ระบบตรวจจับมัลแวร์เชิงพฤติกรรมสำหรับระบบปฏิบัติการวินโดวส์

BEHAVIOR-BASED MALWARE DETECTION SYSTEM FOR WINDOWS

รัตสกานต์ ศรีสวัสดิ์
รุ่งธรรม รอดระวังภัย
วรพรรณ ปาริยพันธ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2553

ปริญญานิพนธ์ปีการศึกษา 2553

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบตรวจจับมัลแวร์เชิงพฤติกรรมสำหรับระบบปฏิบัติการวินโดวส์

BEHAVIOR-BASED MALWARE DETECTION SYSTEM FOR WINDOWS

ผู้จัดทำ

1. นายรัตสกานต์ ศรีสวัสดิ์ รหัสนักศึกษา 50011303
2. นายรุ่งธรรม รอดระวังภัย รหัสนักศึกษา 50011309
3. นางสาววรพรรณ ปาริยพันธ์ รหัสนักศึกษา 50011366

อาจารย์ที่ปรึกษา

(อาจารย์อัครเดช วัชระภูพงษ์)

ระบบตรวจจับมัลแวร์เชิงพฤติกรรมสำหรับระบบปฏิบัติการวินโดวส์

นายรัตสกานต์ ศรีสวัสดิ์ 50011303
นายรุ่งธรรม รอดระวังภัย 50011309
นางสาววรพรรณ ปาริยพันธ์ 50011366
อาจารย์อัครเดช วัชระภูพงษ์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2553

บทคัดย่อ

ในปัจจุบันโปรแกรมตรวจจับมัลแวร์จะเป็นแบบ Signature-Based ซึ่งจะทำการตรวจสอบกับข้อมูลเฉพาะของมัลแวร์แต่ละตัว แต่การตรวจจับแบบนี้เริ่มไม่ได้ผลเนื่องจากมีการ Polymorphic ตัวเองของมัลแวร์ จึงทำให้รูปแบบข้อมูลของมัลแวร์เปลี่ยนไปแต่พฤติกรรมยังคงเหมือนเดิมและทำให้ต้องอัปเดตข้อมูลมัลแวร์ใหม่ๆอยู่เสมอ เพื่อแก้ปัญหาดังกล่าวจึงได้สร้างการตรวจจับแบบ Behavior-Based ซึ่งเป็นการตรวจสอบกับพฤติกรรมการทำงานของมัลแวร์ โดยจะแบ่งการทำงานของโปรแกรมเป็นดังนี้ ส่วนแรกจะเป็นส่วนที่ตรวจสอบการทำงานหรือเปลี่ยนแปลงค่าต่างๆของระบบ ได้แก่ File Monitor, Process Monitor, Registry Monitor, Port Monitor, API hooking และ WinPcap เพื่อตรวจดูว่ามีการกระทำใดเกิดขึ้นภายในระบบบ้าง ส่วนที่สองคือส่วนวิเคราะห์พฤติกรรมมัลแวร์ โดยจะนำข้อมูลที่ได้จาก Monitor ต่างๆมาวิเคราะห์เปรียบเทียบกับฐานข้อมูลพฤติกรรมของมัลแวร์ โดยในการระบุพฤติกรรมมัลแวร์นั้นจะใช้หลักการ Misuse Detection

Behavior-based Malware Detection System for Windows

Mr. Ratsakan Srisawat 50011303

Mr. Rungtham Rodrawangpai 50011309

Ms. Woraphan Pariyaphan 50011366

Mr. Akkradach Watcharapupong Advisor

Academic Year 2010

ABSTRACT

Nowadays, most of anti-malware programs are signature-based. They examine threats by comparing with malware's signature but it does not work well because some malware can polymorphic themselves. Therefore malware's signature has been changed but their behavior still as it was. So we need to update malware's signature regularly. To solve this problem we need to develop behavior-based malware detection that examine by behaviors existing on OS. Our system consist of 2 sub-systems the first is Monitors: File, Process, Registry, Port monitor, API hooking and WinPcap to check state changes on the OS. The last one is analyzer for constructing behavioral models and analyzer for detection by principles of Misuse Detection.

กิตติกรรมประกาศ

โครงงานระบบตรวจจับมัลแวร์เชิงพฤติกรรมสำหรับระบบปฏิบัติการวินโดวส์นี้คงไม่อาจสำเร็จได้ด้วยดีหากไม่ได้รับการดูแลเอาใจใส่ การสนับสนุนและความร่วมมือจากหลายๆฝ่ายช่วยกันผลักดันให้โครงงานนี้ดำเนินไปด้วยดี และประสบความสำเร็จลุล่วงในที่สุด

ขอขอบคุณสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง และสาขาวิชาวิศวกรรมคอมพิวเตอร์ รวมถึงห้องวิจัยและพัฒนาการรักษาความปลอดภัยข้อมูล (Information Security Advisory Group: ISAG) และสถานศึกษาในอดีตสำหรับโอกาสดีๆทางการศึกษาที่เป็นแหล่งประสิทธิ์ประสาทวิชาทำให้ผู้จัดทำได้มีโอกาสในทุกวันนี้

ขอขอบพระคุณอาจารย์อัครเดช วัชระภูพงษ์ ซึ่งเป็นอาจารย์ที่ปรึกษาที่ให้คำแนะนำต่างๆ ชี้แนะข้อบกพร่องต่างๆและแนวทางในการศึกษา ตั้งคำถาม เอาใจใส่ อธิบายข้อข้องใจต่างๆในทุกๆด้าน และช่วยเหลือเสมอมา

และสุดท้ายขอขอบพระคุณบุคคลที่สำคัญที่สุดในชีวิตซึ่งก็คือ บิดา มารดา และผู้มีพระคุณอันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูข้าพเจ้ามาเป็นอย่างดี พร้อมให้โอกาสในการศึกษาอย่างเต็มที่และยังให้กำลังใจเสมอมา ข้าพเจ้าระลึกในพระคุณอันสุดประมาณและขอกราบขอบพระคุณมา ณ ที่นี้

รัตสกานต์ ศรีสวัสดิ์
รุ่งธรรม รอดระวังภัย
วรพรรณ ปาริยพันธ์

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

ตาราง **หน้า**

สารบัญรูป

รูป **หน้า**

สารบัญรูป (ต่อ)

สารบัญรูป (ต่อ)

บทที่ 1
บทนำ

1.1 ความเป็นมาของปัญหา

เนื่องจากโปรแกรมที่ใช้ตรวจจับมัลแวร์ในปัจจุบัน จะเป็นโปรแกรมแบบ Signature-based ในการตรวจจับ โดยมัลแวร์แต่ละตัวจะมีรหัสรูปแบบเฉพาะของตัวเองไม่ซ้ำกัน จึงทำการตรวจจับมัลแวร์จากรูปแบบรหัสเฉพาะนี้ และจะทำการอัปเดตข้อมูลมัลแวร์ (Signatures) เมื่อมีการเชื่อมต่อกับอินเทอร์เน็ต และเมื่อโปรแกรมมีการตรวจจับมัลแวร์ที่มีรูปแบบเฉพาะที่ตรงกับข้อมูลมัลแวร์จึงถูกจับได้ ข้อเสียของวิธีนี้คือต้องมีการติดมัลแวร์ชนิดนี้จากเครื่องใดเครื่องหนึ่งในโลก จึงจะมีการสร้างข้อมูลมัลแวร์ของมัลแวร์ชนิดใหม่ออกมา ซึ่งมัลแวร์ในปัจจุบันนี้และในอนาคตเริ่มมีการเปลี่ยนแปลงรูปแบบของมัลแวร์ (Polymorphism) คือจะมีการสร้างตัวเองขึ้นมาและเปลี่ยนรูปแบบข้อมูลมัลแวร์ไปเรื่อยๆ จากผลการรายงานของ McAfee ในปี 2007 มีมัลแวร์ที่เป็น Polymorphic 1,700,000 ชนิด จากมัลแวร์ทั้งหมด 2,800,000 ชนิด คิดเป็น 60.71% ซึ่งถือว่าเป็นจำนวนเยอะมาก จึงทำให้การตรวจจับมัลแวร์ที่คุณลักษณะเฉพาะนี้กระทำได้ยากขึ้น และจากผลการรายงานของ Symantec ในเดือนเมษายน ปี2009 พบว่าจำนวนของมัลแวร์เพิ่มขึ้นอย่างรวดเร็ว ในปี 2007 พบมัลแวร์ 624,000 ชนิด และในปี 2008 เพิ่มขึ้นไปถึง 1,656,000 ชนิด ทำให้ฐานข้อมูลใหญ่ขึ้นเรื่อยๆ และมัลแวร์บางชนิดก็ใช้ระยะเวลาสั้นๆในการแพร่กระจาย และทำลายระบบต่างๆไปมากทำให้วิธีนี้เริ่มใช้งานไม่ได้ผล ดังนั้นโปรแกรมตรวจจับมัลแวร์เชิงพฤติกรรมนี้ จึงเป็นทางเลือกใหม่ของการตรวจจับมัลแวร์

1.2 วัตถุประสงค์ของโครงงาน

1) เพื่อศึกษาการสร้างโปรแกรมมัลแวร์แบบต่างๆ
2) เพื่อศึกษาพฤติกรรมของมัลแวร์แบบต่างๆ
3) เพื่อสร้างต้นแบบระบบตรวจจับมัลแวร์เชิงพฤติกรรมสำหรับระบบปฏิบัติการวินโดวส์

1.3 ประโยชน์ที่คาดว่าจะได้รับ

1) สามารถสร้างโปรแกรมมัลแวร์แบบต่างๆได้
2) ได้ต้นแบบของพฤติกรรมมัลแวร์แบบต่างๆ
3) ต้นแบบระบบตรวจจับมัลแวร์เชิงพฤติกรรมสำหรับระบบปฏิบัติการวินโดวส์

1.4 ขอบเขตของโครงงาน

1) ศึกษากลไลการทำงานของมัลแวร์แบบต่างๆในระบบปฏิบัติการวินโดวส์
2) สร้างมัลแวร์แบบต่างๆขึ้นเองได้ เช่น Rootkit, Backdoor, Keylogger, Virus, Worm เป็นต้น
3) พัฒนาต้นแบบระบบตรวจจับมัลแวร์เชิงพฤติกรรมสำหรับระบบปฏิบัติการวินโดวส์ โดยเน้นการใช้งานง่าย จัดการแบบรวมศูนย์ ทำงานผสานกับระบบปฏิบัติการ และรักษาความปลอดภัยได้ดี

1.5 ขั้นตอนการดำเนินงาน

1) ศึกษาชนิดและพฤติกรรมของมัลแวร์แบบต่างๆ
2) ทดลองสร้างมัลแวร์แบบต่างๆขึ้น เช่น Rabbit, Macro Virus, Key Logger เป็นต้น
3) ศึกษาการเขียนโปรแกรมทางด้านเครือข่ายโดยใช้ Winsock และทดลองเขียนโปรแกรมโจมตีเครื่องเป้าหมายแบบกระจายเพื่อให้เกิดการปฏิเสธการให้บริการ (Distributed Denial of Service Attacks)
4) ศึกษาเอกสารการวิจัยเรื่องการสังเกตการณ์และวิเคราะห์พฤติกรรมของมัลแวร์ (Malware Behavior Monitor and Analysis)
5) ทดลองเขียน โปรแกรมที่ใช้ในการจัดการโพรเซสและไดเรคทอรี
6) ศึกษาการเขียนโปรแกรม Netstatp เพื่อใช้ในการสังเกตการณ์พอร์ตของเครือข่าย
7) ศึกษาและพัฒนาโปรแกรม Capture-BAT ซึ่งใช้ในการสังเกตการณ์โพรเซส รีจิสทรีและไฟล์
8) ศึกษาการเขียนโปรแกรมระบบของวินโดวส์
9) ศึกษาการรับส่งข้อมูลผ่านอินเทอร์เฟซการ์ดแลนโดยใช้ WinPcap
10) ศึกษาการตรวจจับการเปลี่ยนแปลงสถานะของระบบปฏิบัติการวินโดวส์รูปแบบต่างๆจากเอกสาร Hooking revealed และ Hooking the kernel directly
11) พัฒนาต้นแบบระบบตรวจจับมัลแวร์เชิงพฤติกรรม พร้อมทดลองการตรวจจับมัลแวร์

1.6 ส่วนประกอบของรายงาน

รายงานนี้แบ่งออกเป็น 5 บท ตามรายละเอียดดังต่อไปนี้

1) บทที่ 1 กล่าวถึงความเป็นมาของปัญหา วัตถุประสงค์ของโครงงาน ประโยชน์ที่คาดว่าจะได้รับ ขอบเขตของโครงงาน ขั้นตอนการดำเนินงาน และส่วนประกอบของรายงาน
2) บทที่ 2 กล่าวถึงทฤษฎีที่เกี่ยวข้องที่นำมาใช้ในการทำโครงงาน

3) บทที่ 3 กล่าวถึงการออกแบบและพัฒนา โดยแบ่งส่วนการพัฒนาเป็น 2 ส่วนได้แก่ ส่วนมอนิเตอร์พฤติกรรมของมัลแวร์ และส่วนวิเคราะห์เพื่อตรวจจับมัลแวร์
4) บทที่ 4 กล่าวถึงการทดลอง
5) บทที่ 5 กล่าวถึงบทสรุปของโครงงาน ปัญหาอุปสรรค แนวทางแก้ไข และแนวทางการพัฒนาต่อ

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 Windows XP Service Pack 2

ไมโครซอฟท์วินโดวส์เอกซ์พี เป็นระบบปฏิบัติการที่ไมโครซอฟท์ได้ผลิตออกมาในปี พ.ศ. 2544 ต่อจาก Windows 95/ 98/ Me/ 2000/ NT โดย XP นั้นคือตัวอักษรที่ย่อมาจาก Experience (เอกซ์พีเรียนซ์) ซึ่งมีความหมายว่า ประสบการณ์ ความรู้ที่มีโดยประสบการณ์ Windows XP มีการทำงานแบบที่เรียกว่า GUI หรือ Graphic User Interface ซึ่งเข้ากันได้ทุกระบบ มีมัลติมีเดียที่สมบูรณ์แบบ ระบบป้องกันภัยเป็นเยี่ยม ใช้งานง่าย วินโดวส์เอกซ์พีจัดเป็นระบบปฏิบัติการแรกที่ได้พัฒนาโดยใช้ฐานจากวินโดวส์เอ็นทีและมีกลุ่มเป้าหมายเป็นผู้ใช้ทั่วไป

ไมโครซอฟท์วินโดวส์เอกซ์พีรุ่นปรับปรุง (Service Pack)

1) SP1 ออกมาในปี ค.ศ. 2003 (พ.ศ. 2546)

2) SP1a ออกมาในปี ค.ศ. 2003 (พ.ศ. 2546) หลัง SP1

3) SP2 ออกมาในปี ค.ศ. 2004 (พ.ศ. 2547)

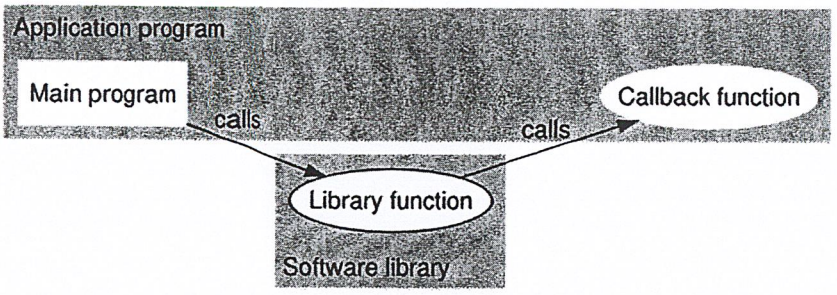
4) SP3 ออกมาในปี ค.ศ. 2008 (พ.ศ. 2551) หลังวินโดวส์วิสต้า

การปรับปรุงซอฟต์แวร์ถือว่าเป็นงานสำคัญอย่างหนึ่งที่ไมโครซอฟท์ทำอย่างต่อเนื่อง ซึ่งส่วนหนึ่งของการปรับปรุงได้แก่ การพัฒนาชุดอัปเดตและแก้ไขข้อบกพร่องต่างๆ พร้อมกับเผยแพร่สู่ผู้ใช้ และโดยปกติแล้ว ไมโครซอฟท์จะรวมรวมการแก้ไขเหล่านี้เข้าไว้ในแพ็คเกจเดียว เพื่อให้ผู้ใช้นำไปติดตั้งในคอมพิวเตอร์ได้อย่างสะดวก ซึ่งจะเรียกแพ็คเกจนี้ว่า Service Pack

Windows XP Service Pack 2 (SP2) จะอัปเดตระบบรักษาความปลอดภัยและติดตั้งนวัตกรรมใหม่ล่าสุด จากไมโครซอฟท์ให้แก่ผู้ใช้ โดยมีการตั้งค่าระบบรักษาความปลอดภัยที่แน่นหนาขึ้น ซึ่งจะช่วยปกป้องคอมพิวเตอร์ของคุณจากไวรัส, แฮ็กเกอร์ และเวิร์ม พร้อมกับมีคุณสมบัติด้าน การรักษาความปลอดภัยใหม่ที่โดดเด่น และได้รับการออกแบบมาให้สามารถปกป้องเครื่อง PC ได้ง่าย วินโดวส์เอ็กพีเซอร์วิสแพ็คสองหยุดพัฒนาไปเมื่อวันที่ 31 กรกฎาคม 2010 สามารถทำการอัปเดตไปยังเซอร์วิสแพ็คสามได้

2.2 Callback

ในการเขียนโปรแกรมคอมพิวเตอร์ Callback จะอ้างถึง Executable Code โดยส่งผ่านอาร์กิวเมนต์ไปยังโค้ดอื่นๆ ที่อนุญาตให้ซอฟต์แวร์ระดับต่ำกว่าเรียกฟังก์ชันหรือ Subroutine ในระดับที่สูงกว่า



รูป 2.1 การทำงานของ Callback

Callback เป็นโค้ดในการจัดการแอปพลิเคชั่นที่ช่วย ทำให้ฟังก์ชันDLLที่ยังไม่มีการจัดการได้ทำงานได้สมบูรณ์ยิ่งขึ้น การเรียก Callback จะส่งค่าทางอ้อมจากการจัดการแอปพลิเคชั่นผ่านฟังก์ชัน DLL และกลับมายังการดำเนินงานบางส่วนของฟังก์ชัน DLL ส่วนมากมักจะเรียกแพลตฟอร์มที่ต้องเรียกฟังก์ชัน Callback เพื่อจัดการโค้ดในการทำงานอย่างถูกต้อง

เมื่อต้องการเรียกฟังก์ชันส่วนใหญ่จากการจัดการโค้ด ต้องสร้างการจัดการคำนิยามของฟังก์ชันและเรียกใช้ ใช้ฟังก์ชัน DLL ที่ต้องการฟังก์ชัน Callback ที่มีขั้นตอนเพิ่มเติม ขั้นแรกตรวจสอบว่าต้องใช้ฟังก์ชัน Callback ขั้นตอนต่อมาคือการสร้างฟังก์ชัน Callbackในแอปพลิเคชั่นและสุดท้ายเรียกฟังก์ชัน DLL โดยส่งพ้อยเตอร์เป็นค่าพารามิเตอร์ไปยังฟังก์ชันCallback

Callback มีการเรียกใช้ได้มากมาย ตัวอย่างเช่น ฟังก์ชันทำการอ่านค่า คอนฟิกูเรชั่นไฟล์ และเชื่อมโยงค่ากับตัวเลือก ถ้าตัวเลือกถูกกำหนดโดยการแฮช (Hash) จากนั้นเขียนฟังก์ชันเพื่อCallback จะทำให้มีความยืดหยุ่นมากขึ้น ผู้ใช้สามารถเลือกอัลกอริทึมในการแฮชแบบใดก็ได้ และฟังก์ชันจะยังคงทำงานต่อไป ตั้งแต่ใช้ Callback เพื่อนำชื่อตัวเลือกเข้าไปใส่แฮชในฟังก์ชันCallback จะให้ผู้ใช้ฟังก์ชันปรับแต่งได้ขณะทำการรัน

2.3 Windows Driver Kit (WDK)

Windows Driver Kit (WDK) เป็นการพัฒนาโปรแกรมควบคุมระบบแบบครบวงจร ที่ประกอบไปด้วย Windows Driver Device Kit (DDK) และเป็นการทดสอบเพื่อความเสถียรและความน่าเชื่อถือของวินโดวส์ไดรเวอร์ เพื่อตรวจสอบระบบปฏิบัติการที่เราสามารถสร้าง ดีบัก และทดสอบไดรเวอร์ โดย WDK สร้างสภาพแวดล้อมที่แตกต่างกันสำหรับระบบปฏิบัติการที่แตกต่างกัน WDK Build 6000 Version 6.0 ที่ใช้อยู่นี้สนับสนุนวินโดวส์วิสต้า และวินโดวส์รุ่นที่ต่ำลงมา ดังนั้นเราจะอ้างอิงวินโดวส์วิสต้าในการทำงานเป็นหลัก

เมื่อทำการติดตั้ง WDK เราสามารถเลือกสร้างสภาพแวดล้อมในการติดตั้งตามอุปกรณ์และระบบปฏิบัติการที่เราจะพัฒนาไดรเวอร์ หากมีการแก้ไขหลังการติดตั้งสามารถทำได้ที่โดยเพิ่มหรือเอาโปรแกรมออกใน Control Panel

ตามนโยบายสนับสนุนของบริษัทไมโครซอฟท์ ไมโครซอฟท์จะถอด Microsoft Windows 2000 จากการสนับสนุนการติดตั้งแพลตฟอร์มสำหรับ WDK การสร้างเครื่องมือและแพลตฟอร์ม หรือสนับสนุนไลบรารี จะจำกัดที่วินโดวส์เอกพี เซอร์วิสแพ็ค2 และวินโดวส์รุ่นที่ใหม่กว่า

วินโดวส์วิสต้ามีคุณสมบัติใหม่และนโยบาย Code-Signing ซึ่งมีผลต่อการติดตั้งและการทำงานของโหมดผู้ใช้ (User Mode) และโหมดเคอร์เนล (Kernel Mode) นโยบายเหล่านี้อาจมีผลต่อองค์ประกอบของซอฟต์แวร์ที่กำหนดเองโดยทีมผู้พัฒนา ทีมพัฒนาต้องปฏิบัติตามข้อกำหนดคุณลักษณะในฟีเจอร์ต่างๆของวินโดวส์ ที่จะใช้สำหรับการพัฒนา สำหรับภาพรวมของคุณสมบัติใหม่เหล่านี้และนโยบาย Code-Signing เป็นไปตามการเข้าสู่ระบบซอฟต์แวร์ต่างๆเพื่อใช้ในสภาพแวดล้อมของการพัฒนา

2.3.1 โครงสร้างไดเร็กทอรีของ WDK

WDK มีองค์ประกอบเฉพาะหลายส่วน ถ้าเราเข้าใจโครงสร้างของ WDK เราจะเข้าใจที่ที่จะสามารถหาไฟล์สนับสนุนหรือข้อมูลได้ โดยปกติส่วนประกอบของ WDK จะติดตั้งใน WDK InstallationPath\BuildNumber\folder (เช่น C:\WinDDK\5600) ส่วนประกอบที่ถูกคัดลอกขึ้นอยู่กับตัวเลือกที่เลือกในตัวช่วยสร้างการติดตั้งไดรเวอร์โหมดยูสเซอร์และไดรเวอร์โหมดเคอร์เนล วินโดวส์ไดรเวอร์สามารถทำงานได้ในทั้งสองโหมด

1) User-Mode ไดรเวอร์ทำงานในโหมด Non Privileged Processor ซึ่งเป็นที่ที่โค้ดของแอปพลิเคชั่นอื่นๆ รวมถึง Protected Subsystem Code, Executes User-Mode ไดรเวอร์ไม่สามารถเข้าถึงข้อมูลระบบยกเว้น โดยการเรียก Win32 API ซึ่งเรียก System Services
2) Kernel-Mode ไดรเวอร์ทำงานเป็นส่วนหนึ่งของระบบผู้บริหารระบบปฏิบัติการภายใต้ส่วนประกอบของระบบปฏิบัติการที่สนับสนุน Protected Subsystems ในโหมดผู้ใช้และโหมดเคอร์เนลไดรเวอร์มีโครงสร้างที่แตกต่างกัน และการเชื่อมต่อระบบต่างๆที่แตกต่างกัน อุปกรณ์ต้องใช้โหมดผู้ใช้หรือโหมดเคอร์เนลไดรเวอร์ขึ้นอยู่กับชนิดของอุปกรณ์ที่รองรับในระบบปฏิบัติการ

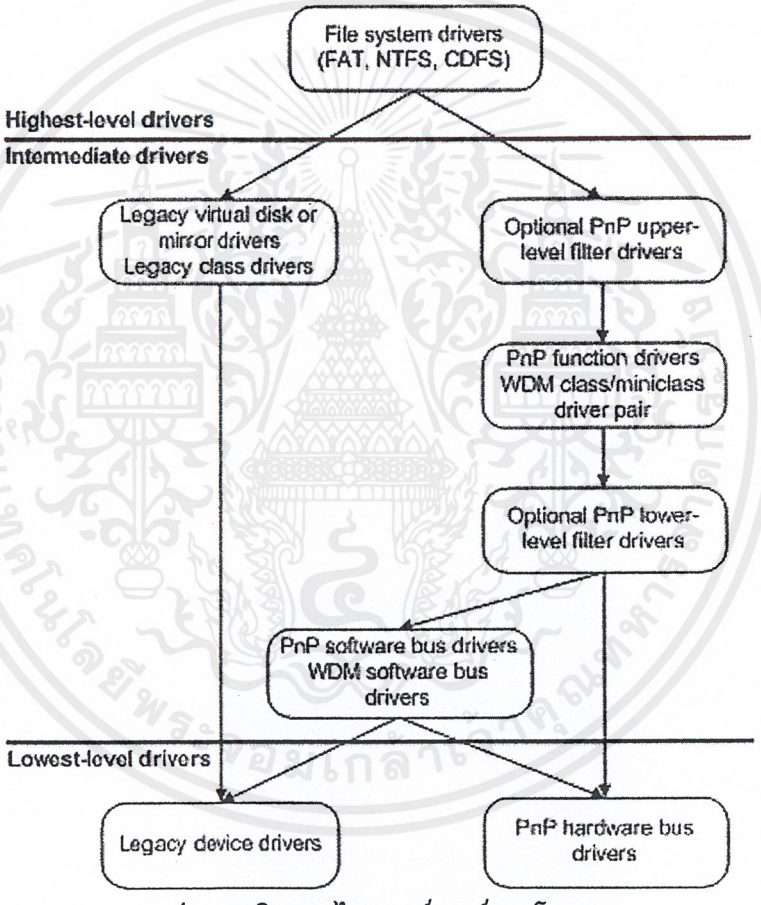
บางโปรแกรมควบคุมอุปกรณ์สามารถทำงานทั้งหมดหรือบางส่วนในโหมดผู้ใช้ ไดรเวอร์ของโหมดผู้ใช้มีพื้นที่ว่างในหน่วยความจำสแตก (Stack) ไม่จำกัด จึงสามารถเข้าถึง Win32 API และง่ายต่อการดีบัก (พร้อม Debuggers ในโหมดผู้ใช้)

ตัวอย่างเช่นโปรแกรมควบคุมเครื่องพิมพ์แบ่งออกเป็นส่วนติดต่อผู้ใช้และแสดงส่วนประกอบ ส่วนติดต่อผู้ใช้ทำงานในโหมดผู้ใช้และเรียก Win32 API เพื่อแสดงรูปภาพ Win32 API ก็เรียกองค์ประกอบการแสดงผลใน Windows Vista องค์ประกอบการแสดงผลต้องทำงานใน

โหมดผู้ใช้ แต่ในวินโดวส์เอ็กพี และวินโดวส์ 2000 องค์ประกอบการแสดงผลสามารถทำงานได้ในโหมดเคอร์เนลและโหมดผู้ใช้ แล้วแต่การเลือกใช้

ไดรเวอร์ของอุปกรณ์ส่วนใหญ่ทำงานในโหมดเคอร์เนล ไดรเวอร์ของโหมดเคอร์เนลสามารถดำเนินการป้องกันและสามารถเข้าถึงโครงสร้างระบบที่ไดรเวอร์โหมดผู้ใช้ไม่สามารถเข้าถึง อย่างไรก็ตามดีบักยากขึ้นและมีโอกาสที่ระบบจะเสียหาย เมื่อโค้ดถูกรันใน Privileged Kernel-Mode โดยการออกแบบระบบปฏิบัติการจะดำเนินการตรวจสอบน้อยกว่าข้อมูลที่สมบูรณ์และความถูกต้อง

2.3.2 ประเภทของวินโดวส์ไดรเวอร์



รูป 2.2 ชนิดของไดรเวอร์เคอร์เนลโหมด

จากรูป 2.2 ไดรเวอร์เคอร์เนลโหมดมี 3 ประเภทพื้นฐานในไดรเวอร์สแตก ได้แก่ ระดับสูงสุด ระดับกลางและระดับต่ำสุด แต่ละประเภทแตกต่างกันเพียงเล็กน้อยในโครงสร้าง แต่แตกต่างกันอย่างมากในการทำงาน

1) ไดรเวอร์ระดับสูงสุด รวมถึง File System Drivers (FSDs) ที่รองรับการทำงานของ File Systems เช่น NTFS, File Allocation Table (FAT), CD-ROM File System

(CDFS) ไดรเวอร์ระดับสูงสุดจะขึ้นอยู่กับการรองรับจากไดรเวอร์ระดับต่ำกว่า เช่น ฟังก์ชันไดรเวอร์ในระดับกลาง และฮาร์ดแวร์บัสไดรเวอร์ในระดับต่ำสุด

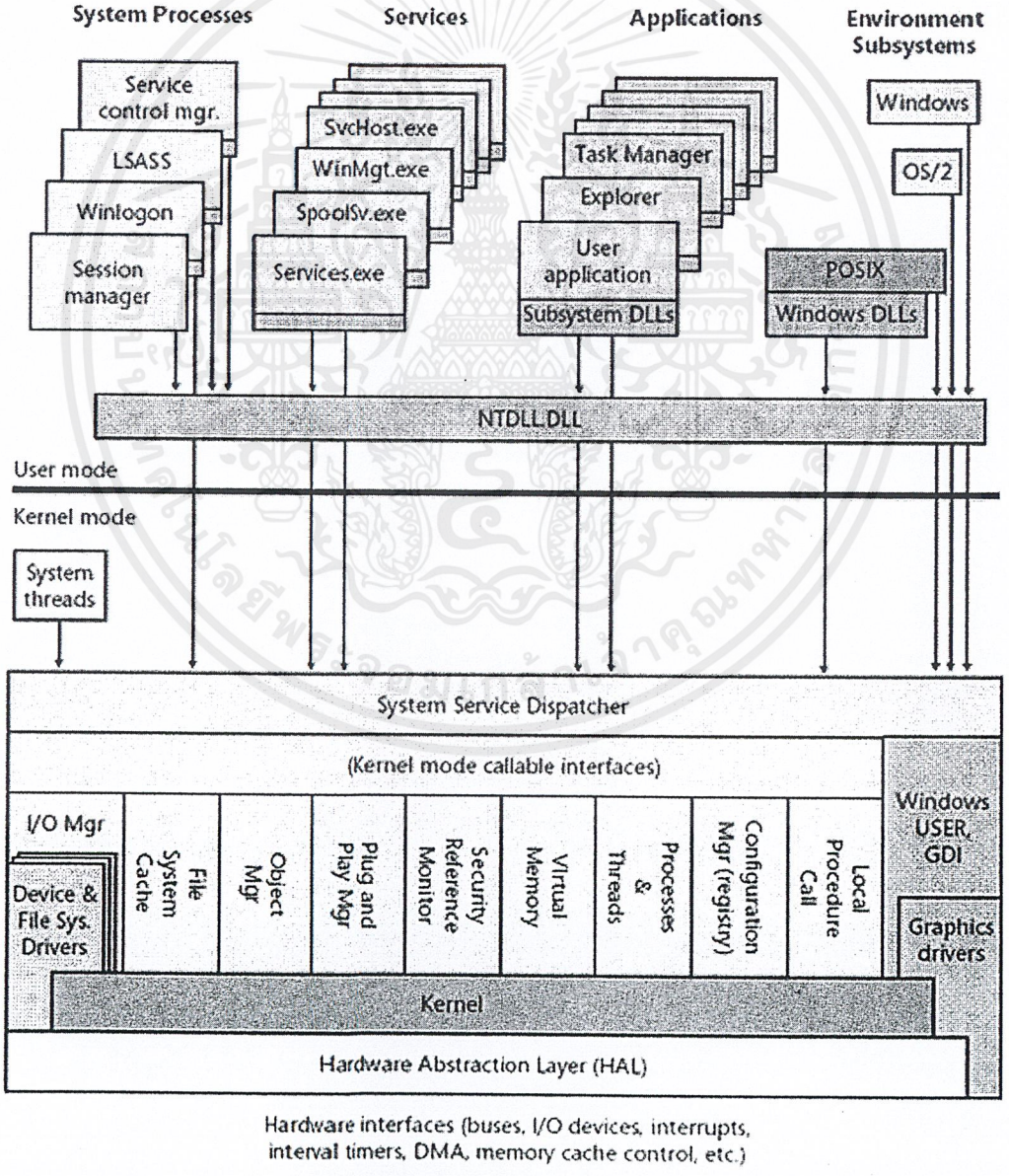
2) ไดรเวอร์ระดับกลาง เช่น Virtual Disk, Mirror โดยไดรเวอร์ระดับกลางนั้นจะขึ้นอยู่กับการรองรับจากไดรเวอร์ระดับต่ำกว่า ไดรเวอร์ระดับกลางแบ่งออกได้ ดังนี้

2.1) Function Drivers ควบคุมเฉพาะ Peripheral Devices ใน I/O Bus

2.2) Filter Drivers จะอยู่ในระดับสูงกว่าหรือต่ำกว่าฟังก์ชันของไดรเวอร์

3) ไดรเวอร์ระดับต่ำสุด ควบคุม I/O Bus ที่มี Peripheral Devices เชื่อมต่ออยู่ไดรเวอร์ระดับต่ำสุด

2.3.3 ส่วนประกอบของวินโดวส์

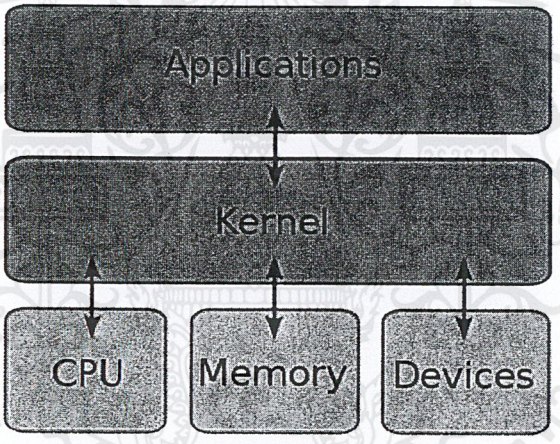


รูป 2.3 องค์ประกอบของวินโดวส์

รูป 2.3 แสดงองค์ประกอบภายในที่สำคัญของระบบปฏิบัติการวินโดวส์ทั้งส่วนประกอบในโหมดผู้ใช้ และโหมดเคอร์เนล ไดรเวอร์จะทำการเรียก Routine ที่ถูกส่งมาจากส่วนประกอบต่างๆของเคอร์แนล เช่น เมื่อจะทำการสร้าง Device Object จะต้องเรียก IoCreateDevice ที่ถูกส่งมาจากตัวจัดการ I/O นอกจากนี้ไดรเวอร์ต้องตอบสนองการเรียกของระบบปฏิบัติการและซิสเต็มคอลอื่นๆ

2.4 เคอร์เนล

เคอร์เนล หมายถึง ส่วนประกอบหลักของระบบปฏิบัติการ ซึ่งคอยดูแลบริหารทรัพยากรของระบบ และติดต่อกับฮาร์ดแวร์และซอฟต์แวร์ เนื่องจากว่าเป็นส่วนประกอบพื้นฐานของระบบปฏิบัติการ เคอร์เนลนั้นเป็นฐานล่างสุดในการติดต่อกับทรัพยากรต่างๆ เช่น หน่วยความจำ หน่วยประมวลผลกลาง และอุปกรณ์อินพุทเอาท์พุท



รูป 2.4 การเชื่อมต่อของเคอร์เนล

2.4.1 การป้องกันเคอร์เนล

การพิจารณาที่สำคัญในการออกแบบเคอร์เนลให้รองรับการป้องกันความผิดพลาด และจากพฤติกรรมที่เป็นอันตรายทั้งสองด้านมักไม่เด่นชัดและการยอมรับความแตกต่างนี้ในการออกแบบเคอร์เนลนำไปสู่การปฏิเสธของโครงสร้างลำดับชั้นเพื่อป้องกัน

กลไกหรือนโยบายที่ทำให้เคอร์เนลสามารถจำแนกตามหลักเกณฑ์หลายๆแบบ ได้แก่ แบบคงที่ (ใช้เมื่อ Compile Time) หรือแบบพลวัต (ใช้เมื่อ Runtime) การตรวจสอบก่อนหรือหลังตามหลักการป้องกันการรองรับฮาร์ดแวร์หรือภาษา ไม่ว่าจะเป็นอีกกลไกที่เปิดหรือซ่อนนโยบายที่มีผลผูกพันและอีกมากมาย

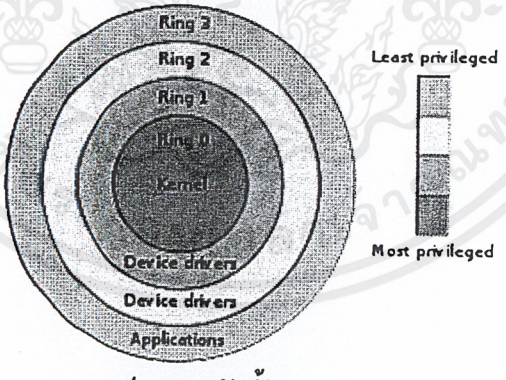
ประโยชน์ของการวัดระดับความผิดพลาดของระบบอย่างใกล้ชิดเป็นวิธีปฏิบัติตามหลักการ Least Privilege ในกรณีที่โปรแกรมหลายๆโปรแกรมกำลังทำงานอยู่ในคอมพิวเตอร์

เครื่องเดียว สิ่งสำคัญในการป้องกันความผิดพลาดในหนึ่งโปรแกรมอาจมีผลต่อโปรแกรมอื่นๆ ความผิดพลาดนี้ยังใช้กับการรักษาความปลอดภัยที่จำเป็นเพื่อป้องกันการเข้าถึงข้อมูลโดยไม่ได้รับอนุญาต

ทั้งสองวิธีทางฮาร์ดแวร์สำหรับการป้องกัน (จากข้อมูลที่สำคัญ) มีการป้องกันเป็นลำดับชั้น (เรียกว่า สถาปัตยกรรม Ring หรือโหมด Supervisor) Privilege Ring เช่นใน x86 มีการปฏิบัติทั่วไปของในการป้องกันเป็นลำดับชั้น ถูกใช้ในการติดต่อจำนวนมากที่มีความผิดพลาดในการทนทานบางระดับ

โดเมนลำดับชั้นการป้องกันมีความยืดหยุ่นน้อยมากเช่นเดียวกับกรณีที่ทุกๆเคอร์เนลกับโครงสร้างลำดับชั้น สมมุติ เช่น การออกแบบระดับ Global ในกรณีที่การคุ้มครองไม่สามารถเป็นไปได้ที่จะกำหนดสิทธิ์ที่แตกต่างกันเพื่อการที่มีสิทธิพิเศษที่ระดับเดียวกัน ดังนั้นจึงไม่สามารถตอบสนองต่อกฎความทนทานต่อความผิดพลาด 4 ข้อของ Denning โดเมนลำดับชั้นการป้องกันมีประสิทธิภาพในการย้อนกลับ เมื่อมีการติดต่อกันระหว่างระดับต่างๆในการป้องกัน เมื่อกระบวนการมีการจัดการโครงสร้างข้อมูลทั้งในโหมดผู้ใช้และโหมด Supervisor ต้องคัดลอกข้อความ (ส่งค่า) เคอร์เนลขึ้นอยู่กับตามความสามารถ แต่จะมีความยืดหยุ่นในการกำหนดสิทธิ์สามารถตอบสนองหลักการของ Denning ในการยอมรับความผิดพลาด และมักจะไม่ประสบปัญหาประสิทธิภาพการทำงานในการคัดลอกค่า

2.4.2 วงแหวน (Ring)



รูป 2.5 ลำดับชั้นวงแหวน

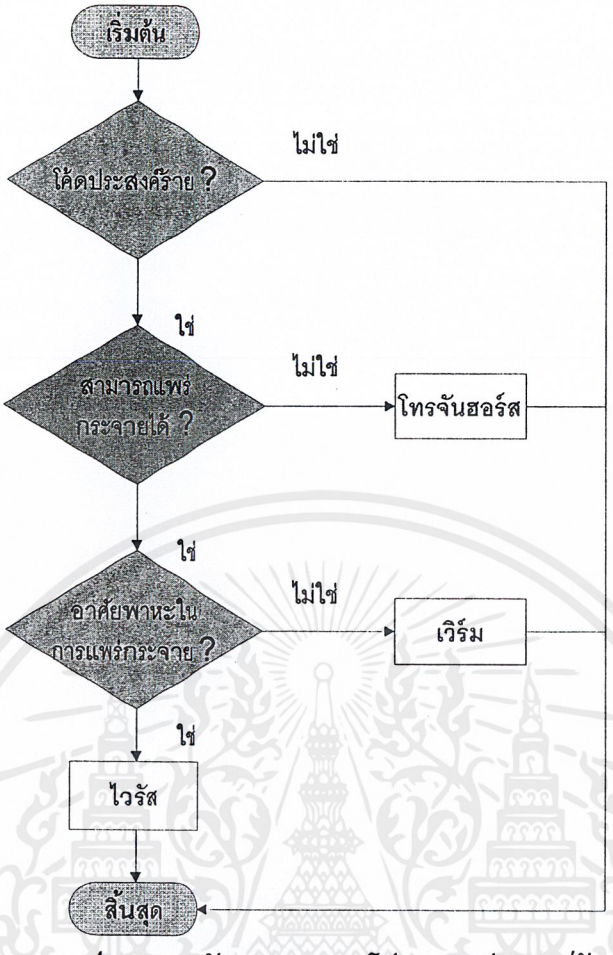
ในวิทยาการคอมพิวเตอร์จะมีโดเมนการป้องกันที่เป็นลำดับชั้น เรียกว่า วงแหวนการป้องกัน (Protection Ring) เป็นกลไกการป้องกันข้อมูลและความผิดพลาดจากการพฤติกรรมที่เป็นอันตราย ระบบปฏิบัติการคอมพิวเตอร์จะมีการแบ่งระดับในการเข้าถึงทรัพยากร ในวงแหวนป้องกัน

Ring จะมีโครงสร้างการจัดเรียงจากระดับสิทธิ์ที่มากสุด (Ring น้อยที่สุด คือ Ring 0) และระดับสิทธิ์น้อยที่สุด (Ring มากที่สุด คือ Ring 3) บนระบบปฏิบัติการส่วนใหญ่ Ring 0 เป็นหน่วยความจำที่มีสิทธิ์เข้าถึงและโต้ตอบมากที่สุดโดยตรงทางกายภาพกับฮาร์ดแวร์ เช่น CPU และหน่วยความจำ โดยในระบบปฏิบัติการวินโดวส์นั้นจะกล่าวถึงแค่ Ring 0 และ Ring 3 เท่านั้น

ทางพิเศษของ Ring จะมีไว้เพื่อให้วงแหวนชั้นนอกสามารถเข้าถึงทรัพยากรของวงแหวนชั้นในในลักษณะที่มีการกำหนดไว้ล่วงหน้า ทางที่ถูกต้องในการเข้าถึงระหว่าง Ring สามารถสร้างความปลอดภัยโดยการป้องกันโปรแกรมหนึ่งจากการเข้าถึงของอีกโปรแกรมหนึ่ง เช่น สปายแวร์ถูกประมวลผลในโปรแกรมของผู้ใช้งานใน Ring 3 ควรป้องกันจากการเปิดเว็บแคมโดยไม่ต้องแจ้งให้ผู้ใช้ เนื่องจากการเข้าถึงฮาร์ดแวร์ควรเป็นฟังก์ชันการใช้งานของ Ring 1 ที่เป็นที่สำหรับไดรเวอร์ของอุปกรณ์ โปรแกรมเช่นเว็บบราวเซอร์ จะทำงานใน Ring ที่มีเลขสูงกว่าและต้องขอการเข้าถึงเครือข่าย และทรัพยากรที่จำกัดใน Ring ที่ระดับต่ำกว่า

2.5 มัลแวร์

มัลแวร์ (Malware) ย่อมาจากคำว่า Malicious Software คือประเภทของโปรแกรมคอมพิวเตอร์ที่ถูกสร้างขึ้นมา โดยมีจุดมุ่งหมายเพื่อที่จะทำลายหรือสร้างความเสียหายให้แก่ระบบคอมพิวเตอร์ ระบบเครือข่าย หรือทรัพย์สินและข้อมูลของผู้ใช้งานคอมพิวเตอร์ ประเภทของมัลแวร์ต่างๆ มีดังต่อไปนี้



รูป 2.6 แผนผังการแยกแยะโปรแกรมประสงค์ร้าย

2.5.1 ไวรัส

พฤติกรรมของไวรัส คือ การขยายพันธุ์หรือก็อปปี้ตัวเอง ไปติดกับไฟล์ Executable (.exe files) หรือไฟล์ของอีกโปรแกรมหนึ่ง หลังจากนั้น หากผู้ใช้งานไปดับเบิ้ลคลิกหรือรันโปรแกรมไฟล์นั้นๆ ไวรัสก็จะถูกกระตุ้นให้ทำงานทันที โดยส่วนที่เป็นพิษของไวรัส เรียกว่า Payload ซึ่งเป็นส่วนที่มีคำสั่งที่ระบุให้ไวรัสทำงานตาม ยกตัวอย่างเช่น Payload ของไวรัส อาจจะระบุไว้ว่าให้ไวรัสทำการลบไฟล์เอกสารต่างๆ ในเครื่องของผู้ใช้งาน หรือ Payload อาจจะสั่งให้ไวรัสลงโปรแกรมประเภทสปายแวร์ไว้ในเครื่องเพื่อคอยดักเก็บข้อมูลส่วนตัวของผู้ใช้งาน เป็นต้น ลักษณะเด่นของไวรัสคือ มันไม่สามารถที่จะทำงานได้ด้วยตัวของมันเอง ไวรัสจะเริ่มการทำงานหรือถูกปลุกให้ตื่นขึ้นมาได้นั้น จะต้องมีการถูกกระตุ้นก่อน เช่น ถ้าดาวน์โหลดไฟล์แนบที่มากับอีเมล์ซึ่งมีไวรัสอยู่ การดาวน์โหลดไฟล์ไวรัสมาไว้ที่เครื่องเฉยๆ นั้น ไวรัสจะยังไม่สามารถทำงานได้ (บางครั้งโปรแกรมแอนติไวรัส อาจจะฟ้องขึ้นมาทันที ว่ากำลังดาวโหลดไฟล์ที่มีไวรัสอยู่ และทำการลบไฟล์นั้นทิ้ง) แต่ถ้าเครื่องคอมพิวเตอร์ไม่มีโปรแกรมแอนติไวรัส หรือโปรแกรมแอนติไวรัส

ของเราไม่รู้จักไวรัสตัวนี้ ไฟล์แนบที่มีไวรัสก็จะถูกโหลดมาไว้ในเครื่องคอมพิวเตอร์ได้สำเร็จ และเมื่อไหร่ก็ตามที่ไปดับเบิ้ลคลิกหรือเปิดไฟล์แนบนี้ขึ้นมาใช้งาน ไวรัสก็จะเริ่มทำงานทันที

2.5.2 เวิร์ม

ลักษณะการทำงานของเวิร์มจะต่างจากไวรัสตรงที่เวิร์มสามารถทำงานได้ด้วยตัวของมันเอง (ทำงานแบบอัตโนมัติ) โดยไม่ต้องอาศัยพาหะ หรือการเกาะไฟล์ Executable อื่นๆ โดยเวิร์มจะใช้การกระจายพันธุ์ผ่านทางระบบเครือข่ายเป็นหลัก ด้วยการคอยสแกนระบบเครือข่ายอยู่ตลอดเวลา เพื่อที่จะหาช่องโหว่ของเครื่องที่มีอยู่ในระบบ ถ้าเครื่องไหนมีช่องโหว่ (ซึ่งเกิดจากการไม่ได้ลง Patch/Service Pack หรือใช้โปรแกรมเวอร์ชั่นเก่า) เวิร์มก็จะโจมตี (Exploit) เครื่องเหล่านั้นทันที และก็ควบคุมเครื่องเหล่านั้น ให้ช่วยกระจายพันธุ์ต่อไป ดังนั้น เรามักจะสังเกตเห็นได้ว่า เมื่อใดก็ตามที่มีการระบาดของเวิร์ม ระบบเครือข่ายของเราก็จะทำงานช้ามาก เพราะว่าแบนด์วิดธ์ของเครือข่ายจะถูกเวิร์มใช้ในการสแกนเครื่องต่างๆ โดยปกติแล้วเวิร์มมักจะแฝงตัวมากับไวรัส ในลักษณะของไฟล์ที่แนบมากับอีเมล์ เพราะว่าการโจมตีผู้ใช้งานแบบนี้ได้ผลดีกว่าและง่ายกว่า และการที่เวิร์มฝังตัวมากับไฟล์ที่แนบมากับอีเมล์นั้น ไฟร์วอลล์ของระบบเครือข่ายก็จะไม่สามารถตรวจจับและหยุดเวิร์มตัวนั้นได้ นอกเสียจากเราจะมีโปรแกรมแอนติไวรัสทำงานอยู่ในเมล์เซิร์ฟเวอร์

2.5.3 ม้าโทรจัน / แบ็คดอร์

ม้าโทรจันถูกสร้างขึ้นมาเพื่อหลอกให้ชาวเมืองทรอยเข้าใจผิด คิดว่าทหารกรีกยอมแพ้และสร้างม้าโทรจันขึ้นมาเพื่อเป็นรางวัลในการขอหย่าศึก แต่จริงๆ แล้ว มีทหารกรีกจำนวนหนึ่งแอบซ่อนตัวอยู่ภายในม้าโทรจัน เมื่อถึงเวลากลางคืน ทหารกรีกที่ซ่อนตัวเหล่านั้นก็ออกมาเปิดประตูเมืองทรอย เพื่อให้ทหารกรีกที่เหลือ ที่แอบซ่อนตัวอยู่ภายนอกกำแพงเข้ามาทำลายเมืองทรอยได้ในที่สุด ซึ่งม้าโทรจันในเชิงของความปลอดภัยด้านไอทีนั้น ก็คือโปรแกรมที่ถูกสร้างขึ้นมาเพื่อใช้ในการหลอกล่อให้ผู้ใช้งานทั่วไปดาวน์โหลดไปใช้งาน โดยที่ไม่รู้ตัวเลยว่ามีโปรแกรมอื่น (ที่ชั่วร้าย) แอบแฝงมาด้วย ยกตัวอย่างเช่น โปรแกรมประเภทที่เป็นตัวการ์ตูนที่เคลื่อนไหวอยู่บนหน้าจอ หรือ โปรแกรมพักหน้าจอ ที่สามารถดาวน์โหลดได้ฟรีจากอินเตอร์เน็ท ถ้าเราดาวโหลดโปรแกรมประเภทนี้ มาจากเว็บไซท์ที่ไม่น่าเชื่อถือ โปรแกรมที่เราดาวน์โหลดมาอาจจะเป็นโทรจันก็ได้ ซึ่งเมื่อเราลงโปรแกรมที่มีโทรจันอยู่ด้วย เราก็ยังสามารถใช้งานโปรแกรมเหล่านั้นได้ตามปกติ แต่ที่จริงแล้ว โทรจันจะทำงานอยู่เงียบๆ เบื้องหลัง โดยการฝังตัวลงในเครื่องของเรา และติดตั้งโปรแกรมประเภทแบ็คดอร์ เพื่อเปิดช่องทางให้ผู้โจมตีเข้าสู่เครื่องของเราได้อย่างลับๆ ทุกเมื่อที่ต้องการ

2.5.4 รูทคิท

ถ้าเครื่องคอมพิวเตอร์ของเราถูกเจาะหรือถูกแฮกได้สำเร็จ สิ่งแรกๆ ที่ผู้โจมตีจะทำก็คือลงโปรแกรมประเภทรูทคิท ในเครื่องของเรา เพราะรูทคิทจะช่วยซ่อนไฟล์ โปรแกรมและโพรเซสต่างๆ ที่ผู้โจมตี ติดตั้งหรือดาวน์โหลดมาวางไว้ในเครื่องของเรา เพื่อไม่ให้เราหรือโปรแกรมแอนติไวรัส ตรวจพบได้นั่นเอง ซึ่งถ้าเครื่องของเราถูกติดตั้งรูทคิทไปแล้วละก็โอกาสที่จะถอนรูทคิทออกจากเครื่องให้ได้นั่น แทบจะเป็นไปไม่ได้เลย เพราะว่ารูทคิทจะไปแก้ไขหรือเปลี่ยนไฟล์ระบบที่สำคัญต่างๆ ทำให้เมื่อเราลบไฟล์ที่เป็นของรูทคิทออก ก็จะมีผลกระทบต่อไฟล์ระบบเหล่านี้ด้วย ทำให้เกิดความเสียหายต่อระบบปฏิบัติการคอมพิวเตอร์ จนอาจจะไม่สามารถบูตเครื่องขึ้นมาทำงานได้อีกต่อไปเลย

2.5.5 บอท (หรือ ซอมบี้)

บอท (หรือที่เรียกอีกชื่อหนึ่งว่า ซอมบี้) คือเครื่องคอมพิวเตอร์ที่ถูกผู้โจมตียึดและควบคุมให้ทำตามคำสั่งต่างๆ ของผู้โจมตีผ่านทางอินเตอร์เน็ท โดยผู้โจมตีมักจะติดตั้งรูทคิทไว้ด้วย เพื่อซ่อนไฟล์และโปรแกรมของผู้โจมตีไม่ให้ถูกจับได้ โดยที่ผู้ใช้งานคอมพิวเตอร์จะไม่รู้ตัวเลยว่าเครื่องของเค้ากลายเป็นบอทไปซะแล้ว ซึ่งในขณะที่เค้ากำลังทำงานตามปกติ เช่น พิมพ์งานเอกสาร ฟังเพลง หรือเล่นเกมส์ ผู้โจมตีก็อาจจะส่งคำสั่งผ่านทางอินเตอร์เน็ทมาให้เครื่องของผู้ใช้งานคนนั้นไปโจมตีเว็บไซต์ต่างๆ หรือส่งสแปมก็ได้ โดยคำสั่งเหล่านี้จะทำงานอยู่เบื้องหลัง ปกติแล้วผู้โจมตีจะยึดเครื่องคอมพิวเตอร์หลายๆเครื่อง เพื่อทำเป็นบอทเอาไว้ใช้ในงานต่างๆ ซึ่งเราเรียกกลุ่มของเครื่องที่เป็นบอทที่สามารถทำงานร่วมกันได้ ภายใต้คำสั่งของผู้โจมตีว่า "Botnet" หรือ "Bot network" ซึ่งบอทเน็ทมักจะทำงานด้วยการรับคำสั่งจากผู้โจมตีผ่านช่องทางของ Internet Relay Chat (IRC) ซึ่งวิธีนี้ทำให้ผู้โจมตีสามารถเชื่อมต่อกับบอทเน็ทได้สะดวก และสามารถกระจายคำสั่งไปยังเครื่องบอทต่างๆ ได้พร้อมกันในเวลาเดียวกัน

2.5.5.1 พฤติกรรมของบอท

1) กิจกรรมของบอทใกล้เคียงกับการโจมตีแบบ DDoS สั่งให้บอทส่งสแปมหรือเพย์โหลดในเวลาเดียวกัน บอททางเครือข่ายเวลาส่วนใหญ่จะอยู่เฉยๆ และมีช่วงเวลาสั้นๆช่วงนึงที่จะมีจำนวนการเชื่อมต่อที่สูง
2) บอทมีจำนวนการเชื่อมต่อที่ล้มเหลวสูง
3) บอทที่ถูกควบคุมด้วย IRC มักจะแสดงจำนวนทราฟฟิกของ IRC ในปริมาณมาก (ถ้าเราไม่ได้เล่นเกมออนไลน์และแชท)
4) HTTP บอทใช้ IP ของเว็บเซิร์ฟเวอร์ มากกว่าใช้ชื่อของเซิร์ฟเวอร์
5) บอทจะมีการส่งอีเมล์จำนวนมาก (SMTP) หรือลง SMTP เซิร์ฟเวอร์ไว้ในเครื่อง

6) บอทบางตัวใช้ UDP ซึ่งโดยส่วนใหญ่แล้วการเชื่อมต่ออินเตอร์เน็ทไม่นิยมใช้กัน

7) HTTP ระหว่างบอทและ C&C เซิร์ฟเวอร์ สามารถรวม URI strings ที่น่าสงสัยหรือ HTTP เฮดเดอร์ที่ไม่เป็นมาตรฐาน

8) เชื่อมต่อกับ IP ที่แตกต่างกันจำนวนมากในระยะเวลาอันสั้น

2.5.6 สปายแวร์

โปรแกรมประเภทสปายแวร์ เช่น Keylogger จะติดตั้งตัวของมันลงในเครื่องของเราอย่างลับๆ โดยไม่ขออนุญาตเราก่อน ว่าอยากจะลงหรือไม่ ซึ่งเมื่อสปายแวร์ถูกติดตั้งลงในเครื่องแล้ว มันจะคอยทำงานโดยการบันทึกข้อมูลต่างๆ ที่เกี่ยวกับการใช้งานเครื่องคอมพิวเตอร์ของเรา เช่น คอยดูว่าเราไปเข้า เว็บไซต์ไหนบ้าง เราพิมพ์ตัวอักษรอะไรบนคีย์บอร์ดบ้าง และอื่นๆ ดังนั้นสปายแวร์จึงสามารถใช้เป็นเครื่องมือสำหรับผู้โจมตีในการขโมย ชื่อผู้ใช้และรหัสผ่าน และข้อมูลเกี่ยวกับบัตรเครดิตของเราได้ด้วย โดยที่สปายแวร์จะส่งข้อมูลที่มันเก็บได้จากเครื่องของเราไปให้เจ้านายของมันผ่านทางอินเตอร์เน็ต หรือระบบเครือข่ายอย่างเงียบๆ เพื่อไม่ให้เรารู้ตัว

2.5.6.1 API ที่ Keylogger นิยมใช้

1) GetAsyncKeyState()

2) GetKeyboardState()

3) GetKeyState()

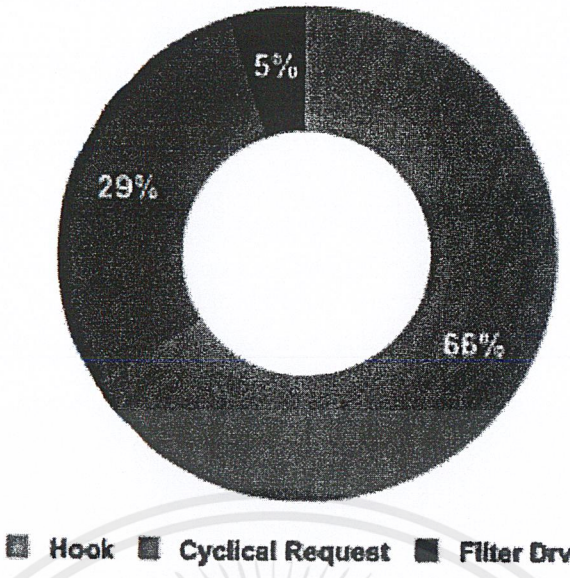
4) GetRawInputData()

2.5.6.2 วิธีที่พบมากในการสร้าง software-base keylogger มีดังนี้

1) System hook ซึ่งจะคอยดัก notification ที่บอกว่าคีย์บอร์ดถูกกด ติดตั้งโดยใช้ WinAPI SetWindowsHook สำหรับ message ต่างๆที่ถูกส่งโดย window procedure

2) Cyclical information keyboard request ที่จะคอยวนลูปเช็คค่าการกดคีย์บอร์ด โดยใช้WinAPI GetAsyncKeyState หรือ GetKeyboardState

3) Filter driver



รูป 2.7 อัตราส่วนในการสร้าง software-base keylogger

2.5.7 แอดแวร์

โปรแกรมประเภทแอดแวร์ (ส่วนใหญ่มักจะมาคู่กับสปายแวร์เพื่อใช้ในการทำการโฆษณาสินค้าแบบเฉพาะเจาะจงกลุ่มเป้าหมาย เพื่อเพิ่มโอกาสในการขายสินค้าให้ได้มากขึ้น) จะคอยแสดงโฆษณาสินค้าต่างๆ บนหน้าจอของเรา (หรือที่เรียกกันสั้นๆ ว่า "Popup Ads") ผ่านทางเว็บบราวเซอร์ เช่น อินเตอร์เน็ท เอ็กพลอเรอร์ หรือ โมซิลล่า ไฟร์ฟอกซ์ ซึ่งสร้างความรำคาญให้กับผู้ใช้งานส่วนมาก ที่ไม่สนใจในโฆษณา หรือ ป๊อบอัพโฆษณาเหล่านั้น โปรแกรมประเภทแอดแวร์มักจะถูกติดตั้งพร้อมกับโปรแกรมที่เราดาวน์โหลดมาฟรีๆ จากเว็บไซต์ที่ไม่น่าเชื่อถือ

2.5.8 ฮ็อกซ์ (Hoaxes)

โดยทั่วไปฮ็อกซ์ (Hoaxes) หมายถึง โปรแกรมที่เขียนขึ้นเพื่อหลอกให้ผู้ใช้ทำบางอย่างให้ โดยฮ็อกซ์จะใช้เทคนิคทางด้านวิศวกรรมสังคม (Social Engineering) เพื่อหลอกให้ผู้ใช้งานคอมพิวเตอร์ทำบางอย่างให้ อย่างไรก็ตามในกรณีนี้ฮ็อกซ์ไม่มีส่วนที่เป็นโค้ดประสงค์ร้ายใดๆ นอกจากแค่ต้องการหลอกเท่านั้น ฮ็อกซ์มีหลากหลายแบบ แต่ที่เห็นบ่อย เช่น การส่งอีเมลเพื่อหลอกว่ามีไวรัสตัวใหม่กำลังแพร่ระบาดและหลอกให้ส่งเมลนี้ต่อไปยังเพื่อนคนอื่นๆ ฮ็อกซ์ประเภทนี้ทำให้ผู้ใช้เสียเวลา โดยใช้รีซอร์สของเมลเซิร์ฟเวอร์และใช้แบนด์วิธของเครือข่ายโดยเปล่าประโยชน์

2.5.9 สแกมส์ (Scams)

สแกมส์ (Scams) หมายถึง การใช้ช่องการสื่อสารโดยอาชญากรเพื่อพยายามจะหลอกให้คนอื่นช่วยเหลือตัวเองในการทำอาชญากรรมบนอินเตอร์เน็ต เว็บไซต์ และอีเมลอาจถูกใช้เพื่อหลอกคนอื่น ยกตัวอย่างเช่น ผู้ร้ายอาจใช้อีเมลเพื่อหลอกให้ผู้ใช้เปิดเผยข้อมูลส่วนตัว เช่น บัญชีธนาคาร หมายเลขบัตรเครดิต แล้วอาชญากรอาจใช้ข้อมูลนี้เพื่อทำในสิ่งที่ผิดกฎหมาย สแกมส์นี้

ส่วนใหญ่จะเรียกว่า ฟิชชิง (Phishing) หรือแบรนด์สปูฟิง (Brand spoofing) หรือการ์ดดิง (Carding)

ตัวอย่างของการทำฟิชชิง เช่น ผู้ร้ายอาจปลอมตัวเป็นตัวแทนจากบริษัท อีเบย์ (eBay) เพื่อหลอกให้ผู้ใช้เผยข้อมูลส่วนตัว เช่น ยูสเซอร์เนม พาสเวิร์ด และบัญชีธนาคาร เป็นต้น ฟิชชิงส่วนใหญ่จะสร้างเว็บไซต์ที่ลอกเลียนแบบเว็บไซต์ที่เป็นทางการของบริษัทซึ่งเป็นที่รู้จักกันโดยทั่วไป และอาจใช้อีเมลเพื่อหลอกให้ผู้ใช้เข้าไปยังอีกเว็บไซต์หนึ่งแล้วหลอกให้กรอกข้อมูลส่วนตัว เช่น บัญชีธนาคารของตัวเอง ข้อมูลนี้ก็จะถูกใช้สำหรับการทำผิดกฎหมายต่อไป ในกรณีนี้ควรมีการจัดการอย่างเด็ดขาด และรายงานให้เจ้าหน้าที่ตำรวจทราบเพื่อดำเนินคดีตามกฎหมายต่อไป

2.5.10 สแปม (Spam)

สแปม (Spam) คือ การส่งอีเมลไปยังผู้ใช้จำนวนมาก โดยมีจุดประสงค์เพื่อการโฆษณาสินค้าหรือบริการ สแปมจัดอยู่ในประเภทสิ่งที่ก่อให้เกิดความรำคาญแต่ไม่ใช่มัลแวร์ เพราะไม่มีจุดประสงค์ร้ายอย่างไรก็ตามการส่งสแปมเมลจำนวนมหาศาลในปัจจุบันได้สร้างความเสียหายเนื่องจากสแปมทำให้ผู้ใช้หรือพนักงานเสียเวลาเนื่องจากต้องลบอีเมลขยะทุกวัน

ที่มาของคำว่าสแปม (Spam) นั้นยังไม่รู้แน่ชัด อย่างไรก็ตามสแปมกลายเป็นสิ่งที่สร้างความรำคาญในสังคมอินเตอร์เน็ตเรื่อยๆ และยังไม่มีทางแก้ไขอย่างถาวร หลายคนจัดว่าสแปมเมลเป็นประเด็นที่สำคัญและเป็นสิ่งที่คุกคามระบบอีเมลไปทั่วโลก เนื่องจากสแปมเมลจะเพิ่มโหลดให้กับเมลเซิร์ฟเวอร์

ถึงแม้ว่าสแปมเมลจะไม่ใช่สิ่งที่เลวร้าย แต่มัลแวร์อาจใช้สแปมเมลเพื่อแพร่กระจายไวรัส เวิร์ม และ โปรแกรมประสงค์ร้ายอื่นๆ ได้ด้วย โดยผู้ส่งสแปมเมล หรือเรียกว่า สแปมเมอร์ (Spammer) อาจติดตั้งเมลเซิร์ฟเวอร์เล็กๆ ในเครื่องของผู้ใช้เอง เพื่อสำหรับใช้ส่งสแปมเมล

2.5.11 อินเทอร์เน็ตคุกกี้ (Internet Cookies)

อินเตอร์เน็ตคุกกี้ (Internet Cookies) คือ เท็กซ์ไฟล์ที่เก็บไว้ที่เครื่องคอมพิวเตอร์ของผู้ใช้โดยเว็บไซต์ที่เข้าไปเยี่ยมชม คุกกี้จะเก็บข้อมูลบางอย่างที่เว็บไซต์นั้นใช้เมื่อครั้งหน้าที่ผู้ใช้เข้าไปเยี่ยมชมอีกครั้ง ซึ่งส่วนใหญ่จะเป็นข้อมูลที่ใช้บอกว่าเป็นผู้ใช้คนนี้ นอกจากนี้ในไฟล์อาจมีข้อมูลอื่นๆ ก็ได้

คุกกี้เป็นเครื่องมือที่ถูกกฎหมายที่หลายเว็บไซต์ใช้สำหรับติดตามข้อมูลของผู้ใช้คนนี้ ยกตัวอย่างเช่น ผู้ใช้อยากชอปปิ้งในเว็บไซต์ขายสินค้าออนไลน์ โดยในขณะที่ผู้ใช้กำลังดูสินค้าและเลือกสินค้าบางชิ้นเพื่อจะซื้อ ซึ่งสินค้าที่เลือกแล้วนั้นเว็บไซต์จะจัดเก็บไว้ในชอปปิ้งคาร์ท ซึ่งผู้ใช้อาจจะยังไม่ได้ตัดสินใจซื้อแต่ก็ไปดูเว็บไซต์อื่นแทน ข้อมูลเกี่ยวกับสินค้าที่ผู้ใช้เลือกไว้ก่อนหน้าอาจถูกเก็บไว้ในคุกกี้ และเมื่อผู้ใช้กลับมาที่เว็บไซต์นี้อีกครั้งก็สามารถช็อปปิ้งต่อได้

ดังนั้น ในเครื่องคอมพิวเตอร์ของผู้ใช้ก็จะมีคุกกี้จากหลายเว็บไซต์ แต่ละเว็บไซต์ควรที่จะเรียกดูข้อมูลที่สร้างไว้โดยเว็บไซต์ตนเองเท่านั้น ถ้าเว็บไซต์นั้นได้เรียกดูข้อมูลที่เก็บไว้ในคุกกี้ที่สร้างโดยเว็บไซต์อื่นอาจเป็นการละเมิดสิทธิส่วนบุคคลของผู้ใช้ก็ได้ และที่ผ่านมาได้มีบางเว็บไซต์ที่พยายามจะเขียนโปรแกรมเพื่อรวบรวมข้อมูลต่างๆ ที่เก็บไว้ในคุกกี้โดยที่ผู้ใช้ไม่รู้ตัว บางเว็บไซต์อาจหลอกผู้ใช้หรือไม่ปฏิบัติตามนโยบาย ยกตัวอย่างเช่น เขาอาจติดตามได้ว่าผู้ใช้เข้าไปดูเว็บไซต์ไหนบ้างโดยที่ไม่ได้แจ้งให้ผู้ใช้รู้ แล้วเขาก็จะจัดโฆษณาที่คาดว่าผู้ใช้น่าจะสนใจให้ ซึ่งการทำเช่นนี้ถือได้ว่าเป็นการละเมิดสิทธิส่วนบุคคล และยิ่งเป็นการยากที่จะแยกแยะระหว่างการละเมิดสิทธิส่วนบุคคลหรือไม่กับคุกกี้ธรรมดาทั่วไป ทำให้ยากต่อการป้องกัน นอกจากนี้นโยบายการรักษาความปลอดภัยของแต่ละเครื่องก็แตกต่างกันทำให้ยิ่งยากต่อการพัฒนาโปรแกรมป้องกันคุกกี้ (Anti-Cookie Program) ที่จะให้ได้ตามความต้องการของทุกคนได้

2.6 คุณสมบัติของมัลแวร์

คุณสมบัติของมัลแวร์แต่ละประเภทนั้นบางทีก็มีความคล้ายกันอยู่บ้าง เช่น ไวรัสและเวิร์มอาจใช้เครือข่ายเพื่อเป็นช่องทางในการแพร่กระจาย อย่างไรก็ตามไวรัสพยายามที่จะฝังตัวในไฟล์ในขณะที่เวิร์มนั้นแค่พยายามจะก็อปปี้ตัวเองไปไว้หลายๆ ที่ ต่อไปนี้จะเป็นคุณสมบัติทั่วไปของมัลแวร์

2.6.1 คุณสมบัติของเป้าหมาย

ในขณะที่มัลแวร์พยายามจะโจมตีโฮสต์ใดๆ ระบบนั้นอาจต้องมีองค์ประกอบบางอย่างก่อนที่จะทำให้การโจมตีเป็นผลสำเร็จได้ ต่อไปนี้เป็นตัวอย่างขององค์ประกอบของระบบที่ต้องมี

1) ประเภทของอุปกรณ์ มัลแวร์บางตัวจะตั้งเป้าหมายไปที่อุปกรณ์เฉพาะบางอย่าง เช่น คอมพิวเตอร์ที่เป็นระบบวินโดวส์ แมคอินทอช หรือแม้กระทั่ง PDA (Personal Digital Assistant) แต่ในปัจจุบันไวรัสของ PDA นั้นแทบจะไม่มี
2) ระบบปฏิบัติการ มัลแวร์สามารถรันได้เฉพาะกับระบบปฏิบัติการหนึ่งเท่านั้น ยกตัวอย่างเช่น ไวรัส CIH หรือ Chernobyl ที่แพร่ในปี 1990 จะโจมตีเฉพาะคอมพิวเตอร์ที่รันวินโดวส์ 95 และ 98 เท่านั้น
3) แอพพลิเคชัน มัลแวร์ต้องอาศัยแอพพลิเคชันบางตัวเพื่อช่วยทำให้สามารถติดได้ เช่น ไวรัส LFM.926 ในปี 2002 สามารถโจมตีได้กับเฉพาะโปรแกรม Shockwave Flash (.swf) เท่านั้น

2.6.2 พาหะนำมัลแวร์

ถ้ามัลแวร์เป็นไวรัส มันจะพยายามทำให้เป้าหมายติดไวรัส จำนวนและประเภทของออบเจ็กต์ที่เป็นเป้าหมายได้นั้นมีหลากหลาย ต่อไปนี้เป็นตัวอย่างบางออบเจ็กต์ที่เป็นเป้าหมายของไวรัส

1) Executable file เป็นเป้าหมายคลาสสิกหรือดั้งเดิม ไวรัสสามารถแพร่กระจายโดยการฝังตัวเองไปกับโปรแกรมอื่น นอกเหนือจากไฟล์ที่สามารถเอ็กซีคิวต์ได้ ซึ่งจะมีนามสกุลเป็น .exe แล้วไฟล์อื่นที่สามารถรันได้ เช่น .com, .sys, .dll, .ovl, .ocx และ .prg ก็สามารถรันได้เช่นกัน
2) Script การโจมตีนี้อาจอาศัยภาษาสคริปต์เพื่อรันและทำให้ติดไวรัส ซึ่งภาษาสคริปต์ที่ว่านี้เช่น Visual Basic, JavaScript, AppleScript หรือ Perl เป็นต้น โดยไฟล์สคริปต์จะมีนามสกุลคือ .vbs, .js, .wsh และ .pl เป็นต้น
3) Macros แมโครเป็นภาษาสคริปต์ของแอพพลิเคชันบางตัว เช่น ไมโครซอฟท์ออฟฟิศ มัลแวร์จะอาศัยการรันแมโครสคริปต์นี้ในการแพร่กระจายหรือติดต่อไปยังไฟล์อื่นหรือระบบอื่น หรือ ทำอันตรายให้กับระบบที่ติด เช่น ไวรัสสามารถใช้ภาษาแมโครของไมโครซอฟท์เวิร์ดหรือ Lotus Ami Pro เพื่อสร้างผลกระทบให้กับระบบหรือโปรแกรมในรูปแบบต่างๆ เช่น การเปลี่ยนคำบางคำหรือการเปลี่ยนสี หรือบางทีสามารถฟอร์แมตฮาร์ดดิสก์ก็ได้
4) Boot Sector พื้นที่บางส่วนของฮาร์ดดิสก์หรือ CD-ROM เช่น MBR (Master boot record) หรือ Dos boot record อาจเป็นเป้าหมายก็ได้ เนื่องจากส่วนนี้สามารถรันโค้ดได้ เมื่อดิสก์ติดไวรัสในส่วนนี้แล้ว การแพร่กระจายก็สามารถเกิดขึ้นได้กับดิสก์นี้ใช้สำหรับบู๊ตระบบอื่น ถ้าไวรัสนั้นสามารถติดได้ทั้งไฟล์ทั่วไปและบู๊ตเซ็กเตอร์ ไวรัสประเภทนี้เรียกว่า มัลติพาร์ไทต์ไวรัส(Multipartite Virus)

2.6.3 กลไกการแพร่กระจาย

มัลแวร์อาจใช้หลากหลายวิธีในการแพร่กระจายตัวเองไปยังเครื่องอื่นๆ ต่อไปนี้เป็นตัวอย่างทั่วๆ ไปที่มัลแวร์มักใช้ในการแพร่กระจายตัวเอง

1) Removable media การแพร่กระจายแบบดั้งเดิมของไวรัสและแบบที่เกิดขึ้นมากที่สุด คือ การก็อปปี้ไฟล์ กลไกนี้เริ่มจากการใช้แผ่นฟล็อฟปี้ดิสก์ หลังจากนั้นก็เปลี่ยนมาเป็นเครือข่าย และปัจจุบันมันกำลังมองหาช่องทางใหม่ เช่น USB (Universal Serial Bus) และไฟร์ไวร์ (Firewire) อย่างไรก็ตามอัตราการแพร่กระจายโดยอาศัยมีเดียต่างๆ นี้ยังเป็นได้ไม่เร็วเท่ากับการแพร่กระจายโดยอาศัยเครือข่าย นอกจากนี้ความเสี่ยงในการติดนั้นยังมีอยู่เสมอเนื่องจากเป็นการยากที่จะป้องกันได้ร้อยเปอร์เซ็นต์

เพราะระบบยังคงต้องมีการแลกเปลี่ยนไฟล์กันอยู่เสมอ

2) Network shares เมื่อคอมพิวเตอร์ถูกเชื่อมต่อเข้ากับเครือข่ายก็จะมีการแชร์ไฟล์กันเพื่อความสะดวกในการแลกเปลี่ยนไฟล์ ซึ่งการแชร์ไฟล์ผ่านเครือข่ายนี้ก็กลายเป็นอีกช่องทางหนึ่งที่มัลแวร์ใช้ในการแพร่กระจายตัวเอง อีกทั้งการแพร่กระจายก็เป็นไปอย่างรวดเร็ว ถ้าเครือข่ายไม่มีระบบป้องกันและรักษาความปลอดภัยที่ดี การแชร์ไฟล์ผ่านเครือข่ายก็อาจทำให้การแพร่กระจายมัลแวร์ไปยังคอมพิวเตอร์จำนวนมากได้ในเวลาอันรวดเร็ว
3) Network scanning มัลแวร์อาจใช้เทคนิคนี้ในการสแกนเครือข่าย เพื่อค้นหาระบบที่มีจุดอ่อนหรือช่องโหว่และโจมตี ยกตัวอย่างเช่น กลไกนี้อาจส่งแพ็กเก็ตที่สามารถเจาะเข้าระบบที่มีช่องโหว่ผ่านทางพอร์ตเฉพาะ เพื่อค้นหาหรือทดสอบว่ามีระบบใดบ้างที่มีจุดอ่อนหรือช่องโหว่อยู่
4) Peer-to-peer networks หลักการทำงานของเพียร์ทูเพียร์เน็ตเวิร์กคือ เครื่องสองเครื่องใดๆที่ต้องการแชร์ข้อมูล ไฟล์ หรือโฟลเดอร์ แต่ละเครื่องจะต้องติดตั้งโปรแกรมไคลเอนท์โดยแต่ละเครื่องจะใช้การสื่อสารผ่านพอร์ตใดพอร์ตหนึ่ง เช่น พอร์ต 6800-6900 เป็นต้น แต่ถ้าองค์กรมีไฟล์วอลล์ผู้ดูแลระบบอาจจะไม่อนุญาตให้การสื่อสารผ่านพอร์ตนี้ได้อย่างไรก็ตามโปรแกรมเหล่านี้สามารถกำหนดให้พอร์ตที่เปิดใช้งานอยู่แล้ว และนั่นก็อาจเป็นช่องทางในการแพร่กระจายไวรัสหรือมัลแวร์ก็ได้
5) E-mail อีเมลกลายเป็นทางเลือกที่นิยมของการแพร่กระจายมัลแวร์ในปัจจุบัน เนื่องจากเป็นวิธีที่ง่าย และคนส่วนใหญ่ก็ใช้อีเมลในการสื่อสารกับผู้ใช้คนอื่นผ่านทางเครือข่ายและอินเตอร์เน็ตอยู่แล้ว โดยการใช้วิศวกรรมสังคมหรือใช้จิตวิทยาทำให้ผู้ใช้อีเมลเปิดไฟล์ที่แนบมาด้วยได้ง่าย ดังนั้น การแพร่กระจายที่รวดเร็วและทำลายคอมพิวเตอร์จำนวนมากในปัจจุบันจะใช้อีเมลเป็นสื่อในการแพร่กระจายมากที่สุด
6) Remote exploit มัลแวร์อาจพยายามใช้ช่องโหว่หรือจุดอ่อนจากเซอร์วิสหรือแอพพลิเคชันเพื่อแพร่กระจายตัวเอง พฤติกรรมอย่างนี้มักพบกันมากสำหรับมัลแวร์ประเภทเวิร์ม ยกตัวอย่างเช่น สแลมเมอร์เวิร์ม (Slammer worm) ใช้ประโยชน์จากช่องโหว่ในไมโครซอฟท์ SQL Server 2000 เวิร์มนี้จะทำให้เกิดบัฟเฟอร์โอเวอร์รัน (Buffer overrun) จนทำให้มันสามารถเขียนโค้ดลงบนบางส่วนของเมมโมรีของระบบ ซึ่งโค้ดนี้สามารถรันตัวเองได้เหมือนกับเป็นเซอร์วิสของ SQL Server บัฟเฟอร์โอเวอร์รัน หมายถึง สภาพที่เกิดจากการป้อนข้อมูลเข้าไปในบัฟเฟอร์มากกว่าที่บัฟเฟอร์จะสามารถรองรับได้ แฮคเกอร์มักนิยมใช้เทคนิคนี้ในการเจาะเข้า

ควบคุมระบบ ไมโครซอฟท์ได้ใช้เวลาหลายเดือนในการแก้ไข หรือปิดช่องโหว่นี้ หลังจากที่สแลมเมอร์ออกมาโจมตี อย่างไรก็ตามถึงแม้ว่าจะมีแพตช์ที่ใช้สำหรับปิดช่องโหว่เหล่านี้แล้วก็ตาม แต่ก็มีแค่ระบบเท่านั้นที่มีการดาวน์โหลดและอัปเดตแพตช์ ทำให้เวิร์มนี้ยังสามารถแพร่กระจายและทำลายระบบได้

2.6.4 เพย์โหลด

หลังจากที่มัลแวร์สามารถเข้ามาในโฮสต์ได้แล้วจะด้วยวิธีใดก็ตามโดยทั่วไปมันจะรันตัวเองหรือโปรแกรมที่ติดมาด้วย ส่วนที่รันบนโฮสต์นี้เรียกว่า เพย์โหลด (Payload) โดยเพย์โหลดเหล่านี้จะมีฟังก์ชันการทำงานหลายรูปแบบ ต่อไปนี้เป็นรูปแบบของเพย์โหลดที่มักพบเห็นทั่วไป

1) Backdoor เพย์โหลดประเภทนี้จะเปิดช่องโหว่เพื่อให้แฮกเกอร์สามารถเข้ามาใช้งานระบบได้ ซึ่งอาจทำให้แฮกเกอร์สามารถควบคุมระบบได้ หรืออาจเป็นแค่การสร้าง FTP เซิร์ฟเวอร์ให้สามารถถ่ายโอนไฟล์ผ่านพอร์ต 21 ได้ หรืออาจเป็นการเปิด Telnet เพื่อเป็นฐานในการโจมตีเครื่องอื่นได้ อย่างที่ได้กล่าวไว้ก่อนหน้าแล้วว่าแบ็คดอร์นั้นบางทีเรียกว่า โทรจันฮอร์สประเภท RAT (Remote Access Trojan) นั่นเอง
2) Data corruption or deletion หนึ่งในเพย์โหลดที่ทำให้มัลแวร์สามารถทำลายเครื่องที่ติดมากที่สุดคือ การลบหรือทำลายไฟล์ข้อมูล หรือทำให้ไฟล์นั้นใช้งานไม่ได้ นักพัฒนามัลแวร์มีทางเลือกอยู่สองทางเลือก ทางเลือกแรกคือ การทำให้เพย์โหลดถูกเอ็กซีคิวต์ให้เร็วที่สุด เพื่อทำลายเครื่องที่ติดไวรัสนั้น ทำให้ยากต่อการป้องกันการแพร่กระจายของมัน อีกทางเลือกหนึ่งคือ การปล่อยเพย์โหลดทิ้งไว้ในเครื่องที่ติดในรูปแบบของโทรจันฮอร์ส เพื่อเอ็กซีคิวต์ในเวลาต่อมา ทำให้มีช่วงเวลาสำหรับการตรวจจับมัลแวร์ประเภทนี้ได้
3) Information theft มัลแวร์อีกประเภทหนึ่งที่สร้างความกังวลให้กับผู้ใช้งานคือ มัลแวร์ ที่ออกแบบมาสำหรับการขโมยข้อมูลที่สำคัญจากระบบที่ติดมัลแวร์ ถ้ามัลแวร์สามารถหลีกเลี่ยง หรือทำลายระบบซีเคียวริตีของเครื่องได้ มัลแวร์ประเภทนี้จะสามารถขโมยและส่งข้อมูลกลับไปยังแฮกเกอร์ได้ เหตุการณ์นี้สามารถเกิดขึ้นได้หลายทาง เช่น การส่งข้อมูลอาจเป็นไปโดยอัตโนมัติโดยมัลแวร์จะส่งข้อมูลที่สำคัญ เช่น ยูสเซอร์และพาสเวิร์ดกลับไปทันที หรืออีกกลไกหนึ่งคือการสร้างสภาวะแวดล้อมในระบบเพื่ออนุญาตให้แฮกเกอร์สามารถล็อกอินเข้ามาในระบบเพื่อควบคุมหรือสามารถเข้าถึงไฟล์ซิสเต็มของเครื่องนั้นได้
4) Denial of Service (Dos) หนึ่งในเพย์โหลดที่ทำได้ง่ายและพบเห็นทั่วไปคือ การโจมตีแบบปฏิเสธการให้บริการ หรือ DoS การโจมตีแบบนี้คือ การทำให้เครื่องที่ถูกโจมตีโอเวอร์โหลดหรือหยุดให้บริการ เช่น เว็บเซิร์ฟเวอร์หรือเมลเซิร์ฟเวอร์ DoS มี

จุดประสงค์เพื่อทำให้เซอร์วิสบางตัวไม่สามารถให้บริการได้ในช่วงเวลาหนึ่ง

5) Distributed Denial of Service (DDoS) การโจมตีแบบ DDoS คือ การใช้เครื่องไคลเอนท์หลายๆ เครื่องช่วยกันโจมตีเครื่องที่เป็นเป้าหมายเครื่องหนึ่ง การทำแบบนี้ก็เพื่อเพิ่มความรุนแรงในการโจมตี เนื่องจากการโจมตีจากหลายเครื่องมักจะทำความเสียหายได้มากกว่าการใช้เครื่องเดียวในการโจมตีหนึ่งครั้ง รูปแบบของการโจมตีนั้นมีความหลากหลายในแต่ละการโจมตี แต่ส่วนใหญ่จะเกี่ยวข้องกับการส่งข้อมูลหรือแพ็กเก็ตจำนวนมากๆ ไปยังเครื่องเป้าหมายจนทำให้ไม่สามารถรองรับแพ็กเก็ตจำนวนมหาศาลนี้ได้ และทำให้เครื่องนั้นไม่สามารถให้บริการได้ ซึ่งการทำเช่นนี้เป็นการฟลัดแบนด์วิธ (Flood Bandwidth) ของเครือข่าย หรือการใช้แบนด์วิธให้หมดเพื่อจะได้ไม่ให้คนอื่นสามารถใช้ได้ การโจมตีแบบนี้เป็นการโจมตีที่ยากต่อการป้องกัน เนื่องจากโฮสต์ที่ใช้โจมตีนั้นส่วนใหญ่เป็นโฮสต์ที่ถูกโจมตี หรือเจาะเข้าไปเพื่อฝังโปรแกรมที่ใช้สำหรับโจมตีอีกทีหนึ่ง DDoS ส่วนใหญ่เกิดจากการใช้โปรแกรมบ็อทส์ (Bots) ซึ่งเป็นโปรแกรมที่ทำหน้าที่บางอย่างอัตโนมัติ เช่น เอ็กก์ดรอป (Eggdrop) เป็นบ็อทส์หนึ่งที่แฮกเกอร์ใช้สำหรับควบคุมคอมพิวเตอร์ผ่านช่องทาง IRC (Internet Relay Chat) เมื่อคอมพิวเตอร์นั้นถูกควบคุมได้แล้ว เครื่องนั้นก็จะกลายเป็นซอมบีส์ (Zombies) ซึ่งแฮคเกอร์สามารถส่งคำสั่งให้เครื่องนี้ทำงานบางอย่างโดยผู้ที่เป็นเจ้าของเครื่องนั้นอาจไม่รู้ตัว

6) Network DoS เพย์โหลดประเภทนี้พยายามที่จะโอเวอร์โหลดรีซอร์สของเครื่องที่โจมตี เช่น โพรเซสเซอร์ เมมโมรี เป็นต้น ซึ่งส่วนใหญ่จะถูกโจมตีแบบ SYN flood โดยแฮกเกอร์จะใช้โปรแกรมสำหรับส่งแพ็กเก็ตแบบ TCP SYN เพื่อไปทำให้คิวหรือบัฟเฟอร์ของเครื่องนั้นเต็ม ทำให้ผู้ใช้ทั่วไปไม่สามารถใช้บริการได้ E-mail bomb ก็เป็นการโจมตีหนึ่งที่ทำให้พื้นที่เก็บเมลเต็มโดยการส่งเมลที่มีขนาดใหญ่มากไปยังที่อยู่อีเมลหนึ่ง เพื่อทำให้เมลหยุดทำงานหรือไม่สามารถรับเมลจากที่อื่นได้อีก

7) System shutdowns การทำให้เครื่องที่ถูกโจมตีถูกชัตดาวน์หรือล้มได้นั้น สามารถทำได้โดยการทำลายเซอร์วิสของระบบหนึ่งเซอร์วิสหรือมากกว่า การโจมตีประเภทนี้มัลแวร์จะค้นหาช่องโหว่หรือจุดอ่อนของแอพพลิเคชันหรือระบบปฏิบัติการ ซึ่งทำให้ระบบชัตดาวน์ลงได้

8) Bandwidth flooding เซอร์วิสส่วนใหญ่ที่ให้บริการจะใช้ช่องการเชื่อมต่อเข้ากับเครือข่าย ซึ่งช่องนี้จะมีข้อจำกัดเกี่ยวกับอัตราการถ่ายโอนข้อมูลหรือแบนด์วิธ ถ้ามัลแวร์สามารถส่งแพ็กเก็ตจนเต็มช่องแบนด์วิธนี้ได้ ผู้ใช้ทั่วไปก็จะไม่สามารถ

เข้ามาใช้เซอร์วิสนี้ได้ เนื่องจากแบนด์วิธถูกใช้ไปหมดแล้ว

9) Service disruption เพย์โหลดประเภทนี้อาจทำให้เกิด DoS ได้ ยกตัวอย่างเช่น ถ้าการโจมตี DNS เซิร์ฟเวอร์ ทำให้เครื่องนั้นไม่สามารถให้บริการ DNS ได้ก็เป็นการโจมตีแบบ DoS ได้เช่นกันถึงแม้ว่าบริการอื่นยังคงใช้งานได้ตามปกติ แต่บริการอื่นที่ว่านี้อาจต้องอาศัย DNS ไคลเอนท์จึงจะสามารถใช้บริการนี้ได้ เช่น เว็บเมล์ เป็นต้น

2.6.5 การจุดชนวน

ก่อนที่จะระเบิดได้นั้นจะต้องมีการจุดชนวนสำหรับไวรัสหรือมัลแวร์อื่นๆ ก็เหมือนกันก็จะต้องมีการจุดชนวนเพื่อให้ไวรัสเริ่มทำงานเพื่อทำลายระบบ หรือเริ่มกระบวนการแพร่กระจายหรือการส่งเพย์โหลดไปยังเครื่องอื่นๆ การจุดชนวนส่วนใหญ่อาจเกิดได้ดังนี้

1) Manual execution การจุดชนวนประเภทนี้คือ การที่ผู้ใช้รันโปรแกรมที่เครื่องโดยตรง ซึ่งอาจจะทำโดยไม่รู้ตัวหรือเป็นการหลอกให้รันโปรแกรม โดยอาจใช้วิธีวิศวกรรมสังคมหรือจิตวิทยา มัลแวร์โดยทั่วไปจะใช้บางรูปแบบของวิศวกรรมสังคมในการหลอกให้ผู้ใช้รันโปรแกรมมัลแวร์เหล่านั้นเอง วิธีนี้เป็นวิธีที่ง่ายมาก ตัวอย่างเช่น ไวรัสที่ใช้อีเมลเป็นสื่อในการแพร่กระจาย โดยไวรัสจะสร้างหัวข้อเรื่องของเมล์ที่ทำให้หน้าสนใจต่อผู้อ่านเพื่อให้เปิดเมลนั้น หรือการสปูฟิง (Spoofing) อีเมล์เพื่อให้ผู้รับเชื่อว่าอีเมลนั้นมาจากแหล่งที่เชื่อถือได้ ตัวอย่างเช่น เวิร์ม ดูมารุ (Dumaru worm) ซึ่งเริ่มแพร่กระจายในปี 2003 โดยเวิร์มจะแก้ไขชื่อผู้ส่ง (From :) เป็นว่าอีเมลนี้ส่งมาจาก security@microsoft.com
2) Semi-automatic execution การจุดชนวนประเภทนี้เริ่มจากการที่ผู้ใช้รันโปรแกรมมัลแวร์เอง แล้วหลังจากนั้นโปรแกรมก็จะทำงานต่อโดยอัตโนมัติ
3) Automatic execution มัลแวร์ประเภทนี้จะรันตัวมันเองโดยอัตโนมัติโดยไม่ต้องอาศัยผู้ใช้เลย
4) Time bomb การจุดชนวนประเภทนี้มัลแวร์จะรันเมื่อช่วงระยะเวลาหนึ่งหลังจากที่เครื่องติดไวรัสแล้ว หรือในวันเวลาใดเวลาหนึ่ง ยกตัวอย่างเช่น เวิร์มมายดูมดอตบี (MyDoom.B worm) กุมภาพันธ์ 2004 และโจมตีเว็บไซต์ของ SCO Group ในวันที่ 1 กุมภาพันธ์ 2004 อย่างไรก็ตามแบ็คดอร์ของไทม์บอมบ์ยังคงทำงานอยู่หลังจากเวลาดังกล่าว
5) Conditional การจุดชนวนประเภทนี้เริ่มเมื่อสภาพแวดล้อมตรงตามเงื่อนไขที่กำหนด ยกตัวอย่างเช่น เมื่อมีการเปลี่ยนชื่อไฟล์ เมื่อมีการกดคีย์บอร์ดบางคีย์ หรือเมื่อเปิดบางโปรแกรม มัลแวร์ประเภทนี้บางทีก็เรียกว่า ลอจิกบอมบ์ (Logic bomb)

2.7 วงจรชีวิตของมัลแวร์

การศึกษาและเข้าใจวงจรชีวิตของมัลแวร์อาจช่วยให้สามารถป้องกันการโจมตีจากมัลแวร์ได้ง่ายขึ้น วงจรชีวิตของมัลแวร์นั้นเริ่มจากการออกแบบคิดค้นไปจนถึงการกำจัดออกจากระบบ ซึ่งมีช่วงระยะดังนี้

1) การค้นพบช่องโหว่ การพัฒนามัลแวร์จะเริ่มเมื่อมีการค้นพบวิธีใหม่ๆ ในการโจมตี การค้นพบช่องโหว่หรือจุดอ่อนของบางโปรแกรม และได้มีการเผยแพร่ในสังคมของแฮกเกอร์ ในช่วงนี้นักเขียนมัลแวร์ก็จะช่วยพัฒนาและค้นหาวิธีการที่จะใช้ประโยชน์จากจุดอ่อนหรือช่องโหว่นี้
2) การพัฒนา การเขียนมัลแวร์จะต้องอาศัยความเข้าใจภาษาแอสเซมบลี (Assembly Language) เข้าใจการทำงานของระบบคอมพิวเตอร์ที่จะโจมตี อย่างไรก็ตามปัจจุบันมีเครื่องมือที่ช่วยในการพัฒนามัลแวร์ได้ง่ายขึ้น ประกอบกับแหล่งความรู้และความช่วยเหลือจากอินเตอร์เน็ตนั้น ทำให้การพัฒนามัลแวร์เป็นเรื่องง่าย
3) การแพร่ระบาด หลังจากมัลแวร์ได้ถูกพัฒนาเสร็จแล้วและได้ประกาศและปล่อยให้สาธารณะทราบ ส่วนใหญ่มันจะแพร่ระบาดไปยังโฮสต์ต่างๆ ที่จะโจมตีก่อนที่จะทำลายระบบเลยทันที ถึงแม้ว่าจะมีโปรแกรมมัลแวร์ที่เป็นที่รู้จักกันมากมาย แต่มีเพียงบางส่วนเท่านั้นที่ถูกปล่อยออกมาสู่สาธารณะ ไวรัสที่เขียนขึ้นส่วนใหญ่จะไม่ถูกปล่อยให้ออกสู่สาธารณเลย ซึ่งไวรัสกลุ่มนี้เป็นที่รู้จักกันในนามสวนสัตว์ไวรัส (Zoo viruses)
4) การทำลาย หลังจากที่มัลแวร์สามารถเข้ามาในระบบแล้วขั้นตอนต่อไปคือ การปล่อยให้เพย์โหลดทำงาน ถ้าโค้ดเพย์โหลดถูกกำหนดให้รันตามเงื่อนไข ช่วงนี้ก็อยู่ในช่วงของการรอให้เป็นไปตามเงื่อนไขนั้น ยกตัวอย่างเช่น เพย์โหลดของมัลแวร์บางประเภทจะถูกรันก็ต่อเมื่อผู้ใช้ได้ทำอะไรบางอย่างหรือถึงเวลาที่กำหนด ส่วนมัลแวร์ที่ไม่ต้องมีเงื่อนไขก็อาจปล่อยให้เพย์โหลดทำงานทันทีที่ระบบนั้นติดไวรัส ยกตัวอย่างเช่น เพย์โหลดที่เก็บล็อคการทำงานของระบบ เมื่อมัลแวร์ติดเครื่องนั้นเพย์โหลดก็จะทำงานทันทีเพื่อเก็บสะสมรวบรวมข้อมูลที่ต้องการ
5) การตรวจพบและแจ้งเตือน เมื่อเวลาผ่านไปช่วงหนึ่งบริษัทผู้ผลิตซอฟต์แวร์ป้องกันไวรัสก็จะค้นพบมัลแวร์หรือไวรัสตัวใหม่นี้ ส่วนใหญ่นั้นขั้นตอนนี้อาจเกิดขึ้นก่อนขั้นตอนที่ 4 หรือ 3 แต่ก็ไม่เสมอไป ช่วงนี้ก็เป็นช่วงเวลาที่จะหาทางป้องกันและตรวจจับไวรัสประเภทนี้
6) การตรวจจับ หลังจากที่ภัยจากมัลแวร์ได้ถูกเปิดเผยแล้ว ซอฟต์แวร์ป้องกันไวรัสก็จะวิเคราะห์และพัฒนาโค้ดที่จะตรวจจับมัลแวร์นี้ หลังจากที่ค้นพบวิธีตรวจจับก็จะอัปเดตซิกเนเจอร์เพื่อให้โปรแกรมป้องกันไวรัสที่ใช้งานอยู่แล้วสามารถตรวจจับไวรัสตัวใหม่นี้

ได้ด้วย ช่วงเวลานี้เป็นช่วงเวลาที่สำคัญที่จะช่วยในการควบคุมการโจมตีและแพร่ระบาดของไวรัสได้

7) การป้องกันและกำจัด หลังจากที่ไวรัสซิกเนเจอร์ได้เผยแพร่ต่อสาธารณะ ขั้นตอนต่อไปก็เป็นความรับผิดชอบของผู้ใช้คอมพิวเตอร์ที่จะต้องอัปเดตซิกเนเจอร์เป็นประจำ เพื่อป้องกันการโจมตี หรือกำจัดไวรัสตัวใหม่นี้ถ้าระบบนั้นได้ติดไวรัสแล้ว

2.8 การแบ่งประเภทมัลแวร์

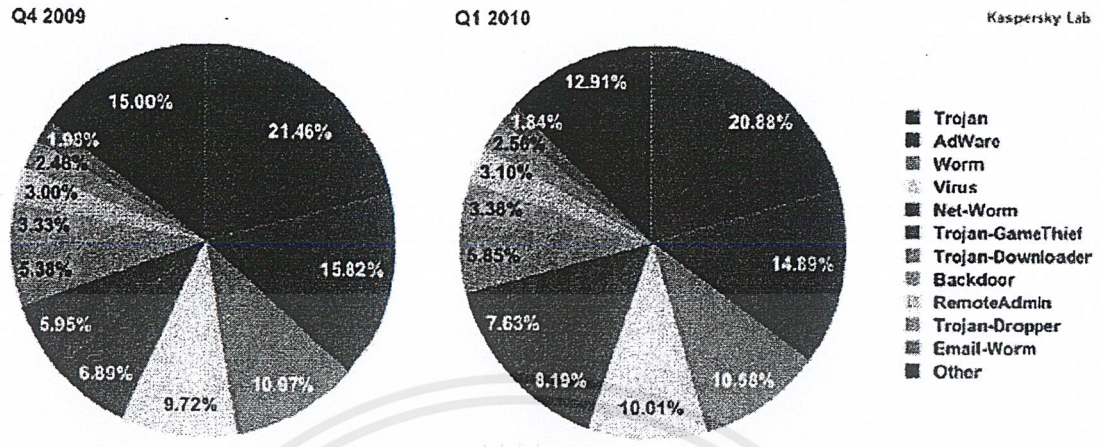
2.8.1 มัลแวร์แบ่งตามชนิดของมัลแวร์

1) มัลแวร์ที่มีจุดประสงค์เพื่อทำให้ติดเชื้อ เช่น Virus, Worm
2) มัลแวร์ที่มีจุดประสงค์เพื่อการปกปิด หรือซ่อนตัว เช่น Key Logger, Spyware
3) มัลแวร์ที่มีจุดประสงค์เพื่อการแสวงหากำไร เช่น Trojan, Rootkit, Backdoor
4) มัลแวร์ที่มีจุดประสงค์เพื่อการขโมยความลับ หรือข้อมูล เช่น Spyware, Key Logger
5) มัลแวร์ที่ทำงาน โดยอาศัยช่องโหว่ เช่น Worm

2.8.2 มัลแวร์แบ่งตามพฤติกรรมการแพร่ระบาด

1) มัลแวร์ที่แพร่ระบาดโดยไม่อาศัยพาหะในการแพร่ (Independence) เช่น เวิร์มจะแพร่ระบาด โดยการคัดลอกตัวเองไปตามเครือข่าย
2) มัลแวร์ที่แพร่ระบาดโดยอาศัยพาหะในการแพร่ (Dependence) เช่น ไวรัสจะคัดลอกตัวเองติดไปกับไฟล์ และเมื่อผู้ใช้งานไปดับเบิ้ลคลิกหรือรันโปรแกรมไฟล์นั้นๆ ไวรัสก็จะทำงานทันที
3) มัลแวร์ที่แพร่ระบาดโดยกึ่งอาศัยพาหะในการแพร่ (Semi-dependence) โดยใช้เทคนิค Social Engineering

2.9 สถิติมัลแวร์ย้อนหลัง



รูป 2.8 สถิติมัลแวร์ย้อนหลัง

จากรูป 2.8 แผนภาพด้านซ้ายแสดงสถิติของมัลแวร์ที่ถูกตรวจจับได้ในช่วงเดือนตุลาคม-ธันวาคมในปี 2009 และแผนภาพด้านขวาแสดงสถิติของมัลแวร์ที่ถูกตรวจจับได้ในช่วงเดือนมกราคม-ธันวาคมในปี 2010 จากแผนภาพทั้งสองจะเห็นได้ว่าโทรจันเป็นมัลแวร์ที่ถูกพบได้มากที่สุด รองลงมาคือแอดแวร์ เวิร์ม และไวรัสตามลำดับ โดยผลการตรวจสอบจาก Kaspersky Lab

2.10 การค้นหาและกำจัดมัลแวร์

มัลแวร์เป็นภัยคุกคามที่สร้างความเสียหายให้กับระบบได้อย่างรุนแรงมาก ดังนั้นโปรแกรมป้องกันมัลแวร์จึงเป็นปัจจัยที่สำคัญมากที่จะช่วยให้เครื่องปลอดภัยจากการคุกคามของมัลแวร์คอมพิวเตอร์ ในปัจจุบันโปรแกรมป้องกันมัลแวร์ก็มีมากขึ้น เทคนิคในการตรวจจับมัลแวร์ได้ถูกพัฒนาขึ้นเพื่อตรวจจับมัลแวร์แตกต่างกันออกไป ซึ่งเทคนิคในการตรวจจับมัลแวร์โดยทั่วไป แบ่งได้ 4 เทคนิคคือ

2.10.1 การตรวจหา (Scanning)

เป็นเทคนิคที่ใช้ตัวตรวจหาเข้าไปค้นหาไฟล์ที่ถูกบ่งบอกว่าถูกไวรัสแฝงตัวอยู่ในหน่วยความจำ ส่วนเริ่มต้นในการบูต (Boot sector) และไฟล์ที่ถูกเก็บอยู่ในฮาร์ดดิสก์ โดยใช้หลักการ Checksum ซึ่งมีวิธีการทำงานคือ ในไฟล์ทุกไฟล์จะมีส่วนที่เก็บข้อมูลว่ามีจุดเริ่มต้นจุดสิ้นสุดของไฟล์ที่ตำแหน่งใด ตามด้วยข้อมูลของไฟล์ และปิดท้ายด้วยค่า Checksum ตัวตรวจหาจะคำนวณหาค่า Checksum ของแต่ละไฟล์แล้วนำไปทำการเปรียบเทียบกับค่า Checksum ของไฟล์นั้นๆ ดังนั้นถ้าไฟล์ใดถูกไวรัสแฝงตัวก็จะทำให้ค่า Checksum ที่คำนวณได้จะไม่เท่ากับค่า

Checksum ที่เป็นข้อมูลของไฟล์ดังกล่าว โปรแกรมป้องกันไวรัสทั่วๆ ไป จะมีวิธีการตรวจหา 2 ชนิดคือ

1) การตรวจหาชนิด On - Access เป็นวิธีการตรวจหาไฟล์ก่อนที่จะถูกโหลดเข้าหน่วยความจำ เพื่อทำการเอ็กซิคิวต์
2) การตรวจหาชนิด On - Demand เป็นวิธีการตรวจหาในหน่วยความจำหลักส่วนเริ่มต้นในการบูต และฮาร์ดดิสก์ ผู้ใช้งานยังสามารถเรียกใช้งานวิธีการตรวจหาชนิดนี้ตามความต้องการได้

ข้อดีของเทคนิคนี้คือตัวตรวจหาสามารถพบไวรัสก่อนที่จะทำการเอ็กซิคิวต์

2.10.2 การตรวจสอบความคงอยู่ (Integrity Checking)

เทคนิคนี้อาศัยตัวตรวจสอบความคงอยู่ (Integrity Checker) ที่เก็บข้อมูลความคงอยู่ (Integrity Information) ของไฟล์สำคัญไว้สำหรับเปรียบเทียบ ตัวอย่างข้อมูลเช่น ขนาดไฟล์ เวลาแก้ไขครั้งล่าสุด และค่า Checksum เป็นต้น ส่วนมากจะใช้ค่าของ Checksum ในการเปรียบเทียบเมื่อมีไฟล์เปลี่ยนแปลงที่มีสาเหตุอันเนื่องจากไวรัสหรือความผิดพลาดใดๆ จนทำให้ข้อมูลความคงอยู่ต่างจากข้อมูลเดิมที่เคยเก็บไว้ ระบบก็จะแจ้งให้ผู้ใช้งานทราบถึงความผิดปกติและยังสามารถมีทางเลือกให้ผู้ใช้สามารถกู้ไฟล์ข้อมูลดังกล่าวคืนไปเป็นไฟล์ก่อนที่จะติดไวรัสได้

ข้อดีของเทคนิคนี้คือ เป็นเทคนิคเดียวที่จะตรวจสอบว่ามีไวรัสทำลายไฟล์หรือไม่ และเกิดความผิดพลาดน้อย ตัวตรวจสอบความคงอยู่ในปัจจุบันมีความสามารถที่จะตรวจจับการทำลายข้อมูลชนิดต่างๆ ได้ เช่นไฟล์ไม่สมบูรณ์ (Corruption) และยังสามารถกู้ไฟล์คืนได้

2.10.3 การตรวจจับไวรัสโดยใช้การวิเคราะห์พฤติกรรม (Heuristic)

เป็นเทคนิคทั่วไปที่นิยมใช้ในการตรวจจับไวรัส โดยจะเปรียบเทียบการทำงานของไวรัสกับกฎ Heuristic (Rules Based System) และชุดกฎ Heuristic ถูกพัฒนาให้สามารถแยกแยะพฤติกรรมการทำงานว่าเป็นการทำงานของไวรัสหรือไม่ มีการเก็บข้อมูลของไวรัสที่รู้จักเพื่อใช้ในการจับคู่แพตเทิร์น และชุดกฎนี้ถูกพัฒนาโดยผู้พัฒนาโปรแกรมป้องกันไวรัส ยกตัวอย่างวิธีการตรวจจับไวรัสชนิดนี้เช่น โปรแกรมป้องกันไวรัสรู้จักพฤติกรรมการทำงานของไวรัสทั่วไป (เช่น การอ่าน/เขียนลงใน Master Boot Record ซึ่งโปรแกรมทั่วๆ ไปจะไม่ทำเช่นนี้) เมื่อโปรแกรมป้องกันไวรัสตรวจพบว่ามีการทำงานที่ผิดปกติขึ้นในเครื่อง โปรแกรมป้องกันไวรัสจะใช้กฎ Heuristic เปรียบเทียบกับลักษณะดังกล่าว เพื่อที่จะระบุว่าเป็นพฤติกรรมการทำงานของไวรัสชนิดใด

ข้อดีของเทคนิคนี้คือมีความยืดหยุ่นในการตรวจจับ และสามารถรู้จักไวรัสชนิดใหม่ๆ ได้เอง

2.10.4 การตรวจจับไวรัสโดยการดักจับ (Interception)

เทคนิคนี้จะเริ่มต้นด้วยการที่โปรแกรมป้องกันไวรัสจะสร้าง Virtual Machine ที่มีความอ่อนแอมากไว้ภายในเครื่อง คอยล่อให้โปรแกรมประเภทไวรัสโจมตี และยังมีหน้าที่เฝ้าดูว่ามีไวรัสหรือโปรแกรมใดบ้างที่มีพฤติกรรมผิดปกติน่าสงสัยเข้ามาทำงานใน Virtual Machine ตัวอย่างเช่น มีโปรแกรมที่ทำการติดตั้งตัวเอง รวมทั้งมีการส่งการร้องขอที่ผิดปกติออกมาเพื่อทำให้เครื่องทำงานผิดพลาด เป็นต้น โปรแกรมที่ผิดปกติหรือน่าสงสัยนี้อาจจะเป็นไวรัสก็ได้

ข้อดีของการใช้เทคนิคนี้คือจะหยุดการทำงานของโปรแกรมไวรัสที่พยายามที่จะฝังตัวในหน่วยความจำได้ดี

เทคนิคในการตรวจจับไวรัสทั้ง 4 เทคนิค เป็นเทคนิคพื้นฐานที่พัฒนาขึ้นเพื่อใช้ในการตรวจจับไวรัส โดยโครงงานนี้นำหลักการการตรวจจับไวรัสโดยใช้การวิเคราะห์พฤติกรรมมาใช้ เนื่องจากโปรแกรมจะมีความยืดหยุ่นในการตรวจจับ และสามารถรู้จักไวรัสชนิดใหม่ๆ ได้เองโดยที่ไม่ต้องทำการอัปเดตข้อมูลไวรัส

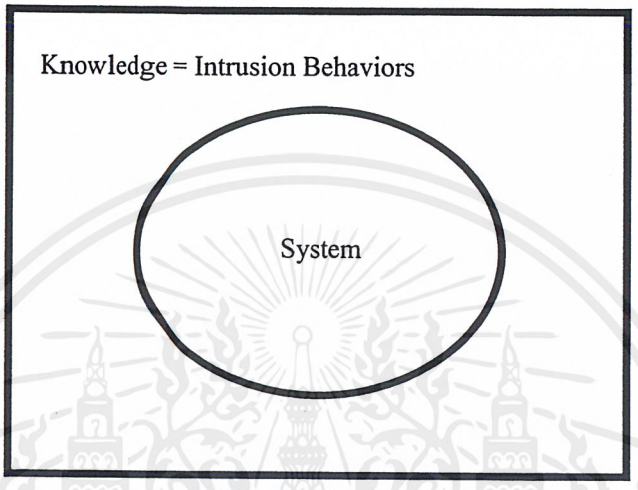
2.11 ระบบตรวจจับการบุกรุก

ระบบตรวจจับการบุกรุก คือ ระบบตรวจจับความผิดปกติต่างๆ ที่เกิดขึ้นในระบบ โดยอุดมคติแล้วจะต้องทราบการทำงานทั้งที่เป็นปกติ และผิดปกติทั้งหมดที่เกิดขึ้นในระบบ แล้วทำการแจ้งเตือนเมื่อเกิดเหตุการณ์ที่ผิดปกติขึ้นในระบบ และคือ ระบบตรวจจับความผิดปกติต่างๆ ที่เกิดขึ้นในระบบและทำการป้องกันการโจมตีหลายๆ รูปแบบได้โดยอัตโนมัติ และจะมีการทำงานสามส่วนหลักๆ คือการเก็บข้อมูล การวิเคราะห์ข้อมูล และการตอบสนองต่อการทำงานในรูปแบบต่างๆ

ในการวิเคราะห์ข้อมูลจะสามารถวิเคราะห์ข้อมูลได้ใน 2 ลักษณะคือ Anomaly Detection และ Misuse Detection โดยการทำงานของ Anomaly Detection จะทำการเก็บข้อมูลเกี่ยวกับการทำงานต่างๆ ที่เป็นการทำงานปกติไว้แล้วทำการเปรียบเทียบกับเหตุการณ์ที่เกิดขึ้นกับข้อมูลที่มีอยู่ ถ้าเปรียบเทียบแล้วมีความแตกต่างกันจะแปลว่าเกิดการทำงานที่ผิดปกติขึ้นจึงดำเนินการตอบสนองในรูปแบบต่างๆ สำหรับ Misuse Detection จะเป็นการเก็บข้อมูลการทำงานที่ผิดปกติไว้ หากเหตุการณ์ใดตรงกับข้อมูลที่เก็บไว้จะหมายความว่าเกิดเหตุการณ์ผิดปกติขึ้น

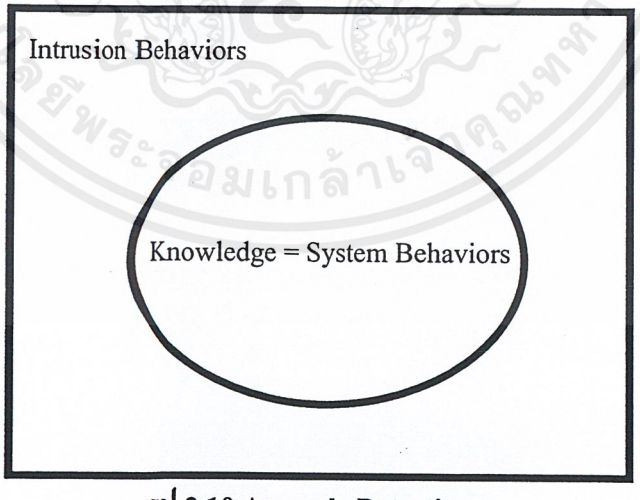
1) Misused Detection เป็นกระบวนการที่จะเก็บข้อมูลการทำงานที่ผิดปกติไว้ เมื่อมีการทำงานในระบบที่ตรงกับ Knowledge จะทำการ แจ้งเตือนให้ทราบว่าเกิดการบุกรุกขึ้น หลักการคือ หาว่าอะไรคือองค์ประกอบของการบุกรุกหรือสิ่งผิดปกติ แล้วพยายามตรวจจับจุดนั้น วิธีการจะใช้พฤติกรรมต้นแบบ (Behavior Model) ที่ผิดปกติเปรียบเทียบกับพฤติกรรมที่เกิดขึ้น หากพฤติกรรมที่เกิดขึ้นนั้นไม่เหมือนกับพฤติกรรมต้นแบบแสดงว่าพฤติกรรมนั้นไม่ผิดปกติ แต่ถ้าพฤติกรรมที่เกิดขึ้นมีลักษณะการกระทำเหมือนกับ

พฤติกรรมผิดปกติต้นแบบ ก็จะแจ้งเตือนผู้ใช้ว่าการทำงานนั้นผิดปกติ ที่เลือกใช้หลักการ Misuse Detection เนื่องจากการทำงานโดยทั่วไปที่เป็นปกตินั้น มีอยู่มากมายทั่วไป พฤติกรรมที่ผิดปกติมีน้อย จึงสามารถทำพฤติกรรมต้นแบบได้ครอบคลุม ข้อดีของ Misuse Detection คือ ง่าย เร็ว โอกาสผิดพลาดมีน้อย แต่ก็มีข้อเสียที่ต้องรู้จักวิธีการบุกรุกถึงจะตรวจจับเจอ และมีโอกาสที่ระบบจะถูกลวงได้ง่าย



รูป 2.9 Misuse Detection

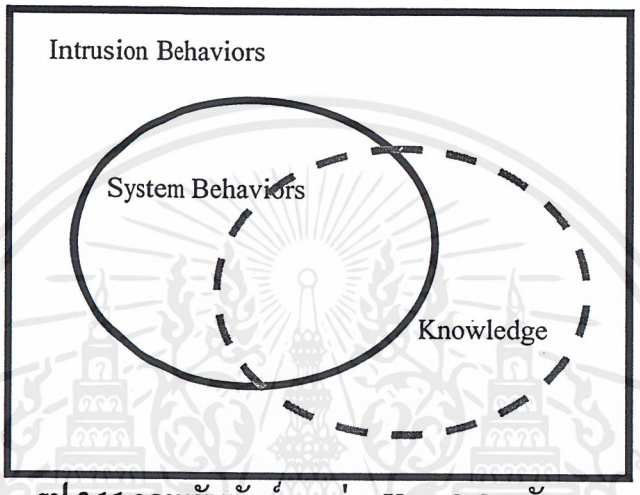
2) Anomaly Detection เป็นกระบวนการที่จะเก็บข้อมูลการทำงานที่เป็นปกติไว้ เมื่อมีการทำงานในระบบที่แตกต่างจาก Knowledge จะทำการแจ้งเตือนให้ทราบว่าเกิดการบุกรุกขึ้น



รูป 2.10 Anomaly Detection

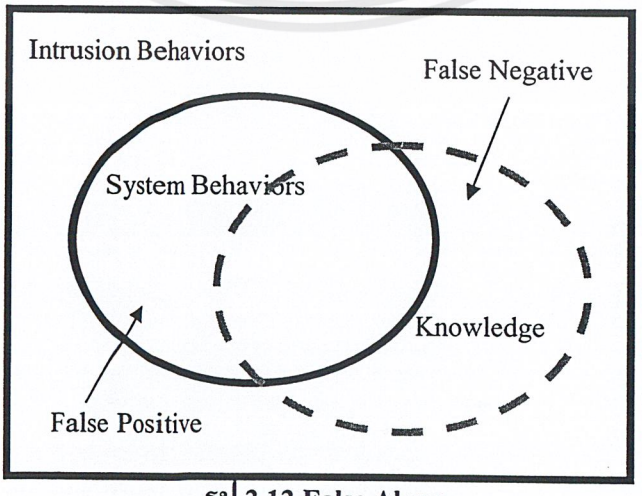
ถึงแม้ว่าการดำเนินการสร้างขอบเขตทั้งสองแนวทางจะมีกรรมวิธีที่แน่นอน แต่ในโลกของความเป็นจริงเราไม่สามารถทำให้ Knowledge และ System Behavior เป็นข้อมูลชุดเดียวกันได้

เนื่องจากที่มาของ Knowledge และความซับซ้อนในการทำงาน เช่น ในกรณีของ Misuse Detection อาจมี Knowledge ของการทำงานที่ผิดปกติไม่ครบถ้วน หรือมีความผิดพลาดทำให้ Knowledge มองว่าการทำงานของระบบบางอย่างเป็นการโจมตี หรือในกรณีของ Anomaly Detection การเก็บข้อมูล Knowledge อาจมีการผิดพลาดเช่น เก็บข้อมูลการทำงานที่ผิดปกติ เป็นการทำงานที่เป็นปกติ หรือไม่มีการเก็บข้อมูลการทำงานที่เป็นปกติของระบบใน Knowledge ไม่ว่าจะเป็นโมเดลในการสร้างขอบเขต แบบใดก็ตาม ความผิดพลาดดังกล่าวจะทำให้เกิดความเหลื่อมล้ำของขอบเขต



รูป 2.11 ความสัมพันธ์ระหว่าง Knowledge กับระบบ

จากความเหลื่อมล้ำระหว่าง Knowledge กับระบบทำให้เกิดความผิดพลาดในการตรวจสอบ เรียกว่า "False Alarm" False Alarm เป็นความผิดพลาดในการแจ้งเตือนเมื่อไม่มี Knowledge ครอบคลุมระบบทั้งหมด จึงเกิดกรณี ความผิดปกติขึ้น 2 กรณีคือ False Positive และ False Negative โดย False Positive คือการที่มีการแจ้งเตือนว่าเกิดการบุกรุกขึ้น แต่ในระบบไม่ถูกบุกรุก และ False Negative คือการที่ไม่มีการแจ้งเตือนว่าเกิดการบุกรุกขึ้นแต่ในระบบถูกบุกรุก



รูป 2.12 False Alarm

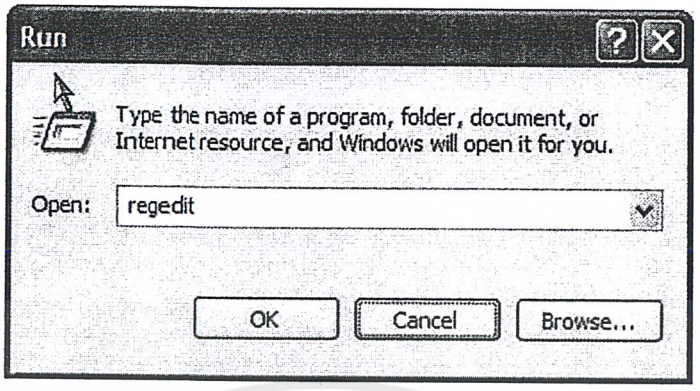
ในการตรวจจับความผิดปกติต่างๆ ในระบบจะสามารถตรวจสอบได้หลายรูปแบบ ขึ้นอยู่กับแหล่งที่มาของข้อมูลที่ใช้ในการวิเคราะห์ ระยะเวลาในการทำงาน กระบวนการที่ใช้ในการตรวจจับและผลลัพธ์ในการทำงาน

ระยะเวลาในการทำงานแบ่งออกได้เป็น 2 รูปแบบคือ แบบ Realtime และแบบ Batch โดยแบบ Realtime คือระบบที่ทำงานอยู่ตลอดเวลา และส่งผลลัพธ์ในการทำงานทันที หลังจากพบความผิดปกติ และแบบ Batch คือระบบที่ทำงานเป็นช่วงเวลาเช่น ทุกๆ เที่ยงคืน ทุกวันเสาร์ หรือช่วงเวลาใดๆ ที่ผู้ใช้กำหนด ซึ่งการทำงานจะเป็นแบบ Realtime หรือ Batch นั้นขึ้นอยู่กับการออกแบบ และวัตถุประสงค์ของการตรวจจับ แต่โดยทั่วไปแล้ว มีปัจจัย 2 ข้อที่ทำให้ทำงานเป็นแบบ Realtime หรือ Batch นั่นคือความเร็วในการตอบสนอง และ ทรัพยากรที่ต้องใช้ในการวิเคราะห์ ในบางสถานการณ์จำเป็นต้องทราบและตอบสนองทันทีเมื่อเกิดปัญหาขึ้น และในบางครั้งไม่จำเป็นต้องมีการตรวจจับอยู่ตลอดเวลาจะทำงานแบบ Batch เป็นต้น

ดังนั้นจากการศึกษากระบวนการที่ใช้ในการตรวจจับแล้วนำมาประยุกต์ใช้กับโปรแกรมตรวจจับมัลแวร์แล้วนั้นสามารถแบ่งการตรวจจับออกเป็น 2 รูปแบบ คือ แบบ Misuse Detection และแบบ Anomaly Detection ซึ่ง Misuse Detection เป็นการตรวจจับความผิดปกติต่างๆ ในระบบโดยมี Signature ของความผิดปกติต่างๆ ในระบบและจะแจ้งเตือนต่อผู้ดูแลระบบเมื่อมีข้อมูลใดๆ Match กับ Signature ยกตัวอย่างเช่น Network Intrusion Detection และ Virus Scan เป็นต้น สำหรับ Anomaly Detection เป็นการตรวจจับความผิดปกติของระบบโดยจะมีการเก็บ Profile การทำงานที่เป็นปกติไว้แล้วทำการเปรียบเทียบข้อมูลที่รับเข้ามากับ Profile หากมีข้อมูลใดแตกต่างจาก Profile จะถือว่ามีการบุกรุกเกิดขึ้น ผลลัพธ์ในการทำงานส่วนใหญ่จะเป็นการแจ้งเตือนและเก็บข้อมูลการแจ้งเตือนในรูปแบบต่างๆ เช่น การแสดง Alert Message ในหน้าจอ เป็นต้น

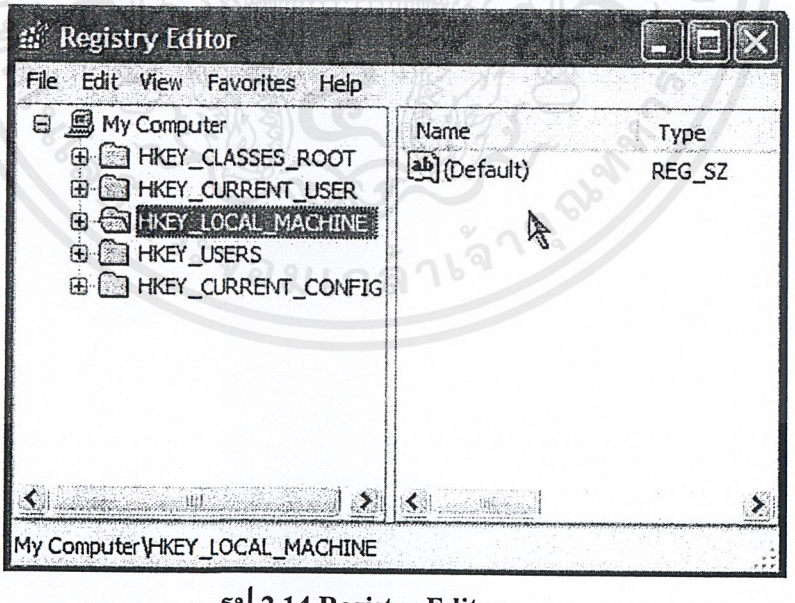
โปรแกรมสแกนไวรัสถือว่าเป็นระบบตรวจจับผู้บุกรุกเช่นเดียวกัน ไม่ว่าจะเป็นการสแกนไฟล์ระบบในเครื่องคอมพิวเตอร์ หรือใน Mailbox แต่วัตถุประสงค์ไม่ได้ใช้ในการตรวจจับ Hacker แต่เป็นการตรวจจับโปรแกรมหรือการทำงานที่ผิดปกติต่างๆ ในระบบ หลักการทำงานจะคล้ายกับ Host Based Intrusion Detection System สภาพแวดล้อมในการตรวจจับคือไฟล์ต่างๆ ในระบบรวมถึง Registry โปรแกรมดังกล่าวมีการทำงานแบบ Batch หรือ Schedule มี Knowledge คือ Virus Signature การทำงานเป็นแบบ Misuse Detection การทำงานของโปรแกรมสแกนไวรัสเป็นแบบ IDS ดังนั้นปัญหา False Alarm จึงเกิดได้เสมอ

2.12 Windows Registry



รูป 2.13 Run Registry

รีจิสทรี คือ ฐานข้อมูลส่วนกลางที่วินโดวส์ใช้เพื่อเก็บค่าทุกอย่างที่เกี่ยวกับวินโดวส์ โดยรีจิสทรีถือ กำเนิดมาจากแนวความคิดในการจัดเก็บไฟล์ INI ของวินโดวส์ 3.1 ซึ่งไฟล์ INI ก็มีข้อจำกัดหลายอย่าง และ OLE เริ่มมีความซับซ้อน มากขึ้นไมโครซอฟท์จึงได้สร้างโครงสร้างใหม่ขึ้นมาเพื่อไว้เก็บข้อมูล ที่จำเป็นสำหรับ OLE ในวินโดวส์ 3 .1 คือใช้ไฟล์ REG และใช้โปรแกรม Registration Editor ดังนั้น จะเห็นได้ว่ารีจิสทรีได้เกิดขึ้นมาตั้งแต่วินโดวส์ 3.1 และได้ถูกพัฒนาต่อเนื่องมาจนถึงปัจจุบัน



รูป 2.14 Registry Editor

โปรแกรมทั้งหมดที่อยู่บนเครื่องคอมพิวเตอร์ ข้อมูลที่เก็บอยู่ในเครื่องจะมีตั้งแต่ค่าอุปกรณ์ฮาร์ดแวร์ไปจนถึงซอฟต์แวร์ทั้งหมด หรือแม้กระทั่งค่าเซตติ้งของผู้ใช้แต่ละคนด้วย เมื่อใดก็ตามที่

มีการเปลี่ยนอุปกรณ์ หรือติดตั้งโปรแกรมใหม่ ข้อมูลในรีจิสทรีจะถูกเรียกใช้และแก้ไขค่าต่างๆ ด้วย โดยที่รีจิสทรีจะมีรูปแบบการเก็บข้อมูล และการเรียกใช้งานที่ต่างจากการเรียกไฟล์ทั่วไป

2.12.1 ส่วนประกอบและหน้าที่ต่างๆของรีจิสทรี

รีจิสทรีในเครื่องคอมพิวเตอร์จะถูกเก็บไว้ในฮาร์ดดิสก์ ซึ่งมีชื่อเรียกว่า "รีจิสทรี ฮีฟ" (Registry Hive) รีจิสทรีไม่ได้เก็บข้อมูลทั้งหมดไว้ในไฟล์ใหญ่ๆ เพียงไฟล์เดียว แต่ในทางตรงกันข้ามจะแบ่งข้อมูลออกไปเป็นไฟล์ย่อยๆ เพื่อจะทำให้มีความยืดหยุ่นและมีความปลอดภัยในการใช้งานวินโดวส์มากยิ่งขึ้น

2.12.1.1 Registry Hive

รีจิสทรี ฮีฟ คือ ไฟล์ที่เก็บค่าเซตติ้งของวินโดวส์หรือส่วนของรีจิสทรี โดยที่รีจิสทรีฮีฟจะแบ่งไฟล์ออกเป็นไฟล์ย่อยๆ หลายๆ ไฟล์ โดยแต่ละไฟล์จะทำหน้าที่เก็บเซตติ้งของวินโดวส์ต่างๆ กันออกไป

ฮีฟไฟล์จะเก็บข้อมูลในรูปของทรี (Tree) โดยจะมีจุดเริ่มต้นอยู่ที่ รูท (RootX ภายใต้รูทจะประกอบไปด้วยส่วนหลักๆ คือส่วนเนื้อหา (Body of Keys), ซับคีย์ (SubKeys) และค่ารีจิสทรีต่างๆ ที่อยู่ในระบบ จากที่ได้กล่าวในข้างต้นจะเห็นว่าไฟล์เหล่านี้มีความสำคัญมาก ดังนั้นวินโดวส์จึงมีการสำรองข้อมูลขณะใช้งาน โดยทำการสำรองไฟล์ที่มีนามสกุลเป็นไฟล์ .log ซึ่งจะถูกเก็บไว้ในที่เดียวกับไฟลรีจิสทรีตัวอื่นๆ

2.12.1.2 Registry Root Key

แบ่งออกเป็นกลุ่มๆ ได้ 5 กลุ่ม แต่ละกลุ่มจะมีหน้าที่ในการทำงานที่ต่างๆ กัน และจะมีความสัมพันธ์ในการทำงาน รายละเอียดแต่ละกลุ่มมีดังนี้

ตาราง 2.1 คำอธิบาย Registry Root Key

โฟลเดอร์/คีย์	คำอธิบาย
HKEY_CURRENT_USER	มีรากของข้อมูลการกำหนดค่าสำหรับผู้ใช้ที่ล็อกอินอยู่ โฟลเดอร์ผู้ใช้ สีหน้าจอและการตั้งค่า Control Panel ถูกเก็บไว้ที่นี่ ข้อมูลนี้เกี่ยวเนื่องกับโปรไฟล์ผู้ใช้ คีย์นี้บางครั้งย่อว่า “HKCU”
HKEY_USERS	มีโปรไฟล์ผู้ใช้ทั้งหมดที่โหลดในคอมพิวเตอร์ HKEY_CURRENT_USER คือคีย์ย่อยของ of HKEY_USERS. HKEY_USERS บางครั้งเรียกย่อว่า “HKU”
HKEY_LOCAL_MACHINE	มีข้อมูลการกำหนดค่าเฉพาะสำหรับคอมพิวเตอร์ (สำหรับผู้ใช้ใดๆ) คีย์นี้บางครั้งย่อว่า “HKLM”
HKEY_CLASSES_ROOT	เป็นคีย์ย่อยของ HKEY_LOCAL_MACHINE\Software. ข้อมูลที่

ตาราง 2.1 คำอธิบาย Registry Root Key (ต่อ)

โฟลเดอร์/คีย์	คำอธิบาย
HKEY_CLASSES_ROOT (ต่อ)	ถูกจัดเก็บไว้ในที่นี้ช่วยให้แน่ใจว่า โปรแกรมที่ถูกต้องจะเปิดขึ้นเมื่อคุณเปิดแฟ้มโดยใช้ Windows Explorer คีย์นี้บางครั้งย่อว่า "HKCR" การเริ่มต้นด้วย Windows 2000 ข้อมูลนี้ได้รับการบันทึกไว้ทั้งในคีย์ HKEY_LOCAL_MACHINE และ HKEY_CURRENT_USER คีย์ HKEY_LOCAL_MACHINE\Software\Classes มีการตั้งค่าเริ่มต้นที่สามารถใช้กับผู้ใช้ทั้งหมดในคอมพิวเตอร์ในระบบคีย์ HKEY_CURRENT_USER\Software\Classes มีการตั้งค่าที่แทนที่การตั้งค่าที่เป็นค่าเริ่มต้น และใช้เฉพาะกับผู้ใช้ที่ไม่ได้ทำงานคีย์ HKEY_CLASSES_ROOT ให้มุมมองของรีจิสทรีที่รวมข้อมูลจากสองแหล่งนี้ นอกจากนี้ HKEY_CLASSES_ROOT ยังให้มุมมองรวมนี้สำหรับโปรแกรมที่ได้รับการออกแบบสำหรับรุ่นก่อนหน้าของ Windows การเปลี่ยนการตั้งค่าสำหรับผู้ใช้ที่ไม่ได้ใช้งาน ต้องทำการแก้ไขใน HKEY_CURRENT_USER\Software\Classes แทนที่ HKEY_CLASSES_ROOT การเปลี่ยนการตั้งค่าที่เป็นค่าเริ่มต้นต้องทำการแก้ไขใน HKEY_LOCAL_MACHINE\Software\Classes หากคุณเขียนคีย์ไปยังคีย์ใต้ HKEY_CLASSES_ROOT ระบบบันทึกข้อมูลไว้ใต้ HKEY_LOCAL_MACHINE\Software\Classes หากคุณเขียนค่าไปยังคีย์ใต้ HKEY_CLASSES_ROOT และคีย์นั้นมีอยู่แล้วใต้ HKEY_CURRENT_USER\Software\Classes ระบบจะบันทึกข้อมูลไว้ที่นี่แทนที่จะเป็นใต้ HKEY_LOCAL_MACHINE\Software\Classes
HKEY_CURRENT_CONFIG	มีข้อมูลเกี่ยวกับโปรไฟล์ฮาร์ดแวร์ที่ใช้งานโดยคอมพิวเตอร์ในระบบเมื่อระบบเริ่มทำงาน

หมายเหตุ รีจิสทรีในรุ่น 64 บิตของ Windows XP, Windows Server 2003 และ Windows Vista แบ่งออกเป็นคีย์แบบ 32 บิตและ 64 บิต คีย์ 32 บิตจำนวนมากมีชื่อเหมือนกับแบบ

64 บิต และอื่นๆ ค่าเริ่มต้นของรุ่น 64 บิตของ Registry Editor ที่รวมรุ่น 64 บิตของ Windows XP, Windows Server 2003 และ Windows Vista จะแสดงคีย์แบบ 32 บิตภายใต้โหนดต่อไปนี้ HKEY_LOCAL_MACHINE\Software\WOW6432Node

2.12.1.3 Registry Data

Data Type จะเป็นชนิดของข้อมูลที่สามารถใช้งานได้บนรีจิสทรีซึ่งทุกชนิดจะต้องมีค่าอย่างน้อย 1 ค่าเสมอ ไม่สามารถมีค่าว่างได้ และค่าที่ปรากฏอยู่นั้นบางครั้งก็ไม่ใช่ค่าจริงเสมอไป โดยวินโดวส์แบ่งชนิดของข้อมูลรีจิสทรีออกได้เป็นหมวดหมู่ต่างๆ ดังนี้

ตารางต่อไปนี้แสดงประเภทข้อมูลที่ระบุอยู่และใช้งาน Windows ขนาดสูงสุดของค่าชื่อเป็นดังนี้

1) Windows Server 2003, Windows XP และ Windows Vista 16,383 อักขระ
2) Windows 2000: อักขระ ANSI 260 อักขระหรือ อักขระ Unicode 16,383 อักขระ
3) Windows Millennium Edition/Windows 98/Windows 95: 255 อักขระ

ค่าแบบยาว (มากกว่า 2,048 ไบต์) ต้องถูกจัดเก็บเป็นแฟ้มที่มีชื่อแฟ้มซึ่งเก็บไว้ในรีจิสทรี ซึ่งช่วยให้รีจิสทรีทำงานได้อย่างมีประสิทธิภาพ ขนาดสูงสุดของค่าเป็นดังนี้ Windows NT 4.0/Windows 2000/WindowsXP/Windows Server 2003/Windows Vista: หน่วยความจำที่ใช้งานได้ Windows Millennium Edition/Windows 98/Windows 95: 16,300 ไบต์

หมายเหตุ มีขีดจำกัด 64K สำหรับขนาดทั้งหมดของค่าคีย์

ตาราง 2.2 คำอธิบาย Registry Data

ชื่อ	ประเภทข้อมูล	คำอธิบาย
ค่าไบนารี	REG_BINARY	ข้อมูลไบนารีแบบดิบ ข้อมูลคอมโพเนนต์ฮาร์ดแวร์ส่วนมากจะได้รับการบันทึกเป็นข้อมูลไบนารีและแสดงใน Registry Editor เป็นรูปแบบฐานสิบหก
ค่า DWORD	REG_DWORD	ข้อมูลแสดงโดยตัวเลขความยาว 4 ไบต์ (ตัวเลข 32 บิต) พารามิเตอร์หลายชนิดสำหรับโปรแกรมควบคุมอุปกรณ์และบริการต่างๆ เป็นประเภทนี้และแสดงใน Registry Editor เป็นไบนารี ฐานสิบหกหรือฐานสิบ ค่าที่เกี่ยวข้องได้แก่ DWORD_LITTLE_ENDIAN (ไบต์ที่สำคัญน้อยที่สุดอยู่ที่ที่อยู่ต่ำสุด) และ REG_DWORD_BIG_ENDIAN (ไบต์ที่

ตาราง 2.2 คำอธิบาย Registry Data (ต่อ)

ชื่อ	ประเภทข้อมูล	คำอธิบาย
ค่า DWORD (ต่อ)	REG_DWORD (ต่อ)	สำคัญน้อยที่สุดอยู่ที่ที่อยู่สูงสุด)
ค่าสายอักขระที่ขยายได้	REG_EXPAND_SZ	สายอักขระข้อมูลที่มีความยาวแบบผันแปรได้ ประเภทข้อมูลนี้รวมถึงตัวแปรที่ได้รับการแก้ไขเมื่อโปรแกรมหรือบริการใช้ข้อมูล
ค่าสายอักขระจำนวนมาก	REG_MULTI_SZ	สายอักขระหลายสาย ค่าต่างๆ ที่มีรายชื่อหรือค่าหลายค่าในรูปแบบที่คนสามารถอ่านได้มักจะเป็นประเภทนี้ รายการต่างๆ ถูกแยกโดยช่องว่าง จุลภาคหรือเครื่องหมายอื่นๆ
ค่าสายอักขระ	REG_SZ	สายอักขระข้อความที่มีความยาวตายตัว
ค่าไบนารี	REG_RESOURCE_LIST	ชุดข้อมูลของอาเรย์ที่ซ้อนกันอยู่ที่ได้รับการออกแบบ เพื่อจัดเก็บรายชื่อทรัพยากรที่ใช้งานโดยโปรแกรมควบคุมอุปกรณ์ฮาร์ดแวร์ หรืออุปกรณ์ทางกายภาพที่ควบคุม ข้อมูลนี้ได้รับการตรวจสอบและเขียนในทรี \ResourceMap โดยระบบและแสดงใน Registry Editor ในรูปแบบฐานสิบหกเป็นค่าไบนารี
ค่าไบนารี	REG_RESOURCE_REQUIREMENTS_LIST	ชุดข้อมูลของอาร์เรย์ที่ซ้อนกันที่ได้รับการออกแบบเพื่อบันทึกรายชื่อโปรแกรมควบคุมอุปกรณ์ของทรัพยากรฮาร์ดแวร์ หรือหนึ่งในอุปกรณ์ทางกายภาพที่การควบคุมสามารถใช้ได้ ระบบเขียนเซตย่อยของรายชื่อนี้ในทรี \ResourceMap ข้อมูลนี้ได้รับการตรวจพบโดยระบบและแสดงใน Registry Editor ในรูปแบบฐานสิบหกเป็นค่าไบนารี
ค่าไบนารี	REG_FULL_RESOURCE_DESCRIPTOR	ชุดข้อมูลของอาร์เรย์ที่ซ้อนกันที่ได้รับการออกแบบเพื่อบันทึกรายชื่อทรัพยากรที่ใช้โดยอุปกรณ์ฮาร์ดแวร์ ข้อมูลนี้ได้รับการตรวจพบและเขียนในทรี \HardwareDescription โดยระบบและแสดงใน Registry Editor ในรูปแบบฐานสิบหกเป็นค่าไบนารี
ไม่มี	REG_NONE	ข้อมูลที่ไม่มีประเภทเฉพาะ ข้อมูลนี้ถูกเขียนไปยังรีจิสทรีโดยระบบหรือโปรแกรมประยุกต์และแสดงใน Registry

ตาราง 2.2 คำอธิบาย Registry Data (ต่อ)

ชื่อ	ประเภทข้อมูล	คำอธิบาย
ไม่มี(ต่อ)	REG_NONE (ต่อ)	Editor ในรูปแบบฐานสิบหกเป็นค่าไบนารี
การเชื่อมโยง	REG_LINK	สายอักขระแบบ Unicode ที่ตั้งชื่อการเชื่อมโยงสัญลักษณ์
ค่า QWORD	REG_QWORD	ข้อมูลที่แสดงโดยตัวเลขที่เป็นจำนวนเต็ม 64 บิต ข้อมูลนี้แสดงใน Registry Editor เป็นค่าไบนารีและมีการใช้งานครั้งแรกใน Windows 2000

2.13 Process

โดยความหมายในทางปฏิบัติแล้ว โพรเซส หมายถึง โปรแกรมที่กำลังถูกประมวลผล ในการทำงานทั่วไปในระบบคอมพิวเตอร์นั้นผู้ใช้อาจต้องการเรียกใช้โปรแกรมต่างๆ ซึ่งโปรแกรมเหล่านี้จะถูกเปลี่ยนให้เป็นโพรเซส ผ่านกระบวนการที่ได้กำหนดไว้ ช่วงชีวิตของโปรแกรมที่กำลังถูกประมวลผลนี้มีอยู่หลายสถานะและตัวของโพรเซสเองก็ต้องมีที่เก็บข้อมูลที่เกี่ยวข้องกับตัวเองซึ่งเราเรียกส่วนนี้ว่า Process Control Block (PCB)

ตาราง 2.3 รายละเอียด Process Control Block

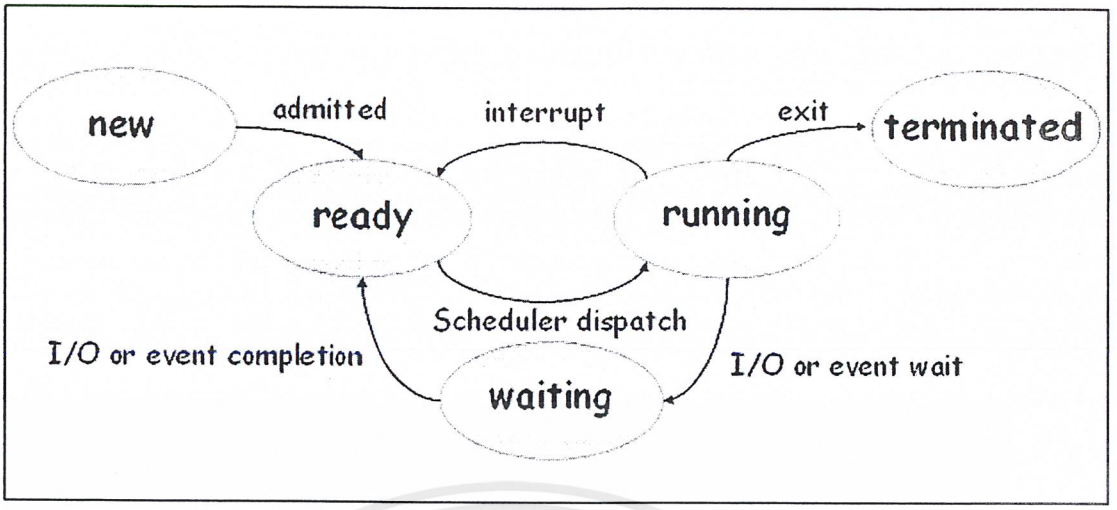
Identifier	หมายเลขประจำตัวของ process
State	สถานะของ process ขณะประมวลผล
Priority	สิทธิของ process เมื่อเปรียบเทียบกับ process อื่น
Program counter	ตำแหน่งของการประมวลผลคำสั่งถัดไป
Memory pointers	ตัวชี้หน่วยความจำที่ process ใช้อยู่
Context data	ข้อมูลที่อยู่ใน register ของ process ขณะถูกประมวลผล
I/O status information	ข้อมูลของ I/O ที่ process เกี่ยวข้องAccounting information ข้อมูลเกี่ยวกับเวลาของ CPU ที่process ใช้, เวลาที่ใช้ไป, ช่วงเวลาที่สามารถใช้ได้ และอื่นๆ ที่เกี่ยวข้อง

2.13.1 สถานะของกระบวนการ (Process State)

ระบบคอมพิวเตอร์แบบหลายโปรแกรม (Multiprogramming) และแบบผู้ใช้หลายคน (Multiuser) จะมีกระบวนการที่ทำงาน อยู่ในระบบหลายกระบวนการพร้อมๆกัน โดยที่บางกระบวนการกำลังขอเข้าใช้งานหน่วยประมวลผลกลาง (CPU) บางกระบวนการกำลังใช้งานหน่วย

ประมวลผลกลางอยู่ บางกระบวนการกำลังร้องขออุปกรณ์รับ-ส่งข้อมูลอยู่ พฤติกรรมของกระบวนการเหล่านี้ เรียกอีกอย่างว่า "สถานะกระบวนการ" (State of Process) กระบวนการ (Process) หมายถึง คำสั่งในโปรแกรมที่ถูกประมวลผลด้วยหน่วยประมวลผลกลางหรืออีกในหนึ่ง ณ เวลาใดๆจะมีเพียงอย่างมาหนึ่งคำสั่งที่ดำเนินการอยู่ สถานะของกระบวนการ (Process state) กระบวนการต่าง ๆ ที่กำลังทำงานอยู่ในระบบเดียวกันจะมีการเปลี่ยนแปลงสถานะของกระบวนการถึง 5 สถานะด้วยกัน ซึ่งสถานะดังกล่าวจะถูกกำหนดขึ้นโดยกิจกรรม ณ เวลาปัจจุบันที่กระบวนการนั้น ๆ กำลังกระทำอยู่ โดยที่แต่ละกระบวนการจะตกอยู่ในสถานะใดสถานะหนึ่งจากสถานะทั้ง 5 ต่อไปนี้

1) New เป็นสถานะของกระบวนการใหม่ที่กำลังถูกสร้างขึ้นหรือกระบวนการเลือกมาจาก หน่วยความจำสำรอง (Disk) ซึ่งเป็นคำสั่งที่ผู้ใช้เรียกใช้ผ่าน Command Interpreter แปลเป็นคำสั่งไปเรียกระบบปฏิบัติการให้ดึงข้อมูลหรือโปรแกรมมาตามคำสั่งของผู้ใช้เพื่อเข้ามาประมวลผลในระบบ เมื่อคำสั่งต่าง ๆ ถูกเรียกเข้ามา คำสั่งเหล่านั้นจะมาเข้าแถวรอในแถวงาน (Job Queue) เตรียมเปลี่ยนสถานะเพื่อทำงาน
2) Ready เป็นสถานะของกระบวนการที่เตรียมตัวเข้าไปใช้งานหน่วยประมวลผลกลาง ในสถานะนี้จะเปลี่ยนมาจาก New หรือ Waiting หรือ Running ก็ได้ กระบวนการที่มาจาก New, Waiting หรือ Running จะเข้าแถวคอยเพื่อเข้าไปใช้หน่วยประมวลผลกลางแถวคอยนี้เราเรียกว่า (Ready Queue)
3) Running เป็นสถานะของกระบวนการที่ได้เข้าไปใช้งานหน่วยประมวลผลกลาง ณ เวลาใดเวลาหนึ่ง จะมีเพียง 1 กระบวนการเท่านั้นที่อยู่ในสถานะนี้ของระบบ 1 ระบบ (มีเพียงกระบวนการเดียวเท่านั้นที่จะได้ใช้หน่วยประมวลผลกลางของแต่ละระบบ) เนื่องจากข้อจำกัดของประมวลผลกลางทำงานด้วยความเร็วสูงมาก จึงไม่มีปัญหาในเรื่องการรอ
4) Terminate เป็นสถานะของกระบวนการที่ได้รับการประมวลผลเสร็จเรียบร้อยแล้ว หรือกระบวนการ มีการทำงานที่ผิดปกติ เช่น มีการหารด้วยศูนย์ระบบจะหยุดการทำงานของกระบวนการนั้น แล้วแจ้งให้ทราบถึงข้อผิดพลาดที่เกิดขึ้น (Error)
5) Waiting เป็นสถานะของกระบวนการที่ได้เข้าไปใช้หน่วยประมวลผลกลางแล้วและมีการเรียกใช้อุปกรณ์รับ-ส่งข้อมูลหรืออุปกรณ์ต่าง ๆ ซึ่งทรัพยากรเหล่านั้นยังไม่ว่างหรือมีกระบวนการอื่นใช้อยู่ (เนื่องจาก CPU ทำงานเร็วกว่าอุปกรณ์รับส่งข้อมูลมาก) กระบวนการเล่านั้นจะเปลี่ยนจาก Running มารอในสถานะนี้อาจมีกระบวนการหลายกระบวนการรออยู่ จึงมีการจัดคิวในการรอทรัพยากรต่างๆ เรียกว่า Device Queue หรือ Waiting Queue



รูป 2.15 สถานะของโพรเซส

2.13.2 การติดต่อสื่อสารระหว่างกระบวนการ (Interprocess Communication)

กระบวนการที่ดำเนินการควบคู่กันในระบบปฏิบัติการอาจจะเป็นกระบวนการอิสระหรือทำงานร่วมกันกับกระบวนการอื่น กระบวนการจะอิสระถ้าไม่กระทบหรือได้รับผลกระทบต่อกระบวนการอื่น ที่กำลังดำเนินการอยู่ในระบบปฏิบัติการ กระบวนการใดๆที่ไม่มีการใช้ข้อมูลร่วมกับกระบวนการอื่นคือเป็นอิสระ กระบวนการกำลังร่วมมือกันถ้ากระทบหรือถูกกระทบโดยกระบวนการอื่น ที่กำลังดำเนินการอยู่บนระบบ อย่างชัดเจนคือ กระบวนการใด ๆ ที่ใช้ข้อมูลร่วมกับกระบวนการอื่น กระบวนการกำลังร่วมมือกันมีหลายสาเหตุที่ต้องจัดหาสภาพแวดล้อมเพื่ออนุญาตให้กระบวนการทำงานร่วมกัน

1) ใช้ข่าวสารร่วมกัน มีผู้ใช้หลายคนมีความสนใจในข่าวสารชิ้นเดียวกัน เราจำเป็นต้องจัดหาสภาพแวดล้อมเพื่ออนุญาตให้เข้าถึงข่าวสารเดียวควบคู่กันได้
2) การประมวลผลเร็วขึ้น ถ้าเราต้องการงานเฉพาะที่ทำงานได้เร็วกว่าเดิม เราจะต้องพักไว้ในงานย่อย แล้วแต่ละอันจะดำเนินการขนานกันไป สังเกตได้ว่าความเร็วที่เพิ่มขึ้นจะสำเร็จได้นั้นถ้าคอมพิวเตอร์มีการประมวลผลแบบหลายกระบวนการ
3) ประกอบขึ้นจากหน่วยย่อยๆ เราอาจจะต้องการสร้างระบบในรูปแบบที่แยกย่อยได้ แบ่งหน้าที่ของระบบกลายเป็นกระบวนการย่อยหรือเธรด ความสะดวกผู้ใช้แต่ละคนอาจจะทำงานหลายงานในเวลาเดียวกัน เช่น กำลังแก้ไข กำลังพิมพ์ และกำลังคอมไพล์ขนานกันอยู่

การทำงานร่วมกันของกระบวนการต้องการกลไกการติดต่อสื่อสารระหว่างกระบวนการที่จะสามารถอนุญาตให้แลกเปลี่ยนข้อมูลและข่าวสาร มีอยู่ 2 รูปแบบพื้นฐานของการติดต่อสื่อสารระหว่างกระบวนการ คือ การใช้หน่วยความจำร่วมกัน (shared memory) และการส่งผ่านข้อความ (message passing) ในรูปแบบการใช้หน่วยความจำร่วมกันพื้นที่ของหน่วยความจำ

จะถูกแบ่งปันกันโดยกระบวนการที่ทำงานร่วมกัน กระบวนการสามารถที่จะแลกเปลี่ยนข่าวสารโดยการอ่านและการเขียนข้อมูลบนพื้นที่ที่ใช้ร่วมกัน ในรูปแบบการส่งผ่านข้อความ การติดต่อสื่อสารเกิดขึ้นเพราะการร่วมมือกันระหว่างกระบวนการ

การส่งผ่านข้อความมีประโยชน์สำหรับการแลกเปลี่ยนข้อมูลที่มีจำนวนน้อยกว่า เพราะว่าไม่มีความขัดแย้งที่จำเป็นต้องหลีกเลี่ยง การส่งผ่านข้อความนั้นง่ายที่จะนำไปทำให้เกิดผลกว่าการใช้หน่วยความจำร่วมกันสำหรับการติดต่อสื่อสารระหว่างคอมพิวเตอร์การใช้หน่วยความจำร่วมกัน อนุญาตให้ใช้ความเร็วสูงสุดและสะดวกในการสื่อสาร ดังนั้นสามารถทำเสร็จที่ความเร็วหน่วยความจำภายในเครื่องคอมพิวเตอร์ การใช้หน่วยความจำร่วมกันเร็วกว่าการส่งผ่านข้อความในระบบการส่งผ่านข้อความถูกทำให้เป็นผลโดยใช้ System Calls และดังนั้นต้องการเวลาอย่างมากในการทำงานในส่วนที่ต่างกัน ในระบบที่ใช้หน่วยความจำร่วมกัน System Calls ต้องการเพียงแค่บริเวณที่ใช้หน่วยความจำร่วมกันเท่านั้น หน่วยความจำที่ใช้งานร่วมกันถูกสร้างขึ้นครั้งหนึ่ง ทุกการเข้าถึงถูกเก็บรักษาที่การเข้าถึงหน่วยความจำ Routine และไม่มีการช่วยเหลือจาก เคอร์เนลที่ถูกเรียกร้อง

2.13.3 ระบบใช้หน่วยความจำร่วมกัน (Shared-Memory Systems)

การติดต่อสื่อสารระหว่างกระบวนการที่ใช้หน่วยความจำร่วมกันต้องการกระบวนการที่สื่อสารกันที่สร้างพื้นที่ของการแบ่งปันหน่วยความจำร่วมกัน บริเวณหน่วยความจำที่ใช้ร่วมกันมีอยู่ในที่อยู่ที่ว่างของกระบวนการที่สร้างหน่วยความจำร่วมกันที่แบ่งแยกออกมา กระบวนการใดที่หวังจะใช้หน่วยความจำร่วมกันจะต้องเพิ่มในที่อยู่ด้วย โดยปกติแล้วระบบปฏิบัติการจะป้องกันกระบวนการหนึ่งจากการเข้าถึงหน่วยความจำของกระบวนการอื่น การใช้หน่วยความจำร่วมกันต้องการสองหรือมากกว่าสองกระบวนการที่ยินยอมกำจัดข้อกำหนดนี้ สามารถแลกเปลี่ยนข่าวสารโดยการอ่านและการเขียนข้อมูลในพื้นที่ที่ใช้ร่วมกัน รูปแบบของข้อมูลและตำแหน่งถูกกำหนดโดยกระบวนการเหล่านี้และไม่ได้อยู่ภายใต้การควบคุมของระบบปฏิบัติการ กระบวนการนั้นๆ จะแน่ใจได้ว่าไม่ได้เขียนที่ตำแหน่งเดียวกัน

มีบัฟเฟอร์ 2 ชนิดที่ใช้ คือ บัฟเฟอร์ไม่จำกัด (unbounded buffer) ตั้งอยู่ในบัฟเฟอร์ที่ไม่ได้จำกัดขนาด ผู้ใช้อาจจะคอยสิ่งของใหม่ๆ แต่ผู้ผลิตสามารถสร้างสิ่งใหม่ๆ อยู่เสมอ บัฟเฟอร์จำกัด (bounded buffer) ก็คือ บัฟเฟอร์ที่มีการจำกัดขนาดคงที่ กรณีนี้ผู้ใช้จะต้องรอให้บัฟเฟอร์ว่างก่อนและผู้ผลิตก็ต้องรอถ้าบัฟเฟอร์เต็ม

2.13.4 ระบบส่งผ่านข้อความ (Message-Passing Systems)

การส่งผ่านข้อความจะจัดหากลไกที่อนุญาตให้กระบวนการติดต่อสื่อสารกันและทำงานพร้อมกันโดยไม่ใช้ที่อยู่เดียวกันและโดยเฉพาะอย่างยิ่งมีประโยชน์ในการกระจายสภาพแวดล้อม ที่ซึ่งกระบวนการติดต่อสื่อสารกันอาจจะอยู่บนคอมพิวเตอร์คนละเครื่องกันที่

เชื่อมต่อโดยเครือข่าย ตัวอย่างเช่น โปรแกรมแชทที่ใช้บนเว็บถูกออกแบบสำหรับให้ผู้คนได้มีส่วนร่วมในการติดต่อสื่อสารซึ่งกันและกันโดยแลกเปลี่ยนข้อความกัน

คุณสมบัติของการส่งผ่านข้อความนั้นได้เตรียมไว้อย่างน้อย 2 อย่างคือ ส่งข้อความและรับข้อความ ข้อความถูกส่งโดยกระบวนการหนึ่งซึ่งสามารถกำหนดขนาดคงที่หรือเปลี่ยนแปลงขนาดได้ ถ้ากำหนดขนาดข้อความคงที่จะสามารถส่งได้ การทำให้เกิดผลจะง่าย ๆ ตรงไปตรงมา แต่ทำให้งานเขียนโปรแกรมยากกว่า ในทางกลับกันข้อความที่เปลี่ยนแปลงขนาดได้ต้องการความซับซ้อนในการทำให้เกิดผลมากกว่า แต่กลับกลายเป็นว่างานเขียนโปรแกรมนั้นง่าย นี่คือชนิดของการแลกเปลี่ยนที่เกิดขึ้นบ่อยๆ โดยตลอดการออกแบบระบบปฏิบัติการ

ถ้ากระบวนการ P และ Q ต้องการจะสื่อสารกันมันจะต้องส่งและรับข้อความจากซึ่งกันและกัน การเชื่อมต่อการสื่อสารจะต้องคงอยู่ระหว่างกระบวนการทั้งสอง การเชื่อมต่อนี้สามารถที่จะทำได้หลากหลายวิธี เรามีความสัมพันธ์กับมันไม่ใช่ด้วยการเชื่อมต่อแบบกายภาพแต่เป็นแบบตรรกะ มีหลากหลายวิธีการสำหรับการเชื่อมต่อแบบตรรกะและการดำเนินการ send() / receive()

1) การสื่อสารแบบทางตรงและแบบทางอ้อม (Direct or indirect communication)
2) การสื่อสารแบบพร้อมกันหรือแบบไม่พร้อมกัน (Synchronous or asynchronous communication)
3) อัตโนมัติหรือบัฟเฟอร์อย่างชัดเจน (Automatic or explicit buffering)

2.13.4.1 Naming

กระบวนการที่ต้องการในการติดต่อสื่อสารจะต้องมีทางที่สามารถทำให้สื่อสารถึงกันและกันได้ สิ่งเหล่านั้นสามารถใช้ติดต่อสื่อสารได้ทั้งทางตรงและทางอ้อม ในการติดต่อสื่อสารทางตรงทุกกระบวนการต้องการที่จะสื่อสารจะต้องมีชื่อของผู้รับและผู้ส่ง ที่มีความชัดเจน เช่น

1) การส่ง (P, message), ส่งข้อความถึงการประมวลผล P
2) การรับ (Q, message) รับข้อความจากการประมวลผล Q

การเชื่อมต่อในการติดตอสื่อสารแบบนี้จะต้องมีคุณสมบัติดังต่อไปนี้

1) การเชื่อมต่อ คือ การสร้างระบบฐานมั่นคงแบบอัตโนมัติระหว่างทุกๆ ส่วนของการประมวลผลที่ต้องการติดต่อสื่อสาร ในการประมวลผลต้องการรู้เพียงข้อมูลของแต่ละบุคคลที่จะติดต่อสื่อสาร
2) การเชื่อมต่อเป็นการเกี่ยวเนื่องกับความเป็นจริงสองการประมวลผลระหว่างแต่ละส่วนของการประมวลผลยังคงมีอยู่อย่างแน่นอนในหนึ่งการเชื่อมต่อ

แบบแผนอันนี้ซึ่งแสดงออกมาอย่างมีสัดสวนในที่อยู่ของผู้ส่งและผู้รับ จะต้องมีชื่อในที่อื่น ของการติดต่อสื่อสารด้วย ความหลากหลายของแบบแผนงานอย่างมีสัดส่วนในที่อยู่ ที่มี

ชื่อผู้ส่งเพียงเท่านั้นและการรับไม่จำเป็นว่าจะต้องมีชื่อผู้ส่งในแบบแผนนี้ การส่งและการรับ มีความชัดเจนดังต่อไปนี้

1) การส่ง (P, message), ส่งข้อความถึงการประมวลผล P
2) การรับ (id, message), รับข้อความจากทุก ๆ การประมวลผล ความแตกต่างของข้อมูลเฉพาะบุคคลเป็นการจัดระบบชื่อของการประมวลผลกับการติดต่อสื่อสารมีการพูดถึงสถานที่ด้วย

ข้อเสียเปรียบของแบบแผนทั้งสองอย่างนี้ (การมีสัดส่วนรองรับและการไม่มีระบบสัดส่วนรองรับ) เป็นเกณฑ์จำกัดของระบบการประมวลผลที่มีผลลัพธ์แบบกำหนด การเปลี่ยน แปลงของข้อมูลส่วนตัวของการประมวลผลบางทีบางทีก็จำเป็นต่อการสอบของข้อกำหนดการประมวลผลอื่น ๆ ทั้งหมด ข้อมูลอ้างอิงทั้งหมดถึงข้อมูลส่วนตัวเดิมต้องถูกพบ ดังนั้นจึงเกิดการพัฒนาเป็นข้อมูลส่วนตัวอันใหม่ ในทางทั่วไป อย่างเช่น เทคนิคของ code ที่ยาก ๆ ซึ่งข้อมูลส่วนตัวนั้นจะต้องมีกระบวนการที่ชัดแจ้งเป็นความต้องการที่น้อยลงมากกว่าเทคนิคก่อให้เกิดการติดต่อสื่อสารทางอ้อมตามมา

การติดต่อสื่อสารทางอ้อมข้อความจะส่งถึงและรับจากเมล์บ๊อกซ์หรือช่องทางเมล์บ๊อกซ์ สามมารถที่จะแสดงการสรุป เช่น หัวเรื่องถึงข้อความสามารถเป็นสถานที่ ๆ ประมวลผลและจากข้อความสามารถลบออกจากแต่ละเมล์บ๊อกซ์ ความพิเศษของแต่ละข้อมูลส่วนตัว ตัวอย่างเช่น ข้อมูล POSIX เป็นแถวใช้จำนวนเต็มของข้อมูลส่วนตัวในเมล์บ๊อกซ์ ในแบบแผนนี้ การประมวล ผลสามารถติดต่อสื่อสารกับการประมวลผลอย่างอื่นผ่านตัวเลขที่แตกต่างกันคนละเมล์บ๊อกซ์ได้ อย่างไรก็ตามการส่งและการรับอย่างมีสัดส่วนรองรับต้องเป็นแบบดังต่อไปนี้

1) การส่งข้อความ A, ถึงเมล์บ๊อกซ์ A
2) การรับข้อมูล A, รับข้อมูลจากเมล์บ๊อกซ์ A

ในแบบแผนนี้การเชื่อมการสื่อสารต้องมีคุณสมบัติ ดังนี้

1) การเชื่อมต่อแบบมั่นคงระหว่างแต่ละคู่ของการประมวลผลเท่านั้น ถ้าคู่จำนวนของคู่ต้องการที่จะแบ่งเมล์บ๊อกซ์กัน
2) การเชื่อมต่ออาจจะต้องมีความสัมพันธ์มากกว่าสองการประมวลผล
3) ระหว่างคู่แต่ละคู่ของการประมวลผลการติดต่อสื่อสาร บางทีอาจจะมีตัวเลขของความแตกต่างของการเชื่อมต่อกับแต่ละการเชื่อมต่อในการตอบรับของหนึ่งเมล์บ๊อกซ์

ขณะนี้การสนับสนุนการประมวลผลนั้น P1, P2, และ P3 ทั้งหมดจะต้องแบ่งเมล์บ๊อกซ์ A การประมวลผล P1 ส่งข้อความถึง A, ระหว่างคู่ของ P2 และ P3 การจัดการรับจาก A ซึ่งการประมวลผลจะรับข้อความส่งโดย P1 การตอบตัดสินโดย การตอบสนองของวิธีที่เราเลือก

1) อนุญาตการเชื่อมอย่างเกี่ยวความเกี่ยวพันกับสองการประมวลผลอย่างมากที่สุด
2) อนุญาตอย่างมากที่สุดของการประมวลผลในเวลาจัดการกระบวนการรับ
3) อนุญาตระบบในการเลือกโดยไม่ใช้เกณฑ์ซึ่งการประมวลผลจะรับข้อความ (มีทั้ง P2 และ P3 แต่ไม่ใช่ทั้งสองที่จะรับข้อมูลได้) ระบบมักจะมีตัวเลขเป็นสัดส่วนในการเลือก ซึ่งการประมวลผลจะรับข้อความ (มี Round Robin ซึ่งประมวลผลจะได้ข้อความที่ได้รับกลับคืนมา) บางทีระบบจะได้รับข้อมูลส่วนตัวของผู้รับถึงผู้ส่ง

การเป็นเจ้าของของเมล์บ็อกซ์อาจจะเป็นด้วยกันทั้งคู่โดยการประมวลผล หรือโดยระบบกระบวนการ ถ้าเมล์บ็อกซ์ถูกเป็นเจ้าของด้วยการประมวลผล (นั่นคือ เมล์บ็อกซ์เป็นส่วนหนึ่งของที่อยู่ของช่องว่างของการประมวลผล) หลังจากนั้นพวกที่ผิดแปลกระหว่างการเป็นเจ้าของ (ผู้ซึ่งสามารถรับข้อความได้จากเมล์บ็อกซ์นี้เท่านั้น) และผู้ใช้ (ผู้ซึ่งสามารถส่งข้อความสู่เมล์บ็อกซ์ได้เท่านั้น) แต่ละเมล์บ็อกซ์มีเจ้าของทั้งหมด สามารถเป็นสิ่งที่ไม่เอาไปปะปนกันเกี่ยวกับผู้รับข้อความที่ถูกส่งในเมล์บ็อกซ์นี้ เมื่อมีการประมวลผลนั้นเจ้าของเมล์บ็อกซ์สุดท้ายเมล์บ็อกซ์ที่สูญหาย การประมวลผลบางทีนั้นต่อมาในการติดต่อกัน เมล์บ็อกซ์นั้นเป็นเจ้าของโดยระบบกระบวนการมีการคงอยู่ของเจ้าของของมันในเวลานี้ และไม่มีการแนบไฟล์ไปสู่ส่วนต่าง ๆ ของการประมวลผล ระบบกระบวนการจะต้องมีการจัดการกับเครื่องกลนั้นจะอนุญาตให้การประมวลผลได้ทำตาม ดังนี้

1) สร้างเมล์บ็อกซ์ใหม่
2) ส่งและรับข้อความได้ในเมล์บ็อกซ์
3) ลบเมล์บ็อกซ์

การประมวลผลนั้นการสร้างเมล์บ็อกซ์ใหม่นั้น เมล์บ็อกซ์ของผู้เป็นเจ้าของโดยการละเลยในขั้นตอนแรก เจ้าของนั้นเป็นผู้ประมวลผลสามารถรับข้อความได้ในเมล์บ็อกซ์ อย่างไรก็ตามส่วนของเจ้าของ และส่วนของผู้รับมีสิทธิพิเศษบางทีอาจจะผ่านไปสู่การประมวลผลอื่นสมควรจะถูกได้เรียกขึ้นมาใช้ แน่นอนการเตรียมสามารถหาผลลัพธ์ได้ในผู้รับหลาย ๆ ฝ่ายของแต่ละเมล์บ็อกซ์

2.13.4.2 Syschronization

การติดต่อสื่อสารระหว่างการประมวลผลสถานที่ที่เรียกว่าผู้ส่งและผู้รับแบบมีสัดส่วนรองรับ มีความแตกต่างในการออกแบบการจัดการสำหรับการสนับสนุนแต่ละสัดส่วนข้อความที่ผ่านบางทีถูกขัดขวาง หรือไม่ขัดขวางมักจะทราบได้โดยระบบที่มีของการจัดการข้อมูลอย่างมีสัดส่วนรองรับและไม่มีสัดส่วนรองรับ

1) การขัดขวางการส่ง การส่งการประมวลผลเป็นการขัดขวางจนกระทั่งข้อความนั้นได้ถูกรับ โดยการประมวลผลรองรับหรือโดยเมล์บ็อกซ์
2) ไม่ขัดขวางการส่ง การประมวลผลการส่งเป็นข้อความและผลลัพธ์ของกระบวนการ
3) การขัดขวางการรับ ผู้รัยจะถูกขัดขวางจนกระทั่งข้อความเท่าที่จะทำได้
4) การไม่ขัดขวางการรับ ผู้รับจะได้รับข้อความที่ถูกต้องด้วยหรือข้อความที่ไม่ได้ใช้ด้วย

ความแตกต่างของการรวมกันของการส่งและการรับเป็นสิ่งที่เป็นไปไม่ได้ เมื่อทั้งคู่ของการส่งและการรับและการขัดขวาง มีที่นัดพบกันระหว่างผู้ส่งและผู้รับ ทางออกของผู้ผลิตและลูกค้าที่มีปัญหาที่กลายมาเป็นสิ่งที่น่ารำคาญ เมื่อพวกเราถูกขัดขวางการส่งและการรับอย่างไม่เป็นระบบ ผู้ผลิตเท่านั้นที่เรียกการอ้างถึงการถูกขัดขวางการส่ง และรอจนกระทั่งข้อความได้ถูกส่งไปสู่ผู้รับหรือเมล์บ็อกซ์ เช่นเดียวกัน เมื่อลูกค้าเท่านั้นได้รับ มันจะถูกขัดขวางจนกระทั่งข้อความสามารถค้นหาได้พบ

2.13.4.3 Buffering

การติดต่อสื่อสารทั้งทางตรงและทางอ้อม ข้อความจะเปลี่ยนโดยกระบวนการการติดต่อสื่อสารแบบเป็นแถวยาวอยู่แบบชั่วคราว โดยพื้นฐานแถวยาวแต่ละแถวสามารถเป็นการสนับสนุนด้วยกันสามทางคือ

1) Zero Capacity แถวยาวมีความยาวของศูนย์ ด้วยเหตุนี้การเชื่อมต่อโดยไม่สามารถมีข้อความรออยู่ได้ในนั้น ในกรณีนี้ ผู้ส่งต้องถูกขัดขวางจนกระทั่งจะได้รับข้อความ
2) Bounded Capacity แถวยาวจะมีแถวยาวที่มีความยาวแน่นอนมีจำนวนจำกัด ด้วยเหตุนี้ ทั้งหมดของข้อมูลสามารถอยู่ในที่ของมันได้ถ้าแถวยาวไม่เต็ม เมื่อข้อความใหม่ถูกส่ง ข้อความที่ถูกส่งไปยังสถานที่ในแถวยาว (ทั้งคู่ของข้อความจะถูกคัดลอก โดยปราศจากการรอคอยการเชื่อมต่อของ Capacity เป็นความยาวที่แน่นอนมีจำกัด อย่างไรก็ตามถ้าการเชื่อมต่อนั้นเต็ม ผู้ส่งจะต้องขัดขวางจนกระทั่งที่ว่างสามารถถูกหาได้ได้พบในแถวยาว)
3) Unbounded Capacity ความยาวของแถวสามารถจะเป็นไปได้อย่างไม่สิ้นสุด ด้วยเหตุนี้ จำนวนต่อของข้อความสามารถรอในตัวของมันได้ ผู้ส่งก็ไม่เคยที่ถูกขัดขวาง

The Zero Capacity ประเด็นนี้บางครั้งอ้างถึงระบบข้อความระบบข้อความกับไม่ผสมหรือปะปนส่วนประเด็นอื่น การอ้างถึงระบบจะต้องมีการผสมหรือปะปนกับระบบอัตโนมัติด้วย

2.14 Port

สำหรับพวก Application ในชั้น layer สูง ๆ ที่ใช้ TCP (Transmission Control Protocol) หรือ UDP (User Datagram Protocol) จะมีหมายเลขพอร์ต หมายเลขของพอร์ตจะเป็นเลข 16 bit เริ่มตั้งแต่ 0 ถึง 65535 หมายเลขพอร์ตใช้สำหรับตัดสินว่า Service ใดที่ต้องการเรียกใช้ ในทางทฤษฎีหมายเลขพอร์ตแต่ละหมายเลขถูกเลือกสำหรับ Service ใดๆ ขึ้นอยู่กับ OS (Operating System) ที่ใช้ไม่จำเป็นต้องเหมือนกัน แต่ได้มีกำหนดขึ้นให้ใช้ค่อนข้างเป็นมาตรฐานเพื่อให้มีการติดต่อการส่งข้อมูลที่ดีขึ้น ทาง Internet Assigned Numbers Authority (IANA) เป็นหน่วยงานกลางในการประสานการเลือกใช้พอร์ตว่าพอร์ตหมายเลขใดควรเหมาะสำหรับ Service ใด และได้กำหนดใน Request For Comments (RFC') 1700 ตัวอย่างเช่น เลือกใช้ TCP พอร์ตหมายเลข 23 กับ Service Telnet และเลือกใช้ UDP พอร์ตหมายเลข 69 สำหรับ Service Trivial File transfer Protocol (TFTP) ตัวอย่างต่อไปนี้เป็นบางส่วนของ File/etc/services แสดงให้เห็นว่า หมายเลขพอร์ตแต่ละหมายเลขได้ถูกจับคู่กับ Transport Protocol หนึ่งหรือสอง Protocol ซึ่งหมายความว่า UPP หรือ TCP อาจจะใช้ หมายเลขพอร์ตเดียวกันก็ได้ เนื่องจากเป็น Protocol ที่ต่างกัน

ทั้งนี้หมายเลข Port ถูกจัดแบ่งเป็น 2 ประเภท ตามที่ได้กำหนดใน RFC' 1700 คือ Well Known Ports และ Registered Ports

1) Well Known Ports คือจะเป็นพอร์ตที่ระบบส่วนใหญ่กำหนดให้ใช้โดย Privileged User (ผู้ใช้ที่มีสิทธิพิเศษ) โดยพอร์ตเหล่านี้ใช้สำหรับการติดต่อระหว่างเครื่องที่มีระบบเวลาที่ยาวนาน วัตถุประสงค์เพื่อให้ Service แก่ผู้ใช้ (ที่ไม่รู้จักหรือคุ้นเคย) แปลกหน้า จึงจำเป็นต้องกำหนดพอร์ตติดต่อสำหรับ Service นั้นๆ

2) Registered Ports จะเป็นพอร์ตหมายเลข 1024 ขึ้นไป ซึ่ง IANA ไม่ได้กำหนดไว้

2.15 WinPcap

The Packet Capture and network monitor library for windows คือ Library หนึ่งของระบบปฏิบัติการวินโดวส์ที่มีหน้าที่สำหรับดักจับข้อมูลต่างๆในเครือข่าย ซึ่ง winpcap จะทำงานเป็นเคอร์เนลตัวหนึ่งของระบบปฏิบัติการวินโดวส์ เราจึงสามารถเรียกใช้งานได้โดยตรง

เปรียบเสมือนกับเป็นไดรเวอร์นั่นเอง ซึ่ง WinPcap นั้นเป็น Open source เราสามารถนำมาพัฒนาเองได้ตามต้องการ แต่โดยปกติแล้วโปรแกรมที่ใช้ดักรับข้อมูลทางเครือข่าย จะอาศัย winpcap เป็นไลบรารีหลัก อาทิเช่น Wireshark, Nmap, Cain, Snort, etc.

2.15.1 WinPcap ประกอบด้วย

1) Network Driver Interface Specification (NDIS) เพื่ออ่านแพ็คเก็จโดยตรงจากเน็ทเวิร์คอแด็ปเตอร์
2) มีการอิมพลีเมนท์ในระดับต่ำของไลบรารีในระบบปฏิบัติการ เพื่อเชื่อมต่อกับไดรเวอร์
3) พอร์ตของ libpcap ที่ใช้ API ของการอิมพลีเมนท์ในระดับต่ำของไลบรารี

บทที่ 3
การออกแบบและพัฒนา

3.1 รายละเอียดโปรแกรมที่พัฒนา (Software Specification)

3.1.1 Input Specification

1) การเปลี่ยนแปลงต่างๆที่มีผลต่อโพรเซส
2) การเปลี่ยนแปลงต่างๆที่มีผลต่อไฟล์
3) การเปลี่ยนแปลงต่างๆที่มีผลต่อรีจิสทรี
4) การเปลี่ยนแปลงต่างๆที่มีผลต่อพอร์ตเครือข่าย
5) การเปลี่ยนแปลงต่างๆที่มีผลต่อการดักจับ API
6) ข้อมูลต่างๆที่ถูกส่งผ่านอินเทอร์เฟซการ์ดแลน

3.1.2 Output Specification

1) รายงานผลการเปลี่ยนแปลงที่มีต่อระบบ
2) ผลลัพธ์ที่ได้จากการวิเคราะห์พฤติกรรมว่าใช่มัลแวร์หรือไม่

3.1.3 Functional Specification

1) ระบบสามารถแสดงการเปลี่ยนแปลงต่างๆที่เกิดกับโปรเซส รีจิสทรี ไฟล์ พอร์ต API และอินเทอร์เฟซการ์ดแลนได้
2) ระบบสามารถตรวจจับมัลแวร์ ที่มีพฤติกรรมคล้ายกับพฤติกรรมต้นแบบที่ระบุได้ โดยมีกล่องข้อความแจ้งเตือนขึ้นมา
3) ระบบไฟล์มอนิเตอร์สามารถตรวจสอบการอ่านและการเขียนไฟล์ได้
4) ระบบรีจิสทรีมอนิเตอร์สามารถตรวจสอบการเขียนคีย์หรือค่าต่างๆลงบนรีจิสทรีของวินโดวส์ได้ เช่น OpenKey, CreateKey, Close-Key, EnumerateValKey, EnumerateKey, QueryValKey, QueryKey, SetValKey, SetKey, DeleteValKey และ DeleteKey
5) ระบบโพรเซสมอนิเตอร์สามารถตรวจสอบการสร้างและการทำลายโพรเซสได้
6) ระบบพอร์ตมอนิเตอร์สามารถตรวจการเปิดพอร์ตเครือข่ายต่างๆได้
7) ตรวจจับการเรียกใช้ API ได้
8) ตรวจจับการรับส่งข้อมูลผ่านอินเทอร์เฟซการ์ดแลนได้

3.1.4 ขอบเขตของโปรแกรมที่พัฒนา

1) ระบบตรวจจับมัลแวร์นี้พัฒนาขึ้นเพื่อเป็นต้นแบบในการตรวจจับมัลแวร์เชิงพฤติกรรมดังนั้นจึงไม่สามารถดักจับมัลแวร์ได้ครอบคลุมทุกตัว จะสามารถจับได้

เฉพาะพฤติกรรมส่วนใหญ่ของมัลแวร์นั้นที่เกิดขึ้น

2) ระบบตรวจจับมัลแวร์นี้พัฒนาขึ้นเพื่อทดลองใช้กับระบบปฏิบัติการวินโดวส์ XP เนื่องจากเราสามารถเขียนไดรเวอร์ได้เองโดยไม่ต้องขออนุญาต

3) ระบบตรวจจับมัลแวร์นี้ไม่ได้พัฒนาขึ้นเพื่อมาแทนที่ระบบการตรวจจับแบบ Signature-based แต่พัฒนาขึ้นเพื่อมาทำงานควบคู่กัน เพื่อป้องกันมัลแวร์แบบ Zero-day Attack

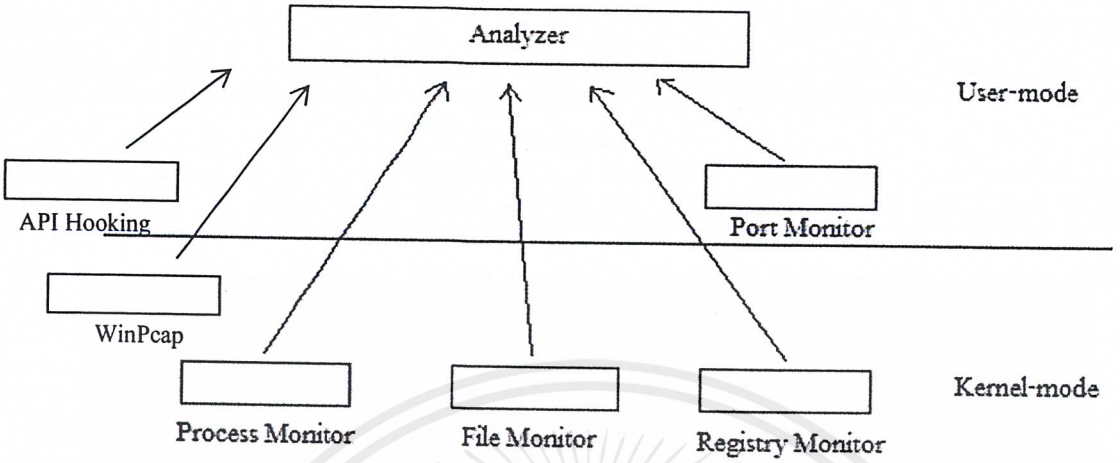
3.1.5 ข้อจำกัดของโปรแกรมที่พัฒนา

1) ระบบตรวจจับมัลแวร์นี้สามารถใช้งานได้กับระบบปฏิบัติการวินโดวส์ XP
2) ระบบตรวจจับมัลแวร์นี้พิจารณาเฉพาะพฤติกรรมของแต่ละโพรเซสเท่านั้น
3) ไม่สามารถตรวจจับมัลแวร์ที่ไม่ได้มีการระบุพฤติกรรมเอาไว้ได้
4) ไม่สามารถดำเนินการแก้ไขไฟล์และค่ารีจิสทรีต่างๆที่มัลแวร์เปลี่ยนแปลงไปแล้วได้
5) ไม่สามารถตรวจจับมัลแวร์ได้หากมัลแวร์นั้นไม่แสดงพฤติกรรมออกมา

3.1.6 เครื่องมือที่ใช้ในการพัฒนา

1) สภาพแวดล้อมในการพัฒนา คือ ระบบปฏิบัติการ Microsoft Windows XP Professional Service Pack 2
2) โปรแกรมต้นแบบที่ใช้ในการพัฒนา Capture-BAT และ Netstatp
3) โปรแกรมและไลบรารีที่ใช้ในการพัฒนา
   - Microsoft Visual Studio 2005
   - Microsoft Platform SDK for Windows Server 2003 R2
   - Windows Driver Kit 6000
   - The Boost C++ libraries 1.34.0
   - WinPcap version 4.1.2
4) ภาษาที่ใช้ในการพัฒนาโปรแกรม คือ ภาษา C/C++

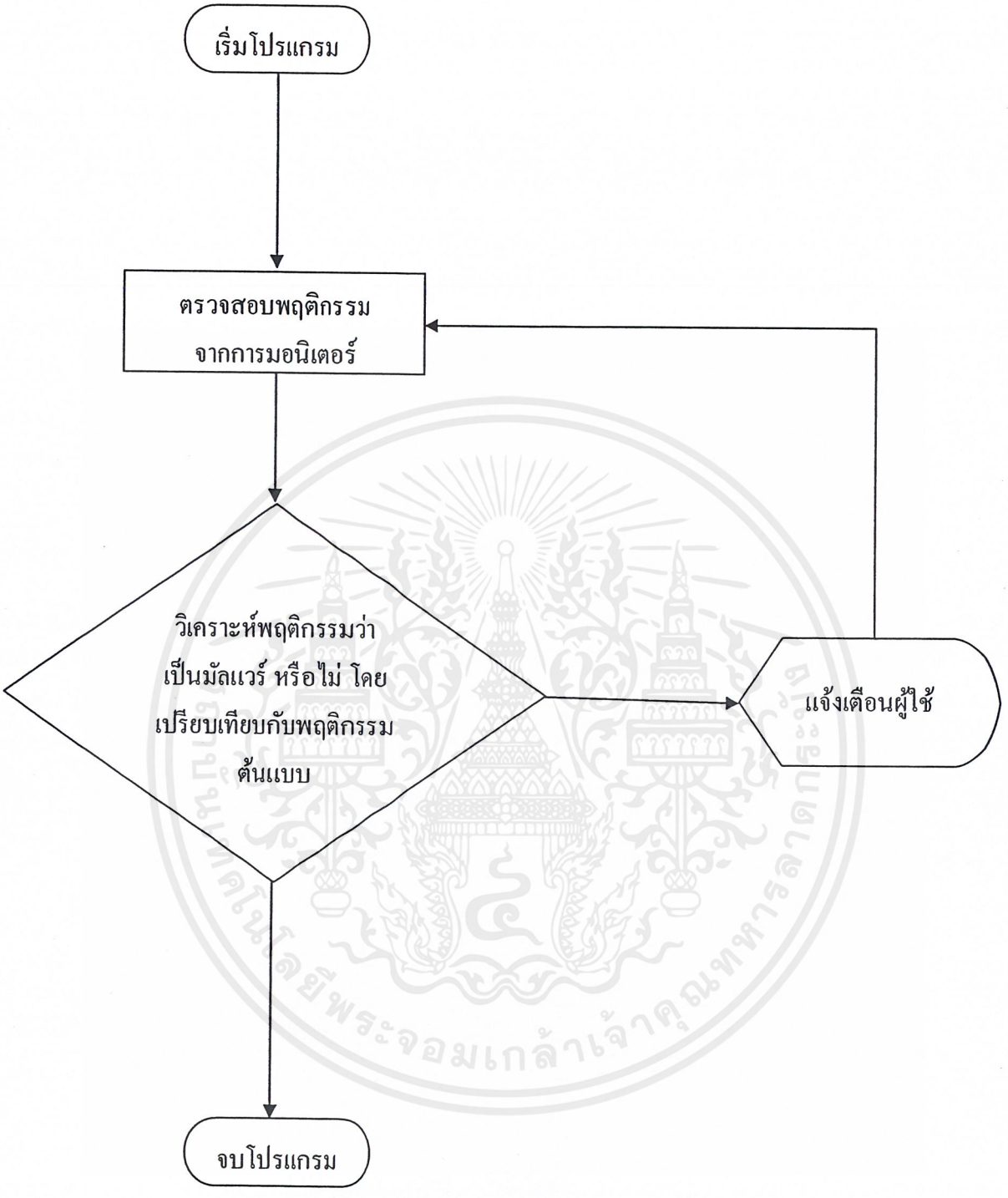
3.2 โครงสร้างและการทำงานของโปรแกรม



รูป 3.1 การทำงานของโปรแกรม

ส่วนสังเกตการณ์ซึ่งประกอบไปด้วย 6 ส่วน คือโพรเซสมอนิเตอร์ ไฟล์มอนิเตอร์ รีจิสทรีมอนิเตอร์ พอร์ตมอนิเตอร์ การดักจับ API และ WinPcap (การดักจับที่ส่วนอินเทอร์เฟซของการ์ดแลน) ซึ่งจะคอยดักจับการกระทำหรือพฤติกรรมต่างๆที่เกิดการเปลี่ยนแปลงขึ้นภายในระบบ โดยจะทำงานในระดับเคอร์เนลเพื่อให้ได้มาซึ่งข้อมูลที่ถูกต้อง เนื่องจากมัลแวร์บางชนิดสามารถซ่อนพฤติกรรมของตนเองในระดับยูสเซอร์โหมดได้ เมื่อระบบสามารถดักจับการกระทำนั้นๆได้แล้วจะทำการส่งข้อไปยังบัฟเฟอร์ที่สร้างขึ้น เพื่อให้กับส่วนวิเคราะห์พฤติกรรมที่ทำงานในยูสเซอร์โหมด

ส่วนวิเคราะห์พฤติกรรม หลังจากที่ได้รับข้อมูลพฤติกรรมหรือการเปลี่ยนแปลงต่างๆที่เกิดขึ้นภายในระบบแล้ว จะนำมาเทียบกับรายการยกเว้น (Exclusion list) เพื่อตัดรายการข้อมูลบางอย่างออกไป เนื่องจากว่าในระบบปฏิบัติการวินโดวส์ขณะอยู่เฉย (Idle) มีการเปลี่ยนแปลงข้อมูลภายในตัวเองอยู่แล้ว เพื่อให้ได้เฉพาะข้อมูลที่ต้องการวิเคราะห์เท่านั้น จากนั้นจึงนำมาเปรียบเทียบกับฐานข้อมูลพฤติกรรมของมัลแวร์ที่ระบุไว้ว่าตรงกันหรือไม่



รูป 3.2 การทำงานส่วนตรวจจับพฤติกรรมที่ผิดปกติและแจ้งเตือน

บทที่ 4
การทดลอง

4.1 บทนำ

4.1.1 เครื่องมือที่ใช้ในการทดสอบโปรแกรม

ในขั้นตอนการทำการทดสอบระบบตรวจจับมัลแวร์เชิงพฤติกรรมสำหรับระบบปฏิบัติการวินโดวส์ กลุ่มผู้วิจัยได้ใช้การทดสอบดังนี้

4.1.1.1 เครื่องคอมพิวเตอร์

1) Intel(R) Core(TM) 2 Duo CPU T7500 ความเร็ว 2.20 GHz
2) RAM 2.00 GB
3) อุปกรณ์พื้นฐาน ได้แก่ หน้าจอ คีย์บอร์ด เมาส์

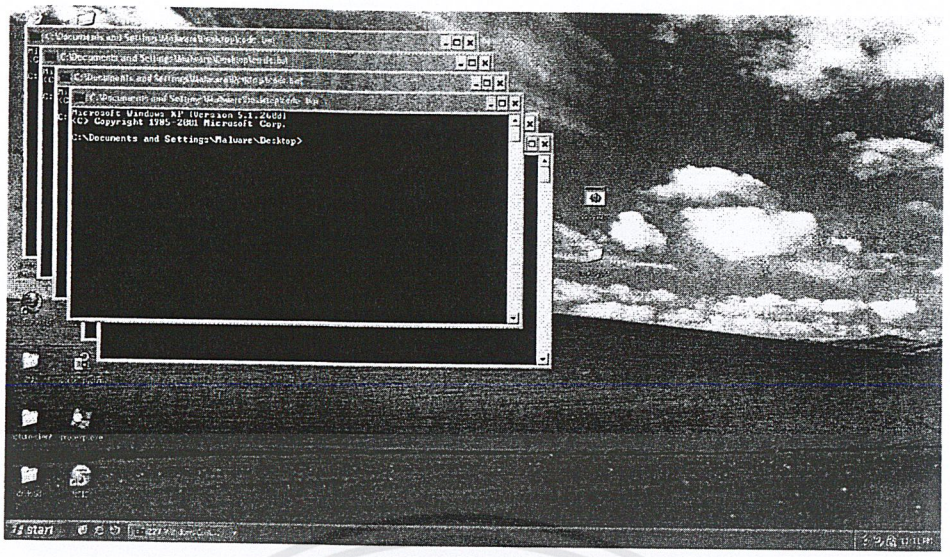
4.1.1.2 เครื่องมือทางด้านซอฟท์แวร์

1) ระบบปฏิบัติการ Microsoft Windows XP Professional Service Pack 2
2) โปรแกรมระบบตรวจจับมัลแวร์เชิงพฤติกรรม
3) ตัวอย่างมัลแวร์เพื่อใช้ในการทดสอบ

4.2 การทดลองเขียนมัลแวร์

4.2.1 การทดลองเขียน Rabbit

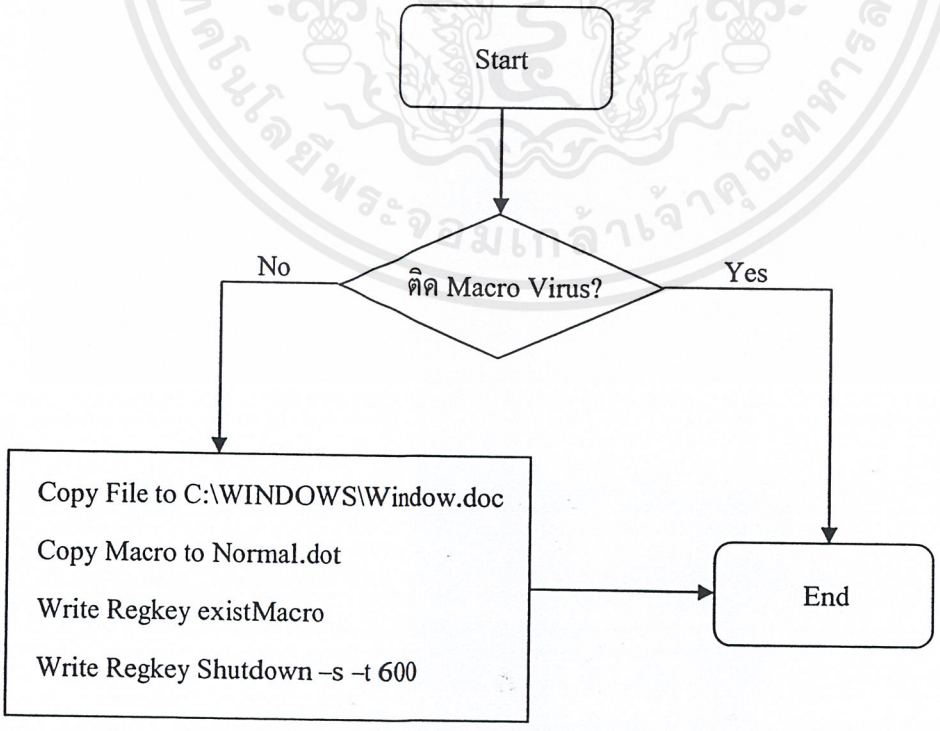
Rabbit เป็นมัลแวร์ที่ทำสำเนาตัวเองซ้ำๆ หรือเรียกอะไรบ้างอย่างไปเรื่อยๆ หรือเรียกว่าเกิด Fork Bomb ก็ได้ จากการทำการทดลองโดยการเขียนแบทไฟล์ (.bat) เพื่อเรียกหน้าต่าง Command Prompt ขึ้นมาเรื่อยๆไม่รู้จบ เพื่อทำให้หน่วยความจำ หรือแรมเต็ม ทำให้ระบบปฏิบัติการทำงานได้ช้าลงเรื่อยๆจนไม่สามารถดำเนินการอื่นๆได้อีก และต้องทำการปิดเครื่องเพื่อเปิดระบบปฏิบัติการขึ้นมาใหม่ ดังรูป 4.1



รูป 4.1 การทำงานของ Rabbit

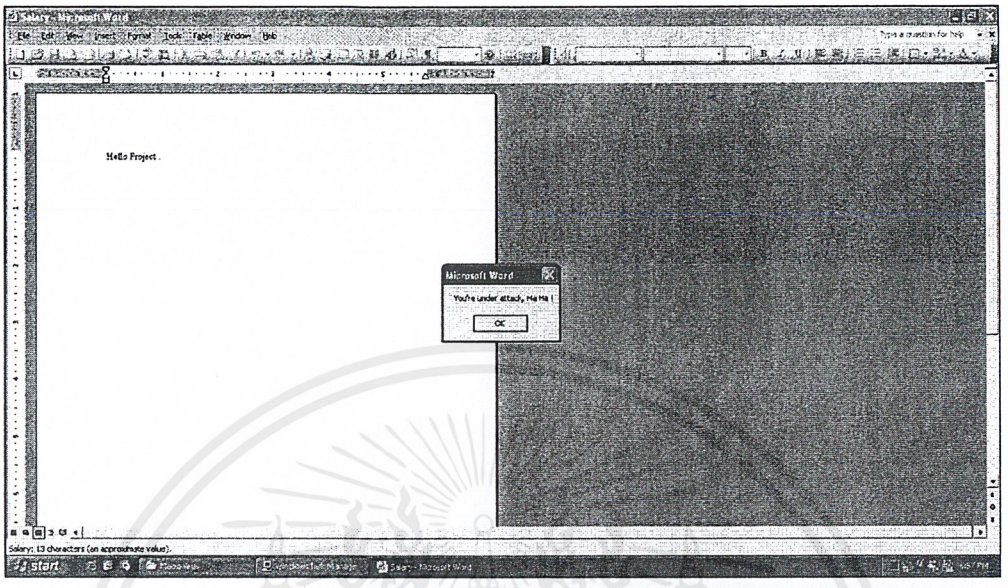
4.2.2 การทดลองเขียนแมโครไวรัส

แมโครไวรัสเป็นไวรัสที่มักแนบมากับไฟล์เอกสารต่างๆ ทำการทดลองโดยการเขียนโค้ดแมโครฝังไว้กับไฟล์เอกสารไมโครซอฟท์เวิร์ด เมื่อเริ่มเปิดเอกสารจะมีการตรวจสอบว่าติดแมโครหรือยัง ถ้าติดแมโครแล้วจะไม่กระทำการใดๆอีก แต่ถ้ายังไม่ติดแมโครจะทำการคัดลอกไฟล์ไปยัง C:\WINDOWS\Window.doc และคัดลอกแมโครไปยัง Normal.dot และเขียน Regkey existMacro และ Regkey Shutdown –s –t 600 โดยมีกระบวนการทำงานดังรูป 4.2



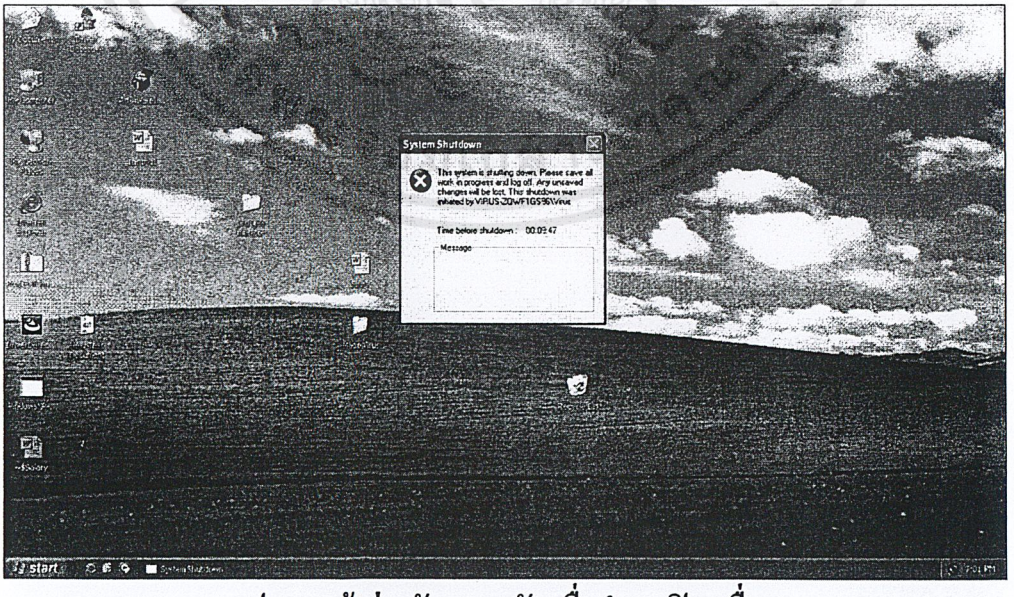
รูป 4.2 การทำงานของ Macro Virus

จากการทดลองเมื่อเปิดไฟล์ที่ติดแมโครไวรัสขึ้นมา จะปรากฏกล่องข้อความ “You’re Under attack haha!” ดังรูป 4.3



รูป 4.3 กล่องข้อความแจ้งว่าติดแมโครไวรัสแล้ว

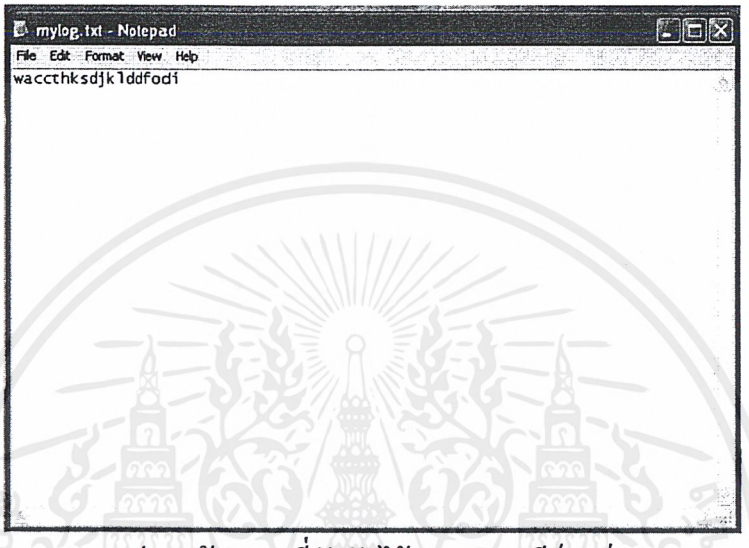
แสดงว่าเครื่องเราติดแมโครไวรัสแล้ว และทุกๆครั้งที่มีการเปิดไฟล์เอกสารไมโครซอฟท์เวิร์ดอื่นๆ จะมีหน้าต่างนี้แสดงขึ้นมา และทุกครั้งที่มีการเปิดเครื่องคอมพิวเตอร์ขึ้นมาใหม่ เครื่องคอมพิวเตอร์จะแสดงหน้าต่างนับถอยหลัง และจะทำการปิดเครื่องลงภายใน 10 นาที ดังรูป 4.4



รูป 4.4 หน้าต่างนับถอยหลัง เพื่อทำการปิดเครื่อง

4.2.3 การทดลองเขียน Key Logger

Key Logger เป็นมัลแวร์ที่ใช้ในการดักจับค่าในการกดคีย์บอร์ด ทำการทดลองโดยการเขียนโปรแกรมเพื่อเรียกใช้ฟังก์ชัน GetAsyncKeyState() เมื่อรันโปรแกรมแล้ว และเมื่อเรากดปุ่มใดๆบนคีย์บอร์ด จะถูกเก็บค่าไว้ในเท็กซ์ไฟล์ด้วย ทำให้เราสามารถถูกดักจับพาสเวิร์ดได้โดยง่าย ดังรูป 4.5

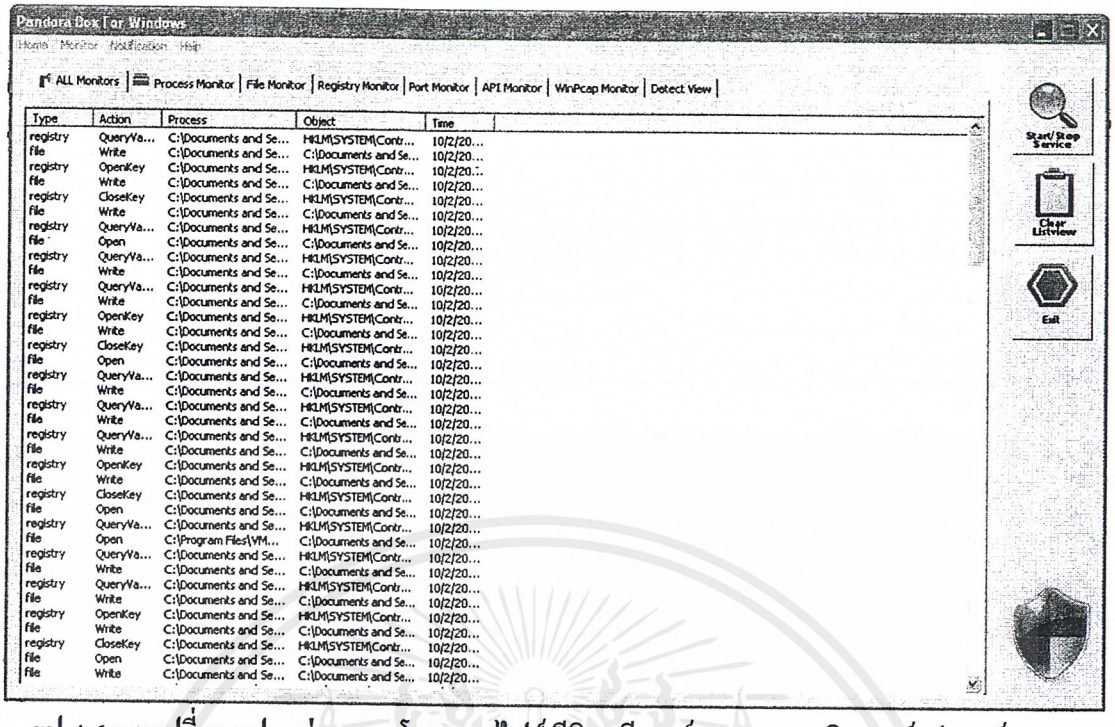


รูป 4.5 ข้อความที่ดักจับได้จากการกดคีย์บอร์ด

4.3 การสังเกตการณ์พฤติกรรมของมัลแวร์

4.3.1 วิธีการสังเกตการณ์พฤติกรรมของมัลแวร์

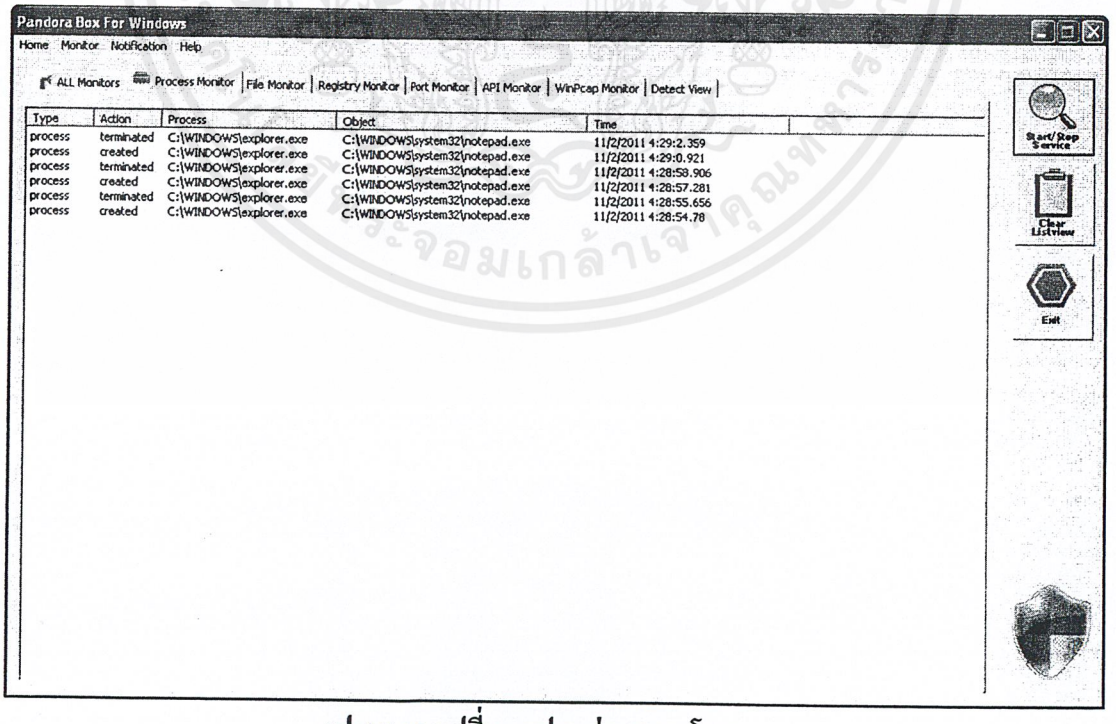
1) เมื่อทำการรันโปรแกรมจะได้หน้าต่างแสดงการเปลี่ยนแปลงต่างๆของโพรเซส ไฟล์ รีจิสทรี พอร์ต API และอินเทอร์เฟซการ์ดแลน ดังรูป 4.6



รูป 4.6 การเปลี่ยนแปลงต่างๆของโพรเซส ไฟล์ รีจิสทรี พอร์ต API และอินเทอร์เฟซการ์ดแลน

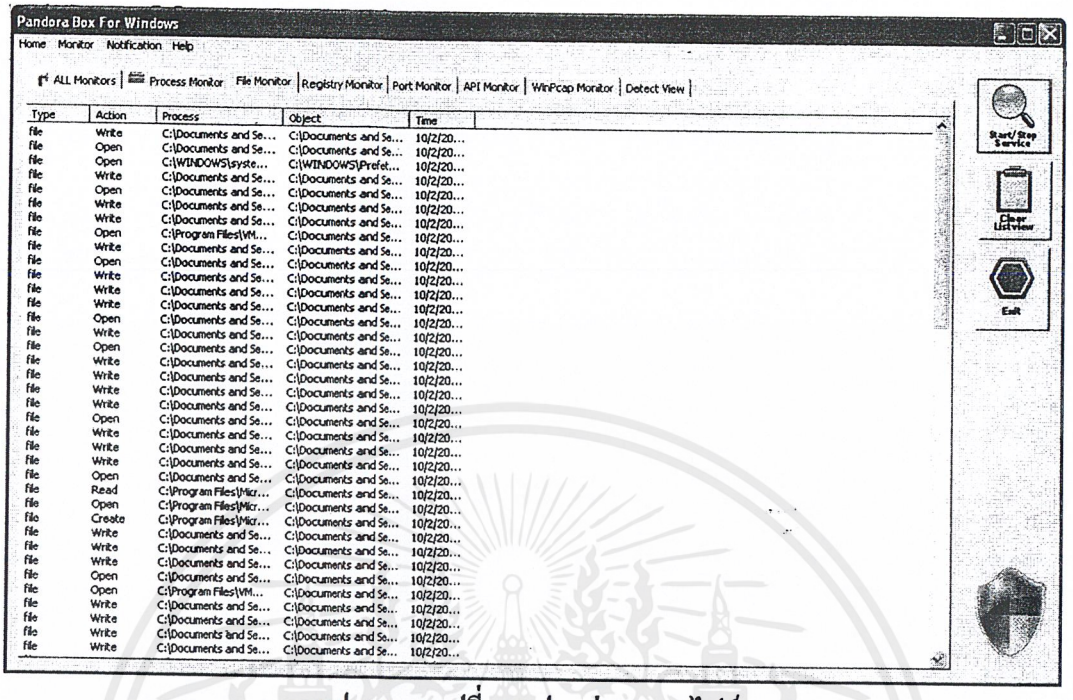
4.3.2 ผลการสังเกตการณ์พฤติกรรมของมัลแวร์

1) คลิกที่แท็บ Process Monitor เมื่อต้องการแสดงการเปลี่ยนแปลงต่างๆของโพรเซส ดังรูป 4.7



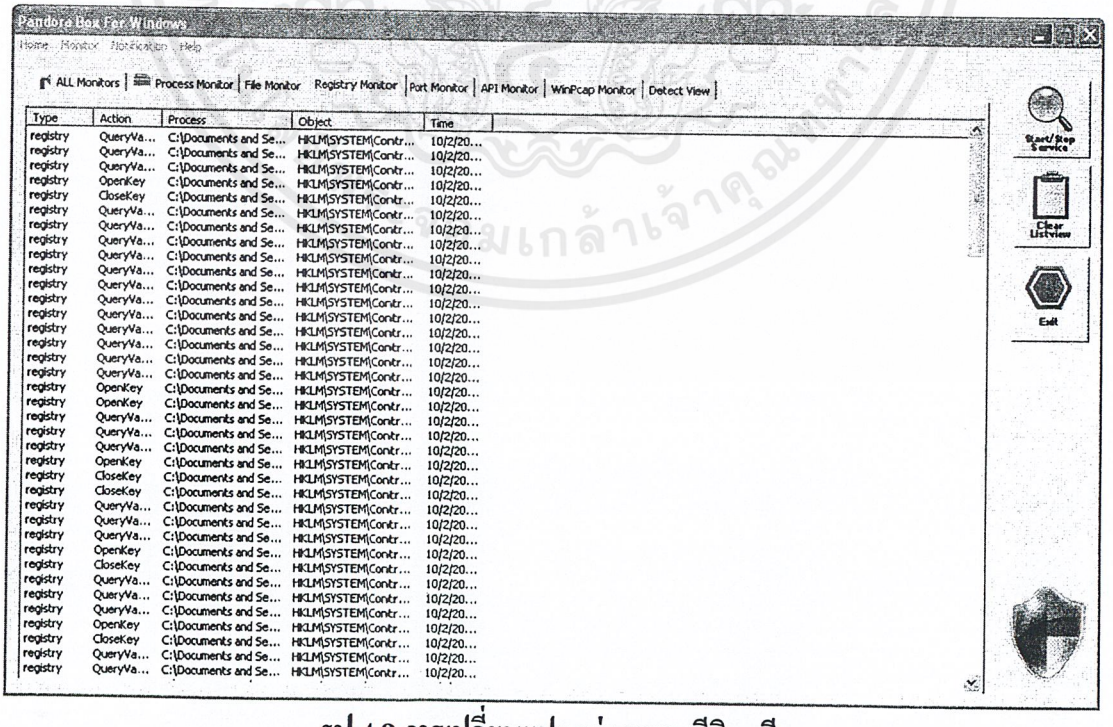
รูป 4.7 การเปลี่ยนแปลงต่างๆของโพรเซส

2) คลิกที่แท็บ File Monitor เมื่อต้องการแสดงการเปลี่ยนแปลงต่างๆของไฟล์ ดังรูป 4.8



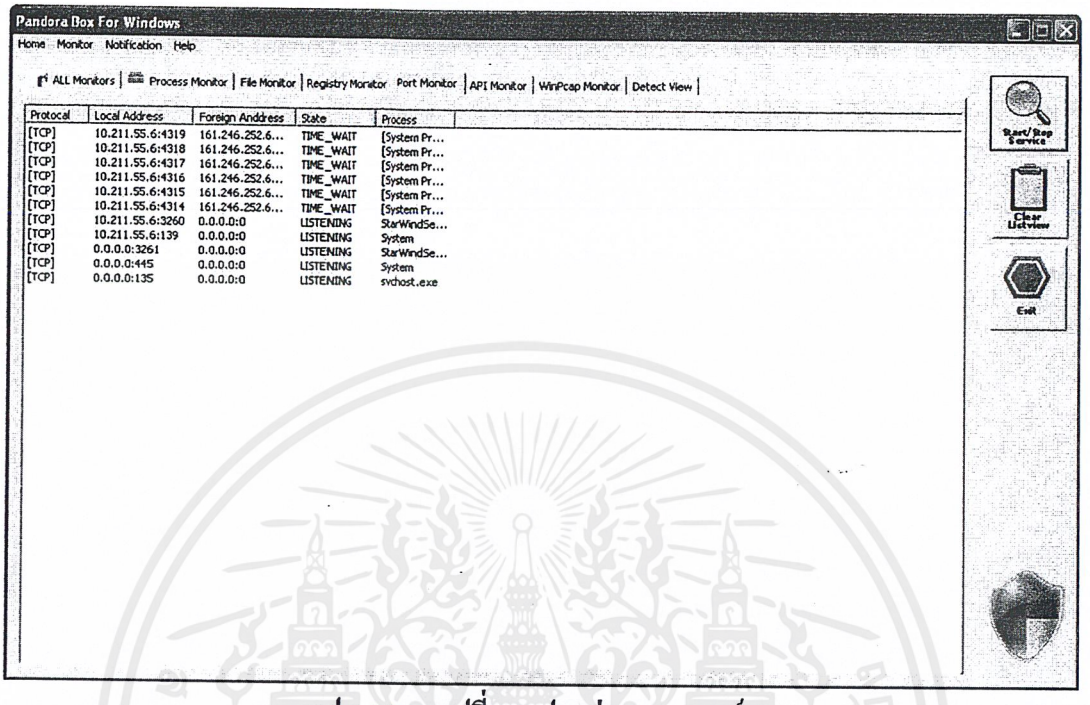
รูป 4.8 การเปลี่ยนแปลงต่างๆของไฟล์

3) คลิกที่แท็บ Registry Monitor เมื่อต้องการแสดงการเปลี่ยนแปลงต่างๆของรีจิสทรี ดังรูป 4.9



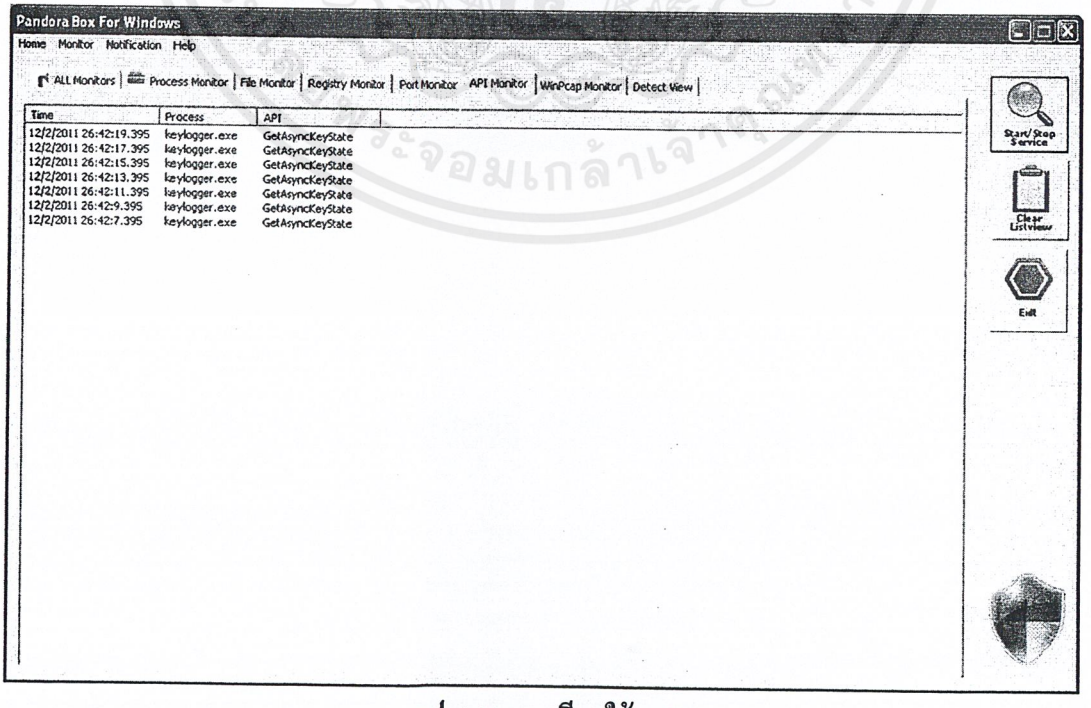
รูป 4.9 การเปลี่ยนแปลงต่างๆของรีจิสทรี

4) คลิกที่แท็บ Port Monitor เมื่อต้องการแสดงการเปลี่ยนแปลงต่างๆของพอร์ต ดังรูป 4.10



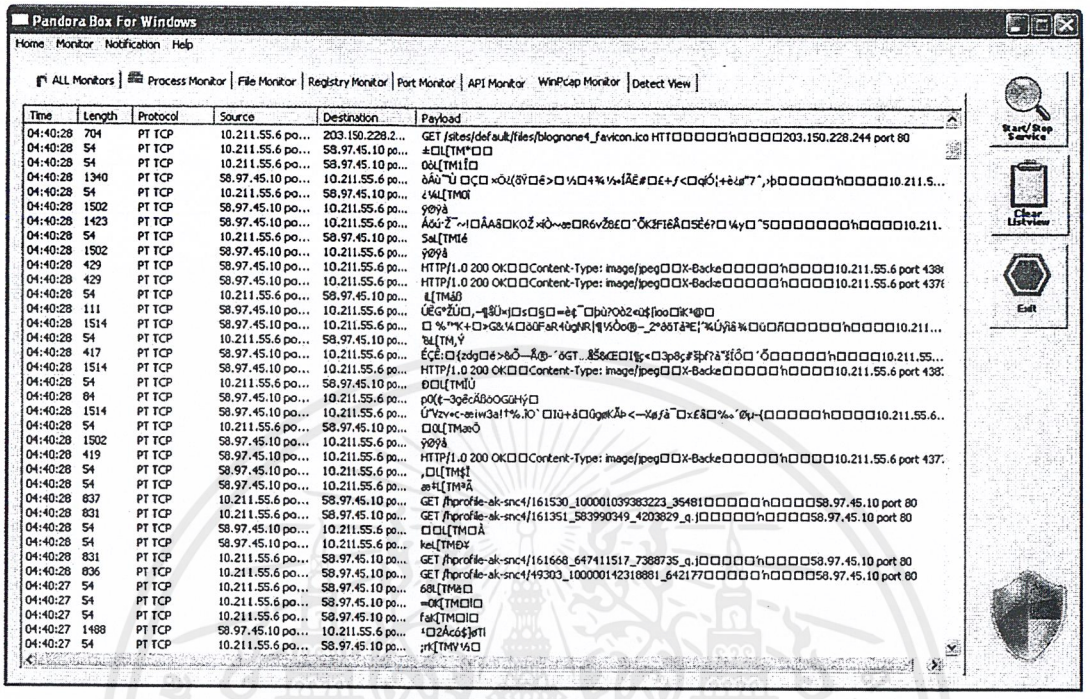
รูป 4.10 การเปลี่ยนแปลงต่างๆของพอร์ต

5) คลิกที่แท็บ API Monitor เมื่อต้องการแสดงการเปลี่ยนแปลงต่างๆของ API ดังรูป 4.11



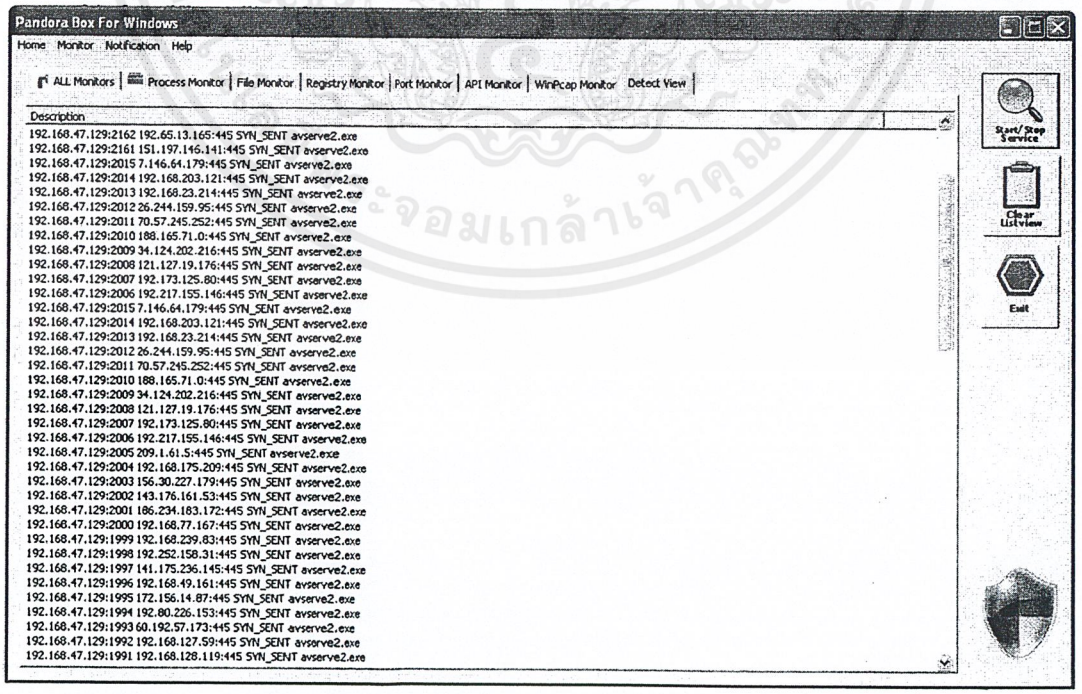
รูป 4.11 การเรียกใช้ API

6) คลิกที่แท็บ WinPcap Monitor เมื่อต้องการแสดงการเปลี่ยนแปลงต่างๆของอินเทอร์เฟซการ์ดแลน ดังรูป 4.12



รูป 4.12 การรับส่งข้อมูลผ่านอินเทอร์เฟซการ์ดแลน

7) คลิกที่แท็บ Detect View เมื่อต้องการแสดงบันทึกการตรวจจับมัลแวร์ ดังรูป 4.13

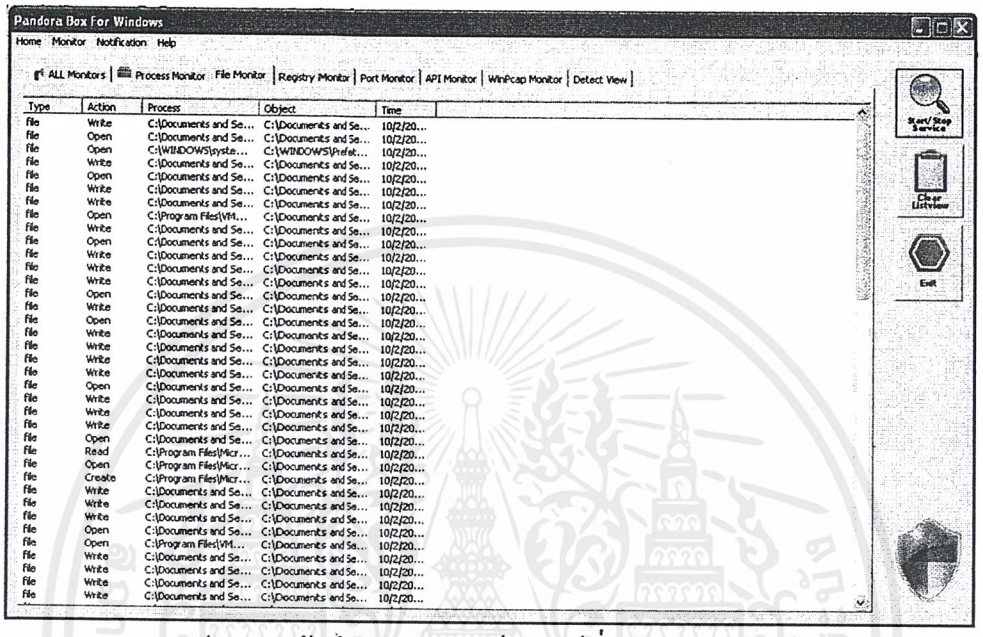


รูป 4.13 บันทึกการตรวจจับมัลแวร์

4.4 การทดลองตรวจจับพฤติกรรมของมัลแวร์ที่มีการสร้างไฟล์และลบไฟล์

4.4.1 การสังเกตการณ์พฤติกรรมของมัลแวร์

จากการสังเกตการณ์การเปลี่ยนแปลงของไฟล์ พบว่ามีไฟล์ต่างๆเปิดใช้งานอยู่ใน C:\Windows\System32



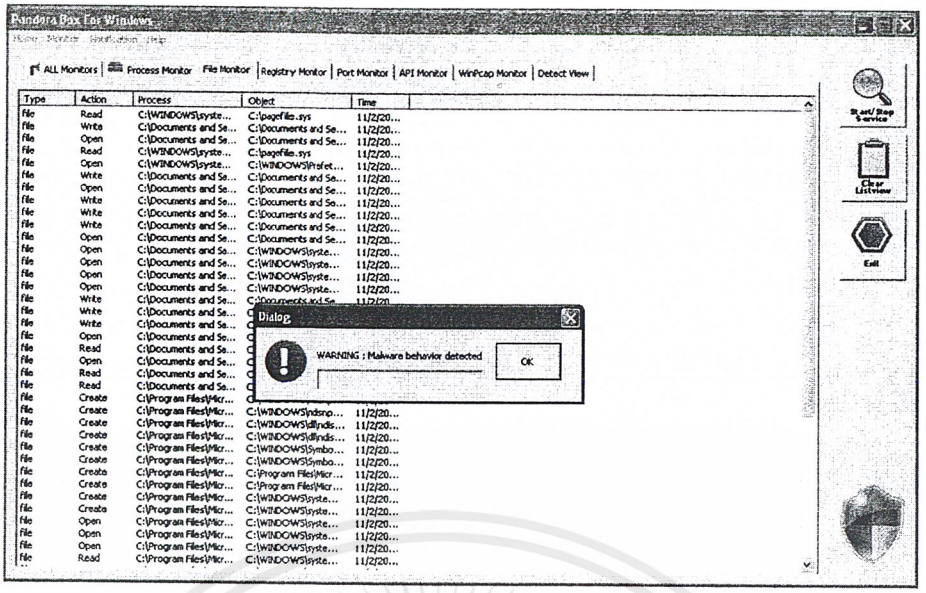
รูป 4.14 หน้าต่างสังเกตการณ์มัลแวร์ที่สร้างและลบไฟล์

4.4.2 การวิเคราะห์พฤติกรรมของมัลแวร์

จากการทดลองเมื่อสังเกตการเปลี่ยนแปลงของไฟล์ ไฟล์ต่างๆที่อยู่ใน C:\Windows\System32 ซึ่งเป็นส่วนที่เก็บไฟล์สำคัญของระบบปฏิบัติการ ดังนั้น ถ้ามีการลบไฟล์โดยที่ได้ทำการลงโปรแกรมหรือลบโปรแกรมแล้วนั้น พฤติกรรมนั้นมีโอกาสเสี่ยงที่จะเป็นมัลแวร์

4.4.3 การตรวจจับมัลแวร์

เมื่อมีการลบไฟล์ ntdll.dll ของระบบ ที่ C:\Windows\System32 จะมีการแจ้งเตือนว่ามีการตรวจพบมัลแวร์ดังรูป 4.15



รูป 4.15 หน้าต่างตรวจพบมัลแวร์

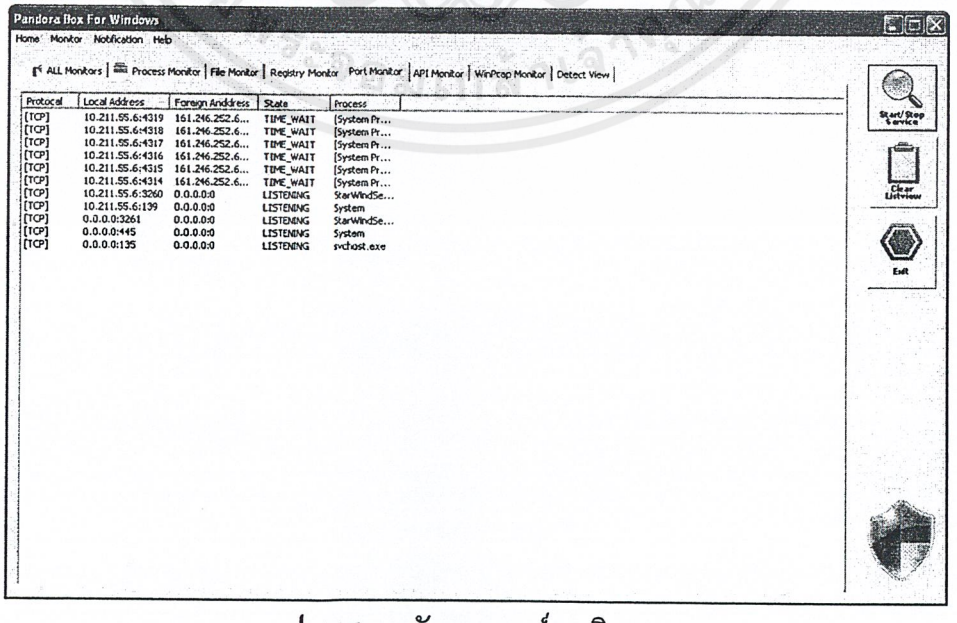
4.4.4 ผลการตรวจจับมัลแวร์

จากการทดลองเมื่อทำการสังเกตการณ์พฤติกรรมของมัลแวร์ สามารถตรวจพบการลบไฟล์ระบบ แล้วทำการแจ้งเตือนแก่ผู้ใช้งานได้

4.5 การทดลองตรวจจับพฤติกรรมของเวิร์มที่มีการโจมตีผ่านพอร์ต 445

4.5.1 การสังเกตการณ์พฤติกรรมของมัลแวร์

จากการสังเกตการณ์การเปิดของพอร์ต พบว่ามีพอร์ตต่างๆเปิดใช้งานอยู่และมีการส่งแพ็คเก็จประเภทต่างๆออกไปทั้งในและนอกเครือข่าย



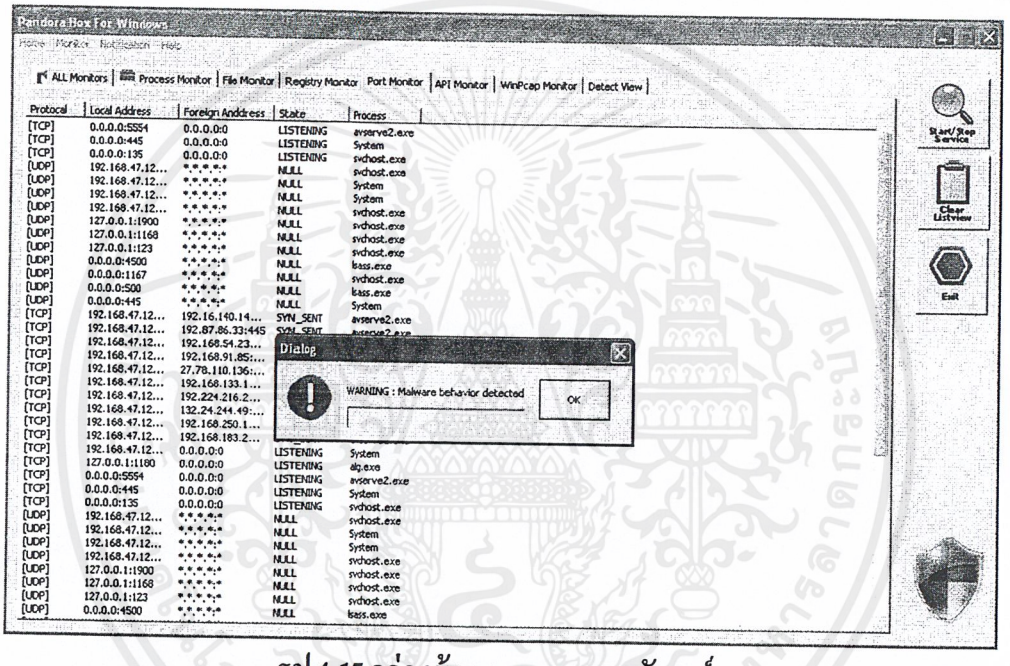
รูป 4.16 การสังเกตการณ์พฤติกรรม

4.5.2 การวิเคราะห์พฤติกรรมของมัลแวร์

จากการทดลองพบว่ามีพอร์ต 445 เปิด จากการทำงานของพอร์ต 445 เป็นพอร์ตที่ใช้ในการแชร์ไฟล์ของระบบปฏิบัติการวินโดวส์ในเครือข่ายเดียวกัน และเมื่อมีการส่งแพ็คเก็จประเภท Sync Sent ออกไปนอกเครือข่ายทำให้ทราบได้ว่านั่นคือ พฤติกรรมที่มีความเสี่ยงติดมัลแวร์

4.5.3 การตรวจจับมัลแวร์

เมื่อมีการตรวจสอบการเปิดของพอร์ต และพบว่ามีการติดต่อกันของพอร์ต 445 ข้ามเครือข่าย นั่นแสดงว่ามีการพบมัลแวร์ในเครือข่าย จะมีการแจ้งเตือนว่ามีการตรวจพบมัลแวร์ ดังรูป 4.17



รูป 4.17 กล่องข้อความตรวจพบมัลแวร์

4.5.4 ผลการตรวจจับมัลแวร์

จากการทดลองเมื่อทำการสังเกตการณ์พฤติกรรมของมัลแวร์ สามารถตรวจพบการสร้างการติดต่อกันของพอร์ต 445 ข้ามเครือข่าย ทำให้ทราบได้ว่าพฤติกรรมนี้เป็นพฤติกรรมของมัลแวร์ จึงทำการแจ้งเตือนแก่ผู้ใช้งานได้

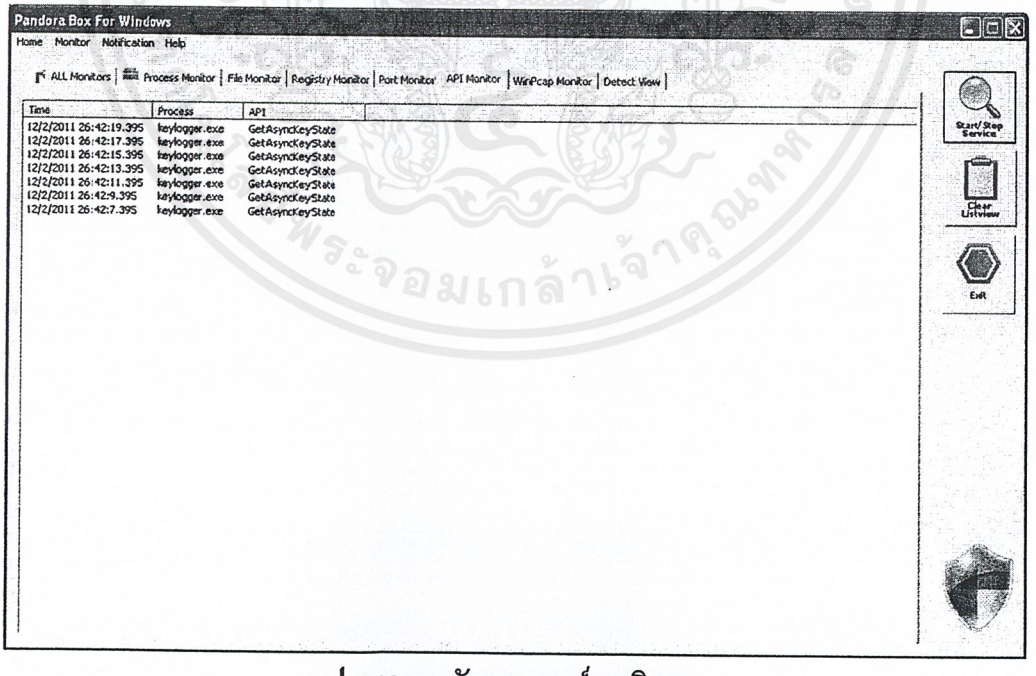
W32.Aizu	W32.Bobax	W32.Bolgi.Worm
W32.Cissi	W32.Cycle	W32.Explet
W32.HLLW.Deborms	W32.HLLW.Deloder	W32.HLLW.Gaobot
W32.HLLW.Lioten	W32.HLLW.Moega	W32.HLLW.Nebiwo
W32.HLLW.Polybot	W32.Ifbo	W32.Janx
W32.Kibuv.Worm	W32.Kiman	W32.Korgo
W32.Mytob	W32.Reatle	W32.Sasser
W32.Scane	W32.Slackor	W32.Spybot
W32.Wallz	W32.Welchia	W32.Zotob

รูป 4.18 รายชื่อมัลแวร์ที่ใช้พอร์ต 445 ในการแพร่กระจาย

4.6 การทดลองตรวจจับพฤติกรรมของคีย์ล็อกเกอร์โดยใช้ API Hooking

4.6.1 การสังเกตการณ์พฤติกรรมของมัลแวร์

จากการสังเกตการณ์ของการใช้งานโปรแกรมที่มีการเรียกใช้ API ที่ตรวจจับคีย์บนคีย์บอร์ด (Key Stoke API) พบว่ามีการเรียกใช้ API ที่ใช้ในการดักจับการกดคีย์บนคีย์บอร์ด



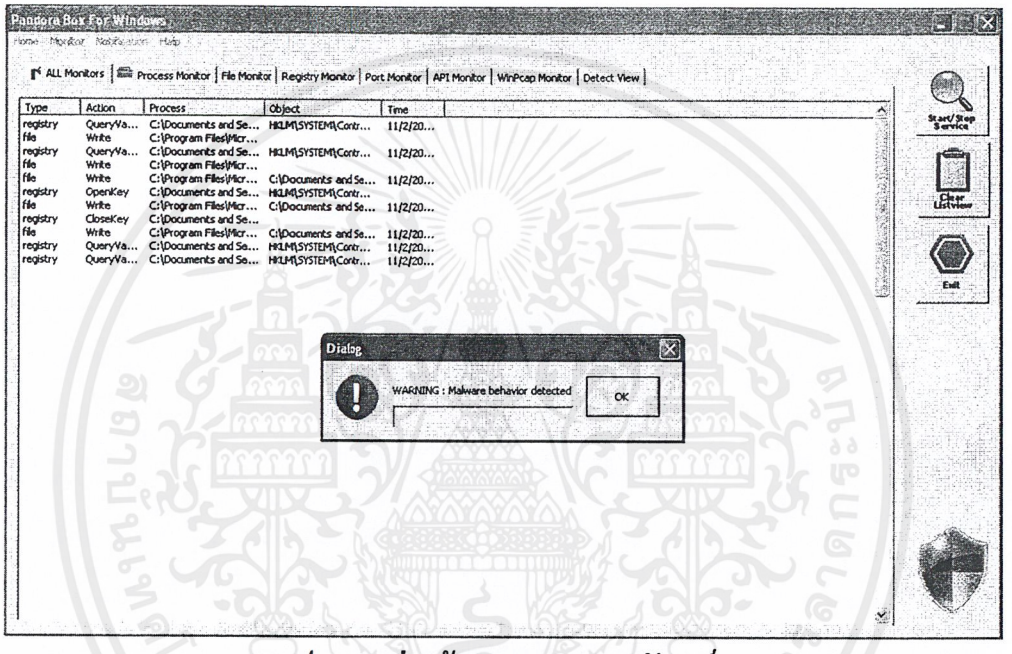
รูป 4.19 การสังเกตการณ์พฤติกรรม

4.6.2 การวิเคราะห์พฤติกรรมของมัลแวร์

จากการทดลองพบว่าเมื่อมีการใช้งานโปรแกรมทั่วไป จะมีการเรียกใช้ API ที่มีการดักจับการกดคีย์บนคีย์บอร์ด ซึ่งโดยปกติแล้วจะไม่มีการเรียกใช้ API ที่ทำการดักจับการกดคีย์บนคีย์บอร์ดทำให้ทราบได้ว่านั่นคือ พฤติกรรมที่มีความเสี่ยงติดมัลแวร์

4.6.3 การตรวจจับมัลแวร์

เมื่อมีการเรียกใช้ API ที่ใช้ในการดักจับการกดคีย์บนคีย์บอร์ด จะมีการแจ้งเตือนว่ามีการตรวจพบมัลแวร์ ดังรูป 4.20



รูป 4.20 กล่องข้อความตรวจพบมัลแวร์

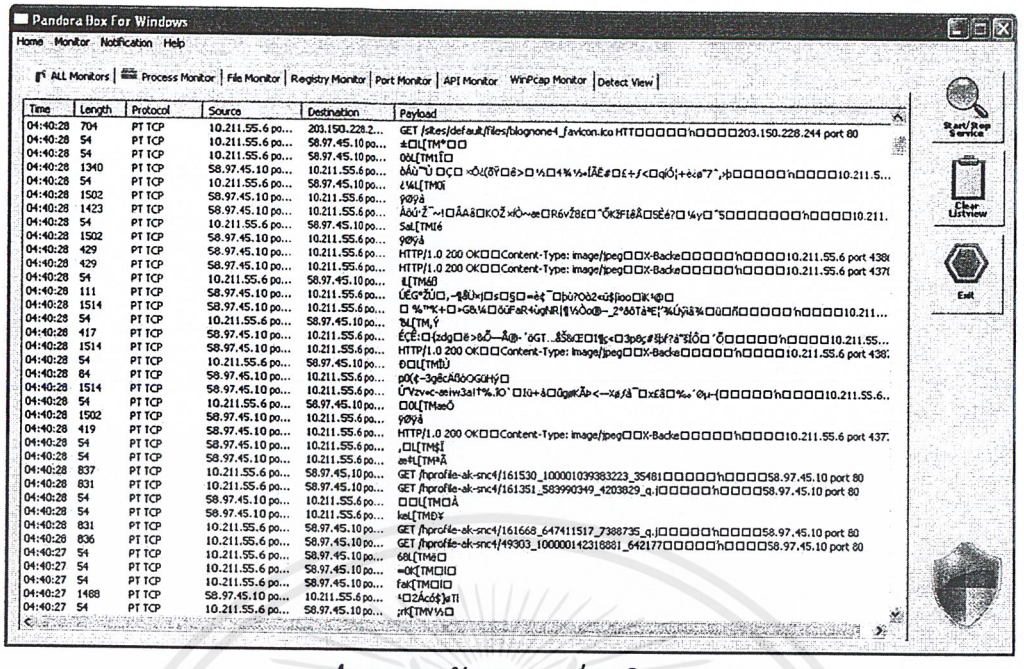
4.6.4 ผลการตรวจจับมัลแวร์

จากการทดลองเมื่อทำการเรียกใช้โปรแกรมโดยทั่วไป และมีการเรียกใช้ API ที่มีการดักจับคีย์บนคีย์บอร์ดซึ่งโดยปกติแล้วไม่ควรมีการทำงานเช่นนี้ ทำให้ทราบได้ว่าพฤติกรรมนี้เป็นพฤติกรรมของมัลแวร์ จึงทำการแจ้งเตือนแก่ผู้ใช้งานได้

4.7 การทดลองตรวจจับพฤติกรรมของบอทเน็ตที่ถูกควบคุมผ่านโปรโตคอล IRC

4.7.1 การสังเกตการณ์พฤติกรรมของมัลแวร์

จากการสังเกตการณ์การใช้งานที่มีการส่ง command ผ่านโปรโตคอล IRC พบว่ามีการส่ง command ผ่านโปรโตคอล IRC เป็นจำนวนมากในระยะเวลาสั้นๆ



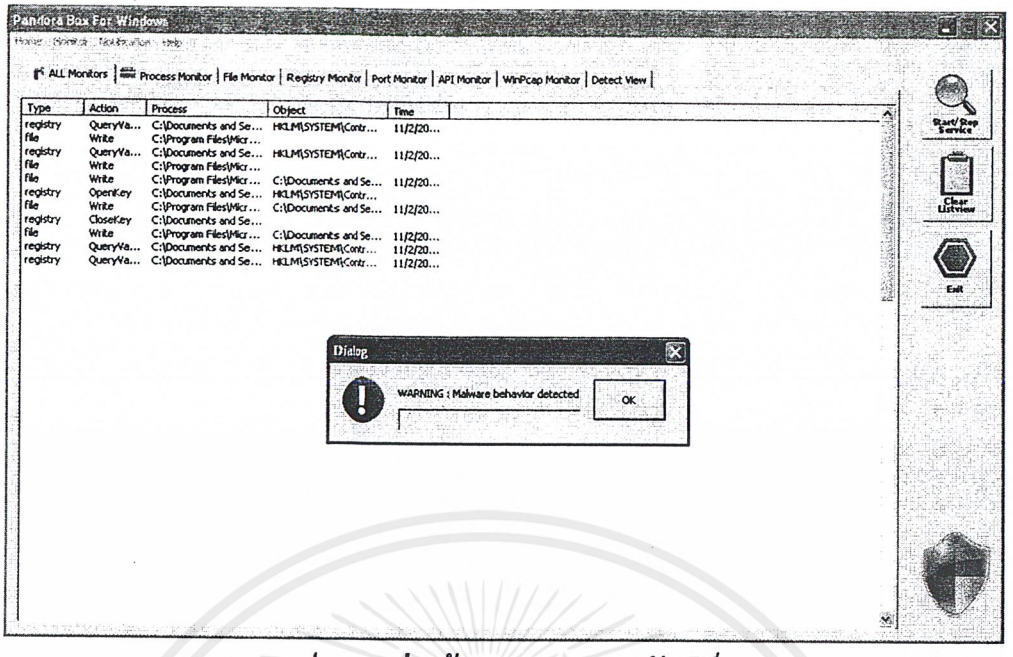
รูป 4.21 การสังเกตการณ์พฤติกรรม

4.7.2 การวิเคราะห์พฤติกรรมของมัลแวร์

จากการทดลองพบว่ามีการส่ง command ผ่านโปรโตคอล IRC เป็นจำนวนมากในระยะเวลาสั้นๆ ซึ่งโดยปกติแล้วการแชทหรือการเล่นเกมออนไลน์นั้นจะไม่มีจำนวนมากในระยะเวลาสั้นๆ ทำให้ทราบได้ว่านั่นคือ พฤติกรรมที่มีความเสี่ยงติดมัลแวร์

4.7.3 การตรวจจับมัลแวร์

เมื่อมีการตรวจสอบการส่ง command ผ่านโปรโตคอล IRC พบว่ามีการส่ง command ผ่านโปรโตคอล IRC เป็นจำนวนมากในระยะเวลาสั้นๆ นั่นแสดงว่ามีการพบมัลแวร์ในเครือข่าย จะมีการแจ้งเตือนว่ามีการตรวจพบมัลแวร์ ดังรูป 4.22



รูป 4.22 กล่องข้อความตรวจพบมัลแวร์

4.7.4 ผลการตรวจจับมัลแวร์

จากการทดลองเมื่อทำการสังเกตการณ์พฤติกรรมของมัลแวร์ สามารถตรวจพบการส่ง command ผ่านโปรโตคอล IRC เป็นจำนวนมากในระยะเวลาสั้นๆ ทำให้ทราบได้ว่าพฤติกรรมนี้เป็นพฤติกรรมของมัลแวร์ จึงทำการแจ้งเตือนแก่ผู้ใช้งานได้

บทที่ 5
บทสรุปและวิจารณ์

5.1 บทสรุป

จากการศึกษาและทดลองมัลแวร์สามารถใช้ประโยชน์จากระบบปฏิบัติการวินโดวส์ได้ในหลากหลายทางเพื่อให้บรรลุวัตถุประสงค์ร้ายๆของตัวเองได้ เนื่องจากวินโดวส์อนุญาตให้ผู้ใช้มีสิทธิ์ในการแก้ไขปรับแต่งระบบได้ และระบบตรวจจับมัลแวร์เชิงพฤติกรรมนี้สามารถตรวจจับมัลแวร์ได้จริง หากมัลแวร์นั้นมีการกระทำกับโพรเซส ไฟล์ รีจิสทรี พอร์ต การเรียกใช้ API และการใช้งานเครือข่ายที่ไม่เหมาะสมและตรงกับพฤติกรรมที่ระบุ

5.2 ปัญหาอุปสรรคและแนวทางแก้ไข

1) ปัญหาการหาข้อมูลและโค้ดต้นฉบับของมัลแวร์ได้ยาก เนื่องจากเป็นสิ่งอันตรายไม่ควรเผยแผ่
   แนวทางการแก้ไขปัญหา : หาได้จากเว็บใต้ดินซึ่งค่อนข้างจะเข้าถึงได้ลำบาก
2) ปัญหาการเขียนโปรแกรมระบบของวินโดวส์และไดรเวอร์ เนื่องจากไม่เคยเขียนมาก่อน
   แนวทางการแก้ไขปัญหา : หาและศึกษาเอกสารการสอนเบื้องต้นจากอินเทอร์เน็ต
3) ปัญหาการพัฒนาต้นแบบระบบตรวจจับมัลแวร์เชิงพฤติกรรม เนื่องจากเป็นการเขียนโปรแกรมในระดับเคอร์เนลของระบบปฏิบัติการวินโดวส์ ซึ่งมีความซับซ้อนและหากเขียนผิดพลาดก็อาจจะทำให้ระบบปฏิบัติการพังตัวลงได้
   แนวทางการแก้ไขปัญหา : หาโปรแกรมต้นแบบที่มีความสามารถในการปฏิบัติงานและมีความน่าเชื่อถือของข้อมูล ซึ่งโปรแกรมที่นำมาใช้นี้คือโปรแกรม Capture-BAT, Netstatp และ WinPcap จากนั้นนำมาพัฒนาต่อยอดเพื่อให้สามารถทำงานเป็นระบบตรวจจับมัลแวร์เชิงพฤติกรรมได้
4) ปัญหาการเพิ่มฐานข้อมูลความรู้พฤติกรรมของมัลแวร์ เนื่องจากไม่สามารถรับรู้ถึงพฤติกรรมทั้งหมดของมัลแวร์ได้
   แนวทางการแก้ไขปัญหา : ทดลองใช้และเรียนรู้ถึงโครงสร้างของระบบปฏิบัติการวินโดวส์ในเชิงลึก เพื่อให้เข้าใจการทำงานและส่วนประกอบต่างๆที่จะทำให้มัลแวร์สามารถนำมาใช้ประโยชน์ได้ โดยในระบบตรวจจับนี้จะใช้เทคนิคที่เรียกว่า "การตรวจจับจากการใช้งานที่ผิดปกติ"
5) เมื่อรันโปรแกรมขึ้นมา โปรแกรมจะใช้ CPU ประมาณ 20%
   แนวทางการแก้ไขปัญหา : ควรจะเขียนโค้ดโปรแกรมให้มีความ Optimize มากกว่านี้

5.3 ข้อเสนอแนะและแนวทางการพัฒนาต่อ

1) เพิ่มฟังก์ชันการอัปเดตฐานข้อมูลความรู้พฤติกรรมมัลแวร์
2) ทำให้โปรแกรมสามารถปกป้องตัวเองจากมัลแวร์ได้
3) ทำให้การส่งผ่านข้อมูลระหว่างตัวโปรแกรมและไดรเวอร์มีความปลอดภัยมากยิ่งขึ้น โดยการเข้ารหัสลับ

บรรณานุกรม

Mark E. Russinovich and David A. Solomon. 2005. **Microsoft Windows Internals.** 4th ed. Washington : Microsoft Press.

Walter Oney. 2003. **Programming the Microsoft Windows Driver Model.** 2nd ed. Washington : Microsoft Press.

Penny Orwick and Guy Smith. 2007. **Developing Drivers with the Windows Driver Foundation.** Washington : Microsoft Press.

จตุชัย แพงจันทร์. 2550. **Master in Security.** นนทบุรี : ไอดีซีฯ.

Qi DeYu, Hu JingLin and Zhang Fu Yong. 2009. **"MBMAS:A System for Malware Behavior Monitor and Analysis."** Institute of Computer Systems South China University of Technology GuangZhou, China.

Martin Apel, Chridtian Bockermann and Michael Meier. 2009. **"Measuring Similarity of Malware Behavior."** University of Dortmund, Germany.

Rebecca Cathey, Ling Ma, Nazli Goharian and David Grossman. 2003. **"Misuse Detection for Information Retrieval Systems."** Department of Computer Science Illinois Institute of Technology, Chicago.

One Microsoft Way Redmond. Volume 9 January through June 2010. **Microsoft | Security Intelligence Report.** Washington : Microsoft Press.

Wikipedia. 2010. **Ring (computer security).** [Online]. Available : http://en.wikipedia.org/wiki/Ring_(computer_security)

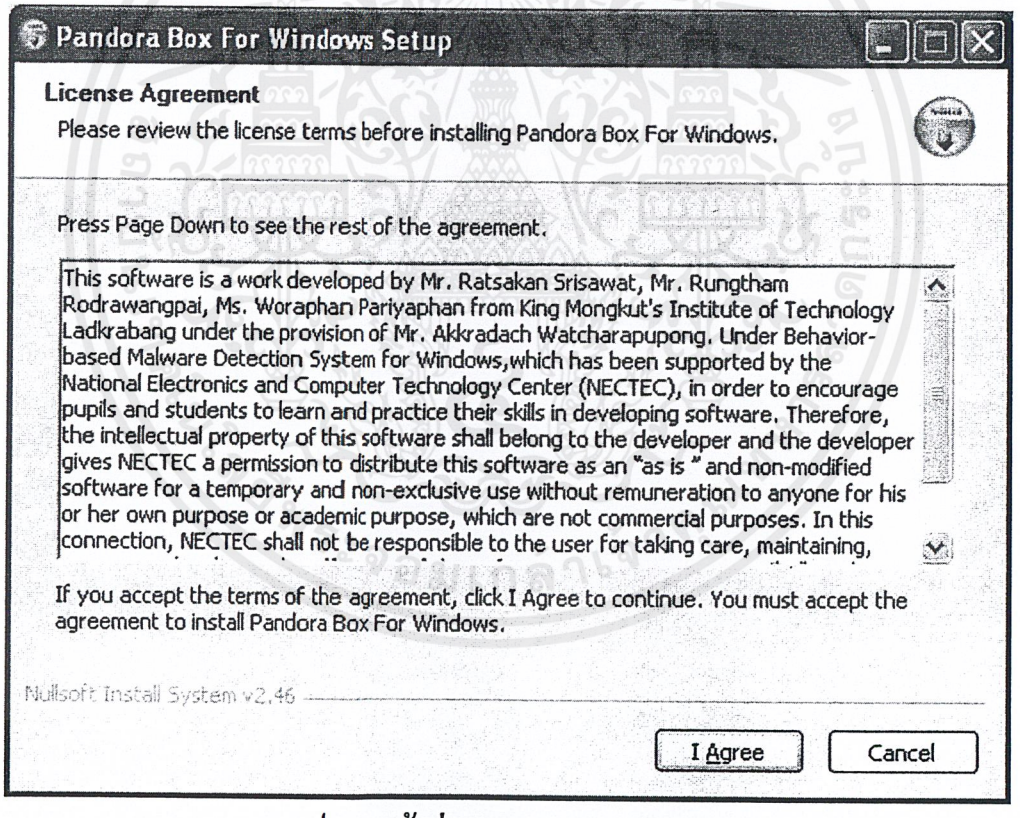
ภาคผนวก ก

คู่มือการติดตั้งโปรแกรมระบบตรวจจับมัลแวร์เชิงพฤติกรรม

1) เริ่มต้นด้วยการดับเบิ้ลคลิกที่ไอคอน Pandorabox_win32.exe

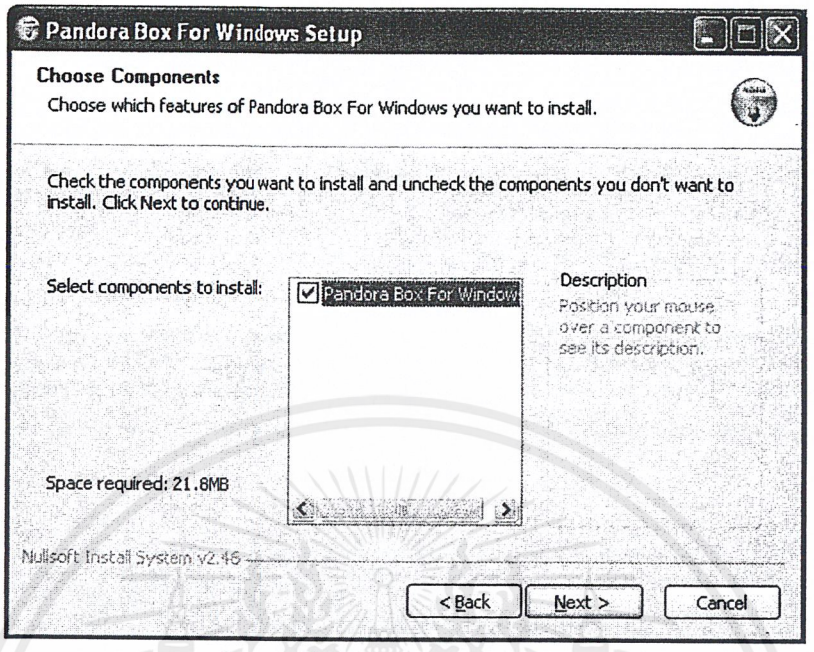
รูป ก.1 ไอคอน Pandorabox_win32.exe

2) เมื่อทำการดับเบิ้ลคลิกที่ไอคอนแล้วจะปรากฏหน้าจอดังรูป



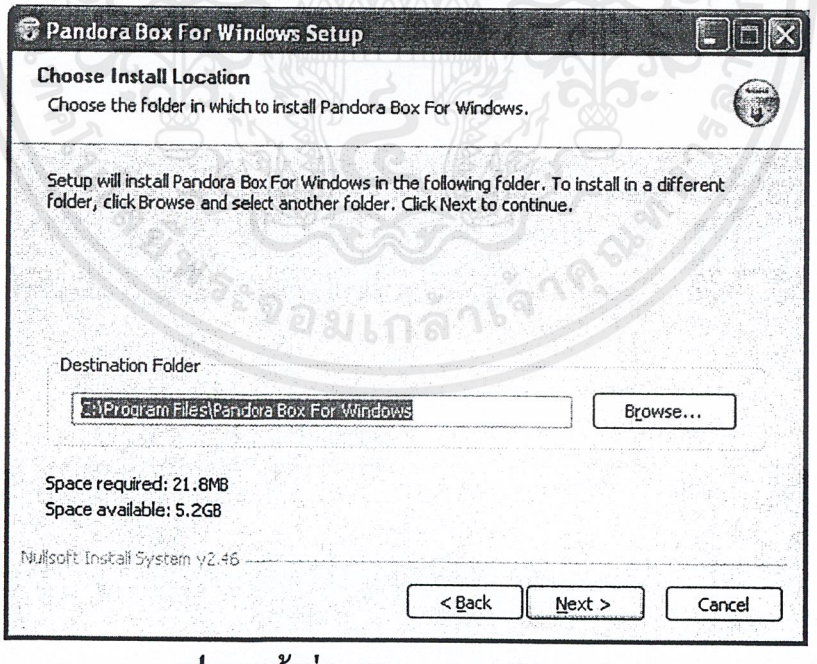
รูป ก.2 หน้าต่าง License Agreement

3) คลิกเครื่องหมายถูกเลือกชุดโปรแกรมที่ต้องการติดตั้ง และกดปุ่ม Next เพื่อไปยังขั้นตอนต่อไป



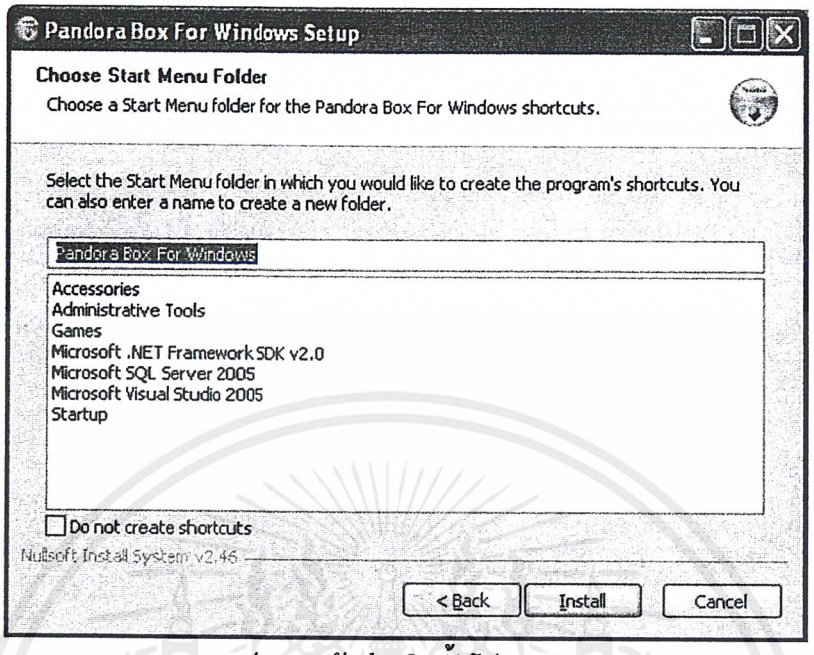
รูป ก.3 หน้าต่าง Choose Component

4) คลิกปุ่ม Next เพื่อไปยังขั้นตอนต่อไป



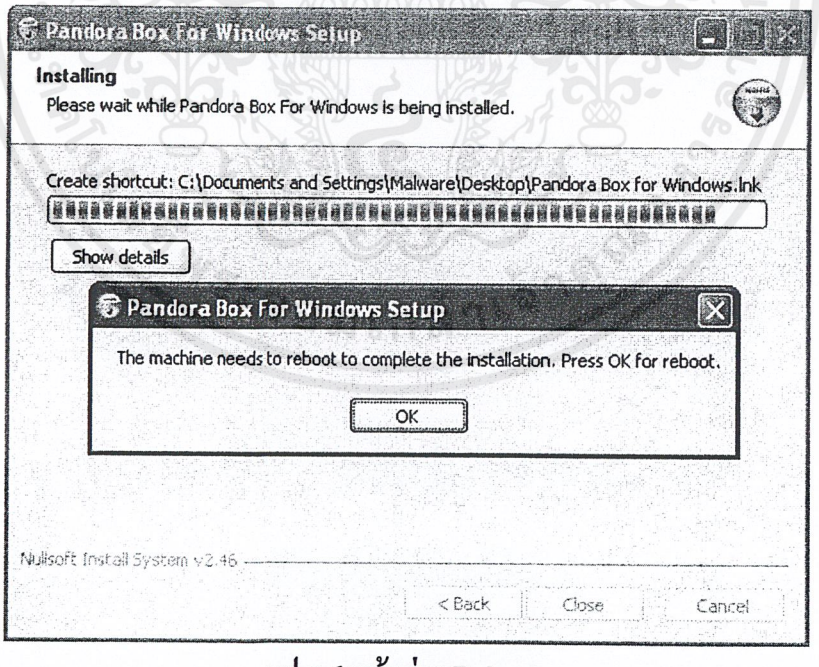
รูป ก.4 หน้าต่าง Choose Install Location

5) คลิกปุ่ม Install เพื่อติดตั้งโปรแกรม



รูป ก.5 หน้าต่างติดตั้งโปรแกรม

6) หลังจากติดตั้งเสร็จจะปรากฏกล่องข้อความ คลิกปุ่ม OK เพื่อรีสตาร์ทเครื่อง



รูป ก.6 หน้าต่าง Reboot

7) เมื่อทำการติดตั้งโปรแกรมเรียบร้อยแล้วจะปรากฏไอคอนโปรแกรมขึ้นมาดังรูป

รูป ก.7 ไอคอนโปรแกรม Pandora Box for Windows

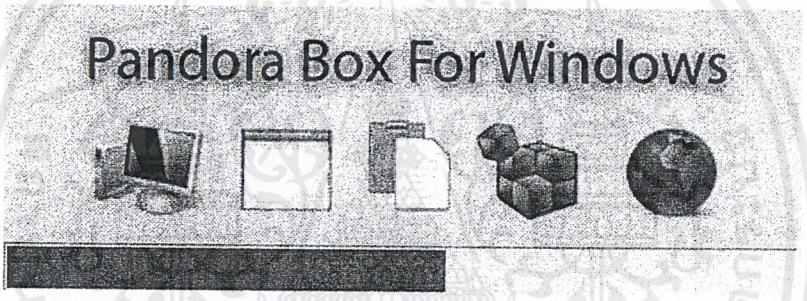
ภาคผนวก ข

คู่มือการใช้งานโปรแกรมระบบตรวจจับมัลแวร์เชิงพฤติกรรม

1) ทำการดับเบิ้ลคลิกที่ไอคอนโปรแกรม Pandora Box for Windows

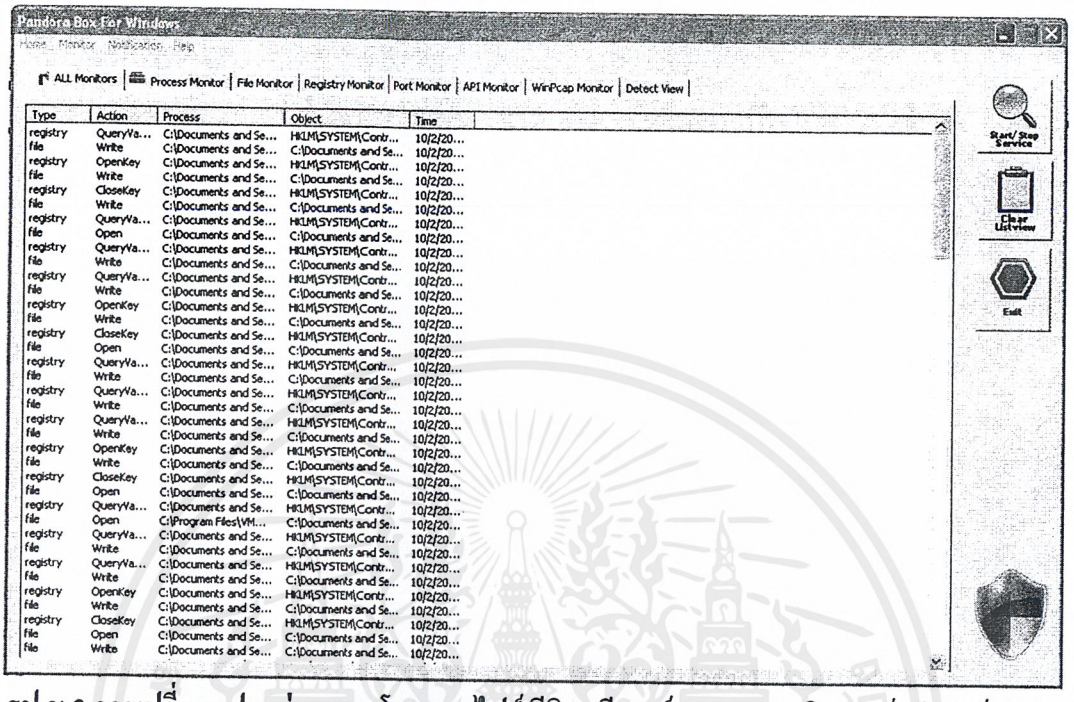
รูป ข.1 ไอคอนโปรแกรม Pandora Box for Windows

2) เมื่อดับเบิ้ลคลิกโปรแกรมเพื่อเปิดการทำงาน จะแสดงหน้าต่างโหลดโปรแกรมขึ้นมา ดังรูป ข.2



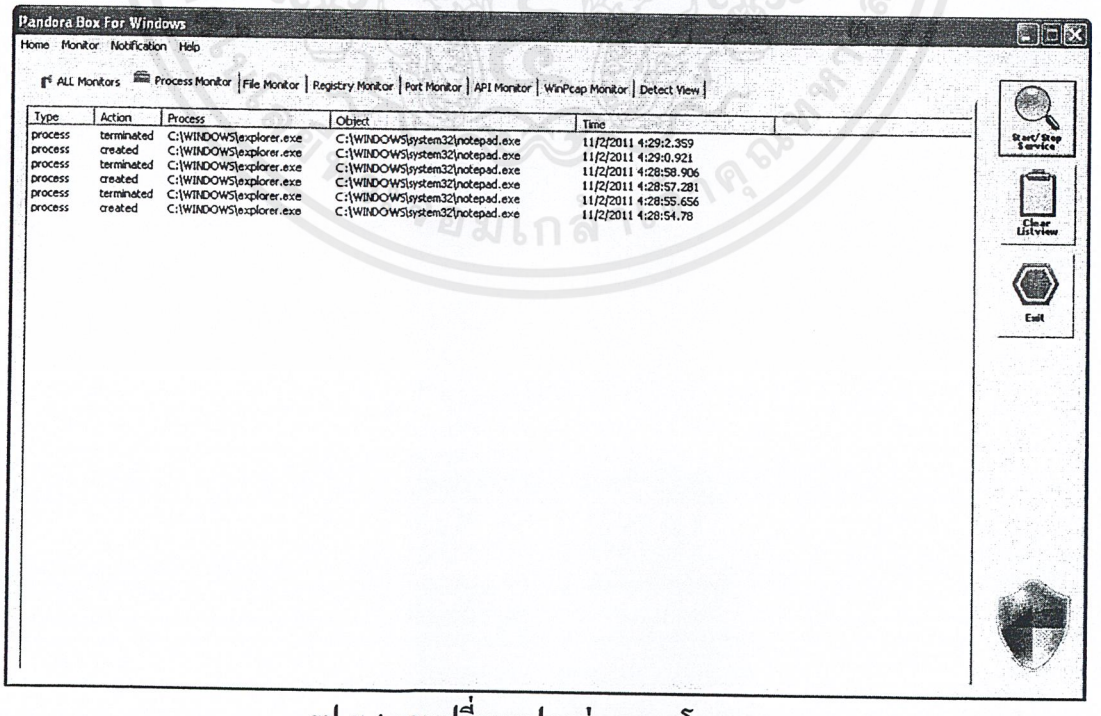
รูป ข.2 หน้าต่างโหลดโปรแกรม

3) เมื่อโหลดโปรแกรมเสร็จจะได้หน้าต่างแสดงการเปลี่ยนแปลงต่างๆของโพรเซส ไฟล์ รีจิสทรี การเปิดพอร์ต API และอินเทอร์เฟซการ์ดแลน ดังรูป ข.3



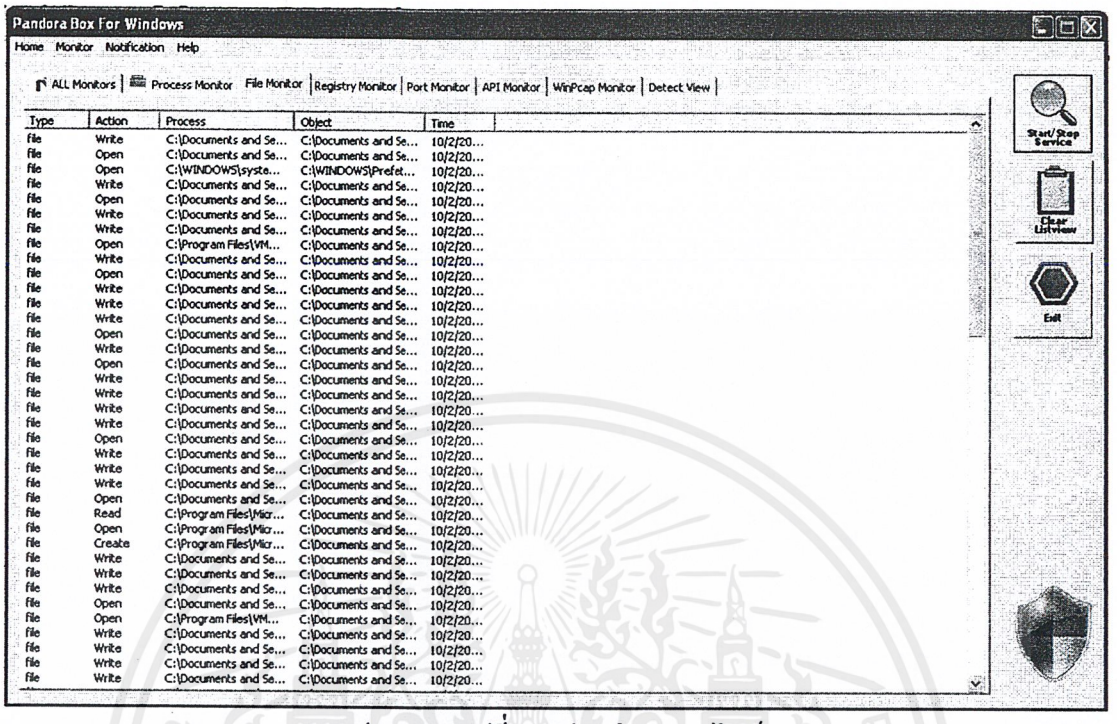
รูป ข.3 การเปลี่ยนแปลงต่างๆของโพรเซส ไฟล์ รีจิสทรี พอร์ต API และอินเทอร์เฟซการ์ดแลน

4) คลิกที่แท็บ Process Monitor เมื่อต้องการแสดงการเปลี่ยนแปลงต่างๆของโพรเซส ดังรูป ข.4



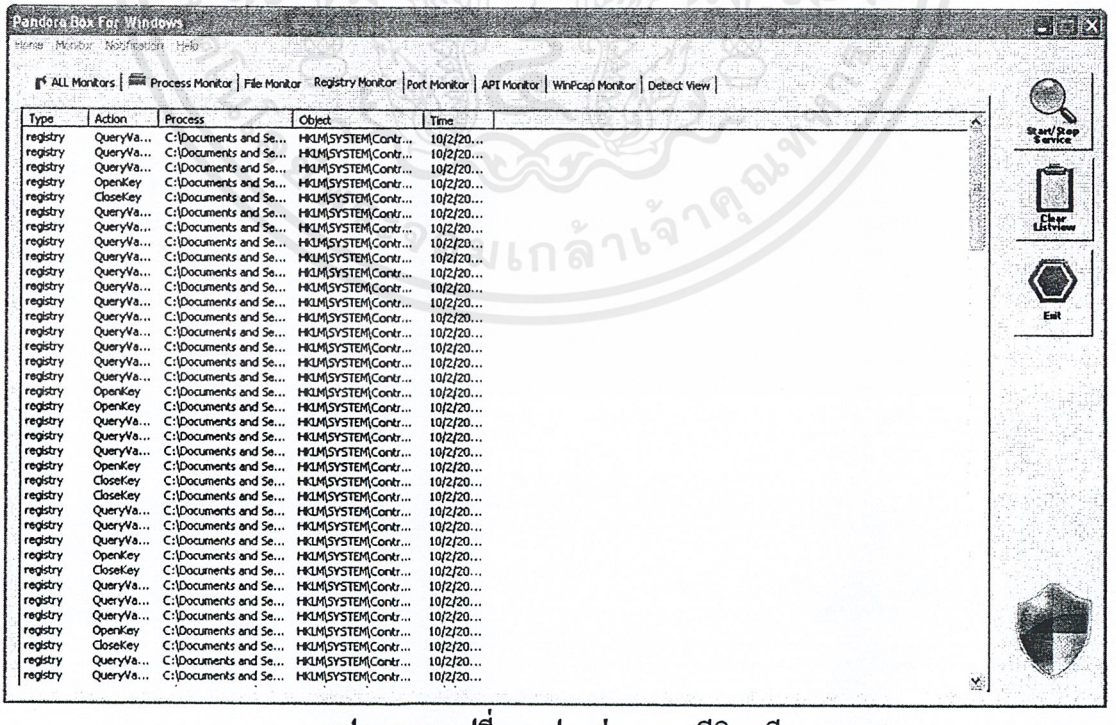
รูป ข.4 การเปลี่ยนแปลงต่างๆของโพรเซส

5) คลิกที่แท็บ File Monitor เมื่อต้องการแสดงการเปลี่ยนแปลงต่างๆของไฟล์ ดังรูป ข.5



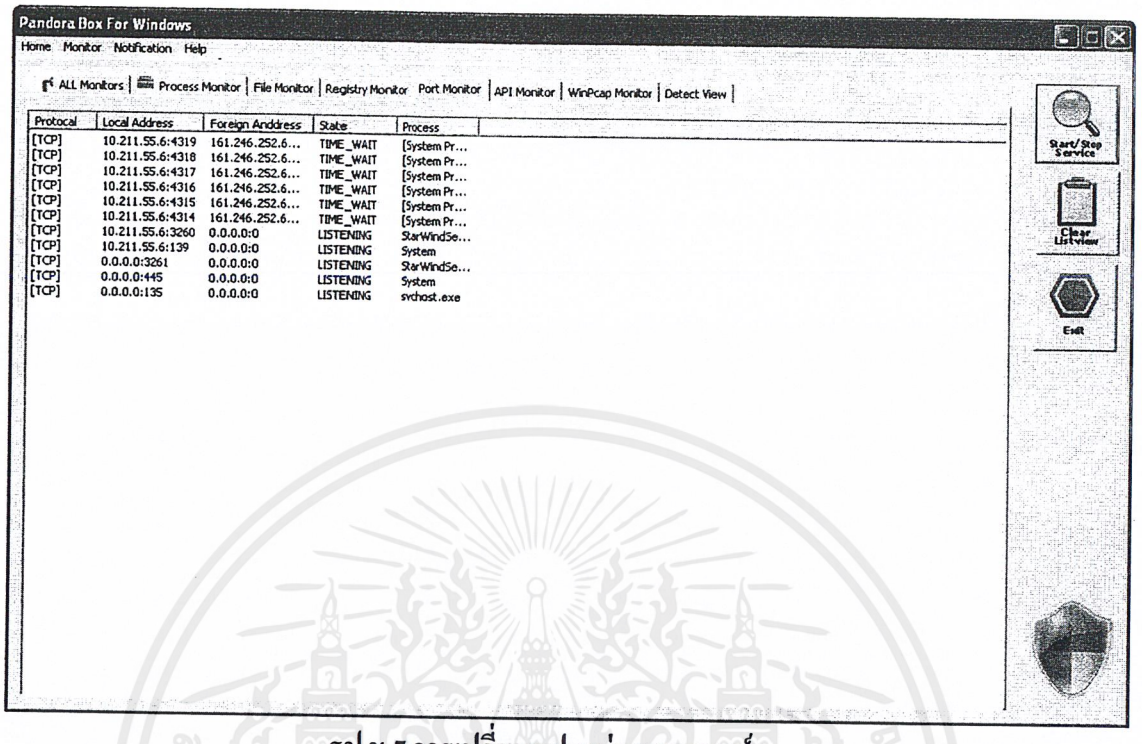
รูป ข.5 การเปลี่ยนแปลงต่างๆของไฟล์

6) คลิกที่แท็บ Registry Monitor เมื่อต้องการแสดงการเปลี่ยนแปลงต่างๆของรีจิสทรี ดังรูป ข.6



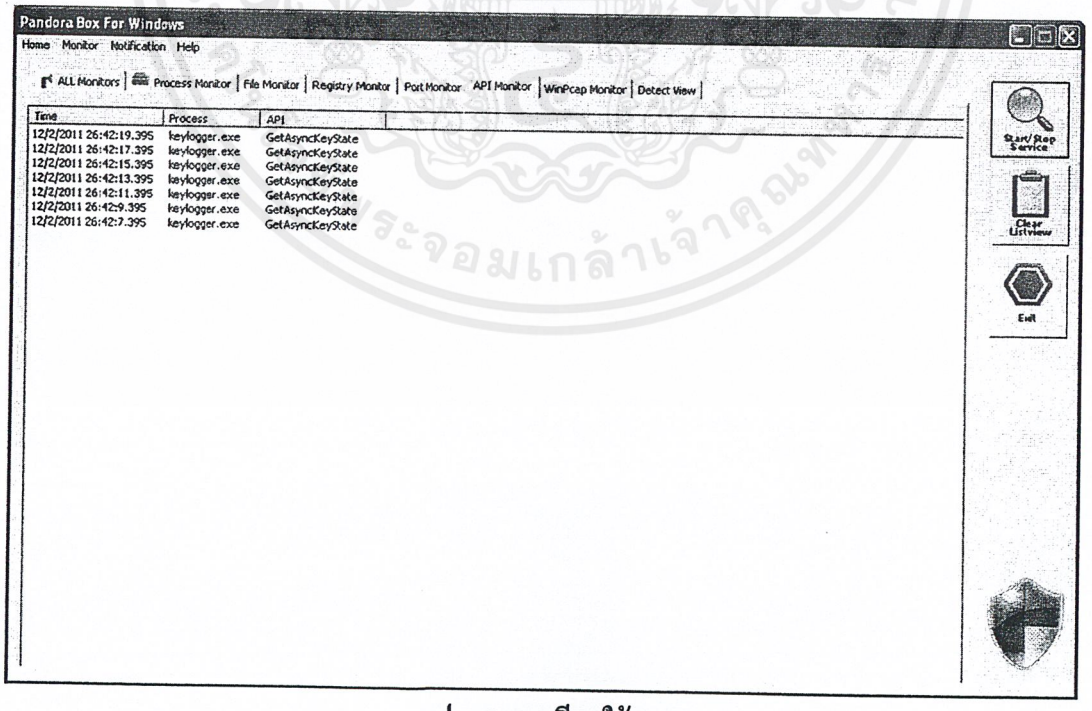
รูป ข.6 การเปลี่ยนแปลงต่างๆของรีจิสทรี

7) คลิกที่แท็บ Port Monitor เมื่อต้องการแสดงการเปิดพอร์ต ดังรูป ข.7



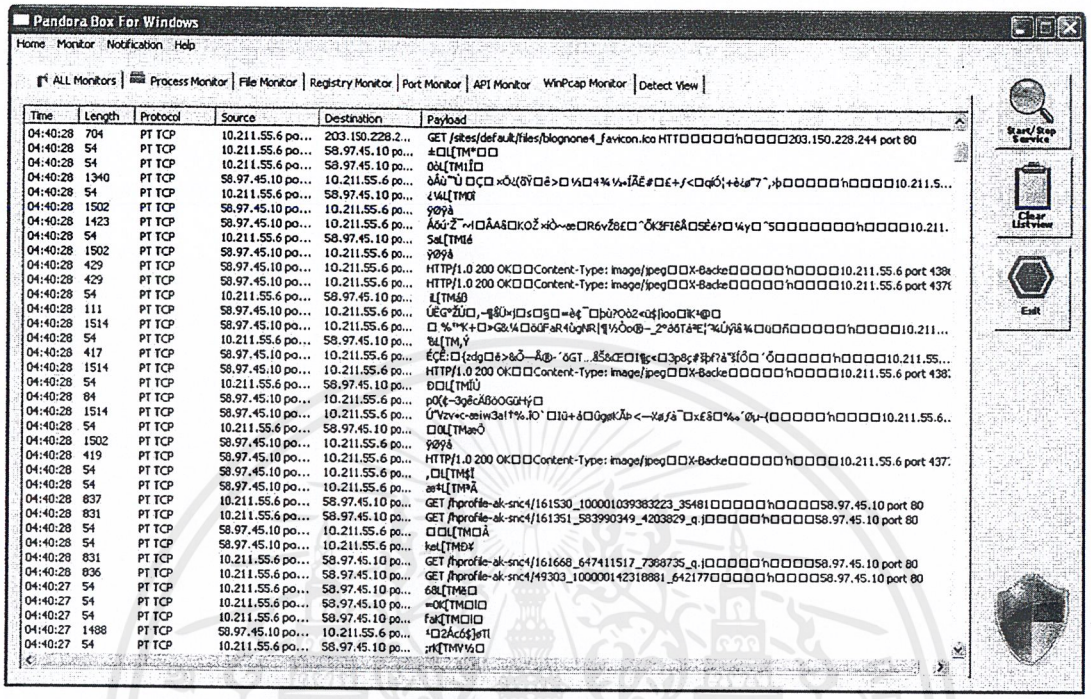
รูป ข.7 การเปลี่ยนแปลงต่างๆของพอร์ต

8) คลิกที่แท็บ API Monitor เมื่อต้องการแสดงการเรียกใช้ API ดังรูป ข.8



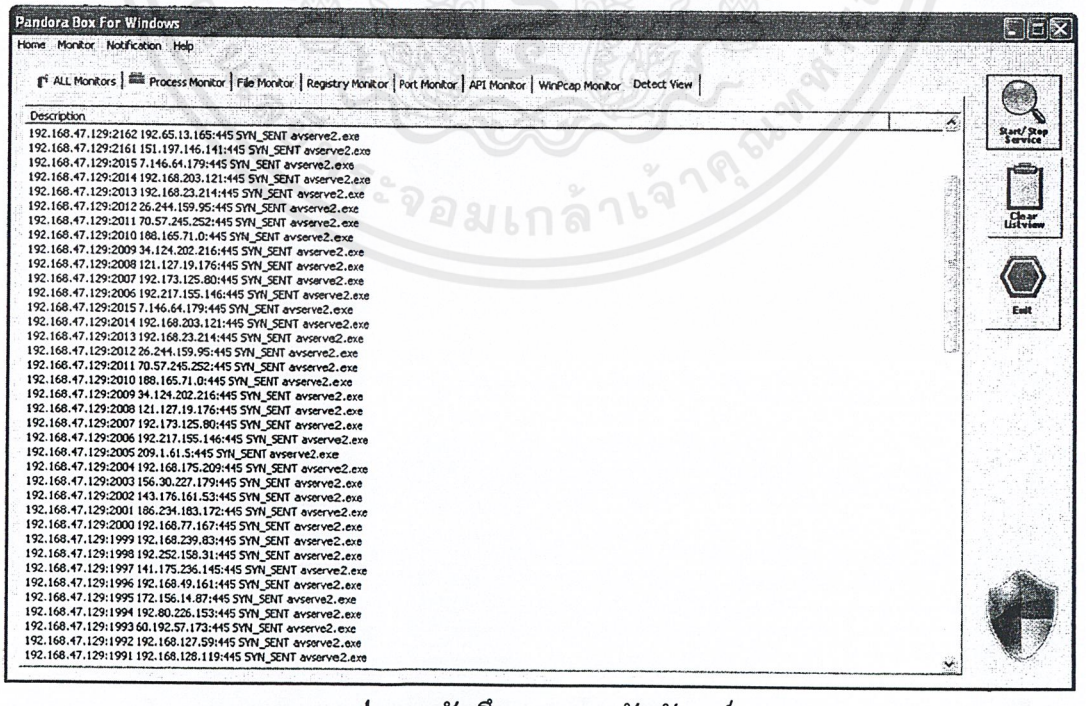
รูป ข.8 การเรียกใช้ API

9) คลิกที่แท็บ WinPcap Monitor เมื่อต้องการแสดงการเปลี่ยนแปลงต่างๆของอินเทอร์เฟซการ์ดแลน ดังรูป ข.9



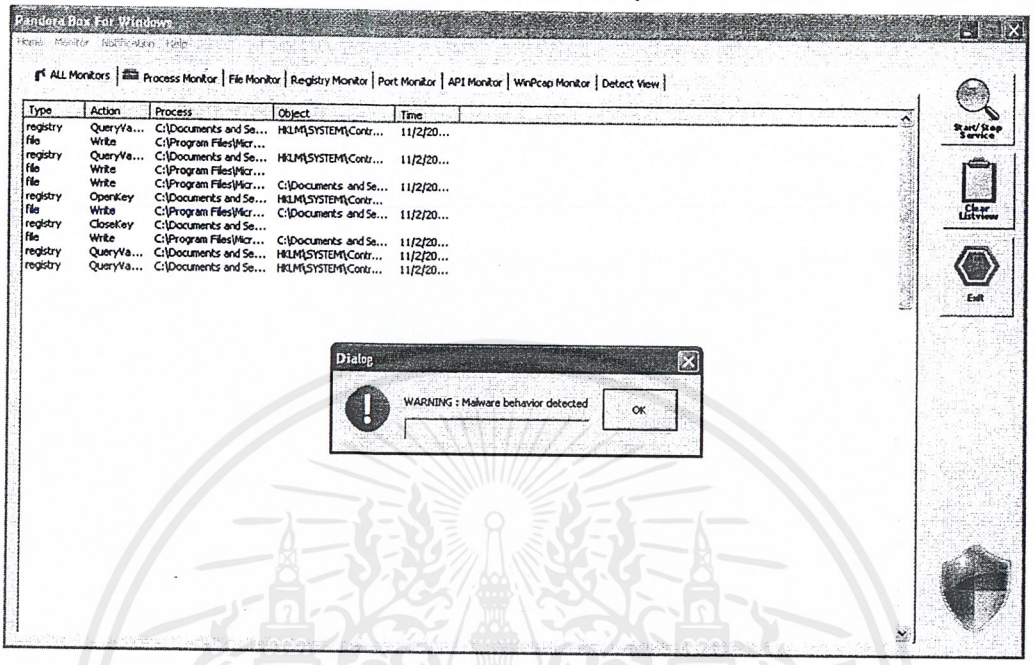
รูป ข.9 การรับส่งข้อมูลผ่านอินเทอร์เฟซการ์ดแลน

10) คลิกที่แท็บ Detect View เมื่อต้องการแสดงบันทึกการตรวจจับมัลแวร์ ดังรูป ข.10



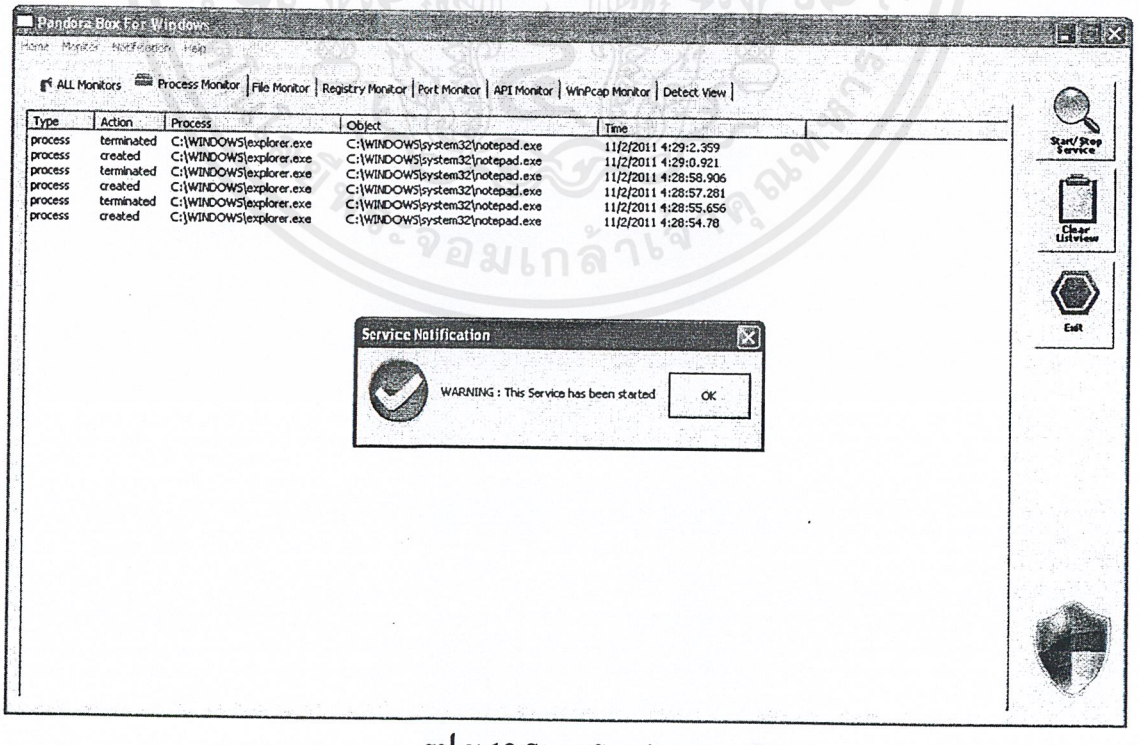
รูป ข.10 บันทึกการตรวจจับมัลแวร์

11) เมื่อมีการตรวจพบมัลแวร์ จะมีการแจ้งเตือนแก่ผู้ใช้งานดังรูป ข.8 ซึ่งจะบอกชื่อมัลแวร์ที่ตรวจพบด้วย



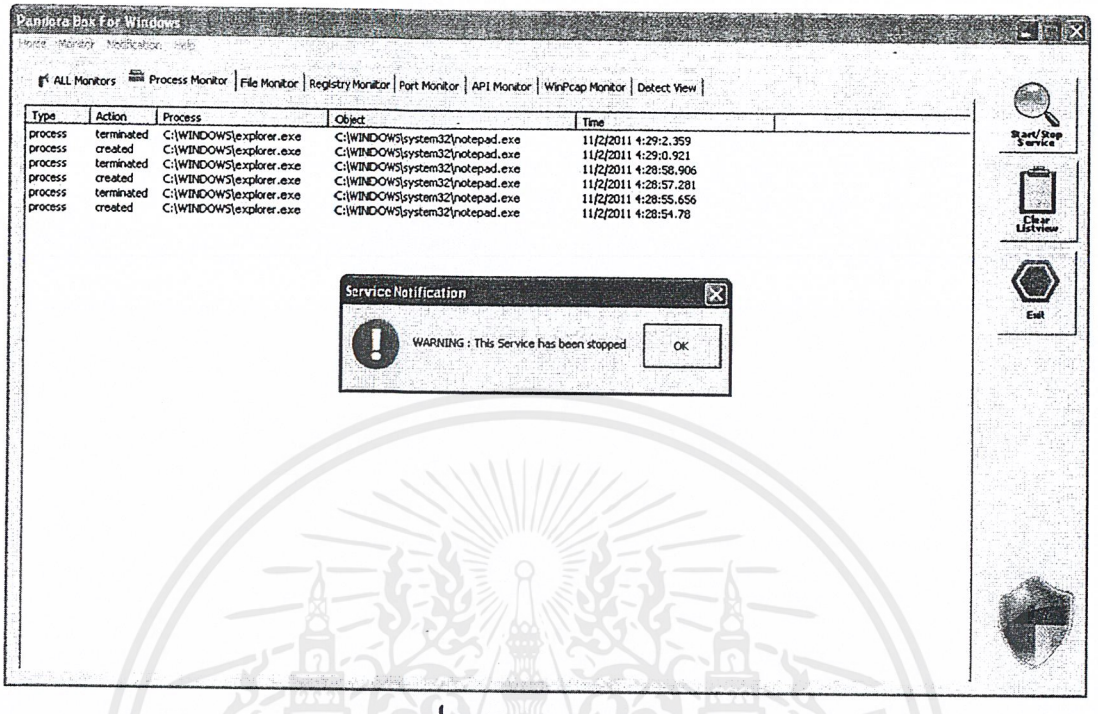
รูป ข.11 กล่องข้อความตรวจพบมัลแวร์

12) คลิกที่ปุ่ม Start Service เมื่อต้องการเริ่มการแสดงสถานะ ดังรูป ข.12



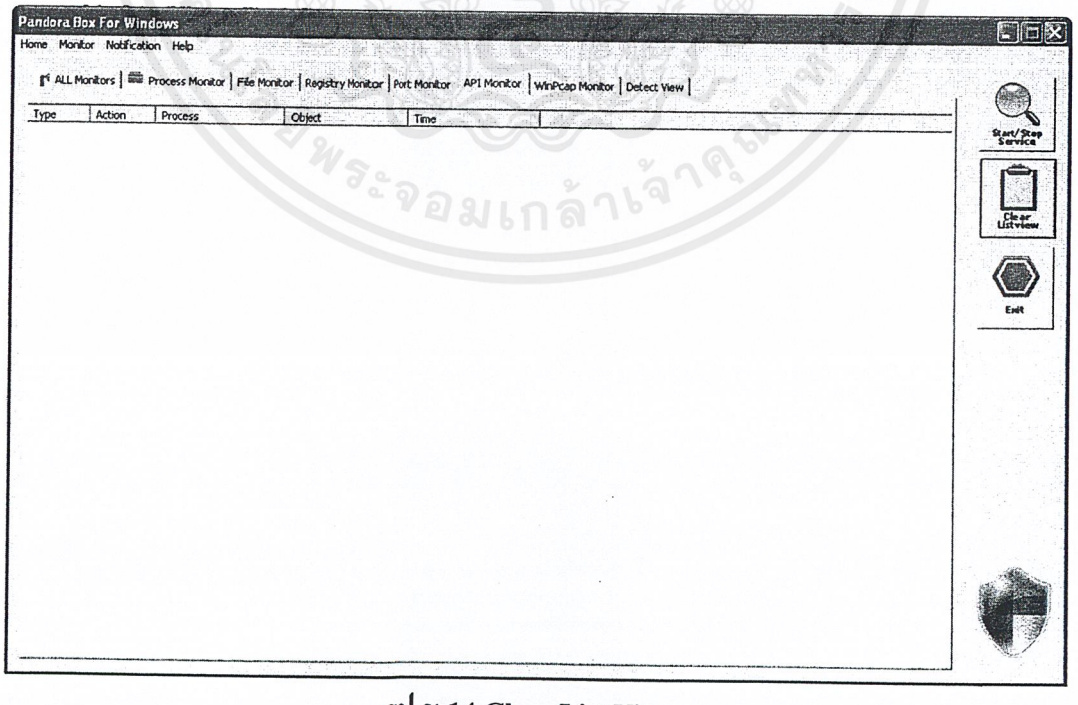
รูป ข.12 Start Service

13) คลิกที่ปุ่ม Stop Service เมื่อต้องการหยุดการแสดงสถานะ ดังรูป ข.13



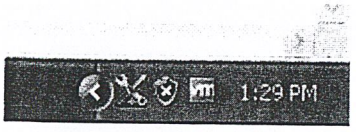
รูป ข.13 Stop Service

14) คลิกที่ปุ่ม Clear List View เมื่อต้องการทำหน้าจอโปรแกรมให้ว่างเพื่อเริ่มการแสดงสถานะใหม่ ดังรูป ข.14



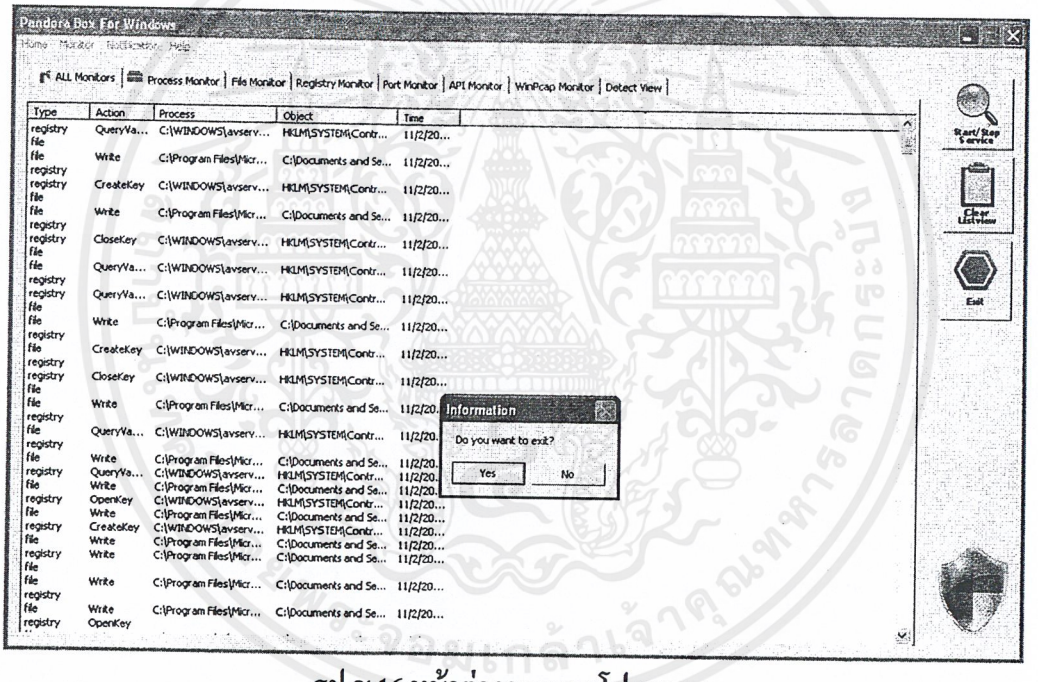
รูป ข.14 Clear List View

15) เมื่อคลิกที่ปุ่ม Minimize หน้าจอโปรแกรมจะถูกย่อลงกลายเป็น System Tray และจะยังคงการทำงานอยู่ ดังรูป ข.15



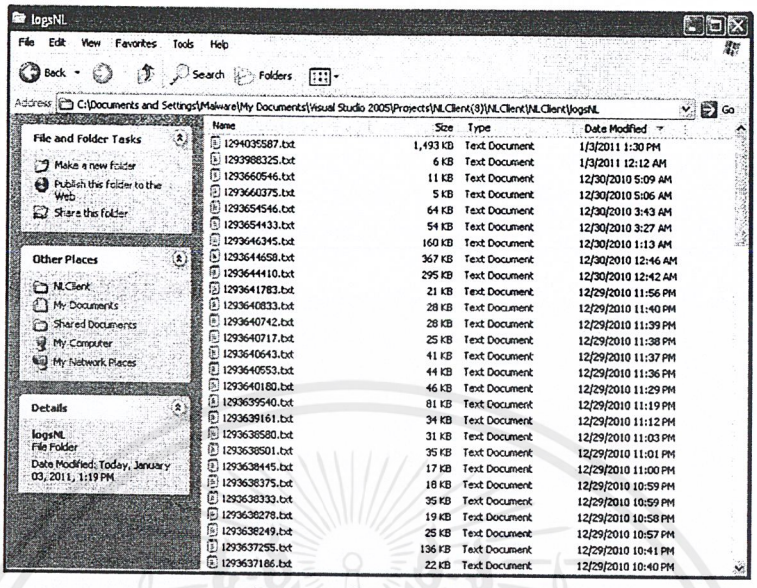
รูป ข.15 System Tray

16) เมื่อคลิกที่กากบาทปิดโปรแกรมหรือปุ่ม Exit จะมีหน้าต่างถามว่าต้องการจะออกจากโปรแกรมหรือไม่ คลิก Yes เมื่อต้องการปิดโปรแกรม คลิก No เมื่อไม่ต้องการออกจากโปรแกรม ดังรูป ข.16



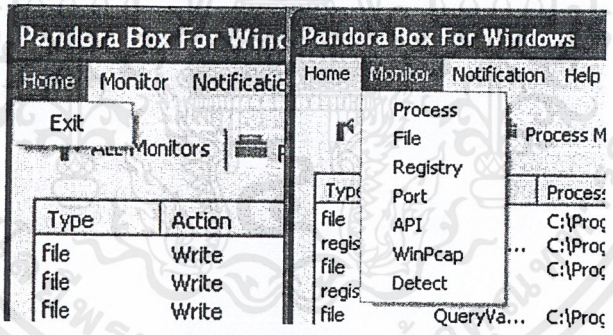
รูป ข.16 หน้าต่างออกจากโปรแกรม

17) หน้าต่าง Log File ดังรูป ข.17



รูป ข.17 หน้าต่าง Log File

18) แถบ Menu Bar ดังรูป ข.18



รูป ข.18 แถบ Menu Bar

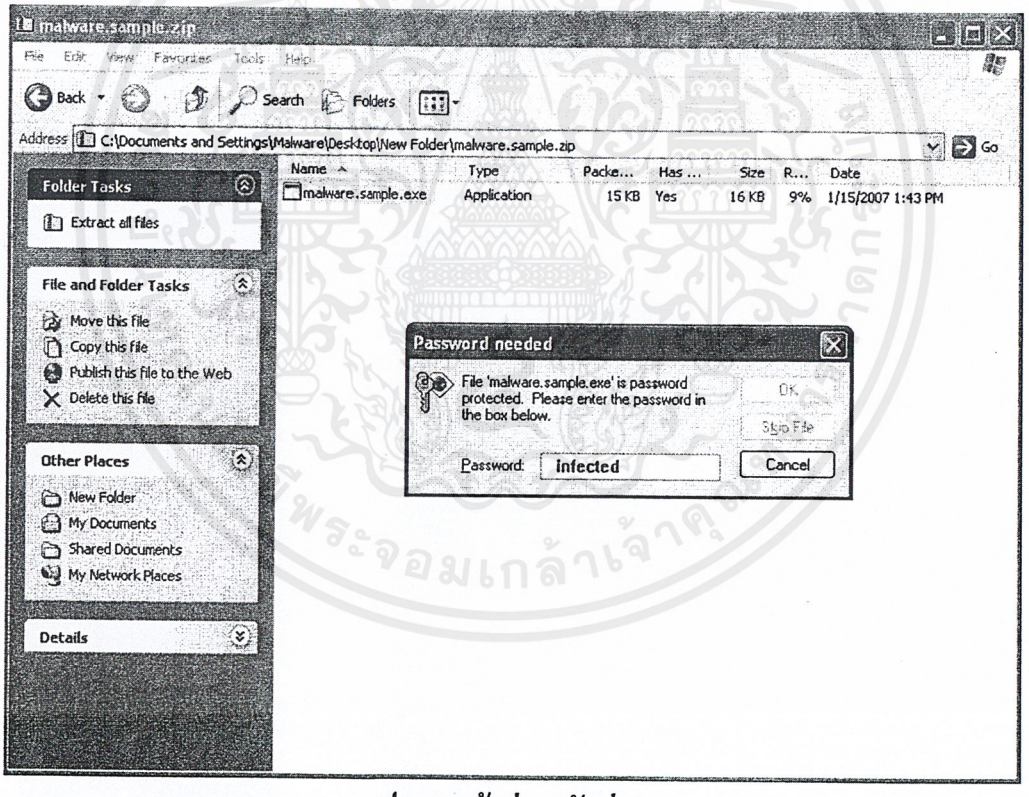
ภาคผนวก ค

คู่มือการทดสอบโปรแกรม

1) เริ่มต้นด้วยการดับเบิ้ลคลิกที่ไอคอน malware.sample.zip

รูป ค.1 ไอคอน malware.sample.zip

2) ดับเบิ้ลคลิกที่ไฟล์ malware.sample.exe จากนั้นโปรแกรมจะถามรหัสผ่าน ให้กรอกรหัสผ่านคำว่า infected เพื่อรันตัวอย่างมัลแวร์



รูป ค.2 หน้าต่างรหัสผ่าน