รายงานฉบับสมบูรณ์

Final Report

เงินงบประมาณแผ่นดิน

การตรวจจับการบุกรุกด้วยระบบการเรียนรู้ของตัวจำแนกประเภท

Using a Learning Classifier System for Intrusion Detection

หัวหน้าโครงการวิจัย: รศ.ดร. บุญวัฒน์ อัตชู
ที่ปรึกษาโครงการวิจัย: รศ.ดร. เอื้อน ปิ่นเงิน
นักวิจัย: นายเกรียงศักดิ์ เตมีย์
นายไพฑูรย์ ศรีนิล
ผู้ช่วยวิจัย: นายพรเทพ โรจนวสุ
นายศรชัย อุดมธนาพงศ์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

กันยายน พ.ศ. 2551

King Mongkut's Institute of Technology Ladkrabang

September 2008

กิตติกรรมประกาศ

งานวิจัยนี้จะไม่สำเร็จลุล่วงหากปราศจากแรงผลักดัน และคำแนะนำที่มีประโยชน์ของหัวหน้าโครงการ ที่ปรึกษาโครงการวิจัย และผู้ร่วมวิจัยทุกท่าน

ผู้จัดทำขอขอบคุณหน่วยงานสถาบันคอมพิวเตอร์ของมหาวิทยาลัยรามคำแหงเป็นอย่างสูง ที่ได้ให้ความอนุเคราะห์ข้อมูลเพื่อใช้ในการทดลอง

ผู้จัดทำขอขอบคุณสำนักงานคณะกรรมการวิจัยแห่งชาติที่อนุมัติทุนสนับสนุนโครงการวิจัยนี้

สุดท้ายนี้คุณค่าและประโยชน์อันพึงมีจากรายงานฉบับนี้ ผู้จัดทำขอมอบให้กับผู้มีพระคุณทุกท่าน หากรายงานฉบับนี้มีข้อผิดพลาดประการใดผู้จัดทำขอน้อมรับไว้เพียงผู้เดียว

โครงการวิจัยนี้ได้รับทุนอุดหนุนจากสำนักงานคณะกรรมการวิจัยแห่งชาติ ประจำปีงบประมาณ พ.ศ. 2551

ผู้จัดทำ

บทคัดย่อ

ในปัจจุบันระบบเครือข่ายคอมพิวเตอร์มีการเติบโตอย่างรวดเร็ว ทำให้ระบบตรวจจับการบุกรุกกลายเป็นส่วนสำคัญส่วนหนึ่งของระบบเครือข่าย ในระบบตรวจจับการบุกรุกบางระบบจะอาศัยพฤติกรรมการใช้งานเครือข่ายแบบปกติเป็นตัวตรวจจับพฤติกรรมที่แปลกปลอม แต่ในบางระบบก็จะอาศัยการจดจำรูปแบบการบุกรุก ระบบตรวจจับการบุกรุกที่พัฒนาในปัจจุบันนี้มีการประยุกต์ใช้ปัญญาประดิษฐ์กันอย่างกว้างขวางเพื่อเพิ่มประสิทธิภาพในการตรวจจับรูปแบบการบุกรุกใหม่ๆ ได้

งานวิจัยฉบับนี้นำเสนอระบบตรวจจับการบุกรุกโดยใช้ระบบ LCS โดยเราพัฒนาส่วนของการแบ่งระดับค่าความผิดปกติของข้อมูลโดยใช้แผนภาพ Self-Organizing Map โดยการเข้ารหัส 2 บิต ขึ้นอยู่กับค่าความเบี่ยงเบนจากพฤติกรรมการใช้งานปกติ เราได้ทำการทดลองกับข้อมูล FTP-Only และข้อมูลที่เก็บมาจากมหาวิทยาลัยรามคำแหง จากผลการทดลองแสดงให้เห็นว่าระบบที่เราได้นำเสนอมีประสิทธิภาพดีกว่าระบบ LCS เดิมและระบบจักรกลเรียนรู้อื่นๆ อีก 12 ระบบ

Abstract

Nowadays, as interconnections among computer systems grow rapidly, Intrusion Detection Systems (IDSs) play an important role of network security. Some systems are anomaly based and others are signature based. However, no detection system can catch all types of intrusions. At the moment most of the researchers are interested in improving intrusion detection which includes artificial intelligence.

In this research, we propose intrusion detection system using Learning Classifier System –LCS. We extends work in the area by applying the Self-Organizing Map (SOM) for creating the new input string by 2-bit encoding rely on degree of deviation of normal behaviour. The performance of systems is investigated under an FTP-only dataset and data from Ramkhamhaeng University. It is shown that the proposed system is able to perform significantly better than the conventional XCS and twelve ML algorithms.

สารบัญ

หน้า

สารบัญตาราง

หน้าที่

สารบัญรูป

หน้าที่

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันมีการใช้งานระบบคอมพิวเตอร์ เพื่อเพิ่มประสิทธิภาพการทำงานของหน่วยงาน โดยที่ลักษณะการทำงานของระบบคอมพิวเตอร์จะทำงานในลักษณะเครือข่ายเปิดที่เชื่อมต่อกับครือข่าภายนอก เช่น อินเทอร์เน็ต เป็นต้น ปัญหาที่เกิดขึ้นตามมาก็คือปัญหาการบุกรุกเข้ามาสร้างความเสียหายให้กับระบบ การบุกรุกที่เกิดขึ้นจะเป็นปัญหากับผู้ดูแลระบบอย่างมากเนื่องจากผู้ดูแลระบบต้องคอยป้องกันและแก้ไขปัญหาต่างๆตลอดเวลา ยิ่งระบบที่มีขนาดใหญ่และมีความซับซ้อนสูง การทำงานของผู้ดูแลระบบจะมีความยุ่งยากมาก ทางออกหนึ่งที่จะช่วยแก้ปัญหาเหล่านี้ ก็คือการใช้ผู้ช่วยดูแลระบบด้วยโปรแกรมคอมพิวเตอร์ที่มีความสามารถตรวจจับสัญญาณของความผิดปกติที่เกิดขึ้นในระบบ คอยแจ้งเตือนให้กับผู้ดูแลระบบ และสามารถแก้ไขปัญหาเบื้องต้นได้ โปรแกรมแกรมคอมพิวเตอร์ดังกล่าวคือระบบตรวจจับการบุกรุก หรือ Intrusion Detection System แต่ปัญหาที่พบในปัจจุบันคือไม่มีระบบที่สามารถตรวจจับลักษณะการบุกรุกได้ครอบคลุมเนื่องจากลักษณะของการบุกรุกมีรูปแบบที่หลากหลายมาก อีกทั้งพฤติกรรมการบุกรุกแต่ละประเภทยังยากต่อการแยกแยะความแตกต่างจากพฤติกรรมของผู้ใช้ปกติหรือกว่าจะสามารถแยกแยะพฤติกรรมดังกล่าวว่าเป็นพฤติกรรมของการบุกรุกได้ก็ต่อเมื่อการบุกรุกได้กระทำความเสียหายแก่ระบบคอมพิวเตอร์ขององค์กรไปเรียบร้อยแล้ว ดังนั้นระบบตรวจจับการบุกรุกที่ดีจึงควรตรวจจับและพยายามยับยั้งการบุกรุกดังกล่าวให้ได้ก่อนที่จะทำความเสียหายต่อระบบคอมพิวเตอร์ ถึงแม้ว่าปัจจุบันมีโปรแกรมตรวจจับการบุกรุกจำหน่ายในท้องตลาดมากมาย แต่โปรแกรมต่างๆดังกล่าวก็มีข้อด้อยข้อเด่นต่างๆกันไม่มีโปรแกรมใดทำให้ได้สมบูรณ์ เช่น NetRanger RealSecure และ OmnigardIntruder เป็นต้น อีกทั้งหากมีรูปแบบการบุกรุกใหม่ๆเกิดขึ้นโปรแกรมตรวจจับการบุกรุกดังกล่าวจำเป็นต้องได้รับการปรับปรุงให้ทันสมัยโดยบริษัทตัวแทนจำหน่ายเท่านั้น ด้วยเหตุนี้ปัจจุบันนักวิจัยจึงหันมาสนใจที่จะสร้างระบบการตรวจจับการบุกรุกแบบเรียนรู้ได้ หรือสร้างด้วยแนวคิดแบบ Artificial Intelligent Systems ผู้วิจัยจึงได้นำระบบการเรียนรู้ของตัวจำแนกประเภท หรือ Learning Classifier System มาประยุกต์ใช้

1.2 ความมุ่งหมายและวัตถุประสงค์ของการวิจัย

สร้างระบบการตรวจจับการบุกรุกแบบเรียนรู้ได้ ตามแนวคิดแบบ Artificial Intelligent Systems โดยนำระบบการเรียนรู้ของตัวจำแนกประเภทมาประยุกต์ใช้ พร้อมทั้งสามารถนำไปทดสอบใช้ในหน่วยงานที่เกี่ยวข้อง

1.3 ขอบเขตของการวิจัย

ในการวิจัยนี้มีขอบเขตการวิจัยบนฐานการสร้างระบบตรวจจับการบุกรุก โดยแยกแยะพฤติกรรมการบุกรุกออกจากพฤติกรรมการใช้งานปกติ โดยตรวจจับพฤติกรรมการใช้งานที่ผิดปกติในระดับ Packet Level – มีการตรวจจับ (เฝ้ามอง) จำนวนและขนาดของ packet และชนิดการเชื่อมต่อ เป็นต้น ในขั้นตอนของการพัฒนาผู้วิจัยจำเป็นต้องใช้ system log files และ shell commands สำหรับเก็บข้อมูลการใช้งานบนระบบ เพื่อใช้เป็นข้อมูลป้อนให้กับระบบที่พัฒนาขึ้นเรียนรู้พฤติกรรมการบุกรุกในลำดับต่อไป

1.4 ขั้นตอนการศึกษา

1. ศึกษาลักษณะการบุกรุกระบบคอมพิวเตอร์
2. ศึกษาการทำงานของระบบการเรียนรู้ของตัวจำแนกประเภท หรือ Learning Classifier System แบบ accuracy-based XCS
3. สร้างโปรแกรมต้นแบบของระบบการตรวจจับการบุกรุก
4. ทดสอบโปรแกรมระบบระบบการตรวจจับการบุกรุก พร้อมทั้งแก้ไข
5. เผยแพร่ผลงานวิจัยให้กับสังคม เช่นเสนอในการประชุมวิชาการระดับชาติและระดับนานาชาติ

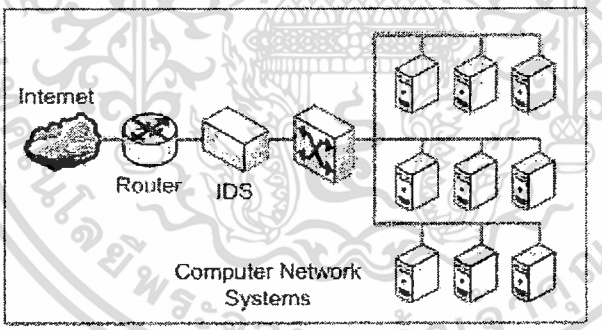
บทที่ 2

ระบบตรวจจับการบุกรุก

ในบทนี้จะกล่าวถึงระบบตรวจจับการบุกรุก หรือ Instruction Detection System (IDS) ซึ่งเป็นระบบที่ใช้ในการตรวจจับ การใช้งานและความพยายามในการใช้งาน คอมพิวเตอร์หรือเครือข่ายคอมพิวเตอร์ซึ่งขัดกับ ข้อบังคับและเจตุจำนงการใช้งาน ส่งผลต่อความปลอดภัยของระบบคอมพิวเตอร์

2.1 ความหมายของระบบตรวจจับการบุกรุก

ระบบตรวจจับการบุกรุก คือ ระบบตรวจจับสัญญาณของความผิดปกติต่างๆ ที่เกิดขึ้นในระบบที่อยู่ในขอบเขตที่ระบบนี้มีหน้าที่ตรวจสอบ ในที่นี้จะหมายถึงโปรแกรมที่ใช้สำหรับตรวจจับความผิดปกติในระบบเครือข่ายคอมพิวเตอร์เท่านั้น โดยตัวโปรแกรมจะมีความสามารถในการตรวจจับสัญญาณของความผิดปกติที่เกิดขึ้นในระบบ ไม่ว่าจะเป็นภายในระบบคอมพิวเตอร์ ระบบปฏิบัติการ โปรแกรมที่รันอยู่ในเครื่อง การทำงานกับฐานข้อมูล หรือแม้แต่ข้อมูลที่วิ่งผ่านไปมาในเครือข่ายด้วย



รูปที่ 2.1 ระบบการตรวจจับการบุกรุก

จากรูปที่ 2.1 ถ้าเรามองระบบเป็นเซตของการทำงานเซตหนึ่ง ระบบตรวจจับการบุกรุกที่แท้จริง (Ideal IDS) ต้องทราบขอบเขตของระบบว่าระบบทำงานอะไรบ้าง การทำงานใดปกติและการทำงานใดผิดปกติ โดยระบบตรวจจับการบุกรุกที่มีประสิทธิภาพ จะทราบขอบเขตของระบบโดยไม่มีการเหลื่อมล้ำเข้าไปในระบบ หรือเหลื่อมล้ำออกนอกระบบอย่างเด็ดขาด แต่การงานในโลกของความเป็นจริงอาจมีความเหลื่อมล้ำกันระหว่าง กรอบของระบบคอมพิวเตอร์ที่ระบบตรวจจับการบุกรุกรับรู้ กับระบบคอมพิวเตอร์ที่ระบบตรวจจับการบุกรุกต้องตรวจสอบ ทำให้เกิดความผิดพลาดในการตรวจสอบได้

2.2 ประเภทของระบบตรวจจับการบุกรุก

ระบบตรวจจับการบุกรุกแบ่งออกเป็นสองรูปแบบ [12] คือ ระบบที่ตรวจหาการทำงานที่ผิดไปจากการทำงานปกติของระบบ เรียกว่า Anomaly Detection ซึ่งเป็นเหมือนกับการตรวจจับคนที่ไม่มีสิทธิทำงานอยู่ในระบบ อีกรูปแบบหนึ่ง คือ ระบบที่ตรวจหาการทำงานที่ไม่ควรเกิดขึ้นในระบบเรียกว่า Misuse Detection ในที่นี้เปรียบเสมือนเป็นบุคคลที่มีสิทธิในระบบ สามารถเข้าออกในระบบได้ แต่เป็นผู้ที่ทำในสิ่งที่ระบบไม่อนุญาตให้ทำ หรือทำการใดๆ ที่อยู่นอกเหนือสิทธิของตนในระบบ

2.2.1 Anomaly Detection

แนวความคิดของการทำ Anomaly Detection คือ การหาเซตของการทำงานที่เป็นปกติย่อยๆ ขึ้นมาแล้วนำมารวมกันเพื่อให้ระบบตรวจจับการบุกรุกทราบข้อมูลของเซตการทำงานที่เป็นปกติทั้งหมดในระบบคอมพิวเตอร์ หลังจากนั้นเมื่อให้ระบบตรวจจับการบุกรุกทำงาน ถ้าเกิดกรณีที่ระบบตรวจจับการบุกรุกตรวจพบการทำงานที่ไม่ได้อยู่ในเซตของการทำงานที่เป็นปกติ ระบบตรวจจับการบุกรุกจะแจ้งเตือนต่อผู้ดูแลระบบทันที สำหรับการสร้างขอบเขตของระบบนั้นอาจสร้างได้โดยการหาข้อมูลการทำงานที่เป็นปกติในระบบขึ้นมา โดยเอาข้อมูลการทำงานของผู้ใช้งานแต่ละคน เวลาที่มีการใช้งาน ทรัพยากรที่ผู้ใช้งานคนนั้นๆ มักจะใช้บ่อยๆ หรือแม้กระทั่งข้อมูลในระบบ หรือในเครือข่ายก็สามารถนำมาสร้างเป็น เซตของระบบได้เช่นกัน

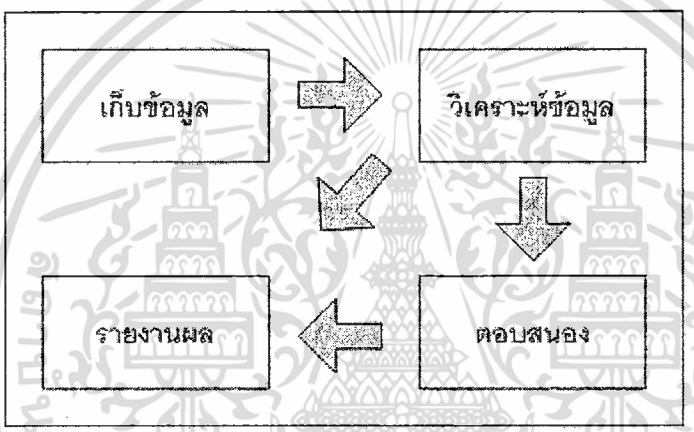
2.2.2 Misuse Detection

เป็นแนวความคิดที่ตรงข้ามกับ Anomaly Detection คือ รูปแบบนี้จะใช้ข้อมูลของการทำงานที่ผิดปกติต่างๆ ที่เคยเกิดขึ้นมาแล้ว สร้างเป็นฐานข้อมูลของการทำงานที่ผิดปกติให้ระบบตรวจจับการบุกรุกจดจำไว้ และในการทำงานของระบบตรวจจับการบุกรุกที่มีการทำงานแบบ Misuse จะนำข้อมูลที่ได้รับจากระบบคอมพิวเตอร์มาค้นหาในฐานข้อมูลว่ามีอยู่หรือไม่ ถ้าระบบตรวจจับการบุกรุกมีข้อมูลของการทำงานรูปแบบนั้นๆ อยู่ ก็แสดงว่าเกิดความผิดปกติขึ้นแล้ว ซึ่งในการทำงานของ Misuse Detection นี้ จะมีข้อเสียคือจะไม่สามารถตรวจจับการบุกรุกชนิดใหม่ๆ ได้ เนื่องจากต้องมีข้อมูลของการบุกรุกอยู่ก่อน จึงจะตรวจจับความผิดปรกติได้

2.3 การทำงานของระบบตรวจจับการบุกรุก

ระบบตรวจจับการบุกรุกแต่ละแบบมีหน้าที่การทำงานที่แตกต่างกันออกไป บางตัวจะตรวจจับความผิดปกติในระบบเครือข่าย บางตัวจะตรวจจับความผิดปกติในระบบฐานข้อมูล แต่โดยการทำงานทั้งหมดแล้วเราสามารถแบ่งการทำงานของระบบตรวจจับการบุกรุกได้เป็น 3 ขั้นตอน คือ การเก็บข้อมูลระบบ การวิเคราะห์ข้อมูลที่เก็บได้ และการรายงานผลการทำงานให้ผู้ดูแลระบบหรือผู้ที่เกี่ยวข้องทราบ

ในการทำงานหลักๆ ของระบบตรวจจับการบุกรุก อาจมีขั้นตอนเสริมอยู่ขั้นตอนหนึ่ง คือ การตอบสนองต่อการบุกรุกนั้นๆ การทำงานในขั้นตอนนี้จะใช้ในกรณีที่การบุกรุกเป็นรูปแบบการบุกรุกที่ระบบตรวจจับการบุกรุกสามารถแก้ไขด้วยตัวเองได้ ซึ่งในบางระบบอาจไม่มีการทำงานในส่วนนี้ ระบบที่ไม่มีการตอบสนองต่อการบุกรุก ส่วนใหญ่จะเป็นระบบที่ไม่ได้ทำงานแบบตอบสนองทันที (real time) คือ จะเก็บข้อมูลของระบบไว้ก่อน แล้วจึงวิเคราะห์ข้อมูลภายหลัง เมื่อทำการวิเคราะห์ข้อมูลแล้วพบว่ามีการบุกรุกเข้าสู่ระบบก็จะทำการแจ้งเตือนในขั้นตอนการทำรายงานผลการทำงาน ระบบตรวจจับการบุกรุกที่ไม่มีการตอบสนองต่อการบุกรุก ก็มักใช้ในงานที่ไม่มีความสำคัญมากนัก แต่ต้องการความถูกต้องสูง โดยลำดับการทำงานของระบบตรวจจับการบุกรุกสามารถมองเป็นขั้นตอนต่างๆ ได้ดังรูปที่ 2.2



รูปที่ 2.2 การทำงานของระบบตรวจจับการบุกรุก

2.3.1 การเก็บข้อมูลในระบบ

จากที่ได้กล่าวมาแล้วว่าระบบตรวจจับการบุกรุกจะมีการทำงานที่แตกต่างกันไป หน้าที่ในการเก็บข้อมูลของระบบที่ต้องการตรวจสอบก็แตกต่างกันไปตามหน้าที่ของระบบตรวจจับการบุกรุกด้วย โดยเราสามารถแบ่งการเก็บข้อมูลของระบบที่ต้องการตรวจสอบออกเป็นกลุ่มต่างๆ ได้ 4 กลุ่มด้วยกันคือ มีการเก็บข้อมูลในชั้นแอปพลิเคชั่น (Application-based Approach) เพื่อนำมาตรวจสอบการทำงานของแอปพลิเคชั่นต่างๆ ว่าผิดปกติหรือไม่ การเก็บข้อมูลของการทำงานของเครื่อง (Host-based Approach) เพื่อนำมาตรวจสอบการทำงานของระบบปฏิบัติการของเครื่องที่ใช้งานอยู่ การเก็บข้อมูลการเปลี่ยนแปลงข้อมูลในระบบ (Target-based Approach) เพื่อนำมาตรวจสอบว่าข้อมูลมีการเปลี่ยนแปลงอย่างไร และการเก็บข้อมูลเครือข่าย (Network-based Approach) เพื่อนำมาตรวจสอบว่ามีการบุกรุกทางระบบเครือข่ายหรือไม่ อย่างไร

2.3.1.1 การเก็บข้อมูลในชั้นแอปพลิเคชั่น

การเก็บข้อมูลในชั้นแอปพลิเคชั่นนั้น เป็นการเก็บข้อมูลที่โปรแกรมต่างๆ สร้างขึ้นมาเพื่อรายงานผลการทำงานของโปรแกรมนั้นๆ เช่น log file หรือ error message ต่างๆ ของเว็บเซิร์ฟเวอร์ ไฟร์วอลล์ หรือโปรแกรมบริหารฐานข้อมูล รวมถึงข้อมูลของการทำงานตอบสนองกันระหว่างผู้ใช้งานโปรแกรม และข้อมูลที่เกี่ยวข้อง ในการเก็บข้อมูลในลักษณะนี้ นอกจากเป็นการทำงานด้านการรักษาความปลอดภัยในระบบแล้ว ยังช่วยในการวิเคราะห์ระบบและปรับปรุงระบบเนื่องจากผลจากการวิเคราะห์ข้อมูลที่ได้ทำให้ทราบว่าการใช้งานโปรแกรมไหนในระบบมีมากน้อยอย่างไร และควรให้ความสำคัญกับการทำงานตรงส่วนไหน แต่การทำงานในส่วนนี้ก็ยังมีข้อเสีย ในกรณีที่มีการบุกรุกแล้วทำการเปลี่ยนแปลงข้อมูลดังกล่าว ทำให้การตรวจจับทำไม่ได้ ดังนั้นจึงควรเก็บข้อมูลดังกล่าวไว้ในที่ๆ ปลอดภัยด้วย

2.3.1.2 การเก็บข้อมูลของการทำงานของเครื่อง

สำหรับการเก็บข้อมูลของการทำงานของเครื่อง จะเน้นไปในการเก็บข้อมูลของระบบปฏิบัติการเป็นหลัก ข้อมูลที่เก็บได้จะอยู่ในรูปของการแจ้งเตือนในระบบเช่นการตั้งค่าบางอย่างไม่สมบูรณ์ การทำงานของโพรเซสบางโพรเซสมีปัญหาหรือปัญหาของฮาร์ดแวร์เป็นต้น หรืออาจอยู่ในรูปข้อมูลของการทำงานโดยปกติของระบบปฏิบัติการนั้นๆ เช่น ข้อมูลการใช้งานของยูสเซอร์แต่ละคน ใคร ทำอะไร เมื่อเวลาเท่าไร ซึ่งเมื่อนำข้อมูลเหล่านี้ไปวิเคราะห์แล้ว จะได้ผลการวิเคราะห์ในลักษณะมีการใช้งานอย่างไม่ถูกต้องหรือไม่ ถ้ามี ใครเป็นผู้ใช้งานนั้นๆ เมื่อเวลาเท่าไรจากที่ไหน ข้อดีอีกข้อหนึ่งก็คือการเก็บข้อมูลในลักษณะนี้สามารถเก็บข้อมูลที่ถูกเข้ารหัสได้ด้วย ส่วนข้อเสียของการเก็บข้อมูลการทำงานของเครื่องก็ คือ ข้อมูลที่ได้มักจะมีขนาดใหญ่ ระบบที่ทำการเก็บข้อมูลลักษณะนี้จะมี Overhead สูงขึ้น นอกจากนี้โปรแกรมที่ทำการเก็บข้อมูลและวิเคราะห์ข้อมูลยังขึ้นอยู่กับ platform และมีราคาสูงมากด้วย

2.3.1.3 การเก็บข้อมูลการเปลี่ยนแปลงข้อมูลในระบบ

การเก็บข้อมูลการเปลี่ยนแปลงข้อมูลในระบบจะใช้หลักการของ integrity analysis ในการตรวจสอบการเปลี่ยนแปลงข้อมูลต่างๆ ในระบบ ในระบบตรวจจับการบุกรุกบางระบบจะใช้ checksum เป็นตัวบ่งบอกการเปลี่ยนแปลงในระบบ การวิเคราะห์ลักษณะนี้จะเริ่มจากการสร้างฐานข้อมูล signature ของไฟล์ต่างๆ ในระบบปกติไว้ เมื่อระบบมีการทำงานก็จะทำการตรวจสอบค่า signature นี้ไปเรื่อยๆ อาจเป็นวันละครั้ง สองครั้ง หรือบ่อยกว่านั้นแล้วแต่ความสำคัญของระบบ เมื่อมีข้อมูลไฟล์ไหนมีการเปลี่ยนแปลงก็จะทราบได้ ข้อดีของการทำ integrity analysis ลักษณะนี้ช่วยให้การตรวจจับการบุกรุกที่มีการเปลี่ยนแปลงระบบ เช่น ทำการเจาะระบบแล้วทำการวางโทรจัน หรือ Back Door ไว้ได้ สำหรับการแก้ไขเมื่อทราบว่ามีการบุกรุกมาเปลี่ยนแปลงไฟล์ข้อมูลในระบบก็ทำการแก้ไขเฉพาะไฟล์ที่ถูกแก้ไขเท่านั้น ไม่จำเป็นต้องทำการติดตั้งระบบใหม่ แต่ระบบนี้ก็มีข้อเสียในกรณีที่เมื่อมีไฟล์ในระบบเยอะมาก การเก็บ

ข้อมูล signature ของไฟล์ต่างๆ และการวิเคราะห์ข้อมูลก็จะใช้เวลานาน ระบบนี้จึงไม่เหมาะในการทำงาน real time เพราะทำให้เกิด overhead ในระบบสูงมาก

2.3.1.4 **การเก็บข้อมูลเครือข่าย**

การเก็บข้อมูลเครือข่ายนั้นนับวันจะมีความสำคัญขึ้นเรื่อยๆ เพราะการบุกรุกทางเครือข่ายมีมากขึ้นเรื่อยๆ การเก็บข้อมูลแบบนี้จะใช้การดักจับข้อมูลที่ผ่านไปมาในเครือข่าย โดยการทำให้เน็ตเวิร์คการ์ดอยู่ใน promiscuous mode เมื่อเน็ตเวิร์คการ์ดอยู่ในโหมดดังกล่าว จะสามารถรับข้อมูลทุกอย่างที่อยู่ในเครือข่ายได้ การเก็บข้อมูลเครือข่ายในลักษณะนี้สามารถตรวจจับการโจมตีทางเครือข่ายได้ เช่น การทำ SYN flood การทำ port scan หรือการส่งแพ็กเก็ตปริมาณมากมารบกวนในระบบ แต่เนื่องจากการเก็บข้อมูลเครือข่ายนี้ใช้ลักษณะการทำสนิฟเฟอร์เป็นหลัก จึงไม่สามารถทำงานในเครือข่ายที่เป็นเครือข่ายสวิชชิ่ง ไม่สามารถทำงานในระบบเครือข่ายที่เข้ารหัสข้อมูล หรือไม่สามารถเก็บข้อมูลในเครือข่ายที่มีข้อมูลหนาแน่น ได้เพราะการทำงานในการเก็บข้อมูลอาจไม่เร็วพอที่จะเก็บข้อมูลทั้งหมดที่ผ่านไปมาในระบบได้ ข้อเสียอีกข้อหนึ่งของการเก็บข้อมูลนี้ก็คือข้อมูลที่เก็บมีขนาดใหญ่มาก โดยเฉพาะอย่างยิ่งในระบบเครือข่ายที่มีการรับส่งแพ็กเก็ตปริมาณมากอยู่ตลอดเวลา

นอกจากนี้ยังมีการเก็บข้อมูลโดยทำการเก็บข้อมูลทั้ง Application-based, Host-based และ Network-based ร่วมกันด้วย เพื่อให้ได้ข้อมูลระบบอย่างครบถ้วน และใช้ข้อมูลจากทั้งสามแหล่งมาประกอบกันในการวิเคราะห์ความผิดปกติที่เกิดขึ้นในระบบด้วย การเก็บข้อมูลในลักษณะนี้โดยทั่วไปรวมเรียกว่า "Integrated-based"

2.3.2 **การวิเคราะห์ข้อมูลระบบ**

เมื่อได้ข้อมูลของระบบที่จำเป็นแล้ว ในขั้นตอนต่อมาเราก็จะนำเอาข้อมูลที่ได้มาวิเคราะห์ว่าระบบของเรามีความผิดปกติเกิดขึ้นหรือไม่ การวิเคราะห์ข้อมูลเราสามารถแบ่งการทำงานตามรูปแบบการวิเคราะห์ข้อมูลได้ 2 รูปแบบ คือ ทำการวิเคราะห์ในขณะที่เก็บข้อมูล (Real Time) หรือ จะเก็บข้อมูลทั้งหมดไว้ก่อนแล้วจึงวิเคราะห์ข้อมูลนั้นๆ ภายหลัง (Batch) ในการวิเคราะห์ข้อมูลทั้งสองรูปแบบจะมีข้อเสียแตกต่างกันไป

2.3.2.1 **การวิเคราะห์ในขณะที่เก็บข้อมูล** (Real Time)

ในการวิเคราะห์ข้อมูลที่ได้ในขณะที่เก็บข้อมูล หรือ แบบ Real Time นั้น ระบบจะจัดเก็บข้อมูล วิเคราะห์ข้อมูล และรายงานผลการวิเคราะห์ ในช่วงเวลาเดียวกัน เมื่อเกิดข้อผิดพลาดขึ้นสามารถตอบสนองได้ทันท่วงที ระบบที่ทำงานแบบ Real Time มีการแจ้งเตือนหลายๆ แบบ เช่น E-mail หรือ Instant Messaging ให้กับผู้ดูแลระบบได้ในช่วงเวลาที่มีการบุกรุกได้ ในการตรวจสอบระบบแบบ Real Time ทำให้ระบบสามารถตรวจสอบข้อผิดพลาดได้อย่างรวดเร็ว แต่ก็ขึ้นอยู่กับความเร็วในการวิเคราะห์

ข้อมูลด้วย ถ้าข้อมูลมีความซับซ้อนมากๆ ก็จะใช้เวลามากตาม ในกรณีที่ระบบทำการตรวจสอบการบุกรุกได้ในขณะที่เพิ่งเกิดการบุกรุกขึ้น ผู้ดูแลระบบหรือระบบตรวจจับการบุกรุกเองสามารถแก้ไขปัญหาที่เกิดขึ้นได้ทันที แต่ทั้งนี้ก็ขึ้นอยู่กับความเร็วในการวิเคราะห์ข้อมูลดังที่กล่าวมาแล้ว และชนิดของปัญหาที่เกิดขึ้นด้วยว่ามีความยุ่งยากในการแก้ปัญหามากน้อยเพียงไรด้วย

ระบบ Real Time ทำงานได้อย่างรวดเร็ว แต่การทำงานที่รวดเร็วดังกล่าวก็ต้องแลกกับการใช้หน่วยความจำปริมาณมากและการประมวลผลที่รวดเร็วมากด้วย อีกทั้งการตอบสนองต่อการบุกรุกโดยอัตโนมัติ อาจทำให้เกิดความเสียหายกับระบบมากกว่าเดิม เพราะว่าในบางครั้งการทำงานที่เร็วเกินไปของระบบนี้ ทำให้เกิดความผิดพลาดในการวิเคราะห์ข้อมูลจนประมวลผลการทำงานที่เป็นปกติกลายเป็นการทำงานที่ผิดปกติ และทำการแก้ไขตามข้อมูลที่มีอยู่ ก็จะยิ่งทำให้ระบบมีความเสียหายมากกว่าเดิม ระบบตรวจจับการบุกรุกที่ทำงานแบบ Real Time จึงเหมาะกับระบบที่มีพฤติกรรมการทำงานปกติไม่มีแบบซึ่งจะทำให้ลดความผิดพลาดในการวิเคราะห์ ต้องการการรายงานอย่างรวดเร็วเมื่อผิดปกติ และข้อมูลที่ต้องนำมาวิเคราะห์ไม่ซับซ้อนมากนัก

2.3.2.2 **การวิเคราะห์ข้อมูลภายหลังจากที่เก็บข้อมูล** (Batch)

อีกรูปแบบหนึ่งในการวิเคราะห์ระบบที่ใช้กัน คือ การวิเคราะห์ข้อมูลภายหลังจากที่เก็บข้อมูลไว้แล้ว หรือการที่ทำงานแบบ Batch การทำงานในแบบนี้เหมาะกับงานที่ไม่จำเป็นต้องตอบสนองทันทีเมื่อเกิดความผิดปกติขึ้น แต่ให้มีการบันทึกและรายงานว่าเกิดความผิดปกติขึ้น การทำงานจะใช้หน่วยความจำและการประมวลผลน้อยกว่าแบบแรก แต่ก็ใช้เนื้อที่ในการเก็บข้อมูลมากกว่าแบบแรก ข้อเสียของการทำงานแบบ Batch คือ มักแก้ปัญหาที่เกิดขึ้นไม่ทัน เพราะกว่าจะทราบว่าเกิดปัญหาขึ้น ปัญหานั้นก็เกิดขึ้นนานมาก ความเสียหายที่เกิดขึ้นก็แก้ไขได้ยาก

ไม่ว่าจะเป็นการวิเคราะห์ระบบแบบ Real Time หรือ Batch ก็จะมีวิธีการวิเคราะห์ระบบที่เหมือนกัน คือ ทำการหารูปแบบของการโจมตีในข้อมูลที่ได้รับมา (Signature Analysis) วิธีการวิเคราะห์ทางสถิติ (Statistical Analysis) และวิธีการตรวจสอบการเปลี่ยนแปลงของระบบ (Integrity Analysis)

1) **การหารูปแบบของการโจมตี** (Signature Analysis)

ในวิธีการวิเคราะห์ระบบแบบ Signature Analysis เป็นการวิเคราะห์ข้อมูลโดยการหาสัญญาณของการโจมตี (Attack Signature) การทำงานจะทำโดยการเปรียบเทียบรูปแบบของข้อมูลกับรูปแบบของการโจมตีในฐานข้อมูล ว่ามีความคล้ายกันหรือไม่ ถ้ามีความคล้ายคลึงกันก็แสดงว่ามีการโจมตีเกิดขึ้นแล้ว ในการเปรียบเทียบอาจเป็นแบบอย่างง่าย คือ การเปรียบเทียบข้อมูลว่ามีความคล้ายกับข้อมูลของการโจมตีเพียงใด หรือเป็นแบบที่มีความสลับซับซ้อนขึ้นอีก เช่น การทำ state transition เป็นต้น

สำหรับโปรแกรมตรวจจับการบุกรุกที่มีจำหน่ายในท้องตลาด ส่วนใหญ่จะทำงานในลักษณะของการเปรียบเทียบรูปแบบกับการโจมตีในฐานข้อมูล ซึ่งบริษัทผู้ขายจะให้ฐานข้อมูลของการโจมตีไว้ด้วย ผู้ใช้งานจะมีการอัพเดทข้อมูลในฐานข้อมูลบ่อยๆ เพื่อเพิ่มความสามารถในการวิเคราะห์ข้อมูลในระบบการวิเคราะห์ระบบด้วยวิธี Signature analysis นี้จะมี overhead ไม่มากนักเพราะเป็นเพียงการเปรียบเทียบข้อมูลกับข้อมูลในฐานข้อมูลเท่านั้น และยังเพิ่มความรวดเร็วในการทำงานโดยรวมมากขึ้น เพราะสามารถนำข้อมูลในฐานข้อมูลเป็นกฎในการกรองข้อมูลที่จะเก็บให้น้อยลงไปด้วย แต่วิธีการนี้ก็มีข้อเสียเพราะฐานข้อมูลจะมีขนาดใหญ่ขึ้นเรื่อยๆ ต้องมีการอัพเดทฐานข้อมูลบ่อยๆ

2) **วิธีการวิเคราะห์ทางสถิติ (Statistical Analysis)**

วิธีการวิเคราะห์ทางสถิติ (Statistical Analysis) เป็นวิธีการวิเคราะห์ข้อมูลอีกแบบหนึ่งที่มีแนวคิดตรงข้ามกับวิธีการแรก คือ จะหารูปแบบของการทำงานที่เป็นปกติ แล้วสร้างเป็นโพรไฟล์ (Profile) ที่สามารถปรับปรุงได้อัตโนมัติตามพฤติกรรมการใช้งานเก็บไว้ ในการวิเคราะห์ข้อมูล จะเปรียบเทียบข้อมูลกับโพรไฟล์ที่สร้างไว้ ถ้าไม่เข้ากันก็แสดงว่ามีความผิดปกติเกิดขึ้นแล้ว สำหรับโพรไฟล์นั้นอาจแยกเป็นเป็นโพรไฟล์สำหรับออปเจ็กต์ต่างๆ ในระบบ เช่น ยูสเซอร์ ไฟล์ ไดเร็กทอรี และอุปกรณ์ต่างๆ โดยรวมละเอียดที่เก็บอยู่ในโพรไฟล์จะมีข้อมูลของจำนวนครั้งที่เข้าสู่ระบบ จำนวนครั้งที่เข้าสู่ระบบผิดพลาด เวลา และข้อมูลอื่นๆ ที่จำเป็น ค่าแต่ละค่าที่เก็บจะเป็นค่าของการใช้งานที่เป็นปกติ การตรวจจับว่าเกิดความผิดปกติขึ้นแล้ว จะดูจากค่าที่ไม่เข้ากับโพรไฟล์ เป็นต้น ยกตัวอย่างเช่น ในการทำงานปกติ ผู้ใช้งานฐานข้อมูลจะมีการเข้าใช้ข้อมูลในฐานข้อมูลตั้งแต่เวลา 8 โมงเช้า ถึง 6 โมงเย็นเท่านั้น แต่ข้อมูลที่ตรวจจับได้มีการเข้าใช้ฐานข้อมูลตอนตีสอง ซึ่งก็สามารถบอกได้ว่ามีการบุกรุกเกิดขึ้นแล้ว

เนื่องจากวิธีการวิเคราะห์ข้อมูลแบบ Statistical Analysis เป็นการตรวจจับโดยใช้หลักการของการอนุญาตให้ใช้งานในการทำงานทั่วๆ ไปและไม่อนุญาตให้ใช้งานนอกเหนือจากที่เคยใช้โดยทั่วไปเท่านั้น การตรวจจับในลักษณะนี้จึงสามารถตรวจจับการบุกรุกที่ไม่เคยเจอมาก่อนได้ และสามารถตรวจจับการบุกรุกในรูปแบบที่ซับซ้อนได้ด้วย เพราะเราถือว่าการทำงานที่สลับซับซ้อนมักจะไม่เหมือนกับการทำงานโดยปกติ แต่วิธีการวิเคราะห์ข้อมูลแบบนี้ก็มีข้อเสียเนื่องจากเป็นการเก็บข้อมูลการทำงานที่เป็นปกติไว้เพื่อเปรียบเทียบกับการทำงานที่ผิดปกติ ดังนั้นเมื่อค่อยๆ เพิ่มความซับซ้อนในการบุกรุกเป็นเวลานานโดยไม่ทำให้ผิดปกติจากโพลไฟล์เดิมมากเกินไปและค่อยๆ ทำซ้ำไปเรื่อยๆ ก็จะทำให้โพรไฟล์มีการเปลี่ยนแปลง ระบบตรวจจับก็จะเห็นว่าการโจมตีในลักษณะนั้นเป็นการทำงานที่เป็นปกติแทน และไม่สามารถตรวจจับการทำงานที่ผิดปกติในลักษณะนั้นได้อีกต่อไป และไม่เหมาะกับองค์กรที่มีการเปลี่ยนแปลงการทำงานบ่อยๆ เพราะทำให้โพรไฟล์มีขนาดใหญ่ ทำให้ระบบตรวจจับมีค่าความแปรปรวนสูงและมีความผิดพลาดในการวิเคราะห์สูง

3) **วิธีการตรวจสอบการเปลี่ยนแปลงของระบบ** (Integrity Analysis)

วิธีสุดท้ายที่นิยมใช้กันในการวิเคราะห์ข้อมูลคือ วิธีการตรวจสอบการเปลี่ยนแปลงของระบบ (Integrity Analysis) ลักษณะการทำงานของวิธีการนี้ คือ การหาว่ามีการเปลี่ยนแปลงเกิดขึ้นในระบบหรือไม่ เช่น มีไฟล์ไหนมีการเปลี่ยนแปลง หรือมีออปเจ็กต์อะไรที่มีการเปลี่ยนแปลงคุณสมบัติบ้าง แล้วทำการแจ้งเตือนกับผู้ดูแลระบบ ในการวิเคราะห์ลักษณะนี้จะใช้แฮชอัลกอริธึม (hash algorithm) เพื่อสร้างแมสเซสไดเจส (message digest) ของข้อมูล แล้วทำการเปรียบเทียบแมสเซสไดเจส ของข้อมูลในช่วงเวลาต่างๆ ว่าเหมือน หรือต่างกันหรือไม่ ถ้าแมสเซสไดเจสต่างกันก็แสดงว่าข้อมูลมีการเปลี่ยนแปลง Integrity Analysis สามารถตรวจจับการบุกรุก ที่เข้ามาเปลี่ยนแปลงข้อมูลในระบบ หรือมีการติดตั้งโปรแกรม เช่น sniffer rootkit ต่างๆ ในระบบได้ แต่ก็มีข้อเสีย คือ การวิเคราะห์ระบบในลักษณะนี้จะทำงานเป็นแบบ batch เท่านั้น ไม่เหมาะกับการทำ real time อย่างยิ่งเพราะจะทำให้เปลืองทรัพยากรมาก

2.3.3 **การตอบสนอง**

เมื่อมีการตรวจพบว่ามีการบุกรุกเกิดขึ้นในระบบ สำหรับระบบตรวจจับที่ทำงานแบบ real time จะมีการตอบสนองต่อการบุกรุกเพื่อไม่ให้เกิดความเสียหาย หรือบรรเทาความเสียหายที่เกิดขึ้นสำหรับระบบที่ทำงานเป็นแบบ batch การตอบสนองอาจทำได้ไม่มากนัก เพราะการบุกรุกนั้นเกิดไปแล้ว ความเสียหายก็เกิดขึ้นแล้ว การตอบสนองอาจอยู่ในรูปแบบการบรรเทาไม่ให้ความเสียหายมีเพิ่มมากขึ้นเท่านั้น การตอบสนองต่อการบุกรุกนั้นแบ่งออกได้เป็นสามแบบด้วยกัน คือ การเปลี่ยนแปลงสภาพของระบบ การแก้ไขความผิดพลาดให้ถูก และการแจ้งเตือนผู้ดูแลระบบเมื่อถูกบุกรุก

2.3.3.1 **การเปลี่ยนแปลงสภาพของระบบ**

สำหรับการตอบสนองต่อการบุกรุกโดยการเปลี่ยนแปลงสภาพของระบบที่ถูกโจมตีก็เพื่อแก้ปัญหาหรือลดความเสียหายที่จะเกิดขึ้น เช่น ตัดการเชื่อมต่อระหว่างระบบกับการบุกรุกออกจากกัน การตั้งค่าอุปกรณ์เครือข่าย หรือไฟร์วอลล์ไม่ให้มีการติดต่อกับระบบของการบุกรุกอีกต่อไป และการหาข้อมูลเกี่ยวกับการโจมตีโดยอัตโนมัติเพื่อตรวจหาการบุกรุกต่อไป

2.3.3.2 **การแก้ไขความผิดพลาดให้ถูก**

การแก้ไขระบบ เป็นการตอบสนองต่อปัญหาที่เกิดขึ้นแล้วในระบบ โดยปกติแล้วการบุกรุกมักเปลี่ยนแปลงค่าต่างๆ ในระบบ โดยเฉพาะเข้ามาทำการเปลี่ยนแปลงข้อมูลในระบบตรวจจับการบุกรุกเพื่อไม่ให้สามารถตรวจจับการบุกรุกได้ การแก้ไขระบบก็เพื่อให้ระบบดังกล่าวสามารถทำงานได้อย่างเป็นปกติ

2.3.3.3 **การแจ้งเตือนผู้ดูแลระบบ**

สุดท้ายเป็นการแจ้งเตือนผู้ดูแลระบบ โดยปกติมักแจ้งเตือนผู้ดูแลทันทีเมื่อทำการวิเคราะห์ได้ว่ามีความผิดปกติเกิดขึ้น เพื่อให้ผู้ดูแลระบบรับรู้และสามารถแก้ไขระบบได้ทันท่วงที สำหรับการแจ้งเตือนนี้

ผู้ดูแลระบบสามารถเลือกได้ว่าจะแจ้งเตือนใครบ้าง และทำการแจ้งเตือนในรูปแบบไหนอาจเป็น E-mail, Pager หรือ Instant Messaging ต่างๆ

2.3.4 การรายงานผลการทำงาน

เมื่อระบบตรวจจับการบุกรุกทำการวิเคราะห์ระบบ และตรวจพบความผิดปกติในระบบ อาจมีการตอบสนองต่อความผิดปกตินั้นถ้าทำได้ จากนั้นระบบตรวจจับการบุกรุกต้องมีการรายงานผลให้กับผู้ดูแลระบบทราบในรูปแบบต่างๆ โดยรายละเอียดของการรายงานผลนั้น จะบอกถึงช่องโหว่ในระบบ การแก้ไขปัญหาคร่าวๆ บางครั้งอาจมีรายละเอียดของความรู้พื้นฐานบางอย่างของระบบ ที่ทำให้เกิดการบุกรุกลักษณะนั้นๆ ได้ การรายงานผลการทำงาน นอกจากเป็นการรายงานต่อผู้ดูแลระบบเพื่อให้ทราบการทำงาน หรือจุดอ่อนในระบบแล้ว ยังเป็นประโยชน์ต่อการวิเคราะห์สถานะของระบบ และการวิเคราะห์ความปลอดภัยในระบบอีกด้วย

2.4 ความสำคัญของระบบตรวจจับการบุกรุก

เมื่อได้ทราบถึงการทำงานคร่าวๆ ของระบบตรวจจับการบุกรุกแล้ว อาจคิดว่าระบบตรวจจับการบุกรุกไม่มีความสำคัญเพราะในเมื่อมีการใช้งานไฟร์วอลล์อยู่แล้ว แต่ความเป็นจริงแล้วถึงแม้ว่าระบบจะมีไฟล์วอลล์อยู่แล้วก็ยังจำเป็นต้องใช้ระบบตรวจจับการบุกรุกด้วยเพราะในงานบางอย่างไฟร์วอลล์ก็ไม่สามารถช่วยได้

จุดประสงค์ของการใช้งานไฟร์วอลล์นั้น สร้างขึ้นเพื่อเป็นเสมือนตัวป้องกันระบบให้แยกตัวออกมาจากเครือข่ายที่ไม่ปลอดภัย เป็นเหมือนเมืองหน้าด่านของระบบ เป็นผู้ป้องกันการบุกรุกจากภายนอก แต่ระบบตรวจจับการบุกรุกนั้นมีจุดประสงค์ที่แตกต่างไป โดยเป็นผู้เฝ้าดูระบบ และเป็นผู้เตือนเมื่อเกิดความผิดปกติเกิดขึ้น ยกตัวอย่างในอาคารใหญ่ๆ จะมี คนคอยดูแลอยู่ภายนอกกับคนที่ไม่ควรเข้ามาในอาคารให้อยู่ภายนอก แต่ภายในตัวอาคารก็จะมีกล้องวีดีโอคอยตรวจตราอยู่ภายในมีกริ่งสัญญาณเตือนเมื่อเกิดความผิดปกติเกิดขึ้น ซึ่งก็ช่วยให้แก้ปัญหาได้ทันท่วงที และในกรณีที่มีความผิดปกติเกิดขึ้น แต่ไม่สามารถตรวจจับได้ในขณะนั้น ระบบตรวจจับการบุกรุกก็มีการจัดเก็บข้อมูลการใช้ระบบไว้ จึงสามารถนำข้อมูลดังกล่าวมาวิเคราะห์หาความผิดปกติได้ภายหลัง โดยจุดประสงค์ในการสร้างระบบตรวจจับการบุกรุกและไฟร์วอลล์ จึงต่างกันโดยสิ้นเชิง แต่ถึงแม้ว่าจุดประสงค์การทำงานของการตรวจจับการบุกรุกและไฟร์วอลล์จะแตกต่างกัน ทั้งสองโปรแกรมก็สามารถทำงานร่วมกันและทำให้ประสิทธิภาพการรักษาความปลอดภัยในระบบดีขึ้นด้วย

บทที่ 3

ระบบตรวจจับการบุกรุกด้วยระบบการเรียนรู้ของตัวจำแนกประเภท

ในบทนี้จะกล่าวถึงหลักการ และการโครงสร้างของระบบตรวจจับการบุกรุกด้วยระบบการเรียนรู้ของตัวจำแนก โดยอธิบายโครงสร้างของระบบทั้งหมด พร้อมทั้งรายละเอียดของแต่ละส่วนตามลำดับ

3.1 บทนำและงานวิจัยที่เกี่ยวข้อง

การตรวจจับการบุกรุกเราสามารถมองว่าเป็นปัญหาเดียวกับการจัดกลุ่มข้อมูล (classification problem) [17] คือระบบต้องสามารถแยกข้อมูลที่เป็นข้อมูลพฤติกรรมปรกติ และข้อมูลพฤติกรรมที่ผิดปรกติ พร้อมทั้งแยกตามประเภทของการบุกรุกออกเป็นกลุ่มๆ ได้ จากที่กล่าวมาในบทก่อนหน้าการตรวจจับการบุกรุกแบ่งออกเป็นสองรูปแบบ คือ Anomaly Detection และ Misuse Detection ซึ่งทั้งสองรูปแบบ มีงานวิจัยที่นำเสนอออกมาอย่างมากมาย อาทิเช่น Xin Xu [19] นำเสนอระบบตรวจจับการบุกรุกโดยอาศัยอัลกอริธึม SVM เข้ามาช่วยในการจัดกลุ่มข้อมูลประเภทการบุกรุกแบบต่างๆ อย่างไรก็ตามผลการทดลองที่ได้จากการทดลองกับฐานข้อมูล KDD Cup 1999 [7] ให้ค่าความถูกต้องของการตรวจจับการบุกรุกแต่ล่ะประเภทได้ไม่สูงมากนัก Liberios และทีมงาน [21] นำเสนอระบบตรวจจับการบุกรุกแบบ Anomaly Dectection ซึ่งเป็นการประยุกต์ใช้นิวรอนเน็ตเวิร์กที่เรียกว่าเซลฟ์ออร์แกนไนสซิ่งแม็พ (Self-Organizing Map) หรือ SOM มาใช้งานในการแบ่งแยกการบุกรุกออกจากการใช้งานปกติ โดยข้อมูลการใช้งานจะถูกจัดกลุ่มไปตามโหนดต่างๆ ของแผนภาพ SOM ซึ่งแผนภาพจะพยายามจัดกลุ่มผู้ใช้งานปกติแบ่งแยกออกจากผู้ใช้งานที่อาจจะเป็นการบุกรุก

ในขณะเดียวกันก็มีงานวิจัยที่นำเอาระบบการสร้างกฎมาประยุกต์ใช้ในระบบตรวจจับการบุกรุก เนื่องจากกฎที่ได้หลังจากการเรียนรู้อยู่ในรูปแบบที่ผู้ใช้สามารถอ่านและทำความเข้าใจได้ง่าย ในระยะเริ่มต้นการสร้างกฎในฐานข้อมูลกฎจะต้องถูกกำหนดจากผู้เชียวชาญ แต่ในระยะต่อมาได้มีการทำเอาเทคนิคการสร้างกฎอัตโนมัติจากฐานข้อมูลที่ใช้ในการเรียนรู้ Agarwal และ Joshi [20] ได้นำเสนอระบบการสร้างกฎที่ชื่อว่า PNrule เพื่อใช้ในการเรียนรู้พฤติกรรมการบุกรุกประเภทต่างๆ โดยใช้ฐานข้อมูล KDD Cup 1999 โมเดลที่นำเสนอสามารถให้ค่าความถูกต้องในการตรวจจับประเภท DoS และ Probe ได้สูง แต่ค่าความถูกต้องของการบุกรุกแบบ U2R และ R2L ยังไม่สูงมากนัก การบุกรุกบางประเภทมีลักษณะซ้อนทับกัน และบางประเภทจะมีจำนวนข้อมูลที่ใช้ในการเรียนรู้น้อย ซึ่งก็ยังมีงานวิจัยอีกมากที่พัฒนาระบบการสร้างกฎแบบอัตโนมัติ [22][23][24][25] ซึ่งระบบต่างๆ เหล่านี้จะสร้างกฎจากฐานข้อมูลตัวอย่างและเป็นกฎที่ตายตัว บางระบบเป็นการสร้างกฎเฉพาะข้อมูลการใช้งานแบบปกติเท่านั้น ซึ่งหากมีพฤติกรรม

การบุกรุกแบบใหม่จำเป็นที่จะต้องทำการปรับปรุงกฎใหม่ให้ทันสมัยตลอดเวลา ด้วยเหตุนี้ปัจจุบันนักวิจัยจึงหันมาสนใจที่จะสร้างระบบการตรวจจับการบุกรุกแบบที่กฎในฐานข้อมูลสามารถปรับเปลี่ยนค่าได้

ระบบการเรียนรู้ของตัวจำแนกประเภท หรือ Learning Classifier System (LCS) เป็นเทคนิคการเรียนรู้ของเครื่องจักกลรูปแบบหนึ่ง ที่ได้รวมเทคนิคหลายอย่างเอาไว้ในระบบเดียวกันเช่น evolution computing, reinforcement learning, supervised learning และ adaptive systems ในระบบนี้จะมีการทำงานบนฐานข้อมูลกฎ (rulebase) จะประกอบไปด้วยกฎ (rule) หลายๆ กฎ โดยที่แต่ละ กฎจะถูกสร้างขึ้นอย่างง่ายๆ ในรูปแบบ "IF state THEN action" ในส่วนของ evolutionary algorithm และ heuristics จะถูกใช้ในการค้นหากฎที่เป็นไปได้ทั้งหมด โดยยึดหลักในการค้นหากฎที่ดีกว่ากฎที่มีอยู่แล้วในระบบฐานข้อมูลกฎ และในส่วนของการปรับค่าพารามิเตอร์ต่างๆ ที่บ่งบอกถึงความเหมาะสมของแต่ละกฎสำหรับการแก้ปัญหาแต่ละปัญหาจะใช้วิธีการของ reinforcement learning ต้นแบบของ LCS ถูกสร้างขึ้นโดย John Holland [8] หลังจากนั้นก็ได้มีการพยายามสร้างระบบที่ง่ายขึ้น และมีประสิทธิภาพดีขึ้นมาตลอดเวลา จนกระทั่ง Stewart Wilson [15] ได้สร้างระบบที่เรียกว่า XCS ซึ่งในปัจจุบันนี้ถือว่าเป็นระบบ LCS ชนิด accuracy-based ที่นิยมมาก ระบบ LCS นี้สามารถแก้ปัญหาที่นิยมใช้ในการทดสอบ(well-known problems) ได้ดีในหลายๆ ด้าน เช่น data mining [26], simulation modelling, robotics และ adaptive control [3] นอกจากนั้นแล้วได้มีการนำระบบ XCS ไปประยุกต์ใช้กับระบบตรวจจับการบุกรุก [13] ข้อมูลที่นิยมใช้สำหรับงานวิจัยระบบตรวจจับการบุกรุกคือข้อมูลจำลองพฤติกรรมการบุกรุกที่ได้จาก KDD Cup 1999 โดยประสิทธิภาพของระบบสามารถการจัดกลุ่มข้อมูลพฤติกรรมและแยกประเภทการบุกรุกออกเป็นกลุ่มๆ ได้ นอกจากนั้นยังได้มีการปรับปรุงระบบ XCS บางส่วนเพื่อให้เหมาะกับการแก้ปัญหา และมีประสิทธิภาพมากขึ้น

ระบบการเรียนรู้ของตัวจำแนกประเภท(LCS)ถือว่าเป็นระบบหนึ่งที่มีจุดเด่นในการสร้างและพัฒนากฎในฐานข้อมูลกฎดังนั้นจึงเหมาะกับการนำไปประยุกต์ใช้กับปัญหาของการตรวจจับการบุกรุก โดย ในปี 2006 [13],[14] Shafi และทีมงานทดลองนำระบบการเรียนรู้ของตัวจำแนกประเภทที่เรียกว่า XCS มาทดลองสร้างระบบตรวจจับการบุกรุกซึ่งผลการทดลองออกมาค่อนข้างน่าพอใจและแสดงให้เห็นว่าระบบ XCS สามารถนำมาสร้างเป็นระบบตรวจจับการบุกรุกได้ แต่ยังคงต้องพัฒนาปรับปรุงบางส่วนอยู่ ในปี 2007 [18] Shafi และทีมงานได้นำเสนอการศึกษาเปรียบเทียบระบบ XCS ที่ได้รับการพัฒนากับระบบการเรียนรู้ของตัวจำแนกประเภทอีกระบบที่เรียกว่า UCS ผลการทดลองแสดงกับฐานข้อมูล KDD Cup 1999 ให้เห็นว่าระบบ XCS ที่ Shafi ได้พัฒนานั้นสามารถใช้งานเป็นระบบตรวจจับการบุกรุกได้เป็นอย่างดี

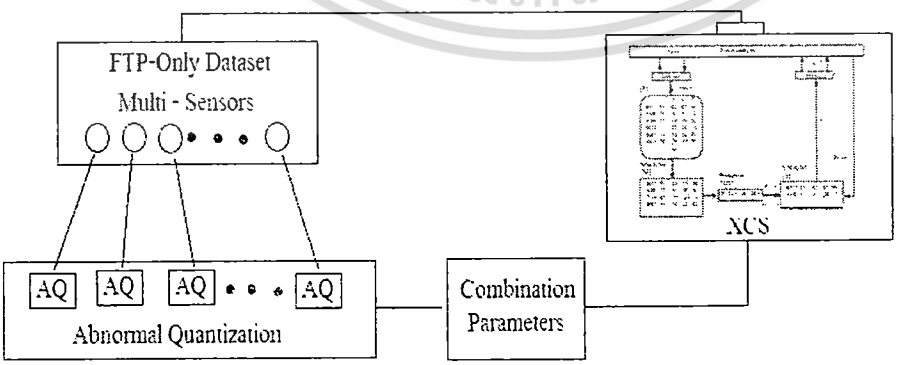
ในงานวิจัยฉบับนี้ได้มีส่วนหลักที่เพิ่มเติมจากงานวิจัยดังกล่าว คือปรับข้อมูลแต่ละพารามิเตอร์ออกเป็นระดับความผิดปรกติ (Abnormal Quantization) โดยนำวิธีการเซลฟ์ออร์แกนไนสซิ่งแม็พ (Self-Organizing Map) หรือ SOM [9] เข้ามาช่วยในการกำหนดระดับความผิดปรกติดังกล่าว หลักจากนั้นก็ส่งให้ระบบ XCS เพื่อทำการเรียนรู้จัดกลุ่มข้อมูลพฤติกรรมและแยกประเภทการบุกรุกออกเป็นกลุ่มๆ จากผล

การทดลองทั้งหมดในงานวิจัยนี้แสดงให้เห็นว่าระบบเราได้นำเสนอมีประสิทธิดีกว่าระบบการเรียนรู้ด้วนวิธีอื่นๆ

3.2 โครงสร้างการทำงานของระบบ

ในงานวิจัยนี้เราได้นำเสนอส่วนขยายของระบบตรวจจับการบุกรุกด้วยระบบการเรียนรู้ของตัวจำแนกประเภท โดยอาศัยแนวคิดที่ว่าพฤติกรรมการใช้งานแบบปกตินั้นอาจจะมีอยู่หลายลักษณะไม่สามารถที่จะกำหนดขอบเขตตายตัวได้ เช่น กรณีการวิเคราะห์คุณลักษณะจำนวนของแพคเกจที่เข้าออก บางกลุ่มคนมีการใช้งานน้อยทำให้จำนวนแพคเกจที่เข้าออกน้อย บางกลุ่มคนมีกาใช้งานมากทำให้จำนวนแพคเกจที่เข้าออกมาก ซึ่งเราไม่สามารถกำหนดได้ว่าถ้ามีการใช้งานแพคเกจมากเกินกว่าค่าที่เรากำหนดแล้วให้ถือว่าเป็นการทำงานที่ผิดปกติได้ ดังนั้นเพื่อให้มีความละเอียดในการวิเคราะห์มากขึ้น เราจึงได้นำเสนอการใช้งานแผนภาพ SOM เพื่อทำการจัดกลุ่มพฤติกรรมการใช้งานแบบปกติออกเป็นกลุ่มๆ หลังจากนั้นจึงวิเคราะห์การใช้งานของแต่ละกลุ่มว่ามีการใช้งานแพคเกจโดยปกติเฉลี่ยประมาณเท่าไหร่ ในกรณีที่ตรวจจับการใช้งานแพคเกจเกิดขึ้น ระบบก็จะทำการจัดกลุ่มการใช้งานนั้นก่อน จากนั้นจึงวิเคราะห์ว่าการใช้งานดังกล่าวมีค่าความเบี่ยงเบนไปจากพฤติกรรมของกลุ่มการใช้งานนั้นมากน้อยเพียงใด โดยเราได้แบ่งระดับของความเบี่ยงเบนออกเป็นหลายระดับเพื่อใช้ในการบ่งบอกว่าการใช้งานนั้นมีโอกาสเป็นการบุกรุกมากน้อยเพียงใด รายละเอียดจะกล่าวถึงในหัวข้อถัดไป

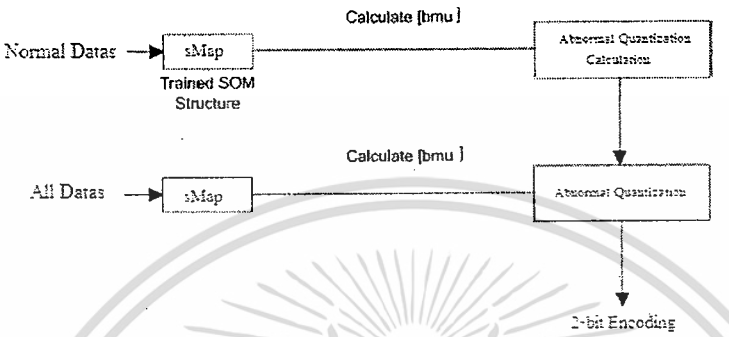
ระบบที่นำเสนอประกอบไปด้วยส่วนประกอบต่างๆ ดังนี้คือ ส่วนการกำหนดระดับความผิดปรกติ (Abnormal Quantization) ส่วนการรวมพารามิเตอร์ (Combination parameters) และส่วนระบบการเรียนรู้ของตัวจำแนกประเภท (XCS) ตามรูปที่ 3.1 โดยรายละเอียดในแต่ละส่วนจะอธิบายตามหัวข้อต่อไปนี้



รูปที่ 3.1 ส่วนต่างๆ ของระบบตรวจจับการบุกรุกด้วยระบบการเรียนรู้ของตัวจำแนกประเภท

3.3 การกำหนดระดับความผิดปรกติ

ในงานวิจัยนี้ได้นำเซลฟ์ออร์แกนไนสซิ่งแม็พ (Self-Organizing Map) หรือ SOM มาใช้ในการกำหนดระดับความผิดปรกติ (Abnormal quantization) ในแต่ละพารามิเตอร์ (Parameter) มีขั้นตอนการทำงานตามรูปที่ 3.2 แบ่งการทำงานเป็นสองส่วนดังนี้



รูปที่ 3.2 ขั้นตอนการทำงานในส่วนของการกำหนดระดับความผิดปรกติ

การคำนวณค่าระดับความผิดปรกติ (Abnormal Quantization Calculation)

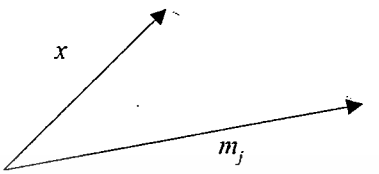
แต่ละโมเดลของ SOM กำหนดให้แทนแต่ละพารามิเตอร์ โดยจะถูกเรียนรู้ด้วยข้อมูลเฉพาะข้อมูลที่มีรูปแบบพฤติกรรมการใช้งานปรกติ (Normal) ข้อมูลแต่ละพารามิเตอร์จะกำหนดให้เป็นอินพุตเวคเตอร์ที่มีขนาด 1 มิติ โมเดลของ SOM ประกอบด้วยเซลล์ 2 ชั้น ชั้นแรกคือชั้นของอินพุต (Input layer) ประกอบไปด้วยเซตของอินพุตเวคเตอร์มีขนาด n มิติ (ในงานวิจัยนี้ n เท่ากับ 1) ชั้นที่สองคือชั้นของแผนภาพโคโฮเนน (Kohonen layer) หรือชั้นของเอาท์พุตประกอบไปด้วยโหนดของนิวรอนที่เรียงตัวอยู่ในรูปแบบของแผนภาพ 1 มิติ (ในงานวิจัยนี้กำหนดให้มี 5 โหนด) ในแต่ละโหนดจะมีค่าเวคเตอร์น้ำหนัก (Weight Vector) เป็นตัวแทนประจำโหนด

กระบวนการเรียนรู้ของ SOM เกิดขึ้นจากการปรับตัวของเวคเตอร์น้ำหนักที่มีต่ออินพุตเวคเตอร์ โดยเริ่มแรกจะกำหนดค่าเวคเตอร์น้ำหนักเริ่มต้นจากการสุ่มค่าที่อยู่ในช่วง 0 ถึง 1 จากนั้นจะเริ่มต้นกระบวนการเรียนรู้ดังนี้

1. เลือกอินพุตเวคเตอร์แบบสุ่มเลือกจากอินพุตโดเมน
2. เปรียบเทียบอินพุตเวคเตอร์ $x(t)$ กับโหนด $m_i(t)$ ทุกโหนดเพื่อหาโหนดที่ใกล้เคียงกับอินพุทเวคเตอร์ที่สุดจากโหนดทั้งหมด โดยเรียกโหนดที่ใกล้ที่สุดนี้ว่า โหนดชนะ (winning node)
3. ปรับเวคเตอร์น้ำหนักของโหนดชนะ เพื่อให้โหนดชนะเข้าใกล้อินพุตมากขึ้น
4. ปรับเวคเตอร์น้ำหนักของโหนดใกล้เคียง เพื่อให้อินพุตเวคเตอร์ถัดไปที่มีค่าใกล้เคียงมีโหนดชนะใหม่อยู่ใกล้กัน

กระบวนการเหล่านี้จะถูกทำซ้ำไปเรื่อยๆ จนกว่าจะสอดคล้องตามเงื่อนไขหรือจนกว่าจะครบจำนวนรอบของการเรียนรู้ จากกระบวนการเรียนรู้ข้างต้นมีการคำนวณที่สำคัญอยู่ 2 ส่วนคือ

ส่วนแรกคือการคำนวณเพื่อหาโหนดชนะ(ขั้นตอนที่ 2) ในการคำนวณหาโหนดชนะอินพุตเวคเตอร์ $x(t)$ ถูกนำไปเปรียบเทียบกับโหนด $m_i(t)$ ทุกโหนดเพื่อหาโหนดชนะจากโหนดทั้งหมด ฟังก์ชั่นที่ใช้ในการเปรียบเทียบโดยทั่วไปแล้วจะใช้ฟังก์ชั่นวัดระยะทางแบบยูคลิด (Euclidean distance) ดังรูปที่ 3.3



รูปที่ 3.3 แสดงระยะทางแบบยูคลิดระหว่างเวคเตอร์ x และ m_j

การหาโหนดที่ชนะ c สามารถหาได้จากโหนดที่มีระยะห่างระหว่างอินพุตเวคเตอร์กับเวคเตอร์น้ำหนักของโหนดนั้นน้อยที่สุดดังสมการที่ 3.1

$$c : m_c(t) = \min_i \| x(t) - m_i(t) \| \tag{3.1}$$

ส่วนที่สองคือการปรับเวคเตอร์น้ำหนัก(ขั้นตอนที่ 3) หลังจากที่ได้โหนดชนะแล้วจะต้องทำการปรับน้ำหนักเพื่อให้เข้าใกล้อินพุตมากขึ้น นอกจากการเรียนรู้ที่เกิดขึ้นที่โหนดชนะแล้ว โหนดใกล้เคียงจะเกิดการเรียนรู้ด้วย ค่าเวคเตอร์น้ำหนักของโหนดใกล้เคียงจะปรับค่าให้เข้าใกล้กับอินพุตเวคเตอร์เดียวกัน เพื่อเพิ่มโอกาสให้อินพุตใหม่ที่ใกล้เคียงกับอินพุตเดิมสามารถที่จะมีโหนดชนะใหม่ใกล้กับโหนดชนะเดิมได้ สมการในการปรับค่าน้ำหนักสามารถแสดงได้ดังสมการที่ 3.2

$$m_i(t+1) = m_i(t) + \alpha(t) \times h_{ci}(t) \times [x(t) - m_i(t)] \tag{3.2}$$

เมื่อ

- t คือรอบปัจจุบันของการเรียนรู้
- $x(t)$ คืออินพุตเวคเตอร์ปัจจุบัน
- $m_i(t)$ คือเวคเตอร์น้ำหนัก
- $\alpha(t)$ คืออัตราการเรียนรู้

โดยที่อัตราการเรียนรู้ $\alpha(t)$ จะขึ้นอยู่กับจำนวนรอบซึ่งแสดงเป็นสมการเชิงเส้นได้ดังสมการที่ 3.3

$$\alpha(t) = \alpha(0) \times \frac{T - t}{T} \tag{3.3}$$

เมื่อ

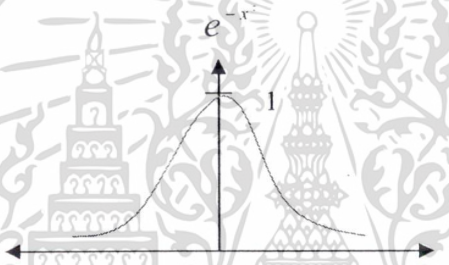
- T คือจำนวนรอบทั้งหมด
- t คือจำนวนรอบปัจจุบัน

$h_{ci}(t)$ คือฟังก์ชั่นที่ใช้ในการกำหนดน้ำหนักในการปรับค่าโหนดใกล้เคียงโดยทั่วไปแล้ว $h_{ci}(t)$ จะใช้ฟังก์ชั่นเกาส์เซียน (Gaussian) ซึ่งสามารถเขียนได้ดังสมการที่ 3.4

$$h_{ci}(t) = \exp(-\frac{\| r_c - r_i \|^2}{2\sigma^2(t)}) \tag{3.4}$$

เมื่อ

- $\|r_c - r_i\|$ คือระยะห่างของตำแหน่งของโหนด i กับโหนดชนะ c
- $\sigma(t)$ คือรัศมีของบริเวณโหนดใกล้เคียง

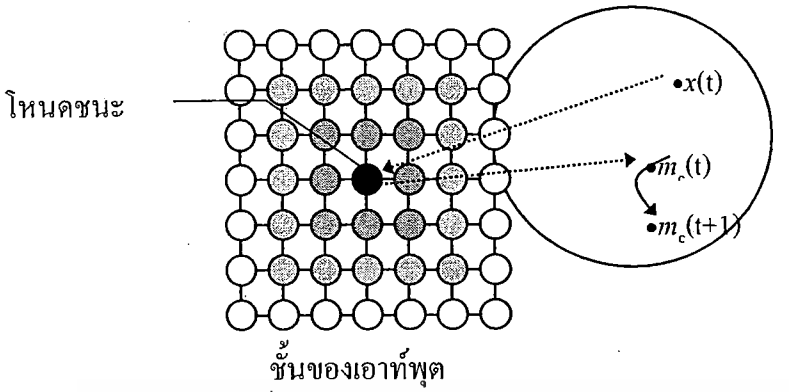


รูปที่ 3.4 แสดงกราฟของฟังก์ชั่น Gaussian ($y = e^{-x}$)

ในรูปที่ 3.4 ลักษณะของฟังก์ชั่นเกาส์เซียนคือ เมื่อค่า x คือค่าระยะห่างมีค่ามากค่าที่ส่งกลับมาจากฟังก์ชั่นจะลดลงไปเรื่อยๆ จนเข้าใกล้ศูนย์ ซึ่งสอดคล้องกับการปรับน้ำหนักของโหนดชนะและโหนดใกล้เคียง โหนดชนะจะมีค่า x เป็นศูนย์ซึ่งจะให้ค่าเกาส์เซียนฟังก์ชั่นออกมาเป็นหนึ่งซึ่งมากที่สุด โหนดที่ใกล้กับโหนดชนะจะมีการปรับค่าเวคเตอร์น้ำหนักมากกว่าโหนดที่อยู่ไกล โดยจะมีการกำหนดรัศมีของโหนดใกล้เคียง

โดยปกติรัศมีของโหนดใกล้เคียงจะค่อยๆ ลดลงตามจำนวนรอบในการเรียนรู้ t ดังสมการที่ 3.5

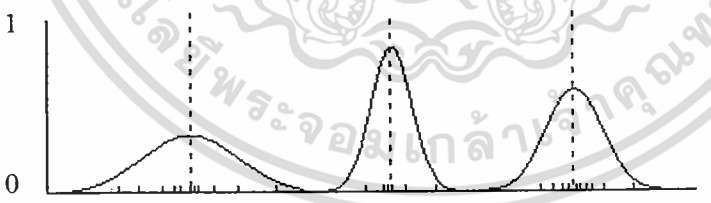
$$\sigma(t+1) = 1 + (\sigma(t) - 1) \times \frac{T - t}{T} \tag{3.5}$$



รูปที่ 3.5 แสดงโครงสร้างของ SOM ขนาด 7x7

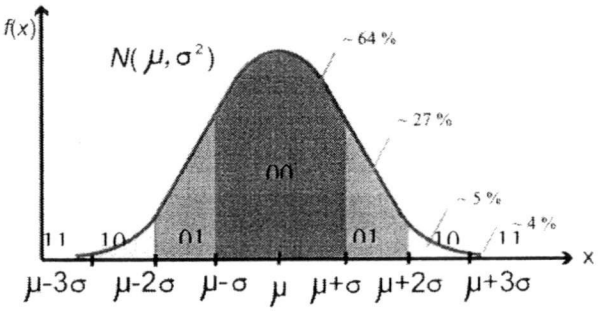
ในรูปที่ 5 แสดง SOM ขนาด 7x7 แบบสี่เหลี่ยมโหนดสีเข้มที่สุดคือโหนดชนะสำหรับอินพุตเวคเตอร์ $x(t)$ จากนั้นค่าเวคเตอร์น้ำหนักของโหนด $m_c(t)$ จะถูกปรับค่าให้เข้าใกล้กับอินพุตเวคเตอร์มากขึ้น หลังจากนั้นจะทำการปรับโหนดใกล้เคียงของโหนดชนะ โดยความเข้มสีของโหนดจะแสดงถึงปริมาณการปรับค่าของเวคเตอร์น้ำหนัก โหนดที่มีสีเข้มมากจะมีการปรับค่าเวคเตอร์น้ำหนักมากกว่าสีเข้มน้อย

กระบวนการเหล่านี้จะถูกทำซ้ำไปเรื่อยๆ จนกว่าจะสอดคล้องตามเงื่อนไขหรือจนกว่าจะครบจำนวนรอบของการเรียนรู้ เมื่อโมเดลของ SOM ผ่านการเรียนรู้เรียบร้อยแล้ว ข้อมูลต่างๆ ที่อยู่ในแต่ละโหนด (Node) สามารถบ่งบอกลักษณะการกระจายข้อมูลที่แสดงพฤติกรรมการใช้งานปรกติ โดยการคำนวณค่าเฉลี่ย (mean) และค่าเบี่ยงเบนมาตรฐาน (standard deviation) ของข้อมูลที่อยู่ในแต่ละโหนด รูปที่ 3.6 แสดงลักษณะของการกระจายข้อมูลในแต่ละโหนด



รูปที่ 3.6 การกระจายของข้อมูลในแต่ละโหนด SOM

การแบ่งระดับความผิดปรกติแต่ละโหนดในโมเดลของ SOM ที่แตกต่างกัน ได้เข้ารหัสเป็น 2 บิต แสดงค่าความแตกต่าง 4 ระดับ รูปที่ 3.7 แสดงขอบเขตแต่ละระดับความผิดปรกติ



รูปที่ 3.7 ขอบเขตแต่ละระดับความผิดปรกติ

ตารางที่ 3.1 สรุปการเข้ารหัสข้อมูลเป็น 2 บิต แบ่งเป็นระดับดังนี้ คือ '00' แทนระดับปรกติ (absolutely normal) '01' แทนระดับเกือบปรกติ (almost normal) '10' แทนระดับมีนัยผิดปรกติ (significantly abnormal) และ '11' แทนระดับผิดปรกติ

ตารางที่ 3.1 แสดงการเข้ารหัสระดับความผิดปรกติ

The level of Abnormal	2 bits - encoding	The range of data
0 - Normal	00	$(\mu\text{-}1\sigma \le x \le \mu\text{+}1\sigma)$
1- Minimal	01	$(\mu\text{-}2\sigma \le x < \mu\text{-}1\sigma \text{ or } \mu\text{+}1\sigma < x \le \mu\text{+}2\sigma)$
2 - Significant	10	$(\mu\text{-}3\sigma \le x < \mu\text{-}2\sigma \text{ or } \mu\text{+}2\sigma < x \le \mu\text{+}3\sigma)$
3 - Dangerous	11	$(x < \mu\text{-}3\sigma \text{ or } x > \mu\text{+}3\sigma)$

การแบ่งระดับความผิดปรกติ (Abnormal Quantization)

จากหัวข้อที่ผ่านมาเราจะได้โมเดลของ SOM ที่แทนแต่ละพารามิเตอร์ ในแต่ละโหนดมีค่าตัวแทนประจำบ่งบอกลักษณะการกระจายข้อมูลที่แสดงพฤติกรรมการใช้งานปรกติ โดยการทำงานในส่วนนี้ก็คือการตรวจสอบว่าอินพุตเวคเตอร์ที่เข้ามามีความแตกต่างจากพฤติกรรมการใช้งานปรกติมากน้อยอย่างไร มีลำดับขั้นตอนดังนี้

1. นำข้อมูลอินพุตเวคเตอร์เปรียบเทียบกับทุกโหนดในโมเดลของ SOM เพื่อหาโหนดชนะ
2. เข้ารหัสตามระดับความผิดปรกติ ประจำโหนดชนะ

3.4 การรวมพารามิเตอร์ (Combination parameters)

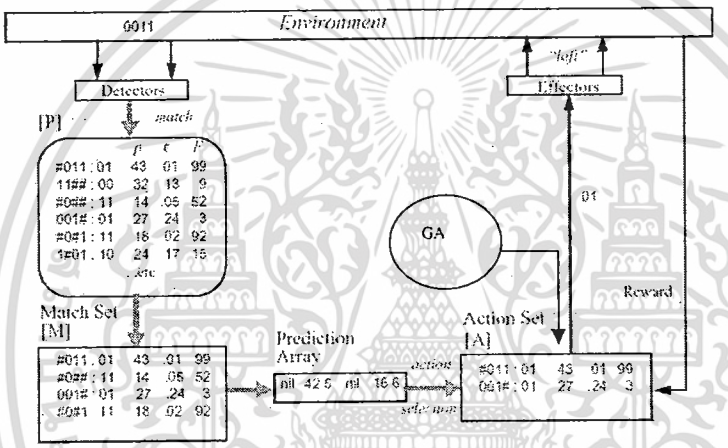
กระบวนการในหัวข้อที่ 3.3 ได้กระทำกับข้อมูลในแต่ละพารามิเตอร์ ดังนั้นกระบวนการต่อไปเป็นการรวมข้อมูลทุกๆ พารามิเตอร์ (ตัวอย่างตามตารางที่ 3.2) กำหนดให้เป็นอินพุตเวคเตอร์ สำหรับระบบการเรียนรู้ของตัวจำแนกประเภทที่จะอธิบายในหัวข้อต่อไป

ตารางที่ 3.2 ตัวอย่างการรวมพารามิเตอร์

Parameters	P1	P2	P3	P4	P5	P6	...	P29
Input string	00	01	00	10	00	11	...	00

3.5 ระบบการเรียนรู้ของตัวจำแนกประเภท (XCS)

ระบบการเรียนรู้ของตัวจำแนกประเภท หรือ Learning Classifier System (LCS) ที่ใช้ในงานวิจัยนี้คือระบบ XCS เป็นส่วนของการเรียนรู้จัดกลุ่มข้อมูลพฤติกรรมและแยกประเภทการบุกรุกออกเป็นกลุ่มๆ จากอินพุตเวคเตอร์ แบบแผนการทำงานของระบบ XCS เป็นไปตามรูปที่ 3.8 มีการทำงานดังนี้



รูปที่ 3.8 การทำงานของ XCS

ฐานข้อมูลตัวจำแนกประเภท หรือ classifier population [P] ใน XCS ประกอบไปด้วย condition⟶action จำนวน N สมาชิก สมาชิกแต่ละตัวในส่วนของ condition อยู่ในรูปแบบสตริง โดยทั่วไปจะใช้ ternary alphabet {0,1,#} และสมาชิกแต่ละตัวในส่วนของ action ในงานวิจัยนี้ประกอบด้วยคลาสที่เป็นพฤติกรรมปกติ แทนด้วย 0 และคลาสของการบุกรุกแบบต่างๆ กำหนดให้แทนด้วย 1 ถึง 4 ในแต่ละ classifier จะมีพารามิเตอร์ 4 ตัว คือ ค่า predicted payoff (p), ค่า error (ε) ที่ได้จากการ predicted payoff, ค่า fitness (F) ค่าวัดความแข็งแรงของกฎนั้น และ niche size estimate (σ) เป็นพารามิเตอร์ที่บ่งบอกการมีส่วนร่วมของ classifier

ในขณะที่มีอินพุตเวคเตอร์ส่งผ่านเข้ามาในระบบ classifier ทั้งหมดที่อยู่ในฐานข้อมูลจะถูกสแกน (scan) โดยทำการหา classifier ที่มี condition สอดคล้องกับอินพุตเวคเตอร์ ตำแหน่งของ classifier ที่สอดคล้องนั้น จะถูกกำหนดให้เป็นสมาชิกของ match set [M] ปัจจุบัน หลังจากนั้นระบบจะทำการเลือก action โดยเลือกจาก classifier ที่อยู่ใน match set และทำการกำหนดให้ classifier ที่มี action เดียวกับ action ที่เลือก เป็นสมาชิกของ action set [A] วิธีการเลือก action มี 2 วิธีการที่ใช้คือ จะทำการเลือกโดย

วิธีสุ่ม และทำการเลือกจากค่าเฉลี่ยของ payoff สูงที่สุด สลับกันไปในแต่ละรอบของการประมวลผล ตามหลักของการ explore และ exploit ในการเรียนรู้แบบ reinforcement

ค่าที่ได้รับกลับเข้ามาในระบบหลังจากที่ระบบได้ส่ง action ออกไปจากระบบ ก็คือ immediate reward (*R*) ในส่วนของ reinforcement นำค่า R มาใช้สำหรับการปรับค่า predicted payoff, error, niche size estimate และ fitness ของ classifier ที่อยู่ใน action set [A] ปัจจุบัน โดยใช้ Widrow-Hoff delta ด้วยค่า learning rate (β) โดยเริ่มจากค่า p ซึ่งจะทำการปรับดังสมการนี้

$$p = p + \beta(R - p) \tag{3.6}$$

จากนั้นจะทำการปรับค่า ε ดังนี้

$$\varepsilon = \varepsilon + \beta(|R - p| - \varepsilon) \tag{3.7}$$

ต่อไปทำการปรับค่า σ

$$\sigma_j = \sigma_j + \beta([[A]] - \sigma_j) \tag{3.8}$$

ในการปรับปรุงค่าฟิตเนส F นั้นมีรายละเอียดเพิ่มขึ้นมาเล็กน้อย เริ่มจากหาค่าความ accuracy ของ classifier คือ κ หลังจากได้ค่า κ จะนำมาหาค่า relative accuracy κ' ซึ่งทั้งสองสามารถคำนวณได้จากสมการดังนี้

$$\kappa = \begin{cases} 1, & if(\varepsilon < \varepsilon_0) \\ \alpha\left(\dfrac{\varepsilon}{\varepsilon_0}\right)^{-\nu}, & otherwise \end{cases} \tag{3.9}$$

$$\kappa' = \frac{\kappa}{\sum_{x \in [A]} \kappa_x} \tag{3.10}$$

หลังจากได้ค่า κ' แล้วก็จะทำการปรับปรุงค่าฟิตเนส F ดังนี้

$$F = F + \beta(\kappa' - F) \tag{3.11}$$

ในระบบจะมีกลไกในการค้นหาสองส่วน คือ genetic algorithm (GA) และ covering operator ในแต่ละรอบของการประมวลผล ถ้าค่าเฉลี่ย time-step (เวลาที่ classifier นั้นอยู่ใน action set[A] และ GA ทำงาน) ใน action set [A] ปัจจุบัน มากกว่า probability θ_{GA} ที่กำหนดไว้ กลไกของ GA จะทำงานโดยทำการเลือก classifier ต้นแบบ (offspring) สอง classifier เพื่อใช้สำหรับสร้าง classifier ใหม่สองตัว ด้วยการใช้ roulette wheel เลือกตามสัดส่วนตามค่า fitness หลังจากนั้น classifier ต้นแบบสองทั้งสองตัวจะผ่านกระบวนการ crossover (single point ด้วย probability χ) และกระบวนการ mutation (probability μ, เปลี่ยนเป็น wildcard (#) ด้วยอัตราส่วน $p_{\#}$) สำหรับค่า parameter ต่างๆ จะถ่ายทอดเหมือนกับ parent หรือแทนด้วยค่าเฉลี่ยถ้าระบบมีการบวนการ crossover ในการแทนที่ classifier ที่ได้จากการค้นหาใหม่ เราจะทำการเลือก classifier ที่จะถูกแทนที่ในฐานข้อมูลตัวจำแนกประเภทด้วยการใช้ roulette wheel เลือกตามสัดส่วนของขนาด estimated niche size และในกรณีที่ในรอบของการประมวลผลไม่มี classifier ใดเลยในฐานข้อมูลที่สอดคล้องกับอินพุตเวคเตอร์ กลไก covering operator จะทำงาน โดยจะสร้าง classifier ให้มี condition สอดคล้องกับอินพุตเวคเตอร์ (กำหนดให้มี wildcards ตามอัตราส่วน $p_{\#}$) พร้อมทั้งสุ่มกำหนด action ให้กับ classifier และกำหนดค่าเริ่มต้นให้กับ พารามิเตอร์ต่างๆ เหมือนกับตอนที่สร้าง population เริ่มต้น และนำ classifier ที่สร้างขึ้นใหม่ไปแทนที่ classifier ในฐานข้อมูลตัวจำแนกประเภทเหมือนกับกระบวนการแทนที่ตามที่กล่าวไปแล้ว ในกระบวนการค้นหา classifier ใหม่ด้วย GA จะไม่เกิดขึ้นในรอบของ exploit

บทที่ 4

การทดลองและผลการทดลอง

ในบทนี้จะกล่าวถึงการทดลองเพื่อทดสอบประสิทธิภาพของระบบตรวจจับการบุกรุก ด้วยระบบการเรียนรู้ของตัวจำแนกที่ผู้วิจัยได้พัฒนาขึ้น เปรียบเทียบระบบอื่นๆ โดยการทดลองประกอบไปด้วย 2 การทดลอง การทดลองที่ 1 เป็นการทดสอบกับชุดข้อมูล FTP-only dataset ที่นำมาจาก KDD Cup 1999 ซึ่งเป็นชุดข้อมูลที่นิยมใช้ในงานวิจัยที่เกี่ยวกับการตรวจจับการบุกรุก การทดลองที่ 2 เป็นการทดสอบการชุดข้อข้อมูลจาก Web Server ที่ทำหน้าที่หลักในการเชื่อมโยงเข้าสู่อินเทอร์เน็ตของมหาวิทยาลัยรามคำแหง

4.1 การทดลองที่ 1

4.1.1 ชุดข้อมูล FTP-only dataset

ชุดข้อมูลที่ใช้ในการทดลองนำมาจาก KDD Cup 1999 [7] ซึ่งเป็นชุดข้อมูลที่นิยมใช้ในงานวิจัยที่เกี่ยวกับการตรวจจับการบุกรุก ข้อมูลดังกล่าวได้จากการวิเคราะห์ระบบเครือข่ายของ US Air Force Research Lab ในช่วงเวลา 1998 ซึ่งแต่ละข้อมูลประกอบไปด้วยคุณลักษณะที่ได้จากการเชื่อมต่อระบบเครือข่าย จำนวน 41 คุณลักษณะ มีทั้งข้อมูลที่มีพฤติกรรมปกติ และพฤติกรรมการบุกรุก โดยพฤติกรรมการบุกรุกได้แบ่งออกเป็น 4 กลุ่มใหญ่ๆ ดังต่อไปนี้

1. Denial of Service เป็นลักษณะของการบุกรุกพยายามโจมตีระบบคอมพิวเตอร์ไม่ให้สามารถให้บริการต่างๆ ได้ เช่น smurf
2. Remote to Local เป็นลักษณะของการบุกรุกที่ไม่ได้เป็นยูสเซอร์ในระบบแต่พยายามเจาะเข้าไปในระบบ เช่น guess password
3. User to Root เป็นลักษณะของการบุกรุกที่พยายามเข้าสู่ระบบโดยการใช้สิทธิ์ของซุปเปอร์ยูสเซอร์ เช่น buffer overflow
4. Probe เป็นลักษณะของการบุกรุกที่พยายามตรวจสอบหาจุดอ่อนของระบบ เช่น portsweep

ในงานวิจัยของ [13], [14] ได้ทำการลดขนาดของข้อมูลลง โดยนำเฉพาะข้อมูลที่มีการเชื่อมต่อระบบเครือข่ายแบบ FTP (port 21) เท่านั้น เรียกว่า FTP-only dataset ทำให้ขนาดของข้อมูลที่ใช้สำหรับการเรียนรู้มี 798 ข้อมูล และสำหรับการทดสอบมี 837 ข้อมูล มีรายละเอียดตามตามรางที่ 4.1 นอกจากนั้น

ยังทำให้จำนวนคุณลักษณะลดลงเหลือ 29 คุณลักษณะ มีทั้งคุณลักษณะแบบสัญลักษณ์ (Symbolic) แบบตัวเลขต่อเนื่อง (continuous) และแบบตัวเลข (discrete) คุณลักษณะที่เป็นแบบสัญลักษณ์ต้องทำการปรับให้เป็นตัวเลขก่อน หลังจากนั้นในทุกๆ คุณลักษณะทำการปรับค่าให้ค่าอยู่ในช่วงตั้งแต่ 0 ถึง 1 (normalization) สำหรับประเภทของการบุกรุกแบบต่างๆ หรือ คลาสของการบุกรุกกำหนดให้แทนด้วยตัวเลข 1 ถึง 4 และ 0 สำหรับที่คลาสที่เป็นพฤติกรรมปกติ ในตารางที่ 4.1 แสดงจำนวนของข้อมูลที่กระจายตามพฤติกรรมการบุกรุกแบบต่างๆ รวมถึงข้อมูลที่เป็นพฤติกรรมปกติด้วยเราสามารถดาวโหลด FTP-only dataset ได้จากhttp://www.itee.adfa.edu.au/~alar/codes/codes.html

ตารางที่ 4.1 จำนวนข้อมูลของ FTP-only dataset ที่กระจายตามคลาสต่างๆ

Class	Training instances	Test instances
Normal	373	122
Probe	5	2
DOS	104	57
U2R	3	15
R2L	313	641
Total	798	837

4.1.2 ผลการทดลอง และการวิเคราะห์

ในการทดลองเบื้องต้นเราได้ทำการทดลองโดยนำ FTP-only dataset ตามตารางที่ 4.1 ทั้งหมด 837 ข้อมูล ส่งให้ระบบการเรียนรู้ของตัวจำแนกประเภทแบบ XCS เรียนรู้ โดยที่ไม่ได้ผ่านขั้นตอนปรับค่าในแต่ละพารามิเตอร์โดยใช้แผนภาพ SOM ตารางที่ 4.2 แสดงผลการทดลองด้วยค่าความถูกต้องเฉลี่ยที่ได้จากการทดลองทั้งหมด 30 ครั้ง

ตารางที่ 4.2 ค่าความถูกต้องของระบบ XCS

Class	XCS
Normal	95.17 ± 0.01
Probe	35.00 ± 0.26
DOS	99.67 ± 0.01
U2R	91.77 ± 0.11
R2L	35.33 ± 0.01
Total Overall	49.27 ± 0.11

จากผลการทดลองแสดงให้เห็นว่าระบบ XCS ให้ประสิทธิภาพความถูกต้องของคลาส Normal (95.17 ± 0.01) คลาส DOS (99.67 ± 0.01) และคลาส U2R (91.77 ± 0.11) สูง แต่คลาส Probe (35.00 ± 0.26) และคลาส R2L (35.33 ± 0.01) ได้ประสิทธิภาพความถูกต้องต่ำมาก จึงส่งผลให้ประสิทธิภาพความถูกต้องโดยรวมทั้งหมดอยู่ในระดับที่ต่ำคือ 49.27 ± 0.11 สอดคล้องกับงานวิจัย [1] ที่ได้ทำการศึกษาผลกระทบของระบบ XCS ที่มีต่อชุดข้อมูลที่มีขนาดข้อมูลในแต่ละคลาสไม่สมดุล (Imbalance Dataset) ข้อมูลการตรวจจับการบุกรุกก็ถือว่าเป็นข้อมูลที่อยู่ในกลุ่มนี้ ได้สรุปไว้อย่างชัดเจนว่าระบบ XCS และระบบการเรียนรู้แบบอื่นๆ ในกรณีที่ระดับของความไม่สมดุลของขนาดในแต่ละคลาสสูง ทำให้ประสิทธิภาพความถูกต้องลดลง สำหรับคลาส Probe และคลาส R2L ที่ได้ประสิทธิภาพความถูกต้องต่ำกว่าคลาสอื่นๆ เป็นเพราะสัดส่วนของจำนวนข้อมูลสำหรับการเรียนรู้ และสำหรับทดสอบไม่เหมาะสม โดยเฉพาะคลาส R2L ที่มีจำนวนข้อมูลสำหรับการเรียนรู้ 313 ข้อมูล แต่มีจำนวนข้อมูลที่เอาไว้ทดสอบระบบสูงถึง 641 ข้อมูล อาจจะมีข้อมูลจำนวนมากที่มีความแตกต่างจากข้อมูลสำหรับการเรียนรู้ จึงทำให้มีประสิทธิภาพความถูกต้องต่ำ

การทดลองต่อไปเราทำการทดลองโดยนำ FTP-only dataset ทดสอบระบบที่เรานำเสนอ ในที่นี้เรียกว่าระบบ SOMXCS ขั้นแรกระบบจะทำการปรับค่าพารามิเตอร์ด้วยการเข้ารหัสตามระดับความผิดปรกติที่วิเคราะห์ได้จากวิธีการของ SOM จากนั้นจะนำข้อมูลที่ได้ไปผ่านระบบการเรียนรู้ของตัวจำแนกประเภทแบบ XCS ค่าความถูกต้องเฉลี่ยจากการทดลองทั้งหมด 30 ครั้ง แสดงได้ในตารางที่ 4.3

จากผลการทดลองแสดงได้ชัดเจนว่าระบบ SOMXCS ได้ประสิทธิภาพดีกว่าในทุกๆ คลาส แสดงให้เห็นว่าการที่เราทำการเข้ารหัสตามระดับความผิดปรกติ ทำให้ค่าพารามิเตอร์ต่างๆ แยกแยะข้อมูลได้ดีมากขึ้น

ตารางที่ 4.3 ค่าความถูกต้องของระบบที่นำเสนอ SOMXCS

Class	The proposed system
Normal	94.09 ± 0.02
Probe	80.00 ± 0.40
DOS	98.05 ± 0.40
U2R	48.00 ± 0.09
R2L	83.15 ± 0.15
Total Overall	85.16 ± 0.11

ในส่วนสุดท้ายเราได้แสดงผลการทดลองทั้งสองรูปแบบที่กล่าวมาเปรียบเทียบกับระบบการเรียนรู้แบบอื่นๆ อีก 12 ระบบ โดยใช้ซอฟแวร์ (Software) Weka [17] ที่เป็นซอฟแวร์สำหรับทดสอบระบบ กำหนดค่าพารามิเตอร์ตามค่ามาตรฐานตามของซอฟแวร์ โดยทดสอบกับข้อมูล KDD FTP-only จำนวน 30 ครั้งเช่นกัน ค่าความถูกต้องเฉลี่ยแสดงได้ในตารางที่ 4.4

ตารางที่ 4.4 ค่าความถูกต้องของระบบการเรียนรู้หลายๆ ระบบ

ML	Normal	Probe	DOS	U2R	R2L	Overall
C4.5	98.36	100.0	89.47	33.33	0.3121	21.51
RF	99.10(0.01)	100.0(0.00)	99.59(0.01)	31.78(0.25)	15.54(0.13)	33.94(0.10)
RT	94.51(0.03)	98.33(0.09)	97.95(0.02)	37.56(0.32)	24.92(0.20)	40.44(0.15)
LMT	97.54	100.0	100.0	86.67	14.04	33.57
NB	92.62	100.0	98.25	60.0	8.89	28.32
BN	94.26	100.0	100.0	93.33	29.02	44.68
Logit	95.90	100.0	100.0	66.67	17.94	35.96
MLP	97.81(0.01)	83.33(0.37)	100.0(0.00)	44.00(0.38)	35.34(0.48)	49.12(0.01)
RBF	94.81(0.03)	100.0(0.00)	96.61(0.01)	86.67(0.00)	10.48(0.04)	30.22(0.03)
SMO	98.36	100.0	100.0	100.0	3.28	25.69
IB1	99.18	100.0	100.0	100.0	11.23	31.90
KSTAR	100.0	100.0	100.0	40.0	18.25	36.32
XCS	95.17(0.01)	35.00(0.26)	99.67(0.01)	91.77(0.11)	35.33(0.15)	49.27(0.11)
SOMXCS	94.09(0.02)	80.00(0.40)	98.05(0.40)	48.00(0.09)	**83.15(0.15)**	**85.16(0.11)**

4.2 การทดลองที่ 2

4.2.1 ข้อมูลจากสถาบันคอมพิวเตอร์ มหาวิทยาลัยรามคำแหง

ในส่วนนี้เราได้ทำการขอความอนุเคราะห์เก็บข้อมูลจากสถาบันคอมพิวเตอร์ มหาวิทยาลัยรามคำแหง โดยทำเก็บข้อมูลที่เป็นที่เครื่องเซิฟเวอร์ (Server) ตัวที่ทำหน้าที่บริการเว็บเพจ (Webpage) หลักของมหาวิทยาลัยดังรูปที่ 4.1 โดยข้อมูล โดยเก็บเฉพาะข้อมูล IP Protocol และวิธีการเก็บข้อมูลมีรายละเอียดดังนี้

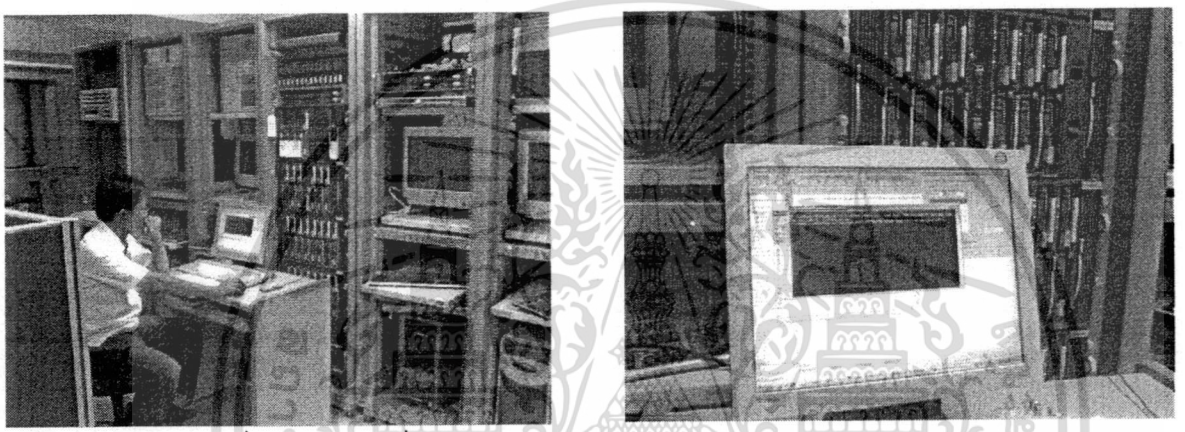
ในการเก็บข้อมูลเราจะทำการแบ่งการเก็บข้อมูลเป็น 2 ส่วน

- ข้อมูลที่มีพฤติกรรมปกติ

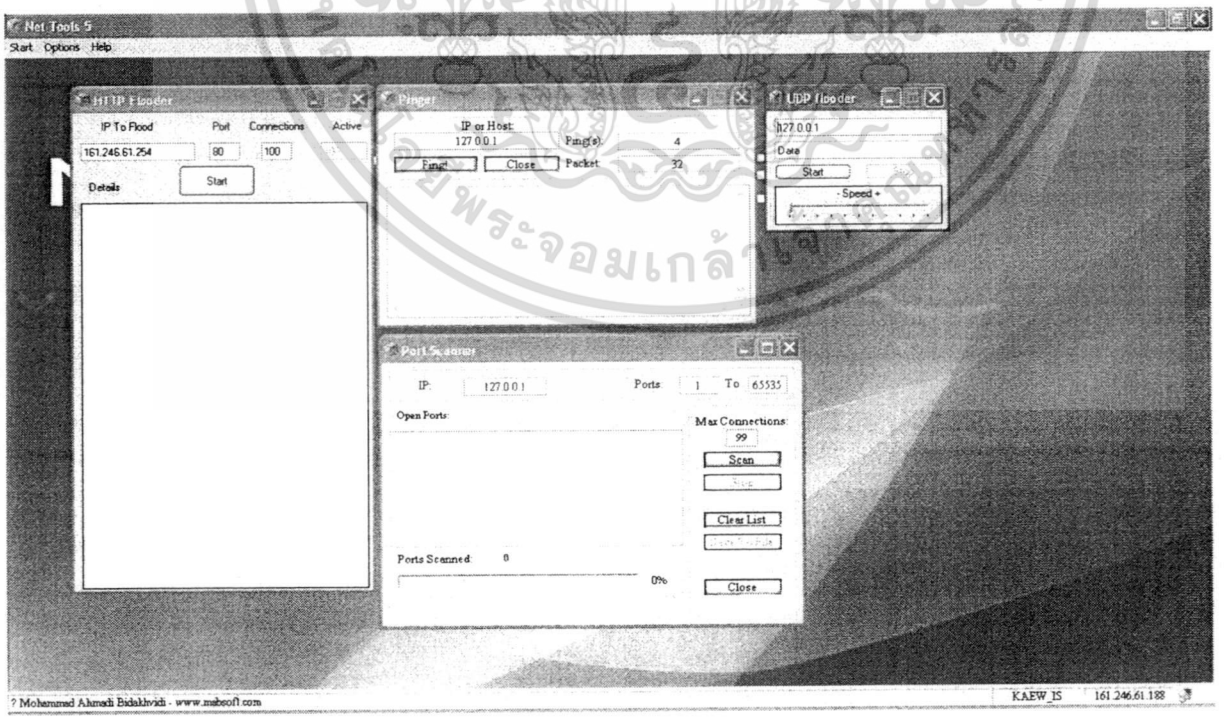
ในการเก็บข้อมูลที่มีพฤติกรรมปกติเพื่อให้ได้พฤติกรรมการใช้ที่ครอบคลุม เราได้ทำการเก็บข้อมูลทั้งหมด 5 วัน แบ่งเป็นวันราชการ 3 วัน วันหยุดราชการ 2 วัน เริ่มเก็บตั้งแต่ 10 นาฬิกา ไปจนถึง 17 นาฬิกา โดยทำการบันทึกข้อมูลชั่วโมงล่ะ 20 นาทีทุกต้นชั่วโมง

- ข้อมูลพฤติกรรมการบุกรุก

  ในการสร้างข้อมูลการบุกรุกเราอาศัยโปรแกรม Net Tool 5 ดังรูป 4.2 เพื่อเป็นสร้างเป็นรูปแบบการโจมตี โดยจะจำลองการโจมตีต่างๆ หลังจากเก็บข้อมูลปกติไปแล้ว โดยรูปแบบการโจมตีที่ทดลองมีดังแสดงในตารางที่ 4.5



รูปที่ 4.1 เว็ปเซิฟเวอร์ที่ใช้เก็บข้อมูลที่สถาบันคอมพิวเตอร์ มหาวิทยาลัยรามคำแหง



รูปที่ 4.2 โปรแกรม Net Tool 5

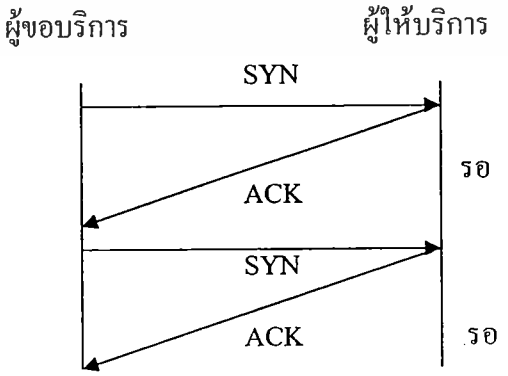
ตารางที่ 4.5 รูปแบบการโจมตีที่จำลอง

รูปแบบการโจมตี	**ฟังก์ชั่นของโปรแกรม Net Tool 5 ที่ใช้**
DOS	Http Flooder, Pinger , UDP
Probe	Fast Port Scanner

4.2.2 การโจมตีแบบ DOS (Denial Of Service)

การโจมตีแบบนี้เป็นการส่งแพ็คเกจปริมาณมากเข้าไปยังระบบเป้าหมาย อาจทำให้ระบบเป้าหมายไม่สามารถให้บริการบางอย่าง หรือไม่สามารถทำงานต่อไปได้ ซึ่งแพ็คเกจที่ส่งออกไปนี้สามารถแบ่งออกได้เป็น

- แพ็คเกจข้อมูล (Data Packets) การโจมตีวิธีนี้ทำให้ได้โดยการส่งแพ็คเกจข้อมูลปริมาณมาก เมื่อข้อมูลเข้ามาสู่เครื่องหมายก็เก็บไว้ในบัฟเฟอร์ก่อนนำมาประมวลผลอีกครั้ง ดังนั้นหากส่งแพ็คเกจเข้ามาปริมาณมาก อาจทำให้บัฟเฟอร์ของเครื่องเป้าหมายไม่เพียงพอที่จะสามารถรองรับแพ็คเกจเหล่านั้นได้ทั้งหมด ซึ่งอาจทำให้เครื่องเป้าหมายให้บริการได้ช้าลง หรือต้องหยุดการให้บริการไปเลย
- แพ็คเกจสำหรับการควบคุม (Control Packets) ตัวอย่างของการโจมตีแบบนี้ ได้แก่ การทำ SYN Flooding ปกติการเชื่อมต่อแบบ Three-way handshake เป็นไปตามลักษณะทั่วไป แต่ในการโจมตีลักษณะนี้ใช้วิธีทำให้การทำ Three-way handshake ไม่สมบูรณ์ กล่าวคือ เครื่องที่ขอบริการส่งสัญญาณ SYN ไป แต่เมื่อได้รับสัญญาณ ACK จากเครื่องที่ให้บริการแล้ว ไม่ส่งสัญญาณ SYN ตอบกลับไป ทำให้เครื่องที่ให้บริการต้องเปิดการเชื่อมต่อรอการตอบกลับ ดังรูปที่ 4.3 ซึ่งการเปิดการเชื่อมต่อรอเอาไว้นี้ต้องใช้ทรัพยากรของระบบส่วนหนึ่ง และหากมีการส่งสัญญาณในลักษณะนี้มากๆ และทรัพยากรของระบบมีไม่เพียงพอ อาจทำให้ระบบไม่สามารถให้บริการอย่างอื่น หรือให้บริการกับผู้ร้องขอรายอื่นได้



รูปที่ 4.3 แสดงการส่งแพ็คเกจแบบ SYN Flood

4.2.3 **การสแกนพอร์ต** (Port Scan)

จากความสำคัญของพอร์ตทีซีพีที่ระบุหมายเลขพอร์ตไว้ในแพคเก็จ เพื่อระบุว่าข้อมูลที่ส่งเข้ามานั้นเป็นของแอพพลิเคชั่นใด และแอพพลิเคชั่นต่างๆ ก็จะเลือกใช้พอร์ตหมายเลขต่างๆ กัน เช่น FTP ให้พอร์ตหมายเลข 21, SMTP ใช้พอร์ตหมายเลข 25 เป็นต้น เมื่อแอพพลิเคชั่นเลือกพอร์ตใดมาใช้งานแล้วก็มีหน้าที่คอยดูว่ามีการติดต่อมาที่พอร์ตของตนหรือไม่ หากมีก็ทำการตอบรับกลับไป ด้วยความสำคัญของพอร์ตนี้เอง พอร์ตจึงเป็นเป้าหมายของการบุกรุก เพื่อที่จะรู้ได้ว่ามีแอพพลิเคชั่นใดบ้างที่ทำงานอยู่บนโฮสต์ โดยทั่วไปแล้วแอพพลิเคชั่นแต่ละชนิดที่เปิดให้บริการอยู่จะใช้หมายเลขพอร์ตที่ตายตัวและรู้จักกันโดยทั่วไป ดังนั้นเมื่อทำการสแกนแล้วก็จะนำผลมาเปรียบเทียบกับมาตรฐาน

การสแกนพอร์ต เป็นการสำรวจแต่ละโฮสต์ โดยมีขอบเขตเฉพาะโฮสต์เพียงตัวเดียว เป็นการส่งสัญญาณไปสอบถามยังทุกๆ พอร์ตที่มีอยู่บนโฮสต์ ทั้ง ทีซีพี และยูดีพี เพื่อตรวจสอบว่ามีการเปิดให้บริการอะไรบ้างบนโฮสต์นั้น ซึ่งนั้นหมายถึงว่ามีแอพพลิเอชั่นประเภทใดอยู่บ้าง เทคนิคต่างๆ ที่นำมาใช้เพื่อการสแกนพอร์ตนั้นล้วนเป็นการดัดแปลงข้อกำหนดในโพรโตคอลมาใช้งานทั้งสิ้น อาจจะมีบางส่วนที่ใช้ช่องว่างที่ไม่มีกำหนดไว้ในโพรโตคอลเพื่อทำให้ได้ผลลัพธ์มาในที่สุด วิธีการที่เรานำมาทดสอบคือ TCP SYN Scan

การทำงาน TCP SYN Scan วิธีนี้ผู้สแกนจะทำการส่ง SYN แพ็คเกจ (เซตค่าแฟล็ก SYN เป็น1) เพื่อทำการติดต่อโดยตรงกับเป้าหมายโดยไม่ผ่านระบบปฏิบัติการและไม่รอผลการตอบรับของเป้าหมายกลับมา ซึ่งหากเป้าหมายทำงานอยู่ก็จะตอบกลับมาด้วย SYN ACK (เป็นเพ็กเก็ตที่ เซตค่าแฟล็ก SYN และ ACK ไว้เป็น 1) หรือหากไม่มีแอพพลิเคชั่นทำงานอยู่จะตอบกลับมาด้วย RST การสแกนแบบนี้หากตรวจสอบบนโฮสต์เป้าหมายจะพบว่ามีการขอเชื่อมต่อเข้ามา แต่ไม่สามารถเปิดการติดต่อได้สำเร็จ เทคนิคนี้บางครั้งถูกเรียกว่า half-open scanning คือไม่สามารถทำ 3-way handshake ได้ จึงไม่มีการเชื่อมต่อใดๆ เกิดขึ้นระหว่างเครื่องผู้สแกนกับเครื่องที่ถูกสแกน

4.2.4 **การดักจับข้อมูล**

ในการดักจับข้อมูลเพื่อเป็นฐานข้อมูลสำหรับโปรแกรมตรวจจับการบุกรุก เราใช้โปรแกรม TCPdump ซึ่งเป็นโปรแกรม Protocol Analyzers สำหรับระบบปฏิบัติการ Unix (สำหรับระบบปฏิบัติการ Windows ใช้โปรแกรม Windump) TCPdump ได้พัฒนามาตั้งแต่ ค.ศ. 1990 ที่ LawrenceBerkeley National Laboratory ในงานวิจัยของเราใช้โปรแกรม TCPdump สำหรับเก็บข้อมูลที่อยู่บนเครือข่าย โดย Option ของ TCPdump ที่ใช้มีดังนี้

tcpdump -i interface

เลือก interface ที่ TCPdump จะคอยรับข้อมูล ใช้สำหรับเครื่องที่มีหลาย interface (มี Lan Card หลายใบ)

tcpdump –n

TCPdump จะไม่พิมพ์ host name ลดการค้นหาชื่อจาก DNS

tcpdump –l

จะเก็บข้อมูลที่จะแสดงผลไว้ใน buffers ซึ่งสามารถนำข้อมูลเก็บลงไฟล์ได้ต่อไป

นอกจากนี้ TCPdump ยังสามารถใช้ Regular expression ได้เช่น

ether proto \ip แสดงการเชื่อมต่อของ IP เพื่อทำการส่งข้อมูล

host <host ip> แสดงเฉพาะ connection ที่ส่งจาก host หรือ ส่งถึง host นั้น

คำสั่งที่ใช้ในการดึงข้อมูลการเชื่อมต่อระหว่างเครื่องข่ายมีดังนี้

tcpdump –i <interface number> -n ether proto \ip and host <host ip>

ผลลัพธ์ที่ได้จากคำสั่ง tcpdump โดยทั่วไปมีดังนี้

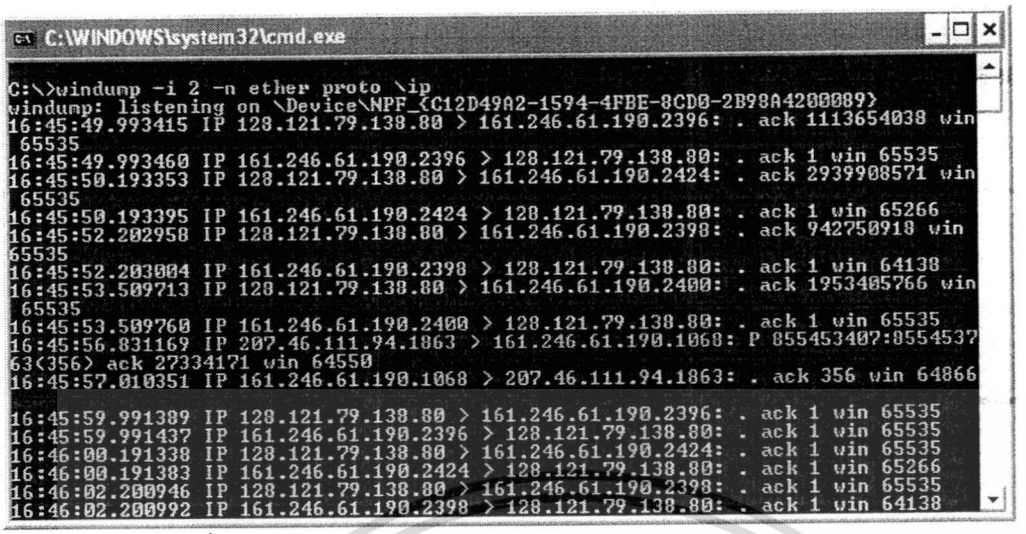
timestamp connection-protocol src > dst: flags data-seq-no. ack win urg options

คำอธิบายผลของคำสั่ง tcpdump แสดงได้ดังตารางที่ 4.6

ตารางที่ 4.6 คำอธิบายส่วนประกอบของผลลัพธ์ที่ได้จาก tcpdump

timestamp	ระบุเวลาที่ packet นั้นๆ ผ่านอุปกรณ์เครื่องข่ายที่ทำการดักจับ packet
connect-protocol	ระบุชนิดของการเชื่อมต่อเป็น IP IPX IGMP หรือ ARP
src	ระบุต้นทางที่ส่ง packet ทั้ง IP address และ port ที่ใช้ในการเชื่อมต่อ
dst	ระบุปลายทางที่ส่ง packet ทั้ง IP address และ port ที่ใช้ในการเชื่อมต่อ
flags	ระบุชนิดของ packet ว่าเป็นแบบใด เช่น flags S (SYN), F (FIN), P (PUSH) หรือ R (RST) หรือ single . (ไม่มีการระบุ flags)
data-seq-no.	ระบุเลขลำดับของ packet ในการส่งข้อมูล
ack	ระบุเลขลำดับของ packet ในการส่งข้อมูลครั้งถัดไป
win	ระบุจำนวน buffer ของอีกฝั่งที่ยังเหลือในหน่วย byte
urg	ระบุว่า packet นั้นมี urgent data ใน packet

ในการทดลองนี้เราใช้โปรแกรม windump ซึ่งเป็นเวอร์ชั่นของ tcpdump บนระบบปฏิบัติการวินโดว์สซึ่งผลของโปรแกรมสามารถแสดงได้ดังรูปที่ 4.4



รูปที่ 4.4 ตัวอย่างผลลัพธ์ที่ได้จากการ ใช้งาน windump

หลังจากบันทึกข้อมูลทั้งข้อมูลที่มีพฤติกรรมปกติและข้อมูลพฤติกรรมการบุกรุกด้วยโปรแกรม windump แล้ว เราจะนำข้อมูลที่ได้มาผ่านกระบวนการแปลงข้อมูลก่อนนำเข้าสู่ระบบ (pre-processing) เพื่อให้อยู่ในรูปแบบที่เหมาะสมซึ่งเป็นโปรแกรมที่เราพัฒนาขึ้นมาเอง

ข้อมูลที่ได้หลังจากผ่านกระบวนการแปลงแล้วจะประกอบไปด้วยคุณลักษณะดังตารางที่ 4.7

ตารางที่ 4.7 คุณลักษณะของข้อมูลหลังจากผ่านกระบวนการ pre-processing

Duration	ระบุระยะเวลาของ connection นั้นตั้งแต่เริ่มจนจบ
Protocol type	ระบุชนิดของโพรโตคอลที่ใช้ในการเชื่อมต่อเช่น TCP, UDP
Source byte	ระบุจำนวน byte ที่ต้นทางส่ง
Destination byte	ระบุจำนวน byte ที่ปลายทางส่ง
Source_Count_connection	ระบุจำนวน connection ที่เกิดขึ้นภายใน 2 วินาทีจากแหล่งที่มาเดียวกัน
Destination_Count_connection	ระบุจำนวน connection ที่เกิดขึ้นภายใน 2 วินาทีถึงปลายทางเดียวกัน
Source_service_count_connection	ระบุจำนวน connection ที่เกิดขึ้นภายใน 2 วินาทีจากแหล่งที่มาเดียวกันและ service เดียวกันเช่น Http ftp
Destination_service_count_connection	ระบุจำนวน connection ที่เกิดขึ้นภายใน 2 วินาทีถึงปลายทางเดียวกันและ service เดียวกันเช่น Http ftp

เมื่อนำข้อมูลที่ได้มาผ่านกระบวนการ pre-processing แล้วเราได้นำชุดข้อมูลมาทดสอบจำนวน 200,000 ข้อมูลโดยแบ่งข้อมูลออกดังตารางที่ 4.8

ตารางที่ 4.8 จำนวนข้อมูลที่ใช้ตามคลาสต่างๆ

Class	**จำนวนข้อมูล**
Normal	100,000
Probe	50,000
DOS	50,000

หลังจากข้อมูลผ่านกระบวนการ pre-processing แล้วก็จะนำข้อมูลที่ได้เข้าสู่กระบวนการกำหนดระดับความผิดปรกติ (Abnormal Quantization) และกระบวนการเรียนรู้ในระบบ SOMXCS ต่อไป

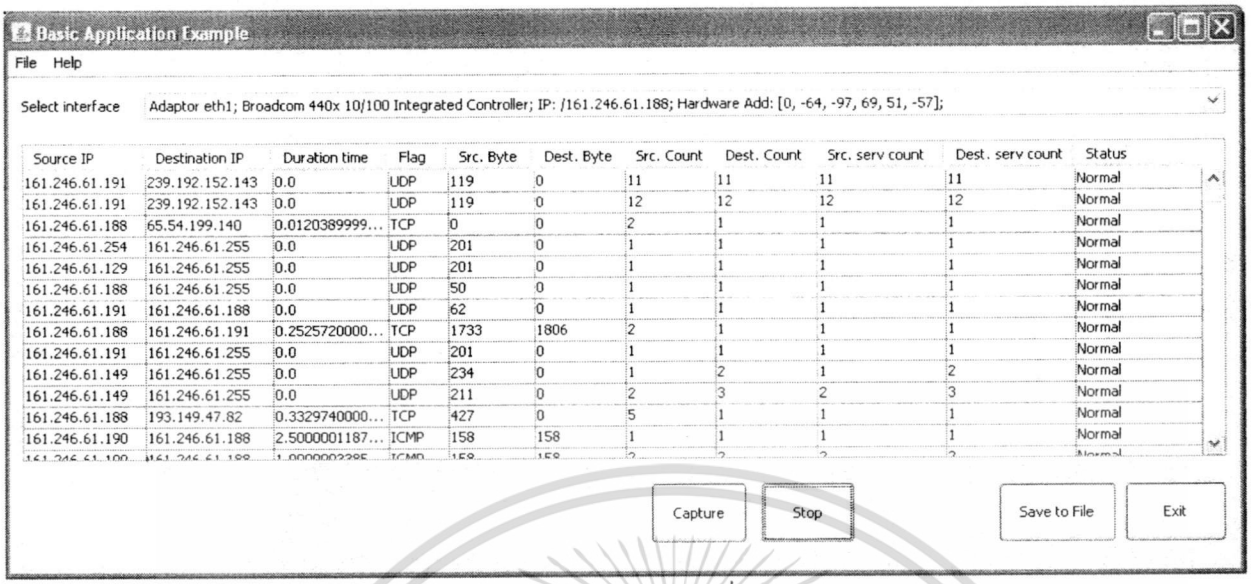
4.2.2 ผลการทดลอง และการวิเคราะห์

ในการทดลองเราได้กำหนดค่าพารามิเตอร์ต่างๆ ของทั้งระบบ SOM และระบบ XCS เหมือนการทดลองกับชุดข้อมูล FTP-only ค่าความถูกต้องที่ได้จากระบบ SOMXCS สามารถแสดงได้ดังตารางที่ 4.9

ตารางที่ 4.9 ค่าความถูกต้องของระบบ SOMXCS

Class	Training instances
Normal	97.53 ± 0.15
Probe	96.27 ± 0.08
DOS	98.10 ± 0.09
Total Overall	97.38 ± 0.12

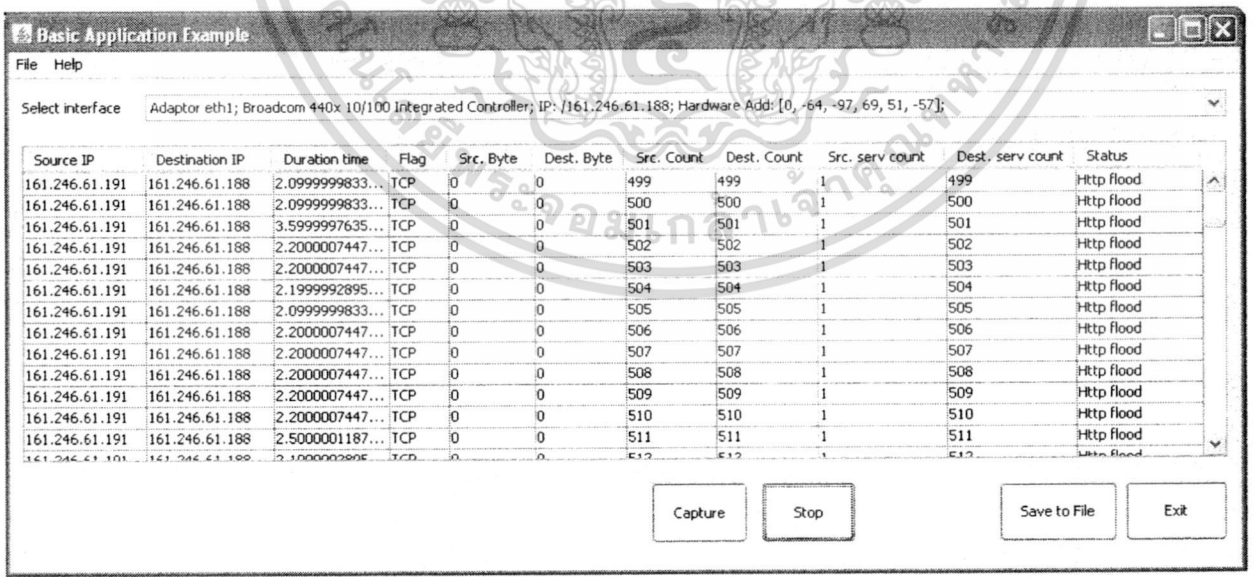
หลังจากผ่านกระบวนการเรียนรู้แล้วเราสามารถนำระบบ SOMXCS ที่ได้ไปใช้งานตรวจจับการบุกรุกได้ โดยเราได้พัฒนาอินเตอร์เฟสเบื้องต้นสำหรับดักจับข้อมูลและแจ้งเตือนให้กับผู้ดูแลทราบว่ามีข้อมูลที่มีลักษณะผิดไปจากพฤติกรรมปกติทั่วไป เพื่อให้ผู้ดูแลได้ดำเนินการต่อไปดังรูปที่ 4.5



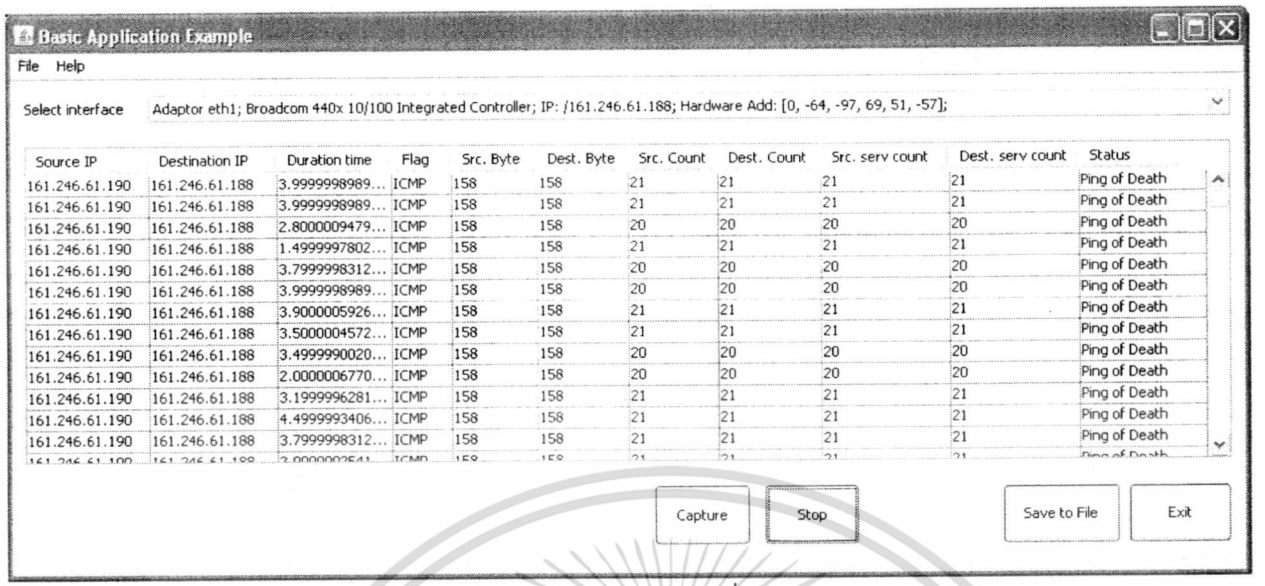
รูปที่ 4.5 โปรแกรมตรวจจับการบุกรุกขณะดับจับข้อมูล ในกรณีที่ข้อมูลเป็นปกติ

จากรูปข้างต้นโปรแกรมดักจับข้อมูลที่เข้าและออกภายในเครื่องในขณะมีข้อมูลเป็นปกติในคอลัมน์ Status แสดงให้เห็นว่าข้อมูลปัจจุบันที่โปรแกรมดักจับได้เป็นข้อมูลการใช้งานแบบปกติ (Status = Normal)

หลังจากที่เรารันโปรแกรม Net tool 5 จากอีกเครื่องเพื่อจำลองการโจมตีแบบ Denial Of Service (Http flood และ Ping of Death) รูปที่ 4.6 และ 4.7 แสดงให้เห็นได้อย่างชัดเจนว่าระบบสามารถตรวจสอบได้ (Status = Http flood และ Ping of Death ตามลำดับ)

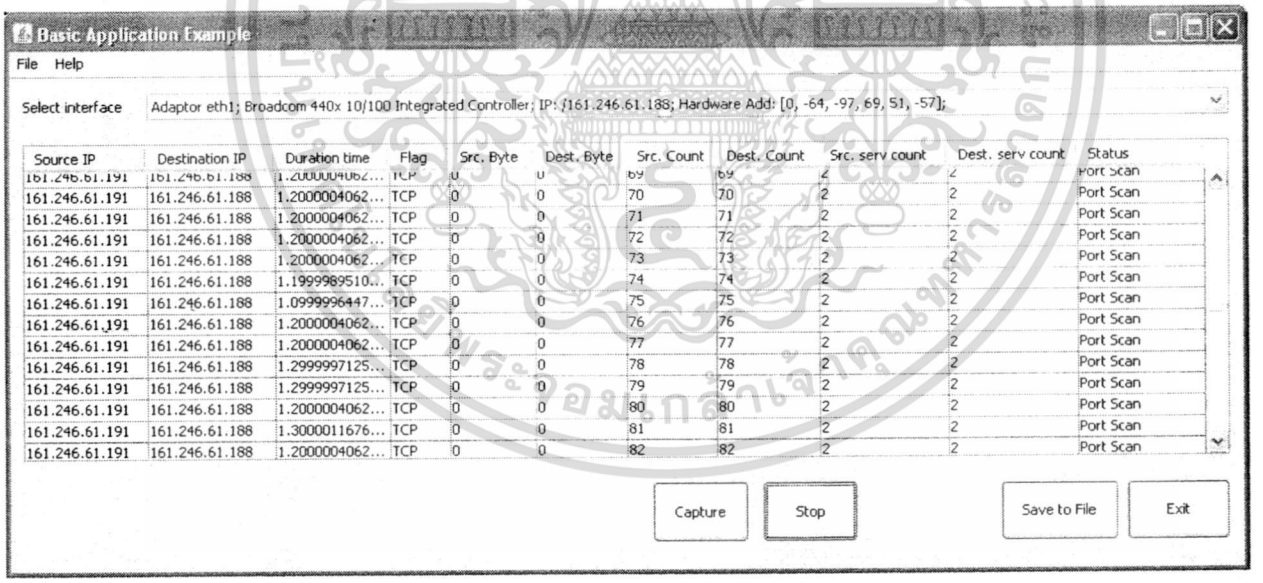


รูปที่ 4.6 โปรแกรมตรวจจับการบุกรุกขณะดับจับข้อมูล ในกรณีที่มีการโจมตีแบบ Httpflood



รูปที่ 4.7 โปรแกรมตรวจจับการบุกรุกขณะดับจับข้อมูล ในกรณีที่มีการโจมตีแบบ Ping of Death

หลังจากที่เรารันโปรแกรม Net tool 5 จากอีกเครื่องเพื่อจำลองการโจมตีแบบ Probe (Port scan) รูปที่ 4.8 แสดงให้เห็นได้อย่างชัดเจนว่าระบบสามารถตรวจสอบได้ (Status = Port scan)



รูปที่ 4.8 โปรแกรมตรวจจับการบุกรุกขณะดับจับข้อมูล ในกรณีที่มีการโจมตีแบบ Port Scan

บทที่ 5
สรุปการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

ในระบบตรวจจับการบุกรุกในบางครั้งมุ่งเน้นแต่การจดจำข้อมูลหรือรูปแบบการบุกรุกในฐานข้อมูลเท่านั้น จึงทำให้ในบางครั้งระบบไม่สามารถจำแนกข้อมูลที่ไม่เคยเรียนรู้มาก่อนได้ ประกอบกับลักษณะของข้อมูลการบุกรุกเป็นข้อมูลที่ไม่สมดุล บางคลาสมีข้อมูลที่ใช้ในการเรียนรู้น้อยมากจึงทำให้ระบบไม่สามารถจำแนกข้อมูลที่มาจากคลาสนั้นได้ ในบางระบบอาจจะทำการเรียนรู้แค่พฤติกรรมแบบปกติของผู้ใช้เพียงอย่างเดียวไม่ได้เรียนรู้ลักษณะการบุกรุก จึงทำให้ผู้ดูแลระบบไม่สามารถรู้ได้ว่าเป็นการบุกรุกแบบใด ควรจัดการกับการบุกรุกนั้นอย่างไรจึงเหมาะสม

ในงานวิจัยนี้เราได้นำเสนอระบบตรวจจับการบุกรุกที่รวมระบบทั้งสองแบบไว้ด้วยกัน เริ่มจากเรียนรู้และแบ่งกลุ่มพฤติกรรมการใช้งานปกติ โดยใช้เซลฟ์ออร์แกนไนสซิ่งแม็พเพื่อแบ่งกลุ่มข้อมูลพฤติกรรมของผู้ใช้หรือข้อมูลที่เป็นปกติ และกำหนดระดับของความผิดปกติจากพฤติกรรมของผู้ใช้ โดยอาศัยค่าเบี่ยงเบนมาตรฐาน ซึ่งจะทำให้เรารู้ได้ว่าข้อมูลที่เข้ามาเบี่ยงเบนจากพฤติกรรมผู้ใช้ปกติมากน้อยเพียงใด ข้อดีของการนำเซลฟ์ออร์แกนไนสซิ่งแม็พมาแบ่งกลุ่มข้อมูลอีกประการหนึ่งคือ พฤติกรรมการใช้งานปกติเองนั้นอาจจะมีหลายกลุ่ม เช่นจำนวนการเชื่อมต่อและการส่งข้อมูลในช่วงเช้าอาจจะมีน้อยกว่าหลังเที่ยงหรือตอนเย็น ถ้าเราไม่แยกพิจารณาจะทำให้จำแนกพฤติกรรมการใช้งานปกติผิดเพี้ยนไปได้ ซึ่งจากการอธิบายหลักการของการใช้เซลฟ์ออร์แกนไนสซิ่งแม็พ แบ่งกลุ่มในบทที่ 3 จะเห็นได้ว่าโหนดของเซลฟ์ออร์แกนไนสซิ่งแม็พจะกระจายไปยังพฤติกรรมปกติกลุ่มย่อยๆ และเป็นตัวแทนของกลุ่มย่อยนั้นเพื่อใช้ในการแบ่งระดับความผิดปกติต่อไป

หลังจากแบ่งระดับความผิดปกติแล้วเราจะใช้ระบบการเรียนรู้มาเรียนรู้ข้อมูลปกติและข้อมูลการบุกรุกที่เตรียมไว้ โดยอาศัยการแบ่งระดับที่ได้ (00,01,10,11) ในแต่ละคุณลักษณะจากขั้นตอนแรกมาสร้างเป็นชุดข้อมูลเพื่อให้ระบบการเรียนรู้จำแนกประเภทอีกที ระบบการเรียนรู้ที่เรานำมาใช้คือระบบ XCS ซึ่งเป็นระบบที่ได้รับการยอมรับกันอย่างแพร่หลายในการจำแนกประเภทข้อมูล

ในการทดลองเราได้ทำการทดลองกับ 2 ชุดข้อมูลคือ ชุดข้อมูล FTP-only และชุดข้อมูลที่เก็บได้จากสำนักคอมพิวเตอร์ มหาวิทยาลัยรามคำแหง สำหรับชุดข้อมูล FTP-only เป็นชุดข้อมูลนำมาจาก KDD Cup 1999 ซึ่งเป็นข้อมูลมาตรฐานที่นิยมนำมาทดสอบกับระบบตรวจจับการบุกรุกทั่วไป ระบบที่นำเสนอได้แสดงให้เห็นว่าสามารถให้ค่าความถูกต้องในการตรวจจับได้สูงกว่าระบบการเรียนรู้อื่นๆ อย่างเห็นได้ชัดสำหรับข้อมูล ชุดข้อมูลที่ 2 คือ ข้อมูลจากสำนักคอมพิวเตอร์ มหาวิทยาลัยรามคำแหง ซึ่งเป็นชุดข้อมูลที่

ผู้วิจัยได้ไปทำการเก็บข้อมูลจากเครื่องเซิฟเวอร์ (Server) ตัวที่ทำหน้าที่บริการเว็บเพจ (Webpage) หลักของมหาวิทยาลัยรามคำแหง ระบบแสดงให้เห็นว่ายังสามารถให้ค่าความถูกต้องได้สูงอยู่

ข้อเสนอแนะระบบที่พัฒนาอยู่ตอบสนองกับผู้ดูแลด้วยการแจ้งเตือนเท่านั้น แต่ยังไม่มีส่วนแก้ไขปัญหาเบื้องต้นให้กับระบบ ดังนั้นถ้าเกิดการโจมตีในระหว่างที่ไม่มีผู้ดูแลอยู่อาจจะทำให้ระบบเกิดความเสียหายได้ แต่การตอบสนองส่วนนี้ต้องทำการออกแบบด้วยความระมัดระวังในการออกแบบ เนื่องจากอาจจะส่งผลกระทบให้กับผู้ใช้งานปกติได้หากระบบตอบสนองผิดพลาด

ข้อเสนอแนะอีกประการเนื่องจากระบบที่ได้พัฒนาขึ้นเป็นระบบที่อาศัยการเรียนรู้จากข้อมูลและพฤติกรรมการใช้งานในระบบเครือข่ายอย่างเฉพาะเจาะจง ดังนั้นการนำเอาระบบที่พัฒนาขึ้นไปใช้กับหน่วยงานอื่นจะต้องทำการเก็บข้อมูลของหน่วยงานนั้นใหม่

เอกสารอ้างอิง

[1] A. Orriols and E. Bernado, "The Class Imbalance Problem in Learning Classifier Systems: A Preliminary Study", *Proceedings of 2005 workshops on Genetic and Evolutionary Computing*, Washington D.C., USA, pages 74-78, 2005.

[2] Axelsson, S., Lindqvist, U., Gustafson, U., Jonsson, E.: An Approach to UNIX security Logging, Technical Report IEEE Network (1996)

[3] Bull, L. (2004)(ed.) *Applications of Learning Classifier Systems*. Springer.

[4] Butz, M. and Wilson, S. (2001) An algorithmic descriptionof XCS. In Lanzi, P. L., Stolzmann, W., and S. W. Wilson (eds.),*Advances in Learning Classifier Systems. Third International Workshop (IWLCS-2000)*. Springer, pp253-272.

[5] Crosbie, M., Spafford, G.: Applying Genetic Programming to Intrusion Detection. COAST Laboratory, Purdue University,(1997) (also published in the proceeding of the Genetic Programming Conference)

[6] Frank, J.: Artificial Intelligence and Intrusion Detection: Current and future directions. In Proceedings of the 17th National Computer Security Conference, (October, 1994)

[7] Hettich S, Bay SD (1999) The UCI KDD Archive. http://www.kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[8] Holland, J.H. (1976) Adaptation. In Rosen & Snell (eds) *Progress in Theoretical Biology*, 4. Plenum

[9] Kohonen, T. (2001). Self-organizing map (3rd ed.). Berlin: Springer-Verlag.

[10] Lunt, T.F.(1990) Real-Time Intrusion Detection. Technical Report Computer Science Journal

[11] Me, L., GASSATA,: A Genetic Algorithm as an Alternative Tool for Security Audit Trail Analysis. in Proceedings of the First International Workshop on the Recent Advances in Intrusion Detection, Louvain-la-Neuve, Belgium, (September,1998) 14–16

[12] Mukherjee, B., Heberline, L.T., Levit, K.: Network Intrusion Detection. IEEE Network (1994)

[13] Shafi K, Abbass HA, Zhu W (2006) An adaptive rule-based intrusion detection architecture. In: of the 2006 RNSA security technology conference. Canberra, Australia, pp 307–319

[14] Shafi K, Abbass HA, Zhu W (2006) The role of early stopping and population size in XCS for intrusion detection. In: Proceedings of the 6th international conference on simulated evolution and learning. Lecture Notes in Computer Science, pp 50–57.

[15] Wilson, S.W. (1995) Classifier Fitness Based on Accuracy. *Evolutionary Computation* 3(2):149-76.

[16] Wilson, S. W. (2000) Get real! XCS with continuous-valued inputs. In P. L. Lanzi, W. Stolzmann and S. W. Wilson (eds.) Learning Classifier Systems. From Foundations to Applications. Springer, pp209–219.

[17] Witten IH, Frank E (2000) Data mining: practical machine learning tools and techniques with java implementations. Morgan Kaufmann

[18] Shafi K, Kovac T, Abbass HA, Zhu W (2007) Intrusion Detection with evolutionary learning classifier systems. Springer

[19] Xin Xu (2006) Adaptive Intrusion Detection Based on Machine Learning: Feature Extraction, Classifier Construction and Sequential Pattern Prediction. In: International Journal of Web Services Practices, Vol. 2, No. 1-2, pp. 49-58

[20] R. Agarwal and M. V. Joshi (2000) PNrule: A New Framework for Learning Classifier Models in Data Mining. Tech. Report, Dept. of Computer Science, University of Minnesota

[21] Liberios V. , Anton B. and Martin C.(2006) Intrusion Detection System using Self-Organizing Map. Acta Electrotechnica et Informatica No.1, Vol.6.

[22] Bernadó, M., E. (2002). Contributions to Genetic Based Classifier Systems. Barcelona, Spain, En-ginyeria i Arquitectura La Salle, Ramon Llull University. PhD.

[23] Lee, W. and S. J. Stolfo (2001). "A framework for constructing features and models for intrusion detection systems." ACM Trans. Inf. Syst. Secur. 3(4): 227-261.

[24] Mahoney, M. V. and P. K. Chan (2003). "Learning rules for anomaly detection of hostile network traffic." Data Mining, 2003. ICDM 2003. Third IEEE International Conference on: 601-604.

[25] Ertoz, L., E. Eilertson, A. Lazarevic, P. N. Tan, V. Kumar, J. Srivastava and P. Dokas (2004). "MINDS-Minnesota Intrusion Detection System." Next Generation Data Mining.

[26] Phillip William Dixon, David Corne, and Martin J. Oates. (2001) A preliminary investigation of modified XCS as a generic data mining tool. In Advances in Learning Classifier Systems: 4th International Workshop, IWLCS, pages 133–150. Springer-Verlag Berlin Heidelberg.

งานวิจัยที่ได้รับการตีพิมพ์

1. Tamee, K., Rojanavasu, P., Udomthanapong, S., Pinngern, O., "Using Self-Organizing Map with Learning Classifier System for Intrusion Detection", PRICAI 2008, Ha-noi, pp. 1071-1076, Vietnam, December 2008.
2. เกรียงศักดิ์ เตมีย์ พรเทพ โรจนวสุ ศรชัย อุดมธนาพงศ์ เอื้อน ปิ่นเงิน "ระบบตรวจจับการบุกรุกโดยอาศัยเซลฟ์ออร์แกนไนสซิ่งแม็พและระบบการเรียนรู้จำแนกประเภท", CIT'2009 Conference on Computer Information Technologies, THAILAND, January 2009. (Accepted to publish)

ภาคผนวก ก

ผลงานวิจัยที่ได้รับการตีพิมพ์

Tu-Bao Ho Zhi-Hua Zhou (Eds.)

PRICAI 2008: Trends in Artificial Intelligence

10th Pacific Rim International Conference on Artificial Intelligence
Hanoi, Vietnam, December 15-19, 2008
Proceedings

Springer

Using Self-Organizing Maps with Learning Classifier System for Intrusion Detection

Kreangsak Tamee[1], Pornthep Rojanavasu[1], Sonchai Udomthanapong[1], and Ouen Pinngern[2]

[1] Department of Computer Engineering, Faculty of Engineering, Research Center for Communication and Information Technology (ReCCIT), King Mongkut's Institute of Technology Ladkrabang, Bangkok, 10200, Thailand
[2] Department of Computer Science, Faculty of Science, Ramkhamhaeng University, Bangkok 10240, Thailand
{s6060207,s8060022,s9060705}@kmitl.ac.th, ouen@ru.ac.th

Abstract. Learning Classifier Systems (LCS) have previously been shown to have application in Intrusion Detection. This paper extends work in the area by applying the Self-Organizing Map (SOM) for creating the new input string by 2-bit encoding rely on degree of deviation of normal behaviour. The performance of systems is investigated under an FTP-only dataset. It is shown that the proposed system is able to perform significantly better than the conventional XCS, modified XCS and twelve ML algorithms.

Keywords: Intrusion Detection, LCS, Self-Organizing Map.

1 Introduction

As interconnections among computer systems grow rapidly, Intrusion Detection Systems (IDSs) is an important part of network security. A detailed survey and taxonomy of practical IDSs may be found in the literature [1]. Some are anomaly based and others are signature based. However, no detection system can catch all types of intrusions. Each model has its strengths and weaknesses in detecting different violations in networked computer systems. At the moment most of the researchers are interested in improving intrusion detection which includes artificial intelligencer [2].

Intrusion detection can be considered as a classification problem, where a bad or illegitimate activity in a computer system must be distinguished from normal activity. One of the classification methods by using both reinforcement learning and genetic algorithms to model a dataset is through the use of production system rules. In these system, known as Learning Classifier System (LCS). XCS was introduced by Wilson [3] as an enhanced version of the traditional LCS proposed by Holland [4] has demonstrated excellent performance on a number of data mining tasks [5]. Applying XCS and develop some modification on the XCS [6] to intrusion detection have proposed by using KDD Cup 99 data set [7]. Their studies showed that XCS outperform other classification algorithms, especially the modified XCS.

T.-B. Ho and Z.-H. Zhou (Eds.): PRICAI 2008, LNAI 5351, pp. 1071–1076, 2008.

The main contribution of this paper is to improve XCS's application in the intrusion detection. We investigate a Self-Organizing Map (SOM) for clustering each feature separately to obtain relatively of normal behaviour patterns. Each feature of incoming data is quantized into four levels with 2-bit encoding rely on degree of deviation of normal behaviour. After that, create the input string for XCS by bit combination of each feature, let's the XCS learn and classify the data. Our findings suggest that the accuracy as well or better that other methods.

The paper is structured as follows. Section 2 will provide the system architecture and brief description of XCS. Section 3 describes the 1999 KDD data set. The experimental is explained in section 4. Section 5 provides the conclusion and future works.

2 System Architecture

The proposed system architecture consists of three main parts: abnormal quantization, combination parameters and XCS as shown in figure 1. The idea is, firstly, we calculate the level of abnormal behavior in each feature from SOM which is trained by normal behavior. Secondly, create the input string for XCS by bit combination of each feature. Lastly, let's the XCS learn and classify the data. The detail of each part is explained in the following sections.

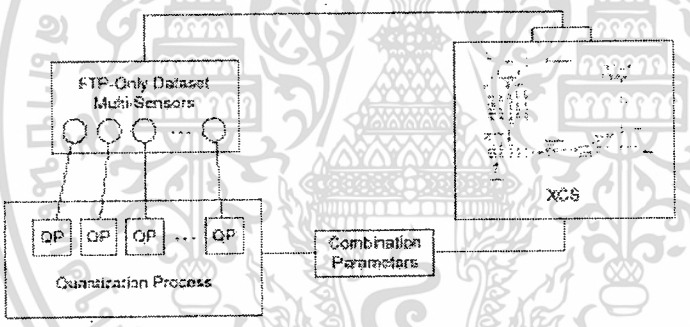


Fig. 1. The proposed IDS architecture

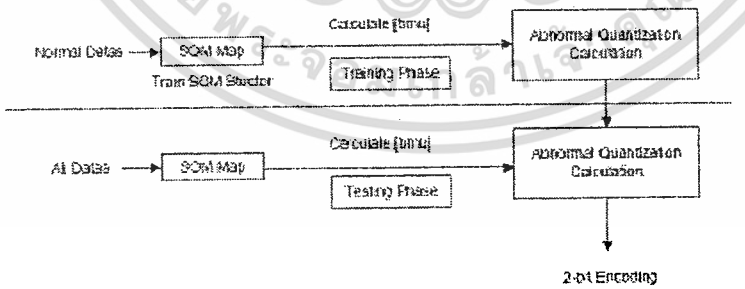


Fig. 2. Flow of data in SOM encoding phase

2.1 Quantization Process

Each module based on building model of normal data. The quantization error is calculated from the normal model in observed data. If the quantization error is grater than the threshold then the observed data is reported as an attack. In our approach, we hypothesize that each feature of data is independent and has one more group of normal behavior on each feature. Therefore, in training phase we create SOM model for clustering each feature separately. After that, in testing phase, we use these models quantize each feature of incoming data into four levels with 2-bit encoding. The overall quantization process can show you as figure 2.

2.1.1 Abnormal Quantization Calculation

In our architecture, we use 29 SOM models for cluster feature of data. Each SOM has size 5×1and trains only normal behavior data.

The learning process of an SOM [8] can be thought in terms of continuous adaptation of nodes for input vectors. We summarize the learning process as a repetition of following basic tasks:

1. The input vector $x(t)$ is fed into every node in the map to identify the output vector's winning node. It is common to use Euclidean distance as the basis to measure similarities.
2. The weight of the winning node c is tuned by the difference between the input vector and the weight vector. Not only the winning node is learning but also its neighborhood nodes are learning as well. After finish the learning process of SOM, we calculate the mean and the standard deviation of each node based on the clustered data. The map of SOM can be represented several normal distribution curves as in figure 3.

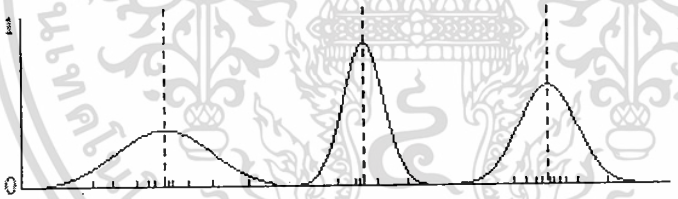


Fig. 3. The normal distribution curve of the SOM

We quantize each cluster (normal distribution curve) into four levels by using 2-bit encoding. Figure 4 shows how to break up the total area under the curve for encoding the input data. Table 1 shows the summarization of 2-bit encoding data. '00' means the data is absolutely normal, '01' means the data is almost normal, '10' means the data is significantly abnormal and '11' means the data is absolutely dangerous to the system.

2.1.2 Abnormal Quantization

Once the SOM is trained and abnormal quantization calculation is obtained, the incoming data can be quantized as follow:

1. For every incoming input, find the winning node of the SOM.
2. Encoding to 2-bit by using mean and standard deviation of winning node.

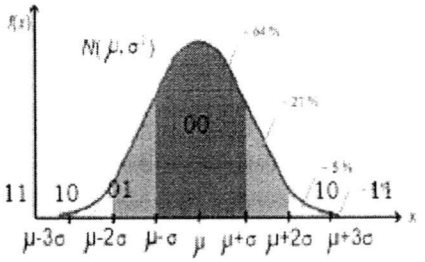


Fig. 4. The total area under the normal distribution curve

Table 1. The 2-bit encoding data

The level of Abnormal	2 bit - encoding	The range of data
0 - Normal	00	$(\mu\text{-}1\sigma \le x \le \mu\text{+}1\sigma)$
1- Minimal	01	$(\mu\text{-}2\sigma \le x < \mu\text{-}1\sigma \text{ or } \mu\text{+}1\sigma < x \le \mu\text{+}2\sigma)$
2 - Significant	10	$(\mu\text{-}3\sigma \le x \le \mu\text{-}2\sigma \text{ or } \mu\text{+}2\sigma < x \le \mu\text{+}3\sigma)$
3 - Dangerous	11	$(x < \mu\text{-}3\sigma \text{ or } x > \mu\text{+}3\sigma)$

2.2 Combination Parameters

After the data is encoded every features, we can create the input for XCS by combining every feature into an input string as shown in table 2.

Table 2. Combination of parameters as input for XCS

Parameters	P1	P2	P3	P4	P5	P6	...	P29
Input string	00	01	00	10	00	11	...	00

2.3 XCS

XCS is a Learning Classifier System without internal memory, where the rule-based consists of a number (N) of condition/action rules. The condition is the ternary alphabet: {0, 1, #} and the action is coded as an integer. Associated with each rule are prediction payoff (p), prediction error (ε), fitness parameters (F) and niche size estimate (σ).

On receipt of an input data, the rule-based is scanned, and any rule whose condition matches the message at each position is tagged as a member of the current match set [M]. An action is then chosen from those proposed by the members of the match set and all rules proposing the selected action form an action set [A]. A version of XCS's explore/exploit action selection scheme will be used here. That is, on one cycle an action is chosen at random and on the following the action with highest total payoff is chosen deterministically.

Once the action is selected, the environment returns a reward (R), which is used to update the prediction payoff (p), the prediction error (ε), the niche size estimate and fitness parameters (F) of each member of the current [A] using the Widrow-Hoff delta rule with learning rate β.

XCS employs two discovery mechanisms, a niche genetic algorithm (GA) and a covering operator. XCS uses a time-based mechanism under which each rule maintains a time-stamp of the last system cycle upon which it was consider by the GA. The GA is applied within the [A] when the average number of system cycles since the last GA in the set is over a threshold θ_{GA}. The reader is referred to [9] for a full algorithmic description of XCS.

3 FTP-Only Dataset

We use the 1999 KDD cup intrusion detection dataset [7] has been extensively used in ID research. Four categories of simulated attacks were injected among the normal traffic. In [6] extracted a small subset consisting of FTP control records (port 21 only) from these training and test datasets, which amounted to 798 training instances and 837 test instances. They call it the FTP-only dataset. The datasets are available from the ALAR LAB website (http://www.itee.adfa.edu.au/~alar/).

4 Experiments and Discussion

In our experiment, we test our system compare with the XCSR (XCS with real-number) and XCS(FC)0.3 [6] which is the extension of XCSR for IDS. We experiment with 5×1 SOM. The parameter setting of XCS similar to use in Wilson [3] as follows: β=0.2; α=0.1; ε_0=10; ν=5; χ=0.8, μ=0.04. θ_{del}=25; θ_{GA}=25; δ=0.1 and the maximum population size is N=2000.

Table 5. The mean and standard deviation of the accuracy of various ML algorithm with XCS, XCS(FC)0.3 and The proposed system

ML	Normal	Probe	DOS	U2R	R2L	Overall
C4.5	98.36	100.0	89.47	33.33	0.3121	21.51
RF	99.10(0.01)	100.0(0.00)	99.59(0.01)	31.78(0.25)	15.54(0.13)	33.94(0.10)
RT	94.51(0.03)	98.33(0.09)	97.95(0.02)	37.56(0.32)	24.92(0.20)	40.44(0.15)
LMT	97.54	100.0	100.0	86.67	14.04	33.57
NB	92.62	100.0	98.25	60.0	8.89	28.32
BN	94.26	100.0	100.0	93.33	29.02	44.68
Logit	95.90	100.0	100.0	66.67	17.94	35.96
MLP	97.81(0.01)	83.33(0.37)	100.0(0.00)	44.00(0.38)	35.34(0.48)	49.12(0.01)
RBF	94.81(0.03)	100.0(0.00)	96.61(0.01)	86.67(0.00)	10.48(0.04)	30.22(0.03)
SMO	98.36	100.0	100.0	100.0	3.28	25.69
IB1	99.18	100.0	100.0	100.0	11.23	31.90
KSTAR	100.0	100.0	100.0	40.0	18.25	36.32
XCS	95.17(0.01)	35.00(0.26)	99.67(0.01)	91.77(0.11)	35.33(0.15)	49.27(0.11)
XCS (FC)0.3	94.37(0.01)	38.33(0.31)	94.53(0.03)	74.87(0.15)	59.13(0.14)	67.03(0.11)
The Proposed system	**94.09(0.02)**	**80.00(0.40)**	**98.05(0.40)**	**48.00(0.09)**	**83.15(0.15)**	**85.16(0.11)**

We then challenge the proposed system with other twelve ML algorithms. The average accuracy of twelve ML algorithms reported in [6] for FTP-only data set. Table 5 illustrates the result of twelve ML algorithm, two versions of XCS and the proposed system. All twelve algorithms perform better on Probe. Consistent with the previous research [6], the result showed XCS perform poor when dealing with the imbalanced problem. In other influence class R2L, however, the proposed system performs well compared to twelve algorithms and the two versions of XCS.

5 Conclusion and Future Works

In this paper, we introduced the use of SOM improve the performance of XCS for intrusion detection. The proposed system uses SOM as input quantizer by training SOM separately for each feature on normal behavior data and uses their mean and standard deviation quantizes the input.

The proposed system was tested on FTP-only dataset which is a sub-set of 1999 KDD cup intrusion detection dataset. It is shown that the proposed system can improve the accuracy of XCS and also use less than memory in term of macro classifiers. We also compare with twelve ML algorithms and found that the proposed system outperform the accuracy obtained by twelve ML algorithms. We are currently our proposed system to the other intrusion detection domain.

References

1. Debar, H., Dacier, M., Wepspi, A.: A revised taxonomy for intrusion 590 detection systems. Technical report, Computer Science/Ma- 591 thematics (1999)
2. Frank, J.: Artificial Intelligence and Intrusion Detection: Current and future directions. In: Proceedings of the 17th National Computer Security Conference (October 1994)
3. Wilson, S.W.: Classifier Fitness Based on Accuracy. Evolutionary Computation 3(2), 149–176 (1995)
4. Holland, J.H.: Adaptation. In: Rosen, Snell (eds.) Progress in Theoretical Biology (1976)
5. Bull, L. (ed.): Applications of Learning Classifier Systems. Springer, Heidelberg (2004)
6. Shafi, K., Kovacs, T., Abbass, H.A., Zhu, W.: Intrusion detection with evolutionary learning classifier systems. Natural Computing (December 2007)
7. Hettich, S., Bay, S.D.: The UCI KDD Archive (1999). `http://www.kdd.ics.uci.edu`
8. Kohonen, T.: Self-organization and associative memory. Springer, New York (1989)
9. Butz, M., Wilson, S.: An algorithmic descriptionof XCS. In: Lanzi, P.L., Stolzmann, W., Wilson, S.W. (eds.) IWLCS 2000. LNCS, vol. 1996, pp. 253–272. Springer, Heidelberg (2001)