้จะปล่อยให้เลเยอร์ที่สูงกว่าเป็นคนตรวจสอบแทน

Version (4 bits)	1HL (4 bits)	Type of Service (8 bits)	Total Length (16 bits)
Identification (16 bits)		Flags (3 bits)	Fragment Offset (13 bits)
Time to Live (8 bits)	Protocol (8 bits)	Header Checksum (16 bits)	
Source Address (32 bits)			
Destination Address (32 bits)			
Options and Padding (multiples of 32 bits)			

รูปที่ 2.1.3 แสดงลักษณะเฮดเดอร์ของดาต้าแกรมไอพี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ไอซีเอ็มพี (Internet Control Message Protocol หรือ ICMP)

ไอซีเอ็มพีเป็นโปรโตคอลที่ใช้ในการตรวจสอบและรายงานสถานภาพของดาต้าแกรม (Datagram) ในกรณีที่เกิดปัญหากับดาต้าแกรม เช่น เร้าเตอร์ไม่สามารถส่งดาต้าแกรมไปถึงปลายทางได้ไอซีเอ็มพีจะถูกส่งออกไปยังโฮสต์ต้นทางเพื่อรายงานข้อผิดพลาด ที่เกิดขึ้น อย่างไรก็ดีไม่มีอะไรรับประกันได้ว่าข้อความไอซีเอ็มพีที่ส่งไปจะถึงผู้รับจริงหรือไม่ หากมีการส่งดาต้าแกรมออกไปแล้วไม่มีข้อความไอซีเอ็มพีฟ้องความผิดพลาดกลับมา ก็แปลความหมายได้สองกรณีคือ ข้อมูลถูกส่งไปถึงปลายทางอย่างเรียบร้อย หรืออาจจะมีปัญหา ในการสื่อสารทั้งการส่งดาต้าแกรมและ ข้อความไอซีเอ็มพีที่ส่งกลับมาก็มีปัญหาระว่างทางก็ได้ ไอซีเอ็มพีจึงเป็นโปรโตคอลที่ไม่มีความน่าเชื่อถือ การแก้ปัญหานี้จะปล่อยให้เลเยอร์ที่สูงกว่าเป็นคนตรวจสอบแทนเช่นกัน

- **ชั้นเชื่อมต่อระหว่างโฮสต์ (Transport layer)**

แบ่งเป็นโปรโตคอล 2 ชนิดตามลักษณะ ลักษณะแรกเรียกว่า "Transmission Control Protocol หรือ TCP" ข้อมูลที่มีปริมาณมากจะถูกแบ่งออกเป็นส่วนเล็กๆ เรียกว่า "message" ซึ่งจะถูกส่งไปยังผู้รับผ่านทางชั้นสื่อสารของอินเตอร์เน็ตและลักษณะที่สองเรียกว่า "User Datagram Protocol หรือ UDP" เป็นการติดต่อแบบไม่ต่อเนื่อง (connectionless) มีการตรวจสอบความถูกต้องของข้อมูลแต่จะไม่มีการแจ้งกลับไปยังผู้ส่ง

1) ยูดีพี (User Datagram Protocol หรือ UDP)

ยูดีพีนั้นจะเป็นการส่งครั้งละ 1 ชุดข้อมูล เรียกว่า "UDP datagram" ซึ่งจะไม่มีความสัมพันธ์กันระหว่างดาต้าแกรมและจะไม่มีกลไกการตรวจสอบความสำเร็จในการรับส่งข้อมูล กลไกการตรวจสอบโดยเช็คซัม (checksum) ของยูดีพีนั้นเพื่อเป็นการป้องกันข้อมูลที่อาจจะถูกแก้ไข หรือมีความผิดพลาดระหว่างการส่ง และหากเกิดเหตุการณ์ดังกล่าว ปลายทางจะได้รู้ว่ามีข้อผิดพลาดเกิดขึ้น แต่มันจะเป็นการตรวจสอบเพียงฝ่ายเดียวเท่านั้น โดยในข้อกำหนดของยูดีพีหากพบว่าความผิดพลาดตรงเช็คซัมก็ให้ผู้รับปลายทางทำการทิ้งข้อมูลนั้น แต่จะไม่มีการแจ้งกลับไปยังผู้ส่งแต่อย่างใด

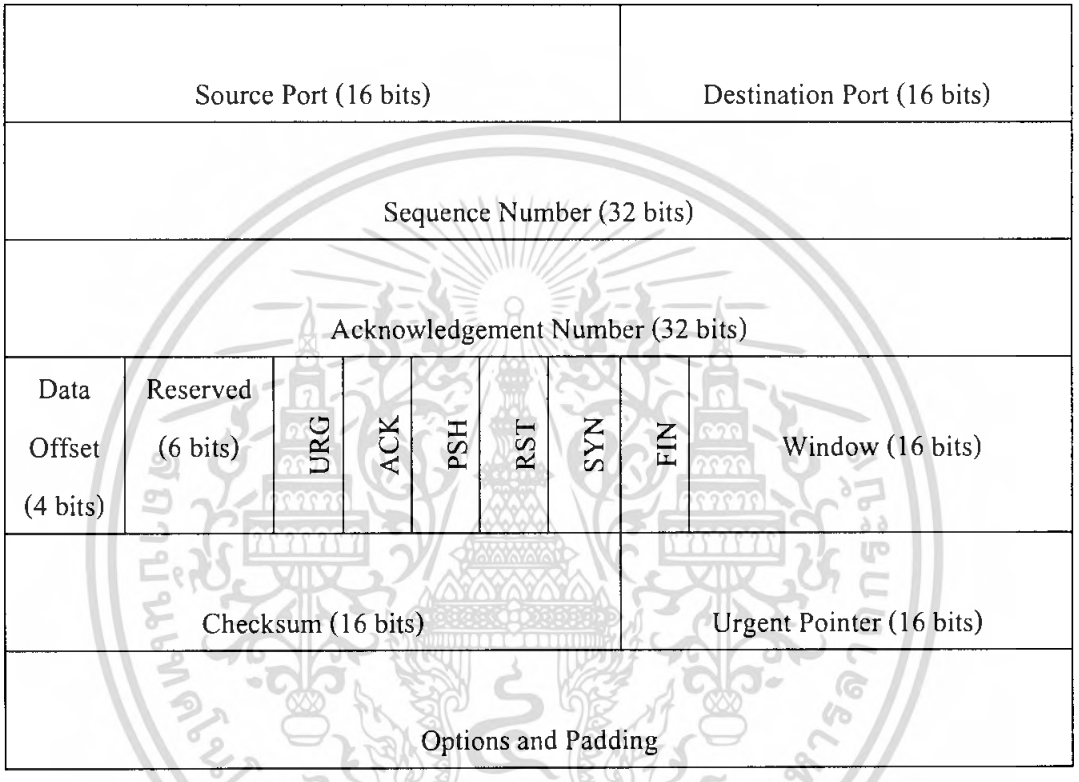
Source Port (16 bits)	Destination Port (16 bits)
Length (16 bits)	Checksum (16 bits)
Data…	

รูปที่ 2.1.4 แสดงลักษณะเฮดเดอร์ของดาต้าแกรมยูดีพี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ทีซีพี (Transmission Control Protocol หรือ TCP)

ทีซีพีทำหน้าที่จัดการและควบคุมการรับส่งข้อมูล ซึ่งมีความสามารถและรายละเอียดมากกว่ายูดีพีโดยดาต้าแกรมของทีซีพีจะมีความสัมพันธ์ต่อเนื่องกัน และมีกลไกควบคุมการรับส่งข้อมูลให้มีความถูกต้อง (reliable) และมีการสื่อสารอย่างเป็นกระบวนการ (connection-oriented) โดยหน่วยของข้อมูลทีซีพีนั้นเรียกว่า เซ็กเมนต์ (Segment) การส่งข้อมูลของทีซีพีจะส่งทีละเซ็กเมนต์



รูปที่ 2.1.5 แสดงลักษณะเฮดเดอร์ของดาต้าแกรมทีซีพี

- **ชั้นประยุกต์ใช้งาน (Application layer)**

เลเยอร์นี้จะเป็นเลเยอร์ชั้นบนสุดที่ติดต่อกับโปรแกรมประยุกต์ของผู้ใช้ โดยจะมีโปรโตคอลหลายประเภท เช่น เอฟทีพี (FTP หรือ File Transfer Protocol) ใช้ในการโอนถ่ายแฟ้มข้อมูล เทลเน็ต (Telnet) ใช้จำลองหน้าจอผ่านระบบเครือข่าย

2.1.2 ระบบตรวจจับการบุกรุก (Intrusion Detection System หรือ IDS)

ระบบตรวจจับการบุกรุกคือระบบที่ใช้ตรวจจับการใช้งานและความพยายามในการใช้งานเพื่อที่จะมุ่งร้ายต่อระบบคอมพิวเตอร์และทรัพยากรที่อยู่บนเครือข่าย โดยระบบจะตรวจจับผู้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บุกรุกก็คือระบบที่ประกอบไปด้วยฮาร์ดแวร์และซอฟท์แวร์สำหรับตรวจดูลักษณะพฤติกรรมผิดปกติ โดยผู้บุกรุกจะหาทางทุกวิถีทางเพื่อที่จะเข้ามาโจมตี ความปลอดภัยจึงเป็นส่วนสำคัญมากในระบบเครือข่าย โปรแกรมไอดีเอสโดยส่วนใหญ่จะใช้วิธีการตรวจสอบหรือศึกษารูปแบบการโจมตีซึ่งเป็นที่รู้จักกันดี นั่นหมายความว่า เทคนิคต่าง ๆ ที่ แฮคเกอร์ (hacker) ใช้ก็จะถูกโค้ดเข้าสู่ระบบเพื่อทำการตรวจจับเทคนิคนั้น วิธีการโดยทั่วไปคือการดรวจดูคอนเทนต์ (content) ในแพ็คเกตว่าประกอบด้วยรูปแบบที่อาจจะแสดงถึงการพยายามเข้าสู่ระบบ

2.1.2.1 กิจกรรมบนเครือข่าย

กระบวนการสื่อสารข้อมูลบนเครือข่ายนั้น แต่ละฝ่ายจะสื่อสารข้อมูลที่ได้รับเท่านั้นและกิจกรรมที่เกิดขึ้นต้องอาศัยเครือข่ายเป็นทางผ่านข้อมูลร่วมกันไม่ว่าจะเป็นวัตถุประสงค์ใด โดยการรับ-ส่งข้อมมูลจะจำแนกออกได้เป็นดังนี้

- **ข้อมูลสนเทศ (Information)**

ข้อมูลสนเทศจะเป็นข้อมูลที่แสดงออกมาเพื่อให้ผู้รับรับรู้ความหมายของผู้ส่ง โดยจะแสดงออกมาในรูปแบบต่างๆ ไม่ว่าจะเป็น ตัวอักษร ข้อความ รูปภาพ เสียง หรือสื่อต่างๆที่มนุษย์สามารถเข้าใจได้ ข้อมูลพวกนี้เมื่อทำการส่งผ่านเครือข่ายแล้วมันจะออกมาในรูปแบบของข้อมูลอิเล็คทรอนิค โดยเมื่อถึงผู้รับมันจะทำการแปลงกลับให้เป็นลักษณะข้อมูลสนเทศในลักษณะดังเดิม

ข้อมูลการรับ-ส่งข้อมูลสนเทศนั้นจะขึ้นอยู่กับการตกลงและข้อกำหนดของทั้งสองฝ่ายหรือโปรโตคอลที่ใช้ในนั้นเอง โดยโปรโตคอลที่กำหนดในชั้นประยุกต์การใช้งานจะเป็นตัวคอยจัดการ โดยกิจกรรมที่ใช้รับ-ส่งข้อมูลสนเทศจะเป็นกิจกรรมที่มีความเสี่ยงต่ำและการบุกรุกทางนี้ก็จะต่ำด้วย เพราะข้อมูลสนเทศนั้นจะถูกจัดการและจัดรูปแบบโดยแอพพลิเคชั่น โดยถ้าเกิดความเสียหายขึ้นก็จะแค่ข้อมูลสนเทศที่ต้องการที่จะรับจะไม่ตรงกับที่รับเข้ามา ตัวสนเทศจะถูกจัดรูปแบบจากโปรโตคอลและแอพพลิเคชั่นอย่างเข้มงวด

ไอดีเอสจะไม่ค่อยสนใจในการตรวจสอบการรับ-ส่งข้อมูลสนเทศเท่าไรเพราะเนื่องจากต้องใช้กำลังในการประมวลผลสูงมาก ประกอบกับส่วนนี้ไม่สามารถใช้ในการบุกรุกได้มากนัก

- **สัญญาณควบคุม (Control signal)**

ข้อมูลส่วนที่เป็นสัญญาณควบคุมนั้นมีความสำคัญต่อการสื่อสารข้อมูลมากเพราะเป็นส่วนหนึ่งของโปรโตคอล ผู้ใช้จะทั่วไปจะไม่ค่อยมีโอกาสเกี่ยวข้องกับส่วนนี้ หน้าที่ของข้อมูลส่วนนี้จะมีอยู่ 2 ประการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) ควบคุมจังหวะการรับ-ส่งข้อมูล

เนื่องจากการสื่อสารที่จะต้องใช้ระบบเครือข่ายนั้น เวลาทำการรับ-ส่งจะต้องมีตัวควบคุมสัญญาณไว้ ไม่ว่าจะเป็นส่วนของข้อมูลสนเทศและส่วนของสัญญาณควบคุมจะต้องถูกรวมส่งไปด้วยกัน โดยวิธีการส่งจะส่งไปเป็นลำดับ จะเริ่มส่งส่วนของสัญญาณควบคุมไปก่อนที่เรียกว่า "เฮดเดอร์ (Header)" และส่งส่วนของข้อมูลสนเทศตามไป โดยจะมีความยาวของข้อมูลเป็นตัวกำหนดในการแบ่งแยกระหว่าง เฮดเดอร์กับข้อมูล ตัวเฮดเดอร์จะเป็นตัวคอยควบคุมสัญญาณการรับส่ง เมื่อผู้รับทำการรับข้อมูลที่ส่งไปเริ่มแรกผู้รับต้องทำการอ่านส่วนของเฮดเดอร์มาประกอบด้วยเพื่อจะได้นำข้อมูลไปใช้อย่างถูกต้อง โปรโตคอล ทีซีพี/ไอพี จะประกอบด้วยสัญญาณควบคุมในทุกๆเลเยอร์

2) การสั่งงานให้อุปกรณ์ในเครือข่ายกระทำอย่างหนึ่งอย่างใด

นอกจากจะคอยควบคุมการรับ-ส่งได้แล้วยังสามารถที่จะออกคำสั่งได้โดยข้อมูลบางตัวที่อยู่ในบางโปรโตคอลที่สั่งงานให้อุปกรณ์ในเครือข่ายทำงานได้ โดยจะให้ตระหนักว่าเครือข่ายกับข้อมูลไม่ได้แยกกันโดยสิ้นเชิง คือ เครือข่ายไม่ได้เป็นแค่สำหรับเป็นที่ผ่านของข้อมูล และข้อมูลไม่ได้ใช้เครือข่ายเป็นแค่สื่อกลางโดยทั้งสองอย่างนี้จะมีผลกระทบซึ่งกันและกัน บางเหตุการณ์เครือข่ายอาจจะเปลี่ยนแปลงสภาพการทำงานโดยมีคำสั่งแอบแฝงภายในข้อมูล โดยเมื่อเกิดเหตุการณ์อย่างนี้ขึ้นรูปแบบเงื่อนไขจะเป็นไปตามที่ระบุไว้ในโปรโตคอล ทำให้อุปกรณ์ไม่สามารถแยกแยะว่าอันไหนเป็นคำสั่งที่ปกติและอันไหนเป็นคำสั่งที่มุ่งร้าย และถ้าเกิดเหตุการณ์นี้ขึ้นผู้ใช้จะไม่สามารถรับรู้ได้เลย

ส่วนนี้จะมีผลกระทบกับเครือข่ายมากกว่าข้อมูลสนเทศ และสามารถนำไปใช้ให้เกิดประโยชน์และโทษได้ โดยหลักพื้นฐานของไอดีเอสก็คือการวิเคราะห์พฤติกรรมของสัญญาณควบคุมเป็นอันดับแรก ซึ่งวิธีการตรวจจับแน้วโน้มการบุกรุกโดยส่วนใหญ่ก็จะดูที่เฮดเดอร์ของข้อมูลและคำสั่งก่อนเสมอ

2.1.2.2 ประเภทของไอดีเอส

- **ไอดีเอสที่ใช้ข้อมูลบนตัวเครื่อง (Host-Bascd IDSs)**

เป็นการทำงานที่ใช้ข้อมูลภายในระบบของตัวเครื่องส่วนบุคคลทำให้การทำงานของไอดีเอสชนิดนี้มีการวิเคราะห์กิจกรรมได้ดีกว่าแบบเครือข่ายไอดีเอส และสามารถทราบถึงผลของการป้องกันการบุกรุกได้ (แบบเครือข่ายไอดีเอสไม่สามารถทราบถึงผลการบุกรุกได้)

ไอดีเอสชนิดนี้จะแบ่งแหล่งข้อมูลออกเป็นสองประเภทคือการใช้บนระบบออดิตไทร์ (Audit Trails) และการใช้ระบบข้อมูลล็อค (System logs) ซึ่งแบบระบบออดิตไทร์นี้ส่วนมากจะสร้างขึ้นมาจากระดับภายในของระบบปฏิบัติการ ทำให้มีรายละเอียด และการป้องกันที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดีกว่าแบบระบบข้อมูลล็อค แต่ระบบข้อมูลล็อคก็มีข้อดีกว่าตรงที่มีขนาดเล็กกว่าความหมายของข้อมูลตรงมากกว่า และสามารถเข้าถึงข้อมูลได้ง่ายกว่า

บางไอดีเอสชนิดนี้จะสร้างมาให้ใช้ได้กับไอดีเอสแบบมีโครงสร้างการควบคุมและการรายงานอยู่ที่ส่วนกลาง (Centralized IDS) ทำให้การจัดการเพียงตัวเดียวสามารถดูแลหลายโฮสต์ และรูปแบบของข้อมูลก็ง่ายต่อการใช้ระบบการจัดการบนเครือข่าย

- **ไอดีเอสที่ใช้ข้อมูลบนเครือข่าย (Network-Based IDSs)**

ไอดีเอสตัวนี้เป็นไอดีเอสส่วนใหญ่ในตอนนี้ที่ใช้กันในระบบป้องกันของการค้าขาย ซึ่งจะทำการป้องกันการโจมตีโดยจับแพ็คเกตแล้วทำมาวิเคราะห์ จะนำมาจากบางส่วนบนเครือข่ายหรือบนสวิตซ์ (switch)

ส่วนมากไอดีเอสชนิดนี้จะประกอบด้วย ตัวกรองข้อมูล (single-purpose sensors) หรือโฮสต์ที่มีหลายจุดบนเครือข่าย (host placed at various points in a network) ซึ่งจะทำการคอยตรวจการไหลของข้อมูลบนเครือข่าย โดยจะมีการวิเคราะห์เป็นส่วน (local analysis) และค่อยทำการส่งข้อมูลของการบุกรุกไปยังส่วนควบคุมหลัก (Central Management Console) และตัวกรองข้อมูลตัวนี้จะเป็นตัวที่แอบซ่อนไว้เพื่อป้องกันผู้บุกรุกรู้ถึงตำแหน่งของระบบป้องกัน

2.1.2.3 ข้อดีของการใช้ไอดีเอส

- **การตอบสนองทันทีทันใด**

จากปกติเวลาถ้าจะต้องการวิเคราะห์ว่าข้อมูลไหนเป็นข้อมูลที่มุ่งร้าย เราจะต้องทำการเก็บข้อมูลมาทั้งหมดที่มันไหลอยู่ในช่วงเวลาใดเวลาหนึ่งแล้วมาทำการวิเคราะห์ว่าข้อมูลนั้นลักษณะเป็นเช่นไร รูปแบบเป็นไง ก็จะหมายความว่าจะต้องมีเหตุการณ์นี้เกิดขึ้นไปแล้วแล้วถึงนำมาวิเคราะห์ได้ แต่ไอดีเอส จะสามารถตรวจจับสิ่งที่ผิดปกติที่เกิดขึ้นได้ทันทีไอดีเอส จะทำงานอยู่ตลอดเวลาไม่มีหยุดและทำโดยอัตโนมัติ แล้วยังตอบสนองกับสิ่งที่มีพฤติกรรมแปลกปลอมได้รวดเร็ว

- **การมีพื้นฐานความรู้ของการวิเคราะห์**

ไอดีเอสจะมีตัวฐานข้อมูลที่เก็บลักษณะแพ็คเกตที่มีพฤติกรรมจะมุ่งร้าย โดยเริ่มแรกเมื่อมีข้อมูลเข้ามาทางเครือข่ายระบบจะทำการตรวจสอบว่าข้อมูลลักษณะนี้มีอยู่ในฐานข้อมูลไหม ถ้ามีก็จะทำการแจ้งเตือนอย่างรวดเร็ว แต่ถ้ายังไม่มีจะทำการตรวจสอบแล้วถ้ามีลักษณะพฤติกรรมที่มุ่งร้ายจะแจ้งเตือนและทำการเก็บลักษณะข้อมูลนี้ไว้ในฐานข้อมูล โดยไอดีเอสเปรียบเสมือนมีฐานความรู้ในการวิเคราะห์การบุกรุกได้ดีระดับหนึ่ง และขีดความสามารถเพิ่มขึ้นเรื่อยๆ ตามปริมาณที่เก็บอยู่ในฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **การช่วยตรวจสอบข้อบกพร่องของระบบป้องกันอื่นๆ**

โดยส่วนมากผู้คนจะรู้จักไฟร์วอลล์ (Firewall) เป็นอย่างดี ไฟร์วอลล์เป็นระบบที่ป้องกันสิ่งแปลกปลอมที่เข้ามาทางเครือข่ายเปรียบเสมือนมีกำแพงกั้นเครื่องคอมพิวเตอร์ แต่ไฟร์วอลล์สามารถป้องกันได้หมด ป้องกันได้เพียงแค่บางส่วนไอดีเอสจึงเป็นตัวรองรับอีกทีที่จะคอยกรองสิ่งที่ผ่านมาจากไฟร์วอลล์ และยังหาจุดอ่อนของไฟร์วอลล์ได้อีกด้วย

2.1.2.4 ข้อเสียของการใช้ไอดีเอส

- **การละเมิดความเป็นส่วนตัวบุคคล**

เนื่องจากไอดีเอส จะเป็นตัวที่คอยนำข้อมูลทั้งหมดที่อยู่บนเครือข่ายมาตรวจสอบหาพฤติกรรมที่มุ่งร้าย ส่วนนี้จึงทำให้ผู้ที่ตรวจสอบสามารถรู้ทุกสิ่งทุกอย่างที่อยู่บนเครือข่าย เช่น การโหลดข้อมูล, การแชทคุยกัน, อีเมลล์ และกิจกรรมที่สื่อสารบนอินเตอร์เน็ต ทำให้ผิดที่ไปละเมิดสิทธิส่วนบุคคล ดังนั้นผู้ที่ทำการตรวจสอบจะต้องเป็นคนที่ไว้วางใจได้ โดยจะต้องไม่นำเรื่องที่รู้ทางเครือข่ายนำไปเปิดเผยให้ผู้อื่นเสียหาย โดยข้อห้ามดังกล่าวเป็นข้อห้ามลำดับต้นๆ ในนโยบายรักษาความปลอดภัย โดยผู้นำไอดีเอส มาติดตั้งจะต้องได้รับอนุมัติจากหน่วยงานอย่างถูกต้องเท่านั้น

- **การตอบโต้อัตโนมัติ**

ไอดีเอสมีบางส่วนที่จำหน่ายอยู่ตามท้องตลาดที่ผู้ใช้สามารถกำหนดการดำเนินการกระทำอย่างหนึ่งที่หลังจากทำการค้นพบการบุกรุก โดยไอดีเอส จะทำการตอบโต้ไปที่ต้นกำเนิดของผู้บุกรุก โดยข้อนี้จะเป็นข้อเสียตรงที่บางทีเหมือนกับไปบอกผู้บุกรุกว่าเข้ามาไม่ได้ เปรียบเสมือนไปทำการท้าให้ผู้บุกรุกทำการบุกรุกเข้ามาอีกและไอดีเอส จะทำการป้องกันอีกที ฉะนั้นก็จะเกิดการสู้รบไม่รู้จบ หรือต้นกำเนิดของผู้บุกรุกเป็นแอดเดรสปลอมทำให้เราส่งการโจมตีย้อนกลับไปที่เครื่องใครก็ไม่รู้ทั้งที่เครื่องนั้นไม่ได้เกี่ยวด้วยเลย

- **การเตือนภัยผิดพลาด**

เมื่อเวลามีลักษณะข้อมูลที่เข้ามามีพฤติกรรมแปลกๆระบบจะทำการแจ้งเตือน แต่ระบบจะไม่รู้เลยว่าพฤติกรรมนั้นอาจจะเป็นแค่พฤติกรรมปกติแต่ลักษณะเหมือนไม่ปกติ จึงค่อยแจ้งเตือนตลอดเวลาทั้งที่เมื่อมีการเตือนแล้ว พอไปดูก็ไม่เห็นอะไรและเหตุการณ์นี้เกิดขึ้นบ่อยทำให้ไอดีเอส ขาดความหน้าเชื่อถือลง เมื่อมีพฤติกรรมที่ผิดปกติเกิดขึ้นแต่เกิดขึ้นพร้อมพฤติกรรมที่ปกติระบบทำการแจ้งเตือน ทำให้เราคิดว่าไม่มีอะไรที่ผิดปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2.5 สรุปความสามารถของไอดีเอส

ไอดีเอสเป็นเครื่องมือสำคัญในการรักษาความปลอดภัยในเครือข่าย ทำหน้าที่ในเชิงรุก (Proactive) สามารถทำให้ผู้ดูแลระบบสามารถป้องกันภัยคุกคามได้ล่วงหน้าก่อนที่การบุกรุกจะเกิดขึ้น หรือก่อนที่การบุกรุกจะทำได้สำเร็จ นอกจากนี้ไอดีเอส ยังสามารถช่วยในการเก็บหลักฐานทางอิเล็กทรอนิคส์ของการบุกรุกที่เกิดขึ้น สามารถนำมาวิเคราะห์และสืบค้นผู้กระทำผิดได้ในภายหลัง แต่ทั้งนี้ไอดีเอส มิใช่เครื่องมืออัตโนมัติที่สามารถจับการบุกรุกได้อย่างถูกต้อง 100 เปอร์เซ็นต์ การนำไปใช้งานต้องอาศัยความเข้าใจ และการปรับแต่งอย่างถูกต้องเหมาะสมกับสิ่งแวดล้อมรวมทั้งอาศัยการบำรุงรักษาและตรวจสอบอยู่สม่ำเสมอ

นอกจากนี้ไอดีเอส เองก็มีทั้งข้อดีและข้อเสียในตัวเองซึ่งผู้ที่จะนำไปใช้จะต้องตระหนักให้มาก และต้องคัดเลือบุคคลากรที่มีความชำนาญและมีความรับผิดชอบที่เหมาะสมให้เป็นผู้ดูแล เพื่อเป็นการป้องกันมิให้ไอดีเอส ถูกนำไปใช้ในทางที่ผิดและส่งผลร้ายต่อบุคคลมากกว่าการใช้การป้องกันระบบของตนเอง

2.1.3 การโจมตีทางเครือข่าย

เครือข่ายเป็นเทคโนโลยีที่อัศจรรย์ แต่ยังคงมีความเสี่ยงอยู่มากถ้าไม่มีการควบคุมหรือป้องกันที่ดี การโจมตีหรือการบุกรุกเครือข่ายหมายถึงความพยามยามที่จะเข้าใช้ระบบ (Access Attack) การแก้ไขข้อมูลหรือระบบ (Modification Attack) ทำให้ระบบไม่สามารถใช้งานได้ (Deny of Service Attack) และการทำให้ข้อมูลเป็นเท็จ (Repudiation Attack) ซึ่งทำให้ผู้ประสงค์ร้าย ผู้ไม่มีสิทธิ หรืออาจเกิดจากความไม่ตั้งใจของผู้ใช้เอง ต่อไปนี้รูปแบบต่างๆที่ผู้ไม่ประสงค์ดีพยายามที่จะบุกรุกเครือข่ายเพื่อลักลอบข้อมูลที่สำคัญหรือเข้าใช้ระบบโดยไม่ได้รับอนุญาติ

- **แพ็คเกตสนิฟเฟอร์**

อย่างที่กล่าวในข้างต้น เวลาคอมพิวเตอร์ต้องการส่งข้อมูลผ่านเครือข่ายนั้น ข้อมูลจะถูกแบ่งออกเป็นลักษณะที่เรียกว่า แพ็คเกต โดยที่เวลาส่งจะต้องผ่านโปรแกรมประยุกต์ที่ใช้งานและส่งผ่านไปในเลเยอร์แต่ละชั้น โดยจะมีการตรวจสอบเครือข่ายว่าแพ็คเกตในเครือข่ายมีข้อมูลอะไรบ้าง ตัวที่จะช่วยในการตรวจสอบแพ็คเกตจะเป็นตัวที่เรียกว่า “แพ็คเกตสนิฟเฟอร์ (Packet Sniffer)” เพื่อที่จำนำแพ็คเกตที่พฤติกรรมผิดปกติมาวิเคราะห์ว่าข้อมูลนั้นมีผิดภัยต่อระบบไหม แต่ตัวสนิฟเฟอร์จะมีข้อเสียอยู่ เนื่องจากปัจจุบัน โปรแกรมแพ็คเกตสนิฟเฟอร์มีให้ดาวน์โหลดบนอินเตอร์เน็ตมากมาย และผู้ใช้ไม่จำเป็นต้องมีความรู้เกี่ยวกับคอมพิวเตอร์มากนัก โดยผู้ที่ไม่ประสงค์ดีจะนำไปใช้ในทางที่ผิด เช่น ตรวจหาชื่อผู้ใช้และรหัสผ่าน เป็นต้น โดยส่วนนี้จะทำให้ผู้ใช้นำข้อมูลที่ได้มาไปโจมตีโปรแกรมประยุกต์ต่างๆได้อย่างง่ายดาย และจะทำให้ระบบเสียหาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **ไอพีสปูนฟิง**

ไอพีสปูฟิง (IP Spoonfing) หมายถึงการที่ผู้บุกรุกต้องการที่จะทำการบุกรุก จึงแกล้งทำเป็นว่าเครื่องคอมพิวเตอร์ที่ใช้ในการบุกรุกเชื่อถือได้ โดยจะใช้ไอพีแอดเดรสเหมือนกับที่ใช้ในเครือข่าย โดยจะทำการเพิ่มหรือเปลี่ยนแปลงข้อมูลลงไปในแพ็คเกตที่รับส่งระหว่างไคลเอนท์และเซิร์ฟเวอร์ การกระทำนี้ผู้ใช้จะต้องปรับเราท์ติ้งเทเบิ้ลของเราท์เตอร์เพื่อให้ส่งแพ็คเกตไปที่เครื่องผู้บุกรุกเสียก่อน โดยเหตุการณ์นี้ไม่จำเป็นที่จะต้องอยู่นอกเครือข่าย อาจจะอยู่ภายในเครือข่ายก็ได้

- **การโจมตีผ่านรหัส**

การโจมตีผ่านรหัส (Password Attack) อันนี้จะเป็นการโจมตีที่ผู้บุกรุกพยายามที่จะเดารหัสผ่านของผู้ใช้เพื่อที่จะนำไปใช้ในการบุกรุก จะมีวิธีต่างๆเพื่อที่จะหารหัสผ่าน โดยเมื่อผู้บุกรุกสามารถที่จะรู้รหัสผ่านแล้วผู้บุกรุกจะทำอะไรกับข้อมูลก็ได้ หรือตั้งรหัสใหม่เพื่อให้ผู้ใช้คนเดิมเข้าไม่ได้

- **การโจมตีแบบคนกลาง (Man-in-the-Middle)**

การโจมตีแบบนี้จะเป็นลักษณะที่ผู้บุกรุกสามารถเข้าถึงแพ็คเกตที่ส่งผ่านระหว่างเครือข่ายได้ โดยจะอยู่ระหว่างกลางระหว่างการส่ง แล้วใช้ตัวแพ็คเกตสนิฟเฟอร์ เป็นตัวดึงข้อมูลมา หรือนำไปวิเคราะห์การทำงานของเครือข่าย

- **การโจมตีแบบดอส (DOS)**

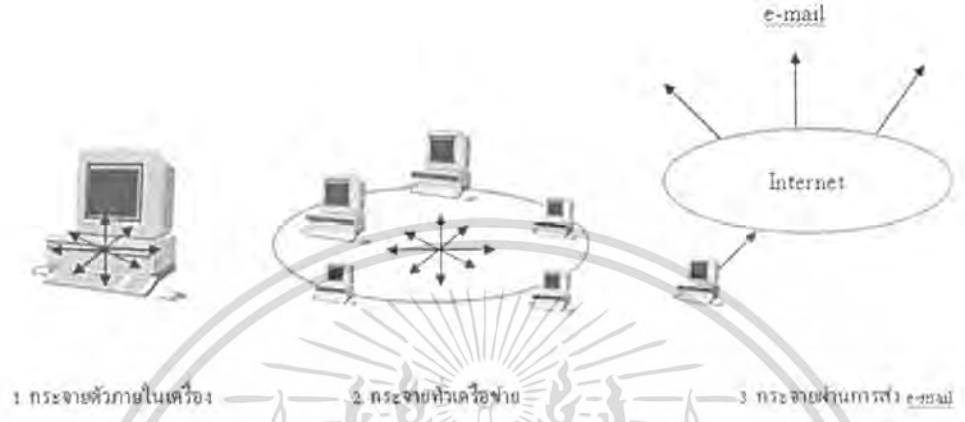
หมายถึง การโจมตีเซิร์ฟเวอร์เพื่อไม่ให้เซิร์ฟเวอร์ทำงานได้ ปกติจะทำให้ทรัพยากรของเซิร์ฟเวอร์หมด หรือเกินขีดจำกัด เพื่อไม่ให้สามารถทำงานให้บริการแก่เครื่องอื่นได้อีก โดยการโจมตีนี้จะอาศัยโปรโตคอลเป็นหลักในการทำ เช่น โปรโตคอลทีซีพี โปรโตคอลไอซีเอ็มพี เป็นต้น การโจมตีนี้ก็จะมีหลายลักษณะ เช่น ส่งข้อมูลไปเป็นจำนวนมากให้เซิร์ฟเวอร์ทำงานหนักๆ จนไม่สามารถไปทำงานกับเครื่องอื่นได้ เป็นต้น

- **การโจมตีโดยการใช้ไวรัส**

ไวรัสคอมพิวเตอร์คือ โปรแกรมคอมพิวเตอร์ประเภทหนึ่ง ที่ถูกเขียนขึ้นมาสอดแทรกและแอบแฝงไปกับไฟล์คอมพิวเตอร์ทั่ว ๆ ไป หรือแอบแฝงไปกับอีเมลล์ (e-mail) หรือสื่อในรูปแบบอื่น ด้วยวัตถุประสงค์ที่ จะไปก่อกวนเสถียรภาพในการทำงาน หรือทำลายข้อมูล หรือสร้างความเสียหายในรูปแบบอื่น ๆ ให้เครื่องคอมพิวเตอร์ โดยที่โปรแกรมไวรัสนี้สามารถที่จะแพร่กระจายตัวเองหรือขยายจำนวนตัวเองได้อย่างรวดเร็วภายในเครื่องคอมพิวเตอร์ที่ได้รับโปรแกรมตัวนี้เข้าไป และนอกจากนี้ยังมีไวรัสบางชนิดสามารถที่จะกระจายตัวเองข้ามจากเครื่องที่ติดไวรัสไปยังเครื่องคอมพิวเตอร์เครื่องอื่น ๆ ที่เชื่อมโยงกันในระบบเครือข่ายได้ และที่ยิ่งไปกว่านั้นยังมีบางชนิดที่สามารถทำการสำเนา (Copy) ตัวเองและแอบแฝงตัวไปในรูปแบบของอีเมลล์โดยมีความฉลาดที่จะจัดส่งสำเนาของตัวเองไปยังอีเมลล์แอเดรส ทั้งหมดที่ถูกบันทึกไว้ในโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รับส่งอีเมลล์ ของเครื่องที่ได้รับโปรแกรมไวรัสนั้น ๆ ด้วย ผลที่ตามมาก็คือความเสียหายที่เกิดขึ้น อันเนื่องมาจากการถูกทำลายของข้อมูล การทำงานที่ล่าช้าอืดอาดของเครื่องคอมพิวเตอร์ หรือถ้าถึงขั้นร้ายแรงก็อาจจะต้องทำการติดตั้งระบบปฏิบัติการ ใหม่ (Windows ฯลฯ) และทำการตั้งค่าที่เกี่ยวข้องทั้งหมดใหม่



รูปที่ 2.1.6 การกระจายของไวรัส

2.1.4 ชนิดของไวรัส

- **ไวรัส (Viruses)**

คือ โปรแกรมไวรัสคอมพิวเตอร์ที่สามารถแพร่กระจายตัวเองจากไฟล์หนึ่งไปยังอีกไฟล์หนึ่งภายในเครื่องคอมพิวเตอร์ โดยไวรัสจะกระจายตัวอย่างรวดเร็วไปยังทุกไฟล์ภายในคอมพิวเตอร์ หรืออาจจะทำให้ไฟล์เอกสารติดเชื้ออย่างช้า ๆ ซึ่งโปรแกรมคอมพิวเตอร์ที่เรียกว่า "ไวรัส" นี้ จะไม่สามารถแพร่กระจายจากเครื่องหนึ่งไปยังอีกเครื่องหนึ่งได้ด้วยตัวของมันเอง แต่จะแพร่ไปยังเครื่องอื่น ๆ ได้โดยอาศัยการนำพาของผู้ใช้ เช่นมีการแลกเปลี่ยนไฟล์ระหว่างเครื่องโดยใช้แผ่นดิสก์เก็ต การสำเนาไฟล์ที่ติดไวรัสไปไว้บนไฟล์เซิร์ฟเวอร์ และเมื่อมีผู้ใช้ทั่วไปรับไฟล์นั้นไปใช้งานก็จะแพร่กระจายในเครื่องนั้น ๆ และจะเป็นการแพร่กระจายขยายตัวตามวงจรในลักษณะนี้ต่อไป

- **หนอน (Worms)**

หนอนหรือวอร์มนั้น เป็นโปรแกรมคอมพิวเตอร์ที่สามารถสร้างความเสียหายต่อระบบได้จากภายใน (คล้ายกับตัวหนอนซึ่งกัดกินผลไม้จากภายใน) และจะสามารถแพร่กระจายตัวข้ามเครื่องไปยังเครื่องอื่นในระบบเครือข่ายได้โดยอัตโนมัติ โดยอาศัยพาหะเช่น อีเมลล์เป็นต้น สำหรับ "หนอน" นั้น มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสามารถในการกระจายตัวได้อย่างรวดเร็วและมีวงกว้าง จึงสามารถที่จะสร้างความเสียหายที่รุนแรงได้มากกว่า "ไวรัส" มาก

- **โทรจัน (Trojan)**

"โทรจัน" เป็นชื่อที่มาจากมหากาพย์เมืองทรอยของโฮมเมอร์ โดยถูกใช้เป็นชื่อของโปรแกรมคอมพิวเตอร์ที่ถูกออกแบบมาให้แฝงตัวเข้าไปในระบบและดักจับเอารหัสผ่านที่ใช้เชื่อมต่อเข้าสู่ระบบต่าง ๆ แล้วส่งกลับไปยัง ผู้ประสงค์ร้ายเพื่อใช้ในการเข้าโจมตีระบบในภายหลัง โดยแฝงมาในรูปแบบของ เกม การ์ดอวยพร อีเมลล์ ฯลฯ โปรแกรม โทรจันนั้นไม่ได้ถูกออกแบบมาให้ทำลายหรือสร้างความเสียหายให้แก่ระบบคอมพิวเตอร์ และไม่สามารถสำเนาตัวเองเพื่อแพร่กระจายได้ ซึ่งจะแตกต่างกับ "ไวรัส" และ "หนอน" แต่จะสร้างความเสียหายให้เกิดขึ้นได้เมื่อโทรจันทำงานและเปิดช่องทางต่าง ๆ ให้ผู้บุกรุกเข้าโจมตีระบบ

2.1.5 โปรแกรมสนอร์ท (Snort)

เราได้เลือกโปรแกรมสนอร์ท (Snort) มาตรวจสอบการตรวจจับการบุกรุกทางเครือข่าย โดยลักษณะโปรแกรมจะเป็นแบบใช้โปรแกมฟรีโดยสามารถที่โหลดมาลงได้โดยจะมีกฎต่างๆที่อยู่ภายในของโปรแกรมมันที่คอยตรวจสอบแพ็คเกตแปลกปลอม สาเหตุที่เลือกเพราะราคาจะไม่มีส่วนมาเกี่ยวข้อง, ความสามารถในการตรวจจับการบุกรุกค่อยข้างสูง, มีความเร็วระดับปานกลาง และสามารถที่จะโหลดมาอัพเดตเวอร์ชั่นได้จากเว็ปไซต์ได้ตลอดเวลา

สนอร์ท มีการทำงานแบบเอ็นไอดีเอส (NIDS : Network Intrusion Detection System) คือจะเป็นการทำงาน โดยตรวจสอบข้อมูลที่วิ่งอยู่ในเครือข่าย แบบเรียลไทม์ (Real Time) ว่าตรงกับกฎที่เก็บอยู่ในฐานข้อมูลหรือไม่ โดยจะตรวจสอบที่เฮดเดอร์ของแพ็คเกต หรือในเพย์โหลด (payload) โดยจะออกมาในรูปลักษณะสัญญาณหลายประเภท โดยจะส่วนนี้จะนำไปสร้างกฎของสนอร์ท (Snort Rule) เพื่อที่จะทำการตรวจสอบแพ็คเกตที่เข้ามา โดยสนอร์ทสามารถที่จะวิเคราะห์เฮดเดอร์ของชั้นเลเยอร์ที่ 3,4 ได้ โดยกฎจะถูกประยุกต์ให้ใช้ทันสมัย มีการอัพเดตตลอดเวลา กฎส่วนนี้จะนำไปใช้ในสัญญาณเตือน หรือบันทึกข้อความ โดยส่วนตัวมันนี้จะไปพิจารณาโปรโตคอลที่สำคัญคือโปรโตคอลทีซีพี/ไอพี

สนอร์ทจะเขียนกฎในไวยากรณ์ (syntax) ที่เข้าใจง่าย กฎส่วนใหญ่จะเขียนบรรทัดเดียว แต่เราสามารถเขียนได้หลายบรรทัดโดยใช้ \(backslash) คั่นเมื่อจบหนึ่งบรรทัด กฎปกติจะอยู่ในไฟล์คอนฟิกเกอร์เรชัน (configuration file : snort.conf)

- **โครงสร้างของกฎ**

กฎสนอร์ท ทุกอันจะมี 2 ส่วนลอจิคอล คือ เฮดเดอร์ของกฎ (rule header) และส่วนออปชันของกฎ (rule options) ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Rule Header	Rule Option

รูปที่ 2.1.7 โครงสร้างพื้นฐานของกฎสนอร์ท

1) เฮดเดอร์ของกฎ บรรจุข้อมูลเกี่ยวกับการกระทำ (Action) ที่กฎจะได้รับ รวมถึงเงื่อนไขที่เป็นมาตรฐานสำหรับไว้จับคู่กันระหว่างกฎกับแพ็คเกตข้อมูล ส่วนออปชันของกฎ บรรจุข้อความแจ้งเตือนและข้อมูลที่เกี่ยวกับส่วนแพ็คเกต ที่จะถูกใช้ในการสร้างข้อความแจ้งเตือน และยังเพิ่มเงื่อนไขที่เป็นมาตรฐานสำหรับไว้จับคู่กันระหว่างกฎกับแพ็คเกตข้อมูล กฎหนึ่งสามารถตรวจพบการบุกรุกได้ตั้งแต่หนึ่งถึงหลายประเภท กฎที่ฉลาดจะประยุกต์ใช้ได้กับหลาย รูปแบบการบุกรุก

โครงสร้างทั่วไปของเฮดเดอร์ของกฎสนอร์ท เป็นดังนี้

Action	Protocol	Address	Port	Direction	Address	Port

รูปที่ 2.1.8 โครงสร้างเฮดเดอร์ของกฎสนอร์ท

ส่วนการกระทำ จะพิจารณาว่าทำการกระทำแบบไหน เมื่อตรงกับเงื่อนไขที่เป็นมาตรฐานและกฎตรงกับแพ็คเกตข้อมูล โดยปกติการกระทำจะสร้างการแจ้งเตือนหรือบันทึกข้อความหรือเรียกกฎอื่น **ส่วนโปรโตคอล** กฎหรือแพ็คเกตจะประยุกต์ใช้กับสำหรับโปรโตคอลนั้นๆ โปรโตคอลที่ใช้ เช่น ไอพี, ไอซีเอ็มพี, ยูดีพี ฯลฯ **ส่วนแอดเดรส** จะกำหนดแอดเดรสต้นทางและปลายทาง แอดเดรสต้นทางและปลายทางจะพิจารณาจากไดเรกชันฟิลด์ (direction field) ตัวอย่างเช่น ถ้าไดเรกชันฟิลด์ เป็น -> ตำแหน่งด้านซ้ายจะเป็นต้นทางส่วนด้านขวาเป็นปลายทาง **ส่วนพอร์ต** ในกรณีของทีซีพีหรือยูดีพี พอร์ตจะใช้ในการพิจารณาต้นทางและปลายทาง ส่วนในกรณีของไอพีและไอซีเอ็มพี หมายเลขพอร์ตไม่มีความสำคัญ **ส่วนทิศทาง** (direction) ใช้พิจารณาแอดเดรสและพอร์ตว่า อันไหนเป็นต้นทางอันไหนเป็นปลายทาง

<u>ตัวอย่าง</u> พิจารณากฎ ที่จะสร้างข้อความแจ้งเตือนเมื่อพบไอซีเอ็มพีปิ๊งแพ็คเกต (ICMP ping packet) ด้วย ทีทีแอล (TTL)= 100

Alert icmp any any -> any any (msg: “Ping with TTL=100”;ttl:100;)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- แอคชัน ในกฎนี้แอคชันคือการแจ้งเตือนเมื่อมีการตรวจพบการบุกรุก
- โปรโตคอล ในกฎนี้โปรโตคอลคือไอซีเอ็มพี หมายความว่ากฎนี้ใช้ได้แค่แพ็คเกตประเภทไอซีเอ็มพี
- แอดเดรสและพอร์ตต้นทาง ในตัวอย่างนี้ใช้ได้กับทุกไอซีเอ็มพีแพ็คเกต ส่วนหมายเลขพอร์ตไม่เกี่ยวกับไอซีเอ็มพีแพ็คเกต
- ทิศทาง ใช้สัญลักษณ์ ->
- แอดเดรสและพอร์ตปลายทาง ในตัวอย่างนี้ใช้ได้กับทุกไอซีเอ็มพีแพ็คเกต ส่วนออปชัน จะอยู่ในวงเล็บ ( ) ซึ่งจะแสดงให้เห็นว่าข้อความแจ้งเตือนที่จะสร้างเท็กซ์สตริง "Ping with TTL=100" เมื่อเกิดทีทีแอล (TTL:Time To Live)=100

เฮดเดอร์ของกฎจะประกอบไปด้วย

1) แอคชันของกฎ (Rule action)

ทำแอคชันเมื่อกฎเป็นจริง ประกอบด้วย 5 แอคชัน

- **การผ่าน (Pass)**

ทำการเพิกเฉยต่อแพ็คเกต ใช้ในกรณีที่ต้องการหาช่องโหว่ในระบบ เพราะมันจะทำได้รวดเร็วกว่าแอคชันอื่น

- **การบันทึก (Log)**

ทำการบันทึกแพ็คเกต เช่นข้อความจะถูกบันทึกให้เป็นไฟล์หรือเก็บในฐานข้อมูลก็ได้ สามารถเก็บ รายระเอียด ได้หลายระดับตามแต่ที่กำหนดไว้ในคอมมานด์ไลน์อากิวเมนต์ (Command line argument) และไฟล์คอนฟิกเกอร์เรชัน (configuration file)

- **การแจ้งเตือน (Alert)**

ใช้สำหรับส่งข้อความแจ้งเตือนเมื่อกฎเป็นจริง การส่งส่งได้หลายวิธี เช่น เตือนผ่านทางไฟล์หรือเตือนผ่านทางคอนโซล ข้อแตกต่างระหว่างการบันทึกกับการแจ้งเตือนคือการแจ้งเตืองจะมีข้อความขึ้นมาและบันทึกแพ็คเกต ส่วนการบันทึกจะทำแค่การบันทึกแพ็คเกต

- **พลวัต (Dynamic)**

ถูกเรียกใช้เมื่อกฎอื่นใช้ "activate action" ตามปกติมันจะไม่ทำงาน มันจะทำงานก็ต่อเมื่อ แอคทิเวทแอคชัน (activate action) เรียกใช้เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
82767

- **ยูสเซอร์ดีไฟน์แอคชัน (User Defined Actions)**

สามารถกำหนดแอคชันได้เอง เช่น

- ส่งข้อความไปยังซิสล็อค (syslog) ซิสล็อคคือตัวบันทึกของระบบที่เป็นเดมอนโพรเซส (demon process) และสร้างไฟล์บันทึกที่ /var/log syslog เปรียบได้กับตัวบันทึกเหตุการณ์ของวินโดว์
- ส่งเอสเอ็นเอ็มพีแทร็ป (snmp traps) เอสเอ็นเอ็มพีแทร็ปจะถูกส่งไปยังระบบจัดการเครือข่าย
- ทำงานหลายๆ แอคชัน บน 1 แพ็คเกต เช่น ส่งเอสเอ็นเอ็มพีแทร็ปและ ข้อมูลบันทึการแจ้งเตือน (log alert data) ไว้ที่ซิสล็อค
- ข้อมูลการบันทึก (Log data) ให้เป็นเอ็กซ์เอ็มแอลไฟล์ (xml files)

สนอร์ทสามารถบันทึกข้อความเก็บไว้ในฐานข้อมูลของมายเอสคิวแอล (MySQL), โพสท์เกรสเอสคิวแอล (PostgressSQL), ออราเคิล และ ไมโครซอฟท์เอสคิวแอลเซิร์ฟเวอร์ (Microsoft SQL Server)

2) โปรโตคอล

ส่วนนี้จะแสดงให้เห็นว่ากฎจะจับแพ็คเกตบนโปรโตคอลไหน

3) แอดเดรส

แบ่งเป็น 2 ส่วน คือส่วนแรกเป็นแอดเดรสต้นทาง ส่วนหลังเป็นแอดเดรสปลายทาง จะเป็นไอพีแอดเดรสหรือเครือข่ายแอดเดรสก็ได้ คำสำคัญ “any” คือทุกๆแอดเดรส ตัวอย่างเช่น ถ้าต้องการสร้างการแจ้งเตือนสำหรับทุกๆ ทีซีพีแพ็คเกตที่มีค่าทีทีแอล (TTL:Time To Live) = 100 ที่วิ่งไปยังเว็บเซิร์ฟเวอร์ 192.168.1.10 ที่ พอร์ต 80 จากต้นทางใดๆ เขียนกฎได้ดังนี้

alert tcp any any -> 192.168.1.10/32 80 (msg : “”TTL=100 “;\ttl:100;”)

- **แอดเดรสที่ถูกยกเว้น (Address Exclusion)**

ใช้สำหรับแอดเดรสที่ยกเว้น ตัวอย่างเช่น จับทุกแพ็คเกตเว้นแต่ที่มาจาก 192.168.2.0

alert icmp ![192.168.2.0/24] any -> any any \ (msg : “Ping with TTL = 100 “ ;ttl : 100;)

ประโยชน์คือใช้ในการทดสอบแพ็คเกตที่ไม่ได้มาจากโฮมเครือข่าย (home network)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **รายชื่อแอดเดรส (Address List)**

ใส่แอดเดรสเป็นรายชื่อ ตัวอย่างเช่น โฮมเครือข่ายประกอบด้วยไอพี 192.168.2.0, 192.168.8.0

alert icmp ![192.168.2.0/24,192.168.8.0/24] any -> any any \ (msg : "Ping " ;)

- **หมายเลขพอร์ต (Port Number)**

ใช้ในการจับแพ็คเกต ว่ามาจากพอร์ตไหนและจะไปยังพอร์ตหรือช่วงของพอร์ตอะไร เช่นใช้พอร์ตต้นทาง หมายเลข 23 ไว้ดักแพ็คเกตที่มาจากเทลเน็ตเซิร์ฟเวอร์

2) ส่วนออปชันของกฎ (Rules Option) ทางเลือก (option) ของกฎ จะอยู่ต่อจากเฮดเดอร์ของกฎและอยู่ในวงเล็บ ซึ่งจะมีทางเลือกเดียวหรือหลายทางเลือกก็ได้ โดยแต่ละทางเลือกจะถูกคั่นด้วยเซมิโคลอน (;) ถ้าคุณเลือกใช้หลายทางเลือกมันจะเหมือนกับการแอนด์ (And) ทางลอจิก การกระทำของเฮดเอร์ของกฎจะเกิดขึ้นเมื่อทางเลือกทุกๆ ทางเป็นจริงขึ้นมา ทุกๆ ทางเลือกจะถูกประกาศโดยใช้คำสำคัญหรือบางตัวบรรจุอากิวเมนต์ (Argument) อยู่ข้างใน โดยทั่วไปแล้ว ทางเลือกนั้นอาจจะมี 2 ส่วนคือ ส่วนคำสำคัญ (Keyword) และส่วนอากิวเมนต์ (Argument) อากิวเมนต์จะถูกแยกจากคำสำคัญโดยโคลอน (:) เช่น msg : "Detected confidential";

- **คำสำคัญ "ACK"**

เฮดเดอร์ของทีซีพีจะบรรจุด้วย แถวของตัวเลขที่บอกถึงสิ่งต่างๆ ด้วยความยาวขนาด 32 บิต แถวของตัวนั้นจะแสดงถึงกระบวนการการรับแพ็คเกตต่อไปจากผู้ที่ส่ง แถวตัวเลขนี้จะปรากฏก็ต่อเมื่อแฟลกแอค (ACK Flag) ในทีซีพีเฮดเดอร์ที่ได้ถูกกำหนดไว้

- **คำสำคัญ "classtypc"**

กฎสามาถที่จะถูกแบ่งได้ตามชนิด และ ความสำคัญ โดยดูรายละเอียดได้จากไฟล์ Classification.conf ซึ่งแต่ละบรรทัดจะบรรจุไวยากรณ์ไว้ คือ ชื่อ คำอธิบาย และเลขลำดับความสำคัญ (Priority)

- **คำสำคัญ "content"**

คุณลักษณะสำคัญของสนอร์ทข้อหนึ่งคือสามารถค้นหาข้อมูลในแพ็กเก็ตได้ โดยข้อมูลนั้นอาจอยู่ในรูปของ แอสกี (ASCII) หรือตัวอักษร เช่นไวรัส, ผู้บุกรุก ซึ่งได้มีคำสำคัญ "content" และสัญลักษณ์ ในการค้นหาพวกมันในแพ็คเกตอยู่แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **คำสำคัญ "offset"**

ใช้ร่วมกับคำสำคัญ "content" สามารถที่จะเริ่มค้นหาได้ตั้งแต่ส่วนที่เป็นข้อมูล จุดเริ่มต้นของแพ็คเกตจะมีการใช้ตัวเลข และอากิวเมนต์ในคำสำคัญนี้

- **คำสำคัญ "depth"**

ใช้ร่วมกับคำสำคัญ "content" เช่นเดียวกัน เพื่อที่จะใช้กำหนดขอบเขตมากสุดของการจับคู่ สามารถที่จะระบุทิศทางจากจุดเริ่มต้นของแพ็คเกตได้ ข้อมูลหลังจากทิศทางที่ระบุไว้จะไม่ถูกตรวจหาข้อมูลในแพ็คเกตถ้าใช้ทิศทาง 2 ทางร่วมกันก็จะสามารถบอกถึงขนาดความยาวของแพ็คเกตได้

- **คำสำคัญ "content-list"**

ใช้กับชื่อของไฟล์ซึ่งเหมือนเป็นอากิวเมนต์ของคำสำคัญแบบนี้ เท็กซ์ไฟล์ตัวนี้จะบรรจุตัวอักษรซึ่งจะใช้ในการค้นหาภายในแพ็คเกต แต่ละอากิวเมนต์จะแบ่งเป็นบรรทัด

- **คำสำคัญ "dsize"**

จะถูกใช้เพื่อหาความยาวของส่วนที่เป็นข้อมูลในแพ็คเกต การโจมตีหลายๆ ครั้งจะใช้วิธีบัฟเฟอร์โอเวอร์โฟลว์ที่จะทำการส่งแพ็คเกตข้อมูลจำนวนมากเข้ามา ด้วยการใช้คำสำคัญนี้ทำให้เราสามารถระบุได้ว่าแพ็กเก็ตที่เข้ามามีขนาดเท่าไหร่และตรงกับการโจมตีแบบใดหรือเปล่า

- **คำสำคัญ "flags"**

ใช้ตรวจสอบว่าแฟลกใดในทีซีพีเฮดเดอร์ที่ส่งมาถูกเซตไว้บ้าง แต่ละแฟลกสามารถใช้เป็นอากิวเมนต์สำหรับกฎของสนอร์ทได้

- **คำสำคัญ "flagbits"**

ในทีซีพีเฮดเดอร์นั้นบรรจุด้วย 3 แฟลกบิต ซึ่งใช้สำหรับการแบ่งส่วนของข้อมูลและการทำรีแอสเซมบลี (Reassembly) บางบิตจะถูกใช้โดย แฮคเกอร์เพื่อที่จะบุกรุกและ ใช้ตรวจสอบข้อมูลบนเครือข่ายนั้น

- **คำสำคัญ "icmp_id"**

ใช้ตรวจจับเลขไอดีเจาะจงที่ใช้ร่วมกับไอซีเอ็มพีแพ็คเกต จะมีประโยชน์มากเมื่อค้นหาว่าแพ็คเกตใดถูกตอบกลับในการร้องแบบเจาะจง

- **คำสำคัญ "icmp_seq"**

ใช้ในการช่วยหาจำนวนตัวเลขที่เจาะจง เพื่อที่จะเอาไปเทียบกับกฎว่าตรงกับกฎไหนหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **คำสำคัญ "itype"**

ไอซีเอ็มพีเฮดเดอร์เข้ามาหลังจากไอพีเฮดเดอร์ และบรรจุด้วยแถวของ ชนิดของไอซีเอ็มพีนั้นๆ ซึ่งคำสำคัญนี้จะใช้ตรวจจับการบุกรุกซึ่งใช้แถวของชนิด ในไอซีเอ็มพีเฮดเดอร์

- **คำสำคัญ "icode"**

ไอซีเอ็มพีเฮดเดอร์เข้ามาหลังจากไอพีเฮดเดอร์และบรรจุด้วยแถวของรหัส (Code) ซึ่งคำสำคัญตัวนี้จะใช้ตรวจสอบตัวรหัสที่บรรจุในไอซีเอ็มพีเฮดเดอร์

- **คำสำคัญ "id"**

ใช้สำหรับจับคู่ส่วนของไอดีของไอพีแพ็กเก็ตเฮดเดอร์เพื่อตรวจจับการบุกรุกที่ทำการคงค่า (Fix) เลขไอดีเอาไว้ ถ้าไอดีเป็น 0 ก็จะหมายถึงตำแหน่งสุดท้ายของไอพีแพ็คเกต

- **คำสำคัญ "ipopts"**

โดยปกติไอพีวี4 (IPv4) จะมีความยาวของเฮดเดอร์เท่ากับ 20 ไบต์แต่อาจจะยาวได้ถึง 40 ไบต์ ไอดีออปชันใช้โดยมีหลายจุดประสงค์ เช่น บันทึกเส้นทาง บันทึกเวลา หรือใช้หาเส้นทาง แฮคเกอร์สามารถใช้ส่วนนี้หาข้อมูลจากเครือข่ายที่ใช้ได้

- **คำสำคัญ "ip_proto"**

ใช้ปลั๊กอินของไอพีโปรโต เพื่อที่จำคำนวณเลขโปรโตคอลที่อยู่ในไอพีเฮดเดอร์ คำสำคัญนี้ต้องใช้เลขโปรโตคอลเป็นอากิวเมนต์หรือสามารถใช้ชื่อของโปรโตคอลก็ได้

- **คำสำคัญ "logto"**

ใช้เพื่อเก็บแพ็คเกตไปยังไฟล์พิเศษ อย่าลืมระบุพาธ (Path) ของไฟล์ที่จะเก็บด้วย

- **คำสำคัญ "msg"**

ใช้เพื่อใส่ตัวอักษรหรือข้อความลงไปในล็อก (Logs) และการแจ้งเตือน โดยใส่ข้อความลงไปใน ""

- **คำสำคัญ "nocase"**

ใช้ร่วมกับคำสำคัญ "content" โดยตัวมันเองไม่มีอากิวเมนต์ จุดประสงค์หลักคือต้องการค้นหาข้อมูลที่อยู่ในรูปแบบอินเซนซิทิฟ (Insensitive) ในแพ็คเกต

- **คำสำคัญ "priority"**

ใช้ใส่ค่าความสำคัญลงไปในกฎ โดยที่ค่าความสำคัญเท่ากับ 1 จะเป็นค่าที่ทำให้กฎนั้นมีความสำคัญสูงสุด ใช้ในการแยกการแจ้งเตือนแบบต่างๆ ออกจากกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **คำสำคัญ "react"**

ใช้ในการทำลายเซสชัน (Session) ปิดกั้นบางเว็บไซต์ หรือบางเซอร์วิส (Service) โดยจะใส่ไว้ส่วนท้ายสุดของกฎ

- **คำสำคัญ "reference"**

ใช้ในการเพิ่มคำอธิบายลงไปในกฎ

- **คำสำคัญ "resp"**

เป็นคำสำคัญที่สำคัญมาก มันสามารถจัดการกับกิจกรรมที่มาจากแฮคเกอร์ โดยส่งแพ็คเกต ไปยังโฮสต์ที่ส่งแพ็คเกตมาตรงกับกฎที่กำหนดไว้

- **คำสำคัญ "rev"**

ใช้เพื่อที่จะแสดงจำนวนรีวิชัน (revision) สำหรับกฎ ถ้าคุณปรับปรุงกฎ คุณสามารถใช้คำสำคัญนี้ในการแสดงจำนวนกฎที่เปลี่ยนไปได้

- **คำสำคัญ "rpc"**

ใช้ตรวจจับการร้องขออาร์พีซี (RPC) โดยใช้ 3 อากิวเมนต์คือเลขของแอพพลิเคชัน เลขโปรซีเยอร์ และเลขเวอร์ชัน

- **คำสำคัญ "sameip"**

ใช้เพื่อเช็คว่าไอพีของจุดกำเนิดกับไอพีปลายทางตรงกันหรือเปล่า เนื่องจากมีการทำการปลอมแปลงไอพีได้

- **คำสำคัญ "seq"**

ใช้ทดสอบลำดับตัวเลขของทีซีพีแพ็คเกต ซึ่งอากิวเมนต์ของตัวมัน

- **คำสำคัญ "flow"**

ใช้เพื่อคำนวณทิศทางของแพ็คเกต

- **คำสำคัญ "session"**

ใช้ในการแสดงแพ็คเกตทั้งหมดที่มาจากทีซีพีเซสชันหรือเฉพาะแค่บางตัวก็ได้ ถ้าใช้ "all" จะเป็นการแสดงทั้งหมด

- **คำสำคัญ "sid"**

ใช้ในการใส่ไอดีของสนอร์ตลงไปยังกฎ เพื่อที่จะให้ส่วนแสดงผลหรือส่วนเก็บข้อมูลเรียกใช้กฎได้ถูก

- **คำสำคัญ "tag"**

เป็นคำสำคัญที่สำคัญอีกตัวหนึ่ง ซึ่งจะใช้เป็นเก็บข้อมูลที่เข้าหรือออกจากโฮสต์ของผู้บุกรุก เมื่อกฎได้ถูกดำเนินการ เพื่อใช้ในการวิเคราะห์พฤติกรรมของผู้บุกรุกในภายหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **คำสำคัญ "tos"**

ใช้ตรวจสอบค่าที่เจาะจงของรูปแบบการใช้บริการ ในไอพีเฮดเดอร์

- **คำสำคัญ "ttl"**

ใช้ตรวจสอบค่าเวลาที่ยังอยู่ (Time To Live) ในไอพีเฮดเดอร์ของแพ็กเก็ต ตัวคำสำคัญควรจะต้องไม่ค่าเท่ากับค่าที่ประมาณไว้พอดี สามารถใช้ได้กับทุกๆ โปรโตคอล ใช้ในการตรวจสอบว่ามีคนพยายามจะทำการ ติดตามเส้นทาง (Traceroute) มายังเครือข่ายของเราหรือไม่

- **คำสำคัญ "uricontent"**

เหมือนกับคำสำคัญ "content" จะแตกต่างตรงที่มันจะคอยตรวจสอบความผิดปกติที่ส่วนยูอาร์ไอ (URI) ในแพ็คเกตเท่านั้น

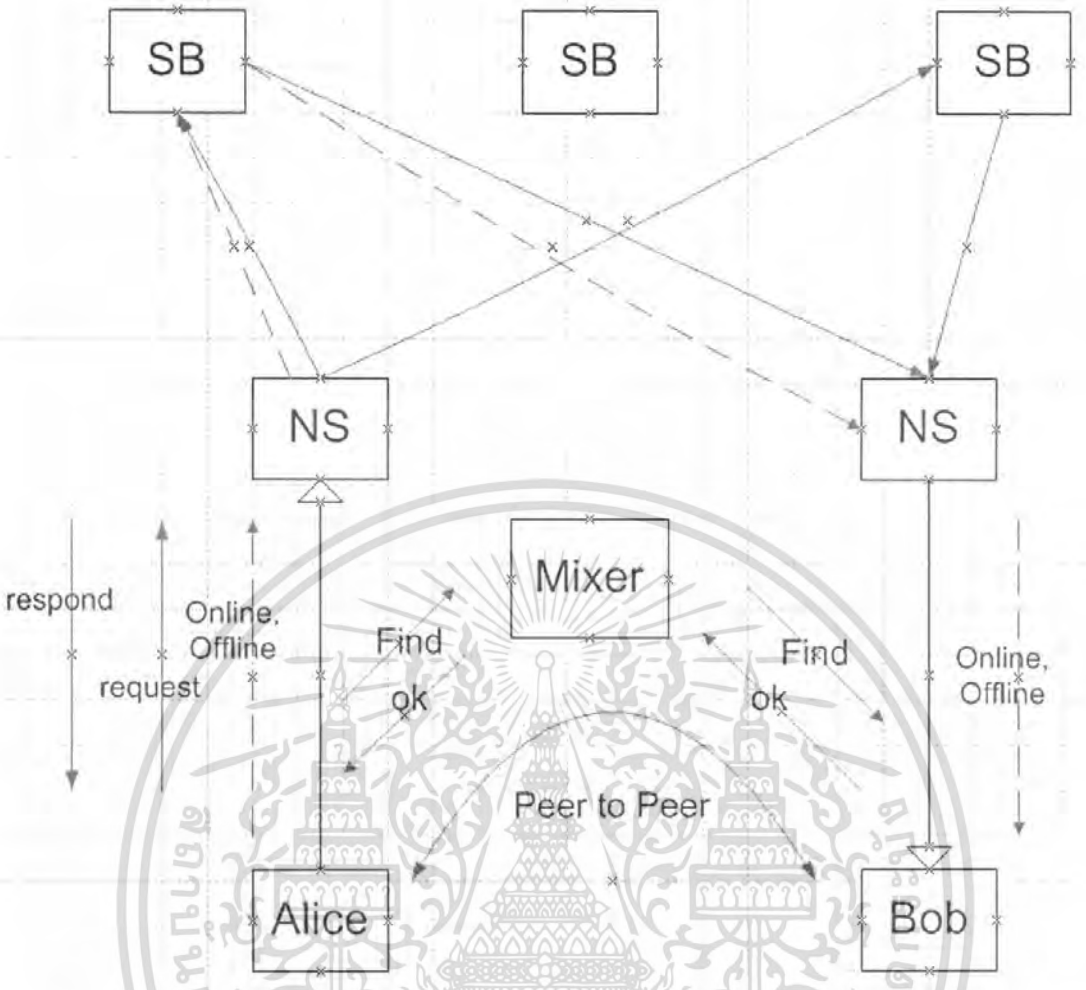
2.1.6 โปรแกรมเอ็มเอสเอ็นแมสเซ็นเจอร์ **(MSN Messenger)**

เอ็มเอสเอ็นแมสเซ็นเจอร์เป็นโปรแกรมที่ใช้คุยผ่านทางอินเตอร์เน็ต โปรแกรมนี้มีการพัฒนามานานจนปัจจุบันพัฒนาถึงเวอร์ชั่น 8.0 โดยผู้ที่ต้องการใช้โปรแกรมนี้จะต้องมีเมลล์เป็นของตัวเอง เวอร์ชั่นก่อนๆจะต้องเป็นเมลล์ของ ฮฮทเมลล์ (Hotmail) เท่านั้น แต่ในเวอร์ชั่นปัจจุบันได้มีการพัฒนามากขึ้นทำให้สามารถที่จะใช้เมลล์อย่างอื่นได้เช่น ย๊าฮู (Yahoo) โดยผู้ที่ใช้เมลล์อื่นจะต้องทำการสมัครสมาชิกกับทางฮฮทเมลล์ก่อน โปรแกรมนี้เมื่อเปิดขึ้นมาจะเป็นเมลล์ปล่าว โดยเมื่อเราจะต้องทำการแอดเมลล์คนอื่นลงไป หลังจากนั้น เครื่องของคนที่เราแอด เมื่อคนคนนั้นเปิดเครื่องขึ้นมา โดยโปรแกรม เอ็มเอสเอ็นแมสเซ็นเจอร์จะต้องทำการตอบรับว่ายอมรับการแอด จากนั้นผู้ใช้ทั้ง 2 ฝ่ายก็สามารถที่จะสื่อสารกันได้ โดยวิธีต่างๆ ไม่ว่าจะเป็น ทางตัวอักษร ทางวีดีโอ ทางเสียงเป็นต้น

2.1.6.1 โครงสร้างการทำงานของโปรแกรมเอ็มเอสเอ็นแมสเซ็นเจอร์ (MSN Messenger)

ลักษณะการทำงานของโปรแกรมเอ็มเอสเอ็นแมสเซ็นเจอร์จะเป็นการทำงานแบบไคล์แอนท์กับเซิร์ฟเวอร์ (Client-Sever) โดยเมื่อทำการติดตั้งโปรแกรมเอ็มเอสเอ็นแมสเซ็นเจอร์ภายในเครื่องคอมของผู้ใช้ เมื่อผู้ใช้ทำการเปิดโปรแกรมที่เป็นไคล์แอนท์ (Client) ขึ้นมาจะทำการแจ้งไปยังเครื่องเซิร์ฟเวอร์และเครื่องเซิร์ฟเวอร์ก็จะแจ้งไปยังเมลล์ต่างๆที่อยู่ในลิสล์ภายในโปรแกรมเอ็มเอสเอ็นแมสเซ็นเจอร์ ของผู้ใช้ว่าผู้ใช้ได้ทำการเปิดสถานะออนไลน์ (online) ขึ้นมาแล้ว ทำให้ผู้ที่อยู่ในลิสล์หรือ เครื่องไคล์แอนท์ต่างๆสามารถทราบได้ว่าผู้ใช้อยู่ในสถานะใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1.9 โครงสร้างภายในของระบบเอ็มเอสเอ็นแมสเซ็นเจอร์

จากรูปที่ 2.1.9 จะเป็นการอธิบายการสื่อสารระหว่าง เครื่อง ไคล์แอนท์ 2 เครื่อง โดยเครื่องแรกจะชื่ออาลิส (Alice) ส่วน อีกเครื่องหนึ่งชื่อบ๊อบ (Bob) เริ่มแรก อาลิสจะอยู่ในสถานะออนไลน์ และเมื่ออาลิส (Alice) เปลี่ยนสถานะเป็นออฟไลน์ (offline) เครื่องไคล์แอนท์จะทำการติดต่อไปยัง ซีเอส (Connection Server หรือ CS) ว่าเครื่องอยู่ในสถานะ ออฟไลน์จากนั้นซีเอสก็จะแจ้งไปยัง พีเอส (Person Server หรือ PS) ว่าเครื่องอาลิสได้เปลี่ยนสถานะเป็นออฟไลด์ โดย ซีเอสจะทำการใช้ฟังก์ชั่นแฮช (Hash) เชื่อมกับพีเอส จากนั้นพีเอส จะไปทำการเชื่อมกับซีเอสอีกตัว หลังจากนั้น ซีเอส ตัวนี้จะทำการแจ้งสถานะของอาลิสให้แก่บ๊อบ

การสื่อสารระหว่างอาลิสกับบ๊อบ ถ้าอาลิสต้องการที่จะส่งไฟล์ไปให้บ๊อบจะเริ่มต้นจากอาลิสร้องขอไปยังซีเอสว่าต้องการสื่อสารกับบ๊อบ หลังจากนั้นพีเอส จะส่งข้อมูลยืนยันโอเคอาลิสก็จะไปทำการใช้โปรโตรคอลมิกเซอร์ (Mixer) ในการค้นหาบ๊อบ จากนั้นบ๊อบก็จะยืนยันกับมายังโปรโตรคอลมิกเซอร์และอาลิส กับบ๊อบก็จะสามารถติดต่อสื่อสารกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.6.2 การทำงานของเอ็มเอสเอ็ม (MSN)

เอ็มเอสเอ็นแมสเซ็นเจอร์หมายถึงส่วนต่างๆ ที่ใช้ในการส่งข้อความในการสนทนา โดยผู้ใช้จะคุยข้ามผ่านเครือข่ายเอ็มเอสเอ็นแมสเซ็นเจอร์และโปรแกรมที่ผู้ใช้นิยมใช้มากที่สุดคือ เอ็มเอสเอ็นแมสเซ็นเจอร์ (MSN Messenger) ซึ่งภาษาที่จะใช้ในการเขียนในการพูดคุยบนเครือข่ายคือเอ็มเอสเอ็มโปรโตคอล ซึ่งเครือข่ายเอ็มเอสเอ็นแมสเซ็นเจอร์ คือเครือข่ายที่ใช้ส่งผ่านข้อความในคณะนั้น

การทำงานของโปรแกรมบนเครื่องของผู้ใช้ที่ถูกเรียกว่าเอ็มเอสเอ็นแมสเซ็นเจอร์ (MSN Messenger) ฝั่งไคล์แอนท์ จะทำการเชื่อมต่อกับ เอ็มเอสเอ็นแมสเซ็นเจอร์ของทางฝั่งเซิร์ฟเวอร์ข้ามผ่านอินเตอร์เน็ต หรืออาจกล่าวได้ว่า ไคล์แอนท์ทำการส่งและรับข้อมูลด้วยกันเองผ่านทางเซิร์ฟเวอร์ โดยปกติแล้วไคล์แอนท์จะทำการสนทนากับทางฝั่งเซิร์ฟเวอร์ นำข้อมูลไปผ่านกระบวนการและค่อยส่งคนอื่นๆ แต่ข้อมูลบางอย่างอาจจะผ่านเซิร์ฟเวอร์โดยที่ไม่ได้ผ่านกระบวนการของเซิร์ฟเวอร์แต่จะไปผ่านกระบวนการโดยไคล์แอนท์แทน

ภาษาที่จะถูกใช้ในการติดต่อสื่อสารระหว่างโปรแกรมคอมพิวเตอร์สองเครื่องจะถูกเรียกว่าโปรโตคอล ส่วนกฎสำหรับการส่งข้อความระหว่างเอ็มเอสเอ็นแมสเซ็นเจอร์ของไคล์แอนท์กับเซิร์ฟเวอร์จะถูกเรียกว่าเอ็มเอสเอ็นแมสเซ็นเจอร์โปรโตคอล (MSN Messenger protocol) ส่วนกฎที่ส่งมาจากหนึ่งไคล์แอนท์ให้กับตัวอื่นๆ ผ่านเซิร์ฟเวอร์เรียกว่าเอ็มเอสเอ็นไคล์แอนท์โพโตรคอล (MSN Client protocol)

เอ็มเอสเอ็นแมสเซ็นเจอร์โปรโตคอล (MSN Messenger protocol) นี้จะประกอบด้วยชุดของคำสั่งที่ส่งระหว่างไคล์แอนท์กับเซิร์ฟเวอร์ อย่างเช่นผู้ใช้งานบนรายชื่อออฟไลน์ เซิร์ฟเวอร์จะส่งข้อความไปทางฝั่งไคล์แอนท์ดังนี้

FLN myname_123@hotmail.com

เมื่อทางฝั่งไคล์แอนท์ได้รับข้อความแล้วจะนำชื่อของผู้ใช้ออกจากรายชื่อที่ออนไลน์และนำไปใส่ในรายชื่อผู้ใช้ออฟไลน์แทน

เอ็มเอสเอ็นไคล์แอนท์โพโตรคอลจะประกอบด้วยข้อความที่ไคล์แอนท์ส่งด้วยกันเอง ตัวอย่างเช่น ถ้าหากผู้ใช้งานพิมพ์ ‘Hello’ ให้กับผู้ใช้งานอีกคน ทางฝั่งไคล์แอนท์จะส่งข้อความให้กับไคล์แอนท์ของพวกเค้าเองด้วย ‘Hello’ เป็นรูปร่างของข้อความ ซึ่งการทำงานของโปรโตคอลนั้น โปรแกรมเอ็มเอสเอ็นแมสเซ็นเจอร์จะติดต่อกับ เอ็นเอส (Notification Server หรือ NS) ซึ่งจะทำการเชื่อมต่อกับเอสบี (Switchboard Servers หรือ SB) ซึ่งเอสบีจะทำหน้าที่เกี่ยวกับการส่งข้อความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **เอ็นเอส (Notification Server หรือ NS)**

การเชื่อมต่อกับเอ็นเอสเป็นส่วนพื้นฐานของเอ็มเอสเอ็นแมสเซ็นเจอร์โดยจะอยู่ในส่วนของข้อมูลผู้ใช้ในขณะนั้น ถ้าหากว่าเราไม่ทำการเชื่อมต่อกับเอ็นเอส ผู้ใช้จะไม่อยู่ในรายชื่อออนไลน์ในรายชื่อของผู้ใช้คนอื่น

เอ็นเอสยังเตรียมบริการอื่นๆ อย่างเช่น มีอีเมลล์ใหม่เข้ามาให้ผู้ใช้งาน และการสร้าง สวิตซ์บอร์ดเซสชั่น (Switchboard sessions) ใหม่ให้กับผู้ใช้พร้อมทั้งเปิดเอ็นเอสไว้อยู่

- **สวิตซ์บอร์ด (Switchboard หรือ SB)**

สวิตซ์บอร์ดจะทำงานในส่วนของข้อความขณะนั้นระหว่างผู้ใช้ด้วยกัน หรืออาจกล่าวได้ว่าผู้ใช้แต่ละคนในเอ็มเอสเอ็นติดต่อ สวิตซ์บอร์ดเซสชั่นร่วมกัน ถ้าหากว่าผู้ใช้มีการสนทนากับสองคนพร้อมกัน หมายความว่าจะทำการเชื่อมต่อกับ สวิตซ์บอร์ดเซิร์ฟเวอร์ 2 เซิร์ฟเวอร์ พร้อมกัน ซึ่งสวิตซ์บอร์ดทำหน้าที่เป็นตัวแทนระหว่างผู้ใช้ในการสื่อสารกันนั่นเอง ซึ่งสวิตซ์บอร์ดเซสชั่นอาจจะมีเยอะเท่าที่ผู้ใช้ต้องการจะทำการสนทนา

2.1.6.3 การขอสวิตซ์บอร์ด แบ่งออกเป็น 2 วิธี คือ

การสร้างสวิตซ์บอร์ด ใหม่

ผู้ใช้ที่ทำการเปิดหน้าต่างเพื่อจะทำการสนทนากับผู้ใช้ท่านอื่นจะต้องทำการร้องขอสวิตซ์บอร์ดใหม่ โดยไคล์แอนท์จะทำการสร้างการเชื่อมต่อทีซีพีเพื่อที่จะนำสวิตซ์บอร์ดไอพี ที่ได้รับมาจากเซิร์ฟเวอร์ไปเชื่อมต่อกับพอร์ตที่กำหนดมาให้ ซึ่งผู้ใช้จะรอการเชื่อมต่อประมาณสองนาที สวิตซ์บอร์ดก็จะทำการเชื่อมต่อ

ในตัวอย่างของการเชื่อมต่อของสวิตซ์บอร์ด ไคล์แอนท์จะทำการส่งคำสั่งไป โดยก่อนหน้านั้นจะส่ง สวิตซ์บอร์ดไอพีและพอร์ตที่เซิร์ฟเวอร์ส่งมาให้ในตอนแรกไปที่เซิร์ฟเวอร์ ในส่วนของการร้องขอจะต้องมีข้อมูลดังนี้ ทีอาร์ไอดี (TrID) ชื่อของผู้ใช้งานจะอยู่ในส่วนแรก และชื่อในการเข้าใช้งานจะอยู่ในส่วนที่สอง คำสั่งนี้จะใช้เวลาประมาณหนึ่งนาที แล้วสวิตซ์บอร์ดจะทำการปิดการเชื่อมต่อ

ถ้าหากการร้องขอสำเร็จ เซิร์ฟเวอร์จะตอบกลับผู้ใช้งาน พร้อมกับทีอาร์ไอดี อยู่ในส่วนแรก ชื่อของผู้ใช้งานอยู่ในส่วนที่สอง และชื่อดิสเพลย์ต่อผู้ใช้คนอื่นในส่วนที่สาม

1. การถูกเชิญเข้าสวิตซ์บอร์ด ที่มีอยู่แล้ว

การทำงานของสวิตซ์บอร์ดชนิดนี้จะมีการทำงานสองส่วนคือ

การทำงานในสวิตซ์บอร์ดเซสชั่น (Switch Session) ผู้ใช้งานที่ถูกเชิญจะได้รับคำสั่ง (RNG) จากเอ็นเอส ไคล์แอนท์ผู้นั้นจะเชื่อมต่อกับสวิตซ์บอร์ดเซสชั่นอัตโนมัติ และจะไม่เปิดหน้าต่างจนกว่าคำพูดของผู้ใช้จะถูกส่งมา โดยจะประกอบด้วย 5 ส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนแรกของคำสั่งจะเป็นส่วนของเซสชั่นไอดีของสวิตซ์บอร์ดเซสชั่น โดยที่ สวิตซ์บอร์ดเซสชั่น จะมีไอดีแค่ไอดีเดียวไม่เหมือนกัน ผู้ใช้ที่ถูกเชิญเข้าร่วมสนทนาจะมีเซสชั่นไอดีที่เหมือนกัน ส่วนที่สองจะเป็นสวิตซ์บอร์ดและพอร์ตส่วนที่สามจะเป็นประเภทของการเข้าใช้งาน ซึ่งควรจะเป็นซีเคไอ (CKI) ส่วนที่สี่จะเป็นอักขระที่จำเป็นในการเข้าร่วมสวิตซ์บอร์ด ส่วนที่ห้าและหกเป็น ชื่อผู้เข้าใช้และชื่อดิสเพลย์ของผู้ใช้งานตามลำดับ

การทำงานในส่วนของสวิตซ์บอร์ด จะเป็นการเชื่อมต่อทีซีพีกับเซิร์ฟเวอร์ซึ่งจะใช้เวลาประมาณสองนาทีในการรับ คำสั่งอาเอ็นจี (RNG) โดยที่ผู้ใช้งานจะส่งคำสั่งเพื่อเชื่อมต่อ (ANS) ไปที่เซิร์ฟเวอร์ก่อน โดยจะประกอบด้วย "TRID account name authentication string และ switchboard session ID"

ถ้าหากว่าสามารถเชื่อมต่อได้แล้ว เซิร์ฟเวอร์จะส่งคำตอบกลับมาเพื่อยืนยันว่าสามารถติดต่อกันได้แล้ว

2.1.6.4 การส่งไฟล์ (File transfer)

โดยโปรโตคอลที่จะไว้ส่งไฟล์มีชื่อว่า "MSNFTP" จัดเก็บข้อมูลต่างๆ ของไคลแอนโดยโปรโตคอลนี้ไม่เกี่ยวข้องกับ "FTP" โดยจะมีการทำ "Invitation stage" เริ่มต้นผ่านสวิตซ์บอร์ดเซสชั่นด้วย "IME-version 1.0" ซึ่งการเริ่มต้น ซึ่งใน "Invitation stage" จะมีการทำงานของคอมพิวเตอร์สองโดยมีการตกลงกันของที่อยู่ไอพีและพอร์ตที่ไคล์แอนท์จะทำการเชื่อมต่อ โดยที่ฝั่งเซิร์ฟเวอร์จะรับรู้ถึงพอร์ตของฝั่งไคลแอนทั้งสอง ซึ่งฝั่งไคลแอนจะพยายามเชื่อมต่อกับที่อยู่ไอพีและพอร์ตถ้าหากว่าการเชื่อมต่อได้ถูกสร้างเสร็จเมื่อไหร่ คอมพิวเตอร์ทั้งสองฝั่งถึงจะหยุดการพยายามเชื่อมต่อกับอีกพอร์ต

2.2 งานวิจัยที่เกี่ยวข้อง

งานวิจัยที่เกี่ยวข้องนี้เราจะนำงานวิจัยกี่ยวกับไวรัสที่ทางไทยเซอร์ด (ThaiCERT) ได้นำมาเผยแพร่ โดยที่งานวิจัยนี้จะเป็นงานวิจัยที่ดูพฤติกรรมของหนอนว่า หนอนนั้นได้ไปทำอะไรบ้างเมื่อมันทำการติดเชื้อแล้ว โดยเราจะเอาหนอนที่เป็นสายพันธ์เก่าของหนอนที่เราวิจัยนี้มาศึกษาต่อว่าสายพันธ์ก่อนที่มันจะพัฒนานั้นเป็นอย่างไร

ที่เราเลือกไวรัส ชนิดแบบหนอนมาอธิบายเพราะ ไวรัสประเภทหนอนนั้นจะเข้ามาทางเครือข่าย โดยการที่เราต้องกดยอบรับให้มันเข้ามาทำความเสียหายกับเครื่องของเรา อาจจะเป็นในลักษณะทาง อีเมลล์ หรือ ในรูปแบบต่างๆที่หลอกให้เรากดยอมรับมัน

- **W32.MSN.Worm**

เป็นหนอนที่แพร่กระจายตัวเองผ่านโปรแกรมสนทนาเอ็มเอสเอ็นแมสเซ็นเจอร์ด้วยไฟล์ที่มีชื่อว่า "Image.zip" เมื่อหนอนชนิดนี้คุกคามภายในเครื่องคอมพิวเตอร์แล้ว หนอนจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สร้างไฟล์ %windir%winlog32.exe และจะพยายามส่งตัวเองไปยังบัญชีรายชื่ออื่นๆ ที่อยู่ในลิสต์ด้วย

หนอนชนิดนี้สามารถหยุดการทำงานของเซอร์วิส "Security Center" และ "winvnc4"

ลักษณะที่หนอนใช้ส่งจะประกอบไปด้วยข้อความต่างๆ แล้วตามด้วยไฟล์

- LOL, you look so ugly in this picture, no joke...
- Should I put this on facebook/myspace?
- Hey m8, who is this on the right, in this picture...
- Sup, seen the pictures from the other night?

รูปที่ 2.1.10 ตัวอย่างลักษณะการส่งของหนอนชนิดนี้

1) วิธีการแพร่กระจาย

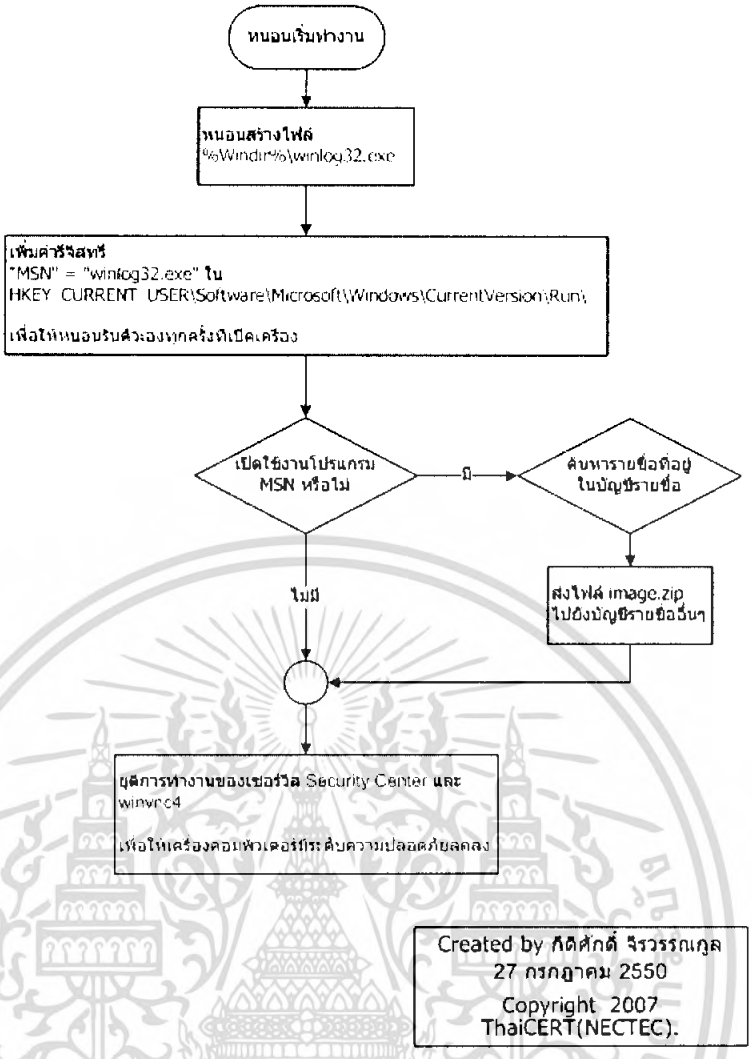
หนอนชนิดนี้สามารถแพร่กระจายผ่านทางโปรแกรมสนทนาเอ็มเอสเอ็น

2) ผลกระทบที่เกิดขึ้น

- **เครื่องอาจทำงานผิดพลาด** : เนื่องจากหนอนชนิดนี้ทำการแก้ไขค่าในรีจิสทรีสร้างไฟล์ขึ้นมา รวมทั้งมีการยุติการทำงานบางเซอร์วิสของระบบปฏิบัติการด้วย
- **เปิดการเชื่อมต่อที่ผิดปกติ** : หนอนชนิดนี้จะส่งไฟล์ของหนอนไปยังบัญชีรายชื่ออื่นๆ ที่อยู่ในลิสต์ของโปรแกรมสนทนาเอ็มเอสเอ็นแมสเซ็นเจอร์

3) การดำเนินการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1.11 วิธีการทำงานของหนอน W32.MSN.Worm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
โครงสร้างการทดลองและการออกแบบโปรแกรม

เนื่องจากผู้ทดลองได้ทำการเลือกโปรแกรมสนอร์ทเข้ามาช่วยทำการทดลองเกี่ยวกับการจับข้อมูลแพ็คเกตต่างๆที่อยู่บนเครือข่าย เพื่อที่จะสามารถดูพฤติกรรมต่างๆได้ โดยผู้ทดลองได้เน้นการสังเกตแพ็คเกตที่เป็นสิ่งที่ไม่ดี ผู้ทดลองจึงได้ทำการออกแบบโครงสร้างของการทดลองขึ้นมาก่อน จากนั้นก็จะทำการดูว่าจะเอาลักษณะแพ็คเกตส่วนไหนมาทำการดูว่าส่วนนั้นเป็นตัวบ่งบอกว่าเป็นแพ็คเกตที่ไม่หวังดี และผู้ทดลองเห็นว่าโปรแกรมสนอร์ทยังมีส่วนที่ยุ่งยากต่อการสร้างกฎ โดยวิธีการสร้างกฎผู้ใช้ได้กล่าวไว้ในบทที่ 2 จึงได้พัฒนาส่วนตรงนี้ด้วยโดยจะออกแบบโปรแกรมขึ้นมาเน้นความสะดวกสบาย ดังนั้นส่วนนี้จึงอธิบายถึงโครงสร้างของการทดสอบแพ็คเกตที่อยู่บนเครือข่าย และโครงสร้างของการพัฒนาโปรแกรม

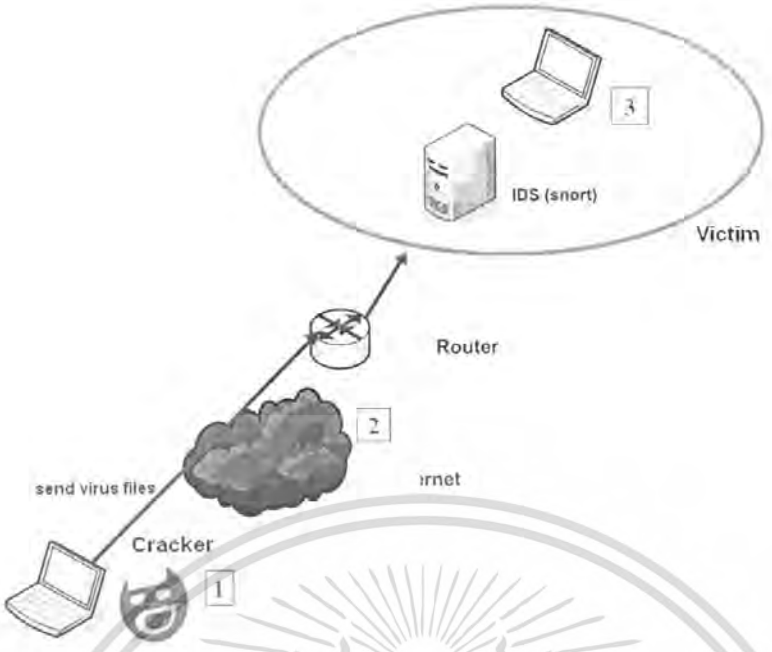
3.1 โครงสร้างการทดลอง

โครงสร้างการทดลองจะเป็นโครงสร้างที่ง่ายต่อการศึกษาแพ็คเกตที่ผ่านอยู่บนระบบเครือข่าย เราจึงสร้างระบบขึ้นมา โดยภายในระบบนั้นจะประกอบด้วย

- เครื่องคอมพิวเตอร์สองเครื่อง
- โปรแกรมสนอร์ทที่ติดตั้งบนเครื่องผู้ที่ถูกโจมตี
- ระบบปฏิบัติการ ลินุกส์

การติดตั้งนั้น เริ่มแรกผู้ทดลองได้ให้เครื่องคอมพิวเตอร์เครื่องหนึ่งเป็นเครื่องโจมตี (จากรูปที่ 3.1.1 เบอร์ 1) โดยผู้ทดลองจะทำให้เครื่องโจมตีทำการติดเชื้อก่อน โดยจะติดเชื้อไวรัสประเภทหนอน จากนั้นก็ทำการลงโปรแกรมสนอร์ทบนเครื่องผู้ถูกโจมตี (จากรูปที่ 3.1.2 เบอร์ 3) โดยผ่านระบบปฏิบัติการ ลิกนุกส์ โดยการทำการทดลองนี้จะต้องทำการผ่านระบบเครือข่าย (จากรูปที่ 3.1.1 เบอร์ 2) โดยสามารถดูโครงสร้างการทดลองโดยรวมได้จากรูปที่ 3.1.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1.1 โครงสร้างของระบบการทดลองแบบเปิด

โดยขั้นตอนการทดลองจะเป็นดังนี้

3.1.1 เริ่มแรกผู้ใช้จะทำการทดลองทดสอบ กฎของสนอร์ทว่าสามารถที่จะป้องกันสิ่งแปลกปลอมที่ไม่ต้องการให้เข้ามาในเครื่องของผู้ถูกโจมตีได้หรือไม่

3.1.2 ผู้ทดลองได้ลองสร้าง กฎของสนอร์ทในการป้องกันการปิง (Ping) มายังเครื่องผู้ถูกโจมตี เพื่อทดสอบกฎของสนอร์ท

3.1.3 ทำการตรวจสอบการทำงานของหนอนบนเครื่องผู้โจมตี (เบอร์ 1) โดยจะดูการทำงานที่เพิ่มเข้ามาหลังจากทำการติดเชื้อ

3.1.4 ดูแพ็คแกจที่ส่งเข้ามายังเครื่องผู้ถูกโจมตีผ่านโปรแกรมสนอร์ (เบอร์ 3)

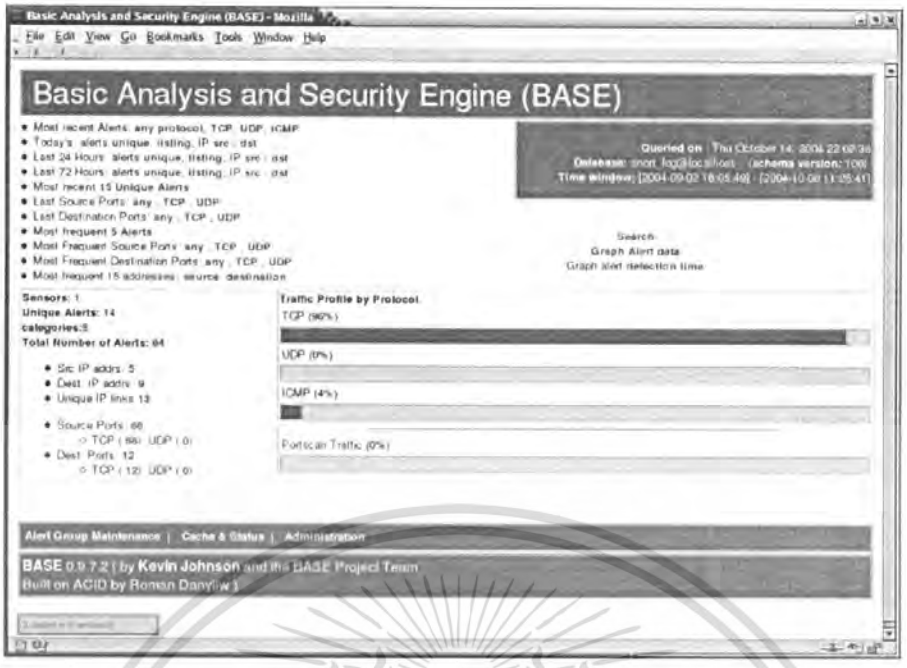
3.1.5 วิเคราะห์อักขระจากแพ็คเกตที่บ่งบอกว่าแพ็คเกตนั้นเป็นสิ่งที่หนอนได้ทำการส่งด้วยตัวหนอนเอง

3.1.6 นำอักขระส่วนนั้นมาทำการสร้างกฎ เพื่อป้องกันไวรัสชนิดหนอน

3.2 รูปแบบเอาท์พุตของแพ็คเกตบนเครือข่าย

การแสดงผลของโปรแกรมจะใช้ร่วมกับเบสโดยเบสนี้จะมีหน้าที่ในการแสดงข้อมูลของแพ็คเกตต่างๆที่เข้ามาที่ตัวเครื่องผู้ถูกโจมตีและสามารถจัดการกับข้อมูลนั้นได้โดยตรง ซึ่งจะแสดงออกมาในรูปแบบของตารางข้อมูลต่างๆ จะแบ่งแยกออกเป็นข้อมูลที่เข้ามาทางแต่ละพอร์ต โดยสามารถเรียกดูผ่านเวป บราว์เซอร์ ดังรูปที่ 3.2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2.1 ลักษณะเอาท์พุตที่ใช้เบสเป็นตัวแสดง

โดยเมื่อเราได้ทำการเข้าไปยังแต่ละพอร์ต ข้อมูลจะขึ้นตรงกับดาต้าเบสที่เราเก็บไว้ โดยภายในประกอบไปด้วยไอพีที่อยู่ของผู้ส่ง ไอพีที่อยู่ของผู้รับ ความยาวของแพ็คเกต ชื่อของเซ็นเซอร์ เป็นต้น ในเบสนี้สามารถชี้เฉพาะไปแต่ละลักษณะแพ็คเกตได้ โดยเราสามารถที่จะเลือกได้ว่าสิ่งใดที่เราต้องการทำการตรวจสอบ จากรูปที่ 3.2.2 จะแสดงข้อมูลแต่ละแพ็คเกตที่เข้ามา โดยส่วนนี้จะเป็นส่วนที่สำคัญที่ผู้ทำการทดลองจะทำการแจ้งเตือนไปยังผู้ถูกโจมตีว่าหนอนชนิดไหนได้เข้ามา ข้อความที่จะแจ้งเตือนจะแสดงตรงส่วนที่อยู่ในวางสีแดงในรูปที่ 3.2.2

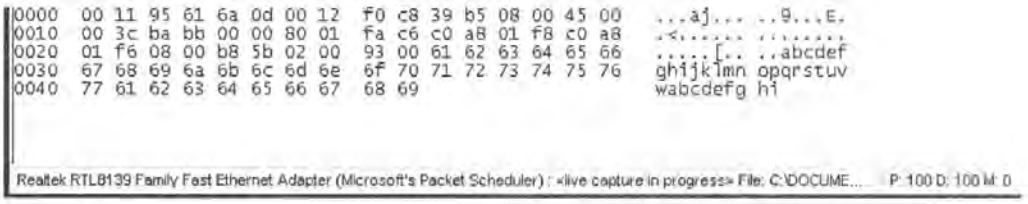
☐	#2016-(1-1057)	[local] [snort] in	2008-03-14 14:55:42	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2017-(1-1058)	[local] [snort] in	2008-03-14 14:55:42	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2018-(1-1059)	[local] [snort] in	2008-03-14 14:55:42	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2019-(1-1060)	[local] [snort] in	2008-03-14 14:55:42	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2020-(1-1061)	[local] [snort] in	2008-03-14 14:55:42	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2021-(1-1062)	[local] [snort] in	2008-03-14 14:55:42	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2022-(1-1063)	[local] [snort] in	2008-03-14 14:55:42	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2023-(1-1064)	[local] [snort] in	2008-03-14 14:55:42	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2024-(1-1065)	[local] [snort] in	2008-03-14 14:55:42	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2025-(1-1066)	[local] [snort] in	2008-03-14 14:55:42	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2026-(1-1067)	[local] [snort] in	2008-03-14 14:55:42	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2027-(1-1068)	[local] [snort] in	2008-03-14 14:55:42	202.43.32.2:80	10.204.0.10:32767	TCP

รูปที่ 3.2.2 ลักษณะการแจ้งเตือนของโปรแกรมสนอร์ท

3.3 รูปแบบของแพ็คเกตบนเครือข่าย

รูปแบบของแพ็คเกตบนเครือข่ายนั้นจะเป็นตัวแปรสำคัญที่สุดในการทดลองเนื่องจากการทดลองจะเป็นการทดลองการป้องกันไวรัสชนิดหนอน โดยรูปแบบแพ็คเกตต่างๆ ภายในจะมีอักขระของแต่ละแพ็คเกตอยู่ ดังรูปที่ 3.3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3.1 รูปแบบแพ็คเกจ

โดยหนอนแต่ละประเภทก็จะมีอักขระที่อยู่ในแพ็คเกตเป็นของตัวหนอนชนิดนั้นเอง ผู้ทดลองจะเอาอักขระส่วนนี้มาทำการเขียนกฎดังรูปที่ 3.3.2

alert tcp any any -> any any (content:"abc"; msg:"virus";)

รูปที่ 3.3.2 ส่วนของอักขระที่เพิ่มลงไปในกฎ

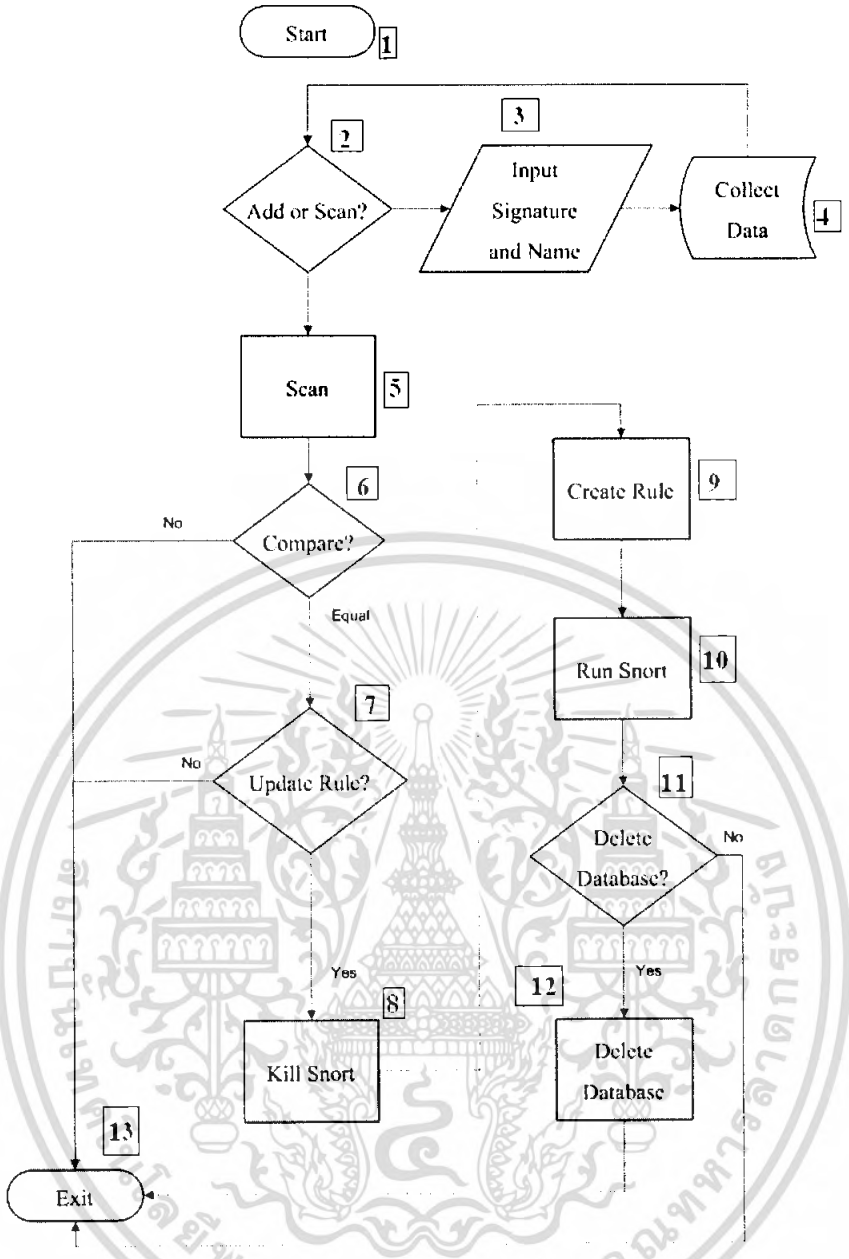
โดยจากรูปที่ 3.3.2 ส่วนที่อยู่ในวงสีแดงจะเป็นส่วนที่นำอักขระที่เป็นตัวบ่งบอกว่าเป็นหนอนชนิดไหนมาใส่ลงในกฎ

3.4 การออกแบบโปรแกรม

กลุ่มผู้ทดลองได้ออกแบบโปรแกรมเพื่อพัฒนาส่วนของการสร้างกฎ เพื่อให้ความสะดวกสบายแก่ผู้ใช้ โดยผู้ใช้เขียนเพียงแค่ใส่อักขระที่บ่งบอกว่าเป็นหนอนชนิดไหนและชื่อหนอนชนิดนั้น โปรแกรมก็จะทำการเก็บข้อมูลของหนอนชนิดต่างๆไว้ เมื่อผู้ใช้ต้องการตรวจสอบข้อมูลแพ็คเกตที่เข้ามาทางเครือข่าย ถ้าเจออักขระที่ตรงกับข้อมูลของหนอนก็จะทำการสร้างกฎการป้องกันหนอนชนิดนั้นขึ้นมา และโปรแกรมจะนำกฎที่กลุ่มผู้ทดลองสร้างไปเก็บในโฟลเดอร์รู (Rule) เอง

เนื่องจากโปรแกรมสนอร์มีกฎภายในตัวโปรแกรมเอง เมื่อได้ทำการเปิดสนอร์ทขึ้นมา สนอร์ทจะรู้จักแค่กฎที่มีก่อนทำการเปิดโปรแกรมขึ้นมาเท่านั้น ดังนั้นเมื่อต้องการสร้างกฎเพิ่มจะต้องทำการปิดโปรแกรมสนอร์ทและทำการเปิดใหม่ก่อน สนอร์ทถึงจะรู้จักกฎใหม่ที่ได้ทำการเพิ่มลงไป กลุ่มผู้ทดลองจึงนำส่วนตรงนี้ไปพัฒนาในโปรแกรมด้วย กลุ่มผู้ทดลองจึงได้ออกแบบโปรแกรมที่ช่วยพัฒนาส่วนของการสร้างกฎ โดยโครงสร้างของโปรแกรมจะเป็นดังรูปที่ 3.4.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4.1 โครงสร้างการทำงานของโปรแกรม

จากแผนภาพของโปรแกรมการทำงาน เบอร์ที่แสดงแต่ละส่วนจะทำงาน ดังนี้

1) เริ่มแรกจะเป็นการเปิดโปรแกรมขึ้นมา
2) ให้ผู้ใช้เลือกว่าต้องการที่จะทำการเพิ่มอักขระของหนอนและชื่อหนอน หรือว่าต้องการที่จะทำการตรวจสอบแพ็คเกตที่เข้ามาว่ามีแพ็คเกตที่หนอนส่งมาหรือไม่
3) เมื่อผู้ใช้เลือกที่จะเพิ่ม โปรแกรมจะให้ผู้ใช้ทำการเพิ่มอักขระของหนอนและชื่อหนอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) เมื่อโปรแกรมทำการเพิ่มเสร็จแล้ว ระบบจะเก็บข้อมูลลงในไฟล์ และจะกลับไปที่เบอร์ 2
5) เมื่อผู้ใช้เลือกที่จะตรวจสอบแพ็คเกต ระบบจะทำการค้นหาแพ็คเกตที่หนอนส่งมา
6) ทำการเปรียบเทียบอักขระของหนอนที่เก็บอยู่ในคลังข้อมูลว่าแพ็คเกตที่เข้ามานั้น อักขระตรงกับในไฟล์หรือไม่ ถ้าตรงก็จะไปทำการแสดงว่าเจอหนอนชนิดใด ถ้าไม่เจอโปรแกรมก็จะแจ้งเตือนว่าแพ็คเกตที่เข้ามาทั้งหมดนั้นปลอดภัย และออกจากโปรแกรม
7) เมื่อทำการเปรียบเทียบแล้วตรงกัน โปรแกรมก็จะแสดงชนิดของหนอนที่อักขระตรงกับในไฟล์ออกมา แล้วให้ผู้ใช้ทำการเลือกว่าต้องการอัพเดตกฎของหนอนชนิดไหน ถ้าผู้ใช่ไม่ต้องการก็จะทำออกจากโปรแกรมทันที
8) เมื่อผู้ใช้ได้ทำการเลือกชนิดของหนอนเรียบร้อยแล้ว โปรแกรมจะเริ่มทำการเพิ่มกฎลงไปในสนอร์ท โดยจะต้องทำการปิดโปรแกรมสนอร์ทก่อน ถึงจะสร้างกฎใหม่ให้สนอร์ทรู้จักได้
9) ทำการสร้างกฎแล้วนำไปเก็บในโฟเดอร์รู (Rule) และทำการเพิ่มข้อความบางส่วนใน Snort.conf
10) หลังจากที่สร้างกฎเสร็จแล้วจะทำการรันโปรแกรมสนอร์ทขึ้นมา
11) เมื่อทำการสร้างกฎเสร็จแล้ว จะทำการถามผู้ใช้ว่าต้องการลบข้อมูลแพ็คเกตเก่าที่อยู่ภายในฐานข้อมูลหรือไม่ ถ้าผู้ใช้ต้องการก็จะทำการลบ ถ้าไม่ต้องการระบบจะทำการออกจากโปรแกรม
12) หลังจากที่ผู้ใช้ต้องการลบข้อมูลแพ็คเกตเก่า ระบบก็จะทำการลบข้อมูลในคลังข้อมูล
13) เมื่อโปรแกรมทำงานเสร็จก็จะออกจากโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการทดลองระบบและการสร้างโปรแกรม

บทนี้จะเป็นส่วนของผลการทดลองของระบบ รูปแบบแพ็คเกตของหนอนที่ส่งมา และการนำอักขระในแพ็คเกตไปสร้างเป็นกฎของโปรแกรมสนอร์ท หลังจากนั้นจะแสดงรูปแบบอินเตอร์เฟสของโปรแกรมที่กลุ่มผู้ทดลองได้สร้างขึ้น

4.1 ผลการทดลองระบบ

ในส่วนนี้จะกล่าวถึงการทดสอบโปรแกรมสนอร์ท การสร้างกฎการปิง (ping) การดูพฤติกรรมของหนอนเอ็มเอสเอ็น การตรวจสอบแพ็คเกตของหนอนและการนำอักขระในแพ็คเกตจากหนอนเอ็มเอสเอ็นมาสร้างเป็นกฎ

4.1.1 การทดสอบโปรแกมสนอร์ท

โดยเริ่มแรกของการใช้โปรแกรม เมื่อปรับแต่งค่าที่ใช้ตั้งค่าในโปรแกรมสนอร์ทแล้วก็จะทำการใช้โปรแกรมและทำการเชื่อมต่อทางอื่นผ่านทางอินเตอร์เน็ต เพื่อดูว่ามีแพ็คเกตเข้าสู่ลิปพีแคปไลบรารีและหลังจากนั้นสนอร์ททำการตรวจสอบแพ็คเกตหรือไม่ ซึ่งจะใช้คำสั่งการใช้งานโปรแกรมดังรูปที่ 4.1.1

รูปที่ 4.1.1.1 ลักษณะหน้าจอที่ใช้รูปแบบคำสั่งทดสอบการทำงานของสนอร์ท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และจะมีการแสดงผลการตรวจสอบการใช้โปรแกรมสนอร์ทดังภาพด้านล่าง

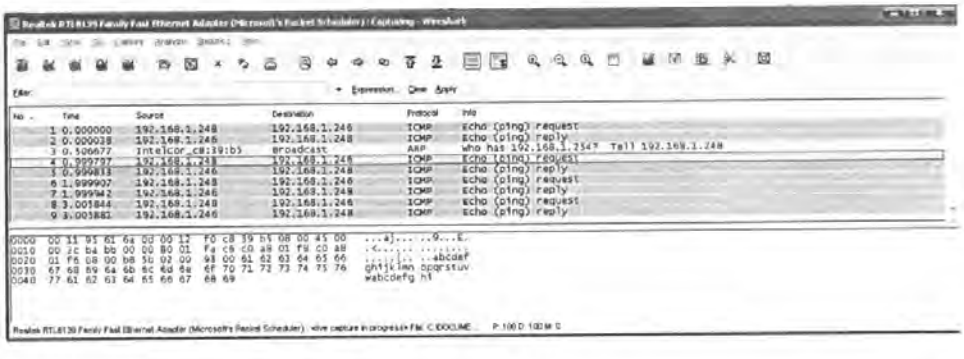
รูปที่ 4.1.1.2 ลักษณะข้อมูลที่เข้าภายในเครื่อง

4.1.2 กฎปิงที่สร้างขึ้นและการทดลองใช้

ขั้นต่อไปของการทดลอง เราจะทำการเขียนกฎของสนอร์ทขึ้นมาเอง และทำการทดสอบโดยการใช้คอมพิวเตอร์อีกเครื่องหนึ่งทำการปิง (Ping) ข้อมูลส่งเข้าเครื่องที่ติดตั้งโปรแกรมสนอร์ท และปิดกฎของสนอร์ททุกกฎยกเว้นกฎที่ผู้ใช้งานเขียนขึ้นมาเอง

ซึ่งการเขียนกฎของสนอร์ทเองนั้น ทำการทดลองเขียนกฎได้โดยจะใช้โปรแกรมไวลชาร์ค (WireShark) เข้ามาช่วย โดยจะติดตั้งที่เครื่องคอมพิวเตอร์ที่มีโปรแกรมสนอร์ท เพื่อตรวจสอบแพ็คเกตทั้งหมดที่เกิดขึ้นในเครือข่าย

เมื่อติดตั้งแล้วจะทำการปิงจากอีกเครื่องหนึ่งเข้าสู่เครื่องที่ติดตั้งโปรแกรม และใช้โปรแกรมไวลชาร์คอ่านข้อมูลของการปิงนั้น เพื่อดูอักขระในแพ็คเกตดังรูปที่ 4.1.2.1



รูปที่ 4.1.2.1 ภาพการใช้โปรแกรมไวลชาร์ค และอักขระด้านล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และเราจะนำอักขระบางส่วนจากทั้งหมดมาเพื่อเขียนเป็นกฎ แล้วนำไปใส่ในไฟล์กฎ ([pathsnort]\rules)

```
0000   00 11 95 61 6a 0d 00 12   f0 c8 39 b5 08 00 45 00   ...aj... ..9...E.
0010   00 3c ba bb 00 00 80 01   fa c6 c0 a8 01 f8 c0 a8   .<...... ........
0020   01 f6 08 00 b8 5b 02 00   93 00 61 62 63 64 65 66   .....[.. ..abcdef
0030   67 68 69 6a 6b 6c 6d 6e   6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040   77 61 62 63 64 65 66 67   68 69                    wabcdefg hi
```

รูปที่ 4.1.2.2 ภาพอักขระของการปิงแพ็คเกต

```
# Sourcefire, Inc. All Rights Reserved.  All other GPL Rules are owned and
# copyrighted by their respective owners, including principal contributors
# Martin Roesch and Brian Caswell. Please see www.snort.org/contributors for
# a list of owners and their respective copyrights.  In order to determine
# what rules are VRT Certified Rules or GPL Rules, please refer to the VRT
# Certified Rules License Agreement.
#
# $Id: scan.rules,v 1.29.2.3 2005/03/10 21:11:39 wwatchinski Exp $
#------------
# SCAN RULES
#------------
# These signatures are representitive of network scanners.  These include
# port scanning, ip mapping, and various application scanners.
#
# NOTE: This does NOT include web scanners such as whisker.  Those are
# in web"
#

#alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"Ping"; content:"|61 62 63 64|"; classtype:policy-violation; sid:12345678;)

#pass icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"TT"; dsize:128; content:"abcdefghijklmn opqrstuvw abcdefg hi"; classtype:icmp-event; sid:1111111; rev:1;)
#log tcp $EXTERNAL_NET any -> any &lt;1023 (msg:"chaying" ; uricontent:"www.chay.com"; sid:1255456;)
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"Ping Ping!!!"; content:"|00 11 95|"; classtype:icmp-event; sid:1111112; rev:2;)
```

รูปที่ 4.1.2.3 ภาพของกฎที่ผู้ใช้ทำการเขียนขึ้น โดยจะใช้บรรทัดสีฟ้าเท่านั้น

```
content:"|00 11 95|";
```

รูปที่ 4.1.2.4 ภาพอักขระบางส่วนจากการปิงข้อมูล ที่นำมาใส่ในกฎการใช้โปรแกรมสนอร์ท

เสร็จแล้วจะทำการปิดกฎทั้งหมดที่สนอร์ทใช้ที่ไฟล์การตั้งค่าของโปรแกรมสนอร์ท (snort.conf) แล้วทำการทดลองใช้โปรแกรมสนอร์ท โดยให้อีกเครื่องหนึ่งทำการปิงแพ็คเกตเข้าสู่เครื่องที่ติดตั้ง เพื่อดูผลว่ากฎสามารถใช้ได้หรือไม่

โดยวิธีการที่กล่าวมาจะเป็นวิธีเริ่มต้นในการทดสอบโปรแกรมสนอร์ทเท่านั้น เราสามารถที่จะทำการเก็บได้ข้อมูลได้หลายรูปแบบและสามารถแจ้งเตือนได้ทันทีออกมาในลักษณะข้อความ

4.1.3 การทำงานของไวรัสเอ็มเอสเอ็นแมสเซ็นเจอร์ (MSN Messenger)

หนอนชนิดนี้จะเป็นสายพันธ์ที่พัฒนามาจากสายพันธ์ W32.Msn.Worm โดยหนอนชนิดนี้จะใช้ควบคู่กับโปรแกรมเอ็มเอสเอ็นแมสเซ็นเจอร์ในการแพร่เชื้อ เริ่มแรกเมื่อติดเชื้อมันจะทำการตรวจสอบรายชื่อในลิสล์ของ โปรแกรมเอ็มเอสเอ็นแมสเซ็นเจอร์ของเรา แล้วหลังจากนั้นก็หนอนจะทำการจัดส่งสิ่งที่ไม่ดีโดยจะคอยส่งไปยังรายชื่อผู้ที่อยู่ในลิสล์รายชื่อ ตัวหนอนจะทำการหลอกผู้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่เป็นเหยื่อให้กดรับข้อมูล โดยหนอนจะทำเป็นคุยกับผู้รับมันจะมีลักษณะคำพูดต่างๆออกมา ข้อความจะเป็นลักษณะเชิญชวน ดังนี้

1) OMG, I found ur pic on cuteornot.com! im not kidding either!!!
2) tell me what you think of this. I Made it in photoshop. do you think its too green?
3) I just made this picture in photoshop It AWESOME for you Desktop.....

โดยเมื่อหนอนได้ทำการส่งข้อความแล้วจะมีไฟล์แนบมาด้วย โดยชื่อไฟล์ที่แนบมานั้นจะมีหลายชื่อโดยหนอนจะเป็นตัวแลนดอมชื่อเอง อย่างเช่น ชื่อ album หรือ photo แล้วตามด้วยหมายเลขต่างๆ โดยจะส่งมาในลักษณะซิปแล้วมันจะให้โหลดเข้ามาภายในเครื่องของผู้ที่รับ เมื่อผู้รับหลงเชื่อคิดว่ามีคนอยากส่งข้อมูลรูปมาให้มาให้ ก็จะทำการโหลด ทั้งที่ผู้ส่งนั้นไม่รู้มาก่อนว่าเครื่องตัวเองได้ทำการส่ง เพราะหนอนได้ทำส่งด้วยตัวมันเองโดยอัตโนมัติ

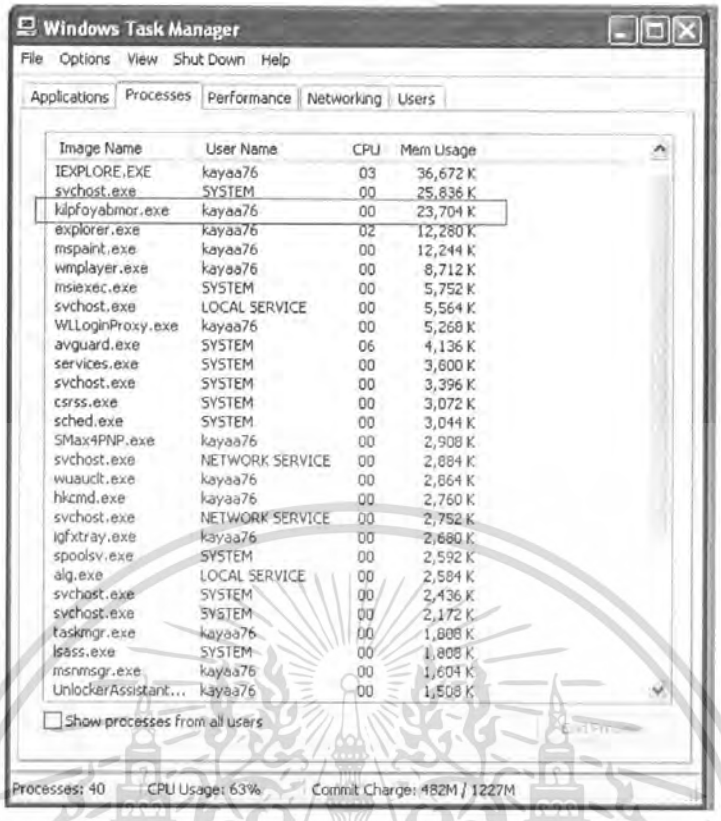
รูปที่ 4.1.3.1 ลักษณะไฟล์ที่ส่งมาชื่อ album_975.zip

เมื่อผู้ที่เหยื่อได้โหลดไฟล์นั้นแล้ว หนอนยังไม่เกิดการแพร่กระจาย หนอนจะแพร่กระจายต่อเมื่อเราไปทำการเปิดไฟล์ที่ชื่อ "album_975.zip" ออกมา จากนั้นผู้รับก็จะติดเชื้อและกลายเป็นผู้โจมตีแทน

- **พฤติกรรมการทำงานของไวรัสเอ็มเอสเอ็น**

เมื่อเราเกิดการติดเชื้อแล้วหนอนจะไปทำการลอกตัวมันเองไปเก็บไว้ใน System32 โดยจะใช้ชื่อที่ไม่ซ้ำกันในการติดแต่ละครั้ง หลังจากที่มันลอกตัวแล้วมันจะทำการเริ่มการทำงานของตัวเองโดยอัตโนมัติ โดยลักษณะจะเป็นดังรูปที่ 4.1.3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1.3.2 มอนิเตอร์ที่แสดงการทำงานของหนอน

โดยเราสามารถยืนยันได้ว่าส่วนที่อยู่ในกรอบสีแดงในมอนิเตอร์เป็นหนอนที่ทำงานด้วยตัวมันเอง ได้จากการสแกนไวรัสโดยใช้ แอนตี้ไวรัส เอวีจี (AVG) เป็นตัวแสดง ดังรูปที่ 4.1.2.3

รูปที่ 4.1.3.3 ตัวแอนตี้ไวรัสเอวีจีที่แสดงผลว่าคอมพิวเตอร์ได้มีการติดเชื้อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.1.3.3 จะเห็นว่าเป็นการแสดงว่าไวรัสที่รันก่อนหน้าเป็นหนอนชนิดหนึ่ง โดยมันจะมีอยู่ สองชื่อเนื่องจากได้ทำการติดเชื้อไวรัสจำนวนสองครั้ง ชื่อแต่ละครั้งจะทำการซุ่มด้วยตัวมันเองโดยอัตโนมัติ

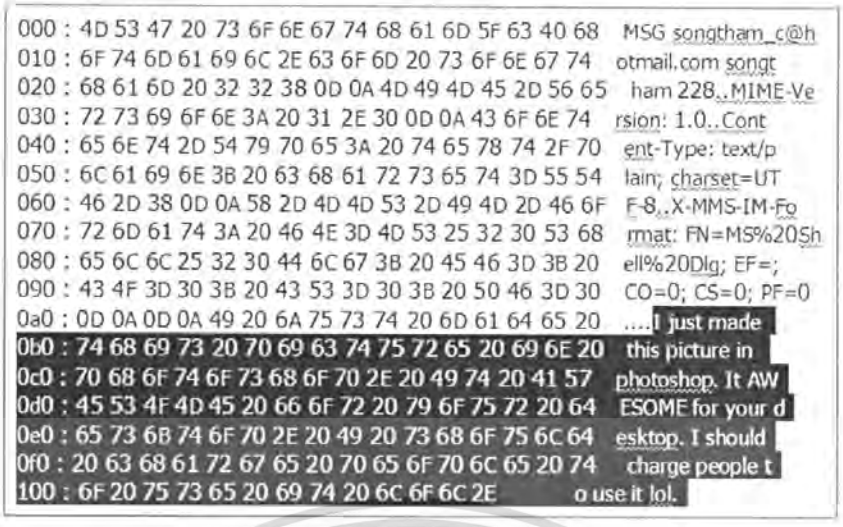
เราสามารถจะดูโครงสร้างของหนอนชนิดนี้ โดยใช้โปรแกรมสนอร์ทเข้ามาช่วยในการดูแพ็คเกตของมัน ว่าลักษณะเป็นเช่นไร เริ่มแรกเมื่อหนอนเกิดการแพร่เชื้อมันจะส่งอักขระที่เชิญชวนพร้อมกับส่งไฟล์แนบไปด้วยไปให้เหยื่อ เราจึงได้ใช้โปรแกรมสนอร์ทเป็นตัวจับแพ็คเกตตั้งแต่หนอนเริ่มทำงานดังรูป

รูปที่ 4.1.3.4 ส่วนที่หนอนเอ็มเอสเอ็มทำการเขียนอักขระและจัดส่งไฟล์

4.1.4 การตรวจสอบแพ็คเกตจากหนอน

หลังจากที่เราได้ใช้โปรแกรมสนอร์ทในการตรวจจับแพ็คเกตของหนอนเอ็มเอสเอ็นแล้ว มันจะมีแพ็คเกตส่งเข้ามาสองส่วน ซึ่งส่วนแรกก็คือส่วนของคำพูดที่หนอนใช้ในการเขียนเพื่อหลอกผู้รับ (ดังรูปที่ 4.1.4.1) และส่วนที่สองจะเป็นการส่งไฟล์ที่จะมีอักขระที่จะบ่งบอกว่า หนอนชนิดนี้เป็นคนจัดการส่งไฟล์นี้เองโดยอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

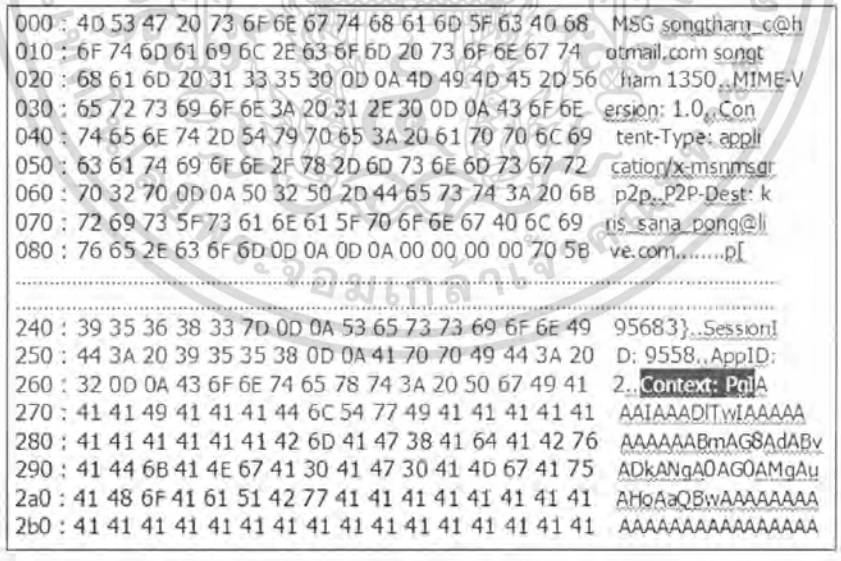


```
000 : 4D 53 47 20 73 6F 6E 67 74 68 61 6D 5F 63 40 68   MSG songtham_c@h
010 : 6F 74 6D 61 69 6C 2E 63 6F 6D 20 73 6F 6E 67 74   otmail.com songt
020 : 68 61 6D 20 32 32 38 0D 0A 4D 49 4D 45 2D 56 65    ham 228..MIME-Ve
030 : 72 73 69 6F 6E 3A 20 31 2E 30 0D 0A 43 6F 6E 74   rsion: 1.0..Cont
040 : 65 6E 74 2D 54 79 70 65 3A 20 74 65 78 74 2F 70   ent-Type: text/p
050 : 6C 61 69 6E 3B 20 63 68 61 72 73 65 74 3D 55 54   lain; charset=UT
060 : 46 2D 38 0D 0A 58 2D 4D 4D 53 2D 49 4D 2D 46 6F   F-8..X-MMS-IM-Fo
070 : 72 6D 61 74 3A 20 46 4E 3D 4D 53 25 32 30 53 68   rmat: FN=MS%20Sh
080 : 65 6C 6C 25 32 30 44 6C 67 3B 20 45 46 3D 3B 20   ell%20Dlg; EF=;
090 : 43 4F 3D 30 3B 20 43 53 3D 30 3B 20 50 46 3D 30   CO=0; CS=0; PF=0
0a0 : 0D 0A 0D 0A 49 20 6A 75 73 74 20 6D 61 64 65 20   ....I just made
0b0 : 74 68 69 73 20 70 69 63 74 75 72 65 20 69 6E 20   this picture in
0c0 : 70 68 6F 74 6F 73 68 6F 70 2E 20 49 74 20 41 57   photoshop. It AW
0d0 : 45 53 4F 4D 45 20 66 6F 72 20 79 6F 75 72 20 64   ESOME for your d
0e0 : 65 73 6B 74 6F 70 2E 20 49 20 73 68 6F 75 6C 64   esktop. I should
0f0 : 20 63 68 61 72 67 65 20 70 65 6F 70 6C 65 20 74    charge people t
100 : 6F 20 75 73 65 20 69 74 20 6C 6F 6C 2E            o use it lol.
```

รูปที่ 4.1.4.1 ส่วนแพ็คเกตของหนอนที่ใช้เป็นคำพูดในการหลอกเหยื่อ

4.1.5 การเปรียบเทียบแพ็คเกตที่หนอนส่งกับแพ็คเกตที่ผู้ใช้ส่งไปเอง

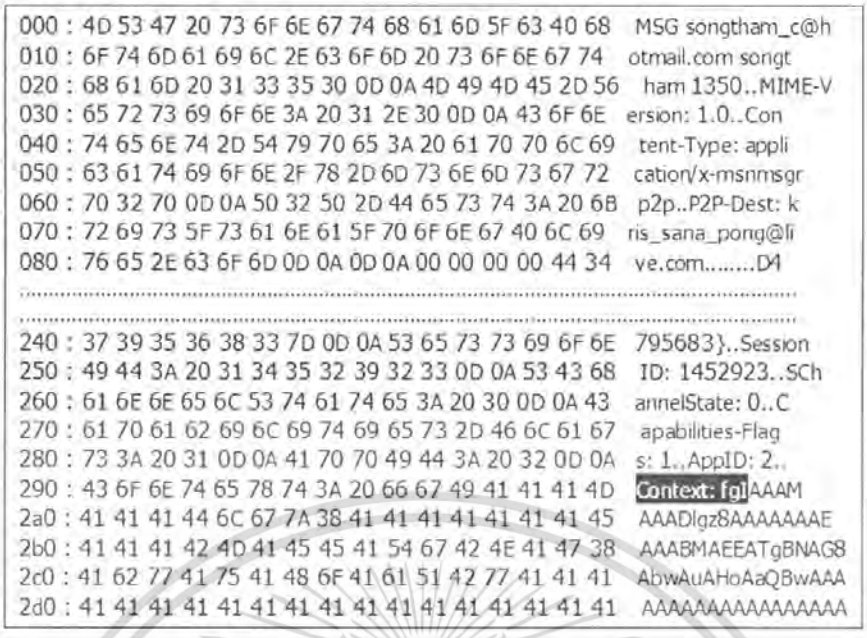
ผู้ทดลองได้ทำการเปรียบเทียบอักขระในแพ็คเกตที่เกี่ยวข้องกับการส่งไฟล์ โดยเริ่มแรกผู้ทดลองจะดูแพ็คเกตที่หนอนทำการส่งไฟล์ด้วยตัวมันเอง และหลังจากนั้นก็จะดูแพ็คเกตที่ผู้ใช้ทำการส่งไฟล์เอง แล้วนำข้อมูลสองส่วนนี้มาเรียบเทียบกัน ดังรูปที่ 4.1.5.1 และ 4.1.5.2



```
000 : 4D 53 47 20 73 6F 6E 67 74 68 61 6D 5F 63 40 68   MSG songtham_c@h
010 : 6F 74 6D 61 69 6C 2E 63 6F 6D 20 73 6F 6E 67 74   otmail.com songt
020 : 68 61 6D 20 31 33 35 30 0D 0A 4D 49 4D 45 2D 56    ham 1350..MIME-V
030 : 65 72 73 69 6F 6E 3A 20 31 2E 30 0D 0A 43 6F 6E   ersion: 1.0..Con
040 : 74 65 6E 74 2D 54 79 70 65 3A 20 61 70 70 6C 69   tent-Type: appli
050 : 63 61 74 69 6F 6E 2F 78 2D 6D 73 6E 6D 73 67 72   cation/x-msnmsgr
060 : 70 32 70 0D 0A 50 32 50 2D 44 65 73 74 3A 20 6B   p2p..P2P-Dest: k
070 : 72 69 73 5F 73 61 6E 61 5F 70 6F 6E 67 40 6C 69   ris_sana_pong@li
080 : 76 65 2E 63 6F 6D 0D 0A 0D 0A 00 00 00 00 70 5B   ve.com.........p[
.....
.....
240 : 39 35 36 38 33 7D 0D 0A 53 65 73 73 69 6F 6E 49   95683}..SessionI
250 : 44 3A 20 39 35 35 38 0D 0A 41 70 70 49 44 3A 20   D: 9558..AppID:
260 : 32 0D 0A 43 6F 6E 74 65 78 74 3A 20 50 67 49 41   2..Context: PgIA
270 : 41 41 49 41 41 41 44 6C 54 77 49 41 41 41 41 41   AAIAAADlTwIAAAAA
280 : 41 41 41 41 41 41 42 6D 41 47 38 41 64 41 42 76   AAAAAABmAG8AdABv
290 : 41 44 6B 41 4E 67 41 30 41 47 30 41 4D 67 41 75   ADkANgA0AG0AMgAu
2a0 : 41 48 6F 41 61 51 42 77 41 41 41 41 41 41 41 41   AHoAaQBwAAAAAAAA
2b0 : 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41   AAAAAAAAAAAAAAAA
```

รูปที่ 4.1.4.2 ส่วนแพ็คเกตของหนอนที่ใช้ส่งไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```
000 : 4D 53 47 20 73 6F 6E 67 74 68 61 6D 5F 63 40 68   MSG songtham_c@h
010 : 6F 74 6D 61 69 6C 2E 63 6F 6D 20 73 6F 6E 67 74   otmail.com songt
020 : 68 61 6D 20 31 33 35 30 0D 0A 4D 49 4D 45 2D 56   ham 1350..MIME-V
030 : 65 72 73 69 6F 6E 3A 20 31 2E 30 0D 0A 43 6F 6E   ersion: 1.0..Con
040 : 74 65 6E 74 2D 54 79 70 65 3A 20 61 70 70 6C 69   tent-Type: appli
050 : 63 61 74 69 6F 6E 2F 78 2D 6D 73 6E 6D 73 67 72   cation/x-msnmsgr
060 : 70 32 70 0D 0A 50 32 50 2D 44 65 73 74 3A 20 6B   p2p..P2P-Dest: k
070 : 72 69 73 5F 73 61 6E 61 5F 70 6F 6E 67 40 6C 69   ris_sana_pong@li
080 : 76 65 2E 63 6F 6D 0D 0A 0D 0A 00 00 00 00 44 34   ve.com.........D4
.....
.....
240 : 37 39 35 36 38 33 7D 0D 0A 53 65 73 73 69 6F 6E   795683}..Session
250 : 49 44 3A 20 31 34 35 32 39 32 33 0D 0A 53 43 68   ID: 1452923..SCh
260 : 61 6E 6E 65 6C 53 74 61 74 65 3A 20 30 0D 0A 43   annelState: 0..C
270 : 61 70 61 62 69 6C 69 74 69 65 73 2D 46 6C 61 67   apabilities-Flag
280 : 73 3A 20 31 0D 0A 41 70 70 49 44 3A 20 32 0D 0A   s: 1..AppID: 2..
290 : 43 6F 6E 74 65 78 74 3A 20 66 67 49 41 41 41 4D   Context: fgIAAAM
2a0 : 41 41 41 44 6C 67 7A 38 41 41 41 41 41 41 41 45   AAADlgz8AAAAAAAE
2b0 : 41 41 41 42 4D 41 45 45 41 54 67 42 4E 41 47 38   AAABMAEEATgBNAG8
2c0 : 41 62 77 41 75 41 48 6F 41 61 51 42 77 41 41 41   AbwAuAHoAaQBwAAA
2d0 : 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41   AAAAAAAAAAAAAAAA
```

รูปที่ 4.1.4.3 ส่วนแพ็คเกตของผู้ทำการส่งไฟล์เอง

จากรูปเมื่อเรานำแพ็คเกตที่ผู้ใช้เป็นคนส่งไฟล์เองกับ หนอนเป็นคนส่งเองโดยอัตโนมัติมาเปรียบเทียบกัน จะทำให้เรารู้ว่า อักขระบางส่วนจะเป็นตัวบ่งบอกว่าผู้ใช้หรือหนอนเป็นคนส่ง โดยถ้าอักขระในแพ็คเกตตรง "context:" เริ่มต้นด้วย "fgI" จะเป็นตัวแสดงว่าผู้ใช้ทำการส่งไฟล์เอง แต่ถ้าอักขระในแพ็คเกตตรง "context:" เริ่มต้นด้วย "PgI" แล้ว จะแสดงว่าหนอนได้ทำการส่งไฟล์นี้ด้วยตัวมันเองโดยอัตโนมัติ

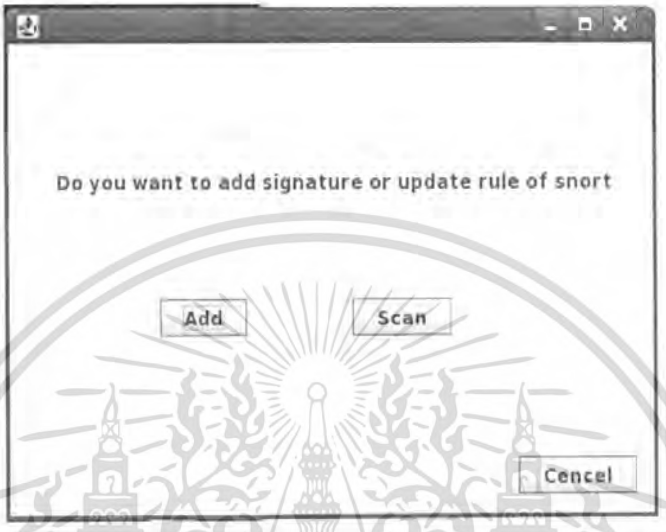
ซึ่งส่วนนี้จะเป็นประโยชน์ให้เราสามารถที่จะนำอักขระที่บ่งบอกว่าไฟล์นี้หนอนเป็นคนส่งมาเอง โดยสามารถที่จะไปเขียนกฎในสนอร์ทให้ทำการแจ้งเตือนแก่ผู้ใช้ได้ เมื่อมีหนอนที่ลักษณะนี้เข้ามาภายในระบบของผู้ที่เป็นเหยื่อ

จากการทำงานของไวรัสนี้ เราจะใช้โปรแกรมสนอร์ทในการตรวจจับแพ็คเกตลักษณะสิ่งที่ไม่ดี โดยภายในโปรแกรมสนอร์ทจะมีกฎต่างๆไว้ที่จะคอยกรองสิ่งที่ไม่ดีจากเครือข่าย โดยกฎนั้นเราสามารถที่จะสร้างขึ้นมาเองได้ โดยจะเขียนผ่านโน๊ตแพทแล้วนำไปเก็บไว้ในโฟลเดอร์รู (Rule) แต่ข้อเสียของโปรแกรมนี้ คือเมื่อเราได้ทำการรันโปรแกรมสนอร์ทตัวโปรแกรมจะรู้จักแค่รู (Rule) ที่อยู่ตอนนั้นเท่านั้น เมื่อเราสร้างกฎขึ้นมาใหม่ เราจะต้องทำการปิดการทำงานของสนอร์ทก่อนแล้วทำการรันโปรแกรมสนอร์ทใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

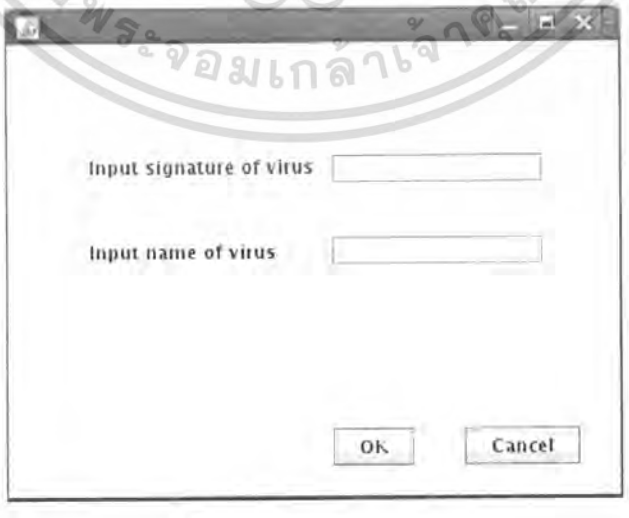
4.2 รูปแบบของโปรแกรม

กลุ่มผู้ใช้งานได้เขียนโปรแกรมขึ้นโดยใช้แอพพลิเคชั่นเน็ตบีนส์ (NetBeans) ภาษาจาวา ในการสร้างโปรแกรมสำหรับพัฒนาสนอร์ท ซึ่งสามารถแสดงอินเตอร์เฟสของโปรแกรมการใช้งานได้ดังนี้



รูปที่ 4.2.1 หน้าต่างในการเพิ่มหรือตรวจสอบกฎ

รูปที่ 4.2.1 เป็นอินเตอร์เฟสแสดงให้ผู้ใช้งานเลือกว่าจะต้องการเพิ่มอักขระของหนอนและชื่อหนอน หรือต้องการทำการตรวจสอบแพ็คเกตที่เข้ามาหรือไม่ โดยเริ่มแรกหากว่าไม่มีกฎในเครื่องนั้นจะต้องทำการกด Add เพิ่มกฎก่อน



รูปที่ 4.2.2 หน้าต่างในการเพิ่ม signature และชื่อของหนอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.2.2 เป็นอินเตอร์เฟสแสดงให้ผู้ใช้งานเลือกว่าต้องการจะเพิ่มข้อมูลของหนอนที่มีอักขระ (Signature) และชื่อของหนอนอย่างไร โดยกรอกในช่องว่าง ซึ่งข้อมูลเหล่านี้จะไปเก็บไว้ในไฟล์ข้อมูล เมื่อต้องการตรวจสอบแพ็คเกตที่เข้ามาก็จะดึงข้อมูลจากไฟล์นี้มาตรวจสอบ

เมื่อทำการเพิ่มข้อมูลของไวรัสแต่ละประเภทเข้าไปเก็บไว้ในไฟล์ข้อมูลแล้ว ผู้ใช้ก็สามารถทำการตรวจสอบแพ็คเกตที่เข้ามาภายในเครือข่ายได้ โดยเมื่อผู้ใช้ต้องการจะตรวจสอบ จะไปทำการกด Scan รูปที่ 4.2.3

รูปที่ 4.2.3 รูปแบบอินเตอร์เฟสตรงสแกนไวรัส

หลังจากนั้นเมื่อผู้ใช้ทำการสแกน แล้วถ้าเจอสิ่งที่คล้ายว่าจะเป็นหนอนระบบจะทำการแจ้งเตือนว่าได้พบลักษณะคล้ายหนอนชนิดไหน โดยจะแสดงหนอนแต่ละประเภทที่เจอแล้วให้ผู้ใช้ทำการเลือกว่าจะทำการอัพเดตกฎของหนอนชนิดส่วนไหน ดังรูปที่ 4.2.4

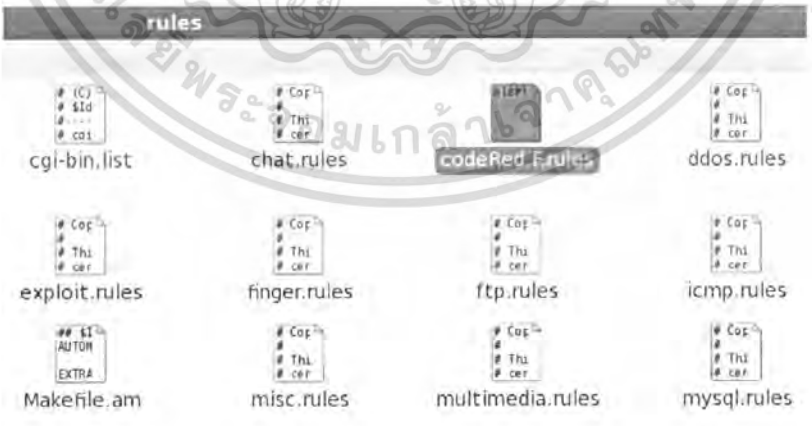
รูปที่ 4.2.4 รูปแบบอินเตอร์เฟสตรงอัพเดต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าผู้ใช้ต้องการอัพเดต โปรแกรมจะทำการเขียนกฎขึ้นมาแล้วไปเก็บในโฟลเดอร์รู (Rule) และเพิ่มการเรียกกฎเข้ามาใช้ลงในไฟล์ Snort.conf ดังรูปที่ 4.2.5

รูปที่ 4.2.5 หน้าต่างแสดงการเพิ่มการเรียกกฎในไฟล์ snort.conf

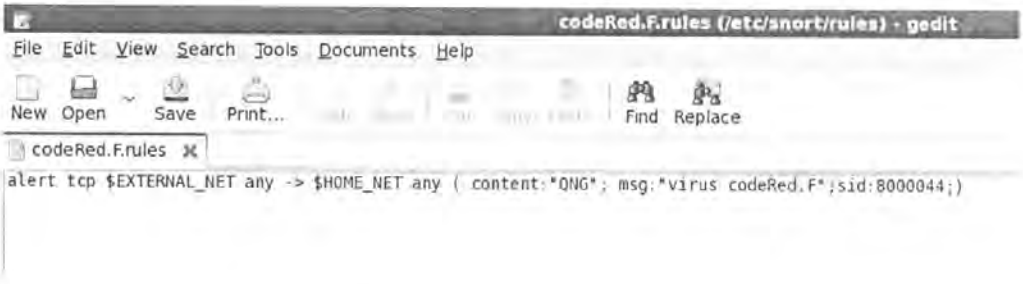
ส่วนไฟล์ที่สร้างขึ้นจะไปถูกเก็บอยู่ที่โฟลเดอร์รู (Rules) เพื่อรอการเรียกใช้งานจากไฟล์ snort.conf ดังจะเห็นได้จากรูปที่ 4.2.4



รูปที่ 4.2.6 หน้าต่างแสดงกฎที่ถูกเพิ่มในโฟลเดอร์รู (Rules)

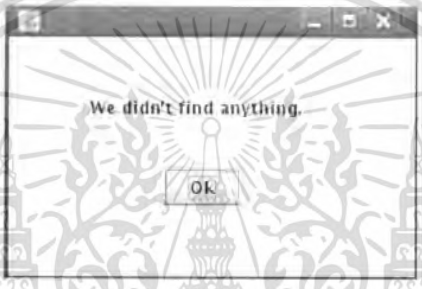
รูปที่ 4.2.6 แสดงภายในไฟล์ของกฎที่สร้างขึ้น โดยจะแสดงเป็นคำสั่งให้เตือนผู้ใช้ ถ้าหากมีอักขระที่ตรงกับคำว่า "QNG" โดยจะให้แสดงเมสเสจว่า "virus condRed.F"

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่ควรนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2.7 หน้าต่างแสดงการสร้างกฎของหนอน

แต่ถ้ากดสแกนแล้วไม่เจออักขระที่คล้ายว่าจะเป็นไวรัสจะทำการแจ้งว่าไม่เจอ ดังรูปที่ 4.2.6



รูปที่ 4.2.8 รูปแบบอินเตอร์เฟสตรงเมื่อไม่เจอไวรัส

หลังจากที่ผู้ใช้ได้ทำการกดอัพเดตกฎหรือไม่อัพเดตแล้ว โปรแกรมจะถามผู้ใช้ว่าต้องการลบข้อมูลเก่าที่อยู่ภายในดาต้าเบสทิ้งหรือไม่ดังรูปที่ 4.2.7

รูปที่ 4.2.9 รูปแบบอินเตอร์เฟสตรงลบข้อมูลในดาต้าเบส

ซึ่งกฎที่สร้างขึ้นนี้ เมื่อมีแพ็คเกตที่ตรงกับกฎดังกล่าวผ่านเข้ามาในระบบของผู้ใช้งาน โปรแกรมสนอร์ทจะทำการแจ้งเตือนผู้ใช้งานว่ามีแพ็คเกตที่มีอักขระตรงกันดังภาพที่ 4.2.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

☐	#2016-(1-1057)	[local] [snort] in	2008-03-14 14:55:42	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2017-(1-1058)	[local] [snort] in	2008-03-14 14:55:42	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2018-(1-1059)	[local] [snort] in	2008-03-14 14:55:42	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2019-(1-1060)	[local] [snort] in	2008-03-14 14:55:42	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2020-(1-1061)	[local] [snort] in	2008-03-14 14:55:42	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2021-(1-1062)	[local] [snort] in	2008-03-14 14:55:42	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2022-(1-1063)	[local] [snort] in	2008-03-14 14:55:42	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2023-(1-1064)	[local] [snort] in	2008-03-14 14:55:42	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2024-(1-1065)	[local] [snort] in	2008-03-14 14:55:42	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2025-(1-1066)	[local] [snort] in	2008-03-14 14:55:42	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2026-(1-1067)	[local] [snort] in	2008-03-14 14:55:42	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2027-(1-1068)	[local] [snort] in	2008-03-14 14:55:42	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2028-(1-1069)	[local] [snort] in	2008-03-14 14:55:42	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2029-(1-1070)	[local] [snort] in	2008-03-14 14:55:42	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2030-(1-1071)	[local] [snort] in	2008-03-14 14:55:42	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2031-(1-1072)	[local] [snort] in	2008-03-14 14:55:42	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2032-(1-1073)	[local] [snort] in	2008-03-14 14:55:42	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2033-(1-1074)	[local] [snort] in	2008-03-14 14:55:42	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2034-(1-1075)	[local] [snort] in	2008-03-14 14:55:42	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2035-(1-1076)	[local] [snort] in	2008-03-14 14:55:42	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2036-(1-1077)	[local] [snort] in	2008-03-14 14:55:42	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2037-(1-1078)	[local] [snort] virus Redcode.F	2008-03-14 14:55:42	207.46.110.83:1863	10.204.0.10:32767	TCP
☐	#2038-(1-1079)	[local] [snort] in	2008-03-14 14:55:42	207.46.110.83:1863	10.204.0.10:32767	TCP
☐	#2039-(1-943)	[local] [snort] in	2008-03-14 14:55:41	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2040-(1-944)	[local] [snort] in	2008-03-14 14:55:41	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2041-(1-945)	[local] [snort] in	2008-03-14 14:55:41	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2042-(1-946)	[local] [snort] in	2008-03-14 14:55:41	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2043-(1-947)	[local] [snort] in	2008-03-14 14:55:41	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2044-(1-948)	[local] [snort] in	2008-03-14 14:55:41	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2045-(1-949)	[local] [snort] in	2008-03-14 14:55:41	202.43.32.2:80	10.204.0.10:32767	TCP
☐	#2046-(1-950)	[local] [snort] in	2008-03-14 14:55:41	202.43.32.2:80	10.204.0.10:32767	TCP

รูปที่ 4.2.10 หน้าต่างแสดงแพ็คเกตจากหนอน Virus RedCode.F ซึ่งโปรแกรมสนอร์ทตรวจพบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการวิจัย การอภิปราย และข้อเสนอแนะ

5.1 สรุปผลการวิจัย

ปัญหาพิเศษฉบับนี้ได้สร้างขึ้นเพื่อใช้ตรวจสอบแพ็คเกตบนเครือข่าย โดยใช้โปรแกรมที่ชื่อว่า สนอร์ท (Snort) ในการตรวจจับแพ็คเกตขึ้นมา เริ่มแรกนั้นผู้ใช้งานใช้โปรแกรมนี้ตรวจสอบแพ็คเกตทั่วไปที่วิ่งเข้าเครื่องที่ทำการติดตั้งโปรแกรมสนอร์ท โดยเริ่มต้นจะทำการตรวจสอบกการใช้งานของโปรแกรมสนอร์ท โดยใช้โปรแกรมไวล์ชาร์คเข้าช่วยตรวจสอบการทำงานของเครือข่าย กฎที่ผู้วิจัยนำมาใช้กับโปรแกรมสนอร์ทนั้น ในเริ่มแรกผู้วิจัยได้ใช้กฎที่สามารถดาวน์โหลดได้ในเว็ปไซต์ของโปรแกรมสนอร์ท เมื่อสามารถใช้งานโปรแกรมได้แล้วจึงสร้างกฎของตนเองขึ้นมาทดลอง กฎที่ผู้วิจัยได้สร้างขึ้นนั้นจะนำมาทดลอง กฎแรกคือกฎสำหรับดูแพ็คเกตที่วิ่งเข้าโปรโตคอลไอซีเอ็มพี หลังจากนั้นเมื่อผู้วิจัยแน่ใจว่ากฎที่สร้างมานั้นถูกต้องแล้วจึงทำการทดลองเกี่ยวกับหนอน โดยหนอนที่ได้นำมาใช้ในการทดลองคือไวรัสเอ็มเอสเอ็น ไวรัสตัวนี้จะมีการทำงานที่การสนทนาข้ามเครือข่ายกันของผู้ใช้งาน หนอนจะมีการส่งตัวเองและมีข้อความหลอกผู้ใช้งานให้ทำการรับไฟล์ที่ตนเองส่ง เมื่อผู้ใช้งานรับไฟล์แล้วหนอนจะสามารถคุกคามเครื่องผู้ใช้งานที่รับไฟล์นั้นได้

ผู้วิจัยจึงทำการศึกษาการทำงานของ โครงสร้างการทำงานของของเอ็มเอสเอ็นก่อนติดไวรัส และหลังจากติดไวรัส ไวรัสเอ็มเอสเอ็นจะแพร่กระจายตัวหลังจากติดเครื่องของผู้ใช้งานและการส่งตัวเองไปยังรายชื่อของผู้ใช้งาน โดยใช้โปรแกรมสนอร์ทเข้าศึกษาถึงแพ็คเกตที่หนอนได้ส่งมาและสร้างกฎสำหรับป้องกันหนอนตัวนี้ โดยกฎที่สร้างขึ้นนั้นผู้วิจัยได้เขียนโปรแกรมให้กฎสามารถพัฒนาตนเองได้ ทำให้ผู้ใช้งานโปรแกรมสนอร์ทนี้สามารถป้องกันการจู่โจมของหนอนที่พัฒนาตัวเองแล้วได้อีกขั้นหนึ่งและมีความสะดวกยิ่งขึ้น

5.2 การวิจารณ์หรืออภิปราย

เนื่องจากโปรแกรมสนอร์ทเป็นโปรแกรมที่ใช้ตรวจสอบเครือข่าย ดังนั้นจึงอาจจะมีผู้นำไปใช้ในทางที่ผิดคือสามารถนำไปใช้ตรวจสอบเครือข่ายของผู้อื่นได้ โปรแกรมน่าจะมีการป้องกันระบบเพื่อให้มีความปลอดภัยสำหรับผู้ใช้งานมากขึ้น และเนื่องจากต้องทำการดาวน์โหลดกฎเพื่อมาใช้ในโปรแกรม ทำให้เกิดความยุ่งยากในการทำงาน ผู้ลักลอบอาจจะมีวิธีใหม่ๆ ในการแอบลักลอบเข้าสู่ระบบของผู้ติดตั้งโปรแกรมที่ไม่ค่อยดาวน์โหลดกฎมาใช้งานได้

โปรแกรมที่กลุ่มผู้วิจัยได้สร้างขึ้นมานั้นเป็นโปรแกรมที่โปรแกรมเมอร์ต้องทำการเขียนกฎอัพเดตตลอด เนื่องจากปัจจุบันได้เกิดไวรัสขึ้นเป็นจำนวนมาก และจะมีการพัฒนาไปเรื่อย จึงต้องมีโปรแกรมเมอร์ที่จะต้องคอยเขียนกฎในการอัพเดตตลอดเวลา เพราะภายในโปรแกรมจะเป็นการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปรียบเทียบกับไวรัสตัวเก่าๆ ทำให้ผู้ใช้งานบางท่านที่เขียนโปรแกรมไม่เป็นไม่สามารถใช้งานในส่วนนี้ได้อย่างมีประสิทธิภาพ

5.3 ข้อเสนอแนะ

เนื่องจากกฎที่ผู้วิจัยได้ทำให้มีการพัฒนาตนเองนั้น คือการใช้อักขระเดิมของกฎที่มีอยู่แล้วตรวจสอบเพียงแค่ส่วนที่ตรงกับกฎเดิม หนอนตัวใหม่อาจจะมีการพัฒนารูปแบบให้แตกต่างจากเดิมได้ ซึ่งกลุ่มผู้วิจัยยังไม่ได้ทำการค้นคว้าในส่วนนี้

ผู้ทำการวิจัยได้เลือกหนอนเอ็มเอสเอ็นขึ้นมาใช้ในการทดลองเนื่องจากหนอนชนิดนี้เป็นที่แพร่หลายในปัจจุบันและไม่มีความเสียหายต่อระบบมาก สามารถทำการจำกัดได้ง่าย แต่ในขณะเดียวกันก็ไม่สามารถทดลองได้ว่ากฎที่เขียนขึ้นนั้นมีละเอียดพอหรือไม่ กฎที่ได้ใช้ในการทดลองสามารถนำไปใช้งานจริงได้ แต่ก็ยังไม่รัดกุมเพียงพอ ผู้ที่จะนำปัญหาพิเศษนี้ไปใช้ต่อ สามารถพัฒนาในส่วนนี้ต่อไปได้ และส่วนโปรแกรมที่ผู้ใช้ทำนั้นยังคงเป็นแบบสแตติค (Static) ที่ทำการป้องกันหนอนที่ผู้ทำการวิจัยทดลองเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายการอ้างอิง

[1] Charlie Scott, Paul Wolfe and Bert Hayes, "SNORT FOR DUMMIES", Wiley Publishing, 2004.

[2] ดร.วีระศักดิ์ ซึงถาวร, "JAVA PROGRAMMING Volume I", ซีเอ็ดยูเคชั่น, 2548.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

ก.1 คู่มือการติดตั้งโปรแกรม

ก.1.1 การติดตั้ง Fedora 8

ผู้ใช้ที่ไม่มีโปรแกรมเฟดอราอยู่ สามารถดาวน์โหลดได้ในเว็ปไซต์นี้ http://torrent.fedoraproject.org/ โดยสามารถเลือกเวอร์ชันที่ผู้ใช้ต้องการจะดาวน์โหลดได้ ในที่นี้กลุ่มผู้วิจัยได้เลือกใช้เวอร์ชัน 8 ซึ่งจะเตรียมโปรแกรม PHP ไว้ให้ และเลือกดาวน์โหลดแบบทั้งหมด โดยกลุ่มผู้วิจัยได้ดาวน์โหลดตามโลเคชั่นของเว็ปดังต่อไปนี้ fedora/linux/releases/8/Live/*arch*/iso/F-8-*arch*-DVD.iso และนำโปรแกรมลงแผ่นที่เตรียมไว้

โดยการลงโปรแกรมนั้นผู้ใช้งานจะต้องปิดเครื่องคอมพิวเตอร์ก่อน แล้วจึงใส่แผ่นเพื่อลงโปรแกรมเฟดอรา 8 โดยผู้ใช้งานควรจะแบ่งส่วนของข้อมูลตามข้อกำหนดไม่ให้น้อยกว่าดังต่อไปนี้

Directory	Minimum size
/	250 MB
/usr	250 MB
/tmp	50 MB
/var	384 MB
/home	100 MB
/boot	65 MB

ก.1.2 การติดตั้งโปรแกรมสนอร์ท

ก.1.2.1 การติดตั้งโปรแกรมอื่นๆ ก่อนจะลงโปรแกรมสนอร์ท

ก่อนที่จะสามารถใช้โปรแกรมสนอร์ทได้ จะต้องการโปรแกรมเพื่อช่วย ซึ่งจะต้องใช้คำสั่งเพื่อลงโปรแกรม mySQL, PHP และอัพเดตไลบารี pcre ดังต่อไปนี้ (ในเวอชั่นเฟดอรา 8 จะมีการเตรียม mySQL, PHP ไว้ให้แล้ว)

```
yum –y install mysql mysql-bench mysql-server mysql-devel mysqlclient10 php-mysql httpd gcc pcre-devel php-gd gd mod_ssl php-pear gliv2-devel gcc-c++
```

- ทำการติดตั้ง ADODB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Tar –zxvf addb493a.tgz –C /var/www/html/

- ทำการติดตั้ง ADODB

Tar –zxvf addb493a.tgz –C /var/www/html/

เมื่อผู้ใช้งานได้ทำการลงโปรแกรม MySQL แล้วให้ทำการสร้างตารางของสนอร์ทตามดังต่อไปนี้ โดยให้เข้าไปที่ Terminal

- ทำการกำหนดพาสเวิร์ดในการเข้าใช้งาน MySQL

mysql

mysql> SET PASSWORD FOR root@localhost=PASSWORD('password');

>Query OK, 0 rows affected (0.25 sec)

- ทำการสร้างตารางเอาไว้เก็บข้อมูลจากสนอร์ท

mysql> create database snort;

>Query OK, 1 row affected (0.01 sec)

- ทำการสร้างพาสเวิร์ดสำหรับสนอร์ท โดยที่พาสเวิร์ดจะต้องตรงกับ snort.conf และสร้างตารางของสนอร์ททั้งหมด

mysql> grant INSERT,SELECT on root.* to snort@localhost;

>Query OK, 0 rows affected (0.02 sec)

mysql> SET PASSWORD FOR snort@localhost=PASSWORD('password_from_snort.conf');

>Query OK, 0 rows affected (0.25 sec)

mysql> grant CREATE, INSERT, SELECT, DELETE, UPDATE on snort.* to snort@localhost;

>Query OK, 0 rows affected (0.02 sec)

mysql> grant CREATE, INSERT, SELECT, DELETE, UPDATE on snort.* to snort;

>Query OK, 0 rows affected (0.02 sec)

mysql> exit

>Bye

mysql -u root -p < ~/snortinstall/snort-2.6.0/schemas/create_mysql snort

Enter password: the mysql root password

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ทำการทดสอบตารางที่สร้างขึ้น โดยจะขึ้นหน้าต่างของตารางฐานข้อมูล ดังนี้

```
mysql -p
>Enter password: (Enter your password)

mysql> SHOW DATABASES;
+------------+
| Database
+------------+
| mysql
| Snort
| test
+------------+
3 rows in set (0.00 sec)

mysql> use snort
>Database changed

mysql> SHOW TABLES;
+------------------+
| Tables_in_snort
+------------------+
| data
| detail
| encoding
| event
| icmphdr
| iphdr
| opt
| reference
| reference_system
| schema
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
| sensor
| sig_class
| sig_reference
| signature
| tcphdr
| udphdr
+------------------+
16 rows in set (0.00 sec)
exit;
```

ก.1.2.2 การติดตั้งโปรแกรมสนอร์ท

กลุ่มผู้วิจัยได้สร้างโฟลเดอร์ของสนอร์ทไว้ที่ / โดยจะใช้คำสั่งในการลงโปรแกรมดังต่อไปนี้

- ดาวน์โหลดโปรแกรมสนอร์ท โดยเปิดที่ Terminal แล้วกดคำสั่งต่อไปนี้

  ```
  wget http://www.snort.org/dl/current/snort-2.8.0.tar.gz
  tar -xvzf snort-2.8.0.tar.gz
  cd snort-2.8.0
  ./configure --with-mysql --enable-dynamicplugin
  make
  make install
  groupadd snort
  useradd -g snort snort –s /sbin/nologin
  mkdir /etc/snort
  mkdir /etc/snort/rules
  mkdir /var/log/snort
  ```

- ทำการดาวน์โหลดกฎของสนอร์ท

  เข้าไปที่โฟลเดอร์ที่ติดตั้งโปรแกรมสนอร์ทไว้อยู่ โดยเปิดจาก Terminal ดังนี้

  ```
  cd ..
  cd snort-2.8.0
  cd etc/
  ```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

wget http://www.snort.org/pub-bin/downloads.cgi/Download/vrt_pr/snortrules-pr-2.8.0.tar.gz

แล้วทำการเปิดซิปไฟล์กฎแล้วทำตามคำสั่งต่อไปนี้

cp * /etc/snort/rules

gedit snort.conf (ใน path /etc/snort)

เมื่อเปิดไฟล์ snort.conf แล้วให้แก้ข้อมูลดังต่อไปนี้

- แก้ไขไอพีแอดเดรส

  var HOME_NET 10.0.0.0/24

  var EXTERNAL_NET !$HOME_NET

  เปลี่ยนจาก "var RULE_PATH ../rules" เป็น "var RULE_PATH /etc/snort/rules"

- เพิ่มคำสั่งหลังจากบรรทัด says "preprocessor stream4_reassemble" ให้กลายเป็น

  "preprocessor stream4_reassemble: both,ports 21 23 25 53 80 110 111 139 143 445 513 1433" (without the quotes)

- แก้ข้อมูลของดาต้าเบสดังต่อไปนี้

  output database: log, mysql, user=snort password=<the password you gave it> dbname=snort host=localhost

- ทำการรันโปรแกรมสนอร์ท

  Change directory to /etc/init.d and type:

  wget http://internetsecurityguru.com/snortinit/snortchmod 755 snort

  chkconfig snort on.

ก.1.2.3 การติดตั้งโปรแกรมเบส (BASE)

1.1.1 การติดตั้งโปรแกรมเบส ให้เข้าไปที่ Terminal และทำตามคำสั่งดังต่อไปนี้

cd /var/www/html

tar –xvzf /root/snortinstall/base-1.2.6.tar.gz

1.1.2 การ configuration โปรแกรมเบส

เปลี่ยนชื่อของโฟลเดอร์จาก base-1.2.6.tar.gz เป็น base

Copy the base_conf.php.dist to base_conf.php

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แก้ไขข้อมูลในไฟล์ base_conf.php ดังนี้

```
$BASE_urlpath = "/base";
$DBlib_path = "/var/www/adodb/ ";
$DBtype = "mysql";
$alert_dbname = "snort";
$alert_host = "localhost";
$alert_port = "";
$alert_user = "snort";
$alert_password = "password_from_snort_conf";
```

แล้วทำการติดต่อกับฐานข้อมูล

```
$archive_exists = 0; # Set this to 1 if you have an archive DB
```

หลังจากที่เปิดการใช้งานของโปรแกรมสนอร์ทแล้ว (chkconfig snort on) ให้กดคำสั่งที่ Terminal ดังต่อไปนี้

```
service snort start
```

ไปที่เว็ปบราวเซอร์แล้วเข้าสู่เซนเซอร์โดยใช้คำสั่งที่เว็ปบราวเซอร์ https://ip_address(ในที่นี้ใช้ localhost)/base

ก.2 คู่มือการใช้งานโปรแกรม

คำสั่งที่จะใช้ในการทำงานของโปรแกรมต่างๆ มีดังนี้

ทำการเปิดเซอวิสที่ต้องการใช้ในโปรแกรมใน Teminal ใช้คำสั่งดังนี้

```
chkconfig httpd on
chkconfig mysqld on
service httpd start
service mysqld start
```

เริ่มการทำงานของสนอร์ทดังนี้

```
/etc/snort –D --c/etc/snort/snort.conf
```

ก.3 การรันโปรแกรม .jar

ทำการเปิดเซอวิสที่ต้องการใช้ในโปรแกรมใน Teminal ใช้คำสั่งดังนี้

```
java –jar  [$path (.jar)]
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้