บทที่ 5

สรุปการวิจัย และข้อเสนอแนะ

5.1 บทนำ

สำหรับในบทนี้จะเป็นการกล่าวสรุปผลการวิจัย โดยจะเริ่มจากสรุปการวิจัยในการตรวจจับการบุกรุก การทำงานของระบบตรวจจับการบุกรุก ปัญหาและข้อจำกัดของระบบตรวจจับการบุกรุก และในหัวข้อสุดท้ายเป็นการกล่าวถึงข้อเสนอแนะ

5.2 สรุปการวิจัยระบบตรวจจับการบุกรุก

วิทยานิพนธ์ชิ้นนี้เป็นการพัฒนาระบบตรวจจับการบุกรุก โดยใช้เทคนิคการวิเคราะห์การเปลี่ยนแปลงสถานะและการเรียกใช้ system call บนระบบปฏิบัติการยูนิกซ์ ซึ่งจากการทดสอบระบบการตรวจจับการบุกรุกดังที่ได้กล่าวไว้ในบทที่ 4 นั้นจะเห็นได้ว่าการตรวจจับการบุกรุกด้วยเทคนิคที่ใช้ในวิทยานิพนธ์ให้ผลการทดสอบที่เป็นที่น่าพอใจ นั่นคือสามารถตรวจจับการบุกรุกตามเงื่อนไขที่กำหนดในกฎที่ใช้สนับสนุนการตรวจจับการบุกรุกได้ในทุกกรณี และช่วยลดปัญหาในเรื่องของการตัดสินใจผิดทางบวกได้เป็นอย่างดี และเมื่อมองในส่วนข้อเสียของระบบตรวจจับการบุกรุกประเภท Misuse Detection ในเรื่องของการตรวจจับการบุกรุกใหม่ๆ จากเทคนิคที่ใช้ในงานวิจัยชิ้นนี้ระบบตรวจจับการบุกรุกค่อนข้างจะครอบคลุมการบุกรุกในแง่ที่ผู้บุกรุกพยายามจะทำความเสียหายให้กับระบบ เมื่อผู้บุกรุกอยู่ในสถานะที่มีสิทธิเทียบเท่ากับผู้มีสิทธิสูงสุด ซึ่งได้แก่การบุกรุกประเภทม้าโทรจัน Buffer Overflow รวมทั้งการบุกรุกที่พยายามใช้ช่องโหว่ของระบบเอง อย่างไรก็ตามไม่มีระบบตรวจจับการบุกรุกใดที่สามารถทำงานได้ครอบคลุมประเภทการบุกรุกทุกประเภท หรือกล่าวอีกนัยหนึ่งนั่นคือ ไม่มีระบบตรวจจับการบุกรุกใดที่ทำงานได้อย่างสมบูรณ์แบบ เนื่องจากลักษณะของการบุกรุกมีหลากหลาย เครื่องมือที่ใช้ตรวจจับการบุกรุกได้ถูกพัฒนาขึ้นตามลักษณะการบุกรุกนั้น ๆ อีกทั้งผู้บุกรุกได้พัฒนาเครื่องมือที่ใช้สำหรับการบุกรุกอยู่เสมอเช่นเดียวกัน ในส่วนถัดไปจะกล่าวถึงขั้นตอนการทำงาน การนำระบบไปใช้งานรวมทั้งปัญหาข้อจำกัดและข้อเสนอแนะ

5.3 การใช้งานระบบตรวจจับการบุกรุก

ระบบตรวจจับการบุกรุกที่พัฒนาขึ้นนี้สามารถนำไปใช้ได้จริงบนระบบปฏิบัติการ NetBSD โดยจะต้องทำการแก้ไข system call *ktrace()* สำหรับการเรียกใช้ค่า UID EUID GID และ EGID ของ โปรเซสในขณะนั้นเพื่อนำมาเป็นเงื่อนไขในการพิจารณาสถานะของโปรเซสในการตรวจจับการบุกรุก หลังจากนั้นสามารถนำตัวโปรแกรมไปใช้งานได้ซึ่งประกอบด้วยแฟ้มเหล่านี้ plistall.c readbf.c idstrace.c และ killdetect.c โดยการคอมไพล์สามารถกระทำได้โดยการเรียกใช้ Makefile นอกจากนี้ยังมีแฟ้ม ids.log ซึ่งเป็นแฟ้มที่ใช้ในการบันทึกการบุกรุกซึ่งสามารถเปลี่ยนชื่อแฟ้มได้ตามแต่ผู้ดูแลระบบนั้น ๆ จะกำหนดหลังจากนั้นเมื่อต้องการใช้โปรแกรมให้ทำการแก้ไขแฟ้ม /etc/rc เพื่อเพิ่มบรรทัดให้มีการเรียกใช้โปรแกรม idstrace ให้ทำงานอยู่เบื้องหลังหลังจากที่บริการอื่นของระบบเริ่มทำงานแล้ว สำหรับผลที่ได้จากการตรวจจับใช้เป็นข้อมูลให้กับผู้ดูแลระบบ เมื่อเกิดปัญหาเกี่ยวกับระบบผู้ดูแลระบบสามารถใช้แฟ้มบันทึกข้อมูลการบุกรุกเป็นฐานในการสืบหาสาเหตุของการเกิดปัญหานั้นได้อีกทางหนึ่งถ้าหากปัญหาที่เกิดขึ้นเกิดจากรูปแบบการบุกรุกที่ระบบตรวจจับครอบคลุม

5.4 ปัญหาและข้อจำกัด

จากขั้นตอนการทำงานของระบบตรวจจับการบุกรุกในส่วนของการติดตามโปรเซสเริ่มต้นหรือบริการที่เกิดขึ้นจากโปรเซส init และโปรเซสที่ให้บริการอยู่ในขณะนั้น จะเห็นได้ว่าจำนวนโปรเซสที่ใช้ในการติดตามเพื่อตรวจจับการบุกรุกจะมีจำนวนเท่ากับจำนวนโปรเซสที่ถูกกำหนดให้บริการใน rc สคริปรวมกับโปรเซส init อีกหนึ่งโปรเซส นั่นหมายความว่าถ้าหากผู้ดูแลระบบมีการกำหนดการให้บริการต่าง ๆ บนระบบเป็นจำนวนมากจำนวนโปรเซสที่ใช้ติดตามเพื่อตรวจจับการบุกรุกก็มีจำนวนมากขึ้นด้วยเช่นเดียวกัน นอกจากนี้เนื่องจากระบบตรวจจับการบุกรุกที่พัฒนาขึ้นใช้ข้อมูลจากการติดตามการเรียกใช้ system call ของคำสั่งที่ถูกเรียกใช้ในระบบ ฉะนั้นผลที่ได้จากการติดตามในแต่ละครั้งจะเป็นข้อมูลที่ถูกดำเนินการไปแล้ว ถึงแม้ว่าระบบตรวจจับการบุกรุกสามารถตรวจจับพฤติกรรมที่เข้าข่ายการบุกรุกตามเงื่อนไขที่กำหนดไว้ได้ ในเรื่องของการตอบสนองต่อพฤติกรรมเหล่านั้นจึงเป็นเพียงการยับยั้งพฤติกรรมเสี่ยงที่จะเกิดขึ้นอันเนื่องมาจากพฤติกรรมที่ตรวจจับได้ และบันทึกข้อมูลที่เกิดขึ้นเพื่อผู้จัดการระบบสามารถใช้แก้ไขความเสียหายหรือเป็นแนวทางในการป้องกันต่อไป ซึ่งถ้าหากสามารถที่จะตรวจจับการบุกรุกก่อนที่ system call

ที่ถูกเรียกใช้จะส่งค่ากลับ นั่นคือก่อนที่ system call จะถูกกระทำก็จะช่วยป้องกันพฤติกรรมที่เข้าข่ายการบุกรุกไม่ให้เกิดขึ้นได้

5.5 ข้อเสนอแนะ

ในกรณีที่นำระบบตรวจจับการบุกรุกไปใช้กับระบบคอมพิวเตอร์ที่มีการให้บริการโปรแกรมต่าง ๆ บนระบบเป็นจำนวนมากนั้นหากระบุตำแหน่งของระบบตรวจจับการบุกรุกให้ทำงานหลังจากที่มีการให้บริการต่าง ๆ นั้นจะทำให้มีจำนวนโปรเซสที่จะต้องติดตามตรวจจับการบุกรุกของโปรเซสเหล่านั้นเป็นจำนวนมาก ในขณะที่โปรเซสบางตัวไม่ได้ถูกเรียกใช้เลยหรือถูกเรียกใช้จากผู้ใช้ในระบบไม่บ่อยนัก โปรเซสเหล่านี้ก็ถูกติดตามโดยโปรเซสตรวจจับการบุกรุกด้วยเช่นกัน ฉะนั้นเราอาจจะลดจำนวนโปรเซสที่จะต้องติดตามตรวจจับการบุกรุกได้ดังนี้

วิธีที่ 1 กำหนดให้บริการบางอย่างที่ไม่ถูกเรียกใช้งานบ่อยนัก อาทิเช่น บริการ FTP ถูกระบุไว้ในแฟ้ม inetd.conf เนื่องจากโปรเซส inetd ถูกติดตามด้วยโปรเซสที่ติดตามตรวจจับการบุกรุกอยู่แล้วเมื่อมีการเรียกใช้บริการใดที่ถูกกำหนดในแฟ้ม inetd.conf บริการนั้นก็จะถูกติดตามด้วยโปรเซสที่ติดตาม inetd ตั้งแต่ต้น

วิธีที่ 2 เขียนโปรแกรมโดยกำหนดโค้ดที่ต้องใช้ร่วมกันแบบ shared library

นอกจากนี้ระบบตรวจจับการบุกรุกที่พัฒนาขึ้นนี้สามารถใช้เป็นแนวทางเพื่อพัฒนาระบบรักษาความปลอดภัยที่มีประสิทธิภาพมากขึ้น โดยการใช้เทคนิคเดียวกันนี้ในการปรับปรุงให้เป็นระบบเคอเนลปลอดภัย หรือ secure kernel นั่นคือใช้เทคนิคในการตรวจจับเพื่อแก้ไข system call ที่เกี่ยวข้องกับเงื่อนไขในการตรวจจับอาทิเช่น system call *open()* ให้สามารถป้องกันพฤติกรรมการบุกรุกก่อนที่ system call open() จะส่งค่ากลับ นั่นคือหากพบว่ามีการเรียกใช้ system call open() เพื่อแก้ไขแฟ้มที่ถูกกำหนดว่าเป็น system file หรือ มีการเรียกใช้ system call *open()* เพื่อกำหนดให้แฟ้มเป็นประเภท setuid หรือ setgid ก็อาจจะกำหนดให้ system call *open()* ส่งค่าผิดพลาดกลับพร้อมทั้งส่งข้อความเตือน นั่นคือเมื่อตรวจพบว่ามีการกระทำที่เข้าข่ายการบุกรุก system call ที่เกี่ยวข้องก็จะไม่กระทำตามคำสั่งนั้น