ระบบต่อต้านชุดโปรแกรมปกปิดการบุกรุก

Anti-Rootkit System

นายกมล เอาหสุขไพศาล

นายจรูญชัย ธรรมพิพัฒน์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2549

ระบบต่อต้านชุดโปรแกรมปกปิดการบุกรุก

Anti-Rootkit System

โดย

นายกมล เฉาหชุชไพศาล

นายจรูญชัย ธรรมพิพัฒน์

อาจารย์ที่ปรึกษา

อาจารย์อัครเดช วัชระภูพงษ์

ผศ. ธนา หงษ์สุวรรณ

อาจารย์ธนัญชัย ตรีภาค

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2549

ปริญญานิพนธ์ปีการศึกษา 2549

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบต่อต้านชุดโปรแกรมปกปิดการบุกรุก

Anti-Rootkit System

ผู้จัดทำ

1. นายกมล เลาหสุขไพศาล รหัสนักศึกษา 46010004

2. นายจรูญชัย ธรรมพิพัฒน์ รหัสนักศึกษา 46010097

_____ อาจารย์ที่ปรึกษา

(อาจารย์ อัครเดช วัชระภูพงษ์)

_____ อาจารย์ที่ปรึกษา

(ผศ. ธนา หงษ์สุวรรณ)

_____ อาจารย์ที่ปรึกษา

(อาจารย์ ธนัญชัย ตรีภาค)

Anti-Rootkit System

Mr. Kamon	Laohasukpaisan	46010004
Mr. Jaroonchai	Thampipat	46010097
Mr. Akkradach	Watcharapupong	Advisor
Asst. Thana	Hongsuwan	Advisor
Mr. Thananchai	Treepak	Advisor

Academic year 2006

ABSTRACT

Computer Security is one of the most important issue in computer systems, even though computer was installed security tools or not. They also have a chance to be attacked by an intruder. After the hacker attack the target system and gain root privilege, they usually install tools for hacking such as packet sniffing, backdoor access, key logger etc. and then install Rootkit which is the toolkit for hide files, process, security vulnerability and attacker's vestige, therefore user in these system cannot detect intrusion from hackers.

From the reason that mentioned above, studying Rootkit's behavior and operation is important to security in computer systems, consequently this project concerned with the study how attackers hide their vestige in UNIX systems and create Anti-Rootkit System which can analyze and detect Rootkit at the present. Furthermore these can help studying attacker's behavior and their vestige in the future.

กิตติกรรมประกาศ

โครงงานระบบต่อต้านชุดโปรแกรมปกปิดการบุกรุก มิอาจสำเร็จสมบูรณ์และลุล่วงไปได้ด้วยดี ถ้าขาดการให้โอกาส การดูแล และการให้คำแนะนำในการพัฒนาและจัดทำโครงการ การสั่งสอนที่เป็นอย่างดีเสมอมา จากท่านอาจารย์ที่ปรึกษา อาจารย์อัครเดช วัชระภูพงษ์, ผู้ช่วยศาสตราจารย์ธนา หงษ์สุวรรณ และอาจารย์ธนัญชัย ตรีภาค ซึ่งขอขอบพระคุณอาจารย์ทั้งสามท่านเป็นอย่างสูงขอขอบพระคุณคณาอาจารย์ทุกท่านในภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ซึ่งขอขอบพระคุณมา ณ ที่นี้อย่างสูง

ขอขอบคุณห้องวิจัยและพัฒนาการรักษาความปลอดภัยข้อมูล (ISAG) ที่เป็นแหล่งประสิทธิ์ประสาทวิชาให้ความรู้ความเข้าใจ และสนับสนุนสถานที่และอุปกรณ์เครือข่ายสำหรับการใช้ในการพัฒนาโครงงาน สุดท้ายขอกราบขอบพระคุณ บิดา มารดา และครอบครัวที่เป็นกำลังใจและให้การสนับสนุนในด้านต่างๆ ทำให้ผู้จัดทำสามารถทำวิทยานิพนธ์ฉบับนี้ลุล่วงด้วยดี ทางผู้จัดทำโครงการชุดโปรแกรมปกปิดการบุกรุก จึงขอขอบพระคุณมา ณ ที่นี้

นายกมล เลาหสุขไพศาล

นายจรูญชัย ธรรมพิพัฒน์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญรูป	VIII
บทที่ 1 บทนำ	
1.1 ความสำคัญและที่มาของโครงงาน	1
1.2 วัตถุประสงค์ของโครงงาน	1
1.3 ประโยชน์ที่คาดว่าจะได้รับ	1
1.4 ขอบเขตของโครงงาน	2
1.5 ขั้นตอนการดำเนินงาน	2
1.6 เนื้อหาของรายงาน	2
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	
2.1 มัลแวร์ (Malware)	3
2.2 ชุดโปรแกรมปกปิดการบุกรุก (Rootkit)	3
2.2.1 จุดเริ่มต้นของรูตคิต	3
2.2.2 พื้นฐานของรูตคิต	4
2.2.3 พฤติกรรมและฟังก์ชันการทำงานของรูตคิต	4
2.3 รูทคิตในโหมดยูสเซอร์ (User-Mode Rootkit)	5
2.3.1 การทำงานของ	5
2.3.2 การตรวจหา	8
2.4 รูตคิตในโหมดเคอร์นัล (Kernel-Mode Rootkit)	15
2.4.1 ลินุกซ์เคอร์นัล	15
2.4.2 วิธีการเปลี่ยนแปลงเคอร์นัลของรูตคิต	18
2.4.3 การตรวจหา	18
2.5 ตัวอย่างการทำงานของรูตคิตที่พบในปัจจุบัน	20
2.6 มาตรการป้องกันและตรวจหารูตคิต	22

สารบัญ (ต่อ)

หน้า

บทที่ 3 การออกแบบและพัฒนา

3.1 บทนำ 23

3.2 โครงสร้างของโครงงาน 23

3.2.1 เครื่องมือที่ใช้ในการพัฒนา 23

3.3 รายละเอียดโปรแกรมที่พัฒนา (Software Specification) 24

3.3.1 รายละเอียดส่วนนำเข้า 24

3.3.2 รายละเอียดส่วนแสดงผล 24

3.3.3 รายละเอียดฟังก์ชัน 24

3.3.4 การออกแบบ 24

3.3.5 ขอบเขตและข้อจำกัดของโปรแกรมที่พัฒนา 25

3.4 การออกแบบและพัฒนา 25

3.4.1 Rootkit-Scan (ส่วนตรวจหา) 25

3.4.2 Rootkit-Guard (ส่วนป้องกัน) 28

บทที่ 4 การทดลองและผลการทดลอง

4.1 บทนำ 31

4.2 การทดลองศึกษาพฤติกรรมของชุดโปรแกรมปกปิดการบุกรุก
กับลินุกซ์เคอร์นัลโมดูลที่ได้สร้างขึ้นเพื่อการทดลอง 31

4.3 การทดลองตรวจหาชุดโปรแกรมปกปิดการบุกรุก Tuxkit
ด้วยระบบต่อต้านชุดโปรแกรมปกปิดการบุกรุก (Loki) 32

4.4 การทดลองติดตั้งชุดโปรแกรมปกปิดการบุกรุกแบบเคอร์นัลโหมด
ตรวจสอบด้วยชุดโปรแกรมปกปิดการบุกรุก (Loki) 40

4.4.1 ทดสอบด้วยชุดโปรแกรมปกปิดการบุกรุก Adore-NG 40

4.4.2 ทดสอบด้วยชุดโปรแกรมปกปิดการบุกรุกตัวอย่างที่ได้สร้างขึ้น 43

บทที่ 5 บทวิจารณ์และสรุป

5.1 สรุปผลการพัฒนา 44

5.2 ปัญหาและอุปสรรค 44

5.3 แนวทางในการพัฒนาต่อ 45

สารบัญ (ต่อ)

	หน้า
5.4 ข้อสรุปและข้อเสนอแนะ	45
บรรณานุกรม	46
ภาคผนวก	47
ภาคผนวก ก. คู่มือการติดตั้งระบบต่อต้านชุดโปรแกรมปกปิดการบุกรุก	48
ภาคผนวก ข. คู่มือการใช้งานระบบต่อต้านชุดโปรแกรมปกปิดการบุกรุก	53

สารบัญตาราง

ตารางที่	หน้า
2.1 ชุดโปรแกรมปกปิดการบุกรุก และไดเร็กทอรีหรือไฟล์ที่ถูกสร้างขึ้น	9
2.2 การตรวจสอบหาไฟล์พื้นฐานของชุดโปรแกรมปกปิดการบุกรุกด้วยคำสั่ง find	12
2.3 การตรวจสอบหาไฟล์พื้นฐานของชุดโปรแกรมปกปิดการบุกรุกด้วยคำสั่ง grep	13
2.4 ไดเร็กทอรีที่มักถูกสร้างขึ้นโดยชุดโปรแกรมปกปิดการบุกรุก	13
2.5 การตรวจสอบคำสั่งในระบบปฏิบัติการลินุกซ์	14
3.1 คำสั่งที่มีการตรวจสอบโดย Rootkit-Scan	28

สารบัญรูป

รูปที่	หน้า
2.1 ความแตกต่างระหว่างมัลแวร์ชนิดม้าโทรจัน และ User-Mode Rootkit	6
2.2 การปกปิดการทำงานของผู้บุกรุกโดยใช้รูตคิต	7
2.3 โครงสร้างการทำงานของลินุกซ์เคอร์นัล	15
2.4 ลินุกซ์เคอร์นัลเปลี่ยนแปลงตารางซีสเต็มคอลเพื่อให้ไปทำงานโมดูลอื่นแทน	17
2.5 การทำงานของระบบปกติ	19
2.6 หลังจากติดตั้ง knark	19
3.1 การทำงานด้วย Silent Mode	26
3.2 ผลลัพธ์จากการแสกนด้วย Silent Mode	26
3.3 การทำงานด้วย Advanced Mode	27
3.4 แสดงการทำงานของ Rootkit – Guard	29
3.5 Rootkit-Guard ขณะยังไม่ทำการตรวจสอบ	29
3.6 Rootkit-Guard ไม่ตรวจพบสิ่งผิดปกติ	30
4.1 แสดงผลการทำงานของ Loadable kernel module ที่สร้างขึ้น	31
4.2 การติดตั้ง tuxkit	32
4.3 ติดตั้ง tuxkit สมบูรณ์	32
4.4 แสดงรายชื่อไฟล์ก่อนทำการติดตั้ง tuxkit	33
4.5 แสดงรายชื่อไฟล์หลังทำการติดตั้ง tuxkit	33
4.6 เปิดโปรแกรมด้วยคำสั่ง. /Anti-Rootkit	34
4.7 หน้าต่างหลักของโปรแกรม	34
4.8 ขณะทำการสแกน	35
4.9 มีการแจ้งเตือนพบชุดโปรแกรมปกปิดการบุกรุก	35
4.10 แสดงคำสั่งที่ถูกเปลี่ยนแปลง	36
4.11 แสดงไดเร็กทอรีที่เสี่ยงต่อการถูกสร้างโดยรูทคิต	37
4.12 วิเคราะห์ผลลัพธ์ว่าเป็น Tuxkit หรือ AjaKit	37
4.13 ผลลัพธ์จากการสแกนด้วย Silent Mode	38
4.14 ผลลัพธ์จากการสแกนด้วย Silent Mode	38
4.15 Rootkit-Guard ไม่พบความผิดปกติใดๆ	39
4.16 แสดงล็อกไฟล์ของ Rootkit-Scan	40

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.17 แสดงเมนูการทำงานของโปรแกรมปกปิดการบุกรุก Adore – NG	41
4.18 แสดงหน้าต่างหลักของโปรแกรมในส่วน Rootkit Guard	41
4.19 ผลลัพธ์ที่ได้ก่อนการติดตั้ง Adore – NG	42
4.20 ผลลัพธ์ที่ได้หลังการติดตั้ง Adore – NG	42
4.21 แสดงการเปลี่ยน sys_getdents64() ในตารางซีสเต็มคอล	43

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของโครงงาน

การรักษาความปลอดภัยในเครื่องคอมพิวเตอร์เป็นสิ่งที่สำคัญอย่างมาก แต่ถึงแม้ว่าเครื่องคอมพิวเตอร์นั้นจะติดตั้งชุดโปรแกรมรักษาความปลอดภัยเท่าใด ก็มีโอกาสที่จะถูกบุกรุกโดยผู้ที่ไม่ประสงค์ดีได้เสมอ และเมื่อเครื่องคอมพิวเตอร์นั้นถูกบุกรุก ผู้บุกรุกเองมักจะติดตั้งชุดโปรแกรมปกปิดการบุกรุก (Rootkit) เพื่อทำการซ่อนหรือปกปิดไฟล์ โปรเซส หรือช่องโหว่ความปลอดภัยที่ผู้บุกรุกได้สร้างขึ้นในเครื่องคอมพิวเตอร์นั้น ทำให้ผู้ถูกบุกรุกไม่รับรู้ถึงการโจมตีนั้นๆ และนั่นจึงเป็นเหตุผลที่ว่า การตรวจสอบว่าเครื่องคอมพิวเตอร์นั้นถูกบุกรุกยากขึ้นทุกที หรือบางทีอาจกลายเป็นเรื่องที่ไปไม่ได้เลย จากสถิติเราพบว่าชุดโปรแกรมปกปิดการบุกรุกมีการแพร่หลายขึ้นมาก ซึ่งมีการประกาศโดยบรรดาผู้ผลิตซอฟท์แวร์ป้องกันไวรัส โดยมีอัตราการเติบโตถึง 700% ในช่วง 2 – 3 ปี ที่ผ่านมา

ด้วยเหตุผลข้างต้น การศึกษาพฤติกรรมของชุดโปรแกรมปกปิดการบุกรุกจึงมีความสำคัญ ทั้งนี้เมื่อสามารถเรียนรู้วิธีการทำงานและพฤติกรรมของชุดโปรแกรมปกปิดการบุกรุก ก็จะเป็นผลที่ทำให้สามารถสร้างโปรแกรมต้นแบบขึ้นมาเพื่อต่อต้าน ใช้เป็นเครื่องมือหนึ่งที่ใช้ช่วยในการวิเคราะห์ หรือค้นหาซอฟแวร์ไม่พึงประสงค์ในเครื่อง และที่สำคัญจะทำให้ระบบคอมพิวเตอร์นั้นๆ มีความปลอดภัยมากยิ่งขึ้น

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อศึกษาวิธีปกปิดพฤติกรรมของผู้บุกรุกระบบคอมพิวเตอร์

1.2.2 เพื่อศึกษากระบวนการทำงานของชุดโปรแกรมปกปิดการบุกรุก

1.2.3 เพื่อศึกษาวิธีการตรวจสอบชุดโปรแกรมปกปิดการบุกรุก

1.2.4 เพื่อสร้างต้นแบบระบบต่อต้านชุดโปรแกรมปกปิดการบุกรุกระบบคอมพิวเตอร์

1.3 ประโยชน์ที่คาดว่าจะได้รับ

1.3.1 ได้รับความรู้ความเข้าใจเกี่ยวกับวิธีปกปิดพฤติกรรมของผู้บุกรุกระบบคอมพิวเตอร์

1.3.2 ได้รับความรู้ความเข้าใจเกี่ยวกับกระบวนการทำงานของชุดโปรแกรมปกปิดการบุกรุก

1.3.3 ได้รับความรู้ความเข้าใจเกี่ยวกับการเขียนโปรแกรมแบบเคอร์นัลโมดูล

1.3.4 ได้รับความรู้ความเข้าใจเกี่ยวกับการตรวจจับชุดโปรแกรมปกปิดการบุกรุก

1.3.5 สามารถสร้างระบบต่อต้านชุดโปรแกรมปกปิดการบุกรุก และตรวจพบชุดโปรแกรมปกปิดการบุกรุกในปัจจุบันเพื่อนำไปใช้งานจริงได้

1.4 ขอบเขตของโครงงาน

1.4.1 ระบบสามารถตรวจสอบชุดโปรแกรมปกปิดการบุกรุกที่ติดตั้งอยู่ภายในเครื่องคอมพิวเตอร์ที่ติดตั้งระบบปฏิบัติการแพลตฟอร์มยูนิกซ์ได้

1.4.2 ระบบสามารถทำงานได้ทั้งในโหมดกราฟฟิกและโหมดคอมมานไลน์

1.4.3 ระบบสามารถทำงานด้วยตนเองเมื่อระบบปฏิบัติการแพลตฟอร์มยูนิกซ์เริ่มต้นทำงาน เพื่อตรวจสอบรูตคิต

1.4.4 ระบบสามารถตรวจสอบคำสั่งในระบบปฏิบัติการที่ถูกชุดโปรแกรมปกปิดการบุกรุกเปลี่ยนแปลงได้

1.5 ขั้นตอนการดำเนินงาน

1.5.1 ศึกษาความรู้พื้นฐานเกี่ยวกับมัลแวร์

1.5.2 ศึกษากระบวนการทำงานของชุดโปรแกรมปกปิดการบุกรุก

1.5.3 ทดลองสร้างชุดโปรแกรมปกปิดการบุกรุก

1.5.4 วิเคราะห์แนวทางในการค้นหาชุดโปรแกรมปกปิดการบุกรุก

1.5.5 สร้างระบบต่อต้านชุดโปรแกรมปกปิดการบุกรุก

1.5.6 ทดลองติดตั้งชุดโปรแกรมปกปิดการบุกรุก และทดสอบด้วยระบบต่อต้านชุดโปรแกรมปกปิดการบุกรุก

1.5.7 ทำเอกสารโครงงานและสรุปผล

1.6 เนื้อหาของรายงาน

1.6.1 บทที่หนึ่งเป็นบทนำซึ่งจะกล่าวถึงความเป็นมาของโครงงาน, วัตถุประสงค์, ประโยชน์ที่คาดว่าจะได้รับ, ขอบเขตของโครงงาน และขั้นตอนการดำเนินงาน

1.6.2 บทที่สองจะเป็นการอธิบายถึงทฤษฎีที่เกี่ยวข้อง

1.6.3 บทที่สามจะเป็นการอธิบายถึงการออกแบบและพัฒนาระบบ

1.6.4 บทที่สี่เป็นการทดลองและผลการทดลองของโครงงาน

1.6.5 บทที่ห้าจะเป็นการสรุปโครงงานและประยุกต์ใช้ร่วมกับงานอื่นๆ ในขั้นต่อไป

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 มัลแวร์ (Malware)

มัลแวร์ (Malware) หรือ ซอฟต์แวร์ไม่พึงประสงค์ทั้งหลายที่มีอยู่ ไม่ว่าจะเป็นไวรัส โค้ดหนอน และสปายแวร์เป็นสิ่งที่ ระบาดกันมากในช่วงหลายปีที่ผ่านมา ตัวเลขทางสถิติได้บ่งชี้ว่า เครื่องคอมพิวเตอร์ที่ใช้วินโดวส์หรือระบบปฏิบัติการอื่นๆ โดยไม่ได้มีการติดตั้งแพตช์ใดๆ เพิ่มเติมจะถูกโจมตีในไม่กี่นาทีที่ต่อเชื่อมเข้าอินเทอร์เน็ตแล้วทำการโหลดซอฟต์แวร์เข้าเครื่อง หรือเพียงเข้าเว็บไซต์บางแห่ง สามารถทำให้ผู้ใช้กลายเป็นเหยื่อของซอฟต์แวร์ที่ไม่พึงประสงค์เหล่านี้ได้ ทว่ามาตรการป้องกันการโรคระบาดเหล่านี้ก็ได้รับการปรับปรุงให้ดีขึ้นเรื่อยๆ ทั้งจากโปรแกรมแอนตี้ไวรัส แอนตี้สปายแวร์ ไฟร์วอลล์ และแพตช์ต่างๆ อย่างไรก็ตามเทคโนโลยีที่เรียกว่าชุดโปรแกรมปกปิดการบุกรุก หรือรูตคิต (Rootkit) ซึ่งคุกคามการรักษาความปลอดภัย และทำให้งานในการตรวจสอบว่าเครื่องคอมพิวเตอร์ปราศจากซอฟต์แวร์ไม่พึงประสงค์หรือไม่นั้นยากขึ้น หรือกลายเป็นเรื่องที่เป็นไปไม่ได้เลย

2.2 ชุดโปรแกรมปกปิดการบุกรุก (Rootkit)

2.2.1 จุดเริ่มต้นของรูตคิต

รูตคิตเริ่มปรากฏให้เห็นในช่วงต้นของทศวรรษ 1990 แต่จำกัดอยู่เฉพาะในวงของผู้ใช้ระบบยูนิกซ์แบบต่างๆ เท่านั้น จนกระทั่งช่วงปลายทศวรรษ 1990 เมื่อกลุ่มนักพัฒนาวินโดวส์เริ่มศึกษาเทคนิคของรูตคิต และโปรแกรมเมอร์หลายๆ คนเริ่มพัฒนาทูลคิต (toolkit) หรือชุดเครื่องมือสำหรับรูตคิตออกมาเพื่อให้โปรแกรมเมอร์อื่นๆ สามารถ ที่แก้ไขและต่อยอดได้ ซึ่งทูลคิตบางตัวนั้นทำไว้ดีมากเสียจนผู้ผลิตมัลแวร์สามารถนำข้อได้เปรียบของการอำพรางตัวของ รูตคิตไปใช้งานได้แทบจะทันที แค่เพียงแก้ไขไฟล์คอนฟิกูเรชันเพียงเล็กน้อยและรวมเข้ากับมัลแวร์ก็เป็นอันเรียบร้อย

อย่างไรก็ตามในช่วงที่ผ่านมารูตคิตได้รับความสนใจค่อนข้างมาก เพราะการประชาสัมพันธ์ของสื่อที่เตือนให้ผู้ใช้ทราบ ถึงอันตรายของรูตคิต อย่าง ความแพร่หลายนี้ ก็กลายเป็นดาบสองคมเพราะในขณะเดียว กันก็ได้ประกาศถึงประสิทธิภาพของรูตคิตให้กับสังคมของผู้เขียนมัลแวร์ให้ทราบ และบรรดาผู้เขียนมัลแวร์ก็จะเริ่มใช้รูตคิตในการทำสงครามกับโปรแกรมแอนตี้ไวรัส และแอนตี้สปายแวร์ที่นับวันจะมีประสิทธิภาพมากขึ้นเรื่อยๆ ซึ่งในอนาคตอันใกล้นี้ก็คงจะมีไวรัส สปายแวร์ และแอดแวร์ที่ไม่สามารถที่จะลบออกจากระบบได้โดยไม่ต้อง อาศัยการฟอร์แมตฮาร์ดดิสก์แล้วติดตั้งระบบปฏิบัติการใหม่ทั้งหมด

รูตคิตนั้นบุคคลทั่วไปจะรู้จักตั้งแต่มีการตรวจพบว่าซีดีเพลงของค่ายหนึ่ง มีการแอบฝังโปรแกรมลงไปในแผ่นซีดีเพื่อที่จะป้องกันการคัดลอกแผ่น เหตุการณ์ดังกล่าวทำให้ชื่อของ รูตคิต ดัง

ขึ้นมาทันที เพราะก่อนหน้านี้ชื่อรูตคิตจะรู้จักกันในวงการผู้ที่สนใจการเจาะระบบและการแคร็กระบบเท่านั้น

2.2.2 พื้นฐานของรูตคิต

รูตคิต คือชุดโปรแกรมที่ใช้เป็นเครื่องมือที่ถูกนำมาใช้ หลังจากที่เราสามารถเข้าสู่ในระบบคอมพิวเตอร์ที่ต้องการได้แล้ว โปรแกรมพวกนี้จะพยายามหลบซ่อนทั้งโพรเซสทั้งไฟล์ ทั้งข้อมูลต่างๆ ที่จำเป็นต้องใช้ ให้หลุดรอดจากตรวจจับของเจ้าของระบบ เพื่อที่จะทำให้ตัวเองนั้นยังสามารถซ่อนอยู่ในระบบนั้นได้ต่อไป

คำว่า “รูตคิต” หรือ "Rootkit" นั้นหมายถึง เครื่องมือในยูนิกซ์ที่ถูก Recompiled แล้ว ซึ่งคำสั่งต่างๆ ที่ใช้ในการตรวจจับการถูกรุกราน เช่น "ps", "netstat", "w" และ "passwd" จะถูกแก้ไขเพื่อที่จะปกปิดร่องรอยของผู้รุกราน ทำให้ผู้รุกรานนั้นยังคงสถานะของความเป็น "root" อยู่ได้โดยที่ Administrator ของระบบยังไม่รู้ตัว เมื่อมัลแวร์ใช้ ความสามารถของรูตคิตแล้ว ก็จะสามารถซ่อนตัวเองไม่ให้ระบบรักษาความปลอดภัย รวมถึงโปรแกรมแอนตี้ไวรัสและ Task Manager ค้นพบ

2.2.3 พฤติกรรมและฟังก์ชันการทำงานของรูตคิต

การทำงานของรูตคิตโดยทั่วไปแล้วคือ การปกปิด User Login ที่ใช้ในการเข้าสู่ระบบ, โพรเซส, ไฟล์, Log, โปรแกรมที่ใช้ในการดักจับข้อมูล และการต่อเชื่อมกับระบบเน็ตเวิร์ก ซึ่งในรูตคิตหลายต่อหลายตัว นั้นถูกจัดให้อยู่ในพวกเดียวกับม้าโทรจันด้วย เนื่องด้วยการทำงานที่คล้ายกันมาก

พฤติกรรมและฟังก์ชันการทำงานของรูตคิตแบ่งได้เป็น 3 หัวข้อใหญ่ ดังนี้

2.2.3.1 ฟังก์ชันการเข้าถึงระบบ

- Backdoor ผู้บุกรุกจะวางช่องโหว่ Backdoor ไว้โดยผ่านโปรโตคอล Telnet, SSH, RSH, IRC หรืออื่นๆ โดยอาจใช้การสื่อสารแพ็คเกตจำพวก TCP/UDP/ICMP โดยทำการดักจับข้อมูล หรือครอบครองระบบเน็ตเวิร์ค
- Outbound Connection อาจทำการซ่อนตัวหลัง Firewall ของระบบ หรือทำอุโมงค์เน็ตเวิร์คผ่านพอร์ต 80 เป็นต้น

2.2.3.2 ฟังก์ชันการโจมตีระบบท้องถิ่น หรือระบบอื่นๆ

- ทำการเก็บข้อมูลของระบบเครือข่าย
- ติดตั้งโปรแกรมดักจับข้อมูลเครือข่าย
- พยายามแพร่กระจายตัวเองออกไป

2.2.3.3 ทำลายหลักฐานทั้งหมดที่ได้กระทำไป ซึ่งฟังก์ชันนี้เป็นสิ่งสำคัญและถือเป็นหัวใจหลักของรูตคิตที่ต้องทำ

- ลบรายละเอียดล็อกไฟล์ทั้งหมดที่ได้กระทำไป

- ซ่อนไฟล์
- ซ่อนโพรเซส
- ซ่อนการเชื่อมต่อเน็ตเวิร์ค
- ซ่อนการล็อกอินเข้าสู่ระบบ

รูตคิตในยุคแรกๆ ใช้วิธีการที่ไม่ซับซ้อนในการแทนที่คำสั่งหลักของระบบด้วยเวอร์ชันที่ซ่อนมัลแวร์และโพรเซสต่างๆไว้ เช่น รูตคิตในเวอร์ชันของยูนิกซ์นั้น โดยปกติยูนิกซ์จะใช้ยูทิลิตี้ ps ในการแสดงรายโพรเซสที่แอ็คทีฟทั้งหมดได้นั้น แต่รูตคิตจะทำให้งดการแสดงรายการโพรเซสของมัลแวร์ที่ใช้อยู่ในเคอร์นัล และเช่นเดียว กันสำหรับยูทิลิตี้ ls ซึ่งใช้แสดงรายการของชื่อไฟล์ในไดเร็กทอรี ก็จะไม่แสดงรายการมัลแวร์ในไฟล์ในไดเร็กทอรี

เมื่อยูทิลิตี้ของระบบเริ่มมีความซับซ้อนมากขึ้น พร้อมทั้งบรรดาแอนตี้ไวรัสทั้งหลายเริ่มแพร่หลายมากขึ้น การใช้เทคนิคการแทนที่ไฟล์เดิมของรูตคิตก็ใช้ไม่ได้อีกต่อไป เพราะการเขียนเพื่อแทนที่ Task Manager และทูลในการแสดงรายการโพรเซสต่างๆ และจะเป็นความพยายามที่สูญเปล่าทันทีถ้าผู้ใช้เรียกโปรแกรมแอนตี้ไวรัส หรือใช้ทูลตัวอื่นในการแสดงรายการโพรเซสทั้งหมด

ผู้พัฒนารูตคิตก็ได้เพิ่มความซับซ้อนมากขึ้นเช่นเดียวกัน โดยแทนที่จะโจมตีที่แอพพลิเคชันตรงๆ ก็เลี่ยงมาโจมตีที่ส่วนของเคอร์นัล และตารางซีสเต็มคอล (System Call Table) ที่แอพพลิเคชันต่างๆ เรียกใช้เพื่อเรียกเก็บข้อมูล ด้วยการดักจับ API เวลาที่แอพพลิเคชันเรียกรายการของโพรเซสที่แอคทีฟแล้วทำการลบโพรเซสของมัลแวร์ออกจากรายการที่จะต้องส่งกลับให้แอพพลิเคชันนั้นๆ ตัวรูตคิตเองก็สามารถซ่อนเร้นตัวเองจากการตรวจจับของ Task Manager และยูทิลิตี้อื่นๆ ที่ใช้ API เหล่านี้ได้ทั้งทางตรงและทางอ้อม ทั้งนี้รูตคิตสมัยใหม่จะใช้เทคนิคนี้ในการซ่อนไฟล์, ไดเร็กทอรี, พอร์ต TCP/IP, ยูสเซอร์แอคเคาต์ และโพรเซสต่างๆ ไม่ให้เห็น

รูตคิตที่ทรงพลังมากที่สุดคือ รูตคิตในโหมดเคอร์นัล อย่างไรก็ตาม รูตคิตในโหมดเคอร์นัลทั้งนี้ ผู้เขียนรูตคิตที่อิมพลีเม็นต์ในโหมดเคอร์นัลนี้จะต้องมีความรู้ที่ลึกซึ้งมาก และต้องเขียนด้วยความระมัดระวังอย่างยิ่ง เนื่องจากระบบปฏิบัติการจะล่มได้ด้วยข้อผิดพลาดเพียงตัวเดียว ซึ่งเป็นสิ่งที่ผู้เขียนรูตคิตต่างก็ไม่ต้องการ เพราะการทำให้แอพพลิเคชันหรือระบบปฏิบัติการล่มจะเป็นการกระทำที่ก่อให้เป็นจุดสนใจ

2.3 รูตคิตในโหมดยูสเซอร์ (User-Mode Rootkit)

2.3.1 การทำงานของรูตคิตในโหมดยูสเซอร์

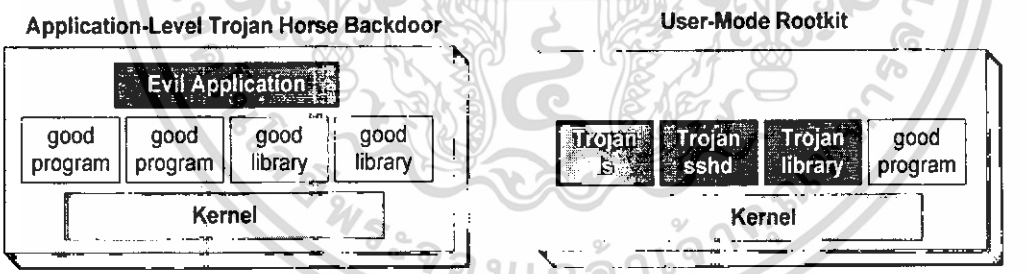
การทำงานของรูตคิตนั้น มีการทำงานซึ่งคล้ายคลึงกับม้าโทรจัน (Trojan Horse) และ Backdoor ซึ่งจากการที่ม้าโทรจันนั้นมีการแก้ไขคำสั่งภายในระบบปฏิบัติการด้วยคำสั่งที่ประสงค์ร้ายต่อระบบ โดยที่คำสั่งที่ถูกแทนที่นั้นมีการทำงานที่ดูเหมือนคำสั่งปกติ แต่จริงๆ แล้วถูกปกคลุมอยู่ด้วยคำสั่งที่มี

ประโยชน์ต่อผู้บุกรุก ตัวอย่างเช่น ในระบบปฏิบัติการประเภทยูนิกซ์ Unix ผู้บุกรุกแก้ไขคำสั่ง ls ซึ่งคำสั่ง ls นั้นเป็นคำสั่งที่ใช้สำหรับการแสดงรายชื่อของข้อมูลต่างๆ ที่อยู่ในไดเรกทอรี่ ซึ่งในเวอร์ชั่นของรูตคิตนั้นจะไม่แสดงรายชื่อไฟล์ของผู้บุกรุกซึ่งในที่นี้คำสั่ง ls ของรูตคิตนั้นทำงานเช่นเดียวกันกับม้าโทรจัน

นอกเหนือจากการทำงานที่คล้ายคลึงกับม้าโทรจันแล้ว รูตคิตยังมีการทำงานที่เป็น backdoor อีกด้วยรูตคิตหลายๆ ตัวทำให้ผู้บุกรุกสามารถกลับเข้ามาบุกรุกยังระบบอีกได้ด้วย backdoor password ซึ่งในการใช้รูตคิตเพื่อเปลี่ยนแปลงหลากหลายคำสั่งในระบบและสามารถควบคุมเครื่องของเหยื่อผ่านทางเน็ตเวิร์คได้อีกด้วย

อีกสิ่งที่สำคัญคือ รูตคิตไม่สามารถทำให้ผู้บุกรุกได้สิทธิของ root หรือผู้ดูแลระบบในครั้งแรกที่บุกรุกโดยทันที แต่ผู้บุกรุกจะต้องหาทางมาซึ่งให้ได้สิทธิของผู้ดูแลระบบ ไม่ว่าจะเป็นการโจมตีแบบการล้นของบัฟเฟอร์ (Buffer Overflow) หรือการทำนายพาสเวิร์ด โดยใช้วิธี Dictionary Attack หรือวิธีอื่นๆ ซึ่งเมื่อผู้บุกรุกเข้าถึงได้สำเร็จ ผู้บุกรุกจะทำการติดตั้งรูตคิต และปรับตั้งค่า ซึ่งจะทำให้ผู้บุกรุกออกจากระบบและกลับมาบุกรุกระบบได้โดยที่ผู้ดูแลระบบไม่สามารถเห็นร่องรอยได้

ข้อแตกต่างระหว่างรูตคิตในโหมดยูสเซอร์กับมัลแวร์ชนิดม้าโทรจันและ backdoor คือ จะเป็นการเขียนโปรแกรมที่ทำงานอยู่ในชั้นแอพพลิเคชั่น ซึ่งจะไม่มีการไปแก้ไขคำสั่ง, ไลบรารี่ และเคอร์นัลในระบบปฏิบัติการ แต่รูตคิตในโหมดนั้น ผู้บุกรุกจะเขียนคำสั่งเพื่อไปแทนที่คำสั่งสำคัญๆ ในระบบปฏิบัติการ เช่น ls, sshd เป็นต้น

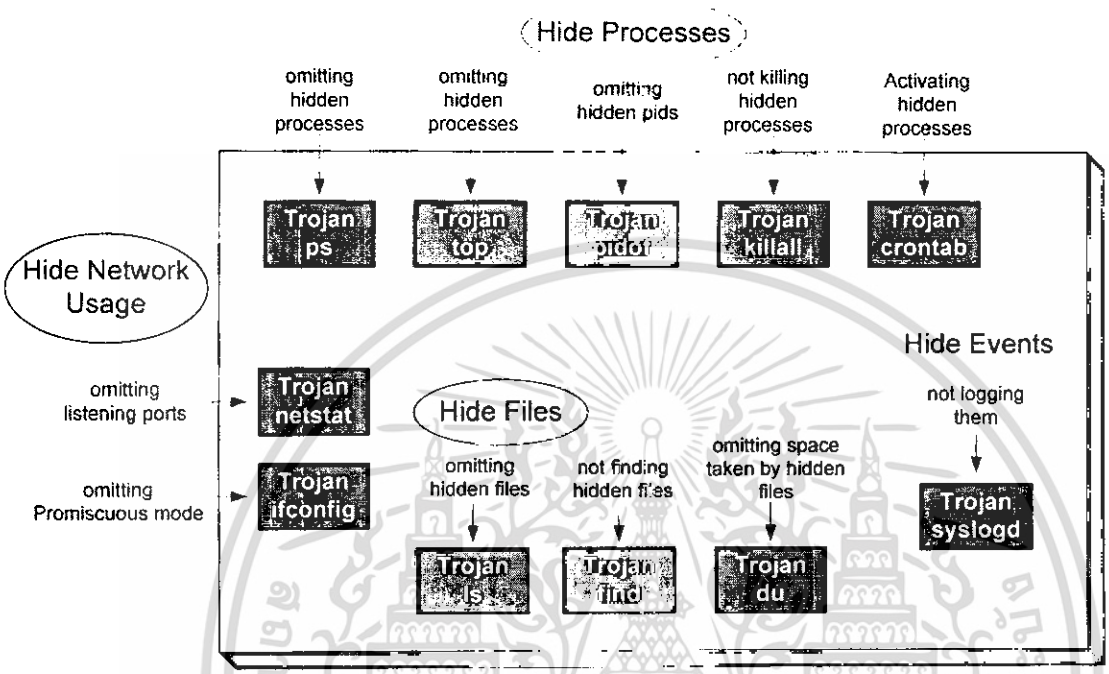


รูปที่ 2.1 ความแตกต่างระหว่างมัลแวร์ชนิดม้าโทรจัน และ User-Mode Rootkit

รูตคิตในปัจจุบันนั้นได้มีการพัฒนาความสามารถให้มีประสิทธิภาพมากกว่าในช่วงแรกๆ ที่มีการพัฒนาขึ้นมา และมีเครื่องมีต่างๆ ที่อยู่รวมในรูตคิต ซึ่งสามารถแบ่งออกเป็น 5 ประเภทหลัก ได้แก่

การแทนที่คำสั่งและทำให้ผู้บุกรุกสามารถทำ backdoor access ได้ เครื่องมือนี้ถือเป็นสิ่งที่สำคัญที่สุดของรูตคิตในโหมดยูสเซอร์ ซึ่งทำได้โดยการเขียนทับคำสั่งและเซอร์วิสต่างๆ ที่ใช้ในการเข้าสู่ระบบ ดังนั้นผู้บุกรุกสามารถเข้าสู่ระบบและได้สิทธิของ root ในระบบที่เป็นเป้าหมาย

การแทนที่คำสั่งและปกปิดผู้บุกรุก เครื่องมือนี้จะทำการเขียนทับคำสั่งในระบบด้วยคำสั่งที่เป็นม้าโทรจันที่ทำให้ไม่สามารถพบผู้บุกรุก ซึ่งคำสั่งใหม่นี้จะทำการให้ข้อมูลเท็จแก่ผู้ดูแลระบบเกี่ยวกับข้อมูล, โพรเซส, และการใช้งานทางเน็ตเวิร์ค



รูปที่ 2.2 การปกปิดการทำงานของผู้บุกรุกโดยใช้ Rootkit

เครื่องมือที่ใช้ในการปกปิดแต่ไม่เขียนทับคำสั่ง โปรแกรมชนิดนี้จะทำการแก้ไขเวลาในการใช้งานล่าสุดของไฟล์ หรือทำการแก้ไขล็อกไฟล์ ซึ่งเกิดขึ้นจากการติดตั้งชุดโปรแกรมปกปิดการบุกรุกเพื่อที่ทำลายหลักฐานของผู้บุกรุก

เครื่องมืออื่นๆ ซึ่งแตกต่างกันไป รูตคิตในระบบปฏิบัติการยูนิกซ์นั้นมีหลากหลายจุดประสงค์ออกไป บางตัวนั้นมีการดักจับข้อมูลที่ผ่านเข้าออกในระบบของเหยื่อ หรืออาจจะมี backdoor shell listener

สคริปต์ที่ใช้ในการติดตั้ง โปรแกรมชนิดนี้จะใช้ในการเปิดเครื่องมืออื่นๆ คอมไพล์ หรือติดตั้งลงในตำแหน่งที่เหมาะสม แทนที่จะต้องเขียนทับคำสั่งแต่ละคำสั่ง เครื่องมือนี้จะช่วยให้ผู้บุกรุกสามารถติดตั้งชุดโปรแกรมปกปิดการบุกรุกได้ในเวลาเพียงไม่นานเท่านั้น หลังจากที่ติดตั้งไฟล์ต่างๆ เรียบร้อยแล้ว สคริปต์นี้ยังสามารถแก้ไขค่าการปรับปรุงไฟล์ครั้งล่าสุด หรือแก้ไขขนาดของคำสั่งให้มีค่าเท่ากับคำสั่งดั้งเดิมได้อีกด้วย

2.3.2 การตรวจพบ User-Mode Rootkit

ในการตรวจพบรูตคิตในโหมดยูสเซอร์นั้น จำเป็นต้องใช้โปรแกรมประเภท File integrity checkers โดยการทำงานของโปรแกรมประเภทนี้คือ เมื่อทำการติดตั้ง โปรแกรมจะสร้างฐานข้อมูลของ cryptographic hashes ของไฟล์คำสั่งที่สำคัญต่างๆ ของระบบปฏิบัติการ ซึ่ง hashes นี้จะทำงานเหมือนกับลายนิ้วมือเพื่อใช้ในการตรวจสอบความถูกต้องของระบบ และฐานข้อมูลที่ได้เข้ารหัสเหล่านี้ควรจะเก็บอยู่ในแหล่งข้อมูลที่มีการป้องกันการเขียนทับ เช่น ซีดีรอม หรือแผ่นดิสก์ โดยที่โปรแกรมเหล่านี้จะใช้การเข้ารหัส เช่น MD5 หรือ SHA-1 เป็นต้น ซึ่งการเข้ารหัสในอัลกอลิทึ่มที่แข็งแกร่งนั้นมีความสำคัญอย่างมากในการตรวจสอบ

หลังจากที่ได้สร้างและเก็บฐานข้อมูลที่ได้เข้ารหัสของไฟล์ซิสเต็มที่สำคัญแล้ว ผู้ดูแลระบบควรจะทำการตรวจสอบเปรียบเทียบความบริบูรณ์ของไฟล์ซิสเต็มอยู่เสมอ เช่น หนึ่งครั้งต่อวัน หรือมากกว่าหนึ่งครั้งในหนึ่งชั่วโมง ทั้งนี้ขึ้นอยู่กับความสำคัญของระบบ

เครื่องมือที่ใช้ในการตรวจสอบความบริบูรณ์ของไฟล์ซิสเต็มนั้นมักจะทำการตรวจสอบคำสั่งที่มักจะถูกเปลี่ยนแปลงโดยผู้บุกรุก เช่น sshd, login, netstat, ps, ls เป็นต้น

2.3.2.1 การตรวจพบ User-Mode Rootkit ด้วย Signature Based

ในการตรวจพบชุดโปรแกรมปกปิดการบุกรุกในโหมดของยูสเซอร์นั้น จะทำการค้นหาไฟล์ข้อมูลต่างๆ ที่มีโอกาสเป็นชุดโปรแกรมปกปิดการบุกรุก และนำมาเปรียบเทียบกับโปรแกรมที่ได้รวบรวมข้อมูลพื้นฐานของรูตคิตชนิดต่างๆ ซึ่งการคัดกรองข้อความหรือผลลัพธ์เพื่อให้ได้ข้อมูลที่เราต้องการนั้น การค้นหาคำที่ต้องการในลินุกซ์หรือยูนิกซ์ สามารถใช้คำสั่ง grep ได้

ซึ่งมีการใช้งานเบื้องต้น ตามตัวอย่างดังนี้

- ค้นหาคำว่า root ในทุกไฟล์ใน /etc

```
# grep root /etc/*
```

- ค้นหาคำว่า tcp ตัวพิมพ์ใหญ่หรือเล็กก็ได้ใน /etc ให้เพิ่มพารามิเตอร์ -i คือ ignore case

```
# grep -i tcp /etc/*
```

- แสดงข้อมูลในแฟ้ม /etc/passwd แต่เลือกเฉพาะบรรทัดที่มีคำว่า adore

```
# more /etc/passwd|grep adore
```

- ดูระบบว่าถูก reboot เมื่อใดบ้าง

```
# last |grep reboot
```

หากต้องการให้แสดงผลเฉพาะชื่อไฟล์ที่พบข้อความ ให้เพิ่ม -l

```
# grep -il tcp /etc/*
```

เป็นต้นว่า การค้นหาเหตุการณ์ใน Log file เป็นต้น

โดยปกติมักจะใช้คำสั่ง grep เป็นหลัก ซึ่งจะใช้เลือกผลลัพธ์ที่มีความสอดคล้องกับที่ระบุไว้ขึ้นมาแต่ถ้าเป็นทางเลือกมากกว่า 1 ตัว การใช้ egrep จะเหมาะสมกว่า เช่น

egrep -i (error|fail) /var/log/messages

จะเป็นการหาทั้งบรรทัดที่มีคำว่า error และ fail

นอกจากนี้ยังมีการใช้คำสั่ง find ในการค้นหาไฟล์พื้นฐานของชุดโปรแกรมปกปิดการบุกรุก ซึ่งคำสั่ง find เป็นคำสั่งที่ใช้สำหรับค้นหาแฟ้มข้อมูล และมีโครงสร้างของคำสั่ง ดังนี้

find [path].. expression

ลักษณะของ expression เช่น

-name [pattern] เพื่อใช้หาชื่อ file ตาม pattern ที่ระบุ

-perm [+-] mode เพื่อใช้หา file ตาม mode ที่ต้องการ

-user NAME หา file ที่เป็นของ user ชื่อ NAME

-group NAME หา file ที่เป็นของ group ชื่อ NAME

ตัวอย่างการใช้คำสั่ง find

find -name *.doc

find /usr -perm +111 (หาแฟ้มที่มี Permission อย่างน้อยเป็น 111)

ดังนั้นการตรวจพบชุดโปรแกรมปกปิดการบุกรุกด้วย Rootkit-Scan จึงต้องทำการรวบรวมข้อมูลพื้นฐานของชุดโปรแกรมปกปิดการบุกรุกแต่ละตัวไว้ เช่น ไฟล์ที่ชุดโปรแกรมปกปิดการบุกรุกได้สร้างไว้ในระบบปฏิบัติการ แต่การใช้คำสั่ง find ในระบบปฏิบัติการนั้น อาจจะไม่สามารถตรวจสอบได้ ดังนั้นจึงใช้คำสั่ง grep ในการตรวจสอบ ซึ่งรวบรวมข้อมูลของชุดโปรแกรมปกปิดการบุกรุก และไดเร็กทอรีที่มีไฟล์พื้นฐานของชุดโปรแกรมบุกรุกชนิดนั้นๆ โดยข้อมูลที่ใช้รวบรวมในการตรวจสอบชุดโปรแกรมปกปิดการบุกรุก มีดังต่อไปนี้

การค้นหาชุดโปรแกรมปกปิดการบุกรุกโดยตรวจสอบจากไดเร็กทอรีที่มักจะมีการเก็บไฟล์ของชุดโปรแกรมปกปิดการบุกรุก หรือถูกชุดโปรแกรมปกปิดการบุกรุกสร้างขึ้นในระบบ มีดังตารางที่ 2.1

ชุดโปรแกรมปกปิดการบุกรุก	ไดเร็กทอรีที่เก็บไฟล์พื้นฐาน
adore LKM	/proc/ksyms
sebek LKM (Adore based)	/proc/ksyms
knark LKM	/proc/knark
HiDrootkit	/var/lib/games/.k
t0rn	/usr/src/.puta /lib/ldlib.tk /usr/info/.t0rn
Lion Worm	/usr/info/.torn

ชุดโปรแกรมปกปิดการบุกรุก	ไดเร็กทอรีที่เก็บไฟล์พื้นฐาน
Lion Worm	/bin/in.telnetd /bin/mjy
RSHA rootkit	/bin/kr4p /usr/bin/n3tstat /usr/bin/chsh2 /usr/bin/slice2 /usr/src/linux/arch/alpha/lib/.lib/.1proc /etc/rc.d/arch/alpha/lib/.lib/.1addr /etc/rc.d/rsha /etc/rc.d/arch/alpha/lib/.lib
RH-Sharpe rootkit	/bin/lps /usr/bin/lpstree /usr/bin/ltop /usr/bin/lkillall /usr/bin/ldu /usr/bin/lnetstat /usr/bin/wp /usr/bin/shad /usr/bin/vadim /usr/bin/slice /usr/bin/cleaner /usr/include/rpcsvc/du
Ambient's rootkit (ark)	/dev/ptyxx /usr/lib/.ark? /usr/doc/
LPD Worm	/dev/.kork /bin/.ps /bin/.login
Ramem Worm	/usr/src/.poop /tmp/ramen.tgz /etc/xinetd.d/asp

ชุดโปรแกรมปกปิดการบุกรุก	ไดเร็กทอรีที่เก็บไฟล์พื้นฐาน
Showtee	/usr/lib/.egcs /usr/lib/.kinetic /usr/lib/.wormie /usr/lib/liblog.o /usr/include/addr.h /usr/include/cron.h /usr/include/file.h /usr/include/proc.h /usr/include/syslogs.h /usr/include/chk.h
Suckit	/dev/.golf
Volc	/usr/bin/volc /usr/lib/volc
Gold2	/usr/bin/ishit
TC2 Worm	/usr/info/.tc2k
ZK Rootkit	/etc/sysconfig/console/load.zk
ShKit	/lib/security/.config /etc/ld.so.hash
AjaKit & Tuxkit	/lib/.ligh.gh /dev/tux
zaRwT	/bin/imin /bin/imout
Fu rootkit	/sbin/xc /bin/.lib /usr/include/ivtype.h
ESRK	/usr/lib/tcl5.3
ENYELKM	/etc/.enyelkmOCULTAR.ko

ตารางที่ 2.1 ชุดโปรแกรมปกปิดการบุกรุก และไดเร็กทอรีหรือไฟล์ที่ถูกสร้างขึ้น

การค้นหาชุดโปรแกรมปกปิดการบุกรุก โดยการใช้คำสั่ง find ในการค้นหาไฟล์พื้นฐานของชุดโปรแกรมปกปิดการบุกรุก มีดังตารางที่ 2.2

ชุดโปรแกรมปกปิดการบุกรุก	คำสั่งที่ใช้ในการตรวจสอบ
t0rn v8	find /lib -name libproc.a find /usr/lib -name libproc.a find /usr/local/lib -name libproc.a
Maniac rootkit	find /usr/bin -name mailrc
RK17 rookit	find /bin -name rtty -o -name squit find /sbin -name pback find /usr/man/man3 -name psid find /proc -name kset
Adore Worm	find /usr/lib /usr/bin -name red.tar -o -name start.sh -o -name klogd.o -o -name 0anacron-bak -o -name adore find /usr/lib/lib /usr/lib/libt
ShitC Worm	find /bin -name homo -o -name frgy -o - name dy find /usr/bin -type d -name dir find /usr/sbin -name in.slogind
Omega Worm	find /dev -name chr
China Worm (Sadmind/IIS Worm)	find /dev/cuc
MonKit	find /lib/defs /usr/lib/libpikapp.a
OpticKit	find /usr/bin/xchk /usr/bin/xsf
T.R.K	find /usr/bin -name xchk -o -name xsf
Mithra's Rootkit	find /usr/lib/locale -name uboot
OpenBSD rootkit v1	find /usr/lib/security/libgcj.security
LOC rootkit	find /tmp -name xp -o -name kidd0.c

ตารางที่ 2.2 การตรวจสอบหาไฟล์พื้นฐานของชุดโปรแกรมปกปิดการบุกรุกด้วยคำสั่ง find

การค้นหาชุดโปรแกรมปกปิดการบุกรุก โดยการใช้คำสั่ง grep และ egrep ในการค้นหาข้อมูล String ที่อยู่ภายในไฟล์ หรือไดเร็กทอรีต่างๆ ในการตรวจสอบหาไฟลพื้นฐานของชุดโปรแกรมปกปิดการบุกรุก มีดังตารางที่ 2.3

ชุดโปรแกรมปกปิดการบุกรุก	คำสั่งที่ใช้ในการตรวจสอบ
HKRK	egrep "\.hk" /etc/rc.d/init.d/network
LPD Worm	grep "^kork" /etc/passwd grep "^ *666" /etc/inetd.conf

ตารางที่ 2.3 การตรวจสอบหาไฟล์พื้นฐานของชุดโปรแกรมปกปิดการบุกรุกด้วยคำสั่ง grep

การตรวจสอบหาชุดโปรแกรมปกปิดการบุกรุกในโหมดยูสเซอร์นั้น ยังต้องมีการตรวจหาไดเร็กทอรีที่ต้องสงสัยจากการสร้างขึ้นจากชุดโปรแกรมปกปิดการบุกรุกหลายๆ ชนิด ดังตารางที่ 2.4

usr/lib/pt07	usr/bin/atm	tmp/.cheese	/dev/ptyzx
dev/ptyzy	usr/bin/sourcemask	dev/ida	dev/xdf1
dev/xdf2	usr/bin/xstat	tmp/982235016-gtkrc-429249277	usr/bin/sourcemask
/usr/bin/ras2xm	usr/sbin/in.telnet	sbin/vobiscum	usr/sbin/jcd
usr/sbin/atd2	usr/bin/.etc	etc/ld.so.hash	sbin/init.zk
usr/lib/in.httpd	usr/lib/in.pop3d		

ตารางที่ 2.4 ไดเร็กทอรีที่มักถูกสร้างขึ้นโดยชุดโปรแกรมปกปิดการบุกรุก

นอกจากมีการรวบรวมไฟล์พื้นฐาน และไฟล์หรือไดเร็กทอรีที่มักจะมีการสร้างขึ้นโดยชุดโปรแกรมปกปิดการบุกรุกสร้างขึ้นแล้ว การตรวจสอบหาชุดโปรแกรมปกปิดการบุกรุกในโหมดยูสเซอร์นั้น จะทำการตรวจสอบคำสั่งที่ใช้ในระบบปฏิบัติการลินุกซ์ ดังตารางที่ 2.5

คำสั่งในระบบปฏิบัติการ	การตรวจสอบความผิดปกติ
chfn	egrep ^/bin/.*sh$\|bash\|elite$|vejeta|\.ark
ls	egrep dev/ttyof|/dev/pty[pqrs]|/dev/hdl0 |\.tmp/lsfile|/dev/hdcc|/dev/ptyxx|duarawkz|^/prof|/dev/tux|/security|file\.h
du	egrep /dev/ttyof|/dev/pty[pqrsx]|w0rm|^/prof|/dev/tux|file\.h
named	egrep blah|bye
netstat	egrep /dev/hdl0/dev/xdta|/dev/ttyoa|/dev/pty[pqrsx]|/dev/cui|/dev/hdn0|/dev/cui221|/dev/dszy|/dev/ddth3|/dev/caca|^/prof|/dev/tux|grep|addr\.h
ps	egrep /dev/xmx|\.1proc|/dev/ttyop|/dev/pty[pqrsx]|/dev/cui|/dev/hda[0-7]|\ /dev/hdp|/dev/cui220|/dev/dsx|w0rm|/dev/hdaa|duarawkz|/dev/tux|/security|^proc\.h
pstree	egrep /dev/ttyof|/dev/hda01|/dev/cui220|/dev/ptyxx|^/prof|/dev/tux|proc\.h
crontab	egrep crontab.*666
top	egrep /dev/xmx|/dev/ttyop|/dev/pty[pqrsx]|/dev/hdp|/dev/dsx|^/prof/|/dev/tux|^/proc\.h
pidof	egrep /dev/pty[pqrs]
killall	egrep /dev/ttyop|/dev/pty[pqrs]|/dev/hda[0-7]|/dev/hdp|/dev/ptyxx|/dev/tux|proc\.h
telnetd	egrep cterm100|vt350|VT100|ansi-term|/dev/hda[0-7]

ตารางที่ 2.5 การตรวจสอบคำสั่งในระบบปฏิบัติการลินุกซ์

2.4 รูตคิตในโหมดเคอร์นัล (Kernel-Mode Rootkit)

จริงๆ แล้วรูตคิตประเภทนี้ เรียกได้ว่าเป็นส่วนเติมเต็มให้กับรูตคิตในโหมดยูส น่าจะเหมาะสมกว่า ในปัจจุบันรูตคิตที่พบส่วนมากจะเป็นชนิดนี้ เนื่องจากผู้ใช้สามารถตรวจพบได้ยากกว่า โดยที่โปรแกรมประเภท File Integrity Checker ไม่อาจสามารถตรวจพบได้ เนื่องจาก รูตคิตประเภทนี้ไม่ได้ทำการเขียนทับไฟล์คำสั่งในระบบปฏิบัติการ แต่จะใช้วิธีการเปลี่ยนแปลงระบบปฏิบัติการของผู้ใช้งานในระดับเคอร์นัล

2.4.1 ลินุกซ์เคอร์นัล

เคอร์นัลเป็นศูนย์กลางของการประมวลผล บริหารจัดการทรัพยากรทั้งหลายภายในระบบ ดังนั้น คุณสมบัติหลายสิ่งหลายอย่างของระบบจึงถูกกำหนดไว้ในตัวเคอร์นัลนี้เอง ได้แก่ ความสามารถในการทำงานกับระบบจัดเก็บไฟล์ต่างๆ โปรโตคอลสำหรับงานด้านสื่อสารข้อมูล และเครือข่าย รวมไปถึงการรองรับอุปกรณ์ชนิดต่างๆ อีกด้วย ดังรูปที่ 2.3

รูปที่ 2.3 โครงสร้างการทำงานของลินุกซ์เคอร์นัล

ซึ่งเคอร์นัลเป็นตัวควบคุมการทำงานที่สำคัญๆ ของระบบปฏิบัติการ ดังนี้

- Process and Thread control เป็นตัวจัดการโพรเซสต่างๆ
- Interprocess communication control เป็นตัวจัดการการทำงานระหว่างโพรเซส
- Memory Control เป็นตัวจัดการเกี่ยวกับหน่วยความจำของระบบ
- File system control จัดการเกี่ยวกับไฟล์ต่างๆ
- Other hardware control เป็นตัวจัดการเกี่ยวกับอุปกรณ์ต่างๆ ที่ต่อเข้ากับระบบปฏิบัติการ เช่น เมาส์, คีย์บอร์ด, จอมอนิเตอร์ เป็นต้น
- Interrupt control เป็นตัวจัดการเกี่ยวกับ interrupt ที่เรียกมาโดยโปรแกรมต่างๆ

ด้วยเหตุนี้จึงมีผู้พยายามที่จะบุกรุกเข้าสู่ระบบปฏิบัติการของผู้ใช้ เพราะถ้าทำการเปลี่ยนแปลงเคอร์นัลได้ จะมีผลกระทบต่อระบบโดยรวมของผู้ใช้งานทั้งหมด

2.4.2 วิธีการเปลี่ยนแปลงเคอร์นัลของรูตคิตในโหมดเคอร์นัล

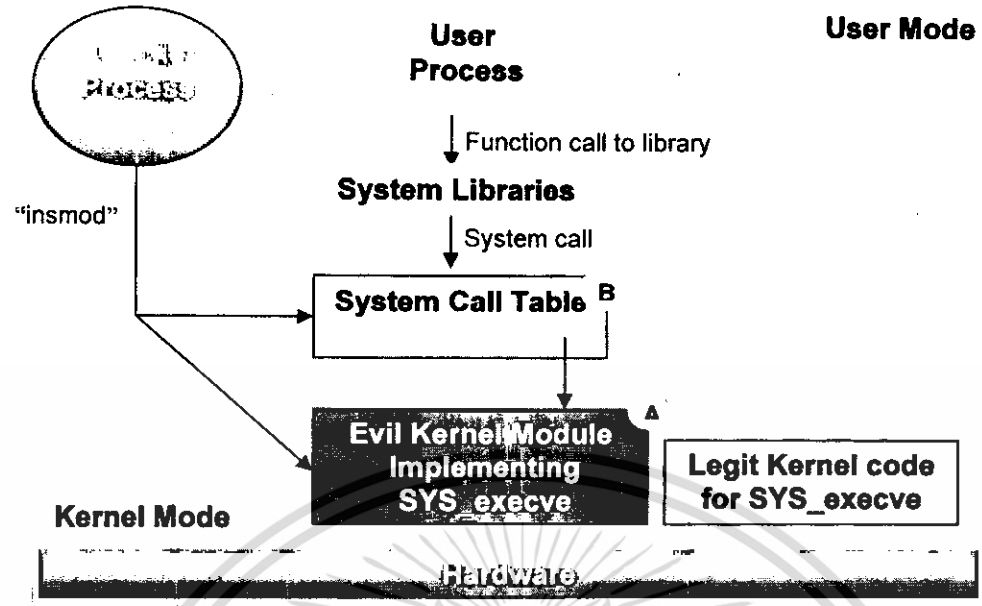
2.4.2.1 เคอร์นัลโมดูล (Kernel Module)

เคอร์นัลโมดูลนั้นถูกพัฒนาขึ้นมาเพื่อเป็นประโยชน์ต่อผู้ใช้ในการเชื่อมต่อฮาร์ดแวร์ใหม่ๆ เข้าสู่ระบบ หรือทำให้สามารถรองรับการทำงานเพิ่มเติมได้โดยที่ไม่ต้องทำการรีบูตระบบปฏิบัติการ แต่ด้วยความสะดวกแก่ผู้ใช้เหล่านี้ ทำให้มีผู้ที่ไม่ประสงค์ดีใช้ในการใส่โค้ดที่มีอันตรายต่อระบบ และยังสามารถใช้งานได้ทันที ซึ่งวิธีนี้เป็นวิธีที่ง่ายที่สุดในการติดตั้งรูตคิตแก่ระบบปฏิบัติการของผู้ใช้งาน

รูตคิตเคอร์นัลโมดูลส่วนมาก มักจะโหลดตัวเองเข้าสู่เคอร์นัล จากนั้นจะทำการเปลี่ยนแปลงตารางซีสเต็มคอลของระบบปฏิบัติการลินุกซ์

ยกตัวอย่าง เช่น คำสั่ง SYS_execve ที่จะถูกเรียกขึ้นมาทุกครั้งเมื่อผู้ใช้ต้องการการรันโปรเซสรูตคิตบางตัวจะเข้าไปทำการเปลี่ยนแปลงคำสั่งนี้ เพื่อคอยตรวจสอบว่า ในกรณีที่ผู้ใช้รันโปรแกรมที่เกี่ยวข้องขึ้นมาทำงาน จะทำการรีไดเร็กไปยังส่วนที่ผู้บุกรุกต้องการ

เพื่อที่จะทำการโจมตีด้วยวิธีการนี้ ผู้บุกรุกจะต้องใช้ 2 ส่วนประกอบกัน คือ A และ B ดังรูปที่ 2.4 โดยที่ผู้บุกรุกจะทำการใส่โมดูลนี้ลงไปยังเคอร์นัลด้วยคำสั่ง insmod เมื่อใช้คำสั่งนี้โมดูลจะถูกทำงานใน kernel mode ในรูปที่ 2.4 ส่วน A จะประกอบไปด้วยโค้ดซึ่งมีการทำงานคล้ายคลึงกับคำสั่ง SYS_execve ดั้งเดิมที่อยู่ในเคอร์นัล เมื่อมีการเรียกใช้คำสั่ง SYS_execve ก็จะถูก redirect ไปยังคำสั่งที่ผู้บุกรุกเพิ่มเติมเข้ามาแทน แต่เมื่อมีการทำงานคำสั่งอื่นๆ ที่ผู้บุกรุกไม่ได้สนใจ ระบบก็จะทำงานตามคำสั่งดั้งเดิมในเคอร์นัล แต่คำสั่ง SYS_execve ของผู้บุกรุกจะไม่สามารถทำงานแทนที่คำสั่งดั้งเดิมได้ หากไม่มีส่วน B คือ เคอร์นัลโมดูลจะทำการเปลี่ยนแปลงตารางซีสเต็มคอล เพื่อที่จะอ้างอิงไปยังคำสั่งใหม่ โดยที่คำสั่ง SYS_execve ดั้งเดิมยังคงอยู่ แต่ไม่มีการใช้งาน ดังนั้นเมื่อผู้ใช้งานตรวจสอบด้วยโปรแกรมประเภท File Integrity Checkers จึงไม่พบความผิดปกติใดๆ และผลลัพธ์ที่ได้ทางหน้าจอจากโปรแกรมของผู้บุกรุกยังเหมือนปกติอีกด้วย



รูปที่ 2.4 kernel module เปลี่ยนแปลงตารางซีสเต็มคอล เพื่อให้ไปทำงานโมดูลอื่นแทน

2.4.2.2 การโจมตีผ่านทาง /dev/kmem

วิธีการนี้มักใช้กับระบบปฏิบัติการที่ไม่รองรับการทำงานของ Linux Kernel Module โดยหลักการเหมือนกับแบบก่อนหน้าที่ทุกประการ ต่างกันตรงที่ว่า ผู้บุกรุกจะเขียนโปรแกรมเข้าไปจองพื้นที่หน่วยความจำในเคอร์นัล จากนั้นจะทำการค้นหาตำแหน่งของตารางซีสเต็มคอล ของระบบ เมื่อโปรแกรมนั้นทำพบ ก็จะทำการเปลี่ยนตำแหน่งหน่วยความจำของตารางซีสเต็มคอลไปยังตำแหน่งที่ผู้บุกรุกต้องการ

2.4.2.3 การแพทช์ Kernel image

เมื่อผู้บุกรุกเข้าไปบุกรุกในเครื่องคอมพิวเตอร์ ก็มักจะทำการติดตั้งหรือเปลี่ยนแปลงเคอร์นัลอิมเมจไฟล์ เพื่อโปรแกรมระบบให้เป็นไปตามที่ต้องการ อย่างเช่น ฟังก์ชันการทำงานของระบบตารางซีสเต็มคอล แทนที่ผู้บุกรุกจะต้องเข้าไปเปลี่ยนปรับเปลี่ยนตารางซีสเต็มคอล แต่ด้วยเคอร์นัลอิมเมจที่ได้จากการคอมไพล์เคอร์นัลใหม่ ทำให้สามารถเปลี่ยนแปลงระบบให้เป็นไปตามที่ต้องการได้ทันที

การแพทช์เคอร์นัลเป็นวิธีการที่ไม่ค่อยนิยมเท่าไรนัก สำหรับผู้บุกรุก เนื่องมาจาก เคอร์นัลเป็นส่วนที่ต้องติดต่อกับอุปกรณ์ฮาร์ดแวร์ในเครื่องคอมพิวเตอร์มากมาย และผู้บุกรุกมักจะบุกรุกเครื่องคอมพิวเตอร์ผ่านทางอินเตอร์เน็ต ดังนั้นผู้บุกรุกมักจะไม่รู้รายละเอียดทางฮาร์ดแวร์ของ

เครื่องผู้ถูกบุกรุก การแพทช์เคอร์นัลโดยที่ไม่ทราบรายละเอียดทางฮาร์ดแวร์ อาจทำให้ระบบล่ม ซึ่งเป็นจุดที่ผู้บุกรุกมักไม่ต้องการ

2.4.3 การตรวจพบชุดโปรแกรมปกปิดการบุกรุกประเภทเคอร์รัล

รูตคิตในโหมดเคอร์นัล ในปัจจุบันเป็นที่นิยมมากสำหรับผู้บุกรุก ซึ่งเคอร์นัล เปรียบเสมือนแกนหลักของระบบปฏิบัติการ โดยที่เคอร์นัลนั้นจะเป็นตัวจัดการเกี่ยวกับ File System, CPU scheduling, การจัดการหน่วยความจำ, และส่วนที่เกี่ยวข้องกับซีสเต็มคอล (System call) ในระบบปฏิบัติการ ซึ่งโปรแกรมที่ทำงานอยู่ในโหมด user-level นั้นจะต้องติดต่อกับเคอร์นัล ผ่านทางซีสเต็มคอล

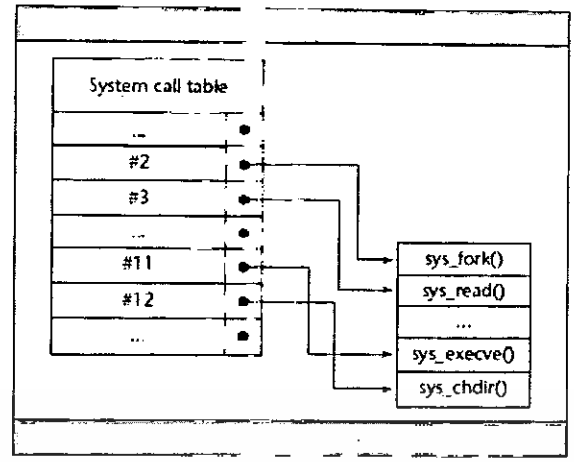
เมื่อโปรแกรมมีการใช้งานซีสเต็มคอล จะส่งผ่านการควบคุมไปยังเคอร์นัลปฏิบัติตามการร้องขอ และส่งผลลัพธ์กลับไปยังโปรแกรมที่ร้องขอ ดังนั้นซีสเต็มคอล จึงเป็นเป้าหมายหลักที่สำคัญของผู้พัฒนารูตคิตในโหมดเคอร์นัล แต่ข้อมูลในเคอร์นัลบางส่วนก็ยังคงเป็นเป้าหมายเช่นกัน

รูตคิตในโหมดเคอร์นัลมีการทำงานซึ่งสามารถแทนที่หรือแก้ไขตารางซีสเต็มคอล และส่วนอื่นๆ ของเคอร์นัล ได้ และแบ่งออกได้เป็น 3 ประเภท ได้แก่

2.4.3.1 แก้ไขข้อมูลภายในตารางซีสเต็มคอล

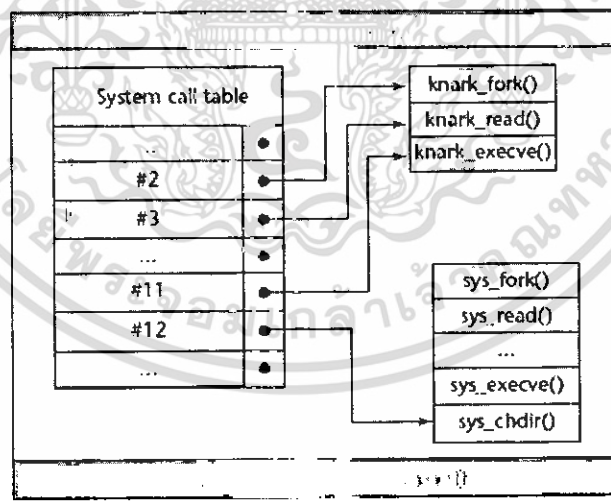
ในการแก้ไขข้อมูลของตารางซีสเต็มคอล ผู้บุกรุกสามารถแก้ไขเคอร์นัลโมดูลให้สามารถซ่อนไฟล์หรือโพรเซสได้ หรือไม่ว่าจะเป็นการสร้าง backdoors สำหรับกลับเข้ามาบุกรุกในภายหลัง และด้วยรูตคิตในประเภทนี้ ผู้บุกรุกสามารถทำการเปลี่ยนแปลงตำแหน่งของซีสเต็มคอลปกติไปเป็นซีสเต็มคอล ที่ถูกสร้างขึ้นมาโดยรูตคิต

ตัวอย่างของ รูตคิตที่ทำงานในประเภทนี้ ได้แก่ knark ซึ่งกำเนิดขึ้นมาเมื่อปี 2001 โดยที่มีการทำงานโดยการมีแก้ไขตารางซีสเต็มคอล โดยในรูปที่ 2.5 จะเป็นตัวอย่างการทำงานตารางซีสเต็มคอล โดยปกติ ก่อนที่จะมีการติดตั้ง knark ซึ่งจะมีการเชื่อมโยงไปยังคำสั่งต่างๆ เช่น sys_fork(), sys_read() เป็นต้น



รูปที่ 2.5 การทำงานของระบบปกติ

หลังจากที่มีการติดตั้งรูตคิตชื่อว่า knark ลงไปยังระบบ จะพบว่ามีการเปลี่ยนแปลงตารางซีสเต็มคอล โดยตัวอย่างเช่น จากปกติคำสั่งในตารางซีสเต็มคอล ที่เรียกไปยังคำสั่ง sys_fork() จะถูกเปลี่ยนแปลงไปยังคำสั่งที่รูตคิตสร้างขึ้นมาคือ knark_fork() เพื่อทำงานตามคำสั่งของ รูตคิตและเมื่อโปรแกรมของผู้ใช้งานมีการเรียกใช้คำสั่งที่โดยปกติแล้วจะทำงานคำสั่ง sys_fork() แล้วจะถูกเปลี่ยนเป็นคำสั่ง knark_fork() โดยอัตโนมัติ



รูปที่ 2.6 หลังจากติดตั้ง knark

2.4.3.2 แก้ไขตำแหน่งของตารางซีสเต็มคอล

ในการทำงานของรูตคิตประเภทนี้ ผู้บุกรุกจะทำการเปลี่ยนแปลง target ของตารางซีสเต็มคอล ไปยังตารางซีสเต็มคอลที่ประกอบไปด้วยโค้ดที่ประสงค์ร้าย โดยที่ไม่ต้องเปลี่ยนแปลง ตาราง

ซีสเต็มคอลปกติของระบบปฏิบัติการ ซึ่งรูตคิตสามารถเขียนทับบางคำสั่งของซีสเต็มคอล ด้วยคำสั่ง jmp ซึ่งจะเปลี่ยนแปลงตำแหน่งไปยังโค้ดที่ประสงค์ร้ายได้ ซึ่งการทำงานของรูตคิตประเภทนี้สามารถตรวจสอบได้โดยการเปรียบเทียบตำแหน่งของ opcode ปัจจุบัน กับค่าที่มาจากแหล่งที่สามารถเชื่อถือได้

2.4.3.3 ทำการเปลี่ยนแปลงตารางซีสเต็มคอล (Redirect system call table)

ในการทำงานของรูตคิตประเภทนี้ ผู้บุกรุกจะทำการเปลี่ยนแปลงตำแหน่งของ ตาราง ซีสเต็มคอลปกติไปยังตารางซีสเต็มคอลใหม่ ที่อยู่ภายในหน่วยความจำของเคอร์นัล ซึ่งตารางใหม่นี้อาจจะประกอบไปด้วยแอดเดรสของซีสเต็มคอลฟังก์ชั่นที่ประสงค์ร้าย

หลังจากที่ศึกษาพฤติกรรมการทำงานที่เกี่ยวกับตารางซีสเต็มคอลของรูตคิตในโหมดเคอร์ ซึ่งมีลักษณะที่สำคัญ 3 ประการ และวิเคราะห์การทำงาน จะสังเกตได้ว่า จะมีการเปลี่ยนแปลงตำแหน่งของคำสั่งในตารางซีสเต็มคอล ดังนั้นในการวิเคราะห์เพื่อตรวจพบรูตคิตในโหมดเคอร์นัล จะต้องประกอบไปด้วย System Integrity Checker คือ การตรวจสอบความถูกต้องของข้อมูล เป็นกลไกที่ทำให้สามารถทราบได้ว่าข้อมูลที่มีความสำคัญนั้นๆ ถูกเปลี่ยนแปลงไปโดยผู้บุกรุกหรือไม่ ซึ่งผู้บุกรุกจะทำการเปลี่ยนแปลงตารางตารางซีสเต็มคอล หรือเคอร์นัลอิมเมจเป็นต้น

2.5 ตัวอย่างการทำงานของรูตคิตที่พบในปัจจุบัน

จากการศึกษาพบว่ารูตคิต ส่วนใหญ่ในปัจจุบันจะใช้วิธีการเขียนโปรแกรมแบบลินุกซ์เคอร์นัลโมดูล ซึ่งในที่นี้จะยกตัวอย่างการทำงานของรูตคิตที่มีอยู่ในปัจจุบัน ดังนี้

2.5.1 Rial

การทำงาน : ทำการปกปิดไฟล์, บางส่วนของไฟล์ และการเชื่อมต่อทางเน็ตเวิร์ค แต่ไม่มีการฝัง backdoor ลงบนเครื่องของผู้ใช้งาน การทำงานในการปกปิดบางส่วนของไฟล์ยังคงมีปัญหาอยู่ ซึ่ง Rial ไม่สามารถปกปิดตัวเองได้ โดยสามารถค้นพบได้โดยการใช้คำสั่ง lsmod หรือ cat /proc/modules เพื่อค้นพบได้

2.5.2 Heroin

การทำงาน : ปกปิดไฟล์ และโพรเซส แต่ไม่ได้สร้าง Backdoor ลงไป รูตคิตตัวนี้ไม่สามารถลบออกจากระบบได้โดยการใช้คำสั่ง rmmod, ซึ่ง heroin มีความพยายามที่จะปกปิดตัวเอง แต่ผู้ใช้งานสามารถค้นพบ heroin ได้โดยการใช้คำสั่ง bash$ cat /proc/ksyms | grep heroin

2.5.3 afhrm

การทำงาน : เปลี่ยนแปลงการเข้าถึงของไฟล์ และมีการปกปิดไฟล์ มีการทำงานบนเคอร์นัล 2.2 แต่การทำงานในการปกปิดไฟล์ของรูตคิตตัวนี้ยังมีปัญหา ซึ่งสามารถปกปิดตัวเองไม่ให้ผู้ใช้ตรวจสอบได้ แต่ในเคอร์นัล 2.2 จะเปิดเผยตัวเองในบางส่วน

2.5.4 Synapsis (v. 0.4)

การทำงาน : ปกปิดไฟล์, โพรเซส, และรายชื่อของผู้ใช้งาน ซึ่ง Synapsis สามารถใช้สิทธิของ root ทำการสร้าง User ID ใหม่โดยส่วนใหญ่จะเป็นเลข 666 และทำการปกปิดการเชื่อมต่อทางเน็ตเวิร์ก และบางทีสามารถหารหัสผ่านจากคำสั่ง cat password อีกด้วย Synapsis ยังสามารถปกปิดตัวเองจากคำสั่ง lsmod แต่จะถูกพบโดยคำสั่ง cat /proc/modules โดยรูตคิตตัวนี้ยังคงมีปัญหาเล็กๆ น้อยๆ เกี่ยวกับการซ่อนไฟล์อยู่

2.5.5 Adore

การทำงาน : ปกปิดไฟล์, โพรเซส, เซอร์วิสต่างๆ และสามารถสั่งการทำงาน โพรเซส เช่น /bin/sh ด้วยสิทธิของ root มีการปกปิดตัวเองไม่ให้ค้นพบได้ และไม่สามารถกำจัดออกจากระบบได้ โดยการใช้คำสั่ง rmmod

2.5.6 knark

การทำงาน : ปกปิดไฟล์, โพรเซส, เซอร์วิสต่างๆ, เปลี่ยนแปลงคำสั่ง และใช้สิทธิของ root ซึ่งมีโปรแกรมที่ช่วยเหลือ เช่น Creed, สามารถใช้คำสั่งโดยผ่านทางไกลได้ อีกทั้ง Knark สามารถปกปิดตัวเองไม่ให้ผู้ใช้ค้นพบได้ และไม่สามารถกำจัดออกจากระบบโดยคำสั่ง rmmod

2.5.7 itf

การทำงาน : สามารถปกปิดไฟล์ และโพรเซส, เปลี่ยนแปลงการเข้าถึงคำสั่ง, มีการดักจับข้อมูลของผู้ใช้งาน, สามารถใช้งานสิทธิของ root โดยที่ itf จะมีการฝังโปรแกรมประเภท backdoor สามามารถปกปิดตัวเองไม่ให้ผู้ใช้ตรวจพบ และไม่สามารถกำจัดออกจากระบบโดยการใช้คำสั่ง rmmod

2.5.8 kis

การทำงาน : คล้ายกับโปรแกรม remote ไปยังเครื่องเครือข่าย โดยฝังตัวเองลงเคอร์นัลรูตคิตลงบน server เพื่อเปิดพอร์ตให้ผู้บุกรุกสามารถเชื่อมต่อเข้ามายัง server ตัวนี้ได้ โดย kis สามารถปกปิดโพรเซส, ไฟล์, และการเชื่อมต่อทางเน็ตเวิร์ก, สามารถเปลี่ยนแปลงการทำงานของคำสั่ง โดยที่สามารถปกปิดตัวเองไม่ให้ผู้ใช้งานตรวจสอบได้ และยังสามารถกำจัด security modules บางตัวของผู้ใช้งานได้

2.6 มาตรการป้องกันและการตรวจหารูตคิต

มาตรการป้องการไม่ให้รูตคิตเข้ามาในระบบเป็นหนทางที่เหมาะสมมากกว่าการพยายามตรวจหารูตคิตที่ติดตั้งตัวเองในระบบแล้วพยายามกำจัด การป้องกันจะหมายถึงการทำตามข้อกำหนดของระบบรักษาความปลอดภัยต่างๆ ซึ่งรวมถึงโซลูชันแอนตี้ไวรัสและแอนตี้สปายแวร์ ไฟร์วอลล์ และการใช้ยูสเซอร์แอคเคาต์ที่ไม่มีสิทธิของการเป็นสมาชิกกลุ่มผู้ดูแลระบบ ถ้าสงสัยว่าถูกรูตคิตโจมตีแล้ว ต้องทำการตรวจสอบระบบด้วยทูลต่างๆ ที่มีอยู่ให้มากที่สุดเท่าที่จะทำได้ แอนตี้สปายแวร์และแอนตี้ไวรัสโซลูชันในปัจจุบันนั้นไม่เพียงพอที่จะต่อกรกับรูตคิตแต่มีโปรแกรมตรวจจับรูตคิต, Kernel debugger และยูทิลิตี้ตรวจสอบโพรเซสที่ตรวจสอบรูตคิตได้ การตรวจหารูตคิต โดยทั่วไปต้องตรวจสอบการทำงานของระบบจากหลายๆ มุมแล้วทำการเปรียบเทียบหาผลที่แตกต่างกันซึ่งอาจจะบ่งชี้การมีอยู่ของรูตคิตได้ ดังนั้นในการตรวจหาการซ่อน พรางโพรเซสของซอฟต์แวร์ไม่พึงประสงค์ ควรรวบรวมเอาต์พุตจากยูทิลิตี้ในการตรวจสอบโพรเซสแล้วทำการเปรียบเทียบกับผลที่ได้จาก System.map การเปรียบเทียบที่ง่ายที่สุดอีกวิธีในการตรวจหาการซ่อนพรางไฟล์และไดเร็กทอรีคือการระบุส่วนที่ส่วนที่กำลังทำงานอยู่ของระบบปฏิบัติการทั้งหมดเทียบกับเครื่องที่เพิ่งทำการติดตั้งใหม่

บทที่ 3
การออกแบบและพัฒนา

3.1 บทนำ

โครงงานระบบต่อต้านชุดโปรแกรมปกปิดการบุกรุกได้พัฒนาขึ้นเพื่อเพิ่มความปลอดภัยของผู้ใช้งานในระบบปฏิบัติการแพลตฟอร์มยูนิกซ์ เพื่อที่จะตรวจสอบชุดโปรแกรมปกปิดการบุกรุก หรือรูตคิต (Rootkit) ซึ่งถือได้ว่าชุดโปรแกรมนี้มีพฤติกรรมการทำงานคล้าย กับมัลแวร์บางชนิด ที่เพิ่มความสามารถในการปกปิดร่องรอยของการทำงานของชุดโปรแกรมเอง และไม่สามารถที่จะตรวจสอบได้ด้วยโปรแกรมต่อต้านไวรัสทั่วไป โดยระบบต่อต้านชุดโปรแกรมปกปิดการบุกรุกจะทำการวิเคราะห์พฤติกรรมต่างๆ ที่ตัวรูตคิตได้มีการทำงาน เช่น การปกปิดไฟล์หรือโฟลเดอร์ที่ผู้บุกรุกได้สร้างขึ้นมาเพื่อวัตถุประสงค์ใดๆ, การปกปิดโพรเซสหรือการทำงานของมัลแวร์, การปกปิดการเชื่อมต่อทางเน็ตเวิร์กของผู้บุกรุก, การปกปิดรายชื่อของผู้ใช้งานที่เป็นของผู้บุกรุก เป็นต้น และระบบต่อต้านชุดโปรแกรมปกปิดการบุกรุกสามารถแจ้งรายงานการตรวจพบรูตคิตให้กับผู้ใช้งานได้ทราบ ซึ่งการพัฒนามีการทำงานหลัก ดังนี้

1. พัฒนาระบบต่อต้านชุดโปรแกรมปกปิดการบุกรุก ซึ่งทำงานบนระบบปฏิบัติการลินุกซ์
2. พัฒนาอัลกอลิทึมที่ใช้ในการวิเคราะห์ และตรวจพบรูตคิต

โครงการนี้ได้ใช้ระบบปฏิบัติการแพลตฟอร์มยูนิกซ์ โดยใช้ระบบปฏิบัติการ Debian 3.1 และใช้ระบบปฏิบัติการลินุกซ์เคอร์นัล 2.6 ในการพัฒนาและทดสอบระบบต่อต้านชุดโปรแกรมปกปิดการบุกรุก

3.2 โครงสร้างของโครงงาน

3.2.1 เครื่องมือที่ใช้ในการพัฒนา

- C/C++ Editor

  เนื่องจากการทำงานส่วนใหญ่ จะทำงานเกี่ยวกับตัวเคอร์นัลเป็นหลัก ดังนั้นจึงเลือกภาษา C และ C++ ซึ่งเป็นภาษาพื้นฐานของระบบในการพัฒนาโปรแกรม

- Linux Kernel

  โครงการนี้มีการใช้ระบบปฏิบัติการ Debian 3.1 โดยที่ใช้ลินุกซ์เคอร์นัล 2.6 ในการสร้างและทดสอบระบบต่อต้านชุดโปรแกรมปกปิดการบุกรุก

- Linux System Call Table Implementation

  ทำการเก็บพฤติกรรม หรือเปลี่ยนแปลงตารางซีสเต็มคอล เพื่อไม่ให้ผู้บุกรุกใช้ชุดโปรแกรมปกปิดการบุกรุกให้สามารถทำงานได้

- Qt Designer Version 4.2

  ใช้ในการพัฒนากราฟฟิกยูสเซอร์อินเทอร์เฟส เพื่อให้ระบบสามารถทำงานในกราฟฟิกโหมด และผู้ใช้สามารถใช้งานได้ง่ายยิ่งขึ้น

- Sun Java Developer Kit (Sun – Java JDK5)

  ใช้ในการพัฒนาโปรแกรมเพื่อทำงานเป็น Daemons ของระบบปฏิบัติการ

- Shell Script

  ใช้ในการตรวจสอบรูตคิตในโหมดยูสเซอร์ เพื่อตรวจสอบคำสั่งต่างๆ ในระบบ หรือตามฐานข้อมูลที่มีอยู่

3.3 รายละเอียดโปรแกรมที่พัฒนา (Software Specification)

3.3.1 รายละเอียดส่วนนำเข้า

- ผู้ใช้งานใช้งานผ่านทางคอมมานไลน์หรือรูปแบบกราฟฟิคอินเตอร์เฟซ
- โปรแกรมทำงานบนระบบปฏิบัติการลินุกซ์เคอร์นัล 2.6
- ผู้ใช้งานเลือกโหมดของการตรวจหาชุดโปรแกรมปกปิดการบุกรุกได้

3.3.2 รายละเอียดส่วนแสดงผล

- ผู้ใช้งานได้รับรายงานการตรวจพบชุดโปรแกรมปกปิดการบุกรุก
- แสดงรายชื่อของคำสั่งในระบบปฏิบัติการที่ถูกเปลี่ยนแปลงโดยชุดโปรแกรมปกปิดการบุกรุก
- การแสดงรายชื่อไฟล์หรือโฟลเดอร์ที่มีความเสี่ยงของไฟล์ที่มีโอกาสเป็นชุดโปรแกรมปกปิดการบุกรุก
- แสดงผลลัพธ์ของการสแกนเพื่อค้นหาชุดโปรแกรมปกปิดการบุกรุกในกราฟฟิกโหมดได้

3.3.3 รายละเอียดฟังก์ชัน

- โปรแกรมสามารถตรวจสอบคำสั่งในระบบปฏิบัติการ ที่ถูกเปลี่ยนแปลงโดยชุดโปรแกรมปกปิดการบุกรุก
- โปรแกรมสามารถตรวจสอบไฟล์ หรือโฟลเดอร์ที่มีความเกี่ยวข้องกับชุดโปรแกรมปกปิดการบุกรุกได้
- โปรแกรมสามารถตรวจสอบ System Call Table ว่าถูกเปลี่ยนแปลงจากชุดโปรแกรมปกปิดการบุกรุกหรือไม่

3.3.4 โครงสร้างของซอฟต์แวร์ (Design)

- มีการแบ่งการทำงานเป็น 2 ส่วนหลัก คือ Rootkit-Scan และ Rootkit-Guard
- Rootkit-Scan อาศัย Signature Based ในการตรวจหาชุดโปรแกรมปกปิดการบุกรุก ซึ่งจะทำให้สามารถค้นพบชุดโปรแกรมปกปิดการบุกรุกประเภท User-Mode Rootkit ได้

- Rootkit-Guard จะมีการทำงานเป็น Daemon ซึ่งช่วยในการตรวจสอบความถูกต้องของตารางซีสเต็มคอลซึ่งจะทำให้สามารถค้นพบชุดโปรแกรมปกปิดการบุกรุกประเภทเคอร์นัลได้

3.3.4 ขอบเขตและข้อจำกัดของโปรแกรมที่พัฒนา

- โปรแกรมทำงานบนระบบปฏิบัติการลินุกซ์เท่านั้น
- สามารถใช้ได้กับลินุกซ์เคอร์นัล 2.6.x เท่านั้น

3.4 การออกแบบและพัฒนา

ระบบต่อต้านชุดโปรแกรมปกปิดการบุกรุกจะเป็นการทำงานในระบบปฏิบัติการลินุกซ์โดยที่จะมีการทำงานเมื่อระบบปฏิบัติการเริ่มต้นทำงาน โดยที่ผู้ใช้สามารถเลือกได้ว่าจะให้ทำงานอยู่ตลอดเวลาเพื่อคอยตรวจสอบ ซึ่งมีการทำงานเป็น Daemon และจะมีการตรวจสอบความถูกต้องของระบบ ซึ่งระบบจะประกอบไปด้วย 2 ส่วนการทำงานหลัก ดังนี้

3.4.1 Rootkit-Scan

Rookit-Scan นั้นจะทำหน้าที่ในการตรวจสอบคำสั่งต่างๆ ที่สำคัญในระบบปฏิบัติการลินุกซ์ว่าถูกเปลี่ยนแปลงไปหรือไม่ และทำการค้นหาไฟล์พื้นฐานของชุดโปรแกรมปกปิดการบุกรุกแต่ละชนิด ซึ่งจะทำงานก็ต่อเมื่อผู้ใช้สั่งให้ทำงานเท่านั้น

Rootkit-Scan นั้นใช้หลักการในการตรวจสอบค้นหาชุดโปรแกรมปกปิดการบุกรุก 2 ทฤษฎี คือ

1. File Integrity Checkers

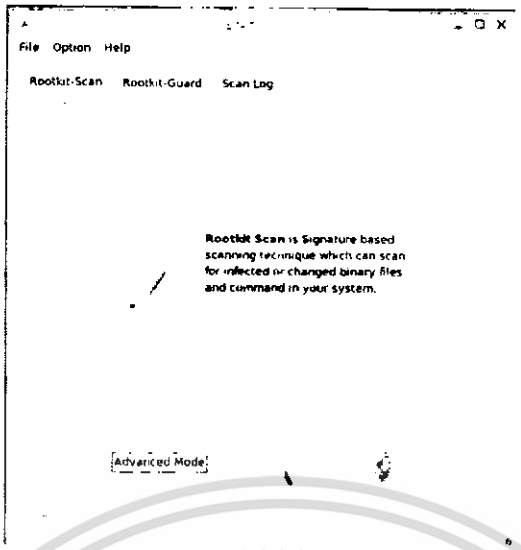
ทำการตรวจสอบข้อมูลคำสั่งที่สำคัญในซิสเต็มว่าถูกเปลี่ยนแปลงไปหรือไม่ ซึ่งจะสามารถตรวจพบชุดโปรแกรมปกปิดการบุกรุกในโหมดยูสเซอร์ได้ แต่จะไม่สามารถตรวจพบ Kernel-mode Rootkit บางตัวได้

2. Signature Based Checkers

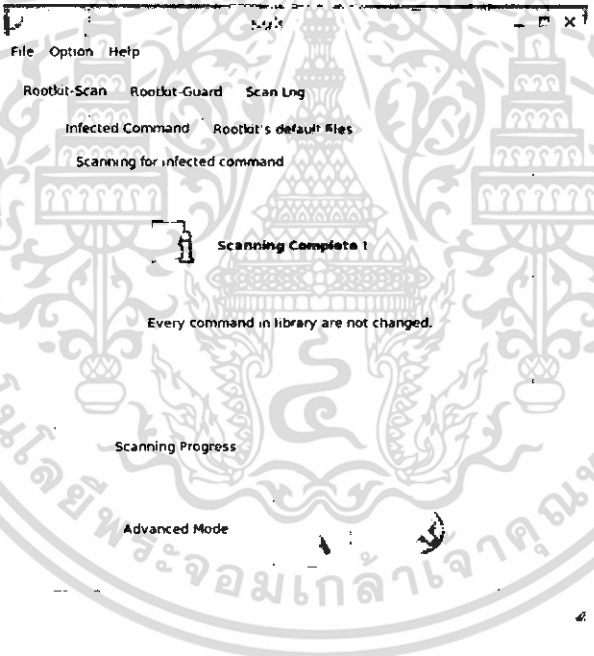
ทำการตรวจสอบลักษณะของไฟล์ที่อาจถูกสร้างโดยรูตคิต เช่น ไฟล์ที่มีชื่อคล้ายคลึงกับคำสั่งต่างๆ ในระบบปฏิบัติการ หรือไฟล์ที่มักจะถูกสร้างโดยรูตคิตชนิดต่างๆ (Rootkit's Default File) โดยในส่วนนี้ต้องทำการเก็บรวบรวมข้อมูลรูตคิตแต่ละชนิดมาเพื่อใช้ในการตรวจสอบ

ซึ่งในการทำงานของ Rootkit-Scan นั้น มีการทำงาน 2 โหมด คือ

- Silent Mode: เป็นโหมดที่จะทำการแสกน และแสดงผลลัพธ์เฉพาะสิ่งผิดปกติที่เกิดขึ้นในระบบปฏิบัติการของผู้ใช้งาน

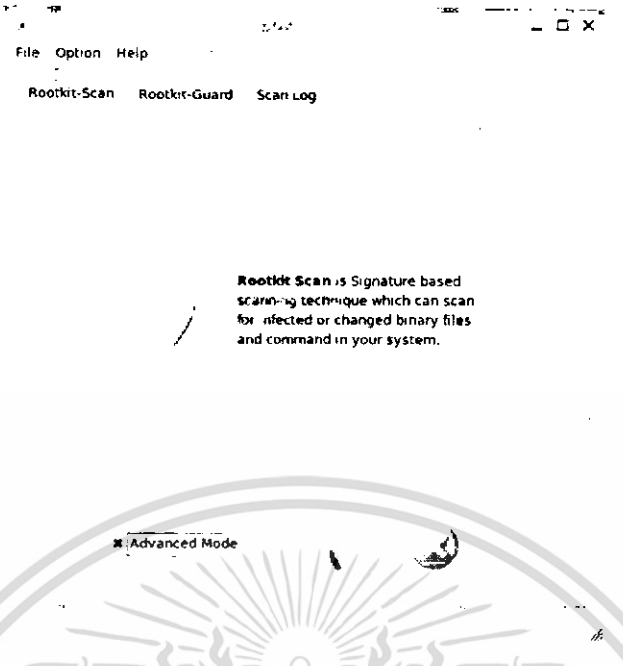


รูปที่ 3.1 การทำงานด้วย Silent Mode



รูปที่ 3.2 ผลลัพธ์จากการแสกนด้วย Silent Mode

- Advanced Mode: เป็นโหมดที่จะแสดงผลลัพธ์จากการแสกนทั้งหมด ทั้งคำสั่งที่มีผลลัพธ์ปกติและผิดปกติในระบบปฏิบัติการของผู้ใช้งาน



รูปที่ 3.3 การทำงานด้วย Advanced Mode

โดยในส่วนของ Rootkit-Scan จะมีการแสดงผลลัพธ์ 2 ส่วนหลัก คือ

- คำสั่ง binary ต่างๆ ถูกแก้ไขหรือไม่
- ค้นหา default file หรือ signature ของชุดโปรแกรมปกปิดการบุกรุกต่างๆ ที่มีอยู่ในปัจจุบัน

Rootkit-Scan นั้นจะทำการตรวจสอบคำสั่งที่สำคัญในระบบปฏิบัติการลินุกซ์ ซึ่งทำการตรวจสอบคำสั่ง มีรายชื่อดังต่อไปนี้

amd	basename	biff	chfn
chsh	cron	crontab	date
du	dirname	echo	egrep
env	find	fingerd	gpm
grep	hdparm	su	ifconfig
inetd	inetdconf	identd	init
killall	ldsopreload	login	ls
lsof	mail	mingetty	netstat
named	passwd	pidof	pop2
pop3	ps	pstree	rpcinfo

rlogind	rshd	slogin	sendmail
shd	ssyslogd	tar	tcpd
tcpdump	top	telnetd	timed
traceroute	vdir	w	write

ตารางที่ 3.1 คำสั่งที่มีการตรวจสอบโดย Rootkit-Scan

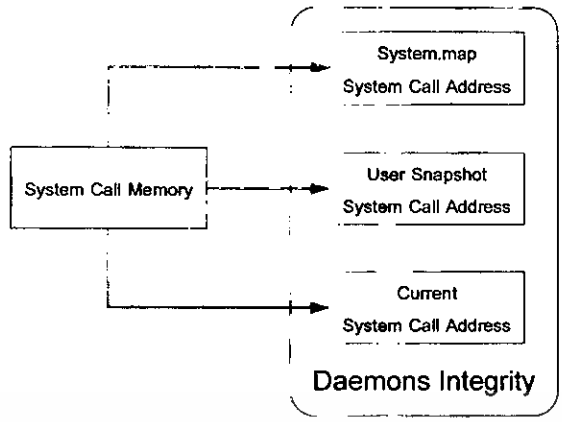
ซึ่งผลลัพธ์จากการสแกน จะแบ่งออกเป็น 4 แบบ คือ

- not found คือ ไม่พบคำสั่งนั้นในระบบปฏิบัติการ
- not infected คือ คำสั่งในระบบปฏิบัติการคำสั่งนั้นไม่มีการเปลี่ยนแปลงโดยชุดโปรแกรมปกปิดการบุกรุก สามารถเชื่อถือผลลัพธ์จากคำสั่งนั้นๆ ได้ เช่น ผลลัพธ์จาก) คำสั่งls หรือ netstat เป็นต้น(
- infected คือ คำสั่งนั้นๆ ถูกเปลี่ยนแปลงโดยชุดโปรแกรมปกปิดการบุกรุก (ไม่ควรเชื่อถือผลลัพธ์ของคำสั่งๆ นั้น เช่น คำวสั่ง ls อาจจะถูกแก้ไขให้ไม่แสดงชื่อไฟล์และไดเร็กทอรีที่เก็บไฟล์ของชุดโปรแกรมปกปิดการบุกรุก(
- not test คือ ไม่สามารถตรวจสอบคำสั่งนั้นๆ ได้ ซึ่งอาจจะเกิดจากสิทธิในการเข้าถึงไฟล์หรือคำสั่งของผู้ใช้งาน

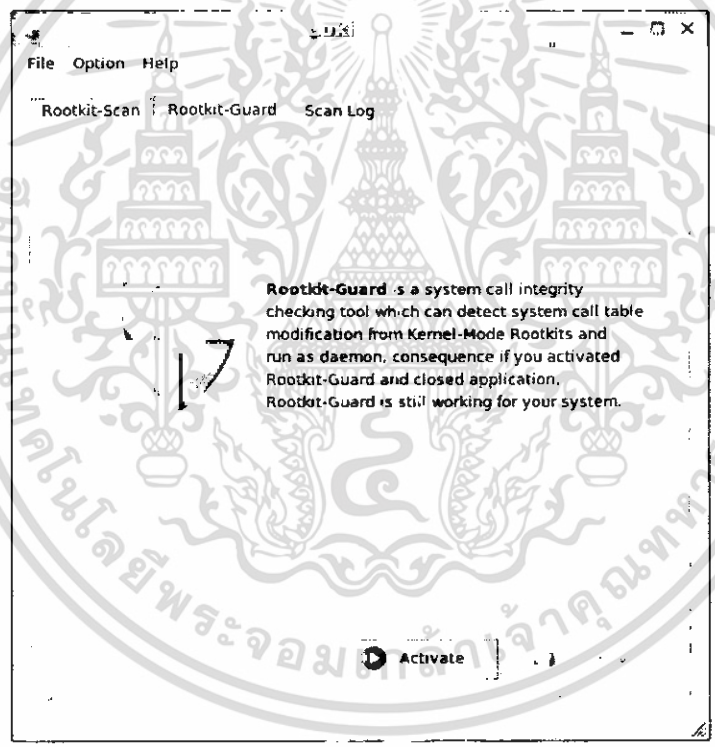
3.4.2 Rootkit-Guard

สำหรับในส่วนของ Rootkit-Guard เป็นส่วนที่ใช้วิเคราะห์หาชุดโปรแกรมปกปิดการบุกรุกในแบบเคอร์นัลที่ใช้หลักการเปลี่ยนแปลงของตารางซีสเต็มคอลในการทำงาน ซึ่งโดยส่วนมากแล้วชุดโปรแกรมปกปิดการบุกรุกประเภทนี้ มักจะใช้เคอร์นัลโมดูลในการทำงาน โดยส่วนของ Rootkit – Guard จะตรวจสอบจากตารางซีสเต็มคอล ว่ามีการเปลี่ยนแปลงไปหรือไม่ ถ้ามีการเปลี่ยนแปลงโปรแกรมจะทำการเตือนทันที ซึ่งถ้าการเปลี่ยนแปลงนั้นตรงกับ Signature ของโปรแกรมที่มีอยู่จะแสดงรายชื่อของชุดโปรแกรมปกปิดการบุกรุกขึ้นมา แต่ถ้าไม่ จะเป็นการเตือนเพื่อให้ผู้ใช้ได้ทราบ

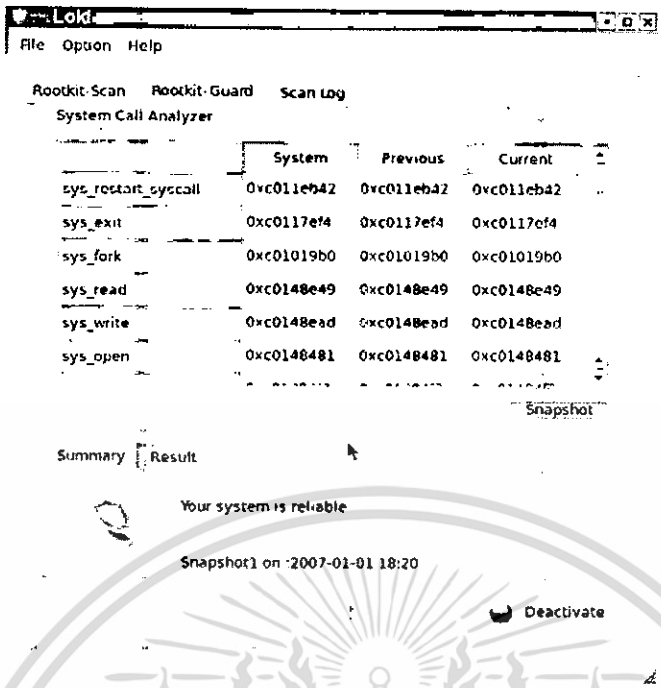
โดยการทำงานในส่วนนี้ จะใช้ระบบที่เรียกว่า Daemon มาใช้งาน เนื่องจากต้องการให้โปรแกรมทำงานอยู่เบื้องหลังอยู่ตลอดเวลา ตัวโปรแกรมจะทำการเปรียบตำแหน่งของหน่วยความจำที่ได้จากตารางซีสเต็มคอล ในรูปแบบต่างๆ ตามรูปที่ 3.1 เมื่อเวลาผ่านไป ถ้าตำแหน่งของหน่วยความจำที่ได้ มีการเปลี่ยนแปลง ตัวโปรแกรมจะสามารถตรวจสอบได้ทันที



รูปที่ 3.4 แสดงการทำงานของ Rootkit - Guard



รูปที่ 3.5 Rootkit-Guard ขณะยังไม่ทำการตรวจสอบ



รูปที่ 3.6 Rootkit-Guard ไม่ตรวจพบสิ่งผิดปกติ

เมื่อ Rootkit-Guard ทำงานแล้วจะมีตารางแสดงการตรวจสอบและวิเคราะห์ตารางซีสเต็มคอล ที่มีในระบบ และมีผลลัพธ์แสดงให้ผู้ใช้งานได้ทราบ โดยมีส่วนต่างๆ ดังนี้

- System Call Table Analyzer แสดงผลตารางของตำแหน่งของหน่วยความจำของตารางซีสเต็มคอลที่มีอยู่ในระบบ โดยจะแบ่งออกเป็น 3 ส่วนซึ่งได้แก่ System ที่ได้มาจาก Symbol map ของเคอร์นัลในระบบ, Previous ซึ่งได้มาจากการที่ User สั่ง Snapshot ระบบ และสุดท้าย Current ซึ่งเป็นตำแหน่งของ System Call Table ของระบบ ณ เวลาปัจจุบัน
- ปุ่ม Snapshot ไว้ให้ผู้ใช้บันทึกตำแหน่งของหน่วยความจำในตารางซีสเต็มคอล ณ เวลาที่ต้องการ เพื่อไว้เปรียบเทียบต่อไปในอนาคต
- แท็บ Summary จะรายงานระบบโดยรวมขณะนั้นของตารางซีสเต็มคอล
- แท็บ Result เป็นผลลัพธ์ที่ได้จากโปรแกรม Loki เมื่อตรวจพบความผิดปกติในส่วนตารางซีสเต็มคอลของระบบ
- ปุ่ม Activate , Deactivate มีไว้เพื่อปิดหรือเปิด Daemons ของระบบ

บทที่ 4
การทดลองและผลการทดลอง

4.1 บทนำ

ในบทนี้การทดลองและผลการทดลองจากโครงงานซึ่ง ในขั้นตอนแรกจะเป็นการศึกษาพฤติกรรมการทำงานของชุดโปรแกรมปกปิดการบุกรุก และทดลองเขียนชุดโปรแกรมปกปิดการบุกรุกเพื่อทดลอง, ส่วนที่สองจะเป็นการทดลองติดตั้งชุดโปรแกรมปกปิดการบุกรุกแบบยูสเซอร์โหมด ซึ่งมีการเปลี่ยนแปลงการทำงานของคำสั่งในระบบปฏิบัติการ แต่ไม่มีการเปลี่ยนแปลงตารางซีสเต็มคอล (System Call Table) และตรวจสอบค้นหาด้วยระบบต่อต้านชุดโปรแกรมปกปิดการบุกรุก (Loki) และส่วนสุดท้ายจะเป็นการทดลองติดตั้งชุดโปรแกรมปกปิดการบุกรุกแบบเคอร์นัลโหมด ซึ่งจะมีการเปลี่ยนแปลงตารางซีสเต็มคอลและตรวจสอบด้วยชุดโปรแกรมปกปิดการบุกรุก (Loki)

4.2 การทดลองศึกษาพฤติกรรมของชุดโปรแกรมปกปิดการบุกรุกกับ Linux kernel module ที่ได้สร้างขึ้นเพื่อการทดลอง

การทดลองสร้างชุดโปรแกรมที่ทำงานเลียนแบบชุดโปรแกรมปกปิดการบุกรุกบางฟังก์ชันการทำงาน โดยในที่นี้คือฟังก์ชันการซ่อนไฟล์ โดยโปรแกรมที่เขียนได้ทำเป็นรูปแบบของ Linux kernel module ที่ไปทำการแก้ไขตารางซีสเต็มคอลของฟังก์ชัน Sys_getdent64 () ซึ่งเป็นฟังก์ชันที่ถูกเรียกเมื่อลินุกซ์ใช้คำสั่งที่เกี่ยวกับการแสดงไฟล์ ยกตัวอย่างเช่น คำสั่ง ls เป็นต้น

ในการทดลองนี้ ชุดโปรแกรมที่สร้างขึ้นจะทำการซ่อนไฟล์ทั้งหมดที่มีคำว่า "xxxxx" อยู่ ซึ่งผลของการทดลองเป็นดังรูปข้างล่าง จากรูปที่ 4.1 ในขั้นตอนที่ 1 แสดงคำสั่ง ls พบไฟล์ "xxxxx.x" ในโฟลเดอร์ /sample ขั้นตอนที่ 2 เมื่อทำใส่เคอร์นัลโมดูลที่เขียนขึ้นมาชื่อว่า "hidels.ko" ลงไปในระบบและเรียกใช้คำสั่ง ls อีกรอบ ปรากฏว่าไม่พบไฟล์ที่ชื่อ "xxxxx.x" อยู่ใน โฟลเดอร์ /sample และเมื่อเอาเคอร์นัลโมดูลที่เขียนขึ้นออก และเรียกใช้คำสั่ง ls อีกครั้งตามขั้นตอนที่ 3 จะพบว่าระบบกลับมาทำงานถูกต้องอีกครั้ง

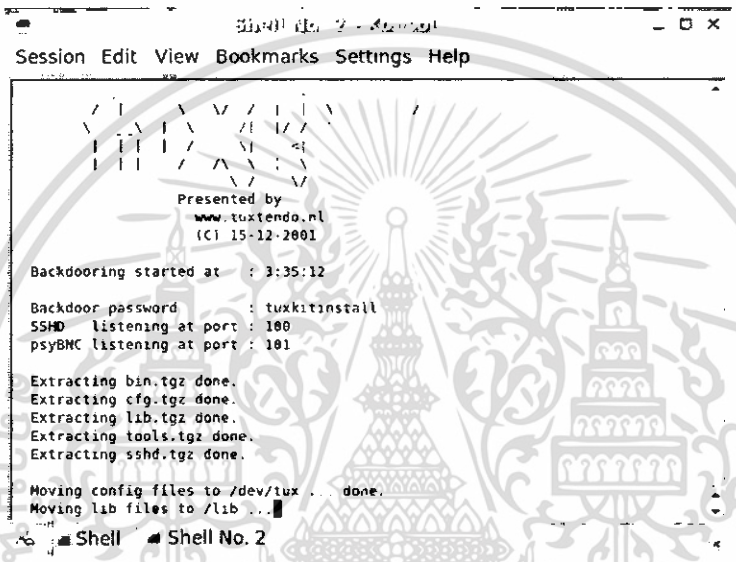
```
debian:~/sample# ls 1
1.txt  5.txt  xxxxx.x
debian:~/sample# insmod /root/hidels.ko 2
debian:~/sample# ls
1.txt  5.txt
debian:~/sample# rmmod hidels.ko 3
debian:~/sample# ls
1.txt  5.txt  xxxxx.x
debian:~/sample#
```

รูปที่ 4.1 แสดงผลการทำงานของ Loadable kernel module ที่สร้างขึ้น

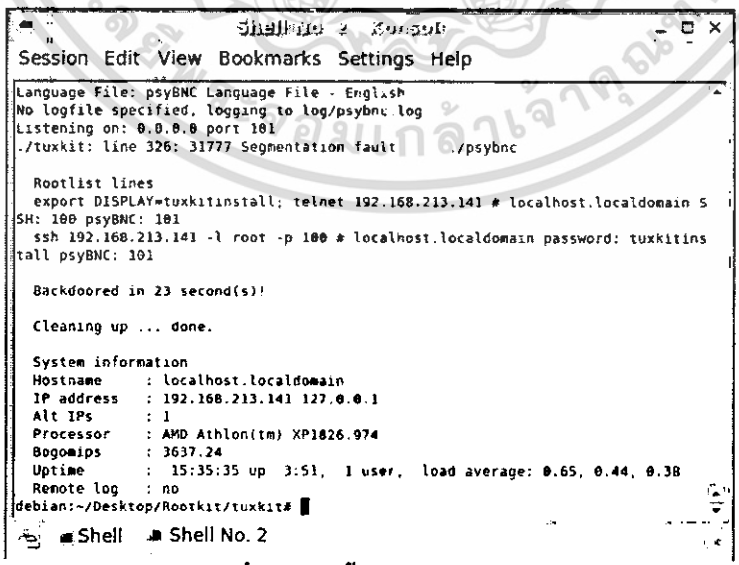
4.3 การทดลองตรวจหาชุดโปรแกรมปกปิดการบุกรุก tuxkit ด้วยระบบต่อต้านชุดโปรแกรมปกปิดการบุกรุก (Loki)

ชุดโปรแกรมปกปิดการบุกรุก tuxkit เป็นชุดโปรแกรมปกปิดการบุกรุกในโหมดยูสเซอร์ โดยจะทำการเปลี่ยนแปลงคำสั่งในระบบปฏิบัติการ เช่น คำสั่ง ls ทำให้ผู้ใช้ม่สามารถเห็น และค้นหาไดเร็กทอรีของ tuxkit ได้

ทำการติดตั้งชุดโปรแกรมปกปิดการบุกรุก tuxkit โดยใช้คำสั่ง ./tuxkit แล้วตามด้วยพาสเวิร์ด และพอร์ตที่ใช้ในการทำ backdoor และ tuxkit จะทำการติดตั้งลงในระบบปฏิบัติการ ดังรูป

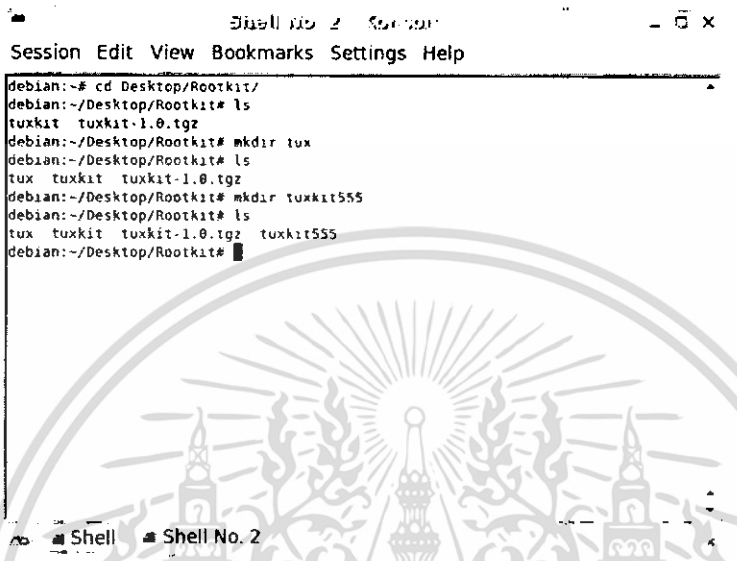


รูปที่ 4.2 การติดตั้ง tuxkit

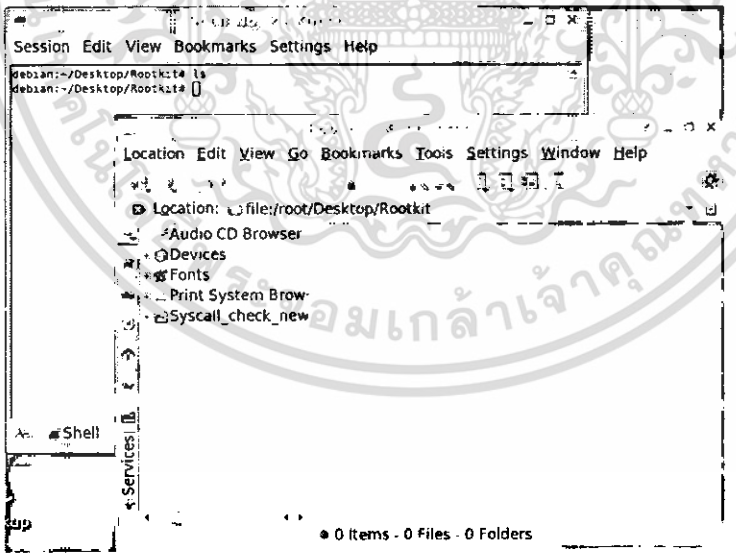


รูปที่ 4.3 ติดตั้ง tuxkit สมบูรณ์

หลังจากที่ติดตั้ง tuxkit เสร็จสมบูรณ์แล้ว ไฟล์และโฟลเดอร์ของ tuxkit จะถูกย้ายไดเร็กทอรีไปฝังลงในระบบปฏิบัติการและเนื่องจาก tuxkit มีการไปเปลี่ยนแปลงคำสั่ง ls ซึ่งใช้ในการดูรายละเอียดของไฟล์และไดเร็กทอรี โดยที่ชุดโปรแกรมปกปิดการบุกรุกตัวนี้ มีการเขียนคำสั่งให้ซ่อนไฟล์และโฟลเดอร์ทั้งหมดที่มีส่วนประกอบของคำว่า tuxkit ดังนั้น ผู้ใช้จะไม่สามารถค้นหาไฟล์ของ tuxkit พบ ดังรูป



รูปที่ 4.4 แสดงรายชื่อไฟล์ก่อนทำการติดตั้ง tuxkit



รูปที่ 4.5 แสดงรายชื่อไฟล์หลังทำการติดตั้ง tuxkit

หลังจากติดตั้งชุดโปรแกรมปกปิดการบุกรุก tuxkit เรียบร้อยแล้ว ขั้นตอนต่อไปเป็นการเปิดระบบต่อต้านชุดโปรแกรมปกปิดการบุกรุก (Loki) เพื่อตรวจสอบหาชุดโปรแกรมปกปิดการบุกรุก

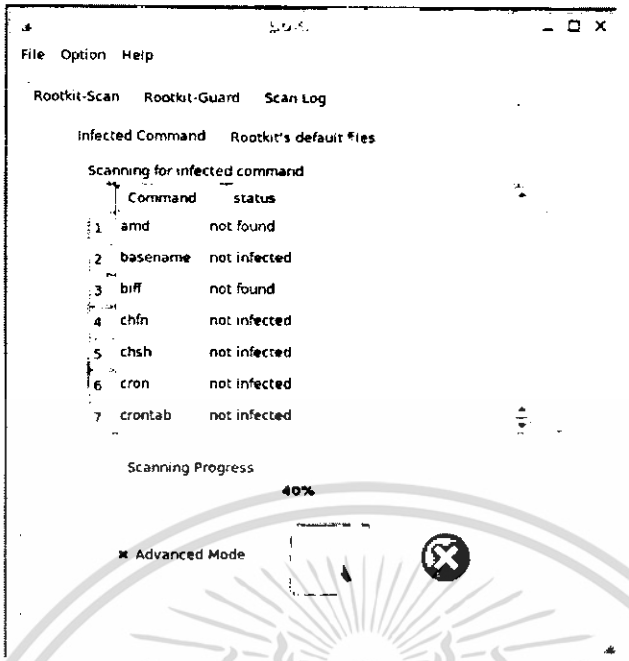
โดย การพิมพ์คำสั่ง ./Anti-Rootkit ดังรูป

```
debian:~# cd /Anti-Rootkit
debian:/Anti-Rootkit# ./Anti-Rootkit
```

รูปที่ 4.6 เปิดโปรแกรมด้วยคำสั่ง ./Anti-Rootkit

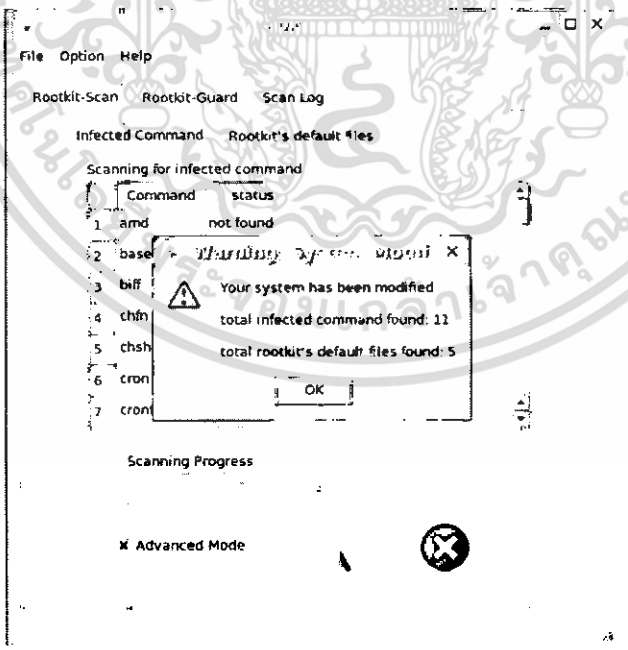
รูปที่ 4.7 หน้าต่างหลักของโปรแกรม

ทดลองทำการแสกนหารูตคิตด้วย Rootkit-Scan โดยใช้ Advanced Mode ดังรูป



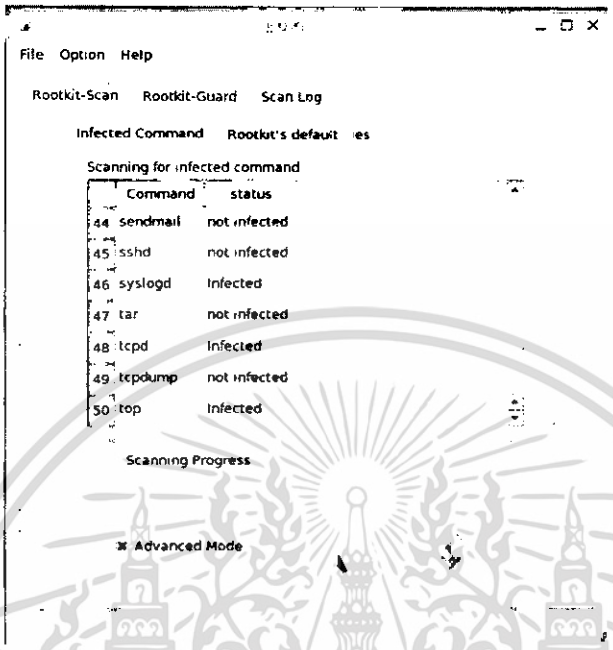
รูปที่ 4.8 ขณะทำการสแกน

หลังจากทำการสแกนเสร็จเรียบร้อยแล้ว โปรแกรมจะทำการแจ้งเตือนการพบคำสั่งในระบบปฏิบัติการ ถูกเปลี่ยนแปลงไป และค้นพบไฟล์พื้นฐานของรูตคิต ดังรูป



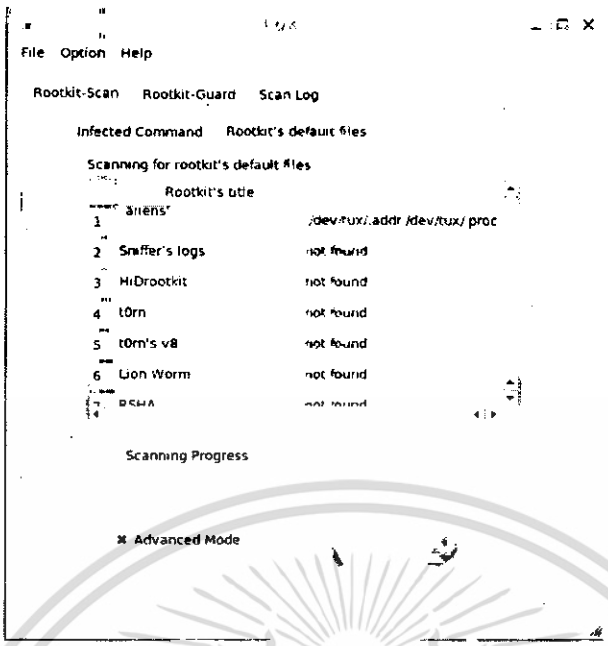
รูปที่ 4.9 มีการแจ้งเตือนพบชุดโปรแกรมปกปิดการบุกรุก

โดยใน Advanced Mode ผู้ใช้สามารถดูแต่ละคำสั่งได้ว่าคำสั่งใดถูกเปลี่ยนแปลงไปโดยชุดโปรแกรมปกปิดการบุกรุก ดังรูป



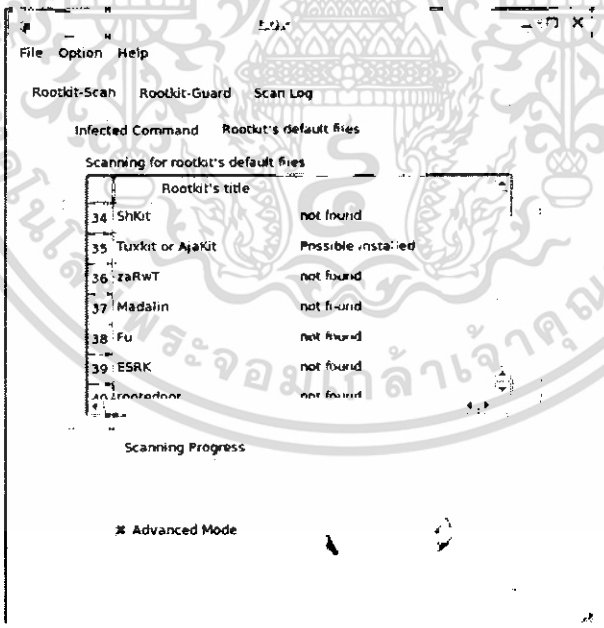
รูปที่ 4.10 แสดงคำสั่งที่ถูกเปลี่ยนแปลง

และในส่วนของผลลัพธ์เกี่ยวกับไฟล์พื้นฐานของรูตคิต จะแสดงโฟลเดอร์ที่ต้องสงสัยว่าจะถูกสร้างขึ้นมาโดยชุดโปรแกรมปกปิดการบุกรุก โดยในการทดลองจะแสดงไดเร็กทอรี /dev/tux/.addr และ /dev/tux/.proc เป็นต้น ซึ่งในที่นี้ถูกสร้างขึ้นโดย Tuxkit ดังรูป



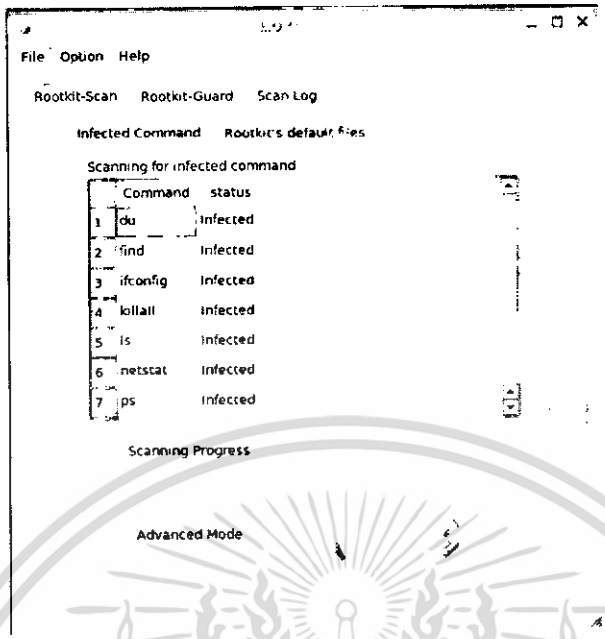
รูปที่ 4.11 แสดงไดเร็กทอรีที่เสี่ยงต่อการถูกสร้างโดยรูตคิต

ผลลัพธ์เกี่ยวกับไฟล์พื้นฐานของรูตคิตได้ประเมินว่าอาจจะเป็นการติดตั้ง Tuxkit หรือ AjaKit ดังรูป



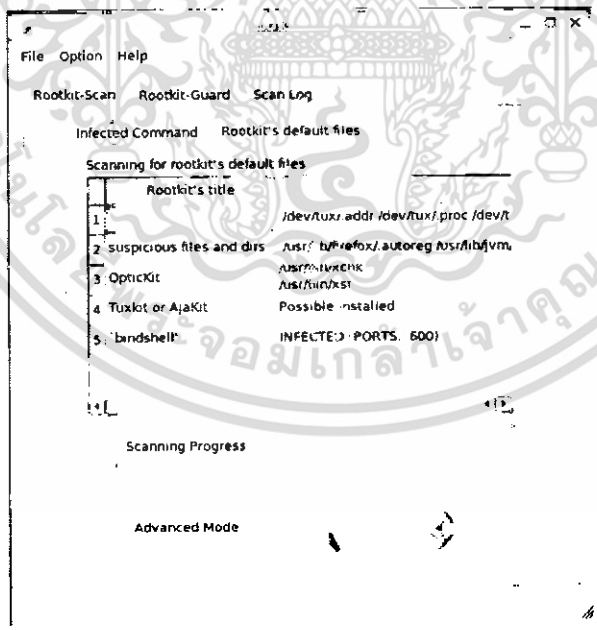
รูปที่ 4.12 วิเคราะห์ผลลัพธ์ว่าเป็น Tuxkit หรือ AjaKit

การสแกนด้วย Silent Mode จะแสดงผลลัพธ์เฉพาะคำสั่งที่ถูกเปลี่ยนแปลงโดยชุดโปรแกรมปกปิดการบุกรุก ดังรูป



รูปที่ 4.13 ผลลัพธ์จากการสแกนด้วย Silent Mode

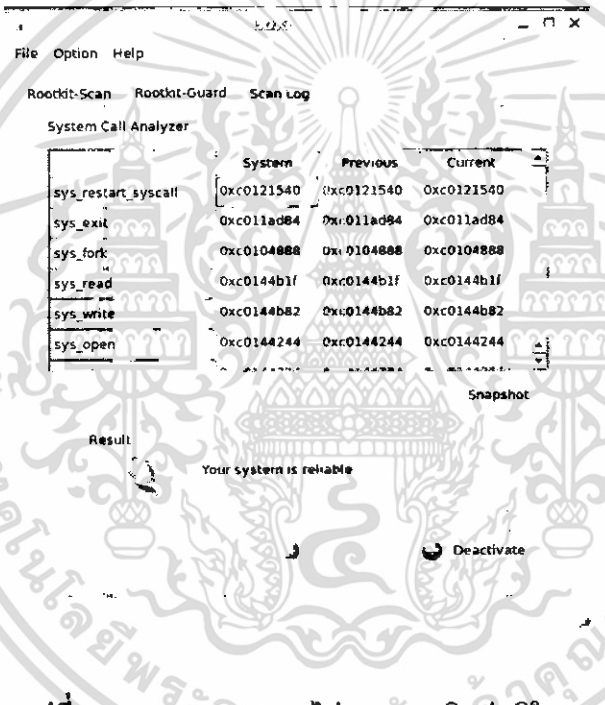
การแสดงผลลัพธ์ในส่วนของไฟล์พื้นฐานของรูตคิตจะแสดงเฉพาะไฟล์ที่ต้องสงสัย ดังรูป



รูปที่ 4.14 ผลลัพธ์จากการสแกนด้วย Silent Mode

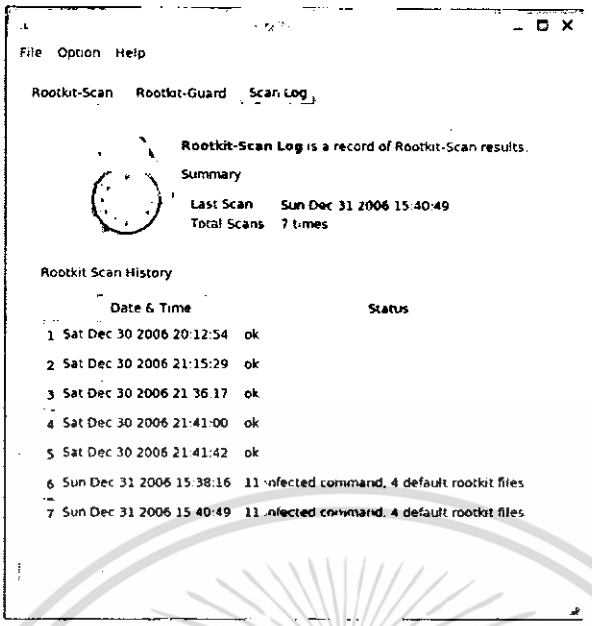
หลังจากสแกนด้วย Rootkit-Scan ซึ่งเป็นการค้นหาชุดโปรแกรมปกปิดการบุกรุกด้วยวิธีอ้างอิงแบบซิกเนเจอร์ แล้ว ผลลัพธ์ที่ได้คือ Loki สามารถค้นพบการเปลี่ยนแปลงของคำสั่งในระบบปฏิบัติการได้ และสามารถวิเคราะห์ได้ถูกต้องว่าระบบนั้นอาจจะถูกติดตั้งชุดโปรแกรมปกปิดการบุกรุก tuxkit อยู่ได้อย่างถูกต้อง

ขั้นตอนต่อไปเป็นการทดลองค้นหาชุดโปรแกรมปกปิดการบุกรุกด้วย Rootkit-Guard ซึ่งเป็นการตรวจสอบความถูกต้องของ System Call Table ซึ่งรูตคิตในโหมดยูสเซอร์ รวมทั้ง tuxkit ไม่ได้ไปทำการเปลี่ยนแปลง System Call Table แต่อย่างใด ดังนั้น Rootkit-Guard จึงไม่ตรวจพบความผิดปกติใดๆ ในระบบปฏิบัติการของผู้ใช้งาน ดังรูป



รูปที่ 4.15 Rootkit-Guard ไม่พบความผิดปกติใดๆ

หลังจากที่สแกนเสร็จเรียบร้อยแล้ว ข้อมูลการสแกนของ Rootkit-Scan จะถูกบันทึกลงล็อกไฟล์ เพื่อให้ผู้ใช้งานสามารถดูผลลัพธ์การสแกนในอดีตได้ ดังรูป



รูปที่ 4.16 แสดงล็อกไฟล์ของ Rootkit-Scan

4.4 การทดลองติดตั้งชุดโปรแกรมปกปิดการบุกรุกแบบเคอร์นัลโหมด ซึ่งจะมีการเปลี่ยนแปลงตารางซีสเต็มคอล และตรวจสอบด้วยชุดโปรแกรมปกปิดการบุกรุก (Lokd)

ระบบตรวจสอบชุดโปรแกรมปกปิดการบุกรุก (Loki) สามารถทำการตรวจสอบชุดโปรแกรมปกปิดการบุกรุกได้โดย ทำการเปรียบเทียบตำแหน่งของหน่วยความจำในตารางซีสเต็มคอล จากแหล่งข้อมูลเบื้องต้น โดยระบบจะทำการเปรียบเทียบกันทั้งหมดสองส่วนได้แก่

- ทำการเปรียบเทียบตารางซีสเต็มคอล จากเคอร์นัลของระบบ กับตารางซีสเต็มคอล ณ เวลาปัจจุบันและจากผู้ใช้
- ทำการเปรียบเทียบตารางซีสเต็มคอล จากผู้ใช้ กับตารางซีสเต็มคอล ณ เวลาปัจจุบัน

สำหรับการทดลองในส่วนนี้ จะทำการติดตั้งชุดโปรแกรมปกปิดการบุกรุกจำนวน 2 ชนิด ซึ่งได้แก่ Adore – NG และชุดโปรแกรมปกปิดการบุกรุกที่ได้สร้างขึ้น โดยใช้ Linux kernel module ในการทำงาน ชุดโปรแกรมนี้จะทำการเปลี่ยนแปลงตำแหน่งของซีสเต็มคอล (System Call) ที่ชื่อว่า sys_getdent64 () เพื่อซ่อนไฟล์บางอย่าง

4.4.1 ทดสอบด้วยชุดโปรแกรมปกปิดการบุกรุก Adore – NG

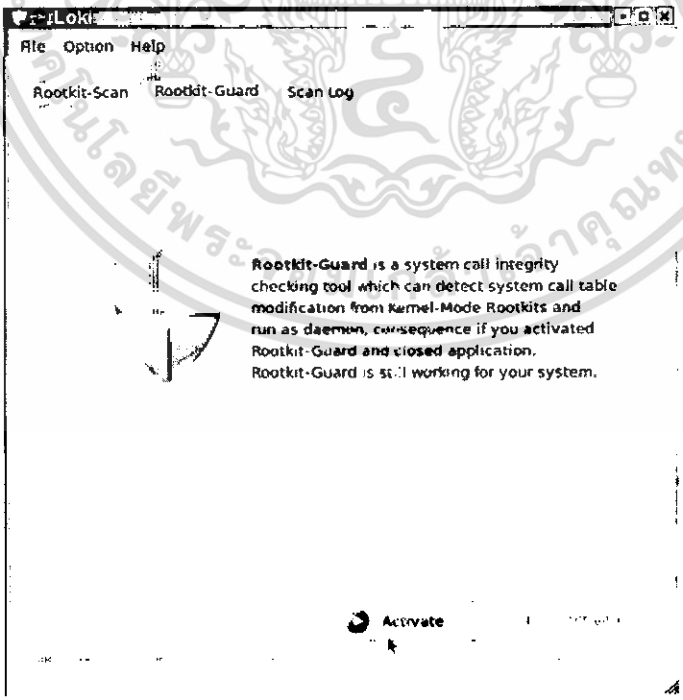
Adore – NG เป็นชุดโปรแกรมปกปิดการบุกรุกที่เป็นที่นิยมมาก โดย Adore – NG จะทำงานผ่านทาง Loadable kernel module โดยเข้าไปทำการเปลี่ยนแปลงเวอร์ชวลไฟล์ซีสเต็ม (Virtual File System (VFS)) หรือถ้าดูจากตารางซีสเต็ม จะมีชื่อว่า inode_operations ซึ่งเป็นตัวชี้ไปยังตำแหน่งหน่วยความจำของ VFS ของระบบ ซึ่ง VFS นี้เองเป็นส่วนที่จะถูกเรียกเมื่อมีการแสดงรายชื่อไฟล์ของคำสั่งต่างๆ บนระบบปฏิบัติการลินุกซ์

Adore – NG สามารถทำการซ่อนไฟล์ ซ่อนโปรเซส หรือแม้กระทั่งทำให้ไฟล์หายไปชนิดที่กู้คืนมาเลยไม่ได้ก็ยังสามารถทำได้ โดยเมนูการทำงานของ Adore – NG เป็นดังรูปที่ 4.17

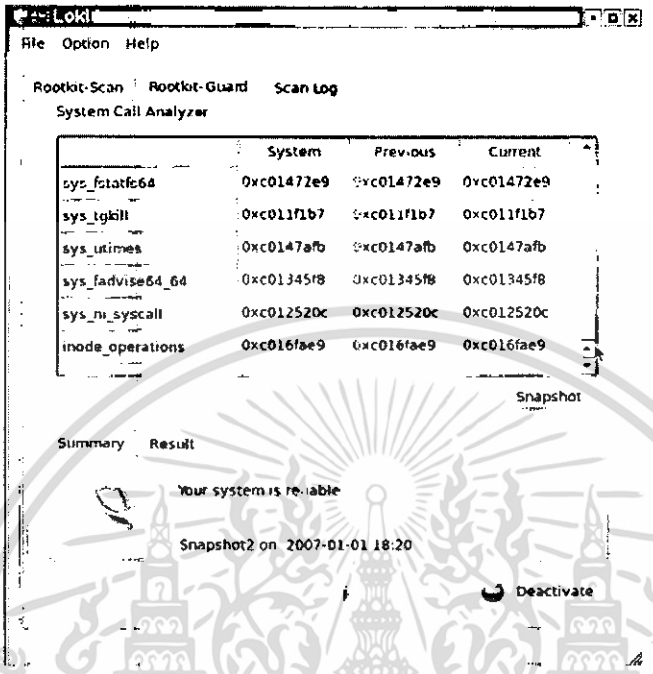
```
debian:~/adore-ng# ./ava
Usage: ./ava {h,u,r,R,i,v,U} [file or PID]

       I print info (secret UID etc)
       h hide file
       u unhide file
       r execute as root
       R remove PID forever
       U uninstall adore
       i make PID invisible
       v make PID visible

debian:~/adore-ng#
```

รูปที่ 4.17 แสดงเมนูการทำงานของโปรแกรมปกปิดการบุกรุก Adore – NG

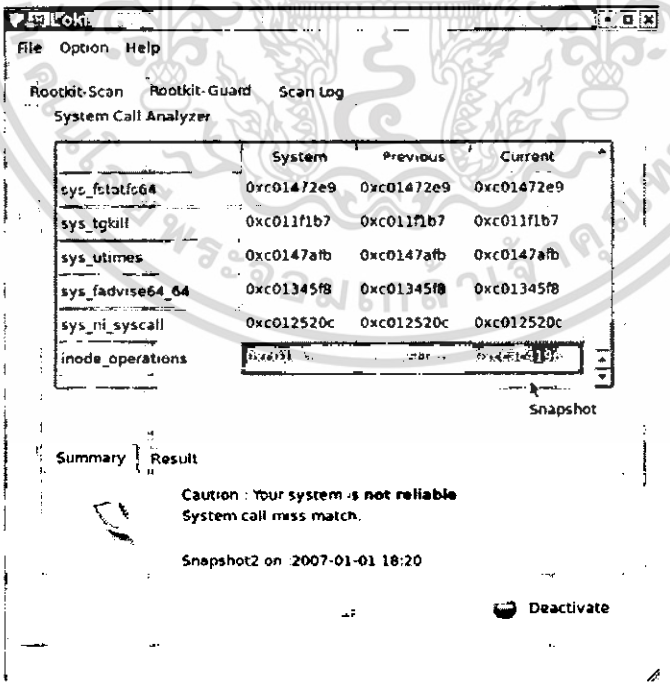
หลังจากที่ติดตั้งเรียบร้อยแล้ว เราจะทดสอบด้วยรันโปรแกรมตรวจสอบชุดโปรแกรมปกปิดการบุกรุก (Loki) ขึ้นมา จากนั้นทำการเรียกตัวเดมอน (Daemons) ขึ้นมาทำงาน และสุดท้ายติดตั้ง Adore – NG เพื่อทดสอบการทำงาน โดยจะสังเกตได้ว่า เมื่อติดตั้งชุดโปรแกรมปกปิดการบุกรุก Adore – NG ลงไปแล้ว ตำแหน่งตารางซีสเต็มในส่วน inode_operations จะไม่เหมือนกันโดยผลลัพธ์ที่ได้เป็นดังรูปที่ 4.18, รูปที่ 4.19, รูปที่ 4.20



รูปที่ 4.18 แสดงหน้าต่างหลักของโปรแกรมในส่วน Rootkit – Guard



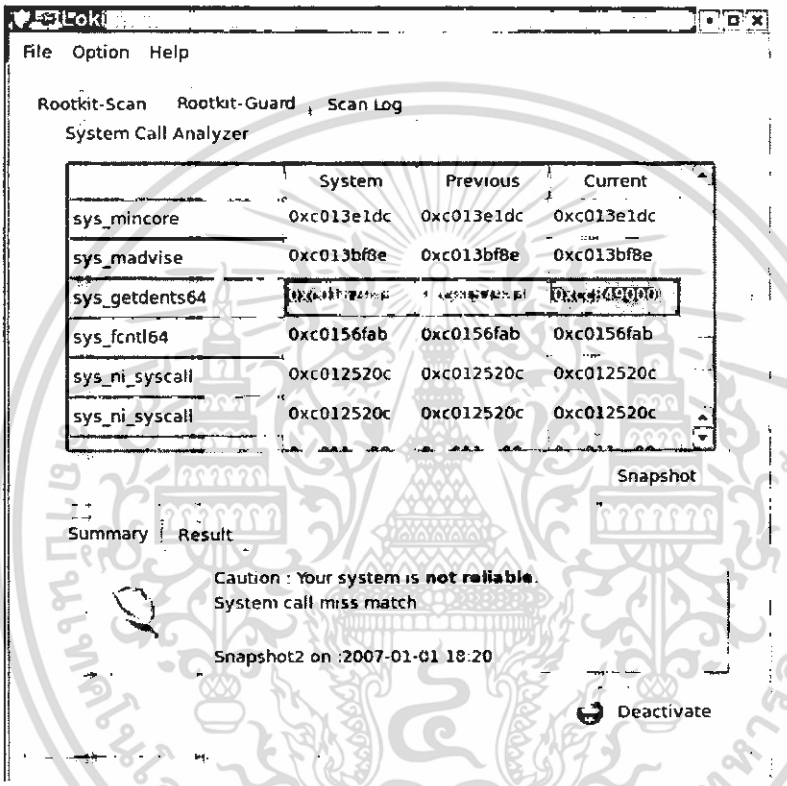
รูปที่ 4.19 ผลลัพธ์ที่ได้ก่อนการติดตั้ง Adore – NG



รูปที่ 4.20 ผลลัพธ์ที่ได้หลังการติดตั้ง Adore – NG

4.4.2 ทดสอบด้วยชุดโปรแกรมปกปิดการบุกรุกตัวอย่างที่ได้สร้างขึ้น เพื่อแก้ไข sys_getdent64 () ให้ทำการซ่อนไฟล์ในระบบ

ชุดโปรแกรมปกปิดการบุกรุกที่สร้างขึ้น จะไปเปลี่ยนแปลงซีสเต็มคอล เพื่อแก้ไข sys_getdent64 () ซึ่งเป็นคำสั่งที่ใช้ในการแสดงผลรายชื่อของไฟล์จากคำสั่ง ls เมื่อทำการติดตั้งชุดโปรแกรมนี้ผ่านทาง Linux Kernel Module โปรแกรม Loki สามารถตรวจพบความเปลี่ยนแปลงได้ทันที ดังรูปที่ 4.21



รูปที่ 4.21 แสดงการเปลี่ยนฟังก์ชัน sys_getdents64() ในตารางซีสเต็มคอล

บทที่ 5
บทวิจารณ์และสรุป

5.1 สรุปผลการพัฒนา

การพัฒนาระบบต่อต้านชุดโปรแกรมปกปิดการบุกรุกนั้น พัฒนาแบ่งออกเป็น 2 ส่วนหลัก คือ Rootkit-Scan และ Rootkit-Guard ซึ่งแต่ละส่วนมีความสามารถ ดังนี้

Rootkit-Scan มีความสามารถ ดังต่อไปนี้

1. สามารถตรวจสอบหารูตคิตในโหมดยูสเซอร์ (User-mode Rootkit)
2. สามารถตรวจสอบหาคำสั่งที่ถูกเปลี่ยนแปลงไปในระบบปฏิบัติการลินุกซ์
3. มีการทำงานเมื่อผู้ใช้ต้องการ
4. ใช้หลักการตรวจสอบที่อ้างอิงตามซิกเนเจอร์ (Signature Based)

Rootkit-Guard

1. ตรวจสอบรูตคิตในโหมดเคอร์นัล (Kernel Mode Rootkit)
2. ใช้หลักการตรวจสอบแบบอินทีกริตี้ (Integrity Checking) และอ้างอิงตามซิกเนเจอร์ โดยการเปรียบเทียบตำแหน่งของหน่วยความจำในตารางซีสเต็มคอล (System Call) จากแหล่งต่างๆ
3. ทำงานตามเวลาที่กำหนด

5.2 ปัญหาและอุปสรรค

ในช่วงระหว่างการดำเนินโครงการระบบต่อต้านชุดโปรแกรมปกปิดการบุกรุกนั้น ได้ประสบปัญหาต่างๆ ซึ่งรวบรวมได้เป็นข้อๆ ดังนี้

1. การใช้ระบบปฏิบัติการลีนุกซ์ในการทำโครงการนั้น ในช่วงแรกประสบปัญหาคือ ผู้พัฒนาไม่มีความรู้ ความชำนาญในการติดตั้งและพัฒนาโปรแกรมบนระบบปฏิบัติการลินุกซ์ ซึ่งแนวทางการแก้ไขปัญหานี้ก็คือ ศึกษาหาความรู้จากอินเตอร์เน็ต และแหล่งความรู้ต่างๆ ทั้งหนังสือ และสอบถามอาจารย์ที่ปรึกษา
2. ในการพัฒนาส่วนของ Rootkit Scan จำเป็นต้องหาข้อมูลซิกเนเจอร์ของชุดโปรแกรมบุกรุกที่มีอยู่ในปัจจุบัน ซึ่งมีจำนวนมาก ซึ่งทำให้เกิดความล่าช้าในการรวบรวมข้อมูลของชุดโปรแกรมปกปิดการบุกรุกแต่ละตัวว่ามีพฤติกรรมการทำงานอย่างไรบ้าง

5.3 แนวทางในการพัฒนาต่อ

5.3.1 พัฒนาส่วนของซิกเนเจอร์ที่ใช้ในการตรวจสอบรูตคิตในโหมดยูสเซอร์ที่เกิดขึ้นใหม่

5.3.2 พัฒนาอัลกอลิทึมที่ใช้ในการวิเคราะห์ และตรวจพบชุดโปรแกรมปกปิดการบุกรุก

5.3.3 พัฒนาการเก็บล็อกไฟล์ในระบบต่อต้านชุดโปรแกรมปกปิดการบุกรุกหลังจากที่ระบบถูกติดตั้งชุดโปรแกรมปกปิดการบุกรุก เพื่อใช้ในการกู้คืนระบบ

5.3.4 พัฒนาควบคู่ไปกับการทำงานของ ชุดโปรแกรมตอบสนองเมื่อเกิดการละเมิดความปลอดภัย (Incident Response) และ ชุดโปรแกรมรักษาความปลอดภัยของเครือข่าย (Network Security Suite)

5.4 ข้อสรุปและข้อเสนอแนะ

ในปัจจุบันนี้การบุกรุกหรือการโจมตีในระบบคอมพิวเตอร์มีเพิ่มมากขึ้นอย่างสูง ซึ่งผู้ที่บุกรุกเริ่มให้ความสนใจกับชุดโปรแกรมปกปิดการบุกรุกเพิ่มมากขึ้นเรื่อยๆ เนื่องมาจากความสามารถของชุดโปรแกรมปกปิดการบุกรุก ในการปิดบังไฟล์หรือโพรเซสที่ผู้บุกรุกได้สร้างขึ้นมาในการบุกรุก อีกทั้งผู้บุกรุกยังสามารถกลับเข้ามาบุกรุกได้อีกโดยที่ผู้ใช้งานไม่ทราบ หรือสามารถใช้เครื่องของเหยื่อนั้นเป็นฐานในการโจมตีระบบอื่นๆ โดยที่เหยื่อนั้นไม่สามารถทราบได้เลย ซึ่งอาจจะใช้เวลาเป็นเดือนหรืออาจใช้เวลาเป็นปีในการตรวจพบ และถือเป็นเรื่องใหญ่ในปัจจุบัน

ด้วยเหตุผลข้างต้น การตระหนักถึงความอันตรายของชุดโปรแกรมปกปิดการบุกรุกนั้นเป็นสิ่งที่สำคัญอย่างยิ่ง ซึ่งผู้ใช้ทั่วไปมักจะยังไม่ทราบถึงความหมาย และอันตรายของชุดโปรแกรมปกปิดการบุกรุก ดังนั้นการเผยแพร่และให้ความเข้าใจในการทำงานและอันตรายของชุดโปรแกรมปกปิดการบุกรุก รวมถึงการติดตั้งโปรแกรม หรือระบบที่สามารถตรวจพบและป้องกันชุดโปรแกรมปกปิดการบุกรุกได้ จะช่วยให้ระบบคอมพิวเตอร์มีความปลอดภัยมากยิ่งขึ้น

บรรณานุกรม

[1] **Daniel P. Bovet, Marco Cesati.** *Understanding the Linux Kernel, 2nd Edition*, O'Reilly.

[2] **Crutcher Dunnavant.** The Journeyman's Guide to Hacking Linux: version 0.1.1 [URL] http://bama.ua.edu/~dunna001/journeyman/html/book1.htm

[3] **Peter Jay Salzman. 2001.** The Linux Kernel Module Programming Guide [URL] http://www.tldp.org/LDP/lkmpg/2.6/html/index.html

[4] **Greg Hoglund, James Bulter.** Subverting the windows kernel Rootkits. Addison Wesley.

[5] **Ed Skoudis.** *Malware* : Fighting Malicious Code , Printice Hall , Peason Education.

[6] **Jasmin Blanchette, Mark Summerfield. 2006.** *C++ GUI Programming with Qt 4*, Prentice Hall.

[7] **Alan Ezust, Paul Ezust. 2006.** *An Introduction to Design Patterns in C++ with Qt 4*, Prentice Hall.

[8] Linux Kernel Rootkits: http://la-samhna.de/library/rootkits/basics.html

[9] http://www.rootkit.com

[10]http://www.thaicert.nectec.or.th/paper/unix_linux/hacked.php

[11]http://en.wikipedia.org/wiki/Rootkit

ภาคผนวก

ภาคผนวก ก.
คู่มือการติดตั้งระบบต่อต้านชุดโปรแกรมปกปิดการบุกรุก

ภาคผนวก ก.
คู่มือการติดตั้งระบบต่อต้านชุดโปรแกรมปกปิดการบุกรุก

ระบบต่อต้านชุดโปรแกรมปกปิดการบุกรุก (Loki) มีความต้องการทางด้านฮาร์ดแวร์ และความต้องการทางด้านซอฟต์แวร์ที่ใช้ในการทำงาน ดังนี้

ความต้องการทางด้านฮาร์ดแวร์

- CPU: ความเร็วอย่างต่ำ 900 MHz
- Hard disk: อย่างต่ำ 1 GB
- RAM: อย่างต่ำ 128 MB

ความต้องการทางด้านซอฟต์แวร์

- ระบบปฏิบัติการ: UNIX, Linux Debian
- X – Window System ; KDE Desktop Environment
- Sun JAVA Runtime Environment 5.0 หรือ JVM อื่นๆ ที่ใช้งานร่วมกับ Bouncy Castle Java Cryptographic Service Provider
- Qt GUI Library Version 4.0

ในการทำงานของชุดโปรแกรมปกปิดการบุกรุก เพื่อที่จะใช้ชุดโปรแกรมปกปิดการบุกรุกสามารถทำงานได้ จำเป็นต้องทำการติดตั้งไลบรารีที่ชุดโปรแกรมปกปิดการบุกรุกต้องการในการทำงาน ดังนี้

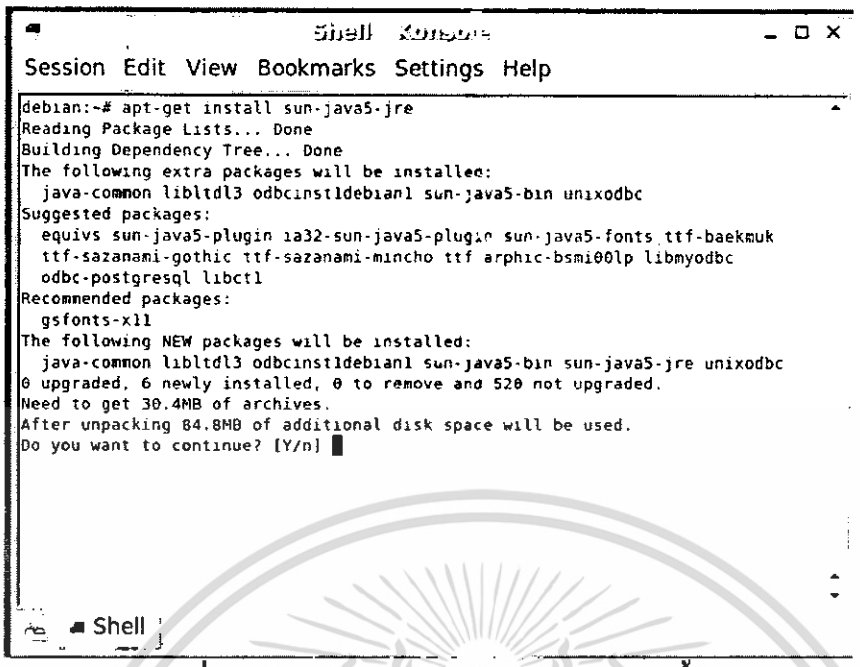
1. **ติดตั้งระบบปฏิบัติการ Linux Debian**

   โดยทำการเลือก KDE เป็น X – Window system หลักของระบบ

2. **การติดตั้ง JAVA Runtime Environment 5.0**

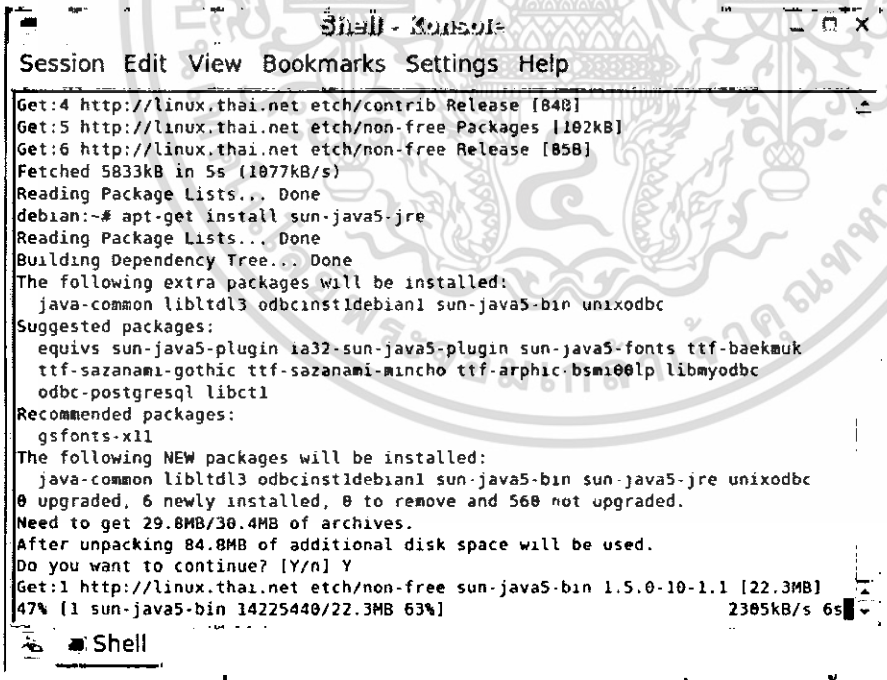
   ระบบต่อต้านชุดโปรแกรมปกปิดการบุกรุก มีการใช้ภาษา Java ในการพัฒนา ดังนั้นเพื่อให้ระบบสามารถทำงานได้ จะต้องติดตั้งไลบรารีในการทำงานภาษา Java ซึ่งมีขั้นตอนการติดตั้ง ดังนี้

   1.1 พิมพ์คำสั่ง apt-get install sun-java5 เพื่อติดตั้งไลบรารี ดังรูป



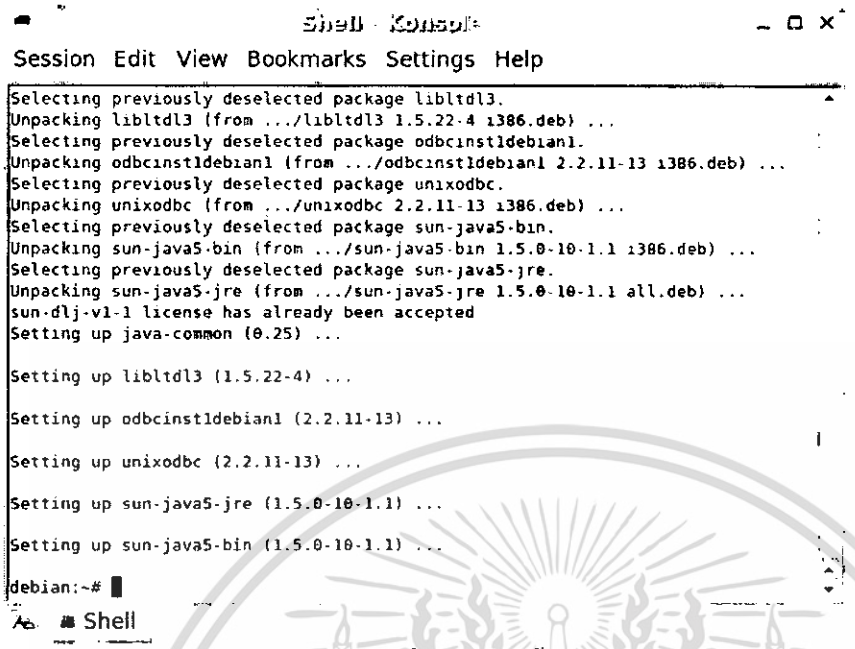
รูปที่ 1 ระบบสอบถามความต้องการในการติดตั้ง

หลังจากนั้นระบบจะแสดงรายละเอียดของไลบรารีที่ต้องการติดตั้ง และสอบถามว่าต้องการติดตั้งหรือไม่ ให้ตอบ Y เพื่อติดตั้ง



รูปที่ 2 ดาวน์โหลด Source จากอินเตอร์เน็ตเพื่อทำการติดตั้ง

ไลบรารีจะถูกติดตั้งลงระบบปฏิบัติการ โดยใช้พื้นที่เก็บข้อมูลประมาณ 84.8 MB



รูปที่ 3 การติดตั้ง JRE 5.0

4. Qt library version 4

ระบบต่อต้านชุดโปรแกรมปกปิดการบุกรุก (Loki) นั้นพัฒนาอินเทอร์เฟสขึ้นด้วย Qt Designer ดังนั้น เพื่อให้ระบบสามารถทำงานได้ จำเป็นต้องติดตั้งไลบรารี ซึ่งมีขั้นตอนดังนี้

4.1 พิมพ์คำสั่ง apt-get install libqt4-gui

5. g++ compiler

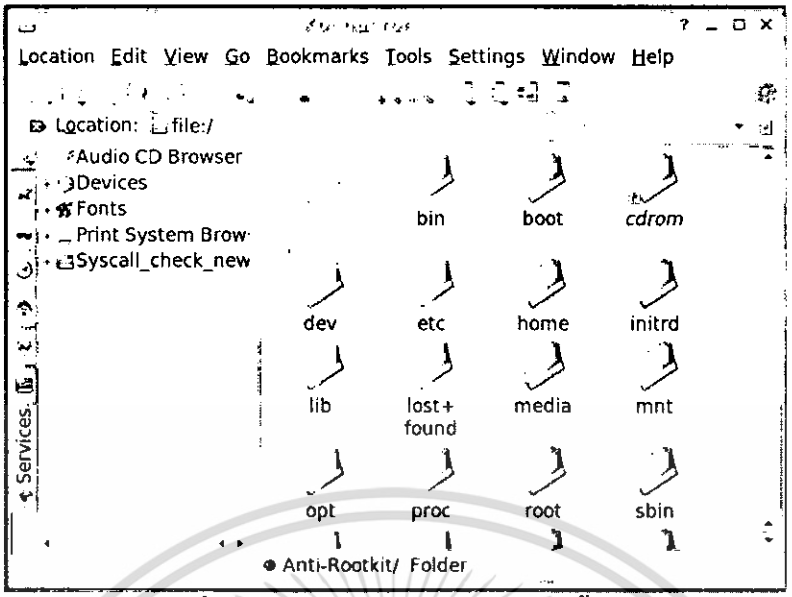
ในการพัฒนาอินเทอร์เฟสของชุดโปรแกรมปกปิดการบุกรุก (Loki) ใช้ภาษา C++ ดังนั้น จะต้องติดทั้งคอมไพเลอร์ เพื่อให้ระบบสามารถทำงานได้ ซึ่งมีขั้นตอนดังนี้

5.1 พิมพ์คำสั่ง apt-get install gcc

6. Loki

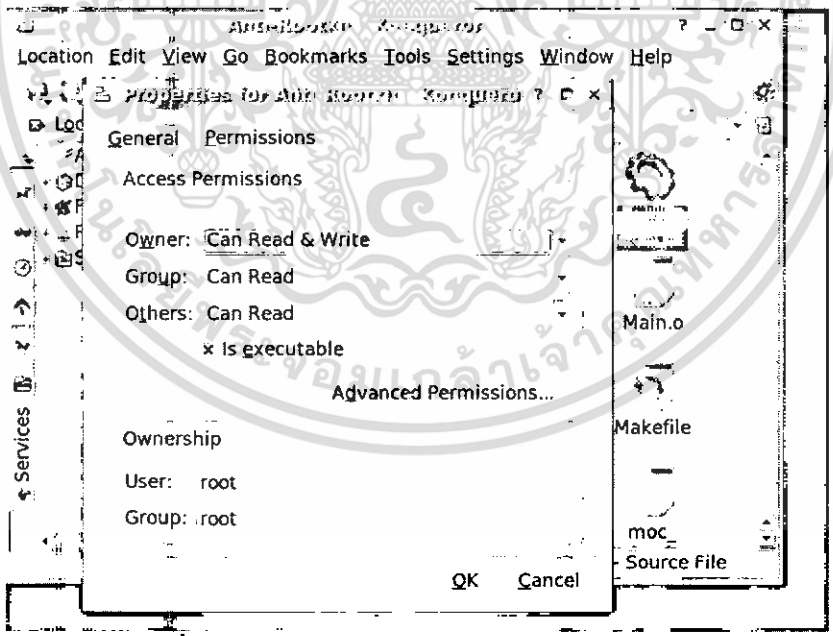
การติดตั้ง Loki มีขั้นตอน ดังนี้

6.1 นำโฟลเดอร์ Anti-Rootkit ติดตั้งที่ / (root directory)



รูปที่ 4 นำโฟลเดอร์ Anti-Rootkit ติดตั้งที่ /

6.2 ตรวจสอบ Permission ของไฟล์ configfile.sh เพื่อให้ระบบสามารถทำงานได้ โดยการคลิกขวาที่ไฟล์ เลือก Properties และเลือกที่แท็ป Permissions หลังจากนั้นปรับ ว่า Is executable ถูกเลือกอยู่หรือไม่ ถ้าไม่ได้ถูกเลือกอยู่ ให้ทำการเลือก



รูปที่ 5 การปรับ Permission ของไฟล์ configfile.sh

6.3 ถ้าต้องการให้ระบบทำงานเป็น Daemons ของระบบให้รันไฟล์ที่ชื่อว่า asdaemon.sh

ภาคผนวก ข.
คู่มือการใช้งานระบบต่อต้านชุดโปรแกรมปกปิดการบุกรุก

ภาคผนวก ข.
คู่มือการใช้งานระบบต่อต้านชุดโปรแกรมปกปิดการบุกรุก

หลังจากที่ได้ทำการติดตั้งองค์ประกอบต่างๆของระบบต่อต้านชุดโปรแกรมปกปิดการบุกรุก (Loki) เป็นที่เรียบร้อยแล้ว ขั้นตอนการใช้งานระบบต่อต้านชุดโปรแกรมปกปิดการบุกรุกเพื่อใช้ในการตรวจสอบความผิดปกติของระบบ และค้นหาชุดโปรแกรมปกปิดการบุกรุก (Rootkit) มีขั้นตอนดังนี้

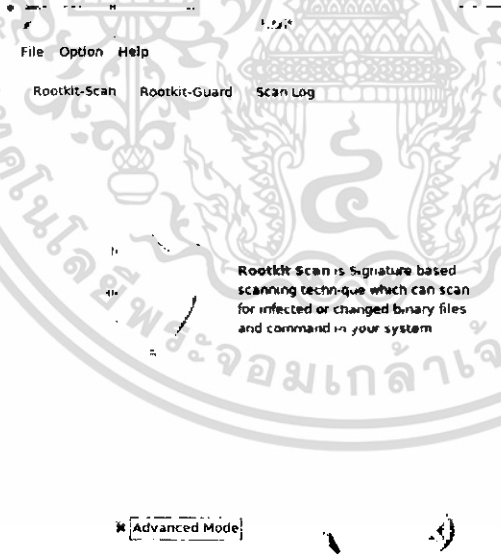
การเปิดระบบต่อต้านชุดโปรแกรมปกปิดการบุกรุก (Loki)

1. เข้าสู่ระบบโดยเป็นผู้ดูแลระบบ Root
2. เปิด Konsole เพื่อเข้าสู่โหมด command line และพิมพ์คำสั่ง cd /Anti-Rootkit เพื่อเข้าสู่ไดเร็กทอรีของโปรแกรม
3. พิมพ์คำสั่ง ./Anti-Rootkit เพื่อเปิดโปรแกรม ดังรูป

```
debian:~# cd /Anti-Rootkit
debian:/Anti-Rootkit# ./Anti-Rootkit
```

รูปที่ 1 เปิดโปรแกรมด้วยคำสั่ง ./Anti-Rootkit

4. เมื่อเปิดโปรแกมขึ้นมาทำงานแล้ว จะเข้าสู่หน้าจอหลักของโปรแกรม ดังรูปที่ 2



รูปที่ 2 หน้าจอหลักของโปรแกรม

Loki มีการทำงาน 2 ฟังก์ชันหลัก คือ Rootkit-Scan และ Rootkit-Guard ซึ่งจะแบ่งการทำงานตามแท็ปในรูปที่ 2 ซึ่งแท็ปแรกและแท็ปที่สาม นั้นจะเป็นการทำงานของ Rootkit-Scan และ แท็ปที่สองนั้นจะเป็นการทำงานของ Rootkit-Guard

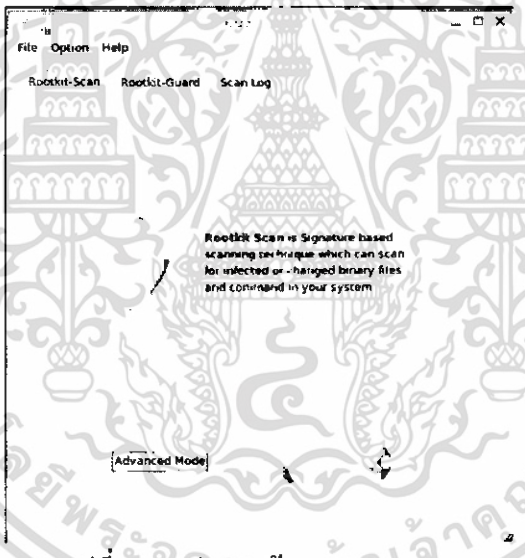
Rootkit-Scan

Rookit-Scan นั้นจะเป็นส่วนของโปรแกรมที่ทำหน้าที่ตรวจสอบไบนารีไฟล์ของคำสั่งในระบบปฏิบัติการ และทำการค้นหาไฟล์มาตรฐานของชุดโปรแกรมปกปิดการบุกรุกแต่ละชนิด ซึ่งจะทำงานก็ต่อเมื่อผู้ใช้คลิกที่ปุ่ม Scan (รูปแว่นขยาย) ซึ่งในการทำงานนั้น มีการทำงาน 2 โหมดคือ

- Silent Mode: เป็นโหมดที่จะทำการแสกน และแสดงผลลัพธ์เฉพาะสิ่งผิดปกติที่เกิดขึ้นในระบบปฏิบัติการของผู้ใช้งาน
- Advanced Mode: เป็นโหมดที่จะแสดงผลลัพธ์จากการแสกนทั้งหมด ทั้งคำสั่งที่มีผลลัพธ์ปกติ และผิดปกติในระบบปฏิบัติการของผู้ใช้งาน

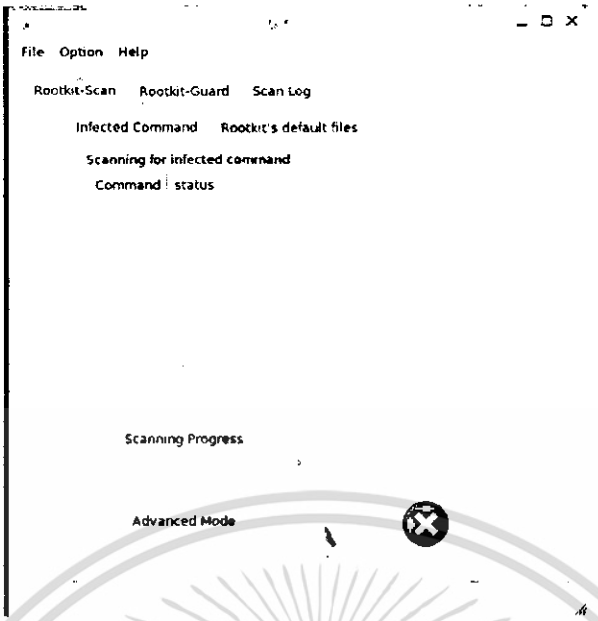
การแสกนด้วย Silent Mode

ในการแสกนด้วย Silent Mode ให้เอาเครื่องหมายเลือกที่ Advanced Mode ออก ดังรูป



รูปที่ 3 การทำงานด้วย Silent Mode

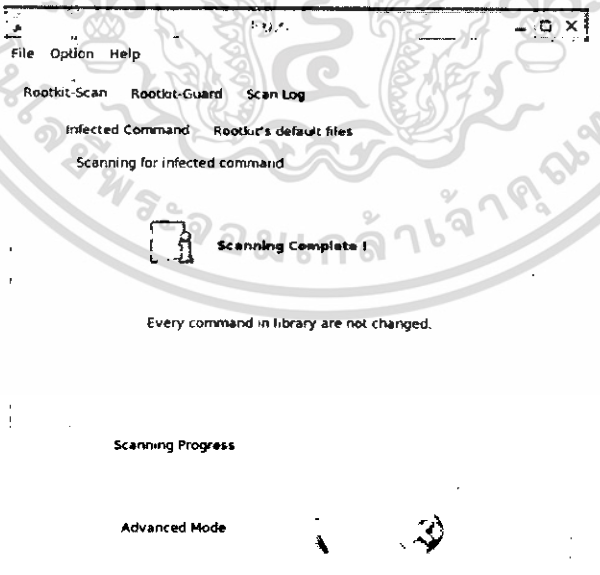
ผู้ใช้งานสามารถเริ่มต้นแสกนด้วยการคลิกเลือกที่ปุ่มแสกน หลังจากนั้นโปรแกรมจะเริ่มต้นทำงานในการตรวจหาชุดโปรแกรมปกปิดการบุกรุก ดังรูปที่ 4



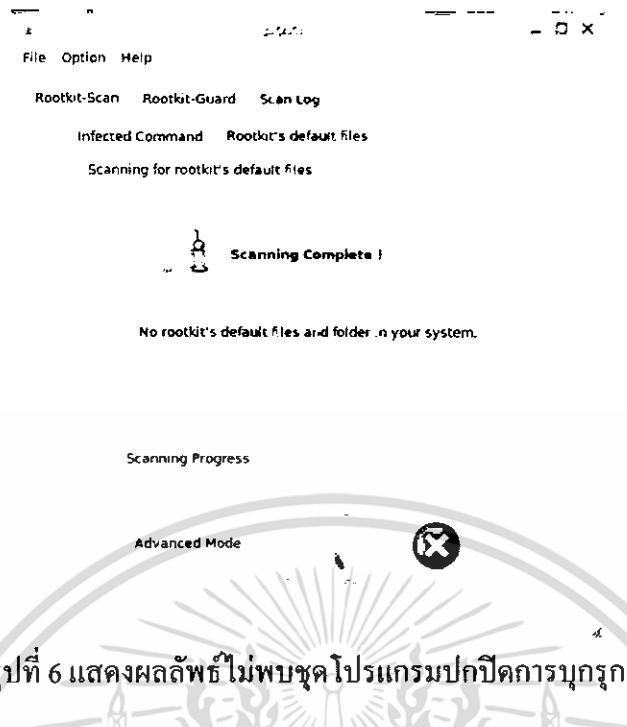
รูปที่ 4 การแสกนด้วย Silent Mode

หลังจากแสกนเรียบร้อยแล้ว โปรแกรมจะแสดงผลลัพธ์ต่อผู้ใช้งาน โดยผลลัพธ์จะแบ่งออกเป็น 2 แท็ป คือ Infected Command จะแสดงผลลัพธ์ของคำสั่งในระบบปฏิบัติการที่ถูกเปลี่ยนแปลงโดยชุดโปรแกรมปกปิดการบุกรุก และ ผลลัพธ์ส่วนที่สองคือ Rootkit's Default File จะแสดงชื่อของ Rootkit ที่ต้องสงสัยว่าจะถูกติดตั้งลงในระบบของผู้ใช้งาน

ในกรณีที่ไม่พบสิ่งผิดปกติบนระบบปฏิบัติการของผู้ใช้ โปรแกรมจะแสดงผลลัพธ์ ดังรูปที่ 5 และรูปที่ 6



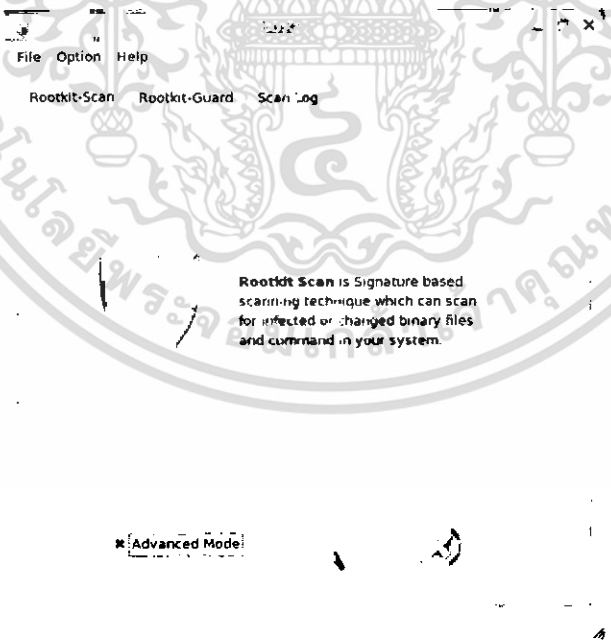
รูปที่ 5 แสดงผลลัพธ์ไม่พบคำสั่งใดถูกเปลี่ยนแปลง



รูปที่ 6 แสดงผลลัพธ์ไม่พบชุดโปรแกรมปกปิดการบุกรุก

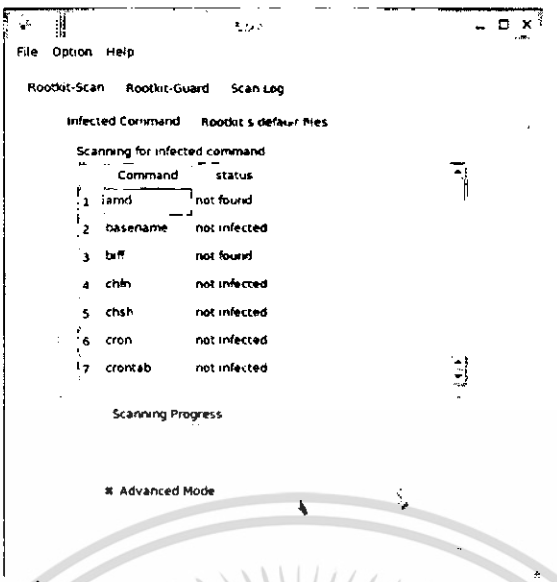
การแสกนด้วย Advanced Mode

ในการแสกนด้วย Silent Mode ให้คลิกเลือกเครื่องหมาย Advanced Mode หรือเมื่อเปิดโปรแกรมขึ้นมาจะถูกเลือกโดยปริยาย ดังรูป



รูปที่ 7 การทำงานด้วย Advanced Mode

เริ่มต้นแสกนด้วยการคลิกเลือกที่ปุ่มแสกน หลังจากนั้นโปรแกรมจะเริ่มต้นทำงาน และแสดงผลลัพธ์ไปพร้อมกับการทำงาน ซึ่งเมื่อแสกนเสร็จเรียบร้อยแล้ว จะแสดงผลลัพธ์ ดังรูป

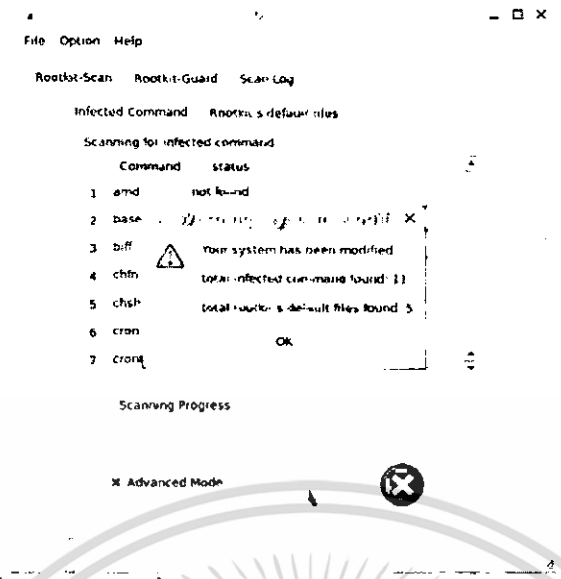


รูปที่ 8 ผลลัพธ์จากการแสกนด้วย Advanced Mode

ซึ่งผลลัพธ์จากการสแกน จะแบ่งออกเป็น 4 แบบ คือ

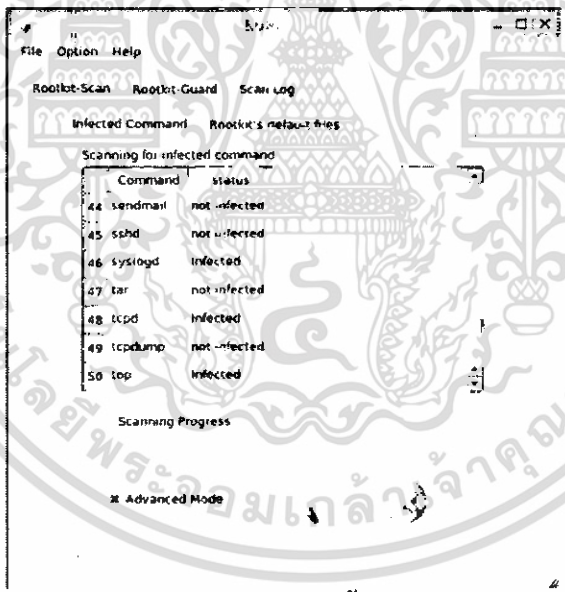
- not found คือ ไม่พบคำสั่งนั้นในระบบปฏิบัติการ
- not infected คือ คำสั่งในระบบปฏิบัติการไม่มีการเปลี่ยนแปลงจากชุดโปรแกรมปกปิดการบุกรุก (สามารถเชื่อถือผลลัพธ์จากคำสั่งนั้นๆ ได้ เช่น ผลลัพธ์จากคำสั่ง ls หรือ netstat เป็นต้น)
- infected คือ คำสั่งนั้นๆ ถูกเปลี่ยนแปลงโดยชุดโปรแกรมปกปิดการบุกรุก (ไม่ควรเชื่อถือผลลัพธ์ของคำสั่งๆ นั้น เช่น คำวสั่ง ls อาจจะถูกแก้ไขให้ไม่แสดงชื่อไฟล์และไดเร็กทอรีที่เก็บไฟล์ของชุดโปรแกรมปกปิดการบุกรุก)
- not test คือ ไม่สามารถตรวจสอบคำสั่งนั้นๆ ได้ ซึ่งอาจจะเกิดจากสิทธิในการเข้าถึงไฟล์หรือคำสั่งของผู้ใช้งาน

ในกรณีที่มีการตรวจพบว่าคำสั่งในระบบปฏิบัติการของผู้ใช้งานหรือตรวจพบไฟล์พื้นฐานของชุดโปรแกรมปกปิดการบุกรุก จะมีผลลัพธ์แจ้งเตือนผู้ใช้ ดังรูปที่ 9 และรูปที่ 10



รูปที่ 9 แจ้งเตือนเมื่อตรวจพบชุดโปรแกรมปกปิดการบุกรุก

ผู้ใช้สามารถเลือกดูรายละเอียดคำสั่งที่ตรวจสอบได้ภายในแท็ปผลลัพธ์ของ Infected Command ดังรูป

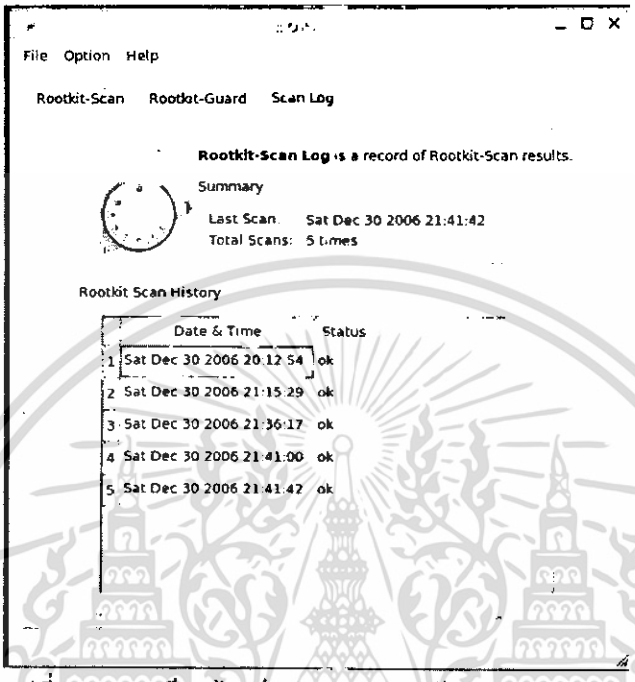


รูปที่ 10 แสดงผลลัพธ์ว่าคำสั่ง syslogd นั้นถูกเปลี่ยนแปลง (infected)

ผลลัพธ์จากการแสกนด้วย Advanced Mode จะแบ่งออกเป็น 2 ส่วนเช่นเดียวกับการแสกนด้วย Silent Mode แต่ผลลัพธ์จะแสดงทุกคำสั่งที่ทำการตรวจสอบไม่ว่าผลลัพธ์นั้นจะผิดปกติหรือไม่ก็ตาม ซึ่งการทำงานด้วยโหมดนี้จะช่วยให้ผู้ใช้ที่เข้าใจในระบบปฏิบัติการลินุกซ์ตรวจสอบและเห็นผลลัพธ์ได้อย่างชัดเจน

Rootkit-Scan Log

Rootkit-Scan Log จะเป็นการแสดงสถิติในการสแกนด้วย Rootkit-Scan ของผู้ใช้งาน เพื่อที่ให้ผู้ใช้งานสามารถตรวจสอบผลการสแกนในอดีต โดยที่ผู้ใช้งานสามารถเลือกไปที่แท็ปที่สาม ดังรูป

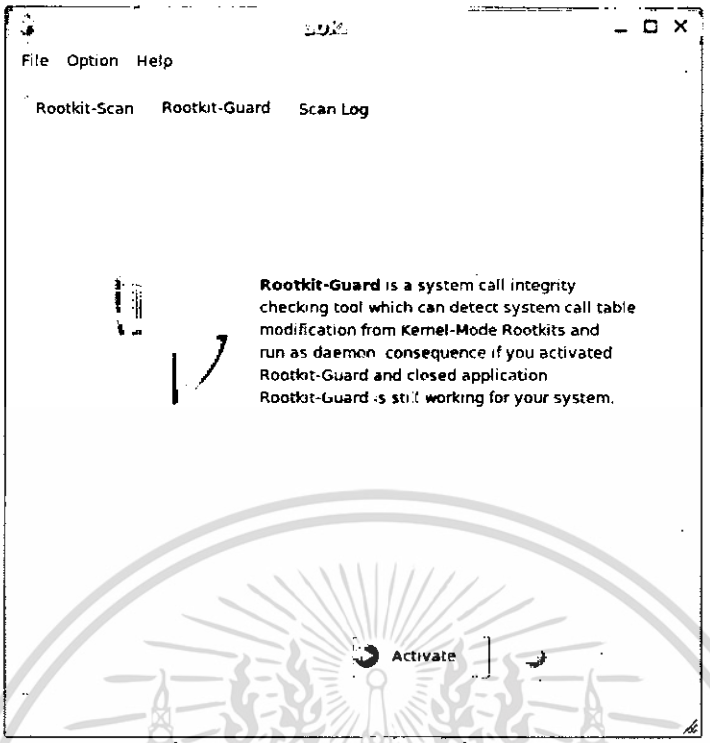


รูปที่ 11 แสดงล็อกไฟล์ของการสแกนด้วย Rootkit-Scan

Rootkit-Guard

ส่วนของ Rootkit-Guard เป็นส่วนที่ใช้วิเคราะห์หาชุดโปรแกรมปกปิดการบุกรุกที่ใช้เคอร์แนลโมดูลทำงาน โดยโปรแกรมจะตรวจสอบจากตาราง System Call ว่ามีการเปลี่ยนแปลงไปหรือไม่ ถ้ามีการเปลี่ยนแปลงโปรแกรมจะทำการเตือนทันที ซึ่งถ้าการเปลี่ยนแปลงนั้นตรงกับ Signature ของโปรแกรมที่มีอยู่จะแสดงรายชื่อของชุดโปรแกรมปกปิดการบุกรุกขึ้นมา แต่ถ้าไม่ จะเป็นการเตือนเพื่อให้ผู้ใช้ได้ทราบ

โดยการทำงานในส่วนนี้ จะทำงาน เป็นแบบ Daemon เนื่องจากโปรแกรมทำงานอยู่เบื้องหลังอยู่ตลอดเวลา และผู้ใช้สามารถเลือกว่าจะให้ทำงานอยู่ตลอดหรือไม่ โดยการกดเลือกที่ปุ่ม Activate เพื่อให้ Rootkit-Guard นั้นมีการทำงานอยู่เบื้องหลังตลอดเวลา และกดเลือกที่ปุ่ม Deactivate เพื่อให้ Rootkit-Guard หยุดการทำงาน ดังรูปที่ 12

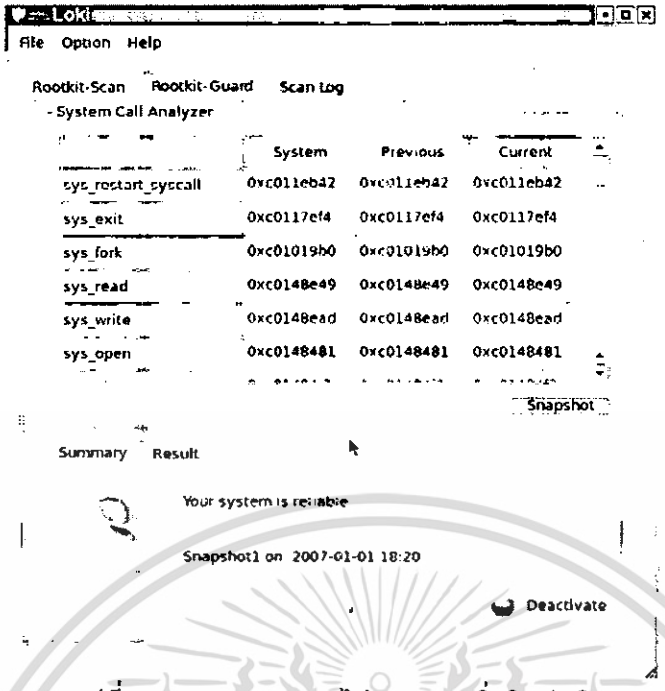


รูปที่ 12 Rootkit-Guard ขณะที่ไม่ทำงาน

เมื่อ Rootkit-Guard ทำงานแล้วจะมีตารางแสดงการตรวจสอบและวิเคราะห์ System Call Table ที่มีในระบบ และมีผลลัพธ์แสดงให้ผู้ใช้งานได้ทราบ โดยมีส่วนต่างๆ ดังนี้

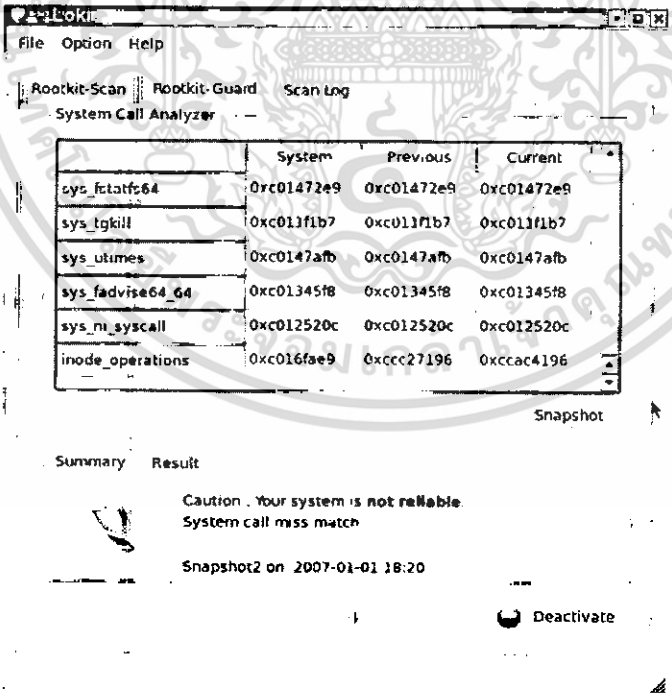
- System Call Table Analyzer แสดงผลตารางของตำแหน่งของหน่วยความจำของ System Call Table ที่มีอยู่ในระบบ โดยจะแบ่งออกเป็น 3 ส่วนซึ่งได้แก่ System ที่ได้มาจาก Symbol map ของเคอร์นัลในระบบ , Previous ซึ่งได้มาจากการที่ User สั่ง Snapshot ระบบ และสุดท้าย Current ซึ่งเป็นตำแหน่งของ System Call Table ของระบบ ณ เวลาปัจจุบัน
- ปุ่ม Snapshot ไว้ให้ผู้ใช้บันทึกตำแหน่งของหน่วยความจำใน System Call Table ณ เวลาที่ต้องการ เพื่อไว้เปรียบเทียบต่อไปในอนาคต
- แท็บ Summary จะรายงานระบบโดยรวมขณะนั้นของ System Call Table
- แท็บ Result เป็นผลลัพธ์ที่ได้จากโปรแกรม Loki เมื่อตรวจพบความผิดปกติในส่วน System Call Table ของระบบ
- ปุ่ม Activate , Deactivate มีไว้เพื่อปิดหรือเปิด Daemons ของระบบ

ซึ่งในกรณีที่ตาราง System Call นั้น ไม่มีการถูกเปลี่ยนแปลงโดยชุดโปรแกรมปกปิดการบุกรุก จะแสดงผลลัพธ์ดังรูปที่ 13



รูปที่ 13 Rootkit-Guard ไม่ตรวจพบสิ่งผิดปกติ

ในกรณีที่ Rootkit-Guard ตรวจพบความเปลี่ยนแปลงของตาราง System call จะมีการแจ้งเตือนให้ผู้ใช้งานทราบ ดังรูปที่ 14 ซึ่งจะเห็นได้ว่าตำแหน่งของหน่วยความจำ inode_operations มีการเปลี่ยนแปลง



รูปที่ 14 Rootkit-Guard ตรวจพบว่าตาราง System call ถูกเปลี่ยนแปลง