สถาปัตยกรรมมาตรฐานโพรไฟล์สำหรับตรวจจับความผิดปกติ

Standard Profile Architecture for Anomaly Detection

ปัทมา แสงหมี

Patthama Sangmee

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา
วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
มหาวิทยาลัยสงขลานครินทร์

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Science
Prince of Songkla University
2555

ลิขสิทธิ์ของมหาวิทยาลัยสงขลานครินทร์

ชื่อวิทยานิพนธ์ สถาปัตยกรรมมาตรฐานโพรไฟล์สำหรับตรวจจับความผิดปกติ
ผู้เขียน นางสาวปัทมา แสงหมี
สาขาวิชา วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

.....
(ผู้ช่วยศาสตราจารย์ ดร.นิษฐิดา เอลซ์)

คณะกรรมการสอบ

.....ประธานกรรมการ
(ดร.กุลชาติ มีทรัพย์หลาก)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.นิษฐิดา เอลซ์)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.ลัดดา ปรีชาวีรกุล)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

.....
(ศาสตราจารย์ ดร.อมรรัตน์ พงศ์ดารา)
คณบดีบัณฑิตวิทยาลัย

ขอรับรองว่า ผลงานวิจัยนี้เป็นผลงานมาจากการศึกษาของนักศึกษาเอง และขอขอบพระคุณผู้ที่มีส่วนเกี่ยวข้องทุกท่านไว้ ณ ที่นี้

ลงชื่อ_____

(ผศ.ดร.นิษฐิดา เอลซ์)

อาจารย์ที่ปรึกษาวิทยานิพนธ์

ลงชื่อ_____

(นางสาวปัทมา แสงหมี)

นักศึกษา

ข้าพเจ้าขอรับรองว่า ผลงานวิจัยนี้ไม่เคยเป็นส่วนหนึ่งในการอนุมัติปริญญาในระดับใดมาก่อน และไม่ได้ถูกใช้ในการยื่นขออนุมัติปริญญาในขณะนี้

ลงชื่อ_____

(นางสาวปัทมา แสงหมี)

นักศึกษา

ชื่อวิทยานิพนธ์ สถาปัตยกรรมมาตรฐานโพรไฟล์สำหรับตรวจจับความผิดปกติ

ผู้เขียน นางสาวปัทมา แสงหมี

สาขาวิชา วิทยาการคอมพิวเตอร์

ปีการศึกษา 2554

บทคัดย่อ

อัตราการบุกรุกระบบเครือข่ายคอมพิวเตอร์เพิ่มขึ้นตามอัตราการเติบโตของระบบเครือข่ายอินเทอร์เน็ต การโจมตีที่เกิดขึ้นบนเครือข่ายมาจากการใช้ประโยชน์จากระบบที่ยังไม่ได้แก้ไขหรือผู้บุกรุกได้รับสิทธิในการเข้าถึงระบบโดยใช้ประโยชน์จากช่องโหว่ของเครือข่ายและระบบปฏิบัติการ ถึงแม้จะมีการนำเอาแนวทางการป้องกันหลากหลาย เช่น ไฟร์วอลล์ โปรแกรมป้องกันไวรัส หรือ การเข้ารหัสข้อมูล มาใช้เพื่อป้องกันทรัพยากรเครือข่าย แต่ก็ยังไม่สามารถป้องกันการโจมตีทุกชนิดได้ เนื่องจากมีการปรับเปลี่ยนและพัฒนาเทคนิคของการโจมตีให้มีความซับซ้อนมากขึ้น หนึ่งในวิธีการที่นำมาช่วยในเรื่องของการป้องกันการโจมตีหรือการบุกรุกที่เกิดขึ้นบนเครือข่ายคือ การติดตั้งระบบตรวจจับการบุกรุก

วิธีการหนึ่งของการตรวจจับการบุกรุกที่เหมาะกับการบุกรุกชนิดใหม่ ๆ คือ วิธีการตรวจจับความผิดปกติของเหตุการณ์ต่าง ๆ ที่เกิดขึ้น โดยเปรียบเทียบกับค่าตัวแทนของความปกติ เช่นปริมาณข้อมูลในเครือข่าย ซึ่งตัวแทนความปกตินี้สามารถสร้างได้จากการเรียนรู้ค่าข้อมูลนั้น ๆ ที่เกิดขึ้นในอดีต และถูกกำหนดให้เป็นข้อมูลโพรไฟล์ของเครือข่ายเป้าหมาย อย่างไรก็ตามโพรไฟล์ของแต่ละเครือข่ายไม่จำเป็นต้องเหมือนกัน และเมื่อนำมาใช้สำหรับการบุกรุกรูปแบบต่างกันก็อาจจะใช้ค่าข้อมูลที่แตกต่างกันได้ ดังนั้นการได้มาซึ่งข้อมูลที่เหมาะสมสำหรับการตรวจจับการบุกรุกต่าง ๆ จะเป็นตัวชี้วัดถึงความสำเร็จของระบบตรวจจับการบุกรุกใด ๆ

วิทยานิพนธ์นี้ได้นำเสนอวิธีการศึกษาเพื่อให้ได้โพรไฟล์ที่เหมาะสมสำหรับการตรวจจับการบุกรุกหลากหลาย และนำเสนอวิธีการได้มาซึ่งข้อมูลที่ต้องใช้สำหรับ โพรไฟล์ต่าง ๆ ในรูปแบบของออบเจ็กต์ของ SNMP MIB เพื่อความสะดวกในการนำไปใช้สำหรับการสร้างโพรไฟล์ของแต่ละองค์กรในภายหน้า ทั้งนี้เพื่อตรวจสอบความเหมาะสมของข้อมูลที่นำเสนอและประสิทธิภาพของรูปแบบใหม่ของโพรไฟล์นี้ ผู้วิจัยได้นำเสนอระบบตรวจจับการบุกรุกเพื่อทดสอบการตรวจจับการบุกรุกในรูปแบบต่าง ๆ พบว่าผลของการตรวจจับได้ผลเป็นที่น่าพอใจ โดยมีความถูกต้องในการตรวจจับ 100 % อัตราความผิดพลาดในการตรวจจับเชิงบวก 3.47% และ อัตราความผิดพลาดในการตรวจจับเชิงลบ 0 %

Thesis Title	Standard Profile Architecture for Anomaly Detection
Author	Miss.Pattahama Sangmee
Major Program	Computer Science
Academic Year	2011

ABSTRACT

The quantity of computer network attacks is almost proportional to the increasing usages of the Internet. Most of these attacks have made use of vulnerable un-patched network elements which some are known as zero-day attacks. Although, several protection mechanisms such as firewalls, antivirus protection or encryption method are used to protect the resources, many systems are still vulnerable to new technique penetrations. Thus, intrusion detection system is also required as one of network defenses.

Anomaly detection technique is one of the intrusion detection methods suitable for a new, or an unknown pattern, attack. It can reveal suspected behavior or usage deviated from normal usage which normally can be set in a profile. Therefore, if a profile of each network or a target system is properly established, the majority of suspicious behavior could be detected. The problem is how to set up an appropriate profile which symbolizes network characteristic so that normal behavior is well represented.

This thesis studied intrusion detection profile measurements and presented several necessarily data in terms of Management Information Base (MIB) objects. These objects are new to the existing MIB-II objects, thus they are implemented as experimental objects in a MIB tree structure. To demonstrate the efficiency of these new objects regarding to the anomaly detection mechanism, several profiles were created in according to some known attacking methods and an intrusion detection system was also created to test these profiles. The result of this test shows that 100 % of tested attempts can be detected with 3.47% of false positive while the false negative is 0%

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

รายการตาราง

ตาราง **หน้า**

รายการภาพประกอบ

ภาพประกอบ **หน้า**

รายการภาพประกอบ (ต่อ)

ภาพประกอบ **หน้า**

รายการภาพประกอบ (ต่อ)

ภาพประกอบ **หน้า**

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของการวิจัย

ความปลอดภัยของระบบคอมพิวเตอร์และเครือข่ายเป็นสิ่งที่มีความจำเป็นและสำคัญมากสำหรับข้อมูลสารสนเทศของแต่ละหน่วยงานหรือองค์กร ซึ่งในปัจจุบันระบบคอมพิวเตอร์และเครือข่ายมีความเสี่ยงต่อภัยคุกคามตลอดเวลา จากรายงานของ Internet World Stats ปี ค.ศ. 2011 เรื่องการใช้งานและการบุกรุกบนอินเทอร์เน็ตพบว่า มีการบุกรุกบนอินเทอร์เน็ตเพิ่มขึ้นจากปี ค.ศ. 2010 จนถึงปี ค.ศ. 2011 เฉลี่ยแล้ว 32.7% (Internet World Stats, 2011) ส่วนใหญ่แล้วการโจมตีที่เกิดขึ้นมาจากโปรแกรมพวกโทรจัน ไวรัสคอมพิวเตอร์ หนอนอินเทอร์เน็ต หรือการใช้ประโยชน์จากช่องโหว่ของระบบคอมพิวเตอร์และเครือข่าย หรือจุดอ่อนของโพรโทคอลที่ใช้ เป้าหมายการโจมตีอาจจะแตกต่างกัน เช่น เพื่อเปิดเผยความลับของข้อมูล ทำลายข้อมูล หรือสร้างความเสียหายให้กับองค์กรหรือหน่วยงานต่างๆ (Qayyum *et al.*, 2005)

แนวทางในการแก้ไขปัญหาที่เกิดขึ้นนั้นมีด้วยกันหลายวิธี เช่น การติดตั้งไฟล์วอลล์ (Firewall) การติดตั้งโปรแกรมป้องกันไวรัส (Antivirus Software) หรือการป้องกันข้อมูลโดยการเข้ารหัสข้อมูล (Encryption) ฯลฯ แต่ก็ยังไม่สามารถป้องกันการโจมตีทุกชนิดได้ เนื่องจากมีการปรับเปลี่ยนและพัฒนาเทคนิคการโจมตีให้มีความซับซ้อนมากขึ้น ระบบตรวจจับการบุกรุก (Intrusion Detection System: IDS) จึงเป็นอีกทางเลือกหนึ่งที่ใช้ในการตรวจจับการกระทำที่อาจจะเป็นการโจมตีระบบคอมพิวเตอร์และเครือข่าย ปัจจัยสำคัญในการสร้างระบบตรวจจับการบุกรุกคือการนำข้อมูลมาใช้สำหรับวิเคราะห์พฤติกรรมหรือการกระทำต่างๆ ทั้งจากของผู้ใช้ทั่วไปหรือการทำงานของโปรแกรม ซึ่งข้อมูลเหล่านี้อาจจะได้มาจากค่าสถิติต่างๆ ที่บ่งบอกคุณสมบัติของระบบคอมพิวเตอร์และเครือข่ายในขณะนั้นๆ เช่น ปริมาณการใช้งาน CPU การใช้งานหน่วยความจำของระบบคอมพิวเตอร์ หรือปริมาณข้อมูลการใช้งานเครือข่ายในขณะใดขณะหนึ่ง เป็นต้น โดยทั่วไประบบตรวจจับการบุกรุกนั้นมีด้วยกันสองประเภทคือ Misuse Detection และ Anomaly Detection โดยที่ Misuse Detection จะเกี่ยวข้องกับการจับคู่รูปแบบ (Matching Pattern) ที่ได้มีการระบุว่าเป็นความผิดปกติไว้ก่อนแล้ว เมื่อมีเหตุการณ์เกิดขึ้นตรงกับรูปแบบที่ได้ระบุไว้จะถือว่าเป็นความผิดปกติ แต่สำหรับแนวทางของ Anomaly Detection จะเกี่ยวข้องกับการตรวจหาสิ่งผิดปกติ

ที่ต้องเก็บบันทึกการทำงานหรือเหตุการณ์ที่แทนถึงความปกติเอาไว้ก่อน หรือเรียกว่าโพรไฟล์ เพื่อใช้สำหรับเป็นตัวเปรียบเทียบกับเหตุการณ์ปัจจุบัน ซึ่งหากเกิดเหตุการณ์หรือมีการทำงานที่เบี่ยงเบนไปจากโพรไฟล์ที่สร้างขึ้นจะถือว่าเป็นความผิดปกติ โดยโพรไฟล์ที่สร้างขึ้นนั้นสร้างมาเพื่อวัตถุประสงค์ในการตรวจจับความผิดปกติชนิดใดชนิดหนึ่งเท่านั้น เช่น เพื่อวิเคราะห์การจราจรบนเครือข่ายหรือดูเสถียรภาพของการจราจรบนเครือข่ายในแง่มุมต่างๆ (Kim *et al.*, 2006) โพรไฟล์มีลักษณะเฉพาะและขึ้นอยู่กับองค์กรหรือเครือข่าย เนื่องจากมีการให้บริการหรือการใช้งานที่แตกต่างกัน ทำให้ไม่สามารถนำโพรไฟล์ขององค์กรหนึ่งไปใช้ได้กับอีกองค์กรหนึ่ง (Salem and Karim, 2008)

จากการศึกษาทั้งรูปแบบการโจมตีและการตรวจจับการบุกรุกต่างๆ (Durgin and Zhang,2005);(Hussain *et al.*, 2003); (Ghali and Masri, 2009) ในอดีตพบว่า ถึงแม้จะมีระบบตรวจจับการบุกรุกหลายชนิดแต่ก็มักจะใช้ข้อมูลในการตรวจจับชนิดเดียวกันหรือคล้ายคลึงกัน เช่น การตรวจจับการโจมตีประเภท Flooding ข้อมูลที่ใช้คือ ปริมาณแพ็กเก็ต TCP หรือปริมาณแพ็กเก็ต ICMP เป็นต้น อย่างไรก็ตามระบบตรวจจับการบุกรุกดังกล่าวต้องทำการวิเคราะห์ข้อมูลและวิธีการได้มาซึ่งข้อมูลที่ต้องการและสร้างส่วนจัดเก็บข้อมูลเองเสมอ ถือได้ว่าเป็นงานที่ซ้ำซ้อน และขณะนี้ยังไม่มีมาตรฐานกลางในการนำเสนอชนิดข้อมูลและรูปแบบข้อมูลที่ควรจัดเก็บเพื่อให้สามารถนำไปใช้ในการวิเคราะห์การบุกรุกได้

การจัดการบริหารเครือข่ายถือเป็นส่วนหนึ่งของการจัดการความปลอดภัยซึ่งมีเครื่องมือที่สนับสนุนคือ Simple Network Management Protocol (SNMP) และข้อมูลหรืออ็อบเจกต์ (Object) ต่างๆ ในรูปแบบของ Management Information Base (MIB) อยู่แล้ว อ็อบเจกต์เหล่านั้นเรียกว่า MIB-Object สามารถนำมาใช้ในการวิเคราะห์การโจมตีได้อย่างสะดวก เช่น icmpInMsgs (จำนวน ICMP Message ที่เข้าสู่เครือข่าย) หรือ ipInReceives (จำนวน IP Datagram ที่เข้าสู่เครือข่าย) แต่เบื้องต้นการสร้าง MIB-Object นั้นใช้สำหรับจัดการเรื่องประสิทธิภาพ (Performance Management) การจัดการความผิดพลาด (Fault Management) การจัดการองค์ประกอบของซอฟต์แวร์ (Configuration Management) หรือ การจัดการด้านบัญชี (Accounting Management) มากกว่าการจัดการเรื่องความปลอดภัย (Security Management) ถึงแม้จะมีงานวิจัยที่นำเสนอการใช้ MIB-Object สำหรับจัดการด้านความปลอดภัย (Gaspary *et al.*, 2005);(Bao Cui-Mei, 2009) แต่ก็ยังคงมีช่องว่างระหว่างการบริหารจัดการเครือข่ายและการจัดการด้านความปลอดภัยเหลืออยู่ นั่นคืออ็อบเจกต์ใน MIB ที่มีอยู่ไม่เพียงพอต่อการนำมาใช้ตรวจจับการบุกรุกบนระบบคอมพิวเตอร์และเครือข่าย

วิทยานิพนธ์นี้ได้นำเสนอสถาปัตยกรรมโพรไฟล์สำหรับใช้ตรวจจับความผิดปกติบนระบบคอมพิวเตอร์และเครือข่าย โดยนำเสนออ็อบเจกต์ใหม่ที่จำเป็นเพื่อใช้เป็นพารามิเตอร์สำหรับการสร้างโพรไฟล์ในการตรวจจับความผิดปกติรูปแบบ Anomaly Detection โดยใช้โพรโทคอล SNMP และ MIB เป็นเครื่องมือในการจัดการกับอ็อบเจกต์ดังกล่าว

1.2 วัตถุประสงค์ของการวิจัย

1. วิเคราะห์ออกแบบและสร้างสถาปัตยกรรมโพรไฟล์สำหรับใช้เป็นแม่แบบในการวิเคราะห์และตรวจสอบการบุกรุกที่เกิดขึ้นบนระบบคอมพิวเตอร์และเครือข่าย
2. สร้างแม่แบบในการตรวจจับความผิดปกติของเครือข่ายที่เกิดจากการโจมตีบางชนิดโดยอิงจากโพรไฟล์ที่สร้างขึ้น

1.3 ขอบเขตการดำเนินการวิจัย

1. สร้างโพรไฟล์เพื่อใช้เป็นแม่แบบในการวิเคราะห์และตรวจสอบการบุกรุกที่เกิดขึ้นบนระบบคอมพิวเตอร์และเครือข่าย
2. สร้างแม่แบบในการตรวจจับความผิดปกติของเครือข่ายที่เกิดจากการโจมตีประเภท SYN Flood Attack, Land Attack, DNS Flood, Null Scan และ Xmas Scan โดยอิงจากโพรไฟล์ที่สร้างขึ้น

1.4 ขั้นตอนและระยะเวลาการดำเนินการ

1.4.1 ขั้นตอนการดำเนินการ

1. ศึกษาเอกสารและงานวิจัยที่เกี่ยวข้อง
2. ศึกษาการตรวจจับการบุกรุกรูปแบบต่างๆ ที่เกิดขึ้นบนเครือข่าย
3. สร้างชนิดข้อมูลสำหรับใช้เป็นพารามิเตอร์ในการตรวจจับการบุกรุก
4. สร้างโพรไฟล์สำหรับใช้ในการตรวจจับการบุกรุกบนระบบคอมพิวเตอร์และเครือข่าย
5. สร้างเครื่องมือสำหรับตรวจจับการบุกรุกโดยอิงจากโพรไฟล์
6. เขียนบทความวิจัยและเผยแพร่
7. จัดทำเอกสารวิทยานิพนธ์

ตารางที่ 1-1 แผนการดำเนินการ

ขั้นตอน	เดือน																		
	2553				2554												2555		
	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4
1.	■	■	■	■	■	■														
2.						■	■	■	■											
3.									■	■	■									
4.											■	■	■	■						
5.														■	■	■				
6.												■	■	■	■	■	■	■		
7.								■	■	■	■	■	■	■	■	■	■	■	■	■

1.4.2 ระยะเวลาการดำเนินการ

มิถุนายน 2553 – พฤษภาคม 2555

1.5 สถานที่และเครื่องมือที่ใช้ดำเนินการวิจัย

1.5.1 สถานที่

ห้องวิจัยกลุ่มคอมพิวเตอร์และเครือข่าย ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่

1.5.2 เครื่องมือที่ใช้

1) ด้านฮาร์ดแวร์

- คอมพิวเตอร์ส่วนบุคคล CPU Intel Core 2 Quad 2.83 GHz, RAM 3.24 GB จำนวน 1 เครื่อง
- คอมพิวเตอร์ส่วนบุคคล CPU Intel ® Celeron ® 2.40 GHz, RAM 256 MB จำนวน 1 เครื่อง
- อุปกรณ์เชื่อมต่อเครือข่ายฮับ

2) ด้านซอฟต์แวร์

- ระบบปฏิบัติการ Windows XP Professional Version 2002 Service Pack 3
- ระบบปฏิบัติการ FreeBSD 7.3
- ระบบปฏิบัติการ Ubuntu 9.10
- ชุดคลังโปรแกรม NET-SNMP 5.4
- โปรแกรมเว็บเซิฟเวอร์ Apache
- เครื่องมือสำหรับสร้างกราฟ RRDtool
- ตัวแปลภาษา C และ PHP
- โปรแกรม Weka

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้สถาปัตยกรรมโพรไฟล์ที่ใช้เป็นแม่แบบสำหรับใช้ในการตรวจจับความผิดปกติบนระบบคอมพิวเตอร์และเครือข่าย

2. ได้แม่แบบของระบบตรวจจับการบุกรุกโดยอิงจากโพรไฟล์ที่สร้างขึ้น เพื่อใช้สำหรับตรวจจับการบุกรุกหรือบอกถึงความผิดปกติที่เกิดจากการโจมตีรูปแบบ DoS บางประเภทได้

เนื้อหาในรายงานวิทยานิพนธ์ชุดนี้ประกอบด้วย บทที่ 2 ซึ่งจะกล่าวถึงทฤษฎีและงานวิจัยที่เกี่ยวข้องเกี่ยวกับ SNMP, MIB, โพรไฟล์ และการตรวจจับความผิดปกติบนระบบคอมพิวเตอร์และเครือข่าย บทที่ 3 คือการวิเคราะห์ ออกแบบและพัฒนาสถาปัตยกรรมโพรไฟล์และ MIB+ ซึ่งคือ กลุ่มของ MIB-Object ที่ได้นำเสนอขึ้นใหม่ บทที่ 4 กล่าวถึงการออกแบบระบบตรวจจับการบุกรุกโดยใช้ MIB+ บทที่ 5 กล่าวถึงการพัฒนาระบบตรวจจับการบุกรุก บทที่ 6 เป็นการทดสอบระบบตรวจจับการบุกรุกที่ได้พัฒนาขึ้น และบทที่ 7 เป็นบทสรุปซึ่งจะกล่าวถึงปัญหาและข้อเสนอแนะของการทำวิทยานิพนธ์ชุดนี้

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 บทนำ

เนื้อหาในบทนี้จะกล่าวถึงทฤษฎี หลักการและงานวิจัยที่เกี่ยวข้องกับการทำวิทยานิพนธ์โดยที่เนื้อหาในส่วนแรกจะกล่าวถึงภัยคุกคามทางคอมพิวเตอร์ คุณสมบัติในการรักษาความปลอดภัย ประเภทของระบบตรวจจับการบุกรุก รายละเอียดของโพรไฟล์ที่ใช้งานกันอยู่ รายละเอียดและหน้าที่ของ Simple Network Management Protocol (SNMP) จากนั้นจะเป็นการอธิบายคุณสมบัติของ Management Information Base (MIB) ที่ใช้เป็นแนวคิดในการพัฒนาอ็อบเจกต์เพื่อนำมาสร้างโพรไฟล์ และเครื่องมือที่เกี่ยวข้องในการทำวิทยานิพนธ์นี้ตามลำดับ

2.2 ภัยคุกคามและการบุกรุกระบบคอมพิวเตอร์

ภัยคุกคาม (Threat) คือ บุคคล สิ่งของหรือเหตุการณ์ต่างๆ ที่มุ่งร้ายหรือเป็นสาเหตุของภัยอันตราย ที่เกิดขึ้นกับระบบคอมพิวเตอร์ในรูปแบบของการทำลาย การเปิดเผยข้อมูล แก้ไขข้อมูลและรวมไปถึงการทำให้ระบบไม่สามารถให้บริการแก่ผู้ใช้ได้ โดยภัยคุกคามอาจจะเกิดขึ้นจากอุบัติเหตุ เช่น ไฟไหม้ น้ำท่วม ฯลฯ หรือเกิดจากเจตนาของบุคคลที่ประสงค์ร้ายต่อระบบ เช่น การพยายามที่จะข้ามผ่านกระบวนการรักษาความปลอดภัยของระบบคอมพิวเตอร์ (NCSC-TG-004, 1988)

การบุกรุก (Intrusion) คือ พฤติกรรมความพยายามกระทำการใดๆ ที่ส่งผลต่อการรักษาความลับ ความสมบูรณ์ และความพร้อมใช้ของทรัพยากรในระบบ หรือพยายามข้ามผ่านมาตรการรักษาความปลอดภัยของระบบคอมพิวเตอร์ เช่น พยายามใช้สิทธินอกเหนือจากที่ได้รับหรือพยายามใช้สิทธินั้นไม่ทางที่ผิด (Bace, 2001)

Stallings (1995) ได้จัดกลุ่มของวิธีการบุกรุกออกเป็น 2 แบบ คือ การบุกรุกแบบ Active และการบุกรุกแบบ Passive ซึ่งการบุกรุกแต่ละแบบมีรายละเอียดดังนี้

2.2.1 การบุกรุกแบบ Active

การบุกรุกแบบ Active เป็นการบุกรุกที่ทำให้เกิดการเปลี่ยนแปลงของข้อมูล หรือสร้างข้อมูลขึ้นมาใหม่โดยการปลอมแปลง เช่น การเปลี่ยนแปลงข้อมูลในไฟล์ไม่ว่าจะเป็นแก้ไข เพิ่มเติม หรือว่าลบข้อมูล รวมถึงการทำให้ระบบไม่สามารถให้บริการผู้ใช้ได้ การโจมตีแบบนี้ผู้ดูแลสามารถตรวจจับได้ง่าย เนื่องจากมีร่องรอยการกระทำที่จัดว่าเป็นการบุกรุก

2.2.2 การบุกรุกแบบ Passive

การบุกรุกแบบ Passive เป็นการบุกรุกที่ไม่ทำให้เกิดการเปลี่ยนแปลงของข้อมูล แต่ผู้บุกรุกสามารถเข้าถึงข้อมูลได้โดยไม่ได้รับอนุญาต เช่น การดักจับข้อมูลในสายสัญญาณหรือเครือข่าย การบุกรุกประเภทนี้ตรวจจับได้ยาก แต่สามารถป้องกันได้ง่าย เช่น การเข้ารหัสข้อมูลก่อนการรับส่ง เป็นต้น

2.3 คุณสมบัติในการรักษาความปลอดภัย

คุณสมบัติ 3 ประการของการรักษาความปลอดภัยซึ่งได้แก่ การรักษาไว้ซึ่งความลับ การรักษาความสมบูรณ์ และการรักษาความพร้อมใช้ของข้อมูล โดยคุณสมบัติแต่ละข้อมีรายละเอียดดังนี้

- **การรักษาไว้ซึ่งความลับของข้อมูล** (Confidentiality) เป็นการรับรองว่าจะมีการเก็บข้อมูลไว้เป็นความลับ และมีเพียงผู้มีสิทธิเท่านั้นจึงจะเข้าถึงข้อมูลได้
- **การรักษาไว้ซึ่งความสมบูรณ์ของข้อมูล** (Integrity) เป็นการรับรองว่าข้อมูลไม่มีการเปลี่ยนแปลงหรือถูกทำลายโดยผู้ไม่มีสิทธิ ไม่ว่าจะเป็นโดยอุบัติเหตุหรือเจตนา
- **การรักษาไว้ซึ่งความพร้อมใช้ของข้อมูล** (Availability) เป็นการรับรองว่าข้อมูลและบริการต่างๆ มีความพร้อมที่จะเรียกใช้ได้ตามสิทธิทุกครั้งที่ร้องขอ

ผู้บุกรุกจะทำการโจมตีในลักษณะหรือรูปแบบที่แตกต่างกัน ทั้งนี้ขึ้นอยู่กับวัตถุประสงค์ในการโจมตี Stallings (1995) ได้แบ่งการบุกรุกตามลักษณะการกระทำได้ 4 ประเภท ดังนี้

1. Interruption คือการขัดขวางการทำงานของระบบคอมพิวเตอร์ ทำให้ไม่สามารถให้บริการหรือไม่สามารถใช้งานได้ตามปกติ เช่น การทำลายอุปกรณ์หรือเครื่องคอมพิวเตอร์ การทำให้เครือข่ายท่วม (Network Flooding) เป็นต้น

2. Interception คือการที่บุคคลหรือโปรแกรมที่ไม่ได้รับอนุญาตสามารถเข้าถึงทรัพยากรหรือข้อมูลโดยวิธีการที่ไม่ได้รับอนุญาต เช่น การดักฟังสัญญาณการสื่อสารในเครือข่าย เป็นต้น

3. Modification คือการที่บุคคลหรือโปรแกรมที่ไม่ได้รับอนุญาตสามารถเข้าถึงทรัพยากรและแก้ไขข้อมูล เช่น การแก้ไขข้อมูลในไฟล์ การปรับเปลี่ยนโปรแกรมให้มีการทำงานที่ต่างไปจาการทำงานปกติหรือการแก้ไขข้อมูลในเครือข่าย เป็นต้น

4. Fabrication คือการที่บุคคลหรือโปรแกรมที่ไม่ได้รับอนุญาตปลอมแปลงข้อมูลขึ้นมาในระบบ เช่น การปลอมข้อมูลข่าวสารที่รับส่งในเครือข่าย เป็นต้น

2.4 การตรวจจับความผิดปกติ

ศุภโชค (2548) ได้ศึกษาแนวทางในการตรวจจับการบุกรุกซึ่งกล่าวโดยสรุปคือสามารถจำแนกวิธีการตรวจจับการบุกรุกได้เป็น 2 ประเภท คือ Misuse Detection และ Anomaly Detection ซึ่งรายละเอียดของแต่ละประเภท จะเสนอดังต่อไปนี้

2.4.1 Misuse Detection

การตรวจจับการบุกรุกประเภทนี้ใช้วิธีการวิเคราะห์พฤติกรรมการบุกรุกโดยจะทำการเปรียบเทียบพฤติกรรมของผู้ใช้ในขณะนั้นเทียบกับพฤติกรรมที่กำหนดไว้ หากพฤติกรรมการใช้งานผิดจากพฤติกรรมที่กำหนดไว้จะถือว่าเป็นการบุกรุก โดยทั่วไปแล้วการตรวจจับการบุกรุกแบบนี้จะใช้กับการบุกรุกที่เกิดจากภายในองค์กร เช่น ผู้ใช้มีสิทธิในการเข้าใช้งานระบบ แต่ใช้สิทธิในทางที่ผิด ยกตัวอย่างเช่น พยายามแก้ไขแฟ้มรหัสผ่านซึ่งอนุญาตให้ผู้ใช้ที่มีสิทธิสูงสุดเท่านั้น พฤติกรรมเช่นนี้ถือว่าเป็นการบุกรุก (Qayyum และคณะ,2005); (Tylman, 2008) ตัวอย่างระบบการตรวจจับการบุกรุกประเภทนี้ ได้แก่ Basset, Snort, NetRedar, IDES/NIDES, NetStalker, TCP Wrapper, Tripwire, SATAN, NIDS-Sax เป็นต้น วิธีการทั้งหมดนี้มีหลักการตรวจจับการบุกรุกคล้ายกันคือ ต้องมีรูปแบบ (Known Pattern) ไว้เป็นต้นแบบในการเปรียบเทียบกับพฤติกรรมที่ตรวจจับ วิธีการตรวจจับการบุกรุกประเภทนี้มีความผิดพลาดในการตรวจจับต่ำ แต่จะไม่สามารถตรวจจับการโจมตีที่เกิดขึ้นมาใหม่ได้ เนื่องจากการโจมตีที่เกิดขึ้นใหม่นั้นจะเป็นพฤติกรรมที่ไม่เคยเกิดขึ้นมาก่อน (Unknown Pattern) ในการสร้างเครื่องมือตรวจจับการบุกรุกรูปแบบนี้อาศัยเครื่องมือต่างๆ เช่น Expert System, Keystroke Monitoring, Model based Intrusion Detection และ State Transition Analysis เป็นต้น (ศุภโชค, 2548)

2.4.2 Anomaly Detection

การตรวจจับการบุกรุกประเภทนี้เป็นวิธีการตรวจจับที่ตรวจสอบพฤติกรรมการใช้งานหรือทรัพยากรในระบบของผู้ใช้ว่าผิดไปจากพฤติกรรมปกติหรือไม่ เช่น ช่วงเวลาการใช้งานของผู้ใช้ผิดไปจากปกติหรือไม่ หรือการใช้พื้นที่ในหน่วยความจำ ตัวอย่างระบบการตรวจจับการบุกรุกที่ใช้เทคนิค Anomaly Detection ได้แก่ SPADE, PHAD และ SVLNM (Salem and Karim, 2008);(Zhang, Han and Ren, 2009);(Hsieh, Huang and Chen, 2011) ซึ่งเทคนิคการตัดสินใจว่าการทำงานเป็นพฤติกรรมการบุกรุกหรือไม่นั้นมีหลายเทคนิคด้วยกัน ได้แก่ Statistical Anomaly Detection, Classifier และ Neural Network เป็นต้น

2.5 ระบบตรวจจับการบุกรุก

ระบบตรวจจับการบุกรุก (Intrusion Detection System : IDS) คือระบบที่ประกอบด้วยฮาร์ดแวร์หรือซอฟต์แวร์ที่ใช้สำหรับตรวจสอบการใช้งานและความพยายามในการใช้งานคอมพิวเตอร์และเครือข่ายที่ขัดกับข้อบังคับและวัตถุประสงค์การใช้งานที่ได้ระบุไว้ในนโยบายการรักษาความปลอดภัยขององค์กร ระบบตรวจจับการบุกรุกนั้นแบ่งออกได้เป็นหลายประเภทตามรายละเอียดในข้อ 2.6 ดังนี้

2.6 ประเภทของระบบตรวจจับการบุกรุก

ในการจัดประเภทของระบบตรวจจับการบุกรุกนั้นนักวิจัยต่างๆ ได้จัดแบ่งออกเป็นหลายรูปแบบ โดยใช้หลักเกณฑ์ต่างๆ เช่น แหล่งข้อมูลที่นำมาวิเคราะห์ แนวทางในการตรวจจับการบุกรุก ช่วงเวลาในการวิเคราะห์การบุกรุกหลังจากเกิดเหตุการณ์ขึ้น เป็นต้น ซึ่งส่วนใหญ่จะนิยมจัดประเภทโดยใช้แหล่งข้อมูลที่นำมาวิเคราะห์ โดยแบ่งเป็น 3 ประเภท คือ Network-based Intrusion Detection System (NIDS) Host-based Intrusion Detection System (HIDS) และ Application-based Intrusion Detection System (AIDS) ซึ่งคุณสมบัติของแต่ละประเภทมีดังนี้

2.6.1 Network-based Intrusion Detection System (NIDS)

เป็นระบบตรวจจับการบุกรุกที่ติดตามและวิเคราะห์แพ็กเก็ตที่รับส่งกันในเครือข่ายเพื่อดูว่ามีผู้บุกรุกหรือมีความผิดปกติเกิดขึ้นหรือไม่ เช่น การโจมตีที่ทำให้เกิดการ

ปฏิเสธการให้บริการ (Denial of Service: DoS) หรือความพยายามในการเจาะเข้ามาในระบบคอมพิวเตอร์หรือเครือข่าย เป็นต้น โดยทำการดักจับข้อมูลบนอุปกรณ์เครือข่าย NIDS สามารถติดตามข้อมูลบนเครือข่ายซึ่งจะมีผลกับเครื่องหลายๆ เครื่องที่เชื่อมต่ออยู่ในเครือข่ายเดียวกัน และสามารถตรวจจับการบุกรุกที่เกิดขึ้นกับเครื่องเหล่านั้นได้อีกด้วย

2.6.2 Host-based Intrusion Detection System (HIDS)

เป็นระบบตรวจจับการบุกรุกที่รวบรวมข้อมูลของแต่ละเครื่องคอมพิวเตอร์ เพื่อตรวจสอบว่าโปรแกรมหรือผู้ใช้คนใดที่ทำให้เกิดการบุกรุกขึ้นในระบบ และผลของการบุกรุกเป็นอย่างไร ระบบ HIDS ส่วนใหญ่จะเก็บรวบรวมข้อมูลจากบันทึกการทำงานของระบบปฏิบัติการแล้วนำข้อมูลเหล่านั้นมาวิเคราะห์เพื่อค้นหาเหตุการณ์ผิดปกติหรือการบุกรุกที่เกิดขึ้นบนเครื่องคอมพิวเตอร์เป้าหมาย

2.6.3 Application-based Intrusion Detection System (AIDS)

ระบบตรวจจับการบุกรุกประเภทนี้จะมีการทำงานที่คล้ายกับ HIDS แต่จะรวบรวมข้อมูลการทำงานของโปรแกรมประยุกต์ที่ทำงานบนเครื่องคอมพิวเตอร์มาใช้ในการวิเคราะห์เพื่อตรวจสอบว่ามีพฤติกรรมผิดปกติหรือใช้สิทธิเกินขอบเขตของผู้ใช้ที่กำหนดไว้หรือไม่

2.7 คุณลักษณะของระบบตรวจจับการบุกรุกที่ดี

ระบบตรวจจับการบุกรุกที่ดีควรมีคุณสมบัติดังต่อไปนี้ (Price, 1998)

1. Run Continually หมายถึง ระบบตรวจจับการบุกรุกจะต้องทำงานอยู่ตลอดเวลาโดยไม่ต้องมีการควบคุมของผู้ดูแลระบบ และต้องมีความน่าเชื่อถือเพียงพอที่จะทำงานในลักษณะอยู่เบื้องหลังได้ (Background Process) แต่ผู้ดูแลระบบต้องสามารถตรวจสอบการทำงานของระบบตรวจจับการบุกรุกได้

2. Fault Tolerant หมายถึง ระบบตรวจจับการบุกรุกต้องยังคงมีความสามารถที่จะทำงานต่อไปในกรณีที่เครื่องคอมพิวเตอร์เกิดปัญหาหรือข้อผิดพลาด และไม่ต้องมีการสร้างฐานข้อมูลความรู้ใหม่ (Knowledge-based) ทุกครั้งที่ระบบเริ่มทำงาน

3. Subversion Resistance หมายถึง ระบบตรวจจับการบุกรุกต้องมีความสามารถในการตรวจสอบตัวเองเพื่อไม่ให้ถูกลบ ทำลาย แก้ไข หรือถูกแทนที่ด้วยโปรแกรมอื่นได้

4. Minimal Overhead หมายถึง การทำงานของระบบตรวจจับการบุกรุกนั้นจะต้องส่งผลกระทบกับการทำงานของระบบคอมพิวเตอร์ให้น้อยที่สุด เช่น การทำงานของระบบตรวจจับการบุกรุกนั้นจะต้องใช้ทรัพยากรของระบบคอมพิวเตอร์ให้น้อยที่สุดเท่าที่จะทำได้

5. Observe Deviation หมายถึง ระบบตรวจจับการบุกรุกจะต้องตรวจสอบการทำงานที่ผิดไปจากรูปแบบการทำงานที่ปกติได้

6. Easily Tailored หมายถึง ระบบตรวจจับการบุกรุกนั้นจะต้องสามารถปรับเปลี่ยนหรือแก้ไขตัวเองให้เข้ากับระบบคอมพิวเตอร์ได้ง่าย เนื่องจากแต่ละระบบคอมพิวเตอร์จะมีรูปแบบการใช้งานและกลไกในการป้องกันที่ต่างกัน

7. Changing System Behavior หมายถึง ระบบตรวจจับการบุกรุกจะต้องสามารถปรับการทำงานให้สอดคล้องกับการเปลี่ยนแปลงพฤติกรรมการใช้งานของระบบได้ เช่น เมื่อมีการติดตั้งโปรแกรมใหม่ ระบบ IDS จะต้องสามารถปรับเปลี่ยนการทำงานให้เข้ากับระบบที่เปลี่ยนไปได้

8. Difficult to Fool หมายถึง ระบบตรวจจับการบุกรุกต้องไม่สามารถถูกหลอกโดยผู้บุกรุกได้ง่าย

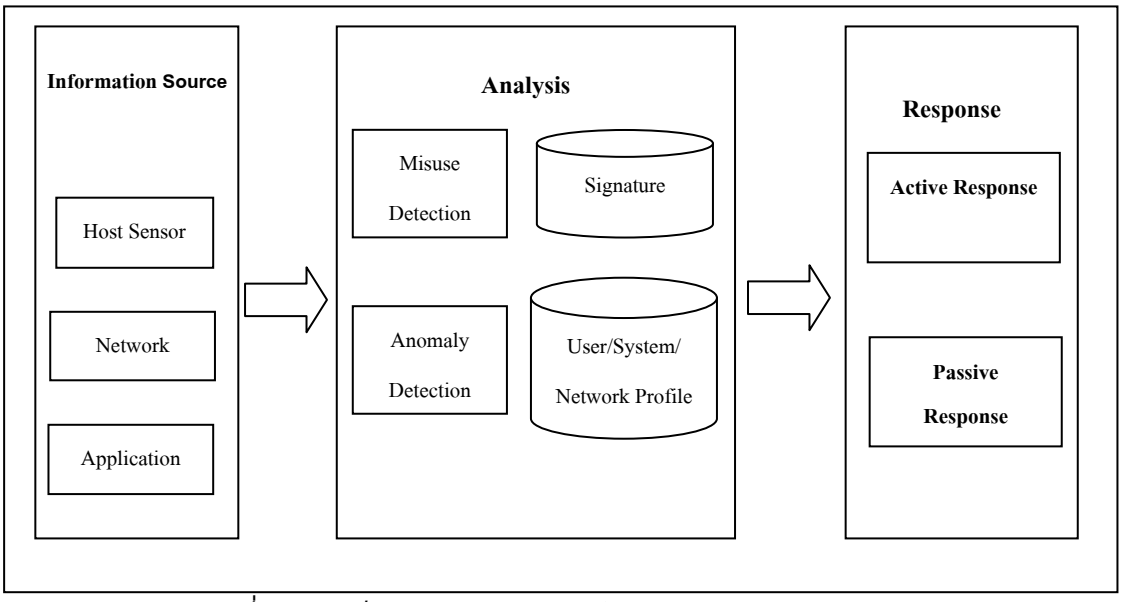
2.8 องค์ประกอบของระบบตรวจจับการบุกรุก

ระบบตรวจจับการบุกรุกส่วนใหญ่จะมีองค์ประกอบและกระบวนการในการทำงานทั่วไปที่เหมือนกัน คือจะประกอบด้วยองค์ประกอบพื้นฐาน 3 ส่วน ดังแสดงในภาพประกอบที่ 2-1 โดยมีรายละเอียดดังนี้ (Bace and Mell, 2001)

1. Information Source (Sensor) ข้อมูลเหตุการณ์และข้อมูลการทำงานจากแหล่งข้อมูลต่างๆ จะถูกนำไปใช้ในการวิเคราะห์เพื่อตัดสินว่า เมื่อใดที่มีการบุกรุกเกิดขึ้น โดยแหล่งข้อมูลเหล่านี้จะนำมาจากข้อมูลในระดับต่างๆ ของระบบ เช่น ข้อมูลระดับเครือข่าย ระดับเครื่องคอมพิวเตอร์ และระดับโปรแกรมประยุกต์ที่มีการใช้งานอยู่ในระบบ

2. Analysis เป็นส่วนที่ทำหน้าที่ในการจัดการและวิเคราะห์ข้อมูลที่ได้รับมาจากแหล่งข้อมูลต่างๆ แล้วตัดสินว่าเหตุการณ์หรือการกระทำที่กำลังเกิดขึ้นเป็นการบุกรุกหรือไม่ โดยแนวทางที่ใช้ในการวิเคราะห์มีสองรูปแบบคือ Anomaly Detection และ Misuse Detection

3. Response เป็นชุดของการกระทำเมื่อระบบตรวจจับได้ว่ามีการบุกรุกเกิดขึ้น



ภาพประกอบที่ 2-1 องค์ประกอบระบบตรวจจับการบุกรุก (Bace and Mell, 2001)

2.9 โพรไฟล์ (Profile)

ในปี ค.ศ. 1980 Anderson (1980) ได้นำเสนอแนวคิดที่ว่า โดยปกติแล้วพฤติกรรมของผู้ใช้งานจะมีความคล้ายคลึงกันเสมอในการใช้ในแต่ละครั้ง และเมื่อพฤติกรรมของผู้ใช้งานเปลี่ยนแปลงไปจากเดิมมากอาจจะสรุปได้ว่ามีความผิดปกติเกิดขึ้น ดังนั้นหากสามารถบันทึกพฤติกรรมการใช้งานของผู้ใช้ไว้ได้ ก็อาจจะสามารถตรวจหาพฤติกรรมที่ผิดปกติได้ โดยการนำพฤติกรรมที่ได้จากการบันทึกเปรียบเทียบกับพฤติกรรมที่เกิดขึ้น ซึ่งพฤติกรรมที่บันทึกไว้นี้เรียกอีกชื่อหนึ่งว่า โพรไฟล์ จากแนวความคิดนี้ จึงนำมาสู่การสร้างเป็นโพรไฟล์เพื่อใช้ตรวจจับการบุกรุกที่รู้จักกันในปัจจุบัน

Lim และคณะ (2008) ได้ให้คำนิยามของโพรไฟล์ว่า คือข้อมูลเฉพาะที่รวบรวมพฤติกรรมต่างๆ ไว้ไม่ว่าจะเป็นการใช้งานของผู้ใช้ การทำงานของระบบ หรือกิจกรรมต่างๆ ที่

เกิดขึ้นบนคอมพิวเตอร์และเครือข่าย มีลักษณะที่เฉพาะเจาะจงกับองค์กร หน่วยงานหรือเครือข่ายคอมพิวเตอร์ ซึ่งรายละเอียดในแต่ละโพรไฟล์นั้นจะขึ้นอยู่กับวัตถุประสงค์ในการใช้งาน เช่น ใช้สำหรับวิเคราะห์การจราจรบนเครือข่าย ใช้สำหรับตรวจจับการทำงานที่ผิดปกติของผู้ใช้ในระบบ เป็นต้น

Amoroso (1999) ได้เสนอแนวทางสำหรับการสร้างโพรไฟล์ว่าควรคำนึงถึงองค์ประกอบพื้นฐานดังนี้

- ในการสร้างโพรไฟล์สำหรับผู้ใช้หรือระบบ จำเป็นต้องมีการประเมินลักษณะพฤติกรรมการใช้งานของผู้ใช้หรือระบบก่อน ซึ่งในการประเมินพฤติกรรมนี้เป็นสิ่งสำคัญซึ่งจะส่งผลต่อการเกิดช่องโหว่และนำไปสู่การบุกรุกได้ในภายหลัง
- ต้องคอยสังเกตพฤติกรรมผู้ใช้และระบบเพื่อที่จะใช้ในการปรับเปลี่ยนรายละเอียดต่างๆ ในโพรไฟล์ ซึ่งในการปรับเปลี่ยนรายละเอียดของโพรไฟล์นั้นต้องคำนึงถึงสถิติความน่าจะเป็นในการเกิดเหตุการณ์ที่ผิดปกติ การปรับปรุงโพรไฟล์ควรทำได้อย่างอัตโนมัติ
- ข้อมูลที่จะนำมาสร้างเป็นโพรไฟล์นั้นควรจะมาจากหลายๆ แหล่งข้อมูล เพื่อจะนำไปใช้ในการทำนายพฤติกรรมหรือเหตุการณ์ที่จะเกิดขึ้นได้อย่างมีประสิทธิภาพ

El-Ghali และ Masri (2009) กล่าวว่า โพรไฟล์เป็นกระบวนการในการเก็บรวบรวมและบันทึกเหตุการณ์การทำงานในระหว่างที่มีการทำงานเกิดขึ้น เพื่อใช้สำหรับตรวจจับพฤติกรรมหรือเหตุการณ์ที่เบี่ยงเบนไปจากเหตุการณ์ที่อยู่ในโพรไฟล์

Zheng และคณะ (2010) ได้กล่าวว่า โพรไฟล์สามารถนำไปประยุกต์ใช้ได้ในหลายเรื่อง สามารถจะสร้างเป็นโครงสร้างข้อมูลที่มีความแตกต่างกันได้ แต่วัตถุประสงค์ของโพรไฟล์ก็เพื่ออธิบายถึงเป้าหมายของระบบ หรือพฤติกรรมของผู้ใช้ ซึ่งใน Network Intrusion Detection System นั้นใช้โพรไฟล์เพื่อบันทึกข้อมูลกิจกรรมของเครือข่าย อาทิเช่น จำนวนข้อมูลที่ถูกประมวลผลในหนึ่งหน่วยเวลา ชนิดของแพ็กเก็ต และโปรโตคอลที่ใช้ เป็นต้น ส่วนโพรไฟล์ใน Host-based Intrusion Detection System นั้นจะเกี่ยวกับการเก็บข้อมูลทรัพยากรของระบบ เช่น สถานการณ์ใช้งานของหน่วยประมวลผล การใช้งานของหน่วยความจำ และข้อมูลความผิดพลาดของระบบ

งานวิจัยที่เกี่ยวข้องกับการศึกษาและพัฒนาระบบตรวจจับการบุกรุกในช่วงต้นๆ ของการศึกษาเรื่องนี้ซึ่งเสนอโดย Anderson (1980) เรื่อง “Computer Security Threat Monitoring และ Surveillance” โดยนำเสนอแนวความคิดพื้นฐานว่า พฤติกรรมปกติของผู้ใช้

สามารถอธิบายได้โดยวิเคราะห์จากกิจกรรมต่างๆ ในล็อกไฟล์และการบุกรุกระบบคอมพิวเตอร์สามารถตรวจจับได้โดยการใช้ข้อมูลที่ได้มาจากลักษณะของพฤติกรรมทั่วไป

ต่อมา Denning และ Neumann (1985) ได้นำเสนอต้นแบบการตรวจจับการบุกรุกแบบ Real-Time ชื่อว่า Intrusion Detection Expert System (IDES) โดยมีการทำงานแบบอิงกฎ (Rule-based) ใช้ในการตรวจจับกิจกรรมที่เป็นอันตรายหรือกิจกรรมที่คาดว่าเป็นการบุกรุก

สองปีต่อมา Denning (1987) ได้สร้างกรอบการทำงาน (Framework) ที่สามารถใช้งานได้ทั่วไปในการตรวจจับการบุกรุก ซึ่งมีโพรไฟล์เป็นองค์ประกอบหลัก โดยโพรไฟล์จะเป็นโครงสร้างลักษณะพฤติกรรมของผู้ทำให้เกิดกิจกรรมในระบบ (Subject) ที่เกี่ยวพันกับการกระทำ (Object) ซึ่งจะมองในเชิงสถิติและรูปแบบการทำกิจกรรม Denning ได้เสนอโครงสร้างของโพรไฟล์ที่มีส่วนประกอบสองส่วนหลักๆ คือ

1. Subject-และ-Object-Dependent
2. Subject-และ-Object-Independent

นอกจากนี้ยังได้คิดวิธีการสร้างโพรไฟล์ ซึ่งมีวิธีการที่เป็นไปได้ในการสร้างโพรไฟล์ด้วยกัน 3 วิธี ได้แก่

1) Manual Create: ผู้ที่ดูแลในเรื่องของความปลอดภัยของเครือข่ายเป็นผู้สร้างโพรไฟล์ขึ้นมาเองทั้งหมด
2) Automatic Explicit Create: โพรไฟล์ทั้งหมดสำหรับผู้ใช้ใหม่หรือ Object จะถูกสร้างขึ้นมาโดยอัตโนมัติเพื่อตอบสนองต่อการบันทึกสำหรับใช้ในตรวจสอบ
3) First Use: โพรไฟล์จะถูกสร้างขึ้นแบบอัตโนมัติเมื่อ Subject มีการใช้งาน Object ในครั้งแรก

ในปี ค.ศ. 1995 ได้มีการปรับปรุงและขยายความสามารถของระบบตรวจจับการบุกรุกภายใต้ชื่อ Next-Generation Intrusion Detection Expert System (NIDES) (SRI International, 1995) ซึ่งเป็นระบบที่ใช้วิธีทางสถิติเข้ามาช่วย ทำให้ระบบตรวจจับการบุกรุกนั้นทำงานได้อย่างมีประสิทธิภาพมากขึ้น IDES นี้จะไม่ทำการเก็บบันทึกกิจกรรมของผู้ใช้แต่ละคน แต่ใช้ชุดย่อยของตัววัดในการบอกถึงกิจกรรมที่ปกติ เช่น การใช้งานของ CPU เวลาที่ใช้ในการเชื่อมต่อ หรือ กิจกรรมที่เกิดขึ้นบนเครือข่าย เป็นต้น ระบบตรวจจับการบุกรุกก็ได้มีการพัฒนาแนวทางและวิธีการตรวจจับการบุกรุกอีกหลายวิธีต่อมา ไม่ว่าจะเป็น Haystack, Multics

Intrusion Detection และ Alerting System (MIDAS), Network Audit Director และ Intrusion Reporter (NADIR)

หากจะพิจารณาเฉพาะระบบตรวจจับการบุกรุกบนเครือข่ายแล้ว การทำงานของระบบคอมพิวเตอร์หรือกิจกรรมต่างๆ ที่เกิดขึ้นบนเครือข่ายนั้นสามารถเก็บอยู่ในรูปแบบที่เรียกว่าโพรไฟล์ ดังนั้นรูปแบบของโพรไฟล์จึงมีลักษณะที่แตกต่างกัน (Kim *et al.*, 2006) ซึ่งในปี ค.ศ. 2005 Durgin และ Zhang (2005) ได้เสนอแนวคิดการตรวจจับความผิดปกติโดยใช้ Profile-based Network สำหรับตรวจจับการบุกรุกที่เกิดขึ้นบนเครือข่าย และใช้เทคนิคของ Data Mining และการเรียนรู้ของเครื่อง (Machine Learning) ในการวิเคราะห์หาความผิดปกติ

ในปี ค.ศ. 2006 Kim และคณะ (2006) ได้สร้างโพรไฟล์เพื่อใช้สำหรับวิเคราะห์การจราจรที่เกิดขึ้นบนเครือข่ายสองเครือข่าย ในงานวิจัยนี้ต้องการแสดงให้เห็นรูปแบบของการจราจรของแต่ละเครือข่ายว่ามีรูปแบบการจรจรที่แตกต่างกัน ดังนั้นการสร้างโพรไฟล์ในการตรวจจับความผิดปกติของแต่ละเครือข่ายจึงมีความแตกต่างกันด้วย

Salem และ Karim (2008) กล่าวว่า วิธีการสำหรับตรวจจับความผิดปกตินั้นสามารถทำได้โดยการรวบรวมการทำงานที่เป็นปกติทั้งหมดไว้ ไม่ว่าจะเป็นการใช้งานของผู้ใช้ การทำงานของโปรแกรม หรือลักษณะข้อมูลบนเครือข่าย โดยจะเรียกว่า Global Profile เพื่อไว้สำหรับเปรียบเทียบกับกิจกรรมที่เกิดขึ้นในปัจจุบัน ทุกๆ การเบี่ยงเบนของเหตุการณ์ที่เกิดขึ้นจะบอกถึงพฤติกรรมที่ผิดปกติ วิธีการนี้มีประสิทธิภาพในการตรวจจับการโจมตีใหม่ๆ แต่จะพบว่าอัตราการแจ้งเตือนความผิดพลาดค่อนข้างสูง เพราะเนื่องจากหากใช้ Global Profile เพื่อใช้เป็นตัวแทนของเครือข่ายเพียงอย่างเดียวอาจจะทำให้ไม่สามารถมองเห็นลักษณะการทำงานที่แท้จริงในเครือข่ายที่มีการให้บริการต่างๆ ได้ Salem และ Karim จึงได้นำเสนอวิธีการที่เรียกว่า Context-based Profiling และแยกประเภทของโพรไฟล์ออกเป็น Service-based Profiling และ Host-based Profiling

จะเห็นได้ว่าการสร้างโพรไฟล์นั้นขึ้นอยู่กับวัตถุประสงค์ในการใช้งาน โครงสร้างและบริการต่างๆ ที่มีอยู่ในแต่ละองค์กรหรือเครือข่าย ส่งผลให้โพรไฟล์มีรูปแบบและรายละเอียดแตกต่างกัน ซึ่งวัตถุประสงค์ในการสร้างโพรไฟล์ก็เพื่อดูความผิดปกติของการจราจรที่อยู่บนเครือข่าย หรือตรวจสอบการทำงานของระบบคอมพิวเตอร์หรือเครือข่าย ฯลฯ แต่ในงานวิจัยนี้จะสร้างโพรไฟล์สำหรับตรวจจับการบุกเพื่อใช้ในการตรวจสอบการบุกรุกที่เกิดขึ้นบนระบบคอมพิวเตอร์และเครือข่าย จากความสำคัญและประโยชน์ของระบบตรวจจับการบุกรุกพร้อมทั้ง

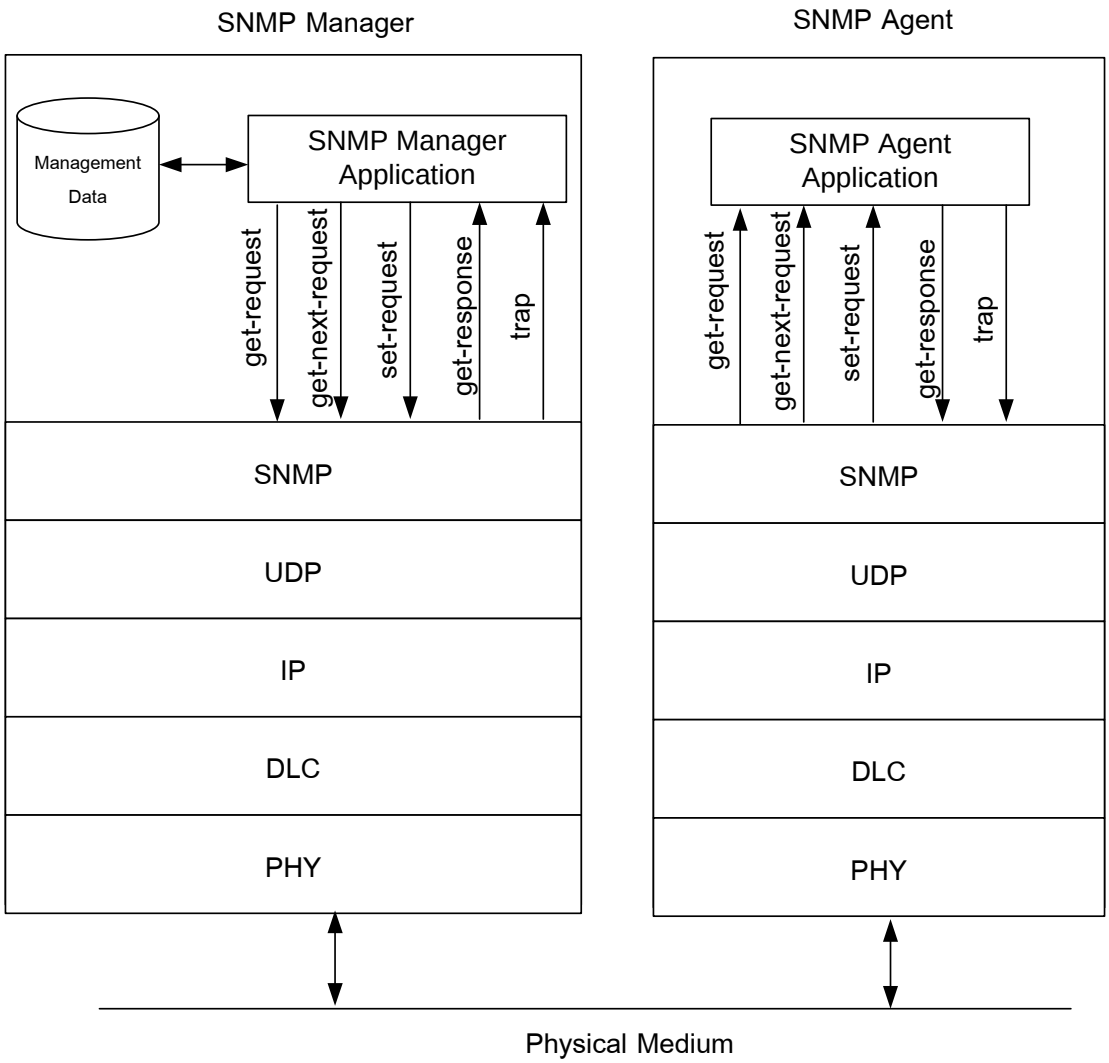
เทคนิคที่ใช้ในการตรวจจับที่ได้กล่าวมานั้น งานวิจัยนี้ได้เลือกใช้แนวทางการตรวจจับการบุกรุกแบบ Anomaly Detection เนื่องจากสามารถตรวจจับการบุกรุกใหม่ๆ ที่เกิดขึ้นได้โดยไม่จำเป็นต้องรู้จักการบุกรุกนั้นมาก่อน และมี False Positive ต่ำ เมื่อเทียบกับแนวทาง Misuse Detection นั่นคือ สามารถตรวจจับการบุกรุกที่เกิดขึ้นทั้งหมดได้

การบุกรุกที่เกิดขึ้นบนระบบคอมพิวเตอร์และเครือข่ายนั้นจะมีการนำเสนอข้อสรุปของชนิดข้อมูลที่สามารถนำไปใช้เป็นตัวแปรในการสร้างโพรไฟล์สำหรับการตรวจจับการบุกรุกในรูปแบบ Anomaly Detection โดยกลุ่มของตัวแปรที่ได้จากการศึกษาครั้งนี้ผู้วิจัยได้นำเสนอในรูปแบบอ็อบเจกต์ของ Management Information Base (MIB) ซึ่งจะกล่าวถึงรายละเอียดของ MIB ในหัวข้อถัดไป

2.10 Simple Network Management Protocol (SNMP)

SNMP เป็นโพรโทคอลในระดับ Application Layer ดังภาพประกอบที่ 2-2 ทำการรับส่งข้อมูลจัดการเครือข่ายระหว่าง SNMP Manager จะทำหน้าที่ในการส่งคำถามไปยัง SNMP Agent เช่น คำสั่ง get-request get-next-request หรือ get-response เป็นต้น และ SNMP Agent จะทำหน้าที่รับและตอบกลับคำร้องที่ส่งมาจาก SNMP Manager ซึ่งในบางครั้งจะส่งข้อความสำหรับแจ้งไปยัง SNMP Manager หากเกิดเหตุการณ์ที่ผิดปกติขึ้น ซึ่งในการส่งข้อมูลกันนี้จะใช้โพรโทคอล UDP ในระดับ Transport Layer ผ่านทางพอร์ต 161 และพอร์ต 162

ภายในส่วนจัดการเครือข่ายจะมีส่วนของ SNMP Manager และฐานข้อมูลอยู่ 2 ชนิด คือ ฐานข้อมูลที่มีอยู่จริง เช่น ฐานข้อมูลเชิงสัมพันธ์ เพื่อใช้เก็บค่าของอ็อบเจกต์ที่ได้จากการส่งคำสั่งสอบถามข้อมูล ซึ่งฐานข้อมูลนี้จะมีค่าของข้อมูลที่มีการเปลี่ยนแปลงบ่อยและมีขนาดใหญ่ และอีกฐานข้อมูลหนึ่งคือ ฐานข้อมูลเสมือน คือ MIB โดย MIB จะถูกแปลงไปเป็นส่วนหนึ่งของทั้งซอฟต์แวร์ Manager และ Agent เพื่อเก็บข้อมูลของอ็อบเจกต์ สำหรับรายละเอียดของ MIB จะกล่าวในหัวข้อถัดไป

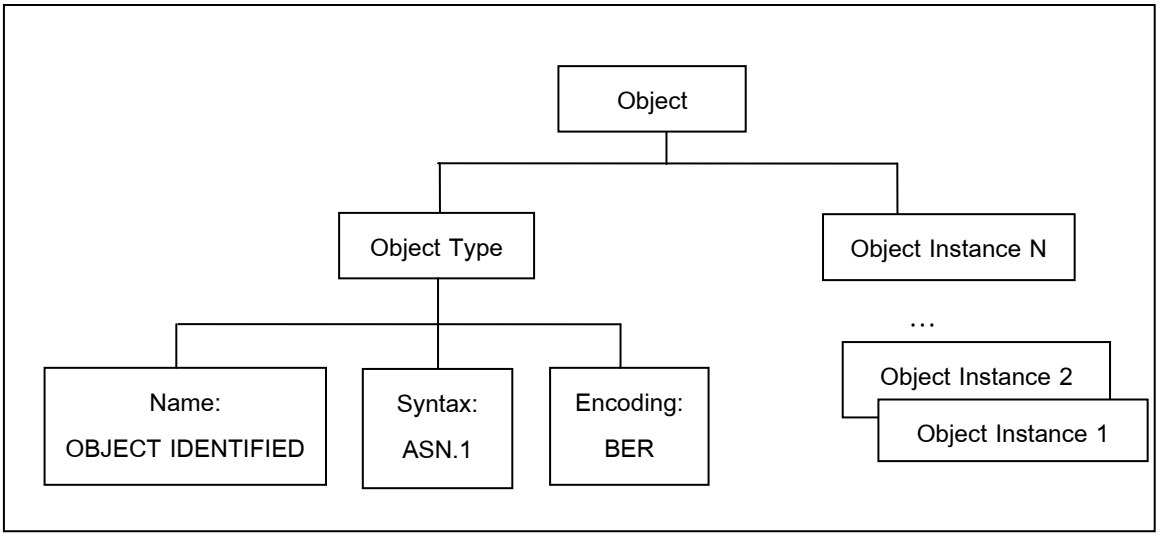


ภาพประกอบ 2-2 SNMP Network Management Architecture (Subramanian,2000)

2.11 Management Information Base (MIB)

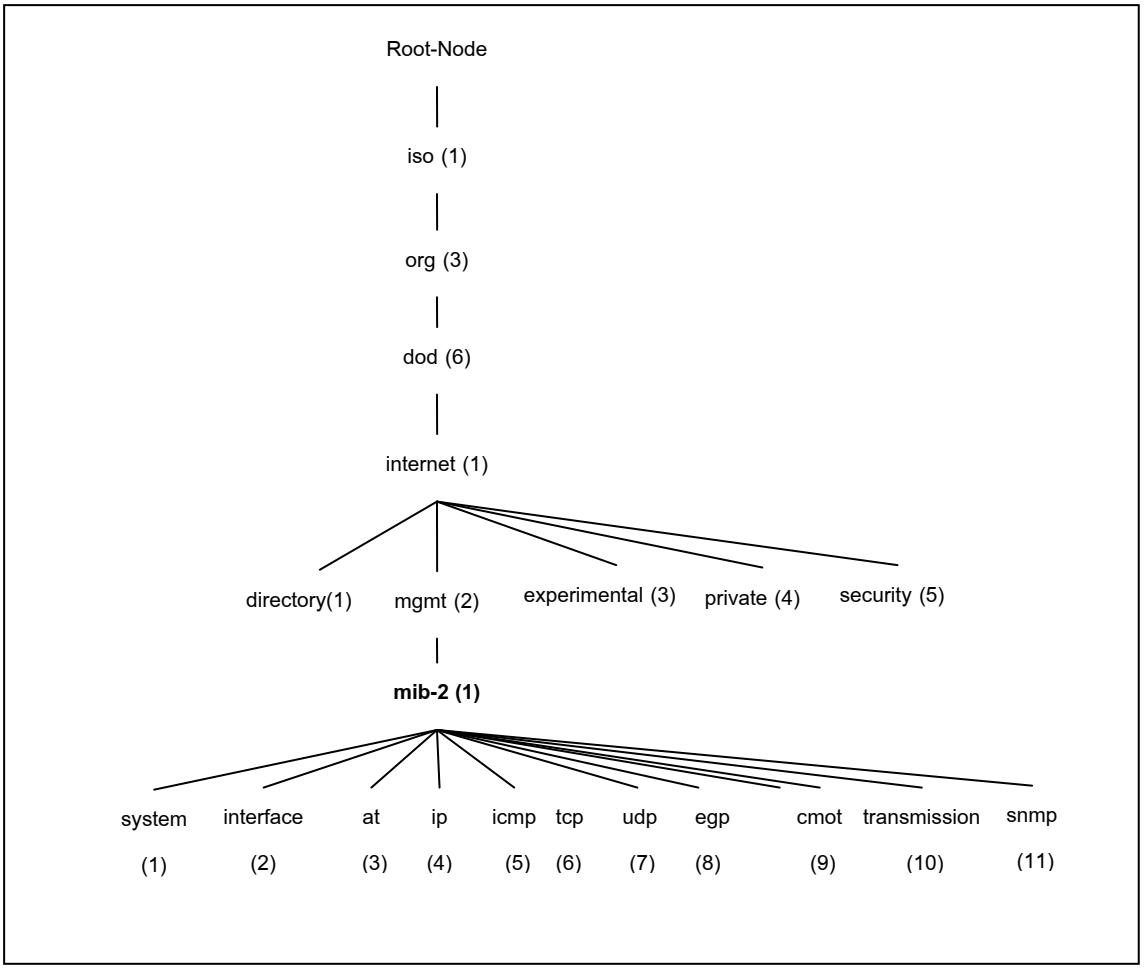
MIB เป็นฐานข้อมูลเสมือนที่ใช้ในการเก็บกลุ่มและความสัมพันธ์ของอ็อบเจกต์ โดยข้อกำหนดและรายละเอียดของ MIB นั้นได้ถูกกำหนดไว้ใน RFC1213 (McCloghrie and Rose, 1991) อ็อบเจกต์ภายใน MIB จะกำหนดและอธิบายรายอะเอียดโดยใช้โครงสร้างของ Structure of Management Information (SMI) โดยที่ SMI นั้นเป็นมาตรฐานหนึ่งที่ใช้กำหนดรายละเอียดและโครงสร้างของอ็อบเจกต์ภายใน MIB ซึ่งรายละเอียดของ SMI นั้นจะกล่าวไว้อยู่ใน RFC1065 (McCloghire and Rose, 1988)

อ็อบเจกต์ คือ โหนดที่เป็นชื่อของตัวแปรหนึ่งภายใน MIB ซึ่งประกอบไปด้วย ชนิดของ อ็อบเจกต์ (Object Type) และตัวแทนของอ๊อบเจ็ค (Object Instance) ดังภาพประกอบที่ 2-3 Object Instance คือส่วนของค่าที่เป็นตัวแทนข้อมูลของอ็อบเจกต์นั้น ใน Object Type เดียวกันอาจจะมี Object Instance ได้มากกว่าหนึ่ง เช่น คอมพิวเตอร์เครื่องหนึ่งมี 2 อินเทอร์เฟสการ์ด



ภาพประกอบ 2-3 ส่วนประกอบของอ็อบเจกต์ (Subranmanian, 1999)

อ็อบเจกต์ในฐานข้อมูล MIB นั้นจะเก็บเรียงตามโครงสร้างข้อมูลแบบต้นไม้ (Management Information Tree: MIT) ของ OSI โดยมีโหนดรากหนึ่งโหนดอยู่ด้านบนสุดและมีโหนดอื่นๆ อยู่ภายใต้โหนดรากในระดับต่ำลงมา ซึ่งในแต่ละโหนดก็จะใช้แทนหนึ่งอ็อบเจกต์ที่ประกอบด้วยชื่อของอ็อบเจกต์และตัวเลขจำนวนเต็มที่มีค่าไม่ซ้ำกับโหนดอื่นๆ เพื่อใช้เป็นตัวระบุหรือใช้อ้างถึงในแต่ละอ็อบเจกต์ การอ้างถึงอ็อบเจกต์จะใช้ตัวเลขของโหนดระบุและจุดคั่นระหว่างตัวระบุของแต่ละอ็อบเจกต์โดยเริ่มที่โหนดรากเสมอ อ็อบเจกต์ทั้งหมดที่ใช้ในมาตรฐานของอินเทอร์เน็ตจะอยู่ภายใต้โหนด internet ซึ่งมีตัวระบุเท่ากับ 1 ดังนั้นในการอ้างถึงอ็อบเจกต์ชื่อ internet จะมีค่าเท่ากับ 1.3.6.1 ดังภาพภาพประกอบที่ 2-4



ภาพประกอบ 2-4 โครงสร้างต้นไม้ของกลุ่มอ็อบเจกต์ใน MIB-II

(McCloghrie และ Rose, 1991)

จากภาพประกอบที่ 2-3 องค์ประกอบของชนิดของอ็อบเจกต์ (Object Type) จะประกอบด้วยชื่อ (Name) ไวยากรณ์ (Syntax) และการเข้ารหัส (Encoding) โดยจะใช้ชื่อเพื่อระบุหรืออ้างถึงอ็อบเจกต์ และใช้ภาษา Abstract Syntax Notation One (ASN.1) ในการกำหนดรายละเอียดของไวยากรณ์ของแต่ละ Object Type และใช้การเข้ารหัสแบบ Basic Encoding Rule (BER) (สุธิพันธ์, 2551) ในการเข้ารหัสข้อมูลของอ็อบเจกต์เพื่อส่งข้อมูลไปมาระหว่าง Manager และ Agent ซึ่งรายละเอียดของแต่ละองค์ประกอบเป็นดังนี้

1. ชื่อ (Name)

ใช้ในการระบุหรืออ้างถึงอ็อบเจกต์ซึ่งต้องมีค่าที่ไม่ซ้ำกันกับอ็อบเจกต์อื่นๆ โดยในอ็อบเจกต์หนึ่งอ็อบเจกต์จะมีชื่อสองอย่าง คือ DESCRIPTOR และ OBJECT IDENTIFIER (OID) ที่สัมพันธ์กัน โดย DESCRIPTOR เป็นชื่อที่เป็นชุดของตัวอักษรภาษาอังกฤษที่ขึ้นต้นด้วยตัวอักษรตัวเล็ก เช่น internet, ipAddTable เป็นต้น ส่วน OBJECT IDENTIFIER เป็นชุดของตัวเลขจำนวนเต็มคั่นด้วยจุดที่สัมพันธ์กับ DESCRIPTOR เช่น อ็อบเจกต์ internet จะมี OBJECT IDENTIFIER เท่ากับ 1.3.6.1 (จากภาพประกอบที่ 2-3) ภาษา ASN.1 ใช้ในการกำหนด OBJECT IDENTIFIER ได้หลายแบบ เช่น

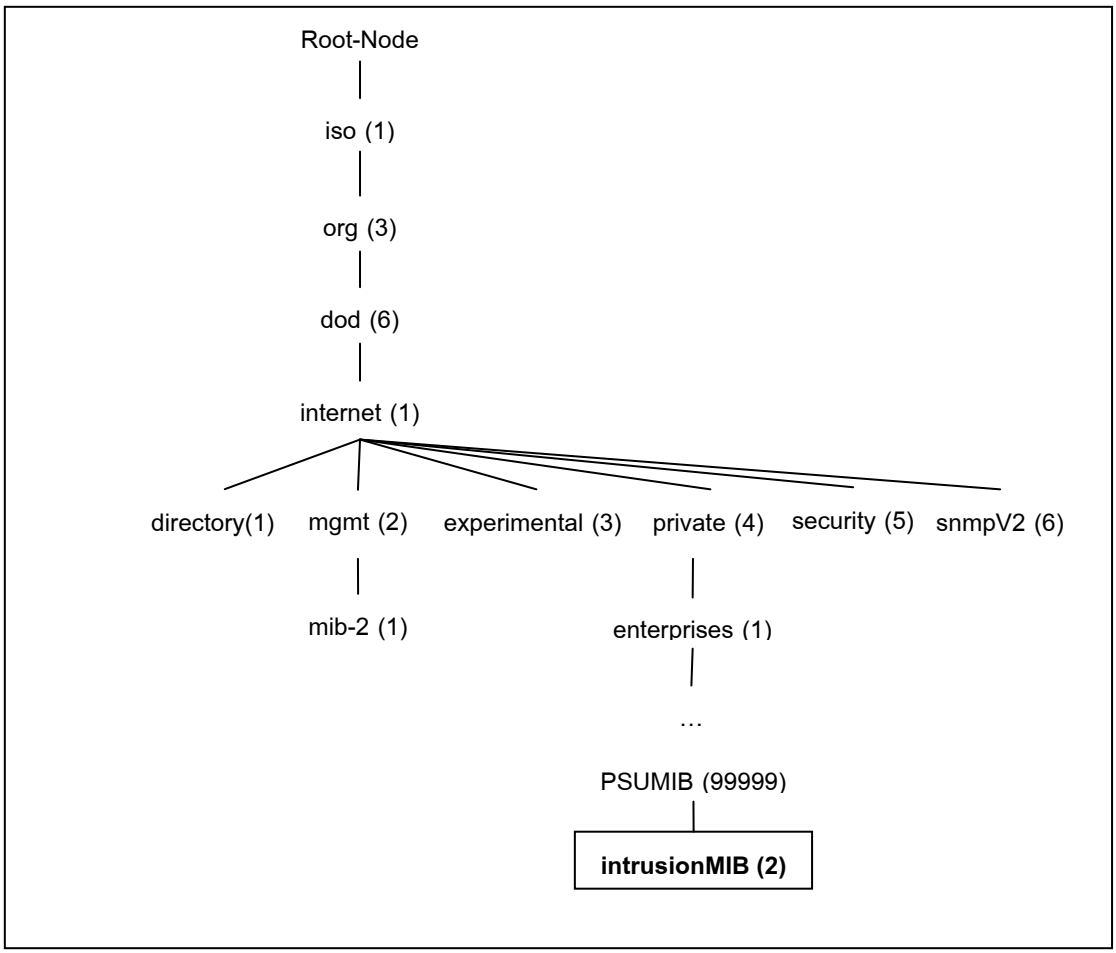
internet OBJECT IDENTIFIER ::= {1 3 6 1}

internet OBJECT IDENTIFIER ::= {ios org dod 1}

internet OBJECT IDENTIFIER ::= {iso(1) org(3) dod(6) internet(1)}

internet OBJECT IDENTIFIER ::= {dod 1}

สำหรับ SNMP นั้นมีอ็อบเจกต์ย่อยภายใต้ internet ดังภาพประกอบที่ 2-3 โดย directory(1) ถูกกำหนดให้เป็นอ็อบเจกต์ที่สงวนเอาไว้ใช้ในอนาคต ส่วน mgmt(2) ใช้ในการเก็บอ็อบเจกต์มาตรฐานทั้งหมดที่กำหนดขึ้นโดย Internet Engineering Task Force (IETF) ส่วน experimental(3) ใช้สำหรับเก็บ อ็อบเจกต์ในการทดลอง private(4) ใช้เก็บอ็อบเจกต์ทั้งหมดของผู้ผลิตอุปกรณ์แต่ละราย งานวิจัยนี้จะพัฒนาเพิ่มเติมอ็อบเจกต์ภายใต้ private(4) enterprises(1) โดยสร้างโหนดชื่อ PSUMIB(99999) ภายใต้ enterprises(1) ซึ่งมีชื่ออ็อบเจกต์ว่า intrustionMIB และมีค่า OID ของอ็อบเจกต์เท่ากับ 1.3.6.1.4.1.99999.2 ดังแสดงในภาพประกอบที่ 2-5



ภาพประกอบ 2-5 โครงสร้างต้นไม้เมื่อเพิ่มกลุ่มอ็อบเจกต์ intrusionMIBใน MIB-II

2. ไวยากรณ์ (Syntax)

เป็นการกำหนดรายละเอียดชนิดของอ็อบเจกต์ โดยใช้ไวยากรณ์ของภาษา ASN.1 (Miller, 1993) ซึ่งสามารถแบ่งกลุ่มชนิดข้อมูลของอ็อบเจกต์เป็น 3 กลุ่ม คือ Primitive Type ที่ใช้เป็นชนิดข้อมูลพื้นฐาน ได้แก่ INTEGER, OCTET STRING, OBJECT IDENTIFIER และ NULL กลุ่มที่สอง คือ Defined Type คือชนิดข้อมูลใหม่ที่กำหนดจากชนิดข้อมูลเดิม ได้แก่ NetworkAddress, IpAddress, Counter, Gauge, TimeTicks และ Opaque และกลุ่มสุดท้ายคือ กลุ่ม Contractor Type สำหรับใช้ในการสร้างลิสต์และสร้างตารางได้แก่ SEQUENCE และ SEQUENCE OF ดังแสดงรายละเอียดชนิดข้อมูลของอ็อบเจกต์ใน SMIv1 และ SMIv2

ตารางที่ 2-1 ชนิดข้อมูลของอ็อบเจกต์ใน SMIv1 และ SMIv2

ชนิดข้อมูล	คำอธิบาย

INTEGER	เป็นตัวเลขจำนวนเต็มบวกหรือลบ และสามารถกำหนดค่าของตัวเลขจำนวนเต็มที่เป็นช่วงได้ เช่น INTEGER (0..255) หรือใช้กำหนดแทนชุดของตัวเลข (Enumerated) ซึ่งจะไม่ใช้ 0 ในการแทนชุดของตัวเลข เช่น ifOperStatus INTEGER {
	up (1)
	down (2)
	testing (3)
	}

OCTET STRING	เป็นสายอักขระของตัวอักษรที่มีขนาด 8 บิต หรือเท่ากับ 1 Octet ตั้งแต่ศูนย์ขึ้นไป โดยที่สามารถกำหนดขนาดได้หลายแบบ เช่น OCTET STRING (SIZE 0..255) OCTET STRING (SIZE 8) OCTET STRING (SIZE 4 | 8) OCTET STRING (SIZE 0..255 | 8)
OBJECT IDENTIFIER	ชุดของตัวเลขจำนวนเต็มฐานสิบที่คั่นด้วยจุด ซึ่งจะใช้แทนตำแหน่งของอ็อบเจกต์ในฐานข้อมูล MIB เช่น 1.3.6.1.2.1.1.6 ที่ใช้แทนอ็อบเจกต์ sysLocation
NULL	ใช้แทนค่าว่าง
NetworkAddress	เป็นตัวเลือกที่ใช้ในการแทนแอดเดรสของชุดโพรโทคอลที่มีอยู่หลากหลาย สำหรับชุดของ TCP/IP ที่ใช้หมายเลขไอพีและชนิดข้อมูลคือ ipAddress
Gauge	เป็นตัวเลขจำนวนเต็มบวกขนาด 32 บิต ซึ่งจะมีค่าอยู่ระหว่าง 0 – 2^{32} (4,294,967,295) โดยที่สามารถเพิ่มหรือลดค่าได้ เช่น ค่าที่ใช้วัดความเร็วในการรับและส่งข้อมูลของ interface
TimeTicks	เป็นตัวเลขจำนวนเต็มบวกขนาด 32 บิต ซึ่งจะมีค่าอยู่ระหว่าง 0 – 2^{32} (4,294,967,295) ใช้ในการวัดค่าเวลาในหน่วยของ 1/100 วินาที เช่น วัดจำนวนของเวลาทั้งหมดที่ระบบเริ่มทำงานมาจนถึงปัจจุบัน

ตารางที่ 2-1 ชนิดข้อมูลของอ็อบเจกต์ใน SMIv1 และ SMIv2 (ต่อ)

ชนิดข้อมูล	คำอธิบาย
Opaque	ให้ชนิดข้อมูลอื่นของ ASN.1 สามารถเข้ารหัสเป็น OCTET STRING

SEQUENCE	ใช้ในการสร้างลิสต์หรือแถวของตาราง ซึ่งมีไวยากรณ์ดังนี้
	SEQUENCE {<type1>,<type2>,…,<typeN>}
	ซึ่ง type คือ ชนิดข้อมูลในกลุ่ม Primitive type เช่น
	ifEntry ::= SEQUENCE {
	ifIndex INTEGER
	ifType INTEGER
	ifSpeed Gauge
	ifSpecific OBJECT IDENTIFIER
	}

SEQUENCE OF	ใช้ในการสร้างตาราง ซึ่งมีไวยากรณ์ดังนี้ SEQUENCE OF <entry> ซึ่ง entry คือ ลิสต์ เช่น ifTable := SEQUENCE OF ifEntry
DisplayString	ใช้แทน OCTET STRING สำหรับแสดงข้อมูลประเภทข้อความที่ผู้ใช้สามารถอ่านเข้าใจได้
PhysAddress	แอดเดรสในระดับ Media หรือ Physical
MacAddress	แอดเดรส MAC ของ IEEE 802 ซึ่งมีความยาวเท่ากับ 6 Octets
Integer32	รายละเอียดเหมือนกับ INTEGER
Counter32	รายละเอียดเหมือนกับ Counter
Unsigned32	ตัวเลขจำนวนเต็มฐานสิบขนาด 32 บิต สามารถแสดงค่าได้ตั้งแต่ $0 – 2^{32}$ (4,294,967,295)
AutonomousType	ค่า OID ที่ใช้สำหรับการกำหนด Subtree ใน MIB
RowPointer	ค่า OID ที่เป็นตัวชี้ไปยังแถวในตาราง
RowStatus	ใช้แทนค่าสถานะของการสร้างและลบแถวในตาราง
StorageType	ใช้สำหรับกำหนดชนิดของหน่วยความจำที่ Agent ใช้

การกำหนดและอธิบายโครงสร้างของอ็อบเจกต์จะประกอบด้วยส่วนสำคัญอยู่ 5 ส่วน คือ Textual Name, Syntax, Definition, Access และ Status ดังแสดงตัวอย่างข้อมูลของ ifNumber ดังภาพประกอบที่ 2-6

```
ifNumber        OBJECT-TYPE
SYNTAX          INTEGER
ACCESS          read-only
STATUS          mandatory
DESCRIPTION
        "The number of network interfaces (regardless of their current
        state) present on this system."
::= { interfaces 1 }
```

ภาพประกอบ 2-6 ตัวอย่างการอธิบายชนิดข้อมูลของอ็อบเจกต์ ifNumber

(McCloghire and Rose, 1991)

- Textual name เป็นชุดของตัวอักษรที่ใช้กำหนด OBJECT DESCRIPTOR ของอ็อบเจกต์ซึ่งจะขึ้นต้นด้วยตัวอักษรตัวเล็ก ในตัวอย่างนี้ คือ sysDescr และมีส่วนของ OBJECT IDENTIFIER ที่สัมพันธ์กัน เพื่อระบุตำแหน่งของอ็อบเจกต์ sysDescr

- **Syntax** เป็นการกำหนดชนิดของข้อมูลของอ็อบเจกต์ sysDescr ซึ่งในตัวอย่างนี้คือ OCTET STRING

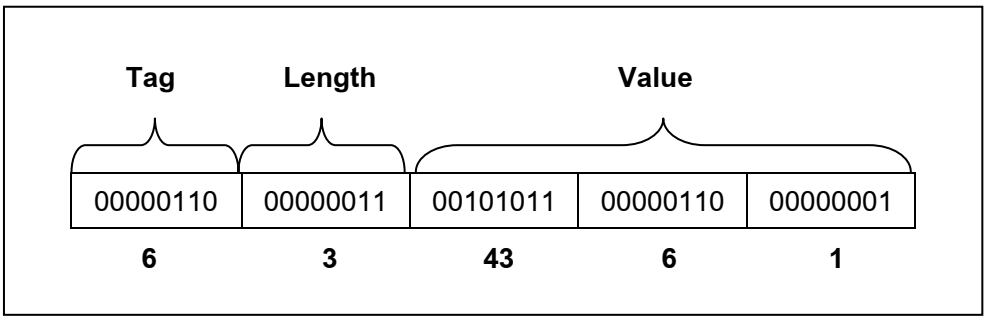
- **Definition** เป็นส่วนที่ใช้ในการอธิบายความหมายของอ็อบเจกต์ sysDescr

- **Access** เป็นส่วนที่ใช้ในการกำหนดสิทธิ์ในการเข้าถึงข้อมูลของอ็อบเจกต์ ซึ่งมี 3 รูปแบบ คือ อ่านได้อย่างเดียว (Read Only) อ่านหรือเขียนได้ (Read Write) และไม่สามารถเข้าถึงได้ (Not Accessible) เช่น อ็อบเจกต์ที่เป็นตาราง

- **Status** เป็นการกำหนดสถานะของอ็อบเจกต์ ซึ่งมี 3 ค่า คือ การกำหนดให้อ็อบเจกต์นี้จำเป็นต้องมี (Mandatory) กำหนดให้อ็อบเจกต์นี้จะมีหรือไม่มีก็ได้ (Optional) และการกำหนดว่าอ็อบเจกต์นี้ถูกเลิกใช้ (Obsolete)

3. การเข้ารหัส (Encoding)

การเข้ารหัสข้อมูลของอ็อบเจกต์นั้นจะใช้วิธีการเข้ารหัสแบบ Basic Encoding Rule ในการส่งข้อมูลไปมาระหว่าง Manager กับ Agent ซึ่งประกอบด้วย 3 ส่วน คือ Tag, Length และ Value หรือเรียกว่า TLV โดยที่ Tag จะใช้ในการกำหนดประเภทของแต่ละชนิดข้อมูลที่จะเข้ารหัส เช่น ชนิดข้อมูล OBJECT IDENTIFIER มีค่า Tag เท่ากับ 00000110_2 (06_{16}) ส่วนฟิลด์ Length จะใช้ในการกำหนดความยาวหรือจำนวนของ OCTET ที่อยู่ในส่วนของ Value โดยกำหนดให้บิตซ้ายสุดมีค่าเป็น 0 ส่วนอีก 7 บิตที่เหลือจะใช้กำหนดความยาว ดังนั้นจึงสามารถใช้กำหนดความยาวของข้อมูลสูงสุดได้ 128 ไบต์ แต่ถ้าข้อมูลมีความยาวมากกว่า 128 ไบต์ บิตซ้ายสุดมีค่าเป็น 1 แล้วใช้อีก 7 บิตที่เหลือกำหนดจำนวนไบต์ที่ใช้ในการกำหนดความยาวที่อยู่ถัดไป ซึ่งเป็นไบต์ที่ใช้กำหนดความยาวข้อมูล และส่วนของฟิลด์ Value จะใช้ในการกำหนดค่าของข้อมูล ตัวอย่างเช่น การเข้ารหัสที่มีข้อมูลแบบ OBJECT IDENTIFIER ของอ็อบเจกต์ internet ที่มีค่าเท่ากับ 1.3.6.1 ซึ่งจะใช้สูตรคณิตศาสตร์เพื่อคำนวณการเข้ารหัส คือ (x * 40) + y โดยที่ x คือ หมายเลขของตัวระบุย่อยตัวที่หนึ่ง ซึ่งในที่นี้คือ 1 และ y คือหมายเลขของตัวระบุย่อยตัวที่สอง ซึ่งในที่นี้คือ 3 ดังนั้นจะได้ในส่วน Tag เท่ากับ 06_{16} ส่วนของ Length เท่ากับ 3 OCTET และส่วนของ Value เท่ากับ 43 6 1 ซึ่งการเข้ารหัสที่ได้จะมีค่าดังภาพประกอบที่ 2-7



ภาพประกอบที่ 2-7 การเข้ารหัสของอ็อบเจกต์ internet (สุทธิพันธ์, 2551)

4. การอ้างถึงค่าข้อมูลของอ็อบเจกต์ (Object Instance)

Object Instance นั้นจะหมายถึงตัวแทนของการอ้างถึงค่าของอ็อบเจกต์ ซึ่งในการอ้างถึงค่าข้อมูลของอ็อบเจกต์ที่ต้องการจาก Agent ทำโดยการระบุชื่อของอ็อบเจกต์และคั่นด้วยจุดกับค่า Object Instance ของอ็อบเจกต์นั้น เช่น ถ้าต้องการข้อมูลของชื่อระบบ (sysName) ก็จะต้องระบุถึง OID และ Instance ได้ดังนี้

Iso	org	dod	internet	mgmt	mib-2	system	sysName	**Instance**
1	3	6	1	2	1	1	5	**0**

ดังนั้นจะได้ตัวแปรในการสอบถามข้อมูลของชื่อระบบ คือ 1.3.6.1.2.1.1.5.0 หรือ sysName.0 ในกรณีที่อ็อบเจกต์ภายในตารางมี Instance ได้มากกว่าหนึ่งค่า วิธีการอ้างถึงข้อมูลที่ตำแหน่งของ Instance นั้นสามารถอ้างถึงข้อมูลได้จากข้อมูลของอ็อบเจกต์ที่ถูกกำหนดให้เป็นอินเด็กซ์ตาราง โดยดูจากรายละเอียดของอ็อบเจกต์ใน MIB เช่น ตาราง ipAddrTable ใน RFC1213 มีอ็อบเจกต์ ipAdEntAddr เป็นอินเด็กซ์ของตาราง

2.12 งานวิจัยที่เกี่ยวข้อง

ในปี ค.ศ. 2005 Lee และคณะ (2005) นำเสนอระบบตรวจจับการบุกรุกแบบที่นำข้อดีของทั้งวิธี Misuse Detection และ Anomaly Detection มาใช้สำหรับตรวจจับการบุกรุก โดยนำเสนอ Multi-step Multi-class Intrusion Detection System (MMIDS) โดยระบบที่นำเสนอนี้ได้ใช้เทคนิค SVM ในการแยกระหว่างเหตุการณ์ปกติหรือการโจมตี และใช้ Fuzzy-Art ซึ่งเป็น Clustering Algorithm เพื่อช่วยในการตรวจจับการโจมตีที่รู้จัก ทำให้มีความรวดเร็วในการแยกรูปแบบการโจมตีที่ไม่รู้จักได้

ในปี ค.ศ. 2008 Bruno และคณะ (2008) ได้นำเสนอการใช้ DSNS (Digital Signature of Network Segment) สำหรับใช้เป็นเครื่องมือในการตรวจจับความผิดปกติของเครือข่าย ซึ่งใช้โพรโทคอล SNMP ในการจัดเก็บข้อมูลของอ็อบเจกต์ใน MIB และหาความสัมพันธ์ของอ็อบเจกต์แต่ละตัว โดยใช้ Dependency Graph ในการอธิบายความสัมพันธ์ของอ็อบเจกต์ต่างๆ เพื่อสร้างโมเดลสำหรับแจ้งเตือนเมื่อพบว่าอ็อบเจกต์ที่ได้ทำการตรวจจับนั้นมีความผิดปกติเกิดขึ้น แต่เนื่องจากมีอัตราความผิดพลาดในการตรวจจับไม่เป็นที่น่าพอใจ

เพราะระบบที่ Bruno นำเสนอนั้นมีข้อจำกัดคือ ระบบไม่สามารถจัดการกับปริมาณ ข้อมูลที่มากในเวลาอันสั้นได้ ในปีต่อมา Bruno และคณะ (2009) จึงได้พัฒนาระบบตรวจจับที่แบ่งการวิเคราะห์ข้อมูลออกเป็น 3 ระดับ คือ Object Level Analysis, Device Level Analysis และ Network Level Analysis เพื่อหวังว่าจะสามารถรายงานเหตุการณ์อย่างละเอียดให้กับผู้ดูแลระบบได้ ซึ่งงานวิจัยนี้สามารถตรวจจับความผิดปกติในระบบอุปกรณ์ และเครือข่ายได้

ในปี ค.ศ. 2009 Lee D.C. และคณะ (2009) ก็ได้นำเสนอการตรวจจับการบุกรุกโดยใช้ MIB อ็อบเจกต์ในกลุ่ม ip และ interface เพื่อใช้ในการหาความผิดปกติบนเครือข่าย อัลกอริทึมที่เขานำเสนอนั้นจะแบ่งการทำงานออกเป็น 2 ขั้นตอน โดยในขั้นตอนแรกคือ จะทำการกรองข้อมูลที่คิดว่าไม่มีความสำคัญออก และขั้นตอนที่ 2 คือ การประมาณค่าของอ็อบเจกต์ในกลุ่ม interface โดยใช้อ็อบเจกต์กลุ่ม ip ในการคำนวณหาค่าของ จำนวนแพ็กเก็ตเข้าและออกของอ็อบเจกต์ interface ซึ่งระบบที่ได้นำเสนอนั้นไม่ทำให้ลดประสิทธิภาพการทำงานของเครื่องคอมพิวเตอร์หรือเซิร์ฟเวอร์ที่ติดตั้งโปรแกรมนี้อยู่ ทำให้มีความรวดเร็วในการวิเคราะห์ข้อมูลหรือความผิดปกติ แต่มีข้อเสียอยู่ตรงที่มี False Positive และ False Negative สูง และในปีเดียวกัน Cup-Mei และ Shandong (2009) นำเสนอระบบตรวจจับการบุกรุกโดยใช้ MIB อ็อบเจกต์ ในการเก็บรวบรวมข้อมูลจาก SNMP Agent เพื่อส่งให้ SVM ทำการวิเคราะห์ความผิดปกติ โดยแบ่งส่วนการวิเคราะห์ออกเป็น 2 ระดับคือ ระดับที่ 1 ในการระบุว่ามีความผิดปกติเกิดขึ้นหรือไม่ และในระดับที่ 2 เป็นการระบุประเภทของการโจมตีที่เกิดขึ้น ซึ่งในระดับที่ 2 นั้นต้องมี Support Vector Data Description (SVDD) ซึ่งใช้เป็นชุดข้อมูลเพื่อใช้ในการระบุชนิดของการบุกรุกด้วย

2.13 บทสรุป

ในบทนี้ได้นำเสนอทฤษฎี หลักการ ที่เกี่ยวข้องกับการทำวิจัยครั้งนี้ ตลอดจนตัวอย่างงานวิจัยที่เกี่ยวข้องทั้งในเรื่องของโพรไฟล์ การตรวจจับการบุกรุกโดยการใช้โพรโตคอล SNMP และ MIB อ็อบเจกต์ในการเก็บรวบรวมและวิเคราะห์ข้อมูล ในบทถัดไปจะนำเสนอในส่วนของการวิเคราะห์ ออกแบบและพัฒนาโพรไฟล์สำหรับใช้ในการตรวจจับการบุกรุกที่เกิดขึ้นบนระบบคอมพิวเตอร์และเครือข่าย

บทที่ 3

การวิเคราะห์ ออกแบบและพัฒนาสถาปัตยกรรมโพรไฟล์

3.1 บทนำ

จากที่กล่าวไว้แล้วในบทที่ 1 ว่าข้อมูลหรืออ็อบเจกต์ใน MIB ที่มีอยู่ไม่เพียงพอต่อการนำมาใช้ตรวจจับการบุกรุกที่เพิ่มขึ้นบนระบบคอมพิวเตอร์และเครือข่าย ดังนั้นผู้วิจัยจึงนำเสนอข้อมูลใหม่ที่จำเป็น ซึ่งจะอยู่ในรูปของอ็อบเจกต์ เพื่อใช้เป็นพารามิเตอร์สำหรับสร้างโพรไฟล์เพื่อตรวจจับความผิดปกติที่เกิดขึ้นโดยใช้โพรโทคอล SNMP และ MIB เป็นเครื่องมือในการนำเสนอข้อมูลดังกล่าว เนื่องจากทั้งโพรโทคอล SNMP และ MIB มีการใช้งานอย่างแพร่หลายและมีการทำงานที่ไม่ซับซ้อน ดังนั้นเนื้อหาในบทนี้จะกล่าวถึงการวิเคราะห์ข้อมูลที่ใช้ในการสร้างโพรไฟล์ ซึ่งได้มาจากการวิเคราะห์การโจมตีที่เกิดขึ้นในบนระบบคอมพิวเตอร์และเครือข่าย

โดยเนื้อหาในส่วนแรกนั้นจะกล่าวถึงการวิเคราะห์คุณลักษณะขององค์กรสำหรับสร้างโพรไฟล์เพื่อนำเสนอชนิดของข้อมูลที่จะนำมาสร้างโพรไฟล์ ต่อมาจะเป็นการอธิบายถึงการจำแนกการโจมตีบนระบบคอมพิวเตอร์และเครือข่าย จากนั้นจะเป็นการจัดอนุกรมวิธานการบุกรุก ต่อจากนั้นจะเป็นอธิบายรายละเอียดของกลุ่มอ็อบเจกต์ที่เพิ่มเติมขึ้นเพื่อใช้สำหรับการวิเคราะห์การบุกรุก ในส่วนสุดท้ายเป็นการนำแนวคิดในการนำข้อมูลที่ได้นำเสนอมาใช้งาน โดยมีรายละเอียดตามลำดับดังนี้

3.2 การวิเคราะห์คุณลักษณะขององค์กรสำหรับสร้างโพรไฟล์

จากที่ได้กล่าวแล้วในบทที่ 2 ว่า คุณลักษณะสำคัญของโพรไฟล์อย่างหนึ่งคือ มีความจำเพาะเจาะจงกับองค์กรหรือหน่วยงาน เนื่องจากคุณลักษณะและการใช้งานเครือข่ายของแต่ละสมาชิกในองค์กรหรือหน่วยงานนั้น มีความแตกต่างกัน ทั้งนี้ขึ้นอยู่กับการลักษณะงานที่ทำ พฤติกรรมการใช้งาน หรือจำนวนผู้ใช้ในแต่ละหน่วยงานหรือองค์กร ส่งผลให้ข้อมูลหรือปริมาณการใช้งานที่เกิดขึ้นในแต่ละองค์กรนั้นมีความแตกต่างกัน

ในการทำการวิจัยครั้งนี้ ผู้วิจัยได้ใช้เครือข่ายการศึกษาของภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่เป็นกรณีศึกษา ซึ่งปัจจัยที่ใช้ในการพิจารณาสำหรับใช้สร้างโพรไฟล์นั้นมีด้วยกัน 2 ปัจจัย คือ คุณลักษณะการทำธุรกรรมของหน่วยงาน และลักษณะการใช้งานเครือข่ายของสมาชิกในหน่วยงาน

1. คุณลักษณะของหน่วยงาน

ภาควิชาวิทยาการคอมพิวเตอร์มีลักษณะการทำธุรกรรมเรื่องการเรียนการสอนทั้งระดับปริญญาตรีและระดับบัณฑิตศึกษา โดยแต่ละปีการศึกษาแบ่งเป็น 2 ภาคการศึกษาหลัก คือภาคการศึกษาที่ 1 เริ่มตั้งแต่เดือนมิถุนายนถึงเดือนตุลาคม และภาคการศึกษาที่ 2 ระหว่าง เดือนตุลาคมถึงเดือนมีนาคม ทั้งนี้อาจจะมีการจัดการศึกษาในภาคฤดูร้อนบ้างในบางปีการศึกษา (ระหว่างเดือนมีนาคมถึงเดือนพฤษภาคม) แต่จำนวนวิชาที่เปิดสอนในภาคการศึกษาฤดูร้อนนี้มีไม่มากนัก ในการจัดการศึกษาหลักนั้น ลักษณะการเรียนการสอนมีทั้งแบบทฤษฎีและปฏิบัติ โดยมีรายละเอียดวิชาที่เปิดสอน (ในปีการศึกษา 2554)[1] ดังนี้ คือ

แผนการสอนประจำภาคการศึกษาที่ 1

ตารางที่ 3-1 แผนการสอนภาคการศึกษาที่ 1

ปริญญาตรีชั้นปีที่ 2			
ประเภทวิชา	รายวิชา	ทฤษฎี	ปฏิบัติ
	Structure Programming	✓	✓
	Computer Logic and Architecture	✓	-
วิชาเลือก	Knowledge Management System	✓	✓
ปริญญาตรีชั้นปีที่ 3			
วิชาบังคับ	Data Structures	✓	✓
	Data Communication and Networking	✓	✓
	Information System Analysis and Design	✓	✓
	Principles of Database System	✓	✓

[1] ปีการศึกษา 2554 เป็นปีการศึกษาที่ผู้วิจัยทำการเก็บข้อมูลการใช้งานเครือข่ายของภาควิชาวิทยาการคอมพิวเตอร์เพื่อนำมาใช้ในการสร้างโพรไฟล์

ตารางที่ 3-1 แผนการสอนภาคการศึกษาที่ 1 (ต่อ)

ปริญญาตรีชั้นปีที่ 3 (ต่อ)			
วิชาบังคับ	Mathematics for Computer Science II	✓	-
วิชาเลือก	Graphics Design and Applied Arts	✓	✓
	Logic Programming and Prolog	✓	✓
	Management Techniques	✓	✓
	Digital Image Processing	✓	✓
ปริญญาตรีชั้นปีที่ 4			
ประเภทวิชา	รายวิชา	ทฤษฎี	ปฏิบัติ
วิชาบังคับ	Project in Computer Science I	-	✓
	Seminar in Computer Science	-	✓
วิชาเลือก	Internet Technology and Applications	✓	✓
	Database Application Management	✓	✓
	Compiler Construction	✓	
	Network Security	✓	✓
	Job Training in Computer	-	-
	Special Topic in Computer Science (Software Testing)	✓	✓

แผนการสอนภาคการศึกษาที่ 2

ตารางที่ 3-2 แผนการสอนภาคการศึกษาที่ 2

ปริญญาตรีชั้นปีที่ 1			
ประเภทวิชา	รายวิชา	ทฤษฎี	ปฏิบัติ
วิชาบังคับ	Fundamentals of Computer Science	✓	✓
	Basic Computer Laboratory	✓	✓
	Fundamental of Digital Systems and Data Communication	✓	-

ตารางที่ 3-2 แผนการสอนภาคการศึกษาที่ 2 (ต่อ)

ปริญญาตรีชั้นปีที่ 2			
ประเภทวิชา	รายวิชา	ทฤษฎี	ปฏิบัติ
	Introduction To Object-Oriented Programming	✓	✓
	Algorithmic Process and Programming	✓	✓
	Computer Organization	✓	-
	Mathematics for Computer Science I	✓	-
	Extracurriculum I	✓	-
วิชาเลือก	Web Programming Techniques	✓	✓
	Business Data Processing and Programming	✓	✓
ปริญญาตรีชั้นปีที่ 3			
วิชาบังคับ	Integrated Laboratory	✓	✓
	Microprocessors and Interfacing	✓	✓
	Operating System	✓	✓
	Software Engineering	✓	✓
	File Organization and Management	✓	✓
วิชาเลือก	Computer Network System	✓	✓
	Simulation	✓	✓
	Database Application Management	✓	✓
	Artificial Intelligence I	✓	-
ปริญญาตรีชั้นปีที่ 4			
วิชาบังคับ	Project in Computer Science II	-	✓
	Seminar in Computer Science	-	✓
วิชาเลือก	Introduction to Computer Graphics	✓	-
	Introduction to Cryptography	✓	-
	Professional Training	✓	-
	SPCS(Problem Solving with Visual Programming)	✓	✓

ตารางที่ 3-2 แผนการสอนภาคการศึกษาที่ 2 (ต่อ)

วิชาที่เปิดสอนให้ภาควิชา/หลักสูตรอื่น			
วิชาบริการสำหรับนักศึกษาจากคณะอื่นๆ มาเรียนที่ภาควิชา	Computers and Applications	✓	✓
	Computer and Programming	✓	✓

ตารางที่ 3-3 สรุปจำนวนนักศึกษาที่เข้าศึกษาในภาควิชาวิทยาการคอมพิวเตอร์

ปีการศึกษา 2554	จำนวน (คน)
บุคลากร	24
นักศึกษา (ปริญญาตรี) ภาควิชา	206
นักศึกษา (ปริญญาโท) ภาควิชา	16
นักศึกษาภาควิชา/คณะอื่นๆ ที่ลงเรียนวิชา Computers and Applications	564

จากตารางที่ 3-1 และ 3-2 จะเห็นได้ว่ามีรายวิชาที่เปิดสอน 70% เป็นวิชาประเภทปฏิบัติการ ซึ่งจะต้องมีการใช้งานเครือข่ายคอมพิวเตอร์ ประกอบกับจำนวนผู้ใช้งานเครือข่ายภาควิชา ฯ ซึ่งปัจจัยทั้งสองนี้จะส่งผลต่อปริมาณข้อมูลในเครือข่าย

2. ลักษณะการใช้งานเครือข่าย

ลักษณะการใช้งานเครือข่ายของภาควิชาวิทยาการคอมพิวเตอร์ขึ้นอยู่กับช่วงเวลาการศึกษา และวิชาที่เปิดสอนในภาควิชา ตามตารางที่ 3-1 และ 3-2 ซึ่งแบ่งเป็น 2 ภาคการศึกษา ส่งผลให้ปริมาณการใช้งานเครือข่ายมีความแตกต่างกันตามระยะเวลาการเปิดและปิดภาคการศึกษาด้วย ระยะเวลาดังกล่าวอ้างตามปฏิทินการศึกษาประจำปี 2554 มีรายละเอียดดังนี้

ภาคเรียนที่ 1 ของปีการศึกษา 2554

- ลงทะเบียนเรียนและเพิ่ม-ถอน เปลี่ยนวิชาเรียน วันที่ 7 กุมภาพันธ์- 20 มีนาคม 2554
- วันเข้าชั้นเรียน วันที่ 6 มิถุนายน 2554
- วันลงทะเบียนเรียนสาย วันที่ 6-12 มิถุนายน 2554
- วันไหว้ครู วันที่ 16 มิถุนายน 2554
- วันสอบกลางภาค 30 กรกฎาคม – 7 สิงหาคม 2554
- สัปดาห์ ม.อ. วิชาการ สัปดาห์ที่ 3 ของเดือนสิงหาคม 2554
- วันสอบไล่ วันที่ 3-14 ตุลาคม 2554
- วันปิดภาคการศึกษา วันที่ 15 ตุลาคม 2554

ภาคเรียนที่ 2 ของปีการศึกษา 2554

- วันลงทะเบียนเรียน เพิ่ม-ถอน เปลี่ยนวิชาเรียน วันที่ 5 กันยายน – 30 ตุลาคม 2554
- วันเข้าชั้นเรียน วันที่ 25 ตุลาคม
- วันลงทะเบียนเรียนสาย วันที่ 25 ตุลาคม – 1 พฤศจิกายน 2554
- วันสอบกลางภาค วันที่ 19 – 29 ธันวาคม 2554
- วันสอบไล่ วันที่ 20 กุมภาพันธ์ – 2 มีนาคม 2554
- วันปิดภาคการศึกษา วันที่ 3 มีนาคม
- มหาวิทยาลัยเป็นเจ้าภาพจัดการแข่งขันกีฬา
  มหาวิทยาลัยแห่งประเทศไทย ครั้งที่ 39 วันที่ 26 เมษายน – 8 พฤษภาคม 2555
  (ไม่มีการเรียนการสอน)

ภาคฤดูร้อน

- ลงทะเบียนเรียนและเพิ่ม-ถอน วันที่ 30 มกราคม – 27 มีนาคม 2555
- วันเข้าชั้นเรียน วันที่ 26 มีนาคม 2555
- วันลงทะเบียนเรียนสาย วันที่ 26 มีนาคม – 1 เมษายน
- วันสอบไล่ วันที่ 14 – 18 พฤษภาคม 2555
- วันปิดภาคการศึกษา วันที่ 21 พฤษภาคม 2555

เนื่องจากผู้ใช้งานเครือข่ายคอมพิวเตอร์ของภาควิชา ฯ ส่วนใหญ่คือนักศึกษาที่ลงทะเบียนรายวิชาของภาควิชา ฯ และการเข้าใช้งานเครือข่ายมีความแตกต่างกันในช่วงเวลาต่างๆ ของแต่ละวันแต่ละเดือน ส่งผลให้ลักษณะข้อมูลในแต่ละวันและแต่ละเดือนนั้นแตกต่างกันด้วย ในการเก็บข้อมูลปริมาณการใช้งานเครือข่ายเพื่อนำมาเป็นข้อมูลวิเคราะห์ความปกติของเครือข่ายนั้น ต้องคำนึงถึงปัจจัยเรื่องช่วงเวลาของการเก็บข้อมูลด้วย เพื่อให้ข้อมูลที่นำมาใช้สามารถเป็นตัวแทนที่ดีในการวิเคราะห์ความปกติได้ ดังนั้นผู้วิจัยจึงได้กำหนดรูปแบบโพรไฟล์ของภาควิชาวิทยาการคอมพิวเตอร์เป็น 3 รูปแบบด้วยกันตามช่วงของเวลาที่เก็บ คือ โพรไฟล์รายวัน รายสัปดาห์ และรายเดือน โดยรายละเอียดของแต่ละรูปแบบมีดังนี้

1. โพรไฟล์รายวัน เก็บข้อมูลการใช้งานเครือข่ายของภาควิชาเพื่อสร้างเป็นโพรไฟล์รายวันของวันจันทร์ถึงวันอาทิตย์ เพื่อใช้เป็นตัวแทนของแต่ละวันทั้ง 7 วัน เนื่องจากสมมุติฐานที่ว่า ปริมาณการใช้งานเครือข่าย (จำนวนผู้เข้าใช้งานเครือข่าย) ของวันจันทร์ไม่เหมือนกับปริมาณการใช้งานเครือข่ายในวันอังคารหรือวันพุธ ฯลฯ เนื่องจากตารางการใช้ห้องปฏิบัติการของแต่ละวันไม่เหมือนกัน

2. โพรไฟล์รายสัปดาห์ เก็บข้อมูลการใช้งานเครือข่ายของภาควิชาเพื่อสร้างเป็นโพรไฟล์รายสัปดาห์ ตั้งแต่สัปดาห์ที่หนึ่งถึงสัปดาห์ที่ 52 ของปีเพื่อใช้เป็นตัวแทนของแต่ละสัปดาห์ โดยมีสมมุติฐานว่า ปริมาณข้อมูลที่เกิดจากการใช้งานในสัปดาห์ที่ Wk_1 ไม่จำเป็นต้องเหมือนกับสัปดาห์ที่ Wk_2 ของปี หากกำหนดให้ Wk_1 ,Wk_2 ... Wk_{52} แทน 52 สัปดาห์ (1 ปี)

3. โพรไฟล์รายเดือน เก็บข้อมูลการใช้งานเครือข่ายของภาควิชาเพื่อสร้างเป็นโพรไฟล์รายเดือน ตั้งแต่เดือนมกราคมถึงเดือนธันวาคม เพื่อใช้เป็นตัวแทนของแต่ละเดือน โดยใช้สมมุติฐานเดียวกันกับข้อ 1 และ 2 นั่นคือปริมาณการใช้งานเครือข่ายในแต่ละเดือนแตกต่างกันไป

3.3 ชนิดข้อมูลโพรไฟล์

ในการตรวจจับความผิดปกติในรูปแบบ Anomaly Detection นั้น เราจำเป็นต้องทราบเหตุการณ์หรือข้อมูลที่ใช้แทนความปกติของการใช้งานหรือเครือข่ายไว้ก่อนแล้วโดยการเก็บบันทึกการใช้งานที่เกิดขึ้นเพื่อไว้เปรียบเทียบกับเหตุการณ์หรือการใช้งานในปัจจุบัน ซึ่งข้อมูลที่จัดเก็บนั้นมีได้หลากหลายทั้งนี้ก็ขึ้นอยู่วัตถุประสงค์ในการใช้ตรวจจับความผิดปกติ

ในงานวิจัยนี้ผู้วิจัยได้เก็บข้อมูลการใช้งานเครือข่ายโดยใช้โพรโทคอล SNMP และ MIB เป็นเครื่องมือในการสอบถามและจัดเก็บข้อมูลดังกล่าว ซึ่งทั้งโพรโทคอล SNMP และ MIB นั้น ต่างก็เป็นเครื่องมือที่ใช้สำหรับจัดการบริหารเครือข่ายที่สะดวก และมีการใช้งานที่ไม่ยุ่งยากซับซ้อน ข้อมูลจาก MIB ที่ผู้วิจัยได้เลือกนำมาใช้ในการเก็บข้อมูลครั้งนี้คือข้อมูลอ็อบเจกต์ ifInOctets ซึ่งก็คือจำนวนแพ็กเก็ตที่เข้ามาผ่านอินเตอร์เฟสที่เราทำการสอบถามข้อมูล และ ifOutOctets ซึ่งคือจำนวนแพ็กเก็ตที่ผ่านออกอินเตอร์เฟสที่เราทำการสอบถามข้อมูล ซึ่งอ็อบเจกต์เหล่านี้เป็นอ็อบเจกต์มาตรฐานที่มีอยู่ใน MIB (Standard MIB) และเป็นอ็อบเจกต์ที่ระบบจัดการเครือข่าย (Network Management System) มักใช้ในการจัดการบริหารเครือข่าย เพราะจะทำให้เห็นภาพรวมปริมาณข้อมูลของการใช้งานของเครือข่ายได้อย่างชัดเจน

3.4 การวิเคราะห์การโจมตีและข้อมูลที่ใช้สำหรับตรวจจับการบุกรุก

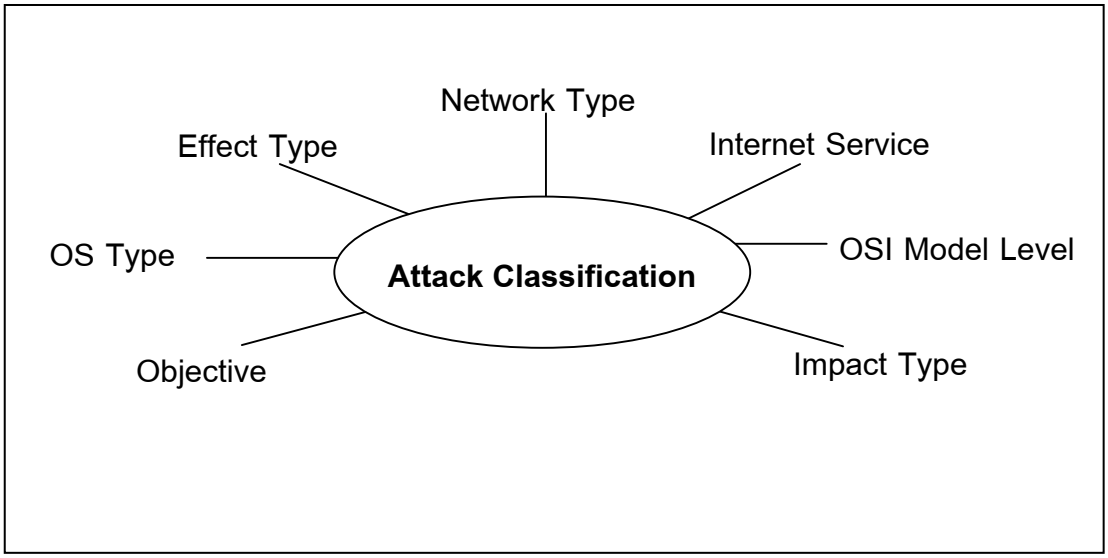
ในการตรวจจับความผิดปกติโดยใช้ข้อมูลหรืออ็อบเจกต์ใน MIB นั่นคือ ifInOctets และ IfOutOctets นั้น ณ เบื้องต้นสามารถทำการตรวจจับความผิดปกติที่เกิดขึ้นในภาพรวมเท่านั้น เพราะเนื่องจากคำนิยามของทั้งสองอ็อบเจกต์นั้นเป็นการมองจำนวนแพ็กเก็ตที่ผ่านเข้า-ออกอินเตอร์เฟสเท่านั้น ไม่สามารถแยกชนิดของข้อมูลได้ว่าเป็นข้อมูลชนิดใดหรือรูปแบบใดทำให้ไม่สามารถระบุชนิดหรือรูปแบบของการกระทำที่ทำให้เกิดความผิดปกติได้ ถึงแม้จะมีบางอ็อบเจกต์ใน MIB มาตรฐานที่สามารถนำไปใช้สำหรับระบุรูปแบบความผิดปกติได้ เช่น ipInReceives จำนวน IP Datagram ที่เข้ามาในเครือข่าย หรือ tcpInSegs จำนวน TCP Segment ที่ผ่านเข้ามาในเครือข่าย เป็นต้น แต่ก็ยังไม่เพียงพอต่อการตรวจจับการโจมตีที่มีเพิ่มมากขึ้นในปัจจุบัน

ถึงแม้การโจมตีที่เกิดขึ้นบนระบบคอมพิวเตอร์และเครือข่ายนั้นจะมีหลากหลายรูปแบบ แต่ก็สามารถนำมาจำแนกประเภทเพื่อบอกคุณลักษณะของการโจมตีเหล่านั้นได้ มีนักวิจัยหลายท่านได้ทำการแบ่งประเภทการโจมตีที่เกิดขึ้นบนระบบคอมพิวเตอร์และเครือข่าย โดยจัดให้อยู่ในรูปแบบที่เรียกว่าอนุกรมวิธาน โดยมีเป้าหมายในการจัดจำแนกจุดอ่อนและรูปแบบของการบุกรุกให้อยู่ในกลุ่มความสัมพันธ์เดียวกัน

ผู้วิจัยจึงได้ศึกษาอนุกรมวิธานที่เกี่ยวกับการโจมตีและการตรวจจับการโจมตี (Specht and Lee, 2003);(Qayyum *et al.*, 2005) (Igure and Williams, 2008) ที่เกิดขึ้นในอดีต

มาใช้เป็นแนวทางในการศึกษาเพื่อหาชนิดข้อมูลที่จะนำมาใช้สำหรับตรวจจับการบุกรุก แต่เนื่องจากชนิดข้อมูลมีจำนวนมากและหลากหลายและไม่มีอยู่ใน MIB มาตรฐาน ผู้วิจัยจึงได้นำเสนอข้อมูลเหล่านั้นให้อยู่ในรูปแบบของอ็อบเจกต์ และเพิ่มเป็นอ็อบเจกต์ใหม่ของ MIB-II เพื่อให้สะดวกต่อการนำไปใช้งานและเพิ่มเติมชนิดข้อมูลได้เมื่อมีการโจมตีใหม่ๆ เกิดขึ้น ซึ่งกลุ่มอ็อบเจกต์เหล่านี้จะนำไปใช้ประกอบกันเพื่อเป็นพารามิเตอร์ในโพรไฟล์สำหรับใช้ตรวจจับการบุกรุกหรือความผิดปกติในระบบตรวจจับการบุกรุกต่อไป

Paulauskas และ Garsva (2006) ได้แยกประเภทของการโจมตีที่เกิดกับระบบคอมพิวเตอร์และเครือข่ายซึ่งจำแนกประเภทการโจมตีที่เกิดขึ้นโดยมีพื้นฐานอยู่บนการวิเคราะห์เพื่อจำแนกการโจมตีที่รู้จักสามารถจำแนกได้หลายคุณลักษณะ ดังภาพประกอบที่ 3-1



ภาพประกอบที่ 3-1 การจำแนกประเภทการโจมตีตามคุณลักษณะต่างๆ
(Paulauskas and Garsva, 2006)

โดยรายละเอียดของแต่ละคุณลักษณะมีดังต่อไปนี้

1. Network Type: เป็นการจำแนกการโจมตีตามตำแหน่งหรือชนิดของเครือข่าย เช่น การโจมตีเครือข่ายแบบ Local System เช่น ใช้วิธีการดักจับรหัสผ่าน (Password Sniffing) (Armstrong, 1996) หรือปลอมหมายเลข MAC (MAC Address Spoofing) (Sasu, 2010) เป็นต้น

- การโจมตีเครือข่ายแบบ Global Network โดยวิธีการทำให้ Routing Table ทำงานผิดพลาด หรืออาศัยช่องโหว่ของโพรโทคอล เป็นต้น โจมตีเครือข่าย Wireless Network เช่น WEP Cracking (Lai *et al.*, 2008) หรือ Sybil Attack (Zhang *et al.*, 2005)

2. Internet Service: เป็นการจำแนกการโจมตีตามบริการต่างๆ ที่มีใช้งานอยู่บนอินเทอร์เน็ต เช่น HTTP Injection, HTTP Session Hijack, FTP Bounce หรือ DNS Flood (Cheng และคณะ, 2010) เป็นต้น

3. OSI Model Level: เป็นการจำแนกการโจมตีตาม OSI Model ซึ่งได้แก่

3.1 Physical Layer จะเกี่ยวข้องกับการส่งข้อมูลในระดับบิตไปยังสื่อที่ใช้ในการส่งข้อมูล ตัวอย่างการโจมตีที่เกิดขึ้นบน Physical Layer ได้แก่ Eavesdropping (Lin *et al.*, 2007) คือการที่ผู้โจมตีสามารถเข้าถึงคลื่นสัญญาณที่มีการสื่อสารกัน เพื่อทำการถอดรหัสข้อมูลที่ส่งไปในคลื่นสัญญาณได้ หรือ Jammer Attack (Salim, 2011) เป็นการสร้างแพ็กเก็ตที่มีปริมาณมากแบบคงที่บนเครือข่ายไร้สาย (Wireless Network) เพื่อให้เกิดสัญญาณรบกวน (Noise) และไม่สามารถเข้าใช้งานเครือข่ายได้ เป็นต้น

3.2 Data Link Layer จะรับข้อมูลมาจาก Physical Layer มีกลไกในการตรวจสอบความผิดพลาดของข้อมูล แต่ถึงแม้จะมีกลไกในการตรวจสอบความผิดพลาดและควบคุมการส่งผ่านข้อมูลแล้ว ก็ยังมีการโจมตีที่เกิดขึ้นบนชั้นนี้ เช่น ในเครือข่ายไร้สายการโจมตีที่เกิดขึ้นคือ Virtual Jamming Attack 802.11 (Salim, 2011) คือผู้โจมตีจะส่งแพ็กเก็ตที่มีการส่งแพ็กเก็ต RST/CTS หรือข้อมูลเป็นจำนวนมากติดต่อกันเพื่อมีปริมาณแพ็กเก็ตเกิดขึ้นบนเครือข่ายจำนวนมากและทำให้เกิดการชนของข้อมูลหรือแพ็กเก็ต RTS/CTS

3.3 Network Layer เป็นการส่งข้อมูลจากต้นทางไปยังปลายทางข้ามเครือข่ายกัน ซึ่งการโจมตีที่เกิดขึ้นจะอาศัยโพรโทคอลที่ทำงานอยู่บน Network Layer เช่น Ping of Death, Ping Flood หรือ Tribe Flood (Keshariya and Foukia, 2010) เป็นต้น ซึ่งผลที่ได้จากการโจมตีเหล่านี้ จะทำให้มีปริมาณแพ็กเก็ตเกิดขึ้นบนเครือข่ายจำนวนมาก ส่งผลให้เครื่องเป้าหมายไม่สามารถให้บริการได้หรืออาจทำให้ทรัพยากรของเครื่องเป้าหมายหมดลง

3.4 Transport Layer จะทำการส่งข้อมูลจากต้นทาง (Source) ไปยังปลายทาง (Destination) ให้ได้อย่างถูกต้อง ซึ่งจะรับผิดชอบในการในการส่งข้อมูลระหว่าง

โพรเซสของต้นทางกับโพรเซสของปลายทาง การโจมตีที่เกิดขึ้นบน Transport Layer นั้นส่วนมากจะอาศัยช่องโหว่ของโพรโทคอลที่ทำงานอยู่บนชั้นนี้ นั่นคือ TCP และ UDP

3.5 Session Layer เป็นเลเยอร์ที่มีการสร้างเซสชันระหว่างเครื่อง เพื่อให้ผู้ใช้สามารถที่จะเชื่อมโยงกับเครื่องอื่นๆ ได้ เช่น การล็อกอินเข้าใช้งานเครื่องระยะไกลในแต่ละครั้ง เป็นต้น

3.6 Presentation Layer เป็นเลเยอร์ที่ช่วยแปลงรูปแบบข้อมูลและแปลข้อมูลเพื่อที่จะให้มีการแลกเปลี่ยนข้อมูลนั้นๆ ในรูปแบบเดียวกัน เช่น การเข้าและถอดรหัสข้อมูล การบีบอัดข้อมูล เป็นต้น

3.7 Application Layer เป็นส่วนที่ทำให้ผู้ใช้สามารถที่จะใช้งานและบริการต่างๆ ที่มีอยู่ในระบบเครือข่ายได้ ซึ่งการโจมตีที่เกิดขึ้นนั้นจะทำให้โปรแกรมประยุกต์ (Application) หรือเซิร์ฟเวอร์ทำงานผิดพลาด เพื่อให้ผู้โจมตีได้มาซึ่งสิทธิในการเข้าใช้งานระบบหรือเครือข่าย

4. Impact Type: โดยจะจำแนกการโจมตีเป็นแบบ Passive และ Active โดยที่การโจมตีแบบ Passive นี้จะไม่ทำให้เกิดการเปลี่ยนแปลงของข้อมูลแต่ยากต่อการตรวจจับ เช่น Sniffer, Wiretap (Communication Security Inc, 2011) และ Eavesdropping (Lin *et al*., 2007) ในขณะที่การโจมตีแบบ Active เป็นการโจมตีที่ทำให้เกิดการเปลี่ยนแปลงของข้อมูลสามารถตรวจจับได้ เช่น ไวรัส หนอนอินเตอร์เน็ต หรือ การโจมตีผ่านทางโพรโทคอลเครือข่าย เช่น UDP Attack, ICMP Attack (Goto and Kojima, 2005) เป็นต้น

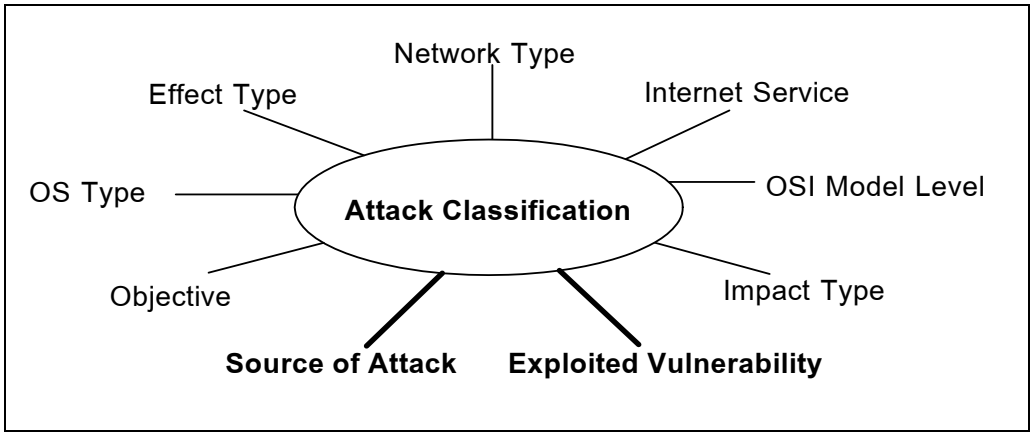
5. Objective: เป็นการโจมตีที่จำแนกตามเป้าหมายของการโจมตี เช่น เพื่อให้ได้มาซึ่งสิทธิในการเข้าใช้งานระบบ เพื่อให้ระบบไม่สามารถทำงานได้ (DoS) เพื่อให้ทรัพยากรของเครื่องเป้าหมายหมดลง เป็นต้น

6. OS Type : เป็นการโจมตีที่จำแนกตามประเภทของระบบปฏิบัติการ เนื่องจากมีระบบปฏิบัติการหลายชนิดที่ใช้งานกันอยู่บนระบบคอมพิวเตอร์และเครือข่าย ไม่ว่าจะเป็น Windows, Linux หรือ Solaris เป็นต้น ซึ่งแต่ละระบบปฏิบัติการนั้นก็ล้วนแล้วแต่มีช่องโหว่ที่ผู้โจมตีสามารถใช้ให้เกิดประโยชน์ในการโจมตีได้

7. Effect Type: เป็นการจำแนกการโจมตีตามประเภทของผลกระทบที่เกิดขึ้นจากการโจมตี เช่น การโจมตีประเภท IP Fragment Flood (Ziemba, 1995) ซึ่งใช้ประโยชน์จากการส่งแพ็กเก็ต IP โดยทำการแบ่งแพ็กเก็ตเป็นขนาดเล็กและส่งไปยังเครื่องเป้าหมายใน

ปริมาณที่มากอย่างรวดเร็วทำให้เครื่องเป้าหมายเกิด Buffer Overflow หรือการใช้ TCP Scan (Gadge and Patil, 2008) เพื่อทำการหาพอร์ตที่เครื่องเป้าหมายเปิดใช้อยู่ เพื่อใช้สำหรับเป็นข้อมูลในการโจมตีต่อไป เป็นต้น

จากการที่ผู้วิจัยได้ทำการศึกษามาพบว่า สามารถนำเสนอประเภทกลุ่มการโจมตีเพิ่มเติมได้อีก 2 กลุ่ม คือ Source of Attack และ Exploited Vulnerability ดังภาพประกอบที่ 3-2 โดยมีรายละเอียดดังนี้



ภาพประกอบที่ 3-2 ประเภทกลุ่มการโจมตีที่เพิ่มขึ้นตามคุณลักษณะต่างๆ

Exploited Vulnerability: เป็นการจำแนกการโจมตีที่ใช้ประโยชน์จากช่องโหว่ต่างๆ โดยวิธีการ Brute Force Attack (Mirkovic, 2004) ซึ่งสามารถแยกได้เป็น Filterable Attack และ Non-Filterable Attack โดยที่ Filterable Attack จะทำการปลอมแปลงแพ็กเก็ต หรือใช้แพ็กเก็ตส่งไปยังเครื่องเป้าหมายเพื่อให้เครื่องเป้าหมายทำการดำเนินการที่ไม่สำคัญ (Non-Critical Services) ส่วน Non-Filterable Attack เป็นการใช้ประโยชน์จากแพ็กเก็ตที่ทำหน้าที่ร้องขอบริการ (Request) ไปยังเครื่องที่ให้บริการโดยจะส่งคำร้องขอเป็นจำนวนมากไปยังเครื่องที่ให้บริการ เช่น HTTP Request Flood การส่ง HTTP Request จำนวนมากไปยัง Web Server หรือ DNS Flood เป็นการส่ง DNS Request จำนวนมากไปยัง Name Server นอกจากนี้ยังมีการโจมตีที่อาศัยประโยชน์จากโพรโทคอลอื่นๆ เช่น TCP SYN Attack, CGI Request Attack และ Authentication Server Attack (Noureldien, 2011)

Source of Attack: แหล่งที่มาของการโจมตีนั้นมีในลักษณะ One-to-One เป็นการโจมตีที่มาจากเครื่องหนึ่งไปยังเครื่องเป้าหมายหนึ่ง Many-to-One เป็นการโจมตีที่มีการโจมตีมาจากหลายแหล่งเพื่อโจมตีไปยังเครื่องเป้าหมายเดียวกัน และ One-to-Many เป็นการโจมตีที่มีมาจากแหล่งโจมตีเดียวทำการโจมตีไปยังหลายเป้าหมาย

จากการจำแนกคุณลักษณะของการโจมตีข้างต้น ผู้วิจัยนำมาจัดทำอนุกรมวิธานโดยใช้แนวคิดของ Simon (Simon, 2005) นั่นคือการจัดทำอนุกรมวิธานโดยใช้แนวคิดของการแบ่งเป็นมิติ (Dimension) สำหรับอนุกรมวิธานนี้ได้กำหนดให้มีการจัดหมวดหมู่ 4 มิติ ได้แก่ First Dimension, Second Dimension, Third Dimension และ Forth Dimension แต่ละมิติมีรายละเอียดดังนี้

- First Dimension: เป็นการจัดกลุ่มการบุกรุกตามชนิดของการบุกรุก ตัวอย่างกลุ่มการบุกรุกในมิตินี้ได้แก่ ไวรัส หนอนอินเทอร์เน็ต โปรแกรมโทรจัน ปัญหาการล้นของบัฟเฟอร์ การปฏิเสธการให้บริการ การโจมตีเครือข่าย การโจมตีรหัสผ่าน การโจมตีทางกายภาพ การขโมยข้อมูล เป็นต้น

- Second Dimension: เป็นการอธิบายถึงเป้าหมายของการบุกรุก ซึ่งแบ่งออกเป็น 2 ระดับ ได้แก่ ระดับฮาร์ดแวร์ และระดับซอฟต์แวร์ ซึ่งมีรายละเอียดดังนี้

1. **เป้าหมายระดับฮาร์ดแวร์**

เป็นการบุกรุกในระดับกายภาพ เช่น การโจมตีอุปกรณ์เครือข่าย หรืออุปกรณ์ภายในคอมพิวเตอร์เพื่อทำลายข้อมูลที่บันทึกไว้ เป็นต้น

2. **เป้าหมายระดับซอฟต์แวร์**

เป้าหมายการบุกรุกในระดับซอฟต์แวร์แบ่งออกเป็นส่วนย่อย ได้แก่ การบุกรุกระบบปฏิบัติการ หรือโปรแกรมเซิร์ฟเวอร์ และการบุกรุกผ่านโปรแกรมประยุกต์ซึ่งรวมไปถึงการบุกรุกผ่านจุดอ่อนของโพรโทคอลเครือข่ายด้วย

- Third Dimension: เป็นส่วนของการอธิบายจุดอ่อนของระบบ เนื่องจากจุดอ่อนของระบบปฏิบัติการหนึ่งจุด ผู้บุกรุกสามารถหาวิธีการโจมตีระบบได้มากกว่าหนึ่งวิธีการ ดังนั้นเมื่อตรวจพบจุดอ่อนในระบบจะต้องกำหนดกลุ่มของจุดอ่อนที่เกิดขึ้นให้ชัดเจน สำหรับการนิยามจุดอ่อนของอนุกรมวิธานนี้ มีสองแนวคิดคือ ในกรณีที่เป็นจุดอ่อนที่เคยเกิดขึ้นมาแล้วให้ใช้ระบบการอ้างอิงจุดอ่อนจาก Common Vulnerabilities and Exposures (CVE) ซึ่งเป็นการ

รวบรวมและเรียบเรียงรายชื่อของจุดอ่อนในระบบคอมพิวเตอร์และเครือข่ายที่เคยเกิดขึ้นโดย บริษัท Mitre (1999) ของแต่ถ้าเป็นจุดอ่อนที่เกิดขึ้นใหม่ให้นิยามจุดอ่อนดังกล่าวตามแนวคิดของ Howard (1997) คือนิยามองค์ประกอบของการจำแนกการบุกรุกเป็น 5 ส่วน ซึ่งได้แก่

1. Attack หมายถึงระดับของบุคคลที่พยายามบุกรุกระบบ

2. Tools หมายถึงเครื่องมือที่ผู้บุกรุกเลือกใช้

3. Access หมายถึงช่องทางหรือจุดอ่อนที่ผู้บุกรุกเลือกใช้ในการเข้าสู่ระบบ

4. Result หมายถึงผลลัพธ์ที่จะเกิดขึ้นเมื่อการบุกรุกประสบความสำเร็จ

5. Objective หมายถึงเป้าหมายของการบุกรุกระบบ

- **Forth Dimension:** เนื่องจากการบุกรุกระบบอาจจะมีมากกว่าหนึ่งช่องทาง การจัดหมวดหมู่ในกลุ่มนี้จึงเป็นการนิยามวิธีการบุกรุกเพิ่มเติมจากมิติที่ 1 เช่น หนอนอินเทอร์เน็ตได้สั่งงานโปรแกรมโทรจันไว้เพื่อรอรับการเชื่อมต่อที่ไม่มีกระบวนการตรวจสอบตัวตน ซึ่งพบว่านอกจากการก่อกวนระบบเครือข่ายด้วยหนอนอินเทอร์เน็ตแล้ว ยังก่อกวนเครื่องเป้าหมายด้วยโปรแกรมโทรจันอีกด้วย จากที่กล่าวมาข้างต้น การจำแนกในมิติที่ 1 เป็นการจัดเหตุการณ์ดังกล่าวเป็นการบุกรุกด้วยหนอนอินเทอร์เน็ต และระบุว่าเหตุการณ์นั้นเป็นการบุกรุกด้วยโปรแกรมโทรจันในมิติที่ 4

จะเห็นได้ว่าการจัดอนุกรมวิธานในแนวคิดของ Simon นั้นจะทำให้เห็นถึงช่องทางการบุกรุก จุดอ่อนของระบบ หรือผลกระทบอื่นๆ ที่ได้มาจากการโจมตีแต่ละชนิดได้อย่างชัดเจน ในการศึกษารูปแบบการบุกรุกบนระบบคอมพิวเตอร์และเครือข่ายครั้งนี้ ผู้วิจัยได้ศึกษารูปแบบการโจมตีที่เกิดขึ้นเพียงบางส่วนเท่านั้น เนื่องจากการศึกษารูปแบบการบุกรุกที่เกิดขึ้นนั้นพบว่า ถึงแม้การโจมตีแต่ละประเภทมีชื่อเรียกที่แตกต่างกัน แต่จะมีวิธีการที่ใช้ในการโจมตีหรือรูปแบบการโจมตีที่คล้ายกัน ดังนั้นผู้วิจัยจึงเลือกรูปแบบการโจมตีบางประเภทมานำเสนอเพื่อใช้เป็นตัวแทนของการโจมตีที่มีรูปแบบที่คล้ายกัน โดยผลที่ได้จากการศึกษาครั้งนี้แสดงให้เห็นดังตารางที่ 3-4

ตารางที่ 3-4 การจัดอนุกรมวิธานการบุกรุกระบบคอมพิวเตอร์และเครือข่าย

Attack Type	1st dimension	2nd dimension	3rd dimension	4th dimension
Virus	Chernobyl (Rouse, 2005)	File infector virus	MS Windows 95 & 98	Corruption of information
	Disk killer (Curio Lab, 2008)	Booth sector virus	Hard disk	Destroys the information
	Michelangelo (Cert, 1997)	System boot record infector virus, DoS family	Boot sector	Corruption of information
	Stone (F-Secure, 2009)	Boot virus	Boot sector	-
Worm	Blaster (Symantec, 2003)	Network-aware worm	MS Windows NT 4.0,2000, XP,Server 2003 CVE-2006-0352	TCP packet flooding DoS, Buffer overflow
	Code Red (Symantec, 2001)	Network-aware worm	IIS4, 5 & 6.0 beta CVE-2001-0500	Stack buffer overflow, TCP packet flooding
	Melissa (Symantec, 2007)	Mass- mailing worm	MS word 97 & 2000	Macro virus& TCP packet flooding
	Nimda (Symantec, 2007)	Mass- mailing worm	Windows 95/98,2000,ME MS IE 5.5 SP1 ,IIS CVE-2001-0333 & CVE-2001-0154	File infector virus, Trojan and DoS

ตารางที่ 3-4 การจัดอนุกรมวิธานการบุกรุกระบบคอมพิวเตอร์และเครือข่าย (ต่อ)

Attack Type	1st dimension	2nd dimension	3rd dimension	4th dimension
	Ramen (Symantec, 2007)	Network-aware worm	RedHat Linux 6.2 &7.0	Host-based DoS, UDP and TCP packet flooding
	Sasser (Symantec, 2004)	Network-aware worm	Windows NT,XP,2000 CVE-2003-0533	Stack buffer overflow
	Slammer(Symantec, 2003)	Network-aware worm	MS SQL Server 2000 CVE-2002-0649	Stack buffer overflow, UDP packet flooding DoS
Password attack	John the Ripper (Peslyak, 2011)	Guessing password	Unix family, Windows NT,2000 & XP	Disclosure of information
	Password sniffing (Armstrong, 1996)	Sniffing	-	Packet sniffing
Information gathering attack	TCP port scan (Gadge และ Patil, 2008)	Port scanning	Protocol	TCP flooding
	Xmas scan (Vizzarro, 2007)	Port scanning	Protocol	TCP flooding

ผลจากการศึกษาการตรวจจับและรูปแบบของการโจมตีที่เกิดขึ้นบนระบบคอมพิวเตอร์และเครือข่ายคือ ข้อมูลที่สามารถนำมาใช้เป็นพารามิเตอร์เพื่อตรวจจับการบุกรุกที่เกิดขึ้นได้ ซึ่งข้อมูลที่ได้พิจารณาจาก ชนิดของข้อมูลและปริมาณที่เปลี่ยนแปลงไปเมื่อเกิดการโจมตี เช่น การโจมตีรูปแบบ SYN Flood (Haris, 2010) เป็นการโจมตีโดยการส่งแพ็กเก็ต TCP ที่มีการตั้งค่า Flag เป็น SYN ไปยังเครื่องเป้าหมายพร้อมทั้งปลอมแปลงหมายเลข IP ต้นทาง (Source) เพื่อให้เครื่องเป้าหมายส่ง SYN-ACK กลับมายังหมายเลข IP ที่ได้ปลอมไว้ หากมีการส่งแพ็กเก็ต TCP ที่ตั้งค่า Flag เป็น SYN จำนวนมากจะทำให้เครื่องเป้าหมายมีภาระงานที่มากขึ้นหรือไม่อาจให้บริการได้ตามปกติ เป็นต้น

จากการศึกษารูปแบบการโจมตีนี้ทำให้เราทราบชนิดข้อมูลที่ใช้สำหรับตรวจจับการโจมตีรูปแบบนี้คือ ปริมาณแพ็กเก็ต TCP ที่ตั้งค่า Flag เป็น SYN และเนื่องจากในการเชื่อมต่อของ TCP แต่ละครั้งนั้น จะมีความสัมพันธ์ของ Flag เกิดขึ้น นั่นคือ SYN จะเกิดคู่กับ FIN และ SYN/ACK ก็จะเกิดคู่กับ FIN ด้วย ดังนั้นชนิดข้อมูลที่ใช้ในการตรวจจับการโจมตีนี้คือ จำนวนแพ็กเก็ต TCP ที่ตั้งค่า Flag เป็น SYN และจำนวนแพ็กเก็ต TCP ที่ตั้งค่า Flag เป็น FIN

ผลที่ได้จากการโจมตีข้างต้นจะทำให้มีปริมาณการใช้งานเครือข่ายที่สูงกว่าปกติ ซึ่งโดยปกติแล้วเราสามารถตรวจจับความผิดปกติที่เกิดขึ้นนี้ได้โดยดูจากค่าข้อมูลของอ็อบเจกต์ ifInOctets และ ifOutOctets นั่นคือดูปริมาณแพ็กเก็ตที่เข้าและออกผ่านอินเตอร์เฟสแต่จะไม่สามารถบอกได้ว่า ความผิดปกติที่เกิดขึ้นนั้น เป็นความผิดปกติรูปแบบใด ด้วยเหตุนี้เองผู้วิจัยจึงได้ทำการวิเคราะห์รูปแบบการการโจมตีและการตรวจจับ เพื่อหาข้อมูลที่สามารถใช้ระบุความผิดปกติที่เกิดขึ้นบนเครือข่ายได้ เพื่อจะได้หาแนวทางในการป้องกันได้ตรงกับประเด็นปัญหาที่เกิดขึ้น

หัวข้อถัดไปจะเป็นการอธิบายถึงการได้มาซึ่งข้อมูลหรืออ็อบเจกต์ที่สามารถนำมาสร้างโพรไฟล์เพื่อใช้ในการตรวจจับการบุกรุกบนเครือข่ายได้ ซึ่งรายละเอียดมีดังต่อไปนี้

3.4.1 การวิเคราะห์ข้อมูลที่ใช้สำหรับตรวจจับการบุกรุก

ผลที่ได้จากการจัดทำอนุกรมวิธานข้างต้น ทำให้เห็นหมวดหมู่ของการโจมตีได้อย่างชัดเจน ซึ่งในการวิเคราะห์ข้อมูลสำหรับใช้ในการตรวจจับการบุกรุกครั้งนี้ ผู้วิจัยได้มุ่งเน้นการตรวจจับความผิดปกติที่เกิดขึ้นบนเครือข่าย ข้อมูลที่ได้จากการวิเคราะห์นั้นส่วนใหญ่จะได้มาจากข้อมูลที่เกิดขึ้นบนเครือข่าย โดยมีหลักการในการวิเคราะห์ดังนี้

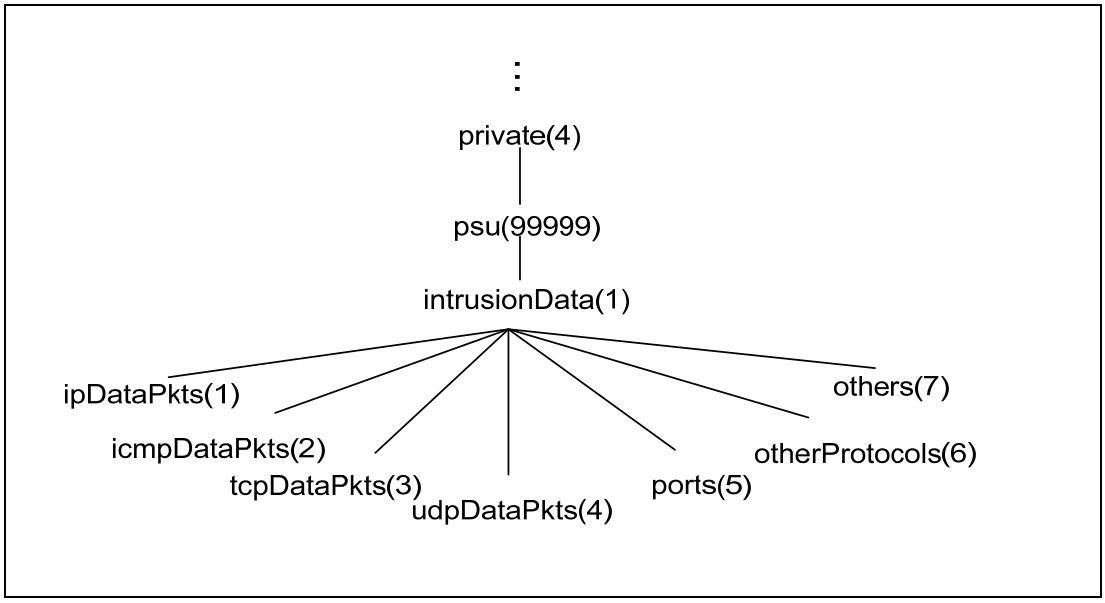
1. มองภาพรวม

จากการศึกษาพบว่าโดยทั่วไปแล้วการโจมตีที่เกิดขึ้นบนเครือข่ายนั้น มักจะอาศัยจุดอ่อนหรือช่องโหว่ของเครือข่ายและโพรโทคอลที่ใช้งานอยู่บนเครือข่าย ซึ่งโดยส่วนใหญ่จะใช้โพรโทคอล IP ICMP TCP และ UDP ซึ่งเป็นโพรโทคอลหลักในการติดต่อสื่อสาร และโดยส่วนใหญ่ผู้โจมตีก็มักจะใช้ประโยชน์จากโพรโทคอลเหล่านี้ในการโจมตีเครือข่าย เช่น Teardrop Attack (Dax Network, 2003) ซึ่งใช้ประโยชน์จากโพรโทคอล IP โดยการปลอมแปลงแพ็กเก็ต IP และส่งไปยังเครื่องเป้าหมายเพื่อให้เครื่องเป้าหมายเกิดความสับสน หรือมีข้อมูลชนกันเกิดขึ้น การโจมตีรูปแบบนี้จะคล้ายกับ Ping of Death แต่ต่างกันตรงที่ Ping of Death นั้นจะใช้โพรโทคอล ICMP เป็นต้น และในปัจจุบันมีการใช้งานเครือข่ายคอมพิวเตอร์เป็นจำนวนมาก และโพรโทคอลที่ใช้งานกันอยู่ในปัจจุบันก็มีหลายโพรโทคอลด้วยกัน เช่น Address Resolution Protocol (ARP) Internet Protocol Security (IPSec) หรือ Internet Protocol version6 (IPv6) เป็นต้น และเนื่องจากผู้วิจัยต้องการนำเสนอข้อมูลที่นำมาใช้ในการตรวจจับการโจมตีเหล่านี้ในรูปแบบอ็อบเจกต์ของ MIB เพื่อให้สอดคล้องกับรูปแบบการสร้างข้อมูล MIB เบื้องต้นผู้วิจัยจึงได้นำเสนอโหนดใหม่ของ MIB โดยแบ่งตามชนิดของโพรโทคอลดังนี้ ipDataPkts, icmpDataPkts, tcpDataPkts, udpDataPkts และ otherProtocols

ในการติดต่อสื่อสารกันของผู้ให้บริการ (Server) และผู้ร้องขอบริการ (Client) ต่างๆ บนเครือข่ายแต่ละครั้งนั้น จะต้องอาศัยช่องทางในการติดต่อสื่อสาร หรือที่เรียกว่าพอร์ต (Port) โดยที่พอร์ตคือหมายเลขเพื่อใช้อ้างถึงในขณะส่งข้อมูลกันระหว่าง Client และ Server ในบางการโจมตีที่เกิดขึ้นนั้นจะอาศัยหมายเลขพอร์ตเป็นช่องทางในการโจมตี เช่น HTTP Injection หรือ DNS Flood (Cheng *et al.*, 2010) เป็นต้น ดังนั้นผู้วิจัยจึงเพิ่มกลุ่มอ็อบเจกต์ขึ้นอีกหนึ่งกลุ่มคือ ports

นอกจากนี้ยังมีผลกระทบอื่นๆ เกิดขึ้นเมื่อเกิดการโจมตี เช่น มีการใช้หน่วยความจำหรือหน่วยประมวลผลที่เพิ่มมากขึ้นกว่าปกติ เป็นต้น ผู้วิจัยจึงได้ตั้งกลุ่มอ็อบเจกต์ที่มีชื่อว่า others ขึ้น เนื่องจากสามารถใช้ข้อมูลในกลุ่มนี้ในการหาความผิดปกติอื่นๆ นอกเหนือจากชนิดโพรโทคอลและพอร์ตได้

จากที่ได้กล่าวมาแล้วข้างต้นสรุปว่า กลุ่มอ็อบเจกต์ที่ผู้วิจัยได้ทำการแบ่งครั้งนี้มีด้วยกัน 7 กลุ่ม คือ ipDataPkts icmpDataPkts tcpDataPkts udpDataPkts ports otherProtocols และ others โดยจะเพิ่มกลุ่มอ็อบเจกต์เหล่านี้ภายใต้โหนด intrusionData และเรียก MIB Object กลุ่มนี้ MIB+ ดังภาพประกอบที่ 3-3



ภาพประกอบที่ 3-3 โครงสร้างต้นไม้ของกลุ่มอ็อบเจกต์ภายใต้ intrusionData(1)

2. เจาะประเด็น

ขั้นตอนต่อไป จะเป็นการหาสมาชิกของแต่ละกลุ่มอ็อบเจกต์ ผู้วิจัยได้ศึกษาเอกสารและงานวิจัยที่เกี่ยวข้องกับการโจมตีระบบคอมพิวเตอร์และเครือข่าย และวิธีการตรวจจับการโจมตีรูปแบบต่างๆ จากการศึกษาพบว่าในการโจมตีเครือข่ายนั้น บางการโจมตีจะใช้ข้อมูลในการโจมตีที่คล้ายคลึงกัน เช่น UDP Flood (Rui *et al.*, 2009) และ Fraggle Attack (Zargar and Kabiri, 2009) ทั้งสองการโจมตีนี้จะใช้ประโยชน์จากโพรโทคอล UDP เหมือนกัน หรือ Smurf Attack (Zargar and Kabiri, 2009) และ Ping of Death (Choundhary and Shilpa, 2011) ที่ใช้โพรโทคอล ICMP ในการโจมตีเหมือนกัน เป็นต้น

เพื่อให้เห็นถึงที่มาของสมาชิกในแต่ละกลุ่มของอ็อบเจกต์ ผู้วิจัยได้อธิบายรายละเอียดของสมาชิกในแต่ละกลุ่มดังนี้

1. การวิเคราะห์หาข้อมูลในกลุ่ม ipDataPkts

จากการศึกษาการโจมตีที่ใช้ประโยชน์จากโพรโทคอล IP (Yang, 1997); (CERT,2001) พบว่า โดยส่วนใหญ่ผู้โจมตีจะใช้ประโยชน์จากโพรโทคอล IP โดยการปลอมแปลงหมายเลข IP ต้นทางให้เหมือนกับหมายเลข IP ปลายทาง เพื่อให้เครื่องเป้าหมายส่งข้อตอบกลับเข้าหาเครื่องตัวเอง (CERT CA-1997-28, 1997) เช่น Land Attack นอกจากนี้ผู้โจมตียังสามารถใช้ประโยชน์จากการส่งแพ็กเก็ต IP แบบ Broadcast ได้อีกด้วยถ้าหากมีการส่งแพ็กเก็ตประเภทนี้จำนวนมากจะเป็นการเพิ่มภาระงานให้แก่เครื่องเป้าหมาย (Choundhary and Shilpa, 2011) ตัวอย่างการโจมตีโดยใช้ IP แบบ Broadcast เช่น Smurf Attack เป็นต้น

ผู้โจมตียังสามารถใช้ประโยชน์จากขนาดของแพ็กเก็ต IP ได้อีกด้วย โดยจะแบ่งแพ็กเก็ต (Fragmentation) (Anderson, 2001) ให้มีขนาดเล็กกว่า MTU เพื่อเพิ่มปริมาณข้อมูลในการส่งไปยังเครื่องเป้าหมาย ซึ่งผลกระทบที่ได้คือ มีปริมาณข้อมูลในเครือข่ายจำนวนมาก ตัวอย่างการโจมตีที่ใช้ประโยชน์จากการแบ่งแพ็กเก็ตคือ Overlapping Fragment Attack หรือ IP Fragment Overrun เป็นต้น

ดังนั้น อ็อบเจกต์ในกลุ่ม ipDataPkts มีสมาชิกดังนี้

ipInBroadcast	คือ จำนวนแพ็กเก็ต IP ที่มีการส่งแบบ Broadcast
ipInAddSrcDest	คือ จำนวนแพ็กเก็ต IP ที่มีหมายเลข IP ต้นทางเหมือนกับหมายเลข IP ปลายทาง
ipInSizePkts	คือ ขนาดของแพ็กเก็ต IP

2. การวิเคราะห์หาข้อมูลในกลุ่ม icmpDataPkts

ผลจากการศึกษาการโจมตีที่ใช้ประโยชน์จากโพรโทคอล ICMP (Lee, 1999) (Carl and Kesidis, 2006) โดยส่วนใหญ่จะประโยชน์จากประเภทของโพรโทคอล ICMP คือ Echo และ Reply ซึ่งโดยปกติแล้วหน้าที่หลักของโพรโทคอล ICMP คือแจ้งหรือแสดงข้อความจากระบบ เพื่อบอกให้ผู้ใช้ทราบว่าเกิดอะไรขึ้นในการส่งผ่านข้อมูล ซึ่งปัญหาส่วนใหญ่คือ ส่งข้อมูลไม่ได้ หรือปลายทางไม่ได้รับข้อมูล เป็นต้น ผู้โจมตีจะส่งแพ็กเก็ต ICMP ประเภท ICMP Echo Request จำนวนมากไปเครือข่ายหรือเครื่องเป้าหมาย แล้วเครื่องเป้าหมายก็จะตอบกลับมาด้วย ICMP Echo Reply ซึ่งหากเป้าหมายได้รับ ICMP Echo Request จำนวนมากอาจจะทำให้หน่วยประมวลผลทำงานมากเกินไป หรือทำให้หน่วยความจำเต็ม จนเครื่องไม่สามารถทำงานได้ตามปกติ นอกจากนี้ผู้โจมตียังสามารถใช้ประโยชน์จากขนาดของโพรโทคอล ICMP ในการโจมตีได้อีกด้วย ตัวอย่างการโจมตีได้แก่ Ping of Death Ping Flood หรือ Smurf Attack เป็นต้น

ดังนั้น อ็อบเจกต์ในกลุ่ม icmpDataPkts มีสมาชิกดังนี้

icmpInEchoPkts	คือ จำนวนแพ็กเก็ต ICMP Echo Request
icmpInPkts	คือ จำนวนรวมของแพ็กเก็ต ICMP
icmpInReplyPkts	คือ จำนวนแพ็กเก็ต ICMP Echo Reply
icmpInSizePkts	คือ ขนาดของแพ็กเก็ต ICMP

3. การวิเคราะห์หาข้อมูลในกลุ่ม tcpDataPkts

โพรโทคอล TCP (Forouzan, 2006) เป็นโพรโทคอลที่ใช้ในการสื่อสารผ่านเครือข่าย โดยมีวัตถุประสงค์เพื่อให้สามารถสื่อสารจากต้นทางไปยังปลายทางได้ โดยโพรโทคอลจะรับประกันความถูกต้องในการรับ-ส่งข้อมูลอีกด้วย และเนื่องจาก Header ของ TCP นั้นมีส่วนที่เรียกว่า Control Bits ซึ่งมีความสำคัญในการกำหนดการทำงานของ TCP ทำหน้าที่ควบคุมอัตราการไหลของข้อมูล การสร้างการติดต่อ การยกเลิกการติดต่อ และวิธีที่ใช้ในการส่งข้อมูล โดยส่วนใหญ่ผู้โจมตีจะใช้ประโยชน์จาก Control Bits นี้ในการสร้างการโจมตี ซึ่งรายละเอียดของและความหมายของแต่ละ Control Bits มีดังนี้

- URG ใช้บอกความหมายว่าเป็นข้อมูลด่วนและมีข้อมูลพิเศษมาด้วย โดยระบบจะทำการประมวลผลแพ็กเก็ตนี้อย่างเร็วที่สุด
- ACK แสดงว่าข้อมูลในฟิลด์ Acknowledge Number นำมาใช้งานได้
- PSH แจ้งให้ผู้รับทราบว่าควรจะส่งข้อมูล Segment นี้ไปยังแอพพลิเคชั่นที่กำลังรออยู่โดยเร็ว
- RST ใช้สำหรับยกเลิกการติดต่อ (Reset) เนื่องจากในกรณีที่เกิดความสับสนขึ้นด้วยเหตุต่าง ๆ เช่น โฮสต์มีปัญหา ให้เริ่มสื่อสารใหม่
- SYN ใช้สำหรับขอเริ่มต้นการติดต่อกับปลายทาง
- FIN ใช้สำหรับแจ้งให้ปลายทางทราบว่ายุติการเชื่อมต่อ

สำหรับการโจมตีที่ใช้ประโยชน์จาก Control Bits ได้แก่ RST and FIN Attack (Krawetz, 2007) Xmas Scan (Gadge and Patil, 2008) หรือ SYN Flood Attack (Haris *et al.*, 2010) เป็นต้น นอกจากนี้ยังมีการโจมตีบางประเภทที่ไม่มีการตั้งค่า Flag (No Flag Set) เช่น Null Scan ซึ่งผู้โจมตีจะส่งแพ็กเก็ต TCP ที่ไม่ตั้งค่า Flag ไปยังเครื่องเป้าหมายเพื่อสแกนพอร์ตของเครื่องเป้าหมาย เป็นต้น

ดังนั้น อ็อบเจกต์ในกลุ่ม tcpDataPkts มีสมาชิกดังนี้

tcpInAckPkts	คือ จำนวนแพ็กเก็ต TCP ที่มี Flag เป็น ACK
tcpInFinPkts	คือ จำนวนแพ็กเก็ต TCP ที่มี Flag เป็น FIN
tcpInNoFlagSetPkts	คือ จำนวนแพ็กเก็ต TCP ที่ไม่มีการตั้งค่า Flag
tcpInPushPkts	คือ จำนวนแพ็กเก็ต TCP ที่มี Flag เป็น PUSH
tcpInRstPkts	คือ จำนวนแพ็กเก็ต TCP ที่มี Flag เป็น RST
tcpInSynPkts	คือ จำนวนแพ็กเก็ต TCP ที่มี Flag เป็น SYN
tcpInUrgPkts	คือ จำนวนแพ็กเก็ต TCP ที่มี Flag เป็น URG

4. การวิเคราะห์หาข้อมูลในกลุ่ม udpDataPkts

UDP (Forouzan, 2006) เป็นโพรโทคอลที่อยู่ในชั้น Transport ซึ่งเป็นชั้นเดียวกับโพรโทคอล TCP ซึ่งการส่งข้อมูลของ UDP นั้นจะเป็นการสั่งครั้งละ 1 ชุดข้อมูล เรียกว่า UDP Datagram ซึ่งจะไม่มีความสัมพันธ์กันระหว่างดาต้าแกรมและจะไม่มีกลไกการตรวจสอบความสำเร็จในการส่งข้อมูลเหมือน TCP ผู้โจมตีสามารถใช้ UDP เป็นเครื่องมือในการโจมตีเครือข่ายได้ โดยการส่งแพ็กเก็ต UDP ไปยังเครื่องเป้าหมายจำนวนมากในเวลาอันรวดเร็ว เพื่อให้เครื่องเป้าหมายไม่สามารถทำงานได้ทัน เกิดการล้นของบัฟเฟอร์ หรือเครื่องอาจจะหยุดทำงานเลยก็เป็นได้ ตัวอย่างการโจมตีโดยใช้ช่องโหว่ของโพรโทคอล UDP ได้แก่ DNS Flood Attack (Ishibashi et al., 2005) Unvalidated InBound Sources UDP หรือ Hijacking (Krawetz, 2007) เป็นต้น และในบางการโจมตีใช้ประโยชน์จากขนาดหรือความยาว (Length) ของแพ็กเก็ต UDP ในการโจมตีอีกด้วย เช่น UDP Bomb Attack (Cisco, 2011)

ดังนั้น อ็อบเจกต์ในกลุ่ม udpDataPkts มีสมาชิกดังนี้

udpInPkts	คือ จำนวนรวมของแพ็กเก็ต UDP
udpInLength	คือ ขนาดของแพ็กเก็ต UDP

5. การวิเคราะห์หาข้อมูลในกลุ่ม ports

พอร์ตถูกกำหนดไว้เพื่อแยกข้อมูลเข้า-ออกของแต่ละแอพพลิเคชั่นไม่ให้ปนเปกัน โดยมี 2 แบบคือ TCP/IP พอร์ต และ UDP พอร์ต ซึ่งจะมีอยู่สองสถานะ คือ เปิด และ ปิด หมายเลขพอร์ตเริ่มจาก 0 ถึง 65,535 โดยพอร์ต 0 – 1,024 จะเรียกว่า Well Known Ports Number คือหมายเลขพอร์ตที่รู้จักกันดีหรือใช้อยู่ทั่วไปซึ่งถูกกำหนดและควบคุมจาก IANA (Internet Assigned Number Authority) เช่น HTTP จะใช้หมายเลขพอร์ต 80 หรือ FTP ใช้หมายเลขพอร์ต 20 และ 21 ส่วน DNS ใช้หมายเลขพอร์ต 53 เป็นต้น และหมายเลขที่นอกเหนือจากนี้จะเรียกว่า Registered Port Number คือหมายเลขพอร์ตที่ต้องลงทะเบียนเพื่อใช้งาน

หากผู้โจมตีทราบว่าเครื่องเป้าหมายเปิดให้บริการอะไร ก็สามารถใช้ช่องทางนี้ให้การโจมตีผ่านทางหมายเลขพอร์ตที่เปิดอยู่ก็เป็นได้ เช่น HTTP Flood Attack, DNS Flood Attack หรือ FTP Bounce Attack เป็นต้น สำหรับข้อมูลของอ็อบเจกต์ในกลุ่มนี้ ผู้วิจัยได้เลือกหมายเลขพอร์ตมาบางหมายเลขเท่านั้น เนื่องจากต้องการตรวจจับความผิดปกติของการบริการบางบริการเท่านั้น ซึ่งในภายหลังหากมีผู้สนใจหรือต้องการจะตรวจจับบริการนอกเหนือจากนี้ ก็สามารถที่จะเพิ่มอ็อบเจกต์ใหม่ที่เกี่ยวข้องกับพอร์ตได้

ดังนั้น อ็อบเจกต์ในกลุ่ม ports มีสมาชิกดังนี้

พอร์ต TCP

tcpSamePortNumber	คือ จำนวนแพ็กเก็ต TCP ที่มีหมายเลขพอร์ตต้นทางและปลายทางเดียวกัน
tcpInPortNumber25	คือ จำนวนแพ็กเก็ต TCP ที่ผ่านพอร์ต 25
tcpInPortNumber80	คือ จำนวนแพ็กเก็ต TCP ที่ผ่านพอร์ต 80
tcpInPortNumber139	คือ จำนวนแพ็กเก็ต TCP ที่ผ่านพอร์ต 139

พอร์ต UDP

udpInPortNumber7	คือ จำนวนแพ็กเก็ต UDP ที่ผ่านพอร์ต 7
udpInPortNumber19	คือ จำนวนแพ็กเก็ต UDP ที่ผ่านพอร์ต 19
udpInPortNumber53	คือ จำนวนแพ็กเก็ต UDP ที่ผ่านพอร์ต 53

6. การวิเคราะห์หาข้อมูลในกลุ่ม otherProtocols

เนื่องจากในปัจจุบันมีการใช้งานเครือข่ายเป็นจำนวนมาก และประกอบกับมีโพรโทคอลอื่นๆ ที่นอกเหนือจาก IP ICMP TCP และ UDP เช่น IPv6 หรือ ARP เป็นต้น ซึ่งโพรโทคอลเหล่านี้ก็ล้วนแล้วแต่เคยถูกโจมตีแล้วทั้งสิ้น ตัวอย่างการโจมตีโพรโทคอลอื่น ๆ ได้แก่ RCP Dump (Cisco, 2011) ARP Spoofing (Whalen, 2001) เป็นต้น

ผู้วิจัยได้เลือกโพรโทคอลอื่นๆ ที่ได้ทำการศึกษารูปแบบการโจมตีมาบางโพรโทคอลเท่านั้น หากต้องการตรวจจับความผิดปกติของโพรโทคอลใดนอกเหนือจากนี้ ผู้สนใจก็สามารถเพิ่มเติมอ็อบเจกต์ใหม่ที่เกี่ยวข้องกับโพรโทคอลที่ต้องการตรวจจับเข้าไปได้ภายใต้กลุ่มอ็อบเจกต์นี้

ดังนั้น อ็อบเจกต์ในกลุ่ม otherProtocols มีสมาชิกดังนี้

โพรโทคอล ARP

arpInRequestPkts	คือ จำนวนแพ็กเก็ต ARP Request
arpInReplyPkts	คือ จำนวนแพ็กเก็ต ARP Reply

7. การวิเคราะห์หาข้อมูลในกลุ่ม others

จากการศึกษาการโจมตีที่เกิดขึ้นบนระบบคอมพิวเตอร์และเครือข่ายของ (Carl and Kesidis, 2006);(Keshariya and Foukia, 2010) ทำให้ทราบว่า เมื่อระบบคอมพิวเตอร์หรือเครือข่ายถูกโจมตี นอกจากปริมาณข้อมูลในเครือข่ายจะเพิ่มมากขึ้นแล้ว การทำงานของเครื่องหรือเซิร์ฟเวอร์ที่ถูกโจมตีจะมีการทำงานที่เพิ่มขึ้นด้วย เช่น ปริมาณการใช้งานหน่วยความจำหรือหน่วยประมวลผลเป็นต้น

ดังนั้น อ็อบเจกต์ในกลุ่ม others มีสมาชิกดังนี้

cpuValue	คือ ค่าของการใช้งานของหน่วยประมวลผล
memoryValue	คือ ค่าการใช้งานหน่วยความจำ

ผลสรุปจากการที่ได้ศึกษาการโจมตีและการตรวจจับที่เกิดบนระบบคอมพิวเตอร์และเครือข่าย ทำให้ได้ข้อมูลที่สามารถใช้ในการตรวจจับความผิดปกติซึ่งอยู่ในรูปของ MIB Object ทั้งหมด 27 อ็อบเจกต์ และในที่นี้ผู้วิจัยขอเรียกว่า MIB+ ซึ่งสามารถทำการเพิ่มเติมอ็อบเจกต์ใหม่ได้ในภายหลัง หากพบว่ามีการโจมตีใหม่ๆ ที่ต้องการตรวจจับเกิดขึ้น

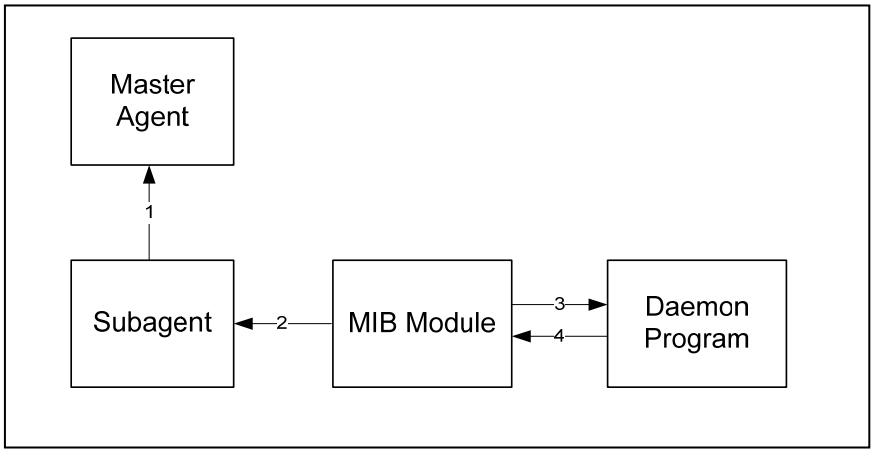
อ็อบเจกต์เหล่านี้เป็นเพียงแค่ชื่อ หรือตัวแปรของข้อมูลเท่านั้น อ็อบเจกต์แต่ละตัวยังไม่มีค่าข้อมูลที่สามารถนำไปใช้ได้ เนื้อหาในหัวข้อถัดไป จะกล่าวถึงการเพิ่มอ็อบเจกต์เหล่านี้ให้ไปอยู่ในฐานข้อมูล MIB+ พร้อมทั้งอธิบายถึงวิธีการระบุค่าให้กับอ็อบเจกต์ต่างๆ

3.5 การกำหนดค่าให้กับแต่ละอ็อบเจกต์

อ็อบเจกต์ที่กล่าวมาทั้งหมดนั้น เป็นเพียงตัวแปรที่กำหนดขึ้นเพื่อใช้สำหรับอ้างอิงข้อมูล ซึ่งส่วนค่าข้อมูลของตัวแปรนั้นผู้วิจัยได้พัฒนาโดยใช้ซอฟต์แวร์เอเจนท์ของกลุ่ม Net-SNMP (2011) เนื่องจากเป็นซอฟต์แวร์ที่ได้รับความนิยมและใช้งานกันอย่างแพร่หลาย อีกทั้งยังเป็นซอฟต์แวร์ที่เปิดเผยชุดคำสั่งโปรแกรมต้นฉบับและยังได้จัดเตรียมชุดคลังโปรแกรม (Library Function) ต่างๆ มากมายเพื่อให้ผู้ใช้สามารถนำไปพัฒนาเพิ่มเติมต่อได้อย่างสะดวก

การกำหนดค่าให้กับแต่ละอ็อบเจกต์นั้นมีด้วยกันสองแนวทางคือ แบบ Dynamically Loadable Object และแบบ Subagent โดยวิธีการแบบ Dynamically Loadable Object นั้นต้องเขียน Agent ให้อยู่ในรูปแบบของ Shared Objects โดยจะได้ Binary File ที่สามารถนำไปใช้สำหรับ SNMPd ซึ่งเป็นโปรแกรมชนิดที่เรียกว่า Daemon ในระบบปฏิบัติการยูนิกซ์ ให้สามารถเข้าถึงการทำงานของ SNMPd ได้โดยตรง สำหรับวิธีการแบบ Subagent นั้นจะใช้ Library AgentX ที่มีมาแล้วใน Net-SNMP เป็นเครื่องมือในการค้นหา Interface ที่ MIB Module API นั้นสามารถเข้าถึงได้ ทำให้มีการใช้งานที่ง่ายขึ้น

ผู้วิจัยได้เลือกใช้แนวทางในการพัฒนาโดยเขียนโปรแกรมแบบ Subagent โดยมีกรอบแนวคิดการทำงานแสดงดังภาพที่ 3-4

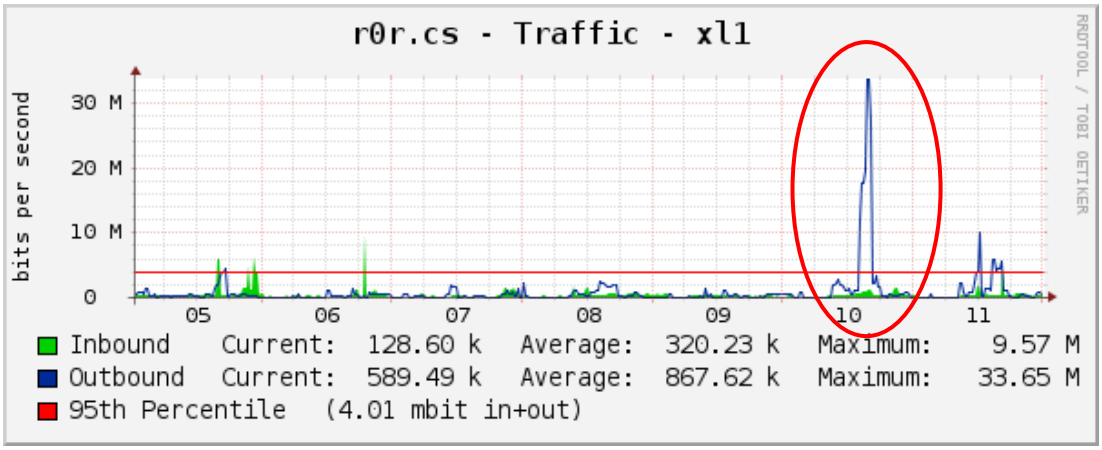


ภาพประกอบที่ 3-4 กรอบแนวคิดการพัฒนาอ็อบเจกต์

เมื่อ Subagent ติดต่อกับ Master Agent แล้ว เราสามารถเรียกใช้ MIB Module ที่สร้างขึ้นใหม่ได้ โดยที่เราสามารถใช้คำสั่งต่างๆ เพื่อดูค่าการทำงานของ Daemon Program ที่เราสร้างขึ้น ซึ่งโพรโทคอล SNMP จะคอยทำหน้าที่ในการสอบถามค่าข้อมูลเหล่านั้นผ่านคำสั่งของ SNMP เช่น snmpget, snmpgetnext หรือ snmptable เป็นต้น

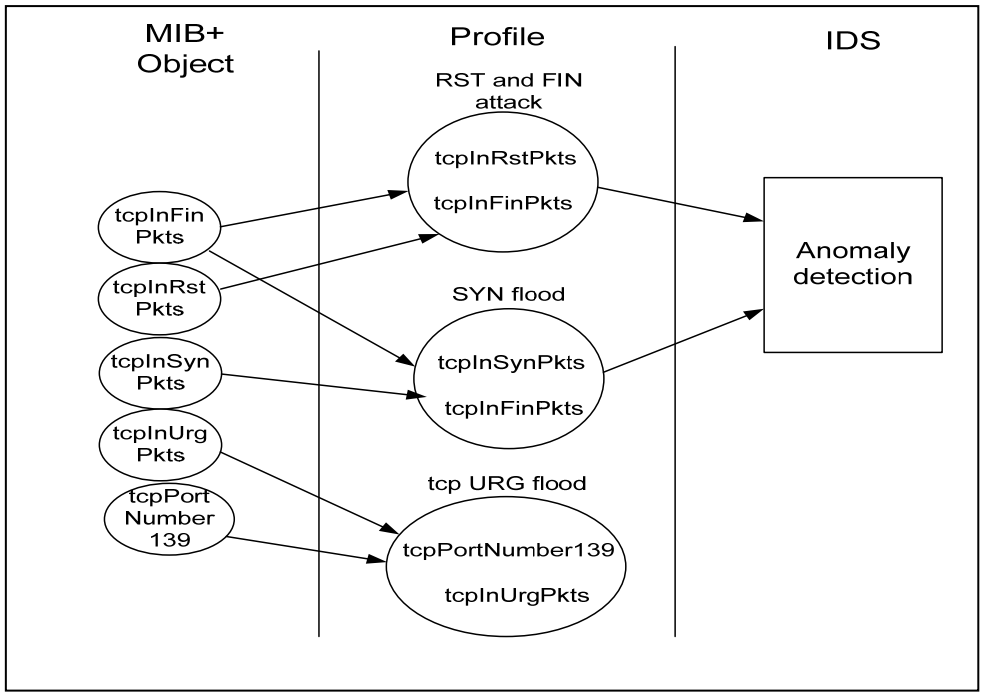
3.6 แนวคิดในการนำอ็อบเจกต์มาใช้สร้างโพรไฟล์

ในการตรวจจับความผิดปกติที่เกิดขึ้นบนเครือข่ายนั้น เบื้องต้นเราสามารถดูความผิดปกติได้จากค่าข้อมูลของอ็อบเจกต์ใน MIB มาตรฐานได้ นั่นคืออ็อบเจกต์ ifInOctets ifOutOctets และอ็อบเจกต์อื่นๆ อีก แต่โดยทั่วไปแล้วใน Network Management System มักใช้อ็อบเจกต์ ifInOctets IfOutOctets ในการดูภาพรวมของเครือข่าย เช่น โปรแกรม mrtg และ cacti แต่เมื่อพบความผิดปกติเกิดขึ้นบนเครือข่าย เราไม่สามารถระบุได้ว่ามีความผิดปกติอะไรเกิดขึ้น ดังภาพประกอบที่ 3-5



ภาพที่ 3-5 ข้อมูลที่แสดงถึงความผิดปกติที่ได้จากโปรแกรม mrtg

เพื่อให้สามารถระบุรูปแบบความผิดปกติที่เกิดขึ้นได้และสามารถหาวิธีการป้องกันได้ตรงกับปัญหาที่เกิดขึ้น อ็อบเจกต์ที่ผู้วิจัยได้นำเสนอไปในหัวข้อก่อนหน้านี้จะเป็นตัวช่วยในการระบุความผิดปกติที่เกิดขึ้นได้ โดยนำอ็อบเจกต์ใน MIB+ ใช้เป็นพารามิเตอร์สำหรับสร้างโพรไฟล์ของเหตุการณ์ปกติและนำไปใช้สำหรับการตรวจจับการบุกรุก เมื่อมีเหตุการณ์ผิดปกติของเครือข่าย ดังภาพที่ 3-5 หนึ่งอ็อบเจกต์ใน MIB+ สามารถนำมาสร้างเป็นโพรไฟล์ที่แสดงความปกติสำหรับตรวจจับการบุกรุกได้มากกว่า 1 รูปแบบดังแสดงในภาพประกอบที่ 3-6



ภาพประกอบที่ 3-6 แนวคิดในการนำอ็อบเจกต์มาใช้งาน

3.6.1 หลักการในการนำอ็อบเจกต์ MIB+ มาใช้

หลักการในการนำอ็อบเจกต์มาใช้งานนั้น มีด้วยกัน 2 วิธี ซึ่งเราสามารถเลือกใช้เพียงวิธีใดวิธีหนึ่ง หรือสามารถใช้ควบคู่กันได้ คือ

1. ใช้องค์ความรู้ที่ได้จากการศึกษารูปแบบ หรือวิธีการตรวจจับที่ผ่านมาในอดีต เพื่อใช้สำหรับเลือกข้อมูลหรืออ็อบเจกต์ที่นำมาใช้สร้างโพรไฟล์สำหรับตรวจจับการบุกรุกต่อไป

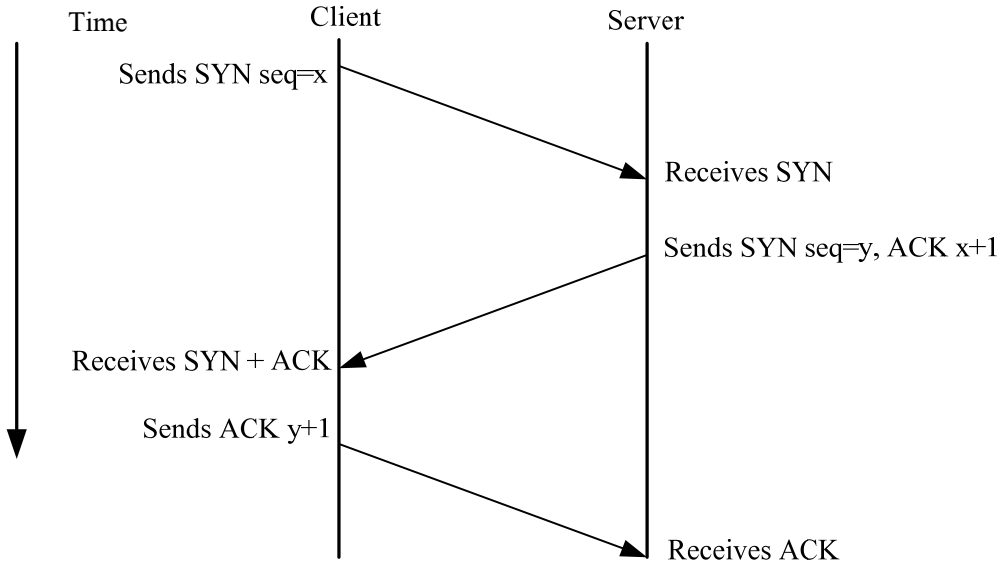
2. ใช้หลักการ Correlation-based Feature Selection: CFS เพื่อให้ได้อ็อบเจกต์ที่เหมาะสมและลดขนาดของข้อมูล (Data Reduction) ที่ไม่จำเป็นในการจัดเก็บ ซึ่งวิธีการนี้จำเป็นต้องใช้องค์ความรู้ที่ได้จากการศึกษารูปแบบการตรวจจับการโจมตีด้วยเช่นกัน ผู้วิจัยได้เลือกใช้วิธีการนี้ร่วมกับวิธีการในข้อ 1 เพื่อเป็นการพิสูจน์หรือยืนยันว่า อ็อบเจกต์ที่ได้เลือกมานั้นมีความเหมาะสมที่จะใช้สำหรับสร้างโพรไฟล์เพื่อตรวจจับความผิดปกติที่ต้องการ และเลือกใช้โปรแกรม Weka ในการสกัดให้ได้มาซึ่งอ็อบเจกต์ที่ใช้ในการตรวจจับ

3.7 การวิเคราะห์อ็อบเจกต์สำหรับตรวจจับการบุกรุก

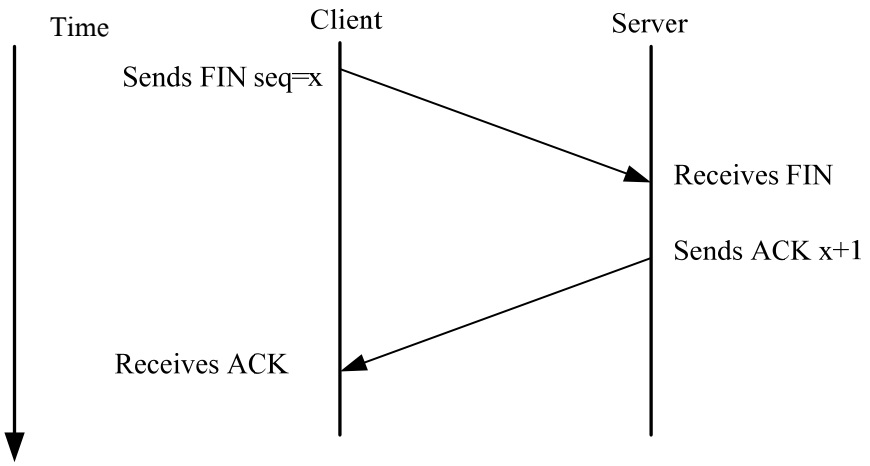
วิทยานิพนธ์ชุดนี้นำเสนอการตรวจจับการบุกรุกบนระบบคอมพิวเตอร์และเครือข่าย 5 รูปแบบ ได้แก่ SYN Flood Attack, DNS Flood Attack, Land Attack, Null Scan และ Xmas Scan ซึ่งมีรายละเอียดในการวิเคราะห์การโจมตีและการตรวจจับการโจมตีแต่ละรูปแบบดังนี้

3.7.1 SYN Flood Attack

ในการเชื่อมต่อของ TCP แบบปกตินั้นเมื่อ Client ต้องการที่จะเชื่อมต่อไปยัง Server จะทำการส่งแพ็กเก็ต TCP ที่ตั้งค่า Flag เป็น SYN และหมายเลขลำดับเริ่มต้น (Initial Sequence Number: ISN) ISN = x ไปยัง Server เพื่อขอเริ่มต้นการเชื่อมต่อ เมื่อ Server ได้รับแพ็กเก็ตดังกล่าวแล้วก็จะส่งแพ็กเก็ต TCP ที่ตั้งค่า Flag เป็น SYN และ ACK พร้อมทั้ง ISN = y ไปยัง Client และเมื่อ Client ได้รับแพ็กเก็ตที่ Server ส่งกลับมาแล้ว Client สามารถทำการเชื่อมต่อกับ Server ได้โดยส่งแพ็กเก็ต TCP ที่ตั้งค่า Flag เป็น ACK และ ISN = y+1 ไปยัง Server ตลอดการเชื่อมต่อ และเมื่อ Client ต้องการจะยกเลิกการเชื่อมต่อ ก็จะส่ง TCP ที่ตั้งค่า Flag เป็น FIN+ACK ไปให้กับ Server เมื่อ Server ได้รับก็จะส่ง ACK กลับมาเป็นอันสิ้นสุดการเชื่อมต่อ การทำงานดังกล่าวแสดงให้เห็นในภาพประกอบที่ 3-7 (a) และ (b)

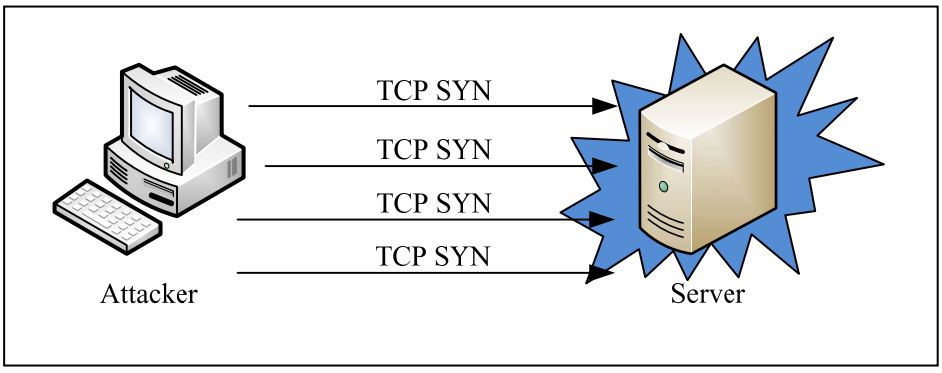


ภาพประกอบที่ 3-7 (a) TCP Connection Establishment



ภาพประกอบที่ 3-7 (b) TCP Connection Terminate

การโจมตีรูปแบบ SYN Flood Attack นั้น ผู้โจมตีทำการส่งแพ็กเก็ต TCP ที่ตั้งค่า Flag เป็น SYN จำนวนมากไปยังเครื่องเป้าหมายเสมือนเป็นการเริ่มต้นการเชื่อมต่อแบบ TCP ตามปกติ หากมีการส่งแพ็กเก็ตประเภทนี้จำนวนมากจะทำให้คิวงานของเครื่องเป้าหมายเต็ม จนไม่สามารถทำงานหรือไม่บริการได้ตามปกติ ดังแสดงให้เห็นในภาพประกอบที่ 3-8



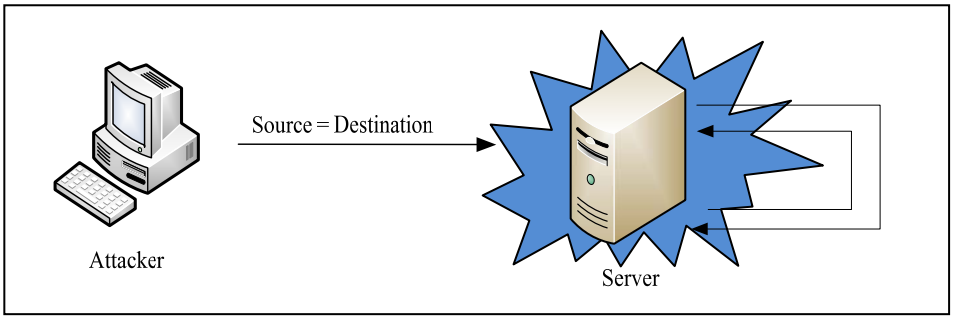
ภาพประกอบที่ 3-8 SYN Flood Attack

จากการเชื่อมต่อ TCP แบบปกติจะเห็นได้ว่าความสัมพันธ์ที่เกิดขึ้นคือ เมื่อมีการเริ่มต้นการเชื่อมต่อจะต้องมีการขอสิ้นสุดการเชื่อมต่อ เพราะฉะนั้นปริมาณหรืออัตราส่วนของ SYN และ FIN จะมีอัตราส่วนที่ใกล้เคียงกัน ดังนั้นหากเกิดการโจมตีรูปแบบนี้เกิดขึ้น จะทำให้ปริมาณและอัตราส่วนระหว่าง FIN และ SYN สูงกว่าปกติ ดังนั้นอ็อบเจกต์ที่ใช้ในการสร้างโพรไฟล์นี้คือ tcpInSynPkts และ tcpInFinPkts

3.7.2 Land Attack

ผู้โจมตีจะทำการปลอมแปลงหมายเลข IP ต้นทางเหมือนกับหมายเลข IP ปลายทางให้เป็นหมายเลขเดียวกันและทำการส่งแพ็กเก็ต TCP ที่ตั้งค่า Flag เป็น SYN ไปยังเครื่องเป้าหมายเพื่อขอการเชื่อมต่อ ซึ่งเครื่องเป้าหมายจะต้องตอบกลับคำร้องด้วย SYN-ACK แต่เนื่องจากว่าหมายเลข IP ของเครื่องนั้นต้นทางนั้นเป็นหมายเลขเดียวกันกับเครื่องเป้าหมาย ทำให้แพ็กเก็ตที่ส่งไปนั้นย้อนกลับเข้าหาตนเอง และการที่ปล่อย SYN-ACK แต่ละครั้ง ต้องมีการปันส่วนของหน่วยความจำเพื่อการทำงานนี้จำนวนหนึ่ง ซึ่งหากผู้โจมตีส่งคำร้องขอเชื่อมต่อมาอย่างต่อเนื่องก็จะเกิดปัญหาในเรื่องของการจัดการหน่วยความจำ ดังภาพประกอบที่ 3-9

การโจมตีรูปแบบนี้จะทำให้มีปริมาณแพ็กเก็ตที่ผิดแปลกไปจากปกตินั่นคือมีปริมาณแพ็กเก็ตที่มีหมายเลข IP ต้นทางและปลายทางเหมือนกัน ซึ่งในปกติแล้วจะไม่พบแพ็กเก็ตประเภทนี้ ดังนั้นอ็อบเจกต์ที่ใช้ในการสร้างโพรไฟล์นี้คือ ipInAddSrcDest



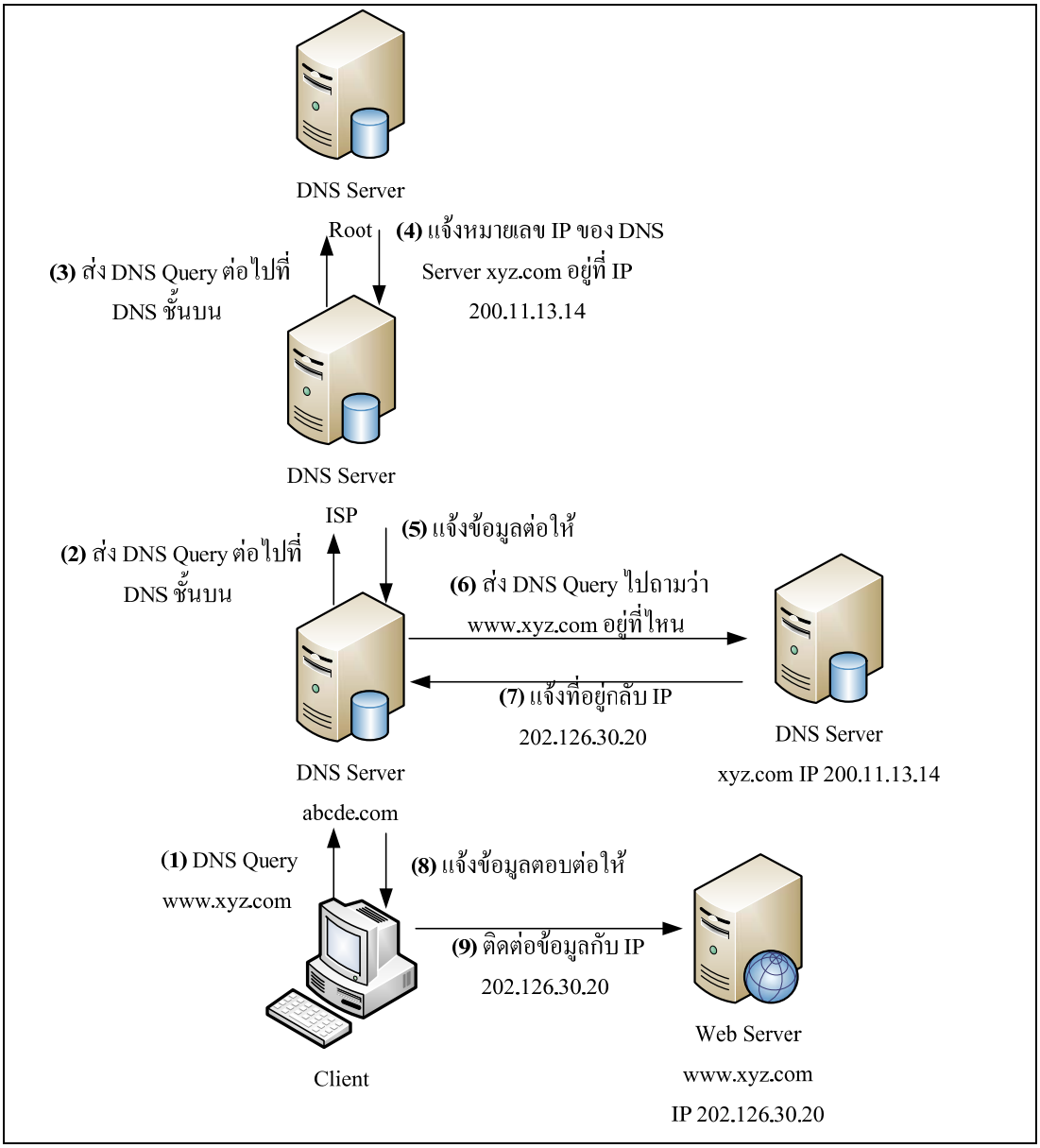
ภาพประกอบที่ 3-9 Land Attack

3.7.3 DNS Flood Attack

หน้าที่หลักของ DNS คือแปลงหมายเลข IP ให้เป็นชื่อโดเมน (Domain Name) ซึ่งมีการทำงานดังแสดงในภาพประกอบที่ 3-10 ส่วนรายละเอียดของการทำงานสามารถอ่านได้จากเอกสารที่เกี่ยวข้องกับ DNS ทั่วๆ ไปหรือจาก (TechNet, 2003)

เป้าหมายในการโจมตี DNS Server นั้นก็เพื่อเพิ่มภาระการทำงานให้กับ Server หรือทำให้เครื่องเป้าหมายไม่สามารถให้บริการได้ตามปกติ ผู้โจมตีจะส่งคำร้องขอไปยัง DNS Server จำนวนมากในเวลาอันรวดเร็ว หรืออาจจะทำการปลอม DNS Request เพื่อเพิ่มภาระการทำงานให้แก่ DNS Server ส่งผลให้ Server ใช้ทรัพยากรที่มากกว่าปกติ เช่น หน่วยความจำหรือ CPU เราสามารถตรวจจับการโจมตีรูปแบบนี้ได้โดยดูจากปริมาณของ

DNS Query ที่เพิ่มขึ้นอย่างรวดเร็ว ซึ่งปกติแล้วการทำงาน DNS นั้นจะทำงานภายใต้โพรโตคอล UDP โดยใช้พอร์ตหมายเลข 53 ดังนั้นเราสามารถตรวจจับการโจมตีประเภทนี้ได้โดยดูจากปริมาณแพ็กเก็ต UDP ที่ใช้พอร์ต 53 อ็อบเจกต์ที่ใช้คือ udpInPortNumber53



ภาพประกอบที่ 3-10 การทำงานของ DNS

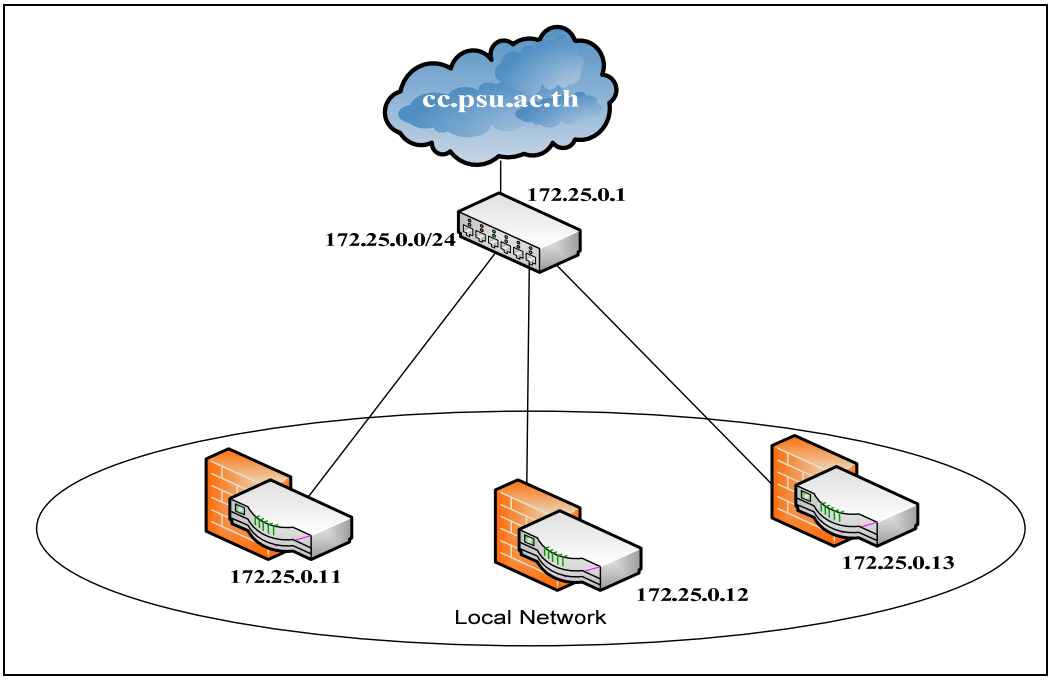
3.7.4 Null Scan

การสแกนเป็นเทคนิคที่รู้จักกันแพร่หลายที่ผู้โจมตีให้ในการค้นหาพอร์ตหรือบริการที่เครื่องเป้าหมายเปิดให้บริการหรือใช้งานอยู่ เพื่อเป็นข้อมูลสำหรับใช้ในการโจมตีบริการต่างๆ ที่เปิดอยู่ได้ การทำงานหรือบริการนั้นอาจจะอยู่บนพอร์ตที่รู้จัก (Well Known Port) หรือไม่เป็นที่รู้จัก (Unknown Port) ซึ่งเทคนิค Null Scan เป็นเทคนิคหนึ่งที่ผู้โจมตีทำการส่งแพ็กเก็ตที่ไม่มีการตั้งค่า Flag ไปยังเครื่องเป้าหมายเพื่อสแกนหาพอร์ตที่เปิดอยู่ ถ้าหากพอร์ตใดปิดอยู่ก็จะส่งแพ็กเก็ต TCP ที่มีค่า Flag RST กลับมา ซึ่งวิธีนี้สามารถตรวจจับได้โดยดูจากจำนวนแพ็กเก็ต TCP ที่ไม่มีการตั้งค่า Flag ที่เข้ามาในเครือข่าย ดังนั้นอ็อบเจกต์ที่ใช้ตรวจจับคือ tcpInNoFlagSetPkts

3.7.5 Xmas Scan

เป็นวิธีการสแกนเพื่อหาพอร์ตบนเครื่องเป้าหมาย โดยจะส่งแพ็กเก็ต TCP ที่มีการตั้งค่า Flag เป็น URG (URG Flag:เป็นการบอกว่าจะต้องมีการจัดการกับแพ็กเก็ตนี้อย่างเร่งด่วน) PSH (PSH Flag:จัดเก็บข้อมูลลงในหน่วยความจำ) และ FIN (FIN Flag:สิ้นสุดการเชื่อมต่อ) พร้อมกัน ไปยังพอร์ตของเครื่องเป้าหมาย ในการเชื่อมต่อหรือส่งข้อมูลกันในสภาวะปกติแล้ว จะไม่พบแพ็กเก็ตที่มีการตั้งค่า Flag ดังกล่าวพร้อมกัน ดังนั้นในการตรวจจับการโจมตีรูปแบบนี้สามารถทำได้โดย ดูจากแพ็กเก็ตที่มีการตั้งค่า Flag เป็น URG, PSH และ FIN พร้อมกัน ดังนั้นอ็อบเจกต์ที่ใช้ในการตรวจจับการโจมตีประเภทนี้คือ tcpInUrgPkts, tcpInPushPkts, และ tcpInFinPkts

เพื่อทดสอบว่าอ็อบเจกต์ที่ได้จากการวิเคราะห์ทั้งหมดข้างต้นนั้นมีความเหมาะสมและเป็นอ็อบเจกต์ที่สามารถใช้ตรวจจับการบุกรุกดังกล่าวได้ ผู้วิจัยจึงได้ทำการเก็บข้อมูลอ็อบเจกต์ที่ได้จากการพิจารณาดังตาราง ที่ 3-5 ข้อมูลที่ทำการเก็บในขั้นต้นนี้ ทำการเก็บในสภาพแวดล้อมที่สามารถควบคุมได้ว่าเป็นสิ่งแวดล้อมปกติไม่มีการบุกรุกหรือถูกโจมตีจากบุคคลอื่น เนื่องจากเครือข่ายของภาควิชา ฯ ได้มีการป้องกันการโจมตีจากภายนอกโดยการกำหนดให้มีโปรแกรมไฟร์วอลล์ติดตั้งในอุปกรณ์เครือข่ายหลักทั้งหมด ดังภาพประกอบที่ 3-11



ภาพประกอบที่ 3-11 เครือข่ายภาควิชาวิทยาการคอมพิวเตอร์

จากนั้นผู้วิจัยได้จำลองสถานการณ์การโจมตีตามรูปแบบข้างต้น และเก็บข้อมูลการโจมตีเพื่อใช้เป็นข้อมูลในการทดสอบ สำหรับในส่วนของการคัดกรองข้อมูลที่ไม่จำเป็นหรือเป็นข้อมูลที่มีความซ้ำซ้อนออกผู้วิจัยได้เลือกใช้หลักการของ Correlation-base Feature Selection (CFS) (ภาคผนวก ก.) ซึ่งเครื่องมือที่ใช้คือ โปรแกรม Weka ซึ่งเป็นเครื่องมือที่เน้นเกี่ยวกับงานด้านการเรียนรู้ของเครื่องและการทำเหมืองข้อมูล (The University of Waikoto, 1997) ผลลัพธ์สุดท้ายที่ได้เมื่อผ่าน CFS แล้วแสดงในตารางที่ 3-6

ตารางที่ 3-5 SNMP MIB Objects ที่ใช้สำหรับตรวจจับการโจมตี

Attack Type	MIB Object
SYN Flood Attack	tcpInFinPkts tcpInSynPkts
Land Attack	ipInAddSrcDest
DNS Flood Attack	udpInPortNumber53
NULL Scan	tcpInNoFlagSetPkts
Xmas Scan	tcpInUrgPkts tcpInPushPkts tcpInFinPkts

ตารางที่ 3-6 SNMP MIB Objects ที่ถูกเลือกจากฟังก์ชัน CFS

Attack Type	MIB Object
SYN Flood Attack	tcpInSynPkts
Land Attack	ipInAddSrcDest
DNS Flood Attack	udpInPortNumber53
NULL Scan	tcpInNoFlagSetPkts
Xmas Scan	tcpInUrgPkts tcpInPushPkts

เพื่อทดสอบความถูกต้องของอ็อบเจกต์ที่ได้จากกระบวนการ CFS ว่าสามารถใช้จำแนกความผิดปกติในรูปแบบที่กำหนดได้หรือไม่ ผู้วิจัยได้นำข้อมูลการโจมตีในแต่ละประเภทมาทำการทดสอบโดยใช้อัลกอริทึม J48 หรืออัลกอริทึม C4.5 (Quinlan, 1993) ซึ่งเป็นอัลกอริทึมแบบ Decision Tree สามารถนำมาใช้ในการจำแนกแยกหมวดหมู่ (Classification) ซึ่งมักถูกใช้บ่อยสำหรับแยกข้อมูลในเชิงสถิติ (Statistical Classifier) ซึ่งค่าความถูกต้องที่ได้จากข้อมูลที่นำมาทดสอบ แสดงดังตารางที่ 3-7

ตารางที่ 3-7 ค่าความถูกต้องในการจำแนกข้อมูลโดยใช้อัลกอริทึม J48

Attack Type	Correctly Classified
SYN Flood Attack	98.67%
Land Attack	100%
DNS Flood Attack	99.59%
NULL Scan	100%
Xmas Scan	99.87%

จากตารางที่ 3-6 จะเห็นได้ว่าอ็อบเจกต์ที่ได้จากการวิเคราะห์การโจมตีข้างต้นและจากการคัดกรองโดย CFS นั้นมีความสอดคล้องกัน แต่เนื่องจากวิธีการของ CFS นี้มีข้อจำกัดตรงที่ว่า วิธีการนี้ต้องมีข้อมูลที่ใช้สำหรับฝึกสอนและทดสอบอยู่ด้วย ซึ่งข้อมูลเหล่านี้จะได้มาจากการโจมตีจริงหรือข้อมูลที่ได้รับการยืนยันจากผู้เชี่ยวชาญในเรื่องดังกล่าวทำการระบุหรือจำแนกข้อมูลว่าข้อมูลที่เกิดขึ้นนั้นเป็นข้อมูลที่มีความผิดปกติหรือเป็นลักษณะข้อมูลการโจมตีประเภทใด หากไม่มีข้อมูลดังกล่าวก็จะไม่สามารถใช้วิธีการของ CFS ได้ CFS จึงเป็นทางเลือกหนึ่งในการช่วยลดขนาดของข้อมูลในการจัดเก็บ ซึ่งแบบจำลองที่ได้มานั้นจะมีประสิทธิภาพดีหรือไม่ทั้งนี้ก็ขึ้นอยู่กับชุดข้อมูลที่นำมาทดสอบด้วย

3.8 สรุป

จากการศึกษาวิเคราะห์และออกแบบสถาปัตยกรรมโพรไฟล์เพื่อใช้ตรวจจับความผิดปกติในบทนี้นั้น ได้ใช้หลักการของ Management Information Base หรือ MIB มาใช้ในการจัดการ โดยผู้วิจัยได้นำเสนอ MIB Object หรือเรียกว่า MIB+ ที่จำเป็นสำหรับใช้ตรวจจับความผิดปกติบนระบบคอมพิวเตอร์และเครือข่าย โดยประกอบด้วยกลุ่มอ็อบเจกต์ทั้งหมด 7 กลุ่ม ซึ่งใช้หลักการพัฒนาอ็อบเจกต์แบบ Subagent มีการทำงานร่วมกับโพรโทคอล SNMP พร้อมทั้งการวิเคราะห์อ็อบเจกต์สำหรับใช้ในการตรวจจับการบุกรุก ในบทถัดไปจะเป็นการกล่าวถึงการพัฒนาระบบตรวจจับการบุกรุกโดยใช้โพรไฟล์สำหรับตรวจจับความผิดปกติและนำอ็อบเจกต์ต่างๆ ที่ได้กล่าวไว้แล้วข้างต้นไปใช้ในการระบุประเภทความผิดปกติที่เกิดขึ้น

บทที่ 4

การออกแบบระบบ

4.1 บทนำ

ในบทที่แล้วเป็นการนำเสนอข้อมูลที่ใช้ในการสร้างโพรไฟล์ ซึ่งสามารถนำมาใช้ตรวจจับการบุกรุกบนระบบคอมพิวเตอร์และเครือข่ายได้ สำหรับในบทนี้จะกล่าวถึงการนำอ็อบเจกต์หรือข้อมูลใน MIB+ ที่ได้นำเสนอในบทที่แล้วมาใช้สร้างโพรไฟล์สำหรับตรวจจับความผิดปกติบนระบบคอมพิวเตอร์และเครือข่าย

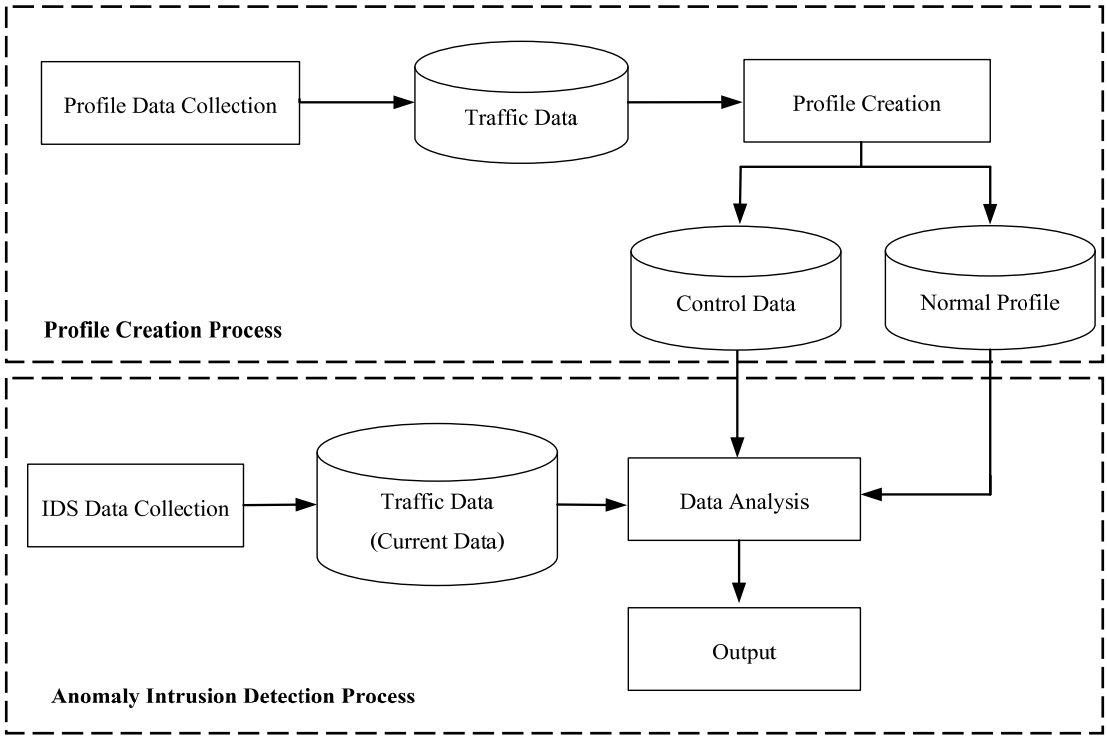
เนื้อหาในบทนี้กล่าวถึงการออกแบบระบบที่ใช้สำหรับตรวจจับความผิดปกติหรือการบุกรุกโดยการใช้เครื่องมือ SNMP และ MIB Object (MIB+) ที่ได้นำเสนอในบทก่อนหน้านี้มาใช้ในการเก็บรวบรวมข้อมูล พร้อมทั้งนำเสนอภาพรวมของระบบตรวจจับการบุกรุกที่พัฒนาขึ้นอีกด้วย

4.2 แผนภาพโดยรวมของการพัฒนาระบบ

ระบบตรวจจับการบุกรุกที่ออกแบบมานั้นใช้สำหรับตรวจจับการบุกรุกที่เกิดขึ้นบนเครือข่าย โดยใช้แนวคิดในการตรวจจับแบบ Anomaly Detection โดยภาพรวมการทำงานของระบบจะแบ่งเป็น 2 ส่วน

- ส่วนแรก เป็นการสร้างโพรไฟล์โดยใช้ทั้งอ็อบเจกต์ใน Standard MIB และ MIB+ ในการเก็บข้อมูลเพื่อนำมาสร้างเป็นโพรไฟล์
- ส่วนที่สอง คือส่วนของการวิเคราะห์และตรวจจับความผิดปกติที่เกิดขึ้นบนเครือข่ายโดยใช้โพรไฟล์ที่ได้จากส่วนแรกเป็นตัวเปรียบเทียบความผิดปกติของการทำงานของเครือข่าย

โดยในการทำงานแต่ละส่วนมีโพรเซสการทำงานย่อยๆ แสดงในภาพประกอบที่ 4-1 และมีรายละเอียดของแต่ละโพรเซสดังนี้



ภาพประกอบที่ 4-1 ภาพรวมของระบบตรวจจับการบุกรุกที่พัฒนาขึ้น

4.2.1 Data Collection

Data Collection หรือกระบวนการเก็บรวบรวมข้อมูลแบ่งได้เป็น 2 ส่วนคือ Profile Data Collection และ IDS Data Collection โดยมีรายละเอียดในแต่ละส่วนดังนี้

4.2.1.1 Profile Data Collection เป็นขั้นตอนแรกในการเก็บข้อมูลของอ็อบเจกต์เพื่อนำมาสร้างโพรไฟล์เพื่อใช้ในการตรวจจับความผิดปกติต่อไป ผู้วิจัยได้เลือกใช้อ็อบเจกต์ทั้งใน Standard MIB นั่นคือ ifInOctets และ ifOutOctets เพื่อให้เห็นถึงลักษณะการใช้งานโดยรวมของเครือข่าย และใช้ MIB+ คือ tcpInFinPkts, tcpInSynPkts, tcpInPushPkts, tcpInUrgPkts, tcpInNoFlagSetPkts, ipInAddSrcDest และ udpInportNumber53 เพื่อระบุรูปแบบของความผิดปกติที่เกิดขึ้น

4.2.1.2 IDS Data Collection เป็นขั้นตอนในการเก็บรวบรวมข้อมูลของอ็อบเจกต์เพื่อใช้ในการตัดสินเหตุการณ์ที่เกิดขึ้น ว่ามีความผิดปกติหรือไม่ ซึ่งข้อมูลอ็อบเจกต์ที่เก็บนั้นจะเป็นอ็อบเจกต์เดียวกันกับขั้นตอน Profile Data Collection ซึ่งทั้งสองขั้นตอนนี้ใช้คำสั่ง snmpget ในการสอบถามค่าข้อมูลของอ็อบเจกต์ในอุปกรณ์ทุกๆ 5 นาที เพราะเป็นระยะเวลาการสอบถามที่มักใช้ในระบบจัดการ เนื่องจากถ้าใช้คำสั่ง SNMP ในการ

สอบถามถี่เกินไปจะเป็นการสร้างภารงานให้แก่อุปกรณ์ที่ถูกสอบถามได้ ตัวอย่างคำสั่งที่ใช้และผลลัพธ์ที่ได้แสดงในภาพประกอบที่ 4-2

คำสั่ง snmpget

snmpget –v 2c –c public 172.25.0.202 INTRUSION-DATA-MIB::tcpInSynPkts.0

ผลลัพธ์

INTRUSION-DATA-MIB::tcpInSynPkts.0 = Counter32: 479250

ภาพประกอบที่ 4-2 ตัวอย่างการใช้งานคำสั่ง snmpget และผลลัพธ์ที่ได้จากคำสั่ง

4.2.2 Profile Creation

Profile Creation หรือกระบวนการสร้างโพรไฟล์ เป็นกระบวนการประมวลผลข้อมูลที่ได้จากกระบวนการ Profile Data Collection ซึ่งโพรไฟล์ที่สร้างขึ้นมีด้วยกันสองประเภท คือ โพรไฟล์ที่ใช้สำหรับดูความผิดปกติโดยรวมของเครือข่าย โดยใช้อ็อบเจกต์ใน Standard MIB คือ ifInOctets และ ifOutOctets ถ้าพบว่ามีความผิดปกติเกิดขึ้นจะใช้ข้อมูลโพรไฟล์ที่ได้จากการเก็บข้อมูลอ็อบเจกต์ของ MIB+ ในการระบุความผิดปกติที่เกิดขึ้นว่าเป็นความผิดปกติรูปแบบใด โดยใช้อ็อบเจกต์ใน MIB+ เป็นพารามิเตอร์ในการตรวจจับการบุกรุกตามที่ได้กล่าวไว้แล้วในบทที่ 3 สำหรับงานวิจัยนี้ได้พัฒนาระบบตรวจจับการบุกรุกให้สามารถตรวจจับการบุกรุกได้ 5 รูปแบบ ได้แก่ SYN Flood Attack, DNS Flood Attack, Land Attack, Null Scan และ Xmas Scan

ผลลัพธ์ที่ได้จากกระบวนการ Profile Creation นี้ จะได้ชุดข้อมูล 2 ชุดข้อมูลคือ Normal Profile และ Control Data โดยที่ Normal Profile คือข้อมูลที่ได้จากการเก็บรวบรวมจากอ็อบเจกต์เพื่อสร้างเป็นโพรไฟล์ ซึ่งจะทำให้เห็นถึงข้อมูลที่ผ่านมาในอดีตได้ แต่เนื่องจากข้อมูลในรูปแบบนี้ ไม่สามารถนำมาใช้ในการคำนวณหาความผิดปกติของเหตุการณ์ที่กำลังเกิดขึ้นได้ ผู้วิจัยจึงนำข้อมูลโพรไฟล์นี้มาคำนวณหาช่วงของข้อมูลเพื่อให้สามารถใช้สำหรับตรวจจับความผิดปกติได้ โดยใช้วิธีทางสถิติที่เรียกว่า Statistical Process Control (ภาคผนวก ก.) โดยผลลัพธ์ที่ได้จากการคำนวณนี้จะเก็บไว้ใน Control Data

4.2.2.1 การคำนวณ Statistical Process Control

หลังจากที่เก็บรวบรวมข้อมูลของอ็อบเจกต์สำหรับสร้างโพรไฟล์แล้ว ขั้นตอนต่อมาคือการคำนวณหาค่าช่วงข้อมูลของอ็อบเจกต์ในแต่ละโพรไฟล์โดยแสดงตัวอย่างการคำนวณดังนี้

ข้อมูลในตารางคือข้อมูล MIB+ ของอ็อบเจกต์ udpInPortNumber53 ที่ได้ทำการเก็บรวบรวมในเดือนมกราคม 2555 ของทุกๆ วันจันทร์ ทำการสอบถามข้อมูลทุกๆ 5 นาที ทำให้ได้จำนวนแถวของข้อมูลเท่ากับ 288 แถว ซึ่งในการคำนวณครั้งนี้จะขอยกตัวอย่างข้อมูลมาบางส่วน เพื่อให้เห็นถึงวิธีการคำนวณ มีขั้นตอนในการคำนวณดังนี้

Sample Number	Monday 1st	Monday 2nd	Monday 3rd	Monday 4th	Average ($\bar{x}$)
1	255	221	238	231	236.25
2	273	285	258	210	256.5
3	246	268	258	256	252.75
4	223	183	184	239	207.25
5	181	230	157	214	195.5
6	251	209	225	201	221.5
7	284	253	234	255	256.5
8	279	256	280	244	264.75
9	192	248	228	230	224.5
10	167	234	212	161	193.5
11	195	183	161	114	163.25
12	251	206	226	300	245.75
Total					**2718**

(1) คำนวณหาค่าเฉลี่ยของข้อมูล ณ ช่วงเวลาที่ตรงกันในแต่ละวัน จากสมการ $\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$ โดยที่ n เท่ากับ 4 แสดงผลลัพธ์ในตาราง

(2) คำนวณหาค่าเฉลี่ยของค่าเฉลี่ยในแต่ละช่วงเวลาจากสมการ

$\bar{\bar{x}} = \frac{\bar{x}_1 + \bar{x}_2 + \bar{x}_3 + ...\bar{x}_k}{k}$ เมื่อ k = 12

$\bar{\bar{x}}$ = Round(2718/12)

$\bar{\bar{x}}$ = 227

(3) คำนวณหาค่าขีดจำกัดบนและขีดจำกัดล่าง ของข้อมูลจากสมการ $UCL = \bar{\bar{x}} + z\sigma_{\bar{x}}$ และ $LCL = \bar{\bar{x}} - z\sigma_{\bar{x}}$ ตามลำดับ โดยคำนวณหาค่าส่วนเบี่ยงเบนมาตรฐานของการกระจายเฉลี่ยกลุ่มข้อมูล ($\sigma_{\bar{x}}$) จากสมการ $\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}}$ โดยที่ σ คือค่าส่วนเบี่ยงเบนมาตรฐานที่คำนวณได้จากสมการ $\sigma = \sqrt{\frac{\sum_{i=1}^{n}\left(x_i - \bar{x}\right)^2}{n-1}}$ เมื่อ n = 12 จะได้

σ = 38.92

(4) นำค่า σ ที่คำนวณได้ไปคำนวณต่อในสมการ $\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}}$ จะได้

$\sigma_{\bar{x}}$ = 11.23

แทนค่าตัวแปรทั้งหมด ในสมการ $UCL = \bar{\bar{x}} + z\sigma_{\bar{x}}$ โดยที่ z มีค่า เป็น 1,2 และ 3 เนื่องจากต้องการกำหนดค่าขอบเขตข้อมูลเป็น 3 ระดับ

UCL_1 = Round(227+1*11.23) = 238

UCL_2 = Round(227+2*11.23) = 249

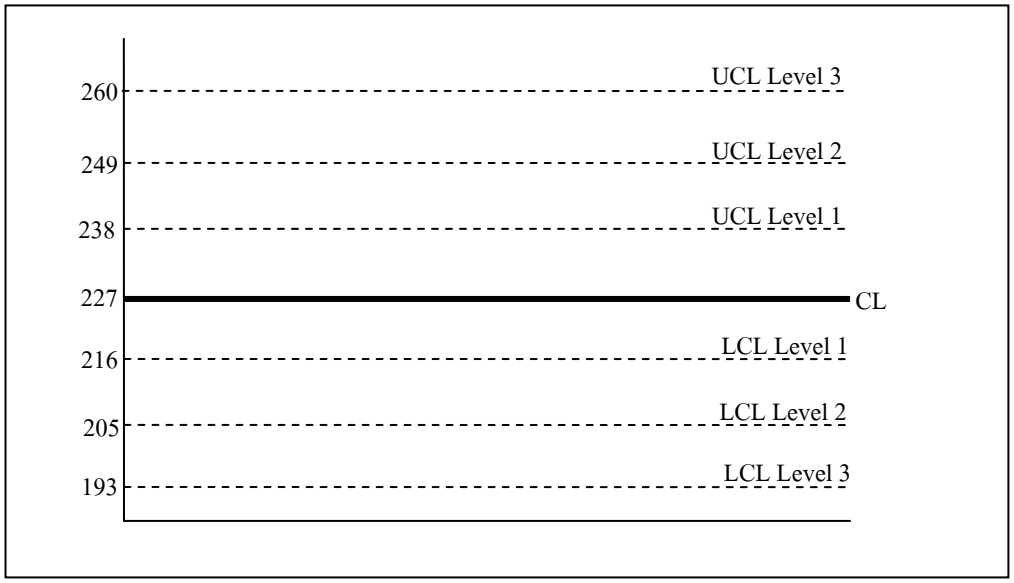
UCL_3 = Round(227+3*11.23) = 260

LCL_1 = Round(227-1*11.23) = 216

LCL_2 = Round(227-2*11.23) = 205

LCL_3 = Round(227-3*11.23) = 193

ค่าที่นำไปใช้เป็นเกณฑ์ในการวิเคราะห์ข้อมูลคือ $\bar{\bar{x}}$ = 227 โดยมีค่าข้อมูลที่อยู่ในขอบเขตจำกัดบนระดับ 1, 2 และ 3 คือ 238, 249 และ 260 ตามลำดับ และมีค่าข้อมูลที่อยู่ในขอบเขตจำกัดล่างระดับ 1, 2 และ 3 คือ 216, 205 และ 193 ตามลำดับ นำค่าที่ได้เก็บลงในฐานข้อมูล Control Chart สำหรับใช้ในขั้นตอนของ Data Analysis ต่อไป สามารถนำมาสร้างเป็น Control Chart ได้ดังภาพประกอบที่ 4-3



ภาพประกอบที่ 4-3 Control Chart สำหรับข้อมูลที่ได้จากการคำนวณ

ในกรณีที่คำนวณได้ค่า Lower Control Limit (LUL) ติดลบนั้น ในกรณีนี้จะถือว่าค่าขอบเขตจำกัดล่างเท่ากับ 0 เนื่องจากค่าข้อมูลของแพ็กเก็ตนั้นจะไม่มีค่าติดลบ

4.2.3 Data Analysis

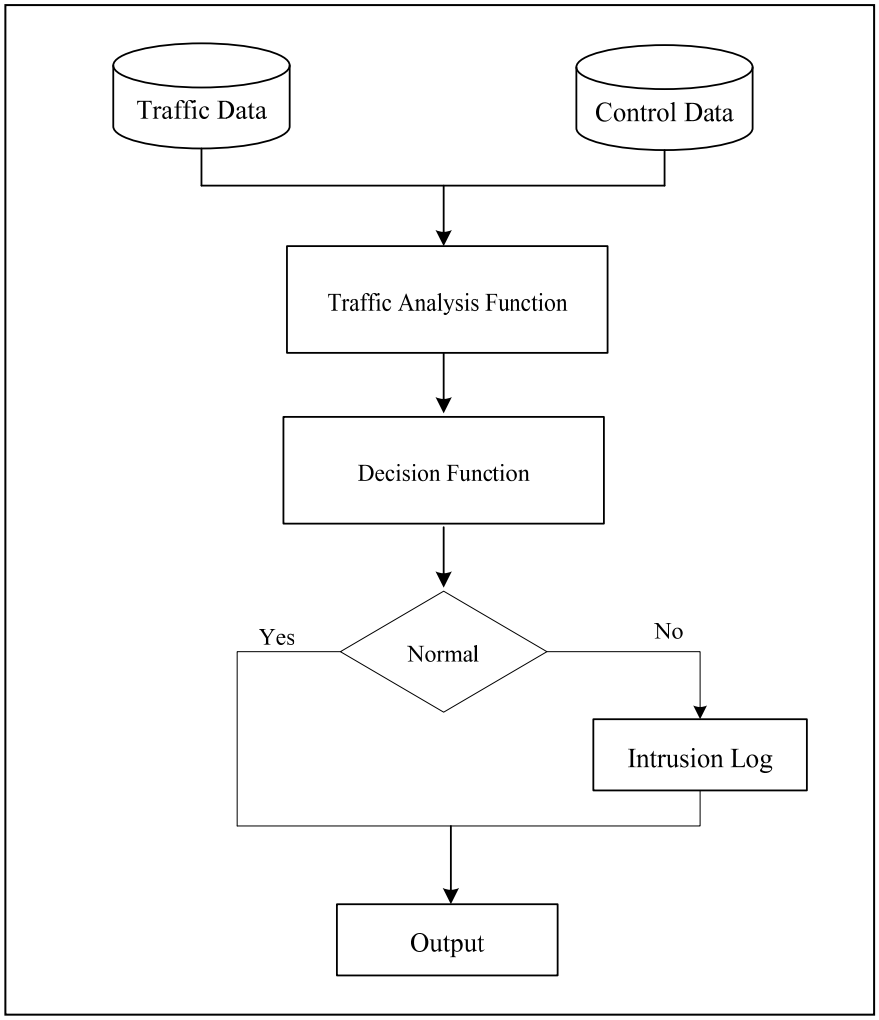
Data Analysis หรือกระบวนการวิเคราะห์ข้อมูลเพื่อตัดสินความผิดปกติหรือโดยเปรียบเทียบข้อมูลที่เข้ามากับค่าข้อมูลในโพรไฟล์ที่ได้มีการคำนวณไว้แล้ว ซึ่งในสภาวะปกตินั้นข้อมูลที่เข้ามาต้องมีค่าใกล้เคียงกับค่าเฉลี่ย (CL) หรืออยู่ในช่วงข้อมูลขอบเขตบนระดับที่ 1 (UCL Level 1) ในการวิเคราะห์ความผิดปกตินี้ผู้วิจัยได้ใช้แนวคิดซิกส์ซิกมา (6σ) ในการวิเคราะห์ ซึ่งเป็นหลักการในการแบ่งระดับเหตุการณ์ออกเป็นช่วงๆ หรือเป็นระดับโดยในแต่ละระดับจะบ่งบอกถึงค่าขอบเขตข้อมูลที่อยู่ห่างจากค่าเฉลี่ยที่ยอมรับได้ งานวิจัยนี้ได้ใช้ค่าระยะ 3 ซิกมา นั่นคือแบ่งระดับความผิดปกติออกเป็น 3 ช่วง คือ

- ช่วงข้อมูลปกติ (Normal) ระยะ 1 ซิกมา คือข้อมูลที่เข้ามานั้นมีปริมาณอยู่ระหว่าง CL และ UCL Level 1

- ช่วงข้อมูลระดับเกินกว่าปกติแต่ไม่ถึงกับวิกฤต (Risk) ระยะ 2 ซิกมาคือข้อมูลที่เข้ามานั้นมีปริมาณอยู่ในช่วง UCL Level 2

- ช่วงข้อมูลวิกฤต (Critical) ระยะ 3 ซิกมา คือข้อมูลที่เข้ามีปริมาณอยู่ในช่วง UCL Level 3 หรือมากกว่า

และหากพบว่าข้อมูลที่ได้ทำการวิเคราะห์นั้นอยู่ในช่วงของ Risk และ Critical นั้นระบบจะทำการบันทึกความผิดปกติดังกล่าวลงใน Log File เพื่อให้ผู้ดูแลระบบสามารถเรียกดูข้อมูลความผิดปกติย้อนหลังได้ แสดงภาพรวมของการทำงานดังภาพประกอบที่ 4-4



ภาพประกอบที่ 4-4 กระบวนการ Data Analysis

4.2.4 Output

ส่วนการแสดงผลจะเป็นการนำเสนอข้อมูลระหว่างข้อมูลปัจจุบันและข้อมูลโพรไฟล์ในช่วงเวลาที่ตรงกันในรูปแบบของกราฟแสดงบนเว็บบราวเซอร์เพื่อให้เห็นถึงแนวโน้มการใช้งานที่เกิดขึ้นในอดีตหรือการใช้งาน ณ ช่วงเวลาปัจจุบัน ทำให้สามารถดูเหตุการณ์ผิดปกติที่เกิดขึ้นได้ง่าย

4.3 สรุป

สำหรับบทนี้ได้นำเสนอการออกแบบระบบที่ใช้สำหรับตรวจจับความผิดปกติบนระบบคอมพิวเตอร์และเครือข่ายโดยใช้ SNMP เป็นเครื่องมือในการเก็บรวบรวมข้อมูล พร้อมทั้งอธิบายส่วนประกอบของระบบตรวจจับการบุกรุกที่ได้นำเสนออีกด้วย ซึ่งในบทต่อไปจะกล่าวถึงรายละเอียดในการพัฒนาระบบตามที่ได้ออกแบบไว้

บทที่ 5

การพัฒนาระบบ

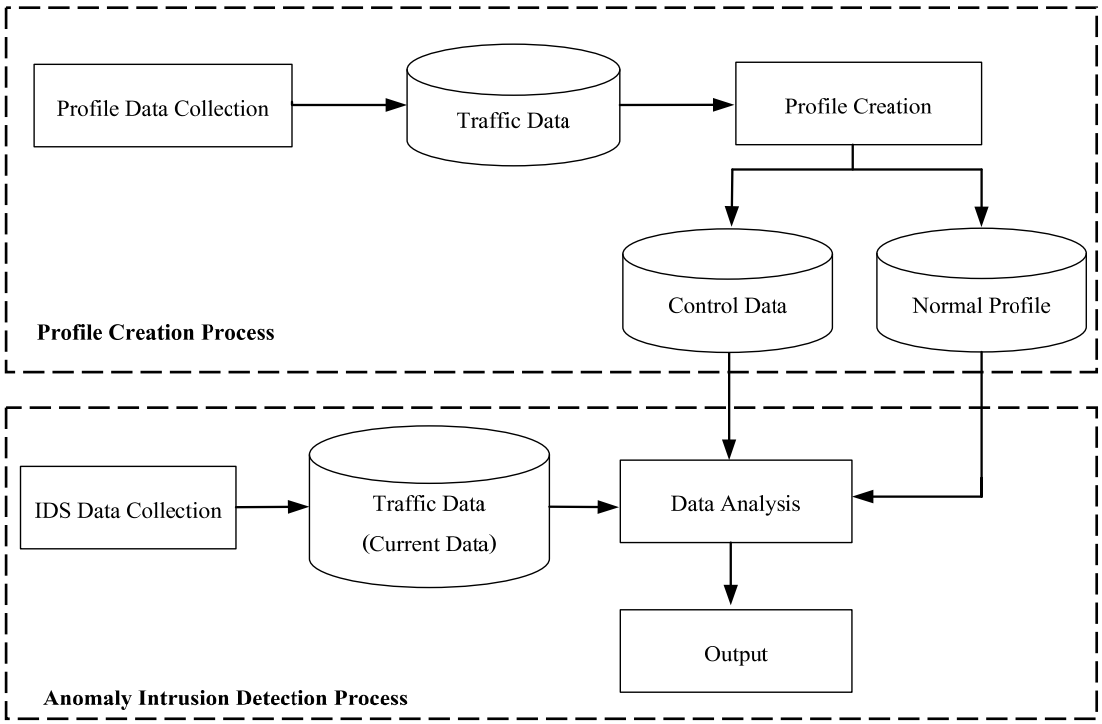
5.1 บทนำ

สำหรับบทนี้จะกล่าวถึงการพัฒนาระบบตรวจจับการบุกรุกที่ได้นำเสนอในบทก่อนหน้านี้ โดยจะกล่าวถึงภาษาที่ใช้ในการพัฒนาโปรแกรมและรายละเอียดในการพัฒนาโปรแกรมในส่วนของฟังก์ชันและโพรเซสการทำงานต่างๆ ในระบบ

5.2 ภาษาที่ใช้ในการพัฒนา

ในการพัฒนาระบบสำหรับวิทยานิพนธ์นี้ได้นำสิ่งที่กล่าวไว้ในบทก่อนหน้านี้มาดำเนินการพัฒนาระบบตรวจจับการบุกรุกโดยพัฒนาขึ้นบนระบบปฏิบัติการ FreeBSD เวอร์ชั่น 6.2 ใช้ภาษาซีเป็นภาษาหลักในการทำงานของโปรแกรม สำหรับส่วนของการแสดงผลนั้นผู้วิจัยได้เลือกใช้ภาษา PHP ในการแสดงผลผ่านทางเว็บบราวเซอร์

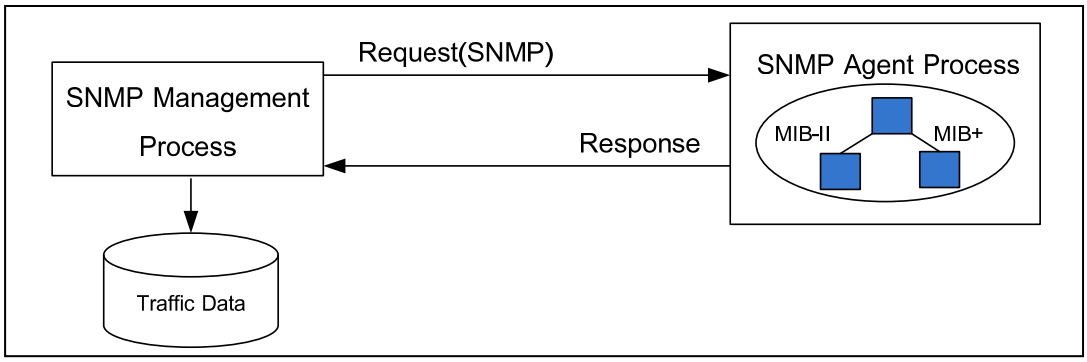
จากภาพรวมของระบบที่ได้ทำออกแบบไว้ดังภาพประกบที่ 5-1 นั้น ในบทนี้ผู้วิจัยจะทำการพัฒนาโพรเซสการทำงานของแต่โพรเซสโดยรายละเอียดในแต่ละโพรเซสจะกล่าวในหัวข้อถัดไป



ภาพประกอบที่ 5-1 ภาพรวมของระบบ

5.3 การพัฒนาโปรแกรมในส่วน Data Collection

การพัฒนาโปรแกรมในส่วนของ Data Collection นั้นจะมีกระบวนการเก็บรวบรวมข้อมูล 2 ขั้นตอนคือ Profile Data Collection และ IDS Data Collection ทั้งสองขั้นตอนนี้มีการทำงานดังภาพประกอบที่ 5-2 โดยทั้ง SNMP Agent Process ที่ต้องการสอบถามนั้น จะต้องติดตั้ง Net-SNMP ซึ่งมี MIB-II และเพิ่ม MIB+ ไว้เรียบร้อยแล้ว SNMP Management Process จะทำหน้าที่ในการส่งคำถามไปถามโหนดโดยใช้คำสั่ง snmpget



ภาพประกอบที่ 5-2 องค์ประกอบของกระบวนการ Data Collection

คำสั่งที่ใช้การสอบถามค่าอ็อบเจกต์ต่างๆ คือ

snmpget –v 2c –c public 127.0.0.1 IF-MIB::ifInOctets.1

snmpget –v 2c –c public 127.0.0.1 INTRUSION-DATA-MIB:: tcpInFinPkts.0

snmpget –v 2c –c public 127.0.0.1 INTRUSION-DATA-MIB:: tcpInNoFlagSetPkts.0

snmpget –v 2c –c public 127.0.0.1 INTRUSION-DATA-MIB:: tcpInSynPkts.0

snmpget –v 2c –c public 127.0.0.1 INTRUSION-DATA-MIB:: tcpInPushPkts.0

snmpget –v 2c –c public 127.0.0.1 INTRUSION-DATA-MIB:: tcpInUrgPkts.0

snmpget –v 2c –c public 127.0.0.1 INTRUSION-DATA-MIB:: tcpInNoFlagSetPkts.0

snmpget –v 2c –c public 127.0.0.1 INTRUSION-DATA-MIB:: udpInPortNumber53.0

เมื่อ IF-MIB และ INTRUSION-DATA-MIB คือ กลุ่มอ็อบเจกต์ที่ทำการสอบถามข้อมูล ซึ่งค่าที่โหนดหรืออุปกณ์นั้นตอบมาคือค่าที่มีชนิดข้อมูลเป็นจำนวนนับ (Counter) ในการนำข้อมูลมาใช้งานนั้นต้องนำมาคำนวณก่อน โดยใช้สูตร

$$\Delta d = \frac{(d_n - d_{n-1})}{(t_n - t_{n-1})60} \times 8$$

โดยที่ Δd คือผลต่างของข้อมูลที่ทำการสอบถาม

d_n คือค่าข้อมูล ณ ช่วงเวลาที่ n

d_{n-1} คือค่าข้อมูล ณ ช่วงเวลาที่ $n-1$

t_n คือช่วงเวลาที่ใช้สอบถาม ณ ช่วงเวลาที่ n

t_{n-1} คือช่วงเวลาที่ใช้สอบถาม ณ ช่วงเวลาที่ $n-1$

ผลต่างของ n ที่คำนวณได้นำมาคูณ 8 (เนื่องจากข้อมูลที่ได้มานั้นอยู่ในรูปของไบต์ต้องการให้อยู่ในรูปของบิต) และหารด้วยผลต่างของ t คูณด้วย 60 เพราะต้องการให้ได้ผลลัพธ์ออกมาในหน่วยของจำนวนบิตต่อวินาที (bit/sec) เมื่อคำนวณเรียบร้อยแล้วระบบจะบันทึกผลที่ได้เก็บลงในฐานข้อมูลชื่อว่า Traffic Data เพื่อให้โพรเซสในขั้นตอนถัดไปสามารถเรียกใช้งานข้อมูลดังกล่าวได้

ในส่วนของ Data Collection มีโพรเซสที่ทำงานในลักษณะ Daemon ซึ่งทำงานอยู่เบื้องหลัง (Background Process) ด้วยกัน 3 โพรเซส คือ

- **main-daemon**: โพรเซสที่เรียกใช้งาน MIB Module API ทำให้เราสามารถเรียกใช้ MIB Module หรือ MIB+ ที่สร้างขึ้นโดยจะทำงานภายใต้ Master Agent ซึ่งเป็นโพรเซสหลักของ Net-SNMP ซึ่งจะต้องมีการแก้ไขไฟล์ snmpd.conf ใน Net-SNMP เพื่อให้ Master AgentX ทำงานได้ด้วยการเพิ่ม “master agentx” ลงในไฟล์ snmpd.conf (ภาคผนวก ค.)

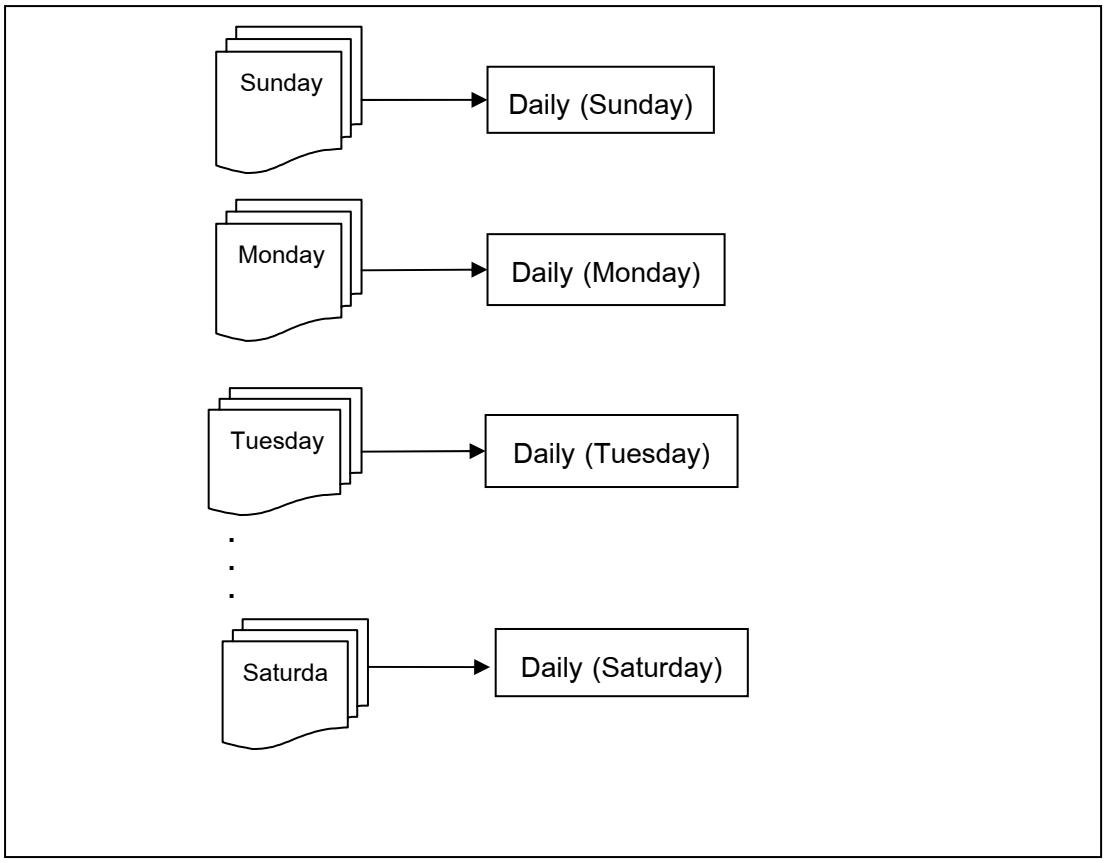
- **pkginfo-daemon**: เป็นโพรเซสที่ทำงานร่วมกับ Library ชื่อ Libpcap ในการเก็บข้อมูลของแพ็กเก็ตที่ผ่านเข้า-ออกอุปกรณ์ ซึ่งอุปกรณ์ที่ใช้สำหรับงานวิจัยนี้คือการ์ดแลน โดยที่ Libpcap จะคอยทำหน้าที่ในการดักจับข้อมูลแพ็กเก็ต เพื่อให้โพรเซส pkginfo-daemon ทำการเก็บและระบุค่าข้อมูลดังกล่าวให้กับอ็อบเจกต์ที่สร้างขึ้น

- **captureDataMib**: เป็นโพรเซสที่ใช้สำหรับสอบถามข้อมูลอ็อบเจกต์เพื่อนำมาสร้างเป็นโพรไฟล์ (Profile Creation) และใช้สำหรับเก็บข้อมูลเพื่อนำมาวิเคราะห์ความผิดปกติในขั้นตอน Data Analysis ด้วย โดยจะใช้คำสั่ง snmpget ในการสอบถามข้อมูลทุกๆ 5 นาที โดย อ็อบเจกต์ที่สอบถามนั้นจะประกอบมีทั้งหมด 8 อ็อบเจกต์ คือ ifInOctets, ipInAddSrcDest, tcpInFinPkts, tcpInSynPkts, tcpInUrgPkts, tcpInPushPkts, tcpInNoFlagSetPkts และ udpInPortNumber53 ตามที่ได้กล่าวไว้แล้วในบทที่ 4

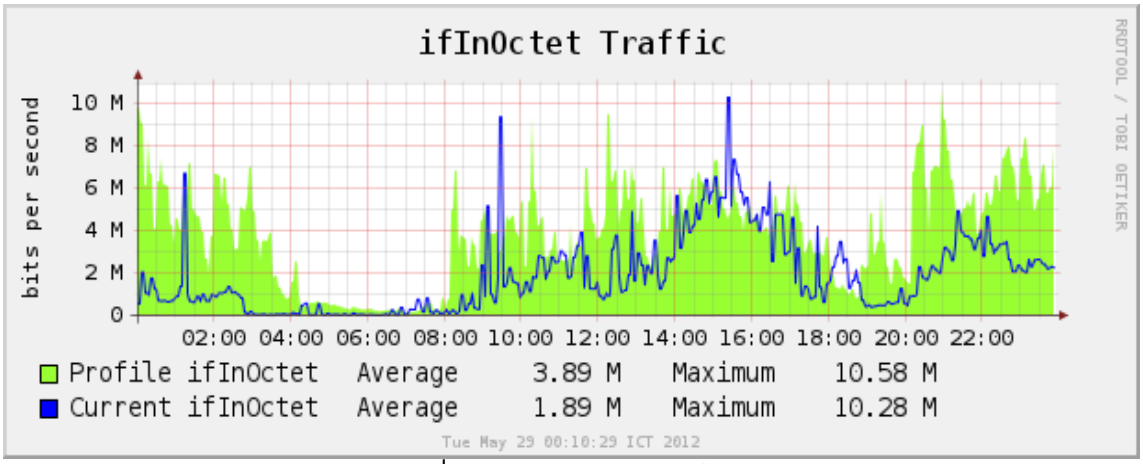
5.4 การพัฒนาโปรแกรมในส่วน Profile Creation

เมื่อระบบทำการบันทึกข้อมูลลงในฐานข้อมูล Traffic Data เรียบร้อยแล้ว ขั้นตอนถัดไปจะเข้าสู่กระบวนการสร้างโพรไฟล์ (Profile Creation) เพื่อหาค่าการใช้งานของเครือข่าย ซึ่งโพรไฟล์ที่สร้างนั้นมีด้วยกัน 3 รูปแบบคือ

1. โพรไฟล์รายวัน (Daily Profile) ประมวลผลข้อมูลในแต่ละวัน ผลลัพธ์ที่ได้คือค่าการใช้งานเฉลี่ยและส่วนเบี่ยงเบนมาตรฐานของแต่ละวันตั้งแต่วันจันทร์– อาทิตย์ ดังภาพประกอบที่ 5-3 โดยมีค่าข้อมูลที่เก็บในโพรไฟล์ประเภทนี้ดังภาพประกอบที่ 5-4



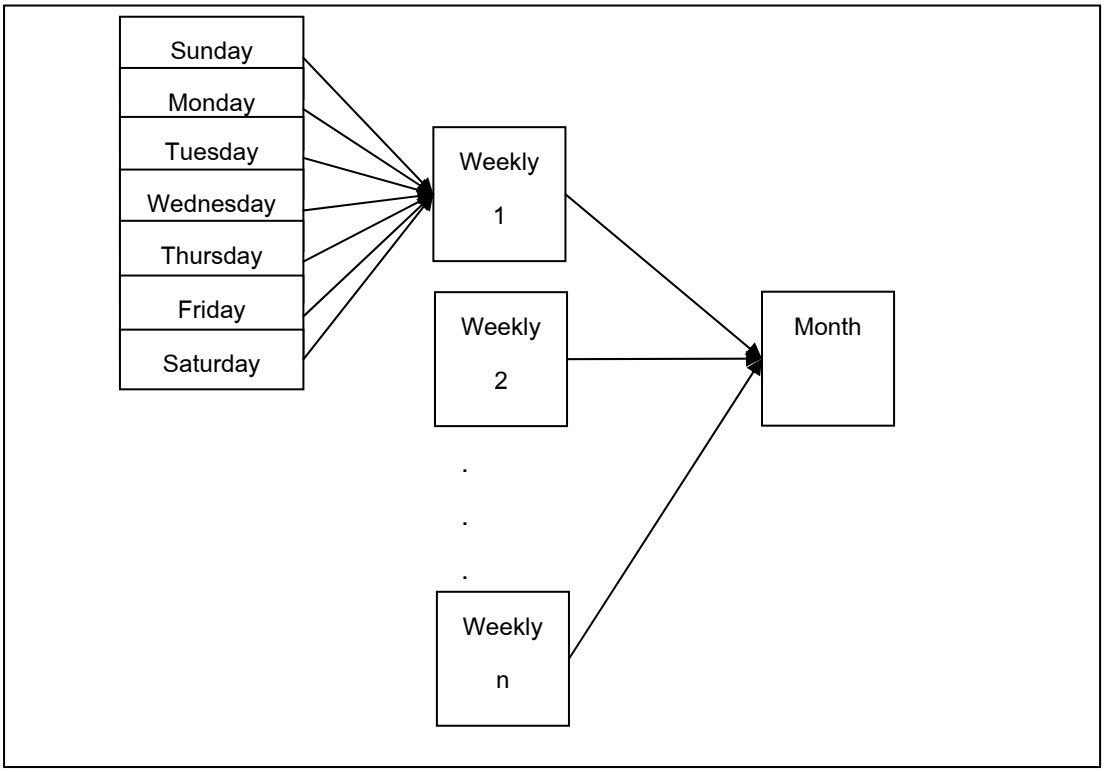
ภาพประกอบที่ 5-3 หลักการสร้างโพรไฟล์รายวัน



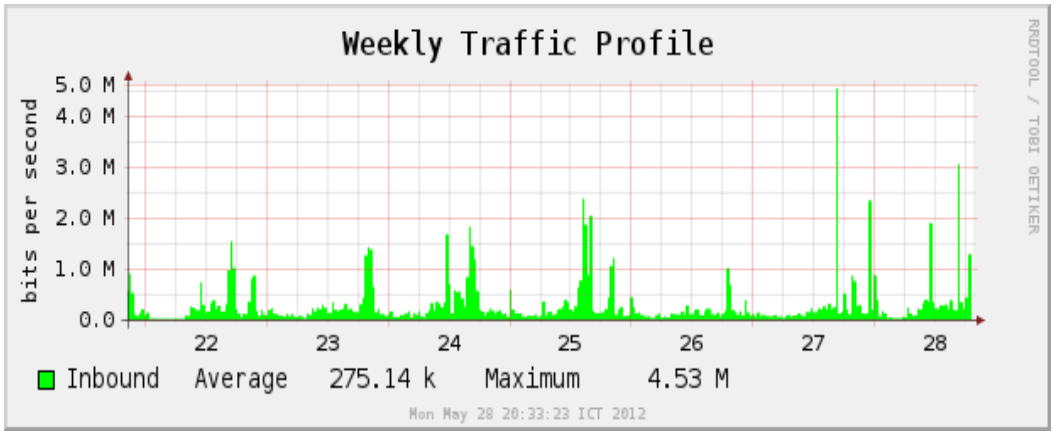
ภาพประกอบที่ 5-4 ค่าข้อมูลของโพรไฟล์รายวัน

2. โพรไฟล์รายสัปดาห์ (Weekly Profile) ประมวลผลข้อมูลรายสัปดาห์ตั้งแต่วันจันทร์-อาทิตย์ ผลลัพธ์ที่ได้คือค่าเฉลี่ยและส่วนเบี่ยงเบนมาตรฐานแต่ในละอาทิตย์แสดงในภาพประกอบที่ 5-5 โดยมีค่าข้อมูลที่เก็บในโพรไฟล์ประเภทนี้ดังภาพประกอบที่ 5-6

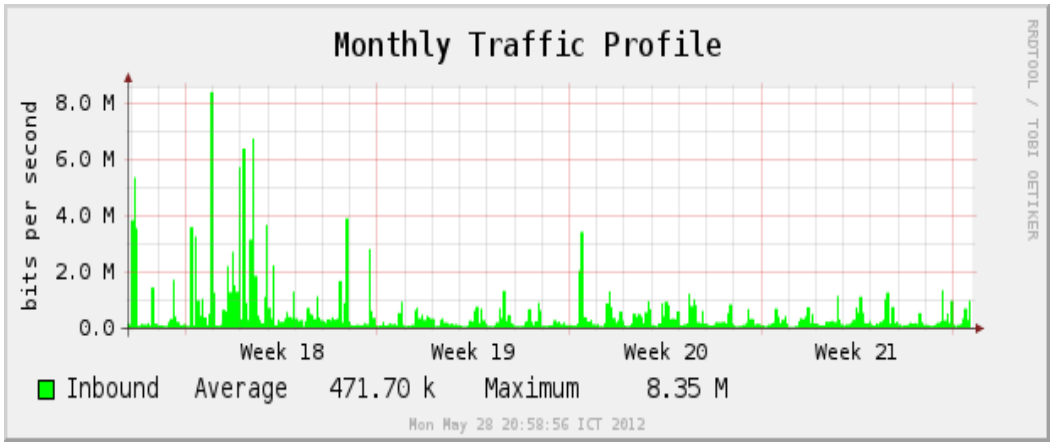
3. โพรไฟล์รายเดือน (Monthly Profile) ประมวลผลข้อมูลรายเดือนหรือดูการใช้งานในแต่ละเดือนว่ามีการใช้งานเป็นอย่างไร แสดงในภาพประกอบที่ 5-5 โดยมีค่าข้อมูลที่เก็บในโพรไฟล์ประเภทนี้ดังภาพประกอบที่ 5-7



ภาพประกอบที่ 5-5 หลักการสร้างโพรไฟล์รายสัปดาห์และรายเดือน



ภาพประกอบที่ 5-6 ค่าข้อมูลของโพรไฟล์รายสัปดาห์



ภาพประกอบที่ 5-7 ค่าข้อมูลของโพรไฟล์รายเดือน

ระบบที่พัฒนาขึ้นนั้นจะประกอบไปด้วยโพรไฟล์ที่ใช้สำหรับตรวจจับความผิดปกติ 5 ชนิด นั่นคือ โพรไฟล์ที่ใช้สำหรับตรวจจับ SYN Flood Attack, Land Attack, DNS Flood Attack, Null Scan Attack และ Xmas Scan Attack

5.5 การพัฒนาโปรแกรมในส่วน Data Analysis

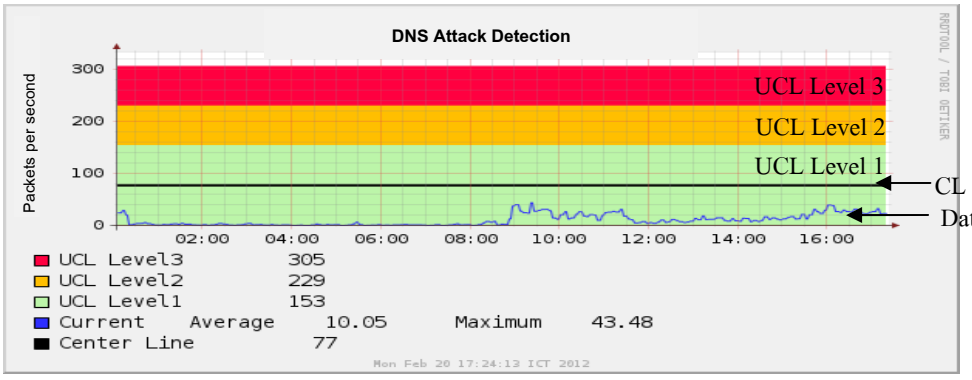
การพัฒนาโปรแกรมในส่วนของ Data Analysis จะมีข้อมูลที่เกี่ยวข้องคือ ข้อมูลใน Normal Profile, Control Data และข้อมูลเหตุการณ์ที่เกิดขึ้นในปัจจุบัน (Current Data) โดยมีฟังก์ชันการทำงานดังนี้

Traffic Analysis Function (TAF): เป็นฟังก์ชันในการคำนวณหาความเบี่ยงเบนของข้อมูลที่เกิดขึ้นโดยใช้ข้อมูลใน Normal Profile ในการวิเคราะห์ โดยใช้หลักการของซิกส์ซิกมา (6σ) (ภาคผนวก ก.)

Decision Function (DF): เป็นฟังก์ชันที่ใช้ในการตัดสินค่าข้อมูลที่ได้จากการคำนวณของฟังก์ชัน TAF โดยเทียบกับข้อมูลใน Control Data ว่ามีค่าข้อมูลอยู่ในระดับใด ซึ่งในที่นี้จะมี 3 ระดับ คือ

- Green แทนด้วยข้อมูลในช่วงปกติ (Normal) คือช่วงข้อมูลอยู่ในระดับที่ 1
- Yellow แทนด้วยข้อมูลอยู่ในระดับเกินกว่าปกติแต่ไม่ถึงกับวิกฤตหรือข้อมูลมีปริมาณสูงกว่าปกติ แต่อาจจะยังไม่ใช่เป็นการบุกรุก (Risk) คือช่วงข้อมูลอยู่ในระดับที่ 2 แต่ไม่ถึงระดับ 3
- Red แทนด้วยเหตุการณ์ที่ผิดปกติหรืออาจจะระบุได้ว่ามีการบุกรุกเกิดขึ้นเนื่องจากข้อมูลในขณะนั้นมีปริมาณเกินกว่าช่วงที่ได้กำหนดไว้ (Critical) คือช่วงข้อมูลที่อยู่ในระดับที่ 3 หรือเกินกว่านั้น

เพื่อให้ง่ายต่อการทำความเข้าใจผู้วิจัยจึงนำเสนอข้อมูลการวิเคราะห์นี้ด้วยกราฟ โดยใช้สีเป็นตัวแบ่งแยกระดับของข้อมูล ดังภาพประกอบที่ 5-8



ภาพประกอบที่ 5-8 ลักษณะข้อมูลที่เกิดขึ้นเมื่อเทียบกับโพรไฟล์ในระดับต่างๆ

Intrusion Log: เป็นฟังก์ชันการบันทึกข้อมูลลงใน Log File หากพบว่าผลลัพธ์ที่ได้จาก DF นั้นมีค่าเป็น Yellow และ Red

5.6 การพัฒนาโปรแกรมในส่วน Output

การพัฒนาในส่วน Output หรือส่วนรายงานผล ผู้วิจัยได้เลือกใช้เครื่องมือที่ชื่อว่า RRDtool ซึ่งเป็นเครื่องมือที่ใช้ในการจัดเก็บและแสดงผลข้อมูลตามช่วงเวลาที่มีความต่อเนื่อง RRDtool มีกระบวนการพิเศษที่ทำการรวบรวมข้อมูลดิบไปรวมกับข้อมูลที่ได้เก็บไว้แล้ว กระบวนการนี้จะทำให้ข้อมูลที่จัดเก็บมีขนาดเล็กลงเพื่อประหยัดพื้นที่ในการจัดเก็บ พร้อมทั้งยังมีฟังก์ชันในการรวมข้อมูล เช่น AVERAGE, MAXIMUM, MINIMUM และ LAST เพื่อให้สะดวกต่อการวิเคราะห์ข้อมูลอีกด้วย คำสั่งในการสร้างฐานข้อมูลใน RRDtool แสดงดังภาพประกอบที่ 5-9

```
rrdtool create graph.rrd--start 1000000000 --step 300 \
DS:udpport53Profile:GAUGE:600:0:4294967295 \
DS:udpport53Current:GAUGE:600:0:4294967295 \
RRA:AVERAGE:0.5:1:600 \
RRA:MIN:0.5:1:600 \
RRA:MAX:0.5:12:600;
```

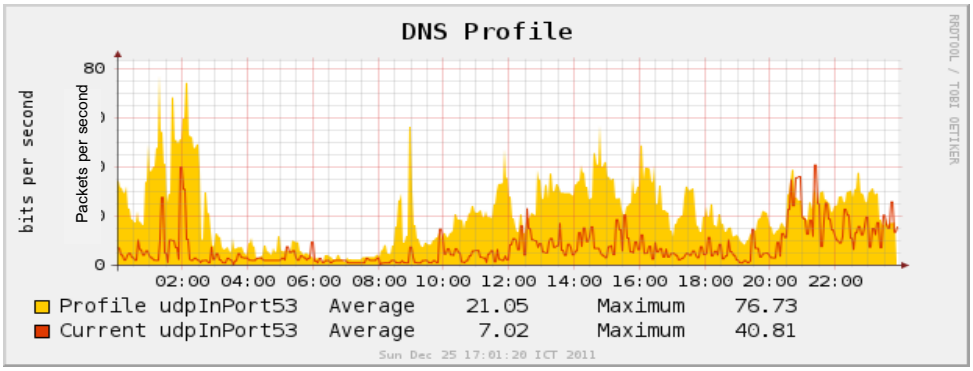
ภาพประกอบที่ 5-9 คำสั่งในการสร้างฐานข้อมูลใน RRDTool

คุณสมบัติพิเศษอีกอย่างของ RRDtool คือมีฟังก์ชันที่ใช้ในการสร้างกราฟ สามารถทำงานร่วมกับ Web Server เพื่อให้ผู้ใช้เรียกดูข้อมูลในรูปแบบกราฟผ่านทางเว็บบราวเซอร์ได้ ซึ่งกราฟที่สร้างขึ้นนั้นจะเป็นลักษณะของไฟล์รูปภาพ โดยคำสั่งในการสร้างกราฟแสดงดังภาพประกอบที่ 5-10

```
rrdtool graph dnsGraph.png --title \"DNS Profile \
--rigid --base=1000 --height=150 --width=500 \
--alt-autoscale-max --lower-limit=0 \
--vertical-label=\"packets per second\" --slope-mode \
--font TITLE:12: --font AXIS:8: --font LEGEND:10: --font UNIT:8: \
DEF:udpport53Profile=graph.rrd:udpport53Profile:AVERAGE    \
DEF:udpport53Current=graph.rrd:udpport53Current:AVERAGE    \
CDEF:cdef= udpport53Profile \
AREA:cdef#FFCC00:\"Profile udpInPort53\" \
GPRINT:cdef:AVERAGE:\"Average %8.2lf \%s \" \
GPRINT:cdef:MAX:\"Maximum %8.2lf %s\" \
CDEF:klmn= udpport53Current \
LINE1:klmn#DF3A01:\"Current udpInPort53\" \
GPRINT:klmn:AVERAGE:\"Average %8.2lf \%s \" \
GPRINT:klmn:MAX:\"Maximum %8.2lf %s\"
```

ภาพประกอบที่ 5-10 คำสั่งในการสร้างกราฟของ RRDTool

โดยในภาพประกอบที่ 5-11 ได้แสดงตัวอย่างของกราฟที่สร้างจาก RRDtool เพื่อให้เห็นปริมาณการใช้งานเครือข่ายระหว่าง Normal Profile กับข้อมูลปัจจุบัน



ภาพประกอบที่ 5-11 Normal Profile กับข้อมูลปัจจุบันของ DNS

5.7 สรุป

ในบทนี้ได้กล่าวถึงการพัฒนาโปรแกรมสำหรับใช้ตรวจจับการบุกรุก โดยใช้ภาษาซีในการพัฒนาโปรแกรม ซึ่งตัวระบบเองประกอบด้วยฟังก์ชันในการทำงานหลัก 4 ฟังก์ชันคือ Data Collection เป็นส่วนของการจัดเก็บรวบรวมข้อมูล Profile Creation เป็นส่วนของการสร้างโพรไฟล์ Data Analysis เป็นส่วนของการวิเคราะห์ความผิดปกติบนเครือข่าย และส่วนของ Output เป็นการนำเสนอผลลัพธ์จากการวิเคราะห์ข้อมูลโดยระบบ และรายงานผลความผิดปกติที่เกิดขึ้น ในบทต่อไปจะเป็นการกล่าวถึงการทดสอบโปรแกรมที่พัฒนาขึ้น

บทที่ 6

การทดสอบระบบตรวจจับการบุกรุก

6.1 บทนำ

บทนี้จะกล่าวถึงการทดสอบและการวัดประสิทธิภาพของระบบตรวจจับการบุกรุกที่พัฒนาขึ้น โดยในส่วนแรกจะกล่าวถึงสภาพแวดล้อมในการทดสอบระบบ ต่อมาจะกล่าวถึงวิธีการทดสอบและตรวจจับความผิดปกติที่เกิดขึ้น และหลังจากนั้นจะเป็นการกล่าวถึงประสิทธิภาพการทำงานของระบบโดยมีรายละเอียดตามลำดับดังต่อไปนี้

6.2 สภาพแวดล้อมในการทดสอบระบบ

สภาพแวดล้อมในการทดสอบระบบครั้งนี้ผู้วิจัยได้ทำการเก็บข้อมูลของเครือข่ายภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยสงขลานครินทร์ ซึ่งเครือข่ายที่ใช้ในการทดสอบแสดงดังภาพประกอบที่ 6-1 ซึ่งทำการติดตั้งโปรแกรมต่างๆ ลงในเครื่องคอมพิวเตอร์ส่วนบุคคลโดยแสดงรายละเอียดและหน้าที่จำเป็นของคอมพิวเตอร์ดังนี้

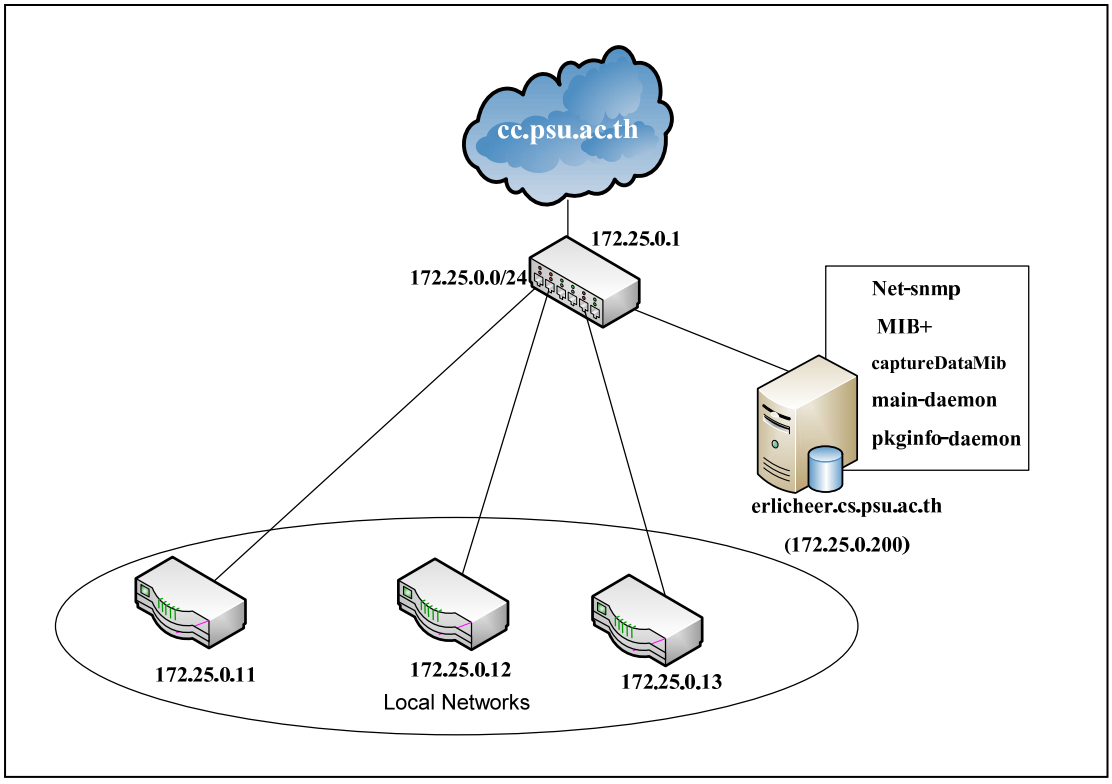
เครื่องคอมพิวเตอร์ที่ใช้งานชื่อว่า erlicheer.cs.psu.ac.th ทำหน้าที่เป็น Network Intrusion Detection System (NIDS) ของเครือข่ายรายละเอียดของระบบปฏิบัติการแสดงดังตารางที่ 6-1 และติดตั้ง

- Net-SNMP เป็นซอฟต์แวร์เอเจนท์ที่ทำงานร่วมกับโพรโทคอล SNMP
- MIB+ เป็นโมดูลของ MIB ที่สร้างขึ้นเพื่อการเรียกดูข้อมูลอ็อบเจกต์ใหม่ที่เพิ่มขึ้น
- main-daemon เป็นโปรแกรมที่ช่วยให้โมดูลเอเจนท์ที่สร้างขึ้นทำงานร่วมกับ Master Agent ของ Net-SNMP ได้
- pkginfo-daemon เป็นเอเจนท์ที่ทำหน้าที่ในการกำหนดและระบุค่าให้กับอ็อบเจกต์ใน MIB+
- captureDataMib เป็นโปรแกรมที่ทำหน้าที่เก็บข้อมูลของอ็อบเจกต์

- ifInOutDetection เป็นโปรแกรมที่ทำหน้าที่ตรวจจับเหตุการณ์ความผิดปกติที่เกิดขึ้น

ตารางที่ 6-1 รายละเอียดระบบปฏิบัติการของเครื่อง erlicheer.cs.psu.ac.th

Machine Processor Architecture Name	i386
Operating System Name	FreeBSD
Operating System Release	6.4
Operating System Version	root@erlicheer.cs.psu.ac.th:/usr/src/sys/i386/compile/NATHOST
CPU	Intel ® Celeron ® 2.40 GHz
Total/Avail Memory	256 MB



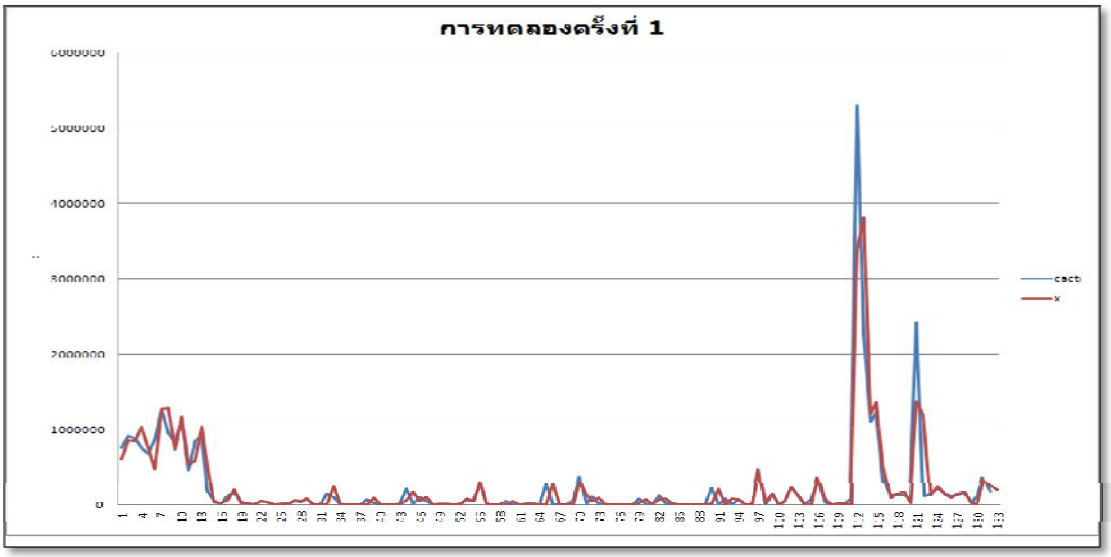
ภาพประกอบที่ 6-1 สภาพแวดล้อมในการทดสอบระบบ

6.3 การทดสอบระบบตรวจจับการบุกรุก

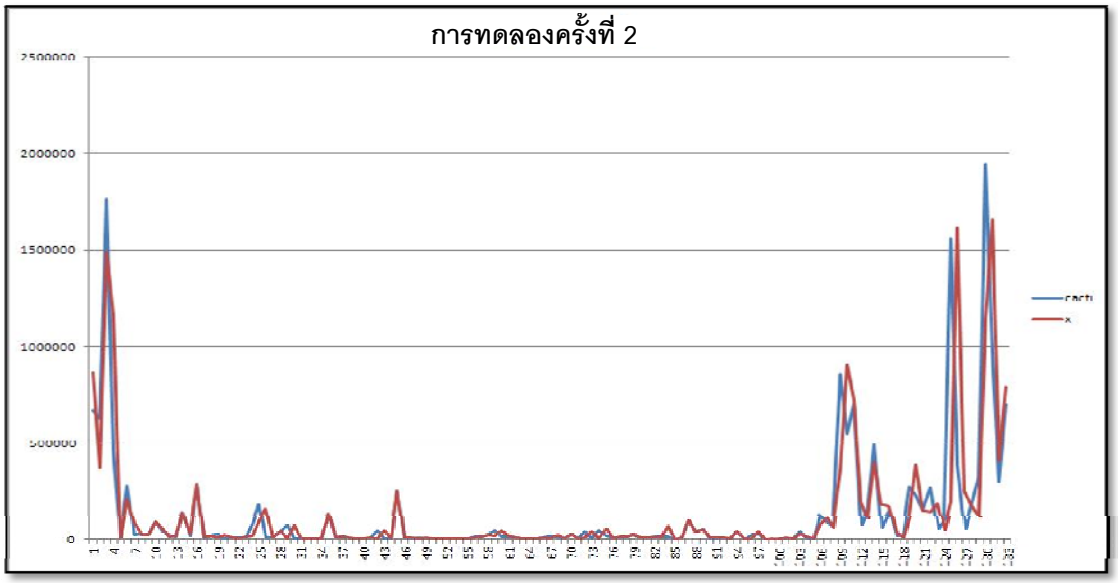
การทดสอบระบบตรวจจับการบุกรุกเป็นการทดสอบความสามารถในการตรวจจับความผิดปกติที่เกิดขึ้น โดยการทดสอบนี้จะใช้โปรแกรมโจมตีหรือเครื่องมือที่มีอยู่บนอินเทอร์เน็ต ซึ่งมีข้อจำกัดและวิธีทดสอบการโจมตีต่างๆ ดังนี้

6.3.1 ข้อจำกัดในการทดสอบ

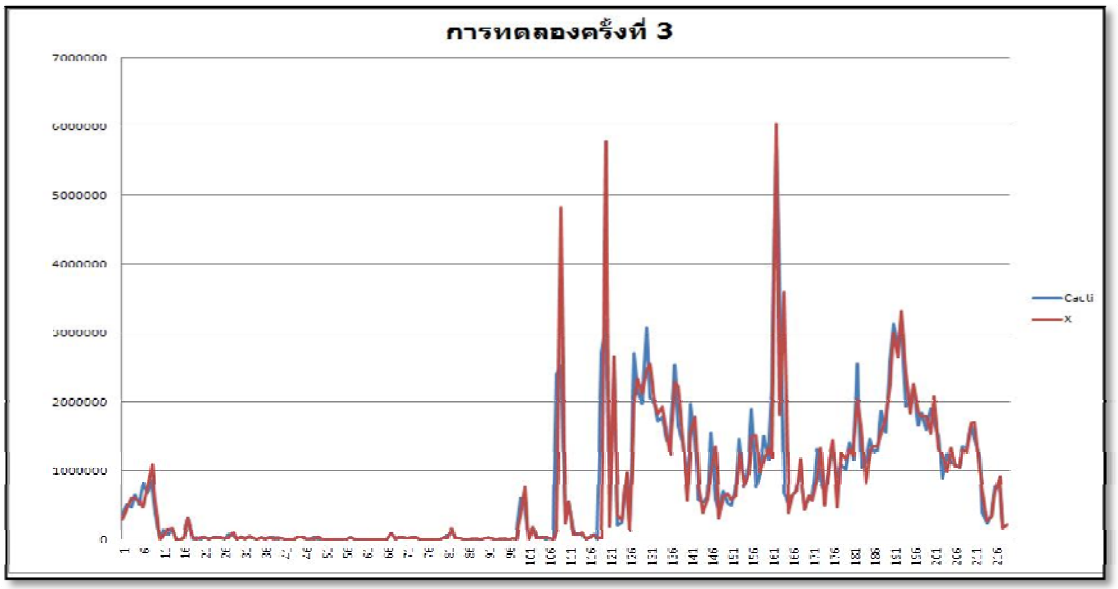
ระบบที่พัฒนาขึ้นนี้เป็นแบบ Network-based IDS (NIDS) ใช้ตรวจจับความผิดปกติที่เกิดขึ้นในเครือข่ายเท่านั้น และเนื่องจากผู้วิจัยไม่สามารถติดตั้งโปรแกรมดังกล่าวลงในเราท์เตอร์ของเครือข่ายได้ ดังนั้นในการเก็บข้อมูลของภาควิชาจึงใช้วิธีการทำ Mirror Port เพื่อคัดลอกข้อมูลเครือข่ายภาควิชามายัง NIDS ของผู้วิจัย ทำให้ข้อมูลที่ผ่านเข้า-ออกของเครือข่ายเท่ากับข้อมูลเข้าที่ผ่านอินเทอร์เฟสของ NIDS ฉะนั้นข้อมูลที่นำมาพิจารณาเพื่อหาความผิดปกติของเครือข่ายนั้นจะพิจารณาเฉพาะข้อมูล ifInOctets และเพื่อพิสูจน์ว่าสมมุติฐานข้างต้นเป็นจริง ผู้วิจัยจึงได้ทำการทดลองเก็บข้อมูล ifInOctets ของอินเทอร์เฟส NIDS เทียบกับข้อมูล IfInOctets + ifOutOctets ของอินเตอร์เฟสที่เป็น Gateway ภาควิชาซึ่งผลการทดลองแสดงดังภาพประกอบที่ 6-2 (a)-(c)



ภาพประกอบที่ 6-2 (a) การเปรียบเทียบข้อมูลระหว่าง NIDS กับ Gateway ครั้งที่ 1



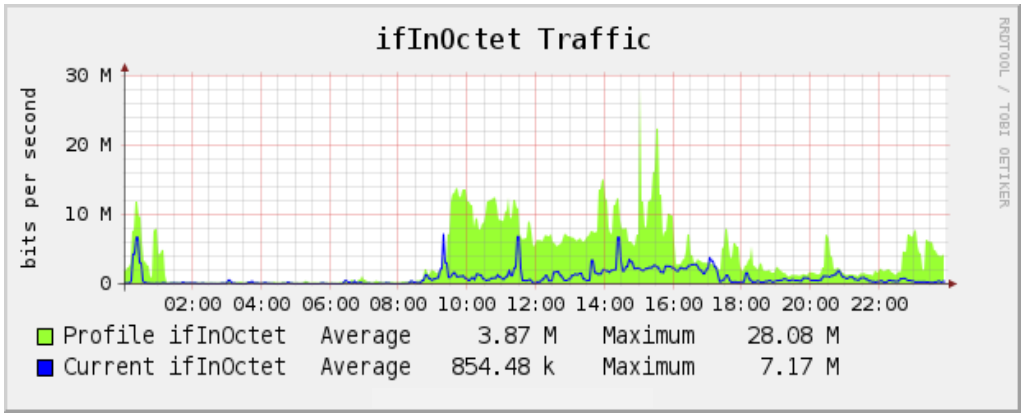
ภาพประกอบที่ 6-2 (b) การเปรียบเทียบข้อมูลระหว่าง NIDS กับ Gateway ครั้งที่ 2



ภาพประกอบที่ 6-2 (c) การเปรียบเทียบข้อมูลระหว่าง NIDS กับ Gateway ครั้งที่ 3

จากผลการทดลองข้างต้นคิดเป็นเปอร์เซ็นความผิดพลาดของข้อมูลที่จัดเก็บโดยอินเทอร์เฟสของ NIDS และ Gateway อยู่ที่ 0.27% ซึ่งถือว่าเป็นความผิดพลาดที่ไม่มีผลต่อการพิจารณาในเรื่องของการโจมตีบนเครือข่าย เพราะการโจมตีบนระบบคอมพิวเตอร์บนเครือข่ายนั้นจำนวนข้อมูลที่เข้ามาจะต้องมีปริมาณมากกว่าค่าความผิดพลาดที่ได้และเกิดขึ้นในเวลาอันรวดเร็ว

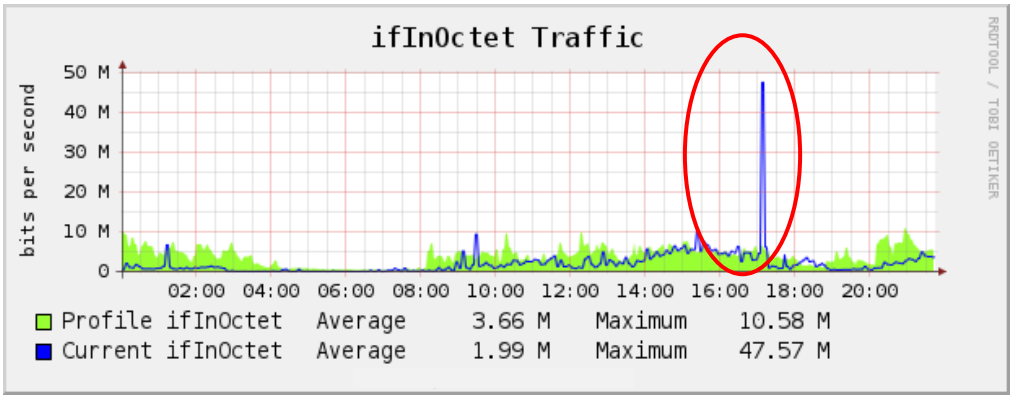
ในการตรวจจับความผิดปกติบนเครือข่ายนั้น สามารถดูได้จากปริมาณของข้อมูลที่เพิ่มขึ้นในเวลาอันรวดเร็ว เพราะเนื่องจากในสภาวะการใช้งานที่เป็นปกตินั้นแนวโน้มของปริมาณการใช้งานในแต่ละช่วงเวลาจะไม่มีความแตกต่างกันมาก ดังภาพประกอบที่ 6-3



ภาพประกอบที่ 6-3 ปริมาณข้อมูลที่เป็นปกติเทียบกับโพรไฟล์

6.3.2 วิธีทดสอบการตรวจจับความผิดปกติบนเครือข่าย

ในการทดสอบตรวจจับความผิดปกติบนเครือข่ายนั้นผู้วิจัยได้สร้างสถานการณ์สำหรับทดสอบโดยทดลองโจมตีเครือข่าย แสดงให้เห็นข้อมูลดังภาพประกอบที่ 6-4

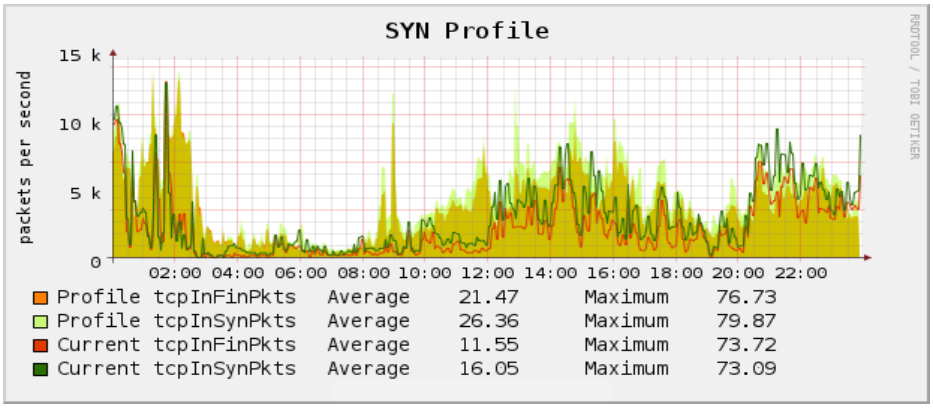


ภาพประกอบที่ 6-4 ปริมาณข้อมูลที่ถูกโจมตีเทียบกับโพรไฟล์

จะเห็นได้ว่ามีปริมาณข้อมูลเพิ่มสูงกว่าปกติ เบื้องต้นสามารถทราบได้ว่ามีความผิดปกติเกิดขึ้นบนเครือข่ายโดยดูจากภาพรวมของเครือข่ายที่มีปริมาณข้อมูลสูงกว่าโพรไฟล์ แต่ไม่สามารถทราบได้ว่า ความผิดปกติที่เกิดขึ้นนั้นเป็นความผิดปกติรูปแบบใด สามารถนำ MIB+ เพื่อใช้เป็นพารามิเตอร์สำหรับสร้างโพรไฟล์ในการระบุความผิดปกติที่เกิดขึ้นได้ สำหรับการทดลองนี้ผู้วิจัยได้ทดลองตรวจจับความผิดปกติ 5 รูปแบบ ซึ่งถ้าหากความผิดปกติที่เกิดขึ้นนั้นไม่ตรงกับเงื่อนไขของโพรไฟล์ทั้ง 5 รูปแบบนี้แล้วระบบจะถือว่าเป็นการโจมตีที่ไม่รู้จัก

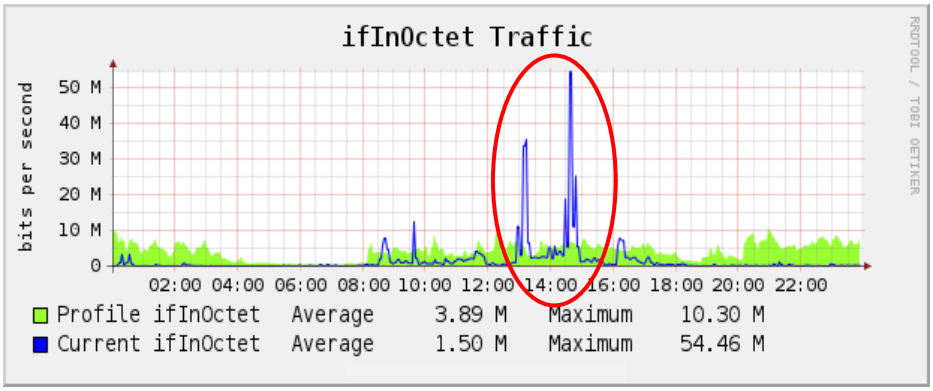
6.3.3 วิธีทดสอบการตรวจจับ SYN Flood Attack

ลักษณะโปรแกรมที่ใช้ทดสอบเป็นโปรแกรมที่อาศัยจุดอ่อนของโพรโทคอล TCP/IP ในการขอการเชื่อมต่อ โดยโปรแกรมที่ใช้ทดสอบการโจมตีนี้จะส่งคำร้องขอการเชื่อมต่อโดยใช้ TCP ที่ตั้งค่า Flag เป็น SYN จำนวนมากไปยังเครื่องเป้าหมาย โดยที่โพรไฟล์ปกติของ SYN และ FIN จะเป็นดังภาพประกอบที่ 6-5

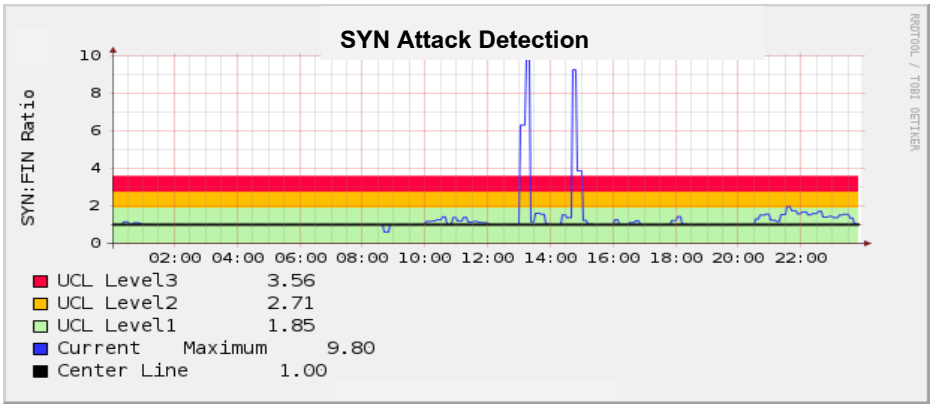


ภาพประกอบที่ 6-5 โพรไฟล์ปกติของ SYN และ FIN

ในการทดสอบครั้งนี้ผู้วิจัยได้ทำการโจมตี SYN Flood ไปยังเครื่องเป้าหมายจำนวน 5 ครั้ง เป็นเวลา ครั้งละ 10 นาที ได้ผลการทดสอบดังภาพประกอบที่ 6-6 จะเห็นได้ว่าปริมาณข้อมูลเครือข่ายในภาพรวมเพิ่มสูงขึ้น และเมื่อดูในรายละเอียดของโพรไฟล์ SYN เห็นได้ว่าอัตราส่วนของ TCP Flag SYN มีมากกว่า FIN ดังภาพประกอบที่ 6-7



ภาพประกอบที่ 6-6 ปริมาณข้อมูลเมื่อโจมตีแบบ SYN Flood



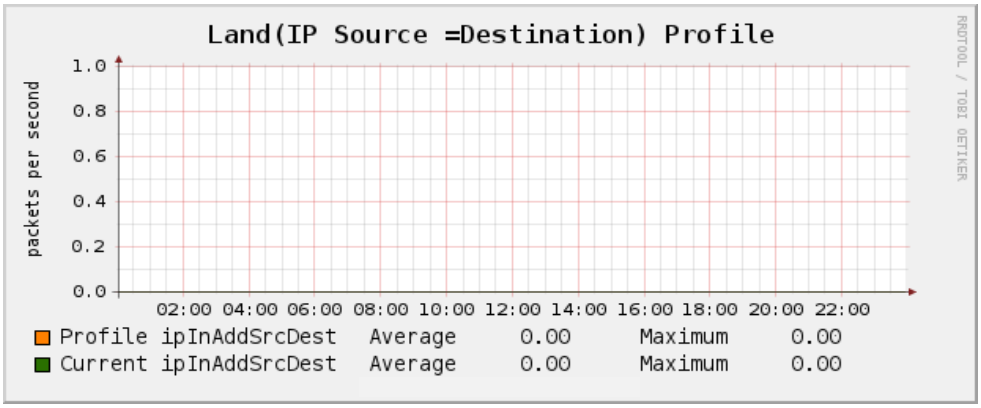
ภาพประกอบที่ 6-7 ผลการทดสอบการตรวจจับ SYN Flood Attack

6.3.4 วิธีทดสอบการตรวจจับ Land Attack

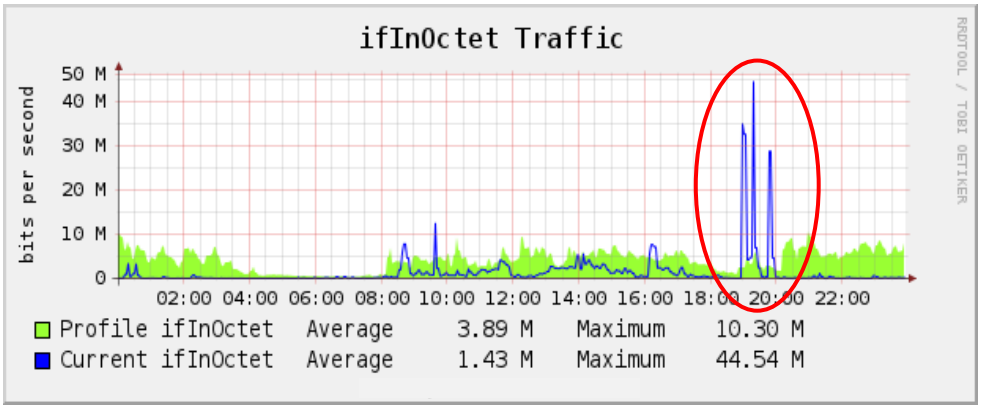
ลักษณะโปรแกรมที่นำมาทดสอบ เป็นโปรแกรมที่สามารถปลอมหมายเลข IP พร้อมทั้งส่งข้อมูลแพ็กเก็ต TCP เพื่อร้องขอการเชื่อมต่อเป็นจำนวนมากไปยังเครื่องเป้าหมายพร้อมกัน ซึ่งเมื่อเครื่องเป้าหมายได้รับแพ็กเก็ตชนิดนี้ ก็จะส่งแพ็กเก็ตตอบกลับเข้าหาตัวเอง โดยในภาพประกอบที่ 6-8 แสดงถึงโพรไฟล์ปกติของแพ็กเก็ตที่มีหมายเลข IP ต้นทางและปลายทางเหมือนกัน จะเห็นได้ว่าไม่พบแพ็กเก็ตประเภทนี้ในสภาวะการทำงานของเครือข่ายที่เป็นปกติเลย

การทดสอบครั้งนี้ผู้วิจัยได้ทำการโจมตีโดยปลอมหมายเลข IP ต้นทางให้เหมือนกับปลายทาง พร้อมกับส่งแพ็กเก็ตจำนวนมากไปยังเครื่องเป้าหมายจำนวน 5 ครั้ง ครั้งละ 10 นาที ผลการทดสอบพบว่ามีปริมาณข้อมูลในเครือข่ายเพิ่มสูงขึ้น และ IP แพ็กเก็ตที่มี

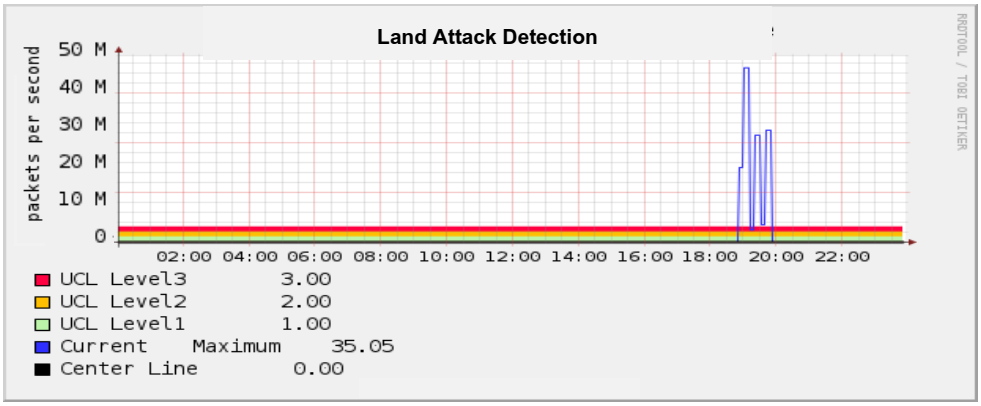
หมายเลข IP ต้นทางเหมือนกับหมายเลข IP ปลายทางเพิ่มขึ้นด้วย แสดงดังภาพประกอบที่ 6-9 และ 6-10 ตามลำดับ



ภาพประกอบที่ 6-8 โพรไฟล์ปกติของแพ็กเก็ตที่มีหมายเลข IP ต้นทางและปลายทางเหมือนกัน



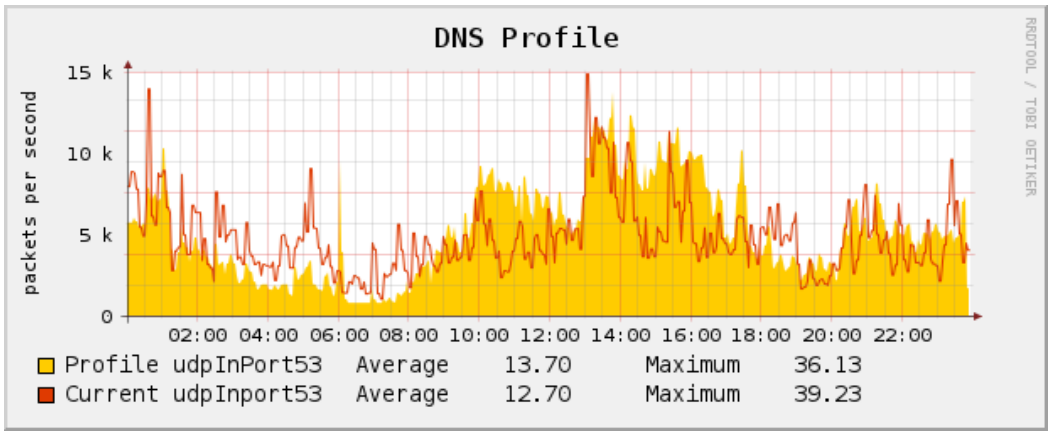
ภาพประกอบที่ 6-9 ปริมาณข้อมูลเมื่อโจมตีแบบ Land Attack



ภาพประกอบที่ 6-10 ผลการทดสอบการตรวจจับโจมตี Land Attack

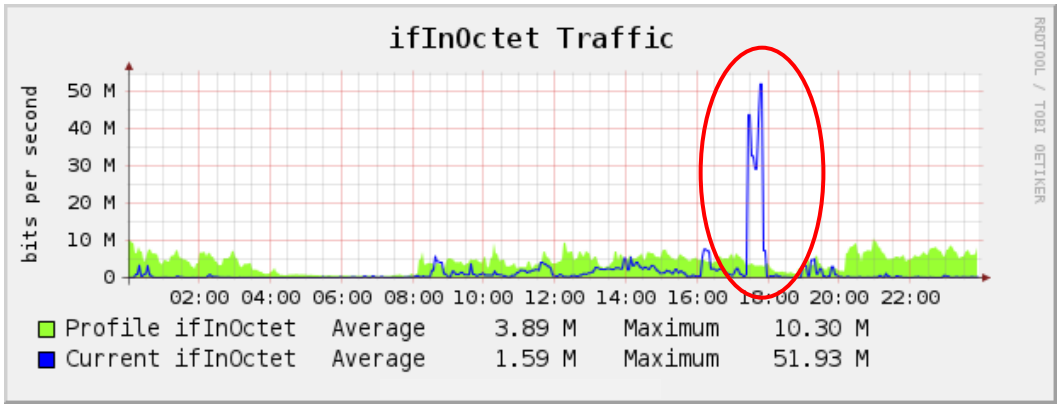
6.3.5 วิธีทดสอบการตรวจจับ DNS Flood Attack

ลักษณะโปรแกรมที่นำมาทดสอบเป็นโปรแกรมที่สามารถส่ง DNS Request ไปยัง DNS Server โดยใช้โพรโทคอล UDP ผ่าน พอร์ต 53 โดยการส่งคำร้องขอไปยัง DNS Server ปริมาณมากในเวลาอันรวดเร็ว ซึ่งในสภาวะปกติมีจำนวนหรือแนวโน้มของแพ็กเก็ต UDP พอร์ต 53 ดังภาพประกอบที่ภาพประกอบที่ 6-11

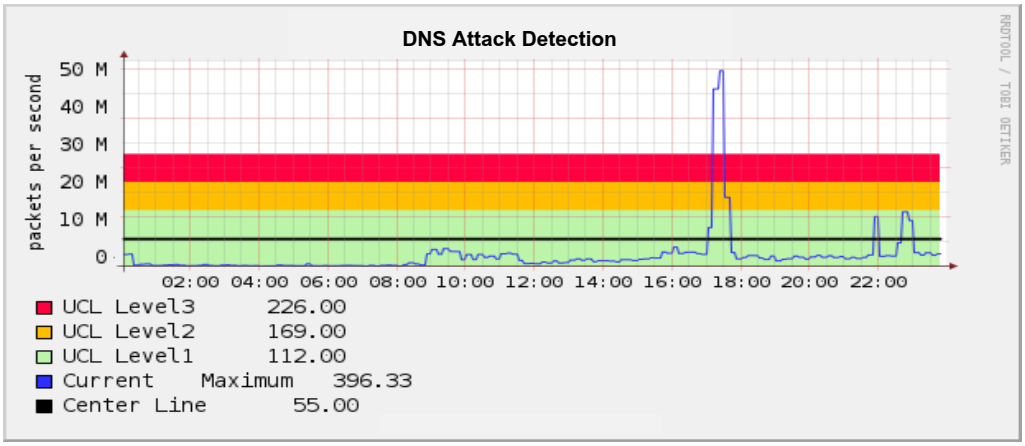


ภาพประกอบที่ 6-11 โพรไฟล์ปกติของ DNS

สำหรับการทดสอบครั้งนี้ผู้วิจัยได้ทำการโจมตี DNS โดยส่งแพ็กเก็ต UDP พร้อมทั้งปลอม URL ที่คาดว่าไม่มีอยู่จริงคือ http://www.attackdnsxzy.com ส่งไปยัง DNS Server จำนวนมาก ในเวลา 10 นาที พบว่าปริมาณข้อมูลในเครือข่ายเพิ่มสูงขึ้น และโพรโทคอล UDP ที่มีหมายเลขพอร์ตเท่ากับ 53 หรือ โพรโทคอล DNS นั้นมีปริมาณเพิ่มสูงขึ้นกว่าปกติด้วย เมื่อเทียบกับโพรไฟล์ ดังภาพประกอบที่ 6-12 และ 6-13 ตามลำดับ



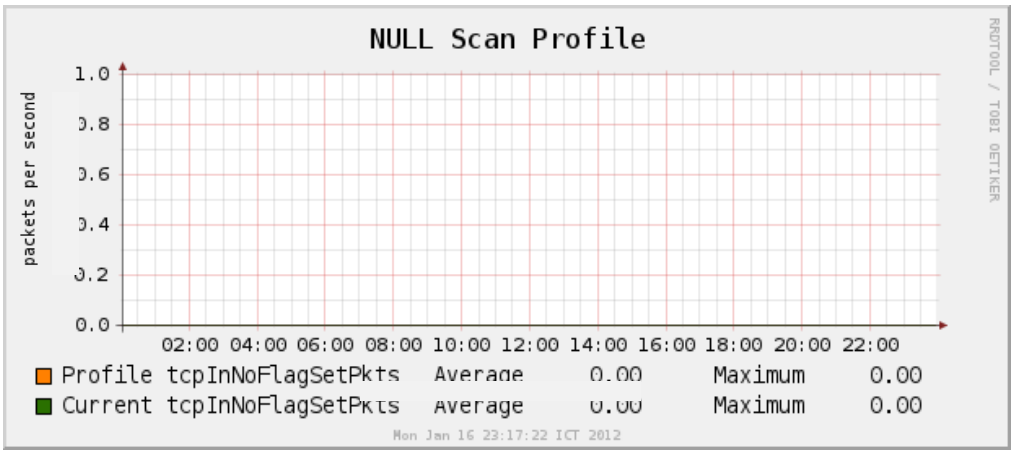
ภาพประกอบที่ 6-12 ปริมาณข้อมูลเมื่อโจมตี DNS Flood



ภาพประกอบที่ 6-13 ผลการทดสอบการตรวจจับ DNS Flood Attack

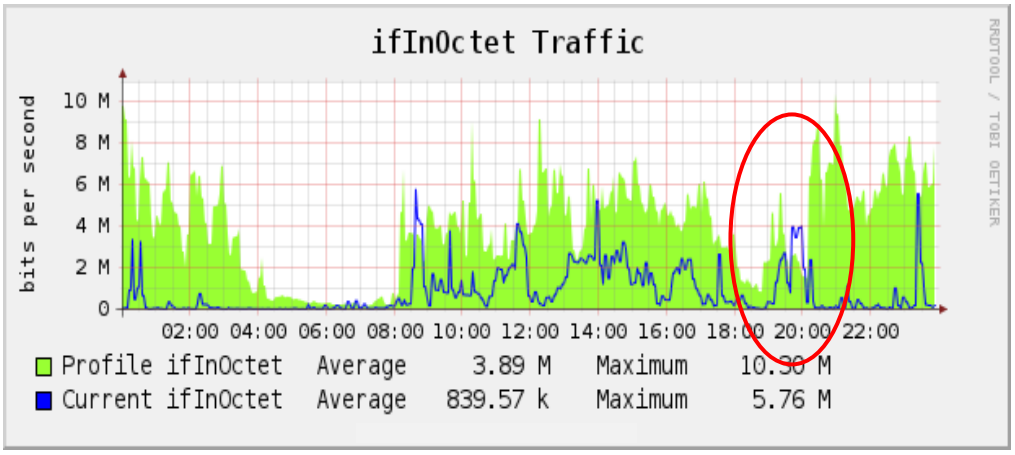
6.3.6 วิธีทดสอบการตรวจจับ Null Scan

สำหรับเครื่องมือที่ใช้ในการทดสอบ Null Scan นั้น ผู้วิจัยเลือกใช้ Nmap ซึ่งเป็นโปรแกรมที่ใช้ในการสำรวจเครือข่ายเพื่อใช้ตรวจดูว่ามีเครื่องหรืออุปกรณ์ใดทำงานอยู่ หรือเปิดให้บริการอะไร โปรแกรมนี้ยังช่วยในการตรวจหาช่องโหว่ของเครือข่ายได้อีกด้วย ผู้โจมตีสามารถนำโปรแกรมนี้ไปใช้ในการเก็บรวบรวมข้อมูลเพื่อหาช่องโหว่ของเครือข่ายเป้าหมายได้เช่นเดียวกัน ภาพประกอบที่ 6-14 แสดงถึงโพรไฟล์ของ Null Scan จะเห็นได้ว่าข้อมูลของแพ็กเก็ต TCP ที่ไม่มีการตั้งค่า Flag จะไม่ปรากฏในโพรไฟล์เลย

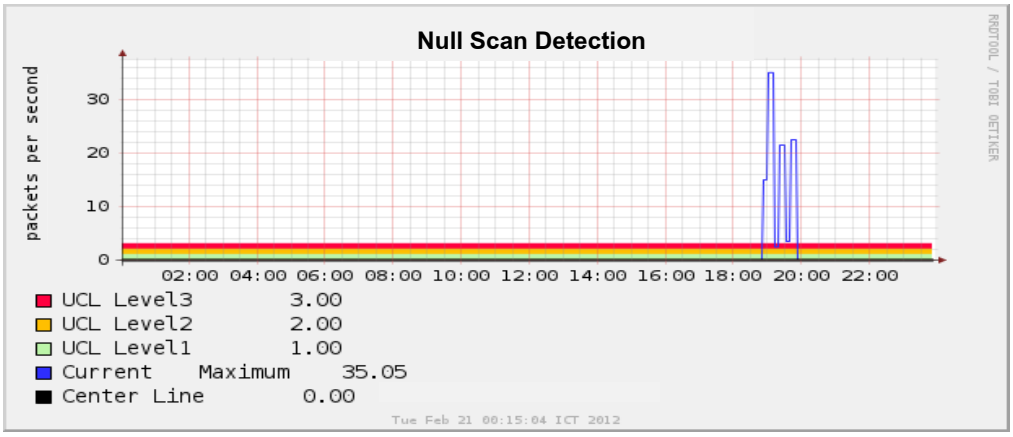


ภาพประกอบที่ 6-14 โพรไฟล์ของแพ็กเก็ต TCP ที่ไม่มีการตั้งค่า Flag

เมื่อผู้วิจัยได้ทำการทดสอบโดยการสแกนเครือข่ายแบบ Null Scan จำนวน 5 ครั้ง แต่เนื่องจากการสแกนเครือข่ายนั้นไม่ส่งผลกระทบในเรื่องของปริมาณข้อมูลบนเครือข่าย จึงไม่เห็นความเปลี่ยนแปลงของข้อมูลในภาพรวมอย่างชัดเจน ดังภาพประกอบที่ 6-15 เพราะในการสแกนเพื่อหาข้อมูลของเครือข่ายเป้าหมายนั้น ไม่จำเป็นต้องใช้แพ็กเก็ตในการสแกนปริมาณมากเหมือนกับการโจมตีที่ผ่านมา แต่หากมองที่โพรไฟล์ของ Null Scan จะพบว่ามีแพ็กเก็ตที่บ่งบอกว่ามีการสแกนแบบ Null Scan เกิดขึ้นในเครือข่าย นั่นคือแพ็กเก็ต TCP ที่ไม่มีการตั้งค่า Flag เกิดขึ้นดังแสดงในภาพประกอบที่ 6-16



ภาพประกอบที่ 6-15 ปริมาณข้อมูลเมื่อโจมตี Null Scan

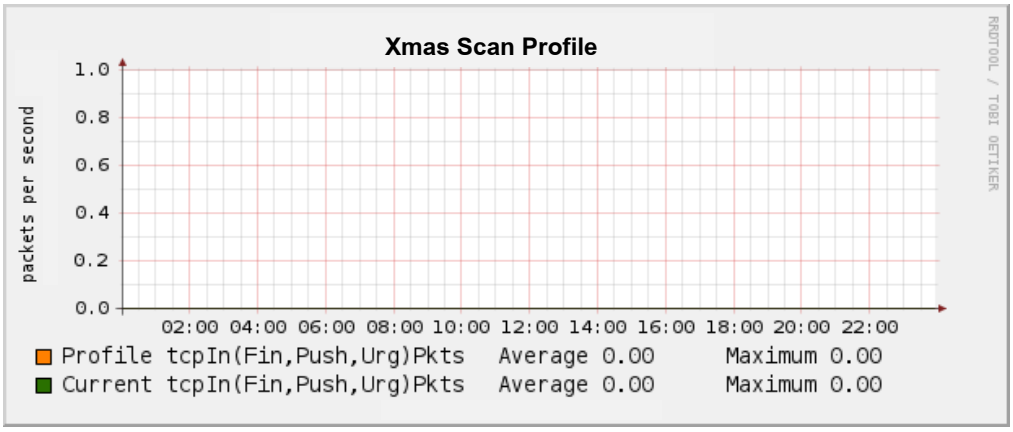


ภาพประกอบที่ 6-16 ผลการทดสอบการตรวจจับ Null Scan

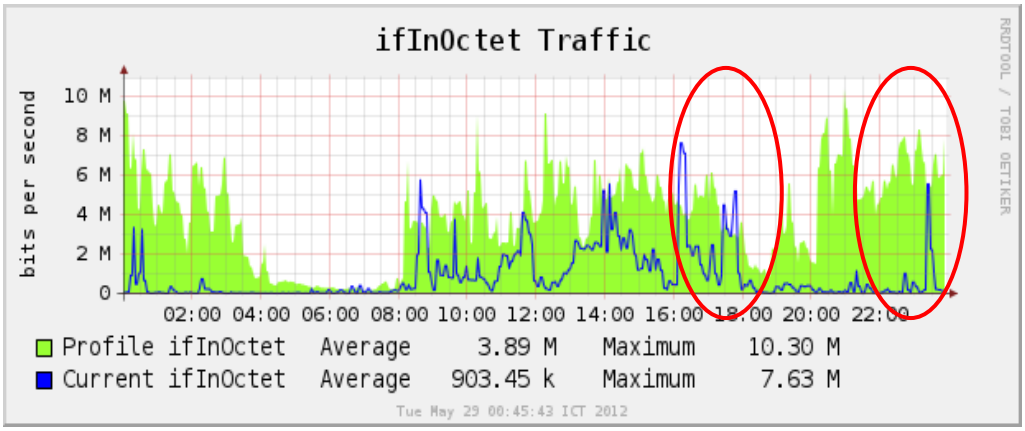
6.3.7 วิธีทดสอบการตรวจจับ Xmas Scan

เครื่องมือที่ใช้ในการทดสอบ Xmas Scan ผู้วิจัยได้เลือกใช้เครื่องมือ Nmap ในการทดสอบเช่นเดียวกับ Null Scan จากภาพประกอบที่ 6-17 จะแสดงโพรไฟล์ของแพ็กเก็ต TCP ที่ตั้งค่า Flag เป็น FIN, URG และ PSH พร้อมกันในเครือข่าย ซึ่งจะเห็นได้ว่าไม่ปรากฏแพ็กเก็ตประเภทดังกล่าวเลย

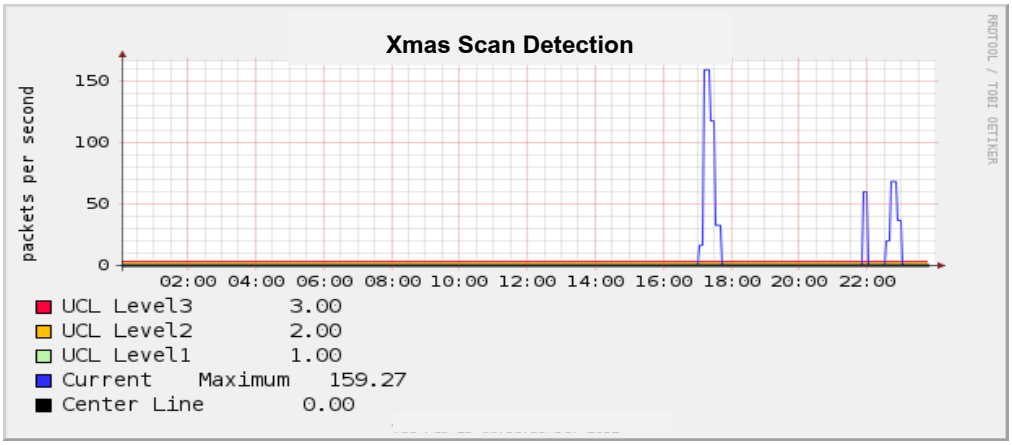
การสแกนรูปแบบนี้เป็นเทคนิคในการหาข้อมูลหรือบริการที่เครื่องเป้าหมายเปิดให้บริการอยู่เช่นเดียวกับ Null Scan ต่างกันเพียงเทคนิคหรือรูปแบบที่ใช้ในการตรวจสอบ ซึ่งการสแกนนี้ไม่ได้ส่งผลกระทบต่อเครือข่ายเช่นเดียวกันกับ Null Scan ดังภาพประกอบที่ 6-18 สำหรับการทดสอบครั้งนี้ผู้วิจัยได้ทำการสแกนเครือข่ายจำนวน 5 ครั้ง พบแพ็กเก็ตที่บ่งบอกว่ามีการสแกนแบบ Xmas Scan เกิดขึ้น ดังภาพประกอบที่ 6-19



ภาพประกอบที่ 6-17 โพรไฟล์ของแพ็กเก็ต TCP ที่ตั้งค่า Flag เป็น FIN, URG และ PSH



ภาพประกอบที่ 6-18 ปริมาณข้อมูลเมื่อโจมตี Xmas Scan



ภาพประกอบที่ 6-19 ผลการทดสอบการตรวจจับ Xmas Scan

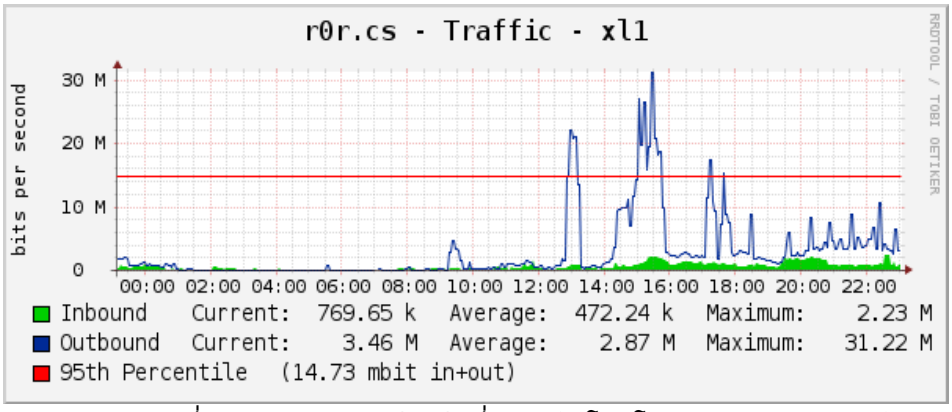
จากการทดสอบนี้ทำให้ทราบว่า โปรแกรมตรวจจับการบุกรุกที่พัฒนาขึ้นโดยใช้อ็อบเจกต์ของ MIB+ นั้น สามารถนำมาใช้เพื่อบ่งบอกถึงเหตุการณ์ที่เกิดขึ้นว่ามีความผิดปกติหรือไม่ ในส่วนการแสดงข้อมูลที่เกิดขึ้นทำให้เราสามารถเห็นถึงความผิดปกติของเหตุการณ์ได้อย่างชัดเจน เนื่องจากนำเสนอข้อมูลในรูปแบบของกราฟ

6.4 การทดสอบประสิทธิภาพ

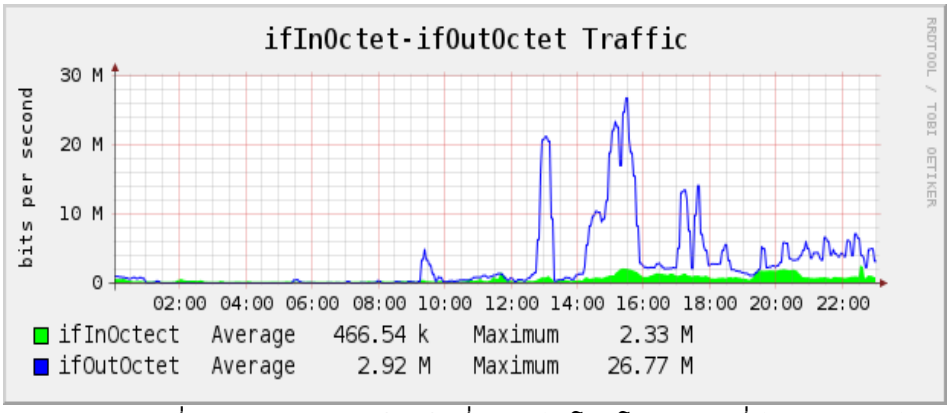
การทดสอบประสิทธิภาพของระบบแบ่งเป็นการทดสอบในส่วนของความถูกต้องในการสอบถามข้อมูลของระบบ และทรัพยากรที่ถูกใช้เมื่อมีการใช้งานระบบตรวจจับการบุกรุก

6.4.1 ความถูกต้องในการเก็บข้อมูล

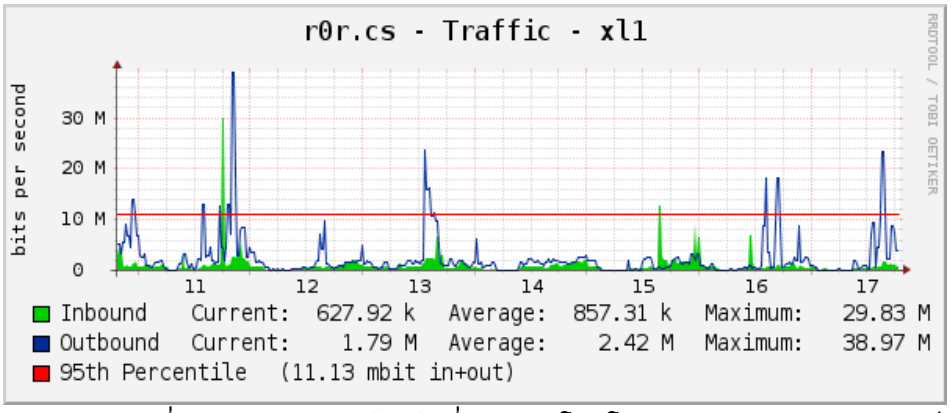
การทดสอบความถูกต้องในการเก็บข้อมูลของระบบนั้น ผู้วิจัยได้เปรียบเทียบความถูกต้องกับโปรแกรม cacti ซึ่งเป็นโปรแกรมที่ใช้ในการจัดการบริหารเครือข่ายที่ได้รับความนิยมใช้งานในปัจจุบัน ในการทดสอบนั้นผู้วิจัยได้ใช้ MIB อ็อบเจกต์ที่เป็นมาตรฐานสำหรับเก็บข้อมูลอ็อบเจกต์ ซึ่งอ็อบเจกต์ที่ใช้คือ ifInOctets และ ifOutOctets ซึ่งใช้สำหรับบอกภาพรวมของการใช้งานเครือข่ายว่ามีปริมาณของแพ็กเก็ตเข้าและออกเป็นอย่างไร ผู้วิจัยได้ทำการทดสอบเก็บข้อมูลรายวันและรายสัปดาห์ จากผลการทดสอบจะเห็นได้ว่าค่าเฉลี่ยในการตรวจจับแพ็กเก็ตของทั้งสองโปรแกรมมีค่าที่ใกล้เคียงกันเมื่อคิดเป็นเปอร์เซ็นจะได้ค่าความถูกต้องที่ 98% ดังแสดงในภาพประกอบที่ 6-20 ถึง 6-23 ตามลำดับ



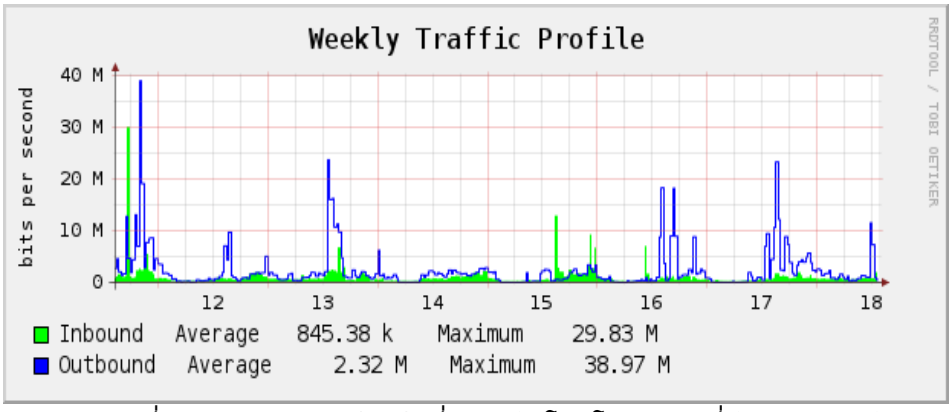
ภาพประกอบที่ 6-20 ปริมาณแพ็กเก็ตที่ตรวจจับโดยโปรแกรม cacti (รายวัน)



ภาพประกอบที่ 6-21 ปริมาณแพ็กเก็ตที่ตรวจจับโดยโปรแกรมที่พัฒนา (รายวัน)



ภาพประกอบที่ 6-22 ปริมาณแพ็กเก็ตที่ตรวจจับโดยโปรแกรม cacti (รายสัปดาห์)



ภาพประกอบที่ 6-23 ปริมาณแพ็กเก็ตที่ตรวจจับโดยโปรแกรมที่พัฒนา (รายสัปดาห์)

6.4.2 ทดสอบความถูกต้องในการตรวจจับ

สำหรับการทดสอบความถูกต้องในการตรวจจับ ผู้วิจัยได้แบ่งการทดสอบออกเป็น 3 ลักษณะคือ การทดสอบอัตราการตรวจจับการโจมตี (Attack Detection Rate: ADR) เป็นการทดสอบความสามารถในการตรวจจับความผิดปกติที่เกิดขึ้น การทดสอบต่อมาคือ ทดสอบอัตราความผิดพลาดในการตรวจจับเชิงบวก (False Positive Rate: FPR) คือการที่มีเหตุการณ์ปกติ แต่ระบบกลับระบุว่ามีการโจมตีหรือการบุกรุกเกิดขึ้น และสุดท้ายคือ การทดสอบอัตราความผิดพลาดในการตรวจจับเชิงลบ (False Negative Rate: FNR) นั่นคือ การที่มีเหตุการณ์ผิดปกติเกิดขึ้นและระบบไม่สามารถตรวจจับหรือระบุความผิดปกติที่เกิดขึ้นได้ สำหรับการทดสอบดังกล่าว มีวิธีการคำนวณดังนี้ (Cui-Mei, 2009)

$$ADR = \frac{\sum_{i=1}^{n} T_i}{\sum_{i=1}^{n} I_i} \qquad (1)$$

$$FPR = \frac{\sum_{i=1}^{n} P_i}{\sum_{i=1}^{n} N_i} \qquad (2)$$

$$FNR = \frac{\sum_{i=1}^{n} F_i}{\sum_{i=1}^{n} I_i} \qquad (3)$$

I คือ จำนวนข้อมูลที่เป็นการโจมตีทั้งหมด

N คือ จำนวนข้อมูลทั้งหมด

T คือ จำนวนข้อมูลที่สามารถตรวจจับได้ว่าเป็นการโจมตี

F คือจำนวนข้อมูลที่เป็นการโจมตีแต่ไม่สามารถตรวจจับได้

P คือ จำนวนข้อมูลที่เป็นเหตุการณ์ปกติแต่ระบบระบุว่าเป็นการโจมตี

ในการทดสอบครั้งนี้ผู้วิจัยได้ทำการทดลองโจมตีการบุกรุกทั้ง 5 รูปแบบ คือ SYN Flood Attack, Land Attack, DNS Flood Attack, Null Scan และ Xmas Scan โดยทำการโจมตีรูปแบบละ 30 ครั้ง ผลที่ได้แสดงดังตารางที่ 6-2

ตารางที่ 6-2 การทดสอบประสิทธิภาพในการตรวจจับของระบบที่นำเสนอ

Attack Type	ADR	FPR	FNR
SYN Flood Attack	100%	3.47%	0%
Land Attack	100%	0%	0%
DNS Flood Attack	100%	2.77%	0%
Null Scan	100%	0%	0%
Xmax Scan	100%	0%	0%

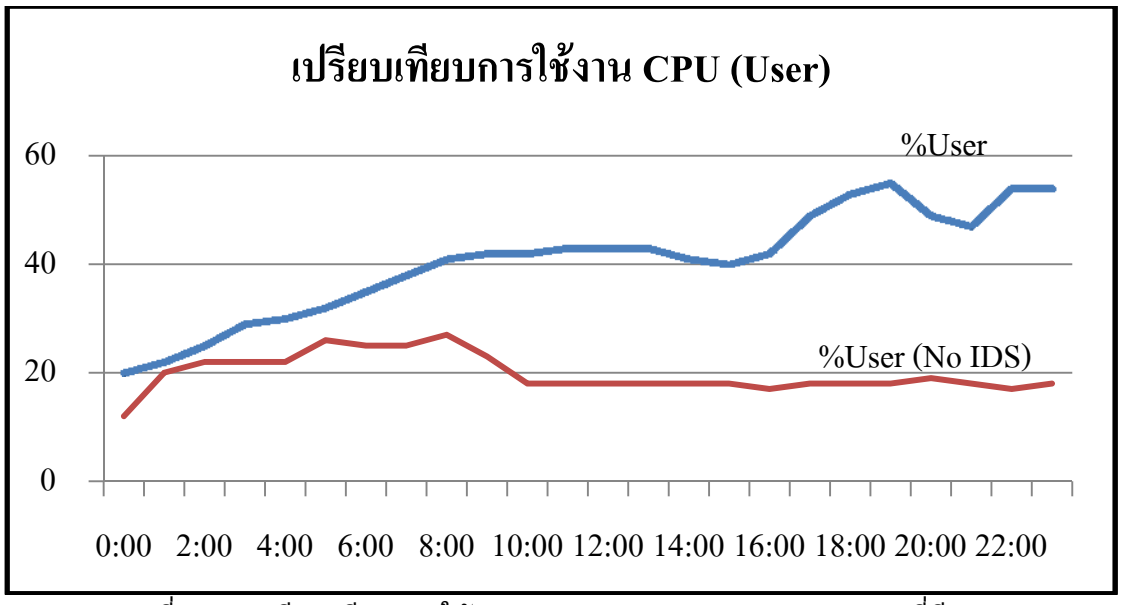
จากผลการทดสอบประสิทธิภาพของการตรวจจับจะเห็นได้ว่าระบบที่ออกแบบนั้นมีความถูกต้องในการตรวจจับการโจมตีสูง เพราะมีอัตราในการตรวจจับการโจมตี 100% คือสามารถตรวจจับความผิดปกติได้ทุกครั้งเมื่อมีความผิดปกติเกิดขึ้น มีค่า False Positive Rate ที่ต่ำ ในขณะที่ไม่มีค่า False Negative Rate

6.4.3 การใช้หน่วยประมวลผลในระบบ

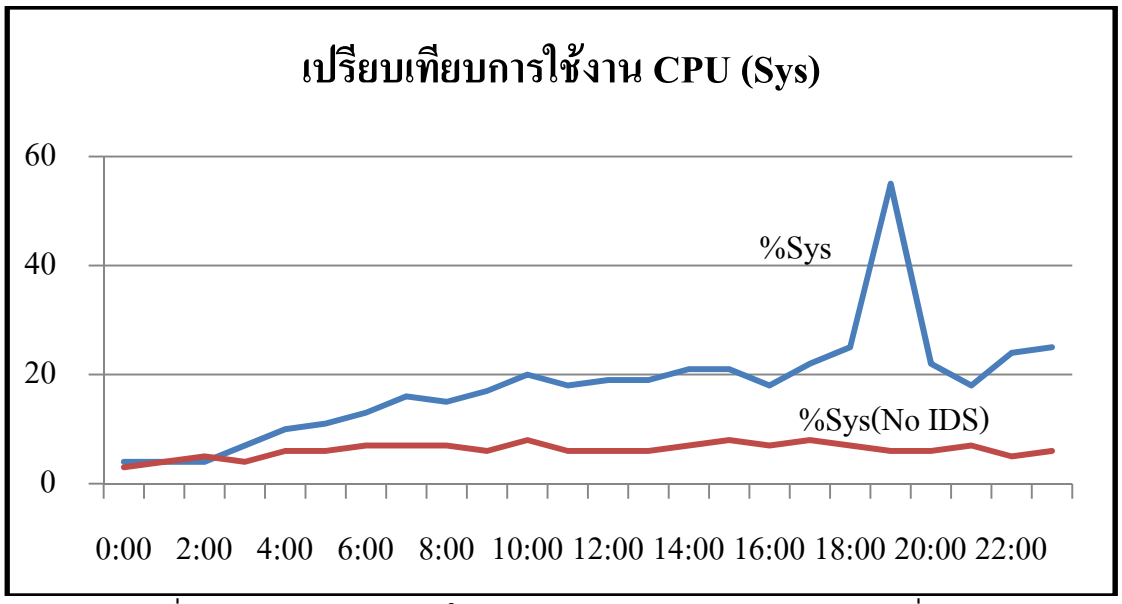
ผู้วิจัยได้เปรียบเทียบการทำงานของหน่วยประมวลผล ระหว่างที่มีระบบตรวจจับการบุกรุกติดตั้งและทำงานอยู่ กับการที่ไม่มีระบบตรวจจับการบุกรุกทำงานอยู่ โดยใช้คำสั่ง “bsdsar” ซึ่งเป็นคำสั่งสำหรับดูสถิติการทำงาน หรือการใช้งานของหน่วยความจำในระบบ ผลที่ได้จากคำสั่ง bsdsar แสดงดังตารางที่ 6-3 และ 6-4 ภาพประกอบที่ 6-24 ถึง 6-28 แสดงการเปรียบเทียบการใช้งานหน่วยประมวลผลในขณะมีระบบตรวจจับการบุกรุกและไม่มีระบบตรวจจับการบุกรุกทำงานอยู่

ตารางที่ 6-3 เปรียบเทียบการใช้งานหน่วยประมวลผลในขณะที่มีระบบตรวจจับการบุกรุกและไม่มีระบบตรวจจับการบุกรุก

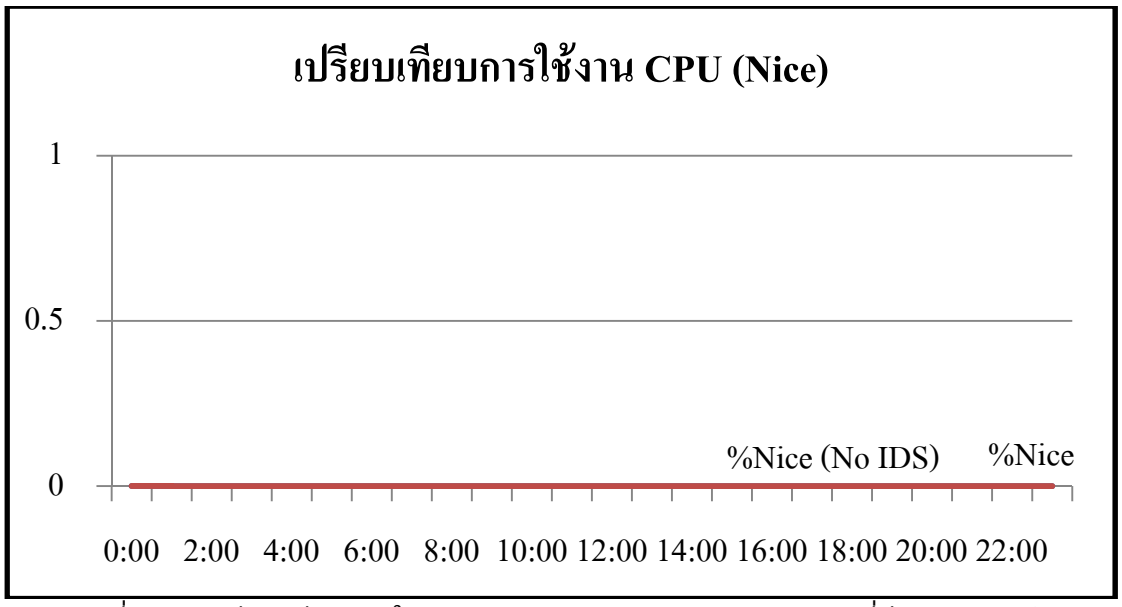
Time	%User		%Sys		%Nice		%Interrupt		%Idle	
	No IDS	IDS	No IDS	IDS	No IDS	IDS	No IDS	IDS	No IDS	IDS
00:00	12	20	3	4	0	0	11	11	73	64
01:00	20	22	4	4	0	0	11	11	64	63
02:00	22	25	5	4	0	0	12	11	62	59
03:00	22	29	4	7	0	0	10	11	63	56
04:00	22	30	6	10	0	0	10	10	62	50
05:00	26	32	6	11	0	0	11	12	57	46
06:00	25	35	7	13	0	0	12	12	56	41
07:00	25	38	7	16	0	0	11	11	57	35
08:00	27	41	7	15	0	0	11	11	55	32
09:00	23	42	6	17	0	0	11	11	59	32
10:00	18	42	8	20	0	0	11	11	66	28
11:00	18	43	6	18	0	0	11	11	64	28
12:00	18	43	6	19	0	0	10	11	64	29
13:00	18	43	6	19	0	0	11	11	66	27
14:00	18	41	7	21	0	0	11	11	64	26
15:00	18	40	8	21	0	0	11	11	64	28
16:00	17	42	7	18	0	0	11	11	64	27
17:00	18	49	8	22	0	0	11	11	63	16
18:00	18	53	7	25	0	0	11	11	65	10
19:00	18	55	6	55	0	0	12	11	64	12
20:00	19	49	6	22	0	0	11	11	64	18
21:00	18	47	7	18	0	0	11	11	65	24
22:00	17	54	5	24	0	0	11	11	66	11
23:00	18	54	6	25	0	0	11	12	65	10



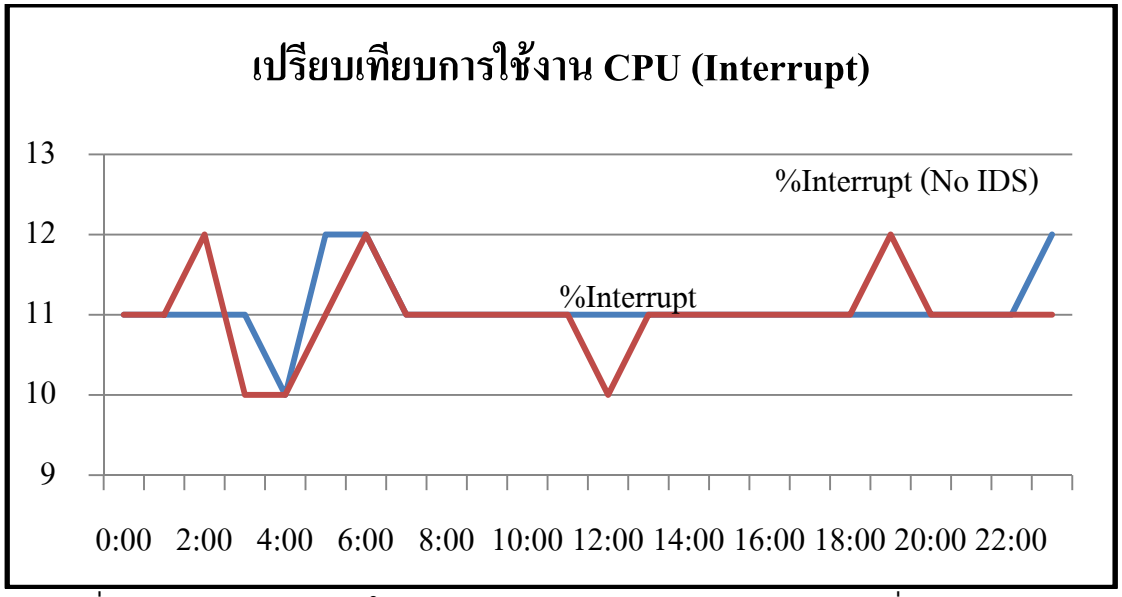
ภาพที่ 6-24 เปรียบเทียบการใช้งานหน่วยประมวลผล (User) ระหว่างที่มีระบบตรวจจับการบุกรุกและไม่มีระบบตรวจจับการบุกรุกทำงาน



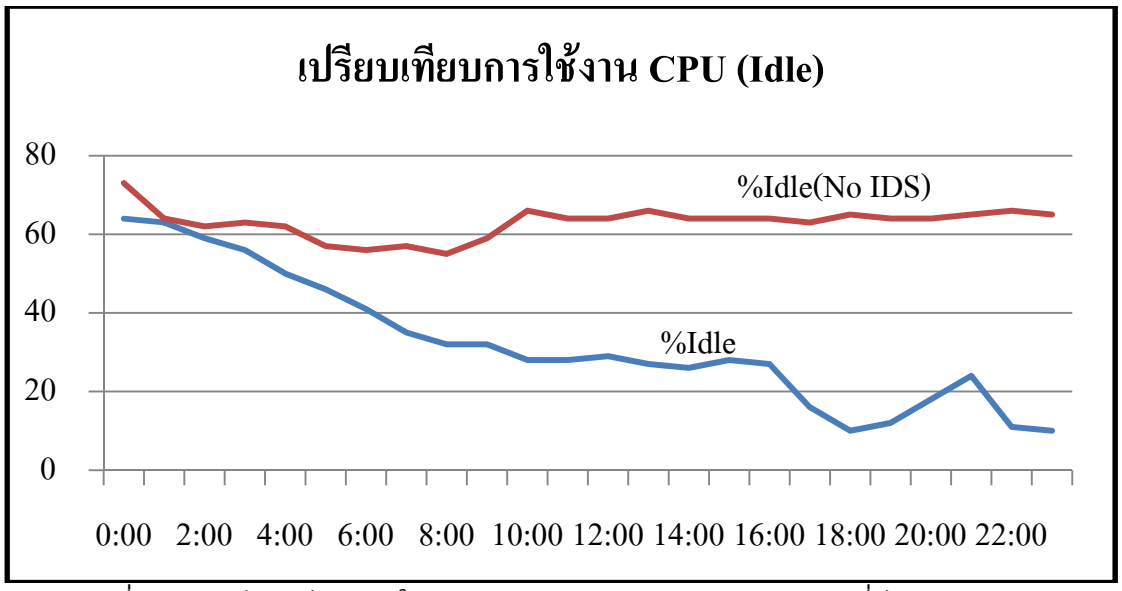
ภาพที่ 6-25 เปรียบเทียบการใช้งานหน่วยประมวลผล (Sys) ระหว่างที่มีระบบตรวจจับการบุกรุกและไม่มีระบบตรวจจับการบุกรุกทำงาน



ภาพที่ 6-26 เปรียบเทียบการใช้งานหน่วยประมวลผล (Nice) ระหว่างที่มีระบบตรวจจับการบุกรุกและไม่มีระบบตรวจจับการบุกรุกทำงาน



ภาพที่ 6-27 เปรียบเทียบการใช้งานหน่วยประมวลผล (Interrupt) ระหว่างที่มีระบบตรวจจับการบุกรุกและไม่มีระบบตรวจจับการบุกรุกทำงาน



ภาพที่ 6-28 เปรียบเทียบการใช้งานหน่วยประมวลผล (Idle) ระหว่างที่มีระบบตรวจจับการบุกรุกและไม่มีระบบตรวจจับการบุกรุกทำงาน

จากภาพที่ 6-24 – 6-28 แสดงการเปรียบเทียบการใช้งานหน่วยประมวลผลของระบบ เห็นได้ว่าในส่วนของ User (จำนวนเปอร์เซ็นของ CPU ที่ถูกใช้งานในโหมด user), Sys (จำนวนเปอร์เซ็นของ CPU ที่ถูกใช้โดยโหมด System), Nice (เวลาที่ CPU ใช้ในโพรเซส Nice โดยที่โพรเซส Nice คือโพรเซสที่ถูกปรับเปลี่ยน Priority ในตารางงาน หากค่า Nice มีค่าน้อยจะมี Priority สูง), Interrupt (จำนวนเปอร์เซ็นของ CPU ที่มีการเกิดฮาร์ดแวร์ Interrupt) และ Idle (จำนวนเปอร์เซ็นของ CPU ที่ว่าง) จะพบว่าระบบตรวจจับการบุกรุกที่พัฒนาขึ้นนั้นมีการใช้งานหน่วยประมวลผลของระบบเพิ่มขึ้น โดยค่าของ User เพิ่มขึ้นจากเดิม 2 เท่า ค่าของ Sys เพิ่มขั้นจากเดิม 3 เท่า ส่วนค่าของ Nice และ Interrupt ไม่เพิ่มขึ้น และค่าของ Idle นั้นลดลง 2 เท่า เมื่อเทียบกับการไม่ติดตั้งระบบตรวจจับการบุกรุก

ถึงแม้ระบบตรวจจับการบุกรุกที่พัฒนาขึ้นนี้ จะใช้งานหน่วยประมวลผลเพิ่มขึ้น แต่ก็มีความถูกต้องแม่นยำในการจัดเก็บข้อมูล ความถูกต้องในการตรวจจับและระบุความผิดปกติที่เกิดขึ้นได้อย่างถูกต้อง

6.5 สรุป

ในบทนี้ได้กล่าวถึงการทดสอบและวัดประสิทธิภาพการทำงานของระบบตรวจจับการบุกรุกที่พัฒนาขึ้น โดยระบบที่พัฒนาขึ้นนั้นมีลักษณะการทำงานเป็นแบบ Network-based IDS คือสามารถตรวจจับความผิดปกติที่เกิดขึ้นในเครือข่ายเท่านั้น โดยลักษณะความผิดปกติที่สามารถตรวจจับได้มี 5 ชนิด คือ SYN Flood Attack, Land Attack, DNS Flood Attack, Null Scan และ Xmas Scan ผลการทดสอบที่ได้แสดงให้เห็นในรูปแบบของกราฟ และสำหรับการทดสอบประสิทธิภาพ ผู้วิจัยได้แบ่งเป็นการทดสอบความถูกต้องในการจัดเก็บข้อมูล ความถูกต้องในการตรวจจับ และการใช้ทรัพยากรของระบบ ผลการทดสอบที่ได้คือ ระบบตรวจจับการบุกรุกที่พัฒนาขึ้นนั้นมีความถูกต้องในการจัดเก็บข้อมูลเทียบเท่ากับโปรแกรม cacti ซึ่งเป็นโปรแกรมจัดการเครือข่าย และมีความถูกต้องสูงในการตรวจจับความผิดปกติอีกด้วย ถึงแม้จะมีการใช้งานหน่วยประมวลผลในระบบเพิ่มขึ้น แต่ก็ไม่ได้ส่งผลกระทบต่อการทำงานของเครื่องที่ติดตั้งโปรแกรมตรวจจับการบุกรุกที่พัฒนาขึ้นเลย

บทที่ 7

สรุป ปัญหาและข้อเสนอแนะ

7.1 บทนำ

สำหรับบทนี้จะเป็นการกล่าวสรุปงานวิจัยและผลที่ได้จากการวิจัยครั้งนี้ รวมถึงปัญหาและอุปสรรคที่เกิดขึ้นในระหว่างการทำวิจัย สุดท้ายจะเป็นการกล่าวถึงข้อเสนอแนะให้แก่ผู้สนใจที่จะนำงานวิจัยนี้ไปพัฒนาต่อไป

7.2 สรุปผลการวิจัย

งานวิจัยนี้เป็นการนำเสนอชนิดข้อมูลที่จำเป็นสำหรับใช้เป็นพารามิเตอร์ในการตรวจจับความผิดปกติบนระบบคอมพิวเตอร์และเครือข่าย โดยชนิดข้อมูลที่นำเสนอนั้นได้มาจากการศึกษา และวิเคราะห์รูปแบบการโจมตีและการตรวจจับการบุกรุกที่เกิดขึ้นในอดีต ซึ่งสามารถนำข้อมูลที่ได้จากการวิเคราะห์ครั้งนี้สร้างเป็นโพรไฟล์สำหรับใช้ในการตรวจจับความผิดปกติบนระบบคอมพิวเตอร์และเครือข่ายได้

เพื่อเป็นการเพิ่มประสิทธิภาพของ MIB อ็อบเจกต์ ซึ่งส่วนใหญ่แล้วใช้สำหรับจัดการบริหารเครือข่าย ให้สามารถใช้จัดการด้านความปลอดภัยได้อย่างมีประสิทธิภาพยิ่งขึ้น ผู้วิจัยจึงนำเสนอข้อมูลที่ได้จากการศึกษาครั้งนี้ให้อยู่ในรูปแบบของ MIB อ็อบเจกต์ ในงานวิจัยนี้เรียกว่า MIB+ อ็อบเจอต์ที่นำเสนอนั้นมีด้วยกัน 7 กลุ่มคือ ipDataPkts, icmpDataPkts, tcpDataPkts, udpDataPkts, otherProtocol, ports และ others สำหรับเหตุผลในการเลือกนำเสนอชนิดข้อมูลหรือพารามิเตอร์ที่ใช้ตรวจจับความผิดปกติในรูปแบบของ MIB อ็อบเจกต์นั้น เพราะ MIB อ็อบเจกต์มีโครงสร้างข้อมูลแบบต้นไม้ ซึ่งเป็นโครงสร้างข้อมูลที่ไม่ซับซ้อน สามารถเพิ่มเติมหรือแก้ไขข้อมูลอ็อบเจกต์ได้ง่าย อีกทั้งยังมีการใช้งานกันอย่างแพร่หลายอีกด้วย

เพื่อเป็นการนำเสนอว่าสามารถใช้อ็อบเจกต์ใน MIB+ ตรวจจับความผิดปกติที่เกิดขึ้นบนระบบคอมพิวเตอร์และเครือข่ายได้จริง ผู้วิจัยจึงได้พัฒนาระบบตรวจจับการบุกรุกโดยนำอ็อบเจกต์ใน MIB+ มาใช้สร้างเป็นโพรไฟล์เพื่อนำไปใช้ในการตรวจจับการบุกรุกหรือความผิดปกติในรูปแบบของ Anomaly Detection ต่อไป โดยในการทดลองครั้งนี้ผู้วิจัยได้ใช้เครือข่ายของภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์เป็น

กรณีศึกษา ซึ่งระบบตรวจจับการบุกรุกที่พัฒนาขึ้นนั้นเป็นแบบ Network-based IDS มีข้อจำกัดคือสามารถตรวจจับความผิดปกติที่เกิดขึ้นบนเครือข่ายเท่านั้น เบื้องต้นผู้วิจัยได้ใช้อ็อบเจกต์ของ Standard MIB นั่นคืออ็อบเจกต์ ifInOctets เพื่อดูภาพรวมของเครือข่ายว่ามีความผิดปกติหรือไม่ หากพบว่ามีความผิดปกติเกิดขึ้น ระบบจะตรวจสอบว่าความผิดปกติที่เกิดขึ้นนั้นเป็นความผิดปกติรูปแบบใด โดยระบบที่พัฒนาขึ้นนี้สามารถตรวจจับความผิดปกติได้ 5 รูปแบบคือ SYN Flood Attack, Land Attack, DNS Flood Attack, Null Scan และ Xmas Scan โดยอ็อบเจกต์ที่ใช้คือ อ็อบเจกต์ tcpInFinPkts และ tcpInSynPkts สำหรับตรวจจับความผิดปกติรูปแบบ SYN Flood Attack อ็อบเจกต์ ipInAddSrc สำหรับตรวจจับความผิดปกติรูปแบบ Land Attack อ็อบเจกต์ udpInPortNumber53 สำหรับตรวจจับความผิดปกติรูปแบบ DNS Flood Attack อ็อบเจกต์ tcpInNoFlagSetPkts สำหรับตรวจจับความผิดปกติรูปแบบ Null Scan อ็อบเจกต์ tcpInUrgPkts, tcpInPushPkts และ tcpInFinPkts สำหรับตรวจจับความผิดปกติรูปแบบ Xmas Scan

ผลการทำงานของโปรแกรมแสดงในรูปแบบของกราฟเพื่อให้เห็นแนวโน้มของการใช้งานเครือข่ายและความผิดปกติได้อย่างชัดเจน นอกจากนี้เมื่อระบบพบว่ามีความผิดปกติเกิดขึ้นจะทำการบันทึกความผิดปกตินั้นลงใน Log File เพื่อให้ผู้ดูแลระบบสามารถเรียกดูข้อมูลความผิดปกติย้อนหลังได้

ในการทดสอบประสิทธิภาพของระบบตรวจจับการบุกรุกที่พัฒนาขึ้นนั้นผู้วิจัยได้แบ่งการทดสอบเป็น 3 เรื่องคือ การทดสอบความถูกต้องในการจัดเก็บข้อมูล ความถูกต้องในการตรวจจับและการใช้งานทรัพยากรระบบระหว่างมีและไม่มีระบบตรวจจับการบุกรุกทำงานอยู่ ซึ่งในการทดสอบความถูกต้องนั้นได้ทำการเปรียบเทียบการเก็บข้อมูลกับโปรแกรม cacti ซึ่งเป็นโปรแกรมที่ใช้ในการจัดการบริหารเครือข่ายที่นิยมและใช้งานกันอย่างแพร่หลายทั้งในอดีตและปัจจุบัน ซึ่งให้ผลในการจัดเก็บข้อมูลมีความถูกต้องที่ 98% สำหรับการทดสอบความถูกต้องในการตรวจจับได้แบ่งเป็นการทดสอบอัตราการตรวจจับการโจมตี (Attack Detection Rate) อัตราความผิดพลาดในการตรวจจับเชิงบวก (False Positive Rate) และอัตราความผิดพลาดในการตรวจจับเชิงลบ (False Negative Rate) ซึ่งมีประสิทธิภาพในการตรวจจับและความถูกต้องสูง และประสิทธิภาพในเรื่องของการใช้งานทรัพยากรเครือข่ายนั้นระบบตรวจจับการบุกรุกที่พัฒนา ขึ้นมีการใช้งานหน่วยประมวลผลค่อนข้างสูงเมื่อเทียบกับระบบที่ไม่ได้ทำการติดตั้งโปรแกรมดังกล่าว

7.3 ปัญหาและอุปสรรคในการวิจัย

7.3.1 เนื่องจากการศึกษาหาพารามิเตอร์ที่จะนำมาใช้ในการสร้างโพรไฟล์สำหรับตรวจจับความผิดปกตินั้น ต้องศึกษาจากการโจมตีที่เกิดขึ้นแล้วในอดีตและรายงานวิจัยต่างๆ ที่ผ่านมา ประกอบกับการโจมตีบนระบบคอมพิวเตอร์และเครือข่ายมีจำนวนมากและหลากหลาย ทำให้ใช้เวลาในการศึกษารูปแบบการโจมตีและการตรวจจับในแต่ละรูปแบบเป็นเวลานาน

7.3.2 เนื่องจากเทคนิคที่ใช้ในการวิเคราะห์การตรวจจับการบุกรุกนั้นมีหลายวิธี ซึ่งแต่ละวิธีก็มีข้อจำกัดที่แตกต่างกันออกไป ทำให้ใช้เวลาในการศึกษารายละเอียดแต่ละเทคนิคเป็นเวลานาน

7.4 ข้อเสนอแนะ

7.4.1 ในส่วนของการสร้างอ็อบเจกต์ใหม่ ผู้ที่สนใจสามารถทำการเพิ่มเติมอ็อบเจกต์นอกเหนือจากนี้ได้ เมื่อพบว่ามีการโจมตีรูปแบบใหม่เกิดขึ้นและ อ็อบเจกต์ที่มีอยู่ไม่เพียงพอหรือสามารถใช้ในการตรวจจับการโจมตีรูปแบบที่ต้องการตรวจจับได้

7.4.2 ในการเพิ่มสมาชิกของกลุ่มใดๆ สิ่งที่ต้องคำนึงถึงคือ จำนวนสมาชิกใหม่ที่จะเพิ่มขึ้น ถ้าหากจำนวนสมาชิกมีกลุ่มย่อยๆลงไปอีก อาจจะเพิ่มอ็อบเจกต์นี้ให้อยู่ในระดับเดียวกับอ็อบเจกต์ที่ได้นำเสนอหรือเป็น Sub-Tree ของ intrusionData ได้ เพราะเนื่องจาก MIB มีโครงสร้างข้อมูลแบบต้นไม้ ถ้าหากโหนดลูกที่ต้องการเรียกใช้งานอยู่ในระดับลึกๆ แล้ว จะทำให้ประสิทธิภาพในการค้นหาช้าลงได้

7.4.3 สำหรับระบบตรวจจับการบุกรุกที่พัฒนาขึ้นนั้น เป็นเพียงต้นแบบเพื่อแสดงให้เห็นถึงการนำ MIB+ มาใช้ในการตรวจจับความผิดปกติ สามารถเพิ่มรูปแบบการตรวจจับการโจมตีอื่นๆ เข้าไปได้ หากมีผู้สนใจนำไปพัฒนาต่อ สามารถเพิ่มเติมในส่วนของรูปแบบการแจ้งเตือน หรือรายงานสรุปเพื่อให้สะดวกต่อการจัดการดูแลเครือข่าย

บรรณานุกรม

คุณชิต สุขพัฒนศรีกุล. 2549. ต้นแบบระบบตรวจจับการบุกรุกแบบผสมโดยใช้วิธีการทำเหมืองข้อมูลส่วนคำสั่งที่เรียกใช้บริการของระบบปฏิบัติการ, วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต สาขาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยสงขลานครินทร์.

ชัยยุทธ จันแดง. 2549. การเพิ่มสมรรถนะระบบปฏิบัติการด้วยฟังก์ชันการตรวจจับและป้องกันการบุกรุกำบนระบบปฏิบัติการเน็ทบีเอสดี, วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต สาขาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยสงขลานครินทร์.

พัฒนาวดี ศิวติณทุโก. 2548. การจัดทำโปรแกรมตรวจจับการบุกรุกบนระบบปฏิบัติการยูนิกซ์, วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยสงขลานครินทร์.

ศุภโชค สุขเกษม. 2548. การวิเคราะห์ข้อมูลกิจกรรมของระบบเพื่อตรวจจับการบุกรุก, วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต สาขาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยสงขลานครินทร์.

สุธิพันธ์ สุวรรณเวลา. 2551. ภาษาสอบถามการจัดการเครือข่ายสำหรับโพรโตคอลจัดการเครือข่าย,วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต สาขาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยสงขลานครินทร์.

Amoroso, E.G. 1999. An Introduction to Internet Surveillance, Correlation, Trace Back, Traps, and Response. First Edition:United States of America.

Anderson, J.P. 1980. Computer Security Threat Monitoring and Surveillance. James P. Anderson Co., Fort Washington, Pennsylvania.

Anderson, J. 2001. Analysis of Fragmentation Attacks. http://www.ouah.org/fragma.html. (accessed 11/05/2010).

Armstrong, D. 1996. Password Sniffing.http://cng.seas.rochester.edu/CNG/docs/Security / node8.html. (accessed 20/05/2011).

Bace, R. and Mell, P. 2000. Intrusion Detection System. In: Network Security, Vol. 2000, pp. 19.

Case, J., Fedor, M. Schoffstall, M. and David, J. 1990. A Simple Network Management Protocol (SNMP). RFC1157.

Cert. 1997. CERT® Advisory CA-1992-02 Michelangelo PC Virus Warning. http://www.cert.org/advisories/CA-1992-02.html. (accessed 11/10/2010).

CISCO. 2011. Preventing Network Attack.http://www.cisco.com/en/US/docs/security/asa /asa 72/asdm52/user/guide /protect.html. (accessed 12/09/2011).

Communication Security Inc. 2011. Telephone Line Attack. http://www.bugsweep.com/ telephone.html. (accessed 15/08/2011).

Dening, D.E. 1987. An Intrusion-Detection Model. IEEE Transaction on Software Engineerin.USA, February, pp. 222-232.

Durgin, N.A. and Zhang, P. 2005. Profile-Based Adaptive Anomaly Detection for Network Security. Technical Report, Sandia National Laboratories. November 01.

Guyon, I. and Elisseeff, A. 2003. An Introduction to Variable and Feature Selection. pp. 1158-1182.

Ghali, M.E. and Masri, W. 2009. Intrusion detection using signatures extracted from execution profiles. ICSE'09 Workshop. Vancouver, Canada, May 19, pp. 17-24.

Gaspary, L.P., Sanchez, R.N., Antunes, D.W. and Meneghetti E. 2005. A SNMP-Based Platform for Distributed Stateful Intrusion Detection in Enterprise Networks. Journal on selected areas in communications, Vol. 23, No. 10, October: 1973-1982.

Hansman, S. 2003. A Taxonomy of Network and Computer Attack Methodologies. Department of Computer Science and Software Engineering University of Canterbury, Christchurch, New Zealand.

Hansman, S. and Hunt, R. 2005. A Taxonomy of Network and Computer Attacks. Comp. & Sec., Vol. 24, No. 1, pp. 31–43.

Howard, J. D. and Longstaff, T. A. 1998. A Common Language for Computer Security Incidents.Sandia tech. rep. SAND98-8667.

Hsieh, C., Huang, Y. and Chen, R. 2011. A Light-weight Ranger Intrusion Detection System on Wireless Sensor Networks. Proceeding of 5th International Conference on Genetic and Evolutionary Computing.Taiwan, August 29 - September1. pp. 49-52.

Hussain, A., Heidemann, J. and Papadopoulos, C. 2003. Denial-of-Service: A Framework for Classifying Denial of Service Attacks. Proceeding of SIGCOMM'03 conference on Applications, technologies, architectures and

protocols for computer communication. Germany, August 25–29. 2003. pp. 99-110.

Igure, V.M. and Willams, R.D. 2008. Taxonomies of Attacks and Vulnerabilities in Computer Systems. The Electronic Magazine of Original Peer-Reviewed Survey Articles. pp. 6-19.

Internet World Status. 2010. World Internet Penetration Rates. http://www.internetworldstats.com/stats.htm. (accessed 20 /09/2011).

KasperskyLab. 2010. Top twenty malicious programs on the internet in 2009. http://www.securelist.com/en/analysis/204792101/Kaspersky_Se curi ty _B ulletin_2009_Statistics_2009. (accessed 11/10/2010).

Keshariya, A. and Foukia, N. 2010. DDoS Defense Mechanisms: A New Taxonomy. In: Lecture Notes in Computer Science, Vol. 5939, pp. 222-236.

Kim, Y.J., Kyunghee, J. and Suh K. 2006. Baseline Profile Stability for Network Anomaly Detection. Proceedings of the Third International Conference on Information IEEE. Las Vegas, April 10-12, pp. 720-725.

Krawetz, N. 2007. Introduction to Network Security. Charles River Media, an imprint of Thomson Learning, Inc. pp. 199-385.

Lim, S.Y. and Jones, A. 2008. Network Anomaly Detection System:The State of Art of Network Behaviour Analysis. Proceeding of the conference on Convergence and Hybrid Information Technology .Daejeon, August 28-30, pp. 459-465.

Lindquist, U. and Jonsson, E. 1997. How to Systematically Classify Computer Security Intrusions. Proc. IEEE Symp. Sec. and Privacy, pp. 154–63.

Lough, D.L. 2001. A Taxonomy of Computer Attacks with Applications to Wireless Networks . Blacksburg, Virginia.

MacGregor, J.F. and Kourti, T. 2000. Statistical Process Control of Multivariate Processes. Chemical Engineering Department, McMaster Advanced Control Consortium, McMaster University, Hamilton, Ontario L8S 4L7, Canada. pp. 403-414.

Marin, J., Ragsdale, D. and Surdu, J. 2001. A Hybrid Approach to the Profile Creation and Intrusion Detection. DARPA Information Survivability Conference & Exposition II, United State of America, pp. 69-76.

McCloghrie, K. and Rose, M. 1991. Management Information Based internet:MIB-II.RFC1213.

McCloghrie, K. and Rose, M. 1988. Structure and Identification of Management Information for TCP/IP Internets. RFC1065.

Murata, T. 1989. Petri Nets: Properties, Analysis and Applications. Proceedings of the IEEE, Vol. 77, No. 4, pp. 541-580.

Newsome J. et al. 2004. The Sybil Attack in Sensor Networks:Analysis & Defenses. Processing in Sensor Networks, Berkeley, pp. 259–68.

Nuansri, N. 1999. A Process State-Transition Analysis and its Application to Intrusion Detection. Proceeding of 15th Annual Computer Security Applications Conference. pp. 378-388.

Oakland, J.S. 2008. Statistical Process Control. John Wiley and Sons Inc., 605 Third Avenue, New York, NY 10150. pp. 237.

Paulauskas, N. and Grasva, E. 2006. Computer System Attack Classification. Proceeding of conference on Electronics And Electrical Engineering. pp. 84-87.

Perry, T.S. and Wallich, P. 1984. Can Computer Crime Be Stopped?. IEEE Spectrum, Vol. 21, No. 5, pp. 34–45.

Peslyak, A. 2011. John the Ripper password cracker. http://www.openwall.com/john/. (accessed 12/09/2011).

Price, K. 1998. Characteristics of a Good Intrusion Detection System COAST. http://www.scribd.com/doc/7148986/Intrusion-Detection-Systems. (accessed 01/01/2011).

Qayyum, A., Islam, M.H. and Jamil, M. 2005. Taxonomy of Statistical Based Anomaly Detetion Techniques for Intrusion Detection. Proceeding of International Conference on Emnerging Technologies. Islarnabad , Septenmber 17-18, pp. 270-276.

Quinlan, J.R. 1993. C4.5:Programs for Machine Learning. Morgan Kaufmann, San Fransisco.

Rouse, M. 2005. Definition Chernobyl Virus. http://searchsecurity.techtarget.com/definition/Chernobyl-virus. (accessed 16/07/2010).

Sabahi, F. and Movaghar, A. 2008. Intrusion Detection: A Survey. Proceeding of The 3rd International Conference on Systems and Networks Communications. October 26- 31, pp. 23-26.

Salem, B. and Karim, T. 2008. Context-based profiling for anomaly intrusion detection with diagnosis. Proceeding of The 3rd International Conference on Availability, Reliability and Security. Barcelona , March 4-7, pp. 618-623.

Sasu, E.C. 2010. Using constant traffic to specific IP destinations for detecting spoofing MAC addresses in Local Area Networks. Computational Cybernetics and Technical Information. May 27-29. pp. 677-681.

SRI International. 1995. Next-generation Intrusion Detection Expert System (NIDES) A Summary. Computer Science Laboratory.

SRI International. 2002. Intrusion Detection History. http://www.sdl.sri.com/programs/intrusion/history.html. (accessed 20/08/2010).

Stallings, W. 1995. Network and Inter-network Security Principles and Practice. Second Edition.

Subranmanian, M. 1999. Network Management: An introduction to principles and practice, Addison Wesley Professional:USA.

System (IDS). Proceeding of the International Conferecnce on Computer as a Tool.Serbia & Montenero, Belgrade, November 22-24, pp. 652-655.

Symantec. 1995. Intrusion Detection System. http://www.symantec.com/index.jsp. (accessed 25/09/2011).

Syverson, P. 1994. A Taxonomy of Replay Attacks [cryptographic protocols]. Proceeding of conference on Computer Security Foundations Workshop. pp. 187–91.

TechNet. 2003. How to DNS Works. http://technet.microsoft.com/en-us/library/cc772774(v=ws.10).aspx. (accessed 20/01/2012)

The University of Waikoto. 1997. Weka. http://www.cs.waikato.ac.nz/ml/weka/. (accessed 12/02/2012).

Wood, A. and Stankovic, J.A. 2002. Denial of Service in Sensor Networks. IEEE Computer, Vol. 35, No. 10, pp. 54–62.

Zhang, J. and Zulkernine, M. 2006. Anomaly Based Network Intrusion Detection with Unsupervised Outlier Detection. IEEE International Conference. Istanbul, June, pp. 2388-2393.

Zhang, W., Yang, Q. and Geng, Y. 2009. A Surveyof Anomaly Detection Methods in Networks. Modern Educational Technology Center Shandong Institute of Light Industry Jinan, Chaina.

Zheng, H., Shirochin, V.P. and Yueping, Y. 2005. An intelligent lightweight Intrusion Detection.

Zhou, M., Lee, R. and Lang, S. 2005. Locality-based Profile Analysis for Secondary Intrusion Detection. Proceedings of the 8th International Symposium on Parallel Architectures. USA, December 07-09.

Ziemba, G., Reed, D. and Traina, P. 1995. Security Considerations for IP Fragment Filtering. RFC1858.

ภาคผนวก

ภาคผนวก ก.

เครื่องมือที่เกี่ยวข้อง

เครื่องมือหรือซอฟต์แวร์ที่แสดงต่อไปนี้ เป็นเครื่องมือที่ใช้ในการพัฒนาระบบและหลักการที่ใช้สำหรับวิเคราะห์วิเคราะห์ข้อมูล ซึ่งมีรายละเอียดดังต่อไปนี้

1. Net-SNMP

ชุดโปรแกรมคำสั่งของ NET-SNMP ซึ่งจะใช้ในการพัฒนาอ็อบเจกต์และเรียกดูค่าของ อ็อบเจกต์ต่างๆ ที่ต้องการ โดยมีรูปแบบคำสั่งแบบ Command line ที่พัฒนาขึ้นโดยกลุ่ม Network Group มหาวิทยาลัย Carnegin-Mellon ชื่อ NET-SNMP ในปัจจุบันได้พัฒนามาถึงรุ่น 6.5.1 (NET-SNMP, 2011: Online) สามารถใช้งานได้ทั้งในระบบปฏิบัติการ Windows, Unix และ Linux โดยโปรแกรมนี้มีคำสั่งที่มีชื่อเดียวกับคำสั่งพื้นฐานของโพรโทคอลการจัดการเครือข่าย (SNMP) เช่น snmpget, snmpgetnext, snmpgetbulk, snmpset และ snmptrap นอกจากนี้ยังมีคำสั่งอื่นๆ ที่พัฒนาขึ้นจากการใช้คำสั่งพื้นฐานของ SNMP เช่น คำสั่ง snmpwalk, snmpbulkwalk, snmptable, snmpdf และ snmpnetstat และยังมีโปรแกรมคำสั่งช่วยในการทำงานอื่นๆ เช่น โปรแกรมคำสั่ง snmptranslate ไว้สำหรับเรียกดูรายละเอียดและโครงสร้างของอ็อบเจกต์ใน MIB

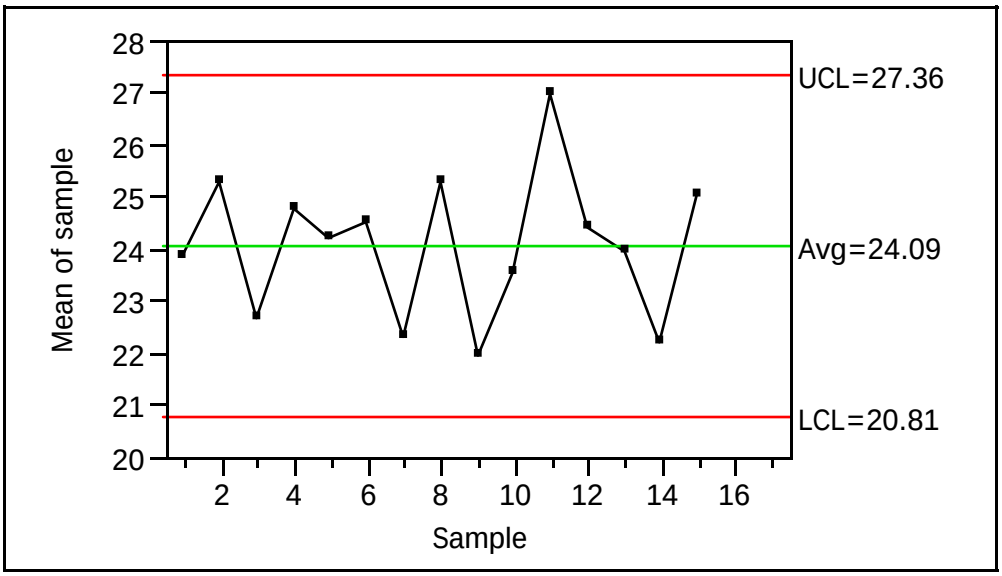
2. การเลือกคุณลักษณะ (Feature Selection)

การลดขนาดข้อมูล (Data Reduction) (Guyon และ Elisseeff, 2003) จัดเป็นกระบวนการหนึ่งในขั้นตอนการเตรียมข้อมูล นั่นคือการทำให้ข้อมูลตั้งต้นมีขนาดลดลง โดยสูญเสียลักษณะสำคัญของข้อมูลน้อยที่สุด และสูญเสียความถูกต้องของผลลัพธ์น้อยที่สุด เนื่องจากบางคุณลักษณะของข้อมูลมีความสำคัญต่อการจำแนกไม่เท่ากัน ดังนั้นด้วยเทคนิคการเลือกข้อมูลที่ดีจะทำให้สามารถเลือกข้อมูลที่มีความสำคัญและสามารถใช้เป็นตัวแทนของข้อมูลส่วนใหญ่ได้ และในความเป็นจริงมักจะเกิดเหตุการณ์ที่เรียกว่า ปัญหาของมิติข้อมูล

(Curse of Dimensionality) ขึ้นเสมอ ซึ่งก็หมายความว่า จำเป็นต้องลดมิติของข้อมูลลง (Dimensionality Reduction) เพื่อให้ตัวจำแนกประเภทสามารถทำงานได้อย่างถูกต้องมากขึ้น

3. กระบวนการควบคุมทางสถิติ (Statistical Process Control)

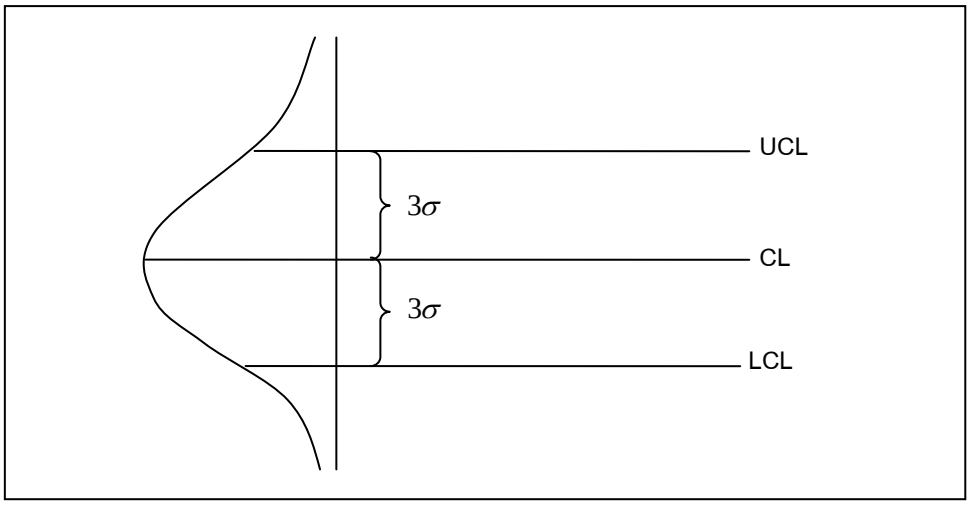
กระบวนการควบคุมทางสถิตหรือ Statistical Process Control Method (MacGregor และ Kourti, 2000) (Oakland, 2008) เป็นเครื่องมือที่ใช้อธิบายข้อมูลโดยใช้วิธีทางสถิติซึ่งข้อมูลที่ได้มานั้นมาจากการสำรวจ (Monitor) หรือเก็บรวบรวมไว้ซึ่งข้อมูลที่ได้รวบรวมมานั้นจะใช้เป็นตัวแทนเพื่อใช้ระบุความผิดปกติหรือการเบี่ยงเบนของข้อมูลที่กำลังเกิดขึ้นในปัจจุบันได้ ซึ่งในการนำเสนอหรือแสดงข้อมูลจะแสดงออกมาในรูปแบบของกราฟที่มีลักษณะดังภาพประกอบที่ ก-1 หรือเรียกว่า Control Chart ซึ่ง เป็นกราฟที่ใช้ตรวจสอบว่าค่าของตัวแปรที่สนใจเกิดความแปรผันเกินจากขอบเขตที่กำหนดไว้หรือไม่



ภาพประกอบที่ ก-1 ตัวอย่างกราฟ Control Chart

3.1 องค์ประกอบของ Control Chart

จากหลักทางสถิติที่ว่า ข้อมูลที่มีการแจกแจงแบบปกติ (Normal Distribution) ซึ่งมีค่าพารามิเตอร์ที่เกี่ยวข้อง 2 ค่า คือ ค่าเฉลี่ย (μ) และส่วนเบี่ยงเบนมาตรฐาน (σ) โดยมีโอกาสหรือความน่าจะเป็นอยู่ในช่วง $\pm 3\sigma$ เท่ากับ 0.9974 สามารถนำหลักการดังกล่าวมาสร้างเป็น Control Chart ซึ่งประกอบด้วยเส้นสำคัญ 3 เส้น ดังภาพประกอบที่ ก-2



ภาพประกอบที่ ก-2 องค์ประกอบของ Control Chart

เส้นแกนกลาง (Central Line: CL) เป็นค่าเฉลี่ยของค่าเฉลี่ยของข้อมูล ดังสมการที่ 1

$$\bar{\bar{x}} = \frac{\bar{x}_1 + \bar{x}_2 + \bar{x}_3 + \ldots \bar{x}_k}{k} \qquad (1)$$

โดยที่ $\bar{x}$ คือ ค่าเฉลี่ยของข้อมูล

k คือ จำนวนข้อมูล

ขีดจำกัดบน (Upper Control Limit: UCL) เป็นเส้นที่มีระยะห่างจากเส้นแกนกลางเท่ากับ 3σ ทางค่ามาก คำนวณได้จากสมการที่ 2

$$UCL = \bar{\bar{x}} + z\sigma_{\bar{x}} \qquad (2)$$

ขีดจำกัดล่าง (Lower Control Limit: LCL) เป็นเส้นที่มีระยะห่างจากเส้นกลางเท่ากับ 3σ ทางค่าน้อย คำนวณได้จากสมการที่ 3

$$LCL = \bar{\bar{x}} - z\sigma_{\bar{x}} \qquad (3)$$

โดยที่ $\bar{\bar{x}}$ คือ ค่าเฉลี่ยของค่าเฉลี่ยของกลุ่มในแต่ละช่วงข้อมูล

z คือ ค่ามาตรฐาน โดยในที่นี้มีด้วยกัน 2 ค่าคือ 2 และ 3 (2 หมายถึงค่าความเชื่อมั่นที่ 95.44%, 3 หมายถึง ค่าความเชื่อมันที่ 99.74%)

$\sigma_{\bar{x}}$ คือ ส่วนเบี่ยงเบนมาตรฐานของการกระจายเฉลี่ยกลุ่มข้อมูล โดยคำนวณได้จากสมการที่ 4 ซึ่งค่าของ σ คำนวณได้จากสมการที่ 5 ตามลำดับ

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}} \qquad (4)$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}\left(x_i - \bar{x}\right)^2}{n-1}} \qquad (5)$$

3.2 ซิกส์ซิกมา (6σ)

สมมุติให้ข้อมูลที่เกิดขึ้นเป็นการแจกแจงแบบปกติ (Normal Distribution) หรือการกระจายเป็นรูประฆังคว่ำทั้งหมด ค่าเฉลี่ยที่จุดกึ่งกลางของการกระจายตัวนั้นคือค่าที่ต้องการ ส่วนซิกส์มาคือหนึ่งช่วงของความเบี่ยงเบนมาตรฐานที่วัดจากจุดกึ่งกลางดังกล่าว และจะมีขอบเขตของการยอมรับได้ 2 ส่วนคือ ขอบเขตจำกัดบน (Upper Specific Limitation) และของจำกัดล่าง (Lower Specific Limitation) ซึ่งในนิยามของซิกส์ซิกมานี้ ถ้าขอบเขตบนและล่างอยู่ห่างจากค่าเฉลี่ยเป็นระยะ 3 ซิกมา ก็จะเรียกว่า 3 ระดับซิกส์มา (3 Sigma Level) ซึ่งในแต่ละระดับจะให้ค่าดังนี้

ตารางที่ ก-1 แสดงค่าความน่าเชื่อถือของซิกส์มาในระดับ 1σ -6σ

ระดับซิกส์มา	ค่าความน่าเชื่อถือ (Reliability)
1.0	68.26894921%
2.0	95.44997361%
3.0	99.73002039%
4.0	99.99366575%
5.0	99.99994267%
6.0	99.99999980%

ภาคผนวก ข.

รายละเอียดอ็อบเจกต์ใน MIB+

รายละเอียดข้อมูลอ็อบเจกต์ที่แสดงต่อไปนี้ เป็นการนำเสนอรายละเอียดของอ็อบเจกต์ตามรูปแบบการอธิบายอ็อบเจกต์ใน MIB-II โดยมีรายละเอียดของแต่ละอ็อบเจกต์ดังนี้

```
INTRUSION-DATA-MIB DEFINITIONS ::= BEGIN
IMPORTS
        enterprises
            FROM RFC1155-SMI
        OBJECT-TYPE
            FROM RFC1212;
PSUMIB    OBJECT IDENTIFIER ::= { enterprises 99999 }
intrusionData MODULE-IDENTITY
LAST-UPDATED "201101041612Z"    --24 December 2010, 15.20
ORGANIZATION "Computer System and Networking (CSN)Group"
CONTACT-INFO
                "       Patthama  Sangmee
                        Postal: Prince of Songkla University,
                        Thailand, 90120
                        E-mail: s5210220150@psu.ac.th
                "
```

```
DESCRIPTION
      "The MIB module using for the Intrusion detection"
::= { PSUMIB 3 }


                    -- The intrusion data general group --
        ipDataPkts OBJECT IDENTIFIER      ::= {intrusionData 1}
        icmpDataPkts OBJECT IDENTIFIER    ::= {intrusionData 2}
        tcpDataPkts OBJECT IDENTIFIER     ::= {intrusionData 3}
        udpDataPkts OBJECT IDENTIFIER     ::= {intrusionData 4}
        otherProtocol OBJECT IDENTIFIER   ::= {intrusionData 5}
        ports OBJECT IDENTIFIER           ::= {intrusionData 6}
        other OBJECT IDENTIFIER           ::= {intrusionData 7}
        tcpPorts OBJECT IDENTIFIER        ::={ports 1}
        udpPorts OBJECT IDENTIFIER        ::={ports 2}


                                  -- Scalars --
ipInAddSrcDest          OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      mandatory
    DESCRIPTION
      "The total number of IP packets that set the source and destination
          are the same."
    ::= { ipDataPkts 1 }
ipInBroadcast           OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
```

STATUS mandatory

DESCRIPTION

"The total number of IP packets that delivered to broadcast."

::= { ipDataPkts 2 }

ipInSizePkts OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS mandatory

DESCRIPTION

"The size of IP packet. "

::= { ipDataPkts 3 }

icmpInEchoPkts OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS mandatory

DESCRIPTION

"The total number of ICMP echo request messages that delivered to network."

::= { icmpDataPkts 1 }

icmpInReplyPkts OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS optional

DESCRIPTION

"The total number of ICMP echo reply message."

::= { icmpDataPkts 2 }

icmpInPkts OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS mandatory

DESCRIPTION

"The total number of ICMP message which received

Note that counter includes all those counted by ICMP message."

::= { icmpDataPkts 3 }

icmpInSizePkts OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS mandatory

DESCRIPTION

"The size of ICMP packets."

::= { icmpDataPkts 4 }

tcpInAckPkts OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS mandatory

DESCRIPTION

"The total number of TCP ACK packets that delivered to network."

::= { tcpDataPkts 1 }

tcpInFinPkts OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS mandatory

DESCRIPTION

"The total number of TCP FIN packets that delivered to network."

::= { tcpDataPkts 2 }

tcpInNoFlagSetPkts OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS optional

DESCRIPTION

"The total number of TCP packets that not to set flag and delivered to network."

::= { tcpDataPkts 3 }

tcpInPushPkts OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS optional

DESCRIPTION

"The total number of TCP PSH packets that delivered to network."

::= { tcpDataPkts 4}

tcpInRstPkts OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS optional

DESCRIPTION

"The total number of TCP RST packet that delivered to network."

::= { tcpDataPkts 5 }

tcpInSynPkts OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS mandatory

DESCRIPTION

"The total number of TCP SYN packets that delivered to network."

::= { tcpDataPkts 6 }

tcpInUrgPkts OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS optional

DESCRIPTION

"The total number of TCP URG packets that delivered to network."

```
    ::= { tcpDataPkts 7 }



udpInPkts               OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      mandatory
    DESCRIPTION
      "The total number of UDP packets that delivered to network."
    ::= { udpDataPkts 1 }

udpInLength             OBJECT-TYPE
    SYNTAX              Gauge32
    MAX-ACCESS  read-only
    STATUS      optional
    DESCRIPTION
      "The length of the UDP datagrams"
    ::= { udpDataPkts 2 }



tcpInSamePortNumber         OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     mandatory
    DESCRIPTION
      "The total number of TCP packets that through same port "
```

```
    ::= { tcpPorts 1 }



tcpInPortNumber25           OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     mandatory
    DESCRIPTION
      "The total number of TCP packets that through port 25"
    ::= { tcpPorts 2 }

tcpInPortNumber80           OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     mandatory
    DESCRIPTION
      "The total number of TCP packets that through port 80"
    ::= { tcpPorts 3 }

tcpInPortNumber139          OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     mandatory
    DESCRIPTION
      "The total number of TCP packets that through port 139"
    ::= { tcpPorts 4 }
```

```
udpInPortNumber7              OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     mandatory
    DESCRIPTION
      "The total number of UDP packets that through port 7"
    ::= { udpPorts 1 }

udpInPortNumber19             OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     mandatory
    DESCRIPTION
      "The total number of UDP packets that through port 19"
    ::= { udpPorts 2 }

udpInPortNumber53             OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     mandatory
    DESCRIPTION
      "The total number of UDP packets that through port 53"
    ::= { udpPorts 3 }
```

```
arpInRequestPkts                OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS read-only
    STATUS      mandatory
    DESCRIPTION
      "The total number of ARP request packets."
    ::= { otherProtocols 1 }
arpInReplyPkts                  OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS read-only
    STATUS      mandatory
    DESCRIPTION
      " The total number of ARP reply packets. "
    ::= { otherProtocols 2 }

cpuValue                OBJECT-TYPE
    SYNTAX      Gauge32
    MAX-ACCESS read-only
    STATUS      mandatory
    DESCRIPTION
      "The value of CPU usage."
      ::= { others 1 }
```

```
memoryValue             OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS read-only
    STATUS      mandatory
    DESCRIPTION
      "The value of memory usage."
      ::= { others 2 }

END
```

ภาคผนวก ค.

ไฟล์ snmpd.conf

ไฟล์ snmpd.conf เป็นไฟล์ที่ใช้ตั้งค่าการทำงานของโปรแกรม Net-SNMP โดยมีรายละเอียดของไฟล์ดังต่อไปนี้

```
############################################################
#
#   An example configuration file for configuring the ucd-snmp snmpd agent.
#
############################################################
#
# This file is intended to only be an example.  If, however, you want
# to use it, it should be placed in /etc/snmp/snmpd.conf.
# When the snmpd agent starts up, this is where it will look for it.
#
# You might be interested in generating your own snmpd.conf file using
# the "snmpconf" program (perl script) instead.  It's a nice menu
# based interface to writing well commented configuration files.  Try it!
#
# Note: This file is automatically generated from EXAMPLE.conf.def.
# Do NOT read the EXAMPLE.conf.def file! Instead, after you have run
# configure & make, and then make sure you read the EXAMPLE.conf file
# instead, as it will tailor itself to your configuration.

# All lines beginning with a '#' are comments and are intended for you
# to read.  All other lines are configuration commands for the agent.
```

```
#
# PLEASE: read the snmpd.conf(5) manual page as well!
#
###############################################################
# Access Control
###############################################################
# YOU SHOULD CHANGE THE "COMMUNITY" TOKEN BELOW TO A NEW
KEYWORD ONLY
# KNOWN AT YOUR SITE.  YOU *MUST* CHANGE THE NETWORK TOKEN BELOW
TO
# SOMETHING REFLECTING YOUR LOCAL NETWORK ADDRESS SPACE.
# By far, the most common question I get about the agent is "why won't
# it work?", when really it should be "how do I configure the agent to
# allow me to access it?"
#
# By default, the agent responds to the "public" community for read
# only access, if run out of the box without any configuration file in
# place.  The following examples show you other ways of configuring
# the agent so that you can change the community names, and give
# yourself write access as well.
#
# The following lines change the access permissions of the agent so
# that the COMMUNITY string provides read-only access to your entire
# NETWORK (EG: 10.10.10.0/24), and read/write access to only the
# localhost (127.0.0.1, not its real ipaddress).
#
# For more information, read the FAQ as well as the snmpd.conf(5)
# manual page.
```

```
####
# First, map the community name (COMMUNITY) into a security name
# (local and mynetwork, depending on where the request is coming from):
#       sec.name  source          community
#com2sec local     localhost       COMMUNITY
#com2sec mynetwork NETWORK/24      COMMUNITY
#com2sec mynetwork 172.25.3.245/24      public
####
# Second, map the security names into group names:
#                   sec.model  sec.name
                    group MyRWGroup     v1          local
                    group MyRWGroup     v2c         local
                    group MyRWGroup     usm         local
                    group MyROGroup v1          mynetwork
                    group MyROGroup v2c         mynetwork
                    group MyROGroup usm         mynetwork
####
# Third, create a view for us to let the groups have rights to:
#           incl/excl subtree                          mask
                    view all    included  .1                               80
####
# Finally, grant the 2 groups access to the 1 view with different
# write permissions:
#                context sec.model sec.level match  read   write  notif
                    access MyROGroup ""      any       noauth    exact  all    none   none
                    access MyRWGroup ""      any       noauth    exact  all    all    none
```

```
# rwuser: a SNMPv3 read-write user
#   arguments:  user [noauth|auth|priv] [restriction_oid]
                rwuser  admin
# rocommunity: a SNMPv1/SNMPv2c read-only access community name
#   arguments:  community [default|hostname|network/bits] [oid]
                rocommunity  public
# rwcommunity: a SNMPv1/SNMPv2c read-write access community name
#   arguments:  community [default|hostname|network/bits] [oid]
                rwcommunity  private
##########################################################################
# System contact information
#

# It is also possible to set the sysContact and sysLocation system
# variables through the snmpd.conf file.  **PLEASE NOTE** that setting
# the value of these objects here makes these objects READ-ONLY
# (regardless of any access control settings).  Any attempt to set the
# value of an object whose value is given here will fail with an error
# status of notWritable.

                syslocation Right here, right now.
                syscontact mercypeary@hotmail.com
                sysservices 76

# Example output of snmpwalk:
#   % snmpwalk -v 1 -c public localhost system
#   system.sysDescr.0 = "SunOS name sun4c"
#   system.sysObjectID.0 = OID: enterprises.ucdavis.ucdSnmpAgent.sunos4
```

```
#   system.sysUpTime.0 = Timeticks: (595637548) 68 days, 22:32:55
#   system.sysContact.0 = "Me <me@somewhere.org>"
#   system.sysName.0 = "name"
#   system.sysLocation.0 = "Right here, right now."
#   system.sysServices.0 = 72
##########################################################################
# Process checks.
#
#  The following are examples of how to use the agent to check for
#  processes running on the host.  The syntax looks something like:
#
#  proc NAME [MAX=0] [MIN=0]
#
#  NAME:  the name of the process to check for.  It must match
#         exactly (ie, http will not find httpd processes).
#  MAX:   the maximum number allowed to be running.  Defaults to 0.
#  MIN:   the minimum number to be running.  Defaults to 0.
#
#  Examples:
#
#  Make sure httpd is running
                    proc httpd
#  Make sure mountd is running
#proc mountd

                    #  Make sure there are no more than 4 ntalkds running, but 0 is ok too.
                    #proc ntalkd 4
```

```
#  Make sure at least one sendmail, but less than or equal to 10 are running.
#proc sendmail 10 1
#  A snmpwalk of the prTable would look something like this:
#
# % snmpwalk -v 1 -c public localhost .1.3.6.1.4.1.2021.2
# enterprises.ucdavis.procTable.prEntry.prIndex.1 = 1
# enterprises.ucdavis.procTable.prEntry.prIndex.2 = 2
# enterprises.ucdavis.procTable.prEntry.prIndex.3 = 3
# enterprises.ucdavis.procTable.prEntry.prNames.1 = "mountd"
# enterprises.ucdavis.procTable.prEntry.prNames.2 = "ntalkd"
# enterprises.ucdavis.procTable.prEntry.prNames.3 = "sendmail"
# enterprises.ucdavis.procTable.prEntry.prMin.1 = 0
# enterprises.ucdavis.procTable.prEntry.prMin.2 = 0
# enterprises.ucdavis.procTable.prEntry.prMin.3 = 1
# enterprises.ucdavis.procTable.prEntry.prMax.1 = 0
# enterprises.ucdavis.procTable.prEntry.prMax.2 = 4
# enterprises.ucdavis.procTable.prEntry.prMax.3 = 10
# enterprises.ucdavis.procTable.prEntry.prCount.1 = 0
# enterprises.ucdavis.procTable.prEntry.prCount.2 = 0
# enterprises.ucdavis.procTable.prEntry.prCount.3 = 1
# enterprises.ucdavis.procTable.prEntry.prErrorFlag.1 = 1
# enterprises.ucdavis.procTable.prEntry.prErrorFlag.2 = 0
# enterprises.ucdavis.procTable.prEntry.prErrorFlag.3 = 0
#  enterprises.ucdavis.procTable.prEntry.prErrMessage.1  =  "No  mountd  process
running."
# enterprises.ucdavis.procTable.prEntry.prErrMessage.2 = ""
# enterprises.ucdavis.procTable.prEntry.prErrMessage.3 = ""
# enterprises.ucdavis.procTable.prEntry.prErrFix.1 = 0
```

```
# enterprises.ucdavis.procTable.prEntry.prErrFix.2 = 0
# enterprises.ucdavis.procTable.prEntry.prErrFix.3 = 0
#
#  Note that the errorFlag for mountd is set to 1 because one is not
#  running (in this case an rpc.mountd is, but thats not good enough),
#  and the ErrMessage tells you what's wrong.  The configuration
#  imposed in the snmpd.conf file is also shown.
#
#  Special Case:  When the min and max numbers are both 0, it assumes
#  you want a max of infinity and a min of 1.
#
###########################################################################
# Executables/scripts
#
#
#  You can also have programs run by the agent that return a single
#  line of output and an exit code.  Here are two examples.
#
#  exec NAME PROGRAM [ARGS ...]
#
#  NAME:     A generic name.
#  PROGRAM:  The program to run.  Include the path!
#  ARGS:     optional arguments to be passed to the program
# a simple hello world
                    exec echotest /bin/echo hello world
# Run a shell script containing:
#
# #!/bin/sh
```

```
# echo hello world
# echo hi there
# exit 35
#
# Note:  this has been specifically commented out to prevent
# accidental security holes due to someone else on your system writing
# a /tmp/shtest before you do.  Uncomment to use it.
#
#exec shelltest /bin/sh /tmp/shtest

# Then,
# % snmpwalk -v 1 -c public localhost .1.3.6.1.4.1.2021.8
# enterprises.ucdavis.extTable.extEntry.extIndex.1 = 1
# enterprises.ucdavis.extTable.extEntry.extIndex.2 = 2
# enterprises.ucdavis.extTable.extEntry.extNames.1 = "echotest"
# enterprises.ucdavis.extTable.extEntry.extNames.2 = "shelltest"
# enterprises.ucdavis.extTable.extEntry.extCommand.1 = "/bin/echo hello world"
# enterprises.ucdavis.extTable.extEntry.extCommand.2 = "/bin/sh /tmp/shtest"
# enterprises.ucdavis.extTable.extEntry.extResult.1 = 0
# enterprises.ucdavis.extTable.extEntry.extResult.2 = 35
# enterprises.ucdavis.extTable.extEntry.extOutput.1 = "hello world."
# enterprises.ucdavis.extTable.extEntry.extOutput.2 = "hello world."
# enterprises.ucdavis.extTable.extEntry.extErrFix.1 = 0
# enterprises.ucdavis.extTable.extEntry.extErrFix.2 = 0

# Note that the second line of the /tmp/shtest shell script is cut
# off.  Also note that the exit status of 35 was returned.
```

```
# Print full status for web server and web objects via Mac OS X Server
# administration tool.
exec web_status /usr/sbin/serveradmin status web
exec wo_status /usr/sbin/serveradmin status webobjects

###########################################################################
# disk checks
#
# The agent can check the amount of available disk space, and make
# sure it is above a set limit.
# disk PATH [MIN=DEFDISKMINIMUMSPACE]
#
# PATH:  mount path to the disk in question.
# MIN:   Disks with space below this value will have the Mib's errorFlag set.
#        Default value = DEFDISKMINIMUMSPACE.
# Check the / partition and make sure it contains at least 10 megs.
                    disk / 10000
# % snmpwalk -v 1 -c public localhost .1.3.6.1.4.1.2021.9
# enterprises.ucdavis.diskTable.dskEntry.diskIndex.1 = 0
# enterprises.ucdavis.diskTable.dskEntry.diskPath.1 = "/" Hex: 2F
# enterprises.ucdavis.diskTable.dskEntry.diskDevice.1 = "/dev/dsk/c201d6s0"
# enterprises.ucdavis.diskTable.dskEntry.diskMinimum.1 = 10000
# enterprises.ucdavis.diskTable.dskEntry.diskTotal.1 = 837130
# enterprises.ucdavis.diskTable.dskEntry.diskAvail.1 = 316325
# enterprises.ucdavis.diskTable.dskEntry.diskUsed.1 = 437092
# enterprises.ucdavis.diskTable.dskEntry.diskPercent.1 = 58
# enterprises.ucdavis.diskTable.dskEntry.diskErrorFlag.1 = 0
# enterprises.ucdavis.diskTable.dskEntry.diskErrorMsg.1 = ""
```

```
##################################################
# load average checks
#

#          load          [1MAX=DEFMAXLOADAVE]          [5MAX=DEFMAXLOADAVE]
[15MAX=DEFMAXLOADAVE]
#
# 1MAX:   If the 1 minute load average is above this limit at query
#         time, the errorFlag will be set.
# 5MAX:   Similar, but for 5 min average.
# 15MAX:  Similar, but for 15 min average.

# Check for loads:
#load 12 14 14

# % snmpwalk -v 1 -c public localhost .1.3.6.1.4.1.2021.10
# enterprises.ucdavis.loadTable.laEntry.loadaveIndex.1 = 1
# enterprises.ucdavis.loadTable.laEntry.loadaveIndex.2 = 2
# enterprises.ucdavis.loadTable.laEntry.loadaveIndex.3 = 3
# enterprises.ucdavis.loadTable.laEntry.loadaveNames.1 = "Load-1"
# enterprises.ucdavis.loadTable.laEntry.loadaveNames.2 = "Load-5"
# enterprises.ucdavis.loadTable.laEntry.loadaveNames.3 = "Load-15"
# enterprises.ucdavis.loadTable.laEntry.loadaveLoad.1 = "0.49" Hex: 30 2E 34 39
# enterprises.ucdavis.loadTable.laEntry.loadaveLoad.2 = "0.31" Hex: 30 2E 33 31
# enterprises.ucdavis.loadTable.laEntry.loadaveLoad.3 = "0.26" Hex: 30 2E 32 36
# enterprises.ucdavis.loadTable.laEntry.loadaveConfig.1 = "12.00"
# enterprises.ucdavis.loadTable.laEntry.loadaveConfig.2 = "14.00"
```

```
# enterprises.ucdavis.loadTable.laEntry.loadaveConfig.3 = "14.00"
# enterprises.ucdavis.loadTable.laEntry.loadaveErrorFlag.1 = 0
# enterprises.ucdavis.loadTable.laEntry.loadaveErrorFlag.2 = 0
# enterprises.ucdavis.loadTable.laEntry.loadaveErrorFlag.3 = 0
# enterprises.ucdavis.loadTable.laEntry.loadaveErrMessage.1 = ""
# enterprises.ucdavis.loadTable.laEntry.loadaveErrMessage.2 = ""
# enterprises.ucdavis.loadTable.laEntry.loadaveErrMessage.3 = ""

#####################################################
# Extensible sections.
#
# This alleviates the multiple line output problem found in the
# previous executable mib by placing each mib in its own mib table:
# Run a shell script containing:
#
# #!/bin/sh
# echo hello world
# echo hi there
# exit 35
#
# Note:  this has been specifically commented out to prevent
# accidental security holes due to someone else on your system writing
# a /tmp/shtest before you do.  Uncomment to use it.
#
# exec .1.3.6.1.4.1.2021.50 shelltest /bin/sh /tmp/shtest
# % snmpwalk -v 1 -c public localhost .1.3.6.1.4.1.2021.50
# enterprises.ucdavis.50.1.1 = 1
# enterprises.ucdavis.50.2.1 = "shelltest"
```

```
# enterprises.ucdavis.50.3.1 = "/bin/sh /tmp/shtest"
# enterprises.ucdavis.50.100.1 = 35
# enterprises.ucdavis.50.101.1 = "hello world."
# enterprises.ucdavis.50.101.2 = "hi there."
# enterprises.ucdavis.50.102.1 = 0

# Now the Output has grown to two lines, and we can see the 'hi
# there.' output as the second line from our shell script.
#
# Note that you must alter the mib.txt file to be correct if you want
# the .50.* outputs above to change to reasonable text descriptions.

# Other ideas:
#
# exec .1.3.6.1.4.1.2021.51 ps /bin/ps
# exec .1.3.6.1.4.1.2021.52 top /usr/local/bin/top
# exec .1.3.6.1.4.1.2021.53 mailq /usr/bin/mailq
##################################################################
# Pass through control.
#
# Usage:
#   pass MIBOID EXEC-COMMAND
# This will pass total control of the mib underneath the MIBOID
# portion of the mib to the EXEC-COMMAND.
# Note:  You'll have to change the path of the passtest script to your
# source directory or install it in the given location.
# Example:  (see the script for details)
#           (commented out here since it requires that you place the
```

```
# script in the right location. (its not installed by default))
# pass .1.3.6.1.4.1.2021.255 /bin/sh PREFIX/local/passtest
# % snmpwalk -v 1 -c public localhost .1.3.6.1.4.1.2021.255
# enterprises.ucdavis.255.1 = "life the universe and everything"
# enterprises.ucdavis.255.2.1 = 42
# enterprises.ucdavis.255.2.2 = OID: 42.42.42
# enterprises.ucdavis.255.3 = Timeticks: (363136200) 42 days, 0:42:42
# enterprises.ucdavis.255.4 = IpAddress: 127.0.0.1
# enterprises.ucdavis.255.5 = 42
# enterprises.ucdavis.255.6 = Gauge: 42
#
# % snmpget -v 1 -c public localhost .1.3.6.1.4.1.2021.255.5
# enterprises.ucdavis.255.5 = 42
#
# % snmpset -v 1 -c public localhost .1.3.6.1.4.1.2021.255.1 s "New string"
# enterprises.ucdavis.255.1 = "New string"
#
# For specific usage information, see the man/snmpd.conf.5 manual page
# as well as the local/passtest script used in the above example.
##############################################################
# Subagent control
# The agent can support subagents using a number of extension mechanisms.
# From the 4.2.1 release, AgentX support is being compiled in by default.
# To use this mechanism, simply uncomment the following directive.
            master  agentx
#   Please see the file README.agentx for more details.
##############################################################
```

ประวัติผู้เขียน

ชื่อ สกุล นางสาวปัทมา แสงหมี

รหัสประจำตัวนักศึกษา 5210220150

วุฒิการศึกษา

วุฒิ	ชื่อสถาบัน	ปีที่สำเร็จการศึกษา
วท.บ. (วิทยาการคอมพิวเตอร์)	มหาวิทยาลัยสงขลานครินทร์	2551

ทุนการศึกษา (ที่ได้รับระหว่างศึกษา)

ปีการศึกษา 2551 ทุนผู้ช่วยวิจัย

ปีการศึกษา 2553 ทุนอุดหนุนวิจัยเพื่อวิทยานิพนธ์ บัณฑิตวิทยาลัย

การตีพิมพ์เผยแพร่ผลงาน

ปัทมา แสงหมี และนิษฐิดา เอลซ์. 2554. การประยุกต์ใช้ Management Information Base (MIB) สำหรับข้อมูลโพรไฟล์ของระบบตรวจจับการบุกรุก. The 15th International Computer Science and Engineering Conference (ICSEC 2011). กรุงเทพมหานคร ประเทศไทย. หน้า 31-36

Sangmee, P. Thanon, N., and Elz, N. 2012. Anomaly Detection using New MIB Traffic Parameters based on Profile. Proceeding of The 8th International Conference on Computing Technology and Information Management 2012 (ICCM 2012).Seoul,Korea, pp. 648-653.