ระบบช่วยงานผู้ดูแลระบบเกี่ยวกับการป้องกันผู้บุกรุก

Administrative tool with Intrusion Detection System

นายกมล ตรีธรรมพินิจ
นายวรวุฒิ อริยสินสุวงศ์
นางสาวอุษณีย์ ศุภณีดิส

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2541

ระบบช่วยงานผู้ดูแลระบบเกี่ยวกับการป้องกันผู้บุกรุก

Administrative tool with Intrusion Detection System

โดย

นายกมล ตรีธรรมพินิจ

นายวรวุฒิ อริยสินสุวงศ์

นางสาวอุษณีย์ ศุภณีดิส

อาจารย์ที่ปรึกษา

อาจารย์สุมณฑา หลิมศิริวงษ์

อาจารย์ธนา หงษ์สุวรรณ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2541

ปริญญานิพนธ์ปีการศึกษา 2541

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบช่วยงานผู้ดูแลระบบเกี่ยวกับการป้องกันผู้บุกรุก

Administrative tool with Intrusion Detection System

ผู้จัดทำ

1. นาย กมล	ตรีธรรมพินิจ	รหัสประจำตัว	38014005
2. นาย วรวุฒิ	อริยสินสุวงศ์	รหัสประจำตัว	38014441
3. นางสาว อุษณีย์	ศุภณีดิส	รหัสประจำตัว	38014656

อาจารย์ที่ปรึกษา

(นางสาว สุมณฑา หลิมศิริวงษ์)

อาจารย์ที่ปรึกษา

(นาย ธนา หงษ์สุวรรณ)

ระบบช่วยงานผู้ดูแลระบบเกี่ยวกับการป้องกันผู้บุกรุก

นายกมล ตรีธรรมพินิจ 38014005
นายวรวุฒิ อริยสินสุวงศ์ 38014441
นางสาวอุษณีย์ ศุภณีดิส 38014005
อ. สุมณฑา หลิมศิริวงษ์ อาจารย์ที่ปรึกษา
อ. ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษา
ปีการศึกษา 2541

บทคัดย่อ

ในปัจจุบันนี้มีการนำระบบสารสนเทศเข้าช่วยการทำงานในทุกด้าน ประกอบกับการเจริญเติบโตของเครือข่ายอินเตอร์เนตทำให้ทุกองค์กรมีความจำเป็นต้องตอบรับต่อเครือข่ายนี้ ไม่ว่าจะเป็นบริการเวิลด์ไวด์เว็บ จดหมายอิเล็กโทรนิกส์ ยูสเน็ต ฯลฯ

การเปิดระบบภายในออกสู่ภายนอกเช่นนี้ปัญหาที่ตามมาก็คือความปลอดภัย ซึ่งย่อมตกเป็นภาระของผู้ดูแลระบบอย่างหลีกเลี่ยงไม่ได้ หน้าที่ในการตรวจสอบความปลอดภัยนั้นเป็นงานที่ต้องทำเป็นประจำและควรจะทำตลอดเวลา แต่เนื่องจากว่าผู้ดูแลระบบไม่สามารถตรวจสอบได้ตลอดเวลา จึงมีความต้องการอุปกรณ์ที่จะคอยช่วยตรวจสอบระบบและแจ้งเตือนได้ตลอดเวลา และควรที่จะสามารถสร้างรายงานประจำวัน เพื่อวิเคราะห์ภาพรวมของระบบได้

ปริญญานิพนธ์ชิ้นนี้เป็นการสร้างอุปกรณ์ช่วยเหลืองานของผู้ดูแลระบบ โดยตัวโปรแกรมจะแบ่งเป็นสองส่วน ส่วนแรกจะทำงานคอยตรวจสอบระบบตลอดเวลา และเมื่อเกิดปัญหาก็สามารถแจ้งเตือนผู้ดูแลระบบได้ทันทีแม้ผู้ดูแลระบบจะไม่ได้ล็อกออนอยู่ในขณะนั้น อีกส่วนหนึ่งจะเป็นส่วนที่ใช้รายงานผลสรุปประจำวัน สาเหตุที่จำเป็นต้องแบ่งออกเป็นสองส่วน เนื่องจากการแจ้งเตือนนั้นควรจะเป็นการรายงานสำหรับเหตุการณ์สำคัญที่มีความเสี่ยงสูง ต้องการการสนใจในทันที ส่วนการรายงานสรุปประจำวันนั้นจะมีลักษณะเป็นผลสรุปเพื่อให้เข้าใจภาพรวมได้โดยง่ายและตัดส่วนที่ไม่สำคัญออกไป

Administrative tool with Intrusion Detection System

Mr. Kamon Treetampinij
Mr. Voravud Ariyasinsuwong
Ms. Audsnee Supaneedis
Ms. Sumonta Limsiriwong Advisor
Mr. Thana Hongsuwan Advisor

ABSTRACT

Nowadays, Information Technology is implemented for helping business process in almost all of modern organization. In addition, the growth of Internet using application such as World Wide Web, Electronic mails, Usenet, etc makes every organization easily communicates with other business parts around the world.

The ease of connecting to the world in vice versa it pursue to be the ease of attack to private information in their own company as well. Internet using increase a lot more concern in computer security for System Administrator, he can't do that all the times. So tools that can check and warn the system when any suspect events occurred all times are needed. And they ought to report the system 's conclusions to the system administrator everyday.

This thesis concern with developing a tool for reducing system administrator 's jobs. The program consists of two parts depend on whether it is the regular or critical report. The output part of this program also depends on how urgent of those reports. We have 3 methods of mailing to mail system, paging and sending message to ICQ. The first runs in background all time to check the critical events and inform the system administrator, even though, the system administrator is not logging on. Another is present daily report. And the daily report is concluded about the record of connection or some risk application in that day.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้จะไม่สามารถเสร็จสมบูรณ์ได้ถ้าไม่ได้รับคำแนะนำ คำเตือน ทั้งหลายจาก อาจารย์ สุมณฑา หลิมศิริวงษ์ และ อาจารย์ ธนา หงษ์สุวรรณ คณะผู้จัดทำจักขอบพระคุณยิ่งสำหรับทุกสิ่งทุกอย่างที่ได้รับจากท่านทั้งสอง

นอกจากนี้ก็ต้องขอขอบพระคุณ อาจารย์ทุกท่านในสถาบันนี้ที่ได้สอนสั่งคณะผู้จัดทำจนมีความรู้ความสามารถจนถึงทุกวันนี้ รวมทั้งคณาจารย์ทุกท่านในภาควิชาคอมพิวเตอร์ที่ทำให้คณะผู้จัดทำได้เป็นวิศวกรคอมพิวเตอร์อย่างเต็มภาคภูมิ

ขอขอบคุณภาควิชาวิศวกรรรมคอมพิวเตอร์ ที่ได้เอื้อเฟื้อสถานที่ ให้คณะผู้จัดทำได้ทำการวิจัยและช่วยอำนวยความสะดวกต่าง ๆ

ขอขอบใจเพื่อน Gang 4d ทุกคนที่ช่วยเหลือคณะผู้จัดทำในการทำงานตลอดเวลา และเป็นที่ปรึกษายามที่เรามีปัญหา

ท้ายที่สุดนี้ต้องขอขอบพระคุณ คุณบิดา มารดาที่ได้ให้กำเนิด คอยสั่งสอน และให้การศึกษา พร้อมทั้งสนับสนุนในกิจกรรมด้านต่าง ๆ นับเป็นพระคุณที่หาที่เปรียบมิได้ ทางคณะผู้จัดทำขอกราบขอพระคุณมา ณ ที่นี้ด้วย

กมล ตรีธรรมพินิจ
วรวุฒิ อริยสินสุวงศ์
อุษณีย์ ศุภณีดิส

สารบัญ

สารบัญตาราง

สารบัญภาพ

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

การศึกษาเกี่ยวกับการรักษาความปลอดภัยในระบบคอมพิวเตอร์เป็นสิ่งที่สำคัญมาก ทั้งนี้เนื่องจากแต่ละหน่วยงานส่วนใหญ่จะต้องมีการเก็บข้อมูลซึ่งเป็นความลับ ซึ่งไม่สามารถเปิดเผยให้ผู้อื่นรู้ได้ไว้ภายในระบบคอมพิวเตอร์ของตนเอง ด้วยเหตุนี้จึงมีผู้ไม่หวังดีคิดหาวิธีการที่จะได้มาซึ่งความลับนั้น ไม่ว่าจะด้วยประโยชน์ส่วนตนหรือด้วยเหตุผลอื่นใดก็ตาม ปัจจุบันมีผู้คิดค้นวิธีการต่าง ๆ เพื่อที่จะป้องกันให้ระบบของตนเองได้รับความปลอดภัยสูงสุด ตัวอย่างของเครื่องมือที่ใช้ก็ได้แก่ ไฟร์วอลล์ (firewall) เป็นต้น

สำหรับระบบที่ได้ทำการศึกษาเป็นระบบรักษาความปลอดภัยระบบหนึ่งคือ ไอดีเอส (IDS : Intrusion Detection System)

ไอดีเอส คือ ระบบการตรวจจับแอคชั่น (Action) หรืออีเวนต์ (Event) ต่าง ๆ เพื่อป้องกันการแอคเซส (Access) เข้ามาของผู้ที่ไม่น่าเชื่อถือ โดยทั่ว ๆ ไปแล้วการพยายามเข้ามานั้นมีจุดประสงค์เพื่อ

1. เข้าถึงข้อมูล (Access Information)
2. โยกย้ายข้อมูลไป (Manipulate Information)
3. ทำให้ระบบนั้นไม่สามารถใช้งานได้ (Render a System Unreliable or Unusable)

1.2 วัตถุประสงค์ของงานวิจัย

1.2.1 เพื่อช่วยเหลือผู้ดูแลระบบ ในการตรวจล็อกไฟล์ของระบบ ซึ่งเป็นงานที่ซ้ำซากและเสียเวลาโดยเปล่าประโยชน์

1.2.2 เพื่อศึกษาเกี่ยวกับตรวจจับผู้บุกรุกเข้ามาในระบบหรือวิเคราะห์ความเป็นไปได้ว่าผู้ใช้คนใดเข้าข่ายเป็นผู้บุกรุกแบบเรียลไทม์ (Real time) และจะต้องแจ้งเตือนแก่ผู้ดูแลระบบทันที

1.2.3 เพื่อสร้างระบบที่ยืดหยุ่น เพื่อที่จะแก้ไขและเพิ่มเติมได้โดยง่าย

1.2.4 เพื่อเป็นแนวทางในการพัฒนาระบบรักษาความปลอดภัยในอนาคต

1.3 ขอบเขตของงานวิจัย

งานวิจัยนี้จะดำเนินการทดลองโดยใช้ Linux Redhat v.5.2 เป็นระบบปฏิบัติการ ซึ่งเป็นระบบปฏิบัติการซึ่งใหม่อยู่ในขณะนี้ จึงยังอาจมีข้อผิดพลาดของระบบเองเกิดขึ้นได้

นอกจากนี้ในปัจจุบันการสื่อสารข้อมูลช่วยทำให้คนทั่วโลกสามารถติดต่อกันได้อย่างรวดเร็ว ข้อมูลต่าง ๆ จึงถูกส่งไปมาระหว่างกันได้โดยง่าย และการค้นพบวิธีการใหม่ ๆ สำหรับการบุกรุกเข้าไปในระบบต่าง ๆ ก็ถูกค้นพบมากขึ้น ดังนั้นทำให้เราสามารถป้องกันระบบได้เพียงในกรณีการบุกรุกที่เราได้ศึกษาไว้เท่านั้น แต่ไม่สามารถป้องกันระบบที่ถูกบุกรุกในกรณีที่นอกเหนือจากที่ศึกษาไว้ได้

นอกจากข้อจำกัดที่ได้กล่าวมาข้างต้นนี้ ยังมีข้อจำกัดของงานวิจัยอีกอย่างหนึ่งคือ ข้อจำกัดเกี่ยวกับล็อกไฟล์ของระบบที่ได้ถูกสร้างขึ้นมา อาจจะมีข้อมูลไม่เพียงพอสำหรับการวิเคราะห์การบุกรุกเข้ามาในระบบ

1.4 วิธีการดำเนินงาน

การดำเนินงานเริ่มต้นด้วยการศึกษาทฤษฎีและแนวความคิดต่าง ๆ ซึ่งเกี่ยวกับระบบรักษาความปลอดภัยและระบบที่จะสร้างขึ้นซึ่งแสดงรายละเอียดต่าง ๆ ในบทที่ 2 จากนั้นจะนำความรู้ที่ได้ศึกษามาข้างต้นมาวิเคราะห์และออกแบบเพื่อสร้างแบบจำลอง สำหรับแบบจำลองที่จะใช้ในงานวิจัยได้แสดงรายละเอียดในบทที่ 3

หลังจากนั้นจะเริ่มเข้าสู่ขั้นตอนการทดลองและค้นหาข้อผิดพลาดแบบจำลองที่ได้ออกแบบมาเพื่อนำมาแก้ไขและพัฒนา

เมื่อได้ทำการทดลองและแก้ไขข้อผิดพลาดต่าง ๆ เรียบร้อยแล้ว ก็จะนำมาสรุปผลการดำเนินงานของงานวิจัยชิ้นนี้ว่ามีข้อดี ข้อเสียอย่างไร รวมทั้งแนวทางการปรับปรุงและพัฒนาในอนาคต พร้อมทั้งจะกล่าวสิ่งที่ได้รับจากงานวิจัย ซึ่งจะอธิบายรายละเอียดในบทที่ 4 - 5

บทที่ 2
ทฤษฎีและหลักการ

2.1 บทนำ

ก่อนที่เราจะเข้านอน เราก็ต้องมั่นใจว่าประตูทุกบานในบ้านล็อคหมดแล้ว เมื่อเราจอดรถ เราก็ต้องตรวจสอบดูว่าเราล็อครถแล้วหรือยัง แล้วในยุคของคอมพิวเตอร์ สิ่งที่มีค่าที่สุดของเรา ก็คือ ข้อมูลต่างๆ ที่มีอยู่ในเครื่องคอมพิวเตอร์ ที่ยิ่งนับวันข้อมูลเหล่านี้ก็ยิ่งมีค่ากับเรามากขึ้น ๆ เปรียบได้กับโลหิตของมนุษย์เลยทีเดียว

เนื่องจากความจำเป็นและความมีค่าของข้อมูลนี่เอง ที่ทำให้ทุกคนปรารถนาที่จะเป็นเจ้าของ และด้วยเทคโนโลยี ทำให้มีเทคนิคเกิดขึ้นมากมายในการจะเป็นเจ้าของข้อมูลเหล่านั้น หรือแม้แต่การทำลาย

หากเรามีระบบ เราก็ต้องมีระบบความปลอดภัยด้วยเช่นกัน เพื่อป้องกันการบุกรุกที่ไม่ประสงค์ดีต่อระบบ ซึ่งจริงๆ แล้วทั้งธรรมชาติและฝีมือมนุษย์สามารถทำอันตรายต่อระบบได้ทั้งสิ้น

แต่เราก็ไม่สามารถป้องกันภัยอันเกิดจากธรรมชาติเหล่านั้นได้ เพราะเราไม่สามารถทราบล่วงหน้าถึงเวลาและความรุนแรงของมัน เราทำได้เพียงรับมือกับเหตุการณ์ที่จะเกิดขึ้นจากธรรมชาตินั้น

แต่ความอันตรายอันเกิดจากฝีมือมนุษย์นั้น หากเรามีความรอบคอบ และหมั่นหาความรู้ เราก็จะสามารถป้องกันการบุกรุกที่จะเกิดขึ้นได้ ซึ่งหน้าที่นี้เป็นหน้าที่หลักของ ผู้ดูแลระบบ (Administrator) ซึ่งทางหนึ่งที่ผู้ดูแลระบบสามารถทำได้ก็คือ การตรวจสอบล็อกไฟล์ (Log file) ที่เกิดขึ้น ซึ่งจะเก็บเหตุการณ์ต่างๆ ของระบบไว้ ให้ผู้ดูแลระบบสามารถมาตรวจตราความเป็นไปของระบบได้จากล็อกไฟล์ แต่เนื่องจาก หากเป็นระบบที่ใหญ่ และมีความสำคัญมาก รองรับผู้ใช้จำนวนมาก ก็จะยิ่งมีล็อกไฟล์ที่มีขนาดใหญ่และโตเร็วมาก งานการตรวจสอบล็อกไฟล์จึงกลายเป็นงานที่น่าเบื่อ และจำเจ แต่หากมีการตรวจสอบล็อกไฟล์อยู่เสมอจะเหมือนมียามเช็คคนเข้าออกเสมอ เพราะฉะนั้นจะเป็นทางป้องกันที่ดีอย่างหนึ่ง แต่หากผู้ดูแลระบบไม่ใส่ใจในเรื่องนี้แล้ว ก็อาจจะทำให้ความปลอดภัยของระบบลดลงได้

เพราะฉะนั้นเราจึงน่าจะมีผู้ช่วยในการตรวจสอบล็อกไฟล์จำนวนมากนั้น ทำให้ผู้จัดทำเลือกที่จะทำระบบผู้ช่วยผู้ดูแลระบบนี้ขึ้นมา และระบบที่จะทำการทดลองนั้นจะต้องมีระบบล็อกไฟล์ที่มีประสิทธิภาพพอในการตรวจสอบความปลอดภัยของระบบ

และองค์ประกอบอื่นที่จำเป็นในการสร้างระบบผู้ช่วยนี้ อย่างเช่นส่วนที่ช่วยในการเช็คล็อกไฟล์จะต้องมีคุณสมบัติในการตรวจสอบรูปแบบของข้อความจำนวนมาก

และยังต้องมีองค์ประกอบที่เป็นส่วนที่ทำงานตลอดเวลาเพื่อความมีประสิทธิภาพสูงในการตรวจสอบความปลอดภัยให้สูงสุดและมีประโยชน์สูงสุดแก่ผู้ดูแลระบบที่ใช้ระบบผู้ช่วยนี้

2.2 ลีนุกซ์ (Linux)

2.2.1 ลีนุกซ์คืออะไร

ลีนุกซ์เป็นระบบปฏิบัติการแบบ 32 บิตอย่างแท้จริง โดยมีลักษณะเป็นยูนิกซ์โคลน (Unix clone) สำหรับเครื่องพีซี และแจกจ่ายให้ใช้ฟรี สนับสนุนการใช้งานแบบหลายงาน หลายผู้ใช้ (MultiUser-MultiTasking) มีระบบเอ็กซ์วินโดว์ (X-windows) ซึ่งเป็นระบบการติดต่อกับผู้ใช้แบบกราฟฟิก ที่ไม่ขึ้นกับระบบปฏิบัติการ (operating system) หรือฮาร์ดแวร์ใดๆ และมีมาตรฐานการสื่อสาร TCP/IP ที่ใช้เป็นมาตรฐานการสื่อสารในอินเทอร์เนตมาให้ในตัว

ลีนุกซ์มีความเข้ากันได้ (compatible) กับ มาตรฐานโพสซิก (POSIX) ซึ่งเป็นมาตรฐานอินเทอร์เฟสที่ระบบยูนิกซ์ ส่วนใหญ่จะต้องมีและมีรูปแบบบางส่วนที่คล้ายกับระบบปฏิบัติการยูนิกซ์จากค่ายเบิร์กเลย์ (Berkeley) และซิสเต็มวี (System V)

โดยความหมายทางเทคนิคแล้วลีนุกซ์ เป็นเพียงเคอร์เนล (kernel) ของระบบปฏิบัติการ ซึ่งจะทำหน้าที่ในด้านของการจัดสรรและบริหารโพรเซสงาน การจัดการไฟล์และอุปกรณ์อินพุทเอ้าท์พุท (I/O)

นอกจากนี้ในปัจจุบัน ลีนุกซ์ยังสามารถใช้งานบนแพลตฟอร์ม(Platform) ต่าง ๆ ได้ ตัวอย่างเช่น DEC Alpha , Motorolla Power-PC เป็นต้น

แล้วข้อดีของลีนุกซ์อีกข้อคือ ลีนุกซ์มีทีมพัฒนาโปรแกรมอย่างต่อเนื่อง ไม่จำกัดจำนวน เนื่องจากส่วนใหญ่ติดต่อกันผ่านอินเทอร์เนต โดยที่ลีนุส (Linus Torvalds) จะเป็นคนรวบรวมโปรแกรมที่ผู้พัฒนาต่างๆได้ร่วมกันทำการพัฒนาขึ้นมาและแจกจ่ายให้ทดลองใช้เพื่อทดสอบหาข้อบกพร่อง

2.2.2 ประวัติความเป็นมา

ลีนุกซ์ถือกำเนิดขึ้นในประเทศฟินแลนด์ ปี คศ. 1980 โดยลีนุส โทรวัลด์ส (Linus Trovalds) นักศึกษาภาควิชาวิทยาการคอมพิวเตอร์ (Computer Science) ในมหาวิทยาลัยเฮลซิงกิ โดยมีความเห็นว่าระบบปฏิบัติการยูนิกซ์บนพีซีในขณะนั้น ยังมีความสามารถไม่เพียงพอกับความต้องการของผู้ใช้ ดังนั้นจึงได้เริ่มทำการสร้างระบบปฏิบัติการยูนิกซ์ของตนเองขึ้น

ลักษณะการพัฒนาของลีนุกซ์ จะใช้อินเทอร์เนตช่วยในการพัฒนาโดยอาศัยความร่วมมือของนักพัฒนาจากสถานที่ต่าง ๆ จึงสังเกตได้ว่าการพัฒนาของระบบปฏิบัติการลีนุกซ์จะเป็นไปอย่างรวดเร็ว

2.2.3 ทำไมถึงเลือกลีนุกซ์ในการทำโครงงาน

เนื่องจากเป็นระบบปฏิบัติการที่ฟรี คุณสามารถจะขอจากผู้ที่มีลีนุกซ์ หรือจะดาวน์โหลดจากอินเทอร์เนต หรือบีบีเอสได้โดยไม่ผิดกฎหมายและมีผู้นิยมใช้มาก ทำให้มีผู้นำลีนุกซ์ไปแก้ไขให้สามารถใช้งานได้บนตัวประมวลผลหลากหลาย นอกจากนี้ยังมีผู้พัฒนาโปรแกรมประยุกต์ออกมากันมากมาย มีประสิทธิภาพและมีคุณภาพสูง ลีนุกซ์เป็นระบบปฏิบัติการ 32 บิตเต็มรูปแบบ ซึ่ง สามารถจะดึงเอาการทำงานของเครื่องคอมพิวเตอร์ออกมาได้อย่างเต็มกำลัง นอกจากนี้มีคุณลักษณะของระบบยูนิกซ์ (UNIX) เต็มรูปแบบ และเป็นระบบหลากผู้ใช้ หลายงานอย่างแท้จริง และยังสนับสนุนโปรโตคอล แบบ TCP/IP

,SLIP, PPP, UUCP และอื่นๆ

สำหรับการใช้ประโยชน์ของลีนุกซ์ในด้านอื่น ๆ เราจะสามารถใช้ลีนุกซ์ทำประโยชน์ได้หลายอย่าง ไม่ว่าจะเอาไว้ทำการศึกษาระบบยูนิกซ์ หรือคุณสามารถจะศึกษาตัวอย่างการเขียนรหัสโปรแกรมที่ดีได้ หากต้องการจะใช้แอพพลิเคชั่นบนดอส หรือบนวินโดว์ ลีนุกซ์ก็จะมีดอสอีมูเลเตอร์ (DOSEMU) และวินโดว์อีมูเลเตอร์ (WINE) ให้ สำหรับอีมูเลเตอร์ทั้งสองตัวนี้ยังอยู่ในขั้นทดสอบ และยังรันแอพพลิเคชั่นของดอสกับวินโดว์ได้ไม่มาก

ลีนุกซ์ได้ทำการเตรียม เครื่องมือพัฒนาโปรแกรมให้เราไว้อย่างครบครันซึ่งจะมีตั้งแต่แอพพลิเคชันมาตรฐาน คือ C/C++ คอมไพเลอร์ของ GNU และหากเราต้องการพัฒนาระบบบนเอ็กซ์ (X) ก็มี TCL/TK เตรียมไว้ให้ด้วย

สำหรับคอมไพเลอร์ภาษาอื่นๆก็มีเช่น เพิร์ล (Perl) , สมอลล์ท็อค (Smalltalk) , ปาสคาล (Pascal) , ลิปส์ (Lisp) เป็นต้น ถ้าคุณมีความเชี่ยวชาญการเขียน โปรแกรมแบบเอ็กซ์เบส (X-Base) หรือฟ็อกซ์โปร (FoxPro) บนลีนุกซ์ก็มีดาต้าเบสที่มีการเขียนโปรแกรมแบบนี้ให้เช่นกัน และล่าสุดลีนุกซ์ก็มีจาวาคอมไพเลอร์ให้สำหรับผู้ที่ชื่นชอบการเขียนแอพเพลตจาวา สำหรับรันบนอินเทอร์เนตด้วย

2.2.4 กลไกการบันทึกเหตุการณ์ของระบบและเน็ตเวิร์ค (System and network logging mechanisms)

การรวบรวมข้อมูลที่ถูกสร้างขึ้นจากระบบ , เน็ตเวิร์ค , แอพพลิเคชั่น และจากพฤติกรรมของผู้ใช้ (user activities) เป็นสิ่งสำคัญสำหรับการทำการวิเคราะห์ความปลอดภัยของระบบและการทำการตรวจจับการบุกรุก โดยล็อกไฟล์ (log file) จะทำหน้าที่เก็บรายละเอียดของเหตุการณ์ที่เกิดขึ้น ในระบบที่แตกต่างกันก็จะมีล็อกไฟล์ที่เก็บรายละเอียดที่แตกต่างกันด้วย บางระบบอาจจะไม่สามารถเก็บรายละเอียดได้เพียงพอ เพราะฉะนั้นสำหรับแต่ละระบบก็ต้องมีการระบุประเภทและกลไกในการเก็บรายละเอียดนั้นๆ เอง(อาจเป็นการเก็บเรื่องของการเข้าถึงไฟล์ , เรื่องของโปรเซส, เรื่องของเน็ตเวิร์ค , เรื่องของแอพพลิเคชั่น , ฯลฯ) เมื่อเรามีการกำหนดประเภทของเหตุการณ์ที่จะทำการล็อกแล้ว เราก็สามารถใช้ประโยชน์จากข้อมูลเหล่านั้นได้เต็มที่

2.2.5 ทำไมถึงสำคัญ?

ข้อมูลในล็อกไฟล์มักเป็นเพียงแค่เรคคอร์ดของพฤติกรรมที่น่าสงสัย แต่ความผิดพลาดจากข้อมูลที่เก็บไว้เหล่านี้เป็นเครื่องช่วยให้เราสามารถตัดสินใจได้ว่าอาจจะมีคนกำลังพยายามตั้งใจที่จะบุกรุกระบบของเราหรือไม่ก็เป็นข้อมูลในการบอกถึงจุดอ่อนของระบบในขณะนั้น ซึ่งในส่วนนี้มีผลิตภัณฑ์อยู่มากที่มาช่วยในการวิเคราะห์และประมวลผลล็อกไฟล์ของคุณ

สิ่งที่ล็อกไฟล์ควรจะทำได้คือ

- สามารถที่จะเตือนให้ระวังเมื่อมีการบุกรุกเกิดขึ้น
- สามารถช่วยในการกู้ (recover) ระบบได้
- สามารถใช้ในการสืบสวนได้ (investigation)
- สามารถเป็นที่อ้างอิงที่แน่นอนได้

2.2.6 ทำอย่างไร?

➢ ระบุข้อมูลที่ต้องการทำการเก็บรายละเอียด

สิ่งแรกที่ต้องทำก็คือการระบุประเภทของรายละเอียดต่างๆที่เราต้องการเก็บ , กลไกในการเก็บ , จะมีการเก็บเกิดขึ้นที่ไหน และจะนำรายละเอียดเหล่านั้นในล็อกไฟล์ไปเก็บไว้ที่ไหน

Log Category	Types of Information to Log
Users	• รายละเอียดเกี่ยวกับการล็อกอิน-ล็อกเอ้าท์ เช่น ความผิดพลาดต่างๆ , เวลา , ความพยายามที่จะล็อกอินเป็นผู้ใช้ที่มีสิทธิ • การเปลี่ยนแปลงของสถานะต่างๆ
Processes	• บอกถึงผู้ใช้ที่ทำการรันโปรเซส • เวลาที่โปรเซสสตาร์ทอัพและค่าอาร์กิวเมนท์ • สถานะ , เวลาที่ใช้ , การใช้งานซีพียู ของโปรเซส
Systems	• การกระทำที่ต้องการสิทธิพิเศษในการทำ • สถานะและข้อผิดพลาดจากฮาร์ดแวร์และซอฟท์แวร์ • การเปลี่ยนแปลงสถานะของระบบ รวมทั้งการชัตดาวน์และรีสตาร์ท
Networks	• การร้องขอเซอร์วิส • ชื่อของผู้ใช้และโฮสต์ที่ทำการร้องขอเซอร์วิส • เน็ตเวิร์คทราฟฟิก (Network traffic) • การเชื่อมต่อใหม่ (New connections) • ระยะของการเชื่อมต่อ (Connection duration)
File Systems	• การเปลี่ยนแปลงการเข้าถึงคอนโทรลลิสต์และการป้องกันไฟล์ (Control lists and file protections) • การเข้าถึงไฟล์ (การเปิด , การสร้าง , การประมวลผล , การลบ)
Application	• รายละเอียดจากแอพพลิเคชั่นและเซอร์วิสเฉพาะ เช่น ล็อกของเมล์ , ล็อกของการ FTP , ล็อกของเวบเซิร์ฟเวอร์ , ล็อกของโมเดม , ล็อกของไฟร์วอลล์ (Firewall)

ตารางที่ 2.1 แสดงการแบ่งประเภทของล็อกและรายละเอียดของล็อก

จะไม่มีการเก็บล็อกของพาสเวิร์ดแม้แต่พาสเวิร์ดอันที่ผิด เพราะว่าการเก็บล็อกของพาสเวิร์ดที่ถูกต้องเอาไว้นั้นเป็นการสร้างรูรั่วใหญ่ให้กับระบบถ้าการเก็บนั้นมีการเข้าถึงอย่างไม่ถูกต้อง การเก็บล็อกของพาสเวิร์ดที่ไม่ถูกต้องก็เป็นอันตรายพอกันเพราะอาจมีพาสเวิร์ดที่ผิดเพียงตำแหน่งเดียวเก็บอยู่หรือเพียงสลับที่กัน เพราะฉะนั้นหากมีการเข้าถึงได้ก็หมายถึงอันตรายมาถึงระบบแล้วเช่นกัน

ข้อมูลที่เราควรจะเก็บจะเป็นในเรื่องของความพยายามที่จะล็อกอินแล้วผิดพลาดกี่ครั้ง หรือการเข้าถึงแอคเค้าท์ที่พิเศษมากกว่า

- ตัดสินใจดูว่าจะมีกลไกการเก็บล็อกอย่างไรถึงจะเพียงพอต่อระบบของเรา

ตัดสินใจดูว่ากลไกการเก็บล็อกอย่างไรจึงจะเหมาะสมและเพียงพอต่อระบบของเรา จะตั้งชื่อล็อกไฟล์เหล่านั้นอย่างไร และจะเก็บล็อกไฟล์เหล่านั้นไว้ที่ไหน เนื่องจากการตั้งชื่อของล็อกไฟล์ของแต่ละผู้ผลิตของแต่ละเวอร์ชั่นถึงแม้จะในระบบปฏิบัติการเดียวกันก็อาจจะแตกต่างกัน เพราะฉะนั้นชื่อจึงเป็นสิ่งสำคัญมากเมื่อมีการปรับปรุงระบบ

- ทำให้การเก็บล็อกนั้นเกิดขึ้นจริงได้

หลังจากที่เราได้เลือกตามขั้นตอนข้างต้นมาแล้ว เราสามารถเลือกทูล (tools) ได้ หลังจากนั้นเราต้องทำตามที่คำแนะนำหรือเอกสารที่ทูลนั้นบอกไว้ทีละขั้นตอนตั้งแต่การเซ็ทอัพไปเรื่อยๆ

ในบางทูลอาจจะอนุญาตให้เราสามารถเป็นผู้กำหนดไฟล์หรือไดเรกทอรี่ที่จะเก็บได้ แต่เราต้องมั่นใจว่าเรามีเนื้อที่พอสำหรับข้อมูลที่จะเกิดขึ้น และมั่นใจว่าล็อกไฟล์นั้นมีการป้องกันอย่างดีแล้ว

- ป้องกันล็อกไฟล์นั้นเพื่อความถูกต้องของล็อกไฟล์

ข้อมูลส่วนใหญ่ในล็อกไฟล์จะเป็นข้อมูลที่เราไม่ต้องการให้ผู้บุกรุกระบบมาเอาไปได้ เพราะฉะนั้นเราต้องทำให้มั่นใจได้ว่าล็อกไฟล์ของเราจะไม่ถูกเข้าถึงหรือถูกแก้ไขจากผู้ใช้ที่ไม่มีสิทธิ จะมีแต่เพียงผู้ใช้ที่มีสิทธิเท่านั้นที่จะสามารถเข้าถึงยูทิลิตี้ที่เป็นตัวกำหนดค่าต่างๆ , การเปิด ปิด , การเขียน , การแก้ไข , หรือการอ่านล็อก

เป็นเรื่องที่สำคัญมากที่เราจะต้องปกป้องล็อกไฟล์ไม่ให้ผู้บุกรุกเข้าถึง , ลบหลักฐานของการบุกรุก , หรือเพิ่มเติมล็อกไฟล์ ด้วยวิธีการต่อไปนี้จะช่วยให้มั่นใจได้ว่าล็อกไฟล์ไม่ได้ถูกแก้ไข

- นำล็อกไฟล์ไปเก็บไว้ยังโฮสต์อื่น ที่ไม่สามารถเข้าถึงได้ง่ายจากระบบเน็ตเวิร์ค
- เก็บล็อกไฟล์เหล่านั้นไว้ในอุปกรณ์ที่เขียนได้ครั้งเดียว แต่สามารถอ่านได้ เช่น ซีดีรอม
- ถ้าเป็นไปได้ให้เซ็ตแอตตริบิวของล็อกไฟล์ให้มีคุณสมบัติที่มีเพียงข้อมูลใหม่เท่านั้นที่จะเขียนเพิ่มไปในไฟล์ได้ ทำให้มั่นใจได้ว่าข้อมูลที่เก็บลงไฟล์ไปแล้วจะไม่มีการแก้ไข
- มีการเข้ารหัสข้อมูลในล็อกไฟล์ ในกรณีที่ต้องมีการส่งข้อมูลที่สำคัญนั้นผ่านระบบเน็ตเวิร์ค

- วางแผนการจัดการกับล็อกไฟล์ที่เกิดขึ้น

โดยทำตามหัวข้อต่อไปนี้

- เตรียมการรับมือกับข้อมูลที่จะเกิดขึ้นจำนวนมาก อย่างที่บอกไว้แล้วว่าล็อกไฟล์จะมีขนาดใหญ่ขึ้นมากในเวลาอันรวดเร็ว ยิ่งระบบใหญ่ก็ยิ่งโตเร็วเท่านั้น และเป็นการยากที่จะคาดเดาได้ว่าล็อกไหนที่จะมีเหตุการณ์ที่เป็นการบุกรุกเก็บอยู่ เราอาจจะรับมือด้วยการทำการบีบอัด (Compress) ล็อกไฟล์เหล่านั้นเก็บไว้ เพื่อที่จะยังคงเก็บไว้เพื่อการกลับมาดูอีกครั้งและเป็นการประหยัดเนื้อที่

- ตัดสินใจว่าข้อมูลจากล็อกไฟล์ไหนที่มีค่ามากที่สุดในการเก็บรักษาไว้ เราจำเป็นต้องให้ความสำคัญกับล็อกไฟล์ของระบบ , เน็ตเวิร์ค , และพฤติกรรมของผู้ใช้ไม่เท่ากันด้วยในแง่ของการเก็บ การประมวลผล การตรวจตรา และความปลอดภัยของมัน คำถามต่อไปนี้จะช่วยเราในการตัดสินใจ

i. โฮสต์มีวัตถุประสงค์หลักอะไร? เช่น ถ้าเป็นเวบเซิร์ฟเวอร์เราก็จะต้องการข้อมูลเก็บข้อมูลที่เป็นเวบล็อก (Web logs)

ii. โฮสต์ของเรารองรับผู้ใช้ได้กี่คนและการล็อกอินของแต่ละคนมีความสำคัญมากแค่ไหน? ข้อมูลเหล่านี้จะช่วยเหลือเราในการตัดสินใจ ว่าเราต้องการข้อมูลการล็อกอิน-ล็อกเอ้าท์มากแค่ไหน

iii. แต่ละล็อกมีความสำคัญมากแค่ไหนในการทำการตรวจสอบการถูกบุกรุก? สิ่งนี้จะช่วยให้เราให้ลำดับความสำคัญของการเก็บได้ เช่น เก็บล็อกของการถ่ายโอนไฟล์ เป็นต้น

iv. ขอบเขตของการให้บริการของโฮสต์? เพื่อที่จะเป็นประโยชน์ในการตรวจจับคนที่ไม่ได้รับอนุญาต

v. ระบบของเรามีความสามารถแค่ไหนในการประมวลผลและวิเคราะห์ล็อกไฟล์เหล่านั้นเมื่อถึงเวลาต้องทำ?

- ทำโรเทท (Rotate) ไฟล์ หมายถึงการทำดังนี้

i. ทำสำเนาของล็อกไฟล์ที่ออนไลน์หรือแอกทีฟ (active) อยู่ในขณะนั้นโดยทำเป็นประจำด้วยช่วงเวลาหนึ่งๆ เช่น อาจจะทำเป็นประจำทุกวันหรือทุกอาทิตย์

ii. เปลี่ยนชื่อไฟล์สำเนานั้น

iii. ทำการรีเซตคอนเทนท์ของไฟล์ (file contents)

iv. ตรวจสอบว่าการเก็บล็อกยังคงทำงานได้ต่อ

การทำเช่นนี้จะช่วยให้เราสามารถจำกัดขนาดของล็อกไฟล์เท่าที่ต้องการให้มีอยู่ในระบบได้

- ทำการแบ็คอัพล็อกไฟล์ ทำการย้ายล็อกไฟล์ของระบบไปอยู่ในหน่วยความจำถาวร (permament storage) เพื่อใช้ในการกลับมาดูอีกครั้ง
- ทำการเข้ารหัสล็อกไฟล์ อย่างที่กล่าวไปแล้วว่าให้ทำการเข้ารหัสสำหรับข้อมูลที่มีความสำคัญมากๆ เราสามารถเข้ารหัสข้อมูลได้ขณะกำลังเก็บ และถ้าเป็นไปได้ควรจะป้องกันซอฟท์แวร์ที่ใช้ในการเข้ารหัสด้วยและทำสำเนาของคีย์ในการเข้ารหัสไว้ในดิสก์หรือซีดีรอม เพราะว่าถ้าคีย์หายไป ล็อกไฟล์ที่เก็บไว้จะไม่มีประโยชน์
- มั่นใจว่าเรามีระบบและทรัพยากรมนุษย์เพียงพอแต่การวิเคราะห์ล็อกไฟล์เหล่านี้เมื่อต้องการ

- การจัดเก็บล็อกไฟล์ เราต้องมั่นใจได้ว่ามีเดีย (media) ที่เราใช้ในการเก็บล็อกไฟล์เหล่านี้มีความปลอดภัยเพียงพอ

2.2.7 Syslogd

Syslogd เป็นกลไกการจัดเก็บล็อกกิ้งเมสเซส (logging messages) จากเคอร์เนลและแอปพลิเคชั่นที่รันอยู่บนระบบปฏิบัติการลีนุกซ์ ซึ่งโดยปกติแล้วเดมอนตัวนี้จะถูกติดตั้งเป็นดีฟอลล์ของระบบอยู่แล้ว และจะต้องมีการคอนฟิก (Config) ว่าเมสเซสใดจะถูกเก็บลงไฟล์หรือส่งต่อไปยังโฮสต์อื่น

การคอนฟิกของ Syslogd ตามดีฟอลล์นั้นจะไม่ได้เก็บเมสเซสจากแอพพลิเคชั่นโปรแกรมในทุกไพออริตี้ (priority) ซึ่งในหัวข้อนี้จะเป็นการอธิบายถึงวิธีการใช้งานและการคอนฟิก Syslodg เมสเซสทุกๆเมสเซสที่ถูกส่งผ่านมายัง Syslodg จะถูกกำหนดระดับความสำคัญ ซึ่งประกอบด้วย

- แฟกซิลิตี้ (facility) เป็นส่วนที่บอกว่าเมสเซสนั้นมาจากส่วนไหน เช่น จากระบบเมล์
- ไพออริตี้ (priority) เป็นส่วนที่บอกระดับความรุนแรงของเมสเซสนั้น เช่น เป็นการเตือน (warning)

การคอนฟิก Syslogd นั้นจะทำโดยใช้ทั้งแฟกซิลิตี้และไพออริตี้ควบคู่กันในการแบ่งแยกเมสเซสต่างๆ ซึ่งมีรายละเอียดดังนี้

แฟกซิลิตี้	คำอธิบาย
auth	ถูกใช้โดยระบบออโธไรเซชั่น (authorization) ได้แก่ ล็อกอิน
security	เหมือนกับ auth (แต่ได้เลิกใช้ไปแล้ว)
authpriv	ระบบออโธไรเซชั่นที่เกี่ยวข้องกับสิทธิพิเศษ
cron	ใช้สำหรับระบบครอน (cron) และเอที (at)
daemon	ซิสเต็มหรือเน็ตเวิร์คเดมอน
kern	ถูกสร้างจากเมสเซสของเคอร์เนล
lpr	ระบบการพิมพ์
mail	ระบบเมล์
mark	ถูกใช้เป็นการภายในสำหรับการทำไทม์สแตมป์ (timestamp)
news	สงวนไว้สำหรับระบบข่าว
syslog	เกี่ยวกับ syslog
user	เป็นแฟกซิลิตี้พื้นฐานสำหรับโปรแกรมใดๆ
uucp	สงวนไว้สำหรับระบบยูยูซีพี
*	หมายถึงทุกแฟกซิลิตี้ ยกเว้น mark
local0..7	สงวนไว้สำหรับการใช้งานโลคอล

ตารางที่ 2.2 แสดงรายละเอียดของแฟกซิลิตี้

ไพออริตี้	คำอธิบาย
none	ไม่ต้องทำการส่งเมสเซสที่มาจากแฟกซิลิตี้ที่กำหนด
debug	ใช้สำหรับการดีบัก
info	เป็นเมสเซสที่ให้ข้อมูล (information)
notice	แสดงสภาพที่ต้องจับตาดูเป็นพิเศษ
warning	คำเตือนต่างๆ
warn	เหมือนกับ warning (แต่ได้เลิกใช้ไปแล้ว)
err	ความผิดพลาด (error) ต่างๆ
error	เหมือน err (แต่ได้เลิกใช้ไปแล้ว)
crit	สภาพที่มีปัญหารุนแรงอย่างเช่นปัญหาเกี่ยวกับฮาร์ดแวร์
alert	สภาพที่ต้องการการดูแลในทันที
emerg	สภาวะเร่งด่วนทุกชนิด
panic	เหมือนกับ emerg (แต่ได้เลิกใช้ไปแล้ว)

ตารางที่ 2.3 แสดงรายละเอียดของไพออริตี้โดยเรียงลำดับตามความสำคัญจากน้อยไปหามาก

แอคชั่นจะเป็นตัวกำหนดว่าจะทำอย่างไร เมื่อมีเมสเซสที่ต้องกับแฟกซิลิตี้และไพออริตี้ที่กำหนดนั้นๆ

แอคชั่น	คำอธิบาย
/dev/console	ส่งข้อความออกทางดีไวซ์
/var/adm/messages	เขียนเมสเซสนั้นลงไฟล์ที่ชื่อ /var/adm/messages
@loghost	ส่งเมสเซสต่อไปยังโฮสต์อื่น
fred,user1	ส่งเมสเซสไปยังผู้ใช้ชื่อ fred และ user1
*	ส่งเมสเซสไปยังผู้ใช้ที่กำลังล็อกออน (log on) อยู่ทุกคน

ตารางที่ 2.4 ตัวอย่างของแอคชั่น

วิธีการในการปกป้องเมสเซสที่เก็บรวบรวมมา ก็สามารถทำได้ด้วยวิธีการที่ได้กล่าวมาแล้วข้างต้นในหัวข้อ "กลไกการบันทึกเหตุการณ์ของระบบและเน็ตเวิร์ค"

นอกจากนี้ยังมีสัญลักษณ์พิเศษที่ใช้ในการกำหนดค่าให้กับ syslog.conf ดังตาราง

สัญลักษณ์	คำอธิบาย
, (คอมม่า)	ใช้สำหรับการกำหนดหลายแฟกซิลิตี้ที่มีไพออริตี้เดียวกัน เช่น mail , auth.info มีความหมายเหมือนกับ mail.info และ auth.info
; (เซมิโคลอน)	เป็นการกำหนดการเลือกแฟกซิลิตี้และไพออริตี้หลายอันเข้ากับแอคชั่นเดียว เช่น mail.info;kern.crit /dev/console
= (เครื่องหมายเท่ากับ)	หมายถึงเฉพาะไพออริตี้นั้นๆเท่านั้น
! (เครื่องหมายอัศเจรีย์)	ทุกไพออริตี้ที่มีความสำคัญต่ำกว่า
#(เครื่องหมาย hash)	เป็นคอมเมนท์ (comment)
| (เครื่องหมายเส้นแนวตั้ง)	ใช้ในส่วนแอคชั่น เพื่อสร้าง fifo ในการเชื่อม 2 คำสั่งเข้าด้วยกัน
@ (เครื่องหมาย AT)	ใช้ในส่วนแอคชั่น เพื่อส่งต่อเมสเซสไปยัง loghost อื่น
* (เครื่องหมายดอกจัน)	หมายถึงทุกไพออริตี้หรือทุกแฟกซิลิตี้

ตารางที่ 2.5 แสดงสัญลักษณ์พิเศษที่ใช้ใน syslog.conf

ตัวอย่างของ syslogd.conf

1. *.=crit;kern.none /var/adm/critical

หมายถึง ทุกแฟกซิลิตี้ที่มีไพออริตี้เป็น crit ขึ้นไปยกเว้นแฟกซิลิตี้ kern ให้มาเก็บไว้ที่ไฟล์ /var/adm/critical

2. kern.* /var/adm/kernel

หมายถึง แฟกซิลิตี้ kern ทุกไพออริตี้ให้เก็บไว้ที่ไฟล์ /var/adm/kernel

3. kern.crit @finlandia

หมายถึง แฟกซิลิตี้ kern ที่มีไพออริตี้ crit ขึ้นไป ให้ส่งต่อไปยัง loghost ชื่อ finlandia

4. kern.crit /dev/console

หมายถึง แฟกซิลิตี้ kern ที่มีไพออริตี้ crit ขึ้นไป ให้แสดงออกทางหน้าจอ

5. kern.info;kern.!err /var/adm/kernel-info

หมายถึงแฟกซิลิตี้ kern ที่มีไพออริตี้ infoขึ้นไปยกเว้นไพออริตี้ err ขึ้นไปให้เก็บไว้ที่ไฟล์/var/adm/kernel-info

6. mail.=info /dev/tty12

หมายถึง แฟกซิลิตี้ mail ที่มีไพออริตี้ info ให้แสดงผลที่เทอร์มินอลหมายเลข 12

7. mail.*;mail.!=info /var/adm/mail

หมายถึง แฟกซิลิตี้ mail ทุกไพออริตี้ ยกเว้น info ให้เก็บไว้ที่ไฟล์ /var/adm/mail

8. mail,news.=info /var/adm/info

หมายถึง แฟกซิลิตี้ mail และ news ที่มีไพออริตี้เป็น info ให้เก็บไว้ที่ไฟล์ /var/adm/info

9. *.=info;*.=notice;mail.none /var/log/messages

หมายถึง ทุกแฟกซิลิตี้ ที่มีไพออริตี้ info และ notice ยกเว้นแฟกซิลิตี้ mail ให้เก็บไว้ที่ไฟล์ /var/log/messages

10. *.=info;mail,news.none /var/log/messages

หมายถึง ทุกแฟกซิลิตี้ ที่มีไพออริตี้เป็น info ยกเว้นแฟกซิลิตี้ mail และ news ให้เก็บไว้ที่ไฟล์ /var/log/messages

11. *.=emerg *

หมายถึง ทุกแฟกซิลิตี้ ที่มีไพออริตี้เป็น emerg ให้ส่งให้ผู้ใช้ทุกคนที่ล็อกออนอยู่

12. *.alert root,joey

หมายถึง ทุกแฟกซิลิตี้ ที่มีไพออริตี้เป็น alert ขึ้นไป ให้ส่งให้ผู้ใช้ที่ชื่อว่า root และ joey

13. *.* @finlandia

หมายถึง ให้ส่งทุกเมสเซสไปยัง loghost ที่ชื่อ finlandia

2.2.8 ระบบล็อกของลีนุกซ์ (Linux's Log system)

เนื่องจากในโครงงานนี้มีการนำเอาล็อกไฟล์ของระบบมาพิจารณาทั้งหมด 4 ล็อกไฟล์ คือ

- **/var/log/cron**

  เป็นล็อกไฟล์ที่เกิดจากล็อกของระบบ cron ที่มาจากเดมอนชื่อ crond

  <u>ตัวอย่างล็อกไฟล์ cron</u>

  root (01/27-05:01:00-514) CMD (run-parts /etc/cron.hourly)

  root (01/27-06:01:00-517) CMD (run-parts /etc/cron.hourly)

  root (01/28-04:02:00-585) CMD (run-parts /etc/cron.daily)

  CRON (01/28-11:15:52-222) STARTUP (fork ok)

- **/etc/log.d/messages**

  เป็นล็อกไฟล์ที่เกิดจาก syslogd จะเก็บทุกเมสเซสที่มีไพออริตี้เป็น info ขึ้นไปยกเว้นจากแฟกซิลิตี้ mail และ authpriv

  ตัวอย่างล็อกไฟล์ messages

  Aug 16 21:52:14 mod syslogd 1.3-3: restart.

  Aug 16 21:53:01 mod login[91]: ROOT LOGIN on `tty1'

  Aug 16 21:53:52 mod passwd[103]: password for `root' changed by `root'

  Aug 16 22:12:14 mod -- MARK --

  Aug 16 22:15:41 mod groupadd[283]: new group: name=web, gid=101

  Aug 16 22:17:13 mod useradd[285]: new user: name=webadmin, uid=1000, gid=101, home=/var/lib/httpd, shell=/bin/bash

  Aug 16 22:17:55 mod chfn[287]: changed user `webadmin' information.

  Aug 16 22:20:41 mod passwd[288]: password for `webadmin' changed by `root'

  Aug 16 22:32:14 mod -- MARK --

  Aug 16 22:52:14 mod -- MARK --

- **/etc/lod.d/xferlog**

  เป็นล็อกไฟล์ที่เกิดจากล็อกของระบบ FTP ที่ชื่อ in.ftpd

  ตัวอย่างล็อกไฟล์ xferlog

  Mon Jan 13 03:30:45 1997 1 sunshine05.ce.kmitl.ac.th 2950 /etc/log.d/scripts/services/cron b _ o r nueng ftp 0 *

  Mon Jan 13 03:30:47 1997 1 sunshine05.ce.kmitl.ac.th 4158 /etc/log.d/scripts/services/ftpd-messages b _ o r nueng ftp 0 *

  Mon Jan 13 03:30:49 1997 1 sunshine05.ce.kmitl.ac.th 3556 /etc/log.d/scripts/services/ftpd-xferlog b _ o r nueng ftp 0 *

  Mon Jan 13 03:30:51 1997 1 sunshine05.ce.kmitl.ac.th 6098 /etc/log.d/scripts/services/identd b _ o r nueng ftp 0 *

  Mon Jan 13 03:30:53 1997 1 sunshine05.ce.kmitl.ac.th 1644 /etc/log.d/scripts/services/init b _ o r nueng ftp 0 *

- **/etc/log.d/secure**

  เป็นล็อกไฟล์เกิดจาก syslogd ที่มีแฟกซิลิตี้เป็น authpriv

  ตัวอย่างล็อกไฟล์ secure

Mar 22 16:21:50 casiopea login: FAILED LOGIN 1 FROM (null) FOR root, Authentica tion failure

Mar 22 16:21:57 casiopea login: FAILED LOGIN 2 FROM (null) FOR root, Authentica tion failure

Mar 22 18:37:29 casiopea in.ftpd[571]: connect from 161.246.6.85

Mar 22 21:47:46 casiopea in.telnetd[581]: connect from 161.246.6.86

Mar 22 21:48:01 casiopea login: FAILED LOGIN SESSION FROM sunshine06.ce.kmitl.a c.th FOR , Error in service module

Mar 22 21:48:02 casiopea in.telnetd[583]: connect from 161.246.6.86

Mar 22 21:51:57 casiopea in.telnetd[608]: connect from 161.246.6.86

Mar 22 21:54:21 casiopea in.telnetd[624]: connect from 161.246.6.86

2.3 เพิร์ล (Perl)

2.3.1 ความเป็นมาและความสำคัญ

เพิร์ล (Perl) คือภาษาที่ใช้ในการเขียนโปรแกรมภาษาหนึ่งย่อมาจาก "Practical Extraction and Report Language " มีลักษณะเป็นสคริปต์ (script) ที่มีตัวแปลภาษาเป็นอินเตอร์พีทเตอร์ (Interpreter) เพิร์ลถูกสร้างขึ้นมาเพื่อรองรับงานในหลายรูปแบบ สามารถทำงานได้ดีกับไฟล์ข้อมูลชนิดตัวอักษร (text file) ซึ่งโดยทั่วไปจะเป็นการประมวลผลจากไฟล์ข้อมูลนั้นแล้วทำการพิมพ์ผลการประมวลนั้นออกมา เพิร์ลเป็นภาษาที่ง่ายต่อการเรียนรู้และเป็นภาษาที่รวมเอาข้อดีหลายๆอย่างของภาษาซี, sed, awk และเชลล์ (shell script) รูปแบบของภาษาเพิร์ลจะใกล้เคียงกับภาษาซีค่อนข้างมาก

และคุณสมบัติที่เป็นจุดเด่นของภาษาเพิร์ลคือ การทำการตรวจสอบรูปแบบของข้อความ (pattern matching) โดยใช้ เรกกูล่าร์เอ็กซ์เพรสชั่น (regular expression) ซึ่งเป็นเทมเพลท(template)ในการทำการตรวจสอบข้อความนั้น โดยการตรวจสอบกับข้อความนั้นอาจจะได้ผลเป็นถูกต้องหรือไม่ถูกต้องก็ได้ โดยภายหลังจากการตรวจสอบนั้นเราอาจจะต้องการแทนที่ข้อความนั้นด้วยข้อความอื่นก็ได้

โดยจุดมุ่งหมายของโครงงานที่กล่าวมาแล้วว่าเป็นระบบที่มีพื้นฐานอยู่บนระบบล็อกไฟล์ (log file) ซึ่งต้องนำล็อกไฟล์มาทำการตรวจสอบและประมวลผลที่ต้องการ หลังจากนั้นก็จะทำการแสดงผลของการประมวลผลนั้นออกมาในรูปแบบต่างๆ ทำให้เพิร์ลเป็นภาษาที่มีลักษณะเหมาะสมที่จะนำมาใช้ในโครงงานนี้

2.3.2 ความรู้ทั่วไปเกี่ยวกับภาษาเพิร์ล

ตัวอย่างโปรแกรมภาษาเพิร์ล

```
#!/usr/bin/perl
#
# This is a first sample of perl
#
print "Hello world.\n";
```

จากตัวอย่าง

ในบรรทัดแรกเป็นจะบอกกับเซิร์ฟเวอร์(server)ว่าต้องทำอะไรในการคอมไฟล์โปรแกรมนี้ เช่น ในบรรทัดนี้เป็นการบอกให้เรียกโปรแกรมเพิร์ลมาใช้ในแปลชุดคำสั่งนี้

ใน 3 บรรทัดต่อมาเป็นคำอธิบายโปรแกรม โดยจะต้องนำหน้าด้วย # และทุกสิ่งที่ตามหลังมาไปจนจบบรรทัดจะไม่ถูกนำมาแปลในการทำงาน

ในบรรทัดสุดท้ายเป็นคำสั่งของภาษาเพิร์ลที่สั่งให้โปรแกรมส่งข้อความออกมาทางสแตนด์ดาร์ดเอ้าท์ (stdout) ว่า "Hello world." และขึ้นบรรทัดใหม่ (นั่นคือ \n)โดยคำสั่งแต่ละคำสั่งในภาษาเพิร์ลจะแยกจากกันด้วย ; (เครื่องหมาย เซมิโคลอน)

2.3.2.1 ตัวแปรในภาษาเพิร์ล

การใช้ตัวแปรในภาษาเพิร์ลนั้นเราสามารถใช้ได้โดยไม่ตั้งกำหนดชนิดทั้งชนิดข้อความและตัวเลข ไม่ต้องมีการประกาศตัวแปรก่อน ในภาษาเพิร์ลมีตัวแปรอยู่ 3 แบบ คือ

- ตัวแปรชนิดสเกล่าร์ (Scalar)
- ตัวแปรชนิดอาร์เรย์ (Array)
- ตัวแปรชนิดแอดโซซิเอทีฟอาร์เรย์ (Associative array)

2.3.2.1.1 ตัวแปรแบบสเกล่าร์

เป็นตัวแปรแบบพื้นฐาน ชื่อตัวแปรจะต้องนำหน้าด้วยสัญลักษณ์ $ (เครื่องหมายดอลลาร์) ชื่อตัวแปรสามารถใช้ได้กับทั้งตัวอักษร ตัวเลข และเครื่องหมาย _ (ขีดล่าง) แต่ต้องไม่ขึ้นต้นด้วยตัวเลข ส่วนตัวแปร $_ เป็นตัวแปรพิเศษ ชื่อตัวแปรสเกล่าร์ในภาษาเพิร์ล เป็นแบบเคสเซนท์ซิทีบ (case sensitive) ตัวใหญ่ตัวเล็กจะไม่เหมือนกัน เพราะฉะนั้น $ids จะต่างกับ $IDS เช่น

```
$IDS = "This is a good project";
```

2.3.2.1.2 ตัวแปรแบบอาร์เรย์

ตัวแปรอาร์เรย์ในภาษาเพิร์ลจะต้องนำหน้าชื่อตัวแปรด้วยสัญลักษณ์ @ ชื่อตัวแปรสามารถใช้ได้ทั้งตัวอักษร ตัวเลข และเครื่องหมาย _ (ขีดล่าง) แต่ต้องไม่ขึ้นต้นด้วยตัวเลข ส่วนตัวแปร @_ เป็นตัวแปรพิเศษ ชื่อตัวแปรอาร์เรย์ในภาษาเพิร์ล เป็นแบบเคสเซนท์ซิทีบ ตัวใหญ่ตัวเล็กจะไม่เหมือนกัน เพราะฉะนั้น @ids จะต่างกับ @IDS เช่น

```
@service = ("login","ftp","telnet","map");
$service[0] จะหมายถึง "login"
```

2.3.2.1.3 ตัวแปรแบบแอดโซซิเอทีฟอาร์เรย์

ตัวแปรแอดโซซิเอทีฟอาร์เรย์ในภาษาเพิร์ล ชื่อตัวแปรจะต้องขึ้นต้นด้วยสัญลักษณ์ % (เครื่องหมายเปอร์เซ็นต์) ในการอ้างอิงถึงข้อมูลในตัวแปรอาร์เรย์ ปกติจะใช้ดรรชนีที่เป็นตัวเลข เพื่ออ้างถึงตัวข้อมูลนั้นๆ เช่น ข้อมูลแรกในอาร์เรย์ @service คือ $service[0] ข้อมูลตัวถัดมาในอาร์เรย์ คือ $service[1] ไปเรื่อยๆ แต่ในภาษาเพิร์ล มีตัวแปรอาร์เรย์ที่สามารถอ้างอิงดรรชนีโดยใช้ข้อความได้ ตัวแปรแบบนี้เรียกว่าแอดโซซิเอทีฟอาร์เรย์ เช่น

```
%price=("Internet Magazine", 50,
        "Byte Thailand", 75,
        "KC Weekly", "30 Bht",
        "Viva Friday", 30);
```

จะสามารถค้นราคาของหนังสือแต่ละเล่มได้ดังต่อไปนี้

```
$price{"Internet Magazine"}; # จะให้ผลลัพธ์เท่ากับ 50
$price{"Byte Thailand"}; # จะให้ผลลัพธ์เท่ากับ 75
$price{"KC Weekly"}; # จะให้ผลลัพธ์เท่ากับ "30 Bht"
$price{"Viva Friday"}; # จะให้ผลลัพธ์เท่ากับ 30
```

ตัวแปรแบบแอดโซซิเอทีฟอาร์เรย์สามารถเปลี่ยนเป็นตัวแปรแบบอาร์เรย์ได้ และตัวแปรแบบอาร์เรย์ก็สามารถแปลงเป็นตัวแปรแบบแอดโซซิเอทีฟอาร์เรย์ได้ด้วยเช่นกัน โดยที่ตัวแปรแบบอาร์เรย์ที่จะแปลงเป็นตัวแปรแบบแอดโซซิเอทีฟอาร์เรย์ต้องมีจำนวนสมาชิกเป็นเลขคู่ เช่น

```
@bookdata = %price      # @bookdata เป็นตัวแปรแบบ อาร์เรย์ # ซึ่งตอนนี้มีจำนวนสมาชิก
                        เท่ากับ 8
$bookdata[5];           # จะให้ผลลัพธ์เท่ากับ "30 Bht"
                        # จากข้อมูลในตัวแปร อาร์เรย์ ที่ชื่อ @bookdata
%newprice = @bookdata   # %newprice เป็นตัวแปรแบบ Hash
                        # มีข้อมูลภายในเหมือนกับ %price
```

2.3.3 โอเปอเรเตอร์ (Operator)

2.3.3.1 การดำเนินการและการกำหนดค่าให้กับตัวแปรสเกล่าร์

ตัวอย่าง	คำอธิบาย
$a = $b;	ข้อมูลในตัวแปร $b ไปเก็บไว้ในตัวแปร $a
$a += $b;	นำข้อมูลในตัวแปร $b ไปบวกเพิ่มกับข้อมูลในตัวแปร $a แล้วเก็บผลลัพธ์ไว้ในตัวแปร $a
$a -= $b;	นำข้อมูลในตัวแปร $b ไปลบออกจากข้อมูลในตัวแปร $a แล้วเก็บผลลัพธ์ไว้ในตัวแปร $a
$a .= $b;	ข้อมูลในตัวแปร $b ไปเพิ่มต่อท้ายข้อมูลในตัวแปร $a แล้วเก็บไว้ในตัวแปร $a
$a = 5*6;	นำ 5 มาคูณ ด้วย 6 แล้วเก็บค่าที่ได้ไว้ในตัวแปร $a
$a = 9 ** 10;	นำ 9 มายกกำลังด้วย 10 แล้วนำค่าที่ได้เก็บไว้ในตัวแปร $a
++$a;	นำค่าในตัวแปร $a มาเพิ่มค่าขึ้นอีก 1และส่งค่าที่เพิ่มค่าแล้วนั้นกลับ
$a++;	นำค่าในตัวแปร $a ส่งกลับไปก่อน แล้วจึงค่อยนำค่าในตัวแปร $a มาเพิ่มค่าอีก 1
$a = $b . $c;	นำข้อความในตัวแปร $b รวมกับข้อความในตัวแปร $c แล้วเก็บไว้ในตัวแปร $a
$a = $b x $c;	นำข้อความในตัวแปร $b มาทำซ้ำเป็นจำนวน $c รอบ แล้วเก็บไว้ในตัวแปร $a
-$a;	ค่าในตัวแปร $a มาลบค่าลงอีก 1 และส่งค่าที่เพิ่มค่าแล้วนั้นกลับ
$a-;	ค่าในตัวแปร $a ส่งกลับไปก่อน แล้วจึงค่อยนำค่าในตัวแปร $a มาลบค่าลงอีก 1

ตารางที่2.6 แสดงตัวอย่างการดำเนินการและการกำหนดค่าให้กับตัวแปรสเกล่าร์

2.3.3.2 การดำเนินการและการกำหนดค่าให้กับตัวแปรอาร์เรย์

สำหรับการเพิ่มค่าให้กับตัวแปรอาร์เรย์นั้นจะใช้ฟังก์ชั่น push ดังตัวอย่างนี้

```
push(@morebook , "game magazine");
```

จะนำข้อมูล "game magazine" ไปใส่ต่อท้ายตัวแปรอาร์เรย์ที่ชื่อว่า @morebook ถ้าหากต้องการเพิ่มข้อมูลมากกว่า 2 ชุดเข้าสู่ตัวแปรแบบอาร์เรย์สามารถทำได้โดยใช้รูปแบบใดรูปแบบหนึ่งตามตัวอย่าง

```
push(@morebook, "game magazine" , "newspaper");
push(@morebook,"(game magazine", "newspaper"));
push(@morebook, @otherbook);
```

หลังจากที่ตัวฟังก์ชั่น push ทำงานข้อมูลก็จะถูกเพิ่มเข้าไปในอาร์เรย์ที่ต้องการ ส่วนตัวฟังกชั่น push เองจะส่งค่ากลับมาเป็นขนาดของตัวแปรอาร์เรย์ที่ถูกเพิ่มข้อมูลเข้าไป

หากต้องการที่จะนำข้อมูลตัวท้ายสุดของตัวแปรอาร์เรย์ออกจากตัวแปรอาร์เรย์ สามารถทำได้โดยใช้ฟังกชั่น pop จากตัวแปรอาร์เรย์ ตัวอย่างการใช้ฟังก์ชั่น pop

```
$newdata = pop(@morebook);        # Now $newdata = "newspaper"
```

2.3.3.3 การดำเนินการกับข้อความ

คุณสมบัติเด่นภาษาเพิร์ล คือการดำเนินการกับข้อความ และตัวแปรที่เก็บข้อมูลชนิดข้อความ ภาษาเพิร์ลสามารถทำได้อย่างมีประสิทธิภาพ โดยการใช้เรกกูล่าร์เอ็กซ์เพรสชั่น ซึ่งมีใช้กัน ในยูทิลิตี้ (utility) หลายๆโปรแกรมในระบบยูนิกซ์ (Unix)

เรกกูล่าร์เอ็กซ์เพรสชั่นจะถูกบรรจุอยู่ภายในเครื่องหมาย / และจะใช้ร่วมกับตัวกระทำ =~ ดังตัวอย่างข้อความต่อไปนี้จะมีผลเป็นจริง ก็ต่อเมื่อมีข้อความ the ปรากฏอยู่ในตัวแปร $test

```
$test =~ /the/;
```

การตรวจสอบของเรกกูล่าร์เอ็กซ์เพรสชั่น เป็นแบบเคสเซนท์ซิทีบหากว่า

```
$test = "The saint is a good movie";
```

ผลลัพธ์ก็จะเป็นเท็จ จากตัวอย่างข้างต้น หากใช้ร่วมกับตัวกระทำ !~ จะหมายถึงว่า "ไม่มีใน" จากตัวแปร $test ข้างต้นเมื่อใช้จะได้เป็น

```
$test !~ /the/
```

ตัวแปรพิเศษ $_ เป็นตัวแปรดีฟอลท์ (default) ที่ใช้ได้กับตัวกระทำ (operation) หลายๆอย่างในภาษาเพิร์ล และใช้ประโยชน์ได้มาก เพราะจะทำให้เขียนคำสั่งได้สั้นลง เช่น

```
if (/under/)
{
  print "Yes sir..\n";
}
```

หมายความว่าหากตัวแปร $_ มีคำว่า “under” อยู่ในข้อความให้พิมพ์ว่า “Yes sir..” จะเห็นได้ว่าจะสามารถใช้คำสั่งได้สั้นลงเพราะไม่ต้องใช้เครื่องหมาย =~ หรือ !=

สัญลักษณ์ พิเศษต่อไปนี้สามารถใช้กับเรกกูล่าร์เอ็กซ์เพรสชั่นได้ โดยมีความหมายของสัญลักษณ์ต่างๆดังต่อไปนี้

สัญลักษณ์	คำอธิบาย
.	ตัวอักษร(ตัวเดียว)ใดๆก็ตามยกเว้นตัวอักษรขึ้นบรรทัดใหม่
^	ส่วนแรกของบรรทัดหรือข้อความ
$	ส่วนท้ายของบรรทัดหรือข้อความ
*	ตัวอักษรตัวสุดท้ายกี่ตัวก็ได้หรือจะไม่มีเลยก็ได้
+	ตัวอักษรตัวสุดท้ายกี่ตัวก็ได้แต่ต้องมีอย่างน้อย 1 ตัว
?	ตัวอักษรตัวสุดท้ายหนึ่งตัวหรือจะไม่มีเลยก็ได้

ตารางที่ 2.7 แสดงตัวอย่างสัญลักษณ์ที่ใช้กับเรกกูล่าร์เอ็กซ์เพรสชั่น

เครื่องหมายวงเล็บเหลี่ยม [และ] หมายถึงตัวอักษรใดๆภายในวงเล็บเหลี่ยม และภายในวงเล็บเหลี่ยม เครื่องหมาย - (ขีดกลาง) จะหมายถึง "ระหว่าง" (between) และ เครื่องหมาย ^ (ชี้ขึ้น)ที่ขึ้นต้นข้อความในวงเล็บเหลี่ยม หมายถึง "ไม่" (not)

สัญลักษณ์	คำอธิบาย
[qjk]	q หรือ j หรือ k ตัวใดตัวหนึ่ง
[^qjk]	ไม่ใช่ q หรือ j หรือ k ตัวใดตัวหนึ่ง
[a-z]	อะไรก็ได้ระหว่าง a ถึง z
[a-zA-Z]	ตัวอักษรใดก็ได้ (ไม่ใช่ตัวเลขและสัญลักษณ์)
[a-z]+	ข้อความที่มีตัวอักษรตัวเล็กประกอบอย่างน้อย 1 ตัวอักษร

ตารางที่ 2.8 แสดงตัวอย่างของการใช้งานเครื่องหมายวงเล็บเหลี่ยม

เครื่องหมายขีดตามแนวตั้ง | หมายถึง "หรือ" (or) และสามารถใช้เครื่องหมายวงเล็บ (...) ในการจัดกลุ่มได้ ดังตัวอย่าง

สัญลักษณ์	คำอธิบาย
Jelly|cream	jelly หรือ cream
(eg|le)gs	eggs หรือ legs
(da)+	da หรือ dada หรือ dadada ไปเรื่อยๆ...

ตารางที่ 2.9 แสดงตัวอย่างของการใช้งานเครื่องหมาย | และ ()

และยังมีอักขระพิเศษที่ใช้ในการควบคุมการแสดงผลของข้อความ ซึ่งมีความหมายแตกต่างกันไปตามตาราง

อักขระพิเศษ	คำอธิบาย
\n	ขึ้นบรรทัดใหม่
\t	ตัวอักษรแทป (Tab)
\w	ตัวอักษรภาษาอังกฤษหรือตัวเลขใดๆก็ได้ มีความหมายเหมือนกับ [a-zA-Z0-9_]
\W	ตัวอักษรใดๆก็ได้ที่ไม่ใช่ตัวอักษรภาษาอังกฤษหรือตัวเลข มีความหมายเหมือนกับ [^a-zA-Z0-0_]
\d	ตัวเลขใดๆก็ตาม มีความหมายเหมือนกับ [0-9]
\D	ตัวอักษรใดๆก็ได้ที่ไม่ใช่ตัวเลข มีความหมายเหมือนกับ [^0-9]
\s	ตัวอักษรช่องว่าง (whitespace) ใดๆ เช่น space, tab, ขึ้นบรรทัดใหม่, ฯลฯ
\S	ตัวอักษรใดๆที่ไม่ใช่ช่องว่าง

ตารางที่ 2.10 แสดงอักขระพิเศษและความหมาย

ถ้าหากต้องการอ้างถึงตัวอักษรใดๆที่ถูกเป็นเครื่องหมายที่ใช้ในเรกกูล่าร์เอ็กซ์เพรสชั่น สามารถใช้เครื่องหมาย \ นำ หน้าเครื่องหมายต่างๆเหล่านั้น ดังนี้

สัญลักษณ์	คำอธิบาย
\/	ขีดตามแนวนอน
\[	วงเล็บเหลี่ยมเปิด
\]	วงเล็บปิด
*	เครื่องหมายดอกจัน
\^	เครื่องหมายลูกศรชี้ขึ้น
\/	เครื่องหมายสแลช (slash)
\\	เครื่องหมายแบล็คสแลช (backslash)

ตารางที่ 2.11 แสดงตัวอย่างการใช้งานเมื่อต้องการใช้อักษรที่เป็นเครื่องหมาย

2.3.3.4 ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operators)

ตัวดำเนินการ	ตัวอย่าง	ความหมาย
+	$a + $b	ผลบวกของ $a และ $b
-	$a - $b	ผลต่างของ $a และ $b
*	$a * $b	ผลคูณของ $a และ $b
/	$a / $b	ผลหารของ $a หารด้วย $b

%	$a % $b	เศษของการหาร $a ด้วย $b
**	$a ** $b	ผลของ $a ยกกำลังด้วย $b

ตารางที่ 2.12 แสดงตัวดำเนินการทางคณิตศาสตร์

2.3.3.5 ตัวดำเนินการที่ใช้ในการกำหนดค่า (Assignment Operators)

ตัวดำเนินการ	ตัวอย่าง	ความหมาย
=	$var = 5	ให้ตัวแปร $var มีค่าเท่ากับ 5
++	$var++ or ++$var	เพิ่มค่าของ $var ขึ้นอีก 1 และเก็บใน $var
-	$var-- or --$var	ลดค่าของ $var ลงอีก 1 และเก็บใน $var
+=	$var += 3	เพิ่มค่าของ $var ขึ้นอีก 3 และเก็บใน $var
-=	$var -= 2	ลดค่าของ $var ลงอีก 2 และเก็บใน $var
.=	$str .= "ing"	นำข้อความ "ing" เพิ่มต่อท้ายเข้าไปใน $var
*=	$var *= 4	คูณค่าใน $var ด้วย 4 และเก็บใน $var
/=	$var /= 2	นำค่าใน $var หารด้วย 2 และเก็บใน $var
**=	$var **= 2	นำค่าใน $var ยกกำลัง 2 และเก็บใน $var
%=	$var %= 2	หารค่าใน $var ด้วย 2 และนำเศษจากการหารเก็บใน $var
x=	$str x= 20	ซ้ำค่าในตัวแปร $str 20 รอบและเก็บใน $str

ตารางที่ 2.13 แสดงตัวดำเนินการและตัวอย่างที่ใช้ในการกำหนดค่า

2.3.3.6 ตัวดำเนินการทางตรรกะ(Logical Operators)

ตัวดำเนินการ	ตัวอย่าง	คำอธิบาย
&&	$a &&$b	ให้ผลเป็นจริงเมื่อ $a และ $b เป็นจริงทั้งคู่ (and)
||	$a || $b	ให้ผลเป็นจริงเมื่อ $a หรือ $b ตัวใดตัวหนึ่งเป็นจริง (or)
!	! $a	ให้ผลเป็นจริงเมื่อ $a เป็นเท็จ (not)

ตารางที่ 2.14 แสดงตัวดำเนินการทางตรรกะและตัวอย่าง

2.3.3.7 ตัวดำเนินการในการตรวจสอบรูปแบบของข้อความ (Pattern Matching Operators)

ตัวดำเนินการ	ตัวอย่าง	คำอธิบาย
=~ //	$a =~ /pat/	ให้ค่าเป็นจริงถ้าหาก $a มี "pat"
=~ s//	$a =~ s/p/r	เปลี่ยน 'p' ใน $a เป็น 'r'

=~ tr//	$a =~ tr/a-z/A-Z	แปลงตัวอักษรตามที่กำหนด
!~ //	$a !~ /pat/	ให้ค่าเป็นจริงถ้าหาก $a ไม่มี "pat"

ตารางที่ 2.15 แสดงตัวดำเนินการและตัวอย่างในการตรวจสอบรูปแบบของข้อความ

2.3.3.8 ตัวดำเนินการกับข้อความ(String Operators)

ตัวดำเนินการ	ตัวอย่าง	คำอธิบาย
==	eq	เท่ากันกับ
!=	ne	ไม่เท่ากันกับ
>	gt	มากกว่า
>=	ge	มากกว่าหรือเท่ากันกับ
<	lt	น้อยกว่า
<=	le	น้อยกว่าหรือเท่ากันกับ

ตารางที่ 2.16 แสดงตัวดำเนินการกับข้อความและตัวอย่างการใช้งาน

2.3.4 การตรวจสอบเงื่อนไข

การตรวจสอบเงื่อนไขในภาษาเพิร์ล สามารถทำได้โดยการใช้ชุดคำสั่งตรวจสอบเงื่อนไขแบบ if/then/else ซึ่งมีหลายโครงสร้าง โดยใช้รูปแบบดังต่อไปนี้

โครงสร้างของการตรวจเงื่อนไขแบบ if .. elsif .. else

- if (Expression) { Block }
- if (Expression) { Block } else { Block2 }
- if (Expression) { Block } elsif (Expression2) { Block2 } else { Block3 }

ตัวอย่างการตรวจเงื่อนไข

```
if ($data){
    print "The string is not empty\n";
}
else{
    print "The string is empty\n";
}
```

จากตัวอย่างถ้าหากว่าไม่มีข้อความในตัวแปร $data การตรวจสอบเงื่อนไขจะให้ค่าเป็น false ซึ่งจะแสดงผลลัพธ์ว่าข้อความ "The string is empty" ขึ้นมา

ถ้าหากว่าต้องการที่จะตรวจสอบเงื่อนไขมากกว่า 1 เงื่อนไข ก็สามารถทำได้ด้วย if/elsif/else ดังตัวอย่างต่อไปนี้

```
if (!$data)
{
    print "The string is empty\n";
}
elsif (length($a) == 1)    # If above fails, try this
{
    print "The string has one character\n";
}
elsif (length($a) == 2)    # If that fails, try this
{
    print "The string has two characters\n";
}
else    # Now, everything has failed
{
    print "The string has lots of characters\n";
}
```

2.3.5 การทำงานแบบวนรอบและเงื่อนไขการทำงานแบบวนรอบ

ภาษาเพิร์ลสนับสนุนการทำงานโดยใช้โครงสร้างการควบคุมการทำงานแบบวนรอบ มีการวนรอบรูปแบบต่างๆให้เลือกใช้ตามความเหมาะสมของงานมากมาย โครงสร้างของการวนรอบในภาษาเพิร์ลไม่ยุ่งยากมากนัก และมีลักษณะคล้ายกับใน ภาษาซีและปาสคาล (pascal) ค่อนข้างมาก โดยจะมีชุดคำสั่งสำหรับทำการวนรอบอยู่สองลักษณะคือ

- while, until วนรอบทำงานจนกว่าจะเป็นไปตามเงื่อนไข
- for, foreach กำหนดเงื่อนไขและจำนวนครั้งของการวนรอบ

2.3.5.1 while และ until

- while loop

  while (Expression) { Block }

จะตรวจสอบเงื่อนไข Expression ก่อนถ้าหากว่าเงื่อนไขเป็นจริงก็จะกระทำชุดคำสั่งใน Block และกลับมาตรวจสอบเงื่อนไขอีกครั้ง ทำอย่างนี้ไปเรื่อยๆจนกว่าเงื่อนไขจะเป็นเท็จ จึงจะจบรอบการทำงาน

- until loop

  until (Expression) { Block }

จะตรวจสอบเงื่อนไข Expression ก่อนถ้าหากว่าเงื่อนไขเป็นเท็จก็จะกระทำชุดคำสั่งใน Block และกลับมาตรวจสอบเงื่อนไขอีกครั้ง ทำอย่างนี้ไปเรื่อยๆจนกว่าเงื่อนไขจะเป็นจริง จึงจะจบรอบการทำงาน

- do/until loop

  do { Block } until (Expression)

จะกระทำชุดคำสั่งใน Block ก่อน หลังจากนั้นจึงจะมาทำการตรวจสอบเงื่อนไข Expression ถ้าหากว่าเงื่อนไขเป็นจริงก็จะจบการวนรอบ หากว่าเงื่อนไขยังเป็นเท็จอยู่ก็จะวนกลับไปทำงานใน Block อีกครั้งและมาทำการตรวจสอบเงื่อนไขอีกทำอย่างนี้ไปเรื่อยๆ จนกว่าเงื่อนไขจะเป็นจริง

- do/while loop

  do { Block } while (Expression)

จะกระทำชุดคำสั่งใน Block ก่อน หลังจากนั้นจึงจะมาทำการตรวจสอบเงื่อนไข Expression ถ้าหากว่าเงื่อนไขเป็นเท็จก็จะจบการวนรอบ หากว่าเงื่อนไขยังเป็นจริงอยู่ก็จะวนกลับไปทำงานใน Block อีกครั้งและมาทำการตรวจสอบเงื่อนไขอีกทำอย่างนี้ไปเรื่อยๆ จนกว่าเงื่อนไขจะเป็นเท็จ ตัวอย่างของโปรแกรมที่ใช้การวนรอบ ตัวอย่างโปรแกรมต่อไปนี้จะรับข้อมูลอินพุท (input) จากคีย์บอร์ด(keyboard) และจะไม่ทำงานต่อจนกว่าพาสเวิร์ด (password) ที่ได้รับจะถูกต้อง

```
#!/usr/local/bin/perl
print "Password : ";    # Ask for input
$a = ;    # Get input
chop $a;    # Remove the newline at end
while ($a ne "mysecretpassword")    # While input is wrong..
{
    print "sorry wrong passwd. Try agin : ";    # Ask again
    $a = ;    # Get input again
    chop $a;    # Chop off newline again
}
```

จากตัวอย่างข้างต้นชุดคำสั่งระหว่างเครื่องหมาย { และ } จะถูกกระทำในขณะที่ข้อมูลที่ได้รับไม่ตรงกับ password จากตัวอย่างข้างต้นสำหรับชุดคำสั่งวนรอบ while อีกจุดที่น่าสังเกตคือเมื่อมีการป้อน password ตัวแปร $a จะมีข้อมูลและข้อมูลนั้นก็จะมีสัญลักษณ์ขึ้นบรรทัดใหม่ (new line) ต่อท้ายมาด้วย ฟังก์ชั่น chop จะนำตัวอักขระตัวท้ายสุดออกจากสตริง (string) ซึ่งในที่นี้ก็คือสัญลักษณ์ขึ้นบรรทัดใหม่

จากโปรแกรมตัวอย่างข้างต้นสามารถเขียนขึ้นใหม่โดยใช้รูปแบบของ do .. while ได้ดังต่อไปนี้

```
#!/usr/local/bin/perl
do
{
    "Password : ";     # Ask for input
    $a = ;     # Get input
    chop $a;     # Chop off newline
}
while ($a ne "mysecretpassword") # Redo while wrong password
```

2.3.5.2 for และ foreach

- for loop

  for (Expression1; Expression2; Expression3) { Block }

  Expression1 เป็นค่าเริ่มต้นของการวนรอบ

  Expression2 ใช้สำหรับตรวจสอบว่าจะวนรอบต่อหรือว่าหยุดการวนรอบ

  Expression3 ใช้สำหรับปรับปรุง(update)ตัวแปรที่ใช้ตรวจสอบการวนรอบ

อันดับแรกจะเริ่มกระทำ Expression1 ก่อนเพื่อกำหนดค่าเริ่มต้นของการวนรอบ หลังจากนั้นจะตรวจสอบ Expression2 หากว่าเงื่อนไขเป็นจริงจะกระทำชุดคำสั่งใน Block เมื่อเสร็จสิ้นการทำงานชุดคำสั่งใน Block แล้วก็จะมากระทำ Expression3 หลังจากนั้นจะวนไปขั้นตอนตรวจสอบ Expression2 อีกครั้ง ทำอย่างนี้ไปจนกระทั่ง Expression2 เป็นเท็จ

- foreach loop

  foreach VARIABLE (ARRAY) { Block }

ARRAY เป็นตัวกำหนดการวนรอบว่าจะวนกี่รอบขึ้นอยู่กับจำนวนข้อมูลในตัวแปร อาร์เรย์

VARIABLE เป็นตัวแปรที่จะรับข้อมูลจาก อาร์เรย์ ตัวปัจจุบันที่ดัชนีตรงกับรอบที่วนทำงานอยู่

ถ้าหากว่าไม่กำหนด VARIABLE จะถือว่าเป็นการเรียกใช้ตัวแปร $_ ต่อไปนี้คือตัวอย่างสำหรับการวนรอบเพื่อแสดงตัวเลข 0 ถึง 9 โดยใช้การวนรอบ (loop) แบบ for

```
for ($i = 0; $i < 10; ++$i)
{
    print "$i\n";
}
```

การวนรอบเพื่อทำงานตามชุดข้อมูลใน อาร์เรย์ หรือข้อมูลโครงสร้างแบบ list อื่นๆ (เช่น ตามแต่ละบรรทัดของแฟ้มข้อมูล) ในภาษาเพิร์ล ใช้คำสั่งวนรอบแบบ foreach ดังตัวอย่างต่อไปนี้

```
foreach $book (@title)
{
    print "$book\n"; # Print the item
}
```

ในการวนรอบแรกข้อมูลใน $book จะถูกกำหนดค่าให้เท่ากับข้อมูลชุดแรกในอาร์เรย์ @title ในการวนรอบครั้งต่อไปก็จะมีค่าเท่ากับข้อมูลชุดที่สองในอาร์เรย์ อย่างนี้ไปเรื่อยๆจนกระทั่งหมดชุดข้อมูลในอาร์เรย์ ถ้าหากว่า @title เป็นอาร์เรย์ว่างชุดคำสั่งในเครื่องหมาย { .. } จะไม่ถูกกระทำ

2.3.6 การจัดการแฟ้มข้อมูล

ต่อไปนี้คือตัวอย่างของโปรแกรมภาษาเพิร์ล ที่ใช้สำหรับเปิดแฟ้มข้อมูลมาแสดงผลแบบง่ายๆ ลักษณะการทำงาน จะคล้ายๆกับคำสั่ง cat ในระบบยูนิกซ์

```
#!/usr/local/bin/perl
#
# Program open the file, read it in,
# print it and close it
#
$file = '/tmp/test.txt';      # Name the file
open(INFO,$file);     # Open the file
@lines = <INFO>;      # Read it into an array
close(INFO);      # Close the file
print @lines;     # Print the array
```

เราใช้ฟังก์ชั่น open สำหรับการเปิดแฟ้มข้อมูลเพื่ออ่านข้อมูล ในฟังก์ชั่น open ตัวพารามิเตอร์ (parameter) ตัวแรกคือ filehandle ซึ่งใช้ในการอ้างอิงในการติดต่อกับแฟ้มข้อมูลในภาษาเพิร์ล ส่วน พารามิเตอร์ ตัวที่สองคือ เอ็กซ์เพรสชั่น ที่บอกถึงชื่อแฟ้มข้อมูลที่ต้องการติดต่อ ถ้าหากระบุชื่อแฟ้มข้อมูลไว้ภายใต้เครื่องหมาย quote '...' จะหมายถึงการอ้างอิงแฟ้มข้อมูลชื่อนั้นๆโดยตรง ไม่สามารถใช้รูปแบบการอ้างอิงที่ใช้ในเชลล์(shell) ได้ (เช่น ~ เพื่อ อ้างอิงถึงโฮมไดเรกทอรี่ (home dir) ของผู้ใช้ (user)) ดังนั้นหากระบุชื่อแฟ้มข้อมูลว่า '~/data/mydata' จะทำให้เกิดข้อผิดพลาดขึ้น ไม่สามารถประมวลผลได้ หากว่าต้องการที่จะใช้รูปแบบการอ้างอิงที่ใช้ใน shell จะต้องระบุชื่อแฟ้มข้อมูลไว้ภายใน เครื่องหมายวงเล็บมากกว่าและน้อยกว่า < . . > เช่น <~/data/mydata> แทน

ส่วนฟังก์ชั่น close จะใช้สำหรับบอกโปรแกรมภาษาเพิร์ล ว่าต้องการที่จะปิดการติดต่อกับแฟ้มข้อมูล

นอกจากนี้ในการเปิดใช้งานแฟ้มข้อมูล ยังสามารถจะกำหนดโหมด (mode) ในการเปิดได้อีกด้วย เช่นการเปิดแฟ้มข้อมูล เพื่ออ่านข้อมูล การเปิดแฟ้มข้อมูลเพื่อเขียนแฟ้มข้อมูลใหม่(ทับของเก่า) การเปิด

แฟ้มข้อมูลเพื่อเขียนข้อมูลเพิ่ม เติม(ต่อท้ายของเก่า) โดยระบุเครื่องหมายพิเศษนำหน้าชื่อแฟ้มข้อมูลที่ต้องการจะเปิด โดยเครื่องหมาย > จะใช้ สำหรับการเปิดแฟ้มข้อมูลเพื่อเขียนแฟ้มข้อมูลใหม่(ทับของเก่า) เครื่องหมาย >> จะใช้สำหรับการเปิดแฟ้มข้อมูล เพื่อเขียนข้อมูลเพิ่มเติม(ต่อท้ายของเก่า) ดูตัวอย่างต่อไปนี้

```
open(INFO, $file);          # Open for input
open(INFO,">$file");        # Open for output
open(INFO,">>$file");       # Open for appending
open(INFO,"<$file");        # Also open for input
```

ถ้าต้องการที่จะเขียนข้อมูลอะไรบางอย่างลงในแฟ้มข้อมูลที่ถูกเปิดไว้สำหรับการเขียนข้อมูลแล้วนั้น สามารถทำได้โดยใช้คำสั่ง print โดยใช้พารามิเตอร์พิเศษ โดยระบุ file handle ที่ถูกเปิดไว้แล้วไว้หน้าข้อมูลที่ต้องการจะเขียนลงแฟ้มข้อมูล ตัวอย่างถ้าต้องการที่จะเก็บข้อความไว้ในแฟ้มข้อมูลที่มี file handle ว่า INFO จะใช้คำสั่งว่า

```
print INFO "Test with test data.. Test write to file..\n";
```

หากต้องการที่จะติดต่อกับสแตนด์ดาร์ดอินพุท (standard input เช่น keyboard) และสแตนด์ดาร์ดเอ้าท์พุท (standard output เช่น จอภาพ) สามารถทำได้โดยใช้คำสั่งดังต่อไปนี้

```
open(INFO,'-');             # Open standard input
open(INFO,'>-');            # Open standard output
```

จากโปรแกรมตัวอย่างข้างต้นข้อมูลจะถูกเก็บไว้ในแฟ้มข้อมูล แฟ้มข้อมูลนี้จะถูกอ้างอิงโดย handle ที่ชื่อว่า INFO ถ้าหากว่าต้องการอ่านข้อมูลจาก INFO ในภาษา เพิร์ล สามารถทำได้โดยใช้เครื่องหมายวงเล็บน้อยกว่ามากกว่า <...> ดังตัวอย่างคำสั่งต่อไปนี้

```
@lines = ;
```

จะอ่านข้อมูลจาก filehandle INFO เข้าสู่ตัวแปรอาร์เรย์ ชื่อว่า @lines ข้อควรจำ จะอ่านข้อมูลจากแฟ้มข้อมูลทีเดียวทั้งหมดรอบเดียว ทั้งนี้เพราะว่าเมื่ออ่านมาแล้วจะนำมาเก็บไว้ในตัวแปรแบบอาร์เรย์ ถ้าหากว่าเปลี่ยนตัวแปรแบบอาร์เรย์เป็นตัวแปรแบบสเกล่าร์ ชื่อว่า $lines เมื่อมีการอ่านข้อมูลจากแฟ้มข้อมูล ข้อมูลก็จะถูกอ่านเข้ามาเพียงบรรทัดเดียวเท่านั้น

2.4 โปรเซส, โปรแกรม, เดมอน (Process, Program, Daemon)

เวลาที่จะเขียนซอร์สโปรแกรมภาษาซี โปรแกรมเมอร์สร้างไฟล์นั้นบนดิสก์ที่มีประโยคภาษาซีที่ทำงานตามต้องการ โดยในแต่ละโปรแกรมก็จะมีรายละเอียดต่างๆ กัน ซึ่งโดยปกติแล้วนั้นจะมีนามสกุลเป็น .c

ต่อมาคอมไพเลอร์ภาษาซีจะแปลซอร์สภาษาซีนั้นไปเป็นออบเจ็คไฟล์ แล้วลิงก์ออบเจ็คไฟล์แต่ละอันเข้าด้วยกันเป็นโมดูลที่สามารถเอ็กซ์ซีคิวได้ โดยโปรแกรมก็คือไฟล์ที่เก็บโมดูลที่เอ็กซ์ซีคิวได้เหล่านั้นนั่นเอง เมื่อโปรแกรมถูกรัน ระบบปฏิบัติการจะทำสำเนาโมดูลเหล่านั้นไปยังหน่วยความจำหลักในรูปของโปรแกรมอิมเมจ โดยโปรเซสก็คือโปรแกรมที่กำลังรันอยู่ในหน่วยความจำนั่นเอง ซึ่งแต่ละโปรเซสจะมีแอดเดรสและสถานะของตนเอง

เวลาที่โปรแกรมจะกลายเป็นโปรเซสก็คือ เมื่อระบบปฏิบัติการอ่านโปรแกรมเข้าหน่วยความจำแต่การจอง (allocation) หน่วยความจำให้กับโปรแกรมนั้นยังไม่ถือว่าเป็นการสร้างโปรเซส โปรเซสจะต้องมี ID (Process ID) ซึ่งระบบปฏิบัติการใช้แยกความแตกต่างของแต่ละโปรเซส

กล่าวคือเมื่อระบบปฏิบัติการเพิ่มเติมข้อมูลของโปรแกรมลงในส่วนเคอร์เนลดาต้า (Kernel data) และจัดสรรทรัพยากรของระบบให้แก่ให้กับโปรแกรมโค้ดแล้วโปรแกรมก็จะกลายเป็นโปรเซส ซึ่งบางคนเรียกว่า เฮฟวี่เวจท์โปรเซส(Heavyweight Process) ซึ่งจะตรงข้ามกับไลท์เวจท์โปรเซส (Lightweight Process) หรือ เทร็ด(Thread)

2.4.1 Process ID

UNIX จะระบุโปรเซสโดยใช้เลขที่ไม่ซ้ำเรียกว่า Process ID โปรเซสที่สร้างโปรเซสใหม่เรียกว่า พาเรนท์ (parent) ของโปรเซสที่ถูกสร้างซึ่งจะเรียกว่า ไชลด์(child) ของโปรเซสที่เป็น พาเรนท์

```
#include <sys/types.h>
#include <unistd.h>

pid_t getpid(void);
pid_t getpid(void);
```

แต่ละโปรเซสจะมียูสเซอร์เป็นเจ้าของ(owner) หากเจ้าของโปรเซสนั้นมีสิทธิอย่างไรโปรเซสนั้นก็จะมีสิทธิตามนั้น โดยแต่ละยูสเซอร์จะมีหมายเลขประจำตัวเรียกว่า user ID โปรเซสจะรู้หมายเลขของเจ้าของโปรเซสโดยใช้คำสั่ง getuid โดยโปรเซสจะมีสิทธิตามค่าของ effective user ID ซึ่ง euid นี้จะเปลี่ยนแปลงได้ระหว่างเอ็กซ์ซีคิว

```
#include <sys/type.h>
#include <unistd.h>

uid_t getuid(void);
uid_t geteuid(void);
```

ตัวอย่าง โปรแกรมจะพิมพ์ process ID , parent process ID ของมัน และ user ID ของเจ้าของ

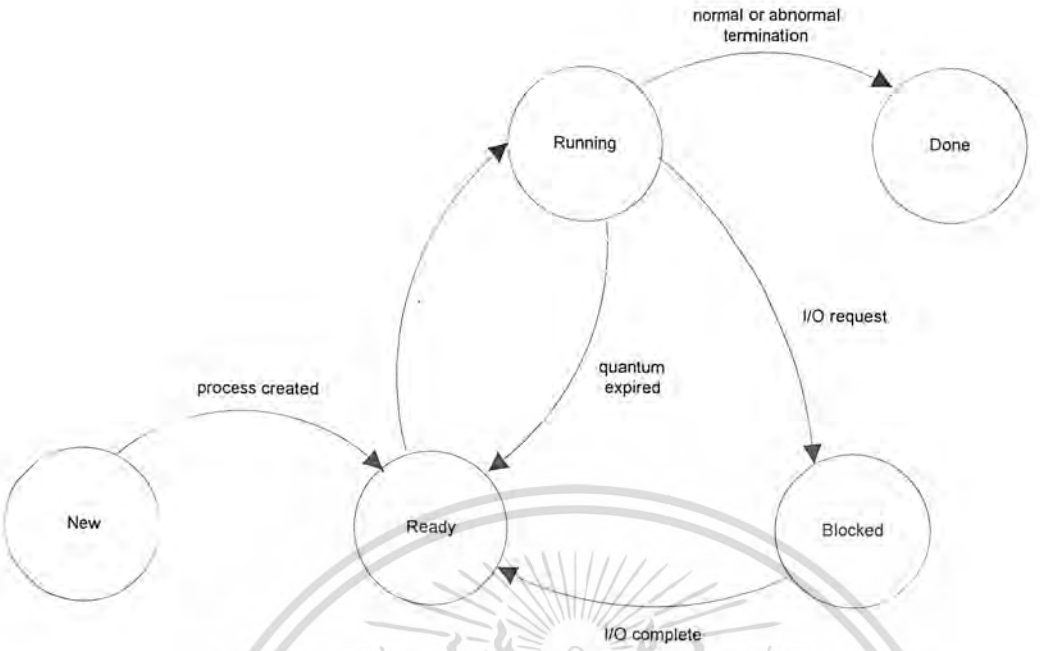
```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>

void main(void)
{
        printf("Process ID: %ld\n",(long)getpid());
        printf("Parent process ID: %ld\n", (long)getppid());
        printf("Parent process ID: %ld\n" (long)getuid());
}
```

2.4.2 Process state

โปรเซสจะมีสถานะเปลี่ยนแปลงตามเวลาโดยจะขึ้นกับกิจกรรมที่โปรเซสกำลังทำอยู่ โดยระบบปฏิบัติการส่วนใหญ่จะมีสถานะของโปรเซสดังตารางและในรูปไดอะแกรมจะแสดงการเปลี่ยนแปลงของโปรเซสจากสถานะหนึ่งไปยังอีกสถานะหนึ่ง

สถานะ	ความหมาย
New	โปรเซสกำลังถูกสร้างขึ้น
Running	คำสั่งของโปรเซสกำลังถูกรัน
Blocked	โปรเซสกำลังคอยอีฟเวนต์ ตัวอย่างเช่น I/O
Ready	โปรเซสกำลังคอยที่จะถูกส่งเข้าไปรัน
Done	โปรเซสเสร็จงานและคืนทรัพยากรให้กับระบบ

ตารางที่ 2.17 แสดงสถานะต่างๆของโปรเซสและความหมาย



รูปที่ 2.1 แสดงสถานะต่างๆของโปรเซส

เมื่อโปรแกรมกำลังกลายเป็นโปรเซสกล่าวได้ว่ามันกำลังอยู่ในสถานะ new เมื่อการเปลี่ยนแปลงดังกล่าวเสร็จสิ้นโปรเซสจะถูกนำไปใส่คิวเพื่อคอยรันเรียกว่าสถานะ ready เมื่อโปรเซสถูกจัดให้รันโปรเซสก็จะอยู่ในสถานะ running ซึ่งโปรเซสจะถูกเอ็กซ์ซีคิวในซีพียู

โปรเซสจะอยู่ในสถานะ blocked เมื่อโปรเซสคอยอีฟเวนท์ (event) ซึ่งเรียกอีกอย่างหนึ่งว่าโปรเซสกำลังหลับอยู่(sleep) หรืออีกนัยหนึ่งโปรเซสจะเข้าสู่ สถานะ blocked เมื่อโปรเซสทำการร้องขอ I/O

การที่มีการหยุดโปรเซสหนึ่งซึ่งกำลังอยู่แล้วเปลี่ยนเป็นอีกโปรเซสหนึ่งเรียกว่าการทำ คอนเท็กซ์สวิทช์(context switch) คอนเท็กซ์ ของโปรเซสได้แก่ข้อมูล(information) ของโปรเซสและเอ็นวิรอนเมนท์ (environment) ของโปรเซสเพื่อที่โปรเซสจะกลับมาทำงานอีกครั้งหลังการสวิทช์ ซึ่ง ส่วนเอ็กเซ็กคิวเทเบิล (exectutable) สแต็ก(stack) รีจิสเตอร์(Register) และโปรแกรมเคาน์เตอร์ก็เป็นส่วนประกอบของคอนเท็กซ์

2.4.3 การสร้างโปรเซส และการ fork ของ Unix

ในระบบ Unix การสร้างโปรเซสทำได้โดยเรียกคำสั่ง fork โดยก็อปปี้เม็มโมรีอิมเมจ (memory image) ของพาเรนท์ ซึ่งทั้งสองโปรเซสจะยังคงเอ็กซ์ซีคิว ต่อไปหลังจากทำคำสั่ง fork แล้ว

```
#include <sys/types.h>
#include <unistd.h>

pid_t fork(void);
```

fork จะรีเทิร์นค่า 0 แก่โปรเซสไชลด์ และรีเทิร์น process ID ของโปรเซสไชลด์แก่พาเรนท์ ซึ่งจากจุดนี้เราสามารถทำให้ไชลด์และพาเรนท์ทำงานแยกจากกันได้

ตัวอย่าง หลังจากทำการ fork แล้ว ไชลด์ และพาเรนท์ จะแสดง process ID ของตัวเอง

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>

if ( (childpid =fork() ) == 0) {
        fprintf(stderr,"I am the child , ID = %ld\n", (long)getpid());
                /*      child code goes here        */
} else if (childpid > 0) {
        fprintf(stderr,"I am the parent, ID = %ld\n", (long)getpid());
                /*      parent code goes here       */
}
```

จากตัวอย่างโปรเซสเดิมจะได้ค่า childpid เป็นค่าของ process ID ของโปรเซสไชลด์มีค่าที่ไม่เท่ากับศูนย์และจะทำคำสั่ง fprintf อันที่สอง ส่วนโปรเซสไชลด์จะมีค่า childpid เท่ากับศูนย์และจะทำ fprintf อันแรกโดยค่าเอาท์พุทอาจจะเรียงลำดับกันอย่างไรก็ได้เนื่องจากโปรเซสทั้งสองนั้นทำงานพร้อมกัน

ตัวอย่าง การสร้างโปรเซสแบบลูกโซ่ (chain process)

```
#inlclude <stdio.h>
#include <sys/types.h>
#include <unistd.h>

int i;
int n;
pid_t childpid;
for (i=1;i<n;++i)
        if (childpid = fork())
                break;
fprintf(stderr,"This is process %ld with parent  %ld\n"), (long)getpid(), (long)getppid());
```

ในการเรียกคำสั่ง fork แต่ละครั้งพาเรนท์จะได้ childpid ที่ไม่เป็นศูนย์และจะออกจากลูป ส่วนของไชลด์จะมีค่าเป็นศูนย์และจะเป็นพาเรนท์ในลูปต่อไป

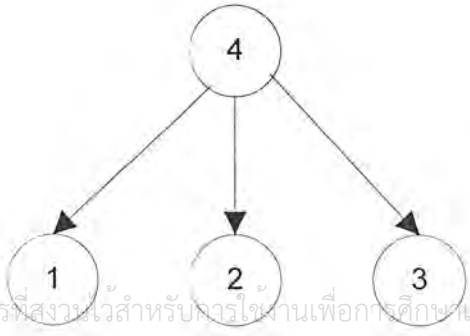
ถ้า n = 4 จะแสดงได้ดังรูป โปรเซสไชลด์จะ fork ต่อไปเรื่อยๆ

รูปที่ 2.2 แสดงการ fork โปรเซส

ตัวอย่าง การสร้างโปรเซสแบบใบพัด (fan of process)

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>

int i;
int n;
pid_t childpid;

for (i=1;i<n;++i)
        if ((childpid=fork()) <= 0)
                break;
fprintf(stderr,"This is process %ld with parent %ld\n", (long)getpid(), (long)getpid());
```

จากรูปจะแสดงการสร้างโปรเซสโดยพาเรนท์ จะสร้างโปรเซสไชลด์เป็นจำนวน n-1 โปรเซส



รูปที่ 2.3 แสดงการสร้างโปรเซส
เป็นจำนวน n -1

2.4.4 คำสั่งซิสเต็มคอล wait

เมื่อพาเรนท์สร้างโปรเซสไชลด์ โปรเซสทั้งสองจะแยกกันรันไปพร้อมๆ กันจากจุดที่เรียกใช้คำสั่ง fork ถ้าหากพาเรนท์ต้องการคอยจนโปรเซสจบการทำงานสามารถทำได้โดยใช้คำสั่ง wait หรือ waitpid

```
#include <sys/types.h>
#include <sys/wait.h>

pid_t wait(int *stat_loc):
pid_t waitpid(pid_t pid, int *stat_loc, int options);
```

คำสั่ง wait นั้นจะทำให้ผู้เรียกคอยจนไชลด์จบการทำงาน หรือจนกระทั่งได้รับสัญญาณ(signal) คำสั่ง wait จะรีเทิร์นทันทีถ้าหากว่าไม่มีโปรเซสไชลด์อยู่เลย ถ้าหากว่าเป็นการรีเทิร์นเพราะโปรเซสไชลด์จบการทำงานจะรีเทิร์นค่าของ process ID ของโปรเซสไชลด์ไม่เช่นนั้นจะรีเทิร์น –1 โดยถ้าหากว่าไม่มีโปรเซสเหลืออยู่จะให้ errno จะเท่ากับ ECHILD และถ้าได้สัญญาณอินเตอรัปท์จะเท่ากับ EINTR

ตัวอย่าง โปรแกรมแสดงการตรวจสอบสถานะการจบ(exit status)ของโปรเซสไชลด์

```
#include <sys/types.h>
#include <sys/wait.h>
#include <errno.h>

pid_t child;
int status;

while (((child = wait(&status)) == -1) && (errno == EINTR))
        ;
if (child == -1)
        perrer("Could not wait for child");
else if (!status)
        printf("Child %ld terminated normally, return status is zero\n",(long)child);
else if (WIFEXITED(status))
        printf("Child %ld terminated normally, return status is %d\n", (long)child, WEXITSTATUS
        (status));
else if (WIFSIGNALED(status) )
        printf("Child %ld terminated due to signal not caught\n",(long)child);
```

ตัวอย่าง โปรแกรมแสดงการใช้งานคำสั่ง wait แบบง่าย

```
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>
#include <stdio.h>

void main(){
        pid_t childpid;
        int status;

        if ( (childpid = fork() ) == -1) {
                perror("The fork failed");
                exit(1);
        }else if (childpid ==0)
                fprintf(stderr,"I am the child with pid = %ld\n", (long)getpid() );
        else if (wait(&status) != childpid)
                fprintf(stderr,"A signal must have interrupted the wait\n");
        else
                fprintf(stderr,"I am the parent with pid = %ld and child pid = %ld\n",
                        (long)getpid, (long)childpid );
        exit(0);
}
```

ตัวอย่าง การใช้งานคำสั่ง wait เพื่อคอยโปรเซสไชลด์จบการทำงาน

```
#include <sys/types.h>
#include <sys/wait.h>
int status;
pid_t childpid;

while (childpid != wait(&status))
        ;
```

จากตัวอย่างนี้จะเห็นว่าอาจเป็นไปได้ที่คำสั่ง wait จะไม่รีเทิร์น childpid เพราะอาจมีเออเรอร์ได้ โดยจะถูกเก็บใน errno เช่นถ้าเท่ากับ EINTR ก็จะหมายถึงถูกอินเตอรัปท์

ตัวอย่าง แก้ไขจากตัวอย่างที่ผ่านมาเพื่อให้มีการตรวจสอบ errno ด้วย

```
#include <sys/types.h>
#include <sys/wait.h>
int status;
pid_t childpid;

while (childpid != wait(&status))
        if ((childpid == -1) && (errno != EINTR) )
                break;
```

คำสั่ง waitpid จะใช้ในกรณีที่คอยโปรเซสไชลด์ที่เฉพาะเจาะจง โดย pid จะเป็นค่าของ process ID ของโปรเซสที่เราคอย และจะเป็น –1 ถ้าหากเป็นการคอยโปรเซสใดๆ

ตัวอย่าง โปรแกรมจะคอยโปรเซสไชลด์ตัวใดก็ได้จบการทำงาน

```
#include <sys/types.h>
#include <sys/wait.h>
#include <errno.h>
int status;
pid_t waitreturned;

while (waitreturned = waitpid(-1,&status, WNOHANG) )
        if ((waitreturned == -1) && (errno != EINTR) )
                break;
```

เมื่อ waitpid รีเทิร์นค่าเป็น 0 แสดงว่ามีโปรเซสไชลด์เหลืออยู่และถ้าหากว่าพาเรนท์ จบการทำงานโดยไม่คอยโปรเซสไชลด์ จะทำให้โปรเซสไชลด์กลายเป็น ซอมบี้ (sombie) และพาเรนท์ ของมันจะถูกกำหนดเป็น 1 ซึ่งก็คือ init นั่นเอง

2.4.5 คำสั่งซิสเต็มคอล exec

คำสั่ง fork จะทำก็อปปี้ของโปรเซสที่เรียกมันส่วนคำสั่ง exec จะเรียกเอ็กซ์ซีคิวโปรแกรมอื่น โดยส่วนใหญ่คำสั่งสองอันนี้จะถูกเรียกใช้ร่วมกันในลักษณะ fork-exec

ตัวอย่าง โปรแกรมนี้จะเรียกรันคำสั่ง ls –l

```
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>

void main(){
        pid_t childpid;
        int status;

        if ((childpid = fork() ) == -1){
                perror()"Error in the fork";
                exit(1);
        }else if (childpid == 0){                    /*      child code      */
                if (execl("/usr/bin/ls","ls","-l",NULL) < 0{
                        perror("Exec of ls failed");
                        exit(1);
                }
        }else if (childpid != wait(&status) )        /*      parent code     */
                perror("A signal occurred begore the chld exited");
        exit(0);
}
```

```
#include <unistd.h>
int execl(const char *path, const char *arg0, … , const *argn, char * /*NULL*/);
int execle(const char *path, const char *arg0, … , const *argn, char * /*NULL*/, char *const envp[]);
int execlp(const char *file, const char *arg0, … , const *argn, char * /*NULL*/);
int execv(const char *path, char *const argv[]);
int execvp(const char *file, char *const argv[]);
int execve(const char *path, char *const argv[],char *const envp[]);
```

คำสั่ง execl จะต้องระบุค่าอาร์กิวเมนต์มาเลยต่างกับ execp ซึ่งจะส่งค่าผ่านอาเรย์ โดยถ้าเป็น คำสั่ง execp จะระบุเป็นไฟล์แล้วชิสเต็มคอลจะไปหาในพาร์ธและคำสั่ง exece จะมีค่าตัวแปรเอนวิรอนเมนท์ด้วย

2.4.6 Background Process and Daemon

เชลล์เป็นคอมมานด์อินเตอร์พรีเตอร์ซึ่งจะคอยรับคำสั่งจากสแตนดาร์ดอินพุท แล้ว fork โปรเซสเพื่อเอ็กซ์ซีคิวคำสั่งนั้น แล้วคอยโปรเซสไชลด์จบการทำงาน

เชลล์ส่วนใหญ่จะสามารถเพิ่มสัญลักษณ์ & เพื่อให้ทำงานเป็นแบ็กกราวด์(background) ซึ่งเชลล์จะไม่คอยให้แบ็กกราวด์ โปรเซสจบการทำงานจะกลับมาคอยรับคำสั่งเลย

เดมอน(daemon) เป็นโปรแกรมที่ทำงานแบ็กกราวด์โดยอัตโนมัติและทำงานในแบ็กกราวด์เท่านั้นจากตัวอย่างคำสั่ง setsid ใช้เพื่อกำหนดให้โปรเซสนั้นเป็นเซสชั่นใหม่เพื่อให้ไม่ต้องรับคีย์ Ctrl-C

ตัวอย่าง โปรแกรม runback จะรับคำสั่งจากคอมมานด์ไลน์แล้วเอ็กซ์ซีคิวเป็นแบ็กกราวด์

```
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
int makeargv(char *s, char *delimiters, char ***argvp);
void main(int argc, char *argv[]){
        char **myargv;
        char delim[] = " \t";
        pid_t childpid;
        if (arfc != 2){
                fprintf(stderr, "Usage: %s string\n", argv[0]);
                exit(1);
        }
        if ( (childpid = fork() ) == -1){
                perror("The fork failed");
                exit(1);
        }else if (childpid == 0){              /*Child process becomes a background process*/
                if (setsid() == -1)
                        perror("Could not becomes a session leader");
                else if (makeargv(argv[1], delim, &myargv) < 0)
                        fprintf(stderr,"Argument array could not be constructed\n");
                else if (execvp(myargv[0], &myargv[0]) < 0)
```

```
                perror("The exec of command failed");
            exit(1);                    /*Child should never return*/
        }
        exit(0);                        /*Parent exits*/
}
```

โปรแกรม runback จะใช้ setsid ในการสร้างเซสชั่นใหม่เพื่อที่จะได้ไม่มีเทอร์มินอลควบคุม Session ID ก็คือตัวบอกว่าโปรเซสใดจะรับคีย์จากเทอมินอลซึ่งจะตอบสนองกับคีย์ Ctrl-c

2.5 Cron and crontab

Cron เป็นระบบที่จะจัดการเอ็กซ์ซีคิวคำสั่งตามเวลาที่กำหนดไว้ล่วงหน้าโดยตัว crond (ครอนเดมอน) จะถูกรันจากไฟล์ /etc/rc หรือ /etc/rc.local โดยตัวเดมอนนี้เมื่อเริ่มต้นทำงานจะอ่านค่าเริ่มต้นจากไฟล์ /var/spool/cron ซึ่งเป็นส่วนที่ยูสเซอร์แต่ละคนสามารถสร้างตารางเวลาของแต่ละคนได้ โดยชื่อไฟล์จะมีชื่อเดียวกันกับ ล็อกอินเนม (login name) ของยูสเซอร์ที่สร้าง ส่วนอีกไฟล์หนึ่งจะเป็นการกำหนดตารางเวลาของระบบ ซึ่งจะอยู่ในไฟล์ /etc/crontab โดยผู้ที่จะจัดการกับไฟล์นี้ได้ต้องเป็น root เท่านั้น

crontab

เป็นไฟล์ที่ใช้กำหนดเวลาล่วงหน้าของระบบ อยู่ในพาร์ธ /etc/crontab โดยคำสั่งที่รันจาก cron ของระบบนี้จะมีสิทธิของ root

ไฟล์ crontab มีข้อกำหนดการเขียนคือ บรรทัดที่ว่าง และบรรทัดที่ขึ้นต้นด้วย # จะถือว่าเป็นคอมเมนต์ ในไฟล์ crontab สามารถกำหนดค่าตัวแปรในรูปแบบ

```
name = value
```

และในส่วนของการกำหนดตารางการทำงาน จะเป็นในรูปแบบ ดังนี้

```
minutes  hour   day of month   month   day of week   [user]   command
```

โดยค่าต่างๆ จะมีช่วงกำหนดดังนี้คือ

ฟิลด์	ค่าที่เป็นไปได้	
minutes	0-59	
hour	0-23	
day of month	0-31	
month	0-12	(หรือเป็นชื่อเดือนก็ได้)
day of week	0-7	(0 หรือ 7 เป็นวันอาทิตย์ หรือเป็นชื่อวันก็ได้)

ค่าของฟิลด์ ถ้าหากเป็น * ก็จะหมายถึงทุกค่าที่เป็นไปได้

สามารถกำหนดค่าเป็นช่วงได้ โดยใช้ เครื่องหมาย – ตัวอย่างเช่น hour เป็น 8-11 ก็หมายถึงเวลา 8,9,10,11 และ เราก็สามารถ กำหนดค่าเป็นลิสท์ได้เช่น 1, 2, 5, 9 หรือ 1-4, 8-12 ก็ได้

เราสามารถกำหนดเป็นสเต็ปได้ เช่น สำหรับ hour กำหนดเป็น 0-23/2 ก็จะหมายถึง 0, 2. 4, 6, 8, 10, 12, 14, 16, 18, 20, 22 ซึ่งหมายถึง " ทุก 2 ชั่วโมง" อาจจะเขียนง่ายๆ ได้ว่า */2 และ เครื่องหมาย % มีความหมายเท่ากับ เครื่องหมายขึ้นบรรทัดใหม่ (new line)

ฟิลด์ month และ day of week จะสามารถใช้ชื่อเป็นภาษอังกฤษสามตัวแรก เช่น วันอาทิตย์ ก็จะเป็น sun เดือนธันวาคมก็จะเป็น Dec โดยจะไม่สนใจว่าเป็นตัวพิมพ์เล็กหรือพิมพ์ใหญ่

ตัวอย่าง

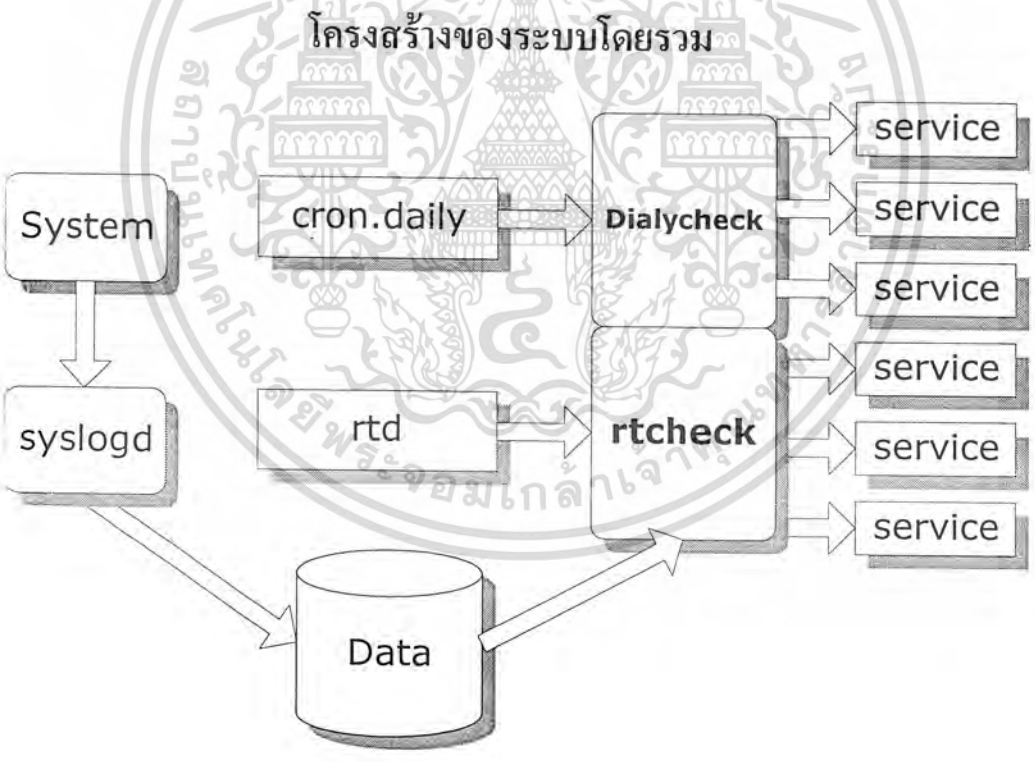
30	4	1,15	*	5	-หมายถึงคำสั่งจะรันที่เวลา 4:30 ทุกวันที่ 1 และ 15 ของเดือนรวมถึงทุกวันศุกร์ด้วย
5	0	*	*	*	-หมายถึงรันทุกวันเวลา 5 นาทีหลังเที่ยงคืน
15	14	1	*	*	-คือรันเวลา 2:15 pm ของวันที่ 1 ของทุกเดือน
0	22	*	*	1-5	-คือ รันในวันธรรมดา เวลายี่สิบสองนาฬิกา
23	0-23/2	*	*	*	-รันที่เวลา 23 นาทีหลังเที่ยงคืน แล้วรันทุกๆ สองชั่วโมง ทุกวัน
5	4	*	*	sun	-รันทุกวันอาทิตย์เวลา 4:05

บทที่ 3
การคำนวณ สร้างและการออกแบบ

3.1 เป้าหมายของโครงงาน

ระบบที่ต้องการต้องเป็นระบบที่สามารถจะช่วยอำนวยความสะดวกในงานของผู้ดูแลระบบในส่วนที่เป็นการตรวจสอบล็อกไฟล์ ซึ่งเป็นงานที่มีความซ้ำซาก จำเจ โดยจะทำงานด้านการรายงานสถานะของระบบผ่านระบบเมล์เป็นประจำทุกวัน โดยในรายงานนั้นจะต้องเป็นการสรุปผลแบ่งเป็นหมวดหมู่ที่อ่านและตีความได้ง่าย แต่ถ้าหากว่าเกิดเหตุการณ์ที่สำคัญและมีผลกระทบรุนแรงต่อระบบ ก็จะต้องสามารถรายงานผู้ดูแลระบบได้ทันทีโดยผ่านหน้าจอที่ผู้ดูแลระบบทำการล็อกอินอยู่ , ทางอินเตอร์เนต เพจเจอร์ และ ทางไอซีคิว (ICQ)

3.2 โครงสร้างของระบบ

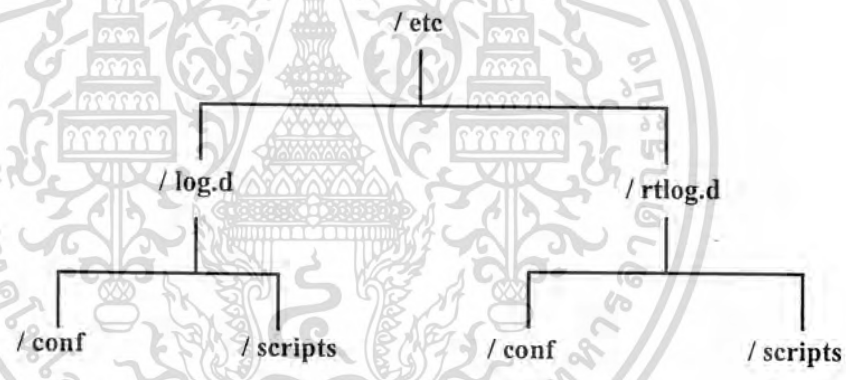


รูปที่ 3.1 แผนภาพเป็นการแสดงโครงสร้างการทำงานของระบบโดยรวม

เนื่องจากหัวใจหลักของระบบ คือ ล็อกไฟล์ ซึ่งสร้างมาจาก syslogd โดยแต่ละไฟล์มีรายละเอียดแบ่งแยกตามประเภทที่กำหนดใน /etc/syslog.conf ในระบบจะมีการตรวจจับ แบ่งเป็น 2 ส่วนใหญ่ คือ

ส่วนวิเคราะห์และสร้างรายงานประจำวันผ่านระบบเมล์ที่ชื่อว่า "Daily check" ซึ่งจะถูกเรียกโดยการจัดตารางเวลาผ่านโปรแกรม "crond" ให้เรียกเซอร์วิสมาเนเจอร์(service manager) ที่ชื่อ "dailycheck.pl" ขึ้นมาทำงาน ซึ่งตัวเซอร์วิสมาเนเจอร์จะเรียกเซอร์วิสที่เกี่ยวข้องกับล็อกไฟล์ต่างๆมาวิเคราะห์และสรุปเป็นรายงานประจำวันให้แก่ผู้ดูแลระบบ

อีกส่วนหนึ่ง จะเป็นส่วนที่ทำงานอยู่ตลอดเวลา โดยในส่วนนี้จะมีเดมอนคอยตรวจสอบล็อกไฟล์ซึ่งจะรันเป็นแบ็กกราวด์อยู่ตลอดเวลาที่ชื่อว่า "rtd" จะมีการกำหนดกลุ่มของล็อกไฟล์ที่มีความสำคัญต่อระบบไว้ล่วงหน้า เพื่อให้ส่วนนี้ทำการตรวจสอบเมื่อมีเมสเซสใหม่เกิดขึ้นกับล็อกเหล่านี้ ก็จะทำการเรียกเซอร์วิสมาเนเจอร์ที่ชื่อว่า "rtcheck.pl" ขึ้นมาทำงาน โดยจะเรียกเซอร์วิสที่เกี่ยวข้องกับล็อกไฟล์นั้นขึ้นมา โดยถ้าหากตรวจจับได้ว่ามีเหตุการณ์ผิดปกติหรือเหตุการณ์สำคัญตามที่ได้กำหนดไว้ ก็จะทำการรายงานต่อผู้ดูแลระบบทันที



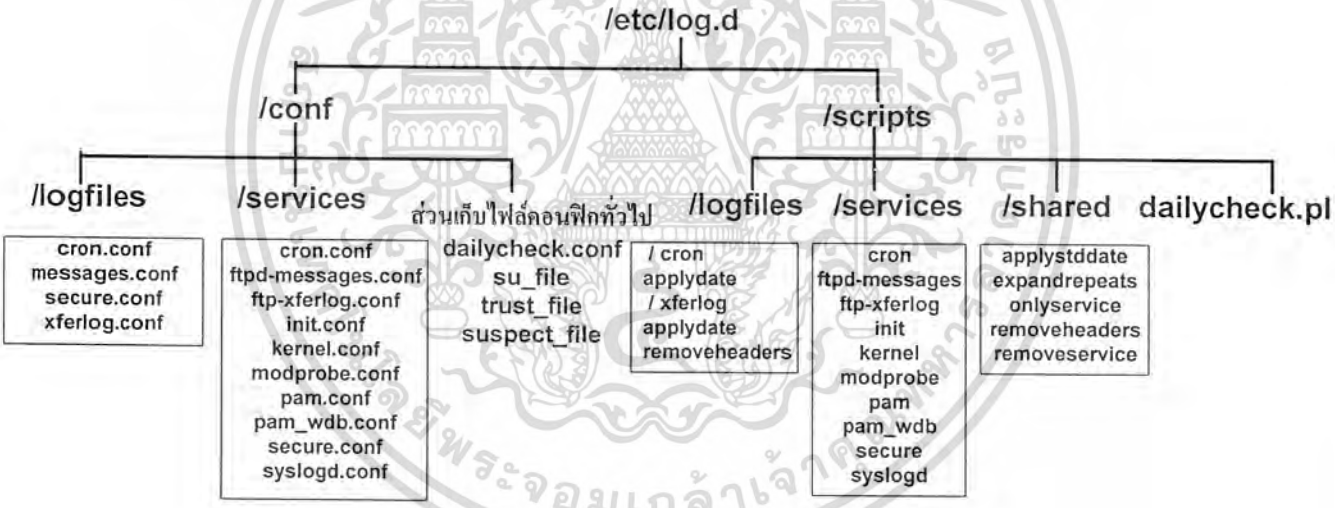
รูปที่ 3.2 แสดงไดเรกทอรี่ของระบบทั้งหมด

3.3 ส่วนวิเคราะห์และสร้างรายงานประจำวัน (Daily check)

3.3.1 ในส่วนวิเคราะห์และสร้างรายงานประจำวัน

เป็นส่วนที่ทำหน้าที่ในการจัดการเกี่ยวกับการตรวจสอบล็อกไฟล์ของระบบโดยสำรวจดูถึงความผิดพลาดต่างๆที่เกิดขึ้นและมีการบันทึกลงในล็อกไฟล์ หลังทำการประมวลผลล็อกไฟล์ต่างๆของระบบแล้วก็ทำการส่งเมล์หาผู้ดูแลระบบซึ่งจะทำเช่นนี้ทุกวัน เพื่อเป็นส่วนช่วยผู้ดูแลระบบในงานการตรวจสอบล็อกไฟล์ของผู้ดูแลระบบ ซึ่งนับเป็นงานที่น่าเบื่อเสียเวลาจำเจมากและทำให้ความผิดพลาดที่เกิดขึ้นในการตรวจสอบล็อกไฟล์นั้นลดลงและประหยัดเวลามากขึ้น ผู้ดูแลระบบสามารถนำเวลาไปจัดการเรื่องอื่นๆได้

ในส่วนนี้จะใช้ภาษาเพิร์ลเป็นสคริปต์ในการทำการส่งเมล์ประจำวัน ด้วยเหตุผลที่ว่าภาษาเพิร์ลเป็นภาษาที่มีจุดเด่นในเรื่องการทำการตรวจสอบรูปแบบข้อความ (Pattern Matching) ที่ดีมาก และงานที่ส่วนวิเคราะห์และสร้างรายงานประจำวันทำส่วนใหญ่เป็นการตรวจสอบล็อกไฟล์ที่เกี่ยวกับความปลอดภัยของระบบ (ตามที่มีการกำหนดไว้ล่วงหน้า)



รูปที่ 3.3 แสดงไดเรกทอรี่ของส่วนวิเคราะห์และสร้างรายงานประจำวันโดยละเอียด

3.3.2 โครงสร้างของส่วนวิเคราะห์และสร้างรายงานประจำวัน

ไดเรกทอรี่หลักของส่วนนี้จะอยู่ที่สับไดเรกทอรี่ /etc/log.d/ ดังรูปที่ 3.3 โดยจะแบ่งออกเป็นอีก 2 สับไดเรกทอรี่ คือ /conf และ /scripts โดยแบ่งแยกกันตามหน้าที่และความหมายมีรายละเอียดในแต่ละสับไดเรกทอรี่ดังนี้

3.3.2.1 /etc/log.d/conf

เป็นสับไดเรกทอรี่ที่จะบรรจุไฟล์ที่เป็นใช้สำหรับให้ผู้ใช้งานส่วนนี้สามารถกำหนดค่าต่างๆได้ตามต้องการ จะแบ่งออกเป็นอีก 3 ส่วนดังนี้

- **ส่วนเก็บไฟล์คอนฟิกทั่วไป**

  ซึ่งจะมีอยู่ 4 ไฟล์ดังนี้

  1) dailycheck.conf

  เป็นไฟล์ที่เป็นการกำหนดค่าเริ่มต้นต่างๆให้กับส่วนนี้ในการทำการส่งเมล์ให้กับผู้ดูแลระบบประจำวัน และเป็นค่าเริ่มต้นให้ในกรณีที่ผู้ดูแลระบบสั่งให้ส่วนนี้กระทำการทันทีโดยอาจจะมีการเปลี่ยนค่าแค่บางค่าเพื่อความต้องการในขณะนั้น แต่ค่าที่เหลือก็ยังคงใช้ค่าที่อยู่ในไฟล์นี้ โดยความหมายของค่าต่างๆมีดังนี้

  - LogDir = /var/log เป็นการกำหนดค่าสับไดเรกทอรี่ในการค้นหาล็อกไฟล์ ในที่นี้กำหนดให้เป็น /var/log
  - MailTo = root เป็นการกำหนดอีเมล์แอดเดรสของผู้ที่จะทำการส่งรายงานประจำวันไปให้ โดยอาจจะกำหนดเป็นอีเมล์แอดเดรสแบบเต็มก็ได้
  - Print = No หาก Print ถูกเซ็ตให้เป็น Yes แล้วจะไม่ทำการเมล์ไปยังผู้ที่ระบุไว้ แต่จะทำการแสดงผลออกมาที่หน้าจอ
  - Save = /tmp/dailycheck หากมีการกำหนดค่านี้ไว้แล้วแทนที่จะทำการแสดงผลหรือเมล์แล้ว มันจะทำการส่งผลดังกล่าวไปไว้ยังที่ที่กำหนดไว้ เช่น /tmp/dailycheck
  - Range = All ค่าของ Range อาจเป็นได้ทั้ง 3 ค่าดังนี้ คือ today,yerterday,all ซึ่งหมายความถึงวันที่ต้องการให้ทำการประมวลผล โดยถ้าเป็น all จะหมายถึงให้ทำทั้ง file ไม่ต้องพิจารณาวันที่
  - Detail = Low เป็นค่าที่กำหนดถึงรายละเอียดของรายงาน โดยอาจเป็นได้ 3 ค่า ดังนี้ Low,Medium,High
  - Service = All เป็นการกำหนดให้ ทำเซอร์วิสที่กำหนด โดยอาจเป็นได้ 10 ค่าดังนี้ all,cron,ftpd-messages,ftpd-xferlog,init,kernel,modprobe,pam_pwdb,secure,syslogd
  - โดย all จะหมายถึงให้ทำทุก เซอร์วิส ส่วนที่เหลือจะเป็นเซอร์วิสต่างๆที่มีให้เลือกใช้

  2) su_file

  เป็นไฟล์ที่ผู้ดูแลระบบสามารถรายชื่อของผู้ใช้ที่มีสิทธิในการใช้คำสั่ง su เปลี่ยนสิทธิเป็นรูท โดยในการใช้งานนั้น ผู้ดูแลระบบสามารถทำการแก้ไขไฟล์นี้ได้ง่ายๆ ด้วยการเพิ่มเติมชื่อของผู้ใช้ระบบที่มีสิทธินั้นลงไปได้เลย ยกตัวอย่างเช่น ถ้าต้องการให้ผู้ใช้ชื่อ nueng , tee , korn สามารถใช้คำสั่งนี้ได้

  nueng

  tee

  korn

ก็จะปรากฏดังนี้ใน su_file และถ้าหากในแต่ละวันมีผู้ใช้คนใดที่ไม่ได้มีรายชื่ออยู่ในไฟล์นี้แต่ได้ทำการเปลี่ยนสิทธิตัวเองเป็นรูทด้วยคำสั่ง su ก็จะมีการรายงานในรายงานประจำวันว่าได้มีใช้คำสั่งนี้โดยผู้ใช้ที่ไม่มีสิทธิ

3) trust_file

เป็นไฟล์ที่ภายในจะประกอบด้วยรายชื่อของโฮสต์ที่ระบบเราให้ความเชื่อใจสามารถติดต่อด้วยเซอร์วิสต่างๆมายังระบบเราได้ ยกตัวอย่างเช่น

sunshine05.ce.kmitl.ac.th

sunshine20.ce.kmitl.ac.th

chaokhun.kmitl.ac.th

หากมีโฮสต์ใดที่ไม่ได้มีรายชื่ออยู่ใน trust_file นี้แล้วจะมีการแสดงรายชื่อโดยเรียงตามเวลาที่เกิดขึ้นต่อระบบ และรายชื่อของโฮสต์ที่ไม่ได้รับความเชื่อถือนี้จะปรากฏอยู่ในไฟล์ suspect_file เพื่อให้ผู้ดูแลระบบสามารถเอาไปพิจารณาเพื่อการตัดสินใจอีกครั้ง

4) suspect_file

เป็นไฟล์ที่ผู้ดูแลระบบจะสามารถมาตรวจดูได้ว่ามีโฮสต์ใดที่ไม่อยู่ในรายชื่อของโฮสต์ที่ระบบให้ความเชื่อถือแล้วมาขอบริการจากระบบบ้างและเมื่อผู้ดูแลระบบได้ทำการตรวจสอบแล้วต้องการจะเพิ่มรายชื่อของโฮสต์ใหม่เหล่านี้ลงใน trust_file ก็สามารถทำได้เลยเพราะมีการจัดเก็บเป็นรูปแบบเดียวกัน ยกตัวอย่างไฟล์ suspect_file

hardware09.ce.kmitl.ac.th

venus08.ce.kmitl.ac.th

alpha.tu.ac.th

- **/logfile**

ในสับไดเรกทอรี่นี้จะเก็บไฟล์ซึ่งเป็นรายละเอียดเกี่ยวกับการกำหนดค่าต่างๆให้กับล็อกไฟล์กรุ๊ป (log file group) ต้องลงท้ายด้วย .conf (ซึ่งในโครงงานนี้มี 4 ล็อกไฟล์กรุ๊ป) โดยในแต่ละล็อกไฟล์กรุ๊ปเราสามารถจะกำหนดสิ่งต่อไปนี้ได้

- ชื่อล็อกไฟล์จริงของระบบที่จะรวมไว้ในล็อกไฟล์กรุ๊ปนี้โดยจะกำหนดเป็นเส้นทางแบบสมบูรณ์ (absolute path) หรือไม่ก็ได้ แต่จะไปนำล็อกไฟล์นั้นมาจาก /var/log
- Archive โดยค่าที่กำหนดให้กับตัวแปร archive นั้นจะเป็นล็อกไฟล์ของระบบที่เราต้องการนำมาร่วมพิจารณาด้วย
- ชื่อโปรแกรมฟิลเตอร์ที่ต้องการให้ล็อกไฟล์ผ่านการจัดการก่อนการพิจารณาในขั้นต่อๆไป โดยในที่นี้สามารถกำหนด อาร์กิวเมนท์ ให้กับชื่อโปรแกรมนั้นๆได้ด้วย

- ตัวอย่างเช่น ไฟล์ secure.conf

```
LogFile = secure
Archive = secure.*
Archive = secure.*.gz
*ApplyStdDate =
```

ในบรรทัดแรกหมายความว่าให้ล็อกไฟล์กรุ๊ป secure นี้นำเอาล็อกไฟล์ของระบบที่ชื่อ secure มาใช้ในการพิจารณา

ใน 2 บรรทัดต่อมาเป็นการบอกว่าในล็อกไฟล์กรุ๊ป secure นี้ นำเอาล็อกไฟล์ของระบบที่มีชื่อเป็น secure.* และ secure.*.gz มารวมในการพิจารณาด้วย แต่จะนำมารวมเมื่อมีการกำหนดค่า archive เป็น 1 เท่านั้น

ในบรรทัดสุดท้ายเป็นการบอกให้ล็อกไฟล์กรุ๊ป secure นี้จะต้องผ่านการทำ ApplyStdDate (โดยในที่นี่ไม่มี อาร์กิวเมนท์)

โดยบรรทัดที่สำคัญที่สุดที่ต้องมีคือ บรรทัดที่บอกถึงล็อกไฟล์ ที่จะต้องนำมาพิจารณา ส่วนบรรทัดอื่นๆไม่จำเป็นต้องกำหนดก็ได้ ทั้งนี้ขึ้นอยู่กับรูปแบบของล็อกไฟล์นั้นๆ ว่าจำเป็นต้องมีการผ่านโปรแกรมบางตัวก่อนไปสู่เซอร์วิสหรือไม่

- /service

ในสับไดเรกทอรี่นี้จะเป็นที่เก็บไฟล์ซึ่งเป็นการกำหนดค่าต่างๆให้กับเซอร์วิสต้องลงท้ายด้วย .conf (ซึ่งในโครงงานนี้มี 10 เซอร์วิส) โดยแต่ละเซอร์วิสจะสามารถกำหนดค่าต่างๆได้ดังนี้

- ชื่อล็อกไฟล์กรุ๊ปที่จะนำมาผ่านเซอร์วิสนี้โดยกำหนดด้วยรูปแบบ LogFile = log-file group name ซึ่งต้องเป็นมีชื่อของล็อกไฟล์กรุ๊ปนี้ที่เป็น .conf อยู่ใน /etc/log.d/conf/logfile/*.conf
- ชื่อโปรแกรมที่ต้องการให้ล็อกไฟล์กรุ๊ปที่กำหนดข้างต้นผ่านการจัดการก่อนที่จะเข้าเซอร์วิส และสามารถกำหนดอาร์กิวเมนท์ได้
- ตัวแปรเอนวิรอนเมนท์ (environment) จะเป็นตัวแปรที่ผู้ใช้สามารถกำหนดค่าต่างๆที่ต้องการให้เป็นตัวแปรที่เซอร์วิสสามารถนำไปประมวลผลได้
- ตัวอย่างเช่น ftpd-messages.conf

```
Logfile = messages
*OnlyService = ftpd
$ftpd_ignore_unmatched = 0
```

ในบรรทัดแรกเป็นการกำหนดล็อกไฟล์กรุ๊ปให้กับเซอร์วิสโดยในที่นี้คือให้นำล็อกไฟล์ "messages" ของระบบมาทำการพิจารณาด้วยเซอร์วิส ftpd-messages

ในบรรทัดที่สอง หมายถึงให้ล็อกไฟล์ messages ผ่านโปรแกรม Onlyservice ก่อน โดยมีการผ่านค่า ftpd ไปให้โปรแกรมด้วย

ในบรรทัดสุดท้ายเป็นการกำหนดให้ตัวแปร $ftpd_ignore_unmatched ให้เป็นตัวแปรที่เซอร์วิสสามารถเรียกใช้ได้ โดยมีการผ่านค่า 0 ไปให้ด้วย

บรรทัดที่มีความสำคัญที่สุดในการกำหนดค่าไฟล์คอนฟิกกูเรชั่นให้กับเซอร์วิสก็คือ บรรทัดที่กำหนดล็อกไฟล์กรุ๊ป ส่วนบรรทัดอื่นๆไม่จำเป็น ทั้งนี้ขึ้นอยู่กับสคริปต์ของเซอร์วิสที่อยู่ใน /etc/log.d/scripts/service ด้วยว่าสามารถรองรับการกำหนดอื่นๆได้หรือไม่

3.3.2.2 /etc/log.d/scripts

แบ่งออกเป็น 4 ส่วนดังนี้

- **dailycheck.pl**

เป็นโปรแกรมหลักในการจัดการในเรื่องการส่งเมล์ ซึ่งรายละเอียดในเมล์นั้นจะเป็นอย่างไรก็ขึ้นอยู่กับปัจจัยต่อไปนี้

- ไฟล์ .conf ใน /etc/log.d/conf/logfile/*.conf นั้นกำหนดรายละเอียดไว้อย่างไร ซึ่งในโครงงานนี้จะมีอยู่ 4 ล็อกไฟล์กรุ๊ปให้เรียกใช้ได้
- ไฟล์ . conf ใน /etc/log.d/conf/service/*.conf นั้นได้กำหนดรายละเอียดไว้อย่างไร ซึ่งในโครงงานนี้จะมีอยู่ 9 เซอร์วิสให้เรียกใช้ได้
- ไฟล์ dailycheck.conf ซึ่งอยู่ที่ /etc/log.d/conf/dailycheck.conf ว่ามีการกำหนดค่าดีฟอลท์ไว้อย่างไร
- ในกรณีที่เป็นการเรียกใช้ที่เชลล์ (shell) แล้วมีการกำหนดค่าออปชั่น (option) ต่างๆ แล้ว รายละเอียดของการแสดงผลก็จะเปลี่ยนแปลงไปด้วย ดังรายละเอียดในภาคผนวก ข

- **/logfile**

เป็นสับไดเรกทอรี่ที่เก็บสคริปต์ของล็อกไฟล์บางล็อกไฟล์ที่ไม่มีรูปแบบเป็นแบบ syslog สแตนด์ตาร์ด ซึ่งในโครงงานนี้จะมีอยู่ 2 ล็อกไฟล์กรุ๊ป คือ

1) /cron

ภายใต้สับไดเรกทอรี่นี้จะมีไฟล์ที่เป็นเอ็กซ์ซีคิวไฟล์อยู่คือ

- *applydate* ซึ่งมีหน้าที่ในการเลือกเอาบรรทัดเฉพาะวันที่ผู้ใช้ต้องการทำการพิจารณาออกมาจากล็อกไฟล์ cron เช่น ถ้าวันนี้เป็นวันที่ 21 มีนาคม 1999 แล้วทำการประมวลผลประจำวัน ซึ่งถ้าใช้ค่า Range เป็น yesterday แล้วล็อกไฟล์ cron เมื่อผ่านโปรแกรมนี้ไปแล้วจะมีแต่เฉพาะบันทึกว่าวันที่ 20 มีนาคม 1999 เท่านั้น เพื่อเตรียมพร้อมในการนำไปพิจารณาในขั้นต่อไป

2) /xferlog

ภายใต้สับไดเรกทอรี่นี้จะมีไฟล์ที่เป็นเอ็กซ์ซีคิวไฟล์อยู่ คือ

- *applydate* เนื่องจากล็อกไฟล์ xferlog มีรูปแบบในการจัดเก็บวันที่ที่ไม่เหมือน syslog ไฟล์ทั่วไป จึงต้องมีโปรแกรมในการคัดเลือกวันขึ้นมาเฉพาะ โดยมีหลักการทำงานเหมือน applydate ของล็อกไฟล์ cron คือ จะพิจารณาขอบเขต (Range) ที่ผู้ใช้โดยกำหนดให้เป็นค่าในการเลือกวันที่

- *removeheaders* ก่อนที่จะนำล็อกไฟล์ xferlog ไปผ่านเซอร์วิสนั้นจะมาผ่านโปรแกรม removeheaders ก่อนเพื่อที่ทำการตัดส่วนที่เป็นวันที่และค่าเวลาทรานสเฟอร์ (transfer time)ทิ้งไป เพื่อความสะดวกในการพิจารณาจากเซอร์วิสต่อไป

- **/shared**

เป็นสับไดเรกทอรี่ที่เป็นที่เก็บโปรแกรมที่ล็อกไฟล์กรุ๊ปที่มีรูปแบบเป็น syslog สแตนด์ดาร์ดจะสามารถมาเรียกใช้ได้ โดยสามารถกำหนดการเรียกใช้ไว้ใน /etc/log.d/conf/logfile/*.conf หรือใน /etc/log.d/conf/service/*.conf ดังที่ได้อธิบายไปแล้วข้างต้น

โดยในโครงงานนี้จะมีอยู่ 5 โปรแกรมให้เรียกใช้โดยมีหน้าที่แตกต่างกันไปดังนี้

- *applystddate* มีหน้าที่ในการเลือกบรรทัดจากล็อกไฟล์กรุ๊ปที่เรียกใช้โดยเลือกเอาเฉพาะวันที่มีค่าตรงกับค่าของ Range ที่กำหนดไว้ โดยอาจเป็นได้ทั้ง yesterday,all,today
- *expandrepeats* เนื่องจากในล็อกไฟล์กรุ๊ปบางไฟล์จะมีการเก็บบันทึกข้อความแบบไม่ซ้ำ และถ้าบรรทัดใดที่ติดกันมีการซ้ำกันเกิดขึ้นจะบันทึกเป็น “last messages repeated X times” แล้วเมื่อผ่านโปรแกรมนี้แล้ว บรรทัดที่มีการเก็บแบบนี้จะถูกเปลี่ยนเป็นข้อความที่มีการซ้ำกันตามจำนวนครั้ง X ที่เกิดขึ้น
- *onlyservice* เนื่องจากในล็อกไฟล์กรุ๊ปบางไฟล์จะมีการเก็บความผิดพลาดที่เกิดขึ้นจากระบบจากหลายๆ เซอร์วิส เพราะฉะนั้นเมื่อล็อกไฟล์กรุ๊ปผ่านโปรแกรมนี้แล้วก็จะได้แต่เซอร์วิสที่มีการส่งเป็นอาร์กิวเมนท์มา
- *removeheaders* จะทำหน้าที่ในการตัดส่วนที่ไม่เกี่ยวข้องในการพิจารณาจากเซอร์วิสออก
- *removeservice* จะทำหน้าที่ในการตัดเซอร์วิสที่ไม่เกี่ยวข้องในโครงงานนี้ออก โดยสามารถส่งค่าที่ต้องการตัดออกเป็นอาร์กิวเมนท์มาได้ ซึ่งมีได้หลายค่า

- **/service**

เป็นสับไดเรกทอรี่ที่เป็นที่เก็บของสคริปต์ของเซอร์วิสที่มีให้เรียกใช้ โดยในโครงงานนี้จะมีให้เรียกใช้อยู่ 9 เซอร์วิส โดยรายละเอียดของเซอร์วิสสามารถเปลี่ยนแปลงและกำหนดเพิ่มเติมได้ที่ไฟล์ใน /etc/log.d/conf/service/*.conf

1) **cron**

ด้วยเซอร์วิสนี้จะมีการตรวจสอบล็อกไฟล์กรุ๊ป ชื่อ cron โดยทำการตรวจสอบและแสดงผลในเรื่องต่อไปนี้

- มีการเช็คว่ามีผู้ใช้คนใดทำการรัน (run) cron ใดบ้างกี่ครั้ง
- มีการเช็คว่ามีผู้ใช้คนใดทำ crontab reloaded บ้างกี่ครั้ง
- มีการเช็คการสตาร์ทอัพของ cron ว่ามีกี่ครั้ง
- มีการเช็คการรีโหลด /etc/crontab ว่ามีกี่ครั้ง
- และในส่วนสุดท้ายจะเป็นส่วนที่จะแสดงทุกบรรทัดที่ไม่ต้องตามเงื่อนไขข้างต้น

การตรวจสอบเซอร์วิส cron จะทำการแสดงผลให้ก็ต่อเมื่อมีการกำหนดค่าของ detail ให้มากหรือเท่ากับ med

2) ftpd-message

ด้วยเซอร์วิสนี้จะมีการตรวจสอบล็อกไฟล์กรุ๊ปชื่อ messages โดยจะทำการตรวจสอบและแสดงผลในเรื่องต่อไปนี้

- จะมีการเช็คว่ามีการทำ FTP จากโฮสต์ไหนด้วย user อะไรเป็นจำนวนกี่ครั้ง
- จะมีการเช็คว่ามีการปฏิเสธเข้าพอร์ตจากโฮสต์ไหนกี่ครั้ง

⇒ ในกรณีที่มีการกำหนดให้มี detail มากกว่าหรือเท่ากับ med จะแสดงผลในเรื่องต่อไปนี้

- จะมีการเช็คว่ามีการทำ FTP login ด้วย Anonymous จากที่ไหนเป็นจำนวนกี่ครั้ง

⇒ ในกรณีที่มีการกำหนดให้มี detail มากกว่าหรือเท่ากับ high จะแสดงผลในเรื่องต่อไปนี้

- จะมีการเช็คการลบไฟล์ผ่านทาง FTP ว่าจากโฮสต์ไหน ผู้ใช้คนใด ลบไฟล์ใด

⇒ ในกรณีที่ค่าของตัวแปร $ftpd_ignore_unmatched เป็น 0 จะมีการแสดงผลเรื่องนี้ด้วย

- จะแสดงผลของบรรทัดที่ไม่เข้ากับเงื่อนไขข้างต้นและที่ไม่ถูกหักออกในตัวสคริปต์จริง

3) ftpd-xferlog

ด้วยเซอร์วิสนี้จะมีการตรวจสอบล็อกไฟล์กรุ๊ปชื่อ xferlog โดยจะทำการตรวจสอบและแสดงผลในเรื่องต่อไปนี้

- จะมีการแสดงค่าของจำนวนไฟล์ที่มีการส่งออกในหน่วย KBytes และ MBytes
- จะมีการแสดงค่าของจำนวนไฟล์ที่มีการส่งเข้าในหน่วย KBytes และ MBytes
- จะมีการเช็คการส่งไฟล์เข้าจาก Anonymous ว่ามาจากโฮสต์ไหนส่งไฟล์อะไรมาและถ้ามี detail มากกว่าหรือเท่ากับ 15 แล้วจะบอกอีเมล์ของผู้ที่ส่งเข้ามาด้วย
- จะแสดงผลของบรรทัดที่ไม่เข้ากับเงื่อนไขข้างต้นออกมา

⇒ ในกรณีที่มีการกำหนดให้มี detail มากกว่าหรือเท่ากับ med จะแสดงผลในเรื่องต่อไปนี้

- จะมีการเช็คการส่งไฟล์ออกจาก Anonymous ว่าส่งไปยังโฮสต์ไหนส่งไฟล์อะไรไปและถ้ามี detail มากกว่าหรือเท่ากับ 15 แล้วจะบอกอีเมล์ของผู้ที่ส่งออกไปด้วย

⇒ ในกรณีที่มีการกำหนดให้มี detail มากกว่าหรือเท่ากับ high จะแสดงผลในเรื่องต่อไปนี้

- จะมีการเช็คการรับไฟล์เข้ามาในโฮสต์ว่ามาจากโฮสต์ใด ชื่อไฟล์ และผู้ใช้ในระบบที่ทำ
- จะมีการเช็คการส่งไฟล์ออกจากโฮสต์ว่าไปยังโฮสต์ใด ชื่อไฟล์ และผู้ใช้ในระบบที่ทำ

4) init

ด้วย เซอร์วิส นี้จะมีการตรวจสอบล็อกไฟล์กรุ๊ปชื่อ messages โดยจะทำการตรวจสอบและแสดงผลในเรื่องต่อไปนี้

- มีการเช็คการ เปลี่ยนเลเวล (Level) ของระบบว่ามีการเปลี่ยนไปเป็นเลเวลใดบ้างกี่ครั้ง
- จะแสดงผลของบรรทัดที่ไม่เข้ากับเงื่อนไขข้างต้นออกมา

5) **kernel**

ด้วยเซอร์วิสนี้จะมีการตรวจสอบล็อกไฟล์กรุ๊ปชื่อ messages โดยจะทำการตรวจสอบและแสดงผลในเรื่องต่อไปนี้

- จะมีการเช็คความน่าจะเป็นที่จะเกิดการทำ SYN Flooding โดยจะบอกว่ามาจากไหน
- จะมีการเช็คเหตุการณ์ต่างๆที่เกี่ยวข้องกับเคอร์เนลว่ามีเหตุการณ์นั้นเกิดขึ้นกี่ครั้ง

6) **modprobe**

ด้วยเซอร์วิสนี้จะมีการตรวจสอบล็อกไฟล์กรุ๊ปชื่อ messages โดยจะทำการตรวจสอบและแสดงผลในเรื่องต่อไปนี้

- จะแสดงผลของบรรทัดที่ไม่เข้ากับเงื่อนไขที่กำหนดในสคริปต์ออกมา

⇒ ในกรณีที่มีการกำหนดให้มี detail มากกว่าหรือเท่ากับ high จะแสดงผลในเรื่องต่อไปนี้

- จะแสดงผลของความผิดพลาดในการเรื่องที่ตั้งโมดูล (Locate module) ว่าเกิดขึ้นกับโมดูลไหนกี่ครั้ง

7) **pam_pwdb**

ด้วยเซอร์วิสนี้จะมีการตรวจสอบล็อกไฟล์กรุ๊ปชื่อ messages โดยจะทำการตรวจสอบและแสดงผลในเรื่องต่อไปนี้

- จะมีการแสดงรายชื่อของผู้ใช้ในระบบที่มีการเรียกใช้คำสั่ง su โดยจะบอกถึงว่าเป็นผู้ใช้คนใด ทำคาร su เป็นใครมีหมายเลขไอดี(user ID)เป็นอะไรในระบบ กี่ครั้ง ในส่วนนี้หากผู้ดูแลระบบ ต้องการให้มีการตรวจสอบที่ละเอียดขึ้นก็สามารถทำได้ โดยการกำหนดค่าให้กับไฟล์ /etc/log.d/conf/su_file ซึ่งภายในไฟล์ ผู้ดูแลระบบสามารถกำหนดรายชื่อของผู้ใช้ที่ผู้ดูแลระบบไว้ใจและให้สิทธิ์ในการ su เป็น root ได้ โดยใส่รายชื่อเป็นบรรทัดๆ หากไฟล์นี้มีรายชื่ออยู่แล้ว ในการประมวลผลของเซอร์วิส pam_pwdb นี้จะนำไฟล์นี้มาพิจารณาด้วย โดยหากผู้ใช้คนใดที่ผู้ดูแลระบบไม่ได้ให้ความไว้วางใจด้วยการใส่ชื่อลงในไฟล์ su_file แล้ว ถ้าทำการ su เป็น root จะมีการฟ้องขึ้นมาในการแสดงผลว่ามีผู้ใช้ที่ผู้ดูแลระบบไม่รู้จักได้ทำการ su เป็น root เป็นจำนวนกี่ครั้ง ณ เวลาใดบ้าง เพื่อให้ผู้ดูแลระบบได้นำไปพิจารณาและตัดสินใจต่อผู้ใช้คนนี้อีกครั้ง
- จะมีการแสดงผล หากว่ามีการเปลี่ยนพาสเวิร์ดของผู้ใช้ในระบบ
- จะมีการแสดงผลการหมดอายุของพาสเวิร์ดของผู้ใช้ในระบบ
- จะมีการเช็คการเปิดเซสชั่น (session) จากผู้ใช้ในระบบว่ามีการเปิดเซสชั่นใดบ้างด้วยผู้ใช้คนใดเป็นจำนวนกี่ครั้ง
- จะมีการเช็คการทำ Remote login ว่ามาจากโฮสต์ไหนด้วยชื่อผู้ใช้คนใด
- จะมีการเช็คการทำการล็อกอินจาก local host ว่าจากผู้ใช้คนใดเป็นจำนวนเท่าไร
- จะมีการเช็คจำนวนครั้งการล็อกอินของ root ว่าจาก tty ใด กี่ครั้ง

- จะแสดงผลของบรรทัดที่ไม่เข้ากับเงื่อนไขที่กำหนดในสคริปต์ออกมา

⇒ ในกรณีที่มีการกำหนดให้มี detail มากกว่าหรือเท่ากับ med จะแสดงผลในเรื่องต่อไปนี้

- จะมีการแสดงผลว่าการทำ authentication failure จากแต่ละผู้ใช้ ว่าทำอะไรบ้าง
- จะมีการแสดงผลการล็อกอินไม่สำเร็จจากแต่ละโฮสต์

⇒ ในกรณีที่มีการกำหนดให้มี detail มากกว่าหรือเท่ากับ high จะแสดงผลในเรื่องต่อไปนี้

- หากมีกรณีที่เป็นเรื่องเกี่ยวกับการพิสูจน์พาสเวิร์ด (authentication could not identify password)

8) secure

ด้วยเซอร์วิสนี้จะมีการตรวจสอบล็อกไฟล์กรุ๊ปชื่อ secure โดยจะทำการตรวจสอบและแสดงผลในเรื่องต่อไปนี้

- จะมีการแสดงผลของการติดต่อที่เกิดขึ้นโดยเรียงตามเซอร์วิสว่ามาจากโฮสต์ไหนกี่ครั้ง ในกรณีนี้หากผู้ดูแลระบบต้องการที่จะให้มีความละเอียดเพิ่มขึ้นก็สามารถทำได้ด้วยการกำหนดค่าให้กับไฟล์ /etc/log.d/conf/trust_file
- จะมีการเช็คความผิดพลาดของการติดต่อ (Refused Connections) ว่าจากเซอร์วิสไหนบ้าง จากที่ไหน
- จะมีการเช็คความผิดพลาดของการล็อกอิน (Failed Login) โดยเรียงตามชื่อผู้ใช้บอกโฮสต์ที่มาและจำนวนครั้งที่พลาด
- จะมีการแสดงผลเมื่อมีความผิดพลาดในการสร้างการเชื่อมต่อให้กับเครื่องไคลเอนท์ที่ไม่สามารถระบุไอพี (IP) ได้
- จะแสดงผลของความผิดพลาดต่างๆที่ล็อกไฟล์ secure ได้ทำการบันทึกไว้
- จะแสดงผลของบรรทัดที่ไม่เข้ากับเงื่อนไขที่กำหนดในสคริปต์ออกมา

9) syslogd

ด้วยเซอร์วิสนี้จะมีการตรวจสอบล็อกไฟล์กรุ๊ปชื่อ messages โดยจะทำการตรวจสอบและแสดงผลในเรื่องต่อไปนี้

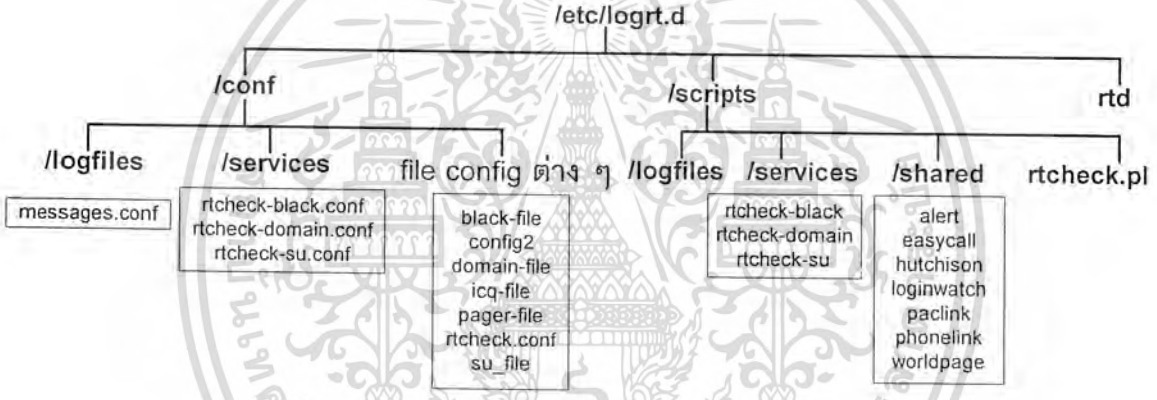
- จะมีการเช็คว่า Syslogd มีการสตาร์ทอัพขึ้นมากี่ครั้ง
- เช็คประโยคในล็อกไฟล์ ว่า "Could not glue messages parts together " ด้วย error อะไร กี่ครั้ง
- จะแสดงผลของบรรทัดที่ไม่เข้ากับเงื่อนไขที่กำหนดในสคริปต์ออกมา

3.4 ส่วนวิเคราะห์และแจ้งเตือนแบบเรียลไทม์ (rtcheck)

ในส่วนเป็นส่วนที่มีความสำคัญมาก เพราะว่าการตรวจเช็คแบบเรียลไทม์ (Real time) จะมีประสิทธิภาพมากเท่าไรก็ขึ้นอยู่ว่าการกำหนดเซอร์วิสที่เป็นตัวเช็คและรายงานผลในส่วนนี้ ซึ่งมีความยืดหยุ่นเหมือนกับการเขียนเซอร์วิสสคริปต์ในส่วนวิเคราะห์และสร้างรายงานประจำวัน ทั้งยังมีโครงสร้างการคอนฟิกกูเรชั่นเป็นรูปแบบเดียวกันกับส่วนวิเคราะห์และสร้างรายงานประจำวัน ทำให้ง่ายต่อการเรียนรู้เพื่อใช้งาน

ในส่วนนี้มีความเหมาะสมสำหรับใช้เป็นอุปกรณ์ตรวจสอบระบบในกรณีที่เหตุการณ์ที่ต้องการตรวจจับนั้นมีความเสี่ยงสูง และมีผลกระทบรุนแรงไม่สามารถรอคอยเวลาที่จะรายงานผลประจำวันได้

ซึ่งในโครงงานนี้เซอร์วิสที่ผู้จัดทำได้กำหนดเป็นต้นแบบของการเขียนเซอร์วิสสคริปต์และการคอนฟิกกูเรชั่น มีดังนี้



รูปที่ 3.4 แสดงไดเรกทอรี่ของส่วนวิเคราะห์และแจ้งเตือนแบบเรียลไทม์โดยละเอียด

1. การเช็คการล็อกอินจากผู้ใช้คนเดียวกันจากหลายโฮสต์ (rtcheck-domain)

<u>แนวความคิด</u>

จากแนวความคิดที่ว่าผู้ใช้ควรจะใช้เครื่องคอมพิวเตอร์ในขณะเวลาหนึ่งๆแค่หนึ่งเครื่องเท่านั้น ถ้าหากปรากฏว่าผู้ใช้คนเดียวกันนี้มีการล็อกอินเข้ามาใช้ระบบพร้อมกันจาก 2 สถานที่ที่อยู่ห่างกัน อาจเป็นไปได้ว่ามีผู้อื่นลักลอบใช้งานแอคเค้าท์ (account) ของผู้ใช้คนนี้อยู่

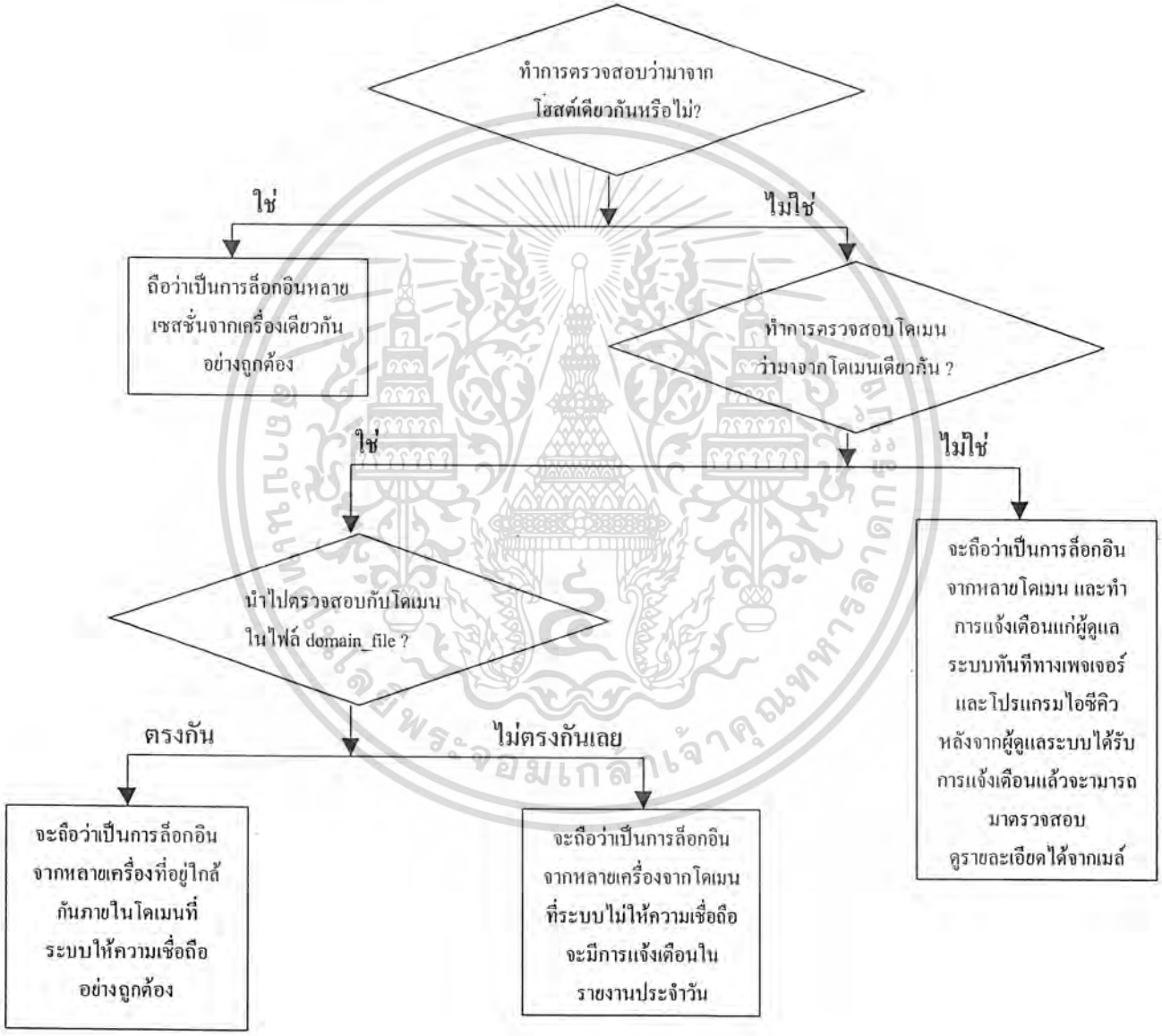
<u>การทำงาน</u>

เมื่อเซอร์วิสนี้ถูกเรียกใช้งานจาก rt-check แล้ว จะเริ่มต้นการทำงานด้วยการไปอ่านเมสเซสที่เพิ่งจะเกิดขึ้นและถูกเก็บไว้ใน /tmp/messages แล้วนำมาตรวจสอบโดยใช้หลักการดังนี้

- ถ้าผู้ใช้ล็อกอินหลายคอนเนคชั่น (connections) มาจากโฮสต์เดียวกัน ถือว่าถูกต้อง

- แต่ถ้ามาจากโฮสต์ที่ต่างกันจะทำการตรวจสอบโดเมน ว่ามาจากโดเมนเดียวกันหรือไม่ ถ้าหากเป็นโดเมนเดียวกัน จะถือว่าผู้ใช้อาจจะล็อกอินมาจากเครื่องที่อยู่ข้างเคียงกัน แต่ก็จะมีการส่งเมล์บอกผู้ดูแลระบบถึงเหตุการณ์นี้ แต่ถ้ามีการล็อกอินมาจากคนละโดเมน จะถือว่ามีผู้ลักลอบใช้งานแอคเค้าท์ของผู้ใช้คนนี้แล้ว และจะมีการแจ้งเตือนต่อผู้ดูแลระบบทันทีทางไอซีคิว (icq) และผ่านระบบเพจเจอร์

ทำงานดังอัลกอริธึมนี้



รูปที่ 3.5 แสดงอัลกอริธึมการทำงานการล็อกอินจากผู้ใช้คนเดียวกันจากหลายโฮสต์ (rtcheck-domain)

การคอนฟิกกูเรชั่น

เซอร์วิสนี้จะมีไฟล์คอนฟิกอยู่ที่ /etc/logrt.d/conf/service/rtcheck-domain.conf และอีกไฟล์ที่ /etc/logrt.d/conf/domain_file โดยไฟล์ rtcheck-domain.conf จะเป็นไฟล์ที่กำหนดว่าเซอร์วิสนี้จะต้องทำการอ่านล็อกไฟล์ชื่ออะไร โดยมีรูปแบบเป็น

Logfile = messages

แสดงว่าเซอร์วิสนี้จะใช้ล็อกไฟล์ชื่อ messages ที่ส่วนวิเคราะห์และแจ้งเตือนแบบเรียลไทม์รู้จัก (ซึ่งจะอยู่ที่ใดก็ขึ้นอยู่กับไฟล์ /etc/logrt.d/conf/logfiles/messages.conf)

ส่วนไฟล์ domain_file เป็นไฟล์ที่กำหนดโดเมนที่เราให้ความเชื่อถือไว้วางใจ ยกตัวอย่าง เช่น

cc.kmitl.ac.th

kmitl.ac.th

tu.ac.th

แสดงว่าระบบของเราให้ความเชื่อถือกับเครื่องที่มาจากโดเมนทั้ง 3 นี้ หากมีการขอใช้บริการจากโดเมนอื่นที่ระบบเราไม่ให้ความเชื่อถือระบบจะทำการแจ้งเตือนทันที

2. การเช็กการใช้คำสั่ง su เปลี่ยนสิทธิของตัวเองเป็น root ของผู้ใช้ที่ไม่ได้มีสิทธิ (rtcheck-su)

แนวความคิด

จากแนวความคิดที่ว่าผู้ใช้ที่จะใช้คำสั่ง su นี้ได้จะต้องเป็นเจ้าหน้าที่ที่ทำหน้าที่ดูแลระบบเท่านั้น ถ้าหาผู้ใช้งานระบบคนอื่นสามารถใช้คำสั่งนี้เปลี่ยนสิทธิของตนได้สำเร็จ ก็หมายความว่าผู้ใช้คนนั้นทราบรหัสผ่านของ root

การทำงาน

เมื่อเซอร์วิสนี้ถูกเรียกใช้งานแล้วจะเริ่มต้นการทำงานด้วยการไปอ่านเมสเซสที่เพิ่งจะเกิดขึ้นและถูกเก็บไว้ใน /tmp/messages แล้วนำมาตรวจสอบโดยใช้หลักการดังนี้

- ตรวจสอบผู้ที่รันโปรแกรม su
- แล้วนำชื่อนั้นไปเปรียบเทียบกับรายชื่อเจ้าหน้าที่ที่อยู่ในไฟล์ /etc/logrt.d/conf/su_file
- ถ้าหากมีผู้ที่ไม่ได้อยู่ในรายชื่อก็แสดงว่าเป็นผู้บุกรุกระบบ แล้วจะแจ้งไปยังผู้ดูแลระบบผ่านทางเพจเจอร์ทันทีและทางไอซีคิว

การคอนฟิกกูเรชั่น

เซอร์วิสนี้จะมีไฟล์คอนฟิกอยู่ที่ /etc/logrt.d/conf/service/rtcheck-su.conf และอีกไฟล์ที่ /etc/logrt.d/conf/su_file โดยไฟล์ rtcheck-su.conf จะเป็นไฟล์ที่กำหนดว่าเซอร์วิสนี้จะต้องทำการอ่านล็อกไฟล์ชื่ออะไร โดยมีรูปแบบเป็น

Logfile = messages

แสดงว่าเซอร์วิสนี้จะใช้ล็อกไฟล์ชื่อ messages ที่ส่วนวิเคราะห์และแจ้งเตือนแบบเรียลไทม์รู้จัก (ซึ่งจะอยู่ที่ใดก็ขึ้นอยู่กับไฟล์ /etc/logrt.d/conf/logfiles/messages.conf)

ส่วนไฟล์ su_file เป็นไฟล์ที่กำหนดผู้ใช้ที่เราให้ความเชื่อถือไว้วางใจในการใช้คำสั่ง su ในการเปลี่ยนสิทธิเป็น root ยกตัวอย่าง เช่น

tee

korn

แสดงว่าระบบของเราให้ความเชื่อถือกับผู้ใช้ทั้ง 2 คนนี้ให้มีสิทธิในการใช้คำสั่ง su เป็น root และหากมีผู้ใดใช้คำสั่ง su เป็น root โดยที่เป็นบุคคลที่ระบบไม่ให้สิทธิ ระบบก็จะทำการแจ้งเตือนทันที

3. การเช็กรายชื่อผู้ใช้ที่ล็อกอินและเป็นผู้ใช้ที่ต้องจับตาดูเป็นพิเศษ (rtcheck-black)

แนวความคิด

จากรายงานประจำวันต่างๆ อาจจะมีผู้ใช้บางคนที่มีพฤติกรรมเข้าข่ายว่ากำลังพยายามแฮก (hack) ระบบอยู่และควรจะได้รับการจับตาดูเป็นพิเศษจากผู้ดูแลระบบ เมื่อผู้ใช้ดังกล่าวทำการล็อกอินเข้าสู่ระบบ

การทำงาน

เมื่อเซอร์วิสนี้ถูกเรียกใช้งานแล้วจะเริ่มต้นการทำงานด้วยการไปอ่านเมสเซสที่เพิ่งจะเกิดขึ้นและถูกเก็บไว้ใน /tmp/messages แล้วนำมาตรวจสอบโดยใช้หลักการดังนี้

- ตรวจสอบไฟล์ /etc/logrt.d/conf/black_list
- เปรียบเทียบผู้ใช้ที่ล็อกอินเข้ามากับรายชื่อจากไฟล์
- ถ้าหากเป็นผู้ที่มีรายชื่ออยู่ในไฟล์นี้ ก็จะแจ้งเตือนแก่ผู้ดูแลระบบด้วยเพจเจอร์และทางไอซีคิวทันที

การคอนฟิกกูเรชั่น

เซอร์วิสนี้จะมีไฟล์คอนฟิกอยู่ที่ /etc/logrt.d/conf/service/rtcheck-black.conf และอีกไฟล์ที่ /etc/logrt.d/conf/black_file โดยไฟล์ rtcheck-black.conf จะเป็นไฟล์ที่กำหนดว่าเซอร์วิสนี้จะต้องทำการอ่านล็อกไฟล์ชื่ออะไร โดยมีรูปแบบเป็น

Logfile = messages

แสดงว่าเซอร์วิสนี้จะใช้ล็อกไฟล์ชื่อ messages ที่ส่วนวิเคราะห์และแจ้งเตือนแบบเรียลไทม์รู้จัก (ซึ่งจะอยู่ที่ใดก็ขึ้นอยู่กับไฟล์ /etc/logrt.d/conf/logfiles/messages.conf)

ส่วนไฟล์ black_file เป็นไฟล์ที่กำหนดผู้ใช้ที่เราต้องจับตาดูเป็นพิเศษเมื่อมีการล็อกอินเข้ามาในระบบ ยกตัวอย่าง เช่น

nueng

kik

แสดงว่าระบบของเราจะแจ้งเตือนทันทีเมื่อมีผู้ใช้หนึ่งในสองคนนี้ทำการล็อกอินเข้ามาในระบบ เพื่อให้ผู้ดูแลระบบมาสังเกตุการณ์การใช้ระบบของผู้ใช้ที่มีรายชื่ออยู่ใน black_file ก่อนที่จะทำความเสียหายให้แก่ระบบ

3.5 ส่วนเดมอน (rtd : Realtime Daemon)

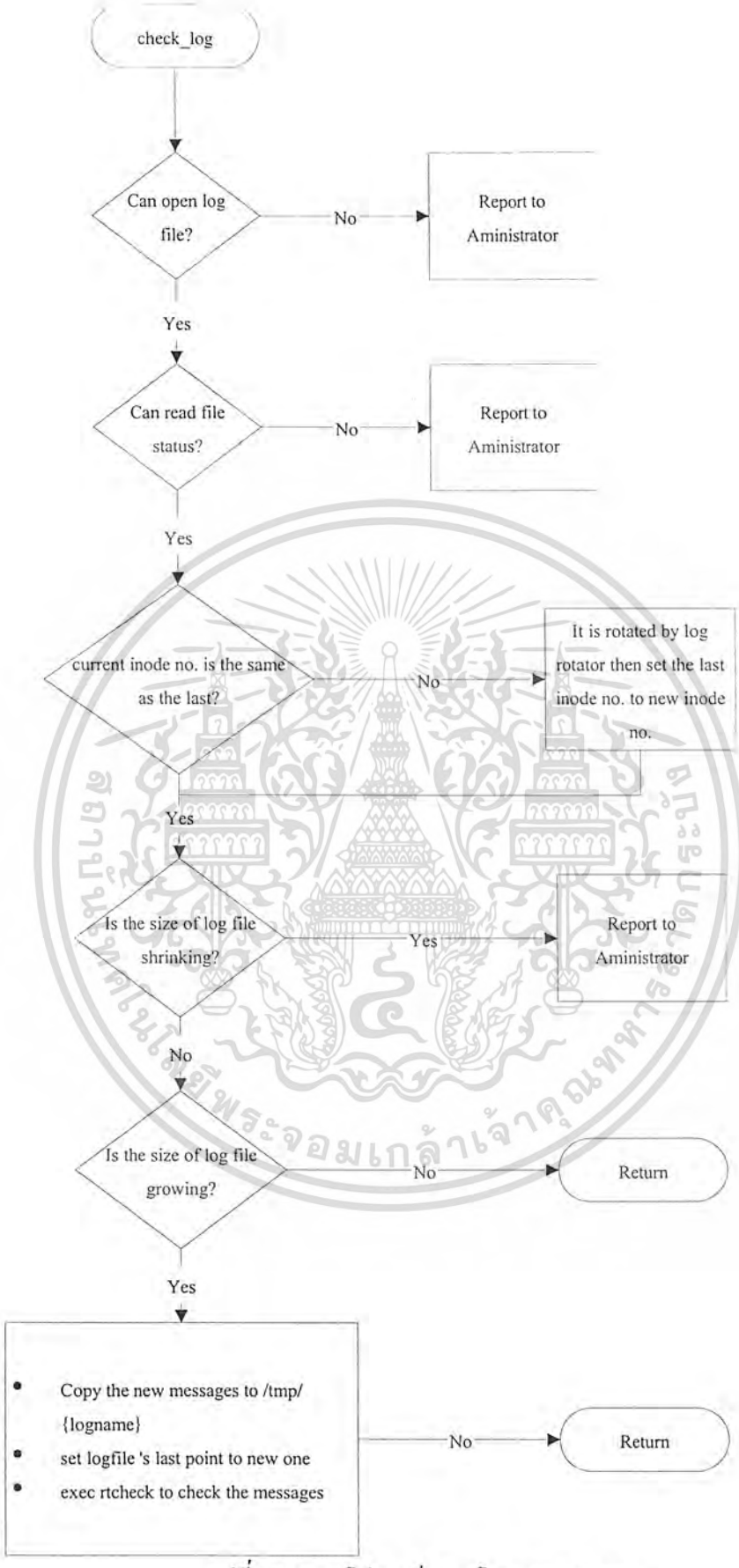
ในส่วนของระบบแบบเรียลไทม์นั้นจะมีเดมอนคอยตรวจสอบระบบของล็อกไฟล์ที่จำเป็นสำหรับเซอร์วิส ซึ่งถ้าหากมีเมสเซสเกิดขึ้นใหม่ตัวเดมอนนี้จะตัดเฉพาะส่วนที่เกิดขึ้นใหม่เก็บไว้ในสับไดเรกทอรี่ /tmp ในชื่อเดียวกับล็อกไฟล์นั้น เช่น ถ้าเป็นล็อกไฟล์ messages ก็จะเก็บอยู่ใน /tmp/messages แล้วเรียกตัว rtcheck.pl ขึ้นมาทำงานโดยระบุว่าเซอร์วิสที่เกี่ยวกับล็อกไฟล์กรุ๊ปใดว่าต้องทำงานบ้าง โดยเมื่อตัวส่วนวิเคราะห์และแจ้งเตือนแบบเรียลไทม์ (rt-check) ทำการวิเคราะห์อยู่นั้น ส่วนนี้จะหยุดคอยจนกระทั่งอีกส่วนทำงานจนเสร็จ แล้วจึงลบไฟล์ชั่วคราวที่อยู่ที่ /tmp ออกแล้วกลับไปคอยตรวจสอบล็อกไฟล์ที่กำหนดต่อไป

หลักการทำงาน

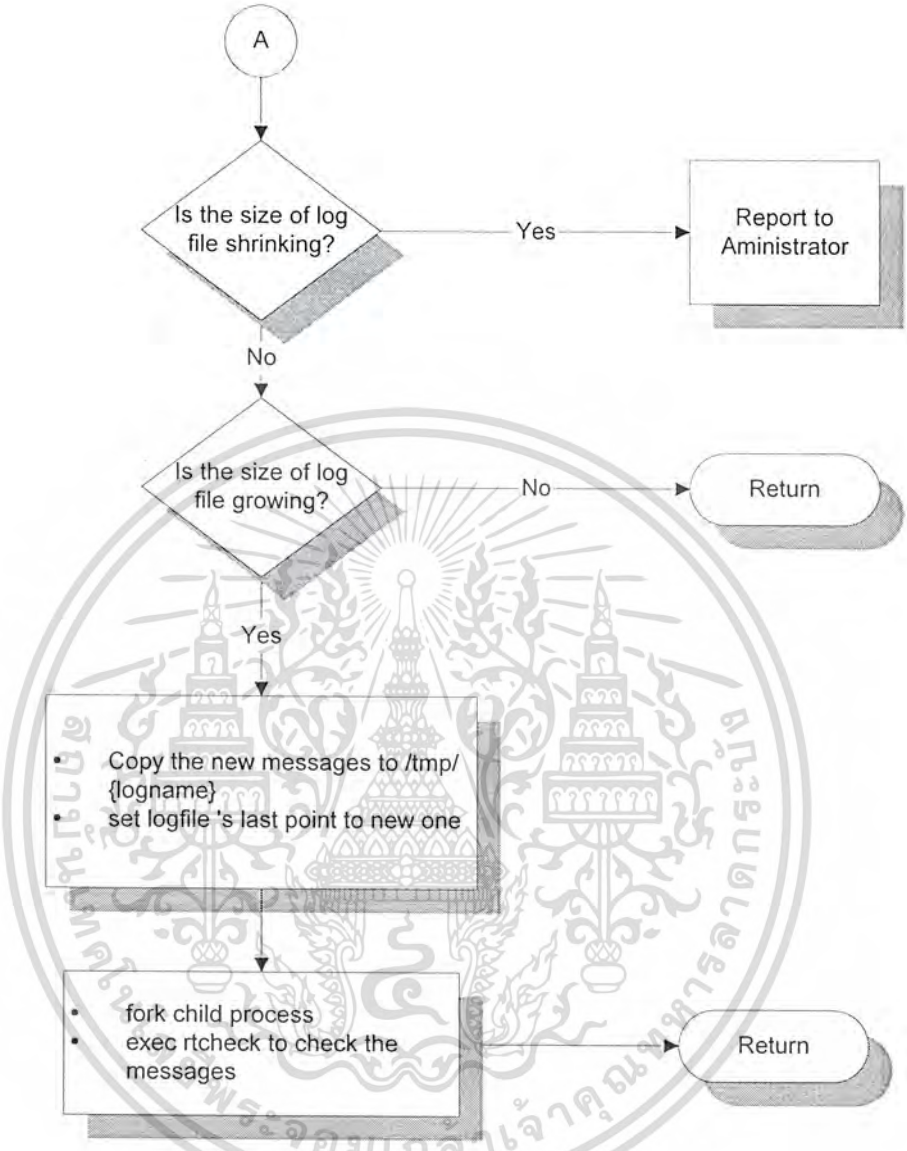
- เมื่อเริ่มต้นทำงานจะอ่านค่าในไฟล์ rtd.conf ว่ามีล็อกไฟล์ใดบ้างที่จำเป็นต้องตรวจสอบ แล้วเก็บชื่อไฟล์เหล่านั้นไว้
- แล้วเรียกโมดูลที่ใช้ตรวจสอบความเปลี่ยนแปลงของล็อกไฟล์ที่มีชื่อว่า check_log ทำงานเป็นแบ็กกราวน์และถ้าหากพบว่าไฟล์มีส่วนเพิ่มเติมขึ้นก็จะคัดลอกส่วนที่เพิ่มขึ้นนั้นเก็บไว้ใน /tmp/ ชื่อล็อกไฟล์เพื่อให้เซอร์วิสนำไปวิเคราะห์ต่อไป
- ส่วนหลักจะคอย(wait) โปรเซสลูกที่ทำการตรวจสอบล็อกไฟล์จนทำงานเสร็จก็จะหลับ (sleep) เป็นช่วงเวลาหนึ่งที่ตั้งไว้แล้วจึงกลับไปตรวจสอบล็อกไฟล์อีกครั้งหนึ่งเพื่อไม่ให้ตัวเดมอนกินซีพียูไทม์มากจนเกินไป โดยจะต้องทำงานได้ประสิทธิภาพในระดับที่ยอมรับได้

หลักการทำงานของโมดูล check_log

- โมดูลจะทดลองเปิดล็อกไฟล์ดูก่อนแล้วอ่านค่าสถานะของไฟล์ ถ้าไม่สามารถอ่านได้ก็แสดงว่ามีปัญหากับล็อกไฟล์โปรแกรมจะหยุดทำงานแล้วแจ้งเตือนผู้ดูแลระบบทันที
- แล้วโมดูลก็จะตรวจสอบหมายเลขไอโหนด(inode number) ถ้าหากมีการเปลี่ยนแปลงไปจะการตรวจสอบครั้งก่อนก็แสดงว่าล็อกไฟล์ถูกสร้างขึ้นใหม่จากระบบล็อกโรเทท (log rotate) ก็จะกำหนดตำแหน่งท้าย(ขนาด) ของล็อกไฟล์ใหม่ให้เป็น 0
- ตรวจสอบขนาดของล็อกไฟล์โดยถ้าหากมีขนาดลดลงก็แสดงว่าเกิดปัญหาขึ้นกับล็อกไฟล์ แล้วเนื่องจากล็อกไฟล์ไม่ควรจะมีขนาดลดลงไม่ว่าในกรณีใดๆ ทั้งสิ้นอาจเป็นไปได้ว่ามีผู้ไม่ประสงค์ดีทำการลบข้อมูลบางส่วนในล็อกไฟล์ออกไป ระบบจะทำการแจ้งเตือนผู้ดูแลระบบทันที
- และถ้าล็อกไฟล์มีขนาดโตขึ้นก็แสดงว่ามีเมสเซสเกิดขึ้นใหม่ ก็จะตัดเฉพาะส่วนที่เกิดขึ้นใหม่เก็บใน /tmp เก็บตำแหน่งท้ายของไฟล์ใหม่ แล้วเรียกคำสั่ง exec โปรแกรม rtcheck ให้ทำงานโดยระบุว่ามีการเปลี่ยนแปลงขึ้นกับล็อกไฟล์นั้นๆ



รูปที่ 3.6 แสดงโฟลชาร์ทของโมดูล check_log



รูปที่ 3.7 แสดงโฟล์ชาร์ทของส่วนหลักของส่วนเดมอน

3.6 ส่วนการแจ้งเตือนผ่านทางเพจเจอร์

เพื่อให้ระบบสามารถรายงานแก่ผู้ดูแลระบบได้ในทันทีไม่ว่าผู้ดูแลระบบจะอยู่ที่ใด ทางผู้จัดทำโครงงานนี้ได้เลือกวิธีการส่งเพจเจอร์ผ่านอินเตอร์เนต เนื่องจากว่าระบบที่ผู้จัดทำใช้ในการทดสอบเป็นระบบที่มีการเชื่อมต่ออินเตอร์เนตตลอดเวลาและเหตุที่ไม่ใช้วิธีส่งผ่านโมเดมเพราะการส่งผ่านอินเตอร์เนตไม่เสียค่าใช้จ่าย โดยการส่งเพจเจอร์ผ่านทางอินเตอร์เนตนี้เราจะต้องทำการติดต่อกับเวบเซิร์ฟเวอร์โดยผ่านโปรโตคอล HTTP

เพื่อให้เพิร์ลสคริปต์สามารถติดต่อกับระบบ HTTP ได้ เราจำเป็นจะต้องทำการติดตั้งไลบรารี่เพิ่มเติมให้กับเพิร์ลที่ชื่อว่า "HTTP Request" ซึ่งก่อนที่จะติดตั้งไลบรารี่นี้เราจะต้องติดตั้งไลบรารี่ที่จำเป็นดังนี้

1. URI
2. HTML :: HeadParser
3. MIME :: Base64
4. Data :: Dumper
5. IO :: Socket
6. Net :: FTP
7. Digest :: MD5

โดยจะต้องทำการติดตั้งเรียงตามลำดับ แล้วจึงสามารถติดตั้งไลบรารี่ "HTTP Request" ได้ ซึ่งทุกไลบรารี่ที่กล่าวมาสามารถดาวน์โหลดได้ที่ ftp://ftp.nectec.or.th/pub/mirrors/CPAN/CPAN.html

<u>วิธีใช้งานไลบรารี่ HTTP Request</u>

```
use HTTP::Request::Common;
$ua = LWP::UserAgent->new;
$ua->proxy('http','http://proxy.foo.bar:8080/');
$ua->request(GET 'http://www.sn.no/');
$ua->request(POST 'http://somewhere/foo', [foo => bar, bar => foo]);
```

ในบรรทัดแรก จะเป็นการบอกถึงโมดูลที่ต้องนำมาใช้ในการรันโปรแกรมนี้

ในบรรทัดที่สอง เป็นการอินสแตนท์ซิเอต (Instantiate) ตัวแปรที่ชื่อว่า ua

ในบรรทัดที่สาม เป็นการกำหนดพร็อกซี่เซิร์ฟเวอร์

ในบรรทัดที่สี่และห้า เป็นตัวอย่างการทำ HTTP Request โดยในบรรทัดที่ 5 เป็นตัวอย่างการกำหนด key / value เพื่อส่งผ่านซีจีไอ (cgi) ของเวบเซิร์ฟเวอร์นั้นๆ

ดังนั้นการที่เราจะสามารถส่งเพจให้กับแต่ละยี่ห้อ เราก็ต้องทราบว่าเวบเซิร์ฟเวอร์ของยี่ห้อนั้นคือที่ไหน และซีจีไอไฟล์ (cgi file) ชื่ออะไร และจะต้องส่งคีย์ชื่ออะไรไป และแต่ละคีย์หมายถึงอะไร ซึ่งมีรายละเอียดดังนี้

ยี่ห้อ	รายละเอียด
	คีย์
1500 (Easycall)	http://203.146.1.214/cgi-bin/web2pager.cgi
	namefrom = ชื่อผู้ส่ง subject = หมายเลขเพจเจอร์ message = ข้อความ
152 (Phonelink)	http://www.phonelink.net/cgi-bin/psnxmas.pl
	Phone = หมายเลขเพจเจอร์ Mesg = ข้อความ encode = spg.net
142 (Worldpage)	http://www.worldpage.co.th/cgi-bin/Squeue.cgi
	page_no = หมายเลขเพจเจอร์ caller_pwd = รหัสลับของผู้ส่ง code_name = ชื่อเนมเพจ msgarea = ข้อความ day = วันที่จะส่ง(now) month = เดือนที่จะส่ง(now) year = ปีที่จะส่ง(1999) hours = ชั่วโมงที่จะส่ง(now) minutes = นาทีที่จะส่ง(now)
162 (Hutchison)	http://www.hutchison.co.th/html/cyberpage/cybermsg.asp
	pager_no = หมายเลขเพจเจอร์ accesspw = รหัสผ่านของเพจเจอร์ subject = ข้อความ from = ชื่อผู้ส่ง
1144 (Paclink)	http://www.paclink.co.th/phtml/send.phtml
	pager_number = หมายเลขเพจเจอร์ message = ข้อความ wday = วันที่จะส่ง(-1) hour = ชั่วโมงที่จะส่ง(00) minute = นาทีที่จะส่ง(00)

1188 (Posttel)	http://www.samart.co.th/cgi-bin/pager/sendpage.cgi iusername = ชื่อล็อกอินผู้ส่ง ipassword = รหัสผ่านของผู้ส่ง ipin = หมายเลขเพจเจอร์ iname = ชื่อผู้ส่ง imessage = ข้อความ month = เดือนที่จะส่ง (now = 1000) date = วันที่จะส่ง (now = 1000) hour = ชั่วโมงที่จะส่ง (now = 1000) minute = นาทีที่จะส่ง (now = 1000) notify = รายงานผลของการส่ง (NO = 1)

ตารางที่ 3.1 แสดงรายละเอียดของเพจเจอร์ยี่ห้อต่างๆ

ตัวอย่างการใช้งานที่ใช้ในโครงงาน

```
#!/usr/bin/perl

use HTTP::Request::Common;
use LWP::UserAgent;
$ua=new LWP::UserAgent;
$ua->proxy('http','http://161.246.10.21:3128/');
$ua->request(POST 'http://203.146.1.214/cgi-bin/web2pager.cgi',
    [ namefrom => $ARGV[0],
      subject => $ARGV[1],
      message => $ARGV[2],
    ]);
print("Ok...\n");
exit;
```

จากสคริปต์นี้ จะรับรายละเอียดผ่านทางคอมมานด์ไลน์ แล้วส่งข้อความไปยังเพจเจอร์ยี่ห้อ Easycall เช่น คุณ pong ต้องการส่งข้อความว่า "I love Pooh" ไปยังเพจเจอร์ยี่ห้อ Easycall หมายเลข 112233

```
$easycall pong 112233 "I love Pooh"
```

การคอนฟิกกูเรชั่นส่วนการแจ้งเตือนผ่านเพจเจอร์

การทำงานในส่วนนี้ผู้ใช้สามารถกำหนดได้เองว่าจะให้ส่งข้อความการแจ้งเตือนนั้นถึงเพจเจอร์เครื่องใดบ้างโดยสามารถกำหนดได้จากไฟล์ /etc/logrt.d/conf/pager_file ยกตัวอย่าง เช่น

easycall = 641000

easycall = 524622

phonelink = 319044

worldpage = 8021929

แสดงว่ามีระบบต้องทำการแจ้งเตือนผ่านทางเพจเจอร์จะทำการอ่านไฟล์นี้ หลังจากทำการประมวลผลแล้วจะส่งข้อความที่ต้องแจ้งเตือนนั้นให้กับผู้ดูแลระบบที่เป็นเจ้าของเพจเจอร์ยี่ห้อแล้วหมายเลขดังนี้ คือ อีซี่คอล หมายเลข 641000 และ 524622 , โฟนลิ้งค์ หมายเลข 319044 , เวิล์ดเพจ หมายเลข 8021929

3.7 ส่วนการแจ้งเตือนผ่านทางโปรแกรมไอซีคิว

นอกจากวิธีการส่งรายงานผ่านทางเพจเจอร์แล้ว ทางผู้จัดทำได้เลือกอีกวิธีการหนึ่งในการแจ้งรายงาน คือ การส่งรายงานผ่านทางไอซีคิว

โดยอาศัยบริการของบริษัท Mirabilis ซึ่งเปิดบริการที่ชื่อว่า "Web messages" ให้สามารถส่งเมล์ไปที่ {icq_no}@pager.mirabilis.com แล้วทางเซิร์ฟเวอร์จะส่งต่อข้อความไปยังผู้ใช้บริการไอซีคิวต่อไป

การคอนฟิกกูเรชั่นส่วนการแจ้งเตือนผ่านทางโปรแกรมไอซีคิว

การทำงานในส่วนนี้ผู้ใช้สามารถกำหนดได้เองว่าจะให้ส่งข้อความการแจ้งเตือนนั้นถึงผู้ดูแลระบบหมายเลขไอซีคิวใดบ้างโดยสามารถกำหนดได้จากไฟล์ /etc/logrt.d/conf/icq_file ยกตัวอย่าง เช่น

21800000

14707293

22000060

แสดงว่ามีระบบต้องทำการแจ้งเตือนผ่านทางโปรแกรมไอซีคิวจะทำการอ่านไฟล์นี้ หลังจากทำการประมวลผลแล้วจะส่งข้อความที่ต้องแจ้งเตือนนั้นให้กับผู้ดูแลระบบที่มีหมายเลขไอซีคิวดังนี้ คือ 21800000, 14707293, 22000060

บทที่ 4
ผลการทดลอง

หลังจากที่ผู้จัดทำได้ทำการสร้างระบบช่วยงานผู้ดูแลระบบขึ้นมาตามแนวคิดในบทที่ 3 แล้ว ทางผู้จัดทำได้ทำการทดลองการทำงานของส่วนต่างๆ โดยแบ่งเป็น 2 ส่วนหลัก คือ ส่วนวิเคราะห์และสร้างรายงานประจำวัน และ ส่วนวิเคราะห์และแจ้งเตือนแบบเรียลไทม์ โดยมีผลการทดลองดังต่อไปนี้

4.1 ผลการทดลองของส่วนวิเคราะห์และสร้างรายงานประจำวัน

เนื่องจากในส่วนนี้ผู้จัดทำได้ทำการสร้างเซอร์วิสในการวิเคราะห์และสร้างรายงานขึ้นมาทั้งหมด 9 เซอร์วิส จึงได้ทำการทดลองทีละเซอร์วิสแล้วดูผลของเมล์ที่ได้รับ โดยในที่นี่เป็นการสั่งให้ทำแต่ละเซอร์วิสจากเชลล์ไม่ได้อาศัยโปรแกรมครอนเดมอนในการทำงาน ทั้งนี้เพื่อความรวดเร็วในการเก็บผลและแก้ไขปรับปรุง

4.1.1 เซอร์วิส cron

เมื่อทำการสั่งคำสั่งนี้ที่เชลล์

$ dailycheck - - service cron

ระบบจะได้รับเมล์รายงานผลจากเซอร์วิสนี้ด้วยรูปแบบดังนี้

```
Date: Sun, 14 Mar 1999 18:36:45 +0700
From: root <root@IDS.localdomain>
To: root@IDS.localdomain
Subject: Daily Check for IDS.localdomain


#################### Daily Check Begin ########################

---------------------- Cron Begin ----------------------
Commands Run:
   User root:
      run-parts /etc/cron.daily: 1 Time(s)
      run-parts /etc/cron.hourly: 25 Time(s)
   CRON Restarted 2 Time(s)

---------------------- Cron End ------------------------


############################ Daily Check End ##############################

                          [END of message]
? Help      < MsgIndex  P PrevMsg    - PrevPage D Delete    R Reply
O OTHER CMDS > ViewAttch N NextMsg   Spc NextPage U Undelete F Forward
```

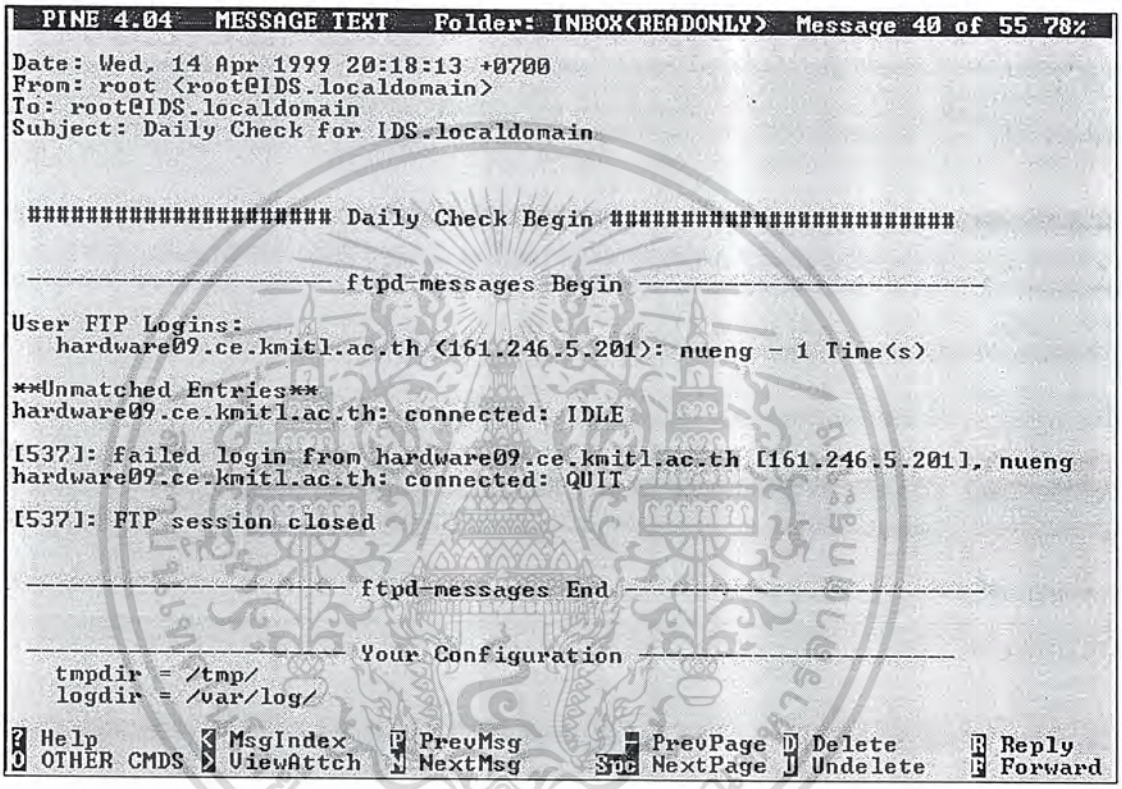
รูปที่ 4.1 แสดงผลการทดลองของเซอร์วิส cron

4.1.2 เซอร์วิส ftpd-messages

เมื่อทำการสั่งคำสั่งนี้ที่เชลล์

$ dailycheck - - service ftp-messages

ระบบจะได้รับเมล์รายงานผลจากเซอร์วิสนี้ด้วยรูปแบบดังนี้



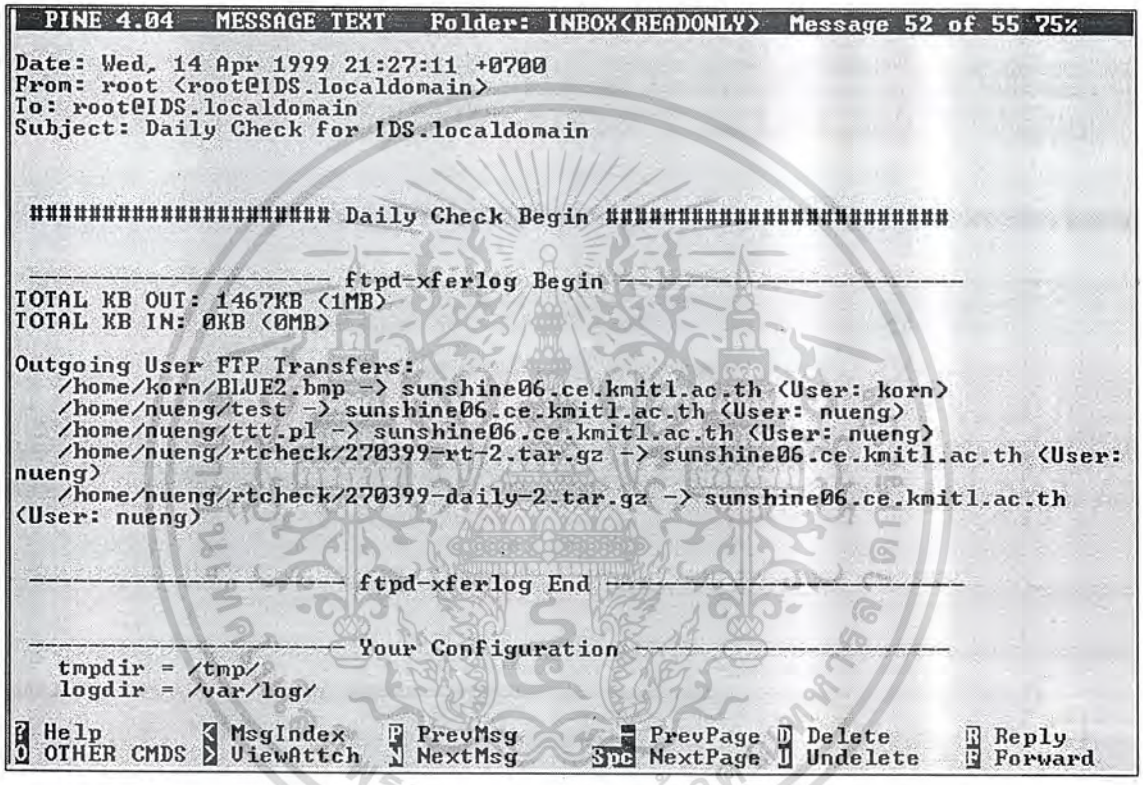
รูปที่ 4.2 แสดงผลการทดลองของเซอร์วิส ftpd-messages

4.1.3 เซอร์วิส ftpd-xferlog

เมื่อทำการสั่งคำสั่งนี้ที่เชลล์

$ dailycheck - - service ftp-xferlog - -detail 10

ระบบจะได้รับเมล์รายงานผลจากเซอร์วิสนี้ด้วยรูปแบบดังนี้



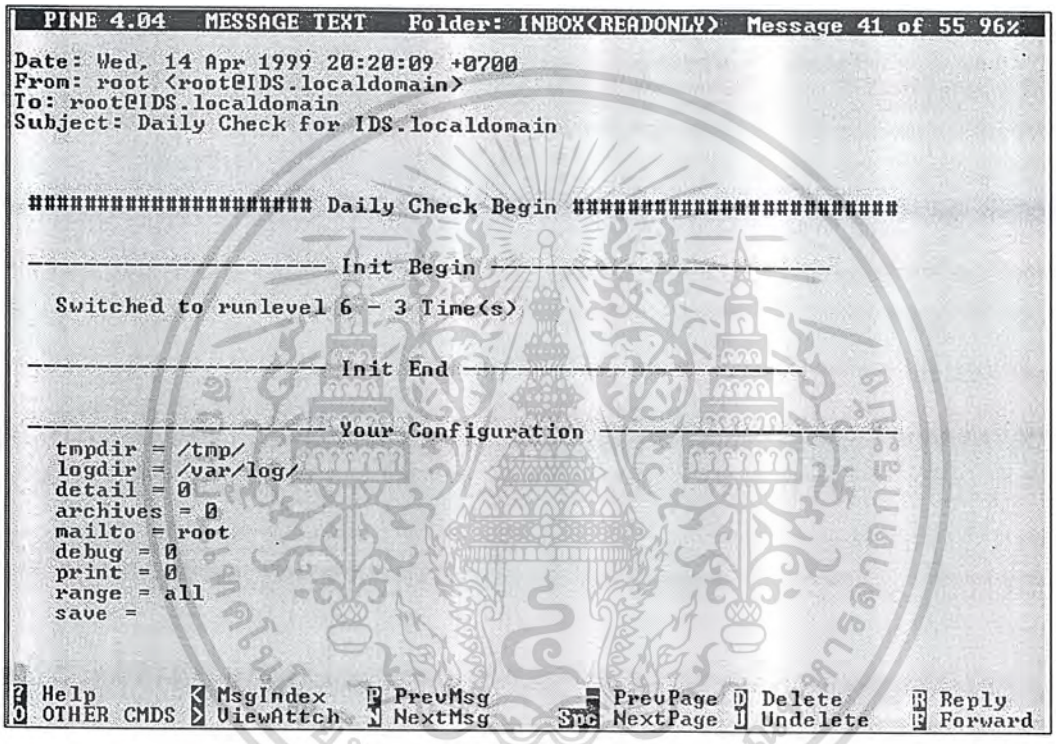
รูปที่ 4.3 แสดงผลการทดลองของเซอร์วิส ftpd-xferlog

4.1.4 เซอร์วิส init

เมื่อทำการสั่งคำสั่งนี้ที่เชลล์

$ dailycheck - - service init

ระบบจะได้รับเมล์รายงานผลจากเซอร์วิสนี้ด้วยรูปแบบดังนี้



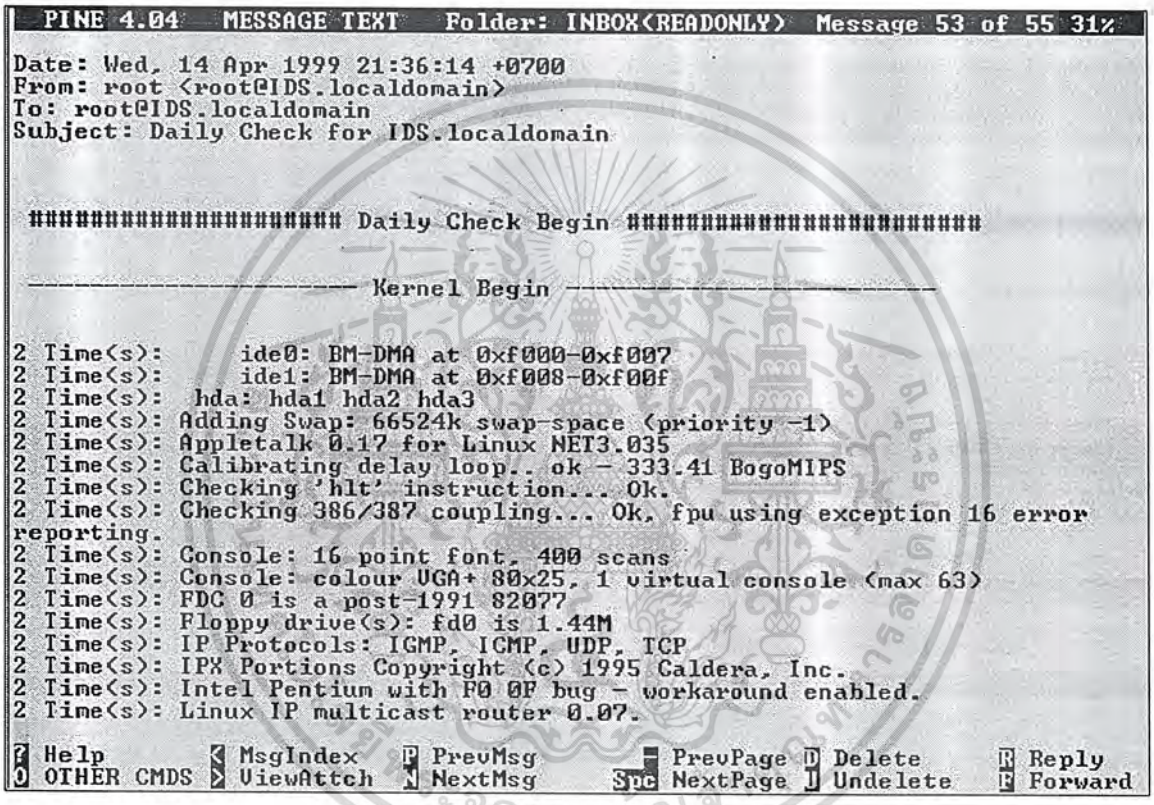
รูปที่ 4.4 แสดงผลการทดลองของเซอร์วิส init

4.1.5 เซอร์วิส kernel

เมื่อทำการสั่งคำสั่งนี้ที่เชลล์

$ dailycheck - - service kernel - -detail 5

ระบบจะได้รับเมล์รายงานผลจากเซอร์วิสนี้ด้วยรูปแบบดังนี้



รูปที่ 4.5 แสดงผลการทดลองของเซอร์วิส kernel

4.1.6 เซอร์วิส modprobe

เมื่อทำการสั่งคำสั่งนี้ที่เชลล์

$ dailycheck - - service modprobe - -detail 20

ระบบจะได้รับเมล์รายงานผลจากเซอร์วิสนี้ด้วยรูปแบบดังนี้

```
  PINE 4.04   MESSAGE TEXT   Folder: INBOX<READONLY>  Message 54 of 55 93%

Date: Wed, 14 Apr 1999 21:37:57 +0700
From: root <root@IDS.localdomain>
To: root@IDS.localdomain
Subject: Daily Check for IDS.localdomain


 ######################### Daily Check Begin ############################


 ---------------------- ModProbe Begin -----------------------

Can't locate these modules:
   binfmt--1025: 2 Time(s)


 ---------------------- ModProbe End -------------------------


 ---------------------- Your Configuration ---------------------
   tmpdir = /tmp/
   logdir = /var/log/
   detail = 20
   archives = 1
   mailto = root
   debug = 0
   print = 0
   range = all
   save =

? Help        < MsgIndex   P PrevMsg     - PrevPage  D Delete     R Reply
O OTHER CMDS  > ViewAttch  N NextMsg   Spc NextPage  U Undelete   F Forward
```

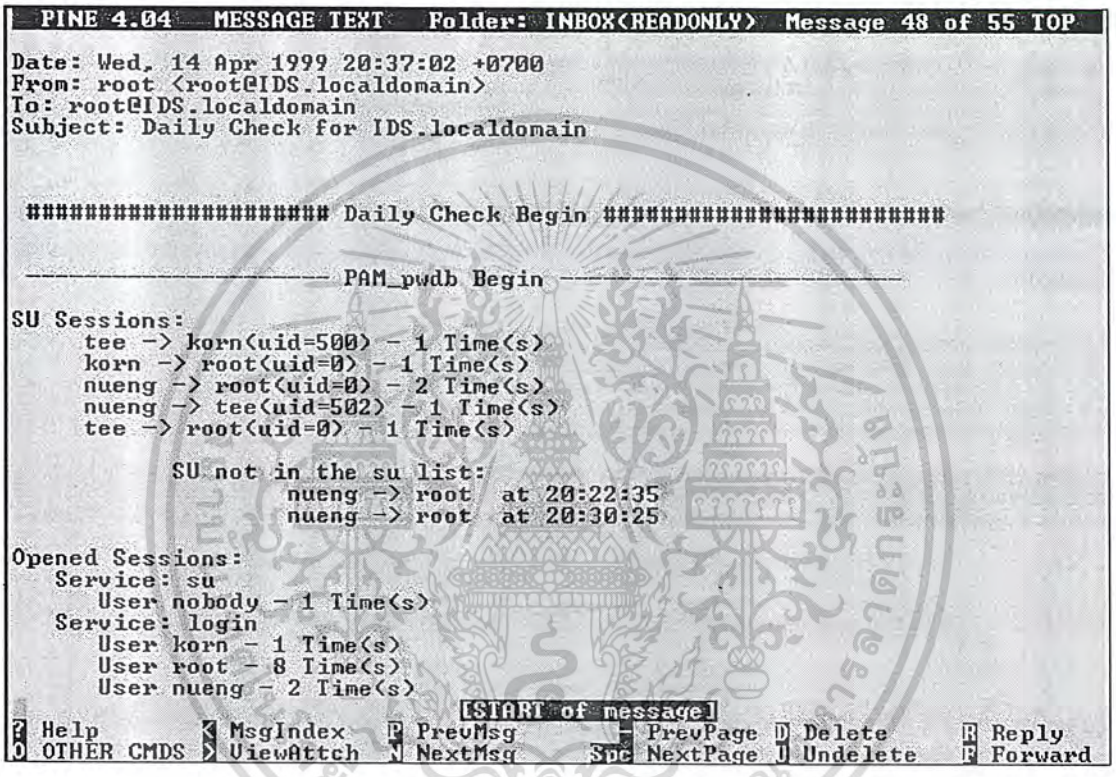
รูปที่ 4.6 แสดงผลการทดลองของเซอร์วิส modprobe

4.1.7 เซอร์วิส pam_pwdb

เมื่อทำการสั่งคำสั่งนี้ที่เชลล์

$ dailycheck - - service pam_pwdb

ระบบจะได้รับเมล์รายงานผลจากเซอร์วิสนี้ด้วยรูปแบบดังนี้



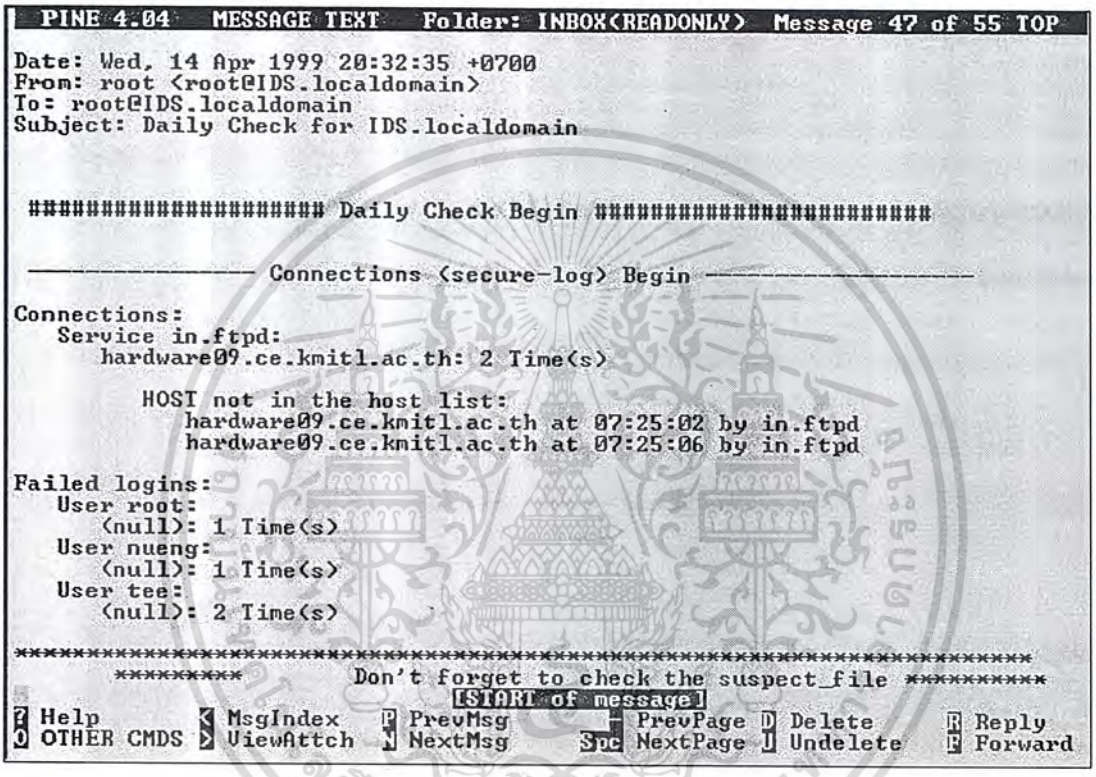
รูปที่ 4.7 แสดงผลการทดลองของเซอร์วิส pam_pwdb

4.1.8 เซอร์วิส secure

เมื่อทำการสั่งคำสั่งนี้ที่เชลล์

$ dailycheck - - service secure

ระบบจะได้รับเมล์รายงานผลจากเซอร์วิสนี้ด้วยรูปแบบดังนี้



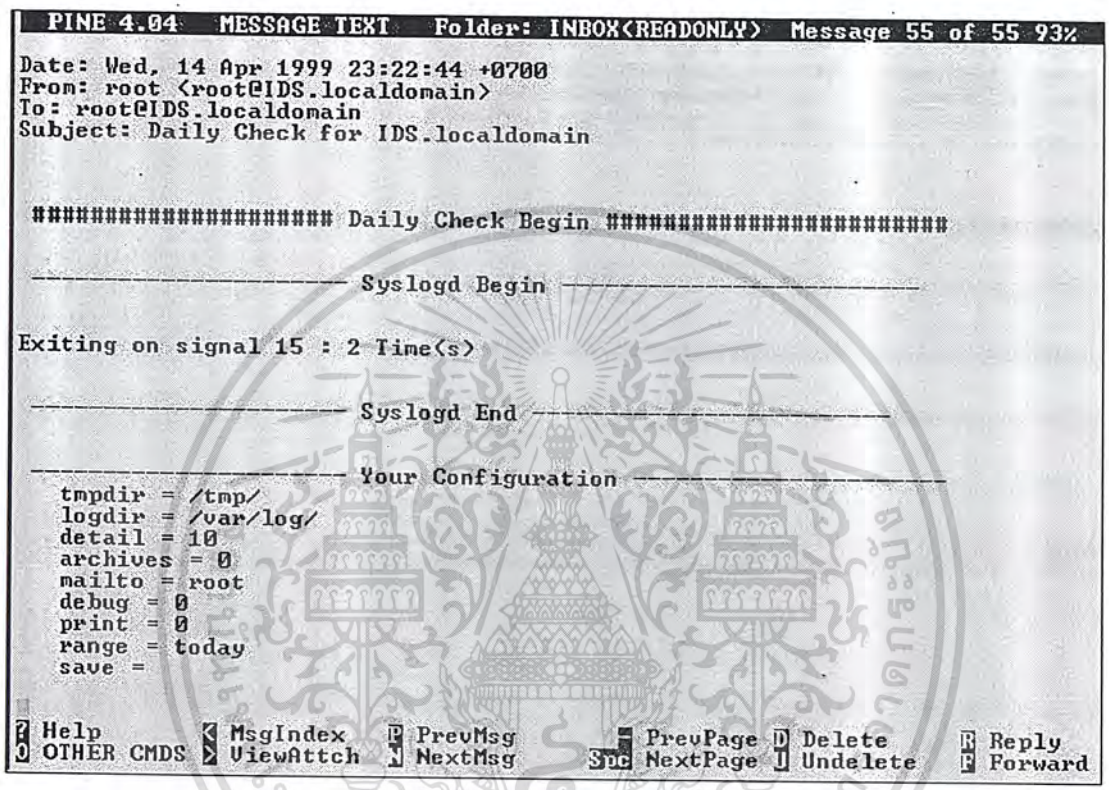
รูปที่ 4.8 แสดงผลการทดลองของเซอร์วิส secure

4.1.9 เซอร์วิส syslogd

เมื่อทำการสั่งคำสั่งนี้ที่เชลล์

$ dailycheck - -service syslogd - -detail 10

ระบบจะได้รับเมล์รายงานผลจากเซอร์วิสนี้ด้วยรูปแบบดังนี้



รูปที่ 4.9 แสดงผลการทดลองของเซอร์วิส syslogd

ทั้งหมดนี้เป็นผลของการทดลองของแต่ละเซอร์วิส แต่ในการทำงานจริงเรากำหนดให้ระบบนี้ทำทั้ง 9 เซอร์วิส และอาศัยบริการของโปรแกรมครอนเดมอนในการปลุกในส่วนวิเคราะห์และสร้างรายงานประจำวันนี้ทำงาน ซึ่งทำได้โดยการสร้างลิงก์ของไฟล์ dailycheck ขึ้นที่ /etc/cron.daily/ ให้ไปยังไฟล์ /etc/log.d/scripts/dailycheck.pl ต่อจากนั้นเราก็จะได้ระบบที่มีส่วนวิเคราะห์และสร้างรายงานประจำวันเพื่อทำหน้าที่แทนในการตรวจสอบล็อกไฟล์ด้วยเซอร์วิสที่เกี่ยวข้องกับความปลอดภัยต่างๆ ทั้ง 9 เซอร์วิส และในแต่ละวันผู้ดูแลระบบก็จะได้รับเมล์ที่มีตัวอย่างดังนี้

```
  PINE 4.04    MESSAGE TEXT                    Folder: INBOX  Message 56 of 57

Date: Wed, 14 Apr 1999 23:52:34 +0700
From: root <root@IDS.localdomain>
To: root@IDS.localdomain
Subject: Daily Check for IDS.localdomain


 ######################## Daily Check Begin ############################


 ---------------------- ftpd-messages Begin -------------------------

User FTP Logins:
   sunshine06.ce.kmitl.ac.th (161.246.6.86): korn - 1 Time(s)
   sunshine06.ce.kmitl.ac.th (161.246.6.86): nueng - 2 Time(s)

**Unmatched Entries**
sunshine06.ce.kmitl.ac.th: connected: IDLE

[1599]: failed login from sunshine06.ce.kmitl.ac.th [161.246.6.86], nueng
sunshine06.ce.kmitl.ac.th: connected: IDLE

[1599]: FTP session closed


 ----------------------- ftpd-messages End --------------------------


 ------------------------ Init Begin ----------------------------

   Switched to runlevel 6 - 2 Time(s)

 ------------------------ Init End ------------------------------


 ---------------------- PAM_pwdb Begin ----------------------------

SU Sessions:
     tee -> korn(uid=500) - 1 Time(s)
     korn -> root(uid=0) - 1 Time(s)
     nueng -> root(uid=0) - 3 Time(s)
     nueng -> tee(uid=502) - 1 Time(s)
     tee -> root(uid=0) - 2 Time(s)

          SU not in the su list:
               nueng -> root  at 20:22:35
               nueng -> root  at 20:30:25
               nueng -> root  at 23:23:59

Opened Sessions:
   Service: login
      User korn - 1 Time(s)
      User root - 8 Time(s)
      User nueng - 2 Time(s)
      User tee - 2 Time(s)


 ----------------------- PAM_pwdb End -------------------------


 ---------------- Connections (secure-log) Begin ------------------

Connections:
   Service in.ftpd:
      sunshine06.ce.kmitl.ac.th: 6 Time(s)
   Service in.telnetd:
      sunshine06.ce.kmitl.ac.th: 2 Time(s)

Failed logins:
```

รูปที่ 4.10 แสดงผลการทดลองเมื่อส่วนวิเคราะห์และสร้างรายงานประจำวันทำงาน

```
   User root:
      (null): 2 Time(s)
   User tee:
      (null): 2 Time(s)

*************************************************************************
      *********        Don't forget to check the suspect_file **********
*************************************************************************

 ------------------ Connections (secure-log) End ---------------------


 ---------------------- Syslogd Begin --------------------------

Exiting on signal 15 : 2 Time(s)

 ---------------------- Syslogd End ----------------------------

 ---------------------- Your Configuration ----------------------
   tmpdir = /tmp/
   logdir = /var/log/
   detail = 0
   archives = 0
   mailto = root
   debug = 0
   print = 0
   range = today
   save =
```

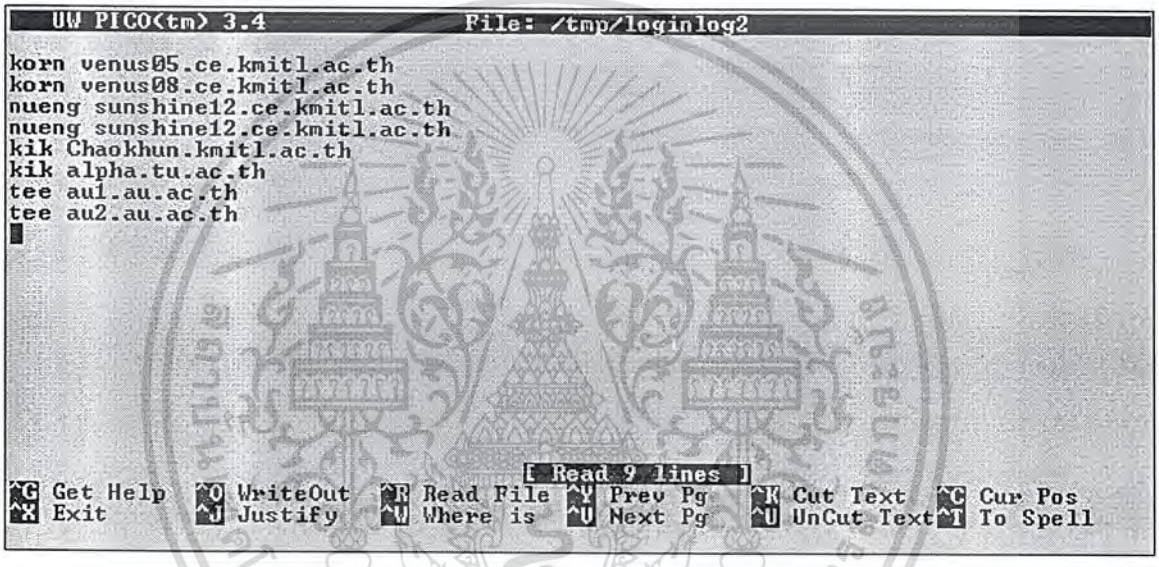
รูปที่ 4.10 แสดงผลการทดลองเมื่อส่วนวิเคราะห์และสร้างรายงานประจำวันทำงาน (ต่อ)

4.2 ผลการทดลองของส่วนวิเคราะห์และแจ้งเตือนแบบเรียลไทม์

ในหัวข้อนี้จะยกตัวอย่างส่วนที่ให้บริการแบบเรียลไทม์ทั้ง 3 ส่วนซึ่งได้แก่

4.2.1 การเช็คการล็อกอินจากผู้ใช้คนเดียวกันจากหลายโฮสต์

ตัวอย่างนี้จะแสดงถึงการที่ผู้ใช้ได้ทำการล็อกอินมาจากโฮสต์มากกว่าหนึ่งโฮสต์ สมมติให้มีผู้ใช้ทำการล็อกอินเข้ามายังระบบ ณ เวลาใดเวลาหนึ่ง เมื่อระบบเรียลไทม์เริ่มทำงาน ระบบจะมีการเก็บข้อมูลการล็อกอินของผู้ใช้แต่ละคนที่ทำการล็อกอินอยู่ในขณะนั้น เพื่อใช้ในการเปรียบเทียบข้อมูลกับข้อมูลที่ได้จาก ล็อกไฟล์ ตัวอย่างของข้อมูลที่ได้จากการล็อกอินแสดงดังรูปที่ 4.11



รูปที่ 4.11 รูปแสดงการเก็บข้อมูลที่ได้จากการล็อกอินของผู้ใช้ต่าง ๆ

หลังจากนั้นระบบจะทำการแยกรายชื่อของผู้ใช้แต่ละคนเมื่อมาทำการตรวจสอบโฮสต์เป็นลำดับ ซึ่งการตรวจสอบโฮสต์จะมีกรณีทำได้กำหนดไว้แล้ว 2 กรณี คือ

<u>กรณีที่ 1</u> ผู้ใช้ทำการล็อกอินมาจากโฮสต์เดียวกัน

โดยตัวอย่างนี้แสดงได้โดยผู้ใช้ที่ชื่อ nueng ทำการล็อกอินมาจากเครื่องชื่อ sunshine12.ce.kmitl.ac.th 2 ครั้ง ในกรณีนี้จะไม่ถือว่าผิดรูปแบบที่ได้กำหนดไว้ระบบจะไม่มีการแสดงผลออกมา

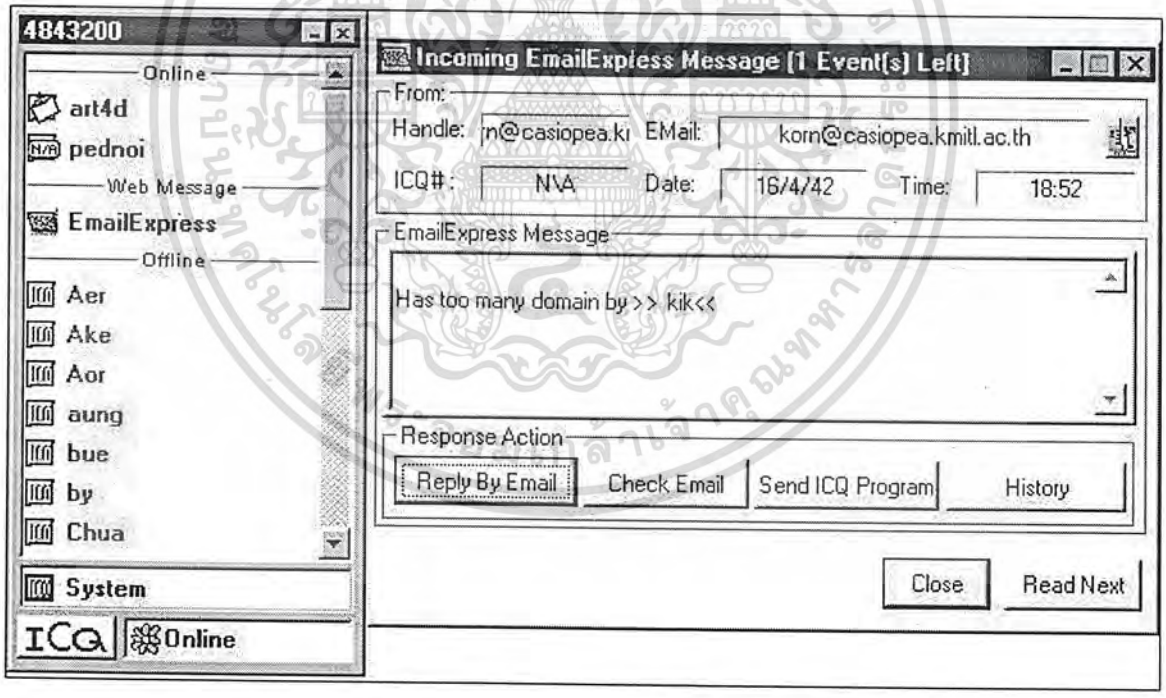
<u>กรณีที่ 2</u> ผู้ใช้ทำการล็อกอินมาจากโฮสต์ที่ต่างกัน

จะสังเกตได้ว่าจากรูปที่ 4.11 ผู้ใช้ที่ชื่อ nueng , tee และ kik ทำการล็อกอินมาจากโฮสต์ที่ต่างกัน สำหรับกรณีนี้จะแบ่งย่อยได้อีกเป็น 3 ส่วน ได้แก่

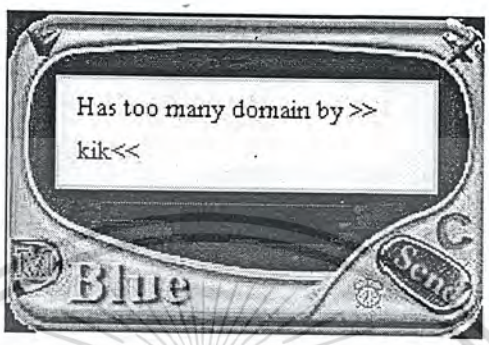
- ผู้ใช้ทำการล็อกอินมาจากโดเมนที่ต่างกัน

ได้แก่ ผู้ใช้ชื่อ kik ซึ่งได้ทำล็อกอินมาจากโดเมนที่ต่างกัน 2 โดเมนคือ kmitl.ac.th และ tu.ac.th ในกรณีนี้มีความเป็นไปได้น้อยมากที่ผู้ใช้จะทำการล็อกอินมาพร้อม ๆ กันจากคนละโดเมน ดังนั้นระบบจะทำการแจ้งเตือนไปยังผู้ดูแลระบบ เพื่อมาทำการตรวจสอบผู้ใช้คนนี้ก่อนที่จะระบบได้รับความเสียหาย

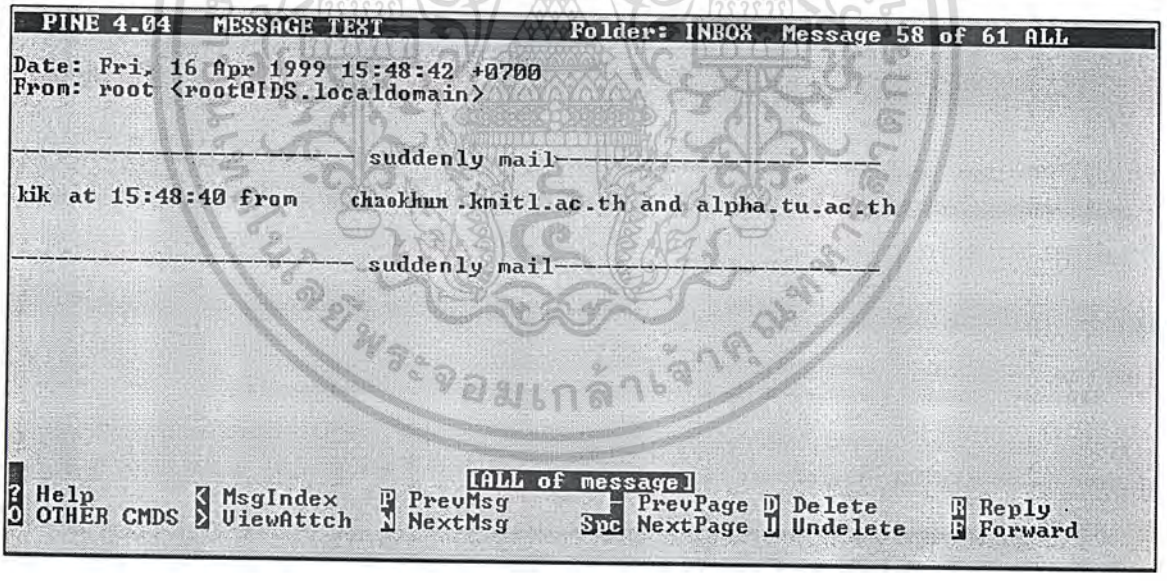
สำหรับวิธีการแจ้งเตือนได้กล่าวไว้แล้วในตอนต้นซึ่งได้แก่ การส่งผ่านทางโปรแกรมไอซีคิวและเพจเจอร์และที่ต่างจากการเตือนในแบบอื่น คือจะมีส่งเมล์ด้วย โดยมีลักษณะข้อความดังรูปที่ 4.12 , 4.13 และ 4.14



รูปที่ 4.12 แสดงข้อความที่ทำการส่งด้วยโปรแกรมไอซีคิว



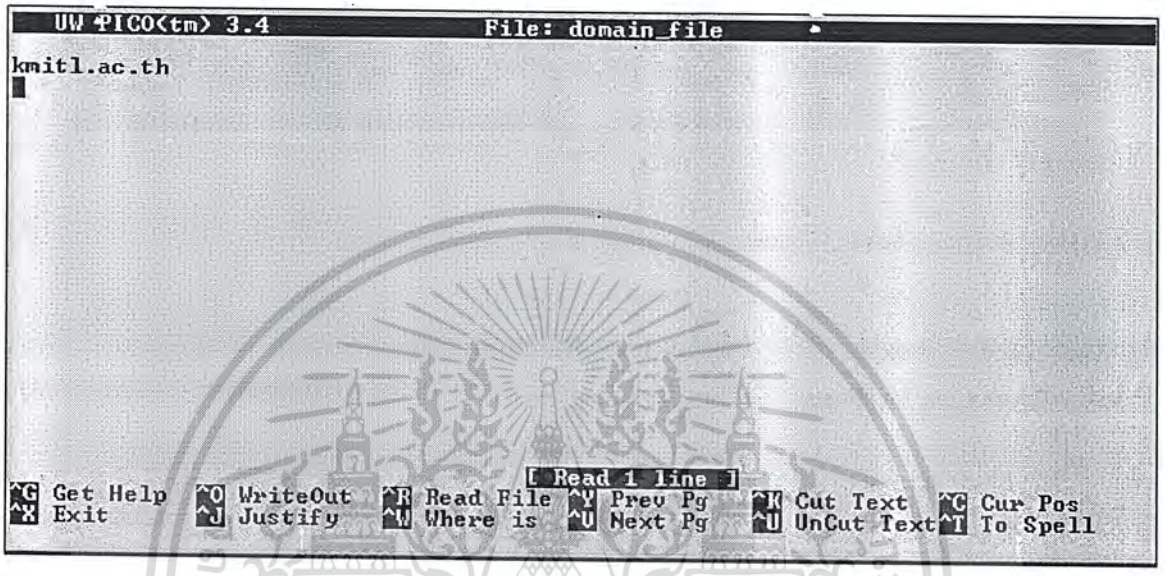
รูปที่ 4.13 แสดงข้อความที่ทำการส่งด้วยเพจเจอร์



รูปที่ 4.14 แสดงข้อความที่ทำการส่งด้วยเมล์

- ผู้ใช้ทำการล็อกอินมาจากโดเมนเดียวกันและเป็นโดเมนที่ระบบให้ความเชื่อถือ

ได้แก่ ผู้ใช้ที่ชื่อ korn ซึ่งได้ทำการล็อกอินมา 2 ครั้ง จากโดเมนเดียวกันแต่และเป็นโดเมนที่ระบบให้ความเชื่อถือ โดยที่ไฟล์ที่เก็บเกี่ยวกับโดเมนที่ระบบให้ความเชื่อถือเก็บอยู่ในไฟล์ domain_file แสดงดังรูปที่ 4.14



รูปที่ 4.15 แสดงรายชื่อของโดเมนที่ระบบให้ความเชื่อถือ

จากรูปที่ 4.11 จะตรวจสอบได้ว่า korn ทำการล็อกอินมาจากโดเมนที่ระบบให้ความเชื่อถือซึ่งก็คือโดเมน kmitl.ac.th ดังนั้นระบบจะไม่มีการแสดงเตือนออกมา เนื่องจากถือว่าเป็นไปได้ที่จะทำการล็อกอินมาจากเครื่องที่อยู่ใกล้เคียงกัน

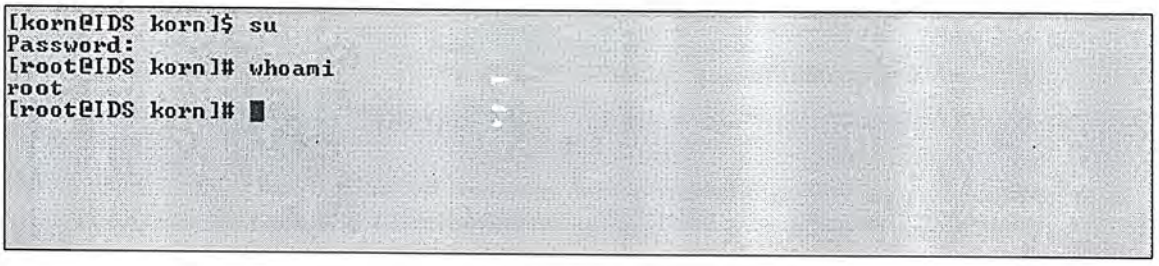
- ผู้ใช้ทำการล็อกอินมาจากโดเมนต์เดียวกันแต่เป็นโดเมนที่ระบบไม่ได้ให้ความเชื่อถือ

ได้แก่ ผู้ใช้ที่ชื่อ tee ซึ่งได้ทำการล็อกอินมา 2 ครั้งจาก au1.au.ac.th และ au2.au.ac.th ซึ่งเป็นโดเมนเดียวกัน แต่เมื่อนำมาตรวจสอบความน่าเชื่อถือกับไฟล์ domain_file แล้วปรากฏว่าไม่มีรายชื่อในนั้น แสดงว่าเป็นโดเมนที่ระบบไม่ได้ให้ความเชื่อถือ ดังนั้นระบบจะทำการแจ้งเตือนไปยังผู้ดูแลระบบ

สำหรับกรณีนี้ถือเป็นกรณีที่มีความเสี่ยงน้อย เนื่องจากมีความเป็นไปได้ที่ผู้ใช้คนเดียวจะทำการล็อกอินมาจาก 2 เครื่องที่อยู่ใกล้กัน ดังนั้นจึงไม่มีการแจ้งเตือนโดยทันทีด้วยโปรแกรมไอซีคิวและเพจเจอร์ แต่จะทำการเมล์ไปให้ผู้ดูแลระบบทราบเป็นรายงานประจำวัน

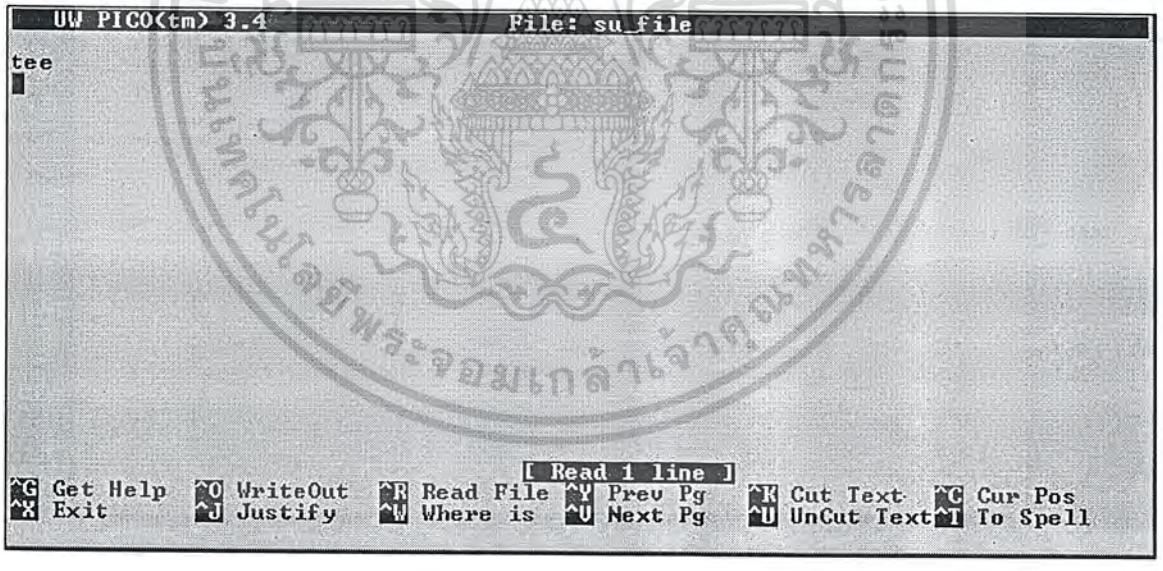
4.2.2 การเช็คการใช้คำสั่ง su เปลี่ยนสิทธิของตัวเองเป็นรูทของผู้ใช้ที่ไม่ได้มีสิทธิ

ตัวอย่างนี้แสดงถึงการที่ผู้ใช้ทำการเปลี่ยนสิทธิของตนเองไปเป็นรูท



รูปที่ 4.16 รูปแสดงการใช้คำสั่ง su เพื่อเปลี่ยนสิทธิเป็นรูท

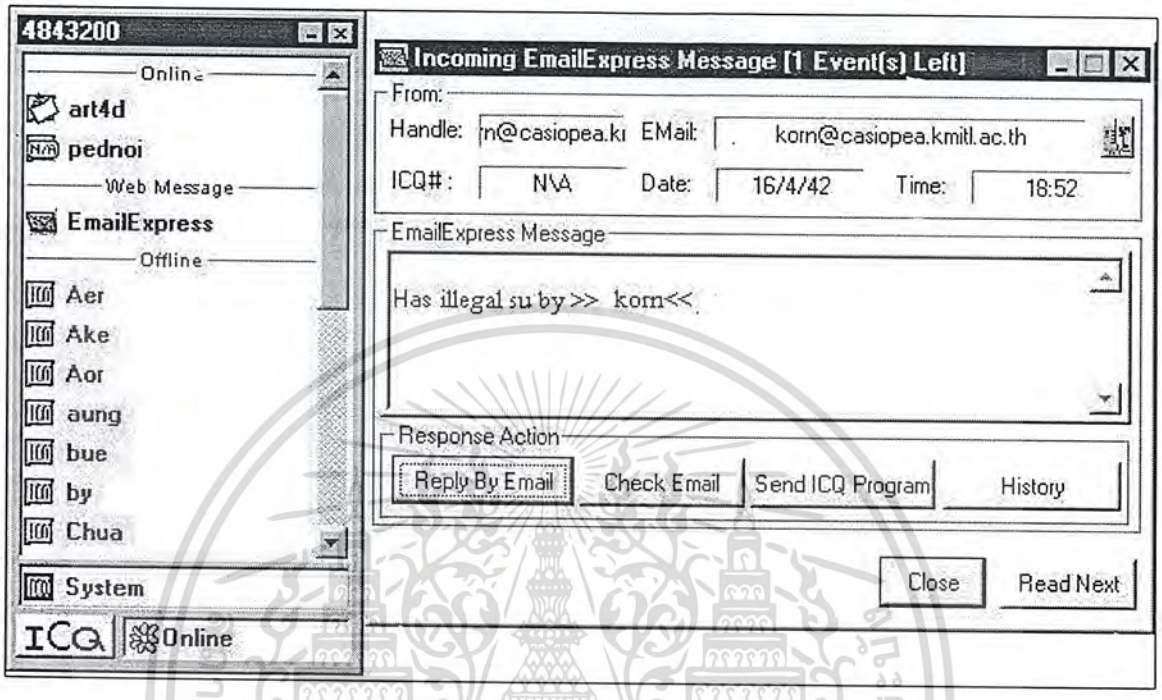
จากรูปที่ 4.16 แสดงตัวอย่างผู้ใช้คนหนึ่งที่ชื่อว่า korn ทำการเปลี่ยนสิทธิตนเองมาเป็นรูท โดยคำสั่ง su ได้สำเร็จ เมื่อระบบเรียลไทม์ทำงาน ระบบจะทำการตรวจสอบล็อกไฟล์ว่าใครทำการ su สำเร็จ ซึ่งก็จะพบว่ามีผู้ใช้ที่ชื่อ korn ทำการ su จากนั้นระบบจะนำชื่อที่ได้นี้มาตรวจสอบกับไฟล์ su_file สำหรับไฟล์นี้แสดงดังรูปที่ 4.17



รูปที่ 4.17 รูปแสดงตัวอย่างไฟล์ su_file

ระบบจะทำการเปรียบเทียบชื่อที่ได้จากล็อกไฟล์และไฟล์ su_file ว่ามีตรงกันหรือไม่ จากตัวอย่างผู้ใช้ที่ชื่อ korn ไม่มีชื่ออยู่ในไฟล์ su_file ซึ่งแสดงว่า korn ไม่ได้รับอนุญาตให้ทำการเปลี่ยนสิทธิเป็นรูทได้ ดังนั้นจึงพฤติกรรมที่ผิด

หลังจากที่ระบบทราบแล้วว่าเป็นการเปลี่ยนสิทธิไม่ถูกต้องจะทำการเตือนไปยังผู้ดูแลผ่านทางโปรแกรมไอซีคิวและเพจเจอร์ ทันที ดังรูปที่ 4.18 และ 4.19



รูปที่ 4.18 แสดงข้อความที่ทำการส่งด้วยโปรแกรมไอซีคิว

รูปที่ 4.19 แสดงข้อความที่ทำการส่งด้วยเพจเจอร์

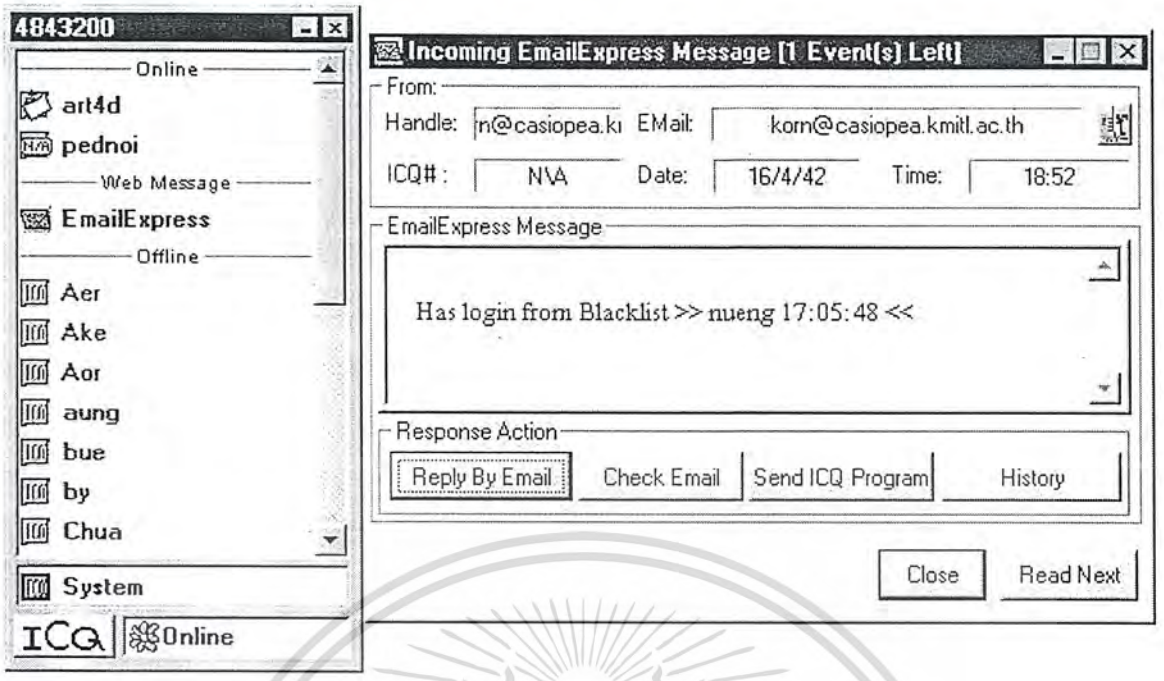
4.2.3 การเช็ครายชื่อผู้ใช้ที่ล็อกอินและเป็นผู้ใช้ที่ต้องจับตาดูเป็นพิเศษ

ตัวอย่างนี้แสดงถึงการตรวจจับผู้ใช้ที่มีพฤติกรรมเสี่ยงต่อการเป็นผู้บุกรุกและต้องจับตาดูเป็นพิเศษ

สมมติว่ามีผู้ใช้คนหนึ่งทำการล็อกอินเข้ามายังระบบ ซึ่งมีชื่อว่า nueng เมื่อระบบเรียลไทม์ทำงาน ระบบจะทำการตรวจล็อกไฟล์ เพื่อดูว่ามีผู้ใช้คนใดล็อกอินเข้ามาบ้าง จากนั้นจะนำรายชื่อของผู้ใช้ในล็อกไฟล์ มาเปรียบเทียบกับไฟล์ black_file

รูปที่ 4.20 แสดงไฟล์ black_file

สำหรับกรณีที่ยกมานี้จะเห็นได้ว่าผู้ใช้ที่ชื่อว่า nueng มีรายชื่ออยู่ใน black_file ด้วย ดังนั้นระบบก็จะทำการเตือนไปยังผู้ดูแลระบบโดยผ่านทางโปรแกรมไอซีคิวและเพจเจอร์ ซึ่งข้อความที่ส่งไปแสดงดังรูป 4.21 และ 4.22



รูปที่ 4.21 แสดงข้อความที่ทำการส่งด้วยโปรแกรมไอซีคิว

รูปที่ 4.22 แสดงข้อความที่ทำการส่งด้วยเพจเจอร์

บทที่5
สรุปผลการทำงาน

การทำงานของระบบเป็นที่น่าพอใจของผู้จัดทำ เพราะหากว่ามีผู้บุกรุกที่เป็นไปในกรณีที่ผู้จัดทำได้สร้างเซอร์วิสแบบเรียลไทม์รองรับไว้แล้ว ผู้ดูแลระบบก็สามารถมั่นใจในระบบการแจ้งเตือนทันทีได้ แต่หากจะทำให้ระบบนี้มีความสามารถที่ครอบคลุมถึงพฤติกรรมของแฮกเกอร์แล้ว ผู้ที่นำไปใช้ต้องทำการเขียนเซอร์วิสเพิ่มเติมตามแนวทางที่ผู้จัดทำได้แนะนำไว้แล้ว รวมถึงการรายงานประจำวันด้วย ผู้ใช้สามารถเพิ่มเติมการสรุปผลล็อกไฟล์ในรูปแบบอื่นๆได้ตามต้องการ โดยการวิเคราะห์เมสเซสในล็อกไฟล์ ผู้จัดทำเลือกใช้ภาษาเพิร์ล ซึ่งผู้จัดทำได้ใช้ความสามารถเกี่ยวกับการทำพาสซิ่ง(Parsing) โดยเขียนเป็น เรกกูล่าเอ็กซ์เพรสชั่น (Regular Expression) ซึ่งเป็นความสามารถที่โดดเด่นของภาษาเพิร์ลทำให้การเขียนเซอร์วิสสคริปต์ เป็นไปได้ด้วยดี ซึ่งหากผู้ใช้มีความถนัดในภาษาอื่นก็สามารถใช้ได้เช่นกัน

แต่ทว่าระบบนี้ก็ยังมีข้อจำกัดอยู่สาเหตุมาจากการใช้ sysklogd เป็นตัวออดิทเตอร์ (auditor) ถ้าหากว่าเหตุการณ์ใดที่ sysklogd ไม่สามารถจับเมสเซสมาได้ระบบก็ไม่สามารถวิเคราะห์ได้ ซึ่งผู้จัดทำเห็นว่าถ้าหากต้องการให้ระบบมีความสามารถยิ่งๆ ขึ้นไปก็ควรที่จะค้นหาตัวออดิทเตอร์ตัวใหม่ที่มีการทำงานคล้ายกับ sysklogd แต่มีความสามารถมากกว่าเพื่อให้ระบบมีขอบเขตการทำงานที่กว้างขวางมากขึ้น

สุดท้ายนี้ ผู้จัดทำขอแนะนะแนวทางการพัฒนาโครงงานในอนาคตสำหรับผู้สนใจต่อไป

1. เพิ่มเซอร์วิสเกี่ยวกับการตรวจสอบเมล์ในรายงานประจำวัน หมายถึงการตรวจปริมาณการเข้า – ออกของเมล์เพื่อเป็นแนวในการตรวจเรื่องเมล์บูมมิ่ง (Mail Booming)
2. เพิ่มเซอร์วิสเกี่ยวกับการถ่ายโอนไฟล์เข้า - ออกจากระบบแบบเรียลไทม์ เพื่อช่วยในการตรวจสอบว่าไฟล์ที่กำลังถ่ายโอนนั้นเป็นไฟล์ที่เป็นไฟล์ระบบหรือไม่
3. เพิ่มเซอร์วิสเกี่ยวกับเรื่องการตรวจสอบเอชทีทีพีดีล็อก (HTTPD Log) ในรายงานประจำวัน เนื่องจากในปัจจุบันการใช้เว็บเซิร์ฟเวอร์ (Web Server) เป็นที่นิยมมาก
4. เพิ่มเซอร์วิสเกี่ยวกับเรื่องการแสดงผลออกทางเว็บ ซึ่งเป็นรูปแบบที่ทำให้น่าดูยิ่งขึ้นและมีประโยชน์อยู่ 2 ประการ ได้แก่ เอกสารบนเว็บมีความสวยงามและสามารถเข้ามาตรวจดูได้ไม่ว่าจะอยู่ที่ใด (สำหรับการแสดงผลทางเว็บจะต้องทำการศึกษาเรื่องความปลอดภัยบนเว็บด้วยเนื่องจากถ้าไม่มีความปลอดภัย จะทำให้ผู้อื่นสามารถเข้ามาดูผลได้ และอาจเกิดอันตรายต่อระบบไม่ทางใดก็ทางหนึ่ง)

ภาคผนวก ก

<u>การติดตั้งระบบปฏิบัติการลีนุกซ์</u>

1. การสร้างแผ่นบู๊ตเพื่อติดตั้งลีนุกซ์

- ถ้าใช้วินโดว์ 95 อยู่ ให้ออกไปที่ดอสโหมด โดยคลิกที่ Start --- Shutdown --- เลือกที่ Restart the computer in MS-DOS Mode แล้วคลิกที่ OK
- ไปที่ไดร์ฟซีดีรอม (D: หรือ E:)
- พิมพ์ EZSTART แล้วกดปุ่ม Enter เพื่อเรียก Bat File สำหรับสร้าง แผ่นบู๊ตลีนุกซ์
- เครื่องจะถามว่า Are you sure you are not running under a shell (Y,N) ให้กด Y
- เครื่องจะถามว่า Enter the drive letter of your CD-ROM drive [C...Z] ให้ใส่ชื่อไดร์ฟของซีดีรอม (D หรือ E)
- จากนั้นจะให้เลือกวิธีการที่จะติดตั้งลีนุกซ์ดังนี้

Select one of the following installation methods:

A: Zero floppy diskette install. No floppy diskettes need. Make sure you not running in windows,but in DOS. This includes running as a DOS under windows.

B: NON-PCMCIA installation 1 3.5" formatted floppy diskette required Label Disk : BOOT

C: PCMCIA installation 2 3.5" formatted diskettes required Label disk 1: BOOT, Label disk 2 : SUPPLEMENTAL

D: Abort installation

Which installation method ? (A,B,C,D)? ให้เลือก B

- ใส่แผ่นดิสก์ที่ไดร์ฟ A แล้ว กดปุ่ม Enter ก็จะได้แผ่นสำหรับ ติดตั้งลีนุกซ์ต่อไป

2. เริ่มต้นติดตั้งลีนุกซ์

- ใส่แผ่นบู๊ตลีนุกซ์ที่ไดร์ฟ A แล้วบู๊ตเครื่องใหม่
- จะปรากฏข้อความต่อไปนี้

  Welcome to REDHAT Linux

  To install or upgrade a system running REDHAT Linux 2.0 or later press the <ENTER> key

  To enable the expert mode, type: expert <ENTER>

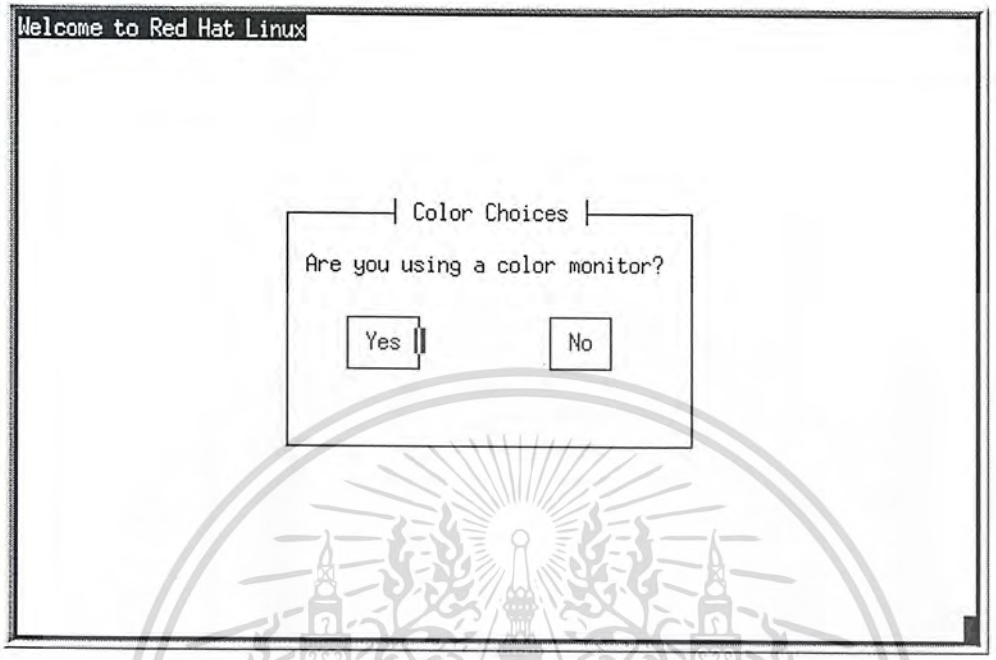
  To use this disk set for system repair or recovery, type: rescue <ENTER>

  Use the function keys listed below for help with all topics

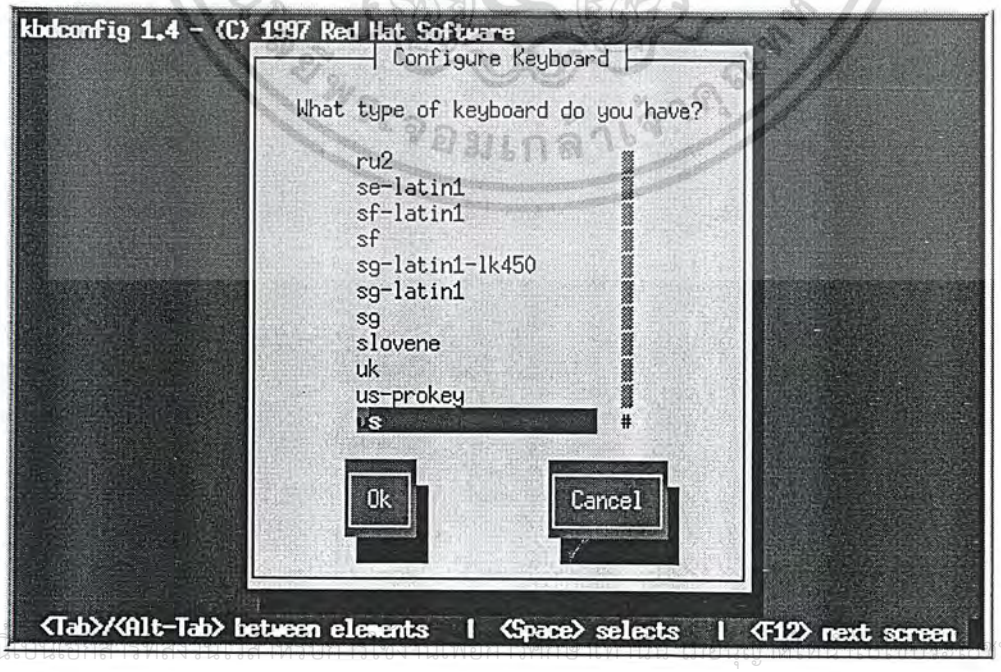
  Boot:

- ให้กดปุ่ม Enter

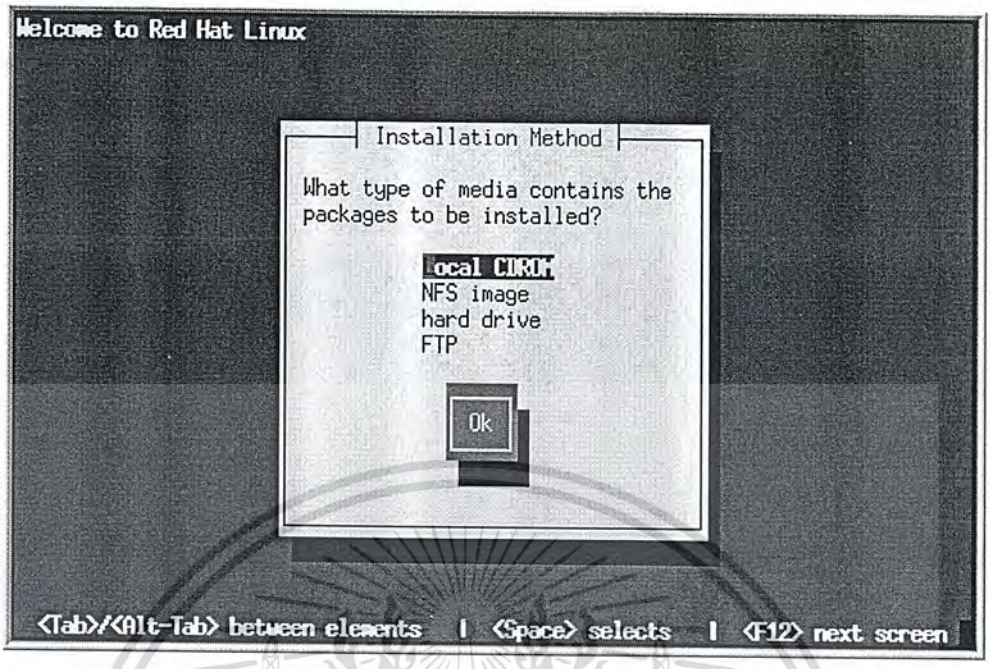
- รอสักครู่เพื่อโหลดไฟล์ในการติดตั้ง จากนั้นจะมีการตรวจสอบฮาร์ดแวร์ และจะมีกรอบ Color Choices ขึ้นมา



- ซึ่งจะถามว่าใช้จอสีอยู่หรือไม่ถ้าใช้ให้เลือกที่ Yes โดยการกดปุ่ม TAB เลือก แล้วกดปุ่ม Enter
- จะมีกรอบข้อความต้อนรับเข้าสู่ REDHAT LINUX ให้กดปุ่ม Enter
- แล้วจะปรากฏกรอบ Configure Keyboard เพื่อเลือกชนิดของคีย์บอร์ด การเลือกให้ใช้ ปุ่มลูกศร ขึ้น-ลง เลื่อนแถบสีน้ำเงินไปอยู่ในชนิดคีย์บอร์ดที่ต้องการ ในที่นี้ให้เลือก US แล้วกดปุ่ม Enter



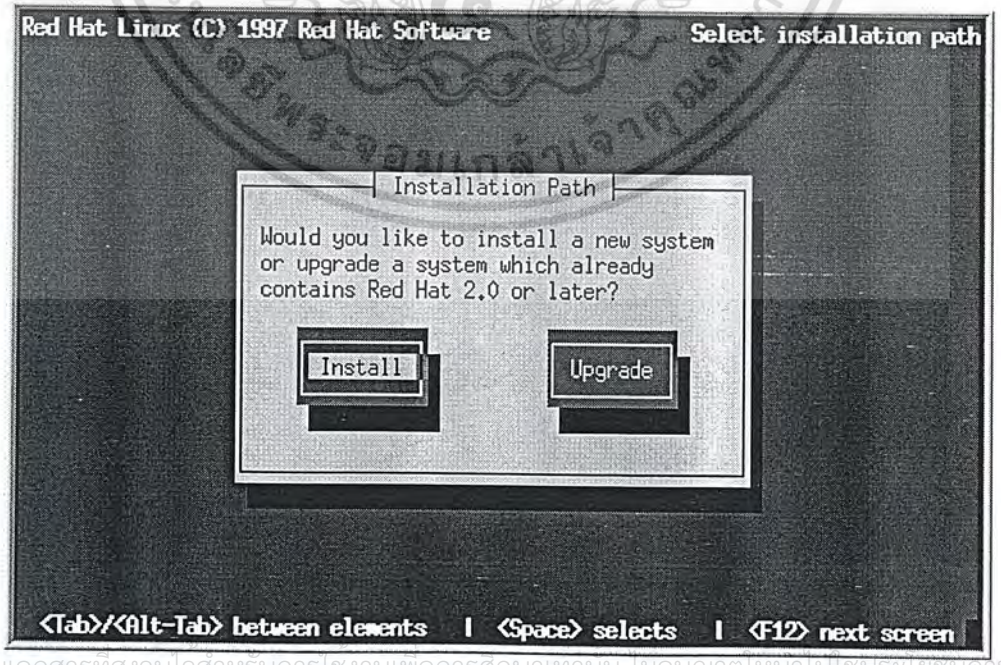
- จะมีกรอบให้เลือกวิธีในการติดตั้ง ให้เลือกที่ Local CDROM แล้วกดปุ่ม Enter



- ใส่แผ่นซีดีลงในไดร์ฟซีดีรอม แล้วกดปุ่ม Enter

3. ติดตั้งหรืออัพเกรด

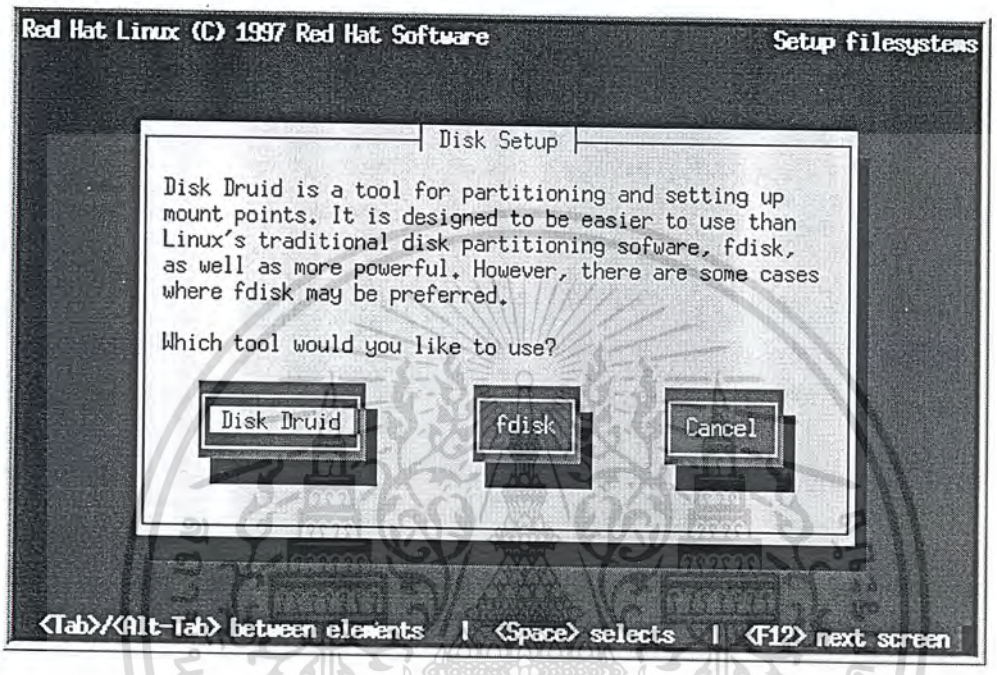
- จะมีกรอบให้เลือกว่าจะ Install หรือ Upgrade ของเดิมที่มีอยู่ ให้เลือกที่ Install แล้วกดปุ่ม Enter



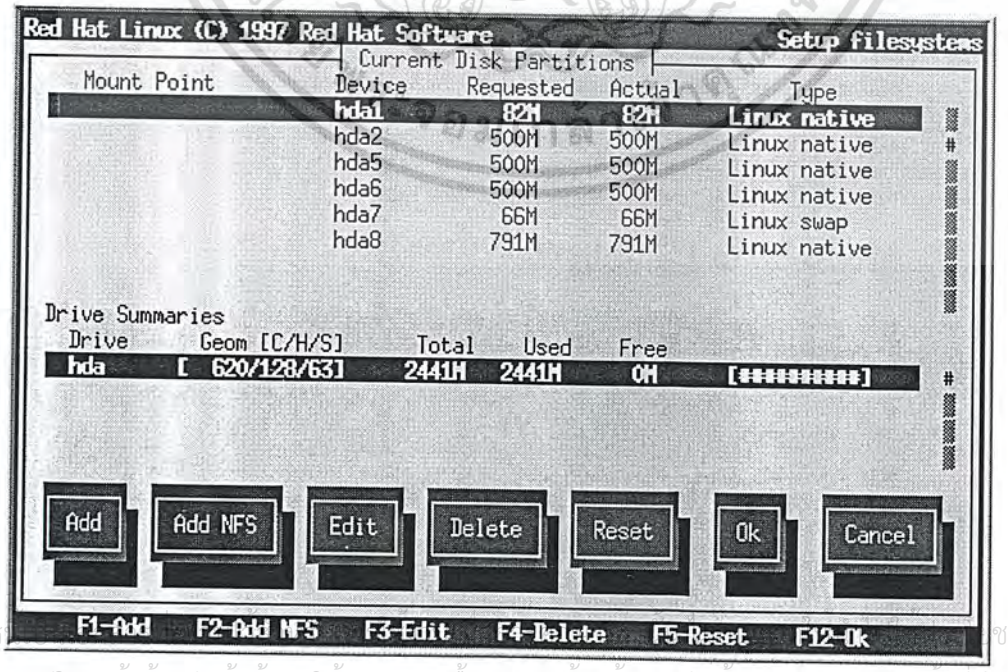
- จะมีกรอบสอบถามว่ามีการติดตั้งอุปกรณ์ SCSI เช่น Harddisk SCSI หรือไม่ ถ้าไม่มีให้เลือกที่ NO แล้วกดปุ่ม Enter

สร้างพาร์ทิชั่นสำหรับลีนุกซ์

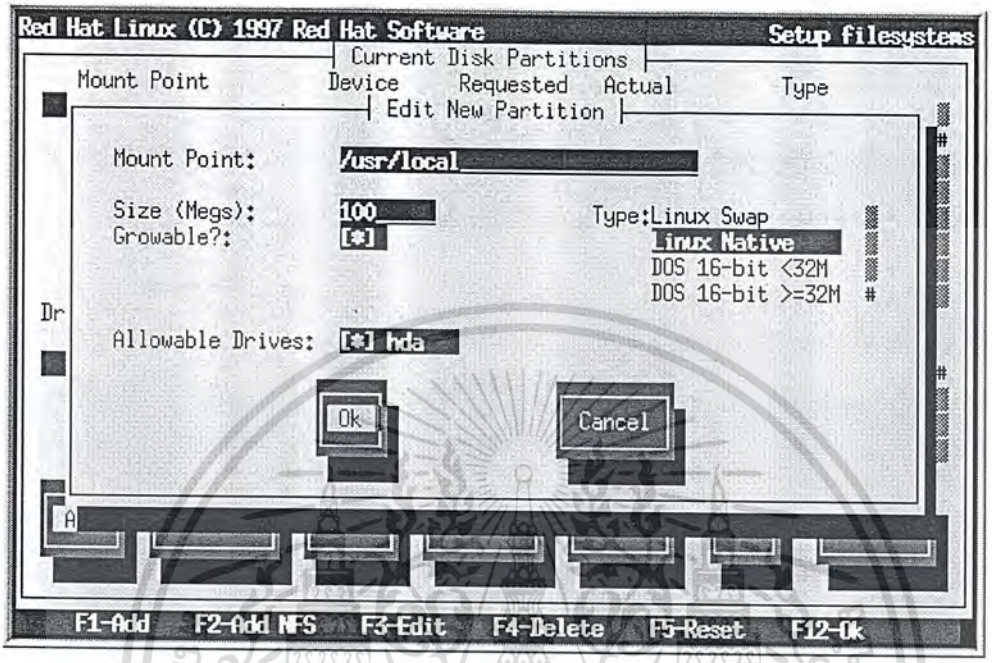
- จะมีกรอบให้เลือกเครื่องมือในการสร้างพาร์ทิชั่น



- สำหรับมือใหม่ให้เลือกที่ Disk Druid แล้วกดปุ่ม Enter
- จากนั้นจะมีกรอบ Current disk partition ขึ้นมา

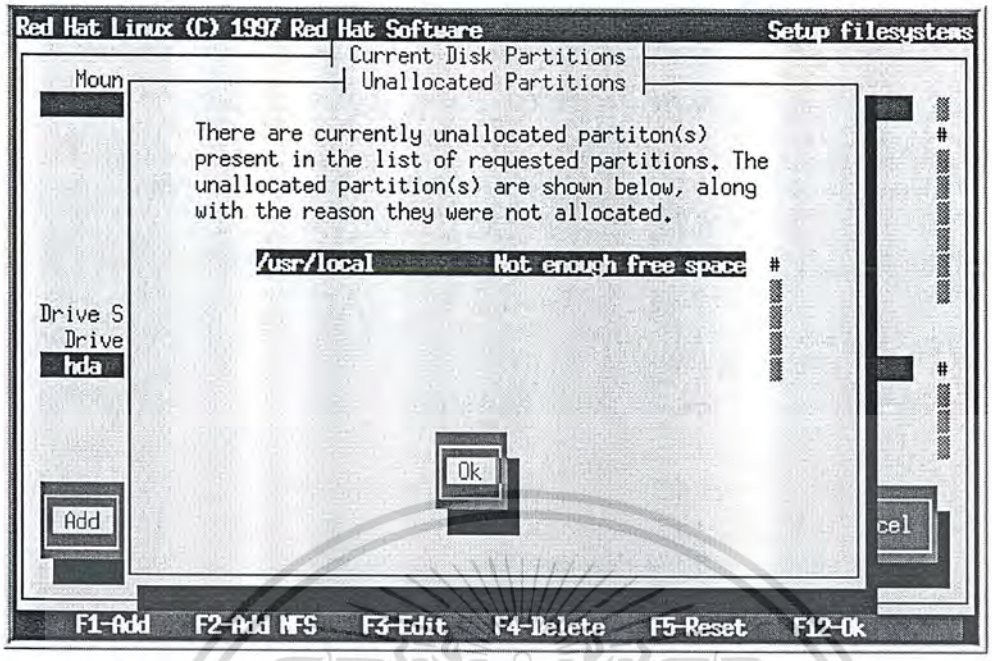


- รายละเอียดต่างๆ จะขอละไว้ ถ้าสนใจจะหาข้อมูลเพิ่มเติมสามารถอ่านได้ที่ www.redhat.com
- การเพิ่มพาร์ทิชั่นให้เลือกที่ ADD โดยกดปุ่ม TAB เพื่อเลื่อนตำแหน่ง จากนั้นกดปุ่ม Enter
- จะมีกรอบ Edit New Partition ขึ้นมา



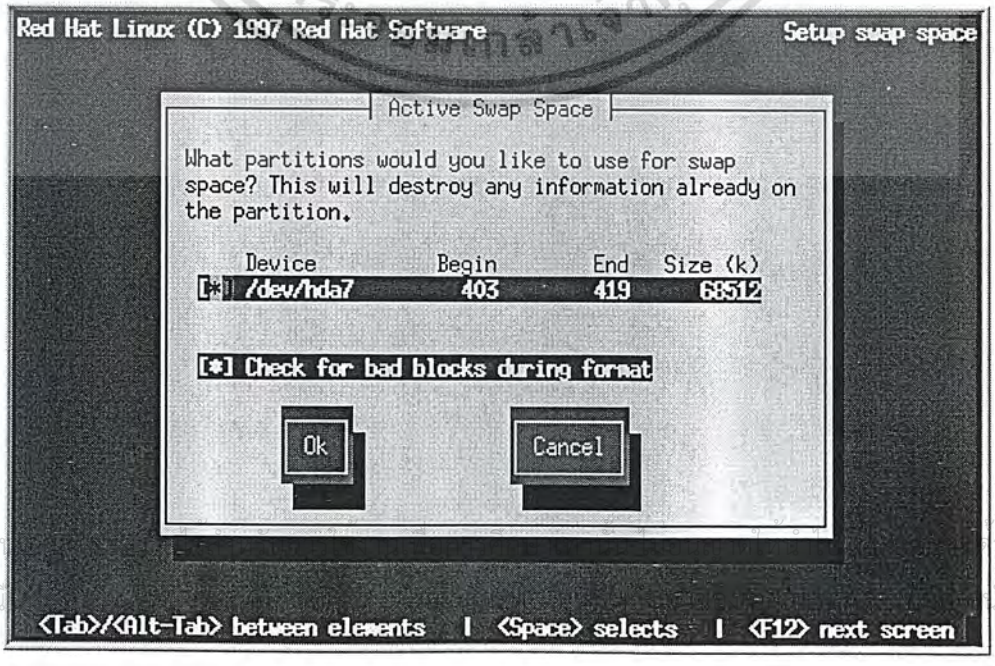
- ขั้นแรกให้สร้างพาริทิชั่น Swap ก่อน
  - ที่ Mount Point: พิมพ์ /swap แล้วกดปุ่ม TAB เพื่อเลือกหัวข้อต่อไป
  - Size (Megs): ให้เลือกขนาด 30 MB
  - Growable?: ให้กด spacebar ให้เครื่องหมายดอกจันขึ้นมา
  - Type: ให้เลือก Linux Swap
  - จากนั้นกดปุ่ม TAB ไปที่ปุ่ม OK แล้วกดปุ่ม Enter

- ต่อไปให้สร้างพาร์ทิชั่น Root
  - Mount Point: พิมพ์เครื่องหมาย / แล้วกดปุ่ม TAB เพื่อเลือกหัวข้อต่อไป
  - Size (Megs): ให้เลือกขนาด 389 MB หรือส่วนที่เหลือทั้งหมด หลังจากสร้าง Partition Swap แล้ว
  - Growable?: ให้กด spacebar ให้เครื่องหมายดอกจันขึ้นมา
  - Type: ให้เลือก Linux Native
  - จากนั้นกดปุ่ม TAB ไปที่ปุ่ม OK แล้วกดปุ่ม Enter
- กรณีที่สร้างพาร์ทิชั่นแล้ว มีเนื้อที่ในฮาร์ดดิสก์ไม่เพียงพอจะมีข้อความแจ้งเตือน ดังนี้

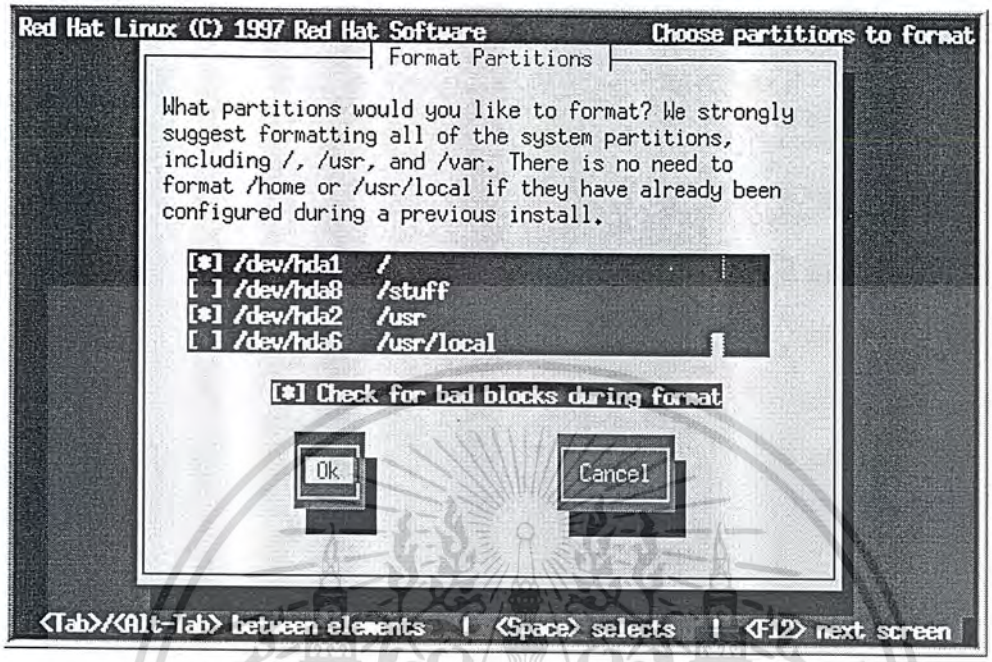


- ถ้าเกิดกรณีนี้ขึ้น ให้กดปุ่ม Enter ผ่านไปก่อน
- จากนั้นให้เลือกที่ ปุ่ม RESET เพื่อยกเลิกการสร้างพาร์ทิชั่นทั้งหมด
- แล้วจึงดำเนินการสร้างพาร์ทิชั่นใหม่โดยต้องคำนวณให้มีพื้นที่เพียงพอ

- เมื่อได้พาร์ทิชั่นตามต้องการแล้ว ให้เลือกที่ OK แล้วกดปุ่ม Enter
- จะมีกรอบ Save changes ขึ้นมา ให้เลือกที่ YES แล้วกดปุ่ม Enter
- จากนั้นจะมีกรอบ Active Swap Space ขึ้น



- ให้กดปุ่ม TAB ไปที่ OK แล้วกดปุ่ม Enter เพื่อทำการฟอร์แมตพาร์ทิชั่น Swap
- จากนั้นจะมีกรอบ Format Partitions อื่นๆ

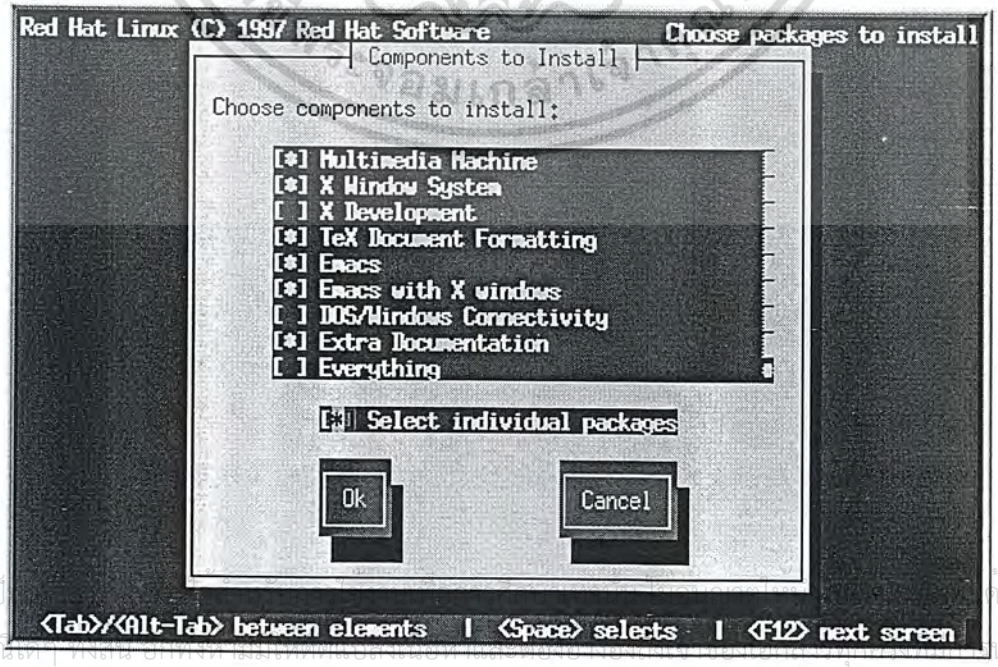


ให้กด Spacebar ให้มีเครื่องหมายดอกจันขึ้น เพื่อเลือก Format Partition ที่ต้องการ ในที่นี้ให้เลือกทั้งหมด

- กดปุ่ม TAB ไปที่ OK แล้วกดปุ่ม Enter

การเลือกและติดตั้งแพ็กเกจ

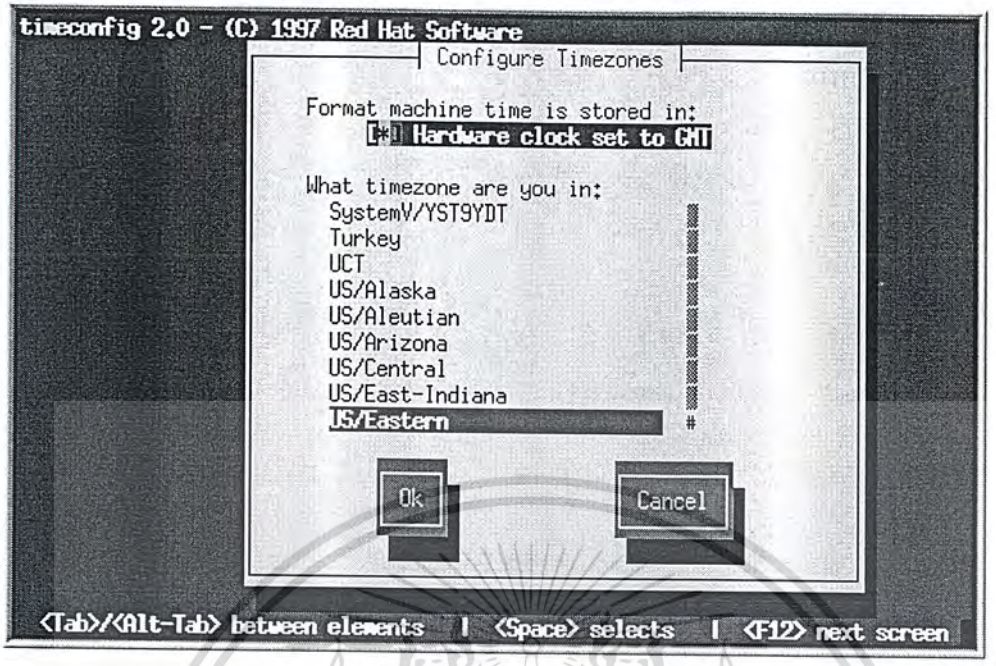
- จะมีกรอบ Components to install ขึ้น



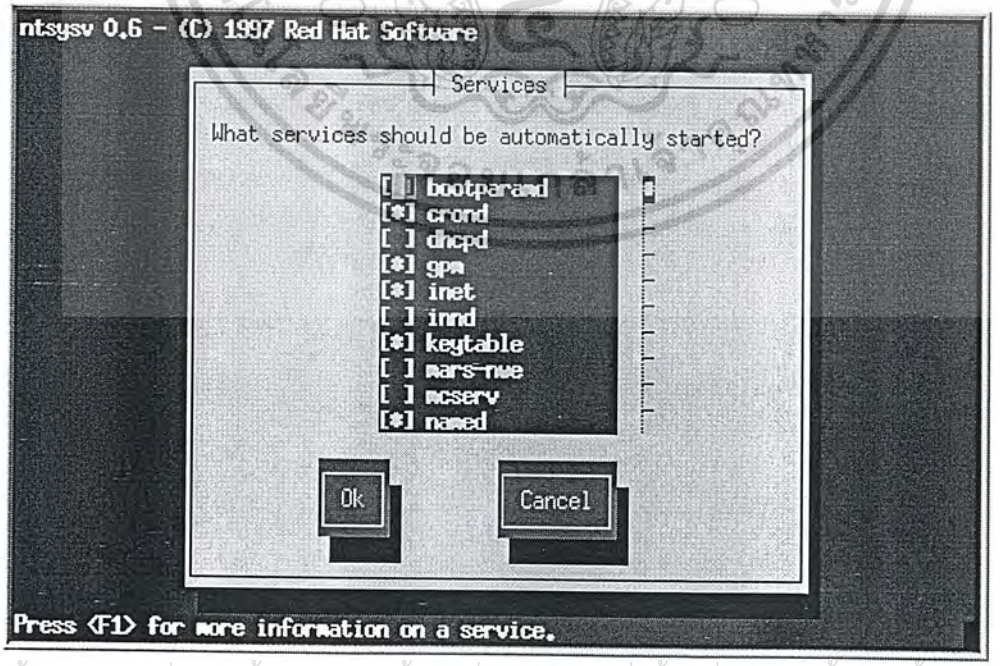
- การเลือกแพ็กเกจโดยการกด Spaecbar ที่แพ็กเกจที่ต้องการแล้วจะมีเครื่องหมายดอกจันขึ้น
- แพ็กเกจที่ควรจะเลือกในเบื้องต้นนี้มีดังนี้
  - Printer Support
  - X window system
  - Mail / WWW / New Tools
  - File managers
  - X Multimedia Support
  - Console Multimedia
  - Networked Workstation
  - Dialup Workstation
- ที่ Select individual packages ไม่ต้องเลือก
- เลือกที่ OK แล้ว กดปุ่ม Enter
- จะมีกรอบ Install Log ขึ้นมา ให้เลือกที่ OK แล้วกดปุ่ม Enter
- จากนั้นจะทำการติดตั้งแพ็กเกจที่ได้เลือกไว้ใช้เวลาประมาณ 10 นาที

การกอนฟิกเมาส์และวีดีโอการ์ด

- หลังจากติดตั้งแพ็กเกจเรียบร้อยแล้ว จะขึ้นกรอบ Probing Result เนื่องจากตรวจพบ Mouse ชนิด PS/2 ที่ port PSAUX กรณีที่เป็น Serial mouse จะขึ้นเป็นข้อความอื่น
- ให้กดปุ่ม Enter เพื่อทำงานต่อไป
- จะมีกรอบ Emulate Three buttons ขึ้นมา ให้เลือกที่ YES แล้วกดปุ่ม Enter
- จะมีกรอบ PCI Probe ขึ้นมาและจะมีรายละเอียดของการ์ดแสดงผล ให้กดปุ่ม Enter
- จะมีกรอบ Monitor Setup ให้เลือกรายการที่ตรงกับ monitor ที่ท่านมีมากที่สุด ถ้าไม่มีรายการที่ใกล้เคียง แนะนำให้เลือก MAG DX1495
- จากนั้นเลือกที่ OK แล้ว กดปุ่ม Enter
- จะปรากฏกรอบ Screen Configuration เพื่อกำหนดค่าให้กับระบบ X window ให้เลือกที่ Don't Probe แล้วกดปุ่ม Enter
- จะมีกรอบ Video Memory ขึ้นมา ให้เลือกจำนวนหน่วยความจำของการ์ดแสดงผลที่มีอยู่ แล้วกดปุ่ม Enter
- จะมีกรอบ Clock chip configuration ขึ้นมา ให้เลือกที่ No clockchip setting แล้วกดปุ่ม Enter
- จะมีกรอบ Select video mode ให้กดปุ่ม TAB ไปเลือกที่ 16 Bit 800x600 แล้วเลือกที่ OK และกดปุ่ม Enter
- จะมีกรอบ Network configuration ขึ้นมา ให้เลือก NO แล้วกดปุ่ม enter
- จะมีกรอบ Configure Times zones ให้เลือกที่ Asia/Bangkok เลือก OK และกดปุ่ม Enter

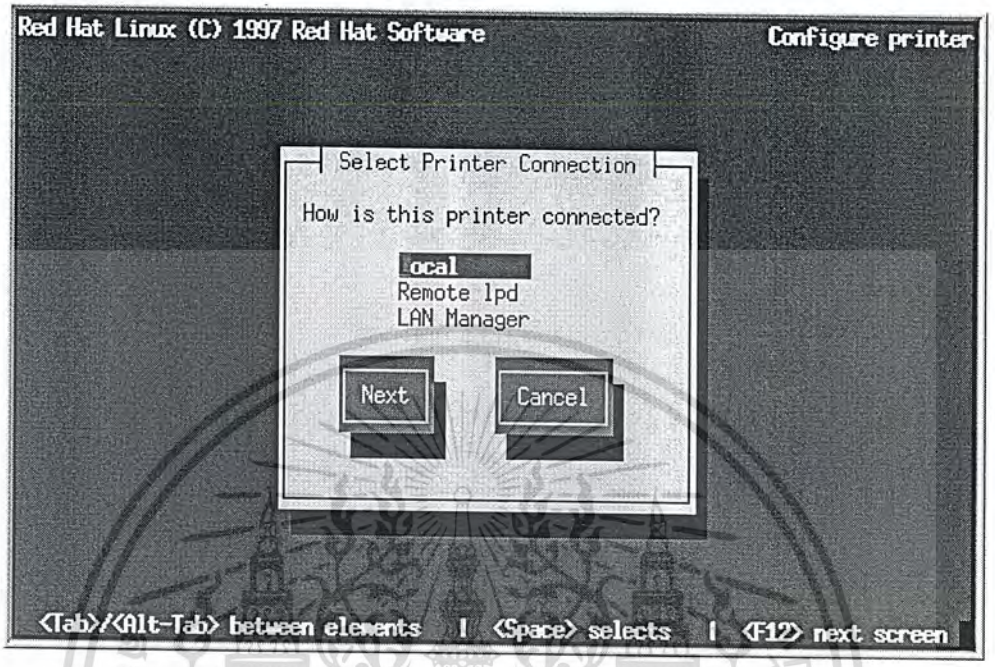


- จะมีกรอบ Services ขึ้นมา ให้เลือกที่ OK แล้วกดปุ่ม Enter

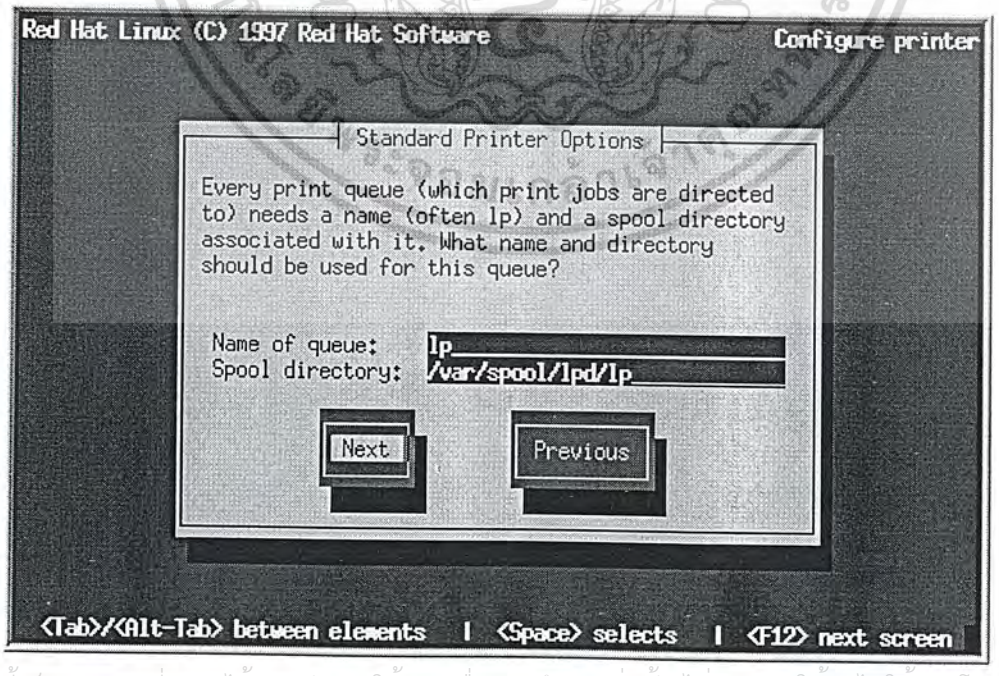


การ Config Printer

- จะมีกรอบ Configure Printer ถ้าไม่มีเครื่องพิมพ์ ให้เลือก NO ถ้ามีให้เลือก YES
- จะมีกรอบ Select printer Connection ขึ้นมา

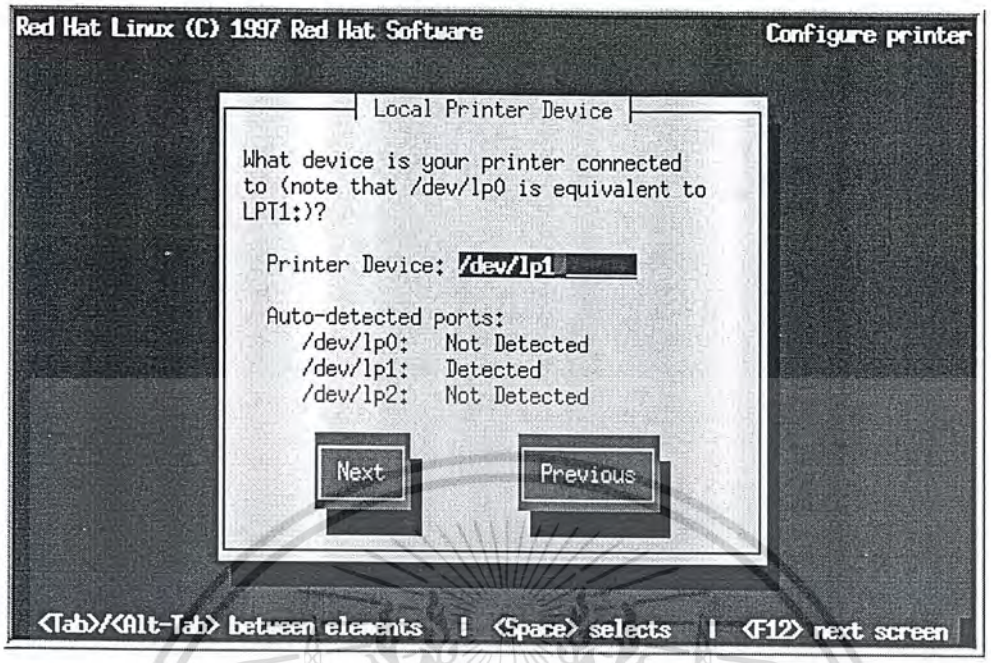


- ให้เลือกที่ local แล้วกดปุ่ม TAB ไปที่ Next และกดปุ่ม Enter
- จะมีกรอบ Standart Printer option ขึ้นมา

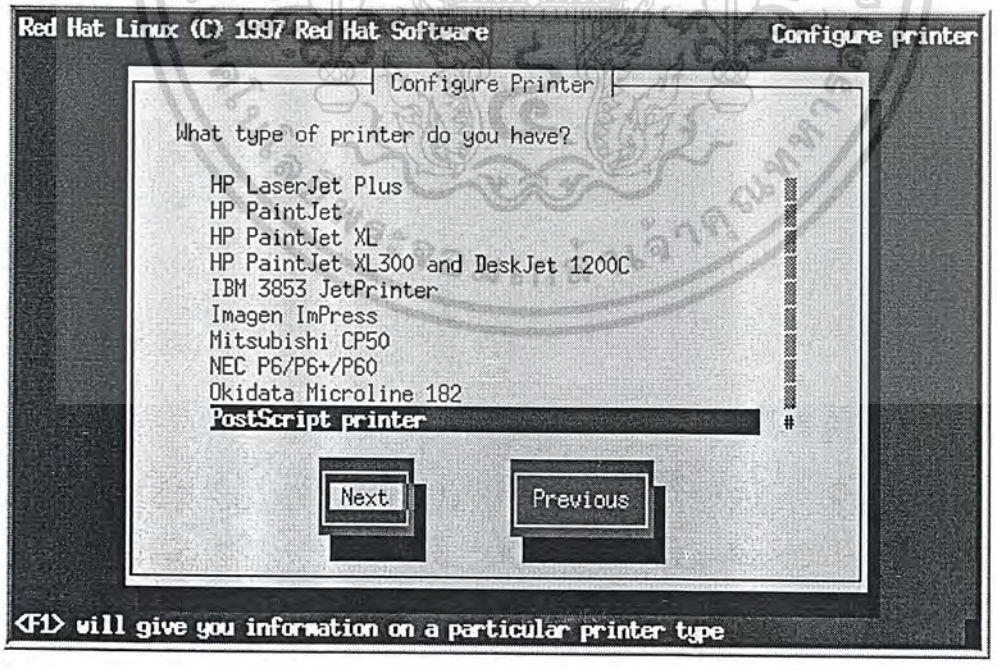


- ไม่ต้องแก้ไขค่าอะไร ให้กดปุ่ม TAB เพื่อเลือก Next แล้วกดปุ่ม Enter

- จะมีกรอบ local Printer Device ขึ้นมา

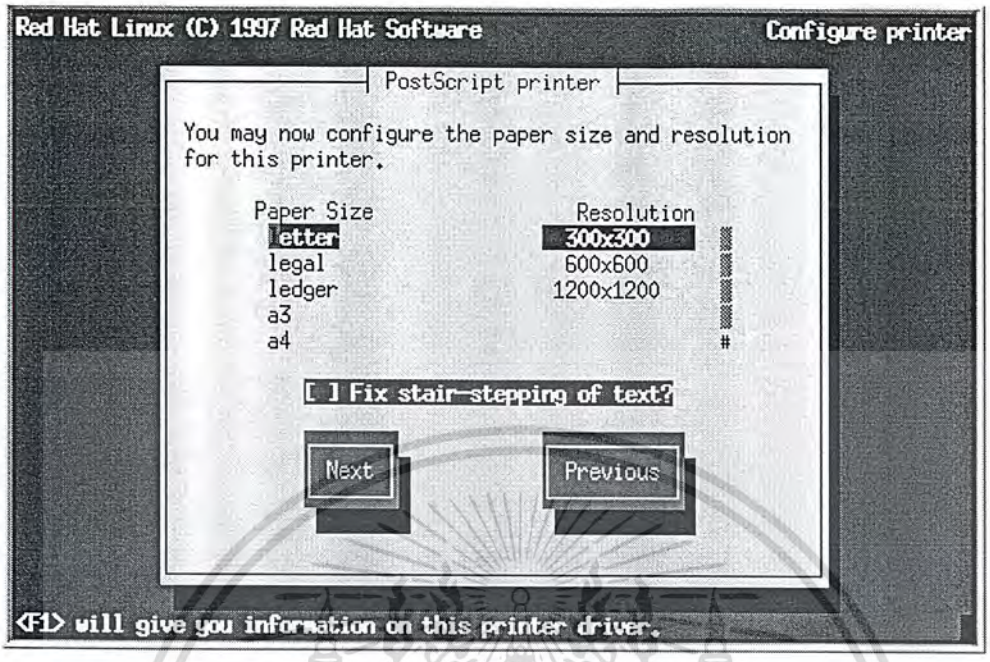


- ไม่ต้องแก้ไขค่าอะไร ให้กดปุ่ม TAB เพื่อเลือก Next แล้วกดปุ่ม Enter
- จะมีกรอบ Configure Printer ขึ้นมา

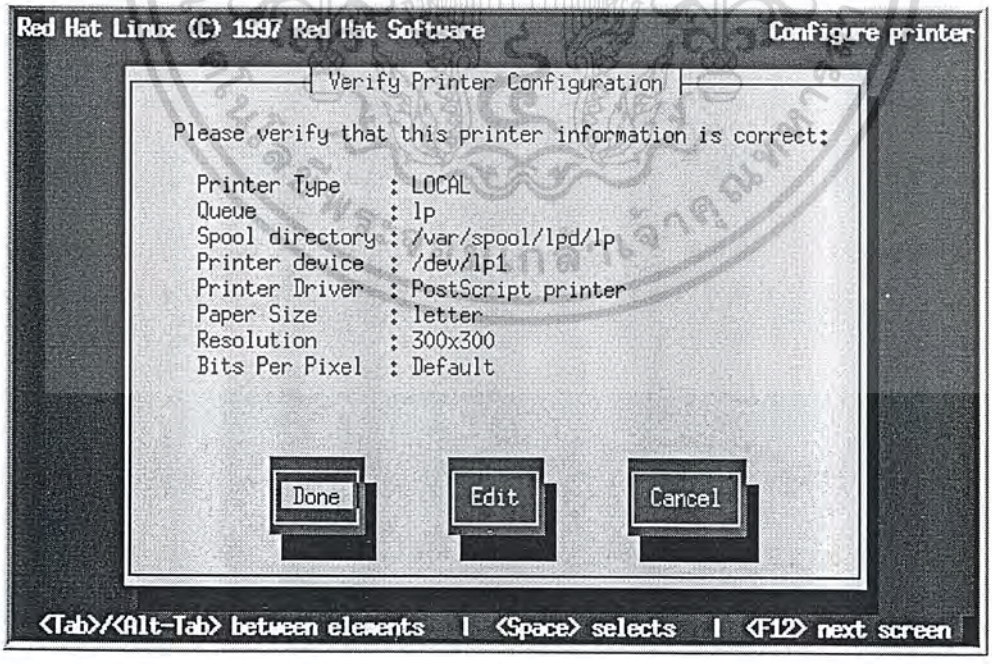


- ให้เลือกชนิดของเครื่องพิมพ์ให้ใกล้เคียงกับที่ท่านมีมากที่สุด แล้วกดปุ่ม TAB ไปที่ Next แล้วกดปุ่ม Enter

- จะมีกรอบให้เลือกชนิดของกระดาษขึ้นมา

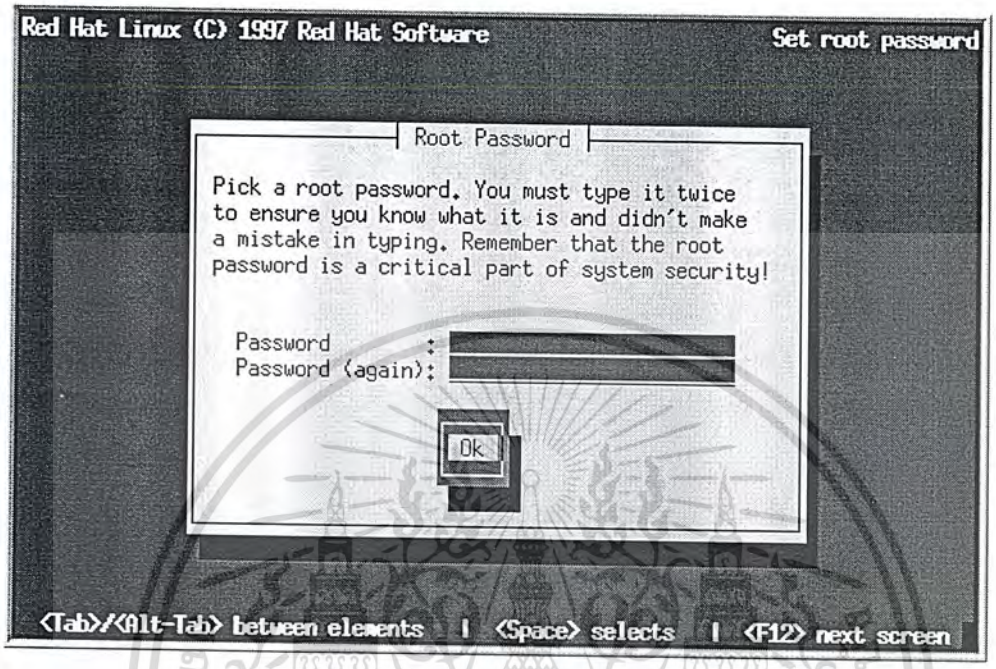


- ให้เลือก Paper Size เป็นชนิด A4 แล้วกดปุ่ม TAB เพื่อเลือก Resolution ถ้าเป็นเครื่องพิมพ์ชนิด Inkjet ให้เลือก 300x300 แล้วกดปุ่ม Tab เลือกที่ Next และกดปุ่ม Enter
- จะมีกรอบ Verify printer configuration ขึ้นมาให้เลือกที่ Done แล้วกดปุ่ม Enter



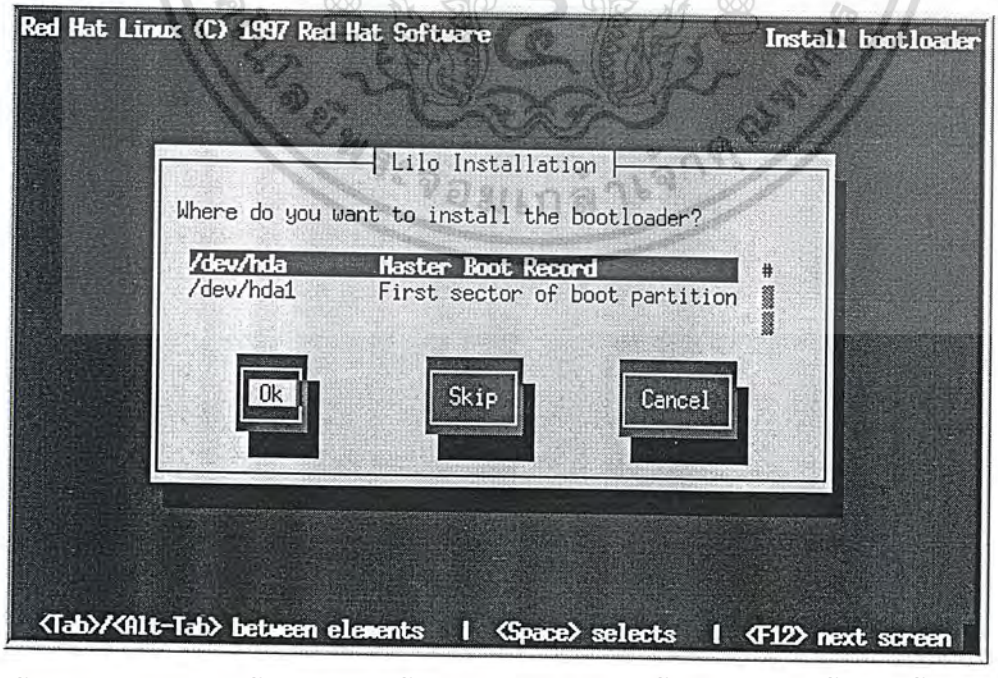
การกำหนดพาสเวิร์ดของรูท

- จะมีกรอบ Root Password ขึ้นมา ให้ใส่พาสเวิร์ดตามต้องการ โดยใส่ให้เหมือนกัน 2 ครั้ง แล้วเลือกที่ OK และ Enter



การติดตั้ง LILO (LInux LOader)

- จะมีกรอบ LILO Installation ขึ้นมา



- ให้เลือกที่ Master Boot Record แล้วกดปุ่ม TAB ไปที่ OK แล้วกดปุ่ม Enter

- จะมีกรอบ LILO Instalation อีกครั้ง ให้กดปุ่ม TAB ไปที่ Ok แล้วกดปุ่ม Enter
- จะมีกรอบ Bootable partition ขึ้นมา ให้เลือก Default Boot ที่ DOS โดยการกดปุ่ม F2 (สังเกต ดอกจัน) กดปุ่ม TAB ไปที่ OK แล้วกดปุ่ม Enter
- จะมีกรอบ Done แจ้งว่า การติดตั้งสมบูรณ์แล้ว ให้เอาแผ่น Boot Linux ออก แล้วกดปุ่ม Enter เพื่อบู๊ตระบบใหม่

คำสั่งแรกในลีนุกซ์

- เมื่อเปิดเครื่องขึ้นมาใหม่ เมื่อผ่านการตรวจสอบ Bios แล้ว จะมี Prompt LILO boot: ขึ้นมาให้พิมพ์ linux แล้วกดปุ่ม Enter ถ้าไม่พิมพ์อะไรภายใน 30 วินาที LILO จะบู๊ต Dos (windows95) ให้เอง
- จากนั้นลีนุกซ์ จะเริ่ม Start ระบบ จนถึงข้อความ

  RED HAT LINUX Release 5.0 (Hurricane)

  Kernel 2.0.32 on an i586

  local host login:
- ให้พิมพ์ root
- แล้วจะถาม password : ให้ใส่พาสเวิร์ดที่ได้กำหนดไว้ลงไป
- จากนั้นจะขึ้น [root@localhost /root]#
- ลองใช้คำสั่ง ls
- ลองใช้คำสั่ง cd /usr
- ใช้คำสั่ง ls
- ใช้คำสั่ง ls -l
- ลองกดปุ่ม Alt +ปุ่ม F2 จะมีหน้าจอใหม่ขึ้นมา หาต้องการกลับไปหน้าจอเดิม ให้กดปุ่ม Alt + ปุ่ม F1
- ใช้คำสั่ง startx ถ้าไม่มีปัญหาอะไร ระบบ x window จะทำงานลองขยับเมาส์ว่าทำงานได้ดีหรือไม่ หากต้องการออกจาก x window ให้กดปุ่ม Ctrl + Alt + Backspace
- การออกจากลีนุกซ์ ให้ใช้คำสั่ง shutdown -r now (ออกแล้วบู๊ตเครื่องใหม่) shutdown -h now (ออกแล้วปิดเครื่อง)

ภาคผนวก ข

คู่มือการใช้งานคำสั่ง dailycheck ผ่านทางคอมมานด์ไลน์

dailycheck [- -detail level] [- -logfile log-file-group] [- -service service-name] [- -print]
[- -mailto address] [- -archives] [- -range range] [- -debug level]
[- -save file-name] [- -help | - -usage]

- -detail	สามารถกำหนดเป็นตัวเลขได้หรือเป็นค่า
- -logfile	เป็นการกำหนดล็อกไฟล์ที่ต้องการให้ทำการวิเคราะห์
- -service	เป็นการกำหนดเซอร์วิสที่ต้องการให้ทำการวิเคราะห์
- -print	เป็นการเลือกที่จะดูผลที่หน้าจอ
- -mailto	เป็นการระบุแอดเดรสของผู้รับเมล์
- -archives	เป็นการระบุให้รวมเอาไฟล์ .? และ .?.gz มาวิเคราะห์ด้วย
- -range	เป็นการระบุขอบเขตของวันในการวิเคราะห์
- -debug	เป็นการระบุเลเวล (level) ของดีบักการประมวลผล
- -save	เป็นการเลือกที่จะให้ผลการวิเคราะห์ไปเก็บไว้ในไฟล์ที่กำหนด
- -help , --usage	เป็นการเรียกดูวิธีใช้ของคำสั่งนี้

บรรณานุกรม

เว็บไซต์อ้างอิง

[1] ftp://coast.cs.purdue.edu/pub/tools/unix/

[2] ftp://ftp.kaybee.org/pub/redhat/RPMS

[3] ftp://ftp.stanford.edu/

[4] http://comsc2.rajabhat.edu/raek/linux/

[5] http://gost.isi.edu/projects/crisis/

[6] http://infothailand.com/sitetutor/

[7] http://jgo.local.net/LinuxGuide/

[8] http://linux.thai.net

[9] http://lulu.mptc.eng.cmu.ac.th/HyperNews/get/ThaiLinux.html

[10] http://members.xoom.com/moha_kouros/unixzone.htm

[11] http://seclab.cs.ucdavis.edu/

[12] http://seclab.cs.ucdavis.edu/papers/

[13] http://software.thai.net/

[14] http://src.doc.ic.ac.uk/computing/bibliographies/Karlsruhe/Misc/intrusion.detection.html

[15] http://thailinux.hypermart.net

[16] http://webevents.broadcast.com/edu/sans/hackers1999a/

[17] http://webevents.broadcast.com/edu/sans/hackers1999a/slides/tsld001.ht

[18] http://www.ai.mit.edu/people/sodabot/

[19] http://www.ai.mit.edu/projects/hci/

[20] http://www.axent.com/

[21] http://www.axent.com/product/ita/ita.htm

[22] http://www.axent.com/swat/hacker.htm

[23] http://www.axent.com/swat/security.htm

[24] http://www.axent.com/swat/swat.htm

[25] http://www.cert.org

[26] http://www.cert.org/security-improvement/modules/m01.html

[27] http://www.choreosystems.com/vendors/haystack/

[28] http://www.cmds.net

[29] http://www.cs.purdue.edu/coast/intrusion-detection/

[30] http://www.deter.com/unix/

[31] http://www.fedu.uec.ac.jp/ZzzThai/Linux/

[32] http://www.freecode.com/

[33] http://www.geek-girl.com/ids/index.html

[34] http://www.geek-girl.com/resources.html

[35] http://www.gocsi.com/

[36] http://www.gocsi.com/ques.htm

[37] http://www.gslink.com/~mkb/

[38] http://www.hackersclub.com/

[39] http://www.haystack.com

[40] http://www.hideaway.net/linux.html

[41] http://www.hps.nu/security.html

[42] http://www.ibbnet.nl/~anne/keyboard.html

[43] http://www.infoworld.com/cgi-bin/displayTC.pl?/980316sb8-audit.htm

[44] http://www.intrusion.com/

[45] http://www.iss.net/

[46] http://www.kaybee.org/~kirk

[47] http://www.linuxbox.com/~taylor/4ltrwrd/

[48] http://www.linuxprogramming.com

[49] http://www.morehouse.org/hin/root/

[50] http://www.newsville.com/news/groups/

[51] http://www.nfr.net/

[52] http://www.oreilly.com/

[53] http://www.psionic.com/abacus/abacus.html

[54] http://www.redhat.com/linux-info/ldp/

[55] http://www.redhat.com:8080/HyperNews/get/khg.html

[56] http://www.sans.org/idresponse.htm

[57] http://www.school.net.th/linux-sis/

[58] http://www.secnet.com/papers/ids-html/

[59] http://www.SecureZone.com/

[60] http://www.SecureZone.com/Software/Intrusion_Detection/

[61] http://www.securitydynamics.com/

[62] http://www.stanford.edu/~chk/about.html

[63] http://www.techmall.com/techdocs/NP980407-5.html

[64] http://www.thaidev.com/

[65] http://www.thainet.org/linux/

[66] http://www.ticm.com/kb/faq/idsfaq.html

[67] http://www-rnks.informatik.tu-cottbus.de/~sobirey/ids.html

หนังสืออ้างอิง

[68] David A. Curry , " *Using C on the UNIX System* " , O 'Rielly & Associates Inc , 1998

[69] Karen A. Forcht , " *Computer Security Management* " , boyd & fraser publishing company , 1994

[70] Larry Wall : Tom Christiansen & Randal L. Schwartz , " *Programming PERL* " , O'Reilly , 1996

[71] Mark F. Komarinski , " *Linux Companion The Essential Guide For Users And System Administrators* " , Prentice Hall PTR , 1996

[72] Michael Beck : Harald Bohme : Mirko Dziadzdka : Ulrich Kunitz : Robert Magnus : Dirk Verworner , " *LINUX Kernel Internals* " , ADDISON-WESLEY , 1996

[73] Randal L. Schwartz , " *Learning Perl* " , O'Reilly & Associates Inc. , 1993