การพัฒนาโปรแกรมตรวจสอบความปลอดภัยพื้นฐาน สำหรับระบบปฏิบัติการลีนุกซ์เรดแฮต

นาย นฤชัย ศรีแสงอยู่

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2547

ISBN : 974-53-1072-7

DEVELOPMENT OF A BASIC SECURITY SCANNING PROGRAM FOR RED HAT LINUX OPERATING SYSTEM

Mr. Naruchai Srisangyu

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computer Science
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2004

ISBN : 974-53-1072-7

หัวข้อวิทยานิพนธ์	การพัฒนาโปรแกรมตรวจสอบความปลอดภัยพื้นฐานสำหรับระบบปฏิบัติการลีนุกซ์เรดแฮต
โดย	นายนฤชัย ศรีแสงอยู่
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษา	อาจารย์ ดร.เศรษฐา ปานงาม
อาจารย์ที่ปรึกษา (ร่วม)	อาจารย์ ธงชัย โรจน์กังสดาล

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดี คณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร. ดิเรก ลาวัณย์ศิริ)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการสอบ
(อาจารย์ ดร.ณัฐวุฒิ หนูไพโรจน์)

..... อาจารย์ที่ปรึกษา
(อาจารย์ ดร.เศรษฐา ปานงาม)

..... อาจารย์ที่ปรึกษา (ร่วม)
(อาจารย์ ธงชัย โรจน์กังสดาล)

..... กรรมการ
(อาจารย์ ดร.ชัย พงศ์พันธ์ภาณี)

นฤชัย ศรีแสงอยู่ : การพัฒนาโปรแกรมตรวจสอบความปลอดภัยพื้นฐานสำหรับระบบปฏิบัติการลีนุกซ์เรดแฮต (Development of A Basic Security Scanning Program for Red Hat Linux Operating System) อ. ที่ปรึกษา: อาจารย์ ดร. เศรษฐา ปานงาม, อ. ที่ปรึกษาร่วม: อาจารย์ ธงชัย โรจน์กังสดาล, 108 หน้า. ISBN 974-53-1072-7.

ในปัจจุบันมีการนำระบบปฏิบัติการลีนุกซ์ตระกูลเรดแฮต มาใช้งานในองค์กรอย่างแพร่หลาย ซึ่งโดยปกติลีนุกซ์จะมีโปรแกรมช่วยจัดการระบบอยู่ในชุดเดียวกับระบบปฏิบัติการ ซึ่งทำให้ผู้ดูแลระบบสามารถทำงานได้ง่ายขึ้น แต่ยังไม่มีโปรแกรม หรือเครื่องมือที่จะช่วยตรวจสอบในเรื่องความปลอดภัยโดยตรงให้กับผู้ดูแลระบบ ผู้ดูแลระบบจำเป็นที่จะต้องใช้ประสบการณ์ส่วนตัว ในการตรวจสอบหาจุดหละหลวมในระบบของตน แต่เนื่องจากระบบลีนุกซ์เป็นระบบปฏิบัติการที่มีความสลับซับซ้อน ผู้ดูแลระบบจึงอาจมองข้ามจุดบางจุด โดยเฉพาะผู้ที่ยังไม่มีประสบการณ์เพียงพอ เพื่อเป็นการลดปัญหาดังกล่าว การวิจัยนี้จึงพัฒนาโปรแกรมตรวจสอบความปลอดภัยพื้นฐานบนระบบ ปฏิบัติการลีนุกซ์ตระกูลเรดแฮต เพื่อช่วยแบ่งเบาภาระของผู้ดูแลระบบในการตรวจสอบจุดหละหลวมในระบบให้มีประสิทธิภาพและมีความสะดวกเพิ่มขึ้น

สำหรับเครื่องมือที่ใช้ในการวิจัย ผู้วิจัยใช้ภาษาเพิร์ลในการพัฒนาโปรแกรมการตรวจสอบและใช้เพิร์ลทีเค ซึ่งเป็นเครื่องมือของภาษาเพิร์ลในการพัฒนาส่วนติดต่อกับผู้ใช้ โดยแยกเป็นหมวดของผู้ใช้งานและหมวดผู้ดูแลระบบ โดยเริ่มต้นโปรแกรมตรวจสอบจะมีการทำงานในการตรวจสอบแยกตามฟังก์ชันได้แก่ ตรวจสอบบิตอนุญาตของแฟ้มข้อมูลและไดเรกทอรี ตรวจสอบระบบบัญชีผู้ใช้และรหัสผ่าน ตรวจสอบระบบทีซีพีไอพี และบริการที่เกี่ยวข้อง ได้แก่ บริการเว็บเซิร์ฟเวอร์ บริการโอนแฟ้มข้อมูล และบริการเมล์เซิร์ฟเวอร์ โดยการตรวจสอบความถูกต้องจะอิงจากเอกสารตรวจสอบความปลอดภัยบนยูนิกซ์เวอร์ชันสองของหน่วยงานเซิร์ต และหน่วยงานเอยูเอสเซิร์ต ในการแสดงผลโปรแกรมสามารถแสดงผลในโหมดภาษาไทยและภาษาอังกฤษตามเวอร์ชันของลีนุกซ์ที่ติดตั้งในระบบ นอกจากนี้โปรแกรมตรวจสอบสามารถทำงานในแบบเบื้องหลังเพื่อเพิ่มความสามารถในการตรวจสอบเมื่อพบปัญหาก็จะทำการแจ้งเมล์ไปยังผู้ดูแลระบบ ซึ่งเป็นประโยชน์สำหรับผู้ดูแลระบบในการป้องกันหรือแก้ไขปัญหาที่จะเกิดขึ้นต่อไปในระบบ

ภาควิชา วิศวกรรมคอมพิวเตอร์	ลายมือชื่อนิสิต.....
สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์	ลายมือชื่ออาจารย์ที่ปรึกษา.....
ปีการศึกษา 2547	ลายมือชื่ออาจารย์ที่ปรึกษาร่วม.....

4471422321 : MAJOR COMPUTER SCIENCE

KEY WORD: LINUX / SECURITY / PROGRAM / SCANNING / REDHAT

NARUCHAI SRISANGYU : DEVELOPMENT OF A BASIC SECURITY SCANNING PROGRAM FOR RED HAT LINUX OPERATING SYSTEM. THESIS ADVISOR : SETHA PAN-NGUM,Ph.D, THESIS COADVISOR : THONGCHAI ROJKANGSADAN,M.Sc, 108 pp. ISBN : 974-53-1072-7

Nowadays, RedHat Platform of Linux Operating System is chosen widely as an operating system in many organizations. The management system in Linux is easy to use for system administrator. However, there is no security checking tool that help user directly. An experienced user might be able to identify any security holes in their systems. Anyhow, Linux operating system is so complicated that less experience users might not be able to. This research is therefore to develop a program to help user to efficiently identify security holes as well as make the system more convenient to use.

Perl Programming Language and Perl TK development tool is used for the research. Perl TK is used to develop a program that connects users which are separated into two types system users and system administrator. At the beginning, the program contains several functions to check out the system such as function to check permission of file and directory, function to check user ID and password, function to check TCP/IP setting and other services such as web server, file transfer, and mail server. The reliability check of the programs is based on Unix Security Checklist v.2.0 of CERT and AusCERT. The result of the operation is displayed according to the language setting from Linux version installed. Furthermore, the program can be run as a background process which will increase its checking capability. When problems occur, the program will automatically e-mail the problems to corresponding users to help protect the system and solve further problems.

Department of Computer Engineering	Student's……
Field of study Computer Science	Advisor's……
Academic year 2004	Co-advisor's……

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยความช่วยเหลืออย่างดียิ่ง ของอาจารย์ ดร. เศรษฐา ปานงาม อาจารย์ที่ปรึกษาวิทยานิพนธ์ และอาจารย์ธงชัย โรจน์กังสดาล อาจารย์ที่ปรึกษาร่วมวิทยานิพนธ์ ซึ่งท่านได้ให้คำแนะนำและข้อคิดเห็นต่างๆ ที่เป็นประโยชน์เป็นอย่างยิ่งแก่ผู้วิจัยและได้ช่วยกรุณาตรวจสอบแก้ไขวิทยานิพนธ์ฉบับนี้ ทำให้มีความถูกต้องและสมบูรณ์มากที่สุด ผู้วิจัยขอขอบพระคุณในความกรุณาเป็นอย่างสูง

ขอขอบพระคุณ ท่านคณะกรรมการสอบวิทยานิพนธ์ ที่ได้ช่วยพิจารณาให้คำแนะนำ ตรวจทาน แก้ไข วิทยานิพนธ์ฉบับนี้

ขอขอบคุณเพื่อนๆ สาขาวิทยาศาสตร์คอมพิวเตอร์ทุกท่านที่ได้ให้ความช่วยเหลือและเป็นกำลังใจแก่ผู้วิจัยตลอดมา

ขอขอบคุณพี่ๆและน้องๆ บริษัทโปรลายน์ทุกท่านที่ได้ให้ความช่วยเหลือในเรื่องเวลาในการทำวิจัย

ท้ายสุดนี้ ผู้วิจัยใคร่กราบขอบพระคุณ บิดา-มารดา ซึ่งเป็นผู้ที่มีพระคุณแก่ผู้วิจัยอย่างหาที่เปรียบมิได้ ซึ่งคอยให้กำลังใจและเชื่อมั่นในตัวผู้วิจัยมาโดยตลอด

สารบัญ

หน้า

บทที่ หน้า

บทที่ หน้า

บทที่ หน้า

สารบัญตาราง

หน้า

สารบัญภาพ

หน้า

หน้า

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ยูนิกซ์ (UNIX) เป็นระบบปฏิบัติการ (Operating System) ที่ถูกคิดค้นโดย เคน ทอมป์สัน (Ken Thompson) และเดนนิส ริทชี (Dennis Ritchie) แห่งบริษัทเอทีแอนด์ที (AT&T) หลังจากนั้นก็ได้มีผู้นำยูนิกซ์ไปติดตั้ง (port) ไปยังเครื่องคอมพิวเตอร์ต่างๆ ทำให้ยูนิกซ์ เป็นที่รู้จักกันอย่างแพร่หลาย

ในปัจจุบันระบบปฏิบัติการยูนิกซ์ ได้แตกออกมาเป็นอีกหลายตระกูลด้วยกัน หนึ่งในนั้นได้แก่ ยูนิกซ์ตระกูลลีนุกซ์ (Linux) ลีนุกซ์ถือกำเนิดขึ้นโดย ลีนุส ทอร์วัลด์ส (Linus Torvalds) เนื่องจากเป็นลักษณะของโอเพนซอส (Open source) จึงไม่เป็นปัญหาเรื่องลิขสิทธิ์ในการใช้งาน อีกทั้งสามารถแก้ไขโปรแกรมของลีนุกซ์ ให้สามารถใช้งานได้บนตัวประมวลผลกลางได้หลากหลาย ตั้งแต่ อินเทล โมโตโรลา ดิจิตอลอัลฟา พาวเวอร์พีซี ไปจนถึง สปาร์คของซัน อีกทั้งยังมีโปรแกรมประยุกต์มากมายที่สามารถทำงาน บนระบบปฏิบัติการลีนุกซ์ ด้วยเหตุนี้ ลีนุกซ์จึงเป็นยูนิกซ์ ที่ได้รับความนิยมอย่างมากในปัจจุบัน

เนื่องจากผู้ออกแบบลีนุกซ์เป็นโปรแกรมเมอร์ จุดประสงค์ของลีนุกซ์เริ่มแรก คือเพื่อให้โปรแกรมเมอร์ใช้ลีนุกซ์เป็นสภาพแวดล้อมในการพัฒนาโปรแกรมบนเครื่องพีซี และเพื่อให้ใช้งานได้โดยสะดวก เนื่องจากระบบมินิกซ์ (Minix) ที่เป็นยูนิกซ์บนพีซีขณะนั้น ยังมีความสามารถไม่เพียงพอแก่ความต้องการ ดังนั้นผู้ออกแบบจึงไม่ได้คำนึงถึงระบบความปลอดภัย (Security) มากนัก ซึ่งต่อมาได้มีนักพัฒนามาช่วยทำการพัฒนาลีนุกซ์ให้มีความสามารถมากขึ้นในด้านต่างๆ แต่ในส่วนของความปลอดภัยยังไม่ได้รับการพัฒนาเพิ่มเติมเท่าที่ควร แม้โดยรวมลีนุกซ์จะมีระบบความปลอดภัยที่ดี แต่ทั้งนี้ก็ขึ้นอยู่กับผู้ดูแลระบบว่ามีความเข้าใจในระบบมากน้อยเพียงใด ซึ่งถ้ามีความเข้าใจในระบบทั้งหมด ก็สามารถที่จะบริหารจัดการปิดจุดอ่อนของระบบได้ ในทางตรงข้ามถ้าหากไม่มีความเข้าใจในระบบดีพอ ระบบปฏิบัติการลีนุกซ์ก็จะเป็นระบบปฏิบัติการที่มีจุดอ่อนในด้านความปลอดภัย ซึ่งในทางปฏิบัติเป็นการยากที่ผู้ดูแลจะมีความเข้าใจในระบบลีนุกซ์ทั้งหมด อีกทั้งในปัจจุบันเทคโนโลยีทางด้านเครือข่าย (Network) ได้เจริญก้าวหน้ามากขึ้น จึงได้มีผู้พัฒนา โปรแกรมเพื่อให้ระบบลีนุกซ์สามารถเป็นเซิร์ฟเวอร์ ให้บริการทางด้านต่างๆ เช่น เมล์เซิร์ฟเวอร์ (Mail Server) อินเตอร์เน็ตเซิร์ฟเวอร์ (Internet Server) หรือเว็บเซิร์ฟเวอร์ แฟ้มเซิร์ฟเวอร์ (File Server) การใช้เครื่องในระยะไกล (Remote Login) และส่งแฟ้มระหว่างเครื่อง

(File Transfer) ทำให้ระบบไม่ได้เป็นระบบปิดแบบสมัยก่อน การควบคุมด้านความปลอดภัยยิ่งต้องเพิ่มความระมัดระวังมากขึ้น ยิ่งกว่านั้น ปัจจุบันเครือข่ายอินเตอร์เน็ตได้เข้ามามีบทบาทสำคัญในการเชื่อมต่อคอมพิวเตอร์ทั่วโลกให้สามารถติดต่อสื่อสารเชื่อมโยงข้อมูลถึงกันได้ง่ายดาย ยิ่งทำให้มีโอกาสหรือช่องทางที่จะมีผู้บุกรุก ทำให้เกิดความล่อแหลมด้านความปลอดภัยต่อระบบได้ ซึ่งสิ่งเหล่านี้ทำให้การดูแลทางด้านความปลอดภัยของระบบลีนุกซ์ทำได้ยากและซับซ้อนยิ่งขึ้น

สำหรับดิสตริบิวเตอร์ (Distributor) ของลีนุกซ์ในปัจจุบันที่มีชื่อเสียงและได้รับความนิยมในการใช้งานก็คือ เรดแฮต เนื่องจากมีอินเตอร์เฟสที่ใช้งานค่อนข้างง่ายและสามารถนำไปติดตั้งบนฮาร์ดแวร์ที่หลากหลาย ตั้งแต่เครื่องระดับไมโครคอมพิวเตอร์จนถึงเครื่องระดับมินิเมนเฟรม ทำให้เรดแฮตเป็นลีนุกซ์ที่ผู้จัดการระบบคอมพิวเตอร์เลือกใช้งานในหน่วยงาน หรือองค์กรของตนเองมากที่สุดในปัจจุบัน

โปรแกรมวิเคราะห์ความปลอดภัยสำหรับลีนุกซ์เรดแฮต จะช่วยแบ่งเบาภาระของผู้จัดการระบบ ทำให้ผู้จัดการระบบทราบถึงปัญหาที่เกิดขึ้น และสามารถแก้ไขหรือป้องกันปัญหาอันจะเกิดขึ้นได้

1.2 วัตถุประสงค์ของการวิจัย

เพื่อพัฒนาโปรแกรมตรวจสอบความปลอดภัยพื้นฐานสำหรับระบบปฏิบัติการลีนุกซ์เรดแฮต

1.3 ขอบเขตของการวิจัย

1.3.1 โปรแกรมตรวจสอบความปลอดภัยพื้นฐานสำหรับระบบปฏิบัติการลีนุกซ์เรดแฮต สามารถที่จะนำมาใช้งานได้บน ระบบปฏิบัติการลีนุกซ์ ตระกูลเรดแฮต ซึ่งใช้ เคอร์เนล (kernel) รุ่น 2.4.18

1.3.2 โปรแกรมช่วยตรวจสอบระบบความปลอดภัยจะแสดงข้อความ (Message) ให้ทราบถึงส่วนที่หละหลวม หรือจุดที่น่าสงสัย โดยโปรแกรมสามารถที่จะแก้ไขปัญหาที่เกิดขึ้นบางอย่าง หรือปัญหาเบื้องต้นได้ โดยมีข้อจำกัดว่าปัญหาที่เกิดขึ้นนั้น ไม่ใช่ปัญหาที่เกิดจากเคอร์เนลหรือปัญหาที่จะต้องมีการแก้ไขเคอร์เนล ซึ่งกรณีที่โปรแกรมไม่สามารถแก้ไขปัญหาที่เกิดขึ้นได้โปรแกรมจะแนะนำวิธีการแก้ไข และให้ผู้ใช้แก้ไขด้วยตนเอง

1.3.3 โปรแกรมสามารถบันทึกล็อก (Log) ของการทำงานที่เกิดขึ้นได้ โดยเมื่อผู้ใช้งานมีการใช้งานโปรแกรมช่วยตรวจสอบระบบความปลอดภัย หรือเมื่อโปรแกรมพบปัญหาและมีการแก้ไขปัญหาที่พบ โปรแกรมจะทำการเขียนกิจกรรมที่ทำ บันทึกลงสู่แฟ้มข้อมูล

1.3.4 ภาษาที่ใช้ในการพัฒนาคือภาษาเพิร์ล (Perl) และโปรแกรมอรรถประโยชน์ (Utility program) ต่างๆ ที่มีอยู่ในระบบยูนิกซ์

1.3.5 ผู้ใช้งาน สามารถที่จะปรับแต่งส่วนติดต่อผู้ใช้ (User Interface) ให้เหมาะสมตามความต้องการของผู้ใช้แต่ละคน

1.3.6 ตัวโปรแกรมมีส่วนที่มีการทำงานเป็นลักษณะดีมอน (Daemon) มีการตรวจจับปัญหาที่เกิดขึ้น กรณีที่มีปัญหาเกิดขึ้นจะมีเมล์แจ้งไปยังผู้ดูแลระบบ โดยสามารถระบุว่า ให้การทำงานเป็นแบบดีมอน ในโมดูลใดบ้าง

1.3.7 โปรแกรมสามารถที่จะแสดงผลเป็นภาษาไทยโดยโปรแกรมจะตรวจสอบจากระบบปฏิบัติการที่ติดตั้งว่ามีฟอนต์ภาษาไทยติดตั้งอยู่ในระบบปฏิบัติการอยู่หรือไม่ ถ้ามีก็จะสามารถแสดงผลอยู่ในรูปแบบภาษาไทย

1.3.8 โปรแกรมสามารถควบคุมระดับการใช้งานของเป็น สอง ระดับ โดยยกเลิกฟังก์ชัน การทำงานบางอย่าง สำหรับผู้ใช้งานที่เป็นผู้ใช้งานธรรมดาและ เปิดฟังก์ชันการทำงานทั้งหมด กับผู้ใช้งาน ที่เป็นระดับผู้ดูแลระบบ (super user) โดยมีรหัสผ่านในการตรวจสอบ ก่อนเข้าในฟังก์ชันการทำงานสำหรับผู้ดูแลระบบ

1.3.9 ส่วนติดต่อผู้ใช้ ใช้โปรแกรมเพิร์ลทีเค (Perl Tk) ในการพัฒนา

1.3.10 ขอบเขตของการตรวจสอบความปลอดภัย ในระบบลีนุกซ์เรดแฮต โดยสามารถตรวจสอบบิตอนุญาต (Permission) ระบบแฟ้มข้อมูล และไดเรกทอรี ตรวจสอบ ระบบทีซีพี ไอพี (TCP/IP) และเซอร์วิส (Service) ที่เกี่ยวข้อง ได้แก่บริการเว็บเซิร์เวอร์ บริการเอฟทีพีเซิร์เวอร์ และบริการเมล์เซิร์เวอร์ เป็นขั้นต่ำ โดยเป็นโปรแกรมที่มาพร้อมกับชุดลีนุกซ์เรดแฮต ได้แก่ โปรแกรมอาปาเช (Apache) และ โปรแกรมเซนเมล์ (Send mail) โดยการตรวจสอบความถูกต้องจะอิงจากเอกสารตรวจสอบความปลอดภัยบนยูนิกซ์เวอร์ชันสอง (Unix Security Checklist v.2.0) ของหน่วยงานเซิร์ต (Computer Emergency Response Team) และหน่วยงานเอยูเอสเซิร์ต (Australian Computer Emergency Resonses Team)

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 ทำให้ผู้ดูแลระบบตระหนักถึงปัญหาของระบบความปลอดภัยในระบบปฏิบัติการลีนุกซ์หรือเครือข่ายที่ตัวเองรับผิดชอบ

1.4.2 ช่วยแบ่งเบาภาระของผู้ดูแลระบบในการตรวจสอบความปลอดภัยทำให้ผู้ดูแลระบบสามารถทราบถึงจุดที่เกิดปัญหา และสามารถแก้ไขได้ทันท่วงที

1.4.3 สามารถใช้เป็นแนวทางในการพัฒนาระบบตรวจสอบความปลอดภัยสำหรับลีนุกซ์ ที่มีความสามารถยิ่งขึ้น

1.5 วิธีดำเนินการวิจัย

1.5.1 ศึกษาโครงสร้างระบบแฟ้มข้อมูลและไดเรกทอรี ที่สำคัญของยูนิกซ์ในด้านความปลอดภัย

1.5.2 ศึกษาโครงสร้าง ระบบโปรโตคอล (Protocol) ทีซีพีไอพี และบริการ (Service) ต่างๆ ที่จะตรวจสอบความปลอดภัย

1.5.3 ศึกษาการเขียนโปรแกรมด้วยภาษาเพิร์ล

1.5.4 ศึกษาการใช้งานโปรแกรมเพิร์ลทีเค

1.5.5 ออกแบบตัวเชื่อมโยงผู้ใช้และ โมดูลของโปรแกรมต่างๆ

1.5.6 พัฒนาโปรแกรม

1.5.7 ทดสอบโปรแกรม

1.5.8 ปรับแก้ไขโปรแกรมเพื่อให้มีประสิทธิภาพมากขึ้น

1.5.9 สรุปผลการวิจัยและ ข้อเสนอแนะ

บทที่ 2
แนวคิดและทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ความปลอดภัยพื้นฐานในระบบยูนิกซ์

โดยปกติผู้ใช้งานทุกคนต้องการความเป็นส่วนตัวในการทำงาน ซึ่งไม่ต้องการให้ผู้อื่นรับรู้ เนื่องจากยูนิกซ์เป็นระบบปฏิบัติการแบบผู้ใช้หลายคน (Multi User) ถ้าผู้หนึ่งผู้ใดสามารถเข้าไปแทรกแซงในการทำงานของผู้อื่นแล้วย่อมเกิดความไม่ปลอดภัย หรือการกระทำของบุคคลใดบุคคลหนึ่งแล้วส่งผลกระทบถึงผู้อื่นก็เช่นกัน ดังนั้นระบบปฏิบัติการประเภทผู้ใช้หลายคนอย่างยูนิกซ์จึงจำเป็นต้องมีมาตรการในการรักษาความปลอดภัยพื้นฐานในระบบ

2.1.1 จุดประสงค์การรักษาความปลอดภัย [9]

ในการรักษาความปลอดภัย โดยพื้นฐานแล้วในระบบยูนิกซ์จะเป็นการรักษาความปลอดภัยของข้อมูลเป็นหลัก เนื่องจากเป็นระบบปฏิบัติการที่มองทุกสิ่งในลักษณะแฟ้มข้อมูลโดยมีจุดประสงค์หลักของการรักษาความปลอดภัยทางข้อมูลคือ

2.1.1.1 รักษาความลับ (Confidentiality) ซึ่งรวมถึงทุกสิ่งทุกอย่างในระบบ เช่น อุปกรณ์ต่าง ๆ หรือบรรดาแฟ้มข้อมูลจะสามารถเข้าถึงได้เฉพาะบุคคลที่ได้รับสิทธิเท่านั้น

2.1.1.2 รักษาการบูรณภาพ (Integrity) ข้อมูลต่างๆ จะมีการแก้ไขหรือเปลี่ยนแปลงได้เพียงบุคคลที่ได้รับสิทธิเท่านั้น ผู้อื่นจะทำการเปลี่ยนแปลงไม่ได้

2.1.1.3 รักษาสภาพพร้อมใช้งาน (Availability) ข้อมูลและอุปกรณ์ต่าง ๆ ภายในระบบจะต้องเข้าถึงได้เสมอเมื่อต้องการใช้ โดยมีเงื่อนไขว่าผู้ใช้จะต้องเป็นผู้มีสิทธิในอุปกรณ์หรือข้อมูลต่าง ๆ นั้น

2.1.2 สิ่งที่ต้องคำนึงการออกแบบการรักษาความปลอดภัยพื้นฐาน

การออกแบบการรักษาความปลอดภัยพื้นฐานเป็นสิ่งที่ยาก เพราะหากมีการควบคุมอย่างละเอียดมากเกินไป การทำงานจะต้องใช้ความระมัดระวังอย่างมาก และเกิดความไม่สะดวกในการทำงาน ระบบการรักษาความปลอดภัยข้อมูลที่ดี นอกจากจะต้องคำนึงถึงหลักการต่าง ๆ แล้ว สิ่งที่จะมองข้ามไม่ได้คือ ความรู้สึกของผู้ใช้งานระบบโดยมีหลักการคือ

2.1.2.1 ระบบรักษาความปลอดภัยต้องง่ายและตรงไปตรงมา

2.1.2.2 ให้สิทธิกับผู้ใช้น้อยที่สุดเท่าที่จะทำงานได้

2.1.2.3 ทำการตรวจสอบระบบอย่างรอบคอบในการเข้าถึงหรือการทำงานทุกอย่าง

2.1.2.4 กำหนดสิทธิแบบเจาะจงว่ามีสิทธิใดบ้างในข้อมูลและอุปกรณ์ต่าง ๆ

2.1.2.5 กลไกการรักษาความปลอดภัยต้องใช้งานง่าย

2.1.3 บุคคลที่เสี่ยงต่อการทำลายข้อมูล

2.1.3.1 ผู้ใช้งานภายในระบบที่มีความรู้ไม่เพียงพอ บุคคลในกลุ่มนี้จัดว่าอันตรายมาก และส่วนใหญ่จะนึกไม่ถึง อันตรายที่จะเกิดจากกลุ่มนี้ส่วนใหญ่เป็นการกระทำที่ไม่มีเจตนาร้าย แต่จะเกิดจากการใช้คำสั่งโดยรู้เท่าไม่ถึงการณ์ เช่น ใช้คำสั่ง “rm” ลบไฟล์สำคัญ ๆ ไป

2.1.3.2 ผู้ใช้งานที่ต้องการศึกษาการทำงานของระบบ (Hacker) บุคคลกลุ่มนี้ส่วนใหญ่จะเป็นนักคอมพิวเตอร์ที่มีความสามารถ ชอบศึกษาระบบและเจาะระบบที่มีการรักษาความปลอดภัย โดยส่วนใหญ่แล้วบุคคลกลุ่มนี้ เมื่อสามารถเจาะเข้าระบบได้แล้วก็จะเพียงประกาศจุดอ่อนของระบบให้ผู้ดูแลระบบนั้นทราบ หรือไม่ก็อาจจะนำโปรแกรมที่สามารถทำให้ประสิทธิภาพโดยรวมของระบบลดลง เช่น โปรแกรมลักษณะลูป เพื่อทำให้ระบบทำงานผิดปกติในช่วงเวลาหนึ่ง แต่ก็มีไม่น้อยที่เข้าไปแล้วทำลายข้อมูล บุคคลกลุ่มนี้การป้องกันจะทำได้ยากกว่ากลุ่มอื่น

2.1.3.3 กลุ่มสอดแนมล้วงความลับข้อมูล (Cracker) บุคคลกลุ่มนี้จะมีเจตนาร้ายโดยจะล้วงความลับข้อมูล โดยเฉพาะอย่างยิ่งข้อมูลทางธุรกิจ ซึ่งถือว่าเป็นความลับสุดยอดของแต่ละบริษัท

2.1.3.4 ผู้ใช้ที่ต้องการทำลายเอง มีจำนวนไม่น้อยที่ปัญหาเกี่ยวกับการรักษาความปลอดภัยข้อมูล เกิดจากผู้ใช้ภายในระบบเองที่อาจจะไม่พอใจอะไรบางอย่าง เช่น ไม่พอใจความยุ่งยากของมาตรการต่าง ๆ เลยหาทางออกโดยการทำลายระบบรักษาความปลอดภัยของข้อมูลเอง

2.1.4 กลวิธีพื้นฐานการเข้าทำลายข้อมูล [9]

กลวิธีพื้นฐานที่นิยมใช้ได้แก่

2.1.4.1 แอบดูรหัสผ่านขณะพิมพ์ ซึ่งส่วนใหญ่จะเป็นบุคคลที่รู้จักกัน

2.1.4.2 ใช้เครื่องมือวัดที่เรียกว่า เครื่องวิเคราะห์โปรโตคอล (Protocol Analyzer) ไปดักจับทางสายสื่อสารข้อมูลและแพ็คเกต ในกรณีที่อินเตอร์เฟสอยู่ในโหมดการทำงานแบบไม่เลือก

2.1.4.3 ม้าโทรจัน (Trojan horse) เป็นการเขียนโปรแกรมหรือสคริปต์ โดยจะทำการสร้างจอภาพให้เหมือนกับจอภาพตอนเริ่ม (login) เข้าสู่ระบบทุกประการ โดยโปรแกรมหรือ

สคริปต์นั้น จะรอให้ผู้ใช้งานที่ไม่ระวังพิมพ์ชื่อ และรหัสผ่านเข้าสู่โปรแกรมแล้วทำการส่งผ่านชื่อและรหัสผ่านของผู้ใช้ไปยังผู้ไม่หวังดี หรืออาจจะเก็บในชื่อแฟ้มใดแฟ้มหนึ่งก็ได้ ซึ่งไม่เป็นที่สะดุดตา หลังจากนั้นโปรแกรมก็จะเรียกเชลล์ของ login แท้จริงออกมา เช่น การสร้างคำสั่ง su ปลอมขึ้นมา การทำงานของสคริปต์หรือโปรแกรมนี้เกิดขึ้นเนื่องจากลำดับการค้นหาในไดเรกทอรี โดยได้กำหนดการค้นหาที่ไดเรกทอรีบ้านของตนเองก่อน ไดเรกทอรีคำสั่งของระบบ เช่น กำหนด PATH เป็น :/bin:/usr/bin แทนที่จะเป็น /bin:/usr/bin:: ในแฟ้ม .profile ในบางกรณีม้าโทรจันใช้วิธีเปิดช่องทางสื่อสารทิ้งไว้ เพื่อติดต่อเข้ามาผ่านทางช่องสื่อสารนี้ภายหลัง

2.1.4.4 ประตูกล (trap door) เป็นการเขียนโปรแกรมเล็ก ๆ ขึ้นมาทิ้งไว้ โดยโปรแกรมนี้จะอาศัยหลักการอยู่ที่การเปลี่ยนค่าบิต setuid เมื่อไหร่ที่บิตนี้ทำงานโปรแกรมจะทำการเปลี่ยนค่าได้ user id เป็น 0 คือ รูท (root) ชั่วคราว แล้วหากเขียนโปรแกรมที่สามารถสร้างเชลล์ขึ้นมาใหม่ในขณะนั้นได้ก็จะมีสิทธิเท่าเทียม กับรูทซึ่งอาจเกิดความไม่ปลอดภัย

2.2 การตรวจสอบความปลอดภัยพื้นฐานในระบบยูนิกซ์

การรักษาความปลอดภัยพื้นฐานในระบบนั้น สามารถที่จะกำหนดการตรวจสอบเพื่อป้องกันควบคุมการรักษาความปลอดภัย โดยสามารถแบ่งการตรวจสอบออกเป็นสองประเภทคือ

2.2.1 การตรวจสอบภายนอกระบบ (PhysicalSecurity) เป็นการตรวจสอบทางกายภาพ โดยไม่ให้ผู้ที่จะมาทำลายข้อมูลกระทำการได้ เช่น

2.2.1.1 ระมัดระวังบุคคลแปลกปลอมที่จะเข้ามาใช้งานระบบ โดยอาจจะมีการจัดเวรยามคอยดูแล

2.2.1.2 ทำการล็อกอุปกรณ์ที่ใช้งานในระบบ

2.2.1.3 เก็บข้อมูลที่มีความสำคัญลงเทปหรือดิสค์แล้วลบข้อมูลในระบบออก

2.2.2 การตรวจสอบภายในระบบ (Logical Security) เป็นการป้องกันภายในตัวระบบเอง โดยจะแบ่งได้เป็น 2 ประการหลัก ๆ คือ

2.2.2.1 ป้องกันเกี่ยวกับการตรวจสอบผู้ใช้งาน

2.2.2.2 ป้องกันเกี่ยวกับระบบแฟ้มข้อมูล

โดยสองประการนี้ เป็นส่วนการป้องกันความปลอดภัยพื้นฐานที่สำคัญในระบบปฏิบัติการประเภทยูนิกซ์ ซึ่งผู้ใช้งานในระบบยังสามารถแบ่งออกเป็นผู้ใช้ประเภทต่างๆ เพื่อแบ่งสิทธิในการใช้งานระบบ ทั้งยังมีในส่วนของกลไกการอารักขาแฟ้มข้อมูล ซึ่งจะช่วยป้องกันความปลอดภัยในการเข้าถึงแฟ้มข้อมูลในระบบ สามารถอธิบายรายละเอียดได้ในหัวข้อโครงสร้างพื้นฐานระบบความปลอดภัยของยูนิกซ์

2.3 โครงสร้างระบบความปลอดภัยของยูนิกซ์

2.3.1 การตรวจสอบผู้ใช้ (User Authentication)

ระบบยูนิกซ์ใช้รหัสผ่าน(Password) ในการตรวจสอบผู้ใช้แต่ละคนหลังจากที่ผู้ใช้เลือกรหัสผ่านของตัวเองแล้ว ระบบยูนิกซ์จะให้ตัวเลขสุ่ม (random number) 12 บิต ที่เรียกว่าซอลต์ (salt) ซึ่งจะนำมาต่อกับรหัสผ่านที่ผู้ใช้เลือก จากนั้นยูนิกซ์จะทำการเข้ารหัสลับ (encrypt) ซึ่งผลลัพธ์ของการเข้ารหัส จะได้ตัวเลขสุ่ม 64 บิตเก็บไว้ที่แฟ้มข้อมูล /etc/passwd และแฟ้มข้อมูล /etc/shadow

2.3.2 ประเภทของผู้ใช้ในระบบยูนิกซ์

2.3.2.1 ผู้จัดการระบบ (Super user) ในระบบยูนิกซ์ทุกเครื่องจะมีผู้ใช้พิเศษซึ่งจะปรากฏในแฟ้ม /etc/passwd โดยมีชื่อบัญชีผู้ใช้ว่า รูท (root) ซึ่งจะมีหมายเลขประจำตัว (User Identification) เป็น 0 ในการที่จะเป็นรูทนี้ ผู้ใช้ต้องใส่รหัสผ่านเหมือนผู้ใช้ประเภทอื่นๆ ซึ่งมักเรียกว่ารหัสผ่านรูท (root password) เนื่องจากว่าผู้จัดการระบบมีอำนาจสิทธิสูงสุดในระบบสามารถ ที่จะล้มเลิกสิทธิใดๆ (override permission) ได้ ดังนั้นรหัสผ่านสำหรับการเป็นผู้จัดการระบบนี้ จึงมีความสำคัญที่สุดสำหรับระบบยูนิกซ์ทุกเครื่อง

2.3.2.2 ผู้ใช้ธรรมดา (Ordinary) ผู้ใช้ประเภทนี้จะมีสิทธิเท่าที่ผู้จัดการระบบอนุญาตเท่านั้น เป็นผู้ใช้งานธรรมดาทั่วๆ ไป ผู้ใช้ส่วนใหญ่มักจะจัดอยู่ในประเภทนี้

2.3.2.3 ผู้ใช้พิเศษ (Special user) เพื่อลดอันตรายอันเกิดจากการที่ผู้จัดการระบบมีอำนาจสูงสุด ระบบยูนิกซ์ส่วนใหญ่จะมีชื่อผู้ใช้ที่มีอำนาจพิเศษบางอย่าง เช่น สามารถเข้าถึงแฟ้มหรือไดเรกทอรีบางส่วนของระบบ ส่วนใหญ่ผู้ใช้พิเศษเหล่านี้จะเกี่ยวข้องกับการทำงานของระบบมากกว่าผู้ใช้ที่มีตัวตนจริง ได้แก่ ชื่อผู้ใช้ปริยาย (default user) เช่น ยูยูซีพี (uucp) เมล์ (mail) แอลพี (lp) บิน (bin) หรือผู้ใช้ที่เกิดจากการติดตั้งเซิร์ฟเวอร์ เช่น เวิลด์วายเว็บ (www)

2.3.3 ประเภทของแฟ้มข้อมูลในระบบยูนิกซ์

2.3.3.1 แฟ้มข้อมูลแบบธรรมดา (Ordinary file) คือแฟ้มข้อมูลที่เก็บข้อมูลในลักษณะข้อความ (text) หรืออาจจะอยู่ในรูปที่สามารถกระทำการได้ (executable program)

2.3.3.2 แฟ้มข้อมูลแบบไดเรกทอรี (Directory file) คือแฟ้มข้อมูลที่เก็บรายการของแฟ้มข้อมูลต่างๆ

2.3.3.3 แฟ้มข้อมูลแบบพิเศษ (Special file) คือแฟ้มข้อมูลที่อ้างอิงกับอุปกรณ์ต่างๆ ผู้ใช้สามารถที่จะเขียนหรืออ่านข้อมูลจากแฟ้มแบบนี้ได้ เหมือนกับแฟ้มข้อมูลธรรมดา แต่จะมีการกระทำกับอุปกรณ์ทางกายภาพจริงๆ

2.3.4 กลไกการอารักขาแฟ้มข้อมูล (File Permission Mechanism) [23]

บิตอนุญาตของแฟ้มข้อมูล (File permission bit) ของระบบยูนิกซ์ สามารถแสดงในตารางที่ 2.1

ตารางที่ 2.1 กลุ่มของบิตอนุญาต

-	r	w	x	r	w	x	r	w	x
Type	Owner (user)			Group			Others		

อธิบายในแต่ละส่วนเพิ่มเติมในตารางที่ 2.2 ได้ดังนี้

ตารางที่ 2.2 คำอธิบายเพิ่มเติมบิตอนุญาตในกลุ่มต่างๆ

Description	**คำอธิบาย**
Type	บอกถึงชนิดของแฟ้ม
Owner (user)	สิทธิ์ในการเข้าถึงแฟ้มของเจ้าของแฟ้ม
Group	สิทธิ์ในการเข้าถึงแฟ้มของผู้ใช้ในระบบที่อยู่ในกรุ๊ปที่ระบุไว้
Others	สิทธิ์ในการเข้าถึงแฟ้มของทุกคนในระบบที่ไม่ใช่เจ้าของแฟ้ม และไม่ได้อยู่ในกรุ๊ป เดียวกันกับ Group ที่ระบุไว้

โดยตารางที่ 2.3 จะอธิบายชนิดของแฟ้มที่มีในระบบลีนุกซ์

ตารางที่ 2.3 ชนิดของแฟ้มข้อมูล

ชนิด	**คำอธิบาย**
-	แฟ้มปกติ
d	ไดเรกทอรี
c	ดีไวซ์แบบอักษร (character)
b	ดีไวซ์แบบบล็อก (block)
l	ซิมโบลิคลิงค์ (symbolic link)
s	ซอกเกต (socket)
p	ไฟโฟ (FIFO)

2.3.5 ความแตกต่างระหว่างบิตอนุญาตของแฟ้มกับของไดเรกทอรี

ลีนุกซ์มองทุกสิ่งทุกอย่างในระบบเป็นแฟ้ม ดังนั้นบิตอนุญาตของแฟ้มจึงเป็นเรื่องสำคัญที่สุดต่อการใช้งาน และมักจะพบว่าเป็นความผิดพลาดเบื้องต้น ที่ผู้ดูแลระบบมองข้ามหรือไม่สนใจอยู่บ่อยครั้งทำให้ระบบเกิดจุดอ่อนได้ง่าย ความหมายของบิตอนุญาตหรือสิทธิ์ในการเข้าถึงแฟ้มปกติ ที่ไม่ใช่ไดเรกทอรีแสดงในตารางที่ 2.4 คือ

ตารางที่ 2.4 บิตอนุญาตในการเข้าถึงแฟ้ม

อักษร	บิตอนุญาต	ความหมาย
r	READ	สามารถเปิดแฟ้มและอ่านเนื้อความในแฟ้มได้
w	WRITE	สามารถเพิ่มข้อความ ลบข้อความ หรือเปลี่ยนแปลงเนื้อความในแฟ้มได้
x	EXECUTE	จะใช้ในกรณีที่แฟ้มนี้สามารถทำงานได้หรือสั่งให้ทำงานได้ ซึ่งจะมีสิทธิ์ในการทำงานนี้ปรากฏอยู่

แต่เมื่อนำมาใช้กับไดเรกทอรีจะมีสิทธิ์ในอีกความหมายหนึ่งแสดงในตารางที่ 2.5 ดังนี้

ตารางที่ 2.5 บิตอนุญาต ในการเข้าถึงไดเรกทอรี

อักษร	บิตอนุญาต	ความหมาย
r	READ	สามารถใช้คำสั่ง ls ในการดูแฟ้มภายในไดเรกทอรีได้
w	WRITE	สามารถเพิ่มแฟ้ม ลบแฟ้ม เปลี่ยนชื่อแฟ้มที่อยู่ในไดเรกทอรีได้
x	EXECUTE	สามารถใช้คำสั่ง cd ทำการเปลี่ยนไดเรกทอรีปัจจุบันเข้าไปได้ หรือใช้ในการเปิดแฟ้มภายในไดเรกทอรีรวมไปถึงไดเรกทอรีย่อยภายใน

2.3.6 ความสัมพันธ์ของหมายเลขประจำตัวผู้ใช้ (UID) กับ ชื่อผู้ใช้งาน (User Name) และหมายเลขประจำกลุ่ม (GID) กับ ชื่อกลุ่ม (Group Name)

ในระบบลีนุกซ์ใช้วิธีในการยืนยันตัวตนและสิทธิ์ที่ได้รับของผู้ที่จะเข้ามาในระบบ โดยผ่านกระบวนการล็อกอินโดยใช้ชื่อผู้ใช้งาน เพื่อระบุผู้ใช้ที่จะเข้ามาในระบบและรหัสผ่านของผู้ใช้คนนั้นเพื่อยืนยันการเข้าสู่ระบบ ซึ่งชื่อผู้ใช้งานของผู้ใช้ในสมัยก่อนอนุญาตให้มีชื่อผู้ใช้งานได้ยาวมากที่สุดเพียงแปดตัวอักษร แต่ในปัจจุบันลีนุกซ์อนุญาตให้ใช้ชื่อผู้ใช้งาน ได้ยาวมากที่สุดถึง 256 ตัวอักษร

สมมุติในระบบมีผู้ใช้คนหนึ่งชื่อ sample พิจารณาแฟ้ม /etc/passwd ที่ระบุชื่อผู้ใช้ชื่อ sample

```
sample:x:100:200:Sample account:/home/sample:/bin/bash
```

อธิบายฟิลด์ต่างๆ แสดงในตารางที่ 2.6 ดังนี้(แต่ละฟิลด์คั่นด้วยเครื่องหมาย ':')

ตารางที่ 2.6 รายละเอียดในแฟ้ม /etc/passwd

ฟิลด์	คำอธิบาย
sample	ชื่อผู้ใช้งาน คือ sample
x	เป็นรหัสผ่านซึ่งจะเก็บค่าไว้ในแฟ้ม /etc/shadow
100	เป็น หมายเลขประจำตัวผู้ใช้ ซึ่งลีนุกซ์จะใช้ในการอ้างถึงเจ้าของแฟ้มนั้นๆ
200	อ้างถึงหมายเลขประจำกลุ่ม ซึ่งลีนุกซ์ใช้ในการบอกกรุ๊ปของผู้ใช้คนนี้
Sample account	รายละเอียดอื่นๆ เพิ่มเติมของผู้ใช้ชื่อ sample
/home/sample	เป็นไดเรกทอรีบ้านของผู้ใช้ sample
/bin/bash	เชลล์หลักของผู้ใช้

ผู้ใช้ที่ชื่อ sample อยู่ในกลุ่ม staff พิจารณาแฟ้ม /etc/group รายละเอียดคือ "staff:x:200:"

อธิบายถึงฟิลด์ต่างๆ แสดงในตารางที่ 2.7 ดังนี้

ตารางที่ 2.7 รายละเอียดของแฟ้ม /etc/groups

ฟิลด์	คำอธิบาย
staff	กรุ๊ปชื่อ staff
x	ใช้ในการระบุรหัสผ่านให้กับกรุ๊ป
200	เป็นหมายเลขประจำกลุ่ม

ผู้ใช้ชื่อ sample มีหมายเลขประจำตัวผู้ใช้งานเป็น 100 และชื่อกลุ่มเป็น staff ซึ่งมีหมายเลขประจำตัวกลุ่ม เป็น 200 ซึ่ง หมายเลขประจำตัวผู้ใช้ และ หมายเลขประจำกลุ่ม เป็นเลขจำนวนจริงขนาด 16 บิตมีค่าตั้งแต่ 0 จนถึง 65535 บางระบบอาจมีหมายเลขประจำตัวผู้ใช้ หรือหมายเลขประจำกลุ่ม เป็นเลข 32 บิตได้ มนุษย์จะใช้ตัวอักษรในการอ้างถึง ส่วนลีนุกซ์จะอ้างถึงเป็นตัวเลขเสมอเป็นแนวคิดที่ไม่ซับซ้อนใช้ได้มีประสิทธิภาพมาก โดยหมายเลขประจำตัวผู้ใช้ และหมายเลขประจำกลุ่ม ของผู้ใช้จะกำหนดสิทธิ์ในการเข้าถึงแฟ้มในระบบ โดยจับคู่กันระหว่างชื่อผู้ใช้กับ หมายเลขประจำตัวผู้ใช้ และกรุ๊ป กับ หมายเลขประจำกลุ่ม ถ้าผู้ใช้สองคนมี หมายเลขประจำตัวผู้ใช้ เหมือนกัน ลีนุกซ์จะถือว่าผู้ใช้สองคนนี้เป็นคนเดียวกันแม้ว่าชื่อบัญชีผู้ใช้ และรหัสผ่านจะต่างกัน เนื่องจากเป็นคนเดียวกันในระบบทำให้มีสิทธิ์เหมือนกันทุกอย่าง เช่น สามารถลบแฟ้มของอีกคนหนึ่งก็ได้ สามารถหยุดโพรเซสของอีกคนหนึ่งได้ (ความหมายของคำว่าอีกคน

หนึ่งในที่นี้คือใช้ชื่อผู้ใช้งานคนละชื่อ) ความไม่รอบคอบในลักษณะนี้อาจจะทำให้ระบบทำงานได้ไม่ถูกต้องและมีผลในแง่ความปลอดภัยของระบบ

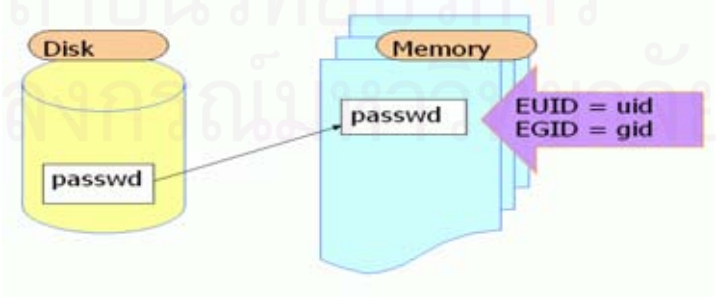
ผู้ใช้ในระบบที่ หมายเลขประจำตัวผู้ใช้ เป็น 0 คือผู้ดูแลระบบ (Superuser) มีสิทธิ์ทุกอย่างในระบบ ในการเปลี่ยนผู้ใช้ในระบบไปเป็นผู้ใช้คนอื่น ลีนุกซ์มีคำสั่งที่เกี่ยวข้องสองคำสั่ง คำสั่งแรกคือ su ซึ่งย่อมาจากคำว่า Substitute User ใช้ในการเปลี่ยนหมายเลขประจำตัวผู้ใช้

2.3.7 ความหมายของสติ้กกี้บิต (Sticky bit)

มีบิตพิเศษอีกหนึ่งบิตเรียกว่า สติ้กกี้บิต ซึ่งในปัจจุบันจะใช้กับไดเรกทอรีเท่านั้นมีความหมายคือ ไดเรกทอรีใดก็ตามที่มีบิตที่เรียกว่าสติ้กกี้บิตอยู่ หมายถึงแฟ้มในไดเรกทอรีจะมีการเปลี่ยนแปลง ลบ เปลี่ยนชื่อ เพิ่มเนื้อหาได้เฉพาะบุคคลสามคนนี้เท่านั้นคือ เจ้าของแฟ้ม (แฟ้มภายใต้ไดเรกทอรีที่มี สติ้กกี้บิต) เจ้าของไดเรกทอรีที่มีบิตพิเศษ และผู้ดูแลระบบหรือผู้ใช้งานที่ชื่อว่ารูท มักใช้ในไดเรกทอรี /tmp ของลีนุกซ์

2.3.8 สิทธิ์ที่ได้ของหมายเลขประจำตัวผู้ใช้ของโพรเซสในการเข้าถึงแฟ้ม

โพรเซสบนยูนิกซ์ในเวลาใดๆ จะมีหมายเลขประจำที่บอกให้ทราบถึงสิทธิ์ในการเข้าถึงแฟ้ม และหรือ ไดเรกทอรีใดๆ ในระบบของโพรเซส เรียกว่า Effective UID หรือ EUID โดยปกติจะมีค่าเท่ากับ หมายเลขประจำตัวผู้ใช้ของผู้ที่สั่งให้โพรเซสในระบบยูนิกซ์ทำงาน ตัวอย่างเช่น สั่งให้โปรแกรม passwd ทำงาน โดยปกติเมื่อผู้ใช้สั่งให้โปรแกรม passwd ทำงาน โพรเซสจะมี EUID เท่ากับ หมายเลขประจำตัวผู้ใช้ ของคนที่สั่งให้โปรแกรมนี้ทำงานขึ้นมา และ EUID นี้เองจะเป็นตัวกำหนดและบอกว่าโพรเซสใดจะมีสิทธิ์ในระบบมากน้อยแค่ไหน ดังรูปที่ 2.1

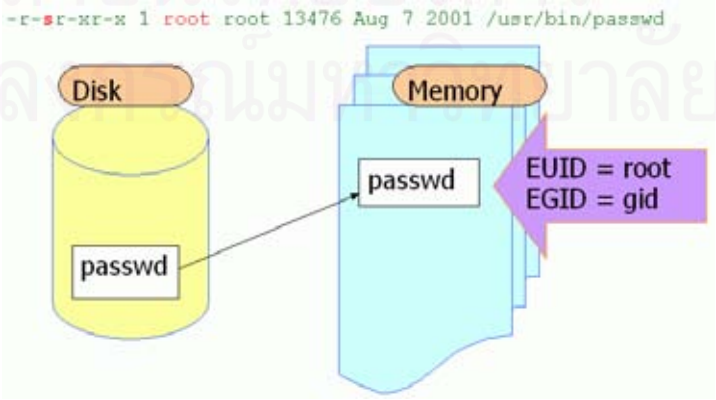


รูปที่ 2.1 หมายเลขประจำตัวผู้ใช้ เมื่อสั่งให้โปรแกรมทำงาน

แต่บางครั้งจำเป็นต้องใช้สิทธิ์มากกว่าที่มีอยู่ ในการเข้าถึงแฟ้มสำคัญ ไม่ว่าจะเป็นการอ่านการเขียนแฟ้มสำคัญของระบบ ซึ่งโดยทั่วไปผู้ใช้ทั่วไปไม่มี ยกตัวอย่างเช่นโปรแกรม passwd ซึ่งเป็นโปรแกรมที่ใช้ในการเปลี่ยนรหัสผ่านของผู้ใช้ในระบบ ซึ่งทราบกันดีว่าข้อมูลของผู้ใช้ในระบบทุกคนจะเก็บไว้ในแฟ้ม /etc/passwd และรหัสผ่านเก็บในแฟ้ม /etc/shadow ซึ่งอนุญาตให้เฉพาะผู้ดูแลระบบหรือ รูท มีสิทธิ์ในการเขียนแฟ้มสองแฟ้มนี้ ดังนั้นตามกฎเกณฑ์ของระบบเมื่อจะทำการเปลี่ยนรหัสผ่านของตนเอง เมื่อสั่งให้โปรแกรม passwd ทำงาน จะได้โพรเซสของ passwd ขึ้นมาซึ่งจะมี EUID เท่ากับ หมายเลขประจำตัวผู้ใช้ ของผู้ใช้ในระบบไม่ใช่ผู้ดูแลระบบที่มีสิทธิ์อ่านเขียนแฟ้ม ดังนั้นจะไม่สามารถเปลี่ยนรหัสผ่านได้

ดังนั้นจึงมีวิธีการที่เรียกว่าบิตกำหนดผู้ใช้ (SUID) และ บิตกำหนดกลุ่มผู้ใช้ (SGID) ในการเพิ่มความสามารถในการทำงานภายใต้กฎเกณฑ์ดังกล่าวนี้ เพื่อแก้ปัญหาที่เกิดขึ้นนี้ ยูนิกซ์รวมไปถึงลีนุกซ์ได้ออกแบบให้โพรเซสที่ทำงานนั้นมีกฎยกเว้นในการเปลี่ยนให้ EUID เป็นผู้ใช้คนอื่นได้ หรือใช้สิทธิ์จาก EUID ของคนอื่นได้เช่นจากตัวอย่างของโปรแกรม passwd นั้นเมื่อผู้ใช้ในระบบสั่งให้โปรแกรมทำงานจะมี EUID เท่ากับ หมายเลขประจำตัวผู้ใช้ ของผู้ใช้ที่สั่งให้โปรแกรม passwd ทำงานแต่ในกรณีนี้ต้องการให้ โพรเซส passwd ใช้สิทธิ์ของผู้ดูแลระบบในที่นี้คือผู้ใช้งานชื่อรูท ในการเข้าถึงแฟ้มสำคัญสองแฟ้ม ดังนั้นจึงจำเป็นต้องเปลี่ยนให้ EUID ของโพรเซสจากหมายเลขประจำตัวผู้ใช้ ของผู้ใช้ที่สั่งให้โปรแกรมทำงานไปเป็นหมายเลขประจำตัวผู้ใช้ของผู้ดูแลระบบ

โปรแกรมที่มีการเปลี่ยน หมายเลขประจำตัวผู้ใช้ ขณะที่มีการทำงานหรือเมื่อเป็นโพรเซสเรียกว่าโปรแกรม SUID (Set User ID) โดยจะใช้วิธีการเปลี่ยนหมายเลขประจำตัวผู้ใช้ ให้ตรงกับเจ้าของโปรแกรมนั้นๆ โดยในตัวอย่างนี้คือ passwd ซึ่งเจ้าของแฟ้มนี้คือผู้ใช้งานชื่อว่ารูท ดังนั้นเมื่อโปรแกรมทำงานจะได้โพรเซสที่มี EUID เป็นรูท หรือเจ้าของโปรแกรม passwd ซึ่งจะไม่ใช่ผู้ใช้ที่สั่งให้โปรแกรม passwd ทำงานอย่างในกรณีทั่วไป ดังรูปที่ 2.2



รูปที่ 2.2 การทำงานของโปรแกรม SUID

สำหรับบิตกำหนดกลุ่มผู้ใช้ ใช้แนวความคิดเดียวกันแต่เปลี่ยนจากเจ้าของแฟ้มเป็น กลุ่มเดียวกับแฟ้มนั้นๆ สำหรับการระบุบิตกำหนดผู้ใช้ และบิตกำหนดกลุ่มผู้ใช้ จะระบุในตำแหน่งเดียวกับ excute ในบิตอนุญาตของแฟ้มหรือโปรแกรมในระบบตัวอย่างเช่น โปรแกรม passwd

```
-r-sr-xr-x 1 root root 13476 Aug 7 2001 /usr/bin/passwd
```

จะเห็นว่ามีเครื่องหมายว่า 's' ใช้ในการระบุ SUID ของโปรแกรม passwd และตัวอย่างโปรแกรมที่มี SGID เช่น

```
-r-xr-sr-x 1 root root 4562 Aug 8 2001 /tmp/samples
```

ผู้ดูแลระบบมักจะเข้าใจผิดว่าแนวความคิดดังกล่าวนี้ ทำให้เกิดจุดอ่อนขึ้นในระบบ แต่ในความเป็นจริงแล้ว จุดอ่อนจะเกิดได้จากสองจุดเท่านั้นคือ

1. การเขียนโปรแกรมที่ต้องใช้บิตกำหนดผู้ใช้หรือบิตกำหนดกลุ่มผู้ใช้ไม่ถูกต้องมีฟังก์ชันในการทำงานเกินขอบเขตของระบบ และเป็นอันตรายต่อระบบ
2. ผู้ดูแลระบบอนุญาตให้ผู้ใช้ทั่วไปในระบบสามารถสร้างแฟ้มหรือโปรแกรมที่มีสิทธิ์ของบิตกำหนดผู้ใช้หรือบิตกำหนดกลุ่มผู้ใช้ ได้ตามอำเภอใจ

ซึ่งจะเห็นได้ว่าไม่ใช่ปัญหาของแนวความคิดในการมีทั้ง SUID และ SGID เพราะว่าความผิดพลาดเกิดจากผู้ดูแลระบบที่ไม่รอบคอบและไม่ตรวจสอบว่าแฟ้มในระบบมีแฟ้มที่มี SUID และ SGID มากน้อยแค่ไหนในระบบ และไม่ทราบถึงขอบเขตการทำงานของโปรแกรมที่มีรูปแบบที่ต้องใช้แนวความคิดนี้ในการทำงาน

2.3.9 โครงสร้างระบบไดเรกทอรีบนลีนุกซ์

ระบบลีนุกซ์ใช้การเก็บข้อมูลโดยใช้แฟ้มและไดเรกทอรีเข้ามาช่วย เช่นเดียวกับในระบบยูนิกซ์ทั่วไป โดยจะมีลักษณะเป็นแบบโครงสร้างแบบต้นไม้ (Hierarchy) ไดเรกทอรีบนสุดเรียกว่าไดเรกทอรีราก (root directory) ซึ่งจะประกอบไปด้วยแฟ้มและไดเรกทอรีต่างๆ ย่อยลงไปเรื่อยๆ ซึ่งราย ละเอียดไดเรกทอรีมาตรฐานในระบบลีนุกซ์แสดงได้ดังนี้

ไดเรกทอรีรูท (/) เป็นไดเรกทอรีแรกสุดในระบบเก็บไดเรกทอรีย่อยต่างๆในระบบ
ไดเรกทอรียูเอสอาร์ (/usr) ใช้เก็บคำสั่งและไลบรารี (library) และคู่มือช่วยเหลือ (man page)

ไดเรกทอรีลิบ (/lib) ใช้เก็บไลบรารีต่างๆ ในระบบแฟ้มรูท
ไดเรกทอรีดีไวซ์ (/dev) ใช้เก็บแฟ้มต่างๆ ที่เกี่ยวข้องกับอุปกรณ์
ไดเรกทอรีบูท (/boot) ใช้เก็บแฟ้มที่จำเป็นในการบู๊ตโหลดเดอร์ (boot loader) ต่างๆ
ไดเรกทอรี (/var) ใช้เก็บแฟ้มที่มีการเปลี่ยนแปลงขนาดไปตามเวลา เช่น ล็อกระบบ
ไดเรกทอรีโฮม (/home) ใช้เก็บไดเรกทอรีบ้านของผู้ใช้แต่ละคน
ไดเรกทอรีพร็อก (/proc) เป็นระบบแฟ้มที่เป็นตัวตนอยู่ในหน่วยความจำ เช่นโพรเซส (process)
ไดเรกทอรีเท็ม (tmp) เป็นไดเรกทอรีที่โปรแกรมต่างๆมักใช้เก็บแฟ้มชั่วคราวในระบบ
ไดเรกทอรีบิน (/bin) เป็นไดเรกทอรีที่เก็บคำสั่งต่างๆของผู้ใช้งานทั่วไป
ไดเรกทอรีเอสบิน (/sbin) เป็นไดเรกทอรีที่เก็บคำสั่งต่างๆของผู้ดูแลระบบ
ไดเรกทอรี (/etc) เป็นไดเรกทอรีใช้เก็บแฟ้มที่เป็นองค์ประกอบต่างๆ ของระบบไว้

2.4 การตรวจสอบความปลอดภัยในระบบลีนุกซ์จากแฟ้มคอนฟิก (configuration file) และแฟ้มล็อก (log file)

2.4.1 การตรวจดูการเปลี่ยนแปลงของระบบจากแฟ้มคอนฟิก

แฟ้มคอนฟิกในระบบลีนุกซ์ มีแฟ้มสำคัญที่ต้องตรวจสอบดังนี้

2.4.1.1 แฟ้ม /etc/passwd ตรวจสอบว่ามีการลบ หรือเพิ่มผู้ใช้บ้างหรือไม่ หรือมีการแก้ไขข้อมูลของผู้ใช้บ้างหรือไม่

2.4.1.2 แฟ้ม /etc/inetd.conf มีการเปิด การให้บริการที่ผิดปรกติ จากเดิมหรือไม่ และ มีการปิดการให้บริการใดบ้าง

2.4.1.3 ระบบที่มีการใช้คำสั่ง "r-commands" (rlogin, rsh, rexec) ต้องตรวจสอบแฟ้ม /etc/hosts.equiv หรือแฟ้มที่เกี่ยวข้องกับ .rhosts ว่ามีการแก้ไขใดๆ บ้างหรือไม่

2.4.1.4 ตรวจสอบบิตกำหนดผู้ใช้ และบิตกำหนดกลุ่มผู้ใช้ของแฟ้มที่ถูกสร้างขึ้นมาใหม่ ซึ่งเป็นไปได้ที่แฟ้มเหล่านั้นอาจจะเป็นโปรแกรมประเภทประตูลับ (Backdoor Program)

2.4.1.5 ตรวจสอบบิตอนุญาตของไดเรกทอรี ที่สำคัญของระบบ เช่น ไดเรกทอรี /bin ไดเรกทอรี /etc และไดเรกทอรี /sbin ฯลฯ

2.4.2 การตรวจสอบแฟ้มล็อกในระบบ

การตรวจสอบการบุกรุกจากแฟ้มล็อกจะช่วยในการหาข้อมูลได้ว่าระบบถูกบุกรุกและแก้ไขอย่างไร สามารถตรวจว่าการบุกรุกเกิดขึ้นเมื่อไร เกิดอะไรบ้างในขณะบุกรุก และตรวจสอบการเข้าใช้ระบบ ในระบบ ลีนุกซ์ หาเส้นทางที่ระบุแฟ้มล็อก ได้จากแฟ้ม /etc/syslog.conf

แฟ้มต่อไปนี้เป็น แฟ้มล็อกในระบบลีนุกซ์ โดยแต่ละแฟ้มมีความแตกต่างตามหน้าที่และโปรแกรมที่ใช้สร้างแฟ้ม

2.4.2.1 แฟ้ม /var/log/message ล็อกประเภทข้อความ (messages log) จะเก็บข้อมูลหลากหลายแบบ ซึ่งแฟ้มนี้สามารถใช้ในการหาการแก้ไขแฟ้ม โดยมีข้อมูลของเวลาและเหตุการณ์ เพื่อช่วยตรวจสอบการบุกรุกได้

2.4.2.2 แฟ้ม /etc/log.d/scripts/logfiles/xferlog การบุกรุกผ่านทาง เอฟทีพี เซิร์ฟเวอร์ (ftp server) แฟ้ม xferlog จะทำหน้าที่เก็บข้อมูลการทำงานของบริการนี้ แฟ้มนี้สามารถตรวจสอบการถ่ายโอนข้อมูลที่แปลกปลอมระหว่างระบบกับผู้บุกรุก

2.4.2.3 แฟ้ม /var/run/utmp เป็นแฟ้มที่ให้ข้อมูลว่า มีใครเข้าสู่ระบบอยู่บ้างในขณะนั้น

2.4.2.4 แฟ้ม /var/log/wtmp เป็นแฟ้มที่ให้ข้อมูลการล็อกอินเข้าสู่ระบบสามารถตรวจสอบว่ามีผู้ใช้คนใดเข้าใช้ และออกจากระบบเมื่อใด

2.4.2.5 แฟ้ม /etc/log.d/script/services/secure เป็นแฟ้มที่เก็บล็อกของโปรแกรม ทีซีพีแรพเปอร์ (tcp wrapper) ในแฟ้มซีเคียวทุกครั้งที่มีการเชื่อมต่อกับระบบผ่านโปรแกรม ไอเน็ตดี (inetd) ที่ใช้โปรแกรมทีซีพีแรพเปอร์ ล็อกแมทเสดจะเกิดขึ้น โดยสามารถตรวจสอบได้ว่ามีการสร้างการเชื่อมต่อจากบริการที่ผิดปรกติหรือมีการเชื่อมต่อบริการกับเซิร์ฟเวอร์ที่ไม่ให้สิทธิหรือไม่

2.4.3 บริการบันทึกล็อกในระบบ

2.4.3.1 โปรแกรมซิสล็อกดีมอน (syslogd) [24] เป็นกลไกที่ใช้ในการเก็บข้อมูลล็อกของเคอร์เนลและแอพพลิเคชันบนระบบลินุกซ์ เป็นดีมอนที่ติดตั้งมาให้พร้อมกับระบบปฏิบัติการลีนุกซ์ในเกือบทุกตระกูล โดยผู้ดูแลระบบสามารถปรับแต่งแฟ้มคอนฟิก เพื่อควบคุมการทำงานของ ซิสล็อกดีมอน ได้ เช่น ให้ ซิสล็อกดีมอนเก็บข้อมูลไปไว้ที่แฟ้มใด หรือให้ส่งข้อมูลล็อกนี้ไปเก็บไว้ยังเครื่องอื่นในเครือข่าย ข้อมูลล็อกที่ควบคุมโดยโปรแกรมซิสล็อกดีนั้น จะถูกกำหนดให้มีค่า facility และค่า priority โดยส่วนของค่า facility นั้น เป็นข้อมูลที่อธิบายถึงแหล่งกำเนิดของข้อมูลล็อกนั้นๆ เช่น ข้อมูลล็อกที่ส่งมาจากระบบเมล์ก็จะมีค่า facility มีค่าเป็น mail ส่วนค่า priority นั้น จะแสดงถึงระดับความสำคัญของเหตุการณ์ที่เกิดขึ้นในแต่ละค่าของ facility สามารถแสดงรายละเอียด ในตารางที่ 2.8 และตารางที่ 2.9

ตารางที่ 2.8 ตารางค่า Priority

Priority	**คำอธิบาย**
emerg	ภาวะฉุกเฉิน
alert	แจ้งเตือนเร่งด่วน
crit	ล่อแหลม
err	มีข้อผิดพลาด
warning	คำเตือน
notice	ข้อสังเกต
info	ข้อมูลทั่วไป
debug	สำหรับใช้ดีบักเท่านั้น

ตารางที่ 2.9 ตารางค่า Facility

Facility	**คำอธิบาย**
auth	เกี่ยวข้องกับการทำ authentication
authpriv	การทำ private authentication เท่านั้น
cron	cron daemon
daemon	system daemons
kern	ส่วนของ kernel
lpr	line printer spooling system
mail	sendmail และซอฟแวร์อื่นที่เกี่ยวกับเมล์
mark	ให้บันทึกเวลาขณะเกิดเหตุการณ์ด้วย
news	usenet news system
security	เหมือนกับ auth
syslog	ข้อมูลล็อกภายในของ syslogd
user	ส่วนของโปรเซสของ user
uucp	สำรองไว้สำหรับ UUCP
local0 - local7	local messages

2.4.3.2 แฟ้ม /etc/syslog.conf การทำงานของซิสล็อกดีมอนนั้น จะขึ้นอยู่กับแฟ้ม /etc/syslog.conf เป็นหลักการแก้ไขใดๆ ที่เกิดขึ้นกับแฟ้มนี้นั้นจะยังไม่มีผลต่อการทำงานของ syslogd ในทันที จะต้องทำการเปิดปิดบริการซิสล็อกดีมอนใหม่เสียก่อน รูปแบบคำสั่งในแฟ้ม /etc/syslog.conf นั้นมีรูปแบบดังนี้

facility.level action

facility1, facility2.level	action
facility1.level1; facility2.level2	action
*.level	action
*.level;badfacility.none	action

หมายความว่า เมื่อมีข้อมูลล็อกที่มีค่า facility และค่า level ที่ตรงหรือมากกว่ากับที่ตั้งไว้ก็จะกระทำตามค่า action ที่กำหนดไว้ทั้งนี้เพราะค่า level ที่ตั้งไว้นั้นเป็นค่าต่ำสุด (minimum) ซึ่งหมายความว่าถ้าตั้งค่า level เท่ากับ debug ก็จะคลอบคลุมทุกระดับของค่า facility นั้นๆ เลย ทั้งนี้สามารถใช้เครื่องหมาย “ * ” แทนทุกๆ ค่าใน facility หรือค่า priority level นั้นๆ ได้ เช่น mail.* /var/log/mail หมายความว่าให้ซิสล็อกดีมอนเก็บข้อมูลล็อกของเมล์ ทุก level ไปไว้ยังแฟ้ม /var/log/mail ในขณะที่ level ที่มีค่าเท่ากับ “none” นั้น หมายความว่าไม่ให้สนใจ facility ที่ประกาศค่า level เป็น none เช่น

*.emerg;mail.none /var/log/emer.log

คือให้เก็บข้อมูลล็อกที่มี level เป็นค่า “emerg” สำหรับทุก facility ยกเว้น mail facility
สำหรับค่า Action นั้นสามารถเลือกได้ดังนี้คือ

filename : เก็บข้อมูลล็อกนั้นลงในแฟ้มที่กำหนด

@hostname : ส่งต่อข้อมูลล็อกไปยังซิสล็อกดีมอนบนเครื่องที่กำหนด

@ipaddress : ส่งต่อข้อมูลล็อกไปยังเซิร์ฟเวอร์ที่มีเลขที่อยู่ไอพีตามที่กำหนด

user1, user2 : ส่งข้อมูลล็อกไปยังหน้าจอของผู้ใช้งานที่กำหนด ถ้าผู้ใช้งานเหล่านั้นยังล็อกอินอยู่ในระบบ

ส่งข้อมูลล็อกไปยังทุกๆ ผู้ใช้งานที่ยังล็อกอินอยู่ในระบบ

/dev/console เพื่อส่งข้อมูลล็อกไปยังคอนโซลดีไวซ์ หรือดีไวซ์อื่นๆ ตามที่ต้องการ

สำหรับลีนุกซ์เรดแฮต ได้ขยายความสามารถของซิสล็อกดีมอนเพิ่มเติมโดยอนุญาตให้ข้อมูลล็อกสามารถถูกส่งแบบไพป์ (pipe) ไปยังแฟ้มได้ โดยแก้ไขในแฟ้ม syslog.conf และยังสามารถใช้เครื่องหมาย = และ ! ในแฟ้ม syslog.conf ได้

เครื่องหมาย = หมายถึง priority ที่กำหนดเท่านั้น

เครื่องหมาย ! หมายถึง priority อื่นที่ไม่ใช่ priority นี้และสูงกว่า

ตัวอย่าง เช่น

mail.info ความหมายคือ ข้อมูลล็อกที่เกี่ยวข้องกับเมล์และมีค่า priority เป็น “info” และสูงกว่า

mail.=info ความหมายคือ ข้อมูลล็อกที่เกี่ยวข้องกับเมล์และมีค่า priority เป็น“info”เท่านั้น

mail.info;mail.!err ความหมายคือ ข้อมูลล็อกที่เกี่ยวข้องกับเมล์และมี priority เป็น “info” “notice”และ“warning”

mail.debug;mail.!=warning ความหมายคือ ข้อมูลล็อกที่เกี่ยวข้องกับเมล์และมี priority ทุกระดับที่ไม่ใช่ warning และลีนุกซ์เรดแฮตนั้น โดยปกติจะเก็บข้อมูลล็อกไว้ในแฟ้มซึ่ง อยู่ภายใต้โฟลเดอร์ /var/log และถูกติดตั้งมาพร้อมกับล็อกโรเทน (logrotate) ซึ่งเป็นเครื่องมือที่ช่วยจัดการล็อกแฟ้มได้อย่างมีประสิทธิภาพ ปกติแล้วจะโรเทนล็อกแฟ้มอาทิตย์ละครั้ง และจะเก็บล็อกไว้ 4 รอบ หรือ 1 เดือน ผู้ดูแลระบบสามารถปรับเปลี่ยนค่าเหล่านี้ได้ที่แฟ้ม /etc/logrotate.conf

2.5 การทำเมล์รีเลย์ (mail relay) ในระบบเมล์เซิร์ฟเวอร์ [20]

ความหมายของคำว่า รีเลย์ คือการส่งอี-เมล์จากภายนอกโดเมนผ่านเมล์เซิร์ฟเวอร์ไปที่โดเมนอื่น โดยที่ทั้งผู้ส่งและผู้รับไม่ใช่ผู้ที่อยู่ภายใต้โดเมนที่เมล์เซิร์ฟเวอร์ให้บริการนี้ จะถือว่าเป็นการรีเลย์ในอดีตการรีเลย์ช่วยให้การส่งอี-เมล์ทำได้สะดวกเป็นการส่งกันต่อไปเป็นทอดๆ เหมือนกับที่แพ็กเกจของโพรโตคอลทีซีพีไอพี ที่วิ่งผ่านเส้นทางอื่นได้ แต่ปัจจุบันการเปิดให้เมล์เซิร์ฟเวอร์สามารถรีเลย์ได้ถือเป็นอันตรายอย่างยิ่งเพราะสามารถนำไปใช้ในทำเมล์สแปม (mail spamming) ทำการส่งอี-เมล์จำนวนมหาศาลไปยังผู้รับจำนวนมาก ซึ่งเป็นการเปลืองทรัพยากรระบบโดยใช่เหตุ ดังนั้นการกำหนด รีเลย์ จึงควรคำนึงถึงความจำเป็นและความเหมาะสม

2.6 ระบบการขนส่งแฟ้มผ่านเน็ตเวิร์ค (FTP)

ระบบการขนส่งแฟ้มผ่านเน็ตเวิร์ค (File Transfer Protocol) เป็นโพรโตคอลที่ใช้ในการขนส่งแฟ้มระหว่างเครื่อง 2 เครื่องซึ่งโพรโตคอลเอฟทีพีนั้นทำงานอยู่บน โพรโตคอลทีซีพีไอพีซึ่งจะแบ่งออกเป็นในส่วนการให้บริการ กับส่วนที่ใช้บริการโดย เอฟทีพีเซิร์ฟเวอร์ นั้นจะกำหนด

ทรัพยากรที่ต้องการแชร์ให้กับผู้ใช้บริการเข้ามาดาวน์โหลด หรือผู้ใช้บริการสามารถอัพโหลดข้อมูลมาเก็บไว้ก็ได้ โดยสามารถกำหนดสิทธิ์ตามผู้ใช้ได้

2.6.1 การล็อกอินด้วยผู้ใช้นิรนาม (Anonymous FTP)

เป็นการให้บริการเอฟทีพีเซิร์ฟเวอร์ กับผู้ใช้บริการทั่วไป ซึ่งอาจมีเป็นจำนวนมากซึ่งไม่สามารถที่จะมาสร้างบัญชีผู้ใช้ให้รองรับกับทุกๆคนได้ การล็อกอินในแบบนี้จึงทำขึ้นมาเพื่อให้ผู้ใช้บริการที่ไม่มีบัญชีผู้ใช้อยู่บนเซิร์ฟเวอร์สามารถเข้ามาใช้บริการได้ โดยล็อกอินเข้ามาด้วยผู้ใช้งานนิรนาม (anonymous user) และระบุรหัสผ่านเป็น อีเมล์แอดเดรส ของผู้เข้ามาใช้บริการก็สามารถใช้บริการได้แล้ว โดยปกติในระบบปฏิบัติการลีนุกซ์ถ้ามีการติดตั้ง เอฟทีพีเซิร์ฟเวอร์ ก็จะทำการติดตั้งระบบการล็อกอินด้วยผู้ใช้ที่ไม่มีอยู่ในระบบมาโดยอัตโนมัติ โดยจะติดตั้งแพ็จเกจต่างๆ ที่ไดเรกทอรี /var/ftp/ ซึ่งถือว่าเป็นไดเรกทอรีรูทจำลองของ ผู้ใช้งานนิรนาม โดยจำกัดขอบเขตการเข้าถึงข้อมูลภายในเครื่องให้อยู่ภายใต้ไดเรกทอรีที่กำหนด ซึ่งภายใต้ไดเรกทอรี /var/ftp/ ก็จะประกอบไปด้วย ไดเรกทอรี /etc/ ซึ่งจะเก็บแฟ้ม passwd และ แฟ้ม group เพื่อกำหนดค่าผู้ใช้งานเอฟทีพีสามารถล็อกอินเข้ามาที่ไดเรกทอรีหรือห้ามใช้คำสั่งในเส้นทาง(path) ใด และสร้างกลุ่มของเอฟทีพี นอกจากนั้นยังมีในส่วนของไดเรกทอรี /bin และ /lib ซึ่งเก็บคำสั่งที่ใช้ในการขนส่งแฟ้ม และคำสั่งอำนวยความสะดวกต่างๆ ในการใช้งานเอฟทีพี ซึ่งจะสังเกตุได้ว่า ไดเรกทอรีและแฟ้มต่าง ๆ ที่เกี่ยวข้องกับการทำการล็อกอินด้วยผู้ใช้ที่ไม่มีอยู่ในระบบ ก็เพื่อจำกัดขอบเขตในการเข้าถึงข้อมูลภายในเครื่อง ซึ่งหากมีการกำหนด บิตอนุญาต ที่ไม่ถูกต้องย่อมเป็นอันตรายต่อความปลอดภัยของระบบ จึงควรประเมินความจำเป็นในการทำงานเอฟทีพีเซิร์ฟเวอร์ และคำนึงถึงการเปิดการล็อกอินด้วยผู้ใช้นิรนามถือเป็นเรื่องจำเป็นด้วยหรือไม่

2.7 การดักจับแพ็คเกตในเครือข่าย (Packet Sniffing)

การดักจับข้อมูลที่ผ่านไปมาระหว่าง เน็ตเวิร์คเรียกว่าสนิพฟิง (Sniffing) เนื่องจากระบบอีเทอร์เน็ต ถูกสร้างขึ้นมาบนกฎของการใช้ร่วมกันนั่นคือคอมพิวเตอร์ทุกเครื่องบนเครือข่ายท้องถิ่น (Local Area Network) จะใช้สายเน็ตเวิร์ค เดียวกัน ฮาร์ดแวร์ของอีเทอร์เน็ต ถูกสร้างมาด้วย “ตัวกรอง” (filter) ที่จะไม่สนใจข้อมูลที่วิ่งในเครือข่าย ที่เป็นของคนอื่น ซึ่งทำได้โดยจะไม่สนใจเฟรมที่มีหมายเลขประจำเครื่อง (MAC address) ไม่ตรงกับของตนเอง แต่โปรแกรม สนิพฟิ่ง หรือวายแท็บ (wiretap) จะตัดตัวกรองนี้ออกไปทำให้ฮาร์ดแวร์อินเตอร์เฟสอยู่ในภาวะ“การทำงานแบบไม่เลือก” (promiscuous mode) หมายถึงจะรับทุก ๆ แพ็กเก็ตโดยไม่สนใจว่าแพ็กเก็ตเฮดเดอร์จะสื่อสารว่า

อย่างไร ซึ่งผู้ไม่ประสงค์ดีมักใช้คุณสมบัตินี้ในการดักจับและวิเคราะห์แพ็คเก็ตทุกๆ แพ็คเก็ตที่วิ่งในเครือข่ายนั้น

2.8 **ม้าโทรจัน** (Trojan house) [3]

ม้าโทรจันเป็นโปรแกรมที่ทำหน้าที่ในการสร้างประตูลับไว้ในระบบ เพื่อวัตถุประสงค์บางอย่างของผู้สร้างโปรแกรมโทรจัน ตัวอย่างเช่น การปล่อยโปรเซสซึ่งจะคอยฟังอยู่บนพอร์ต ทีซีพี (TCP listener) และขโมยเชลล์ (shell) ผ่านทางช่องทางสื่อสารกับเซอร์วิสที่มาติดต่อพอร์ต ทีซีพี นั้นๆ กลับไปยังเครื่องแฮกเกอร์ เพื่อให้แฮกเกอร์ติดต่อไปยังพอร์ตที่เปิดไว้เพื่อเข้าไปในระบบ ปริมาณของเทคนิคการใช้โทรจันที่มีแนวโน้มเป็นไปได้นั้นมีมากมาย ซึ่งขึ้นอยู่กับจินตนาการและความคิดของแฮกเกอร์การเฝ้าตรวจตรา และคอยมอนิเตอร์ดูการใช้งานรวมทั้งจัดทำฐานข้อมูลที่เก็บหมายเลขพอร์ตทั้งหมดที่เปิดไว้ในระบบนั้น มีส่วนช่วยป้องกันการโจมตีของม้าโทรจันได้

ต่อไปนี้เป็นฐานข้อมูลตัวอย่างของ ทีซีพี พอร์ตซึ่งม้าโทรจันนิยมใช้ในการติดต่อเข้ามาในระบบ จากเว็บไซต์ http://www.anti-trojan.com ในตารางที่ 2.10

ตารางที่ 2.10 ตัวอย่างของ ทีซีพี พอร์ต ซึ่งม้าโทรจันนิยมใช้ในการติดต่อ

ม้าโทรจัน	พอร์ตที่ใช้	ม้าโทรจัน	พอร์ตที่ใช้	ม้าโทรจัน	พอร์ตที่ใช้
BackOrifice 1.x	31337	GateCrasher	6969	GirlFriend	21554
NetBus 1.x	12346	NetBus 2.x	20034	NetSphere	30100
Portal of Doom	10167	The tHing	6400	SubSeven	1243
Deep Throath 1,2,3.x	6670	Master Paradise	31	Silencer	1001
Millenium	20000	Devil 1.03	65000	NetMonitor	7306
Streaming Audio Trojan	1170	Socket23	30303	Telecommando	31466
Gjamer	12076	IcqTrojen	4950	Priotrity	16969

2.9 **หมายเลขเซอร์วิสหรือแอพพลิเคชันพอร์ต** [3]

เนื่องจากอุปสรรคที่ใหญ่ที่สุดในการประเมินด้านความปลอดภัย ก็คือการทำความเข้าใจว่ามีเซอร์วิสอะไรกำลังเปิดให้บริการหรือกำลังทำงานอยู่ในระบบ หมายเลขพอร์ต (Port Listing) พร้อมด้วยรายชื่อเซอร์วิสที่กำลังทำงานอยู่บนพอร์ตนั้น เป็นปัจจัยในการแยกแยะหาช่องโหว่ต่างๆ ที่มีอยู่ในระบบ การสแกนหมายเลขพอร์ต ทีซีพี ทั้งหมด 65535 พอร์ต ปกติเป็นสิ่งที่ต้องใช้เวลานาน หากมีรายชื่อพอร์ตของเซอร์วิสที่ได้ตรวจสอบแล้วว่า มักเป็นที่นิยมใช้บ่อยซึ่งรวบรวมโดย สถาบันวิทยาการสารสนเทศแห่ง University of Southern California จัดทำไว้ที่ไซต์

ftp://ftp.isi.edu/in-notes/iana/assignments/port-numbers ซึ่งจะช่วยให้สามารถพุ่งเป้าไปที่เซอร์วิสที่มีแนวโน้มว่าจะเกิดช่องโหว่ขึ้นได้

2.10 ภาษา เพิร์ล (Perl)

เพิร์ล ย่อมาจาก Practical Extraction and Report Language เป็นภาษาที่ใช้ใน การเขียนโปรแกรม สามารถใช้กับงานต่างๆ ได้หลายรูปแบบ สามารถทำงานได้ดีกับแฟ้มข้อมูลประเภทข้อความ (text file) ประมวลผลข้อมูลจากแฟ้มข้อมูลประเภทข้อความ และพิมพ์ผลลัพธ์การประมวลผลนั้นออกมา เพิร์ล เป็นภาษาที่ง่ายกับการเรียนรู้และใช้งาน เพิร์ล มีข้อดีหลายๆ อย่างของภาษาซี (C) ภาษาเอสอีดี (sed) ภาษาเอดับบลิวเค (awk) และเชลล์สคริปต์ เพิร์ลยังมีส่วนเพิ่มเติมในการพัฒนาส่วนติดต่อกับผู้ใช้งานแบบกราฟิก เรียกว่า ทีเค ซึ่งเป็นทูลคิต (Toolkit) ที่จะช่วยให้การพัฒนากราฟิกสำหรับ เพิร์ล ทำได้ง่ายขึ้น

2.11 เพิร์ล ทีเค (Perl Tk)

ทีเค เป็นไลบรารี (library) ที่ใช้เรียกในส่วนของการพัฒนา ส่วนติดต่อผู้ใช้เป็นกราฟิก ทีเค พัฒนาโดย John Qusterhout ซึ่งในตอนแรกมีไว้ใช้สำหรับภาษาทีซีแอล (TCL) เพื่อใช้ในการเขียนส่วนติดต่อผู้ใช้ที่เป็นกราฟิก ต่อมา Nick Ing-Simmons ทำการผนวก ทีเค เวอร์ชัน 4 เป็นส่วนเพิ่มเติมของ เพิร์ล เรียกโดยทั่วไปว่า เพิร์ล ทีเค และยังได้เพิ่มวิดเก็ด (widgets) ใหม่ๆ เข้าไปเพื่อเพิ่มความสามารถของ ทีเค ใช้ในการพัฒนาส่วนติดต่อกับผู้ใช้งาน

2.12 การใช้งานภาษาไทยบนระบบเอกซ์วินโดว์ (X-Windows)

2.12.1 มาตรฐานของรหัสตัวอักษรภาษาไทย [18]

2.12.1.1 TIS-620 หรือมอก. 620 หรือที่เรียกกันทั่วไปว่า รหัส สมอ. เป็นมาตรฐานของรหัสตัวอักษร (Charset Code) ที่ใช้บนระบบคอมพิวเตอร์ ซึ่งกำหนดโดยสำนักงานมาตรฐานอุตสาหกรรม หรือ สมอ. (Thai Indrustial Standards Institute [TISI]). TIS-620 เป็นรหัสตัวอักษรที่ต่อเพิ่มจากรหัสตัวอักษรของ ISO-646 ซึ่งเป็น รหัสตัวอักษรแบบ 7 bit คล้ายกับรหัสแอสกี (ASCII) มาตรฐาน TIS-620 ตัวแรกคือ TIS-620 2529 (1986) ซึ่งได้มีการแก้ไขเพิ่มเติมอีก ในปี 2533 เป็น TIS-620 2533 (1990) เพื่อเพิ่มเนื้อหาบางส่วนให้สอดคล้องกับ ISO/IEC 2022 แต่ตารางรหัสตัวอักษรทั้งหมดยังคงเดิม ปัจจุบัน GNU C library (GLIBC) ได้สนับสนุนมาตรฐาน TIS-620 ในการใช้งาน สำหรับกับท้องถิ่นประเทศไทยและภาษาไทย ภายใต้ชื่อ th_TH (th_TH.TIS-620)

2.12.1.2 ISO8859-11 รหัสตัวอักษรแบบ 8 bit ของ TIS-620 คล้ายกับ กับรหัสตัวอักษรในระบบ ISO/IEC 8859 มาก เนื่องจาก สมอ. (TISI) นั้นไม่ประสบความสำเร็จมากนักในการกระตุ้นให้ TIS-620 เป็นมาตรฐานจึงได้คิดจะใส่ไว้ในระบบ ISO/IEC 8859 แทนเพื่อให้ในระบบอุตสาหกรรมต่าง ๆ หันมาใช้ตารางรหัสภาษาไทยตามมาตรฐานมากขึ้น ตารางนี้ได้รับการใส่ไว้ในส่วนที่ 11 (Part 11) ของมาตรฐาน ISO/IEC 8859 ถึงแม้จะมีการปฏิเสธการใช้มาตรฐานนี้เนื่องจากภาษาไทยนั้นต่างจากภาษาแบบละติน ตรงที่มีต้องมีการประกอบตัวอักษรเข้าด้วยกัน แต่ในภายหลังก็มีการผลักดันให้ ISO/IEC 8859 Part 11 ผ่านในที่ประชุม ISO และประกาศเป็นทางการในปลายปี พ.ศ. 2544

2.12.1.3 ISO-10646-1 โปรแกรมในปัจจุบันได้เริ่มออกแบบให้สามารถใช้ได้หลายภาษา โดยใช้ มาตรฐานของตัวอักษร ของ ISO/IEC 10646 (Universal Multi-octet Coded Character Set - UCS) ซึ่งเป็นระบบสำหรับเก็บข้อมูลตัวอักษรสากลในระบบ 8bit (หรือ byte) ซึ่งอาจอยู่ในรูป 8 bit หลาย ๆ ตัวต่อกัน และรู้จักกันดีในชื่อ Unicode UCS หรือ UTF-8

2.12.2 แบบตัวอักษรชนิดต่าง ๆ

2.12.2.1 แบบตัวอักษรชนิดบิตแมพ (Bitmapped) สำหรับเอกซ์วินโดว์จะมีสองรูปแบบ คือ BDF และ PCF ฟอนต์ BDF จะมีรูปแบบเป็นแฟ้มข้อความ สามารถใช้เอดิเตอร์ (editor) เปิดและแก้ไขได้ ส่วน PCF จะเป็นแฟ้มไบนารีที่คอมไพล์แล้ว ซึ่งจะมีขนาดเล็กกว่า

2.12.2.2 แบบตัวอักษรชนิด Postscript Type 1 เป็นเทคโนโลยีฟอนต์เวกเตอร์จากค่าย Adobe และนิยมใช้มากในงานพิมพ์

2.12.2.3 แบบตัวอักษรชนิด True Type (TTF) เป็นฟอนต์เวกเตอร์ที่นิยมใช้มากในไมโครซอฟท์วินโดว์ โดยเฉพาะในงานสร้างเอกสารแบบ What You See Is What You Get (WYSIWYG) สำหรับเอกซ์วินโดว์แล้วก็ได้มีความพยามยามสนับสนุนฟอนต์ทรูไทร์ (TrueType Font) มาเป็นลำดับ

2.12.3 การแสดงผลและแบบตัวอักษร

เอกซ์วินโดว์สนับสนุนการใช้แบบตัวอักษรหลายชนิด โดยเฉพาะรุ่นใหม่ ๆ นั้น สามารถใช้งาน แบบตัวอักษรที่เป็นทรูไทร์ (True Type) ได้ด้วย (ฟอนต์ที่ใช้บน ไมโครซอฟท์วินโดว์) แต่อย่างไรก็ตามการใช้ภาษาไทยก็ยังทำได้ไม่ดีนัก เนื่องจากโปรแกรมต่าง ๆ บน เอกซ์วินโดว์ยังใช้งานรหัสตัวอักษรภาษาไทย เช่น TIS-620 ไม่ได้

2.13 เอกสารและงานวิจัยที่เกี่ยวข้อง

2.13.1 เอกสารตรวจสอบความปลอดภัยบนยูนิกซ์เวอร์ชันสอง [13]

หน่วยงานเซิร์ต CERT (Computer Emergency Response Team) และหน่วยงานเอยูเอสเซิร์ต AusCERT (Australian Computer Emergency Reponses Team) ได้ร่วมมือกันจัดทำเอกสารตรวจสอบความปลอดภัยบนยูนิกซ์เวอร์ชันสองเป็นเวอร์ชันล่าสุด ทำการเผยแพร่ตั้งแต่วันที่ 8 ตุลาคม ค.ศ. 2001 ซึ่งจะมีรายละเอียดเกี่ยวกับการแก้ไขและข้อแนะนำในการป้องกันความปลอดภัยของระบบปฏิบัติการยูนิกซ์ โดยสามารถจำแนกเป็นหมวดการตรวจสอบพื้นฐานของระบบปฏิบัติการได้ดังนี้คือ

2.13.1.1 **หมวดบริการเครือข่าย (Network Service)** เป็นส่วนการตรวจสอบความปลอดภัยเกี่ยวกับแฟ้มที่เกี่ยวข้องในการทำงานด้านเครือข่ายโดยมีรายละเอียดดังนี้

แฟ้ม /etc/inetd.conf

- ต้องแน่ใจว่าบิตอนุญาตในการเข้าถึงแฟ้มถูกตั้งค่าเป็น 600
- ต้องแน่ใจว่าเจ้าของแฟ้มถูกตั้งเป็น root
- ทำการยกเลิกบริการต่างๆที่ไม่ต้องการใช้ ทำการยกเลิกบริการทั้งหมดโดยการใส่เครื่องหมาย “#” ไว้ต้นบรรทัดทุกๆ บรรทัด
- หลีกเลี่ยงการใช้คำสั่ง “tftp” ซึ่งอาจเป็นแหล่งกำเนิดของความไม่ปลอดภัย

แฟ้ม /etc/hosts.equiv

- ตรวจสอบว่ามีความจำเป็นต้องใช้แฟ้ม /etc/hosts.equiv ซึ่งถ้ามีการเรียกคำสั่ง “r” แฟ้มนี้จะเป็นการอนุญาตให้ระบบเชื่อถือ host อื่นได้ หากไม่ได้มีการเรียกคำสั่ง “r” หรือไม่ต้องการไว้ใจระบบอื่นๆ แนะนำว่าไม่ควรมีแฟ้มนี้อยู่ในระบบ
- หากจำเป็นต้องการใช้แฟ้ม /etc/hosts.equiv ควรแน่ใจว่าไม่มีเครื่องหมาย “+” ที่ใดก็ตามในแฟ้มซึ่งอาจอนุญาตให้ผู้ใช้งานเข้าถึงระบบได้
- ควรแน่ใจว่า บิตอนุญาตในการเข้าถึงแฟ้มถูกตั้งมีค่าเท่ากับ 600
- ควรแน่ใจว่าเจ้าของแฟ้มถูกตั้งเป็นรูท

แฟ้ม /etc/netgroup

- ควรแน่ใจว่าบิตอนุญาตในการเข้าถึงแฟ้มนี้มีค่าเท่ากับ 600
- ควรแน่ใจว่าเจ้าของแฟ้มถูกตั้งเป็นรูท

แฟ้ม $HOME/.rhost

- ควรแน่ใจว่าไม่มีผู้ใช้งานคนใดมีแฟ้ม .rhosts ในไดเรกทอรีบ้าน แฟ้มเหล่านี้มีความเสี่ยงต่อความปลอดภัยมากกว่าแฟ้ม /etc/hosts.equiv เพราะผู้ใช้งานคนหนึ่งสามารถสามารถสร้างแฟ้มนี้ได้หนึ่งแฟ้ม ควรพิจารณาเป็นกรณีในการใช้แฟ้มนี้

แฟ้ม /etc/services

- ควรแน่ใจว่าบิตอนุญาตในการเข้าถึงแฟ้มนี้มีค่าเท่ากับ 644
- ควรแน่ใจว่าเจ้าของแฟ้มถูกตั้งเป็นรูท

แฟ้ม /etc/hosts.lpd

- ควรแน่ใจว่าบิตอนุญาตในการเข้าถึงแฟ้มนี้มีค่าเท่ากับ 600
- ควรแน่ใจว่าเจ้าของแฟ้มถูกตั้งเป็นรูท

แฟ้ม /etc/securetty

- ควรแน่ใจว่าบิตอนุญาตในการเข้าถึงแฟ้มนี้มีค่าเท่ากับ 600
- ควรแน่ใจว่าเจ้าของแฟ้มถูกตั้งเป็นรูท
- ไม่อนุญาตให้เข้าถึงระบบด้วย รูท จากระยะไกล

บริการ Trivial FTP (tftp)

- ทำการยกเลิกบริการทีเอฟทีพีถ้าไม่มีความจำเป็น

บัญชีผู้ใช้ ยูยูซีพี

- ทำการยกเลิก บัญชีผู้ใช้ยูยูซีพีหากไม่มีความจำเป็น
- ลบแฟ้ม .rhosts ที่อยู่ในไดเรกทอรีบ้านของบัญชีผู้ใช้ ยูยูซีพี

2.13.1.2 **หมวดบิตอนุญาตของแฟ้ม** (File System Security) เป็นส่วนการตรวจสอบความปลอดภัยของแฟ้มต่างๆ ในระบบในด้านของบิตอนุญาต และความเป็นเจ้าของแฟ้มที่เหมาะสม

- ตรวจสอบให้แน่ใจว่าเคอร์เนล ถูกครอบครองโดยรูท และมีค่าบิตอนุญาตเป็น 644
- ตรวจสอบให้แน่ใจว่าไดเรกทอรี /etc /bin /sbin /usr/bin /var/tmp ถูกครอบครองโดยรูท และมีค่าบิตอนุญาตเป็น 644
- ตรวจสอบให้แน่ใจว่าไดเรกทอรี /tmp ถูกครอบครองโดยรูท และมีค่าบิตอนุญาตเป็น 1777
- ตรวจสอบให้แน่ใจว่าไม่มี world writable file ในไดเรกทอรีที่ไม่ได้ถูกสร้างจากระบบ
- ตรวจสอบแฟ้มซึ่งมีการกำหนดค่า suid และ sgid bit ซึ่งควรจะเป็นแฟ้มตามค่าปริยายของระบบ

- ตรวจสอบให้แน่ใจว่าทุกแฟ้มที่อยู่ในไดเรกทอรี /dev เป็นแฟ้มชนิดพิเศษ กล่าวคือ ตำแหน่งแรกของบิตอนุญาตของแฟ้มดังกล่าวจะเป็นตัวอักษร เช่น “c” สำหรับ character และ “b” สำหรับ block

2.13.1.3 **หมวดความปลอดภัยของบัญชีผู้ใช้** (Account Security) เป็นส่วนการตรวจสอบความปลอดภัยของบัญชีผู้ใช้ในระบบ

- ตรวจสอบให้แน่ใจว่า บัญชีผู้ใช้ในระบบมีรหัสผ่าน
- ทำการยกเลิกบัญชีผู้ใช้ที่ไม่มีรหัสผ่าน
- ตรวจสอบให้แน่ใจว่ามีการใช้ ชาโดว์พาสเวิสด์ (shadow password)ในระบบ การทำชาโดว์พาสเวิสด์ จะจำกัดการเข้าถึงของผู้ใช้ที่มีการเข้ารหัสไว้
- ทำการยกเลิกผู้ใช้งานปริยายทั้งหมด ซึ่งมากับระบบปฏิบัติการซึ่งควรมีการตรวจสอบทุกครั้งหลังจากมีการ update หรือติดตั้ง
- ไม่ควรล็อกอินด้วย รูท ข้ามเครือข่าย
- ควรใช้วิธีเรียกคำสั่ง “su” ของผู้ใช้ธรรมดาในการล็อกอินเป็นรูทแทน
- ควรแน่ใจว่าไม่มี “.” อยู่ในเส้นทางค้นหาคำสั่งของรูท
- ควรแน่ใจว่ารูท ไม่มีแฟ้ม ~/.rhosts อยู่

2.13.1.4 **หมวดการพิสูจน์ตน** (Authentication) เป็นส่วนการตรวจสอบ การใช้งานระบบชาโดว์พาสเวิสด์ (Password Shadowing) หรือผลิตภัณฑ์อื่น ๆ ที่เกี่ยวข้องการทำชาโดว์พาสเวิสด์

2.13.1.5 **หมวดการเฝ้าตรวจระบบ** (System Monitoring) เป็นส่วนการตรวจสอบข้อมูลระบบโดยใช้โปรแกรมซิสล็อกดีมอน ในการบันทึกเหตุการณ์ในเครือข่าย และเก็บไว้ในเครื่องอื่นๆ ด้วยถ้าเป็นไปได้

2.13.1.6 **การตรวจสอบบริการพื้นฐาน** เป็นการตรวจสอบบริการพื้นฐานที่มากับระบบ ได้แก่ บริการด้าน เมล์เซิร์ฟเวอร์ เว็บเซิร์ฟเวอร์ และเอฟทีพีเซิร์ฟเวอร์

บริการเอฟทีพีเซิร์ฟเวอร์

- ไม่ควรเปิดบริการเอฟทีพีแบบนิรนามหากไม่มีความจำเป็น หรือยกเลิกบัญชีผู้ใช้ เอฟทีพี

- ควรแน่ใจว่าตั้งค่าบิตอนุญาตของ ไดเรกทอรีบ้านของเอฟทีพี (FTP home directory) มีค่าเป็น 555 และตั้งค่าเจ้าของเป็นรูท
- ควรแน่ใจว่าไม่มี แฟ้ม .rhost ในไดเรกทอรีบ้านของเอฟทีพี
- ควรแน่ใจว่าไม่มีแฟ้มหรือไดเรกทอรีใดในพื้นที่ของ เอฟทีพีที่ทุกคนเขียนทับได้
- ควรแน่ใจว่าบิตอนุญาตของแฟ้มใน ~/ftp/bin มีค่าเป็น 111 และถูกครอบครองโดยรูท
- ควรแน่ใจว่าบิตอนุญาตของแฟ้มใน ~/ftp/etc มีค่าเป็น 444 และถูกครอบครองโดยรูท
- ควรแน่ใจว่าบิตอนุญาตของแฟ้มใน /usr/spool/mail/ftp มีค่าเป็น 400 และถูกครอบครองโดยรูท

บริการเว็บเซิร์ฟเวอร์

- ควรให้ รูทครอบครองไดเรกทอรีของสคริปต์ เช่น cgi-bin และตั้งค่าบิตอนุญาตเป็น 751 เพื่อป้องกันไม่ให้ผู้ใช้งานเข้าดูรายละเอียดระหว่างที่ดีมอนรันสคริปต์ทำงานภายใต้ไดเรกทอรีนั้น
- ควรตั้งให้รูทเป็นเจ้าของไดเรกทอรีเว็บเซิร์ฟเวอร์
- ควรแน่ใจแฟ้มคอนฟิก แฟ้มล็อก และแฟ้มไบนารีทั้งหมดที่ใช้สำหรับเว็บเซิร์ฟเวอร์ทั้งหมดถูกครอบครองโดยรูท และมีค่าบิตอนุญาตเป็น 755
- ควรใช้เครื่องเว็บเซิร์ฟเวอร์เพื่อจุดประสงค์ของการให้บริการด้านเว็บเท่านั้น กล่าวคือไม่ควรติดตั้งบริการอื่นๆ เช่น เมล์ ดีเอนเอส และทำการลบ บัญชีผู้ใช้อื่นๆ ที่ไม่จำเป็นออก

บริการเมล์เซิร์ฟเวอร์

- ควรแน่ใจว่าได้ทำการติดตั้ง แพ็ทช์ (patch) ล่าสุด เพื่อป้องกันแหล่งที่อาจจะเป็นช่องโหว่ต่างๆ
- ควรแน่ใจว่าโปรแกรมเซนเมล์ จะไม่ทำการรีเลย์จากเซิร์ฟเวอร์ที่ไม่รู้จัก ซึ่งจะเป็นการป้องกันไม่ให้เซนเมล์ถูกใช้ไปในทางที่ไม่เหมาะสม
- ควรแน่ใจว่าไดเรกทอรีซึ่งใช้เก็บแฟ้มคอนฟิก มีค่าบิตอนุญาตเป็น 755 และถูกครอบครองโดยรูท

หน่วยงานเซิร์ตแห่งประเทศสหรัฐอเมริกา ซึ่งมีที่ทำการ ณ มหาวิทยาลัยคาร์เนกีเมลลอน เป็นศูนย์กลางการแก้ปัญหาภัยคุกคามบนอินเตอร์เน็ตของอเมริกา โดยเซิร์ทได้เผยแพร่คำแนะนำ

(Advisories) ข้อมูลของความไม่ปลอดภัยและข้อบกพร่องในระบบคอมพิวเตอร์ บนเว็บไซต์ http://www.cert.org

2.13.2 การพัฒนาโปรแกรมตรวจสอบความมั่นคงสำหรับยูนิกซ์ [10]

โปรแกรมช่วยตรวจสอบความปลอดภัยของระบบปฏิบัติการยูนิกซ์ตระกูลบีเอสดี 4.2 และซิสเต็มแฟ้มใช้ตรวจสอบระบบความปลอดภัยในแบบเท็กซ์โหมด และจะแสดงข้อความให้ทราบถึงจุดที่หละหลวม หรือจุดที่น่าสงสัย แต่จะไม่เข้าไปแก้ไขในส่วนนั้นโดยโปรแกรมจะแนะนำวิธีแก้ไขให้และผู้ใช้งานจะดำเนินการแก้ไขด้วยตนเอง

2.13.3 SATAN (Security Administrator Tool For Analyzing Networks) [14]

เครื่องมือที่ใช้จากที่อื่น (Remote site) ในการตรวจสอบและพิสูจน์ทราบถึงความล่อแหลมของระบบ บนเครือข่ายไอพี เป็นโปรแกรมฟรีแวร์ (freeware) ที่มีขีดความสามารถสูงที่ช่วยในการหาจุดอ่อนทางด้านความปลอดภัยของระบบ ปัจจุบันไม่ได้เป็นที่นิยมใช้แล้ว

2.13.4 SAINT (Security Administrator's Integrated Network Tool) [15]

เป็นเครื่องมือที่ช่วยหาและพิสูจน์ทราบถึงความล่อแหลมของระบบเครือข่ายไอพี โดยรวบรวมข้อมูลของการรีโมด (Remote) ผ่านเครือข่ายและทำการตรวจสอบ จำแนก วิเคราะห์ความเป็นไปได้ที่จะเกิดปัญหาและข้อบกพร่อง กับบริการเครือข่ายประเภทต่างๆ โดยดูจากข้อมูลที่แสดงออกมาจากโปรแกรม ซึ่งโปรแกรมเซ็นต์ (SAINT) นี้เป็นโปรแกรมที่พัฒนาเพิ่มเติมความสามารถจากโปรแกรมซาตานในอดีต

2.13.5 COPS (Computerized Oracle and Password System) [16]

COPS เป็นเครื่องมือที่ทำหน้าที่ ช่วยตรวจสอบสถานะความปลอดภัยบนระบบยูนิกซ์ ซึ่งมีใช้ตั้งแต่ปี 1990 โดยจุดที่ตรวจสอบ จะเป็นจุดในเรื่องความปลอดภัยโดยทั่วๆ ไป ไม่ว่าจะเป็นเรื่องของการตรวจสอบสิทธิ์ ในการเข้าถึงแฟ้มข้อมูลต่างๆ ตรวจสอบการ SUID (Set User ID) การตรวจสอบการกำหนดรหัสผ่าน ของผู้ใช้งานขั้นต่ำให้มีความปลอดภัย โดยจุดประสงค์หลักคือ เพื่ออำนวยความสะดวกให้กับผู้ใช้งานบนระบบยูนิกซ์ ที่จะง่ายในการค้นหาความล่อแหลมที่จะเกิดปัญหาด้านความปลอดภัยกับระบบได้ ทั้งนี้ผู้ใช้งานจะต้องเป็นผู้แก้ไขปัญหาจากการตรวจสอบจุดที่ล่อแหลมนั้นด้วยตนเอง

2.13.6 CVE (Common Vulnerabilities and Exposure) [17]

CVE (Common Vulnerabilities and Exposure) ซึ่งเป็นของ Mitre Corporation (cve.mitre.org) ทำหน้าที่รวบรวมข้อบกพร่องจากแหล่งต่างๆ และจัดหมวดหมู่ให้อยู่ในฐานการอ้างอิงเดียวกัน เพื่อจะได้ตรงกันและลดปัญหาความซ้ำซ้อนของวิธีการอ้างอิงโดยใช้ชื่อที่แตกต่างกันไปของแต่ละหน่วยงาน ตัวอย่าง เอกสารหมายเลข CAN-2003-0694 หัวข้อ Buffer Overflow in Send mail แสดงรายละเอียดช่องโหว่ที่พบในโปรแกรมเซนเมล์ อนุญาตให้ผู้โจมตีจากภายนอกส่งโปรแกรมเข้ามาทำงานในระดับสิทธิ์ของโปรแกรมเซนเมล์ ซึ่งโดยปกติจะเป็นรูท หรือเอกสารหมายเลข CAN-2002-1219 หัวข้อ Multiple Vulnerabilities in BIND แสดงรายละเอียดช่องโหว่ของ BIND ซึ่งมีผลกระทบแตกต่างกันออกไป โดยช่องโหว่เหล่านี้จะอนุญาตให้ผู้โจมตีทำการเรียกโค้ดโดยใช้สิทธิพิเศษของผู้ที่เรียกบริการเนมดีมอน (named) ปกติจะเป็นรูท หรือของแอพพลิเคชันอาจจะอนุญาตให้ผู้โจมตีทำการรบกวนการทำงานของดีเอนเอสที่รันอยู่บนเซิร์ฟเวอร์ เป็นต้น

บทที่ 3
การวิเคราะห์และออกแบบระบบ

3.1 การวิเคราะห์ระบบ

ในระบบปฏิบัติการประเภทยูนิกซ์ โดยทฤษฎีด้านการรักษาความปลอดภัยพื้นฐานแล้วจะเกี่ยวข้องกับแฟ้มข้อมูลและผู้ใช้งานเป็นหลัก เนื่องจากยูนิกซ์มองทุกอย่างในระบบเป็นลักษณะแฟ้มข้อมูล โดยมีผู้ใช้งานเป็นผู้เรียกแฟ้มข้อมูลในระบบเพื่อจุดประสงค์ในการทำงานต่างๆ จึงทำให้การตรวจสอบหรือป้องกันภายในตัวระบบ จะเกี่ยวข้องกับการตรวจสอบผู้ใช้งานและการตรวจสอบเกี่ยวกับแฟ้มข้อมูลในระบบ โดยปกติระบบปฏิบัติการยูนิกซ์เองจะมีกลไกการอารักขาแฟ้มข้อมูล ซึ่งจะช่วยปกป้องการเข้าถึงแฟ้มข้อมูล และผู้ใช้งานจะมีรหัสผ่านในการตรวจสอบสิทธิของผู้ใช้งานในการเข้าถึงแฟ้มข้อมูลในระบบ แต่เนื่องจากกลไกการป้องกันเหล่านี้มักถูกผู้ไม่หวังดีทำการเปลี่ยนแปลงแก้ไขค่า ทำให้เกิดช่องโหว่ในระบบซึ่งไม่ปลอดภัย จึงจำเป็นต้องมีการตรวจสอบค่าคอนฟิกและกลไกการป้องกันในระบบให้มีความถูกต้องมากที่สุด โดยการวัดค่าความถูกต้องจะใช้มาตรฐานความปลอดภัยพื้นฐานบนยูนิกซ์ จากเอกสารตรวจสอบความปลอดภัยบนยูนิกซ์เวอร์ชันสอง ซึ่งเป็นเอกสารแสดงรายละเอียดขั้นตอนการปรับปรุงระบบความปลอดภัยของระบบปฏิบัติการประเภทยูนิกซ์ ตามมาตรฐานของหน่วยงานเซิร์ต และหน่วยงานเอยูเอสเซิร์ต ซึ่งรายละเอียดค่าความถูกต้องจากเอกสารนี้ จะเป็นการแนะนำข้อปฏิบัติเกี่ยวกับ การกำหนดค่าคอนฟิกของระบบ บิตอนุญาตที่เหมาะสมเกี่ยวกับแฟ้มข้อมูลและบัญชีผู้ใช้งานพื้นฐานในระบบเพื่อให้เกิดความปลอดภัย แต่ทั้งนี้ก็ขึ้นอยู่กับสภาพแวดล้อมของระบบนั้นๆ ว่าเหมาะสมหรือสามารถที่จะทำตามค่าที่แนะนำตามเอกสารตรวจสอบความปลอดภัยได้มากน้อยแค่ไหน ซึ่งอาจแตกต่างกันโดยจุดประสงค์ของการใช้งานระบบและนโยบายของผู้ดูแลระบบนั้นๆ โดยรายการตรวจสอบความปลอดภัยพื้นฐาน จะเป็นเกณฑ์ในการวัดค่าความปลอดภัยของระบบเพื่อให้ระบบมีการทำงานที่มีเสถียรภาพมากที่สุด

3.2 แนวคิดการออกแบบระบบ

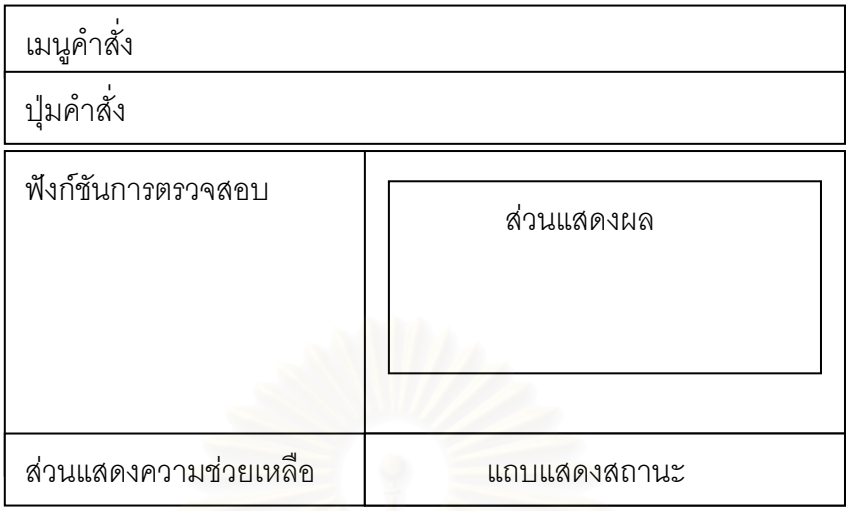
แนวคิดในการออกแบบโปรแกรมวิเคราะห์ความปลอดภัยแบ่งเป็นโมดูล โดยแต่ละโมดูลจะมีหน้าที่ในการตรวจสอบความปลอดภัยในหมวดต่างๆ ซึ่งการตรวจสอบความถูกต้องจะอิงตามเอกสารตรวจสอบความปลอดภัยบนยูนิกซ์เวอร์ชันสอง ในหมวดการตรวจสอบความปลอดภัยพื้นฐานที่เกี่ยวข้องของระบบปฏิบัติการลีนุกซ์เรดแฮต โดยการออกแบบโปรแกรมตรวจสอบความ

ปลอดภัยของระบบลีนุกซ์เรดแฮตนี้ ได้แยกออกเป็นสองส่วนด้วยกัน คือ โปรแกรมส่วนติดต่อกับผู้ใช้ และชุดโปรแกรมที่ใช้ในการตรวจสอบความปลอดภัย โดยในบทนี้จะกล่าวถึงการออกแบบและพัฒนาโปรแกรมส่วนติดต่อผู้ใช้

3.3 การออกแบบโปรแกรมส่วนติดต่อกับผู้ใช้

ในการออกแบบส่วนติดต่อกับผู้ใช้ พัฒนาโดยใช้ เพิร์ลทีเค ซึ่งเป็นทูลคิดในการสร้างโปรแกรมแบบกราฟิกของภาษา เพิร์ล บนสภาพแวดล้อมเอกซ์วินโดว์ เพิร์ลทีเค มีรูปแบบการเขียนโปรแกรมเป็นแบบภาษาสคริปต์ โดยจะมีรูปแบบคล้ายกับภาษาเพิร์ลปกติแต่จะมีตัวแปรและฟังก์ชัน ในส่วนของการสร้างส่วนประกอบกราฟิก (Graphic Component) เพิ่มเข้ามาได้แก่ หน้าต่างหลัก (Main Windows) ปุ่ม (Button) ลาเบล (Label) เช็คบอกส์ (Checkbox) เฟรม (Frame) เรดิโอ บัททอน (Radio Button) ฯลฯ โดยในการเรียกใช้จะมีการประกาศให้เพิร์ลคอมไพเลอร์ (perl compiler) ทราบก่อนว่า มีการใช้ เพิร์ลทีเค เพื่อเรียกส่วนประกอบกราฟิกต่างๆ ที่ต้องการใช้งาน ลักษณะการประกาศตัวแปรแบบกราฟิกของ เพิร์ลทีเค จะมีลักษณะเรียงตามลำดับชั้นของตัวแปรชนิดกราฟิกที่ประกาศ โดยมีการอ้างอิงกับตัวแปรก่อนหน้าที่เรียกใช้ เช่น ถ้ามีการประกาศตัวแปร หน้าต่างหลัก เป็นลำดับชั้นที่หนึ่ง ภายใต้ตัวแปรหน้าต่างหลัก จะประกอบด้วย ตัวแปรเฟรม ซึ่งอิงกับตัวแปรหน้าต่างหลัก เพื่อเป็นการบอกว่าตัวแปรเฟรมอยู่ภายใต้ตัวแปรหน้าต่างหลักเป็นลำดับที่สอง หากต้องการเพิ่มส่วนประกอบกราฟิก ในตัวแปรเฟรม เช่น ลาเบล ก็ต้องประกาศตัวแปรชนิด ลาเบล โดยอิงกับตัวแปรชนิด เฟรม เป็นลำดับชั้นที่สาม หากต้องการเพิ่มส่วนประกอบ กราฟิกอื่นๆ เพิ่มเติมจำเป็นต้องมีการประกาศตัวแปรแบบกราฟิกเพิ่มในลักษณะที่กล่าวมา ซึ่งอาจเป็นลำดับชั้นเดียวกันกับตัวแปรก่อนหน้าหรือต่างลำดับชั้นกันก็ได้เพียงแต่ต้องมีการอ้างชื่อตัวแปรก่อนหน้า การประกาศตัวแปรแบบอิงกันเป็นลำดับชั้นนี้ เพื่อให้เพิร์ลคอมไพเลอร์ทราบว่าตัวแปรที่ประกาศทั้งหมดเป็นส่วนเดียวกัน โดยปกติมักมีการเรียกใช้ตัวแปรแบบกราฟิกไม่เกินสามลำดับ ใช้ในส่วนประกอบเดียวกัน ตัวอย่างเช่น ในหน้าต่างหลักประกอบด้วย เฟรม และใน เฟรม อาจจะมีส่วนประกอบกราฟิก ชนิด ลาเบล และชนิดบัททอน ซึ่งทั้งหมดอยู่ในส่วนประกอบเดียวกัน นอกจากนี้ตัวแปรชนิดกราฟิกของ เพิร์ลทีเค สามารถที่จะกำหนดคุณสมบัติที่จะแสดงผลในโหมดกราฟิก เช่น สี ขนาด ฟอนต์ หรือความสามารถในการเรียกโปรแกรมย่อย (sub program) ของเพิร์ลสคริปต์ ทั้งนี้การกำหนดคุณสมบัติต่างๆ ยังแตกต่างกัน ขึ้นอยู่กับชนิดของตัวแปรแบบกราฟิกที่เรียกใช้ด้วย

โดยในการพัฒนาส่วนติดต่อผู้ใช้สามารถแสดงรูปแบบดังรูปที่ 3.1



รูปที่ 3.1 แบบโครงสร้างของส่วนติดต่อผู้ใช้

โดยโปรแกรมที่ออกแบบมีลักษณะดังต่อไปนี้

1. มีส่วนติดต่อผู้ใช้ลักษณะเป็นกราฟิก ซึ่งทำงานบนสภาพแวดล้อมเอกซ์วินโดว์
2. แบ่งเป็นส่วนของฟังก์ชันที่ใช้ในการเลือกตรวจสอบความปลอดภัย และส่วนแสดงผลการตรวจสอบ
3. มีในส่วนของเมนูคำสั่งและปุ่มคำสั่งได้แก่ เมนูคำสั่ง จะแสดงเอกสารความต้องการของระบบและเอกสารอ้างอิงความถูกต้องของการตรวจสอบความปลอดภัยบนยูนิกซ์ ส่วนปุ่มคำสั่ง จะใช้ในการเริ่มต้นการทำงานและหยุดการทำงานของโปรแกรมในแบบเบื้องหลัง (Daemon Program) ใช้แสดงผลล็อกของระบบ การกำหนดค่าที่ใช้ในการตรวจสอบล็อก และคำสั่งออกจากโปรแกรม
4. โปรแกรมตรวจสอบมีทั้งรุ่นที่แสดงผลที่เป็นภาษาอังกฤษและรุ่นที่แสดงผลภาษาไทย

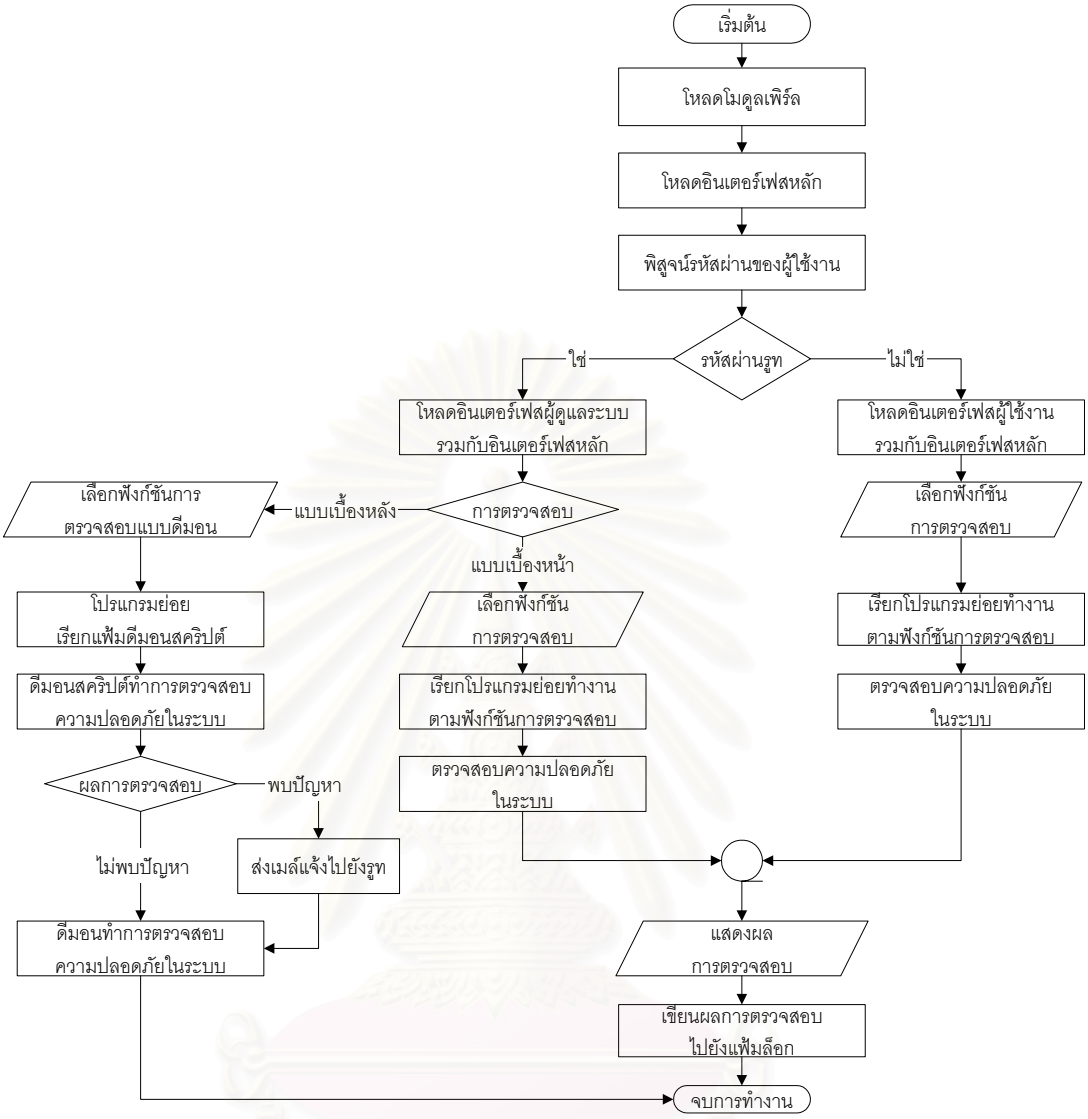
การออกแบบโปรแกรมส่วนติดต่อกับผู้ใช้ได้แยกออกเป็น 2 ลักษณะได้แก่ โหมดตรวจสอบความปลอดภัยของผู้ดูแลระบบ และโหมดตรวจสอบความปลอดภัยสำหรับผู้ใช้งาน

3.4 การออกแบบการทำงานของโปรแกรม

ในการออกแบบสามารถแบ่งเป็นขั้นตอนการทำงานของโปรแกรม และสถาปัตยกรรมระบบ

3.4.1 ขั้นตอนการทำงานของโปรแกรม

ในการทำงานของโปรแกรมสามารถแสดงในรูปแบบผังการทำงานได้ในรูปที่ 3.2



รูปที่ 3.2 ผังการทำงานของโปรแกรม

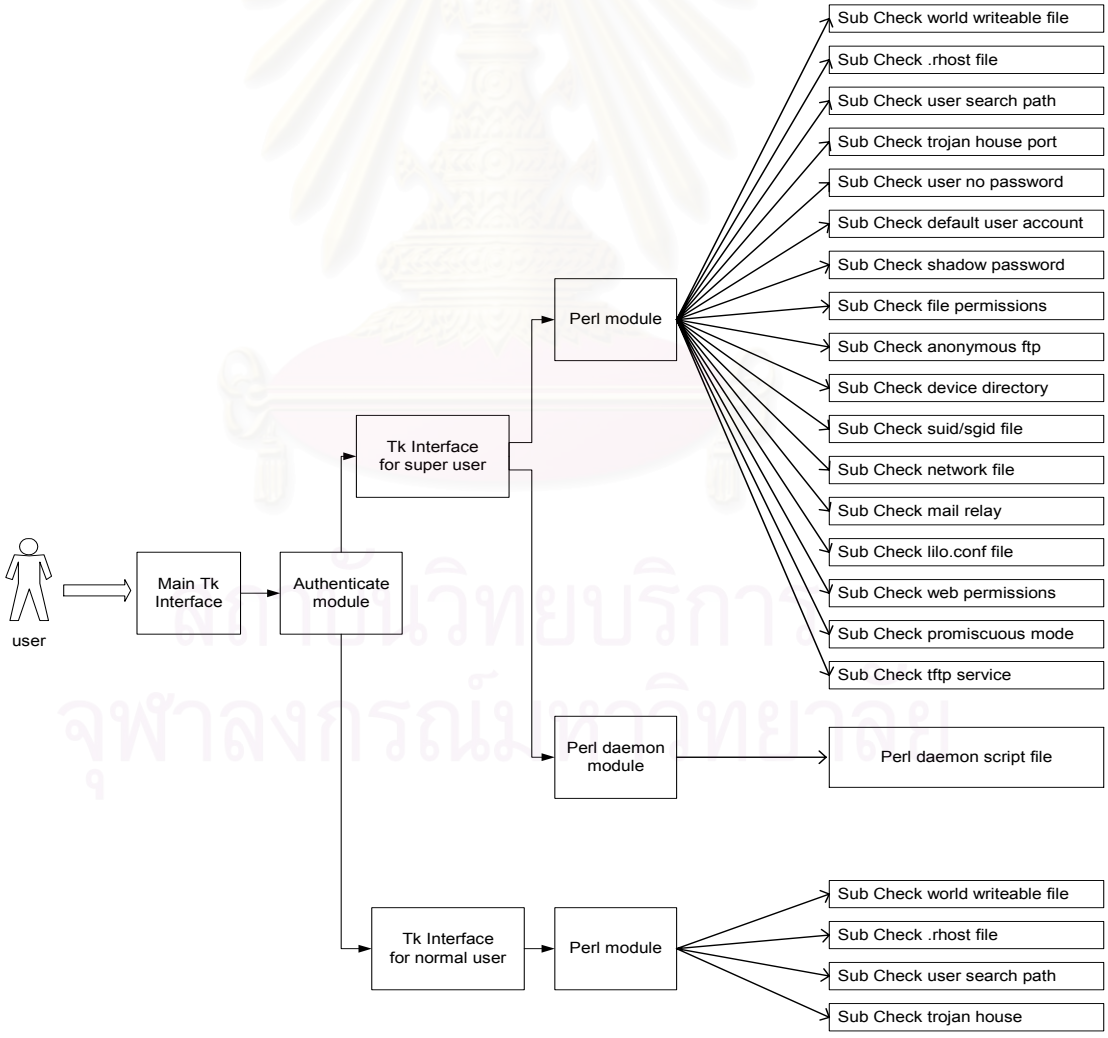
สามารถอธิบายขั้นตอนการทำงานของโปรแกรมตามรูปผังงานดังนี้

โปรแกรมจะทำการโหลดโมดูลที่จำเป็นในการใช้งาน ทั้งโมดูลแบบกราฟิกของทีเค และโมดูลที่ใช้ในการทำงานของเพิร์ล จากนั้นทำการสร้างส่วนติดต่อผู้ใช้หลัก เพื่อรองรับส่วนติดต่อผู้ใช้ลักษณะของผู้ใช้งานกับส่วนติดต่อผู้ใช้ลักษณะผู้ดูแลระบบ ซึ่งจะมารวม ประสานกับส่วนติดต่อผู้ใช้หลักหลังจากผ่านการพิสูจน์รหัสผ่านของผู้ใช้งาน ในกรณีที่ผู้ใช้งานที่ผ่านการพิสูจน์ไม่ใช่รูท ก็จะเข้าสู่ส่วนติดต่อผู้ใช้ลักษณะผู้ใช้งาน พร้อมทั้งแสดงฟังก์ชันการตรวจสอบ เมื่อผู้ใช้งานเลือกฟังก์ชันการตรวจสอบ โปรแกรมจะเรียกโปรแกรมย่อยตามฟังก์ชันนั้นๆ ตรวจสอบความปลอดภัยในระบบพร้อมทั้งแสดงผลการตรวจสอบ และทำการเขียนผลการตรวจสอบไปยัง

แฟ้มล็อกของโปรแกรม ในกรณีที่การพิสูจน์ผู้ใช้งานผลเป็นรูท โปรแกรมจะทำการโหลดส่วนติดต่อผู้ใช้ในลักษณะผู้ดูแลระบบ ผู้ใช้งานที่เป็นผู้ดูแลระบบสามารถเลือกการตรวจสอบทั้งในแบบเบื้องหน้าและการตรวจสอบในแบบเบื้องหลัง โดยการตรวจสอบในแบบเบื้องหน้าจะมีลักษณะเช่นเดียวกับลักษณะผู้ใช้งานแบบปกติ แต่จะมีฟังก์ชันการตรวจสอบในระบบส่วนของผู้ดูแลระบบเพิ่มเติมเข้ามา ส่วนการตรวจสอบในแบบเบื้องหลัง โปรแกรมตรวจสอบจะทำการเรียกแฟ้มดีมอนสคริปต์เพื่อทำงานในแบบดีมอน โดยสามารถเลือกว่าจะให้มีการตรวจสอบฟังก์ชันใดในแบบเบื้องหลัง ในกรณีเจอปัญหาจะทำการแจ้งเมล์ไปยังผู้ดูแลระบบได้แก่รูท

3.4.2 **สถาปัตยกรรมระบบ**

สถาปัตยกรรมระบบ จะแสดงโครงสร้างของระบบในโปรแกรมตรวจสอบความปลอดภัย โดยจะประกอบไปด้วยส่วนของทีเคอินเตอร์เฟสและเพิร์ลโมดูลต่างๆ ดังแสดงในรูปที่ 3.3



รูปที่ 3.3 สถาปัตยกรรมของระบบ

สถาปัตยกรรมในระบบประกอบด้วย

1. Main Tk Interface เป็นอินเตอร์เฟสหลักสร้างโดยทีเค ทำหน้าที่ประสานกับอินเตอร์เฟสผู้ดูแลระบบ (Interface for super user) หรือ อินเตอร์เฟสผู้ใช้งานปกติ (Interface for normal user) ซึ่งจะตรวจสอบโดยโมดูลการพิสูจน์ตน (Authenticate module) เพื่อเป็นส่วนติดต่อผู้ใช้

2. Authenticate module เป็นโมดูลในการพิสูจน์รหัสผ่านของผู้ใช้ ถ้าผู้ใช้งานเป็นรูท จะโหลดอินเตอร์เฟสผู้ดูแลระบบรวมกับอินเตอร์เฟสหลัก ถ้าผู้ใช้งานไม่ใช่รูท จะโหลดอินเตอร์เฟสผู้ใช้งานปกติรวมกับอินเตอร์เฟสหลัก

3. Tk Interface for super user เป็นอินเตอร์เฟสสำหรับผู้ดูแลระบบสร้างโดยทีเค ทำหน้าที่แสดงฟังก์ชันการตรวจสอบของผู้ดูแลระบบ

4. Tk Interface for normal user เป็นอินเตอร์เฟสสำหรับผู้ใช้งานปกติสร้างโดยทีเค ทำหน้าที่แสดงฟังก์ชันการตรวจสอบของผู้ใช้งานปกติ

5. Perl module เป็นโมดูลของฟังก์ชันการตรวจสอบสร้างโดยเพิร์ล เป็นโปรแกรมย่อยทำหน้าที่ตรวจสอบความปลอดภัยในระบบ จะถูกเรียกทำงานตามฟังก์ชันที่ผู้ใช้งานเลือกจากอินเตอร์เฟส

6. Perl daemon module เป็นโมดูลของฟังก์ชันการตรวจสอบแบบเบื้องหลังสร้างโดยเพิร์ล เป็นแฟ้มเพิร์ลสคริปต์ จะถูกเรียกทำงานตามฟังก์ชันที่ผู้ใช้งานเลือกจากอินเตอร์เฟส

3.5 การปรับเปลี่ยนส่วนติดต่อผู้ใช้งาน

ผู้ใช้งานในลักษณะของผู้ดูแลระบบ สามารถที่จะปรับเปลี่ยนรูปแบบเพิ่มหรือลดฟังก์ชันการตรวจสอบความปลอดภัยโดยการเรียกโปรแกรมเพิร์ลสคริปต์ ในการคอมไพล์ตัวโปรแกรมหลักใหม่ โดยจะมีการตอบคำถามเพื่อเลือก เพิ่มหรือลดฟังก์ชันที่จะใช้ในการตรวจสอบ ให้เหมาะสมตามความต้องการดังแสดงรายละเอียดในภาคผนวก ง หัวข้อการปรับเปลี่ยนฟังก์ชันการตรวจสอบ

บทที่ 4
การออกแบบและพัฒนาชุดโปรแกรมตรวจสอบความปลอดภัย

4.1 การออกแบบโปรแกรมตรวจสอบความปลอดภัย

ในการออกแบบชุดของโปรแกรมตรวจสอบความปลอดภัยได้คำนึงถึงสิ่งต่อไปนี้

1. มีแฟ้ม ที่ทำหน้าที่ค้นหาชื่อเต็ม (Absolute pathname) ของไดเรกทอรีต่างๆ ซึ่งในโมดูลของโปรแกรมที่มีการค้นหาแฟ้มที่กำหนดจะเรียกใช้ แฟ้มนี้ ทั้งนี้เพราะว่าระบบลีนุกซ์ต่างรุ่นกันจะมีโครงสร้างของไดเรกทอรีต่างกัน การที่กำหนดให้มีแฟ้มที่เก็บชื่อไดเรกทอรีและทุกโปรแกรมจะเรียกใช้แฟ้มนี้ เพื่อความสะดวกในการนำโปรแกรมไปทำงานยังระบบลีนุกซ์เครื่องอื่น แทนที่จะต้องไปแก้ไขทุกโปรแกรม เพื่อแก้ไขโครงสร้างไดเรกทอรีของระบบปฏิบัติการ โดยที่นี้ใช้แฟ้ม find.dir ซึ่งเก็บเส้นทางของไดเรกทอรีที่โปรแกรมใช้ในการค้นหาและแฟ้ม perms.dir ซึ่งเก็บต้นแบบของบิตอนุญาตของแฟ้มและไดเรกทอรีที่ใช้ในการเปรียบเทียบ แสดงรายละเอียดของแฟ้มใน ภาคผนวก จ ซึ่งอิงจากเอกสารตรวจสอบความปลอดภัยบนยูนิกซ์เวอร์ชันสอง

2. คำสั่งหรือโปรแกรมอรรถประโยชน์ที่ชุดของโปรแกรมเหล่านี้ใช้ จะเลือกใช้เฉพาะคำสั่งที่มีอยู่ในระบบลีนุกซ์ส่วนใหญ่เท่านั้น จะหลีกเลี่ยงการใช้คำสั่งที่มีอยู่ในลีนุกซ์เวอร์ชันใหม่ๆ ถึงแม้คำสั่งใหม่จะมีประสิทธิภาพมากกว่า เพื่อหลีกเลี่ยงปัญหาการเคลื่อนย้ายของโปรแกรม ไปยังเครื่องอื่น และในกรณีที่ต้องเลือกใช้คำสั่ง จะเลือกใช้คำสั่งภายในมากกว่าคำสั่งภายนอก เนื่องจากคำสั่งภายในทำงานได้รวดเร็วกว่าคำสั่งภายนอก

3. แต่ละส่วนของโปรแกรมที่ทำหน้าที่ตรวจสอบความปลอดภัยจะเป็นลักษณะโปรแกรมย่อย (sub program) เพื่อสะดวกแก่การเรียกใช้งาน และแก้ไข

4.2 หมวดของโปรแกรมตรวจสอบความปลอดภัย

4.2.1 หมวดโปรแกรมตรวจสอบความปลอดภัยสำหรับผู้ใช้

ตรวจสอบบิตอนุญาตของแฟ้มที่เหมาะสมในไดเรกทอรีบ้าน (Home directory) **ของผู้ใช้ ได้แก่ แฟ้มที่มีลักษณะเปิดบิตอนุญาตทั้งหมด** (World Writable File)

หลักการคือ ตรวจสอบแฟ้มที่มีลักษณะเปิดบิตอนุญาตทั้งหมด ซึ่งสามารถเรียกโดยผู้ใช้ที่ไม่ใช่เจ้าของแฟ้มซึ่งไม่ปลอดภัย โดยตรวจสอบในไดเรกทอรีบ้านของผู้ใช้งานที่รันโปรแกรม โดยใช้ชิส

เต็มคอลล์ (system call) ชื่อ lstat ซึ่งเป็นฟังก์ชันที่แสดงค่าข้อมูลต่างๆ ของแฟ้มในไดเรกทอรีที่ระบุ ได้แก่ หมายเลขของไอโหนด (i-node number) โหมดของบิตอนุญาต (permission mode) จำนวนฮาร์ดลิงค์ (hardlink) หมายเลขประจำตัวผู้ใช้ (user id) กลุ่มของเจ้าของแฟ้ม (group id) และประเภทของแฟ้ม (Device type) ซึ่งเมื่อพบว่าโหมดของแฟ้มใดที่จะตรวจสอบมีการเปิดสิทธิของบิตอนุญาตทั้งหมดจะทำการรายงานผล

ตรวจสอบว่าผู้ใช้คนใดมีแฟ้ม .rhosts

หลักการคือ ผู้ใช้ที่อยู่บนเครื่องที่มีกำหนดไว้ในแฟ้ม .rhosts สามารถเข้าไปเป็นผู้ใช้เจ้าของแฟ้ม .rhosts ได้โดยไม่ต้องใส่รหัสผ่าน ซึ่งอาจไม่ใช่คนๆ เดียวกันจึงไม่ควรมีแฟ้มนี้ในไดเรกทอรีของผู้ใช้ ในโหมดผู้ใช้งานจะทำการค้นหาแฟ้ม .rhosts โดยใช้ตัวตรวจสอบ –e ในภาษาเพิร์ล ตรวจสอบในไดเรกทอรีบ้านของผู้ใช้งานซึ่งล็อกอินอยู่ ณ ขณะนั้น

ตรวจสอบผู้ใช้คนใดมีการกำหนดการค้นหาคำสั่ง (Search path) ที่เป็นอันตราย

หลักการคือ โดยปกติการกำหนดการค้นหาที่ถูกต้องควรเป็นดังนี้

PATH=/bin:/usr/:bin/etc/.

แต่ถ้าผู้ใช้กำหนดการค้นหาคำสั่งเป็นดังนี้

PATH=.:/bin:/etc:/usr/bin

เป็นการค้นหาคำสั่งโดยเริ่มจากไดเรกทอรีที่อยู่ปัจจุบัน แล้วจึงตามด้วยไดเรกทอรีที่กำหนดไว้ซึ่งอาจเรียกแฟ้มซึ่งมีชื่อเหมือนกับคำสั่งพื้นฐาน (System binary file) ซึ่งอาจเป็นม้าโทรจัน

ตรวจสอบช่องทางการสื่อสารของม้าโทรจันในระบบ

หลักการคือ การค้นหาจะอิงจากเอกสารดังแสดงรายละเอียดในบทที่ 2 หัวข้อม้าโทรจัน ซึ่งจะแสดง พอร์ต ของม้าโทรจันที่นิยมใช้ในการเข้าถึงระบบจากเว็บไซต์ www.anti-trojan.com ว่ามีการเปิดใช้งานอยู่หรือไม่ โดยทำการติดต่อไปยังแต่ละพอร์ต โดยทำการเปรียบเทียบกับตัวแปรซึ่งเก็บหมายเลขพอร์ตที่ม้าโทรจันนิยมใช้ หากพบว่ามีการเปิดพอร์ตตามที่ระบุในตัวแปร ระบบจะทำการแสดงผลการตรวจสอบ พร้อมทั้งแสดงชื่อม้าโทรจันที่คาดว่าจะใช้พอร์ตดังกล่าว ในการติดต่อ

4.2.2 หมวดโปรแกรมตรวจสอบความปลอดภัยสำหรับผู้ดูแลระบบ

ในหมวดนี้ ระบบจะแสดงฟังก์ชันการตรวจสอบทั้งหมด ทั้งในส่วนของการตรวจสอบความปลอดภัยสำหรับผู้ใช้ ซึ่งจะแสดงรายละเอียดในหมวดโปรแกรมตรวจสอบความ

ปลอดภัยสำหรับผู้ใช้ และส่วนของการตรวจสอบความปลอดภัยสำหรับผู้ดูแลระบบ ดังแสดงในรายละเอียดการตรวจสอบดังต่อไปนี้

ตรวจสอบแฟ้มที่มีลักษณะเปิดบิตอนุญาตทั้งหมด (World Writeable File) ในระบบ

หลักการคือ ตรวจสอบแฟ้มที่มีลักษณะเปิดบิตอนุญาตทั้งหมดในระบบ โดยตรวจสอบจากไดเรกทอรีที่จะค้นหา โดยจะมองจากแฟ้มที่เก็บค่าไดเรกทอรีของระบบไว้ หรือจากการระบุไดเรกทอรีโดยตรงจากผู้ใช้งาน โดยจะใช้ซิสเต็มคอลล์ ชื่อ lstat ซึ่งเป็นฟังก์ชันที่แสดงค่าข้อมูลต่างๆ ของแฟ้มในไดเรกทอรีที่ระบุ ได้แก่ หมายเลขของไอโหนด (i-node number) โหมดของบิตอนุญาต (permission mode) จำนวนฮาร์ดลิงค์ (hardlink) หมายเลขประจำตัวผู้ใช้ (user id) กลุ่มของเจ้าของแฟ้ม (group id) และประเภทของแฟ้ม (Device type) ซึ่งเมื่อพบว่าโหมดบิตอนุญาตของแฟ้มใดที่จะตรวจสอบมีการเปิดสิทธิของ บิตอนุญาตทั้งหมด ก็จะทำการรายงานผล

ตรวจสอบแฟ้ม .rhosts ในระบบ

หลักการคือ ในโหมดผู้ดูแลระบบจะทำการค้นหาแฟ้ม .rhosts โดยตรวจสอบไดเรกทอรีที่จะค้นหาจากแฟ้มที่เก็บไดเรกทอรีของระบบไว้ซึ่งก็คือแฟ้ม find.dir ซึ่งเก็บเส้นทางของไดเรกทอรีมาตรฐานในระบบลีนุกซ์ได้แก่ /boot, /dev, /etc, /home, /lib, /opt, /root, /sbin, /share, /src, /tmp, /usr, /var โดยใช้ตัวตรวจสอบ –e ในภาษาเพิร์ลในการตรวจสอบว่ามีแฟ้ม .rhosts ในไดเรกทอรีดังกล่าวหรือไม่พร้อมแสดงผลการตรวจสอบ

ตรวจสอบว่าผู้ดูแลระบบได้แก่ รูท มีการกำหนดการค้นหาคำสั่งที่เป็นอันตรายหรือไม่

หลักการคือ โดยปกติในลีนุกซ์การกำหนดการค้นหาที่ถูกต้องในลีนุกซ์ควรเป็นดังนี้

PATH=/bin:/usr/:bin/etc/.

แต่ถ้าผู้ใช้กำหนดการค้นหาคำสั่งเป็นดังนี้

PATH=.:/bin:/etc:/usr/bin

เป็นการค้นหาคำสั่งโดยเริ่มจากไดเรกทอรีปัจจุบัน แล้วจึงตามด้วยไดเรกทอรีที่กำหนดไว้ซึ่งในลักษณะนี้ รูท อาจเรียกคำสั่งพื้นฐานที่ไม่ได้อยู่ในไดเรกทอรีคำสั่งของระบบซึ่งอาจเป็นม้าโทรจัน

ตรวจสอบ ช่องทางการสื่อสารของม้าโทรจันในระบบ

หลักการคือ โดยการค้นหาจะอิงจากเอกสารดังแสดงรายละเอียดในบทที่ 2 หัวข้อม้าโทรจัน ซึ่งจะแสดง พอร์ตของม้าโทรจัน ที่นิยมใช้ในการเข้าถึงระบบจากเว็บไซต์ www.anti-trojan.com ว่ามีการเปิดใช้งานอยู่หรือไม่ โดยทำการติดต่อไปยังแต่ละพอร์ต โดยทำการเปรียบเทียบกับตัวแปรซึ่ง

เก็บหมายเลขพอร์ตที่ม้าโทรจันนิยมใช้ หากพบว่ามีการเปิดพอร์ตตามที่ระบุในตัวแปร ระบบจะทำการแสดงผลการตรวจสอบพร้อมทั้งแสดงชื่อม้าโทรจันที่คาดว่าจะใช้ พอร์ตดังกล่าวในการติดต่อ

ตรวจสอบผู้ใช้คนใดที่ไม่มีรหัสผ่าน ซึ่งถ้าพบจะทำการยกเลิกการใช้งานของผู้ใช้ คนดังกล่าว

หลักการคือ ผู้ใช้ทุกคนควรมีรหัสผ่าน ทั้งนี้เพื่อไม่ให้บุคคลภายนอกสามารถเข้ามาแอบใช้ได้ แฟ้ม /etc/passwd เป็นแฟ้มที่เก็บรายชื่อผู้ใช้ทุกคนในระบบ โดยในระบบลีนุกซ์จะเก็บรหัสผ่านที่ผ่านการเข้ารหัสแล้วในแฟ้ม /etc/shadow ซึ่งแฟ้มนี้ไม่ควรให้ผู้ใช้ทั่วไปสามารถดูได้ โปรแกรมจะทำการตรวจสอบผู้ใช้ที่ไม่มีรหัสผ่านในแฟ้มนี้ ในคอลัมน์ (column) ของรหัสผ่านหากไม่พบค่าการเข้ารหัสผ่านจะแสดงหน้าต่างข้อความ เพื่อยืนยันการยกเลิกการใช้งานของผู้ใช้ โดยใช้คำสั่งบนระบบลีนุกซ์คือ คำสั่ง “usermod” (modify user account) โดยใช้ parameter “-L” คำสั่งนี้จะทำเครื่องหมาย “!” หน้าส่วนคอลัมน์การเข้ารหัสของรหัสผ่าน (encrypted password column) เพื่อไม่ให้ผู้ใช้งานนั้น เข้าสู่ระบบได้

ตรวจสอบในแฟ้ม /etc/passwd ว่ามีบัญชีผู้ใช้โดยปริยาย (default user account) อื่น ๆที่ไม่จำเป็น เช่น “gopher” “news” “bin” “www” หรือไม่

หลักการคือ โปรแกรมจะตรวจสอบ แฟ้ม /etc/shadow โดยมีเงื่อนไขว่าเมื่อพบบัญชีผู้ใช้งานที่ไม่จำเป็น และไม่มีเครื่องหมาย “!” ในคอลัมน์การเข้ารหัสผ่านของผู้ใช้งาน ซึ่งเป็นสัญลักษณ์บ่งถึงการยกเลิกการใช้งานบัญชีผู้ใช้งานในแฟ้ม /etc/shadow โดยรายชื่อผู้ใช้งานที่ไม่จำเป็นจะเก็บอยู่ในตัวแปรเพื่อใช้ในการเปรียบเทียบขณะตรวจสอบ หากพบก็จะแสดงหน้าต่างข้อความแสดงการยืนยันเพื่อทำการยกเลิกบัญชีผู้ใช้งานที่ไม่จำเป็นนั้น โดยใช้คำสั่งบนระบบลีนุกซ์คือ คำสั่ง “usermod” (modify user account) โดยใช้พารามิเตอร์ “-L” คำสั่งนี้จะทำเครื่องหมาย “!” หน้าส่วนคอลัมน์การเข้ารหัสผ่าน (encrypted password column) เพื่อไม่ให้ผู้ใช้งานนั้นเข้าสู่ระบบได้

ตรวจสอบว่าระบบมีการใช้ชาโดว์พาสเวิร์ด (Shadow password) หรือไม่

หลักการคือ โปรแกรมตรวจสอบความปลอดภัยจะทำการตรวจสอบแฟ้ม /etc/passwd โดยจะทำการตรวจสอบในคอลัมน์รหัสผ่านว่ามีรูปแบบ (pattern) เท่ากับ “*:x:” อยู่หรือไม่ หากมีแสดงว่าระบบมีการทำชาโดว์พาสเวิร์ดอยู่แล้ว ระบบจะแสดงผลการตรวจสอบหากทำการตรวจสอบแล้วไม่พบระบบจะแสดงผลการตรวจสอบ พร้อมทั้งแนะนำให้เรียกใช้คำสั่ง “pwconv” ซึ่งเป็นคำสั่งในลีนุกซ์เรดแฮต ที่ใช้ในการทำชาโดว์พาสเวิร์ด นอกจากนี้ระบบจะทำการตรวจสอบแฟ้ม /etc/group โดยใช้หลักการเดียวกัน หากไม่มีการทำกรุ๊ปชาโดว์พาสเวิร์ด (group shadow

password) ระบบจะแนะนำให้เรียกใช้คำสั่ง “grpconv” ซึ่งเป็นคำสั่งในลีนุกซ์เรดแฮตที่ใช้ในการทำกรุ๊ปชาโดว์พาสเวิร์ด

ตรวจสอบว่าผู้ใช้ทั่วไปสามารถที่จะเขียนลงในแฟ้มและไดเรกทอรีระบบได้หรือไม่ (ตรวจสอบบิตอนุญาตของแฟ้ม และไดเรกทอรี)

หลักการคือ ไดเรกทอรีระบบที่จะตรวจสอบ เช่น /bin /boot /etc /usr /home /var /sbin ซึ่งโดยทั่วไป บิตอนุญาตของไดเรกทอรีระบบ ควรจะเป็นค่า 755 หรือ 750 ยกเว้นบางไดเรกทอรีเท่านั้น เช่น ไดเรกทอรี /tmp ซึ่งผู้ใช้ทั่วไปสามารถที่จะเก็บแฟ้มไดเรกทอรีเหล่านี้ได้ หรือแฟ้มบางแฟ้มในไดเรกทอรี /etc ควรจะมีบิตอนุญาตมีค่าเป็น 600 คือเฉพาะ ผู้ดูแลระบบเท่านั้นที่จะมีสิทธิแก้ไขแฟ้ม โปรแกรมตรวจสอบความปลอดภัยจะมีแฟ้มซึ่งเก็บรายละเอียดต้นแบบของบิตอนุญาตของแฟ้ม ไดเรกทอรี รายละเอียดของเจ้าของแฟ้ม และกลุ่มของเจ้าของแฟ้มที่เหมาะสม โดยอิงจากเอกสารตรวจสอบความปลอดภัยบนยูนิกส์เวอร์ชันสองในแฟ้มที่ชื่อว่า perms.dir โดยแบ่งแต่ละส่วนแยกออกเป็นดังนี้คือ แฟ้มหรือไดเรกทอรี บิตอนุญาต ชื่อเจ้าของแฟ้ม และกลุ่มเจ้าของแฟ้ม เริ่มแรกระบบจะทำการตรวจสอบว่ามี แฟ้มหรือไดเรกทอรี ที่จะตรวจสอบหรือไม่โดยเทียบกับข้อมูลในแฟ้ม perms.dir ถ้าไม่พบชื่อแฟ้มหรือไดเรกทอรีที่ระบุในแฟ้ม perms.dir ระบบจะทำการข้ามการตรวจ สอบไปยังแฟ้มหรือไดเรกทอรีอื่นในบรรทัดถัดไป แต่ถ้าหากพบแฟ้มที่ระบุจะทำการเปรียบเทียบบิตอนุญาตของแฟ้มหรือไดเรกทอรีนั้น กับข้อมูลในแฟ้ม perms.dir โดยใช้ ซิสเต็มคอลล์คือ “stat” ซึ่งจะแสดงค่าข้อมูล (information) ต่างๆ เกี่ยวกับแฟ้มส่วนหนึ่งที่แสดงคือข้อมูลของค่าบิตอนุญาตของแฟ้มหากข้อมูลบิตอนุญาตตรงกันก็จะแสดงผลรายงานการตรวจสอบ แต่หากพบว่าข้อมูลไม่ตรง โปรแกรมจะแสดงผลการตรวจสอบพร้อมทั้งแนะนำ ค่าบิตอนุญาตที่เหมาะสม อันดับต่อไประบบจะทำการตรวจสอบ ค่าความเป็นเจ้าของของแฟ้มหรือไดเรกทอรี (หมายเลขประจำตัวผู้ใช้) โดยใช้คำสั่งบนลีนุกซ์ “getpwnam” เพื่อแสดงค่าความเป็นเจ้าของ ของแฟ้มและไดเรกทอรีที่ระบุเพื่อเปรียบเทียบกับข้อมูลที่อยู่ในแฟ้ม perms.dir และแสดงผลการตรวจสอบ ขั้นตอนสุดท้ายระบบจะทำการตรวจสอบค่าความเป็นกลุ่มของแฟ้มหรือไดเรกทอรี (หมายเลขประจำกลุ่ม)โดยใช้คำสั่งบนลีนุกซ์ “getgrnam” เพื่อแสดงค่าความเป็นกลุ่มของของแฟ้มและไดเรกทอรีที่ระบุเพื่อเปรียบเทียบกับข้อมูลที่อยู่ในแฟ้ม perms.dir และแสดงผลการตรวจสอบ

ตรวจสอบบิตอนุญาตของแฟ้มและไดเรกทอรีในการทำเอฟทีพีแบบนิรนาม

หลักการคือ โปรแกรมตรวจสอบความปลอดภัยจะทำการตรวจสอบสถานะของ ทีซีพี พอร์ต 21

ว่ามีการเปิดใช้งานอยู่หรือไม่หากไม่มี ระบบจะทำการรายงานว่าบริการ เอฟทีพี ไม่ได้เปิดการให้บริการ หากพบว่า ทีซีพี พอร์ต 21 เปิดอยู่โปรแกรมจะทำการตรวจสอบบิตอนุญาตของแฟ้มและไดเรกทอรีในการทำเอฟทีพีแบบนิรนาม และทำการแสดงผลการตรวจสอบ

ตรวจสอบบิตอนุญาตของไดเรกทอรีแฟ้มอุปกรณ์ (device file) **และตรวจสอบแฟ้มที่ไม่ใช่ชนิดแฟ้มอุปกรณ์** (device file) **ว่าปรากฏอยู่ในไดเรกทอรี** /dev **หรือไม่**

หลักการคือ โดยทั่วไปแฟ้มอุปกรณ์ควรจะอยู่ในไดเรกทอรี /dev ซึ่งไม่ควรมีแฟ้มอื่นที่ไม่ใช่แฟ้มชนิดอุปกรณ์บรรจุอยู่ภายใต้ไดเรกทอรีนี้ โดยโปรแกรมจะตรวจสอบแฟ้มที่ไม่ใช่แฟ้มอุปกรณ์และแสดงผลการตรวจสอบ ปกติแฟ้มอุปกรณ์จะมีชนิด (type) บอกใน ตำแหน่งแรกของบิตอนุญาต ได้แก่ type c จะแสดงถึงแฟ้มอุปกรณ์แบบ character และ type b แสดงถึงแฟ้มอุปกรณ์แบบ block type p แสดงถึงแฟ้มแบบแน็บไพป์ (name pipe) type d แสดงถึงไดเรกทอรี และ type s แสดงถึงแฟ้ม socket โดยในการตรวจสอบโปรแกรมจะใช้คำสั่งบนลีนุกซ์ในการตรวจสอบชนิดของแฟ้มที่อยู่นอกเหนือชนิดที่ได้กล่าวมา โดยมีรูปแบบของคำสั่งคือ “/usr/bin/find /dev –not –xtype b -not -xtype c –not –xtype s –not –xtype p –not –xtype d” หากพบแฟ้มที่มีชนิดนอกเหนือ ระบบจะแสดงชื่อแฟ้มดังกล่าว

ตรวจสอบแฟ้ม lilo.conf **โดยจะตรวจสอบค่าในแฟ้มว่ามีการกำหนดพารามิเตอร์รหัสผ่าน** (password parameter) **หรือไม่ รวมทั้งตรวจสอบค่าบิตอนุญาตที่เหมาะสม**

หลักการคือ โปรแกรมจะทำการตรวจสอบแฟ้ม lilo.conf ว่ามีพารามิเตอร์รหัสผ่านอยู่หรือไม่โดยการตรวจสอบในเนื้อหาของแฟ้มและตรวจสอบบิตอนุญาตที่เหมาะสมพร้อมทั้งแสดงผลการตรวจสอบ

ตรวจสอบเพื่อให้แน่ใจว่าเคอร์เนลถูกครอบครองโดยกรุ๊ปรูท (root group) **และ บิตอนุญาตมีค่าเป็น** 644

หลักการคือ ปกติในระบบปฏิบัติการลีนุกซ์ เคอร์เนลจะถูกเก็บอยู่ภายใต้ไดเรกทอรี /boot ซึ่งชื่อแฟ้มจะเท่ากับ vmlinux ตามด้วยเวอร์ชันของเคอร์เนล โดยโปรแกรมตรวจสอบจะทำการเปรียบค่าบิตอนุญาต และชื่อเจ้าของแฟ้มกับแฟ้ม perms.dir ซึ่งเก็บรายละเอียดต้นแบบของบิตอนุญาตที่เหมาะสมของแฟ้ม และรายละเอียดของเจ้าของแฟ้มรวมทั้งกลุ่มของเจ้าของแฟ้มที่เหมาะสม หากข้อมูลตรงกัน ก็จะแสดงผลรายงานการตรวจสอบแต่หากข้อมูลไม่ตรงระบบจะแสดงผลการตรวจสอบพร้อมทั้งแนะนำ ค่าบิตอนุญาตและชื่อเจ้าของแฟ้มที่เหมาะสม

ตรวจสอบแฟ้มที่มี SUID และ SGID ในระบบ

หลักการคือ เนื่องจากแฟ้มที่มี SUID และ SGID สามารถทำให้ผู้ใช้คนอื่นที่เรียกแฟ้มเหล่านี้ มีสิทธิเท่าเจ้าของแฟ้ม โดยยิ่งเสี่ยงต่อความปลอดภัยถ้าแฟ้มนั้นมีค่า SUID เป็นรูท ทำให้ผู้ใช้ที่เรียกแฟ้มนั้นสามารถมีสิทธิเทียบเท่าเป็นรูทได้ โปรแกรมจะทำการค้นหาแฟ้ม SUID และ SGID ทุกๆ ไดเรกทอรีในระบบตามค่าที่เก็บไว้ในแฟ้ม find.dir ซึ่งเป็นแฟ้มที่เก็บข้อมูลไดเรกทอรีต่างๆ ของระบบ ปฏิบัติการลีนุกซ์เรดแฮตไว้ และผู้ใช้งานสามารถที่จะระบุไดเรกทอรีที่จะตรวจสอบแฟ้ม SUID และ SGID ได้ตามต้องการ

ตรวจสอบแฟ้มด้านเครือข่ายในระบบเป็นการตรวจสอบทีซีพีไอพีพื้นฐาน

ทำการตรวจสอบบิตอนุญาตของแฟ้มที่เหมาะสม และโครงสร้างภายในแฟ้มที่เกี่ยวข้องกับโปรแกรมทีซีพีไอพีพื้นฐาน ได้แก่ แฟ้ม /etc/hosts แฟ้ม /etc/hosts.equiv แฟ้ม /etc/ftpusers แฟ้ม /etc/service แฟ้ม /etc/inetd.conf และแฟ้ม /etc/securetty ดังต่อไปนี้

แฟ้ม /etc/hosts จะเก็บชื่อเครื่องและเลขที่อยู่ไอพี ที่อยู่บนเครือข่าย แฟ้ม /etc/hosts.equiv เก็บชื่อเครื่องซึ่งผู้ใช้ที่มีชื่อเหมือนกันสามารถเข้าไปใช้ได้โดยไม่ต้องใส่รหัสผ่าน ซึ่งเครื่องที่เก็บในแฟ้ม /etc/hosts.equiv มีคำว่า “+” หมายถึงยอมให้ทุกเครื่องเป็น ทรัซโฮส (trust host) ซึ่งเป็นสิ่งที่ไม่เหมาะสม จึงต้องมีการตรวจสอบทรัซโฮสในแฟ้มนี้

หลักการคือ โปรแกรมตรวจสอบจะทำการตรวจสอบเนื้อหาของแฟ้ม /etc/hosts.equiv ถ้าพบเครื่องหมาย “+” ก็จะทำการรายงานผล

แฟ้ม /etc/ftpusers เก็บชื่อผู้ใช้ที่ไม่สามารถใช้คำสั่งเอฟทีพี ในกรณีที่ระบบมีแฟ้ม /etc/ftpusers ควรจะมีชื่อ รูท เป็นเจ้าของแฟ้มนี้ เพื่อป้องกันผู้อื่นแก้ไขค่าโดยเพิ่มชื่อบัญชี รูท ในการ เอฟทีพี

หลักการคือ โปรแกรมตรวจสอบจะทำการตรวจสอบบิตอนุญาตของแฟ้ม /etc/ftpusers โดยใช้หลักการเดียวกันกับฟังก์ชันการตรวจสอบบิตอนุญาตของแฟ้มในระบบ คือเทียบค่าบิตอนุญาตกับแฟ้มต้นแบบ ซึ่งเก็บค่าบิตอนุญาตและชื่อเจ้าของแฟ้มที่เหมาะสม หากข้อมูลตรงกัน ก็จะแสดงผลรายงานการตรวจสอบ แต่หากข้อมูลไม่ตรงระบบจะแสดงผลการตรวจสอบพร้อมทั้งแนะนำ ค่าบิตอนุญาตและ ชื่อเจ้าของแฟ้มที่ควรจะเป็น

แฟ้ม /etc/services เป็นแฟ้มที่เก็บชื่อบริการ (service) ของโปรแกรมทีซีพีไอพีที่อยู่บนเครื่องนั้น ควรแน่ใจว่า เจ้าของแฟ้มนี้เป็นรูท

หลักการคือ โปรแกรมตรวจสอบความปลอดภัยจะทำการตรวจสอบบิตอนุญาตของแฟ้ม /etc/services โดยใช้หลักการเดียวกันกับการตรวจสอบบิตอนุญาตของแฟ้มในระบบ

แฟ้ม /etc/inetd.conf เป็นแฟ้มที่เก็บชื่อบริการ ซึ่งโปรแกรมไอเน็ตดีมอน (inetd) จะอ่านจากแฟ้มนี้ซึ่งหากโครงสร้างภายในแฟ้มทั้งสองนี้ไม่ถูกต้อง จะมีผลทำให้โปรแกรมทีซีพีไอพีทำงานผิดผลาดได้ ควรทำเครื่องหมาย "#" ไว้ที่ต้นบรรทัด เพื่อยกเลิกบริการที่เสี่ยงต่อความปลอดภัย เช่น บริการ echo และบริการ changen ก็อาจถูกใช้ในการโจมตีด้วย DoS (Deny off Services) ได้ หรือควรหลีกเลี่ยงคำสั่ง " r " เช่น rsh, rlogin และทีเอฟทีพีซึ่งอาจเป็นแหล่งกำเนิดของความไม่ปลอดภัย เป็นต้น

หลักการคือ โปรแกรมตรวจสอบความมั่นคงจะทำการตรวจสอบเนื้อหาของแฟ้ม /etc/ined.conf ถ้าพบบรรทัดที่ไม่มีเครื่องหมาย “#” ก็จะทำการรายงานผลการให้บริการของระบบไอเน็ตดีมอนเพื่อพิจารณา

แฟ้ม /etc/securetty เป็นแฟ้มที่เก็บข้อมูลเพื่อให้ผู้ใช้งานที่ใช้ชื่อบัญชีรูทสามารถล็อกอินผ่านทางทีทีวาย ดีไวซ์ (tty devices) หรือผ่านทางเครือข่ายจากระยะไกลที่ไม่ใช่คอนโซลเทอร์มินอล (console terminal) ซึ่งอาจเกิดความไม่ปลอดภัยหากมีผู้ไม่หวังดีทำการดักจับข้อมูลในเครือข่ายเพื่อทราบรหัสผ่านของผู้ดูแลระบบ ซึ่งทำการล็อกอินจากเครือข่ายระยะไกลได้

หลักการคือ โปรแกรมตรวจสอบความปลอดภัยจะทำการตรวจสอบแฟ้ม /etc/securetty ว่ามีการกำหนดค่าเท่ากับ “pts/x” หรือไม่ โดยค่า “x” คือลำดับของเครื่องที่สามารถจะล็อกอินจากเครือข่ายระยะไกลได้ เนื่องจากการกำหนดค่านี้ในแฟ้มจะเป็นอนุญาตให้บัญชีผู้ใช้งานรูท สามารถล็อกอินจากเครือข่ายระยะไกลที่ไม่ใช่ เครื่องคอนโซล

ตรวจสอบการโอนแฟ้มโดยใช้คำสั่งทีเอฟทีพี tftp (Trivial FTP) ได้หรือไม่

หลักการคือ ทีเอฟทีพีเป็นโปรแกรมที่ใช้ในการโอนแฟ้มระหว่างเครื่อง ซึ่งมีความเสี่ยงต่อความปลอดภัย โปรแกรมทีเอฟทีพี จึงไม่ควรเปิดบริการใช้งานหากไม่มีความจำเป็น โปรแกรมทีเอฟทีพีจะใช้โปรโตคอลยูดีพี (udp) พอร์ต 69 ในการติดต่อสื่อสาร คำสั่งในระบบลีนุกซ์ที่ใช้ในการตรวจสอบสถานะ การทำงานของโปรแกรมทีเอฟทีพี คือคำสั่ง “/bin/netstat –na | grep :69” ซึ่งหากโปรแกรมตรวจสอบความปลอดภัยพบว่า มีโปรโตคอลยูดีพี พอร์ต 69 เปิดอยู่แสดงว่าโปรแกรมทีเอฟทีพีมีการทำงานอยู่ในระบบ โปรแกรมตรวจสอบความปลอดภัยจะทำการแสดงผลการตรวจสอบ

ตรวจสอบแฟ้มและไดเรกทอรีการให้บริการด้านเว็บเซิร์ฟเวอร์ (Program Apache)

หลักการคือ ตรวจสอบความเป็นเจ้าของและสิทธิ์การใช้งานไดเรกทอรีและแฟ้มที่เกี่ยวข้องให้เหมาะสม ได้แก่ ไดเรกทอรีที่เก็บแฟ้มคอนฟิกของเว็บเซิร์ฟเวอร์ (conf) ไดเรกทอรีที่เก็บแฟ้มเอกสารทั้งหมดของเว็บไซต์ (Document Root) ไดเรกทอรีที่เก็บสคริปต์ซีจีไอ (CGI-Bin) ไดเรกทอรีที่เก็บแฟ้มล็อกของบริการเว็บ ไดเรกทอรีที่เก็บคำสั่งของโปรแกรมอาปาเช โปรแกรมตรวจสอบจะมีแฟ้มซึ่งเก็บรายละเอียดของบิตอนุญาตของแฟ้ม ไดเรกทอรีที่เกี่ยวข้องกับการบริการด้านเว็บและรายละเอียดของเจ้าของแฟ้มรวมทั้งกลุ่มของเจ้าของแฟ้มที่ถูกต้อง โดยอิงจากเอกสารตรวจสอบความปลอดภัยบนยูนิกซ์เวอร์ชันสอง ในแฟ้มที่ชื่อว่า webperms.dir โดยแบ่งแต่ละส่วนออกเป็นดังนี้คือ แฟ้มหรือ ไดเรอทอรี บิตอนุญาต ชื่อเจ้าของแฟ้ม กลุ่มเจ้าของแฟ้ม เริ่มแรกระบบจะทำการตรวจสอบว่ามี แฟ้มหรือไดเรกทอรี ที่จะตรวจสอบหรือไม่โดยมองข้อมูลในแฟ้ม webperms.dir ถ้าไม่ ระบบจะทำการข้ามการตรวจสอบไปยังแฟ้มถัดไปในแฟ้ม webperms.dir จากนั้นทำการเปรียบเทียบบิตอนุญาตของแฟ้มหรือไดเรกทอรีนั้นกับข้อมูลในแฟ้ม webperms.dir หากข้อมูลตรงกัน ก็จะแสดงผลรายงานการตรวจสอบ แต่หากข้อมูลไม่ตรงระบบจะแสดงผลการตรวจสอบพร้อมทั้งแนะนำค่าบิตอนุญาตที่ควรจะเป็น อันดับต่อไประบบจะทำการตรวจสอบ ค่าความเป็นเจ้าของของแฟ้มหรือไดเรกทอรี (หมายเลขประจำตัวผู้ใช้) โดยใช้คำสั่งบนลีนุกส์ “getpwnam” เพื่อแสดงค่าความเป็นเจ้าของ ของแฟ้มและไดเรกทอรีที่ระบุ เพื่อเปรียบเทียบกับข้อมูลที่อยู่ในแฟ้ม perms.dir และทำการแสดงผลการตรวจสอบ ขั้นตอนสุดท้ายระบบจะทำการตรวจสอบค่าความเป็นกลุ่มของแฟ้มหรือไดเรกทอรี (หมายเลขประจำกลุ่ม) โดยใช้คำสั่งบนลีนุกส์ “getgrnam” เพื่อแสดงค่าความเป็นกลุ่มของของแฟ้มและไดเรกทอรีที่ระบุเพื่อเปรียบเทียบกับข้อมูลที่อยู่ในแฟ้ม webperms.dir และทำการแสดงผลการตรวจสอบ

ตรวจสอบการทำรีเลย์ ผ่านเมล์เซิร์ฟเวอร์

ในระบบปฏิบัติการลีนุกซ์ โปรแกรมเซนเมล์ เป็นโปรแกรมที่ใช้ในการให้บริการเมล์เซิร์ฟเวอร์ ปัญหาหนึ่งของเมล์เซิร์ฟเวอร์ก็คือการถูกการทำ รีเลย์ ซึ่งการทำรีเลย์ คือการส่งอีเมล์จากภายนอกโดเมนผ่านเมล์เซิร์ฟเวอร์ไปที่โดเมนอื่น โดยที่ทั้งผู้ส่งและผู้รับไม่ใช่ผู้ที่อยู่ภายใต้โดเมนที่เมล์เซิร์ฟเวอร์ให้บริการนี้จะถือว่าเป็นการ รีเลย์ ในอดีต การรีเลย์ช่วยให้การส่งอีเมล์ทำได้สะดวกเป็นการส่งกันต่อไปเป็นทอดๆ เหมือนกับที่แพ็กเกจของโพรโตคอลทีซีพีไอพีที่วิ่งผ่านเส้นทางอื่นได้ แต่ปัจจุบันการเปิดให้เมล์เซิร์ฟเวอร์สามารถ รีเลย์ ได้ถือเป็นอันตรายอย่างยิ่ง เพราะสามารถนำไปใช้ในการทำเมล์สแปม (mail spamming) ทำการส่งอีเมล์จำนวนมหาศาลไปยังผู้รับจำนวนมาก ซึ่งเป็นการเปลืองทรัพยากรโดยใช่เหตุ

หลักการคือ โดยปกติการพิจารณาสิทธิ์ของเครื่องใดบ้างที่อนุญาตให้ รีเลย์ ผ่านไปได้จะแก้ไขแฟ้ม /etc/mail/access โปรแกรมตรวจสอบความปลอดภัยจะทำการตรวจสอบพอร์ต การให้บริการเมล์ คือ ทีซีพี พอร์ตหมายเลข 25 เมื่อตรวจพบจะทำการทดสอบการ รีเลย์ โดยการส่งแพคเก็ด (packet) รีเลย์ผ่าน พอร์ต นี้ ถ้าสามารถส่งได้นั่นคือ โปรแกรมเซนเมล์ ได้เปิดการรีเลย์ไว้ ระบบจะทำการรายงานผล

ตรวจสอบสถานะ การเปิดช่องทางสื่อสารในระบบ

หลักการคือ เป็นการตรวจสอบช่องทางสื่อสาร (tcp port) ที่ใช้ในระบบพร้อมรายงานผลการตรวจสอบ โดยอิงจากหมายเลขพอร์ตที่สถาบันวิทยาการสารสนเทศแห่ง University of Southem California จัดทำไว้ดังกล่าวในบทที่ 2 หัวข้อหมายเลขเซอร์วิสหรือแอพพลิเคชันพอร์ต

ตรวจสอบภาวะ การทำงานแบบไม่เลือกในอินเตอร์เฟส

หลักการคือ จะทำการสอบถามไปยังอินเตอร์เฟส ว่ากำลังทำงานในภาวะ การทำงานแบบไม่เลือก โดยใช้คำสั่ง “/sbin/ifconfig“ ถ้าพบว่าทำงานในภาวการณ์ทำงานแบบไม่เลือก โดยแสดงผลลัพธ์ของการเรียกคำสั่งมีค่าเท่ากับ “PROMISC” ในอินเตอร์เฟสใดๆ ก็ตาม โปรแกรมตรวจสอบความปลอดภัย จะทำการแสดงผลการตรวจสอบว่า อินเตอร์เฟสกำลังทำงานในภาวะ การทำงานแบบไม่เลือก หากไม่พบโปรแกรมตรวจสอบความปลอดภัยจะแสดงผลการตรวจสอบว่า อินเตอร์เฟสไม่ได้ทำงานในภาวะ การทำงานแบบไม่เลือก

ตรวจสอบคำหรือข้อความที่ล่อแหลมต่อความปลอดภัย ในแฟ้มล็อกของระบบ

ผู้ใช้งานทำการกำหนดข้อความที่ต้องการให้โปรแกรมตรวจสอบในล็อกของระบบ เมื่อโปรแกรมตรวจสอบพบข้อความที่ระบุ โปรแกรมจะทำการส่งเมล์แจ้งเตือนไปยัง รูท ตัวอย่างเช่น ต้องการตรวจสอบการใช้คำสั่งการเปลี่ยนรหัสผ่านในระบบ คือคำสั่ง “passwd” ดังนั้นผู้ใช้งานสามารถใช้คำว่า “passwd” มาใช้เป็น trigger ได้ โดยระบุคำที่ใช้เป็น trigger ป้อนให้กับระบบ โดยเมื่อโปรแกรมพบคำว่า “passwd” ในล็อกของระบบ โปรแกรมจะทำการแจ้งเมล์เตือนไปยังผู้ดูแลระบบ หลักการคือ โดยปกติในลีนุกซ์จะมีบริการซิสล็อก (syslog) ทำหน้าที่ในการเขียนล็อกที่เกิดขึ้นในระบบลงสู่แฟ้มล็อกชนิดต่างๆซึ่งแฟ้มที่เก็บล็อกจากบริการซิสล็อกจะระบุอยู่ในแฟ้ม /etc/syslog.conf โดยบริการซิสล็อกจะมาอ่านค่าในแฟ้มนี้และทำการเขียนล็อกของระบบลงสู่แฟ้มล็อกตามที่ระบุในแฟ้มดังกล่าว ในโปรแกรมตรวจสอบความปลอดภัยในส่วนการตรวจการณ์ล็อก (Monitor Log) จะอ่านค่าล็อกในระบบผ่านแฟ้มชนิดแนบไพป์ Named pipe (FIFO) ซึ่งเป็น

แฟ้มที่รับค่าล็อกจากบริการซิสล็อกก่อนส่งผ่านไปยังแฟ้มล็อกอีกทีหนึ่ง ซึ่งผู้ใช้งานต้องระบุค่าแฟ้มชนิดแนบไพป์ ในแฟ้ม /etc/syslog.conf ก่อน เมื่อบริการซิสล็อกทำงานจะทำการเขียนข้อมูลล็อกของระบบ ผ่านทางแฟ้มแนบไพป์ก่อนเขียนลงในแฟ้มล็อกของระบบ โปรแกรมตรวจสอบจะทำการกรองข้อความที่เขียนผ่านแฟ้มแนบไพป์ว่าตรงกับข้อความที่ต้องการตรวจสอบ โดยข้อความที่จะตรวจสอบ ผู้ใช้งานจะระบุไว้ในแฟ้มที่ชื่อว่า mlog.conf เมื่อข้อความตรวจสอบตรงกับคำหรือข้อความของล็อก ที่บริการซิสล็อกเขียนลงผ่านแฟ้มแนบไพป์ ก็จะทำการส่งเมล์แจ้งไปยังผู้ดูแลระบบ

4.3 การพัฒนาโปรแกรมย่อยแบบดีมอนด้วยภาษาเพิร์ล

เนื่องจากโปรแกรมตรวจสอบความปลอดภัยบนระบบปฏิบัติการลีนุกซ์ เป็นโปรแกรมที่จะทำงานเมื่อมีการเรียกโปรแกรมขึ้นมาใช้งานจึงจะสามารถตรวจสอบความปลอดภัยในหัวข้อต่างๆ ได้ แต่ในบางครั้งหากมีความจำเป็นที่จะต้องการมีตรวจสอบความปลอดภัยของระบบ เพื่อให้แน่ใจว่ามีการตรวจเช็คความปลอดภัยในระบบตลอดเวลา เพื่อป้องกันปัญหาที่จะเกิดขึ้นด้านความปลอดภัยกับระบบ ทั้งนี้เมื่อตรวจสอบพบปัญหาด้านความปลอดภัยก็ให้มีการแจ้งเตือนผู้ดูแลระบบเพื่อให้ผู้ดูแลระบบแก้ไขปัญหาได้ทันท่วงที และในการตรวจสอบความปลอดภัยของระบบ ในบางกรณีที่จำเป็นต้องมีการค้นหาแฟ้มทุกๆไดเรกทอรี ซึ่งค่อนข้างจะกินทรัพยากรของระบบเป็นอันมาก หากทำงานในลักษณะการทำงานเบื้องหน้า (Foreground Process) เพราะจะทำให้ไม่เหลือทรัพยากรสำหรับโปรแกรมอื่นๆ ในการทำงาน ฉะนั้นในบางกรณีจึงจำเป็นต้องให้โปรแกรมตรวจสอบความปลอดภัยทำงานในลักษณะของการทำงานเบื้องหลัง (Background process) หรือ การทำงานแบบดีมอน (daemon process) เพื่อให้คอยตรวจสอบปัญหาที่จะเกิดด้านความปลอดภัยตลอดเวลา และเหลือทรัพยากรสำหรับโปรแกรมอื่นๆให้สามารถทำงานได้

4.4 การพัฒนาส่วนแสดงผลภาษาไทยในระบบเอกซ์วินโดว์

ส่วนของการแสดงผลภาษาไทยผู้พัฒนาได้ติดตั้ง ไทยเอกซ์เทนชัน (ThaiExtension) [19] หรือเรียกย่อๆว่าทีอี (TE) เป็นแพกเกจโปรแกรมหรือสคริปต์ที่อำนวยความสะดวกในการติดตั้งฟอนต์ โปรแกรมที่สนับสนุนภาษาไทยเพิ่มเติม จากดิสทริบิวชันที่ไม่สนุนภาษาไทยอย่างเต็มที่ ดิสทริบิวชันที่ไม่สนับสนุนภาษาไทยเต็มที่นี้ หมายถึงดิสทริบิวชันภาษาอังกฤษเช่น เรดแฮต สแลกแวร์ ซูซี ฯลฯ ดิสทริบิวชันเหล่านี้ยังไม่สนับสนุนภาษาไทยอย่างเป็นทางการ บางดิสทริบิวชันไม่มีฟอนต์ภาษาไทย (หรือมีเพียงบางส่วน) ไม่มีการปรับแต่งคีย์บอร์ดภาษาให้ หรือขาดโปรแกรมใช้

งานที่สนับสนุนภาษาไทย ไทยเอกซ์เทนชันจะหมดหน้าที่ก็ต่อเมื่อโปรแกรมสากลต่างๆเริ่มสนับสนุน การใช้ภาษาไทยโดยปริยาย

โดยทั่วไปในระบบเอกซ์วินโดว์ ของระบบปฏิบัติการลีนุกซ์เรดแฮตจะมีการออกแบบการทำงานในระบบไคลเอนต์เซิร์ฟเวอร์โดยแยกการทำงานออกเป็นสองส่วนหลักๆ คือ เอกซ์ไคลเอนต์และเอกซ์เซิร์ฟเวอร์ โดยเอกซ์ไคลเอนต์ในการแสดงผลข้อความบนจอ เอกซ์เซิร์ฟเวอร์จะเป็นผู้จัดการเรื่องฟอนต์ต่างๆ โดยเอกซ์ไคลเอนต์สามารถขอดูรายชื่อฟอนต์ที่มีอยู่ทั้งหมด เลือกฟอนต์ที่ใช้และสั่งเอกซ์เซิร์ฟเวอร์ให้วาดตัวอักษรบนจอภาพด้วยฟอนต์ที่กำหนด นอกจากนี้ฟอนต์สำหรับเอกซ์เซิร์ฟเวอร์สามารถแยกออกมาเป็นบริการต่างหากที่เรียกว่า เอกซ์ฟอนต์เซิร์ฟเวอร์ (X font server) หรือบางครั้งเรียกสั้นๆว่าฟอนต์เซิร์ฟเวอร์ หรือเอกซ์เอฟเอส (XFS)

4.4.1 การใช้ฟอนต์ผ่านฟอนต์เซิร์ฟเวอร์ [18]

โดยปกติ เอกซ์เซิร์ฟเวอร์ในระบบปฏิบัติการลีนุกซ์เรดแฮต สามารถเรียกใช้ฟอนต์จากฟอนต์เซิร์ฟเวอร์ได้ โดยทำการระบุในฟอนต์พาธของเอกซ์เซิร์ฟเวอร์ด้วยค่าที่แทนบริการของเอกซ์เอฟเอส โดยมีรูปแบบคือ “Protocol/[host]:port” โดยค่าปริยายที่ใช้คือ tcp/:7100 หมายถึงใช้ Unix domain socket ที่พอร์ต 7100 ที่เครื่องเดียวกับเอกซ์เซิร์ฟเวอร์นั่นเอง โดยจะกำหนดที่แฟ้ม /etc/X11/fs/config นอกจากนี้ยังสามารถระบุพาธของฟอนต์ที่ต้องการติดตั้งเพิ่มเติมในแฟ้มดังกล่าว โดยบริการเอกซ์เอฟเอส จะมาอ่านค่าจากแฟ้มนี้

4.4.2 การติดตั้งฟอนต์บนเอกซ์วินโดว์

ดังที่ได้กล่าวไปแล้วว่าในระบบปฏิบัติการลีนุกซ์เรดแฮต ค่าปริยายเอกซ์เซิร์ฟเวอร์จะเรียกใช้ฟอนต์จากฟอนต์เซิร์ฟเวอร์ ดังนั้นในการติดตั้งฟอนต์ภาษาไทยบนเอกซ์วินโดร์จะติดตั้งที่ฟอนต์เซิร์ฟเวอร์ โดยฟอนต์ภาษาไทยที่ติดตั้งจะเป็นแพกเกจ Thaixfont โดยมีขั้นตอนคือทำการติดตั้งฟอนต์ภาษาไทยในไดเรกทอรีที่กำหนดและทำการสร้างแฟ้ม fonts.dir ด้วยคำสั่ง “ mkfontdir ” โดยคำสั่งนี้จะทำการสแกนฟอนต์ทั้งหมดในไดเรกทอรีที่ทำการติดตั้ง สร้างเป็นรายการต่างๆ ใน fonts.dir จากนั้นจึงทำการลงทะเบียนไดเรกทอรีที่ทำการติดตั้งฟอนต์ภาษาไทย เข้าไปยังฟอนต์เซิร์ฟเวอร์เพื่อให้ระบบรู้จักฟอนต์ที่เพิ่มเข้าไป

4.4.3 การเรียกใช้งานฟอนต์ภาษาไทยใน เพิร์ลทีเค

เพิร์ลทีเค แอพพลิเคชัน สามารถกำหนดฟอนต์ในแต่ละไอเท็ม (item) ภายในแอพพลิเคชันโดยมีรูปแบบดังนี้

" $Item(ตัวแปรของไอเท็ม) = (-font => 'ชื่อของฟอนต์',ขนาด','รูปแบบ') "

ในกรณีที่ต้องการกำหนดรูปแบบของฟอนต์ให้กับทุกไอเท็มใน เพิร์ลทีเค แอพพลิเคชัน สามารถกำหนดได้โดยมีรูปแบบคือ " perl application.pl –fn fontname " ซึ่งในกรณีหลัง ทุกๆ ไอเท็ม ในแอพพลิเคชันนั้น ไม่ว่าจะเป็นเท็ก (text) ลาเบล (label) บัททอน (button) เมนู (menu) จะมีรูปแบบของฟอนต์เหมือนกันทั้งหมดตามที่กำหนดไว้ในครั้งแรกตอนเรียกแอพพลิเคชัน ซึ่งค่อนข้างสะดวกและรวดเร็วในการพัฒนาโปรแกรม เพราะไม่จำเป็นต้องไปกำหนดรูปแบบฟอนต์ในแต่ละไอเท็ม ของแอพพลิเคชัน ในการแสดงรูปแบบของฟอนต์ภาษาไทยในเพิร์ลทีเคแอพพลิเคชันในระบบเอกซ์วินโดว์นั้น จะต้องตรวจสอบว่ามีแฟ้มรหัสอักษรของ tis-620-0 อยู่หรือไม่ โดยแฟ้มดังกล่าวจะใช้ชื่อว่า iso-8859-11.enc ปกติจะอยู่ในไดเรกทอรี /usr/X11R6/lib/X11/fonts/encodings ซึ่งถ้าปรากฏแฟ้มนี้ แสดงว่าในระบบเอกซ์วินโดว์มีการ สนับสนุนภาษาไทย โดยจะมีการติดตั้งฟอนต์ภาษาไทยไว้แล้ว โดยโปรแกรมตรวจสอบความปลอดภัยพื้นฐานสำหรับระบบปฏิบัติการลีนุกซ์เรดแฮต ในที่นี้จะใช้ฟอนต์ภาษาไทยในรูปแบบตัวอักษรชนิด บิตแมพ เนื่องจากเป็นฟอนต์ภาษาไทยพื้นฐานที่มักจะมีมากับระบบปฏิบัติการลีนุกซ์ตระกูลเรดแฮตหรือในเครื่องที่มีการติดตั้ง TE (Thai Exension) และฟอนต์ชนิดนี้ยังมีให้เลือกใช้หลายขนาดผิดกับฟอนต์ภาษาไทยชนิดอื่น ที่มักมีขนาดที่กำหนดมาให้เลย (fix size) โดยจะใช้ฟอนต์ที่มีชื่อว่า thai6x14.bdf ซึ่งยังมีขนาดอื่นๆ ให้เลือกได้แก่ "thai7x18" "thai8x13" "thai8x20" "thai9x20" ตามลำดับ ฉะนั้น ในการแสดงผลภาษาไทยของโปรแกรมตรวจสอบความปลอดภัยพื้นฐานสำหรับระบบปฏิบัติการลีนุกซ์เรดแฮต ในระบบ เอกซ์วินโดว์นอกจากการตรวจสอบว่ามีแฟ้มรหัสอักษรของ tis-620-0 แล้ว จึงต้องตรวจสอบว่ามีการติดตั้งฟอนต์เหล่านี้ได้สำเร็จหรือไม่ โดยใช้คำสั่ง ' xlsfonts | grep "thai6x14" ' ซึ่งจะปรากฏว่ามีฟอนต์ชนิดนี้ติดตั้งอยู่ในระบบเอกซ์วินโดว์เพื่อใช้ในการแสดงผล ในการเรียกโปรแกรมตรวจสอบความปลอดภัยพื้นฐานสำหรับระบบปฏิบัติการลีนุกซ์เรดแฮต เพื่อแสดงผลภาษาไทยจึงมีรูปแบบคือ "perl linucscth.pl –fn tha6x14" ซึ่งสามารถเปลี่ยนชนิดของฟอนต์ได้ตามความเหมาะสมเพื่อแสดงผล

บทที่ 5
รายงานผลการวิจัย

ในบทนี้จะกล่าวถึงผลการวิจัยโดย รายงานผลการทำงานของแต่ละฟังก์ชัน

5.1 สภาพแวดล้อมของการพัฒนาโปรแกรม

ในการพัฒนาได้ทำการพัฒนาโปรแกรมบนสภาพแวดล้อมของ ฮาร์ดแวร์ และ ซอฟท์แวร์ ดังนี้

- หน่วยประมวลผลกลาง ตระกูลอินเทลรุ่น เพนเทียม ทู 350 เมกะเฮิร์ทซ์
- หน่วยความจำ 128 เมกกะไบต์ บนระบบปฏิบัติการลีนุกซ์ตระกูลเรดแฮตเวอร์ชัน 8 ในสภาพแวดล้อมแบบเอกซ์วินโดว์
- เครื่องมือที่ใช้ในการพัฒนาโปรแกรมคือ เพิร์ล เวอร์ชั่น 5.6 และในส่วนติดต่อผู้ใช้งานใช้ เพิร์ลทีเค เวอร์ชั่น 800.025 ซึ่งเป็นทูลคิตของ เพิร์ล ในการพัฒนา

ต่อมาได้นำโปรแกรมไปติดตั้งยังเครื่องไอบีเอ็มอีซีรีย์ ซึ่งใช้ระบบปฏิบัติการลีนุกซ์ตระกูลเรดแฮตเวอร์ชั่น 9 เป็นเครื่องที่ให้บริการอินเตอร์เน็ตและบริการรีเลย์เมล์เซิร์ฟเวอร์ ที่บริษัทโปรลายน์ (ประเทศไทย) จำกัด

5.2 สภาพแวดล้อมของการทดสอบโปรแกรม

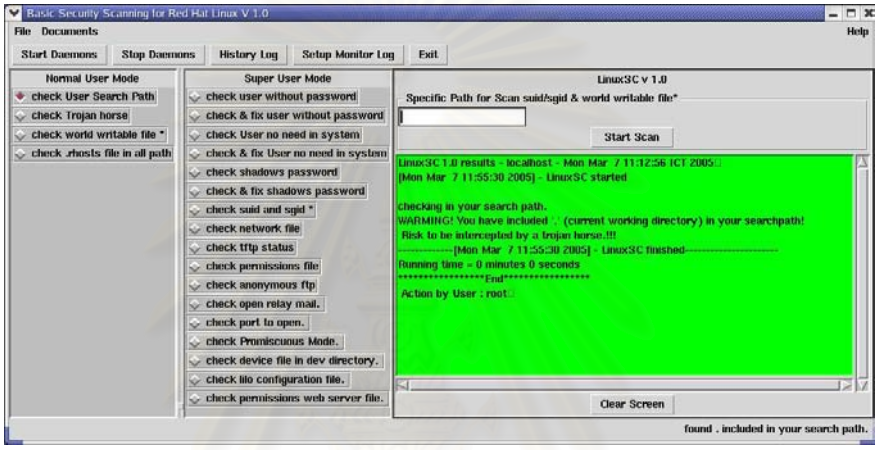
ทดสอบการทำงานบนเครื่องไอบีเอ็มอีซีรีย์ ซึ่งมีคุณสมบัติของเครื่องแสดงในตารางที่ 5.1 ดังนี้

ตารางที่ 5.1 คุณสมบัติเครื่องไอบีเอ็มอีซีรีย์ และเครื่องคอมแพคอีโว

คุณสมบัติของเครื่อง	IBM e-Series	Compaq Evo
ชนิดของ CPU	เพนเทียมทรี 1 GHz	เพนเทียมทู 250 GHz
จำนวน CPU	1	1
หน่วยความจำ	512	128
ขนาดเนื้อที่ทั้งหมด	40 GB	10 GB
ขนาดเนื้อที่ใช้งาน	6 GB	4 GB
จำนวนผู้ใช้งาน	100 User	10
ระบบปฏิบัติการ	ลีนุกซ์เวอร์ชั่น 9	ลีนุกซ์เวอร์ชั่น 8

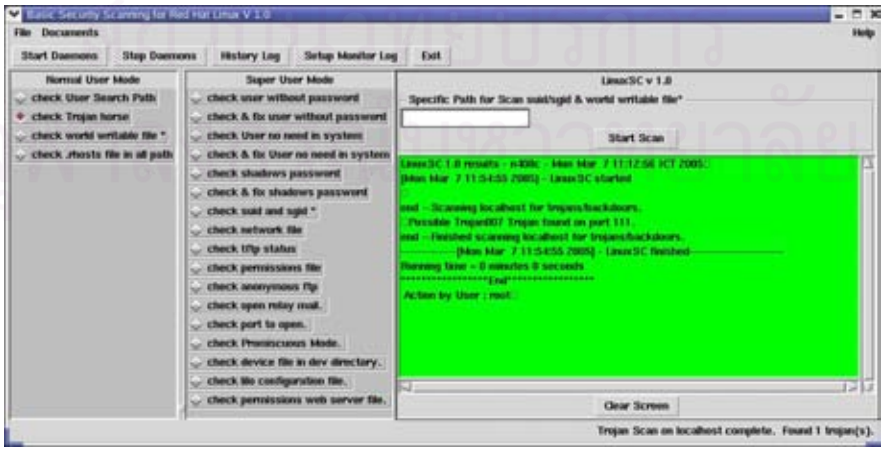
5.3 การทดสอบการทำงานของโปรแกรมในแต่ละฟังก์ชัน

5.3.1 การตรวจสอบเส้นทางค้นหาที่เป็นอันตราย โปรแกรมตรวจสอบจะทำการตรวจสอบแฟ้ม .profile ในไดเรกทอรีบ้านของผู้ใช้งานที่เรียกโปรแกรมตรวจสอบ ว่ามีการค้นหาคำสั่งจากไดเรกทอรีปัจจุบันก่อนหรือไม่ โดยในการทดสอบได้ทำการแก้ไขแฟ้ม .profile โดยการใส่เครื่องหมาย "." ไว้หน้าส่วนของการค้นหาคำสั่ง ซึ่งหมายถึงให้ค้นหาคำสั่งจากไดเรกทอรีปัจจุบันก่อนซึ่งไม่ปลอดภัย เพื่อทดสอบการทำงานของโปรแกรมดังแสดงผลในรูปที่ 5.1



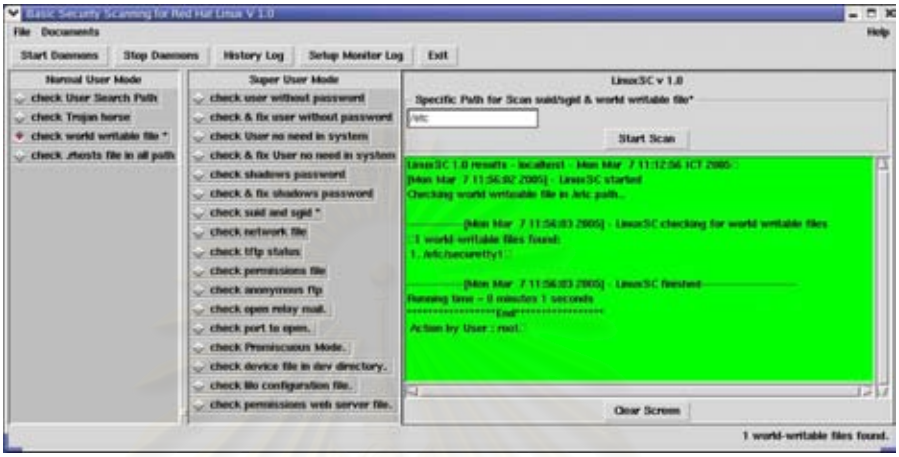
รูปที่ 5.1 ผลการทดสอบเส้นทางค้นหาที่เป็นอันตราย

5.3.2 การตรวจสอบช่องทางสื่อสารของม้าโทรจัน โปรแกรมตรวจสอบจะทำการตรวจสอบช่องทางการสื่อสารที่ม้าโทรจันมักนิยมใช้ในการทดสอบ ได้ทำการเปิดช่องทางสื่อสารโดยเลียนแบบการทำงานของม้าโทรจันโดยใช้โปรแกรม "netcat" [25] เพื่อทดสอบการทำงานของโปรแกรมดังแสดง ผลในรูปที่ 5.2



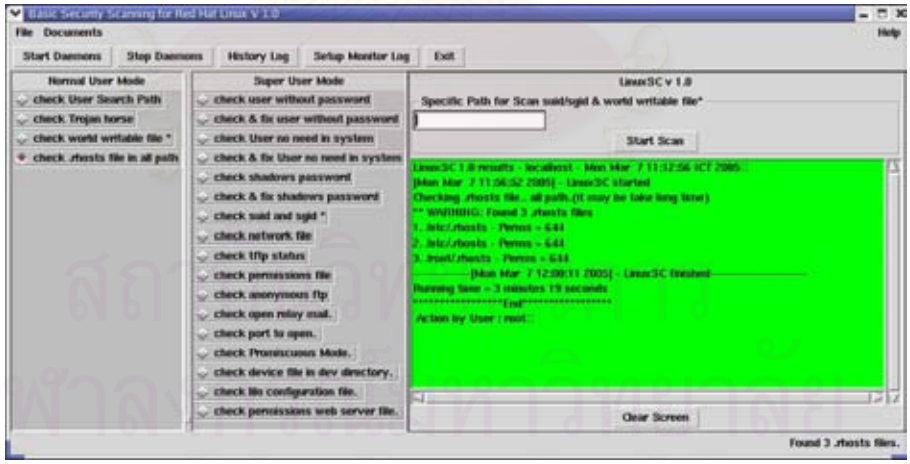
รูปที่ 5.2 การตรวจสอบช่องทางสื่อสารของม้าโทรจัน

5.3.3 การตรวจสอบแฟ้มที่เปิดสิทธิเต็ม โปรแกรมตรวจสอบจะทำการตรวจสอบแฟ้มที่มีการเปิดสิทธิเต็มพร้อมทั้งรายงานผลจำนวนแฟ้มที่พบ ในการทดสอบได้ทำการเปลี่ยนบิตอนุญาตของแฟ้มที่จะทดสอบให้มีสิทธิเต็ม เพื่อทดสอบการทำงานของโปรแกรมดังแสดงผลในรูปที่ 5.3



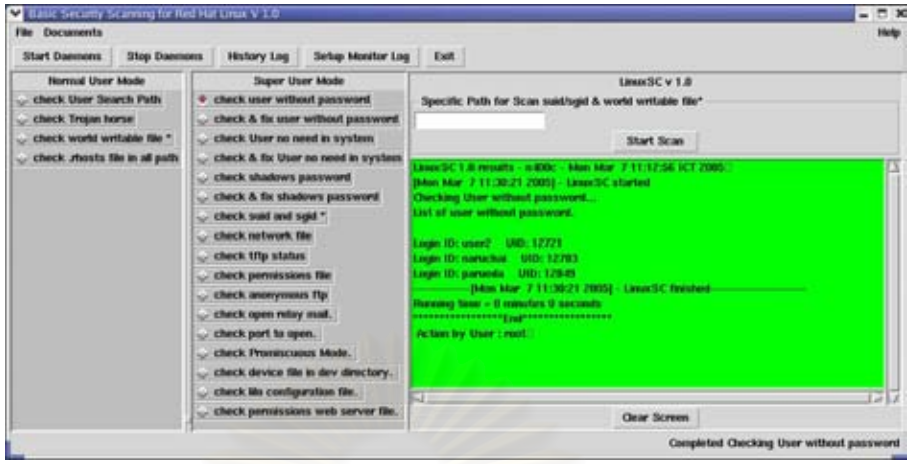
รูปที่ 5.3 การตรวจสอบแฟ้มที่เปิดสิทธิเต็ม

5.3.4 การตรวจสอบแฟ้ม .rhost โปรแกรมตรวจสอบจะทำการค้นหาแฟ้ม .rhost ในระบบ โดยในการทดสอบได้ทำการสร้างแฟ้ม .rhost ในไดเรกทอรีบ้านของผู้ใช้งานที่จะทดสอบและในไดเรกทอรี /etcเพื่อทดสอบการทำงานของโปรแกรมดังแสดงผลในรูปที่ 5.4



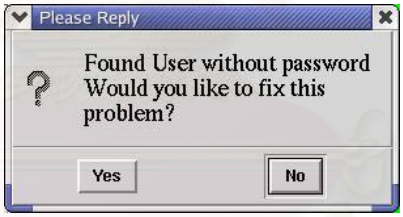
รูปที่ 5.4 การตรวจสอบแฟ้ม .rhost

5.3.5 ตรวจสอบผู้ใช้งานที่ไม่มีรหัสผ่าน โปรแกรมตรวจสอบจะทำการตรวจสอบผู้ใช้งานที่ไม่มีรหัสผ่านในระบบ โดยในการทดสอบได้ทำการลบรหัสผ่านของผู้ใช้งานที่ทดสอบ เพื่อทดสอบการทำงานของโปรแกรมดังแสดงผลในรูปที่ 5.5

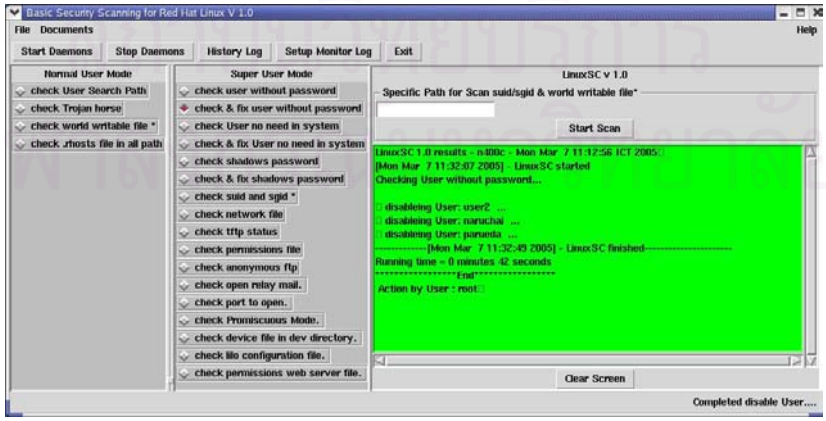


รูปที่ 5.5 การตรวจสอบผู้ใช้งานที่ไม่มีรหัสผ่าน

5.3.6 ตรวจสอบผู้ใช้งานที่ไม่มีรหัสผ่านและแก้ไข โปรแกรมตรวจสอบจะทำการตรวจสอบผู้ใช้งานที่ไม่มีรหัสผ่านในระบบ โดยในการทดสอบได้ทำการลบรหัสผ่านของผู้ใช้งานที่ทดสอบ เมื่อโปรแกรมตรวจสอบพบผู้ใช้งานที่ไม่มีรหัสผ่านจะแสดงหน้าต่างข้อความเพื่อให้ผู้ดูแลระบบทำการยกเลิกการใช้งานของผู้ใช้ดังกล่าว เพื่อทดสอบการทำงานของโปรแกรมดังแสดงผลในรูปที่ 5.6 และรูปที่ 5.7

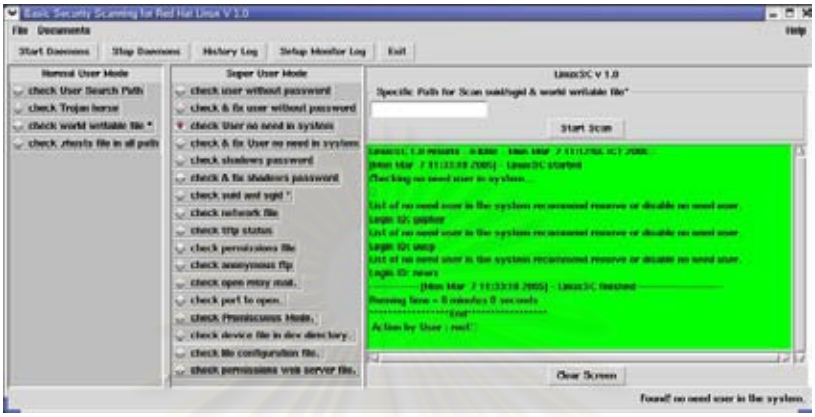


รูปที่ 5.6 หน้าต่างข้อความยืนยันการยกเลิกผู้ใช้งานที่ไม่มีรหัสผ่าน



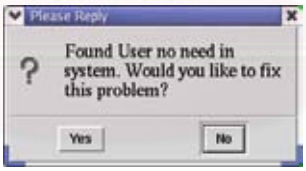
รูปที่ 5.7 การตรวจสอบผู้ใช้งานที่ไม่มีรหัสผ่านและแก้ไข

5.3.7 ตรวจสอบผู้ใช้งานที่ไม่จำเป็นในระบบ โปรแกรมตรวจสอบจะทำการตรวจสอบผู้ใช้งานที่ไม่จำเป็นในระบบลีนุกซ์ โดยอิงจากเอกสารตรวจสอบความปลอดภัยบนยูนิกซ์เวอร์ชันสอง โปรแกรมตรวจสอบจะทำการรายงานผลรายชื่อผู้ใช้งานที่ไม่จำเป็นในระบบดังแสดงผลในรูปที่ 5.8

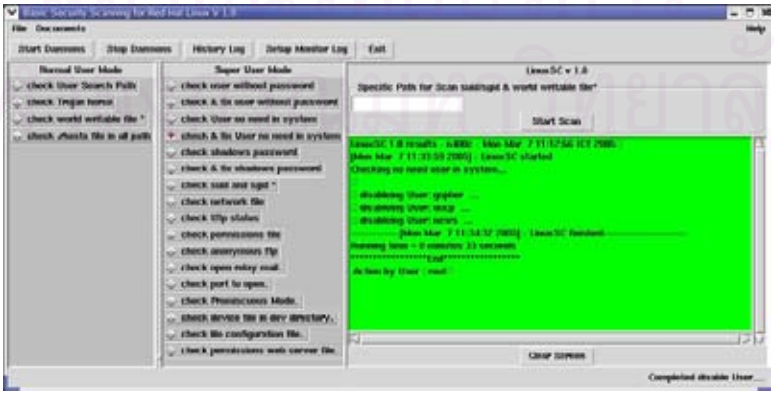


รูปที่ 5.8 การตรวจสอบผู้ใช้งานที่ไม่จำเป็นในระบบ

5.3.8 ตรวจสอบผู้ใช้งานที่ไม่จำเป็นในระบบและแก้ไข โปรแกรมตรวจสอบจะทำการตรวจสอบผู้ใช้งานที่ไม่จำเป็นในระบบ เมื่อโปรแกรมตรวจสอบผู้ใช้งานที่ไม่จำเป็นในระบบจะแสดงหน้าต่างข้อความเพื่อให้ผู้ดูแลระบบทำการยกเลิกการใช้งานของผู้ใช้ดังกล่าว ดังแสดงผลในรูปที่ 5.9 และรูปที่ 5.10

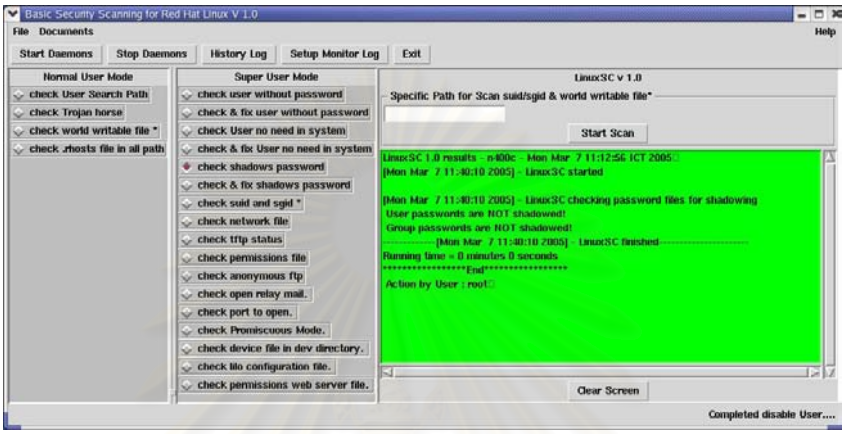


รูปที่ 5.9 หน้าต่างข้อความเพื่อยืนยันการยกเลิกผู้ใช้งานที่ไม่จำเป็น



รูปที่ 5.10 การตรวจสอบผู้ใช้งานที่ไม่จำเป็นในระบบและแก้ไข

5.3.9 การตรวจสอบชาโดว์/กรุ๊บพาสเวิสด์ โปรแกรมตรวจสอบจะทำการตรวจสอบระบบว่ามีการใช้ ชาโดว์/กรุ๊บพาสเวิสด์ หรือไม่ ในการทดสอบโดยปกติในระบบลีนุกซ์จะมีการทำ ชาโดว์พาสเวิสด์ หรือ กรุ๊บพาสเวิสด์ อยู่แล้ว จึงได้ทำการจำลองโดยการลบแฟ้ม /etc/shadow เพื่อทดสอบการทำงานของโปรแกรมดังแสดงผลในรูปที่ 5.11

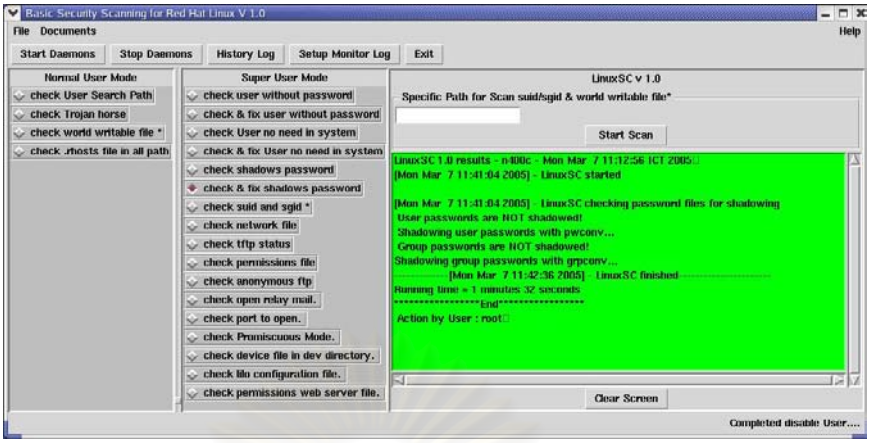


รูปที่ 5.11 การตรวจสอบ ชาโดว์/กรุ๊บพาสเวิสด์

5.3.10 การตรวจสอบชาโดว์/กรุ๊บพาสเวิสด์และแก้ไข โปรแกรมตรวจสอบจะทำการตรวจสอบการทำ ชาโดว์/กรุ๊บพาสเวิสด์ ในระบบเมื่อพบว่าไม่มีการทำ ชาโดว์/กรุ๊บพาสเวิสด์ จะแสดงหน้าต่างข้อความเพื่อให้ผู้ดูแลระบบทำการยืนยันการทำ ชาโดว์/กรุ๊บพาสเวิสด์ โดยใช้คำสั่ง “pwconv” สำหรับชาโดว์พาสเวิสด์ และคำสั่ง “grpconv” สำหรับกรุ๊บพาสเวิสด์ เพื่อทดสอบการทำงานของโปรแกรมดังแสดงผลในรูปที่ 5.12 รูปที่ 5.13 และรูปที่ 5.14

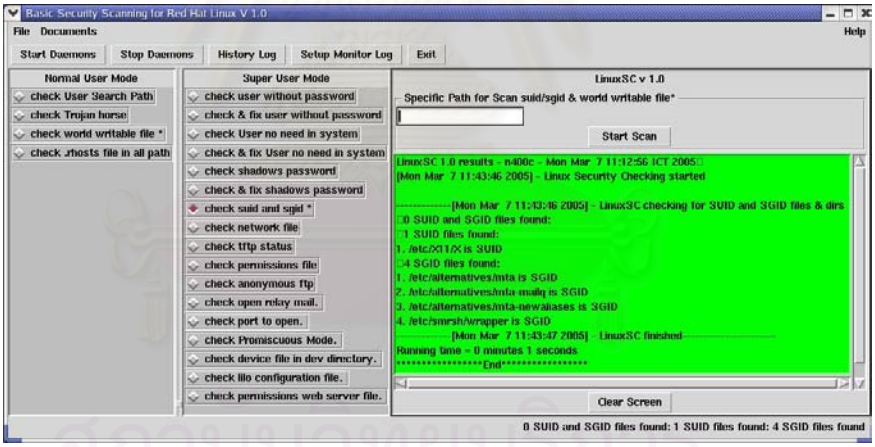
รูปที่ 5.12 หน้าต่างข้อความเพื่อยืนยันการทำชาโดว์พาสเวิสด์

รูปที่ 5.13 หน้าต่างข้อความเพื่อยืนยันการทำกรุ๊บพาสเวิสด์



รูปที่ 5.14 แสดงการตรวจสอบชาโดว์/กรุ๊บพาสเวิสด์ และแก้ไข

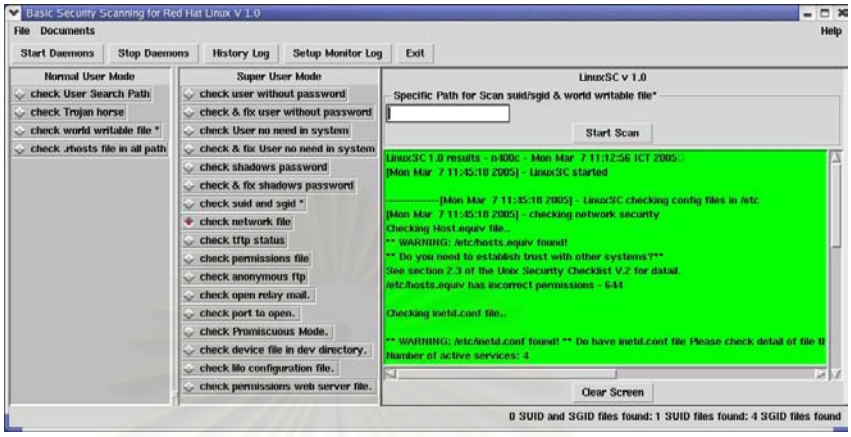
5.3.11 การตรวจสอบแฟ้ม SUID และ SGID โปรแกรมตรวจสอบจะทำการตรวจสอบระบบว่ามีแฟ้ม SUID และ SGID พร้อมทั้งรายงานผลการตรวจสอบจำนวนแฟ้มที่พบ โดยในการทดสอบจะทำการระบุไดเรกทอรีที่จะตรวจสอบ หรือเป็นการตรวจสอบทั้งระบบเพื่อทดสอบการทำงานของโปรแกรมดังแสดงผลในรูปที่ 5.15



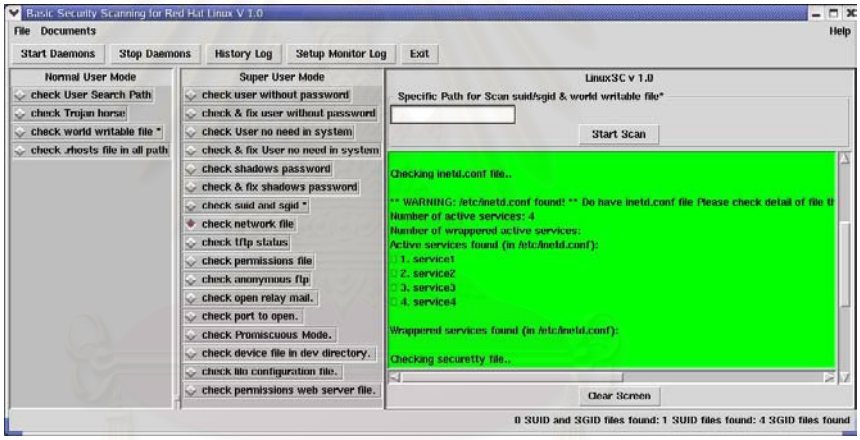
รูปที่ 5.15 การตรวจสอบแฟ้ม SUID และ SGID

5.3.12 การตรวจสอบแฟ้มเครือข่ายในระบบ โปรแกรมตรวจสอบจะทำการตรวจสอบแฟ้มเครือข่ายพื้นฐาน ได้แก่แฟ้ม /etc/inetd.conf แฟ้ม /etc/securetty แฟ้ม /etc/hosts.equiv จะเป็นการตรวจสอบเนื้อหาที่เหมาะสมของแฟ้ม ส่วนในแฟ้ม /etc/ftpusers แฟ้ม /etc/service และแฟ้ม /etc/hosts จะเป็นการตรวจสอบเฉพาะบิตอนุญาตซึ่งจะอยู่ในหัวข้อฟังก์ชันการตรวจสอบบิตอนุญาต โดยในการทดสอบจะทำการเพิ่มบริการต่างๆ เข้าไปในแฟ้ม /etc/inetd.conf และระบุชื่อผู้ใช้งานและเครื่องหมาย “+” ในแฟ้ม /etc/hosts.equiv ซึ่งเป็นอนุญาตผู้ใช้งานเข้าถึงระบบได้ซึ่งไม่เหมาะสม ส่วนในแฟ้ม /etc/securetty จะทดสอบโดยการเพิ่มค่า “pts/x” เข้าไปในแฟ้มเพื่อ

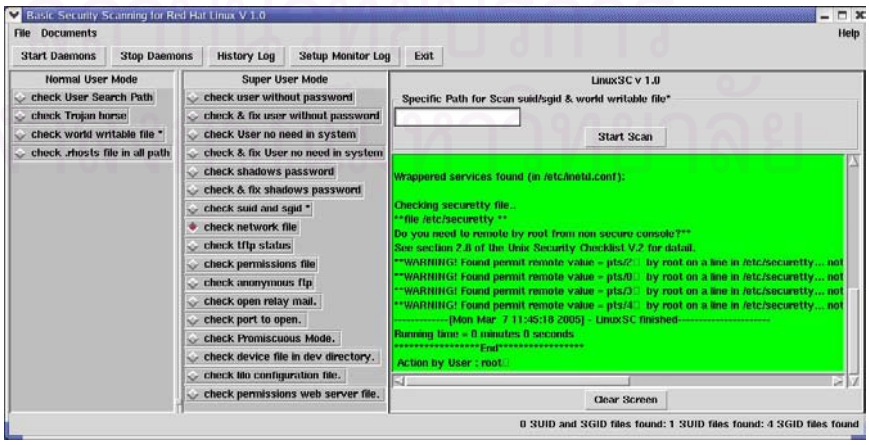
อนุญาตให้ผู้ดูและระบบสามารถล็อกอินเข้าระบบจากระยะไกลได้ เพื่อทดสอบการทำงานของโปรแกรมดังแสดงผลในรูปที่ 5.16 รูปที่ 5.17 และรูปที่ 5.18



รูปที่ 5.16 การตรวจสอบแฟ้มเครือข่าย

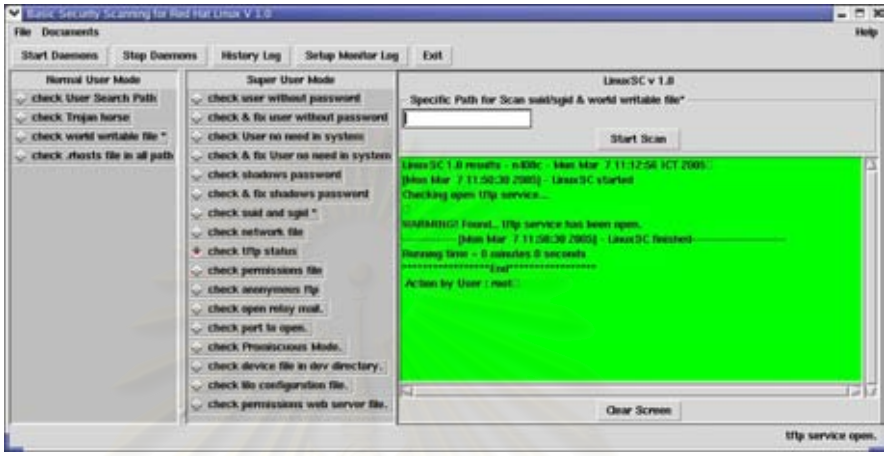


รูปที่ 5.17 การตรวจสอบแฟ้มเครือข่าย (ต่อ)



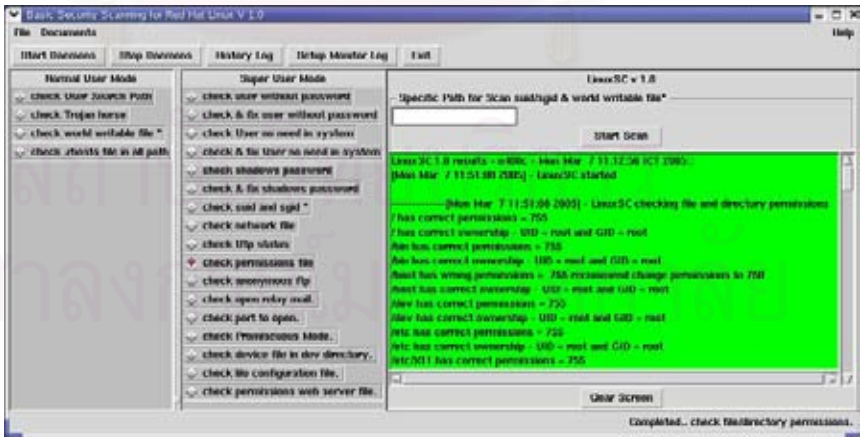
รูปที่ 5.18 การตรวจสอบแฟ้มเครือข่าย (ต่อ)

5.3.13 การตรวจสอบสถานะการบริการทีเอฟทีพี โปรแกรมตรวจสอบจะทำการตรวจสอบสถานะ การให้บริการ ทีเอฟทีพี โดยในการทดสอบจะทำการปิดและเปิดการให้บริการเพื่อทดสอบการทำงานของโปรแกรมดังแสดงผลในรูปที่ 5.19



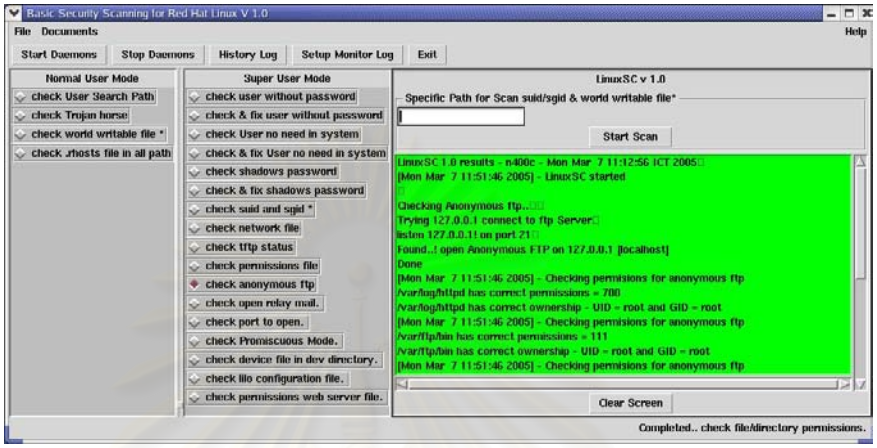
รูปที่ 5.19 การตรวจสอบสถานการณ์บริการเอฟทีพี

5.3.14 การตรวจสอบบิตอนุญาตของแฟ้ม โปรแกรมตรวจสอบ จะทำการตรวจสอบบิตอนุญาตของแฟ้มและไดเรกทอรี เปรียบเทียบตามรายละเอียดในแฟ้ม perms.dir ซึ่งแสดงในภาคผนวก จ โดยการทดสอบโปรแกรมจะทำการแสดงผลการทดสอบเปรียบเทียบและข้อแนะนำในกรณีที่บิตอนุญาตของแฟ้มที่ทำการตรวจสอบไม่ตรงกับแฟ้มบิตอนุญาตต้นแบบ ดังแสดงผลการทำงานของโปรแกรมในรูปที่ 5.20



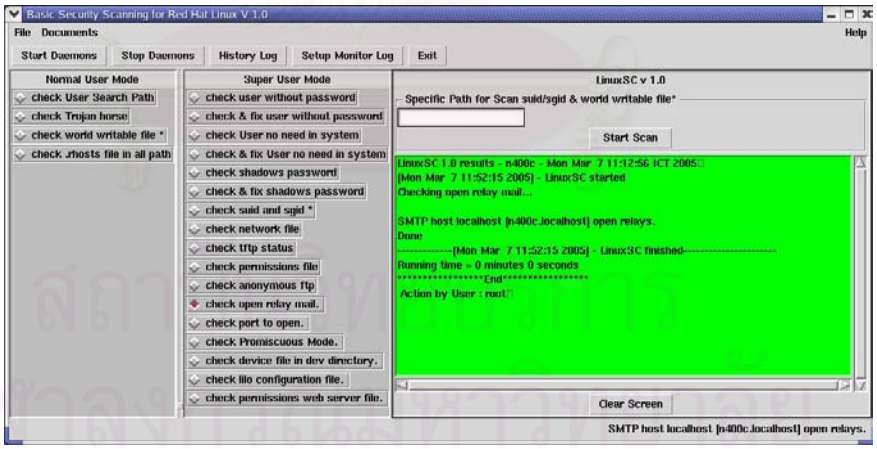
รูปที่ 5.20 การตรวจสอบบิตอนุญาตของแฟ้ม

5.3.15 การตรวจสอบสถานะเอฟทีพีนิรนาม โปรแกรมตรวจสอบทำการตรวจสอบสถานะให้บริการเอฟทีพีนิรนาม และบิตอนุญาตของแฟ้มที่ใช้ในการทำเอฟทีพีนิรนาม โดยในการทดสอบจะทำการปิดและเปิดการให้บริการเพื่อทดสอบการทำงานของโปรแกรมดังแสดงผลในรูปที่ 5.21



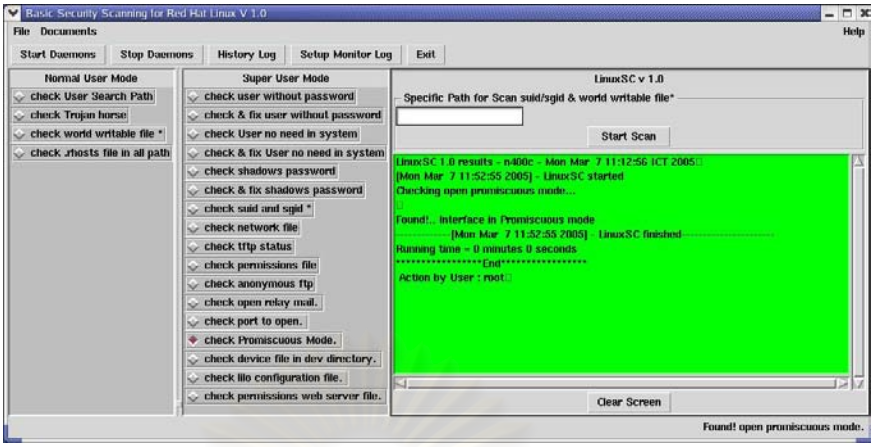
รูปที่ 5.21 การตรวจสอบสถานะเอฟทีพีแบบนิรนาม

5.3.16 การตรวจสอบสถานะการเปิดรีเลย์เมล์ โปรแกรมตรวจสอบจะทำการตรวจสอบสถานะ การให้บริการรีเลย์เมล์ โดยในการทดสอบจะทำการเปิด/ปิดการให้บริการและปรับเปลี่ยนค่าการอนุญาตรีเลย์ เพื่อทดสอบการทำงานของโปรแกรมดังแสดงผลในรูปที่ 5.22



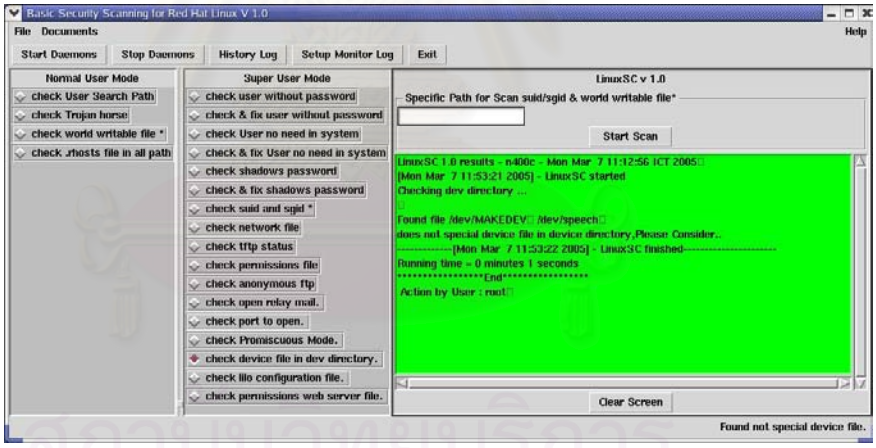
รูปที่ 5.22 การตรวจสอบสถานะ การเปิด รีเลย์เมล์

5.3.17 การตรวจสอบอินเตอร์เฟสในภาวะการทำงานแบบไม่เลือก โปรแกรมตรวจสอบจะทำการตรวจสอบว่าอินเตอร์เฟสอยู่ในภาวการณ์ทำงานแบบไม่เลือกหรือไม่ ในการทดสอบจะทำการใช้คำสั่ง “ifconfig interface name promisc” เพื่อให้อินเตอร์เฟสอยู่ในภาวะ การทำงานแบบไม่เลือก เพื่อทดสอบการทำงานของโปรแกรมดังแสดงผลในรูปที่ 5.23



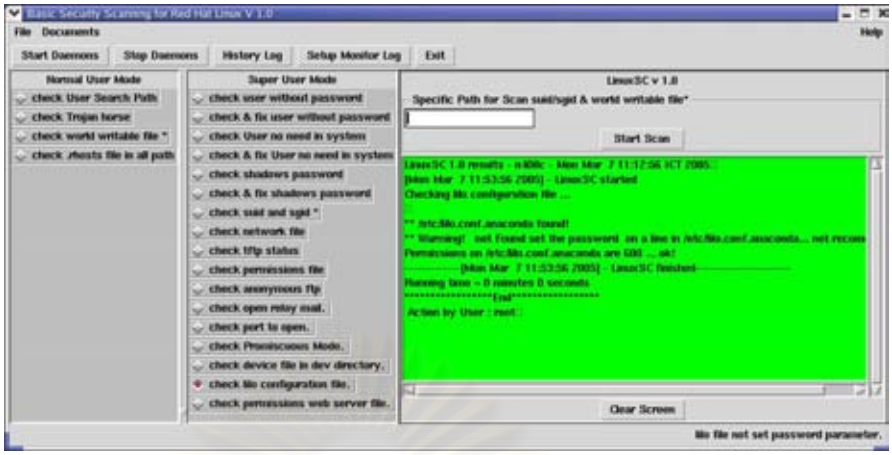
รูปที่ 5.23 การตรวจสอบอินเตอร์เฟสในภาวะการทำงานแบบไม่เลือก

5.3.18 การตรวจสอบแฟ้มในไดเรกทอรีอุปกรณ์ โปรแกรมตรวจสอบทำการตรวจสอบแฟ้มที่ไม่ใช่แฟ้มอุปกรณ์ที่อยู่ในไดเรกทอรีอุปกรณ์ ในการทดสอบจะทำการสร้างแฟ้มขึ้นมาในไดเรกทอรีอุปกรณ์ซึ่งเป็นแฟ้มที่ไม่ใช่แฟ้มอุปกรณ์เพื่อทดสอบการทำงานของโปรแกรมดังแสดงผลในรูปที่ 5.24



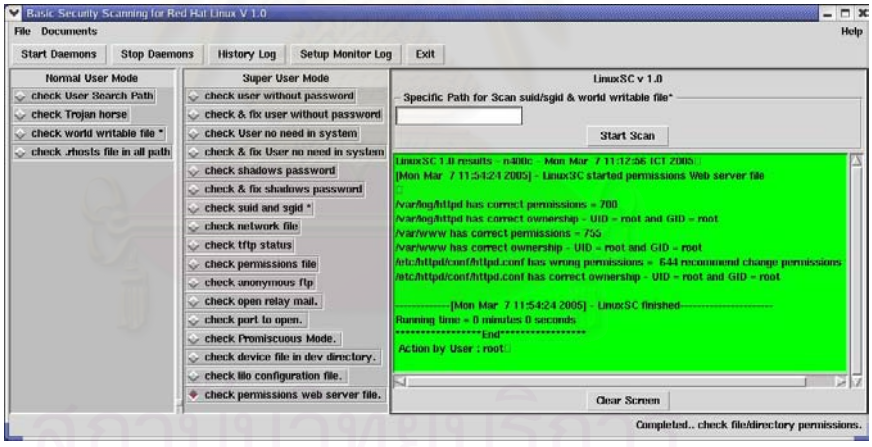
รูปที่ 5.24 การตรวจสอบแฟ้มอุปกรณ์ในไดเรกทอรีอุปกรณ์

5.3.19 การตรวจสอบแฟ้ม lilo.conf โปรแกรมตรวจสอบจะทำการตรวจสอบพารามิเตอร์รหัสผ่าน และบิตอนุญาตที่เหมาะสมของแฟ้ม lilo.conf การทดสอบจะทำการยกเลิกพารามิเตอร์รหัสผ่าน ในแฟ้ม lilo.conf เพื่อทดสอบการทำงานของโปรแกรมดังแสดงผลในรูปที่ 5.25



รูปที่ 5.25 การตรวจสอบแฟ้ม lilo.conf

5.3.20 การตรวจสอบบิตอนุญาตของแฟ้มเว็บเซิร์ฟเวอร์ โปรแกรมตรวจสอบจะทำการตรวจสอบ บิตอนุญาตที่เหมาะสมของแฟ้มในการทำเว็บเซิร์ฟเวอร์ โดยในการทดสอบจะทำการปรับเปลี่ยนบิตอนุญาตของแฟ้มที่จะทำการตรวจสอบเพื่อทดสอบการทำงานของโปรแกรมดังแสดงผลในรูปที่ 5.25



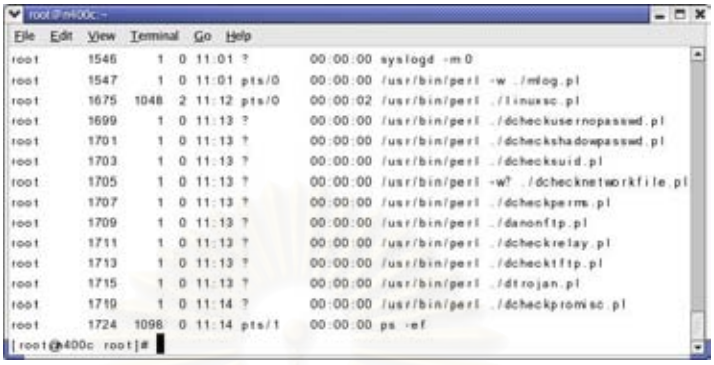
รูปที่ 5.25 การตรวจสอบบิตอนุญาตของแฟ้มเว็บเซิร์ฟเวอร์

5.4 การทดสอบฟังก์ชันการทำงานของโปรแกรมตรวจสอบในแบบดีมอน

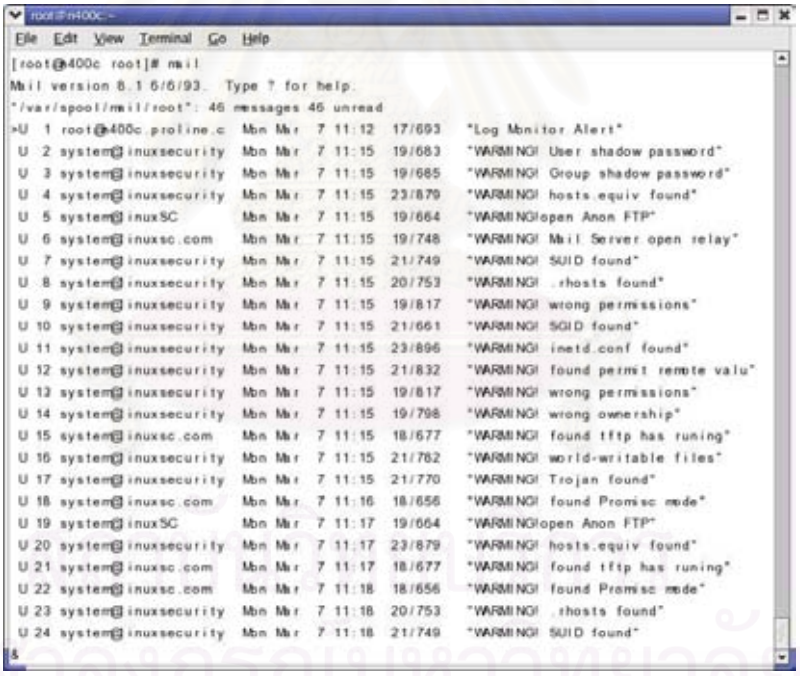
5.4.1 การทดสอบการทำงานของโปรแกรมดีมอนและการแจ้งผลทางเมล์

เมื่อโปรแกรมเริ่มต้นการทำงานในแบบดีมอน จะมีการทำงานเช่นเดียวกันกับในแบบปกติ แต่ในส่วนการแสดงผลจะเป็นในรูปแบบของอีเมล์ โดยการทดสอบจะใช้เงื่อนไขในการทดสอบเช่น

เดียวกันกับในแบบปกติแต่จะตรวจสอบเฉพาะในการแสดงผล ในที่นี้จะเป็นรูปแบบอีเมล์แจ้งไปยังผู้ดูแลระบบเมื่อพบปัญหา ดังแสดงผลในรูปที่ 5.26 และ รูปที่ 5.27

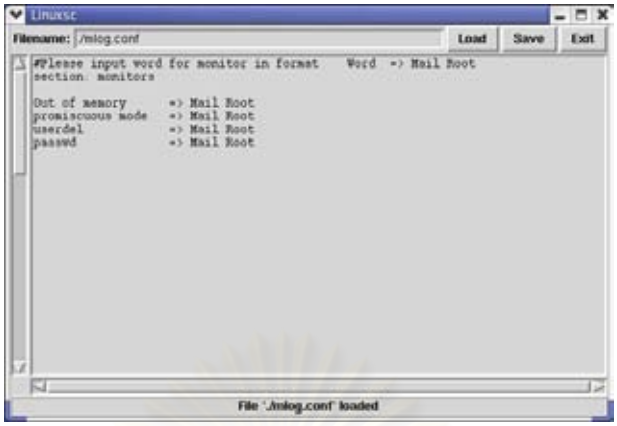


รูปที่ 5.26 โปรเซสของการทำงานโปรแกรมแบบดีมอน

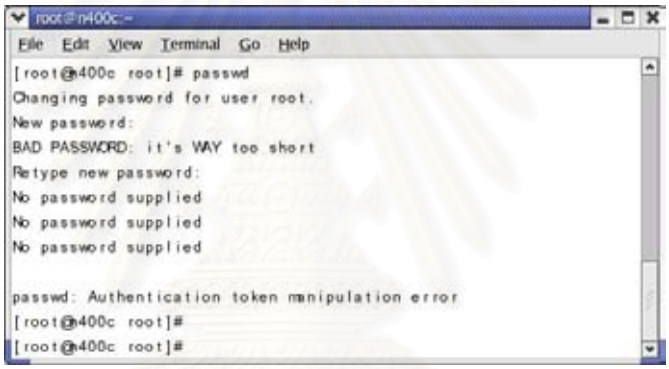


รูปที่ 5.27 ลักษณะเมล์ที่แจ้งไปยังผู้ดูระบบในกรณีพบปัญหา

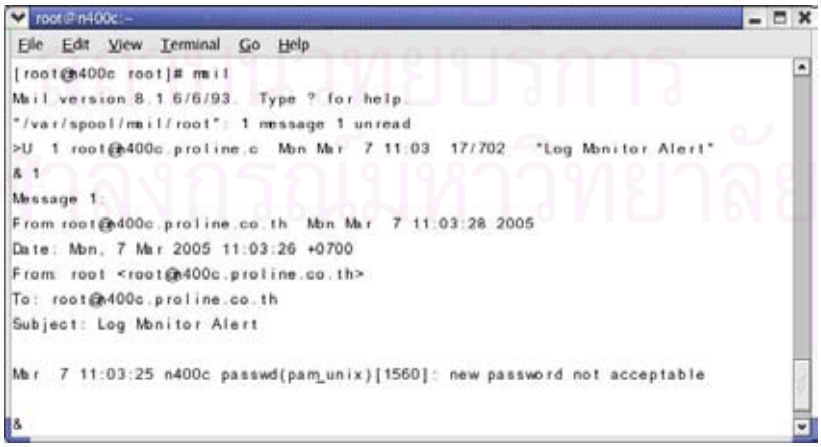
5.4.2 การทดสอบการตรวจสอบล็อกในระบบ ในการทดสอบจะทำการกำหนดคำ ที่จะใช้ในการตรวจสอบในตัวอย่างการทดสอบจะตรวจสอบคำว่า “passwd” ซึ่งเป็นคำสั่งที่ใช้ในการเปลี่ยนรหัสผ่านของผู้ใช้งานในระบบเพื่อต้องการทราบเมื่อมีการพยายามเปลี่ยนรหัสผ่านในระบบซึ่งอาจโดนขโมยรหัสผ่านโดยผู้ไม่หวังดีซึ่งเมื่อโปรแกรมตรวจสอบในล็อกพบ คำดังกล่าว จะทำการส่งเมล์แจ้งไปยังผู้ดูแลระบบ ดังแสดงผลในรูปที่ 5.28 รูปที่ 5.29 และรูปที่ 5.30



รูปที่ 5.28 การกำหนดคำหรือข้อความในการตรวจสอบล็อก



รูปที่ 5.29 การใช้คำสั่งเปลี่ยนรหัสผ่านของผู้ใช้งานในระบบ



รูปที่ 5.30 รายละเอียดเนื้อหาของเมล์แจ้งไปยังผู้ดูแลระบบ

สรุปรายงานผลการทดสอบการทำงานในแต่ฟังก์ชันในตารางที่ 5.2

ตารางที่ 5.2 ผลการตรวจสอบของการทำงานแต่ละฟังก์ชันการตรวจสอบ

ฟังก์ชันการตรวจสอบ	**กำหนดสภาพแวดล้อมในการตรวจสอบ**	Redhat V.8	Redhat V.9
		การตรวจพบ	**การตรวจพบ**
ตรวจสอบเส้นทางค้นหาที่เป็นอันตราย	ทดสอบกำหนดค่า “.” ในแฟ้ม .profile ของผู้ใช้งาน	ตรวจพบ	ตรวจพบ
ตรวจสอบช่องทางสื่อสารของม้าโทรจัน	ติดตั้งโปรแกรมโทรจันพื้นฐาน netcat	ตรวจพบ	ตรวจพบ
ตรวจสอบแฟ้มที่เปิดสิทธิเต็ม	เปิดสิทธิเต็มของแฟ้มที่ตรวจสอบ	ตรวจพบ	ตรวจพบ
ตรวจสอบแฟ้ม .rhost	สร้างแฟ้ม .rhost ในไดเรกทอรีต่างๆ	ตรวจพบ	ตรวจพบ
ตรวจสอบผู้ใช้งานที่ไม่มีรหัสผ่าน	ลบรหัสผ่านของผู้ใช้งาน	ตรวจพบ	ตรวจพบ
ตรวจสอบผู้ใช้งานที่ไม่มีรหัสผ่านและแก้ไข	ลบรหัสผ่านของผู้ใช้งาน	ตรวจพบ/แก้ไขปัญหา	ตรวจพบ/แก้ไขปัญหา
ตรวจสอบผู้ใช้งานที่ไม่จำเป็นในระบบ	ใช้ค่าปริยายของระบบ	ตรวจพบ	ตรวจพบ
ตรวจสอบผู้ใช้งานที่ไม่จำเป็นในระบบและแก้ไข	ใช้ค่าปริยายของระบบ	ตรวจพบ/แก้ไขปัญหา	ตรวจพบ/แก้ไขปัญหา
ตรวจสอบ ชาโดว์/กรุ๊บพาสเวิสด์	ใช้ค่าปริยายของระบบ	ตรวจพบ	ตรวจพบ
ตรวจสอบ ชาโดว์/กรุ๊บพาสเวิสด์และแก้ไข	ใช้ค่าปริยายของระบบ	ตรวจพบ/แก้ไขปัญหา	ตรวจพบ/แก้ไขปัญหา
ตรวจสอบแฟ้ม suid และ sgid	ใช้ค่าปริยายของระบบ	ตรวจพบ	ตรวจพบ
ตรวจสอบแฟ้ม เครือข่ายในระบบ	สร้างแฟ้มและเพิ่มเงื่อนไขที่ไม่แนะนำให้กับแฟ้ม เครือข่ายในระบบ	ตรวจพบ	ตรวจพบ
ตรวจสอบสถานะการบริการ tftp	ทดสอบเปิด/ปิด บริการทีเอฟทีพี	ตรวจพบ	ตรวจพบ
ตรวจสอบ บิตอนุญาตของแฟ้มของแฟ้ม	เปรียบเทียบกับแฟ้ม perms.dir	ตรวจพบ	ตรวจพบ
ตรวจสอบสถานะเอฟทีพีนิรนาม	เปิดบริการ เอฟทีพีนิรนาม	ตรวจพบ	ตรวจพบ
ตรวจสอบสถานะการเปิด รีเลย์ mail	ปรับเปลี่ยนค่าในการอนุญาต รีเลย์	ตรวจพบ	ตรวจพบ
ตรวจสอบภาวะ การทำงานแบบไม่เลือก	เปิดภาวะ การทำงานแบบไม่เลือกให้กับอินเตอร์เฟส	ตรวจพบ	ตรวจพบ
ตรวจสอบแฟ้มอุปกรณ์ในไดเรกทอรี dev	สร้างแฟ้มที่ไม่ใช่แฟ้มอุปกรณ์ในไดเรกทอรี dev	ตรวจพบ	ตรวจพบ
ตรวจสอบแฟ้ม lilo.conf	ตัดพารามิเตอร์รหัสผ่าน และทดสอบเปลี่ยนบิตอนุญาตของแฟ้ม	ตรวจพบ	ตรวจพบ
ตรวจสอบบิตอนุญาตของไดเรกทอรีเว็บเซิร์ฟเวอร์	ทดสอบเปลี่ยนบิตอนุญาตของไดเรกทอรีเว็บเซิร์ฟเวอร์	ตรวจพบ	ตรวจพบ
ทดสอบการตรวจสอบล็อกในระบบ	กำหนดคำที่ใช้ในการตรวจสอบ	ตรวจพบ	ตรวจพบ

5.5 รูปแบบการแสดงผล

สำหรับรูปแบบส่วนของหน้าจอติดต่อกับผู้ใช้ สามารถแสดงในภาคผนวก ก ในภาคผนวก ข และภาคผนวก ค จะเป็นการแสดงฟังก์ชันการทำงาน ในโหมดผู้ดูแลระบบและในโหมดผู้ใช้งานตามลำดับ ส่วนภาคผนวก ง เป็นการกำหนดค่าคอนฟิกระบบและการแสดงผลในรูปแบบภาษาไทย

5.6 ข้อแนะนำในการนำโปรแกรมไปใช้งาน

5.6.1 ผู้ใช้งานสามารถเรียกโปรแกรมตรวจสอบความปลอดภัย โดยผ่านทางส่วนติดต่อผู้ใช้ในสภาพแวดล้อมแบบกราฟิกในระบบเอกซ์วินโดว์ โดยเลือกหัวข้อที่จะตรวจสอบได้ โดยแบ่งเป็น 2 โหมด ได้แก่ โหมดผู้ใช้งาน และ โหมดผู้ดูแลระบบ

5.6.2 ในกรณีที่ผู้ใช้งานต้องการให้โปรแกรมทำงานแบบดีมอนผู้ใช้งานสามารถที่จะเรียกโปรแกรมทำงานผ่านทางส่วนติดต่อผู้ใช้ หรือเรียกโปรแกรมทำงานแบบดีมอนได้โดยตรงจากชื่อโปรแกรมนั้นจากเชลล์

5.6.3 การเปลี่ยนแปลงอินเตอร์เฟส เพิ่มหรือลด แต่ละฟังก์ชันของการตรวจสอบความปลอดภัยในโหมดผู้ดูแลระบบ ผู้ใช้งานจะต้องเรียกโปรแกรมกำหนดค่าดังกล่าวโดยตรงจากเชลล์โดยที่ไม่ผ่านทางส่วนติดต่อผู้ใช้

5.6.4 ผู้ใช้งานสามารถที่จะกำหนดให้โปรแกรมแต่ละโปรแกรมที่ทำงานในแบบดีมอน มีช่วงเวลาในการทำงานของโปรแกรม โดยให้ตรวจสอบทุกๆ กี่นาทีแยกตามแต่ละโปรแกรมโดยขึ้นอยู่กับลำดับตามความสำคัญของการตรวจสอบเพื่อประหยัดทรัพยากรของระบบ ณ ขณะนั้น โดยปกติค่าช่วงเวลาในการตรวจสอบจะตรวจสอบทุกๆ 60 นาที ผู้ใช้งานจะต้องเรียกโปรแกรมกำหนดค่าช่วงเวลาการตรวจสอบ โดยตรงจากเชลล์โดยที่ไม่ผ่านทางส่วนติดต่อผู้ใช้

5.6.5 ผู้ใช้งานสามารถที่จะเพิ่มขอบเขต ของเส้นทางของไดเรกทอรี ที่โปรแกรมจะตรวจสอบในแฟ้ม find.dir ซึ่งเก็บเส้นทางของไดเรกทอรีที่จะตรวจสอบของระบบไว้ซึ่งโดยปกติจะเก็บเส้นทางไดเรกทอรีมาตรฐานไว้ ได้แก่ /boot /dev /etc /home /lib /opt /root /sbin /share /src /tmp /usr /var

5.6.6 ผู้ใช้งานสามารถที่จะกำหนดหรือเพิ่ม ต้นแบบบิตอนุญาตของแฟ้ม ได้ในแฟ้ม perms.dir ซึ่งเก็บต้นแบบบิตอนุญาตของไดเรกทอรีและแฟ้ม ในกรณีที่ลีนุกซ์มีการเพิ่มเติมแฟ้มในระบบนอกเหนือจากที่มีอยู่ในปัจจุบัน

บทที่ 6
สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ

6.1 สรุปผลการวิจัย

ผลจากการวิจัยครั้งนี้ทำให้ได้ โปรแกรมตรวจสอบความปลอดภัยพื้นฐานสำหรับระบบปฏิบัติการลีนุกซ์เรดแฮต ซึ่งมีส่วนติดต่อผู้ใช้มีลักษณะเป็นกราฟิก ที่ใช้งานง่ายประกอบด้วยฟังก์ชันที่ใช้ในการตรวจสอบจุดหละหลวมและความปลอดภัยพื้นฐานต่างๆ ในระบบ ซึ่งอิงตามเอกสารตรวจสอบความปลอดภัยพื้นฐานบนยูนิกซ์ ของหน่วยงาน เซิร์ต และหน่วยงาน เอยูเอสเซิร์ต ซึ่งพัฒนาโดยใช้ภาษา เพิร์ล และเพิร์ลทีเค และโปรแกรมอรรถประโยชน์ต่างๆ ที่มีอยู่ในระบบปฏิบัติการลีนุกซ์ตระกูลเรดแฮต

จากการทดสอบโปรแกรม พบว่าสามารถตรวจสอบพบจุดหละหลวมต่างๆ ในระบบได้รวดเร็ว โดยผู้ใช้โปรแกรม สามารถที่จะเลือกตรวจสอบในโหมดผู้ใช้งาน หรือโหมดผู้ดูแลระบบเฉพาะหัวข้อที่ตนเองต้องการ ซึ่งจะแสดงผลในขณะนั้น หรือสามารถเก็บผลลัพธ์ไว้ในแฟ้มฮีสทอรีล็อก (histoty log) แล้วสามารถย้อนดูที่หลังได้ ทั้งยังสามารถกำหนดให้โปรแกรมแต่ละฟังก์ชันแยกทำงานในลักษณะดีมอน โดยเลือกฟังก์ชันที่ต้องการและสามารถกำหนดความถี่ของระยะเวลาในการตรวจสอบในแบบดีมอน เพื่อไม่ให้กินทรัพยากรของระบบมากเกินไปในระบบ กรณีที่พบจุดหละหลวม โปรแกรมสามารถที่จะส่งเมล์แจ้งไปยังผู้ดูแลระบบ และสามารถแสดงส่วนติดต่อผู้ใช้ และแสดงผลลัพธ์ในรูปแบบภาษาไทย ถ้าระบบปฏิบัติการนั้นมีฟอนต์ภาษาไทยติดตั้งอยู่

6.2 อภิปรายผลการวิจัย

6.2.1 เนื่องจากว่าปัจจุบันระบบปฏิบัติการลีนุกซ์ตระกูลเรดแฮดได้พัฒนามาจนมีหลากหลายเวอร์ชันจากเวอร์ชันที่เป็นโอเพนซอส (open source) จนถึงเวอร์ชันที่ทำออกมาในรูปแบบของเชิงพาณิชย์ ในรูปแบบเชิงพาณิชย์เองก็ยังแบ่งชนิดออกไปหลายชนิด ซึ่งระบบปฏิบัติการลีนุกซ์ตระกูลเรดแฮดในเวอร์ชันใหม่ๆ ก็มีความแตกต่างกันในด้านคำสั่งที่ใช้บางคำสั่ง หรือมีคำสั่งบางคำสั่งที่เพิ่มเข้ามาใหม่ ดังนั้นในการเขียนโปรแกรมจึงต้องคำนึงถึงความเข้ากันได้ของโปรแกรม ดังนั้นคำสั่งบางคำสั่งซึ่งถึงแม้จะมีประสิทธิภาพหรือทำงานได้เร็ว ก็ไม่สามารถที่จะนำมาใช้ในโปรแกรมได้ จำเป็นต้องใช้คำสั่งอื่นหรือคำสั่งเดิมที่มีอยู่ซึ่งถึงแม้จะด้อยประสิทธิภาพ

แต่แพร่หลายมากกว่าแทน

6.2.2 โปรแกรมตรวจสอบความปลอดภัยพื้นฐานสำหรับระบบปฏิบัติการลีนุกซ์เรดแฮต นี้ใช้สำหรับตรวจสอบจุดหละหลวมพื้นฐานซึ่งมักพบเสมอในระบบปฏิบัติการลีนุกซ์ทั่วไป ในกรณีที่ต้องการตรวจสอบจุดหละหลวมที่เฉพาะเจาะจง หรือตรวจสอบในระบบปฏิบัติการลีนุกซ์เวอร์ชันใหม่ๆหรือระบบปฏิบัติการลีนุกซ์ ที่มีการคอนฟิกหรือติดตั้งโปรแกรมหรือส่วนประกอบเพิ่มเติมจากค่าคอนฟิกพื้นฐานของระบบ ผู้ใช้จำเป็นต้องเข้าไปตรวจสอบด้วยตนเอง

6.2.3 ปัจจุบันเนื่องจากมีการพัฒนาโปรแกรมประยุกต์ใหม่ๆ ในระบบปฏิบัติการลีนุกซ์ตลอดเวลา ซึ่งโปรแกรมประยุกต์ใหม่ๆเหล่านี้ บางครั้งอาจมีจุดหละหลวมแฝงเร้น ดังนั้นจึงจำเป็นต้องคอยติดตามข้อมูลเพื่อที่จะสามารถปรับปรุงโปรแกรมให้สามารถตรวจสอบได้

6.2.4 เนื่องจากโปรแกรมตรวจสอบความปลอดภัย ฟังก์ชันในการตรวจสอบจะอิงตามจากเอกสารตรวจสอบความปลอดภัยพื้นฐานบนยูนิกซ์เวอร์ชันสอง ของหน่วยงานเซิร์ตและองค์กรเอยูเอสเซิร์ต ซึ่งย่อมจะมีการเพิ่มเติมข้อมูลหรือตัดข้อมูลบางอย่างที่ล้าสมัยในเรื่องความปลอดภัยและข้อแนะในการตรวจสอบความปลอดภัย ดังนั้นจึงจำเป็นต้องคอยติดตามข้อมูลเพื่อที่จะสามารถปรับปรุงโปรแกรมให้ทันสมัย

6.3 ข้อเสนอแนะ

6.3.1 เพิ่มเติมการตรวจสอบจุดหละหลวมอื่นๆ เช่น เน็ตเวิร์คไฟล์ซิสเต็ม ดีเอนเอสเซิร์ฟเวอร์ และการตรวจสอบบริการ ที่ใช้โปรโตคอลยูดีพี

6.3.2 โปรแกรมควรที่จะพิมพ์รายงานสรุปถึงผลการตรวจสอบ และจุหละหลวมทั้งหมดที่ค้นพบ โดยรายงานสามารถที่จะเก็บในรูปแบบของแฟ้มส่งทางอิเลคทรอนิกส์เมล์ให้กับผู้ใช้ หรือพิมพ์ออกทางเครื่องพิมพ์ได้ทันที

6.3.3 พัฒนาโปรแกรมให้สามารถตรวจสอบกับดิสทริบิวชันของลีนุกซ์ค่ายอื่นๆ

6.3.4 ในส่วนติดต่อผู้ใช้ ซึ่งจำเป็นต้องมี เพิร์ลทีเค ติดตั้งเพื่อเรียกไลบรารีกราฟิกเข้ามาใช้ซึ่งไม่ใช่โปรแกรมพื้นฐานที่มีมากับระบบปฏิบัติการลีนุกซ์ จึงควรใช้โปรแกรมที่สามารถคอมไพล์ส่วนติดต่อผู้ใช้เข้ากับส่วนโปรแกรมในการพัฒนา เพื่อให้มีความยืดหยุ่นในการนำโปรแกรมไปใช้งาน

6.3.5 เพิ่มเติมความสามารถในการเลือกขนาดของฟอนต์ สี เพื่อใช้ในการแสดงผล

6.3.6 ในส่วนฐานข้อมูลช่องทางการสื่อสารของโทรจัน ให้ผู้ใช้งานสามารถเพิ่มเติมได้ภายหลัง

6.4 แนวทางวิจัยต่อ

6.4.1 พัฒนาเป็นระบบผู้เชี่ยวชาญ (Expert System) ทางด้านการตรวจสอบความปลอดภัยในระบบปฏิบัติการยูนิกซ์

รายการอ้างอิง

ภาษาอังกฤษ

[1] Randal L. Schwartz and Tom Christiansen. Learning Perl. Second Edition. (n.p.): O'Reilly & Associates Publishers,1997.

[2] Steve Lidie and Nancy Walsh. Mastering Perl/Tk. First Edition. (n.p.): O'Reilly & Associates Publishers,2002.

[3] Joel Scambray, Stuart McClure, and George Kurtz. HACKING EXPOSED. Second Edition. (n.p.): 2001.

[4] Simson Garfinkel and Gene Spafford. Practical Unix & Internet Security. (n.p.): O'Reilly & Associates Publishers, 1996.

[5] Michael D. Bauer. Building Secure Servers with LINUX. (n.p.): O'Reilly & Associates Publishers, 2002.

[6] Sriram Srinivasan. Advanced Perl Programming. (n.p.): O'Reilly & Associates Publishers, 1997.

[7] Tom Christiansen and Nathan Torkington. Perl CookBook. (n.p.): O'Reilly & Associates Publishers, 1997.

[8] W. Richard Stevens. Unix Network Programming. (n.p.): Prentice Hall, 1997.

[9] Deborah Russell and G.T. Gangemi Sr. Computer Security Basics. (n.p.): O'Reilly & Associates Publishers,1991.

[10] Thongchai Rojkangsadan. Development of Security Checking Program for UNIX. Master's Thesis. Department of Computer Engineering. Graduate School. Chulalongkorn University, 1993.

[11] Red Hat Linux Security Guide. Available from : http://www.redhat.com/docs/manuals/linux/rhl-sg-en-9.pdf

[12] The Official Red Hat Linux Reference Guide. Available from : http://www.redhat.com/docs/manuals/linux/RHL-8.0-Manual/rhl-rg-en-80.pdf

[13] CERT and CERT Coordination Center. Unix Security Checklist V2.0. Available from : http://www.cert.org/tech_tips/unix_security_checklist2.0.html

[14] SATAN (Security Admin Tool For Analyzing Networks). Available from : http://www.fish.com/satan/

[15] SAINT Corporation. SAINT. Available from : http://www.saintcorporation.com/products/saint_documentation.html

[16] Daniel Farmer, Eugene H. Spafford. The COPS Security Checker System. Purdue University Technical Report CSD-TR-933.September 19, 1991.

[17] CVE (Common Vulnerabilities and Exposure). Available from : http://www.cve.mitre.org/

[18] The Linux Thai-HOWTO. Available from : http://linux.thai.net/~sfalpha/thai-howto/Thai-HOWTO.html

[19] Thai Extension. Available from : http://linux.thai.net/plone/TLWG/TE

[20] Secure Mail Server. Available from : http://thaicert.nectec.or.th/paper/unix_linux/sendmail.php

[21] Configuration Xinetd. Available from : http://thaicert.nectec.or.th/paper/unix_linux/xinetd.php

[22] Set System Access Security Policies. Available from : http://thaicert.nectec.or.th/paper/unix_linux/set_system_access_security_poilcies.php

[23] File Permission. Available from : http://thaicert.nectec.or.th/paper/unix_linux/file_permission.php

[24] Know About Syslogd. Available from : http://thaicert.nectec.or.th/paper/unix_linux/linux_syslog.php

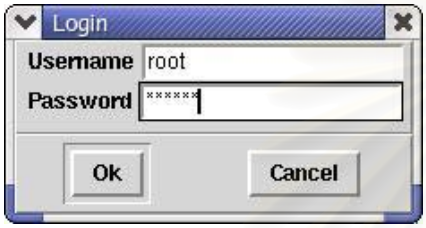
[25] Netcat. Available from : http://www.atstake.com/research/tools/network_utilities/

ภาคผนวก

ภาคผนวก ก

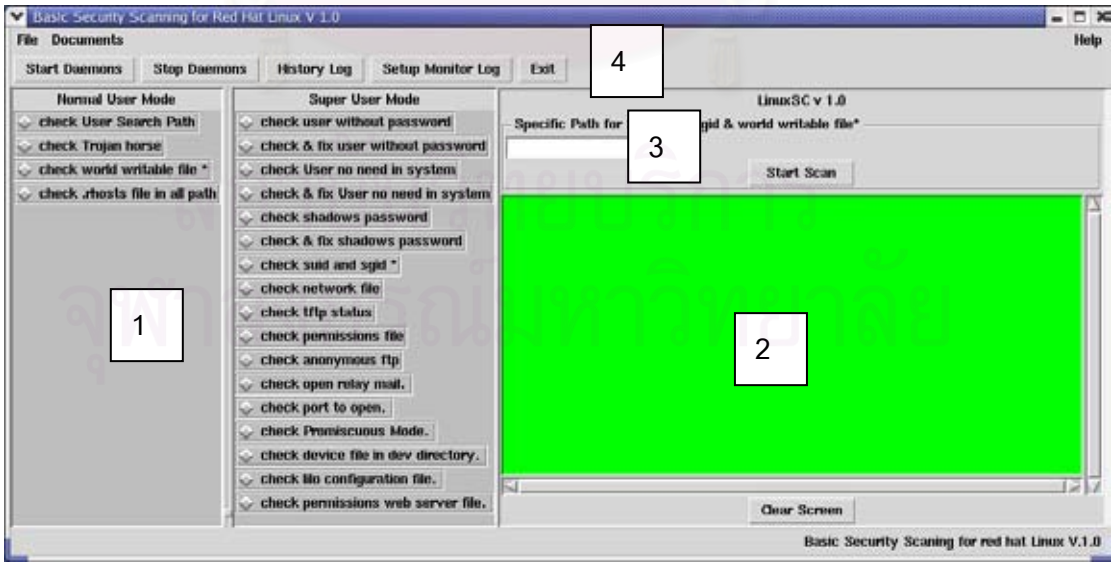
ส่วนของหน้าจอติดต่อกับผู้ใช้

เมื่อเรียกโปรแกรมจะปรากฏหน้าต่างล็อกอินเข้าสู่โปรแกรมเพื่อตรวจสอบโหมดของผู้ใช้งาน



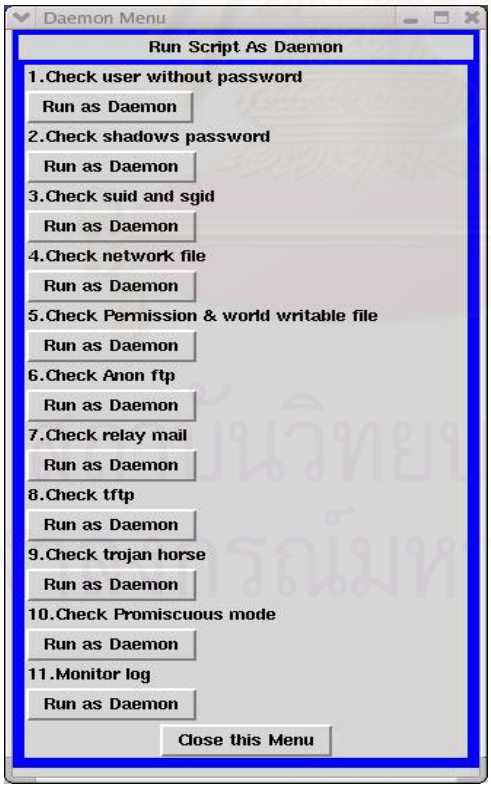
ในกรณีที่ผู้ใช้งานเป็นรูท โปรแกรมจะเข้าสู่โหมดผู้ดูแลระบบ

หน้าจอหลักของโปรแกรมในโหมดผู้ดูแลระบบ



หน้าจอหลักของโปรแกรมจะประกอบไปด้วยส่วนหลักๆคือ

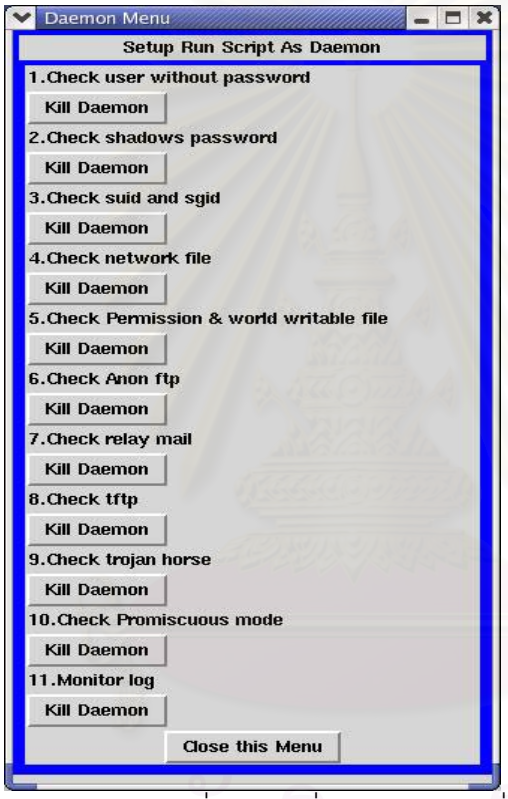
1. ฟังก์ชันการตรวจสอบในหมวดต่างๆ โดยผู้ใช้งานสามารถที่จะเลือกจากส่วนด้านซ้ายของโปรแกรมซึ่งมีลักษณะเป็นเรดิโอบัททอน โดยในโหมดผู้ดูแลระบบจะมีฟังก์ชันการตรวจสอบในโหมดผู้ใช้งานอยู่ด้วยแต่จะทำการตรวจสอบทั้งระบบซึ่งจะต่างกับในโหมดผู้ใช้งานที่จะตรวจสอบเฉพาะไดเรกทอรีบ้านของผู้ใช้งานที่ใช้โปรแกรมอยู่ ณ ขณะนั้น จากนั้นทำการกดปุ่ม Start Scan โปรแกรมจะเริ่มทำการตรวจสอบพร้อมแสดงผล
2. ส่วนแสดงผล ซึ่งมีลักษณะเป็นพื้นสีเขียว ตัวอักษรในการแสดงผลจะเป็นสีดำและในด้านล่างส่วนแสดงผลจะมีปุ่ม Clear Screen ซึ่งใช้ในการเคลียร์ส่วนแสดงผลหลังจากทำการตรวจสอบ
3. ช่องระบุไดเรกทอรีในการตรวจสอบ จะใช้สำหรับระบุขอบเขตของไดเรกทอรีที่ต้องการตรวจสอบ โดยจะใช้เฉพาะสำหรับฟังก์ชัน การตรวจสอบแฟ้ม SUID SGID และแฟ้มที่มีการเปิดสิทธิทั้งหมด
4. ปุ่มคำสั่งในโปรแกรมตรวจสอบ ซึ่งประกอบไปด้วยปุ่มคำสั่งและหน้าที่ดังต่อไปนี้
   - ชุดปุ่มคำสั่ง Start Daemon ซึ่งใช้ในการรันโปรแกรมตรวจสอบในแบบดีมอน โดยจะแบ่งออกเป็นโปรแกรมย่อยเพื่อใช้เลือกรันโปรแกรมในการตรวจสอบ



ในการใช้งาน ผู้ใช้งานสามารถที่จะเลือกฟังก์ชันที่ใช้ในการตรวจสอบแบบ ดีมอน ในแต่ละฟังก์ชันได้ตามความต้องการ เช่น ถ้าต้องการเลือกฟังก์ชันในการตรวจสอบ

ภาวะ การทำงานแบบไม่เลือก ในระบบ สามารถเลือกปุ่มในหัวข้อที่ต้องการเพื่อสั่งให้รันในแบบดีมอนจะปรากฏหน้าต่างแสดงการทำงานของฟังก์ชันนั้น

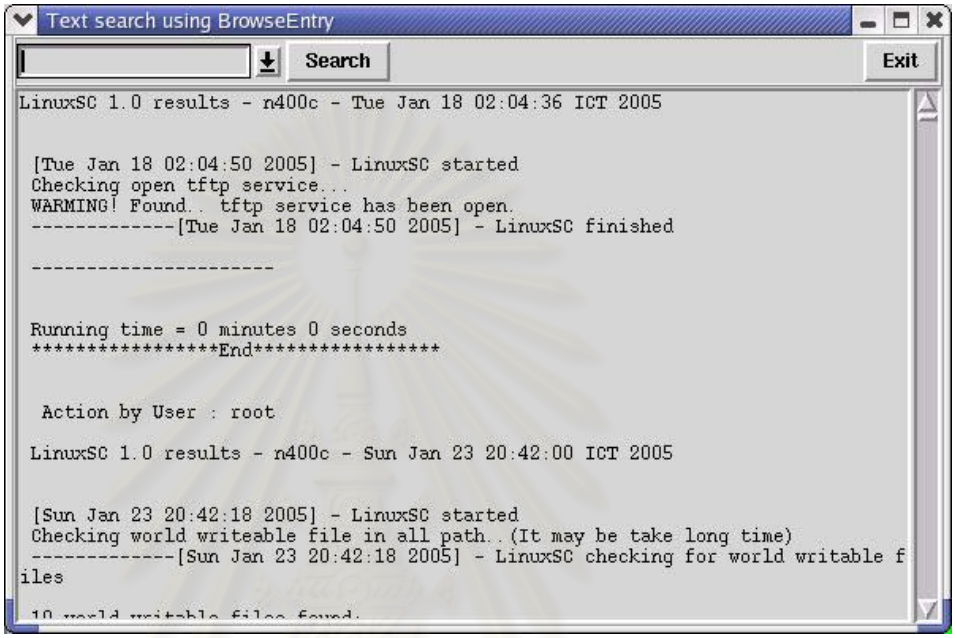
- ชุดปุ่มคำสั่ง Stop Daemon ซึ่งใช้ในการหยุดโปรแกรมตรวจสอบในแบบดีมอน



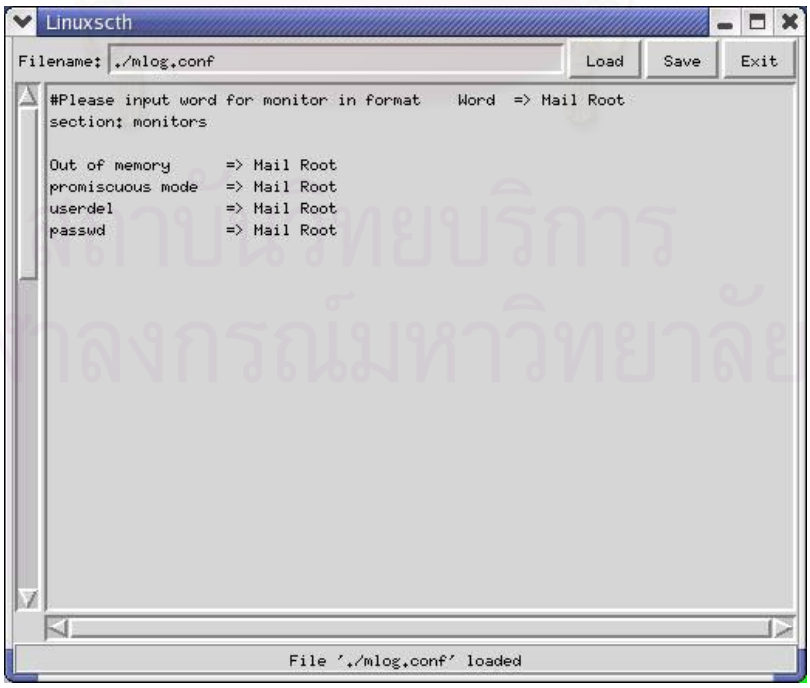
ผู้ใช้งานสามารถที่จะเลือกที่จะหยุดฟังก์ชันที่รันแบบดีมอน ได้ตามความต้องการ โดยการเลือกที่ปุ่มคำสั่งตามหัวข้อที่ต้องการจะหยุดการทำงาน จะปรากฏหน้าต่างแสดงการหยุดการทำงานของฟังก์ชันนั้น

ตัวอย่าง หน้าต่างการหยุดการทำงานแบบดีมอน ของฟังก์ชันการตรวจสอบภาวะการทำงานแบบไม่เลือก

- ปุ่มคำสั่ง History Log ซึ่งใช้ในการแสดงรายละเอียดผลการตรวจสอบของโปรแกรมตั้งแต่ทำการทำงานพร้อมทั้งสามารถระบุคำที่ต้องการค้นหาได้



- ปุ่มคำสั่ง Setup Monitor Log ใช้ในการกำหนดค่าของคำที่ใช้ในการตรวจสอบล็อกของระบบ

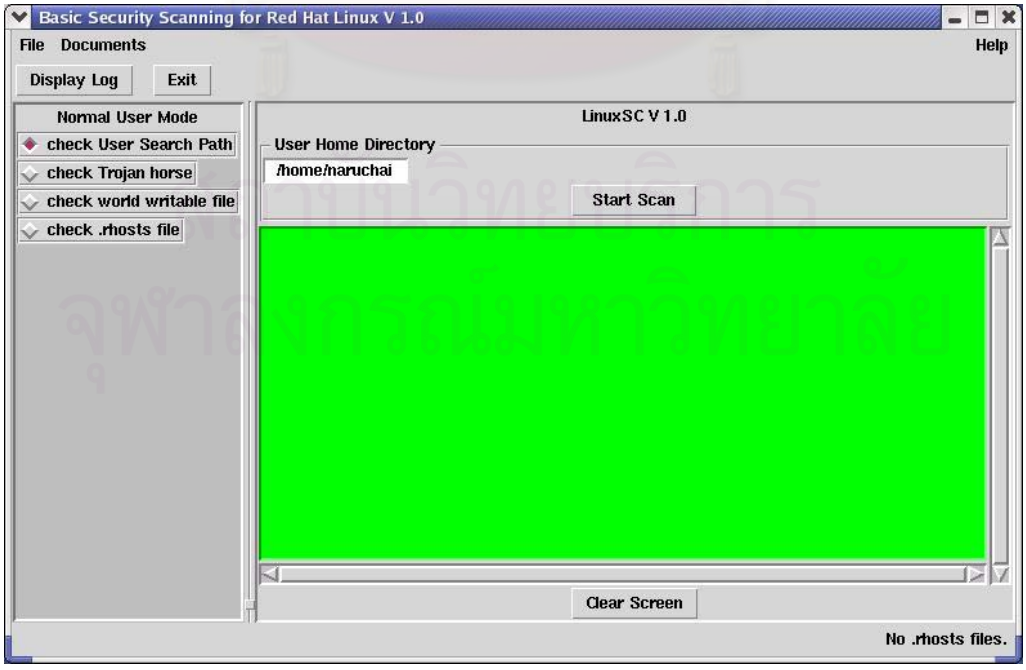


- ปุ่มคำสั่ง Exit ใช้ในการออกจากการทำงานของโปรแกรม

ส่วนของหน้าจอติดต่อกับผู้ใช้ ในโหมดผู้ใช้งาน

เมื่อรันโปรแกรมจะปรากฏหน้าต่างล็อกอินเข้าสู่โปรแกรมเพื่อตรวจสอบโหมดของผู้ใช้งาน กรณีที่ผู้ใช้งานไม่ได้ล็อกอินด้วยบัญชีผู้ใช้ รูท ระบบจะเข้าสู่ในโหมดของผู้ใช้งาน

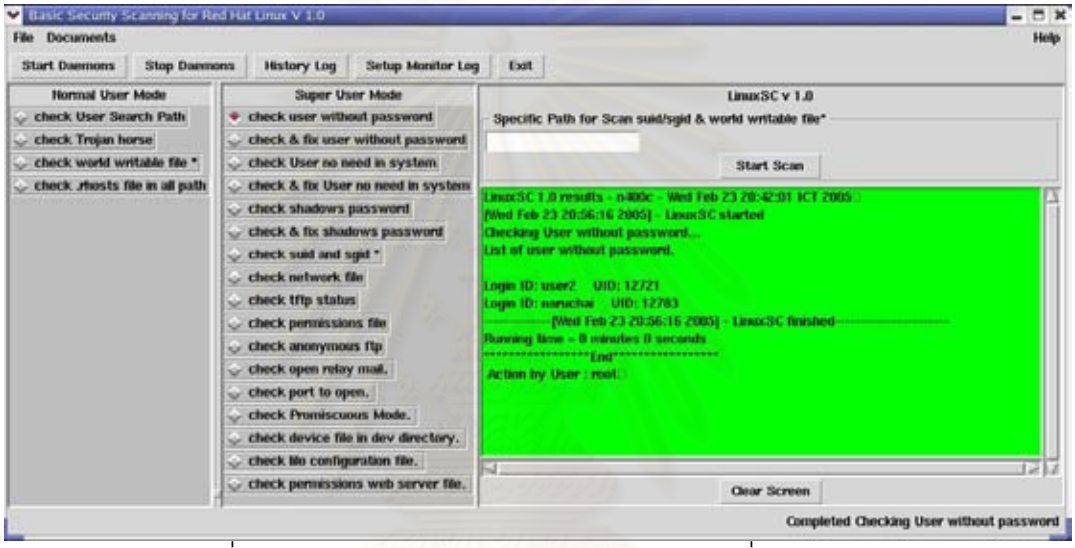
หน้าจอหลักของโปรแกรมในโหมดผู้ใช้งาน



ภาคผนวก ข

ฟังก์ชันการทำงานในโหมดผู้ดูแลระบบ

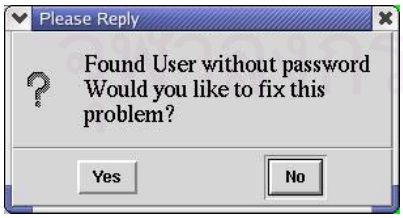
ฟังก์ชันการตรวจสอบผู้ใช้งานที่ไม่มีรหัสผ่าน

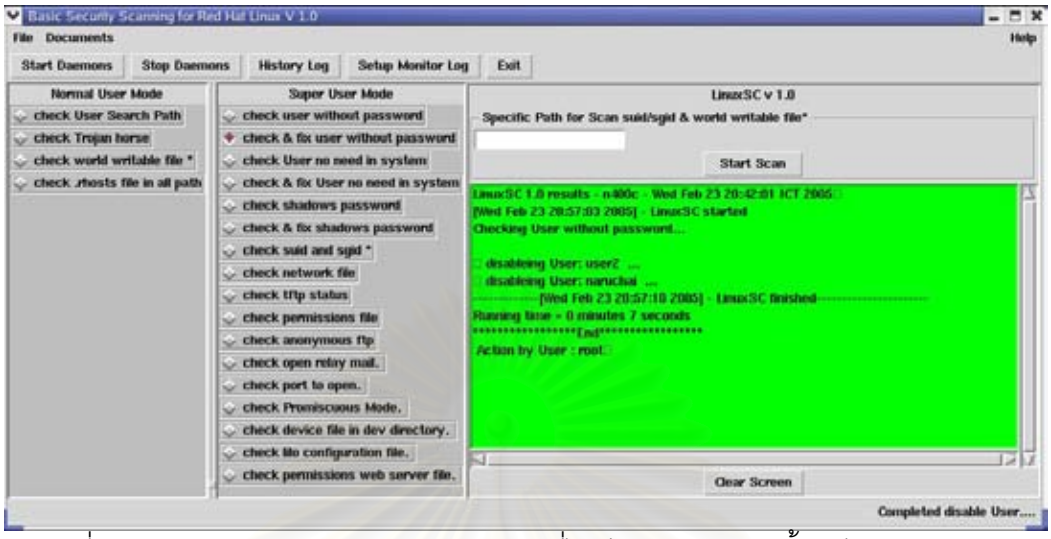


รูปที่ 1 ผลลัพธ์ของฟังก์ชันการตรวจสอบผู้ใช้งานที่ไม่มีรหัสผ่าน

ฟังก์ชันการตรวจสอบผู้ใช้งานที่ไม่มีรหัสผ่านพร้อมทั้งแก้ไขจุดบกพร่อง

ระบบทำการตรวจสอบพร้อมทั้งแสดงหน้าต่างเพื่อยืนยันการแก้ไขจุดบกพร่องโดยทำการยกเลิกการใช้งานของผู้ใช้ที่ไม่มีรหัสผ่านเพื่อความปลอดภัย

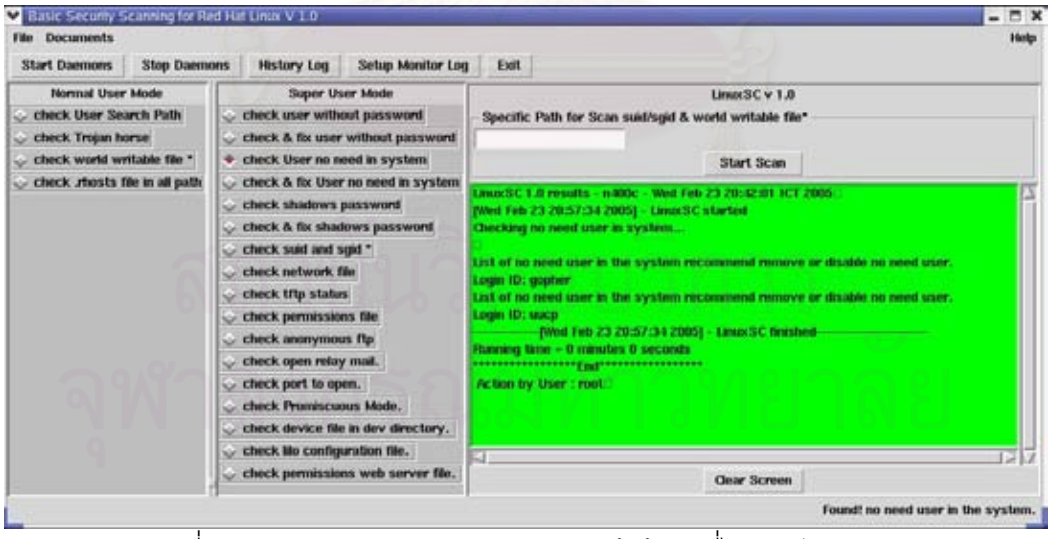




รูปที่ 2 ผลลัพธ์ฟังก์ชันการตรวจสอบผู้ใช้งานที่ไม่มีรหัสผ่านพร้อมทั้งแก้ไขจุดบกพร่อง

ฟังก์ชันการตรวจสอบผู้ใช้งานที่ไม่จำเป็นในระบบ

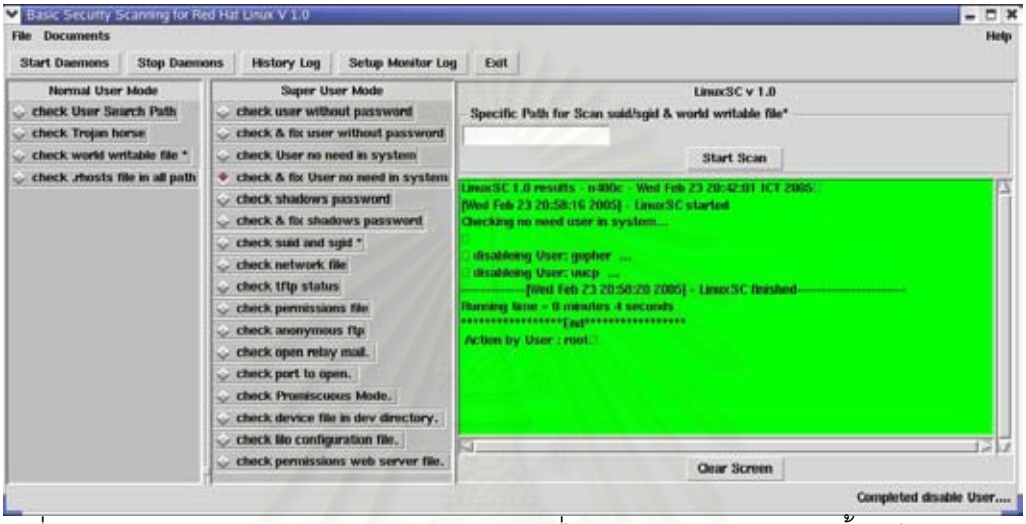
ระบบจะทำการตรวจสอบรายชื่อผู้ใช้งานที่ไม่จำเป็นในระบบพร้อมทั้งแสดงผลการตรวจสอบในที่นี้คือรายชื่อผู้ใช้งานที่ไม่จำเป็นในระบบ วันที่ เวลาและระยะเวลาที่ใช้ในการตรวจสอบ



รูปที่ 3 ผลลัพธ์ของฟังก์ชันการตรวจสอบผู้ใช้งานที่ไม่จำเป็นในระบบ

ฟังก์ชันการตรวจสอบผู้ใช้งานที่ไม่จำเป็นในระบบพร้อมทั้งแก้ไขจุดบกพร่อง

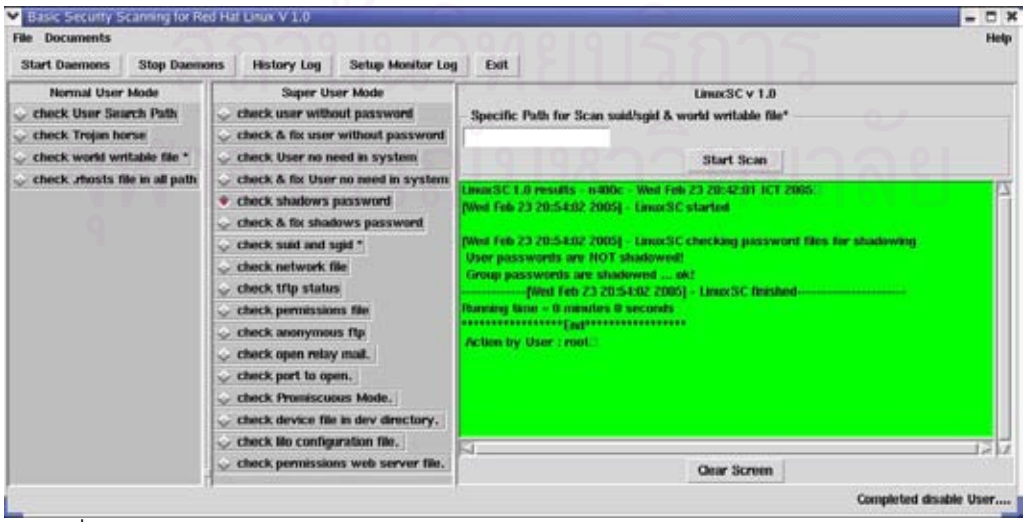
ระบบทำการตรวจสอบพร้อมทั้งแสดงหน้าต่างเพื่อยืนยันการแก้ไขจุดบกพร่องโดยทำการยกเลิกการใช้งานของผู้ใช้ที่ไม่จำเป็นในระบบเพื่อความปลอดภัย



รูปที่ 4 ผลลัพธ์ของฟังก์ชันการตรวจสอบผู้ใช้งานที่ไม่จำเป็นในระบบพร้อมทั้งแก้ไขจุดบกพร่อง

ฟังก์ชันตรวจสอบชาโดว์พาสเวิสด์ และ กรุ๊บพาสเวิสด์ ในระบบ

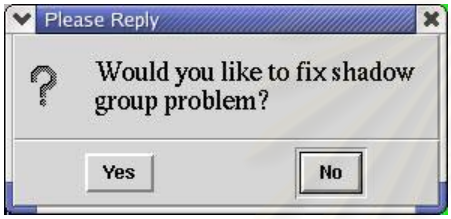
ระบบจะทำการตรวจสอบชาโดว์พาสเวิสด์ และ กรุ๊บพาสเวิสด์ในระบบพร้อมทั้งแสดงผลการตรวจสอบในที่นี้คือมีการทำ ชาโดว์พาสเวิสด์ และ กรุ๊บพาสเวิสด์ ในระบบหรือไม่ พร้อมทั้งวันที่ เวลาและระยะเวลาที่ใช้ในการตรวจสอบ

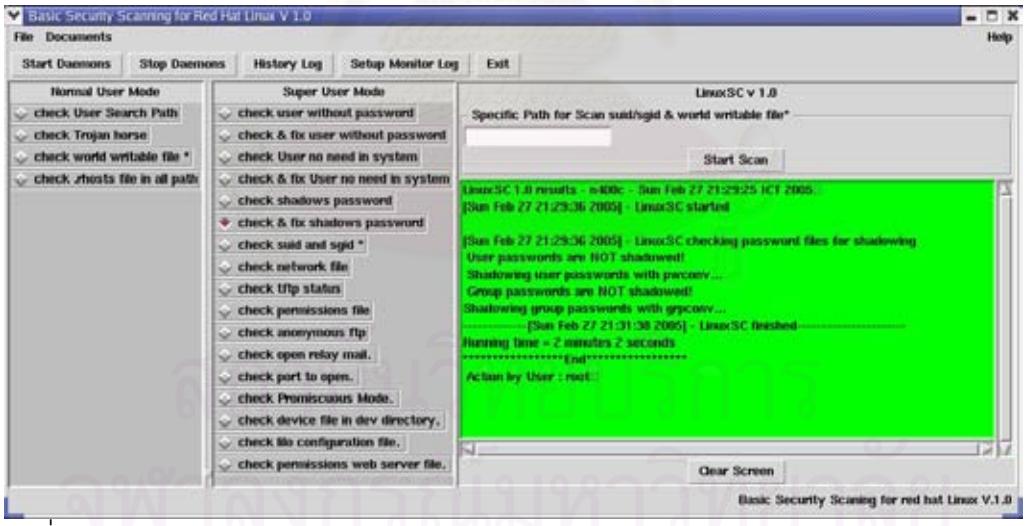


รูปที่ 5 ผลลัพธ์ของฟังก์ชันตรวจสอบ ชาโดว์พาสเวิสด์และ กรุ๊บพาสเวิสด์ ในระบบ

ฟังก์ชันการตรวจสอบ ชาโดว์พาสเวิสด์และ กรุ๊บพาสเวิสด์ในระบบพร้อมทั้งแก้ไขจุดบกพร่อง

โปรแกรมทำการตรวจสอบพร้อมทั้งแสดงหน้าต่างข้อความเพื่อยืนยันการแก้ไขจุดบกพร่องโดยทำ ชาโดว์พาสเวิสด์และ กรุ๊บพาสเวิสด์ให้กับระบบ

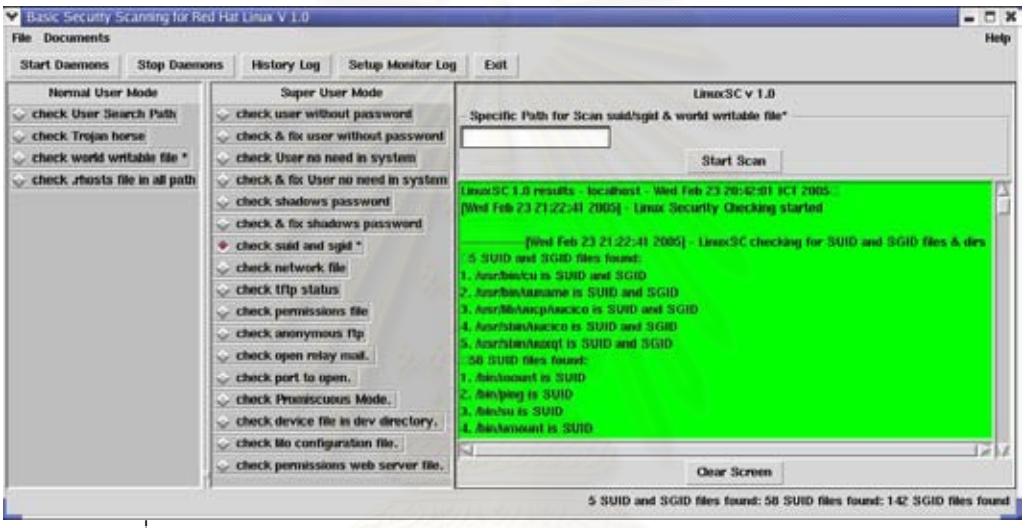




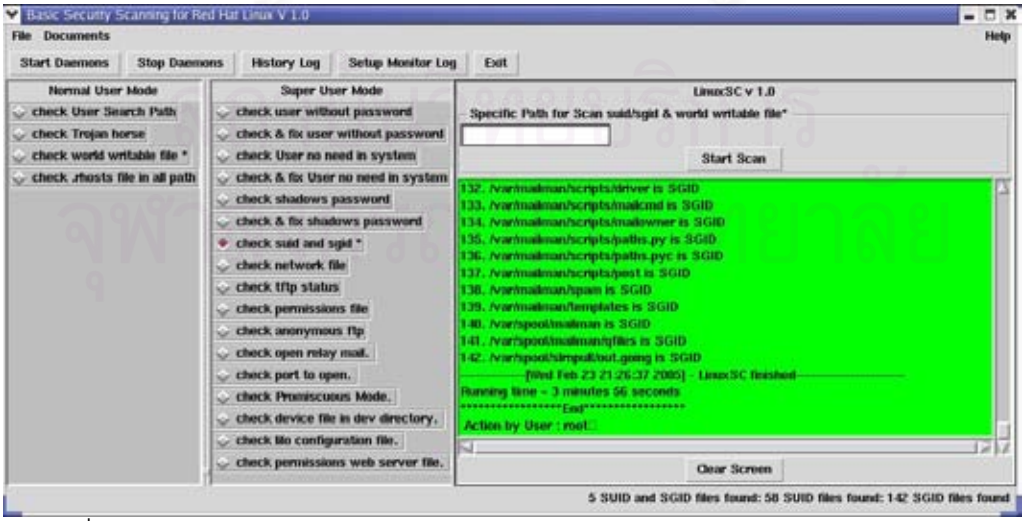
รูปที่ 6 ผลลัพธ์ของฟังก์ชันการตรวจสอบ ชาโดว์พาสเวิสด์และ กรุ๊บพาสเวิสด์ในระบบพร้อมทั้งแก้ไขจุดบกพร่อง

ฟังก์ชันการตรวจสอบแฟ้ม SUID และแฟ้ม SGID ในระบบ

ระบบจะทำการตรวจสอบแฟ้ม SUID และ แฟ้ม SGID ในระบบพร้อมทั้งแสดงผลการตรวจสอบในที่นี้คือแสดงจำนวนแฟ้มที่เป็น SUID แฟ้ม SGID และแฟ้มที่เป็นทั้ง SUID และ SGID ในระบบ พร้อมทั้งแสดง วันที่ เวลาและระยะเวลาที่ใช้ในการตรวจสอบ นอกจากนี้ผู้ใช้งานยังสามารถกำหนดเส้นทางที่ใช้ในการค้นหาแฟ้ม SUID และแฟ้ม SGID ได้ในช่องระบุเส้นทางในการค้นหา



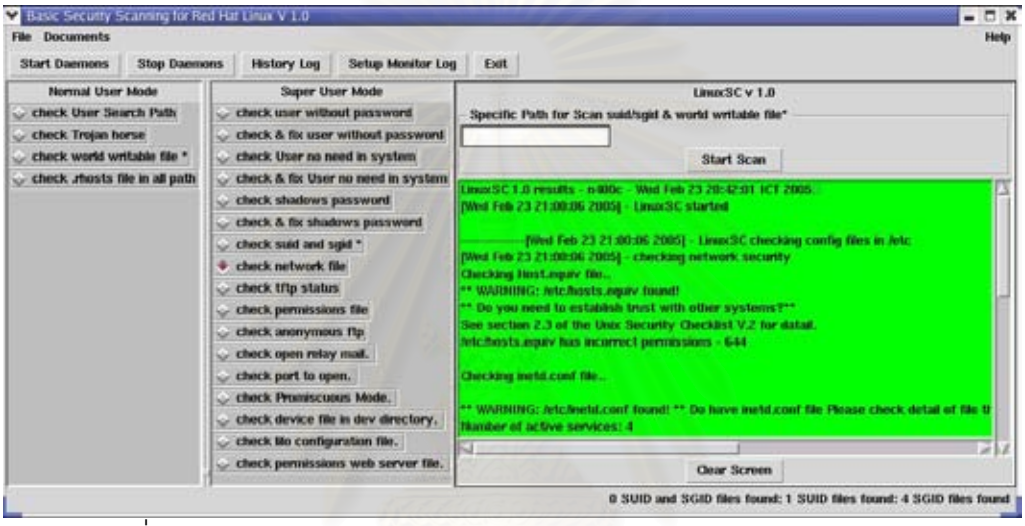
รูปที่ 7 ผลลัพธ์ของฟังก์ชันการตรวจสอบแฟ้ม SUID และแฟ้ม SGID ในระบบ



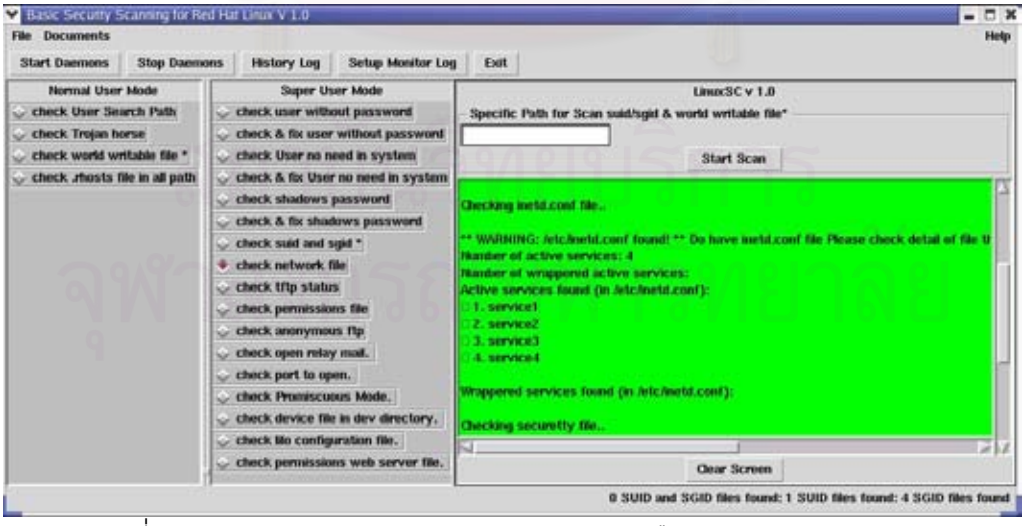
รูปที่ 8 ผลลัพธ์ของฟังก์ชันการตรวจสอบแฟ้ม SUID และแฟ้ม SGID ในระบบ (ต่อ)

ฟังก์ชันการตรวจสอบแฟ้มเครือข่ายในระบบ

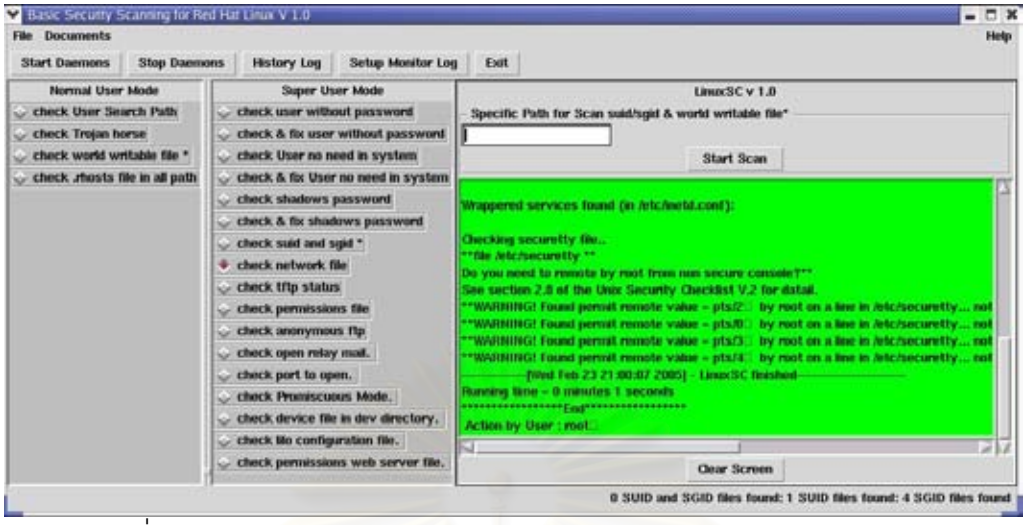
ระบบจะทำการตรวจสอบแฟ้มเครือข่ายในระบบได้แก่ แฟ้ม /etc/inetd.conf แฟ้ม /etc/hosts.equiv แฟ้ม /etc/services และแฟ้ม /etc/securetty พร้อมทั้งแสดงผลการตรวจสอบในที่นี้คือเนื้อหาที่ไม่เหมาะสมภายในแฟ้ม พร้อมทั้งแสดง วันที่ เวลาและระยะเวลาที่ใช้ในการตรวจสอบ



รูปที่ 9 ผลลัพธ์ของฟังก์ชันการตรวจสอบแฟ้มเครือข่ายในระบบ



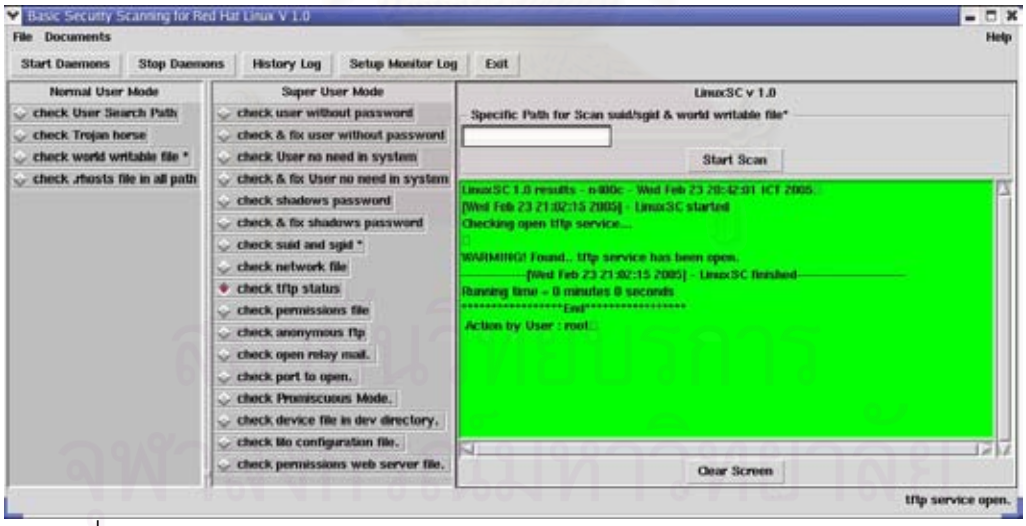
รูปที่ 10 ผลลัพธ์ของฟังก์ชันการตรวจสอบแฟ้มเครือข่ายในระบบ (ต่อ)



รูปที่ 11 ผลลัพธ์ของฟังก์ชันการตรวจสอบแฟ้มเครือข่ายในระบบ (ต่อ)

ฟังก์ชันการตรวจสอบสถานการณ์บริการทีเอฟทีพีเซิร์ฟเวอร์

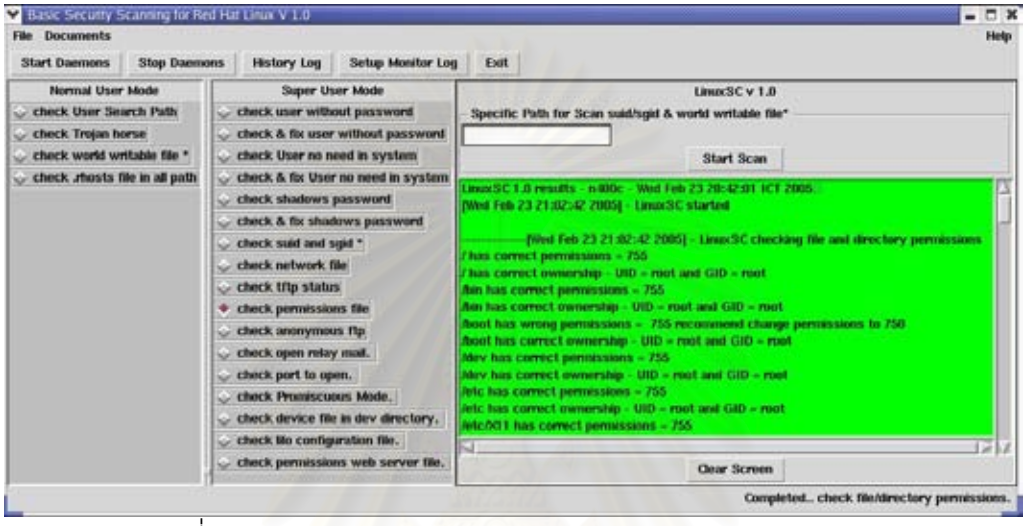
ระบบจะทำการตรวจสอบสถานะของการบริการทีเอฟทีพีและแสดงผลการตรวจสอบ พร้อมทั้งวันที่ เวลาและระยะเวลาที่ใช้ในการตรวจสอบ



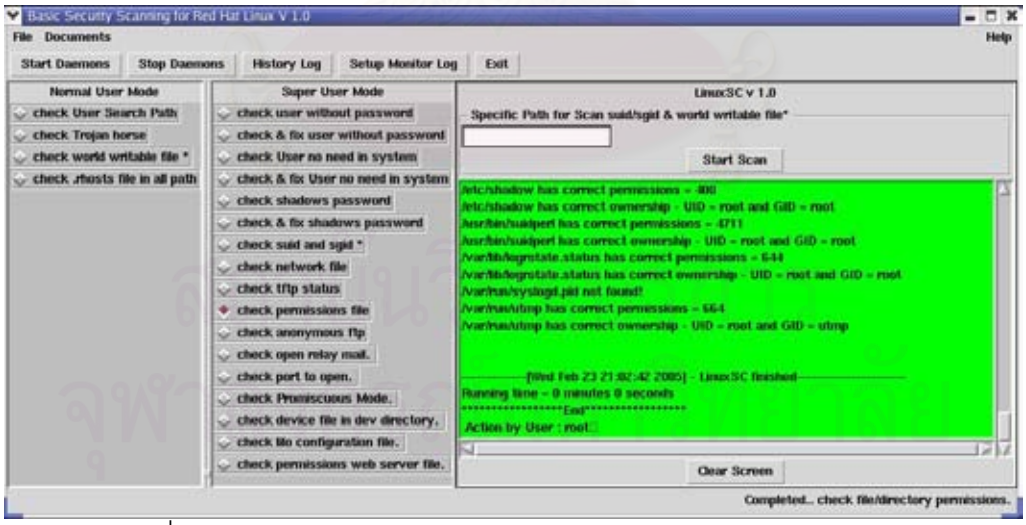
รูปที่ 12 ผลลัพธ์ของฟังก์ชันการตรวจสอบสถานะการบริการ ทีเอฟทีพีเซิร์ฟเวอร์

ฟังก์ชันการตรวจสอบบิตอนุญาตของแฟ้ม

ระบบจะทำการตรวจสอบพร้อมแสดงผลการตรวจสอบบิตอนุญาตของแฟ้ม เจ้าของแฟ้ม และกลุ่มของเจ้าของแฟ้ม พร้อมทั้งแนะนำค่าที่ถูกต้องในกรณีที่แฟ้มที่ตรวจสอบมีค่าบิตอนุญาตที่ไม่ถูกต้อง พร้อมทั้งแสดงวันที่ เวลาและระยะเวลาที่ใช้ในการตรวจสอบ



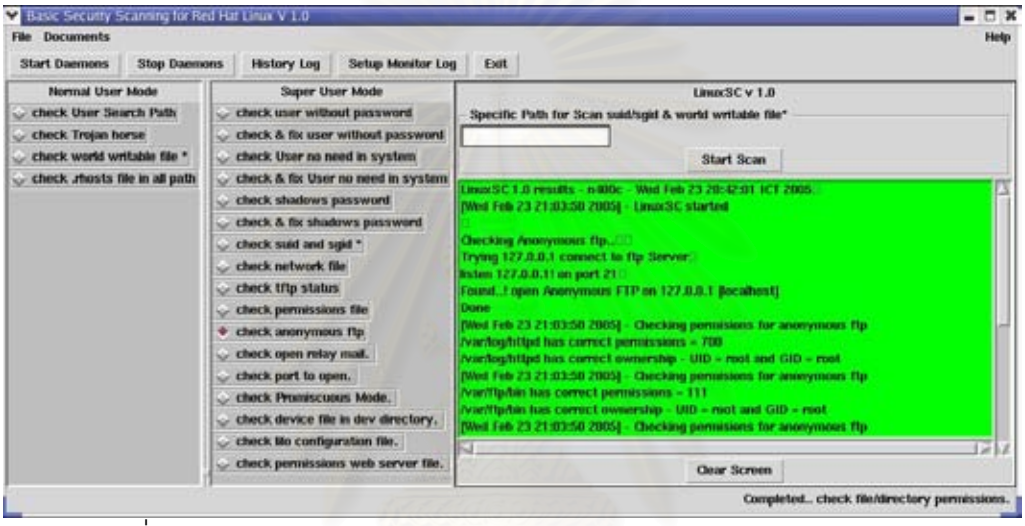
รูปที่ 13 ผลลัพธ์ของฟังก์ชันการตรวจสอบบิตอนุญาตของแฟ้ม



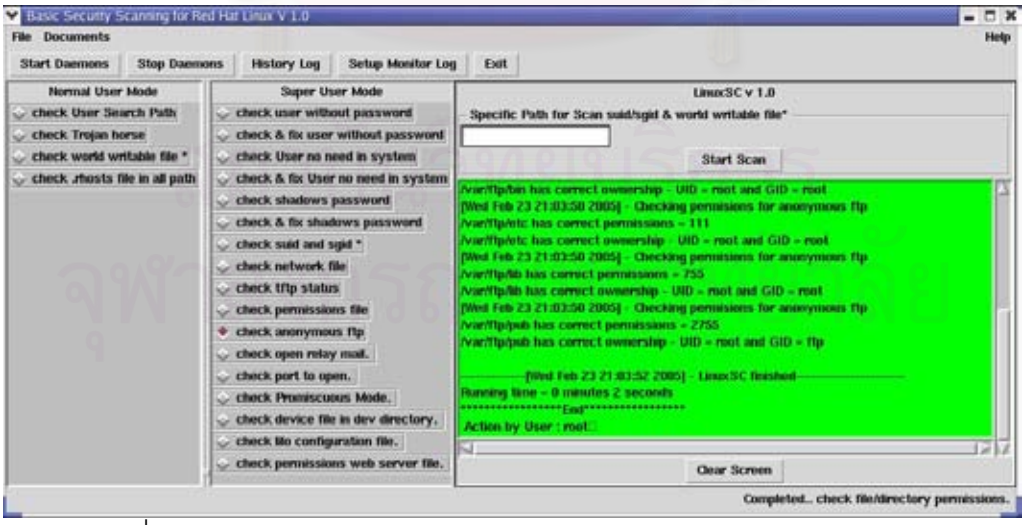
รูปที่ 14 ผลลัพธ์ของฟังก์ชันการตรวจสอบบิตอนุญาตของแฟ้ม (ต่อ)

ฟังก์ชันการตรวจสอบการเอฟทีพีแบบนิรนาม

ระบบจะทำการตรวจสอบพร้อมแสดงผลการตรวจสอบ สถานะการให้บริการของการเอฟทีพีแบบนิรนาม และบิตอนุญาตของแฟ้ม ที่เกี่ยวข้องกับการทำเอฟทีพีแบบนิรนาม เจ้าของแฟ้ม และกลุ่มของเจ้าของแฟ้ม พร้อมทั้งแนะนำค่าที่ถูกต้องในกรณีที่แฟ้มที่เกี่ยวข้องกับการทำ เอฟทีพีแบบนิรนามที่ตรวจสอบมีค่าบิตอนุญาตที่ไม่ถูกต้อง พร้อมทั้งแสดงวันที่ เวลาและระยะเวลาที่ใช้ในการตรวจสอบ



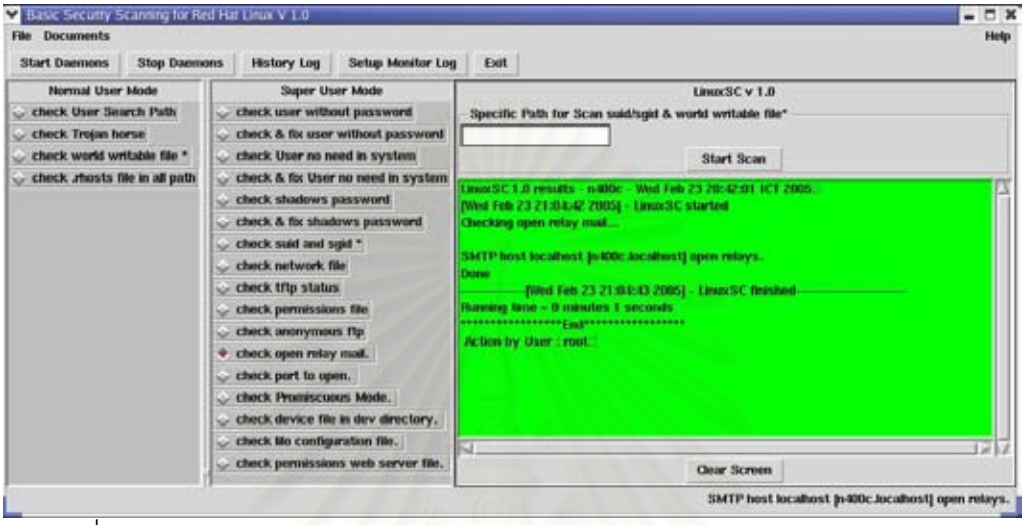
รูปที่ 15 ผลลัพธ์ฟังก์ชันการตรวจสอบ การทำเอฟทีพีแบบนิรนาม



รูปที่ 16 ผลลัพธ์ฟังก์ชันการตรวจสอบ การทำเอฟทีพีแบบนิรนาม (ต่อ)

ฟังก์ชันในการตรวจสอบสถานการณ์เปิด การรีเลย์ ในระบบเมล์เซิร์ฟเวอร์

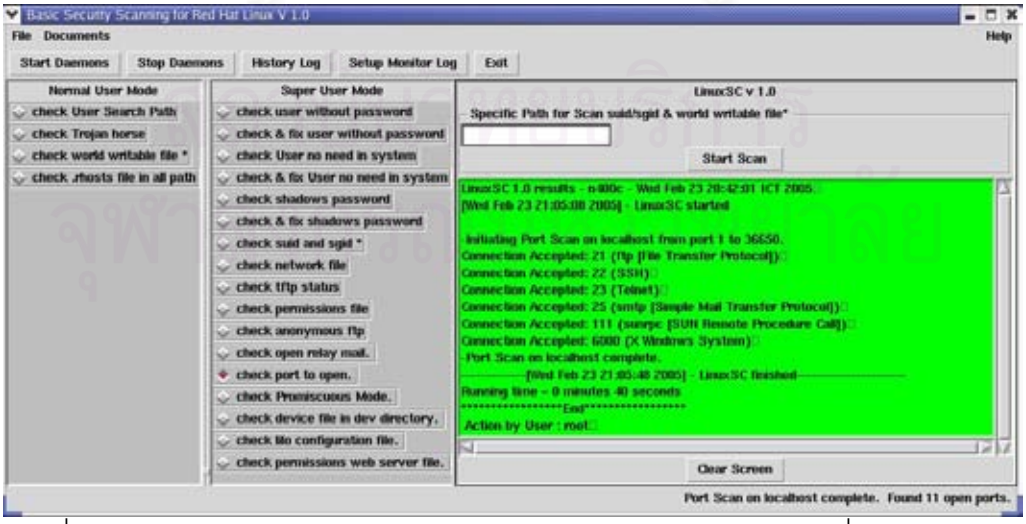
ระบบจะทำการตรวจสอบพร้อมแสดงผลการตรวจสอบ สถานะการเปิดการรีเลย์ ในระบบเมล์เซิร์ฟเวอร์ พร้อมทั้งแสดงวันที่ เวลาและระยะเวลาที่ใช้ในการตรวจสอบ



รูปที่ 17 ผลลัพธ์ของฟังก์ชันในการตรวจสอบสถานะการรีเลย์ ในระบบเมล์เซิร์ฟเวอร์

ฟังก์ชันในการตรวจสอบสถานะการเปิดช่องทางการสื่อสารในระบบ

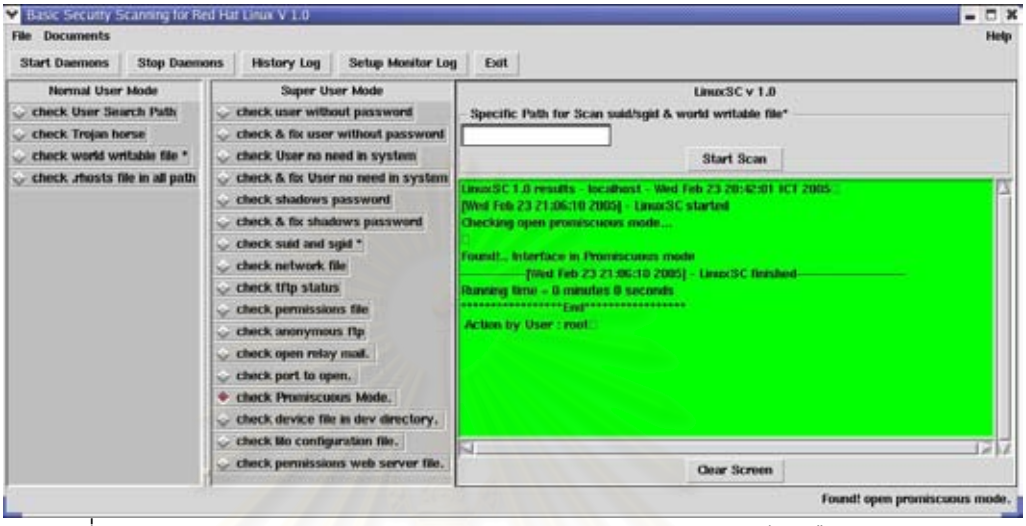
ระบบจะทำการตรวจสอบพร้อมแสดงผลการตรวจสอบ ในที่นี้คือช่องทางการสื่อสารในระบบ พร้อมทั้งแสดงวันที่ เวลาและระยะเวลาที่ใช้ในการตรวจสอบ



รูปที่ 18 ผลลัพธ์ของฟังก์ชันในการตรวจสอบสถานะการเปิดช่องทางการสื่อสารในระบบ

ฟังก์ชันการตรวจสอบภาวะ การทำงานแบบไม่เลือกในระบบ

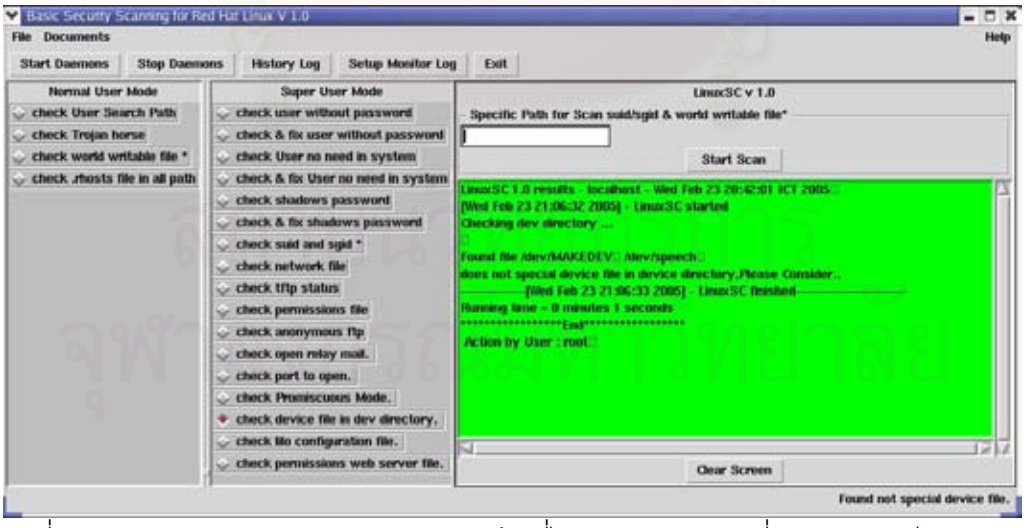
ระบบจะทำการตรวจสอบพร้อมแสดงผลการตรวจสอบภาวะ การทำงานแบบไม่เลือกของอินเตอร์เฟส ในระบบ พร้อมทั้งแสดงวันที่ เวลาและระยะเวลาที่ใช้ในการตรวจสอบ



รูปที่ 19 ผลลัพธ์ของฟังก์ชันการตรวจสอบภาวะ การทำงานแบบไม่เลือกในระบบ

ฟังก์ชันการตรวจสอบแฟ้มที่ไม่ใช่ชนิดแฟ้มอุปกรณ์ว่าปรากฏอยู่ในไดเรกทอรี /dev หรือไม่

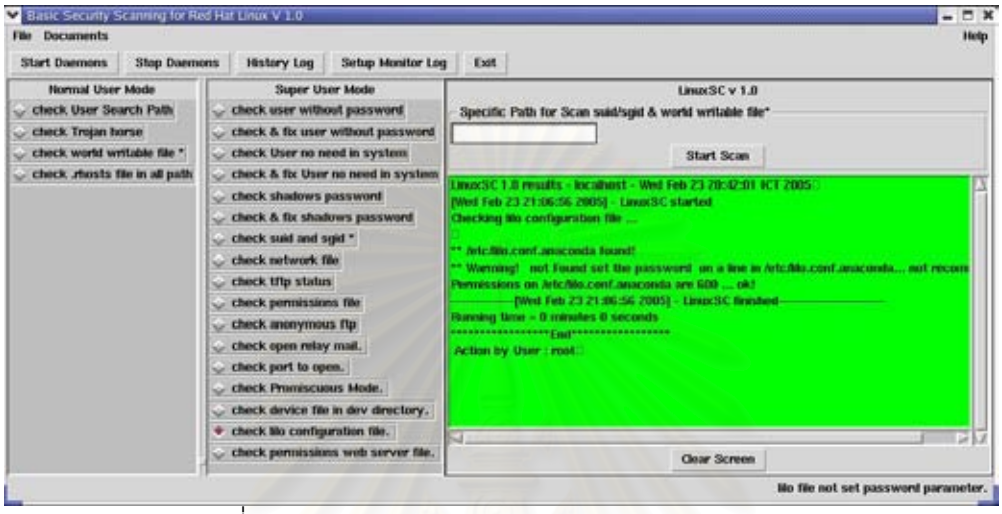
ระบบจะทำการตรวจสอบพร้อมแสดงผลการตรวจสอบ แฟ้มที่ไม่ใช่ชนิดอุปกรณ์ ในไดเรกทอรี /dev ในระบบพร้อมทั้งแสดงวันที่ เวลาและระยะเวลาที่ใช้ในการตรวจสอบ



รูปที่ 20 ผลลัพธ์ของฟังก์ชันการตรวจสอบแฟ้มที่ไม่ใช่ชนิดอุปกรณ์ที่ปรากฏอยู่ในไดเรกทอรี /dev

ฟังก์ชันการตรวจสอบแฟ้ม lilo.conf

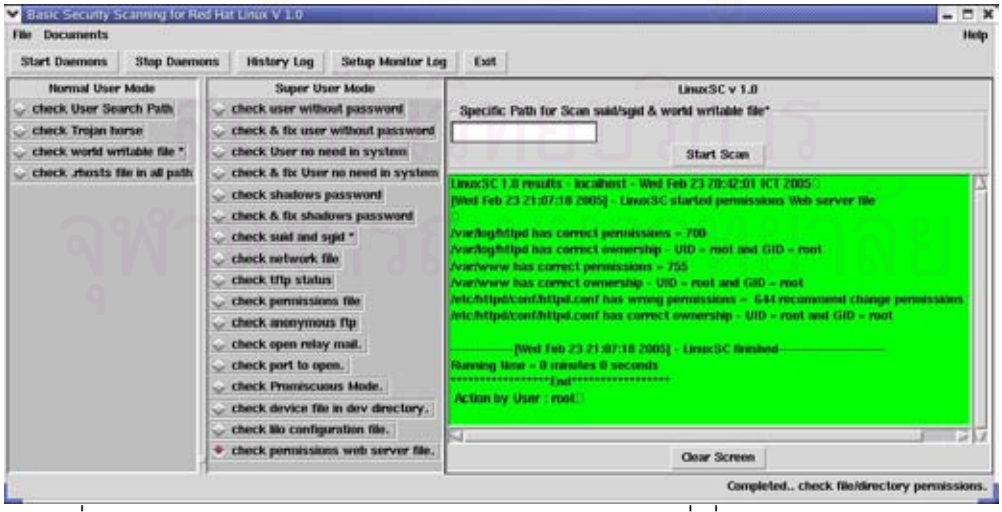
ระบบจะทำการตรวจสอบพร้อมแสดงผลการตรวจสอบ ค่าพารามิเตอร์ที่เหมาะสมพร้อมทั้งแสดงวันที่ เวลาและระยะเวลาที่ใช้ในการตรวจสอบ



รูปที่ 21 ผลลัพธ์ของฟังก์ชันการตรวจสอบแฟ้ม lilo.conf

ฟังก์ชันการตรวจสอบบิตอนุญาตของแฟ้ม และไดเรกทอรีที่เกี่ยวข้องกับเว็บเซิร์ฟเวอร์

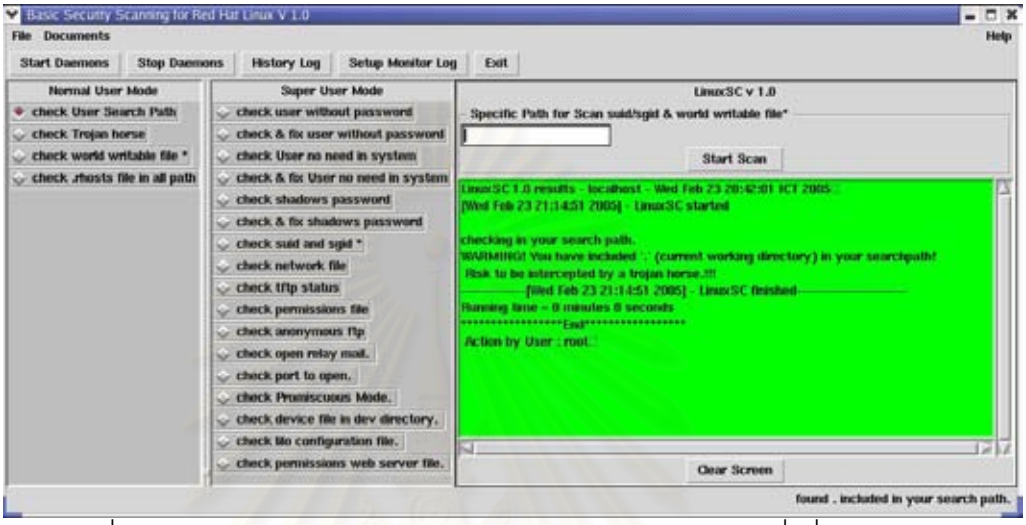
ระบบจะทำการตรวจสอบพร้อมแสดงผลการตรวจสอบบิตอนุญาตของแฟ้ม เจ้าของแฟ้ม และกลุ่มของเจ้าของแฟ้ม พร้อมทั้งแนะนำค่าที่ถูกต้องในกรณีที่แฟ้มที่ตรวจสอบมีค่าบิตอนุญาตที่ไม่ถูกต้อง พร้อมทั้งแสดงวันที่ เวลาและระยะเวลาที่ใช้ในการตรวจสอบ



รูปที่ 22 ผลของฟังก์ชันการตรวจสอบแฟ้มและไดเรกทอรีที่เกี่ยวข้องกับเว็บเซิร์ฟเวอร์

ฟังก์ชันการตรวจสอบตรวจสอบการกำหนดการค้นหาคำสั่งที่เป็นอันตราย

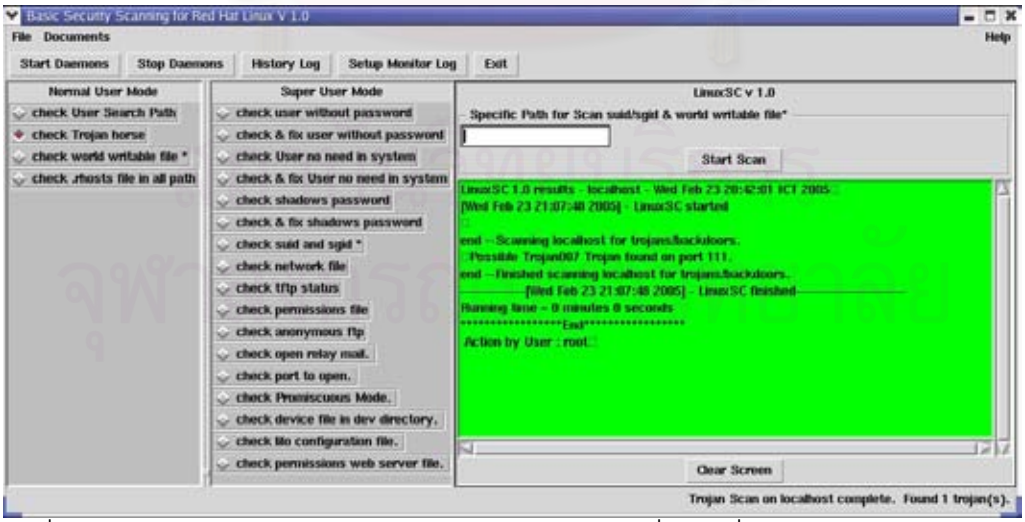
ระบบจะทำการตรวจสอบพร้อมแสดงผลการตรวจสอบ การค้นหาคำสั่งที่เป็นอันตรายของผู้ใช้งานปัจจุบัน พร้อมทั้งแสดงวันที่ เวลาและระยะเวลาที่ใช้ในการตรวจสอบ



รูปที่ 23 ผลลัพธ์ฟังก์ชันการตรวจสอบการกำหนดการค้นหาคำสั่งที่เป็นอันตราย

ฟังก์ชันในการตรวจสอบช่องทางการสื่อสารที่ม้าโทรจันมักมีการใช้งาน

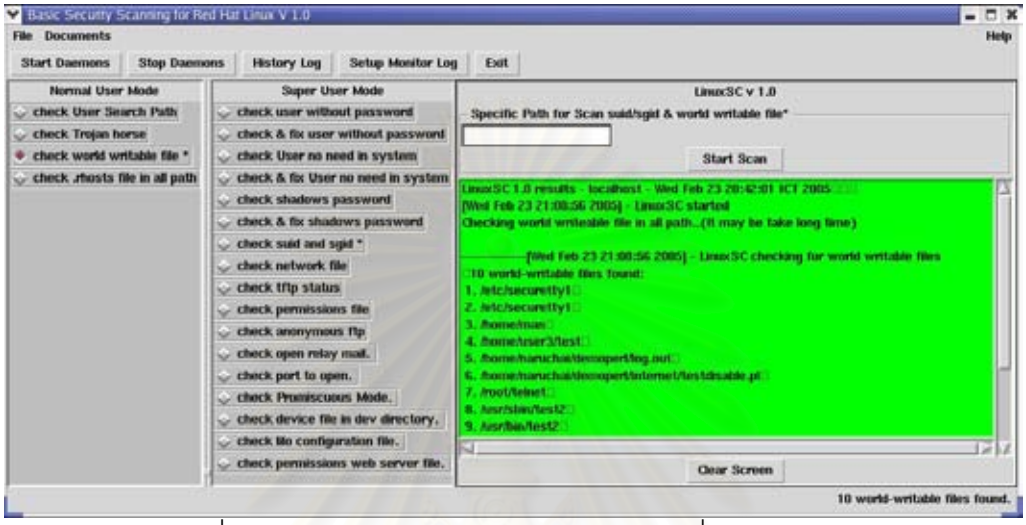
ระบบจะทำการตรวจสอบพร้อมแสดงผลการตรวจสอบ ช่องทางการสื่อสาร ที่เป็นอันตรายที่ม้าโทรจันใช้ในการติดต่อสื่อสาร พร้อมทั้งแสดงวันที่ เวลาและระยะเวลาที่ใช้ในการตรวจสอบ



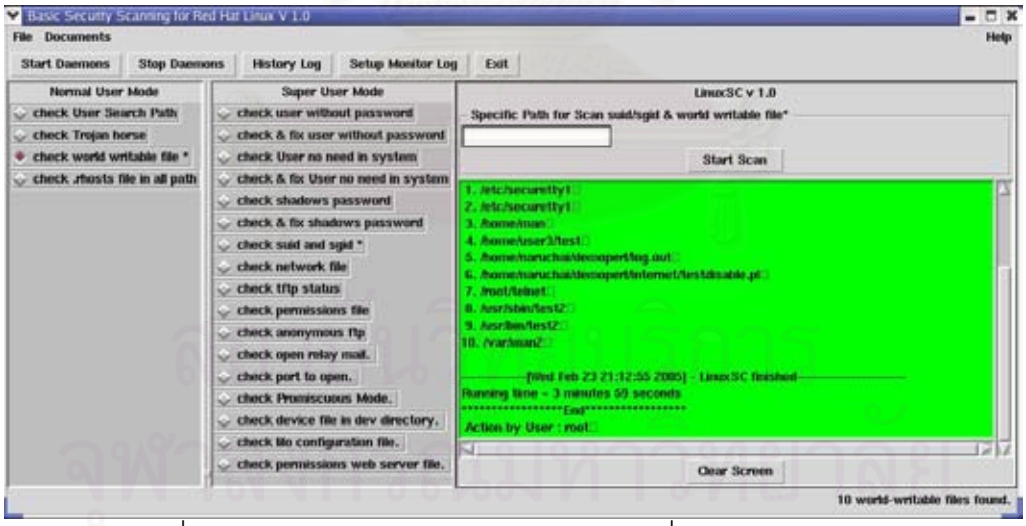
รูปที่ 24 ผลลัพธ์ของฟังก์ชันในการตรวจสอบช่องทางการสื่อสารที่ม้าโทรจันมักมีการใช้งาน

ฟังก์ชันในการตรวจสอบแฟ้มที่มีการเปิดบิตอนุญาตทั้งหมด

ระบบจะทำการตรวจสอบพร้อมแสดงผลการตรวจสอบ จำนวนแฟ้มที่เปิดสิทธิเต็ม ทุกไดเรกทอรีในระบบพร้อมทั้งแสดงวันที่ เวลาและระยะเวลาที่ใช้ในการตรวจสอบ



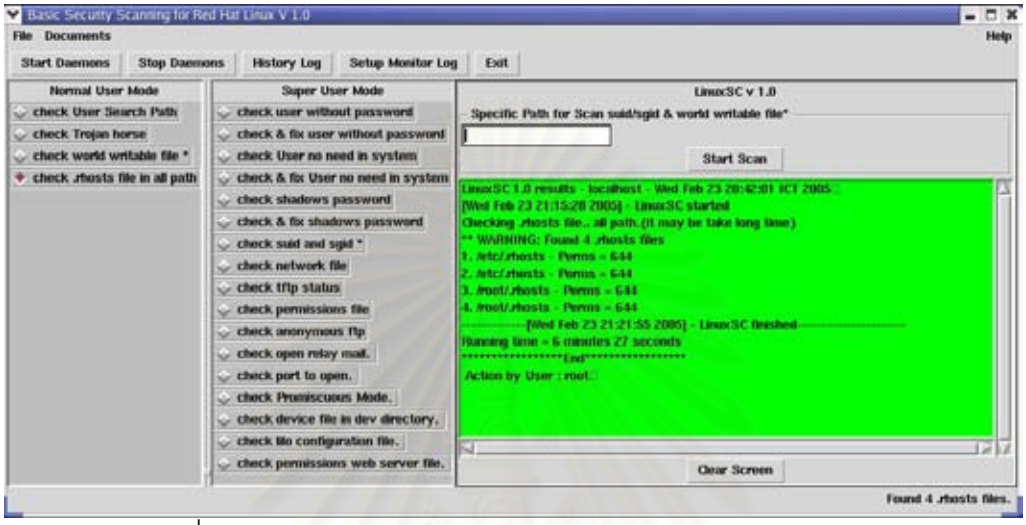
รูปที่ 25 ผลลัพธ์ฟังก์ชันในการตรวจสอบแฟ้มที่เปิดสิทธิเต็ม



รูปที่ 26 ผลลัพธ์ฟังก์ชันในการตรวจสอบแฟ้ม ที่เปิดสิทธิเต็ม (ต่อ)

ฟังก์ชันการตรวจสอบแฟ้ม .rhost ในระบบ

ระบบจะทำการตรวจสอบพร้อมแสดงผลการตรวจสอบ จำนวนแฟ้ม .rhost ทุกไดเรกทอรีในระบบพร้อมทั้งแสดงวันที่ เวลาและระยะเวลาที่ใช้ในการตรวจสอบ



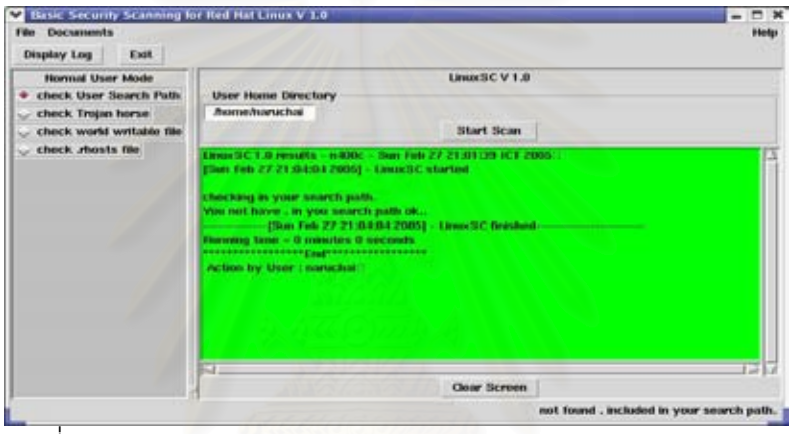
รูปที่ 27 ผลลัพธ์ของฟังก์ชันการตรวจสอบแฟ้ม .rhost ในระบบ

ภาคผนวก ค

ฟังก์ชันการทำงานในโหมดผู้ใช้งาน

ฟังก์ชันในการตรวจสอบการกำหนดการค้นหาคำสั่งที่เป็นอันตรายของผู้ใช้งาน

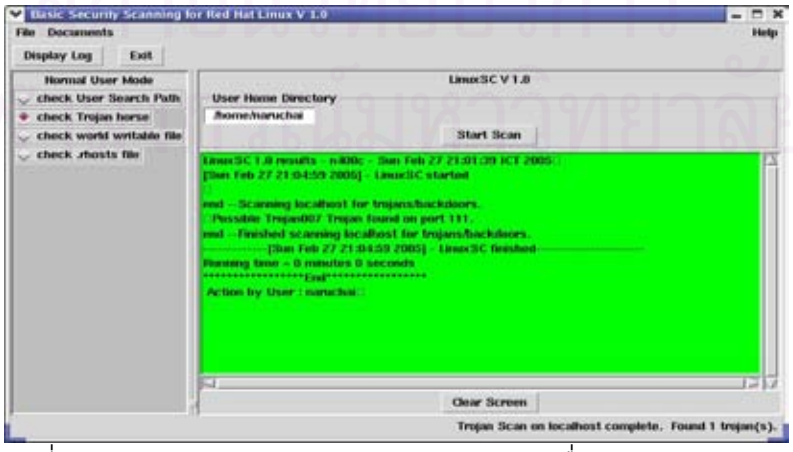
ระบบจะทำการตรวจสอบพร้อมแสดงผลการตรวจสอบการค้นหาคำสั่งที่เป็นอันตรายของผู้ใช้งาน พร้อมทั้งแสดงวันที่ เวลาและระยะเวลาที่ใช้ในการตรวจสอบ



รูปที่ 28 ผลลัพธ์ฟังก์ชันในการตรวจสอบ search path ของผู้ใช้งาน

ฟังก์ชันในการตรวจสอบ ช่องทางสื่อสารของม้าโทรจัน

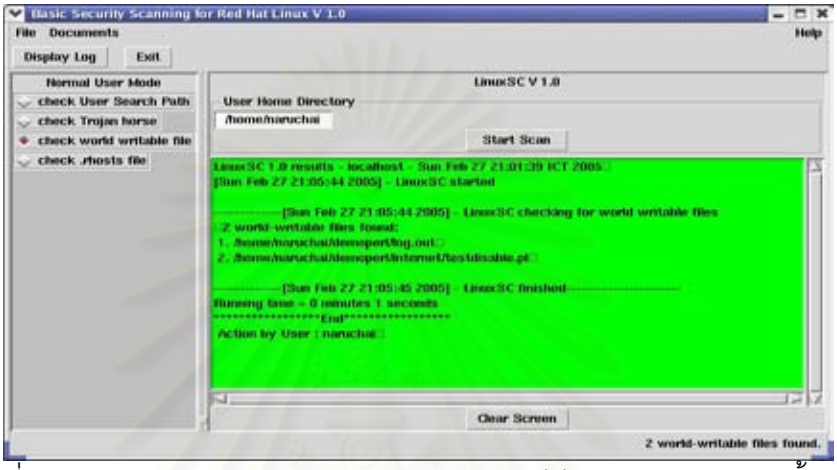
ระบบจะทำการตรวจสอบพร้อมแสดงผลการตรวจสอบ ช่องทางสื่อสารของม้าโทรจัน พร้อมทั้งแสดงวันที่ เวลาและระยะเวลาที่ใช้ในการตรวจสอบ



รูปที่ 29 ผลของฟังก์ชันในการตรวจสอบ ช่องทางสื่อสารของม้าโทรจัน

ฟังก์ชันในการตรวจสอบแฟ้มที่มีการเปิดบิตอนุญาตทั้งหมด

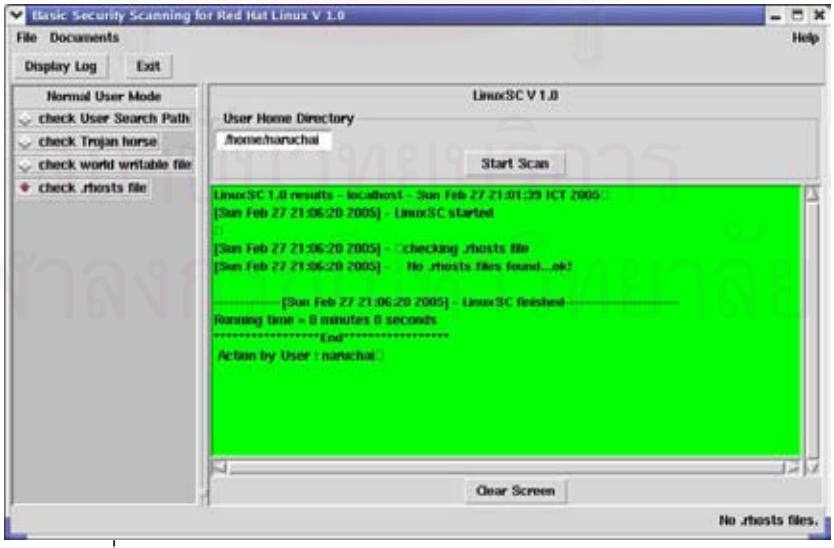
ระบบจะทำการตรวจสอบพร้อมแสดงผลการตรวจสอบ จำนวนแฟ้มที่มีการเปิดบิตอนุญาตทั้งหมด ในไดเรกทอรีบ้านของผู้ใช้งาน พร้อมทั้งแสดงวันที่ เวลาและระยะเวลาที่ใช้ในการตรวจสอบ



รูปที่ 30 ผลลัพธ์ของฟังก์ชันในการตรวจสอบแฟ้มทีมีการเปิดบิตอนุญาตทั้งหมด

ฟังก์ชันการตรวจสอบแฟ้ม .rhost ในระบบ

ระบบจะทำการตรวจสอบพร้อมแสดงผลการตรวจสอบ จำนวนแฟ้ม .rhost ในไดเรกทอรีบ้านของผู้ใช้งาน พร้อมทั้งแสดงวันที่ เวลาและระยะเวลาที่ใช้ในการตรวจสอบ



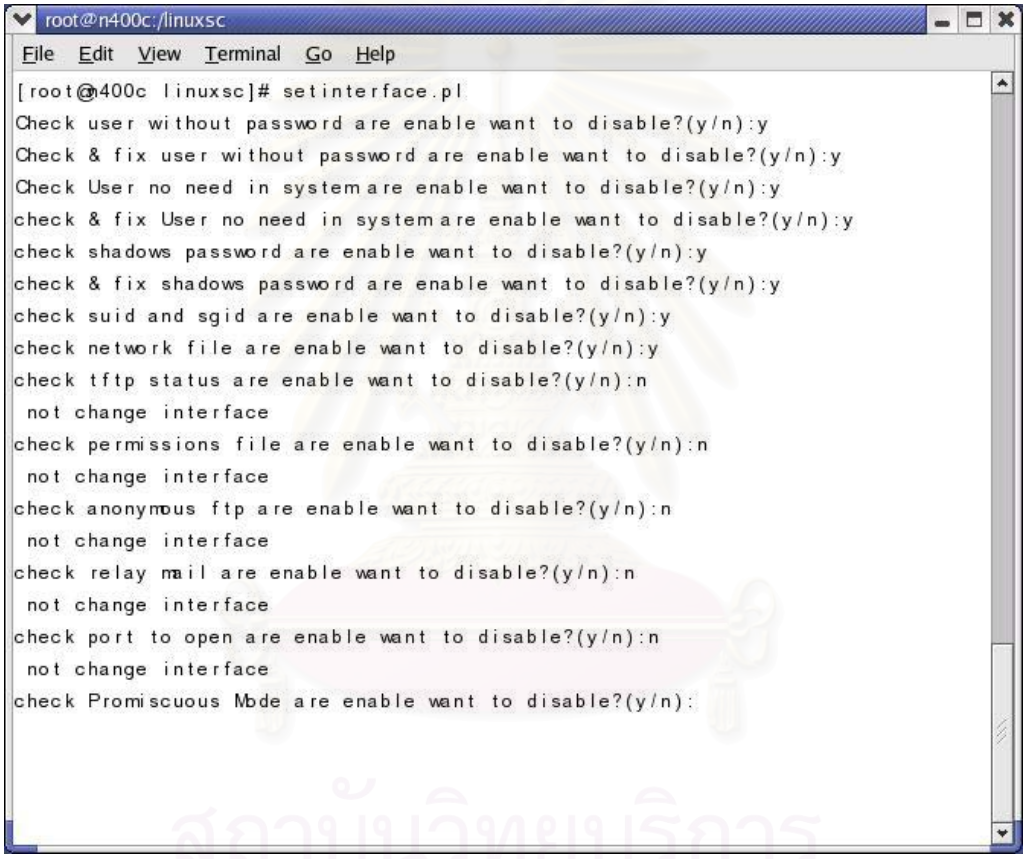
รูปที่ 31 ผลลัพธ์ฟังก์ชันการตรวจสอบแฟ้ม .rhost ในระบบ

ภาคผนวก ง

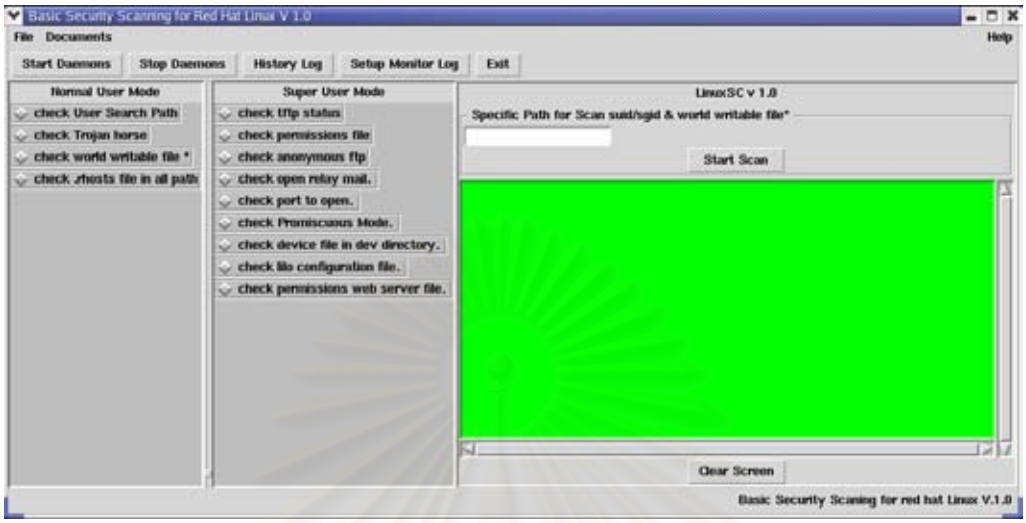
การกำหนดค่าคอนฟิกระบบและการแสดงผลในรูปแบบภาษาไทย

การปรับเปลี่ยนฟังก์ชันการตรวจสอบ

ทำการรันสคริปต์ setinterface.pl จะปรากฏเป็นลักษณะคำถามเพื่อเลือกว่าต้องการให้ฟังก์ชันในการตรวจสอบใดปรากฏในโปรแกรม

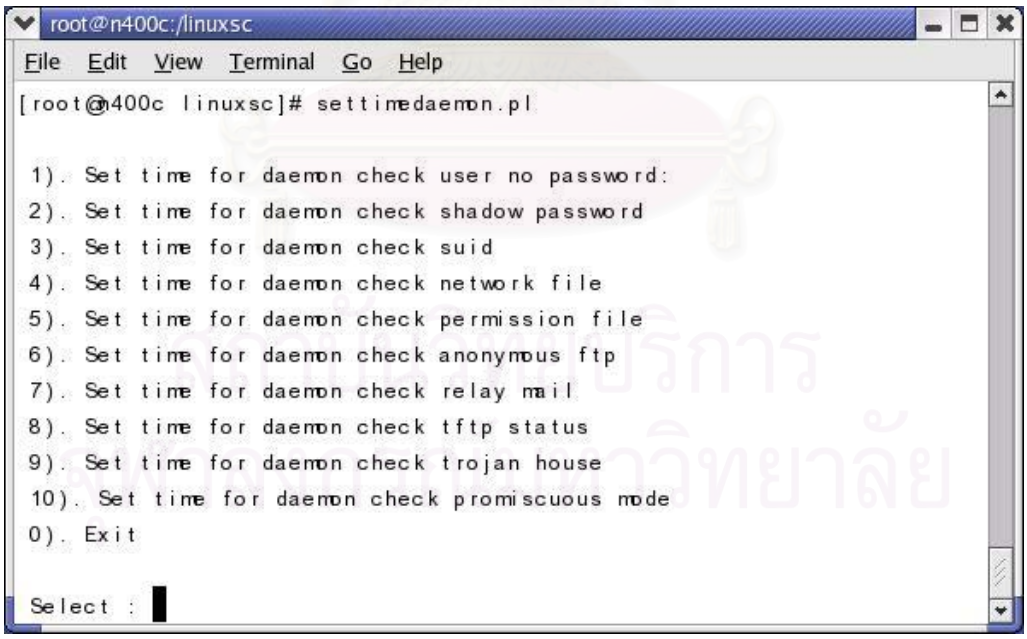


เมื่อเลือกฟังก์ชันที่ต้องการเรียบร้อย หลังจากเรียกโปรแกรมอีกครั้งจะปรากฏเฉพาะฟังก์ชันที่ทำการเลือกในโปรแกรม

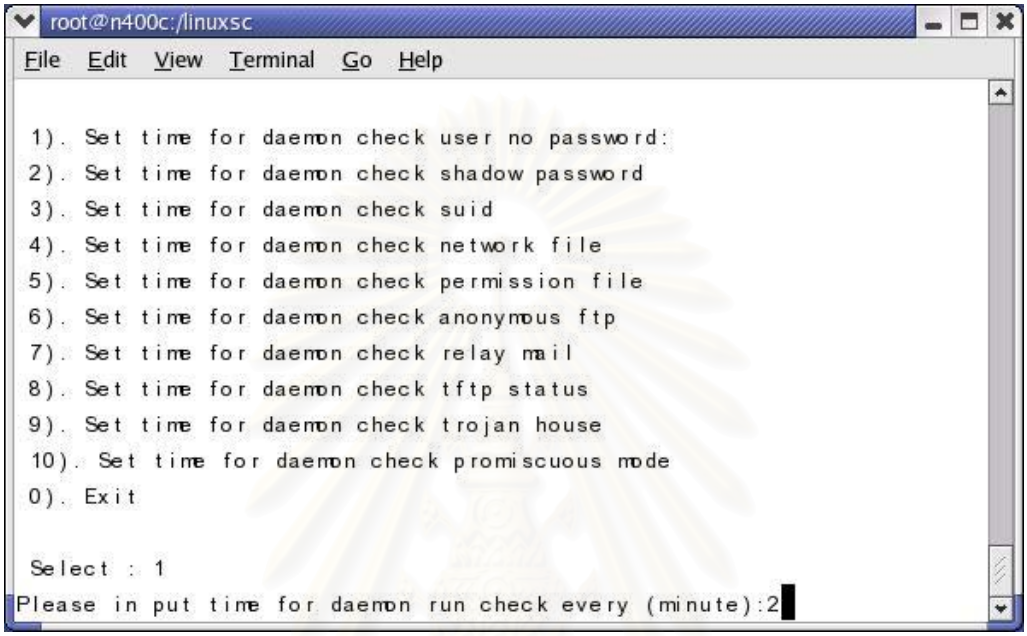


การตั้งเวลาให้โปรแกรมที่ทำงานในโหมดดีมอน

ทำการเรียกสคริปต์ settimedaemon.pl จะปรากฏเมนูของแต่ละฟังก์ชันที่จะตั้งเวลาในการรันแบบดีมอน



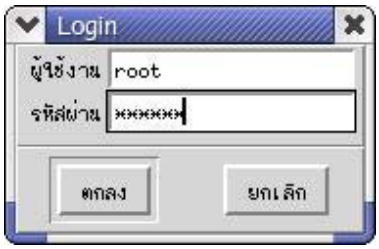
ทำการเลือกข้อของฟังก์ชันที่ต้องการและกำหนด เวลาในการทำงานของดีมอน โดยมีหน่วยเวลาการทำงานเป็นนาที ตัวอย่างในรูปด้านล่าง เป็นการเลือกฟังก์ชันในการตรวจสอบผู้ใช้งานที่ไม่มีรหัสผ่านในแบบดีมอน โดยให้ตรวจสอบทุกๆ 2 นาที



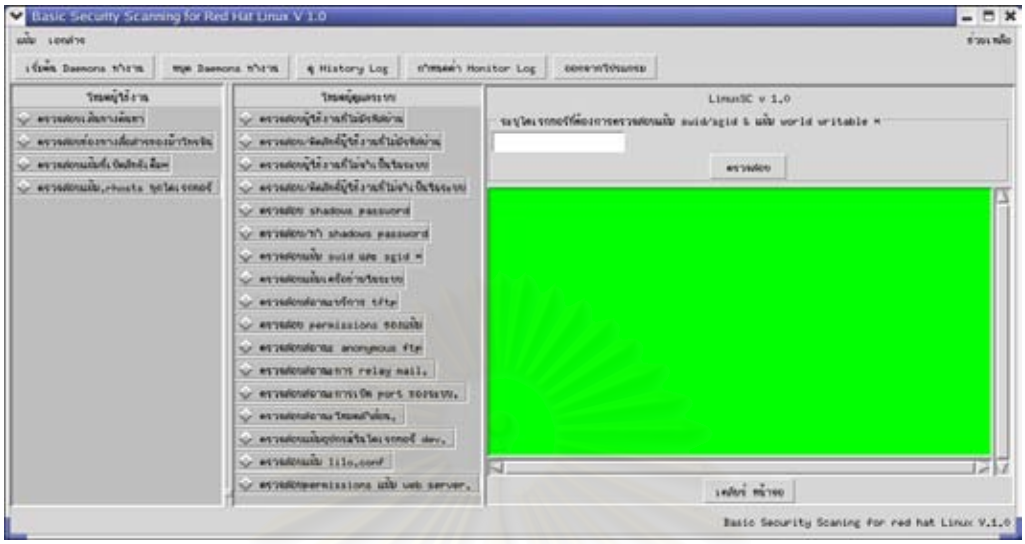
โปรแกรมการตรวจสอบความปลอดภัยพื้นฐานในโหมดภาษาไทย

โปรแกรมสามารถแสดงผลในโหมดภาษาไทยได้เมื่อในระบบปฏิบัติการลีนุกซ์เรดแฮตมีการติดตั้งฟอนต์ภาษาไทย โดยยังคงมีฟังก์ชันการทำงานทุกอย่างเหมือนกับในโหมดมาตรฐาน โดยผู้ใช้งานเองสามารถที่จะเลือกการรันได้ทั้งสองโหมด โดยเมื่อต้องการรันโปรแกรมในโหมดภาษาอังกฤษ โดยเรียกคำสั่ง ./linuxscen.pl ที่ไดเรกทอรีของโปรแกรม และในโหมดภาษาไทยโดยใช้คำสั่ง /linuxscth.pl ที่ไดเรกทอรีของโปรแกรมเช่นเดียวกัน

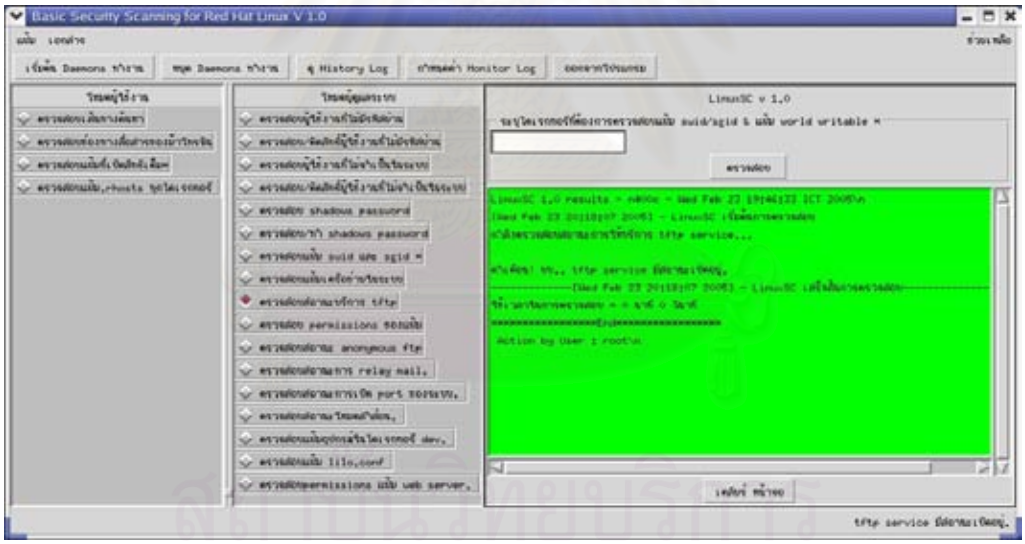
การตรวจสอบผู้ใช้งาน



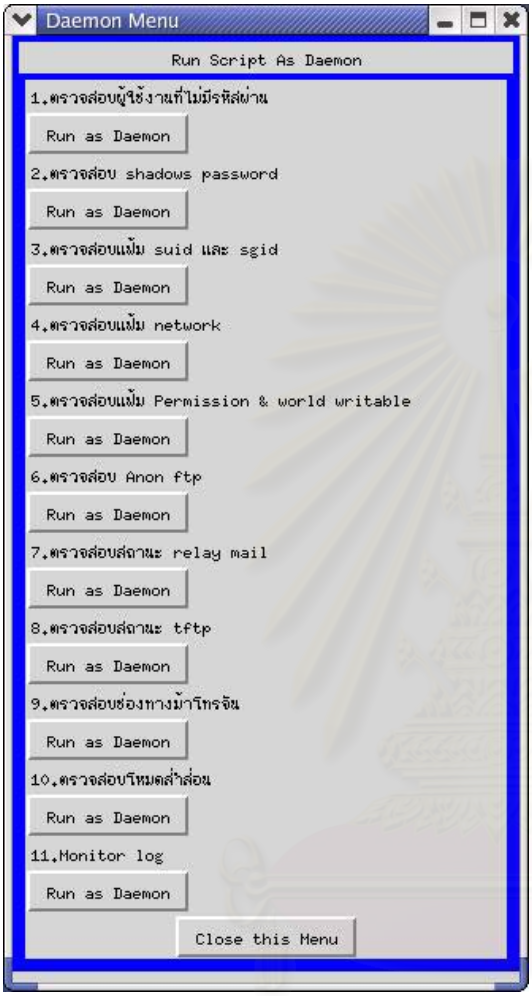
หน้าจอโปรแกรมหลักของโปรแกรมในโหมดภาษาไทย

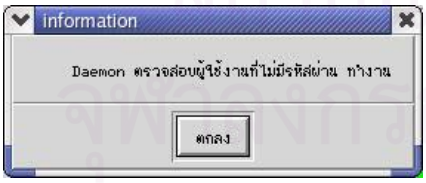


ตัวอย่าง บางฟังก์ชันการตรวจสอบ ในโหมดภาษาไทย



การกำหนดการทำงานของฟังก์ชันการตรวจสอบแบบดีมอน ในโหมดภาษาไทย ซึ่งจะมีการกำหนดเช่นเดียวกันกับในโหมดมาตรฐาน





ภาคผนวก จ

แสดงรายละเอียดของแฟ้มซึ่งเก็บเส้นทางไดเรกทอรี และแฟ้มต้นแบบของบิตอนุญาต

แฟ้ม find.dir เป็นแฟ้มซึ่งเก็บเส้นทางไดเรกทอรีของระบบที่โปรแกรมใช้ในการค้นหาโดยมีเนื้อหาดังต่อไปนี้

/boot, /dev, /etc, /home, /lib, /opt, /root, /sbin, /share, /src, /tmp, /usr, /var

แฟ้ม perms.dir เป็นแฟ้มซึ่งเก็บต้นแบบของบิตอนุญาตของแฟ้มและไดเรกทอรีและชื่อเจ้าของแฟ้มและกลุ่มของเจ้าของแฟ้มอิงตามเอกสารตรวจสอบความปลอดภัยบนยูนิกซ์เวอร์ชันสองโดยมีรูปแบบคือ

“แฟ้มหรือไดเรกทอรี” “บิตอนุญาตของแฟ้ม” “ชื่อเจ้าของแฟ้ม” “ชื่อกลุ่มของแฟ้ม“ โดยมีเนื้อหาดังต่อไปนี้

ต้นแบบบิตอนุญาตของไดเรกทอรี
Directories:
/,755,root,root
/bin,755,root,root
/boot,750,root,root
/dev,755,root,root
/etc,755,root,root
/etc/X11,755,root,root
/etc/cron.daily,750,root,root
/etc/cron.weekly,750,root,root
/etc/cron.hourly,750,root,root
/etc/cron.monthly,750,root,root
/etc/default,750,root,root
/etc/logrotate.d,750,root,root

ต้นแบบบิตอนุญาตของไดเรกทอรี(ต่อ)

/etc/mail,750,root,root

/etc/pcmcia,750,root,root

/etc/ppp,750,root,root

/etc/profile.d,750,root,root

/etc/rc.d,750,root,root

/etc/security,750,root,root

/etc/skel,750,root,root

/etc/sysconfig,750,root,root

/home,755,root,root

/proc,555,root,root

/root,750,root,root

/sbin,750,root,root

/tmp,1777,root,root

/usr/X11R6,755,root,root

/usr/src,750,root,root

/usr/local/sherpa,750,root,root

/usr/local/src,750,root,root

/usr/local/bin,755,root,root

/usr/local/sbin,755,root,root

/usr/bin,755,root,root

/usr/sbin,755,root,root

/var/log,750,root,root

/var/spool/at,700,daemon,daemon

/var/spool/cron,700,root,root

/var/spool/mail,775,root,mail

/var/spool/lpd,775,root,daemon

/var/tmp,1777,root,root

ต้นแบบบิตอนุญาตของแฟ้ม

/etc/amd.conf,600,root,root

/etc/at.deny,600,root,root

/etc/conf.modules,600,root,root

/etc/ftpaccess,600,root,root

/etc/ftpconversions,600,root,root

/etc/ftpgroups,600,root,root

/etc/ftphosts,600,root,root

/etc/ftpusers,600,root,root

/etc/group,644,root,root

/etc/inetd.conf,600,root,root

/etc/motd,644,root,root

/etc/mtab,644,root,root

/etc/passwd,644,root,root

/etc/securetty,750,root,root

/etc/services,644,root,root

/etc/shadow,400,root,root

/usr/bin/suidperl,4711,root,root

/var/lib/logrotate.status,644,root,root

/var/run/syslogd.pid,644,root,root

/var/run/utmp,664,root,utmp

แฟ้ม webperms.dir เป็นแฟ้มซึ่งเก็บต้นแบบของบิตอนุญาตของแฟ้มและไดเรกทอรีที่เกี่ยวกับเวปเซิร์ฟเวอร์ที่ใช้เปรียบเทียบในการตรวจสอบโดยมีรายละเอียดในแฟ้มดังต่อไปนี้

Directories:

/var/log/httpd,700,root,root

/etc/httpd,700,root,root

/var/www,750,root,root

Files

/etc/httpd/conf/httpd.conf,600,root,root

ภาคผนวก ฉ

ส่วนประกอบต่าง ๆ ภายในสคริปต์โปรแกรม

แสดงส่วนประกอบต่างๆ ภายในสคริปต์โปรแกรม

1. Load perl modules and declare (perl)
2. Main Interface (Tk)
3. Verify User Account (Perl)
4. Interface for user (Tk)
5. Interface for super user (Tk)
6. Sub Program for check foreground security (Perl)
7. Sub Program for check background security (Perl)
8. Print output and write to log file (Perl)

1. การโหลดโมดูลต่าง ๆ และการประกาศตัวแปรของเพิร์ล (Load perl module and declare variable (perl))

- การโหลดโมดูลต่างๆ ของเพิร์ลที่ใช้ในโปแกรม เช่น โมดูลการใช้งานส่วนประกอบแบบกราฟิกของทีเค โมดูลการอ่านอินพุทจากคีย์บอร์ด โมดูลการใช้คำสั่งเชลล์ และโมดูลในการใช้ซ็อกเก็ตในการติดต่อพอร์ตต่างๆ เป็นต้น โดยในการโหลดโมดูลในเพิร์ลจะใช้คำสั่ง “use” ตามด้วยชื่อโมดูลที่ต้องการใช้งาน แสดงตัวอย่างได้ดังนี้

```
use Tk;
use Tk::LabFrame;
use Tk::Button;
use Tk::Frame;
use Tk::Label;
use Tk::Listbox;
use Term::ReadKey;
```

```
use Shell;
use Socket;
use Net::SMTP;
```

- การประกาศตัวแปรส่วนกลาง (Global Variable) ในการประกาศตัวแปรส่วนกลางมีตัวแปรที่สำคัญซึ่งมีหน้าที่แยกเป็นแต่ละส่วนได้แก่

ตัวแปรที่เก็บเส้นทางที่อยู่ของแฟ้มต่างๆ

```
$program_home="/linuxsc";                    ระบุที่อยู่ของไดเรกทอรีโปรแกรมที่ติดตั้ง
$config_loc="/etc";                          ระบุที่อยู่ของไดเรกทอรีที่เก็บแฟ้มคอนฟิกระบบ
$log_dir="/var/log";                         ระบุที่อยู่ของไดเรกทอรีที่เก็บแฟ้มล็อกของระบบ
$dirs2find = "$program_home/find.dir";       ระบุที่อยู่ของแฟ้มที่เก็บเส้นทางของระบบปฏิบัติการ
```

ตัวแปรที่เก็บแฟ้มต้นแบบบิตอนุญาตใช้ในการเปรียบเทียบ

```
$perms_list="$program_home/perms.dir";       เป็นการระบุที่อยู่ของแฟ้มที่เก็บบิตอนุญาตของแฟ้มและไดเรกทอรีระบบ
$webperms_list="$program_home/webperms.dir"; เป็นการระบุที่อยู่ของแฟ้มที่เก็บต้นแบบบิตอนุญาตของแฟ้มเว็บ
$anonftpperms_list="$program_home/anonftpperms.dir"; เป็นการระบุที่อยู่ของแฟ้มที่เก็บต้นแบบบิตอนุญาตของแฟ้มเอฟทีพี
```

ตัวแปรที่ใช้ในการสร้างสภาพแวดล้อมของการรายงานของผลการตรวจสอบ

```
$today=`date`;
$ftoday=`date +%y%m%d`; chop($ftoday);
$host=$ENV{"HOSTNAME"};
```

ตัวแปรที่ใช้ในการระบุแฟ้มที่เก็บล็อกการทำงานของโปรแกรม

```
$LinuxSC_outfile = "$program_home/logs.out";
```

2. **การสร้างหน้าต่างหลักของส่วนติดต่อผู้ใช้** (Main Interface (Tk)) ในการสร้างหน้าต่างหลัก ของส่วนติดต่อผู้ใช้ เพื่อเป็นโครงสร้างหลักในการรองรับการโหลดส่วนติดต่อผู้ใช้ในแต่ละแบบ ได้แก่ ส่วนติดต่อผู้ใช้ในลักษณะผู้ใช้งาน และส่วนติดต่อผู้ใช้ในลักษณะผู้ดูแลระบบ หลังจากที่มีการพิสูจน์จากชื่อและรหัสผ่านของผู้ใช้งานแล้วรวมเข้าไปในหน้าต่างหลัก โดยจะทำการประกาศตัวแปรแบบกราฟิกชนิด หน้าต่างหลัก เฟรม และเมนู รอไว้

3. **การพิสูจน์ผู้ใช้งานที่เข้าสู่โปรแกรม** (Verify User Account (perl)) ในระบบผู้ใช้งานโปรแกรมตรวจสอบความปลอดภัย จะแบ่งเป็นสองลักษณะได้แก่ ผู้ใช้งานที่เป็นผู้ใช้งานทั่วไป (Normal User) และผู้ใช้งานที่เป็นผู้ดูแลระบบ (Super User) หรือรูท โดยมีฟังก์ชันในภาษาเพิร์ล ที่ใช้ในการตรวจสอบลักษณะผู้ใช้งานคือ

การโหลดโมดูล ของซีแพน (CPAN module) ชื่อว่า Term::ReadKey ร่วมกับคำสั่ง readMode('noecho') เป็นการกำหนดค่า input mode ให้เป็น noecho เพื่อไม่แสดงค่าของรหัสผ่าน (password) ของ ผู้ใช้งานที่เป็นผู้ดูแลระบบ และใช้คำสั่ง ReadLine ในการอ่านค่าจากคีย์บอร์ด (keyboard) โดยมีหลักการในการพิสูจน์คือ ระบบที่มีการใช้ shadow password เฉพาะผู้ใช้งาน รูท เท่านั้นสามารถที่จะรับรูปแบบของการเข้ารหัส (encrypted form) ของรหัสผ่านด้วยคำสั่ง "getpwuid" ผู้ใช้งานอื่นๆ นอกจากนั้นจะได้เป็นค่า * โดยจะเปรียบเทียบกับค่ารหัสผ่านจากคีย์บอร์ดที่ทำการเข้ารหัสด้วยคำสั่ง "crypt" เพื่อเปรียบเทียบรูปแบบการเข้ารหัส หากมีค่าตรงกันแสดงว่าเป็นรูท แสดงรูปแบบฟังก์ชันดังนี้

```
#!/usr/bin/perl –w                                          (1)
use Term::ReadKey                                           (2)
print "Enter your password;";                              (3)
ReadMode 'noecho';                                          (4)
$password = ReadLine 0;                                     (5)
chomp $passwdrd;                                            (6)
ReadMode 'normal';                                          (7)
Print "\n";                                                 (8)
($username, $encrypted) = ( getwuid $<)[0,1];               (9)
if (crypt($password, $encrypted) ne $encrypted) {           (10)
die " You are not $username\n";                             (11)
}else {                                                     (12)
print "Welcome, $username\n";                               (13)
```

เมื่อผ่านการตรวจสอบผู้ใช้งานด้วยฟังก์ชันนี้ ผู้ใช้งานที่ใช้ชื่อบัญชีผู้ใช้ชื่อว่า รูท ก็จะเข้าลักษณะตรวจสอบความปลอดภัยของผู้ดูแลระบบ ส่วนผู้ใช้งานอื่นๆ ก็จะเข้าสู่ลักษณะตรวจสอบความปลอดภัยสำหรับผู้ใช้งาน

4. **ส่วนติดต่อผู้ใช้ในสำหรับผู้ใช้งาน** (Interface for user (Tk))

จะทำการประกาศตัวแปรกราฟิกชนิด ทีเค เพื่อใช้ในการสร้างอินเตอร์เฟสโดยจะแสดงเฉพาะฟังก์ชันการตรวจสอบสำหรับผู้ใช้ โดยเมื่อผ่านการพิสูจน์รหัสผ่านของผู้ใช้ที่ไม่ใช่ รูท อินเตอร์เฟสที่สร้างขึ้นสำหรับผู้ใช้ก็จะไปรวมกับอินเตอร์เฟสหลักที่โหลดรอไว้ก่อนหน้านี้ เพื่อพร้อมใช้สำหรับผู้ใช้งานที่ไม่ใช่รูท

5. **ส่วนติดต่อผู้ใช้ในสำหรับผู้ดูแลระบบ** (Interface for super user (Tk))

จะทำการประกาศตัวแปรชนิด ทีเค เพื่อใช้ในการสร้างอินเตอร์เฟสโดยจะแสดงฟังก์ชันการตรวจสอบทั้งหมดสำหรับผู้ดูแลระบบ โดยเมื่อผ่านการพิสูจน์รหัสผ่านของผู้ใช้ที่เป็นรูท อินเตอร์เฟสสำหรับผู้ดูแลระบบ จะไปรวมกับอินเตอร์เฟสหลักที่โหลดรอไว้ก่อนหน้านี้ เพื่อพร้อมใช้สำหรับผู้ใช้งานที่เป็น รูท

6. **โปรแกรมย่อยตรวจสอบความปลอดภัยแบบเบื้องหน้า** (Sub Program for check foreground security (Perl))

ในแต่ละโมดูลของการตรวจสอบความปลอดภัยจะเป็นลักษณะของโปรแกรมย่อย แยกตามหน้าที่การตรวจสอบในแต่ละโมดูล โดยจะถูกเรียกจากส่วนติดต่อผู้ใช้ในแต่ละลักษณะ ทั้งในลักษณะผู้ใช้งาน และในลักษณะผู้ดูแลระบบ โดยจะทำหน้าที่แยกตามฟังก์ชันการตรวจสอบโดยจะทำการประกาศโปรแกรมย่อยไว้เพื่อเรียกโปรแกรมย่อยดังกล่าวทำงาน เมื่อผู้ใช้เลือกฟังก์ชันการตรวจสอบ โปรแกรมจะเรียกโปรแกรมย่อยตามฟังก์ชันนั้นๆ ทำงาน แสดงได้ดังนี้

```
If ($select eq “checkpromisc”) # กรณีเลือกการตรวจสอบ promiscuous mode
{
&check_promisc;          # เรียกโปรแกรมย่อย check_promisc ทำงาน
&prtmsg;                 # เรียกโปรแกรมย่อยแสดงผลการทำงาน
}elsif ($select eq “”)    # กรณีไม่มีการเลือกการตรวจสอบใดๆ
{
$msg “กรุณาเลือกฟังก์ชันการตรวจสอบ”;
sub prtmsg;
}
```

```
#โปรแกรมย่อย การตรวจสอบโหมดการทำงานแบบไม่เลือก
sub check_promisc {
check promiscuous mode for interface;
}
```

7. โปรแกรมย่อยตรวจสอบความปลอดภัยแบบเบื้องหลัง (Sub Program for check background security (Perl))

กรณีผู้ดูแลระบบต้องการให้มีการตรวจสอบความปลอดภัยในแบบเบื้องหลัง ผู้ดูแลระบบสามารถสั่งให้โปรแกรมทำงานในแบบเบื้องหลังได้ โดยจะมีแฟ้มดีมอนสคริปต์ซึ่งทำหน้าที่ในการตรวจสอบความปลอดภัยในหัวข้อต่างๆ เช่นเดียวกับการตรวจสอบแบบเบื้องหน้า ซึ่งในสคริปต์ของโปรแกรมหลักจะมีโปรแกรมย่อยที่ไปเรียกแฟ้มดีมอนสคริปต์เหล่านี้ เพื่อทำงานในแบบเบื้องหลังอีกทีหนึ่ง

```
แสดงโปรแกรมย่อยซึ่งเรียกแฟ้มดีมอนสคริปต์ทำงานในแบบเบื้องหลังดังนี้
sub daemon1{
system" ./dcheckusernopasswd.pl";
}
แสดงรายละเอียดแฟ้มดีมอนสคริปต์ซึ่งทำงานตรวจสอบในแบบเบื้องหลังดังนี้
$time = 120;                    # ช่วงเวลาที่ให้โปรแกรมย่อยทำงานทุกๆ กี่วินาที
Main:
&daemonize;                     # เรียกโปรแกรมย่อยที่กำหนดการทำงานในแบบเบื้องหลัง
while(1) {
  sleep($time);
      &checkuser_nopasswd;}  # เรียกโปรแกรมย่อยในการตรวจสอบผู้ใช้งานที่ไม่มีรหัสผ่าน
close;
sub checkuser_nopasswd {

ตรวจสอบผู้ใช้งานที่ไม่มีรหัสผ่าน เมื่อพบทำการส่งเมล์ไปยังรูท; }

sub daemonize {

กำหนดการทำงานในแบบเบื้องหลัง; }
```

โดยจะมีแฟ้มดีมอนสคริปต์ที่ทำหน้าที่ตรวจสอบความปลอดภัยแบบเบื้องหลังดังนี้

1. แฟ้ม dcheckusernopasswd.pl ทำหน้าที่ในการตรวจสอบผู้ใช้งานที่ไม่มีรหัสผ่าน

2. แฟ้ม dchecksuid.pl ทำหน้าที่ในการตรวจสอบแฟ้ม suid และแฟ้ม sgid

3. แฟ้ม dcheckperms.pl ทำหน้าที่ในการตรวจสอบบิตอนุญาตของแฟ้ม

4. แฟ้ม dcheckrelay.pl ทำหน้าที่ในการตรวจสอบการทำรีเลย์ในเมล์เซิร์ฟเวอร์

5. แฟ้ม dchecknetworkfile.pl ทำหน้าที่ในการตรวจสอบแฟ้มเครือข่ายในระบบ

6. แฟ้ม dheckshadowpasswd.pl ทำหน้าที่ในการตรวจสอบการใช้รหัสผ่านชาโดร์ในระบบ

7. แฟ้ม dchecktftp.pl ทำหน้าที่ในการตรวจสอบสถานการณ์เปิดบริการทีเอฟทีพีเซิร์ฟเวอร์

8. แฟ้ม dcheckanonftp.pl ทำหน้าที่ในการตรวจสอบการเอฟทีพีแบบนิรนาม

9. แฟ้ม dcheckpromisc.pl ทำหน้าที่ในการตรวจสอบโหมดการทำงานแบบไม่เลือก

10. แฟ้ม dtrojan.pl ทำหน้าที่ในการตรวจสอบม้าโทรจันในระบบ

11. แฟ้ม mlog.pl ทำหน้าที่ในการตรวจสอบล็อกของระบบ

โดยแฟ้มดีมอนสคริปต์ทั้งหมดจะถูกเรียกจากโปรแกรมย่อยในสคริปต์ของโปรแกรมหลัก

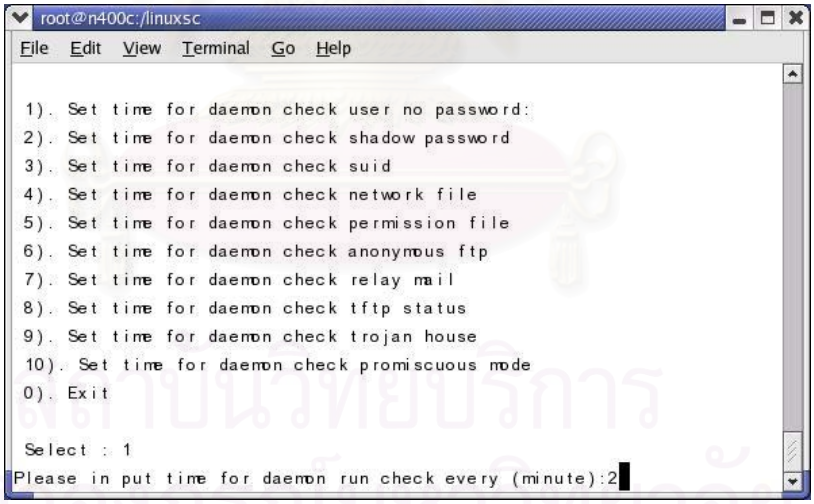
8. **ส่วนของการแสดงผลการตรวจสอบ** (Print output and write to log file (Perl))

ในการแสดงผลการตรวจสอบจะมีโปรแกรมย่อยชื่อว่า “prtmsg” ซึ่งทำหน้าที่ในการพิมพ์ผลการตรวจสอบสู่หน้าจอ และเขียนผลการตรวจสอบไปยังแฟ้มฮีสทอรีล็อก เพื่อเก็บล็อกการทำงานของโปรแกรม แสดงรายละเอียดได้ดังนี้

```
sub prtmsg {
 $list->insert('end', $msg);
        print OUT $msg; }
```

การกำหนดช่วงเวลาในการทำงานของการตรวจสอบแบบดีมอน

ในการตรวจสอบโดยใช้ดีมอนสคริปต์ สามารถที่จะกำหนดช่วงเวลาในการตรวจสอบ โดยมีสคริปต์ชื่อว่า “settimedemon.pl” เป็นตัวกำหนดช่วงเวลาในการทำงาน โดยสคริปต์จะรับค่าของเวลา ซึ่งกำหนดจากผู้ใช้งานโดยเลือกตามฟังก์ชันที่จะให้ตรวจสอบมีหน่วยเป็นนาที แล้วทำการแปลงเป็นวินาที เนื่องจากในเพิร์ล โดยปกติจะรับค่าเวลาเป็นวินาที จากนั้นจะนำค่าที่แปลงได้ไปใส่ในตัวแปรชื่อ “$time” ของดีมอนสคริปต์ทีทำหน้าที่ในการตรวจสอบตามที่ผู้ใช้งานเลือก เพื่อใช้เป็นค่าของช่วงเวลาในการเรียกโปรแกรมแกรมย่อยในดีมอนสคริปต์ให้ทำงาน ซึ่งแนวคิดในการกำหนดช่วงเวลาในการทำงานของดีมอนโดยใช้เพิร์ลสคริปต์นี้ โดยไม่ใช้ความสามารถในการตั้งเวลาของโปรแกรมที่มีอยู่ในลีนุกซ์ เช่น โปรแกรม “crontab” ก็เพื่อความสะดวกและง่ายต่อการควบคุม เนื่องจากสามารถที่จะกำหนดหรือเปลี่ยนแปลงค่าหรือเงื่อนไขต่างๆ ในการตั้งเวลา โดยไม่มีผลกระทบหรือต้องขึ้นอยู่กับตัวระบบปฏิบัติการรวมทั้งค่าพารามิเตอร์ต่างๆ ของโปรแกรมตั้งเวลาที่มีอยู่ในตัวระบบปฏิบัติการ ซึ่งเป็นการแยกสภาพแวดล้อมในการทำงานของโปรแกรมเพื่อป้องกันปัญหา ในกรณีย้ายโปรแกรมไปทำงานในระบบปฏิบัติการลีนุกซ์ตระกูลอื่นๆ แสดงผลการเรียกสคริปต์ “settimedemon.pl” ได้ดังนี้



ประวัติผู้เขียนวิทยานิพนธ์

นายนฤชัย ศรีแสงอยู่ เกิดเมื่อวันที่ 14 ธันวาคม พ.ศ. 2516 ที่จังหวัดนครปฐม สำเร็จการศึกษาระดับปริญญาวิศวกรรมศาสตร์บัณฑิต จากภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสยามในปี พ.ศ. 2539 มีประสบการณ์การทำงานทางด้านเทคโนโลยีสารสนเทศมาเป็นเวลา 8 ปี ขณะทำวิทยานิพนธ์ (พ.ศ.2547) ทำงานอยู่ที่บริษัทโปรลายน์ (ประเทศไทย) จำกัด ซึ่งเป็นบริษัทที่ให้บริการด้านเทคโนโลยีสารสนเทศครบวงจรตั้งแต่ระดับเครื่องคอมพิวเตอร์ส่วนบุคคลจนถึงระดับเครื่องมินิเมนเฟรม ในตำแหน่งผู้ช่วยผู้จัดการฝ่ายซอฟต์แวร์ รับผิดชอบในส่วนของการดูแลซอฟต์แวร์ประเภทกรุ๊ปแวร์ บนทุกระบบปฏิบัติการ และรับผิดชอบกำกับและควบคุมดูแลงานด้านเทคโนโลยีสารสนเทศภายในองค์กร

โดยส่วนตัวมีความสนใจเป็นพิเศษ เกี่ยวกับระบบป้องกันและการรักษาความปลอดภัยในระบบปฏิบัติการต่างๆ ในเครือข่ายคอมพิวเตอร์ เพื่อใช้เรียนรู้ช่องโหว่ภายในและการโจมตีจากภายนอก เพื่อเพิ่มประสิทธิภาพในการป้องกันความปลอดภัยของเครือข่ายคอมพิวเตอร์ให้ดียิ่งขึ้น