ระบบต้นแบบในการล็อกอินเข้าลีนุกซ์ด้วยลายนิ้วมือ

LINUX LOGIN BY FINGERPRINT SYSTEM

โดย

นาย กีรพัฒน์ อรุณรังษี

นาย เกรียงไกร นิตรานนท์

อาจารย์ที่ปรึกษา

อาจารย์ อัครเดช วัชระภูพงษ์

อาจารย์ ธนา หงษ์สุวรรณ

อาจารย์ ธนัญชัย ตรีภาค

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

ระบบต้นแบบในการล็อกอินเข้าลีนุกซ์ด้วยลายนิ้วมือ

LINUX LOGIN BY FINGERPRINT SYSTEM

นาย กีรพัฒน์ อรุณรังษี
นาย เกรียงไกร นิตรานนท์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2547

ปริญญานิพนธ์ปีการศึกษา 2547

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบต้นแบบในการล็อกอินเข้าลีนุกซ์ด้วยลายนิ้วมือ

LINUX LOGIN BY FINGERPRINT SYSTEM

ผู้จัดทำ

1. นาย กีรพัฒน์ อรุณรังษี รหัสประจำตัว 44010030
2. นาย เกรียงไกร นิตรานนท์ รหัสประจำตัว 44010037

_____ อาจารย์ที่ปรึกษา

(อ. อัครเดช วัชระภูพงษ์)

_____ อาจารย์ที่ปรึกษา

(อ. ธนา หงษ์สุวรรณ)

_____ อาจารย์ที่ปรึกษา

(อ. ธนัญชัย ตรีภาค)

ระบบต้นแบบในการล็อกอินเข้าลีนุกซ์ด้วยลายนิ้วมือ

นาย กีรพัฒน์ อรุณรังษี รหัสประจำตัว 44010030
นาย เกรียงไกร นิตรานนท์ รหัสประจำตัว 44010037
อ. อัครเดช วัชระภูพงษ์ อาจารย์ที่ปรึกษา
อ. ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษา
อ. ธนัญชัย ตรีภาค อาจารย์ที่ปรึกษา
ปีการศึกษา 2547

บทคัดย่อ

เนื่องจากในปัจจุบันนี้การใช้งานในระบบปฏิบัติการลีนุกซ์ได้รับความนิยมเพิ่มมากขึ้นเรื่อยๆ และก่อนที่จะสามารถเข้าใช้งานในระบบได้ จะต้องมีการพิสูจน์ตนผู้เข้าใช้งานก่อน ทั้งนี้ก็เพื่อความปลอดภัยในการเข้าถึงข้อมูลภายในระบบ

ในการพิสูจน์ตนผู้เข้าใช้งานนี้ จะใช้ยูสเซอร์เนมและรหัสผ่านเป็นสำคัญ แต่การป้อนข้อมูลยูสเซอร์เนมและรหัสผ่านอาจไม่เพียงพอต่อความปลอดภัยของระบบ เพราะอาจมีผู้อื่นล่วงรู้รหัสผ่านนั้น หรือผู้ใช้เองอาจลืมรหัสผ่านได้ ฉะนั้นจึงพัฒนาระบบการล็อกอินด้วยลายนิ้วมือเพื่อความปลอดภัยที่มากขึ้น อีกทั้งเพื่อความสะดวกในการเข้าใช้งานของผู้ใช้ โดยจะเปลี่ยนระบบการเข้าใช้งานจากเดิมโดยการแก้ไขไฟล์คอนฟิกของระบบ และเขียนโปรแกรมรองรับการเข้าใช้งานใหม่ โดยเปลี่ยนการเข้าใช้งานมาเป็นการสแกนลายนิ้วมือของผู้ใช้เพื่อทำการเข้าใช้งาน จากนั้นระบบจะเปรียบเทียบลายนิ้วมือกับข้อมูลในฐานข้อมูลว่าถูกต้องตรงกันหรือไม่ ถ้าถูกต้องก็สามารถเข้าใช้งานได้ ถ้าไม่ถูกต้องก็จะต้องทำการล็อกอินใหม่อีกครั้ง

ในส่วนของการตรวจสอบยืนยันผู้ใช้งานนั้นจะใช้งาน PAM API (Pluggable Authentication Modules Application Programming Interface) ในการพัฒนา ซึ่งเป็นโมดูลในการตรวจสอบผู้ใช้งานของระบบปฏิบัติการลีนุกซ์ ทำให้การพัฒนาโปรแกรมด้านแอพพลิเคชั่นนั้นมีความยืดหยุ่นมากขึ้น และสามารถนำไปใช้งานร่วมกับโปรแกรม ssh, telnet, xscreensaver และ ftp ได้อีกด้วย

Linux Login by Fingerprint System

Mr. Keeraphat Arunrangsee
Mr. Kriangkrai Nitranont
Mr. Akkradach Watcharapupong Advisor
Mr. Thana Hongsuwan Advisor
Mr. Thanunchai Threepak Advisor
Academic Year 2004

Abstract

Using Linux operating system is more popular now. For security, user authentication must used before access into system.

The authentication is mainly use username and password. But username and password insufficient secure for system. Because of using password may be forgotten by user or may be known by others. So that we developed login system by fingerprint in order to more secure and easy to use for user. This system will change config file and developed new login program to be new system that user must scan their fingerprint before access into system and then system will compare that fingerprint with data in database. If it matches, that user can access into system. But if it doesn't match, that user must login again.

About the authentication part, It works with PAM API (Pluggable Authentication Modules Application Programming Interface) which are modules about authentication in Linux operating system. And that make our program more flexible. It means that we can work with other program for example ssh, telnet, xscreensaver, ftp.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดี เนื่องมาจากการให้โอกาส การดูแล การให้คำแนะนำต่างๆ การสนับสนุน การให้คำสั่งสอนและให้คำปรึกษาเป็นอย่างดีเสมอมาจากอาจารย์อัครเดช วัชระภูพงษ์ อาจารย์ธนา หงษ์สุวรรณ และอาจารย์ธนัญชัย ตรีภาค ซึ่งต้องจักขอบพระคุณท่านเป็นอย่างสูง

ขอขอบคุณภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง และห้องวิจัยและพัฒนาการรักษาความปลอดภัยข้อมูล (ISAG) ที่ได้เอื้อเฟื้อสถานที่ และสิ่งอำนวยความสะดวก เพื่อให้โครงการดำเนินไปอย่างราบรื่น ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆสมาชิกห้องวิจัย ISAG ที่คอยให้ความช่วยเหลือในการทำงาน อีกทั้งเป็นที่ปรึกษา และให้กำลังใจตลอดมา

ท้ายที่สุด ต้องขอขอบพระคุณบุคคลที่สำคัญที่สุดในชีวิตของข้าพเจ้าที่ทำให้ข้าพเจ้ามีทุกวันนี้ คือ บิดา มารดา และบุคคลทุกคนในครอบครัวของข้าพเจ้า อันเป็นที่เคารพรัก อบรมสั่งสอนข้าพเจ้ามาเป็นอย่างดี ทั้งยังให้ความรักความห่วงใยเสมอมา ข้าพเจ้าต้องขอกราบขอบพระคุณมา ณ ที่นี้

กีรพัฒน์ อรุณรังษี
เกรียงไกร นิตรานนท์

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

การใช้งานระบบปฏิบัติการลีนุกซ์นั้น ผู้ใช้งานระบบต้องทำการพิสูจน์ตนก่อนจึงสามารถเข้าใช้งานในระบบได้ ทั้งนี้เพื่อความปลอดภัยต่อระบบปฏิบัติการ ความปลอดภัยต่อข้อมูลภายใน และเป็นการกำหนดสิทธิการเข้าใช้งานของผู้ใช้งานในระดับต่างๆ

วิธีการในการตรวจสอบการพิสูจน์ตนของผู้ใช้นี้ทำโดยการกำหนดยูสเซอร์เนมและรหัสผ่านเป็นหลัก ซึ่งการใช้วิธีการป้อนข้อมูลยูสเซอร์เนมและรหัสผ่านอาจไม่เพียงพอต่อความปลอดภัย เนื่องจากผู้ใช้อาจลืมรหัสผ่าน ทำรหัสผ่านหาย หรืออาจมีผู้อื่นล่วงรู้รหัสผ่านนั้นได้ ฉะนั้นจึงพัฒนาระบบที่นำลายนิ้วมือมาใช้ในการพิสูจน์ตนผู้เข้าใช้งานในระบบปฏิบัติการลีนุกซ์ แทนการใช้รหัสผ่านซึ่งเป็นวิธีการเฉพาะเจาะจงต่อตัวผู้ใช้ และมีความปลอดภัยสูงกว่า เนื่องจากมนุษย์จะมีลายนิ้วมือที่เป็นเอกลักษณ์เฉพาะตัวที่แตกต่างกันออกไป ยากต่อการลอกเลียนแบบ ไม่สามารถทำหายได้ อีกทั้งไม่สามารถให้ผู้อื่นใช้ได้อีกด้วย

ระบบต้นแบบในการล็อกอินเข้าลีนุกซ์ด้วยลายนิ้วมือนี้ถูกพัฒนาโดยมุ่งเน้นในส่วนของการนำเอาระบบการพิสูจน์ตนของผู้เข้าใช้งานระบบด้วยลายนิ้วมือแทนการใช้รหัสผ่าน โดยเมื่อผู้ใช้ต้องการล็อกอินเข้าในบัญชีรายชื่อ (Account) ของตนของตนก็สามารถทำได้โดยการสแกนลายนิ้วมือ ถ้าผลการตรวจสอบตรงกันกับลายนิ้วมือของเจ้าของบัญชีรายชื่อ ผู้ใช้จะสามารถเข้าใช้งานระบบในบัญชีรายชื่อของตนได้ตามปกติ แต่ถ้ามีความผิดพลาดเกิดขึ้นก็จะไม่สามารถใช้งานได้ จะต้องทำการล็อกอินใหม่อีกครั้ง ทั้งนี้ระบบสามารถทำการเพิ่มผู้ใช้งาน และลบผู้ใช้งานได้อีกด้วย

ระบบได้พัฒนาในส่วนของโปรแกรมการล็อกอินขึ้นใหม่ เพื่อเป็นการรองรับระบบล็อกอิน ที่มีการป้อนข้อมูลรหัสของผู้ใช้ (UID) เรียกกระบวนการนี้ว่า Identification และรับลายนิ้วมือของผู้ใช้งานเพื่อนำไปประมวลผลซึ่งในส่วนของการประมวลผลจะนำโมดูล PAM เข้ามาช่วย เรียกกระบวนการนี้ว่า Verfication

การตรวจสอบนั้นระบบจะทำงานร่วมกับ PAM API (Pluggable Authentication Modules Application Programming Interface) เป็นโมดูลที่สนับสนุนการพิสูจน์ตนของผู้ใช้งานในระบบปฏิบัติการลีนุกซ์ ซึ่งสนับสนุนระบบการพิสูจน์ตนหลายๆระบบ สนับสนุนกลไกและนโยบายที่เฉพาะเจาะจงของแต่ละแอพพลิเคชันได้ นอกจากนี้ PAM ยังทำให้สามารถทำให้ใช้เพียงรหัสผ่านเดียวต่อการล็อกอินหลายๆกลไกได้ ในการทำงานร่วมกับ PAM นี้จะทำการพัฒนาโปรแกรมในส่วนของ pam_anubis ขึ้นมาแทน pam_unix เพื่อรองรับการสแกนลายนิ้วมือของผู้ใช้ และนำลายนิ้วมือนั้นมาเปรียบเทียบกับข้อมูลที่มีอยู่ว่าถูกต้องตรงกันหรือไม่

การตรวจสอบลายนิ้วมือใช้วิธีการเปรียบเทียบแบบจับคู่ (Verfication) โดยการนำลายนิ้วมือที่ผู้ใช้สแกนผ่านเครื่องสแกนลายนิ้วมือ AES 4000 Entre PAD มาเปรียบเทียบกับข้อมูลลายนิ้วมือที่เก็บไว้ตามรหัสของผู้ใช้ (UID)

จากที่กล่าวมาจะเห็นได้ว่าระบบต้นแบบในการล็อกอินเข้าลีนุกซ์ด้วยลายนิ้วมือนั้นให้ความปลอดภัยที่มากขึ้น เนื่องจากการปลอมแปลงลายนิ้วมือนั้นทำได้ยาก ในส่วนของผู้ใช้งานระบบก็ได้รับความสะดวกสบายมากขึ้น อีกทั้งการใช้งาน PAM นั้นมีประโยชน์ในด้านที่ทำให้การพัฒนาส่วนของการพิสูจน์ตนของผู้ใช้งานแยกพัฒนาเป็นโมดูล ไม่จำเป็นต้องอยู่ในส่วนของแอพพลิเคชันต่างๆเอง ทำให้มีความยืดหยุ่นต่อการใช้งานมาก ส่งผลให้ผู้พัฒนาในส่วนของแอพพลิเคชันไม่ต้องคำนึงถึงฟังก์ชันในการทำการพิสูจน์ตน

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อมุ่งเน้นเรื่องความปลอดภัยในการล็อกอินเข้าสู่ระบบปฏิบัติการลีนุกซ์
2. เพื่ออำนวยความสะดวกแก่ผู้ใช้งานระบบปฏิบัติการลีนุกซ์
3. เพื่อพัฒนาเทคโนโลยีระบบรักษาความปลอดภัยให้มีประสิทธิภาพมากยิ่งขึ้น
4. เพื่อเป็นการนำเสนอแนวคิด และศึกษาความเป็นไปได้ และเป็นต้นแบบ ในการใช้วิธีการทางชีวมาตร (Biometric) ร่วมกับระบบปฏิบัติการลีนุกซ์

1.3 ขอบเขตของโครงงาน

1. ผู้ใช้สามารถล็อกอินเข้าใช้งานในระบบปฏิบัติการลีนุกซ์ได้โดยใช้การสแกนลายนิ้วมือ โดยผ่านโปรแกรมล็อกอินที่พัฒนาขึ้น (Anubislogin) ทั้งทางเท็กซ์โหมดและกราฟฟิกโหมด
2. ผู้ดูแลระบบสามารถเพิ่มและลบบัญชีรายชื่อผู้ใช้งานระบบได้
3. การพิสูจน์ตนของผู้ใช้งานจะทำงานร่วมกับระบบ PAM ซึ่งเป็นเฟรมเวิร์กในการพิสูจน์ตนของผู้ใช้งานของระบบปฏิบัติการลีนุกซ์
4. ในส่วนของฮาร์ดแวร์ใช้อุปกรณ์สแกนลายนิ้วมือ AES 4000 Entre PAD (USB) เป็นเซนเซอร์ชิพซิลิกอนพัฒนาโดยบริษัท AuthenTec
5. ระบบปฏิบัติการ คือ GNU/Linux Debian 3.0
6. เวอร์ชันของเคอร์เนล (Kernel) ที่ใช้ได้แก่ 2.4.26

โดยภาพรวมของงานวิจัยนี้ขอบเขตของงานวิจัยมุ่งเน้นทางด้านการทำงานระหว่างระบบชีวมาตรกับระบบปฏิบัติการลีนุกซ์ ในด้านการพิสูจน์ตนของผู้ใช้งานด้วยโมดูล PAM แต่งานวิจัยนี้ไม่มุ่งเน้นในแง่ของการจดจำรูปแบบลายนิ้วมือ (Pattern Recognize) และการติดต่อกันระหว่างอุปกรณ์สแกนลายนิ้วมือกับระบบปฏิบัติการลีนุกซ์ ดังนั้นในส่วนของการทำงานสองอย่างนี้จะใช้ Verifinger4.2 Linux SDK มาช่วยในการทำงาน

1.4 วิธีการดำเนินงาน

การดำเนินงานเริ่มต้นด้วยการศึกษาทฤษฎีและแนวความคิดต่างๆ ซึ่งได้แก่

1. รายละเอียดของโมดูล PAM กับการพิสูจน์ตนของผู้ใช้งาน
2. รายละเอียดเกี่ยวกับกระบวนการล็อกอินเข้าสู่ระบบปฏิบัติการลีนุกซ์
3. รายละเอียดของการเขียนโปรแกรมบนระบบปฏิบัติการลีนุกซ์
4. รายละเอียดของลายนิ้วมือ และอุปกรณ์สแกนลายนิ้วมือ
5. นำความรู้ที่ได้ศึกษามาวิเคราะห์และออกแบบเพื่อสร้างโปรแกรม (ซึ่งรายละเอียดของโครงสร้างโปรแกรมอยู่ในบทที่ 4)
6. ทำการทดลอบและค้นหาข้อผิดพลาดของโปรแกรมที่ได้ออกแบบมาเพื่อนำมาแก้ไขและพัฒนาต่อไป (ซึ่งรายละเอียดของทดสอบโปรแกรมอยู่ในบทที่ 5)
7. สรุปผลการดำเนินงานโครงงาน รวมทั้งแนวทางในการปรับปรุงและพัฒนาโครงงานในอนาคต

บทที่ 2
PAM สำหรับระบบปฏิบัติการลีนุกซ์

2.1 บทนำ

เฟรมเวิร์ก PAM (Pluggable Authentication Module) เป็นเทคโนโลยีในการพิสูจน์ตนผู้เข้าใช้งานแบบใหม่ที่ใช้แนวคิดแบบปลั๊กอินโดยไม่ต้องทำการเปลี่ยนแปลงคำสั่งใดๆ เช่น login, ftp, telnet เป็นต้น และ PAM สามารถใช้งานรวมกับการล็อกอินในระบบยูนิกซ์ด้วยกลไกความปลอดภัยอื่นๆ เช่น DCE หรือ Kerberos และภายในเฟรมเวิร์กนี้ยังประกอบด้วยกลไกสำหรับการจัดการบัญชีรายชื่อ (Account), การจัดการเซสชัน (Session) และการจัดการรหัสผ่าน (Password) อีกด้วย

PAM ทำให้ผู้ดูแลระบบสามารถเลือกเซอร์วิซต่างๆในการทำกระบวนการพิสูจน์ตน ซึ่งมีประโยชน์ต่อผู้ดูแลระบบ ดังนี้

- นโยบายในการปรับตั้งค่าระบบที่ยืดหยุ่น
  - นโยบายการพิสูจน์ตนในแต่ละแอพพลิเคชัน
  - สามารถเลือกกลไกการพิสูจน์พื้นฐานสำหรับแอพพลิเคชันที่ไม่มีการเฉพาะเจาะจงไว้
  - ใช้รหัสผ่านหลายรหัสบนระบบที่มีความปลอดภัยสูง

อีกทั้งยังง่ายต่อการใช้งานสำหรับผู้ใช้งานระบบทั่วไปอีกด้วย ดังนี้

- ไม่ต้องทำการป้อนรหัสผ่านใหม่ ในกรณีที่เป็นผู้ใช้งานเดียวกัน
- ใช้เพียงใช้เพียงรหัสผ่านเดียวเท่านั้น แม้รหัสผ่านนั้นเกี่ยวข้องกับวิธีการพิสูจน์ตนที่ แตกต่างกันออกไป โดยใช้การเปรียบเทียบรหัสผ่าน

โมดูล PAM นี้แบ่งออกเป็นสี่รูปแบบที่แตกต่างกันโดยแบ่งตามหน้าที่การทำงาน คือ การพิสูจน์ตน การจัดการบัญชีรายชื่อ การจัดการเซสชัน และการจัดการรหัสผ่าน

โมดูลของการพิสูจน์ตนนั้นทำหน้าที่เกี่ยวกับการพิสูจน์ตนสำหรับผู้ใช้งานระบบ มีการตั้งค่าเปลี่ยนแปลงค่า หรือทำลายค่าการรับรองของผู้ใช้งาน โมดูลนี้จะอนุญาตให้สำหรับผู้ใช้งานที่ได้รับการพิสูจน์ตนแล้วเท่านั้น

โมดูลในการจัดการบัญชีรายชื่อจะทำการตรวจสอบอายุของรหัสผ่าน การหมดอายุของบัญชีรายชื่อ และการกำหนดชั่วโมงการเข้าใช้งานระบบ เมื่อผู้ใช้งานระบบพิสูจน์ตนด้วยการใช้โมดูลการพิสูจน์ตนแล้ว โมดูลจัดการบัญชีรายชื่อจะพิจารณาต่อไปถ้าผู้ใช้นั้นสามารถเข้าใช้งานนระบบได้

โมดูลในการจัดการเซสชันเป็นพื้นฐานในการจัดการเกี่ยวกับการเปิดและปิดเซสชันในการพิสูจน์ตน ทั้งนี้สามารถบันทึกกิจกรรมต่างๆที่เกิดขึ้น หรือสามารถลบรายละเอียดต่างๆหลังจากเซสชันนั้นเสร็จสิ้นแล้ว

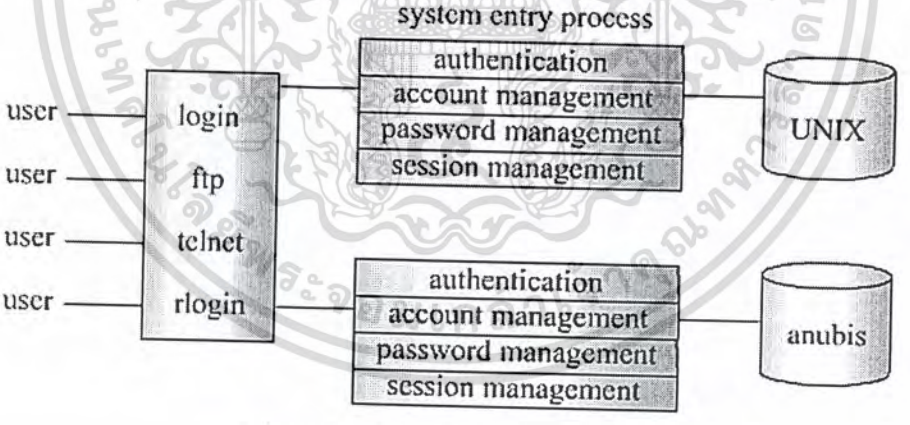
โมดูลในการจัดการรหัสผ่านจัดการเกี่ยวกับการเปลี่ยนรหัสผ่าน และคุณลักษณะที่เกี่ยวข้องกับรหัสผ่าน

PAM อนุญาตสำหรับการพิสูจน์ตนด้วยวิธีการต่างๆผ่านทางสแต็ก เมื่อผู้ใช้งานระบบได้รับการพิสูจน์ตนผ่าน PAM จะมีการเลือกวิธีการต่างๆเพื่อตรวจสอบผู้ใช้งาน ทั้งนี้ขึ้นอยู่กับการตั้งค่าระบบไว้ ผู้ใช้งานระบบต้องมีรหัสผ่านที่พร้อมสำหรับแต่ละวิธีการพิสูจน์ตน ดังนั้นผู้ใช้ไม่จำเป็นต้องจำรหัสผ่านในการดำเนินการในคำสั่งต่างๆเพื่อใช้ในการพิสูจน์ตน คำสั่งที่ใช้ในการพิจารณาจะอยู่ในไฟล์ปรับค่าระบบ (Configuration File) /etc/pam.conf

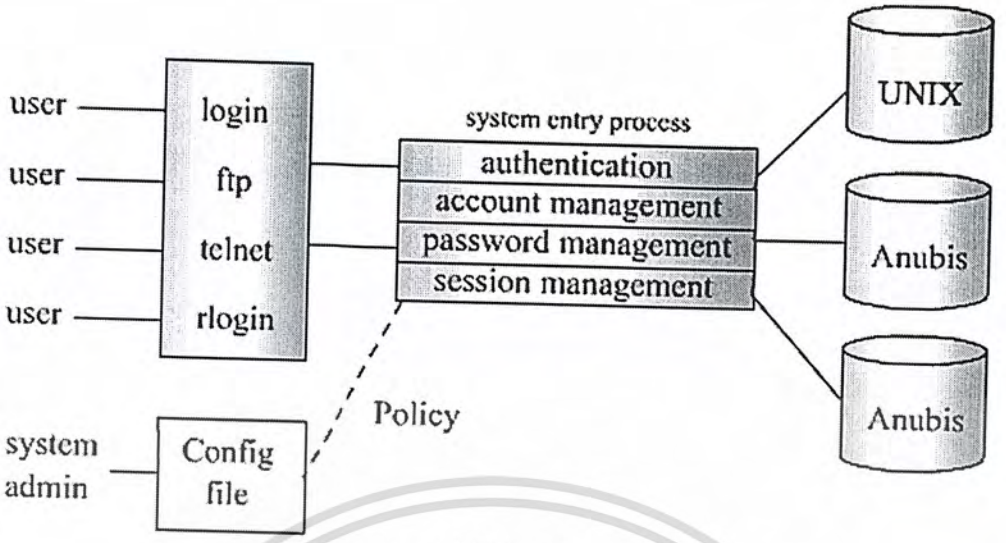
สำหรับสแต็กนั้นต้องการใช้เมื่อผู้ใช้งานระบบต้องจำรหัสผ่านหลายๆรหัส แต่เนื่องจากการใช้วิธีการแมปรหัสผ่าน (Password-mapping) ซึ่งใช้รหัสผ่านหลักในการถอดรหัสผ่านอื่นๆ ดังนั้นจะส่งผลให้ผู้ใช้งานระบบไม่จำเป็นต้องจำ หรือป้อนข้อมูลรหัสผ่านหลายๆรหัส สำหรับทางเลือกอื่นๆคือการใช้งานรหัสผ่านในแต่ละกลไกการพิสูจน์ตนพร้อมๆกัน โดยเป็นที่น่าสังเกตว่าจะมีผลต่อความเสี่ยงด้านความปลอดภัย เพราะความปลอดภัยของแต่ละกลไกมีอยู่จำกัดเท่านั้น

2.2 การทำงานของ PAM

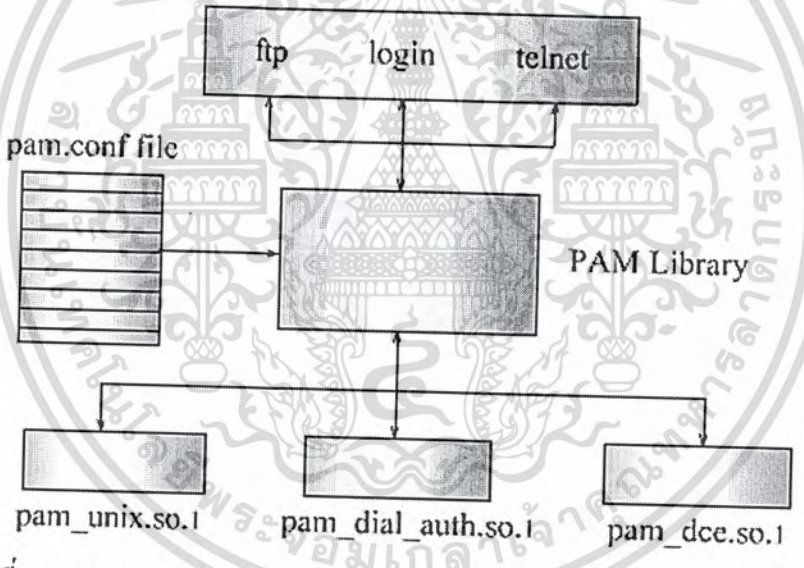
รูปที่ 2-2 และ 2-3 แสดงความสัมพันธ์ระหว่างแอพพลิเคชัน ไลบรารี และโมดูลต่างๆ โดยที่แอพพลิเคชัน (ftp, telnet, login) ใช้ PAM ไลบรารีในการเข้าถึงโมดูลที่เหมาะสม ซึ่งไฟล์ pam.conf นั้นบอกว่าโมดูลใดสมควรเรียกใช้งานในแต่ละแอพพลิเคชัน และโมดูลจะตอบสนองโดยการส่งผ่านมาทางไลบรารีไปยังแอพพลิเคชั่น



รูปที่ 2-1 แสดงระบบพิสูจน์ตนแบบดั้งเดิม



รูปที่ 2-2 แสดงระบบพิสูจน์ตนโดยใช้ PAM



รูปที่ 2-3 แสดงความสัมพันธ์ระหว่างแอพพลิเคชัน PAM ไลบรารี และPAM โมดูล

2.3 PAM ไฟล์

PAM ประกอบด้วยไลบรารี่ โมดูลหลายโมดูล และการตั้งค่าระบบไฟล์ โดย PAM ที่ใช้งานอยู่ในปัจจุบันประกอบด้วยคำเวอร์ชันใหม่ของหลายๆคำสั่งซึ่งใช้ในอินเทอร์เฟส PAM

2.3.1 ไลบรารี PAM

ไลบรารี PAM (/usr/lib/libpam) ทำให้เฟรมเวิร์กมีการจัดการสแต็ก (Stack) และโหลดโมดูลที่เหมาะสม ซึ่งไลบรารี PAM ทำให้เกิดโครงสร้างทั่วไปสำหรับทุกโมดูลที่จะปลั๊กเข้ามาใหม่

2.3.2 โมดูล PAM

แต่ละโมดูลทำให้มีการสร้างกลไกที่เฉพาะเจาะจง และทำให้รูปแบบของโมดูลหลายโมดูล (auth, account, session และ password) สัมพันธ์กับแต่ละโมดูล โดยแต่ละโมดูลต้องการจัดการรูปแบบของโมดูลอย่างน้อยหนึ่งรูปแบบ ตัวอย่างเช่น

- โมดูล pam_unix (/usr/lib/security/pam_unix.so.1) ให้การสนับสนุนสำหรับการพิสูจน์ตน การจัดการบัญชีรายชื่อ การจัดการเซสชัน และการจัดการรหัสผ่าน โมดูลนี้เป็นตัวใช้งานรูปแบบทั้งสี่นี้ ซึ่งก็คือการใช้รหัสผ่านยูนิกซ์สำหรับการพิสูจน์ตน
- โมดูล dial_auth (/usr/lib/security/pam_dial_auth.so.1) สามารถใช้งานสำหรับการพิสูจน์ตนเท่านั้น โดยใช้ข้อมูลที่เก็บในไฟล์ /etc/dialups และไฟล์ /etc/d_password สำหรับการพิสูจน์ตน ทั้งนี้ login ใช้งานโมดูลนี้เป็นหลัก
- โมดูล rhost_auth (/usr/lib/security/pam_rhosts_auth.so.1) ใช้สำหรับการพิสูจน์ตนเท่านั้น โดยใช้ข้อมูลที่เก็บอยู่ในไฟล์ ~/.rhosts และ /etc/host.equiv ผ่านทาง ruserok( ) ทั้งนี้ rlogin และ rsh เรียกใช้งานโมดูลนี้เป็นหลัก
- โมดูล pam_dce (/usr/lib/security/pam_dce.so.1) ให้การสนับสนุนสำหรับการพิสูจน์ตน การจัดการบัญชีรายชื่อ การจัดการรหัสผ่าน โมดูลนี้ใช้ DCE รีจิสทรีสำหรับการพิสูจน์ตน

ด้วยเหตุผลทางด้านความปลอดภัยแล้ว เป็นสิ่งจำเป็นที่ไฟล์เหล่านี้ต้องเป็นสิทธิของ root และต้องมีการตั้งค่าสิทธิไว้ด้วย เช่นสิทธิ group หรือ other ไม่สามารถเขียนไฟล์ได้ โดยถ้าไฟล์ไม่เป็นของ root แล้ว PAM ไม่สามารถโหลดโมดูลได้

2.3.3 PAM คอนฟิกกูเรชันไฟล์

PAM คอนฟิกกูเรชันไฟล์ (/etc/pam.comf) สามารถแก้ไขได้เพื่อเลือกกลไกในการพิสูจน์ตนสำหรับแต่ละแอพพลิเคชัน ภายในไฟล์ประกอบด้วย

service_name module_type control_flag module_path module_options

service_name	บ่งบอกชื่อของเซอร์วิซ
module_type	เป็นรูปแบบโมดูลสำหรับเซอร์วิซ
control_flag	เป็นตัวพิจารณาว่าโมดูลนี้จะดำเนินการต่อไปหรือล้มเหลว
module_path	บ่งบอกเส้นทางไปยังไลบรารี่ที่จะทำหน้าที่ของการเซอร์วิซ
module_options	บ่งบอกออฟชันเฉพาะที่ถูกส่งไปยังเซอร์วิซโมดูล

(สำหรับค่าอื่นๆของไฟล์ต้องมีการประกาศไว้ และสามารถใส่คอมเม้นต์ไว้ที่ไฟล์ได้ โดยเริ่มต้นบรรทัดด้วย #) และจะไม่มีการสนใจข้อมูลในไฟล์นี้หากเกิดสภาวะดังนี้

- มีข้อมูลน้อยกว่าสี่ฟิลด์ในบรรทัด
- มีการให้ข้อมูลที่ไม่ถูกต้องสำหรับ *module_type* หรือ *control_flag*
- ไม่มีการพบชื่อของโมดูล

Service Name	Daemon or Command	Module Type
dtlogin	/usr/dt/bin/dtlogin	auth, account, session
ftp	/usr/sbin/in.ftpd	auth, account, session
init	/usr/sbin/init	session
login	/usr/bin/login	auth, account, session
passwd	/usr/bin/passwd	password
rexd	/usr/sbin/rpc.rexd	auth
rlogin	/usr/sbin/in.rlogind	auth, account, session
rsh	/usr/sbin/in.rshd	auth, account, session
sac	/usr/lib/saf/sac	session
su	/usr/bin/su	auth, account, session
telnet	/usr/sbin/in.telnetd	auth, account, session
ttymon	/usr/lib/saf/ttymon	session
uucp	/usr/sbin/in.uucpd	auth, account, session

ตารางที่ 2-1 ชื่อเซอร์วิซสำหรับ /etc/pam.conf

ตารางที่ 2-1 แสดงรายชื่อของเซอร์วิซที่ถูกต้อง รูปแบบของโมดูลที่สามารถถูกใช้งานกับเซอร์วิสได้ และเดม่อนหรือคำสั่งที่สัมพันธ์กับชื่อของเซอร์วิซ มีหลายรูปแบบโมดูลที่ไม่เหมาะสมสำหรับแต่ละเซอร์วิซ ตัวอย่างเช่น รูปแบบโมดูลรหัสผ่านต้องใช้งานเฉพาะคำสั่ง passwd เท่านั้น คำสั่งนี้ไม่เกี่ยวข้องกับการพิสูจน์ตนดังนั้นจึงไม่มีรูปแบบ โมดูล auth มาเกี่ยวข้องด้วย

สำหรับแต่ละเอนทรี (Entry) จะต้องมีการประกาศ Control_flags อย่างน้อยหนึ่งรูปแบบ เพื่อพิจารณาพฤติกรรมจากโมดูลว่าจะดำเนินการต่อไปหรือล้มเหลว โดยแฟลก (Flags) เหล่านี้จะพิจารณาว่าผลลัพธ์สุดท้ายจะเป็นอย่างไร มีค่าดังนี้

- *required* โมดูลนี้ต้องรีเทิร์นค่าที่สำเร็จ เพื่อที่จะมีผลลัพธ์ทั้งหมดประสบความสำเร็จ
- *optional* ถ้าโมดูลนี้ล้มเหลวแล้วผลลัพธ์ทั้งหมดยังสามารถประสบความสำเร็จได้ถ้าโมดูลอื่นๆในสแต็กนี้รีเทิร์นค่าที่สำเร็จ
- *sufficient* ถ้าโมดูลนี้ประสบความสำเร็จ ให้มองข้ามโมดูลอื่นๆที่ยังคงอยู่ในสแต็ก แม้โมดูลเหล่านั้นระบุเป็น *required* ก็ตาม

ถ้าทุกๆโมดูลประกาศเป็น *required* แล้วการพิสูจน์ตนผ่านทุกๆโมดูลต้องประสบผลสำเร็จ ถ้าปรากฏว่ามีบางโมดูลล้มเหลวแล้วจะมีการรายงานค่าความผิดพลาดจากความผิดพลาดครั้งแรก และทุกโมดูลในสแต็กยังพยายามทำงานต่อไปแต่การเข้าใช้งานก็จะถูกปฏิเสธ

ถ้าไม่มีโมดูลใดประกาศเป็น *required* แล้วจะมีอย่างน้อยหนึ่งเอนทรีสำหรับเซอร์วิซที่ต้องประสบผลสำเร็จ และใช้ในแฟลก *optional* เมื่อมีการประสบความสำเร็จเพียงพอในสแต็ก ควรใช้แฟลกนี้

เท่านั้นถ้ากลไกในการประสบความสำเร็จไม่เป็นที่สำคัญ ตัวอย่างเช่น ถ้าผู้ใช้งานระบบจำเป็นต้องได้สิทธิที่เกี่ยวข้องกับกลไกเฉพาะเพื่อที่จะได้มาซึ่งงานที่สำเร็จแล้วไม่ควรประกาศเป็น *optional*

แฟลก *sufficient* ใช้สำหรับการพิสูจน์ตนที่ประสบผลสำเร็จเพียงหนึ่งครั้งสำหรับผู้เข้าใช้งานในระบบ

โดยทั่วไปไฟล์ pam.conf เป็นดังนี้

```
# PAM configuration
#
# Authentication management
#
login   auth      required    /usr/lib/security/pam_unix.so.1
login   auth      required    /usr/lib/security/pam_dial_auth.so.1
rlogin  auth      sufficient  /usr/lib/security/pam_rhost_auth.so.1
rlogin  auth      required    /usr/lib/security/pam_unix.so.1
dtlogin auth      required    /usr/lib/security/pam_unix.so.1
telnet  auth      required    /usr/lib/security/pam_unix.so.1
su      auth      required    /usr/lib/security/pam_unix.so.1
ftp     auth      required    /usr/lib/security/pam_unix.so.1
uucp    auth      required    /usr/lib/security/pam_unix.so.1
rsh     auth      required    /usr/lib/security/pam_rhost_auth.so.1
OTHER   auth      required    /usr/lib/security/pam_unix.so.1
#
# Account management
#
login   account   required    /usr/lib/security/pam_unix.so.1
rlogin  account   required    /usr/lib/security/pam_unix.so.1
dtlogin account   required    /usr/lib/security/pam_unix.so.1
telnet  account   required    /usr/lib/security/pam_unix.so.1
ftp     account   required    /usr/lib/security/pam_unix.so.1
OTHER   account   required    /usr/lib/security/pam_unix.so.1
#
# Session management
#
login   session   required    /usr/lib/security/pam_unix.so.1
rlogin  session   required    /usr/lib/security/pam_unix.so.1
dtlogin session   required    /usr/lib/security/pam_unix.so.1
telnet  session   required    /usr/lib/security/pam_unix.so.1
uucp    session   required    /usr/lib/security/pam_unix.so.1
OTHER   session   required    /usr/lib/security/pam_unix.so.1
#
# Password management
#
passwd  password  required    /usr/lib/security/pam_unix.so.1
OTHER   password  required    /usr/lib/security/pam_unix.so.1
```

รูปที่ 2-4 แสดงไฟล์ pam.conf

จากรูปที่ 2-4 ไฟล์มีการเจาะจงไว้ว่าเมื่อมีการรันคำสั่ง login แล้วการพิสูจน์ตนจะต้องประสบความสำเร็จสำหรับทั้งโมดูล pam_unix และ pam_dial_auth ส่วน rlogin นั้นการพิสูจน์ตนผ่านทางโมดูล pam_unix จะต้องประสบความสำเร็จ ถ้าพิสูจน์ตนผ่านทาง pam_rhost_auth นั้นล้มเหลวในส่วนของแฟลก *sufficient* จะบ่งบอกว่าสำหรับ rlogin แล้วการพิสูจน์ตนที่ประสบความสำเร็จโดยโมดูล pam_rhost_auth นั้นเพียงพอแล้ว และสำหรับเอนทรีถัดไปจะถูกเพิกเฉย

คำสั่งส่วนใหญ่ต้องการการพิสูจน์ตนที่ประสบผลสำเร็จผ่านทางโมดูล pam_unix แต่การพิสูจน์ตนสำหรับ rsh จะต้องประสบผลสำเร็จโดยผ่านทางโมดูล pam_rhost_auth

การเลือก OTHER นั้นมีสำหรับชื่อเซอร์วิซที่ตั่วค่าไว้เป็นตัวเลือกอัตโนมัติสำหรับคำสั่งอื่นๆที่ต้องการการพิสูจน์ตนที่ไม่ได้รวมอยู่ในไฟล์ โดยตัวเลือก OTHER นี้ทำให้การจัดการไฟล์ง่ายขึ้น เนื่องจากเอนทรีเดียวสามารถใช้หลายคำสั่งได้ครอบคลุมภายในโมดูลเดียวกัน อีกทั้งเมื่อใช้งานตัวเลือก OTHER นั้นเป็น Catch-all ทำให้มั่นใจได้ว่าการเข้าถึงแต่ละครั้งนั้นครอบคลุมเพียงโมดูลเดียวเท่านั้น โดยข้อตกลงของเอนทรี OTHER นั้นคือต้องอยู่ในส่วนล่างสุดของแต่ละรูปแบบโมดูล และในส่วนชื่อเซอร์วิซเป็นแบบ Case-insensitive

เอนทรีที่เหลือในไฟล์ควบคุมคือ การจัดการบัญชีรายชื่อ การจัดการเซสชั่น และการจัดการรหัสผ่าน ด้วยประโยชน์ของชื่อเซอร์วิสที่เป็นตัวเลือกอัตโนมัติ (OTHER) แล้ว ไฟล์ควรเป็นดังนี้

```
#
# PAM configuration
#
# Authentication management
#
login   auth      required    /usr/lib/security/pam_unix.so.1
login   auth      required    /usr/lib/security/pam_dial_auth.so.1
rlogin  auth      sufficient  /usr/lib/security/pam_unix.so.1
rlogin  auth      required    /usr/lib/security/pam_rhost_auth.so.1
rsh     auth      required    /usr/lib/security/pam_rhost_auth.so.1
OTHER   auth      required    /usr/lib/security/pam_unix.so.1
#
# Account management
#
OTHER   account   required    /usr/lib/security/pam_unix.so.1
#
# Session management
#
OTHER   session   required    /usr/lib/security/pam_unix.so.1
#
# Password management
#
OTHER   password  required    /usr/lib/security/pam_unix.so.1
```

รูปที่ 2-5 แสดงไฟล์ pam.conf ที่มีชื่อเซอร์วิซเป็น OTHER

โดยปกติแล้วเอนทรีสำหรับ *module_path* คือ root-relative ถ้าเอนทรีสำหรับ *module_path* ไม่ได้เริ่มต้นด้วยเครื่องหมาย / แล้วเส้นทาง /usr/lib/security ยังไม่เป็นเส้นทางที่บ่งบอกไปยังชื่อไฟล์ เส้นทางที่โมดูลบ่งบอกไปยังตำแหน่งของโมดูลในไดเรคทอรีอื่นๆต้องเริ่มต้นจาก root

ค่าสำหรับ *module_option* นั้นอยู่ใน man pages สำหรับโมดูล (ตัวอย่างเช่น pam_unix(5) และ pam_dce(5)) และโมดูล pam_unix และ pam_dce จะสนับสนุนตัวเลือก use_first_pass และ try_first_pass ซึ่งมีไว้สำหรับการใช้งานรหัสผ่านเดิมอีกครั้งหนึ่งสำหรับการพิสูจน์ตน

ถ้า login กำหนดการพิสูจน์ตนผ่านทั้ง pam_unix และ pam_dce แ ล้วผู้ใช้งานระบบต้องทำการป้อนข้อมูลรหัสผ่านสำหรับแต่ละโมดูล ในสถานการณ์เช่นนี้ถ้ารหัสผ่านเหมือนกันแล้วจะมีการใช้ตัวเลือกโมดูล use_first_pass สำหรับรหัสผ่านเดียวเท่านั้น และควรใช้รหัสผ่านนั้นในการพิสูจน?ตน

สำหรับทุกโมดูล ในกรณีที่รหัสผ่านแตกต่างกันการพิสูจน์ตนจะล้มเหลวและผู้ใช้งานระบบไม่สามารถทำการล็อกอินได้ โดยทั่วไปแล้วตัวเลือกนี้ควรใช้งานกับแฟลกควบคุม *optional* เพื่อให้มั่นใจว่าผู้ใช้งานระบบยังสามารถเข้าใช้งานได้อยู่ ดังตัวอย่างข้างล่างนี้

```
# Authentication management
#
login   auth    required    /usr/lib/security/pam_unix.so.1
login   auth    optional    /usr/lib/security/pam_dce.so.1  use_first_pass
```

รูปที่ 2-6 แสดงไฟล์ pam.conf ที่ใช้แฟลก optional

ถ้าใช้งานตัวเลือกโมดูล try_first_pass แทนแล้วโมดูล DCE จะพร้อมสำหรับรหัสผ่านที่สองในกรณีที่รหัสผ่านไม่เข้ากันหรือมีข้อผิดพลาดเกิดขึ้น ถ้าวิธีการพิสูจน์ตนทั้งสองวิธีนั้นจำเป็นสำหรับผู้ใช้งานระบบในการมีสิทธิเข้าถึงทุกๆเครื่องมือที่ผู้ใช้งานระบบต้องการ การใช้ตัวเลือกนี้อาจนำมาซึ่งความสับสนเพราะผู้ใช้นั้นสามารถมีสิทธิเข้าถึงด้วยรูปแบบการพิสูจน์ตนแบบเดียวเท่านั้น

2.4 การตั้งค่าระบบ PAM

2.4.1 การวางแผนสำหรับ PAM

เมื่อพิจารณาที่จะใช้งาน PAM ในระบบแล้ว ควรมุ่งเน้นในด้านต่างๆดังนี้

- ตัดสินใจว่าความต้องการของระบบคืออะไร โดยเฉพาะอย่างยิ่งโมดูลไหนที่ควรเลือกใช้งาน
- เจาะจงเซอร์วิสที่ต้องการใช้งานเป็นพิเศษ ใช้ OTHER ถ้าเหมาะสม
- พิจารณาคำสั่งที่จะทำให้โมดูลนั้นทำงาน
- เลือกแฟลกควบคุมสำหรับโมดูลนั้น
- เลือกตัวเลือกที่จำเป็นสำหรับโมดูลนั้น

สำหรับการเปลี่ยนแปลงคอนฟิกกูเรชั่นไฟล์ควรปฏิบัติดังนี้

- ใช้เอนทรี OTHER สำหรับแต่ละรูปแบบโมดูลเพื่อที่แต่ละแอพพลิเคชั่นไม่ต้องถูกรวมเข้าไปด้วย
- ทำให้แน่ใจว่ามีการพิจารณาความปลอดภัยของแฟลกควบคุม *sufficient* และ *optional*
- ทบทวน man pages ที่เกี่ยวข้องกับโมดูลเพื่อทำความเข้าใจว่ามันมีหน้าที่อย่างไรและมีตัวเลือกอะไรบ้าง
- ทบทวน man pages เพื่อศึกษาการตอบสนองระหว่างโมดูลสแต็กต่างๆ

2.4.2 การเพิ่มโมดูล DCE PAM

ทำได้โดยการแก้ไขไฟล์ /etc/pam.conf ดังนี้

```
#
# PAM configuration
#
# Authentication management
#
login   auth      sufficient  /usr/lib/security/pam_dce.so.1
login   auth      required    /usr/lib/security/pam_unix.so.1
rlogin  auth      required    /usr/lib/security/pam_unix.so.1
rsh     auth      required    /usr/lib/security/pam_rhost_auth.so.1
OTHER   auth      required    /usr/lib/security/pam_unix.so.1
#
# Account management
#
login   account   required    /usr/lib/security/pam_dce.so.1
login   account   required    /usr/lib/security/pam_unix.so.1
OTHER   account   required    /usr/lib/security/pam_unix.so.1
#
# Session management
#
OTHER   session   required    /usr/lib/security/pam_unix.so.1
#
# Password management
#
passwd  password  required    /usr/lib/security/pam_dce.so.1
passwd  password  required    /usr/lib/security/pam_unix.so.1
```

รูปที่ 2-7 แสดงไฟล์ pam.conf ที่ถูกแก้ไข

แฟลก *sufficient* บนเอนทรี login บ่งบอกให้รู้ว่าถ้าผู้ใช้งานระบบสามารถพิสูจน์ตนผ่านทางโมดูล pam_dce ได้ก็เพียงพอแล้ว และไม่ต้องทำการพิสูจน์ตนผ่านทางโมดูล pam_unix อีกด้วย ช่วงเวลาที่ตรวจสอบที่โมดูล pam_unix คือเมื่อเกิดความล้มเหลวในการพิสูจน์ตนผ่านทาง pam_dce

สองเอนทรีสำหรับการพิสูจน์ตนผ่าน login จะอนุญาตให้ผู้ใช้งานระบบที่มีสิทธิเป็น root สามารถเข้าถึงระบบภายในได้ บรรทัดพิเศษนี้จำเป็นเพราะ DCE ไม่อนุญาตสำหรับการเข้าถึงของ root ถ้าผู้ใช้งานระบบปกติไม่มีรหัสผ่านยูนิกซ์แล้วผู้ใช้งานนนั้นไม่สามารถเข้าใช้งานระบบได้ แต่ root สามารถเข้าใช้งานได้

สังเกตได้จากตัวอย่างนี้ โมดูล DCE ใช้สำหรับ login เท่านั้น แต่ถ้าต้องการเพิ่มเติมแล้วสามารถทำการเพิ่มโมดูล DCE สำหรับเซอร์วิซอื่นๆได้ด้วย ถ้ามีการเพิ่มโมดูล DCE เป็นโมดูล auth สำหรับ login ก็ควรเพิ่มโมดูล account ด้วย และเอนทรี DCE สำหรับเซอร์วิซ passwd ทำให้มั่นใจได้ว่า มีการเปลี่ยนแปลงรหัสผ่าน DCE เมื่อผู้ใช้งานระบบรันคำสั่ง password

2.4.3 การเปลี่ยนแปลงไฟล์ /etc/pam.conf

ถ้าไม่มีการปรับเปลี่ยนค่าของ PAM คอนฟิกกูเรชันไฟล์ อาจเป็นสาเหตุให้ผู้ใช้งานระบบที่มีสิทธิเป็น root ไม่สามารถล็อกอินเข้าใช้งานระบบได้ เพราะ sulogin ไม่ได้ใช้ PAM ผู้ใช้งานระบบที่มีสิทธิเป็น root ต้องทำการปิดและเปิดเครื่องใหม่เพื่อเข้าโหมด single user และทำการแก้ไข

หลังจากทำการเปลี่ยนแปลงไฟล์และทำการตรวจสอบมากเท่าที่จะทำได้ขณะที่ยังเป็น root อยู่ ให้ทำการทดสอบคำสั่งที่อาจมีผลกระทบจากการเปลี่ยนแปลงไฟล์ ตัวอย่างเช่น ถ้ามีการเพิ่มโมดูลใหม่ในเซอร์วิซ telnet แล้วให้ทำการใช้คำสั่ง telnet ไปยังระบบและตรวจสอบว่าการเปลี่ยนแปลงนี้มีผลเป็นไปตามที่คาดหวังไว้หรือไม่

2.4.4 การเพิ่มโมดูล

1. ศึกษาเอกสารที่เกี่ยวกับโมดูล และพิจารณาว่าควรใช้แฟลกควบคุม และตัวเลือกอื่นๆอย่างไร
2. คัดลอกโมดูลใหม่ไปยัง /usr/lib/security
3. ตั้งค่าสิทธิในการเข้าใช้งานระบบ โดยไฟล์โมดูลจะเป็นสิทธิของ root และมีสิทธิเป็น 555
4. แก้ไขไฟล์ PAM คอนฟิกกูเรชัน (/etc/pam.conf) และเพิ่มโมดูลนี้ไปยังเซอร์วิซที่เหมาะสม
5. ทดสอบการเปลี่ยนแปลง

ถ้าเซอร์วิซคือเดม่อนซึ่งเรียกใช้งานเพียงครั้งเดียวเมื่อระบบเริ่มทำงานจำเป็นต้องทำการปิดและเปิดเครื่องใหม่อีกครั้งก่อนการทดสอบ นับเป็นสิ่งสำคัญมากที่ต้องทดสอบระบบก่อนจะทำการปิดและเปิดเครื่องใหม่อีกครั้ง เพื่อป้องกันกรณีที่มีการตั้งค่าระบบที่ผิดพลาดเกิดขึ้น อย่างน้อยที่สุดพยายามทดสอบ rlogin su และ telnet ก่อน

2.4.5 การสร้างรายงานความผิดพลาด

เพิ่มเอนทรีใน /etc/syslog.conf โดยซิสล็อกเดม่อน (Syslog daemon) ต้องรีสตาร์ท หรือ SIGHUP สัญญาณส่งไปหามันสำหรับการเปลี่ยนแปลงใดๆที่มีผลกระทบ การเลือกนี้สามารถเพิ่มไปในไฟล์เพื่อรวมข้อมูลเกี่ยวกับ PAM ดังนี้

- auth.alert ข้อมูลเกี่ยวกับสภาวะที่ควรได้รับการแก้ไขทันที
- auth.crit ข้อมูลที่อันตราย
- auth.err ข้อมูลที่ผิดพลาด
- auth.info ข้อมูลที่บอกรายละเอียด
- auth.debug ข้อมูลที่ดีบัก

เอนทรีข้างล่างนี้แสดงข้อมูลที่แจ้งเตือนทั้งหมดบนคอนโซล (Console) โดยที่มีการส่งอีเมล์ข้อมูลอันตรายไปยัง root และมีการเพิ่มข้อมูลที่บอกรายละเอียดและข้อมูลดีบักในไฟล์ /var/log/pamlog

```
auth.alert                    /dev/console
auth.crit                     'root'
auth.info;auth.debug          /var/log/pamlog
```

รูปที่ 2-8 แสดงรายละเอียดในไฟล์ /var/log/pamlog

แต่ละบรรทัดในล็อกประกอบด้วย Timestamp ชื่อของระบบที่สร้างมันขึ้นมา และข้อมูล โดยไฟล์ pamlog สามารถเก็บรายละเอียดข้อมูลได้จำนวนมาก

บทที่ 3
ทฤษฎีและหลักการวิเคราะห์ลายนิ้วมือ

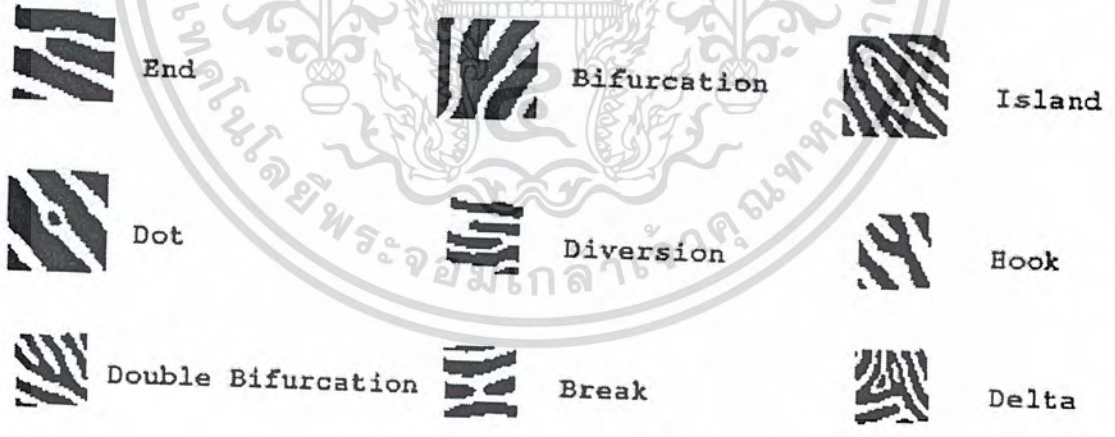
3.1 ความรู้เบื้องต้นของลายนิ้วมือ

บริเวณปลายนิ้วมือของมนุษย์โดยทั่วไปจะเห็นลายนิ้วมือที่มีลักษณะประกอบไปด้วยเส้นสองลักษณะคือ เส้นสันเขา (Ridges) และเส้นหุบเขา (Valleys) ซึ่งทั้งสองลักษณะจะอยู่สลับกันไปตลอด

3.1.1 จุดลักษณะสำคัญบนลายนิ้วมือ (Characteristics)

คือ ตำหนิต่างๆบนลายนิ้วมือ สามารถแบ่งได้เป็นสองลักษณะดังนี้

1. ตำหนิและลักษณะต่างๆของลายเส้นทั่วๆไปเช่น เส้นตรง เส้นโค้ง จุด เส้นแตก เส้นวกกลับ เส้นเวียน เส้นขาด เส้นทะเลสาป และเส้นสองเส้นมาพบกัน(เส้นหักมุม)
2. ลักษณะพิเศษบางอย่าง เช่น
   - ไบเฟอร์เคชั่น คือ เส้นขอบหนึ่งที่แยกออกเป็นสองเส้นหรือมากกว่าสองเส้น
   - ไดเวอร์เจ้น คือ เส้นขอบที่วิ่งขนานกันมาหรือเกือบจะขนาน และแยกห่างออกไป
   - จุดมินูเทีย (minutiae) คือ จุดบนเส้นหยุดหรือเส้นแยก



รูปที่ 3-1 แสดงจุดลักษณะสำคัญบนลายนิ้วมือ

3.1.2 คำจำกัดความที่สำคัญบนลายนิ้วมือ

เป็นการอธิบายคุณลักษณะสำคัญที่ต้องศึกษาและทำความเข้าใจเพราะมีคุณประโยชน์ที่แสดงให้เห็นถึงความแตกต่างของแต่ละลายนิ้วมือซึ่งมีอยู่ 4 ข้อ ได้แก่

- เส้นขอบ (Type Line) คือ เส้นคู่ขนานคู่ในสุด ซึ่งคู่กันมาแล้วแยกตัวออกจากกันเพื่อจะโอบล้อมหรือพยายามโอบล้อมบริเวณลายนิ้วมือที่อยู่ภายใน
- สันดอน (Delta) คือ ลายเส้นในลายนิ้วมือซึ่งอยู่ตรงหน้าและใกล้ที่สุดกับกึ่งกลางของปากทางแยกของเส้นขอบ
- จุดใจกลาง (Core) คือ จุดใดจุดหนึ่งบนปลายเส้นหรือบนบ่าหรือไหล่ของเส้นวกกลับรูปในสุด และต้องอยู่ภายในของลายนิ้วมือ
- บริเวณลายนิ้วมือที่อยู่ภายใน (Pattern Area) คือ พื้นที่บริเวณภายในของลายนิ้วมือที่ถูกเส้นขอบโอบล้อม

3.1.3 รูปแบบของลายนิ้วมือ

แบ่งเป็นสี่กลุ่มดังนี้

3.1.3.1 เส้นโค้ง (Arch)

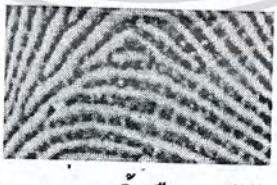
1. โค้งราบ (Plain Arch = PA)

ลายเส้นวิ่งหรือไหลออกไปข้างหนึ่ง ไม่เกิดมุมแหลมหรือพุ่งขึ้นตรงกลาง

รูปที่ 3-2 แสดงลายนิ้วมือแบบโค้งราบ

2. โค้งกระโจม (Tented Arch = TA)

ลายเส้นตรงกลางเกิดเป็นลายเส้นพุ่งขึ้นจากแนวนอนเป็นมุมแหลมหรือมุมฉาก



รูปที่ 3-3 แสดงลายนิ้วมือแบบโค้งกระโจม

3.1.3.2 ลูปหรือมัดหวาย (Loop)

1. มัดหวายปัดขวา (Right Slant Loop = RSL)

มีสันดอนเพียงจุดเดียว มีเส้นวกหลักที่สมบูรณ์อย่างน้อยหนึ่งเส้น มีทิศทางไปด้านขวา

รูปที่ 3-4 แสดงลายนิ้วมือแบบมัดหวายปัดขวา

2. มัดหวายปัดซ้าย (Left Slant Loop = LSL)

มีสันดอนเพียงจุดเดียว มีเส้นวกหลักที่สมบูรณ์อย่างน้อยหนึ่งเส้น มีทิศทางไปด้านซ้าย

รูปที่ 3-5 แสดงลายนิ้วมือแบบมัดหวายปัดซ้าย

3. มัดหวายคู่หรือมัดหวายแฝด (Duoble Loop = DL)

มีลักษณะคล้ายกับลายนิ้วมือแบบมัดหวายข้างบนแต่มากอดหรือกล้ำกันจนเกิดมีสันดอนสองจุด โดยไม่จำเป็นต้องมีขนาดเท่ากัน ประกอบด้วย

รูปที่ 3-6 แสดงลายนิ้วมือแบบมัดหวายคู่หรือมัดหวายแฝด

3.1.3.3 ก้นหอย (Whorl)

ลายนิ้วมือที่มีเส้นเวียนรอบเป็นวงจร ลักษณะเหมือนลานนาฬิกา รูปไข่ วงกลม ประกอบด้วย

1. ก้นหอยธรรมดา (Plain Whorl = W)

รูปที่ 3-7 แสดงลายนิ้วมือแบบก้นหอยธรรมดา

2. ก้นหอยกระเป๋ากลางปัดขวา (Right Central Pocket = RCP)

รูปที่ 3-8 แสดงลายนิ้วมือแบบก้นหอยกระเป๋ากลางปัดขวา

3. ก้นหอยกระเป๋ากลางปัดซ้าย (Left Central Pocket = LCP)

รูปที่ 3-9 แสดงลายนิ้วมือแบบก้นหอยกระเป๋ากลางปัดซ้าย

4. ก้นหอยกระเป๋าข้างปัดขวา (Right Lateral Pocket = RLP)

รูปที่ 3-10 แสดงลายนิ้วมือแบบก้นหอยกระเป๋าข้างปัดขวา

5. ก้นหอยกระเป๋าข้างปัดซ้าย (Left Lateral Pocket = LLP)

รูปที่ 2-11 แสดงลายนิ้วมือแบบก้นหอยกระเป๋าข้างปัดซ้าย

3.1.3.4 ซับซ้อน (Accidental Whorl = AW)

ลายนิ้วมือที่มีลักษณะพิเศษที่ไม่จัดเข้าเป็นลายนิ้วมือชนิดใดโดยเฉพาะ ประกอบด้วยลายนิ้วมือสองแบบมาผสมกัน และมีสันดอนสองสันดอน หรือมากกว่าเช่น กรณีที่ไม่สามารถเข้ากับลายนิ้วมือกลุ่มที่กล่าวมาข้างต้นไม่ได้เลย โดยมีความยุ่งเหยิงและเป็นรูปแบบที่ไม่แน่นอน

รูปที่ 3-12 แสดงลายนิ้วมือแบบซับซ้อน

3.2 หลักการวิเคราะห์ลายนิ้วมือ

การวิเคราะห์ลายนิ้วมือของบุคคลโดยทั่วไปนั้น จะเริ่มด้วยการนำลายนิ้วมือของแต่ละบุคคลแต่ละนิ้วมาหาจุดลักษณะเฉพาะที่สำคัญ

กระบวนการแรกเริ่มของการตรวจพิสูจน์ลายนิ้วมือคือ การอ่านภาพลายนิ้วมือเข้ามาเก็บไว้ในหน่วยความถาวร โดยข้อมูลที่อ่านหรือสแกนเข้ามานั้นจะนำมาผ่านการประมวลผลก่อนแล้วจึงเก็บข้อมูลนั้นไว้ ซึ่งข้อมูลนี้จะถูกเก็บไว้เป็นต้นแบบหรือรหัสของผู้ใช้แต่ละคน

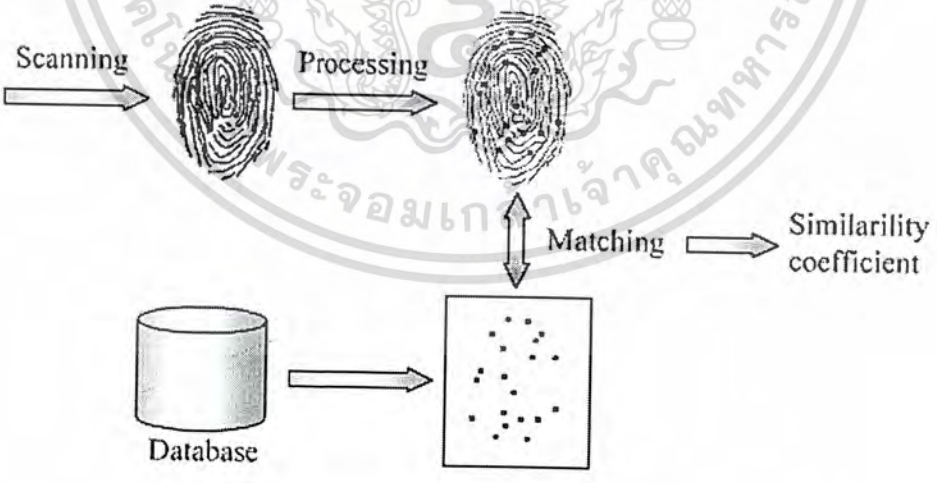
ในขั้นตอนก่อนที่จะนำลายนิ้วมือเข้าไปเก็บนั้นจะต้องผ่านขั้นตอนของการประมวลผลก่อน ในกระบวนการนี้จะทำให้ภาพที่ได้รับการสแกนเข้ามาเกิดความสมบูรณ์มากขึ้นเพราะเมื่อเครื่องได้รับการสแกนภาพเข้ามาแล้ว ภาพที่อ่านได้อาจไม่ชัดเจน พร่าเลือน ก็จะทำให้การประมวลผลในขั้นตอนถัดไปทำได้ด้วยความยากลำบากหรือทำไม่ได้ ซึ่งจะทำให้ผลที่ได้ก็อาจไม่ถูกต้องตามที่ควรจะเป็น เมื่อเกิดปัญหาเช่นนี้ในกระบวนการนี้จึงได้มีการกระทำหลายกระบวนการด้วยกันคือ การกำจัดสัญญาณรบกวน การปรับความมืดสว่างและความแตกต่างของตัวภาพและฉากของภาพ การแปลงภาพเป็นภาพสองระดับ (Binary) การทำให้เส้นลายนิ้วมือบาง (Thinning) การปรับภาพหลังจากแปลงเป็นภาพสองระดับ การหาค่า Threshold ของการปรับภาพเป็นสองระดับและอื่นๆอีกมาก ซึ่งกระบวนการจะมากหรือน้อยขึ้นอยู่กับกับตัวอุปกรณ์นั้นมีการอ่านค่าลายนิ้วมือที่ได้ภาพละเอียดและสมบูรณ์แค่ไหน

เมื่อได้ลายนิ้วมือที่ผ่านการประมวลผลแล้ว ก็นำข้อมูลหรือภาพนี้ไปจัดเก็บในหน่วยความจำถาวร ซึ่งสามารถลบข้อมูลใหม่ด้วยไฟฟ้า โดยภาพที่ถูกจัดเก็บไว้นี้จะถูกเก็บไว้เพื่อใช้ในการเปรียบเทียบกับลายนิ้วมือที่ได้รับการสแกนเข้ามาเมื่อนำตัวอุปกรณ์นี้ไปใช้งาน

รูปที่ 3-13 แสดงกระบวนการทำงานของการวิเคราะห์ลายนิ้วมือ

จากรูปที่ 2-18 เริ่มด้วยการสแกนลายนิ้วมือเข้ามาแล้วนำภาพที่ได้ผ่านการประมวลผลซึ่งจะได้ภาพที่มีประสิทธิภาพมากขึ้นแล้วจึงเก็บภาพนั้นไว้

หลังจากเก็บภาพไว้แล้วนั้นก็มาถึงขั้นตอนการนำไปใช้งาน เมื่อตัวอุปกรณ์ได้ถูกบันทึกหรือเก็บลายนิ้วมือของผู้ที่จะนำไปใช้แล้ว ขั้นตอนในการใช้ก็จะคล้ายกับตอนอ่านลายนิ้วมือเข้ามาเก็บไว้เพียงแต่การอ่านเข้ามาครั้งนี้ข้อมูลที่ได้จะถูกนำเก็บไว้ที่หน่วยความจำชั่วคราว ซึ่งหลังจากสแกนเข้ามาแล้วประมวลผลแล้วก็จะทำการเก็บข้อมูลไว้ที่ส่วนของหน่วยความชั่วคราว ถัดไปก็จะนำข้อมูลที่เก็บอยู่ในส่วนของหน่วยความจำถาวร กับส่วนที่เก็บอยู่ในหน่วยความจำชั่วคราวนั้นมาทำการเปรียบเทียบกัน (Matching) เมื่อได้ผลแล้วก็จะแจ้งผลให้ผู้ใช้ทราบว่ามีความเหมือนกันมากน้อยแค่ไหน



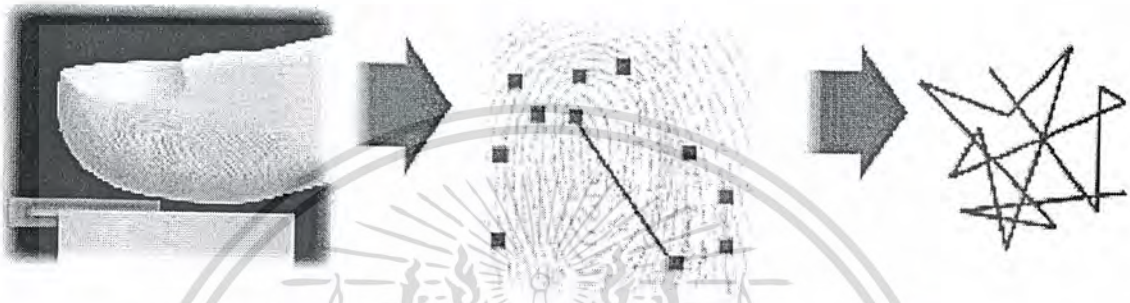
รูปที่ 3-14 แสดงกระบวนการเปรียบเทียบลายนิ้วมือ

จากรูป 2-19 แสดงให้เห็นถึงกระบวนการเปรียบเทียบลายนิ้วมือที่ได้รับการสแกนเข้ามา โดยเริ่มที่การสแกนภาพเข้ามา แล้วทำการประมวลขั้นตอนเดียวกันกับการจัดเก็บตอนแรกแล้วนำภาพที่เก็บไว้ใน

ตอนแรกมาเปรียบเทียบกับภาพที่สแกนเข้ามา ณ ตอนนั้นเพื่อเปรียบเทียบว่ามีความเหมือนหรือแตกต่างมากเพียงใด

3.3 หลักการเปรียบเทียบลายนิ้วมือ

จากทฤษฎี Automated Fingerprint Identification System (AFIS) มีหลักการคือ ระบบAFIS จะตรวจสอบและค้นหา "จุดสำคัญ" บนลายนิ้วมือ และหา "ความสัมพันธ์" ระหว่างจุดต่างๆ เหล่านั้น



รูปที่ 3-15 แสดงกระบวนการวิเคราะห์ลายนิ้วมือระบบ AFIS

วิธีการเปรียบเทียบนี้เริ่มจากการรับข้อมูลลายนิ้วมือจากอุปกรณ์สแกนลายนิ้วมือโดยรูปที่ได้จากอุปกรณ์สแกนลายนิ้วมือจะประกอบไปด้วยอัตราของสีที่ไม่สม่ำเสมอ ดังนั้นก่อนส่งรูปภาพนี้ไปเก็บในไลบรารีของการพิสูจน์ตนจะต้องทำการกรองรูปภาพ (Filter) ก่อน

รูปที่ 3-16 แสดงลายนิ้วมือที่ได้จากอุปกรณ์สแกนลายนิ้วมือ

การกรองรูปภาพที่ได้รับมาจากการสแกนลายนิ้วมือนั้นเป็นการทำให้รูปภาพมีขอบเขตของสีอยู่ระหว่างช่วงสีดำ – สีขาว (0-255) โดยจะทำให้รูปภาพที่มีสีเทาเข้มกลายเป็นสีดำ และสีเทาอ่อนกลายเป็นสีขาว กระบวนการกรองรูปภาพมีรายละเอียดดังนี้

ถ้าเป็นสีขาวหรือสีเทาอ่อน จะพิจารณาให้เป็นขอบนอกของลายนิ้วมือ และแปลงเป็นสีขาว

จำนวนสีที่เกิดขึ้นในแต่ละโทนสีเทาในลายนิ้วมือจะถูกบันทึกไว้ โดยสีเทาที่เข้มที่สุดถูกพิจารณาให้เป็นเสมือนสันเขา (Ridges) และสีเทาที่อ่อนที่สุดถูกพิจารณาให้เป็นเสมือนหุบเขา (Valleys)

ความแตกต่างระหว่างสันเขาและหุบเขาถูกคำนวนและแบ่งครึ่ง โดยค่าของสีเทาที่เข้มมากกว่าสีดำ (0) จนถึงค่าที่น้อยกว่าสีเทาที่เข้มที่สุดที่เป็นแนวสันเขาบวกกับค่าความแตกต่างจะต้องเปลี่ยนเป็นสีดำ

ทั้งหมด และสีเทาที่เข้มกว่าแนวหุบเขาจนถึงค่าที่เทาอ่อนบวกกับค่าความแตกต่างจะต้องเปลี่ยนเป็นสีขาว ดังสมการ

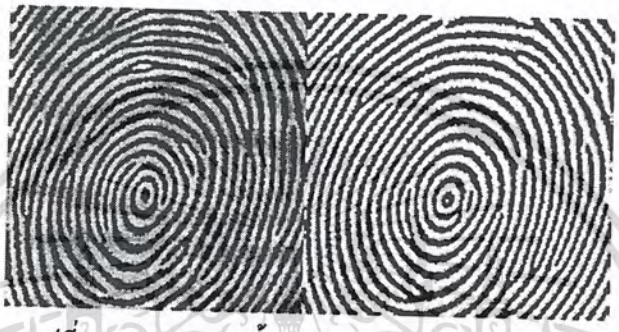
l = สีเทาที่เข้มมากที่สุด

d = สีเทาที่เข้มน้อยที่สุด

x = (l - d) / 2

Ridges = 0 <= l <= (l + x)

Valleys = (l + x) < d <= 255



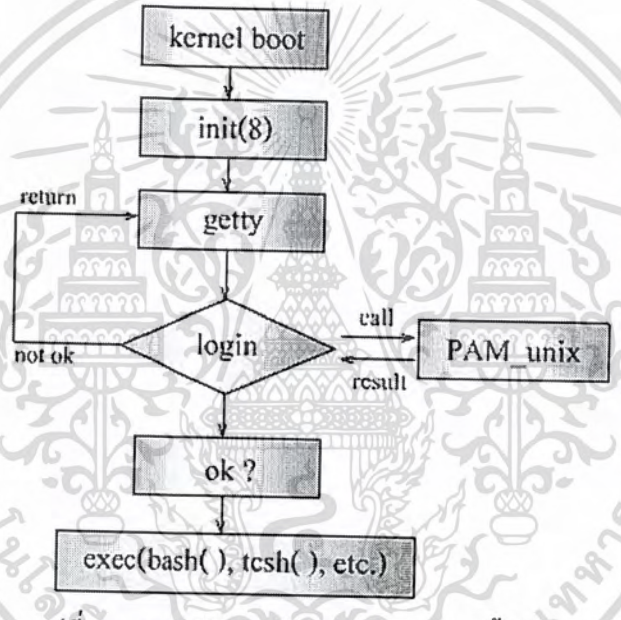
รูปที่ 3-17 แสดงลายนิ้วมือก่อนและหลังทำการกรอง

หลังจากได้รูปภาพที่ผ่านกระบวนการกรองออกมาแล้ว ก็นำรูปภาพนั้นไปทำการเปรียบเทียบและวิเคราะห์ โดยการเปรียบเทียบจะพิจาณาในส่วนของสันเขาและหุบเขา การค้นหานั้นเริ่มต้นด้วยการเลือกจุดเริ่มต้น และทำการพิจารณาไปตามแนวสันเขาจนกระทั่งพบจุดปลายของรูปแบบลายนิ้วมือแบบ Bifurcation ก็จะทำเครื่องหมายเอาไว้ว่าอยู่ในตำแหน่งพิกัด x, y ที่เท่าไหร่ โดยกระบวนเช่นนี้จะทำไปเรื่อยๆจนกระทั่งหมดทั้งรูปภาพ จากนั้นพิจารณาพิกัด x, y ที่ได้ออกมาเพื่อนำไปเปรียบเทียบกับข้อมูลที่เก็บไว้ว่าถูกต้องตรงกันหรือไม่ ในการเปรียบเทียบนั้นจะทำการหมุนภาพลายนิ้วมือที่ได้มาใหม่จนกระทั่งพบว่ามันถูกต้องตรงกันหรือเกิดความล้มเหลวในการเปรียบเทียบ โดยอัตราในการเปรียบเทียบนั้นจะเปรียบเทียบภายใน 30 พิกเซลต้องมีอย่างน้อย 20 พิกเซลขึ้นไปที่ถูกต้องตรงกันจึงสามารถบอกได้ว่าถูกต้อง (โดยสามารถตั้งค่าไว้ได้ว่าต้องการความละเอียดในการตรวจสอบมากน้อยแค่ไหน และขึ้นอยู่กับคุณภาพของอุปกรณ์สแกนลายนิ้วมืออีกด้วย)

บทที่ 4
การออกแบบและพัฒนาโปรแกรม

4.1 รายละเอียดของการพัฒนา

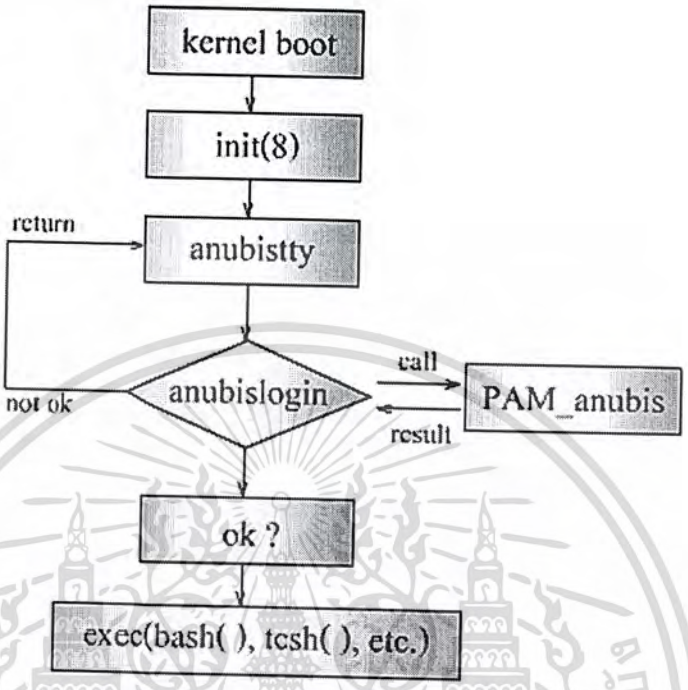
ในการพัฒนาระบบต้นแบบในการล็อกอินเข้าลีนุกซ์ด้วยลายนิ้วมือนี้จำเป็นต้องศึกษาทฤษฎี PAM และทฤษฎีการเปรียบเทียบลายนิ้วมือโดยมีหลักการและแนวคิดที่เปลี่ยนกลไลการล็อกอินแบบเดิมที่ใช้การป้อนข้อมูลยูสเซอร์เนมและรหัสผ่านมาเป็นการสแกนลายนิ้วมือ โดยวิธีการล็อกกอินแบบเดิมนั้นมีกระบวนการดังนี้



รูปที่ 4-1 แสดงโครงสร้างการล็อกอินแบบดั้งเดิม

- หลังจากที่เปิดเครื่อง ระบบเรียกใช้งานเคอร์เนลบูต (kernel boots)
- ต่อมาทำการเรียก init ซึ่งภายในประกอบด้วยไฟล์ /etc/initab ที่เก็บข้อมูลการเรียกใช้งานgetty ไว้
- ต่อมาทำการเรียก getty มาเพื่อทำ opentty( ) คือทำการเปิด /dev/tty และทำ do_prompt( ) คือแสดงรายละเอียดจาก /etc/issue
- ต่อมาใช้งาน login ซึ่งจะทำการรอรับข้อมูลยูสเซอร์เนมและรหัสผ่านจากผู้ใช้งานระบบ
- เมื่อได้รับข้อมูลยูสเซอร์เนมและรหัสผ่านจากผู้ใช้งานระบบแล้ว จะทำการเรียกใช้งานโมดูล pam_unix เพื่อตรวจสอบการพิสูจน์ตนของผู้เข้าใช้งานระบบ
- ถ้าการตรวจสอบถูกต้องผู้ใช้งานก็สามารถเข้าใช้งานระบบได้ ถ้าเกิดความผิดพลาดของข้อมูลที่ป้อนเข้ามาผู้ใช้งานก็ไม่สามารถเข้าใช้งานระบบได้

แต่สำหรับระบบต้นแบบในการล็อกอินเข้าลีนุกซ์ด้วยลายนิ้วมือในทำการเปลี่ยนแปลงขั้นตอนของโครงสร้างการล็อกอิน โดยมีขั้นตอนดังนี้



รูปที่ 4-2 แสดงโครงสร้างการล็อกอินที่พัฒนาใหม่

- หลังจากที่เปิดเครื่อง ระบบเรียกใช้งานเคอร์เนลบูต (kernel boots)
- ต่อมาทำการเรียก init ซึ่งภายในประกอบด้วยไฟล์ /etc/initab ที่เก็บข้อมูลการเรียกใช้งาน anubistty ไว้
- ต่อมาทำการเรียก anubistty มาเพื่อทำ opentty( ) คือทำการเปิด /dev/tty
- ต่อมาใช้งาน anubislogin ที่พัฒนาขึ้นใหม่ ซึ่งจะทำการรอรับข้อมูลรหัสผู้ใช้จากผู้ใช้งานระบบ
- เมื่อได้รับข้อมูลรหัสผู้ใช้จากผู้ใช้งานระบบแล้ว จะทำการเรียกใช้งานโมดูล pam_anubis ซึ่งพัฒนาขึ้นใหม่ เพื่อทำการรอรับลายนิ้วมือจากการสแกนลายนิ้วมือของผู้ใช้งานระบบ และทำการพิสูจน์ตนของผู้เข้าใช้งานระบบโดยการเปรียบเทียบลายนิ้วมือที่ได้จากการสแกนกับข้อมูลลายนิ้วมือที่มีอยู่ในระบบ
- ถ้าการตรวจสอบถูกต้องผู้ใช้งานก็สามารถเข้าใช้งานระบบได้ ถ้าเกิดความผิดพลาดของข้อมูลที่ป้อนเข้ามาผู้ใช้งานก็ไม่สามารถเข้าใช้งานระบบได้

ทั้งนี้ระบบได้พัฒนาในส่วนของ pam_anubis ซึ่งเป็นโมดูล PAM ที่ทำการพิสูจน์ตนที่ทำหน้าที่ในการรับข้อมูลลายนิ้วมือจากการสแกนลายนิ้วมือของผู้ใช้งานระบบ และนำมาเปรียบเทียบกับข้อมูลลายนิ้วมือที่มีอยู่ในระบบ ด้วยวิธีการ Verification คือการใช้รหัสผู้ใช้ (UID) ในการดึงข้อมูลลายนิ้วมือที่มี

อยู่มาเปรียบเทียบกับข้อมูลที่ได้รับมาใหม่จากการสแกนลายนิ้วมือ และรีเทิร์นค่าผลลัพธ์ที่ได้กลับไปยังโปรแกรม anubislogin

นอกจากนี้ยังได้พัฒนาโปรแกรม anubislogin โปรแกรม anubisadd และโปรแกรมanubisdel เพื่อรองรับการใช้งานในการทำการล็อกอินเข้าสู่ระบบ การเพิ่มผู้ใช้งานระบบ และการลบผู้ใช้งานออกจากระบบอีก โดยแอพลลิเคชันต่างๆเหล่านี้จะทำการ include ไลบรารีไว้ จากนั้นการทำงานจะอยู่ที่ไฟล์ pam_anubis.so ซึ่งการจะรู้ได้ว่าเป็นไฟล์ไหนนั้นสามารถรู้ได้โดยการดูใน PAM คอนฟิกกูเรชันไฟล์

4.2 รายละเอียดโปรแกรมที่ได้พัฒนาในเชิงเทคนิค

4.2.1 ส่วนของการล็อกอิน (anubislogin)

4.2.1.1 input/Output Specification

Input : ผู้ใช้งานระบบป้อนข้อมูลรหัสผู้ใช้ (UID) และทำการสแกนลายนิ้วมือไปที่อุปกรณ์สแกนลายนิ้วมือที่ต่ออยู่กับ Linux box

Output : shell prompt หรือ xwindows

4.2.1.2 Functional Specification

1. anubistty(8) : เป็นส่วนที่มาแทนที่ getty โดยทำการเปลี่ยนแปลงใน inittab ในลักษณะดังนี้

```
1 : 2345 : respawn : / sbin / getty  38400  tty 1
2 : 23   : respawn : / sbin / getty  38400  tty 2
3 : 23   : respawn : / sbin / anubistty  tty3
```

ซึ่งภายในจะมีรายละเอียดดังนี้

1.1 *opentty()* : ทำการเปิด /dev/tty3

1.2 *do_prompt()* : ทำการแสดงรายละเอียดจาก /etc /issue

2. pam_anubis( ) : โมดูล PAM ใช้เพื่อการพิสูจน์ตนของผู้ใช้งานระบบ ซึ่งภายในจะมีรายละเอียดดังนี้

2.1 *get_data_from_scanner()* : รับข้อมูลจากเครื่องสแกน

2.2 *get_data_from_file()* : รับข้อมูลจาก database ลายนิ้วมือของแต่ละ user

2.3 *verification(data_fromfile,data_fromscanner)* : ทำการเปรียบเทียบลายนิ้วมือของผู้ใช้งานระบบที่ได้จากอุปกรณ์สแกนลายนิ้วมือกับข้อมูลลายนิ้วมือของผู้ใช้งานที่มีอยู่ระบบ

3. anubislogin : หลังจากการทำการพิสูจน์ตนเรียบร้อยแล้ว สามารถเกิดผลลัพธ์ได้สองสถานการณ์คือ ประสบผลสำเร็จ (success) และล้มเหลว (failure) จะใช้ฟังก์ชันดังนี้ (โดยหัวข้อ 3.1-3.5 เป็นกรณีที่ประสบผลสำเร็จ และหัวข้อ 3.6-3.8 เป็นกรณีที่ล้มเหลว)

3.1 *pam_start()* : เริ่มทำ pam_handle_t สำหรับ anubistty

3.2 *pam_anubis()* : ทำการพิสูจน์ตนผู้ใช้งานระบบ

3.3 *pam_set_item()* : ทำการตั้งค่ายูสเซอร์เนมสำหรับการพิสูจน์ตนครั้งนี้

3.4 *pam_get_item()* : ทำการนำ (get) ค่ายูสเซอร์มา

3.5 *getpwnam(3)* : ทำการนำ (get) ค่าต่างๆมาจากไฟล์ /etc /passwd เช่น group , shell เป็นต้น

3.6 *fork(2)*

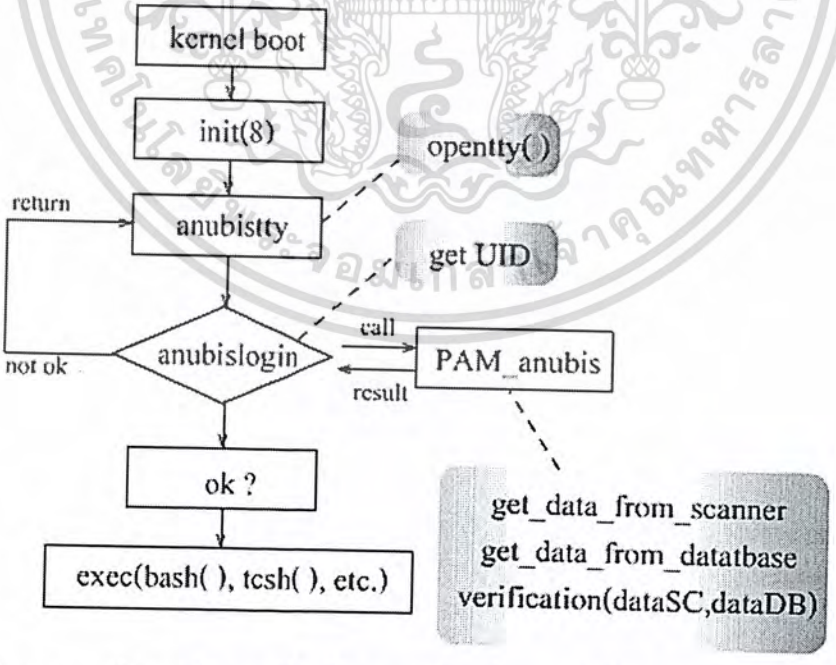
3.7 *execvp(3)* : ทำการ launch shell (เช่น bash(1), tcsh(1) เป็นต้น)

3.8 *return(pam_authen_err)* : การพิสูจน์ตนเกิดความผิดพลาด

3.9 *syslog(3)* : ทำการเก็บข้อมูลลงล็อกและแสดงข้อความที่ผิดพลาด

3.10 *exit(3)* : ย้อนกลับไปที่ขั้นตอนแรก

4.2.1.3 โครงสร้างของซอฟต์แวร์ (Design)



รูปที่ 4-3 แสดงโครงสร้างซอฟต์แวร์ของการพิสูจน์ตนของผู้ใช้ด้วยการสแกนลายนิ้วมือ

4.2.2 ส่วนของการเพิ่มและลบผู้ใช้งานระบบ

4.2.2.1 input/Output Specification

Input : รับค่าข้อมูล (information) ของผู้ใช้งานระบบ เช่น ยูสเซอร์เนม

Output : บันทึกข้อมูล (information) ของผู้ใช้งานระบบนั้นๆ และค่าที่ได้จากการสแกนลายนิ้วมือ หรือ ทำการลบผู้ใช้งานระบบออก

4.2.2.2 Function Specification

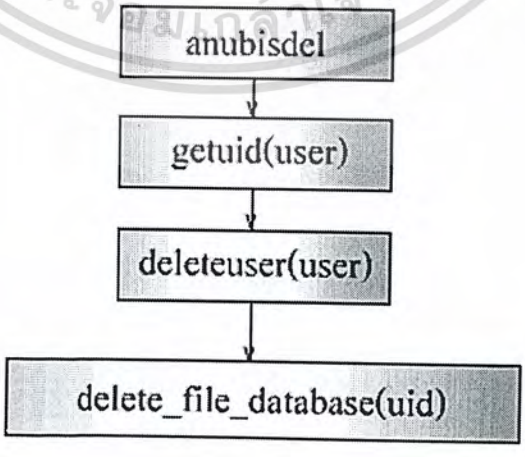
1. anubisadd

- *exec(adduser,user)* : รันคำสั่ง adduser ก่อนเพื่อให้ระบบ สร้างรหัสผู้ใช้ (UID)
- *get_uid(user)* : รับรหัสผู้ใช้จากผู้ใช้งานระบบที่เพิ่มเข้ามาใหม่
- *get_data_from_scanner(3)* : รับข้อมูลจากอุปกรณ์สแกนลายนิ้วมือสามครั้งแล้วหาค่าเฉลี่ย ในกรณีข้อมูลลายนิ้วมือที่ได้นั้นไม่สมบูรณ์ โปรแกรมจะทำการแจ้งเตือนผู้ใช้งานระบบให้ทำการสแกนลายนิ้วมือใหม่อีกสามครั้ง
- *save_data_to_database(data,uid)* : บันทึกข้อมูลโดยใช้ UID เป็นคีย์ (Key) ในการเรียกคืน

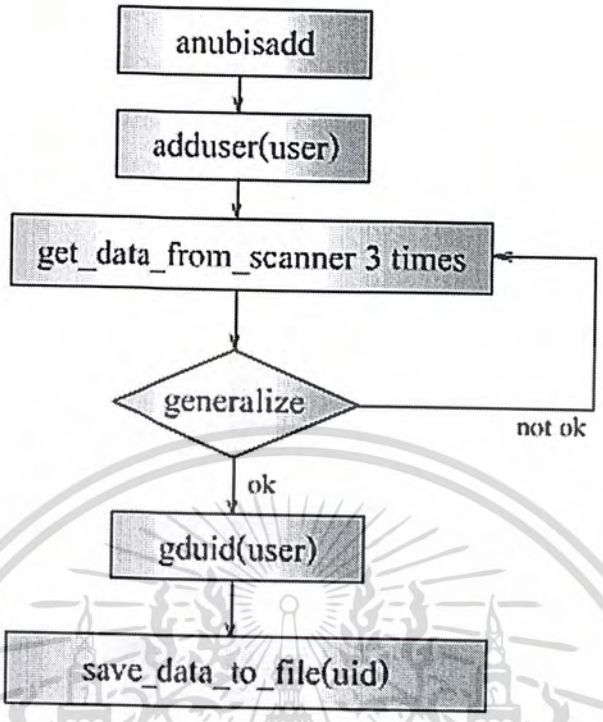
2. anubisdel

- *get_uid(user)* : รับรหัสผู้ใช้จากผู้ใช้งานระบบที่ป้อนเข้ามา
- *delete_database(uid)* : ทำการลบข้อมูลลายนิ้วมือที่เก็บไว้
- *delete_user(user)* : ทำการลบผู้ใช้งานระบบนั้นออก

4.2.2.3 โครงสร้างของซอฟต์แวร์ (Design)



รูปที่ 4-4 แสดงโครงสร้างซอฟต์แวร์ของการลบผู้ใช้งานออกจากระบบ



รูปที่ 4-5 แสดงโครงสร้างซอฟต์แวร์ของการเพิ่มผู้ใช้งานในระบบ

4.3 เครื่องมือที่ใช้ในการพัฒนา

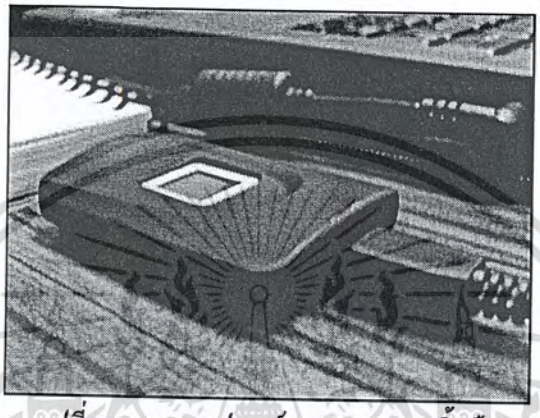
- ใช้ภาษาซี (C Language) ในการพัฒนาระบบ
- Editor VI, Editor Anjuta
- Compiler GCC
- อุปกรณ์สแกนลายนิ้วมือAES4000 EntrePad (USB) เป็นเซนเซอร์ชิพซิลิกอนพัฒนาโดยบริษัท AuthenTec
- Verifinger 4.2 Linux SDK
- ระบบปฏิบัติการ คือ GNU/Linux Debian 3.0
- เวอร์ชันของเคอร์เนล (Kernel) ที่ใช้ได้แก่ 2.4.26
- PAM API

4.4 คุณลักษณะของอุปกรณ์ที่ใช้กับโปรแกรม

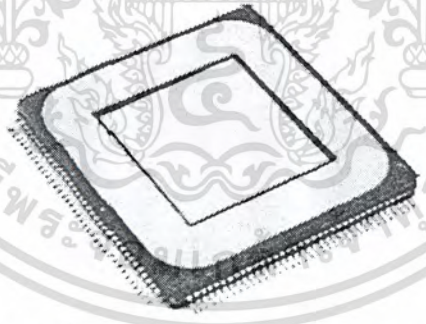
4.4.1 ซอฟต์แวร์ชุดพัฒนา (Verifinger4.2 Linux Software Develop Kit)

เป็นชุดพัฒนาสำหรับผู้พัฒนาระบบชีวมาตร เพื่อความรวดเร็วในการพัฒนาแอพพลิเคชันชีวมาตร โดยเรียกใช้งานฟังก์ชันต่างๆจากไลบรารี (Verifinger DLL) ทำให้มีความน่าเชื่อถือในการตรวจสอบลายนิ้วมือทั้งแบบ Verifinger (1:1) และแบบ Identification (1:N) ทั้งนี้ฟังก์ชัน SDK สามารถใช้ในการติดต่อกับอุปกรณ์สแกนลายนิ้วมือใดๆ ฐานข้อมูลใดๆ และยูสเซอร์อินเทอร์เฟสใดๆ ก็ได้

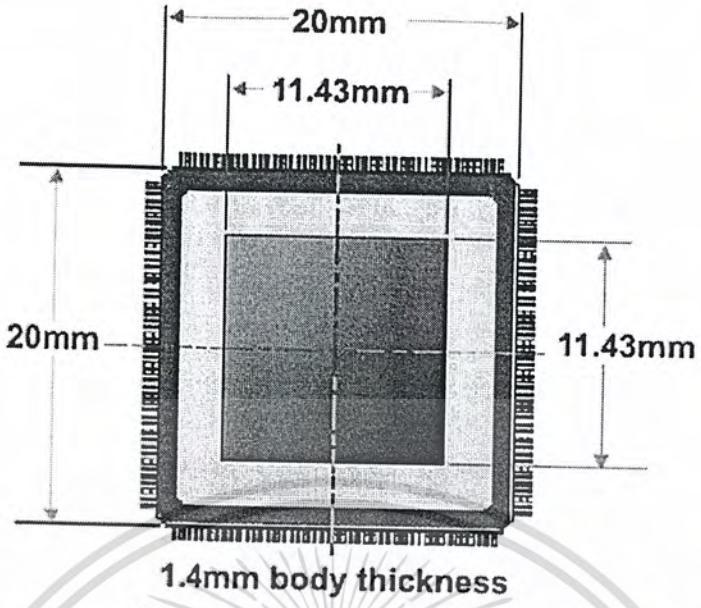
4.4.2 อุปกรณ์สแกนลายนิ้วมือ (AES4000 EntrePad (USB)



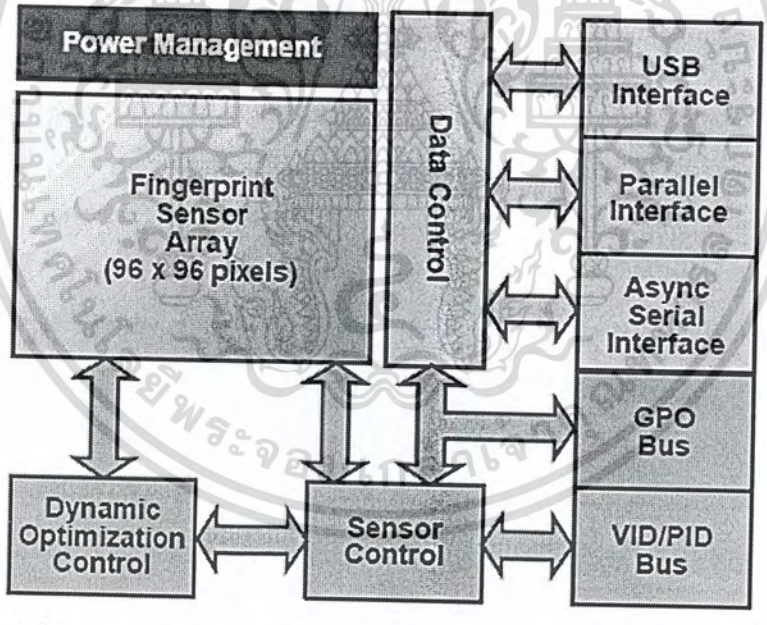
รูปที่ 4-6 แสดงอุปกรณ์การสแกนลายนิ้วมือ



รูปที่ 4-7 แสดง AES4000 Entre PAD



รูปที่ 4-8 แสดงขนาดของ Entre PAD AES4000



รูปที่ 4-9 แสดงความสัมพันธ์ระหว่าง Entre PAD AES4000 กับส่วนต่างๆ

Sensor Name	AES4000 EntrePad
Manufacturer	AuthenTec,Inc
Resolution	250 dpi
Sensor Size	20x 20x 1.5 mm (0.79"x 0.79"x 0.06")
Image capture area	11.43x 11.43 mm (0.45"x 0.45")
Supported OS	MS Windows, Linux
Operating Voltage Range	3.3V or 5.0V
Commercial Temp. Range	0°C through +70°C
High-Rate Image Capture	Up to 52 frames/second
ESD Resistance	IEC 61000-4-2 Level 3 (±8KV)

ตารางที่ 4-1 แสดงคุณสมบัติของอุปกรณ์สแกนลายนิ้วมือ

4.5 กลุ่มผู้ใช้งานโปรแกรม

4.5.1 ผู้ใช้งานระบบปฏิบัติการลีนุกซ์ทั่วไป

เหมาะสำหรับผู้ใช้งานทุกท่านที่ต้องการใช้งานระบบปฏิบัติการลีนุกซ์ ทั้งนี้เพื่อความสะดวกสบายในการใช้งานและความปลอดภัยที่สูงขึ้นกว่าระบบเดิม เนื่องจากไม่ต้องจำรหัสผ่านที่อาจนำมาซึ่งปัญหาต่างๆแก่ผู้ใช้ ได้แก่การลืมรหัสผ่าน การทำหรัสผ่านหาย และการที่ผู้อื่นล่วงรู้รหัสผ่านของตนได้ ผู้ใช้เพียงแค่สแกนลายนิ้วมือเพื่อพิสูจน์ตนก็สามารถเข้าใช้งานระบบได้

4.5.2 ผู้ใช้งานระบบปฏิบัติการลีนุกซ์ที่เป็นผู้ดูแลระบบ

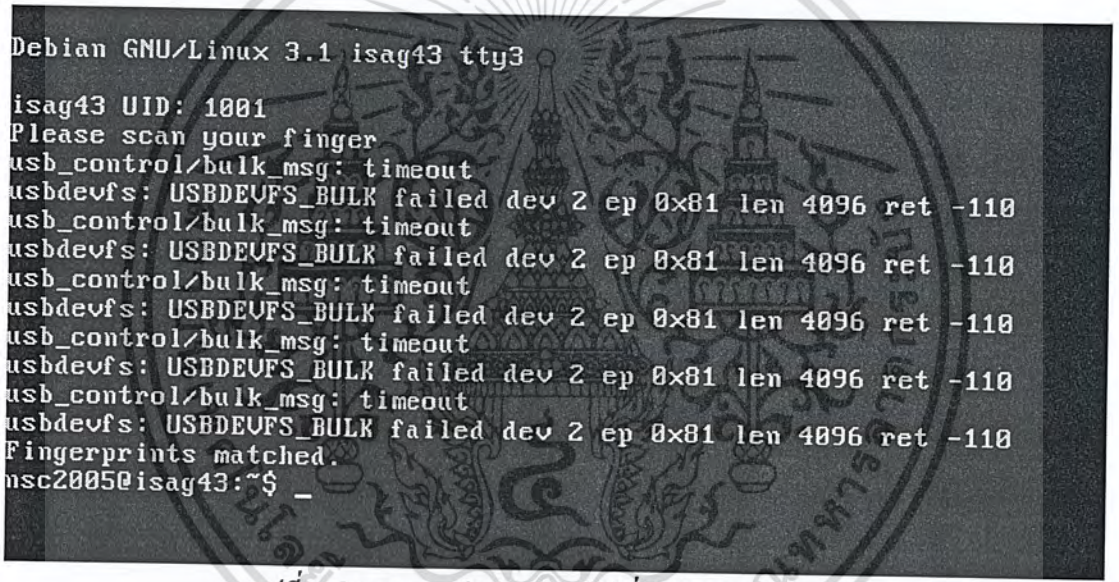
ทางด้านผู้ดูแลระบบก็สามารถทำการเพิ่มและลบบัญชีรายชื่อผู้ใช้ได้เหมือนกับการใช้งานระบบเพิ่มและลบผู้ใช้ทั่วๆไป จึงไม่เกิดปัญหาสับสนในการใช้งานระบบแบบที่พัฒนาขึ้นมาใหม่นี้.

บทที่ 5
การทดสอบการทำงาน

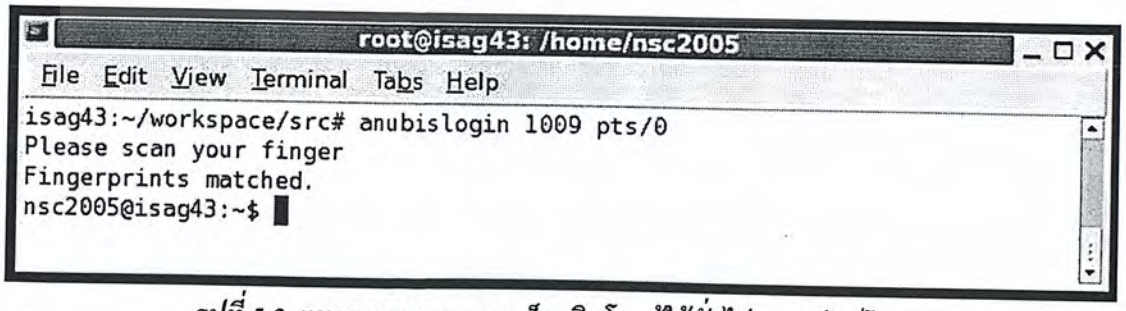
การทดสอบแบ่งออกเป็นสามขั้นตอนหลักได้แก่ การล็อกอิน (anubislogin) การเพิ่มผู้ใช้งานระบบ (anubisadd) และการลบผู้ใช้งานระบบ (anubisdel) โดยมีรายละเอียดในแต่ละขั้นตอนดังนี้

5.1 การทดสอบการล็อกอิน (anubislogin)

5.1.1 การล็อกอินปกติทั่วไป



รูปที่ 5-1 แสดงการล็อกอินโดยผู้ใช้ทั่วไปแบบเท็กซ์โหมด



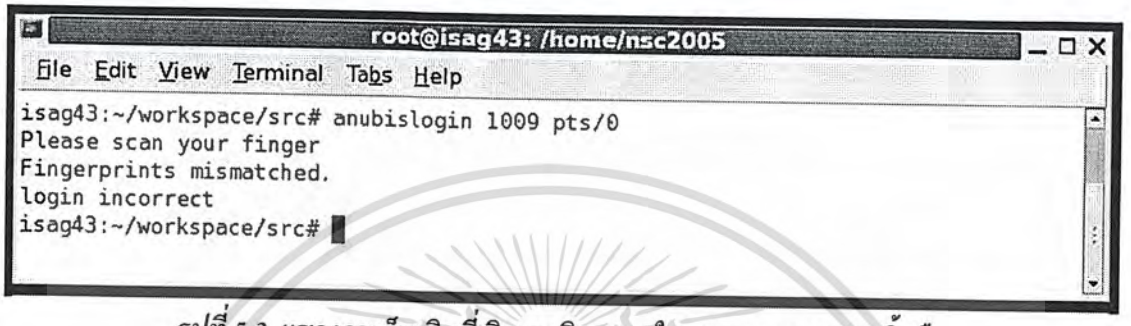
รูปที่ 5-2 แสดงการทดสอบการล็อกอินโดยผู้ใช้ทั่วไปแบบเท็กซ์โหมด

ในการล็อกอินของผู้ใช้งานระบบแบบปกติทั่วไปนั้น การทดสอบเป็นดังนี้

1. ผู้ใช้ล็อกอินโดยการใช้คำสั่ง anubislogin และป้อนข้อมูล UID และ ttyn ของตน

2. ระบบแจ้งบอกให้ผู้ใช้ทำการสแกนลายนิ้วมือ
3. ผลการตรวจสอบลายนิ้วมือถูกต้อง
4. ผู้ใช้สามารถเข้าใช้งานระบบได้ตามปกติ

5.1.2 การล็อกอินที่ไม่ถูกต้อง

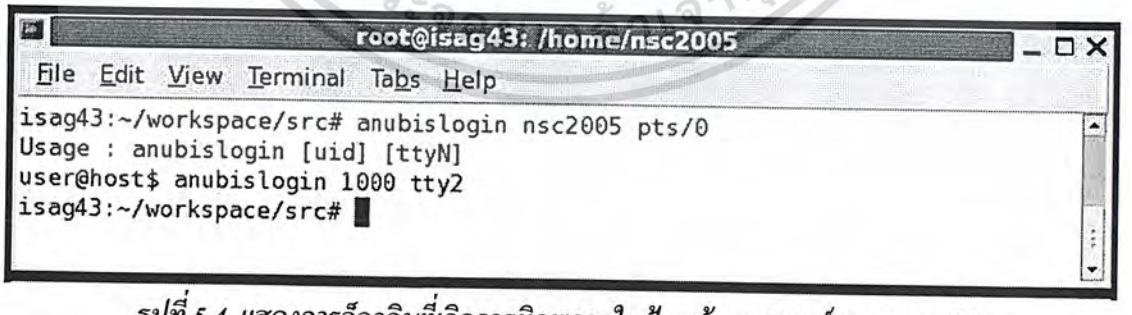


รูปที่ 5-3 แสดงการล็อกอินที่เกิดการผิดพลาดในการตรวจสอบลายนิ้วมือ

ในกรณีที่ผู้ใช้สแกนลายนิ้วมือไม่ดีพอ ซึ่งอาจเกิดจากการวางนิ้วมือไม่ตรง การวางนิ้วมือเบาเกินไป อาจทำให้เกิดข้อผิดพลาดในการสแกนลายนิ้วมือได้ การทดสอบเป็นดังนี้

1. ผู้ใช้ล็อกอินโดยการใช้คำสั่ง anubislogin และป้อนข้อมูล UID และ ttyn ของตน
2. ระบบแจ้งบอกให้ผู้ใช้ทำการสแกนลายนิ้วมือ
3. ผลการตรวจสอบลายนิ้วมือผิดพลาด
4. ผู้ใช้ไม่สามารถเข้าใช้งานระบบได้

5.1.3 การล็อกอินโดยการใส่ยูสเซอร์เนมแทนรหัสผู้ใช้

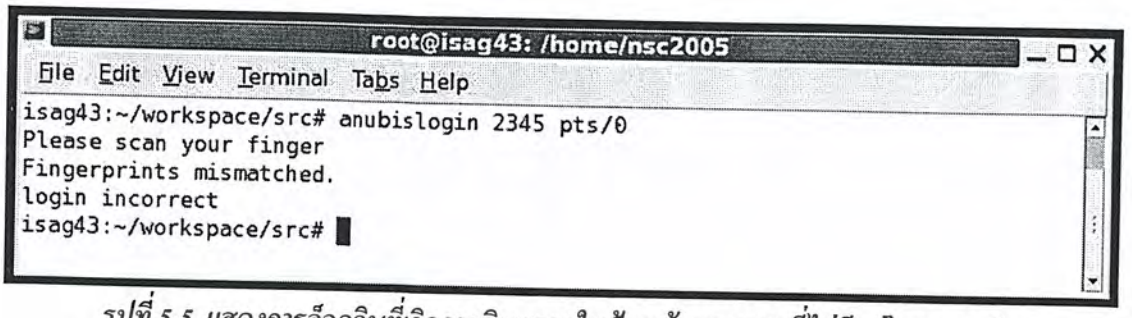


รูปที่ 5-4 แสดงการล็อกอินที่เกิดการผิดพลาดในป้อนข้อมูลยูเซอร์เนมแทน UID

ในกรณีที่การล็อกอินมีการใส่ยูสเซอร์เนมแทน UID นั้นระบบจะไม่อนุญาติให้ล็อกอินได้ตามปกติ การทดสอบเป็นดังนี้

1. ผู้ใช้ล็อกอินโดยการใช้คำสั่ง anubislogin และป้อนข้อมูลยูสเซอร์เนมและ ttyn ของตน
2. ระบบแจ้งบอกให้ผู้ใช้ทราบถึงวิธีการป้อนข้อมูลที่ถูกต้องในการล็อกอิน พร้อมยกตัวอย่าง
3. ผู้ใช้ไม่สามารถเข้าใช้งานระบบได้

5.1.4 การล็อกอินโดยการใส่รหัสผู้ใช้ที่ไม่มีอยู่จริงในระบบ



รูปที่ 5-5 แสดงการล็อกอินที่เกิดการผิดพลาดในป้อนข้อมูล UID ที่ไม่มีอยู่ในระบบจริง

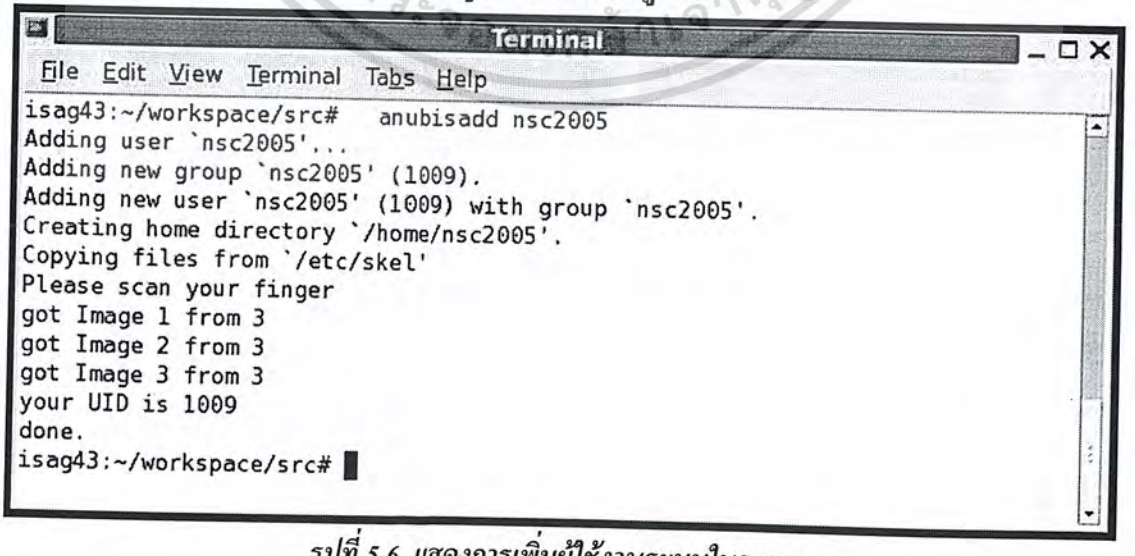
ในกรณีที่การล็อกอินมีการใส่ UID ที่ไม่มีอยู่จริงในระบบ จะไม่สามารถทำการล็อกอินได้ การทดสอบเป็นดังนี้

1. ผู้ใช้ล็อกอินโดยการใช้คำสั่ง anubislogin และป้อนข้อมูล UID และ ttyn ของตน (UID ที่ไม่มีอยู่จริงในระบบ)
2. ระบบแจ้งบอกให้ผู้ใช้ทำการสแกนลายนิ้วมือ
3. หลังจากผู้ใช้สแกนลายนิ้วมือแล้ว ระบบจะแจ้งบอกว่าผลการตรวจสอบลายนิ้วมือผิดพลาด
4. ผู้ใช้ไม่สามารถเข้าใช้งานได้

5.2 การทดสอบการเพิ่มผู้ใช้งานระบบ (anubisadd)

ในการเพิ่มยูสเซอร์ในระบบนั้น จะสงวนไว้ให้ผู้ที่มีสิทธิความเป็น root เท่านั้นที่สามารถทำการเพิ่มผู้ใช้งานระบบได้

5.2.1 ผู้ใช้ที่มีสิทธิความเป็นรูททำการเพิ่มผู้ใช้งานระบบ



รูปที่ 5-6 แสดงการเพิ่มผู้ใช้งานระบบในระบบ

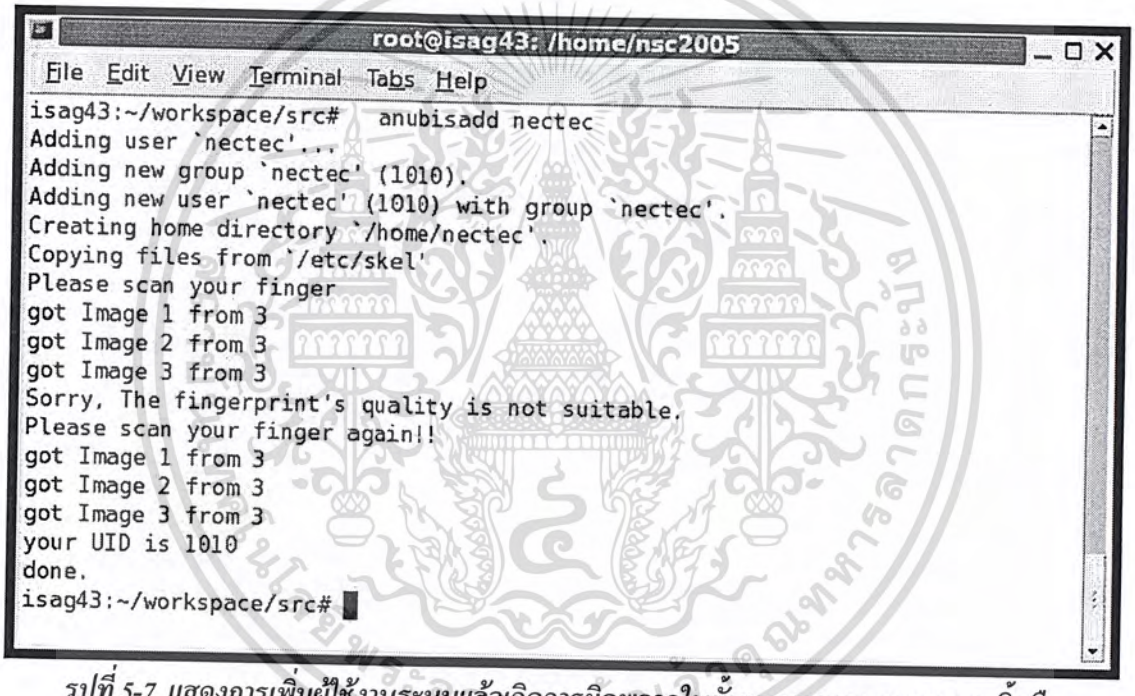
การทดสอบเป็นดังนี้

1. ผู้ใช้ (ที่มีสิทธิความเป็น root) เพิ่มผู้ใช้งานระบบโดยการใช้คำสั่ง anubisadd และป้อนยูสเซอร์เนม

2. ระบบแจ้งบอกให้ผู้ใช้ทำการสแกนลายนิ้วมือ 3 ครั้ง

3. การเพิ่มผู้ใช้งานระบบเสร็จเรียบร้อย

5.2.2 ผู้ใช้ที่มีสิทธิความเป็นรูททำการเพิ่มผู้ใช้งานระบบแล้วเกิดการผิดพลาดในขั้นตอนการสแกนลายนิ้วมือ



รูปที่ 5-7 แสดงการเพิ่มผู้ใช้งานระบบแล้วเกิดการผิดพลาดในขั้นตอนของการสแกนลายนิ้วมือ

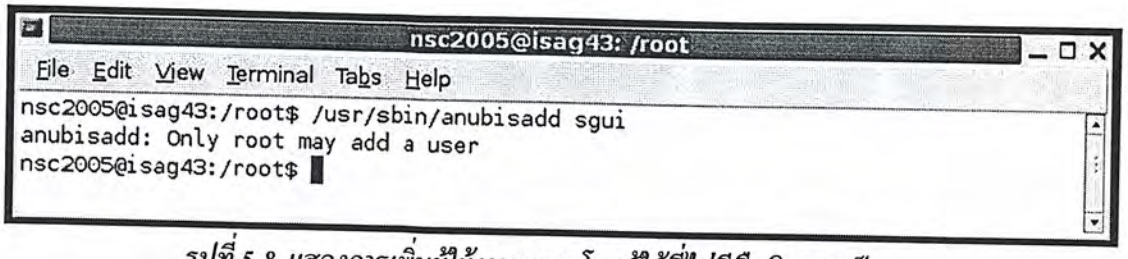
การทดสอบเป็นดังนี้

1. ผู้ใช้ (ที่มีสิทธิความเป็น root) เพิ่มผู้ใช้งานระบบโดยการใช้คำสั่ง anubisadd และป้อนยูสเซอร์เนม

2. ระบบแจ้งบอกให้ผู้ใช้ทำการสแกนลายนิ้วมือ 3 ครั้ง

3. การเพิ่มผู้ใช้งานระบบเสร็จเรียบร้อย

5.2.3 ผู้ใช้งานทั่วไปทำการเพิ่มผู้ใช้งานระบบ



รูปที่ 5-8 แสดงการเพิ่มผู้ใช้งานระบบโดยผู้ใช้ที่ไม่มีสิทธิความเป็น root

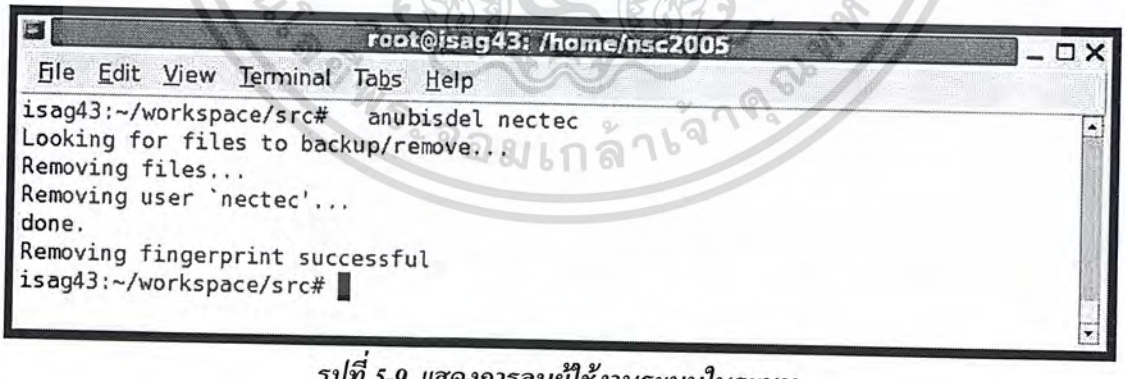
เนื่องจากผู้ใช้งานทั่วไปไม่มีสิทธิความเป็น root ฉะนั้นจึงไม่สามารถทำการเพิ่มผู้ใช้งานระบบได้ การทดสอบเป็นดังนี้

1. ผู้ใช้ทั่วไปทำการเพิ่มผู้ใช้งานระบบ
2. ระบบแจ้งบอกว่าไม่สามารถทำการเพิ่มผู้ใช้งานระบบได้

5.3 การทดสอบการลบผู้ใช้งานระบบ (anubisdel)

ในการลบผู้ใช้งานระบบนั้น จะสงวนไว้ให้ผู้ที่มีสิทธิความเป็น root เท่านั้นที่สามารถทำการลบผู้ใช้งานระบบได้ การทดสอบเป็นดังนี้

5.3.1 ผู้ใช้ที่มีสิทธิความเป็น root ทำการลบผู้ใช้งานระบบ



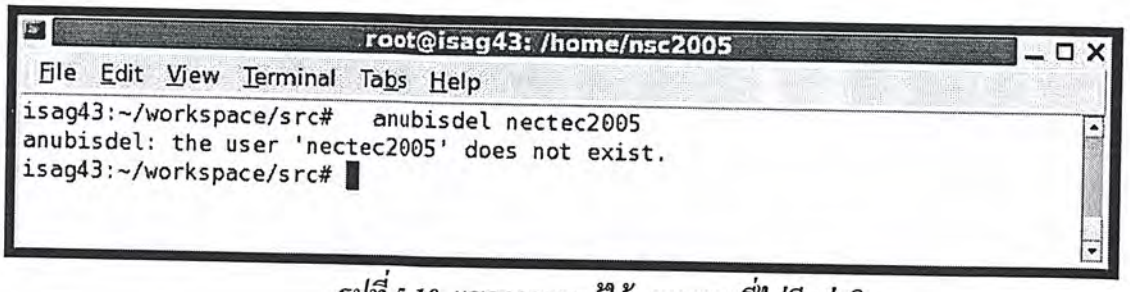
รูปที่ 5-9 แสดงการลบผู้ใช้งานระบบในระบบ

การทดสอบเป็นดังนี้

1. ผู้ใช้ (ที่มีสิทธิความเป็น root) ลบผู้ใช้งานระบบโดยการใช้คำสั่ง anubisdel และป้อนยูสเซอร์เนม

2. การลบผู้ใช้งานระบบเสร็จเรียบร้อย

5.3.2 ผู้ใช้ที่มีสิทธิความเป็นรูททำการลบผู้ใช้งานระบบที่ไม่มีอยู่จริง

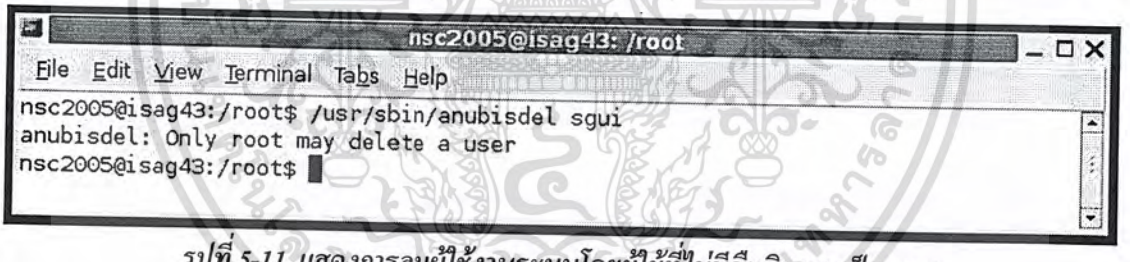


รูปที่ 5-10 แสดงการลบผู้ใช้งานระบบที่ไม่มีอยู่จริง

การทดสอบเป็นดังนี้

1. ผู้ใช้ (ที่มีสิทธิความเป็น root) ลบผู้ใช้งานระบบโดยการใช้คำสั่ง anubisdel และป้อนยูสเซอร์เนม
2. ระบบจะแจ้งบอกว่าไม่สามารถทำการลบผู้ใช้งานระบบนั้นได้

5.3.3 ผู้ใช้งานทั่วไปทำการลบผู้ใช้งานระบบ

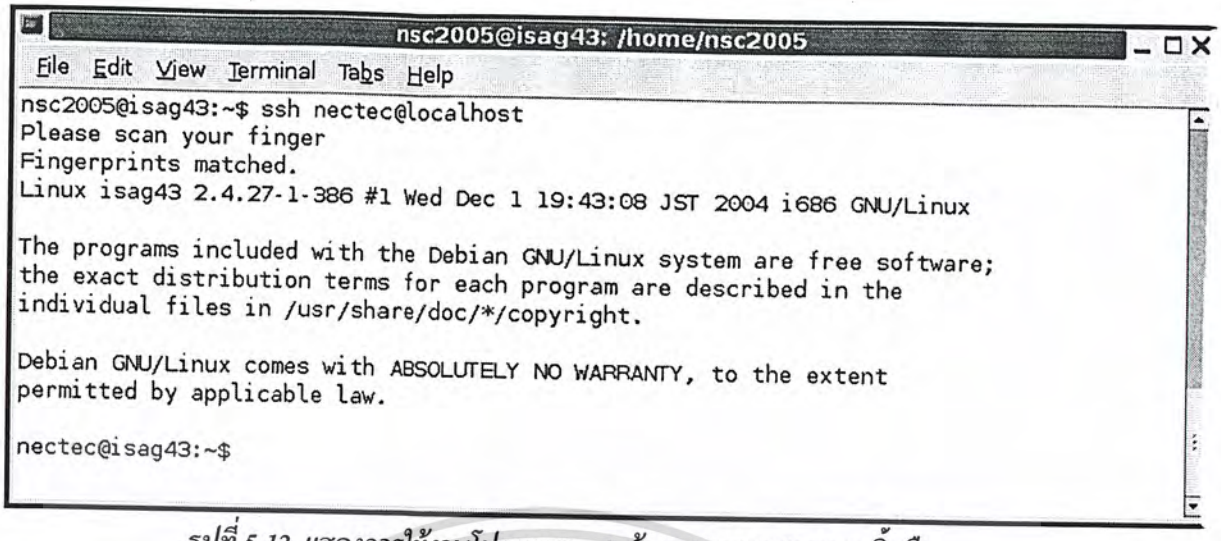


รูปที่ 5-11 แสดงการลบผู้ใช้งานระบบโดยผู้ใช้ที่ไม่มีสิทธิความเป็น root

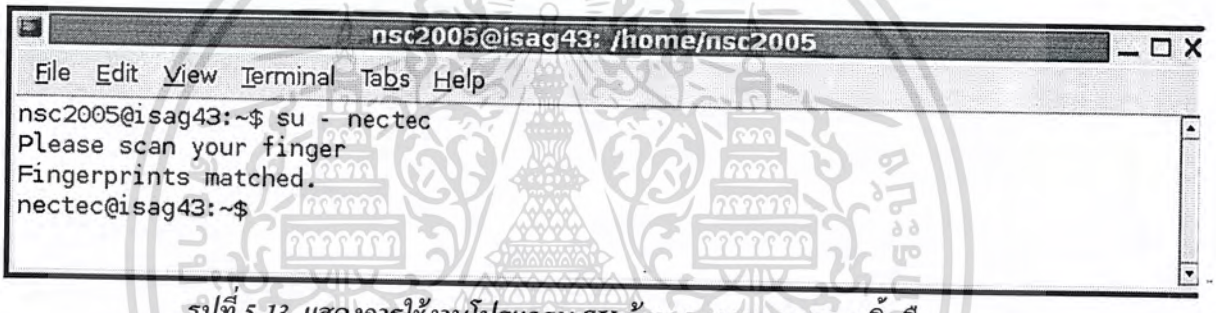
เนื่องจากผู้ใช้งานทั่วไปไม่มีสิทธิความเป็น root ฉะนั้นจึงไม่สามารถทำการลบผู้ใช้งานระบบได้ การทดสอบเป็นดังนี้

1. ผู้ใช้ทั่วไปลบผู้ใช้งานระบบโดยการใช้คำสั่ง anubisdel และป้อนยูสเซอร์เนม
2. ระบบแจ้งบอกว่าไม่สามารถทำการลบผู้ใช้งานระบบได้

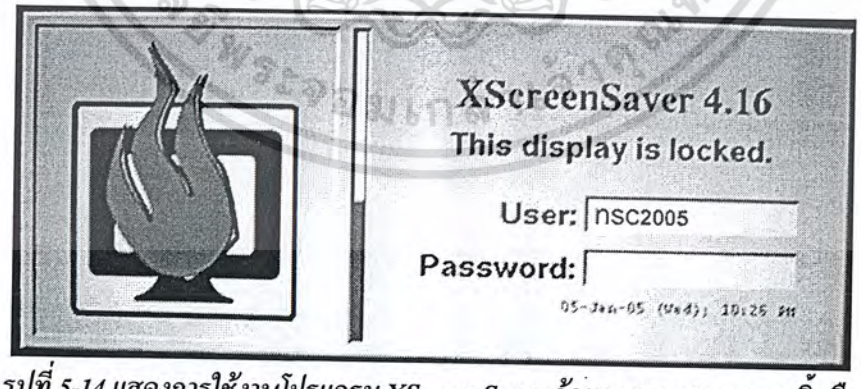
นอกจากการทดสอบ 3 ขั้นตอนหลักดังที่ได้กล่าวมาแล้วนั้น ยังได้ทำการทดลองในส่วนของแอพพลิเคชั่นอื่นๆที่ทำงานร่วมกับโมดูลในการตรวจสอบผู้ใช้งานของระบบปฏิบัติการลีนุกซ์ PAM (Pluggable Authentication Modules) API อีกด้วยได้แก่ โปรแกรมจำพวก ssh, Telnet, xscreensaver, kscreensaver, su และ ftp



รูปที่ 5-12 แสดงการใช้งานโปรแกรม ssh ด้วยการตรวจสอบลายนิ้วมือ



รูปที่ 5-13 แสดงการใช้งานโปรแกรม SU ด้วยการตรวจสอบลายนิ้วมือ



รูปที่ 5-14 แสดงการใช้งานโปรแกรม XScreenSaver ด้วยการตรวจสอบลายนิ้วมือ

บทที่ 6
สรุปและวิจารณ์

6.1 ปัญหาและอุปสรรคในการพัฒนาโครงงาน

1. เนื่องด้วยการพัฒนาโครงการนี้เป็นเทคโนโลยีที่ใหม่ โดยเฉพาะอย่างยิ่งการพัฒนาวิธีชีวมาตร ให้ใช้งานร่วมกับระบบปฏิบัติการลีนุกซ์ ยังไม่แพร่หลาย ส่งผลให้ไม่มีเอกสารอ้างอิงที่อธิบายเกี่ยวข้องกับ API ให้ศึกษามากเท่าใดนัก

2. ระบบนี้พัฒนาขึ้นโดยใช้ภาษาซีบนระบบปฏิบัติการลีนุกซ์จึงพบปัญหาในการเขียนโปรแกรมในส่วนที่เกี่ยวข้องกับ PAM โมดูล และโปรแกรมการล็อกอิน การเพิ่มและลบผู้ใช้งานระบบ

3. ปัญหาเกี่ยวกับผู้ใช้งานระบบเอง คือการสแกนลายนิ้วมือของผู้ใช้งานระบบต้องเป็นการสแกนที่ถูกต้อง การรับข้อมูลทางเครื่องสแกนลายนิ้วมือจึงจะไม่มีข้อผิดพลาด

6.2 แนวทางการแก้ไขปัญหาและอุปสรรค

1. ในส่วนของข้อมูลที่เกี่ยวข้องกับ API ทางผู้พัฒนาได้แก้ไขปัญหาโดยการติดต่อสอบถามจากฝ่ายเทคนิคของผู้จัดจำหน่ายอุปกรณ์สแกนลายนิ้วมือโดยตรง โดยสอบถามทางอีเมล์ และทางโทรศัพท์ทางไกล

2. ในส่วนของการเขียนโปรแกรมบนระบบปฏิบัติการลีนุกซ์ด้วยภาษาซีนั้น ทางผู้วิจัยโครงการได้แก้ไขปัญหาโดยการค้นคว้าหาข้อมูลจากหนังสือต่างๆ (Text book) ค้นคว้าจากอินเทอร์เน็ต และสอบถามจากอาจารย์ที่ปรึกษาโดยตรง

3. เพื่อลดปัญหาในการสแกนลายนิ้วมือ ทางผู้วิจัยโครงการได้จัดทำคู่มือการใช้งานเพื่อเป็นแนวทางในการใช้งาน

6.3 แนวทางการพัฒนาต่อในอนาคต

ระบบต้นแบบในการล็อกอินเข้าลีนุกซ์ด้วยลายนิ้วมือนี้พัฒนาขึ้นเพื่อความปลอดภัยที่มากขึ้นต่อระบบ ทั้งนี้จะมุ่งเน้นในด้านการติดต่อกับระบบปฏิบัติการลีนุกซ์เป็นสำคัญ โดยในส่วนของการพิสูจน์ตนของผู้ใช้งานนั้นจะใช้งาน PAM API เข้ามาพัฒนา ซึ่งเป็นโมดูลในการพิสูจน์ของผู้ใช้งานของระบบปฏิบัติการลีนุกซ์ ดังนั้นทำให้การพัฒนาโปรแกรมด้านแอพพลิเคชั่นนั้นจะมีความยืดหยุ่นมากขึ้น นั่นหมายความว่าสามารถนำไปใช้งานร่วมกับโปรแกรมจำพวก ssh, telnet, xscreensaver และ ftp ได้ นับเป็นการส่งผลให้ในการตรวจสอบผู้ใช้งานของโปรแกรมด้านแอพพลิเคชั่นต่างๆนั้นมีความยืดหยุ่นและปลอดภัยมากยิ่งขึ้น

อีกทั้งระบบต้นแบบนี้ยังสามารถนำมาพัฒนาในส่วนของวิธีชีวมาตรให้ใช้งานร่วมกับระบบปฏิบัติการลีนุกซ์ได้อีกด้วย ไม่ว่าจะเป็นการตรวจสอบเสียง (Voice recognition) ตรวจสอบม่านตา (Eyes scan) ตรวจสอบใบหน้า (Face scan) ตรวจสอบจังหวะในการพิมพ์คีย์บอร์ด (Key stroke) เป็นต้น ซึ่งนับเป็นประโยชน์อย่างยิ่งต่อตัวผู้ใช้งานเองและระบบปฏิบัติการลีนุกซ์อีกด้วย

6.4 ข้อสรุปและข้อเสนอแนะ

ระบบต้นแบบในการล็อกอินเข้าลีนุกซ์ด้วยลายนิ้วมือนี้สามารถทำงานตามวัตถุประสงค์ได้เป็นอย่างดี คือ เน้นความปลอดภัยในการเข้าใช้งานระบบปฏิบัติการลีนุกซ์เป็นสำคัญ และเพิ่มความสะดวกต่อผู้ใช้งานระบบ ทั้งยังเป็นการส่งเสริมวิธีการทางไบโอเมทริกให้ใช้งานร่วมกับระบบปฏิบัติการลีนุกซ์อีกด้วย

และเพื่อให้เกิดประโยชน์มากยิ่งขึ้นสมควรเป็นอย่างยิ่งในการที่จะนำระบบต้นแบบในการล็อกอินเข้าลีนุกซ์ด้วยลายนิ้วมือนี้ไปพัฒนาต่อตามแนวทางในการพัฒนาและประยุกต์ใช้ร่วมกับงานอื่นๆในขั้นต่อไป

บรรณานุกรม

[1] Neil Matthew and Richard Stones : "*Beginning Linux Programming 3rd Edition*", Wiley Publisher,Inc.

[2] W.Richard Stevens : "*Advanced Programming in the UNIX Environment*", Addison-Wesley

เวบไซต์อ้างอิง

[1] "*Biometric and Linux*" :
http://downloads.planetmirror.com/pub/lca/2003/proceedings/papers/Alexander_Reeder/Alexander_Reeder.pdf

[2] "*Biometric and Linux*" :
http://www.linux.org.au/conf/2003/abstracts.cgi_id=P02.html

[3] "*Digital Persona, Inc.*" :
http://www.comptalk.com/UareUPro.pdf

[4] "*Biometrika : Security and Personal Identification Systems*" :
http://www.biometrika.it/eng/

[5] "*User Authentication HOWTO*" :
http://www.ibiblio.org/pub/Linux/docs/HOWTO/other-formats/pdf/User-Authentication-HOWTO.pdf

[6] "*The Linux-PAM System Administrators' Guide*" :
http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam.html

[7] "*The Linux-PAM Module Writers' Guide*" :
http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam_modules.html

[8] "*The Linux-PAM Application Developers' Guide*" :
http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam_appl.html

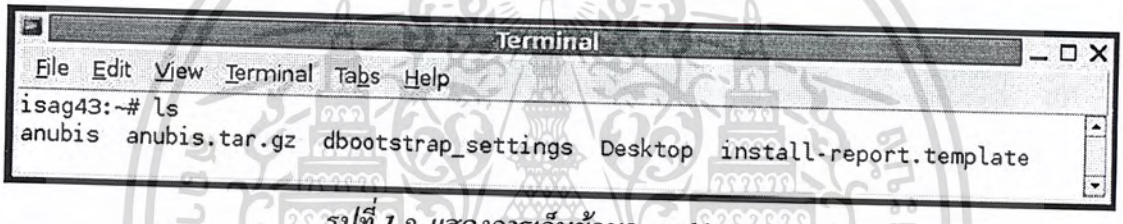
[9] "*Neurotechnologija offices and distributors*" :
http://www.neurotechnologija.com/vf_sdk.html

ภาคผนวก ก.
คู่มือการติดตั้ง

ระบบต้นแบบในการล็อกอินเข้าลีนุกซ์ด้วยลายนิ้วมือนี้ถูกพัฒนาขึ้นตามวัตถุประสงค์ที่มุ่งเน้นความปลอดภัยในการเข้าใช้งานระบบปฏิบัติการลีนุกซ์เป็นสำคัญ และเพิ่มความสะดวกต่อผู้ใช้งานระบบทั้งยังเป็นการส่งเสริมวิธีการทางไบโอเมตริกให้ใช้งานร่วมกับระบบปฏิบัติการลีนุกซ์อีกด้วย

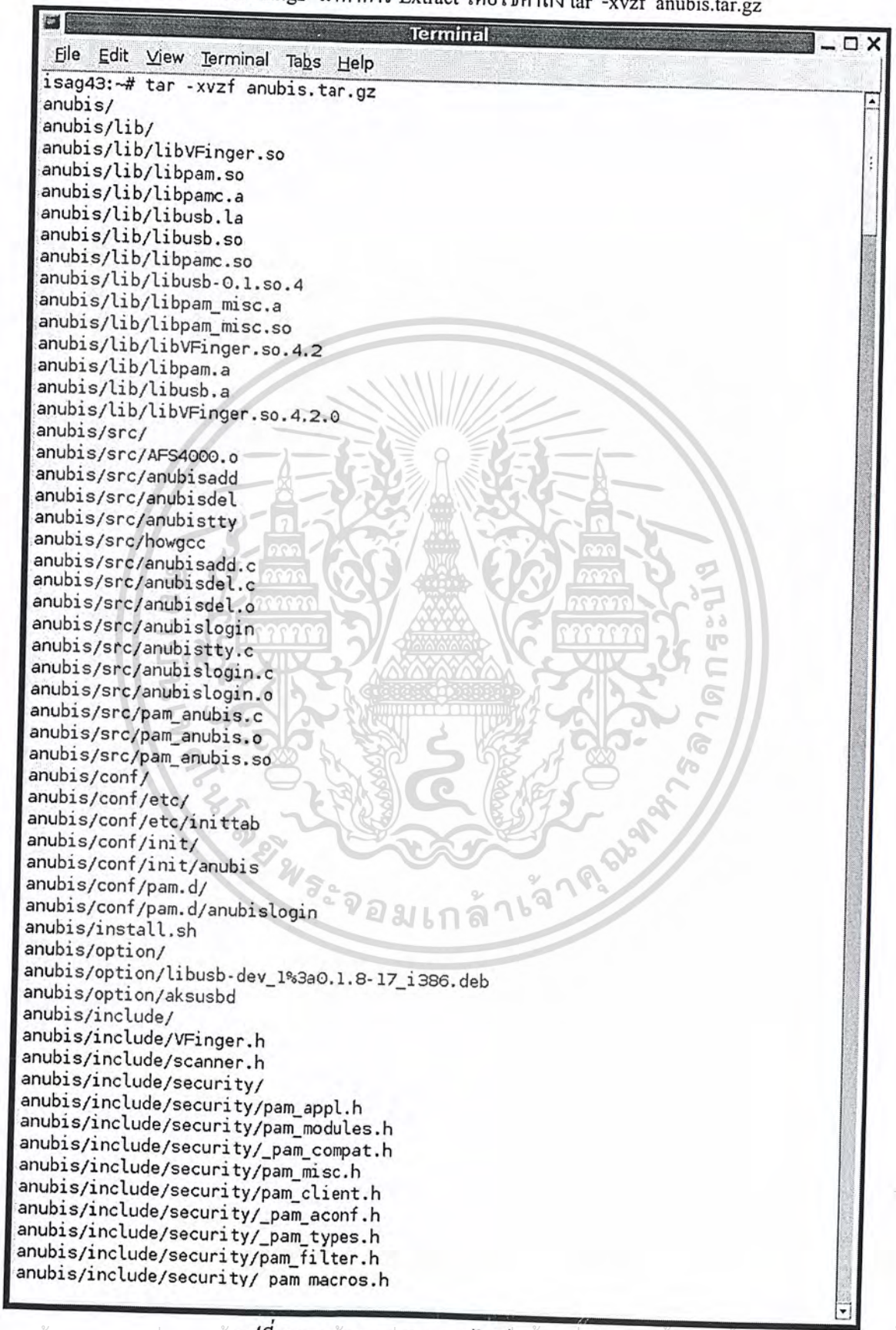
การติดตั้งระบบต้นแบบในการล็อกอินเข้าลีนุกซ์ด้วยลายนิ้วมือนี้มีขั้นตอนในการติดตั้ง ดังนี้

1. ต้องนำไฟล์ "anubis.tar.gz" ไปเก็บไว้ในไดเรคทอรี่ใดๆ ของผู้ใช้ที่มีสิทธิความเป็น root



รูปที่ 1-ก แสดงการเก็บข้อมูล anubis.tar.gz

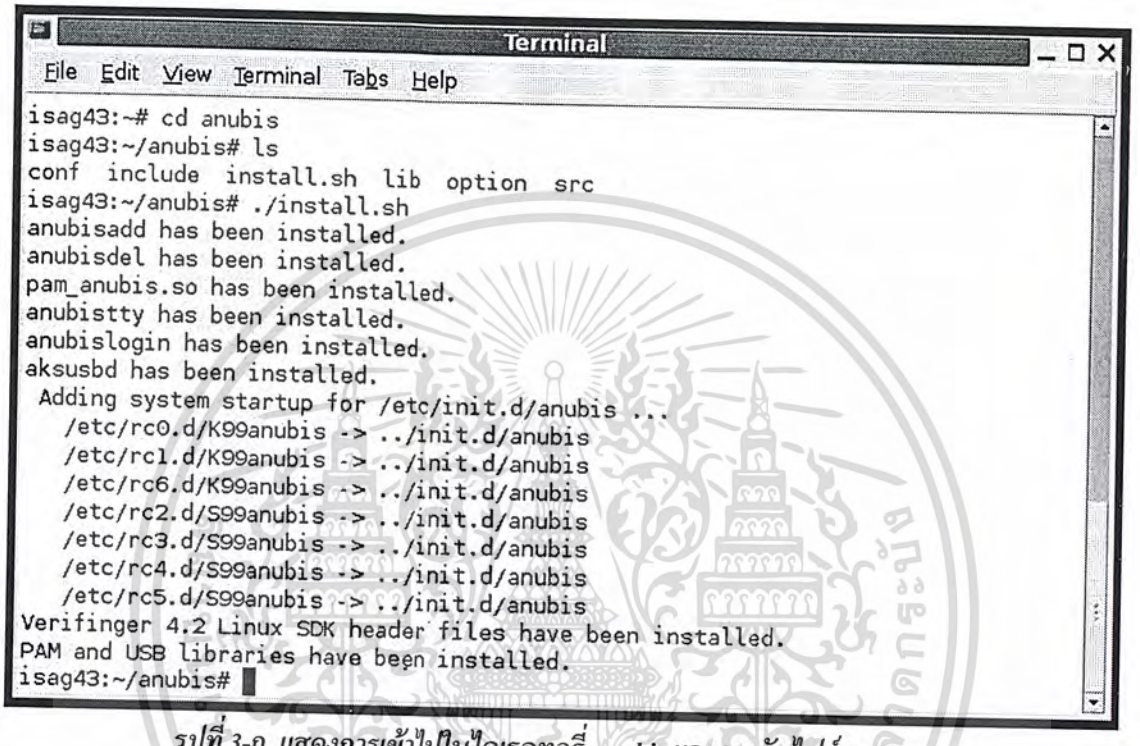
2. นำไฟล์ “anubis.tar.gz” มาทำการ Extract โดยใช้คำสั่ง tar -xvzf anubis.tar.gz



รูปที่ 2-ก แสดงการ Extract ไฟล์ anubis.tar.gz

3. หลังจากได้ทำการ Extract ไฟล์ anubis.tar.gz แล้วจะพบว่ามีไดเรคทอรี่ anubis อยู่ ให้ทำการเข้าไปในไดเรคทอรีนั้น ด้วยคำสั่ง cd anubis

4.ภายในไดเรคทอรี anubis มีไฟล์ "install.sh" อยู่ ให้ทำการรันไฟล์นั้นด้วยคำสั่ง ./install.sh



รูปที่ 3-ก แสดงการเข้าไปในไดเรคทอรี่ anubis และการรันไฟล์ install.sh

5. ทั้งนี้เมื่อรันไฟล์ install.sh เรียบร้อยแล้ว ระบบจะแจ้งให้ผู้ใช้ทราบว่าการติดตั้งระบบต้นแบบในการล็อกอินเข้าลีนุกซ์ด้วยลายนิ้วมือเสร็จเรียบร้อยแล้ว

6. ทำการติดตั้งอุปกรณ์การสแกนลายนิ้วมือเข้าเครื่องคอมพิวเตอร์

เพียงขั้นตอนเพียงเท่านี้การติดตั้งระบบต้นแบบในการล็อกอินเข้าลีนุกซ์ด้วยลายนิ้วมือก็เป็นอันเสร็จสมบูรณ์

ภาคผนวก ข.

คู่มือการใช้งาน

ระบบต้นแบบในการล็อกอินเข้าลีนุกซ์ด้วยลายนิ้วมือนี้ถูกพัฒนาขึ้นตามวัตถุประสงค์ที่มุ่งเน้นความปลอดภัยในการเข้าใช้งานระบบปฏิบัติการลีนุกซ์เป็นสำคัญ และเพิ่มความสะดวกต่อผู้ใช้งานระบบทั้งยังเป็นการส่งเสริมวิธีการทางไบโอเมตริกให้ใช้งานร่วมกับระบบปฏิบัติการลีนุกซ์อีกด้วย

การทำงานของระบบต้นแบบในการล็อกอินเข้าลีนุกซ์ด้วยลายนิ้วมือแบ่งออกเป็นสองส่วนการทำงานหลัก ได้แก่ ทางด้านซอฟต์แวร์ และฮาร์ดแวร์ ซึ่งในแต่ละส่วนจะมีรายละเอียดดังที่จะกล่าวต่อไป

ส่วนของซอฟทต์แวร์มีวิธีการใช้งานดังนี้

การใช้งานระบบต้นแบบในการล็อกอินเข้าลีนุกซ์ด้วยลายนิ้วมือนี้มีขั้นตอนในการใช้งาน แบ่งออกเป็นสามขั้นตอนหลักได้แก่

1. การล็อกอิน (login)
2. การเพิ่มผู้ใช้งานระบบ (add user)
3. การลบผู้ใช้งานระบบ (delete user)

ทั้งนี้ก่อนการใช้งานระบบต้นแบบในการล็อกอินเข้าลีนุกซ์ด้วยลายนิ้วมือจะต้องมีการแก้ไขข้อมูลในไฟล์ PAM Configuration ก่อนเพื่อเป็นการเปลี่ยนระบบจากการล็อกอินด้วยการป้อนข้อมูลยูสเซอร์เนมและรหัสผ่าน มาเป็นการสแกนลายนิ้วมือ โดยขั้นตอนคือ

1. การแก้ไขข้อมูลในไฟล์ /etc/pam.d/common-auth
2. โดยทำการเพิ่มข้อมูล auth required pam_anubis.so ลงไป
3. ทำการคอมเมนท์ที่ @include common-auth โดยทำการใส่ # หน้าประโยค ดังรูป

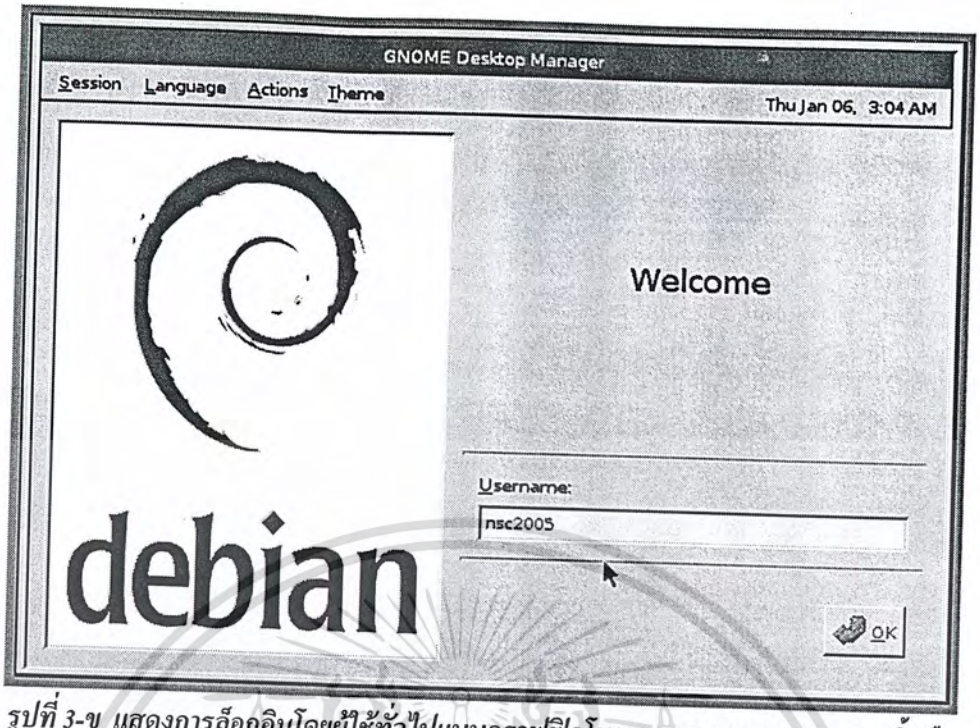
```
# This module parses /etc/environment (the standard for setting
# environ vars) and also allows you to use an extended config
# file /etc/security/pam_env.conf.
# (Replaces the `ENVIRON_FILE' setting from login.defs)
auth       required   pam_env.so

# Standard Un*x authentication. The "nullok" line allows passwordless
# accounts.
#@include common-auth
auth    required        pam_anubis.so

# This allows certain extra groups to be granted to a user
# based on things like time of day, tty, service, and user.
# Please uncomment and edit /etc/security/group.conf if you
# wish to use this.
# (Replaces the `CONSOLE_GROUPS' option in login.defs)
# auth       optional   pam_group.so
```

รูปที่ 1-ข แสดงการแก้ไขไฟล์ PAM Configuration

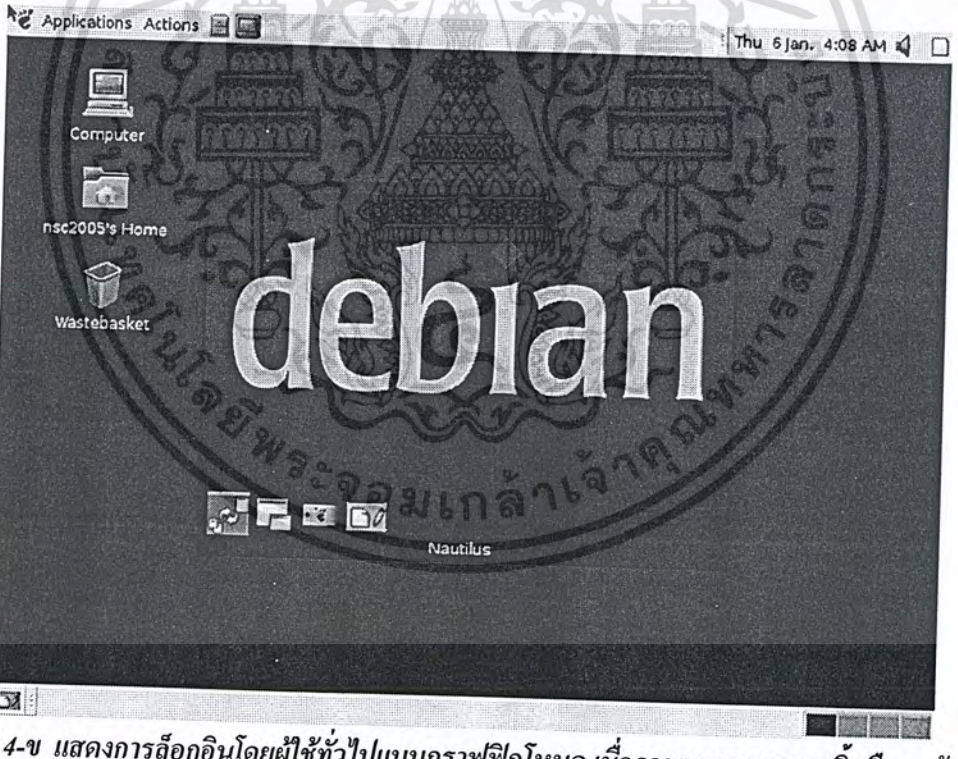
1.การล็อกอิน

1.1 การล็อกอินปกติทั่วไป

รูปที่ 2-ข แสดงการล็อกอินโดยผู้ใช้ทั่วไปแบบกราฟฟิกโหมด



รูปที่ 3-ข แสดงการล็อกอินโดยผู้ใช้ทั่วไปแบบกราฟฟิกโหมด ขณะรอการสแกนลายนิ้วมือ

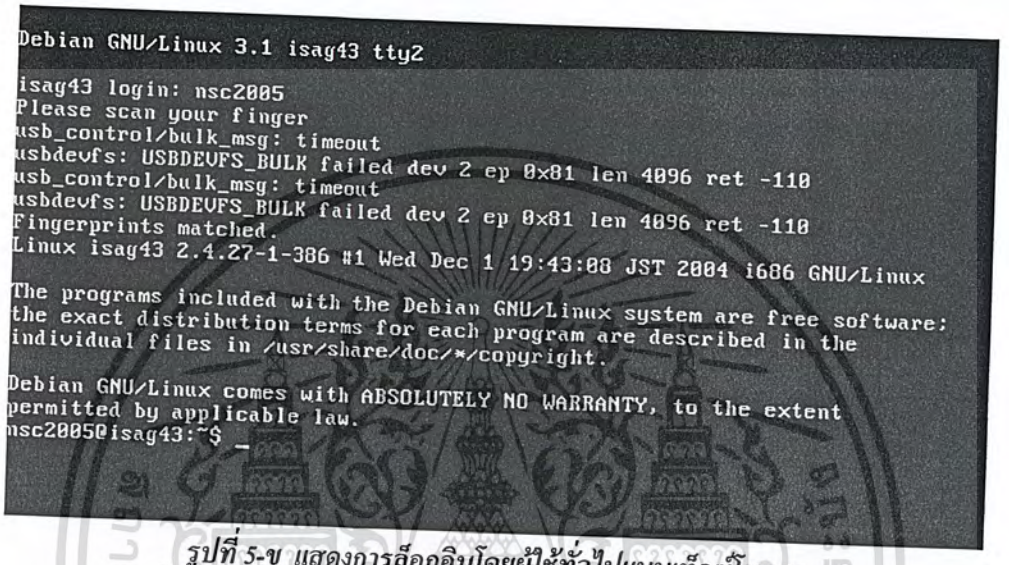


รูปที่ 4-ข แสดงการล็อกอินโดยผู้ใช้ทั่วไปแบบกราฟฟิกโหมด เมื่อการตรวจสอบลายนิ้วมือถูกต้อง

การล็อกอินแบบกราฟฟิกโหมด

ในการล็อกอินของผู้ใช้งานแบบปกติทั่วไปนั้น ซึ่งมีขั้นตอนในการล็อกอินดังนี้

1. ผู้ใช้ทำการป้อนข้อมูลยูสเซอร์เนมของตน
2. ผู้ใช้ทำการสแกนลายนิ้วมือของตนเพื่อพิสูจน์ตน
3. ผลการตรวจสอบลายนิ้วมือถูกต้อง
4. ผู้ใช้สามารถเข้าใช้งานระบบได้ตามปกติ



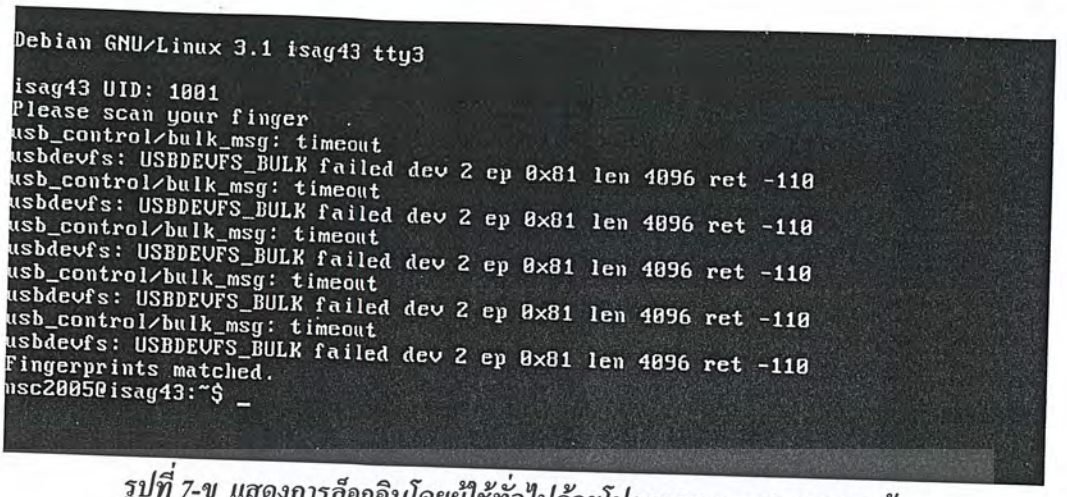
รูปที่ 5-ข แสดงการล็อกอินโดยผู้ใช้ทั่วไปแบบเท็กซ์โหมด

การล็อกอินแบบเท็กซ์โหมด

ในการล็อกอินของผู้ใช้งานแบบปกติทั่วไปนั้น ซึ่งมีขั้นตอนในการล็อกอินดังนี้

1. ผู้ใช้ทำการป้อนข้อมูลยูสเซอร์เนมของตน
2. ผู้ใช้ทำการสแกนลายนิ้วมือของตนเพื่อพิสูจน์ตน
3. ผลการตรวจสอบลายนิ้วมือถูกต้อง
4. ผู้ใช้สามารถเข้าใช้งานระบบได้ตามปกติ

รูปที่ 6-ข แสดงการล็อกอินโดยผู้ใช้ทั่วไปด้วยโปรแกรม anubis



รูปที่ 7-ข แสดงการล็อกอินโดยผู้ใช้ทั่วไปด้วยโปรแกรม anubis แบบถูกต้อง

การล็อกอินด้วยโปรแกรม anubislogin

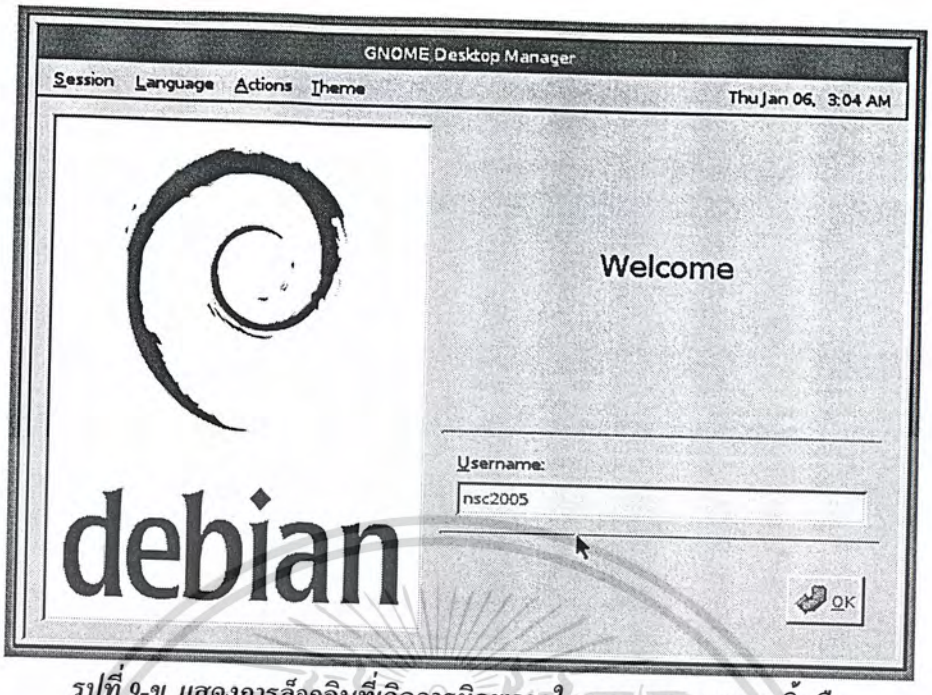
ในการล็อกอินของผู้ใช้งานแบบปกติทั่วไปนั้น ซึ่งมีขั้นตอนในการล็อกอินดังนี้

1. ผู้ใช้ทำการป้อนข้อมูล UID ของตน
2. ผู้ใช้ทำการสแกนลายนิ้วมือของตนเพื่อพิสูจน์ตน
3. ผลการตรวจสอบลายนิ้วมือถูกต้อง
4. ผู้ใช้สามารถเข้าใช้งานระบบได้ตามปกติ

1.2 การล็อกอินที่เกิดการผิดพลาดในการตรวจสอบลายนิ้วมือ

ในกรณีที่ผู้ใช้สแกนลายนิ้วมือไม่ดีพอ ซึ่งอาจเกิดจากการวางนิ้วมือไม่ตรง การวางนิ้วมือเบาเกินไป อาจทำให้เกิดข้อผิดพลาดในการสแกนลายนิ้วมือได้ ซึ่งมีขั้นตอนดังนี้

รูปที่ 8-ข แสดงการล็อกอินที่เกิดการผิดพลาดแบบกราฟฟิกโหมด



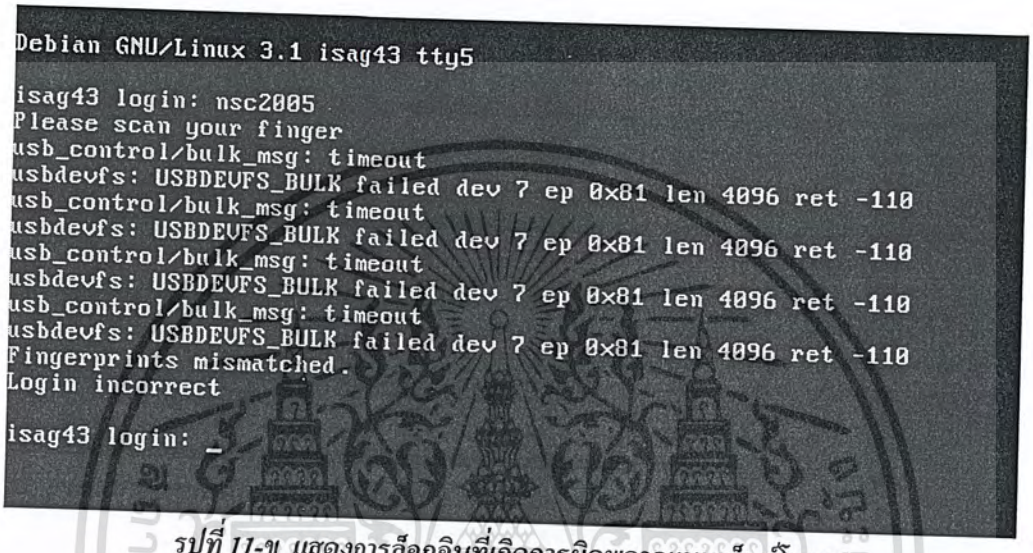
รูปที่ 9-ข แสดงการล็อกอินที่เกิดการผิดพลาดในการตรวจสอบลายนิ้วมือ

รูปที่ 10-ข แสดงหน้าจอรอรับการป้อนข้อมูลจากผู้ใช้งานระบบ

การล็อกอินแบบกราฟฟิกโหมด

ในการล็อกอินของผู้ใช้งานแบบปกติทั่วไปนั้น ซึ่งมีขั้นตอนในการล็อกอินดังนี้

1. ผู้ใช้ทำการป้อนข้อมูลยูสเซอร์เนมของตน
2. ผู้ใช้ทำการสแกนลายนิ้วมือของตนเพื่อพิสูจน์ตน
3. ผลการตรวจสอบลายนิ้วมือไม่ถูกต้อง
4. ระบบกลับมาอยู่หน้าที่ให้ผู้ใช้ป้อนข้อมูลยูเซอร์เนมอีกครั้ง

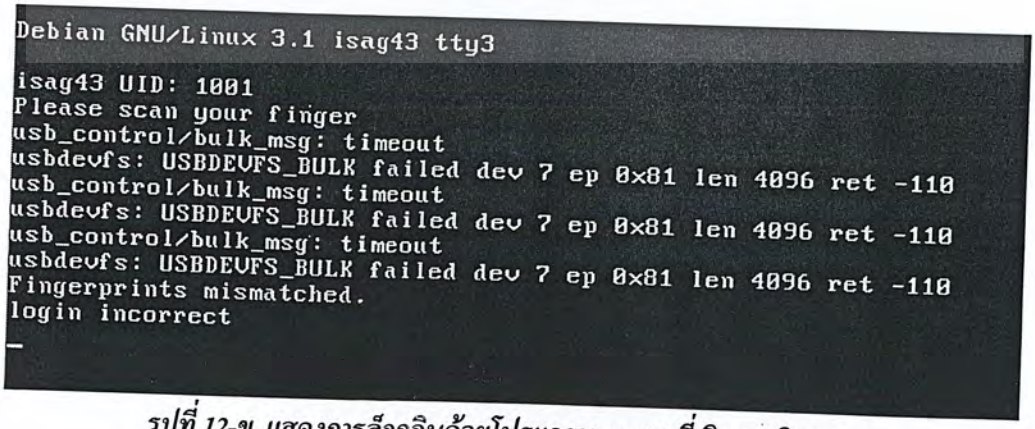


รูปที่ 11-ข แสดงการล็อกอินที่เกิดการผิดพลาดแบบเท็กซ์โหมด

การล็อกอินแบบเท็กซ์โหมด

ในการล็อกอินของผู้ใช้งานแบบปกติทั่วไปนั้น ซึ่งมีขั้นตอนในการล็อกอินดังนี้

1. ผู้ใช้ทำการป้อนข้อมูลยูสเซอร์เนมของตน
2. ผู้ใช้ทำการสแกนลายนิ้วมือของตนเพื่อพิสูจน์ตน
3. ผลการตรวจสอบลายนิ้วมือไม่ถูกต้อง
4. ระบบกลับมาอยู่หน้าที่ให้ผู้ใช้ป้อนข้อมูลยูเซอร์เนมอีกครั้ง



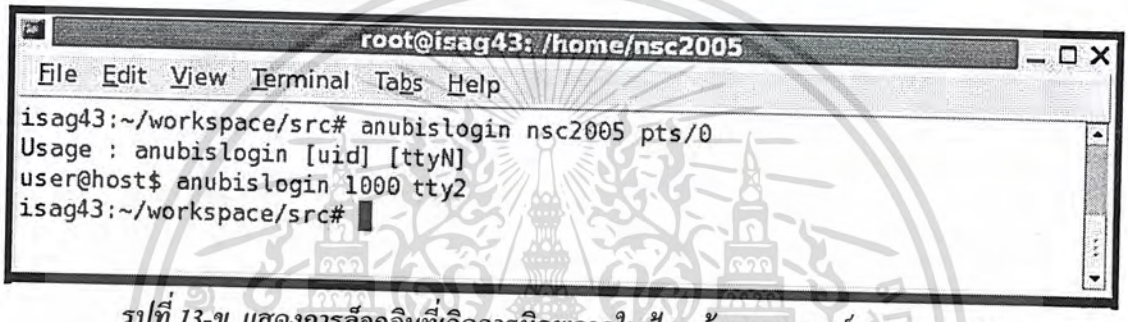
รูปที่ 12-ข แสดงการล็อกอินด้วยโปรแกรม anubisที่เกิดการผิดพลาด

การล็อกอินด้วยโปรแกรม anubislogin

ในการล็อกอินของผู้ใช้งานแบบปกติทั่วไปนั้น ซึ่งมีขั้นตอนในการล็อกอินดังนี้

1. ผู้ใช้ทำการป้อนข้อมูล UID ของตน
2. ผู้ใช้ทำการสแกนลายนิ้วมือของตนเพื่อพิสูจน์ตน
3. ผลการตรวจสอบลายนิ้วมือถูกต้อง
4. ผู้ใช้สามารถเข้าใช้งานระบบได้ตามปกติ

1.3 การล็อกอินโดยการใส่ยูสเซอร์เนมแทน UID

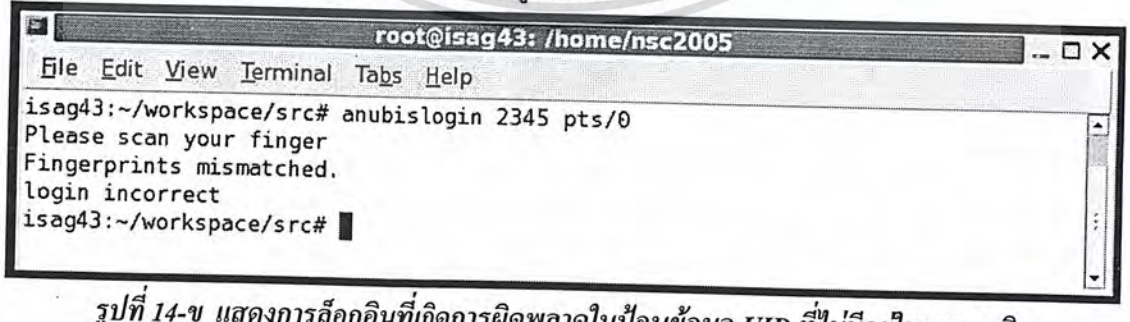


รูปที่ 13-ข แสดงการล็อกอินที่เกิดการผิดพลาดในป้อนข้อมูลยูสเซอร์เนมแทน UID

ในกรณีที่การล็อกอินมีการใส่ยูสเซอร์เนมแทน UID นั้นระบบจะไม่อนุญาติให้ล็อกอินได้ ซึ่งมีขั้นตอนดังนี้

1. ผู้ใช้ล็อกอินโดยการใช้คำสั่ง anubislogin และป้อนข้อมูลยูสเซอร์เนมและ ttyn ของตน
2. ระบบแจ้งบอกให้ผู้ใช้ทราบถึงวิธีการป้อนข้อมูลที่ถูกต้องในการล็อกอิน พร้อมยกตัวอย่าง
3. ผู้ใช้ไม่สามารถเข้าใช้งานระบบได้

1.4 การล็อกอินโดยการใส่ UID ที่ไม่มีอยู่จริงในระบบ



รูปที่ 14-ข แสดงการล็อกอินที่เกิดการผิดพลาดในป้อนข้อมูล UID ที่ไม่มีอยู่ในระบบจริง

ในกรณีที่การล็อกอินมีการใส่ UID ที่ไม่มีอยู่จริงในระบบ จะไม่สามารถทำการล็อกอินได้ ซึ่งมีขั้นตอนดังนี้

1. ผู้ใช้ล็อกอินโดยการใช้คำสั่ง anubislogin และป้อนข้อมูล UID และ ttyn ของตน

2. ระบบแจ้งบอกให้ผู้ใช้ทำการสแกนลายนิ้วมือ

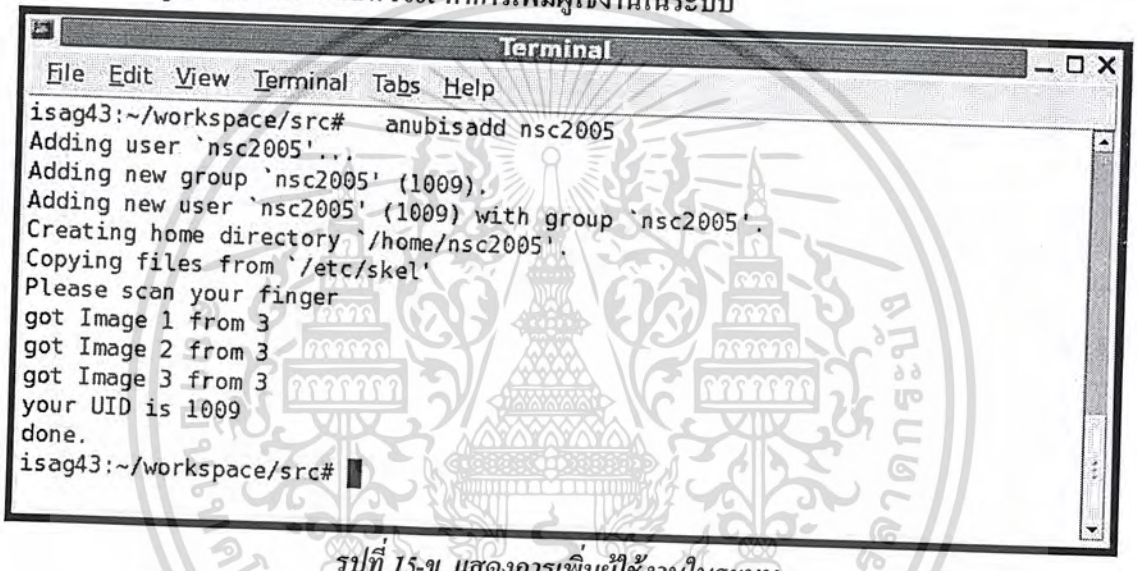
3. หลังจากที่ผู้ใช้สแกนลายนิ้วมือแล้ว ระบบจะแจ้งบอกว่าผลการตรวจสอบลายนิ้วมือผิดพลาด

4. ผู้ใช้ไม่สามารถเข้าใช้งานระบบได้

2. การเพิ่มผู้ใช้งานระบบ (anubisdel)

ในการเพิ่มผู้ใช้งานในระบบนั้น ระบบจะสงวนไว้ให้ผู้ที่มีสิทธิความเป็น root เท่านั้นที่สามารถทำการเพิ่มผู้ใช้งานในระบบได้

2.1 ผู้ใช้ที่มีสิทธิความเป็น root ทำการเพิ่มผู้ใช้งานในระบบ



รูปที่ 15-ข แสดงการเพิ่มผู้ใช้งานในระบบ

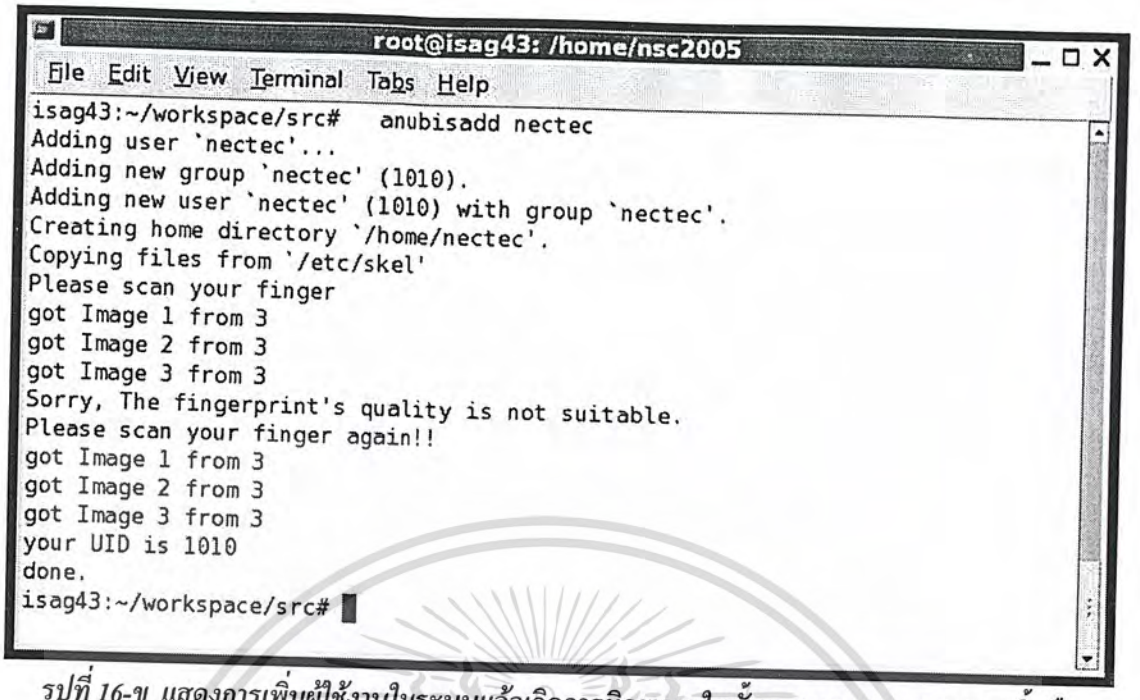
การเพิ่มผู้ใช้งานในระบบมีขั้นตอนดังนี้

1. ผู้ใช้ (ที่มีสิทธิความเป็น root) เพิ่มผู้ใช้งานในระบบโดยการใช้คำสั่ง anubisadd และป้อนยูสเซอร์เนม

2. ระบบแจ้งบอกให้ผู้ใช้ทำการสแกนลายนิ้วมือ 3 ครั้ง

3. การเพิ่มผู้ใช้งานในระบบเสร็จเรียบร้อย

2.2 ผู้ใช้ที่มีสิทธิความเป็น root ทำการเพิ่มผู้ใช้งานในระบบแล้วเกิดการผิดพลาดในขั้นตอนการสแกนลายนิ้วมือ

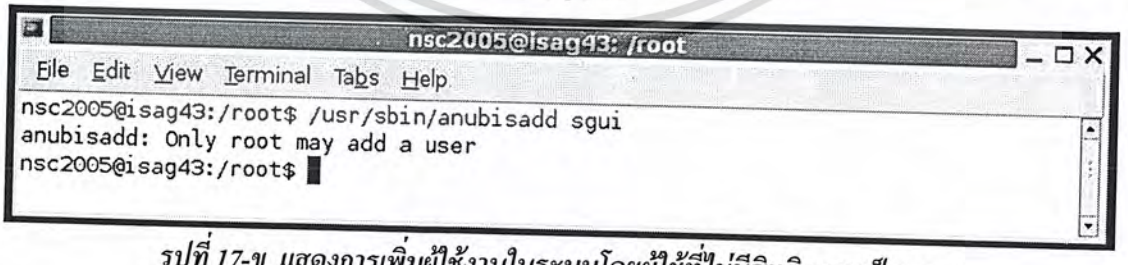


รูปที่ 16-ข แสดงการเพิ่มผู้ใช้งานในระบบแล้วเกิดการผิดพลาดในขั้นตอนของการสแกนลายนิ้วมือ

ซึ่งมีขั้นตอนดังนี้

1. ผู้ใช้ (ที่มีสิทธิความเป็น root) เพิ่มผู้ใช้งานในระบบโดยการใช้คำสั่ง anubisadd และป้อนยูสเซอร์เนม
2. ระบบแจ้งบอกให้ผู้ใช้ทำการสแกนลายนิ้วมือ 3 ครั้ง
3. หลังจากทำการสแกนลายนิ้วมือครบ 3 ครั้งแล้ว เกิดข้อผิดพลาดในการสแกนลายนิ้วมือขึ้น
4. ระบบจะแจ้งบอกให้ผู้ใช้ทำการสแกนลายนิ้วมือใหม่อีก 3 ครั้ง
5. การเพิ่มผู้ใช้งานในระบบเสร็จเรียบร้อย

2.3 ผู้ใช้งานทั่วไปทำการเพิ่มผู้ใช้งานในระบบ



รูปที่ 17-ข แสดงการเพิ่มผู้ใช้งานในระบบโดยผู้ใช้ที่ไม่มีสิทธิความเป็น root

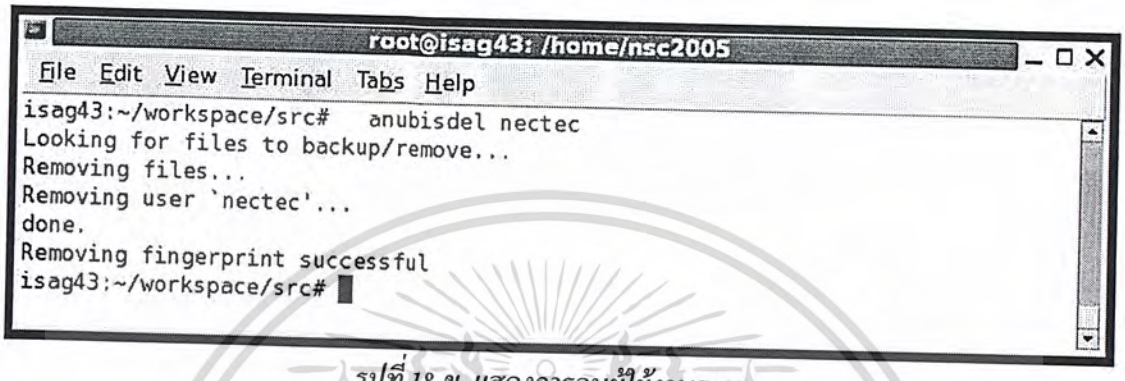
เนื่องจากผู้ใช้งานทั่วไปไม่มีสิทธิความเป็น root ฉะนั้นจึงไม่สามารถทำการเพิ่มผู้ใช้งานในระบบได้ ซึ่งมีขั้นตอนดังนี้

1. ผู้ใช้ทั่วไปทำการเพิ่มผู้ใช้งานในระบบ
2. ระบบแจ้งบอกว่าไม่สามารถทำการเพิ่มผู้ใช้งานในระบบได้

3. การลบผู้ใช้งานระบบ (anubisdel)

ในการลบผู้ใช้งานระบบ ในระบบนั้น ระบบจะสงวนไว้ให้ผู้ที่มีสิทธิความเป็น root เท่านั้นที่สามารถทำการลบผู้ใช้งานระบบ ได้ การทดสอบเป็นดังนี้

3.1 ผู้ใช้ที่มีสิทธิความเป็น root ทำการลบผู้ใช้งานระบบ



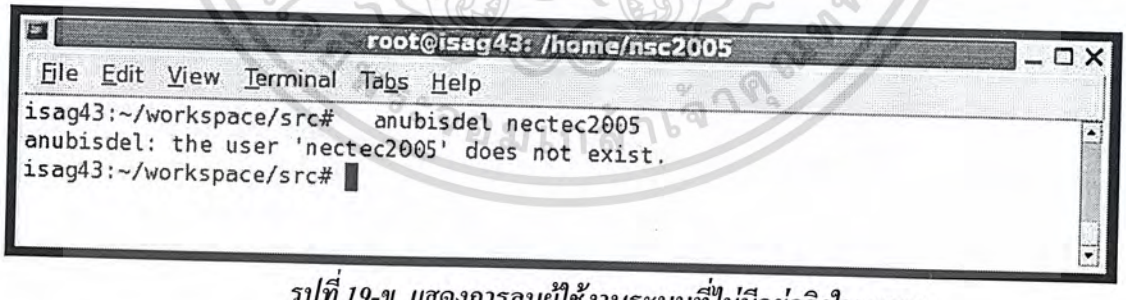
รูปที่ 18-ข แสดงการลบผู้ใช้งานระบบ

ซึ่งมีขั้นตอนดังนี้

1. ผู้ใช้ (ที่มีสิทธิความเป็น root) ลบผู้ใช้งานระบบโดยการใช้คำสั่ง anubisdel และป้อนยูสเซอร์เนม

2. การลบผู้ใช้งานระบบเสร็จเรียบร้อย

3.2 ผู้ใช้ที่มีสิทธิความเป็น root ทำการลบผู้ใช้งานระบบ ที่ไม่มีอยู่จริงในระบบ



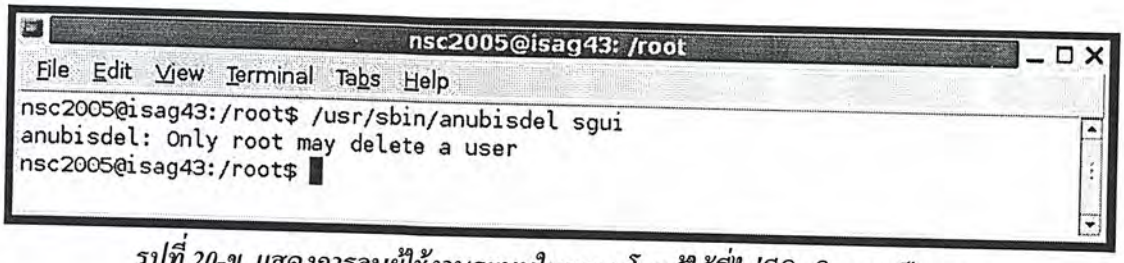
รูปที่ 19-ข แสดงการลบผู้ใช้งานระบบที่ไม่มีอยู่จริงในระบบ

ซึ่งมีขั้นตอนดังนี้

1. ผู้ใช้ (ที่มีสิทธิความเป็น root) ลบผู้ใช้งานระบบโดยการใช้คำสั่ง anubisdel และป้อนยูสเซอร์เนม

2. ระบบจะแจ้งบอกว่าไม่สามารถทำการลบผู้ใช้งานระบบนั้นได้

3.3 ผู้ใช้งานทั่วไปทำการลบผู้ใช้งานระบบ

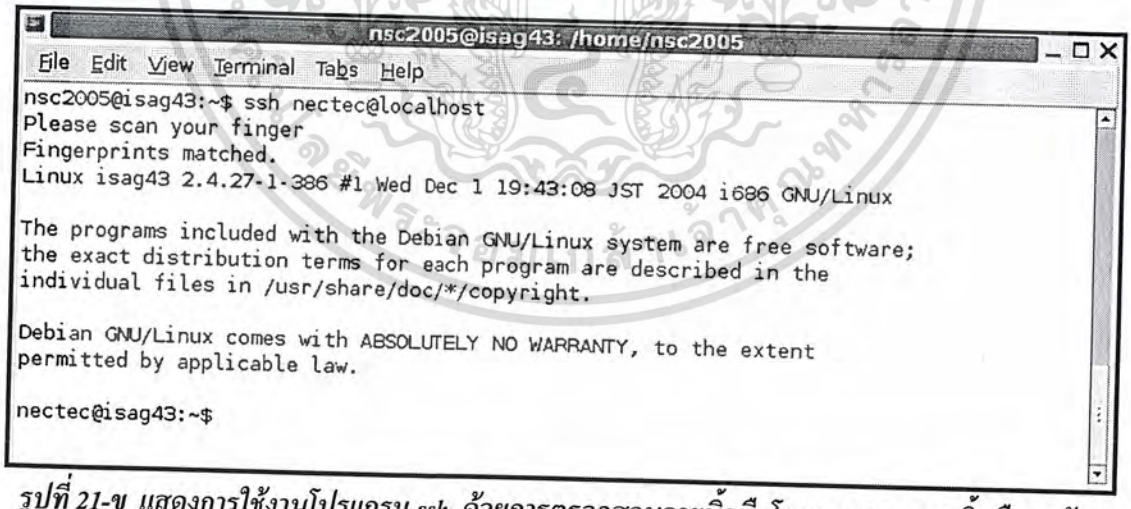


รูปที่ 20-ข แสดงการลบผู้ใช้งานระบบในระบบโดยผู้ใช้ที่ไม่มีสิทธิความเป็น root

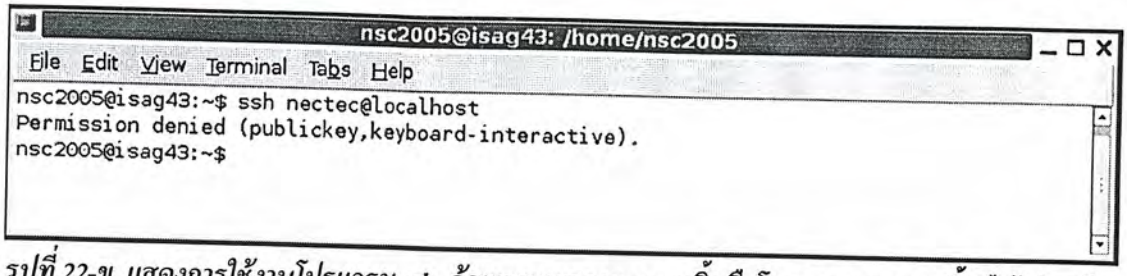
เนื่องจากผู้ใช้งานทั่วไปไม่มีสิทธิความเป็น root ฉะนั้นจึงไม่สามารถทำการลบผู้ใช้งานระบบได้ ซึ่งมีขั้นตอนดังนี้

1. ผู้ใช้ทั่วไปลบผู้ใช้งานระบบโดยการใช้คำสั่ง anubisdel และป้อนยูสเซอร์เนม
2. ระบบแจ้งบอกว่าไม่สามารถทำการลบผู้ใช้งานระบบได้

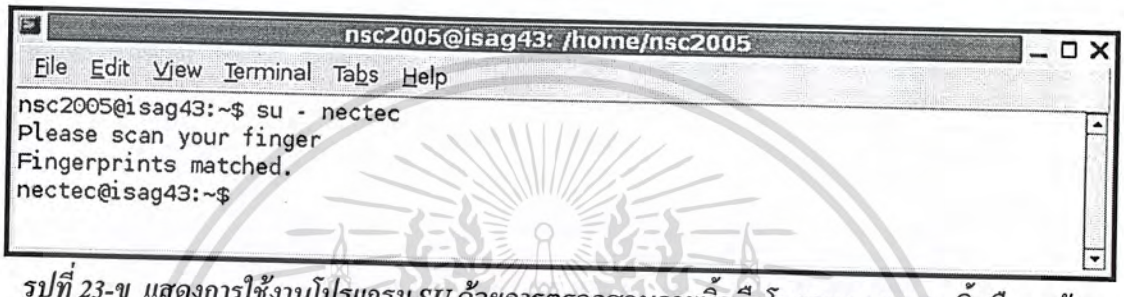
นอกจากการทำงานใน 3 ขั้นตอนหลักดังที่ได้กล่าวมาแล้วนั้น ระบบต้นแบบในการล็อกอินเข้าลีนุกซ์ด้วยลายนิ้วมือยังสามารถทำวานในส่วนของแอพพลิเคชั่นอื่นๆที่ทำงานร่วมกับ PAM (Pluggable Authentication Modules) API ซึ่งเป็น โมดูลในการตรวจสอบผู้ใช้งานของระบบปฏิบัติการลีนุกซ์ อีกด้วย ได้แก่ โปรแกรมจำพวก ssh, Telnet, xscreensaver, kscreensaver, su และ ftp



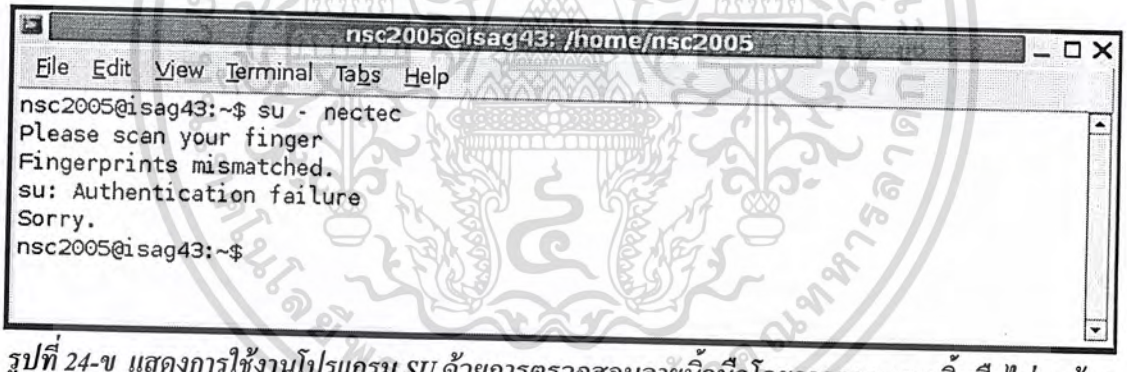
รูปที่ 21-ข แสดงการใช้งานโปรแกรม ssh ด้วยการตรวจสอบลายนิ้วมือโดยการตรวจลายนิ้วมือถูกต้อง



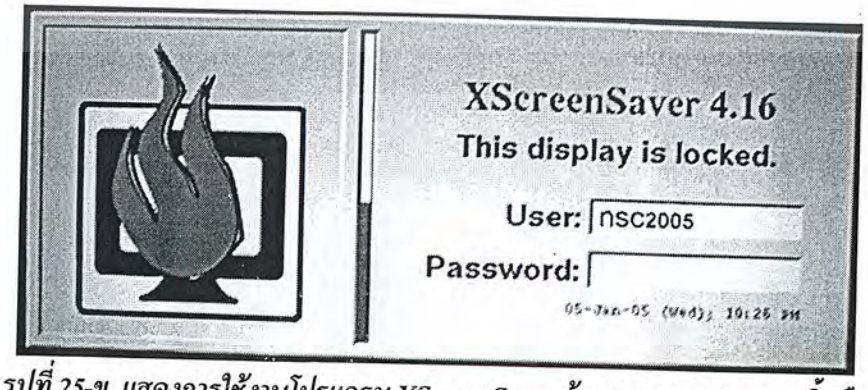
รูปที่ 22-ข แสดงการใช้งานโปรแกรม ssh ด้วยการตรวจสอบลายนิ้วมือโดยการตรวจลายนิ้วมือไม่ถูกต้อง



รูปที่ 23-ข แสดงการใช้งานโปรแกรม SU ด้วยการตรวจสอบลายนิ้วมือโดยการตรวจลายนิ้วมือถูกต้อง



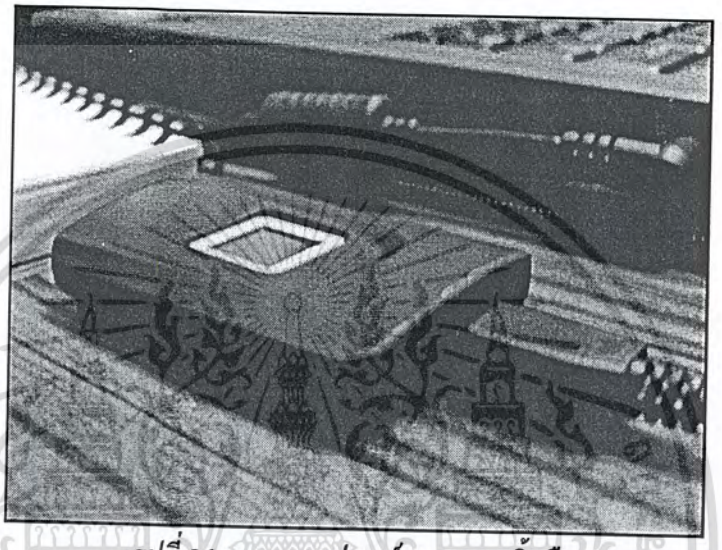
รูปที่ 24-ข แสดงการใช้งานโปรแกรม SU ด้วยการตรวจสอบลายนิ้วมือโดยการตรวจลายนิ้วมือไม่ถูกต้อง



รูปที่ 25-ข แสดงการใช้งานโปรแกรม XScreenSaver ด้วยการตรวจสอบลายนิ้วมือ

ส่วนของฮาร์ดแวร์มีวิธีการใช้งานดังนี้

ในการทำการสแกนลายนิ้วมือของผู้ใช้งานระบบนั้นจะต้องเป็นการสแกนลายนิ้วมือที่ถูกต้อง คือ การวางนิ้วมือทาบลงบนอุปกรณ์สแกนลายนิ้วมือด้วยน้ำหนักที่พอดี ไม่ควรมากหรือน้อยเกินไป เพื่อความชัดเจนของลายนิ้วมือที่จะสแกนออกมาได้



รูปที่ 26-ข แสดงอุปกรณ์สแกนลายนิ้วมือ

รูปที่ 27-ข แสดงการวางนิ้วมือบนอุปกรณ์สแกนลายนิ้วมือที่ถูกต้อง