การพัฒนาระบบเพื่อตรวจจับและวิเคราะห์การสื่อสารข้อมูลแบบเพียร์ทูเพียร์

DEVELOPMENT OF DETECTING AND ANALYZING PEER TO PEER COMMUNICATION SYSTEM

ธเนศ ศิริเวชวราวุธ

นิธิ ยอดมงคล

ยศพล จิตบรรเทิงพันธ์

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

DEVELOPMENT OF DETECTING AND ANALYZING PEER TO PEER COMMUNICATION SYSTEM

THANET SIRIWETWARAWUT

NITI YODMONGKOL

YOSAPHOL JITBANTERNGPHAN

A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR OF SCIENCE

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE

FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2006

หัวข้อปัญหาพิเศษ	การพัฒนาระบบเพื่อตรวจจับและวิเคราะห์การสื่อสารข้อมูลแบบเพียร์ทูเพียร์ DEVELOPMENT OF DETECTING AND ANALYZING PEER TO PEER COMMUNICATION SYSTEM	
ชื่อนักศึกษา	นายธเนศ ศิริเวชวราวุธ	46050293
	นายนิธิ ยอดมงคล	46050300
	นายยศพล จิตบรรเทิงพันธ์	46050313
ปริญญา	วิทยาศาสตรบัณฑิต	
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์	
สาขาวิชา	วิทยาการคอมพิวเตอร์	
อาจารย์ที่ปรึกษา	อ.ศังกรศรัณย์ ล่องชูผล	

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2549

คณะกรรมการสอบ		ลายมือชื่อ
ประธานกรรมการ	รศ. ไพรบูลย์ พันธรักษ์พงษ์	
กรรมการ	อ. อัคเดช อุดมชัยพร	
กรรมการและอาจารย์ที่ปรึกษา	อ. ศังกรศรัณย์ ล่องชูผล	

(รองศาสตราจารย์ ดร. วีระ บุญจริง)
หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

หัวข้อปัญหาพิเศษ	การพัฒนาระบบเพื่อตรวจจับและวิเคราะห์การสื่อสารข้อมูลแบบเพียร์ทูเพียร์ DEVELOPMENT OF DETECTING AND ANALYZING PEER TO PEER COMMUNICATION SYSTEM
ชื่อนักศึกษา	นายธเนศ ศิริเวชวราวุธ 46050293 นายนิธิ ยอดมงคล 46050300 นายยศพล จิตบรรเทิงพันธ์ 46050313
ปริญญา	วิทยาศาสตรบัณฑิต
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์
สาขาวิชา	วิทยาการคอมพิวเตอร์
ปีการศึกษา	2549
อาจารย์ที่ปรึกษา	อ.ศังกรศรัณย์ ส่องชูผล

บทคัดย่อ

ในปัจจุบันระบบอินเตอร์เน็ตมี บทบาทเป็นอย่างมากในด้านต่างๆไม่ว่าจะเป็นด้านธุรกิจ ด้านการศึกษา ด้านมัลติมีเดียและในอีกหลายๆด้าน ซึ่งในแต่ละวันจะมีผู้ใช้ระบบเครือข่ายมากมาย เพื่อติดต่อธุรกิจและค้นหาข้อมูลต่างๆ ซึ่งผลที่ตามมาก็คือ การขยายตัวและพัฒนาการของระบบเครือข่าย ซึ่งในปัจจุบันมีการสื่อสารแบบใหม่ที่เรียกว่า การสื่อสารแบบเพียร์ทูเพียร์ ซึ่งการสื่อสารรูปแบบนี้นั้นทำให้เกิดปริมาณการรับส่งข้อมูลจำนวนมาก ทำให้รบกวนปริมาณการรับส่งข้อมูลรวมทั้งหมดของการใช้งานเครือข่ายภายในองค์กร

ในการศึกษานี้เราจึงได้พยายามที่จะพัฒนาระบบที่สามารถตรวจสอบและแจ้งเตือนและทำการเก็บข้อมูลเหล่านั้น เพื่อนำไปใช้ในการพิจารณาและทำการกำจัดการสื่อสารแบบเพียร์ทูเพียร์ ซึ่งเป็นการอำนวยความสะดวกต่อผู้ดูแลระบบในการตรวจสอบการใช้งานแบบเพียร์ทูเพียร์ ซึ่งระบบนี้ได้นำเอาคุณสมบัติของระบบตรวจจับผู้บุกรุกมาใช้ในการพัฒนา

Special Topic	DETECTING AND ANALYZING PEER TO PEER COMMUNICATION SYSTEM DEVELOPMENT	
Students	Mr.Thanet Sriwetwarawut	46050293
	Mr.Niti Yodmongkol	46050300
	Mr.Yosaphol Jitbanterngphan	46050313
Degree	Bachelor of Science	
Department	Mathematics and Computer Science	
Programme	Computer Science	
Academic Year	2006	
Special Project Advisor	Sungkornsarun Longchupol	

ABSTRACT

Nowadays, the Internet has an importance in business, education, multimedia and others. Many people surf the Internet for many reasons, such as searching information, or business contact. The expansion of the internet causes a new kind of communication called P2P communication which it consumes a lot of bandwidth making traffic problems on the network.

The special problem "Detecting and Analyzing Peer to Peer communication system development" is developed to detect some protocols of P2P communication, then alert and store detected information into a database. Therefore, the system can help an administrator detects and tracks a source of the detected P2P communication later.

กิตติกรรมประกาศ

โครงงานปัญหาพิเศษเรื่อง การพัฒนาระบบเพื่อจำกัดการสื่อสารข้อมูลแบบเพียร์ทูเพียร์ สามารถสำเร็จลุล่วงไปได้ด้วยดี ด้วยความช่วยเหลือและความร่วมมือจากหลายๆ ท่าน คณะผู้จัดทำต้องขอขอบพระคุณ อ.ศังกรศรัณย์ ล่องชูผล ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงานปัญหาพิเศษนี้ที่กรุณาให้คำแนะนำในการแก้ปัญหาต่างๆ คอยดูแลเอาใจใส่ และให้การสนับสนุนทางด้านทางด้านซอฟต์แวร์และฮาร์ดแวร์ รวมทั้งเป็นผู้ตรวจสอบความถูกต้องของโครงงานพิเศษฉบับนี้ นอกจากนี้ต้องขอขอบพระคุณศูนย์พัฒนาและวิจัยคอมพิวเตอร์ที่ให้ความช่วยเหลือทางด้านเครื่องมือและอุปกรณ์ที่ใช้สำหรับโครงงานนี้ และต้องขอขอบพระคุณนายพงศกร ปิยะตระกูล (พี่กอล์ฟ) ที่ให้คำแนะนำและช่วยแก้ปัญหาต่างๆ ตลอดทั้งโครงงานนี้

นอกจากนี้คณะผู้จัดทำต้องขอขอบพระคุณ บิดา มารดา ที่ได้ให้ความสนับสนุนทางด้านกำลังใจและทุนทรัพย์ จนการทำปัญหาพิเศษนี้สำเร็จลุล่วงไปด้วยดี รวมทั้งเพื่อนๆ พี่ๆ ทุกคนที่ให้ความช่วยเหลือในด้านต่างๆ เกี่ยวกับปัญหาพิเศษไว้ ณ ที่นี้

คณะผู้จัดทำ

มีนาคม 2550

สารบัญ

หน้า

สารบัญตาราง

ตารางที่ **หน้า**

สารบัญรูป

รูปที่ **หน้า**

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันการสื่อสารข้อมูลในลักษณะเพียร์ทูเพียร์(P2P หรือ Peer to Peer) ได้รับความนิยมอย่างแพร่หลายเนื่องจากความนิยมในการกระจายซอฟต์แวร์และเนื้อหาของข้อมูลบันเทิงต่างๆเช่น เพลง หรือภาพยนตร์ แต่โปรแกรมเพียร์ทูเพียร์กลายเป็นช่องทางของการแพร่กระจายของไวรัส (Virus) และเวอร์ม (Worm) ต่าง ๆ ตลอดจน โปรแกรมเพียร์ทูเพียร์มักจะมาพร้อมกับโปรแกรมแอดแวร์ (Adware) และโปรแกรมสปายแวร์(Spyware) เพราะโปรแกรมเหล่านี้มีผลประโยชน์ทางธุรกิจร่วมกันอยู่ นอกจากโปรแกรมเพียร์ทูเพียร์จะมีช่องโหว่ดังกล่าวแล้ว โปรแกรมเหล่านี้ยังใช้แบนด์วิท(Bandwidth) ของระบบเครือข่ายอย่างมหาศาล ทำให้รบกวนการส่งข้อมูล(Traffic) ของการใช้งานเครือข่ายในองค์กรอย่างหลีกเลี่ยงไม่ได้

องค์กรควรกำหนด "นโยบายในการใช้งานระบบคอมพิวเตอร์ให้ปลอดภัย" ("Security Policy") ซึ่งคือการจัดสรรและการใช้งานเครือข่ายอย่างมีประสิทธิภาพเพื่อให้ผู้ใช้งานในรูปแบบอื่นอย่างข้อมูลสารสนเทศต่างๆที่ร่วมใช้เครือข่ายอินเตอร์เน็ตไม่ถูกแย่งชิงการใช้งานจากปริมาณข้อมูลของการสื่อสารแบบเพียร์ทูเพียร์ จัดการกับโปรแกรมเพียร์ทูเพียร์เหล่านี้ โดยการบล็อกช่องทางหรือพอร์ต(Port) ที่ใช้ในการสื่อสารแบบเพียร์ทูเพียร์เพื่อป้องกันโปรแกรมแอดแวร์และโปรแกรมสปายแวร์ และลดปัญหาการส่งข้อมูลภายในองค์กร

เนื่องจากในขณะนี้องค์กรต่างๆยังไม่มีระบบป้องกันการสื่อสารแบบเพียร์ทูเพียร์ คณะผู้จัดทำจึงคิดค้นระบบที่ใช้ตรวจจับและจำกัดการสื่อสารแบบเพียร์ทูเพียร์เพื่อเพิ่มความปลอดภัยภายในองค์กร โดยการพัฒนาระบบจากโปรแกรมโอเพน ซอร์ซ (Open Source)

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1) เพื่อศึกษารูปแบบการสื่อสารแบบเพียร์ทูเพียร์

2) เพื่อสร้างระบบช่วยในการพิจารณาการจำกัดการใช้งานให้แก่ผู้ดูแลระบบเครือข่าย

3) เพื่อนำตัวจำกัดการสื่อสารแบบเพียร์ทูเพียร์ที่สร้างขึ้นไปประยุกต์ใช้ในระบบงานจริง

1.3 ขอบเขตของปัญหา

1) การสร้างระบบตรวจจับการสื่อสารแบบเพียร์ทูเพียร์นั้นจะพัฒนาบนเครื่องคอมพิวเตอร์ส่วนบุคคลทั่วไป (PC-Based) ที่มีระบบปฏิบัติการลีนุกซ์(Linux)

2) ใช้โปรแกรมโอเพน ซอร์ซ ในการพัฒนาระบบจำกัดการสื่อสารแบบเพียร์ทูเพียร์

6) สายยูทีพี (UTP) 10/100 พร้อมหัวอาร์เจ-45 (RJ)-45

3) เก็บข้อมูลการใช้งานโปรโตคอล(Protocol)เพียร์ทูเพียร์ของผู้ใช้งานในเครือข่าย

4) สามารถทำให้ระบบแจ้งเตือนการสื่อสารบนเพียร์ทูเพียร์ที่นิยม 3 ตัว คือ บิททอเรนต์ (BitTorrent) จีนูเทลล่า (Gnutella) และอีดอนกี้ (eDonkey) ที่เลือกโปรโตคอล 3 ตัวนี้เพราะมีแอพพลิเคชันที่สนับสนุนการทำงานของโปรโตคอลนี้อยู่เป็นจำนวนมาก

5) นำล็อกไฟล์ (log file) มาเขียนเป็นรายงานให้ผู้ดูแลระบบนำไปใช้ต่อไป

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1) ลดต้นทุนในการซื้อฮาร์ดแวร์ราคาแพง

2) สามารถนำระบบนี้ไปพัฒนาต่อเพื่อเป็นระบบที่สมบูรณ์ยิ่งขึ้นได้

3) สามารถนำระบบนี้มาใช้ในระบบงานจริงได้เพื่อจำกัดการสื่อสารแบบเพียร์ทูเพียร์ที่มีอยู่ในเครือข่ายออกไป

4) สามารถนำข้อมูลการใช้งานของผู้ใช้งานในเครือข่ายไปวิเคราะห์ และนำไปแก้ปัญหาการใช้งานเครือข่าย

1.5 ขั้นตอนในการดำเนินการ

1) ศึกษาการทำงานและพฤติกรรมของเพียร์ทูเพียร์

2) ค้นคว้าหาโปรแกรมโอเพน ซอร์ซ หลายๆตัว แล้วเลือกตัวใดตัวหนึ่งมาพัฒนา

3) ศึกษาโปรแกรมโอเพน ซอร์ซ ที่ต้องการนำมาใช้ คือโปรแกรมสนอร์ท อีเทอร์เรียล และเพียร์ทูเพียร์แอพพลิเคชันเช่น อีมูลล์(Emule) , แซดพีทูพี(ZP2P)และบิตโคเม็ต(Bit Torrent)

4) สร้างระบบที่ต้องการขึ้น คือระบบตรวจจับและวิเคราะห์การสื่อสารแบบเพียร์ทูเพียร์

5) ทดสอบระบบที่สร้างขึ้นโดยทำการสร้างกฎของสนอร์ตและทดลองตรวจจับ

6) ปรับปรุงและแก้ไขข้อผิดพลาด

1.6 อุปกรณ์ที่ใช้ในการทำปัญหาพิเศษ

1.6.1 รายละเอียดด้านอุปกรณ์

1) คอมพิวเตอร์ 3 เครื่อง

2) เครื่องพิมพ์ 1 เครื่อง

3) กระดาษ 80 แกรม ขนาด A4

4) ซีดี-อาร์ (CD-R) 700 เมกะไบต์ (Megabytes)

5) ฮับ (Hub)

6) สายยูทีพี (UTP) 10/100 พร้อมหัวอาร์เจ-45 (RJ)-45

1.6.2 รายละเอียดทางด้านซอฟท์แวร์

1) โปรแกรมโอเพน ซอร์ซ ได้แก่ โปรแกรมสนอร์ทเวอร์ชั่น 2.6.0.2 , อาปาเช่(Apache)เวอร์ชั่น 2.2.4 , พีเอชพี(PHP)เวอร์ชั่น 5.2.0 , อีเทอร์เรียล(Ethereal)เวอร์ชั่น 0.99 และ มายเอสคิวแอล(mySQL) เวอร์ชั่น 5.0

2) ระบบปฏิบัติการลีนุกซ์ฟีโดร่าคอร์4

บทที่ 2
ความรู้เบื้องต้น

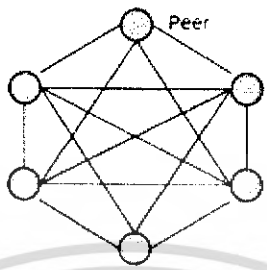
ก่อนที่เราจะเริ่มต้นการพัฒนาระบบเพื่อจำกัดการสื่อสารแบบเพียร์ทูเพียร์เราต้องมีความรู้พื้นฐานในเรื่องต่างๆที่เกี่ยวกับระบบก่อน

2.1 การสื่อสารแบบเพียร์ทูเพียร์

ในปัจจุบันเทคโนโลยีระบบเครือข่ายแบบเพียร์ทูเพียร์ได้รับความสนใจ และเข้ามามีบทบาทในการใช้อินเตอร์เน็ตมากขึ้น เทคโนโลยีนี้ช่วยทำให้ผู้ใช้สามารถแลกเปลี่ยนข้อมูลบริการและทรัพยากรอื่นๆในเครื่องคอมพิวเตอร์ ที่อยู่บนเครือข่ายได้สะดวกมากยิ่งขึ้น ดังเช่น แนพสเตอร์(Napster), จีนูเทลล่า(Gnutella) และ ฟรีเน็ต(Freenet) ซึ่งเป็นโปรแกรมประยุกต์ที่ยอมให้ผู้ใช้อินเตอร์เน็ตค้นหา และแลกเปลี่ยนไฟล์ข้อมูลต่างๆระหว่างคอมพิวเตอร์ซึ่งกันและกันได้ โดยไม่จำเป็นต้องมีคอมพิวเตอร์แม่ข่าย (Central Server) ซึ่งต่างจากระบบไคลเอนต์-เซิร์ฟเวอร์(Client-Server) ซึ่งต้องมีคอมพิวเตอร์แม่ข่าย (Server) คอยให้บริการตามคำขอของเครื่องลูกข่าย (Client) ในการขอข้อมูล บริการ และไฟล์ข้อมูล ตังตัวอย่างที่พบเห็นโดยทั่วไปคือ เวิลด์ ไวด์ เว็บ (WWW) ทั่วไปที่มีอยู่โดยผู้ใช้อินเตอร์เน็ตซึ่งเปรียบได้เสมือนเครื่องลูกข่ายจะใช้เว็บบราวเซอร์(Web Browser)ในการแสดงผลข้อมูลที่มาจากเครื่องแม่ข่าย (Web Server) โดยใช้ โปรโตคอล เฮชทีทีพี(HTTP) เป็นมาตรฐานในการสื่อสารและมีรูปแบบการแสดงผลเป็นแบบ เฮชทีเอ็มแอล(HTML) ซึ่งหากจะเปรียบไปแล้วเทคโนโลยีระบบเครือข่ายแบบเพียร์ทูเพียร์จะมีการทำงานในลักษณะที่เป็นดีเซ็นทรัลไลเซชัน(Decentralization) ส่วนระบบไคลเอนต์-เซิร์ฟเวอร์มีการทำงานเป็นแบบเซ็นทรัลไลเซชัน(Centralization) นั้นเอง เพียร์ทูเพียร์สามารถแบ่งออกได้เป็น 3 แบบ คือ เพียวร์เพียร์ทูเพียร์(Pure P2P), ไฮบริดเพียร์ทูเพียร์(Hybrid P2P), และซูเปอร์เพียร์ทูเพียร์(Super Peer)

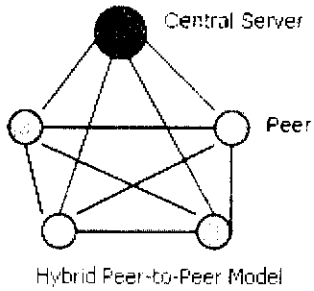
เพียวร์เพียร์ทูเพียร์ จะมีลักษณะที่ตรงข้ามกับโมเดลแบบศูนย์กลางตรงที่ทุกๆเพียร์สามารถติดต่อและแลกเปลี่ยนข้อมูลกันได้โดยตรงโดยไม่ต้องผ่านเครื่องเซิร์ฟเวอร์กลาง จุดเด่นของโมเดลแบบนี้คือความสามารถในการขยายขนาดเครือข่าย, ความคงทน(Fault Tolerant) โดยถ้ามีเพียร์เสียหรือออกไปจากระบบก็จะไม่ส่งผลกระทบต่อระบบโดยรวม แต่โมเดลแบบนี้ก็มีข้อจำกัดตรงที่ควบคุมการไหลของข้อมูลได้ยากทำให้มีปัญหาเรื่องการใช้แบนด์วิธสิ้นเปลือง และโมเดลแบบนี้จะมีความปลอดภัยที่ต่ำเนื่องจากแต่ละเพียร์สามารถเข้าสู่ครือข่ายได้โดยไม่ต้องมีการยืนยันผู้ใช้งาน(Authentication) (โมเดล

แบบนี้ทำการยืนยันผู้ใช้งานได้ยาก) และสามารถที่จะส่งข้อมูลที่อันตรายเข้าสู่เครือข่ายได้โดยง่าย เนื่องจากข้อเสียที่มากของโมเดลแบบนี้ทำให้โมเดลนี้ไม่เป็นที่นิยมเท่าที่ควร



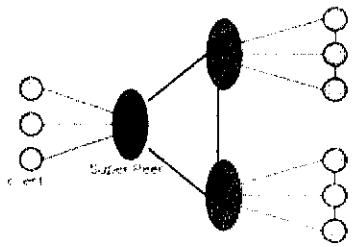
รูปที่ 2.1 แบบจำลองเพียวร์เพียร์ทูเพียร์

ไฮบริดเพียร์ทูเพียร์ จะมีเครื่องเซิร์ฟเวอร์ ที่ทำหน้าที่ควบคุมรายละเอียดของข้อมูลที่อยู่ภายในเครือข่ายแต่การส่งข้อมูลจะเป็นแบบเดียวกับโมเดลเพียวร์เพียร์ทูเพียร์ (ส่งถึงกันโดยตรง) โมเดลแบบนี้จะช่วยลดปัญหาเรื่องการจัดการข้อมูลที่ทำได้ยากในโมเดลแบบเพียวร์เพียร์ทูเพียร์ โดยเครื่องเซิร์ฟเวอร์จะทำหน้าที่คอยตรวจสอบสถานะของทุกๆเพียร์ และควบคุมการไหลของข้อมูลในเครือข่ายแต่เพราะยังต้องใช้เครื่องเซิร์ฟเวอร์กลางอยู่ ดังนั้นถ้าเครื่องเซิร์ฟเวอร์กลางเสียไปก็จะเสียการควบคุมข้อมูลไปแต่ละเพียร์ แต่ยังคงสามารถแลกเปลี่ยนข้อมูลกันได้อยู่ เนื่องจากมีการควบคุมข้อมูลที่ดีดังนั้นโมเดลนี้จึงมีความสามารถในการขยายขนาดเครือข่ายได้ดีกว่าโมเดลเพียวร์เพียร์ทูเพียร์ แต่ก็ยังมีขีดจำกัดของการขยายอยู่ที่จำนวนเครื่องลูกของเครื่องเซิร์ฟเวอร์ที่จะรับได้ โมเดลแบบนี้มีประสิทธิภาพที่จะนำไปใช้กับแอพพลิเคชั่นต่างๆ แต่ไม่สามารถนำไปใช้กับแอพพลิเคชั่นที่มีขนาดของปัญหาใหญ่ๆได้



รูปที่ 2.2 แบบจำลองไฮบริดเพียร์ทูเพียร์

ซูปเปอร์เพียร์ทูเพียร์ เป็นโมเดลใหม่ที่เพิ่งจะเกิดขึ้นไม่นานมานี้โดยเป็นการเอาระบบแบบศูนย์กลางไปรวมอยู่ในระบบแบบกระจาย โมเดลแบบซูปเปอร์เพียร์ทูเพียร์จะช่วยลดปริมาณในการจัดการของเซิร์ฟเวอร์ อีกทั้งช่วยเพิ่มความสามารถในเรื่องของการขยายขนาดและความคงทนของเครือข่าย และลดปัญหาอื่นๆที่เกิดขึ้นในโมเดลแบบเพียวร์เพียร์ทูเพียร์และไฮบริดเพียร์ทูเพียร์ คือเพียร์ที่ทำหน้าที่เหมือนเป็นเซิร์ฟเวอร์กลางให้กับกลุ่มของไคลเอนต์ แต่ละกลุ่มไคลเอนต์จะส่งคำร้องขอและรับผลลัพธ์ของคำร้องขอนั้นจากซูปเปอร์เพียร์ทูเพียร์ในขณะที่ซูปเปอร์เพียร์ทูเพียร์แต่ละเพียร์ก็จะเชื่อมต่อถึงกันด้วยเครือข่ายแบบเพียวร์เพียร์ทูเพียร์ โดยซูปเปอร์เพียร์จะทำหน้าที่เป็นตัวควบคุม (Controller), ปรับแต่ง (Configuration), ดูแล (Administration) และรักษาความปลอดภัย (Security) ให้กับไคลเอนต์ ที่อยู่ในกลุ่มดังนั้นในแต่ละซูปเปอร์เพียร์จะต้องมีโปรโตคอลในการติดต่อสื่อสารอยู่ 2 โปรโตคอล คือโปรโตคอลในการติดต่อสื่อสารระหว่างซูปเปอร์เพียร์กับไคลเอนต์ และโปรโตคอลในการติดต่อสื่อสารระหว่างซูปเปอร์เพียร์กับซูปเปอร์เพียร์อื่น โมเดลแบบซูปเปอร์เพียร์มีจุดเด่นคือช่วยลดเวลาและแบนด์วิธที่ใช้ในการค้นหา, แต่ละหน่วยจะมีความเป็นอิสระสูง, สามารถควบคุมและจัดการได้ง่าย, สามารถทำ Load Balancing ได้เป็นต้น แต่โมเดลซูปเปอร์เพียร์นี้ถ้าซูปเปอร์เพียร์เสียก็จะทำให้ไคลเอนต์ที่อยู่ในกลุ่มนั้นไม่สามารถทำงานได้ แต่ปัญหานี้สามารถลดได้โดยการที่ให้มีซูปเปอร์เพียร์มากกว่าหนึ่งเพียร์ ในแต่ละกลุ่ม



รูปที่ 2.3 แบบจำลองซูปเปอร์เพียร์ทูเพียร์

2.2 ความรู้เบื้องต้นเกี่ยวกับระบบตรวจจับผู้บุกรุก (IDS: Instrusion Detection System)

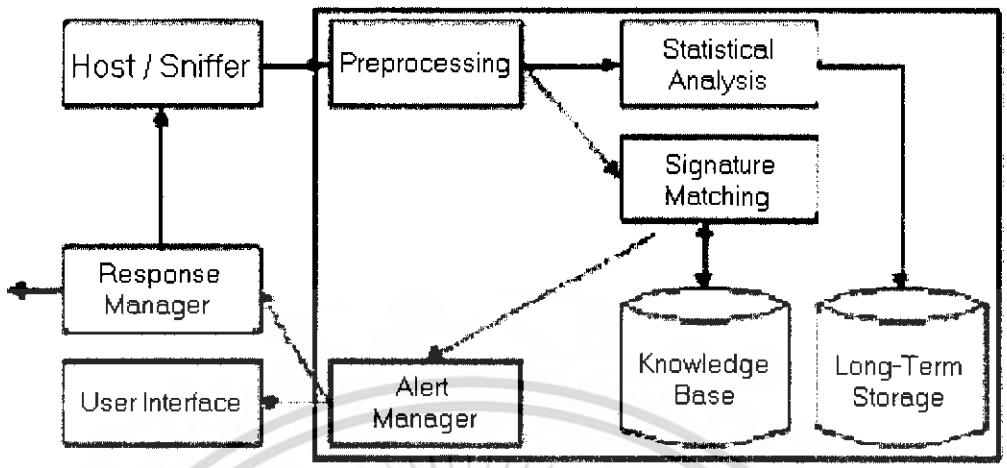
ระบบตรวจจับผู้บุกรุก คือ กระบวนการจำแนกและตอบสนองต่อกิจกรรมที่เกิดขึ้นในเน็ตเวิร์กที่มุ่งร้ายต่อระบบคอมพิวเตอร์และทรัพยากรที่อยู่บนเน็ตเวิร์กนั้น หรือหากอธิบายในภาษาง่ายๆก็คือ เป็นกระบวนการตรวจจับการบุกรุกและการก่อกวนที่เกิดขึ้นบนเครือข่ายนั่นเอง ดังนั้นระบบตรวจจับผู้บุกรุกก็คือระบบที่ประกอบด้วยฮาร์ดแวร์และซอฟต์แวร์สำหรับทำหน้าที่คอยตรวจตราความเป็นไปและพฤติกรรมของข้อมูลที่ผ่านไปมาบนเน็ตเวิร์กว่าน่าสงสัยและมีสิ่งผิดปกติหรือไม่

ถ้าหน่วยงานใดมีการติดตั้งเครือข่ายภายใน (LAN) หรือเครือข่ายอินเทอร์เน็ต เพื่อใช้งานในองค์กร ถึงแม้ว่าระบบจะมีการติดตั้งไฟร์วอลล์ เพื่อป้องกันการบุกรุกจากบุคคลภายนอกแล้วก็ตาม ระบบก็ยังมีช่องโหว่ที่ทำให้เกิดความเสี่ยงต่อการเสียหายได้ดังต่อไปนี้

ความเสี่ยงเนื่องจาก การเปิดบริการใช้งานบางประเภทที่ไฟร์วอลล์ ให้กับบุคคลภายนอกที่มาจากอินเทอร์เน็ต เช่น การบริการเว็บแอพพลิเคชั่น การบริการรับส่งเมล์และการบริการถ่ายโอนข้อมูล (FTP) เป็นต้น เนื่องจาก ผู้ไม่ประสงค์ดี (Hacker) จะนิยมเข้ามาจู่โจมและก่อให้เกิดความเสียหายแก่ระบบงานภายในองค์กร โดยเข้าทางที่ไฟร์วอลล์อนุญาตเป็นส่วนมาก

2.2.1 ส่วนประกอบของระบบตรวจจับผู้บุกรุก

โดยทั่วไป ไอดีเอส มีส่วนประกอบตามรูปข้างล่างนี้



รูปที่ 2.4 ส่วนประกอบของไอดีเอส

- โฮสท์ซิสเต็ม/เน็ตเวิร์กสนิฟเฟอร์ (Host System/Network Sniffer): ทำหน้าที่เป็นตัวตรวจจับเหตุการณ์ต่างๆ ที่เกิดขึ้นในโฮสต์ หรือเครือข่ายข้อมูลจากส่วนนี้เป็นเหมือนอินพุต ที่เข้าไปประมวลผลในไอดีเอส
- พรีโพรเซสซิ่ง (Pre-Processing) : จัดรูปแบบของอินพุต ที่รับเข้ามาเพื่อให้นำไปประมวลผลได้สะดวกต่อไป
- ส่วนวิเคราะห์ผลทางสถิติ (Statistical Analysis)
- ส่วนวิเคราะห์จากพฤติกรรมที่มีแบบแผน (Signature Matching): แนวความคิดตรงนี้นำมาจากที่ว่า การบุกรุกมักจะมีรูปแบบที่ค่อนข้างแน่นอน ส่วนนี้จะทำงานได้ก็ต่อเมื่อมีข้อมูลมากพอที่จะวิเคราะห์ได้ว่าพฤติกรรมที่ปรากฏในระบบเป็นรูปแบบของการบุกรุกหรือไม่
- ฐานความรู้ (Knowledge Base): เป็นตัวเก็บข้อมูลเกี่ยวกับพฤติกรรมของการบุกรุก ข้อมูลนี้ถูกใช้โดยส่วนวิเคราะห์จากพฤติกรรมที่มีแบบแผน ข้อมูลส่วนนี้มีความสำคัญมากกับระบบเป็นตัวตัดสินเลยว่าระบบฉลาดพอที่จะตรวจจับการบุกรุกได้หรือไม่
- อเลิร์ตแมนเนเจอร์ (Alert Manager): ทำหน้าที่เป็นตัวตัดสินใจว่าจะเหตุการณ์ที่เกิดขึ้นในระบบควรจะต้องเตือนหรือไม่ ระบบจะมีความอ่อนไหว มากน้อยแค่ไหนอยู่ที่ตัวนี้ด้วย
- ยูสเซอร์อินเตอร์เฟซ (User Interface): เป็นส่วนที่โต้ตอบกับผู้ใช้ อาจจะเป็นการแสดงผลที่หน้าจอ ส่งเสียงเตือนถึงการบุกรุก สั่งพิมพ์เป็นฮาร์ดก็อบปี้ หรือแม้แต่เชื่อมกับเพจเจอร์ / โทรศัพท์ นอกจากนี้ ยูสเซอร์อินเตอร์เฟซ ยังเป็นส่วนโต้ตอบระหว่างผู้ใช้กับระบบเพื่อ

เปลี่ยนแปลงหรือปรับปรุง ข้อมูลใน ฐานความรู้

• เรสพอนซ์แมนเนเจอร์ (Response Manager): รับข้อมูลจากอเลิร์ตแมนเนเจอร์ เพื่อนำมาตัดสินใจว่าจะโต้ตอบกับการบุกรุกอย่างไร

2.2.2 ประเภทของระบบตรวจจับผู้บุกรุก ออกเป็นกลุ่มต่าง ๆ ได้ดังนี้

2.2.2.1 ระบบตรวจจับผู้บุกรุกบนพื้นฐานโฮสท์ (Host-Based Intrusion Detection System)

เป็นระบบไอดีเอสที่ออกแบบมาเพื่อจุดประสงค์ในการตรวจสอบการจู่โจมที่เกิดขึ้นกับเครื่องใดเครื่องหนึ่งโดยเฉพาะ ซึ่งจะตรวจสอบการถูกจู่โจมที่จะเกิดขึ้นกับบริการต่างๆ หรือแอพพลิเคชันแต่ละตัวได้ เช่น การพยายามที่จะเดารหัสผ่านเข้าสู่ระบบบริการ นอกจากนี้ระบบตรวจจับผู้บุกรุกบนพื้นฐานโฮสต์ยังสามารถที่จะตรวจสอบถึงการเข้ารหัสข้อมูลที่เครื่องนั้นๆ ติดต่อสื่อสารกับระบบอื่นๆ ในเครือข่ายได้เช่น ระบบเครือข่ายที่มีการเชื่อมต่อ VPN อีกทั้งตัวระบบก็ติดตั้งเข้ากับเครื่องเซิร์ฟเวอร์หรือเครื่องคอมพิวเตอร์ที่ต้องการจะป้องกันได้โดยตรง ดังนั้นไม่จำเป็นต้องเพิ่มฮาร์ดแวร์อื่นเข้าไปในระบบอีก

2.2.2.2 ระบบตรวจจับผู้บุกรุกบนพื้นฐานเน็ตเวิร์ก (Network-Based Intrusion Detection System)

เป็นระบบไอดีเอสที่ออกแบบมาทำการเฝ้าดูข้อมูลบนเครือข่ายโดยที่ระบบดังกล่าว จะทำการรับข้อมูลทั้งหมดที่อยู่บนส่วนของเครือข่ายที่รับผิดชอบนอกเหนือจากส่วนของเครือข่ายที่รับผิดชอบและชนิดของการสื่อสารอื่นๆ แล้วระบบดังกล่าวก็ไม่สามารถทำการตรวจจับแพ็กเก็ตต่างๆ ซึ่งจะถูกตรวจจับโดยเซ็นเซอร์ของระบบไอดีเอสและเซ็นเซอร์จะมองเห็นเฉพาะแพ็กเก็ตที่ผ่านส่วนของเครือข่ายที่เซ็นเซอร์นั้นติดอยู่ ดังนั้น แพ็กเก็ตต่างๆ จะเป็นที่สนใจของเซ็นเซอร์ ก็ ต่อเมื่อแพ็กเก็ตนั้นเข้ากับสัญญาณ (Signature) ที่กำหนดซึ่งปกติแล้ว สัญญาณ จะมี 3 ประเภท คือ

1) สัญญาณของสตริง (String Signature)

จะมองหาเท็กซ์สตริง (Text String) ซึ่งอาจบ่งบอกถึงการโจมตีตัวอย่าง เช่น "cat" + + "7%rhost" อาจทำให้ระบบยูนิกซ์ เกิดช่องโหว่ต่อการโจมตีบนเครือข่าย

2) สัญญาณของพอร์ต (Port Signatures)

จะเฝ้าดูการพยายามติดต่อเข้ามาทางพอร์ตที่รู้จักกันดี และมักจะถูกโจมตี เช่น เทลเน็ต จะใช้ทีซีพีพอร์ต 23, เอฟทีพีจะใช้ทีซีพีพอร์ต 21/20, ซันอาร์พีซี (SUNRPC) ใช้ทีซีพี/ยูดีพีพอร์ต 111 และไอแม็ป(IMAP) จะใช้ ทีซีพีพอร์ต 143 ซึ่งถ้าระบบของเราไม่ได้เปิดพอร์ตดัง กล่าวแต่มีการพยายามเชื่อมต่อเข้ามาแสดงว่าแพ็กเก็ตดังกล่าวอาจจะมีประสงค์ร้ายก็ได้

3) สัญญาณของเฮดเดอร์ (Header Signatures)

พยายามมองหาคอมไบเนชัน (Combination) ที่อันตรายและผิดกฎของแพ็กเก็ตเฮดเดอร์ ตัวอย่างที่เห็นได้ชัดของสัญญาณเฮดเดอร์ คือทีซีพีแพ็กเก็ต ซึ่งมีทั้งแฟลก (Flags) แบบซิน (SYN) และฟิน (FIN) ในการติดตั้งใช้งานไอดีเอส ในระบบเครือข่ายคอมพิวเตอร์นั้น การติดตั้งและใช้งานระบบไอดีเอส เพียงประเภทใดประเภทหนึ่งจะได้รับความปลอดภัยเพียงระดับหนึ่ง เพื่อความปลอดภัยที่สูงขึ้น จำเป็นต้องติดตั้งระบบไอดีเอส หลายประเภท รวมทั้ง ระบบไอดีเอส จะต้องมีความสามารถในการทำงานร่วมกัน กับระบบป้องกันรักษาความปลอดภัยประเภทอื่น

2.2.3 ข้อดีของการใช้ระบบตรวจจับผู้บุกรุก

2.2.3.1 การตอบสนองทันทีทันใด

จริงๆแล้วการวิเคราะห์การบุกรุกนั้นหากเป็นผู้เชี่ยวชาญที่มีความรู้ทางด้านเน็ตเวิร์กและโปรโตคอลเป็นอย่างดีก็จะสามารถวิเคราะห์ได้โดยอาศัยเครื่องมือเพียงเล็กน้อยเท่านั้นคือใช้เครื่องมือทำการจัดเก็บบันทึกข้อมูลทั้งหมดที่มีการสื่อสารบนเน็ตเวิร์ก แล้วนำข้อมูลที่ได้เหล่านั้นมาวิเคราะห์โดยพฤติกรรมและความสัมพันธ์ ก็จะสามารถหาสิ่งผิดปกติที่เกิดขึ้นได้ แต่การวิเคราะห์ในลักษณะดังกล่าวจะกระทำได้ก็ต่อเมื่อได้เกิดเหตุการณ์ไปแล้ว เนื่องจากการวิเคราะห์จะเป็นไปในลักษณะวิเคราะห์ข้อมูลย้อนหลัง ไม่สามารถกระทำได้ในทันที ซึ่งไอดีเอส จะช่วยแก้ไขข้อบกพร่องในส่วนนี้ เพราะไอดีเอส สามารถตรวจจับได้ทันทีที่มีความผิดปกติเกิดขึ้น และช่วยให้ทำการแก้ไขได้ทันท่วงที การทำงานพื้นฐานของไอดีเอส จะเหมือนกับการวิเคราะห์ที่ทำโดยคน เพียงแต่ไอดีเอสนั้นทำโดยอัตโนมัติและทำงานอยู่ตลอดเวลาไม่มีหยุด จึงสามารถตอบสนองต่อสิ่งผิดปกติได้รวดเร็วกว่า ซึ่งถ้าให้คนมานั่งตรวจจับก็ไม่สามารถทำได้ตลอดเวลา

2.2.3.2 การมีฐานความรู้ของการวิเคราะห์

จากที่ได้กล่าวมาข้างต้น การที่จะตรวจจับสิ่งผิดปกติและแยกแยะกิจกรรมเหล่านั้นออกจากการสื่อสารข้อมูลตามปกติได้นั้นต้องอาศัยความชำนาญ และเข้าใจรูปแบบการสื่อสารข้อ มูลและการบุกรุกเป็นอย่างดี นั่นคือทักษะที่จำเป็นของนักวิเคราะห์การบุกรุก (Intrusion Analyst) ซึ่งผู้เชี่ยวชาญในระดับที่จะทำงานเช่นนี้ได้มีไม่มากนัก ประกอบกับเทคนิคและกลวิธีในการบุกรุกและก่อกวนนั้นได้พัฒนาขึ้นทุกวัน วิธีตรวจจับและวิเคราะห์จำเป็นต้องพัฒนาตามให้สอดคล้องกันจึงจะตรวจจับได้อย่างมีประสิทธิภาพ ซึ่งในส่วนนี้ผู้เชี่ยวชาญอาจจะทำได้ไม่ดีเท่าไอดีเอส สามารถช่วยแบ่งเบาการวิเคราะห์ลงได้มาก โดยหากรู้พฤติกรรมแน่ชัดว่าเป็นการ มุ่งร้ายก็ให้จัดเก็บข้อมูลเหล่านี้ให้ไอดีเอส เสีย เมื่อกิจกรรมดังกล่าวเกิดขึ้นในเน็ตเวิร์ก ไอดีเอสก็สามารถตรวจพบได้ทันที และเมื่อค้นพบรูปแบบใหม่ก็จัดเก็บลงในไอดีเอสอีก ทำให้ไอดีเอสเสมือนมีฐานความรู้ในการวิเคราะห์การบุกรุกได้ดีในระดับหนึ่ง และขีดความสามารถก็จะเพิ่มขึ้นเรื่อยๆ ตามปริมาณที่เก็บอยู่ในฐานความรู้นั่นเอง หากมี

การบำรุงรักษาฐานความรู้ในตัวไอดีเอสได้ดีและนำไอดีเอสไปได้ ถึงขั้นนี้แล้วถึงแม้ว่าจะเป็นไอดีเอสแบบธรรมดาๆ ก็มีความสามารถมากกว่าผู้บริหารในระดับทั่วไปเสียอีก สำหรับนักวิเคราะห์แล้วเมื่อมีไอดีเอสจะทำให้ไม่ต้องห่วงหน้าพะวงหลัง เพราะการบุกรุกที่สามารถตรวจจับได้ง่ายๆ ก็สามารถตรวจพบได้โดยไอดีเอส อย่างน้อยไอดีเอสก็ช่วยกลั่นกรองข้อมูลเบื้องต้นได้ในระดับหนึ่งและแบ่งเบาภาระได้ดีพอสมควร

2.2.3.3 การช่วยตรวจสอบข้อบกพร่องของระบบป้องกันอื่นๆ

เน็ตเวิร์กของผู้ใช้อาจจะมีการป้องกันการบุกรุกอยู่แล้วโดยใช้ไฟร์วอลล์ (Firewall) อย่างไรก็ตามไฟร์วอลล์มิใช่เครื่องมือที่จะป้องกันการบุกรุกได้โดยอัตโนมัติจะต้องอาศัยผู้ที่บริหารระบบกำหนดกฎให้เหมาะสมกับการใช้งาน อีกประการหนึ่ง ถึงแม้ว่าจะมีการตั้งกฎที่เหมาะสม แล้วก็ตาม แต่กฎเหล่านั้นอาจไม่สามารถป้องกันการบุกรุกได้ การบริหารไฟร์วอลล์ที่ดีก็ควรตรวจสอบย้อนหลัง (Audit) และการทดสอบเจาะระบบ (Penetration Test) เพื่อเป็นการตรวจทานระบบอีกครั้งหนึ่ง

ไอดีเอสสามารถช่วยได้มาก โดยติดตั้งไอดีเอสไว้หลังไฟร์วอลล์ และทำการทดสอบเจาะระบบด้วยวิธีการต่างๆ เพื่อดูว่าจะมีเทคนิคใดที่สามารถเจาะผ่านไฟร์วอลล์ได้บ้าง และหากมีแพ็กเก็ตใดผ่านเข้าไปไดไอดีเอส ก็จะตรวจพบ ทำให้ผู้บริหารระบบสามารถปรับปรุงกฎให้รัดกุมมากขึ้น

2.2.4 ข้อเสียของการใช้ระบบตรวจจับผู้บุกรุก

ถึงแม้ว่าไอดีเอส จะมีประโยชน์ค่อนข้างมากในการรักษาความปลอดภัย และเตือนภัย ล่วงหน้า แต่ก็มีข้อเสียอยู่หลายประการซึ่งผู้ที่จะนำไปใช้จะต้องตระหนักไว้

2.2.4.1. การละเมิดความเป็นส่วนบุคคล

เนื่องจากไอดีเอส มีพื้นฐานจากการนำข้อมูลทั้งหมดที่ทำการสื่อสารกันมาทำการวิเคราะห์ ซึ่งข้อมูลเหล่านั้นจะต้องครอบคลุมข้อมูลทั่วไปที่มีการสื่อสารกันตามปกติ และการที่จะทราบว่ามีความผิดปกติหรือไม่นั้นก็จะต้องอ่านข้อมูลทั้งหมดด้วย ดังนั้นไม่ว่ากิจกรรมใดที่เกิดขึ้นในเน็ตเวิร์กไม่ว่าจะท่องเว็บ, การดาวน์โหลดข้อมูล, การแชทคุยกัน, ไอซีคิว, อีเมล์ และกิจกรรมอื่นๆ ที่สื่อสารผ่านเน็ตเวิร์ก ก็จะเปิดอ่านได้จากไอดีเอส นั้นหมายความว่าไอดีเอส สามารถนำไปใช้งานในทางที่ผิดเพื่อละเมิดสิทธิส่วนบุคคลได้ การทำงานของ ไอดีเอส เปรียบเสมือนการที่ตำรวจต้องทำการตรวจสอบและดักจับผู้ไม่หวังดีที่คอยโทรศัพท์ก่อกวนชาวบ้านในหมู่บ้าน และเพื่อการนี้ตำรวจจึงต้องทำการดักฟังโทรศัพท์ของทุกคนที่อยู่ในหมู่บ้านนั้น ซึ่งอาจจะมีเพียงหนึ่งในพันที่เป็นผู้ร้ายแต่ตำรวจผู้ทำหน้าที่ดักฟังก็จะรู้ความลับของคนทุกคน บางทีการที่มีคนมาดักฟัง ความลับของทุกคนอาจจะ เป็นอันตรายกว่าการโดนผู้ร้ายก่อกวนก็เป็นได้

ดังนั้นการนำไอดีเอส มาติดตั้งในเน็ตเวิร์กจะต้องได้รับอนุมัติจากหน่วยงานอย่างถูกต้องแล้วเท่านั้น และผู้ที่ทำหน้าที่ในด้านนี้ต้องเป็นผู้ที่ได้รับความไว้วางใจและมีความรับผิดชอบสูงอันที่จะไม่ละเมิดสิทธิส่วนบุคคลของผู้อื่น และหากเห็นข้อมูลใดๆ ก็ต้องไม่เปิดเผยข้อมูลเหล่านั้นแก่บุคคลอื่น โดยทั่วไปแล้วการติดตั้งอุปกรณ์ที่สามารถดักอ่านข้อมูลของผู้อื่นบนเน็ตเวิร์กได้นั้นจะ เป็นข้อห้ามอันดับต้นๆ ในนโยบายรักษาความปลอดภัยเลยทีเดียว สิ่งที่เป็นข้อสังเกตคือ การกระทำในลักษณะนี้ยากต่อการป้องกันทางเทคนิค และมีบทลงโทษสำหรับผู้ที่ละเมิดในขั้นรุนแรง

2.2.4.2 การตอบโต้อัตโนมัติ

ไอดีเอสที่มีจำหน่ายอยู่ในท้องตลาด จะมีส่วนหนึ่งที่ผู้ใช้สามารถกำหนดการดำเนินการอย่างหนึ่ง อย่างใดเมื่อตรวจพบการบุกรุกเกิดขึ้น เช่น ส่งจดหมายเตือนผู้ดูแลระบบ, เรียกวิทยุติดตามตัว, ส่งข้อมูลไปยังไฟร์วอลล์เพื่อจำกัดการเข้าออกของข้อมูล และสิ่งสำคัญที่สุดที่อาจจะเกิดผลเสียหายใหญ่มากต่อเจ้าของได้ก็คือ การโจมตีกลับไปยังต้นกำเนิดของการบุกรุก (Counter Attack) โดยที่ไอดีเอส เองก็จะรู้จักวิธีการโจมตีต่างๆ ดีอยู่แล้ว จึงมิใช่เรื่องยากเย็นแต่อย่างใดที่จะทำการโจมตีผู้อื่น ผู้ผลิตจึงมักเพิ่มเติมส่วนนี้ให้แก่ไอดีเอส เสมือนหนึ่งติดอาวุธไว้ให้ต่อกรกับแฮคเกอร์เลยทีเดียว ผู้ดูแลระบบบางส่วนอาจจะรู้สึกสะใจและคิดว่าเหมาะสมแล้วกับการโจมตีกลับไปยังแฮคเกอร์เหล่านั้นให้หลาบจำและไม่เข้ามาข้องแวะอีก เป็นนโยบายการรักษาความปลอดภัยแบบตาต่อตาฟันต่อฟัน และเชื่อว่าหากกำหนดให้การโจมตีกลับเป็นไปอย่างอัตโนมัติ แล้วน่าจะทำให้ปลอดภัยมากขึ้น ในค่ำคืนที่สงบใครจะไปรู้ว่าไอดีเอส กำลังต่อกรอยู่กับแฮคเกอร์ที่กำลังแอบเข้ามาในระบบอย่างสุดกำลัง และสู้ยิบตาเพื่อรักษามิให้แฮคเกอร์บุกรุกเข้ามาในเน็ตเวิร์กได้ ในความเป็นจริงแล้วการตัดสินว่าผู้ใดเป็นแฮคเกอร์อย่างชัดเจนมิได้ทำได้โดยง่ายและในเวลาอัน รวดเร็ว การที่กำหนดให้ไอดีเอส ทำการตอบโต้กลับไปในทันทีโดยมีข้อมูลเพียงผิวเผินนั้น นอกจากจะไม่ช่วยให้เน็ตเวิร์กเราปลอดภัยแล้วยังจะทำให้เรากลายไปเป็นแฮคเกอร์ที่คอยโจมตีผู้อื่นเสียเองยกตัวอย่างความเสียหายเช่น

- การวิเคราะห์ผิดพลาดเข้าใจว่ากิจกรรมที่เกิดขึ้นเป็นการบุกรุกและไอดีเอสก็ดำเนินการโจมตีกลับไปในทันที กรณีนี้ผู้บริสุทธิ์ก็จะถูกโจมตีจากไอดีเอส ของเราโดยที่ไม่รู้เรื่องใดๆ
- การวิเคราะห์ถูกต้องแต่แอดเดรสปลายทางเป็นแอดเดรสปลอมกรณีนี้หากไอดีเอสไม่มีกลไกในการตรวจสอบแอดเดรสที่มีประสิทธิภาพ อาจไม่สามารถแยกแยะได้ว่าต้นทางของการโจมตีแท้จริงนั้นเป็นที่ไหน และเมื่อทำการโจมตีกลับไปก็อาจจะมิใช่ตัวการที่แท้จริง และเหตุการณ์จะเลว ร้ายยิ่งขึ้นหากแอดเดรสที่ปลอมมานั้นเป็นหน่วยงานทางความมั่นคงหรือหน่วยงานทางการทหาร และเมื่อผู้ดูแลระบบอาจจะตระหนักได้ว่าไอดี

เอสตัวเดียวอาจจะทำให้เขาต้องไปนอนในคุกหลายคืน เทคนิคการปลอมแอดเดรสในลักษณะนี้อาจจะเป็นการยืมมือ ไอดีเอส ของเราไปโจมตีผู้อื่นอีกทอดหนึ่งได้เป็นอย่างดี

- การวิเคราะห์แอดเดรสที่ถูกต้อง และการโจมตีกลับไปก็ตรงไปยังแฮคเกอร์อย่างถูกต้องตามที่ต้องการ แต่ผลที่ได้ก็เพียงอาจจะทำให้แฮคเกอร์หยุดความพยายามไปชั่วขณะเท่านั้น อีกไม่นานก็จะหาวิธีกลับมาใหม่ และไม่เกิดผลใดๆ เลยนอกจากเป็นการยั่วยุให้เกิดความรุนแรงมากขึ้นเท่านั้นสิ่งสำคัญที่ผู้ทำหน้าที่ด้านความปลอดภัยและผู้บริหารระบบต้องตระหนักไว้คือท่านไม่มีสิทธิพิเศษที่จะตอบโต้ผู้บุกรุกโดยการโจมตีกลับไม่ว่าจะกรณีใดสิ่งที่ท่านจะทำได้ดีที่สุดคือทำให้ระบบแข็งแรงมั่นคงและปลอดภัยที่สุดเท่านั้น นั้นคือปิดประตูบ้านให้แน่น ตรวจตราอย่างรัดกุม และใช้งานเท่าที่จำเป็น ส่วนผู้ที่กระทำผิดเหล่านั้นควรปล่อยให้เป็นไปตามกฎหมายและกระบวนการยุติธรรมจะดีที่สุด เพราะการตอบโต้การกระทำผิดกฎหมายด้วยวิธีที่ผิดกฎหมายจะทำให้เรากลายเป็นจำเลยไปด้วยในที่สุด

2.2.4.3 การเตือนภัยผิดพลาด

ข้อนี้อาจจะไม่ใช่ข้อเสียที่สำคัญที่สุดของการใช้ ไอดีเอส หากผู้ใช้มีความรู้ในการใช้งานที่ดีพอและเข้าใจหลักการวิเคราะห์การบุกรุกของไอดีเอส ได้ดีอย่างที่ได้กล่าวมาแล้วข้างต้นก็คือ อาจจะมีกิจกรรมหลายอย่างที่มีลักษณะใกล้เคียงหรือบางครั้งเหมือนกับการบุกรุก ซึ่งแน่นอนว่า หากไอดีเอสถูกกำหนดให้ตรวจจับกิจกรรมประเภทดังกล่าวแล้ว ก็จะมีการเตือนในทันทีที่ตรวจพบและเป็นหน้าที่ของนักวิเคราะห์ที่จะทำการสืบค้นข้อมูลด้านอื่นๆมาประกอบการวินิจฉัยอีกครั้งหนึ่งว่าพฤติกรรมที่ตรวจพบนั้นเป็นการบุกรุกหรือไม่ อย่างไร ไอดีเอสที่ถูกกำหนดให้มีความว่องไวเป็นพิเศษมักจะสามารถตรวจจับพฤติกรรมที่ก้ำกึ่งนั้นได้มากเป็นพิเศษ

ตัวอย่างเช่นไอดีเอสได้ถูกกำหนดไว้ว่า เมื่อได้รับปิงแพ็กเก็ต (Ping Packet) จากแอดเดรสเดิมติดต่อกัน 10 แพ็กเก็ตภายใน 30 วินาที ให้เตือนว่าเป็นการพยายามโจมตีโดยเทคนิคปิงฟลัด (Ping Flood) เป็นต้นหากเน็ตเวิร์กดังกล่าวเป็นเน็ตเวิร์กที่ใช้งานโดยวิศวกรระบบ และมีการทดสอบการปิง (Ping) บ่อยๆ อาจจะทำให้ ไอดีเอส เตือนอยู่แทบตลอดเวลาโดยไม่ได้มีการบุกรุกที่แท้จริง

การเตือนโดยมิได้มีการบุกรุกจริงนั้น อาจจะดูเหมือนว่าไม่ส่งผลเสียหายประการใดและน่าจะเกิดประโยชน์เสียด้วยซ้ำ เพราะจะทำให้ผู้ดูแลระบบมีความตื่นตัวตลอดเวลาแต่ในความเป็น จริงแล้วธรรมชาติของมนุษย์มีแนวโน้มจะละเลยต่อสิ่งเหล่านี้ หากมีการเตือนแล้วไม่มีการบุกรุกจริงอยู่บ่อยครั้งเข้า ความน่าเชื่อถือของไอดีเอสจะลดลงตามลำดับ และเมื่อมีความบุกรุกจริงก็จะไม่ได้ให้ความสนใจเท่าที่ควรและไม่ได้หาทางป้องกันอย่างเหมาะสม นั่นคือไอดีเอสจะกลายเป็นเด็กเลี้ยงแกะที่เวลา

หมาป่าเข้ามาจริงก็ไม่มีผู้ใดรับฟัง ดูเผินๆ อาจจะเหมือนว่ายังดีกว่าถ้าไม่มีไอดีเอสเสียเลย แต่การมีไอดีเอสอยู่ในระบบโดยไม่นำมาปรับแต่งอย่างเหมาะสม และเชื่อมั่นว่าไอดีเอสสามารถจะคอยระแวดระวังและเก็บหลักฐานต่างๆ ไว้ให้นั้นจะทำให้ผู้บริหารระบบนิ่งนอนใจ และคลายความเคร่งครัดในการปฏิบัติงานลง อาจถึงขั้นหย่อนยานกว่าการป้องกันในระดับปกติที่ไม่มีไอดีเอสได้

นอกจากนี้หากปล่อยให้ไอดีเอสมีการเตือนอย่างไม่เหมาะสมจะทำให้เกิดข้อมูลในลักษณะที่เป็นการบุกรุกจริงและการเตือนผิดพลาดผสมกันอยู่ อาจทำให้การเตือนที่เป็นของจริงถูกกลบไปและยากต่อการสังเกต อย่าลืมว่าแฮคเกอร์ที่มีความสามารถจะทิ้งร่องรอยไว้เพียงเล็กน้อยอาจจะมีเพียง 2-3 ร่องรอยเท่านั้นที่ไอดีเอสสามารถตรวจพบได้ หากร่องรอยเหล่านี้ถูกนำไปผสมปะปนกับการตรวจจับอื่นๆ อีกนับพัน ย่อมมีโอกาสสูงที่จะถูกมองเลยไปโดยไม่มีผู้ใดให้ความสนใจ

2.2.5 สรุปความสามารถของระบบตรวจจับผู้บุกรุก

ไอดีเอสเป็นเครื่องมือสำคัญในการรักษาความปลอดภัยในเน็ตเวิร์ก ทำหน้าที่ในเชิงรุก (Proactive) สามารถทำให้ผู้ดูแลระบบสามารถป้องกันภัยคุกคามได้ล่วงหน้าก่อนที่การบุกรุกจะเกิดขึ้นหรือก่อนที่การบุกรุกจะกระทำได้สำเร็จ นอกจากนี้ไอดีเอสยังสามารถช่วยในการเก็บหลักฐานทางอิเล็กทรอนิคส์ของการบุกรุกที่เกิดขึ้น สามารถนำไปวิเคราะห์และสืบค้นผู้กระทำผิดได้ในภายหลัง แต่ทั้งนี้ไอดีเอสมิใช่เครื่องมืออัตโนมัติที่สามารถจับการบุกรุกได้อย่างถูกต้อง 100 เปอร์เซ็นต์การนำไปใช้งานต้องอาศัยความเข้าใจ และการปรับแต่ง อย่างถูกต้องเหมาะสมกับสิ่งแวดล้อมรวมทั้งต้องอาศัยการบำรุงรักษาและตรวจสอบอยู่สม่ำเสมอ

นอกจากนี้ไอดีเอสเองก็มีทั้งข้อดีและข้อเสียในตัวเองซึ่งผู้ที่จะนำไปใช้จะต้องตระหนักให้มาก และต้องคัดเลือกบุคคลากรที่มีความชำนาญและมีความรับผิดชอบที่เหมาะสมให้เป็นผู้ดูแล เพื่อเป็นการป้องกันมิให้ไอดีเอสถูกนำไปใช้ในทางที่ผิดและส่งผลร้ายต่อบุคคลอื่นมากกว่าจะใช้ในการป้องกันระบบของตนเอง

2.3 เทคนิคทางด้านเครือข่ายและโปรโตคอล

2.3.1 ความหมายเบื้องต้นของโปรโตคอล

โปรโตคอล คือ เป็นตัวที่มีหน้าที่สำหรับคอยตกลงระเบียบวิธี ที่กำหนดขึ้นสำหรับสื่อสารข้อมูล โดยสามารถส่งผ่านข้อมูลไปยังปลายทางได้อย่างถูกต้อง เมื่อคอมพิวเตอร์เครื่องหนึ่งต้องการรับส่งข้อมูลกับคอมพิวเตอร์อีกเครื่องหนึ่งที่มีระบบแตกต่างกัน หรือคนละผู้ผลิตเป็นส่งที่ทำให้การติดต่อทำได้ยากมาก จึงต้องมีตัวที่เป็นมาตรฐานส่วนกลาง ที่จำเป็นต้องใช้ในการรับส่งข้อมูล

2.3.2 ความรู้เกี่ยวกับโมเดลเครือข่ายแบบโปรโตคอลทีซีพี/ไอพี(TCP/IP)

โปรโตคอลทีซีพี/ไอพี เป็นโปรโตคอลที่ใช้กันแพร่หลายที่สุด โดยเฉพาะเมื่อถูกนำไปใช้กับเครือข่ายบนอินเตอร์เน็ต ทีซีพี/ไอพี มีการแบ่งโปรโตคอลสื่อสารออกเป็นชั้นๆ โดยจะมีการเรียกลำดับชั้นของทีซีพี/ไอพีว่า ทีซีพี/ไอพี สแต็ก(TCP/IP Stack) โดยทีซีพี/ไอพี สแต็ก มีทั้งหมด 4 ชั้น คือ

2.3.2.1 ลำดับชั้นของโมเดลเครือข่ายแบบโปรโตคอลทีซีพี/ไอพี

1) ลำดับชั้นที่ 4: โปรเซสเลเยอร์(Process Layer)

จะเป็นแอพพลิเคชัน โปรโตคอล(Application Protocol) ที่ทำหน้าที่เชื่อมต่อกับผู้ใช้ และให้บริการต่างๆ เช่น เอฟทีพี(FTP) , เทลเน็ต(Telnet) , เอสเอ็นเอ็มพี(SNMP) เป็นต้น

2) ลำดับชั้นที่ 3: โฮสต์ทูโฮสต์เลเยอร์(Host to Host Layer)

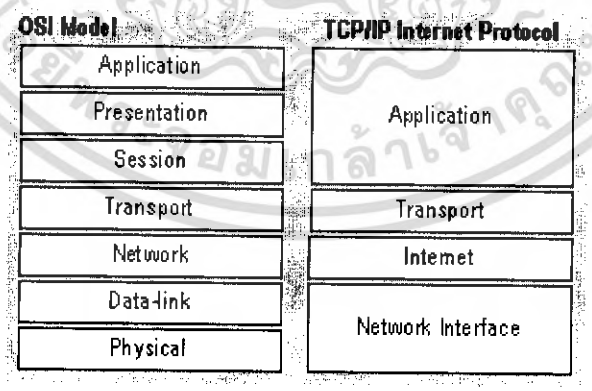
จะเป็นทีซีพี หรือยูดีพี(UDP) ที่ทำหน้าที่คล้ายกับเลเยอร์ที่สี่ของแบบจำลองโอเอสไอ(OSI Model) ควบคุมการรับส่งข้อมูลจากปลายด้านส่งถึงปลายด้านรับข้อมูล

3) ลำดับชั้นที่ 2: อินเตอร์เน็ตเวิร์ก(Internetwork)

ได้แก่ ส่วนของโปรโตคอลไอพี(IP) ซึ่งทำหน้าที่คล้ายกับเลเยอร์ที่สามของแบบจำลองโอเอสไอเชื่อมต่อคอมพิวเตอร์ของด้านรับ และด้านส่งเข้าหากันผ่านระบบเครือข่ายพร้อมทั้งเลือก หรือกำหนดเส้นทางที่จะใช้ในการรับส่งข้อมูลระหว่างกัน และส่งผ่านข้อมูลที่ได้รับไปยังอุปกรณ์ในเครือข่ายต่างๆ จนกระทั่งถึงปลายทาง ข้อมูลที่รับส่งกันจะอยู่ในรูปแพ็กเก็ต หรือเฟรมข้อมูล

4) ลำดับชั้นที่ 1: เน็ตเวิร์กอินเตอร์เฟซ(Network Interface)

ทำหน้าที่คล้ายกับเลเยอร์ที่หนึ่งและสองของแบบจำลองโอเอสไอ คือเชื่อมต่อการรับส่งข้อมูลในระดับฮาร์ดแวร์ โดยทำหน้าที่แปลคำสั่งนั้นๆ ให้เป็นคำสั่งควบคุมฮาร์ดแวร์ และแก้ไขข้อผิดพลาดที่ตรวจพบนั้น ซึ่งที่ใช้กันอยู่จะเป็นตามมาตรฐาน IEEE ข้อมูลในชั้นนี้จะอยู่ในรูปเฟรม



รูปที่ 2.5 โมเดลเครือข่ายแบบโปรโตคอลทีซีพี/ไอพี

2.3.2.2 หลักการทำงานของโมเดลเครือข่ายแบบโปรโตคอลทีซีพี/ไอพี

แนวคิดหลักของระบบเครือข่ายคอมพิวเตอร์ ก็คือ การเชื่อมโยงอุปกรณ์เข้าด้วยกัน ไม่ว่าจะเป็นเครื่องเซิร์ฟเวอร์ และอุปกรณ์ในเครือข่ายอื่นๆ จึงจะต้องมีการไอพีแอดเดรส(IP Address) ที่เป็นค่าที่อยู่ไว้อ้างอิงถึงที่ที่จะส่งข้อมูลผ่าน

1) ไอพีแอดเดรส

ไอพีแอดเดรสถูกกำหนดขึ้นมาให้เป็นหมายเลขอ้างอิงประจำตัวของอุปกรณ์ต่างๆ ที่เชื่อมต่ออยู่ในเครือข่ายอินเตอร์เน็ต โดยการกำหนดไอพีแอดเดรสให้แต่ละเครื่อง หรือแต่ละอุปกรณ์นี้จะต้องไม่ซ้ำกัน ซึ่งไอพีแอดเดรสนี้จะไม่ถูกผูกติดกับตัวฮาร์ดแวร์ จึงสามารถกำหนดใหม่ หรือแก้ไขเปลี่ยนแปลงได้เมื่อมีการเปลี่ยนแปลงตัวฮาร์ดแวร์

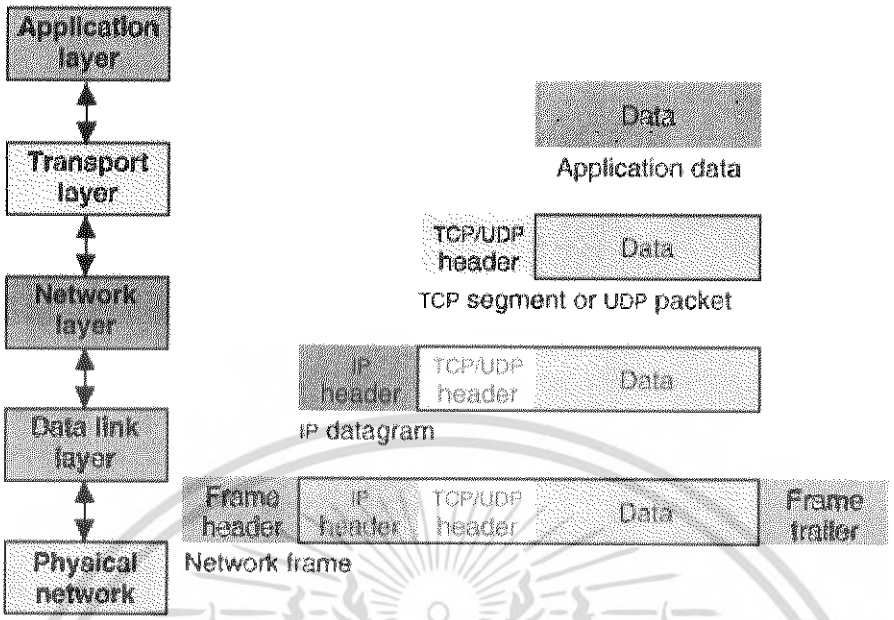
โปรโตคอลไอพีจำเป็นต้องอาศัยไอพีแอดเดรสเพื่อระบุถึงอุปกรณ์ต่างๆ ที่อยู่ในเครือข่าย ไม่ว่าจะเป็นเว็บเซิร์ฟเวอร์ เมล์เซิร์ฟเวอร์ อุปกรณ์เราเตอร์ ไอพีแอดเดรสจะเป็นค่าตัวเลขขนาด 32 บิต ถูกแบ่งออกเป็น 4 ส่วน ส่วนละ 8 บิต และถูกคั่นแต่ละส่วนด้วยเครื่องหมายจุด(.)

2) แพ็กเก็ตข้อมูล (Data Packet)

เมื่อมีการส่งข้อมูลบนเครือข่ายอินเตอร์เน็ตนั้น ที่แสดงออกมาทางบราวเซอร์ ข้อมูลจำเป็นต้องมีการทำให้มีขนาดเล็กลง โดยแบ่งออกเป็นย่อยๆ เรียกว่าแพ็กเก็ตข้อมูลหรือดาต้าแกรม (Datagram) โดยข้อมูลจะถูกแบ่งออกเป็นย่อยๆ มีประโยชน์ คือ ทำให้เครือข่ายนั้นสามารถรองรับการติดต่อ และรับส่งข้อมูลกันได้อย่างราบรื่นไม่ติดขัด หรือถ้าเกิดปัญหาเครือข่ายทำงานช้า เมื่อมีการรับส่งข้อมูลขนาดใหญ่ เนื่องจากสายสัญญาณเชื่อมโยงเป็นสื่อที่ต้องแบ่งกันใช้ นอกจากนี้การแบ่งข้อมูลออกเป็นส่วนย่อยๆ ยังทำให้สามารถเพิ่มกระบวนการตรวจทานความถูกต้องของข้อมูลที่ปลายทาง และแก้ไขเมื่อข้อมูลผิดพลาด หรือดกหล่นได้โดยง่ายอีกด้วย

3) การห่อหุ้ม (Encapsulation)

การผนึกข้อมูลหนึ่งให้ไปเป็นอีกรูปแบบหนึ่งนี้ จะเป็นกลไกที่สำคัญของการใช้งานโปรโตคอลทีซีพี/ไอพีมาก โดยที่ขบวนการที่ใช้จะมีขั้นตอนคร่าวๆ ดังรูป



รูปที่ 2.6 Header ในแต่ละลำดับชั้นของ TCP/IP

2.4 ลักษณะของโปรโตคอลที่ใช้ในโปรแกรมประยุกต์เพียร์ทูเพียร์

โปรแกรมประยุกต์เพียร์ทูเพียร์นั้นมีหลากหลายโปรแกรมมาก เช่น Bitcomet , LimeWire , Kazaa เป็นต้น ซึ่งแต่ละโปรแกรมมีการใช้ซิกเนเจอร์(Signature) ที่มีลักษณะเฉพาะสำหรับแต่ละโปรโตคอลที่โปรแกรมนั้นใช้เป็นช่องทางในการติดต่อสื่อสาร ในที่นี้จะยกตัวอย่างซิกเนเจอร์ของแพ็กเก็ตของโปรโตคอลที่มีความนิยมมากกว่าตัวอื่นๆ คือ Gnutella , eDonkey , DirectConnect , BitTorrent และ Kazaa

1) Gnutella Protocol

จะมีเฮดเดอร์ของคำร้องขอ(request header) ดังนี้

GET /get/<File Index>/<File Name>

/HTTP/1.0 \r \n

Connection: Keep-Alive\r\n

Range: byte=0-\r\n

User-Agent: <Name>\r\n

\r\n

ส่วนเฮดเดอร์ของคำตอบกลับ(response header) จะเป็นดังนี้

HTTP 200 OK\r\n

Server: <Name>\r\n

Content-type: \r\n

Content-length: \r\n

\r\n

ฉะนั้นถ้าเฮดเดอร์ใดมีสตริงต่อไปนี้ ถือว่าเป็นแพ็กเก็ตของ Gnutella

- เฮดเดอร์ที่ขึ้นต้นด้วยคำว่า GET หรือ HTTP หรือ GNUTELLA

- ถ้าเฮดเดอร์ใดที่ขึ้นต้นด้วย GET หรือ HTTP จะต้องตามด้วยสตริงต่อไปนี้

- User-Agent: <Name>

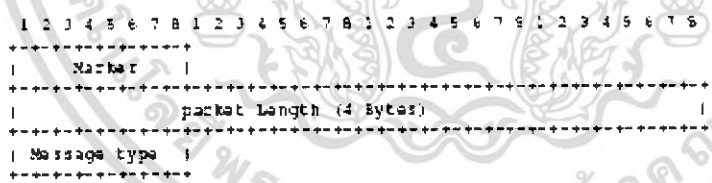
- UserAgent: <Name>

- Server: <Name>

โดยที่ <Name> อาจจะเป็น LimeWire, Bear-Share, Gnucleus, MorpheusOS, XoloX, MorpheusPE, gtkgnutella,Acquisition, Mutella-0.4.1, MyNapster, Mutella-0.4.1, MyNapster, Mutella-0.4, Qtella, AquaLime, NapShare,Comebaek, Go, PHEX, SwapNut, Mutella-0.4.0, Sharcaza,Mutella-0.3.9b, Morpheus, FreeWire, Openext, Mutella-0.3.3,Phex.

2) eDonkey Protocol

เฮดเดอร์ของมันจะต่อจากเฮดเดอร์ของทีซีพีมีลักษณะดังนี้คือ



รูปที่ 2.7 ลักษณะเฮดเดอร์ของ eDonkey Protocol

- 1 ไบต์แรกจะเป็นมาร์กเกอร์ซึ่งมีค่า 0xE3 (เลขฐาน 16)

- 4 ไบต์ต่อมาเป็นขนาดของแมสเสจ(message)

- 1 ไบต์สุดท้ายเป็นแมสเสจไอดี(message ID)ที่เป็นเอกลักษณ์ไม่ซ้ำ

3) DirectConnect Protocol

ลักษณะเฮดเดอร์ของมัน จะเป็นดังนี้

$command_type field1 field2 ...|

ฉะนั้นถ้าเฮดเดอร์ใดมีสตริงต่อไปนี้ ถือว่าเป็นแพ็กเก็ตของ DirectConnect

- ไบต์แรกที่ต่อจากเฮดเดอร์ของโปรโตคอลทีซีพี/ไอพีเป็น "$" และ ไบต์สุดท้ายเป็น "|"

- ตัวที่ต่อจาก "$" จะเป็น command_type ได้แก่ MyNick,Lock, Key, Direction, GetListLen, ListLen, MaxedOut, Error,Send, Get, FileLength, Canceled, HubName, ValidateNick, ValidateDenide,GetPass, Mypass, BadPass, Version, Hello, Logedin,MyINFO, GetINFO, GetNickList, NickList, OpList, To, Connect-ToMe, MultiConnectToMe, RevConnectToMe, Search, MultiSearch,SR, Kick, OpForceMove, ForceMove, Quit.

4) BitTorrent Protocol

เฮดเดอร์ของมันมีรูปแบบดังนี้

<a character(1 byte)><a string(19 byte)>

ฉะนั้นถ้าเฮดเดอร์ใดมีสตริงต่อไปนี้ ถือว่าเป็นแพ็กเก็ตของ BitTorrent

- ไบต์แรกจะเป็นตัวอักษร 19 (0x13)

- 19 ไบต์ต่อมาจะมีคำว่า "BitTorrent Protocol" อยู่

5) Kazaa Protocol

เฮดเดอร์ของคำร้องขอ จะมีลักษณะดังนี้

GET /.files HTTP/1.1\r\n

Host: IP address/port\r\n

UserAgent: KazaaClient\r\n

X-Kazaa-Username: \r\n

X-Kazaa-Network: KaZaA\r\n

X-Kazaa-IP: \r\n

X-Kazaa-SupernodeIP: \r\n

เฮดเดอร์ของคำตอบกลับจะมีลักษณะดังนี้

HTTP/1.1 200 OK\r\n

Content-Length: \r\n

Server: KazaaClient\r\n

X-Kazaa-Username: \r\n

X-Kazaa-Network: \r\n

X-Kazaa-IP: \r\n

X-Kazaa-SupernodeIP: \r\n

Content-Type: \r\n

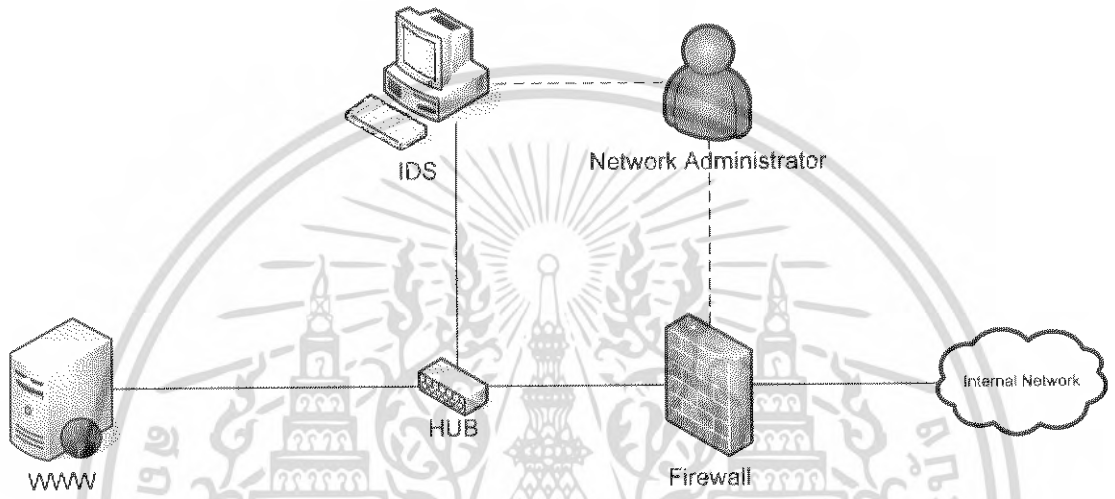
ฉะนั้นการที่จะระบุว่าเป็นเฮดเดอร์ของ Kazaa Protocol จะต้องมีลักษณะดังนี้

- มีคำว่า GET หรือ HTTP
- ต้องมีฟิลด์ที่เป็นคำว่า X-Kazaa

บทที่ 3
วิธีดำเนินการวิจัย

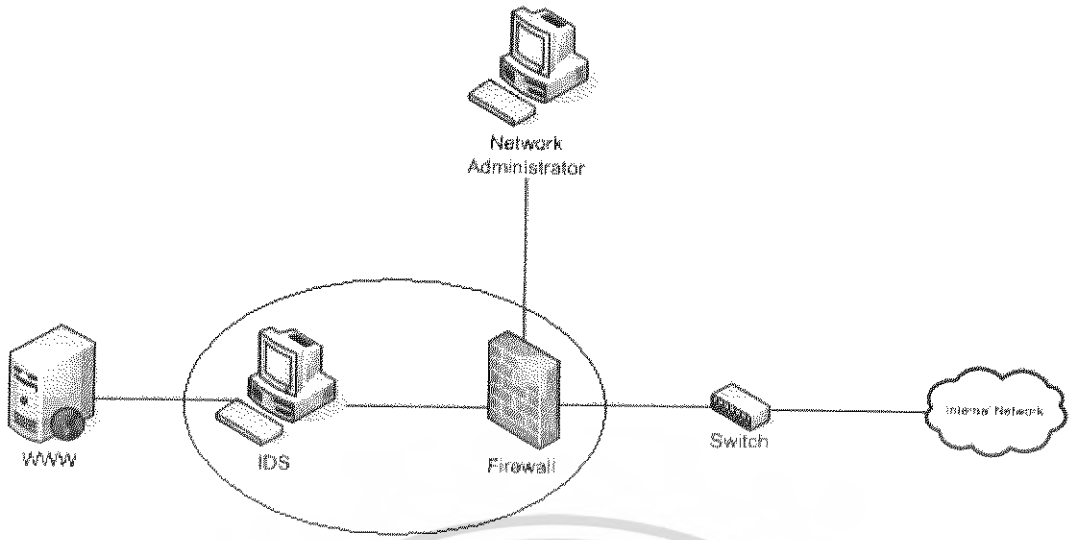
3.1 รูปแบบโครงสร้างของระบบตรวจจับผู้บุกรุกที่ใช้ในการดำเนินการวิจัย

ระบบตรวจจับผู้บุกรุกจะมีลักษณะโครงสร้างอยู่ 2 รูปแบบ คือ อินไลน์(In-Line Architecture) และ วันอาร์ม(One-Arm Architecture) ดังรูป



รูปที่ 3.1 One-Arm Architecture

ในการทำงานแบบวันอาร์ม แพ็กเก็ตข้อมูลจะวิ่งผ่านสวิตช์(Switch)หรือฮับ(Hub) และถูกทำสำเนาส่งไปให้ ระบบตรวจจับผู้บุกรุก ตรวจดูแพ็กเก็ตนั้นว่ามีซิกเนเจอร์ตรงกับซิกเนเจอร์ของเพียร์ทูเพียร์หรือไม่ ถ้าตรงก็จะสั่งให้ไฟร์วอลล์ (Firewall) ให้บล็อกไอพี(IP) นั้นในระยะเวลาหนึ่ง ดังภาพที่ 3.1



รูปที่ 3.2 In-Line Architecture

ในการทำงานแบบอินไลน์ แพ็กเก็ตจะวิ่งจากเครือข่ายอินเตอร์เน็ตไปยังเครือข่ายแลนหรือจากเครือข่ายแลนไปยังเครือข่ายอินเตอร์เน็ตโดยผ่านระบบตรวจจับผู้บุกรุกที่ทำงานร่วมกับไฟร์วอลล์หรือสวิตช์ ถ้าหากพบว่ามีแพ็กเก็ตที่มีซิกเนเจอร์เป็นแพ็กเก็ตของเพียร์ทูเพียร์ จะสั่งให้ไฟร์วอลล์บล็อกหมายเลขไอพีนั้นในระยะเวลาหนึ่ง

ในการวิจัยเราได้ใช้ระบบตรวจจับผู้บุกรุกแบบวันอาร์ม เพราะข้อดีของมันคือ ถ้าหากตัวระบบตรวจจับผู้บุกรุกเกิดความเสียหายไม่สามารถทำงานต่อได้ เครือข่ายแลนและเครือข่ายอินเตอร์เน็ตยังคงสามารถติดต่อสื่อสารกันได้ แต่ถ้าเป็นแบบอินไลน์เมื่อระบบตรวจจับผู้บุกรุกเสียหายเครือข่ายแลนและเครือข่ายอินเตอร์เน็ตจะไม่สามารถติดต่อสื่อสารกันได้

3.2 ลักษณะของโปรแกรมที่ทำการทดลอง

3.2.1 โปรแกรมที่ใช้ในการตรวจจับการบุกรุก

โปรแกรมที่นำมาใช้คือโปรแกรมสนอร์ต(Snort) ซึ่งเป็นระบบตรวจจับผู้บุกรุกแบบ โอเพน ซอร์ซ สามารถดาวน์โหลดโปรแกรมสนอร์ตเวอร์ชันล่าสุดได้จาก http://www.snort.org/

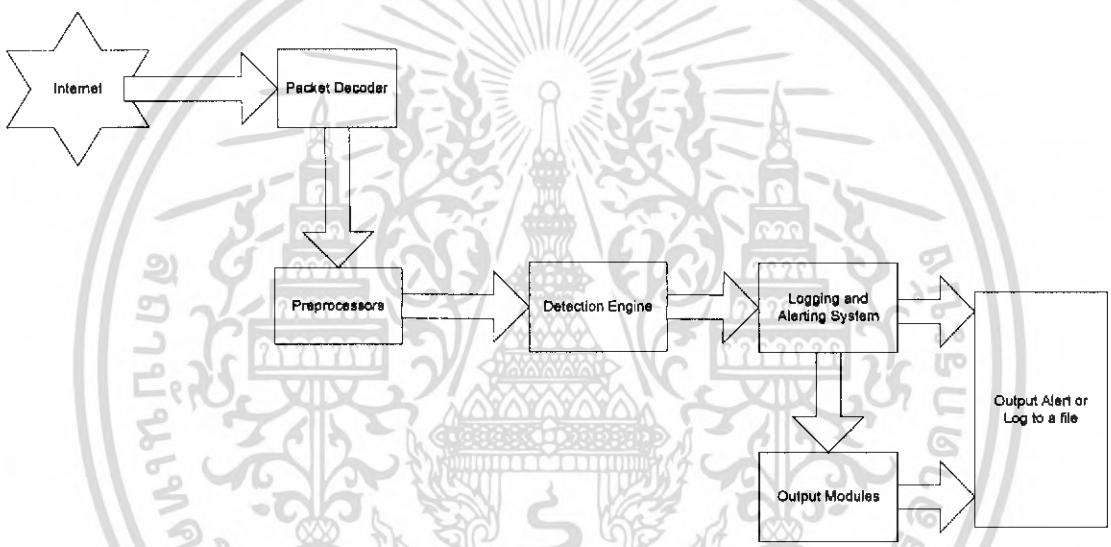
เหตุผลที่เลือกใช้โปรแกรมสนอร์ต เนื่องจากมีประสิทธิภาพในการตรวจจับผู้บุกรุกได้ดีเหมาะในการนำมาใช้ในการพัฒนาระบบ เป็นฟรีแวร์(Freeware) สามารถนำไปใช้หรือพัฒนาต่อโดยไม่เสียค่าใช้จ่ายใดๆ มีผู้ใช้เป็นจำนวนมากทำให้มีเอกสารอ้างอิงในการแก้ปัญหาเยอะ

สนอร์ต มีการทำงานแบบเอ็นไอดีเอส (NIDS : Network Intrusion Detection System) คือจะเป็นการทำงานโดยตรวจสอบข้อมูลที่วิ่งอยู่ในเน็ตเวิร์ก แบบเรียลไทม์(Realtime) ว่าตรงกับรูปแบบของกฎ (Rule) ที่เก็บไว้ในฐานข้อมูลหรือไม่ ในการบุกรุกจะมีรูปแบบสัญญาณ (Signature) หลายประเภท ข้อมูลเกี่ยวกับรูปแบบเหล่านี้จะถูกใช้ในการสร้างกฎสนอร์ต (Snort Rules) เราใช้

รูปแบบการบุกรุก ตรวจหาว่ามีใครพยายามหาประโยชน์จากจุดที่เป็นช่องโหว่ รูปแบบเหล่านี้อาจอยู่ในส่วนเฮดเดอร์(Header) ของแพ็กเก็ต หรือในเพย์โหลด (Payload) ระบบตรวจจับสนอร์ต จะมีพื้นฐานมาจากกฎ ซึ่งกฎเหล่านี้มาจากรูปแบบการบุกรุก กฎของสนอร์ต ตรวจสอบได้หลายส่วนของแพ็กเก็ตข้อมูล สนอร์ตสามารถวิเคราะห์เฮดเดอร์ของชั้นเลเยอร์ที่ 3 , 4 (แอพพลิเคชันเลเยอร์) ได้ กฎจะถูกประยุกต์ให้ทันสมัยกับทุกแพ็กเก็ต กฎอาจใช้ในการสร้างข้อความเตือน (Alert Message) , บันทึกข้อความ (Log Message) หรือในรูปแบบของสนอร์ต การผ่าน Pass) ของแพ็กเก็ตข้อมูล

3.2.1.1 ส่วนประกอบของโปรแกรมสนอร์ต

ส่วนประกอบของสนอร์ตเป็นดังรูป



รูปที่ 3.3 ส่วนประกอบของสนอร์ต

1) ตัวถอดแพ็กเก็ต (Packet Decoder) เป็นส่วนที่ทำหน้าที่เกี่ยวกับการนำแพ็กเก็ตจากหลายๆ โปรโตคอลและเตรียมแพ็กเก็ตโดยตัดเอาแพ็กเก็ตส่วนที่พรีโพรเซสเซอร์รู้จักออกมาเพื่อทำ การพรีโพรเซส (Preprocess) หรือส่งไปยังดีเทคชันเอนจิน (Detection Engine)

2) พรีโพรเซสเซอร์ (Preprocessor) เป็นส่วนประกอบหรือปลั๊กอิน(Plug-In) ที่ใช้ในสนอร์ตเพื่อการจัดเรียงหรือเปลี่ยนแปลงแพ็กเก็ตข้อมูล ก่อนที่ ดีเทคชันเอนจิน จะทำการค้นหาแพ็กเก็ตที่เป็นการบุกรุกพรีโพรเซสเซอร์บางตัว จะทำการตรวจสอบโดยการหาเฮดเดอร์ของแพ็กเก็ตที่ไม่ปกติและทำการเตือนขึ้นมา พรีโพรเซสเซอร์เป็นสิ่งที่สำคัญอย่างยิ่งสำหรับระบบตรวจจับผู้บุกรุก คือ เตรียมแพ็กเก็ตข้อมูลเพื่อการตรวจสอบกับกฎที่อยู่ในดีเทคชันเอนจิน ตัวอย่างเช่น แฮค

เกอร์โจมตีในเฮชทีทีพีแพ็กเก็ต (HTTP packet) วิธีแก้คือเราสร้างกฎที่ค้นหารูปแบบของคำว่า "scripts/iisadmin" (แพ็กเก็ตเพย์โหลด (Packet payload)) แต่ถ้าแฮกเกอร์ เปลี่ยนสตริงเช่น

- "scripts/./iisadmin"
- "scripts/examples/../iisadmin"
- "scripts\iisadmin"
- "scripts/.\iisadmin"

ระบบตรวจจับผู้บุกรุกจะไม่สามารถตรวจจับได้เลยจึงต้องมีพรีโพรเซสเซอร์ที่ช่วยในการจัดเรียงสตริงเพื่อตรวจจับการบุกรุกประเภทนี้

3) ดีเทคชันเอนจิน เป็นส่วนที่ทำหน้าที่ในการใช้ กฎในการตรวจจับแพ็กเก็ต ถ้าแพ็กเก็ตเป็นการบุกรุก โดยกฎจะทำการอ่านโครงสร้างภายในของข้อมูล ถ้ามีแพ็กเก็ตที่ตรงกับกฎ จะมีการกระทำเกิดขึ้นโดยอาจจะเป็นการบันทึกแพ็กเก็ต (Log Packet) หรือ การเตือน (Alert) ถ้าไม่ใช่การบุกรุกแพ็กเก็ตนั้น จะถูกพาส(Pass)ไป ในส่วนของดีเทคชันเอนจิน ระยะเวลาที่ใช้ในการตรวจจับจะขึ้นอยู่กับ

- ประสิทธิภาพของเครื่องคอมพิวเตอร์ที่ใช้โปรแกรมสนอร์ต
- จำนวนกฎที่กำหนด
- ความเร็วของอินเทอร์นอลบัส (Internal Bus) ของเครื่องคอมพิวเตอร์
- จำนวนการจราจร (Traffic) ที่อยู่บนเครือข่าย

4) ระบบการบันทึกและเตือน (Logging and Alerting System) อาจจะบันทึกหรือแจ้งเตือนก็ได้ โดยการบันทึก จะเก็บเป็นเท็กซ์ไฟล์ หรือฐานข้อมูล ก็ได้

5) เอาท์พุทโมดูล (Output Modules) ควบคุมประเภทของการสร้างเอาท์พุทโดยจะเป็นการบันทึกหรือแจ้งเตือนคือ

- ไฟล์บันทึก (Log file)
- ส่งเอสเอ็นเอ็มพีแทร็ป (SNMP (Simple Network Management Protocol) Traps)
- ไฟล์บันทึก ไปยังฐานข้อมูล
- สร้างเอ็กซ์เอ็มแอล (XML Output)
- ปรับเปลี่ยนค่าของเราเตอร์ (Router) และไฟร์วอลล์ (Firewall)
- ส่งข้อความเอสเอ็มบี (SMB :Server Message Block) ไปยังเครื่องคอมพิวเตอร์ที่เป็นวินโดวส์

ชื่อ	คำอธิบาย
Packet Decoder	เตรียมแพ็กเก็ตสำหรับประมวลผล
Preprocessor or Input Plugins	ใช้ในการปรับเฮดเดอร์ของโปรโตคอลให้อยู่ในสภาวะปกติ , ตรวจสอบแพ็กเก็ตที่ผิดปกติ , รวมแพ็กเก็ตและทีซีพีสตรีม(TCP Stream ขึ้นมาใหม่
Detection Engine	ตรวจดูว่ากฎตรงกับแพ็กเก็ตใด
Logging and Alerting System	สร้างอเลิร์ตและล็อกแมสเสจ
Output Modules	ประมวลผลอเลิร์ตและล็อกเพื่อสร้างเอาต์พุตสุดท้าย

ตารางที่ 3.1 ส่วนประกอบของสนอร์ต

3.2.1.2 สิ่งที่จำเป็นในการติดตั้งโปรแกรมสนอร์ต

- Libpcab library (ftp://ftp.ee.lbl.gov/libpcap.tar.Z)
- C Compiler ซึ่งปกติจะมีติดตั้งให้ในทุก OS อยู่แล้ว หรืออาจจะพิจารณาใช้ GNU C (ftp://ftp.gnu.org/gnu/gcc/)
- Utility ที่ใช้สำหรับขยายไฟล์ เช่น GZIP (ftp://ftp.gnu.org/gnu/gzip/)
- MD5 Cryptographic Checksum Program เพื่อใช้เช็คว่าซอร์ซโค้ด(Source code) ที่ได้มานั้นไม่ได้ถูกแก้ไขไปก่อนหน้านี้โดยผู้ไม่ประสงค์ดี

3.2.1.3 การติดตั้งโปรแกรม

ขยายไฟล์ที่ดาวน์โหลดมา

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

ย้ายไดเร็คทอรีไปยังเป้าหมายที่ขยายไป จากนั้นใช้คำสั่ง

```
#./configure (ใช้ ./configure --help เพื่อดู Option ทั้งหมด)
#make
#make install
```

โดยปกติสนอร์ตจะเก็บข้อมูลล็อกอยู่ในรูปของล็อกไฟล์คล้ายๆ กับ Syslog แต่เราสามารถสั่งให้สนอร์ตเก็บข้อมูลที่ต้องการไว้ใน Database เช่น MySQL, PosgrestSQL หรือ MSSQL ได้ เพื่อให้สามารถดึงข้อมูลมาตรวจสอบได้สะดวกยิ่งขึ้นผ่านทางปลั๊กอิน เช่น ACID ทั้งนี้จะมีตัวอย่างการคอมไพล์(Compile) เพื่อให้ใช้งานกับ MySQL อีกครั้งในบทความฉบับต่อไป

Create Directory for Snort Log Files

สร้างไดเร็คทอรีเพื่อเก็บล็อกไฟล์ของสนอร์ตทั้งหมดแยกต่างหาก และควรป้องกันไม่ให้บุคคลอื่น Access เข้ามาที่ไดเร็คทอรีนั้นๆ โดยปกติแล้วจะสร้างไว้ที่ /var/log/snort

#mkdir /var/log/snort

#chmod 700 /var/log/snort

ทดสอบโปรแกรม

ทดลองรันคำสั่ง snort -? เพื่อแสดง Help ของสนอร์ตทั้งหมด

สร้างไฟล์ Configuration และ Rules

จริงๆ แล้ว ขั้นตอนนี้จะไม่ถือว่าเป็นการสร้างเป็นเพียงการประกอบข้อมูลที่สนอร์ตให้เรามาแล้วนั้น นำมาจัดให้เป็นระเบียบเรียบร้อยเท่านั้นเอง ซึ่งอาจจะไม่จำเป็น ในที่นี้จะสร้างโฟลเดอร์ขึ้นมาที่ /etc/snort เพื่อใช้เก็บ Configuration และ Rules Files ของสนอร์ตไว้ต่างหาก

#mkdir /etc/snort

จากนั้นให้ก๊อปปี้ข้อมูล Configuration และ Rules Files จากซอร์ซของสนอร์ต

#cd /usr/local/src/snort

#cp snort.conf /etc/snort

#cp *.rules /etc.snort

#cp classification.config /etc/snort

แก้ไข /etc/snort/snort.conf

เราจะใช้ไฟล์ snort.conf เป็นไฟล์หลักในการรันสนอร์ตโดยจำเป็นต้องแก้ไขข้อมูลในบางส่วนดังต่อไปนี้ (vi snort.conf)

แก้ไข HOME_NET ให้เป็นเน็ตเวิร์ก แอดเดรส(Network Address) ของเครือข่ายที่ต้องการมอนิเตอร์(Monitor) เช่น var HOME_NET 10.10.10.0/24

แก้ไขค่า Network IP Address อื่นให้ตรงกับความต้องการ เช่น SMTP , SQL_SERVERS

ค่าที่ควรแก้ไขคือ VAR DNS_SERVERS แก้ให้เป็น DNS Server ที่ใช้ภายในหน่วยงาน เพื่อป้องกัน Fault Alarm

สำหรับพารามิเตอร์(Parameter) อื่นๆ นั้น ท่านสามารถหาข้อมูลเพิ่มเติมได้ที่ www.snort.org พร้อมกับคู่มือการเขียนกฎ (โดยปกติแล้ว ไม่มีความจำเป็นต้องเขียนกฎเอง เพียงแต่หมั่นติดตามข่าวกฎใหม่ๆ ที่ www.snort.org เท่านั้นเอง)

การรันสนอร์ต (Daemon)

ทดลองรัน /usr/local/bin/snort -c /etc/snort/snort.conf ถ้าไม่มีข้อผิดพลาด(Error) ใดๆ แสดงว่าสามารถใช้งานได้ เพียงแต่การใช้งานจริงนั้นจะรันใน Daemon Mode โดยจะใช้คำสั่งดังนี้

#/usr/local/bin/snort -D -c /etc/snort/snort.conf

ถ้าต้องการให้สนอร์ตรันใน Daemon Mode ทุกครั้งที่บูตเครื่องขึ้นมาก็ให้แก้ไขไฟล์ /etc/rc.local แล้วใส่คำสั่งด้านบน เพื่อให้สนอร์ททำงานเมื่อมีการบูตเครื่องใหม่

3.2.2 โปรแกรมเพียร์ทูเพียร์ประยุกต์(Peer-to-Peer Application)

โปรแกรมเพียร์ทูเพียร์(เช่น BitTorrent , BitComet) ใช้สำหรับกระจาย หรือเผยแพร่ไฟล์โดยเฉพาะไฟล์ใหญ่ๆ แบบเครื่องสู่เครื่องได้อย่างรวดเร็วมาก ซึ่งการกระทำเช่นนี้เป็นการเพิ่มความคับคั่งให้ระบบเครือข่ายส่วนรวม ซึ่งมีหลายโปรแกรมให้เลือกใช้ แต่เราจะใช้โปรแกรมบิตโคเม็ต 0.71 , อีมุลล์และแซดเพียร์ทูเพียร์

3.2.3 โปรแกรมที่ใช้ตรวจสอบรายละเอียดของแพ็กเก็ต

ส่วนนี้จะใช้โปรแกรมอีเทอเรียล (Ethereal) ซึ่งเป็นฟรีแวร์ที่มีประสิทธิภาพสูงใช้ตรวจจับทราฟฟิกแบบเรียลไทม์ โดยสามารถตรวจจับและถอดรหัสโปรโตคอลต่าง ๆ ได้มากมายถึง 400 โปรโตคอล เพื่อนำมาวิเคราะห์เครือข่ายหรือเน็ตเวิร์ก ทราฟฟิก (Network Traffic นำแพ็กเก็ตที่ดักจับได้มาทำการถอดรหัส และแสดงผลข้อมูลออกมาในรูปแบบที่สามารถอ่านและเข้าใจได้ง่าย) สามารถทำงานได้ทั้งบนวินโดวส์(Windows) หรือ ยูนิกซ์(Unix) รวมทั้งลีนุกซ์(Linux)

3.3 การทำงานของโปรแกรมสนอร์ต

3.3.1 กฎ(rules)

สนอร์ตจะเขียนกฎในไวยากรณ์ (Syntax) ที่เข้าใจง่าย กฎส่วนใหญ่จะเขียนบรรทัดเดียว แต่เราก็สามารถเขียนได้หลายบรรทัดโดยใช้ \ (backslash) คั่นเมื่อจบหนึ่งบรรทัด กฎปกติจะอยู่ในไฟล์คอนฟิกเกอร์เรชัน (Configuration File : snort .conf)

ตัวอย่างกฎ

```
alert ip any any -> any any ( msg : " IP Packet detected " ; )
```

- คำว่า " alert " แสดงให้เห็นว่ากฎนี้จะสร้างข้อความเตือน (Alert Message) เมื่อเจอแพ็กเก็ตแบบเดียวกับที่เราสร้างเงื่อนไขที่เป็นมาตรฐานไว้ ซึ่งเงื่อนไขที่เป็นมาตรฐานจะนิยามด้วยคำที่ตามมา

- ส่วน " ip " แสดงให้เห็นว่ากฎนี้ ใช้ได้กับทุกไอพีแพ็กเก็ต (IP Packets)

- " any " แรกถูกใช้สำหรับไอพีแอดเดรสต้นทาง (Source IP Address)

- " any " ตัวที่สองใช้สำหรับหมายเลขพอร์ต เพราะหมายเลขพอร์ด ไม่เกี่ยวกับไอพีเลเยอร์

- เครื่องหมาย -> แสดงทิศทางของแพ็กเก็ต

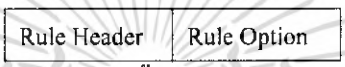
- " any " ตัวที่สามถูกใช้สำหรับไอพีแอดเดรสปลายทาง (Destination IP Address)

- " any " ตัวที่สี่ถูกใช้สำหรับพอร์ตปลายทาง (Destination Port)

- ส่วนสุดท้าย เป็นส่วนออปชันของกฎ (Rule Options) และบรรจุข้อความ ที่ไว้แจ้งเตือนตอนเกิดการแจ้งเตือน

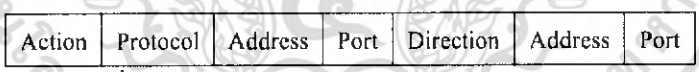
3.3.2 โครงสร้างของกฎ

กฎสนอร์ต ทุกอันจะมี 2 ส่วนลอจิคอล คือ เฮดเดอร์ของกฎ (Rule Header) และส่วนออปชันของกฎ (Rule Options) ดังรูป



รูปที่ 3.4 โครงสร้างพื้นฐานของกฎสนอร์ต

1) เฮดเดอร์ของกฎ บรรจุข้อมูลเกี่ยวกับการกระทำ (Action) ที่กฎจะได้รับ รวมถึงเงื่อนไขที่เป็นมาตรฐานสำหรับไว้จับคู่กันระหว่างกฎกับแพ็กเก็ตข้อมูล ส่วนออปชันของกฎ บรรจุข้อความแจ้งเตือนและข้อมูลที่เกี่ยวกับส่วนแพ็กเก็ต ที่จะถูกใช้ในการสร้างข้อความแจ้งเตือน และยังเพิ่มเงื่อนไขที่เป็นมาตรฐานสำหรับไว้จับคู่กันระหว่างกฎกับแพ็กเก็ตข้อมูล กฎหนึ่งสามารถตรวจพบการบุกรุกได้ตั้งแต่หนึ่งถึงหลายประเภท กฎที่ฉลาดจะประยุกต์ใช้ได้กับหลาย รูปแบบการบุกรุกโครงสร้างทั่วไปของเฮดเดอร์ของกฎสนอร์ต เป็นดังนี้



รูปที่ 3.5 โครงสร้างของเฮดเดอร์ของกฎสนอร์ต

ส่วนแอ็คชัน จะพิจารณาว่าทำการกระทำแบบไหน เมื่อตรงกับเงื่อนไขที่เป็นมาตรฐาน และ กฎตรงกับแพ็กเก็ตข้อมูล โดยปกติการกระทำจะสร้างการแจ้งเตือนหรือบันทึกข้อความหรือเรียกกฎอื่น

ส่วนโปรโตคอล กฎหรือแพ็กเก็ตจะประยุกต์ใช้กับสำหรับโปรโตคอลนั้นๆ โปรโตคอลที่ใช้เช่นไอพี , ไอซีเอ็มพี , ยูดีพี ฯลฯ

ส่วนแอดเดรส จะกำหนดแอดเดรสต้นทางและปลายทาง แอดเดรสต้นทางและปลายทางจะพิจารณาจากไดเรกชันฟิลด์ (Direction Field) ตัวอย่างเช่น ถ้าไดเรกชันฟิลด์ เป็น -> ตำแหน่งด้านซ้ายจะเป็นต้นทางส่วนด้านขวาเป็นปลายทาง

ส่วนพอร์ต ในกรณีของทีซีพีหรือยูดีพี พอร์ตจะใช้ในการพิจารณาต้นทางและปลายทาง ส่วนในกรณีของไอพีและไอซีเอ็มพี หมายเลขพอร์ตไม่มีความสำคัญ

ส่วนทิศทาง (Direction) ใช้พิจารณาแอดเดรสและพอร์ตว่า อันไหนเป็นต้นทางอันไหนเป็นปลายทาง

ตัวอย่าง พิจารณากฎ ที่จะสร้างข้อความแจ้งเตือนเมื่อพบไอซีเอ็มพีปิงแพ็กเก็ต (ICMP Ping Packet) ด้วย ทีทีแอล (TTL)= 100

alert icmp any any -> any any (msg: " Ping with TTL=100 " ; ttl :100 ;)

- แอคชัน ในกฎนี้แอคชันคือการแจ้งเตือนเมื่อมีการตรวจพบการบุกรุก

- โปรโตคอล ในกฎนี้โปรโตคอลคือไอซีเอ็มพี หมายความว่ากฎนี้ใช้ได้แค่แพ็กเก็ตประเภทไอซีเอ็มพี

- แอดเดรสและพอร์ตต้นทาง ในตัวอย่างนี้ใช้ได้กับทุกไอซีเอ็มพีแพ็กเก็ต ส่วนหมายเลขพอร์ตไม่เกี่ยวกับไอซีเอ็มพีแพ็กเก็ต

- ทิศทาง ใช้สัญลักษณ์ ->

- แอดเดรสและพอร์ตปลายทาง ในตัวอย่างนี้ใช้ได้กับทุกไอซีเอ็มพีแพ็กเก็ต ส่วนออปชัน จะอยู่ในวงเล็บ ( ) ซึ่งจะแสดงให้เห็นว่าข้อความแจ้งเตือนที่จะสร้างเท็กซ์สตริง " Ping with TTL=100 " เมื่อเกิดทีทีแอล (TTL:Time To Live) = 100

เฮดเดอร์ของกฎจะประกอบไปด้วย

1.1) แอคชันของกฎ (Rule Action) ทำแอ็คชันเมื่อกฎเป็นจริง ประกอบด้วย 5 แอ็คชัน

1.1.1) การผ่าน (Pass)

ทำการเพิกเฉยต่อแพ็กเก็ต ใช้ในกรณีที่ต้องการหาช่องโหว่ในระบบ เพราะมันจะ ทำได้รวดเร็วกว่าแอ็คชันอื่น

1..1.2) การบันทึก (Log)

ทำการบันทึกแพ็กเก็ต เช่นข้อความจะถูกบันทึกให้เป็นไฟล์หรือเก็บในฐานข้อมูลก็ได้ สามารถเก็บ รายละเอียด ได้หลายระดับตามแต่ที่กำหนดไว้ในคอมมานด์ไลน์อากิวเมนต์ (Command Line Argument) และไฟล์คอนฟิกเกอร์เรชัน (Configuration File)

1.1.3) การแจ้งเตือน (Alert)

ใช้สำหรับส่งข้อความแจ้งเตือนเมื่อกฎเป็นจริง การส่งส่งได้หลายวิธี เช่น เตือนผ่านทางไฟล์หรือเตือนผ่านทางคอนโซล ข้อแตกต่างระหว่างการบันทึกกับการแจ้งเตือนคือ การแจ้งเตือนจะมีข้อความขึ้นมาและบันทึกแพ็กเก็ต ส่วนการบันทึกจะทำแค่การบันทึกแพ็กเก็ต

1.1.4) พลวัต (Dynamic)

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

1.1.5) ยูสเซอร์ดีไฟน์แอคชัน (User Defined Actions)

สามารถกำหนดแอ็คชันได้เอง เช่น

• ส่งข้อความไปยังซิสล็อก(Syslog) ซิสล็อกคือตัวบันทึกของระบบที่เป็นเดมอนโพรเซส(Daemon Process) และสร้างไฟล์บันทึกที่ / var / log syslog เปรียบได้กับตัวบันทึกเหตุการณ์ของวินโดว์

• ส่งเอสเอ็นเอ็มพีแทร็ป (SNMP Traps) เอสเอ็นเอ็มพีแทร็ปจะถูกส่งไปยังระบบจัดการเครือข่าย

• ทำงานหลายๆแอ็คชัน บน 1 แพ็กเก็ตเช่น ส่งเอสเอ็นเอ็มพีแทร็ปและ ข้อมูลบันทึกการแจ้งเตือน(Log Alert Data) ไว้ที่ซิสล็อก

• ข้อมูลการบันทึก (Log Data) ให้เป็นเอ็กซ์เอ็มแอลไฟล์ (XML Files)

สนอร์ตสามารถบันทึกข้อความเก็บไว้ในฐานข้อมูลของมายเอสคิวแอล(MySQL),โพสต์เกรสเอสคิวแอล(PostgressSQL) , ออราเคิล(Oracle) และ ไมโครซอฟต์เอสคิวแอลเซิร์ฟเวอร์ (Microsoft SQL Server)

แอ็คชันจะถูกกำหนดไว้ในไฟล์คอนฟิกเกอเรชัน (snort.conf) การกำหนดแอ็คชันทำได้ดังนี้

```
rule type action_name
{
action definition
}
```

ตัวอย่างของกฎ

```
rule type smb_db_alert
{
type alert
output alert_smb : workstation.list
output database : log ,mysql,user = rr password=rr \
dbname = snort hsot = localhost
}
```

1.2) โปรโตคอล ส่วนนี้จะแสดงให้เห็นว่ากฎจะจับแพ็กเก็ตบนโปรโตคอลไหน

1.3) แอดเดรส แบ่งเป็น 2 ส่วน คือส่วนแรกเป็นแอดเดรสต้นทาง ส่วนหลังเป็นแอดเดรสปลายทาง จะเป็นไอพีแอดเดรสหรือเน็ตเวิร์กแอดเดรสก็ได้ คำสำคัญ "any " คือ ทุกๆ

แอดเดรส ตัวอย่างเช่น ถ้าต้องการสร้างการแจ้งเตือนสำหรับทุกๆ ทีซีพีแพ็กเก็ตที่มีค่าทีทีแอล (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;)

1.3.1) แอดเดรสที่ถูกยกเว้น (Address Exclusion)

ใช้สำหรับแอดเดรสที่ยกเว้น ตัวอย่างเช่น จับทุกแพ็กเก็ตเว้นแต่ที่มาจาก 192.168.2.0

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

ประโยชน์คือใช้ในการทดสอบแพ็กเก็ตที่ไม่ได้มาจากโฮมเน็ตเวิร์ก (Home Network)

1.3.2) รายชื่อแอดเดรส (Address List)

ใส่แอดเดรสเป็นรายชื่อ ตัวอย่างเช่น โฮมเน็ตเวิร์กประกอบด้วยไอพี 192.168.2.0 , 192.168.8.0 เพิ่มในกฎที่แล้ว ทำได้ดังนี้

alert icmp ![192.168.2.0/24,192.168.8.0/24] any -> any any \ (msg : "Ping with TTl = 100 ";ttl :100;)

: ใช้เมื่อมี !

1.4) หมายเลขพอร์ต (Port Number) ใช้ในการจับแพ็กเก็ต ว่ามาจากพอร์ตไหน และจะไปยังพอร์ตหรือช่วงของพอร์ตอะไร เช่นใช้พอร์ตต้นทาง หมายเลข 23 ไว้ดักแพ็กเก็ตที่มาจากเทลเน็ตเซิร์ฟเวอร์

2) ส่วนออปชันของกฎ (Rules Option)

ทางเลือก (Option) ของกฎ จะอยู่ต่อจากเฮดเดอร์ของกฎและอยู่ในวงเล็บ ซึ่งจะมีทางเลือกเดียวหรือหลายทางเลือกก็ได้ โดยแต่ละทางเลือกจะถูกคั่นด้วยเซมิโคลอน (;) ถ้าคุณเลือกใช้หลายทางเลือกมันจะเหมือนกับการแอนด์ (And) ทางลอจิก การกระทำของเฮดเดอร์ของกฎจะเกิด ขึ้นเมื่อทางเลือกทุกๆทางเป็นจริงขึ้นมา ทุกๆทางเลือกจะถูกประกาศโดยใช้คำสำคัญหรือบางตัวก็บรรจุอากิวเมนต์(Argument) อยู่ข้างใน โดยทั่วไปแล้ว ทางเลือกนั้นอาจจะมี 2 ส่วนคือ ส่วนคำสำคัญ (Keyword) และส่วนอากิวเมนต์ (Argument) อากิวเมนต์จะถูกแยกจากคำสำคัญโดยโคลอน (:) เช่นmsg : " Detected confidential " ;

2.1) คำสำคัญ "ack"

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

2.2) คำสำคัญ "classtype"

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

2.3) คำสำคัญ "content"

คุณลักษณะสำคัญของสนอร์ตข้อหนึ่งคือสามารถค้นหาข้อมูลในแพ็กเก็ตได้ โดยข้อมูลนั้นอาจอยู่ในรูปของ แอสกี (ASCII) หรือตัวอักษร เช่นไวรัส, ผู้บุกรุก ซึ่งได้มีคำสำคัญ "content" และสัญลักษณ์ ในการค้นหาพวกมันในแพ็กเก็ตอยู่แล้ว

2.4) คำสำคัญ "offset"

ใช้ร่วมกับคำสำคัญ "content" สามารถที่จะเริ่มค้นหาได้ตั้งแต่ส่วนที่เป็นข้อมูลจุดเริ่มต้นของแพ็กเก็ตจะมีการใช้ตัวเลข และอาก๊วเมนต์ในคำสำคัญนี้

2.5) คำสำคัญ "depth"

ใช้ร่วมกับคำสำคัญ "content" เช่นเดียวกัน เพื่อที่จะใช้กำหนดขอบเขตมากสุดของการจับคู่ สามารถที่จะระบุทิศทางจากจุดเริ่มของแพ็กเก็ตได้ ข้อมูลหลังจากทิศทางที่ระบุไว้จะไม่ถูกตรวจหาข้อมูลในแพ็กเก็ตถ้าใช้ทิศทาง 2 ทางร่วมกันก็จะสามารถบอกถึงขนาดความยาวของแพ็กเก็ตได้

2.6) คำสำคัญ "content-list"

ใช้กับชื่อของไฟล์ซึ่งเหมือนเป็นอาก๊วเมนต์ของคำสำคัญแบบนี้ เท็กซ์ไฟล์ตัวนี้จะบรรจุตัวอักษรซึ่งจะใช้ในการค้นหาภายในแพ็กเก็ต แต่ละอาก๊วเมนต์จะแบ่งเป็นบรรทัด

2.7) คำสำคัญ "dsize"

จะถูกใช้เพื่อหาความยาวของส่วนที่เป็นข้อมูลในแพ็กเก็ต การโจมตีหลายๆครั้งจะใช้วิธีบัฟเฟอร์โอเวอร์โฟลว์ที่จะทำการส่งแพ็กเก็ตข้อมูลจำนวนมากเข้ามา ด้วยการใช้คำสำคัญนี้ทำให้เราสามารถระบุได้ว่าแพ็กเก็ตที่เข้ามามีขนาดเท่าไหร่และตรงกับการโจมตีแบบใดหรือเปล่า

2.8) คำสำคัญ "flags"

ใช้ตรวจสอบว่าแฟลกใดในทีซีพีเฮดเดอร์ที่ส่งมาถูกเซตไว้บ้าง แต่ละแฟลกสามารถใช้เป็นอาก๊วเมนต์สำหรับ กฎของสนอร์ตได้

2.9) คำสำคัญ "flagbits"

ในทีซีพีเฮดเดอร์นั้นบรรจุด้วย 3 แฟลกบิต ซึ่งใช้สำหรับการแบ่งส่วนของข้อมูล และการทำรีแอสเซมบลี (Reassembly) บางบิตจะถูกใช้โดย แฮคเกอร์เพื่อที่จะบุกรุกและ ใช้ตรวจสอบข้อมูลบนเครือข่ายนั้น

2.10) คำสำคัญ "icmp_id"

ใช้ตรวจจับเลขไอดีเจาะจงที่ใช้ร่วมกับไอซีเอ็มพีแพ็กเก็ต จะมีประโยชน์มากเมื่อค้นหาว่าแพ็กเก็ตใดถูกตอบกลับในการร้องแบบเจาะจง

2.11) คำสำคัญ "icmp_seq"

ใช้ในการช่วยหาจำนวนตัวเลขที่เจาะจง เพื่อที่จะเอาไปเทียบกับกฎว่าตรงกับกฎไหนหรือไม่

2.12) คำสำคัญ "itype"

ไอซีเอ็มพีเฮดเดอร์เข้ามาหลังจากไอพีเฮดเดอร์ และบรรจุด้วยแถวของ ชนิดของไอซีเอ็มพีนั้นๆ ซึ่งคำสำคัญ นี้จะใช้ตรวจจับการบุกรุกซึ่งใช้ แถวของชนิด ในไอซีเอ็มพีเฮดเดอร์

2.13) คำสำคัญ "icode"

ไอซีเอ็มพีเฮดเดอร์เข้ามาหลังจากไอพีเฮดเดอร์และบรรจุด้วยแถวของรหัส(Code) ซึ่งคำสำคัญตัวนี้จะใช้ตรวจสอบตัวรหัสที่บรรจุในไอซีเอ็มพีเฮดเดอร์

2.14) คำสำคัญ "id"

ใช้สำหรับจับคู่ส่วนของไอดีของไอพีแพ็กเก็ตเฮดเดอร์เพื่อตรวจจับการบุกรุกที่ทำการคงค่า(Fix) เลขไอดีเอาไว้ ถ้าไอดีเป็น 0 ก็จะหมายถึงตำแหน่งสุดท้ายของไอพีแกเก็ต

2.15) คำสำคัญ "ipopts"

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

2.16) คำสำคัญ "ip_proto"

ใช้ปลั๊กอินของไอพีโปรโต เพื่อที่จำคำนวณเลขโปรโตคอลที่อยู่ในไอพีเฮดเดอร์ คำสำคัญนี้ต้องใช้เลขโปรโตคอลเป็นอากิวเมนต์หรือสามารถใช้ชื่อของโปรโตคอลก็ได้

2.17) คำสำคัญ "logto"

ใช้เพื่อเก็บแพ็กเก็ตไปยังไฟล์พิเศษ อย่าลืมระบุพาธ(Path) ของไฟล์ที่จะเก็บด้วย

2.18) คำสำคัญ "msg"

ใช้เพื่อใส่ตัวอักษรหรือข้อความลงไปในล็อก(Logs) และการแจ้งเตือน โดยใส่ข้อความลงไปใน " "

2.19) คำสำคัญ "nocase"

ใช้ร่วมกับคำสำคัญ "content" โดยตัวมันเองไม่มีอากิวเมนต์ จุดประสงค์หลักคือต้องการค้นหาข้อมูลที่อยู่ในรูปแบบอินเซนซิทีฟ(Insensitive) ในแพ็กเก็ต

2.20) คำสำคัญ "priority"

ใช้ใส่ค่าความสำคัญลงไปในกฎ โดยที่ค่าความสำคัญเท่ากับ 1 จะเป็นค่าที่ทำให้กฎนั้นมีความสำคัญสูงสุด ใช้ในการแยกการแจ้งเตือนแบบต่างๆออกจากกัน

2.21) คำสำคัญ "react"

ใช้ในการทำลายเซสชัน (Session) ปิดกั้นบางเว็บไซต์ หรือบางเซอร์วิส (Service) โดยจะใส่ไว้ส่วนท้ายสุดของกฎ

2.22) คำสำคัญ "reference"

ใช้ในการเพิ่มคำอธิบายลงไปในกฎ

2.23) คำสำคัญ "resp"

เป็นคำสำคัญที่สำคัญมาก มันสามารถจัดการกับกิจกรรมที่มาจากแฮคเกอร์โดยส่งแพ็กเก็ต ไปยังโฮสต์ที่ส่งแพ็กเก็ตมาตรงกับกฎที่กำหนดไว้

2.24) คำสำคัญ "rev"

ใช้เพื่อที่จะแสดงจำนวนรีวิชัน (revision) สำหรับกฎ ถ้าคุณปรับปรุงกฎ คุณสามารถใช้คำสำคัญนี้ในการแสดงจำนวนกฎที่เปลี่ยนไปได้

2.25) คำสำคัญ "rpc"

ใช้ตรวจจับการร้องขออาร์พีซี (RPC) โดยใช้ 3 อากิวเมนต์คือเลขของแอพพลิเคชัน เลขโปรซีเยอร์ และเลข เวอร์ชัน

2.26) คำสำคัญ "sameip"

ใช้เพื่อเช็คว่าไอพีของจุดกำเนิดกับไอพีปลายทางตรงกันหรือเปล่า เนื่องจากมีการทำการปลอมแปลงไอพีได้

2.27) คำสำคัญ "seq"

ใช้ทดสอบลำดับตัวเลขของทีซีพีแพ็กเก็ต ซึ่งอากิวเมนต์ของตัวมันคือ เลขลำดับ

2.28) คำสำคัญ "flow"

ใช้เพื่อคำนวณทิศทางของแพ็กเก็ต

2.29) คำสำคัญ "session"

ใช้ในการแสดงแพ็กเก็ตทั้งหมดที่มาจากทีซีพีเซสชันหรือเฉพาะแค่บางตัวก็ได้ ถ้าใช้ "all" จะเป็นการแสดงทั้งหมด

2.30) คำสำคัญ "sid"

ใช้ในการใส่ไอดีของสนอร์ตลงไปยังกฎ เพื่อที่จะให้ส่วนแสดงผลหรือส่วนเก็บข้อมูลเรียกใช้กฎได้ถูก

2.31) คำสำคัญ "tag

เป็นคำสำคัญที่สำคัญอีกตัวหนึ่ง ซึ่งจะใช้เป็นเก็บข้อมูลที่เข้าหรือออกจากโฮสต์ของผู้บุกรุก เมื่อกฎได้ถูกดำเนินการ เพื่อใช้ในการวิเคราะห์พฤติกรรมของผู้บุกรุกในภายหลัง

2.32) คำสำคัญ "tos"

ใช้ตรวจสอบค่าที่เจาะจงของรูปแบบการให้บริการ ในไอพีเฮดเดอร์

2.33) คำสำคัญ "ttl"

ใช้ตรวจสอบค่าเวลาที่ยังอยู่ (Time To Live) ในไอพีเฮดเดอร์ของแพ็กเก็ต ตัวคำสำคัญควรจะต้องมีค่าเท่ากับค่าที่ประมาณไว้พอดี สามารถใช้ได้กับทุกๆโปรโตคอลใช้ในการตรวจสอบว่ามีคนพยายามจะทำการ ติดตามเส้นทาง (Traceroute) มายังเครือข่ายของเราหรือไม่

2.34) คำสำคัญ "uricontent"

เหมือนกับคำสำคัญ "content" จะแตกต่างตรงที่มันจะคอยตรวจสอบความผิดปรกติที่ส่วนยูอาร์ไอ (URI) ในแพ็กเก็ตเท่านั้น

3.3.3 วิธีปรับแต่งโปรแกรมสนอร์ต

- แก้ไฟล์ snort.conf ที่ตัวแปร RULE_PATH ให้ตรงกับ path ที่เก็บกฎ(rules)ไว้ดังนี้

  var RULE_PATH /rules

  หรือถ้าใช้กับระบบปฏิบัติการวินโดว์ให้เซตเป็นลักษณะดังนี้

  var RULE_PATH c:\snort\rules

- เมื่อกำหนดพาธของกฎเรียบร้อยแล้วให้ทำการเรียกใช้กฎนั้นโดยพิมพ์คำสั่ง

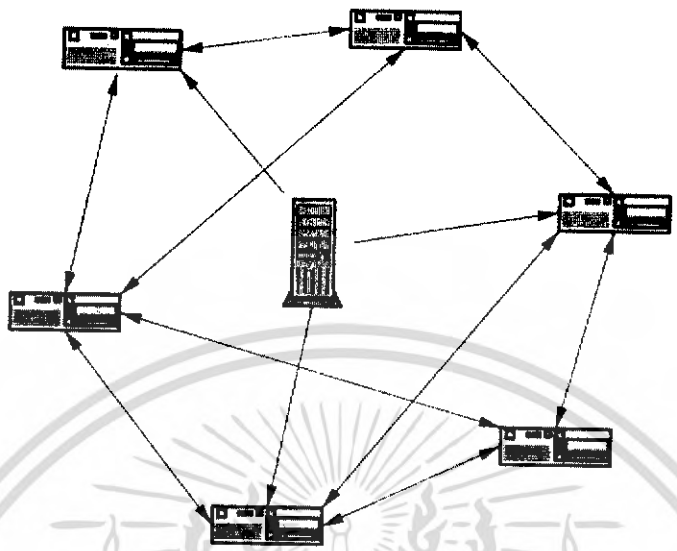
  Include something.rules เช่น include $RULE_PATH/p2p.rules

  หรือถ้าในไฟล์ snort.conf มีคำสั่งอยู่แล้ว ให้เอาเครื่องหมาย "#" ออก

3) แทร็กเกอร์ เซิร์ฟเวอร์จะตรวจสอบว่าไฟล์ที่ขอมามีการลงทะเบียนไว้ในระบบหรือไม่

3.4 การทำงานและพฤติกรรมของการสื่อสารแบบเพียร์ทูเพียร์

3.4.1 องค์ประกอบของเพียร์ทูเพียร์



รูปที่ 3.6 ภาพแสดงองค์ประกอบของเพียร์ทูเพียร์

ไคลเอนต์ คือโปรแกรมที่จะนำมาใช้ในการใช้งานการสื่อสารแบบเพียร์ทูเพียร์ ซึ่งมีอยู่หลายตัว เช่น Bitcomet , BitTornado , Azureus , TorrentStorm โปรแกรมเหล่านี้จะใช้ในการดาวน์โหลดไฟล์และอัพโหลดไฟล์

แทร็กเกอร์ เซิร์ฟเวอร์**(Tracker server)** คือเซิร์ฟเวอร์ที่ทำหน้าที่เป็นแม่ข่ายกลางระหว่างไคลเอนต์

ทอร์เรนต์ ไฟล์**(Torrent file)** เป็นไฟล์ที่เก็บข้อมูลเพื่อใช้ในการดาวน์โหลด ซึ่งประกอบไปด้วย แอดเดรสของแทร็กเกอร์ , Check sum เพื่อเอาไว้ใช้ในการตรวจสอบความถูกต้องของไฟล์

3.4.2 การทำงานของเพียร์ทูเพียร์

มีกระบวนการทำงานทั้งหมด ดังต่อไปนี้

1) ค้นหาไฟล์ .torrent มาจากเว็บไซต์ แล้วนำมาเปิดด้วยโปรแกรมที่ใช้ในการสื่อสารแบบเพียร์ทูเพียร์(บิตโคเม็ต , อีมุลล์ และแซคพีทูพี เป็นต้น)

2) โปรแกรมจะอ่านค่าแอดเดรสของแทร็กเกอร์ เซิร์ฟเวอร์ในไฟล์ที่เปิด แล้วทำการติดต่อไปหาแทร็กเกอร์เพื่อทำการส่งข้อมูลเกี่ยวกับไฟล์ .torrent

3) แทร็กเกอร์ เซิร์ฟเวอร์จะตรวจสอบว่าไฟล์ที่ขอมามีการลงทะเบียนไว้ในระบบหรือไม่ ถ้ามีจะตรวจดูว่ามีไอพีแอดเดรสอะไรบ้างที่มีการอัพโหลดและดาวน์โหลดไฟล์นี้อยู่

4) ส่งรายการไอพีแอดเดรสของคนที่มีการติดต่ออยู่กลับไป และทำการเก็บไอพีแอดเดรสของเราไว้ด้วย (เก็บไว้ส่งให้คนอื่น)

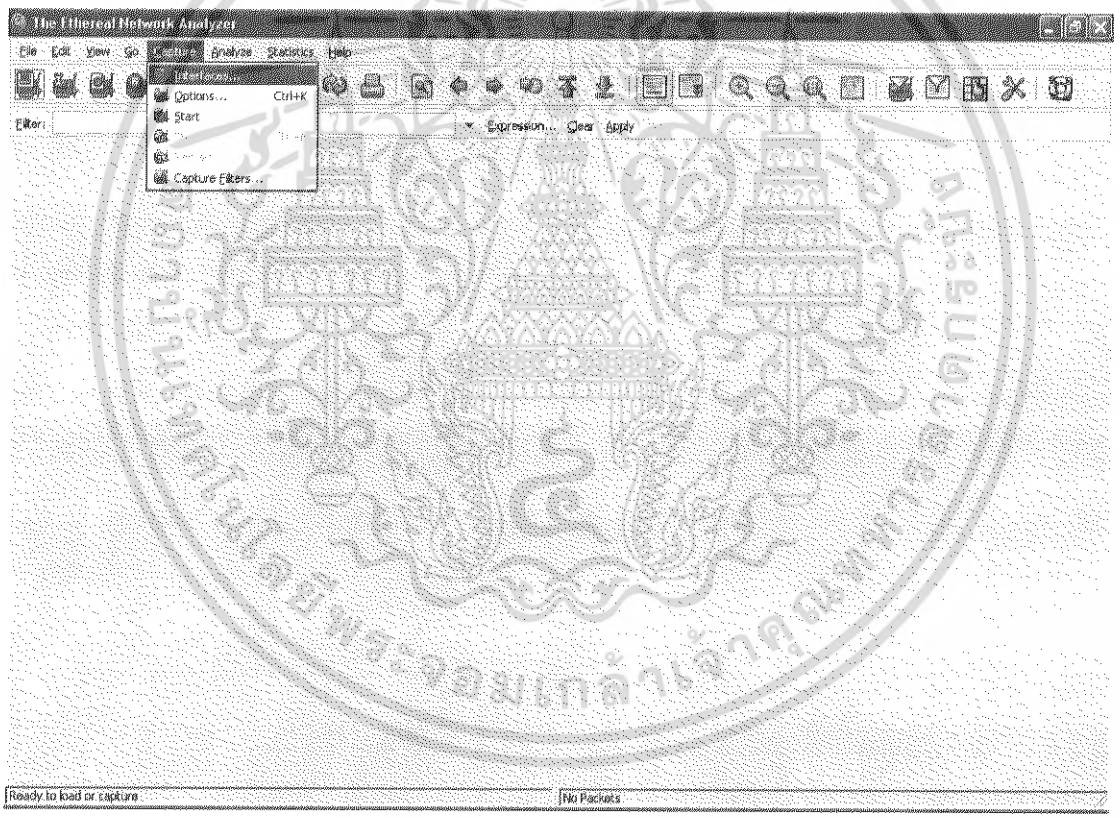
5) เมื่อทอร์เรนต์ไคลเอนต์(Torrent client) ได้ไอพีแอดเดรสแล้ว มันก็จะทำการติดต่อไปยังไอพีแอดเดรสที่ได้มาโดยจะส่งคำร้องขอไปสอบถามว่าแต่ละไอพีแอดเดรสนั้นมีส่วนไหนของไฟล์ที่ต้องการนี้อยู่บ้าง

6) แล้วปลายทางก็จะบอกว่าส่วนไหนของไฟล์ที่ปลายทางมีอยู่ แล้วส่งไปบอกต้นทาง

7) ต้นทางก็จะทำการตรวจว่ามีส่วนไหนของไฟล์ที่ยังขาดอยู่บ้าง แล้วส่งคำขอเฉพาะส่วนที่ต้องการ แล้วปลายทางก็จะส่งส่วนของไฟล์ที่ต้องการนั้นมาให้

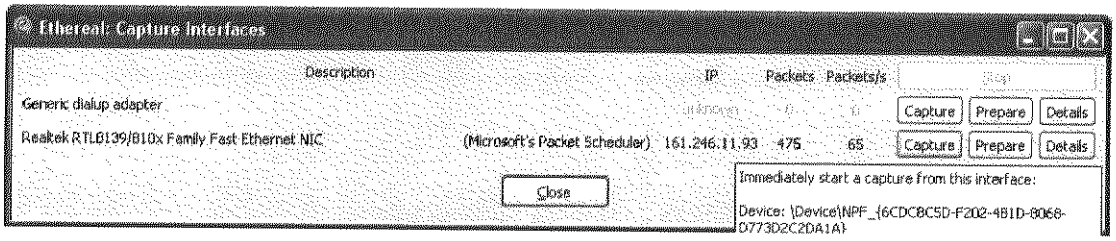
3.5 การทำงานของโปรแกรมอีเทอร์เรียล

มีการทำงานดังนี้



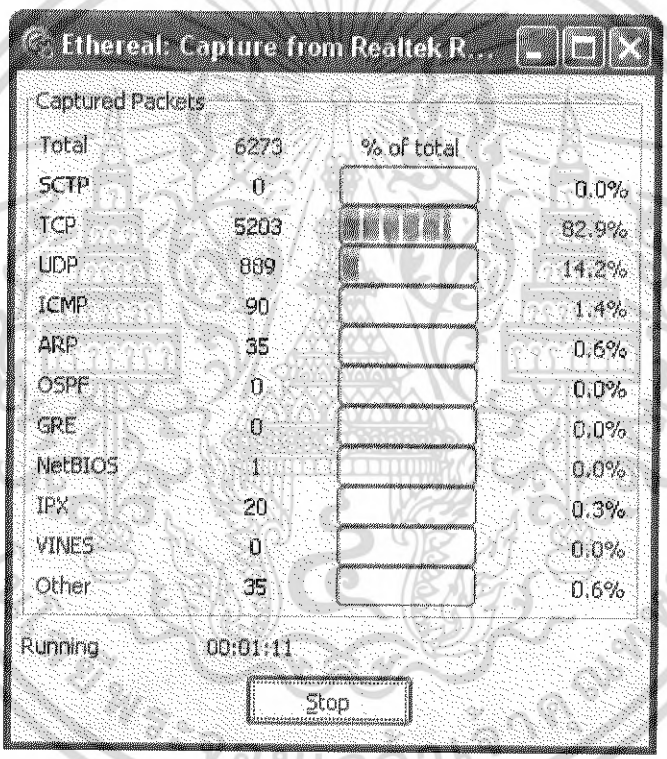
รูปที่ 3.7 แสดงหน้าจอหลักของโปรแกรมอีเทอร์เรียล

ให้เปิดไปที่เมนู Capture -> Interface จากนั้นจะปรากฏหน้าจอดังนี้



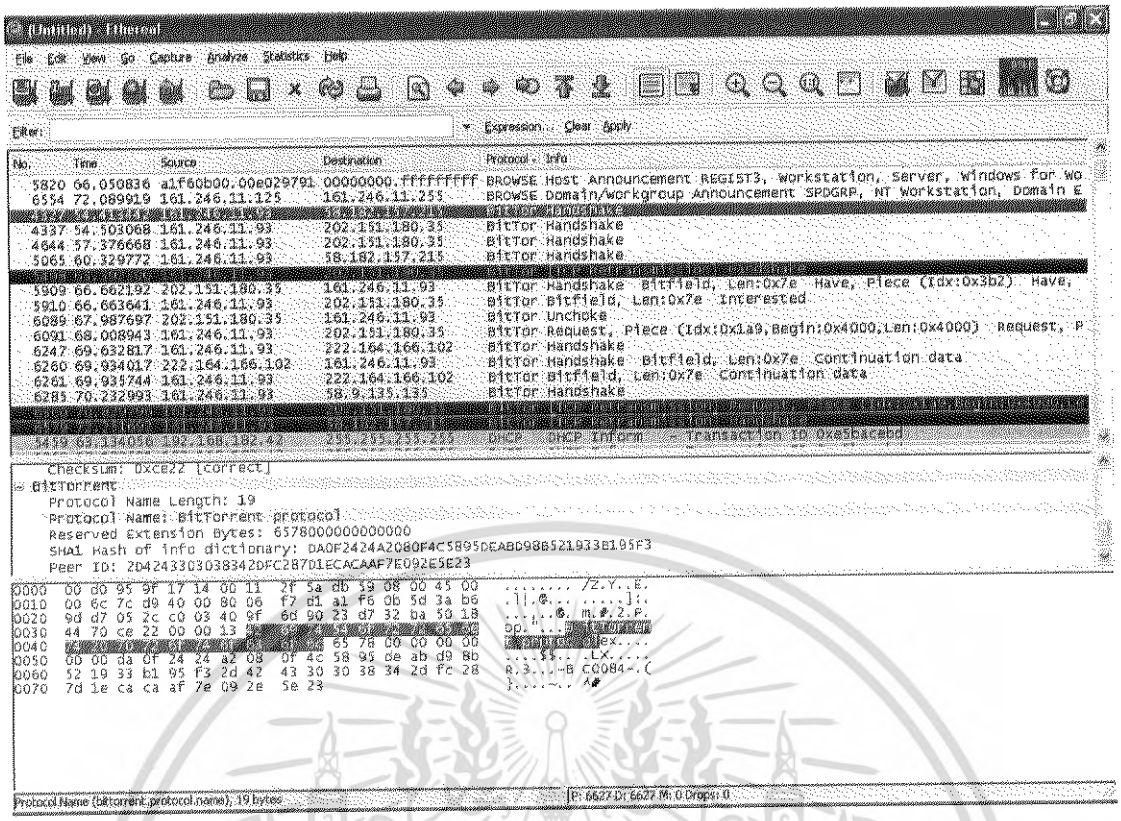
รูปที่ 3.8 แสดงหน้าจอตอนเลือก Network Card ว่าจะจับแพ็กเก็ตที่ Network Card ใด

ซึ่งจะแสดงถึง Network Card ที่มีอยู่ แล้วให้คลิกไปที่ปุ่ม Capture ของ Network Card ที่เราต้องการใช้ในการตรวจจับ Packet แล้วจะมีหน้าจอแสดงถึง Packet ต่าง ๆ ที่ผ่านเข้าออกผ่าน Network Card นี้



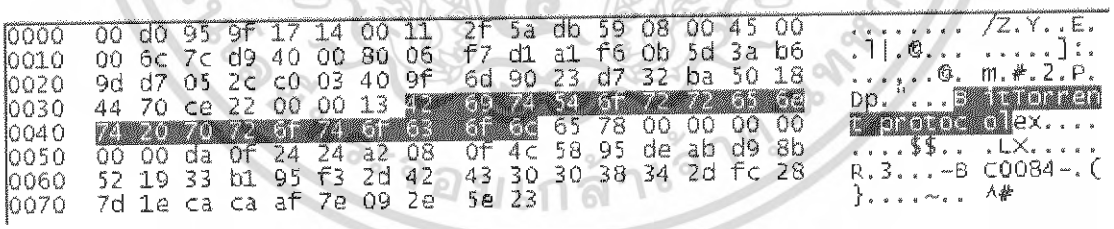
รูปที่ 3.9 หน้าจอแสดงผลการเปรียบเทียบจำนวนแพ็กเก็ตที่จับได้ในแต่ละโปรโตคอล

Packet ต่าง ๆ ที่ผ่านเข้าออกจะแสดงออกมาเป็นประเภท ๆ จากนั้นให้คลิก Stop แล้วจะมีหน้าจอแสดงรายละเอียดต่าง ๆ ของแต่ละ Packet ที่ผ่าน



รูปที่ 3.10 หน้าจอแสดงรายละเอียดของแพ็กเก็ตต่างๆที่จับได้

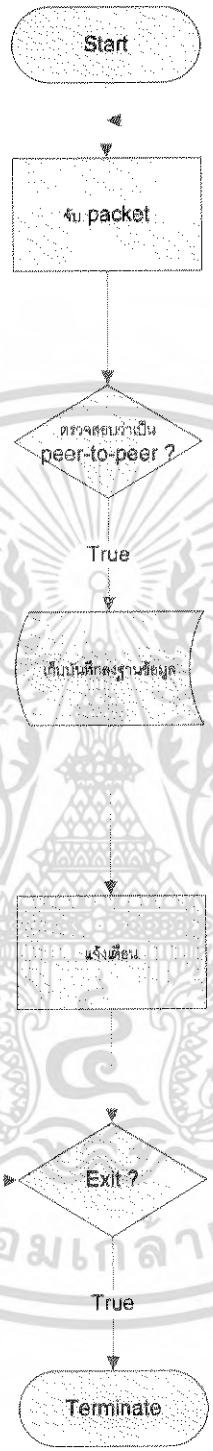
ซึ่งด้านบนจะแสดงถึง Packet ต่างๆที่โปรแกรมจับได้ พร้อมทั้งแสดงไอพีแอดเดรสของผู้ส่งและผู้รับ และแสดง Protocol ที่ Packet นั้นใช้ เมื่อคลิกไปที่ Packet จะมีข้อมูลด้านล่างซึ่งก็คือรายละเอียดของ Packet



รูปที่ 3.11 แสดงรายละเอียดของแพ็กเก็ต

ใน Packet ที่มีการใช้งาน Bittorrent Protocol จะมี Header ใน Packet ต่าง ๆ ที่มีลักษณะเหมือนกันคือ จะมีคำว่า “Bittorrent protocol” เป็นต้น

3.6 การทำงานของระบบ



รูปที่ 3.12 Flow chart การทำงานของระบบ

เริ่มแรกจะทำการตรวจจับและตรวจสอบแพ็กเกตที่วิ่งเข้ามาในระบบว่าตรงกับกฎที่กำหนดไว้หรือไม่ซึ่งก็คือเป็นแพ็กเก็ตข้อมูลที่เป็นการสื่อสารแบบเพียร์ทูเพียร์หรือไม่ ถ้าใช่ก็จะทำการบันทึกข้อมูลของแพ็กเก็ตนั้นลงในฐานข้อมูลและแจ้งเตือนบนหน้าจอ ถ้าไม่ใช่ก็จะตรวจจับและตรวจสอบแพ็กเก็ตข้อมูลต่อไป

บทที่ 4
ผลการดำเนินงานวิจัยวิจัย

4.1 เครื่องมือที่ใช้ในการทดสอบโปรแกรม

4.1.1 ความต้องการทางด้านฮาร์ดแวร์

1) เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นระบบตรวจจับผู้บุกรุก

- คอมพิวเตอร์ที่มีหน่วยประมวลผลกลางรุ่นเพนเทียมทรี ขึ้นไป
- สามารถเชื่อมต่อกับเครือข่ายทีซีพี / ไอพีได้
- หน่วยความจำขนาด 128 เมกะไบต์ ขึ้นไป
- พื้นที่ว่างในฮาร์ดดิสก์ 10 กิกะไบต์ ขึ้นไป

2) อุปกรณ์ทางด้านเครือข่าย

- ฮับ (Hub)
- สายยูทีพี (UTP) 10/100 พร้อมหัวอาร์เจ-45 (RJ)-45

4.1.2 ความต้องการทางด้านซอฟต์แวร์

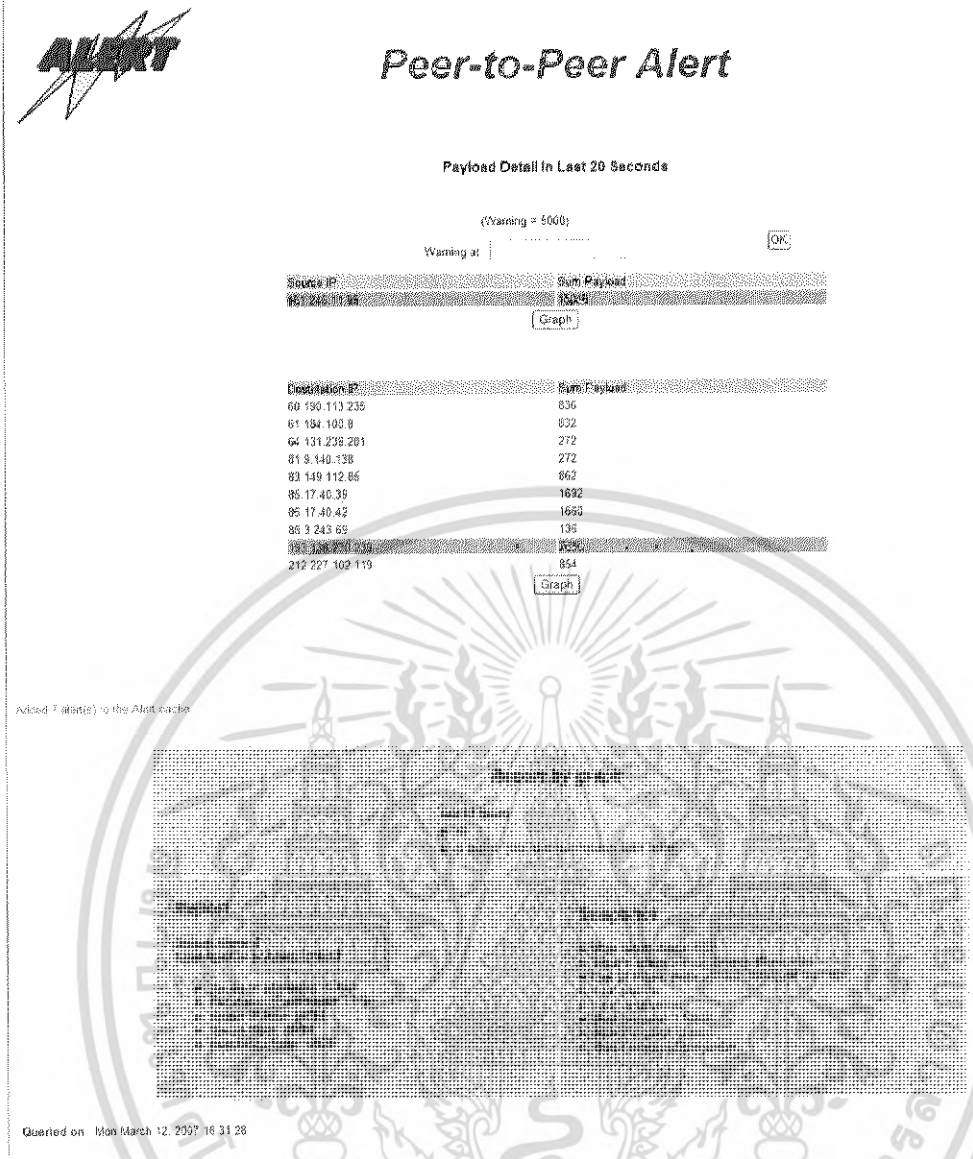
- ระบบปฏิบัติการฟีโดราคอร์ 4 (Fedora Core 4)
- โปรแกรมอาปาเช่ เวอร์ชั่น2.2.4 (Apache v.2.2.4) , มายเอสคิวแอล เวอร์ชั่น 4.1.20 (MySQL v.4.1.20) และพีเอชพีเวอร์ชั่น 5.2.0 (PHP v.5.2.0)

4.2 ขั้นตอนการติดตั้งโปรแกรม

ทำการติดตั้งระบบปฏิบัติการ และซอฟต์แวร์ ทั้งหมดที่จำเป็นต้องใช้ลงในเครื่องที่กำหนดให้เป็นเครื่องตรวจจับ

4.3 การทำงานของโปรแกรม

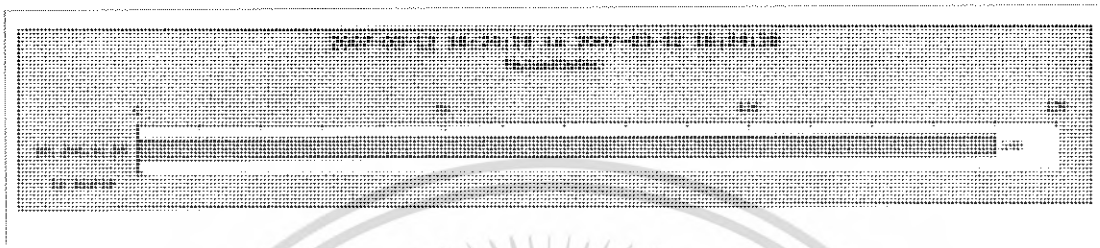
ในการพัฒนาระบบจะทำการสร้างส่วนติดต่อกับผู้ใช้งาน (User Interface) โดยผ่านทางเวบเพจ (Web Page) สามารถเรียกใช้งานได้โดยเปิดบราวเซอร์ (Browser) และพิมพ์ในช่องแอดเดรสว่า http://localhost/acid/acid_main.php



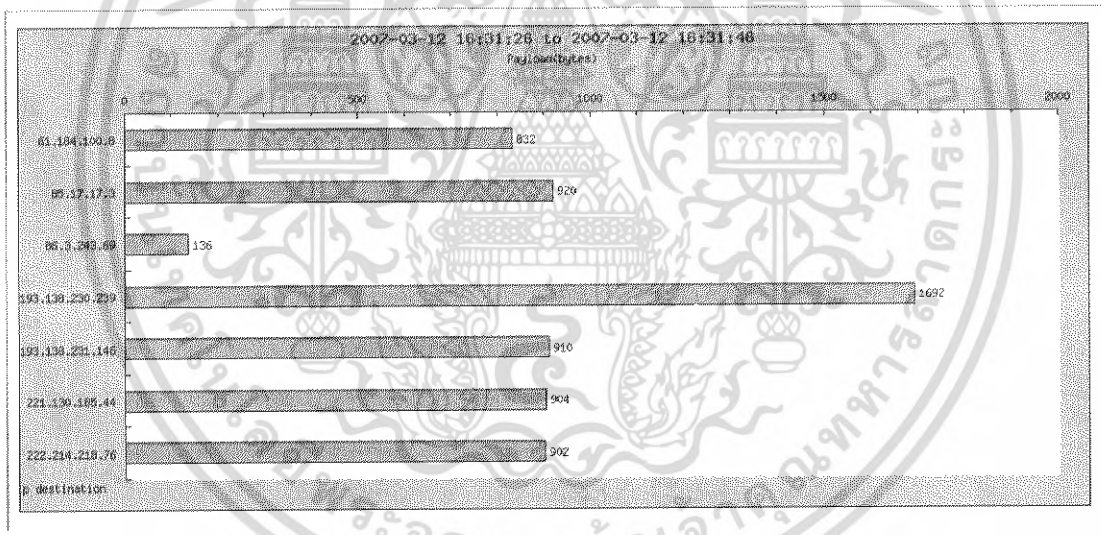
รูปที่ 4.1 หน้าจอหลักของระบบ

4.3.1 เพย์โหลด ดีเทล(payload detail)

โดยจะพิจารณาจากต้นทางและปลายทางที่ตรวจจับได้ว่าแต่ละแอดเดรสใช้เพย์โหลดไปเท่าไร ถ้าเกิน 2,000 ไบต์ จะแสดงเป็นแถบสีแดง(ปกติเป็นแถบสีเหลือง) โดยจะทำการรีเฟรช(Refresh)หน้าจอใหม่ทุกๆ 20 วินาที เมื่อต้องการดูกราฟให้คลิกที่ปุ่ม “Graph” จะได้ผลดังรูปที่ 4.2 และ 4.3



รูปที่ 4.2 หน้าจอแสดงกราฟแท่งแสดงความสัมพันธ์ระหว่างไอพีแอดเดรสต้นทางกับเพย์โหลดที่ใช้

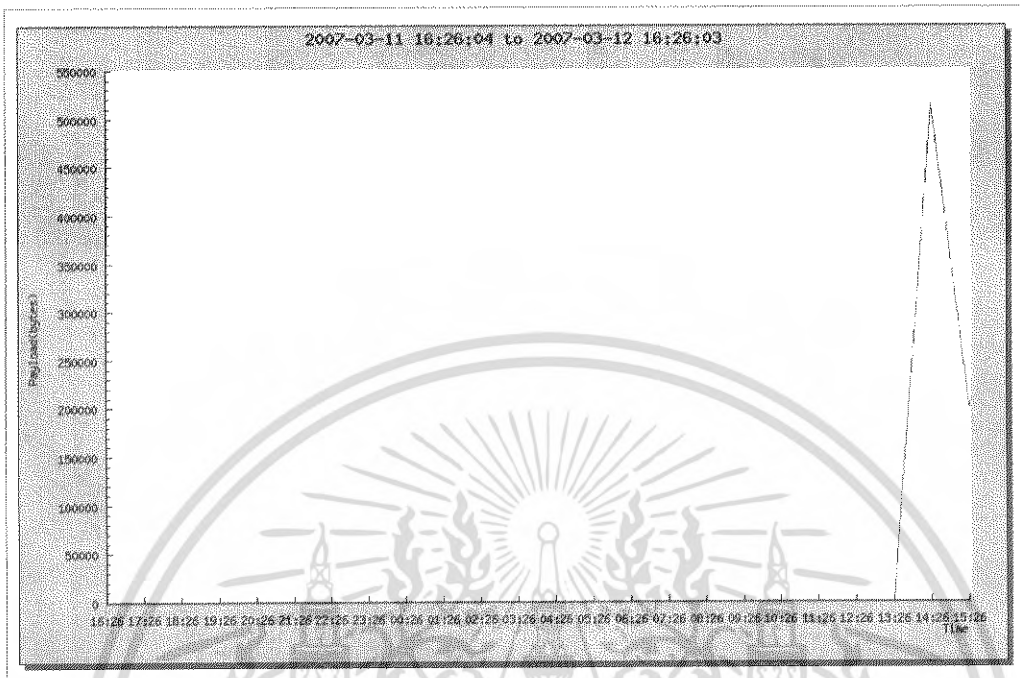


รูปที่ 4.3 หน้าจอแสดงกราฟแท่งแสดงความสัมพันธ์ระหว่างไอพีแอดเดรสปลายทางกับเพย์โหลดที่ใช้

4.3.2 รายงานแสดงผลด้วยกราฟในเวลา 24 ชั่วโมงล่าสุด

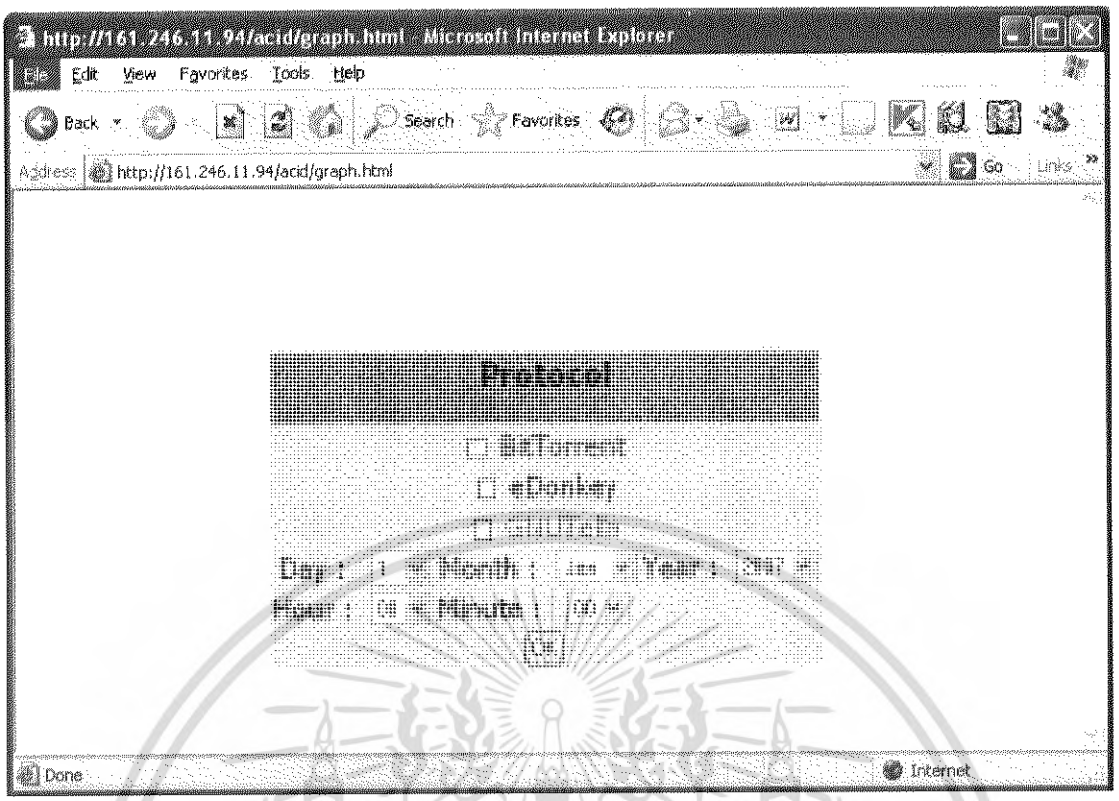
มีให้เลือก 2 แบบคือ

1) กราฟเส้นแสดงเพย์โหลดของโปรโตคอลเพียร์ทูเพียร์ทั้งหมดรวมกัน ใน 24 ชั่วโมงล่าสุด ดังรูปที่ 4.4

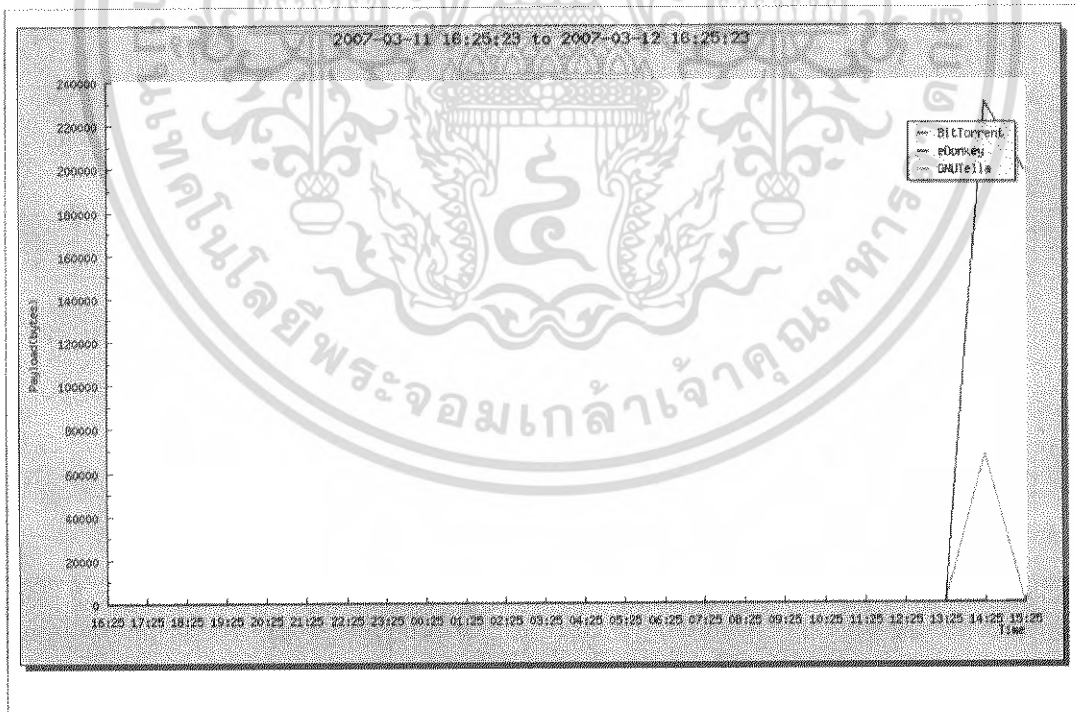


รูปที่ 4.4 หน้าจอแสดงกราฟเส้นแสดงความสัมพันธ์ระหว่างเพย์โหลดของโปรโตคอลเพียร์ทูเพียร์ทั้งหมดรวมกันกับเวลา 24 ชั่วโมงล่าสุด

2) กราฟเส้นแสดงเพย์โหลดของโปรโตคอลเพียร์ทูเพียร์ตามแต่ผู้ใช้จะเลือก ใน 24 ชั่วโมงล่าสุด ดังรูปที่ 4.5 และ 4.6



รูปที่ 4.5 หน้าจอแสดงการเลือกว่าจะแสดงรายงานด้วยกราฟของโปรโตคอลตัวใดและวันเวลาใด



รูปที่ 4.6 หน้าจอแสดงกราฟเส้นแสดงความสัมพันธ์ระหว่างเพย์โหลดของแต่ละโปรโตคอลใน 24 ชั่วโมงล่าสุด

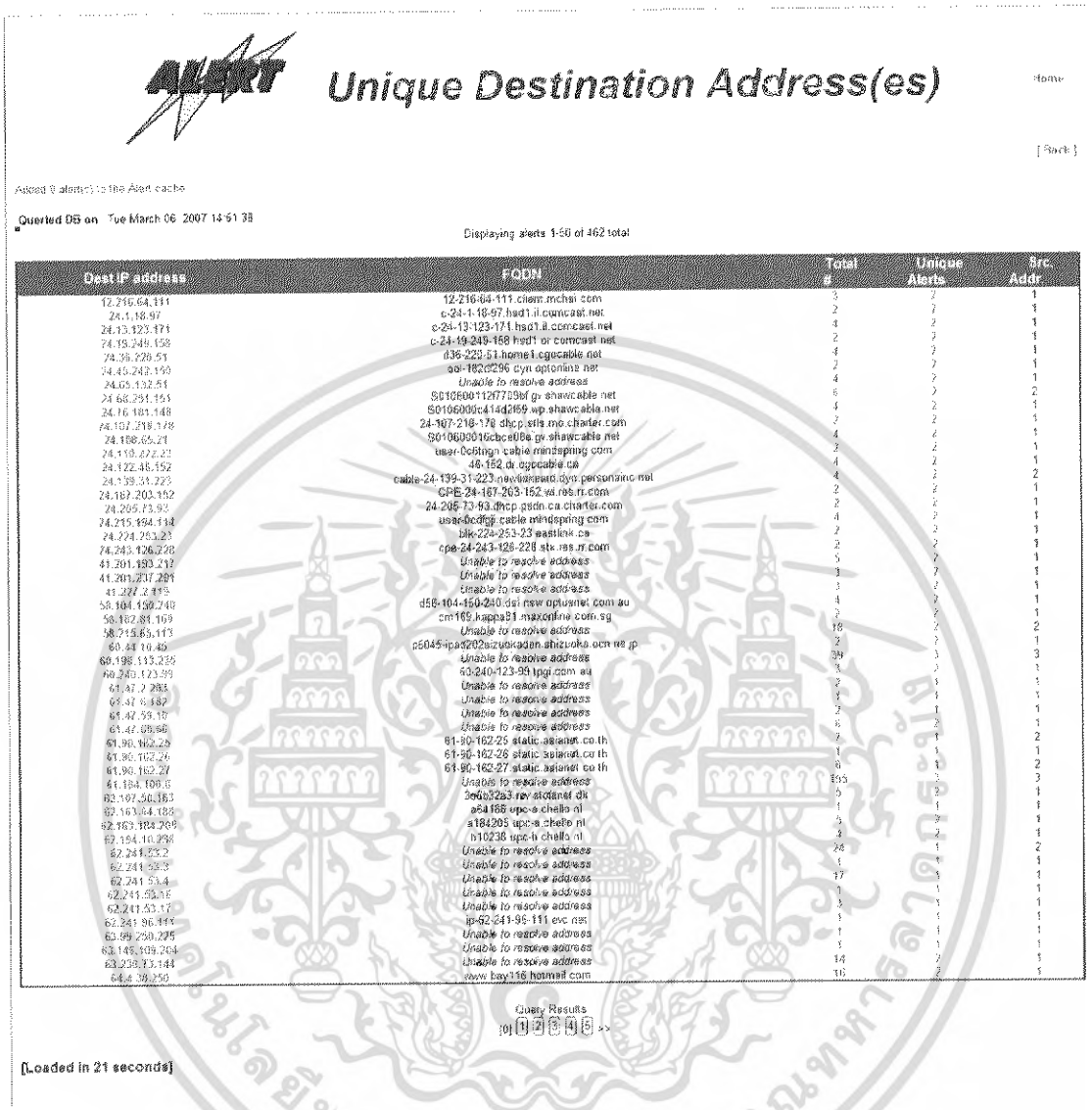
4.3.3 การแสดงรายงานต่างๆ

มีรายละเอียดดังนี้

สามารถแสดงผลตามไอพีแอดเดรสต้นทางได้โดยระบบจะแสดงว่าไอพีแอดเดรสนั้นๆใช้งานทั้งหมดกี่แพ็กเก็ต ใช้งานกี่รูปแบบ และไอพีแอดเดรสปลายทางมีจำนวนเท่าไร ดังรูป

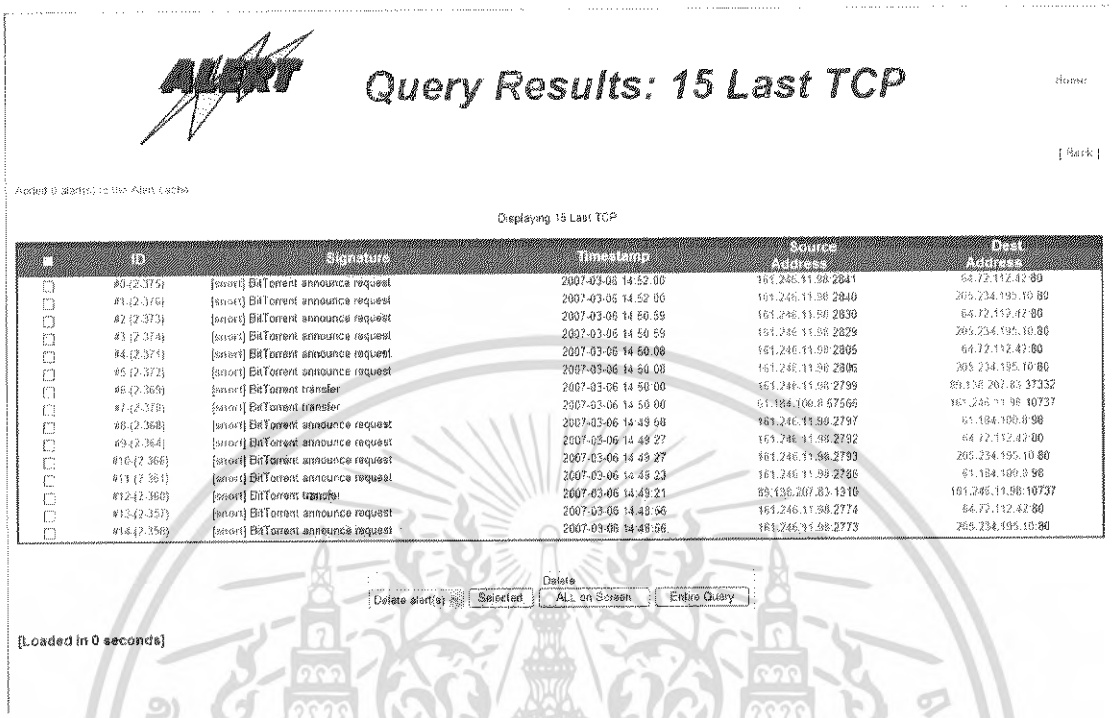
รูปที่ 4.7 หน้าจอแสดงผลตามไอพีแอดเดรสต้นทาง

สามารถแสดงผลตามไอพีแอดเดรสปลายทางได้ โดยระบบจะแสดงว่าไอพีแอดเดรสนั้นๆใช้งานทั้งหมดกี่แพ็กเก็ต ใช้งานกี่รูปแบบ และไอพีแอดเดรสต้นทางมีจำนวนเท่าไร ดังรูป



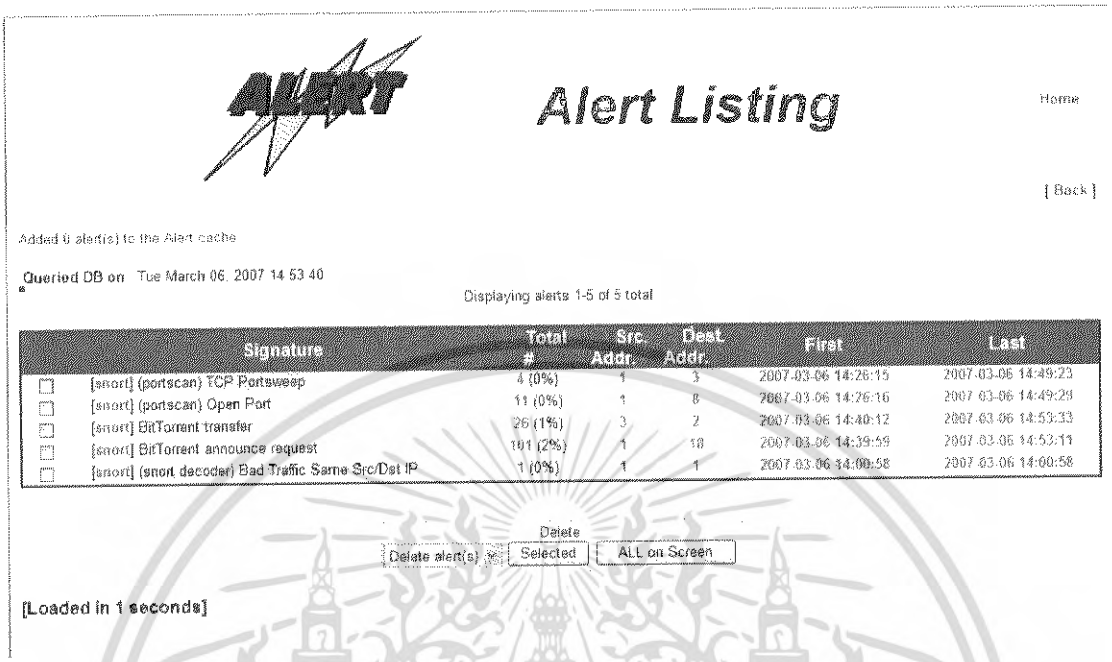
รูปที่ 4.8 หน้าจอแสดงผลตามไอพีแอดเดรสปลายทาง

ระบบยังสามารถแสดงผลการตรวจจับโดยเรียงแพ็กเก็ต 15 แพ็กเก็ตล่าสุดที่บันทึกไว้ดังรูป



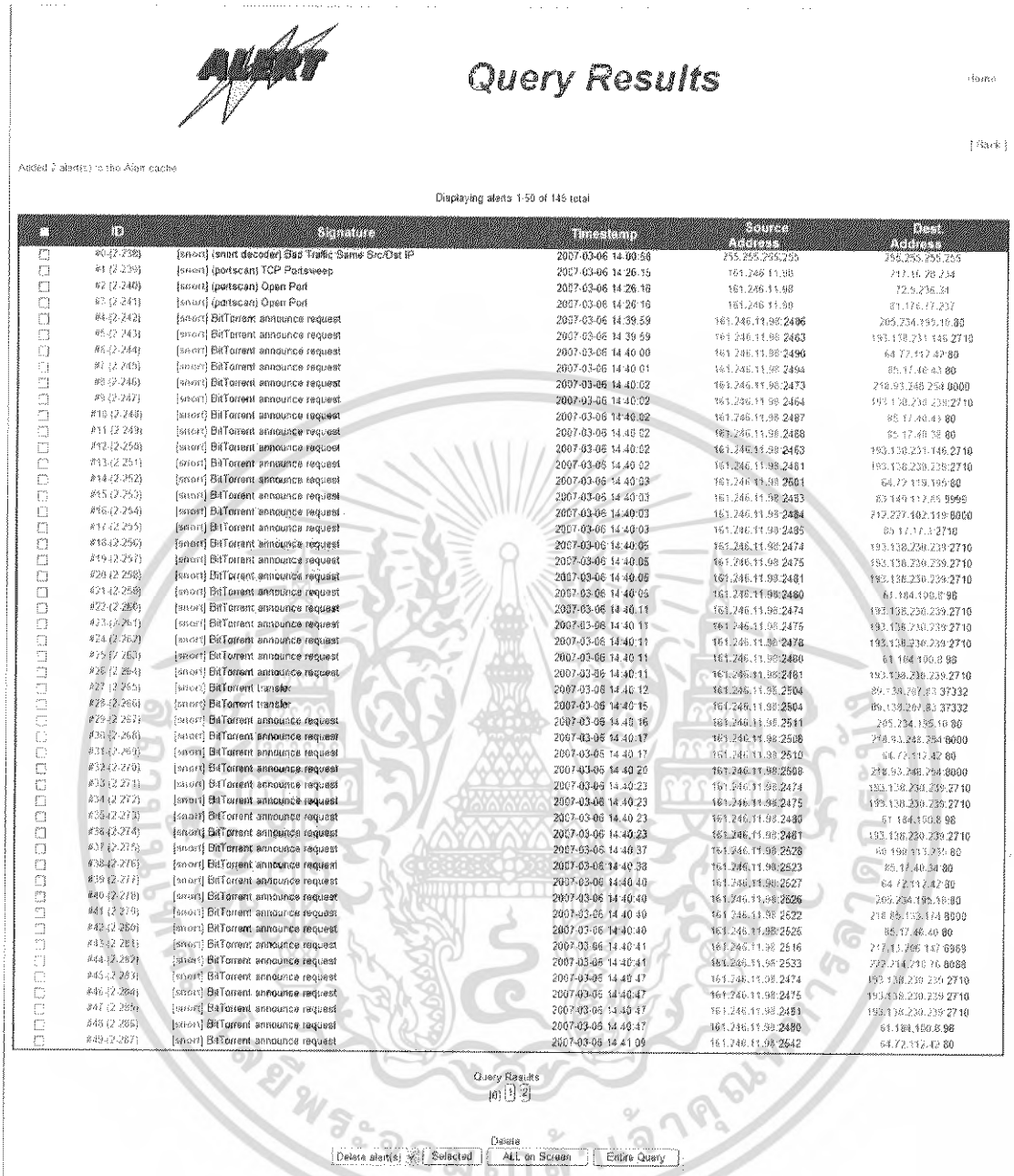
รูปที่ 4.9 หน้าจอแสดงผลโดยเรียงแพ็กเก็ต 15 แพ็กเก็ตล่าสุด

ระบบสามารถแสดงผลของการตรวจจับได้ในวันนี้โดยแสดงผลที่ไม่ซ้ำกันออกมา ดังรูป



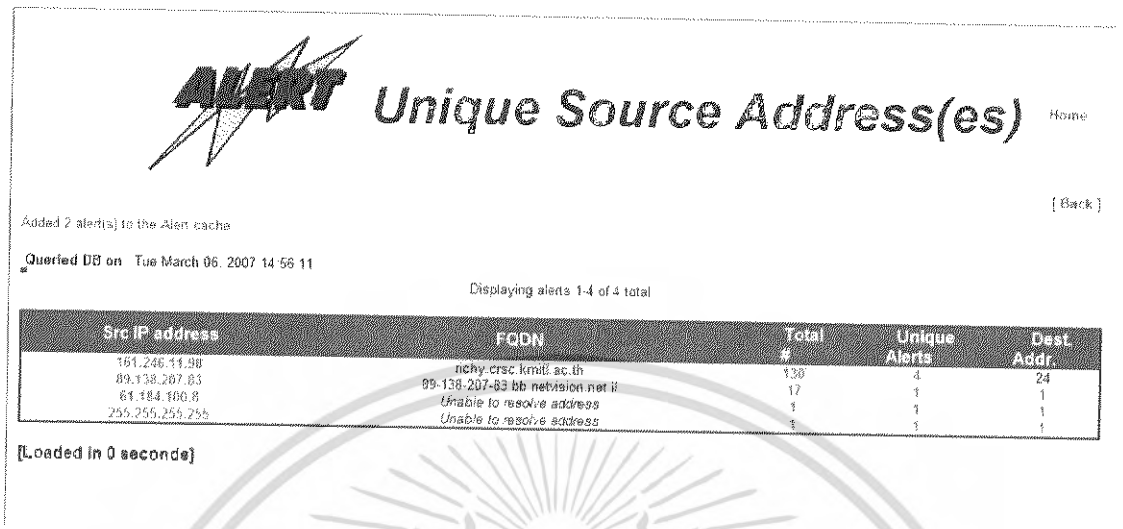
รูปที่ 4.10 หน้าจอแสดงผลของการตรวจจับโดยแสดงซิกเนเจอร์ที่ไม่ซ้ำกันในวันนี้

ระบบสามารถแสดงผลของการตรวจจับในวันนี้ได้ทั้งหมด ดังรูป



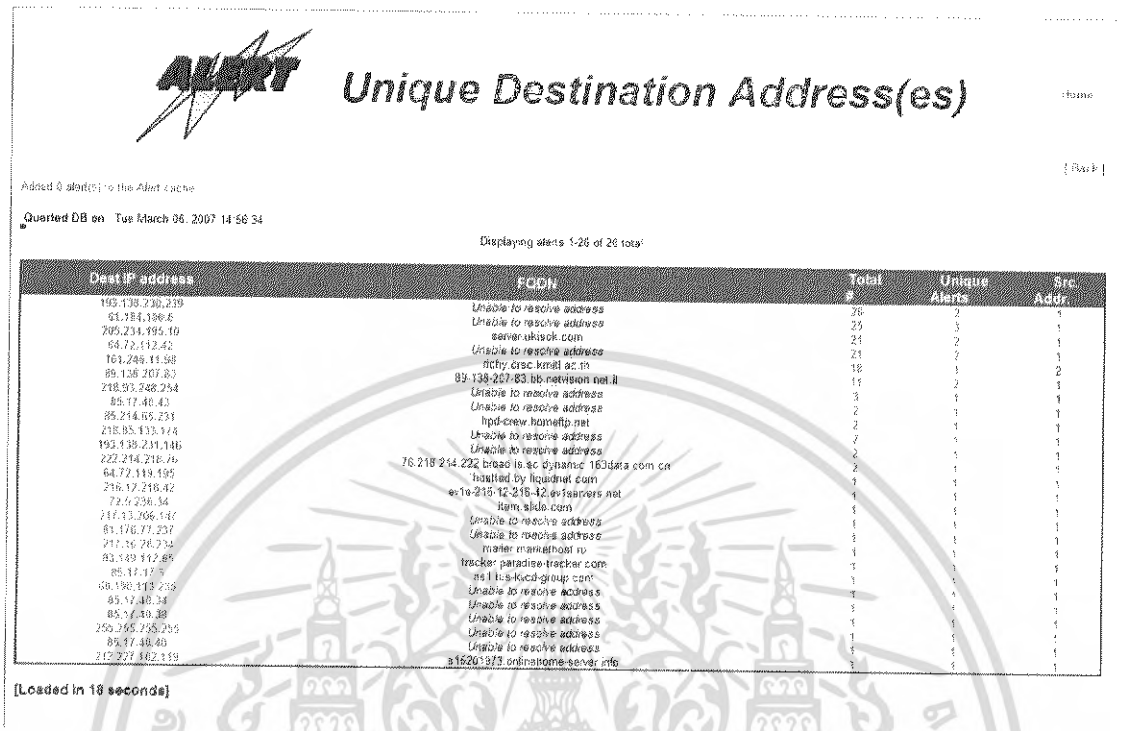
รูปที่ 4.11 หน้าจอแสดงผลของการตรวจจับทั้งหมดที่ตรวจจับได้ในวันนี้

ระบบสามารถแสดงผลของการตรวจจับวันนี้โดยพิจารณาจากไอพีแอดเดรสต้นทาง ดังรูป



รูปที่ 4.12 หน้าจอแสดงผลของการตรวจจับวันนี้โดยพิจารณาจากไอพีแอดเดรสต้นทาง

ระบบสามารถแสดงผลของการตรวจจับในวันนี้โดยพิจารณาจากไอพีแอดเดรสปลายทาง ดังรูป



ALERT

Unique Destination Address(es)

Home

[Back]

Added 0 alert(s) to the Alert cache

Queried DB on : Tue March 06, 2007 14:56:34

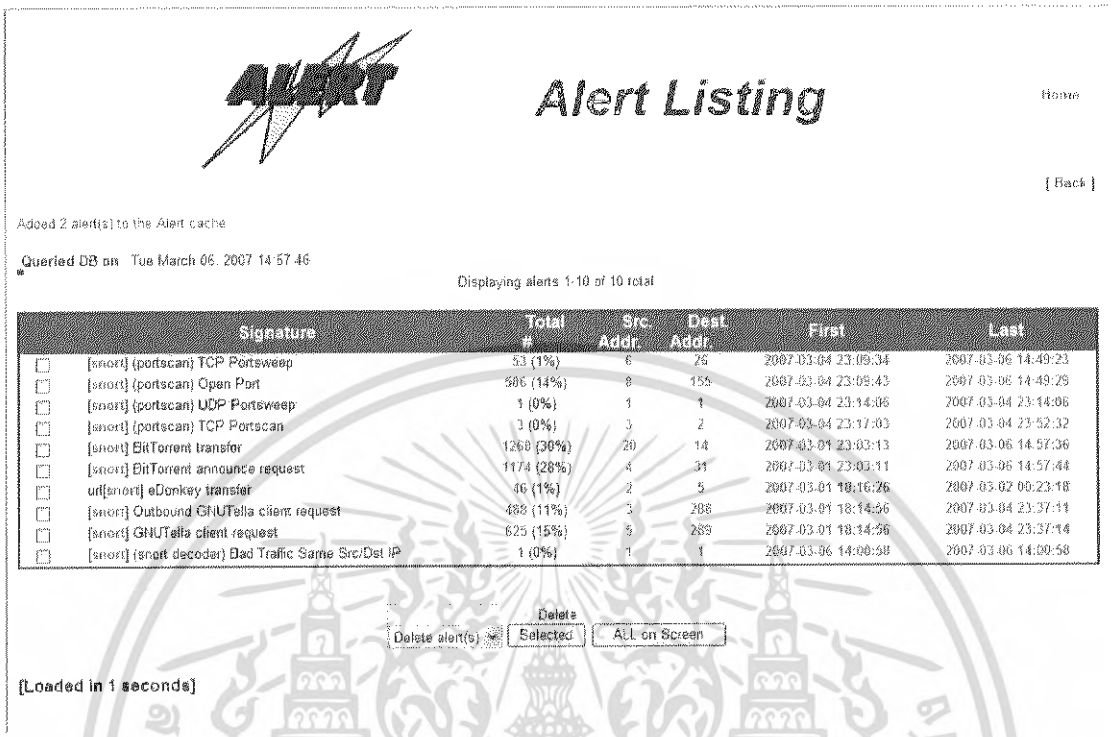
Displaying alerts 1-26 of 26 total

Dest IP address	FQDN	Total #	Unique Alerts	Src. Addr.
193.138.210.219	Unable to resolve address	26	2	1
61.184.186.6	Unable to resolve address	25	3	1
205.234.195.10	server.ukisvk.com	24	2	1
64.72.112.42	Unable to resolve address	21	2	1
161.246.11.58	atchy.csc.kmitl.ac.th	18	3	2
89.138.207.83	89-138-207-83.bb.netvision.net.il	11	2	1
218.93.248.254	Unable to resolve address	3	1	1
85.17.46.43	Unable to resolve address	2	1	1
85.214.65.231	hpd-crew.homeftp.net	2	1	1
218.85.133.174	Unable to resolve address	2	1	1
193.138.231.146	Unable to resolve address	2	1	1
222.214.218.76	76.218.214.222.broad.ls.sc.dynamic.163data.com.cn	2	1	1
64.72.119.195	hosted.by.liquidnet.com	1	1	1
216.17.218.42	ea1a-215-12-218-42.evrsaimers.net	1	1	1
72.0.236.34	item.slide.com	1	1	1
217.13.206.147	Unable to resolve address	1	1	1
81.176.77.237	Unable to resolve address	1	1	1
217.26.28.234	mailer.mainethost.ru	1	1	1
83.149.112.85	tracker.paradise-tracker.com	1	1	1
85.17.17.5	as11.s-kod-group.com	1	1	1
68.180.113.238	Unable to resolve address	1	1	1
85.17.46.34	Unable to resolve address	1	1	1
85.17.46.38	Unable to resolve address	1	1	1
255.255.255.255	Unable to resolve address	1	1	1
85.17.46.46	Unable to resolve address	1	1	1
212.227.182.119	a1520.1f13.onlinehome-server.info	1	1	1

[Loaded in 18 seconds]

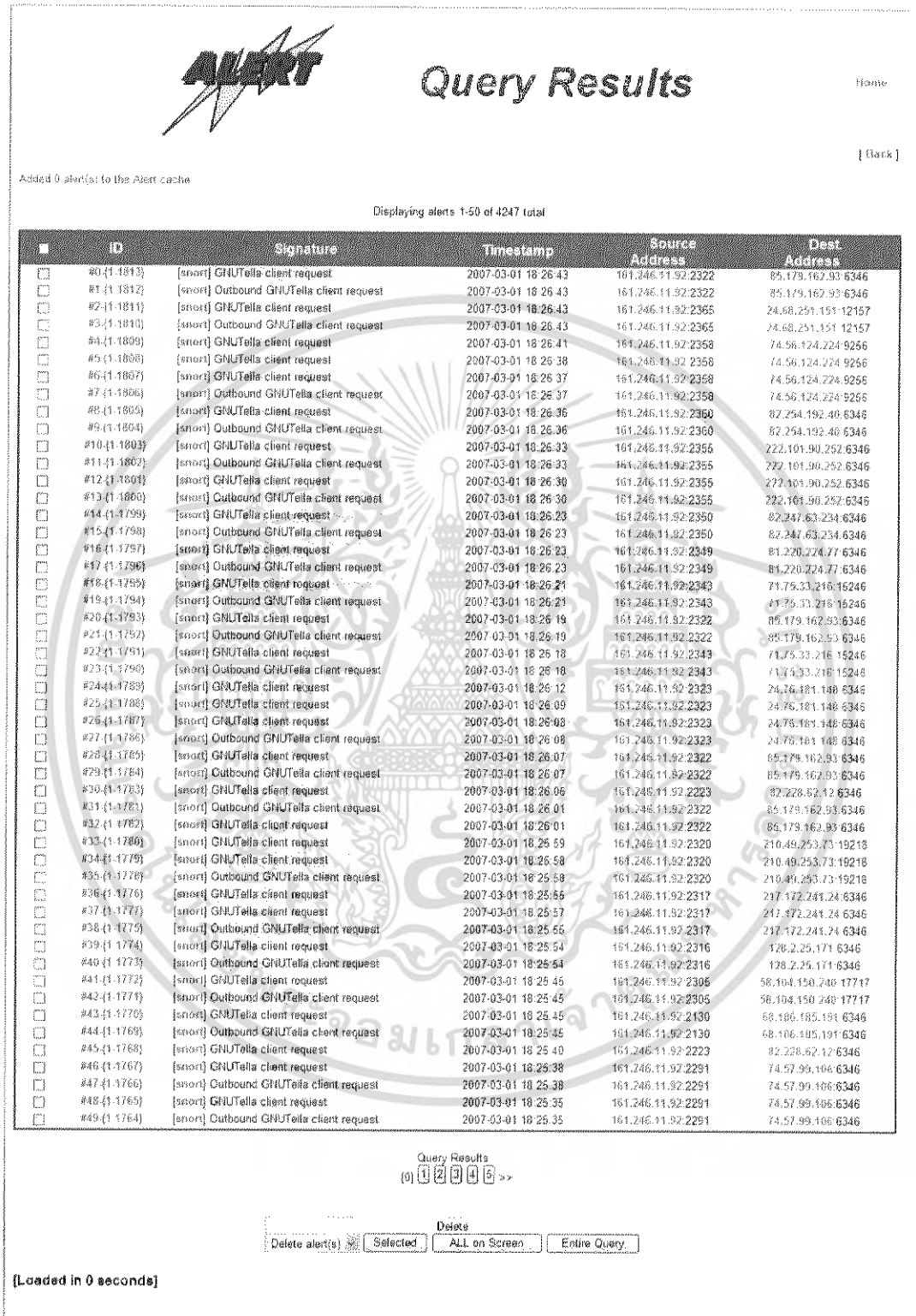
รูปที่ 4.13 หน้าจอแสดงผลของการตรวจจับในวันนี้โดยพิจารณาจากไอพีแอดเดรสปลายทาง

ระบบสามารถแสดงผลของการตรวจจับที่เกิดขึ้นภายใน 24 ชั่วโมงล่าสุด โดยแสดงผลที่ไม่ซ้ำกันออกมา ดังรูป



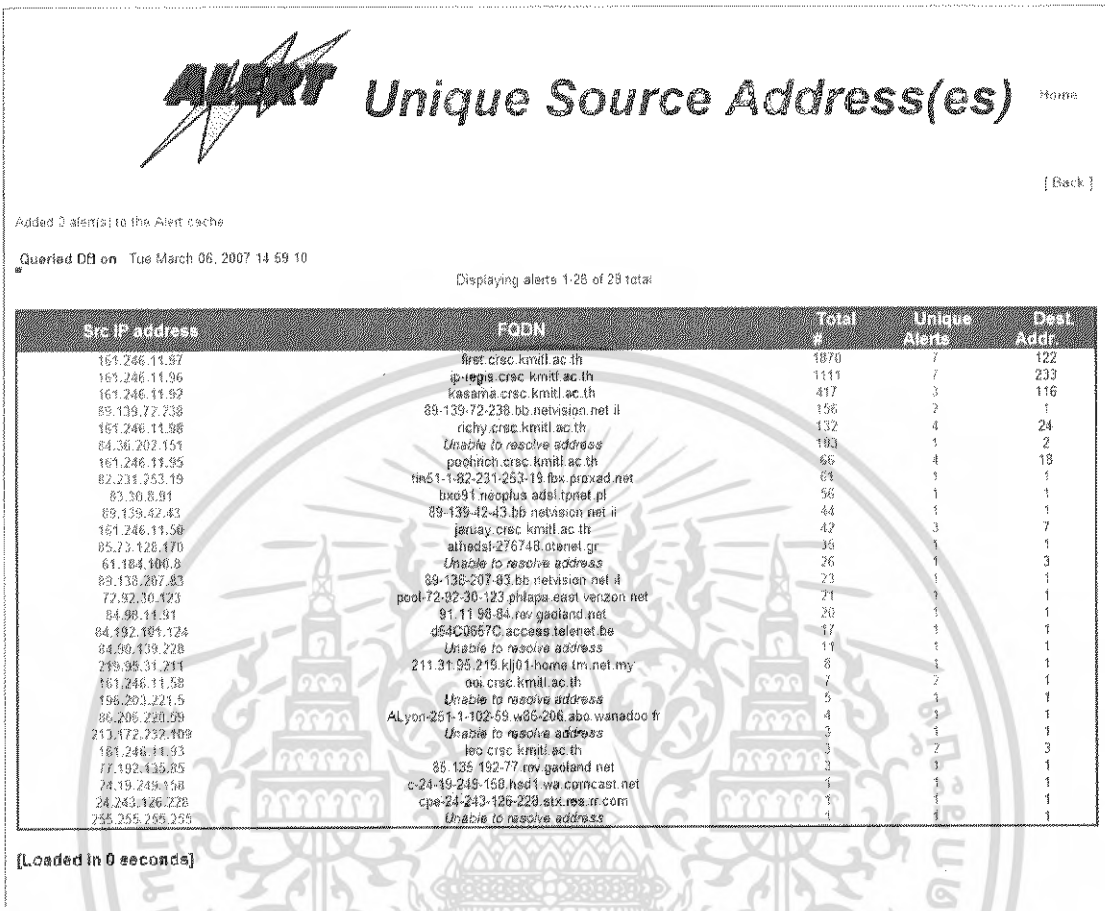
รูปที่ 4.14 หน้าจอแสดงผลของการตรวจจับโดยแสดงซิกเนเจอร์ที่ไม่ซ้ำกันและเป็นรายการที่เกิดขึ้น ใน 24 ชั่วโมงล่าสุด

ระบบสามารถแสดงผลของการตรวจจับที่เกิดขึ้นภายใน 24 ชั่วโมงล่าสุดโดยแสดงรายการที่ตรวจจับได้ทั้งหมดทั้งหมด ดังรูป



รูปที่ 4.15 หน้าจอแสดงผลของการตรวจจับทั้งหมดที่ตรวจจับได้ใน 24 ชั่วโมงล่าสุด

ระบบสามารถแสดงผลของการตรวจจับใน 24 ชั่วโมงล่าสุดโดยพิจารณาจากไอพีแอดเดรสต้นทาง ดังรูป



ALERT Unique Source Address(es) Home

[Back]

Added 0 alert(s) to the Alert cache

Queried DB on Tue March 06, 2007 14:59:10

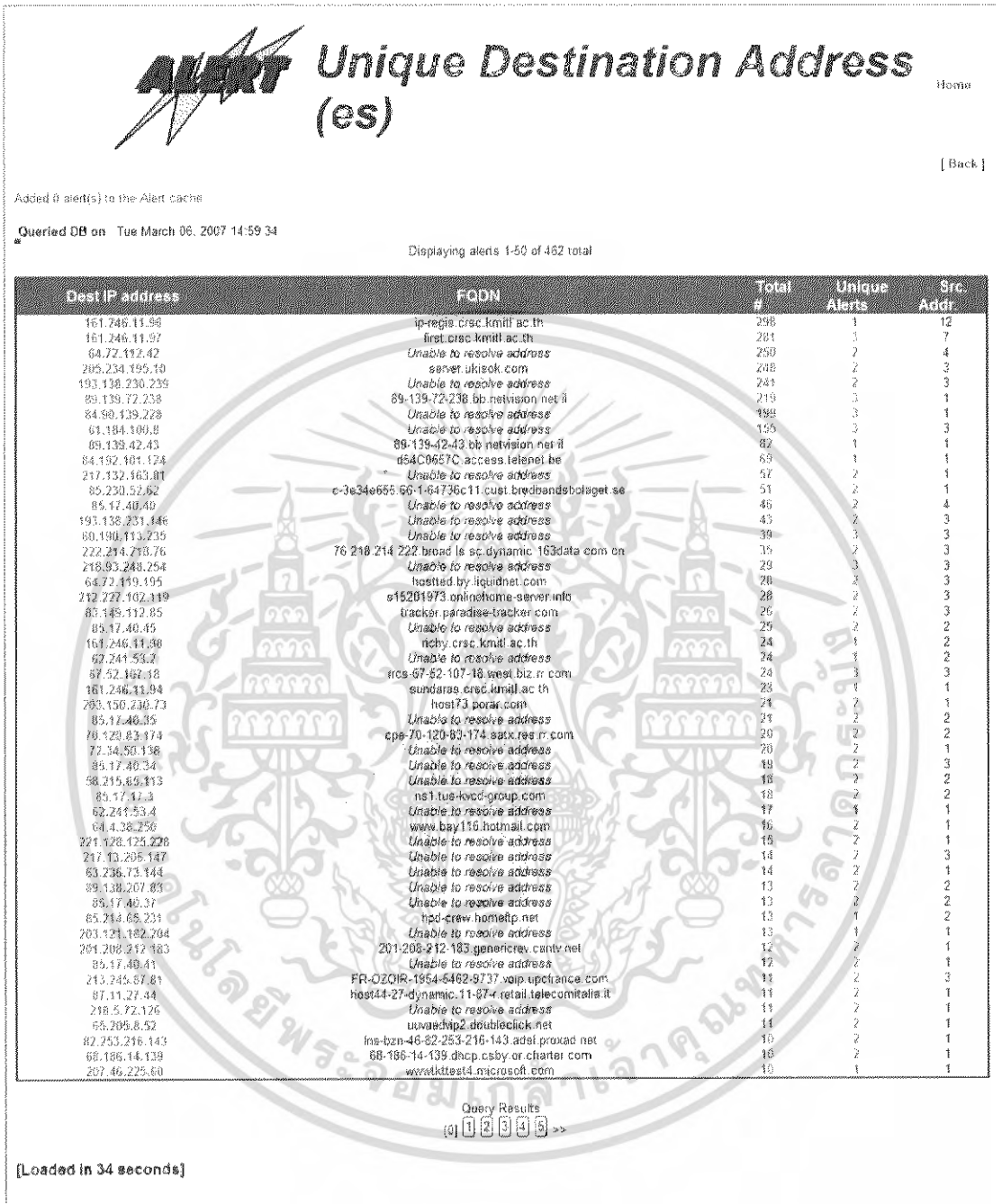
Displaying alerts 1-28 of 28 total

Src IP address	FQDN	Total #	Unique Alerts	Dest. Addr.
161.246.11.97	first.crsc.kmitl.ac.th	1870	7	122
161.246.11.96	ip-regis.crsc.kmitl.ac.th	1111	7	233
161.246.11.92	kasama.crsc.kmitl.ac.th	417	3	116
89.139.72.238	89-139-72-238.bb.netvision.net.il	156	2	1
161.246.11.98	richy.crsc.kmitl.ac.th	132	4	24
84.36.202.151	*Unable to resolve address*	103	1	2
161.246.11.95	poohrich.crsc.kmitl.ac.th	66	4	18
82.231.253.19	tln51-1-82-231-253-19.fbx.proxad.net	61	1	1
83.30.8.91	bxo91.neoplus.adsl.tpnet.pl	56	1	1
89.139.42.43	89-139-42-43.bb.netvision.net.il	44	1	1
161.246.11.50	jaruay.crsc.kmitl.ac.th	42	3	7
85.73.128.170	athedsl-276748.otenet.gr	36	1	1
61.184.100.8	*Unable to resolve address*	26	1	3
89.138.207.83	89-138-207-83.bb.netvision.net.il	23	1	1
72.92.30.123	pool-72-92-30-123.phlapa.east.verizon.net	21	1	1
84.98.11.91	91.11.98-84.rev.gaoland.net	20	1	1
84.192.181.124	d54C0557C.access.telenet.be	17	1	1
84.90.139.228	*Unable to resolve address*	11	1	1
219.95.31.211	211.31.95.215.klj01-home.tm.net.my	8	1	1
161.246.11.58	ooi.crsc.kmitl.ac.th	7	2	1
196.203.221.5	*Unable to resolve address*	5	1	1
86.206.220.59	ALyon-251-1-102-59.w86-206.abo.wanadoo.fr	4	1	1
213.172.232.109	*Unable to resolve address*	3	1	1
161.246.11.93	leo.crsc.kmitl.ac.th	3	2	3
77.192.115.85	85.115.192-77.rev.gaoland.net	2	1	1
74.19.249.158	c-24-19-249-158.hsd1.wa.comcast.net	1	1	1
24.243.126.228	cpe-24-243-126-228.stx.res.rr.com	1	1	1
255.255.255.255	*Unable to resolve address*	1	1	1

(Loaded in 0 seconds)

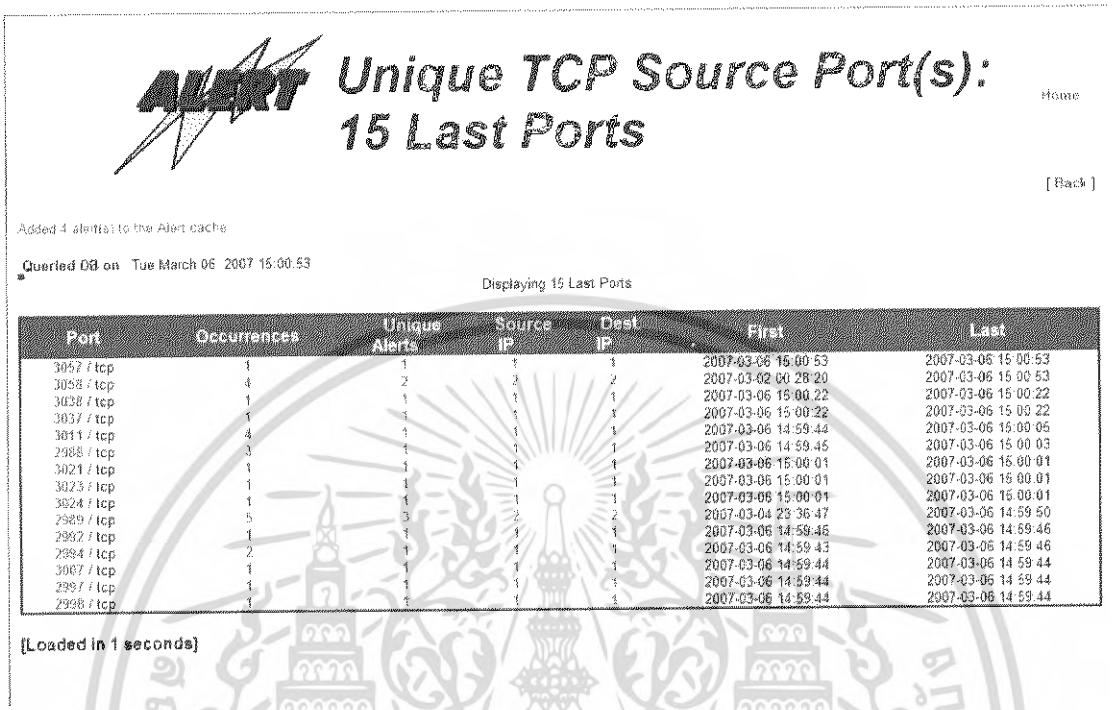
รูปที่ 4.16 หน้าจอแสดงผลของการตรวจจับ 24 ชั่วโมงล่าสุดโดยพิจารณาจากไอพีแอดเดรสต้นทาง

ระบบสามารถแสดงผลของการตรวจจับใน 24 ชั่วโมงล่าสุดโดยพิจารณาจากไอพีแอดเดรสปลายทาง ดังรูป



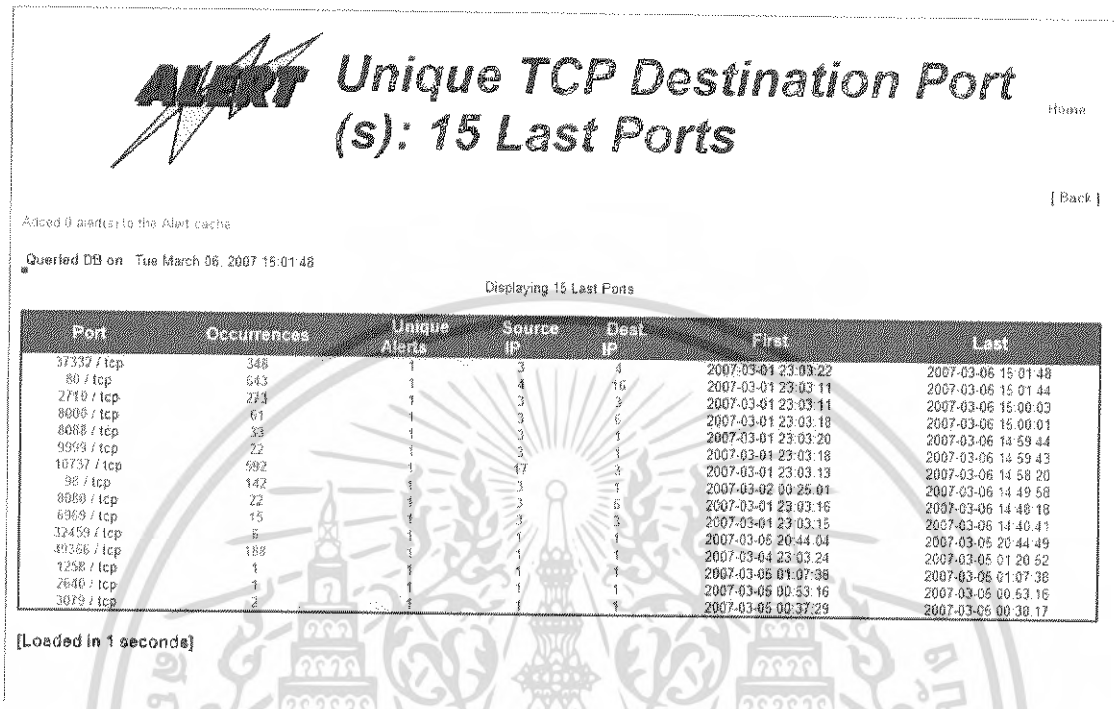
รูปที่ 4.17 หน้าจอแสดงผลของการตรวจจับ 24 ชั่วโมงล่าสุดโดยพิจารณาจากไอพีแอดเดรสปลายทาง

ระบบสามารถแสดงผลของการตรวจจับโดยแสดงหมายเลขพอร์ตของต้นทางที่ถูกตรวจจับนั้น จำนวนที่ถูกแจ้งเตือน วันและเวลาที่แพ็กเก็ตแรกและสุดท้ายที่ถูกตรวจจับ ซึ่งจะแสดง 15 รายการล่าสุดที่ถูกตรวจจับ ดังรูป



รูปที่ 4.18 หน้าจอแสดงผลของการตรวจจับโดยแสดงถึง 15 รายการล่าสุดที่พอร์ตต้นทางถูกแจ้งเตือน

ระบบสามารถแสดงผลของการตรวจจับโดยแสดงหมายเลขพอร์ตของปลายทางที่ถูกตรวจจับนั้น จำนวนที่ถูกแจ้งเตือน วันและเวลาที่แพ็กเก็ตแรกและสุดท้ายที่ถูกตรวจจับ ซึ่งจะแสดง 15 รายการล่าสุดที่ถูกตรวจจับ ดังรูป



ALERT **Unique TCP Destination Port (s): 15 Last Ports**

Home

[Back]

Added 0 alert(s) to the Alert cache

Queried DB on Tue March 06, 2007 15:01:48

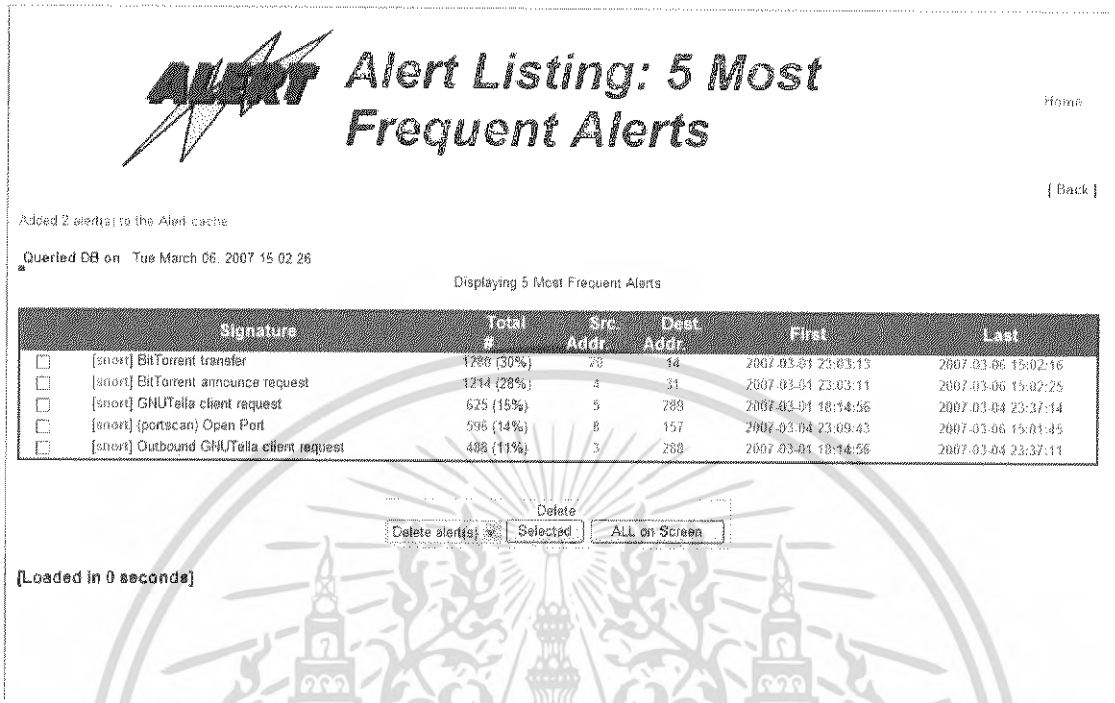
Displaying 15 Last Ports

Port	Occurrences	Unique Alerts	Source IP	Dest. IP	First	Last
37337 / tcp	348	1	3	4	2007-03-01 23:03:22	2007-03-06 15:01:48
80 / tcp	643	1	4	16	2007-03-01 23:03:11	2007-03-06 15:01:44
2719 / tcp	231	1	3	3	2007-03-01 23:03:11	2007-03-06 15:00:03
8009 / tcp	61	1	3	6	2007-03-01 23:03:18	2007-03-06 15:00:01
8088 / tcp	33	1	3	1	2007-03-01 23:03:20	2007-03-06 14:59:44
9999 / tcp	22	1	3	1	2007-03-01 23:03:18	2007-03-06 14:59:43
10737 / tcp	592	1	17	3	2007-03-01 23:03:13	2007-03-06 14:58:20
98 / tcp	142	1	3	1	2007-03-02 00:25:01	2007-03-06 14:49:58
8080 / tcp	22	1	3	6	2007-03-01 23:03:16	2007-03-06 14:48:18
6969 / tcp	15	1	3	3	2007-03-01 23:03:15	2007-03-06 14:40:41
32459 / tcp	6	1	1	1	2007-03-05 20:44:04	2007-03-05 20:44:49
49386 / tcp	188	1	1	1	2007-03-04 23:03:24	2007-03-05 01:20:52
1258 / tcp	1	1	1	1	2007-03-05 01:07:38	2007-03-05 01:07:38
2640 / tcp	1	1	1	1	2007-03-05 00:53:16	2007-03-05 00:53:16
3079 / tcp	2	1	1	1	2007-03-05 00:37:29	2007-03-05 00:38:17

[Loaded in 1 seconds]

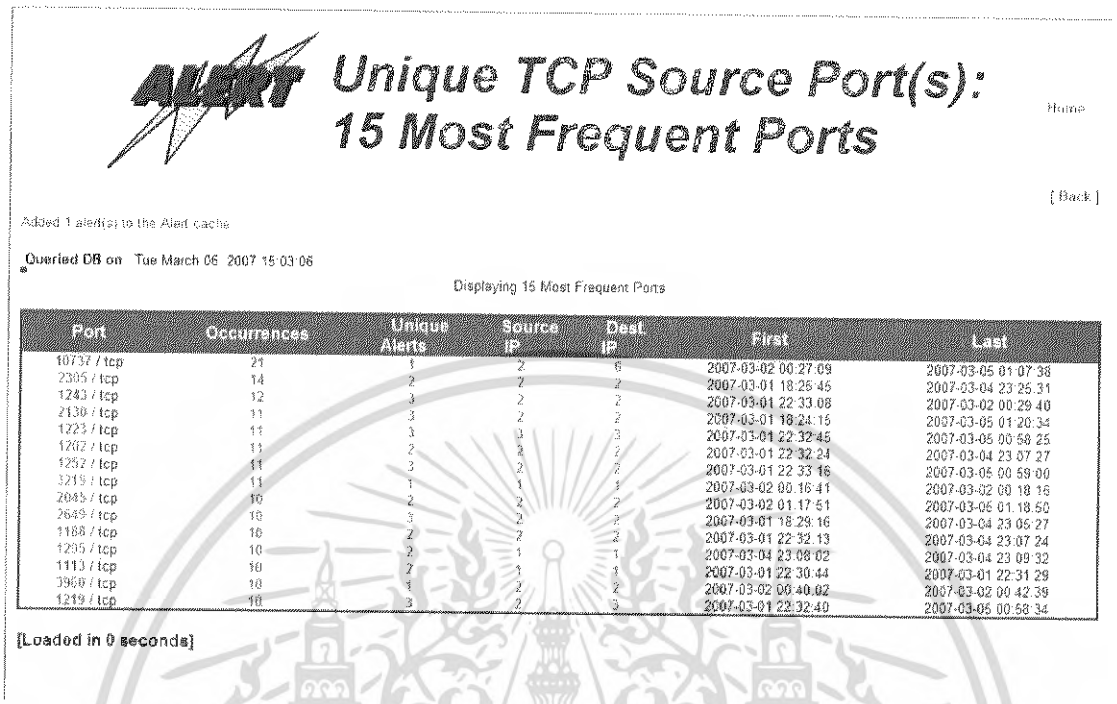
รูปที่ 4.19 หน้าจอแสดงผลของการตรวจจับโดยแสดงถึง 15 รายการล่าสุดที่พอร์ตปลายทางถูกแจ้งเตือน

ระบบสามารถแสดงผลของการตรวจจับโดยแสดงประเภทของการแจ้งเตือนที่ตรวจจับพบบ่อยที่สุด 5 อันดับ ดังรูป



รูปที่ 4.20 หน้าจอแสดงผลของการตรวจจับโดยแสดงประเภทของการแจ้งเตือนที่พบบ่อยที่สุด 5 อันดับ

ระบบสามารถแสดงผลของการตรวจจับโดยแสดงหมายเลขพอร์ตของต้นทางที่ถูกแจ้งเตือนบ่อยที่สุด 15 อันดับ ดังรูป



ALERT Unique TCP Source Port(s): 15 Most Frequent Ports

Home

[Back]

Added 1 alert(s) to the Alert cache

Queried DB on: Tue March 06, 2007 15:03:06

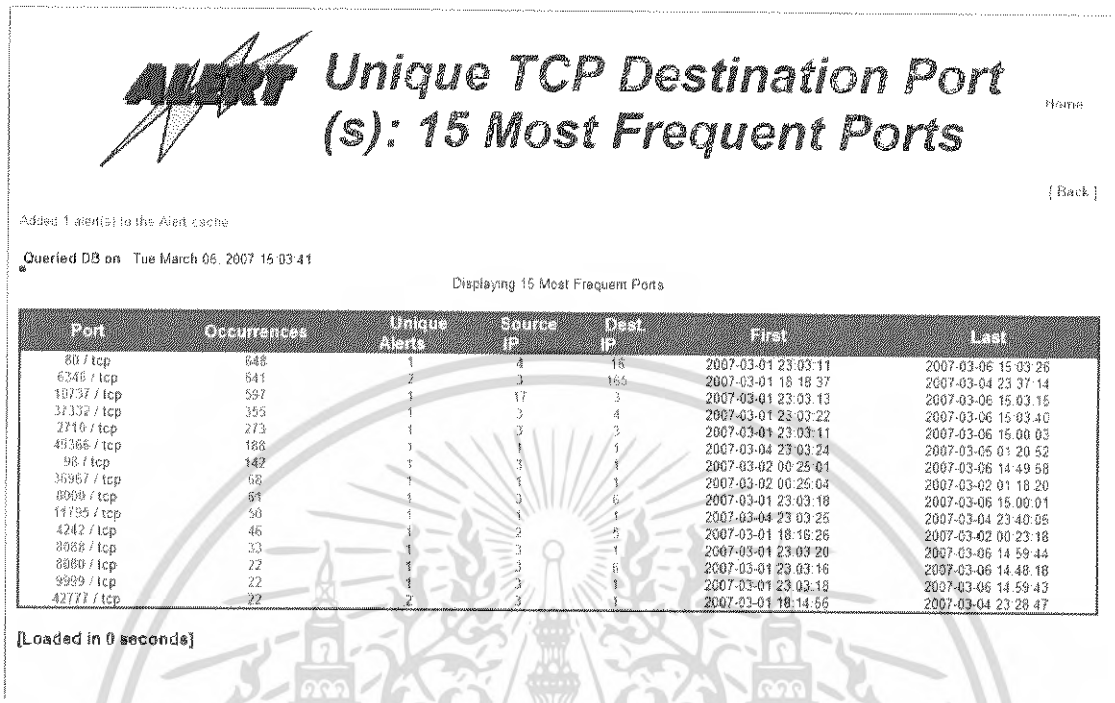
Displaying 15 Most Frequent Ports

Port	Occurrences	Unique Alerts	Source IP	Dest. IP	First	Last
10737 / tcp	21	1	2	6	2007-03-02 00:27:09	2007-03-05 01:07:38
2305 / tcp	14	2	2	2	2007-03-01 18:25:46	2007-03-04 23:25:31
1243 / tcp	12	3	2	2	2007-03-01 22:33:08	2007-03-02 00:29:40
2130 / tcp	11	3	2	2	2007-03-01 18:24:15	2007-03-05 01:20:34
1223 / tcp	11	3	3	3	2007-03-01 22:32:45	2007-03-05 00:58:25
1262 / tcp	11	2	2	2	2007-03-01 22:32:24	2007-03-04 23:07:27
1252 / tcp	11	3	2	2	2007-03-01 22:33:16	2007-03-05 00:59:00
3219 / tcp	11	1	1	1	2007-03-02 00:16:41	2007-03-02 00:18:16
2045 / tcp	10	2	2	2	2007-03-02 01:17:51	2007-03-05 01:18:50
2649 / tcp	10	3	2	2	2007-03-01 18:29:16	2007-03-04 23:05:27
1188 / tcp	10	2	2	2	2007-03-01 22:32:13	2007-03-04 23:07:24
1205 / tcp	10	2	1	1	2007-03-04 23:08:02	2007-03-04 23:09:32
1113 / tcp	10	2	1	1	2007-03-01 22:30:44	2007-03-01 22:31:29
3960 / tcp	10	1	2	2	2007-03-02 00:40:02	2007-03-02 00:42:39
1219 / tcp	10	3	2	3	2007-03-01 22:32:40	2007-03-05 00:58:34

[Loaded in 0 seconds]

รูปที่ 4.21 หน้าจอแสดงผลของการตรวจจับโดยแสดงหมายเลขพอร์ตของต้นทางที่ถูกแจ้งเตือนบ่อยที่สุด 15 อันดับ

ระบบสามารถแสดงผลของการตรวจจับโดยแสดงหมายเลขพอร์ตของปลายทางที่ถูกแจ้งเตือนบ่อยที่สุด 15 อันดับ ดังรูป



Unique TCP Destination Port (s): 15 Most Frequent Ports

Home

[Back]

Added 1 alert(s) to the Alert cache

Queried DB on: Tue March 06, 2007 15:03:41

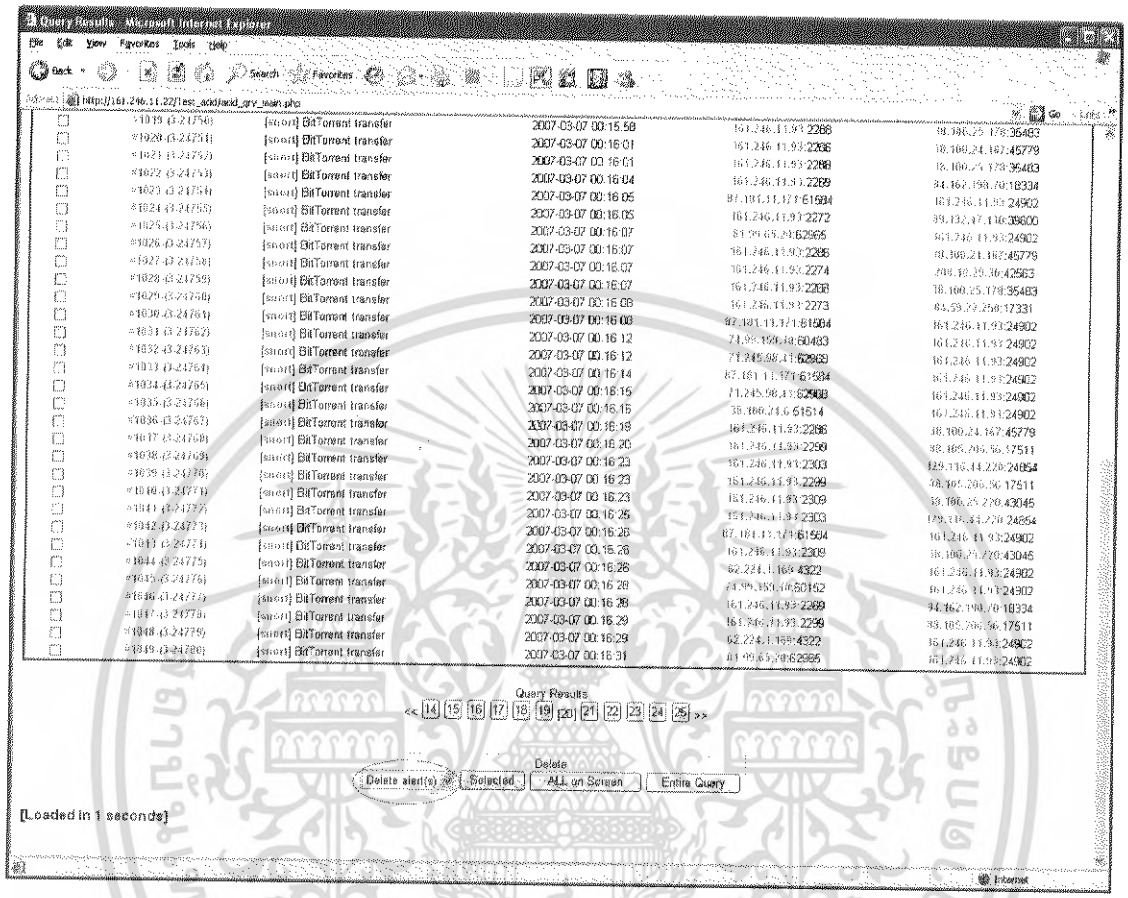
Displaying 15 Most Frequent Ports

Port	Occurrences	Unique Alerts	Source IP	Dest. IP	First	Last
80 / tcp	648	1	4	16	2007-03-01 23:03:11	2007-03-06 15:03:26
6346 / tcp	641	2	3	166	2007-03-01 18:18:37	2007-03-04 23:37:14
18737 / tcp	597	1	17	3	2007-03-01 23:03:13	2007-03-06 15:03:15
37332 / tcp	356	1	3	4	2007-03-01 23:03:22	2007-03-06 15:03:40
2710 / tcp	273	1	3	3	2007-03-01 23:03:11	2007-03-06 15:00:03
49366 / tcp	188	1	1	1	2007-03-04 23:03:24	2007-03-05 01:20:52
98 / tcp	142	1	3	1	2007-03-02 00:25:01	2007-03-06 14:49:58
36967 / tcp	68	1	1	1	2007-03-02 00:25:04	2007-03-02 01:18:20
8069 / tcp	64	1	3	6	2007-03-01 23:03:18	2007-03-06 15:00:01
11795 / tcp	50	1	1	1	2007-03-04 23:03:25	2007-03-04 23:40:06
4242 / tcp	46	1	2	2	2007-03-01 18:16:26	2007-03-02 00:23:18
8088 / tcp	33	1	3	1	2007-03-01 23:03:20	2007-03-06 14:59:44
8080 / tcp	22	1	3	6	2007-03-01 23:03:16	2007-03-06 14:48:18
9999 / tcp	22	1	3	1	2007-03-01 23:03:18	2007-03-06 14:59:43
42777 / tcp	22	2	3	1	2007-03-01 18:14:56	2007-03-04 23:28:47

[Loaded in 0 seconds]

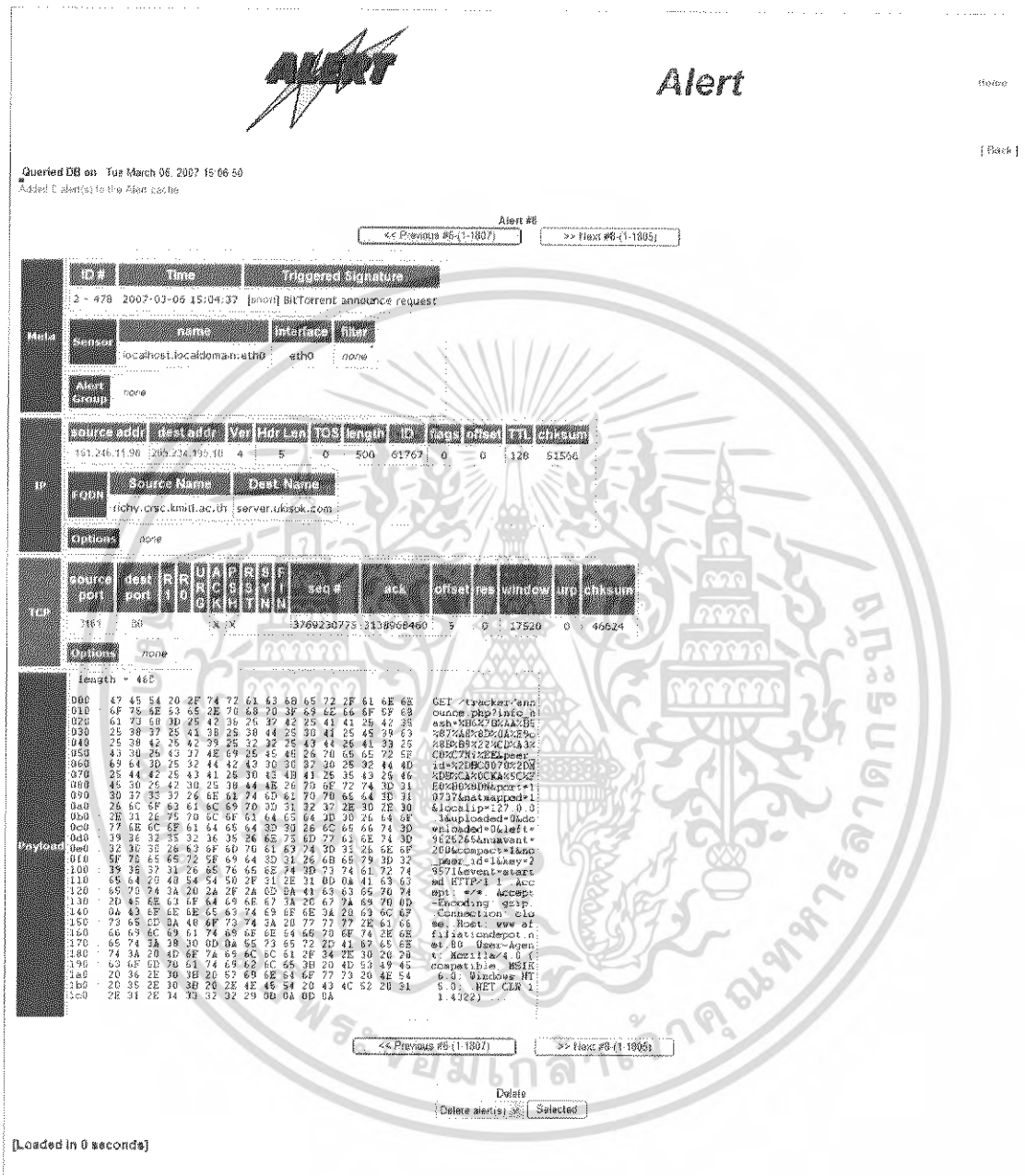
รูปที่ 4.22 หน้าจอแสดงผลของการตรวจจับโดยแสดงหมายเลขพอร์ตของปลายทางที่ถูกแจ้งเตือนบ่อยที่สุด 15 อันดับ

ระบบสามารถทำการลบบันทึกรายการได้จากช่องแอ็คชัน (Action) ซึ่งสามารถเลือกได้ว่าจะลบรายการที่เลือกไว้ หรือรายการที่ปรากฏบนหน้าจอ หรือรายการที่ได้มาจากการดึงข้อมูลนี้ ดังรูป



รูปที่ 4.23 หน้าจอแสดงการลบบันทึกรายการของแพ็กเก็ต

ระบบสามารถแสดงผลเป็นรายละเอียดเฮดเดอร์ของแพ็กเก็ตที่ได้มา โดยจะแสดงรายละเอียดของแพ็กเก็ตนั้นด้วย เช่น เวลาที่แพ็กเก็ตผ่านเข้ามา ซิกเนเจอร์ (Signature) ไอพีต้นทาง และไอพีปลายทาง เป็นต้น



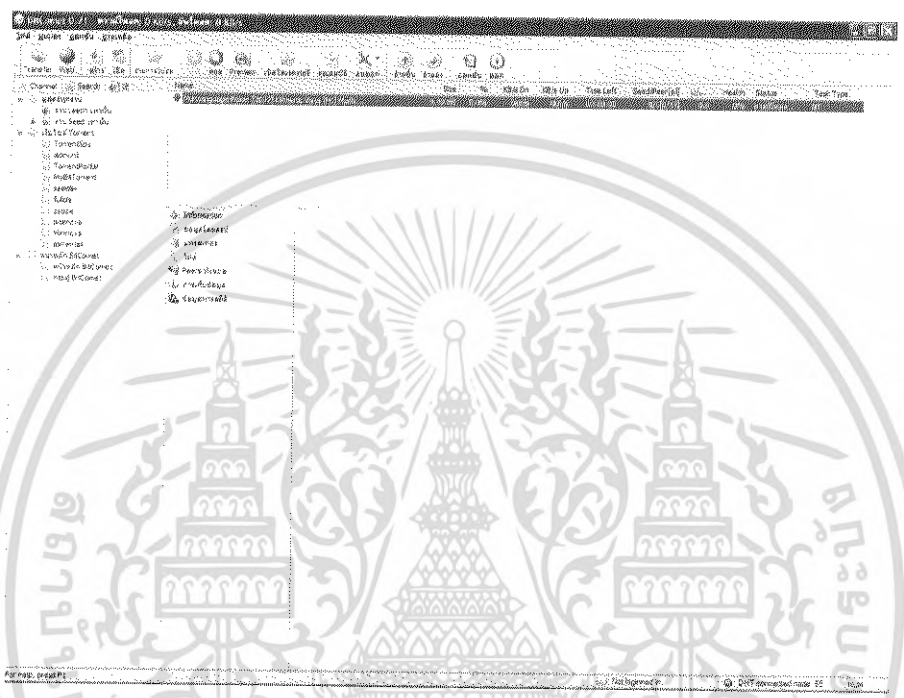
รูปที่ 4.24 หน้าจอแสดงรูปแบบของแพ็กเก็ต และรายละเอียดที่มากับแพ็กเก็ต

4.4 การทดสอบระบบ

การทดสอบโปรแกรมเราจะใช้โปรแกรมบิทโคเม็ท เวอร์ชั่น 0.72 แซดพีทูพี (ZP2P) และอีมุลล์ (eMule) ซึ่งเป็นโปรแกรมแลกเปลี่ยนไฟล์ (File-Sharing) ผ่านโปรโตคอลบิททอร์เรนท์ จีนูเทลล่า และอีดอนกี้

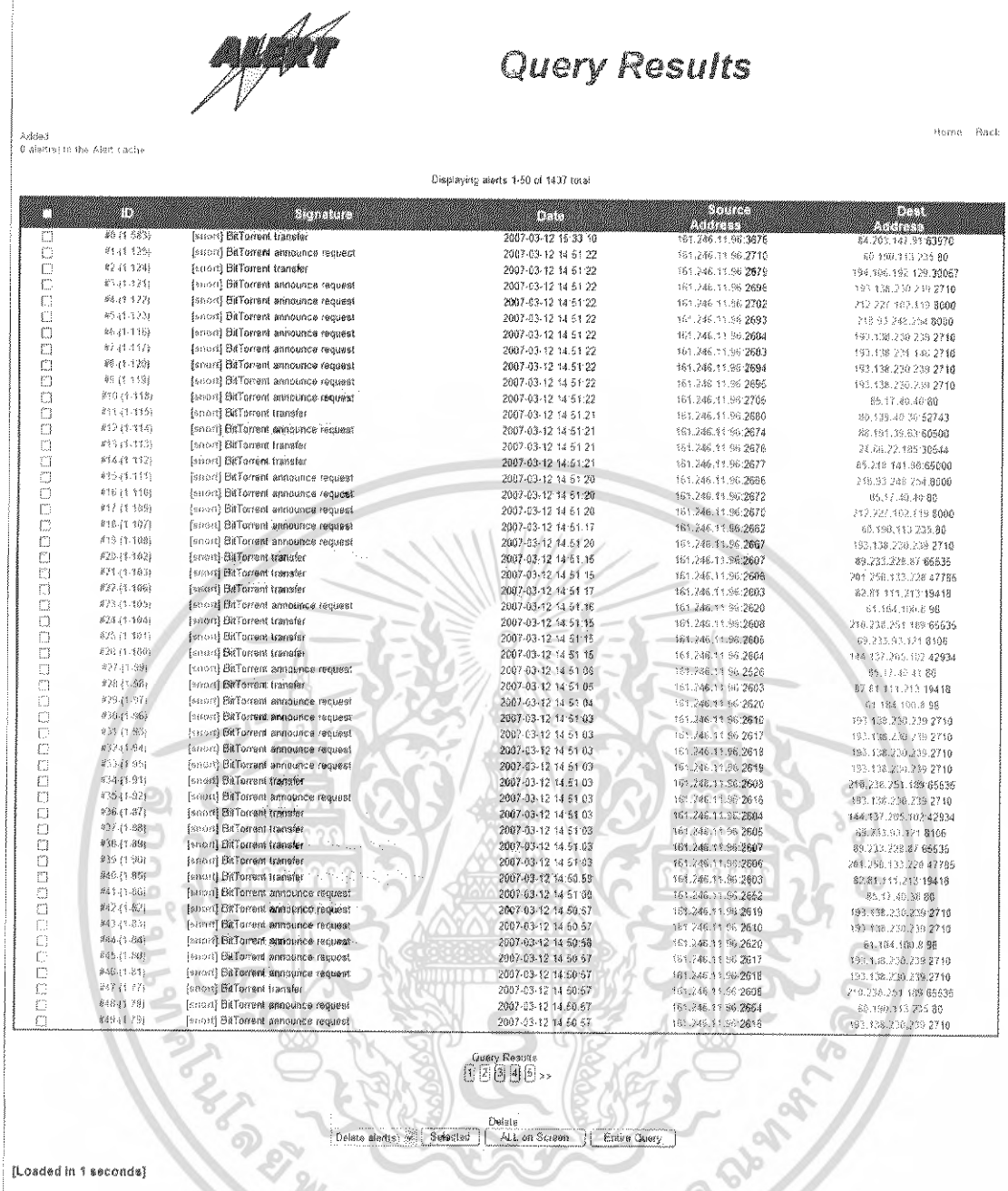
4.4.1 การทดสอบกับโปรโตคอลบิททอร์เรนท์โดยใช้โปรแกรมบิทโคเม็ท

โดยเราจะทำการโหลดไฟล์ผ่านโปรแกรมบิทโคเม็ท ดังรูป



รูปที่ 4.25 หน้าจอแสดงการโหลดไฟล์ผ่านโปรแกรมบิทโคเม็ท

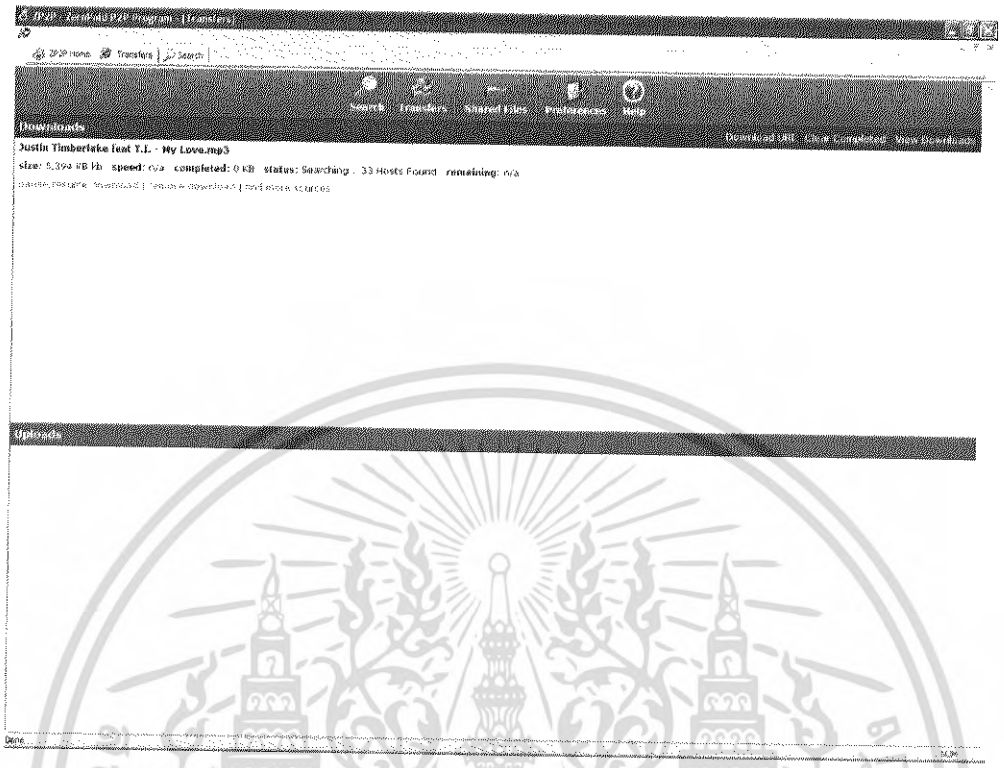
ซึ่งการโหลดไฟล์ผ่านโปรแกรมบิทโคเม็ทนั้นจะเป็นการใช้งานโปรโตคอลบิททอร์เรนท์ ซึ่งระบบของเราจะทำการตรวจจับได้ดังรูป



รูปที่ 4.26 หน้าจอแสดงผลของโปรแกรมที่ได้หลังจากการโหลดไฟล์ผ่านโปรแกรมบิทโคเม็ท

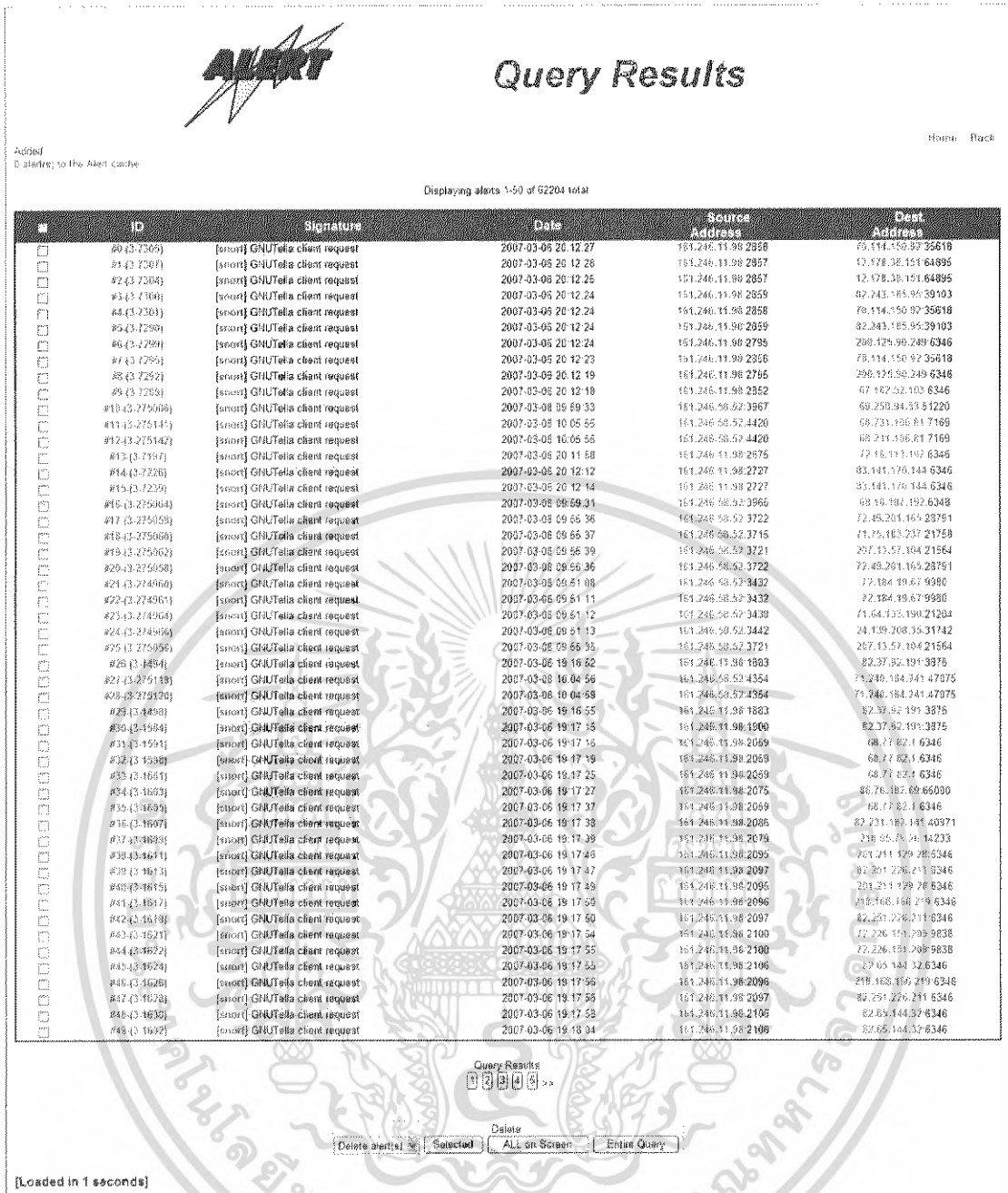
4.4.2 การทดสอบกับโปรโตคอลจีนูเทลล่าโดยใช้โปรแกรมแซดพีทูพี

โดยทำการโหลดไฟล์ด้วยโปรแกรมแซดพีทูพี ดังรูป



รูปที่ 4.27 หน้าจอแสดงการโหลดไฟล์ผ่านโปรแกรมแซดพีทูพี

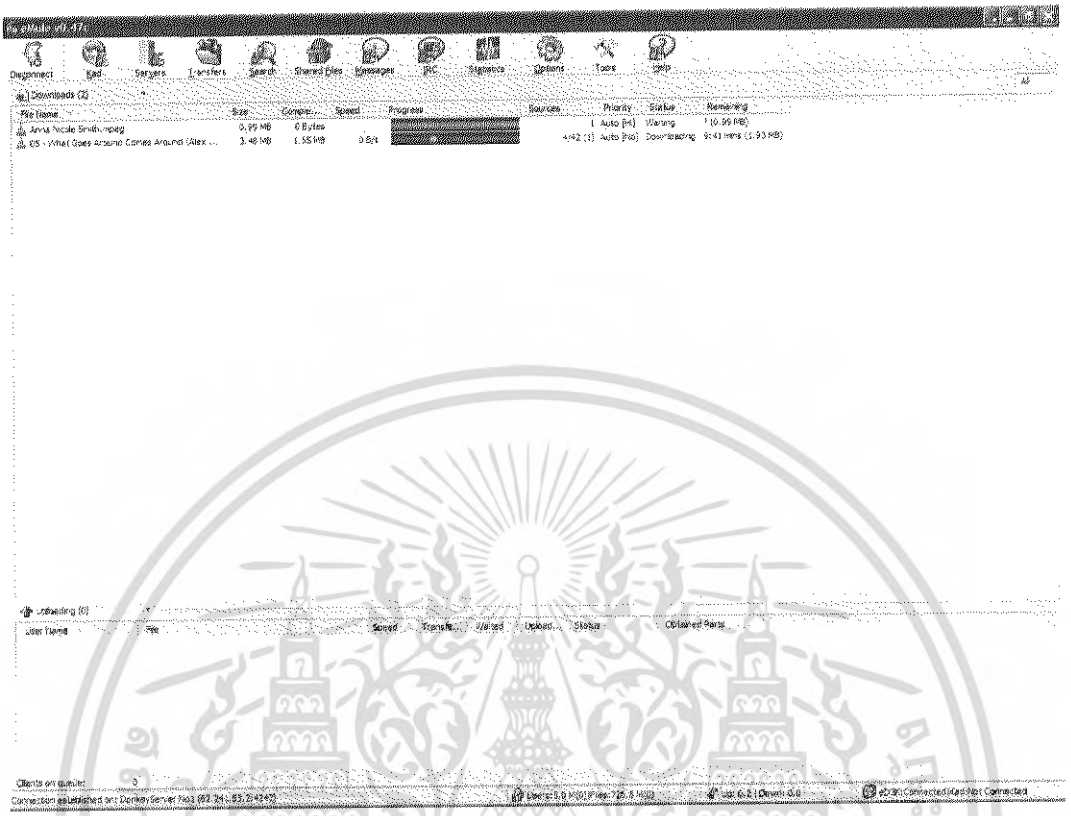
ซึ่งการโหลดไฟล์ผ่านโปรแกรมแซดพีทูพีนั้นจะเป็นการใช้งานโปรโตคอลจีนูเทลล่าซึ่งระบบของเราจะทำการตรวจจับได้ ดังรูป



รูปที่ 4.28 หน้าจอแสดงผลของโปรแกรมที่ได้หลังจากการโหลดไฟล์ผ่านโปรแกรมแซดพีทูพี

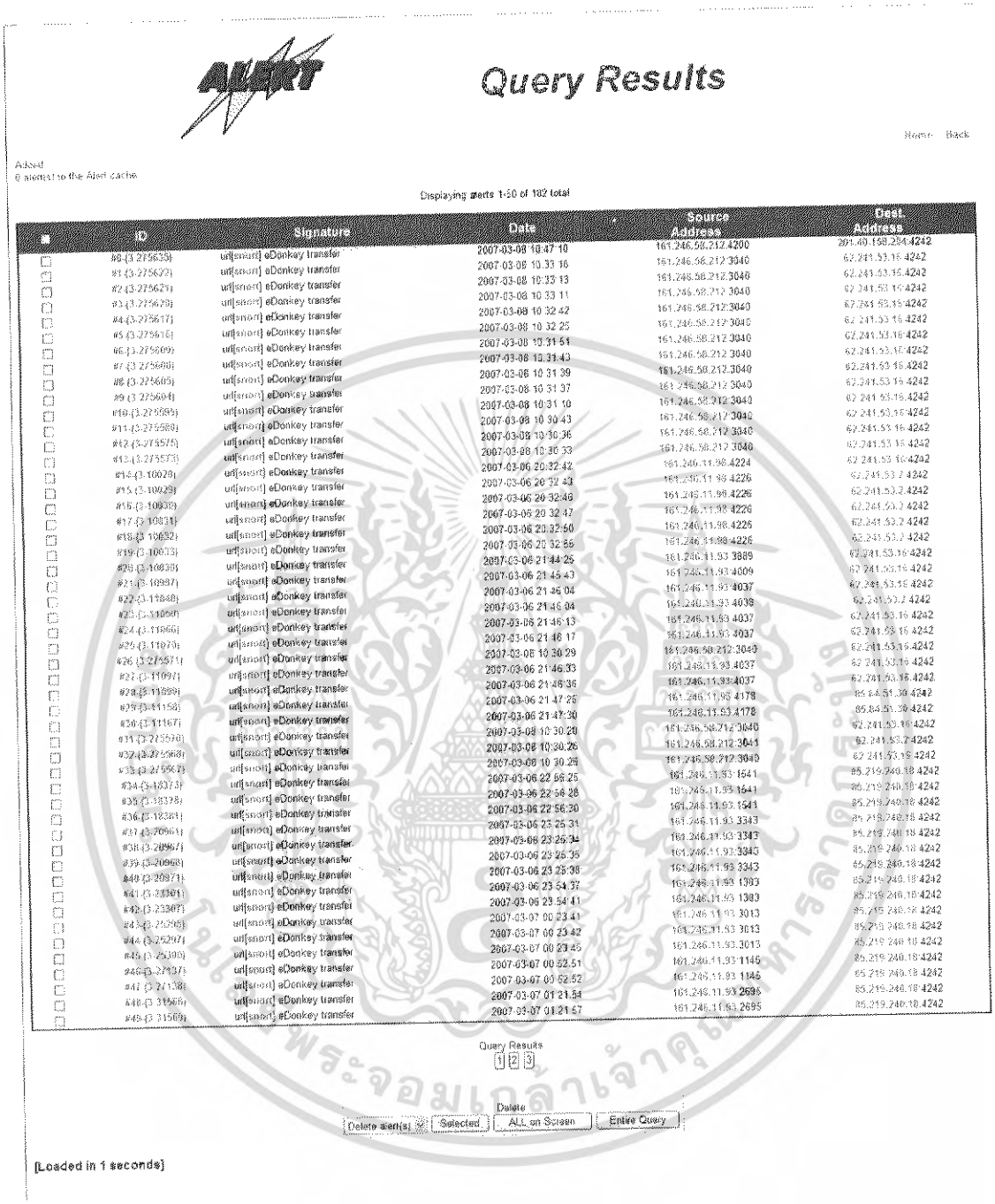
4.4.3 การทดสอบกับโปรโตคอลอีดอนกี้โดยใช้โปรแกรมอีมุลล์

โดยเราจะทำการโหลดไฟล์ผ่านโปรแกรมอีมุลล์ ดังรูป



รูปที่ 4.29 หน้าจอแสดงการโหลดไฟล์ผ่านโปรแกรมอีมุลล์

ซึ่งการโหลดไฟล์ผ่านโปรแกรมอีมุลล์นั้นจะเป็นการใช้งานโปรโตคอลอีดอนกี้ซึ่งระบบของเราจะทำการตรวจจับได้ดังรูป



รูปที่ 4.30 หน้าจอแสดงผลของโปรแกรมที่ได้หลังจากการโหลดไฟล์ผ่านโปรแกรมอีมุลล์

โดยรายชื่อแพ็คเก็ตที่พบทั้งหมด แสดงได้ดังนี้

- BitTorrent announce request
- BitTorrent transfer
- Outbound GNUTella client request
- GNUTella client request
- eDonkey transfer

4.5 สรุปผลการทำงาน

จากการทดลองข้างต้นทำให้เราทราบได้ว่าเครื่องคอมพิวเตอร์ลูกข่ายเครื่องไหนที่ทำการใช้การสื่อสารแบบเพียร์ทูเพียร์ ใช้ผ่านโปรโตคอลไหน และใช้เมื่อเวลาเท่าไหร่ ซึ่งเราสามารถนำข้อมูลที่ได้มาพิจารณาเพื่อทำการจำกัดการสื่อสารเครื่องคอมพิวเตอร์ลูกข่าย ซึ่งระบบนี้สามารถนำไปพัฒนาต่อในส่วนของการจำกัดการใช้งานเครือข่ายเพิ่มเติมเพื่อทำให้ผู้ดูแลระบบสามารถจำกัดการใช้งานได้ในระบบเดียวเลย

บทที่ 5
สรุปผลและข้อเสนอแนะ

5.1 สรุปผลปัญหาพิเศษ

การดำเนินงานตามปัญหาพิเศษนี้มีวัตถุประสงค์คือ ต้องการตรวจจับแพ็กเก็ตที่เป็นโปรโตคอลเพียร์ทูเพียร์ คือ โปรโตคอลบิททอร์เรนท์ อีดอนกี้ และจีนูเทลล่า เนื่องจากการสื่อสารประเภทนี้ใช้แบนด์วิธมาก และสามารถแจ้งเตือนไปยังผู้ดูแลระบบเมื่อเครื่องลูกข่ายใช้การสื่อสารแบบเพียร์ทูเพียร์ โดยสามารถแสดงถึงข้อมูลการใช้งานต่างๆของเครื่องลูกข่ายที่อยู่ในระบบได้โดยจะรันโปรแกรมอยู่ตลอดเวลา

จากการศึกษาปัญหาพิเศษการพัฒนาระบบเพื่อป้องกันการสื่อสารแบบเพียร์ทูเพียร์โดยใช้โปรแกรมโอเพนซอร์ซที่รันบนระบบปฏิบัติการลีนุกซ์นั้น ระบบมีความสามารถดังนี้

1. สามารถตรวจจับแพ็กเก็ตต่างๆที่เป็นของโปรโตคอลเพียร์ทูเพียร์
2. สามารถแจ้งเตือน(alert) ให้กับผู้ควบคุมและดูแลระบบ เมื่อระบบมีการใช้การสื่อสารแบบเพียร์ทูเพียร์
3. สามารถแสดงข้อมูลการใช้งานของเครื่องลูกข่ายต่างๆที่อยู่ภายในระบบเมื่อใช้การสื่อสารแบบเพียร์ทูเพียร์กับเครื่องอื่นๆที่อยู่ทั้งภายในและภายนอกของระบบ

ระบบที่ทางคณะผู้จัดทำได้พัฒนาขึ้นนั้นมีข้อดีกว่าระบบอื่นๆคือ มีความยืดหยุ่นกว่า สามารถปรับเปลี่ยนรูปแบบการตรวจจับ(update)ให้ทันกับซิกเนเจอร์ของโปรโตคอลเพียร์ทูเพียร์ที่มีการเปลี่ยนแปลงอยู่ตลอดเวลา และเป็นระบบที่พัฒนามาจากโอเพน ซอร์ซ เสียค่าใช้จ่ายในการพัฒนาน้อย

5.2 ข้อจำกัดปัญหาพิเศษ

ปัญหาพิเศษการพัฒนาระบบเพื่อจำกัดการสื่อสารแบบเพียร์ทูเพียร์นั้นยังมีข้อจำกัดในการทำงานอยู่ คือ การทำงานของระบบ ระบบยังไม่สามารถที่จะตรวจจับโปรโตคอลเพียร์ทูเพียร์ได้ครอบคลุมทุกชนิดเนื่องจากโปรโตคอลเพียร์ทูเพียร์มีการแก้ไขตัวเองอยู่เสมอเพื่อไม่ให้ระบบสามารถตรวจจับโปรโตคอลเพียร์ทูเพียร์ของตนได้ และไม่สามารถที่จะทำลายการสื่อสารแบบเพียร์ทูเพียร์ที่เกิดขึ้นในระบบออกไปได้ ทำได้แค่เพียงตรวจจับและแจ้งเตือนไปยังผู้ควบคุมและดูแลระบบ อีกทั้งการสื่อสารแบบเพียร์ทูเพียร์มักจะมาพร้อมกับแอดแวร์ , สปายแวร์ และไวรัส ซึ่งระบบทำได้แค่เพียงแจ้งเตือนผู้ควบคุมและดูแลระบบ แต่ไม่สามารถที่จะทำลายสิ่งเล่านั้นได้

5.3 ข้อเสนอแนะและแนวทางการศึกษาต่อ

เนื่องจากมีข้อจำกัดบางประการ ฉะนั้นระบบควรจะมีความสามารถดังนี้

1. ระบบควรมีความสามารถในการตรวจจับโปรโตคอลเพียร์ทูเพียร์ได้ครอบคลุมทุกชนิด โดยการจับแพ็กเก็ตของเพียร์ทูเพียร์มาวิเคราะห์หาสิ่งที่เป็นจุดเด่นของโปรโตคอลชนิดต่างๆอยู่สม่ำเสมอ เพื่อนำมาเขียนกฎในระบบไอดีเอสและเป็นการอัพเดทกฎให้มีความทันสมัยอยู่เสมอ

2. ระบบควรมีความสามารถในการหยุดการสื่อสารแบบเพียร์ทูเพียร์ออกไปจากระบบได้โดยทำงานร่วมกับไฟร์วอลล์โดยบล็อกไอพีแอดเดรสที่มีการสื่อสารแบบเพียร์ทูเพียร์

3. ระบบควรมีความสามารถในการกำจัดพวกแอดแวร์ , สปายแวร์ และไวรัส โดยสามารถทำงานร่วมกับโปรแกรมแอนติไวรัสได้(anti-virus)

บรรณานุกรม

หนังสือ

[1] ประภาพร ช่างไม้ , "Linux redhat ฉบับผู้เริ่มต้น" , Info Press , 2547

[2] ก่อกิจ วีระอาชากุล , "ติดตั้งและปรับแต่งเซิร์ฟเวอร์ Linux สำหรับ Admin Linux โดยเฉพาะ" , Info Press , 2546

[3] Rafeeq Rehman , "Intrusion Detection with SNORT" , Prentice Hall PTR , 2546

[4] สมประสงค์ ธิตินิลนิธิ, "เรียนลัด PHP 4 : ครอบคลุมเวอร์ชั่น 4.2", โปรวิชั่น จำกัด, 2547

[5] ณัฐพล เหลืองวัฒนไพศาล พรชัย ประสิทธ์สุวรรณ และพิริยะ อรรถทวีสิน . 2547. "ระบบตรวจจับผู้บุกรุกบนเครือข่ายคอมพิวเตอร์ขนาดเล็ก." วิทยานิพนธ์วิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

บทความ

[1] ภูวดล ด่านระหาญ , "การติดตั้ง Snort ร่วมกับ ACID (+MySQL)", http://www.thaicert.nectec.or.th/paper/ids/snort2.php, เผยแพร่เมื่อ : 3 สิงหาคม 2544

[2] Subhabrata Sen, Oliver Spatscheck, Dongmei Wang, "Accurate, Scalable In-Network Identi_cation of P2P Traffic Using Application Signatures", http:// etd.lib.nsysu.edu.tw/ETD-db/ETD-search/getfile?URN=etd-0706105-135129&filename=etd-0706105-135129.pdf

เว็บไซต์ที่เกี่ยวข้อง

[1] http://www.snort.org

[2] http://www.thaicert.nectec.or.th

[3] http://acidlab.sourceforge.net/

[4] http://www.php.net

[5] http://www.w3schools.com/

[6] http://www.google.co.th

ภาคผนวก

1. คู่มือการติดตั้งโปรแกรม

1.1 การติดตั้ง Httpd (Web server)

เปิดโปรแกรม Terminal และใช้คำสั่งดังต่อไปนี้

tar –zxvf httpd-2.2.4 –C /usr/local/

cd /usr/local/httpd-2.2.4

./configure

make

make install

- เพื่อตรวจสอบการทำงานของ Httpd ควรเรียกใช้คำสั่งดังต่อไปนี้

/sbin/chkconfig httpd on

/sbin/service httpd start

/sbin/chkconfig mysqld on

/sbin/service mysqld start

- ตั้งค่ารหัสผ่านของ mysql

mysqladmin –u root password '123456'

1.2 การติดตั้ง PHP

tar –zxvf php-5.2.0.tar.gz

cd php-5.2.0

./configure –with-gd –enable-sockets –with-mysql

make

/sbin/service httpd stop

make install

vi /etc/httpd/conf/httpd.conf

หา LoadModule php5_module /usr/lib/httpd/modules/libphp5.so และเพิ่ม # ก่อนหน้าข้อความนั้น

LoadModule php5_module /usr/lib/httpd/modules/libphp5.so

กด escape และ พิมพ์ :wq

/sbin/service httpd restart

1.3 การติดตั้งโปรแกรมส่วนติดต่อผู้ใช้งาน

นำโฟลเดอร์ p2p-alert ไปไว้ใน /var/www/html/

1.4 การติดตั้ง ADODB

เปิดโปรแกรม Terminal และใช้คำสั่งดังต่อไปนี้

tar –zxvf adodb493a.tgz –C /var/www/html/

1.5 การติดตั้ง Snort

เปิดโปรแกรม Terminal และใช้คำสั่งดังต่อไปนี้

tar –zxvf snort-2.6.0.2.tar.gz –C /usr/local/

cd /usr/local/snort

./configure --with-mysql-includes=/usr/include/mysql --with-mysql-

libraries=/usr/lib/mysql

make

make install

จากนั้นให้ก๊อปปี้ข้อมูล configuration และ rules files จาก source ของ Snort ไปยัง /etc/snort เพื่อความเป็นระเบียบ

mkdir /etc/snort

cd /usr/local/src/snort

cp snort.conf /etc/snort

cp *.rules /etc.snort

cp classification.config /etc/snort

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

mkdir /var/log/snort

chmod 700 /var/log/snort

-สร้าง database structure สำหรับ Snort

ให้ล็อกอินเข้าไปยัง MySQL และสร้าง database ชื่อ snort ขึ้นมา

#mysql -uroot -p

123456

mysql>CREATE DATABASE snort;

จากนั้นให้สร้าง MySQL account ขึ้นมาเพื่อให้มีสิทธิในการจัดการกับฐานข้อมูล

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

mysql>flush privileges;

จากนั้นก็ต้องสร้าง database structure ตามที่ Snort กำหนดไว้

cd /usr/local/snort

vi contrib/create_mysql แล้วเพิ่มคำว่า USE snort; ไว้ที่บรรทัดบนสุด

mysql < ./contrib/create_mysql -uroot –p

- นำไฟล์ snort.conf ไปไว้ในโฟลเดอร์ /etc/snort/

- นำไฟล์ p2p.rules ไปไว้ในโฟลเดอร์ /etc/snort/rules

รัน Snort (daemon)

ทดลองรัน /usr/local/bin/snort -c /etc/snort/snort.conf ถ้าไม่มี error ใดๆ แสดงว่าสามารถใช้งานได้ เพียงแต่การใช้งานจริงนั้นจะรันใน daemon mode โดยจะใช้คำสั่งดังนี้

/usr/local/bin/snort -D -c /etc/snort/snort.conf

2. การเปิดโปรแกรมเริ่มต้นการทำงาน

ต้องทำการเริ่มต้นการทำงานของ Httpd ก่อนโดยใช้คำสั่ง

/sbin/service httpd restart

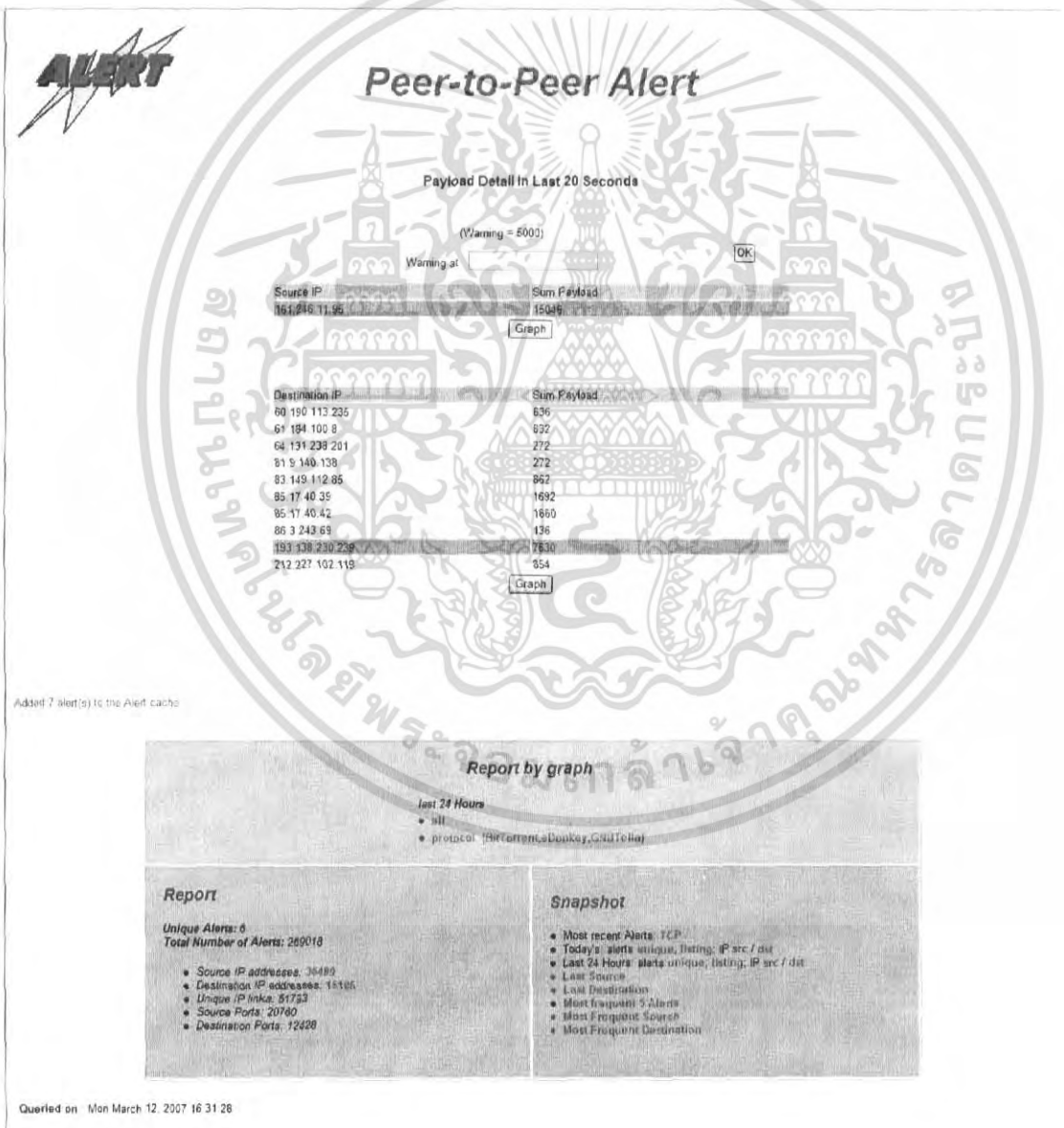
หลังจากนั้นก็เริ่มการทำงานของสนอร์ต

/usr/local/bin/snort -D -c /etc/snort/snort.conf

เปิดโปรแกรมใช้งานได้จากบราวเซอร์ โดยเรียกไปที่แอดเดรสนี้

http://localhost/p2p-alert/acid_main.php

จะได้ผลดังภาพ



รูปที่ ก 1 หน้าจอแสดงผลลัพธ์ที่ได้หลังจากเรียกโปแกรมได้สำเร็จ