บทที่ 3

ระบบตรวจจับการบุกรุก

3.1 บทนำ

จากปัญหาภัยคุกคามและการโจมตีในรูปแบบต่าง ๆ ที่ส่งผลกระทบต่อความปลอดภัยของระบบคอมพิวเตอร์และเครือข่ายตามที่ได้กล่าวไว้ในบทที่ 2 ดังนั้นจำเป็นที่จะต้องมีกระบวนการในการตรวจจับการโจมตีหรือการบุกรุก เพื่อที่ได้รับทราบและทำการแก้ไขปัญหาดังกล่าวได้อย่างทันท่วงที ในบทนี้จะกล่าวถึงรายละเอียดเกี่ยวกับระบบตรวจจับการบุกรุก (Intrusion Detection System) ซึ่งเรียกโดยย่อว่า IDS การแบ่งประเภทของ IDS แหล่งที่มาของข้อมูลที่ใช้ในการวิเคราะห์เพื่อตรวจจับการบุกรุกแต่ละรูปแบบ จุดเด่นและจุดด้อยของ IDS ในแต่ละประเภท แนวทางและวิธีการต่าง ๆที่ใช้ในการตรวจจับการบุกรุกและคุณลักษณะของระบบตรวจจับการบุกรุกที่ดี

3.2 ระบบตรวจจับการบุกรุก

SANS (SysAdmin, Audit, Network, Security) Institute (www.sans.org) ซึ่งเป็นองค์กรที่เป็นแหล่งข้อมูล การฝึกอบรม การสอบประกาศนียบัตรด้านการรักษาความปลอดภัย ได้ให้ความหมายของการตรวจจับการบุกรุกคือ เทคนิคในการตรวจจับการบุกรุกเข้าสู่เครื่องคอมพิวเตอร์หรือเครือข่ายโดยการตรวจสอบข้อมูลในบันทึกความปลอดภัย (security logs) หรือข้อมูลบันทึกการใช้งาน (audit log) สำหรับกระบวนวิเคราะห์ข้อมูลเพื่อตรวจจับการบุกรุกหรือความพยายามที่จะบุกรุกอาจจะทำโดยผู้ดูแลระบบหรือใช้โปรแกรมช่วยในการวิเคราะห์ก็ได้ [SANS Institute, 2001]

Rebecca Bace และ Peter Mell [Bace and Mell, 2001] ได้ให้นิยามระบบตรวจจับการบุกรุก คือ ระบบที่ประกอบด้วยฮาร์ดแวร์หรือซอฟต์แวร์สำหรับทำงานในกระบวนการตรวจสอบเหตุการณ์ต่าง ๆที่เกิดขึ้นในระบบคอมพิวเตอร์และเครือข่ายเพื่อวิเคราะห์หาร่องรอยของการบุกรุกโดยอัตโนมัติ และให้นิยามการบุกรุก คือ ความพยายามหรือการกระทำที่ส่งผลต่อความบูรณภาพ (Integrity) ความลับ (Confidentiality) และ ความพร้อมใช้งาน (Availability) ของทรัพยากรในระบบหรือการกระทำเพื่อข้ามผ่านมาตรการในการควบคุมความปลอดภัยของระบบคอมพิวเตอร์และเครือข่าย

3.3 ความจำเป็นที่จะต้องใช้ระบบตรวจจับการบุกรุก

เมื่อคำนึงถึงเรื่องความปลอดภัยของคอมพิวเตอร์มักเป็นการยากในการมองภาพที่ชัดเจนว่า อะไรที่จะบ่งบอกได้ว่าการใช้งานคอมพิวเตอร์มีความปลอดภัย เนื่องจากความปลอดภัยของคอมพิวเตอร์เป็นสิ่งที่มองเห็นยากและยากต่อการวัด แต่อย่างไรก็ตามเราสามารถเปรียบเทียบความปลอดภัยของคอมพิวเตอร์กับการรักษาความปลอดภัยสถานที่ ในการรักษาความปลอดภัยของสถานที่นั้น นอกจากการจัดบริเวณที่ต้องรักษาความปลอดภัยให้มีรั้วรอบขอบชิด มีกุญแจที่ใช้ล็อกประตูหรือทางเข้าออก สิ่งหนึ่งที่จะขาดไม่ได้คือการจัดให้มีบุคคลหรืออุปกรณ์ที่คอยตรวจสอบการละเมิดต่ออุปกรณ์หรือเครื่องกีดขวางที่จัดตั้งเพื่อความปลอดภัย ทั้งนี้เนื่องจากอาจมีผู้ไม่หวังดีพยายามบุกรุกโดยทำลายอุปกรณ์หรือเครื่องกีดขวางดังกล่าว ดังนั้นเราจึงต้องอาศัยระบบที่ใช้ตรวจสอบเมื่อมีการทำลายหรือล่วงล้ำต่ออุปกรณ์หรือเครื่องกีดขวางที่ได้ติดตั้งไว้อีกชั้นหนึ่ง ตัวอย่างอุปกรณ์ที่ใช้ตรวจสอบเช่น ระบบสัญญาณเตือนขโมยที่ใช้ควบคู่กับรั้วที่แข็งแรง

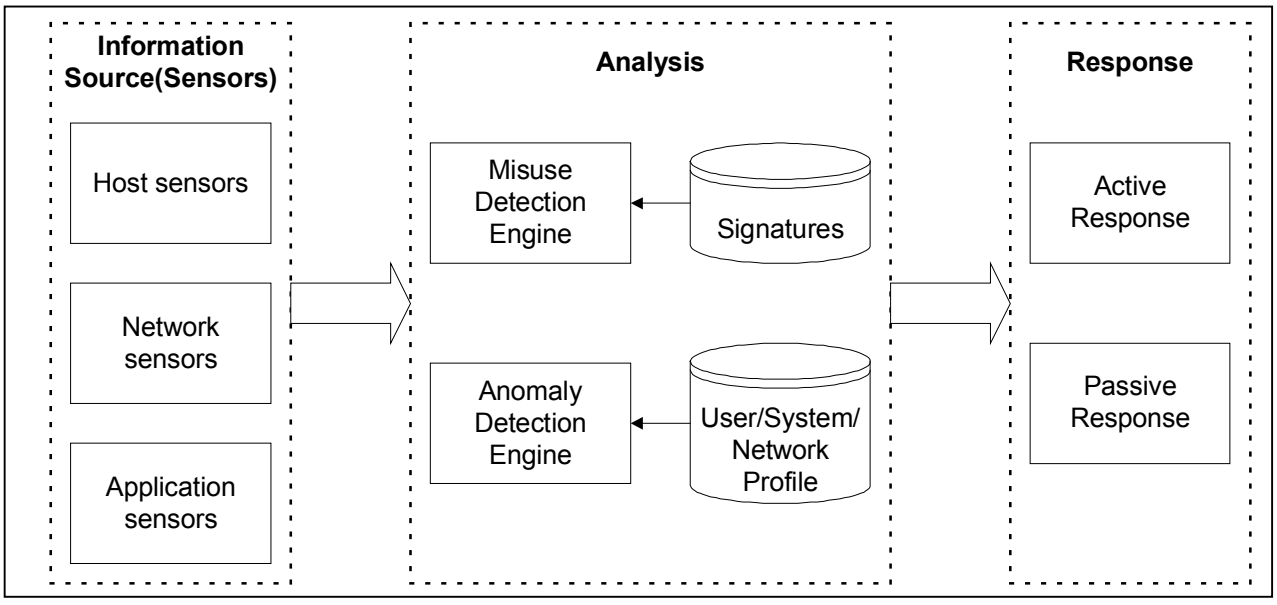
ระบบเครือข่ายคอมพิวเตอร์ก็เช่นเดียวกัน บุคคลทั่วไปมักคิดว่า การมีกลไกในการควบคุมการใช้งานและการติดตั้งไฟร์วอลล์ (Firewall) ก็สามารถทำให้เครือข่ายคอมพิวเตอร์มีความปลอดภัย แต่อย่างไรก็ตาม การติดตั้งไฟร์วอลล์ให้กับระบบเครือข่ายคอมพิวเตอร์ก็เปรียบเสมือนการสร้างรั้วหรือกำแพงเพื่อตรวจสอบบุคคลที่จะเข้ามาในสถานที่ที่จะรักษาความปลอดภัย แต่หากมีบุคคลไม่หวังดีสามารถปีนรั้วเข้ามาได้ การรักษาความปลอดภัยโดยใช้รั้วก็หมดความหมาย ดังนั้นในการเพิ่มความปลอดภัยอีกประการหนึ่งคือการใช้ระบบตรวจจับการบุกรุกซึ่งมีคุณลักษณะที่กล่าวมาในตอนต้น

3.4 องค์ประกอบสำหรับระบบตรวจจับการบุกรุก

ถึงแม้ว่าปัจจุบันระบบตรวจจับการบุกรุกมีหลากหลายประเภท แต่ละประเภทมีลักษณะการทำงานและแนวทางในการวิเคราะห์เพื่อตรวจจับการบุกรุกที่แตกต่างกัน แต่ระบบตรวจจับการบุกรุกส่วนใหญ่จะมีองค์ประกอบและกระบวนการในการทำงานทั่วไปเหมือนกันคือประกอบด้วย 3 องค์ประกอบพื้นฐานสามส่วนดังนี้ [Bace and Mell, 2001]

1. Information Source (Sensor) ข้อมูลเหตุการณ์และข้อมูลการทำงานจากแหล่งข้อมูลต่าง ๆ จะถูกนำใช้ในการวิเคราะห์เพื่อตัดสินว่าเมื่อใดที่มีการบุกรุกเกิดขึ้น โดยแหล่งข้อมูลเหล่านี้จะนำมาจากข้อมูลในระดับต่าง ๆของระบบเช่น ข้อมูลในระดับเครือข่าย ระดับเครื่องคอมพิวเตอร์ และโปรแกรมประยุกต์ที่มีการใช้งานอยู่ในระบบ

2. Analysis เป็นส่วนที่ทำหน้าที่ในการจัดการและวิเคราะห์ข้อมูลที่ได้รับจากแหล่งข้อมูลต่าง ๆ แล้วตัดสินว่าเหตุการณ์หรือการกระทำใดที่บ่งชี้ว่ากำลังมีการบุกรุกเกิดขึ้นหรือได้มีการบุกรุกเกิดขึ้นแล้วในระบบ โดยแนวทางที่ใช้ในการวิเคราะห์มีสองรูปแบบคือ Anomaly Detection และ Misuse Detection ซึ่งจะได้กล่าวต่อไปในหัวข้อที่ 3.6
3. Response เป็นชุดของการกระทำเมื่อระบบตรวจจับการบุกรุกตรวจจับได้ว่ามีการบุกรุกเกิดขึ้น โดยการกระทำเหล่านี้สามารถจัดกลุ่มได้เป็นการกระทำแบบ active และ passive โดยที่การกระทำแบบ active จะก่อให้มีการกระทำอื่น ๆที่จะตอบสนองต่อเหตุการณ์การบุกรุกนั้นโดยอัตโนมัติ เช่น การปรับค่าของเราเตอร์หรือไฟร์วอลล์ให้ปิดกั้นการเชื่อมต่อที่ส่งมาจากผู้บุกรุก ปิดกั้นพอร์ตหรือโปรโตคอลที่ถูกใช้ผู้บุกรุก เป็นต้น ส่วนการกระทำแบบ passive จะเป็นการรายงานหรือแจ้งเตือนการบุกรุกไปยังบุคคลที่รับผิดชอบเพื่อแก้ไขปัญหา โดยอาศัยข้อมูลที่ได้รับรายงาน



ภาพประกอบ 3.1 องค์ประกอบระบบตรวจจับการบุกรุก

3.5 ประเภทของระบบตรวจจับการบุกรุก

ในการจัดประเภทของระบบตรวจจับการบุกรุกสามารถแบ่งได้หลายรูปแบบ โดยใช้หลักเกณฑ์ต่าง ๆ เช่น แหล่งข้อมูลที่นำมาวิเคราะห์ แนวทางในการตรวจจับการบุกรุก ช่วงระยะ

เวลาที่ทำการวิเคราะห์การบุกรุกหลังจากเกิดเหตุการณ์ขึ้น เป็นต้น ซึ่งส่วนใหญ่จะนิยมจัดประเภทโดยใช้แหล่งข้อมูล (Information Source) ที่นำมาวิเคราะห์ในการจัดประเภท ซึ่งสามารถแบ่งได้เป็น 3 ประเภท คือ Network-Based Intrusion Detection System (NIDS) Host-Based Intrusion Detection System (HIDS) และ Application-Based Intrusion Detection System (HIDS) โดยระบบตรวจจับการบุกรุกประเภทแรกจะวิเคราะห์แพ็กเก็ตที่รับส่งกันในเครือข่ายโดยการดักจับข้อมูลแพ็กเก็ตบนเครือข่ายหลักหรือเครือข่ายในแต่ละส่วนเพื่อตรวจจับการบุกรุก ขณะที่ระบบตรวจจับประเภทที่สองวิเคราะห์ข้อมูลบันทึกการทำงาน (Audit Trail) ที่ได้จากระบบปฏิบัติการ ส่วนระบบตรวจจับประเภทสุดท้ายจะวิเคราะห์ข้อมูลบันทึกการทำงานของระบบโปรแกรมประยุกต์ต่างๆ ที่ทำงานอยู่บนเครื่องนั้น เพื่อหาร่องรอยของการบุกรุก

3.5.1 Network-Based Intrusion Detection System (NIDS)

เป็นระบบตรวจจับการบุกรุกที่ติดตามและวิเคราะห์แพ็กเก็ตที่รับส่งกันในเครือข่ายเพื่อดูว่ามีผู้บุกรุกหรือความผิดปกติเกิดขึ้นหรือไม่ โดยจะดักจับข้อมูลบน Network Segment หรืออุปกรณ์สวิตซ์ ดังนั้น NIDS หนึ่งระบบก็จะสามารถติดตามข้อมูลบนเครือข่ายซึ่งจะมีผลกับหลายๆ เครื่องคอมพิวเตอร์ที่เชื่อมต่ออยู่ในเครือข่ายเดียวกันและตรวจจับการบุกรุกที่จะเกิดขึ้นกับเครื่องเหล่านั้นได้ด้วย ทั้งนี้แพ็กเก็ตจะถูกตรวจจับโดยตัวดักจับ (Sensor) และวิเคราะห์ว่าเข้ากับรูปแบบหรือร่องรอยการบุกรุก (Signature) ที่กำหนดไว้ในฐานข้อมูลการบุกรุกของ NIDS หรือไม่ เช่น ระบบเฝ้าตรวจสอบการร้องขอการเชื่อมต่อบน TCP ที่พยายามจะเชื่อมต่อมายังพอร์ตต่างๆ ของเครื่องเป้าหมาย โดย NIDS นั้นอาจจะถูกติดตั้งบนเครื่องเป้าหมายเองและคอยตรวจทุกแพ็กเก็ตของตัวเองหรืออาจจะถูกติดตั้งบนเครื่องที่แยกอยู่ต่างหากเพื่อคอยตรวจจับทุกแพ็กเก็ตในเครือข่ายที่เครื่องเป้าหมายอยู่ก็ได้

จุดเด่นของ NIDS มีดังนี้

- มีค่าใช้จ่ายต่ำ เนื่องจากการติดตั้งสามารถติดตั้งระบบได้ในจุดเข้าถึง (Access Point) ของแต่ละเครือข่ายที่ต้องการได้ ทำให้สามารถตรวจสอบข้อมูลของทุกเครื่องที่อยู่บนเครือข่าย โดยไม่ต้องต้องติดตั้งโปรแกรมเพิ่มเติมลงในแต่ละเครื่องของเครือข่ายนั้นๆ ซึ่งทำให้ลดค่าใช้จ่ายในการติดตั้งและดูแลระบบตรวจจับแบบนี้
- สามารถตรวจจับการบุกรุกบางอย่างที่ HIDS ทำไม่ได้ เนื่องจาก NIDS จะตรวจสอบแพ็กเก็ตทั้งหมดเพื่อวิเคราะห์หาร่องรอยของการกระทำที่มุ่งประสงค์ร้าย แต่ HIDS ไม่ได้ตรวจดูในส่วนของแพ็กเก็ต ดังนั้นจึงไม่สามารถตรวจจับการบุกรุกที่เป็นการโจมตีผ่านเครือข่ายได้ เช่น การโจมตี

แบบ DoS, Land Attack เป็นต้น ซึ่งจะสามารถระบุได้โดยการตรวจดูแพ็กเก็ตที่รับส่งบนเครือข่ายเท่านั้น

- ติดตั้งได้ง่าย ไม่มีผลกระทบกับระบบและโครงสร้างเดิมของระบบ เนื่องจากการทำงานของ NIDS ไม่ขึ้นกับระบบปฏิบัติการของแต่ละเครื่องในเครือข่าย และการดักจับข้อมูลที่รับส่งในเครือข่ายจะกระทำในลักษณะ passive ซึ่งจะไม่รบกวนการทำงานปกติของเครือข่าย
- ระบบ NIDS สามารถทำให้มีความปลอดภัยต่อการโจมตีได้ดีกว่า ส่วนใหญ่ระบบ NIDS จะถูกออกแบบให้ทำงานในลักษณะ Stealth Mode ซึ่งทำให้ผู้โจมตีไม่สามารถทราบได้ว่ามีระบบ NIDS หรือไม่และติดตั้งอยู่ที่ตำแหน่งใด ทำให้เป็นการยากที่ผู้โจมตีจะทำลายหลักฐานหรือร่องรอยต่างๆ ที่เกิดขึ้นจากการบุกรุกหรือโจมตีได้
- สามารถตรวจจับการบุกรุกได้อย่างทันทีทันใด (Real-Time Detection) ดังนั้นทำให้การตอบสนองต่อการโจมตีสามารถทำได้อย่างรวดเร็ว
- สามารถตรวจจับความพยายามบุกรุกที่ไม่สำเร็จได้ ในกรณีที่ติดตั้งระบบ NIDS นอกระบบป้องกันไฟร์วอลล์ ทำให้สามารถตรวจจับการโจมตีทรัพยากรต่างๆ ที่อยู่หลังไฟร์วอลล์ได้ ถึงแม้ว่าการโจมตีเหล่านั้นอาจจะถูกป้องกันไว้โดยไฟร์วอลล์แล้วก็ตาม

จุดอ่อนของ NIDS มีดังนี้

- ในสภาวะที่มีการจราจรคับคั่งบนเครือข่ายขนาดใหญ่ จำนวนแพ็กเก็ตที่รับส่งผ่านเครือข่ายมีจำนวนมาก ทำให้การดักจับและวิเคราะห์ทุกแพ็กเก็ตนั้นทำได้ค่อนข้างยาก ทำให้บางครั้งไม่สามารถตรวจจับการบุกรุกที่เกิดขึ้นในช่วงการจราจรคับคั่งได้ แต่ผู้ผลิตระบบ NIDS บางรายได้จัดสร้างระบบนี้ในลักษณะของฮาร์ดแวร์เพื่อให้มีการทำงานที่เร็วขึ้นและพัฒนาวิธีการตรวจจับที่ใช้ทรัพยากรน้อยลง เพื่อให้สามารถรองรับปริมาณการจราจรบนเครือข่ายขนาดใหญ่ได้
- การจัดวางตำแหน่งของระบบ NIDS อาจจะทำได้ยาก โดยเฉพาะเมื่อต่อระบบ NIDS กับอุปกรณ์สวิตช์รุ่นใหม่ที่มีความสามารถในการแบ่งเครือข่ายออกเป็น Network Segment เล็กหลายๆ ส่วน ซึ่งมีผลทำให้ระบบ NIDS ไม่สามารถดักจับและตรวจสอบข้อมูลทั้งหมดที่ผ่านอุปกรณ์สวิตช์
- ระบบ NIDS ไม่สามารถวิเคราะห์ข้อมูลที่ถูกเข้ารหัส ซึ่งเป็นปัญหาที่กำลังเพิ่มมากขึ้นขององค์กรที่มีการใช้ Virtual Private Networks (VPN)

- ระบบ NIDS ไม่สามารถบอกได้ว่าการบุกรุกนั้นสำเร็จหรือไม่ เนื่องจากสามารถตรวจจับได้เฉพาะช่วงเริ่มต้นของการบุกรุกเท่านั้น นั่นคือหลังจาก NIDS ตรวจจับได้ว่ามีการบุกรุกเกิดขึ้น ระบบสามารถทำได้แค่การส่งสัญญาณเตือนให้ผู้ดูแลระบบทราบถึงการบุกรุก ผู้ดูแลระบบจะต้องตรวจสอบดูด้วยตนเองหรือใช้วิธีการอื่นๆ ประกอบในการตรวจสอบว่าการบุกรุกนั้นสำเร็จหรือไม่

3.5.2 Host-Based Intrusion Detection System (HIDS)

เป็นระบบตรวจจับการบุกรุกที่รวบรวมข้อมูลจากแต่ละเครื่องคอมพิวเตอร์ เพื่อตรวจสอบว่าโปรแกรมหรือผู้ใช้คนใดที่ทำให้เกิดการบุกรุกขึ้นในระบบ และผลของการบุกรุกเป็นอย่างไรระบบ HIDS ส่วนใหญ่จะเก็บรวบรวมข้อมูลจากบันทึกการทำงานของระบบปฏิบัติการแล้วนำข้อมูลเหล่านั้นมาวิเคราะห์ด้วยวิธีการและเทคนิคต่างๆ เพื่อค้นหาเหตุการณ์ผิดปกติหรือการบุกรุกที่เกิดขึ้น

จุดเด่นของ HIDS มีดังนี้

- สามารถตรวจสอบได้ว่าการบุกรุกที่เกิดขึ้นนั้นสำเร็จหรือไม่ อย่างไร เพราะว่า HIDS จะวิเคราะห์ของจากล็อกไฟล์ของระบบจึงทำให้ทราบถึงเหตุการณ์และกิจกรรมต่างๆ ที่เกิดขึ้นในระบบอย่างละเอียด
- สามารถตรวจจับการบุกรุกที่เกิดขึ้นบนเครื่องคอมพิวเตอร์นั้นๆ โดยตรง (Local Attack) โดยไม่ผ่านเครือข่าย ซึ่ง NIDS ไม่สามารถตรวจจับได้ เช่น ผู้บุกรุกเข้ามาที่เครื่องคอมพิวเตอร์และใช้คีย์บอร์ดที่ต่ออยู่กับเครื่องคอมพิวเตอร์ ในการบุกรุก ดังนั้นจึงไม่มีข้อมูลบุกรุกส่งผ่านเครือข่ายเป็นต้น
- HIDS ไม่ได้รับผลกระทบในกรณีที่มีการเข้ารหัสข้อมูลในการรับส่งข้อมูลบนเครือข่าย เนื่องจากข้อมูลเหล่านี้จะถูกสร้างขึ้นแล้วเข้ารหัสก่อนที่ส่งออกไปในเครือข่ายและเมื่อถึงเครื่องปลายทางก็จะมีการถอดรหัสข้อมูลก่อนใช้งาน นอกจากนี้ HIDS ก็ไม่ได้รับผลกระทบในกรณีที่มีการใช้อุปกรณ์สวิตซ์ในเครือข่าย ดังนั้น HIDS จึงเหมาะสำหรับเครือข่ายที่เชื่อมต่อกันโดยใช้อุปกรณ์ สวิตซ์และมีการเข้ารหัสข้อมูลที่รับส่งในเครือข่าย
- สามารถช่วยในการตรวจจับโปรแกรมประเภทม้าโทรจัน หรือการบุกรุกอื่นๆ ที่ผลกับความสมบูรณ์ (Integrity) ของโปรแกรมและข้อมูลต่างๆ ของระบบคอมพิวเตอร์ได้

- การตรวจจับการบุกรุกและการตอบสนองต่อการบุกรุก สามารถทำได้เกือบทันทีทันใด ถึงแม้ว่าระบบ HIDS ไม่สามารถตอบสนองต่อการบุกรุกได้ในทันทีทันใดเหมือนกับ NIDS แต่หากมีการติดตั้งและการกำหนดค่าต่าง ๆ ในการติดตั้งระบบที่เหมาะสมแล้ว ก็จะทำให้การตอบสนองหลังเกิดการบุกรุกทำได้อย่างรวดเร็วและแม่นยำขึ้น ซึ่งจะทำให้มีความเสียหายไม่มากนัก
- ไม่ต้องมีอุปกรณ์เพิ่มเติมสำหรับการติดตั้งระบบ HIDS ในระบบและค่าใช้จ่ายสำหรับการติดตั้งระบบ HIDS จะต่ำกว่า NIDS

จุดอ่อนของ HIDS มีดังนี้

- การจัดการและดูแล HIDS ทำได้ยากกว่า เนื่องจากต้องมีการกำหนดค่าที่เหมาะสมและจัดการทุกเครื่องคอมพิวเตอร์ที่ติดตั้งระบบ HIDS
- เนื่องจากข้อมูลและระบบ HIDS นั้นเก็บอยู่บนเครื่องเป้าหมาย ดังนั้น HIDS ก็อาจจะถูกโจมตีจนไม่สามารถทำงานได้ รวมถึงถ้าหากการบุกรุกสำเร็จผู้บุกรุกอาจจะลบข้อมูลกิจกรรม เหตุการณ์ต่าง ๆ ที่เก็บในล็อกไฟล์ออกได้
- ไม่สามารถตรวจจับการโจมตีที่เป็นลักษณะสำรวจเครือข่าย (Network Scan) ของทั้งเครือข่ายได้ เนื่อง HIDS จะมองเห็นเฉพาะข้อมูลเครือข่ายที่เป็นของตนเองเท่านั้น
- เนื่องจากระบบ HIDS ใช้ข้อมูลในล็อกไฟล์ของระบบมาใช้ในการวิเคราะห์ ซึ่งข้อมูลมีปริมาณมากดังนั้นจึงต้องมีพื้นที่สำหรับเก็บข้อมูลเพิ่มเติมในระบบคอมพิวเตอร์
- ระบบ HIDS จะต้องใช้ทรัพยากรของเครื่องคอมพิวเตอร์ที่ได้รับการติดตั้งอยู่เพื่อทำการวิเคราะห์การบุกรุก ดังนั้นก็จะมีผลกระทบต่อประสิทธิภาพการทำงานของเครื่องนั้นด้วย

3.5.3 Application-Based Intrusion Detection System (subset of Host-based IDS)

ระบบตรวจจับการบุกรุกประเภทนี้จะมีการทำงานคล้ายกับ HIDS แต่จะรวบรวมข้อมูลจากการทำงานของโปรแกรมประยุกต์ที่ทำงานบนเครื่องคอมพิวเตอร์ มาใช้ในการวิเคราะห์เพื่อตรวจสอบว่ามีพฤติกรรมผิดปกติหรือการใช้สิทธิเกินขอบเขตที่กำหนดไว้ของผู้ใช้หรือไม่

จุดเด่นของ Application-Based IDS มีดังนี้

- สามารถติดตามการใช้งานโปรแกรมประยุกต์ของผู้ใช้แต่ละคนได้ว่ามีพฤติกรรมที่ผิดปกติหรือใช้สิทธิเกินขอบเขตที่กำหนดไว้หรือไม่

- สามารถใช้ในสภาพแวดล้อมที่มีการเข้ารหัสข้อมูล เนื่องจากระบบตรวจจับการบุกรุกแบบนี้จะทำงานหลังจากโปรแกรมประยุกต์ได้ทำงานเสร็จแล้ว ซึ่งข้อมูลที่นำมาแสดงหรือใช้งานก็อยู่ในรูปแบบปกติที่ไม่มีการเข้ารหัส

จุดอ่อนของ Application-Based IDS มีดังนี้

- Application-based IDS อาจจะมีช่องโหว่ให้โจมตีมากกว่า HIDS เนื่องจากล็อกไฟล์ของโปรแกรมประยุกต์ไม่ได้รับการป้องกันเป็นอย่างดีเหมือนกับล็อกไฟล์ของระบบที่ใช้สำหรับ HIDS
- Application-based IDS ส่วนใหญ่จะสามารถติดตามการใช้งานในระดับผู้ใช้เท่านั้น ทำให้ไม่สามารถตรวจจับโปรแกรมประเภทม้าโทรจันหรือการโจมตีอื่นที่เกี่ยวกับโปรแกรมประยุกต์โดยตรง ดังนั้นจึงควรใช้งานระบบนี้ร่วมกับ HIDS หรือ NIDS เพื่อให้การทำงานมีประสิทธิผลมากขึ้น

3.6 แนวทางในการตรวจจับการบุกรุก

ปัจจุบันแนวทางในการตรวจจับการบุกรุกมี 2 แนวทาง คือ Anomaly Detection และ Misuse Detection โดยแนวทางแรกเป็นการตรวจจับที่ตั้งอยู่บนพื้นฐานการตรวจหาพฤติกรรมและการใช้ทรัพยากรของระบบคอมพิวเตอร์ที่ผิดไปจากสภาวะการใช้งานปกติ เช่น ตามปกติแล้วผู้ใช้ ก. ใช้คอมพิวเตอร์จากที่ทำงานในระหว่าง 9.00 น. ถึง 17.00 น. เท่านั้น แต่ถ้าหากมีการใช้งานในช่วงกลางคืนก็จะผิดปกติและอาจจะเป็นการบุกรุก เป็นต้น ดังนั้นแนวทางในการตรวจจับวิธีนี้จะพยายามแยกพฤติกรรมการใช้ปกติหรือยอมรับได้ออกมาและให้พฤติกรรมที่ผิดปกติที่เหลือเป็นการบุกรุก[Kumar, 1995] การตรวจจับการบุกรุกด้วยแนวทางนี้เกิดขึ้นในช่วงแรก ๆ ของงานวิจัยที่เกี่ยวกับการตรวจจับการบุกรุก โดยงานวิจัยเรื่อง Computer Security Threat Monitoring and Surveillance ของ Anderson [Anderson, 1980] ถือได้ว่าเป็นจุดเริ่มต้นที่สำคัญชิ้นหนึ่งที่นำไปสู่งานวิจัยอื่น ๆของระบบการตรวจจับการบุกรุกที่ใช้แนวทางนี้ในภายหลัง เช่น IDES, NIDES เป็นต้น

ส่วนอีกแนวทางที่สองคือ Misuse Detection จะตรวจจับโดยอาศัยคุณลักษณะหรือรูปแบบที่ได้กำหนดไว้แล้วว่าเป็นการบุกรุกแล้วนำมาเปรียบเทียบหรือค้นหากับเหตุการณ์ที่เกิดขึ้นในระบบเพื่อหาการบุกรุกโดยตรง ดังนั้นวิธีการนี้จึงต้องอาศัยความรู้เกี่ยวกับการบุกรุกและพฤติกรรมที่ไม่เหมาะสมเป็นอย่างดี เพื่อที่สามารถกำหนดรูปแแบบเพื่อใช้ในการค้นหาการบุกรุกได้อย่างถูกต้องและคลอบคลุมการบุกรุกทั้งหมดที่รู้จัก

สำหรับรายละเอียดของแนวทางการตรวจจับการบุกรุกทั้งสองแบบมีดังนี้

3.6.1 Anomaly Detection

จากที่กล่าวแล้วว่าหลักการของการตรวจจับการบุกรุกด้วยแนวทาง Anomaly Detection ถือว่ากิจกรรมการบุกรุกทั้งหมดเป็นกิจกรรมที่ผิดปกติ ในการตรวจจับการบุกรุกจึงต้องแยกกิจกรรมการทำงานปกติหรือยอมรับได้ออกมาและให้กิจกรรมที่เหลือเป็นกิจกรรมที่ผิดปกติและถือว่าเป็นการบุกรุก ดังนั้นในระบบตรวจจับการบุกรุกจึงจะต้องมีส่วนที่เก็บกิจกรรมหรือพฤติกรรมการใช้งานปกติ (Normal Activity Profile) ซึ่งจะเก็บข้อมูลกิจกรรมหรือพฤติกรรมปกติของผู้ใช้ เครื่องคอมพิวเตอร์ หรือการเชื่อมต่อเครือข่าย เป็นต้น ข้อมูลเหล่านี้ถูกสร้างขึ้นโดยใช้ข้อมูลประวัติการใช้งานในช่วงการทำงานปกติ แล้วตัวตรวจจับจะเก็บข้อมูลเหตุการณ์ต่างๆ ในขณะเวลาหนึ่งๆ และใช้เกณฑ์ในการชี้วัดต่างๆ เช่น ตัวชี้วัดทางสถิติ ตัวชี้วัดด้านความถี่ เป็นต้น เพื่อบ่งบอกว่าเมื่อใดที่พฤติกรรมมีความผิดปกติ นั่นคือหากมีการกระทำใดที่ต่างจากที่กำหนดไว้ในระบบที่ระดับนัยสำคัญทางสถิติที่ระบุไว้ก็จะตีความว่าเป็นความพยายามที่จะบุกรุก หากพิจารณาถึงการกระทำที่เป็นการบุกรุกและการกระทำที่ผิดปกติจะพบว่ามีความเป็นไปได้ที่การกระทำที่ผิดปกติแต่ไม่ใช่การบุกรุกจะตรวจจับว่าเป็นการบุกรุก (เกิดปัญหา False Positive) และการกระทำที่เป็นการบุกรุกจะไม่ถูกตรวจจับว่าเป็นการกระทำที่ผิดปกติ (เกิดปัญหา False Negative) ซึ่งเป็นปัญหาที่สำคัญมาก

ประเด็นที่สำคัญอีกประการหนึ่งสำหรับระบบที่ตรวจจับการบุกรุกตามแนวทางนี้ คือ การเลือกระดับของการกระทำที่จะถือว่าเป็นการบุกรุกและการเลือกคุณลักษณะที่จะตรวจจับ นอกจากนี้ยังเป็นระบบที่มีค่าใช้จ่ายสูงเนื่องจากต้องทำการเก็บข้อมูลสำหรับการตรวจสอบและอาจต้องปรับปรุงเกณฑ์ในการชี้วัดพฤติกรรมปกติของระบบบ่อยๆ เพื่อให้การตรวจจับมีความถูกต้องมากขึ้น เทคนิคที่ใช้สำหรับการตรวจจับการบุกรุกแนวทาง Anomaly Detection มีดังนี้

3.6.1.1 Statistical approach

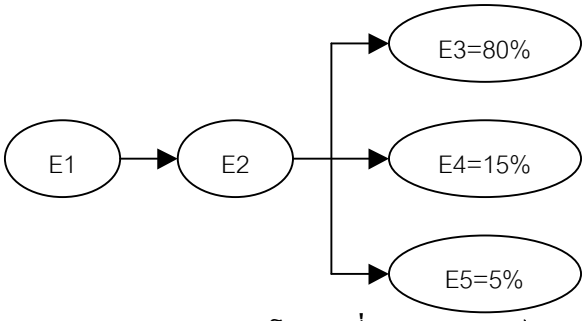
การใช้สถิติสำหรับการวิเคราะห์เพื่อตรวจจับการบุกรุกนั้นสามารถแบ่งได้ 2 วิธีการ คือ การวิเคราะห์เชิงปริมาณ (Quantitative Analysis) และใช้การวัดทางสถิติ (Statistical Measure) ซึ่งการทำงานของการตรวจจับโดยการวิเคราะห์เชิง ส่วนใหญ่จะอยู่ในรูปแบบของการตรวจจับโดยใช้ความถี่ (Threshold Detection) ในเทคนิคนี้คุณสมบัติต่างๆ ของผู้ใช้และพฤติกรรมของระบบจะถูกอธิบายหรือแสดงอยู่ในรูปแบบของตัวเลขหรือจำนวนนับ เช่น จำนวนไฟล์ที่มีการเข้าถึงโดยผู้ใช้ในช่วงเวลาหนึ่งๆ จำนวนครั้งของการพยายามที่ไม่สำเร็จในการเข้าระบบ ปริมาณการใช้งาน CPU ของ process เป็นต้น โดยเมื่อระบบตรวจจับเหตุการณ์ที่มีจำนวนครั้งหรือนับได้เกินตัวเลขขีดจำกัดหรือเงื่อนไขที่กำหนดไว้สำหรับแต่ละเหตุการณ์ก็จะถือว่ามีความผิดปกติเกิดขึ้น โดยระดับตัวเลข (Thresholds) อาจจะเป็นตัวเลขที่คงที่ (Static Thresholds)

หรือเปลี่ยนแปลงไปตามสภาพการใช้งาน (Heuristic Thresholds) ก็ได้ ซึ่งก็จะทำให้ความแม่นยำในการตรวจจับเพิ่มขึ้น อย่างไรก็ตามปัญหาที่สำคัญของวิธีการนี้ก็คือการกำหนดค่าระดับค่าที่เหมาะสมที่ใช้ในการตรวจจับของแต่ละเหตุการณ์ที่บ่งชี้ว่าเป็นพฤติกรรมที่ผิดปกติ โดยหากกำหนดให้มีค่าต่ำก็จะมีผลให้เกิดปัญหา False Positive แต่ถ้ากำหนดสูงเกินไปก็จะมีปัญหา False Negative เกิดขึ้น ส่วนการตรวจจับโดยการวัดทางสถิติ มีงานวิจัยที่ใช้แนวทางนี้และรู้จักกันอย่างแพร่หลายก็คือ Intrusion Detection Expert System (IDES) ซึ่งพัฒนาโดย SRI International และปัจจุบันได้พัฒนาเป็น Next-Generation Intrusion Detection Expert System(NIDES) และ Event Monitoring Enabling Responses to Anomalous Live Disturbances (EMERALD) [SRI International, 2002] โดยการทำงานจะเริ่มจากการสร้างไฟล์ข้อมูลพฤติกรรมของผู้ใช้ (Behavior Profile) เก็บไว้ก่อน เมื่อมีการใช้งานระบบก็จะมีการสร้างไฟล์พฤติกรรมขึ้นใหม่จากต้นฉบับเดิม โดยไฟล์เหล่านี้จะถูกออกแบบให้ใช้หน่วยความจำน้อยและมีประสิทธิภาพสำหรับการปรับปรุง เนื่องจากจะต้องมีการปรับปรุงข้อมูลในไฟล์ค่อนข้างมากในแต่ละครั้งที่ได้ข้อมูลการทำงานของโปรแกรมและข้อมูลการทำงานของระบบ เมื่อข้อมูลเหล่านี้ผ่านการประมวลผลแล้ว ระบบตรวจจับการบุกรุกก็จะได้ค่าตัวเลขซึ่งเป็นเกณฑ์ในการวัดความผิดปกติของไฟล์ข้อมูลพฤติกรรม โดยค่านี้เป็นส่วนหนึ่งของค่าความผิดปกติทั้งหมดที่มีอยู่ในไฟล์พฤติกรรม ไฟล์พฤติกรรม (Behavior Profile) หนึ่งจะประกอบด้วยเกณฑ์ในการชี้วัดหลายอย่าง เช่น อัตราการประมวลผลหรือการทำกิจกรรม การเข้าถึงไฟล์และกิจกรรมการนำเข้า-ส่งออกข้อมูลทั้งหมดของผู้ใช้แต่ละคน เวลาในการทำงานของ CPU จำนวนการเชื่อมต่อในช่วงเวลาหนึ่ง เป็นต้น โดยพฤติกรรมการใช้งานปัจจุบันของผู้ใช้แต่ละคนจะเก็บไว้ในไฟล์พฤติกรรมและจะมีการนำมาผสานรวมกันกับไฟล์พฤติกรรมที่เคยสร้างเก็บไว้เป็นระยะ ๆ เพื่อให้ระบบได้มีการเรียนรู้พฤติกรรมการใช้งานของผู้ใช้อยู่ตลอด ซึ่งจะมีผลให้การตรวจจับมีความถูกต้องมากขึ้น ในการตรวจจับพฤติกรรมที่ผิดปกติจะตัดสินโดยการเปรียบเทียบไฟล์พฤติกรรมปัจจุบันกับไฟล์พฤติกรรมที่เก็บไว้ แต่ปัญหาที่เกิดขึ้นกับแนวทางก็คือ หากการตรวจจับใช้เกณฑ์ทางสถิติเพียงอย่างเดียว ระบบตรวจจับอาจจะถูกฝึกฝนโดยผู้บุกรุกจนถึงจุดที่ทำให้มีการมองเหตุการณ์ที่เป็นการบุกรุกเป็นเหตุการณ์ปกติได้ เกณฑ์ระดับตัวเลข(Thresholds) ที่ใช้ในการตรวจจับว่าเป็นการบุกรุกนั้นอาจถูกกำหนดให้สูงหรือต่ำเกินไป และเนื่องการวัดทางสถิติจะไม่สนใจลำดับเหตุการณ์ที่เกิดขึ้น ซึ่งจะทำให้ไม่สามารถตรวจจับการบุกรุกที่สามารถบ่งชี้โดยใช้ลำดับและความสัมพันธ์กันของเหตุการณ์ต่าง ๆ

3.6.1.2 Predictive pattern generation

เป็นเทคนิคในการตรวจจับความผิดปกติที่ตั้งอยู่บนสมมุติฐานที่ว่า ลำดับของเหตุการณ์จะไม่กระจายแต่จะเป็นไปในรูปแบบที่สามารถทราบล่วงหน้าได้ ซึ่งให้ผลลัพธ์ในการตรวจจับการบุกรุกที่ดีกว่า เพราะว่าเทคนิคนี้จะพิจารณาความสัมพันธ์และลำดับของ

เหตุการณ์ด้วย ดังนั้นในการตรวจจับวิธีนี้จึงเป็นวิธีที่จะพยายามทำนายเหตุการณ์ที่อาจเกิดขึ้นในอนาคตจากเหตุการณ์ที่เคยเกิดขึ้นมาแล้ว ว่าจะมีโอกาสเกิดเหตุการณ์นั้นได้กี่เปอร์เซ็นต์และสร้างเป็นกฎไว้ [Teng et al., 1990] ตัวอย่าง เช่น



ภาพประกอบ 3.2 แสดงโอกาสที่เกิดเหตุการณ์ต่าง ๆ

โดยที่ E1-E5 เป็นเหตุการณ์ที่เกี่ยวกับความปลอดภัย จากกฎข้างต้นหมายความว่า ถ้าเกิดเหตุการณ์ E1 และ E2 โดย E2 เกิดหลัง E1 ก็จะมีความน่าจะเป็น 80% ที่จะเกิด E3 ตามมา หรือมีความน่าจะเป็น 15% ที่จะเกิด E4 ตามมา หรือมีความน่าจะเป็น 5% ที่จะเกิด E5 ตามมา

โดยชุดของกฎจะสร้างขึ้นจากการสังเกตพฤติกรรมของผู้ใช้ที่มีอยู่ในไฟล์พฤติกรรม ปัญหาของเทคนิคนี้ก็คือ การบุกรุกบางอย่างที่ไม่ได้ถูกสร้างเป็นกฎก็จะไม่ถูกตรวจจับว่าเป็นการบุกรุก ดังนั้นถ้าลำดับเหตุการณ์ A - B - C ที่เป็นการบุกรุก แต่ไม่ได้กำหนดเป็นกฎไว้ก็จะถูกจัดไว้ในกลุ่มที่ไม่สามารถระบุได้ ซึ่งปัญหานี้จะแก้ได้โดยการกำหนดว่าลำดับเหตุการณ์ที่ไม่รู้จักก็ให้เป็นการบุกรุกด้วย ซึ่งจะเป็นการเพิ่มโอกาสของการเกิด False Positive หรือกำหนดว่าไม่เป็นเหตุการณ์ที่เป็นการบุกรุก ก็จะเป็นการเพิ่มโอกาสของการเกิด False Negative ด้วยเช่นกัน อย่างไรก็ตามปกติเหตุการณ์ที่ถูกระบุว่าเป็นการบุกรุกซึ่งตรงตามกฎทางซ้าย แต่ค่าสถิติ (เปอร์เซ็นต์)ทางขวาก็จะเบี่ยงเบนไปจากค่าที่ทำนายไว้ค่อนข้างมาก แต่ข้อดีของแนวทางนี้ก็มีหลายประการด้วยกันคือ

- กฎที่มีรูปแบบเป็นลำดับสามารถตรวจจับกิจกรรมที่ผิดปกติซึ่งการตรวจจับด้วยวิธีการดั้งเดิมทำได้ยาก
- ระบบที่ใช้แนวทางนี้จะมีความสามารถในการปรับตัวได้สูงสำหรับการเปลี่ยนแปลงเนื่องจากรูปแบบที่ไม่ดีจะค่อย ๆถูกกำจัดออกไป
- ระบบสามารถตรวจจับผู้ใช้ที่พยายามฝ่ากระบบในระหว่างที่ระบบกำลังเรียนรู้ได้ง่าย

- กิจกรรมที่ผิดปกติสามารถตรวจจับและรายงานได้อย่างรวดเร็วหลังจากที่ได้รับข้อมูลเหตุการณ์

3.6.1.3 Neural networks

เป็นแนวทางในการตรวจจับการบุกรุกโดยใช้วิธีการเรียนรู้จากข้อมูลในอดีต โดยการใช้ neural network เพื่อให้สามารถบ่งชี้ถึงความผิดปกติที่เกิดขึ้นในระบบ ซึ่ง neural network จะถูกฝึกให้เรียนรู้โดยใช้ข้อมูลการทำงานของผู้ใช้หรือระบบในอดีต และต้องเป็นข้อมูลของระบบที่ยังไม่มีการบุกรุกเกิดขึ้น หลังจากนั้นระบบตรวจจับก็จะพยายามเปรียบเทียบข้อมูลการทำงานที่เกิดขึ้นจริงกับข้อมูลที่เก็บอยู่ใน neural network เพื่อตรวจจับหาความผิดปกติ ข้อดีของวิธีการนี้คือ ผลของการตรวจจับด้วยเทคนิคนี้จะไม่ขึ้นกับการตั้งสมมุติฐานทางสถิติเกี่ยวกับลักษณะของข้อมูลเหมือนกับวิธีการ statistical approach จึงสามารถรับมือกับข้อมูลที่ความซับซ้อนได้ดี แต่อย่างไรก็ตาม แนวทางนี้ก็มีปัญหาบางประการคือ หากเลือกช่วงเวลาที่สั้นเกินไปในการเก็บข้อมูลเพื่อใช้สำหรับการเรียนรู้ ก็จะส่งผลให้ได้ผลลัพธ์การทำงานที่ไม่ดี ทำให้เกิด false positive ขณะที่หากเลือกช่วงเวลาที่กว้างก็จะมีผลให้มีข้อมูลที่ได้มีการกระจายไม่สัมพันธ์กันและทำให้เพิ่มโอกาสของการเกิด false negative เช่นกัน นอกจากนี้ผู้บุกรุกสามารถที่จะฝึก neural network ได้หากระบบตรวจจับอยู่ในระยะของฝึกการเรียนรู้

3.6.2 Misuse Detection

หลักการของการตรวจจับการบุกรุกด้วยแนวทาง Misuse Detection คือ ตัวตรวจจับการบุกรุกจะวิเคราะห์กิจกรรมของระบบ โดยการพิจารณาเหตุการณ์หรือชุดของเหตุการณ์ที่ตรงกับรูปแบบเหตุการณ์ที่ได้กำหนดไว้แล้วว่าเป็นการบุกรุก โดยจะอธิบายถึงการบุกรุกต่างๆ ที่รู้จัก รูปแบบของเหตุการณ์การบุกรุกที่รู้จักเหล่านี้จะเรียกว่า ร่องรอยการบุกรุก (Signatures) ดังนั้นบางครั้งจึงเรียกแนวทาง Misuse Detection ว่า Signature-based Detection แนวทางนี้จะใช้ข้อมูลความรู้เกี่ยวกับพฤติกรรมที่เป็นการบุกรุกหรือไม่ยอมรับ และค้นหาเพื่อตรวจจับพฤติกรรมเหล่านี้โดยตรง ซึ่งตรงกันข้ามกับ Anomaly Detection ที่จะค้นหาเพื่อตรวจจับส่วนตรงกันข้ามกับพฤติกรรมปกติ

ประเด็นสำคัญของการตรวจจับแนวทางนี้ก็คือ จะเขียนรูปแบบหรือร่องรอยการบุกรุกอย่างไรให้คลอบคลุมถึงความหลากหลายทั้งหมดที่เป็นไปได้ของการบุกรุก และเขียนรูปแบบหรือร่องรอยการบุกรุกอย่างไรที่จะไม่ตรงกับกิจกรรมที่ไม่ใช่การบุกรุก เทคนิคที่ใช้สำหรับแนวทางการตรวจจับ misuse detection มีดังนี้

3.6.2.1 Production/Expert systems

เป็นหนึ่งในเทคนิคสำหรับการตรวจจับการบุกรุกที่ใช้ระบบตรวจจับยุคแรกๆ ซึ่งระบบตรวจจับการบุกรุกที่ใช้เทคนิคนี้ ตัวอย่าง เช่น NIDES และ EMERALD ซึ่งเป็นระบบตรวจจับการบุกรุกที่ใช้ทั้งแนวทาง Anomaly Detection และ Misuse Detection โดยในส่วนของ Anomaly Detection นั้นใช้เทคนิค Statistical approach ใช้ตัวชี้วัดทางสถิติในการตรวจจับ และ Misuse Detection ใช้เทคนิค Expert System เหมือนกัน แต่เครื่องมือที่ใช้ในการวิเคราะห์การบุกรุกนั้นต่างกันโดย NIDES และ EMERALD จะใช้ P-BEST (The Production-Based Expert System Toolset) [Lindqvist and Porras, 1999] ซึ่งออกแบบโดย Alan Whitehurst และทีมงานของ Multics Intrusion Detection and Alerting System (MIDAS) [Whitehurst et al., 1998] ส่วนระบบ Computer Misuse Detection System (CMDS) จะใช้ CLIPS (The C Language Integrated Production System) ซึ่งพัฒนาโดย The National Aeronautics and Space Administration (NASA) [Biege, 2001] [Riley, 2004]

จุดเด่นของเทคนิคนี้อยู่ที่การแยกส่วนของการวินิจฉัย (Inference Engine) ที่มีหน้าที่ในการตัดสินใจว่ามีการบุกรุกเกิดขึ้นหรือไม่โดยใช้กฎ (Rule) และข้อเท็จจริง (Fact) ที่เกิดขึ้นออกมาจากส่วนข้อกำหนดหรือกฎที่อธิบายถึงรูปแบบการบุกรุกแบบต่างๆ ดังนั้นผู้ใช้สามารถเพิ่มกฎหรือฐานความรู้เกี่ยวกับการบุกรุกได้สะดวกและข้อเท็จจริงจะระบุในรูปแบบของเหตุการณ์ Audit Trail โดยฐานความรู้การบุกรุกจะเขียนในรูปแบบของกฎ if-then โดยเงื่อนไขที่แสดงถึงการบุกรุกจะอยู่ทางด้านซ้าย (ส่วนของ if) ของกฎ เมื่อเงื่อนไขการทำงานเป็นไปตามกฎการกระทำที่กำหนดไว้ทางด้านขวาของกฎ (ส่วนของ then) ก็จะถูกสั่งให้ทำงาน ดังนั้นก่อนที่จะมีใช้ระบบตรวจจับที่ใช้เทคนิคนี้ผู้ใช้จะต้องมีการกำหนดกฎที่อธิบายถึงรูปแบบการบุกรุกที่จะใช้ในการตรวจจับ ซึ่งเป็นงานที่ยากและใช้เวลามากทีเดียว ข้อเสียของแนวทางนี้มีดังนี้

- ต้องมีทักษะและความชำนาญเป็นอย่างมากสำหรับการที่จะทำให้ระบบที่ใช้เทคนิคนี้มีการทำงานที่ดี นั่นคือมีโอกาสที่ระบบนี้จะทำงานผิดพลาดในการตรวจจับการบุกรุกได้
- สามารถตรวจจับได้เฉพาะการบุกรุกที่กำหนดไว้ในกฎเท่านั้น
- การเพิ่มและการลบกฎ ต้องพิจารณาถึงผลกระทบที่จะเกิดขึ้นเนื่องการเปลี่ยนแปลงดังกล่าวกับกฎที่เหลืออยู่ในฐานความรู้ด้วย
- เกิดปัญหาในทางปฏิบัติ โดยเฉพาะเมื่อนำไปใช้ในการตรวจจับในงานที่มีข้อมูลในปริมาณมากๆ เนื่องจากส่วนของการกำหนดกฎที่ใช้อธิบายการบุกรุก ส่วนใหญ่จะทำงานในลักษณะของ Interpreter ซึ่งทำงานช้ากว่าระบบที่ผ่านการ Compile

3.6.2.2 Keystroke monitoring

เป็นเทคนิคที่ใช้ข้อมูลการกดคีย์บอร์ดเพื่อตรวจจับการบุกรุก โดยระบบนี้จะคอยติดตามเก็บข้อมูลการกดคีย์บอร์ดแล้วนำเปรียบเทียบกับรูปแบบลำดับการกดคีย์บอร์ดที่บ่งชี้ว่าเป็นการบุกรุก ซึ่งเป็นเทคนิคที่ง่ายแต่ก็มีข้อเสียหลายอย่างเช่น มีวิธีการแสดงรูปแบบการกดคีย์บอร์ดได้หลากหลายมากสำหรับการบุกรุกแต่ละวิธี เทคนิคนี้สามารถหลบเลี่ยงได้ง่ายสำหรับระบบที่ใช้ shells เช่น *bash*, *ksh*, และ *tcsh* ซึ่งยอมให้มีการกำหนดชื่ออื่น ๆ (Alias) แทนคำสั่งเดิม เว้นแต่มีการวิเคราะห์ความหมายของการกดคีย์บอร์ดด้วย นอกจากนี้วิธีการนี้ไม่ได้วิเคราะห์โปรแกรมที่กำลังทำงานแต่ติดตามเฉพาะการกดคีย์เท่านั้น จึงไม่สามารถตรวจจับโปรแกรมที่ประสงค์ร้ายได้เช่น โปรแกรมประเภทม้าโทรจัน และระบบปฏิบัติการไม่มีกลไกที่สนับสนุนการดักจับการกดคีย์บอร์ด ดังนั้นจึงส่วนที่ใช้ติดตามการกดคีย์บอร์ดจะต้องคอยดักจับและวิเคราะห์การกดคีย์บอร์ดก่อนที่คีย์เหล่านั้นจะถูกส่งไปยังผู้รับปลายทาง

3.6.2.3 Model Based Intrusion Detection

เป็นเทคนิคที่ใช้หลักการที่ว่า รูปแบบเหตุการณ์ที่แน่นอนจะถูกวินิจฉัยได้โดยการสังเกตจากกิจกรรมอื่น ๆ ที่ได้กำหนดไว้ และมีทางเป็นไปได้ที่จะค้นหาความพยายามในการบุกรุกโดยการมองดูกิจกรรมที่แสดงถึงรูปแบบการบุกรุก ถ้ามีการติดตามและตรวจสอบกิจกรรมเหล่านี้ การตรวจจับรูปแบบนี้ประกอบด้วยโมดูลที่สำคัญ 3 โมดูล โดย *anticipator* ซึ่งใช้ Active Models และ Scenario Models เพื่อทำนายขั้นตอนถัดไปในเหตุการณ์ที่คาดว่าจะเกิดขึ้น โดย Scenario Model คือ ฐานความรู้ของข้อกำหนดเหตุการณ์ที่เป็นการบุกรุก แล้ว *planner* จะทำหน้าที่แปลสมมติฐานไปเป็นรูปแบบที่แสดงพฤติกรรมที่จะเกิดขึ้นใน Audit Trail ซึ่ง planner จะใช้ข้อมูลที่ได้ทำนายไว้เพื่อวางแผนว่าจะค้นหาอะไรในลำดับถัดไป แล้ว *interpreter* ก็จะทำหน้าที่ค้นหาข้อมูลเหล่านี้ใน Audit Trail และจะกระทำในลักษณะนี้ไปเรื่อย ๆ จนได้หลักฐานสำหรับการบุกรุกที่มากพอจนกระทั่งถึงระดับตัวเลขที่กำหนดแล้วจึงส่งสัญญาณบอกว่ามีความพยายามในการบุกรุกเกิดขึ้น

แนวทางนี้เป็นแนวทางที่ดี เนื่องจาก planner และ interpreter รู้ว่าจะค้นหาอะไรในแต่ละขั้นตอน โดยข้อมูลจำนวนมากที่ไม่เกี่ยวข้องใน Audit Trail จะถูกกรองออกไป ซึ่งจะนำไปสู่ประสิทธิภาพการทำงานที่ดีขึ้น นอกจากนี้ระบบสามารถทำนายการทำงานแต่ละขั้นตอนของการบุกรุกบนพื้นฐานของต้นแบบการบุกรุก โดยข้อมูลการทำนายเหล่านี้สามารถใช้สำหรับการตรวจสอบสมมุติฐานของการบุกรุกได้ เพื่อนำไปสู่มาตรการในการป้องกันหรือตัดสินว่าจะค้นหาข้อมูลใดในขั้นตอนถัดไป

แต่อย่างไรก็ตามก็มีประเด็นที่ต้องระมัดระวังสำหรับเทคนิคนี้คือรูปแบบสำหรับเหตุการณ์การบุกรุกจะต้องสามารถตรวจจับได้ง่าย รูปแบบนั้นจะต้องเกิดขึ้นเสมอใน

พฤติกรรมที่กำลังเฝ้าดูอยู่ และรูปแบบเหล่านั้นต้องแตกต่างกันและไม่มีความสัมพันธ์กับพฤติกรรมปกติอื่น ๆ

3.6.2.4 State Transition Analysis Technique

เทคนิคนี้ใช้สำหรับการแสดงลำดับของการกระทำที่ผู้บุกรุกปฏิบัติเพื่อที่จะบุกรุกเข้ามาในระบบ โดยการกระทำและความต้องการจะถูกแสดงด้วยแผนภาพการเปลี่ยนสถานะ (State Transition Diagram) โดยอยู่บนหลักการที่ว่า การบุกรุกทั้งหมดจะมีคุณลักษณะทั่วไป 2 ประการคือ ผู้บุกรุกจะเข้าถึงระบบเป้าหมายไม่ทางใดก็ทางหนึ่ง และผลลัพธ์จากการบุกรุกจะทำให้ผู้บุกรุกได้มาซึ่งความสามารถบางอย่างที่ไม่เคยมีมาก่อน ดังนั้นการบุกรุกจะทราบได้จากลำดับการกระทำของผู้บุกรุกที่นำระบบ จากสถานะเริ่มต้น (Initial State) ไปยังสถานะที่ถูกบุกรุก (Compromised State) โดยผ่านสถานะระหว่างกลาง (Intermediate States) จำนวนหนึ่ง ซึ่งสถานะเริ่มต้น (Initial State) คือสถานะของระบบก่อนการบุกรุก สถานะที่ถูกบุกรุก (Compromised State) เป็นสถานะของระบบภายหลังการบุกรุกประสบความสำเร็จ ขั้นตอนที่ผู้บุกรุกกระทำจะถูกแสดงในรูปแบบการเปลี่ยนสถานะ (State Transition) โดยสถานะในที่นี้จะหมายถึงสถานะของระบบซึ่งประกอบด้วย สิทธิของผู้ใช้ (User Privileges) ไฟล์ที่มีอยู่ในระบบหรือสถานะปัจจุบันของเครือข่ายหรือบริการของระบบ เป็นต้น

ระบบตรวจจับการบุกรุกที่ใช้แนวทางนี้มีแผนภาพการเปลี่ยนสถานะ (State Transition Diagram) สำหรับทุกรูปแบบการบุกรุก เมื่อมีกิจกรรมเกิดขึ้นในระบบ ส่วนของการวิเคราะห์ก็จะตรวจสอบกิจกรรมนั้นกับแผนภาพการเปลี่ยนสถานะและความเปลี่ยนแปลงจากสถานะปัจจุบันไปยังสถานะใหม่ด้วย ถ้ากิจกรรมนั้นตรงกับการเปลี่ยนสถานะ ก็แสดงว่ามีความเป็นไปได้ที่จะเกิดการบุกรุกหรือสาเหตุของการบุกรุก และถ้ากิจกรรมในขั้นตอนถัดไปเป็นไปตามรูปแบบการบุกรุกจนถึงขั้นตอนสุดท้าย ก็จะหมายความว่ามีการบุกรุกเกิดขึ้น

ระบบที่ใช้เทคนิคนี้คือ STAT (State Transition Analysis) และได้มีการนำไปประยุกต์ใช้สำหรับระบบยูนิกซ์ในชื่อ USTAT (State Transition Analysis Tool for UNIX) ซึ่งทั้งสองระบบพัฒนาโดย University of California, Santa Barbara โดยที่ Phillip Porras และ Richard Kemmerer พัฒนา STAT [Porras, 1992] ส่วน Koral llgun และ Richard Kemmerer พัฒนา USTAT และได้มีการนำเทคนิค STAT นี้ไปประยุกต์สำหรับการตรวจจับการบุกรุกโดยการวิเคราะห์การเปลี่ยนสถานะของโปรเซส [Nuansri, 1999] โดยที่สถานะของโปรเซส จะถูกระบุโดยใช้ค่าที่ใช้ในการบ่งบอกถึงฐานะของผู้ใช้แต่ละคน คือ ค่า User-Id และ Group-Id (UID,GID) การเปลี่ยนสถานะจะเกิดขึ้นก็ต่อเมื่อค่าดังกล่าวเปลี่ยนจากค่าหนึ่งไปเป็นอีกค่าหนึ่ง ดังนั้นหากมีการกำหนดค่าสถานะของโปรเซสและข้อกำหนดของการเปลี่ยนสถานะไว้เป็นกฎแล้วหากต่อมามีเหตุการณ์ความพยายามบุกรุกหรือละเมิดกฎเกิดขึ้นก็หมายความว่ามีการละเมิดกฎ

ของการรักษาความปลอดภัยเกิดขึ้นในระบบ โดยระบบตรวจจับการบุกรุกจะใช้ system call ชื่อ ktrace() ที่ผ่านการปรับปรุงให้สามารถนำมาประยุกต์ใช้ในการติดตามสถานะและการเปลี่ยนสถานะของโปรเซส

สำหรับข้อดีของแนวทางนี้คือ สามารถตรวจจับการร่วมกันโจมตี และสามารถมองเห็นสถานะของระบบก่อนที่การบุกรุกจะสำเร็จ แต่ปัญหาสำหรับแนวทางก็คือ ต้องระบุรูปแบบการโจมตีในลักษณะลำดับของเหตุการณ์เท่านั้น ทำให้การกำหนดรูปแบบการบุกรุกมีความซับซ้อนมากขึ้น นอกจากนี้การตรวจจับด้วยแนวทางนี้จะไม่สามารถตรวจจับการโจมตีแบบ DoS การใช้งานที่ผิดไปจากรูปแบบการใช้งานปกติ และการดักจับข้อมูลได้ เนื่องจากการกระทำเหล่านี้ไม่มีการบันทึกข้อมูลการทำงานลงในล็อกไฟล์หรือการกระทำดังกล่าวไม่สามารถแสดงได้ด้วยแผนภาพแสดงการเปลี่ยนสถานะได้

3.7 คุณลักษณะของระบบตรวจจับการบุกรุกที่ดี

ระบบตรวจจับการบุกรุกที่ดีควรมีความสามารถและคุณสมบัติดังต่อไปนี้ [Price, 1998]

1. ทำงานอยู่ตลอดเวลาได้เองโดยไม่ต้องมีการควบคุมของผู้ดูแลระบบ และระบบต้องมีความน่าเชื่อถือที่เพียงพอที่จะทำงานในลักษณะอยู่เบื้องหลัง (background process) แต่ผู้ดูแลระบบต้องสามารถตรวจสอบการทำงานจากภายนอกได้
2. เป็นระบบที่เป็น fault tolerant ในความหมายที่ว่ายังคงสามารถที่จะทำงานต่อไปได้ในกรณีที่ระบบคอมพิวเตอร์เกิดปัญหาหรือมีข้อผิดพลาดเกิดขึ้นและไม่ต้องมีการสร้างฐานข้อมูลความรู้(knowledge-base) ทุกครั้งที่เริ่มระบบ
3. มีความสามารถในการตรวจสอบตัวเองเพื่อไม่ให้ถูกลบ ทำลาย แก้ไข หรือถูกแทนที่ด้วยโปรแกรมอื่นได้
4. ส่งผลกระทบกับการทำงานของระบบคอมพิวเตอร์น้อยที่สุด นั่นคือในการทำงานของระบบตรวจจับจะต้องใช้ทรัพยากรของระบบคอมพิวเตอร์น้อยที่สุดเท่าที่จะทำได้ เพราะถ้าหากทำให้ระบบคอมพิวเตอร์ทำงานช้าลงแล้วก็จะส่งผลให้ไม่มีการใช้งานระบบตรวจจับการบุกรุก
5. ตรวจสอบการทำงานที่ผิดไปจากรูปแบบการทำงานปกติได้
6. ปรับเปลี่ยนหรือแก้ไขให้เข้ากับระบบคอมพิวเตอร์ได้ง่าย เนื่องจากแต่ละระบบคอมพิวเตอร์จะมีรูปแบบการใช้งานและกลไกในการป้องกันที่ต่างกัน

7. ปรับการทำงานให้สอดคล้องกับการเปลี่ยนแปลงพฤติกรรมการใช้งานของระบบได้ เช่นเมื่อมีการติดตั้งโปรแกรมใหม่ ระบบ IDS ก็ต้องสามารถปรับเปลี่ยนการทำงานให้เข้ากับระบบที่เปลี่ยนไปได้
8. มีความผิดพลาดในการทำงานน้อยที่สุด

ซึ่งความผิดพลาดจากการทำงานของระบบตรวจจับการบุกรุกสามารถจำแนกออกเป็นประเภทต่าง ๆ ได้แก่ False positive, False negative และ Subversion error

ความผิดพลาดในทางบวก (False positive) จะทำให้ผู้ใช้ระบบ IDS ต้องคอยละเลยผลลัพธ์การทำงานที่ผิดพลาดซึ่งระบุว่าเป็นการบุกรุก ความผิดที่เกิดขึ้นในลักษณะนี้ควรจะเกิดขึ้นน้อยที่สุด(อาจจะเป็นไปไม่ได้ที่จะไม่ให้เกิดขึ้นเลย) แต่ถ้าหากเกิด false positive มาก ๆ ผู้ดูแลระบบก็จะละเลยผลลัพธ์จากการทำงานของระบบ IDS ตลอดเวลา ซึ่งจะนำไปสู่เหตุการณ์ที่มีการบุกรุกที่เกิดขึ้นจริงและระบบตรวจจับได้แต่ถูกละเลยโดยผู้ดูแลระบบ

ความผิดพลาดในทางลบ (False negative) เกิดขึ้นเมื่อระบบ IDS ไม่พบความผิดปกติใด ๆ เมื่อมีการบุกรุกเกิดขึ้นจริงในระบบ ซึ่งเป็นความผิดพลาดที่ร้ายแรงกว่าความผิดพลาดแบบ false positive เนื่องจากระบบตรวจจับการบุกรุกจะทำให้เกิดความเข้าใจที่ผิดพลาดในเรื่องการความปลอดภัยของระบบ นอกจากนี้การปล่อยให้เกิดการบุกรุกเกิดขึ้นหรือมีเหตุการณ์ผิดปกติเกิดขึ้นโดยไม่มีการแจ้งเตือนไปยังผู้ดูแลระบบ ซึ่งกรณีนี้ระบบตรวจจับการบุกรุกมีผลกระทบให้เกิดความปลอดภัยน้อยกว่าก่อนที่จะมีระบบตรวจจับการบุกรุก

Subversion error เป็นความผิดพลาดที่ซับซ้อนและก่อให้เกิดผิดพลาดในทางลบ (false negative) โดยผู้บุกรุกอาจจะใช้ความรู้เกี่ยวกับการทำงานภายในระบบตรวจจับบุกรุก เพื่อปรับเปลี่ยนการทำงานของระบบตรวจจับให้ไม่สามารถที่จะตรวจสอบการกระผิดที่กำลังกระทำอยู่

3.8 Audit Trail

Audit Trail คือ ชุดของรายการข้อมูลเหตุการณ์ต่าง ๆ ของเครื่องคอมพิวเตอร์ที่เกี่ยวกับระบบปฏิบัติการ โปรแกรมประยุกต์หรือกิจกรรมต่าง ๆของผู้ใช้ ในระบบคอมพิวเตอร์หนึ่ง ๆ อาจจะมี Audit Trail หลายตัวโดยแต่ละตัวก็จะใช้สำหรับบันทึกกิจกรรมใดกิจกรรมหนึ่งโดยเฉพาะ ซึ่งบางครั้งอาจจะใช้คำว่า Audit Log แทนในกรณีนี้ก็ได้ ดังนั้นจึงมีการนำข้อมูลชุดรายการเหล่านี้ไปใช้ในงานที่เกี่ยวข้องกับการรักษาความปลอดภัยในหลายวัตถุประสงค์ด้วยกัน เช่น

- Individual Accountability : การใช้งานและกิจกรรมของแต่ละคนในระบบจะถูกบันทึกไว้ใน Audit Trail โดยอนุญาตให้ผู้ใช้สามารถดูรายละเอียดการใช้งานของตนเองได้ ซึ่งจะช่วยยับยั้ง

ไม่ให้ผู้ใช้ฝ่าฝืนนโนบายเกี่ยวกับการรักษาความปลอดภัย ถึงแม้ว่าจะมีการฝ่าฝืนก็จะมีรายละเอียดการทำงานต่าง ๆ เก็บอยู่ใน Audit Trail

- Reconstructing Event : ข้อมูลใน Audit Trail สามารถใช้ในการสร้างเหตุการณ์ขึ้นมาใหม่หลังจากที่เกิดปัญหาขึ้น โดยปริมาณความเสียหายที่เกิดขึ้นกับเหตุการณ์นั้นจะสามารถประเมินได้ง่ายขึ้น โดยการตรวจสอบจาก Audit Trail ของกิจกรรมระบบที่บ่งบอกตำแหน่งว่าเหตุการณ์นั้นเกิดขึ้นได้อย่างไร เมื่อใด และทำไมจึงเกิดขึ้น
- Problem Analysis : Audit Trail อาจจะถูกใช้เป็นเสมือนเครื่องมือแบบ on-line ในการช่วยในการติดตามและระบุปัญหาที่เกิดขึ้น เช่น ใช้ในการตรวจจับปัญหาของเครื่องคอมพิวเตอร์ เช่น ดิสก์ไม่ทำงาน การใช้ทรัพยากรของระบบหรือเครือข่ายมากเกินไป เป็นต้น
- *Intrusion Detection* : Audit Trail สามารถช่วยในการตรวจจับการบุกรุกอย่างได้ผล ถ้าหากได้มีกำหนดให้มีการบันทึกข้อมูลที่เหมาะสมสำหรับการวิเคราะห์ข้อมูล ซึ่งในการตรวจจับการบุกรุกนั้นสามารถทำได้ทั้งในลักษณะทันทีทันใด (Real Time) โดยการตรวจสอบข้อมูล Audit Log ในทันทีที่มีการสร้างขึ้นมา หรือภายหลังเกิดเหตุการณ์แล้ว (After the Fact)

3.9 C2 Audit

จากประโยชน์ของการใช้ข้อมูล Audit Trail ในงานที่เกี่ยวกับการรักษาความปลอดภัยของระบบคอมพิวเตอร์ดังที่กล่าวมาแล้ว ดังนั้นศูนย์ความปลอดภัยคอมพิวเตอร์กระทรวงกลาโหม สหรัฐอเมริกา (Department of Defense Computer Security Center) หรือ ตัวย่อ DoDCSC จึงได้พัฒนา Trusted Computer System Evaluation Criteria (TCSEC) ขึ้นในปี 1985 [Department Of Defense Standard, 1985] เพื่อใช้เป็นเกณฑ์ในการพิจารณาระบบคอมพิวเตอร์ต่าง ๆ ว่ามีความปลอดภัยมากน้อยเพียงใด โดยกระทรวงกลาโหมสหรัฐฯ ได้อ้างอิงถึงเกณฑ์นี้เพื่อใช้ภายในกระทรวงเอง นอกจากนี้แล้ว TCSEC ยังได้รับการยอมรับกันทั่วไป โดยผู้ใช้และผู้ขายระบบคอมพิวเตอร์และเครือข่ายได้มีการอ้างอิงถึงเกณฑ์นี้กันอย่างแพร่หลาย

เกณฑ์นี้ได้แบ่งระดับความปลอดภัยของระบบคอมพิวเตอร์เป็น 4 ระดับคือ D, C, B, และ A เรียงตามลำดับจากความปลอดภัยน้อยที่สุดไปถึงความปลอดภัยสูงที่สุด โดยที่ในระดับความปลอดภัย C และ B จะแบ่งออกเป็นระดับความปลอดภัยย่อย ๆอีกคือ C1, C2, B1, B2, และ B3

ใน TCSEC ระบุว่าสำหรับระดับความปลอดภัยตั้งแต่ C2 ถึงระดับ A1 นั้น การกระทำต่าง ๆของผู้ใช้จะต้องสามารถให้มีการ audit ได้ ซึ่งการ audit นี้ก็คือกระบวนการบันทึกตรวจสอบ และวิเคราะห์ทบทวนกิจกรรมที่เกี่ยวข้องกับความปลอดภัยทั้งหลายในระบบ และ

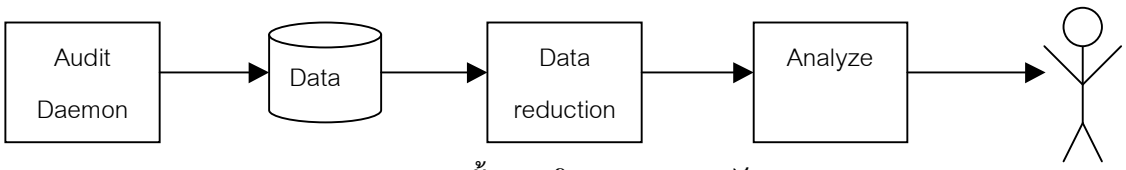
ระดับความปลอดภัย C2 นั้นเป็นที่ยอมรับกันทั่วไปว่ามีความปลอดภัยเพียงพอสำหรับการปฏิบัติงานที่ไม่เกี่ยวข้องกับปฏิบัติการทางทหาร

ถึงแม้ว่าจะมีเกณฑ์ที่เป็นมาตรฐาน TCSEC ที่ระบุถึงรายละเอียดข้อมูลสำหรับการ audit แต่ก็ยังไม่มีมาตรฐานที่เป็นที่ยอมรับสำหรับรูปแบบข้อมูลของข้อมูลล็อก จึงทำให้แต่ละระบบที่สนับสนุนการ audit ในระดับ C2 ก็จะมีรูปแบบการบันทึกข้อมูลลงในล็อกไฟล์ที่ต่างกัน การควบคุมจัดการที่ต่างกัน และรวมทั้งที่เก็บล็อกไฟล์ก็ต่างกันด้วย

3.10 ขั้นตอนในการวิเคราะห์ข้อมูลล็อก

โดยทั่วไปแล้วภาพรวมของกระบวนการในตรวจสอบและวิเคราะห์ข้อมูลล็อกทั้งระบบนั้นสามารถแบ่งได้เป็นขั้นตอนดังนี้ [Wetmore, 1993]

1. เลือกชนิดของข้อมูลที่ต้องการเก็บรวบรวม
2. รวบรวมข้อมูล
3. ลดปริมาณของข้อมูลที่ได้เก็บรวบรวมมา โดยการตัดส่วนที่ไม่มีประโยชน์หรือไม่เกี่ยวข้องกับการวิเคราะห์ข้อมูลออกไป
4. วิเคราะห์ข้อมูล
5. แจ้งผลที่ได้จากการวิเคราะห์ไปยังผู้รับผิดชอบ และ
6. กระทำกิจกรรมต่าง ๆที่เหมาะสมกับเหตุการณ์



ภาพประกอบ 3.3 ขั้นตอนในการวิเคราะห์ล็อก

การตรวจสอบหรือวิเคราะห์ข้อมูลในล็อกไฟล์นั้นสามารถแบ่งแนวทางในการตรวจสอบออกได้เป็น 3 แนวทาง คือ

1. Audit trail review after an event เป็นการตรวจสอบข้อมูลล็อกเมื่อพบว่าระบบหรือโปรแกรมที่ใช้งานเกิดปัญหาขึ้นหรือมีการฝ่าฝืนข้อบังคับการใช้งานของผู้ใช้
2. Periodic review of audit trail data หรืออาจจะเรียกย่อว่า Periodic Analysis จะเป็นการตรวจสอบข้อมูลในล็อกไฟล์ตามระยะเวลาที่เหมาะสมซึ่งกำหนดโดยผู้ดูแลระบบ เช่น ตรวจสอบข้อมูลล็อกวันละครั้ง เป็นต้น

3. Real-time audit analysis ข้อมูลล็อกจะถูกตรวจสอบและวิเคราะห์โดยทันทีหลังจากที่มีการบันทึกลงในล็อกไฟล์และในกรณีที่ตรวจพบเหตุการณ์ผิดปกติหรือน่าสงสัยก็จะมีการตอบสนองโดยการทำกิจกรรมหรือเหตุการณ์ที่กำหนดไว้ ซึ่งเป็นการวิเคราะห์ข้อมูลล็อกในทันทีทันใดหรือในเวลาที่ใกล้เคียงกับเวลาที่เกิดเหตุการณ์จริง ซึ่งอาจจะเรียกได้ว่าเป็นการวิเคราะห์ข้อมูลล็อกแบบ Real-time analysis

3.11 เครื่องมือที่ใช้ในการวิเคราะห์ล็อก

โดยทั่วไปแล้วการตรวจสอบข้อมูลล็อกไฟล์นั้น ควรจะตรวจสอบอย่างสม่ำเสมอ แต่ปัญหาสำคัญในการวิเคราะห์และตรวจสอบล็อกไฟล์ให้มีประสิทธิภาพก็คือ ขนาดของล็อกไฟล์จะมีขนาดใหญ่ และข้อมูลล็อกส่วนใหญ่เป็นข้อมูลที่ไม่มีประโยชน์หรือเกี่ยวข้องกับการรักษาความปลอดภัย ทำให้การตรวจสอบข้อมูลมีความยุ่งยาก บางครั้งผู้ดูแลระบบเองอาจจะเผลอเรอหรือมองข้ามในบางจุดไป ซึ่งอาจจะก่อให้เกิดความเสียหายต่อระบบได้ ดังนั้นจึงมีการพัฒนาเครื่องมือเพื่อช่วยในการตรวจสอบและวิเคราะห์ล็อกไฟล์ โดยมีแนวทางในการวิเคราะห์ที่แตกต่างกัน เช่น SWATCH และ Logsurfer จะใช้แนวทางในการวิเคราะห์แบบ Real-Time Analysis ส่วน Logcheck และ Logwatch ใช้แนวทางในการวิเคราะห์แบบ Periodic Analysis เป็นต้น

3.11.1 SWATCH : The Simple WATCHer [Hansen and Atkins, 1993]

SWATCH เป็นเครื่องมือสำหรับการวิเคราะห์ล็อกไฟล์และจัดเป็นระบบตรวจจับการบุกรุกแบบ HIDS ซึ่งพัฒนาโดย Stephen Hansen และ Todd Atkins ด้วยภาษา Perl เป็นโปรแกรมที่ถูกออกแบบมาเพื่อให้ค้นหาสิ่งผิดปกติที่เกิดขึ้นในล็อกไฟล์และแจ้งเตือนผู้ดูแลระบบผ่านการกระทำต่าง ๆ ที่สามารถกำหนดได้ค่อนข้างยืดหยุ่น เช่น จดหมายอิเล็กทรอนิกส์ ส่งเสียง beep หรือสั่งให้โปรแกรมที่เขียนเตรียมไว้แล้วทำงาน โดยสามารถทำงานในโหมด real-time และทั้งโหมด periodic ซึ่งใช้ในกรณีที่ต้องการค้นหาสิ่งที่ต้องการในล็อกไฟล์เก่า ๆ ที่ไม่มีการเพิ่มของข้อมูลแล้ว โดยทั่วไปแล้วนิยมใช้ SWATCH ทำงานในโหมด real-time โดยจุดประสงค์หลักที่สำคัญในการสร้าง SWATCH ขึ้นมานั้นมีดังนี้

1. ผู้ดูแลระบบสามารถเรียนรู้เพื่อกำหนดการใช้งานได้ในระยะเวลาสั้น ๆ
2. มีการกระทำ (action) ง่าย ๆ ที่สามารถใช้งานได้
3. อนุญาตให้ผู้ใช้กำหนดการกระทำ (action) ของตัวเองขึ้นมาใช้งานได้ รวมทั้งสามารถส่งผลลัพธ์ที่ได้ไปยังโปรแกรมที่เขียนขึ้นมารองรับได้

4. ขณะที่ SWATCH กำลังทำงาน สามารถปรับเปลี่ยน configuration ได้โดยไม่จำเป็นต้องหยุดการทำงาน และเริ่มการทำงานของโปรแกรมใหม่

ส่วนของการวิเคราะห์ข้อมูลของ SWATCH นั้นจะใช้วิธีการวิเคราะห์ข้อมูลล็อกทีละบรรทัด และใช้แนวทางในการตรวจจับการบุกรุกแบบ misuse detection แต่ผู้พัฒนาไม่ได้ให้ตัวอย่างรูปแบบข้อมูลที่จะใช้ในการตรวจจับมา ดังนั้นในการใช้งานจริง ผู้ใช้จึงต้องค้นหาและกำหนดรูปแบบข้อมูลที่ต้องการตรวจจับด้วยตนเอง นอกจากนี้เนื่องจากโปรแกรม SWATCH มีการเรียกใช้งาน Perl library หลายตัว ดังนั้นในการติดตั้งจึงต้องติดตั้งส่วนของ Perl library ที่จำเป็นสำหรับ SWATCH ให้เสร็จก่อนจึงจะสามารถติดตั้งโปรแกรมได้ ซึ่งทำให้การติดตั้งโปรแกรมค่อนข้างยุ่งยากพอสมควร

3.11.2 Logsurfer [Ley and Ellerman, 1996]

Logsurfer เป็นโปรแกรมที่ใช้ในการติดตามตรวจสอบข้อมูลในล็อกไฟล์ พัฒนาโดย Wolfgang Ley and Uwe Ellermanis มีหลักการทำงาน แนวทางในการตรวจจับการบุกรุกและการตอบสนองจะค่อนข้างเหมือนกับ SWATCH แต่ได้ปรับปรุงข้อด้อยบางประการของ SWATCH ให้ดีขึ้น ซึ่งข้อแตกต่างของ Logsurfer จาก SWATCH มีดังนี้

1. ในการเปรียบเทียบข้อมูลล็อกกับรูปแบบข้อมูลที่ต้องการตรวจจับจะใช้ 2 regular expression (ข้อมูลล็อกจะต้องตรงกับ regular expression แรกแต่จะต้องไม่ตรงกับ regular expression ที่สอง) ดังนั้นจึงทำให้สามารถระบุข้อยกเว้นในการตรวจจับได้ ซึ่งจะทำให้ได้ผลการตรวจจับที่ดีกว่า SWATCH
2. ความสามารถในการเก็บข้อมูลล็อกได้หลายบรรทัด (Context) ทำให้สามารถบันทึกข้อมูลการบุกรุกได้ตั้งแต่ต้นจนจบการบุกรุก
3. สามารถปรับเปลี่ยนกฎได้ในขณะที่โปรแกรมกำลังทำงานอยู่
4. พัฒนาด้วยภาษา C ทำให้สามารถทำงานได้หลาย platform และไม่ต้องมีการติดตั้งส่วนของ Perl library

จากความสามารถดังกล่าวข้างต้นจึงทำให้ Logsurfer เป็นเครื่องมือที่มีศักยภาพในการวิเคราะห์ข้อมูลในล็อกไฟล์ที่ดี แต่เนื่องจากปัญหาความยุ่งยากและซับซ้อนในการสร้างข้อกำหนดหรือเงื่อนไขในการวิเคราะห์ข้อมูล ซึ่งส่งผลให้ไม่ค่อยได้รับความนิยมในการใช้งานมากนัก

3.11.3 Logwatch [Bauer, 2003]

Logwatch พัฒนาด้วยภาษา Perl โดย Kirk Bauer โดยทำการวิเคราะห์ข้อมูลของ syslog ในช่วงเวลาที่กำหนด ซึ่งผู้ใช้สามารถกำหนดช่วงเวลาได้ผ่านโปรแกรม cron ซึ่งปกติจะมีการกำหนดไว้ให้วิเคราะห์ทุกวัน และเมื่อวิเคราะห์ข้อมูลในล็อกไฟล์เสร็จแล้วก็จะส่งรายงานการ

วิเคราะห์ทางอีเมล์ไปยังผู้ดูแลระบบ โดยในการวิเคราะห์จะแสดงข้อมูลของเหตุการณ์ที่น่าสงสัยว่าจะเป็นความผิดปกติและตัดข้อมูลในส่วนของเหตุการณ์ปกติออกไป ส่วนข้อมูลล็อกที่ไม่สามารถระบุได้ก็จะจัดกลุ่มเข้าไว้ด้วยกัน นอกจากนี้ผู้ใช้งานสามารถควบคุมระดับความละเอียดของข้อมูลที่จะนำมาแสดงในรายงานได้ ผู้พัฒนามีกฎและข้อกำหนดในการวิเคราะห์มาให้พอมากสมควร แต่การสร้างกฎในการวิเคราะห์ข้อมูลขึ้นมาใหม่ หรือการเพิ่มเติมกฎหรือข้อกำหนดจากเดิมที่มีอยู่ค่อนข้างยุ่งยากและไม่สะดวก จึงทำให้ไม่ค่อยยืดหยุ่นในการทำงาน

3.11.4 **Logcheck** [Rowland, 2003]

Logcheck พัฒนาด้วยใช้ Shell Script และ ภาษา C โดย Craig Rowland การทำงานของ Logcheck มีไฟล์ที่ใช้สำหรับกำหนดข้อความที่บ่งบอกว่าเป็นเหตุการณ์แบบใด ซึ่งประกอบด้วย logcheck.violations , logcheck.hacking, logcheck.alerts และ logcheck.ignore โดยค้นหาและเปรียบเทียบข้อมูลล็อกกับข้อมูลที่กำหนดไว้ในแต่ละไฟล์ข้างต้น หากมีข้อมูลที่ตรงกัน ก็จัดแบ่งข้อมูลล็อกตามเหตุการณ์ที่กำหนดและส่งรายงานผลการทำงานไปยังผู้ดูแลระบบ สำหรับการสั่งให้ Logcheck ทำงานตามช่วงเวลาที่กำหนด เช่น ทุกวัน หรือทุกชั่วโมง เป็นต้น สามารถกำหนดผ่านโปรแกรม cron ซึ่งทำให้การติดตั้งและการใช้งาน logcheck มีความสะดวก และสามารถเพิ่มเติม แก้ไข ข้อกำหนดในการวิเคราะห์ข้อมูลได้ง่าย

3.12 สรุป

สำหรับบทนี้ได้กล่าวถึงรายละเอียดและองค์ประกอบ การจัดหมวดหมู่ จุดเด่นและจุดอ่อนของระบบตรวจจับการบุกรุกแบบต่าง ๆ และแนวทางในการตรวจจับการบุกรุก รวมทั้งความสำคัญของล็อกไฟล์ การวิเคราะห์และโปรแกรมต่าง ๆ ที่ใช้สำหรับการวิเคราะห์ล็อกไฟล์ที่มีอยู่ในปัจจุบัน จากความสำคัญและประโยชน์ของล็อกไฟล์ที่มีต่อระบบตามที่ได้กล่าวมาแล้ว ดังนั้นการจัดทำวิทยานิพนธ์นี้จึงได้เลือกใช้วิธีการวิเคราะห์ล็อกไฟล์มาใช้ในการตรวจจับการบุกรุก โดยใช้ข้อมูลจากล็อกไฟล์ต่าง ๆที่มีอยู่บนระบบปฏิบัติการยูนิกซ์ ซึ่งจะได้กล่าวถึงรายละเอียดของล็อกไฟล์บนระบบยูนิกซ์ รวมทั้งการออกแบบระบบที่จัดทำขึ้นในบทที่ 4