สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

โปรแกรมตรวจสอบช่องโหว่ความปลอดภัยระบบ

Security System Scanner

โดย

นางสาว พรรณสิริ งามมณีวัฒน์

นางสาว วรรณา พิสิฐภิญโญ

อาจารย์ที่ปรึกษา

อาจารย์ ธนา หงษ์สุวรรณ

อาจารย์ อัครเดช วัชระภูพงษ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลขหมู่.....

เลขทะเบียน 49900

วัน,เดือน,ปี - 2 เม.ย. 2547

.b.....

.i.....

โปรแกรมตรวจสอบช่องโหว่ความปลอดภัยระบบ

Security System Scanner

โดย

นางสาว พรรณสิริ งามมณีวัฒน์

นางสาว วรรณา พิสิฐภิญโญ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

b11376417

ปริญญานิพนธ์ปีการศึกษา 2545

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง โปรแกรมตรวจสอบช่องโหว่ความปลอดภัยระบบ

Security System Scanner

ผู้จัดทำ

1. นางสาว พรรณสิริ งามมณีวัฒน์ รหัสนักศึกษา 42010220
2. นางสาว วรรณา พิสิฐภิญโญ รหัสนักศึกษา 42010309

..... อาจารย์ที่ปรึกษา

(อาจารย์ ธนา หงษ์สุวรรณ)

..... อาจารย์ที่ปรึกษา

(อาจารย์ อัครเดช วัชระภูพงษ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมตรวจสอบช่องโหว่ความปลอดภัยระบบ

นางสาวพรรณสิริ งามมณีวัฒน์ 42010220
นางสาววรรณา พิสิฐภิญโญ 42010309
อาจารย์ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษา
อาจารย์อัครเดช วัชระภูพงษ์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2545

บทคัดย่อ

ในปัจจุบันระบบคอมพิวเตอร์มีความสำคัญต่อระบบการทำงานหลายๆ งาน ส่วนใหญ่ก็จะเป็นที่เก็บข้อมูลสำคัญของหน่วยงานนั้นๆ การรักษาความปลอดภัยของข้อมูลจึงเป็นสิ่งสำคัญอย่างยิ่งที่ควรคำนึงถึง ซึ่งการป้องกันระบบคอมพิวเตอร์ก็สามารถทำได้หลายวิธี ทั้งไฟร์วอลล์หรือโปรแกรมตรวจสอบแบบทันท่วงที แต่ถ้าผู้บุกรุกสามารถเจาะระบบเข้ามาได้โดยผ่านระบบป้องกันต่างๆ แล้ว ด่านสุดท้ายก็คงต้องเป็นความแข็งแกร่งของตัวระบบเอง

โครงงานนี้จึงเป็นโครงงานที่พัฒนาขึ้นบนระบบปฏิบัติการลีนุกซ์เพื่อเป็นเครื่องมือช่วยผู้ดูแลระบบตรวจสอบระบบของตนเอง ซึ่งจะมุ่งเน้นการตรวจสอบช่องโหว่ที่เกิดจากความผิดพลาดของการตั้งค่าระบบของผู้ดูแลเป็นส่วนใหญ่ เพื่อเพิ่มความมีประสิทธิภาพในการป้องกันผู้บุกรุกได้ดียิ่งขึ้น และมีหลักการที่ว่าเมื่อตรวจพบช่องโหว่หรือจุดที่อาจเป็นช่องโหว่ก็จะรายงานผลเก็บไว้ในล็อกไฟล์ โดยใช้เรื่องของสิทธิการใช้งานควบกับคำสั่ง find ในการตรวจสอบ และใช้ภาษาเพิร์ลในการพัฒนา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Security System Scanner

Miss Parnsiri	Ngammaneewat	42010220
Miss Wanna	Pisitpinyo	42010309
Mr. Thana	Hongsuwan	Advisor
Mr. Akkradach	Watcharapupong	Advisor

ABSTRACT

Nowadays, the computer system is important to many work systems especially, with the organization's database. Thus the security of the database is the most important. There are so many ways to prevent the computer system from the intruder such as the Intrusion Detection System, Firewall. If the intruder can attack the system through these walls, the last wall that has left is its own strength.

Therefore, we designed and developed this project called "Security System Scanner" on Linux to help administrators to inspect their own system. The security system scanner is focused on checking all the breach that could almost happen because of the administrator's mistake from configure the system. This can increase the performance of preventing the intruders. When the security system scanner finds the breach, it will report in a log file by use the permission files with Find command to inspect and use Perl language to developed.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานและปริญญานิพนธ์ฉบับนี้เสร็จสมบูรณ์ได้ เนื่องจากได้รับคำแนะนำ สนับสนุน และให้คำปรึกษาเป็นอย่างดีจาก อาจารย์ ธนา หงษ์สุวรรณ และ อาจารย์ อัครเดช วัชระภูพงษ์ อาจารย์ที่ปรึกษาปริญญานิพนธ์ ซึ่งต้องขอขอบพระคุณเป็นอย่างสูง

นอกเหนือจากนี้ต้องขอขอบพระคุณคณะอาจารย์ทุกท่าน ในภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังเป็นอย่างยิ่ง ที่ช่วยประสิทธิ์ประสาทวิชาความรู้ให้แก่คณะผู้จัดทำ อีกทั้งภาควิชาวิศวกรรมคอมพิวเตอร์ ที่ได้เอื้อเฟื้ออุปกรณ์ทรัพยากร สถานที่ และอำนวยความสะดวกต่าง ๆ ในการทำโครงงานนี้ด้วย

ขอขอบคุณ พี่ตั้ม, พี่ก่อ, พี่ต่อ, พี่เชาว์, พี่เดียร์, วงศ์, บอมบ์ และเพื่อนๆ ชาว ISAG ทุกคน ที่คอยให้คำแนะนำ แก้ไขปัญหาอุปสรรคต่าง ๆ และเป็นกำลังใจที่ดีเสมอมา

ขอขอบคุณ ผู้ดูแลระบบคอมพิวเตอร์ภาควิชาวิศวกรรมศาสตร์คอมพิวเตอร์ และสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่อำนวยความสะดวกในการใช้งานเครือข่าย

สุดท้ายนี้ต้องขอขอบพระคุณเป็นอย่างสูงสำหรับบุคคลที่สำคัญที่สุด ที่ทำให้คณะผู้จัดทำมีวันนี้ได้ คือ คุณพ่อบุญเลี้ยง พิสิฐภิญโญ, คุณแม่จันทิมา พิสิฐภิญโญ, คุณพ่อพรชัย งามมณีวัฒน์, คุณแม่ศรัณยา งามมณีวัฒน์ ผู้เป็นบิดามารดาที่เคารพรักยิ่ง ซึ่งท่านได้ให้กำเนิด เลี้ยงดู อบรมสั่งสอน ให้โอกาสในการศึกษาอย่างสูงสุด จึงขอกราบขอบพระคุณมา ณ ที่นี้

พรรณสิริ งามมณีวัฒน์
วรรณา พิสิฐภิญโญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	X
สารบัญภาพ	XI
บทที่ 1 บทนำ	**1**
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของโครงงาน	1
1.3 ขอบเขตของโครงงาน	1
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
1.5 วิธีการดำเนินงาน	2
บทที่ 2 เริ่มต้นการใช้งานลีนุกซ์	**3**
2.1 ความเป็นมาของระบบยูนิกซ์	3
2.2 จากยูนิกซ์จนเป็นลีนุกซ์	3
2.3 เวอร์ชันของลีนุกซ์	3
2.4 ข้อดีและประโยชน์ของลีนุกซ์	4
2.5 สัญลักษณ์ของลีนุกซ์	6
2.6 การนำลีนุกซ์ไปใช้งาน	6
2.7 หัวใจของลีนุกซ์	7
2.8 เชล	7
2.8.1 เชลเริ่มทำงานได้อย่างไร	8
2.8.2 วิวัฒนาการของเชล	8
2.8.3 การเลือกใช้เชล	9
2.8.4 การทำงานของเชล	10
2.8.4.1 การทำงานแบบซิงโครนัส	10
2.8.4.2 การทำงานแบบอะซิงโครนัส	10
บทที่ 3 ความรู้พื้นฐานในการบุกรุกระบบ	**13**
3.1 อาชญากรและอาชญากรรมคอมพิวเตอร์	13
3.2 อาชญากรรมคอมพิวเตอร์	14
3.3 ประเภทของการบุกรุกระบบ	15
3.4 ปัญหาทางด้านความปลอดภัยของระบบ	15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 สาเหตุของการเกิดปัญหาความไม่ปลอดภัยในระบบ 16
3.5.1 ความผิดพลาดของผู้ดูแลระบบ 16
3.5.1.1 ติดตั้งโปรแกรมที่ไม่ได้ใช้งาน หรือเปิดพอร์ตที่ไม่ควรเปิด 16
3.5.1.2 ตั้งค่าต่างๆ ในโปรแกรมมีปัญหา 16
3.5.1.3 รหัสผ่านตั้งไว้ไม่ดีพอ 16
3.5.2 ความผิดพลาดของตัวโปรแกรมเอง 17
3.5.2.1 Vulnerablility 17
3.5.2.2 Buffer Overflow 17
3.5.2.3 Exploit 17
3.5.3 การประทุษร้ายจากบุคคลอื่นๆ 19
3.6 การทำงานหลังจากที่ผู้บุกรุกมีสิทธิสูงสุดในระบบแล้ว 19
3.7 ระบบรักษาความปลอดภัย 22
3.7.1 ความหมายของความปลอดภัยข้อมูล 22
3.7.2 ความปลอดภัยของข้อมูลต้องคำนึงถึงอะไรบ้าง 23
3.8 การทำงานควบคุมเพื่อให้เกิดความปลอดภัยในระบบ 23
3.8.1 การควบคุมดูแลทั้งทางด้านกายภาพและตัวของระบบเอง 23
3.8.2 อ่านและวิเคราะห์ข้อมูลในล็อกไฟล์อย่างสม่ำเสมอ 24
3.8.3 รักษาความปลอดภัยไฟล์และระบบไฟล์ 25
3.8.4 ติดตั้งโปรแกรมและเปิดพอร์ตเฉพาะที่จำเป็นเท่านั้น 25
3.8.5 ตั้งค่าต่างๆ ในระบบและแอพพลิเคชันให้ปลอดภัย 26
3.8.6 การรักษาความปลอดภัยในส่วนของรหัสผ่านและการเข้ารหัส 26
บทที่ 4 ระบบและประเภทของไฟล์ **28**
4.1 ระบบโครงสร้างของไฟล์ 28
4.2 ไอโหนด คืออะไร 29
4.3 ประเภทของไฟล์ 30
4.3.1 ไฟล์ทั่วไป (Regular File) 30
4.3.2 ไดเรกทอรี (Directories) 30
4.3.3 ไฟล์พิเศษ (Character and Block Device File) 31
4.3.4 ไฟล์ที่เชื่อมต่อ (Link File) 32
4.4 การดำเนินการกับไฟล์ 35
4.4.1 การสร้างไฟล์และไดเรกทอรี 35
4.4.2 การลบไฟล์และไดเรกทอรี 35
บทที่ 5 สิทธิการใช้งานไฟล์ **37**
5.1 สิทธิการใช้งานไฟล์ (File Permission) 37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ความหมายของสิทธิต่างๆ 39
5.2.1 ความหมายของสิทธิในการเข้าถึงไฟล์ปกติ 39
5.2.2 ความหมายของสิทธิในการเข้าถึงไดเรกทอรี 40
5.3 ความสัมพันธ์ของ UID กับ username และ GID กับ groupname 41
5.4 ความหมายของ Sticky bit 42
5.5 การกำหนดสิทธิต่างๆ ในไฟล์โดยอัตโนมัติ 42
5.6 การเปลี่ยนแปลงสิทธิต่างๆ ในไฟล์ 43
5.6.1 กำหนดด้วยเลขฐานแปด 44
5.6.2 กำหนดโดยใช้ตัวอักษร(Symbolic setting) 45
5.7 การเปลี่ยนแปลงสิทธิต่างๆ ของไดเรกทอรี 46
บทที่ 6 การใช้คำสั่ง find **47**
6.1 รูปแบบคำสั่ง find 47
6.2 ตัวอย่างการนำคำสั่ง find ไปใช้งาน 48
6.3 การค้นหาไฟล์ในรูปแบบต่างๆ ที่อาจเกิดเป็นช่องโหว่ 51
6.3.1 การตรวจสอบหาUnowned file 51
6.3.2 การตรวจสอบหาไฟล์ลิงค์ 52
6.3.3 การตรวจหา SUID หรือ SGID 53
6.3.4 ไฟล์และไดเรกทอรีที่เป็น World Writable 54
บทที่ 7 ความปลอดภัยของรหัสผ่าน **55**
7.1 เรื่องของรหัสผ่าน 55
7.2 มาตรการการป้องกันการโดนแฮ็กรหัสผ่าน 55
7.3 ตัวอย่างการสร้างรหัสผ่าน 56
7.4 การเข้ารหัสแบบต่างๆ 58
7.4.1 การเข้ารหัสแบบ DES (Data Encryption Standard) 58
7.4.2 การแปลงข้อมูลโดยใช้การคำนวณแบบ MD5 (Message Standard) 58
7.5 หลักในการแคร็กรหัสผ่าน 58
7.5.1 ดิกชันนารีแอ็ตแท็ก 58
7.5.2 การ Brute Force 59
7.6 ตัวอย่างโปรแกรมที่ใช้ตรวจสอบการแคร็กรหัสผ่าน 60
7.6.1 Crack 5.0a 60
7.6.2 John the Ripper 62
7.7 ไฟล์สำคัญต่างๆ ใน /etc ที่เกี่ยวกับรหัสผ่าน 63
7.7.1 ไฟล์ /etc/passwd 63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.7.1.1 /etc/passwd คืออะไร ? 63
7.7.1.2 ความหมายของแต่ละคอลัมน์ในไฟล์ /etc/passwd 64
7.7.2 ไฟล์ /etc/shadow 65
7.7.2.1 /etc/shadow คืออะไร ? 65
7.7.2.2 ความหมายของแต่ละคอลัมน์ในไฟล์ /etc/shadow 66
7.7.3 ไฟล์ /etc/group 66
7.7.3.1 /etc/group คืออะไร? 66
7.7.3.2 ความหมายของแต่ละคอลัมน์ในไฟล์ /etc/group 67
7.7.4 ไฟล์ /etc/gshadow 68
7.7.4.1 /etc/gshadow คือ อะไร ? 68
7.7.4.2 ความหมายของแต่ละคอลัมน์ในไฟล์ /etc/gshadow 68
บทที่ 8 การตั้งค่าในระบบ 69
8.1 inetd คืออะไร 69
8.2 xinetd คืออะไร 69
8.3 ส่วนประกอบของ xinetd 69
8.3.1 /etc/xinetd.conf 70
8.3.2 /etc/xinetd.d/ 70
8.4 กฎเกณฑ์ และค่าของระบบที่สำคัญ 71
8.5 คุณสมบัติเพิ่มเติมของ xinetd กับการปรับเปลี่ยนการเชื่อมต่อ (Redirect Features) 74
8.6 คุณสมบัติเพิ่มเติมของ xinetd กับการกำหนดการเชื่อมต่อ (Bind Feature) 76
8.7 บริการที่จำเป็น 77
8.8 บริการอื่นๆ 78
8.9 ตัวอย่างการคอนฟิกค่าในโปรแกรม sshd 81
8.10 ไฟล์ /etc/services 83
8.11 ไฟล์ /etc/securetty 84
8.12 ไฟล์ lilo 85
8.13 ไฟล์ .bash_history 85
บทที่ 9 ภาษาPerl 87
9.1 ประวัติย่อ 87
9.2 การติดตั้งเพิร์ลบนยูนิกซ์หรือลีนุกซ์ 87
9.3 เริ่มต้นการใช้เพิร์ล 88
9.4 ตัวแปรในภาษาเพิร์ล 89
9.4.1 สเกลลาร์ 89
9.4.2 อาร์เรย์ 89

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.4.3 Associative Array 89
9.5 โอเปอเรเตอร์และการกำหนดค่าให้กับตัวแปรชนิดสเกลลาร์ 90
9.6 โอเปอเรเตอร์และการกำหนดค่า ให้กับตัวแปรชนิดอาร์เรย์ 91
9.7 การดำเนินการ กับข้อความ 92
9.7.1 Regular expression 92
9.7.2 ตัวแปรพิเศษ $_ 92
9.8 ตัวดำเนินการ (Operator) ในภาษาเพิร์ล 93
9.8.1 ตัวดำเนินการ ทางคณิตศาสตร์ (Arithmetic Operators): 93
9.8.2 ตัวดำเนินการ ทางตรรกะ(Logical Operators): 93
9.8.3 ตัวดำเนินการเปรียบเทียบ (Relational Operators): 94
9.9 การตรวจสอบเงื่อนไข 94
9.9.1 โครงสร้างของการตรวจเงื่อนไขแบบ if .. elsif .. else 94
9.10 การทำงานแบบวนรอบ และเงื่อนไขการทำงาน แบบวนรอบ 95
9.10.1 while และ until 95
9.10.2 for และ foreach 96
9.11 การจัดการแฟ้มข้อมูล 96
บทที่ 10 แนวคิดและการออกแบบโปรแกรม 98
10.1 สรุปปัญหาและทางแก้ไข 98
10.1.1 เรื่องของ Unowned files 98
10.1.2 เรื่องของ Symbolic link 98
10.1.3 เรื่องของ SUID/SGID 98
10.1.4 เรื่องของ World Writable 99
10.1.5 เรื่องของไฟล์ .bash_history 101
10.1.6 เรื่องของ Xinetd 101
10.1.7 เรื่องของเวอร์ชัน 101
10.1.8 เรื่องของไฟล์ Lilo 102
10.1.9 เรื่องของไฟล์ Securetty 102
10.1.10 เรื่องของไฟล์ .rhost 102
10.1.11 เรื่องของรหัสผ่าน 103
10.2 ออกแบบโปรแกรม 104
10.3 ฟังก์ชันการทำงานของโปรแกรมตรวจสอบช่องโหว่ความปลอดภัยระบบ 106
10.4 ข้อจำกัดของระบบ 108
บทที่ 11 สรุป 109
11.1 สรุป 109

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11.2 แนวทางการปรับปรุงในอนาคต 109

ภาคผนวก ก - การใช้โปรแกรม crontab **111**

ภาคผนวก ข - CVE **113**

บรรณานุกรม **117**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ 5-1 : ความหมายของฟิลด์เมื่อใช้คำสั่ง ls –l 37
ตารางที่ 5-2 : แสดงคำสั่งเวลาที่มีการบันทึกในระบบ 38
ตารางที่ 5-3 : แสดงการแบ่งกลุ่มของเพอร์มิสชัน 38
ตารางที่ 5-4 : คำอธิบายเพิ่มเติมของสิทธิในการใช้งานของกลุ่มต่างๆ 39
ตารางที่ 5-5 : ตัวอย่างของสิทธิในการใช้งาน 39
ตารางที่ 5-6 : ตารางแสดงความหมายของสิทธิต่างๆ ของไฟล์ 40
ตารางที่ 5-7 : ตารางแสดงความหมายของสิทธิต่างๆ ของไดเรกทอรี 40
ตารางที่ 5-8 : แสดงสิทธิขั้นต่ำในการใช้งานในแบบต่างๆ 40
ตารางที่ 5-9 : แสดง รายละเอียดในไฟล์ /etc/passwd 41
ตารางที่ 5-10 : ตารางแสดงรายละเอียดของไฟล์ /etc/groups 41
ตารางที่ 5-11 : แสดงค่าดีฟอลต์ของสิทธิการใช้งานของระบบ 43
ตารางที่ 5-12 : แสดงการ XOR เมื่อใช้ umask เป็น 0022 กับไฟล์ที่สร้างใหม่ 43
ตารางที่ 5-13 : แสดงการ XOR เมื่อใช้ umask เป็น 0022 กับโปรแกรมใหม่ 43
ตารางที่ 5-14 : แสดงตารางช่วยจำ 44
ตารางที่ 5-15 : แสดงการลงตาราง 44
ตารางที่ 5-16 : แสดงการลงตาราง 45
ตารางที่ 5-17 : ตารางแสดงการเปลี่ยนแปลงสิทธิโดยใช้ตัวอักษร 45
ตารางที่ 7-1 : ฟรีแวร์ที่ช่วยในการปกป้องรหัสผ่านให้รอดพ้นจากการคาดเดารหัส 57
ตารางที่ 7-2 : รูปแบบของไฟล์ /etc/passwd 65
ตารางที่ 7-3 : รูปแบบของไฟล์ /etc/shadow 66
ตารางที่ 7-4 : รูปแบบของไฟล์ /etc/group 67
ตารางที่ 7-5 : รูปแบบของไฟล์ /etc/gshadow 68
ตารางที่ 8-1 : ตัวกระทำของไฟล์ระบบในโปรแกรม xinetd 71
ตารางที่ 8-2 : แสดงค่าคุณสมบัติที่น่าสนใจของระบบในโปรแกรม xinetd 73
ตารางที่ 8-3 : แสดงความหมายของไฟล์ ftp ข้างต้น 74
ตารางที่ 8-4 : ความหมายของการคอนฟิกไฟล์ /etc/ssh/sshd_config 83
ตารางที่ 9-1 : การกำหนดค่าและ การดำเนินการกับ ตัวแปรชนิดสเกลลาร์ 90
ตารางที่ 9-2 : การดำเนินการ กับตัวแปรชนิดสเกลลาร์กับข้อมูล ชนิดค่าตัวเลข 90
ตารางที่ 9-3 : การดำเนินการ กับตัวแปรชนิดสเกลลาร์กับข้อมูลชนิด ข้อความ 91
ตารางที่ 9-4 : แสดงตัวดำเนินการทางคณิตศาสตร์ 93
ตารางที่ 9-5 : แสดงตัวดำเนินการทางตรรกะ 93
ตารางที่ 9-6 : แสดงตัวดำเนินการเปรียบเทียบ 94
ตารางที่ 11-1 : ตารางเปรียบเทียบระหว่าง Vetescan Local และ Security System Scanner 110

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

รูปที่ 2-1 : การทำงานแบบมัลติทาส์กิง 4
รูปที่ 2-2 : การทำงานแบบมัลติยูสเซอร์ 5
รูปที่ 2-3 : สัญลักษณ์ของลีนุกซ์ค่ายต่างๆ 6
รูปที่ 2-4 : โครงสร้างของระบบลีนุกซ์ 7
รูปที่ 2-5 : แสดงการเปลี่ยนเชลใหม่ 10
รูปที่ 2-6 : ตัวอย่างการทำงานฉากหน้า 10
รูปที่ 2-7 : ตัวอย่างการทำงานฉากหลัง 11
รูปที่ 2-8 : ระบบหน้าต่าง 12
รูปที่ 3-1 : การเกิด buffer overflow และ Exploit 18
รูปที่ 3-2 : แสดงผลการ finger 20
รูปที่ 3-3 : แสดง ps –def | grep ksh 20
รูปที่ 3-4 : แสดงคำสั่ง ifconfig 21
รูปที่ 3-5 : แสดงผล lsmod 22
รูปที่ 3-6 : ข้อมูลในไฟล์ access_log ของ Apache 24
รูปที่ 3-7 : ข้อมูลในไฟล์ mail.log ของ sendmail 25
รูปที่ 3-8 : ข้อมูลภายในไฟล์ /etc/inetd.conf 26
รูปที่ 3-9 : การสร้างเงื่อนไขไฟล์ hosts.deny หรือ hosts.allow 27
รูปที่ 4-1 : แสดงโครงสร้างของระบบปฏิบัติการยูนิกซ์เป็นรูปต้นไม้กลับหัว 28
รูปที่ 4-2 : แสดงรายละเอียดของไฟล์ทั่วไป 30
รูปที่ 4-3 : แสดงรายละเอียดของไดเรกทอรี 31
รูปที่ 4-4 : แสดงรายละเอียดของไฟล์พิเศษทั้ง 2 ชนิด 31
รูปที่ 4-5 : แสดงรายละเอียดของไฟล์พิเศษในแบบต่างๆ เช่น ฟลอปปี้ดิสก์ ดิสก์ พรินเตอร์ ตามลำดับ32
รูปที่ 4-6 : ลักษณะของไฟล์เชื่อมต่อทางกายภาพ 33
รูปที่ 4-7 : แสดงการสร้างไฟล์แบบ hard link 33
รูปที่ 4-8 : ลักษณะของไฟล์เชื่อมต่อโดยนัย 34
รูปที่ 4-9 : แสดงการสร้างไฟล์เชื่อมต่อโดยนัยชื่อ sym2 ชี้ไปยัง symfile 34
รูปที่ 4-10 : แสดงการสร้างไฟล์เชื่อมต่อโดยนัยชื่อ sym3 ชี้ไปยัง sym2 34
รูปที่ 4-11 : แสดงการลบไฟล์เชื่อมต่อโดยนัยที่เป็นไฟล์ต้นฉบับที่ชื่อ symfile 34
รูปที่ 4-12 : การสร้างไฟล์และไดเรกทอรีด้วยวิธีต่างๆ 35
รูปที่ 4-13 : การลบไฟล์ fileone ออกจากระบบ 36
รูปที่ 4-14 : การลบไดเรกทอรี directory1 ออกจากระบบ 36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5-1 : แสดงรายละเอียดของไฟล์ test 37
รูปที่ 5-2 : ตัวอย่างการใช้คำสั่ง chmod ด้วยตัวอักษร 45
รูปที่ 5-3 : ตัวอย่างการใช้คำสั่งกำหนดระดับการเข้าถึงไดเรกทอรี 46
รูปที่ 6-1 : รูปแบบของคำสั่ง find 47
รูปที่ 6-2 : แสดงการใช้คำสั่ง find 48
รูปที่ 6-3 : แสดงการหาตำแหน่งของไฟล์ชื่อ test โดยการใช้คำสั่ง find 49
รูปที่ 6-4 : แสดงรายละเอียดของไฟล์โดยใช้คำสั่ง find ร่วมกับคำสั่ง ls 49
รูปที่ 6-5 : แสดงการใช้คำสั่ง find โดยไม่กำหนดอะไรเลย 49
รูปที่ 6-6 : แสดงการค้นหาไฟล์ที่มีรูตเป็นเจ้าของ 50
รูปที่ 6-7 : แสดงการค้นหาไฟล์ที่มีขนาดมากกว่า 5000 50
รูปที่ 6-8 : แสดงการหาไฟล์ลิงก์พร้อมกับแสดงรายละเอียดของไฟล์ด้วย 50
รูปที่ 6-9 : แสดงการเพิ่มผู้ใช้งานคนใหม่ที่มี uid เป็นของคนเดิม 52
รูปที่ 6-10 : คำสั่ง find หา Unowned file 52
รูปที่ 6-11 : คำสั่ง find เพื่อหา Symbolic link 53
รูปที่ 6-12 : การทำงานของโปรแกรม SUID 53
รูปที่ 6-13 : คำสั่ง find หา SUID หรือ SGID 54
รูปที่ 6-14 : คำสั่ง find หา world writable 54
รูปที่ 7-1 : การแคร็กรหัสผ่านทำงานอย่างไร 59
รูปที่ 7-2 : สำเนาไฟล์ /etc/shadow 61
รูปที่ 7-3 : แสดงการใช้คำสั่งในการรันโปรแกรม Crack 61
รูปที่ 7-4 : ผลลัพธ์จะแสดงหลังจากรันโปรแกรม Reporter 62
รูปที่ 7-5 : การรัน john 63
รูปที่ 7-6 : ตัวอย่างไฟล์ /etc/passwd 64
รูปที่ 7-7 : ตัวอย่างไฟล์ /etc/shadow 65
รูปที่ 7-8 : ตัวอย่างไฟล์ /etc/group 67
รูปที่ 7-9 : ตัวอย่างไฟล์ /etc/gshadow 68
รูปที่ 8-1 : แสดงขั้นตอนการทำงานของโปรแกรม INETD และ TCPD เมื่อมีการเรียกใช้งาน ftpจากเครือข่าย 69
รูปที่ 8-2 : โครงสร้างไฟล์ /etc/xinetd.conf 70
รูปที่ 8-3 : ไฟล์ /etc/xinetd.conf 70
รูปที่ 8-4 : โครงสร้างไฟล์ /etc/xinetd.d ของแต่ละไฟล์ 70
รูปที่ 8-5 : ไฟล์ของโปรแกรม ftp ที่อยู่ใน /etc/xinetd.d/ 74
รูปที่ 8-6 : แสดง การร้องขอ ssh บน xinetd ที่อาศัยคุณสมบัติ redirect 75
รูปที่ 8-7 : ไฟล์ /etc/xinetd.d/ssh ตามรูปที่ 8-6 75

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 8-8 : แสดงการเปิดให้บริการ FTP ทั้งภายใน และภายนอกองค์กร ที่ใช้คุณสมบัติ bind 76
รูปที่ 8-9 : ไฟล์ /etc/xinetd.d/ftp ตามรูปที่ 8-8 77
รูปที่ 8-10 : ใช้ telnet ในการทำ remote access ไปยังเซิร์ฟเวอร์และผู้โจมตีสามารถดักข้อมูลได้ 78
รูปที่ 8-11 : ใช้ ssh ให้สังเกตว่ารหัสผ่านที่ส่งจากไคลเอ็นต์นั้นจะถูกเข้ารหัสไว้ 78
รูปที่ 8-12 : การคอนฟิกค่าในไฟล์ /etc/ssh/sshd_config 81
รูปที่ 8-13 : ตัวอย่างรายละเอียดของไฟล์ /etc/services 83
รูปที่ 8-14 : ตัวอย่างรายละเอียดของไฟล์ /etc/securetty 84
รูปที่ 8-15 : แสดงผลลัพธ์จากการใช้คำสั่ง w 84
รูปที่ 8-16 : แสดงการตั้งค่าในไฟล์ lilo.conf 85
รูปที่ 10-1 : แสดงโฟลว์ชาร์ตของตัวโปรแกรม 105

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

คงจะไม่มีใครปฏิเสธว่า ในปัจจุบันนี้คอมพิวเตอร์มีความสำคัญกับชีวิตประจำวันมากขึ้น ไม่ว่าจะเป็นธุรกิจทางด้านไหนก็ตามล้วนมีความเกี่ยวข้องทั้งสิ้น เนื่องจากความต้องการในการนำคอมพิวเตอร์ไปใช้ในระบบต่างๆ มีมากขึ้น จึงส่งผลให้มีผู้ไม่ประสงค์ดีที่ต้องการจะบุกรุกทำลายระบบคอมพิวเตอร์มีมากขึ้นด้วย

ในเมื่อการบุกรุกทำลายระบบมีมากขึ้น รูปแบบและวิธีการในการเข้าถึงระบบมากขึ้นด้วย ดังนั้นแนวทางในการป้องกันและลดความเสี่ยงจากการถูกบุกรุกนั้นจึงเป็นเรื่องที่สำคัญมาก ในเมื่อเป็นเช่นนี้แล้วเราจึงควรจะป้องกันโดยการหาจุดบกพร่องหรือช่องโหว่ที่อาจเกิดเป็นจุดอ่อนที่ทำให้ผู้บุกรุกสามารถบุกเข้ามาทำลายระบบของเราได้

ปริญญานิพนธ์นี้จึงมุ่งเน้นการศึกษาจุดอ่อนของระบบที่เป็นเรื่องพื้นฐานที่ผู้ดูแลระบบ (administrator) อาจมองข้ามไปได้ รวมถึงแนวทางการแก้ไขที่ควรจะทำ เพื่อพัฒนาโปรแกรมบนระบบปฏิบัติการลีนุกซ์ให้สามารถรายงานผลลัพธ์การเกิดจุดบกพร่องขึ้นในระบบได้อย่างมีประสิทธิภาพ

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อศึกษาช่องโหว่และการตรวจสอบความปลอดภัยภายในระบบคอมพิวเตอร์
2. เพื่อศึกษาและใช้งานระบบปฏิบัติการลีนุกซ์
3. เพื่อฝึกการเขียนโปรแกรมบนแพลตฟอร์มลีนุกซ์
4. เพื่อสร้างโปรแกรมตรวจสอบช่องโหว่

1.3 ขอบเขตของโครงงาน

โปรแกรมตรวจสอบช่องโหว่ความปลอดภัยระบบเป็นเครื่องมือที่ช่วยผู้ดูแลระบบ ทำการตรวจสอบไฟล์ข้อมูลบนระบบ ว่ามีไฟล์ใดบ้างที่มีความน่าจะเป็นหรือมีโอกาสที่จะเป็นช่องโหว่ให้ผู้อื่นเข้ามากระทำการอันมิชอบได้บ้าง ซึ่งช่องโหว่เหล่านั้นอาจจะเกิดจากความไม่รู้ของผู้ใช้งานระบบ (user) หรืออาจเกิดจากผู้ไม่ประสงค์ดีต่อระบบมากระทำไว้ โดยโปรแกรมจะทำงานก็ต่อเมื่อผู้ดูแลระบบได้สั่งให้สแกนโดยตรง (unreal time) หรือสั่งผ่านโปรแกรม cron ก็ได้ และจะรายงานผลว่าลักษณะหรือไฟล์ข้อมูลใดที่มีโอกาสเสี่ยงทำให้เกิดช่องโหว่กับระบบได้บ้าง เพื่อเป็นข้อมูลสำหรับประกอบการตัดสินใจให้กับผู้ดูแลระบบที่จะดำเนินการแก้ไขกับลักษณะของไฟล์ข้อมูลนั้นต่อไป

ซึ่งช่องโหว่ที่เราทำการตรวจสอบได้แบ่งเป็นประเภทได้ 4 ประเภทดังนี้

1. ไฟล์ (ประเภทของไฟล์ หรือ สิทธิของการใช้งานไฟล์)
   - unowned files

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- symbolic link
- suid, sgid
- world writable files

2. เซอร์วิส
   - xinetd
3. การตั้งค่า
   - เวอร์ชันของเซอร์วิสที่เปิด
   - ไฟล์ .rhost
   - ไฟล์ securetty
   - ไฟล์ lilo.conf
4. รหัสผ่าน
   - crack password

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ศึกษาช่องโหว่ต่างๆ ภายในระบบ
2. ได้ศึกษาการใช้งานระบบปฏิบัติการลีนุกซ์
3. ได้โปรแกรมที่เป็นเครื่องมือช่วยของผู้ดูแลระบบ

1.5 วิธีการดำเนินงาน

- ศึกษารายละเอียดการใช้งานลีนุกซ์ทั้งหมด
- ศึกษาเรื่องต่างๆ เช่น สิทธิการใช้งานต่างๆ (permission), คำสั่ง find และคำสั่งอื่นๆ, บริการที่มีอยู่ในระบบ, เรื่องของการใช้รหัสผ่าน (password)
- ออกแบบโครงสร้างของโปรแกรม
- ศึกษาการเขียนโปรแกรมติดต่อกับดาต้าเบส (Database)
- พัฒนาโปรแกรม
- ทดสอบและปรับปรุงโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
เริ่มต้นการใช้งานลีนุกซ์

2.1 ความเป็นมาของระบบยูนิกซ์

เมื่อประมาณช่วงปี 1970 นั้น ระบบปฏิบัติการยูนิกซ์ได้ถือกำเนิดขึ้น ซึ่งเป็นระบบปฏิบัติการที่นิยมมากในสมัยนั้น ได้มีการนำไปใช้ในหลายสายอาชีพไม่ว่าจะเป็นในสถานศึกษาต่างๆ หรือสถานที่ทำงาน ถึงแม้ว่าจะยุ่งยากในการใช้งานก็ตาม

เป็นเพราะคำสั่งที่ค่อนข้างจะมีมาก ซึ่งตรงกันข้ามกับระบบปฏิบัติการวินโดวส์ เนื่องจากจะมีไอคอนให้เราคลิ้กเพื่อใช้งานได้เลย แต่ยูนิกซ์ก็จะมีข้อดีตรงที่ประสิทธิภาพในการทำงานของมันจะสูงกว่าวินโดวส์มากเมื่อมีระดับของฮาร์ดแวร์เท่ากัน

2.2 จากยูนิกซ์จนเป็นลีนุกซ์

เมื่อยูนิกซ์ได้มีการพัฒนามาเรื่อยๆ จนถึงยุคหนึ่ง คราวนี้ก็ถึงคราวของลีนุกซ์บ้าง ลีนุกซ์ได้ถูกสร้างโดยนำต้นแบบมาจากระบบปฏิบัติการยูนิกซ์ โดยนายลีนัส โทรวัลด์ส (Linus Trovalds) ซึ่งขณะนั้นเป็นนักศึกษาภาควิชาวิทยาการคอมพิวเตอร์ (Computer Science) ในมหาวิทยาลัยเฮลซิงกิ แห่งประเทศฟินแลนด์ โดยในระหว่างที่กำลังศึกษาอยู่นั้นเขาได้ใช้ระบบมินิกซ์ (Minix) ที่เป็นระบบปฏิบัติการยูนิกซ์บนพีซี ลีนัสรู้สึกว่าระบบมินิกซ์นี้ยังมีความสามารถไม่ดีพอ จึงได้ร่วมมือกับนักพัฒนาโปรแกรมอื่นๆ คิดค้นระบบใหม่ขึ้น เพื่อให้มีประสิทธิภาพมากกว่าระบบมินิกซ์ที่ใช้อยู่ ซึ่งก็คือ ลีนุกซ์ (Linux) นั่นเอง

ลีนุกซ์เป็นระบบปฏิบัติการที่เปิดเผยซอร์สโค้ด (open source code) นักพัฒนาโปรแกรมอื่นๆ จึงสามารถที่จะนำซอร์สโค้ดนี้มาพัฒนาต่อได้ ทำให้ลีนุกซ์มีพัฒนาการที่รวดเร็วมาก รวมทั้งมีแอพพลิเคชันที่ถูกสร้างขึ้นมาเพื่อใช้งานลีนุกซ์มากด้วย เพราะเหตุนี้จึงไม่แปลกที่ลีนุกซ์จะเป็นระบบปฏิบัติการยูนิกซ์ที่เป็นที่นิยมมากในสมัยนี้

ลีนุกซ์มีการแตกแขนงต่างๆ ออกมาตามผู้ที่ทำการพัฒนา ซึ่งที่คนไทยรู้จักกันดีก็จะมีจากหลายค่ายด้วยกัน เช่น Slackware, Redhat และ Kaiwal เป็นต้น

2.3 เวอร์ชันของลีนุกซ์

เนื่องจากมีเวนเดอร์ (vendor) อยู่หลายบริษัทที่ให้บริการเกี่ยวกับลีนุกซ์ ดังนั้นจึงมีความสับสนอยู่พอสมควรในการระบุถึงลีนุกซ์ โดยทั่วไปแล้ว release จะหมายถึง เวนเดอร์ที่เป็นผู้ให้บริการลีนุกซ์นั้นๆ เช่น Slackware, Redhat โดยทั่วไปแล้วความแตกต่างกันในเรื่องของโพรเซสในการทำงานจะมีไม่มากนัก แต่จะแตกต่างกันในส่วนของซอฟต์แวร์สนับสนุน เช่น ยูทิลิตี้ หรือ ไดรฟ์เวอร์ของแต่ละเวนเดอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนคำว่าเวอร์ชัน (version) จะหมายถึง รุ่นของ release นั้นๆ เช่น Slackware 2.3, Slackware 2.4 เป็นต้น ดังนั้นเพื่อไม่ให้เกิดความสับสน เมื่ออ้างถึงลีนุกซ์จึงใช้ทั้งสองอย่างพร้อมกัน เมื่อมีการอัพเดตเวอร์ชันใหม่ขึ้นมา เช่น จาก 2.3 เป็น 2.4 จะเห็นได้ว่าจะแตกต่างกันค่อนข้างน้อย การใช้งานโดยทั่วไปจะไม่มีผลมากนัก แต่ถ้ามีเวอร์ชันแตกต่างกันมาก เช่น จาก 1.x เป็น 2.x ก็แสดงว่ามีการเปลี่ยนแปลงค่อนข้างมาก เช่น อาจจะมีการเปลี่ยนแปลงการทำงานของเคอร์เนลหรือมีการเพิ่มไดร์ฟเวอร์ก็ควรจะทำการอัพเดต

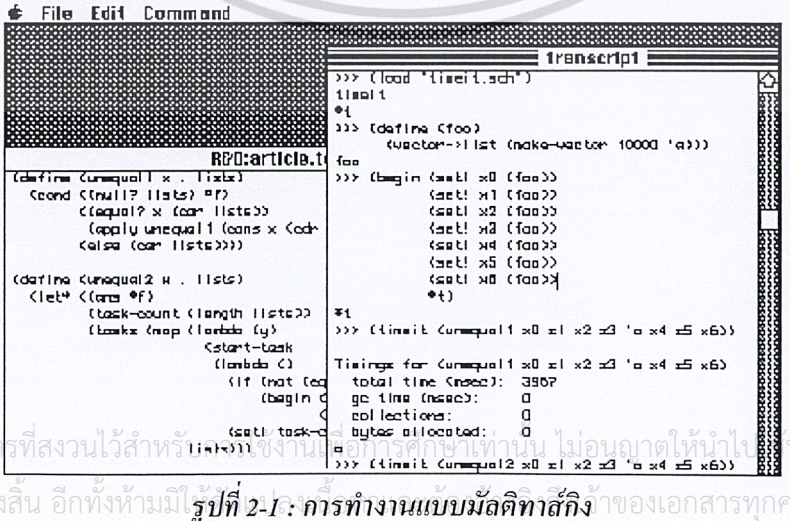
โดยทั่วไปแล้ว เคอร์เนลประกอบด้วยตัวเลข 3 ชุด x.x.x ตัวเลขตัวแรกเป็น เวอร์ชันหลัก ตัวที่สองเป็นเวอร์ชันรอง ตัวที่ 3 เป็นการปรับปรุงครั้งที่ของเวอร์ชันนั้น เวอร์ชันหลักจะมีการเปลี่ยนแปลงก็ต่อเมื่อมีการปรับปรุงเปลี่ยนแปลง หรือได้รับการพัฒนาแตกต่างไปจากเดิมไปอย่างมาก ตัวเลขชุดที่ 2 ตัวเลขชุดนี้บอกว่าเคอร์เนลอยู่ระหว่างพัฒนา ให้ดูว่าถ้าเคอร์เนลที่เสถียรจะเป็นเลขคู่ ถ้าตัวเลขไม่เสถียรจะเป็นเลขคี่ เช่น 2.2.x จะเป็นเคอร์เนลที่เสถียร ส่วน 2.3.x นั้นเป็นเคอร์เนลที่ไม่เสถียร ส่วนตัวเลขชุดที่ 3 บอกครั้งที่ของการปรับปรุงเคอร์เนลในเวอร์ชันนั้นๆ

2.4 ข้อดีและประโยชน์ของลีนุกซ์

การที่ลีนุกซ์ได้รับความนิยมและมีผู้สนใจนำไปใช้งานเพิ่มมากขึ้น ก็เนื่องมาจากประสิทธิภาพการทำงาน ข้อดีและประโยชน์ของลีนุกซ์มีอยู่มากมาย โดยสามารถสรุปเป็นหัวข้อได้ดังต่อไปนี้

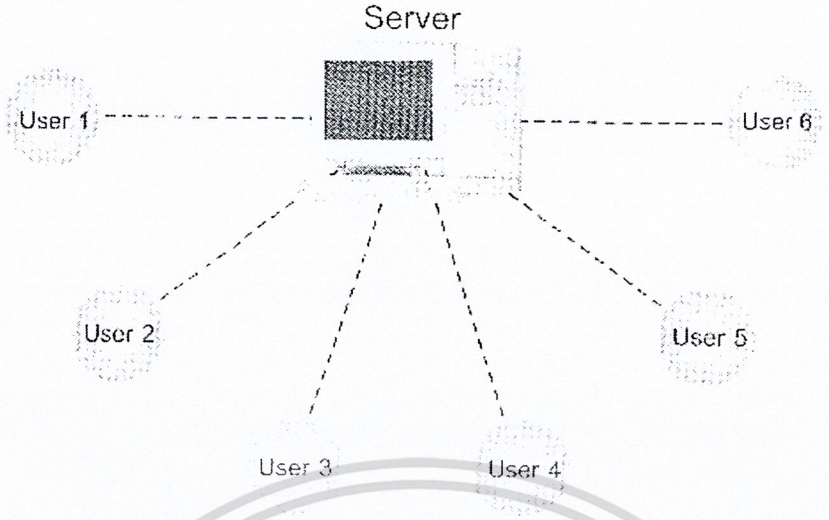
✓ ลีนุกซ์ถอดแบบมาจากยูนิกซ์

ยูนิกซ์เป็นระบบปฏิบัติการที่เก่าแก่และขึ้นชื่อมานานในเรื่องประสิทธิภาพการทำงาน ลีนุกซ์เป็นการถอดแบบมาจากยูนิกซ์ ดังนั้นคุณสมบัติของยูนิกซ์ทั้งเรื่องของระบบความปลอดภัย ความสามารถในการทำงานพร้อมกันหลายงาน โดยที่งานเหล่านั้นทำงานเป็นอิสระกัน(Multi Tasking) ใช้งานได้พร้อมกันหลายคน โดยผู้ใช้แต่ละคนทำงานได้เป็นอิสระจากกัน(Multi User) ทำงานได้ทั้งแบบเท็กซ์โหมด(Text Mode) และกราฟิกโหมด(Graphic Mode) ทำงานได้ทั้งแบบหน้าเครื่อง(Console) หรือจากเครื่องอื่น(Remote) ได้ประสิทธิภาพในการใช้งานเป็นเซิร์ฟเวอร์ รวมทั้งคุณสมบัติอื่นๆ อีกมากมาย จึงได้รับการถ่ายทอดมาที่ลีนุกซ์ด้วย



รูปที่ 2-1 : การทำงานแบบมัลติทาส์กิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-2 : การทำงานแบบมัลติยูสเซอร์

✓ ใช้งานได้โดยไม่ต้องเสียค่าใช้จ่าย

ลีนุกซ์และโปรแกรมที่พัฒนาขึ้นเพื่อใช้งานบนลีนุกซ์จะอยู่ภายใต้ลิขสิทธิ์ที่เรียกกันว่า General Public License (GPL) ซึ่งหมายความว่า เราสามารถนำลีนุกซ์มาใช้งานได้ฟรี นำไปใช้ได้ไม่จำกัดจำนวนครั้ง ปรับปรุงแก้ไขได้ตามต้องการ โดยซอร์สโค้ดที่ได้ทำการแก้ไขจะต้องเผยแพร่ให้ผู้อื่นใช้ได้ฟรีเหมือนกับต้นฉบับ

✓ ความปลอดภัยในการทำงาน

ลีนุกซ์เป็นระบบที่มีความปลอดภัยในการใช้งานสูง ก่อนที่จะเข้าไปใช้งานทุกครั้งจะมีการตรวจสอบโดยผู้ใช้ต้องทำการป้อนชื่อและรหัสผ่าน เพื่อแสดงสิทธิในการใช้งาน (หรือที่เรียกว่าการ Log in) ให้ถูกต้องจึงจะเข้าใช้งานลีนุกซ์ได้

✓ เสถียรภาพในการทำงาน

ลีนุกซ์มีเสถียรภาพในการทำงานสูง ปัญหาระบบล่มในระหว่างทำงานจะไม่ค่อยมีให้พบ โดยความพิเศษของลีนุกซ์ Redhat อยู่ที่การตรวจสอบความสัมพันธ์ของโปรแกรมในการทำงาน เช่น ถ้าเราติดตั้งโปรแกรม A มันจะทำการตรวจสอบว่าโปรแกรม A มีการเรียกใช้งานโปรแกรมอื่นทำงานร่วมด้วยหรือไม่ ถ้าต้องใช้ redhat จะให้เลือกติดตั้งโปรแกรมนั้นลงไปด้วย นอกจากนี้ถ้าทำการติดตั้งหรือลบโปรแกรมออกจากระบบ เราไม่ต้องบูตเครื่องใหม่สามารถทำงานต่อไปได้ทันที

✓ สนับสนุนฮาร์ดแวร์ทั้งใหม่และเก่า

เทคโนโลยีของอุปกรณ์ฮาร์ดแวร์มีการเปลี่ยนแปลงไปอย่างรวดเร็ว ระบบปฏิบัติการโดยส่วนใหญ่มักจะออกมาเพื่อรองรับประสิทธิภาพการทำงานของฮาร์ดแวร์ที่พัฒนาขึ้น จนทำให้บางครั้งต้องมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลง เนื้อหาและต้องอ้างอิงถึงเอกสารนี้ทุกครั้งที่มีการนำไปใช้

การอัพเกรดเครื่องตาม แต่สำหรับลีนุกซ์จะยังคงสนับสนุนฮาร์ดแวร์เก่าให้สามารถใช้งานได้ โดยจะเพิ่มในส่วนของการสนับสนุนฮาร์ดแวร์ตัวใหม่ลงไปเท่านั้น ทำให้เราไม่จำเป็นต้องเปลี่ยนฮาร์ดแวร์ ซึ่งช่วยประหยัดค่าใช้จ่ายลงไปได้มาก

✓ ลีนุกซ์กับระบบเครือข่าย

จุดเด่นอีกเรื่องที่สำคัญของลีนุกซ์ คือ การใช้งานกับระบบเครือข่าย ลีนุกซ์สามารถใช้เป็นเซิร์ฟเวอร์ (server) ในระบบเครือข่ายร่วมกับเครื่องไคลเอ็นต์ (Client) ซึ่งติดตั้งระบบปฏิบัติการอื่นได้ นอกจากนี้ลีนุกซ์ยังสนับสนุนโพรโตคอลในการทำงานกับระบบเครือข่ายมากมาย อย่างเช่น TCP/IP, DNS, FTP

2.5 สัญลักษณ์ของลีนุกซ์

สิ่งหนึ่งที่เป็นสัญลักษณ์หรือมอสคอทของลีนุกซ์ คือ นกเพนกวินที่ชื่อว่า "Tux" ซึ่งลีนุกซ์เป็นผู้เลือกด้วยตัวเอง ปัจจุบันนี้สัญลักษณ์ของเพนกวินนั้นมีอยู่ในรูปแบบและอริยาบถต่างๆ มากมายตามที่ผู้ใช้ลีนุกซ์ทั่วโลกช่วยกันออกแบบ

รูปที่ 2-3 : สัญลักษณ์ของลีนุกซ์ค่ายต่างๆ

2.6 การนำลีนุกซ์ไปใช้งาน

ปัจจุบันมีการนำระบบลีนุกซ์ไปประยุกต์เป็นระบบปฏิบัติการสำหรับด้านต่างๆ เช่น งานด้านการคำนวณทางวิทยาศาสตร์ ใช้เป็นสถานีบริการอินเทอร์เน็ต อินทราเน็ต การประกอบพาณิชย์อิเล็กทรอนิกส์ หรือใช้ในการเรียน การสอนและการทำวิจัย หรือแม้แต่ใช้ในการสร้างกราฟิกส์ในอุตสาหกรรมภาพยนตร์ นอกจากนี้ยังใช้เป็นระบบปฏิบัติการสำหรับพัฒนาโปรแกรมต่างๆ เนื่องจากมีเครื่องมือมากมาย เช่น โปรแกรมภาษาซี (C) ซีพลัสพลัส (C++) ปาสคาล (Pascal) ฟอร์แทรน (Fortran) ลิสป์ (Lisp) โปรล็อก (Prolog) เอดา (ADA) มีภาษาสคริปต์ เช่น เชล (Shell) บาชเชล (Bash Shell) ซีเชล (C Shell) คอร์นเชล (Korn Shell) เพิร์ล (Perl) พายตัน (Python) TCL/TK นอกจากนี้ยังมีโปรแกรมประยุกต์ในสาขาต่างๆ อีกมากมาย ทั้งที่เป็นซอฟต์แวร์ที่แจกฟรีและซอฟต์แวร์ทางการค้า โดยข้อมูลของโปรแกรมเหล่านี้ได้รวบรวมไว้ที่ Linux Software Map (LSM) http://www.linux.org/apps/lsm.html

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 หัวใจของลีนุกซ์

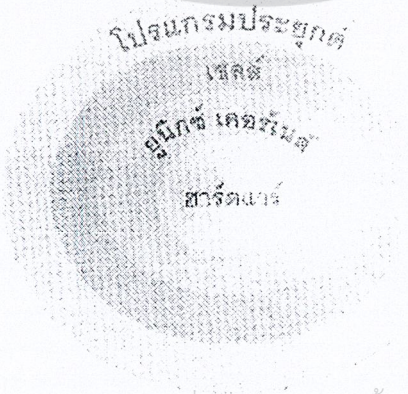
เคอร์เนลของลีนุกซ์เป็นส่วนสำคัญในการทำงานของลีนุกซ์ มีหน้าที่สำคัญคือ การจัดการกับโพรเซสและการจัดการกับไฟล์ รวมทั้งการจัดการกับอุปกรณ์ต่างๆ เป็นส่วนที่จะทำหน้าที่ในการบริหารรวมทั้งควบคุมระบบทั้งหมด เริ่มแรกได้ถูกพัฒนาขึ้นมาสำหรับ CPU Intel 80386 protect mode ซึ่งถูกออกแบบมาให้ทำงานบนระบบมัลติทาส์กิงเป็นครั้งแรก และลีนุกซ์สามารถนำคำสั่งของ CPU ตัวนี้มาใช้งานได้เป็นอย่างดี (CPU Instructions set)

นอกจากนี้แล้วยังมีลักษณะพิเศษในการจัดการกับหน่วยความจำและไฟล์อีก คือ

- ไฟล์เอ็กซิคิวต์เทเบิล (executable file) สามารถแชร์พื้นที่ในหน่วยความจำได้ลักษณะในการทำงานแบบนี้เรียกว่า copy on write page ซึ่งเป็นการเพิ่มประสิทธิภาพในการใช้งานหน่วยความจำ
- สามารถทำงานในระบบ demand paging ซึ่งเป็นการนำเอาโปรแกรมที่ใช้งานบ่อยมาเก็บไว้ในหน่วยความจำ และถ้ามีการใช้งานระบบมากขึ้น ความจุดิสก์แคชจะถูกปรับโดยอัตโนมัติ
- เมื่อต้องการที่จะใช้งานหน่วยความจำเพิ่มขึ้น ลีนุกซ์สามารถที่จะนำพื้นที่บนดิสก์มาทำหน้าที่เสมือนหน่วยความจำได้ (swap space) แต่ก็มีข้อเสียอันเนื่องมาจากการแอ็กเซสข้อมูลที่ช้าของดิสก์เมื่อเทียบกับหน่วยความจำจริง
- นอกจากนี้แล้ว ถ้ามีแอพพลิเคชันใดใช้งานไลบรารีเหมือนกัน ลีนุกซ์สามารถที่จะจัดการเพื่อให้แอพพลิเคชันเหล่านั้นใช้งานไลบรารีร่วมกัน เป็นการลดขนาดแอพพลิเคชันลง (dynamic shared libraries)
- ลีนุกซ์ซัพพอร์ตไฟล์หลายชนิด เช่น DOS, OS-2 สำหรับไฟล์ของลีนุกซ์เองมีชื่อว่า ext2fs

2.8 เชล

โครงสร้างของระบบลีนุกซ์เป็นชั้นวงแหวนหลักๆ 4 ระดับด้วยกัน ซึ่งแต่ละระดับก็จะมีความสามารถและหน้าที่ในการทำงานแตกต่างกัน สามารถแสดงเป็นรูปภาพได้ดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2-4 : โครงสร้างของระบบลีนุกซ์

- **ฮาร์ดแวร์** (hardware) ส่วนนี้เปรียบเสมือนหัวใจหลักของคอมพิวเตอร์ เป็นส่วนที่เราพอจะมองเห็นเป็นรูปเป็นร่างมากที่สุด เช่น จอภาพ เมาท์ คีย์บอร์ด ฮาร์ดดิสก์ ซีพียู เป็นต้น
- **ยูนิกซ์ เคอร์เนล** เป็นส่วนที่สำคัญมากส่วนหนึ่งของระบบ เพราะมันจะมีหน้าที่ควบคุมการทำงานทั้งหมดของระบบ ไม่ว่าจะเป็นการจัดสรรทรัพยากรของระบบ การเตรียมระบบจัดเก็บข้อมูลบริหารหน่วยความจำ โดยมันจะควบคุมอุปกรณ์ที่อยู่ทั้งในและนอกเครื่องทั้งหมด ดังนั้นเคอร์เนลจึงขึ้นอยู่กับตัวฮาร์ดแวร์ ถ้าตัวฮาร์ดแวร์เปลี่ยนรุ่นไป ตัวเคอร์เนลก็จะต้องเปลี่ยนไปด้วย
- **เชล** เป็นเสมือนตัวกลางระหว่างผู้ใช้กับตัวเคอร์เนล มันจะคอยรับคำสั่งจากผู้ใช้ทางอุปกรณ์อินพุท เช่น คีย์บอร์ด แล้วทำการแปลเป็นภาษาให้เครื่องเข้าใจ หรือจะเรียกได้อีกอย่างหนึ่งว่า command interpreter และยังสามารถนำคำสั่งเหล่านี้มาเขียนรวมกันในลักษณะของโปรแกรม หรือที่เราเรียกว่า เชลสคริปต์ นอกจากนี้ยังเป็นตัวควบคุมและกำหนดทิศทางของเอาต์พุตด้วยว่าจะให้เข้าหรือออกมาทางใด เช่น อาจให้เอาต์พุตออกหน้าจอหรืออาจเก็บเป็นไฟล์ก็ได้
- **โปรแกรมประยุกต์** เป็นส่วนของโปรแกรมที่ยูนิกซ์มีไว้เพื่ออำนวยความสะดวกให้กับผู้ใช้ เช่น โปรแกรม pine ซึ่งเป็นโปรแกรมรับส่งจดหมายอิเล็กทรอนิกส์ เป็นต้น

2.8.1 เชลเริ่มทำงานได้อย่างไร

มันจะเริ่มทำงานก็ต่อเมื่อมีการล็อกอินเข้าสู่ระบบ จากนั้นจะรอรับคำสั่งจากผู้ใช้งาน จนกระทั่งผู้ใช้งานล็อกเอาต์ ผู้ใช้งานแต่ละคนจะมีดีฟอลต์เชลของตนเอง โดยจะกำหนดในไฟล์ /etc/passwd ซึ่งในไฟล์นั้นจะประกอบด้วยหลายส่วนด้วยกัน

2.8.2 วิวัฒนาการของเชล

การพัฒนาของเชลนั้นเริ่มต้นมาจากระบบยูนิกซ์ ซึ่งถือว่าเป็นระบบปฏิบัติการต้นแบบของลีนุกซ์ โดยเชลนั้นมีการพัฒนาปรับปรุงควบคู่กับระบบยูนิกซ์มาโดยตลอดรวมทั้งลีนุกซ์ด้วยดังนี้

- ทอมสันเชล (Thompson Shell) เป็นเชลตัวแรกที่ใช้ในระบบยูนิกซ์ แต่ช่วงระยะเวลาในการใช้นั้นไม่นานนัก
- มาร์เชย์เชล (Mashey Shell) เป็นเชลที่ จอห์น มาร์เชย์ (Johm Mashey) และทีมงานได้ปรับปรุงเชลเดิม โดยเพิ่มขีดความสามารถในการทำงานเข้าไป อีกทั้งยังกำหนดให้มีการควบคุมทิศทางการไหลของโปรแกรม โดยมีความตั้งใจให้เชลนี้สามารถประพฤติตัวเสมือนภาษาโปรแกรม
- บอร์นเชล (Bourne Shell) เป็นเชลที่ สตีฟ บอร์น (Steve Bourne) ได้กำหนดโครงสร้างภาษาโปรแกรมขึ้นโดยมีลักษณะคล้ายๆ กับภาษาอัลกอล (Algol) ไว้ในเชลชนิดนี้ พร้อมทั้งยังให้มีการใช้ในลักษณะของโพรซีเจอร์ (Procedure) ด้วย
- ซีเชล (C Shell) เป็นเชลที่ บิลล์ จอย (Bill Joy) แห่งเบอร์กเลย์ได้สร้างเชลใหม่ขึ้นมา ซึ่งมีลักษณะของไวยากรณ์คล้ายกับภาษาซี โดยสามารถควบคุมการไหลของข้อมูลและมีแฟ้มเก็บประวัติการใช้งานคำสั่งอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ซิสเต็มส์โฟว์เชล (System IV Shell) เป็นบอร์นเชลอีกเวอร์ชันหนึ่งที่เพิ่มฟังก์ชันการใช้งานต่างๆ เข้าไป
- V sh และ E sh เป็นบอร์นเชล เช่นเดียวกันแต่มีการเพิ่มเอดิเตอร์ vi กับ Emacs เข้าไป
- คอร์นเชล (Korn Shell) เป็นเชลที่ออกแบบและพัฒนาโดยเดวิด จี คอร์น (David G.Korn) แห่งศูนย์วิจัยเบลล์ โดยเชลนี้เป็นลักษณะของคำสั่งโต้ตอบ และคอร์นเชลมีขนาดใหญ่กว่าเชลอื่น ทำให้สามารถขยายขอบเขต และเพิ่มประสิทธิภาพการใช้งานได้มากยิ่งขึ้น การเขียนเชลสคริปต์ในคอร์นเชลนี้เขียนได้ง่ายและรัดกุม เป็นไปตามมาตรฐาน IEEE POSIX 1003.2 Shell and Utilities Language Committee
- บาชเชล (Bourne Again Shell หรือ Bash Shell) เป็นเชลที่กำหนดให้ผู้ใช้โดยอัตโนมัติในระบบลีนุกซ์ เชลนี้เป็นอีกโปรแกรมหนึ่งที่พัฒนาภายใต้ Free Software Foundation โดยเป็นไปตามมาตรฐานของ IEEE POSIX 1003.2 Shell and Utilities Language Committee และมีการนำเอาคุณสมบัติที่เด่นของคอร์นเชลและซีเชลมารวมกัน

2.8.3 การเลือกใช้เชล

เชลที่ใช้กันอยู่ในขณะนี้มีอยู่ด้วยกัน 3 ชนิด คือ ซีเชล คอร์นเชล และบาชเชล ซึ่งแต่ละเชลต่างก็มีข้อดีข้อเสียแตกต่างกันไป

- ซี หรือ ทีซีเชล (csh/tcsh) มีขีดความสามารถสูงกว่าบอร์นเชล มีแฟ้มเก็บคำสั่งการทำงาน แต่ไม่สามารถเข้าใจโครงสร้างของสคริปต์บอร์นเชลได้ อย่างไรก็ตามซีเชลนั้นสามารถเรียกบอร์นเชลขึ้นมาเพื่อรันสคริปต์ของบอร์นเชลได้
- คอร์นเชล (ksh) เป็นเชลที่มีการใช้งานกันอย่างแพร่หลายในกลุ่มผู้ใช้ยูนิกซ์ คอร์นเชลได้รวมคุณลักษณะเด่นของบอร์นเชลและซีเชลเข้าด้วยกัน อย่างไรก็ตาม เมื่อผู้ใช้งานลีนุกซ์ต้องการใช้งานคอร์นเชล จำเป็นต้องติดตั้งเพิ่มเติมโดยเป็นคอร์นเชลสาธาณะอย่าง pdksh (public domain korn Shell) ก็ได้
- บาชเชล (bash) ถือว่าเป็นเชลมาตรฐานชนิดใหม่ของระบบลีนุกซ์ โดยเชลนี้สามารถใช้รันสคริปต์ที่เขียนด้วยซีเชลและคอร์นเชลได้

การเลือกใช้เชลว่าจะใช้เชลชนิดใดนั้น ขึ้นอยู่กับความต้องการของผู้ใช้ เนื่องจากผู้ใช้ลีนุกซ์สามารถเลือกชนิดของเชลได้ตามความต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
[ohh@isag07 ohh]$
[ohh@isag07 ohh]$ echo $SHELL
/bin/bash
[ohh@isag07 ohh]$ chsh
Changing shell for ohh.
Password:
New shell [/bin/bash]: /bin/tcsh
Shell changed.
[ohh@isag07 ohh]$
[ohh@isag07 ohh]$ chsh
Changing shell for ohh.
Password:
New shell [/bin/tcsh]: /bin/bash
Shell changed.
[ohh@isag07 ohh]$ echo $BASH_VERSION
2.05b.0(1)-release
[ohh@isag07 ohh]$
[ohh@isag07 ohh]$ import /mnt/d/pic/2-5.jpg
import: Unable to open file (/mnt/d/pic/2-5.jpg).
[ohh@isag07 ohh]$
```

รูปที่ 2-5 : แสดงการเปลี่ยนเชลใหม่

2.8.4 การทำงานของเชล

ลีนุกซ์เป็นระบบปฏิบัติการแบบมัลติทาส์กิง หมายความว่า ระบบสามารถทำงานได้พร้อมกันหลายๆ งาน แต่สำหรับเชลนั้นจะรองรับการทำงานได้เพียงงานเดียวเท่านั้น ดังนั้นลีนุกซ์จึงต้องมีระบบการจัดการที่ทำให้รองรับการทำงานหลายงานพร้อมๆ กันได้โดยไม่ต้องรอให้งานอื่นเสร็จก่อน

การจัดการในระบบลีนุกซ์ในยุคต้นๆ นั้นการทำงานของคำสั่งจะมี 2 ลักษณะ คือ

2.8.4.1 การทำงานแบบซิงโครนัส (Synchronous Execution) เป็นการทำงานตามลำดับคำสั่ง คำสั่งก่อนหน้าต้องทำงานจนจบก่อนจึงจะพร้อมรับหรือดำเนินการคำสั่งต่อไป การทำงานในลักษณะนี้เรียกว่า การทำงานฉากหน้า (foreground mode)

```
[ohh@isag07 ohh]$
[ohh@isag07 ohh]$ who; date; cal
root     :0           Jan 29 22:44
root     pts/0        Jan 29 22:44
root     pts/1        Jan 29 22:44
root     pts/2        Jan 29 22:47
Wed Jan 29 23:09:11 ICT 2003
    January 2003
Su Mo Tu We Th Fr Sa
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31

[ohh@isag07 ohh]$
```

รูปที่ 2-6 : ตัวอย่างการทำงานฉากหน้า

2.8.4.2 การทำงานแบบอะซิงโครนัส (Asynchronous Execution) เป็นการทำงานตามคำสั่งที่กำหนดพร้อมกับสร้างเชลขึ้นใหม่เพื่อรอรับคำสั่งถัดไป โดยต้องรอให้คำสั่งก่อนหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

ทำงานเสร็จสิ้นก่อน การทำงานในลักษณะนี้เรียกว่า การทำงานในฉากหลัง (background mode) การกำหนดให้ทำงานแบบนี้เพียงแค่เติมเครื่องหมาย ampersand (&) ที่ท้ายคำสั่ง

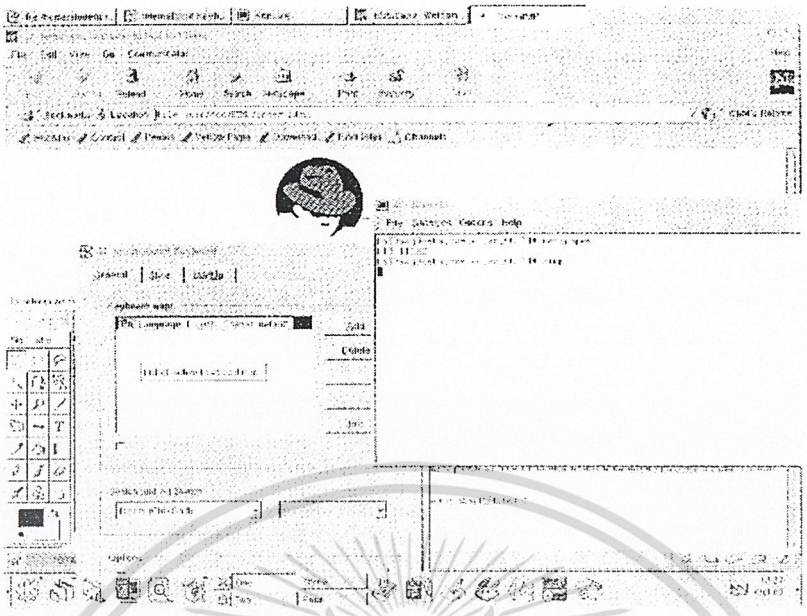
```
[ohh@isag07 ohh]$ who& date& cal&
[1] 1262
[2] 1263
[3] 1264
[ohh@isag07 ohh]$ root     :0           Jan 29 22:44
root     pts/0        Jan 29 22:44
root     pts/1        Jan 29 22:44
root     pts/2        Jan 29 22:47
Wed Jan 29 23:10:48 ICT 2003
    January 2003
Su Mo Tu We Th Fr Sa
           1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31

[ohh@isag07 ohh]$
[1]   Done                    who
[2]-  Done                    date
[3]+  Done                    cal
[ohh@isag07 ohh]$
```

รูปที่ 2-7 : ตัวอย่างการทำงานฉากหลัง

การทำงานฉากหลังสามารถทำได้หลายคน แต่สำหรับการทำงานฉากหน้านั้นจะมีเพียงงานเดียวเท่านั้น เพราะหากมีงานฉากหน้าหลายงานแล้ว ผลลัพธ์ที่แสดงออกมานั้นอาจจะก่อให้เกิดความสับสนกับผู้ใช้ได้ว่าผลลัพธ์ที่แสดงนั้นเป็นของงานใด ในส่วนของการทำงานหลายงานในฉากหน้านี้ได้มีการพัฒนาเพื่อให้สามารถใช้งานได้ในรูปแบบของวิธีควบคุมงาน (job control) และระบบหน้าต่าง (windows)

- **วิธีควบคุมงาน** เป็นวิธีที่ให้ระบบลีนุกซ์สามารถกระทำหรือหยุดรอของโพรเซส รวมถึงการติดต่อกับอินพุตและเอาต์พุตด้วย โดยการควบคุมงานทำให้มีการสลับเปลี่ยนระหว่างงานฉากหน้ากับงานฉากหลังได้ โดยการสลับโพรเซสนี้ อาจใช้ปุ่มควบคุมเป็นคำสั่งในการจัดการ อย่างไรก็ตามลักษณะของวิธีควบคุมงานนั้นจะไม่สามารถเก็บสภาวะต่างๆ ได้หมดในสภาวะพักงาน (suspend)
- **ระบบหน้าต่าง** เป็นลักษณะของหน้าต่างเล็กๆ บนจอภาพที่เรียกว่าวินโดว์ส (windows) โดยแต่ละวินโดว์สสามารถทำงานกับโพรเซสได้เป็นอิสระต่อกัน และทุกโพรเซสในแต่ละวินโดว์สนั้นับว่าเป็นโพรเซสฉากหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-8 : ระบบหน้าต่าง

ระบบวินโดว์สเป็นระบบที่มีความนิยมเป็นอย่างมากในปัจจุบัน เพราะสามารถทำงานได้หลายงานตามความต้องการ เช่น ใช้เอดิเตอร์ในการเขียนโปรแกรมในขณะเดียวกันก็คอมไพล์โปรแกรม แต่สิ่งที่ต้องสูญเสียในการใช้ระบบวินโดว์ส คือ ค่าใช้จ่ายสูงสำหรับอุปกรณ์ต่างๆ เช่น การ์ดและจอภาพที่มีประสิทธิภาพและที่สำคัญอีกประการหนึ่งคือ จะต้องมีการแบ่งเวลาการประมวลผลในการควบคุมการทำงานของระบบวินโดว์สด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ความรู้พื้นฐานในการบุกรุกระบบ

ในยุคปัจจุบัน คงไม่มีใครปฏิเสธหากจะเอ่ยว่า คอมพิวเตอร์กับมนุษย์ต่างก็มีความสัมพันธ์กันมากขึ้นทุกที มนุษย์ต้องการให้คอมพิวเตอร์ยื่นมือเข้ามามีบทบาทในทุกๆ เรื่องเท่าที่จะเป็นไปได้เพื่อความสะดวกสบายของตน การประยุกต์คอมพิวเตอร์เข้ามาใช้กับทุกๆ เรื่องในชีวิตประจำวันกลายมาเป็นเรื่องปกติธรรมดา

ความไว้ใจในคอมพิวเตอร์ทำให้มันเป็นที่เก็บข้อมูลที่สำคัญของตัวเองและผู้อื่นเอาไว้เป็นจำนวนมาก เช่น ข้อมูลส่วนตัว รูปพรรณสัณฐาน อายุ น้ำหนัก ส่วนสูง ระดับการศึกษา สถานภาพทางการสมรส ณ นาทีนี้ ไม่ว่าเราจะยินยอมหรือไม่ยินยอมก็ตาม ข้อมูลส่วนตัวพวกนี้จำเป็นต้องมีอยู่ในคอมพิวเตอร์ทั้งสิ้น

เมื่อคอมพิวเตอร์ถูกป้อนข้อมูลมากขึ้นเรื่อยๆ ไม่ว่าจะเป็นเรื่องใดก็ต้องใช้คอมพิวเตอร์ทั้งนั้น จนพูดได้ว่า “ทุกๆ อย่างขึ้นอยู่กับคอมพิวเตอร์ อะไรๆ ก็ต้องใช้คอมพิวเตอร์” ถ้าหากว่าระบบคอมพิวเตอร์สมบูรณ์แบบทุกอย่างมันก็เป็นเรื่องดีทีเดียว แต่หากเป็นเช่นนั้นไม่ เพราะในความเป็นจริงอะไรก็ตามที่มีความสำคัญมักจะเป็นที่ๆ ต้องการของอาชญากร อย่างเช่น ในธนาคารมีเงินเก็บอยู่มหาศาล ดังนั้นโจรจึงไปปล้นกันที่นั่น ซึ่งเทียบได้กับว่า ในเครื่องคอมพิวเตอร์มีข้อมูลสำคัญอยู่ จึงเป็นที่ต้องการของผู้ที่สนใจ เป็นเหตุให้เกิดอาชญากรรมทางคอมพิวเตอร์ขึ้นมา

อาชญากรรมทางคอมพิวเตอร์ เป็นอาชญากรรมในรูปแบบใหม่ที่เกิดขึ้นในโลกที่ไร้พรมแดน หรือที่เราเรียกว่า ไซเบอร์สเปซ (cyberspace) มันเป็นสิ่งที่อาจเกิดขึ้นได้เสมอทุกๆ วัน จากคอมพิวเตอร์แวดล้อมรอบข้างตัวเรา โดยตัวของไซเบอร์สเปซเองไม่มีข้อจำกัดทางภูมิศาสตร์ ด้วยเหตุนี้เราจึงไม่สามารถปฏิเสธได้เลยว่า มีโอกาสสูงขึ้นเรื่อยๆ ที่เราทุกคนจะต้องเผชิญกับอาชญากรรมคอมพิวเตอร์ ไม่วันนี้ก็วันหน้า

3.1 อาชญากรและอาชญากรรมคอมพิวเตอร์

อาชญากรคอมพิวเตอร์ คือ ผู้กระทำการผิดกฎหมายโดยใช้เทคโนโลยีคอมพิวเตอร์เป็นส่วนสำคัญ มีการจำแนกเอาไว้ดังนี้

- Novico เป็นพวกเด็กหัดใหม่ (newbies) ที่เพิ่งเริ่มใช้คอมพิวเตอร์มาได้ไม่นาน หรืออาจหมายถึงพวกที่เพิ่งได้รับความไว้วางใจให้เข้าสู่ระบบเครือข่ายคอมพิวเตอร์
- Daranged person คือ พวกจิตวิปริต ผิดปกติ มีลักษณะเป็นพวกชอบความรุนแรงและอันตราย มักจะเป็นพวกที่ชอบทำลายทุกสิ่งที่ขวางหน้า ไม่ว่าจะเป็นสิ่งของ บุคคล หรือสภาพแวดล้อม เนื่องจากอาชญากรประเภทนี้มีไม่มากนัก จึงทำให้ผู้รักษากฎหมายไม่ค่อยได้ให้ความสนใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Organized Crime พวกนี้เป็นกลุ่มอาชญากรที่ร่วมมือกันกระทำในลักษณะขององค์กรใหญ่ๆ ที่มีระบบ พวกเขาจะใช้คอมพิวเตอร์ในลักษณะที่แตกต่างกัน โดยส่วนหนึ่งอาจใช้เป็นเครื่องมือในการหาข่าวสารเหมือนกับองค์กรธุรกิจทั่วไป อีกส่วนหนึ่งคือ ใช้เทคโนโลยีเพื่อเป็นตัวประกอบสำคัญในการก่ออาชญากรรม หรือใช้เทคโนโลยีนี้เพื่อกลบเกลื่อนร่องรอย ซ่อน หลบหนี ให้พ้นจากการติดตามของเจ้าหน้าที่
- Career Criminal พวกอาชญากรมืออาชีพ เป็นกลุ่มอาชญากรคอมพิวเตอร์ที่มีอยู่มาก กลุ่มนี้น่าเป็นห่วงมากที่สุด เนื่องจากนับวันจะทวีจำนวนมากขึ้นเรื่อยๆ กระทำความผิดซ้ำแล้วซ้ำเล่า โดนจับกุมบ่อยครั้ง สำนึกแห่งความดีมิได้ฝังอยู่ในสันดาน
- Com Artiot คือ พวกหัวพัฒนา เป็นพวกทีชอบใช้ความก้าวหน้าทางคอมพิวเตอร์เพื่อให้ได้มาซึ่งผลประโยชน์ส่วนตน อาชญากรประเภทนี้จะฉวยใช้ความรู้เกี่ยวกับระบบคอมพิวเตอร์และเทคโนโลยีอันล้ำสมัย เพื่อหาเงินให้กับตนเอง โดยมิชอบด้วยกฎหมาย
- Dreamer พวกบ้าลัทธิ เป็นพวกที่คอยทำผิดเนื่องจากมีความเชื่อถือสิ่งหนึ่งสิ่งใดอย่างรุนแรง
- Cracker หมายถึง ผู้ที่มีความรู้และทักษะทางคอมพิวเตอร์เป็นอย่างดี จนสามารถลักลอบเข้าสู่ระบบได้ ซึ่งจุดประสงค์คือเข้าไปหาประโยชน์จากการบุกรุกโดยที่ไม่ได้รับอนุญาต ผลที่ติดตามมาคือการสร้างปัญหาและการก่อความเสียหายให้เกิดขึ้น เช่น การทำลายหรือลบไฟล์ หรือทำให้เครื่องคอมพิวเตอร์ใช้การไม่ได้รวมทั้งทำลายระบบปฏิบัติการให้พังพินาศ
- Hacker คือบุคคลที่มีความสนใจในกลไกการทำงานของระบบปฏิบัติการบนคอมพิวเตอร์อย่างลึกซึ้ง แฮ็กเกอร์ส่วนใหญ่ต้องมีความรู้เทียบเท่าหรือเหนือกว่าโปรแกรมเมอร์ โดยจะเป็นเช่นนั้นได้ เพราะว่าพวกเขามีความใส่ใจที่จะนำความรู้พื้นฐานที่ผู้อื่นมองว่าธรรมดามาประยุกต์ใช้ ทั้งนี้เพื่อให้เกิดแนวความคิดสร้างสรรค์ใหม่ๆ ขึ้นอยู่ในสังคมดิจิตอลตลอดเวลา แฮ็กเกอร์จะมีความเข้าใจในจุดอ่อนของระบบและที่มาของจุดอ่อนนั้น เนื่องจากคอยติดตามข่าวสารและความรู้ใหม่ๆ อยู่ตลอดเวลา การกระทำใดๆ ที่เกิดจากการศึกษาของแฮ็กเกอร์จะต้องแน่ใจแล้วว่า ไม่ก่อให้เกิดความเสียหายแก่ข้อมูล

3.2 อาชญากรรมคอมพิวเตอร์

อาชญากรคอมพิวเตอร์จะก่ออาชญากรรมหลายรูปแบบ ซึ่งปัจจุบันทั่วโลกจัดออกเป็น 9 ประเภท (ตามข้อมูลที่ได้จากสัมมนาพิเศษที่จัดขึ้นโดยคณะอนุกรรมการเฉพาะกิจยกร่างกฎหมายอาชญากรรมทางคอมพิวเตอร์ กลางเดือนตุลาคม พ.ศ. 2541) คือ

1. การขโมยข้อมูลทางอินเทอร์เน็ต ซึ่งรวมถึงการขโมยประโยชน์ในการลักลอบใช้บริการ
2. อาชญากรนำเอาระบบการสื่อสารมาปกปิดความผิดของตัวเอง เช่น ใช้คอมพิวเตอร์ปกปิดความผิดตนเองโดยตั้งรหัสไม่ให้คนอื่นรู้
3. การละเมิดลิขสิทธิ์ปลอมแปลงรูปแบบเสียงและลักลอบเลียนแบบระบบซอฟต์แวร์โดยมิชอบ
4. ใช้คอมพิวเตอร์แพร่ภาพ เสียง ลามาอนาจาร และข้อมูลที่ไม่สมควร เช่น วิธีก่อการร้าย วิธีผลิตระเบิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ใช้คอมพิวเตอร์ฟอกเงิน
6. อันธพาลทางคอมพิวเตอร์ที่เข้าไปก่อกวน ทำลาย ระบบสาธารณูปโภค เช่น ระบบจ่ายไฟ จ่ายน้ำ ระบบจราจร
7. หลอกลวงให้ร่วมค้าขายหรือลงทุนปลอม
8. แทรกแซงข้อมูลแล้วนำข้อมูลนั้นมาเป็นประโยชน์ต่อตนโดยมิชอบ เช่น ลักลอบเข้าระบบอินเทอร์เน็ตแล้วค้นรหัสบัตรเครดิตคนอื่นมาใช้ ดักฟังข้อมูลทางการค้าของผู้อื่นเพื่อประโยชน์ทางการค้าของตนเอง
9. ใช้คอมพิวเตอร์แอบโอนเงินบัญชีผู้อื่นเข้าบัญชีตัวเอง

3.3 ประเภทของการบุกรุกระบบ

การบุกรุกเข้าสู่ระบบแบ่งออกเป็นประเภทหลักๆ 3 ประเภทดังนี้ คือ

1. การบุกรุกทางกายภาพ (Physical Intrusion) ผู้บุกรุกพยายามบุกรุกที่เครื่องเป้าหมายโดยตรง อาจเข้ามาใช้สิทธิพิเศษจากการทำงานที่คอนโซล (Console) เช่น การ Boot Single Mode เพื่อให้ได้สิทธิพิเศษเลย (ใช้ในกรณีที่ลืมรหัสผ่านรูต) หรือถอดย้ายอุปกรณ์ เช่น ฮาร์ดดิสก์ ซึ่งอาจนำไปเขียนหรืออ่านภายหลังหรือบายพาสไบออสได้

2. การบุกรุกทางระบบ (System Intrusion) ผู้บุกรุกเข้ามาในระบบ โดยปกติมักเป็นผู้ใช้งานที่มีสิทธิต่ำ ถ้าระบบไม่ได้ใส่แพตช์ (Patch) ที่สามารถแก้บักของแอพพลิเคชันแล้ว จุดนี้ก็เป็นช่องโหว่ที่ทำให้ผู้ใช้งานคนนั้นสร้างสิทธิของตัวเองให้มากขึ้น จนเทียบเท่ารูตได้ เนื่องจากแอพพลิเคชันที่ใช้งานเกือบทุกอย่างมีบักอยู่ ถ้ายังไม่สามารถทำให้บักนั้นหมดไปหรือลดลงไปได้ มันจะเป็นช่องทางที่ง่ายในการบุกรุกระบบมาก

3. การบุกรุกระยะไกล (Remote Intrusion) ผู้บุกรุกติดต่อผ่านทางเน็ตเวิร์ก มีหลายเทคนิคในการบุกรุกระบบแบบนี้ ปัจจุบันมีแอพพลิเคชันประเภทไฟร์วอลล์ (Firewall) ทำหน้าที่เป็นด่านแรกในการป้องกันการบุกรุกทางเน็ตเวิร์ก แต่ไม่สามารถกันผู้บุกรุกได้ร้อยเปอร์เซนต์ อันเนื่องมาจากการตั้งค่าโดยใช้ค่าที่ไม่เหมาะสมที่ระบบกำหนดมาให้

3.4 ปัญหาทางด้านความปลอดภัยของระบบ

ปัญหาทางด้านความปลอดภัยในระบบปฏิบัติการยูนิกซ์นั้น สามารถแบ่งออกได้เป็น 2 ส่วนหลักๆ คือ ปัญหาในส่วนของโฮส และปัญหาในส่วนของเครือข่าย โดยส่วนของโฮสนั้นจะเกิดจากการตั้งค่าบางอย่างในระบบผิดพลาด ทำให้เกิดช่องโหว่ขึ้นในระบบ ยกตัวอย่างเช่น การตั้งรหัสผ่านของผู้ใช้งานแต่ละคนง่ายต่อการคาดเดา หรือง่ายต่อการถอดรหัส การตั้งค่าเพอร์มิสชันของไฟล์ต่างๆ ในระบบไม่ถูกต้อง การตั้งค่าการทำงานต่างๆ ของแอพพลิเคชันไม่ดีพอ ทำให้เกิดการทำงานที่ไม่พึงประสงค์เกิดขึ้น ส่วนของเครือข่ายจะเกิดจากความบกพร่องในการตรวจสอบการบริการทางเครือข่ายของระบบ ยกตัวอย่างเช่น การเปิดให้บริการอื่นๆ ที่ไม่ได้ใช้งาน การเปิดการบริการที่มีปัญหาด้านความปลอดภัย หรือแม้แต่การตั้งค่าการทำงานของแอพพลิเคชันทางเครือข่ายไม่ดีพอ ก็ทำให้เกิดปัญหาได้เช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 สาเหตุของการเกิดปัญหาความไม่ปลอดภัยในระบบ

ปัญหาความไม่ปลอดภัยในระบบยูนิกซ์ที่อาจทำให้เกิดช่องโหว่ หรือการถูกโจมตีนั้น สามารถเกิดขึ้นจากทุกๆ ส่วนที่เกี่ยวข้องตั้งแต่ ไลบรารี, เพอร์มิสชัน, เซอร์วิส รวมถึงการตั้งค่าคอนฟิกูเรชั่นที่ไม่ถูกต้อง โดยเราสามารถแยกปัญหาความปลอดภัยออกได้เป็น 3 แบบใหญ่ๆ คือ

3.5.1 ความผิดพลาดของผู้ดูแลระบบ

3.5.2 ความผิดพลาดจากตัวโปรแกรม

3.5.3 การประทุษร้ายจากบุคคลภายนอก

3.5.1 ความผิดพลาดของผู้ดูแลระบบ

3.5.1.1 ติดตั้งโปรแกรมที่ไม่ได้ใช้งาน หรือเปิดพอร์ตที่ไม่ควรเปิด

ผู้ดูแลระบบที่ดีควรทราบว่าระบบของตนมีการทำงานอะไรได้บ้าง และมีการติดตั้งแอพพลิเคชันใดบ้างในระบบ ซึ่งข้อผิดพลาดสำหรับการดูแลระบบอันดับแรก คือ การติดตั้งโปรแกรมมากกว่าการใช้งานจริงๆ โดยคิดว่าอาจจะได้ใช้ในอนาคต ความผิดพลาดนี้ทำให้เกิดช่องโหว่ในระบบขึ้นได้ เนื่องจากโปรแกรมที่ลงไปนั้นอาจจะมี bug ที่ทำให้ระบบเกิดช่องโหว่ได้ ส่วนข้อผิดพลาดอีกข้อหนึ่ง คือ การที่ผู้ดูแลระบบไม่รู้ว่าในระบบของตนได้ติดตั้งโปรแกรมอะไรไว้บ้าง ซึ่งจะเกิดปัญหาในกรณีที่พบ bug ของโปรแกรมที่ได้ติดตั้งไปในระบบ แต่ผู้ดูแลระบบไม่ทราบจึงไม่ได้ patch เพื่อแก้ไข bug ดังกล่าวจะทำให้เกิดอันตรายต่อระบบในระยะยาวได้เช่นกัน

นอกจากการติดตั้งโปรแกรมในระบบแล้ว การเปิดพอร์ตเพื่อรอรับการเชื่อมต่อจากภายนอกก็เป็นปัญหาเช่นเดียวกัน ผู้ดูแลระบบควรเปิดพอร์ตเฉพาะพอร์ตที่จำเป็นต้องเปิดเท่านั้น ไม่ควรเปิดพอร์ตที่ไม่จำเป็นเพราะทุกๆ พอร์ตที่เปิดสามารถเป็นช่องทางในการบุกรุกเข้าสู่ระบบได้ทั้งนั้น ผู้ดูแลระบบที่ดีควรระลึกไว้เสมอว่าเพื่อความปลอดภัยแล้ว การติดตั้งโปรแกรมหรือเปิดพอร์ตใดๆ ควรทำเฉพาะที่จำเป็นเท่านั้น

3.5.1.2 ตั้งค่าต่างๆ ในโปรแกรมมีปัญหา

เป็นปัญหาความปลอดภัยที่เกิดจากความผิดพลาดของผู้ดูแลระบบเอง เนื่องจากโปรแกรมหนึ่งๆ ที่จะรันในระบบจะมีการตั้งค่าต่างๆ มากมาย โดยผู้ดูแลระบบจะต้องทำการปรับเปลี่ยนให้เหมาะสม แต่ส่วนใหญ่แล้วผู้ดูแลระบบมักจะไม่ใส่ใจในการตั้งค่าเหล่านี้เท่าใดนัก โดยจะทำเพียงตั้งค่าการทำงานให้สามารถทำงานได้เท่านั้น หรือไม่ก็ใช้ค่าดีฟอลต์เลย ทำให้เกิดการทำงานอื่นๆ ที่ไม่ควรให้ระบบทำงาน ในบางโปรแกรมจะมีการตั้งค่าเพื่อความปลอดภัยอยู่ด้วย แต่ก็ถูกละเลยไปเช่นกัน

3.5.1.3 รหัสผ่านตั้งไว้ไม่ดีพอ

ส่วนใหญ่เกิดจากผู้ใช้งานมักจะใช้ชื่อที่คุ้นเคย เช่น ชื่อตัวเอง ชื่อเพื่อน สัตว์เลี้ยง วันเกิด หรือเบอร์โทรศัพท์เป็นรหัสผ่าน ซึ่งผู้บุกรุกสามารถที่จะคาดเดาได้ง่าย แต่ถ้ารหัสผ่านนั้นไม่สามารถที่จะคาดเดาเองได้แล้วนั้น ผู้บุกรุกก็จะใช้วิธีการเดาจากคำในพจนานุกรม หรืออาจใช้วิธีการบุกรุกโดยพลการ (Brute Force Attack) หากว่าผู้ใช้งานตั้งรหัสผ่านไว้ไม่ดีพอ ก็อาจทำให้จุดนี้เกิดเป็นช่องโหว่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

3.5.2 ความผิดพลาดของตัวโปรแกรมเอง

3.5.2.1 Vulnerablility

ปัญหา Vulnerable คือ ปัญหาที่เกิดจากระบบมีจุดอ่อน ที่อาจทำให้ระบบถูกโจมตีได้ ในรูปแบบใดรูปแบบหนึ่ง ซึ่งอาจเกิดขึ้นกับโปรแกรมใดๆ ในระบบก็ได้ ไม่ว่าจะเป็นเซอร์วิสที่ให้บริการอยู่ในระบบ เช่น เมล์เซิร์ฟเวอร์ เวปเซิร์ฟเวอร์ ดีเอ็นเอสเซิร์ฟเวอร์ เซอร์วิสอื่นๆ หรือแม้แต่คำสั่งต่างๆ ในระบบ ซึ่งอาจเกิดขึ้นเนื่องจากมีความผิดปกติของการออกแบบการทำงานของโปรแกรมนั้นๆ หรืออาจเป็นความผิดปกติของไลบรารีที่โปรแกรมนั้นๆ ใช้งานอยู่ เป็นต้น จากช่องโหว่ดังกล่าวจะทำให้ระบบเกิดอันตรายตั้งแต่การใช้ช่องทางดังกล่าวเพื่อทำ Denial of Service การส่งคำสั่งให้โปรแกรมดังกล่าวช่วยรันให้ หรือการเข้าไปครอบครองระบบโดยผ่านช่องทางดังกล่าว

3.5.2.2 Buffer Overflow

ปัญหา Buffer Overflow คือ ปัญหาเกี่ยวกับการใช้งานหน่วยความจำเกินจากส่วนที่จองไว้ ซึ่งมักจะเป็นส่วนที่เป็นปัญหามากที่สุดสำหรับโปรแกรมต่างๆ ในภาษา C สำหรับปัญหา Buffer Overflow นี้ทำให้ service ที่ให้บริการนั้นๆ มีปัญหา และไม่สามารถทำงานต่อไปได้

ตัวอย่างสำหรับโปรแกรมที่ทำให้เกิด Buffer Overflow ได้เช่น

```
Void func(void)
{
        int m ;
        char buffer[256] ;
        for ( m = 0; m < 512; m++; )
                buffer[m] = 'a' ;
        return ;
}
```

โดยโปรแกรมตัวอย่างจะมีการจองเนื้อที่หน่วยความจำไว้แค่ 256 ไบต์ แต่เมื่อมีการทำงานจะมีการใส่ค่าตัวอักษร 'a' ลงในหน่วยความจำมากกว่าที่ได้จองไว้ คือใส่ไปถึง 512 ไบต์ ในกรณีนี้โปรแกรมจะเกิดปัญหา buffer overflow ขึ้นทันที กรณีที่จะทำให้เกิด buffer overflow ขึ้นในแอพพลิเคชันนั้นเกิดได้ในหลายๆ กรณี เช่น การป้อนอินพุทในปริมาณมากไปยังระบบซึ่งไม่มีการตรวจสอบขอบเขตของอินพุทก่อนที่จะทำงาน ทำให้การส่งค่าอินพุทเก็บไว้ในหน่วยความจำเกินขอบเขตที่ได้จองไว้

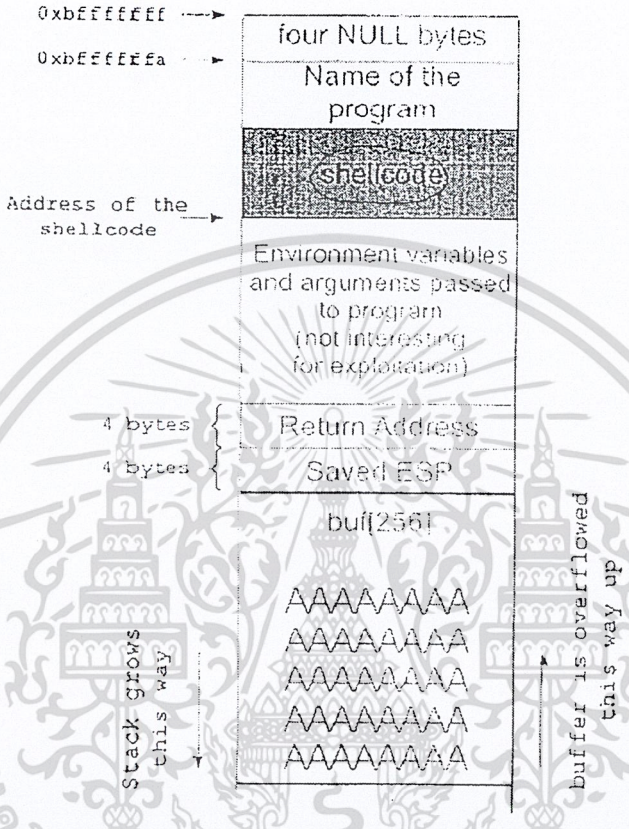
3.5.2.3 Exploit

ปัญหา exploit คือ ปัญหาที่สืบเนื่องมาจาก vulnerable และการทำ buffer overflow คือ จะเป็นการใช้ประโยชน์จากระบบที่มีปัญหา ให้ทำงานบางอย่างให้ โดยปัญหาดังกล่าวเกิดขึ้นเพราะความผิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

49900

พลาดในการเขียนโปรแกรม ทำให้ผู้ไม่ประสงค์ดีสามารถโจมตีได้โดยการส่งโค้ด ที่ระบบเปิดช่องโหว่ให้โค้ดนั้นทำงานได้ ยกตัวอย่างเช่น cifslogin ใน HP-UX ซึ่งสามารถส่งโค้ดบางอย่างให้ระบบทำงานและได้สิทธิในการทำงานตามสิทธิของโปรแกรม cifslogin ด้วย



รูปที่ 3-1 : การเกิด buffer overflow และ Exploit

โดยปกติแล้ว การทำงานของระบบปฏิบัติการเมื่อมีการเรียกโปรแกรมใดๆ หรือฟังก์ชันใดๆ ขึ้นมาทำงาน จะมีการเก็บค่า retun address ไว้ใน stack เพื่อให้ระบบกลับมาทำงานที่จุดเดิมก่อนการเรียกฟังก์ชัน แล้วตามด้วยค่า ESP ซึ่งถ้าในฟังก์ชันที่มีการจองเนื้อที่หน่วยความจำไว้สำหรับเก็บตัวแปรต่างๆ ก็จะมีการจองหน่วยความจำไว้ต่อจาก ESP ดังตัวอย่างโปรแกรม

```
Int main ( int argv. Char** argc )
{
        Char buf[256] ;
        Strcpy (buf, argc[1]) ;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตัวอย่างโปรแกรมจะ push ค่า return address เก็บไว้ใน stack แล้วตามด้วยค่า ESP หลังจากนั้นจะมีการจองเนื้อที่ไว้ขนาด 256 ตัวอักษร แต่ในขณะทำงานถ้ามีการป้อนข้อมูลเกินกว่าที่โปรแกรมได้จองไว้จะเกิดปัญหา buffer overflow ขึ้น ซึ่งในกรณีของการทำ exploit นั้น เราจะมีการทำงานอีกสองส่วนเพิ่มขึ้นมา คือ การเปลี่ยนแปลงค่า return address ให้ไปทำงานยังจุดที่เราต้องการแทนที่จะให้โปรแกรมกลับไปทำงานที่จุดเดิมก่อนการเรียกใช้ โดยปัญหานี้เกิดจากการออกแบบระบบให้ทิศทางการขยายตัวของ stack กับทิศทางการขยายตัวของ buffer สวนทางกัน ดังรูปที่ 3-1

3.5.3 การประทุษร้ายจากบุคคลอื่นๆ

ปัญหาการประทุษร้ายจากบุคคลอื่นๆ นั้น ส่วนใหญ่จะเป็นปัญหา Denial of Service ซึ่งก็คือการรบกวนระบบทำให้ระบบทำงานผิดปกติ หรือทำให้ระบบไม่สามารถทำงานได้เลย การทำ Denial of Service นั้นสามารถทำได้ทั้งระดับเครือข่ายและระดับแอพพลิเคชัน โดยในระดับเครือข่ายนั้นจะเป็นการสร้างแพ็กเก็ตที่มีความผิดปกติทั้งในด้านรูปแบบ ขนาด หรือลำดับการส่งแพ็ตเก็ตที่ผิดไปจากโปรโตคอล ทำให้การทำงานของระบบเครือข่ายผิดพลาดไปได้

สำหรับในส่วนของแอพลิเคชันนั้น จะเป็นการสร้างการทำงานเพื่อใช้ทรัพยากรในระบบเป็นหลัก ยกตัวอย่างเช่น การทำ super forker เป็นต้น ความอันตรายของการทำ Denial of Service นั้นถือว่ามีระดับความสูงมาก เนื่องจากเป็นเหมือนการสร้างการทำงานมหาศาลขึ้นในระบบ ในบางระบบหรือบางอุปกรณ์ที่ไม่สามารถทำงานได้ถึงระดับนั้น อาจมีปัญหาหรือเสียไปเลยก็ได้

วิธีป้องกันและแก้ไขในส่วนของการโจมตีผ่านเครือข่ายจะทำได้โดยการปรับปรุงสแตกโปรโตคอลของระบบ การติดตั้งไฟร์วอล ซึ่งในระบบยูนิกซ์ มักจะไม่มีปัญหาในส่วนนี้แล้ว แต่สำหรับการทำ Denial of service ในระดับแอพพลิเคชันนั้น ทำได้โดยการจัดสรรทรัพยากร และมีระบบป้องกันการจองทรัพยากรไว้มากเกินไปของผู้ใช้งานแต่ละคน เช่น จำกัดให้ผู้ใช้งานในระบบแต่ละคนจะรันโพรเซสได้เพียง 20-30 โพรเซส เป็นต้น

3.6 การทำงานหลังจากที่ผู้บุกรุกมีสิทธิสูงสุดในระบบแล้ว

- ทำลายข้อมูลทั้งหมด

ในกรณีที่เจาะเข้าสู่ระบบได้แล้วสามารถครอบครองระบบทั้งหมดได้ และมีสิทธิเป็นผู้ดูแลระบบแล้ว การทำงานที่ก่อให้เกิดความเสียหายสูงสุด คือ การลบข้อมูลทุกอย่างในระบบ การป้องกันไม่สามาถทำได้เลยเนื่องจากระบบยูนิกซ์จะให้สิทธิเต็มที่กับผู้ดูแลระบบ ให้สามารถทำงานใดๆ ก็ได้กับข้อมูล การแก้ไขคือต้องมีระบบการแบ็กอัพและการกู้คืนที่ดีเท่านั้น

- ปกปิดร่องรอย

เป็นการลบร่องรอยต่างๆ ของการเข้ามาในระบบ ซึ่งจะเป็นการลบข้อมูลภายในล็อกไฟล์ ซึ่งเป็นไฟล์ที่เก็บรายละเอียดการทำงานของระบบไว้ รวมถึงข้อมูลประวัติการเข้าใช้งานต่างๆ ด้วย และเป็นการลบร่องรอยการคงอยู่ของตัวเองในระบบได้ สำหรับในระบบยูนิกซ์นั้นจะมีไฟล์ที่เก็บการเข้าใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบ และประวัติการเข้าใช้งานระบบอยู่ในไฟล์ชื่อ utmp และ wtmp การเข้าไปแก้ไขไฟล์ทั้งสองจะทำให้ผู้ที่ใช้งานระบบไม่สามารถรู้ได้เลยว่ามีผู้บุกรุกอยู่ในระบบ ซึ่งปกติเราสามารถตรวจสอบผู้ที่เข้ามาใช้งานระบบได้ด้วยคำสั่ง finger และสามารถตรวจสอบประวัติการเข้าใช้งานระบบด้วยคำสั่ง last แต่เมื่อเราลบข้อมูลของตนเองใน wtmp และ utmp จะไม่มีใครเห็นผู้บุกรุกได้เลยจากคำสั่ง finger หรือ last ได้เลย แม้แต่ผู้ดูแลระบบเองก็ตาม

```
[root@isag07 home]# finger
Login    Name                    Tty      Idle  Login Time   Office     Office Phone
ann      Anny narak              tty2       42  Jan 24 17:08
hitomi   Viraiwan Madee          tty3           Jan 24 17:50
ohh      phunsiri Ngammaneewat   tty1       51  Jan 24 16:59
root     root                   *:0            Jan 24 16:55
root     root                    pts/0      55  Jan 24 16:55
root     root                   *pts/1          Jan 24 16:55
root     root                   *pts/2       4  Jan 24 17:03
[root@isag07 home]#
```

รูปที่ 3-2 : แสดงผลการ finger

แต่อย่างไรก็ตามผู้ดูแลระบบก็ยังสามารถตรวจหาร่องรอยการเปลี่ยนแปลง utmp และ wtmp ได้ โดยการเปรียบเทียบจำนวนผู้ล็อกอินกับจำนวนเชลในระบบได้ สำหรับการทำงานที่เป็นปกติเมื่อจับคู่ระหว่างเชลกับจำนวนผู้ที่ล็อกอินต้องไม่เหลือเศษ (แต่ในบางกรณีอาจเกิดลักษณะของโพรเซสที่ kill แล้วแต่ไม่สำเร็จ จึงค้างอยู่ในระบบ ผู้ดูแลระบบจะต้องตรวจสอบอีกครั้งว่าเป็นโพรเซสที่ไม่ยอมตายหรือว่าเป็นการบุกรุก)

```
[root@isag07 etc]# ps -def | grep sh
root         6     1  0 16:54 ?        00:00:00 [bdflush]
root       685     1  0 16:54 ?        00:00:00 /usr/sbin/sshd
root       726     1  0 16:54 ?        00:00:00 /bin/sh /usr/bin/safe_mysqld --d
root       924   913  0 16:55 ?        00:00:00 /bin/sh /usr/bin/startkde
root       984   924  0 16:55 ?        00:00:00 /usr/bin/ssh-agent /etc/X11/xini
root      1094  1092  0 16:55 pts/1    00:00:00 /bin/bash
ohh       1133   862  0 16:59 tty1     00:00:00 -bash
root      1207  1092  0 17:03 pts/2    00:00:00 /bin/bash
ann       1259   863  0 17:08 tty2     00:00:00 -bash
root      1371  1368  0 17:34 ?        00:00:00 /bin/sh /usr/share/redhat-config
root      1372  1371  0 17:34 ?        00:00:00 /usr/bin/python2.2 /usr/share/re
root      1390  1207  0 17:46 pts/2    00:00:03 /usr/bin/wish ./amsn
hitomi    1417   864  0 17:50 tty3     00:00:00 -bash
root      1850   844  0 18:05 ?        00:00:00 /bin/bash /usr/bin/run-parts /et
root      1855  1850  0 18:05 ?        00:00:00 /bin/bash /etc/cron.weekly/makew
root      1858  1855  0 18:05 ?        00:00:00 /bin/sh /usr/sbin/makewhatis -w
[root@isag07 etc]#
```

รูปที่ 3-3 : แสดง ps –def | grep ksh

- วางโทรจันหรือ backdoor & sniffer

การทำงานที่เป็นที่นิยมในหมู่ผู้บุกรุกระบบ คือ การฝังโทรจันและแบล็คดอร์ต่างๆ เพื่อให้การทำงานในระบบนั้นๆ ง่ายขึ้น เช่น การซ่อนตัวในระบบ การซ่อนโพรเซส การซ่อนไฟล์ต่างๆ ในระบบ หรือการสร้างช่องทางเพื่อเข้ามาในระบบอีกครั้งโดยไม่ให้ใครรู้ และอาจเป็นการเตรียมทางหนีที่ไล่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับตัวเองในกรณีฉุกเฉินด้วย และเมื่อผู้บุกรุกสามารถครอบครองระบบได้ทั้งหมดแล้ว นอกจากจะเตรียมทางหนีทีไล่สำหรับตัวเองแล้ว ยังต้องการข้อมูลเพิ่มเติมเพื่อทำการเจาะระบบอื่นๆ อีก โดยเทคนิคที่เป็นที่นิยม คือ การติดตั้งโปรแกรม sniffer ดักจับข้อมูลในเครือข่ายแล้วคัดเฉพาะส่วนของยูสเซอร์เนมและรหัสผ่านเท่านั้น เมื่อมีผู้ใช้งานในระบบนั้น telnet เข้ามาในระบบนั้น ก็สามารถดักจับแอคเคาน์ของระบบนั้นได้ และถ้ามีผู้ใช้งานคนใด telnet หรือ ftp ออกไปยังระบบอื่นๆ ก็จะได้แอคเคาน์ของระบบนั้นอีกเช่นกัน

สำหรับการตรวจสอบโปรแกรม sniffer นั้นทำได้โดยการตรวจสอบโหมดการทำงานของเน็ตเวิร์คการ์ด เนื่องจากโปรแกรม sniffer จะมีการเซตค่าแฟลกในเน็ตเวิร์คการ์ดให้ทำงานใน promiscuous mode ซึ่งทำให้เราสามารถตรวจสอบการติดตั้งโปรแกรม sniffer ได้โดยตรวจสอบจากแฟลกดังกล่าว คำสั่งที่ใช้ตรวจสอบ คือ คำสั่ง ifconfig ซึ่งวิธีนี้ก็มีข้อจำกัดอยู่ คือ ใช้กับ sniffer ที่มีการเซต promiscuous mode เท่านั้น

```
[root@isag07 root]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:C0:26:6D:8B:25
          inet addr:161.246.5.7  Bcast:161.246.5.255  Mask:255.255.255.0
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:3035 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:278011 (271.4 Kb)  TX bytes:420 (420.0 b)
          Interrupt:11 Base address:0x4000
```

รูปที่ 3-4 : แสดงคำสั่ง ifconfig

จากตัวอย่าง ถ้าเครื่องไม่มีการรันโปรแกรม sniffer จะไม่มีคำวา promisc อยู่ในผลลัพธ์ของคำสั่ง ซึ่งในกรณีนี้ก็ถึงเวลาที่ผู้ดูแลระบบจะต้องตรวจสอบระบบอย่างละเอียด

- ติดตั้งลีนุกซ์เคอร์เนลโมดูล (Linux Kernel Module)

เนื่องจากระบบยูนิกซ์จะให้สิทธิเต็มที่แก่ผู้ดูแลระบบให้สามารถติดตั้งหรือทำการเปลี่ยนแปลงทุกๆ อย่างในระบบ ดังนั้นผู้บุกรุกที่สามารถเปลี่ยนสิทธิให้เทียบเท่ากับผู้ดูแลระบบ ก็สามารถเปลี่ยนแปลงสิ่งต่างๆ ในระบบได้ทั้งหมดเช่นกัน แม้แต่การถอดโมดูลออกจากระบบ หรือการเปลี่ยนแปลงโมดูลต่างๆ ในระบบ สำหรับโมดูลนั้นคือส่วนการทำงานของเคอร์เนลที่แยกออกเป็นอิสระ สามารถประกอบเข้าในเคอร์เนลเพื่อเพิ่มความสามารถของเคอร์เนลและสามารถถอดออกจากเคอร์เนลได้ การตรวจสอบว่าในระบบมีเคอร์เนลอะไรบ้างนั้น สามารถตรวจสอบได้โดยใช้คำสั่ง lsmod

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
[root@isag07 root]# lsmod
Module                  Size  Used by    Not tainted
nls_iso8859-1           3516   1  (autoclean)
nls_cp437               5116   1  (autoclean)
vfat                   13084   1  (autoclean)
fat                    38712   0  (autoclean) [vfat]
i810_audio             25224   1  (autoclean)
ac97_codec             13384   0  (autoclean) [i810_audio]
soundcore               6532   2  (autoclean) [i810_audio]
autofs                 13348   0  (autoclean) (unused)
8139too                17704   1
mii                     2156   0  [8139too]
ipt_REJECT              3736   2  (autoclean)
iptable_filter          2412   1  (autoclean)
ip_tables              14936   2  [ipt_REJECT iptable_filter]
mousedev                5524   1
keybdev                 2976   0  (unused)
hid                    22244   0  (unused)
input                   5888   0  [mousedev keybdev hid]
usb-ohci               21320   0  (unused)
usbcore                77024   1  [hid usb-ohci]
ext3                   70368   2
jbd                    52212   2  [ext3]
[root@isag07 root]#
```

รูปที่ 3-5 : แสดงผล lsmod

จากตัวอย่างในระบบนี้จะมีการใช้งาน 2 โมดูล คือ rtl8139 และ pci-scan ซึ่งเป็นโมดูลในการใช้งานเน็ตเวิร์คการ์ดที่เป็น rtl ถ้าถอดโมดูลดังกล่าวออกจะไม่สามารถใช้เน็ตเวิร์คการ์ดได้ เพราะว่าเคอร์เนลจะไม่สามารถทำงานกับอุปกรณ์ได้ ถ้าผู้บุกรุกสามารถสร้างโมดูลแล้วประกอบเข้าในเคอร์เนลด้วยคำสั่ง insmod ได้จะเป็นปัญหาอย่างมากเพราะจะถือว่าผู้บุกรุกไม่ได้สิทธิในข้อมูลเท่านั้น ยังได้สิทธิในระบบนั้นทั้งหมดเลย มีความสามารถในการเปลี่ยนแปลงการทำงานในระดับ system call ได้เลย สามารถซ่อนโพรเซส ซ่อนไฟล์ ซ่อนการเชื่อมต่อ ทำให้ไม่สามารถเห็นการทำงานใดๆ ของผู้บุกรุกได้เลยโดยสมบูรณ์

การตรวจสอบสามารถตรวจสอบได้โดยผู้ดูแลระบบต้องทราบว่าโมดูลต่างๆ ในระบบมีอะไรบ้าง และทำงานอะไรได้บ้าง ตั้งแต่ติดตั้งระบบหรือเปรียบเทียบผลลัพธ์ของคำสั่ง ismod กับข้อมูลใน /proc/modules ว่าเหมือนกันหรือไม่ ถ้ามีการเพิ่มเติมโมดูลเข้าไปในระบบข้อมูลทั้งสองจะไม่เหมือนกัน โดยผลลัพธ์ของคำสั่ง ismod จะมีรายชื่อของ module มากกว่าใน /proc/modules แต่สำหรับวิธีนี้ผู้บุกรุกมักจะลบคำสั่ง ismod ออกจากระบบไปเพื่อป้องกันการตรวจสอบดังกล่าว

3.7 ระบบรักษาความปลอดภัย

เนื่องจากการบุกรุกก่อให้เกิดความเสียหายกับระบบแล้ว ดังนั้นเราควรจะมีระบบรักษาความปลอดภัยเป็นตัวป้องกันด้วย

3.7.1 ความหมายของความปลอดภัยข้อมูล

วัตถุประสงค์หรือเป้าหมายของการรักษาความปลอดภัยระบบใดๆ ก็ตามไม่เฉพาะในระบบลีนุกซ์เท่านั้น หากรักษาคุณสมบัติหลักได้ดังนี้ นับได้ว่าระบบรักษาความปลอดภัยนั้นประสบความสำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้นำไปใช้

1. รักษาสิ่งต่างๆ เป็นความลับ (Security) ซึ่งรวมถึงทุกสิ่งทุกอย่างในระบบ เช่น อุปกรณ์ต่างๆ หรือแฟ้มข้อมูล จะสามารถเข้าถึงได้เฉพาะบุคคลที่ได้รับสิทธิเท่านั้น
2. รักษาความปลอดภัยได้ง่าย (Integrity) ข้อมูลต่างๆ จะมีการแก้ไขหรือเปลี่ยนแปลงได้เพียงบุคคลที่ได้รับสิทธิเท่านั้น ผู้อื่นจะทำการเปลี่ยนแปลงแก้ไขไม่ได้
3. รักษาความสมดุล (Availability) ข้อมูลและอุปกรณ์ต่างๆ ในระบบจะต้องเข้าถึงได้เสมอ เมื่อเราต้องการใช้ในเงื่อนไขว่าเราเป็นผู้มีสิทธิในอุปกรณ์และข้อมูลนั้นๆ

3.7.2 ความปลอดภัยของข้อมูลต้องคำนึงถึงอะไรบ้าง

การออกแบบรักษาความปลอดภัยข้อมูลเป็นสิ่งที่ยากมาก เพราะหากการควบคุมอย่างเข้มงวดจนเกินไป มีข้อจำกัดจุกจิกมากมายจนทำให้การใช้งานเหมือนเป็นเรื่องที่ยุ่งยากวุ่นวาย ผู้ใช้งานจะทำอะไรต้องมีความระมัดระวังไม่ให้ไปขัดต่อข้อกำหนด คงจะลำบากใจพอดูทีเดียว ดังนั้นการรักษาความปลอดภัยข้อมูลที่ดีต้องคำนึงถึงหลักการต่างๆ มากพอควร

1. ระบบรักษาความปลอดภัยต้องง่ายและตรงไปตรงมา
2. ผู้ใช้งานทุกคนในระบบจะได้รับสิทธิน้อยที่สุดเท่าที่พอเพียงในการทำงานนั้น เพื่อป้องกันไม่ให้ไปทำลายในส่วนอื่นโดยไม่ได้ตั้งใจ
3. มีการตรวจสอบของระบบอย่างรอบคอบในการเข้าถึงหรือการทำงานทุกอย่าง
4. กำหนดสิทธิแบบเจาะจงว่า มีสิทธิใดบ้างในข้อมูลและอุปกรณ์ต่างๆ
5. กลไกรักษาความปลอดภัยต้องใช้ง่าย เพราะถ้าหากยุ่งยากแล้วผู้ใช้งานส่วนใหญ่จะละเลยเสียเอง

3.8 การทำงานควบคุมเพื่อให้เกิดความปลอดภัยในระบบ

เราสามารถควบคุมดูแลระบบให้เกิดความปลอดภัยในระบบได้โดยการ

3.8.1 การควบคุมดูแลทั้งทางด้านกายภาพและตัวของระบบเอง

ทางด้านกายภาพ เป็นการป้องกันโดยที่ไม่ให้ผู้ที่จะมาทำลายความปลอดภัยข้อมูลสามารถทำได้ เช่น

- ระวังบุคคลแปลกปลอมเข้ามาใช้งานระบบ โดยอาจจะมีการจัดเวรยามดูแลเพื่อป้องกันการใช้งาน และการก่อให้เกิดความเสียหายในกรณีใดๆ ก็ตาม
- มีการล็อกอุปกรณ์ที่สำคัญที่ใช้งานในระบบ
- สำรองข้อมูลสำคัญไว้ในอุปกรณ์ภายนอก เช่น เทป ดิสก์ แล้วทำการลบข้อมูลที่สำคัญหรือเป็นความลับออกจากระบบ
- มีการป้องกันการเปลี่ยนแปลงในส่วนของ BIOS
- ป้องกันในส่วนของการบูตระบบจากอุปกรณ์อื่น เช่น ดิสก์ หรือ ซีดี ในการเริ่มต้นระบบจากอุปกรณ์พิเศษนั้น ระบบจะอยู่ในโหมดของผู้ใช้งานคนเดียวหรือผู้ดูแลระบบ หากเป็นเช่นนี้แล้วไม่ว่าเป็นบุคคลใดก็ตามที่สามารถบูตระบบขึ้นใหม่ได้ก็จะมีสิทธิเท่ากับผู้ดูแลระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทางด้านภายในตัวของระบบเอง เช่น

- การสร้างผู้ใช้งานใหม่ ต้องมีสิทธิต่ำสุดเท่าที่สามารถใช้งานได้ มีกำหนดช่วงเวลาและสถานที่การใช้งานและลบรายชื่อออกจากระบบในกรณีที่ไม่มีการใช้งานจริง
- หลีกเลี่ยงการใช้งานโดยใช้ชื่อรูต เนื่องจากรูตมีสิทธิสูงสุดในระบบ ควรใช้เมื่อจำเป็นและเป็นช่วงระยะเวลาสั้นๆ เท่านั้น และไม่ควรใช้คำสั่งในชุดคำสั่ง r (r command) เนื่องจากเป็นการเสี่ยงต่อการถูกรุกรานได้ และการใช้ชื่อรูตในการใช้งานอาจเกิดผลกระทบกับระบบได้เพราะสิทธิของรูตนั้นสามารถทำได้ทุกอย่างกับระบบ

3.8.2 อ่านและวิเคราะห์ข้อมูลในล็อกไฟล์อย่างสม่ำเสมอ

ล็อกไฟล์เป็นไฟล์ข้อมูลที่เก็บรายละเอียดของการทำงานต่างๆ ในระบบ ไม่ว่าจะเป็นของระบบปฏิบัติการ หรือของแอพพลิเคชัน ซึ่งจะเป็นประโยชน์ต่อผู้ดูแลระบบในการตรวจสอบระบบว่ามีความปกติ หรือความผิดปกติอย่างไร เนื่องจากการทำงานต่างๆ ในระบบนั้นเราไม่สามารถทราบได้ด้วยตาเปล่าเลยว่า ระบบมีการเปลี่ยนแปลงหรือมีการบุกรุกอย่างไรบ้าง ดังนั้นผู้ดูแลระบบควรหมั่นตรวจสอบล็อกไฟล์อยู่เสมอ โดยระบบปฏิบัติการนี้จะมีล็อกไฟล์ชื่อ syslog ซึ่งส่วนใหญ่จะอยู่ในไดเรกทอรี var แต่ในบางระบบก็จะอยู่ในไดเรกทอรีอื่นๆ ส่วนล็อกไฟล์ของแอพพลิเคชันนั้น ก็จะมีชื่อไฟล์และตำแหน่งที่เก็บไฟล์ต่างๆ กันออกไป เช่น Apache ก็จะมีไฟล์ชื่อ access_log เป็นไฟล์ที่เก็บ URL ที่ถูกเรียกใช้งานจาก web client, Sendmail ซึ่งทำหน้าที่เป็นเมล์เซิร์ฟเวอร์ก็จะมีล็อกไฟล์ชื่อ mail.log เพื่อเก็บการทำงานของเมล์เซิร์ฟเวอร์เช่นกัน

การอ่านล็อกไฟล์ นอกจากจะทำให้ทราบการทำงานที่ผิดปกติในระบบแล้ว ยังสามารถตรวจจับการบุกรุกได้อีกด้วย ยกตัวอย่างเช่น การค้นหาเครื่องที่ติดไวรัส nimda เราก็สามารถดูได้จากล็อกไฟล์ของเวปเซิร์ฟเวอร์ได้เช่นกัน ยกตัวอย่าง ข้อมูลใน access_log ของ apache

```
161.246.34.247 - - [09/Feb/2001:17:07:36 +0700] "GET /scripts/..%c1%1c../winnt/system32/cmd.exe?/c+dir+c:\ H
TTP/1.0" 404 310
161.246.34.247 - - [09/Feb/2001:17:07:36 +0700] "GET /scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir+c:\ H
TTP/1.0" 404 310
161.246.34.247 - - [09/Feb/2001:17:07:36 +0700] "GET /scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir+c:\ H
TTP/1.0" 404 310
```

รูปที่ 3-6 : ข้อมูลในไฟล์ access_log ของ Apache

เราจะเห็นว่ามีข้อมูลการ GET โดยป้อน path ที่มี Unicode ที่อาจจะเป็นปัญหาในระบบอยู่ ซึ่งในลักษณะนี้ เครื่อง 161.246.34.247 น่าจะเป็นเครื่องที่ติด nimda มากกว่าการพยายามเข้ามาบุกรุก ในบางกรณีเราจะเห็นสัญญาณการบุกรุกอย่างชัดเจนจากล็อกไฟล์เช่นในตัวอย่างของ mail.log ของ sendmail

เอกสารนี้มีลิขสิทธิ์ ห้ามนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Jan  6 21:00:43 diamond sendmail[16085]: NOQUEUE: isag25.ce.kmitl.ac.th [161.246.5.25]: vrfy thanumch [rejected]
Jan  6 21:01:02 diamond sendmail[16085]: NOQUEUE: isag25.ce.kmitl.ac.th [161.246.5.25]: vrfy s4811056@ [rejected]
```

รูปที่ 3-7 : ข้อมูลในไฟล์ mail.log ของ sendmail

ในตัวอย่างนี้เราได้ตั้งค่า privacy option ไม่ให้ใช้คำสั่ง vrfy แต่มีความพยายามที่จะใช้งานเพื่อหาข้อมูลของผู้ใช้งานแต่ละคนจากเครื่อง diamond ซึ่งผู้ดูแลระบบสามารถตรวจสอบได้ว่าใครคือผู้ต้องสงสัยได้จากวัน และเวลาขณะทำงาน โดยตรวจสอบกับผู้ดูแลระบบของเครื่อง isag25 ว่ามีใครใช้งานอยู่และมีการเชื่อมต่อเข้ามายังเครื่อง diamond นอกจากจะดูสัญญาณการบุกรุกได้จากล็อกไฟล์แต่ละบรรทัดแล้ว ลักษณะโดยรวมของล็อกไฟล์ก็ยังให้ข้อมูลได้อีกเช่นกัน ถ้ามีจำนวนการเชื่อมต่อมากๆ ในเวลาไล่เลี่ยกัน หรือติดกันมากๆ แล้วมันจะเป็นสัญญาณที่บอกได้ว่า เกิดความพยายามในการบุกรุกระบบแล้ว ยกตัวอย่างเช่น ความพยายามในการค้นหารหัสผ่านของผู้ใช้งานบางคนในระบบ โดยใช้วิธการสุ่มรหัสผ่านทีละคำ เป็นต้น

3.8.3 รักษาความปลอดภัยไฟล์และระบบไฟล์ โดยการ

- กำหนดค่า umask (Umask Setting)
- กำหนดสิทธิของไฟล์ (File Permission)

ลักษณะไฟล์ข้อมูลของลีนุกซ์จะมีลักษณะพิเศษ คือ มีส่วนควบคุมในการใช้งานไฟล์ โดยการเก็บค่าสิทธิการใช้งานไฟล์แต่ละไฟล์ไว้ ส่วนนี้จะเป็นส่วนที่ตัดสินใจว่าผู้ใช้งานคนใดหรือกลุ่มใดสามารถดำเนินการกับไฟล์ใดได้บ้าง และทำได้มากน้อยเพียงใด โดยแบ่งการจัดการเป็นการอนุญาตให้อ่าน (read) เขียน (write) และทำงาน (execute) โดยมีการแบ่งพวกของผู้ใช้งานเป็น 3 กลุ่ม คือ เจ้าของ (owner) พวกเดียวกับเจ้าของ (group) และผู้ใช้งานผู้อื่น (other)

การเปลี่ยนแปลงกลุ่มและเจ้าของไฟล์นั้น สร้างขึ้นเพื่ออำนวยความสะดวกในการจัดการไฟล์ร่วมกันของผู้ใช้งานที่ต้องการใช้ไฟล์เดียวกัน โดยให้สามารถเปลี่ยนกลุ่มและเปลี่ยนเจ้าของได้โดยใช้คำสั่ง chgrp และ chown

3.8.4 ติดตั้งโปรแกรมและเปิดพอร์ตเฉพาะที่จำเป็นเท่านั้น

ก่อนการติดตั้งระบบปฏิบัติการควรวางแผนก่อนว่าจะต้องใช้แอพพลิเคชันอะไรบ้าง แล้วติดตั้งแอพพลิเคชันตามที่วางแผนไว้เท่านั้น เพื่อลดจำนวนจุดอ่อนในระบบที่อาจจะเกิดขึ้นได้จากแอพพลิเคชันที่ติดตั้งไป รวมถึงการเปิดพอร์ตจะต้องเปิดเฉพาะพอร์ตที่จำเป็นต้องเปิดเท่านั้น เพื่อลดช่องทางในการบุกรุกเข้าสู่ระบบ

วิธีการตรวจสอบว่าพอร์ตใดเปิดอยู่บ้างจะใช้โปรแกรมสแกนพอร์ต อย่างเช่น nmap ทำการสแกนหา ถ้าพบว่าพอร์ตใดเปิดอยู่แล้วไม่ได้ใช้งานก็ควรปิดพอร์ตนั้นเสีย โดยพอร์ตที่เปิดอยู่ทุกพอร์ตจะมีความสัมพันธ์กับโปรแกรมที่ผู้ดูแลระบบเรียกให้ทำงาน ซึ่งสามารถเรียกผ่าน command line หรือผ่านไฟล์ /etc/inetd.conf ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#<service_name><sock_type><proto><flags><user><server_path><args>

time  stream  tcp  nowait  root  internal
time  dgram  udp  wait  root  internal
ftp  stream  tcp  nowait  root  /usr/sbin/tcpd  wu.ftpd -l -i -a
telnet  stream  tcp  nowait  root  /usr/sbin/tcpd in.telnetd
```

รูปที่ 3-8 : ข้อมูลภายในไฟล์ /etc/inetd.conf

3.8.5 ตั้งค่าต่างๆ ในระบบและแอพพลิเคชันให้ปลอดภัย

สำหรับ service configuration ที่ควรพิจารณาเป็นอันดับต้นๆ คือ

1. ถ้าไม่จำเป็น ไม่ควรให้รูตเป็นผู้รันแอพพลิเคชันใดๆ ยกตัวอย่างเช่น ในโปรแกรม sendmail, apache, mysql ถ้าไม่จำเป็นไม่ควรใช้รูตเป็นผู้รันโปรแกรมโดยตรงเนื่องจากโพรเซสที่รันจะมีสิทธิตามผู้ที่รัน ถ้าโปรแกรมมีปัญหา ผู้บุกรุกจะสามารถเปลี่ยนสิทธิของตนเองให้กลายเป็นสิทธิของคนที่รันได้

2. ถ้าไม่จำเป็น ไม่ควรมี sbit ในโปรแกรมที่ใช้งานเนื่องจากเป็นบิตที่อันตรายมาก สามารถเปลี่ยนสิทธิของผู้ที่เรียกใช้โปรแกรมให้มีสิทธิเทียบเท่าเจ้าของ หรือกลุ่มเจ้าของของโปรแกรมนั้นๆ ได้

3. นอกจากการตั้งค่าให้ระบบทำงานได้แล้วนั้น ต้องพิจารณาการตั้งค่าเพื่อความปลอดภัยด้วย

4. ใช้งานโปรแกรมช่วยดูแลระบบและโปรแกรมช่วยรักษาความปลอดภัยในระบบ

3.8.6 การรักษาความปลอดภัยในส่วนของรหัสผ่านและการเข้ารหัส

- PGP เป็นการเข้ารหัสข้อมูลผ่านเครือข่ายโดยใช้ Public Key Crytography
- การเข้ารหัสไม่เพียงใช้เฉพาะในส่วนของไฟล์ที่เก็บรายชื่อและรหัสผ่านเท่านั้น ในลีนุกซ์มีความสามารถให้ผู้ใช้งานสามารถเข้ารหัสให้กับไฟล์ได้โดยใช้คำสั่ง crypt แล้วทำการใส่รหัสที่ต้องการ ซึ่งรหัสนี้จะเป็นส่วนที่ช่วยในการรักษาความปลอดภัยข้อมูลอีกระดับหนึ่ง
- SSH (Secure Shell) เป็นเชลชนิดหนึ่งที่มีการเข้ารหัสในการใช้งานทุกครั้งที่มีการติดต่อข้ามเครื่อง โดยใช้ Public Key Crytography
- Shadow Password ในระบบนี้ผู้ใช้งานทุกคนจะมีรายชื่อและรหัสผ่านโดยระบบจะมีผู้ใช้งานพิเศษคือ รูต ซึ่งสามารถทำการเปลี่ยนแปลงแก้ไขทุกอย่างที่อยู่ภายในระบบได้

นอกจากจะมีการตรวจสอบ ป้องกันควบคุมดูแลและติดตั้งระบบเป็นอย่างดีแล้ว ควรมีเครื่องมือช่วยในการระบบอย่างมีประสิทธิภาพด้วย เช่น

- Tcpwrapper

การทำ tcpwrapper เป็นการตั้งเงื่อนไขในไฟล์ชื่อ hosts.deny หรือ hosts.allow โดยสามารถตั้งเงื่อนไขเพื่ออนุญาตให้โฮสใดบ้างสามารถใช้งาน หรือไม่ให้ใช้งานเซอร์วิสใน /etc/inetd.conf ได้โดยเรา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้องระบุเซอร์วิสที่จะตั้งค่า เช่น “in.telnetd” หรือ “Win.ftpd” และระบุไอพี . โดเมนของเครื่อง เช่น Ohh.kmitl.ac.th หรือ เครือข่าย เช่น 161.246.5.7 ก็ได้ ยกตัวอย่างเช่น

```
host.deny:
        ALL:ALL
host.allow:
        intelnetd:.kmitl.ac.th
        ALL:.ce.kmitl.ac.th
```

รูปที่ 3-9 : การสร้างเงื่อนไขไฟล์ hosts.deny หรือ hosts.allow

ถ้ามีการตั้งค่าในลักษณะนี้ จะทำให้เครื่องปฏิเสธการเชื่อมต่อทุกๆ อย่าง แต่จะอนุญาตให้ telnet จากโดเมน .kmitl.ac.th ได้ และอนุญาตให้เชื่อมต่อจากโดเมน .ce.kmitl.ac.th เท่านั้น

นอกจากนี้ก็อาจใช้งานโปรแกรมช่วยรักษาความปลอดภัยอื่นๆ เช่น โปรแกรมที่ทำหน้าที่เป็น firewall ให้กับระบบบอย่าง iptables หรือโปรแกรมที่ใช้ตรวจสอบความปลอดภัยอื่นๆ มาใช้งานร่วมด้วย เพื่อเพิ่มความปลอดภัยในระบบให้มากขึ้น

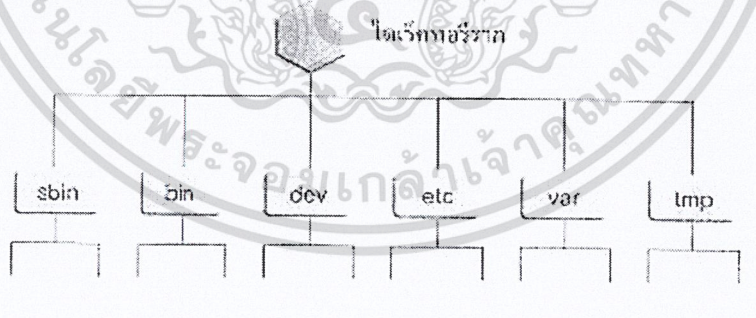
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ระบบและประเภทของไฟล์

4.1 ระบบโครงสร้างของไฟล์

ในระบบยูนิกซ์นั้นก็จะมีส่วนที่ประกอบกันขึ้นมาเป็นระบบโครงสร้างไฟล์เหมือนกับระบบปฏิบัติการอื่นๆ คือ จะประกอบด้วยไฟล์และไดเรกทอรี แต่เนื่องจากระบบปฏิบัติการยูนิกซ์นี้เป็นระบบที่สนับสนุนให้มีการทำงานพร้อมกันในระบบได้หรือที่เรียกว่า มัลติยูสเซอร์ (multiusers) ทำให้ต้องมีข้อมูลที่ใช้งานร่วมกันของผู้ใช้งานหลายๆ คน ระบบปฏิบัติการยูนิกซ์จึงมีการจัดการโครงสร้างที่ค่อนข้างเป็นระเบียบที่สุด เพื่อให้การทำงานเป็นไปได้อย่างมีประสิทธิภาพ

ในตอนแรกโครงสร้างของไฟล์และไดเรกทอรีนั้นเป็นแบบของ FSSTND (Linux File System Standard) แต่ต่อมาได้มีระบบหลายระบบนำรูปแบบการจัดการไฟล์และไดเรกทอรีแบบนี้ไปใช้และพัฒนาเป็นระบบโครงสร้างใหม่ที่เรียกว่า FHS (Filesystem Hierarchy Standard) แทน มีการจัดเป็นแบบลำดับชั้น ซึ่งมีลักษณะคล้ายกับต้นไม้กลับหัว โดยจุดเริ่มต้นหรือชั้นแรกจะเรียกว่าเป็นราก (root) จะเขียนแทนด้วยเครื่องหมาย / ไฟล์แต่ละไฟล์นั้นอาจจะเป็นข้อมูลที่สร้างขึ้น หรือโปรแกรมก็ได้ หรือแม้แต่ใช้เป็นที่เก็บไฟล์เองก็ได้ ซึ่งไฟล์ลักษณะนี้จะจัดว่าเป็นไฟล์ประเภทไดเรกทอรี รูปแบบการจัดเก็บไฟล์แบบลำดับชั้นนี้จะทำให้การจัดไฟล์เป็นระบบ ทำให้ง่ายต่อการใช้งานและดูแล โครงสร้างหลักของการจัดระบบไฟล์เป็นดังนี้



รูปที่ 4-1 : แสดงโครงสร้างของระบบปฏิบัติการยูนิกซ์เป็นรูปต้นไม้กลับหัว

จากรูปด้านบนจะเป็นรูปที่แสดงโครงสร้างระบบไฟล์ของยูนิกซ์ได้เป็นอย่างดี โดยบนสุดจะเรียกว่าไดเรกทอรีราก (Root Directory) ซึ่งภายในจะประกอบด้วยไดเรกทอรีและไฟล์อื่นอีกมากมาย สำหรับระบบยูนิกซ์แล้วจะมีบางไดเรกทอรีที่ถูกสร้างมาให้อยู่แล้ว และไดเรกทอรีเหล่านี้ก็ล้วนแต่มีหน้าที่สำคัญที่ได้รับมอบหมายมาแล้วว่าภายใต้ไดเรกทอรีแต่ละไดเรกทอรีเหล่านี้จะมีไว้เพื่อเก็บอะไร โดยมีรายละเอียดดังต่อไปนี้

เอกสารนี้เป็นการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

/ เป็นไดเรกทอรีรากที่ใช้เก็บไฟล์เคอร์เนล

/bin เป็นไดเรกทอรีที่ใช้เก็บคำสั่งทั่วไปของระบบ

/sbin เป็นไดเรกทอรีที่ใช้เก็บไฟล์โปรแกรมที่ถูกใช้โดยผู้ดูแลระบบหรือผู้ใช้งาน ไฟล์โปรแกรมที่จำเป็นต้องใช้ในระหว่างการบูตระบบ ไฟล์โปรแกรมที่ต้องใช้ในการเชื่อมต่อกับพาร์ทิชันต่างๆ ไฟล์โปรแกรมที่ต้องใช้ในการแก้ไขปัญหาต่างๆ ที่เกิดขึ้นกับระบบ

/etc เป็นไดเรกทอรีที่ใช้เก็บไฟล์กำหนดคุณสมบัติของเครื่องคอมพิวเตอร์ (Config File) และเก็บคำสั่งต่างๆ ของผู้ดูแลระบบ โดยปกติไดเรกทอรีจะไม่มีการเก็บไฟล์โปรแกรม

/lib เป็นไดเรกทอรีที่ใช้เก็บไฟล์ไลบรารี สำหรับไฟล์โปรแกรมต่างๆ ที่ต้องการใช้

/root เป็นไดเรกทอรีบ้านของผู้ดูแลระบบ

/dev เป็นไดเรกทอรีที่ใช้เก็บไฟล์ที่เกี่ยวกับอุปกรณ์ต่างๆ ที่ใช้ในระบบ

/tmp เป็นไดเรกทอรีที่ใช้เก็บไฟล์ชั่วคราว โดยจะจำลองตัวเองเหมือนหน่วยความจำเสมือนเก็บข้อมูลไว้ที่ไดเรกทอรีนี้ก่อนที่จะเก็บลงหน่วยความจำจริงๆ ทำให้มีการทำงานที่เร็วขึ้น แต่ต้องระวังเพราะว่าระบบจะมีการลบไฟล์ในไดเรกทอรีนี้ทิ้งทุกครั้งที่ระบบมีการรีสตาร์ทเครื่องใหม่ ฉะนั้นไฟล์ที่เก็บในนี้ควรเป็นไฟล์ชั่วคราวและไม่สำคัญมากนัก

/boot เป็นไดเรกทอรีที่ใช้เก็บไฟล์ที่ใช้โดย bootstrap loader อย่างเช่น LILO

/usr เป็นไดเรกทอรีที่ใช้เก็บไฟล์โปรแกรมที่ถูกใช้โดยผู้ใช้งานทั่วไป โดยปกติจะเป็นไฟล์โปรแกรมที่ถูกติดตั้งมากับระบบปฏิบัติการลีนุกซ์ ไม่ใช่ไฟล์โปรแกรมของแอปพลิเคชันที่ถูกติดตั้งพิเศษ

/mnt เป็นจุดที่ใช้สำหรับผู้ดูแลระบบ mount ไฟล์ซิสเต็ม หรืออุปกรณ์อย่างเช่น แผ่นดิสก์

/var เป็นที่เก็บไฟล์ต่างๆ ที่มักมีการเปลี่ยนแปลงขนาดไปตามเวลา เช่น ที่เก็บข้อมูลการส่งพิมพ์ของอุปกรณ์พรินเตอร์ ไฟล์ที่บันทึกการทำงาน หรือข้อผิดพลาดของระบบ เป็นต้น

/proc เป็นจุดเชื่อมต่อของไฟล์ซิสเต็มอื่น

/usr เป็นที่เก็บคำสั่งของระบบไว้ด้วยเหมือนกัน รวมทั้งยังเก็บไลบรารี และคู่มือช่วยเหลือ (man page) ไว้ด้วย

/home เป็นจุดเชื่อมต่อของไฟล์ซิสเต็มอื่น และมักใช้เป็นตำแหน่งของไดเรกทอรีบ้านของผู้ใช้งาน

4.2 ไอโหนด คืออะไร

เคอร์เนลจะไม่สามารถติดต่อหรือรู้จักไฟล์โดยตรงแต่เคอร์เนลจะรับรู้สิ่งต่างๆ เกี่ยวกับไฟล์ที่มีอยู่ได้จากโครงสร้างข้อมูลที่เรียกว่าไอโหนด (inode) โดยไอโหนดจะประกอบด้วยข้อมูลต่างๆ เกี่ยวกับไฟล์เป็นจำนวนมาก เช่น

- การแสดงสิทธิต่างๆ ของไฟล์
- การบอกถึงประเภทของไฟล์
- แสดงถึงเจ้าของ และกลุ่มของเจ้าของไฟล์
- วัน เวลา ที่สร้างไฟล์ ที่เปลี่ยนแปลงไฟล์ หรือแอ็กเซสไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- จำนวนลิงก์ที่ไฟล์เชื่อมต่อด้วย
- เก็บตำแหน่งที่อยู่ของข้อมูลในไฟล์

ไอโหนดนั้นจะถูกสร้างขึ้นมาโดยอัตโนมัติตอนสร้างระบบไฟล์ และถ้าหากไอโหนดนั้นเต็มจะไม่สามารถสร้างไฟล์ขึ้นมาได้เลย ถึงแม้ว่าจะยังมีเนื้อที่ว่างอยู่

4.3 ประเภทของไฟล์

ไฟล์ในระบบลีนุกซ์ได้มีการแบ่งประเภทออกไปหลายรูปแบบ คือ

- ไฟล์ทั่วไป (Regular File)
- ไดเรกทอรี (Directories)
- ไฟล์พิเศษ (Character and Block Device File)
- ไฟล์ที่เชื่อมต่อ (Link File)
- ไฟล์ไปป์ (Pipe File)

4.3.1 ไฟล์ทั่วไป (Regular File)

เป็นไฟล์ธรรมดาที่อาจสร้างขึ้นเองจากเอดิเตอร์ หรืออาจจะทำการคัดลอกมาจากไฟล์อื่น หรืออาจจะเป็นโปรแกรมใช้งานต่างๆ ก็ได้ โดยลักษณะการรายงานรายละเอียดต่างๆ ของไฟล์ เช่น สิทธิไฟล์ เจ้าของไฟล์ กลุ่มเจ้าของไฟล์ ขนาดหรือวันเวลาที่ทำการสร้าง เปลี่ยนแปลงแก้ไขไฟล์ ซึ่งจะได้ข้อมูลต่างๆ เหล่านี้มาจากไอโหนด ใช้สัญลักษณ์แทนด้วย " – "

```
[root@isag07 bigdir]# ls -la
total 12
drwxr-xr-x    2 root     root         4096 Jan 18 18:08 .
drwxr-xr-x    6 root     root         4096 Jan 18 18:08 ..
-rw-r--r--    1 root     root            2 Jan 18 18:08 bigfile
-rw-r--r--    1 root     root            0 Jan 18 18:07 picfile
-rw-r--r--    1 root     root            0 Jan 18 18:06 smallfile
[root@isag07 bigdir]#
```

รูปที่ 4-2 : แสดงรายละเอียดของไฟล์ทั่วไป

4.3.2 ไดเรกทอรี (Directories)

เป็นไฟล์ชนิดหนึ่งซึ่งใช้เก็บบรรดาไฟล์ต่างๆ หรือไดเรกทอรีด้วยกันเอง ปริมาณไฟล์ในระบบลีนุกซ์นั้นมีเป็นจำนวนมาก จึงความจำเป็นต้องมีรูปแบบการจัดเก็บไฟล์ให้อยู่ในไดเรกทอรีที่เหมาะสม เพื่อให้ผู้ใช้งานสามารถทำนายได้ว่าไฟล์ใดควรอยู่ในไดเรกทอรีไหน ใช้สัญลักษณ์แทนด้วย " d "

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
[root@isag07 panny]# ls -la
total 24
drwxr-xr-x    6 root     root         4096 Jan 18 18:08 .
drwxr-xr-x    3 root     root         4096 Jan 18 17:42 ..
drwxr-xr-x    2 root     root         4096 Jan 18 18:08 bigdir
drwxr-xr-x    2 root     root         4096 Jan 18 18:01 directory1
drwxr-xr-x    2 root     root         4096 Jan 18 18:04 directory2
drwxr-xr-x    2 root     root         4096 Jan 18 18:04 directory3
[root@isag07 panny]#
```

รูปที่ 4-3 : แสดงรายละเอียดของไดเรกทอรี

รูปแบบกรรมวิธีการอ้างถึงไฟล์ตามไดเรกทอรีนั้นกระทำโดยการอ่านค่าตำแหน่งจากไอโหนดว่าไฟล์ที่ต้องการนั้นอยู่ในบล็อกใด ระบบลีนุกซ์มีการแบ่งประเภทไดเรกทอรีให้เป็นไปตามหน้าที่ของกลุ่มไฟล์ที่อยู่ในไดเรกทอรีนั้นๆ จนลีนุกซ์จากหลายๆ ค่ายแทบจะมีการจัดเก็บไฟล์ต่างๆ ในไดเรกทอรีที่มีชื่อเหมือนๆ กัน หรือคล้ายๆ กัน

4.3.3 ไฟล์พิเศษ (Character and Block Device File)

ลีนุกซ์มีไฟล์ชนิดพิเศษอยู่ 2 ชนิด คือ แบบบล็อก (block device file) และแบบตัวอักษร (character device file) ใช้สัญลักษณ์แทนด้วย "b" หรือ "c" ตามลำดับ โดยลักษณะความต่างกันของแบบบล็อกกับแบบตัวอักษร คือ ลักษณะการรับส่งข้อมูล ซึ่งถ้าเป็นไฟล์แบบตัวอักษรจะมีการรับส่งทีละ 1 ตัวอักษร การส่งไฟล์แบบบล็อกนั้นจะมีการรับส่งข้อมูลทีละบล็อก ซึ่งอาจจะเป็น 512, 1024 หรือ 2048 ตัวอักษรก็ได้ ไฟล์พิเศษเหล่านี้จะอยู่ในไดเรกทอรี dev ซึ่งเป็นไดเรกทอรีที่เก็บอุปกรณ์ต่างๆ ของระบบ เช่น ฮาร์ดดิสก์, ดิสก์ไดรฟ์, จอภาพ, หน่วยความจำ เป็นต้น

```
[root@isag07 dev]# ls -la |more
total 360
drwxr-xr-x   20 root     root       118784 Jan 14 21:21 .
drwxr-xr-x   20 root     root         4096 Jan 14 21:21 ..
crw-------    1 root     root      10,  10 Aug 31 06:31 adbmouse
crw-r--r--    1 root     root      10, 175 Aug 31 06:31 agpgart
crw-------    1 root     root      10,   4 Aug 31 06:31 amigamouse
crw-------    1 root     root      10,   7 Aug 31 06:31 amigamouse1
crw-------    1 root     root      10, 134 Aug 31 06:31 apm_bios
drwxr-xr-x    2 root     root         4096 Oct 23 10:34 ataraid
crw-------    1 root     root      10,   5 Aug 31 06:31 atarimouse
crw-------    1 root     root      10,   3 Aug 31 06:31 atibm
crw-------    1 root     root      10,   3 Aug 31 06:31 atimouse
crw-------    1 root     root      14,   4 Aug 31 06:31 audio
crw-------    1 root     root      14,  20 Aug 31 06:31 audio1
crw-------    1 root     root      14,   7 Aug 31 06:31 audioctl
brw-rw----    1 root     disk      29,   0 Aug 31 06:31 aztcd
crw-------    1 root     root      10, 128 Aug 31 06:31 beep
brw-rw----    1 root     disk      41,   0 Aug 31 06:31 bpcd
crw-------    1 root     root      68,   0 Aug 31 06:31 capi20
crw-------    1 root     root      68,   1 Aug 31 06:31 capi20.00
crw-------    1 root     root      68,   2 Aug 31 06:31 capi20.01
crw-------    1 root     root      68,   3 Aug 31 06:31 capi20.02
--More--
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4-4 : แสดงรายละเอียดของไฟล์พิเศษทั้ง 2 ชนิด

การแสดงรายละเอียดของไฟล์พิเศษนี้ต่างไปจากไฟล์ทั่วไปและไดเรกทอรี คือ จะไม่บอกขนาดของไฟล์ แต่จะแสดงตัวเลข 2 แถวแทน โดยตัวเลขแต่ละตัวมีความหมายต่างกันไป ตัวเลขตัวแรกเป็นการบอกเคอร์เนลว่าไฟล์นั้นอ้างถึงดีไวซ์ไดรเวอร์ไหน ซึ่งตัวเลขนี้จะเรียกว่า หมายเลขหลัก (major number) ส่วนตัวเลขอีกหลักหนึ่งจะเป็นส่วนที่บอกกับดีไวซ์ไดรเวอร์ว่าไฟล์พิเศษนั้นอ้างถึงคุณสมบัติของอุปกรณ์นั้น (physical unit) ตัวเลขนี้เรียกว่าตัวเลขรอง (minor number)

```
brw-rw----    1 root     floppy     2,   0 Aug 31 06:31 fd0
brw-rw----    1 root     floppy     2,   4 Aug 31 06:31 fd0CompaQ
brw-rw----    1 root     floppy     2,   4 Aug 31 06:31 fd0d360
brw-rw----    1 root     floppy     2,  12 Aug 31 06:31 fd0D360

  brw-rw----    1 root     disk      22,  95 Aug 31 06:31 hdd31
  brw-rw----    1 root     disk      22,  96 Aug 31 06:31 hdd32
  brw-rw----    1 root     disk      22,  68 Aug 31 06:31 hdd4
  brw-rw----    1 root     disk      22,  69 Aug 31 06:31 hdd5
  brw-rw----    1 root     disk      22,  70 Aug 31 06:31 hdd6

crw-rw----    1 root     lp        99,   4 Aug 31 06:31 parport4
crw-rw----    1 root     lp        99,   5 Aug 31 06:31 parport5
crw-rw----    1 root     lp        99,   6 Aug 31 06:31 parport6
crw-rw----    1 root     lp        99,   7 Aug 31 06:31 parport7
```

รูปที่ 4-5 : แสดงรายละเอียดของไฟล์พิเศษในแบบต่างๆ เช่น ฟลอปปี้ดิสก์ ดิสก์ พรินเตอร์ ตามลำดับ

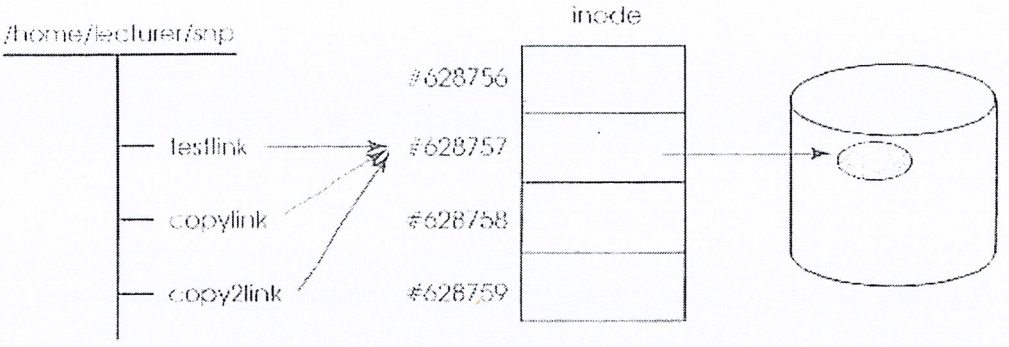
จากตัวอย่างจะเห็นได้ว่า ไม่ว่าจะเป็นฟลอปปี้ดิสก์ ดิสก์ หรือพรินเตอร์ ก็ตามจะมีการอ้างถึงหลายไฟล์ โดยมีการกำหนดหมายเลขหลักเหมือนกันจะต่างกันที่หมายเลขรอง ซึ่งหมายเลขรองนี้จะเป็นการระบุหรือชี้ถึงคุณสมบัติของอุปกรณ์ เช่นว่าเป็นฟลอปปี้ดิสก์ชนิดความหนาแน่นสูงหรือเป็นแบบฟลอปปี้ดิสก์ชนิดความหนาแน่นต่ำ เป็นต้น

4.3.4 ไฟล์ที่เชื่อมต่อ (Link File)

ลักษณะการเชื่อมต่อไฟล์นั้นมีอยู่ 2 แบบ คือ การเชื่อมต่อทางกายภาพ (hard link) กับการเชื่อมต่อโดยนัย (symbolic link)

1. การเชื่อมต่อทางกายภาพ (hard link) การเชื่อมต่อแบบนี้จะมีการใช้ไอโหนดเดียวกันกับไฟล์ต้นฉบับเสมือนกับการสร้างไฟล์ใหม่ขึ้นมา เพียงแต่ค่าไอโหนดนั้นจะซ้ำกับของเดิมและไอโหนดนับจำนวนไฟล์ที่เชื่อมต่อด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-6 : ลักษณะของไฟล์เชื่อมต่อทางกายภาพ

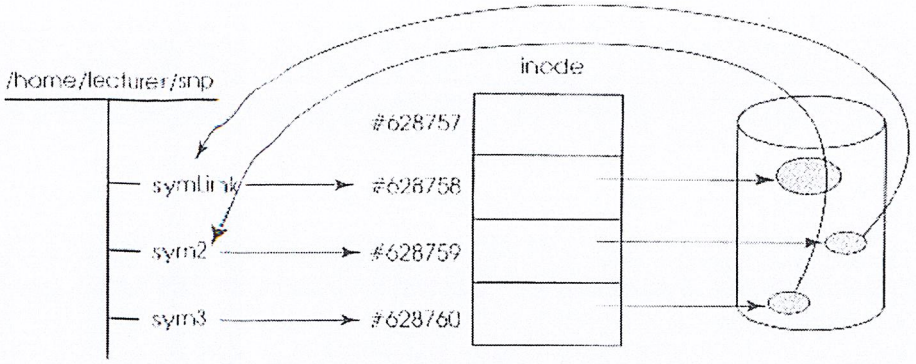
หากมีการเปลี่ยนแปลงข้อมูลภายในไฟล์หนึ่งก็จะส่งผลกระทบถึงกันเพราะข้อมูลเก็บที่เดียวกัน วิธีการนี้เหมาะกับการเชื่อมต่อไฟล์ต่างๆ ที่เก็บอยู่ในพาร์ทิชันเดียวกัน ทำให้ประหยัดเนื้อที่ในการเก็บข้อมูลสามารถอ้างถึงได้จากหลายๆ ที่ไม่สามารถใช้งานได้กับการเชื่อมต่อไฟล์ที่อยู่ต่างพาร์ทิชันกัน

```
[root@isag07 project]# cat > testlink
Testing to create a hard link.
[root@isag07 project]# ls -il *link
 261470 -rw-r--r--    1 root     root                31 Jan 15 02:52 testlink
[root@isag07 project]# ln testlink copylink
[root@isag07 project]# ls -il *link
 261470 -rw-r--r--    2 root     root                31 Jan 15 02:52 copylink
 261470 -rw-r--r--    2 root     root                31 Jan 15 02:52 testlink
[root@isag07 project]# ln testlink copy2link
[root@isag07 project]# ls -il *link
 261470 -rw-r--r--    3 root     root                31 Jan 15 02:52 copy2link
 261470 -rw-r--r--    3 root     root                31 Jan 15 02:52 copylink
 261470 -rw-r--r--    3 root     root                31 Jan 15 02:52 testlink
[root@isag07 project]# rm testlink
rm: remove regular file `testlink'? y
[root@isag07 project]# ls -il *link
 261470 -rw-r--r--    2 root     root                31 Jan 15 02:52 copy2link
 261470 -rw-r--r--    2 root     root                31 Jan 15 02:52 copylink
[root@isag07 project]# cat copy2link
Testing to create a hard link.
[root@isag07 project]#
```

รูปที่ 4-7 : แสดงการสร้างไฟล์แบบ hard link

2. การเชื่อมต่อโดยนัย (symbolic link) เป็นลักษณะการเชื่อมต่อเช่นเดียวกัน แต่จะมีการสร้างไอโหนดของไฟล์ตัวเองขึ้นใหม่ โดยไฟล์ใหม่ที่ทำการสร้างขึ้นจะเก็บไดเรกทอรีและชื่อไฟล์ที่ต้องการเชื่อมต่อด้วย เช่นเดียวกับ Shortcut ในระบบของ MS Windows แต่หากมีการเปลี่ยนแปลงที่ไฟล์ต้นฉบับจะส่งผลกระทบต่อไฟล์ที่เชื่อมต่อด้วย ใช้สัญลักษณ์แทนด้วย “ l ”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-8 : ลักษณะของไฟล์เชื่อมต่อโดยนัย

แต่ถ้าหากลบไฟล์ที่เชื่อมจะไม่ส่งผลกระทบต่อไฟล์ต้นฉบับ ในทางกลับกันหากลบไฟล์ต้นฉบับแล้ว ไฟล์ที่เชื่อมต่อจะไม่สามารถเรียกใช้งานได้ เนื่องจากไม่มีข้อมูลอยู่จริง วิธีการนี้สามารถใช้เชื่อมต่อไฟล์ต่างๆ ที่เก็บอยู่ในพาร์ทิชันเดียวกันหรือต่างพาร์ทิชันกันได้

```
[root@isag07 project]# cat > symfile
This is a file for testing Symbolic Link.
[root@isag07 project]# ls -il sym*
 261471 -rw-r--r--    1 root     root           42 Jan 15 03:07 symfile
[root@isag07 project]# ln -s symfile sym2
[root@isag07 project]# ls -il sym*
 261472 lrwxrwxrwx    1 root     root            7 Jan 15 03:08 sym2 -> symfile
 261471 -rw-r--r--    1 root     root           42 Jan 15 03:07 symfile
[root@isag07 project]# cat sym2
This is a file for testing Symbolic Link.
[root@isag07 project]#
```

รูปที่ 4-9 : แสดงการสร้างไฟล์เชื่อมต่อโดยนัยชื่อ sym2 ชี้ไปยัง symfile

```
[root@isag07 project]# ln -s sym2 sym3
[root@isag07 project]# ls -il sym*
 261472 lrwxrwxrwx    1 root     root            7 Jan 15 03:08 sym2 -> symfile
 261473 lrwxrwxrwx    1 root     root            4 Jan 15 03:10 sym3 -> sym2
 261471 -rw-r--r--    1 root     root           42 Jan 15 03:07 symfile
[root@isag07 project]# cat sym3
This is a file for testing Symbolic Link.
[root@isag07 project]#
```

รูปที่ 4-10 : แสดงการสร้างไฟล์เชื่อมต่อโดยนัยชื่อ sym3 ชี้ไปยัง sym2

```
[root@isag07 project]# rm symfile
rm: remove regular file `symfile'? y
[root@isag07 project]# ls -il sym*
 261472 lrwxrwxrwx    1 root     root            7 Jan 15 03:08 sym2 -> symfile
 261473 lrwxrwxrwx    1 root     root            4 Jan 15 03:10 sym3 -> sym2
[root@isag07 project]# cat sym2
cat: sym2: No such file or directory
[root@isag07 project]# cat sym3
cat: sym3: No such file or directory
[root@isag07 project]#
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4-11 : แสดงการลบไฟล์เชื่อมต่อโดยนัยที่เป็นไฟล์ต้นฉบับที่ชื่อ symfile

4.4 การดำเนินการกับไฟล์

ในระบบลีนุกซ์และยูนิกซ์นั้นมีการดำเนินการเกี่ยวกับไฟล์เหมือนกัน รูปแบบการดำเนินการกับไฟล์ เช่น การสร้าง การทำสำเนา การเปลี่ยนชื่อ การย้ายตำแหน่ง การลบ และการอ้างถึงไฟล์

4.4.1 การสร้างไฟล์และไดเรกทอรี

การสร้างไฟล์นั้นสามารถดำเนินการได้หลายวิธี อย่างเช่น การสร้างไฟล์จากการเปลี่ยนทิศทางผลลัพธ์ การสร้างไฟล์จากเอดิเตอร์ สร้างจากคำสั่ง touch (ซึ่งเป็นไฟล์ที่มีขนาดเป็นศูนย์) และการทำสำเนาจากไฟล์ที่มีอยู่แล้วโดยใช้คำสั่ง cp สำหรับไดเรกทอรีนั้นสามารถสร้างได้โดยใช้คำสั่ง mkdir (make directory) ตามด้วยชื่อไดเรกทอรีใหม่ ส่วนการเปลี่ยนชื่อไฟล์หรือย้ายตำแหน่งเก็บไฟล์ใช้คำสั่ง mv (move)

```
[root@isag07 panny]# touch file1
[root@isag07 panny]# file file1
file1: empty
[root@isag07 panny]# cp file1 file2
[root@isag07 panny]# ls
bigdir  file1  file2
[root@isag07 panny]# mkdir directory1
[root@isag07 panny]# ls -la
total 16
drwxr-xr-x    4 root     root         4096 Jan 24 18:36 .
drwxr-xr-x    3 root     root         4096 Jan 18 17:42 ..
drwxr-xr-x    2 root     root         4096 Jan 18 18:08 bigdir
drwxr-xr-x    2 root     root         4096 Jan 24 18:36 directory1
-rw-r--r--    1 root     root            0 Jan 24 18:36 file1
-rw-r--r--    1 root     root            0 Jan 24 18:36 file2
[root@isag07 panny]# mv file1 fileone
[root@isag07 panny]# mv file2 directory1
[root@isag07 panny]# ls
bigdir  directory1  fileone
[root@isag07 panny]# cd directory1/
[root@isag07 directory1]# ls
file2
[root@isag07 directory1]#
```

รูปที่ 4-12 : การสร้างไฟล์และไดเรกทอรีด้วยวิธีต่างๆ

4.4.2 การลบไฟล์และไดเรกทอรี

การลบไฟล์สามารถกระทำได้ในกรณีที่ผู้ลบต้องมีสิทธิในการดำเนินการกับไฟล์นั้น ซึ่งสามารถลบได้โดยใช้คำสั่ง rm (remove) สำหรับการลบไดเรกทอรีที่ไม่ต้องการออกนั้นสามารถกระทำได้โดยการใช้คำสั่ง rmdir (remove directory) ซึ่งมีรูปแบบเดียวกันกับการสร้างไดเรกทอรีขึ้นใหม่ แต่การลบไดเรกทอรีด้วยคำสั่งนี้นั้นไดเรกทอรีที่ต้องการลบจะต้องเป็นไดเรกทอรีว่าง คือไม่มีไฟล์หรือไดเรกทอรีย่อยใดอยู่เลย แต่ถ้ามีอยู่ก็สามารถลบด้วยคำสั่ง rm –r เพื่อลบไดเรกทอรีย่อยหรือไฟล์ต่างๆ ที่มีอยู่ได้โดยอัตโนมัติ ดังนั้นการใช้คำสั่งนี้จะต้องใช้อย่างระมัดระวัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
[root@isag07 project]# ls
class            copylink    fileone        file_unowned  project.sh  test
code_unowned.sh  crack       file_suid      file_world    [redacted]  uolist
copy2link        directory1  file_symbolic  homework
[root@isag07 project]# rm fileone
rm: remove regular empty file `fileone'? y
[root@isag07 project]# ls
class            copylink    file_suid      file_world    [redacted]  uolist
code_unowned.sh  crack       file_symbolic  homework
copy2link        directory1  file_unowned   project.sh  test
[root@isag07 project]#
```

รูปที่ 4-13 : การลบไฟล์ fileone ออกจากระบบ

```
[root@isag07 project]# rmdir directory1/
rmdir: `directory1/': Directory not empty
[root@isag07 project]# rm -r directory1/
rm: descend into directory `directory1/'? y
rm: remove regular empty file `directory1//file2'? y
rm: remove directory `directory1/'? y
[root@isag07 project]# ls
class            copylink    file_symbolic  homework      [redacted]
code_unowned.sh  crack       file_unowned   project.sh  test
copy2link        file_suid   file_world     [redacted]  uolist
[root@isag07 project]#
```

รูปที่ 4-14 : การลบไดเรกทอรี directory1 ออกจากระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สิทธิการใช้งานไฟล์

5.1 สิทธิการใช้งานไฟล์ (File Permission)

ลีนุกซ์มองทุกสิ่งทุกอย่างในระบบเป็นไฟล์ ดังนั้นสิทธิการใช้งานของไฟล์จึงเป็นเรื่องสำคัญที่สุดต่อการใช้งาน และมักจะพบว่าเป็นความผิดพลาดเบื้องต้น ที่ผู้ดูแลระบบมองข้ามหรือไม่สนใจอยู่บ่อยครั้งทำให้ระบบเกิดจุดอ่อนได้ง่าย

คำสั่ง ls ใช้ในการแสดงไฟล์หรือไดเรกทอรีที่มีในระบบ สมมติตัวอย่างการใช้คำสั่ง ls -l เพื่อแสดงไฟล์ test ในระบบ

-rwxr–xr-- 1 ohho staff 852 Jul 9 10:10 test

รูปที่ 5-1 : แสดงรายละเอียดของไฟล์ test

ไฟล์ชื่อ test มีเจ้าของเป็น ohho และอยู่ในกลุ่มที่ชื่อ staff อธิบายถึงความหมายในฟิลด์ต่างๆ ได้ดังนี้

ฟิลด์	คำอธิบาย
-rwxr-xr--	สิทธิการใช้งานของไฟล์ (permission)
1	จำนวน hard link ของไฟล์นี้
ohho	ยูสเซอร์ที่เป็นเจ้าของไฟล์ (UID)
staff	กลุ่ม (GID)
852	ขนาดของไฟล์ หน่วยเป็นไบต์
Jul 9	วันที่มีการแก้ไขล่าสุด
10:10	เวลาที่มีการแก้ไขล่าสุด
test	ชื่อไฟล์

ตารางที่ 5-1 : ความหมายของฟิลด์เมื่อใช้คำสั่ง ls -l

การพิจารณาวันเวลาที่มีการบันทึกในไฟล์และไดเรกทอรีในระบบก็เป็นเรื่องสำคัญที่ไม่ควรมองข้าม ใช้ในการตรวจสอบหาลำดับเวลาเมื่อเกิดเหตุการณ์ผิดปกติกับระบบ ระบบการบันทึกวันและเวลาของไฟล์ใน ลีนุกซ์มีวันและเวลาในการบันทึกทั้งหมดสามรูปแบบคือ

เอกสารนี้เป็นลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบ	ชื่อย่อ	การใช้คำสั่ง ls	คำอธิบาย
Modify time	mtime	ls -l	วันเวลาที่มีการแก้ไขครั้งล่าสุด
Access time	atime	ls -lu	วันเวลาที่มีการแก้ไขวันและเวลาโดยใช้ system call ของระบบเช่น utime() เป็นต้น (โดยมากมักไม่ได้ใช้ แต่มักจะใช้ในการวิเคราะห์ไฟล์ในระดับลึกลงไป)
Change time	ctime	ls -lc	วันเวลาที่แก้ไขโดยระบบปฏิบัติการเมื่อมีการเปลี่ยนแปลงโครงสร้างของไฟล์ ซึ่งโดยปกติ ค่าจะไม่มีการเปลี่ยนแปลง

ตารางที่ 5-2 : แสดงคำสั่งเวลาที่มีการบันทึกในระบบ

ลีนุกซ์เป็นระบบปฏิบัติการที่มีการใช้งานไฟล์ต่างๆ ร่วมกันในกลุ่มของผู้ใช้งาน หากให้สิทธิแก่ผู้ใช้งานทั้งหมดในการจัดการหรือกระทำสิ่งใดๆ กับไฟล์ได้นั้นย่อมก่อให้เกิดความวุ่นวายและสับสนได้ ดังนั้นลีนุกซ์จึงได้มีการจำกัดสิทธิการใช้งานไฟล์โดยการกำหนดประเภทของผู้ใช้งานไฟล์ออกเป็น 3 กลุ่มใหญ่ๆ คือ

- เจ้าของไฟล์หรือผู้สร้างไฟล์ (owner)
- ผู้ใช้งานในกลุ่มเดียวกับเจ้าของไฟล์ (group)
- บุคคลอื่น (other or universe)

ซึ่งไฟล์เหล่านั้นจะประกอบไปด้วยสิทธิต่างๆ คือ

- สิทธิในการอ่านไฟล์ (read permission) “ r ” หรือ “ read ”
- สิทธิในการเขียนหรือแก้ไขเปลี่ยนแปลงไฟล์ (write permission) “ w ” หรือ “write ”
- สิทธิในการรันไฟล์ (execute permission) “ x ” หรือ “ execute ”

รูปแบบการแสดงสิทธิในไฟล์นั้นจะใช้ตัวเลข 9 บิต หรือเรียกว่า nine permission bits โดย 3 บิตแรกเป็นการแสดงสิทธิของเจ้าของไฟล์ 3 บิตถัดมานั้นเป็นการแสดงสิทธิของกลุ่มเจ้าของไฟล์และ 3 บิตสุดท้ายเป็นการแสดงสิทธิของบุคคลอื่น และถ้าไม่มีสิทธิในบิตนั้นจะแสดงด้วยสัญลักษณ์ขีด “ - ”
เช่น - r w - r - - r - - 1 ohho staff 852 Jul 9 10:10 test

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

ตารางที่ 5-3 : แสดงการแบ่งกลุ่มของเพอร์มิสชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในแต่ละส่วนอธิบายเพิ่มเติมได้ดังนี้

Description	คำอธิบาย
Type	บอกถึงประเภทของไฟล์ ซึ่งเป็นไฟล์ทั่วไป
Owner (user)	สิทธิในการเข้าถึงไฟล์ของเจ้าของไฟล์ ซึ่งในที่นี้อ่านและแก้ไขไฟล์ได้
Group	สิทธิในการเข้าถึงไฟล์ของผู้ใช้งานในระบบที่อยู่ในกลุ่มที่ระบุไว้ ในที่นี้อ่านไฟล์ได้เท่านั้น
Others	สิทธิในการเข้าถึงไฟล์ของทุกคนในระบบที่ไม่ใช่เจ้าของไฟล์และไม่ได้อยู่ในกลุ่ม เดียวกันกับ group ที่ระบุไว้ ซึ่งในที่นี้อ่านไฟล์ได้เท่านั้นเช่นกัน

ตารางที่ 5-4 : คำอธิบายเพิ่มเติมของสิทธิในการใช้งานของกลุ่มต่างๆ

ตัวอย่างสิทธิในการใช้งานต่างๆ

สิทธิในการใช้งาน	รหัส
r w x r w x r w x	0777
r w x r w x - - -	0770
r w x - - - - - -	0700
r w – r w – r w -	0666
r w – r w - - - -	0660
r w - - - - - - -	0600
r - - - - - - - -	0400

ตารางที่ 5-5 : ตัวอย่างของสิทธิในการใช้งาน

5.2 ความหมายของสิทธิต่างๆ

เมื่อนำเพอร์มิสชันทั้ง 3 มาใช้กับไฟล์ปกติทั่วไปและไดเรกทอรีแล้ว จะมีความหมายที่แตกต่างกันออกไป คือ

5.2.1 ความหมายของสิทธิในการเข้าถึงไฟล์ปกติ

ความหมายของสิทธิในการเข้าถึงไฟล์ปกติที่ไม่ใช่ไดเรกทอรีนั้น จะมีความหมายดังตารางข้างล่างนี้ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟล์		
Character	Permission	ความหมาย
R	READ	สามารถเปิดไฟล์และอ่านเนื้อความในไฟล์ได้
W	WRITE	สามารถเพิ่มข้อความ ลบข้อความ หรือเปลี่ยนแปลงเนื้อความในไฟล์ได้
X	EXECUTE	จะใช้ในกรณีที่ไฟล์นี้สามารถทำงานได้หรือสั่งให้ทำงานได้ ซึ่งจะมีสิทธิในการทำงานนี้ปรากฏอยู่

ตารางที่ 5-6 : ตารางแสดงความหมายของสิทธิต่างๆ ของไฟล์

5.2.2 ความหมายของสิทธิในการเข้าถึงไดเรกทอรี

เมื่อนำมาใช้กับไดเรกทอรีจะมีสิทธิต่างๆ ซึ่งมีความหมายที่แตกต่างกันดังนี้

ไดเรกทอรี		
Character	Permission	ความหมาย
R	READ	สามารถใช้คำสั่ง ls ในการดูไฟล์ภายในไดเรกทอรีได้
W	WRITE	สามารถเพิ่มไฟล์ ลบไฟล์ เปลี่ยนชื่อไฟล์ที่อยู่ในไดเรกทอรีได้
X	EXECUTE	สามารถใช้คำสั่ง cd ทำการเปลี่ยนไดเรกทอรีปัจจุบันเข้าไปได้ หรือใช้ในการเปิดไฟล์ภายในไดเรกทอรีรวมไปถึงไดเรกทอรีย่อยภายใน

ตารางที่ 5-7 : ตารางแสดงความหมายของสิทธิต่างๆ ของไดเรกทอรี

ตัวอย่างสิทธิขั้นต่ำในการใช้งานไฟล์

การใช้งาน	สิทธิขั้นต่ำ	ตัวอย่างการใช้งาน
พิมพ์ไฟล์	r	$ cat sample_ file
แก้ไขไฟล์	rw	$ vi sample_file
ลบไฟล์	w	$ rm simple_file
เพิ่มข้อมูล	w	$ cat >> simple_file
เรียกใช้โปรแกรมไบนารี	x	$ run binary
เรียกใช้สคริปต์	rw	$ run script

ตารางที่ 5-8 : แสดงสิทธิขั้นต่ำในการใช้งานในแบบต่างๆ

เอกสารนี้จัดทำขึ้นเพื่อการที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 ความสัมพันธ์ของ UID กับ username และ GID กับ groupname

พิจารณาไฟล์ /etc/passwd ที่ระบุชื่อผู้ใช้งานชื่อ ohho

ohho:x:100:200:ohho account:/home/ohho:/bin/bash

ฟิลด์	คำอธิบาย
ohho	ชื่อยูสเซอร์เนม คือ ohho
X	เป็นรหัสผ่านซึ่งจะเก็บค่าไว้ในไฟล์ /etc/shadow
100	เป็น User ID หรือ UID ซึ่งลีนุกซ์จะใช้ในการอ้างถึงเจ้าของไฟล์นั้นๆ
200	อ้างถึง GID ซึ่งลีนุกซ์ใช้ในการบอกกรุ๊ปของผู้ใช้งานคนนี้
ohho account	รายละเอียดอื่นๆ เพิ่มเติมของผู้ใช้งานชื่อ ohho
/home/ohho	เป็นโฮมไดเรกทอรีของผู้ใช้งาน ohho
/bin/bash	เชลหลักของผู้ใช้งาน

ตารางที่ 5-9 : แสดง รายละเอียดในไฟล์ /etc/passwd

และผู้ใช้งานที่ชื่อ ohho อยู่ในกรุ๊ป staff พิจารณาไฟล์ /etc/group

staff:x:200:

ฟิลด์	คำอธิบาย
staff	กรุ๊ปชื่อ staff
X	ใช้ในการระบุรหัสผ่านให้กับกรุ๊ป
200	เป็น Group ID หรือ GID

ตารางที่ 5-10 : ตารางแสดงรายละเอียดของไฟล์ /etc/groups

ผู้ใช้งานชื่อ ohho มี uid เป็น 100 และกรุ๊ปเป็น staff gid เป็น 200 ซึ่ง uid และ gid เป็นเลขจำนวนจริงขนาด 16 บิตมีค่าตั้งแต่ 0 จนถึง 65535 บางระบบอาจมี uid หรือ gid เป็นเลข 32 บิตได้ มนุษย์จะใช้ตัวอักษรในการอ้างถึง ส่วนลีนุกซ์ จะอ้างถึงเป็นตัวเลขเสมอ เป็นแนวความคิดที่ไม่ซับซ้อนและใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้มีประสิทธิภาพมาก โดย uid และ gid ของผู้ใช้งานจะกำหนดสิทธิในการเข้าถึงไฟล์ในระบบ โดยจับคู่กันระหว่างชื่อผู้ใช้งานกับ uid และ กรุ๊ป กับ gid

ถ้าผู้ใช้งานสองคนมี uid เหมือนกัน ลีนุกซ์จะถือว่าผู้ใช้งานสองคนนี้เป็นคนเดียวกันแม้ว่ายูสเซอร์เนมและรหัสผ่านจะต่างกันก็ตาม เนื่องมาจากเป็นคนเดียวกันในระบบ ทำให้มีสิทธิเหมือนกันทุกอย่าง เช่นสามารถลบไฟล์ของอีกคนหนึ่งก็ได้ สามารถหยุดโพรเซสของอีกคนหนึ่งได้ (ความหมายของคำว่าอีกคนหนึ่งในที่นี้คือใช้ยูสเซอร์เนมคนละชื่อ) ความไม่รอบคอบในลักษณะนี้อาจจะทำให้ระบบทำงานได้ไม่ถูกต้องและมีผลในแง่ความปลอดภัยของระบบ

ผู้ใช้งานในระบบที่ uid เป็น 0 คือ superuser หรือรูตมีสิทธิทุกอย่างในระบบ ซึ่งเป็นจุดประสงค์หลักของผู้บุกรุกในการพยายามที่จะทำให้ตัวเองมีสิทธิเทียบเท่ารูตไม่วิธีใดก็วิธีหนึ่ง ผู้ดูแลระบบควรจะป้องกันไว้ด้วยวิธีต่างๆ เช่น

- เก็บไฟล์สำคัญไว้ในอุปกรณ์บันทึกข้อมูลที่ถอดเข้าออกได้ เช่น ฟลอปปี้ดิสก์ ซีดีรอม เทปแบ็กอัพ เป็นต้น หรือเก็บข้อมูลบนสื่อที่อ่านได้อย่างเดียว
- พยายามเข้ารหัสไฟล์ทุกไฟล์ที่สำคัญและมีสิทธิเข้าถึงได้โดยรูตเท่านั้น
- อย่าลืมแบ็กอัพระบบอย่างสม่ำเสมอ ทำให้ข้อมูลแบ็กอัพตามทันข้อมูลปัจจุบันอย่างเสมอ

5.4 ความหมายของ Sticky bit

มีบิตพิเศษอีกหนึ่งบิตเรียกว่า Sticky ซึ่งในปัจจุบันจะใช้กับไดเรกทอรีเท่านั้นมีความหมายคือ ไดเรกทอรีใดก็ตามที่มีบิตที่เรียกว่า Sticky อยู่ นั่นหมายถึงไฟล์ในไดเรกทอรีจะมีการเปลี่ยนแปลง ลบ เปลี่ยนชื่อ เพิ่มเนื้อหาได้เฉพาะบุคคลสามคนนี้เท่านั้นคือ

- เจ้าของไฟล์ (ไฟล์ภายใต้ไดเรกทอรีที่มี Sticky bit)
- เจ้าของไดเรกทอรีที่มี Sticky bit
- ผู้ดูแลระบบหรือรูต

มักใช้ในไดเรกทอรี /tmp ของลีนุกซ์

5.5 การกำหนดสิทธิต่างๆ ในไฟล์โดยอัตโนมัติ

เมื่อมีการสร้างไฟล์ขึ้นมาใหม่นั้น ในเชลจะมีคำสั่ง umask ไว้กำหนดสิทธิต่างๆ ในไฟล์โดยอัตโนมัติ ทั้งนี้ก็เพื่อที่จะได้ไม่ต้องทำการกำหนดสิทธิหรือเปลี่ยนแปลงสิทธิของไฟล์ทุกครั้งที่มีการสร้างขึ้นใหม่

คำสั่ง umask ย่อมาจาก user file – creation mode mark จะตามด้วยเลขฐานแปดจำนวน 4 บิต แล้วนำไปทำการ XOR กับค่าดีฟอลต์ในการสร้างไฟล์ในระบบ ซึ่งตัวเลขเหล่านั้นจะแทนสิทธิต่างๆ ในทางตรงกันข้ามกับการกำหนดสิทธิของไฟล์

โดยทั่วไปแล้ว ค่า umask จะมีค่าเป็น 0022 คือให้เจ้าของไฟล์นั้นมีสิทธิทุกอย่างในการดำเนินการกับไฟล์ ส่วนสมาชิกในกลุ่มและผู้ใช้งานคนอื่นนั้นกำหนดให้เพียงสามารถอ่านและรันไฟล์ได้เท่านั้น

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะของไฟล์	ค่าดีฟอลต์ของสิทธิการใช้งานของระบบ
ไฟล์ปกติที่ไม่ใช่โปรแกรม	0666
โปรแกรมใช้งานในระบบ	0777

ตารางที่ 5-11 : แสดงค่าดีฟอลต์ของสิทธิการใช้งานของระบบ

ค่า umask ที่กำหนดโดยผู้ใช้งานระบบจะนำไป XOR กับค่าดีฟอลต์ในการสร้างไฟล์ของระบบ ตัวอย่างเช่น ถ้าใช้ค่า umask เป็น 0022 ดังนั้นไฟล์ที่ไม่ใช่โปรแกรมที่สร้างใหม่จะมี เพอร์มิสชัน เป็น 0644 หรือ

r w - r - - r - -

default file-creation	0666	XOR
umask	0022	
result	0644	

ตารางที่ 5-12 : แสดงการ XOR เมื่อใช้ umask เป็น 0022 กับไฟล์ที่สร้างใหม่

หรือตัวอย่างในการสร้างโปรแกรมใหม่จะมี เพอร์มิสชัน เป็น 0755 หรือ r w x r – x r - x ดังตาราง

default file-creation	0777	XOR
umask	0022	
result	0755	

ตารางที่ 5-13 : แสดงการ XOR เมื่อใช้ umask เป็น 0022 กับโปรแกรมใหม่

ปกติมักนิยมใช้ค่าที่มากับระบบ แต่ถ้าผู้ดูแลระบบต้องการความยืดหยุ่นอาจปรับค่าที่มากับระบบได้

5.6 การเปลี่ยนแปลงสิทธิต่างๆ ในไฟล์

การเปลี่ยนแปลงระดับการแอ็กเซสไฟล์ เช่น ถ้าต้องการที่จะเปลี่ยนระดับการแอ็กเซสเพื่อให้สามารถรันไฟล์ หรือ จำกัดระดับการแอ็กเซสไฟล์ของผู้ใช้งานคนอื่น ทำได้โดยใช้คำสั่ง chmod (Change mode for a file) ใช้คำสั่ง chmod กำหนดระดับการแอ็กเซส ซึ่งมีอยู่ 2 แบบคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.6.1 กำหนดด้วยเลขฐานแปด

ในการใช้งาน ไม่ว่าจะเป็นการอ่าน การเปลี่ยนแปลงสิทธิการใช้งานของไฟล์ใดๆ ในระบบนั้น มักจะระบุในรูปเลขฐานแปดแทนการเขียนว่า – r w x r – x r - - เพราะสะดวกกว่า สรุปเป็นตารางในรูปแบบเพื่อช่วยจำได้ว่า

special				owner			Group			others		
SUID	S	4										
SGID	S	2	r	w	x	R	w	x	r	W	x
Sticky	T	1	4	2	1	4	2	1	4	2	1

ตารางที่ 5-14 : แสดงตารางช่วยจำ

วิธีการดูตาราง คือในฟิลด์ special จะรวมเลขเป็นแนวตั้งลงมา ส่วนฟิลด์ owner, group และ others จะรวมเลขเป็นแนวนอน ตัวอย่างเช่น สิทธิการใช้งานไฟล์เป็น – r w x r – x r -- นำมาจับใส่ในตารางจะได้เป็น

special				owner			group			others		
SUID	-	4										
SGID	-	2	r	W	x	r	-	x	r	-	-
Sticky	-	1	4	2	1	4	2	1	4	2	1
0			7			5			4		

ตารางที่ 5-15 : แสดงการลงตาราง

ซึ่งไม่มีบิตพิเศษไม่ว่าจะเป็น SUID, SGID และ sticky-bit ใดๆ ดังนั้นอ่านเฉพาะตัวอักษรที่มีแล้วนำมารวมกัน โดยเริ่มจากบิตแรกไม่มีบิตพิเศษใด ผลจึงเป็น 0 จากนั้นช่อง owner รวมได้ 7 ของ group ได้ 5 และถัดมาเป็น others ได้ 4 นำมาเขียนเรียงกันได้ดังนั้น – r w x r – x r - - มีคำตอบเป็น 0754 หรืออีกตัวอย่างหนึ่ง เช่น – r – s r – x r - x จับใส่ในตารางได้เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

special				owner			group			others		
SUID	s	4										
SGID	-	2	r	-	x	r	-	x	r	-	x
Sticky	-	1	4	2	1	4	2	1	4	2	1
4			5			5			5		

ตารางที่ 5-16 : แสดงการลงตาราง

ในกรณีที่มีบิตพิเศษ ให้แทนที่ EXECUTE หรือ x ด้วยบิตพิเศษ จากนั้นคำนวณด้วยวิธีการเดิม เห็นว่ามีบิตพิเศษเป็น SUID ดังนั้นได้เป็น 4 ถัดมาเป็น owner รวมกันได้ 5 ส่วน group รวมกันได้ 5 และ others ก็ 5 เช่นเดียวกัน ดังนั้น – r – s r – x r - x มีคำตอบเป็น 4555

5.6.2 กำหนดโดยใช้ตัวอักษร(Symbolic setting)

กลุ่มผู้ใช้งาน	เครื่องหมายการจัดการ	รูปแบบของสิทธิ
u (เจ้าของไฟล์)	+ (เพิ่มสิทธิ)	r (การอ่าน)
g (บุคคลในกลุ่มเจ้าของไฟล์)	= (กำหนดสิทธิ)	w (การเขียน)
o (บุคคลทั่วไป)	- (ลดสิทธิ)	x (รันได้)
a (ทุกคนทุกกลุ่ม)		

ตารางที่ 5-17 : ตารางแสดงการเปลี่ยนแปลงสิทธิโดยใช้ตัวอักษร

และการกำหนดสิทธิในส่วนของกลุ่มผู้ใช้งานและรูปแบบของสิทธินั้น เราสามารถกำหนดได้มากกว่าหนึ่งกลุ่มได้ เช่น g+w จะหมายถึงกำหนดเพิ่มให้บุคคลในกลุ่มเจ้าของไฟล์สามารถแก้ไขไฟล์ได้

```
[root@isag07 bigdir]# ls -l bigfile
-rw-r--r--    1 root     root            0 Jan 18 17:49 bigfile
[root@isag07 bigdir]# chmod g+w bigfile
[root@isag07 bigdir]# ls -l bigfile
-rw-rw-r--    1 root     root            0 Jan 18 17:49 bigfile
[root@isag07 bigdir]#
```

รูปที่ 5-2 : ตัวอย่างการใช้คำสั่ง chmod ด้วยตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.7 การเปลี่ยนแปลงสิทธิต่างๆ ของไดเรกทอรี

การกำหนดสิทธิต่างๆ ทำได้โดยใช้คำสั่ง chmod โดยมีลักษณะการใช้งานเช่นเดียวกับการกำหนดสิทธิในไฟล์ ถ้าไม่มีเพอร์มิสชัน x (execute) จะไม่สามารถใช้คำสั่ง cd เข้าไปดูข้อมูลในไดเรกทอรีได้ รวมทั้งไม่สามารถรันไฟล์ในไดเรกทอรีได้ ดังนั้นการกำหนดเพอร์มิสชัน x ให้กับผู้ใช้งานจะมีความสำคัญเป็นอย่างยิ่ง

ไดเรกทอรีส่วนใหญ่จะมีเพอร์มิสชันเป็น d r w x r - x r - x ซึ่งจะกำหนดให้ผู้ใช้งานที่เป็นเจ้าของไฟล์สามารถที่จะเพิ่มหรือลบไฟล์ในไดเรกทอรีได้ ส่วนผู้ใช้งานอื่นสามารถจะดูรายชื่อและรันไฟล์ได้เพียงอย่างเดียว

```
[root@isag07 bigdir]# ls -la
total 12
drwxr-xr-x    2 root     root         4096 Jan 18 18:08 .
drwxr-xr-x    6 root     root         4096 Jan 18 18:08 ..
-rw-r--r--    1 root     root            2 Jan 18 18:08 bigfile
-rw-r--r--    1 root     root            0 Jan 18 18:07 picfile
-rw-r--r--    1 root     root            0 Jan 18 18:06 smallfile
[root@isag07 bigdir]#
```

รูปที่ 5-3 : ตัวอย่างการใช้คำสั่งกำหนดระดับการเข้าถึงไดเรกทอรี

การทำความเข้าใจเรื่องสิทธิในการเข้าถึงทั้งไฟล์หรือไดเรกทอรี เป็นสิ่งจำเป็น เพราะจะช่วยแก้ปัญหาที่เกิดขึ้นได้อย่างรวดเร็ว มีประเด็นสำคัญที่ควรให้ความสนใจบางประเด็นเช่น

- ไดเรกทอรีจำเป็นต้องมีสิทธิเอ็กซิคิวต์สำหรับการเปลี่ยนเป็นไดเรกทอรีปัจจุบัน โดยการใช้คำสั่ง cd หรือ chdir รวมไปถึงการจะเปลี่ยนเป็นไดเรกทอรีที่อยู่ภายใต้ไดเรกทอรีที่มีสิทธิดังกล่าวอยู่นี้
- ถ้าไดเรกทอรีใดไม่มีสิทธิในการ เอ็กซิคิวต์ จะไม่สามารถเข้าถึงไฟล์ภายใต้ไดเรกทอรีนั้นได้เลย ไม่มีข้อยกเว้น รวมไปถึงเจ้าของไดเรกทอรีด้วย ก็ไม่สามารถทำได้
- ถ้ามีสิทธิเฉพาะเอ็กซิคิวต์แต่ไม่มีสิทธิอ่านจะสามารถเข้าไปในไดเรกทอรีนั้นได้ แต่จะไม่สามารถใช้คำสั่ง ls ในการดูไฟล์ภายในไดเรกทอรีนั้นได้ (คือไม่มีสิทธิ์ในการอ่านเนื้อความภายในไดเรกทอรี) แต่ถ้าทราบชื่อไฟล์และได้รับสิทธิเพียงพอในการเข้าใช้งานไฟล์เหล่านั้น มันจะเป็นประโยชน์ต่อเจ้าของไฟล์ในการซ่อนเนื้อความในไดเรกทอรีนั้น ซึ่งจะพบเห็นได้บ่อยครั้ง
- ถ้าจะสร้างลิงค์ไปยังไฟล์ในไดเรกทอรีใด ไดเรกทอรีนั้นจำเป็นต้องมีสิทธิอ่านและเอ็กซิคิวต์ ในกรณีลบก็เช่นเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
การใช้คำสั่ง find

6.1 รูปแบบคำสั่ง find

เคยเจอปัญหาที่ว่า ไม่รู้ว่าไฟล์ที่ต้องการใช้ถูกเก็บไว้ที่ไหนบ้าง ในเมื่อไฟล์และไดเรกทอรีมีอยู่ตั้งมากมาย ถ้าคิดจะเปิดหาทุกไดเรกทอรี คงต้องใช้เวลาในการค้นหาเป็นวันแน่ๆ

ลีนุกซ์มีวิธีที่ง่าย สะดวก และรวดเร็วที่สุด และข้อดีของมันก็คือ ถ้าจำชื่อไฟล์ไม่ได้ สามารถใช้เงื่อนไขอื่นในการหาไฟล์นั้นได้ เช่น รู้ขนาดของไฟล์ หรือวันที่ที่แก้ไขไฟล์นั้นเป็นครั้งสุดท้ายก็ค้นหาที่อยู่ของมันได้ คำสั่งที่ใช้ค้นหาก็คือ “find” ซึ่งมีรูปแบบของคำสั่งดังต่อไปนี้

```
find [path...] [expression]
```

รูปที่ 6-1 : รูปแบบของคำสั่ง find

จากรูปแบบคำสั่ง โดยคำสั่งจะเริ่มจาก find แล้วตามด้วย path แล้วปิดท้ายด้วย expression แล้ว path และ expression คืออะไร?

[path...] คือ เป็นการกำหนดว่าให้ค้นหาโดยเริ่มจากไดเรกทอรีนี้

[expression] คือ การกำหนดการกระทำให้กับคำสั่ง find

ซึ่ง expression นี้จะแยกได้อีกเป็น 2 ส่วน คือ [options] และ [action]

[options] เป็นส่วนแรกของ expression ซึ่งจะเป็นตัวกำหนดว่าให้คำสั่ง find ค้นหาอะไร จะมีรายละเอียดดังนี้

-help, - -help	: ใช้แสดงการใช้คำสั่ง
-version,--version	: ใช้แสดง version
+n	: ใช้ต่อท้ายคำสั่ง มีความหมายว่า มากกว่า "n" โดยที่ n คือ ตัวเลข
-n	: ใช้ต่อท้ายคำสั่ง มีความหมายว่า น้อยกว่า "n" โดยที่ n คือ ตัวเลข
n	: ใช้ต่อท้ายคำสั่ง มีความหมายว่า เท่ากับ "n" โดยที่ n คือตัวเลข
-amin +-n	: ใช้ค้นหาไฟล์ตามเวลาที่เข้าไปใช้ครั้งสุดท้าย โดยคิดที่ นาที
-atime +-n	: ใช้ค้นหาไฟล์ตามเวลาที่เข้าไปใช้ครั้งสุดท้าย โดยคิดที่ ชั่วโมง
-cmin +-n	: ใช้ค้นหาไฟล์ตามเวลาที่เปลี่ยน สถานะโดยคิดที่ นาที
-ctime +-n	: ใช้ค้นหาไฟล์ตามเวลาที่เปลี่ยน สถานะโดยคิดที่ ชั่วโมง
-gid n	: ใช้ค้นหาไฟล์ตาม numeric group ID ที่ n
-group g_name	: ใช้ค้นหาไฟล์ที่มี group owner เป็น g_name
-mmin +-n	: ใช้ค้นหาไฟล์ตามเวลาที่แก้ไขไฟล์ครั้งสุดท้าย คิดที่นาที
-mtime +-n	: ใช้ค้นหาไฟล์ตามเวลาที่แก้ไขไฟล์ครั้งสุดท้าย คิดที่ ชั่วโมง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-name file_name : ใช้ค้นหาไฟล์ตามชื่อ file_name ที่ระบุ

-size +-n : ใช้ค้นหาไฟล์ตามขนาดของมัน

-type b : ใช้ค้นหาไฟล์ที่มีการอ่านที่บล็อก

-type c : ใช้ค้นหาไฟล์ที่มีอักขระพิเศษ

-type d : ใช้ค้นหาไดเรกทอรี

-type f : ใช้ค้นหาไฟล์ทั่วไป

-type l : ใช้ค้นหาไฟล์ที่มีการ link

-type p : ใช้ค้นหาไฟล์ที่เป็น name pipe(FIFO)

-type s : ใช้ค้นหาไฟล์ที่เป็น socket

-uid n : ใช้ค้นหาไฟล์ตาม numeric user ID ที่ n

-user name : ใช้ค้นหาไฟล์ตามชื่อเจ้าของไฟล์

[action] คือส่วนที่เป็นตัวกำหนดว่าถ้าเจอตัวที่ค้นหาแล้วให้ทำอย่างไร มีรายละเอียดดังนี้

-print : ใช้แสดง path ของไฟล์ที่เจอออกทางหน้าจอ

-ls : แสดง path และแสดงรายละเอียดของไฟล์ที่เจอด้วย

-exec command{} ; : ใช้สั่งให้เมื่อเจอไฟล์ที่ค้นหาแล้วก็ทำตาม command ที่กำหนด

รายละเอียดที่นำมาแสดงให้ดูนี้เป็น action ที่มักจะใช้เท่านั้น ถ้าจะดูให้ละเอียดให้ดูที่ help

6.2 ตัวอย่างการนำคำสั่ง find ไปใช้งาน

เมื่อทราบรูปแบบของคำสั่งแล้วลองมาดูตัวอย่างการใช้คำสั่ง find กันบ้าง

```
[root@isag07 root]# find --help
Usage: find [path...] [expression]
default path is the current directory; default expression is -print
expression may consist of:
operators (decreasing precedence; -and is implicit where no others are given):
      ( EXPR ) ! EXPR -not EXPR EXPR1 -a EXPR2 EXPR1 -and EXPR2

      EXPR1 -o EXPR2 EXPR1 -or EXPR2 EXPR1 , EXPR2
options (always true): -daystart -depth -follow --help
      -maxdepth LEVELS -mindepth LEVELS -mount -noleaf --version -xdev
tests (N can be +N or -N or N): -amin N -anewer FILE -atime N -cmin N

      -cnewer FILE -ctime N -empty -false -fstype TYPE -gid N -group NAME
      -ilname PATTERN -iname PATTERN -inum N -ipath PATTERN -iregex PATTERN
      -links N -lname PATTERN -mmin N -mtime N -name PATTERN -newer FILE

      -nouser -nogroup -path PATTERN -perm [+-]MODE -regex PATTERN
      -size N[bckw] -true -type [bcdpfls] -uid N -used N -user NAME
      -xtype [bcdpfls]

actions: -exec COMMAND ; -fprint FILE -fprint0 FILE -fprintf FILE FORMAT
      -ok COMMAND ; -print -print0 -printf FORMAT -prune -ls

Report bugs to <bug-findutils@gnu.org>.
[root@isag07 root]#
```

รูปที่ 6-2 : แสดงการใช้คำสั่ง find

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตัวอย่างที่เห็นเป็นการขอดูการใช้คำสั่ง find ตัวอย่างต่อไปลองใช้คำสั่ง find หาไฟล์ที่ ชื่อ test โดยให้แสดง path name ออกมาทางจอภาพ และเริ่มค้นหาจาก /

```
[root@isag07 root]# find / -name test -print
/var/lib/mysql/test
/root/download/c50a/src/libdes/t/test
/root/scanner/test
/usr/bin/test
/usr/share/doc/db4-devel-4.0.14/ref/test
/usr/share/doc/ncurses-devel-5.2/test
[root@isag07 root]#
```

รูปที่ 6-3 : แสดงการหาตำแหน่งของไฟล์ชื่อ test โดยการใช้คำสั่ง find

ตัวอย่างข้างล่างนี้ให้แสดงรายละเอียดด้วย

```
[root@isag07 root]# find / -name test -ls
261415    4 drwx------   2 mysql    mysql        4096 Jan 10 16:52 /var/lib/mysql/test
117083    4 -rw-------   1 ohh      ohh           531 Nov  5  1996 /root/download/c50a/src/libdes/t/test
149122    4 drwxr-xr-x   2 root     root         4096 Nov  7 17:11 /root/scanner/test
336928   24 -rwxr-xr-x   1 root     root        24504 Aug 30 03:56 /usr/bin/test
177921    4 drwxr-xr-x   2 root     root         4096 Oct 23 10:42 /usr/share/doc/db4-devel-4.0.14/ref/tes
274381    4 drwxr-xr-x   2 root     root         4096 Oct 23 10:43 /usr/share/doc/ncurses-devel-5.2/test
[root@isag07 root]#
```

รูปที่ 6-4 : แสดงรายละเอียดของไฟล์โดยใช้คำสั่ง find ร่วมกับคำสั่ง ls

ตัวอย่างต่อไปเป็นการใช้คำสั่ง find โดยไม่ได้กำหนดอะไรเลย

```
[root@isag07 scanner]# find
.
./project.sh
./file_unowned
./file_symbolic
./file_suid
./file_world
./test
./test/ownerroot
./test/ownerohh
./test/ownerpanny
./test/links
./test/root
./test/newlinks
./uolist
./.uo.tmp
./homework
./code_unowned.sh
./class
[root@isag07 scanner]#
```

รูปที่ 6-5 : แสดงการใช้คำสั่ง find โดยไม่กำหนดอะไรเลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อไปเป็นการค้นหาไฟล์ที่มีรูตเป็นเจ้าของไฟล์

```
[root@isag07 scanner]# find /root/scanner -user root -print
/root/scanner
/root/scanner/project.sh
/root/scanner/file_unowned
/root/scanner/file_symbolic
/root/scanner/file_suid
/root/scanner/file_world
/root/scanner/test
/root/scanner/test/ownerroot
/root/scanner/test/links
/root/scanner/test/root
/root/scanner/uolist
/root/scanner/.uo.tmp
/root/scanner/homework
/root/scanner/code_unowned.sh
/root/scanner/class
[root@isag07 scanner]#
```

รูปที่ 6-6 : แสดงการค้นหาไฟล์ที่มีรูตเป็นเจ้าของ

ตัวอย่างต่อไปเป็นการค้นหาไฟล์ที่มีขนาด มากกว่า 5000 โดยให้เริ่มค้นหาที่ไดเรกทอรีรูตและแสดงรายละเอียดของไฟล์ที่เจอด้วย

```
[root@isag07 scanner]# find /root -size +5000 -ls
309460 2900 -rw-------   1 ohh      ohh       2964507 Oct 25 06:16 /root/download/Crack_5.0a.tar.gz
325593 9912 -rw-r--r--   1 root     root     10131865 Jan 12 22:43 /root/amsn_received/lab_hand.tgz
 53004 2516 -rw-r--r--   1 root     root      2571292 Jan 10 14:10 /root/backupproject2/magick.miff
  4925 3436 -rw-r--r--   1 root     root      3513016 Jan 10 14:11 /root/panny/2.jpg
  4927 6636 -rw-r--r--   1 root     root      6781889 Jan 10 14:11 /root/panny/4.jpg
277448 6216 -r-xr-xr-x   1 root     root      6350840 Jan 10 14:11 /root/panny/picture/Trip04.bmp
277449 4624 -r-xr-xr-x   1 root     root      4718646 Jan 10 14:11 /root/panny/picture/02.bmp
277451 4624 -r-xr-xr-x   1 root     root      4718646 Jan 10 14:11 /root/panny/picture/03.bmp
117143 4148 -rw-r--r--   1 root     root      4241310 Jan 11 22:43 /root/picfind/9
213359 2712 -rw-r--r--   1 root     root      2769210 Jan 11 22:37 /root/.compupic/def.phd
293675 3896 -rw-r--r--   1 root     root      3983360 Jan 13 00:22 /root/backupproject2.tgz
[root@isag07 scanner]#
```

รูปที่ 6-7 : แสดงการค้นหาไฟล์ที่มีขนาดมากกว่า 5000

ตัวอย่างสุดท้ายเป็นการหา ไฟล์ link โดยเริ่มการค้นหาจากไดเรกทอรีย่อย download ซึ่งอยู่ในไดเรกทอรีรูตเมื่อเจอแล้วก็ให้แสดงรายละเอียดด้วย

```
[root@isag07 scanner]# find /root/download/ -type l -ls
 20986    0 lrwxrwxrwx   1 root     root            9 Nov  8 00:45 /root/download/john-1.6.32/src/arch.h -> gene
261398    0 lrwxrwxrwx   1 root     root            4 Nov  8 00:46 /root/download/john-1.6.32/run/unshadow -> jo
261399    0 lrwxrwxrwx   1 root     root            4 Nov  8 00:46 /root/download/john-1.6.32/run/unafs -> john
261400    0 lrwxrwxrwx   1 root     root            4 Nov  8 00:46 /root/download/john-1.6.32/run/unique -> john
[root@isag07 scanner]#
```

รูปที่ 6-8 : แสดงการหาไฟล์ลิงก์พร้อมกับแสดงรายละเอียดของไฟล์ด้วย

เมื่อรู้จักประโยชน์และการใช้คำสั่ง find อย่างคร่าวๆ แล้ว ดังนั้นเวลาที่จะค้นหาไฟล์ในลีนุกซ์ก็สามารถทำได้ง่ายและไม่เสียเวลาอีกต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 การค้นหาไฟล์ในรูปแบบต่างๆ ที่อาจเกิดเป็นช่องโหว่

ต่อไปจะเป็นการค้นหาไฟล์ที่อาจจะเป็นจุดอ่อนหรือจุดบกพร่องที่ทำให้เสี่ยงต่อการเกิดความไม่ปลอดภัย ซึ่งจะแบ่งเป็นการตรวจสอบไฟล์หลายๆ แบบ ดังนี้

6.3.1 การตรวจสอบหา Unowned file

Unowned file คือ ไฟล์ที่ไม่มีทั้งเจ้าของและไม่มีกลุ่ม หรือไฟล์ที่ไม่มีเจ้าของแต่มีกลุ่ม หรือไฟล์ที่มีเจ้าของแต่ไม่มีกลุ่ม (โดยปกติแล้วไฟล์ทุกไฟล์จะต้องมีเจ้าของและเจ้าของก็จะมีกลุ่ม) การไม่มีเจ้าของก็คือ เจ้าของไฟล์ ๆ นั้นไม่ได้อยู่ใน /etc/passwd ในทำนองเดียวกันการไม่มีกลุ่มก็คือ กลุ่มของไฟล์ ๆ นั้นไม่มีอยู่ใน /etc/group

โดยปกติแล้วเมื่อผู้ใช้งานทำการสร้างไฟล์ขึ้นมา ไฟล์ ๆ นั้นจะมีเจ้าของเป็นผู้ใช้งานคนนั้นทันที และไฟล์นั้นจะอยู่ในกลุ่มเดียวกับผู้ใช้งานด้วย

แล้วจะเกิด Unowned file ได้อย่างไร ?

Unowned file สามารถเกิดขึ้นได้เนื่องจากผู้ดูแลระบบได้ลบผู้ใช้งานคนนั้นออกไปจากระบบ อย่างเช่น ในบางระบบที่ไม่ฉลาดพอและผู้ดูแลระบบลืม การลบผู้ใช้งานออกจะทำแค่ลบชื่อผู้ใช้งานออกไปจากไฟล์ /etc/passwd เท่านั้นแต่มันไม่ฉลาดพอที่จะติดตามไปลบไฟล์ต่างๆ ของผู้ใช้งานคนนั้นที่เคยสร้างไว้ออกไปจากระบบได้ เช่น ไฟล์ในโฮมไดเรกทอรีหรือไฟล์ใน /tmp ทำให้ยังคงมีไฟล์ของคนๆ นั้นค้างอยู่ในระบบ โดยที่ไม่สามารถระบุได้ว่า ใครเป็นเจ้าของ แต่ id ของไฟล์นั้นจะมีค่าเท่ากับ user id ของเจ้าของคนเดิม

ตัวอย่างเช่น hitomi มี uid = 508 แล้วผู้ดูแลระบบได้ลบ hitomi ออกจาก /etc/passwd แต่ไม่ได้ลบไฟล์ที่ hitomi สร้างเอาไว้ ทำให้ไฟล์ที่ hitomi ได้เคยสร้างไว้ไม่มีเจ้าของแต่ยังมี id ของไฟล์ = 508 อยู่ ต่อมาได้เพิ่มผู้ใช้งานคนใหม่ชื่อ kotaro และมี uid = 508 เหมือนเดิม ทำให้ไฟล์ทั้งหมดที่มี id = 508 นั้นมีเจ้าของเป็น kotaro ทันที หรือพูดกลับอีกอย่างว่า kotaro เป็นเจ้าของไฟล์เหล่านั้น ซึ่ง kotaro จะมีสิทธิทุกอย่างในไฟล์เหล่านั้นด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
[root@isag07 home]# ls -la
total 44
drwxr-xr-x    8 root     root         4096 Jan 24 17:33 .
drwxr-xr-x   20 root     root         4096 Jan 27 22:53 ..
drwx------    3 ann      ann          4096 Jan 24 18:49 ann
drwx------    3 508      hitomi       4096 Jan 24 18:49 hitomi
drwx------    2 root     root        16384 Oct 23 10:29 lost+found
drwx------   26 ohh      ohh          4096 Jan 24 18:46 ohh
drwx------    3 orn      orn          4096 Jan 24 17:32 orn
drwx------   11 panny    panny        4096 Jan 10 14:32 panny
[root@isag07 home]# ls -la
total 48
drwxr-xr-x    9 root     root         4096 Jan 27 23:02 .
drwxr-xr-x   20 root     root         4096 Jan 27 22:53 ..
drwx------    3 ann      ann          4096 Jan 24 18:49 ann
drwx------    3 kotaro   hitomi       4096 Jan 24 18:49 hitomi
drwx------    3 kotaro   kotaro       4096 Jan 27 23:02 kotaro
drwx------    2 root     root        16384 Oct 23 10:29 lost+found
drwx------   26 ohh      ohh          4096 Jan 24 18:46 ohh
drwx------    3 orn      orn          4096 Jan 24 17:32 orn
drwx------   11 panny    panny        4096 Jan 10 14:32 panny
[root@isag07 home]#
```

รูปที่ 6-9 : แสดงการเพิ่มผู้ใช้งานคนใหม่ที่มี uid เป็นของคนเดิม

รูปแบบคำสั่ง มีดังนี้

```
[root@isag07 root]# find /root \( -nouser -o -nogroup \) -ls
309478   60 -rw-r-----   1 20207    102         55583 Jun 28  1993 /root/download/part01
309479   60 -rw-r-----   1 20207    102         55433 Jun 28  1993 /root/download/part02
309480   60 -rw-r-----   1 20207    102         55690 Jun 28  1993 /root/download/part03
309481   60 -rw-r-----   1 20207    102         55546 Jun 28  1993 /root/download/part04
309482   60 -rw-r-----   1 20207    102         55182 Jun 28  1993 /root/download/part05
309483   60 -rw-r-----   1 20207    102         55066 Jun 28  1993 /root/download/part06
309484   60 -rw-r-----   1 20207    102         55605 Jun 28  1993 /root/download/part07
309485   60 -rw-r-----   1 20207    102         55478 Jun 28  1993 /root/download/part08
309486   60 -rw-r-----   1 20207    102         55482 Jun 28  1993 /root/download/part09
309487   60 -rw-r-----   1 20207    102         55363 Jun 28  1993 /root/download/part10
309488   48 -rw-r-----   1 20207    102         48041 Jun 28  1993 /root/download/part11
309489   12 -rw-r-----   1 20207    102          8348 Jun 28  1993 /root/download/patch01
309490   68 -rw-r-----   1 20207    102         63728 Jun 28  1993 /root/download/patch02
309491   44 -rw-r-----   1 20207    102         43378 Jun 28  1993 /root/download/patch03
309492   48 -rw-r-----   1 20207    102         46424 Jun 28  1993 /root/download/patch04a
309493   56 -rw-r-----   1 20207    102         51167 Jun 28  1993 /root/download/patch04b
309494   48 -rw-r-----   1 20207    102         45868 Jun 28  1993 /root/download/patch04c
309495   44 -rw-r-----   1 20207    102         42480 Jun 28  1993 /root/download/patch05
309496   64 -rw-r-----   1 20207    102         58894 Jun 28  1993 /root/download/patch06a
309497   60 -rw-r-----   1 20207    102         56519 Jun 28  1993 /root/download/patch06b
309498   56 -rw-r-----   1 20207    102         53225 Jun 28  1993 /root/download/patch06c
[root@isag07 root]#
```

รูปที่ 6-10 : คำสั่ง find หา Unowned file

6.3.2 การตรวจสอบหาไฟล์ลิงก์

Symbolic link หรือ symlink หรือ soft link คือ ไฟล์พิเศษชนิดหนึ่งที่ชี้ไปยังไฟล์อื่น คล้ายๆ กับ shortcut ใน Windows symbolic link นี้ไม่ได้มีข้อมูลบรรจุอยู่จริงๆ มันเป็นเพียงแค่จุดที่ชี้ไปยังที่ไฟล์อื่นในระบบ ซึ่งตรงกันข้ามกับ hard link ดังที่ได้กล่าวไว้แล้วในบทก่อน

ตัวอย่างที่ใช้ประโยชน์จากการมี Symbolic link เช่น ในไฟล์ไลบรารีจะมีไฟล์ header ที่ใช้ในการคอมไพล์โปรแกรม บางครั้งอาจมีการอัพเดตเพื่อแก้จุดอ่อนบางอย่างหรือว่าเราอาจต้องการเพิ่มเติมหรือเปลี่ยนแปลง จึงมีการลงไฟล์ไลบรารีเวอร์ชันใหม่ลงไปและถ้าเราต้องการให้โปรแกรมเก่าที่เราเคยใช้

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้นำไปดัดแปลง เผยแพร่ หรือนำไปใช้

เขียนไว้เรียกใช้ไฟล์ header เวอร์ชันใหม่นี้ แทนที่จะเข้าไปแก้ไขในโปรแกรมทุกโปรแกรม เราเพียงแต่ทำการสร้างลิงก์จาก library เวอร์ชันเก่าไปยังเวอร์ชันใหม่ ทันทีที่มีการรันโปรแกรม ตัวโปรแกรมก็จะเข้าไปที่เวอร์ชันเก่าแล้วลิงก์ไปยังเวอร์ชันใหม่เอง ทำให้เป็นการประหยัดพื้นที่ เพราะไม่งั้นก็ต้องมีไฟล์ของ header นั้นถึง 2 ไฟล์ ก็คือ เวอร์ชันเก่าและเวอร์ชันใหม่

รูปแบบคำสั่ง มีดังนี้

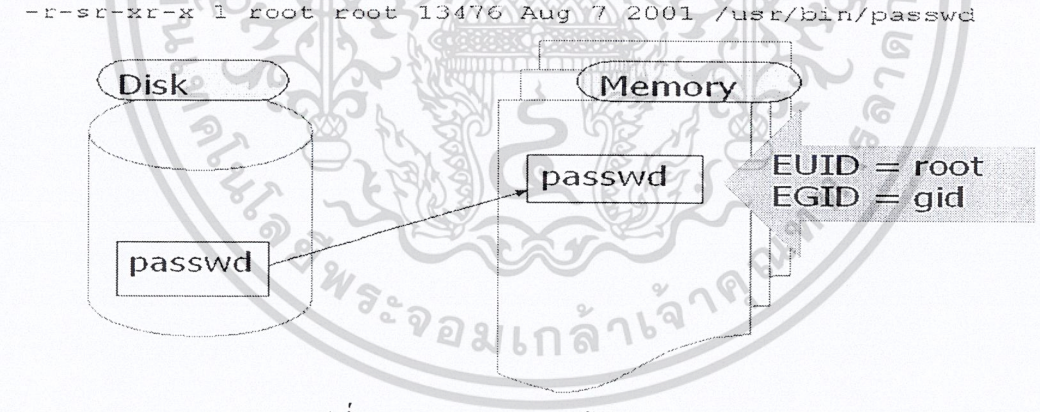
```
[root@isag07 root]# find /root/download/ -type l -ls
 20986     0 lrwxrwxrwx   1 root     root            9 Nov  8 00:45 /root/download/john-1.6.32/src/arch.h -> gene
261398     0 lrwxrwxrwx   1 root     root            4 Nov  8 00:46 /root/download/john-1.6.32/run/unshadow -> jo
261399     0 lrwxrwxrwx   1 root     root            4 Nov  8 00:46 /root/download/john-1.6.32/run/unafs -> john
261400     0 lrwxrwxrwx   1 root     root            4 Nov  8 00:46 /root/download/john-1.6.32/run/unique -> john
[root@isag07 root]#
```

รูปที่ 6-11 : คำสั่ง find เพื่อหา Symbolic link

6.3.3 การตรวจหา SUID หรือ SGID

โปรแกรมที่มีการเปลี่ยน uid ขณะที่มีการทำงานหรือเมื่อเป็นโพรเซส เรียกว่าโปรแกรม SUID โดยจะใช้วิธีการเปลี่ยน uid ให้ตรงกับเจ้าของโปรแกรมนั้นๆ

ในตัวอย่างนี้ คือ passwd ซึ่งเจ้าของไฟล์นี้คือ รูต ดังนั้นเมื่อโปรแกรมทำงานจะได้โพรเซสจะมี EUID เป็นรูต ซึ่งจะไม่ใช่ผู้ใช้งานที่สั่งให้โปรแกรม passwd ทำงานอย่างในกรณีทั่วไป



รูปที่ 6-12 : การทำงานของโปรแกรม SUID

สำหรับ SGID ใช้แนวความคิดเดียวกันแต่เปลี่ยนจากเจ้าของไฟล์เป็น group เดียวกับไฟล์นั้นๆ

สำหรับการระบุ SUID และ SGID จะระบุในตำแหน่งเดียวกับ execute ใน permission ของไฟล์หรือโปรแกรมในระบบ ตัวอย่างเช่น โปรแกรม passwd

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

จะเห็นว่ามีเครื่องหมายว่า 's' ใช้ในการระบุ SUID ของโปรแกรม passwd และตัวอย่างโปรแกรมที่มี SGID เช่น

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบคำสั่ง มีดังนี้

```
[root@isag07 root]# find /root \( -perm -2000 -o -perm -4000 \) -ls
213339    4 -rws--x---   1 root     root           91 Jan 10 14:10 /root/backupproject/ss
213343    4 -rwsr-sr-x   1 root     root            3 Jan 10 14:10 /root/backupproject/test1
213351    4 -rwsr-sr-x   1 root     root            3 Jan 10 14:10 /root/backupproject/world
213353    0 -rwsr-sr-x   1 root     root            0 Jan 10 14:10 /root/backupproject/x
213354    4 -rws--x---   1 root     root           91 Jan 10 14:10 /root/backupproject/xxx
[root@isag07 root]#
```

รูปที่ 6-13 : คำสั่ง find หา SUID หรือ SGID

6.3.4 ไฟล์และไดเรกทอรีที่เป็น World Writable

World-Writable คือ การที่อนุญาตให้ทุกคนมีสิทธิในการเขียนไฟล์ได้ ถึงแม้จะเป็น other ซึ่งในเวลาปกติแล้วจะไม่มีสิทธิก็ตาม

World-Writable file อาจจะเกิดจากการจงใจหรือตั้งใจทำขึ้น และใช้ประโยชน์จากการอนุญาตให้ทุกคนสามารถเข้ามาเขียนได้ อย่างเช่น Tmp file, ไฟล์ที่ต้องการ share (shared file) ในกรณีที่เราต้องการทำ shared file เราก็จะเซตสิทธิของไฟล์นั้นให้ other สามารถเขียนและอ่านไฟล์นั้นได้ (read ใช้ในกรณีที่ต้องการดึงข้อมูล, write ใช้ในกรณีที่ต้องการนำข้อมูลมาลงไว้)

```
[root@isag07 root]# find /root/scanner/ \( -perm -2 ! -type l \) -ls
149123    4 -rw-rw-rw-   1 root     root           11 Oct 24 23:29 /root/scanner/test/ownerroot
149124    4 -rw-rw-rw-   1 ohh      ohh            10 Oct 24 23:29 /root/scanner/test/ownerohh
165454    4 -rwxrwxrwx   1 root     root            3 Jan 11 22:19 /root/scanner/homework
[root@isag07 root]#
```

รูปที่ 6-14 : คำสั่ง find หา world writable

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
ความปลอดภัยของรหัสผ่าน

7.1 เรื่องของรหัสผ่าน

ในชีวิตประจำวันระบบต่างๆ ได้มีการนำเรื่องของรหัสผ่านมาใช้ประโยชน์ เช่น ระบบ ATM, ระบบลงทะเบียนเรียน เป็นต้น รหัสผ่านที่ใช้ในระบบเหล่านี้มีไว้เพื่อพิสูจน์ตนเองก่อนที่จะเข้าใช้งานในระบบได้ ว่าบุคคลนั้นๆ เป็นตัวจริงหรือไม่ ความสำคัญของมันจึงอยู่ในระดับสูงทีเดียวก็ว่าได้ เช่น ถ้าโดนขโมยรหัสผ่านไป บุคคลนั้นก็จะมีสิทธิที่จะโดนแอบอ้างจากคนอื่นได้ แต่ในความเป็นจริงมันก็ไม่ง่ายอย่างที่คิด เพราะแทบทุกระบบก็ว่าได้จะมีเงื่อนไขเพิ่มเติมในการขอเข้าใช้งานระบบ ซึ่งจะป้องกันในส่วนนี้ เช่น ต้องมีบัตร ATM พร้อมกับการกดรหัสผ่านที่ถูกต้องได้ไม่เกิน 3 ครั้ง จึงจะสามารถถอนเงินได้ หรือว่าต้องมียูสเซอร์เนมและรหัสผ่านซึ่งมันจะต้องตรงกับที่ระบบได้ทำการเก็บข้อมูลไว้ จึงจะสามารถเข้าใช้งานในระบบได้เช่นกัน

ความปลอดภัยด่านแรกของระบบปฏิบัติการลีนุกซ์นั้น คือ การยืนยันตัวผ่านกระบวนการล็อกอินโดยการใช้ยูสเซอร์เนมและรหัสผ่านเป็นเครื่องพิสูจน์ตน ซึ่งยูสเซอร์เนมของผู้ใช้งานในสมัยก่อนอนุญาตให้มีอักขระยาวมากที่สุดเพียงแปดตัวอักษร แต่ในปัจจุบันอนุญาตให้ใช้ได้ยาวมากที่สุดถึง 256 ตัวอักษร

ดังนั้นการตั้งรหัสผ่านก็เป็นเรื่องที่สำคัญและไม่ควรมองข้ามเป็นอันขาด เพราะถ้าระบบของเราโดนแคร็กรหัสผ่านสำเร็จ อาจจะโดนแฮ็กเกอร์ล้วงข้อมูลสำคัญไป ถึงแม้ว่าเครื่องเราจะไม่มีข้อมูลสำคัญอะไรมากนัก แต่แฮ็กเกอร์อาจใช้เครื่องของเราเป็นทางผ่านเพื่อไปแฮ็กเครื่องอื่นต่อไปได้อีก

7.2 มาตรการการป้องกันการโดนแคร็กรหัสผ่าน

เนื่องจากรหัสผ่านมีความสำคัญอย่างเช่นที่กล่าวมาแล้ว ดังนั้นมาตรการที่ดีที่สุดก็คือ ให้ผู้ใช้งานตั้งรหัสผ่านที่ไม่ง่ายนักต่อการเดา และเมื่อถูกกำหนดให้เปลี่ยนรหัสผ่าน ก็ไม่ควรตั้งรหัสผ่านซ้ำเดิมก่อนหน้านี้ ฟรีแวร์ยูทิลิตี้ในตารางที่ 7-1 จะช่วยทำให้การคาดเดารหัสผ่านยากขึ้น

นอกเหนือจากเครื่องมือเหล่านี้แล้ว คุณควรมีนโยบายในการจัดการรหัสผ่านที่ดี ให้พิจารณาประเด็นต่อไปนี้ด้วย

1. ให้แน่ใจว่าผู้ใช้งานทุกคนมีรหัสผ่าน
2. บังคับให้มีการเปลี่ยนรหัสผ่านทุกๆ 30 วัน สำหรับแอคเคาน์สำคัญๆ และทุกๆ 60 วัน สำหรับผู้ใช้งานธรรมดาทั่วไป
3. กำหนดความยาวต่ำสุดของรหัสผ่านไว้อย่างน้อย 6 ตัวอักษร หรือถ้าจะให้ดีก็ 8 ตัวอักษร และมีทั้งตัวเลขและตัวอักษรผสมกัน
4. ให้เก็บบันทึกเหตุการณ์ที่เกิดจากการล็อกอินไม่สำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. คอนฟิกเซอร์วิสให้ทำการดิสคอนเน็ก (disconnect) ผู้ใช้งานออกไป หลังจากล็อกอินล้มเหลวครบ 3 ครั้ง
6. ให้มีการล็อคเอาต์แอคเคานต์หลังจากล็อกอินไม่สำเร็จเกินจำนวนครั้งที่กำหนด (ให้ระวังว่าแฮ็กเกอร์เปลี่ยนมาโจมรชตีด้วย Denial of Service เข้าทางแอคเคาน์ที่ถูกล็อกเอาต์ไว้ ซึ่งเกิดขึ้นจากความตั้งใจของแฮ็กเกอร์เอง)
7. ดิสเอเบิลเซอร์วิสที่ไม่จำเป็นต้องใช้งาน
8. ให้ประยุกต์ใช้เครื่องมือพิเศษบางอย่างที่ช่วยป้องกันไม่ให้ผู้ใช้งานตั้งรหัสผ่านที่ง่ายจนเกินไป เครื่องมือนี้จะบังคับให้รหัสผ่านที่ผู้ใช้งานตั้งขึ้นมาต้องมีทั้งตัวอักษรเล็กใหญ่ ผสมตัวเลขด้วย
9. อย่าใช้รหัสผ่านเดียวกันกับทุกๆ เซอร์วิสที่ต้องล็อกอินเข้าไป
10. อย่าเขียนรหัสผ่านแปะทิ้งไว้ที่หน้าจอเครื่อง
11. อย่าบอกรหัสผ่านของคุณแก่ผู้อื่น
12. เมื่อถูกกำหนดให้เปลี่ยนรหัสผ่าน ก็ไม่ควรตั้งรหัสผ่านซ้ำเดิมก่อนหน้านี้
13. ให้แน่ใจว่าดีฟอลต์แอคเคานต์อย่างเช่น setup และ admin ไม่ได้มีรหัสผ่านเป็นค่าดีฟอลต์

7.3 ตัวอย่างการสร้างรหัสผ่าน

มีวิธีการมากมายในการสร้างรหัสผ่านให้ยากต่อการเดาออก เช่น

1. พิมพ์เป็นภาษาอังกฤษแต่ท่องเป็นภาษาไทย เช่น vpkd.shgdiffuq0y'g]p (= อยากให้เกรดดีๆ จังเลย)
2. คิดวลีที่จำได้ง่ายมาหนึ่งข้อความ อย่างเช่น

"over the hills and far away , to grandmothers house we go."

- เอาเฉพาะตัวอักษรตัวแรกของคำแต่ละคำมาประกอบกัน (รวมทั้งเครื่องหมายวรรคตอนต่างๆ ด้วย)

othafa,tgmhwg.

- เพิ่มความซับซ้อนด้วยการแทนที่ตัวอักษรด้วยตัวเลขและสัญญลักษณ์ในขั้นที่แล้ว อย่างเช่น กำหนดให้แทนที่ t และ a ด้วย 7 และ @ ตามลำดับ

o7h@f@,7gmhwg.

- อาจเพิ่มให้มีความซับซ้อนขึ้นอีกโดยการเปลี่ยนจากตัวอักษรตัวเล็กเป็นตัวใหญ่ อย่างเช่น กำหนดให้เป็น H

o7H@f@,7gmHwg.

วิธีการเหล่านี้สามารถดัดแปลงไปใช้ได้อีกมากมายหลายวิธี ซึ่งจะทำให้การคาดเดานั้นทำได้ยากจริงๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากที่ได้กล่าวมาทั้งหมดเกี่ยวกับเรื่องของความปลอดภัยของรหัสผ่านนั้น เป็นการยากที่ผู้ดูแลระบบจะสามารถตรวจสอบได้ดังนั้นจึงมีเครื่องมือช่วยในการปกป้องรหัสผ่านให้รอดพ้นจากการคาดเดาได้ ดังตาราง 7-1

เครื่องมือ	หน้าที่	หาได้จาก
S/Key	ห้ามไม่ให้ตั้งรหัสผ่านซ้ำเดิมก่อนหน้านี้	http://www.yak.net/skey/
One Time Passwords In Everything (OPIE)	ห้ามไม่ให้ตั้งรหัสผ่านซ้ำเดิมก่อนหน้านี้	ftp.nrl.navy.mil/pub/security/opie
CrackLib	บังคับให้รหัสผ่านที่ผู้ใช้งานตั้งขึ้นมาต้องไม่ง่ายจนเกินไป คือ มีทั้งตัวอักษรเล็กใหญ่ผสมด้วยตัวเลข	ftp.//ftp.cert.org/pub/tools/cracklib
Npasswd	ใช้แทนคำสั่ง passwd ในยูนิกซ์	http://www.utexas.edu/cc/unix/software/npasswd/
Secure Remote Password	เป็นกลไกใหม่ที่สร้างความปลอดภัยให้กับการตรวจสอบผู้ใช้งานด้วยรหัสผ่านและให้กับการแลกเปลี่ยนคีย์สำหรับเข้ารหัสบนเน็ตเวิร์ก	http://srp.stanford.edu/srp/
SSH	เข้าไปแทนที่คำสั่งตระกูล R ที่มีอยู่เดิม เพื่อช่วยให้การใช้คำสั่งตระกูล R มีความปลอดภัยมากขึ้นด้วยกลไกการเข้ารหัสและการตรวจสอบผู้ใช้งานด้วยอัลกอริทึมของ RSA	http://www.cs/hut.fi/ssh

ตารางที่ 7-1: ฟรีแวร์ที่ช่วยในการปกป้องรหัสผ่านให้รอดพ้นจากการคาดเดารหัส

ถึงแม้ว่าจะมีรหัสผ่านที่ดีอย่างไร แต่ถ้าเก็บแบบอย่างเปิดเผย (สามารถนำไปใช้ได้เลยโดยที่ไม่ต้องผ่านกรรมวิธีใดๆ) มันก็ไม่มีความหมายในเรื่องความปลอดภัยแต่อย่างไร เพราะฉะนั้นจึงต้องมีการปกปิดโดยทำการแปลงรหัสผ่านเหล่านั้นด้วยวิธีการเข้ารหัส เพื่อไม่ให้มีการขโมยการใช้รหัสผ่านได้ง่าย ซึ่งรหัสผ่านก่อนการเข้ารหัสจะเรียกว่า เพลนเท็กซ์ (Plain Text) และหลังจากเข้ารหัสเรียบร้อยแล้วจะได้ไซเฟอร์เท็กซ์ (Cipher Text)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยปกติแล้วรหัสผ่านจะเก็บไว้ในคอลัมน์รหัสผ่านของไฟล์ /etc/passwd แต่ถ้ามีการแปลงเป็นไซเฟอร์เท็กซ์แล้วจะเก็บรหัสผ่านในรูปของ "x" และในไฟล์ etc/shadow จะเก็บในรูปของไซเฟอร์เท็กซ์ ซึ่งเป็นข้อมูลที่ไม่สามารถอ่านหรือตีความได้ด้วยตาเปล่า

7.4 การเข้ารหัสแบบต่างๆ (ในที่นี้จะขอยกตัวอย่างวิธีการเข้ารหัสคร่าวๆ แค่ 2 วิธีต่อไปนี้)

7.4.1. การเข้ารหัสแบบ DES (Data Encryption Standard)

DES คือ การเข้ารหัสข้อมูลโดยใช้อัลกอริทึมทางคณิตศาสตร์ ซึ่งจะใช้ข้อมูลขนาด 64 บิตและกุญแจ (key) ขนาด 64 บิต (โดย 8 บิต เพิ่มเข้ามาเพื่อใช้ในการตรวจสอบความผิดพลาด) ซึ่งในการเข้าและและการถอดรหัสนั้นได้ใช้ระบบการเข้ารหัสแบบสมมาตร (Symmetric Cryptosystem) คือ ใช้กุญแจตัวเดียวกันในการเข้าและถอดรหัส

สำหรับกระบวนการถอดรหัสนั้น ทำเช่นเดียวกันกับกระบวนการเข้ารหัส คือ ใส่ข้อมูลที่ถูกเข้ารหัสข้อมูลมาแล้วที่มีขนาด 64 บิต พร้อมกับกุญแจตัวเดิมที่ใช้ในการเข้ารหัส แล้วทำตามขั้นตอนต่างๆ เหมือนกับการเข้ารหัสข้อมูล แต่ต้องทำการเปลี่ยนลำดับของกุญแจใหม่ โดยใช้กุญแจจากลำดับท้ายสุดเรียงมาจนถึงลำดับแรกสุด คือ K16, K15, … , K1

7.4.2 การแปลงข้อมูลโดยใช้การคำนวณแบบ MD5 (Message Standard)

เป็นอัลกอริทึมที่รับข้อความที่มีความยาวไม่จำกัดเข้ามาแล้วได้ผลลัพธ์ออกมาเป็นเมสเซจไดเจสต์ (Message Digest) ที่มีขนาด 128 บิต โดยอาศัยฟังก์ชันทางคณิตศาสตร์ที่เป็นฟังก์ชันทางเดียวมาใช้ในการทำงานของอัลกอริทึมนี้ ซึ่งในการที่ใช้ฟังก์ชันทางเดียวนั้น หมายถึงว่ามีความเป็นไปได้ยากมากที่จะมีข้อความ 2 ข้อความใดที่สามารถสร้างออกมาเป็นเมสเซจไดเจสต์ได้เหมือนกัน และไม่สามารถสร้างข้อความใดๆ จากเมสเซจไดเจสต์นี้ได้เลย

ในส่วนของการถอดรหัสนั้นมิสามารถทำได้เหมือนกับ DES นั่นคือ ไม่สามารถใช่กุญแจตัวเดิมได้ แต่จะใช้วิธีเดาคำๆ หนึ่ง แล้วนำคำนั้นมาเข้ารหัส ซึ่งจะได้เมสเซจไดเจสต์ออกมา แล้วนำมาเปรียบเทียบกับเมสเซจไดเจสต์ที่มีอยู่แล้ว ถ้าหากมันเหมือนกันจะแสดงว่าคำที่เดามาก็คือรหัสผ่านนั่นเอง

7.5 หลักในการแคร็กรหัสผ่าน

7.5.1 ดิกชันนารีแอ็ตแท็ก

7.5.2 การ Brute Force

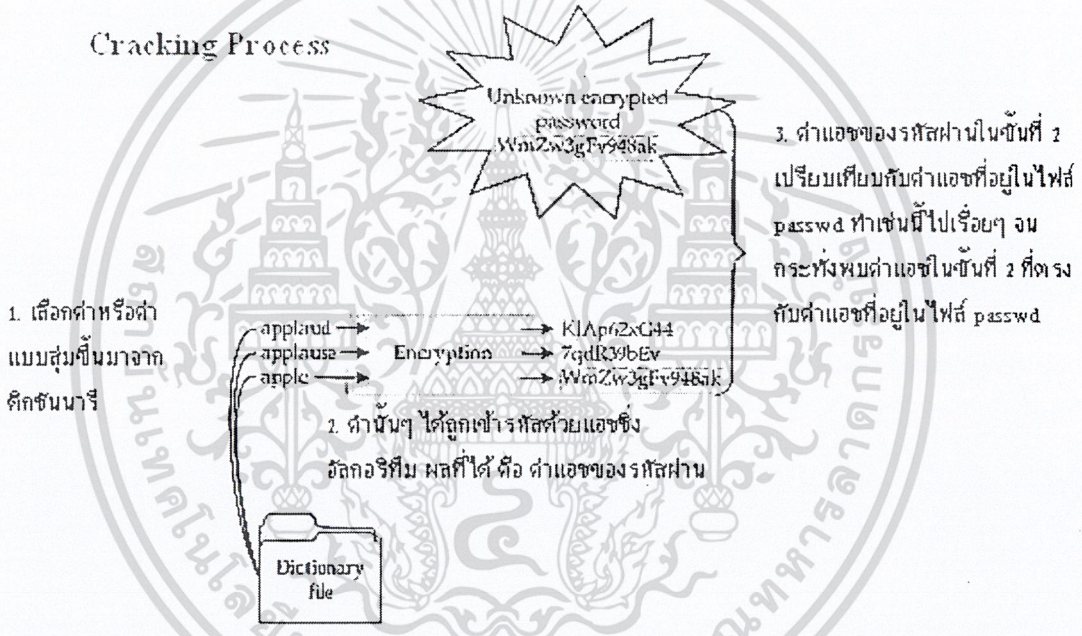
7.5.1 ดิกชันนารีแอ็ตแท็ก

การโจมตีด้วยการคาดเดารหัสผ่านจากคำในดิกชันนารีโดยอัตโนมัติ (automated dictionary attack) ซึ่งมักจะเรียกกันง่ายๆ ว่า "ดิกชันนารีแอ็ตแท็ก" การคาดเดารหัสสามารถกระทำแบบออฟไลน์ (offline) ได้คือ เดาที่เครื่องของตนเองโดยไม่ต้องทดลองล็อกอินไปยังเซิร์ฟเวอร์ เพื่อทำให้ผู้ดูแลระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สังเกตเห็นได้ยาก นี่ถือว่าเป็น โลคอลแอ็ตแท็ก (Local Attack) ก็เพราะว่าแฮ็กเกอร์ต้องได้ไฟล์ /etc/passwd หรือไฟล์รหัสผ่านที่ถูกชาโดว์ไว้มา (/etc/shadow)

สำหรับการแคร็กรหัสผ่านด้วยวิธีนี้นั้น ทำได้โดยพยายามเดารหัสผ่านของแต่ละแอคเคาน์โดยการใช้โปรแกรมแคร็กเข้ารหัสคำต่างๆ ที่มีอยู่ในดิกชันนารีหรือเข้ารหัสคำต่างๆ ที่ผสมผสานกับแบบสุ่ม (ผลที่ได้ คือ ค่าแฮชของรหัสผ่านที่ถูกคำนวณโดยโปรแกรมแคร็กรหัสผ่าน) แล้วเปรียบเทียบดูผลลัพธ์ที่ได้กับค่าแฮชของรหัสผ่านที่เก็บอยู่ในไฟล์ /etc/shadow ว่าเท่ากันหรือไม่ ถ้าค่าแฮชของรหัสผ่านที่โปรแกรมแคร็กรหัสผ่านอัตโนมัติคำนวณได้ ตรงกันกับค่าแฮชของรหัสผ่านที่เก็บอยู่ในไฟล์ /etc/passwd นั่นแสดงว่า คำในดิกชันนารีหรือคำแบบสุ่มนั้นๆ ที่โปรแกรมแคร็กเลือกขึ้นมาคือรหัสผ่านของผู้ใช้งาน กระบวนการนี้สามารถอธิบายได้ด้วยรูปที่ 7-1



รูปที่ 7-1 : การแคร็กรหัสผ่านทำงานอย่างไร

7.5.2 การ Brute Force

เป็นเทคนิคที่พื้นฐานที่สุดของการแคร็กรหัส และเป็นวิธีที่มีประสิทธิภาพที่สุดวิธีหนึ่ง จริงๆ แล้วการวิธีการ brute force ก็คือการคาดเดารหัสผ่านโดยทดลองผสมผสานตัวอักษร ตัวเลขต่างๆ เข้าด้วยกันแล้วเดาว่าใช่รหัสผ่านหรือไม่ โดยทั่วไปแล้ว ตัวอักษรและตัวเลข สัญลักษณ์พิเศษที่ผสมผสานขึ้นมานั้น จะเป็นคำที่ไม่มีในดิกชันนารี

ถึงแม้เราจะคาดเดารหัสผ่านด้วยตนเองได้ก็ตาม แต่เครื่องมือสำหรับแคร็กรหัสผ่านด้วยเทคนิคนี้แบบอัตโนมัติก็มีอยู่หลายตัว เครื่องมือเหล่านี้ได้แก่

- brutus http://www.hoobie.net/brutus
- brute_web.c http://packetstorm.security.com/Exploit_Code_Archive/brute_web.c
- pop.c http://packetstorm.security.com/groups/ADM/ADM-pop.c

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- middlefinger http://www.njh.com/lastest/9709/970916-05.html

- Teenet http://www.phenoelit.de/tn/

7.6 ตัวอย่างโปรแกรมที่ใช้ตรวจสอบการแคร็กรหัสผ่าน

จากความรู้ที่ว่า แฮ็กเกอร์ส่วนใหญ่จะพยายามดิ้นรนให้ตัวเองได้รับเชล (shell) ของยูนิกซ์มาเสมือนว่านั่งอยู่ที่หน้าจอเครื่องโดยตรง เพื่อให้ได้สิทธิเท่าเทียมกับรูตก็ตาม และแม้ว่าส่วนใหญ่แล้วแฮ็กเกอร์จะมุ่งความสนใจไปที่รูต แต่หลายครั้งที่แฮ็กเกอร์อาจต้องการแอคเคานต์ธรรมดาอื่นที่ไม่ใช่รูต อย่างเช่น ถ้าต้องการขโมยข้อมูลจากฐานข้อมูลออราเคิล แฮ็กเกอร์อาจต้องการแค่เพียง Oracle ID เท่านั้น ไม่ถึงกับต้องให้ได้รูต ถ้ารหัสผ่านที่ตั้งไว้ไม่ดีพอ ความเสี่ยงของช่องโหว่นี้จะมีมากที่สุด

โปรแกรมแคร็กรหัสผ่านที่ดีที่สุดสองโปรแกรม ได้แก่ Crack 5.0a จาก AlecMuffett และ John the Ripper จาก Solar Designer Crack 5.0a หรือเรียกสั้นๆ ได้ว่า "Crack" เป็นโปรแกรมแคร็กที่ได้รับความนิยมมากที่สุดในขณะนี้และได้รับการพัฒนาอย่างต่อเนื่องตั้งแต่มันเริ่มประกาศตัวออกมา Crack มาพร้อมกับไฟล์ดิกชันนารีที่สมบูรณ์และครบถ้วนมาก คำที่บรรจุอยู่ภายในนั้นมีตั้งแต่ คำสั้นๆ หนึ่งหรือสองตัวอักษรไปจนกระทั่งถึงคำยาวๆ และคำผสมต่างๆ เช่น Star Trek ยิ่งไปกว่านั้น Crack ยังได้ให้ความสามารถในการกระจายโหลดหรือภาระของการแคร็กไปยังเครื่องคอมพิวเตอร์อื่นๆ ในเน็ตเวิร์กด้วยเพื่อช่วยทำให้การแคร็กเร็วขึ้น

ส่วน John the Ripper หรือเรียกสั้นๆ ว่า "John" นั้นเป็นโปรแกรมที่ใหม่กว่า Crack 5.0a และได้รับการปรับแต่งมาเป็นอย่างดีสำหรับการแคร็กรหัสผ่านจำนวนมากในเวลาอันสั้นมาก นอกจากนั้นประเภทของแฮชชิ่งอัลกอริทึมที่ John รู้จักยังมีมากกว่า Crack ด้วย ทั้ง John และ Crack ต่างก็ให้เครื่องมืออำนวยความสะดวกในการนำเอาคำต่างๆ ที่มีอยู่ในดิกชันนารีมาผสมผสานกันเป็นคำ คำใหม่ๆ หรือคำแปลกๆ ที่ไม่มีในดิกชันนารีโดยตรง โดยดีฟอลต์ เครื่องมือทั้งสองตัวนี้มีกฎกว่า 2,400 กฎที่ใช้ในการผสมผสานคำต่างๆ จากดิกชันนารีเข้าด้วยกันเป็นคำๆ ใหม่ เพื่อให้สามารถแคร็กรหัสผ่านที่ค่อนข้างยากและไม่ใช่คำในดิกชันนารีตรงๆ ได้ อีกทั้งเครื่องมือทั้งสองก็ยังได้ให้เอกสารประกอบการใช้งานซึ่งน่าสนใจทีเดียว แทนที่เราจะกล่าวกันอยู่เฉพาะฟีเจอร์ของแต่ละเครื่องมือ เราจะไปดูวิธีการรัน Crack และรีวิวดูผลลัพธ์ที่ได้กัน สิ่งสำคัญอย่างหนึ่งที่คุณควรคุ้นเคยก็คือ ไฟล์ที่เก็บรหัสผ่านอยู่ที่ไหนและชื่ออะไร รวมถึงโครงสร้างภายในไฟล์เป็นอย่างไรบ้างแต่ละคอลัมน์หมายถึงอะไร

7.6.1 Crack 5.0a

การรัน Crack กับไฟล์รหัสผ่านเป็นเรื่องง่ายมาก เพียงแต่โหลดไฟล์รหัสผ่านให้มันแล้วก็รอดูผลลัพธ์เท่านั้นเอง Crack เป็นโปรแกรมที่คอมไพล์ได้ในตัวเอง และเมื่อถูกเอ็กซีคิวต์แล้ว มันจะเริ่มต้นสร้างองค์ประกอบบางตัวที่จำเป็นต่อการทำงานขึ้นมาเอง หนึ่งในจุดแข็งของ Crack ก็คือ กฎต่างๆ จำนวนมากที่นำมาใช้ในการผสมผสานคำใหม่ๆ ได้ นอกจากนั้นในแต่ละครั้งที่มันเอ็กซีคิวต์ มันจะสร้างไฟล์เวิร์ดลิสต์ (wordlist) ใหม่ๆ ขึ้นมาให้ด้วย ซึ่งมักเป็นรายชื่อผู้ใช้งานรวมทั้งข้อมูลต่างๆ ในคอลัมน์ GECOS หรือในคอลัมน์คอมเมนต์ อย่าได้มองข้ามความสำคัญของคอลัมน์ GECOS เป็นอันขาด เพราะในขณะที่มันกำลังแคร็กรหัสผ่านมันจะเอาตัวอักษรในคอลัมน์นี้มาผสมผสานกันเพื่อเอามาเดาเป็นรหัสผ่าน ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนใหญ่ในคอลัมน์นั้นจะเป็นรายละเอียดของผู้ใช้งาน อย่างเช่น ชื่อเต็มๆ (full name) จะประกอบด้วย ชื่อต้น นามสกุล ที่อยู่ สามารถนำมาผสมกันแล้วสืบหารหัสผ่านที่ตั้งไว้ไม่ค่อยดีได้อย่างรวดเร็ว

```
root:cwIBREDaWLHmo:0:0:root:/root:/bin/bash
bin:*:1:1:bin:/bin
daemon:*:2:2:daemon:/sbin
<other locked accounts omitted>
nobody:*:99:99:Nobody:/:
eric:GmTFg0AavFA0U:500:0::/home/eric:/bin/csh
samantha:XaDeasK8g8g3s:501:503::/home/samantha:/bin/bash
temp:kRWegG5iTZP5o:502:506::/home/temp:/bin/bash
hackme:nh.StBNcQnyE2:504:1::/home/hackme:/bin/bash
bob:9wynbWzXinBQ6:506:1::/home/bob:/bin/bash
es:0xUH89TiymLcc:501:501::/home/es:/bin/bash
mother:jxZd1tcz3wW2Q:505:505::/home/mother:/bin/bash
jfr:kyzKR0ryhFDE2:506:506::home/jfr:/bin/bash
```

รูปที่ 7-2 : สำเนาไฟล์ /etc/shadow

ส่วนการเอ็กซิคิวต์โปรแกรม Crack กับสำเนาไฟล์รหัสผ่านข้างต้น เราจะรันคำสั่งต่อไปนี้

```
[tsunami] # Crack passwd
Crack 5.0a: The Password Cracker.
(c) Alec Muffett, 1991, 1992, 1993,1994, 1995, 1996
System: Linux 2.0.36 #1 Tue Oct 13 22:17:11 EDT 1998 i686 unknown
<omitted for brevity>
Crack: The dictionaries seem up to date...
Crack: Sorting out and merging feedback, please be patient...
Crack: Merging password files...
Crack: Creating gecos-derived dictionaries
mkgecosd: making non-permuted words dictionary
mkgecosd: making permuted words dictionary
Crack: launching: cracker –kill run/system.11324
Done
```

รูปที่ 7-3 : แสดงการใช้คำสั่งในการรันโปรแกรม Crack

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาถึงจุดนี้ Crack กำลังทำงานในฉากหลังหรือแบ็คกรานด์ และได้บันทึกรหัสผ่านที่มันแคร็กได้ลงในไฟล์ฐานข้อมูลของมันเอง ถ้าต้องการดูไฟล์ฐานข้อมูลนี้และหาว่ารหัสผ่านใดได้ถูกแคร็กไปแล้วบ้าง เราจะต้องรันโปรแกรม Reporter :

```
[tsunami] # Reporter –quiet
- - - - passwords cracked as of Sat 13:09:50 EDT - - - -

Guessed eric [jenny]          [passwd /bin/csh]
Guessed hackme [hackme]       [passwd /bin/bash]
Guessed temp [temp]           [passwd /bin/bash]
Guessed es [eses]             [passwd /bin/bash]
Guessed jfr [solaris1]        [passwd /bin/bash]
```

รูปที่ 7-4 : ผลลัพธ์จะแสดงหลังจากรันโปรแกรม Reporter

เราได้แสดงรหัสผ่านทั้งหมดที่มันแคร็กได้จนถึงขณะนี้โดยการใช้ออปชั่น -quiet ถ้าคุณเอ็กซิคิวต์ Report โดยไม่ระบุออปชั่นอะไร มันจะแสดงทั้งข้อผิดพลาด คำเตือนและรหัสผ่านที่ถูกล็อคเอาไว้ด้วย มีสคริปต์อยู่หลายตัวที่ให้มาพร้อมกับ Crack ที่มีประโยชน์มากพอสมควร หนึ่งในสคริปต์เหล่านั้นได้แก shadmrg.sv สคริปต์ตัวนี้จะถูกใช้เพื่อรวมเอาไฟล์รหัสผ่านของยูนิกซ์เข้ากับไฟล์ชาโดวส์ (shadow file) เพื่อให้ข้อมูลที่เกี่ยวข้องทั้งหมดถูกรวมเข้าเป็นไฟล์เดียวเพื่อสะดวกต่อการแคร็ก ส่วนคำสั่งอื่นที่น่าสนใจได้แก่ make tidy ซึ่งถูกใช้เพื่อยกเลิกแอคเคาน์ตของผู้ใช้งานและรหัสผ่านที่แคร็กเสร็จแล้วออกไปจากไฟล์รหัสผ่าน

สิ่งสุดท้ายที่ควรกล่าวถึงก็คือ การเรียนรู้วิธีการค้นหาชื่ออัลกอริทึมที่ใช้ในการเข้ารหัสรหัสผ่านไฟล์รหัสผ่านที่เราทดสอบข้างต้นนี้ได้ใช้อัลกอริทึมของ DES เพื่อเข้ารหัสรหัสผ่าน ซึ่งเป็นอัลกอริทึมมาตรฐานบนยูนิกซ์ส่วนใหญ่ แต่เมื่อพิจารณาถึงการเพิ่มระดับการรักษาความปลอดภัยแล้ว ผู้ผลิตยูนิกซ์ส่วนใหญ่จะใช้อัลกอริทึม MD5 และ blowfish แทน ค่าแฮชของรหัสผ่านที่คำนวณได้จากอัลกอริทึมของ MD5 จะมีความยาวมากกว่าค่าแฮชของรหัสผ่านที่คำนวณได้จากอัลกอริทึมของ DES และค่าแฮชที่ได้จะมีตัวอักขระ “$1” สองตัวนำหน้า ในทำนองเดียวกัน อัลกอริทึมของ blowfish จะมี “$2” นำหน้า

7.6.2 John the Ripper

John the Ripper เป็นเครื่องมือในการแคร็กรหัสผ่านที่ดีที่สุดอันหนึ่งและใช้งานง่ายมาก ในการรัน John ให้โหลดไฟล์รหัสผ่านให้มันก่อน (/etc/shadow) และสั่งให้มันเริ่มทำงาน ก่อนอื่นมันจะแยกแยะก่อนว่าอัลกอริทึมที่ใช้เข้ารหัสเป็นอัลกอริทึมใด ในตัวอย่างของเราเป็น DES และเริ่มคาดเดารหัส

เอกสารนี้มีไว้เพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ทำซ้ำ คัดลอก ดัดแปลง เผยแพร่ หรือนำไปใช้ประโยชน์ทางการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้นำไฟล์นี้ไปเผยแพร่ต่อในทางสาธารณะ

ผ่าน โดยอันดับแรก มันจะเริ่มเดาจากคำในไฟล์ดิกชันนารี (password.lst) ถ้าไม่พบคำในดิกชันนารี มันจะเปลี่ยนไปใช้การเดาแบบ Brute Force

```
[shadow]# john passwd
Loaded 9 passwords with 9 different salts (Standard DES [24/32 4K])
hacker           (hackme)
temp             (temp)
oscs             (es)
jenny            (esic)
t78              (bob)
guesses: 5  time: 0:00:04:26 (3)  c/s: 16270  trying: pirath - stuart
```

รูปที่ 7-5 : การรัน john

สามารถหารายละเอียดได้ที่ http://www.openwall.com/john/.

เมื่อเปรียบเทียบกันระหว่าง Crack ในตัวอย่างที่แล้วและ John ในตัวอย่างนี้ เราจะเห็นได้ว่าเวอร์ชันเก่าของ John สามารถเดารหัสผ่านของผู้ใช้งานชื่อ Bob ได้ ขณะที่ Crack สามารถเดารหัสผ่านของผู้ใช้งานชื่อ jfr ได้ ผลลัพธ์ที่ได้จึงแตกต่างกันไปสำหรับแต่ละโปรแกรม

7.7 ไฟล์สำคัญต่างๆ ใน /etc ที่เกี่ยวกับรหัสผ่าน

ในระบบลีนุกซ์นั้น ข้อมูลเกี่ยวกับผู้ใช้งานและกลุ่มของผู้ใช้งานจะถูกบันทึกเก็บไว้เป็นเท็กซ์ไฟล์ต่างๆ ภายในไดเรกทอรี /etc/ เมื่อผู้ดูแลระบบทำการเพิ่มผู้ใช้งานขึ้นในระบบ รายละเอียดของผู้ใช้งานคนนั้นก็จะถูกเพิ่มเติมลงในไฟล์เหล่านั้นด้วย

7.7.1 ไฟล์ /etc/passwd

7.7.1.1 /etc/passwd คืออะไร ?

ไฟล์นี้ทุกคนสามารถอ่านได้และเป็นที่เก็บรายละเอียดต่างๆ ของผู้ใช้งานทุกคน รวมถึงข้อมูลเบื้องต้นเกี่ยวกับตัวผู้ใช้งานเอง เช่น ชื่อเต็ม หมายเลขประจำตัว หมายเลขกลุ่มผู้ใช้งาน เป็นต้น ซึ่งจะแยกกันคนละบรรทัด ในหนึ่งบรรทัดจะประกอบไปด้วย 7 คอลัมน์ และจะคั่นด้วยเครื่องหมาย ":"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
rpm:x:37:37::/var/lib/rpm:/bin/bash
mailnull:x:47:47::/var/spool/mqueue:/sbin/nologin
smmsp:x:51:51::/var/spool/mqueue:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
pcap:x:77:77::/var/arpwatch:/sbin/nologin
xfs:x:43:43:X Font Server:/etc/X11/fs:/sbin/nologin
named:x:25:25:Named:/var/named:/sbin/nologin
gdm:x:42:42::/var/gdm:/sbin/nologin
desktop:x:80:80:desktop:/var/lib/menu/kde:/sbin/nologin
postgres:x:26:26:PostgreSQL Server:/var/lib/pgsql:/bin/bash
apache:x:48:48:Apache:/var/www:/sbin/nologin
postfix:x:89:89::/var/spool/postfix:/sbin/nologin
mysql:x:27:27:MySQL Server:/var/lib/mysql:/bin/bash
ohh:x:500:500:phunsiri Ngammaneewat:/home/ohh:/bin/bash
panny:x:505:501:panny:/home/panny:/bin/bash
ann:x:506:506:Anny narak:/home/ann:/bin/bash
orn:x:507:507:Piyaon Kongsuktakul:/home/orn:/bin/bash
hitomi:x:508:508:Viraiwan Madee:/home/hitomi:/bin/bash
[root@isag07 home]#
```

รูปที่ 7-6 : ตัวอย่างไฟล์ /etc/passwd

7.7.1.2 ความหมายของแต่ละคอลัมน์ในไฟล์ /etc/passwd

Account : Password : UID : GID : GECOS : Home Directory : Command	
Account	เป็นบัญชีชื่อของผู้ใช้งานและยังใช้เป็นชื่อล็อกอินด้วย
Password	รหัสผ่านที่เข้ากระบวนการสร้างรหัสแล้ว ซึ่งเราไม่สามารถเปลี่ยนรหัสผ่านได้โดยการใช้ editor นอกจากทำการลบรหัสผ่าน การเปลี่ยนรหัสต้องใช้คำสั่ง passwd เท่านั้น
UID	หมายเลขประจำตัวของผู้ใช้งาน ไม่ควรจะซ้ำกันด้วย เนื่องจาก UID จะเป็นหมายเลขที่บ่งบอกว่าใครเป็นใคร และไฟล์ที่ถูกสร้างขึ้นต้องมีการระบุว่าผู้ใช้งานคนไหนเป็นเจ้าของไฟล์ตาม UID ดังนั้นถ้าเรากำหนด UID ซ้ำกัน ผู้ใช้งานรายอื่นจะมีสิทธิเป็นเจ้าของไฟล์ตามที่ UID นั้นสร้างไว้ได้
GID	หมายเลขประจำกลุ่มผู้ใช้งาน มีการอ้างอิงกับไฟล์ /etc/group ซึ่ง GID เป็นการระบุว่าผู้ใช้งานอยู่ในกลุ่มไหน โดยจะมีผลในเรื่องสิทธิการเข้าถึงไฟล์ในระดับกลุ่มได้ GID จึงสามารถกำหนดซ้ำกับผู้ใช้งานรายอื่นได้
GECOS (General Electric Comprehensive Operating System)	เป็นข้อความแสดงรายละเอียดเกี่ยวกับผู้ใช้งาน โดยทั่วไปจะใส่เป็นชื่อ และนามสกุลของผู้ใช้งานแทน แต่อาจจะใส่เป็นที่อยู่ เบอร์โทร วันเดือนปีเกิด ก็ได้
Home Directory	เมื่อผู้ใช้งานล็อกอินเข้ามาในระบบแล้ว โฮมไดเรกทอรีจะเป็นไดเรกทอรีแรกที่เริ่มใช้งาน และเป็นไดเรกทอรีที่ผู้ใช้งานแต่ละรายมีสิทธิในการสร้างและลบไฟล์ได้อย่างอิสระ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Command	เป็นคำสั่งแรกที่ถูกเรียกขึ้นมาใช้หลังจากผ่านการล็อกอินเข้ามาแล้ว โดยปกติของผู้ใช้งานจะกำหนดให้เป็น shell ใดๆ เช่น /bin/bash, /bin/sh เป็นต้น แต่ถ้าเป็นบัญชีที่กำหนดให้ทำงานเฉพาะอย่างก็อาจไม่ได้กำหนดให้ใช้ shell command แต่จะระบุเป็นคำสั่งอื่น เช่น บัญชีชื่อ shutdown มีการกำหนดคำสั่งแรกเป็น /sbin/shutdown นั่นคือ เมื่อล็อกอินเข้ามาด้วยบัญชีชื่อผู้ใช้งาน shutdown จะทำการปิดระบบให้

ตารางที่ 7-2 : รูปแบบของไฟล์ /etc/passwd

7.7.2 ไฟล์ /etc/shadow

7.7.2.1 /etc/shadow คืออะไร ?

เป็นไฟล์ที่สามารถอ่านได้เฉพาะผู้ดูแลระบบเท่านั้น จะเก็บรหัสผ่านของผู้ใช้งานแต่ละคน และส่วนอื่นๆ ที่เกี่ยวกับรหัสผ่าน ซึ่งจะสัมพันธ์กับไฟล์ /etc/passwd ด้วย สำหรับรหัสผ่านจะเป็นรหัสที่ผ่านการเข้ากระบวนการสร้างรหัสแล้ว (Encoding) ดังนั้นจึงไม่สามารถอ่านเข้าใจได้

แต่เดิมนั้นรหัสผ่านจะถูกเก็บไว้ที่ไฟล์ /etc/passwd ในคอลัมน์ที่ 2 แต่เมื่อมีการพัฒนาต่อมาเพื่อความปลอดภัยมากขึ้น จึงได้แยกเก็บรหัสผ่านไว้อีกไฟล์ นั่นคือไฟล์ /etc/shadow ซึ่งต้องมีบัญชีผู้ใช้งานเหมือนกับไฟล์ /etc/passwd โดยไฟล์ /etc/passwd ก็ยังคงมีคอลัมน์ของรหัสผ่านอยู่แต่ค่าที่เห็นจะเป็น “x” แทน

```
sshd:!!:11983:0:99999:7:::
rpm:!!:11983:0:99999:7:::
mailnull:!!:11983:0:99999:7:::
smmsp:!!:11983:0:99999:7:::
rpcuser:!!:11983:0:99999:7:::
nfsnobody:!!:11983:0:99999:7:::
pcap:!!:11983:0:99999:7:::
xfs:!!:11983:0:99999:7:::
named:!!:11983:0:99999:7:::
gdm:!!:11983:0:99999:7:::
desktop:!!:11983:0:99999:7:::
postgres:!!:11983:0:99999:7:::
apache:!!:11983:0:99999:7:::
postfix:!!:11983:0:99999:7:::
mysql:!!:11983:0:99999:7:::
ohh:$1$TF4McO.k$25V/mQOJdODK1273OjqYX1:11984:0:99999:7:::
panny:$1$92lORxGF$v9DyMVtyrzMvXwgxbW7g5.:12037::99999::::
ann:$1$Wn/C1cpY$F/cTUtcsFxB1fKAffKlFjO:12076:0:99999:7:::
orn:$1$GnE9fML1$UJf/5Wj5VgT.VnoWG8OGI/:12076:0:99999:7:::
hitomi:$1$neB.qWq0$P.eoFZuWbx2fI8PnV1LS./:12076:0:99999:7:::
[root@isag07 home]#
```

รูปที่ 7-7 : ตัวอย่างไฟล์ /etc/shadow

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.7.2.2 ความหมายของแต่ละคอลัมน์ในไฟล์ /etc/shadow

Account : Password : Lastchg : Min : Max : Warn : Inactive : Expire : Flag	
Account	บัญชีชื่อของผู้ใช้งาน ซึ่งจะเหมือนกับไฟล์ /etc/passwd ทุกประการ
Password	รหัสผ่านที่เข้ากระบวนการสร้างรหัสแล้ว จะอ้างอิงกับคอลัมน์ที่ 2 ของไฟล์ /etc/passwd
Lastchg	ค่าที่แสดงว่ามีการเปลี่ยนแปลงรหัสผ่านครั้งล่าสุดเมื่อไร
Min	จำนวนวันที่น้อยที่สุดที่ยอมให้มีการเปลี่ยนรหัสได้
Max	จำนวนวันที่มากที่สุดที่รหัสผ่านจะยังสามารถใช้งานได้
Warn	จำนวนวันที่มีการเตือนผู้ใช้งานก่อนที่รหัสผ่านจะหมดอายุ
Inactive	จำนวนวันที่ยอมให้ผู้ใช้งานไม่มีการแก้ไขอะไรเลย
Expire	วันที่แน่นอนที่รหัสผ่านหมดอายุ และผู้ใช้งานจะไม่สามารถล็อกอินเข้ามาได้อีก
Flag	เป็นคอลัมน์ที่ Red Hat ไม่สนใจ (ยังไม่มีการใช้งาน)

ตารางที่ 7-3 : รูปแบบของไฟล์ /etc/shadow

7.7.3 ไฟล์ /etc/group

7.7.3.1 /etc/group คืออะไร ?

เป็นไฟล์แบบอ่านได้อย่างเดียว และเป็นที่เก็บของรายชื่อกลุ่มต่าง ๆ ของผู้ใช้งาน หมายเลขกลุ่ม รวมถึงสมาชิกรายย่อยด้วย การเก็บจะแยกกันคนละบรรทัด ซึ่งแต่ละบรรทัดจะมี 4 คอลัมน์ ใช้เครื่องหมาย ":" ในการแยกเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
sshd:x:74:
rpm:x:37:
mailnull:x:47:
smmsp:x:51:
slocate:x:21:
rpcuser:x:29:
nfsnobody:x:65534:
pcap:x:77:
xfs:x:43:
named:x:25:
gdm:x:42:
desktop:x:80:
postgres:x:26:
apache:x:48:
postdrop:x:90:
postfix:x:89:
mysql:x:27:
ohh:x:500:
panny:x:501:
ann:x:506:
orn:x:507:
hitomi:x:508:
[root@isag07 home]#
```

รูปที่ 7-8 : ตัวอย่างไฟล์ /etc/group

7.7.3.2 ความหมายของแต่ละคอลัมน์ในไฟล์ /etc/group

Group_Name : Password : GID : Members	
Group_Name	บัญชีชื่อกลุ่ม
Password	เป็นรหัสผ่านซึ่งโดยปกติจะไม่ได้ใช้
GID	เป็นหมายเลขประจำกลุ่ม ซึ่งจะใช้อ้างอิงไฟล์ /etc/passwd ด้วย สำหรับผู้ใช้งานที่กำหนด GID เหมือนกันจะมีสิทธิในระดับกลุ่มเท่ากัน ค่ามากที่สุดที่เป็นได้คือ 2, 137, 483, 647 แต่ค่าที่เหมาะสมควรกำหนดไม่เกิน 60000
Members	เป็นการกำหนดสมาชิกภายในกลุ่ม โดยปกติในไฟล์ /etc/passwd จะมีการกำหนด GID ให้กับผู้ใช้งานทั่วไปแล้ว แต่ถ้าผู้ใช้งานบางคนจำเป็นต้องเป็นสมาชิกในหลายๆ กลุ่ม เราสามารถกำหนดเพิ่มในคอลัมน์นี้ได้เช่นกัน

ตารางที่ 7-4 : รูปแบบของไฟล์ /etc/group

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.7.4 ไฟล์ /etc/gshadow

7.7.4.1 /etc/gshadow คือ อะไร ?

เป็นไฟล์ที่อ่านได้โดยผู้ดูแลระบบอย่างเดียวและเก็บ password ที่เข้ารหัสของแต่ละกลุ่มไว้ ซึ่งเก็บได้ดีเช่นเดียวกับข้อมูลของสมาชิกในแต่ละกลุ่มและผู้ดูแลระบบ การเก็บก็จะแยกเก็บทีละบรรทัด ในแต่ละบรรทัดแบ่งเป็น 4 คอลัมภ์ แยกด้วยเครื่องหมาย ":"

```
sshd:x::
rpm:x::
mailnull:x::
smmsp:x::
slocate:x::
rpcuser:x::
nfsnobody:x::
pcap:x::
xfs:x::
named:x::
gdm:x::
desktop:x::
postgres:x::
apache:x::
postdrop:x::
postfix:x::
mysql:x::
ohh:!::
panny:!!::
ann:!::
orn:!::
hitomi:!::
```

รูปที่ 7-9 : ตัวอย่างไฟล์ /etc/gshadow

7.7.4.2 ความหมายของแต่ละคอลัมน์ในไฟล์ /etc/gshadow

Group_Name : Encrypted password : Group administrators : Group members	
Group_Name	ชื่อของกลุ่ม
Encrypted password	รหัสผ่านที่เข้ารหัสแล้วของกลุ่ม
Group administrators	รายชื่อสมาชิกของกลุ่มผู้ดูแล ถ้ามีหลายคนให้ใช้คอมม่าคั่น สามารถเพิ่มหรือลดสมาชิกด้วยการใช้คำสั่ง gpasswd
Group members	รายชื่อสมาชิกของกลุ่ม ถ้ามีหลายคนให้ใช้คอมม่าคั่น

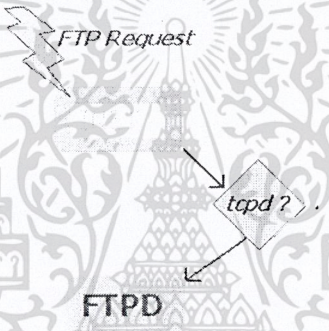
ตารางที่ 7-5 : รูปแบบของไฟล์ /etc/gshadow

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8
การตั้งค่าในระบบ

8.1 inetd คืออะไร

inetd เป็นโปรแกรมที่ควบคุมโปรแกรมที่ให้บริการในระบบเครือข่ายบนระบบปฏิบัติการยูนิกซ์ เมื่อมีการเรียกใช้บริการที่โปรแกรม inetd ควบคุมอยู่ เช่น ftp, telnet เป็นต้น (ผ่านการควบคุมของไฟล์ /etc/inetd.conf) และถูกกำหนดให้เรียกใช้โปรแกรม tcpd เพื่อตรวจสอบความถูกต้องของไอพีแอดเดรสที่เรียกใช้บริการว่า ถูกต้องตามกฎที่ตั้งไว้ในไฟล์ /etc/hosts.allow และ /etc/hosts.deny หรือไม่ หากถูกต้องโปรแกรม inetd จึงจะเรียกใช้โปรแกรมที่ให้บริการบนระบบเครือข่ายดังกล่าวนั้นอีกทอดหนึ่ง ซึ่งการทำงานของกระบวนการดังกล่าวจะเป็นไปตามรูปที่ 8-1



รูปที่ 8-1 : แสดงขั้นตอนการทำงานของโปรแกรม INETD และ TCPD เมื่อมีการเรียกใช้งาน ftp จากเครือข่าย

8.2 xinetd คืออะไร

ในระบบปฏิบัติการลีนุกซ์บางระบบ เช่น Red Hat เวอร์ชันตั้งแต่ 7.0 ขึ้นไปเป็นต้น จะติดตั้งโปรแกรม xinetd ให้ทำงานแทนการใช้งานโปรแกรม inetd และ tcpd เดิม เนื่องจากโปรแกรม xinetd มีคุณสมบัติที่น่าสนใจ คือ เป็นโปรแกรมที่ให้บริการผ่านโพรโตคอล TCP, UDP, RPC สามารถควบคุมช่วงเวลาการเปิดให้ใช้งานบริการต่างๆ ได้ มีความสามารถในการสร้างไฟล์ล็อกได้เองโดยไม่ต้องผ่านโปรแกรมsyslog สามารถจำกัดจำนวนผู้ใช้งาน และทรัพยากรของระบบเพื่อลดปัญหาจากการบุกรุกแบบ Denial of Services (DoS) สามารถปรับเปลี่ยนการเชื่อมต่อ (Redirect connection) และการกำหนดการเชื่อมต่อ (Bind Connection) ของแต่ละบริการได้

ซึ่งเมื่อเปรียบเทียบกันแล้วระหว่างโปรแกรม inetd และ xinetd แล้ว จะเห็นได้ชัดเจนว่าโปรแกรม xinetd นั้นมีจุดเด่นและข้อดีมากกว่าโปรแกรม inetd

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.3 ส่วนประกอบของ xinetd

ไฟล์หลักที่กำหนดการทำงานของโปรแกรม xinetd ประกอบด้วยสองส่วนคือ

8.3.1 /etc/xinetd.conf เป็นไฟล์ที่ระบุค่าของระบบโดยรวมของโปรแกรม xinetd ภายในไฟล์นี้จะประกอบด้วยส่วนของค่าของระบบที่เรียกว่า ดีฟอลต์เซกชัน (defaults section) ซึ่งเป็นค่าของระบบโดยรวมของทุกบริการที่อยู่ภายใต้ xinetd มีรูปแบบดังนี้

```
defaults
{
        attribute operator value(s)
        ...
}
```

รูปที่ 8-2 : โครงสร้างไฟล์ /etc/xinetd.conf

ตัวอย่างไฟล์ /etc/xinetd.conf มีลักษณะดังนี้

```
[root@isag07 root]# cat /etc/xinetd.conf
#
# Simple configuration file for xinetd
#
# Some defaults, and include /etc/xinetd.d/

defaults
{
        instances               = 60
        log_type                = SYSLOG authpriv
        log_on_success          = HOST PID
        log_on_failure          = HOST
        cps                     = 25 30
}

includedir /etc/xinetd.d

[root@isag07 root]#
```

รูปที่ 8-3 : ไฟล์ /etc/xinetd.conf

8.3.2 /etc/xinetd.d/ เป็นไฟล์ที่เก็บค่าคอนฟิกของเซอร์วิสทุกตัวที่ถูกควบคุมโดย xinetd ซึ่งภายในแต่ละไฟล์จะประกอบด้วยค่าของระบบในรูปแบบดังต่อไปนี้

```
service service_name
{
        attribute operator value(s)
        ...
}
```

รูปที่ 8-4 : โครงสร้างไฟล์ /etc/xinetd.d ของแต่ละไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตั้งค่าของระบบในไฟล์นี้มีรูปแบบเหมือนกับไฟล์ /etc/xinetd.conf แต่ค่าระบบในส่วนนี้จะมีผลต่อเฉพาะแต่ละบริการ (แทนที่ service_name ด้วยชื่อของบริการ เช่น telnet หรือ ssh เป็นต้น)

เดมอน inetd มีความสามารถในการเริ่มต้นการทำงานของบริการ 2 รูปแบบ บริการแบบ standard (standard service) ทำงานผ่าน inetd โดยอาศัยช่องทางของพอร์ตตามที่ได้รับ ซึ่งมักจะเป็นพอร์ตที่รู้จักโดยทั่วไป (ตัวอย่างเช่น telnetd ทำงานผ่านพอร์ต 23 หรือ ftpd ทำงานผ่านพอร์ต 21) รายการของบริการแบบนี้สามารถพบได้ในไฟล์ **/etc/services** ซึ่งถือเป็นไฟล์สำคัญที่แสดงรายการของบริการทั้งหมดในระบบ รวมถึงพอร์ตและโพรโตคอลที่บริการแต่ละชนิดใช้งาน บริการอีกชนิดหนึ่งที่สามารถเรียกใช้งานได้คือ TCPMUX นำมาจาก RFC 1078 บริการชนิด TCPMUX ถือเป็นบริการแบบ non-standard ทำงานผ่านพอร์ตที่แตกต่างกันออกไป การจัดให้บริการแต่ละชนิดอยู่ภายใต้การควบคุมของ inetd จะช่วยลดการใช้งานทรัพยากรของระบบ เนื่องจากเดมอนของบริการแต่ละอันจะถูก inetd สั่งให้เริ่มต้นทำงานก็ต่อเมื่อมีการร้องขอ (หรือสอบถาม) เพื่อใช้งานบริการดังกล่าวเกิดขึ้นเท่านั้น

8.4 กฎเกณฑ์ และค่าของระบบที่สำคัญ

ไฟล์ระบบของโปรแกรม xinetd กำหนดให้มีตัวกระทำ (Operator) ทั้งหมด 3 ตัว ดังตารางที่ 8-1

ตัวกระทำ	ความหมาย
=	กำหนดค่าของระบบใดๆ ให้เท่ากับค่าที่ระบุ
+=	เพิ่มค่าที่ระบุให้กับค่าของระบบ
-=	ลบค่าที่ระบุให้กับค่าของระบบ

ตารางที่ 8-1 : ตัวกระทำของไฟล์ระบบในโปรแกรม xinetd

ค่าคุณสมบัติที่น่าสนใจต่อระบบในโปรแกรม xinetd แสดงได้ดังตารางที่ 8-2

ค่าคุณสมบัติ	ความหมาย
Flags	บ่งบอกคุณสมบัติเพิ่มเติมของแต่ละ service โดยค่า และความหมายของคุณสมบัตินี้ขึ้นกับเวอร์ชันของโปรแกรม xinetd สามารถศึกษาได้จากคำสั่ง man xinetd.conf ตัวอย่างของค่าคุณสมบัติ เช่น • NODELAY : ถ้า service เป็นประเภท tcp service แล้ว API socket จะตั้งค่า TCP_NODELAY เพื่อใช้ในการเชื่อมต่อของแต่ละ service ระหว่างการใช้งาน IPv6 : เพื่อบ่งบอกว่า service นั้นให้บริการบนระบบเครือข่ายทีซีพี/ไอพีชนิด IPV6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Log_type	บ่งบอกถึงรูปแบบการเก็บล็อกของ service ต่างๆ ที่ภายใต้โปรแกรม xinetd มีสองแบบคือ • SYSLOG syslog_facility [syslog_level] ผลของล็อกจะเก็บผ่าน syslog ที่บ่งบอก facility จากค่า syslog_facility ได้แก่ daemon, auth, authpriv, user และ local0-7 ค่าระดับของล็อก (Log level) จะบ่งบอกผ่านค่า syslog_level ได้แก่ emerg, alert, crit, err, warning, notice, info และ debug • FILE file [soft_limit [hard_limit]] ผลของล็อกจะเก็บในไฟล์ที่กำหนดผ่านชื่อไฟล์ file โดยมีคุณสมบัติเพิ่มเติม คือ soft_limit, hard_limit
log_on_success	บ่งบอกรายละเอียดข้อมูลการใช้งานของแต่ละ service ที่จะถูกเก็บสู่ไฟล์ล็อก หลังจากที่เริ่มทำงาน หรือภายหลังจากผู้ใช้งานผ่านขั้นตอนการล็อกอิน • PID : ค่า Process ID ของ service • HOST : ไอพีของผู้ใช้งาน service • USERID : User ID ของผู้ใช้งาน service • EXIT : เวลาที่ผู้ใช้ยกเลิกการใช้งาน service • DURATION : ช่วงเวลาการใช้งาน service
log_on_failure	บ่งบอกรายละเอียดข้อมูลการใช้งานของแต่ละ service ที่จะถูกเก็บสู่ไฟล์ล็อก เมื่อการร้องขอใช้ service ผิดไปจากกฎระเบียบที่กำหนดในค่าระบบของโปรแกรม xineted • HOST, USERID : มีความหมายเดียวกับค่าใน **log_on_success** • ATTEMPT : บ่งบอกถึงรายระเอียด และประเภทของการพยายามเรียกใช้งาน service ที่ผิดพลาด
no_access	บ่งบอกถึงเครื่องลูกข่ายที่ไม่ต้องการให้ใช้งาน service สามารถบ่งบอกได้ในรูปแบบของ Domain Name หรือไอพี คล้ายกับการกำหนดในไฟล์ hosts.deny ของ tcpwrapper
only_from	บ่งบอกถึงเครื่องลูกข่ายที่ต้องการให้ใช้งาน service สามารถบ่งบอกได้ในรูปแบบเดียวกับ **no_access**
port	บ่งบอกถึงหมายเลขพอร์ตที่ service เปิดให้บริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

protocol	ชนิดของโพรโตคอลของ service
server	บ่งบอกถึงไฟล์โปรแกรมของ service
server_args	ค่าอาร์กิวเมนต์เพิ่มเติมของโปรแกรม ในคุณสมบัติ **server**
instances	บ่งบอกจำนวนโพรเซสสูงสุดของ service ที่ทำงานในช่วงเวลาเดียวกัน
disable	บ่งบอกว่า service มีการเปิดให้บริการหรือไม่ ถ้าค่าเป็น yes หมายถึงงดให้บริการ และ no หมายถึงเปิดให้บริการ
cps	บ่งบอกถึงจำนวนครั้งของการเชื่อมต่อสูงสุด ต่อวินาที
access_times	บ่งบอกช่วงเวลาที่เปิดให้บริการ service โดยรูปแบบของค่าคือ *ชั่วโมง:นาที(ที่เริ่มให้บริการ)-ชั่วโมง:นาที(ที่หยุดให้บริการ)* เช่น ต้องการเปิดให้บริการตั้งแต่ 12 นาฬิกา ถึง 13 นาฬิกา ค่าของระบบคือ access_time = 12:00 - 13:00
rlimit_cpu	บ่งบอกถึงจำนวนของ CPU seconds สูงสุดที่ให้กับแต่ละ service
rlimit_stack	บ่งบอกถึงจำนวนของ stack สูงสุดที่ให้กับแต่ละ service
per_source	บ่งบอกจำนวนการเชื่อมต่อสูงสุดของการใช้งาน service จากเครื่องลูกข่ายเดียวกัน

ตารางที่ 8-2 แสดงค่าคุณสมบัติที่น่าสนใจของระบบในโปรแกรม xinetd

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างของไฟล์ในไดเรกทอรี /etc/xinetd.d/ ที่กำหนดการให้บริการของโปรแกรม ftp

```
service ftp
{
        socket_type = stream
        wait = no
        user = root
        server = /usr/sbin/in.ftpd
        server_args = -l
        instances = 4
        access_times = 7:00-12:30 13:30-21:00
        nice = 10
        only_from = 192.168.1.0/24
}
```

รูปที่ 8-5 : ไฟล์ของโปรแกรม ftp ที่อยู่ใน /etc/xinetd.d/

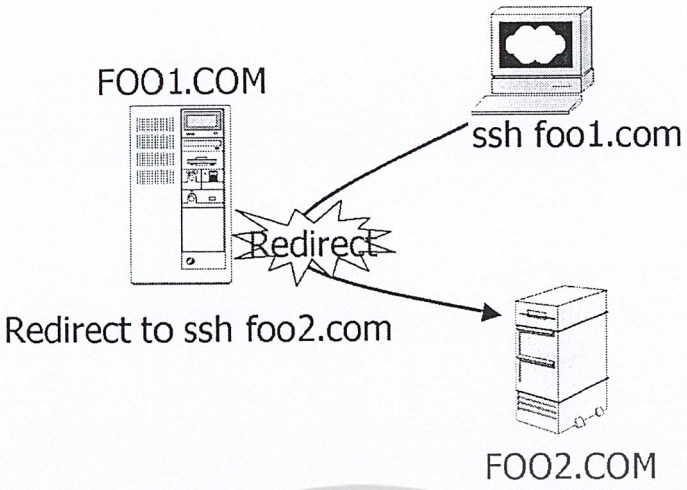
ค่าคุณสมบัติ	ความหมาย
instances = 4	กำหนดให้ยอมรับการขอใช้บริการ ftp พร้อมกันได้สูงสุด 4 โปรเซส
access_times = 7:00-12:30 13:30-21:00	การให้บริการ ftp จะเปิดให้ผู้ใช้งานสามารถใช้ได้เฉพาะเวลา 7:00-12:30 และ 13:30-21:00
only_from = 192.168.1.0/24	ยอมรับการใช้บริการผ่านไอพีที่มาจากผู้ใช้งานในวง 192.168.1.0/24
log_type = FILE /var/log/servicelog	กำหนดให้เก็บไฟล์ล็อกของการใช้งาน xinetd ที่ /var/log/servicelog แทน /var/log/message
log_on_success = HOST PID USERID DURATION EXIT	การกำหนดค่าที่ต้องการเก็บในไฟล์ล็อก หลังจากการเข้าใช้ประสบความสำเร็จ
log_on_failure = HOST USERID RECORD	การกำหนดค่าที่ต้องการเก็บในไฟล์ล็อก หลังจากการเข้าใช้ที่ล้มเหลว

ตารางที่ 8-3 : แสดงความหมายของไฟล์ ftp ข้างต้น

8.5 คุณสมบัติเพิ่มเติมของ xinetd กับการปรับเปลี่ยนการเชื่อมต่อ (Redirect Features)

การปรับเปลี่ยนการเชื่อมต่อของการขอใช้บริการผ่านโปรแกรม xinetd เป็นคุณสมบัติอย่างหนึ่งที่เป็นประโยชน์มาก ตัวอย่างของการใช้งานคุณสมบัตินี้แสดงได้ตามรูปที่ 8-6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8-6 : แสดง การร้องขอ ssh บน xinetd ที่อาศัยคุณสมบัติ redirect

จากรูปที่ 8-6 เครื่องลูกข่ายร้องขอการใช้บริการโปรแกรม ssh จากเครื่อง foo1.com ซึ่งถูกกำหนดให้การให้บริการโปรแกรม ssh ถูกควบคุมโดยโปรแกรม xinetd และกำหนดให้การร้องขอการใช้บริการโปรแกรม ssh ปรับเปลี่ยนไปใช้บริการที่เครื่อง foo2.com แทน ดังนั้นเสมือนว่าเครื่องลูกข่ายได้ขอใช้บริการโปรแกรม ssh จากเครื่อง foo1.com แต่ที่จริงแล้วการให้บริการนี้จะถูกควบคุมจากเครื่อง foo2.com โดยที่การเชื่อมต่อระหว่างเครื่องลูกข่ายกับเครื่อง foo1.com ก็ยังคงอยู่ กระบวนการเช่นนี้คล้ายคลึงกับการทำงานของ Transparent Proxy

ตัวอย่างไฟล์ /etc/xinetd.d/ssh ของเครื่อง foo1.com ที่รองรับการทำงานตามตัวอย่างในรูปที่ 8-6 คือ

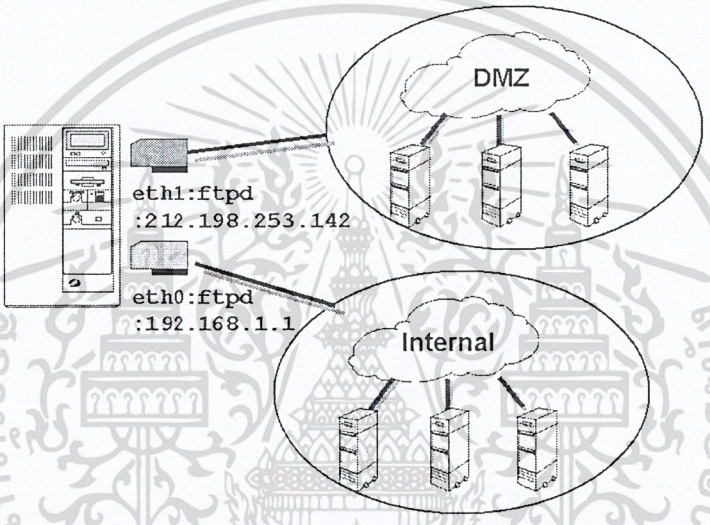
```
service ssh
{
        socket_type = stream
        wait = no
        user = root
        server = /usr/sbin/sshd
        server_args = -i
        log_on_success += DURATION USERID
        log_on_failure += USERID
        nice = 10
        redirect = foo2.com 22
        disable = yes
}
```

รูปที่ 8-7 : ไฟล์ /etc/xinetd.d/ssh ตามรูปที่ 8-6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

8.6 คุณสมบัติเพิ่มเติมของ xinetd กับการกำหนดการเชื่อมต่อ (Bind Feature)

โปรแกรม xinetd มีคุณสมบัติในการกำหนดการเชื่อมต่อของไอพีในแต่ละบริการได้ การทำงานนี้จะมีประโยชน์กับเครื่องที่มีอุปกรณ์เชื่อมต่อระบบเครือข่าย (Network Interface) อย่างน้อยสองตัวขึ้นไป ตัวอย่างเช่น หากในองค์กรหนึ่งต้องการเปิดให้บริการ FTP ภายใน และภายนอกองค์กร แนวทางการใช้โปรแกรม xinetd กับความต้องการนี้ คือ การแยกการให้บริการ FTP ออกเป็นสองส่วน คือ ส่วนของภายใน และภายนอกองค์กร ผ่านการตั้งค่า bind (ต้องกำหนดค่า id เสมอเพื่อแยกแยะความแตกต่างของ FTP ทั้งสองส่วน) รูปที่ 8-8 แสดงรูปแบบการทำงานของโปรแกรม xinetd ที่ให้บริการ FTP server ซึ่งกำหนดให้ไอพีภายในเป็น 192.168.1.1 และไอพีภายนอกเป็น 212.198.253.142



รูปที่ 8-8 : แสดงการเปิดให้บริการ FTP ทั้งภายใน และภายนอกองค์กร ที่ใช้คุณสมบัติ bind

ตัวอย่างไฟล์ /etc/xinetd.d/ftp ที่รองรับการทำงานตามตัวอย่างในรูปที่ 8-8 คือ

```
service ftp
{
        id = ftp-public
        wait = no
        user = root
        server = /usr/sbin/in.ftpd
        server_args = -l
        instances = 4
        nice = 10
        only_from = 0.0.0.0/0
        bind = 212.198.253.142
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
service ftp
{
        id = ftp-internal
        socket_type = stream
        wait = no
        user = root
        server = /usr/sbin/in.ftpd
        server_args = -l
        only_from = 192.168.1.0/24
        bind = 192.168.1.1
}
```

รูปที่ 8-9 : ไฟล์ /etc/xinetd.d/ftp ตามรูปที่ 8-8

8.7 บริการที่จำเป็น

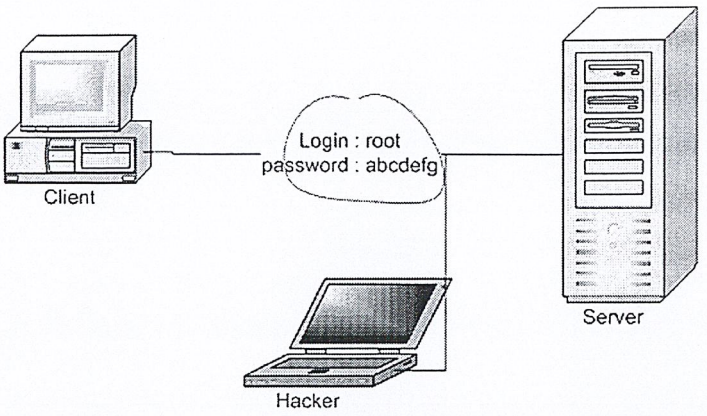
เมื่อพิจารณาไฟล์ inetd.conf ประกอบด้วยรายการของบริการต่างๆ มากมาย บริการกลุ่มแรกที่จะยกมาอธิบายคือ บริการที่จำเป็น (vital) ซึ่งบริการที่จำเป็นในที่นี้ หมายความถึงบริการที่เป็นมาตรฐานและมีส่วนสำคัญต่อทุกระบบปฏิบัติการยูนิกซ์ที่จะต้องนำมาใช้งานในเครือข่าย

บริการที่จำเป็นที่สุดสองอันคือ **telnetd** และ **ftpd** อย่างไรก็ตาม บริการทั้งสองอันเป็นการส่งข้อมูลโดยไม่ได้รับการเข้ารหัส ส่งผลให้เครื่องให้บริการอาจถูกละเมิดการใช้งานได้ง่าย โดยเฉพาะอย่างยิ่งในทุกวันนี้ ที่ไม่มีเครือข่ายใดปลอดภัย 100% ดังนั้นผู้ใช้งานจึงต้องระมัดระวังต่อการส่งข้อมูลผ่านโพรโตคอลที่ปราศจากการเข้ารหัสเป็นพิเศษ รายละเอียดเพิ่มเติมเกี่ยวกับบริการเหล่านี้สามารถอ่านศึกษาเพิ่มเติมได้จากเอกสาร RFC หนังสือ หรือสื่ออื่นๆ ได้ จึงจะไม่นำมาอธิบายในที่นี้

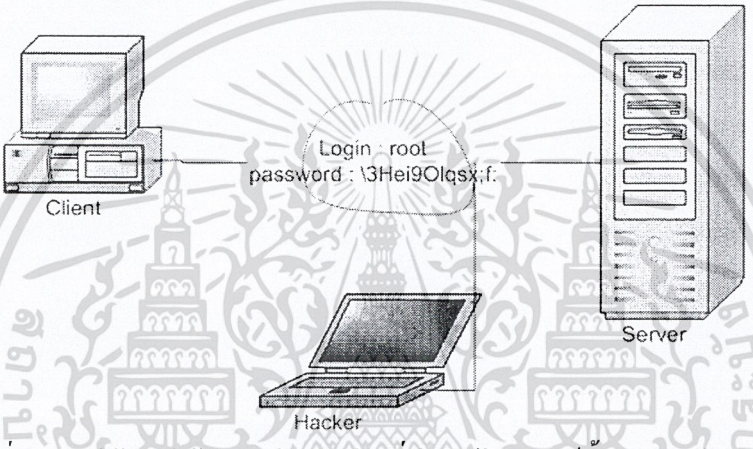
แต่ในบางสถานการณ์ อาจไม่มีทางเลือกอื่นนอกจากการเปิดใช้งานบริการเหล่านั้นผู้ใช้งานจึงควรจำกัดความเสี่ยงโดยนำเอาไฟร์วอลล์มาใช้งาน ให้ติดต่อผ่านเครือข่าย VPN (Virtual Private Network) ที่ได้รับการเข้ารหัส จำกัดการเข้าใช้งาน กำหนดให้ใช้ rsh (restricted shell) หรือใช้ tcp wrappers

อย่างไรก็ตาม อีกวิธีการหนึ่งที่สมควรจะนำมาใช้งานคือ ใช้การติดต่อผ่านโพรโตคอล SSH การนำวิธีการเข้ารหัสมาใช้กับการติดต่อนั้น จะช่วยให้การส่งข้อมูลได้รับความปลอดภัยทั้งจากภายในและภายนอก ในปัจจุบัน SSH ได้ถูกนำมาใช้งานแทนที่ telnetd และ ftpd เพื่อช่วยเพิ่มความปลอดภัยให้กับเครือข่ายทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8-10 : ใช้ telnet ในการทำ remote access ไปยังเซิร์ฟเวอร์และผู้โจมตีสามารถดักข้อมูลได้



รูปที่ 8-11 : ใช้ ssh ให้สังเกตว่ารหัสผ่านที่ส่งจากไคลเอ็นต์นั้นจะถูกเข้ารหัสไว้

8.8 บริการอื่นๆ

ผู้ใช้งานบางคนอาจจะรู้สึกว่า "ในไฟล์ inetd.conf ประกอบด้วยบริการต่างๆ มากมาย" และเกิดคำถามว่า "บริการใดบ้างที่จำเป็น หรือเป็นประโยชน์" ในความเป็นจริงแล้วบริการอื่นๆ ที่เหลือเปรียบเสมือนเครื่องมือช่วยในการใช้งานระบบปฏิบัติการยูนิกซ์ วิธีการที่จะตัดสินใจได้ว่าบริการใดบ้างที่ต้องใช้งาน คือ ทำการศึกษาว่าบริการนั้นๆ ทำหน้าที่ใด และใช้งานเพื่ออะไร โดยเริ่มต้นตั้งแต่บริการแรก ไปจนถึงบริการสุดท้ายที่ปรากฏในไฟล์ inetd.conf ซึ่งได้ติดตั้งพร้อมกับตอนติดตั้งระบบปฏิบัติการ

หลังจาก telnetd และ ftpd ที่ได้กล่าวถึงแล้ว บริการต่อไปคือ **name** ทำงานบนโพรโตคอล udp เมื่อถูกร้องขอให้ทำงานจะไปเรียกใช้งานเดมอนชื่อ **/usr/sbin/in.tnamed** โดย in.tname พัฒนาขึ้นมาจาก DARPA เป็นเดมอนที่ทำงานตามมาตรฐาน IEN116 เดิม ซึ่งเป็นต้นกำเนิดของบริการ **DNS** ในปัจจุบัน ทำให้บริการ **in.tnamed** นี้ไม่ถูกนำมาใช้งานอีกต่อไป

บริการต่อมาคือ **shell** หรือ **remote shell** ซึ่ง **rsh** นี้ทำงานบนโพรโตคอล tcp ผ่านพอร์ต 514 เพื่ออนุญาตให้ผู้ใช้งานติดต่อเข้าใช้งาน พิสูจน์ตัวตนหรือเรียกใช้งานคำสั่งจากระยะไกลมายังเครื่องนั้นๆ ได้ มีลักษณะการให้บริการคล้ายคลึงกับ telnet แต่วิธีการสั่งงานเหมือนกับการทำงานผ่าน shell อย่างไรก็ตามบริการนี้มีข้อบกพร่องด้านความปลอดภัยอย่างมาก เนื่องจากเป็นบริการที่ทำงานตาม access control

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

list (ACL) 2 ชนิดคือ รายการในไฟล์ **/etc/hosts.equiv** และไฟล์ **$HOMEDIR/.rhosts** หลักการทำงานคือ ถ้าผู้ใช้งานคนใดต้องการติดต่อไปยังเครื่องผ่านทาง rsh ผู้ใช้งานคนดังกล่าวต้องได้รับการเพิ่มชื่อเครื่องที่จะใช้ลงในไฟล์ **/etc/hosts.equiv** หากไม่เป็นตามนี้ เครื่องจะตรวจสอบในระดับของผู้ใช้งาน โดยการค้นหาชื่อผู้ใช้งาน และชื่อเครื่องที่ขอติดต่อเข้ามาว่ามีอยู่ในไฟล์ **.rhost** หรือไม่ การทำงานในลักษณะดังกล่าวนี้ส่งผลให้เกิดปัญหา 2 ข้อ ข้อแรกคือ หากคอมพิวเตอร์เครื่องใดในเครือข่ายถูกละเมิดการใช้งาน โดยที่ชื่อเครื่องดังกล่าวถูกกำหนดไว้ในไฟล์ **hosts.equiv** หรือ **.rhost** ของเครื่องอื่นในเครือข่าย จะทำให้เครื่องที่เหลือเหล่านั้นถูกละเมิดการใช้งานด้วย ข้อที่สองเกิดขึ้นจากการถูกปลอมแปลง (spoof) ชื่อผู้ใช้งานและ/หรือชื่อเครื่อง ซึ่งเป็นเหตุการณ์ที่สามารถเกิดขึ้นได้และป้องกันได้ยาก ปัญหาอีกข้อหนึ่งของ rsh ก็คือไม่มีการนำเอาวิธีการเข้ารหัสใดๆ มาใช้ในการส่งข้อมูลระหว่างเครื่อง ทำให้เกิดการดักจับข้อมูลในเครือข่ายที่ใช้งาน เพื่อนำมาใช้ขโมยเทอร์มินอลหรือดักเก็บรหัสผ่านของผู้ใช้งานคนอื่นๆ เพื่อให้สามารถเข้าใช้งานระบบได้ นอกจากนั้น ข้อมูลทั้งหมดที่ใช้ในการพิสูจน์ตัวตนถูกส่งไปมาโดยไม่ได้รับการเข้ารหัส ทั้งหมดนี้ถือเป็นข้อบกพร่องของ rsh

ถัดมาคือ **login** หรือ **remote login** ใช้เดมอนชื่อ **rlogind** หลักการทำงานของ **rlogin** มีรูปแบบเดียวกับ **rsh** ระบบที่ต้องการติดต่อจากระยะไกลจะถูกกำหนดไว้ในไฟล์ **/etc/hosts.equiv** หรือกำหนดชื่อผู้ใช้ในไฟล์ **.rhost** ภายในไดเรกทอรี home ของแต่ละคน ปัญหาที่เกิดขึ้นจากการใช้งานก็เช่นเดียวกันกับปัญหาของ rsh ได้แก่ ข้อบกพร่องจากการพิสูจน์ตัวตนผู้ใช้ การปลอมแปลง การละเมิดการใช้งานเครื่องที่ได้รับการ trust และขาดวิธีการเข้ารหัสข้อมูล

rexec เป็นบริการที่ทำงานบนโพรโตคอล tcp ผ่านพอร์ต 512 เพื่อใช้ในการเรียกใช้งานคำสั่งตามที่ต้องการจากระยะไกลไปยังเดมอนที่รองรับ **rexec** จะเลือกใช้วิธีการพิสูจน์ตัวตนผู้ใช้งานเพื่อสร้างการติดต่อและรอรับการสั่งงานได้ 2 วิธี วิธีแรก ผู้ใช้งานจะส่งไฟล์ **.netrc** ซึ่งเก็บข้อมูลของตัวผู้ใช้งานเพื่อแสดงตัวตนให้เครื่องที่จ้องการใช้งานทราบ หรือวิธีที่สอง เมื่อเกิดการสร้างการติดต่อ เดมอนภายในเครื่องจะรอรับชื่อและรหัสผ่านของผู้ใช้งาน อย่างไรก็ตาม ปัญหาที่สำคัญของด้านความปลอดภัยของเดมอนนี้เหมือนกับปัญหาของ rsh และ rlogin

comsat เป็นเดมอนที่ทำหน้าที่แจ้งเกี่ยวกับอี-เมล์ ทำงานโดยใช้โพรโตคอล udp ทำงานผ่านพอร์ต 512 มีประโยชน์ต่อเครื่องให้บริการที่ทำงานเป็นเกตเวย์สำหรับการดำเนินงานอื่นๆ หรือเครื่องเวิร์กสเตชันที่ใช้รับอี-เมล์ส่งต่อทั้งหมดของผู้ใช้งานแต่ละคน ถึงแม้ว่าบริการนี้จะช่วยอำนวยความสะดวกต่อผู้ใช้งาน แต่ไม่มีความสำคัญถึงระดับที่จำเป็นต่อระบบ วิธีการทำงานของบริการนี้คือ จะรอรับอี-เมล์ที่ถูกส่งเข้ามาที่พอร์ต 512 ทันทีที่อี-เมล์แต่ละฉบับมาถึง ตัวเดมอนจะแสดงข้อความ 7 บรรทัดแรกของเนื้อหาในอี-เมล์ให้ปรากฏบนหน้าจอ

talk หรือ **talkd** เป็นบริการที่ใช้งาน udp อีกอันหนึ่ง ทำงานผ่านพอร์ต 517 บริการนี้จะทำให้ผู้ใช้งานสามารถติดต่อเข้ามายังเครื่องและสร้าง talk session กับผู้ใช้งานคนอื่นๆ ที่กำลังใช้งานเครื่องคอมพิวเตอร์เครื่องอื่นได้จากระยะไกล ยังไม่มีการนำขั้นตอนการสร้างความปลอดภัยมาใช้กับบริการนี้แต่อย่างใด ปัญหาสำคัญที่เกิดขึ้นจากบริการนี้ก็คือ ผู้ใช้งานจากภายนอกสามารถเข้ามาพูดคุยกับผู้ที่ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องได้ ส่งผลให้เกิดกระบวนการ social engineering โดยการล่อหลอกถามชื่อผู้ใช้งานและรหัสผ่านที่จะเข้าใช้งานเครื่องของผู้ใช้งานเครื่องคนนั้น แล้วนำข้อมูลที่ได้มาใช้ในการเข้าสู่ระบบได้

uucp หรือ **Unix to Unix copy protocol** เป็นบริการของ tcp ที่ทำงานบนพอร์ต 540 บริการนี้เป็นบริการที่เก่าและไม่เหมาะสมต่อการใช้งาน จึงถูกใช้งานแทนด้วย **SMTP** และ **NNTP** แล้วทั้งหมด โดยปัญหาหลักของบริการนี้คือ ข้อมูลที่ส่งขาดการเข้ารหัส ก่อให้เกิดความเสี่ยงที่จะถูกดักจับรหัสผ่าน

tftp หรือ **trivial file transfer protocol** บริการนี้ใช้งานโพรโตคอล udp ผ่านพอร์ต 69 สำหรับบางที พบว่าบริการนี้จำเป็นต่อการใช้งานเมื่อต้องการบูตเครื่องคอมพิวเตอร์ชนิด diskless client การเปิดใช้งาน tftp นำไปสู่ความเสี่ยงด้านความปลอดภัยอย่างรุนแรงเนื่องจากสาเหตุ ข้อแรกมาจากกระบวนการทำงานของ tftp ที่ไม่มีขั้นตอนการพิสูจน์ตัวตนผู้ใช้งานใดๆ เลย ข้อที่สอง เครื่องให้บริการ tftp จะพยายามเปลี่ยนไดเรกทอรี home เป็น **/tftpboot** ถึงแม้ว่าจะไม่มีไดเรกทอรีดังกล่าวอยู่ในเครื่องก็ตาม และไม่สามารถคาดเดาถึงเหตุการณ์ที่จะตามมาได้ ข้อที่สาม เดมอน tftpd อนุญาตให้ผู้ใช้งานแต่ละคนอ่านไฟล์ทุกไฟล์ที่เผยแพร่แบบอนุญาตให้อ่านโดยใช้ชื่อเป็น nobody และยังอนุญาตให้ผู้ใช้งานแต่ละคนเขียนไฟล์ทุกไฟล์ที่เผยแพร่แบบอนุญาตให้เขียน หากจำเป็นต้องเปิดใช้งานบริการนี้ ให้ตามด้วยออปชัน **-s** เสมอเพื่อบังคับว่าการเปลี่ยนแปลงไดเรกทอรีที่เกิดขึ้นจะต้องสำเร็จทุกครั้ง (อ่านรายละเอียดเพิ่มเติมได้จากหน้า manual ของโปรแกรม)

finger, systat และ **netstat** โดย **finger** เป็นบริการที่อนุญาตให้ผู้อื่นจากภายนอกเข้ามาค้นหาข้อมูลของผู้ใช้งานภายในเครื่อง ทำงานผ่านพอร์ต 79 ส่วน **systat** จะแสดงรายการของโพรเซสที่กำลังทำงานอยู่บนเครื่องให้ผู้ใช้งานจากภายนอกรับทราบ ทำงานผ่านพอร์ต 11ของโพรโตคอล tcp และ **netstat** ทำงานบนโพรโตคอลผ่านพอร์ต 15 โดยทำหน้าที่แสดงข้อมูลการติดต่อกับเครือข่ายทั้งหมดที่เกิดขึ้นกับเครื่องในขณะนั้น บริการทั้งสามอันนี้ก่อให้เกิดความเสี่ยงด้านความปลอดภัยที่เห็นได้อย่างชัดเจน เนื่องจากทำให้ผู้อื่นจากภายนอกรับทราบข้อมูลการทำงานที่สำคัญของเครื่อง ซึ่งสามารถนำไปใช้เพื่อโจมตีระบบได้

time เป็นโพรเซสชนิดเดมอนที่ทำงานภายในระบบ ทำงานผ่านอาศัยพอร์ต 37 ทั้งของโพรโตคอล tcp และ udp เพื่อจัดการกับเวลาของเครื่อง โดยใช้การคำนวณเปรียบเทียบค่าเวลาที่เปลี่ยนไปในหน่วยวินาที จากเวลาเที่ยงคืนของวันที่ 1 มกราคมปี 1900 การอนุญาตให้ผู้อื่นจากภายนอกใช้งานข้อมูลเวลาทำให้คนเหล่านั้นทราบว่าควรจะโจมตีเครื่องเมื่อใด เพราะทราบได้ว่าขณะใดน่าจะเป็นช่วงเวลาที่มีคนใช้งานเครื่องน้อย หรือผู้ดูแลระบบไม่ใช้งานเครื่อง

echo ใช้งานพอร์ต 7 ของ tcp และ udp ทำงานตาม RFC 862 ซึ่งถูกนำมาใช้เป็นเครื่องมือในการประเมินคุณภาพและตรวจสอบเครือข่าย ทำงานโดยการส่งกลับข้อมูลที่ได้รับไปยังต้นทาง การเปิดใช้งานบริการนี้ส่งผลให้เครื่องถูกโจมตีแบบ Denial of Service ได้โดยง่าย

discard เป็นการทำงานที่เกิดขึ้นบนพอร์ต 9 ของทั้ง tcp และ udp บริการนี้เกิดขึ้นจาก RFC 863 และถูกนำมาใช้เพื่อเป็นเครื่องมือตรวจสอบระบบ ทำหน้าที่ส่งคืนข้อมูลทุกอันที่เครื่องได้รับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

daytime มีการทำงานคล้ายคลึงกับ time ยกเว้น รูปแบบของข้อมูลที่มนุษย์สามารถเข้าใจได้ ทำงานผ่านพอร์ต 13

chargen เป็นการทำงานที่เกิดขึ้นบนพอร์ต 19 อาศัยทั้งโพรโตคอล tcp และ udp ในขณะที่ทำงานภายใต้ tcp บริการ **chargen** จะรอรับการติดต่อ และเมื่อมีการติดต่อจากเครื่องอื่นเกิดขึ้น ตัวเดมอนของ chargen จะส่งข้อมูลในรูปแบบของ stream แบบต่อเนื่องไปยังเครื่องที่ติดต่อมา จนกระทั่งเครื่องต้นทางที่ติดต่อมาตัดขาดการติดต่อ ส่วนการทำงานภายใต้ udp บริการ **chargen** จะรอรับ datagram จากเครือข่าย หลังจากที่ได้รับ ตัวเดมอนจะทำการตอบกลับด้วยดาต้าแกรมที่มีข้อมูลขนาด 0 ถึง 512 ตัวอักษร วิธีการทำงานดังกล่าวนี้นำไปสู่การโจมตีแบบ Denial of Service

บริการสุดท้ายที่จะกล่าวถึงคือ เดมอนของเครื่องพิมพ์ (**printer**) การทำงานของเดมอนเครื่องพิมพ์จะทำผ่านพอร์ต 515 โดยใช้โพรโตคอล tcp ถือเป็นเครื่องมือที่เป็นประโยชน์ ทำให้สามารถพิมพ์งานจากเครื่องเวิร์กสเตชันไปที่เครื่องพิมพ์ที่อยู่ในเครือข่ายได้ ปัญหาที่เกิดขึ้นจากการใช้งานคือเรื่องความปลอดภัย หากต้องการสร้างความปลอดภัยให้กับระบบที่ใช้จะต้องติดตั้งเครื่องพิมพ์และทรัพยากรที่เกี่ยวข้องไว้ในเครือข่ายภายใน หรือนำวิธีการของ ACL (access control list) มาใช้ป้องกันเครื่องพิมพ์และเครื่องให้บริการที่เกี่ยวข้อง

8.9 ตัวอย่างการตั้งค่าในโปรแกรม sshd

การปรับแต่งจะมุ่งเน้นไปที่การปรับแต่งค่าของ SSHD เป็น daemon ของฝั่งเซิร์ฟเวอร์ที่รอรับการติดต่อจากไคลเอ็นต์ ให้มีความปลอดภัยมากที่สุด โดยจะทำการแก้ไขในไฟล์ที่ชื่อ /etc/ssh/sshd_config ซึ่งรายละเอียดในไฟล์ที่สำคัญๆ มีดังนี้ และให้ลบ # ที่อยู่หน้า option ต่อไปนี้

```
# This is ssh server systemwide configuration file.

Port 22
ListenAddress 192.168.1.1
HostKey /etc/ssh/ssh_host_key
ServerKeyBits 1024
LoginGraceTime 600
KeyRegenerationInterval 3600
PermitRootLogin no
IgnoreRhosts yes
IgnoreUserKnownHosts yes
StrictModes yes
X11Forwarding no
PrintMotd yes
SyslogFacility AUTH
LogLevel INFO
RhostsAuthentication no
RhostsRSAAuthentication no
RSAAuthentication yes
PasswordAuthentication yes
PermitEmptyPasswords no
AllowUsers admin
Subsystem sftp /usr/libexec/openssh/sftp-server
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 8-12 : การคอนฟิกค่าในไฟล์ /etc/ssh/sshd_config

คำแนะนำในการปรับแต่งค่าต่างๆ ที่สำคัญในไฟล์ /etc/ssh/sshd_config ให้มีความปลอดภัย มีดังนี้

Port 22	ระบุหมายเลขพอร์ตที่ใช้ในการติดต่อ โดยค่า default แล้วจะอยู่ที่พอร์ตที่ 22 แต่ถ้าแก้ไขให้ใช้พอร์ตอื่นก็จะสามารถหลอกแฮ็กเกอร์ที่อ่อนประสบการณ์ที่ได้รหัสผ่านแต่ก็ยังไม่ทราบพอร์ตได้
ListenAddress 192.168.1.1	ระบุหมายเลข IP ให้กับฝั่งเซิร์ฟเวอร์ว่าให้รอรับการติดต่อทาง IP ใด จะเห็นผลชัดเจนยิ่งขึ้น ถ้าเครื่องเซิร์ฟเวอร์ดังกล่าวให้บริการไฟร์วอลล์ด้วย
HostKey /etc/ssh/ssh_host_key	HostKey /etc/ssh/ssh_host_key
ServerKeyBits 1024	ระบุจำนวนบิตที่จะใช้ในคีย์ของเซิร์ฟเวอร์ ซึ่ง default คือ 768 ให้ทำการแก้ไขเป็น 1024
LoginGraceTime 600	ระบุเวลาในหน่วยวินาทีหลังจากมีการร้องขอการติดต่อจากไคลเอ็นต์จนกระทั่งสิ้นสุดการติดต่อ โดยที่ยังไม่มีการ Login
KeyRegenerationInterval 3600	ระบุอายุของคีย์ หรือระยะเวลาเป็นวินาทีที่จะสร้างคีย์ใหม่ โดยค่า default อยู่ที่ 3600
PermitRootLogin no	ระบุว่าอนุญาตให้ accout ของรูตทำการ login เข้าระบบได้โดยตรงหรือไม่ ให้เปลี่ยนเป็น no
IgnoreRhosts yes	ระบุไม่อนุญาตให้ใช้ไฟล์ rhosts หรือ shosts สำหรับการ authenticate
StrictModes yes	ก่อนที่จะมีการอนุญาตให้ Login เข้ามาในระบบได้ sshd จะตรวจสอบ permission ในไดเรกทอรี home ของผู้ใช้งานแต่ละคน และไฟล์ rhosts ก่อน ดังนั้นออปชันนี้ให้เลือก yes เพราะบางครั้งผู้ใช้งานบางคนอาจจะเปลี่ยนแปลง permission ให้ผู้อื่นสามารถเขียนได้โดยไม่เจตนา
PrintMotd yes	ระบุให้ sshd แสดงข้อความที่อยู่ในไฟล์ /etc/motd เมื่อมีผู้ใช้งานระบบทำการ Login ซึ่งไฟล์ /etc/motd นั้นรู้จักกันในชื่อของ "message of the day"
RhostsAuthentication no	ระบุให้ sshd สามารถที่จะใช้ rhosts ในการ authenticate ได้ แนะนำว่าไม่ควรที่จะใช้เพราะว่า rhosts นั้นไม่มีความปลอดภัย
RhostsRSAAuthentication no	ระบุว่าจะให้สามารถใช้ rhosts ร่วมกับ RSA ในการ authenticate ได้หรือไม่
RSAAuthentication yes	ระบุว่าให้ใช้ RSA ในการ Authenticate โดย RSA จะใช้คีย์คู่ทั้ง public และ private ที่ถูกสร้างโดย ssh- keygen1utility ในกระบวนการ Authenticate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PasswordAuthentication yes	ระบุว่าให้ใช้รหัสผ่านในการทำ Authenticate
PermitEmptyPasswords no	ระบุว่าจะอนุญาตให้ Login โดยไม่ต้องใช้รหัสผ่านหรือไม่ ให้เลือก no เพื่อบังคับให้ป้อนรหัสผ่านทุกครั้ง
AllowUsers admin	ระบุว่าจะอนุญาตให้ใคร Login ได้บ้าง โดยค่า default แล้วจะอนุญาตให้ทุกคนสามารถ Login เข้าระบบได้ และสามารถเพิ่มชื่อผู้ใช้งานที่อนุญาตได้หลายคน โดยแยกแต่ละชื่อด้วยเครื่องหมายช่องว่างหรือ space
Subsystem sftp /usr/libexec/openssh/sftp-server	ระบุให้ใช้ sftp โดยค่า default จะอนุญาตอยู่แล้ว

ตารางที่ 8-4 : ความหมายของการคอนฟิกไฟล์ /etc/ssh/sshd_config

8.10 ไฟล์ /etc/services

ไฟล์ /etc/services จะกำหนดหน้าที่ต่างๆ ของระบบเน็ตเวิร์คบนเครื่อง โดยจะประกอบไปด้วยแอพพลิเคชันต่างๆ ที่มีบนโพรโตคอล tcp, udp หมายเลขพอร์ต / ชนิดของโพรโตคอล รวมทั้งชื่อที่กำหนดเป็น alias ของแอพพลิเคชันต่างๆ

service-name port/protocol aliases # optional comment

```
# Each line describes one service, and is of the form:
#
# service-name  port/protocol  [aliases ...]   [# comment]

tcpmux          1/tcp                           # TCP port service multiplexer
tcpmux          1/udp                           # TCP port service multiplexer
rje             5/tcp                           # Remote Job Entry
rje             5/udp                           # Remote Job Entry
echo            7/tcp
echo            7/udp
discard         9/tcp           sink null
discard         9/udp           sink null
systat          11/tcp          users
systat          11/udp          users
daytime         13/tcp
daytime         13/udp
qotd            17/tcp          quote
qotd            17/udp          quote
msp             18/tcp                          # message send protocol
msp             18/udp                          # message send protocol
chargen         19/tcp          ttytst source
chargen         19/udp          ttytst source
ftp-data        20/tcp
ftp-data        20/udp
```

รูปที่ 8-13 : ตัวอย่างรายละเอียดของไฟล์ /etc/services

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.11 ไฟล์ /etc/securetty

ไฟล์ /etc/securetty จะถูกเรียกผ่านโปรแกรมล็อกอิน ซึ่งปกติจะอยู่ที่ /bin/login โดยมีรูปแบบโครงสร้างภายในเป็นรายชื่อของ tty device name ที่อนุญาตให้รูตสามารถล็อกอินผ่านได้โดยตรง

```
[root@Isag32 etc]# cat securetty
vc/1
vc/2
vc/3
vc/4
vc/5
vc/6
vc/7
vc/8
vc/9
vc/10
vc/11
#tty0          <-------
#tty1          <-------
#tty2          <-------
tty3
tty4
tty5
tty6
tty7
tty8
tty9
tty10
tty11
[root@Isag32 etc]#
```

รูปที่ 8-14 : ตัวอย่างรายละเอียดของไฟล์ /etc/securetty

จากรูปที่ 8-14 จะปรากฏรายชื่อ tty ต่างๆ มากมาย นั่นหมายความว่า อนุญาตให้รูตสามารถล็อกอินผ่านคอนโซลเทอร์มินัลได้โดยตรง สำหรับ tty device name อื่น ๆ ที่ไม่ปรากฏอยู่ หรือ ถูกคอมเมนต์โดยเครื่องหมาย '#' จะหมายถึงไม่อนุญาตให้รูตล็อกอินผ่านเทอร์มินัลนั้นได้โดยตรง ซึ่งเราควรป้องกันไม่ให้รูตสามารถล็อกอินผ่านเทอร์มินัลจากระยะไกลโดยตรงได้ แต่จะสามารถใช้โปรแกรม su เพื่อเปลี่ยนจากการ ล็อกอินจากยูสเซอร์ทั่วไป มาเป็นรูตได้ เช่น โปรแกรม telnet ซึ่งมี tty device name เป็น ttyp* ดังนั้นเราจึงควรทำการคอมเมนต์ที่ ttyp* (ttyp0,ttyp1,ttyp2,...) ซึ่งสามารถแก้ไขได้โดยใช้โปรแกรมเท็กซ์เอดิเตอร์ทั่วไปได้

จะรู้ได้ยังไงว่า เราใช้ tty อะไรอยู่ ทำได้โดยการใช้คำสั่ง w ดู

```
[root@Isag32 root]# w
 11:55am  up 10 min,  3 users,  load average: 0.10, 0.34, 0.24
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU  WHAT
panny    tty1     -                 11:46am  8:23   0.17s  0.09s  bash
root     pts/0    -                 11:47am  7:30   0.00s   ?     -
root     pts/1    -                 11:47am  0.00s 23.83s  0.02s  w
[root@Isag32 root]# tty
/dev/pts/1
[root@Isag32 root]#
```

รูปที่ 8-15 : แสดงผลลัพธ์จากการใช้คำสั่ง w

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ได้นี้ จะเห็นว่าขณะนี้มีทั้งรูตและยูสเซอร์ panny ล็อกอินอยู่ และเรากำลังใช้ tty = pts1 อยู่ (ใช้คำสั่ง tty)

8.12 ไฟล์ lilo

LILO เป็นระบบการบูตเครื่องที่อนุญาตให้ผู้ใช้งานเลือกอิมเมจที่จะบูต อาจจะมีหลายๆระบบปฏิบัติการอยู่ในเครื่องเดียวกัน สามารถมีให้เลือกได้ถึง 16 อิมเมจ LILO สามารถใช้เป็นมาสเตอร์บูตได้ด้วย

มาสเตอร์บูตคือเซ็กเตอร์แรกของฮาร์ดดิสก์ซึ่งเป็นที่ที่ไบออสของคอมพิวเตอร์จะควานหาคำสั่งที่จะโหลดระบบปฏิบัติการใดขึ้นมา ก็ได้ขึ้นอยู่กับการตั้งค่าในไฟล์ lilo.conf

```
  UW PICO(tm) 4.2  File: /etc/lilo.conf    Modified

prompt
timeout=50
default=linux
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
restricted
password=xxxxx
message=/boot/message
linear

image=/boot/vmlinuz-2.4.18-14
        label=linux
        initrd=/boot/initrd-2.4.18-14.img
        read-only
        append="root=LABEL=/"
```

รูปที่ 8-16 : แสดงการตั้งค่าในไฟล์ lilo.conf

จากไฟล์ดังกล่าวนี้ได้ตั้งค่า timeout ไว้ที่ 50 วินาที ถ้าไม่มีคำสั่ง หรือ line command ภายในเวลา 10 วินาที LILO จะทำการ boot ค่าdefault ทันที ในไฟล์ /etc/lilo.conf คือ เป็นการจำกัดเวลาในการส่งคำสั่งรวมทั้งค่าพารามิเตอร์ นอกจากนี้ผู้ใช้งานยังต้องใส่ password เมื่อต้องการรันคำสั่งใดๆ ด้วย

8.13 ไฟล์ .bash_history

เป็นไฟล์ที่คอยเก็บคำสั่งที่เราใช้ตั้งแต่เราล็อกอินจนกระทั่งเราล็อกเอาต์จากระบบ โดยจำนวนที่จะเก็บนั้นจะเก็บตามที่ได้กำหนดไว้ในไฟล์ "/etc/profile" และจะเก็บจากคำสั่งปัจจุบันถอยลงไปเรื่อยๆ ซึ่ง .bash_history จะมีอยู่ในโฮมไดเรกทอรีของยูสเซอร์ทุกคน

ดังนั้นเป็นการเสี่ยงมากถ้ามีผู้บุกรุกเข้ามาดูที่ไฟล์ .bash_history ของยูสเซอร์ผู้นั้น จึงมีวิธีในการป้องกันความเสี่ยงดังกล่าวคือ ให้ยูสเซอร์ลบไฟล์ .bash_history ทิ้ง ก่อนที่จะออกจากระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิจารณาในไฟล์ /etc/profile

ขั้นแรก คือ

ตัวแปร HISTFILESIZE และ HISTSIZE จะเป็นตัวที่เก็บค่าจำนวนของคำสั่งในไฟล์ bash_history ของยูสเซอร์แต่ละคนด้วย เราควรจะกำหนดค่าให้มันต่ำไว้ เช่น 30

HISTFILESIZE = 30

HISTSIZE = 30

ซึ่งหมายความว่าไฟล์ bash_history ของแต่ละโฮมไดเรกทอรีสามารถเก็บคำสั่งได้อย่างมาก 30 คำสั่ง

ขั้นที่สอง คือ

ผู้ดูแลระบบควรจะเพิ่มไฟล์ bash_logout เข้าไปในโฮมไดเรกทอรีของแต่ละคนด้วย โดยมีเนื้อหาในไฟล์ว่า " rm -f $HOME/bash_history " ดังนั้นเมื่อใดที่ยูสเซอร์ออกจากระบบ ไฟล์ bash_history ก็จะถูกลบทิ้งด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 9
ภาษาเพิร์ล

9.1 ประวัติย่อ

เพิร์ล (Perl) ย่อมาจาก Practical Extraction and Report Language ถูกพัฒนาขึ้นในปี 1986 โดย นาย Larry Wall เป็นภาษาที่ใช้ทำ CGI ภาษาหนึ่ง ซึ่งมีรากฐานการพัฒนามาจากภาษา C ดังนั้น ผู้ที่เคยศึกษาภาษา C มาบ้างแล้ว สามารถศึกษาภาษาเพิร์ลได้โดยไม่ยาก

เพิร์ล คือ ภาษาที่ใช้ในการเขียนโปรแกรม สามารถใช้กับงานต่างๆ ได้หลายรูปแบบ สามารถทำงานได้ดีกับข้อมูลเท็กซ์ไฟล์ ประมวลผลข้อมูลจากเท็กซ์ไฟล์และพิมพ์ผลลัพธ์ การประมวลผลนั้นออกมา ผู้ออกแบบภาษาเพิร์ล กล่าวไว้ว่า

เพิร์ลเป็นภาษาที่ง่ายกับการเรียนรู้และใช้งาน เพิร์ลมีข้อดีหลายๆ อย่างของภาษา C, sed, awk, และ sh (shell script) ดังนั้นสำหรับผู้ที่คุ้นเคยอยู่กับภาษาเหล่านั้นอยู่แล้ว ก็ยิ่งทำความเข้าใจกับเพิร์ลได้ง่ายมากขึ้น รูปแบบของภาษาเพิร์ลจะใกล้เคียงกับภาษา C ค่อนข้างมาก แต่เพิร์ลไม่จำกัดขนาดของข้อมูลเหมือนๆ กับยูทิลิตี้สำหรับยูนิกซ์อื่นๆ สามารถใช้ขนาดของข้อมูลได้เต็มที่เท่าที่หน่วยความจำของเครื่องมี เพิร์ลสามารถอ่านไฟล์ทั้งไฟล์ไว้ในตัวแปรสตริงตัวเดียวได้ สำหรับการทำงานแบบ Recursion ก็ไม่จำกัดจำนวนรอบที่จะวนทำงาน

ภาษาเพิร์ลมีความปลอดภัยมากกว่าภาษา C เพราะว่าเป็นภาษาโปรแกรมในระดับสูง เปิดโอกาสให้เขียนโปรแกรมติดต่อกับตัวระบบได้ยากกว่า C จึงมีความปลอดภัยมากกว่า โดยเฉพาะกับระบบที่มีผู้ใช้งานหลายคน หรือระบบเปิด เช่น เวปเซิร์ฟเวอร์ (Web server) ที่มีผู้ใช้งานระบบมาหน้าหลายตา

โปรแกรมที่เขียนด้วยภาษาเพิร์ลจะเรียกว่า เพิร์ลสคริปต์ (Perl Script) ส่วนคำว่า เพิร์ลโปรแกรม (Perl program) จะหมายถึงโปรแกรมเพิร์ลที่เป็นตัวแปลภาษาเพิร์ล ใช้ในการเรียกใช้งานโปรแกรมภาษาเพิร์ล

ตัวแปลภาษาเพิร์ล เป็นตัวแปลภาษาแบบอินเทอร์พรีเตอร์ (interpreter) ไม่ใช่คอมไพเลอร์ (compiler) ดังนั้นในขณะที่โปรแกรมภาษาเพิร์ลทำงาน จะใช้กำลังของซีพียูมากกว่าโปรแกรมที่เป็นลักษณะของคอมไพล์ เช่น โปรแกรมภาษา C แต่มองในอีกมุมหนึ่ง การเขียนโปรแกรมด้วยภาษาเพิร์ลแทน C ก็ช่วยประหยัดเวลาในการทำงานด้วย

9.2 การติดตั้งเพิร์ลบนยูนิกซ์หรือลีนุกซ์

เพิร์ล เป็นโปรแกรมที่ถูกติดตั้งมาแล้วกับยูนิกซ์ทุกเครื่อง ไม่ว่าจะเป็นเครื่อง Sun, HP, หรือแม้กระทั่งลีนุกซ์จะแตกต่างกันที่เวอร์ชันและสถานที่ติดตั้งบนไดเรกทอรีของเครื่อง เพิร์ลเป็น โปรแกรมที่คงที่แล้ว คาดว่าคงจะยังไม่ออกเวอร์ชันใหม่ง่ายๆ ดังนั้นเราจึงไม่จำเป็นต้องติดตั้งเพิร์ลบนเครื่องยูนิกซ์อีก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราจะต้องหาเพิร์ลให้เจอว่า ถูกติดตั้งไว้ณ.ที่ใด เพื่อจะได้ใช้เป็น Heading อ้างอิงภายในสคริปต์เพิร์ล โดยค้นหาดังนี้

<u>รูปแบบคำสั่ง</u>

$ whereis สิ่งที่ต้องการค้นหา

<u>ตัวอย่าง</u>

$whereis perl

<u>ผลลัพธ์ที่ได้</u>

ส่วนใหญ่นิยมใช้กันที่

/usr/bin/perl หรือ

/usr/local/bin/perl/Perl

ในตอนนี้เราค้นพบเพิร์ลในเครื่องของเราแล้ว ต่อไปเราจะมาทดสอบเขียน สคริปต์เพิร์ล กัน

9.3 เริ่มต้นการใช้เพิร์ล

และนี่คือโปรแกรม ตามมาตรฐาน โลก ที่เราจะใช้ในการ เริ่มต้นเรียนรู้ภาษาเพิร์ล

```
#!/usr/local/bin/perl
#
# This is my first program in perl.
#
print "Content-type:text/html\n\n";
print 'Hello world.'; # Print a standard message..
```

<u>แต่ละส่วนของโปรแกรมมีความหมายดังต่อไปนี้</u>

บรรทัดแรก โปรแกรมภาษาเพิร์ลทุกโปรแกรมจะ เริ่มด้วยบรรทัดลักษณะคล้ายๆ อย่างนี้ (คล้ายๆ หรืออาจจะเหมือน แล้วแต่เซิร์ฟเวอร์) บรรทัดนี้จะเป็นตัวบอกกับเซิร์ฟเวอร์ ว่าจะต้องทำอะไรกับ ไฟล์โปรแกรมไฟล์นี้ เมื่อมีการสั่งให้ทำงานเกิดขึ้น (ในที่นี้ก็จะสั่งให้ เรียกโปรแกรมเพิร์ลมาแปลชุดคำสั่งภาษาเพิร์ลโปรแกรมนี้) ในส่วนของบรรทัดนี้ จะแตกต่างกันออกไป ขึ้นอยู่กับระบบที่คุณใช้งาน บางเซิร์ฟเวอร์อาจจะเป็น #!/usr/bin/perl หรืออย่างอื่นก็ได้ ขึ้นอยู่กับ ผู้ดูแลระบบจะกำหนด

คำอธิบายโปรแกรม (Comment)

คำอธิบายโปรแกรม สามารถแทรกอยู่ในส่วนไหน ของโปรแกรมก็ได้ โดยจะต้องเริ่มต้น ด้วยตัวสัญลักษณ์ # ทุกสิ่งที่อยู่หลัง # ไปจนจบบรรทัด จะถูกมองข้าม ไปไม่นำมาทำงาน (ยกเว้นที่อยู่ที่บรรทัดแรก) ถ้าต้องการจะเขียน คำอธิบายโปรแกรมหลายๆ บรรทัดก็ทำได้โดยการใส่ # นำหน้าไว้ทุกบรรทัด (จะไม่เหมือนกับ C ที่ใช้ /* และ */ ข้ามบรรทัดได้)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง (Statement)

```
print "Content-type:text/html\n\n";
print 'Hello world.';
```

ส่วนนี้จะเป็นส่วนของ คำสั่งในภาษาเพิร์ลเป็นการสั่งให้โปรแกรม ส่งข้อความว่า "Hello world." ออกมา (ทาง stdout ในที่นี้ ขอทำความตกลงว่า คือ browser) คำสั่งในภาษาเพิร์ลจะต้องจบท้ายคำสั่ง ด้วยเครื่องหมาย ; (semicolon) เราจะใช้เครื่องหมาย ; เป็นตัวแยกแต่ละคำสั่งออกจากกัน (ส่วนนี้จะเหมือนกับในภาษา C) ส่วนบรรทัดบนจะบอกเซิร์ฟเวอร์ว่าผลลัพธ์จะส่งออกมา เป็นข้อมูลชนิด text/html (บรรทัดนี้กับบางเซิร์ฟเวอร์ เช่น hypermart ถ้าไม่ระบุ อาจจะได้ผลลัพธ์เป็น 501 Internal server error ได้)

9.4 ตัวแปรในภาษาเพิร์ล

ตัวแปรในภาษาเพิร์ล เป็นตัวแปรแบบ ไม่ได้กำหนดชนิด สามารถใช้ได้ทั้ง ข้อความ หรือ ตัวเลข ไม่ต้องมีการกำหนดชนิด ไม่ต้องมี การประกาศชื่อตัวแปรก่อน สามารถเรียกใช้งาน ในส่วนไหนของโปรแกรมก็ได้ (คิดว่าสะดวกกว่า C ไหม?) ในภาษาเพิร์ล มีตัวแปรอยู่ 3 แบบ คือ

9.4.1 สเกลลาร์

- ชื่อตัวแปรจะต้องขึ้นต้นด้วย $ (เช่น **$varname**)
- ใช้ได้ทั้งข้อมูลตัวอักษร ข้อความและตัวเลข

9.4.2 อาร์เรย์

- ชื่อตัวแปรจะต้องขึ้นต้นด้วย @ (เช่น **@arrayname**)
- อ้างอิงถึงข้อมูล แต่ละตัวในอาร์เรย์ได้โดยใช้รูปแบบ : **$arrayname[n]**
- ใช้ได้ทั้งข้อมูล ตัวอักษร ข้อความและตัวเลข ภายในอาร์เรย์เดียวกันสามารถใช้ได้ กับทั้ง ตัวอักษร ข้อความ และตัวเลข

9.4.3 Associative Array (หรือที่เรียกว่า Hash)

- ชื่อตัวแปรจะต้องขึ้นต้นด้วย % (เช่น **%hashname**)
- อ้างอิงถึงข้อมูล แต่ละตัวใน Hash ได้โดยใช้รูปแบบ : **$arrayname{'key'}**
- เก็บข้อมูลได้ทั้ง ตัวอักษร ข้อความและตัวเลข โดยประกอบด้วย **key** และ **value** เช่น

  "Viva Friday","Cartoon"

  "Internet Magazine","Computer Magazine"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.5 โอเปอเรเตอร์และการกำหนดค่าให้กับตัวแปรชนิดสเกลลาร์

การกำหนดค่า ให้กับตัวแปรชนิดต่างๆ ในภาษาเพิร์ล (รูปแบบจะเหมือนๆ กับในภาษา C มาก)

	คำอธิบาย
$a = $b;	นำข้อมูล ในตัวแปร $b ไปเก็บไว้ ในตัวแปร $a
$a += $b;	นำข้อมูล ในตัวแปร $b ไปบวกเพิ่ม กับข้อมูล ในตัวแปร $a แล้วเก็บผลลัพธ์ ไว้ในตัวแปร $a
$a -= $b;	นำข้อมูล ในตัวแปร $b ไปลบออก จากข้อมูล ในตัวแปร $a แล้วเก็บผลลัพธ์ ไว้ในตัวแปร $a
$a .= $b;	นำข้อมูล ในตัวแปร $b ไปเพิ่ม ต่อท้ายข้อมูล ในตัวแปร $a แล้วเก็บไว้ในตัวแปร $a

ตารางที่ 9-1 : การกำหนดค่าและ การดำเนินการกับ ตัวแปรชนิดสเกลลาร์

<u>หมายเหตุ : ในภาษาเพิร์ลการกำหนดค่า ให้กับตัวแปรโดยใช้ $a = $b จะเป็นการนำสำเนา ของข้อมูลในตัวแปร $b มากำหนด ให้กับตัวแปร $a เพราะฉะนั้น เมื่อมีการเปลี่ยนแปลง ข้อมูลในตัวแปร $b จะไม่ส่งผล กับข้อมูลในตัวแปร $a</u>

ดูตัวอย่าง การดำเนินการ กับตัวแปรชนิดสเกลลาร์กับข้อมูล ชนิดค่าตัวเลข

	คำอธิบาย
$a = 1 + 2;	นำ 1 บวกกับ 2 แล้วเก็บค่า ที่ได้ไว้ ในตัวแปร $a
$a = 3 - 4;	นำ 3 ลบออกด้วย 4 แล้วเก็บค่า ที่ได้ไว้ ในตัวแปร $a
$a = 5 * 6;	นำ 5 มาคูณ ด้วย 6 แล้วเก็บค่า ที่ได้ไว้ ในตัวแปร $a
$a = 7 / 8;	นำ 7 มาหารด้วย 8 แล้วนำค่า ที่ได้เก็บไว้ ในตัวแปร $a
$a = 9 ** 10;	นำ 9 มายกกำลังด้วย 10 แล้วนำค่า ที่ได้เก็บไว้ ในตัวแปร $a
$a = 5 % 2;	นำ 5 มาหารด้วย 2 แล้วนำเศษที่เหลือ จากผลหารเก็บไว้ ในตัวแปร $a
++$a;	นำค่าในตัวแปร $a มาเพิ่มค่าขึ้นอีก 1 และส่งค่า ที่เพิ่มค่า แล้วนั้นกลับ
$a++;	นำค่าในตัวแปร $a ส่งกลับไปก่อน แล้วจึงค่อยนำค่า ในตัวแปร $a มาเพิ่มค่าอีก 1
-$a;	นำค่าในตัวแปร $a มาลบค่าลงอีก 1 และส่งค่า ที่ลดค่าแล้ว นั้นกลับ
$a-;	นำค่าในตัวแปร $a ส่งกลับไปก่อน แล้วจึงค่อยนำค่า ในตัวแปร $a มาลบค่าลงอีก 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 9-2 : การดำเนินการ กับตัวแปรชนิดสเกลลาร์ กับข้อมูล ชนิดค่าตัวเลข

ดูตัวอย่าง การดำเนินการ กับตัวแปรชนิดสเกลลาร์กับข้อมูลชนิด ข้อความ

	คำอธิบาย
$a = $b . $c;	นำข้อความ ในตัวแปร $b รวมกับข้อความ ในตัวแปร $c แล้วเก็บ ไว้ในตัวแปร $a
$a = $b x $c;	นำข้อความ ในตัวแปร $b มาทำซ้ำ เป็นจำนวน $c รอบ แล้วเก็บไว้ ในตัวแปร $a

ตารางที่ 9-3 : การดำเนินการ กับตัวแปรชนิดสเกลลาร์กับข้อมูลชนิด ข้อความ

9.6 โอเปอเรเตอร์และการกำหนดค่าให้กับตัวแปรชนิดอาร์เรย์

ในการเขียนโปรแกรมภาษาเพิร์ล ถึงแม้จะใช้วิธีการ (expression) ที่เหมือนกัน แต่ใช้กับข้อมูลที่ต่างกัน ก็จะผลลัพธ์ที่ต่างกัน ดูตัวอย่างต่อไปนี้

```
@book=("cartoon","computer magazine");
@morebook=("travel magazine", @cartoon, "internet magazine");
@morebook=("travel magazine", "cartoon", "computer magazine", "internet magazine");
```

ในบรรทัดที่สอง ของชุดคำสั่ง จะนำตัวแปรอาร์เรย์ที่ชื่อว่า @cartoon มาแตกย่อยออกมา เพราะฉะนั้น ผลของชุดคำสั่งในบรรทัดที่สองจึงเหมือนกับผลของชุดคำสั่งในบรรทัดที่สาม

สำหรับการ เพิ่มข้อมูลเข้าอาร์เรย์วิธีการที่ดีที่สุด แนะนำให้ใช้ function **push** ตัวอย่างการใช้ function **push**

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

จะนำข้อมูล *"game magazine"* ไปใส่ต่อท้าย ตัวแปรอาร์เรย์ที่ชื่อว่า @morebook ถ้าหากต้องการ เพิ่มข้อมูลมากกว่า 2 ชุดเข้าสู่ ตัวแปรแบบอาร์เรย์สามารถทำได้ โดยใช้รูปแบบใด รูปแบบหนึ่ง ตามตัวอย่างต่อไปนี้

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

หลังจากที่ตัว function **push** ทำงานข้อมูล ก็จะถูกเพิ่มเข้าไปในอาร์เรย์ที่ต้องการ ส่วนตัว function **push** เองจะส่งค่ากลับ มาเป็นขนาดของ ตัวแปรอาร์เรย์ที่ถูกเพิ่ม ข้อมูลเข้าไป

หากต้องการที่จะนำข้อมูล ตัวท้ายสุดของตัวแปรอาร์เรย์ออกจากตัวแปรอาร์เรย์ สามารถทำได้ โดยใช้ function **pop** จากตัวแปรอาร์เรย์ตัวอย่าง ที่ผ่านมาเมื่อใช้ function **pop** ตัว function **pop** จะส่งค่ากลับ "*newspaper*" มา และตัวแปรอาร์เรย์ @morebook จะมีสมาชิกลดลง ดูตัวอย่างการใช้ function **pop**

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราสามารถกำหนด ค่าตัวแปรอาร์เรย์ให้กับตัวแปรแบบสเกลลาร์ได้ด้วยเช่นกัน แต่ต้องระวัง ในการระบุค่า ที่จะกำหนดให้ ให้สังเกต บรรทัดคำสั่ง สองคำสั่งต่อไปนี้

$n = @morebook

จะนำขนาด ของตัวแปร @morebook ไปกำหนดค่า ให้กับตัวแปร $n แต่

$n = "@morebook"

จะเปลี่ยนข้อมูล สมาชิกในตัวแปรอาร์เรย์เป็น ข้อมูลแบบข้อความ โดยแบ่งสมาชิกแต่ละตัวในอาร์เรย์ออกจากกัน โดยใช้ ช่องว่าง (space)

9.7 การดำเนินการ กับข้อความ

คุณสมบัติเด่น อีกอย่างหนึ่งของ ภาษาเพิร์ล คือการดำเนินการกับข้อความและตัวแปรที่เก็บข้อมูล ชนิดข้อความ ภาษาเพิร์ลสามารถทำได้ค่อนข้างมีประสิทธิภาพ หัวใจหลักของการดำเนินการกับข้อมูลข้อความ นี้ก็คือ *regular expression* ซึ่งมีใช้กันใน utility หลายๆ โปรแกรมในระบบยูนิกซ์

9.7.1 Regular expression

regular expression จะถูกบรรจุ อยู่ในเครื่องหมาย / (เครื่องหมายทับ ที่ใช้ในบ้านเลขที่) และจะใช้ร่วม กับตัวกระทำ =~ ตัวอย่าง expression ต่อไปนี้ จะมีผลเป็นจริง ก็ต่อเมื่อมีข้อความ *the* ปรากฏอยู่ในตัวแปร $test

$test =~ /the/

การตรวจสอบของ regular expression เป็นแบบ case sensitive (ตัวอักษรใหญ่ ตัวอักษรเล็ก ต่างกัน) เพราะฉะนั้นถ้าหากว่า

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

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

$test !~ /the/

ผลก็คือจริง เพราะว่าไม่มี *the* อยู่ในตัวแปร $test

9.7.2 ตัวแปรพิเศษ $_

เราสามารถกำหนด เงื่อนไขโดยใช้รูปแบบ ต่อไปนี้ได้

```
if ($test =~ /the/)
{
   print "Yes sir..\n";
}
```

ถ้าหากกำหนดตัวแปร $test ดังตัวอย่างต่อไปนี้ ชุดคำสั่งข้างบน ก็จะพิมพ์ข้อความออกมา

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

$test = "The mall is big";

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ถ้าหากว่าเรากำหนด ค่าของตัวแปร test ให้กับตัวแปรพิเศษ $_ เราก็ไม่จำเป็น ต้องใช้ตัวกระทำ =~ หรือ !~ และเงื่อนไขข้างบน ก็จะเขียนได้ง่ายลง เป็นดังนี้

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

ตัวแปรพิเศษ $_ เป็นตัวแปร default ที่ใช้ได้กับ ตัวกระทำ (operation) หลายๆ อย่างในภาษาเพิร์ลและใช้ประโยชน์ ได้มาก

9.8 ตัวดำเนินการ (Operator) ในภาษาเพิร์ล

9.8.1 ตัวดำเนินการ ทางคณิตศาสตร์ (Arithmetic Operators):

โอเปอเรเตอร์	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 9-4 : แสดงตัวดำเนินการทางคณิตศาสตร์

9.8.2 ตัวดำเนินการ ทางตรรกะ(Logical Operators):

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

ตารางที่ 9-5 : แสดงตัวดำเนินการทางตรรกะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.8.3 ตัวดำเนินการเปรียบเทียบ (Relational Operators):

กับค่าตัวเลข	กับข้อความ	ตัวอย่าง	ความหมาย
==	eq	$str eq "Word"	เท่ากันกับ
!=	ne	$str ne "Word"	ไม่เท่ากันกับ
>	gt	$var > 10	มากกว่า
>=	ge	$var >= 10	มากกว่า หรือเท่ากันกับ
<	lt	$var < 10	น้อยกว่า
<=	le	$var <= 10	น้อยกว่า หรือเท่ากันกับ

ตารางที่ 9-6 : แสดงตัวดำเนินการเปรียบเทียบ

9.9 การตรวจสอบเงื่อนไข

การตรวจสอบเงื่อนไข ในภาษาเพิร์ลสามารถทำได้โดยการใช้ชุดคำสั่งตรวจสอบเงื่อนไขแบบ if/then/else โดยใช้รูปแบบดังต่อไปนี้

9.9.1 โครงสร้างของการตรวจเงื่อนไขแบบ if .. elsif .. else

if (Expression) { Block }

if (Expression) { Block } else { Block2 }

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

<u>ตัวอย่างการตรวจเงื่อนไข</u>

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

จากตัวอย่างถ้าหากว่า ไม่มีข้อความในตัวแปร $data การตรวจสอบเงื่อนไข จะให้ค่าเป็น false ซึ่งจะแสดงผลลัพธ์ว่า ข้อความว่างเปล่าขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อควรจำ : **elsif** ในภาษาเพิร์ลจะไม่มีตัวอักษร "**e**" ไม่ใช่การพิมพ์ผิด หรือพิมพ์ตกแต่อย่างใด และ Block การทำงาน ในการตรวจสอบเงื่อนไขจะต้องอยู่ภายใน **เครื่องหมายวงเล็บปีกกาเสมอ** แม้จะมีคำสั่งที่ต้องทำ เพียงคำสั่งเดียวก็ตาม ในจุดนี้จะต่างจากภาษา C

9.10 การทำงานแบบวนรอบ และเงื่อนไขการทำงาน แบบวนรอบ

ภาษาเพิร์ลสนับสนุนการทำงาน โดยใช้โครงสร้างการควบคุม การทำงานแบบวนรอบ มีการวนรอบรูปแบบต่างๆ ให้เลือกใช้ตามความเหมาะสม ของงานมากมาย โครงสร้างของการวนรอบ ในภาษาเพิร์ลไม่ยุ่งยากมากนัก และมีลักษณะ คล้ายกับในภาษา C และ Pascal ค่อนข้างมาก โดยจะมีชุดคำสั่งสำหรับทำการวนรอบ อยู่สองลักษณะคือ

while, until วนรอบทำงาน จนกว่าจะเป็นไปตามเงื่อนไข

for, foreach กำหนดเงื่อนไข และจำนวนครั้งของการวนรอบ

9.10.1 while และ until มีรูปแบบการใช้งาน ดังต่อไปนี้

while loop

while (Expression) { Block }

จะตรวจสอบเงื่อนไข (expression) ก่อนถ้าหากว่าเงื่อนไขเป็น **จริง** ก็จะกระทำชุดคำสั่งใน {Block} และกลับมา ตรวจสอบเงื่อนไขอีกครั้ง ทำอย่างนี้ไปเรื่อยๆ จนกว่าเงื่อนไขจะเป็น **เท็จ** จึงจะจบรอบการทำงาน

until loop

until (Expression) { Block }

จะตรวจสอบเงื่อนไข (expression) ก่อนถ้าหากว่าเงื่อนไขเป็น **เท็จ** ก็จะกระทำชุดคำสั่งใน {Block} และกลับมา ตรวจสอบเงื่อนไขอีกครั้ง ทำอย่างนี้ไปเรื่อยๆ จนกว่าเงื่อนไขจะเป็น จริง จึงจะจบรอบการทำงาน

do/until loop

do { Block } until (Expression)

จะกระทำชุดคำสั่งใน { Block } ก่อน หลังจากนั้นจึงจะมาทำ การตรวจสอบเงื่อนไข (expression) ถ้าหากว่าเงื่อนไขเป็น **จริง** ก็จะจบการวนรอบ หากว่าเงื่อนไขยังเป็น **เท็จ** อยู่ก็จะวนกลับ ไปทำงานใน {Block} อีกครั้งและ มาทำการตรวจสอบเงื่อนไขอีก ทำอย่างนี้ไปเรื่อยๆ จนกว่าเงื่อนไขจะเป็น **จริง**

do/while loop

do { Block } while (Expression)

จะกระทำชุดคำสั่งใน { Block } ก่อน หลังจากนั้นจึงจะมาทำ การตรวจสอบเงื่อนไข (expression) ถ้าหากว่าเงื่อนไขเป็น **เท็จ** ก็จะจบการวนรอบ หากว่าเงื่อนไขยังเป็น **จริง** อยู่ก็จะวนกลับไปทำงานใน {Block} อีกครั้งและ มาทำการตรวจสอบเงื่อนไขอีก ทำอย่างนี้ไปเรื่อยๆ จนกว่าเงื่อนไขจะเป็น **เท็จ**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.10.2 for และ foreach มีรูปแบบ การใช้งานดังต่อไปนี้

for loop

for (Expression1; Expression2; Expression3) { Block }

Expression1 เป็นค่าเริ่มต้น ของการวนรอบ

Expression2 ใช้สำหรับตรวจสอบ ว่าจะวนรอบต่อ หรือว่าหยุดการวนรอบ

Expression3 ใช้สำหรับปรับปรุง (update) ตัวแปรที่ใช้ตรวจสอบการวนรอบ

อันดับแรกจะเริ่มกระทำ Expression1 ก่อน เพื่อกำหนดค่าเริ่มต้น ของการวนรอบ

หลังจากนั้นจะตรวจสอบ Expression2 หากว่าเงื่อนไข เป็นจริงจะกระทำชุดคำสั่งใน { Block }

เมื่อเสร็จสิ้นการทำงานชุดคำสั่งใน { Block } แล้วก็จะมากระทำ Expression3

หลังจากนั้นจะ วนไปขั้นตอนตรวจสอบ Expression2 อีกครั้ง ทำอย่างนี้ไปจนกระทั่ง Expression2 เป็นเท็จ

foreach loop

foreach VARIABLE (ARRAY) { Block }

ARRAY เป็นตัวกำหนดการวนรอบ ว่าจะวนกี่รอบขึ้นอยู่กับ จำนวนข้อมูลในตัวแปรอาร์เรย์

VARIABLE เป็นตัวแปร ที่จะรับข้อมูลจากอาร์เรย์ตัวปัจจุบันที่ดัชนี ตรงกับรอบที่วนทำงานอยู่

ถ้าหากว่าไม่กำหนด VARIABLE จะถือว่าเป็นการเรียกใช้ตัวแปร $_

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

การวนรอบเพื่อทำงาน ตามชุดข้อมูลในอาร์เรย์หรือข้อมูลโครงสร้างแบบลิสต์อื่นๆ

9.11 การจัดการแฟ้มข้อมูล

ต่อไปนี้คือตัวอย่าง ของโปรแกรมภาษาเพิร์ลที่ใช้สำหรับ เปิดแฟ้มข้อมูลมาแสดงผล แบบง่ายๆ ลักษณะการทำงาน จะคล้ายๆกับคำสั่ง **cat** ในระบบ Unix

```
#!/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
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราใช้ function **open** สำหรับการเปิดแฟ้มข้อมูล เพื่ออ่านข้อมูล ใน function **open** ตัว parameter ตัวแรกคือ filehandle ซึ่งใช้ในการอ้างอิง ในการติดต่อกับแฟ้มข้อมูลในภาษาเพิร์ล ส่วน parameter ตัวที่สองคือ expression ที่บอกถึง ชื่อแฟ้มข้อมูลที่ต้องการติดต่อ ถ้าหากระบุชื่อแฟ้มข้อมูล ไว้ภายใต้เครื่องหมาย quote '...' จะหมายถึงการอ้างอิง แฟ้มข้อมูลชื่อนั้นๆ โดยตรง ไม่สามารถใช้รูปแบบ การอ้างอิงที่ใช้ใน shell ได้ (เช่น ~ เพื่ออ้างอิงถึงโฮมไดเรกทอรีของยูสเซอร์) ดังนั้นหากระบุ ชื่อแฟ้มข้อมูลว่า '~/data/mydata' จะทำให้ เกิดข้อผิดพลาดขึ้น ไม่สามารถประมวลผลได้ หากว่าต้องการที่จะใช้ รูปแบบการอ้างอิง ที่ใช้ใน shell จะต้องระบุ ชื่อแฟ้มข้อมูลไว้ภายใน เครื่องหมายวงเล็บ มากกว่าและน้อยกว่า <..> เช่น <~/data/mydata> แทน

ส่วน function **close** จะใช้สำหรับบอก โปรแกรมภาษาเพิร์ลว่าต้องการที่จะปิดการติดต่อกับแฟ้มข้อมูล

หากต้องการ ที่จะติดต่อกับ standard input (เช่น keyboard) และ standard output (เช่น จอภาพ) สามารถทำได้โดย ใช้คำสั่งดังต่อไปนี้

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

จากโปรแกรมตัวอย่างข้างต้น ข้อมูลจะถูกเก็บไว้ในแฟ้มข้อมูล แฟ้มข้อมูลนี้จะถูกอ้างอิงโดย handle ที่ชื่อว่า INFO ถ้าหากว่า ต้องการอ่านข้อมูลจาก INFO ในภาษาเพิร์ลสามารถทำได้ โดยใช้เครื่องหมาย วงเล็บน้อยกว่า มากกว่า <...>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 10
แนวคิดในการออกแบบและการทดสอบ

10.1 สรุปปัญหาและทางแก้ไข

ต่อไปนี้จะเป็นการสรุปปัญหาที่เกิดขึ้นจนเป็นช่องโหว่หรือจุดอ่อน ที่ผู้บุกรุกสามารถเจาะผ่านเข้าระบบได้

10.1.1 เรื่องของ Unowned files

Unowned file เกิดจาก 2 วิธี

1. การติดตั้ง หรือ แตกไฟล์โปรแกรม เช่น .tar.gz บางครั้ง ไฟล์นั้นเก็บสิทธิการใช้งานของ source มาด้วย ทำให้เมื่อแตกไฟล์มาแล้ว ไฟล์ภายในนั้นจะกลายเป็น unowned file

2. ผู้ดูแลระบบทำการลบยูสเซอร์คนนั้นออกไปจากระบบ แต่ไม่ได้ทำการตามไปลบไฟล์ที่เกิดจากการสร้างของยูสเซอร์คนนั้น ทำให้ มีไฟล์ค้างอยู่ในระบบโดยไม่สามารถระบุเจ้าของได้

สิ่งที่เกิดขึ้นคือ ตัวไฟล์จะมี owner id เก็บอยู่ หากผู้ดูแลระบบทำการ adduser คนใหม่เข้าไป ซึ่งมี uid เท่ากัน ทำให้ไฟล์เก่านั้นจะกลายเป็นของยูสเซอร์คนใหม่ทันที

วิธีการป้องกัน คือ คอยตรวจสอบ โดยใช้คำสั่ง find / -nouser -o –nogroup สำหรับไฟล์ที่ทำการติดตั้งมาควรจะกำหนดค่าให้เป็นของเราเอง ส่วนไฟล์ที่ค้างระบบที่เกิดจากการลบยูสเซอร์ ไปนั้น ควรลบออกเสีย

10.1.2 เรื่องของ Symbolic link

ปัญหาด้านความปลอดภัยอีกประการหนึ่งมักเกิดมาจากการที่บางโปรแกรมมักหลับหูหลับตาเปลี่ยนตาม symbolic link เข้าไปที่ไฟล์อื่น ไฟล์ที่เก็บ symbolic link ไว้ (ไฟล์นี้สร้างขึ้นมาได้ด้วยคำสั่ง ln) หมายถึงไฟล์ที่เก็บพอยเตอร์ที่ชี้ไปยังไฟล์อื่นนั่นเอง ปัญหามักเกิดจากการสร้างไฟล์ชั่วคราวของโปรแกรมหากไม่ตรวจดูให้รอบคอบก่อนว่าไฟล์ชั่วคราวดังกล่าวนั้นเป็นไฟล์จริงหรือเป็นเพียงไฟล์ที่เก็บ symbolic link นั้นเป็นไฟล์จริงหรือเป็นเพียงไฟล์ที่เก็บ symbolic link นั้น หากแฮ็กเกอร์สร้าง symbolic link ไว้ในไฟล์ชื่อเดียวกันรอไว้ก่อน โดยให้ชี้ไปยังไฟล์อื่นที่อยู่ในระบบไฟล์ เช่น /etc/passwd อาจทำให้เปลี่ยนความเป็นเจ้าของ หรือ อาจทำให้ถูกเขียนทับลงไปได้

วิธีการป้องกัน นักพัฒนาโปรแกรมควรเช็คดูด้วยว่า ไฟล์นี้มีอยู่แล้วหรือไม่ ก่อนที่จะพยายามสร้างไฟล์ใหม่ขึ้นมา

10.1.3 เรื่องของ SUID/SGID

โปรแกรมที่มีการกำหนด set-uid หรือ set-gid ไว้ ก่อให้เกิดความไม่ปลอดภัยต่อระบบ เราจะทำการยกตัวอย่าง โปรแกรม shell.c ซึ่งมีการกำหนด s bit ที่ยูสเซอร์และกลุ่มไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <stdio.h>
#include <string.h>
#include <sys/types.h>
#include <unistd.h>
int setuid(uid_t uid);
int setgid(gid_t gid);
  int main()
  {
            setuid(0);
            setgid(0);
            system("/bin/sh");
  }
```

โปรแกรมข้างต้นจะทำหน้าที่ในการเปลี่ยนสิทธิจากยูสเซอร์ธรรมดาที่สั่งรันโปรแกรม ให้มีสิทธิกลายเป็นรูต คือ uid=0 , gid=0 ทันที โปรแกรมข้างต้นนี้เป็นตัวอย่างโปรแกรมที่ผู้บุกรุกฝังโปรแกรมที่ทำการคอมไพล์แล้ว เอาไว้ในระบบภายหลังจากที่สามารถเจาะเข้ามาในระบบได้แล้ว และเคยได้รับสิทธิเป็นรูต แล้วจึงเขียนโปรแกรมนี้ขึ้นมา พร้อมทั้ง set-uid ,set-gid ไว้ เพื่อคราวหน้าจะได้เข้ามาในฐานะยูสเซอร์ธรรมดา แล้วสามารถเปลี่ยนเป็นรูตได้ทันที

มาตรการป้องกันการโจมตีผ่านทาง SUID/SGID ก็คือ ให้ทำการหาไฟล์ในระบบที่มีการset SUID/SGID ไว้แล้วยกเลิกบิต SUID/SGID ออกจากไฟล์ต่างๆ ให้มากที่สุดเท่าที่จะทำได้ แต่อาจจะมีไฟล์จำนวนเล็กน้อยบนลีนุกซ์ทุกๆ ยี่ห้อซึ่งจำเป็นจะต้องเซตบิต SUID/SGID ให้เป็นของรูต เพื่อให้ระบบทำงานได้อย่างถูกต้อง

10.1.4 เรื่องของ World Writable

ไฟล์ที่เปิดให้ทุกคนสามารถเขียนได้ การคอนฟิกระบบที่ไม่เหมาะสมที่พบบ่อยอีกอันหนึ่ง ก็คือการกำหนดสิทธิบนไฟล์สำคัญบางไฟล์ให้ทุกคนสามารถแก้ไขไฟล์นั้นๆ ได้ ซึ่งคล้ายๆ กับไฟล์ที่ทำการเซตบิต SUID ไว้ตรงที่ว่าสิทธิดังกล่าวถูกเซตขึ้นมาด้วยจุดประสงค์ในเรื่องของสะดวกเป็นหลัก แต่ผลข้างเคียงที่ได้รับจากความสะดวกก็คือ มันอาจนำไปสู่ช่องโหว่ด้านความปลอดภัยที่ไม่เป็นผลดีนักต่อระบบ ไฟล์ทั่วไปที่มักถูกเซตให้ทุกคนเขียนได้จะประกอบด้วยไฟล์ระบบที่ถูกอ่านขณะระบบปฏิบัติการกำลังบูต คอนฟิกกูเรชั่นไฟล์ที่สำคัญของระบบและสตาร์ทอัพไฟล์ของผู้ใช้งาน วิธีการตัวอย่างที่แฮ็กเกอร์ใช้ในการค้นหาและฉวยโอกาส เช่น Find / -perm -2 –type f –print

คำสั่ง find จะถูกใช้เพื่อค้นหาไฟล์ที่เซตไว้ให้ทุกคนเขียนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

/etc/rc.d/rc3.d/S99local

/var/tmp

/var/tmp/.X11-unix

/var/tmp/.X11-unix/X0

/var/tmp/.font-unix

/var/lib/game/xgalscores

/var/lib/news/innd/ctlinnda28392

/var/lib/news/innd/ctlinnda18685

/var/spool/fax/outgoing

/var/spool/fax/outgoing/locks

/home/public

จากผลลัพธ์ข้างต้น เราสามารถเห็นปัญหาได้หลายอย่าง ปัญหาอย่างแรก /etc/rc.d/rc3.d/S99local เป็นสตาร์ทอัพสคริปต์ที่เปิดให้ทุกคนสามารถเขียนได้ ปัญหานี้เป็นอันตรายต่อระบบอย่างมาก เพราะเป็นการเปิดช่องให้แฮ็กเกอร์สามรถยกระดับสิทธิให้เท่าเทียมรูตได้โดยง่าย เมื่อระบบปฏิบัติการสตาร์ทขึ้นมา S99local จะถูกเอ็กซีคิวต์ภายใต้สิทธิพิเศษของรูต ดังนั้นแฮ็กเกอร์สามารถสร้างเชลที่เอ็กซีคิวต์คำสั่งได้ตามสิทธิพิเศษของรูตได้ในครั้งถัดไปที่ระบบปฏิบัติการรีสตาร์ทขึ้นมา โดยการทำสิ่งต่อไปนี้

```
[isag07]$ echo "/bin/cp /bin/sh /tmp/.sh ; /bin/chmod 4755 /tmp/.sh"\
/etc/rc.d/rc3.d/S99local
```

ครั้งถัดไปที่ระบบรีบูต เชลดังกล่าวจะถูกสร้างขึ้นมาในไดเรกทอรี /tmp นอกจากนั้นไดเรกทอรี /home/public ก็ยังเปิดให้ทุกคนสามารถเขียนได้ด้วย ดังนั้น แฮ็กเกอร์จึงสามารถเขียนทับไฟล์ใด ๆ ในไดเรกทอรีนี้ด้วยคำสั่ง mv ได้ ซึ่งก็เป็นไปได้เพราะสิทธิที่เซตไว้บนไดเรกทอรีจะมีผลเหนือว่าสิทธิที่เซตไว้บนไฟล์ โดยทั่วไปแฮ็กเกอร์จะแก้ไขเชลสตาร์ทอัพสคริปต์ ซึ่งเป็นสคริปต์ที่ถูกรันไม่ว่าผู้ใช้งานคนไหนจะล็อกออนเข้ามา สคริปต์นี้อยู่ในไดเรกทอรี public (อย่างเช่น สคริปต์ที่ชื่อ .login หรือ .bashrc) สิ่งที่แฮ็กเกอร์แก้ไขก็คือ เพิ่มคำสั่งในการสร้าง SUID user file ขึ้นมา เพื่อให้หลังจากที่ public ล็อกออนเข้าไปในระบบ, SUID public shell จะรอคอยอยู่แล้วสำหรับแฮ็กเกอร์

มาตรการป้องกัน ให้ใช้คำสั่ง find เพื่อค้นหาไฟล์ที่เปิดสิทธิไว้ให้ทุกคนเขียนได้บนเครื่อง และให้เปลี่ยนแปลงสิทธิบนไฟล์หรือไดเรกทอรีต่าง ๆ ที่ไม่มีเหตุผลจำเป็นต้องเปิดให้ใครก็ได้เข้ามาเขียนข้อมูล ถ้าไฟล์นั้นเป็นไฟล์ที่ระบบปฏิบัติการต้องอ่านขณะกำลังสตาร์ทอัพระบบ , คอนฟิกกูเรชั่นไฟล์ที่สำคัญของระบบปฏิบัติการ หรือสตาร์ทอัพไฟล์ที่ต้องรันหลังจากผู้ใช้งานล็อกออนเข้ามา ไฟล์นั้นก็ไม่

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ควรเปิดสิทธิให้ทุกคนสามารถเขียนข้อมูลลงไปได้ แต่บางครั้งมันเป็นเรื่องจำเป็นสำหรับบางดีไวซ์ที่อยู่ในไดเรกทอรี /dev ที่ต้องเปิดสิทธิให้ทุกคนเขียนข้อมูลลงไปได้เพื่อให้มันทำงานได้ตามปกติ

10.1.5 เรื่องของไฟล์ .bash_history

ไฟล์ .bash_history เป็นไฟล์ที่ใช้สำหรับเก็บคำสั่งเชลที่เราพิมพ์ลงไป เหมือนเก็บประวัติ เหตุการณ์ว่าเราทำอะไรลงไปบ้าง ซึ่งปกติจะเป็นไฟล์ธรรมดามีอยู่ในโฮมไดเรกทอรีแต่ยูสเซอร์แต่ละคน จะมีประโยชน์เช่น เอาไว้ดูว่าเราพิมพ์อะไรไป กรณีที่ลืมไปแล้วหรือว่า มีการตั้งค่าผิด ทำให้สามารถรู้และแก้ไขได้ สำหรับตัวอย่างเหตุการณ์ที่ไม่ปกติคือ หลังจากที่ผู้บุกรุกทำการบุกรุกเข้ามาในระบบแล้ว และได้สิทธิรูตไป คำสั่งต่าง ๆ ที่ ผู้บุกรุกใช้จะถูกเก็บอยู่ใน .bash_history ของรูต ก่อนจะออกจากระบบจะทำการลบไฟล์ .bash_history หรือพยายามทำให้คำสั่งที่ใช้หายไปไม่ถูกบันทึก ซึ่งสามารถทำให้ .bash_history ไม่บันทึกคำสั่งที่ใช้ไป ได้หลายวิธี ตัวอย่างเช่น ทำการลบ .bash_history ของเดิมไปแล้วสร้างใหม่เป็นไดเรกทอรีหรือเป็นsymlink ทำการลิงก์ไปที่ /dev/null ซึ่งจะทำให้ ไม่สามารถเก็บเหตุการณ์คำสั่งที่ใช้ไปได้ นั่นหมายถึง เราจะไม่สามารถตามรอยได้ว่าผู้บุกรุกทำอะไรไปกับระบบของราได้บ้าง วิธีการแก้ไขคือ ควรตรวจสอบอย่างสม่ำเสมอว่า ถูกเป็นไปหรือไม่ แล้วทำการแก้ไขเป็นไฟล์ปกติ

10.1.6 เรื่องของ Xinetd

เครื่องให้บริการที่ดีควรจะเปิดให้บริการเฉพาะที่จำเป็นต่อการใช้งานเท่านั้น เนื่องจากการเปิดให้บริการมากเกินความจำเป็นจะทำให้เกิดเป็นช่องโหว่ของระบบ ส่งผลให้ผู้บุกรุกนำไปใช้ในการบุกรุกหรือโจมตีระบบได้ นอกจากนั้นยังเป็นภาระแก่ผู้ดูแลระบบที่จะต้องคอยตรวจสอบติดตามผลการให้บริการ และตรวจสอบข่าวสารเกี่ยวกับความปลอดภัยของบริการที่เปิดใช้อยู่เสมอ เพื่อให้เครื่องให้บริการมีความปลอดภัยสูงสุด

การติดตั้งลีนุกซ์โดยกำหนดเป็นดีฟอลต์จะมีการใช้เซอร์วิสที่มีประโยชน์อื่นๆ อีก ขอให้ทำการตรวจสอบและกำจัดเซอร์วิสที่ไม่จำเป็นเหล่านั้นออก โดยให้ตรวจสอบไฟล์ในไดเรกทอรี /etc/xinetd.d ภายในไฟล์นี้จะเป็นไฟล์ตั้งค่าของแต่ละเซอร์วิสที่รันผ่าน xinetd เป็นตัวกำหนดว่าเซอร์วิสใดบ้างที่ใช้งานได้ การกำจัดเซอร์วิสที่ไม่ต้องการทำได้โดย เข้าไปแก้ไขรายละเอียดของแต่ ละเซอร์วิสโดย set “disable = yes” และสามารถตรวจสอบอีกครั้งเพื่อความแน่ใจโดยใช้คำสั่ง chkconfig –list (ในระบบปฏิบัติการ Redhat)

10.1.7 เรื่องของเวอร์ชัน

กรณีเลือกใช้โปรแกรมซึ่งมีเวอร์ชันที่มีปัญหา เช่น มีบั๊กของโปรแกรมอยู่ โดยปกติแล้วถ้าทราบว่าโปรแกรมของตัวเองนั้นมีช่องโหว่อยู่ ทางผู้ผลิตโปรแกรมนั้นมักจะประกาศให้ผู้ใช้งานโปรแกรมทำการดาวน์โหลดแพตช์ไปติดตั้งเพื่อแก้ปัญหา แต่บางครั้งผู้ดูแลระบบไม่ได้ทำการติดตามข่าวทำให้ไม่ได้อัพเดตแพตช์ของโปรแกรมหรือเซอร์วิส ผู้บุกรุกจึงมักจะใช้จุดบกพร่องในส่วนนี้ค้นหาระบบที่ไม่ได้อัพเดต แล้วทำการบุกรุกผ่านช่องโหว่ของโปรแกรมนั้น ด้วยวิธีการที่แตกต่างกันตามแนวทางต่างๆ เพื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้สามารถผ่านเข้ามาในระบบ ได้ วิธีการแก้ปัญหา คือต้องคอยติดตามข่าวประกาศรายงานบั๊กแทร็กของโปรแกรมต่างๆ ที่เราใช้ อาจทำโดยการเข้าไปตรวจสอบในเวปของผู้ผลิตเอง หรือทำการสมัครรับรายงานทางอีเมล์จากเวปไซด์ต่าง ๆ ที่ให้บริการในเรื่องนี้ ซึ่งหากมีรายงานว่าโปรแกรมในเวอร์ชันที่เราใช้อยู่นั้นมีปัญหา ก็ให้รีบทำการแพตช์หรือเปลี่ยนไปใช้โปรแกรมเวอร์ชันที่ใหม่กว่าหรือใหม่ล่าสุด แต่ก่อนอื่นใดจะต้องทำการตรวจสอบโปรแกรมเวอร์ชันเดิมที่เราใช้อยู่ก่อนว่ามีปัญหาหรือไม่

10.1.8 เรื่องของไฟล์ Lilo

เนื่องจากผู้ดูแลระบบ ส่วนมากมักจะติดตั้ง LILO กับลีนุกซ์จึงมีความจำเป็นต้องกล่าวถึง LILO ซึ่งย่อมาจาก Linux Loader เป็นส่วนที่ใช้สำหรับเลือกอิมเมจที่จะบูตระบบเมื่อทำการเปิดเครื่องและยังสามารถผ่านพารามิเตอร์ต่างๆ ผ่านทางคอมมานไลน์อีกด้วย LILO มักจะเป็นสิ่งที่ถูกมองข้ามไป ถ้าผู้ไม่ประสงค์ดีผ่านค่าพารามิเตอร์บางอย่างจนทำให้สามารถผ่านขั้นตอนการตรวจสอบความปลอดภัยได้ ดังตัวอย่างต่อไป Linux init = /bin/sh

พารามิเตอร์ที่ส่งไปทำให้ผู้ใช้งานสามารถเข้าสู่ลีนุกซ์โดยตรงและได้สิทธิเท่ารูต คือ สามารถทำทุกอย่างได้หรือรันคำสั่งเช่นเดียวกับรูต โดยไม่ต้องต้องใส่รหัสผ่าน จากตัวอย่างจะเห็นได้ว่าจำเป็นต้องมีมาตรการการควบคุมการส่ง คือ การตั้งรหัสผ่านใน Lilo เพื่อให้ผู้ใช้งานต้องใส่รหัสผ่านเมื่อต้องการรันคำสั่งใดๆ แต่เนื่องจากว่ารหัสผ่านจะถูกเก็บเป็นตัวอักษรธรรมดา จึงจำเป็นต้องเปลี่ยนโหมดไฟล์ให้เป็นโหมด 0640 คือ จะอนุญาตให้รูตเพียงผู้เดียวที่ทำการอ่านและเขียนได้ การแปลงโหมดทำโดยใช้คำสั่ง

chmod 0640 /etc/lilo.conf

10.1.9 เรื่องของไฟล์ Securetty

ภายในไฟล์ /etc/securetty จะเก็บรายชื่อของเทอร์มินอลที่อนุญาตให้รูตสามารถล็อกอินผ่านเทอร์มินอล นั้นเข้ามาได้ และถ้ามีเทอร์มินอลมากเท่าไรจะยิ่งเป็นการเปิดโอกาสให้แฮ็กเกอร์พยายามเดารหัสผ่านรูตได้หลายหนทาง และหลายรูปแบบมากขึ้นเท่านั้น

วิธีการป้องกันอย่างหนึ่งคือ ลบรายการเทอร์มินอลที่ไม่จำเป็นออก อาจเหลือไว้ใช้แค่เพียง tty1 เท่านั้น นั่นหมายความว่ารูตจะสามารถทำการล็อกอินเข้าระบบผ่าน คอนโซลได้โดยตรงเท่านั้น หากต้องการล็อกอินจากระยะไกล จะต้องล็อกอินผ่านยูสเซอร์แอคเคาน์ปกติเข้ามาก่อนแล้วจึงใช้คำสั่ง su เปลี่ยนสิทธิเป็นรูต ซึ่งวิธีนี้ถือเป็นการป้องกันอันตรายได้อย่างหนึ่ง เนื่องจากหากแฮ็กเกอร์สามารถทำการแคร็กรหัสผ่านรูต หรือดักจับรหัสผ่านรูตได้ ก็จะยังไม่สามารถล็อกอินเข้าสู่ระบบได้ในทันที จำเป็นต้องแคร็กรหัสผ่านของยูสเซอร์ธรรมดาในระบบเสียก่อน ถือเป็นการป้องกันความปลอดภัยมากขึ้นอีก 1 ชั้น

10.1.10 เรื่องของไฟล์ .rhost

ประตูหลังrhosts ++ บนระบบเครือข่ายยูนิกซ์เซอร์วิส เช่น Rsh และ Rlogin ใช้ขั้นตอนวิธีการพิสูจน์ตนอยู่บนพื้นฐานของชื่อเครื่องซึ่งปรากฏอยู่ในไฟล์ .rhosts ยูสเซอร์สามารถตั้งค่าได้ง่ายซึ่งไม่ต้อง

เอกสารนี้เป็น... ไม่สามารถ... ใช้...

ใช้รหัสผ่านในการล็อกอินเข้ามา และผู้บุกรุกสามารถได้รับสิทธิของใครก็ได้ที่มีไฟล์ .rhosts ซึ่งมีเครื่องหมาย " + + " ในไฟล์ ซึ่งจะอนุญาตให้ใครก็ได้จากที่ไหนก็ได้สามารถล็อกอินเข้ามาใช้แอคเคาน์นั้นได้ โดยปราศจากรหัสผ่าน ผู้บุกรุกส่วนมากใช้วิธีการพิเศษเมื่อ NFS ทำการส่งโฮมไดเรกทอรีไปยังภายนอก แอคเคาน์นี้จะกลายเป็น backdoor สำหรับผู้บุกรุกที่จะกลับเข้ามายังระบบอีกครั้ง ผู้บุกรุกส่วนมากชอบที่จะใช้ Rsh มากกว่า Rlogin เนื่องจากหลายครั้งที่ Rsh ไม่มีการเก็บล็อกไว้ และผู้ดูแลระบบควรตรวจสอบเครื่องหมาย " + + " ป้องกันผู้บุกรุกอาจทำการใส่ชื่อโฮสต์เนมและยูสเซอร์เนมจากที่อื่น ทำให้มองเห็นได้ยาก

10.1.11 เรื่องของรหัสผ่าน

เรื่องความไม่ปลอดภัยในระบบอันเกิดจากการเลือกใช้รหัสผ่านที่ไม่ดีพอ หรือง่ายต่อการคาดเดานั้น ทำให้ผู้บุกรุกสามารถทำการแคร็กรหัสผ่านได้ง่าย เช่น ตั้งรหัสผ่าน ตามชื่อเล่น ชื่อจริง นามสกุล วันเดือนปีเกิด หรือตัวเลขเรียงกัน และภายหลังที่ทราบรหัสผ่านของยูสเซอร์ในระบบแล้ว ผู้บุกรุกจะทำการล็อกอินเข้ามาในระบบ โดยปลอมตัวมาเป็นยูสเซอร์ปกติที่แคร็กรหัสผ่านได้ และพยายามค้นหาช่องโหว่ของระบบอย่างอื่น เพื่อนำมาให้ได้ซึ่งสิทธิของรูต เราสามารถป้องกันไม่ให้เกิดปัญหานี้ได้โดยการตั้งนโยบายบังคับให้ยูสเซอร์ในระบบตั้งรหัสผ่านที่มั่นคงปลอดภัย ไม่ง่ายเกินไปที่จะเดาได้ หรืออาจจะทำการทดลองแคร็กรหัสผ่านของยูสเซอร์ในระบบเองก็ได้ เป็นต้น

จากที่กล่าวมาข้างต้นทำให้เห็นถึงอันตรายที่จะเกิดขึ้นต่อระบบที่เกิดจากสาเหตุข้างต้น ทำให้เกิดการพัฒนาโปรแกรม Security System Scanner เพื่อให้มีความสามารถในการตรวจสอบหาช่องโหว่ของระบบอันเกิดจาก

1. Unowned Files ไฟล์ที่ไม่สามารถระบุเจ้าของได้
2. Symbolic link หรือ Soft link ที่ลิงก์ไปในที่ที่ไม่สมควรอนุญาต หรือเป็นลิงก์เสีย
3. SUID/SGID ไฟล์ที่มีการ set-uid หรือ set-gid
4. World Writable ไฟล์ที่อนุญาตให้ทุกคนสามารถลบหรือเขียนได้
5. /etc/securetty การอนุญาตให้รูตสามารถล็อกอินจากระยะไกลเข้าระบบได้
6. xinetd การเปิดเซอร์วิสไว้มากเกินความจำเป็น
7. /etc/lilo.conf การไม่ตั้งรหัสผ่าน LILO
8. รหัสผ่านแบบง่ายๆ การเลือกใช้รหัสผ่านที่ไม่ปลอดภัยพอง่ายต่อการคาดเดา
9. ไฟล์ .rhosts การอนุญาตให้มีไฟล์ .rhosts อยู่ในโฮมไดเรกทอรีของยูสเซอร์แต่ละคน โดยเฉพาะรูตโฮมไดเรกทอรี
10. .bash_history ไฟล์ .bash_history ที่ผิดปกติ คือ เป็นไดเรกทอรีหรือ symbolic link ลิงก์ไปยัง /dev/null
11. การใช้โปรแกรมเวอร์ชันมีปัญหา มีบั๊ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยเลือกใช้ภาษาเพิร์ลสคริปต์และ mysql server บนระบบปฏิบัติการลีนุกซ์อ้างอิง Redhat 8.0 เป็นหลัก เนื่องจากภาษาเพิร์ลมีความสามารถอย่างดีในเรื่องของการทำ เท็กซ์แมนิปูเลชั่น และ สามารถนำไปพัฒนาต่อได้ง่าย เนื่องจากจะต้องมีการเปรียบเทียบกับข้อมูลรายงานบั๊กแทร็ก ซึ่งได้จากการรวบรวมและจัดทำเป็นฐานข้อมูลเก็บไว้เพื่อเผยแพร่ของ http://cve.mitre.org จึงเลือกใช้ mysql server เป็นฐานข้อมูล และสาเหตุที่เลือกระบบปฏิบัติการลีนุกซ์ เนื่องจากระบบปฏิบัติการลีนุกซ์เป็นระบบปฏิบัติการแบบ open source ทำให้ง่ายต่อการพัฒนา ค้นคว้า และวิจัย ต่อไป

10.2 ออกแบบโปรแกรม

จุดประสงค์หลักของโปรแกรม Security System Scanner คือ การตรวจสอบให้ได้ว่าอะไรเป็นช่องโหว่ที่อยู่ในระบบที่เกิดจากเหตุผลที่กล่าวไปแล้วข้างต้น จึงเลือกใช้วิธีการสแกนไฟล์ทั้งระบบเพราะจะสามารถทำได้อย่างละเอียด

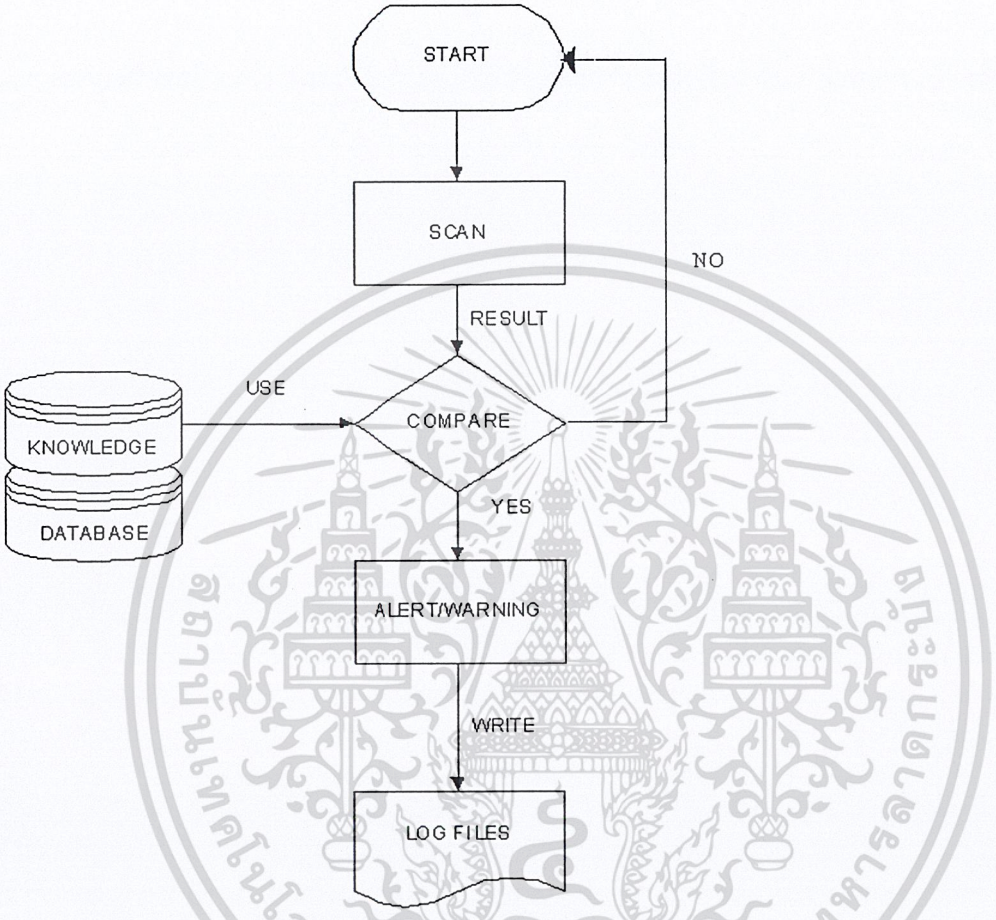
โดยจะสแกนเป็นรอบของแต่ละหัวข้อ เพราะการใช้คำสั่ง find นั้นมีข้อจำกัด คือ จะไม่สามารถสแกนหาไฟล์หลายๆ ประเภทที่ไม่เกี่ยวข้องกันภายในรอบเดียวกันได้

หลังจากหาไฟล์ที่ได้ผลลัพธ์ตามที่ต้องการแล้วจะนำไปเปรียบเทียบกับ ฐานข้อมูลหรือฐานความรู้เพื่อใช้ในการคัดแยกว่า อะไรคือช่องโหว่ที่แท้จริง เพราะบางครั้งเราจำเป็นที่จะต้องอนุญาตให้มีอยู่ในระบบเพราะมิฉะนั้นแล้วจะไม่สามารถใช้งานได้ ซึ่งจะทำการเทียบผลลัพธ์จากการสแกนกับฐานข้อมูล

แล้วทำการรายงานผลออกมา พร้อมทั้งจัดเก็บผลลัพธ์ทั้งหมดไว้ในล็อกไฟล์ ซึ่งตั้งชื่อตามวันที่และเวลาที่ทำการสแกน เพื่อป้องกันการซ้ำกันของชื่อล็อกไฟล์ และสามารถตรวจสอบเปรียบเทียบช่องโหว่ตามวันที่และเวลาที่ทำการสแกนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยภาพรวมของลักษณะการทำงานของแต่ละฟังก์ชันของโปรแกรมจะมีลักษณะคล้ายกันซึ่งเป็นไปตามรูป



รูปที่ 10-1 : แสดงโฟลว์ชาร์ตของตัวโปรแกรม

สำหรับโครงสร้างภายในได้แบ่งเป็นฟังก์ชัน ซึ่งแต่ละฟังก์ชันจะทำหน้าที่หาช่องโหว่ที่เกิดจากสาเหตุแต่ละประเภท เพื่อให้ยูสเซอร์สามารถเลือกสแกนหาช่องโหว่เฉพาะบางหัวข้อได้ และทำงานเป็นรูทีน เมื่อควบคู่กับ cron

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10.3 ฟังก์ชันการทำงานของโปรแกรมตรวจสอบช่องโหว่ความปลอดภัยระบบ

โครงสร้างโปรแกรมตรวจสอบช่องโหว่ความปลอดภัยระบบ ประกอบด้วยฟังก์ชันดังต่อไปนี้

- **sub unowned()**

หน้าที่ เป็นฟังก์ชันที่ใช้สำหรับสแกนหาไฟล์ที่มีคุณสมบัติเป็น unowned file คือ ไฟล์ที่ไม่มีเจ้าของ โดยใช้คำสั่ง find / (-nouser –o –nogroup)

ผลลัพธ์ แสดงผลการสแกนที่ได้ไว้ในไฟล์ unowned.tmp

- **sub symbolic()**

หน้าที่เป็นฟังก์ชันที่ใช้สำหรับสแกนหาไฟล์ที่มีคุณสมบัติเป็น symbolic link โดยใช้คำสั่ง find / -type l แล้วนำไฟล์ที่เป็น symbolic linkที่ได้ ไปตรวจสอบดูว่า

1. เป็นไฟล์ที่มีเจ้าของเป็นรูตแล้วลิงก์ไปยัง world writable file หรือไม่
2. เป็นไฟล์ที่มีเจ้าของเป็นยูสเซอร์ธรรมดาแล้วลิงก์ไปยังไฟล์ที่มีเจ้าของเป็นรูตหรือไม่
3. เป็นไฟล์ที่เป็นลิงก์เสีย อันเกิดจากไฟล์ปลายทางไม่มีอยู่จริง หรือเกิดจากไฟล์ปลายทางเป็นsymbolic link แล้วลิงก์เป็นลูปกลับมาที่ตัวเอง

ผลลัพธ์ที่ได้จากการเปรียบเทียบ 3 ข้อข้างต้นจะถูกแสดงไว้ที่ file symbolic.tmp

- **sub suid()**

หน้าที่เป็นฟังก์ชันที่ใช้สำหรับสแกนหาไฟล์ที่มีการกำหนด SUID/SGID โดยใช้คำสั่ง find / (-perm –2000 –o –perm –4000) แล้วนำไฟล์ SUID/SGID ที่ได้เปรียบเทียบกับฐานความรู้ที่มีเก็บไว้

1. กรณีที่ตรงกับข้อมูลในฐานความรู้ จะทำการเตือน พร้อมให้คำแนะนำ
2. กรณีที่ไม่ตรงกับข้อมูลในฐานความรู้ จะทำรายงานผลออกมาว่าไฟล์นี้เป็น SUID/SGID ให้ผู้ดูแลระบบทำการพิจารณาเองว่า จะยกเลิก s bit หรือไม่

ผลลัพธ์ที่ได้จากการเปรียบเทียบนั้น จะถูกแสดงไว้ที่ file suidsgid.tmp

- **sub world()**

หน้าที่เป็นฟังก์ชันที่ใช้สำหรับสแกนหาไฟล์หรือไดเรกทอรีที่อนุญาตให้ทุกคนสามารถ เขียน หรือ ลบไฟล์ได้ โดยใช้คำสั่ง find / (-perm –2 ! –type l)

ผลลัพธ์ที่ได้จากการสแกนนั้น จะถูกแสดงไว้ที่ file world.tmp

- **sub securetty()**

หน้าที่เป็นฟังก์ชันที่ใช้สำหรับสแกนดูรายละเอียดภายในไฟล์ /etc/securetty ว่ามีรายการของ ttyp* อยู่หรือไม่ โดยการวนลูปและตัดคำมาเทียบทีละบรรทัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลลัพธ์จะถูกรายงานใน file securetty.tmp เฉพาะที่มี ttyp อยู่ในรายการ

- **sub xinetd()**

หน้าที่เป็นฟังก์ชันที่ใช้สำหรับสแกนดูรายละเอียดของทุกไฟล์ที่อยู่ใน /etc/xinetd.d ว่ามีเซอร์วิสใดเปิดอยู่ ซึ่งภายในจะมีบรรทัดที่เขียนว่า disable = no ถ้ามีให้เปรียบเทียบกับฐานความรู้ว่าเซอร์วิส เพื่อขอคำแนะนำและความรู้เกี่ยวกับเซอร์วิสนั้น

ผลลัพธ์จะถูกรายงานเฉพาะ serviceที่เปิดอยู่ อยู่ใน file xinetd.tmp

- **sub lilo()**

หน้าที่เป็นฟังก์ชันที่ใช้สำหรับสแกนดูรายละเอียดภายในไฟล์ /etc/lilo.conf ว่ามีการกำหนด รหัสผ่านของ lilo ไว้หรือไม่ โดยหาบรรทัดที่เขียนว่า "password = " ถ้ามีแสดงว่ามีการกำหนดรหัสผ่านไว้

ผลลัพธ์จะถูกรายงานออกมาในกรณีที่ไม่มีการกำหนด password ไว้

- **sub crack()**

หน้าที่เป็นฟังก์ชันที่ใช้สำหรับทดสอบรหัสผ่านของยูสเซอร์ที่มีการกำหนดรหัสผ่านในระบบว่าตั้งรหัสผ่านที่ง่ายเกินไปหรือไม่ โดยทำการสุ่มรหัสผ่านจาก GECOS และตัวเลข 4 หลักตั้งแต่ 0000-3113 ซึ่งได้จาก วันที่และเดือน หรือ เดือนและวันที่ ซึ่งตัวเลขนี้ครอบคลุมทั้ง 31 วัน 12 เดือน และคำศัพท์ในดิกชันนารีมาเข้าฟังก์ชัน crypt คู่กับ salt ซึ่ง salt จะหาได้มาจากไฟล์ /etc/shadow

ถ้าเป็นการเข้ารหัสแบบ DES salt จะเป็นอักขระ 2 ตัวแรก ของ shadow

ถ้าเป็นMD5 salt จะเป็นอักขระ 12 ตัว ขึ้นต้นด้วยเครื่องหมาย 1 และ อักขระอื่น ๆ 8 ตัว ปิดท้ายด้วยอักษขระ $ อีก 1 ตัว เช่น 1ÄY8öÜISù$

แล้วนำผลลัพธ์ที่ได้จากฟังก์ชัน crypt ไปเทียบกับ shadow ส่วนที่ตัด salt ออกแล้ว ถ้าตรงกันจะสามารถสรุปได้ว่ารหัสผ่านของยูสเซอร์คนนั้นคือคำที่สุ่มมาได้ ผลลัพธ์ คือ รหัสผ่านที่สามารถแคร็กได้จะถูกรายงานเก็บไว้ที่ file password.tmp

- **sub rhosts()**

หน้าที่เป็นฟังก์ชันที่ใช้สำหรับสแกนหาไฟล์ที่มีชื่อ .rhosts ในระบบ ว่ามีอยู่ที่ไหนบ้างแล้วทำการรายงานผลออกมา

ผลลัพธ์จะถูกรายงานออกมาใน file rhosts.tmp

- **sub bash_history()**

หน้าที่เป็นฟังก์ชันที่ใช้สำหรับสแกนหาไฟล์ในระบบที่มีชื่อว่า .bash_history ที่เป็นไดเรกทอรีหรือ เป็น symbolic link ที่ลิงก์ไปยัง /dev/null

ผลลัพธ์จะถูกเก็บอยู่ใน bashhistory.tmp

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **sub version()**

เป็นฟังก์ชันที่ใช้สำหรับตรวจสอบเวอร์ชันของ package rpm ที่ลงอยู่ในเครื่องว่าเป็น เวอร์ชันที่มีการรายงานเกี่ยวกับบั๊กไว้หรือไม่ โดยการเทียบกับฐานข้อมูลที่ดาวน์โหลดจาก http://cve.mitre.org โดยการใช้คำสั่ง rpm –qai เป็นการหารายชื่อพร้อมเลขเวอร์ชันของ package rpm ที่ลงอยู่ นำไปquery หาผลลัพธ์ในฐานข้อมูล บน mysql serve ซึ่งถ้ามีอยู่ในฐานข้อมูลแสดงว่าเซอร์วิสเวอร์ชันนั้นมีปัญหาไม่ควรใช้ ควรทำการแพตช์ หรือ อัพเดตใช้โปรแกรมที่มีเวอร์ชันที่ใหม่กว่า

ผลลัพธ์ที่ได้จะเป็น ชื่อ โปรแกรม และเวอร์ชัน และ คำรายงาน bugtrag ของโปรแกรมที่มีปัญหา

10.4 ข้อจำกัดของระบบ

ระบบตรวจสอบช่องโหว่ความปลอดภัยระบบ สร้างขึ้นโดยมีข้อจำกัดของการใช้งาน ดังต่อไปนี้

- ระบบนี้สร้างขึ้นบนระบบปฏิบัติการลีนุกซ์ ซึ่งอาจแตกต่างกัน หากนำไปใช้งานในระบบอื่น
- ผู้ที่มีสิทธิรูตเท่านั้นที่จะสามารถใช้งานโปรแกรมนี้ได้
- ระบบนี้สามารถตรวจสอบช่องโหว่ความปลอดภัยระบบได้เฉพาะเครื่องที่รันโปรแกรมเท่านั้น
- ในกรณีที่ระบบเป็นระบบขององค์กรขนาดใหญ่ มีผู้ใช้งานในระบบเป็นจำนวนมาก จะทำให้ใช้เวลาในการตรวจสอบนานขึ้น
- ในกรณีที่ใช้ควบคู่กับ cron เพื่อใช้ในการตั้งเวลาการตรวจสอบช่องโหว่เป็นช่วงระยะเวลา เช่น ทุก ๆ สัปดาห์ มีข้อจำกัดคือ ระยะเวลาที่ตั้งไว้ต้องห่างกันมากกว่าเวลาที่ใช้ในการตรวจสอบในแต่ละรอบ ซึ่งเวลาที่ใช้ในการตรวจสอบระบบจะขึ้นกับขนาดของระบบ เช่น ถ้าใช้เวลาในการตรวจสอบระบบ 10 นาที จะต้องตั้งเวลาให้ทำการตรวจสอบระบบซ้ำอีกรอบอย่างน้อย ห่างกันมากกว่า 10 นาทีเป็นต้นไป
- หากต้องการข้อมูลรายงานบั๊กแทร็กล่าสุด จะต้องทำการดาวน์โหลดและเปลี่ยนข้อมูลลงสู่ฐานข้อมูลเอง โดยดาวน์โหลดได้จาก http://cve.mitre.org/cve เนื่องจากยังไม่มีการทำส่วนการอัพเดตฐานข้อมูลอย่าง อัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 11
สรุป

11.1 สรุป

ระบบตรวจสอบช่องโหว่ความปลอดภัยระบบ ถูกออกแบบมาเพื่อสำหรับการใช้งานของผู้ดูแลระบบ เพื่อช่วยอำนวยความสะดวก ง่ายต่อการตรวจหาความเสี่ยงของช่องโหว่ในระบบ เป็นระบบที่ช่วยตรวจสอบประเมินความเสี่ยงของระบบ ไม่สามารถที่จะดักจับผู้บุกรุกได้ จะมีประโยชน์ก็ต่อเมื่อผู้ดูแลระบบนำผลลัพธ์ที่ได้ไปปรับปรุงแก้ไข

11.2 แนวทางการปรับปรุงในอนาคต

ระบบตรวจสอบช่องโหว่ความปลอดภัยระบบ ยังมีข้อเสียหรือจุดด้อยบางประการ กล่าวคือ ฟีเจอร์ที่สามารถสแกนได้ยังมีจำนวนน้อย ซึ่งจะสามารถพัฒนาได้โดยเพิ่มฟีเจอร์เป็นโมดูลเข้าไปได้ทันที และเนื่องจากระบบตรวจสอบช่องโหว่ความปลอดภัยระบบนี้เป็นโลคอลสแกนเนอร์ (local scanner) คือ จะสามารถทำการันได้เฉพาะเครื่องที่ทำการติดตั้งโปรแกรมเท่านั้น จึงควรจะปรับปรุงให้สามารถสแกนผ่านเน็ตเวิร์กได้ เพื่อการมีประสิทธิภาพที่ดีขึ้น ดูฟีเจอร์ที่สามารถทำได้ดังตารางที่ 11-1

ผลการสแกนที่ได้เป็นเพียงการรายงานผลเท่านั้น วิธีหนึ่งที่จะช่วยอำนวยความสะดวกให้แก่ผู้ดูแลระบบมากขึ้น คือ พัฒนาโปรแกรมให้มีความสามารถที่จะปรับปรุงซ่อมแซมช่องโหว่ได้อย่างอัตโนมัติ ข้อจำกัดอีกอย่างหนึ่งของโปรแกรมคือไม่สามารถอัพเดตฐานข้อมูลล่าสุดเองได้ รายงานบั๊กแทร็ค จาก http://cve.mitre.org แล้วทำการคอนเวิร์ตลงฐานข้อมูลได้โดยอัตโนมัติ ทำให้เกิดข้อเสียคือ ผู้ดูแลระบบยังจะต้องติดตามข่าวรายงานเวอร์ชันของโปรแกรมที่มีปัญหาอยู่ดี

การพัฒนาระบบตรวจสอบช่องโหว่ความปลอดภัยระบบนี้ ทำการพัฒนาบนระบบปฏิบัติการลีนุกซ์ redhat 8.0 ซึ่งมีโครงสร้างระบบไฟล์แตกต่างจากลีนุกซ์ยี่ห้ออยู่อีก ดังนั้นควรพัฒนาโปรแกรมให้สามารถรองรับหลาย ๆ ยี่ห้อได้ แต่ทั้งนี้ผู้ที่ทำการพัฒนาจะต้องมีความรู้ความเข้าใจในเรื่องการทำงานของระบบปฏิบัติการลีนุกซ์ได้เป็นอย่างดี และมีความรู้ความสนใจระบบทางด้านความปลอดภัย ระบบตรวจสอบผู้บุกรุกไฟร์วอลล์ เพื่อการพัฒนาที่ถูกต้องและรวดเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Checks	Vetescan Local	SSS
HOME/ROOT Directory accessability.	Y	
inetd services.	Y	Y
/etc/securetty	Y	Y
SUID/SGID.	Y	Y
World-Writable	Y	Y
Unowned Files	Y	Y
.rhosts	Y	Y
Cracks Password/Shadow	Y	Y
RootKits	Y	
.bash_history		Y
Version Services		Y
Symbolic link		Y
Lilo.conf		Y

ตารางที่ 11-1 : ตารางเปรียบเทียบระหว่าง Vetescan Local และ Security System Scanner

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก
การใช้โปรแกรม crontab

อะไรคือ crontrab ?

crontab เป็นโปรแกรมที่ยอมให้ผู้ใช้สร้างงานขึ้นมาโดยการตั้งเวลาให้กับมัน ผู้ใช้แต่ละคนก็จะมีไฟล์ crontab ของแต่ละคน และทั้งระบบก็จะมีโปรแกรม crontab ที่สามารถกำหนดจากรูตเท่านั้น ถ้าคุณเป็นรูตแล้วอยากจะกำหนด crontab ให้กับยูสเซอร์คนใด สามารถทำได้โดยใช้คำสั่ง su แล้วตามด้วยยูสเซอร์เนมนั้น

รูปแบบของไฟล์นั้นประกอบด้วย 6 ฟิลด์ แต่ละฟิลด์แยกกันด้วยการเว้นวรรค ใน 5 ฟิลด์แรกจะเป็นตัวบอกเวลาว่าจะให้โปรแกรมนั้นๆ รันที่เวลาอะไร และฟิลด์ที่ 6 จะเป็นคำสั่งว่าให้รันโปรแกรมอะไร

รูปแบบ

minute	hour	day	month	weekday	command

minute - จำนวนนาทีโดยนับจากชั่วโมงเต็ม (0-59)
hour - ชั่วโมงในหนึ่งวัน (0-23)
day - วันที่ในหนึ่งเดือน (1-31)
month - เดือนในหนึ่งปี (1-12)
weekly - วันในหนึ่งสัปดาห์ (0-6 ; โดย 0 หมายถึง วันอาทิตย์)

ให้ใช้เครื่องหมายดอกจัน (*) เมื่อต้องการให้คำสั่งวิ่งทุกๆ ค่าของฟิลด์นั้นๆ เช่น ในฟิลด์ hour เป็นดอกจัน จะหมายถึงให้วิ่งทุกๆ ชั่วโมง

ข้อกำหนด

crontab –e

แก้ไข crontab ปัจจุบัน หรือสร้างcrontab อันใหม่. *

crontab -l

แสดงรายการสิ่งที่บรรจุของไฟล์ crontab.

crontab -r

เอา ไฟล์ crontab ออก.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้าง crontab สำหรับใช้กับ AccessWatch

เมื่อต้องการสร้างไฟล์ crontab เราจะต้องเป็นผู้ใช้ปกติ อย่าสร้างเมื่อเป็นผู้ใช้ปกติโดยผ่านคำสั่ง su เข้ามา

1. su userid ex. su ohho
2. สร้าง crontab ด้วยคำสั่ง

crontab –e

ตัวอย่าง 1

30 23 * * * /home/userid/www/aw/aw-parser.pl -d

45 23 * * * /home/userid/www/aw/aw-report.pl

หมายความว่า AccessWatch จะทำงานเก็บข้อมูลทุกวันเวลา 23.30ไว้ในล็อกไฟล์ในโฮมไดเรกทอรีของแต่ละคน และออกมาเป็นรายงานทุกวันในเวลา 23.45

ตัวอย่าง2

Clean up core files every weekday morning at 3:15 am

15 3 * * 1-5 find $HOME -name core 2>/dev/null | xargs rm -f

ตัวอย่าง 3

Mail a birthday greeting:

0 12 14 2 * mailx john%Happy Birthday!%Time for lunch.

ตัวอย่าง 4 เป็นตัวอย่างการเจาะจงของวัน 2 ชนิด

0 0 1,15 * 1

would run a command on the first and fifteenth of each month, as well as on every Monday. To specify days by only one field, the other field should be set to *, for example:

0 0 * * 1

would run a command only on Mondays.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

CVE

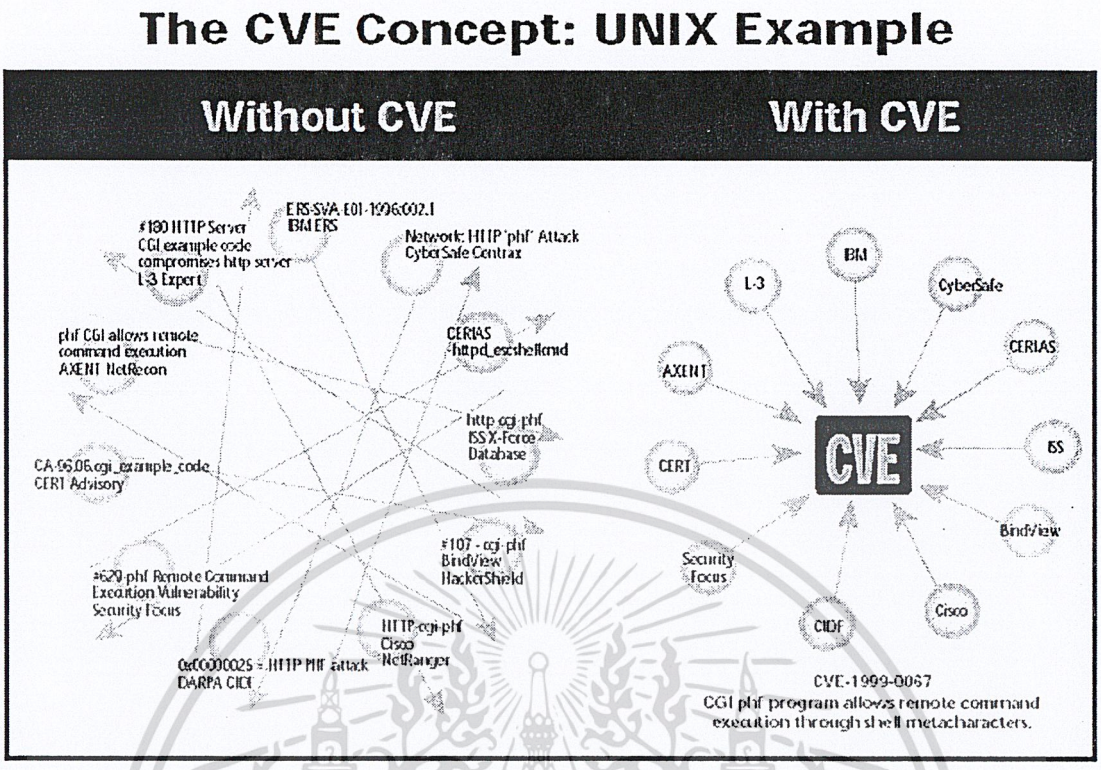
What Is CVE?

Common Vulnerabilities & Exposures (CVE) is a list or dictionary that provides common names for publicly known information security vulnerabilities and exposures. Using a common name makes it easier to share data across separate databases and tools that until now were not easily integrated. This makes CVE the key to information sharing. If a report from one of your security tools incorporates CVE names, you may then quickly and accurately access fix information in one or more separate CVE-compatible databases to remediate the problem.

CVE is:

- One name for one vulnerability or exposure
- One standardized description for each vulnerability or exposure
- A dictionary rather than a database
- How disparate databases and tools can "speak" the same language
- The way to interoperability and better security coverage
- A basis for evaluation among tools and databases
- Accessible for review or download from the Internet
- Industry-endorsed via the CVE Editorial Board

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Why CVE?

Most information security tools include a database of security vulnerabilities and exposures; however, there is significant variation among them and no easy way to determine when different databases are referring to the same problem. The consequences are potential gaps in security coverage and no effective interoperability among the disparate databases and tools. In addition, each tool vendor currently uses different metrics to state the number of vulnerabilities or exposures they detect, which means there is no standardized basis for evaluation among the tools. With a standard list of vulnerabilities and exposures such as CVE, your databases and tools can "speak" to each other. And, you'll know exactly what each tool covers because CVE provides you with a baseline for evaluating the coverage of your tools. This means you can determine which tools are most effective and appropriate for your organization's needs. In short, CVE-compatible tools and databases will give you better coverage, easier interoperability, and enhanced security. CVE is also endorsed by leading representatives from the information security community. CVE's content results from the collaborative efforts of the CVE Editorial Board, which includes representatives from more than 15 information ecurity-related organizations.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Who Is the Editorial Board?

The CVE Editorial Board includes members from numerous information security-related organizations including commercial security tool vendors, members of academia, research institutions, government agencies, and other prominent security experts. Through open and collaborative discussions, the Board identifies which vulnerabilities or exposures are included in CVE, then determines the common name and description for each entry. The MITRE Corporation created the Editorial Board, moderates Board discussions, and provides guidance throughout the process to ensure that CVE serves the public interest. Archives of Board meetings and discussions are available for review on the CVE web site. Other information security experts will be invited to participate on the Board on an as-needed basis based upon recommendations from Board members.

What It Means To Be CVE Compatible

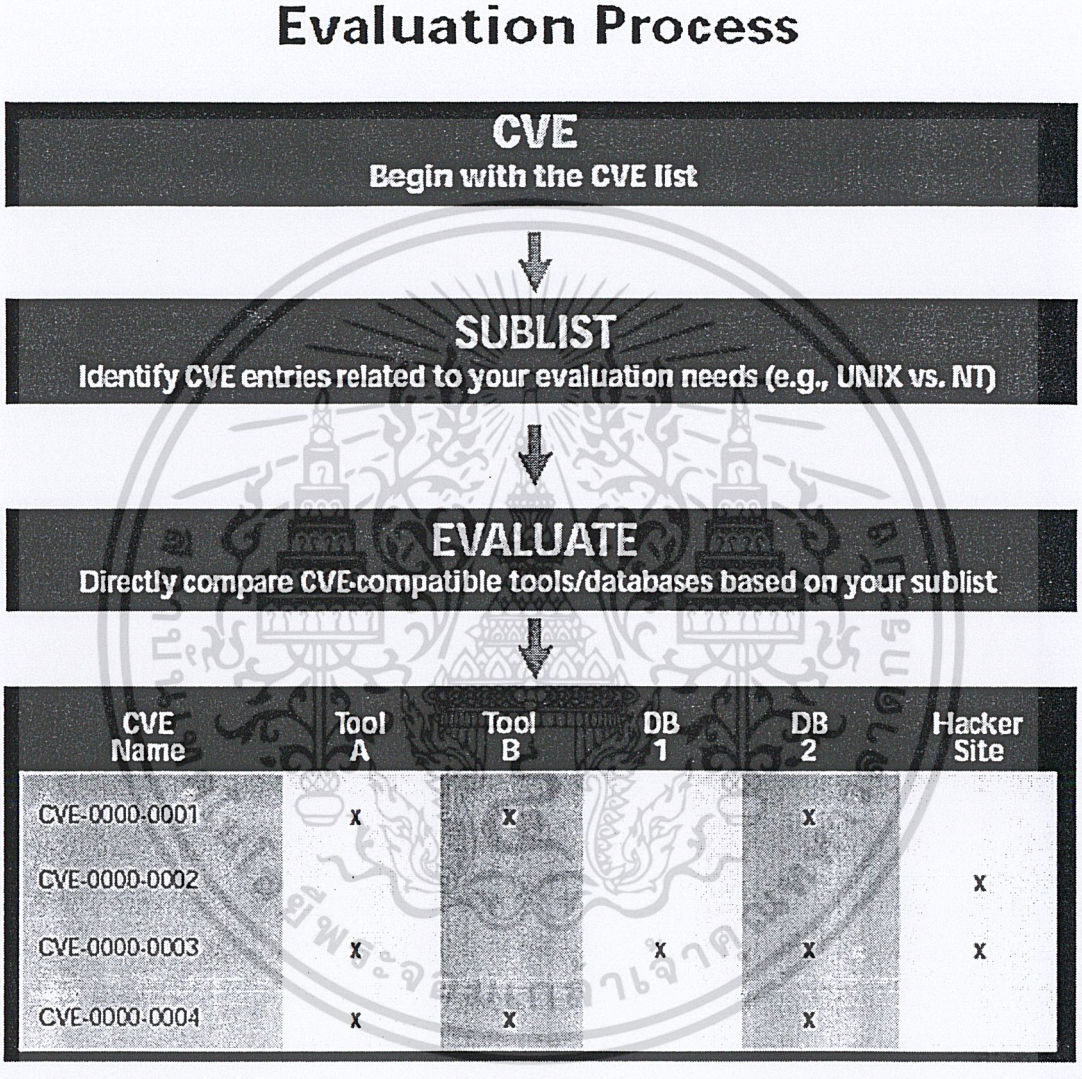
"CVE-Compatible" means that a tool, Web site, database, or other security product uses CVE names in a manner that allows it to be cross-referenced with other products that employ CVE names. "CVE compatible" means: _ CVE SEARCHABLE — A user can search using a CVE name to find related information. _ CVE OUTPUT — Information is presented that includes the related CVE name(s). _ ACCURACY — The provider has made a good faith effort to ensure that CVE names are used accurately in the product. Different tools provide different coverage/cross-referencing of CVE names e.g., some tools might cover Unix, while others cover Windows NT). You will need to evaluate any CVE-compatible products based upon your organization's specific requirements. Visit the site for the most urrent information regarding the types and availability of CVE-compatible products.

The CVE Naming Process

The process begins with the discovery of a potential security vulnerability or exposure. The informations then assigned a CVE candidate number. The Editorial Board discusses the candidate and votes on hether or not it should become a CVE entry. If the

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

candidate is accepted, it is entered into CVE and is published on the CVE Web site. Candidates can be searched on the site, but the CVE and candidates lists are separate.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

หนังสืออ้างอิง

(1) สมชาย นำประเสริฐชัย, "การใช้งานระบบปฏิบัติการลีนุกซ์", ซีเอ็ดยูเคชัน, 2543

(2) วรวุฒิ เที่ยงธรรม, สันติ ศรีลาศักดิ์, "รู้จักลีนุกซ์", ออฟเซท เพรส, 2542

(3) ก่อกิจ วีระอาชากุล, "ติดตั้ง และปรับแต่งเซิร์ฟเวอร์ Linux สำหรับ Admin Linux โดยเฉพาะ", อินโฟเพรส, 2545

(4) สันติ ศรีลาศักดิ์, วรวุฒิ เที่ยงธรรม, "เจาะประเด็นงานเขียนโปรแกรมบนลีนุกซ์", ออฟเซท เพรส, 2542

(5) Joel Scambray, Stuart MnNlure, George Kurtz, เรียบเรียงโดย เอกสิทธิ์ วิริยจารี, "Hacking Exposed ปิดทางแฮกเกอร์", ซีเอ็ดยูเคชัน, 2544

(6) Jon Orwant, "PERL 5", eZONE, 1998

(7) Eric C. Herrmann, "Perl 5", SYBEX

เวปไซด์อ้างอิง

(1) http://www.google.co.th

(2) http://thaicert.nectec.or.th

(3) http://cve.mitre.org

(4) http://perldoc.com

(5) http://mysql.com

(6) http://fish.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้