การจำแนกประเภทการแจ้งเตือนการบุกรุก โดยใช้ล็อก IDS และเว็บพร็อกซีร่วมกัน

INTRUSION ALERT CLASSIFICATION USING NETWORK-BASED IDS AND WEB PROXY CORRELATION LOGS

บุญลือ ด้วงสำรวย
BOONLUE DUANGSUMRUAY

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาเทคโนโลยีสารสนเทศ
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2547

ISBN 974-15-1048-9

การจำแนกประเภทการแจ้งเตือนการบุกรุก โดยใช้ล็อก IDS และเว็บพร็อกซีร่วมกัน

INTRUSION ALERT CLASSIFICATION USING NETWORK-BASED IDS AND WEB PROXY CORRELATION LOGS

บุญลือ ด้วงสำรวย
BOONLUE DUANGSUMRUAY

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาเทคโนโลยีสารสนเทศ
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2547

ISBN 974-15-1048-9

INTRUSION ALERT CLASSIFICATION USING NETWORK-BASED IDS AND WEB PROXY CORRELATION LOGS

BOONLUE DUANGSUMRUAY

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF SCIENCE IN INFORMATION TECHNOLOGY
SCHOOL OF GRADUATE STUDIES
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2004

ISBN 974-15-1048-9

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การจำแนกประเภทการแจ้งเตือนการบุกรุกโดยใช้ล็อก IDS และเว็บพร็อกซีร่วมกัน

INTRUSION ALERT CLASSIFICATION USING NETWORK-BASED IDS AND WEB PROXY CORRELATION LOGS

ชื่อนักศึกษา นายบุญลือ ด้วงสำรวย

รหัสประจำตัว 42067006

ปริญญา วิทยาศาสตรมหาบัณฑิต

สาขาวิชา เทคโนโลยีสารสนเทศ

อาจารย์ผู้ควบคุมวิทยานิพนธ์ ผศ.ดร.จันทร์บูรณ์ สถิตวิริยวงศ์

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
ผศ.ดร.จันทร์บูรณ์	สถิตวิริยวงศ์	
ผศ.ดร.นพพร	โชติกกำธร	
ผศ.ดร.โชติพัชร์	ภรณวลัย	
ผศ.อัครินทร์	คุณกิตติ	
ดร.พิเชฐ์	ม่วงนวล	

วัน/เดือน/ปี ที่สอบ 30 พฤษภาคม 2547 เวลา 9.00 น. เป็นต้นไป

สถานที่สอบ ณ ห้อง M 21 (ชั้นลอย) อาคารเรียนรวมและปฏิบัติการคณะเทคโนโลยีสารสนเทศ

บัณฑิตวิทยาลัยรับรองแล้ว

(ผศ.ดร.อารุวัตร เจริญสุข)

คณบดีบัณฑิตวิทยาลัย

วันที่ 24 **เดือน** มิถุนายน **พ.ศ.** 2547

หัวข้อวิทยานิพนธ์	การจำแนกประเภทการแจ้งเตือนการบุกรุก โดยใช้ล็อก IDS และเว็บพร็อกซีร่วมกัน
ชื่อนักศึกษา	นาย บุญลือ ด้วงสำรวย
รหัสประจำตัว	42067006
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
พ.ศ.	2547
อาจารย์ผู้ควบคุมวิทยานิพนธ์	ผศ. ดร. จันทร์บูรณ์ สถิตวิริยวงศ์

บทคัดย่อ

ระบบรักษาความปลอดภัยบนเครือข่ายที่ให้บริการเว็บเซิร์ฟเวอร์ในปัจจุบันประกอบด้วยระบบไฟร์วอลล์ (Firewall) สำหรับกลั่นกรองข้อมูลระหว่างเครือข่ายและระบบตรวจสอบการบุกรุก (Intrusion Detection System-IDS) สำหรับตรวจสอบการบุกรุกภายในเครือข่าย โดยระบบตรวจสอบการบุกรุกบนเครือข่าย (Network-based IDS) มักจะแสดงผลการแจ้งเตือน (alert) ที่ผสมกันระหว่างการแจ้งเตือนการบุกรุกที่ประสบความสำเร็จและการบุกรุกที่ไม่ประสบความสำเร็จเป็นจำนวนมาก ทำให้ผู้ดูแลระบบประสบปัญหาในการจัดลำดับความสำคัญในการป้องกันและแก้ไขปัญหาการบุกรุกได้อย่างเหมาะสม ปกติแล้วทั้งไฟร์วอลล์และ IDS มักจะบันทึกข้อมูลต่าง ๆ ลงล็อกไฟล์ (log files) เป็นจำนวนมาก การวิเคราะห์ข้อมูลจากล็อกในปัจจุบันนิยมทำแยกกันระหว่างไฟร์วอลล์และ IDS ในความเป็นจริงข้อมูลล็อกของไฟร์วอลล์และ IDS จะมีข้อมูลบางส่วนที่มีความสัมพันธ์เชื่อมโยงกันอยู่ ดังนั้นวิทยานิพนธ์นี้จึงนำเสนอวิธีการจำแนกประเภทการแจ้งเตือนการบุกรุก โดยการนำล็อกไฟล์จากเว็บพร็อกซีซึ่งเป็นไฟร์วอลล์ในระดับแอพพลิเคชันชนิดหนึ่งมาหาความสัมพันธ์กับล็อกของ Network-based IDS ทำให้สามารถทราบถึงข้อมูลประกอบอื่น ๆ ของการบุกรุกนั้น ๆ ซึ่งข้อมูลเหล่านี้จะตัวแปรสำคัญในการพิจารณาจัดลำดับการแจ้งเตือนการบุกรุกของระบบ IDS ในปัจจุบัน

Thesis Title	Intrusion Alert Classification using Network-based IDS and Web Proxy Correlation Logs
Student	Mr. Boonlue Duangsumruay
Student ID.	42067006
Degree	Master of Science
Programme	Information Technology
Year	2004
Thesis Advisor	Asst. Prof. Dr. Chanboon Sathitwiriyawong

ABSTRACT

Presently, network security system consists of firewall for data filtering transferring between network and intrusion detection system (IDS) for detecting intrusion. Network-based IDS always launches alerts that comprise of many of succeeded and unsucceeded attack. These make troubles for administrator to arrange the priority of protection and intrusion problem solving appropriately. Firewall and IDS always record data into many of log files. Today, analyzing log files data for firewall and IDS have been separated but some part of log data from firewall related to IDS log data. This thesis proposed approach to classified the intrusion alerts by correlating log data from WEB proxy in application layer to log data from network-based IDS. By this way, we will able to get other intrusion information in which important for arranging the priority of intrusion alert for IDS.

กิตติกรรมประกาศ

วิทยานิพนธ์เล่มนี้สำเร็จได้ด้วยความกรุณาจากอาจารย์ที่ปรึกษา ผศ. ดร. จันทร์บูรณ์ สถิตวิริยวงศ์ ที่ได้ให้ความช่วยเหลือ ให้คำชี้แนะและให้คำปรึกษาในการแก้ปัญหาต่าง ๆ ตลอดจนให้ความรู้และประสบการณ์ที่ดีแก่ข้าพเจ้า

ขอขอบพระคุณกรรมการสอบหัวข้อวิทยานิพนธ์และโครงร่างวิทยานิพนธ์ทุกท่าน ที่ได้กรุณาให้คำชี้แนะตลอดจนแนะนำ จนในที่สุดทำให้วิทยานิพนธ์เล่มนี้สำเร็จได้อย่างสมบูรณ์

ขอขอบคุณ ดร. บรรจง หะรังษี, คุณชวลิต ทินกรศูติบุตร และคุณเลอศักดิ์ ลิ้มวิวัฒน์กุล นักวิชาการประจำศูนย์ประสานงานการรักษาความปลอดภัยคอมพิวเตอร์ประเทศไทย (ThaiCERT: Thai Computer Emergency Response Team) เป็นอย่างสูงที่ได้ให้คำแนะนำที่ดีในการปรับปรุงแก้ไขการทดลองประกอบงานวิจัยนี้

ขอขอบคุณเพื่อน ๆ is7 ปกติทุกท่าน ที่ให้กำลังใจในการทำวิทยานิพนธ์ ตลอดจนพี่ ๆ และน้อง ๆ ซึ่งช่วยกระตุ้นให้วิทยานิพนธ์นี้ดำเนินต่อไปได้ด้วยดี

ขอขอบคุณบุคคลอีกหลายท่านที่ข้าพเจ้าไม่ได้เอ่ยชื่อไว้ ที่ท่านได้ให้ความช่วยเหลือ ให้ข้อมูลที่มีประโยชน์ และเป็นกำลังใจให้ข้าพเจ้าในทุก ๆ ส่วนของการทำวิทยานิพนธ์เล่มนี้ จนสำเร็จลุล่วงไปด้วยดี

สำหรับคุณความดีอันใดที่เกิดขึ้นจากวิทยานิพนธ์นี้ ข้าพเจ้าขอมอบให้กับบิดามารดาซึ่งเป็นที่เคารพและรักยิ่ง ตลอดจนครูอาจารย์ที่เคารพรักทุกท่านที่ได้ประสิทธิประสาทวิชาความรู้ สั่งสอนอบรม และถ่ายทอดประสบการณ์ที่ดีให้แก่ข้าพเจ้า

บุญลือ ด้วงสำรวย

สารบัญ

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

สารบัญรูป

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

1.1.1 สาเหตุและลักษณะของปัญหา

ระบบเครือข่ายในปัจจุบันต้องเผชิญกับการบุกรุกและการโจมตีจากผู้ไม่หวังดีอย่างหลีกเลี่ยงมิได้ โดยเฉพาะอย่างยิ่งการโจมตีไปยังเว็บเซิร์ฟเวอร์ซึ่งมักเป็นบริการเป้าหมายอันดับต้น ๆ ของผู้บุกรุกทั้งมืออาชีพและผู้บุกรุกมือสมัครเล่น เนื่องจากจำนวนเป้าหมายซึ่งมีอยู่เป็นจำนวนมากบนเครือข่ายอินเทอร์เน็ตในปัจจุบัน อีกทั้งยังมีความสะดวกในการโจมตีเบื้องต้นโดยอาศัยเพียงเว็บเบราเซอร์ซึ่งหาใช้งานได้ง่ายมากในปัจจุบัน เครื่องมือส่วนใหญ่ที่ใช้ในการต่อกรกับการบุกรุกเหล่านี้ได้แก่ ไฟร์วอลล์ซึ่งใช้ในการกลั่นกรองแพ็กเก็ตข้อมูลให้เป็นไปตามกฎที่กำหนดไว้และ IDS ซึ่งใช้ในการบันทึกพฤติกรรมต่าง ๆ ที่น่าสงสัย ซึ่งหากเป็นการบุกรุกผ่านทางเว็บแล้วนั้นก็จะสามารถเจาะทะลุไฟร์วอลล์เข้ามาในเครือข่ายได้อย่างง่ายดาย ทำให้ภาระทั้งหมดจึงต้องตกไปอยู่ที่ระบบ IDS ที่ต้องตรวจสอบการบุกรุกทางเว็บนี้โดยลำพัง ส่งผลให้ระบบ IDS สร้างการแจ้งเตือนการบุกรุก (alert) ออกมาเป็นจำนวนมาก โดยเฉพาะอย่างยิ่งการแจ้งเตือนการบุกรุกในเหตุการณ์ที่ไม่เกิดขึ้นจริง (unsuccessful attack alert) ซึ่งมักเกิดขึ้นเป็นจำนวนมากอยู่เสมอ การแจ้งเตือนการบุกรุกที่ไม่เกิดขึ้นจริงจำนวนมากเหล่านี้มักเป็นอุปสรรคในการค้นหาการบุกรุกที่แท้จริง ทำให้ความน่าเชื่อถือในการวิเคราะห์การบุกรุกลดลงตามไปด้วย สาเหตุส่วนใหญ่ของ unsuccessful attack alert คือ ในปัจจุบันผู้บุกรุกส่วนมากนิยมใช้เครื่องมือทดสอบข้อบกพร่อง (Vulnerable Scanner) ทำการสำรวจเว็บเซิร์ฟเวอร์เป้าหมายก่อนเสมอ ทำให้ IDS เกิดการแจ้งเตือนการบุกรุกเป็นจำนวนมาก แม้ว่าการบุกรุกดังกล่าวนั้นจะไม่เกิดขึ้นจริงบนเว็บเซิร์ฟเวอร์นั้นก็ตาม ดังนั้นในงานวิจัยนี้จึงได้นำเสนอวิธีการจำแนกประเภทการแจ้งเตือนการบุกรุกของระบบ IDS โดยการนำข้อมูลล็อกไฟล์ของเว็บพร็อกซีมาร่วมพิจารณาประกอบการวิเคราะห์

1.1.2 แนวทางที่น่าสนใจเพื่อใช้แก้ปัญหา

ปัญหาที่พบของในการจำแนกประเภทการแจ้งเตือนการบุกรุก สาเหตุหนึ่งเกิดจากข้อมูลดิบที่นำมาวิเคราะห์ไม่ชัดเจนหรือมีปริมาณไม่เพียงพอ วิธีแก้ไขปัญหา คือ การรวบรวมข้อมูลที่เกี่ยวข้องอันเกิดมาจากเหตุการณ์ซึ่งสงสัยว่าการบุกรุกนั้นให้ได้ปริมาณมากที่สุด เพื่อนำข้อมูลที่มีค่าเหล่านั้นมาหาความสัมพันธ์ระหว่างกันและนำข้อมูลทั้งหมดมาวิเคราะห์ว่า การแจ้งเตือนการบุกรุกนั้นเป็นการแจ้งเตือนการบุกรุกที่ประสบความสำเร็จหรือไม่ ดังนั้นจึงมีงานวิจัยหลาย ๆ งาน

ที่ได้พัฒนาการรวบรวมข้อมูลของเหตุการณ์การบุกรุกให้ได้มากที่สุด เพื่อเป็นข้อมูลประกอบในการวิเคราะห์การบุกรุกให้มีประสิทธิภาพสูงสุด เช่น การแลกเปลี่ยนเหตุการณ์การบุกรุกที่สัมพันธ์กันระหว่างระบบ IDS ด้วยกัน การปรับปรุงเคอร์เนลล์ (kernel) ของระบบเพื่อให้สามารถนำข้อมูลในระดับเน็ตเวิร์กเลเยอร์มาวิเคราะห์ร่วมกับข้อมูลในระดับแอพพลิเคชันเลเยอร์ เป็นต้น ดังนั้นในงานวิจัยนี้จึงได้พัฒนาแนวคิดในการรวบรวมเหตุการณ์การบุกรุกโดยอาศัยข้อมูลที่เกิดขึ้นเป็นปกติตามระบบเครือข่ายทั่ว ๆ ไป คือ การนำข้อมูลเหตุการณ์ที่สงสัยว่าจะเป็นการบุกรุกของระบบ Network-based IDS มาหาความสัมพันธ์กับข้อมูลที่บันทึกลงล็อกไฟล์ของระบบเว็บพร็อกซีซึ่งเป็นไฟร์วอลล์ในระดับแอพพลิเคชันชนิดหนึ่ง เมื่อเชื่อมโยงความสัมพันธ์ระหว่างเหตุการณ์ที่เกิดขึ้นระหว่าง Network-based IDS และเว็บพร็อกซีได้ ก็จะทำให้ระบบสามารถวิเคราะห์เพื่อจำแนกการแจ้งเตือนการบุกรุกที่มีแนวโน้มว่าจะเป็นการแจ้งเตือนการบุกรุกที่ไม่ประสบความสำเร็จได้

1.1.3 บทสรุป

จากปัญหาต่าง ๆ ในระบบการวิเคราะห์การบุกรุกบนเครือข่ายที่ได้กล่าวมาข้างต้นและแนวทางการแก้ปัญหาที่ได้นำเสนอไป จึงทำให้เกิดความสนใจในการพยายามที่จะทำการวิจัยเพื่อปรับปรุงการบุกรุกบนระบบเครือข่ายโดยเฉพาะอย่างยิ่งการลด false alert ซึ่งเกิดขึ้นเป็นจำนวนมากบนระบบการวิเคราะห์การบุกรุกในปัจจุบัน ซึ่งหนทางหนึ่งที่สามารถนำมาแก้ปัญหานี้ได้ก็คือการรวบรวมข้อมูลที่เกี่ยวข้องกับเหตุการณ์ที่สงสัยว่าจะเป็นการบุกรุกให้ได้มากที่สุด แล้วจึงเรียบเรียงข้อมูลทั้งหมดไปทำการวิเคราะห์ต่อไป เมื่อวิเคราะห์ว่าเหตุการณ์ใดมีแนวโน้มเป็นการบุกรุกที่ไม่ประสบความสำเร็จหรือไม่มีเจตนาเป็นการบุกรุกที่แท้จริง ก็จะกำหนดให้จำแนกการแจ้งเตือนนั้นเป็นการแจ้งเตือนการบุกรุกที่ไม่ประสบความสำเร็จ และนำการแจ้งเตือนส่วนที่เหลือไปวิเคราะห์เพื่อหาการบุกรุกที่แท้จริงต่อไป

ในส่วนต่าง ๆ ของวิทยานิพนธ์เล่มนี้ จะใช้คำว่า squid log แทนการเรียกดูข้อมูลเว็บผ่านเว็บพร็อกซีไปยังเว็บเซิร์ฟเวอร์ และเมื่อกล่าวถึงการแจ้งเตือน (alert) หรือ snort alert จะหมายถึงข้อมูลจากไฟล์ Network-based IDS ที่สงสัยเหตุการณ์ใดว่ามีแนวโน้มเป็นการบุกรุกก็จะบันทึกเป็นการแจ้งเตือน โดยจะใช้คำว่า successful attack alert แทนการแจ้งเตือนการบุกรุกที่ประสบความสำเร็จ, unsuccessful attack alert แทนการแจ้งเตือนในการบุกรุกที่ไม่ประสบความสำเร็จ ซึ่งอาจเกิดมาจากสาเหตุการใช้เครื่องมือตรวจสอบข้อบกพร่องหรือการสุ่มทดลองเจาะระบบจากผู้บุกรุก และ undecided attack alert แทนการแจ้งเตือนการบุกรุกที่ไม่สามารถวิเคราะห์ได้ว่าเป็นการบุกรุกที่แท้จริงหรือไม่

ลดจำนวน alert ลงไปได้อย่างเห็นได้ชัด ในงานวิจัย [7] ได้นำเสนอวิธีการปรับปรุงเว็บแอพพลิเคชันทางฝั่งเซิร์ฟเวอร์ให้มีความสามารถในการจัดการกับการบุกรุกได้ในทันที และในทางกลับกันงานวิจัย [10] ก็นำเสนอวิธีการปรับปรุงระบบให้สามารถเตรียมข้อมูลในระดับเน็ตเวิร์กเลเยอร์เพื่อไปเสริมข้อมูลในการวิเคราะห์การบุกรุกบนเครื่องนั้น ๆ ได้ จากผลลัพธ์ของหลายงานวิจัยให้ผลเป็นที่น่าพอใจ ประกอบกับรูปแบบหนึ่งของระบบรักษาความปลอดภัยบนเว็บเซิร์ฟเวอร์ในปัจจุบันนิยมติดตั้งพร็อกซีเซิร์ฟเวอร์ไว้เป็นด่านแรกของเครือข่าย โดยทำหน้าที่เสมือนเป็นไฟร์วอลล์คั่นกลางระหว่างเครือข่ายภายในและเครือข่ายภายนอก นอกจากนี้ยังทำหน้าที่เป็นแคชเซิร์ฟเวอร์เพื่อช่วยเสริมประสิทธิภาพในการเรียกดูเว็บไซต์ที่ซ้ำๆกัน ส่วนภายในเครือข่ายเองก็จะติดตั้ง Network- based IDS เพื่อตรวจสอบแพ็กเก็ตข้อมูลที่น่าสงสัยที่เข้ามายังเว็บเซิร์ฟเวอร์ ซึ่งทั้งเว็บพร็อกซีและ Network-based IDS ต่างก็บันทึกเหตุการณ์ต่าง ๆ ลงล็อกไฟล์เป็นปกติอยู่แล้ว จึงทำให้เกิดแนวคิดในการนำข้อมูลในระดับแอพพลิเคชันเลเยอร์จากเว็บพร็อกซีมาใช้ในการวิเคราะห์ร่วมกันการแจ้งเตือนของ Network-based IDS ซึ่งสามารถตรวจสอบข้อมูลได้ถูกต้องในระดับเน็ตเวิร์กเลเยอร์เท่านั้น ทำให้ข้อมูลที่ใช้ในการวิเคราะห์การบุกรุกมีความชัดเจน ครอบคลุมข้อมูลได้หลายระดับมากยิ่งขึ้น ส่งผลให้การวิเคราะห์การบุกรุกมีความถูกต้องและรวดเร็วมากยิ่งขึ้น

1.4 ขอบเขตการวิจัย

ในการศึกษาหลักการทำงานของระบบการวิเคราะห์การบุกรุกในปัจจุบันนั้นจะทำการศึกษาโดยใช้โปรแกรม snort ซึ่งเป็น Network-based IDS แบบเปิดเผยแหล่งกำเนิด (open source) ที่ได้รับความนิยมสูงสุดเป็นหลัก โดยจะศึกษารายละเอียดในขั้นตอนการปรับแต่งรูปแบบการบุกรุก (signature) การกำหนดประเภทของการบุกรุก (classification) การกำหนดระดับความรุนแรง (priority) รวมถึงการนำล็อกไฟล์ของ snort ไปวิเคราะห์เพื่อค้นหาการบุกรุกที่แท้จริง เนื่องจากในการวิจัยนี้เป็นการวิเคราะห์โดยนำล็อกไฟล์จากเว็บพร็อกซีมาวิเคราะห์ร่วม ดังนั้นในระบบเครือข่ายนั้นจำเป็นที่จะต้องมีการกำหนดให้การเรียกใช้บริการเว็บจำเป็นที่จะต้องเรียกผ่านเว็บพร็อกซีเสมอ อาจใช้วิธีการกำหนดเว็บพร็อกซีที่เว็บเบราเซอร์โดยตรง (direct proxy) หรือกำหนดให้ที่เครื่องเกตเวย์ทำการบังคับการเข้าถึงบริการเว็บที่พอร์ต 80 ของเว็บเซิร์ฟเวอร์ใด ๆ ให้เปลี่ยนไปยังบริการพร็อกซีที่พอร์ต 3128 ของเว็บพร็อกซีเซิร์ฟเวอร์โดยอัตโนมัติ (transparent proxy) จึงจะสามารถกำหนดได้ว่าเหตุการณ์ที่เป็นการบุกรุกไปยังเว็บเป้าหมายจะต้องมีบันทึกในล็อกไฟล์ของเว็บพร็อกซีเสมอ เนื่องจากข้อมูลที่บันทึกจากล็อกไฟล์เว็บพร็อกซีเซิร์ฟเวอร์ในปัจจุบัน (squid) สามารถจัดการกับเหตุการณ์ที่เกี่ยวข้องกับบริการเว็บเท่านั้น ดังนั้นทำให้ผลการวิเคราะห์การบุกรุกในงานวิจัยนี้จึงจำกัดเพียงการบุกรุกบนเว็บด้วยเช่นกัน โดยการจำแนกประเภทการแจ้งเตือนการบุกรุกนี้จะเห็นผลได้ชัดเจนเมื่อผู้บุกรุกบนเว็บใช้เครื่องมือตรวจสอบข้อบกพร่อง (Vulnerable

Scanner) ทำการสำรวจเว็บเซิร์ฟเวอร์เป้าหมายหรือใช้รูปแบบการโจมตีที่ถูกกำหนดไว้แล้วในฐานข้อมูลรูปแบบการบุกรุกของ snort เท่านั้น โดยในการวิเคราะห์การบุกรุกไม่สามารถจัดการกับการบุกรกที่ใช้เทคนิคความชำนาญขั้นสูงได้ เช่น การปลอมแปลงเบอร์ไอพีต้นทาง (fake IP) รวมถึงไม่สามารถจัดการกับกลุ่มการบุกรุกที่ใช้เบอร์ไอพีที่มีการแบ่งจำนวนเครื่อง (Subnet Mask) ที่นอกเหนือมาตรฐาน Class A,B,C ได้

1.5 ขั้นตอนของการศึกษา

1. ศึกษาทฤษฎีและหลักการทำงานของระบบเว็บพร็อกซี
2. ศึกษาทฤษฎีและหลักการทำงานของระบบ Network-based IDS
3. ศึกษาการวิเคราะห์การบุกรุกโดยใช้ล็อกไฟล์
4. ศึกษาการบุกรุกบนเว็บในปัจจุบัน
5. ศึกษาบทความและงานวิจัยต่าง ๆ ที่มีความเกี่ยวข้องกับการวิเคราะห์การบุกรุกโดยใช้ล็อกที่แตกต่างกัน
6. ศึกษาและจัดหาอุปกรณ์เครื่องมือและวิธีการที่เหมาะสมที่จะใช้ในงานวิจัย
7. ออกแบบการทดลอง
8. ทดลองการบุกรุกบนเว็บกับระบบการวิเคราะห์ในปัจจุบัน
9. เขียนโปรแกรมเพื่อเชื่อมความสัมพันธ์ระหว่างล็อก snort และล็อก squid
10. เขียนโปรแกรมเพื่อวิเคราะห์การบุกรุกโดยข้อมูลระหว่างสองล็อกที่สัมพันธ์กัน
11. ทำการทดลอง เปรียบเทียบและวัดผล
12. สรุปผลการดำเนินการและจัดทำเอกสารประกอบวิทยานิพนธ์

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงการบุกรุกบนระบบเครือข่ายโดยเฉพาะเครือข่ายที่ให้บริการทางด้านเว็บ หลักการทำงานของระบบรักษาความปลอดภัยบนเครือข่ายที่ใช้ระบบไฟร์วอลล์และ IDS ร่วมกัน, หลักการทำงานของแอพพลิเคชันพร็อกซีไฟร์วอลล์ที่ใช้งานวิจัยนี้ คือ squid ซึ่งจะอธิบายถึงส่วนประกอบต่าง ๆ และรูปแบบการบันทึกล็อกไฟล์ นอกจากนี้ยังกล่าวถึง snort ซึ่งเป็นโปรแกรม Network-based IDS ที่ใช้งานวิจัยนี้ ซึ่งจะอธิบายถึงส่วนประกอบต่าง ๆ หลักการทำงาน การกำหนดรูปแบบการบุกรุกและรูปแบบการบันทึกล็อกไฟล์ ตลอดจนการนำล็อก snort มาวิเคราะห์หาการบุกรุก โดยในท้ายที่สุดจะกล่าวถึงงานวิจัยอื่น ๆ ที่เกี่ยวข้องกับงานวิจัยนี้

2.1 ระบบรักษาความปลอดภัยบนเครือข่าย (Network Security)

การใช้งานอินเทอร์เน็ตในปัจจุบันได้เจริญเติบโตสูงขึ้นอย่างรวดเร็ว โดยเฉพาะอย่างยิ่งบริการเผยแพร่ข้อมูลข่าวสารบนเครือข่ายใยแมงมุม (World Wide Web) เพราะทุกคนสามารถเข้าถึงได้อย่างรวดเร็วและสะดวกที่สุดนั่นเอง แต่เนื่องจากอินเทอร์เน็ตเป็นสังคมเปิดขนาดใหญ่ทำให้ยากแก่การควบคุมการใช้งานให้อยู่ในกฎเกณฑ์ที่กำหนด ดังนั้นจึงมีผู้ที่ไม่ประสงค์ดีพยายามฝ่าฝืนและหาผลประโยชน์โดยไม่ได้รับอนุญาติจากระบบอินเทอร์เน็ตนี้อยู่เสมอ ทำให้แต่ละเครือข่ายที่ให้บริการข้อมูลจำเป็นที่จะต้องมีมาตรการในการรักษาความปลอดภัยของเครือข่ายตนเอง โครงสร้างระบบเครือข่ายที่ปลอดภัยในปัจจุบันจะประกอบไปด้วยเครือข่าย 3 ส่วนหลัก ๆ คือ เครือข่ายภายใน (Trusted Zone) สำหรับผู้ใช้งานภายในองค์กร เครือข่ายในเขตปลอดทหาร (De Militarized Zone-DMZ) สำหรับเครื่องเซิร์ฟเวอร์ซึ่งให้บริการข้อมูลแก่ผู้ใช้ภายนอกและเครือข่ายภายนอก (Untrusted Zone) สำหรับผู้ใช้อินเทอร์เน็ตทั่ว ๆ ไป ปกติไฟร์วอลล์จะถูกติดตั้งระหว่างทางเชื่อมต่อระหว่างเครือข่ายเพื่อกลั่นกรองแพ็กเก็ตข้อมูลที่รับส่งระหว่างเครือข่ายให้เป็นไปตามกฎ (Access Rules) ที่กำหนดไว้ และติดตั้ง IDS ไว้ภายในเครือข่ายเพื่อบันทึกพฤติกรรมการใช้งานเครือข่ายต่าง ๆ ที่มีแนวโน้มไปในทางมิชอบ เมื่อมีผู้ใช้งานต้องการใช้บริการข้อมูลเครือข่ายภายในของเราจากภายนอก จะต้องผ่านทางไฟร์วอลล์ก่อนเสมอ ซึ่งหากเป็นการบุกรุกหรือใช้งานบริการที่นอกเหนือจากที่เรากำหนดไว้ในกฎ ก็จะถูกไฟร์วอลล์ทำการบล็อก (Block) การเรียกใช้บริการนั้นไว้ได้ แต่ถ้าเป็นการบุกรุกโดยอาศัยข้อบกพร่อง (Vulnerability) ของบางบริการที่อยู่ภายในเครือข่ายของเรา การบุกรุกนั้นก็จะสามารถผ่านไฟร์วอลล์เข้ามาได้ แต่การเรียกใช้บริการใด ๆ ที่เกิดขึ้นภายในเครือข่ายที่มีแนวโน้มเป็นการกระทำที่ไม่สมควรก็จะถูกระบบ IDS บันทึกลงล็อกไฟล์ไว้สำหรับตรวจสอบต่อไป

อุปสรรคสำคัญที่ส่งผลต่อระบบรักษาความปลอดภัยบนเครือข่าย ได้แก่

- ไวรัสและหนอนคอมพิวเตอร์ (Virus/Worm) คือ โปรแกรมที่ทำงานโดยอัตโนมัติซึ่งมักก่อให้เกิดแพ็กเก็ตจำนวนมากในระบบเครือข่าย ทำให้ระบบเครือข่ายเกิดความล่าช้าในการรับส่งข้อมูล บางครั้งอาจทำให้เครือข่ายล่มไม่สามารถให้บริการข้อมูลได้ เช่น Sircam, Blaster เป็นต้น
- ม้าโทรจันและแบล็คดอร์ (Trojan House/Backdoor) คือ การเปิดช่องทางให้ผู้อื่นสามารถเข้าถึงข้อมูลโดยไม่ได้รับอนุญาต เช่น Back Orifice, Beast เป็นต้น
- จุดอ่อนของระบบปฏิบัติการ (OS Vulnerability) คือ ข้อบกพร่องของระบบปฏิบัติการเองที่เปิดโอกาสผู้ไม่หวังดีสามารถควบคุมการทำงานของเครื่องได้ เช่น Winnuke, DCOM RPC เป็นต้น
- จุดอ่อนของโปรโตคอล (Protocol Vulnerability) คือ ข้อบกพร่องของโปรโตคอลที่ใช้ในการสื่อสารข้อมูล ซึ่งอาจมีผลทำให้เกิดการโจมตีจากผู้ไม่หวังดีทำให้ระบบเครือข่ายไม่สามารถทำงานได้ตามปกติ เช่น SYN Flooding, Teardrop เป็นต้น
- จุดอ่อนของโปรแกรม (Application Vulnerability) คือ ข้อบกพร่องของโปรแกรมที่ให้บริการต่าง ๆ ซึ่งอาจทำให้ผู้ไม่หวังดีสามารถเข้าถึงข้อมูลที่ไม่ได้รับอนุญาต เช่น Unicode, Frontpage Ext เป็นต้น
- แฮกเกอร์ (Hacker/Cracker) คือ บุคคลผู้ไม่ประสงค์ดีซึ่งมักก่อให้เกิดความเสียหายแก่ระบบเครือข่ายอยู่เสมอ เช่น การโจมตีด้วย DDOS (Distributed Denial of Service) เพื่อให้เครือข่ายหยุดทำงาน การโจมตีด้วย Buffer Overflow เพื่อให้ได้รับข้อมูลที่เป็นความลับ เป็นต้น

จากอุปสรรคดังกล่าวจะเห็นได้ว่า อุปสรรคบางอย่างสามารถทำการแก้ไขได้โดยการปรับปรุงซอฟต์แวร์ (patch) หรือใช้ซอฟต์แวร์ช่วยในการตรวจสอบและป้องกัน แต่อุปสรรคที่สำคัญที่สุด คือ แฮกเกอร์หรือมนุษย์นั่นเอง เนื่องจากมีการพัฒนาการบุกรุก การหลบเลี่ยงการตรวจสอบอยู่ตลอดเวลา ทำให้เป็นการยากที่จะมีระบบรักษาความปลอดภัยใดที่สามารถรอดพ้นจากการคุกคามของแฮกเกอร์ได้ 100%

2.2 การบุกรุกบนระบบเครือข่าย

จากข้อบกพร่องที่มีอยู่ในระบบซึ่งผู้ดูแลระบบอาจจะยังไม่ได้แก้ไขหรือข้อบกพร่องซึ่งถูกค้นพบล่าสุดซึ่งยังไม่มีวิธีการแก้ไข อาจเป็นช่องทางให้คนกลุ่มหนึ่งพยายามแสวงหาผลประโยชน์อันมิชอบจากข้อบกพร่องดังกล่าว โดยการพยายามโจมตีหรือบุกรุกไปยังระบบที่มีจุดอ่อนนั้น โดย

เฉพาะระบบที่ให้บริการเว็บซึ่งแพร่หลายมากในระบบอินเทอร์เน็ตในปัจจุบัน การบุกรุกในปัจจุบันสามารถกระทำได้อย่างง่ายดาย โดยเฉพาะการบุกรุกบนเว็บ บางครั้งสามารถใช้เพียงเว็บเบราเซอร์ธรรมดากับความรู้ของข้อบกพร่องเพียงนิดหน่อยก็สามารถเข้าถึงข้อมูลทั้งหมดของเว็บเซิร์ฟเวอร์ได้ เช่น ในกรณีของบัก (bug) Unicode ของเว็บเซิร์ฟเวอร์ที่ใช้ IIS (Internet Information Services) บนระบบปฏิบัติการ Windows NT 4.0 หรือ Windows 2000 ซึ่งยังไม่ได้อัพเดตซอฟต์แวร์ IIS เพื่อปิดข้อบกพร่องดังกล่าว ผู้บุกรุกสามารถใช้เพียงเบราเซอร์ทั่ว ๆ ไปร้องขอข้อมูลที่เป็น Unicode ผ่าน URL ไปยังเว็บเซิร์ฟเวอร์ก็สามารถเรียกใช้คำสั่งบนเว็บเซิร์ฟเวอร์นั้นได้ รูปแบบคำสั่ง Unicode มีลักษณะดังรูปที่ 2.1

```
/scripts/..%255c..
/_vti_bin/..%255c..%255c..%255c..%255c..%255c..
/cgi-bin/..%255c..%255c..%255c..%255c..%255c..
/iisadmpwd/..%255c..%255c..%255c..%255c..%255c..
/_vti_cnf/..%255c..%255c..%255c..%255c..%255c..
/msadc/..%255c..%255c..%255c..
```

รูปที่ 2.1 รูปแบบข้อบกพร่อง Unicode ของ IIS

คำสั่งที่นิยมใช้ผสมกับบัก Unicode เพื่อบุกรุกระบบเว็บเซิร์ฟเวอร์ ได้แก่ cmd.exe และ net.exe ซึ่งสามารถกระทำการเรียกดูข้อมูลในไฟล์สำคัญ ๆ หรือเพิ่มยูสเซอร์ระบบโดยไม่ต้องใช้เครื่องมือเพิ่มเติมใด ๆ เลย ซึ่งจากข้อบกพร่องดังกล่าว ผู้ดูแลระบบสามารถแก้ไขได้โดยปิดบริการเว็บถ้าไม่จำเป็นต้องให้บริการหรือปรับปรุงโดยอัตเดตซอร์ฟแวร์ IIS เป็นเวอร์ชันใหม่ มิฉะนั้นต้องเลี่ยงไปใช้ซอฟต์แวร์เว็บเซิร์ฟเวอร์อื่น ๆ เช่น Apache เป็นต้น

นอกจากนี้แฮกเกอร์ที่มีความชำนาญยังใช้เทคนิคอื่น ๆ ประกอบอีกมากมายเพื่อรวบรวมข้อมูลให้ได้มากที่สุดก่อนพิจารณาเลือกใช้วิธีการโจมตีหรือการบุกรุกที่เหมาะสมต่อไป ขั้นตอนการบุกรุกแสดงเป็นลำดับดังต่อไปนี้ ได้แก่

- การแกะรอย (Foot printing) คือ การค้นหาเบอร์ไอพี (IP Address) ชื่อโดเมน (Domain name) และรายละเอียดอื่น ๆ ของบุคคลหรือองค์เป้าหมาย โดยใช้เทคนิคต่าง ๆ เช่น การใช้ Search Engine โดยทั่ว ๆ ไปค้นหาข้อมูลหรือข้อความที่เกี่ยงข้องทั้งหมดที่ปรากฏบนอินเทอร์เน็ต การใช้ Whois เพื่อแสดงรายละเอียดตามที่บุคคลหรือองค์กรนั้น ๆ ได้ลงทะเบียนไว้ การ DNS zone transfer เพื่อแสดงรายละเอียดเกี่ยวกับการแม็พ DNS และไอพี

- การสแกน (Scanning) คือ การสำรวจบริการบนเครื่องเป้าหมาย เพื่อตรวจหาประเภทของระบบปฏิบัติการที่ใช้และเพื่อแยกแยะบริการที่เปิดและปิดออกจากกัน โดยใช้โปรแกรมช่วย เช่น การสแกนพอร์ตโดย nmap, Superscan เป็นต้น
- การค้นหาและรวบรวมรายละเอียดต่าง ๆ (Enumeration) คือ การตรวจสอบบริการว่ามีเวอร์ชันเท่าไหร่ มีข้อบกพร่องใดบ้างที่ยังไม่ได้อัพเดทแพทช์ เพื่อมุ่งประเด็นการบุกรุกไปยังบริการที่มีจุดอ่อนมากที่สุดก่อนเสมอ การแยกแยะหาแอคเคานต์ที่มีสิทธิ์ในเครื่องเซิร์ฟเวอร์สูง แต่มีระดับการป้องกันที่ต่ำ รวมถึงทรัพยากรระบบอื่น ๆ ที่ไม่ได้รับการป้องกันไว้เพียงพอ โดยอาจใช้เทคนิคต่าง ๆ เช่น การอ่านแบนเนอร์โดยใช้ telnet, netcat เป็นต้น
- การได้รับสิทธิ์ในการเข้าถึงระบบ (Gaining Access) คือ การพยายามนำข้อมูลที่รวบรวมได้ทั้งหมดมาปลอมแปลงเป็นบุคคลอื่นที่มีสิทธิ์ในระบบนั้น ๆ โดยใช้เทคนิคต่าง ๆ เช่น การดักจับรหัสผ่านบนเน็ตเวิร์กโดยใช้ Sniffer และ tcpdump การใช้เทคนิคการเดารหัสผ่านแบบ Brute Force โดยใช้ Burtus การขโมยไฟล์เก็บรหัสผ่านโดยใช้ pwdump2 และการใช้เทคนิคบัฟเฟอร์โอเวอร์โฟลว์ (Buffer Overflow) โดยใช้ Ttdb เป็นต้น
- การยกระดับสิทธิให้เท่าเทียมผู้ดูแลระบบ (Escalating privilege) คือ การยกระดับสิทธิจากผู้ใช้ทั่วไปมาเป็นสิทธิของผู้ดูแลระบบซึ่งสามารถควบคุมระบบได้ทั้งหมด โดยใช้เทคนิคการแคร็กรหัสผ่าน โดยใช้ John หรือฉวยโอกาสใช้ช่วงโหว่ที่มีอยู่จำลองผู้ใช้ธรรมดาเป็นผู้ดูแลระบบได้ โดยใช้ getadmin เป็นต้น
- การขโมยข้อมูลเพิ่มเติม (Pilfering) คือ กระบวนการในการรวบรวมข้อมูลตั้งแต่เริ่มต้นอีกครั้ง เพื่อความหากลไกในการเข้าถึงระบบอื่น ๆ ที่ระบบปัจจุบันให้ความเชื่อถืออยู่ โดยใช้เทคนิคการประเมินความเชื่อถือระหว่างกันและการค้นหารหัสผ่านที่อยู่ในรูปแบบเคลียร์เท็กซ์
- การปิดบังอำพรางตัว (Covering tracks) คือ การพยายามให้ทำให้รอดพ้นจากผู้ดูแลระบบตัวจริง ในกรณีที่สามารถเข้าถึงสิทธิในระดับผู้ดูแลระบบนั้นได้แล้ว โดยใช้เทคนิคการลบล็อกไฟล์หรือซ่อนเครื่องมือ
- การสร้างประตูทางลับไว้ (Creating back doors) คือ การสร้างเส้นทางเข้าออกระบบสำรองไว้ ใช้ในกรณีต้องการเข้าถึงเครื่องเป้าหมายได้โดยง่ายในภายหลัง ซึ่งสามารถกระทำได้หลายเทคนิค คือ การสร้างแอคเคานต์ปลอม การตั้งเวลาให้แบตช์จ็อบทำงานเบื้องหลังโดยอัตโนมัติ การฝังโปรแกรมไว้กับส่วนสตาร์ทอัพ การฝังโปรแกรม

เป็นรีโมตคอนโทรล การติดตั้งกลไกการแอบมอนิเตอร์และการแทนที่โปรแกรมตามปกติด้วยโปรแกรมโทรจัน

- การทำให้เซิร์ฟเวอร์ปฏิเสธการให้บริการ (Denial of Service) คือ การพยายามทำให้เครื่องเซิร์ฟเวอร์นั้นหยุดบริการ อาจมีเหตุผลเพื่อต้องการให้เซิร์ฟเวอร์ต้องรีเซ็ตระบบเครื่องใหม่ทำให้โปรแกรมที่แฮ็กเกอร์ฝังไว้เริ่มทำงาน หรือเพื่อทำให้เซิร์ฟเวอร์หยุดให้บริการ เพื่อแฮ็กเกอร์จะได้ใช้เครื่องตนเองให้บริการแทนเบอร์ไอพีเป้าหมาย (fake IP) และในบางครั้งอาจเกิดจากสาเหตุที่แฮ็กเกอร์ไม่สามารถเข้าถึงสิทธิ์ในระดับผู้ดูแลระบบของเครื่องนั้น ๆ ได้ เทคนิคที่ใช้เพื่อจุดหมายดังกล่าว เช่น SYN flood, Out of bounds TCP option (OOB) เป็นต้น

เนื่องจากการบุกรุกบนเว็บโดยใช้เพียงเว็บเบราเซอร์ธรรมดาเพื่อนำไปสู่การได้สิทธิ์ในระดับผู้ดูแลระบบนั้น จะต้องอาศัยช่องทางที่ข้อมูลจะเข้าไปถึงเว็บเซิร์ฟเวอร์ ช่องทางที่กล่าวถึงนี้ คือ Uniform Resource Locator (URL) ซึ่งได้กำหนดไว้ใน RFCs 1808 และ 2396

URL คือกลไกหรือวิธีการในการระบุถึงทรัพยากรที่ต้องการบนเว็บหรือทรัพยากรอื่น ๆ ในระดับแอพพลิเคชันเลเยอร์ ซึ่ง URL จะถูกส่งเว็บเบราเซอร์ไปยังเว็บเซิร์ฟเวอร์เพื่อให้เว็บเซิร์ฟเวอร์ปฏิบัติตามไฟล์ที่ถูกร้องขอไป โครงสร้างพื้นฐานของ URL แสดงดังรูปที่ 2.2

```
protocol://server/directory/file?parameters
```

รูปที่ 2.2 โครงสร้างพื้นฐานของ URL

แต่ละองค์ปะกอบสามารถอธิบายได้ดังนี้

- protocol คือ โปรโตคอลในระดับแอพพลิเคชันเลเยอร์ มีโปรโตคอลพื้นฐาน คือ http ส่วนโปรโตคอลอื่น คือ https, ftp และอื่น ๆ ขึ้นอยู่กับว่าเว็บเบราเซอร์จะสนับสนุนหรือไม่
- server คือ ชื่อเครื่องในรูปแบบ DNS หรือเบอร์ไอพีของเครื่องที่เก็บทรัพยากรที่กำลังถูกร้องขออยู่
- directory คือ ห้องที่ใช้เก็บทรัพยากรนั้น ซึ่งอาจจะมีหรือไม่มีก็ได้ และถ้ามีการมีการซ้อนกันของห้องได้ไม่จำกัดจำนวนห้อง
- file คือ ชื่อทรัพยากรที่ต้องการ ทรัพยากรที่ต้องการอาจเป็นได้ทั้งไฟล์สแตติกและไฟล์ที่ถูกประมวลผลที่เซิร์ฟเวอร์แล้วนำผลลัพธ์มาแสดงผล
- parameters คือ ข้อมูลเสริมเพิ่มเติมที่ถูกส่งไปพร้อมกับ file เพื่อประกอบในการให้ข้อมูลเพิ่มเติม มักใช้ในกรณีทรัพยากรนั้นเป็นการประมวลผลที่เซิร์ฟเวอร์

ซึ่งการส่งผ่าน parameter นี้ไปกับ URL นี้ ทางฝั่งเว็บเซิร์ฟเวอร์จำเป็นที่มีกรรมวิธีในการจัดการข้อมูล parameter ในส่วนนี้ เช่น การโต้ตอบแบบ Common Gateway Interface (CGI) เป็นต้น

2.3 ระบบไฟร์วอลล์ (Firewall)

ไฟร์วอลล์ทำหน้าที่ในการกรองแพ็กเก็ตให้เป็นไปตามกฎที่ตั้งไว้ ซึ่งหากมีแพ็กเก็ตที่นอกเหนือจากกฎฝ่าฝืนเข้ามาก็จะถูกไฟล์วอลล์บล็อกและบันทึกค่าลงล็อกไฟล์เอาไว้ แต่เนื่องจากไฟร์วอลล์ต้องตรวจสอบแพ็กเก็ตทุกแพ็กเก็ตที่รับส่งไปมาระหว่างเครือข่ายทั้งหมด ทำให้โอกาสที่ข้อมูลในล็อกไฟล์ก็จะมีมากตามไปด้วย ข้อมูลล็อกไฟร์วอลล์จะเป็นค่าตัวเลขและมีจำนวนซ้ำ ๆ กันเป็นจำนวนมากซึ่งเป็นการยากที่จะทำความเข้าใจ เป็นเหตุให้ผู้ดูแลระบบ (System Admin) ส่วนมากละเลยในการตรวจสอบล็อกไฟร์วอลล์ บางครั้งถึงกับยกเลิกการบันทึกล็อกไปเลยก็มี การบันทึกล็อกของไฟร์วอลล์จะมีรูปแบบที่แตกต่างกันอีกไปตามแต่ประเภทหรือผู้ผลิตแต่ละยี่ห้อ

2.3.1 ประเภทของไฟร์วอลล์

ไฟร์วอลล์คือเครื่องมือที่ใช้ในการป้องกันไม่ให้ข้อมูลที่ไม่สมควรผ่านเข้าออกระหว่างเครือข่าย เราสามารถแบ่งประเภทไฟร์วอลล์ตามลักษณะการทำงานได้ดังนี้ คือ

Packet Filtering Firewall

คือไฟร์วอลล์ที่ทำหน้าที่ควบคุมการรับส่งข้อมูลระหว่างเครือข่าย โดยการพิจารณาตรวจสอบข้อมูลที่ปรากฏอยู่ในแพ็คเก็ตให้เป็นไปตามกฎที่เรากำหนดไว้ ซึ่งข้อมูลที่ไฟร์วอลล์ชนิดนี้สามารถตรวจสอบได้ คือ Source IP, Source Port, Destination IP, Destination Port และข้อมูลประกอบของโปรโตคอล เช่น TCP Flag, ICMP Massage, MSS ซึ่งล้วนแต่เป็นข้อมูลในระดับเลเยอร์ต่ำกว่าแอพพลิเคชันเลเยอร์ทั้งสิ้น ทำให้การทำงานไม่จำเป็นต้องใช้อุปกรณ์ที่มีความสามารถในการประมวลผลสูง จึงมักนิยมนำคุณสมบัติชนิดนี้ใส่ไว้ในอุปกรณ์เราเตอร์ซึ่งเป็นตัวคั่นกลางระหว่างจุดต่อของเครือข่าย ข้อเสียที่เห็นได้ชัดเจนของไฟร์วอลล์ชนิดนี้ได้แก่ การที่จะต้องเปิดพอร์ตที่มีค่ามากกว่า 1024 เอาไว้เสมอ เพื่อรองรับการบริการบางอย่างที่ต้องการใช้พอร์ตเพิ่มเติมซึ่งมันไม่สามารถคาดเดาได้ล่วงหน้า ทำให้อาจเป็นช่องโหว่ให้ผู้บุกรุกใช้ในการโจมตีได้

Stateful Inspection Firewall

คือไฟร์วอลล์ที่พัฒนามาจาก Packet Filtering Firewall ซึ่งนอกจากจะสามารถควบคุมแพ็กเก็ตข้อมูลที่รับส่งระหว่างเครือข่ายได้แล้ว ยังสามารถในการวิเคราะห์และรับรู้ความต่อเนื่องของแพ็กเก็ตข้อมูลได้ด้วย เพราะไฟร์วอลล์ชนิดนี้สามารถจดจำสถานะของการสื่อสาร (state) เพื่อ

นำไปวิเคราะห์ว่าการสื่อสารนั้น ๆ เกิดจากการสื่อสารตามปกติที่ถูกต้องหรือไม่ (TCP-3 ways handshack) และเมื่อสามารถจดจำและติดตามการเชื่อมต่อได้ ทำให้มันสามารถรู้ได้ว่าบริการชนิดใดที่จำเป็นที่จะต้องมีการเปิดพอร์ตเพิ่มเติมในระหว่างการทำงาน ทำให้มันสามารถปิดพอร์ตที่มีค่ามากกว่า 1024 ในตามปกติได้ โดยจะเปิดใช้เป็นบางพอร์ตในกรณีที่มีการร้องขอมาจากบริการที่ได้รับอนุญาตเท่านั้น และจะปิดพอร์ตเหล่านั้นเมื่อสิ้นสุดการเชื่อมต่อ

Application Proxy Firewall

คือไฟร์วอลล์ที่ทำหน้าเป็นตัวกลางในการเชื่อมต่อ เพื่อป้องกันไม่ให้เกิดการเชื่อมต่อโดยตรงระหว่างสองเครือข่าย โดยจะทำงานในระดับแอพพลิเคชันเลเยอร์ทำให้สามารถตรวจสอบถึงข้อมูลในระดับเนื้อหาได้ (Content Filter) ทำให้มีระดับความปลอดภัยมากกว่าสองประเภทแรก แต่ทำงานได้ช้ากว่าเนื่องจากจะต้องมีขั้นตอนการทำงานที่มากกว่านั่นเอง เราสามารถเพิ่มเติมหน้าที่การทำงานของไฟร์วอลล์ชนิดนี้ได้แก่ การทำเว็บพร็อกซี่ที่สามารถควบคุมเว็บไซต์ที่ไม่เหมาะสม การทำแคชข้อมูลเพื่อลดปริมาณการสื่อสารข้อมูลที่มีการเรียกใช้บ่อย ๆ การลดการแพร่กระจายของไวรัสบางชนิดที่อาศัยช่องโหว่ของ URL (cmd.exe , default.ida) การตรวจสอบสิทธิ์ผู้ใช้งาน (User Authenticate) ข้อเสียที่สำคัญของไฟร์วอลล์ชนิดนี้ คือ มีบางแอพพลิเคชันเท่านั้นที่สามารถเชื่อมต่อผ่าน Proxy ได้ เช่น HTTP,FTP

2.3.2 ความสามารถของไฟร์วอลล์

1. สามารถควบคุมการเข้าออกของข้อมูล (Access Control) เช่น การจำกัดข้อมูลขาเข้าและขาออก (Inbound Access, Outbound Access)
2. สามารถจำกัดการตรวจสอบข้อมูลภายในเครือข่าย (Network Scanning, Host Scanning) โดยการกำหนดให้เปิดให้เข้าถึงพอร์ตเฉพาะบริการที่กำหนดเท่านั้น
3. สามารถจำกัดแพ็กเก็ตที่มีคุณสมบัติที่ไม่เหมาะสม เช่น การควบคุมปริมาณข้อมูลที่มีลักษณะเป็น Network Denial Of Service (DOS)
4. สามารถจำกัดการเปิดพอร์ตที่ไม่ควรเปิดของบางแอพพลิเคชัน เช่น พอร์ตของม้าโทรจันและแบล็คดอร์ต่าง ๆ (Trojan House, Backdoor)

2.3.3 ข้อจำกัดของไฟร์วอลล์

1. มีการทำงานที่เฉพาะเจาะจงบนตัวอุปกรณ์สูง (Platform Dependencies)
2. อาจมีข้อบกพร่องที่เกิดจากแอพพลิเคชัน (Application Vulnerability) ซึ่งเป็นสาเหตุทำให้เกิดข้อบกพร่องซึ่งนำไปสู่การบุกรุกได้ (Buffer Overflow, Unicode, ...)

3. อาจมีข้อบกพร่องของระบบปฏิบัติการ (OS Vulnerability) ทำให้อาจถูกโจมตีได้ เช่น การใช้ Winnuke หรือ Teardrop เป็นต้น

4. ไม่สามารถจัดการกับไวรัสบางตัวที่แฝงมาในการใช้งานตามปกติ เช่น การแนบไฟล์ที่ติดไวรัสบนอีเมล์ เช่น Bugbear หรือ Sircam เป็นต้น

2.4 โปรแกรม squid

ระบบเว็บพร็อกซีมักถูกติดตั้งบนเครื่องเกตเวย์เซิร์ฟเวอร์ซึ่งเป็นจุดเชื่อมระหว่างเครือข่าย และกำหนดให้ทำงานใน Transparent Mode เพื่อบังคับให้ทุก ๆ การร้องขอเว็บจะต้องกระทำผ่านเว็บพร็อกซีเสมอ ทำให้เว็บพร็อกซีสามารถตรวจสอบการร้องขอและจัดเก็บข้อมูลลงแคชได้ตามความเหมาะสม เมื่อเว็บพร็อกซีปฏิบัติตามการร้องขอจากไคลเอนต์แต่ละรายการเสร็จสิ้นก็จะทำการบันทึกผลการทำงานของการร้องขอนั้นลงสู่ล็อกไฟล์ โดยในงานวิจัยนี้ได้ใช้โปรแกรม squid [4] เวอร์ชัน 2.5 ทำหน้าที่เป็นเว็บพร็อกซี กำหนดให้บันทึกข้อมูลลงล็อกไฟล์ชื่อ access.log โดยในไฟล์ล็อก 1 บรรทัดจะเก็บบันทึกข้อมูลที่เกี่ยวข้องกับการร้องขอต่าง ๆ เรียงตามลำดับเวลาก่อนหลัง ซึ่งใน 1 บรรทัดจะประกอบไปด้วยข้อมูล 10 ฟิลด์ข้อมูล ดังรูปที่ 2.3

```
Timestamp Elapsed Client-Addr Log-Tag/HTTP-Code Size Req-Method
URL rfc931 Hierarchy/Hostname Client-Type
```

รูปที่ 2.3 รูปแบบล็อกไฟล์ของ squid

ในที่นี้จะขออธิบายความหมายเฉพาะฟิลด์ที่เกี่ยวข้องและนำมาใช้ในงานวิจัยเท่านั้น ได้แก่

Timestamp คือค่าเวลาปัจจุบันที่ squid จะบันทึกไว้เมื่อเสร็จสิ้นการร้องขอนั้นในรูปแบบ UTC

Client-Addr คือเบอร์ไอพีของไคลเอนต์

HTTP-Code คือสถานะตอบรับของการร้องขอ

URL คือ URL ของการร้องขอจากไคลเอนต์

Hostname คือเบอร์ไอพีของเว็บเซิร์ฟเวอร์

แต่ละฟิลด์ของล็อก squid จะถูกนำไปหาความสัมพันธ์กับฟิลด์ของล็อก snort ต่อไป

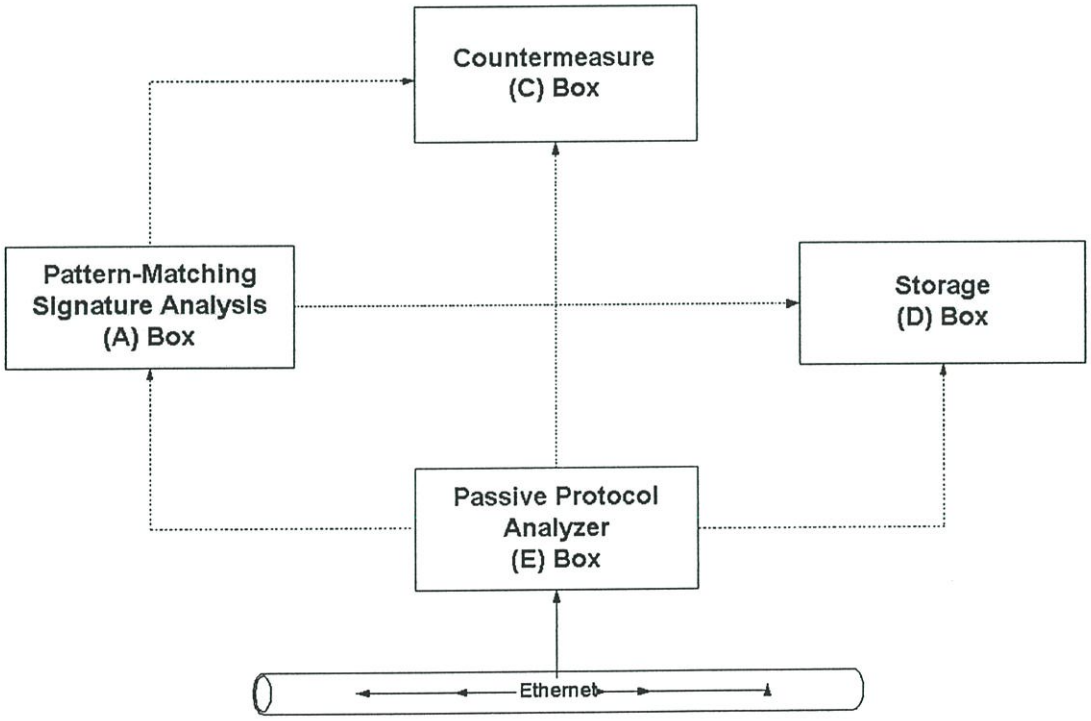
2.5 ระบบตรวจจับการบุกรุก (Intrusion Detection System)

IDS ทำหน้าที่ในการสอดส่องพฤติกรรมการใช้งานระบบเครือข่าย โดยเปรียบเทียบจากฐานข้อมูลรูปแบบการบุกรุก (signatures) ซึ่งกำหนดไว้ล่วงหน้า แล้วพิจารณาเปรียบเทียบพฤติกรรมใด ๆ ที่มีเงื่อนไขตรงกับที่ระบุไว้ใน signatures ว่าอาจจะเป็นการบุกรุกและบันทึกลงในล็อกไฟล์ ข้อผิด

พลาดที่เกิดจากการบันทึกล็อกของ IDS ได้แก่ กรณีที่มีการใช้งานเครือข่ายปกติใด ๆ ที่บังเอิญไปตรงกับ signatures ดังกล่าว แล้ว IDS ตรวจสอบเจอ ก็จะบันทึกการใช้งานที่ปกตินั้นลงล็อกไปด้วย (false positives) และกรณีมีบุกรุกใด ๆ ที่มีการทำงานที่ไม่ตรงกับ signatures ไม่ว่าด้วยเหตุผลใด ๆ ก็ตาม เช่นอาจเป็นการบุกรุกแบบใหม่ซึ่งยังไม่ได้กำหนดไว้ใน signatures หรือผู้บุกรุกใช้วิธีหลีกเลี่ยง (detection evading) แล้ว IDS ไม่สามารถตรวจสอบได้ ก็จะไม่มีการบันทึกค่าลงล็อก (false negatives) จากข้อผิดพลาดดังกล่าว ทำให้ข้อมูลล็อก IDS ขาดข้อมูลที่ควรมี (false negatives) และมีข้อมูลที่ไม่ควรมี (false positives) ซึ่งมีผลทำให้การวิเคราะห์ล็อก IDS ขาดประสิทธิภาพเท่าที่ควร

IDS คือเครื่องมือที่ใช้ในการตรวจสอบเหตุการณ์ที่น่าสงสัย โดยที่คุณสมบัติของ IDS ส่วนหนึ่งมีพื้นฐานมาจากความล้มเหลวของคุณสมบัติไฟร์วอลล์นั่นเอง IDS

2.5.1 หลักการทำงานของ IDS



รูปที่ 2.4 แบบจำลอง Common Intrusion Detection Framework

โมเดลที่ใช้เป็นมาตรฐานของ IDS ได้แก่ Common Intrusion Detection Framework (CIDF) [2] ซึ่งได้กล่าวถึงส่วนประกอบหลักของ IDS ไว้สี่ส่วน ดังรูปที่ 2.4 คือ

1. event generators (E-boxes) ทำหน้าที่รวบรวมข้อมูล ซึ่งจะต้องทำงานในระดับเลเยอร์ต่ำ โดยการดักจับแพ็กเก็ตข้อมูลทั้งหมดแล้วส่งไปยังหน่วยวิเคราะห์หรือหน่วยจัดเก็บข้อมูลต่อไป ถือว่าเป็นส่วนอินพุทที่ใหญ่ที่สุดของ IDS
2. analysis engines (A-boxes) ทำหน้าที่วิเคราะห์การบุกรุก โดยการใช้วิธีการ pattern-matching กับ signature แล้วส่งผลที่ได้ไปแสดงผลหรือจัดเก็บต่อไป ในบางครั้งอาจตัดสินใจโต้ตอบการบุกรุกนั้น ๆ ถ้าการวิเคราะห์นั้นมีความน่าเชื่อถือสูง
3. storage mechanisms (D-boxes) เก็บข้อมูลเพื่อใช้การวิเคราะห์ในภายหลัง ทั้งข้อมูลดิบที่ได้จากการดับจับแพ็กเก็ตและข้อมูลที่ได้จากการวิเคราะห์แล้ว
4. even countermeasures (C-boxes) เป็นมาตรการโต้ตอบการบุกรุกที่กระทำหลังจากมั่นใจได้ว่ามีการบุกรุกเกิดขึ้นจริง ๆ เช่น การ block IP ต้นทาง หรือการสั่งตัดการเชื่อมต่อนั้น ๆ

2.5.2 ประเภทของ IDS

เราสามารถแบ่ง IDS ตามลักษณะการทำงานได้หลายประเภท คือ

1. *misuse detection* คือ IDS ที่ใช้หลักการการเปรียบเทียบเหตุการณ์ (pattern matching) กับฐานข้อมูล (signatures) ที่ได้กำหนดไว้ล่วงหน้าแล้ว ซึ่งทำให้ IDS ชนิดนี้สามารถตรวจสอบได้เฉพาะการบุกรุกที่เคยเกิดขึ้นมาแล้ว เช่น รูรั่วของซอฟต์แวร์ เป็นต้น เนื่องจากการทำงานชนิดนี้ขึ้นอยู่กับหลักการกำหนด signature ซึ่งกระทำโดยมนุษย์ ที่ต้องมีทักษะสูง ดังนั้นจึงอาจเกิดข้อผิดพลาดได้หากกำหนด signature ไม่ถูกต้องและครอบคลุมเพียงพอ
2. *anomaly detection* คือ IDS ที่ใช้หลักการในการเปรียบเทียบเหตุการณ์นั้นกับเหตุการณ์ตามปกติที่ควรจะเป็น หากเหตุการณ์ใดอยู่นอกเหนือการทำงานตามปกติ IDS จะสงสัยเหตุการณ์นั้นเป็นการบุกรุก เช่น การพยายามฝืนใช้คำสั่งที่ต้องใช้สิทธิ์ระดับผู้ดูแลระบบ (root) ของผู้ใช้ธรรมดา (user) การบันทึกเหตุการณ์ปกติ อาจใช้วิธีการเก็บพฤติกรรมการใช้งานตามปกติเป็นระยะเวลาช่วงหนึ่ง

นอกจากนี้ในงานวิจัยใหม่ ๆ จะกล่าวถึงการตรวจสอบนี้ด้วย คือ

3. *threshold detection* คือ IDS ที่ใช้หลักการตรวจสอบโดยการหนดจำนวนครั้งของเหตุการณ์นั้นขึ้นมา หากมีจำนวนครั้งมากกว่าค่าที่ตั้งไว้ ก็จะให้สงสัยว่าเหตุการณ์นั้นเป็นการบุกรุก เช่น การเรียกใช้บริการบางบริการที่ถี่เกินความจำเป็นเพื่อเตือนการบุกรุกแบบ Denial Of Service (DOS)

นอกจากนี้เรายังสามารถแบ่ง IDS ตามตำแหน่งได้ 2 ประเภทเช่นกัน คือ

1. *network-based IDS* คือ IDS ที่ใช้สำหรับตรวจสอบหาแพ็กเก็ตผิดปกติที่รับส่งไปมาระหว่างเครือข่ายทั้งหมด นิยมนำไปติดตั้ง ณ ตำแหน่งที่สามารถมองเห็นแพ็กเก็ตข้อมูลทั้งหมดของเครือข่ายได้ เช่น จุดเชื่อมต่อของเราเตอร์ หรือ mirror port ของอุปกรณ์สวิทช์
2. *host-based IDS* คือ IDS ที่ใช้สำหรับดูแลตรวจสอบโฮสต์ใดโฮสต์หนึ่งเป็นพิเศษ นิยมนำไปติดตั้งบนโฮสต์นั้นเลย เพื่อให้สามารถตรวจสอบข้อมูลทั้งหมดที่รับส่งระหว่างโฮสต์นั้น ๆ เพื่อนำมาวิเคราะห์ว่าเป็นการบุกรุกเซิร์ฟเวอร์นั้น ๆ หรือไม่

การแจ้งเตือน (alert) ที่ผิดพลาดของ IDS แบ่งออกได้เป็นสองประเภท คือ

1. *false positives alert* คือการแจ้งเตือนในเหตุการณ์ที่ปกติซึ่งอาจบังเอิญไปตรงกับเหตุการณ์การบุกรุกที่ได้กำหนดเอาไว้ ผลที่ตามมาถึงแม้จะไม่เกิดอันตรายมากนัก แต่ก็จะทำให้เกิดความสับสนในการวิเคราะห์การบุกรุกจริง ๆ ได้
2. *false negatives alert* คือการไม่แจ้งเตือนเมื่อมีการบุกรุกจริง ๆ เข้ามา สาเหตุอาจเกิดมาจากการกำหนด signature ที่ไม่ดีพอ หรือ การเกิดการจงใจหลีกเลี่ยงการตรวจสอบด้วย IDS ผลตามมา ทำให้ผู้บุกรุกสามารถโจมตีเราได้ โดยที่เราไม่ทันตั้งตัว

ผลเสียที่เกิดจาก false negatives มักจะรุนแรงกว่า false positives เสมอ ซึ่งการปล่อยให้การบุกรุกเพียงครั้งเดียวหลุดพ้นการตรวจสอบไปอาจก่อให้เกิดผลเสียหายที่อาจประมาณค่ามิได้ จากหลักการทำงานของระบบ IDS ที่อาศัยการตรวจสอบข้อมูลกับฐานข้อมูลเพื่อวิเคราะห์การบุกรุก (pattern matching) ยังอาศัยการเปรียบเทียบข้อมูลของกฎ (rules) กับ URL ของเหตุการณ์นั้น ๆ สูง จึงก่อให้เกิดเทคนิคมากมายที่จะเลี่ยงการตรวจสอบ pattern matching ของ IDS

2.5.3 เทคนิคการหลบเลี่ยง IDS

เทคนิคที่ใช้ในการหลบเลี่ยงการตรวจสอบด้วย IDS [2] ซึ่งก่อให้เกิด false negatives ตามมาได้แก่

1. *Insertion* คือการทำให้ IDS เห็นข้อมูลผิดเพี้ยนไปจากความเป็นจริงเนื่องมาจากการเพิ่มของข้อมูลบางตัวทำให้ IDS ใช้วิธีการ pattern matching ในการตรวจสอบไม่พบ เช่น cat+/etc/passwd
2. *Evasion* คือการทำให้ IDS เห็นข้อมูลผิดเพี้ยนไปจากความเป็นจริงเนื่องมาจากการขาดหายไปของข้อมูลบางตัวทำให้ IDS ใช้ pattern matching ในการตรวจสอบไม่พบ

3. *Denial of Service* คือการโจมตีไปยังระบบ IDS เพื่อให้ IDS ไม่สามารถทำการตรวจสอบแพ็กเก็ตข้อมูลได้ตามปกติ ทำให้อาจมีข้อมูลการบุกรุกบางส่วนไม่ได้รับการตรวจสอบ

2.5.4 ข้อจำกัดของ Network-based IDS [1]

1. *Line speed* วิเคราะห์ข้อมูลจำนวนมากบนเครือข่าย
2. *Encryption Data* ไม่สามารถวิเคราะห์ข้อมูลที่เข้ารหัสข้อมูลได้ (SSL, VPN, SSH)
3. *Content Inspection* ไม่สามารถตรวจสอบข้อมูลในระดับเนื้อหาได้
4. *Countermeasure* ไม่สามารถทำการตอบโต้การบุกรุกได้อย่างเหมาะสม

2.5.5 ข้อจำกัดของ Host-based IDS

1. *System Resource* ใช้ทรัพยากรสูงในการวิเคราะห์การบุกรุก
2. *Real-time response* เบียดเบียนทรัพยากรปกติของเซิร์ฟเวอร์นั้น
3. *Countermeasure* ไม่สามารถทำการตอบโต้การบุกรุกได้อย่างเหมาะสม

ในงานวิจัย [10] ได้กล่าวถึงวิธีการในการกำจัดจุดอ่อนของ Host-based IDS ที่ไม่สามารถจัดการข้อมูลในระดับเน็ตเวิร์กเลเยอร์ได้โดยการ ปรับปรุงเคอเนลล์ของโฮสต์นั้น ๆ ให้ส่งข้อมูลระดับต่ำที่เป็นประโยชน์เพื่อให้ H-IDS สามารถวิเคราะห์การบุกรุกได้อย่างละเอียดยิ่งขึ้น

เทคนิคอื่น ๆ ที่ใช้ในการหลบเลี่ยง IDS [11] ได้แก่ การเข้ารหัสข้อมูลด้วย SSL การส่งข้อมูลในลักษณะแฟรกเมนต์ที่ไม่เรียงลำดับ นอกจากนี้ยังมีวิธีการส่งชุดของข้อมูลที่อยู่ในรูปแบบของ "polymorphic shellcode" ซึ่งเป็นการเขียน URL ที่แตกต่างกันออกไปเพื่อหลบเลี่ยงการตรวจสอบ pattern matching ของ IDS เช่น วิธีการเอ็นโค้ดตัวอักษรให้อยู่ในรูปของเลขฐานสิบหก การเขียน URL ด้วยฟอร์แมตยูนิโค้ด (unicode) การเพิ่มเส้นทาง (path) ปลอมเข้าไปด้วย ../ , /././ , //// และการใช้ตัวแบ่งพาทที่ไม่เป็นมาตรฐาน \ ตลอดจนการใช้หลาย ๆ เทคนิคเข้าร่วมกัน

ต่อมาได้มีการพัฒนาให้ไฟร์วอลล์และ IDS มีการทำงานที่ประสานกันมากขึ้นเช่น เมื่อ IDS ตรวจสอบพบการบุกรุกก็จะส่งคำสั่งไปให้ไฟร์วอลล์ทำการ block IP ต้นทางหรือ reset การเชื่อมต่อนั้น ๆ เช่น โครงการ Guardian (snort+swatch+iptables) เป็นต้น นอกจากนี้ยังได้มีการพัฒนา IDS ให้มีคุณสมบัติในการ block การโจมตีในขณะนั้น (real-time) เรียกว่า Intrusion Prevention System (IPS) เช่นโครงการ Hogwash, Snort-Inline (IDS+drop action) รวมถึงได้มี

การพัฒนาแอพพลิเคชันเซิร์ฟเวอร์ให้มีคุณสมบัติ IDS อยู่ในตัว เช่นโครงการ mod_security ใน apache server

2.6 โปรแกรม snort

ระบบการตรวจสอบการบุกรุกเครือข่ายในปัจจุบันนิยมใช้แบบ Network-based IDS ซึ่งมีค่าติดตั้งที่ถูกกว่าแบบ Host-based IDS มาก เนื่องมาจากสามารถตรวจสอบเครื่องภายในเครือข่ายได้ทั้งหมดโดยไม่ต้องมาติดตั้ง Host-based IDS สำหรับทุก ๆ เครื่อง แต่ Network-based IDS ก็มีข้อจำกัด คือ ไม่สามารถจัดการกับข้อมูลที่มีการเข้ารหัสได้, อาจไม่สามารถจัดการกับแพ็กเก็ตข้อมูลจำนวนมากที่อยู่บนเครือข่ายได้ทัน และไม่สามารถรับรู้ถึงข้อมูลในระดับแอพพลิเคชันได้ [1] ทำให้ IDS ขาดข้อมูลบางส่วนในการวิเคราะห์ ส่งผลให้เกิดการแจ้งเตือนที่ผิดพลาดอยู่เสมอ

Network-based IDS จะทำงานโดยการกำหนดให้เน็ตเวิร์กการ์ดทำงานใน promiscuous mode ทำให้สามารถดักจับแพ็กเก็ตข้อมูลทั้งหมดได้ แล้วนำรายละเอียดข้อมูลภายในแพ็กเก็ตมาเปรียบเทียบกับรูปแบบการบุกรุก (signatures) ซึ่งสามารถกำหนดได้จากรูปแบบการบุกรุกในอดีต หากแพ็กเก็ตใดมีลักษณะข้อมูลตรงตาม signatures ที่ถูกกำหนดไว้ IDS ก็จะบันทึกรายละเอียดแพ็กเก็ตนั้นลงในล็อกไฟล์ โดยในงานวิจัยนี้ได้ใช้โปรแกรม snort [12] เวอร์ชัน 2.1.1 ซึ่งเป็นโปรแกรมที่แจกจ่ายฟรี (Open Source) เป็นโปรแกรม Network-based IDS ของระบบเครือข่าย

2.6.1 signature, classification, priority

snort พิจารณารูปแบบแพ็กเก็ตใดว่าเป็นการบุกรุก โดยใช้อัลกอริทึม Boyer-Moore [13] ในการเปรียบเทียบ (pattern matching) แพ็กเก็ตกับ signatures โปรแกรม snort เวอร์ชัน 2.1.1 นี้ได้กำหนด signatures การบุกรุกมาตรฐานไว้ทั้งหมด 2163 signatures และแบ่งการบุกรุกออกเป็นประเภท (classifications) เพื่อรวบรวม signatures ที่มีลักษณะการบุกรุกคล้ายกันจัดให้อยู่ในประเภทเดียวกันทั้งหมด 34 ประเภท ดังแสดงวิธีการคำนวณในรูปที่ 2.5

```
[root@athlon rules]# grep ^alert *.rules|wc –l
2163
[root@athlon etc]# grep ^config classification.config | wc -l
34
```

รูปที่ 2.5 จำนวน signatures และ classifications ของ snort

แต่ละประเภทของการบุกรุกจะกำหนดค่าระดับความรุนแรง (priority) กำกับไว้เสมอ ค่า priority นี้จะลดระดับความรุนแรงตามค่าตัวเลขที่เพิ่มขึ้น โดยเริ่มต้นจาก 1 ที่มีค่าความรุนแรงสูงสุด ดังแสดงในรูปที่ 2.6

```
config classification: web-application-attack,Web Application Attack,1
config classification: web-application-activity,access to a potentially vulnerable web application,2
config classification: string-detect,A suspicious string was detected,3
```

รูปที่ 2.6 ตัวอย่างค่า classifications และ priority ของ snort

นอกจาก signatures และ classifications ที่ snort กำหนดมาให้เป็นมาตรฐานแล้ว ผู้ใช้สามารถกำหนด signatures และ classifications ของการบุกรุกตลอดจนกำกับค่า priority ตามความเหมาะสมเองได้ ตัวอย่าง signatures ของ snort มีรูปแบบดังรูปที่ 2.7

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-ATTACKS
/etc/shadow access"; flow:to_server,established; content:"/etc/shadow";nocase; sid:1372; classtype:web-
application-activity; rev:4;)
```

รูปที่ 2.7 รูปแบบ signature ของ snort

เมื่อ snort ตรวจสอบพบว่าแพ็กเก็ตใดมีรูปแบบตรงตาม signature ที่กำหนดไว้ ก็จะทำการบันทึกรายละเอียดแพ็กเก็ตนั้นลงล็อกไฟล์ตามรูปแบบที่เรากำหนด โดยในงานวิจัยนี้ได้กำหนดให้บันทึกการแจ้งเตือน (alert) เป็นแบบเท็กซ์ไฟล์ลงล็อกไฟล์ชื่อ alert การเก็บบันทึกจะเรียงลำดับตามเวลาการบุกรุกก่อนหลัง แต่ละ alert จะถูกแยกด้วยบรรทัดว่างหนึ่งบรรทัดเสมอ โดย alert จะมีข้อมูลรายละเอียดได้หลายบรรทัด รูปแบบการแจ้งเตือนประกอบด้วยฟิลด์ตามตัวอย่าง ดังรูปที่ 2.8

```
1...[**] [1:884:8] WEB-CGI formmail access [**]
2...[Classification: access to a potentially vulnerable web application] [Priority: 2]
3...11/23-02:31:39.187007 192.168.1.131:51424 -> 192.168.0.10:80
4...TCP TTL:127 TOS:0x0 ID:6192 IpLen:20 DgmLen:456 DF
5...***AP*** Seq: 0xB04FCD60 Ack: 0x4CE5CFCD Win: 0x7D00 TcpLen: 20
6...[Xref => http://www.whitehats.com/info/IDS226][Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-
1999-0172][Xref => http://www.securityfocus.com/bid/1187][Xref =>
http://cgi.nessus.org/plugins/dump.php3?id=10076][Xref
=> http://cgi.nessus.org/plugins/dump.php3?id=10782]
```

รูปที่ 2.8 รูปแบบล็อกไฟล์ของ snort

อธิบายรายละเอียดของแต่ละฟิลด์ในแต่ละบรรทัดตามลำดับได้ดังนี้

บรรทัดที่ 1 Alert Message คือ คำอธิบายของการบุกรุก

บรรทัดที่ 2 Classification คือ ประเภทของการบุกรุกและ Priority คือ ระดับความรุนแรงของการบุกรุก

บรรทัดที่ 3 Timestamp คือ ค่าเวลาปัจจุบันที่ Snort จะบันทึกไว้เมื่อเสร็จสิ้นการร้องขอนั้น, Source IP คือเบอร์ไอพีของผู้บุกรุก, Source Port คือเบอร์พอร์ตของผู้บุกรุก, Target IP คือเบอร์ไอพีของเป้าหมาย และ Target Port คือเบอร์พอร์ตของเป้าหมาย

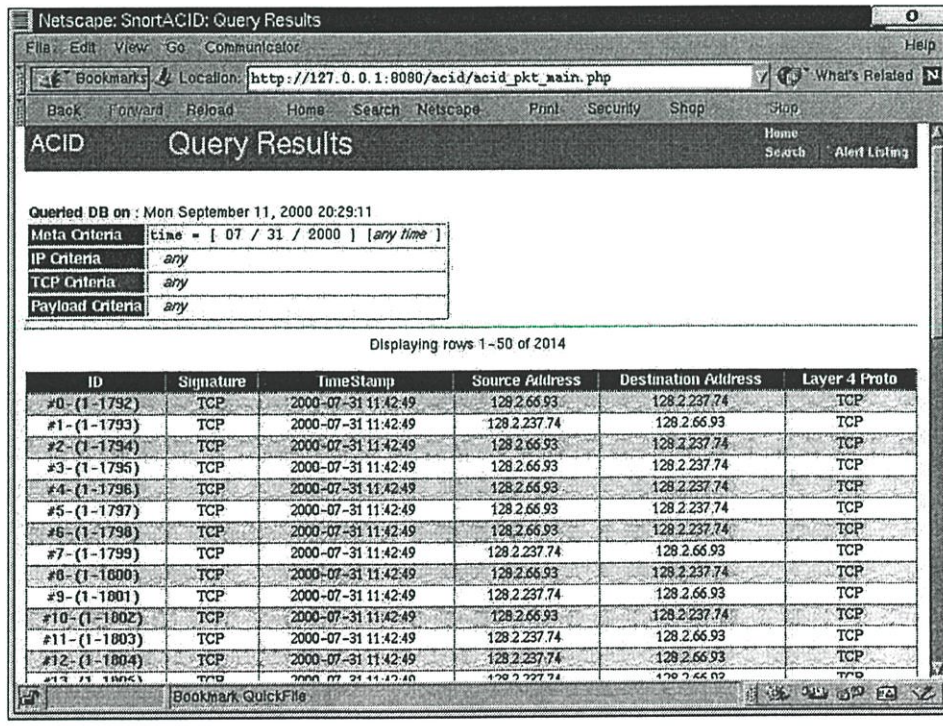
บรรทัดที่ 4,5 ค่ารายละเอียดของโพรโตคอล

บรรทัดที่ 6 แหล่งอ้างอิงของการบุกรุก

ในที่นี้เราจะเลือกใช้เฉพาะฟิลด์ที่เกี่ยวข้องและนำมาใช้ในงานวิจัยเท่านั้น ได้แก่ Alert Message, Priority, Timestamp, Source IP และ Target IP

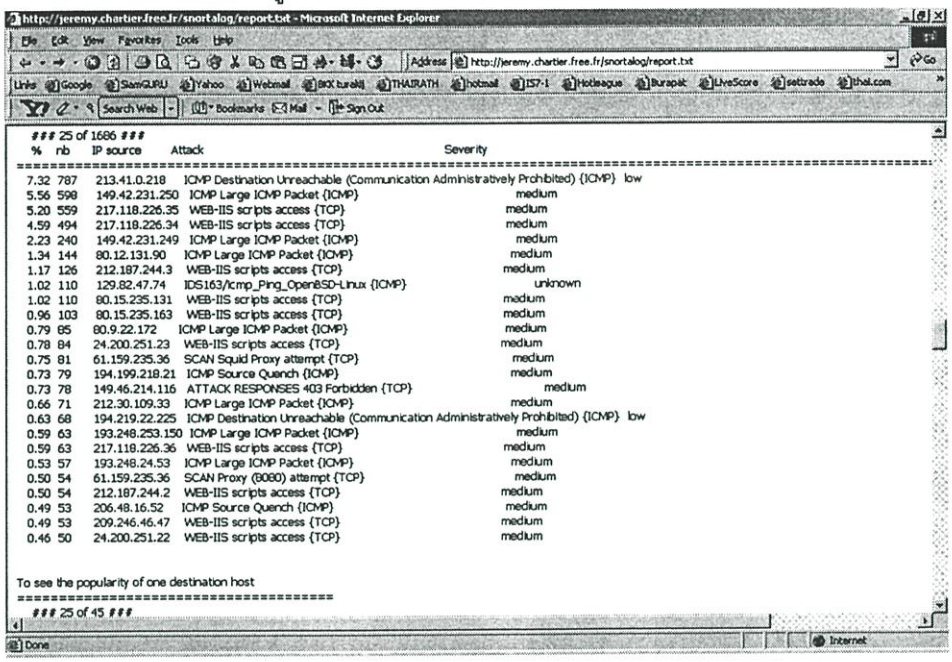
2.6.2 การวิเคราะห์ล็อกไฟล์

- *Analysis Console for Intrusion Databases (ACID)* [15] เป็นโปรแกรมที่เขียนโดยใช้ภาษา PHP ใช้วิเคราะห์ล็อกที่เป็นฐานข้อมูล สามารถแสดงผลลัพธ์เป็นรูปแบบกราฟิกผ่านเว็บเบสได้ ดังรูปที่ 2.9



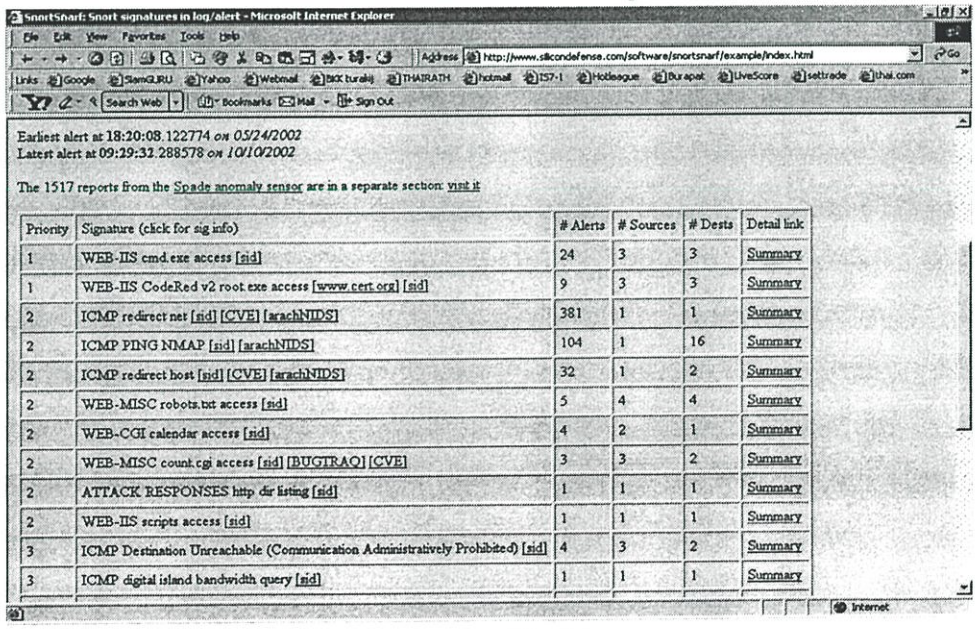
รูปที่ 2.9 แสดงตัวอย่างผลการวิเคราะห์ด้วย ACID

- *Snortalog* ใช้ภาษา Perl วิเคราะห์ล็อกที่เป็นเท็กซ์ไฟล์ แสดงผลลัพธ์ได้ทั้งทางโหมดตัวอักษรและบนเว็บ ดังรูปที่ 2.10



รูปที่ 2.10 แสดงตัวอย่างผลการวิเคราะห์ด้วยโปรแกรม snortalog

- *SnortSnarf (Silicon Defense)* [16] ใช้ภาษา Perl วิเคราะห์ล็อกที่เป็นเท็กซ์ไฟล์ แสดงผลลัพธ์ออกมาเป็นเอกสาร HTML แบบเท็กซ์โหมด ดังรูปที่ 2.11



รูปที่ 2.11 แสดงตัวอย่างผลลัพธ์การวิเคราะห์ด้วยโปรแกรม SnortSnarf

2.7 งานวิจัยที่เกี่ยวข้อง

ปกติโปรแกรม Network-based IDS เช่น snort จะทำหน้าที่ในการบันทึกข้อมูลที่น่าสงสัยลงสู่ล็อกไฟล์เพียงอย่างเดียว หน้าที่ในการวิเคราะห์การบุกรุกจะต้องใช้โปรแกรมเสริมอ่านข้อมูลจากล็อกไฟล์เหล่านั้นแล้วนำไปวิเคราะห์อีกทอดหนึ่ง ซึ่งปัจจุบันผลการวิเคราะห์ส่วนมากมักประกอบไปด้วยการแจ้งเตือนที่ผิดพลาดจำนวนมาก ทำให้ในบางครั้งผลการแจ้งเตือนที่ผิดพลาดเหล่านี้ไปบิดเบือนผลการแจ้งเตือนการบุกรุกที่แท้จริง ทำให้ประสิทธิภาพในการวิเคราะห์การบุกรุกลดลงไปเป็นอย่างมาก โปรแกรมวิเคราะห์การบุกรุกที่ได้รับความนิยมได้แก่ Analysis Console for Intrusion Databases (ACID) [15] และ SnortSnarf [16] เป็นต้น

ปัจจุบันอัตราการบุกรุกทางเครือข่ายได้เพิ่มขึ้นอย่างรวดเร็ว ทำให้การวิเคราะห์การบุกรุกจึงต้องมีการพัฒนาให้สอดคล้องตามไปด้วย จากการวิเคราะห์โดยใช้ข้อมูลเพียงลำพัง ก็ได้เริ่มพัฒนาให้มีการใช้ข้อมูลจากหลาย ๆ แหล่งเพิ่มเติม เช่น ในงานวิจัยของ Alfonso Valdes และ Keith Skinner [3] แห่ง SRI International ได้กล่าวถึง การหาความสัมพันธ์ระหว่างอุปกรณ์ตรวจสอบ (Sensor Correlation) ไว้ว่า โดยทั่วไป sensor correlation ประกอบด้วย 3 ส่วนหลัก คือ การรวบรวมเหตุการณ์ (event aggregation) การเชื่อมความสัมพันธ์ระหว่างตัวตรวจสอบ (sensor coupling) และการสรุปผลการวิเคราะห์ (meta alert fusion) โดยที่ sensor correlation ที่ดีควรมีคุณสมบัติ ดังนี้ คือ สามารถหาความความสัมพันธ์ของเหตุการณ์ในระดับต่ำ (low-level) จำนวนมากได้ สามารถสรุปเหตุการณ์การบุกรุกจากหลาย ๆ ตัวตรวจสอบ (sensors) ได้โดยเร็วที่สุดและให้มีข้อผิดพลาด (false alert) ที่น้อยที่สุดและสามารถวิเคราะห์การบุกรุกใหม่ ๆ ที่อาจเกิดได้ในอนาคต นอกจากนี้ยังมีการใช้ความสัมพันธ์ระหว่างการแจ้งเตือน (alert correlation) โดยใช้ทฤษฎีความน่าจะเป็น (Probabilistic) [4] เข้ามาช่วย ซึ่งใช้ similarity function เข้ามาช่วยคำนวณหาค่าเหตุการณ์ที่คล้ายกัน โดยผลลัพธ์ที่ได้จะอยู่ในช่วง 0 (mismatch) ถึง 1 (perfect match) ในงานวิจัยของ Alfonso Valdes และชาวคณะได้ทำการทดลอง sensor correlation โดยใช้ตัว sensor สองชนิด [5] คือ snort [12] และ EMERALD [17] แลกเปลี่ยนข้อมูลกันโดยใช้มาตรฐานแบบ Intrusion Detection Message Exchange Format (IDMEF) ซึ่งเป็นข้อมูลในรูปแบบของ XML syntax พัฒนาโดย IETF/IDWG ในการเก็บข้อมูล ผลลัพธ์ที่ได้คือ ในกรณีของ snort เมื่อมีการใช้ correlation สามารถลดการแจ้งเตือนได้ถึง 10 ต่อ 1 ส่วนถ้าเป็น EMERALD สามารถลดได้ 3 ต่อ 1 แต่ถ้านำทั้งหมดมารวมกันสามารถลดได้ในอัตรา 7 ต่อ 1 นอกจากนี้ยังมีงานวิจัยของ Herve Debar และ Andreas Wespi [6] แห่ง IBM Reseach, Zurich Research Laboratory Switzerland ได้ประยุกต์ใช้วิธีการ correlation กับระบบ Tivoli Enterprise Console (TEC) ของ IBM หลังการ correlation สามารถจัดการกับปัญหาข้อบกพร่องของ IDS ได้ เช่น การเลือกแสดงผลเฉพาะกลุ่มของการบุกรุกที่สำคัญๆ เป็นต้น เห็นได้ว่าตลอดเวลาที่ผ่านมาได้มีการ

พัฒนาวิธีการวิเคราะห์การบุกรุกโดยอาศัยการรวบรวมข้อมูลที่สัมพันธ์กันจากเครื่องมือตรวจสอบหลายๆชนิดหลายๆตัวเข้าด้วยกัน ประโยชน์เพื่อรวบรวมข้อมูลการบุกรุกให้ได้มากที่สุดนั่นเอง ดังนั้นในงานวิจัยนี้จึงได้นำเสนอแนวคิดในการเชื่อมความสัมพันธ์ระหว่างล็อกไฟล์จากเครื่องมือตรวจสอบต่างชนิดกัน อันได้แก่ เว็บพร็อกซีและ Network-based IDS มาหาความสัมพันธ์ระหว่างกัน ก่อนนำไปวิเคราะห์การบุกรุกที่แท้จริงต่อไป

An Approach to Sensor Correlation [3]

Alfonso Valdes และ Keith Skinner แห่ง SRI International ได้กล่าวถึง sensor correlation ไว้ว่า Sensor Correlation ประกอบด้วย 3 ส่วนหลัก คือ

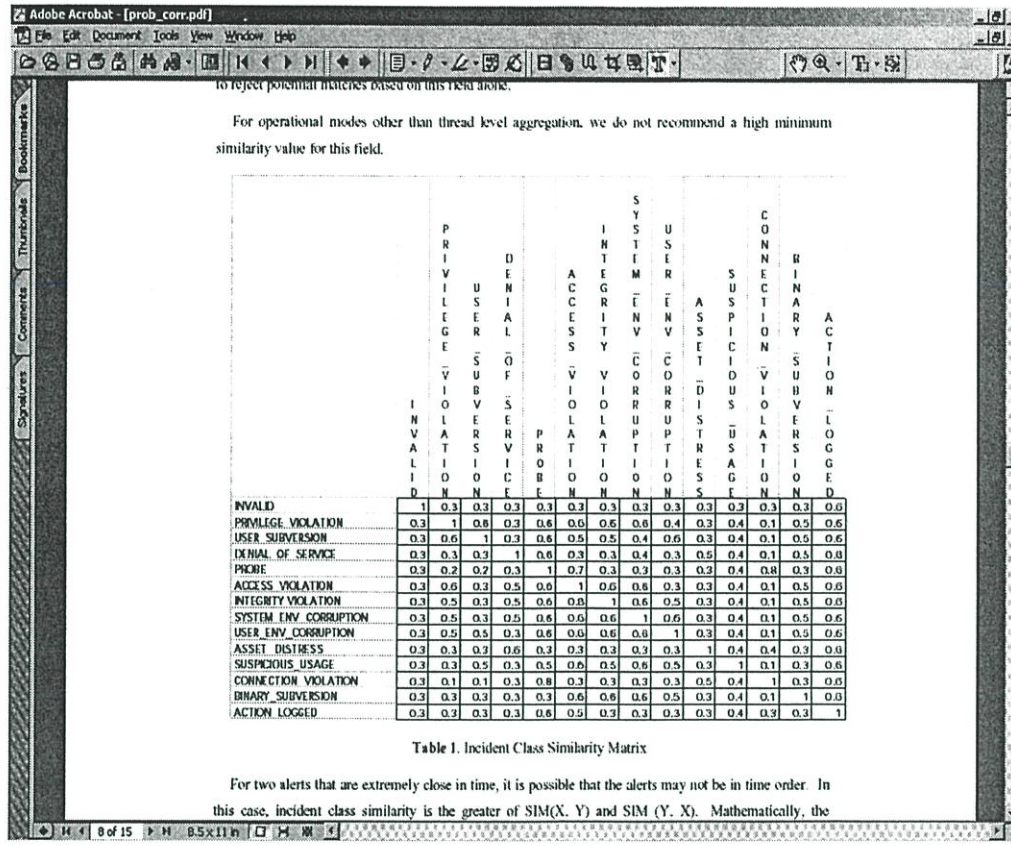
1. event aggregation, การรวบรวมเหตุการณ์
2. sensor coupling, การเชื่อมความสัมพันธ์ระหว่างตัวตรวจสอบ
3. meta alert fusion, การสรุปผลการวิเคราะห์

โดยที่ Sensor Correlation ที่ดีควรมีคุณลักษณะ ดังนี้

1. สามารถหาความความสัมพันธ์ของเหตุการณ์ในระดับต่ำ (low-level) จำนวนมากให้ได้
2. สามารถสรุปเหตุการณ์การบุกรุกจากหลาย ๆ ตัวตรวจสอบ (sensors) ให้ได้โดยเร็วที่สุดและให้มีข้อผิดพลาด (false alarm) ที่น้อยที่สุด
3. สามารถวิเคราะห์การบุกรุกใหม่ ๆ ที่อาจเกิดได้ในอนาคต

Probabilistic Alert Correlation [4]

Alfonso Valdes และ Keith Skinner แห่ง SRI International ได้ใช้การ alert correlation โดยใช้ทฤษฎีความน่าจะเป็น (Probabilistic) เข้ามาช่วย โดยใช้ similarity function เข้ามาช่วยคำนวณหาค่าเหตุการณ์ที่คล้ายกัน โดยผลลัพธ์ที่ได้จะอยู่ในช่วง 0 (mismatch) ถึง 1 (perfect match) แสดงตัวอย่างดังรูปที่ 2.12



To reject potential matches based on this field alone.

For operational modes other than thread level aggregation, we do not recommend a high minimum similarity value for this field.

	INVALID	PRIVILEGE_VIOLATION	USER_SUBVERSION	DENIAL_OF_SERVICE	PROBE	ACCESS_VIOLATION	INTEGRITY_VIOLATION	SYSTEM_ENV_CORRUPTION	USER_ENV_CORRUPTION	ASSET_DISTRESS	SUSPICIOUS_USAGE	CONNECTION_VIOLATION	BINARY_SUBVERSION	ACTION_LOGGED
INVALID	1	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.6
PRIVILEGE_VIOLATION	0.3	1	0.8	0.3	0.6	0.6	0.6	0.6	0.4	0.3	0.4	0.1	0.5	0.6
USER_SUBVERSION	0.3	0.6	1	0.3	0.6	0.5	0.5	0.4	0.6	0.3	0.4	0.1	0.5	0.6
DENIAL_OF_SERVICE	0.3	0.3	0.3	1	0.6	0.3	0.3	0.4	0.3	0.5	0.4	0.1	0.5	0.6
PROBE	0.3	0.2	0.2	0.3	1	0.7	0.3	0.3	0.3	0.3	0.4	0.8	0.3	0.6
ACCESS_VIOLATION	0.3	0.6	0.3	0.5	0.6	1	0.6	0.6	0.3	0.3	0.4	0.1	0.5	0.6
INTEGRITY_VIOLATION	0.3	0.5	0.3	0.5	0.6	0.8	1	0.6	0.5	0.3	0.4	0.1	0.5	0.6
SYSTEM_ENV_CORRUPTION	0.3	0.5	0.3	0.5	0.6	0.6	0.6	1	0.6	0.3	0.4	0.1	0.5	0.6
USER_ENV_CORRUPTION	0.3	0.5	0.5	0.3	0.6	0.6	0.6	0.6	1	0.3	0.4	0.1	0.5	0.6
ASSET_DISTRESS	0.3	0.3	0.3	0.6	0.3	0.3	0.3	0.3	0.3	1	0.4	0.4	0.3	0.6
SUSPICIOUS_USAGE	0.3	0.3	0.5	0.3	0.5	0.6	0.5	0.6	0.5	0.3	1	0.1	0.3	0.6
CONNECTION_VIOLATION	0.3	0.1	0.1	0.3	0.8	0.3	0.3	0.3	0.3	0.5	0.4	1	0.3	0.6
BINARY_SUBVERSION	0.3	0.3	0.3	0.3	0.3	0.6	0.6	0.6	0.5	0.3	0.4	0.1	1	0.6
ACTION_LOGGED	0.3	0.3	0.3	0.3	0.6	0.5	0.3	0.3	0.3	0.3	0.4	0.3	0.3	1

Table 1. Incident Class Similarity Matrix

For two alerts that are extremely close in time, it is possible that the alerts may not be in time order. In this case, incident class similarity is the greater of SIM(X, Y) and SIM (Y, X). Mathematically, the

รูปที่ 2.12 ตัวอย่างผลลัพธ์งานวิจัย Probabilistic Alert Correlation

Heterogeneous Sensor Correlation: A Case Study of Live Traffic Analysis [5]

Alfonso Valdes และชาวคณะได้ทำการทดลองการ Sensor Correlation โดยใช้ตัว sensor สองชนิดคือ snort และ EMERALD โดยใช้มาตรฐานข้อมูลแบบ Intrusion Detection Message Exchange Format (IDMEF) ซึ่งเป็นข้อมูลในรูปแบบของ XML syntax พัฒนาโดย IETF/IDWG ในการเก็บข้อมูล ซึ่งได้ผลลัพธ์ ดังตารางที่ 2.1

ตารางที่ 2.1 แสดงตัวอย่างข้อมูลของงานวิจัย Heterogeneous Sensor Correlation

Sensor	Raw Alerts	Correlated Alerts
SNORT	4816	487
EMERALD	1586	523
Composite	6402	869

ซึ่งในกรณีของ snort เมื่อการใช้ correlation สามารถลดการแจ้งเตือนได้ถึง 10/1 ส่วนถ้าเป็น EMERALD สามารถลดได้ 3/1 แต่ถ้านำทั้งหมดมารวมกันสามารถลดได้ในอัตรา 7/1

Aggregation and Correlation of Intrusion-Detection Alerts [6]

Herve Debar และ Andreas Wespi แห่ง IBM Reseach, Zurich Research Laboratory Switzerland ได้ประยุกต์ใช้วิธีการ correlation กับระบบ Tivoli Enterprise Console (TEC) ของ IBM หลังการ correlation สามารถจัดการกับปัญหาข้อบกพร่องของ IDS เหล่านี้ได้โดย

Flooding ผู้บุกรุกสามารถสร้าง flood ต่อ IDS ได้ง่ายมากและการจะไปจำกัดหรือลดจำนวน signature ก็ไม่ใช่วิธีแก้ไขที่ถูกต้อง เพราะการลด signature ทำให้ลด false positive ก็จริง แต่ก็มีโอกาสในการเพิ่ม false negative ด้วย เช่นกันและ alerts แต่ละตัวมีค่าเสมอ ไม่ควรมองข้าม แม้ว่าไม่ควรแสดงตอนเจ้าหน้าที่ตรวจดูก็ตาม แต่อาจมีประโยชน์ในการวิเคราะห์ภายหลัง (offline mode) ได้ โดยการ correlation จะทำหน้าที่จับกลุ่มของ alert เข้าด้วยกันและจะแสดงเป็นลักษณะของ multi-view โดยจะเลือกแสดงผลเฉพาะ view ที่สำคัญ ๆ เท่านั้น

Context ตามปกติแล้วผู้บุกรุกมักจะสำรวจเป้าหมายก่อนเสมอ แล้วค่อยเริ่มเจาะระบบจริง ๆ การ correlation จะวิเคราะห์และลำดับเหตุการณ์การบุกรุกที่เกี่ยวข้องกันทั้งหมดไว้ด้วยกัน ทำให้เจ้าหน้าที่สามารถเรียนรู้ลำดับการบุกรุกเหล่านี้เพื่อให้สามารถจัดการได้อย่างเหมาะสม และหาทางป้องกันและแก้ไขต่อไป (HoneyPot, HonetD, HoneyNet)

False alerts การเลี่ยง false negatives จำเป็นจะต้องยอมรับการเพิ่มของ false positives ด้วย ข้อผิดพลาดในการแจ้งเตือนเกิดจาก Intrinsic inaccuracy เกิดจากการกำหนดรหัสการตรวจสอบ ที่ไม่ดีพอดังนั้นมันจึงไม่สามารถแยะแยะการทำงานตามปกติกับการบุกรุกได้โดยเฉพาะในกรณีการบุกรุกที่มีความซำนาญและ Relative inaccuracy เกิดจากการทำงานตามปกติซึ่งไปคล้ายคลึงกับรูปแบบการบุกรุก ซึงการ correlation สามารถจัดการได้โดยการใช้ค่า confidence value ซึ่งค่านี้จะกำหนดได้โดยอิสระในแต่ละกลุ่มของเหตุการณ์ โดยค่าปกติ (default confidence value) จะถูกกำหนดมาให้เหมาะสมกับแต่ละเหตุการณ์ (intrinsic) และสามารถปรับแต่งให้เหมาะสมกับการทำงานที่เป็นปกติของแต่ละเครือข่าย (Relative)

Constructing Attack Scenarios through Correlation of Intrusion Alerts [9]

การบุกรุกส่วนมากมักจะเป็นเหตุการณ์ที่ต่อเนื่องกัน (series of attacks) ซึ่งโดยปกติหลาย ๆ เหตุการณ์มักไม่เกิดแยกจากกัน แต่จะเกิดขึ้นแบบต่อเนื่องและมีความสัมพันธ์ระหว่างเหตุการณ์กันอยู่ ทางเราจึงได้เสนอวิธีการที่ทำให้เจ้าหน้าที่หรือระบบการตรวจสอบสามารถจัด

การกับการบุกรุกได้เหมาะสมและสะดวกยิ่งขึ้น โดยการสร้างสถานการณ์การบุกรุกจากเหตุการณ์การบุกรุกที่สัมพันธ์กันบนพื้นฐานของ เหตุการณ์ที่เกิดขึ้นก่อน (prerequisites) และเหตุการณ์ที่เกิดตามหลัง (consequences) จากการบุกรุก โดย prerequisites คือเหตุการณ์ที่จำเป็นที่จะต้องเกิดก่อนเหตุการณ์ที่เป็นการบุกรุกที่แท้จริงจะสำเร็จผล ส่วน consequences นั้น คือเหตุการณ์ที่มักเป็นการเก็บเกี่ยวผลประโยชน์หลังจากเกิดเหตุการณ์การบุกรุกนั้นจริง ๆ การวิเคราะห์การบุกรุกโดยพิจารณาถึงเหตุการณ์ที่ต้องเกิดขึ้นก่อนและเหตุการณ์ที่มักเกิดขึ้นตามมา ทำให้เรามีข้อมูลที่เป็นประโยชน์เพิ่มมากยิ่งขึ้นในการวิเคราะห์ ช่วยลดโอกาสที่จะเกิดการแจ้งเตือนที่ไม่ถูกต้องและการหลีกเลี่ยง IDS ได้

บทที่ 3
การวิเคราะห์การบุกรุกบนเว็บโดยใช้ล็อกเว็บพร็อกซี

3.1 วิธีดำเนินการวิจัย

งานวิจัยนี้ดำเนินการวิจัยในรูปแบบของการวิจัยทดลองโดยตั้งสมมุติฐานเริ่มแรก คือในการบุกรุกโดยเฉพาะการบุกรุกในปัจจุบันมักมีขั้นตอนตามลำดับ ดังนี้ ผู้บุกรุกนิยมใช้วิธีการสำรวจ (survey) เป้าหมายก่อนเสมอ แล้วจึงทำการโจมตี (attack) และเก็บเกี่ยวผลประโยชน์ (exploit) จากเป้าหมายนั้น ๆ ในที่สุด ซึ่งการลำดับขั้นตอนการบุกรุกดังกล่าวนี้เอง ทำให้เป็นสาเหตุหนึ่งที่ทำให้เกิดระบบ IDS ในปัจจุบันเพื่อที่จะทำการตรวจสอบเหตุการณ์ที่มีแนวโน้มเป็นการการบุกรุกโดยเฉพาะในขั้นตอนของการสำรวจหรือการโจมตี ถ้าระบบสามารถตรวจจับขั้นตอนการสำรวจของผู้บุกรุกได้ ก็จะมีโอกาสในการหาป้องกันขั้นตอนการโจมตีหรือการหาประโยชน์ของผู้บุกรุก แม้ว่าระบบสามารถตรวจจับขั้นตอนการโจมตีได้ ก็ยังมีโอกาสในการป้องกันการเก็บเกี่ยวผลประโยชน์ได้ทัน ขึ้นอยู่กับช่วงเวลาที่ระบบสามารถตรวจสอบและเวลาที่ผู้บุกรุกใช้ในการปฏิบัติการทั้งหมด ดังนั้นจะเห็นได้ว่าหากระบบสามารถตรวจสอบการบุกรุกในแต่ละขั้นตอนได้แล้ว จะเกิดประโยชน์อย่างมหาศาลในการแก้ไขและป้องกันการบุกรุกในช่องทางนั้น ๆ แต่ทว่าการตรวจจับการบุกรุกจะไม่เกิดประโยชน์ขึ้นเลย หากการตรวจจับการบุกรุกนั้นมีข้อผิดพลาด (false alert) ซึ่งนอกจากจะไม่เกิดประโยชน์ขึ้นแล้ว ยังอาจก่อให้เกิดโทษที่ตามมาด้วย เช่น ระบบการวิเคราะห์ทำงานหนักโดยไม่มีความจำเป็น ปริมาณการตรวจสอบการบุกรุกที่ผิดพลาดจำนวนมากบดบังการตรวจสอบการบุกรุกที่ถูกต้องซึ่งมักมีจำนวนน้อยกว่าและทำให้ผู้บริสุทธิ์ต้องกลายเป็นผู้บุกรุก เป็นต้น ดังนั้นการวิเคราะห์การบุกรุกจึงนับว่าเป็นกระบวนที่สำคัญเป็นอย่างยิ่งในขั้นตอนการตรวจจับการบุกรุกทั้งหมด ปัจจุบันจึงมีงานวิจัยที่ได้มุ่งพัฒนาการวิเคราะห์การบุกรกให้มีประสิทธิภาพมากยิ่งขึ้นเป็นจำนวนมาก จากส่วนประกอบของระบบ IDS ทั้งหมด ในหลายงานวิจัยในมุ่งไปยังการรวบรวมข้อมูล (event generators) ก่อนที่จะนำไปวิเคราะห์ให้ได้มากที่สุด โดยการหาความสัมพันธ์ของเหตุการณ์ที่สงสัยว่าเป็นการบุกรุกกับเหตุการณ์อื่น ๆ ประกอบกัน การหาความสัมพันธ์ของแต่ละเหตุการณ์สามารถรวบรวมได้จากหลายแหล่งข้อมูล เช่น ข้อมูลจากล็อกไฟล์ IDS ต่างชนิดกัน [4] [5] [6] ข้อมูลที่มีลักษณะต่างระดับเลเยอร์กัน [7] [10] เป็นต้น ในงานวิจัยนี้จะได้ทดลองนำข้อมูลระดับแอพพลิเคชันเลเยอร์จากอุปกรณ์ที่ทำงานครอบคลุมทั้งเครือข่าย (เว็บพร็อกซีบนเครื่องเกตเวย์) มาหาความสัมพันธ์กับข้อมูลระดับแอพพลิเคชันเลเยอร์จากอุปกรณ์การตรวจสอบภายในเครือข่าย (Network-based IDS ภายในเครือข่าย) แล้วนำข้อมูลทั้งหมดไปวิเคราะห์เพื่อจำแนก

ประเภทของการแจ้งเตือนการบุกรุก โดยเฉพาะอย่างยิ่งการแจ้งเตือนการบุกรุกที่ไม่ประสบความสำเร็จซึ่งมีปริมาณมากในปัจจุบัน

ในงานวิจัยนี้จะทดลองจำลองการบุกรุกไปยังเว็บเซิร์ฟเวอร์ซึ่งอยู่ภายในเครือข่ายที่มีรูปแบบระบบรักษาความปลอดภัยดังกล่าว โดยแบ่งการทดลองออกเป็นสองส่วน คือ ส่วนแรกจะจำลองการบุกรุกโดยใช้เครื่องมือทดสอบข้อบกพร่อง (vulnerable scanner) ที่ได้รับความนิยมโดยลำพัง และส่วนหลังจะจำลองการบุกรุกในสถานการณ์จริง กล่าวคือ มีทั้งการเรียกดูเว็บไซต์โดยทั่วไปตามปกติผสมผสานกับมีการบุกรุกบนเว็บเกิดขึ้นไปพร้อม ๆ กัน และในระบบรักษาความปลอดภัยจะมีโปรแกรมที่พัฒนาขึ้นเพื่อนำล็อกไฟล์จากเว็บพร็อกซีและ Network-based IDS มาจัดเก็บลงสู่ฐานข้อมูล จากนั้นเมื่อต้องการวิเคราะห์การบุกรุกก็จะทำการค้นหาเหตุการณ์ที่สงสัยว่าจะเป็นการบุกรุกบนเว็บแล้วนำไปหารายละเอียดเพิ่มเติมจากฐานข้อมูลล็อกเว็บพร็อกซี แล้วนำรายละเอียดเพิ่มเติมที่ได้มาเข้าอัลกอริทึมเพื่อพิจารณาชนิดของ alert นั้น ๆ ต่อไป

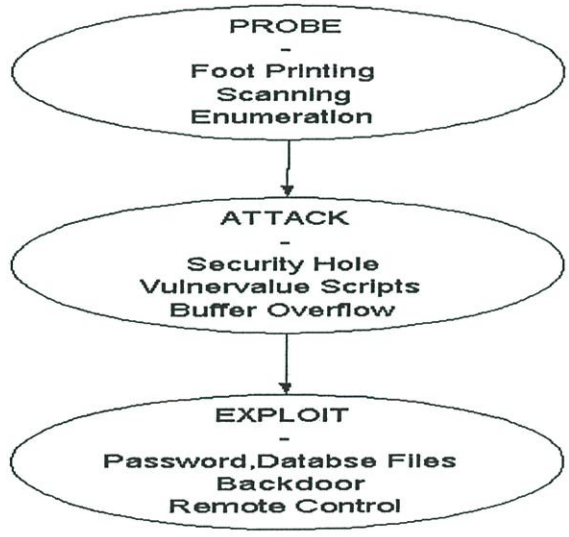
3.2 แนวคิดและทฤษฎีที่ใช้ในงานวิจัย

จากรูปแบบระบบรักษาความปลอดภัยพื้นฐาน คือ โครงสร้างเครือข่ายประกอบด้วยเว็บเซิร์ฟเวอร์อย่างน้อยหนึ่งเซิร์ฟเวอร์บริการข้อมูลข่าวสารบนเว็บอยู่ภายในเครือข่าย โดยระหว่างเครือข่ายภายในที่ให้บริการเว็บกับเครือข่ายภายนอกซึ่งมีการเรียกดูข้อมูลข่าวสารผ่านเว็บนั้น จะต้องทำการติดตั้งเว็บพร็อกซีเซิร์ฟเวอร์บนตำแหน่งที่เป็นเกตุเวย์ของเครือข่ายเพื่อทำให้สามารถบันทึกการร้องขอไฟล์บนเว็บได้ทั้งหมด และภายในเครือข่ายเดียวกับเว็บเซิร์ฟเวอร์จะต้องติดตั้งระบบ Network-based IDS เพื่อตรวจสอบและบันทึกเหตุการณ์ที่น่าสงสัยว่าจะเป็นการบุกรุกบนเว็บ ซึ่งระบบโครงสร้างเครือข่ายรูปแบบนี้มักเป็นระบบรักษาความปลอดภัยบนเครือข่ายที่ได้รับความนิยมในปัจจุบัน จากการศึกษางานวิจัย [5] [6] ทำให้เล็งเห็นประโยชน์ของการทำล็อกไฟล์จากระบบ IDS ต่างชนิดกันมาหาความสัมพันธ์กัน ทำให้มีข้อมูลที่ใช้ในการวิเคราะห์การบุกรุกที่กว้างมากยิ่งขึ้น ส่งผลให้เกิดความแม่นยำในการวิเคราะห์การบุกรุกมากยิ่งขึ้น

จากการสังเกตที่ว่าหากมีการบุกรุกบนเว็บบนระบบเครือข่ายดังกล่าว (เว็บพร็อกซีและ Network-based IDS) มีผลทำให้การร้องขอทั้งรูปแบบปกติและรูปแบบที่เป็นการบุกรุกบนเว็บจะต้องถูกบันทึกโดยเว็บพร็อกซีก่อนเสมอ เนื่องจากทุกแพ็กเก็ตที่จะผ่านเข้าไปยังเครือข่ายภายในนั้นจะต้องผ่านทางเว็บพร็อกซีนี้ก่อนเสมอ หลังจากนั้นเมื่อมีการบุกรุกบนเว็บเกิดขึ้นบนเว็บเซิร์ฟเวอร์ซึ่งไม่ว่าการบุกรุกนั้นจะสำเร็จหรือไม่ หากมีการบุกรุกหรือมีรูปแบบการร้องขอไฟล์ตามปกติซึ่งบังเอิญไปตรงกับรูปแบบการบุกรุก Network-based IDS ก็จะทำการบันทึกลงล็อกไฟล์ไว้เช่นกัน ประกอบกับประโยชน์จากงานวิจัยขึ้นต้นที่นำล็อกไฟล์ของ IDS ต่างชนิดกันมาวิเคราะห์ จึงก่อให้เกิดแนวคิดในการนำทั้งสองล็อกไฟล์จากเว็บพร็อกซีและ Network-based IDS มาทำการหาความ

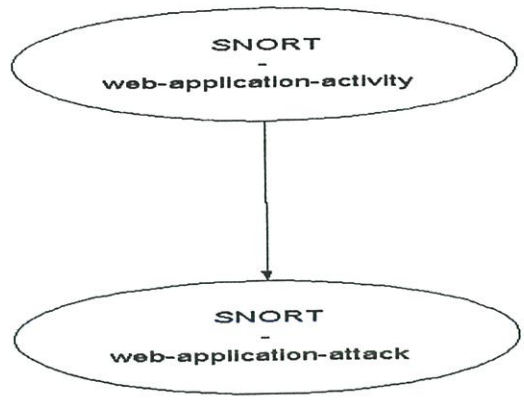
สัมพันธ์เพื่อค้นหาฟิลด์ในล็อกไฟล์ทั้งสองที่มาจากเหตุการณ์เดียวกัน ทำให้ในแต่ละเหตุการณ์จะมีข้อมูลที่บันทึกจากทั้งเว็บพร็อกซีและ Network-based IDS ซึ่งรายละเอียดที่บันทึกก็จะมีความแตกต่างกันไป ทำให้ในการวิเคราะห์การบุกรุกจะมีรายละเอียดของเหตุการณ์ที่สงสัยว่าจะเป็นการบุกรุกเพิ่มมากขึ้น รายละเอียดของเว็บพร็อกซีที่เป็นประโยชน์ในการวิเคราะห์การบุกรุก ได้แก่ เบอร์ไอพีต้นทางที่แท้จริงของการบุกรุกก่อนเข้าสู่เครือข่าย และสถานะตอบกลับจากเว็บเซิร์ฟเวอร์ ซึ่งรายละเอียดของการวิเคราะห์จะกล่าวถึงต่อไป

เนื่องจากในการบุกรุกจะมีลำดับขั้นตอนเพื่อจุดประสงค์ที่แตกต่างกันออกไป คือ การสำรวจ (scan) เพื่อตรวจสอบข้อบกพร่อง ตลอดจนรายละเอียดของเป้าหมาย, การโจมตี (attack) คือการโจมตีไปยังข้อบกพร่องเหล่านั้นเพื่อทำให้เป้าหมายขาดเสถียรภาพในการป้องกันตนเอง และการเก็บเกี่ยวผลประโยชน์ (exploit) คือการเรียกดูข้อมูลที่สำคัญ ๆและการเข้าไปงานในระบบเป้าหมายโดยไม่ได้รับอนุญาต ดังรูปในรูปที่ 3.1



รูปที่ 3.1 ลำดับขั้นตอนการบุกรุกโดยทั่ว ๆ ไป

จากขั้นตอนการบุกรุกนี้เองทำให้ในการวิเคราะห์การบุกรุกได้มีการกำหนดค่าระดับความรุนแรง (priority) ของแต่ละเหตุการณ์กำกับเสมอ เช่น ในกรณีของการบุกรุกบนเว็บ snort ซึ่งเป็นโปรแกรม Network-based IDS ที่ได้รับความนิยม กำหนดระดับความรุนแรงไว้สองระดับ คือ การบุกรุกทางเว็บแบบสำรวจ (web-application-activity) ซึ่งมีระดับความรุนแรงเป็น 2 ซึ่งน้อยกว่าการบุกรุกทางเว็บแบบโจมตีและแบบเก็บเกี่ยวผลประโยชน์ (web-application-attack) ซึ่งมีค่าระดับความรุนแรงเป็น 1 โดยในโปรแกรมการวิเคราะห์การบุกรุก [15] [16] นิยมแสดงผลลัพธ์โดยเรียงตามลำดับความรุนแรงของเหตุการณ์ด้วย ระดับความรุนแรงของ snort แสดงดังรูปที่ 3.2



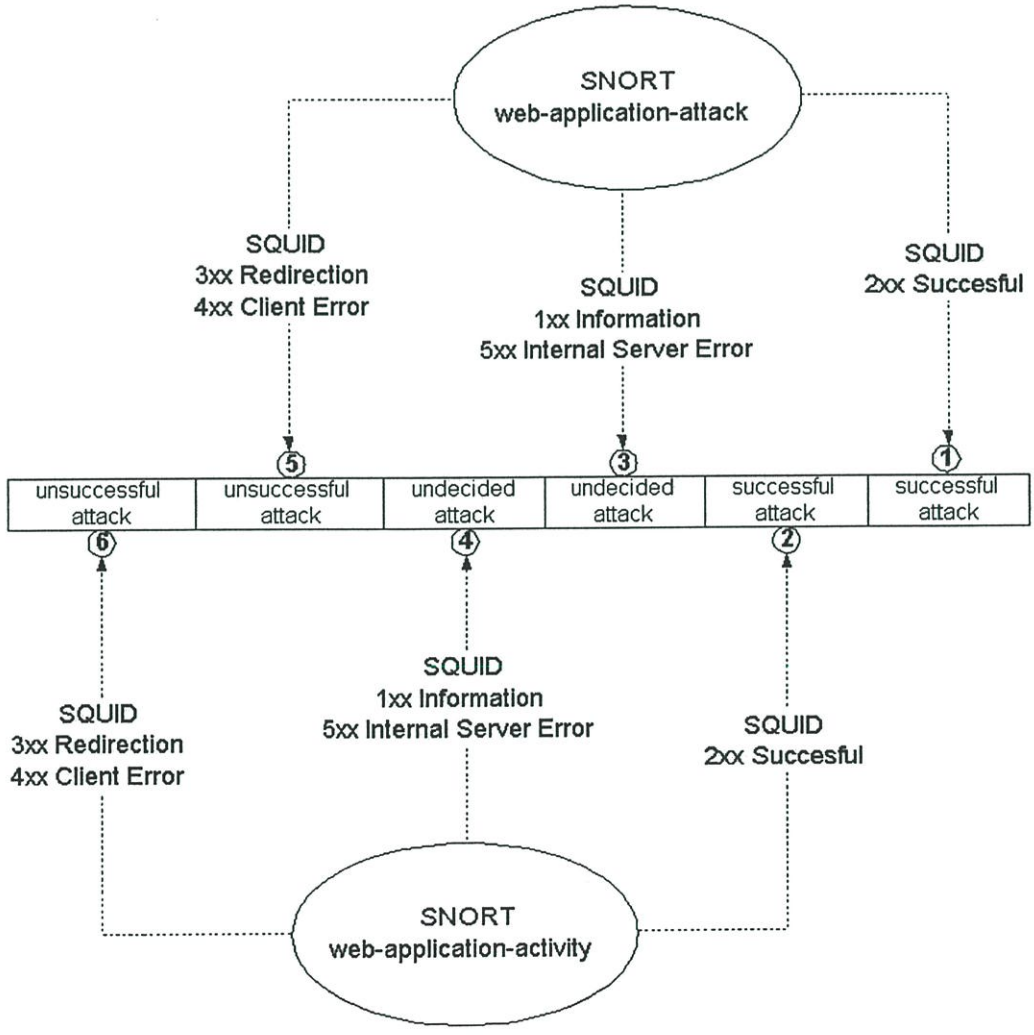
รูปที่ 3.2 ระดับความรุนแรงของการบุกรุกบนเว็บของ snort

บางครั้งมีการบุกรุกที่มีลักษณะซับซ้อนเป็นจำนวนมาก ทำให้ค่าระดับความรุนแรงนั้นอาจไม่เพียงในการแยกแยะระดับการบุกรุก ดังนั้นในงานวิจัยนี้จึงได้ใช้ประโยชน์จากข้อมูลของเว็บเซิร์ฟเวอร์ในส่วนของค่าตอบกลับของเว็บเซิร์ฟเวอร์ (return code) มากำหนดเป็นระดับความรุนแรงของการบุกรุกเพิ่มอีกส่วนหนึ่ง โดยกำหนดให้การบุกรุกที่มีค่า return code เป็น 2xx (Successful) เป็นการบุกรุกที่ประสบความสำเร็จ (successful attack) มีระดับความรุนแรงสูง ส่วนการบุกรุกที่มีค่า return code เป็น 1xx (Information) และ 5xx (Internal Server Error) เป็นการบุกรุกที่ไม่สามารถตีความได้ว่าประสบความสำเร็จหรือไม่

ตารางที่ 3.1 แสดงระดับความรุนแรงการบุกรุกบนเว็บเมื่อใช้ล็อกไฟล์ร่วมกัน

alert level	alert type	return code (SQUID)	classification (SNORT)
1	successful attack	2xx (Successful)	web-application-attack
2	successful attack	2xx (Successful)	web-application-activity
3	undecided attack	1xx (Information) 5xx (Internal Server Error)	web-application-attack
4	undecided attack	1xx (Information) 5xx (Internal Server Error)	web-application-activity
5	unsuccessful attack	3xx (Redirection) 4xx (Client Error)	web-application-attack
6	unsuccessful attack	3xx (Redirection) 4xx (Client Error)	web-application-activity

เนื่องจากมีข้อมูลไม่เพียงพอจะกำหนดให้เป็นการบุกรุกแบบปกติ (undecided attack) และจัดการบุกรุกที่มีค่า return code เป็น 3xx (Redirection) และ 4xx (Client Error) จัดว่าเป็นการบุกรุกที่ไม่ประสบความสำเร็จ ไม่น่าจะมีอันตรายใด ๆ (unsuccessful attack) เนื่องมาจากการบุกรุกนั้นไม่เกิดขึ้นจริงบนเครื่องเป้าหมาย เพราะ 3xx ถือว่าเป็นการนำข้อมูลจากพร็อกซีเซิร์ฟเวอร์นั้นแทน ไม่เกิดการกระทำที่เว็บเซิร์ฟเวอร์นั้นจริง ๆ และ 4xx ถือว่าเป็นการบุกรุกที่ไม่เกิดขึ้นจริง เช่น อาจไม่มีข้อบกพร่องดังกล่าวที่เว็บเซิร์ฟเวอร์ สรุปการวิเคราะห์การบุกรุกในงานวิจัยนี้เมื่อนำล็อก snort และล็อก squid มาวิเคราะห์ร่วมกันสามารถแบ่งระดับความรุนแรงได้ดังตารางที่ 3.1 และจากตารางที่ 3.1 สามารถนำการกำหนดระดับความรุนแรงของการใช้ล็อกไฟล์ snort และ squid ร่วมกันมาแสดงความสัมพันธ์ได้ดังรูปที่ 3.3 ซึ่งกำหนดให้ตัวเลขในวงกลมแสดงระดับความรุนแรง โดยให้ตัวเลขที่มีค่าน้อย (1) มีระดับความรุนแรงสูงกว่าตัวเลขค่ามาก ๆ (6)



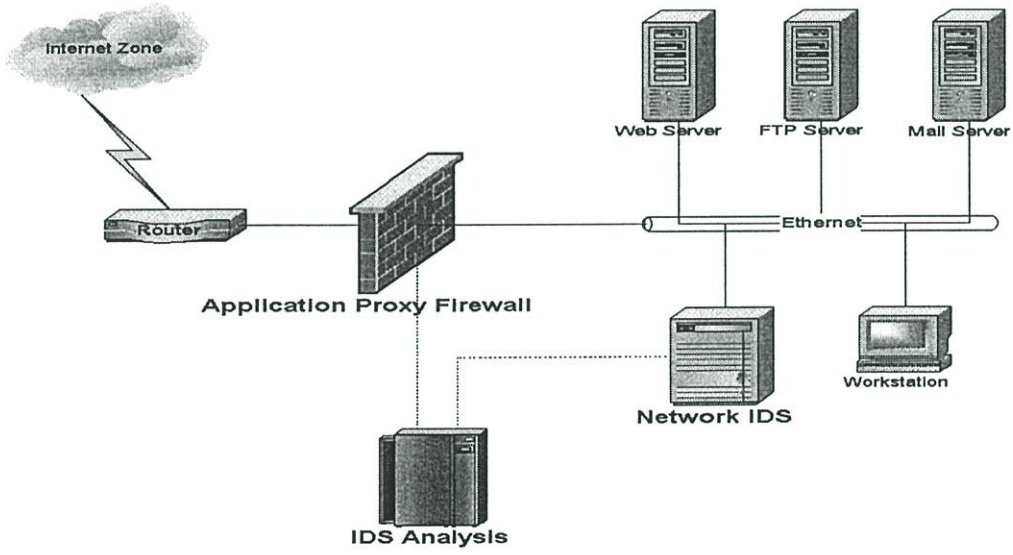
รูปที่ 3.3 ความสัมพันธ์ของระดับความรุนแรงเมื่อวิเคราะห์โดยใช้ล็อกไฟล์ร่วมกัน

จากตารางและรูปจะเห็นได้ว่าเราสามารถจำแนกระดับความรุนแรงได้ถึง 6 ระดับ ทำให้ผลการวิเคราะห์ที่ได้เราสามารถเรียงลำดับการบุกรุกได้อย่างละเอียดมากยิ่งขึ้น

3.3 การเตรียมข้อมูลสำหรับทำการทดลอง

3.3.1 ลักษณะข้อมูล การเลือกข้อมูล และเหตุผลในการเลือกข้อมูล

งานวิจัยนี้เป็นการวิเคราะห์การบุกรุกโดยการรวบรวมข้อมูลเหตุการณ์ที่สัมพันธ์กับการบุกรุกให้ได้มากที่สุด โดยรวบรวมข้อมูลจากรูปแบบหนึ่งของระบบรักษาความปลอดภัยบนเว็บเซิร์ฟเวอร์ในปัจจุบัน ที่นิยมติดตั้งเว็บพร็อกซีไว้เป็นด่านแรกของเครือข่าย โดยทำหน้าที่เปรียบเสมือนเป็นไฟร์วอลล์คั่นกลางระหว่างเครือข่ายภายในและเครือข่ายภายนอก นอกจากนี้ยังทำหน้าที่เป็นแคชเซิร์ฟเวอร์เพื่อช่วยเสริมประสิทธิภาพในการเรียกดูเว็บไซต์ที่ซ้ำ ๆ กัน ส่วนภายในเครือข่ายเองก็จะติดตั้ง Network- based IDS เพื่อตรวจสอบแพ็กเก็ตข้อมูลที่น่าสงสัย ดังแสดงในรูปที่ 3.4



รูปที่ 3.4 ระบบรักษาความปลอดภัยบนเครือข่าย

หากมีการบุกรุกทางเว็บเข้ามายังเว็บเซิร์ฟเวอร์ซึ่งอยู่ภายในเครือข่าย การร้องขอที่เป็นการบุกรุกก็จะผ่านเข้ามายังเว็บ พร็อกซีก่อนเสมอ เพื่อใช้เว็บพร็อกซีเป็นตัวกลางในการเชื่อมต่อไปยังเว็บเซิร์ฟเวอร์ที่แท้จริงต่อไป ไม่ว่าการร้องขอที่เป็นการบุกรุกนั้นจะสำเร็จหรือไม่ เว็บพร็อกซีก็จะบันทึกการร้องขอนั้นลงล็อกไฟล์รวมถึงสถานะตอบกลับของเว็บเซิร์ฟเวอร์ที่แท้จริงด้วย เมื่อมี

การร้องขอและเว็บพร็อกซีทำการเชื่อมต่อไปยังตัวเว็บเซิร์ฟเวอร์ที่แท้จริง แพ็กเก็ตการร้องขอที่ผ่านเข้ามายังเครือข่ายภายในก็จะถูก Network-based IDS ตรวจจับและนำไปเปรียบเทียบกับรูปแบบการบุกรุก ถ้ามีรูปแบบที่ตรงกันก็จะทำการบันทึกการร้องขอนั้นว่าเป็นการบุกรุกโดยไม่สนใจว่าเว็บเซิร์ฟเวอร์นั้นจะปฏิบัติตามการร้องขอนั้นได้สำเร็จหรือไม่ ดังนั้นเมื่อมีการวิเคราะห์การบุกรุกก็จะนำการแจ้งเตือนเหล่านี้ไปวิเคราะห์ด้วย ซึ่งผลลัพธ์ที่ได้บางส่วนก็จะเป็นการบุกรุกไปยังเหตุการณ์ที่ยังไม่เกิดขึ้นจริง (unsuccessful attack) หากผู้บุกรุกใช้เครื่องมือช่วยในการตรวจสอบข้อบกพร่องของเว็บเซิร์ฟเวอร์ ก็จะมีแพ็กเก็ตการร้องขอจำนวนมากที่เว็บเซิร์ฟเวอร์ไม่สามารถปฏิบัติได้จริง อันเนื่องมาจากไม่มีข้อบกพร่องดังกล่าวบนเว็บเซิร์ฟเวอร์นั้นๆ แต่ IDS ก็จะทำการบันทึก alert ลงล็อกไฟล์ ทำให้เวลานำล็อกไฟล์เหล่านี้ไปวิเคราะห์ก็จะยิ่งเกิดปัญหาในการจัดการกับการบุกรุกมากขึ้นเป็นทวีคูณ ดังนั้นหากทราบสถานะของการร้องขอ (return code) ที่มีรูปแบบตรงตาม signatures ก็จะทำให้ทราบถึงสถานะของการบุกรุกนั้น ซึ่งจะเกิดประโยชน์ในตอนวิเคราะห์การบุกรุก ซึ่งสามารถกำหนดระดับความรุนแรงที่แตกต่างกันได้ เช่น การกำหนดให้การบุกรุกที่กระทำการสำเร็จ มีการตอบรับจากเว็บเซิร์ฟเวอร์ (200-OK) ถือว่าเป็นการบุกรุกที่มีความรุนแรงสูงกว่าการบุกรุกที่กระทำการไม่สำเร็จ ถูกปฏิเสธจากเว็บเซิร์ฟเวอร์ (404-File Not Found) หรือการบุกรุกในเหตุการณ์ซ้ำ ๆ (304-Not Modified) ซึ่งไม่มีผลต่อการบุกรุกกับเว็บเซิร์ฟเวอร์จริง ๆ เนื่องมาจากเป็นข้อมูลชุดเดิมจากเว็บเซิร์ฟเวอร์ (ไม่มีการบุกรุกเพิ่มเติม) เป็นต้น

ดังนั้นลักษณะข้อมูลที่ดีในการทดลองประกอบงานวิจัยนี้ จะต้องเป็นการบุกรุกเว็บโดยทั่วไปอาจใช้เครื่องมือในการบุกรุกหรือกระทำการบุกรุกโดยใช้เว็บเบราเซอร์ธรรมดา ๆ ซึ่งการบุกรุกนั้นจะต้องกระทำผ่านเว็บพร็อกซีและถูกบันทึกลงล็อกไฟล์ access ของ squid และเมื่อผ่านเข้ามายังเครือข่ายภายใน การบุกรุกนั้นก็จะต้องถูกกำหนดไว้แล้วในรูปแบบ signature ของ snort ทำให้เมื่อมีการตรวจสอบการบุกรุก ก็จะทำการบันทึกลงล็อกไฟล์ alert ของ snort ซึ่งในการทำงานของโปรแกรมหาความสัมพันธ์นี้ จะคัดเลือกเฉพาะการบุกรุกที่เกี่ยวข้องกับเว็บเท่านั้น โดยดูได้จากลักษณะ message ของการแจ้งเตือนของ snort จะต้องขึ้นด้วย web เท่านั้น เนื่องจากระบบพร็อกซีที่ใช้ในปัจจุบันนิยมใช้เฉพาะรูปแบบบนเว็บเท่านั้น ดังนั้นในการคัดเลือกล็อกที่จะมาหาความสัมพันธ์จึงคัดเลือกเฉพาะล็อกที่สงสัยเฉพาะเหตุการณ์ที่เกี่ยวข้องกับการบุกรุกบนเว็บเท่านั้น

3.3.2 เครื่องมือและวิธีการ

ในการทดลองกำหนดให้เครือข่ายภายในประกอบด้วยเครื่องเซิร์ฟเวอร์ทำหน้าที่ให้บริการเว็บข้อมูลข่าวสารแก่ผู้ใช้ทั่วไปและมีเครื่องเซิร์ฟเวอร์ทำหน้าที่เป็น Network-based IDS โดยติดตั้งโปรแกรม snort เพื่อตรวจสอบแพ็กเก็ตที่เข้ามาเรียกใช้บริการเว็บเซิร์ฟเวอร์ กำหนดให้ที่

จุดเชื่อมต่อระหว่างเครือข่ายมีเครื่องเซิร์ฟเวอร์ที่ทำหน้าที่เป็นเว็บพร็อกซีระหว่างสองเครือข่าย โดยบังคับการเรียกใช้บริการเว็บที่พอร์ต 80 ของเว็บเซิร์ฟเวอร์ภายในให้เปลี่ยนไปยังพอร์ต 3128 ของเว็บพร็อกซีเซิร์ฟเวอร์โดยอัตโนมัติ (Transparent Proxy) เพื่อให้เว็บพร็อกซีเซิร์ฟเวอร์สามารถบันทึกการร้องขอทั้งหมดได้ ในการทดลองประกอบงานวิจัยนี้กำหนดให้เว็บเซิร์ฟเวอร์มีข้อบกพร่องทางเว็บ 12 จุด (vulnerable) และสร้างสถานะการณ์การบุกรุกโดยใช้เครื่องมือเพียงลำพัง เครื่องมือทดสอบข้อบกพร่องที่ได้รับความนิยมหลาย ๆ ชนิดในการทดสอบ ได้แก่ Nessus [18], Internet Scanner [19], Shadow Security Scanner [20], N-Stealth Security Scanner [21] โดยกำหนดให้มีการเทคนิคการหลีกเลี่ยง IDSประกอบกัน (N-Stealth with ids evasion technique) รวมถึงการใช้เว็บเบราเซอร์จำลองการบุกรุกเบื้องต้น นอกจากนี้ยังได้จำลองสถานการณ์การบุกรุกสถานการณ์จริงในปัจจุบันซึ่งมีการเรียกใช้งานเว็บตามปกติผสมกับการบุกรุกทางเว็บ โดยที่ในแต่ละเครื่องมือจะกำหนดให้มีการทดสอบข้อบกพร่องทั้งแบบทั่วไปและแบบที่เกี่ยวข้องเว็บเท่านั้น

การจำลองสถานการณ์การบุกรุกโดยใช้เครื่องมือตรวจสอบข้อบกพร่องที่ได้รับความนิยมโดยลำพังนี้ เพื่อเป็นการทดลองการทำงานของโปรแกรมที่ได้เขียนขึ้นมาเพื่อทำการหาความสัมพันธ์ระหว่างล็อกไฟล์ทั้งสอง เมื่อได้ทดลองและวิเคราะห์ผล แล้วนำส่วนที่ผิดพลาดไปแก้ไขโปรแกรม จนผลการวิเคราะห์ออกมาถูกต้องตามต้องการแล้ว จึงจำลองสถานการณ์การบุกรุกในรูปแบบสถานการณ์จริงที่มีการใช้งานจริงผสมกับการบุกรุก โดยการจำลองสถานการณ์แบบเหมือนจริงนี้ จะจำลองสถานการณ์เป็นช่วง ๆ โดยดูตามปริมาณข้อมูลที่ผ่านเข้าออกเว็บเซิร์ฟเวอร์โดยประมาณ โดยกำหนดตามขนาดของล็อกไฟล์ access ของ squid เป็นดังนี้ ข้อมูลล็อกไฟล์ access มีขนาดประมาณ 1 ล้านไบต์, 5 ล้านไบต์, 10 ล้านไบต์, 25 ล้านไบต์, 50 ล้านไบต์และ 100 ล้านไบต์ตามลำดับ เมื่อข้อมูลมีขนาดที่แตกต่างกันทำให้เราสามารถคำนวณเวลาที่ใช้ในการวิเคราะห์หาความสัมพันธ์ร่วมล็อกไฟล์ได้อย่างเหมาะสม

3.4 ขั้นตอนในการรวบรวมข้อมูล

3.4.1 การออกแบบการทดลอง

เนื่องจากในงานวิจัยนี้เป็นการทดสอบเฉพาะการบุกรุกบนเว็บเท่านั้น ดังนั้นจึงต้องมีการปรับ snort ให้ทำการตรวจสอบเฉพาะการบุกรุกที่เกี่ยวข้องกับเว็บเท่านั้น ซึ่งหากพิจารณาตรวจสอบเฉพาะการบุกรุกบนเว็บจะมีรูปแบบที่เกี่ยวข้องทั้งหมด 978 signatures โดยแบ่งออกตามประเภทได้ 2 classifications คือ การบุกรุกทางเว็บแบบสำรวจ (web-application-

activity) ที่มีระดับความรุนแรงเป็น 2 ซึ่งน้อยกว่าการบุกรุกทางเว็บแบบโจมตี (web-application-attack) ที่มีระดับความรุนแรงเป็น 1 ดังแสดงดังรูปที่ 3.5

```
[root@athlon rules]# grep ^alert web*.rules|wc -l
978
[root@athlon etc]# grep web classification.config |wc -l
2
[root@athlon etc]# grep web classification.config
config classification: web-application-activity,access to a potentially vulnerable web application,2
config classification: web-application-attack,Web Application Attack,1
```

รูปที่ 3.5 ค่าต่าง ๆ ของ snort ที่เกี่ยงข้องกับการบุกรุกบนเว็บ

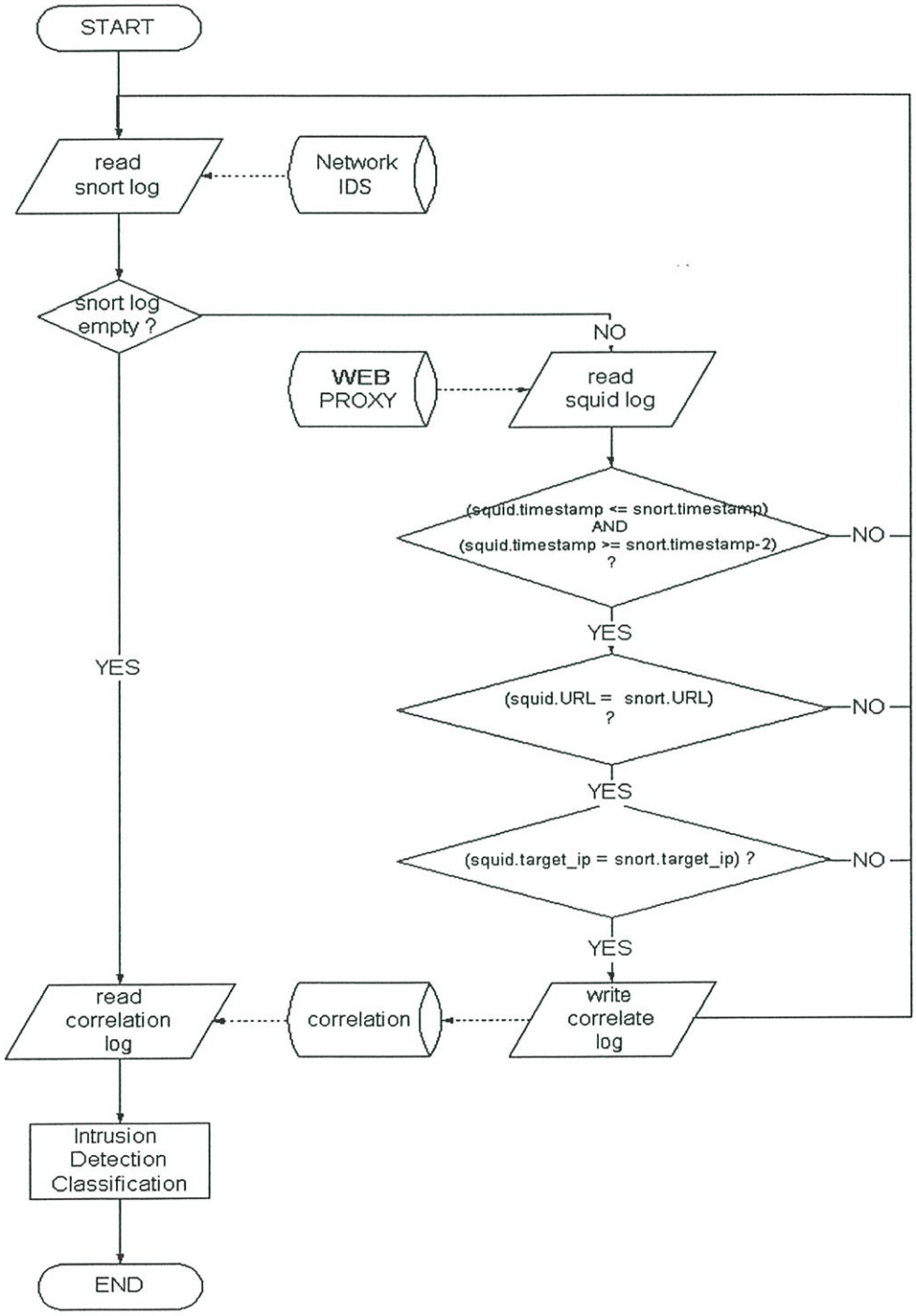
ในการพิจารณา alert ใดว่ามีโอกาสน่าจะเป็น unsuccessful attack alert นั้นจำเป็นที่จะต้องนำข้อมูลของ squid มาพิจารณาด้วย โดยการค้นหาสถานะที่แท้จริงของการ alert นั้น ๆ การค้นหาสถานะที่แท้จริงของการบุกรุกสามารถกระทำได้โดยการนำค่า timestamp, target_ip และ URL จากล็อก snort ไปเปรียบเทียบกับบางฟิลด์ในล็อก squid เพื่อค้นหาบันทึกการ access ของ squid ที่ก่อให้เกิดการ alert ใน snort

สาเหตุที่ไม่สามารถนำค่า timestamp จากล็อก snort ไปเปรียบเทียบกับ timestamp ของล็อก squid ได้โดยตรง คือ เนื่องจากล็อกทั้งสองอยู่บนอุปกรณ์เครือข่ายคนละจุดกัน (เว็บพร็อกซี และ Network-based IDS) ทำให้ timestamp ที่แพ็กเก็ตข้อมูลเดินทางผ่านแต่ละจุดจะต้องแตกต่างกัน ซึ่งโดยปกติแล้ว ภายในสภาวะของอีเทอร์เน็ต 100 เมกบิตต่อวินาที จะยอมให้มีการล่าช้า (delay) ของแพ็กเก็ตภายในเครือข่าย (จากเว็บเซิร์ฟเวอร์ไป Network-based IDS) ไม่ควรเกิน 2 วินาที [9] จึงจะถือว่าทั้งสองแพ็กเก็ตเป็นแพ็กเก็ตเดียวกัน ส่วนค่า URL ซึ่งตามปกติไม่ปรากฏในล็อกของ snort นั้น เราจำเป็นที่จะต้องนำค่าฟิลด์คำอธิบายการบุกรุก (msg) ของ snort ไปเปรียบเทียบกับฟิลด์ msg ของรูปแบบการบุกรุก เพื่อค้นหารูปแบบ URL ที่เป็นการบุกรุกต่อไป

การหาความสัมพันธ์ที่เหมือนกันของ snort alert กับ squid access จะต้องมีคุณสมบัติดังต่อไปนี้ คือ URL และเบอร์ไอพีเป้าหมายต้องเหมือนกัน และมีช่วงเวลา (timestamp) แตกต่างกันไม่เกิน 2 วินาที เมื่อได้ข้อมูลการ access ที่เว็บเซิร์ฟเวอร์ของการ alert จาก snort แล้ว ทำให้การ alert นั้นจะมีข้อมูลส่วนที่เพิ่มเติม คือ สถานะของการร้องขอ (return_code ของ squid) และเบอร์ไอพีผู้บุกรุกที่แท้จริง (source_ip ของ squid) ซึ่งสามารถนำไปใช้ประโยชน์ในการวิเคราะห์ต่อไป สรุปความสัมพันธ์ที่เหมือนกันของเหตุการณ์ในล็อก snort และล็อก squid จะต้องมีคุณสมบัติดังต่อไปนี้

1. เวลาของ snort จะต้องเท่ากับหรือมากกว่าเวลาของ squid ไม่เกิน 2 วินาที
2. URL ของ snort และ URL ของ squid ต้องเหมือนกัน
3. target ip ของ snort และ squid ต้องเหมือนกัน

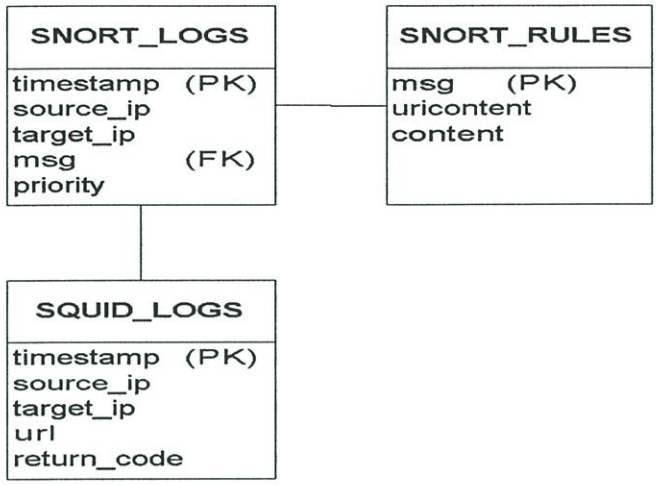
ซึ่งสามารถแสดงตัวอย่างล็อกไฟล์ squid และ snort ที่สัมพันธ์กัน ดังรูปที่ 3.6



รูปที่ 3.6 แสดงขั้นตอนการหาความสัมพันธ์ระหว่างล็อกไฟล์

3.4.2 การออกแบบฐานข้อมูล

เนื่องจากในการวิเคราะห์การบุกรุกที่ใช้ประกอบในงานวิจัยนี้จำเป็นที่จะต้องใช้ข้อมูลจากล็อก snort และล็อก squid ซึ่งมีรูปแบบล็อกไฟล์ที่แตกต่างกัน ดังนั้นก่อนนำข้อมูลทั้งสองมาหาความสัมพันธ์กัน จำเป็นที่จะต้องมีการปรับแต่งบางฟิลด์ให้มีชนิดข้อมูลที่ตรงกัน โดยจัดเก็บข้อมูลทั้งหมดในรูปแบบฐานข้อมูลเชิงสัมพันธ์ (Relation Database) กำหนดให้ฐานข้อมูลชื่อ IDSFW



รูปที่ 3.6 แสดง Entity Relation Diagram (ERD)

รูปที่ 3.6 แสดงความสัมพันธ์ระหว่างตาราง (Entity-Relation Diagram) ของฐานข้อมูล IDSFW ซึ่งประกอบด้วย 3 ตาราง ได้แก่

1. *ตาราง SNORT_RULES* สำหรับเก็บ URL ที่เป็นการบุกรุกและข้อความแจ้งเตือนของการบุกรุกนั้น ๆ ประกอบด้วย 3 ฟิลด์ ได้แก่ msg คือข้อความแจ้งเตือน, uricontent และ content คือรูปแบบ URL ที่มีลักษณะเป็นการบุกรุก
2. *ตาราง SQUID_LOGS* สำหรับเก็บข้อมูลการร้องขอ URL ผ่านเว็บพร็อกซีทั้งหมด ประกอบด้วย 5 ฟิลด์ ได้แก่ timestamp คือเวลาที่สิ้นสุดการร้องขอ, source_ip คือเบอร์ไอพีผู้บุกรุก, target_ip คือเบอร์ไอพีเป้าหมาย, URL คือการร้องขอ และ return_code คือผลตอบกลับของการร้องขอนั้น
3. *ตาราง SNORT_LOGS* สำหรับเก็บข้อมูลการแจ้งเตือนของ snort ประกอบด้วย 5 ฟิลด์ ได้แก่ timestamp คือเวลาที่เกิดการบุกรุก, source_ip คือเบอร์ไอพีผู้บุกรุก, target_ip คือเบอร์ไอพีเป้าหมาย, msg คือข้อความแจ้งเตือนและ priority คือระดับความรุนแรงของการบุกรุก

3.4.3 การเตรียมข้อมูล (data preprocessing)

เมื่อพิจารณา alert ในล็อกไฟล์ของ snort เห็นได้ว่า snort จะบันทึกเฉพาะข้อความแจ้งเตือน (msg) ไม่ได้มีการบันทึก URL ของการบุกรุกนั้น ๆ ดังนั้นในการที่นำฟิลด์ข้อมูลจาก snort ไปหาความสัมพันธ์กับการบุกรุกเดียวกันที่เกิดขึ้นใน squid จะต้องใช้ฟิลด์ URL ในการเปรียบเทียบด้วย ดังนั้นจึงต้องมีการเตรียมข้อมูลในส่วนของ msg และ URL ไว้ก่อน โดยการอ่านข้อมูลจากไฟล์ signature ที่เกี่ยวข้องกับการบุกรุกทางเว็บ เลือกเฉพาะฟิลด์ข้อความเตือน (msg) และ uricontent,content (URL) เก็บในตาราง SNORT_RULES

ต่อมาเป็นการอ่านข้อมูลทั้งหมดจากล็อก squid โดยเลือกเฉพาะฟิลด์ที่ใช้ในการทดลอง คือ timestamp, source_ip, target_ip, URL และ return_code จัดเก็บลงสู่ตาราง SQUID_LOGS ส่วนการอ่านข้อมูลจากล็อก snort ก็จะคัดเลือกเฉพาะฟิลด์ timestamp, source_ip, target_ip, msg และ priority จัดเก็บลงตาราง SNORT_LOGS เนื่องจากการเก็บ timestamp ของ squid เป็นรูปแบบ UTC ส่วน timestamp ของ snort เป็นรูปแบบ GMT ซึ่งในการทดลองกำหนดให้เลือกใช้เป็นแบบ UTC เป็นหลักเพื่อความรวดเร็วในการเรียงลำดับและค้นหา ดังนั้นในส่วนของฟิลด์ timestamp ของ snort จึงจำเป็นที่จะต้องใช้ฟังก์ชันในการแปลงเวลาจาก GMT เป็น UTC ก่อนจัดเก็บลงฐานข้อมูล

3.5 วิธีวิเคราะห์ข้อมูล

การวิเคราะห์การบุกรุกโดยใช้ฐานข้อมูล IDS และเว็บพร็อกซีที่สัมพันธ์กัน เราจะนำเสนอในรูปของใคร? ทำอะไร? กับใคร? อย่างไร? (Who? What? Whom? How?) โดยพิจารณาจากฟิลด์ข้อมูลดังต่อไปนี้

- source IP
- alert type
- target IP
- URL
- severity level

โดย source IP ทำให้เราทราบถึงที่มาของผู้บุกรุก (Who) alert type ทำให้เราทราบว่าผู้บุกรุกใช้การบุกรุกหรือการโจมตีแบบใด (What) target IP ทำให้เราทราบเป้าหมายหรือเหยื่อของการบุกรุก (Whom) , URL ทำให้เราทราบว่าลักษณะการบุกรุกเป็นไปในรูปแบบใด (How) และ severity level หรือระดับความรุนแรงของการบุกรุกซึ่งสามารถคำนวณได้จากอัลกอริทึมดังกล่าวข้างต้น

การพิจารณาสถานการร์การบุกรุกสามารถดูได้จากค่าข้อมูลที่อยู่ในฟิลด์ข้อมูลดังกล่าว ทำให้สามารถแบ่งได้เป็นหลาย ๆ สถานการณ์ดังนี้

Situation 1 แจ้งเตือนเมื่อตรวจสอบพบว่าทั้ง source, target และ alert มีค่าเหมือนกัน ใช้สำหรับตรวจสอบผู้บุกรุกที่โจมตีมายังการบริการเป้าหมายบนเครื่อง server เครื่องใดเครื่องหนึ่ง เช่น การ nuke

Situation 2 แจ้งเตือนเมื่อตรวจสอบพบว่า source และ target มีค่าตรงกัน ใช้สำหรับตรวจสอบผู้บุกรุกที่โจมตีมายังการบริการหลายๆอย่างบนเครื่องเซิฟเวอร์เครื่องเดียว เช่น การ scan port เป้าหมาย

Situation 3 แจ้งเตือนเมื่อตรวจสอบพบว่า target และ alert มีค่าตรงกัน ใช้สำหรับตรวจสอบผู้บุกรุกที่ใช้การโจมตีจากหลาย ๆ เครื่องมายังเครื่องเซิฟเวอร์เครื่องเดียว เช่น การ distributed ping of death

Situation 4 แจ้งเตือนเมื่อตรวจสอบพบว่า source และ alert มีค่าตรงกัน ใช้สำหรับตรวจสอบผู้บุกรุกที่ใช้การโจมตีไปยังเป้าหมายจำนวนมาก เช่น การโจมตี name server

Situation 5 แจ้งเตือนเมื่อตรวจสอบพบว่า source ที่ตรงกัน ใช้สำหรับตรวจสอบผู้บุกรุกที่ใช้การโจมตีหลายๆอย่างไปยังเป้าหมายจำนวนมาก เช่น การ scan หาข้อบกพร่องของ service ตาม network ต่าง ๆ

Situation 6 แจ้งเตือนเมื่อตรวจสอบพบว่า target ที่ตรงกัน ใช้สำหรับตรวจสอบผู้บุกรุกที่ใช้การโจมตีหลายๆอย่างไปยังเป้าหมายที่เดียว เช่น การ distributed attacks

Situation 7 แจ้งเตือนเมื่อตรวจสอบพบว่า alert ที่ตรงกัน ใช้สำหรับตรวจสอบกลุ่มผู้บุกรุกที่ใช้การโจมตีหลายๆอย่างไปยังเป้าหมายที่เดียว เช่น การร่วมมือกันของ hackers เพื่อโจมตีเว็บเป้าหมาย

การเกิดเหตุการณ์ Situation ใด เหตุการณ์หนึ่งได้เพียงหนึ่งเดียวเท่านั้น กล่าวคือ ถ้าเกิด Situation 1 แล้วจะไม่มีทางเกิด Situation 2 ได้อีก โดยประโยชน์ที่ได้จากข้อมูลที่ได้รับเพิ่มเติมจากเว็บพร็อกซี ได้แก่

- *รูปแบบการร้องขอไฟล์ (URL)* ใช้สำหรับตรวจสอบการแจ้งเตือนของ IDS ว่าเป็นลักษณะ false positives อันเนื่องมาจาก URL pattern ของการร้องขอไฟล์ตามปกติที่บังเอิญไปตรงกับ pattern ที่ถูกกำหนดไว้ใน signature ของ IDS เช่น กรณีผู้ใช้งานเรียกข้อมูลตามปกติไปยังเว็บเพื่อค้นหาคู่มือของคำสั่ง ดังนี้

  http://161.246.49.111/cgi-bin/manual.cgi?command=cmd.exe

  ซึ่งเป็นการขอเรียกดูรายละเอียดของคำสั่ง cmd.exe ผ่าน CGI ตามปกติ แต่เนื่องจาก IDS ใช้วิธีการ pattern matching ซึ่งบังเอิญไปตรงกับการบุกรุกทำให้ IDS แจ้งการบุกรุกมาเป็น WEB-IIS cmd.exe access เมื่อพิจารณาข้อมูล URL ของเว็บพร็อกซีในส่วนนี้ประกอบทำให้เราทราบว่าการแจ้งเตือนของ IDS ในครั้งนี้เป็นการแจ้งเตือนที่ไม่ถูกต้อง

- *ผลการร้องขอไฟล์ (return code)* ใช้สำหรับตรวจสอบผลที่ได้จากการบุกรุก เพื่อเราจะได้ทราบสถานะของการบุกรุก เช่น หากเป็นการบุกรุกผ่านทางเว็บและที่ไฟร์วอลล์ให้ผลการร้องขอไฟล์เป็น 404 (File Not Found) เราก็จะถือว่าการบุกรุกครั้งนี้ไม่ประสบความสำเร็จ มีระดับความอันตรายลดน้อยลงไป

นอกจากนี้ยังมีข้อมูลบางส่วนที่น่าจะเป็นประโยชน์ในการประกอบการพิจารณาเพิ่มเติม ได้แก่ IP ต้นทางที่แท้จริงและเวลาที่ใช้ในการประมวลผล (duration) เป็นต้น

บทที่ 4
รายงานผลการทดลอง

4.1 ขั้นตอนของการทดลองการบุกรุกบนเว็บ

1. เตรียมระบบเครือข่ายที่ภายในมีเว็บเซิร์ฟเวอร์ เครื่องที่ทำหน้าที่เป็น Network-based IDS ,เว็บพร็อกซีบนเกตเวย์ของเครือข่ายและเครื่องที่ใช้ในการบุกรุก
2. ทดลองการบุกรุกตามสถานการณ์ต่าง ๆ ที่กำหนด
3. ใช้โปรแกรมจัดเตรียมข้อมูลล็อก snort และ squid ลงสู่ฐานข้อมูล
4. ใช้โปรแกรมคำนวณหาความสัมพันธ์ระหว่างล็อก
5. ทำการวิเคราะห์การบุกรุกที่เกิดขึ้น

4.2 ผลการทดลอง

ในการทดลองนี้ได้กำหนดให้มีเว็บเซิร์ฟเวอร์โดยติดตั้งโปรแกรม Apache และ IIS และที่เว็บเซิร์ฟเวอร์ที่เป็นเป้าหมายในการบุกรุกนี้ กำหนดให้มีข้อบกพร่องทางเว็บ 12 จุด (vulnerability) โดยภายในเครือข่ายเดียวกันนี้มีเครื่องที่ทำหน้าที่เป็น Network-based IDS โดยติดตั้งโปรแกรม snort และบริเวณเกตุเวย์ของเครือข่ายมีเครื่องที่ทำหน้าที่เว็บเซิร์ฟเวอร์โดยติดตั้งโปรแกรม squid และบังคับการเรียกใช้บริการเว็บให้เปลี่ยนไปใช้บริการเว็บพร็อกซีก่อนเสมอ (transparent proxy) ในการทดลองแบ่งการทดลองออกเป็นสองช่วง คือ ช่วงแรกเป็นการจำลองสถานการณ์โดยใช้เครื่องมือเพียงลำพังและในช่วงหลังเป็นการจำลองสถานการณ์ใช้งานจริงที่มีการเรียกใช้เว็บไซต์ตามปกติผสมกับการใช้เครื่องมือในการบุกรุก

จากการทดลองจำลองการบุกรุกสถานการณ์ใช้เครื่องมือตรวจสอบข้อบกพร่องเพียงลำพัง โดยเลือกใช้เครื่องมือที่ได้รับความนิยม 4 ชนิด คือ Internet Scanner (ISS), Nessus, Shadow Security Scanner และ N-Stealth Security Scanner โดยที่ในแต่ละเครื่องมือจะกำหนดให้มีรูปแบบการโจมตีขั้นพื้นฐาน เน้นถึงรูปแบบโจมตีทางเว็บโดยเฉพาะซึ่งสามารถกำหนดจำนวนรูปแบบการโจมตี (scan rule) ได้แตกต่างกัน ซึ่งรูปแบบการโจมตีเหล่านี้จะถูกทดสอบไปยังเว็บเซิร์ฟเวอร์ซึ่งได้กำหนดให้มีข้อบกพร่องทางเว็บจำนวน 12 จุดในทุก ๆ สถานการณ์ (ข้อบกพร่องบางจุดสามารถใช้รูปแบบการโจมตีได้มากกว่าหนึ่งรูปแบบ และอาจสร้าง alert ที่เกี่ยวข้องได้มากกว่าหนึ่ง alert) โดยจำนวน access ทั้งหมดของเว็บพร็อกซี (squid log) และถูกนำมาวิเคราะห์ร่วมกับจำนวน alert ทั้งหมดของ Network-based IDS (snort log) การวิเคราะห์การบุกรุกจะพิจารณาข้อมูลจากล็อก snort เป็นหลักเชื่อมความสัมพันธ์ไปยังล็อก squid แล้ววิเคราะห์ว่า alert จาก snort นั้น ๆ ควรจัดอยู่ในประเภท alert ใด ดังต่อไปนี้ คือ การบุกรุกที่ประสบความสำเร็จ (successful

นั้น ๆ ควรจัดอยู่ในประเภท alert ใด ดังต่อไปนี้ คือ การบุกรุกที่ประสบความสำเร็จ (successful attack), การบุกรุกตามปกติ (undecided attack) และการบุกรุกที่ไม่ประสบความสำเร็จ (unsuccessful attack) ในการทดลองนี้จะมีการจำลองการใช้งานปกติผสมกับการบุกรุกโดยเครื่องมือต่าง ๆ

โดยในแต่ละขั้นตอนจะบันทึกเวลาที่ทำกระบวนการนั้นไว้เสมอ คือ เวลาที่ใช้ในการตรวจสอบข้อบุกรุกโดยใช้เครื่องมือชนิดต่าง ๆ (scan exploit), เวลาที่ใช้ในการแปลง squid access log ลงตาราง SQUID_LOGS (squid log), เวลาที่ใช้ในการแปลง snort alert log ลงตาราง SNORT_LOGS (snort log) และเวลาที่ใช้ในการหาความสัมพันธ์ระหว่าง snort และ squid รวมถึงการวิเคราะห์ชนิดของ alert (correlation analysis) หน่วยเวลาที่ใช้ในการบันทึกทั้งหมด คือ วินาที (second) ระหว่างการจำลองในแต่สถานการณ์ที่เว็บเซิร์ฟเวอร์, เว็บพร็อกซีเซิร์ฟเวอร์และเครื่องที่ทำหน้าที่เป็น IDS จะต้องมีการเคลียร์ข้อมูลทั้งในหน่วยความจำและล็อกไฟล์ทุกครั้งเสมอ

รายละเอียดฟิลด์ข้อมูลต่าง ๆ ในตารางต่อ ๆ ไป มีดังต่อไปนี้ คือ

- scan rule คือจำนวนรูปแบบการบุกรุกที่ใช้ในการทดลอง (บางเครื่องมืออาจรวมการบุกรุกขั้นพื้นฐานที่ไม่เกี่ยวข้องกับเว็บด้วย)
- squid log คือจำนวนล็อกทั้งหมดที่บันทึกในไฟล์ squid access
- snort alert คือจำนวน alert ทั้งหมดที่บันทึกในไฟล์ snort alert
- successful attack alert คือจำนวน alert ที่วิเคราะห์ว่าเป็นการบุกรุกที่แท้จริง อันเนื่องมาจากมีค่าตอบกลับจากเว็บเซิร์ฟเวอร์เป็น 2xx Successful ซึ่งแสดงว่าสามารถปฏิบัติตามการร้องขอนั้นได้ (การบุกรุกเกิดขึ้นจริง)
- undecided attack alert คือจำนวน alert ที่ไม่สามารถวิเคราะห์ได้ว่าเป็นการบุกรุกที่แท้จริงหรือไม่ เนื่องจากมีข้อมูลไม่ชัดเจนเพียงพอ อันเนื่องมาจากมีค่าตอบกลับจากเว็บเซิร์ฟเวอร์เป็น 1xx Information และ 5xx Internal Server Error
- unsuccessful attack alert คือจำนวน alert ที่วิเคราะห์ว่าไม่เป็นการบุกรุกที่แท้จริง อันเนื่องมาจากมีค่าตอบกลับจากเว็บเซิร์ฟเวอร์เป็น 3xx Redirection และ 4xx Client Error ซึ่งแสดงว่าเว็บเซิร์ฟเวอร์ไม่ได้ปฏิบัติตามการร้องขอนั้นจริง เช่น 304 Not Modified ข้อมูลเหมือนเดิมไม่จำเป็นต้องปฏิบัติซ้ำ และ 404 File Not Found ไม่มีไฟล์ตามการร้องขอนั้น

4.2.1 ผลการทดลองสถานการณ์การบุกรุกโดยใช้ ISS

Internet Scanner (ISS) เป็นเครื่องมือทดสอบการบุกรุกที่ได้รับความนิยมชนิดหนึ่ง ในการทดลองกำหนดให้มีการทดสอบการบุกรุกเฉพาะบนเว็บไปยังเครื่องเซิร์ฟเวอร์เป้าหมาย มีผลการทดลองดังต่อไปนี้

ตารางที่ 4.1 แสดงจำนวน alert ของการจำลองการบุกรุกโดยใช้ ISS

scan rule	squid log	snort alert	successful attack alert	undecided attack alert	unsuccessful attack alert
90	108	80	0	29	51

ตารางที่ 4.2 แสดงเวลาที่ใช้ในการจำลองการบุกรุกโดยใช้ ISS

scan exploit	SQUID LOGS	SNORT LOGS	correlation analysis	correlation per alert
1020	<1	<1	1	0.0125

จากข้อมูลในตารางที่ 4.1 และ 4.2 จะเห็นได้ว่าโปรแกรม ISS ใช้เวลาที่ใช้ในการตรวจสอบเครื่องเป้าหมายนานพอสมควร และยังมีรูปแบบการบุกรุกไปยังเว็บที่น้อยมากคือ มีเพียง 90 rules เท่านั้น โดย snort สามารถตรวจสอบการบุกรุกบนเว็บได้ 80 alert ซึ่งสามารถวิเคราะห์เป็น unsuccessful attack alert ได้มากถึง 51 alert คิดเป็น 51/80 = 64% ของ alert ทั้งหมด สาเหตุที่ไม่สามารถตรวจสอบ successful attack ได้เลย อันเนื่องมาจากการตรวจสอบของ ISS เป็นการตรวจสอบขั้นพื้นฐานเฉพาะใน directory cgi-bin เท่านั้น ซึ่งในปัจจุบันเว็บเซิร์ฟเวอร์สามารถกำหนดให้ script สามารถทำการ executed ที่ใดก็ได้ ทำให้ผลการทดลองในการตรวจสอบไม่สามารถตรวจสอบ successful attack ได้เลย แต่จะสามารถตรวจสอบได้เป็น unsuccessful attack แทน และจากตารางเวลาจะเห็นได้ว่าการวิเคราะห์การบุกรุกใช้เวลาเพียงหนึ่งวินาทีเท่านั้น ซึ่งสามารถคำนวณเวลาเฉลี่ยที่ใช้ในการพิจารณาแต่ละ alert (correlation per alert) คิดเป็น 1/80 = 0.0125 วินาที

4.2.2 ผลการทดลองสถานการณ์การบุกรุกโดยใช้ Nessus

Nessus เป็นเครื่องมือทดสอบการบุกรุกที่ใช้การทำงานแบบไคลเอนต์-เซิร์ฟเวอร์ โดยส่วนเซิร์ฟเวอร์จะต้องติดตั้งที่ UNIX เท่านั้น ส่วนไคลเอนต์สามารถติดตั้งได้ทั้งบนวินโดวส์และ UNIX ในการทดลองกำหนดให้มีการทดสอบการบุกรุกเฉพาะบนเว็บไปยังเครื่องเซิร์ฟเวอร์เป้าหมาย มีผลการทดลองดังต่อไปนี้

ตารางที่ 4.3 แสดงจำนวน alert ของการจำลองการบุกรุกโดยใช้ Nessus

scan rule	squid log	snort alert	successful attack alert	undecided attack alert	unsuccessful attack alert
1978	204	46	1	18	27

ตารางที่ 4.4 แสดงเวลาที่ใช้ในการจำลองการบุกรุกโดยใช้ Nessus

scan exploit	SQUID LOGS	SNORT LOGS	correlation analysis	correlation per alert
2502	2	2	6	0.1304

จากข้อมูลในตารางที่ 4.3 และ 4.4 จะเห็นได้ว่าโปรแกรม Nessus ใช้เวลาที่ใช้ในการตรวจสอบเครื่องเป้าหมายสูงมาก อันเนื่องมาจากมีรูปแบบการบุกรุกไปยังเว็บพอสมควรคือ 1978 rules โดย snort สามารถตรวจสอบการบุกรุกบนเว็บได้ 46 alert ซึ่งสามารถวิเคราะห์ unsuccessful attack ได้เพียง 27 alert คิดเป็น 27/46 = 59% ของ alert ทั้งหมด สาเหตุที่สามารถตรวจสอบ successful attack ได้น้อยมาก เนื่องมาจากรูปแบบการการตรวจสอบของ Nessus ใช้การตรวจสอบตามชื่อ directory มาตรฐานเท่านั้น ทำให้ไม่สามารถหาความสัมพันธ์ของล็อกไฟล์ได้เช่นในกรณีเดียวกับ ISS ส่งผลให้ส่วนใหญ่สามารถจำแนกได้เป็นประเภท unsuccessful attack และจากตารางเวลาจะเห็นได้ว่าการวิเคราะห์การบุกรุกใช้เวลาประมาณ 6 วินาที ซึ่งสามารถคำนวณเวลาเฉลี่ยที่ใช้ในการพิจารณาแต่ละ alert (correlation per alert) คิดเป็น 6/46 = 0.1304 วินาที

4.2.3 ผลการทดลองสถานการณ์การบุกรุกโดยใช้ Shadow

Shadow Security Scanner เป็นเครื่องมือทดสอบการบุกรุกที่เปิดให้ผู้ใช้งานสามารถดาวน์โหลดไปใช้งานได้ชั่วระยะเวลาหนึ่ง ในการทดลองกำหนดให้มีการทดสอบการบุกรุกเฉพาะบนเว็บไปยังเครื่องเซิร์ฟเวอร์เป้าหมาย มีผลการทดลองดังต่อไปนี้

ตารางที่ 4.5 แสดงจำนวน alert ของการจำลองการบุกรุกโดยใช้ shadow

scan rule	squid log	snort alert	successful attack alert	undecided attack alert	unsuccessful attack alert
2254	1741	273	0	260	13

ตารางที่ 4.6 แสดงเวลาที่ใช้ในการจำลองการบุกรุกโดยใช้ shadow

scan exploit	SQUID LOGS	SNORT LOGS	correlation analysis	correlation per alert
47	3	2	54	0.1978

จากข้อมูลในตารางที่ 4.5 และ 4.6 จะเห็นได้ว่าโปรแกรม shadow ใช้เวลาที่ใช้ในการตรวจสอบเครื่องเป้าหมายที่เร็วมากเมื่อเทียบกับจำนวนรูปแบบการบุกรุก โดยมีรูปแบบการบุกรุกไปยังเว็บพอสมควรคือ 2254 rules โดย snort สามารถตรวจสอบการบุกรุกบนเว็บได้ 273 alert ซึ่งสามารถวิเคราะห์ unsuccessful attack ได้ 13 alert คิดเป็น 13/270 = 5% ของ alert ทั้งหมด สาเหตุที่สามารถตรวจสอบ unsuccessful attack ได้จำนวนน้อย เนื่องมาจากการตรวจสอบการบุกรุกใช้เทคนิคชั้นสูงในการตรวจสอบซึ่งให้ค่า return code เป็น 5xx เสียส่วนมาก ทำให้การวิเคราะห์ส่วนมากเป็น undecided attack จากตารางเวลาจะเห็นได้ว่าการวิเคราะห์การบุกรุกใช้เวลาประมาณ 54 วินาที ซึ่งสามารถคำนวณเวลาเฉลี่ยที่ใช้ในการพิจารณาแต่ละ alert (correlation per alert) คิดเป็น 54/273 = 0.1978 วินาที

4.2.4 ผลการทดลองสถานการณ์การบุกรุกโดยใช้ N-Stealth

N-Stealth Security Scanner เป็นเครื่องมือทดสอบการบุกรุกที่สามารถกำหนดให้ใช้เทคนิคการหลบเลี่ยงการตรวจสอบการบุกรุกได้ ในการทดลองกำหนดให้มีการทดสอบการบุกรุกเฉพาะบนเว็บไปยังเครื่องเซิร์ฟเวอร์เป้าหมาย มีผลการทดลองดังต่อไปนี้

ตารางที่ 4.7 แสดงจำนวน alert ของการจำลองการบุกรุกโดยใช้ N-Stealth

Scan rule	squid log	snort alert	successful attack alert	undecided attack alert	unsuccessful attack alert
16025	16036	3813	185	702	2926

ตารางที่ 4.8 แสดงเวลาที่ใช้ในการจำลองการบุกรุกโดยใช้ N-Stealth

scan exploit	SQUID LOGS	SNORT LOGS	correlation analysis	correlation per alert
194	6	3	665	0.1744

จากข้อมูลในตารางที่ 4.7 และ 4.8 จะเห็นได้ว่าโปรแกรม N-Stealth มีใช้เวลาที่ใช้ในการตรวจสอบเครื่องเป้าหมายที่เร็วมาก กล่าวคือมีรูปแบบการบุกรุกไปยังเป้าหมายที่เยอะมาก คือ 16025 rules โดย snort สามารถตรวจสอบการบุกรุกได้ 3813 alert ซึ่งสามารถวิเคราะห์ unsuccessful attack ได้ 2926 alert คิดเป็น 2926/3813 = 76% ของ alert ทั้งหมด สาเหตุที่สามารถตรวจสอบ unsuccessful attack ได้จำนวนมาก เกิดมาจากรูปแบบการบุกรุกของ N-Stealth ใช้การตรวจสอบการบุกรุกโดยใช้รูปแบบการโจมตีแบบง่าย ๆ (ใช้เวลาน้อย) เป็นจำนวนมาก ทำให้สามารถวิเคราะห์ unsuccessful attack ได้เป็นจำนวนมากตามไปด้วย จากตารางเวลาจะเห็นได้ว่าการวิเคราะห์การบุกรุกใช้เวลาประมาณ 665 วินาที ซึ่งสามารถคำนวณเวลาเฉลี่ยที่ใช้ในการพิจารณาแต่ละ alert (correlation per alert) คิดเป็น 665/3813 = 0.1744 วินาที

4.2.5 สรุปผลการทดลองสถานการณ์การบุกรุกโดยใช้เครื่องมือ

การทดลองจำลองสถานการณ์การบุกรุกโดยใช้เครื่องมือตรวจสอบข้อบกพร่องชนิดต่าง ๆ มีผลสรุปการทดลองแสดงจำนวน alert และเวลาที่ใช้ในแต่ละขั้นตอน ดังตารางที่ 4.9 และตารางที่ 4.10 ตามลำดับ

ตารางที่ 4.9 แสดงจำนวน alert กรณีจำลองสถานการณ์โดยใช้เครื่องมือเพียงลำพัง

tools	snort alert	successful attack alert	undecided attack alert	unsuccessful attack alert	% successful attack	% undecided attack	% unsuccessful attack
ISS	80	0	29	51	0	36.25	63.75
Nessus	46	1	18	27	2.17	39.13	58.70
Shadow	273	0	260	13	0	95.23	4.77
N-Stealth	3813	185	702	2926	4.85	18.41	76.74

รายละเอียดฟิลด์ข้อมูลต่าง ๆ ในตารางที่ 4.9 โดยสรุป มีดังต่อไปนี้

- snort alert คือจำนวน alert ทั้งหมดที่บันทึกในไฟล์ snort alert
- successful attack alert คือจำนวน alert ที่วิเคราะห์ว่าเป็นการบุกรุกที่แท้จริง อันเนื่องมาจากมีค่าตอบกลับจากเว็บเซิร์ฟเวอร์เป็น 2xx Successful ซึ่งแสดงว่าสามารถปฏิบัติตามการร้องขอนั้นได้ (การบุกรุกเกิดขึ้นจริง)
- undecided attack alert คือจำนวน alert ที่ไม่สามารถวิเคราะห์ได้ว่าเป็นการบุกรุกที่แท้จริงหรือไม่ เนื่องจากมีข้อมูลไม่ชัดเจนเพียงพอ อันเนื่องมาจากมีค่าตอบกลับจากเว็บเซิร์ฟเวอร์เป็น 1xx Information และ 5xx Internal Server Error
- unsuccessful attack alert คือจำนวน alert ที่วิเคราะห์ว่าไม่เป็นการบุกรุกที่แท้จริง อันเนื่องมาจากมีค่าตอบกลับจากเว็บเซิร์ฟเวอร์เป็น 3xx Redirection และ 4xx Client Error ซึ่งแสดงว่าเว็บเซิร์ฟเวอร์ไม่ได้ปฏิบัติตามการร้องขอนั้นจริง เช่น 304 Not Modified ข้อมูลเหมือนเดิมไม่จำเป็นต้องปฏิบัติซ้ำ และ 404 File Not Found ไม่มีไฟล์ตามการร้องขอนั้น
- % successful attack คืออัตราส่วนระหว่าง successful attack alert กับจำนวน snort alert
- % undecided attack คืออัตราส่วนระหว่าง undecided attack alert กับจำนวน snort alert

- % unsuccessful attack คืออัตราส่วนระหว่าง unsuccessful attack alert กับจำนวน snort alert

ตารางที่ 4.10 แสดงเวลาที่ใช้ในกรณีจำลองสถานการณ์โดยใช้เครื่องมือเพียงลำพัง (วินาที)

tools	scan exploit	SQUID LOGS	SNORT LOGS	correlation analysis	correlation per alert
ISS	1020	1	1	1	0.0125
Nessus	2502	2	2	6	0.1304
Shadow	47	3	2	54	0.1978
N-Stealth	194	6	3	665	0.1744

รายละเอียดฟิลด์ข้อมูลต่าง ๆ ในตารางที่ 4.10 มีดังต่อไปนี้

- scan exploit คือเวลาที่ใช้การตรวจสอบโดยใช้เครื่องมือชนิดต่าง ๆ
- SQUID LOGS คือเวลาที่ใช้ในการแปลง squid access log ลงตาราง SQUID_LOGS
- SNORT LOGS คือเวลาที่ใช้ในการแปลง snort alert log ลงตาราง SNORT_LOGS
- correlation analysis คือเวลาที่ใช้ในการหาความสัมพันธ์ระหว่าง snort และ squid รวมถึงการวิเคราะห์การบุกรุกจำแนกตามประเภทต่าง ๆ
- correlation per alert คือเวลาเฉลี่ยที่ใช้ในการพิจารณาแต่ละการแจ้งเตือน (correlation per alert =correlation analysis /snort alert)

ในการทดลองวิเคราะห์ข้อมูลจากการจำลองสถานการณ์การบุกรุกโดยใช้เครื่องมือพิจารณาจากตารางที่ 4.9 ทำให้เห็นว่า หากต้องการให้มีผลการจำแนก unsuccessful attack ให้ได้ในอัตราที่พอใจ เครื่องมือที่ใช้ในการตรวจสอบข้อบกพร่องก็จะต้องมีการตรวจสอบข้อบกพร่องโดยละเอียด กล่าวคือ ในการตรวจสอบข้อบกพร่องหนึ่งจุดจะใช้รูปแบบการบุกรุกหลายรูปแบบทำการตรวจสอบ ทำให้มีบางรูปแบบซึ่งสามารถวิเคราะห์โดยใช้ความสัมพันธ์ระหว่างล็อกไฟล์ได้ ซึ่งถ้าเป็นการตรวจสอบโดยใช้เทคนิคการบุกรุกขั้นสูงจะทำให้ระบบไม่สามารถหาความสัมพันธ์ระหว่างล็อกได้ ทำให้ผลการจำแนกจึงเป็น undecided attack เสียส่วนมาก นอกจากนี้ในตารางที่ 4.10 ยังใช้เวลาเฉลี่ยที่ใช้ในการวิเคราะห์ในอัตราที่ต่ำมาก (น้อยกว่า 1 วินาที) ส่งผลให้มีแนวโน้มที่จะนำการจำแนกการแจ้งเตือนการบุกรุกโดยใช้ล็อกไฟล์ร่วมกันนี้ไปทดลองใช้ในสถานการณ์จริงต่อไป

4.2.6 ผลการทดลองสถานการณ์การบุกรุกจริง

การจำลองสถานการณ์การบุกรุกจริง โดยกำหนดให้มีการเรียกใช้งานเว็บไซต์ตามปกติและมีการบุกรุกโดยใช้เครื่องมือชนิดต่าง ๆ ตลอดจนการบุกรุกโดยใช้เพียงเว็บเบราเซอร์ธรรมดาสลับกันไป โดยช่วงเวลาที่ใช้ในการทดลองจะพิจารณาจากขนาดล็อกไฟล์ของ squid เป็นหลัก โดยในสถานการณ์นี้ กำหนดให้ล็อกเว็บพร็อกซีมีขนาดประมาณ 1,5,10,25,50,75,100 ล้านไบต์ตามลำดับ ซึ่งสามารถสรุปปริมาณข้อมูลได้ดังนี้

- สถานการณ์ Mix1M SQUID มีขนาดล็อกไฟล์ประมาณ 1,000,000 ไบต์และมีจำนวนการ access ประมาณ 8K records
- สถานการณ์ Mix5M SQUID มีขนาดล็อกไฟล์ประมาณ 5,000,000 ไบต์และมีจำนวนการ access ประมาณ 39K records
- สถานการณ์ Mix10M SQUID มีขนาดล็อกไฟล์ประมาณ 10,000,000 ไบต์และมีจำนวนการ access ประมาณ 74K records
- สถานการณ์ Mix25M SQUID มีขนาดล็อกไฟล์ประมาณ 25,000,000 ไบต์และมีจำนวนการ access ประมาณ 195K records
- สถานการณ์ Mix50M SQUID มีขนาดล็อกไฟล์ประมาณ 50,000,000 ไบต์และมีจำนวนการ access ประมาณ 364K records
- สถานการณ์ Mix75M SQUID มีขนาดล็อกไฟล์ประมาณ 75,000,000 ไบต์และมีจำนวนการ access ประมาณ 548K records
- สถานการณ์ Mix100M SQUID มีขนาดล็อกไฟล์ประมาณ 100,000,000 ไบต์และมีจำนวนการ access ประมาณ 730K records

ซึ่งให้ผลการทดลองแสดงดังตาราง 4.11 และ 4.12 ดังต่อไปนี้

ตารางที่ 4.11 แสดงจำนวน alert ของการจำลองการบุกรุกจริง

squid size (MB)	squid logs	snort alerts	successful attack alert	undecided attack alert	unsuccessful attack alert
Mix1M	8295	288	1	63	224
Mix5M	39943	4107	55	955	3097
Mix10M	74332	6262	91	1606	4565
Mix25M	195868	23888	135	13909	9844
Mix50M	364218	48541	N/A	N/A	N/A
Mix75M	548526	77167	N/A	N/A	N/A
Mix100M	730133	107119	N/A	N/A	N/A

ตารางที่ 4.12 แสดงเวลาที่ใช้ในการจำลองการบุกรุกจริง

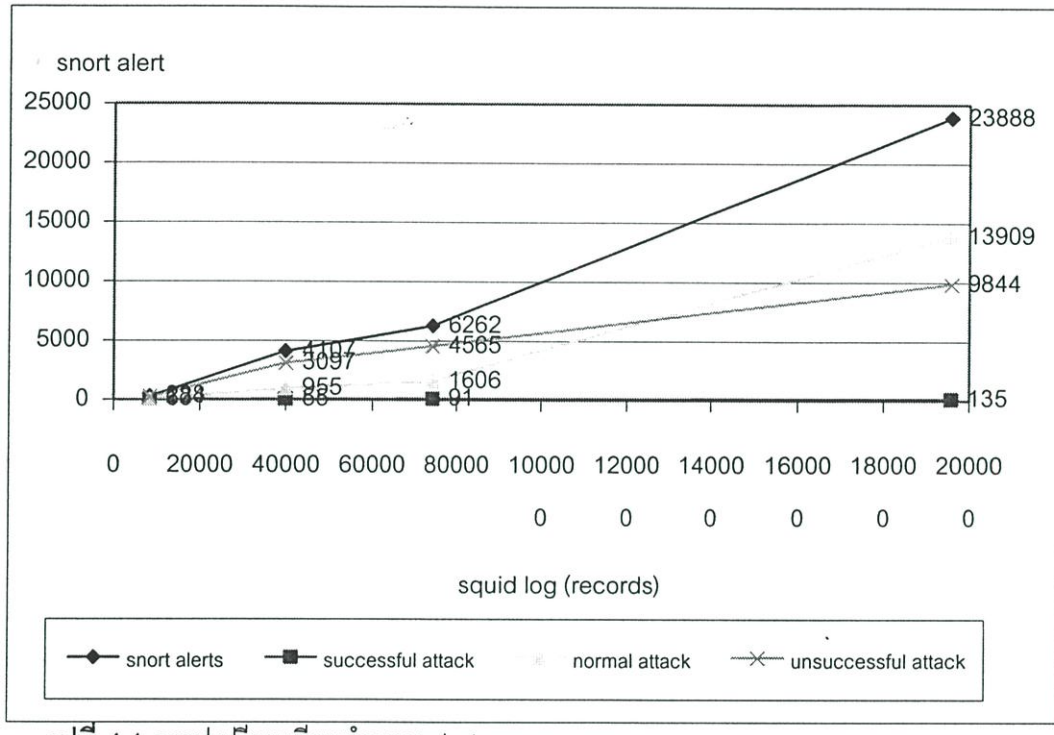
squid size (MB)	SNORT RULES	SQUID LOGS	SNORT LOGS	correlation analysis	correlation per alert
Mix1M	<1	4	1	28	0.097
Mix5M	<1	20	2	1780	0.433
Mix10M	<1	36	3	5192	0.829
Mix25M	<1	127	38	49815	2.085
Mix50M	<1	178	26	N/A	(~4.679)
Mix75M	<1	343	135	N/A	(~7.944)
Mix100M	<1	415	165	N/A	(~14-95)

จากข้อมูลในตารางที่ 4.11 และ 4.12 จะเห็นได้ว่าในสถานการณ์จริงนี้ snort จะสามารถตรวจสอบการบุกรุกได้มากขึ้นตามจำนวนขนาดล็อก squid ค่าเวลาเฉลี่ยที่ใช้ในการวิเคราะห์แต่ละ alert จะยังต่ำกว่า 1 วินาที เมื่อมีขนาดล็อก squid ไม่เกิน 10 ล้านไบต์ หรือมีจำนวนการ access ประมาณเกือบแสนครั้ง นั่นหมายถึงการวิเคราะห์การบุกรุกโดยใช้ความสัมพันธ์ระหว่างล็อกนี้สามารถทำงานได้อย่างมีประสิทธิภาพสูงสุด เมื่อมีขนาดล็อกเว็บประมาณแสนครั้ง เพราะเมื่อพิจารณาถึงการทดลองกับขนาดล็อกเว็บขนาดประมาณ 25 ล้านไบต์มีการ access ประมาณสองแสนครั้ง สามารถวิเคราะห์ alert ได้ในอัตราเฉลี่ย 2 วินาทีเศษ ซึ่งเป็นการคัดแยก unsuccessful attack alert จำนวน 9844 จากจำนวน alert ทั้งหมด 23888 alert คิดเป็น 9844/23888 = 41% โดยใช้ระยะเวลาทั้งสิ้น 49815 วินาที หรือ 13.8375 ชั่วโมง และในการ

ทดลองกับล็อก squid ขนาดข้อมูล 50,75,100 ล้านไบต์นั้น ก็จะยิ่งใช้เวลานานมากยิ่งขึ้น ซึ่งมีค่าเวลาเฉลี่ยที่ใช้ในการพิจารณาแต่ละ alert เป็น 4.679, 7.944 และ 29.98 ตามลำดับ จากแนวโน้มอัตราการพิจารณา alert โดยเฉลี่ยที่เพิ่มสูงมากขึ้น ทำให้ในการวิเคราะห์การบุกรุกควรจำกัดหรือแบ่งล็อกไฟล์ใหญ่ ๆ ให้มีขนาดประมาณ 10 ล้านไบต์ เพื่อประสิทธิภาพในการวิเคราะห์สูงสุด

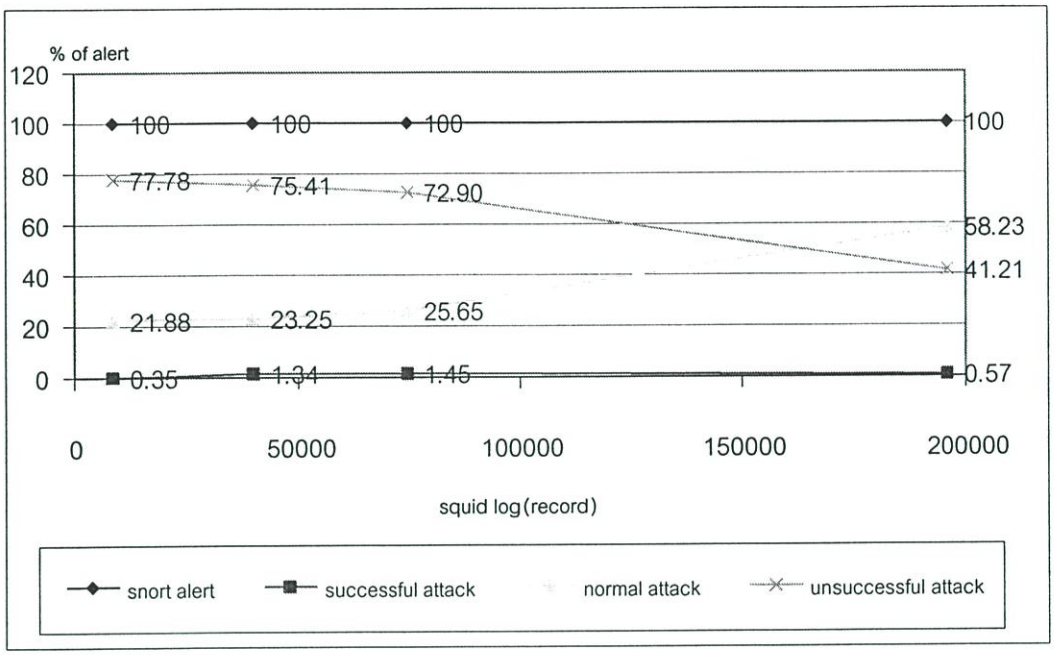
4.3 การวิเคราะห์ผลการทดลอง

จากตารางที่ 4.9 และ 4.11 สามารถนำข้อมูลจำนวน alert มาสร้างกราฟได้ดังรูปที่ 4.1



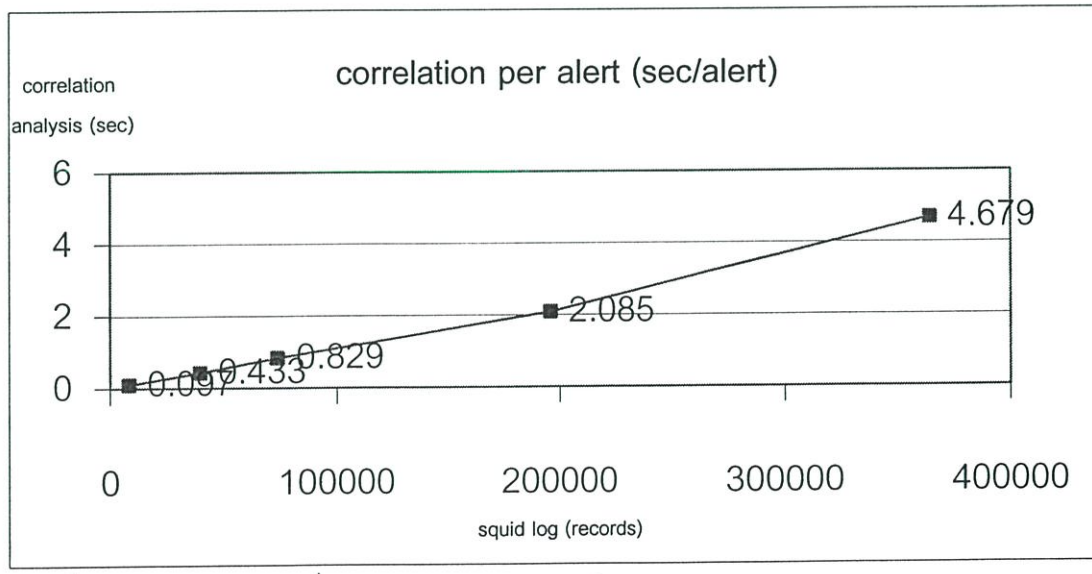
รูปที่ 4.1 กราฟเปรียบเทียบจำนวน alert

จากรูปที่ 4.1 เป็นกราฟแสดงการเปรียบเทียบจำนวน alert ในแต่ละประเภทการแจ้งเตือนการบุกรุก ซึ่งแสดงให้เห็นว่า เมื่อมีการนำ snort alert นำไปหาความสัมพันธ์กับ squid access แล้วทำให้สามารถจำแนก snort alert ที่เป็น unsuccessful attack ลงได้เป็นจำนวนมาก ทำให้ลดเวลาในการวิเคราะห์เพื่อหาการบุกรุกที่แท้จริงต่อไป แม้ว่าในตารางที่ 4.10 และ 4.12 จะเห็นว่าได้เวลาที่ใช้ในการวิเคราะห์หาความสัมพันธ์ (correlation analysis) มีแนวโน้มเพิ่มขึ้นตามปริมาณข้อมูล snort และ squid แต่เมื่อพิจารณาถึงอัตราการวิเคราะห์ต่อหน่วย (correlation per alert) แล้วนับว่ามีอัตราที่น่าพอใจเป็นอย่างยิ่งเมื่อเทียบกับปริมาณ unsuccessful attack ที่สามารถแยกออกไปได้ ดังแสดงรายละเอียดการจำแนกการแจ้งเตือนการบุกรุกแต่ละประเภทดังรูปที่ 4.2



รูปที่ 4.2 กราฟเปรียบเทียบจำนวนเปอร์เซนต์ของ alert แต่ละประเภท

สำหรับในกรณีที่ล็อก IDS และเว็บพร็อกซีมีขนาดใหญ่มาก (มากกว่า 10 ล้านไบต์) ทำให้เวลาที่ใช้ในการหาความสัมพันธ์ (correlation analysis) มีแนวโน้มเพิ่มสูงขึ้นนั้น สามารถแก้ไขได้โดยการแบ่งล็อกไฟล์ออกเป็นไฟล์ย่อย ๆ ได้ แล้วทำการหาความสัมพันธ์ได้ตามปกติ โดยไม่ส่งผลต่อการวิเคราะห์การบุกรุกแต่อย่างใด ซึ่งสามารถแสดงได้ดังรูปที่ 4.3



รูปที่ 4.3 กราฟแสดงเวลาที่ใช้ในการหาความสัมพันธ์

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

5.1 รายงานสรุป

ในงานวิจัยนี้เสนอวิธีการใหม่ในการจำแนกประเภทการแจ้งเตือนการบุกรุกของระบบ IDS ในปัจจุบัน โดยเฉพาะอย่างยิ่งการแจ้งเตือนในเหตุการณ์การบุกรุกที่ไม่ประสบความสำเร็จ (unsuccessful attack) โดยการนำข้อมูลที่มีอยู่แล้วตามปกติของเว็บพร็อกซีมาช่วยในการวิเคราะห์ โดยการหาความสัมพันธ์ระหว่างล็อก IDS และล็อกเว็บพร็อกซี ทำให้ได้รับข้อมูลบางส่วนที่เป็นประโยชน์จากเว็บพร็อกซีมาร่วมวิเคราะห์ ได้แก่ ข้อมูลการตอบกลับจากเว็บเซิร์ฟเวอร์ที่แท้จริง (return code) เพื่อจำแนก unsuccessful attack ที่มีสาเหตุมาจากการบุกรุกที่ไม่เกิดขึ้นจริงบนเว็บเซิร์ฟเวอร์ และข้อมูลเบอร์ไอพีต้นทาง ทำให้ทราบเบอร์ไอพีที่แท้จริงของผู้บุกรุกก่อนผ่านเข้าสู่ระบบเครือข่าย

เมื่อได้รับข้อมูลที่มากขึ้นแล้ว สามารถกำหนดระดับความรุนแรงของการบุกรุกเพิ่มขึ้นได้โดยพิจารณาจากค่า priority ของ IDS ประกอบกับค่า return code ของเว็บพร็อกซี ทำให้การวิเคราะห์ผลลัพธ์เกิดความชัดเจนและละเอียดมากยิ่งขึ้น

เนื่องจากล็อกที่นำมาวิเคราะห์ร่วมเป็นล็อกของการเข้าถึงข้อมูลในระดับแอพพลิเคชันเลเยอร์ ซึ่งครอบคลุมเฉพาะโปรโตคอล HTTP ทำให้ผลการวิเคราะห์การบุกรุกในงานวิจัยนี้จะครอบคลุมเฉพาะการบุกรุกบนเว็บเท่านั้น

5.2 ข้อเสนอแนะเพื่องานวิจัยในอนาคต

จากผลการทดลองในงานวิจัยแสดงให้เห็นว่าการวิเคราะห์การบุกรุกสามารถนำข้อมูลอื่นที่มีอยู่แล้วในระบบมาประยุกต์ใช้งานเพื่อให้เกิดประโยชน์สูงสุด โดยในงานวิจัยนี้ได้นำ alert จากล็อกไฟล์ของ snort มาหาความสัมพันธ์กับข้อมูลการ access ของล็อกไฟล์ของ squid ซึ่งทำให้ระบบสามารถทราบถึงรายละเอียดเพิ่มเติมของเหตุการณ์ที่สงสัยว่าจะเป็นการบุกรุกนั้น ๆ ทำให้สามารถพิจารณาจำแนกชนิดของ alert นั้นได้ 3 ชนิด คือ successful attack, undecided attack และ unsuccessful attack ในการวิเคราะห์การบุกรุกจึงสามารถแยกการแจ้งเตือนการบุกรุกที่ประสบความสำเร็จ (successful attack) และ การแจ้งเตือนการบุกรุกที่ไม่ประสบความสำเร็จ (unsuccessful attack) ได้อย่างชัดเจน ทำให้เกิดประสิทธิภาพในการจัดลำดับการป้องกันและแก้ไขปัญหาการบุกรุกได้อย่างเหมาะสม เนื่องจากในงานวิจัยนี้เน้นการจำแนกประเภทการแจ้งเตือน

การบุกรุกเป็นหลัก ซึ่งแท้ที่จริงแล้ว ในระบบ IDS ยังมี false alert อีก คือ false positives alert และ false negatives alert โดยที่ false positives alert คือ alert ซึ่งเกิดจากการเข้าใจผิดของ IDS ที่ตรวจสอบการทำงานปกติเป็นการบุกรุก ส่วน false negatives alert มักเป็น alert ที่เกิดขึ้นกับการบุกรุกแบบใหม่ ในการลด alert ชนิดนี้จำเป็นที่จะต้องใช้อัลกอริทึมที่ซับซ้อนมากยิ่งขึ้นในการวิเคราะห์ เพราะฉะนั้นในการพัฒนาการวิเคราะห์การบุกรุกต่อไปควรมีการนำค่า false alert มาพิจารณาประกอบการวิเคราะห์ด้วย

ในโปรแกรมการวิเคราะห์การบุกรุกควรที่จะสามารถปรับปรุงแก้ไขให้สามารถประยุกต์ใช้งานได้กับล็อกของโปรโตคอลในระดับแอพพลิเคชันอื่น ๆ เช่น SMTP, POP3, FTP ได้โดยง่าย และเพื่อทำให้โปรแกรมที่ใช้ในการวิเคราะห์ข้อมูลมีความยืดหยุ่นในการใช้งานมากยิ่งขึ้น ควรกำหนดให้โปรแกรมสามารถนำข้อมูลล็อกไฟล์จากอุปกรณ์ที่อยู่ต่างเครื่องกันได้โดยอัตโนมัติ ซึ่งจะส่งผลทำให้โปรแกรมสามารถทำงานในแบบทันทีทันใด (real time) ได้ในระดับหนึ่ง

เอกสารอ้างอิง

[1] C. Sample, M. Nikle and I. Poynter. "Firewall And IDS Shortcomings". SANS Network Security, Monterey, California, October 2000.

[2] T. Ptacek and T. Newsham. "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection". Technical report, Secure Networks, Inc. January, 1998.

[3] A. Valdes and K. Skinner. "An Approach to Sensor Correlation". In DARPA ISO Principal Investigator Meeting, July 2000.

[4] A. Valdes and K. Skinner. "Probabilistic Alert Correlation". In Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection (RAID2001), pages54–68, 2001.

[5] D. Anderson, M. Fong and A. Valdes. "Heterogeneous Sensor Correlation: A Case Study of Live Traffic Analysis". In DARPA ISO Principal Investigator Meeting, SRI International 2000.

[6] H. Debar and A. Wespi. "Aggregation and Correlation of Intrusion-Detection Alerts". In Recent Advances in Intrusion Detection, number 2212 in Lecture Notes in Computer Science, pages 85 – 103, 2001.

[7] M. Almgren and U. Lindqvist. "Application-Integrated Data Collection for Security Monitoring". In Recent Advances in Intrusion Detection (RAID2001), pages 22-36, 2001.

[8] N. Desai. "Incresing Performance in High Speed NIDS" Snort Project. [Online]. Available: http://www.snort.org.

[9] P. Ning, Y. Cui, and D. S Reeves. "Constructing attack scenarios through correlation of intrusion alerts." In Proceedings of the 9th ACM Conference on Computer and Communications Security, Washington, D.C., pages 245-254, November 2002.

[10] T. Daniels and E. Spafford. "A Network Audit System for Host-based Intrusion Detection (NASHID) in Linux". CERIAS Technical Report 99/10 Purdue University, 2000.

[11] S. McClure and S. Shah "WEB HACKING: ATTACKS AND DEFENSE". Addison Wesley Professional 2003.

[12] Martin Roesch. 2002. Snort. [Online]. Available: http://www.snort.org/

[13] Robert Sedgewick, "Algorithms in C: Fundamentals, Data Structures, Sorting, Searching", Addison-Wesely Publishing Company, 1997

[14] Squid. [Online]. Available: http://www.squid-cache.org/

[15] Analysis Console for Intrusion Databases (ACID). [Online]. Available: http://www.andrew.cmu.edu/~rdanyliw/snort/snortacid.html

[16] SnortSnarf, Silicon Defense. [Online]. Available: http://www.silicondefense.com/software/snortsnarf/

[17] Event Monitoring Enabling Responses to Anomalous Live Disturbances (EMERALD) [Online] Available: http://www.sdl.sri.com/projects/emerald/

[18] Nessus. [Online]. Available: http://www.nessus.org/

[19] Internet Scanner. [Online]. Available: http://www.iss.net/

[20] Shadow Security Scanner. [Online]. Available: http://www.safety-lab.com/

[21] N-Stealth Security Scanner. [Online]. Available: http://www.nstalker.com/nstealth/

ภาคผนวก ก.
ตัวอย่างโปรแกรม

1. snort_rules2mysql.pl

```
#!/usr/bin/perl -w
#
die("Usage: snort_rules2mysql.pl <snort rule files>\n") unless(@ARGV);

use DBI;
use DBD::mysql;
$driver="mysql";
$host="localhost";
$user="root";
$pass="";
$database="IDSFW";
$dsn = "DBI:$driver:database=$database;host=$host";
$dbh = DBI->connect($dsn,$user,$pass);
#$logfile="/src/snort-2.1.1/rules/*";       # command line
$dbh->do("DELETE FROM `SNORT_RULES`");
foreach $file (@ARGV) {
        open(RULES, $file) or die( "Cannot open file: $file\n" );

LOOP:
        while(<RULES>) {
                next if(/^\s$/);
                next if(/^\#/);
                if (/\((.*)\)/) {
                        $action = $1;
```

```
                $uricontent = "";
                $content = "";
                $msg = "";
                foreach $rule (split(/;\s+/, $action)) {
                        if ($rule =~ /uricontent:\s*\"(.*)\"/) {
                                $uricontent = $1;
                        } elsif ($rule =~ /content:\s*\"(.*)\"/) {
                                $content = $1;
                        } elsif ($rule =~ /msg:\s*\"(.*)\"/) {
                                $msg = $1;
                        }
                }
                # decode URL decoding
                $uricontent =~ s/%([a-fA-F0-9][a-fA-F0-9])/\\x$1/sg;
                $content =~ s/%([a-fA-F0-9][a-fA-F0-9])/\\x$1/sg;
                if ($content =~ /\|/) {
                        $content="NOP"
                        # next LOOP;
                }
                if ($uricontent =~ /\|/) {
                        $uricontent="NOP"
                        # next LOOP;
                }
                $msg=$dbh->quote($msg);
                $uricontent=$dbh->quote($uricontent);
                $content=$dbh->quote($content);
                $dbh->do("INSERT INTO SNORT_RULES set msg=$msg,
                uricontent=$uricontent, content=$content");
        }
}
close(RULES);
```

```
}
```

2. SQUID.pl

```
#!/usr/bin/perl -w

use DBI;
use DBD::mysql;
$driver="mysql";
$host="161.246.49.144";
$user="root";
$pass="";
$database="IDSFW";
$dsn = "DBI:$driver:database=$database;host=$host";
$dbh = DBI->connect($dsn,$user,$pass);
$logfile="/var/log/squid/access.log";

open (LOG,$logfile) or die "couldn't open file";
$dbh->do("DELETE FROM `SQUID_LOGS`");
while($logline=<LOG>){
        chomp($logline);
        if ($logline=~/^\s+$/) {
                next;
        }

        ($time,$elapsed,$remotehost,$stat,$bytes,$method,$url,$dash,$result,$objtype)
=" ";
         my @record = split(/\s+/,$logline);

        ($time,$elapsed,$remotehost,$stat,$bytes,$method,$url,$dash,$result,$objtype)
        =@record;
```

```
        $qtime=$dbh->quote($time);
        $qremote=$dbh->quote($remotehost);
        $qstat=$dbh->quote(substr($stat,index($stat,"/")+1));
        $qurl=$dbh->quote($url);
        $qresult=$dbh->quote(substr($result,index($result,"/")+1));

        $dbh->do("INSERT INTO SQUID_LOGS set time_stamp=$qtime,
        source_ip=$qremote, target_ip=$qresult,  url=$qurl,  return_code=$qstat");
 }
$dbh->disconnect();
```

3. SNORT.pl

```
#!/usr/bin/perl -w

use DBI;
use DBD::mysql;
use Time::Local;
$driver="mysql";
$host="localhost";
$user="root";
$pass="";
$database="IDSFW";
$dsn = "DBI:$driver:database=$database;host=$host";
$dbh = DBI->connect($dsn,$user,$pass);
$logfile="/var/log/snort/alert";

open (LOG,$logfile) or die "couldn't open file";
$dbh->do("DELETE FROM `SNORT_LOGS`");
```

```
$total=0;
$filter=0;
while($logline=<LOG>){
        chomp($logline);
        if ($logline=~/^\s+$/) {
                next;
        }
        if ($logline=~/^\[\*\*\]/) {
        $total++;
        $temp=substr($logline,index($logline," ")+1);
        $temp=substr($temp,index($temp," ")+1);
        $msg=substr($temp,0,index($temp,"*")-2);
        }elsif ($logline=~/Priority/){
                $priority=substr($logline,length($logline)-3,1);
        }elsif ($logline=~/^[0-9]/) {
                if (index($logline,"/")  < 1) {
                        next;
                }
                if ($msg !~/^WEB/) {
                        next;
                }
                $year=(localtime)[5];
                $month=substr($logline,0,2)-1;
                $day=substr($logline,3,2);
                $hours=substr($logline,6,2);
                $minutes=substr($logline,9,2);
                $seconds=substr($logline,12,2);
                $milliseconds=substr($logline,15,3);
                $time_stamp = timelocal($seconds, $minutes, $hours, $day,
$month,$year).".".$milliseconds;
```

```
                $temp=substr($logline,index($logline," ")+1);
                $source_ip=substr($temp,0,index($temp,":"));

                $temp=substr($logline,index($logline,"\>")+2);
                $target_ip=substr($temp,0,index($temp,":"));

                $time_stamp=$dbh->quote($time_stamp);
                $source_ip=$dbh->quote($source_ip);
                $target_ip=$dbh->quote($target_ip);
                $msg=$dbh->quote($msg);
                $priority=$dbh->quote($priority);

                if (length($priority)==3) {
                        $filter++;
                        $dbh->do("INSERT INTO `SNORT_LOGS` set
time_stamp=$time_stamp, source_ip=$source_ip, target_ip=$target_ip, msg=$msg,
priority=$priority");
                }
        }
}
$dbh->disconnect();

print("\n\nTotal=[$total] rule(s)\tWeb Filter=[$filter] rule(s)\n");
```

4. correlation.pl

```
#!/usr/bin/perl -w

use DBI;
use DBD::mysql;
$driver="mysql";
```

```
$host="localhost";
$user="root";
$pass="";
$database="IDSFW";
$dsn = "DBI:$driver:database=$database;host=$host";
$dbh = DBI->connect($dsn,$user,$pass);

my $sql = qq{select time_stamp,target_ip,uricontent,content,priority from
SNORT_LOGS,SNORT_RULES where (SNORT_LOGS.msg LIKE SNORT_RULES.msg)
ORDER BY SNORT_LOGS.time_stamp};
my $sth = $dbh->prepare( $sql );
$sth->execute();
my( $time_stamp,$target_ip,$uricontent ,$content,$priority);
$sth->bind_columns( undef, \$time_stamp,\$target_ip,\$uricontent ,\$content,\$priority);

while( $sth->fetch() ) {
                if ($uricontent eq "") {
                        $uricontent=$content;
                }
                $uricontent=~s/\\\./M\./;
                $uricontent=~s/M\.\./;

                        if ($content eq "NOP") {
                                next;
                        }
        print"\n----------START----------\n";
        print "==SNORT
TIME=[$time_stamp],IP=[$target_ip],URL=[$uricontent],PRIORITY=[$priority]";

                my $sql1 = qq{select  time_stamp,source_ip,url,return_code from
SQUID_LOGS where (time_stamp BETWEEN $time_stamp-2 AND $time_stamp) AND
```

```
(url LIKE "%$uricontent%") AND (target_ip LIKE "%$target_ip%") ORDER BY
time_stamp};
                my $sth1 = $dbh->prepare( $sql1);

                $sth1->execute();

        print"\t\t\Found===[".$sth1->rows."]\n";
        if ($sth1->rows==0) {
                print"----------SKIP----------\n";
                next;
        }
                my( $time_stamp, $source_ip, $url ,$return_code);
                $sth1->bind_columns( undef, \$time_stamp, \$source_ip, \$url,
\$return_code);
                $sth1->fetch();
                 print "==SQUID
TIME=[$time_stamp],IP=[$source_ip],URL=[$url],RESULT=[$return_code]\n";
                $sth1->finish();
}
$sth->finish();
$dbh->disconnect();
```

5. correlation logs

squid log

```
n1079012311.380     4 161.246.49.142 TCP_MISS/404 509 GET http://161.246.49.144/cgi-bin/formmail.pl -
DIRECT/161.246.49.144 text/html
```

snort alert

```
n[**] [1:884:8] WEB-CGI formmail access [**]
[Classification: access to a potentially vulnerable web application] [Priority: 2]
03/11-20:38:31.146601 161.246.49.152:38275 -> 161.246.49.144:80
TCP TTL:64 TOS:0x0 ID:2828 IpLen:20 DgmLen:355 DF
***AP*** Seq: 0xC44C35D3  Ack: 0x4FFC7179  Win: 0x16D0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 1523416 3211425
```

[Xref => http://www.whitehats.com/info/IDS226]

[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0172]

[Xref => http://www.securityfocus.com/bid/1187]

[Xref => http://cgi.nessus.org/plugins/dump.php3?id=10076]

[Xref => http://cgi.nessus.org/plugins/dump.php3?id=10782]

snort rule (web-cgi.rules)

nalert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-CGI formmail access";
flow:to_server,established; uricontent:"/formmail"; nocase; reference:nessus,10782; reference:nessus,10076;
reference:bugtraq,1187; reference:cve,CVE-1999-0172; reference:arachnids,226; classtype:web-application-activity;
sid:884; rev:8;)

ประวัติผู้เขียน

ชื่อ-นามสกุล	นายบุญลือ ด้วงสำรวย
วัน เดือน ปีเกิด	26 กันยายน 2519
สถานที่เกิด	ชลบุรี
ที่อยู่	116/1 หมู่ 9
	ต.สัตหีบ อ.สัตหีบ จ.ชลบุรี 20180
ประวัติการศึกษา	2542 วิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์
	สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ