โครงงานวิศวกรรมคอมพิวเตอร์

เรื่อง

โปรแกรมวิเคราะห์ลักษณะการบุกรุก
สำหรับระบบตรวจจับการบุกรุกทางเครือข่าย
Attack Signature Analysis for
Network-based Intrusion Detection System

โดย

นาย เฉลิมคล จงสงวน

เสนอ

ภาควิชาวิศวกรรมศาสตร์คอมพิวเตอร์

คณะวิศวกรรมศาสตร์มหาวิทยาลัยเกษตรศาสตร์

พ.ศ. 2544

โปรแกรมวิเคราะห์ลักษณะการบุกรุก สำหรับระบบตรวจจับการบุกรุกทางเครือข่าย

Attack Signature Analysis for Network-based Intrusion Detection System

นาย เฉลิมคล จงสงวน

ห้องปฏิบัติการวิจัยเครือข่ายประยุกต์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์
อาจารย์ที่ปรึกษา : ผศ. สุรศักดิ์ สงวนพงษ์

บทคัดย่อ

โครงงานนี้เป็นการออกแบบ และพัฒนาซอฟต์แวร์ สำหรับตรวจจับการโจมตีทางเครือข่ายจากภายนอกแบบ network-based หาค่าที่เกินปกติของการบุกรุกแบบ flooding ของแพ็กเก็ต และสังเกตความผิดปกติของการเชื่อมต่อที่เกิดขึ้น นำไปสู่การสรุปผล เพื่อป้องกันเครือข่ายจากการโจมตีดังกล่าว โดยส่งคำสั่งไปให้ส่วนควบคุมเกี่ยวกับไฟร์วอลล์อีกต่อหนึ่ง.

สารบัญ

สารบัญรูป

หน้า

บทที่ 1
บทนำ

1.1 วัตถุประสงค์ของโครงงาน

เพื่อตรวจหาการบุกรุกแบบ flood ซึ่งตรวจจับได้ยากกว่าการบุกรุกแบบแพ็กเก็ตเดี่ยว เมื่อพบการโจมตีแล้วจึงส่งคำสั่งไปยังไฟร์วอลล์ เพื่อป้องกันการโจมตีที่จะเข้ามาอีก.

1.2 ลักษณะและขอบเขตของโครงงาน

โครงงานนี้มีเป้าหมายที่จะออกแบบการตรวจจับการบุกรุกแบบ flood สำหรับป้องกันเครือข่ายคอมพิวเตอร์จากภายนอก โดยได้รับความร่วมมือจากโครงงานการพัฒนาตัวตรวจจับการบุกรุกด้วยไฟร์วอลล์ ในส่วนของการป้องกันการเชื่อมต่อ.

โดยการวิเคราะห์ทั้งหมดในโครงงานนี้ จะทำโดยแบ่งออกตามเลขไอพีของสถานีเป้าหมาย ซึ่งเป็นสถานีภายในเครือข่าย ที่ต้องการป้องกันจากการโจมตี (โดยสถานีดังกล่าวระบุไว้ในไฟล์ station.conf ซึ่งรูปแบบของไฟล์จะกล่าวต่อไปในภายหลัง) และแบ่งแยกต่อไปตามพอร์ตต่างๆ ที่สำคัญๆเช่น telnet (tcp port:21), ssh (tcp port:22), telnet (tcp port:23) เป็นต้น

1.3 เครื่องมือพัฒนาโปรแกรม

- ระบบปฏิบัติการลินุกซ์
- ANSI C

บทที่ 2
ทฤษฎีพื้นฐาน

2. ประเภทของระบบตรวจจับการบุกรุก หรือ IDS

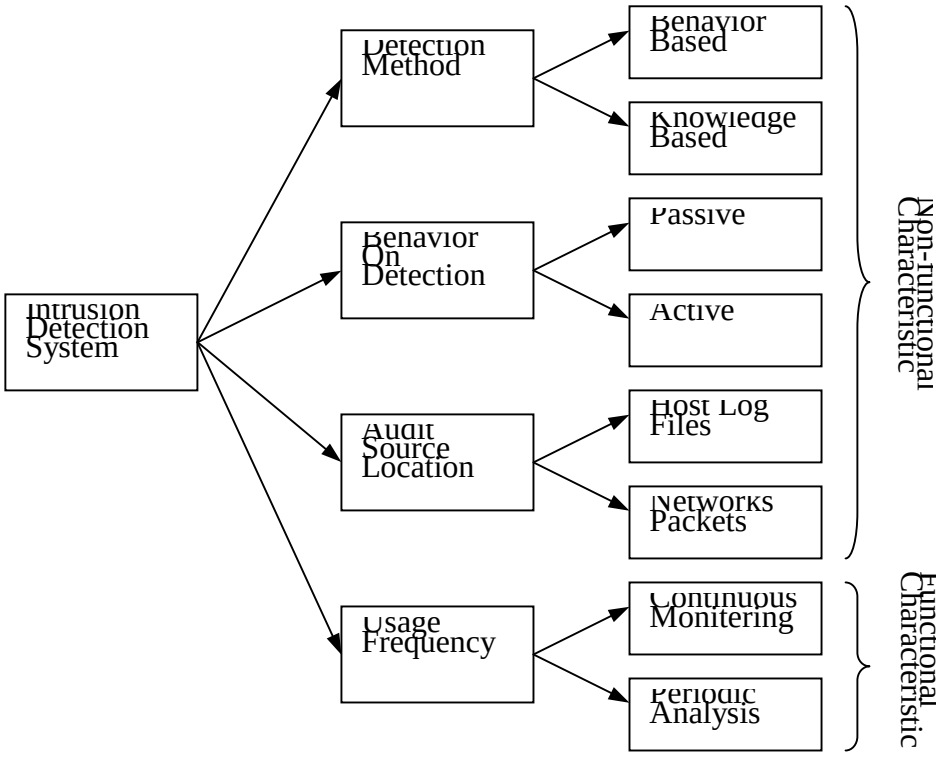
แนวคิดการสร้าง IDS มีหลายแบบ ดังรูปที่ 2

Detection method แสดงลักษณะของตัววิเคราะห์. มี 2 แบบ คือ behavior-based จะใช้พฤติกรรมปกติของระบบ knowledge-based ใช้ข้อมูลเกี่ยวกับการบุกรุก.

Behavior on detection อธิบายการตอบโต้ต่อการบุกรุก. ถ้า Corrective (ตอบโต้โดยปิดช่องโหว่) และ proactive (เช่น ให้ logout และ ตัดการเชื่อมต่อ) โดยทันที ก็เป็นแบบแอคทีฟ. ถ้าแสดงเพียงการเตือน (รวมถึง paging และอื่นๆ) ก็เป็นแบบพาสซีฟ.

Audit source location แบ่งตามแหล่งข้อมูลเข้า ของข้อมูลที่จะวิเคราะห์. อาจเป็น audit trail, system log หรือ แพ็กเก็ตในเครือข่าย.

Usage frequency ประเภทแรกจะทำงานแบบเวลาจริง อีกประเภทหนึ่งจะทำงานเป็นช่วงๆ.



รูปที่ 2.1 ลักษณะของ IDS

3 หัวข้อแรกจะเกี่ยวกับลักษณะการทำงาน เพราะกล่าวถึงการทำงานภายใน (คือ ข้อมูลอินพุท กลไกการใช้เหตุผล และการตอบโต้) หัวข้อสุดท้ายจะแยกระหว่าง RTID(Real-Time Intrusion Detection) กับสแกนเนอร์(ที่ใช้ตรวจความปลอดภัยระบบ แต่ที่แท้จริงแล้วไม่ใช่ IDS).

2.1 ระบบตรวจจับการบุกรุกแบบ knowledge-based และbahavior-based.

- Knowledge-based IDS, misuse detection หรือ detection by appearance - หาหลักฐานการบุกรุก โดยมีพื้นฐานอยู่บนความรู้เกี่ยวกับการบุกรุกที่รู้จัก.
- Behavior-based IDS, anomaly detection หรือ detection by behavior – หาการเบี่ยงเบนในโมเดลเกี่ยวกับพฤติกรรมที่ผิดปกติ เปรียบเทียบกับการสังเกตการทำงานโดยปกติของระบบ.

2.1.1 ระบบตรวจจับการบุกรุกแบบ Knowledge-based

ประยุกต์ฐานความรู้ เกี่ยวกับการโจมตีเฉพาะอย่าง และความอ่อนแอของระบบ. คอยสังเกตความพยายามที่จะจู่โจมจุดอ่อนนั้น เมื่อพบก็ดำเนินการต่อไป(เมื่อไม่พบว่าเป็นการโจมตีก็ถือว่ายอมรับ). ประสิทธิภาพในด้านความแม่นยำดี แต่ในด้านความสมบูรณ์ต้องการการอัพเดตความรู้เกี่ยวกับการบุกรุกเสมอๆ.

ข้อดี คือ มี False alarm rate ต่ำมาก.

ข้อเสีย คือ มีความยุ่งยาก ในการรวบรวมข้อมูลเกี่ยวกับการบุกรุก การอัพเดตก็ต้องทำด้วยความระมัดระวังและกินเวลามาก. ความรู้เกี่ยวกับการบุกรุกนี้ขึ้นกับระบบปฏิบัติการ เวอร์ชัน แพล็ตฟอร์ม และแอพพลิเคชัน เครื่องมือตรวจจับการบุกรุกจึงเกี่ยวกับสภาพแวดล้อมอย่างมาก. อย่างไรก็ตาม การตรวจจับการโจมตีจากภายใน(insider attacker) ถือว่าทำได้ยากกว่า เพราะถือว่าไม่มีช่องโหว่ที่ถูกโจมตีโดยผู้บุกรุก.

2.1.1.1 Expert systems.

Expert systems ประกอบด้วย ชุดของกฎที่อธิบายการบุกรุก. Expert systems จะแปลงเหตุการณ์อินพุทเป็นข้อเท็จจริงที่บรรจุ semantic signification , inference engine จะใช้กฎแปลง fact ดังกล่าวเป็นผลสรุปอีกที. วิธีนี้จะเพิ่ม abstraction level โดยเพิ่มความหมายให้มัน.

Rule-based language เป็นภาษาที่ใช้แสดงความรู้เกี่ยวกับการบุกรุก. ทำให้สามารถเปรียบเทียบหาหลักฐานการบุกรุกในสายอินพุทได้อย่างเป็นระบบ. และยังใช้หาแอพลิเคชันที่เหมาะสมกับนโยบายความปลอดภัยในองค์กรได้ด้วย.

Rule-based language มีข้อจำกัดบางประการ คือ

- Knowledge engineering (เกี่ยวกับ completeness) การเอาความรู้จากการบุกรุก และการแปลงความรู้ดังกล่าวเป็นกฎ ทำได้ไม่ง่ายนัก. บางครั้งข้อมูลส่วนที่ต้องการก็ไม่มีอยู่ในอินพุทนั้น เพราะว่าช่องโหว่หนึ่งๆ นั้นสามารถถูกโจมตีได้หลายทาง ทำให้ต้องมีหลายกฎ.
- Processing speed (เกี่ยวกับ performance) เนื่องจากเชลล์ของ expert system ต้องการอินพุททุกอย่าง เพื่อให้กฎทำงานต่อไป. แม้ว่าบาง expert system อนุญาตให้คอมไพล์กฎได้ แต่ส่วนใหญ่ประสิทธิภาพก็ต่ำ.

จาก processing speed ที่ไม่ดี ทำให้ shell ของ expert system ถูกใช้เป็นเพียงโปรโตไทป์เท่านั้น.

2.1.1.2 Signature analysis.

ก็ต้องใช้ความรู้ที่ได้มา เช่นเดียวกับ expert system แต่ใช้ต่างกัน. Semantic description ของการบุกรุกจะถูกแปลงเป็นข้อมูล ที่อาจถูกพบในสายอินพุตอย่างตรงไปตรงมา. ตัวอย่างเช่น ลำดับการโจมตีถูกแปลงเป็นลำดับเหตุการณ์อินพุท(แบบที่มันสร้างขึ้น) หรือแปลงเป็นข้อมูลในรูปแบบที่ใช้ค้นหาในสายอินพุทได้. วิธีนี้จะลด semantic level ของลักษณะการบุกรุก.

เทคนิคนี้เป็นเทคนิคที่มีประสิทธิภาพมาก จึงถูกใช้อย่างแพร่หลาย. แต่ก็มีข้อเสียเช่นเดียวกับ knowledge-based ทั่วไปคือ ต้องการการอัพเกรดบ่อยๆ. และ signature ก็มีมากมาย อย่างน้อย 1 signature ต่อ 1 OS ที่ต้องการติดตั้ง.

2.1.1.3 State-transition analysis

เทคนิคนี้เสนอโดย Porras และ Kemmerer ถูกใช้ครั้งแรกบนระบบปฏิบัติการยุนิกซ์. โดยแนวความคิดแล้วเทคนิคนี้เหมือนกับการอ้างเหตุผลแบบ model-based อธิบายการโจมตีด้วยชุดของ goal และ transition แต่แสดงด้วย state-transition diagram.

2.1.2 ระบบตรวจจับการบุกรุกแบบ Bahavior-based

โดยสันนิษฐานว่า การบุกรุกถูกพบได้ โดยการสังเกตพฤติกรรมที่เบี่ยงเบนออกจากปกติ(หรือที่คาดไว้)ของระบบหรือยูสเซอร์. รูปแบบพฤติกรรมที่เหมาะสมจะถูกเก็บจากแหล่งข้อมูลอ้างอิงที่มาจากหลายวิธี. ถ้าพฤติกรรมในปัจจุบันมีการเบี่ยงเบนออกจะดำเนินการต่อไป. กล่าวอีกนัยหนึ่ง พฤติกรรมอื่นที่ไม่ได้เรียนรู้ไว้ก็ถือว่าเป็นการบุกรุก. Behavior-based นี้อาจทำให้สมบูรณ์ได้ แต่การทำให้มีความแม่นยำยังทำได้ยาก.

ข้อดี คือ มันสามารถตรวจจับความพยายามบุกรุกแบบใหม่ได้ แม้แต่ค้นหาการบุกแบบใหม่(บางส่วน)โดยอัตโนมัติ. แทบไม่ขึ้นกับระบบปฏิบัติการ และสามารถใช้ช่วยตรวจการใช้ privilege ในทางที่ผิด(ซึ่งไม่เกี่ยวกับช่องโหว่ของระบบ)ได้ด้วย.

ข้อเสียหลัก คือ false alarm rate สูง เพราะว่าขณะเรียนรู้อาจไม่ครอบคลุมพฤติกรรมของระบบทั้งหมด. พฤติกรรมของระบบยังเปลี่ยนไปตามกาลเวลาจึงต้องอัพเดตเป็นระยะๆ ทำให้ไม่สามารถทำงานได้ หรือ อาจทำให้ false alarm มากขึ้นอีก. โดยเฉพาะอย่างยิ่งหากอัพเดตขณะมีการบุกรุกจริง ก็จะยอมรับการบุกรุกนั้นเป็นพฤติกรรมปกติด้วย.

2.1.2.1 Statistics

เป็นแบบที่นิยมใช้กันอย่างแพร่หลาย. พฤติกรรมของผู้ใช้หรือของระบบ จะถูกวัดโดยสุ่มหาค่าตัวแปรชุดหนึ่งในช่วงระยะเวลาหนึ่ง. เช่น เวลาล็อคอิน-ล็อคเอ้าท์ การใช้ทรัพยากรต่างๆ และ ปริมาณการใช้โปรเซสเซอร์-หน่วยความจำ-ดิสก์ ของเซสชัน. ช่วงเวลาระหว่างการสุ่มหาค่าอาจสั้น(2-3 นาที) หรือยาวได้(เดือน).

รูปแบบอย่างง่ายใช้การตรวจสอบกับค่าเฉลี่ย หาค่าที่เกินเมื่อคำนวณเทียบกับส่วนเบี่ยงเบนมาตรฐาน. หรือแม้แต่เปรียบเทียบตัวแปรของยูสเซอร์ กับกลุ่มของยูสเซอร์นั้น. ดังนั้น จึงมีการพัฒนาจนซับซ้อนขึ้น มีการเปรียบเทียบพฤติกรรมระยะสั้นกับพฤติกรรมระยะยาวของยูสเซอร์(โดยปกติแล้ว จะอัพเดตเมื่อพฤติกรรมของยูสเซอร์เปลี่ยนแปลง).

2.1.2.2 Expert systems.

เก็บข้อมูลเกี่ยวกับพฤติกรรมที่เหมาะสมของยูสเซอร์ในช่วงเวลาหนึ่ง นำมาสร้างเป็นกฎของพฤติกรรมปกติ เปรียบเทียบกิจกรรมของยูสซอร์กับกฎดังกล่าว. ทั้งนี้กฎก็ต้องถูกสร้างขึ้นใหม่เสมอๆ เพื่อให้ทันสมัยต่อรูปแบบการใช้.

Expert system นี้เหมาะกับข้อมูลการใช้ที่มีพื้นอยู่บนนโยบาย. แต่มีประสิทธิภาพด้อยกว่าแบบ statistic เมื่อทำงานกับข้อมูลขนาดใหญ่.

2.1.2.3 Neural netwerks.

เป็นอัลกอริทึมที่เรียนรู้ความสัมพันธ์ระหว่างเวกเตอร์อินพุท-เอาท์พุท และ generalize เพื่อให้ได้อินพุท-เอาท์พุทเวกเตอร์ใหม่. Neural network ถูกใช้ในแบบ knowledge-based เพื่อเรียนรู้ร่องรอยการโจมตี แล้วค้นหาในสายอินพุท. อย่างไรก็ตาม ก็ไม่สามารถใช้ระบุเหตุผลหรืออธิบายการบุกรุกได้.

ดังนั้น ประโยชน์หลักของ neural network สำหรับการตรวจจับการบุกรุก คือ ใช้เรียนรู้พฤติกรรม(ของยูสเซอร์, เดมอน). ข้อได้เปรียบ เมื่อเทียบกับแบบสถิติ คือ สามารถแสดงความสัมพันธ์ที่ไม่ใช่เส้นตรงของตัวแปร และเรียนรู้ปรับปรุงตัวมันเองได้อย่างอัตโนมัติ. จากการทดลองพบว่า พฤติกรรมของผู้ดูแลระบบสามารถทำนายได้ (เช่น การทำแบบอัตโนมัติของระบบ เดมอน และอื่นๆ), พฤติกรรมของยูสเซอร์ส่วนใหญ่ทำนายได้ และ มีเพียงยูสเซอร์ส่วนน้อยเท่านั้นที่ทำนายไม่ได้. Neural network เป็นเทคนิคที่เกี่ยวข้องกับการคำนวณมาก จึงไม่ค่อยมีผู้นิยมใช้กันมากนัก.

2.1.2.4 User Intention Identification.

ถูกพัฒนาในโปรเจค SECURENET สร้างพฤติกรรมปกติของยูสเซอร์ โดยใช้เซตของงานระดับสูง ที่ยูสเซอร์นั้นใช้กับระบบ. แปลงงานเหล่านั้นให้กลายเป็น action ซึ่งเกี่ยวข้องกับ audit event ของระบบ. ตัววิเคราะห์จะเก็บชุดของ task ของแต่ละยูสเซอร์ เมื่อพบ action ที่ไม่เข้ากับรูปแบบนี้ก็จะแสดงการเตือน. (พบเฉพาะในโปเจค SECURENET)

2.1.2.5 Computer immunology.

Forrest et al. สร้างโมเดลพฤติกรรมปกติของบริการในยูนิกซ์ ประกอบด้วยลำดับ system call สั้นๆของโปรเซส. โดยมีสมมุติฐานว่า การบุกรุกที่มีการเปลี่ยนแปลงโค้ดโปรแกรมดูเหมือนว่าจะมีลำดับที่ไม่ปกติ. เริ่มต้นด้วยการเก็บชุดพฤติกรรมที่เหมาะสมของบริการต่างๆ แล้วสร้างตารางของลำดับ system call ที่ดีเพื่อใช้อ้างอิงกับลำดับของ system call ในปัจจุบัน.

ข้อดี ถ้าเก็บข้อมูลได้มากพอ เทคนิคนี้จะมี false alarm rate ต่ำมาก.

ข้อเสีย เทคนิคนี้ไม่ครอบคลุมการตั้งค่าผิด เกี่ยวกับบริการ เมื่อการโจมตีใช้บริการของระบบอย่างถูกต้อง เพื่อการเข้าถึงสิ่งที่ไม่ได้รับอนุญาต.

2.2 ระบบตรวจจับการบุกรุกแบบ Passive และ Active.

ส่วนใหญ่เครื่องมือตรวจจับการบุกรุกเป็นแบบพาสซีพ หมายความว่า เมื่อพบการโจมตีก็จะแสดงการเตือน แต่ไม่มีการตอบโต้เพื่อขัดขวางการโจมตีนั้น. ซึ่งให้ความหมายในเชิงการวิจัยมากกว่า และเครื่องมือพวกนี้มักจะสร้าง false alarm จำนวนมาก ซึ่งอาจส่งผลกับสภาพของระบบได้.

เครื่องมือจับการบุกรุกส่วนใหญ่ มีการวิเคราะห์เป็นช่วงๆ และมีการแก้ไขบ้างเมื่อพบการตั้งค่าที่ผิดพลาด. เพื่อความปลอดภัยยิ่งขึ้นก็สามารถกลับไปสู่ค่าเดิมได้อย่างรวดเร็ว. ตัวอย่างได้แก่ Ballista ของ Secure Network.

ช่วงหลังๆ นี้ก็มีความสนใจ ในการเพิ่มมาตรการตอบโต้มากขึ้น เช่น Realcesure, NetRanger, Webstalker ก็เพิ่มความสามารถในการตัดการเชื่อมต่อ ป้องกันข้อมูลที่เข้ามา ตั้งค่าอุปกรณ์พวกเราเตอร์-ไฟร์วอลล์. ทำให้เครื่องมือเหล่านี้มีความน่าเชื่อถือมากขึ้น.

2.3 ระบบตรวจจับการบุกรุกแบบ Host-based และ network-based.

เริ่มแรกการตรวจจับการบุกรุกเป็นแบบ Host-based เพื่อป้องกันเครื่องเมนเฟรม ประกอบกับผู้ใช้ทุกคนก็อยู่ภายในบริเวณเดียวกัน. ทำให้การป้องกันเป็นไปได้ง่าย เพราะข้อมูลจากภายนอกมีน้อยมาก การวิเคราะห์ก็ทำบนเครื่องเดียวกัน(หรืออีกเครื่องแยกต่างหาก) แล้วจึงรายงานพฤติกรรมที่น่าสงสัย.

เมื่อรูปแบบการคำนวณเปลี่ยนแปลงจากเมนเฟรม ไปเป็นเครือข่ายแบบกระจาย จึงมีการพัฒนาให้เข้ากับเครือข่าย โดยเริ่มแรกทำให้มีการติดต่อสื่อสารระหว่าง host-based ด้วยกัน. ในเครือข่ายแบบกระจาย ยูสเซอร์สามารถเปลี่ยนแปลงตัวตนระหว่างข้าม hop และปล่อยการโจมตีไปยังสถานีต่างๆได้. ดังนั้น IDS ภายในเวิร์คสเตชัน จึงมีการแลกเปลี่ยนข้อมูลกับอีกเครื่อง ในหลายระดับ เช่น แลกเปลี่ยนสายข้อมูลดิบที่จะวิเคราะห์ ผ่านเครือข่าย(à la Stalker) หรือ โดยส่งการเตือนจากการวิเคราะห์ภายใน. ซึ่งทั้ง 2 วิธีนี้ มีข้อเสีย คือ การส่งผ่านข้อมูลดิบจะกินแบนด์วิทย์มาก และ การวิเคราะห์ภายในส่งผลกระทบต่อประสิทธิภาพของระบบ.

เมื่อการใช้อินเตอร์เน็ตเริ่มแพร่หลาย IDS จึงหันมาเน้นป้องกันการโจมตีทางเครือข่าย (เช่น DNS spoofing, TCT hijacking, port scanning, ping of death เป็นต้น) ซึ่งไม่สามารถตรวจจับได้โดยใช้เพียงข้อมูลภายในโฮสต์นั้น. จึงมีการพัฒนาให้ตรวจสอบแพ็กเก็ตในเครือข่าย(รวมถึงข้อมูลภายใน) เพื่อมองหาคำสั่งที่น่าสงสัย. เพียงผู้ดูแลระบบใช้เครื่องมือจำพวกนี้จำนวนไม่มาก ก็สามารถป้องกันการโจมตีได้เกือบทั้งหมด.

พวกลูกผสมที่เป็นกึ่งๆ network-based กับ host-based ในสภาพแวดล้อมแบบมัลติโฮสต์ (เช่น เครือข่ายของเวิร์คสเตชั่น เป็นต้น) เช่น ชุดของ DIDS ซึ่งประกอบด้วย Haystack(มีการรันทุกเครื่องเพื่อตรวจการโจมตีภายใน) กับ NSM(สังเกตเครือข่าย) แล้วทั้งสองส่วนก็รายงานให้ DIDS Director เพื่อวิเคราะห์ขั้นสุดท้ายอีกที.

นอกจากนี้ ยังมีฟีเจอร์อื่น เช่น สามารถตรวจสอบอุปกรณ์ภายในเครือข่าย เช่น ไฟร์วอลล์ (NetStalker) เราเตอร์(NetRanger) มองหาการโจมตีที่เฉพาะเจาะจงอุปกรณ์ในเครือข่ายมากขึ้น.

2.3.1 ข้อมูลที่ใช้ใน Host-based

ข้อมูลภายในโฮสต์เป็นที่มาเพียงแหล่งเดียวที่จะได้พฤติกรรมของยูสเซอร์ ในขณะเดียวกันก็ถูกแก้ไขได้ง่ายเช่นกันในกรณีที่การโจมตีสำเร็จ เพราะฉะนั้นแบบ host-based นี้จึงต้องทำงานแบบเวลาจริง เพื่อค้นพบและส่งสัญญาณเตือนก่อนที่ผู้โจมตีจะโจมตีได้สำเร็จ แล้วกลบร่องรอยของตน.

2.3.1.1 System sources.

ระบบปฏิบัติการทุกตัวมีคำสั่งที่ใช้ดู ภาพรวมของโปรเซสที่กำลังทำงานอยู่ในขณะนั้น ในยูนิกส์ เช่น ps, pstat, vmstat, getrlimit เป็นต้น. คำสั่งเหล่านี้ให้ข้อมูลที่ละเอียด เพราะเข้าไปตรวจสอบการใช้หน่วยความจำของเคอร์เนลโดยตรง แต่ก็ยังมีความยุ่งยากถ้าต้องการข้อมูลเหล่านี้อย่างต่อเนื่อง เพราะข้อมูลเหล่านี้ไม่ได้เก็บอย่างเป็นโครงสร้าง.

2.3.1.2 Accounting.

เป็นหนึ่งในแหล่งข้อมูลเกี่ยวกับพฤติกรรมของระบบที่เก่าที่สุด. ให้ข้อมูลเกี่ยวกับทรัพยากรของระบบที่ยูสเซอร์ใช้ เช่น เวลาที่ใช้ประมวลผล, หน่วยความจำ, ดิสก์ หรือการใช้เน็ตเวิร์ค เป็นต้น. พบได้ทั่วไปตั้งแต่ อุปกรณ์เครือข่าย จนถึงเมนเฟรม. ทำให้มีความพยายามที่จะใช้เป็นแหล่งข้อมูลเพื่อวิเคราะห์.

ในยูนิกส์ accounting เป็นแหล่งข้อมูลที่มีความเป็นมาตรฐานมาก รูปแบบเรคอร์ดของ accounting เหมือนกันหมด ข้อมูลจะถูกบีบอัดเพื่อประหยัดเนื้อที่ และมีโอเวอร์เฮดที่เล็ก(เพื่อแนะนำโปรเซส). ยังถูกใส่ไว้ในระบบปฏิบัติการใหม่ๆ อย่างประณีต ง่ายต่อการแก้ไขและนำไปใช้.

อย่างไรก็ตาม ข้อมูลของ accounting ก็มีข้อเสียมาก ทำให้ไม่สมควรนำไปใช้เพื่อความปลอดภัยระบบ. เช่น โดยดีฟอลต์แล้ว ไฟล์ accounting อยู่ในดิสก์บริเวณเดียวกับ /tmp ซึ่งยูสเซอร์เติมได้ถึง 90% ทำให้ account หยุดทำงาน. และข้อเสียอื่นๆ อีก เช่น

- ขาดพารามิเตอร์ – accounting มีเพียงเปิดและปิดเท่านั้น. ไม่สามารถใช้เฉพาะยูสเซอร์ได้.
- ขาด time stamp ที่ละเอียด – เวลาในเรคอร์ด accounting มีความละเอียดแค่วินาที ไม่สามารถใช้ในการจัดลำดับได้ ทั้งนี้เวลาที่ได้เป็นเวลาที่จบการทำงานเท่านั้น ทำให้สร้างลิสต์ของลำดับการเข้าทำงานไม่ได้ ซึ่งเป็นส่วนสำคัญที่ใช้ในเทคนิคตรวจสอบบางอย่าง.
- ระบุรายละเอียดคำสั่งไม่ได้ – เพียง 8 ตัวอักษรแรกของคำสั่งยูสเซอร์เท่านั้น ที่ถูกเก็บในเรคอร์ด accounting. อาร์กิวเมนต์ต่างๆ ซึ่งเป็นรายละเอียดของคำสั่งหายไป. ทำให้เทคนิค knowledge อาจตรวจไม่พบม้าโทรจันได้.
- ขาดข้อมูลการทำงานของเดมอน – เพราะเก็บข้อมูลขณะเลิกทำงานเท่านั้น. เดมอนทำงานตลอดเวลาจึงไม่ถูกตรวจสอบ.
- ความล่าช้าของข้อมูล – เรคอร์ด accounting ถูกสร้างขึ้นเมื่อเลิกการทำงานเท่านั้น. ทำให้ควบคุมความเสียหาย ในเมื่อเกิดการบุกรุกเสร็จแล้วเท่านั้น.

ด้วยข้อเสียเหล่านี้ knowledge-based จึงมักไม่ใช้ accounting. ถึงมี behavior-based ใช้บ้างก็น้อยมากส่วนโมดูลสถิติ และ neural network ของ Hyperview ก็มีใช้แค่เพียงส่วนเสริมเท่านั้น.

2.3.1.3 Syslog.

เป็นบริการทั่วไปของยูนิกส์และระบบปฏิบัติการอื่น เก็บสตริงข้อความจากแอพพลิเคชันบันทึกเวลาเริ่มต้น. เพื่อใช้เป็นข้อมูลสำหรับวิเคราะห์พฤติกรรมผู้ใช้ ความผิดพลาดจากการใช้โปรแกรมในสถานการณ์ต่างๆ กัน ซึ่งอาจมีผลต่อความปลอดภัยของระบบ.

Syslog นั้นใช้ได้ง่ายเป็นแหล่งข้อมูลที่ใช้ได้ทันที. ถูกใช้โดยผู้พัฒนาแอพพลิเคชันต่างๆ แอพพลิเคชั่นและบริการเน็ตเวิร์คจำนวนมากเรียกใช้ เช่น login, sendmail, nfs, http รวมถึงเครื่องมือเพื่อความปลอดภัยอย่างเช่น sudo, klaxon หรือ TCP wrapper. มี IDS สองสามตัวที่ใช้ syslog เช่น Swatch แม้ว่าข้อมูลที่ได้จาก syslog แต่ละเครื่องนั้นไม่มาก ในเครือข่ายขนาดใหญ่ก็สามารถสร้างเมสเส็จจำนวนมากได้ แต่มีส่วนสำคัญเกี่ยวกับความปลอดภัยเพียงเล็กน้อยเท่านั้น. Swatch นั้นมีฟีเจอร์ซึ่งลดขนาดของเมสเส็จได้โดยนำมารวมกัน(เช่น ถ้าหลายสถานีรายงานเหมือนๆกันว่าเซิร์ฟเวอร์ nfs ตาย. ก็รวมเป็นอันเดียว เป็นต้น) และเน้นเฉพาะส่วนที่เกี่ยวกับความปลอดภัยเท่านั้น.

แต่ Syslog เองก็ไม่ได้ปลอดภัยนัก. เดมอนของ syslog ก็ถูกทำให้ buffer overflow ได้หากรันโค้ดบางอย่าง(จากเอกสารของ CERT).

2.3.1.4 C2 security audit.

(C2 คือระดับความปลอดภัยของ TCSEC) Security audit บันทึกอีเวนต์ทุกอย่าง ที่เกี่ยวข้องกับความปลอดภัยของระบบ. และต่อมากลายเป็นแหล่งสายข้อมูลที่ใช้เพื่อวิเคราะห์กรบุกรุก เช่น BSM ของ SUN, ชุดโปรแกรมของ Shield หรือ audit ของ AIX.

Security audit ทั้งหมดมีหลักการเดียวกัน คือ บันทึกทุกคำสั่งที่ทำงานข้ามไปมา ระหว่างบริเวณของยูสเซอร์ กับบริเวณ TCB(Trust Computing Base). ด้วยเหตุผลของโมเดลความลอดภัยที่ว่าโปรแกรมที่รันอยู่ใน TCB นั้นเชื่อถือได้ เพราะฉะนั้น การทำงานในส่วนของยูสเซอร์จะไม่สามารถทำอันตรายความปลอดภัยของระบบได้ และการทำงานที่ส่งผลกับระบบทำได้เมื่อยูสเซอร์เรียกใช้บริการจาก TCB เท่านั้น.

ในระบบยูนิกส์ TCB อยู่ในเคอร์เนล. ดังนั้นระบบ audit จะบันทึกการทำงานของ system call ของทุกโปรเซสที่เรียกโดยยูสเซอร์. เปรียบเทียบกับเส้นทางการทำงานแบบเต็ม สาย audit จะหาภาพรวมของ context switch, การใช้หน่วยความจำ, ซีมาฟอร์ภายใน. และมักมีการทำนายลำดับการทำงานไว้ล่วงหน้าด้วย.

เรคอร์ด security audit ของยูนิกซ์ประกอบด้วยข้อมูลของอีเวนต์มากมาย รวมถึงรายละเอียดตัวตนของยูสเซอร์หรือกลุ่ม(ตั้งแต่ล็อคอิน จนกระทั่งเรียก system call), พารามิเตอร์ของ system call(ชื่อไฟล์ รวมพาท, อาร์กิวเมนต์ของบรรทัดคำสั่ง เป็นต้น), return code จากการทำงาน และ error code.

ข้อได้เปรียบหลักๆ ของ security audit คือ

- การระบุตัวตนของยูสเซอร์ที่ละเอียด – ตั้งแต่ล็อคอิน (login identity), ตัวตนแท้จริง (real/current identity), ตัวตนที่ต้องการ (effective identity ดูจากบิต set-user-id), ตัวตนที่ของกลุ่มที่ต้องการ (effective group identity ดูจากบิต set-user-id).
- การแบ่งเหตุการณ์ audit เพิ่มให้เป็นคลาส เพื่อให้การคอนฟิกระบบง่ายขึ้น.
- พารามิเตอร์ของข้อมูลที่ละเอียดยิ่งขึ้น โดยรวบรวมข้อมูลจากยูสเซอร์, คลาส, audit event หรือ ความสำเร็จ/ล้มเหลวในการเรียก system call.
- ปิดเครื่อง เมื่อระบบ audit อยู่ในภาวะล้มเหลว(โดยปกติเมื่อเนื้อที่หมด).

ข้อเสียหลักๆ ของ security audit คือ

- มีการใช้ทรัพยากรระบบมาก เมื่อมีการวิเคราะห์อย่างละเอียด. การใช้โปรเซสเซอร์อาจลดลงถึง 20% และความต้องการใช้เนื้อที่บนดิสก์สูงมาก(ทั้ง local และส่วนรวม).
- มีความเป็นไปได้ของการบุกรุกแบบ DOS โดยทำให้ระบบไฟล์ของ audit เต็ม.
- ระบบ audit แก้ไขได้ยากเพราะมีพารามิเตอร์เยอะ. ค่าที่ตั้งมาจากผู้ผลิตมีประสิทธิภาพต่ำมาก โดยเก็บเฉพาะชุดเหตุการณ์ที่ไม่ค่อยพบ(เช่น การทำงานของผู้ดูแลระบบ, การล็อคอิน-ล็อคเอ้าท์). แต่ข้อมูลที่ IDS ต้องการละเอียดกว่า เช่น การใช้ไฟล์ การรันโปรเซสต่างๆ เป็นต้น.
- ข้อมูลที่ได้ใช้ประโยชน์ยาก เนื่องจากขนาดและความซับซ้อน เพราะอินเตอร์เฟสของระบบ audit ที่ต่างกัน และรูปแบบของเรคอร์ด audit ก็ต่างกันในแต่ละระบบปฏิบัติการด้วย.

Security audit ของ C2 เป็นแหล่งข้อมูลพื้นฐานของโปรโตไทป์และเครื่องมือ ตรวจจับการบุกรุกแบบ host-based จำนวนมาก เพราะเป็นแหล่งข้อมูลเดียวที่เชื่อถือได้ ในการรวบรวมข้อมูลที่ละเอียดสำหรับการกระทำของยูสเซอร์. ในปัจจุบันก็ยังมีการค้นคว้าเกี่ยวกับสิ่งที่ควรอยู่ใน security audit trail และรูปแบบสามัญที่ควรจะเป็นของเรคอร์ด audit trail.

2.3.2 ข้อมูลที่ใช้ใน Network-based

2.3.2.1 ข้อมูลจาก SNMP.

SNMP(Simple Network Management Protocol) MIB(Management Information Base) เป็นแหล่งข้อมูลที่ใช้เพื่อการจัดการเครือข่าย. ประกอบด้วยข้อมูลการตั้งค่าต่างๆ(routing tables, address, names) และข้อมูลเกี่ยวกับประสิทธิภาพ หรือ accounting (ตัวนับเพื่อวัดทราฟฟิกของอินเตอร์เฟสต่างๆ และที่ชั้นต่างๆกันของ เน็ตเวิร์ค). โดยกล่าวถึงการทดลองในโปรเจค SECURENET ที่ใช้ SNMP V1 MIB แบบง่ายๆ สำหรับอีเธอร์เน็ตและทีซีพี/ไอพี. ซึ่งการตรวจจับการบุกรุกในงานอื่นๆ จะใช้ SNMPv2 และ SNMPv3.

โปรเจค SECURENET พยายามค้นหา เกี่ยวกับตัวนับ(counter)ใน MIB เพื่อใช้เป็นข้อมูลใน IDS แบบ behavior-based. เริ่มโดยตรวจสอบตัวนับที่ระดับอินเตอร์เฟส เพราะเป็นที่เดียวที่สามารถแยกความแตกต่าง ระหว่างข้อมูลที่ถูกส่งผ่านสาย กับข้อมูลที่ถูกส่งภายในระบบปฏิบัติการผ่านลูปแบ็คอินเตอร์เฟส. โดยต้นแบบมีการนับแบบเพิ่ม 2 อย่างคือ จำนวนไบต์และจำนวนแพ็กเก็ตที่เข้าและออก ทุกๆ 5 นาที. แต่

ผลที่ได้จากการใช้ค่าเฉลี่ยกับส่วนเบี่ยงเบนมาตรฐานอย่างง่าย ก็ไม่เป็นที่น่าพอใจนัก เนื่องจากส่วนเบี่ยงเบนมาตรฐานมักมากกว่าค่าเฉลี่ยทุกชุดข้อมูล และไม่มีความสัมพันธ์กันระหว่าง 2 อินเตอร์เฟสเลย.

ตัวนับของ MIB ที่โปรโตคอลระดับสูงขึ้นก็ไม่ได้มีข้อมูลเพิ่มขึ้นมากนัก. ในชั้นของ IP, TCP และ UDP ตัวนับพวกนี้ก็มีพฤติกรรมคล้ายกัน เนื่องจากตัวนับในชั้นเหล่านี้มีมากมาย จึงไม่มีการหาความสัมพันธ์ทั้งหมดที่เป็นไปได้ระหว่างตัวนับเหล่านี้. ตัวนับของ ICMP นั้นมีลักษณะที่ตรงกันมากกว่า แต่ก็ยังไม่มีการทดลองเทียบกับการโจมตีของ ICMP.

จากการศึกษาพบว่า MIB หลายอันของ SMNP นั้นสามารถใช้เป็นแหล่งข้อมูลของ IDS ได้. แต่การใช้ SNMPv2 เนื่องจากขาดความสอดคล้องกัน เกี่ยวกับฟีเจอร์ในด้านความปลอดภัยนั้นเอง. ภายหลังก็มี SNMPv3 ซึ่งมีฟีเจอร์ต่างๆ ให้เครื่องมือรักษาความปลอดภัยใหม่ๆ ได้ใช้.

2.3.2.2 Network packets.

จากความสนใจในการใช้สนิฟเฟอร์ที่เพิ่มขึ้นในหมู่ผู้โจมตี ก็มีการใช้เพื่อรวบรวมข้อมูลเกี่ยวกับเหตุการณ์ต่างๆ ที่เกิดขึ้นภายเครือข่าย. ซึ่งตรงกับการเปลี่ยนแปลงรูปแบบการคำนวณ จากแบบศูนย์กลางเป็นแบบกระจาย. การเข้าถึงสถานีที่สำคัญในปัจจุบัน ก็มักผ่านเครือข่าย ดังนั้น การจับแพ็กเก็ตก่อนเข้าเวิร์ฟเวอร์จึงเป็นวิธีที่มีประสิทธิภาพมากที่สุด ในการตรวจสอบเซิร์ฟเวอร์นั้นๆ.

ปัญหาของการใช้ network packet คือ

- ใช้เฉพาะตรวจจับการโจมตีกับเน็ตเวิร์ค – การโจมตีแบบ DOS ไม่สามารถตรวจได้โดยการค้นหาในข้อมูล audit ในโฮสต์นั้น แต่ต้องวิเคราะห์ข้อมูลที่ส่งผ่านเครือข่าย.
- ผลกระทบกับประสิทธิภาพของระบบ – ข้อมูลทั้งหมดเก็บอยู่บนเครื่องแยกต่างหากได้ เรื่องของการตั้งค่าต่างๆและประสิทธิภาพ ไม่มีผลกระทบต่อระบบทั้งหมด.
- รูปแบบสาย audit แตกต่างกัน – ปัจจุบันมีการทำให้เป็นมาตรฐานเดียวกันโดย TCP/IP ให้ความสะดวกในการได้ข้อมูล, รูปแบบ และการเปลี่ยนแพล็ตฟอร์ม.
- เครื่องมือที่วิเคราะห์ข้อมูลของแพ็กเก็ต ซึ่งตรวจจับการบุกรุกแบบ signature analysis เพื่อประสิทธิภาพ ก็ยังต้องการข้อมูลเกี่ยวกับชนิดของเครื่อง หรือแอพลิเคชันที่เจาะจงด้วย.

ข้อเสียที่มี เช่น

- ระบุผู้กระทำผิดได้ยากเมื่อเกิดการโจมตีขึ้น – เพราะไม่มีการเชื่อมโยงที่เชื่อถือได้ ระหว่างข้อมูลภายในแพ็กเก็ต กับการระบุยูสเซอร์ที่ส่งคำสั่งไปยังโฮสต์นั้นๆ.
- จากเครือข่ายแบบสวิทซ์ (switched Ethernet, switched Token Ring, ATM) – ก็ยังไม่แน่ชัดว่าควรจะวางสนิฟเฟอร์ไว้ที่ใด. มักอยู่บนบนสวิทซ์ (ได้ข้อมูลที่ดีกว่า แต่เสียค่าใช้จ่ายมากกว่า) หรือบนเกตเวย์ระหว่างระบบภายใน กับโลกภายนอก. อย่างไรก็ดี ก็ควรใช้เครือข่ายแบบสวิทซ์ ถูกสนิฟได้ยากกว่าจึงมีความปลอดภัยมากกว่า.
- การเข้ารหัส - ทำให้ไม่สามารถวิเคราะห์ข้อมูลสำคัญที่อยู่ภายในได้. แต่ก็เป็นไปได้ที่จะซ่อนส่วนประกอบภายในได้โดยไม่ได้เข้ารหัส.
- การสแกนอย่างเป็นระบบ (อย่างเช่นที่ ไฟร์วอลล์ เป็นต้น) ทำได้ยาก เพราะอาจทำให้เกิดปัญหาคอขวดขึ้นได้.

- เครื่องมือพวกนี้ยังอ่อนแอต่อการโจมตีแบบ DOS เมื่อยังต้องการข้อมูลเครือข่ายที่ได้จากระบบปฏิบัติการ. โดยเมื่อระบบปฏิบัติการนั้นมี network stack ที่อ่อนแอต่อการโจมตี IDS ก็ย่อมอ่อนแอด้วย.

ปัจจุบันทั้งซอฟท์แวร์การค้าและในด้านการวิจัย ใช้แพ็กเก็ตเครือข่ายเป็นแหล่งข้อมูลที่ใช้ตรวจสอบ. Ptacek และ Newsham กล่าวว่าเป็นไปได้ที่ผู้โจมตีที่มีความชำนาญ จะหลบการตรวจจับได้. และระบุอีกว่า การจัดการเกี่ยวกับ IP fragmentation ก็ยังไม่ดีนัก การใช้ wildcard และลำดับของอักษรควบคุม ในโปรโตคอลต่างๆ เช่น http ทำให้หลบการตรวจจับแบบ signature ได้.

2.4 การวิเคราะห์อย่างต่อเนื่อง และ การวิเคราะห์เป็นช่วงๆ

แนวทางการวิเคราะห์มี 2 แบบคือ ทำต่อเนื่องตลอดเวลากับแบ่งเป็นช่วงๆ. แบบต่อเนื่องหรือแบบเวลาจริง ต้องการข้อมูลของเหตุการณ์ทันทีหลังจากที่เกิดขึ้น. แบบสแตติกหาภาพ(snapshot)ของระบบขณะหนึ่ง แล้ววิเคราะห์หาซอฟท์แวร์ที่อ่อนแอต่อการโจมตี, การตั้งค่าผิดพลาด หรืออื่นๆ.

แบบสแตติกใช้แก้ปัญหา เกี่ยวกับระดับความปลอดภัยของการตั้งค่า. ในสภาพแวดล้อมแบบโฮสต์ เช่น COPS และ Tiger, สำหรับสภาพแวดล้อมแบบเน็ตเวิร์ค เช่น Satan และ Ballista (หรือปัจจุบันคือ CyberCop Scanner). การตรวจสอบไวรัสก็อยู่ในกลุ่มเดียวกัน -- ตรวจดิสก์หารูปแบบเพื่อระบุไวรัสที่รู้จัก. การตรวจสอบเหล่านี้รวมถึงเวอร์ชันของแอพพลิเคชั่นที่ถูกติดตั้ง เพื่อให้แน่ใจว่าใช้ patch ล่าสุด, ระบุสิ่งที่อยู่ภายในไฟล์พิเศษในไดเรคทอรีโฮมของยูสเซอร์, หรือระบุการตั้งค่าของบริการเครือข่ายที่เปิด. ภาพของระบบที่ได้จากการวิเคราะห์นั้นใช้ได้ในช่วงเวลาหนึ่งเท่านั้น.

แม้เครื่องมือเหล่านี้เป็นที่รู้จักและใช้กันอย่างแพร่หลายโดยผู้ดูแลระบบ แต่ยังไม่ประสิทธิภาพที่จะเชื่อถือได้มากนัก. การรันเครื่องมือตรวจซ่อมความปลอดภัยเหล่านี้กินเวลามาก โดยเฉพาะในเครือข่าย เพราะทุกเครื่องต้องถูกตรวจสอบทั้งหมด. และจำเป็นต้องรัน 2 ครั้งต่อเนื่องกัน(ห่างกันประมาณ 1 วัน) เพื่อจะได้เห็นว่ามีจุดบกพร่องต่างๆ ลดลง.

เครื่องมือจับการบุกรุกแบบไดนามิกส์ ตรวจสอบการกระทำที่เกิดขึ้นบนระบบ จะเฝ้าดูการทำงานแบบเวลาจริง ตรวจไฟล์ audit หรือแพ็กเก็ตเครือข่ายที่ได้สะสมในช่วงเวลาหนึ่ง. การตรวจแบบไดนามิกส์ประยุกต์การวิเคราะห์แบบเวลาจริง และอนุญาตให้ประเมินความปลอดภัยของระบบ. อย่างไรก็ตาม การเคลื่อนย้าย audit แล้ววิเคราะห์ ก็ยังเป็นโปรเซสที่เสียค่าใช้จ่ายมาก.

บทที่ 3
แนวคิดและการออกแบบ

3.1 หลักการและเหตุผล

โครงงานนี้มีจุดประสงค์เพื่อตรวจจับการบุกรุกทางเครือข่าย ที่ใช้กันอย่างแพร่หลายคือเครือข่ายอีเธอร์เน็ต การบุกรุกเหล่านี้สร้างความเสียหายให้แก่อุปกรณ์เครือข่าย ตลอดจนสถานีภายในซึ่งเป็นเป้าหมายของการโจมตีนั้นๆ โครงงานนี้จึงออกแบบมาเพื่อตรวจจับการบุกรุกแบบหลายแพ็กเก็ต ทำงานร่วมกับส่วนตรวจจับการบุกรุกแบบแพ็กเก็ตเดี่ยว และส่วนควบคุมไฟร์วอลล์เพื่อสกัดกั้นการบุกรุก. เพื่อป้องกันเครือข่ายภายใน จากการโจมตีที่รู้จักและสามารถป้องกันได้.

การทำงานโดยรวมมีลักษณะเป็นสนิฟเฟอร์ ตรวจข้อมูลที่ส่งผ่านไปมาอยู่ในสายตัวนำของเครือข่าย โดยปรับให้ฮาร์ดแวร์เครือข่ายอยู่ในภาวะโพรมิคูอัส จับข้อมูลทั้งหมดขึ้นมาวิเคราะห์หาการบุกรุก.

หลักการที่ใช้ในโครงงาน

Detection method - knowledge-based

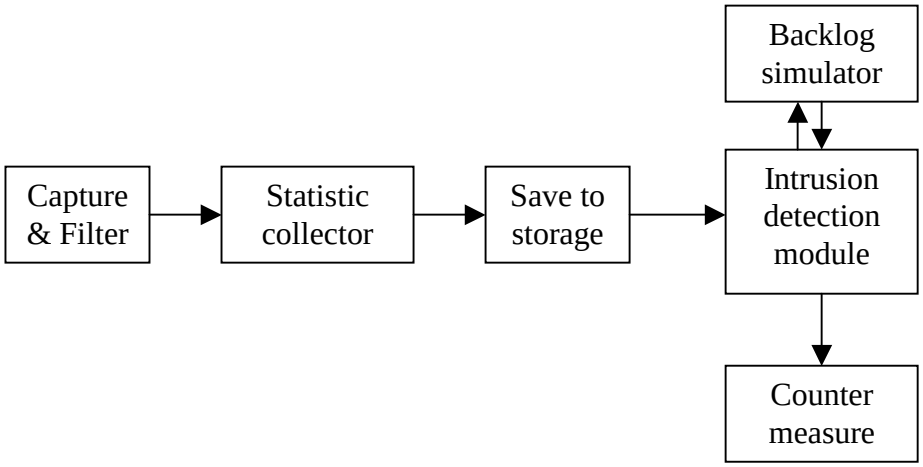
Behavior on detection – กึ่ง Active

Audit source location – Network Packets

Usage frequency – Periodic Analysis

3.2 โครงสร้างการทำงานของระบบ

โครงสร้างการทำงานของระบบ ดังรูป

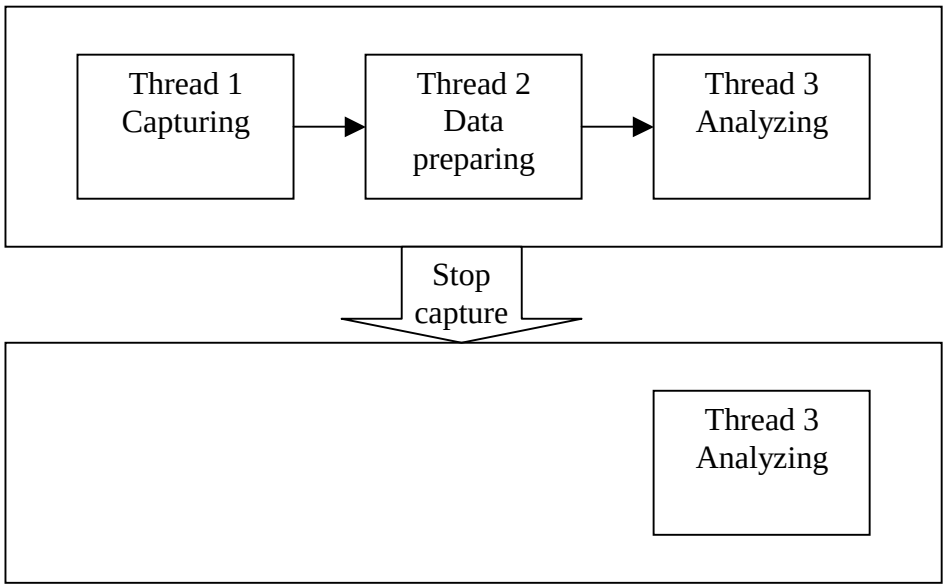


รูปที่ 3.1 โครงสร้างการทำงานของตัววิเคราะห์ลักษณะการบุกรุก

Capture & Filter	ทำหน้าที่จับแพ็กเก็ตในเครือข่าย
Statistic collector	หาค่าทางสถิติเกี่ยวแพ็กเก็ตเพื่อเป็นข้อมูลให้ส่วนวิเคราะห์
Save to storage	เก็บข้อมูลทั้งหมดทั้งข้อมูลดิบ และทางสถิติลงดิสก์ แล้วเรียกใช้อีกทีโดยส่วน

	ตรวจจับการบุกรุก
Intrusion detection module	วิเคราะห์การบุกรุกแบบหลายแพ็กเก็ต
Backlog simulator	จำลองแบ็คล็อคของพอร์ตต่างๆ ที่สำคัญของโปรโตคอลทีซีพี

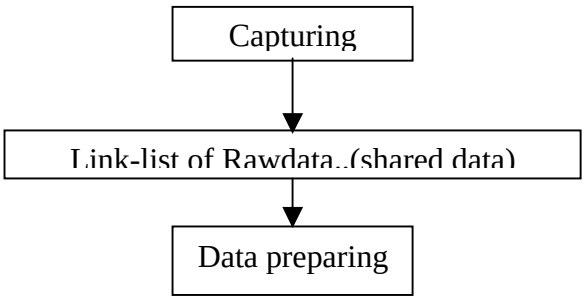
การทำงานของโปรแกรม แบ่งการทำงานออกเป็น 3 ส่วน ทั้ง 3 ส่วนทำงานพร้อมกันตลอดเวลา ในส่วนของ การทำงานในขณะจับแพ็กเก็ต และการทำงานในส่วนการวิเคราะห์อย่างเดียวหลังจากเลิกจับแพ็กเก็ตข้อมูลแล้ว



รูปที่ 3.2 ขั้นตอนการทำงานหลัก

- thread แรกทำหน้าที่จับข้อมูลเพียงอย่างเดียว เพื่อจับข้อมูลในเครือข่ายให้ทันเวลา
- thread ที่สองทำหน้าที่เตรียมและจัดเก็บข้อมูลทั้งหมดลงในดิสก์
- thread ที่สามดึงข้อมูลจากดิสก์ แล้ววิเคราะห์ข้อมูลทั้งหมดที่ได้จาก thread ที่สอง

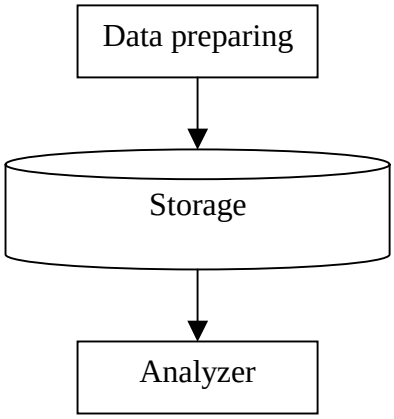
ความสัมพันธ์ของ thread แรกและ thread ที่สอง มีการส่งผ่านข้อมูลทางเดียว ตามหลักการของผู้ผลิตและผู้บริโภค (producer - consumer) ดังรูป



รูปที่ 3.3 ข้อมูลดิบถูกส่งผ่านระหว่าง thread 1-2

ข้อมูลที่ส่งระหว่างกันมีโครงสร้างเป็นลิงก์ลิสต์ Data preparing ได้รับพอยต์เตอร์ข้อมูลเริ่มแรกเป็นค่าเริ่มต้นการทำงาน แล้วสามารถอ่านข้อมูลถัดไปได้เรื่อยๆ โดยตลอด.

เมื่อ Data preparing อ่านข้อมูลขึ้นมาจากบัฟเฟอร์ และนำไปคำนวณค่าทางสถิติเพิ่มเข้าไป แล้วจึงส่งข้อมูลทั้งสองส่วนลงดิสก์ ดังรูป



รูปที่ 3.4 ข้อมูลถูกส่งผ่านระหว่าง thread 2-3

แล้วจึงเข้าสู่การทำงานของ thread ที่ 3 ซึ่งเป็นการทำงานของตัวตรวจจับการบุกรุก จะอธิบายในส่วน 3.3 การวิเคราะห์ข้อมูล

3.3 การจัดเก็บข้อมูล

ส่วนแรกสุดของโปรแกรมจัดเก็บแพ็กเก็ตที่จับได้ ลงในหน่วยความจำหลักของระบบ เป็นส่วนหนึ่งของกระบวนการ consumer-producer ในขั้นต้น ซึ่งเก็บลงในลิงก์ลิสต์เพื่อแยกการทำงานออกจากกัน ทำให้สนับสนุนการทำงานแบบเวลาจริง ในขณะเดียวกันเมื่อส่วน producer อ่านข้อมูลนั้นจากหน่วยความจำแล้ว ก็จะคืนหน่วยความจำในส่วนนั้นกลับคืนสู่ระบบ ซึ่งต้องการความเร็วของการทำงานบ้างพอสมควร มีโครงสร้างข้อมูลที่เกี่ยวข้องคือ

```
struct thread_arg {
      int   s;                ;เก็บซ็อคเก็ตนัมเบอร์
      char  *buff;            ;เก็บแพ็กเก็ต
      int   len;              ;เก็บความยาวแพ็กเก็ต
      struct thread_arg *next;      ;ชี้โครงสร้างถัดไป
}
```

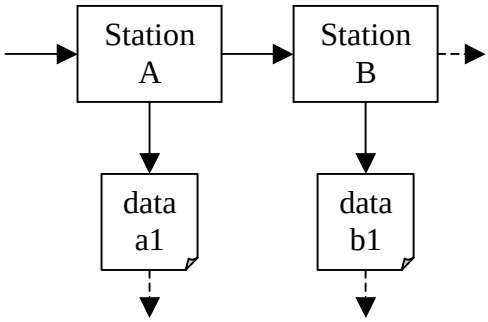
ส่วนแรกใน thread ที่สองก็เช่นกัน เมื่อโปรแกรมรับข้อมูลดิบมาประมวลผล มีการจัดเก็บข้อมูลขั้นต่อไป 2 ส่วนลงหน่วยความจำ(ดิสก์) คือ ข้อมูลดิบที่เลือกเอาแต่เฉพาะฟิลด์บางอย่าง(ในขนาดและจำนวนไม่เกินบัฟเฟอร์คงที่ของโปรแกรม)และข้อมูลที่ได้จากการวิเคราะห์ชุดข้อมูลนั้น จัดเก็บเป็นคู่ลงในดิสก์เมื่อบัฟเฟอร์ดังกล่าวเต็ม เพื่อการวิเคราะห์ต่อไป

โครงสร้างของบัฟเฟอร์คงที่ คือ

```
struct  flag_ip {
```

```
      unsigned  int     m:1;
      unsigned  int     d:1;
      unsigned  int     pad:6;
};
struct  flag_tcp_bit {
      unsigned  int     fin:1;
      unsigned  int     syn:1;
      unsigned  int     rst:1;
      unsigned  int     psh:1;
      unsigned  int     ack:1;
      unsigned  int     urg:1;
      unsigned  int     pad:2;
};
union  flag_tcp {
      u_int8_t          ival;
      struct  flag_tcp_bit    bval;
};
struct str_ip {
      u_int16_t         tot_len;
      u_int16_t         id;
      u_int8_t          proto;
      u_int8_t          s_ip[4];
      u_int8_t          d_ip[4];
};
struct  str_tcp {
      u_int16_t         s_port;
      u_int16_t         d_port;
      u_int32_t         seq;
      u_int32_t         ack
      union  flag_tcp   flag;
      u_int16_t         window;
      u_int16_t         urg_ptr;
};
struct  str_udp {
      u_int16_t         s_port;
      u_nit16_t         d_port;
};
struct  str_icmp {
      u_int8_t          type;
      u_int8_t          code;
};
struct  tcp_pkt {
      long  long  tstamp;
      struct  str_ip    ip;
      struct  str_tcp   tcp;
};
struct  tcp_pkt  buf_tcp[MXCHBUFF];
struct  udp_pkt  buf_udp[MXCHBUFF];
struct  icmp_pkt  buf_icmp[MXCHBUFF];
```

เมื่อจับข้อมูลได้แล้ว จึงเก็บข้อมูลทั้งโครงสร้างลงดิสก์ แล้วจึงเรียกใช้ขึ้นมาอีกต่อหนึ่ง ข้อมูลที่ถูกอ่านขึ้นมาเพื่อการทำงานบนหน่วยความจำ มีลักษณะดังรูป



รูปที่ 3.5 โครงสร้างข้อมูลแบบ link-list ซึ่งใช้ในส่วนหลัง(TCP,UDP,ICMP)

ซึ่งข้อมูลนี้ทั้งหมดมี 3 ชุด คือ โปรโตคอล TCP UDP และ ICMP.

คือ struct tcp_b_send {}, struct udp_b_send {}, struct icmp_b_send {}

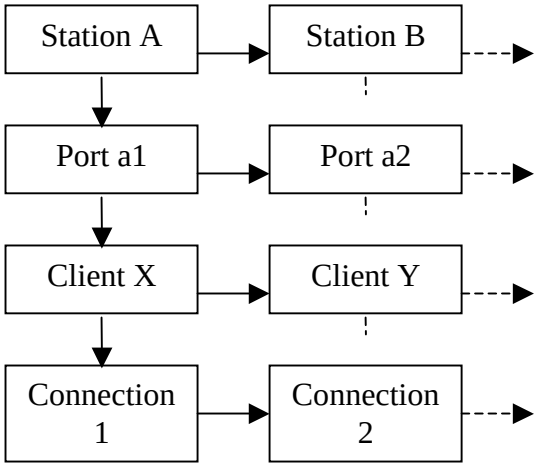
สำหรับโปรโตคอล TCP

```
struct tcp_b_data {
      long  long        tstamp;
      struct  str_ip          ip;
      struct  str_tcp         tcp;
      int               direction;
      struct  tcp_b_data            *next;
};
struct  udp_b_data {
      long  long        tstamp;
      struct  str_ip          ip;
      struct  str_udp         udp;
      struct  udp_b_data            *next;
};
struct  icmp_b_data {
      long  long        tstamp;
      struct  str_ip          ip;
      struct  str_icmp        icmp;
      struct  icmp_b_data     *next;
};

struct  tcp_b_send {
      u_int8_t                ip[4];
      struct  tcp_b_data            *data;
      struct  tcp_tail        *tail;
      struct  tcp_b_send            *next;
};
struct  udp_b_send {
      u_int8_t                ip[4];
      struct  udp_b_data            *data;
      struct  udp_tail        *tail;
      struct  udp_b_send            *next;
};
struct  icmp_b_send {
      u_int8_t                ip[4];
      struct  icmp_b_data           *data;
      struct  icmp_tail             *tail;
      struct  icmp_b_send           *next;
};
```

ส่วนหลังวิเคราะห์การบุกรุก สำหรับการบุกรุกแบบ flooding.

สำหรับโปรโตคอล TCP



รูปที่ 3.6 โครงสร้างการเก็บข้อมูลของส่วนวิเคราะห์ syn-flooding

ประกอบด้วย 4 โครงสร้างข้อมูล คือ

- โครงสร้างแรก คือ stuct Sstation{} เก็บสถานีภายใน
- โครงสร้างที่สอง คือ struct Sportconn{} เก็บพอร์ตต่างๆ ของสถานีภายใน ที่ต้องการสังเกต
- โครงสร้างที่สาม คือ struct Cstation{} เก็บรายละเอียดไคลเอนต์ต่างๆ ที่ขอการติดต่อเข้ามา
- โครงสร้างที่สี่ คือ struct Cconn {} เก็บรายละเอียดสถานะการเชื่อมต่อ ที่เกิดขึ้น

```
struct Sstation {
   u_int8_t                S_ip[4];
   int                found;
   struct  Sportconn       *portconn
   struct  Station         *next;
}
```

เก็บหมายเลขไอพีของสถานีภายใน

```
struct Sportconn {
   u_int16_t               S_port;
   long               sta[5];
   struct  Cstation        *csta;
   struct  Sportconn       *next;
}
```

เก็บหมายเลขพอร์ตต่างๆ สถานะการเชื่อมต่อโดยรวมทั้งพอร์ต

```
struct Cstation {
   u_int8_t                C_ip[4];
   long               sta[5];
   struct  Cconnection           *cconn;
   struct  Cstation        *next;
}
```

เก็บหมายเลขไอพีของไคลเอ็นต์ สถานะการเชื่อมต่อโดยของไคลเอ็นต์

```
struct Cconnection {
   long  long  tstamp;
   u_int16_t         C_port;
   u_int32_t         seq;
```

```
    u_int32_t         ack;
    int         STT_status;
}
```

เก็บหมายเลขพอร์ตต้นทาง สถานะการเชื่อมต่อ ได้แก่ ซีเควนซ์และ แอคนัมเบอร์ และเวลาที่เข้ามา

3.4 การวิเคราะห์ข้อมูล

การวิเคราะห์ในขั้นแรก ระหว่างที่มีการคัดเลือกจับข้อมูล เช่น จำนวนแพ็กเก็ตของบริการต่างๆ ในช่วงเวลา 1 วินาที เวลาที่แพ็กเก็ตนั้นมาถึง ช่วงระยะเวลาที่ห่างกับแพ็กเก็ตที่แล้วของสถานีเดียวกัน ลักษณะซีเควนซ์นัมเบอร์ของแพ็กเก็ต(เฉพาะของแพ็กเก็ตที่บรรจุโปรโตคอลทีซีพี) รวมทั้งค่าเฉลี่ยและการกระจายของข้อมูลดังกล่าว จัดเก็บเป็นคู่กับข้อมูลดิบชุดเดียวกันลงในดิสก์

เวลาที่แพ็กเก็ตมาถึง เพื่อความละเอียดและแม่นยำยิ่งขึ้นจึงเลือกใช้เวลาที่ขนาด usec เพื่อความละเอียดในการคำนวณ และเก็บค่าลงในตัวแปรขนาด long long

การวิเคราะห์แพ็กเก็ตที่จับมาได้ แบ่งตามหมายเลยไอพีของสถานีภายในทั้งหมด และไม่สนใจสถานีอื่นใดที่อยู่นอกขอบเขตนี้ แบ่งต่อไปตามโปรโตคอลหลักที่ใช้ คือ

- TCP (Transmission Control Protocol) เป็นโปรโตคอลที่มีจุดเด่นและจุดอ่อนเดียวกันคือต้องสถาปนาการเชื่อมต่อก่อนที่จะสามารถส่งผ่านข้อมูลถึงกันได้ ทั้งรับประกันการรับส่งข้อมูลด้วย
- UDP(Usr Datagram Protocol) โปรโตคอลให้บริการในการส่งข้อความ โดยไม่คำนึงถึงผู้รับ
- ICMP (Internet Control Message Protocol) โปรโตคอลให้บริการรายงานสาเหตุความผิดพลาดในการส่งข้อมูลระหว่างสถานี

และแต่ละโปรโตคอลดังกล่าวก็แบ่งย่อย ออกเป็นบริการของสถานีภายในต่อไปอีก(port, type) คือ

- TCP port 21(ftp), 22(ssh), 23(telnet) เป็นต้น
- UDP port 7(echo), 13(daytime), 19(chargen) เป็นต้น
- ICMP type 0(echo reply), 8(echo request) เป็นต้น

แต่ละบริการย่อยๆ ทุกบริการที่เฝ้าดู ก็จะหาค่า n/sec, Δt, Δseq ของแต่ละชุดข้อมูลนำมาหาค่าเฉลี่ยและส่วนเบี่ยงเบนที่ได้จากการคำนวนชุดข้อมูลนั้น เพื่อใช้เปรียบเทียบกับค่าที่ได้จากการทดลองกับโปรแกรมเพื่อการบุกรุกที่สามารถหาได้ นำมาหาลักษณะร่วมหรือลักษณะที่ใกล้เคียงกับการบุกรุก

ส่วนวิเคราะห์ส่วนหลังจะจำลองการเชื่อมต่อที่อาจเกิดขึ้นตามกระบวนการเชื่อมต่อของโปรโตคอลทีซีพี ที่เรียกกันว่า 3-way handshake เก็บสถานะต่างๆที่อาจเกิดขึ้นกับสถานีภายใน จำแนกตามพอร์ต แล้วจำแนกต่อไปคตามสถานีต้นทาง เก็บสถานะการเชื่อมต่อที่เป็นไปได้ทั้งหมดและสังเกตการเปลี่ยนสถานะของการเชื่อมต่อที่พบ ซึ่งหากพบการเชื่อมต่อที่ไม่สำเร็จมากผิดปกติ ก็จะเริ่มกระบวนการตอบโต้ต่อไป.

การโจมตีที่ทำการวิเคราะห์มีดังนี้

TCP syn-flooding ผู้ที่ถูกโจมตีลักษณะนี้ จะได้รับการร้องขอการเชื่อมต่อจำนวนมาก และการเชื่อมต่อพวกนี้ไม่มีวันสมบูรณ์ได้(half-open connection) จากจุดประสงค์ของการบุกรุก ที่ต้องการโจมตีตาราง backlog ซึ่งเก็บการเชื่อมต่อที่ยังไม่สำเร็จของบริการเครือข่ายที่เปิดไว้ในสถานีหนึ่งๆ โดยการเชื่อมต่อในตาราง backlog นี้มีจำนวนจำกัด ทำให้โจมตีบริการของระบบได้ โดยการทำให้ตารางดังกล่าวนี้เต็ม(ค่าดีฟอลต์ของระบบปฏิบัติการลินุกซ์ คือ 129 การเชื่อมต่อ) ทำให้การเชื่อมต่อใหม่ๆ ไม่สามารถเข้ามาได้.

UDP flooding บนโปรโตคอล UDP มีบริการหลายอย่างที่มีการส่งข้อความกลับไปยังผู้ส่ง จุดประสงค์เพื่อการตรวจสอบระบบ การโจมตีแบบนี้ใช้ประโยชน์จากบริการดังกล่าว ปลอมแอดเดรสผู้ส่งเป็นผู้ที่ต้องการโจมตี(เหยื่อ) ประกอบกับการส่งแพ็กเก็ตออกไปยังเครือข่ายโดยใช้แอดเดรสแบบบรอดคาสต์ ทำให้เครื่องที่ได้รับแพ็กเก็ตนี้ ส่งข้อความกลับไปยังเหยื่อ เหยื่อจะได้รับข้อความจำนวนมหาศาลเข้ามายังพอร์ตที่เป็นเป้าหมายการบุกรุก.

ICMP flooding หรือ Ping flooding โดยหลักการแล้วเหมือนกับ UDP flooding แต่ส่งแพ็กเก็ต ICMP แทน. โดย ICMP มีรูปแบบบริการที่ส่งข้อความกลับเช่นกัน คือ echo request(type 8) และผู้ได้รับจะตอบด้วย echo reply(type 0).

TCP จะวิเคราะห์โดยนำแพ็กเก็ตทั้งหมดไปจัดประเภทข้างต้น จำลองแบ็กล็อดของเครื่องเซิร์ฟเวอร์ภายใน สังเกตหาค่าที่มากเกินไปของการเชื่อมต่อที่ไม่สำเร็จ(การเชื่อมต่อที่ถูกรีเซต หรืออยู่ในสถานะ SYNRECV มากเกินปกติ – ในโปรแกรมใช้ค่า 20).

UDP และ ICMP โดยนำไปจัดประเภทเช่นกัน แต่หาการใช้โปรโตคอลที่มีการส่งข้อมูลกลับอย่างมากเกินไป ในช่วงเวลาสั้นๆ (ในโปรแกรมใช้ค่า 50 แพ็กเก็ต/วินาที).

อย่างไรก็ตาม อาจมีการปลอมแปลงผู้ส่งร่วมด้วย ทำให้ไม่สามารถระบุผู้บุกรุกที่แท้จริงได้

ผลการวิเคราะห์ที่ได้ นำไปสร้างกฎสกัดกั้นการเชื่อมต่อข้อมูล โดยส่งชุดคำสั่งไปยังส่วนควบคุมเกี่ยวกับไฟร์วอลล์ เพื่อป้องกันการบุกรุกไม่ให้เข้ามาอีก.

3.5 มาตรการตอบโต้

การทำงานในส่วนการตอบโต้ ได้รับการสนับสนุนจาก โครงงานการตรวจจับการบุกรุกโดยใช้ไฟร์วอลล์. จำแนกประเภทได้ตามชนิดของโปรโตคอลที่พบการบุกรุก.

ทีซีพี – เมื่อตรวจพบว่ามีความน่าจะเป็นที่จะถูกโจมตีจาก syn-flooding ซึ่งเข้ามายังพอร์ตเฉพาะของสถานีหนึ่งๆ จะตอบโต้โดยกั้นการเชื่อมต่อที่จะเกิดขึ้นอีก โดยวิเคราะห์แอดเดรสผู้ส่ง แล้วป้องกันเฉพาะการเชื่อมต่อกับสถานีภายในจากผู้ส่งนั้น. ถ้าหากมีการปลอมแอดเดรสผู้ส่งร่วมด้วยก็จะป้องกันการเชื่อมต่อโดยใช้ แต่แอดเดรสของสถานีภายใน.

ยูดีพี และ ไอซีเอ็มพี - โดยปกติแล้วแพ็กเก็ตที่ใช้ 2 โปรโตคอลนี้ ไม่พบมากนัก สังเกตได้ง่ายหากมีจำนวนมากมาย ซึ่งชี้ว่ามีการโจมตีเกิดขึ้น. โดยการตอบโต้ก็เป็นในลักษณะเดียวกัน.

สร้างกฎเกี่ยวกับ ip-chain ส่งคำสั่งไปยังส่วนควบคุมไฟร์วอลล์ให้ปิดกั้นการเชื่อมต่อที่สงสัยไม่ให้เข้ามาได้อีก.

3.6 ปัญหาและการแก้ไข

โครงสร้างการทำงานเดิมเป็นแบบเส้นตรง วิเคราะห์แพ็กเก็ตทันที ไม่สามารถจับข้อมูลได้ทัน จึงแก้ไขโดยเปลี่ยนโครงสร้างการทำงานเป็น thread แยกการทำงานเป็น 3 ส่วน คือ ส่วนจับแพ็กเก็ต ส่วนจัดการข้อมูล และส่วนวิเคราะห์. การวิเคราะห์ตอนแรกเป็นแบบ off-line ปรับปรุงให้เป็น on-line เพื่อให้ใกล้เคียงกับการทำงานจริง.

บทที่ 4
บทสรุป และแนวทางการพัฒนา

4.1 การประยุกต์ใช้งาน

ในส่วนของโครงงาน มีส่วนการทดลองเก็บค่า ในไฟล์ชั่วคราวในหลายส่วนการทำงาน อยู่ในไดเรคทอรี tmp_file ซึ่งผู้ใช้สามารถเข้าไปดูรายละเอียดของค่าต่างๆ ที่เกี่ยวของกับการโจมตีได้.

4.2 ปัญหาและข้อบกพร่อง

ปัญหาและข้อพกพร่องที่พบในการดำเนินงานมีดังนี้

เกี่ยวกับการทำงาน

- ในการวิเคราะห์ต้องการโปรเซสเซอร์ที่มีความเร็ว
- ต้องการฮาร์ดดิสก์มาก(ขึ้นกับความเร็วโปรเซสเซอร์)
- ต้องการหน่วยความจำมาก

เกี่ยวกับการดำเนินงาน

- การทำงานเป็นทีมเข้ากันได้ไม่ดีเท่าที่ควร
- โครงสร้างข้อมูลออกแบบมายังไม่ยืดหยุ่น
- ยังขาดส่วนของไฟล์คอนฟิกกูเรชันของการทำงาน

4.3 แนวทางในการพัฒนา

โครงงานนี้ใช้เป็นฐานการพัฒนาโปรแกรมรักษาความปลอดภัยในเครือข่าย โดยจัดหาข้อมูลเพื่อการวิเคราะห์การบุกรุกแบบแพ็กเก็ตเดี่ยว และดัดแปลงเพื่อเพิ่มความสามารถในการหาค่าที่สำคัญต่างๆ เพิ่มขึ้นเพื่อวิเคราะห์การบุกรุกแบบหลายแพ็กเก็ตในขอบเขตของโครงงาน เพื่อความสมบูรณ์ในการตรวจจับการบุกรุกจากผู้ไม่ประสงค์ดีที่เข้ามาโจมตีระบบในรูปแบบต่างๆ ได้

4.4 กิตติกรรมประกาศ

ข้าพเจ้าขอขอบพระคุณ ผู้ให้การสนับสนุนและช่วยเหลือข้าพเจ้าให้ทำงานได้สำเร็จลุล่วง ขอขอบพระคุณอาจารย์สุรศักดิ์ สงวนพงษ์ ที่ปรึกษาโครงงานของข้าพเจ้า และสำหรับคำแนะนำของ คุณ อุรุพจน์ กัลยาสิริ สำหรับข้อมูลและแนวคิด และขอขอบคุณนายนฤชิต ชินะวงศ์ ที่ให้ความร่วมมือกับโครงงานนี้ในส่วนของการวิเคราะห์การบุกรุกในแบบต่างๆ และส่วนของการป้องกันการบุกรุกแบบไฟร์วอลล์ มา ณ ที่นี้

เอกสารอ้างอิง

สุรศักดิ์ สงวนพงษ์, สถาปัตยกรรมและโปรโตคอลทีซีพี/ไอพี, สำนักพิมพ์ซีเอ็ด 2543.

Herve Debar, Marc dacier, Andreas Wespi – Towards a taxonomy of intrusion-detection systems., 1999.