การประเมินประสิทธิภาพการตรวจจับการบุกรุกในระบบเครือข่ายคอมพิวเตอร์

PERFORMANCE EVALUATION ON NETWORK INTRUSION DETECTION SYSTEM

ธนภัทร ขันธราชา

TANAPAT KHANTARACHA

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2552

การประเมินประสิทธิภาพการตรวจจับการบุกรุกในระบบเครือข่ายคอมพิวเตอร์

PERFORMANCE EVALUATION ON

NETWORK INTRUSION DETECTION SYSTEM

ธนภัทร ขานทะราชา

TANAPAT KHANTARACHA

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2552

KMITL-2009-EN-M-230-087

PERFORMANCE EVALUATION ON

NETWORK INTRUSION DETECTION SYSTEM

TANAPAT KHANTARACHA

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF ENGINEERING IN INFORMATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2009

KMITL-2009-EN-M-230-087

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การประเมินประสิทธิภาพการตรวจจับการบุกรุกในระบบเครือข่ายคอมพิวเตอร์

Thesis Title Performance Evaluation on Network Intrusion Detection System

นักศึกษา นายธนภัทร ขานทะราชา

รหัสประจำตัว 47061118

ปริญญา วิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชา วิศวกรรมสารสนเทศ

อาจารย์ที่ปรึกษาวิทยานิพนธ์ ผศ.มยุรี เลิศเวชกุล

หมายเลขวิทยานิพนธ์ KMITL-2009-EN-M-230-087

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
ผศ.ดร.พิทักษ์	ธรรมวาริน	
รศ.ดร.ปิติเขต	สู้รักษา	
รศ.ดร.ชวลิต	เบญจางคประเสริฐ	
ดร.วีระพล	โมนยะกุล	
ผศ.มยุรี	เลิศเวชกุล	

วัน / เดือน/ ปี ที่สอบ วันพฤหัสบดีที่ 28 พฤษภาคม พ.ศ. 2552 เวลา 10.30-12.30 น.

สถานที่สอบ ณ อาคาร A ชั้น 3 ห้องประชุม 2

คณะวิศวกรรมศาสตร์ รับรองแล้ว

(รองศาสตราจารย์ ดร.กอบชัย เดชหาญ)

คณบดี คณะวิศวกรรมศาสตร์

วันที่ 28 พฤษภาคม พ.ศ. 2552

หัวข้อวิทยานิพนธ์	การประเมินประสิทธิภาพการตรวจจับการบุกรุก ในระบบเครือข่ายคอมพิวเตอร์
นักศึกษา	นายธนภัทร ขานทะราชา
รหัสนักศึกษา	47061118
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมสารสนเทศ
พ.ศ.	2552
อาจารย์ที่ปรึกษาวิทยานิพนธ์	ผศ. มยุรี เลิศเวชกุล

บทคัดย่อ

ระบบตรวจจับการบุกรุกเครือข่ายคอมพิวเตอร์ (NIDS) เป็นเครื่องมือสำคัญในการวัดความปลอดภัยของเครือข่ายในปัจจุบัน แต่เนื่องจากการที่มีความพยายามค้นหาวิธีการใหม่ ๆ เพื่อเจาะระบบอยู่เสมอ ดังนั้นจึงมีการเพิ่มจำนวนของกฎและสัญลักษณ์สำหรับการตรวจจับการบุกรุกอยู่ตลอดเวลาทำให้ภาระในการตรวจสอบกฎและสัญลักษณ์เพิ่มสูงขึ้น อีกทั้งระบบเครือข่ายคอมพิวเตอร์ได้รับการปรับปรุงให้มีความเร็วสูงขึ้นมากอาจจะส่งผลต่อประสิทธิภาพของระบบตรวจจับการบุกรุกให้ลดลงได้ งานวิจัยนี้ต้องการประเมินประสิทธิภาพของ NIDS โดยการใช้โปรแกรม Snort เวอร์ชัน 2.8.2.2 เป็นตัวแทนเพื่อทดลองศึกษาผลกระทบของจำนวนกฎและสัญลักษณ์ที่มีต่อประสิทธิภาพของการตรวจจับการบุกรุก โดยใช้โปรแกรม TCPReplay-3.4.0 ในการจำลองทราฟฟิกที่มีความเร็วแตกต่างกัน ผลการทดลองได้แสดงว่าประสิทธิภาพของ NIDS จะเริ่มลดลงเมื่อความเร็วในการส่งข้อมูลเพิ่มสูงขึ้นจนถึงจุดอิ่มตัวของ NIDS และจากจุดนั้นเป็นต้นไปจะพบว่า NIDS ที่กำหนดใช้กฎและสัญลักษณ์ด้วยจำนวนมากที่สุดจะมีประสิทธิภาพลดลงมากกว่าเมื่อเปรียบเทียบกับการทดลองที่ใช้ชุดกฎและสัญลักษณ์น้อยกว่า ซึ่งผลการวิจัยนี้ชี้ปัญหาที่เกิดจากความเร็วในการส่งข้อมูลและแนวทางในการกำหนดใช้กฎและสัญลักษณ์อย่างเหมาะสม

Thesis Title	Performance Evaluation on Network Intrusion Detection System
Student	Mr. Tanapat Khantaracha
Student ID	47061118
Degree	Master of Engineering
Program	Information Engineering
Year	2009
Thesis Advisor	Asst. Prof. Mayuree Lertwetchakul

ABSTRACT

At present, Network Intrusion Detection System (NIDS) is an essential tool to measure network security system. Since the numbers of Rules and intrusion Signature have been increases everyday, this may effect to NIDS performance especially for a high speed Networking environment. This research do experiments as to a popular open source NIDS, Snort version 2.8.2.2. Experiments were done in order to find the consequence of number of rules and signatures to the NIDS performance. By using TCPReplay-3.4.0 to generate normal traffic together with attacking traffic patterns at various speeds. And we tried to limit the number of Snort rules and signatures to figure out how the number of rule effect to Snort performance. The experimental results have shown that the performance of Snort will be dropping when the network traffic rate is increased to its saturate point. At the saturate point performance of Snort, And at the higher traffic rate, Performance of Snort with the greatest number of rules and signatures setting was reduced significantly compared to the others. The results have made us to realize the problem of NIDS performance that may be caused by high speed traffic rate. And it is necessary to control and limit the set of rules and signatures to be the optimal set.

กิตติกรรมประกาศ

วิทยานิพนธ์เล่มนี้สำเร็จได้ด้วยความกรุณาจากอาจารย์ที่ปรึกษา ผศ. มยุรี เลิศเวชกุล ที่ให้ความช่วยเหลือให้คำชี้แนะช่วยแก้ปัญหาตลอดจนให้ความรู้และประสบการณ์ที่ดีแก่ข้าพเจ้า

ขอขอบพระคุณกรรมการสอบหัวข้อและโครงร่างวิทยานิพนธ์ที่ได้กรุณาให้คำแนะนำตลอดจนข้อชี้แนะ จนในที่สุดทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลงได้

ขอขอบคุณอาจารย์ทุกท่านที่กรุณาให้การอบรมสั่งสอนและให้ความรู้เสมอมา

สุดท้ายต้องขอขอบคุณเพื่อนร่วมรุ่น ภาควิชาวิศวกรรมสารสนเทศ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้ให้ความช่วยเหลือแนะนำคำปรึกษาต่างๆ ในการทำวิจัยสำเร็จลุล่วงด้วยดีและทุกท่านที่ไม่ได้กล่าวถึงในที่นี้ ที่ให้ความช่วยเหลือ และกำลังใจในการทำวิทยานิพนธ์ ฉบับนี้จนสำเร็จ

สำหรับคุณงามความดีอันใดที่เกิดจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบให้กับบิดามารดา ซึ่งเป็นที่รักและเคารพยิ่ง ตลอดจนครูอาจารย์ที่เคารพทุกท่านที่ได้ให้วิชาความรู้และถ่ายทอดประสบการณ์ที่ดีของข้าพเจ้า

ธนภัทร ขานทะราชา

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

สารบัญรูป

สารบัญรูป(ต่อ)

หน้า

ส่วนงานวิจัยนี้ได้เสนอแนวทางการประเมินประสิทธิภาพของระบบตรวจจับการบุกรุกที่แตกต่างกันออกไป โดยมุ่งเน้นในการหาค่าประสิทธิภาพของโปรแกรมตรวจจับการบุกรุกระบบเครือข่ายคอมพิวเตอร์โดยทดสอบกับโปรแกรม Snort เวอร์ชัน 2.8.2.2 เพื่อหาผลกระทบเนื่องจากการเพิ่มขึ้นของจำนวนกฎ (Rules) และสัญลักษณ์ (Signatures) ที่อาจจะส่งผลต่อประสิทธิภาพของการตรวจจับการบุกรุกในสถานการณ์ ที่ปริมาณและอัตราความเร็วในการส่งข้อมูลผ่านระบบเครือข่ายมีระดับแตกต่างกัน

1.2 วัตถุประสงค์ของวิทยานิพนธ์

โดยงานวิจัยนี้ได้เน้นการนำเสนอค่าประสิทธิภาพของการตรวจจับการบุกรุก ที่มีจำนวนของกฎ (Rule) และสัญลักษณ์ (Signature) แตกต่างกัน ในภาวะที่มีความเร็วในการส่งข้อมูลผ่านระบบเครือข่ายเพิ่มสูงขึ้น เพื่อแสดงให้เห็นปัญหาที่เกิดขึ้นดังนี้

1.2.1 เพื่อทราบถึงความสามารถของการตรวจจับการบุกรุกในภาวะที่มีความเร็วในการส่งข้อมูลผ่านระบบเครือข่ายเพิ่มสูงขึ้น

1.2.2 เพื่อทราบถึงความสามารถของการตรวจจับการบุกรุกเมื่อจำนวนของกฎ (Rule) และสัญลักษณ์ (Signature) เพิ่มมากขึ้น

1.2.3 เพื่อตระหนักถึงการกำหนดชุดของกฎที่ใช้ในการตรวจสอบการบุกรุกที่เหมาะสม เพื่อให้การตรวจจับการบุกรุกอย่างมีประสิทธิภาพ

1.3 สมมติฐานของการศึกษา

เนื่องจากผู้บุกรุกจะค้นหาวิธีใหม่ ๆ เพื่อเจาะระบบเครือข่ายคอมพิวเตอร์อยู่เสมอ จึงทำให้จำนวนกฎของโปรแกรม Snort ที่ถูกใช้งานมีแนวโน้มเพิ่มมากขึ้นเรื่อย ๆ และจากสมมติฐานว่าถ้าหากมีการเพิ่มจำนวนของกฎ (Rule) และสัญลักษณ์ (Signature) สำหรับการตรวจจับการบุกรุกจะทำให้การทำงานของโปรแกรม Snort เกิดปัญหาประสิทธิภาพในการตรวจจับการบุกรุกได้ลดลงเมื่อความเร็วและปริมาณของการส่งข้อมูลในระบบเครือข่ายเพิ่มขึ้น จุดประสงค์ของการวิจัยนี้ก็เพื่อทดลองให้เห็นว่าสมมติฐานข้างต้นมีแนวโน้มถูกต้องหรือไม่ และเสนอแนวทางในการกำหนดใช้กฎและสัญลักษณ์อย่างเหมาะสมเพื่อเพิ่มประสิทธิภาพการตรวจจับการบุกรุก

1.4 วิธีการที่นำเสนอ

การหาค่าประสิทธิภาพการตรวจจับการบุกในระบบเครือข่ายคอมพิวเตอร์ทำการทดสอบโดยจัดกลุ่มของกฎและสัญลักษณ์สำหรับตรวจจับการบุกรุกออกเป็น 3 ชุด โดยในแต่ละรอบของการทดลองจะเพิ่มอัตราความเร็วในการส่งแพ็คเก็ตข้อมูลสื่อสารทั่วไป (Background traffic) ด้วย

การใช้โปรแกรม TCPReplay-3.4.0 แล้วทำการตรวจวัดจำนวนครั้งที่โปรแกรม Snort สามารถตรวจจับการบุกรุกได้ เพื่อเปรียบเทียบกับการทดลองแต่ละรอบที่กำหนดในโปรแกรม Snort โดยที่กฎและสัญลักษณ์ชุดต่าง ๆ มีข้อมูลที่แตกต่างกัน 3 ชุดคือ กลุ่มที่หนึ่งมีจำนวน 9337 สัญลักษณ์ กลุ่มที่สองมีจำนวน 4471 สัญลักษณ์และกลุ่มที่สามมีจำนวน 2326 สัญลักษณ์ โดยเครื่องผู้บุกรุก (Intrusion PC) จะทำการเพิ่มความเร็วของการส่งข้อมูลจากข้อมูลชุดเดิมโดยใช้โปรแกรม TCPReplay-3.4.0 [9] เพื่อจำลองความเร็วในการสื่อสารข้อมูลเป็น 0.127 Mbps, 100 Mbps, 200 Mbps, 300 Mbps, 400 Mbps, 500 Mbps, 600 Mbps, 700 Mbps, 800 Mbps, 900 Mbps และ 1 Gbps ตามลำดับ อีกการทดลองหนึ่งคือการทดสอบปรับแต่งกฎให้เหมาะสมตามความเสี่ยงและสภาวะแวดล้อม โดยดักจับแพ็คเก็ตข้อมูลจากการใช้งานจริงในบริเวณเครือข่ายซึ่งติดตั้งสำหรับเซิฟเวอร์ (Server Farm) แล้วใช้แบบจำลองการหาค่าประสิทธิภาพการตรวจจับการบุกรุกมาช่วยเพื่อเปรียบเทียบจำนวนการแจ้งเตือนระหว่างการใช้ตัวตรวจจับการบุกรุกซึ่งยังไม่ได้ปรับแต่งกฎให้เหมาะสมและใช้ตัวตรวจจับการบุกรุกที่ได้รับการปรับแต่งกฎให้เหมาะสมตามสภาพแวดล้อม

1.5 ขอบเขตของวิทยานิพนธ์

1.5.1 ศึกษาโครงสร้างการทำงานของโปรแกรม Snort

1.5.2 ศึกษาประสิทธิภาพการตรวจจับการบุกรุกเมื่อเพิ่มความเร็วในการสื่อสารข้อมูล

1.5.3 ศึกษาประสิทธิภาพการตรวจจับการบุกรุกเมื่อเพิ่มหรือลดจำนวนสัญลักษณ์

1.5.4 เสนอแนะการใช้กฎและสัญลักษณ์ของโปรแกรมตรวจจับการบุกรุกอย่างมีประสิทธิภาพ

1.5.5 จำลองการหาประสิทธิภาพการตรวจจับการบุกรุกโดยเพิ่มความเร็วในการสื่อสารข้อมูลในแต่ละชุดของจำนวนสัญลักษณ์

1.6 ขั้นตอนการทำวิทยานิพนธ์

1.6.1 กำหนดจุดประสงค์ หัวข้อ และขอบเขตการทำวิทยานิพนธ์

1.6.2 ศึกษาทฤษฎีที่เกี่ยวข้อง หลักการพื้นฐานและแนวคิดของการตรวจจับการบุกรุกเครือข่ายคอมพิวเตอร์ที่จะนำมาใช้ในการทำวิทยานิพนธ์

1.6.3 ศึกษาการทำงานของโปรแกรม Snort และการกำหนดชุดของกฎที่ใช้ในการตรวจจับการบุกรุก

1.6.4 สร้างแบบจำลองการหาประสิทธิภาพการตรวจจับการบุกรุกโดยทำการเพิ่มความเร็วในการสื่อสารข้อมูลสำหรับแต่ละชุดของกฎและสัญลักษณ์ที่แตกต่างกัน

1.6.5 ทำการทดลอง บันทึก และวัดผล

1.6.6 จัดทำเอกสารประกอบวิทยานิพนธ์

1.7 โครงสร้างของวิทยานิพนธ์

วิทยานิพนธ์นี้มีหัวข้อการนำเสนอโดยสรุปดังนี้ บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง โดยกล่าวถึงการตรวจจับการบุกรุก บทที่ 3 กล่าวถึงการทำงานของโปรแกรม Snort และการกำหนดชุดของกฎที่ใช้ในการตรวจสอบการบุกรุก บทที่ 4 กล่าวถึงรูปแบบการจำลองเพื่อหาประสิทธิภาพการตรวจจับการบุกรุกและทดสอบปรับแต่งกฎให้เหมาะสม บทที่ 5 แสดงผลการทดลองประเมินประสิทธิภาพของโปรแกรมตรวจจับการบุกรุก และทำการทดสอบปรับแต่งกฎให้เหมาะสมกับสภาพแวดล้อม บทที่ 6 เป็นการสรุปเนื้อหาของวิทยานิพนธ์และข้อเสนอแนะการตรวจจับการบุกรุกอย่างมีประสิทธิภาพ.

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

ปัจจุบันการค้าบนอินเทอร์เน็ตมีเพิ่มสูงขึ้นมากเพราะมีจุดเด่นในเรื่องของการประหยัดค่าใช้จ่ายและเพิ่มประสิทธิภาพในการดำเนินธุรกิจ โดยลดความสำคัญขององค์ประกอบของธุรกิจที่มองเห็นจับต้องได้ เช่นอาคารที่ทำการ ห้องจัดแสดงสินค้า คลังสินค้า พนักงานขายและพนักงานให้บริการต้อนรับลูกค้า เป็นต้น ดังนั้นข้อจำกัดเรื่องระยะทางและเวลาทำการที่แตกต่างกัน จึงไม่เป็นอุปสรรคต่อการทำธุรกิจอีกต่อไป แต่สิ่งที่ตามมาคือผู้ไม่ประสงค์ดีที่อาจจะสร้างความเสียหายให้กับระบบข้อมูลขององค์กร ดังนั้นระบบรักษาความปลอดภัยจึงต้องมีความสามารถและพร้อมรับมือกับการบุกรุกที่อาจเกิดขึ้น โดยเครื่องมือรักษาความปลอดภัยในระบบคอมพิวเตอร์เช่น ไฟร์วอลล์ และโปรแกรมตรวจจับไวรัส รวมทั้งโปรแกรมตรวจจับการบุกรุกจะต้องทำงานอย่างมีประสิทธิภาพ เพื่อลดความเสียหายที่จะเกิดกับข้อมูลหรือการให้บริการทางธุรกิจ

2.1 การโจมตีระบบเครือข่ายคอมพิวเตอร์

การรายงานการตรวจพบการโจมตีระบบเครือข่ายหรือระบบคอมพิวเตอร์นั้นควรมีการลำดับความสำคัญหรือความรุนแรงของปัญหาที่เกิดขึ้นประกอบด้วย IDS รายงานเหตุการณ์ที่มีความรุนแรงสูงผู้ดูแลระบบจะได้สามารถตอบสนองกับเหตุการณ์นั้นทันที เพื่อป้องกันหรือลดความเสียหาย บางครั้ง IDS อาจไม่สามารถแยกแยะได้ว่าการโจมตีครั้งนั้นเป็นการโจมตีจริง ๆ หรือเป็นแค่การสแกนหาช่องโหว่ของระบบ เนื่องจากเหตุการณ์ทั้งสองประเภทจะมีสัญลักษณ์ของการโจมตีเหมือนกันดังนั้นผู้ดูแลระบบอาจต้องวิเคราะห์เองเพิ่มเติม แต่ปกติแล้วการสแกนหาช่องโหว่ถ้า IDS ตรวจพบก็จะรายงานการโจมตีในลักษณะถี่ ๆ ในขณะที่อาจจะเกิดเพียงแค่ครั้งเดียวกับเท่านั้น [12]

2.1.1 ประเภทของการโจมตี (Attack Types) การโจมตีทางเครือข่ายส่วนใหญ่จะเป็นการใช้รูปแบบเฉพาะ เพื่อทำให้ระบบการรักษาความปลอดภัยของระบบใช้การไม่ได้ ยกตัวอย่างเช่น การโจมตีบางประเภทอาจเป็นการทำให้ผู้บุกรุกสามารถอ่านไฟล์บางไฟล์ได้ แต่ในการโจมตีครั้งนี้จะไม่แก้ไขไฟล์หรือส่วนอื่น ๆ ของระบบการโจมตีอีกประเภทหนึ่งอาจเป็นการทำให้ผู้บุกรุกสามารถปิดระบบได้แต่จะไม่สามารถแก้ไขไฟล์ได้ถึงแม้การโจมตีจะมีหลากหลายรูปแบบก็ตามแต่ผลลัพธ์ก็คือเป็นการทำลายคุณสมบัติ 4 ประการของการรักษาความปลอดภัยซึ่งก็คือ

- ความลับ (Confidentiality) การโจมตีที่เป็นการทำลายความลับของข้อมูลคือการที่ผู้บุกรุกสามารถเข้าถึงข้อมูลโดยที่ไม่ได้รับอนุญาตจากเจ้าของข้อมูล ไม่ว่าจะเป็นการตั้งใจหรือไม่ก็ตาม

- ความน่าเชื่อถือ (Integrity) การโจมตีที่ถือว่าเป็นการทำลายความคงสภาพของข้อมูล หมายถึง การที่ผู้บุกรุกสามารถเปลี่ยนแปลงแก้ไขระบบ หรือข้อมูลที่อยู่ในระบบหรือแก้ไขข้อมูลที่อยู่ระหว่างการถ่ายโอนผ่านเครือข่าย

- ความพร้อมใช้งาน (Availability) การโจมตีที่เป็นการทำลายความพร้อมใช้งาน หมายถึงการโจมตีที่ทำให้ผู้ใช้งานไม่สามารถเข้าถึงและใช้งานระบบหรือข้อมูลได้

- การควบคุมการเข้าถึง (Access Control) การโจมตีที่ทำให้ผู้ที่ไม่ได้รับอนุญาตหรือผู้บุกรุกสามารถใช้ระบบได้ซึ่งเป็นผลทำให้ความลับ ความน่าเชื่อถือและความพร้อมใช้งานถูกละเมิดด้วยเช่นกัน

2.1.2 การโจมตีที่มักจะถูกตรวจจับ โดยส่วนใหญ่ IDS สามารถตรวจจับการโจมตีได้ 3 ประเภท คือ การสแกนระบบ (System Scanning) การปฏิเสธการให้บริการ (Deny of Service) และการเจาะเข้าระบบ (System Penetration) การโจมตีเหล่านี้อาจเริ่มจากภายในหรือภายนอกผ่านทางเครือข่าย ผู้ที่ดูแลระบบตรวจจับการบุกรุกต้องเข้าใจข้อแตกต่างระหว่างการโจมตีทั้ง 3 ประเภทนี้เนื่องจากการโจมตีแต่ละประเภทจะต้องใช้การตอบโต้ที่แตกต่างกัน

- การโจมตีแบบสแกนระบบ (Scanning Attack) หมายถึง การทดสอบระบบว่าใช้งานอะไรได้บ้างซึ่งทำได้โดยการส่งแพ็คเก็ตประเภทต่าง ๆ ไปยังระบบ การโจมตีแบบนี้จะคล้ายกับการสแกนเพื่อตรวจหาจุดอ่อนหรือช่องโหว่ของระบบ (Vulnerability Scan) ต่างกันที่ความตั้งใจในการสแกนเท่านั้น โดยข้อมูลที่ได้จากแพ็คเก็ตที่ตอบกลับมาจากการสแกนนั้นจะใช้เป็นข้อมูลสำหรับเรียนรู้คุณสมบัติเฉพาะของระบบและจุดอ่อนหรือช่องโหว่ ดังนั้น การสแกนระบบจึงเป็นเครื่องมือสำหรับผู้บุกรุกที่ใช้สำหรับการเรียนรู้ระบบนั้น ๆ ก่อนที่จะลงมือจริง ๆ การสแกนระบบหรือเครือข่ายไม่ใช่การเจาะเข้าระบบหรือการทำให้ระบบใช้งานไม่ได้ การโจมตีแบบสแกนอาจให้ข้อมูลต่อไปนี้แก่ผู้ดูแลระบบ

- ประเภทของทราฟฟิกที่อนุญาตให้ผ่านไฟร์วอลล์ได้
- โทโปโลยีของเครือข่ายที่จะโจมตี
- โฮสต์ที่เปิดใช้งานอยู่
- ประเภทและเวอร์ชันของระบบปฏิบัติการที่โฮสต์ในเครือข่ายใช้
- ซอฟต์แวร์ที่รันบนแต่ละเซิร์ฟเวอร์
- เวอร์ชันของซอฟต์แวร์เหล่านั้น

การสแกนเพื่อค้นหาจุดอ่อนของระบบ (Vulnerability Scan) เป็นอีกประเภทหนึ่งของการสแกนเพื่อค้นหาจุดอ่อนเฉพาะของโฮสต์นั้น ๆ ดังนั้น ผู้บุกรุกอาจใช้เครื่องมือนี้เพื่อให้รู้ว่ามี

โฮสต์ใดที่คาดว่าจะมีจุดอ่อนที่สามารถใช้เครื่องมือเฉพาะในการเจาะระบบนั้น ๆ ได้ด้วยข้อมูลเหล่านี้ผู้บุกรุกสามารถเรียนรู้ระบบเป้าหมายในเครือข่าย รวมถึงว่าจะต้องใช้การโจมตีแบบใดให้เหมาะกับจุดอ่อนของระบบนั้น ๆ

เนื่องจากการสแกนระบบหรือเครือข่ายไม่ได้เป็นการโจมตีจริง ดังนั้น บางครั้งการสแกนก็อาจไม่ใช่เรื่องที่ถูกพิจารณาว่าผิดกฎหมาย เพราะผู้ที่สแกนอาจมีความตั้งใจแค่เพื่อให้ทราบว่าระบบหรือเครือข่ายมีบริการอะไรที่เขาสามารถใช้งานได้บ้าง อย่างไรก็ตามการสแกนเครือข่ายก็ถ้าเป็นขั้นตอนแรกของความพยายามที่จะโจมตีเครือข่าย ดังนั้น IDS ก็จะรายงานเหตุการณ์นี้ เพราะมีความเป็นไปได้สูงว่าหลังจากการสแกนระบบหรือเครือข่ายแล้ว อาจมีการโจมตีตามมาก็ได้ผู้ดูแลระบบก็จะได้ระวังมากขึ้น

การสแกนระบบหรือเครือข่ายนั้นเกิดขึ้นเป็นประจำบนอินเทอร์เน็ต ซึ่งบางครั้งก็เป็นการกระทำที่ถูกกฎหมาย เช่น เว็บเสิร์ชเอ็นจิน (Web Search Engine) อย่างกูเกิล (Google) เป็นต้น ก็จะมีระบบสแกนอินเทอร์เน็ตโดยอัตโนมัติ เพื่อค้นหาเว็บใหม่ที่เกิดขึ้นเพื่ออัพเดตฐานข้อมูลของตัวเอง และก็ถือว่าไม่ใช่สิ่งที่กฎหมาย เนื่องจากจุดประสงค์ไม่ใช่เพื่อที่จะทำลายหรือโจมตีระบบนั้น ๆ แต่เครื่องมือที่ใช้สแกนเพื่อค้นหาจุดอ่อนของระบบนั้นก็เป็นเครื่องมือเดียวกันกับที่เอาไว้ใช้หาบริการ ดังนั้น IDS ที่ดีส่วนใหญ่ควรจะสามารถแยกแยะได้ว่าการสแกนใดคือการสแกนเพื่อประสงค์ร้าย อย่างไรก็ตามการสแกนก็เป็นเรื่องธรรมดาที่เกิดขึ้นเป็นประจำไม่ว่าจะด้วยจุดประสงค์ใดก็ตาม ทันทีที่เราเชื่อมต่อระบบเข้ากับอินเทอร์เน็ตระบบก็จะถูกสแกนแน่นอน อาจจะมากหรืออาจจะน้อยเท่านั้นเอง

- การโจมตีแบบปฏิเสธการให้บริการ (Denial of Service : DOS) เป็นการพยายามที่จะทำให้ระบบที่เป็นเป้าหมายทำงานช้าลงหรือถึงกับให้บริการไม่ได้เลย บนอินเทอร์เน็ตนั้น DOS ก็เป็นสิ่งที่เกิดขึ้นเป็นประจำเช่นกัน การใช้ DOS อย่างนี้อาจไม่สร้างความเสียหายมากนัก แต่สามารถใช้เป็นเครื่องมือสำหรับการโจมตีองค์กรใหญ่ ๆ ได้เช่นกันโดยการทำให้ลูกค้าไม่สามารถใช้บริการ เช่น การสั่งซื้อสินค้าได้ เป็นต้น DOS แบ่งออกได้เป็น 2 ประเภทคือ การโจมตีช่องโหว่ของระบบเซิฟเวอร์ (Flaw Exploitation DOS) และการฟลัดดิง (Flooding) โดยผู้ดูแลระบบ IDS จำเป็นที่จะต้องเข้าใจข้อแตกต่างระหว่างการ DOS ทั้ง 2 ประเภทนี้เพื่อจะได้ปฏิบัติได้อย่างถูกต้อง

- **การโจมตีช่องโหว่ (Flaw Exploitation DOS Attacks)** การโจมตี DOS แบบนี้คือ การโจมตีช่องโหว่ของระบบเพื่อให้เกิดข้อผิดพลาด หรือทำให้รีซอร์สของระบบถูกใช้งานจนหมด ยกตัวอย่างเช่น การโจมตีแบบปิงออฟเดท (Ping of Death) ซึ่งเป็นการปิงที่ส่งแพ็คเก็ตขนาดใหญ่มาก ไปยังระบบที่ใช้ระบบปฏิบัติการวินโดวส์บางเวอร์ชั่น ซึ่งทำให้ระบบที่ถูกโจมตีไม่สามารถจัดการกับแพ็คเก็ตที่ผิดปกตินี้ได้ ทำให้ระบบล่มในที่สุดผลของการโจมตีคือ ทำให้รีซอร์สของระบบถูกใช้งานหมด ไปซึ่งรีซอร์สของระบบในที่นี้อาจรวมถึง ซีพียู เมมโมรี ฮาร์ดดิสก์ บัฟเฟอร์ หรือ

แบนด์วิธของเครือข่าย ส่วนใหญ่การป้องกันการโจมตีแบบนี้ก็ทำได้โดยการปิดช่องโหว่หรือการอัพเกรดซอฟต์แวร์

- **การโจมตีแบบฟลัดดิง (Flooding Dos Attacks)** การโจมตีแบบฟลัดดิงนี้หมายถึงการโจมตีโดยการส่งข้อมูลไปยังระบบเครือข่ายหรือระบบคอมพิวเตอร์เกินกว่าจะรับไหวในกรณีที่ผู้บุกรุกไม่สามารถส่งข้อมูลจนเกินกว่า ระดับที่ระบบปลายทางรับได้นั้นแต่ผู้บุกรุกยังอาจจะใช้แบนด์วิธของเครือข่ายนั้นจนหมดไป ทำให้ผู้ใช้อื่นไม่สามารถเข้ามาใช้งานระบบได้ การโจมตีแบบนี้ไม่ใช่การโจมตีช่องโหว่ของระบบ ดังนั้นการปิดช่องโหว่หรือการอัพเกรดซอฟต์แวร์ก็จะไม่ช่วยป้องกันการโจมตีได้ซึ่งนี่ก็เป็นเหตุผลหลักที่ทำให้หลาย ๆ องค์กรมีความกังวลอย่างมาก

- การโจมตีแบบแยกกระจาย (Distributed DOS: DDOS) เป็นซับเซตของ DOS ซึ่งหมายถึงคอมพิวเตอร์ที่ใช้โจมตีนั้นใช้คอมพิวเตอร์หลาย ๆ เครื่องในการโจมตีเป้าหมายเดียว โดยคอมพิวเตอร์ที่ใช้โจมตีนี้จะถูกควบคุมจากศูนย์กลาง โดยมีคอมพิวเตอร์ของผู้บุกรุกเป็นเครื่องควบคุมโดยปกติผู้บุกรุกไม่สามารถใช้เครื่องเดียวในการโจมตีระบบใหญ่ ๆ ได้แต่ผู้บุกรุกสามารถใช้คอมพิวเตอร์หลาย ๆ เครื่องเพื่อโจมตีระบบให้ล่มได้

- การโจมตีแบบเจาะเข้าระบบ (Penetration Attacks) จะเกี่ยวกับการเข้ามาในระบบโดยที่ไม่ได้รับอนุญาตและเปลี่ยนแปลงสิทธิ์รีซอร์สและข้อมูลที่อยู่ในระบบการโจมตีแบบนี้ เป็นการโจมตีโดยการหลีกเลี่ยงหรือฝ่าฝืนระบบควบคุมการเข้าถึง และเป็นการทำลายความน่าเชื่อถือของระบบ ซึ่งจะต่างจาก DOS เนื่องจาก DOS ไม่ได้ฝ่าฝืนระบบการควบคุมหรือแก้ไขข้อมูลใด ๆ แต่เป็นการทำให้ระบบนั้นไม่พร้อมใช้งานเท่านั้น การโจมตีแบบเจาะเข้าระบบนั้นผู้บุกรุกสามารถเข้าถึงและควบคุมระบบได้โดยการเจาะเข้าทางช่องโหว่ของซอฟต์แวร์นั้น ๆ การโจมตีแบบนี้มีเทคนิคที่หลากหลายโดยแต่ละวิธีก็จะมีผลตามมาที่แตกต่างกัน แต่เราสามารถจำแนกออกเป็นประเภท ต่าง ๆ ได้ดังนี้

- **User to Root:** ยูสเซอร์ของระบบสามารถเลื่อนสิทธิ์ให้ตัวเองเป็นผู้ดูแลระบบซึ่งหมายถึง Root สำหรับยูนิกซ์ และ Administrator สำหรับระบบวินโดวส์
- **Remote to User:** ผู้บุกรุกจากเครือข่ายสามารถเลื่อนสิทธิ์ตัวเองให้เป็นผู้ใช้ของโฮสต์เป้าหมาย
- **Remote to Root:** ผู้บุกรุกจากเครือข่ายสามารถเลื่อนสิทธิ์ตัวเองเป็นผู้ดูแลระบบซึ่งสามารถควบคุมระบบได้ทั้งหมด
- **Remote Disk Read:** ผู้บุกรุกจากเครือข่ายสามารถทำให้ตัวเองมีสิทธิ์ในการอ่านไฟล์ข้อมูลในระบบโดยที่ไม่ต้องได้รับอนุญาตจากเจ้าของไฟล์
- **Remote Disk Write:** ผู้บุกรุกจากเครือข่ายสามารทำให้ตัวเองมีสิทธิ์ในการเขียนหรือแก้ไขไฟล์ข้อมูลที่อยู่ในระบบโดยที่ไม่ต้องได้รับอนุญาตจากเจ้าของไฟล์

2.1.3 แหล่งกำเนิดของการโจมตี จำแนกออกได้เป็น 2 ประเภทดังนี้

- **Authorized User Attacks:** การโจมตีประเภทนี้จะเกิดจากบุคคลภายใน ซึ่งเป็นผู้ใช้ของระบบที่ได้รับอนุญาตให้เข้าใช้ระบบได้ ซึ่งผู้ใช้พยายามที่จะเปลี่ยนสิทธิ์ในการเข้าใช้ระบบของตัวเองให้มากขึ้น

- **Public User Attacks:** การโจมตีจากภายนอกหรือผู้ใช้ทั่วไปจะเริ่มจากการที่ผู้บุกรุกไม่มีสิทธิพิเศษใด ๆ ที่จะเข้าใช้ระบบ ซึ่งการโจมตีนั้นจะเริ่มจากเครื่องรีโมทโดยใช้ช่องทางที่ระบบเปิดไว้ให้ หรือเป็นช่องโหว่ของระบบเอง

รูปแบบของการโจมตีหนึ่งที่เกิดขึ้นบ่อย คือผู้บุกรุกจะเริ่มการโจมตีจากภายนอกเพื่อให้สามารถเข้าใช้ระบบจึงค่อยอัพเกรดสิทธิ์ของตัวเองให้เป็นผู้ใช้ของระบบเพื่อจะได้สามารถควบคุมหรือมีสิทธิ์ในการใช้ระบบมากขึ้น

2.1.4 การวิเคราะห์หาแหล่งที่มาของการโจมตี ถ้ามีรายงานแจ้งเตือนว่ามีการเจาะระบบเกิดขึ้น IDS จะระบุแหล่งที่มาของผู้บุกรุก ซึ่งแหล่งที่มานี้ส่วนใหญ่จะรายงานเป็นหมายเลขไอพี (IP Address) ซึ่งข้อมูลนี้ก็ได้จากแพ็คเก็ตที่ IDS ได้รับนั่นเอง อย่างไรก็ตามผู้บุกรุกสามารถเปลี่ยนหมายเลขไอพีที่อยู่ในแพ็คเก็ตนี้ได้ ดังนั้น ข้อมูลนี้อาจไม่ใช่แหล่งที่มาของแพ็คเก็ตจริง ๆ ก็ได้หลักการของการวิเคราะห์ว่าหมายเลขไอพีที่ถูกรายงานนี้เป็นไอพีของต้นทางหรือไม่โดยการวิเคราะห์ประเภทของการโจมตีนั้น ๆ ว่าผู้บุกรุกต้องการแพ็คเก็ตตอบกลับจากเครื่องเป้าหมายหรือไม่ ถ้าการโจมตีเป็นแบบทางเดียว เช่น DOS ซึ่งผู้บุกรุกไม่จำเป็นต้องเห็นแพ็คเก็ตที่ตอบกลับผู้บุกรุกก็สามารถเปลี่ยนไอพีแหล่งที่มาของแพ็คเก็ตเป็นไอพีอะไรก็ได้ กรณีนี้ก็เทียบได้กับการที่ผู้บุกรุกส่งไปรษณีย์บัตรโดยจ่าหน้าซองที่อยู่ผู้ส่งเป็นที่อยู่อื่น ดังนั้นถ้ามีการตอบกลับก็จะถูกส่งไปยังที่อยู่ดังกล่าวในขณะที่ผู้บุกรุกนั้นก็จะไม่ได้รับการตอบกลับเลย

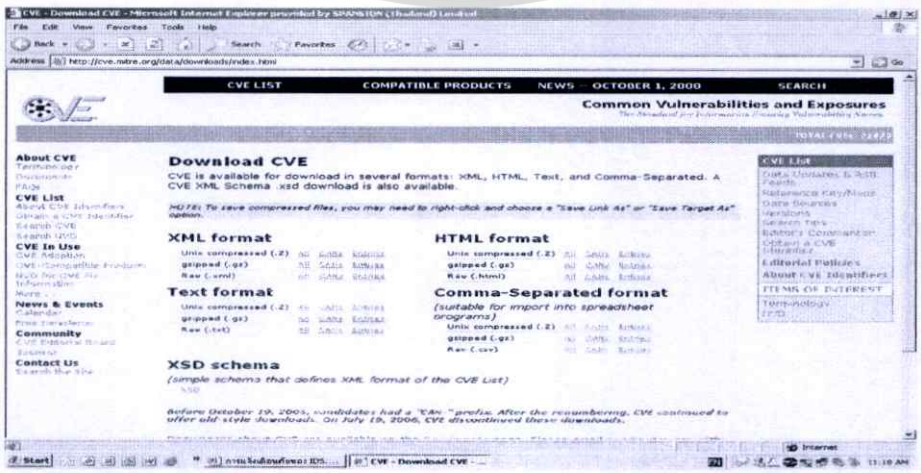
อย่างไรก็ตาม ถ้าผู้บุกรุกต้องการการตอบกลับจากเป้าหมาย เช่น การโจมตีแบบเจาะเข้าระบบในกรณีนี้ส่วนใหญ่ผู้บุกรุกจะไม่เปลี่ยนหมายเลขไอพีแหล่งที่มาของแพ็คเก็ต สรุปคร่าว ๆ ก็คือ ถ้าเป็นการโจมตีแบบ DOS นั้น หมายเลขไอพีแหล่งที่มาอาจจะไม่ใช่แหล่งที่มาจริง ๆ แต่ถ้าเป็นการโจมตีเพื่อเจาะเข้าระบบส่วนใหญ่จะระบุหมายเลขไอพีแหล่งที่มาจริง อย่างไรก็ตามข้อสรุปข้างต้นก็ไม่จริงเสมอไป โดยเฉพาะถ้าผู้บุกรุกมีความชำนาญมากตัวอย่าง เช่น ผู้บุกรุกอาจโจมตีโดยการส่งแพ็คเก็ตไปยังเป้าหมายและกำหนดไอพีแหล่งที่มาเป็นไอพีปลอมผู้บุกรุกอาจใช้การ “แท็บ” สายสัญญาณเพื่อตรวจจับเอาแพ็คเก็ตที่ตอบกลับจากแหล่งเป้าหมายได้ โดยที่ผู้บุกรุกไม่จำเป็นต้องเจาะเข้าระบบที่มีหมายเลขไอพีปลอมดังกล่าวเลย ซึ่งการปลอมแปลงหมายเลขไอพีนี้จะเรียกว่า “ไอพีสปูฟิง (IP Spoofing)”

2.1.5 ระดับความรุนแรงของการโจมตี โดย IDS หลายประเภทได้กำหนดระดับความรุนแรงของการโจมตี (Severity Level) การทำอย่างนี้ก็เพื่อเป็นข้อมูลที่ช่วยให้ผู้ดูแล IDS ในการวิเคราะห์และประเมินผลกระทบที่เกิดจากการโจมตีแต่ละครั้ง เพื่อจะได้ดำเนินการต่อได้อย่าง

เหมาะสมอย่างไรก็ตามผลกระทบและระดับความปลอดภัยนั้นอาจถือเป็นเรื่องของการคาดเดามากกว่า ดังนั้นการตอบโต้การโจมตีจึงมักขึ้นอยู่กับสภาพแวดล้อมของเครือข่าย ความสำคัญของโฮสต์และลักษณะการทำงานขององค์กรนั้น ๆ ยกตัวอย่าง เช่น ถ้าการโจมตีเป็นประเภทที่โจมตีกับโฮสต์ประเภทยูนิกซ์ในขณะที่เครือข่ายใช้วินโดวส์ทั้งหมด ดังนั้นการโจมตีนั้นก็ถือว่ามีระดับผลกระทบที่ตามมาต่ำแต่ IDS อาจจะรายงานว่ามีระดับความรุนแรงสูง เนื่องจาก IDS อาจไม่รู้ว่าเครือข่ายเป็นวินโดวส์เบสทั้งหมด การรายงานแจ้งเตือนของ IDS ก็ยังเป็นข้อมูลที่สำคัญสำหรับผู้บริหารด้านการรักษาความปลอดภัยแต่ก็ต้องพิจารณาสภาวะแวดล้อมของระบบเครือข่ายที่ IDS นั้นกำลังทำงานอยู่

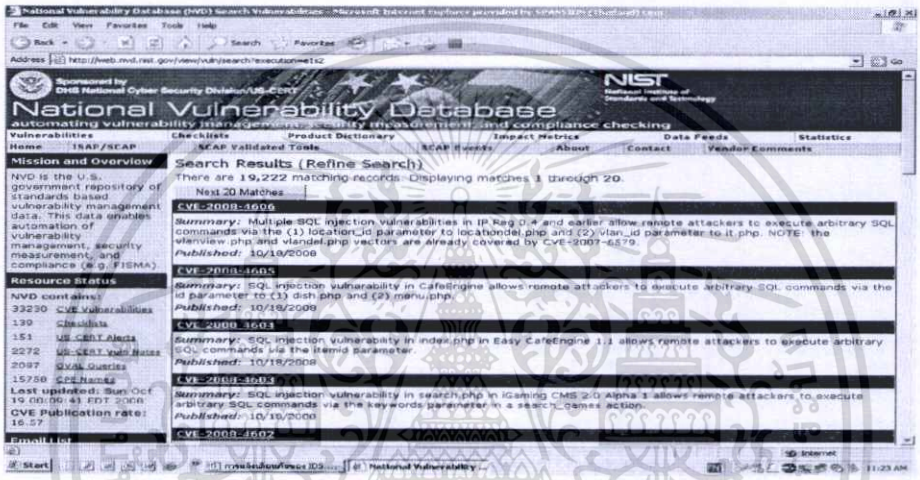
2.1.6 ช่องโหว่ของระบบคอมพิวเตอร์ (Computer Vulnerabilities) ปัจจุบันยังไม่มีมาตรฐานที่ใช้เรียกชื่อการโจมตี (Attack Naming Conventions) ประเภทต่าง ๆ ด้วยเหตุนี้จึงเป็นการยากที่จะเปรียบเทียบประสิทธิภาพของ IDS แต่ละตัว เนื่องจาก IDS แต่ละตัวจะสร้างการรายงานหรือแจ้งเตือนที่แตกต่างกันถึงแม้จะเป็นการโจมตีแบบเดียวกันก็ตาม ทำให้ยากต่อการใช้ IDS หลาย ๆ ตัวในเครือข่ายเดียวกัน เนื่องจาก IDS แต่ละตัวจะรายงานต่างกันในการโจมตีแบบเดียวกัน

โดยมีความพยายามที่จะสร้างมาตรฐานสำหรับการเรียกชื่อที่เกี่ยวกับช่องโหว่และการโจมตี มาตรฐานซึ่งที่เป็นที่นิยมมากที่สุดก็คือ CVE (Common Vulnerabilities and Exposures) ซึ่งสร้างขึ้นโดยบริษัท MITRE ฐานข้อมูลช่องโหว่ CVE ถือได้ว่าเป็นที่ยอมรับและเป็นฐานข้อมูลที่มีประโยชน์ต่อผู้ใช้งานซอฟต์แวร์ในการค้นหาช่องโหว่ของระบบ นอกจากข้อมูลเบื้องต้นของช่องโหว่แล้ว ฐานข้อมูล CVE ยังมีส่วนของลิงก์สำหรับหาข้อมูลเพิ่มเติม รวมทั้งวิธีการในการปิดช่องโหว่อีกด้วย ประโยชน์ที่เห็นได้ชัดเจนอีกสิ่งหนึ่งคือ CVE ถือเป็นฐานข้อมูลหลักสำหรับเครื่องมือทดสอบช่องโหว่ (Vulnerability Scanner) โดยรายละเอียดสามารถดูได้จาก http://cve.mitre.org



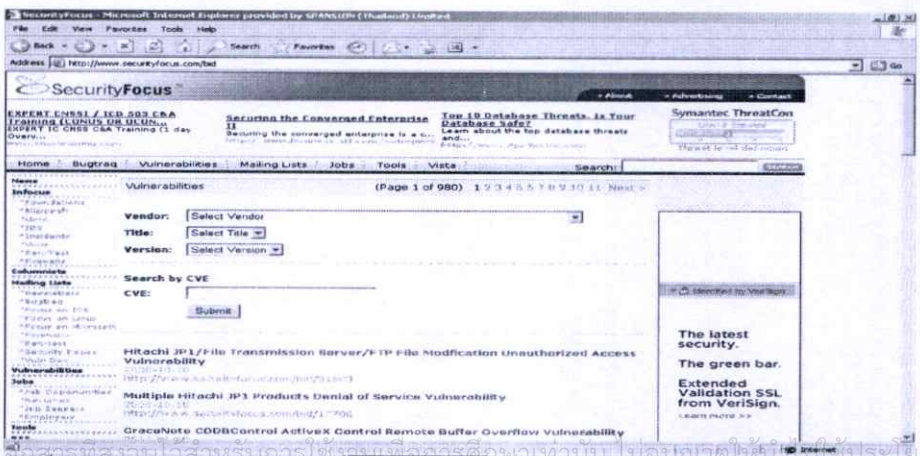
รูปที่ 2.1 เว็บไซต์ http://cve.mitre.org

ส่วนใหญ่ IDS จะรายงานแจ้งเตือนการบุกรุกโดยจะมีข้อมูลและคำอธิบายของการโจมตีนั้นประกอบ นอกจากนี้ยังอาจจะรายงายประเภทของจุดอ่อนหรือช่องโหว่ที่การโจมตีนั้นใช้ประโยชน์ ข้อมูลเหล่านี้เป็นสิ่งที่สำคัญมากหลังจากที่เกิดการโจมตี เพราะผู้ดูแลระบบสามารถวิเคราะห์และปิดช่องโหว่หรือจุดอ่อนดังกล่าวได้ สถาบันมาตรฐานและเทคโนโลยีแห่งประเทศสหรัฐฯ หรือ NIST (National Institute of Standard and Technology) ได้แนะนำให้ใช้ ICAT Metabase ซึ่งเป็นฐานข้อมูลที่เก็บข้อมูลเกี่ยวกับการโจมตีประเภทต่าง ๆ ICAT จะให้ข้อมูลเกี่ยวกับการโจมตีประเภทต่าง ๆ และรายละเอียดคำแนะนำเกี่ยวกับการแก้ไขช่องโหว่หรือจุดอ่อนของระบบ ผู้อ่านสามารถศึกษารายละเอียดเกี่ยวกับ ICAT ได้ที่เว็บไซต์ http://nvd.nist.gov



รูปที่ 2.2 เว็บไซต์ http://nvd.nist.gov

นอกจากช่องโหว่หรือจุดอ่อนของระบบคอมพิวเตอร์ที่มีการรายงานและเก็บไว้ในฐานข้อมูล ICAT แล้ว Bugtraq (http://www.securityfocus.com/bid) ก็เป็นอีกรูปแบบหนึ่งในการรายงานช่องโหว่ ซึ่งเป็นเมลลิงลิสต์ (mailing list) ที่สร้างขึ้นสำหรับการโต้ตอบกันเกี่ยวกับช่องโหว่ของระบบคอมพิวเตอร์ วิธีใช้ประโยชน์จากช่องโหว่และวิธีในการแก้ไขโดยส่วนใหญ่ถ้ามีช่องโหว่ใหม่ ๆ ที่ค้นพบก็จะมีการสนทนากันเป็นที่แรก ๆ



รูปที่ 2.3 เว็บไซต์ http://www.securityfocus.com/bid

ในหัวข้อต่อไปนี้จะเป็นการอธิบายถึงประเภทหลัก ๆ ของช่องโหว่หรือจุดอ่อนของคอมพิวเตอร์ โดยมีความพยายามในการจัดประเภทของช่องโหว่ที่แตกต่างกันไป อย่างไรก็ตามได้มีการพัฒนามาตรฐานสำหรับการเรียกชื่อช่องโหว่ต่าง ๆ โดยด้านล่างจะเป็นรายการของช่องโหว่ที่มักจะพบเห็นหรือเกิดขึ้นอยู่เป็นประจำและได้จัดกลุ่มดังต่อไปนี้

- **Input Validation Error** ช่องโหว่ของซอฟต์แวร์ประเภทนี้เกิดจากการที่ระบบรับอินพุตที่ไม่ได้ตรวจสอบความถูกต้องของข้อมูลก่อน ซึ่งเป็นผลทำให้การประมวลผลที่เกิดจากข้อมูลผิดพลาดได้ การโจมตีเกิดขึ้นโดยการส่งข้อมูลเฉพาะไปยังระบบ ช่องโหว่ประเภทความไม่ถูกต้องของอินพุตนี้ แบ่งออกได้เป็น 2 ประเภทคือ บัฟเฟอร์โอเวอร์โฟลว์ (Buffer Overflow) และความผิดพลาดอันเนื่องจากขอบเขตข้อมูล (Boundary Condition Error)

  - **Buffer Overflow** จุดอ่อนประเภทบัฟเฟอร์โอเวอร์โฟลว์ เกิดขึ้นโดยการที่ระบบได้รับอินพุตที่มีขนาดใหญ่กว่าขนาดของตัวแปรที่กำหนด แต่ระบบไม่ได้ตรวจสอบความถูกต้องก่อนทำให้บัฟเฟอร์ที่เก็บอินพุตนี้ไม่สามารถเก็บข้อมูลได้หมดส่งผลให้ข้อมูลบางส่วนไปทับแทนที่ข้อมูลหรือคำสั่งในเมมโมรีส่วนอื่น โดยมีการจัดการข้อมูลส่วนเกินนี้ด้วยเทคนิคบางอย่าง ซึ่งผู้บุกรุกสามารถรันโปรแกรมอื่นแทนโปรแกรมที่กำลังรันอยู่ได้ ตัวอย่างเช่น การโจมตีแบบฟิงเกอร์เอ็กซ์พล็อต (Fingered Exploit) ซึ่งเกิดขึ้นโดยผู้บุกรุกส่งคำสั่งฟิงเกอร์ไปยังระบบพร้อมด้วยอินพุตที่มีขนาดเกินกว่า 80 ตัวอักษร
  - **Boundary Condition Error** ข้อผิดพลาดประเภทนี้เกิดขึ้นโดย ระบบได้รับอินพุตเกินขอบเขตที่โปรแกรมกำหนดไว้ ซึ่งข้อมูลอินพุตนี้อาจมาจากผู้ใช้หรือจากการคำนวณ โดยโปรแกรมการล้นของข้อมูลอินพุตนี้ทำให้เกิดช่องโหว่ของโปรแกรม ยกตัวอย่างเช่น ระบบที่ใช้รีซอร์สจนหมด เช่น เมมโมรี ฮาร์ดดิสก์ หรือแบนด์วิธของเน็ตเวิร์ค อีกตัวอย่างหนึ่ง เช่น ตัวแปร (Variable) อาจถึงค่าสูงสุดแล้วทำให้รีเซตกลับมาเป็นค่าเริ่มต้นหรือค่าต่ำสุด อีกตัวอย่างหนึ่งคือ ตัวแปรถูกเซตเป็นศูนย์แล้วคำนวณโดยการหารด้วยศูนย์ (Division by Zero) เบาดารีคอนดิชันเออเรอร์เป็นซับเซตของอินพุตวาลิเดชันเออเรอร์

- **Access Validation Errors** เกิดขึ้นเนื่องจากกลไกควบคุมการเข้าถึงระบบ (Access Control) ทำงานผิดพลาด ความผิดพลาดนี้ไม่ได้เกิดจากข้อผิดพลาดของการคอนฟิกแต่เป็นระบบควบคุมเองที่ทำงานผิดพลาด

- **Exceptional Condition Handling Error** ข้อผิดพลาดนี้เกิดขึ้นเนื่องจาก การจัดการเกี่ยวกับข้อยกเว้นของเงื่อนไขการทำงานของโปรแกรมโดยฟังก์ชันของโปรแกรมที่จัดการข้อยกเว้นต่าง ๆ อาจทำงานผิดพลาดเอง หรือเกิดจากการจัดการผิดพลาดไม่ถูกต้องก็ได้

- **Environmental Error** ข้อผิดพลาดอาจเกิดขึ้นเนื่องจากระบบถูกติดตั้งในสภาวะแวดล้อมที่ไม่เหมาะสม ซึ่งกลายเป็นช่องโหว่ของระบบที่อาจเกิดจากเหตุที่ว่าแอพพลิเคชันอาจ

ทำงานได้ไม่ดีในระบบปฏิบัติการ หรืออาจเกิดจากการทำงานที่ขัดกันของแอพพลิเคชันที่รันบนระบบเดียวกัน จุดอ่อนหรือช่องโหว่นี้อาจตรวจสอบไม่เจอในช่วงของการพัฒนาแอพพลิเคชันแต่อาจเกิดขึ้นได้เมื่อติดตั้งและใช้งานจริง เหตุผลก็เนื่องจากสภาวะแวดล้อมของระบบที่ใช้ในการพัฒนาซอฟต์แวร์อาจไม่เหมือนกับสภาวะแวดล้อมของระบบที่ใช้งานจริง

- **Configuration Error** ข้อผิดพลาดที่เกิดจากการคอนฟิกระบบไม่ถูกต้องอาจกลายเป็นช่องโหว่ของระบบได้ ช่องโหว่ไม่ได้เกิดจากการออกแบบระบบ แต่เกิดจากผู้ดูแลระบบคอนฟิกค่าต่าง ๆ ของระบบไม่ถูกต้องหรือสมบูรณ์ ตัวอย่างเช่น ระบบปฏิบัติการบางระบบอาจคอนฟิกด้วยค่าดีฟอลต์ที่ไม่เข้มงวดมากพอหรือเพื่อสดวกต่อการใช้งาน แต่อาจกลายเป็นช่องโหว่ที่แฮค-เกอร์ใช้สำหรับการโจมตีระบบได้

- **Race Condition** เกิดขึ้นเมื่อมีการหน่วงระหว่างเวลาเมื่อระบบการรักษาความปลอดภัยของระบบเช็คเพื่อดูว่าการทำงานของโปรแกรมได้รับอนุญาตหรือไม่ กับเวลาที่โปรแกรมนั้นทำงานจริง ๆ ปัญหาจริง ๆ เกิดจากสภาวะแวดล้อมที่ระบบการรักษาความปลอดภัยของระบบทำงานต่างกับสภาวะแวดล้อมที่โปรแกรมนั้นทำงานจริง ๆ ผู้บุกรุกอาจใช้ประโยชน์จากช่วงเวลาสั้น ๆ นี้และทำให้ระบบเชื่อว่าการทำงานของโปรแกรมนั้นทำงานภายใต้ยูสเซอร์ที่มีสิทธิ์หรือได้ถูกตรวจสอบ โดยระบบการรักษาความปลอดภัยแล้วแต่ที่จริงแล้วยังไม่ได้ตรวจสอบเพราะขณะนั้นระบบการรักษาความปลอดภัยอาจยังไม่ถึงช่วงเวลาที่กำหนดให้ทำงาน

2.1.7 การสำรวจเครือข่าย ซึ่งเหตุการณ์ที่เป็นการสำรวจเครือข่ายเป็นการพยายามของผู้บุกรุกที่จะรวบรวมข้อมูลเกี่ยวกับระบบเครือข่ายก่อนที่จะโจมตีจริง ๆ เช่น

- **IP Scans:** ไอพีสแกนเป็นความพยายามของผู้บุกรุกที่ทราบเกี่ยวกับโฮสต์ต่าง ๆ ที่อยู่ในเครือข่าย ซึ่งระบบที่สแกนนั้นอาจใช้การปิง (Ping) ช่วงของหมายเลขไอพีของเครือข่ายนั้น

- **Port Scans:** หลังจากที่ผู้บุกรุกได้ข้อมูลเกี่ยวกับว่าเครือข่ายมีโฮสต์ใดอยู่บ้าง ข้อมูลต่อมาที่ผู้บุกรุกต้องการคือ บริการใดบ้างที่แต่ละโฮสต์ให้บริการอยู่ ซึ่งหมายเลขพอร์ตนั้นจะเป็นสิ่งที่บ่งบอกว่ามีแอพพลิเคชันใดบ้างที่ให้บริการอยู่

- **Trojan Scans:** การสแกนโทรจันนั้นเป็นความพยายามของผู้บุกรุกที่จะตรวจเช็คว่ามีพอร์ตของโทรจันใดบ้างที่เปิดอยู่

- **Vulnerability Scans:** การสแกนหาจุดอ่อนหรือช่องโหว่ของระบบ เป็นความพยายามที่จะใช้การโจมตีหลาย ๆ แบบกับระบบใดระบบหนึ่งเพื่อตรวจเช็คดูระบบนี้มีจุดอ่อนอย่างไร

- **File Snooping:** ไฟล์สนูปิงเป็นการตรวจเช็คว่ายูสเซอร์มีสิทธิ์อย่างไรกับไฟล์นั้น ซึ่งระบบไฟล์ส่วนใหญ่จะมีการกำหนดสิทธิ์ของผู้ใช้ในการเข้าใช้แต่ละไฟล์

2.1.8 เหตุการณ์ที่น่าสงสัย เหตุการณ์อื่น ๆ ที่ผิดปกติและไม่ได้จัดอยู่ในประเภทต่าง ๆ ที่กล่าวข้างต้นถือว่าเป็นเหตุการณ์ที่น่าสงสัยว่าอาจมีการโจมตีเครือข่ายเกิดขึ้น ซึ่งผู้ดูแลระบบต้องวิเคราะห์และสืบหาสาเหตุของเหตุการณ์ที่ว่านี้ต่อไป ตัวอย่างเช่น บางโฮสต์อาจส่งแพ็คเก็ตที่

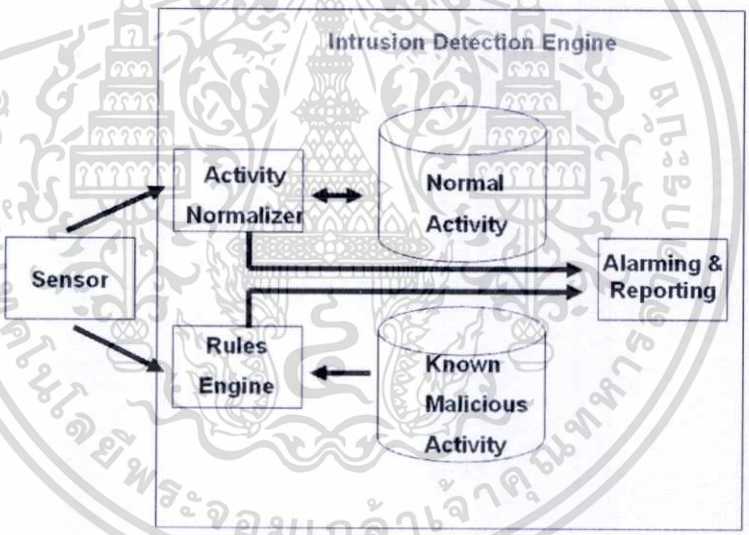
มีข้อมูลส่วนหัวผิดไปจากที่กำหนดในมาตรฐาน ซึ่งอาจเป็นการโจมตีแบบใหม่ หรือเน็ตเวิร์คการ์ดเครื่องส่งอาจเสีย หรือข้อมูลอาจเกิดผิดพลาดในระหว่างการส่งผ่านสายสัญญาณ IDS จะไม่มีข้อมูลเพียงพอที่จะบอกได้ว่าเหตุการณ์นี้เกิดขึ้นเพราะอะไร แต่จะแจ้งเตือนให้ผู้ดูแลระบบทราบเพื่อสืบค้นหาสาเหตุที่แท้จริงต่อไป

2.2 ระบบตรวจจับการบุกรุก

ระบบตรวจจับการบุกรุก (Intrusion Detection System) คือระบบที่ทำหน้าที่ติดตามดูการทำงานที่เกิดขึ้นบนระบบคอมพิวเตอร์ เพื่อหาสัญญาณบ่งชี้ว่ามีการบุกรุกระบบคอมพิวเตอร์ อาทิ เช่น การพยายามเจาะเข้าระบบ หรือ การเข้าใช้ระบบเกินขอบเขตที่ได้รับอนุญาต [11]

2.2.1 องค์ประกอบของระบบตรวจจับการบุกรุก

องค์ประกอบทั่ว ๆ ไปของระบบตรวจจับการบุกรุกจะเป็นดังรูปที่ 2.4



รูปที่ 2.4 องค์ประกอบของระบบตรวจจับการบุกรุก

- **Sensors** คือส่วนที่ทำหน้าที่รวบรวมข้อมูลต่าง ๆ เช่นแพ็คเก็ตที่ไหลอยู่บนเครือข่าย ไฟล์บันทึกกิจกรรมที่เกิดขึ้นบนระบบเครือข่ายโดยข้อมูลเหล่านี้จะถูกส่งต่อเพื่อยังระบบตรวจจับการบุกรุกเพื่อวิเคราะห์ข้อมูลต่อไป

- **Rules Engine** คือส่วนที่วิเคราะห์ว่ามีการบุกรุกเกิดขึ้นในระบบหรือไม่ โดยเปรียบเทียบกิจกรรมที่เกิดขึ้นกับสัญลักษณ์ (Signature) ที่เก็บไว้ Known Malicious Activity

- **Activity Normalizer** คือส่วนที่วิเคราะห์ว่าเกิดกิจกรรมที่ผิดปกติขึ้นในระบบหรือไม่ โดยพิจารณาจากค่าอ้างอิงของกิจกรรมปกติ (Baseline) ของข้อมูลในฐานข้อมูลของ Normal Activity กับ สถิติของกิจกรรมที่เกิดขึ้นในปัจจุบัน ตัวอย่างของค่าที่ใช้เก็บเป็นสถิติได้แก่ การใช้งานซีพียู การใช้งานสื่อบันทึกข้อมูลจำนวนแพ็คเก็ตที่วิ่งเข้าออกในระบบ

- **Normal Activity** คือฐานข้อมูลที่เก็บสถิติการใช้งานระบบจากกิจกรรมต่าง ๆ ในสภาวะปกติ

- **Known Malicious Activity** คือฐานข้อมูลที่เก็บรูปแบบของการบุกรุกหรือโจมตีระบบ

- **Alarming and Reporting** คือส่วนที่ใช้แจ้งเตือนและรายงานให้ผู้ดูแลระบบทราบว่ามีการบุกรุกระบบเกิดขึ้น

2.2.2 วิธีการตรวจจับการบุกรุก

- **วิธีเปรียบเทียบพฤติกรรมผู้ใช้กับรูปแบบการบุกรุกที่รู้จัก (Misused Intrusion Detection)** วิธีนี้จะอาศัยรูปแบบของการบุกรุกที่เคยเกิดขึ้นแล้วเป็นตัวตรวจสอบ โดยจะรวบรวมรูปแบบของการบุกรุกแล้วเก็บเป็นกฎไว้ เมื่อมีการใช้งานระบบคอมพิวเตอร์ที่เป็นพฤติกรรมการใช้งานจะถูกนำมาเปรียบเทียบกับกฎ หากตรงกับกฎระบบตรวจจับการบุกรุกจะแจ้งเตือนวิธีการนี้ผู้ดูแลระบบสามารถเปลี่ยนแปลงหรือเพิ่มกฎได้ แต่สิ่งที่จะละเลยไม่ได้คือต้องมีการปรับปรุงกฎให้ทันสมัยอยู่เสมอ

- **วิธีตรวจสอบการใช้งานทรัพยากรระบบที่ผิดปกติ (Anomaly Intrusion Detection)** วิธีนี้จะอยู่บนสมมติฐานที่ว่ากิจกรรมใด ๆ ที่เป็นการบุกรุกจะมีการใช้งานทรัพยากรของระบบอย่างผิดปกติ ดังนั้นจะมีการเก็บบันทึกข้อมูลการใช้งานระบบคอมพิวเตอร์เอาไว้สำหรับใช้เปรียบเทียบเพื่อหาว่ากิจกรรมใดเป็นกิจกรรมที่ผิดปกติ โดยการเปรียบเทียบจะใช้หลักการของสถิติเข้าช่วย

2.2.3 ประเภทของระบบการตรวจจับการบุกรุก

- **ระบบตรวจจับการบุกรุกเฉพาะเครื่อง (Host-based Intrusion Detection System)** เป็นซอฟต์แวร์ที่ติดตั้งลงบนเครื่องที่ต้องการความปลอดภัยสูงอย่างเช่น เครื่องเซิร์ฟเวอร์ซึ่งทำหน้าที่วิเคราะห์กิจกรรมต่าง ๆ ที่เกิดขึ้นบนเครื่องที่ได้ติดตั้ง HIDS เท่านั้นกิจกรรมที่มุ่งร้ายต่อระบบสามารถตรวจสอบได้จากการบันทึกข้อมูลของระบบปฏิบัติการและการบันทึกข้อมูลของโปรแกรมสำเร็จรูปบนเครื่องเซิร์ฟเวอร์ ดังนั้น HIDS จะต้องนำการบันทึกข้อมูลเหล่านั้นมาตรวจสอบโดยใช้ทรัพยากรของระบบให้น้อยที่สุด ซึ่ง HIDS ในปัจจุบันไม่มีตัวไหนที่นำการบันทึกข้อมูลทั้งหมดมาวิเคราะห์เพราะจะทำให้สิ้นเปลืองทรัพยากรและเวลามากจนเกินไปด้วยเหตุนี้การเลือกใช้งานก็ต้องพิจารณาว่าการบันทึกข้อมูลที่ HIDS เพื่อนำไปวิเคราะห์นั้นมีโอกาสที่จะเจอการบุกรุกอยู่ในเกณฑ์ที่สามารถยอมรับได้หรือไม่ โดยไม่ใช้ทรัพยากรและเวลามากจนเกินไปจากที่ได้กล่าวมาข้างต้นจะเห็นว่าการบันทึกข้อมูลต่าง ๆ เป็นปัจจัยสำคัญในการตรวจจับการบุกรุก ถ้ามีการเปลี่ยนแปลงการบันทึกข้อมูลเหล่านั้นก็จะทำให้การตรวจจับผิดพลาด ดังนั้น HIDS ที่ดีจะต้องตรวจสอบการเปลี่ยนแปลงของ การบันทึกข้อมูลด้วยตัวอย่างของการบันทึกข้อมูลที่ใช้ในการตรวจจับการบุกรุกได้แก่ Syslog บนระบบปฏิบัติการ UNIX หรือ System events บนระบบปฏิบัติการ Windows

ข้อดีของ HIDS

- ไม่ต้องจัดเตรียมฮาร์ดแวร์เพิ่มเติม สำหรับการเฝ้าระวังการบุกรุก เพราะติดตั้งลงบนเครื่องคอมพิวเตอร์ที่ใช้งานในระบบอยู่แล้ว
- สามารถตรวจสอบกิจกรรมที่ NIDS ไม่สามารถตรวจสอบได้เช่น ข้อมูลที่มีการเข้ารหัสจากต้นทางและถอดรหัสออกที่เครื่องปลายทางเท่านั้น
- ตรวจสอบการบุกรุกรูปแบบใหม่ ๆ ได้ดีกว่า NIDS
- ตอบสนองต่อการบุกรุกได้อย่างรวดเร็ว
- เกิดความผิดพลาดแบบ (False Positive) น้อยกว่า NIDS

ข้อเสียของ HIDS

- เครื่องที่ติดตั้ง HIDS จะต้องแบ่งปันทรัพยากรมาให้ HIDS แทนที่จะนำไปใช้ในงานบริการหลักอย่างเต็มที่
- รับรู้หรือตรวจสอบการโจมตีได้เพียงบางส่วนของระบบเครือข่ายคอมพิวเตอร์
- อาจถูกขัดขวางการทำงาน จากการโจมตีแบบ Denial of Service (DOS)

- ระบบตรวจจับการบุกรุกในเครือข่าย (Network-based Intrusion Detection System) ทั้งแบบที่เป็นฮาร์ดแวร์และซอฟต์แวร์ทำหน้าที่วิเคราะห์ข้อมูลบนเครือข่ายที่รับผิดชอบว่ามีการบุกรุกเกิดขึ้นหรือไม่และเนื่องจากต้องนำข้อมูลบนเครือข่ายมาวิเคราะห์ ดังนั้น IDS ประเภทนี้จะต้องมีกลไกในการดักจับข้อมูลเหมือนกับโปรแกรมดักฟังระบบเครือข่าย (Sniffer) ระบบตรวจจับการบุกรุกในเครือข่าย (NIDS) ใช้ในการตรวจสอบแพ็คเก็ตที่ผิดปกติหรือกล่าวอีกนัยหนึ่งคือ ตรวจจับการบุกรุกหรือโจมตีโดยทั่ว ๆ ไปจะมี 3 วิธีคือ

- Signature Detection คือ การตรวจหารูปแบบผิดปกติในเนื้อข้อมูลของแพ็คเก็ตในระบบเครือข่ายโดยเทียบกับฐานข้อมูลที่มีอยู่หากพบว่าตรงกันก็จะส่งสัญญาณหรือแจ้งเตือน วิธีนี้เรียกอีกอย่างหนึ่งว่า Pattern Matching ข้อดีของวิธีนี้คือ ผู้ดูแลสามารถทราบว่ากำลังถูกบุกรุกหรือโจมตีโดยรูปแบบหรือวิธีการใดอยู่ทำให้สามารถเตรียมหาวิธีป้องกันและแก้ไขได้ทันทีแต่จุดอ่อนของวิธีนี้คือระบบจะสามารถตรวจพบรูปแบบการบุกรุกที่มีอยู่ในฐานข้อมูลเท่านั้น หากเป็นรูปแบบการบุกรุกหรือโจมตีใหม่ ๆ ที่มันยังไม่รู้จักหรือที่ยังไม่มีในฐานข้อมูลแล้วระบบจะไม่สามารถตรวจพบความผิดปกติได้ ดังนั้นผู้ดูแลระบบจึงจำเป็นต้องอัพเดทฐานข้อมูลให้ทันสมัยอยู่เสมอ ในลักษณะเดียวกับการอัพเดทฐานข้อมูลของโปรแกรมป้องกันไวรัส (Antivirus) ทั่ว ๆ ไป นอกจากนี้เทคนิคของ Signature Detection ยังมีจุดอ่อนคือมักจะเกิดการแจ้งเตือนที่ผิดพลาดบ่อย ๆ หรือที่เรียกว่า False Positive ดังนั้น ในปัจจุบันจึงได้มีการปรับปรุงเทคนิคดังกล่าวขึ้นและเรียกชื่อว่า Stateful Signature Detection หรือ Stateful Pattern Matching ซึ่งได้เพิ่มการตรวจสอบ

ความต่อเนื่องของแพ็คเก็ตทั้งก่อนหน้า และหลังจากแพ็คเก็ตที่ถูกระบุว่าเป็นอันตรายต่อระบบทำให้สามารถตรวจสอบและแจ้งเตือนได้แม่นยำขึ้นทำให้โอกาสที่จะเกิด False Positive หรือการแจ้งเตือนที่ผิดพลาดจะน้อยลงด้วย

- **Behavioral Anomaly Detection** คือการตรวจหาพฤติกรรมการใช้งานที่มีลักษณะผิดแผกแตกต่างไปจากการใช้งานปกติประจำวัน ตัวอย่างเช่นในระบบเครือข่ายขององค์กรโดยปกติในวันหยุดสุดสัปดาห์มักจะไม่มีการใช้งานหรือมีก็น้อยมาก ดังนั้นหากพบว่าในวันหยุดใดมีการใช้งานระบบเครือข่ายเป็นปริมาณสูงก็ถือเป็นความผิดปกติได้ ข้อดีของวิธีนี้คือสามารถตรวจสอบความผิดปกติได้โดยไม่จำเป็นต้องอาศัยฐานข้อมูลเหมือนกรณี Signature Detection ตัวอย่างเช่นในกรณีที่ระบบเครือข่ายกำลังมี เวิร์ม (Worms) แพร่ระบาดอยู่ ซึ่งพฤติกรรมของเวิร์มโดยทั่วไปหลังจากที่มีเข้าไปสู่เครื่องเป้าหมายได้แล้วมันมักจะสแกนพอร์ตเพื่อหาเครื่องเป้าหมายอื่น ๆ หรืออาจจะทำการโจมตีเครื่องอื่น ๆ ด้วย วิธีการส่งแพ็คเก็ตที่มีลักษณะก่อกวนเครื่องเป้าหมายให้หยุดบริการหรือล่มไปและทำให้เกิดการใช้งานเครือข่ายปริมาณสูงผิดปกติ อย่างไรก็ตามผู้ดูแลระบบอาจจะไม่ทราบว่ากำลังถูกเวิร์มชนิดไหนเล่นงานอยู่แต่ก็เพียงพอให้ทราบว่าระบบเครือข่ายที่ดูแลอยู่กำลังถูกโจมตี สำหรับจุดอ่อนของวิธีนี้คือผู้ดูแลจะต้องสร้างหรือกำหนดข้อมูลที่ใช้เป็นฐานในการเปรียบเทียบซึ่งอาจทำได้โดยวิธีการเก็บสถิติการใช้งานในช่วงเวลาหนึ่ง โดยทั่วไปจะใช้เวลาประมาณ 1 สัปดาห์ถึง 1 เดือน ดังนั้นบางคนจึงเรียกวิธีนี้ว่า Statistical Anomaly Detection การเก็บสถิติการใช้งานเครือข่ายเพื่อนำมาใช้เป็นฐานในการเปรียบเทียบนั้น NIDS สามารถทำได้โดยอัตโนมัติหรืออาจถูกกำหนดโดยผู้ดูแลระบบเองก็ได้และถ้าหากข้อมูลที่ใช้เป็นฐานในการเปรียบ-เทียบไม่ดีพอก็อาจทำให้เกิดการแจ้งเตือนที่ผิดพลาดบ่อย ๆ หรือ False Positive หรือในทางกลับกันก็อาจจะตรวจไม่พบความผิดปกติทั้ง ๆ ที่มันมีความผิดปกติเกิดขึ้นจริง ๆ ซึ่งเราจะเรียกกรณีนี้ว่า False Negative ดังนั้นกว่าจะได้ข้อมูลที่ใช้เป็นฐานในการเปรียบเทียบที่ทำให้เกิดการแจ้งเตือนที่ผิดพลาดน้อยสุดก็อาจต้องผ่านการปรับเปลี่ยนหรือลองผิดลองถูกกันอยู่หลายครั้งวิธีนี้ยังมีจุดอ่อนคือไม่สามารถครอบคลุมรูปแบบการบุกรุกหรือโจมตีได้ทุกกรณี

- **Protocol Anomaly Detection** คือการตรวจแพ็คเก็ตที่ผ่านเข้าออกระบบเครือข่ายในระดับโครงสร้างของแพ็คเก็ตพร้อมทั้งลำดับขั้นตอนการสื่อสารของโปรโตคอลชนิดต่าง ๆ เช่น SMTP, POP3, IMAP, NetBIOS, SNMP, RPC, DNS, FTP หรือ HTTP ว่าเป็นไปตามข้อกำหนดมาตรฐานของโปรโตคอลนั้น ๆ หรืออธิบายง่าย ๆ คือเป็นการตรวจหาช่องโหว่ในตัวโปรโตคอลแต่ละชนิดนั่นเอง วิธีนี้จะมีข้อดีคือจะเป็นการตรวจสอบในระดับของโปรโตคอลที่ตรงไปตรงมาไม่สลับซับซ้อนมีกฎเกณฑ์ในการตรวจสอบที่แน่นอนทำให้ไม่เป็นภาระการทำงานของตัว NIDS

มากนักจึงทำงานได้อย่างรวดเร็วและนอกจากนี้ก็ไม่ต้องการอัพเดทฐานข้อมูลให้ทันสมัยอยู่เสมอ เหมือนในกรณีของ Signature Detection และยังสามารถตรวจพบรูปแบบการบุกรุกใหม่ ๆ ที่อาศัยช่องโหว่หรือการทำงานที่ผิดพลาดในตัวโปรโตคอลได้ ซึ่งข้อกำหนดมาตรฐานโปรโตคอลแต่ละประเภทถูกกำหนดไว้ในเอกสาร RFC (Request for Comment) ซึ่งมีรายละเอียดอยู่ในเว็บไซต์ www.ietf.org

ข้อดีของ NIDS

- สามารถติดตั้งบน Switching Hub แทนการติดตั้งบนเครื่องคอมพิวเตอร์
- สามารถจัดการระบบตรวจจับการบุกรุกแบบรวมศูนย์ได้ในกรณีที่มีตัวตรวจจับการบุกรุกมากกว่า 1 ตัวบนเครือข่าย
- รับรู้หรือตรวจสอบการโจมตีบนระบบเครือข่ายคอมพิวเตอร์ได้ในวงกว้างกว่า HIDS

ข้อเสียของ NIDS

- ไม่สามารถวิเคราะห์ข้อมูลที่มีการเข้ารหัส
- ความยุ่งยากในการจัดการจะแปรผันตามขนาดของระบบเครือข่ายคอมพิวเตอร์
- ไม่สามารถระบุได้ว่าการบุกรุกนั้นประสบความสำเร็จหรือล้มเหลว

2.2.4 ความผิดพลาดในระบบตรวจจับการบุกรุก มีอยู่ 2 แบบคือ false positive กับ false negative

- **False positive** เป็นความผิดพลาดที่เกิดจากการที่ระบบตรวจจับการบุกรุกแจ้งเตือนว่ามีการบุกรุกแต่แท้ที่จริงแล้วไม่มีการบุกรุกเกิดขึ้นความผิดพลาดแบบนี้ ไม่ก่อให้เกิดอันตรายเพียงแต่ผู้ดูแลระบบต้องมีความรู้เพียงพอ
- **False negative** เป็นความผิดพลาดที่อันตรายเพราะมีการบุกรุกเกิดขึ้นบนระบบคอมพิวเตอร์แต่ระบบตรวจจับการบุกรุกตรวจไม่เจอ

2.2.5 ประเภทของการรายงานแจ้งเตือนภัย การทำงานของ IDS จะรายงานเฉพาะสิ่งที่กำหนดให้เท่านั้น ซึ่งมีอยู่สองสิ่งที่ผู้ดูแลระบบจะต้องกำหนดให้กับ IDS สิ่งแรกคือ สัญลักษณ์ของการบุกรุก สิ่งที่สองคือ เหตุการณ์ที่ผู้ดูแลระบบให้ความสำคัญหรือเหตุการณ์ที่คาดว่าจะเป็นผลไปสู่การบุกรุกในภายหน้า ซึ่งเหตุการณ์ต่าง ๆ เหล่านี้เป็นทราฟฟิกที่ไม่ปกติหรืออาจเป็นบางข้อความในล็อค การคอนฟิกสัญลักษณ์ให้กับ IDS ของแต่ละองค์กรนั้นอาจจะไม่เหมือนกัน ซึ่งจะขึ้นอยู่กับว่าองค์กรนั้นให้ความสนใจกับการบุกรุกประเภทใด

เมื่อ IDS ได้ถูกคอนฟิกอย่างถูกต้องแล้ว เหตุการณ์ที่ IDS จะรายงานให้ทราบนั้นสามารถแบ่งออกได้เป็น 3 ประเภท คือ

- การสำรวจเครือข่าย
- การโจมตี
- เหตุการณ์ที่น่าสงสัยหรือผิดปกติ

2.2.6 รายงานการแจ้งเตือน โดยส่วนใหญ่ IDS จะเก็บรายละเอียดเกี่ยวกับการโจมตีหรือสิ่งที่ผิดปกติ โดยสรุปเป็นข้อความสั้น ๆ ภายในหนึ่งบรรทัด ซึ่งข้อมูลที่รายงานนนี้ส่วนใหญ่จะมีรายละเอียดดังรายการด้านล่าง

- IP ของ IDS ที่รายงาน
- วันเวลา
- ชื่อของการโจมตี ซึ่งอาจเป็นชื่อเรียกมาตรฐานหรืออาจเป็นชื่อที่กำหนดเอง
- หมายเลขไอพีของต้นทางและปลายทาง
- หมายเลขพอร์ตของต้นทางและปลายทาง
- ชื่อโปรโตคอลที่ใช้สำหรับการโจมตี

IDS ส่วนใหญ่จะมีคำอธิบายคร่าว ๆ เกี่ยวกับแต่ละประเภทของการโจมตี ซึ่งข้อมูลนี้มีความสำคัญ เนื่องจากจะเป็นข้อมูลสำหรับผู้ดูแลระบบที่จะใช้วิเคราะห์ และวัดความเสียหายที่เกิดจากการโจมตีดังกล่าวซึ่งคำอธิบายเกี่ยวกับการโจมตีนั้นส่วนใหญ่จะประกอบด้วยข้อมูลดังนี้

- คำอธิบายของการโจมตี
- ระดับความรุนแรงของการโจมตี
- ประเภทของความเสียหายซึ่งเป็นผลที่เกิดจากการโจมตี
- ประเภทของจุดอ่อนหรือช่องโหว่ที่ผู้บุกรุกใช้โจมตี
- รายชื่อของซอฟต์แวร์และเวอร์ชันที่มีจุดอ่อนหรือช่องโหว่
- ข้อมูลเกี่ยวกับแพตช์ (Patch) ที่ใช้สำหรับแก้ไขช่องโหว่ดังกล่าว
- แหล่งข้อมูลที่ให้คำปรึกษาเกี่ยวกับการโจมตีหรือช่องโหว่ที่ถูกเจาะเข้ามา

ภายในองค์กรควรที่จะมีแผนการและขั้นตอนการปฏิบัติเมื่อเกิดเหตุการณ์ขึ้น เช่น เมื่อมีไวรัสแพร่กระจายในเครือข่ายผู้ใช้ภายในใช้งานเครือข่ายในทางที่ผิดหรือเป็นการโจมตีอย่างน้อยควรกำหนดหน้าที่และความรับผิดชอบให้กับแต่ละส่วนและปฏิบัติการตามแผนงานเมื่อเกิดเหตุการณ์นั้นขึ้นและควรจัดการฝึกอบรมบุคลากรเกี่ยวกับหน้าที่และความรับผิดชอบเมื่อเกิดเหตุการณ์นอกเหนือจากนี้ดังนั้นภายในองค์กรควรทดสอบหรือฝึกการปฏิบัติเสมือนเกิดเหตุการณ์จริงเพื่อให้พนักงานที่ปฏิบัติการเกี่ยวกับ IDS อย่างคุ้นเคยและโครงสร้างของระบบการรักษาความปลอดภัยและควรแก้ไขและปรับปรุงขั้นตอนการปฏิบัติให้เหมาะสมกับสภาพแวดล้อมขององค์กรด้วย

2.2.7 ตำแหน่งการติดตั้งระบบตรวจจับการบุกรุก NIDS โดยทั่วไปจะวางไว้หลังอุปกรณ์ ไฟร์วอลล์ ตัวที่เชื่อมต่อกับระบบเครือข่ายภายนอกหรืออินเทอร์เน็ต (Perimeter Firewall) ส่วนการวางไว้หน้า ไฟร์วอลล์ ก็ขึ้นอยู่กับวัตถุประสงค์ของการใช้งาน หรือวาง NIDS ไว้ทั้ง 2 ตำแหน่งก็ทำได้ แต่การวาง NIDS ไว้หน้า ไฟร์วอลล์ นั้นหากไม่ปรับจูนให้ดีจะทำให้เกิดการแจ้งเตือนอยู่บ่อย ๆ เนื่องจากมันจะต้องรับภาระหนักในการตรวจสอบทราฟฟิกของแพ็คเก็ตที่ไหลมาจากอินเทอร์เน็ตจำนวนมหาศาลอยู่ตลอดเวลา ซึ่งส่วนใหญ่แล้วการแจ้งเตือนที่เกิดขึ้นมักจะไม่ได้เกิดจากการตรวจพบการบุกรุกที่แท้จริงซึ่งเราเรียกกรณีนี้ว่า False Positive ซึ่งวิธีการลดอัตราการเกิด False Positive ก็คือการปรับตั้งให้ตัว NIDS มีความไวน้อยลงแต่ผลกระทบที่ตามมาคือบางครั้งหากมีการบุกรุกเครือข่ายเกิดขึ้นจริงมันก็อาจจะตรวจไม่พบก็ได้ซึ่งในกรณีนี้จะเรียกว่า False Negative

ดังนั้นการใช้งานโดยทั่วไปจะวาง NIDS ไว้หลัง ไฟร์วอลล์ จะเหมาะสมกว่า เพราะจะทำให้มันไม่ต้องรับภาระหนักจากทราฟฟิกแพ็คเก็ตปริมาณมหาศาลจากอินเทอร์เน็ต เนื่องจากตัวไฟร์วอลล์ จะทำหน้าที่คัดกรองแพ็คเก็ตที่ไม่เกี่ยวข้องออกไปให้เหลือเฉพาะแพ็คเก็ตที่เกี่ยวข้องหรือที่ได้รับอนุญาตให้ผ่านไฟร์วอลล์เข้ามาในระบบ LAN แล้วเท่านั้น ซึ่งเป็นการลดภาระการทำงานของตัว NIDS ลงอย่างมากและส่งผลให้การส่งสัญญาณแจ้งเตือนมีความแม่นยำมากขึ้น กล่าวคืออัตราการเกิด False Positive จะมีน้อยมาก [13]

ค่าของระดับที่ยอมรับได้ของการแจ้งเตือนที่ไม่ได้เกิดการโจมตีขึ้นจริงขึ้นอยู่กับทราฟ-ฟิกในระบบเครือข่ายและการออกแบบ IDS โดยทั่วไประบบตรวจจับระบบการบุกรุกระบบเครือข่ายที่ไม่ได้ทำการปรับแต่งกฎและสัญลักษณ์นั้นจะมีเพียง 10% เท่านั้นที่การแจ้งเตือนเกี่ยวข้องกับการบุกรุกจริง ๆ แต่ส่วนที่เหลือ 90 % นั้นจะเป็นการแจ้งเตือนที่ไม่ได้เกิดการโจมตีขึ้นจริง ซึ่งอาจเป็นที่ถกเถียงกันในการพิจารณาและเปอร์เซ็นต์ที่ยอมรับได้ของการแจ้งเตือนที่ไม่ได้เกิดการโจมตีขึ้นจริง (False alarms) และความเหมาะสมในการปรับแต่งกฎและสัญลักษณ์ ซึ่งปกติแล้วค่าเฉลี่ยของการแจ้งเตือนจริง ๆ น่าจะอยู่ที่ประมาณ 60 % ขึ้นไปจึงจะยอมรับได้และอาจจะถึง 90 % ทั้งนี้ขึ้นอยู่กับระดับความเข้มงวดของการปรับแต่งกฎและสัญลักษณ์รวมทั้งชนิดของทราฟฟิกบนระบบเครือข่ายก็ส่งผลต่อการตรวจจับการบุกรุกด้วยเช่นกัน [15]

ตำแหน่งที่ติดตั้งระบบตรวจจับการบุกรุกบนเครือข่ายคอมพิวเตอร์นั้นขึ้นอยู่กับวัตถุประสงค์ของการใช้งาน ซึ่งสามารถแบ่งเครือข่ายออกเป็น 5 ส่วน [16] ดังแสดงในรูปที่ 2.5

- **ตำแหน่งที่ 1 External Zone** ด้านหน้าไฟร์วอลล์หรือ Internet Zone เป็นส่วนที่มีความเสี่ยงสูงที่สุดเพราะเป็นจุดแรกที่จะถูกโจมตีจากการบุกรุก โดยที่ระบบตรวจจับการบุกรุกจะถูก

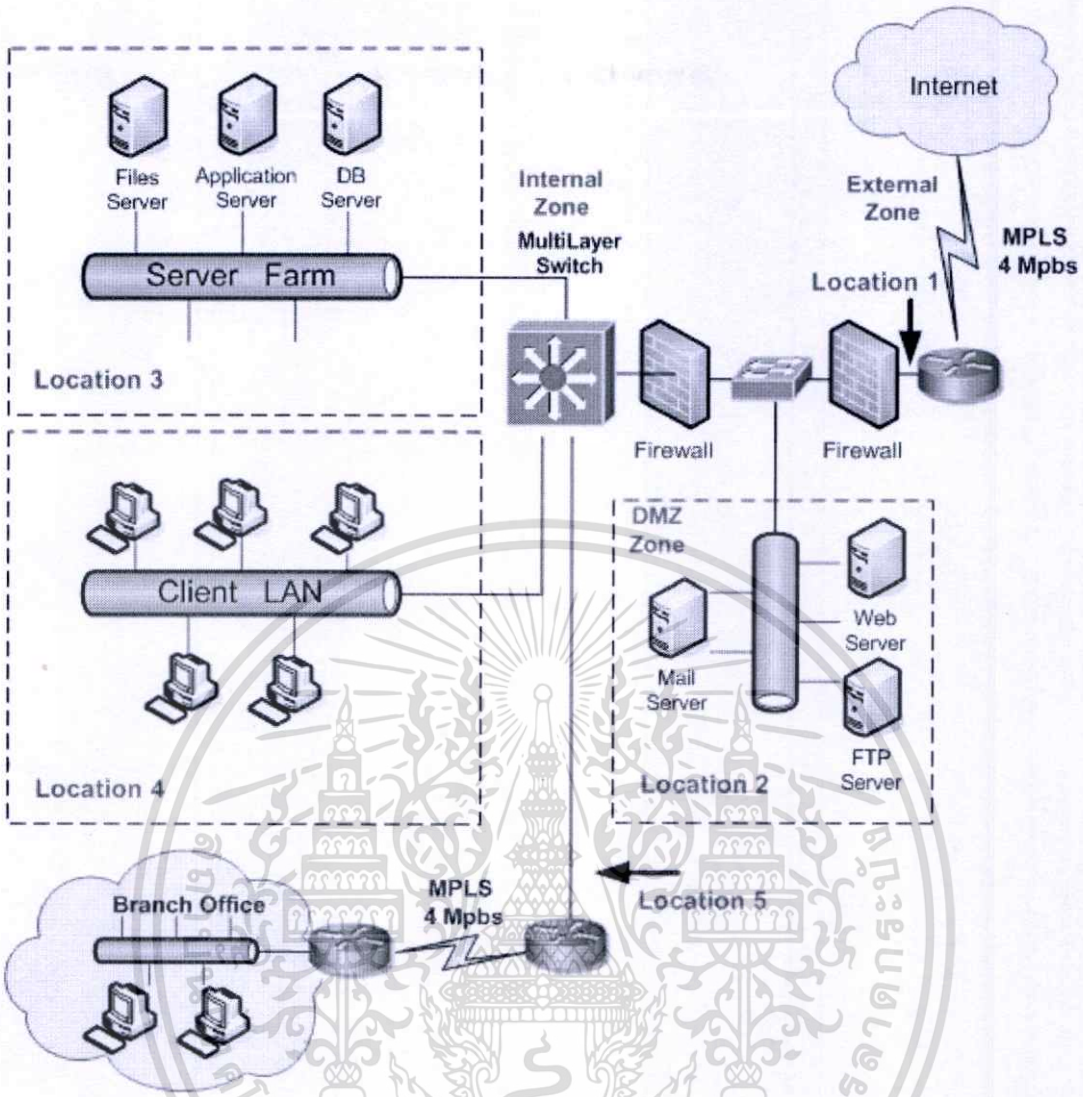
ปรับแต่งให้มีความไวต่อการตรวจจับการโจมตีมากที่สุดเพราะว่าระบบเครือข่ายคอมพิวเตอร์ในส่วนนี้จะมีข้อมูลไหลผ่านมากที่สุด และเป็นส่วนที่มีการแจ้งเตือนมากที่สุด

- **ตำแหน่งที่ 2 Demilitarized Zone (DMZ)** เป็นส่วนที่เครือข่ายภายนอกและเครือข่ายภายในเข้ามาใช้งานได้ตามกฎที่ไฟร์วอลล์ได้ตั้งไว้ซึ่งเป็นส่วนที่ระบบตรวจจับการบุกรุกจะถูกปรับแต่งให้มีความไวต่อการตรวจจับการโจมตีน้อยกว่าส่วนด้านหน้าไฟร์วอลล์ เพราะว่าเครือข่ายในส่วนนี้จะทำงานอยู่หลังไฟร์วอลล์ที่ได้ถูกปรับแต่งให้เหมาะสมแล้วระดับหนึ่ง เนื่องจากผู้ที่จะผ่านเข้ามาได้ต้องเป็นผู้ที่ได้รับสิทธิเท่านั้น

- **ตำแหน่งที่ 3 Server Farms** โดยปกติแล้วเซิร์ฟเวอร์จะถูกติดตั้งไว้กับเน็ตเวิร์กของตัวเองแต่ปัญหาที่เกิดขึ้นคือ NIDS ไม่สามารถรองรับขนาดของทราฟฟิกได้ สำหรับเซิร์ฟเวอร์ที่มีความสำคัญมาก ๆ ท่านอาจจะติดตั้ง NIDS แยกสำหรับเซิร์ฟเวอร์นั้นและเนื่องจาก NIDS ควรจะใช้กับแอพพิเคชั่นเซิร์ฟเวอร์มากกว่า

- **ตำแหน่งที่ 4 Internal Zone** เป็นส่วนของระบบเครือข่ายภายใน ซึ่งต้องคำนึงถึงการโจมตีที่อาจเกิดจากผู้บุกรุกเครือข่ายภายใน รวมทั้งผู้ใช้ที่ใช้สิทธิในทางที่ผิดหรือการลักลอบใช้สิทธิของผู้ใช้คนอื่น ๆ ที่มีสิทธิเหนือกว่า เป็นต้น แต่เนื่องจากว่าโครงข่ายข้อมูลภายในมีขนาดของปริมาณข้อมูล (Traffic) ที่ค่อนข้างสูง ซึ่งผู้ผลิตบางรายจึงใช้ NIDS ร่วมกับสวิทซ์ (Switch) บนระบบเครือข่ายด้วย

- **ตำแหน่งที่ 5 WAN Backbone** มีไว้เพื่อให้ผู้ใช้ที่อยู่อีกเครือข่ายที่อยู่ห่างไกลกันเข้ามาใช้งานเซิร์ฟเวอร์หรือทรัพยากรร่วมกันภายในหน่วยงาน ซึ่งเป็นอีกตำแหน่งที่ NIDS มีประสิทธิภาพการทำงานสูง เนื่องจากบ่อยครั้งที่มีการบุกรุกจากเครือข่ายที่อยู่ห่างไกลกัน



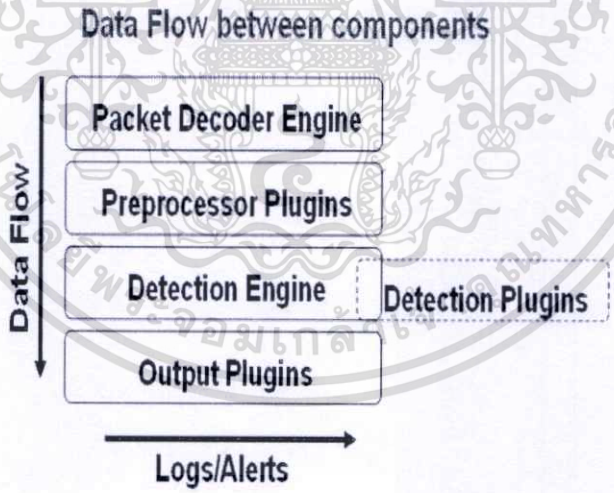
รูปที่ 2.5 ตำแหน่งที่ติดตั้งระบบตรวจจับการบุกรุกบนเครือข่ายคอมพิวเตอร์

บทที่ 3

หลักการทำงานของ SNORT

3.1 โครงสร้างของ Snort

Snort เป็นโปรแกรมเครื่องมือที่ใช้ตรวจจับการบุกรุกทางระบบเครือข่าย (Network Intrusion Detection System : NIDS) ที่นำเสนอโดย Martin Roesch การทำงานของ Snort จะใช้ไลบรารีพื้นฐานชื่อ libpcab ซึ่งใช้กันโดยทั่วไปในบรรดาโปรแกรมดักฟังระบบเครือข่าย (Network sniffer) และโปรแกรมวิเคราะห์ระบบเครือข่าย (Network analyzer) ทั้งหลายสำหรับโปรแกรม Snort นั้นมีความสามารถในการวิเคราะห์แพ็กเก็ตสื่อสาร (protocol analysis) ตรวจสอบเนื้อความภายในแพ็กเก็ต (content searching/matching) ตรวจจับการบุกรุก (cracking in) โจมตี (attack) และการสืบหาช่องโหว่ของระบบ (probe) เช่น buffer overflow, stealth port scan, CGI attack, SMB probe, OS Fingerprint และอื่น ๆ [11] โดยการทำงานของโปรแกรม Snort [5], [6] ดังรูปที่ 3.1

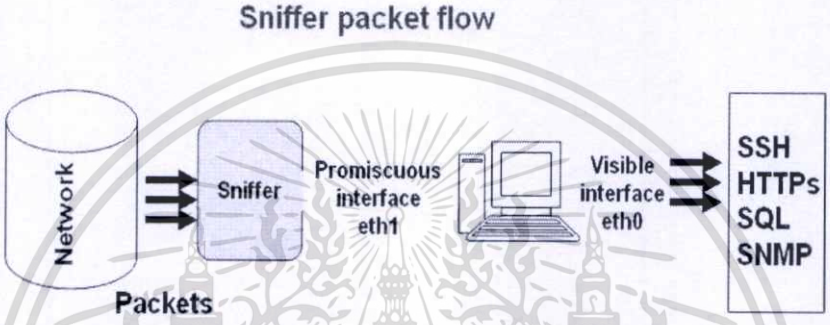


รูปที่ 3.1 การไหลของข้อมูลภายในโปรแกรม Snort

Snort ประกอบไปด้วย 4 กลไกหลักคือ Packet Decoder Engine, Preprocessor Plugins, Detection Engine และ Output Plugins ซึ่งมีหน้าที่ดังต่อไปนี้

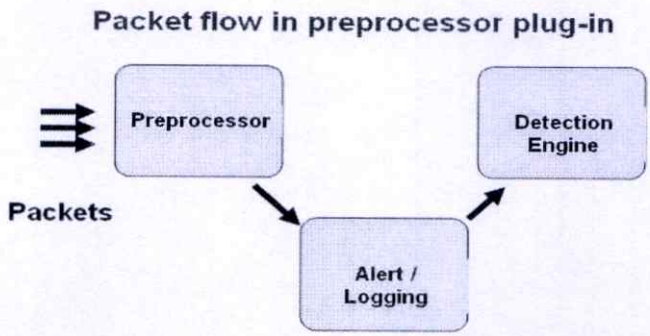
3.1.1 Packet Decoder Engine ทำหน้าถอดรหัสข้อมูลจากแพ็กเก็ตที่ดักฟังได้จากระบบเครือข่ายและนำไปบรรจุไว้ในโครงสร้างข้อมูลที่โปรแกรม Snort เข้าใจ จากนั้นก็ทำการตีความว่า

เป็นเฟรมของโปรโตคอลในชั้นเชื่อมต่อข้อมูล (Data Link Layer) ชนิดใด โดยมีองค์ประกอบที่สำคัญ 2 ส่วนคือ ไลบารี่ libcap และฟังก์ชัน ProcessPacket ส่วนของไลบารี่ libcap จะเปลี่ยนโหมดการทำงานของการ์ดสื่อสาร (Network interface card) ให้เป็นโหมดรับแพ็กเก็ต (Promiscuous) เพื่อรับแพ็กเก็ตทั้งหมดที่รับส่งกันอยู่บนระบบเครือข่าย ทำสำเนาข้อมูลเอาไว้และใช้ฟังก์ชั่น ProcessPacket เพื่อแยกแพ็กเก็ตข้อมูลตามชนิดของโปรโตคอลสื่อสารในชั้นเชื่อมต่อข้อมูล เช่น Ethernet, Wi-fi หรือ Token-ring เป็นต้น หลังจากนั้นจะส่งข้อมูลเหล่านั้นไปให้ Preprocessor



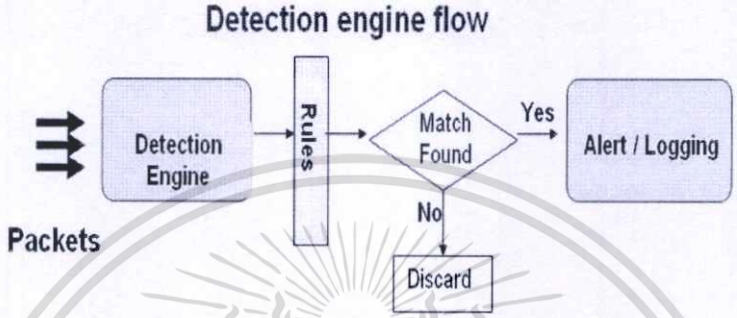
รูปที่ 3.2 การทำงานของ Packet Decoder Engine

3.1.2 Preprocessor Plugins ทำหน้าที่แปลงข้อมูลให้อยู่ในรูปแบบที่กลไก Detection engine สามารถทำความเข้าใจได้ ซึ่งนอกจากจะตรวจสอบแพ็คเก็ตโดยทั่วไปแล้วยังสามารถแก้ไขข้อความภายใน (contents) ของแพ็กเก็ตเพื่อ normalize ข้อมูลเป็นต้น และยังสามารถทำสัญลักษณ์แพ็กเก็ต (Tag) ไม่ให้ส่งไปยัง Detection engine อีกทั้งสร้างการแจ้งเตือนและบันทึกข้อมูลของแพ็กเก็ต โดยการสั่งให้ preprocessor ทำงานนั้นต้องเปิดใช้ preprocessor directive โดนในไฟล์กฎ ของแต่ละ preprocessor มีคำสำคัญและ argument list ของตัวเองที่ไม่ซ้ำกัน และเมื่อ preprocessor จบการทำงาน แพ็กเก็ตจะถูกส่งต่อไปยัง Detection Engine เพื่อเปรียบเทียบ packet flags ด้วยเงื่อนไขตามกฎ (rule) ที่กำหนดไว้



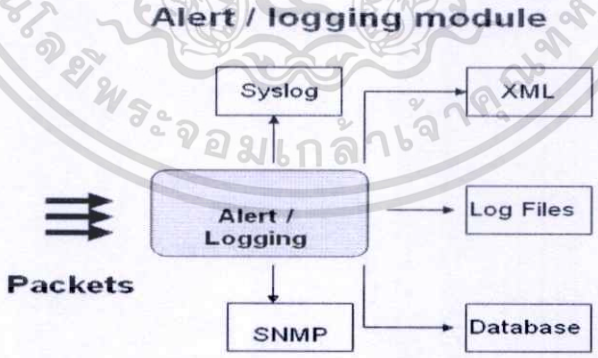
รูปที่ 3.3 การทำงานของ Preprocessor plug-in

3.1.3 Detection Engine เป็นส่วนสำคัญที่สุดของโปรแกรม Snort ใช้ตรวจสอบว่าแพ็กเก็ตข้อมูลที่เข้ามาในระบบเครือข่ายนั้นเป็นการบุกรุกหรือไม่ การทำงานของ Detection Engine คือ การนำเอาข้อมูลที่ได้รับจาก Preprocessor มาเปรียบเทียบกับกฎ ถ้าตรงกับกฎก็จะทำงานตาม Rule action ที่ได้กำหนดเอาไว้ แต่ถ้าข้อมูลไม่ตรงกับกฎก็จะตรวจสอบข้อมูลถัดไป



รูปที่ 3.4 การทำงานของ Detection engine

3.1.4 Output Plugins ทำหน้าที่แสดงผลของการตรวจสอบการบุกรุกในกรณีที่ข้อมูลที่เข้ามาตรงกับกฎที่ตั้งไว้ใน Detection Engine ให้ผู้ดูแลระบบทราบ ซึ่งผู้ใช้สามารถกำหนดให้เกิดการแจ้งเตือนได้หลายรูปแบบ เช่น ส่งไปยังเก็บยังไฟล์ต่าง ๆ เช่น Syslog, Log files เก็บลงในฐานข้อมูล แจ้งเตือนผ่านทาง SNMP หรือแสดงผลเป็นเอกสาร XML เป็นต้น



รูปที่ 3.5 การทำงานของ Alert / logging module

3.2 โหมดการทำงานของ Snort

โปรแกรม Snort [13] จะมีโหมดการทำงานอยู่ 4 โหมดคือ

3.2.1 Packet Sniffer Mode เป็นโหมดที่ใช้ดักจับข้อมูลที่วิ่งบนเครือข่ายคอมพิวเตอร์ แล้วแสดงผลบนจอมอนิเตอร์ให้ผู้ดูแลทราบ

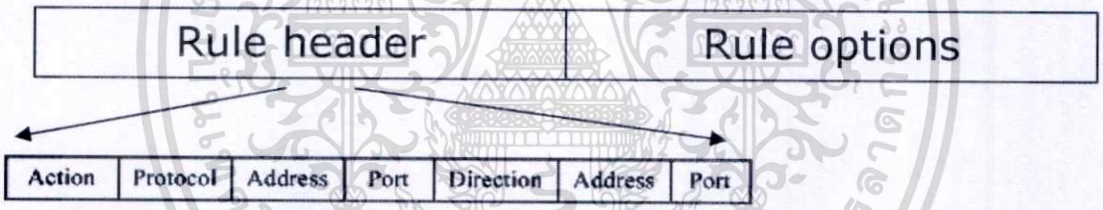
3.2.2 Packet Logger Mode เป็นโหมดที่ใช้บันทึกข้อมูลจากการดักจับข้อมูลที่วิ่งบนเครือข่ายคอมพิวเตอร์ลงสู่สื่อบันทึกข้อมูลที่กำหนดไว้

3.2.3 Intrusion Detection System (IDS) Mode เป็นโหมดที่ใช้วิเคราะห์ข้อมูลบนเครือข่ายคอมพิวเตอร์โดยจะอาศัยกฎ (Rules) ในการตัดสินใจว่าข้อมูลใดเป็นการโจมตี

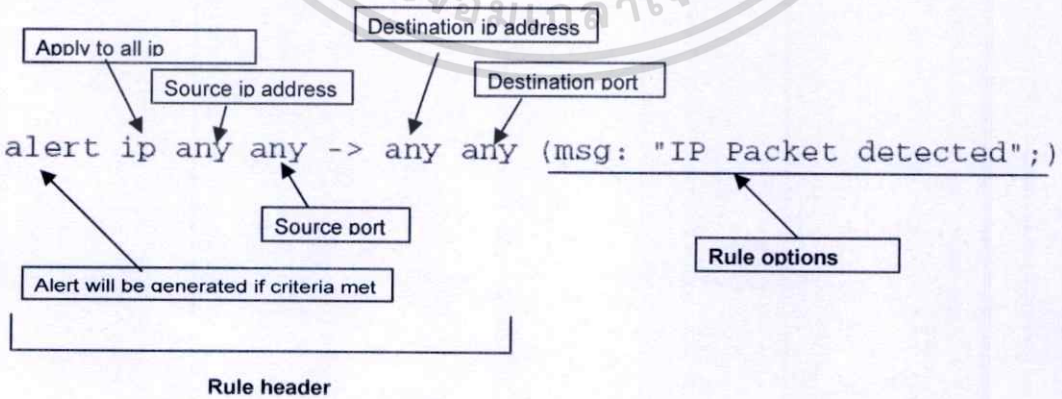
3.2.4 Intrusion Protection System (IPS) Mode หรือ Inline Mode สามารถทำการ Block, Drop session ที่ทำการเชื่อมต่อเข้าสู่เครือข่ายโดยจะอาศัยกฎ (Rules) ในการตัดสินใจเหมือนกับ IDS

3.3 กฎสำหรับตรวจจับการบุกรุกของ Snort (Snort Rules)

กฎสำหรับตรวจจับการบุกรุกของ Snort [6], [15] ประกอบด้วย 2 ส่วนคือ Rule Header และ Rule Body



รูปที่ 3.6 โครงสร้างสัญลักษณ์สำหรับตรวจจับการบุกรุก



รูปที่ 3.7 ตัวอย่างสัญลักษณ์สำหรับตรวจจับการบุกรุก

เพื่อความเข้าใจในการกำหนดรูปแบบสัญลักษณ์มากยิ่งขึ้น จึงยกตัวอย่างบางสัญลักษณ์จากกฎของ ddos.rules โดยประกอบด้วย Rule Header จะเก็บข้อมูลเกี่ยวกับกิจกรรมที่ Snort ต้องกระทำหรือข้อกำหนดที่ใช้ในการจับคู่กฎกับแพ็กและ Rule Body ซึ่งเป็นข้อความที่เขียนต่อจาก Rule Header ของกฎโดยจะอยู่ในวงเล็บ โดยส่วนของ Rule Body นั้นเกิดจากการนำ Rule Option ต่าง ๆ มาประกอบกันโดยจะใช้เครื่องหมาย " ; " เป็นตัวแบ่งแยก Rule Option ดังต่อไปนี้

- **Rule Header**

alert icmp $EXTERNAL_NET any -> $HOME_NET any

- **Rule Body**

(msg:"DDOS TFN Probe"; icmp_id:678; itype:8; content:"1234"; metadata:policy balanced-ips drop, policy security-ips drop; reference:arachnids,443; reference:cve,2000-0138; classtype:attempted-recon; sid:221; rev:7;)

3.3.1 Rule Header

จะเก็บข้อมูลเกี่ยวกับกิจกรรมที่ Snort ต้องกระทำ หรือข้อกำหนดที่ใช้ในการจับคู่กฎกับแพ็กเก็ต ซึ่งจะประกอบไปด้วยฟิลด์ต่าง ๆ ดังนี้

3.3.1.1 Rule Action เป็นสิ่งที่ใช้บอกว่า Snort ต้องทำอย่างไรกับแพ็กเก็ต ที่ตรงกับเงื่อนไขของกฎที่ตั้งไว้ โดยมีการกำหนดรูปแบบการตอบสนองไว้ 8 รูปแบบคือ

- ALERT คือการสร้างการแจ้งเตือนตามวิธีการเตือนที่เลือกไว้ พร้อมทั้งบันทึกข้อมูลลงในสื่อบันทึกข้อมูลที่กำหนดไว้
- LOG คือการบันทึกข้อมูลจากการดักจับแพ็กเก็ตลงในสื่อบันทึกข้อมูลที่กำหนดไว้
- PASS คือการปล่อยผ่านแพ็กเก็ตควรใช้กับแพ็กเก็ตที่มาจากเครือข่ายที่เชื่อถือได้
- ACTIVATE จะทำการแจ้งเตือน และสั่งให้ กฎแบบไดนามิกที่เกี่ยวข้องทำงาน
- DYNAMIC จะไม่ทำอะไรจนกว่าจะถูกกระตุ้นจากกฎแบบ ACTIVATE โดยเมื่อทำงานแล้วจะทำการบันทึก แพ็กเก็ตในสื่อบันทึกข้อมูลที่กำหนดไว้
- DROP สร้าง iptables ลบแพ็กเก็ตและบันทึกแพ็กเก็ต
- REJECT สร้าง iptables ลบแพ็กเก็ตและบันทึกแพ็กเก็ตหลังจากนั้นส่ง TCP reset ถ้าโปโตคอล TCP หรือ ICMP port ส่ง unreachable message ถ้าเป็น UDP โปโตคอล
- STOP สร้าง iptables ลบแพ็กเก็ตแต่ไม่บันทึกแพ็กเก็ต

3.3.1.2 Protocol คือโพรโตคอลที่ต้องการให้กฎตรวจสอบใน Snort เวอร์ชั่นปัจจุบันจะสนับสนุน 4 โพรโตคอลคือ IP ICMP TCP และ UDP และในอนาคตอาจจะสนับสนุนโปรโตคอลเพิ่มมากขึ้น เช่น ARP, IGRP, GRE, OSPF, RIP, IPX เป็นต้น

3.3.1.3 IP addresses หมายเลขไอพีแอดเดรสหรือหมายเลขเครือข่าย (IP/Network Address) และหมายเลขพอร์ตที่กำหนดในกฎเพื่อระบุ Source Information คือหมายเลขไอพีแอดเดรสหรือหมายเลขเครือข่าย (IP/Network Address) และหมายเลขพอร์ต (Port Number) ของต้นทางและDestination Information คือหมายเลขไอพีแอดเดรสหรือหมายเลขเครือข่าย (IP/Network Address) และหมายเลขพอร์ต (Port Number) ของปลายทาง การเขียนกฎนอกจากจะระบุเป็นหมายเลขเครือข่ายแล้วยังสามารถใช้ any ซึ่งหมายถึงทุกไอพีแอดเดรสเป็นต้น

3.3.1.4 Port Numbers การกำหนดพอร์ตสามารถทำได้ทั้งระบุหมายเลขพอร์ตตรง ๆ หรือกำหนดเป็นช่วงได้ดังตัวอย่างด้านล่างนี้

กำหนดหมายเลขพอร์ตเท่ากับ 6000

log tcp any any -> 192.168.1.0/24 ***6000***

กำหนดได้ทั้งหมายเลขพอร์ตเป็นช่วงตั้งแต่ 1 ถึง 1024

log udp any any -> 192.168.1.0/24 ***1:1024***

กำหนดช่วงของหมายเลขพอร์ตที่น้อยกว่าหรือเท่ากับ 6000

log tcp any any -> 192.168.1.0/24 ***:6000***

กำหนดช่วงของหมายเลขพอร์ตมากกว่า 500 ขึ้นไป

log tcp any any -> 192.168.1.0/24 ***500:***

กำหนดช่วงของหมายเลขพอร์ตที่น้อยกว่าหรือเท่ากับ 6010 ยกเว้น 6000

log tcp any any -> 192.168.1.0/24 ***!6000:6010***

3.3.1.5 Direction Operation คือทิศทางในการเคลื่อนที่ของแพ็กเก็ตที่จะตรวจสอบ มีอยู่ 2 ทิศทางด้วยกันคือ

-> หมายถึงพิจารณาเพียงทิศทางเดียว

<> หมายถึงพิจารณาทั้งสองทิศทาง

3.3.1.6 Activate/Dynamic Rules คือคู่ของกฎที่เพิ่มความสามารถให้ Snort โดยกฎที่เป็น Activate จะทำหน้าที่ แจ้งเตือน และ กฎที่เป็น Dynamic จะทำหน้าที่บันทึกข้อมูลที่ตรงกับเงื่อนไขที่กำหนดไว้

3.3.2 Rule Option

เป็นข้อความที่เขียนต่อจาก Rule Header ของกฎโดยจะอยู่ในวงเล็บ Rule Body นั้นเกิดจากการนำ Rule Option ต่าง ๆ มาประกอบกันโดยจะใช้เครื่องหมาย " ; " เป็นตัวแบ่งแยก **Rule Option** แบ่งออกเป็น 6 กลุ่มด้วยกันดังต่อไปนี้

3.3.2.1 Meta-data Rule Option

- **msg** ใช้สำหรับกำหนดข้อความที่ต้องการเก็บไว้ใน log หรือข้อความสำหรับแจ้งเตือน

  msg :"< ข้อความที่ต้องการ >";

- **reference** ใช้กำหนดแหล่งที่มาของกฎ ในกรณีที่กฎนี้ได้มาจากระบบอื่น

  ref :< id system >, < id >;

ตารางที่ 3.1 ระบบที่รองรับการอ้างอิงของ Snort

ระบบ	คำนำหน้า URL
bugtraq	http://www.securityfocus.com/bid/
cve	http://cve.mitre.org/cgi-bin/cvename.cgi?name=
nessus	http://cgi.nessus.org/plugins/dump.php3?id=
arachnids	(currently down) http://www.whitehats.com/info/IDS
mcafee	http://vil.nai.com/vil/dispVirus.asp?virus k=
url	http://

- **classtype** ใช้สำหรับจัดกลุ่มโดยแต่ละกลุ่มนั้นจะแบ่งโดยใช้ประเภทของการโจมตี

  classtype :< ชื่อประเภทของการโจมตี >;

ตารางที่ 3.2 ระดับความสำคัญของการตรวจจับการบุกรุกประเภทรุนแรง

ลำดับที่	ประเภทรุนแรง Critical Classifications (Priority 1)	
1	attempted-admin	พยายามเพิ่มสิทธิทียบเท่าผู้ดูแลระบบ
2	attempted-user	พยายามเพิ่มสิทธิทียบเท่าผู้ใช้งานในระบบ
3	shellcode-detect	ตรวจพบ Executable code
4	successful-admin	เพิ่มสิทธิเทียบเท่าผู้ดูแลระบบสำเร็จ
5	successful-user	เพิ่มสิทธิเทียบเท่าผู้ใช้งานในระบบสำเร็จ
6	trojan-activity	ตรวจพบโทรจันในระบบเครือข่าย
7	unsuccessful-user	เพิ่มสิทธิทียบเท่าผู้ใช้งานในระบบแต่ไม่สำเร็จ
8	web-application-attack	โจมตีการให้บริการทางเว็บ

ตารางที่ 3.3 ระดับความสำคัญของการตรวจจับการบุกรุกประเภทปานกลาง

ลำดับที่	ประเภทปานกลาง Intermediate Classifications (Priority 2)	
1	attempted-dos	พยายามขัดขวางการให้บริการ
2	attempted-recon	พยายามเจาะข้อมูล
3	bad-unknown	มีความเป็นไปได้ที่จะเป็นข้อมูลอันตราย
4	denial-of-service	ตรวจพบการพยายามขัดขวางการให้บริการ
5	misc-attack	การโจมตีที่มีรูปแบบหลากหลาย
6	non-standard-protocol	ตรวจพบโปรโตคอลที่ไม่มาตรฐาน
7	rpc-portmap-decode	ร้องขอการถอดรหัส RPC
8	successful-dos	พยายามขัดขวางการให้บริการ
9	successful-recon-largescale	เจาะข้อมูลได้จำนวนมาก
10	successful-recon-limited	เจาะข้อมูลได้จำนวนจำกัด
11	suspicious-filename-detect	ตรวจพบโปรโตคอลตรวจพบไฟล์ชื่อต้องสงสัย
12	suspicious-login	พยายามเข้าล็อกอินเข้าระบบด้วยชื่อที่ต้องสงสัย
13	system-call-detect	ตรวจพบการเรียกใช้ไฟล์ระบบ
14	unusual-client-port-connection	เครื่องลูกข่ายถูกใช้พอร์ตที่ไม่เคยงานมาก่อน
15	web-application-activity	การเข้าถึงช่องโหว่ของการให้บริการเว็บ

ตารางที่ 3.4 ระดับความสำคัญของการตรวจจับการบุกรุกประเภทความเสี่ยงต่ำ

ลำดับที่	ประเภทความเสี่ยงต่ำ Low-Risk Classifications (Priority 3)	
1	icmp-event	เหตุการณ์ ICMP ทั่ว ๆ ไป
2	misc-activity	การโจมตีที่มีรูปแบบหลากหลาย
3	network-scan	ตรวจพบการสแกนระบบเครือข่าย
4	not-suspicious	ข้อมูลที่ไม่ต้องสงสัย
5	protocol-command-decode	คำสั่งถอดรหัสโปรโตคอลทั่ว ๆ ไป
6	string-detect	ตรวจพบสตริงต้องสงสัย
7	unknown	ไม่ทราบชนิดของทราฟฟิค

- **sid** ใช้สำหรับกำหนดหมายเลขให้กับกฎของ Snort ค่า SID ต้องไม่ซ้ำกันและมีค่าเสมอ ค่าที่เป็นไปได้ของหมายเลขกฎมีดังนี้

< 100 สงวนไว้ใช้ในอนาคต

100 – 1,000,000 ใช้กับกฎที่สร้างโดย Snort.org

> 1,000,000 ใช้กับกฎที่ผู้ใช้สร้างขึ้นเอง (Local Rules)

sid :< หมายเลขกฎ >;

- **rev** ใช้สำหรับกำหนดหมายเลขเวอร์ชันให้กับกฎของ Snort

rev :< หมายเลขเวอร์ชันของกฎ >;

- **priority** ใช้สำหรับกำหนดความสำคัญให้กับกฎ

priority :< หมายเลขลำดับความสำคัญ >;

3.3.2.2 Payload Detection Rule Options ใช้กำหนดเนื้อหาที่ต้องการให้ Snort ตรวจสอบ

- **content** เป็นส่วนที่สำคัญของโปรแกรม Snort ช่วยให้ผู้ใช้งานตรวจสอบข้อมูลภายใน Payload ของแพ็กเก็ตได้ โดยข้อมูลหรือข้อความที่ต้องการตรวจสอบนั้นเป็นได้ทั้งตัวอักษรธรรมดา และแบบไบนารี ถ้าข้อความเป็นแบบไบนารีจะต้องใช้เครื่องหมาย " | " ครอบข้อความไว้ ดังตัวอย่างต้องการตรวจสอบข้อความ 5C 00 ก็ให้พิมพ์ดังนี้ content: "|5C 00|";

content: [!] "< ข้อความที่ต้องการให้ตรวจจับ >";

หมายเหตุ ถัดจากนี้ไปจะใช้ประโยคว่า รูปแบบที่ถูกกำหนด (The specified pattern) แทนข้อความที่ต้องการให้ตรวจจับของ Content

- **nocase** ใช้เพื่อกำหนดให้ Snort ค้นหาข้อความในส่วนของ Content โดยไม่ต้องสนใจว่าตัวอักษรนั้นจะ เป็นตัวอักษรพิมพ์เล็กหรือใหญ่

nocase;

- **rawbytes** ใช้เพื่อกำหนดให้ Snort ตรวจสอบแพ็กเก็ตจากข้อมูลดิบ (Raw packet data) ซึ่งเป็นข้อมูลที่ยังไม่ได้ผ่านการถอดรหัส

rawbyte;

- **depth** ใช้เพื่อกำหนดว่าจะให้ค้นหารูปแบบที่ถูกกำหนด ภายในแพ็กเก็ตเป็นจำนวนเท่าไหร่ (ความลึก)

depth :< จำนวนไบท์ที่ให้ค้นหา >;

- **offset** เป็นส่วนที่อนุญาตให้ผู้ใช้งานสามารถกำหนดจุดเริ่มต้นในการค้นหารูปแบบที่ถูกกำหนดภายในแพ็กเก็ต โดยจะบอกเป็นจำนวนไบท์ที่ต้องการข้าม ซึ่งจะนับจากไบท์แรก

offset :< จำนวนไบท์ที่ข้าม >;

- **distance** ใช้กำหนดจำนวนไบท์ที่ต้องการข้ามก่อนที่จะเริ่มค้นหารูปแบบที่ถูกกำหนด การนับนั้นจะเริ่มนับ ณ จุดสุดท้ายที่ค้นพบรูปแบบที่ถูกกำหนดตัวก่อนหน้า (Previous pattern match)

distance: *< จำนวนไบท์ >;*

- **within** ใช้กำหนดจำนวนไบท์ภายในแพ็กเก็ตที่ต้องการค้นหารูปแบบที่ถูกกำหนด ออกแบบมาให้ใช้คู่กับ Distance

within : *< จำนวนไบท์ >;*

- **http client body** ใช้กำหนดข้อมูลที่ต้องห้ามในส่วนที่เป็นการร้องขอ HTTP จากเครื่องลูกข่าย

http_client_body;

- **http uri** ใช้กำหนดข้อมูลที่ต้องห้ามในส่วนที่เป็น URI

http_uri;

- **uricontent** ใช้ตรวจหาข้อมูลในส่วนที่เป็น URI

uricontent:[!] < ข้อมูลในส่วนของ URI ที่ต้องการ >;

- **urilen** กำหนดความยาวต่ำสุดและสูงสุดหรือช่วงของความยาว URI

urilen: *จำนวนเต็ม<>จำนวนเต็ม;*

urilen: *[<,>] <จำนวนเต็ม>;*

- **isdataat** ตรวจสอบ payload ว่ามีข้อมูลที่ระบุสถานที่เพื่อดึงข้อมูลก่อนที่จะดูใน content

isdataat:*<จำนวนเต็ม>[,relative];*

- **pcre** อนุญาติให้ใช้เขียนกฎที่ใช้ภาษา perl ได้

pcre:[!]"(/<regex>/\m<delim><regex><delim>)[ismxAEGRUB]";

- **byte_test** ใช้ทดสอบข้อมูลแบบไบนารี กับค่าที่กำหนด

byte_test:*< จำนวนไบต์ >, [!]< operator >, <ค่า >, < offset > [,relative][,<endian>] [,<ตัวเลขของชนิด>, string];*

- **byte_jump** ใช้เพื่อให้ Snort ข้ามข้อมูลบางส่วนซึ่งตรงกับเงื่อนไขที่ใส่ลงไป

byte_jump: *< จำนวนไบต์>, < offset > [,relative] [,ตัวคูณ <ค่าตัวคูณ >] [,big] [,little][,string][,hex] [,dec] [,oct] [,align] [,from_beginning];*

- **ftpbounce** ตรวจจับการโจมตี FTP bounce attacks.

ftpbounce;

- **asn1** ตรวจจับการถอดรหัส plugin ของแพ็คเก็ตและช่องทางสื่อสาร อีกทั้งเฝ้าดูแพ็คเก็ตที่ประสงค์ร้าย

asn1: *เงื่อนไข [argument] [, เงื่อนไข [argument]] . . .*

- **cvs** ตรวจจับ plugin ของ Bugtraq-10384, CVE-2004-0396:"Malformed Entry Modified and Unchanged flag insertion" ปกติ CVS เซิร์ฟเวอร์ใช้พอร์ต 2401 และ 514 รวมทั้งพอร์ตปกติที่ใช้สำหรับ stream reassemble

cvs :<เงื่อนไข>;

3.3.2.3 Non-payload Detection Rule Options บอกให้ Snort ทราบว่าต้องการตรวจสอบข้อมูลภายใน IP Header ด้วย

- **fragoffset** ใช้สำหรับตรวจจับแพ็กเก็ตที่มีค่าในฟิลด์ Fragment Offset ตรงกับเงื่อนไขที่กำหนดลงไป ซึ่งเงื่อนไขนั้นสามารถใส่เครื่องหมาย " < " หรือ " > " ตามด้วยค่า fragment offset หรือจะใส่แค่ค่า fragment offset เพียงอย่างเดียวก็ได้

fragoffset:[<|>] < ค่า fragment offset >;

- **ttl** ใช้ตรวจสอบค่า Time To Live ที่ส่งมาในแพ็กเก็ต

ttl: *< จำนวน Time To Live ที่ต้องการตรวจสอบ >;*

- **tos** ใช้ตรวจสอบฟิลด์ Type Of Service (TOS) ใน IP Header ฟิลด์ TOS นี้ใช้ในการแสดงคำร้องขอการดำเนินการแบบพิเศษการดำเนินการแบบไม่ต้องหน่วงเวลาหรือการดำเนินการที่ต้องการความรวดเร็วสูง และอื่น ๆ แต่อย่างไรก็ตามปัจจุบันข้อมูลในฟิลด์นี้ไม่ได้รับความสนใจจากเราเตอร์แล้ว

tos: *< หมายเลข Type Of Service >;*

- **id** ใช้ตรวจสอบฟิลด์ fragment ID ใน IP Header

id: *< หมายเลข fragment >;*

- **ipopts** ใช้เพื่อตรวจสอบถ้าระบุไอพีแอดเดสตามเงื่อนไขได้

ipopts:*<rr|eol|nop|ts|sec|esec|lsrr|ssrr|satid|any>;*

- **fragbits** ใช้ตรวจสอบ fragment bit ซึ่งอยู่ที่ฟิลด์ flag ของ IP Header ฟิลด์นี้มีขนาด 3 บิตด้วยกันดังนี้

- R (Reserved Bit) มีค่าเป็น 0 เสมอ
- D (Don't Fragment) ถ้ามีค่าเป็น 1 แสดงว่าแพ็กเก็ตนี้ไม่สามารถแตกชิ้นส่วนได้
- M (More Fragment) แสดงว่าแพ็กเก็ตนี้มีการแตกชิ้นส่วน จะมีค่าเป็น 0 เมื่อเป็นชิ้นส่วนย่อยอันสุดท้าย

fragbits: *< ค่าของ fragment bit >;*

- **dsize** ใช้สำหรับทดสอบขนาดของ Payload ในแพ็กเก็ตว่ามีขนาดผิดแปลกไปจากปกติหรือไม่

dsize: *[< >] < ขนาดของ Payload > [< ขนาดของ Payload >];*

- **ip_protocol** ใช้เพื่อให้กฎนั้นตรวจจับแพ็กเก็ตซึ่งมีค่าที่อยู่ในฟิลด์ Protocol ตรงกับค่าที่กำหนดไว้ โดยค่าที่ใส่ไว้จะเป็นชื่อโปรโตคอลหรือหมายเลขที่ใช้แทนโปรโตคอลก็ได้ (ท่านสามารถดูว่าหมายเลขใดแทนโปรโตคอลอะไรได้ที่ /etc/protocols)

ip_proto: *[! < >] < ชื่อโปรโตคอล หรือหมายเลขที่ใช้แทน >;*

- **ip_option** เป็นส่วนเพิ่มเติมของกฎที่ช่วยให้ผู้ใช้สามารถระบุ IP Option ที่ต้องการตรวจจับได้ โดยปกติแล้วแพ็กเก็ตจะไม่มีข้อมูลในส่วนของ IP option ตารางที่ 3.5 แสดงรายการของ IP Option ที่ Snort ได้จัดเตรียมไว้

ตารางที่ 3.5 IP Option บนโปรแกรม Snort

IP Option	คำอธิบายอย่างสั้น
Eol	End of list ใช้แสดงว่าเป็นจุดสิ้นสุดในรายการของ Option
Lsrr	Loose source routing กำหนดการเลือกเส้นทางที่ฝั่งต้นทางและการบันทึกเส้นทางเป็นแบบหละหลวมโดยใช้ดาตาแกรม
Nop	No option ใช้เติมฟิลด์วางในรายการของ Option ให้เต็ม
Rr	Record route ให้บันทึกเส้นทางที่เลือก แต่ละเส้นทางคือแอดเดรส ที่อยู่ในช่องว่างที่จัดเตรียมไว้ และเป็นพื้นที่ในการปรับปรุงพอยเตอร์ที่ชี้ไปยังเราเตอร์
Satid	Stream identifier ถูกกำหนดขึ้นมาใช้กับเครือข่าย Atlantic Satellite Network ซึ่งปัจจุบันเลิกการใช้งานไปแล้ว
Sec	IP security option หรือที่รู้จักกันในชื่อ IPSec
Ssrr	Strict source routing กำหนดการเลือกเส้นทางที่ฝั่งต้นทางและการบันทึกเส้นทางเป็นแบบเคร่งครัด ความแตกต่างของการเลือกแบบนี้กับแบบหละหลวมคือเราเตอร์ระหว่างทางมีความยืดหยุ่น ในการเลือกเส้นทางให้แพ็กเก็ตได้น้อยกว่า
Ts	Time Stamp กำหนดให้บันทึกเวลาที่เดตาแกรมถูกประมวลผล

ipopts: *< option >;*

ข้อจำกัดของส่วนเพิ่มเติมนี้คือจะใส่ ipoption ได้เพียง 1 อย่างต่อกฎ 1 ข้อ

- **sameip** ใช้เพื่อให้กฎนั้นตรวจจับแพ็กเก็ตที่มี หมายเลขไอพีต้นทางและปลายทางเหมือนกัน

sameip;

3.3.2.4 TCP Option บอกให้ Snort ทราบว่าต้องการตรวจสอบข้อมูลภายใน TCP Header ด้วย

- **seq** เป็นส่วนเพิ่มเติมที่ใช้ตรวจสอบค่า TCP Sequence number ในแพ็กเก็ต

seq: *< หมายเลข Sequence >;*

- **ack** ใช้ตรวจสอบแพ็กเก็ตที่ผ่านกระบวนการ 3 way handshaking แล้ว (ACK Flag = 1) แต่มี Acknowledgment number เป็น 0 ซึ่งส่วนใหญ่จะเป็นแพ็กเก็ตที่เกิดจากโปรแกรม NMAP

ack: *< หมายเลขของ Acknowledgment >;*

- **window** ใช้ตรวจสอบขนาดของ window เพราะโปรแกรม back door บางตัวจะกำหนดขนาดของ Window ให้มีขนาดใหญ่มาก ๆ

window: *< ขนาดของ window >;*

- **flags** ใช้สำหรับตรวจสอบ TCP Flags โดยค่า flag ที่snort ตรวจสอบได้ ดังตารางที่ 3.6

flags: *< TCP Flag >;*

ตารางที่ 3.6 TCP Flag ที่โปรแกรม Snort สามารถตรวจสอบได้

TCP Flags	คำอธิบายอย่างสั้น
A (ACK)	ตรวจสอบเมื่อ ACK Flag มีค่าเป็น 1
F (FIN)	ตรวจสอบเมื่อ FIN Flag มีค่าเป็น 1
P (PSH)	ตรวจสอบเมื่อ PSH Flag มีค่าเป็น 1
R (RST)	ตรวจสอบเมื่อ RST Flag มีค่าเป็น 1
S (SYN)	ตรวจสอบเมื่อ SYN Flag มีค่าเป็น 1
U (URG)	ตรวจสอบเมื่อ URG Flag มีค่าเป็น 1
0	ตรวจสอบถ้า TCP Flag ไม่มีการเซตค่า (เป็น 0 หมดทุกตัว)
1	ตรวจสอบเมื่อบิตที่ 1 ในฟิลด์ Reserved ของ TCP Header มีค่าเป็น 1
2	ตรวจสอบเมื่อบิตที่ 2 ในฟิลด์ Reserved ของ TCP Header มีค่าเป็น 1

3.3.2.5 ICMP Option บอกให้ Snort ทราบว่าต้องการตรวจสอบข้อมูลภายใน ICMP Header

- **itype** ใช้ตรวจสอบค่าฟิลด์ ICMP Typeสำหรับค่าที่สามารถกำหนดได้นั้นดูได้ที่ decode.h แพ็กเก็ตที่มีค่า ICMP Type ผิดปกติบางครั้งอาจมาจากการโจมตีแบบ Denial of service (DOS)

itype: *< หมายเลขของ ICMP Type >;*

- **icode** ใช้กำหนดค่าฟิลด์ ICMP Code ที่จะตรวจสอบเกี่ยวข้องกับค่าของ ICMP Type

icode: *[< | >] < หมายเลขของ ICMP Code >*
[< > < หมายเลขของ ICMP Code >];

- **icmp_id** ใช้กำหนดค่าฟิลด์ ICMP ID ที่จะตรวจสอบ

icmp_id: *< หมายเลขของ ICMP ID >;*

- **icmp_seq** ใช้กำหนดค่าฟิลด์ ICMP Sequence ที่จะตรวจสอบ

icmp_seq: *< หมายเลขของ ICMP Sequence >;*

3.3.2.6 Miscellaneous เป็น Rule Option ที่ไม่สามารถจัดเข้ากลุ่มใดได้

- **logto** ใช้เพื่อกำหนดให้โปรแกรม Snort เก็บบันทึกแพ็กเก็ตซึ่งตรงกับกฎ ลงในไฟล์ที่กำหนด (Output log file) ข้อควรระวังคือส่วนเพิ่มเติมนี้ ไม่สามารถใช้งานได้ถ้า Snort อยู่ในโหมดการทำงานแบบ Binary logging

logto: *< ชื่อไฟล์ที่ต้องการเก็บผลลัพธ์ >;*

- **session** ใช้สำหรับดึงข้อมูลภายในแอพพลิเคชันที่ใช้ TCP Session ออกมา ตัวอย่างของข้อมูลเหล่านั้นได้แก่ ชื่อ รหัสผ่านของผู้ใช้ เป็นต้น

session: [printable | all];

- **rpc** ใช้สำหรับตรวจหาการให้บริการแบบ RPC บนเครือข่าย

rpc:*< หมายเลขแอพพลิเคชัน >, < [< หมายเลขเวอร์ชัน > | *] >,*
*< [< หมายเลขโพรซีเจอร์ > | *]>;*

หมายเหตุ ในส่วนของหมายเลขเวอร์ชันและหมายเลขโพรซีเจอร์สามารถใส่ครื่องหมาย * ได้โดยมีความหมายว่าเป็นเวอร์ชันและโพรซีเจอร์ใดก็ได้สนใจเฉพาะหมายเลขแอพพลิเคชั่นก็พอ

- **resp** เป็นส่วนที่จะพยายามปิดการเชื่อมต่อถ้ามีการแจ้งเตือนเกิดขึ้น

ตารางที่ 3.7 กลไกในการตอบสนอง

กลไกในการตอบสนอง คำอธิบายอย่างสั้น	
rst_snd	ส่งแพ็กเก็ต TCP-RST ผ่านทางซ็อกเก็ตที่ทำหน้าที่ส่งข้อมูล
rst_rcv	ส่งแพ็กเก็ต TCP-RST ผ่านทางซ็อกเก็ตที่ทำหน้าที่รับข้อมูล
rst_all	ส่งแพ็กเก็ต TCP-RST ไปทั้ง 2 ทิศทาง
icmp_net	ส่ง ICMP Net Unreach ไปยังผู้ส่งแพ็กเก็ต
icmp_host	ส่ง ICMP Host Unreach ไปยังผู้ส่งแพ็กเก็ต
icmp_port	ส่ง ICMP Port Unreach ไปยังผู้ส่งแพ็กเก็ต
icmp_all	ส่ง ICMP ที่กล่าวมาข้างต้นทั้งหมดไปยังผู้ส่งแพ็กเก็ต

หมายเหตุ: ผู้ใช้สามารถกำหนดกลไกในการตอบสนองได้มากกว่า 1 วิธี

resp: *< กลไกในการตอบสนอง > [,< กลไกในการตอบสนอง > [,< กลไกในการตอบสนอง >]];*

- **react** เป็นส่วนที่กำหนดว่าจะให้ทำอะไรหลังจากกลไกการตอบสนองทำงานไปแล้ว ใช้ตอบสนองนี้จะใช้งานได้ก็ต่อเมื่อมีการใช้ resp ค่าที่เป็นไปได้ของ react มี 2 ค่าคือ block กับ warn และส่วนเพิ่มเติมอีก 2 ค่าคือ msg กับ proxy รายละเอียดดังข้างล่างนี้

block จะทำการปิดการเชื่อมต่อ และส่งข้อความที่ผู้ใช้สังเกตเห็นได้

warn ส่งข้อความเตือนที่ผู้ใช้สังเกตเห็นได้

msg จะรวมข้อความในส่วนเพิ่มเติม msg ไปกับข้อความที่เกิดจากการ block

proxy : < หมายเลขพอร์ตของ proxy > จะส่งข้อความที่ผู้ใช้สังเกตเห็นได้ไปทางหมายเลขพอร์ตที่กำหนด

react: *< block | warn [, msg | proxy: < หมายเลขพอร์ต >];*

- **tag** นำมาใช้เพื่อบันทึกแพ็กเก็ตอื่น ๆ ที่มีความเกี่ยวข้องกับแพ็กเก็ตที่กระตุ้นให้กฎทำงาน (trig) อาทิเช่นแพ็กเก็ตที่มาจากเครื่องเดียวกันกับแพ็กเก็ตที่กระตุ้นให้กฎทำงาน

tag: *< type >, < count >, < metric >, [direction];*

type มีค่าได้ 2 อย่างคือ session หรือ host

count คือจำนวนที่ต้องการ tag ไว้ (หน่วยของ count นั้นจะอยู่ที่ metric)

metric คือหน่วยของจำนวนที่ต้องการ tag มีให้เลือก 2 อย่างคือ second กับ packet

3.4 ประเภทของกฎในโปรแกรม Snort

กฎทั้งหมดในโปรแกรม WINSNORT สามารถ Download จาก http://www.winids.com ในการทดลองนี้ได้ใช้ packet winids_support_pak-012609 แบ่งออกเป็น 52 ไฟล์ และ 13,618 สัญลักษณ์ดังตารางที่ 3.8

ไฟล์ที่เกี่ยวข้องกับการทำงานของโปรแกรม Snort ในโหมด NIDS

- Rule files (*.rules) คือไฟล์ที่บรรจุสัญลักษณ์หรือสัญญาณที่บ่งบอกถึงการบุกรุก
- snort.conf คือไฟล์ที่เก็บค่าพารามิเตอร์ต่าง ๆ บนโปรแกรม Snort
- reference.config คือไฟล์ที่ใช้จับคู่ระหว่าง ชื่อแหล่งอ้างอิงที่ปรากฏบนกฎ (หลัง Rule Option "reference") กับ URL ของแหล่งที่มาของกฎ ประโยชน์ของไฟล์ reference.config คือช่วยให้การจัดการกับแหล่งอ้างอิงหรือแหล่งที่มาของกฎทำได้ง่าย อย่างเช่นกรณีมีการเปลี่ยนแปลง URL ของแหล่งที่มาของกฎ ก็เพียงเข้าไปแก้ไขในไฟล์นี้เท่านั้น ไม่ต้องไปตามแก้ไขในไฟล์กฎทุก ๆ ไฟล์ที่เกี่ยวข้อง
- classification.config คือไฟล์ที่เก็บรายละเอียดประเภทของสัญลักษณ์สำหรับตรวจจับการบุกรุก ซึ่งรายละเอียดเหล่านั้นก็จะมีชื่อคำอธิบายสั้น ๆ และค่าความสำคัญของกลุ่มสัญลักษณ์ (Rule files) โดยไฟล์นี้มีความสัมพันธ์โดยตรงกับ Rule Option ที่ชื่อ "classtype" ในการใช้งานถ้าต้องการสร้าง ลบ หรือแก้ไข ประเภทของกลุ่มสัญลักษณ์สำหรับตรวจจับการบุกรุก ผู้ดูแลระบบต้องเข้าไปแก้ไขในไฟล์นี้

ตารางที่ 3.8 ชื่อไฟล์ของกฎที่ใช้ในการตรวจจับการบุกรุก

ลำดับที่	ชื่อไฟล์กฎ	ความหมาย
1	attack-responses.rules	รูปแบบของการตอบสนองเมื่อมีการโจมตีหรือเข้าข่ายว่าจะโจมตีในระบบ
2	backdoor.rules	การโจมตีที่อาศัยช่องโหว่ที่มาจากโปรแกรมประเภทโทรจัน
3	bad-traffic.rules	รูปแบบของแพ็กเก็ตที่ไม่เคยมีการพบเห็นมาก่อนในระบบเครือข่ายต่าง ๆ
4	chat.rules	การโจมตีที่อาศัยช่องทางของโปรแกรมที่ใช้ในการแชท เช่น AIM ICQ และ IRC เป็นต้น

ตารางที่ 3.8 ชื่อไฟล์ของกฎที่ใช้ในการตรวจจับการบุกรุก (ต่อ)

ลำดับที่	ชื่อไฟล์กฎ	ความหมาย
5	content-replace	แก้ไขข้อความบนโปรแกรมที่ใช้ในการแชท เช่น MSN, Yahoo Messenger, Jabber, AIM และ IRC เป็นต้น
6	ddos.rules	การโจมตีที่มาได้จากหลาย ๆ จุด เพื่อทำให้เซิร์ฟเวอร์บนระบบเครือข่ายปฏิเสธการให้บริการ
7	deleted.rules	แจ้งเตือนเมื่อมีการลบสถานะการทำงานของการให้บริการต่าง ๆ
8	dns.rules	การโจมตีเครื่องเซิร์ฟเวอร์ที่ทำหน้าที่เป็น DNS server
9	dos.rules	การโจมตีที่มีจุดประสงค์เพื่อทำให้เครือข่ายรวมทั้งเซิร์ฟเวอร์บนเครือข่ายปฏิเสธการให้บริการ
10	experimental.rules	ใช้เพื่อทดสอบสัญลักษณ์ที่สร้างขึ้นใหม่ก่อนนำไปใช้จริง
11	exploit.rules	การโจมตีที่ใช้การบุกรุกผ่านทางช่องโหว่ต่าง ๆ
12	finger.rules	การ Finger เข้ามาและมีคำสั่งอื่นติดมาด้วย ทำให้ผู้ใช้คำสั่งนั้นสามารถดูข้อมูลบางอย่างในเครื่องได้
13	ftp.rules	การโจมตีที่อาศัยช่องทางของ FTP
14	icmp-info.rules	การกระทำที่น่าสงสัยว่าจะเข้าข่ายการโจมตีโดยใช้โปรโตคอล ICMP เป็นเครื่องมือในการสแกน
15	icmp.rules	การโจมตีโดยใช้โปรโตคอล ICMP เป็นเครื่องมือในการสแกน
16	imap.rules	การโจมตีที่อาศัยช่องทางของ IMAP
17	info.rules	การกระทำที่น่าสงสัยว่าจะเข้าข่ายการโจมตี ซึ่งถ้าหากมีการแจ้งเตือนบ่อยครั้ง อาจสรุปได้ว่ามีการโจมตีเกิดขึ้น
18	local.rules	ตรวจสอบโดยใช้สัญลักษณ์ที่สร้างขึ้นมาเอง
19	misc.rules	ตรวจสอบการโจมตีจากหลากหลายรูปแบบ

ตารางที่ 3.8 ชื่อไฟล์ของกฎที่ใช้ในการตรวจจับการบุกรุก (ต่อ)

ลำดับที่	ชื่อไฟล์กฎ	ความหมาย
20	multimedia.rules	การโจมตีผ่านทางบิตสตรีม(Streaming Multimedia Technologies)
21	mysql.rules	การโจมตีที่อาศัยช่องโหว่ของฐานข้อมูล Mysql
22	netbios.rules	การโจมตีที่อาศัยช่องโหว่ NETBIOS ที่เป็นอินเทอร์เฟสระหว่างระบบปฏิบัติการกับ Hardware
23	nntp.rules	การโจมตีที่อาศัยช่องโหว่ของโปรโตคอล Network News Transmission Protocol หรือ ยูสเน็ต
24	oracle.rules	การโจมตีที่อาศัยช่องโหว่ของฐานข้อมูล Oracle
25	other-ids.rules	การโจมตีที่ใช้ในการตรวจหาการตรวจจับการบุกรุกอื่นที่ติดตั้งอยู่ในระบบเครือข่าย
26	p2p.rules	ตรวจจับการรับส่งข้อมูลที่ใช้โปรโตคอลแบบ Peer to Peer ซึ่งละเมิดนโยบายบริษัท
27	policy.rules	ตรวจจับการละเมิดกฎซึ่งผู้ดูแลระบบตั้งไว้
28	pop2.rules	การโจมตีที่มีการใช้โปรโตคอล POP2
29	pop3.rules	การโจมตีที่มีการใช้โปรโตคอล POP3
30	porn.rules	ตรวจจับสื่อลามกอนาจาร
31	rpc.rules	การโจมตีผ่านระบบสั่งงานระยะไกล (Remote Procedure Call) เพื่อล้วงเอาข้อมูลสำคัญออกมาแสดงให้เห็น
32	rservices.rules	การโจมตีผ่านระบบที่ต้องการติดต่อจากระยะไกลซึ่งมีการทำ access control list เพื่อให้สิทธการเข้าถึงข้อมูล
33	scan.rules	แจ้งเตือนเมื่อถูกสแกนจากโปรแกรมสำเร็จรูปชนิดต่าง ๆ ยกเว้น Web scanner ซึ่งอยู่ในกฎของเว็บ
34	shellcode.rules	แจ้งเตือนเมื่อมีการใช้คำสั่ง Shellcode

ตารางที่ 3.8 ชื่อไฟล์ของกฎที่ใช้ในการตรวจจับการบุกรุก (ต่อ)

ลำดับที่	ชื่อไฟล์กฎ	ความหมาย
35	smtp.rules	การโจมตีที่มีการใช้โปรโตคอล SMTP
36	specific-threats	แจ้งเตือนการบุกรุกจาก Spyware และเทคโนโลยีไม่พึงประสงค์
37	snmp.rules	ตรวจสอบความผิดปกติของการใช้บริการโปรโตคอล SNMP
38	spyware-put	ตรวจสอบ Spyware ที่ฝังตัวอยู่ในระบบคอมพิวเตอร์
39	sql.rules	ตรวจสอบคำสั่งที่ใช้เพื่อหาประโยชน์จากช่องโหว่จากคำสั่ง SQL
40	telnet.rules	ตรวจสอบความผิดปกติของการใช้บริการ TELNET
41	tftp.rules	ตรวจสอบไฟล์อันตรายที่ส่งผ่านบริการ TFTP
42	virus.rules	ตรวจสอบไวรัสไฟล์
43	voip.rules	ตรวจสอบความผิดปกติของการใช้บริการ Voice over IP
44	web-attacks.rules	ตรวจสอบคำสั่งที่ใช้เพื่อหาประโยชน์จากช่องโหว่จากการให้บริการบนเว็บ
45	web-cgi.rules	ตรวจสอบคำสั่งที่ใช้เพื่อหาประโยชน์จากช่องโหว่จากการให้บริการเว็บโดยโปรแกรมCGI
46	web-client.rules	การโจมตีโดยใช้คำสั่งเพื่อหาประโยชน์จากช่องโหว่ของโปรแกรมเว็บของเครื่องลูกข่าย
47	web-coldfusion.rules	การโจมตีโดยใช้คำสั่งเพื่อหาแก้ไขข้อมูลบนเว็บ
48	web-frontpage.rules	ตรวจสอบคำสั่งที่ใช้เพื่อหาประโยชน์จากช่องโหว่จากการให้บริการเว็บโดยโปรแกรม Front Page
49	web-iis.rules	ตรวจสอบคำสั่งที่ใช้เพื่อหาประโยชน์จากช่องโหว่จากการให้บริการ IIS Web server

ตารางที่ 3.8 ชื่อไฟล์ของกฎที่ใช้ในการตรวจจับการบุกรุก (ต่อ)

ลำดับที่	ชื่อไฟล์กฎ	ความหมาย
50	web-misc.rules	ตรวจสอบคำสั่งที่ใช้เพื่อหาประโยชน์จากช่องโหว่จากการให้บริการเว็บหลากหลายรูปแบบ
51	web-php.rules	ตรวจสอบคำสั่งที่ใช้เพื่อหาประโยชน์จากช่องโหว่จากการให้บริการเว็บโดยโปรแกรม PHP
52	X11.rules	ตรวจสอบความผิดปกติของคุกกี้ metadata: service x11

3.5 รูปแบบของสัญลักษณ์

รูปแบบของสัญลักษณ์ที่ถูกเขียนขึ้นเพื่อประโยชน์ในการตรวจจับการบุกรุกประกอบด้วย

- **สัญลักษณ์ที่เขียนขึ้นในการตรวจสอบข้อมูลตาม ห่วงโซ่ของเหตุการณ์ (Chain of event)** ซึ่งส่วนใหญ่เป็นข้อมูลปกติ เช่น การเล่นเว็บ, Mail, Chat, Upload / Download, VoIP บางโปรแกรม ทั้งหมดนี้รวมถึงการ Authentication ตาม Application ต่าง ๆ เช่น MSN, Yahoo, SkyP เป็นต้น

- **สัญลักษณ์ที่เขียนขึ้นในการตรวจสอบข้อมูลที่มีความเสี่ยงและมีความผิดปกติ (Threat Data)** ซึ่งในส่วนนี้เราใช้เทคนิคระบบ IDS/IPS (Intrusion Detection & Prevention System) มาช่วย

3.6 ส่วนประกอบของไฟล์ Snort.conf

โครงสร้างตรวจจับการบุกรุกโดยใช้ของโปรแกรม Snort นั้นจะใช้ไฟล์ Snort.conf ที่ประกอบไปด้วย 4 ส่วนหลัก เพื่อกำหนดคุณสมบัติการทำงานของโปรแกรมดังนี้

- ส่วนที่ใช้กำหนดค่าพารามิเตอร์ของเครือข่ายที่ต้องการตรวจจับการบุกรุก
- ส่วนที่ใช้กำหนดค่า Preprocessors
- ส่วนที่ใช้กำหนด Output plugin
- ส่วนที่ใช้กำหนดไฟล์กฎในการตรวจจับการบุกรุก

3.7 โปรแกรมสำหรับติดตั้ง WINSNORT

ในส่วนนี้จะกล่าวถึงรายละเอียดของซอฟต์แวร์ที่นำมาสร้าง WINSNORT Sensor ที่สามารถ Download ได้จาก http://www.winids.com ในการทดลองนี้ได้ใช้ packet **winids_support_pak-012609** ซึ่งมีดังต่อไปนี้

- **Snort_2_8_2_2_Installer.exe** คือไฟล์สำหรับติดตั้งโปรแกรม Snort
- **WinPcap_4_0_2.exe** คือไฟล์ที่ติดตั้งลงไปเพื่อทำให้เครื่องคอมพิวเตอร์นั้นสามารถดักจับข้อมูลบนเครือข่ายได้
- **Snortrules-snapshot-CURRENT.ZIP** คือไฟล์ Snort Rules เป็นไฟล์ที่บรรจุเงื่อนไขหรือเกณฑ์และรูปแบบการบุกรุกประเภทต่าง ๆ ไว้เพื่อให้โปรแกรม Snort ใช้เป็นฐานข้อมูลในการเปรียบเทียบกับแพ็คเก็ตที่อ่านได้จากระบบเครือข่าย
- **mysql-essential-5.0.51b-win32.msi** คือโปรแกรมระบบฐานข้อมูล MySQL Server
- **mysql-gui-tools-5.0-r12-win32.msi** คือชุดของโปรแกรมบริหารฐานข้อมูล MySQL
- **php-5.2.8-Win32.zip** คือไฟล์ที่ติดตั้งเพื่อทำให้เครื่องเวปเซิร์ฟเวอร์สามารถรันสคริปต์ได้ ใช้ในการสร้าง ไดนามิกเว็บเพจ
- **ADODB498.zip** คือไฟล์ที่ติดตั้งลงไปเพื่อทำให้ PHP สามารถติดต่อกับฐานข้อมูลได้
- **BASE-1.4.0.zip** คือโปรแกรมที่เขียนด้วย PHP ที่แสดงข้อมูลการบุกรุกรายงานปละสถิติต่าง ๆ
- **IIS 5.1** มีทำหน้าที่เป็นเวปเซิร์ฟเวอร์

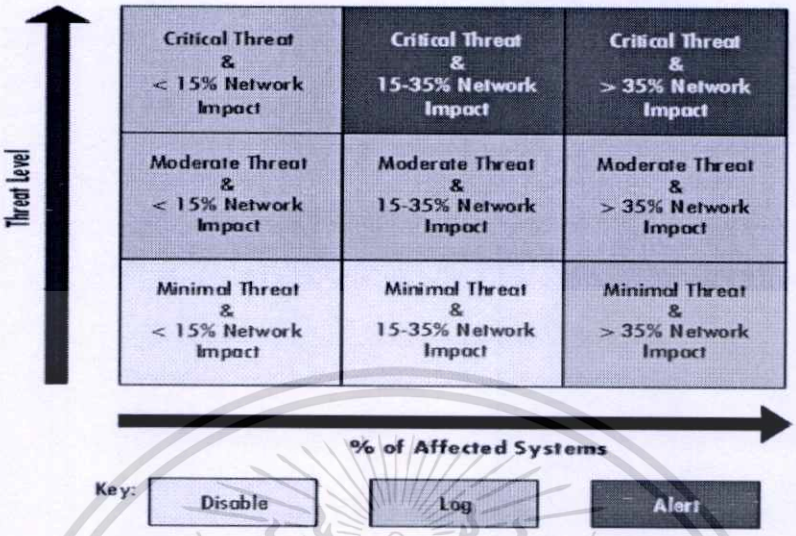
3.8 การกำหนดชุดของกฎที่ใช้ตรวจสอบการบุกรุก

การกำหนดใช้กฎของ Snort เพื่อเป็นการตั้งค่าให้กับระบบตรวจจับการบุกรุกในระบบเครือข่ายอย่างเหมาะสมนั้นนิยมยึดถือหลักเกณฑ์สองข้อหลัก ๆ ดังต่อไปนี้ [4]

3.8.1 การระบุชนิดของโปรโตคอลและบริการที่เปิดใช้บนระบบเครือข่าย เช่น ถ้ามีเพียงบริการของ NetBIOS และ HTTP บนระบบเครือข่ายก็ควรมีการใช้เพียงกฎที่อ้างถึงบริการเหล่านี้เท่านั้น ส่วนกฎอื่น ๆ ที่กำหนดถึงความพยายามเชื่อมต่อขอใช้บริการจากภายนอกซึ่งไม่ได้เปิดให้บริการนั้นควรจะมีไว้เพื่อเก็บบันทึกข้อมูล (log) ของทราฟฟิก เช่นกัน

3.8.2 การกำหนดระดับความสำคัญของสภาวะแวดล้อมบนระบบเครือข่าย เช่น ถ้าเป็นระบบเครือข่ายที่ใช้โดยกลุ่มผู้พัฒนาระบบ การกำหนดใช้กฎในการตรวจจับการบุกรุกอาจจะไม่ต้องเข้มงวดเท่ากับการกำหนดกฎสำหรับระบบเครือข่ายสำหรับฝ่ายการเงิน หรือระบบเครือข่ายที่เปิดการให้บริการสู่สาธารณะ เป็นต้น

ในรูปที่ 3.8 แสดงวิธีการแบ่งประเภทของกฎ (Rulesets) เพื่อตรวจสอบและตอบสนองต่อการบุกรุกตามระดับความรุนแรงของการคุกคาม (threat level) อย่างเหมาะสม ดังตัวอย่างต่อไปนี้



รูปที่3.8 การแยกประเภทของกฎ

- การคุกคามในระดับรุนแรง (Critical threats) อาทิเช่น SQL Slammer worm, CodeRed และ IIS Unicode attacks ซึ่งสามารถค้นหาและบุกรุกผ่านช่องโหว่ที่ยังไม่ได้รับการแก้ไขรวมถึงการแพร่กระจายอย่างรวดเร็ว

- การคุกคามในระดับปานกลาง (Moderate threats) อาทิเช่น MDAC remote buffer overflow, Wu-FTP buffer overflow, OpenSSL bugs ซึ่งทำให้เกิดบัฟเฟอร์ล้นจนไม่สามารถให้บริการได้ตามปกติส่วนใหญ่ไม่ได้ส่งผลกระทบอย่างกว้างขวาง

- การคุกคามในระดับเล็กน้อย (Minimal threats) อาทิเช่น Bind TSIG, การใช้ช่องโหว่ของ CGI และโปรโตคอล SMTP เป็นต้น ซึ่งทำให้ระบบไม่สามารถให้บริการได้ตามปกติ อย่างไรก็ตามช่องโหว่นี้ไม่ได้เป็นช่องโหว่ที่มีความเสี่ยงสูง

เครื่องมือนี้ใช้เพียงช่วยสร้างเกณฑ์ขั้นต่ำ (Baseline) ให้เราสามารถกำหนดระดับของการถูกบุกรุกว่าเกี่ยวข้องกับระบบภายนอกหรือระบบที่ให้บริการอยู่ภายในโดยนำทั้งสองส่วนมาวิเคราะห์ร่วมกันและทำการตัดสินใจบนช่วงเวลานั้นๆ

บทที่ 4

การทดลอง

ระบบตรวจจับการบุกรุก (Network Intrusion Detection System : NIDS) เป็นเครื่องมือที่สำคัญอย่างยิ่งในระบบเครือข่ายคอมพิวเตอร์ปัจจุบันที่จะใช้ในป้องกันการบุกรุกในระบบเครือข่าย ดังนั้นจึงได้มีการศึกษาเพื่อหาค่าประสิทธิภาพของการตรวจจับการบุกรุกโดยงานวิจัยต่าง ๆ เช่นงานวิจัยเรื่อง "Investigation of the Intrusion Detection System "Snort" Performance" [1] ได้ทำการวิจัยถึงปัจจัยของการใช้ทรัพยากรฮาร์ดแวร์และซอฟท์แวร์ระบบจัดการฐานข้อมูลที่แตกต่างกันว่าส่งผลกระทบต่อประสิทธิภาพการตรวจจับการบุกรุกของโปรแกรม Snort หรือไม่อย่างไร เมื่อมีการส่งข้อมูลบนระบบเครือข่ายคอมพิวเตอร์ในอัตราความเร็วที่แตกต่างกัน ส่วนงานวิจัยเรื่อง "Characterizing the Performance of Network Intrusion Detection Sensors" [2] ได้ทำการวิจัยถึงประสิทธิภาพของระบบตรวจจับการบุกรุกเมื่อถูกติดตั้งลงบนระบบปฏิบัติการที่แตกต่างกัน โดยได้ทดสอบกับระบบปฏิบัติการ Linux 3.0 kernel 2.4.19 และ FreeBSD 4.5 รวมทั้งศึกษาเปรียบเทียบประสิทธิภาพเมื่อใช้ตัวประมวลผลเดียว (Single processor) และตัวประมวลผลคู่ (Dual processor) ซึ่งก็ได้แสดงให้เห็นว่าตัวประมวลผลคู่ไม่ได้มีนัยที่จะช่วยเพิ่มประสิทธิภาพในการตรวจจับการบุกรุกให้มากขึ้นเท่าที่ควรเมื่อเทียบกับค่าใช้จ่ายที่เพิ่มมากขึ้น

4.1 การทดสอบหาค่าประสิทธิภาพการตรวจจับการบุกรุก

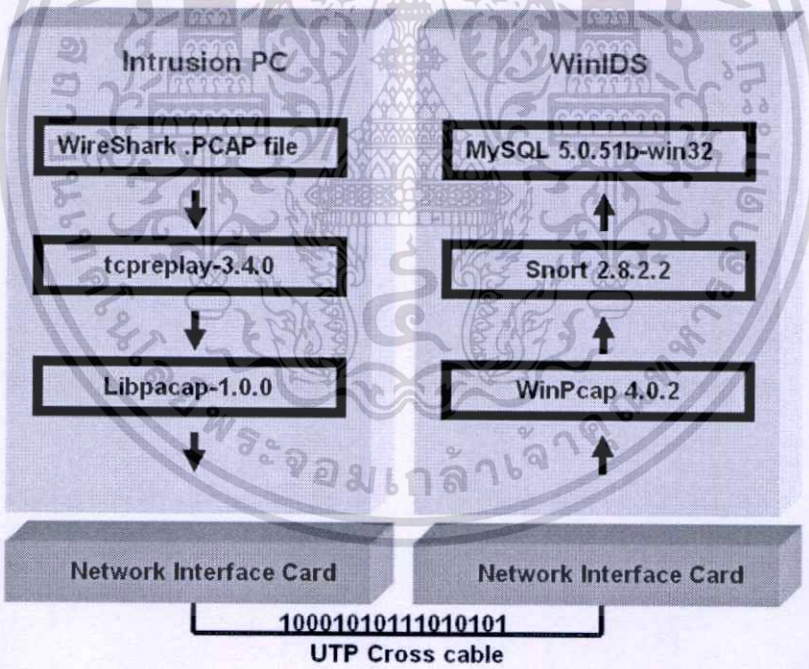
เนื่องจากในปัจจุบันนี้ ผู้บุกรุกมักจะค้นหาวิธีใหม่ ๆ เพื่อเจาะระบบเครือข่ายคอมพิวเตอร์อยู่เสมอ จึงทำให้จำนวนกฎของโปรแกรม Snort ที่ถูกใช้งานมีแนวโน้มเพิ่มมากขึ้นเรื่อย ๆ และจากสมมุติฐานว่าถ้าหากมีการเพิ่มจำนวนของกฎและสัญลักษณ์สำหรับการตรวจจับการบุกรุกจะทำให้การทำงานของโปรแกรม Snort เกิดปัญหาประสิทธิภาพในการตรวจจับการบุกรุกได้ลดลงเมื่อความเร็วและปริมาณของการส่งข้อมูลในระบบเครือข่ายคอมพิวเตอร์เพิ่มขึ้น งานวิจัยนี้จึงได้เสนอแนวทางในการประเมินประสิทธิภาพของระบบตรวจจับการบุกรุกของโปรแกรมตรวจจับการบุกรุกระบบเครือข่ายคอมพิวเตอร์ โดยการทดสอบกับโปรแกรม Snort เวอร์ชั่น 2.8.2.2 เพื่อหาผลกระทบเนื่องจากการเพิ่มขึ้นของจำนวนกฎ (Rules) และสัญลักษณ์ (Signatures) ที่อาจจะส่งผลต่อประสิทธิภาพของการตรวจจับการบุกรุกในสถานการณ์ที่ปริมาณและอัตราความเร็วในการส่งข้อมูลผ่านระบบเครือข่ายมีระดับแตกต่างกัน

การทดลองได้ดำเนินการด้วยการติดตั้งโปรแกรม Snort version 2.8.2.2 บนระบบปฏิบัติการ Windows XP บนเครื่องพีซี 1 เครื่อง และจำลองการทำงานของเครื่องผู้บุกรุก (Intrusion PC) ด้วยเครื่องพีซีอีกเครื่องหนึ่งที่ติดตั้งโปรแกรม TCPRplay-3.4.0 เพื่อทำการอ่าน

ไฟล์ข้อมูลจำลองการบุกรุกจากโปรแกรม Zenmap4.98 โดยโปรแกรม TCPReplay-3.4.0 จะทำหน้าที่ในการปรับความเร็วของการส่งผ่านข้อมูลที่อ่านจากไฟล์ดังกล่าวให้มีความเร็วเพิ่มขึ้นในแต่ละรอบของการทดลองเพื่อส่งผ่านไปยังสายสื่อสาร (cross line) ที่เชื่อมโยงระหว่างเครื่องพีซีทั้งสองเครื่องนั้น ดังโครงสร้างที่แสดงไว้ในรูปที่ 4.1 โดยเครื่องพีซีทั้งสองนั้นมีรายละเอียดดังตารางที่ 4.1

ตารางที่ 4.1 รายละเอียดเครื่องเซิร์ฟเวอร์ที่ใช้ในการทดลอง

Model	HP dx5150 MT
CPU	Athlon 64 3200+ (2.0Ghz/512KB cache/ 2000 MHz)
RAM	2GB DDR2-400
NIC	Broadcom NetXtreme Gigabit Ethernet
HDD	80GB SATA 1.5Gb/s (7200rpm) Hard Drive



รูปที่ 4.1 การทดสอบเพื่อค่าหาประสิทธิภาพของ Snort

การกำหนดรูปแบบการจำลองการบุกรุกจากโปรแกรม Zenmap4.98 มีรายละเอียดดังต่อไปนี้

ข้อมูลการจำลองการบุกรุกโดยโปรแกรม Zenmap4.98

- จำนวนของทราฟฟิกบุกรุกรวม 100,045 แพ็คเก็ต แบ่งออกเป็น TCP 19.45%, UDP 41.64%, ICMP 38.90%

- ค่าเฉลี่ยความเร็วของการส่งข้อมูล (Average data transmission rate) 0.127 Mbit/sec

จำนวนครั้งของการโจมตีทั้งหมดโดยโปรแกรม Zenmap 4.98 แยกตามชนิดการโจมตีได้เป็น

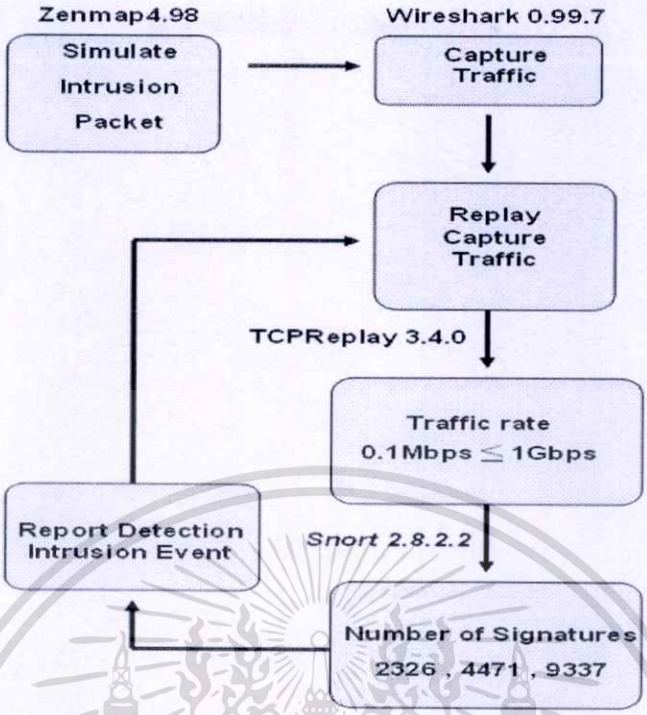
- การส่ง Executable code (Shellcode-detect) จำนวน 351 ครั้ง
- การพยายามเจาะระบบ (Attempted-recon) จำนวน 164 ครั้ง
- การบุกรุกรูปแบบที่ไม่สามารถระบุได้ (Misc-attack) จำนวน 328 ครั้ง
- จำนวนรวมของการบุกรุกทั้งหมด 843 ครั้ง

และเพื่อพิสูจน์ว่าจำนวนของกฎและสัญลักษณ์มีผลต่อประสิทธิภาพในการตรวจจับการบุกรุกหรือไม่ จึงได้ทดสอบโดยจัดกลุ่มของกฎและสัญลักษณ์สำหรับตรวจจับตามรูปแบบการบุกรุกออกเป็น 3 ชุด โดยในแต่ละรอบของการทดลองจะเพิ่มอัตราความเร็วในการส่งแพ็คเก็ตข้อมูลสื่อสารทั่วไปด้วยการใช้โปรแกรม TCPReplay-3.4.0 แล้วทำการตรวจวัดจำนวนครั้งที่โปรแกรม Snort สามารถตรวจจับการบุกรุกได้ เพื่อเปรียบเทียบกับการทดลองแต่ละรอบที่กำหนดในโปรแกรม Snort โดยที่กฎและสัญลักษณ์ชุดต่าง ๆ มีข้อมูลที่แตกต่างกัน 3 ชุดคือ กลุ่มที่หนึ่งมีจำนวน 9,337 สัญลักษณ์ กลุ่มที่สองมีจำนวน 4,471 สัญลักษณ์ และกลุ่มที่สามมีจำนวน 2,326 สัญลักษณ์ ดังแสดงในตารางที่ 4.2

ตารางที่ 4.2 กลุ่มของจำนวนสัญลักษณ์สำหรับการทดลอง

ชื่อกฎ	สัญลักษณ์ทั้งหมด	กลุ่มที่ 1	กลุ่มที่ 2	กลุ่มที่ 3
icmp-info.rules	22	22	22	22
icmp.rules	93	93	93	93
misc.rules	152	152	152	152
netbios.rules	5,786	4,159	4,159	2,014
shellcode.rules	29	29	29	29
snmp.rules	19	16	16	16
Other rules	7,517	4,266	n/a	n/a
สัญลักษณ์รวม	13,618	9,337	4,471	2,326

โดยเครื่องผู้บุกรุก (Intruder PC) จะทำการจำลองความเร็วของการส่งข้อมูลด้วยโปรแกรม TCPReplay-3.4.0 ในระดับอัตราความเร็วในการสื่อสารข้อมูลเป็น 0.127 Mbps, 100 Mbps, 200 Mbps, 300 Mbps, 400 Mbps, 500 Mbps, 600 Mbps, 700 Mbps, 800 Mbps, 900 Mbps และ 1 Gbps ตามลำดับ แล้วทำการตรวจวัดจำนวนครั้งที่โปรแกรม Snort สามารถตรวจจับการบุกรุกได้ เพื่อเปรียบเทียบกับผลการทดลองที่ได้ในแต่ละรอบขั้นตอนดังรูปที่ 4.2



รูปที่ 4.2 ขั้นตอนการหาค่าประสิทธิภาพ

ตัวอย่างคำสั่งการใช้โปรแกรม TCPReplay-3.4.0 เพื่ออ่านไฟล์ข้อมูลจำลองการบุกรุกที่ได้จากการใช้โปรแกรม Zenmap4.98 ชื่อ Intrusion.pcap ที่อัตราความเร็วปกติ 0.127 Mbps

 # tcpreplay --intf1=eth0 Intrusion.pcap

ตัวอย่างคำสั่งการใช้โปรแกรม TCPReplay-3.4.0 โดยทำการเปลี่ยนอัตราความเร็วในแต่ละรอบตามในเงื่อนไข *--mbps=อัตราความเร็ว* เพื่อจำลองการบุกรุกที่ความเร็วในแต่ละรอบจาก 100 Mbps, 200 Mbps, 300 Mbps, 400 Mbps, 500 Mbps, 600 Mbps, 700 Mbps, 800 Mbps, 900 Mbps และ 1 Gbps

ตัวอย่างข้างล่างนี้เป็นการกำหนดความเร็วในการส่งข้อมูลที่อัตราความเร็ว 100 Mbps

 # tcpreplay --mbps=100.0 --intf1=eth0 Intrusion.pcap

4.2 การทดสอบปรับแต่งกฎให้เหมาะสมตามความเสี่ยงและสภาวะแวดล้อม

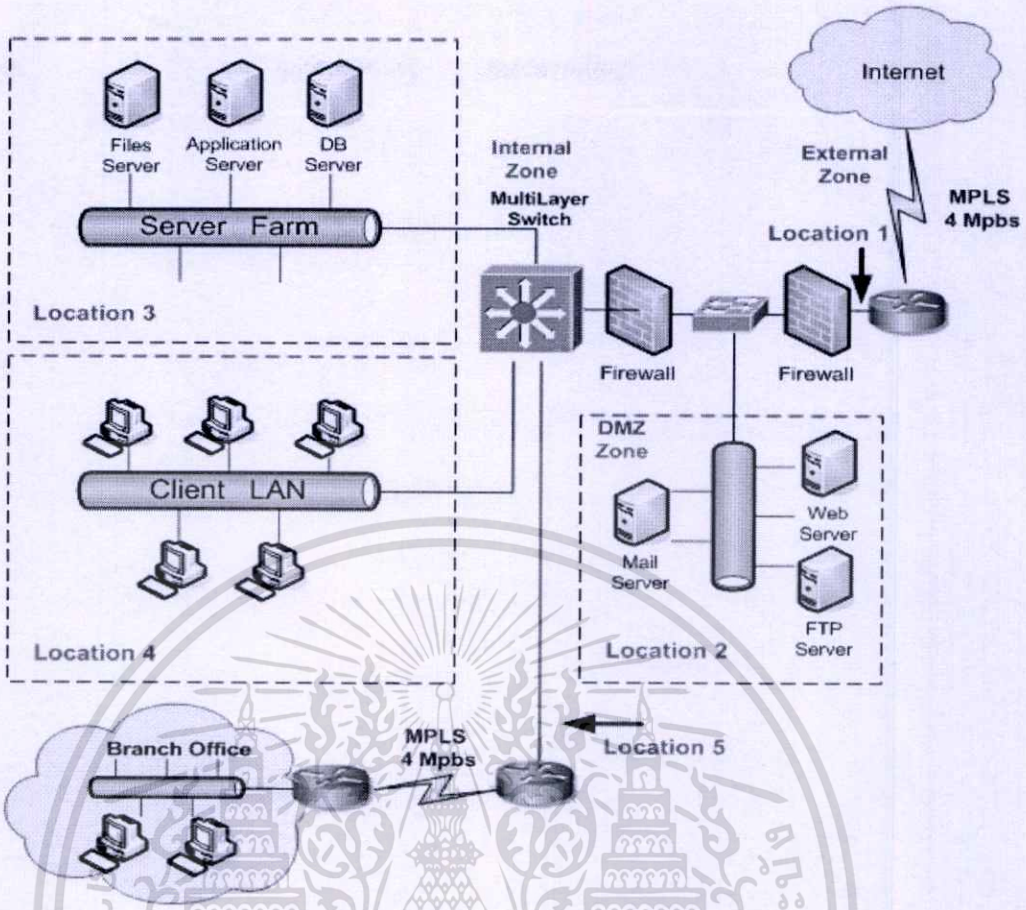
การทดสอบปรับแต่งกฎให้เหมาะสมตามความเสี่ยงและสภาวะแวดล้อมระบบตรวจจับการบุกรุก ซึ่งโดยส่วนใหญ่แล้วการแจ้งเตือนที่เกิดขึ้นมักจะไม่ได้เกิดจากการตรวจพบการบุกรุก

ที่แท้จริงซึ่งเราเรียกกรณีนี้ว่า False Positive ซึ่งวิธีการลดอัตราการเกิด False Positive ก็คือการปรับตั้งให้การทำงานของ NIDS มีความไวในการตรวจจับน้อยลง แต่ผลกระทบที่ตามมาคือบางครั้งหากมีการบุกรุกเครือข่ายเกิดขึ้นจริง NIDS ก็อาจจะตรวจไม่พบก็ได้ ซึ่งในกรณีนี้จะเรียกว่า False Negative และโดยทั่วไประบบตรวจจับระบบการบุกรุกระบบเครือข่ายที่ไม่ได้ทำการปรับแต่งกฎและสัญลักษณ์นั้นการแจ้งเตือนที่เกี่ยวข้องกับการบุกรุกจริง ๆ จะมีเพียง 10% เท่านั้นที่ แต่ส่วนที่เหลือ 90 % นั้นจะเป็นการแจ้งเตือนที่ไม่ได้เกิดการโจมตีขึ้นจริง (False alarms) จึงจำเป็นในการปรับแต่งกฎและสัญลักษณ์ให้มีความเหมาะสมเพื่อให้ NIDS มีประสิทธิภาพในการตรวจจับการบุกรุกมากยิ่งขึ้น

โดยทำการทดสอบเพื่อแสดงให้เห็นผลของการปรับแต่งกฎให้เหมาะสมตามสภาพแวดล้อม จึงได้ทำการดักจับข้อมูลการใช้งานจริงในบริษัทเอกชนแห่งหนึ่งเป็นเวลา 4 ชั่วโมงในบริเวณเครือข่ายกลุ่มเซิร์ฟเวอร์ (Server farm) ดังแสดงในรูปที่ 4.3 ซึ่งสามารถตรวจวัดจำนวนของทราฟฟิกได้ 1,000,124 แพ็คเก็ต ในอัตราค่าเฉลี่ยความเร็วของการส่งข้อมูล (Average data transmission rate) 0.191 Mbit/sec โดยนำแพ็คเก็ตข้อมูลที่ดักจับมาได้นั้นมาทดสอบในแบบจำลองหาประสิทธิภาพโดยใช้โปรแกรม TCPReplay-3.4.0 ทำการส่งแพ็คเก็ตข้อมูลผ่านมายังตัวตรวจจับการบุกรุกซึ่งยังไม่ได้ปรับแต่งกฎให้เหมาะสมตามสภาพแวดล้อมและทำการส่งแพ็คเก็ตข้อมูลเดิมอีกรอบโดยใช้กฎที่ปรับแต่งให้เหมาะสมตามสภาพแวดล้อมดังแสดงในตารางที่ 4.3 โดยใช้หลักเกณฑ์การปรับลดบางกฎที่ไม่มีการให้บริการในระบบเครือข่ายบริเวณเครือข่ายกลุ่มเซิร์ฟเวอร์ (Server Farm) รวมทั้งลดจำนวนกฎในส่วนของรูปแบบการตรวจจับการบุกรุกแบบต่าง ๆ ซึ่งไม่มีการให้บริการลงด้วยเช่นกัน

ส่วนการปรับลดกฎ bad-traffic.rules คือตรวจจับรูปแบบของแพ็คเก็ตที่ไม่เคยมีการพบเห็นมาก่อนในระบบเครือข่ายต่าง ๆ และปรับลดกฎ icmp-info.rules คือการกระทำที่น่าสงสัยว่าจะเข้าข่ายการโจมตีโดยใช้โปรโตคอล ICMP เป็นเครื่องมือในการสแกน ซึ่งมีผลต่อการลดจำนวนการแจ้งเตือนการบุกรุกที่ไม่ได้เกิดการบุกรุกที่แท้จริง (False Positive) แต่อย่างไรก็ตามอาจทำให้ลดความสามารถในการตรวจจับการบุกรุกในรูปแบบดังกล่าวลงเช่นเดียวกัน ซึ่งหากเกิดการบุกรุกรูปแบบดังกล่าวอาจทำให้ไม่สามารถตรวจพบได้ ในกรณีนี้เรียกว่า False Negative

การทดสอบดังกล่าวมีจุดประสงค์เพื่อเปรียบเทียบผลการตรวจจับการบุกรุกและแสดงให้เห็นอัตราการลดลงของการแจ้งเตือนที่ไม่ได้เกิดจากการตรวจพบการบุกรุกที่แท้จริงที่เรียกว่า False Positive รวมทั้งการใช้กฎและสัญลักษณ์ที่มีจำนวนลดลงอีกด้วย



รูปที่ 4.3 ระบบเครือข่ายคอมพิวเตอร์ที่นำทราฟฟิกมาวิเคราะห์

ตารางที่ 4.3 ค่าการปรับแต่งกฎให้เหมาะสมตามสภาพแวดล้อม

ชื่อกฎ	ไม่ปรับแต่งกฎ	ปรับแต่งกฎ
attack-responses.rules	✓	✓
backdoor.rules	✓	✓
bad-traffic.rules	✓	-
chat.rules	✓	-
content-replace	-	-
ddos.rules	✓	✓
deleted.rules	✓	-
dns.rules	✓	✓
dos.rules	✓	✓
experimental.rules	-	-

ตารางที่ 4.3 ค่าการปรับแต่งกฎให้เหมาะสมตามสภาพแวดล้อม(ต่อ)

ชื่อกฎ	ไม่ปรับแต่งกฎ	ปรับแต่งกฎ
exploit.rules	✓	✓
finger.rules	✓	-
ftp.rules	✓	✓
icmp-info.rules	✓	-
icmp.rules	✓	✓
imap.rules	✓	✓
info.rules	✓	-
local.rules	✓	-
misc.rules	✓	✓
multimedia.rules	✓	-
mysql.rules	✓	✓
netbios.rules	✓	✓
nntp.rules	✓	-
oracle.rules	✓	✓
other-ids.rules	✓	✓
p2p.rules	✓	-
policy.rules	✓	-
pop2.rules	✓	-
pop3.rules	✓	-
porn.rules	✓	-
rpc.rules	✓	✓
rservices.rules	✓	✓
scan.rules	✓	✓
shellcode.rules	✓	✓
smtp.rules	✓	✓
specific-threats	✓	✓

ตารางที่ 4.3 ค่าการปรับแต่งกฎให้เหมาะสมตามสภาพแวดล้อม(ต่อ)

ชื่อกฎ	ไม่ปรับแต่งกฎ	ปรับแต่งกฎ
snmp.rules	✓	✓
spyware-put	✓	✓
sql.rules	✓	✓
telnet.rules	✓	✓
tftp.rules	✓	✓
virus.rules	✓	✓
voip	✓	-
web-attacks.rules	✓	-
web-cgi.rules	✓	-
web-client.rules	✓	-
web-coldfusion.rules	✓	-
web-frontpage.rules	✓	-
web-iis.rules	✓	✓
web-misc.rules	✓	✓
web-php.rules	✓	-
X11.rules	✓	-

บทที่ 5
ผลการทดลอง

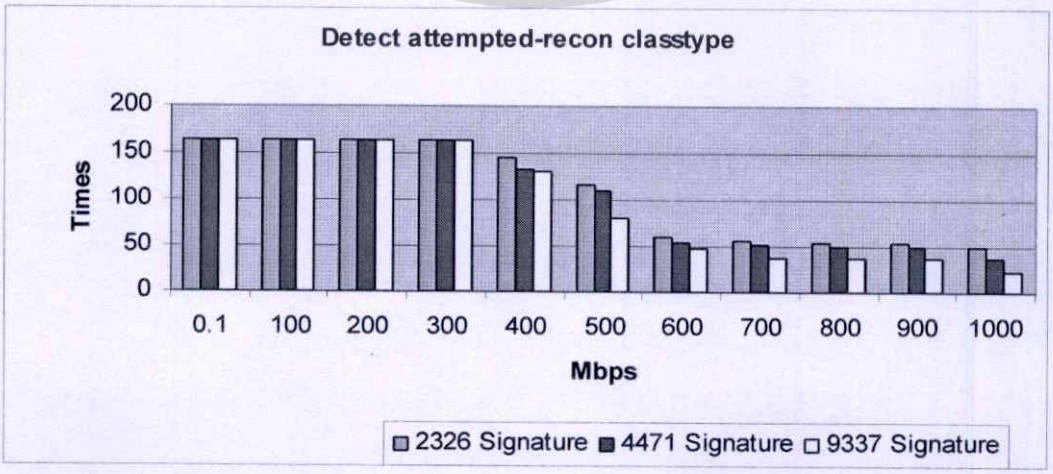
จากผลการทดลอง ดังรูปที่ 5.1 การบุกรุกแบบ Attempted-recon ซึ่งเป็นการพยายามเจาะระบบจำนวน 164 ครั้งเป็นโปรโตคอลชนิด TCP และ รูปที่ 5.2 การบุกรุกโดย Executable code จำนวน 351 ครั้งเป็นโปรโตคอลชนิด UDP และรูปที่ 5.3 การบุกรุกแบบทั่ว ๆ ไปจำนวน 328 ครั้ง ซึ่งเป็นโปรโตคอลชนิด ICMP และรูปที่ 5.4 เป็นการบุกรุกรวมทั้งหมด จะเห็นได้ว่าประสิทธิภาพการตรวจพบการบุกรุกของ Snort จะเริ่มลดลงเมื่อความเร็วของการส่งข้อมูลอยู่ในระดับ 300 Mbps ซึ่งถือได้ว่าเป็นจุดอิ่มตัวของโปรแกรม Snort ที่ทำงานบนเครื่องคอมพิวเตอร์ที่มีรายละเอียดดังที่แสดงประกอบไว้ในบทที่ 4 ซึ่งจำนวนครั้งของการตรวจพบการบุกรุกนั้นจะมีความสัมพันธ์กับเปอร์เซ็นต์การใช้งานซีพียู (CPU usage) ดังแสดงในรูปที่ 5.5 โดยพบว่าที่ความเร็วในการสื่อสารข้อมูลในระดับ 300 Mbps ซึ่งเป็นจุดอิ่มตัวนั้น เป็นตำแหน่งที่มีเปอร์เซ็นต์การใช้งานซีพียูประมาณ 40% ซึ่งส่งผลให้เริ่มมีการทิ้งแพ็คเก็ตข้อมูลก่อนเวลาอันสมควร หลังจากนั้นประสิทธิภาพการตรวจจับการบุกรุกก็จะยิ่งลดลงเรื่อย ๆ เมื่อความเร็วในการส่งผ่านข้อมูลในระบบเครือข่ายเพิ่มสูงขึ้นมาก และผลการทดลองได้แสดงให้เห็นว่าเมื่อกำหนดให้โปรแกรม Snort ตรวจจับการบุกรุกโดยการตรวจสอบด้วยกฎและสัญลักษณ์จำนวนมากนั้นจะมีประสิทธิภาพการตรวจจับการบุกรุกที่ด้อยกว่าการทำงานของโปรแกรมเดียวกันในสภาพแวดล้อมเดียวกันแต่กำหนดให้มีการใช้กฎและสัญลักษณ์ที่มีจำนวนน้อยกว่า โดยผลการทดลองทั้งหมดนั้นสอดคล้องกันทั้งสำหรับการตรวจจับการบุกรุกแบบแยกตามชนิดของการโจมตีดังแสดงในรูปที่ 5.1 ถึง 5.3 และผลสรุปการตรวจจับการบุกรุกทั้งหมดดังแสดงในรูปที่ 5.4 ซึ่งผลการทดลองที่ได้สอดคล้องกับสมมุติฐานในการทดลองว่าจำนวนของกฎและสัญลักษณ์ของโปรแกรม Snort น่าจะมีผลกระทบกับประสิทธิภาพโดยรวมของการตรวจจับการบุกรุกเมื่อความเร็วในการส่งผ่านข้อมูลในระบบเครือข่ายความเร็วสูงที่มีปริมาณทราฟฟิกมาก ๆ ส่วนการทดสอบปรับแต่งกฎให้เหมาะสมตามความเสี่ยงและสภาวะแวดล้อมโดยใช้แบบจำลองการหาค่าประสิทธิภาพการตรวจจับการบุกรุกมาช่วยนั้น ผลการทดสอบทำให้เห็นว่าการใช้ตัวตรวจจับการบุกรุกซึ่งยังไม่ได้ปรับแต่งกฎให้เหมาะสมตามสภาวะแวดล้อมพบว่ามีรายงานการตรวจจับการบุกรุกซึ่งไม่ได้เกิดจากการบุกรุกที่แท้จริง (False Positive) จำนวนมากเมื่อเปรียบเทียบกับการใช้กฎที่ได้รับการปรับแต่งให้เหมาะสม ซึ่งสามารถลดจำนวนครั้งของการแจ้งเตือนจาก 3,539 ครั้ง เหลือเพียง 795 ครั้งเท่านั้นดังแสดงใน ตารางที่ 5.6 เป็นการลดรายงานการแจ้งเตือนการตรวจจับการบุกรุกซึ่งไม่ได้เกิดจากการบุกรุกที่แท้จริง (False Positive) ถึง 2,744 ครั้ง คิดเป็น 77.5% ของการแจ้งเตือนทั้งหมดและสามารถลดจำนวนการใช้กฎเพื่อตรวจสอบการบุกรุกจากเดิม 9,337 สัญลักษณ์เหลือเพียง 7,453 สัญลักษณ์ ซึ่งส่งผลดีต่อประสิทธิภาพโดยรวมของระบบตรวจจับการบุกรุกอีกด้วย

5.1 การทดสอบหาค่าประสิทธิภาพการตรวจจับการบุกรุก

5.1.1 จำนวนการตรวจจับการบุกรุกแบบ Attempted-recon

ตารางที่ 5.1 จำนวนการตรวจจับการบุกรุกแบบ Attempted-recon ได้ เมื่อเพิ่มอัตราความเร็วในการส่งข้อมูลและจำนวนสัญลักษณ์

อัตราความเร็วของการส่งผ่านข้อมูล	2,326 สัญลักษณ์	4,471 สัญลักษณ์	9,337 สัญลักษณ์
	จำนวนการตรวจจับ	จำนวนการตรวจจับ	จำนวนการตรวจจับ
0.1 Mbps	164	164	164
100 Mbps	164	164	164
200 Mbps	164	164	164
300 Mbps	164	164	164
400 Mbps	144	133	132
500 Mbps	117	111	79
600 Mbps	56	48	46
700 Mbps	56	51	36
800 Mbps	54	36	34
900 Mbps	60	50	36
1000 Mbps	49	50	22

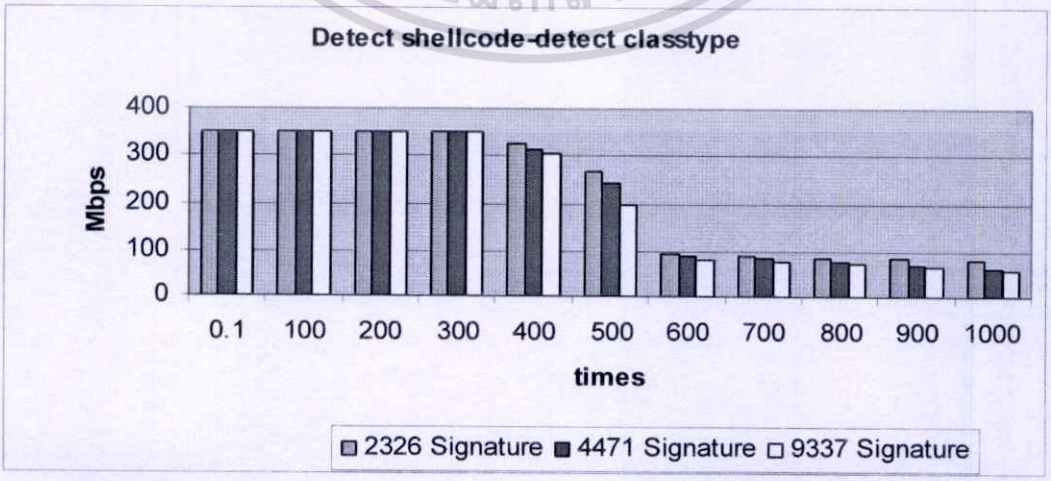


รูปที่ 5.1 ประสิทธิภาพการตรวจจับ Attempted-recon

5.1.2 จำนวนการตรวจจับการบุกรุกแบบ Shellcode-detect

ตารางที่ 5.2 จำนวนการตรวจจับการบุกรุกแบบ Shellcode-detect ได้ เมื่อเพิ่มอัตราความเร็วในการส่งข้อมูลและจำนวนสัญลักษณ์

อัตราความเร็วของการส่งผ่านข้อมูล	2,326 สัญลักษณ์	4,471 สัญลักษณ์	9,337 สัญลักษณ์
	จำนวนการตรวจจับ	จำนวนการตรวจจับ	จำนวนการตรวจจับ
0.1 Mbps	351	351	351
100 Mbps	351	351	351
200 Mbps	351	351	351
300 Mbps	351	351	351
400 Mbps	326	313	306
500 Mbps	269	245	200
600 Mbps	95	90	81
700 Mbps	92	85	77
800 Mbps	87	77	75
900 Mbps	86	71	65
1000 Mbps	84	62	57

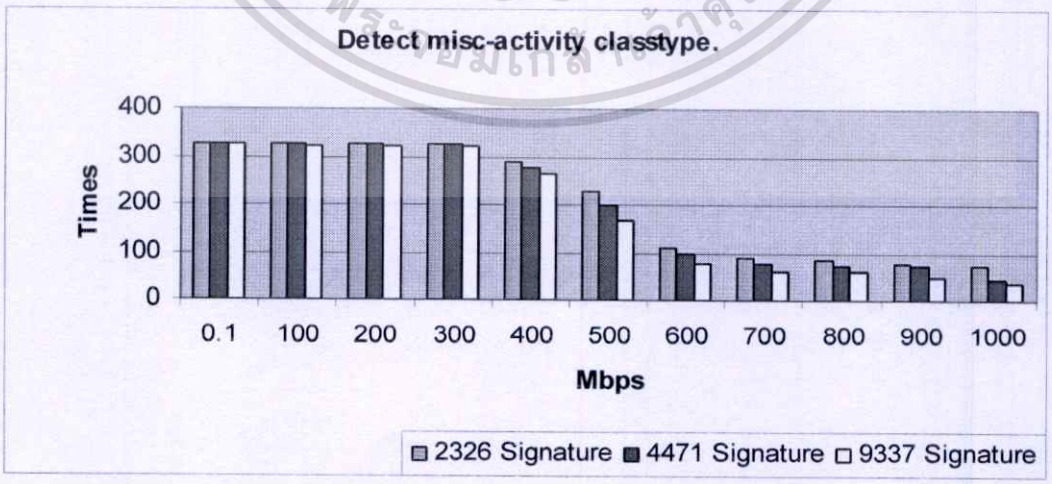


รูปที่ 5.2 ประสิทธิภาพการตรวจจับ Shellcode-detect

5.1.3 จำนวนการตรวจจับการบุกรุกแบบ Misc-activity

ตารางที่ 5.3 จำนวนการตรวจจับการบุกรุกรวม Misc-activity ได้เมื่อเพิ่มอัตราความเร็วในการส่งข้อมูลและจำนวนสัญลักษณ์

อัตราความเร็วของการส่งผ่านข้อมูล	2,326 สัญลักษณ์ จำนวนการตรวจจับ	4,471 สัญลักษณ์ จำนวนการตรวจจับ	9,337 สัญลักษณ์ จำนวนการตรวจจับ
0.1 Mbps	328	328	328
100 Mbps	326	326	324
200 Mbps	326	326	324
300 Mbps	326	326	324
400 Mbps	290	278	266
500 Mbps	230	200	167
600 Mbps	109	97	79
700 Mbps	90	78	60
800 Mbps	85	75	60
900 Mbps	79	72	48
1000 Mbps	72	45	36

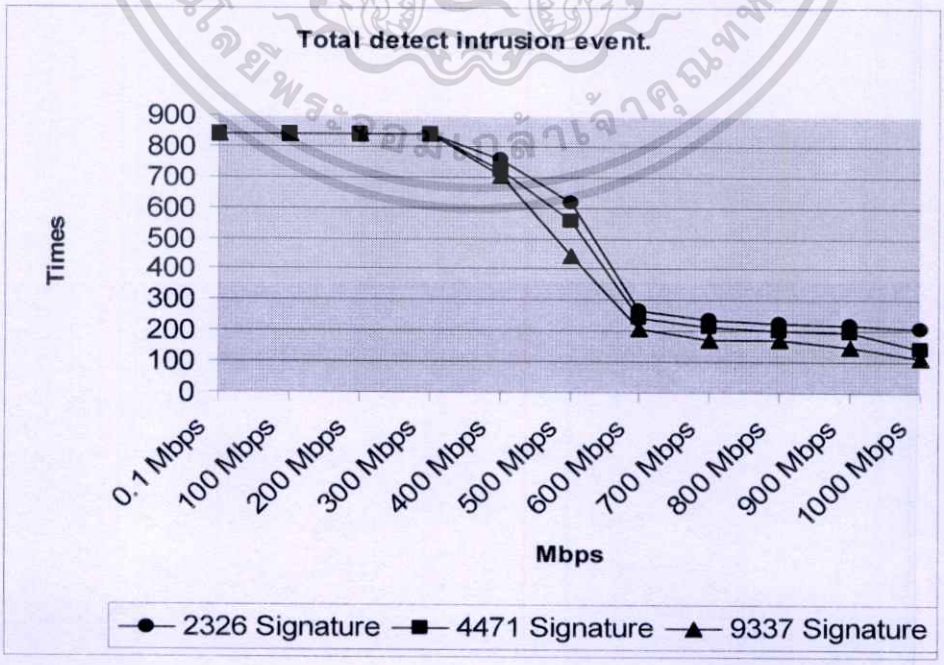


รูปที่ 5.3 ประสิทธิภาพการตรวจจับ Misc-activity

5.1.4 จำนวนการตรวจจับการบุกรุกรวม

ตารางที่ 5.4 จำนวนการตรวจจับการบุกรุกรวมได้เมื่อเพิ่มอัตราความเร็วในการส่งข้อมูลและจำนวนสัญลักษณ์

อัตราความเร็วของการส่งผ่านข้อมูล	จำนวนสัญลักษณ์		
	2326 สัญลักษณ์	4471 สัญลักษณ์	9337 สัญลักษณ์
0.1 Mbps	843	843	843
100 Mbps	841	841	839
200 Mbps	841	841	839
300 Mbps	841	841	839
400 Mbps	760	724	702
500 Mbps	616	556	446
600 Mbps	264	241	206
700 Mbps	238	214	173
800 Mbps	226	202	171
900 Mbps	219	193	149
1000 Mbps	205	143	115

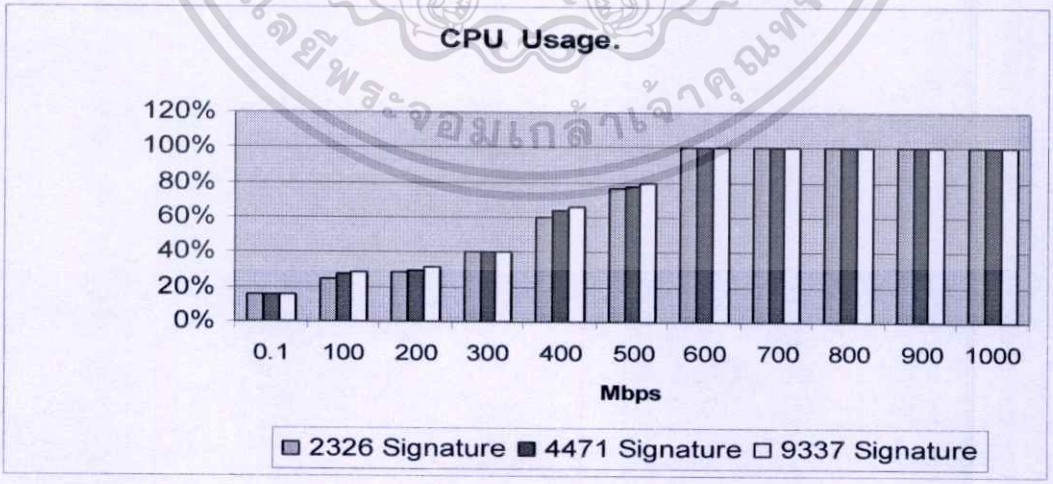


รูปที่ 5.4 ประสิทธิภาพการตรวจจับทั้งหมด

5.1.5 อัตราการใช้งานซีพียู NIDS

ตารางที่ 5.5 อัตราการใช้งานซีพียูบนเครื่องที่ให้บริการตรวจจับการบุกรุกเมื่อเพิ่มอัตราความเร็วในการส่งข้อมูล

อัตราความเร็วของการส่งผ่านข้อมูล	จำนวนสัญลักษณ์		
	2326 สัญลักษณ์	4471 สัญลักษณ์	9337 สัญลักษณ์
0.1 Mbps	16%	16%	16%
100 Mbps	25%	28%	29%
200 Mbps	29%	30%	31%
300 Mbps	40%	40%	40%
400 Mbps	60%	64%	66%
500 Mbps	77%	78%	80%
600 Mbps	100%	100%	100%
700 Mbps	100%	100%	100%
800 Mbps	100%	100%	100%
900 Mbps	100%	100%	100%
1000 Mbps	100%	100%	100%



รูปที่ 5.5 เปอร์เซ็นต์การใช้งานซีพียูระบบการตรวจจับการบุกรุก

5.2 การทดสอบปรับแต่งกฎให้เหมาะสมตามความเสี่ยงและสภาวะแวดล้อม

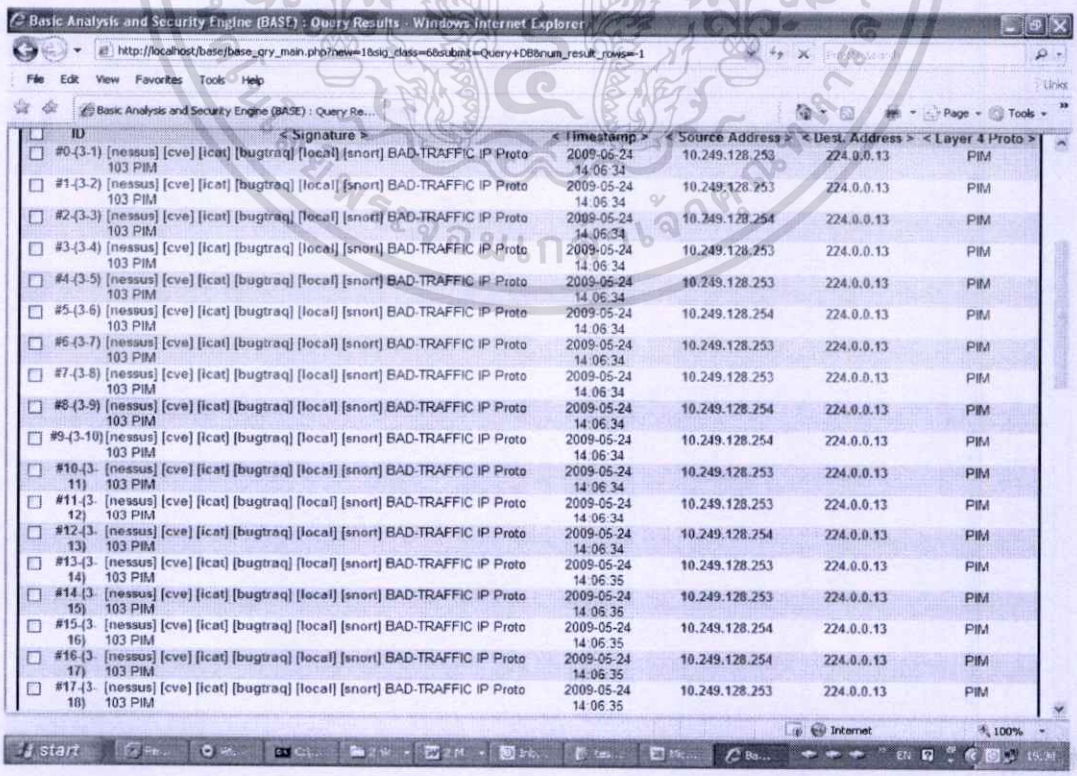
ผลการทดสอบข้อมูลการใช้งานจริงในบริษัทเอกชนแห่งหนึ่ง โดยดักจับแพ็คเก็ตบริเวณเครือข่ายกลุ่มเซิร์ฟเวอร์ (Server Farm) โดยนำแพ็คเก็ตข้อมูลที่ดักจับมาได้นั้นมาทดสอบในแบบจำลองการหาประสิทธิภาพโดยใช้โปรแกรม TCPReplay-3.4.0 แล้วใช้ตัวตรวจจับการบุกรุกซึ่งยังไม่ได้ปรับแต่งกฎให้เหมาะสมตามสภาพแวดล้อม พบว่ามีรายงานการตรวจจับการบุกรุกซึ่งไม่ได้เกิดจากการบุกรุกที่แท้จริง (False Positive) อยู่จำนวนมากเช่น

BAD-TRAFFIC IP Proto 103 PIM หมายถึงเหตุการณ์ที่เกิดขึ้นเมื่อตรวจพบแพ็คเก็ตที่มีโปรโตคอลไม่ปกติที่ส่งจากเราท์เตอร์ ซึ่งจริง ๆ แล้วเป็นการทำงานปกติของ Multilayer switch ที่ใช้งาน PIM (Protocol Independent Multicast) IP 224.0.0.13 ดังแสดงในรูปที่ 5.6

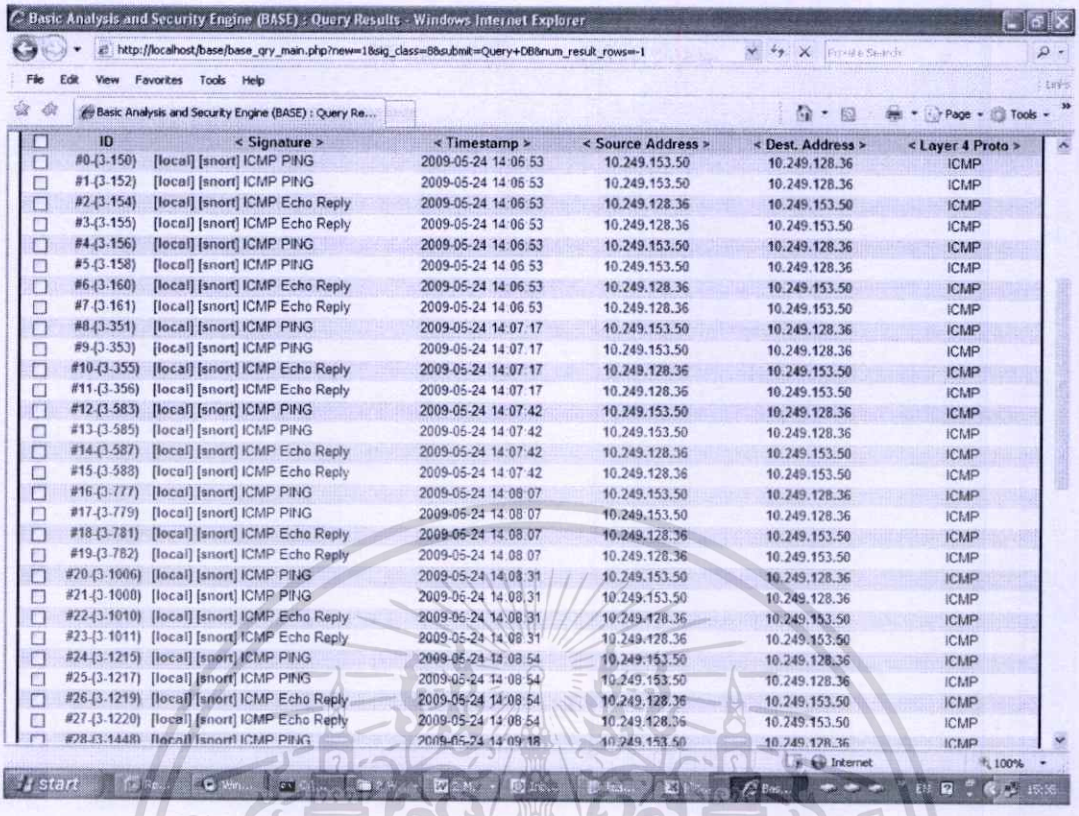
ICMP PING หมายถึงมีการสร้าง ICMP Echo Request ภายในระบบเครือข่าย

ICMP Echo Reply มีการตอบกลับโดยใช้ ICMP Echo Reply

ซึ่งเป็นการทำงานปกติของเครื่องลูกข่ายในระบบเครือข่ายคอมพิวเตอร์ ดังรายงานรูปที่ 5.7 ดังนั้นจึงควรปรับแต่งกฎให้ตัว NIDS ให้มีความไวน้อยลง โดยไม่ให้แสดงรายงานการแจ้งเตือนการบุกรุกเหล่านี้



รูปที่ 5.6 รายงานการแจ้งเตือน BAD-TRAFFIC IP Proto 103 PIM



รูปที่ 5.7 รายงานการแจ้งเตือน ICMP PING & ICMP Echo Reply

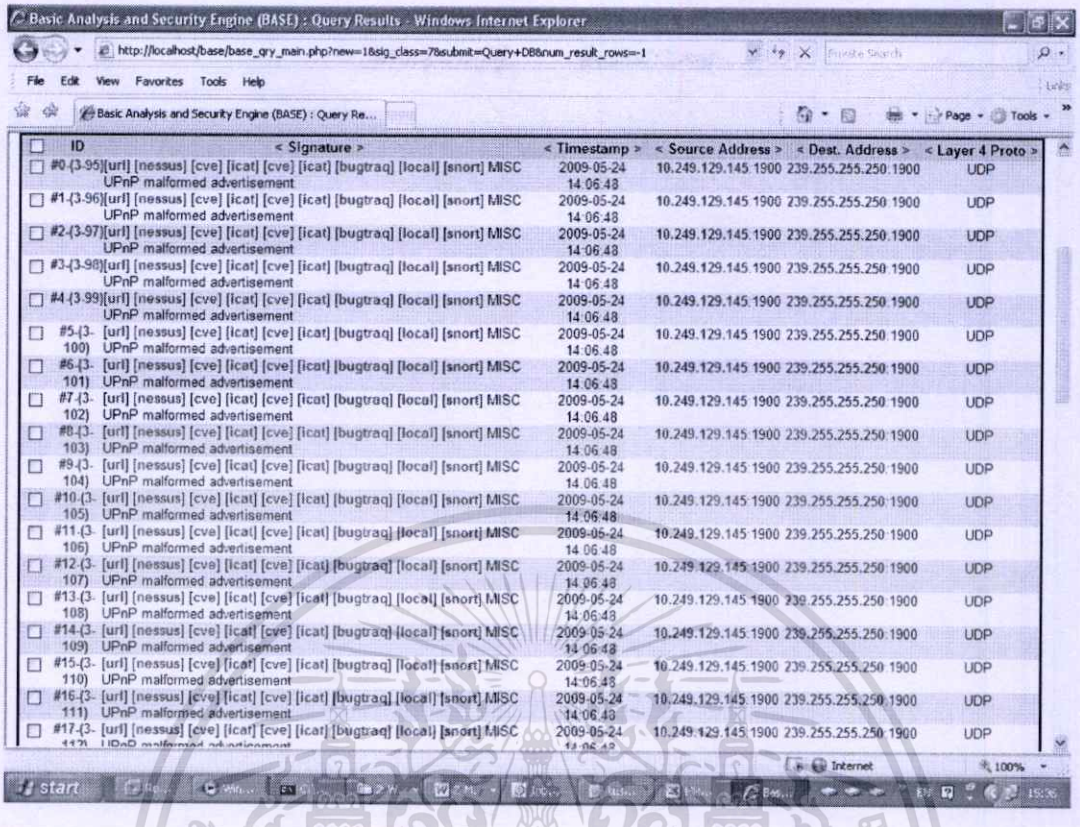
จากนั้นมาพิจารณาการแจ้งเตือนที่บ่งชี้ถึงการบุกรุกอื่น ๆ เช่น

MISC UPnP malformed advertisement หมายถึงการพยายามใช้สิทธิผู้ดูแลระบบเข้าสู้ระบบหรือทำให้ระบบหยุดการให้บริการ หากการโจมตีทำได้สำเร็จอาจทำให้เครื่องเซิร์ฟเวอร์ไม่สามารถให้บริการต่อไปได้หรืออาจจะเข้าไปแก้ไข source code ได้ตามอำเภอใจโดยใช้สิทธิผู้ดูแลระบบ ดังแสดงในรูปที่ 5.8

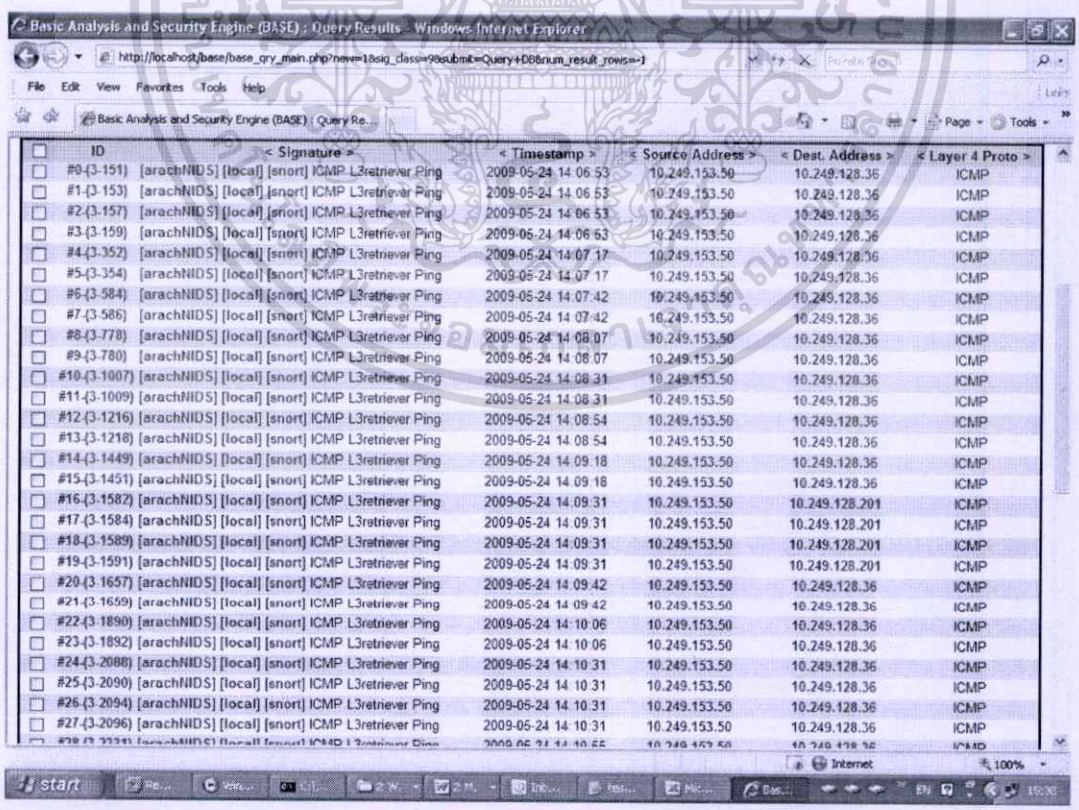
ICMP L3retriever Ping การรวบรวมข้อมูล โดยใช้ ICMP echo request เพื่อให้ทราบถึงสถานะการทำงานของเครื่องปลายทาง ดังแสดงในรูปที่ 5.9

ICMP PING NMAP มีการใช้โปรแกรม NMAP สแกนในระบบซึ่งเป็นพฤติกรรมที่มีแนวโน้มในที่เป็นจะทำอันตรายต่อระบบ ดังแสดงในรูปที่ 5.10

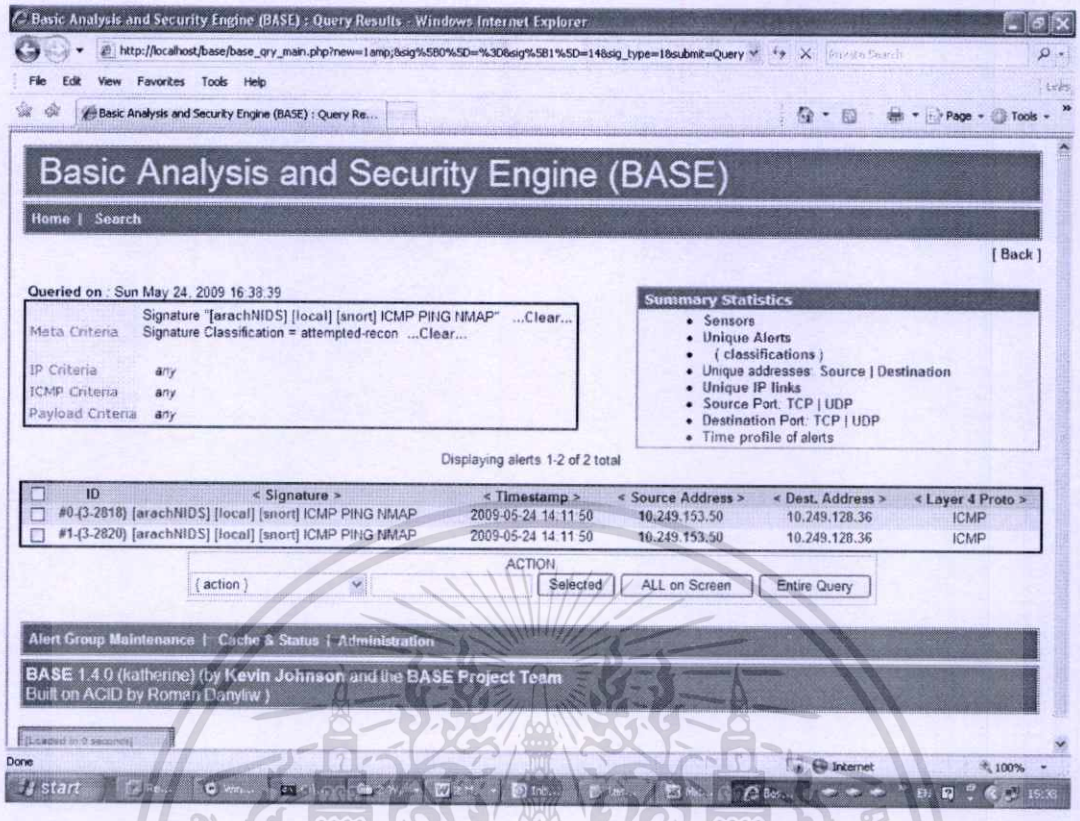
WEB-IIS view source via translate header การรวบรวมข้อมูลโดยการโจมตีโดยเปิดเผย source code ของไฟล์ซึ่งไม่ใช่การใช้งานปกติ ดังแสดงในรูปที่ 5.11



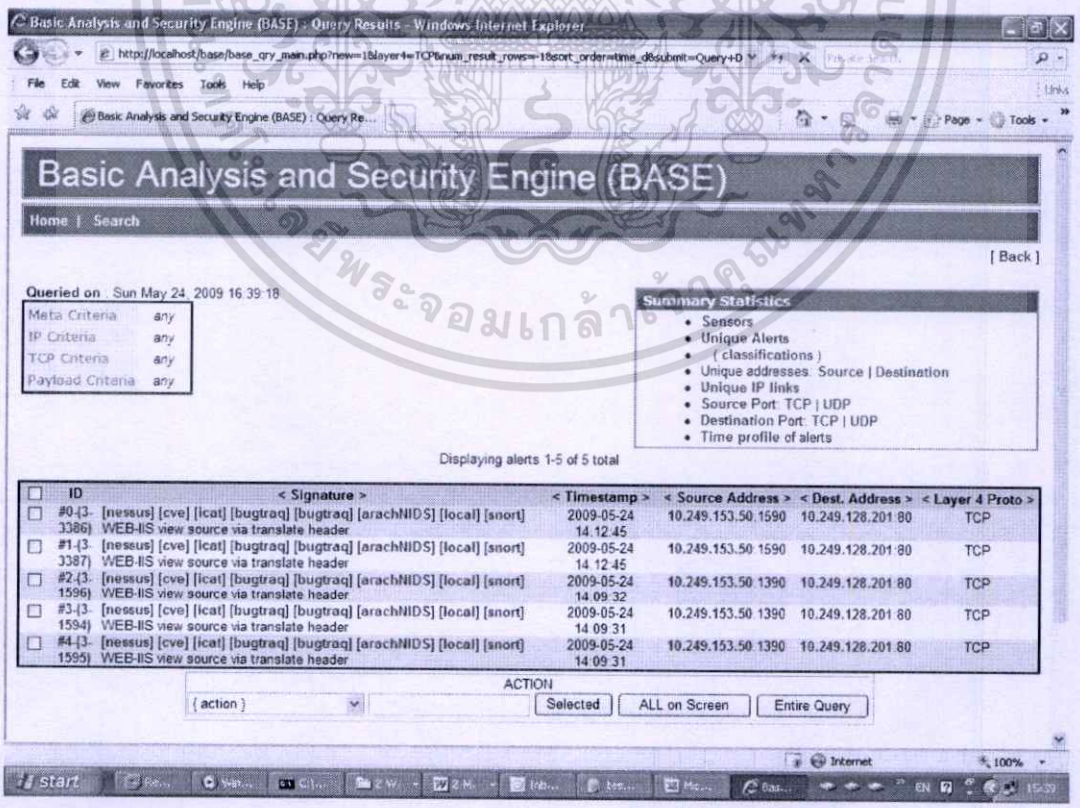
รูปที่ 5.8 รายงานการแจ้งเตือน MISC UPnP malformed advertisement



รูปที่ 5.9 รายงานการแจ้งเตือน ICMP L3retriever Ping



รูปที่ 5.10 รายงานการแจ้งเตือน ICMP PING NMAP



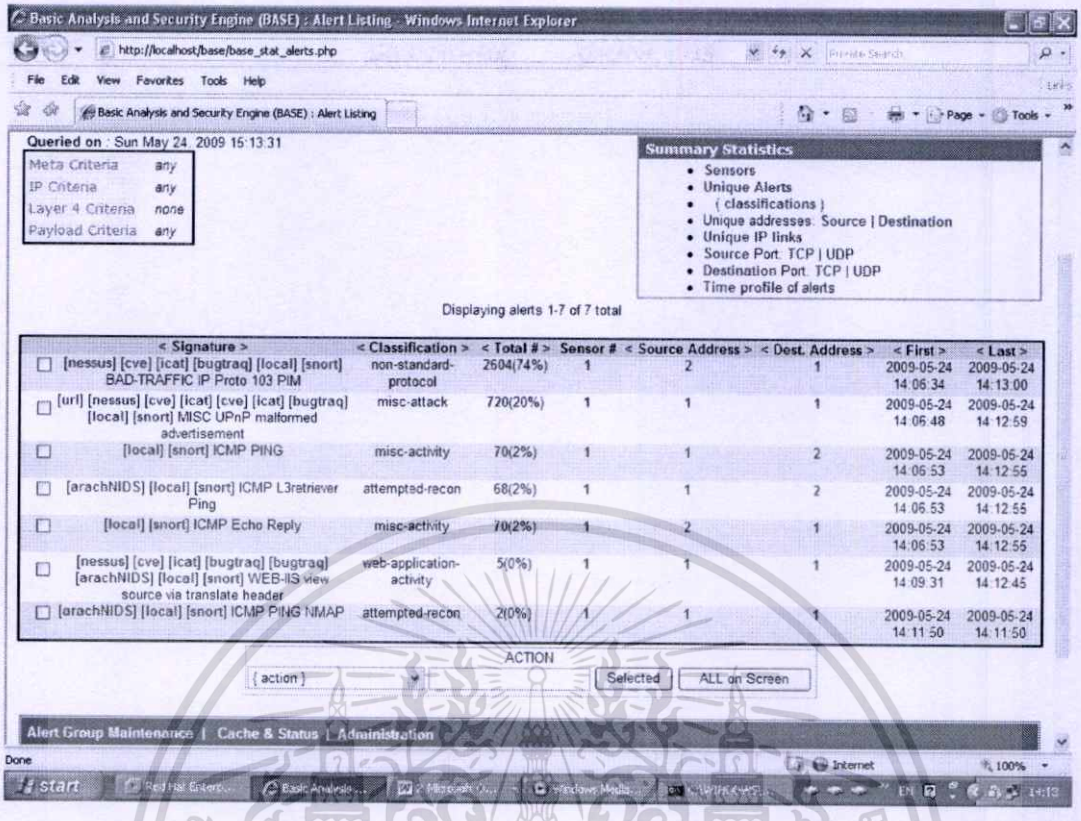
รูปที่ 5.11 รายงานการแจ้งเตือน WEB-IIS view source via translate header

ผลการทดสอบเพื่อเปรียบเทียบจำนวนการแจ้งเตือน หลังจากนำแพ็คเก็ตข้อมูลที่ทำการดักจับมาจากการใช้งานในระบบเครือข่ายบริเวณเครือข่ายกลุ่มเซิร์ฟเวอร์ (Server Farm) นั้นมาทดสอบในแบบจำลองการหาประสิทธิภาพโดยใช้โปรแกรม TCPReplay-3.4.0 ทำการส่งแพ็คเก็ตข้อมูลผ่านมายังตัวตรวจจับการบุกรุก ซึ่งยังไม่ได้ปรับแต่งกฎให้เหมาะสมตามสภาพแวดล้อมและทำการส่งแพ็คเก็ตข้อมูลเดิมอีกรอบ โดยใช้ตัวตรวจจับการบุกรุกที่ปรับแต่งกฎให้เหมาะสมตามสภาพแวดล้อม เพื่อเปรียบเทียบจำนวนการแจ้งเตือน

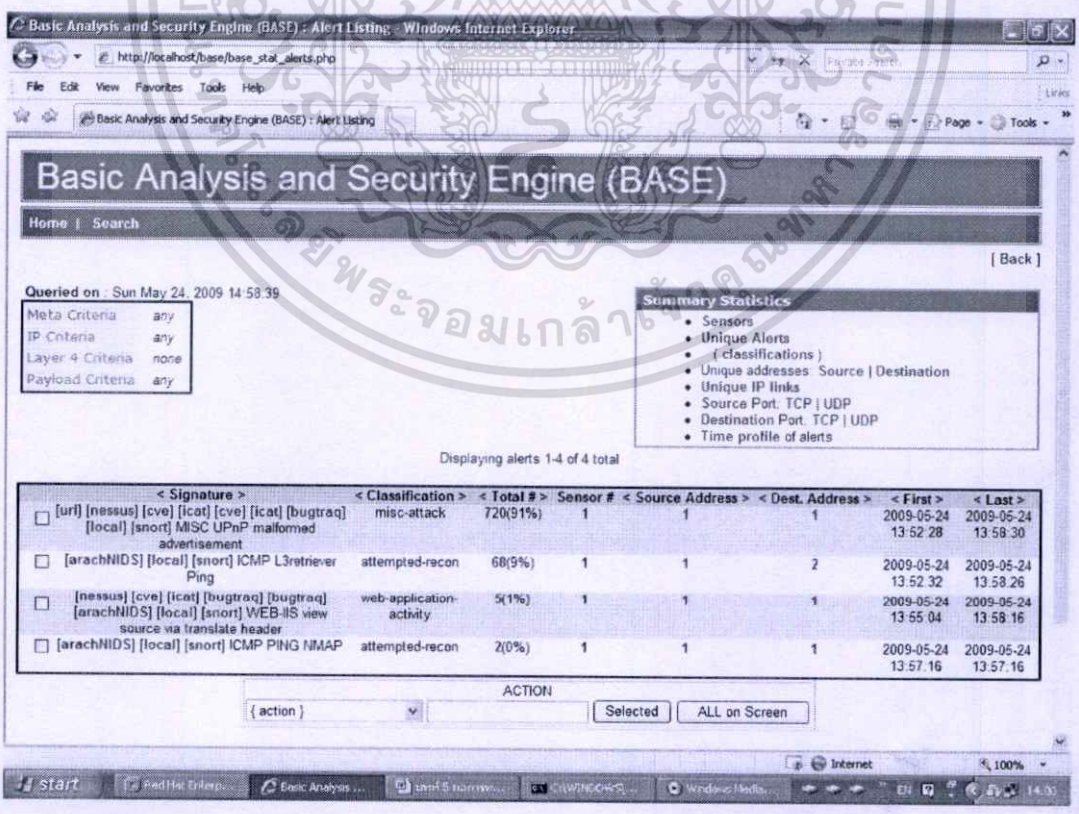
รายงานการแจ้งเตือนรวมเมื่อไม่ปรับแต่งกฎดังแสดงในรูปที่ 5.12 และรายงานการแจ้งเตือนรวมเมื่อทำการปรับแต่งกฎให้เหมาะสมตามสภาพแวดล้อมดังแสดงในรูปที่ 5.13 ซึ่งแสดงให้เห็นว่าสามารถลดจำนวนครั้งของการแจ้งเตือนจาก 3,539 ครั้งเหลือเพียง 795 ครั้ง ดังแสดงในตารางที่ 5.6 และเป็นการลดรายงานการตรวจจับการบุกรุกซึ่งไม่ได้เกิดจากการบุกรุกที่แท้จริง (False Positive) จาก BAD-TRAFFIC IP Proto 103 PIM จำนวน 2604 ครั้ง ICMP PING จำนวน 70 ครั้ง และ ICMP Echo Reply จำนวน 70 ครั้ง รวมเป็น 2,744 ครั้ง คิดเป็น 77.5% ของการแจ้งเตือนทั้งหมด อีกทั้งยังสามารถลดจำนวนกฎและสัญลักษณ์ที่ใช้ในการตรวจสอบจากเดิม 9,337 สัญลักษณ์เหลือเพียง 7,453 สัญลักษณ์ ซึ่งสามารถเพิ่มประสิทธิภาพโดยรวมของระบบตรวจจับการบุกรุกอีกด้วย

ตารางที่ 5.6 เปรียบเทียบจำนวนการแจ้งเตือนหลังปรับแต่งกฎให้เหมาะสมตามสภาพแวดล้อม

จำแนกตามชนิดการแจ้งเตือน	สัญลักษณ์ที่มีการแจ้งเตือน	จำนวนการแจ้งเตือน	
		ไม่ปรับแต่งกฎ	ปรับแต่งกฎ
non-standard-protocol	BAD-TRAFFIC IP Proto 103 PIM	2,604	0
misc-attack	MISC UPnP malformed advertisement	720	720
misc-activity	ICMP PING	70	0
	ICMP Echo Reply	70	0
attempted-recon	ICMP L3retriever Ping	68	68
	ICMP PING NMAP	2	2
web-application-activity	WEB-IIS view source via translate header	5	5
	จำนวนการแจ้งเตือนรวม	3,539	795



รูปที่ 5.12 รายงานการแจ้งเตือนรวมเมื่อไม่ปรับแต่งกฎ



รูปที่ 5.13 รายงานการแจ้งเตือนรวมเมื่อปรับแต่งกฎ

บทที่ 6
สรุปผลการวิจัยและข้อเสนอแนะ

6.1 สรุปผลการวิจัย

การทดลองหาค่าประสิทธิภาพของการตรวจจับการบุกรุกจากงานวิจัยนี้ได้แสดงให้เห็นว่าปัจจัยของจำนวนกฎและสัญลักษณ์ที่ถูกใช้ในการตรวจสอบแพ็คเก็ตข้อมูลนั้นมีผลโดยตรงต่อประสิทธิภาพการตรวจจับการบุกรุก เมื่อความเร็วในการส่งผ่านข้อมูลเพิ่มสูงขึ้น โดยการทดลองได้แสดงให้เห็นว่าประสิทธิภาพของ NIDS จะเริ่มลดลงเมื่อความเร็วในการส่งผ่านข้อมูลเพิ่มสูงขึ้นจนถึงจุดอิ่มตัวที่ 300 Mbps ของการใช้งานทรัพยากรของโปรแกรม NIDS บนเครื่องคอมพิวเตอร์ที่ใช้ในการทดลอง เมื่อเปอร์เซ็นต์การใช้งาน CPU เกินกว่า 40% นั้นส่งผลให้เริ่มมีการทิ้งแพ็คเก็ตข้อมูลก่อนเวลาอันสมควร และจากจุดอิ่มตัวนี้เป็นต้นไปจึงจะสังเกตเห็นได้ว่าประสิทธิภาพของ NIDS ที่ใช้กฎและสัญลักษณ์ในจำนวนที่มากกว่านั้นจะเริ่มแสดงประสิทธิภาพที่แย่ลงเมื่อเทียบกับระบบ NIDS ที่กำหนดใช้กฎและสัญลักษณ์ในจำนวนที่น้อยกว่า และเมื่อความเร็วในการส่งผ่านข้อมูลยิ่งเพิ่มสูงขึ้นประสิทธิภาพของ NIDS ที่ต้องตรวจสอบกฎและสัญลักษณ์มากกว่านั้นก็จะตรวจจับการบุกรุกได้ลดลงต่ำกว่าอย่างชัดเจน ดังนั้นการติดตั้งกฎและสัญลักษณ์ในการตรวจจับการบุกรุกที่เหมาะสมสำหรับระบบเครือข่ายเป็นเรื่องที่มีนัยสำคัญ อีกการทดสอบหนึ่งคือการปรับแต่งกฎให้เหมาะสมตามความเสี่ยงและสภาวะแวดล้อมโดยใช้แบบจำลองการประเมินประสิทธิภาพการตรวจจับการบุกรุกมาช่วยในการทดสอบนั้น การใช้ตัวตรวจจับการบุกรุกที่ยังไม่ได้ปรับแต่งกฎให้เหมาะสมตามสภาพแวดล้อมจะพบว่ามีรายงานการตรวจจับการบุกรุกที่ไม่ได้เกิดจากการบุกรุกที่แท้จริง (False Positive) จำนวนมากเมื่อเปรียบเทียบกับกฎที่ได้รับการปรับแต่งให้เหมาะสม โดยการทดสอบปรับกฎให้เหมาะสมนั้นสามารถลดจำนวนครั้งของการแจ้งเตือนจาก 3,539 ครั้งเหลือเพียง 795 ครั้งเท่านั้น ซึ่งเป็นการแจ้งเตือนการตรวจจับการบุกรุกที่ไม่ได้เกิดจากการบุกรุกที่แท้จริง (False Positive) ถึง 2744 ครั้ง คิดเป็น 77.5% และสามารถลดจำนวนการใช้กฎเพื่อตรวจสอบจากเดิม 9,337 สัญลักษณ์เหลือเพียง 7,453 สัญลักษณ์ เท่านั้น

โดยงานวิจัยอื่นๆ ที่ได้พิจารณาปัจจัยของทรัพยากรฮาร์ดแวร์ที่แตกต่างกัน อาทิเช่น ความเร็วของ CPU ขนาดของหน่วยความจำ ขนาดของฮาร์ดดิสก์ ชนิดและความเร็วของการ์ดสื่อสาร ชนิดของฐานข้อมูล รวมทั้งปัจจัยของระบบปฏิบัติการ [1] , [2] ซึ่งงานวิจัยนี้ได้เสนอการประเมินประสิทธิภาพเพิ่มเติมโดยชี้ให้เห็นผลกระทบเนื่องจากการเพิ่มขึ้นของจำนวนกฎและสัญลักษณ์ที่ส่งผลต่อประสิทธิภาพของการตรวจจับการบุกรุกที่มีการส่งข้อมูลปริมาณมากผ่านบน

ระบบเครือข่ายที่มีความเร็วสูง รวมทั้งการเสนอแนะแนวทางการใช้กฎและสัญลักษณ์อย่างเหมาะสมอีกด้วย

6.2 ข้อเสนอแนะ

ถ้าหากกำหนดโดยใช้กฎและสัญลักษณ์ที่สอดคล้องกับสภาวะแวดล้อมในการทำงานของระบบเครือข่าย รวมทั้งกำหนดความเข้มงวดของการใช้กฎและสัญลักษณ์ตามความจำเป็นของสภาวะแวดล้อมนั้นๆ ก็จะสามารถลดจำนวนกฎและสัญลักษณ์และการแจ้งเตือนที่ไม่จำเป็น อีกทั้งสามารถเพิ่มประสิทธิภาพของการตรวจจับการบุกรุกได้ โดยเฉพาะอย่างยิ่งในระบบเครือข่ายความเร็วสูงที่ต้องรองรับทราฟฟิกเป็นปริมาณมาก ๆ ซึ่งเราสามารถนำแบบจำลองการประเมินประสิทธิภาพการตรวจจับการบุกรุกนี้ มาช่วยในการปรับแต่งกฎและสัญลักษณ์ให้เหมาะสมกับแต่ละสภาพแวดล้อม เพื่อเป็นแนวทางในการปรับแต่งกฎและสัญลักษณ์ของ NIDS บนระบบเครือข่ายคอมพิวเตอร์ที่ให้บริการในองค์กรได้

บรรณานุกรม

[1] N. Paulauskas., J. Skudutis, **Investigation of the Intrusion Detection System "Snort" Performance,** Department of Computer Engineering, Vilnius Gediminas Technical University, Lithuania, 2008.

[2] Lambert Schaelicke, Thomas Slabach, Branden Moore and Curt Freeland, **Characterizing the Performance of Network Intrusion Detection Sensors,** LNCS, Vol. 2820, pp. 155-172, Springer Berlin / Heidelberg, 2003.

[3] Lambert Schaelicke , Kyle Wheeler , Curt Freeland, **SPANIDS: A Scalable Network Intrusion Detection Loadbalancer,** ACM, pp. 315 – 322, New York, USA 2005.

[4] Amruta Inamdar, Manasi Joshi, **Intrusion Detection Systems and a Case Study of SNORT,** University of Minnesota, USA, 2003.

[5] Sourcefire Network Security Inc, **Snort 2.1 Second Edition,** pp. 488-490, May 2004.

[6] Sourcefire Inc, **SnortTM Users Manual 2.8.2 The Snort Project,** May 7, 2008.

[7] Windows Intrusion Detection System, **http://www.winids.com**

[8] SNORT an open source network intrusion detection system, **http://www.snort.org**

[9] Replays pcap files at arbitrary speeds onto the network, **http://tcpreplay.sourceforge.net**

[10] Nmap Security scanner, **http://nmap.org/zenmap**

[11] Snort installation, **http://thaicert.nectec.or.th/paper/ids/snort.php**

[12] จตุชัย แพงจันทร์, **Master in Security**, นนทบุรี : อินโฟเพรส 2550

[13] สมเกียรติ รุ่งเรืองลดา, **คู่มือดูแลระบบ Network ฉบับมืออาชีพ,** บริษัท ซีเอ็ดยูเคชั่น จำกัด มหาชน 2551

[14] บัณฑิต จามรภูติ, **คัมภีร์ RedHat Enterprise เล่ม 1-2,** สำนักพิมพ์บัณฑิต 2547

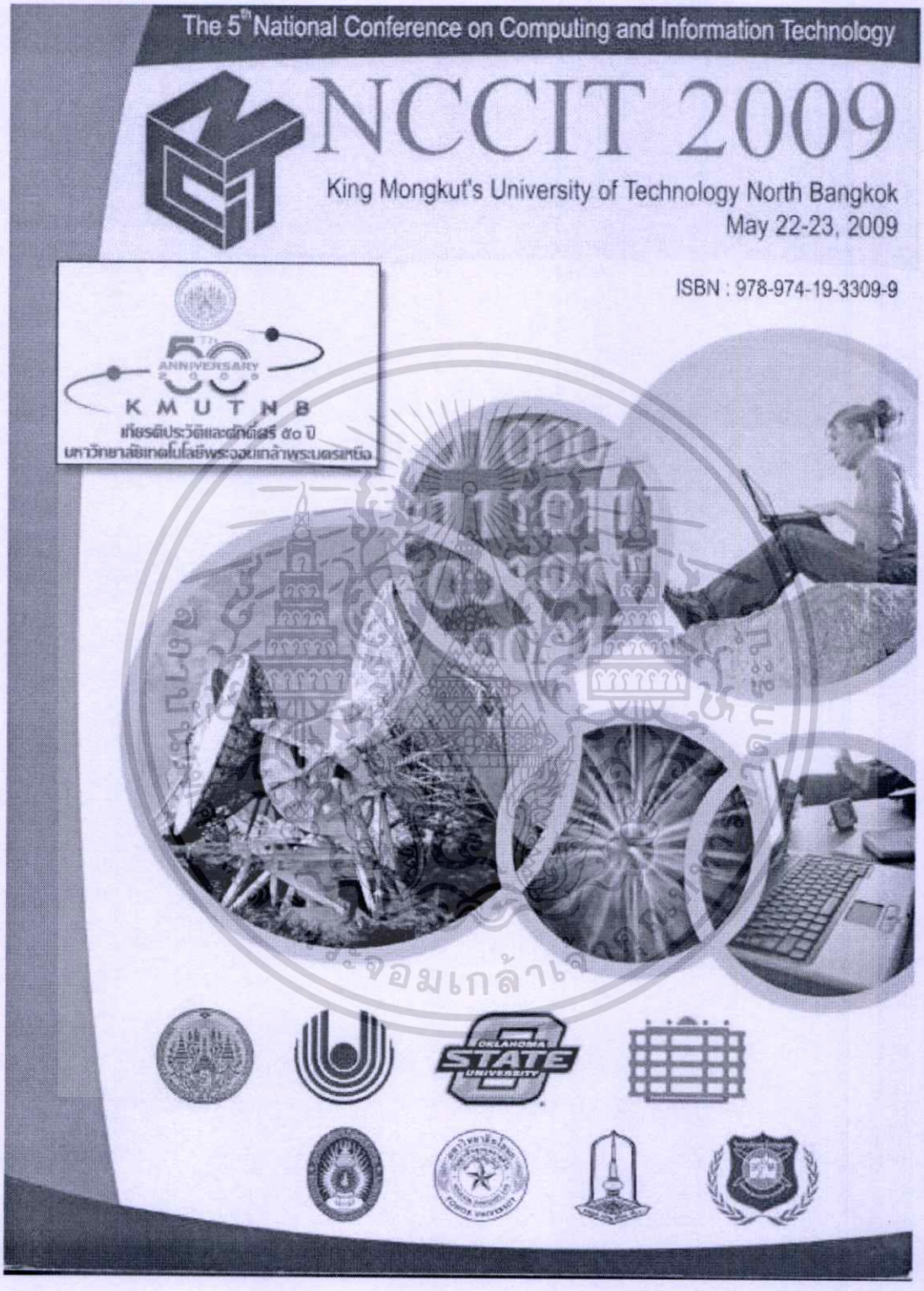
[15] Strategies to Reduce False Positives and False Negatives in NIDS , **http://www.securityfocus.com/infocus/1463**

[16] Basic Intrusion Detection System, ThaiCERT, **http://www.thaicert.nectec.or.th/paper/ids/idsfaq2.php**

ภาคผนวก

ภาคผนวก ก.
ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่

1. Tanapat Khantaracha and Asst.Prof. Mayuree Lertwatechakul, "**Network Intrusion Detection System Performance Evaluation**," The 5th National Conference on Computing and Information Technology (NCCIT 2009), pp.162-163, Bangkok, Thailand, May 22-23, 2009.



NCCIT09-IT84

การหาค่าประสิทธิภาพการตรวจจับการบุกรุกในระบบเครือข่ายคอมพิวเตอร์

Network Intrusion Detection System Performance Evaluation

ธนภัทร ขานทะราช[1] มยุรี เลิศวรกุล[2]
ภาควิชาวิศวกรรมสารสนเทศ
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
[1]S7061118@Gmail.com, [2]kimyurs@kmitl.ac.th

บทคัดย่อ

ระบบตรวจจับการบุกรุกเครือข่ายคอมพิวเตอร์ (NIDS) เป็นเครื่องมือสำคัญในการวัดความปลอดภัยของเครือข่ายในปัจจุบัน แต่เนื่องจากการที่มีความพยายามค้นหาวิธีการใหม่ ๆ เพื่อเจาะระบบอยู่เสมอ ดังนั้นจึงมีการเพิ่มจำนวนของกฎและสัญลักษณ์สำหรับการตรวจจับการบุกรุกอยู่ตลอดเวลาทำให้ภาระในการตรวจสอบกฎและสัญลักษณ์เพิ่มสูงขึ้น นอกทั้งระบบเครือข่ายคอมพิวเตอร์ได้รับการปรับปรุงให้มีความเร็วสูงขึ้นมากยาจจะส่งผลต่อประสิทธิภาพของระบบตรวจจับการบุกรุกให้ลดลงได้ งานวิจัยนี้ต้องการประเมินประสิทธิภาพของ NIDS โดยการใช้โปรแกรม Snort เวอร์ชั่น 2.8.2.2 เป็นตัวแทนเพื่อทดลองศึกษาผลกระทบของจำนวนกฎและสัญลักษณ์ที่มีต่อประสิทธิภาพของการตรวจจับการบุกรุก โดยใช้โปรแกรม TCPReplay-3.4.0 ในการจำลองทราฟฟิกที่มีความเร็วแตกต่างกัน ผลการทดลองได้แสดงว่าประสิทธิภาพของ NIDS จะเริ่มลดลงเมื่อความเร็วในการส่งข้อมูลเพิ่มสูงขึ้นจนถึงจุดอิ่มตัวของ NIDS และจากจุดนั้นเป็นต้นไปจะพบว่าประสิทธิภาพของ NIDS ที่มีจำนวนกฎและสัญลักษณ์มากที่สุดจะลดลงมากกว่าเมื่อเปรียบเทียบกับการทดลองที่ใช้ชุดกฎและสัญลักษณ์น้อยกว่า ซึ่งผลการวิจัยนี้ชี้ปัญหาที่เกิดจากความเร็วในการส่งข้อมูล อีกทั้งความจำเป็นต้องควบคุมการใช้กฎและสัญลักษณ์อย่างเหมาะสม

Abstract

At present, Network Intrusion Detection System (NIDS) is an essential tool to measure security of a network system. Since there have always been malicious attempts to seek new technology to hack into the networks, this increases number of rules and intruder signatures of NIDS. Moreover, today networking has been rapidly developed to be higher speed and all of the mentioned factors could overload NIDS and affect to their performance. This research focuses on the performance evaluation of Snort Network Intrusion Detection System version 2.8.2.2. Experiments were done in order to find the consequence of number of rules and signatures to the NIDS performance. By using TCPReplay-3.4.0 to generate normal traffic together with attacking traffic patterns at various speeds. And we tried to limit the number of Snort rules and signatures to figure out how the number of rule effect to NIDS performance. The experimental results have shown that the performance of NIDS will be dropping when the network traffic rate is increased to its saturate point. At the saturating traffic rate, higher number of rules and signatures of NIDS began to show bad effect to NIDS

performance. And at the higher traffic rate, NIDS performance with the greatest number of rules and signatures setting was reduced significantly compared to the others. The result has made us to realize the problem of NIDS performance that may be caused by high speed traffic rate. And it is necessary to control and limit the set of rules and signatures to be the optimal set.

NCCIT09-IT85

ระบบวินิจฉัยโรคเบื้องต้นจากระดูขาวโดยใช้โครงข่ายประสาทเทียมแบบแพร่กลับ

Diagnosis preliminary for leukorrhea system using back-propagation neural network

สุรเดช บุญลือ, ทาณิกา คำนะนาด, กัลญา แก้วแสน
ภาควิชาวิทยาการคอมพิวเตอร์
คณะเทคโนโลยีสารสนเทศ วิทยาลัยนอร์ทกรุงเทพ
bosuradej@northbkk.ac.th, joy.2499@hotmail.com, kawsan447@hotmail.com

บทคัดย่อ

งานวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาระบบวินิจฉัยโรคเบื้องต้นจากระดูขาวโดยโครงข่ายประสาทเทียม ข้อมูลที่นำมาใช้ได้มาจากศูนย์ควบคุมโรคติดต่อทางเพศสัมพันธ์โรงพยาบาลบางรัก ข้อมูลที่ได้ก่อนประมวลผลจะประกอบด้วยข้อมูลจำนวน 72 ระเบียน 17 คุณลักษณะ ทีมวิจัยนี้ได้ใช้สองเทคนิคในการพัฒนาระบบร่วมกัน คือ การประมวลผลภาพ และโครงข่ายประสาทเทียมแบบย้อนกลับ ได้มีการสอนให้มีการเรียนรู้แล้วทำการทดสอบ ผลการทดสอบมีความถูกต้องแม่นตรง 96% ของการเรียนรู้และการทดสอบของข้อมูลทั้งหมดระบบนี้พัฒนาขึ้นโดยใช้โปรแกรม Matlab 7.1 ในการพัฒนาได้ทำการทดสอบใช้งานกับกลุ่มผู้เชี่ยวชาญทางด้านโรคติดต่อทางเพศสัมพันธ์ มาประเมินคุณภาพของระบบ เพื่อหาค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐาน พบว่าค่าเฉลี่ยอยู่ในระดับ 4.15(SD = 0.17) ดังนั้นแสดงให้เห็นว่าระบบที่ได้พัฒนานี้มีคุณภาพในระดับดีสามารถนำไปใช้งานได้

Abstract

The project aims to development of diagnosis preliminary for leucorrhea system using Back-Propagation Neural Network. The data used is the Bangluk hospital. The preprocessed data set consists of 72 records, which have all the available 17 fields from the Bangluk hospital database. We have investigated two in the development with Image processing techniques: and the back-propagated neural network the instruction has had learning already does the testing The Result testing has accurate justice 96% of learning and the testing of the all data this system develops by program Matlab 7.1 The system would be evaluated in vary aspects such as the users demand. Moreover, the user satisfaction was brought to evaluation the users requirement of using the system. By doing so, disease sexual is related. As the result, the mean of the disease sexual is related expert group is 4.15 (SD = 0.17) It is generally speaking that the system has been performed

การหาค่าประสิทธิภาพการตรวจจับการบุกรุกในระบบเครือข่ายคอมพิวเตอร์

Network Intrusion Detection System Performance Evaluation

นายธนภัทร ขานทะราชา[1] และ ผศ.มยุรี เลิศเวชกุล[2]

ภาควิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

[1]S7061118@Gmail.com, [2]klmayure@kmitl.ac.th

Abstract

At present, Network Intrusion Detection System (NIDS) is an essential tool to measure security of a network system. Since there have always been malicious attempts to seek new technology to hack into the networks, this increases number of rules and intruder signatures of NIDS. Moreover, today networking has been rapidly developed to be higher speed and all of the mentioned factors could overload NIDS and affect to their performance. This research focuses on the performance evaluation of Snort Network Intrusion Detection System version 2.8.2.2. Experiments were done in order to find the consequence of number of rules and signatures to the NIDS performance. By using TCPReplay-3.4.0 to generate normal traffic together with attacking traffic patterns at various speeds. And we tried to limit the number of Snort rules and signatures to figure out how the number of rule effect to NIDS performance. The experimental results have shown that the performance of NIDS will be dropping when the network traffic rate is increased to its saturate point. At the saturating traffic rate, higher number of rules and signatures of NIDS began to show bad effect to NIDS performance. And at the higher traffic rate, NIDS performance with the greatest number of rules and signatures setting was reduced significantly compared to the others. The result has made us to realize the problem of NIDS performance that may be caused by high speed traffic rate. And it is necessary to control and limit the set of rules and signatures to be the optimal set.

Keywords: NIDS, Performance, Rules, Signatures

บทคัดย่อ

ระบบตรวจจับการบุกรุกเครือข่ายคอมพิวเตอร์ (NIDS) เป็นเครื่องมือสำคัญในการวัดความปลอดภัยของเครือข่ายในปัจจุบัน แต่เนื่องจากการที่มีความพยายามค้นหาวิธีการใหม่ ๆ เพื่อเจาะระบบอยู่เสมอ ดังนั้นจึงมีการเพิ่มจำนวนของกฎและสัญลักษณ์สำหรับการตรวจจับการบุกรุกอยู่ตลอดเวลาทำให้ภาระในการตรวจสอบกฎและสัญลักษณ์เพิ่มสูงขึ้น อีกทั้งระบบเครือข่ายคอมพิวเตอร์ได้รับการปรับปรุงให้มีความเร็วสูงขึ้นมากอาจจะส่งผลต่อประสิทธิภาพของระบบตรวจจับการบุกรุกให้ลดลงได้ งานวิจัยนี้ต้องการประเมินประสิทธิภาพของ NIDS โดยการใช้โปรแกรม Snort เวอร์ชั่น 2.8.2.2 เป็นตัวแทนเพื่อทดลองศึกษาผลกระทบของจำนวนกฎและสัญลักษณ์ที่มีต่อประสิทธิภาพของการตรวจจับการบุกรุก โดยใช้โปรแกรม TCPReplay-3.4.0 ในการจำลองทราฟฟิกที่มีความเร็วแตกต่างกัน ผลการทดลองได้แสดงว่าประสิทธิภาพของ NIDS จะเริ่มลดลงเมื่อความเร็วในการส่งข้อมูลเพิ่มสูงขึ้นจนถึงจุดอิ่มตัวของ NIDS และจากจุดนั้นเป็นต้นไปจะพบว่าประสิทธิภาพของ NIDS ที่มีจำนวนกฎและสัญลักษณ์มากที่สุดจะลดลงมากกว่าเมื่อเปรียบเทียบกับการทดลองที่ใช้ชุดกฎและสัญลักษณ์น้อยกว่า ซึ่งผลการวิจัยนี้ชี้ปัญหาที่เกิดจากความเร็วในการส่งข้อมูล อีกทั้งความจำเป็นต้องควบคุมการใช้กฎและสัญลักษณ์อย่างเหมาะสม

คำสำคัญ: ประสิทธิภาพ กฎ สัญลักษณ์

1. บทนำ

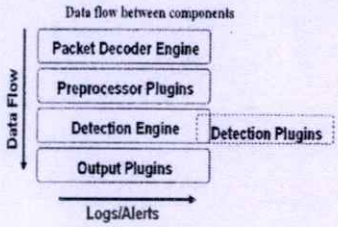
นับว่าระบบตรวจจับการบุกรุก (Network Intrusion Detection System : NIDS) เป็นเครื่องมือที่สำคัญอย่างยิ่งในระบบเครือข่ายคอมพิวเตอร์ที่จะใช้ในการรับมือกับการบุกรุกโจมตีระบบเครือข่าย โปรแกรม NIDS ในปัจจุบันมีแนวโน้มที่จะมีการพัฒนาให้สามารถทำงานประสานกับโปรแกรมเครื่องมือสำหรับรักษาความปลอดภัยของระบบคอมพิวเตอร์ประเภทอื่นๆ มากยิ่งขึ้น อาทิเช่น มีการทำงานประสานกับโปรแกรมไฟร์วอลล์ โดยหากตรวจพบว่ามีการโจมตีจากเครื่องที่ใช้ไอพีแอดเดรสใดก็อาจสั่งการให้ไฟร์วอลล์ระงับการสื่อสารของไอพีแอดเดรสนั้น หรือสั่งให้ไฟร์วอลล์กำหนดใช้นโยบายควบคุมการใช้งานระบบเครือข่ายที่เข้มงวดมากยิ่งขึ้น นอกจากนี้ยังอาจจะทำงานประสานกับโปรแกรมตรวจจับไวรัส โดยทำให้ระบบสามารถตรวจจับไวรัสได้จากแพ็กเก็ต (Packet) ที่บรรจุไวรัสขณะที่แพ็คเก็จนั้นกำลังถูกส่งผ่านระบบเครือข่ายซึ่งทำให้เครื่องคอมพิวเตอร์ของผู้ใช้มีโอกาสในการติดไวรัสที่แพร่กระจายผ่านทางระบบเครือข่ายลดลง

เนื่องจาก NIDS เป็นเครื่องมือที่สำคัญในการรักษาความปลอดภัยของระบบเครือข่ายคอมพิวเตอร์ ดังนั้นจึงได้มีการศึกษาประสิทธิภาพของการตรวจจับการบุกรุกโดยงานวิจัยต่าง ๆ เช่นงานวิจัยเรื่อง "Investigation of the Intrusion Detection System "Snort" Performance" [1] ได้ทำการวิจัยถึงปัจจัยของการใช้ทรัพยากรฮาร์ดแวร์และซอฟท์แวร์ระบบจัดการฐานข้อมูลที่แตกต่างกันว่าส่งผลกระทบต่อประสิทธิภาพการตรวจจับการบุกรุกของโปรแกรม Snort หรือไม่อย่างไร เมื่อมีการส่งข้อมูลบนระบบเครือข่ายคอมพิวเตอร์ในอัตราความเร็วที่แตกต่างกัน โดยพิจารณาเปรียบเทียบเมื่อปรับเปลี่ยนปัจจัยของทรัพยากรต่างๆ อาทิเช่น ความเร็วของซีพียู (CPU) ขนาดของหน่วยความจำ ขนาดของฮาร์ดดิสก์ ชนิดและความเร็วของการ์ดสื่อสาร ซึ่งแสดงให้เห็นว่าที่ความเร็วในการสื่อสารข้อมูลระดับเดียวกัน ฮาร์ดแวร์ที่ประสิทธิภาพต่ำจะมีค่าการใช้งานซีพียู (CPU usage) ที่สูงมากซึ่งสัมพันธ์กับจำนวนการตรวจจับการบุกรุกที่ลดลงมากเช่นเดียวกัน เมื่อเปรียบเทียบกับเครื่องที่มีฮาร์ดแวร์ประสิทธิภาพสูงกว่า รวมทั้งการจัดเก็บในฐานข้อมูลของ MySQL นั้นจะพบว่าประสิทธิภาพการตรวจจับการบุกรุกจะต่ำกว่าการเก็บข้อมูลบน Barnyard 0.2.0 ส่วนงานวิจัยเรื่อง "Characterizing the Performance of Network Intrusion Detection Sensors" [2] ได้ทำการวิจัยถึงประสิทธิภาพของระบบตรวจจับการบุกรุกเมื่อถูกติดตั้งลงบนระบบปฏิบัติการที่แตกต่างกัน โดยได้ทดสอบกับระบบปฏิบัติการ Linux 3.0 kernel 2.4.19 และ FreeBSD 4.5 รวมทั้งศึกษาเปรียบเทียบประสิทธิภาพเมื่อใช้ตัวประมวลผลเดียว (Single processor) และตัวประมวลผลคู่ (Dual processor) ซึ่งก็ได้แสดงให้เห็นว่าตัวประมวลผลคู่ไม่ได้มีนัยที่จะช่วยเพิ่มประสิทธิภาพในการตรวจจับการบุกรุกให้มากขึ้นเท่าที่ควรเมื่อเทียบกับค่าใช้จ่ายที่เพิ่มมากขึ้น ส่วนงานวิจัยนี้ได้เสนอแนวทางการประเมินประสิทธิภาพของระบบตรวจจับการบุกรุกที่แตกต่างกันออกไป โดยมุ่งเน้นในการหาค่าประสิทธิภาพของโปรแกรมตรวจจับการบุกรุกระบบเครือข่ายคอมพิวเตอร์โดยการทดสอบกับโปรแกรม Snort เวอร์ชั่น 2.8.2.2 เพื่อหาผลกระทบเนื่องจากการเพิ่มขึ้นของจำนวนกฎ (Rules) และสัญลักษณ์ (Signatures) ที่อาจจะส่งผลต่อประสิทธิภาพของการตรวจจับการบุกรุกในสถานการณ์ที่ปริมาณและอัตราความเร็วในการส่งข้อมูลผ่านระบบเครือข่ายมีระดับแตกต่างกัน

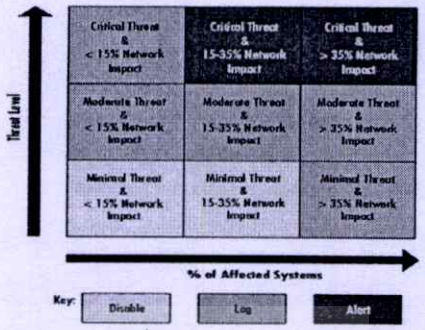
2. โครงสร้างของโปรแกรม Snort

2.1 สถาปัตยกรรม Snort โปรแกรม Snort เป็นเครื่องมือที่ใช้ตรวจจับการบุกรุกทางระบบเครือข่าย (Network Intrusion Detection System : NIDS) ที่นำเสนอโดย Martin Roesch การทำงานของ Snort จะใช้ไลบรารีพื้นฐานชื่อ Libpcab ซึ่งใช้กันโดยทั่วไปในบรรดาโปรแกรมดักฟังระบบเครือข่าย (Network sniffer) และโปรแกรมวิเคราะห์ระบบเครือข่าย (Network analyzer) สำหรับโปรแกรม Snort นั้นมีความสามารถในการวิเคราะห์แพ็คเก็ตสื่อสาร (Protocol analysis) ตรวจสอบเนื้อความภายในแพ็กเก็ต (Content searching / Matching) ตรวจจับการบุกรุก (Cracking in) การโจมตี (Attack) และการสืบหาช่องโหว่ของระบบ (Probe) เช่น Buffer overflow, Stealth port scan, CGI attack, SMB probe, OS Fingerprint และอื่นๆ

โดยโมดูลการทำงานของโปรแกรม Snort สามารถแสดงได้ดังรูปที่ 1



รูปที่ 1 โมดูลการทำงานของโปรแกรม Snort



รูปที่ 2 การแยกประเภทของกฎ

โปรแกรม Snort ประกอบไปด้วย 4 กลไกหลักคือ Packet Decoder Engine, Preprocessor Plugins, Detection Engine และ Output Plugins [4], [7]. ที่มีหน้าที่ตรวจจับการบุกรุกโดยการเปรียบเทียบกับกฎและสัญลักษณ์ที่กำหนดไว้ ซึ่งถ้าพบว่าแพ็กเก็ตชุดใดมีลักษณะตรงกับกฎที่ตรวจสอบก็จะทำงานตาม Rule action ที่ได้กำหนดเอาไว้และแสดงผลของการตรวจสอบการบุกรุกให้ผู้ดูแลระบบทราบ เพื่อดำเนินการแก้ไขได้อย่างถูกต้อง

2.2 หลักการกำหนดกฎและสัญลักษณ์ของ Snort เพื่อเป็นการตั้งค่าให้กับระบบตรวจจับการบุกรุกในระบบเครือข่ายอย่างเหมาะสมนั้นนิยมยึดถือหลักเกณฑ์สองข้อหลัก ๆ ดังต่อไปนี้ [5]

- **การระบุชนิดของโปรโตคอลและบริการที่เปิดใช้บนระบบเครือข่าย** เช่น ควรกำหนดให้แจ้งเตือนสำหรับชุดของกฎและสัญลักษณ์ที่อ้างถึงบริการและโปรโตคอลที่ในระบบเครือข่ายคอมพิวเตอร์เปิดให้บริการเท่านั้น ส่วนความพยายามในการสร้างการเชื่อมต่อขอใช้บริการจากภายนอกมายังบริการหรือโปรโตคอลที่ระบบไม่ได้เปิดให้บริการนั้น ควรที่จะกำหนดไว้เพื่อเก็บบันทึกข้อมูล (log) ของทราฟฟิก เท่านั้น
- **การกำหนดระดับความสำคัญของสภาวะแวดล้อมของระบบเครือข่าย** เช่น ถ้าเป็นระบบเครือข่ายที่ถูกใช้โดยกลุ่มผู้พัฒนาระบบ ควรกำหนดใช้กฎในการตรวจจับการบุกรุกให้เข้มงวดน้อยกว่าการกำหนดกฎสำหรับระบบเครือข่ายสำหรับฝ่ายการเงิน หรือระบบเครือข่ายที่เปิดการให้บริการสู่สาธารณะ เป็นต้น

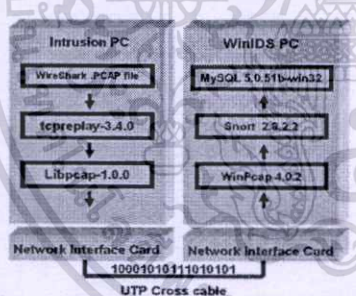
ในรูปที่ 2 แสดงวิธีการแบ่งประเภทของกฎ (Rule sets) เพื่อตรวจสอบและตอบสนองต่อการบุกรุกตามระดับความรุนแรงของการคุกคาม (Threat level) อย่างเหมาะสม เพื่อช่วยสร้างเกณฑ์ขั้นต่ำ (Baseline) ให้เราสามารถกำหนดระดับความอันตรายของการบุกรุกได้ดังตัวอย่างต่อไปนี้

- **การคุกคามในระดับรุนแรง (Critical threats)** อาทิเช่น SQL Slammer worm, CodeRed และ IIS Unicode attacks ซึ่งสามารถค้นหาและบุกรุกผ่านช่องโหว่ที่ยังไม่ได้รับการแก้ไข รวมถึงการแพร่กระจายอย่างรวดเร็ว
- **การคุกคามในระดับปานกลาง (Moderate threats)** อาทิเช่น MDAC remote buffer overflow, Wu-FTP buffer overflow, OpenSSL bugs ซึ่งทำให้เกิดบัฟเฟอร์ล้นจนไม่สามารถให้บริการได้ตามปกติส่วนใหญ่ไม่ได้ส่งผลกระทบอย่างกว้างขวาง
- **การคุกคามในระดับเล็กน้อย (Minimal threats)** อาทิเช่น Bind TSIG, การใช้ช่องโหว่ของ CGI และโปรโตคอล SMTP เป็นต้น ซึ่งทำให้ระบบไม่สามารถให้บริการได้ตามปกติอย่างไรก็ตามช่องโหว่นี้ไม่ได้เป็นช่องโหว่ที่มีความเสี่ยงสูง

3. รูปแบบการทดลอง

เนื่องจากผู้บุกรุกจะค้นหาวิธีใหม่ๆ เพื่อเจาะระบบเครือข่ายคอมพิวเตอร์อยู่เสมอ จึงทำให้จำนวนกฎและสัญลักษณ์ที่โปรแกรม Snort จะต้องตรวจสอบมีแนวโน้มที่จะเพิ่มจำนวน

มากขึ้นเรื่อยๆ และจากสมมุติฐานที่ว่าถ้าหากมีการเพิ่มจำนวนของกฎและสัญลักษณ์ ซึ่งเป็นการเพิ่มภาระในการตรวจสอบให้แก่ระบบการตรวจจับการบุกรุก ก็อาจจะส่งผลให้ประสิทธิภาพการทำงานของโปรแกรม Snort เกิดปัญหาได้ โดยเฉพาะอย่างยิ่งเมื่อความเร็วและปริมาณของ ทราฟฟิกในการส่งข้อมูลของระบบเครือข่ายเพิ่มสูงมากขึ้นและเพื่อศึกษาผลกระทบดังกล่าวการวิจัยนี้จึงได้ออกแบบการทดลองเพื่อจำลองสถานการณ์ดังข้างต้น การทดลองได้ดำเนินการด้วยการติดตั้งโปรแกรม Snort version 2.8.2.2 บนระบบปฏิบัติการ Windows XP [6] บนเครื่องพีซี 1 เครื่อง และจำลองการทำงานของเครื่องผู้บุกรุก (Intrusion PC) ด้วยเครื่องพีซีอีกเครื่องหนึ่งที่ติดตั้งโปรแกรม TCPReplay-3.4.0 [8] เพื่อทำการอ่านไฟล์ข้อมูลที่จำลองการบุกรุกโดยการใช้โปรแกรม Zenmap 4.98 [9] โดยโปรแกรม TCPReplay-3.4.0 จะทำหน้าที่ในการปรับความเร็วของการส่งผ่านข้อมูลที่อ่านจากไฟล์ดังกล่าวให้มีความเร็วเพิ่มขึ้นในแต่ละรอบของการทดลอง ผ่านไปยังสายสื่อสาร (cross line) ที่เชื่อมโยงระหว่างเครื่องพีซีทั้งสองเครื่องนั้นดังโครงสร้างที่แสดงไว้ในรูปที่ 3 โดยเครื่องพีซีทั้งสองนั้นมีรายละเอียดดังนี้ Model: HP DX5150MT, CPU: Athlon64 2.0Ghz/512KB cache, RAM: 2GB DDR2-400, NIC: Broadcom NetXtreme Gigabit Ethernet, HDD: 80GB SATA (7200rpm)



รูปที่ 3 โครงสร้างการจำลองเพื่อวัดประสิทธิภาพของ NIDS

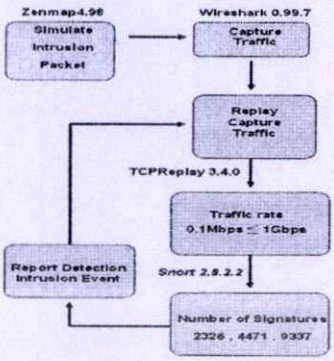
- การกำหนดรูปแบบการจำลองการบุกรุกโดยโปรแกรม Zenmap 4.98 มีรายละเอียดดังต่อไปนี้
- จำนวนของทราฟฟิกบุกรุกรวม 100,045 แพ็กเก็ต แบ่งออกเป็น TCP 19.45% UDP 41.64% และ ICMP 38.90%
- ค่าเฉลี่ยความเร็วของการส่งข้อมูล (Average data transmission rate) 0.127 Mbps
- จำนวนครั้งของการโจมตีทั้งหมด ที่จำลองโดยโปรแกรม Zenmap4.98 แยกตามชนิดการโจมตีได้เป็นการส่ง Executable code (Shellcode-detect) จำนวน 351 ครั้ง การพยายามเจาะระบบ (Attempted-recon) จำนวน 164 ครั้ง การบุกรุกรูปแบบอื่น ๆ (Misc-attack) จำนวน 328 ครั้ง รวมจำนวนครั้งของการบุกรุกทั้งสิ้นเป็นจำนวน 843 ครั้ง

และเพื่อพิสูจน์ว่าจำนวนของกฎและสัญลักษณ์มีผลต่อประสิทธิภาพในการตรวจจับการบุกรุกหรือไม่ จึงได้ทดสอบโดยจัดกลุ่มของกฎและสัญลักษณ์ที่ใช้ในการตรวจจับการบุกรุกในโปรแกรม Snort ออกเป็น 3 กลุ่มคือ กลุ่มที่หนึ่งมีจำนวน 9,337 สัญลักษณ์ กลุ่มที่สองมีจำนวน 4,471 สัญลักษณ์ และกลุ่มที่สาม มีจำนวน 2,326 สัญลักษณ์ดังแสดงในตารางที่ 1

ตารางที่ 1 ชนิดและจำนวนกฎและสัญลักษณ์สำหรับทดลอง

ชื่อกฎ	สัญลักษณ์ทั้งหมด	กลุ่มที่1	กลุ่มที่2	กลุ่มที่3
icmp-info.rules	22	22	22	22
icmp.rules	93	93	93	93
misc.rules	152	152	152	152
netbios.rules	5,786	4,159	4,159	2,014
shellcode.rules	29	29	29	29
snmp.rules	19	16	16	16
Other rules	7,517	4,866	n/a	n/a
สัญลักษณ์รวม	13,618	9,337	4,471	2,326

โดยเครื่องผู้บุกรุก (Intruder PC) จะทำการจำลองความเร็วของการส่งข้อมูลด้วยโปรแกรม TCPReplay-3.4.0 ในระดับอัตราความเร็วในการสื่อสารข้อมูลที่มีค่าตั้งแต่ 0.127 Mbps, 100 Mbps, 200 Mbps, 300 Mbps, 400 Mbps, 500 Mbps, 600 Mbps, 700 Mbps, 800 Mbps, 900 Mbps และ 1 Gbps แล้วทำการตรวจวัดจำนวนครั้งที่โปรแกรม Snort สามารถตรวจจับการบุกรุกได้ เพื่อเปรียบเทียบกับผลการทดลองที่ได้แต่ละรอบ ดังขั้นตอนการหาค่าประสิทธิภาพ ในรูปที่4

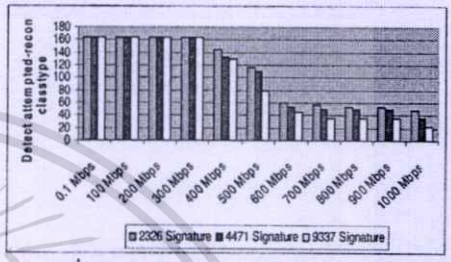


รูปที่ 4 ขั้นตอนการหาค่าประสิทธิภาพ

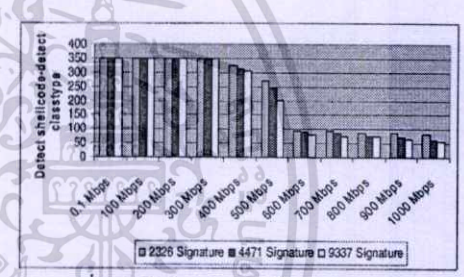
4. ผลการทดลอง

จากผลการทดลองดังรูปที่ 5 ถึงรูปที่ 8 จะเห็นได้ว่าประสิทธิภาพการตรวจพบการบุกรุกของ Snort จะเริ่มลดลงเมื่อความเร็วของการส่งข้อมูลเพิ่มขึ้นถึงระดับ 300 Mbps ซึ่งถือได้ว่าเป็นจุดอิ่มตัวของโปรแกรม Snort ที่ทำงานบนเครื่องคอมพิวเตอร์ที่ใช้ในการทดลองซึ่งมีรายละเอียดดังแสดงประกอบไว้ในบทที่ 3 และเมื่อความเร็วในการสื่อสารข้อมูลเพิ่มขึ้นสูงกว่าจุดอิ่มตัวของเครื่องที่ให้บริการตรวจจับการบุกรุกประสิทธิภาพของการตรวจจับการบุกรุกก็จะยิ่งลดลงไปเรื่อย ๆ ซึ่งจำนวนครั้งของการตรวจพบการบุกรุกนั้นจะมีความสัมพันธ์กับเปอร์เซ็นต์การใช้งานซีพียู (CPU usage) ที่แสดงในรูปที่ 9 โดยพบว่าที่ความเร็วในการสื่อสารข้อมูลในระดับ 300 Mbps ซึ่งเป็นจุดอิ่มตัวนั้น เป็นตำแหน่งที่มีเปอร์เซ็นต์การใช้งานซีพียูประมาณ 40% จะส่งผลให้เริ่มมีการทิ้งแพ็กเก็ตข้อมูลก่อนเวลาอันสมควร

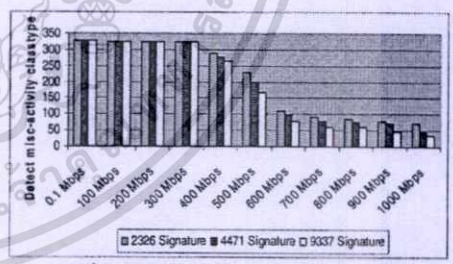
การทดลองได้แสดงให้เห็นว่าเมื่อกำหนดให้โปรแกรม Snort ตรวจจับการบุกรุกโดยการตรวจสอบด้วยกฎและสัญลักษณ์จำนวนมากนั้น โปรแกรม Snort จะมีประสิทธิภาพการตรวจจับการบุกรุกที่ด้อยกว่าการทำงานในสภาพแวดล้อมเดียวกันแต่กำหนดให้มีการใช้กฎและสัญลักษณ์ที่มีจำนวนน้อยกว่า โดยผลการทดลองทั้งหมดนั้นสอดคล้องกันทั้งสำหรับการตรวจจับการบุกรุกแบบแยกตามชนิดของการโจมตีดังแสดงในรูปที่ 5 ถึง 7 และผลสรุปการตรวจจับการบุกรุกทั้งหมดดังแสดงในรูปที่ 8 ซึ่งก็เป็นไปตามสมมุติฐานในการทดลองว่าจำนวนของกฎและสัญลักษณ์ของโปรแกรม Snort น่าจะมีผลกระทบกับประสิทธิภาพโดยรวมของการตรวจจับการบุกรุกเมื่อความเร็วในการส่งผ่านข้อมูลในระบบเครือข่ายสูงขึ้น



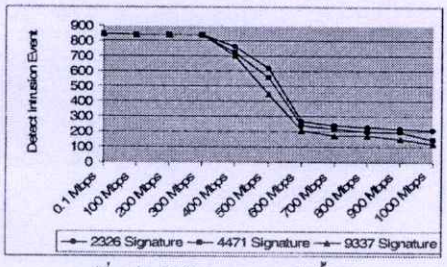
รูปที่ 5 ประสิทธิภาพการตรวจจับ Attempted-recon



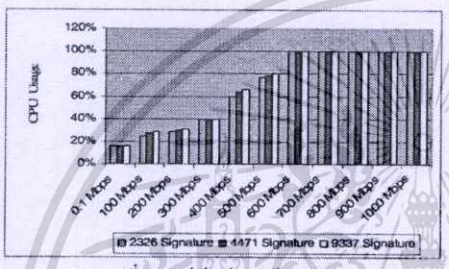
รูปที่ 6 ประสิทธิภาพการตรวจจับ Shellcode-detect



รูปที่ 7 ประสิทธิภาพการตรวจจับ Misc-activity



รูปที่ 8 ประสิทธิภาพการตรวจจับทั้งหมด



รูปที่ 9 เปอร์เซ็นต์การใช้งานซีพียู

5. บทสรุป

จากงานวิจัยอื่นๆ ที่ได้ทำการทดลองและสรุปผลได้ว่าประสิทธิภาพของการตรวจจับการบุกรุกภายในระบบเครือข่ายนั้นขึ้นอยู่กับปัจจัยหลายประการทั้งปัจจัยเกี่ยวกับฮาร์ดแวร์หรือซอฟท์แวร์ อาทิเช่น ความเร็วของซีพียู ขนาดของหน่วยความจำ ขนาดของฮาร์ดดิสก์ ชนิดและความเร็วของการ์ดสื่อสาร ซอฟท์แวร์ระบบจัดการฐานข้อมูล [1] รวมทั้งปัจจัยของระบบปฏิบัติการ (Operating System) [2] นั้น งานวิจัยนี้ได้ทำการทดลองและตรวจสอบหาปัจจัยอื่น ๆ ที่ส่งผลต่อประสิทธิภาพของระบบตรวจจับการบุกรุกโดยรวม โดยพิจารณาด้วยการปรับจำนวนของกฎและสัญลักษณ์รวมถึงความเร็วของการสื่อสารข้อมูล โดยผลการทดลองได้แสดงให้เห็นว่าประสิทธิภาพของระบบตรวจจับการบุกรุกเครือข่าย (NIDS) จะเริ่มลดลงเมื่อความเร็วในการส่งผ่านข้อมูลเพิ่มสูงขึ้นจนถึงจุดอิ่มตัวของเครื่องผู้ให้บริการตรวจจับการบุกรุก ซึ่งสัมพันธ์กับเปอร์เซ็นต์การใช้งานซีพียูที่เพิ่มสูงขึ้นประมาณ 40% จะส่งผลให้เริ่มมีการทิ้งแพ็กเก็ตข้อมูลก่อนเวลาอันสมควร และจากจุดอิ่มตัวนี้เป็นต้นไปจึงจะสังเกตุเห็นได้ว่าประสิทธิภาพของระบบตรวจจับการบุกรุกที่ใช้กฎและสัญลักษณ์ในจำนวนที่มากกว่านั้นจะเริ่มแสดงประสิทธิภาพที่แย่กว่าระบบตรวจจับการบุกรุกที่ใช้กฎและสัญลักษณ์ในจำนวนน้อยกว่า ดังแสดงในรูปที่ 5 ถึงรูปที่ 8 และเมื่อความเร็วในการส่งผ่านข้อมูลยิ่งเพิ่มสูงขึ้น ประสิทธิภาพของระบบตรวจจับการบุกรุกที่ต้องตรวจสอบกฎและสัญลักษณ์จำนวนมากกว่านั้น ก็จะยิ่งตรวจจับการบุกรุกได้น้อยกว่าอย่างชัดเจน

ดังนั้นการคิดตั้งกฎและสัญลักษณ์ในการตรวจจับการบุกรุกที่เหมาะสมสำหรับระบบเครือข่ายจึงถือเป็นเรื่องที่มีนัยสำคัญ ถ้าหากกำหนดให้ระบบตรวจจับการบุกรุกใช้กฎและสัญลักษณ์ที่สอดคล้องกับสภาวะแวดล้อมในการทำงานของระบบเครือข่าย รวมทั้งกำหนดความเข้มงวดของการใช้กฎและสัญลักษณ์ตามความจำเป็นของสภาวะแวดล้อมนั้น ๆ ก็จะสามารถลดจำนวนกฎและสัญลักษณ์และการแจ้งเตือนที่ไม่จำเป็น อีกทั้งสามารถเพิ่มประสิทธิภาพของการตรวจจับการบุกรุกได้ โดยเฉพาะอย่างยิ่งสำหรับระบบเครือข่ายความเร็วสูงที่ต้องรองรับทราฟฟิกปริมาณมาก

6. เอกสารอ้างอิง

[1] N. Paulauskas., J. Skudutis, "Investigation of the Intrusion Detection System "Snort" Performance," Computer Engineering, Vilnius Gediminas Technical University, Lithuania 2008

[2] Lambert Schaelicke, Thomas Slabach, Branden Moore and Curt Freeland, "Characterizing the Performance of Network Intrusion Detection Sensors." *LNCS*, Vol. 2820, pp 155-172, Springer Berlin / Heidelberg , 2003

[3] Lambert Schaelicke , Kyle Wheeler , Curt Freeland, "SPANIDS: A Scalable Network Intrusion Detection Loadbalancer." *ACM*, pp 315 – 322. New York, USA, 2005

[4] Amruta Inamdar, Manasi Joshi, "Intrusion Detection Systems and a Case Study of SNORT." University of Minnesota, University of Minnesota, USA, 2003

[5] Sourcefire Network Security Inc. "Snort 2.1 Second Edition." pp 488-490, May 2004

[6] Windows Intrusion Detection System, http://www.winids.com

[7] SNORT an open source NIDS, http://www.snort.org

[8] Replays pcap files, http://tcpreplay.sourceforge.net

[9] Nmap Security scanner, http://nmap.org/zenmap

ประวัติผู้เขียน

ชื่อผู้เขียน	นายธนภัทร ขานทะราชา
วัน เดือน ปีเกิด	9 มิถุนายน พ.ศ.2519
ที่อยู่	หมู่บ้าน ภัสสร 7 ดิอิลิแกนซ์ 87/178 หมู่ 6 ซอย 14
	ถนน บางกรวย-ไทรน้อย อำเภอบางบัวทอง
	จังหวัดนนทบุรี 11110
ประวัติการศึกษา	2542 อุตสาหกรรมศาสตรบัณฑิต
	สาขาเทคโนโลยีอิเล็กทรอนิกส์
	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ความชำนาญเฉพาะด้าน	1.) ระบบโทรคมนาคม
	2.) ระบบเครือข่ายคอมพิวเตอร์
	3.) ระบบรักษาความปลอดภัยเครือข่ายคอมพิวเตอร์
ประสบการณ์การทำงานและผลงานวิจัย	
2543-2545	ตำแหน่งวิศวกร ประจำศูนย์ปฏิบัติระบบเครือข่ายคอมพิวเตอร์
	บริษัทสามารถคอมมิวนิเคชั่นเซอร์วิส จำกัด
2545-2545	ตำแหน่งวิศวกรระบบชุมสายโทรศัพท์เคลื่อนที่ ประจำประเทศ
	กัมพูชา บริษัท Cambodia Samart Communications Co., Ltd
2546-2548	ตำแหน่งที่ปรึกษาฝ่ายเทคนิคด้านระบบเครือข่ายคอมพิวเตอร์
	บริษัทอินโฟเทควิชั่นจำกัด ในเครือเจริญโภคภัณฑ์
2548-ปัจจุบัน	ตำแหน่งวิศวกรระบบเครือข่ายคอมพิวเตอร์อาวุโส
	บริษัทสแปนชั่นไทยแลนด์ จำกัด
2548	ประกาศนียบัตร Cisco Certified Network Associate (CCNA)
2551	ประกาศนียบัตร Cisco Certified Network Professional (CCNP)