การตรวจจับระบบที่ถูกละเมิดความปลอดภัย

COMPROMISED SYSTEM DETECTION

ฐิติพร ทองเจริญประเสริฐ

ธนุสั้กทร อินทร์สุวรรณโณ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

การตรวจจับระบบที่ถูกละเมิดความปลอดภัย

Compromised System Detection

ฐิติพร จงเจริญประเสริฐ
อนุภัทร อินทร์สุวรรณโณ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

ปริญญานิพนธ์ปีการศึกษา 2556

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การตรวจจับระบบที่ถูกละเมิดความปลอดภัย

Compromised System Detection

ผู้จัดทำ

1. นายฐิติพร	จงเจริญประเสริฐ	รหัสนักศึกษา	53010412
2. นายอนุภัทร	อินทร์สุวรรณโณ	รหัสนักศึกษา	53011846

..... อาจารย์ที่ปรึกษา

(อ.อัครเดช วัชระภูพงษ์)

การตรวจจับระบบที่ถูกละเมิดความปลอดภัย

นาย ฐิติพร	จงเจริญประเสริฐ	53010412
นาย อนุภัทร	อินทร์สุวรรณโณ	53011846
อ. อัครเดช	วัชระภูพงษ์	อาจารย์ที่ปรึกษา

ปีการศึกษา 2556

บทคัดย่อ

ปัจจุบันมีการละเมิดระบบรักษาความปลอดภัยอย่างกว้างขวาง โดยมีลักษณะทั้งเพื่อก่อกวนและเพื่อหวังผลประโยชน์ อาศัยการล่อหลอกและเทคนิคที่แปรเปลี่ยนไปไม่ตายตัว อีกทั้งดำเนินกิจกรรมด้วยกลุ่มคนผสมกับเครื่องมืออัตโนมัติต่างๆ ไฟร์วอลหรือโปรแกรมตรวจจับไวรัสคอมพิวเตอร์และมัลแวร์จึงไม่เพียงพอต่อกร

โครงงานนี้มุ่งเน้นการตรวจจับระบบซึ่งถูกละเมิดความปลอดภัยไปแล้ว เพื่อคัดแยกออกจากระบบที่ยังสะอาดหรือยังไม่ถูกละเมิด รวมถึงวินิจฉัยกลวิธีที่ใช้ละเมิดอีกด้วย โดยจัดทำระบบต้นแบบในภาวะจำลองเพื่อศึกษา พัฒนา และทดลองตามมาตรการและเทคนิคที่เหมาะสมสำหรับแต่ละสภาพแวดล้อม อาทิ คอมพิวเตอร์ส่วนบุคคล เครื่องแม่ข่าย อุปกรณ์ระบบเครือข่าย ระบบปฏิบัติการ แพลตฟอร์มโปรแกรมประยุกต์ และแม้กระทั่งอุปกรณ์รักษาความปลอดภัยเอง

Compromised System Detection

Mr. Titiporn	Chongcharoenprasert	53010412
Mr. Anupat	Insuvanno	53011846
Mr. Akkradach	Watcharapupong	Advisor

Academic Year 2013

ABSTRACT

There's wide spread of security breaches, either for disturbing or for profit. Relies on social engineering and other techniques indefinitely. Because of the mix of activities with a group of people with various automation tools, firewall and Anti-virus/malware can not be sufficient.

This project focuses on detection of the system, which is already compromised. To separate out the cleaning has not been violated. And to diagnose the techniques used to abuse. The prototype system will be developed in some simulation environments to study and test measures and techniques suitable for each environments, such as personal computing systems, computer servers, networking equipments, operating systems, application platforms, and even security devices themself.

กิตติกรรมประกาศ

โครงงานการตรวจจับระบบที่ถูกละเมิดความปลอดภัย สามารถสำเร็จสมบูรณ์และลุล่วงไปได้ด้วยดีด้วยคำปรึกษาการพัฒนาโครงงานและการวางแผนการดำเนินการจาก อ. อัครเดช วัชระภูพงษ์ ซึ่งเป็นอาจารย์ที่ปรึกษาของโครงงานนี้ที่ให้การดูแล ให้คำแนะนำ และมีการติดตามความคืบหน้าของโครงงานอย่างสม่ำเสมอ

ขอบพระคุณอาจารย์ทุกท่านในสาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่ได้กรุณาช่วยวิจารณ์โครงงานเพื่อนำไปปรับปรุงให้ดียิ่งขึ้น

ขอขอบคุณรุ่นพี่ทุกท่าน รวมถึงเพื่อนๆที่คอยให้คำแนะนำการทำโครงงานในด้านต่างๆ

ขอบคุณห้องวิจัย Information Security Advisory Group ที่เป็นแหล่งสนับสนุนสถานที่และอุปกรณ์ในการพัฒนาโครงงานได้อย่างสะดวก

สุดท้ายนี้ขอกราบขอบพระคุณบิดามารดา ที่ให้กำลังใจและสนับสนุนในทุก ๆ ด้าน ด้วยคุณค่าและประโยชน์อันพึงมาจากโครงงานนี้ เราขอมอบแด่ผู้มีพระคุณทุกท่าน

นาย ฐิติพร จงเจริญประเสริฐ

นาย อนุภัทร อินทร์สุวรรณโณ

สารบัญ

หน้า

สารบัญตาราง

ตารางที่ **หน้า**

สารบัญรูป

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของโครงงาน

การตรวจจับระบบที่ถูกละเมิดความปลอดภัยเป็นการศึกษาวิธีการป้องกันจากการถูกคุกคามหรือบุกรุกจากผู้ไม่ประสงค์ดีในโลกคอมพิวเตอร์ ซึ่งมีปรากฏขึ้นอยู่ทั่วไปทุกหนทุกแห่ง ทั้งในระดับผู้ใช้งานตามบ้านเรือน หรือแม้กระทั่งในระดับองค์กรขนาดใหญ่ โดยผู้ใช้ส่วนมากไม่รู้ด้วยซ้ำว่าเครื่องคอมพิวเตอร์ที่ตนเองใช้งานอยู่นั้นถูกคุกคามหรือถูกละเมิดความปลอดภัย และอาจนำมาซึ่งความเสียหายในด้านต่างๆทั้งชีวิตและทรัพย์สิน จึงจำเป็นต้องมีระบบที่ทำหน้าที่ตรวจจับการคุกคามในรูปแบบต่างๆ เพื่อลดความเสี่ยงจากการคุกคามที่อาจเกิดขึ้น โดยโครงงานการตรวจจับระบบที่ถูกละเมิดความปลอดภัยนี้ ถูกสร้างขึ้นเพราะความต้องการในการระบุตำแหน่งที่มัลแวร์ฝังตัวอยู่ อีกทั้งไฟร์วอลล์หรือโปรแกรมตรวจจับไวรัส ไม่เพียงพอต่อการต่อกร รวมถึงการคัดแยกระบบที่ถูกและไม่ถูกละเมิดความปลอดภัยนั้นทำได้ยาก

1.2 วัตถุประสงค์ของโครงงาน

1) เพื่อให้ผู้ใช้งานหรือผู้ดูแลระบบ สามารถคัดแยกระบบที่ถูกและไม่ถูกละเมิดความปลอดภัยออกจากกัน
2) เพื่อแสดงตำแหน่งที่มีโอกาสที่มัลแวร์จะฝังตัวอยู่ในเครื่องที่ถูกละเมิด

1.3 ขอบเขตของโครงงาน

1) เป็นโปรแกรมประเภท Scan Once ไม่ใช่โปรแกรมประเภท Always On
2) ซอฟต์แวร์สามารถทำงานได้บนระบบปฏิบัติการไมโครซอฟต์วินโดว์ 7 (Microsoft Windows 7) และไมโครซอฟต์วินโดวส์ 8 (Microsoft Windows 8)
3) โปรแกรมไม่ได้ทำงานในเรื่องของ prevention ทำแค่เพียงการ detection เท่านั้น
4) ต้องติดตั้ง Visual C++ Redistributable for Visual Studio 2012 Update 4
5) มีการเชื่อมต่ออินเทอร์เน็ตขณะเรียกใช้งาน
6) ตั้งค่าให้เรียกใช้ Google Chrome เป็นเบราวเซอร์เริ่มต้น เพื่อการแสดงผลที่สมบูรณ์แบบ

1.4 วิธีการดำเนินการ

เพื่อให้โครงงานนี้ดำเนินไปอย่างราบรื่น จึงได้วางแผนการทำงานล่วงหน้าโดยกำหนดช่วงวันและหัวข้อในรูปแบบของ Gantt Chart ดังตารางที่ 1.1 และตารางที่ 1.2 ตามลำดับ

ตารางที่ 1.1 วิธีดำเนินการในภาคเรียนที่ 1

หัวข้อย่อย	มิถุนายน				กรกฎาคม				สิงหาคม				กันยายน			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1. ศึกษาวิธีการทำงานของโปรแกรม Hijack this																
2. วิเคราะห์และศึกษาเกี่ยวกับตำแหน่งต่างๆในระบบที่มีโอกาสที่มัลแวร์(Malware)จะฝังตัวอยู่																
3. คิดวิธีการวิเคราะห์เพื่อแยกมัลแวร์ออกจากโปรแกรมธรรมดา																
4. สร้างโปรแกรมต้นแบบเพื่อหา Best practice ในการพัฒนา และเพื่อที่จะพิสูจน์ว่าสามารถทำงานได้ผลลัพธ์จริง																
5. ทดลองโดยการทำให้ระบบติดเชื้อและทำการดักจับมัลแวร์																
6. ดีบักและปรับแก้โปรแกรม เพื่อหา Best practice ต่อไป																
7. ดีบักและปรับแก้โปรแกรมให้ต้านทานต่อฟอลซ เนกาทีฟ (False Negative)																

ตารางที่ 1.2 วิธีดำเนินการในภาคเรียนที่ 2

หัวข้อย่อย	พฤศจิกายน				ธันวาคม				มกราคม				กุมภาพันธ์			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1. ออกแบบยูสเซอร์ อินเตอร์เฟส (User interface) และโครงสร้างของโปรแกรม																
2. พัฒนาโปรแกรมโดยใช้ Best practice และองค์ความรู้ที่สะสมมาจากภาคการเรียนที่ 1																
3. ดีบักและปรับแก้โปรแกรม																
4. คิดค้นตำแหน่งต่างๆที่มัลแวร์มีโอกาสฝังตัวอยู่เพิ่มเติม																
5. ทำรูปเล่มปริญญานิพนธ์																

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1) ลดจำนวนการคุกคามของโปรแกรมแปลกปลอมในรูปแบบต่างๆลงได้
2) ลดความเสียหายที่เกิดขึ้นจากการคุกคามของโปรแกรมแปลกปลอม
3) เพิ่มระดับความปลอดภัยให้กับเครื่องคอมพิวเตอร์ของผู้ใช้งาน จากการถูกละเมิดความปลอดภัยในด้านต่างๆ
4) ป้องกันการถูกโจรกรรมข้อมูล หรือทรัพย์สินต่างๆขององค์กร จากการถูกคุมคามเครื่องคอมพิวเตอร์ของบุคคลากรในองค์กร

1.6 ส่วนประกอบของปริญญานิพนธ์

เนื้อหาของปริญญานิพนธ์ฉบับนี้ประกอบด้วย 5 บท คือ บทนำ ทฤษฎีที่เกี่ยวข้อง การออกแบบและพัฒนาซอฟต์แวร์ การทดลองและผลการทดลอง และบทสรุป โดยสามารถจำแนกรายละเอียดได้ดังนี้

บทที่ 1 บทนำ กล่าวถึง ความสำคัญและที่มาของโครงงาน วัตถุประสงค์ของโครงงาน ขอบเขตของโครงงาน วิธีการดำเนินการ และ ประโยชน์ที่คาดว่าจะได้รับ

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง กล่าวถึง แกรบเบอร์ (Grabber) อนาไลซ์เซอร์(C#) (Analyzer (C#)) อนาไลซ์เซอร์(WEB) (Analyzer(WEB)) และข้อมูลทั่วไป

บทที่ 3 การออกแบบและพัฒนาซอฟต์แวร์ กล่าวถึง เครื่องมือที่ใช้ในการพัฒนา ภาพรวมของโปรแกรม และกระบวนการทำงานของโปรแกรม

บทที่ 4 การทดลองและผลการทดลอง กล่าวถึง กล่าวถึง วิธีการทดลองและผลลัพธ์หลังจากโปรแกรมเริ่มทำการตรวจจับ

บทที่ 5 บทสรุป กล่าวถึง ผลที่ได้จากการทำโครงงาน ปัญหาอุปสรรคและแนวทางการแก้ไขและแนวทางในการพัฒนาต่อ

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

การที่เราจะทำการตรวจจับระบบที่ถูกละเมิดความปลอดภัยนั้นเราจำเป็นต้องเข้าใจทฤษฎีต่าง ๆ เป็นอย่างดีเพื่อที่เราจะสามารถเข้าใจถึงการทำงานของโปรแกรมแปลกปลอมที่เข้ามาในระบบได้อย่างถูกต้อง ซึ่งทฤษฎีที่เกี่ยวข้องกับการตรวจจับระบบที่ถูกละเมิดความปลอดภัยนั้น แบ่งออกเป็น 4 ส่วนอันได้แก่ แกรบเบอร์ (Grabber) อนาไลซ์เซอร์(C#) (Analyzer (C#)) อนาไลซ์เซอร์(เวป) (Analyzer(WEB)) และข้อมูลทั่วไป

2.1 แกรบเบอร์ (Grabber)

แกรบเบอร์ คือ ส่วนของโปรแกรมที่ทำหน้าที่ค้นหาและดึงข้อมูลที่จำเป็นต้องใช้ในการวิเคราะห์ ซึ่งข้อมูลที่ว่าเหล่านั้นประกอบไปด้วย

2.1.1 32-64 บิต รีจิสทรีคีย์ (32-64 Bit Registry Key)

2.1.1.1 การเข้าถึงมุมมองของรีจิสทรีที่สลับสับเปลี่ยนกัน

โดยปกติแล้ว แอปพลิเคชันประเภท 32 บิตจะทำงานบน WOW64 ซึ่งสามารถเข้าถึงโดยมุมมองของรีจิสทรีแบบ 32 บิต ส่วนแอปพลิเคชันประเภท 64 บิตสามารถเข้าถึงโดยมุมมองของรีจิสทรีแบบ 64 บิตเช่นกัน โดยแฟลก(flag)ด้านล่างนี้สามารถให้แอปพลิเคชันแบบ 32 บิต สามารถเข้าถึงโดยผ่านคีย์ในรูปแบบของรีจิสทรี 64 บิต และเช่นกันสำหรับแอปพลิเคชันแบบ 64 บิต สามารถเข้าถึงโดยผ่านคีย์ในรูปแบบของรีจิสทรี 32 บิตได้ด้วย ซึ่งการใช้แฟลกดังกล่าวไม่มีผลกระทบใดๆต่อการแบ่งปันรีจิสทรีคีย์

ตารางที่ 2.1 แสดงรายละเอียดของแฟลกแต่ละตัว

ชื่อแฟลก	ค่า	อธิบาย
KEY_WOW64_64KEY	0x0100	เข้าถึงคีย์แบบ 64 บิตจากแอปพลิเคชันแบบ 32 และ 64 บิต
KEY_WOW64_32KEY	0x0200	เข้าถึงคีย์แบบ 32 บิตจากแอปพลิเคชันแบบ 32 และ 64 บิต

2.1.1.2 ข้อมูลของแอปพลิเคชันแบบ 32 บิต และ 64 บิตในรีจิสทรี

สำหรับวินโดวส์แบบ 64 บิตนั้น ในส่วนรายการของรีจิสทรีจะถูกเก็บไว้โดยแยกกันระหว่างแอปพลิเคชัน 32 บิต กับ แอปพลิเคชัน 64 บิต โดยจะถูกเชื่อมโยงกันในรูปแบบของโลจิคอลรีจิสทรี (logical registry) ที่แยกออกจากกันโดยใช้รีจิสทรีรีไดเร็คเตอร์ (registry redirector) และ รีจิสทรีรีเฟล็กซ์ชัน(registry reflection) และด้วยเหตุที่ว่าแอปพลิเคชันในเวอร์ชั่น 64 บิตนั้น อาจใช้รีจิสทรีคีย์และค่าที่แตกต่างจากเวอร์ชั่น 32 บิต ทำให้ต้องมีการแบ่งปันรีจิสทรีคีย์โดยปราศจากการทำทั้งรีไดเร็คและรีเฟล็กซ์

ในการเรียกใช้งานหรือหยุดการทำงานของรีจิสทรีรีเฟล็กซ์ชันสำหรับแต่ละคีย์นั้น จะใช้ฟังก์ชั่น RegDisableReflectionKey และ RegEnableReflectionKey โดยแอปพลิเคชันควรที่จะหยุดการทำงานของ รีเฟล็กซ์ชันนั้น ควรทำเฉพาะรีจิสทรีคีย์ที่ถูกสร้างขึ้นและไม่พยายามหยุดการทำงานของรีเฟล็กซ์ชันสำหรับทั้ง HKEY_LOCAL_MACHINE หรือ HKEY_CURRENT_USER ซึ่งการกำหนดว่าคีย์ไหนจะอยู่ในรายการของการ รีเฟรกชั่นนั้น จะเรียกใช้ฟังก์ชั่น RegQueryReflectionKey

2.1.7 บีเอชโอ (BHO)

2.1.7.2 Browser Helper Object (BHO)

บีเอชโอ (BHO) เป็น ดีแอลแอล (DLL) โมดูลที่ถูกออกแบบขึ้นเพื่อให้เป็นปลั๊กอิน (plugin) สำหรับ Microsoft's Internet Explorer web browser เพื่อใช้ในการเพิ่มฟังก์ชั่นต่างๆที่ต้องการ โดยส่วนมากแล้ว BHO จะถูกโหลดเพียงครั้งเดียวในตอนที่ทำการเรียกใช้งาน อินเทอร์เน็ต เอกซ์โพลเรอร์ (Internet Explorer) เท่านั้น แต่ในส่วนของ วินโดวส์ เอกซ์โพลเรอร์ (Windows Explorer) นั้นบีเอชโอใหม่จะถูกเรียกใช้ในแต่ละวินโดวส์

2.1.2.2 การดำเนินงานของบีเอชโอ

ในแต่ละครั้งที่มีการเรียกใช้งานอินเทอร์เน็ตเอกซ์โพลเรอร์ใหม่นั้น จะมีการตรวจสอบวินโดวส์รีจิสทรีคีย์

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Browser Helper Objects

ถ้าหากอินเทอร์เน็ตเอกซ์โพลเรอร์พบว่ามีคีย์นี้อยู่ในรีจิสทรีแล้ว มันจะทำการหา CLSID คีย์ซึ่งอยู่ภายใต้รายการของคีย์นั้น โดย CLSID คีย์ซึ่งอยู่ภายใต้บีเอชโอนี้จะบอกบราวเซอร์ (Browser) ให้รู้ว่าต้องทำการโหลดบีเอชโอใด ซึ่งการย้ายรีจิสทรีคีย์ออกไปนั้น ก็เพื่อป้องกันไม่ให้บีเอชโอถูกโหลดขึ้นมา สำหรับแต่ละ CLSID ที่ถูกระบุไว้สำหรับแต่ละบีเอชโอคีย์นั้นนั้น อินเทอร์เน็ตเอกซ์โพลเรอร์จะ

ทำการเรียก CoCreateInstance เพื่อเริ่มต้นเรียกบีเอชโอใน โพรเซส สเปซ (Process Space) เดียวกันให้เป็นบราวเซอร์ ซึ่งหากบีเอชโอเริ่มต้นทำงานในส่วน IObjectWithSite แล้วมันจะสามารถควบคุมและทราบถึงเหตุการณ์ที่เกิดขึ้นต่างๆของอินเทอร์เน็ตเอกซ์โพลเรอร์

2.1.3 API Hooking

API Hooking ประกอบด้วยการขัดขวางฟังก์ชั่นคอล (function call) ในโปรแกรมและทำการรีไดเรกท์ (redirect) มันไปยังฟังก์ชั่นอื่น ซึ่งการทำเช่นนี้จะทำให้ตัวแปรสามารถเปลี่ยนแปลงแก้ไขค่าได้ โดยโปรแกรมเริ่มต้นอาจถูกปลอมแปลงหากเลือกคืนค่าที่ผิดแทนที่จะเป็นการทำให้สำเร็จลุล่วงแทน และการทำทั้งหมดนี้จะเกิดขึ้นก่อนที่ฟังก์ชั้นที่แท้จริงจะถูกเรียกขึ้นมา และเรียกอีกครั้งเมื่อฟังก์ชั่นหลักสิ้นสุดการทำงาน โดยหลังจากที่มันทำการปรับเปลี่ยน, เก็บค่า หรือขยายตัวแล้วนั้น มันก็จะคืนการควบคุมกลับไปให้ยังฟังก์ชั่นเริ่มต้น

2.1.7 ดีแอลแอล (ไดนามิก ลิงค์ ไลบรารี) (DLL(Dynamic Link Library))

ดีแอลแอล ย่อมาจาก ไดนามิก ลิงค์ ไลบรารี (Dynamic Link Library) ซึ่งจริงๆแล้วคือโปรแกรมขนาดเล็กที่ถูกเรียกใช้ให้ทำงานเฉพาะบางอย่างโดยโปรแกรมหลัก โปรแกรมต่างๆที่ทำงานในระบบปฏิบัติการวินโดวส์ อย่างเช่น เวิร์ดและเอ็กซ์เซลจะมีการเรียกใช้ไฟล์ดีแอลแอลเป็นจำนวนมาก

ข้อแตกต่างที่เห็นได้ชัดก็คือ โปรแกรมส่วนใหญ่จะผ่านการเชื่อมโยงองค์ประกอบทั้งหมดที่เกี่ยวข้องเพื่อแปลงเป็นโค้ดโปรแกรมที่ต้องการก่อนทำงาน แต่ดีแอลแอลจะถูกโหลดเมื่อจำเป็นต้องใช้งานเท่านั้น ด้วยคุณสมบัตินี้ทำให้ประหยัดหน่วยความจำ (แรม (RAM)) ได้ (ไม่ต้องโหลดทั้งโปรแกรมไว้ในหน่วยความจำของเครื่องคอมพ์ตลอดเวลา) โดยไฟล์ดีแอลแอลจะมีนามสกุลเป็น .dll

จากนี้ไปจะเป็นการยกตัวอย่างให้เห็นภาพ โดยสมมติว่า ในช่วงเวลาที่กำลังใช้โปรแกรมเวิร์ดเพื่อแก้ไขข้อความในเอกสาร ขณะนั้นไฟล์ดีแอลแอลที่ใช้ควบคุมการพิมพ์ข้อความออกทางเครื่องพิมพ์จะไม่ถูกโหลดเข้าไปในหน่วยความจำ แต่เมื่อผู้ใช้ต้องการจะพิมพ์เอกสารที่แก้ไขเสร็จแล้วโดยคลิกเมนูพรินต์ (print) โปรแกรมเวิร์ดก็จะโหลดไฟล์ดีแอลแอลของเครื่องพิมพ์ให้ทำงาน ด้วยเหตุนี้การลบไฟล์ดีแอลแอลด้วยตนเองจึงไม่ใช่สิ่งที่ควรกระทำ เพราะหากทำการลบไฟล์ดีแอลแอลที่มีความสำคัญต่อระบบปฏิบัติการ เครื่องคอมพิวเตอร์อาจจะต้องได้รับการติดตั้ง หรือซ่อมแซมระบบปฏิบัติการก็เป็นได้

2.1.7 DLL Injection

การอินเจกต์ดีแอลแอล (DLL Injection) เข้าไปยังโพรเซสแอดเดรสสเปซ (Process Address Space) ทำโดยการใช้ฟังก์ชั่นคอล CreateRemoteThread ซึ่งฟังก์ชั่นนี้จะทำการสร้างเทรท (thread) ในแอดเดรสสเปซจำลอง (virtual address space) ของโพรเซสนั้นทันที ซึ่งตัวแปรต่างๆเราต้องทำการส่งค่าผ่านฟังก์ชั่นนั้นด้วย

ตัวแปรต่างๆที่ต้องทำการส่งค่าให้กับฟังก์ชั่น ได้แก่

ตารางที่ 2.2 แสดงคำอธิบายของตัวแปรต่างๆที่ใช้ส่งค่าให้กับฟังก์ชัน

ตัวแปร	คำอธิบาย
hProcess	จัดการโพรเซสที่เราจะทำการสร้างเทรทใหม่
lpThreadAttributes	เป็นพอยต์เตอร์ (pointer) ชี้ไปยังโครงสร้างของ SECURITY_ATTRIBUTES ซึ่งระบุถึงองค์ประกอบทางด้านความปลอดภัยของเทรทที่ถูกสร้างขึ้นใหม่
dwStackSize	ค่าขนาดเริ่มต้นของสแตค (stack)
lpStartAddress	พอยต์เตอร์ชี้ไปยัง LPTHREAD_START_ROUTINE ซึ่งเป็นฟังก์ชั่นที่ดำเนินการโดยเทรทที่ส้รางขึ้นใหม่
lpParameter	พอยต์เตอร์ชี้ไปยังตัวแปรเพื่อทำการผ่านค่าไปยังฟังก์ชั่นของเทรท
dwCreationFlags	ค่าที่ทำการควบคุมการส้รางใหม่ของเทรท
lpThreadId	พอยต์เตอร์ชี้ไปยังตัวแปรที่ได้รับเทรทไอดี (thread ID)

2.1.7 IPC – mail Slot

ไอพีซี (IPC : Inter-Process Communication) คือเซ็ทของเทคนิคที่ใช้ในการแลกเปลี่ยนข้อมูลระหว่างเทรทหลายๆเทรทใน 1 หรือ มากกว่า 1 โพรเซส ซึ่งโพรเซสอาจกำลังทำงานอยู่บน 1 หรือ มากกว่า 1 เครื่องคอมพิวเตอร์ที่เชื่อต่อกันภายในระบบเครือข่าย โดยไอพีซีเทคนิคนั้นประกอบด้วย Named Pipes, File Mapping, Mailslot, Remote Procedure Calls (RPC) และอื่นๆ

2.1.7 วินโดวส์ สตาร์ทอัพ โลเคชัน (Windows Start Up Location)

เนื่องจากเครื่องที่ถูกละเมิดความปลอดภัยไปนั้น แฮกเกอร์จะวางโปรแกรมแบ็คดอร์ที่จะต้องการเรียกใช้ขึ้นมาทุกครั้งตอนที่เปิดเครื่อง ดังนั้นวิธีการดักจับจึงต้องไปศึกษาวิธีการสตาร์ทอัพโปรแกรมของวินโดวส์ด้วย ข้อมูลต่อไปนี้จะเป็นวิธีการสตาร์ทอัพทั้งหมดที่เป็นไปได้ของวินโดวส์

2.1.7.1 ในรีจิสทรี รัน คีย์ (Registry RUN key)

Registry Keys:

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\ CurrentVersion\Run

HKEY_CURRENT_USER\Software\Microsoft\Windows\ CurrentVersion\Run

โดยถ้าเป็น HKEY_LOCAL_MACHINE ไฟล์นั้นจะสตาร์ทอัพทุกครั้งไม่ว่าจะป็นผู้ใช้งานใดๆ แต่ถ้าเป็น HKEY_CURRENT_USER ไฟล์นั้นจะสตาร์ทอัพทุกครั้งเฉพาะการล็อกอิน (login) ของผู้ใช้งานปัจจุบันเท่านั้น โดยทั่วไปแล้วไฟล์นั้นจะไม่ถูก สตาร์ทอัพใน เซฟโหมด (Safe mode) แต่ในกรณีที่ต้องการให้รันในเซฟโหมดด้วยต้องใส่ * เข้าไปหน้าแพทช์ (Path) จึงจะสามารถสตาร์ทอัพตอนเซฟโหมดได้

2.1.7.2 ในรีจิสทรี รันวันส์ โลคอลแมชชีน คีย์ (Registry RunOnce Local Machine key)

Registry Keys:

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\

โดยคีย์นี้มีไว้สำรับโปรแกรมประเภทเซ็ตอัพ (SetUp) ซึ่งเมื่อโปรแกรมดังกล่าวถูกรัน จนจบโปรแกรมแล้ว ค่าของโปรแกรมดังกล่าวในคีย์นี้ก็จะโดนลบทิ้งไปจากคีย์แต่ถ้าโปรแกรมทำงานไม่สิ้นสุด คีย์นี้ก็จะไม่ถูกลบทิ้ง เอ็นทรี่ (Entry) ต่างๆในคีย์นั้นจะถูกทำงานพร้อมๆกันดังนั้นคีย์จึงจำเป็นที่จะต้องทำงานจนเสร็จก่อนเอ็นทรี่ต่างๆใน HKEY_LOCAL_MACHINE\...\Run,HKEY_CURRENT_USER\...\Run,HKEY_CURRENT_USER\...\RunOnce และสตาร์ทอัพ โฟลเดอร์ (Startup Folders) จะถูกโหลดขึ้น

2.1.7.3 ในรีจิสทรี รันวันส์ เคอร์เรนท์ยูสเซอร์ คีย์ (RegistryRunOnceCurrentUserKey)

Registry Key:

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce

2.1.7.4 ในรันเซอร์วิสวันส์ คีย์ (RunServicesOnce Key)

Registry Key:

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce

2.1.7.5 ในรันเซอร์วิส คีย์ (RunServices Key)

Registry Keys:

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServices

2.1.7.6 ในเอกซ์พลอเยอย์ รัน คีย์ (Explorer Run key)

Registry Keys:

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run

2.1.7.7 ยูสเซอร์อินอิท คีย์ (UserInit Key)

Registry Key:

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows

คีย์นี้จะระบุว่าโปรแกรมอะไรจะถูกรันหลังจากการล็อกอินเข้าสู่วินโดวส์โดยโปรแกรมดีฟอล์ต (default) ที่ถูกเรียกใช้งานคือC:\windows\system32\userinit.exe

(Userinit.exe เป็นโปรแกรมที่จะรีสตอร์พรีไฟล์ (restore prefile) เช่น ฟอนต์(font), สี, etc. สำหรับยูสเซอร์ที่ล็อกอิน (login) เข้าไป โดยถ้าต้องการแอ็ด (add) โปรแกรมเพิ่มเข้าไปให้ใส่คอมมา (comma)

ตัวอย่างเช่น

HKLM\Software\Microsoft\WindowsNT\CurrentVersion\Winlogon\Userinit=C:\windows\system32\userinit.exe, c:\windows\badprogram.exe

โดยจะทำให้ 2 โปรแกรม (userinit, badprogram) นี้ถูกรันขึ้นมาโดยคีย์นี้จะเป็นวิธีทั่วไปที่โทรจัน (trojan), ไฮแจ็คเกอร์ (hijacker), สปายแวร์ (spyware) ใช้เพื่อทำออโตรัน (auto run)

2.1.7.8 โหลด คีย์ (Load Key)

Registry Key:

HKEY_CURRENT_USER\Software\Microsoft\WindowsNT\CurrentVersion\Windows\load

2.1.7.9 โนติฟาย คีย์ (Notify key)

Registry Key:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\Winlogon\Notify

คีย์นี้ถูกใช้เพื่อเพิ่มโปรแกรมที่จะถูกรันเฉพะตอนที่เกิดเหตุการณ์ที่กำหนด ตัวอย่างเช่น เหตุการณ์ล็อกออน (logon), ล็อกออฟ (logoff), สตาร์ทอัพ(startup), ชัทดาวน์(shutdown), สตาร์ทสกรีนเซฟเวอร์ (startscreensaver) และสตอปสกรีนเซฟเวอร์ (stopscreensaver) โดยอีเวนต์ (event) ดังกล่าวจะถูกสร้างจากโปรแกรม winlogon.exe เมื่อ winlogon.exe ทำอีเวนต์ดังกล่าวแล้ววินโดวส์ก็จะไปดูในโนติฟาย รีจิสติ คีย์ (notify registry key) เพื่อดูดีแอลแอล (DLL) ที่จะรอรับอีเวนต์ ดังกล่าว

2.1.7.10 แอพอินอิท ดีแอลแอล (AppInit_DLLs)

Registry Key:

HKEY_LOCAL_MACHINE\Software\Microsoft\WindowsNT\CurrentVersion\Windows

แอพอินอิท ดีแอลแอล (AppInit_DLLs) นั้นจะเก็บลิสต์ (list) ของที่จะถูกโหลด (load) เมื่อ user32.dll ได้ถูกโหลดซึ่งโปรแกรมที่เรียกใช้งานบน แพลทฟอร์ม(platform) วินโดวส์ (windows)

นั้นส่วนมากจะใช้ user32.dll ทั้งสิ้น ซึ่งก็หมายความว่าโปรแกรมส่วนมากก็จะใช้ดีแอลแอลตัวที่เราเพิ่มเข้าไปในแอพอินอิท ดีแอลแอล เช่นกัน

2.1.7.11 เชลล์เซอร์วิสอ็อบเจกต์ดีเลย์โหลด (ShellServiceObjectDelayLoad)

Registry Key:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\ShellServiceObjectDelayLoad

2.1.7.12 แชร์ดทาสก์สเกดูล์เลอร์ (SharedTaskScheduler)

Registry Key:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\SharedTaskScheduler

2.1.7.13 รีจิสติ คีย์ (Registry Key) อื่นๆ

นอกจากนั้นวินโดวส์ยังสตาร์ทอัพผ่าน รีจิสติ คีย์ อื่นๆอีก ได้แก่

[HKEY_CLASSES_ROOT\exefile\shell\open\command] ="\"%1\" %*"

[HKEY_CLASSES_ROOT\comfile\shell\open\command] ="\"%1\" %*"

[HKEY_CLASSES_ROOT\batfile\shell\open\command] ="\"%1\" %*"

[HKEY_CLASSES_ROOT\htafile\Shell\Open\Command] ="\"%1\" %*"

[HKEY_CLASSES_ROOT\piffile\shell\open\command] ="\"%1\" %*"

[HKEY_LOCAL_MACHINE\Software\CLASSES\batfile\shell\open\command] ="\"%1\" %*"

[HKEY_LOCAL_MACHINE\Software\CLASSES\comfile\shell\open\command] ="\"%1\" %*"

[HKEY_LOCAL_MACHINE\Software\CLASSES\exefile\shell\open\command] ="\"%1\" %*"

[HKEY_LOCAL_MACHINE\Software\CLASSES\htafile\Shell\Open\Command] ="\"%1\" %*"

[HKEY_LOCAL_MACHINE\Software\CLASSES\piffile\shell\open\command] ="\"%1\" %*"

"\"%1" %*" เป็นค่าดีฟอล์ตถ้าค่านี้ถูกแก้ไขเป็น "\"spyware.exe %1\" %*\" แล้ว spyware.exe จะถูกเรียกใช้งานตอนสตาร์ทอัพทุกครั้ง

2.1.7.14 แอคทีฟเอ็กซ์ คอมโพเนนท์ (ActiveX Component)

Registry keys:

HKEY_CURRENT_USER\Software\Microsoft\Active Setup\Installed Components\{4175C5F3-D47F-143B-DD4D-E67A0EB4E773} – StubPath = "Exe path"

HKEY_LOCAL_MACHINE\Software\Microsoft\Active Setup\Installed Components\{4175C5F3-D47F-143B-DD4D-E67A0EB4E773} – StubPath = "Exe path"

2.1.7.15 สตาร์ทอัพ โฟลเดอร์ (Startup Folder) สำรับยูสเซอร์ทุกคน

โปรแกรมในโฟลเดอร์ด้านล่างนี้จะถูกรันทุกครั้งเมื่อยูสเซอร์คนใดก็ได้ล็อกออน

ระบบปฏิบัติการวินโดวส์ 7

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup

ระบบปฏิบัติการวินโดวส์ XP

C:\Documents and Settings\All Users\Start Menu\Programs\Startup

2.1.7.16 สตาร์ทอัพ โฟลเดอร์ (Startup Folder) สำรับแต่ละยูสเซอร์

โปรแกรมในโฟลเดอร์ด้านล่างนี้จะถูกรันทุกครั้งเมื่อยูสเซอร์คนดังกล่าวได้ล็อกออน

ระบบปฏิบัติการวินโดวส์ 7

C:\Users\<username>\AppData\Roaming\Microsoft\Windows\StartMenu\Programs\Startup

ระบบปฏิบัติการวินโดวส์ XP

C:\Documents and Settings\<LoginName>\Start Menu\Programs\Startup

2.1.7.17 ไฟล์อื่นๆ

อันได้แก่

c:\autoexec.bat

c:\config.sys

windir\wininit.ini (โดยทั่วไปถูกใช้โดยเซ็ตอัพโปรแกรม (Setup Program) ที่ต้องถูกลบทิ้งเมื่อรัน windir\winstart.bat)

windir\win.ini – [windows] "load"

windir\win.ini – [windows] "run"

windir\system.ini – [boot] "shell"

windir\system.ini – [boot] "scrnsave.exe"

windir\dosstart.bat (ใช้ในระบบปฏิบัติการวินโดวส์ 95 หรือ ระบบปฏิบัติการวินโดวส์ 98 เมื่อเลือก Restart in MS-DOS mode ในเมนู)

windir\system\autoexec.nt

windir\system\config.nt

2.2 อนาไลซ์เซอร์(C#) (Analyzer (C#))

2.2.1 XPath

XPath เป็นภาษาที่ใช้สำหรับการค้นหาข้อมูลในเอกสาร XML หรือเป็นซินแท็ก (syntax) สำหรับการกำหนดส่วนต่างๆในเอกสาร XML โดยจะใช้พาทซ์เอ็กเพรซชั่น (path expressions) ในการระบุตำแหน่งในเอกสาร ซึ่งประกอบไปด้วยไลบรารี (library) ของฟังก์ชั่นพื้นฐานต่างๆ

XPath Path Expressions

XPathจะใช้พาทซ์เอ็กเพรซชั่นเพื่อเลือกโหนด (node) หรือเซ็ทของโหนดในเอกสาร XML โดยพาทซ์นี้จะมีหน้าตาคล้ายกับที่ใช้ในไฟล์ซิสเท็ม (file system) ทั่วไป

2.2.2 WPF

WPF (Windows Presentation Foundation) ที่จริงแล้วคือ เฟรมเวิร์ค (framework) ตัวใหม่ที่ถูกนำเสนอโดย .NET framework 3.0 ซึ่งทำให้เราสามารถเขียนโปรแกรมได้อย่างมีประสิทธิภาพและมีความคล่องตัวมากยิ่งขึ้น โดยมันจะเรียก Direct3D ในการทำงานซึ่งใช้กราฟฟิกการ์ดในการสร้างผลลัพธ์ออกทางหน้าจอ ดังนั้นการวาดหรือสร้างสรรค์ผลงานด้วยรูปแบบดังกล่าวนี้จะช่วยเพิ่มความละเมียดละไมอีกทั้งยังมีโอกาสที่จะเพิ่มประสิทธิภาพการทำงานของอุปกรณ์ในตัวเครื่องอีกด้วย

2.3 อนาไลซ์เซอร์(เว็บ) (Analyzer(WEB))

2.3.1 CSS

CSS (Cascading Style Sheets) คือ สไตล์ (Style) ที่ใช้ในการกำหนดว่า จะใช้วิธีใดในการแสดงผล HTML Element หรือเป็นสไตล์ที่ใช้ในการจัดวางสี หรือลักษณะของการแสดงหน้าเว็ปที่ต้องการสื่อออกมา

2.3.2 HTML

HTML (Hyper Text Markup Language) ภาษาคอมพิวเตอร์ที่ใช้ในการแสดงผลของเอกสารบนเว็บไซต์หรือที่เรียกว่าเว็บเพจ ถูกพัฒนาและกำหนดมาตรฐานโดยองค์กร World Wide Web Consortium (W3C) และจากการพัฒนาทางด้านซอฟต์แวร์ของไมโครซอฟท์ทำให้ภาษา HTML เป็นอีกภาษาหนึ่งที่ใช้เขียนโปรแกรมได้ หรือที่เรียกว่า HTML Application

HTML เป็นภาษาประเภท มาร์คอัพ (Markup) สำหรับการสร้างเว็บเพจ โดยใช้ภาษา HTML สามารถทำโดยใช้โปรแกรม Text Editor ต่างๆ เช่น Notepad, Editplus หรือจะอาศัยโปรแกรมที่เป็นเครื่องมือช่วยสร้างเว็บเพจ เช่น Microsoft FrontPage, Dream Weaver ซึ่งอำนวยความสะดวกในการสร้างหน้า HTML ส่วนการเรียกใช้งานหรือทดสอบการทำงานของเอกสาร HTML จะใช้โปรแกรมเว็ปเบราเซอร์เช่น IE Microsoft Internet Explorer (IE), Mozilla Firefox, Safari, Opera, และ Netscape Navigator เป็นต้น

2.3.3 Java Script

จาวาสคริปต์ (JavaScript) เป็นภาษาโปรแกรมประเภทหนึ่ง ที่เรียกกันว่า "สคริปต์" ซึ่งมีวิธีการทำงานในลักษณะแปลความและดำเนินงานไปทีละคำสั่ง โดยภาษานี้มีชื่อเดิมว่า LiveScript ได้รับการพัฒนาขึ้นโดย Netscape ด้วยวัตถุประสงค์เพื่อช่วยให้เว็บเพจสามารถแสดงเนื้อหาที่มีการเปลี่ยนแปลงไปได้ ตามเงื่อนไขหรือสภาพแวดล้อมที่แตกต่างกัน หรือสามารถโต้ตอบกับใช้งานได้มากขึ้น ทั้งนี้เพราะภาษา HTML แต่เดิมนั้นเหมาะสำหรับใช้แสดงเอกสารที่มีเนื้อหาคงที่แน่นอนและไม่มีลูกเล่นอะไรมากมาย

2.3.4 Backbone.js

Backbone.js เป็นเฟรมเวิร์คที่ช่วยให้การเขียนโค้ดของจาวาสคริปต์เป็นไปตามหลักของ MVC

2.3.5 Underscore.js

Underscore.js เป็นไลบรารี (Library) ที่ช่วยในการจัดการกับอาร์เรย์ (Array) หรืออ็อบเจกต์ (Object) เช่น การไล่ลำดับของอาร์เรย์

2.4 ข้อมูลทั่วไป

ประกอบไปด้วยเนื้อหาที่เกี่ยวข้องในส่วนต่างๆ อันได้แก่

2.4.1 ระบบตรวจจับการบุกรุก

ในการพัฒนาซอฟต์แวร์ของโครงงานนี้ผู้ทำได้ทำการศึกษาเกี่ยวกับโมเดล (model) ของระบบตรวจจับการบุกรุกเช่นกัน เพื่อหาแนวทางและแนวคิดในการดักจับ มัลแวร์ (malware)

อินทรูชั่น ดีเทคชัน ซิสเท็ม (Intrusion Detection System (IDS)) คือ ระบบที่คอยตรวจจับการบุกรุกและวิเคราะห์ข้อมูลที่อยู่บนเครือข่ายภายในและมาจากเครือข่ายภายนอกว่ามีพฤติกรรมที่เป็นความเสี่ยงและก่อความเสียหายต่อระบบงานภายในองค์กรหรือไม่ โดยระบบจะแจ้งเตือนให้กับผู้ดูแลระบบทราบ และหยุดพฤติกรรมดังกล่าวทันที

2.4.1.1 ผู้บุกรุกระบบ (Intruder)

แบ่งออกเป็น 2 ประเภท ได้แก่

1) เอาท์ไซด์เดอร์ (Outsider)

เอาท์ไซด์เดอร์ หมายถึง ผู้บุกรุกจากภายนอกเครือข่าย และบุคคลที่อาจจะโจมตีมาจากภายนอกเช่น การเปลี่ยนแปลงหน้ากากของเว็บเซิร์ฟเวอร์ (web server) หรือการฟอร์เวิร์ดเมล์ (forward mail) ผ่านทางอีเมลเซิร์ฟเวอร์ (e-mail server) ซึ่งการบุกรุกจากภายนอกอาจมาจากอินเทอร์เน็ต การไดอัล-อัพ (dial-up) การบุกเข้าไป หรือเครือข่ายของคู่ค้าที่เชื่อมต่อมายังเครือข่าย

2) อินไซด์เดอร์ (Insider)

อินไซด์เดอร์ หมายถึง ผู้บุกรุกที่มีสิทธิ์ในการใช้เครือข่ายภายใน รวมทั้งผู้ใช้ที่ใช้สิทธิ์ในทางที่ผิด หรือการลักลอบใช้สิทธิ์ของผู้ใช้คนอื่นๆ ที่มีสิทธิ์เหนือกว่า

2.4.1.2 วิธีการเจาะเข้าสู่ระบบของผู้บุกรุก

แบ่งออกเป็น 3 วิธี ได้แก่

1) ฟิสิคอล อินทรูชั่น (Physical Intrusion)

ผู้บุกรุกมีการเชื่อมต่อทางกายภาพกับเครื่องหรือระบบเครือข่าย

2) ซิสเท็ม อินทรูชั่น (System Intrusion)

ผู้บุกรุกมีบัญชีผู้ใช้งานเรียบร้อยแล้ว แต่มีสิทธิการเข้าถึงต่ำ ถ้าไม่ได้มีการ อัพเดทแพทช์ (update patch) ให้กับระบบ ผู้บุกรุกจะใช้ช่องโหว่ของระบบในการครอบครองสิทธิ์ของผู้ดูแลระบบ

3) รีโมท อินทรูชั่น (Remote Intrusion)

ผู้บุกรุกพยายามที่จะเจาะเข้าสู่ระบบข้ามเครือข่าย ซึ่งผู้บุกรุกจะไม่มีสิทธิ์ใดๆเลยบนเครือข่ายนั้น

2.4.1.3 ประเภทของระบบการตรวจจับการบุกรุก

แบ่งตามลักษณะของข้อมูลที่นำมาวิเคราะห์ได้ 2 ประเภท ได้แก่

1) โฮสต์-เบส ไอดีเอส (Host-based IDS)

ทำการตรวจจับข้อมูลที่ไหลเข้าและออกคอมพิวเตอร์แต่ละเครื่อง นอกจากนั้นระบบก็ยังตรวจสอบความสมบูรณ์ของซิสเท็มไฟล์ (system files) และเฝ้าดูโพรเซส (processes) ที่น่าสงสัย

2) เน็ตเวิร์ก-เบส ไอดีเอส (Network-based IDS)

ทำการเฝ้าดูข้อมูลบนเครือข่ายโดยที่ระบบดังกล่าวจะทำการรับข้อมูลทั้งหมดที่อยู่บนส่วนของเครือข่ายที่รับผิดชอบ นอกเหนือจากส่วนของเครือข่ายที่รับผิดชอบและชนิดของการสื่อสารอื่นๆ แล้ว ระบบดังกล่าวก็ไม่สามารถทำการตรวจจับแพ็คเกตต่างๆได้ โดยจะถูกตรวจจับโดยเซนเซอร์ (sensor) ของระบบไอดีเอส ซึ่งเซนเซอร์จะมองเห็นเฉพาะแพ็คเกตที่ผ่านส่วนของเครือข่ายที่ sensor นั้นติดอยู่ โดยแพ็คเกตต่างๆจะเป็นที่สนใจของเซนเซอร์ก็ต่อเมื่อแพ็คเกตนั้นเข้ากับซิกเนเจอร์ (signature) ที่กำหนด ซึ่งปกติแล้วซิกเนเจอร์จะมี 3 ประเภทคือ

- สตริง ซิกเนเจอร์ (String signature)

จะมองหาเท็กซ์สตริง (text string) ซึ่งอาจบ่งบอกถึงการโจมตี ตัวอย่างเช่น " cat " + + " 7% rhost " อาจทำให้ระบบยูนิกซ์ (UNIX) เกิดช่องโหว่ต่อการโจมตีบนเครือข่าย

- พอร์ท ซิกเนเจอร์ (Port signature)

จะเฝ้าดูการพยายามติดต่อเข้ามาทางพอร์ท (port) ที่รู้จักกันดี และมักจะถูกโจมตี เช่น เทลเน็ต(telnet) จะใช้ทีซีพี (TCP) พอร์ท 23, เอฟทีพี (FTP) จะใช้ทีซีพี พอร์ท 21/20, ซันอาร์พีซี (SUNRPC) ใช้ ทีซีพี/ยูดีพี (UDP) พอร์ท 111 และไอแม็พ (IMAP) จะใช้ทีซีพี

พอร์ท 143 ซึ่งถ้าระบบของเราไม่ได้เปิดพอร์ทดังกล่าว แต่มีการพยายามเชื่อมต่อเข้ามา แสดงว่าแพ็คเกตดังกล่าวอาจจะมีจุดประสงค์มุ่งร้ายก็เป็นได้

- เฮดเดอร์ ซิกเนเจอร์ (Header signature)

พยายามมองหาคอมบิเนชั่น (combination) ที่อันตรายและผิดกฎของแพ็คเกตเฮดเดอร์ (packet header) ตัวอย่างที่เห็นได้ชัดของเฮดเดอร์ซิกเนเจอร์คือ ทีซีพีแพ็คเกต ซึ่งมีทั้งซินแฟล็กซ (SYN Flags) และฟินแฟล็กซ (FIN Flags)

2.4.1.4 กระบวนการตรวจจับการบุกรุก

แบ่งได้เป็น 2 วิธี ได้แก่

1) โนวเลจ-เบส ไอดีเอส (Knowledge-based IDS)

โนวเลจ-เบส ไอดีเอส จะอาศัยข้อมูลที่เกี่ยวกับการโจมตีชนิดต่างๆ พร้อมทั้งช่องโหว่ของระบบ ในการตรวจจับการให้ใช้ช่องโหว่ต่างๆ เมื่อความพยายามในการเข้าใช้นั้นถูกจับได้ ไอดีเอสก็จะทำการแจ้งเตือน เพราะฉะนั้นจะเห็นได้ว่า ความสมบูรณ์และประสิทธิภาพของไอดีเอสชนิดนี้จะขึ้นอยู่กับความทันสมัยของข้อมูลเกี่ยวกับการโจมตีต่าง

ข้อดีของวิธีการแบบนี้คือ อัตราการเกิดการแจ้งเตือนผิดๆนั้นจะต่ำ และข้อมูลที่ได้จากไอดีเอสนั้นจะมีรายละเอียดที่ดีทำให้ง่ายต่อผู้ใช้ในการป้องกันและแก้ไขการโจมตี

ข้อเสียของวิธีการแบบนี้คือ ความยากในการรวบรวมข้อมูลเกี่ยวกับรูปแบบการโจมตี และการปรับปรุงข้อมูลเกี่ยวกับช่องโหว่ต่างๆให้ทันสมัยอยู่เสมอ เนื่องจากข้อมูลต่างๆนั้นขึ้นอยู่กับระบบปฏิบัติการ, เวอร์ชั่น, แพลทฟอร์ม(platform) และ แอปพลิเคชัน(application) นอกจากนั้นการตรวจจับการโจมตีจากภายในก็ทำได้ยาก เนื่องจากการโจมตีจากภายในเกี่ยวกับการละเมิดสิทธิของผู้ใช้งาน (user) ซึ่งไม่ได้เกี่ยวข้องกับช่องโหว่แต่อย่างใด

2) บีเฮฟวิเออร์-เบส ไอดีเอส (Behavior-based IDS)

จะมีการแจ้งเตือนเมื่อระบบมีการตรวจพบความเบี่ยงเบนและความผิดปกติของระบบหรือของผู้ใช้จากการใช้ระบบปกติ ซึ่งรูปแบบของพฤติกรรมที่เป็นปกตินั้นจะถูกรวบรวมจากข้อมูลอ้างอิงต่างๆ หลังจากนั้นไอดีเอสจะทำการเปรียบเทียบระหว่างพฤติกรรมในขณะนั้นกับรูปแบบอ้างอิง ดังนั้นจะเห็นได้ว่า ฟอลซ อลาม (false alarm) จะเกิดขึ้นได้บ่อยครั้ง

ข้อดีของการตรวจจับโดยใช้เทคนิคลักษณะนี้ คือสามารถที่จะตรวจจับการบุกรุกแบบใหม่ๆ ที่ไม่เคยมีมาก่อน และความเกี่ยวโยงกับระบบปฏิบัติการค่อนข้างต่ำ รวมทั้งยังสามารถที่จะตรวจจับการบุกรุกที่ไม่ได้โจมตีช่องโหว่เช่น การโจมตีจากภายใน

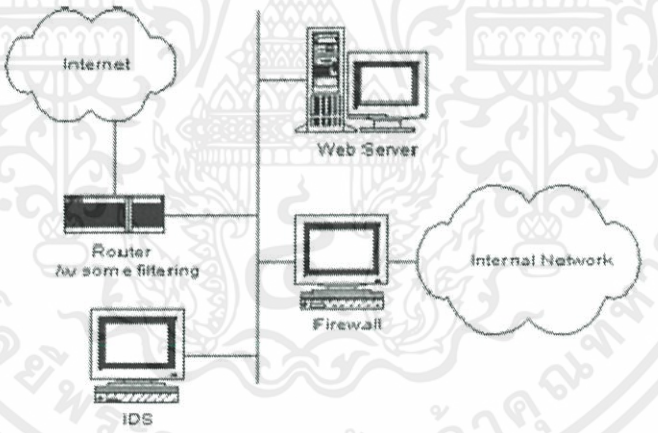
ข้อเสียที่สำคัญที่สุดคือ ฟอลซ อลามจะค่อนข้างสูง ในช่วงของการศึกษาพฤติกรรมของระบบ และเนื่องจากพฤติกรรมจะเปลี่ยนแปลงอยู่ตลอดเวลา เพราะฉะนั้นไอดีเอสก็ต้องใช้เวลาในการศึกษา และเป็นเหตุให้ไอดีเอสขัดข้องหรืออาจทำให้เกิดฟอลซ อลามมากขึ้น

2.4.1.5 การวางระบบอินทรูชั่น ดีเทคชัน

สามารถทำได้ 2 แบบ ได้แก่

1) แอทแทค ดีเทคชัน (Attack Detection)

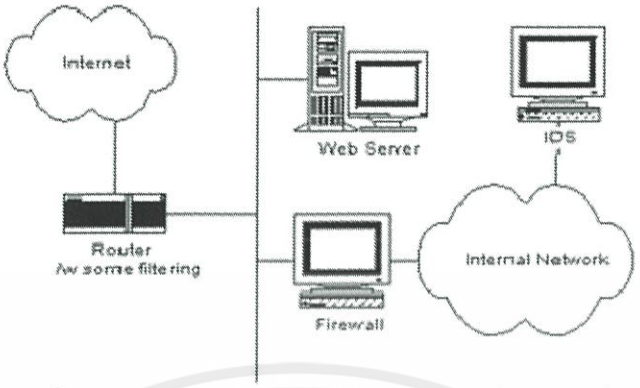
แอทแทค ดีเทคชัน คือ การตรวจจับการโจมตี จะตรวจจับการบุกรุกก่อนที่จะเข้าถึงเครือข่ายภายใน และวางไอดีเอสไว้หน้าไฟร์วอลล์ (firewall) การตรวจจับการบุกรุกไฟร์วอลล์ และเว็บเซิร์ฟเวอร์ (เพราะสองตัวนี้อยู่ด้านนอกไฟร์วอลล์) ผลการตรวจจับช่วยให้วิเคราะห์แนวโน้มความเสี่ยงได้เช่น เดือนนี้มีคนพยายามแฮ็ค (hack) มากกว่าเดือนที่แล้ว xxx ครั้ง หรือ แฮกกิ้ง (hacking) เกิดช่วงตี 4-5 มากที่สุด เป็นต้น



รูปที่ 2.1 การวางระบบไอดีเอสแบบแอทแทค ดีเทดชั่น

2) อินทรูชั่น ดีเทคชัน (Intrusion Detection)

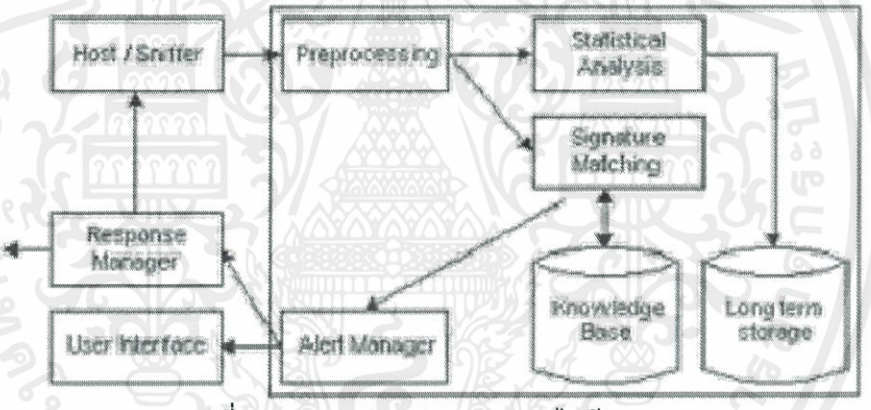
อินทรูชั่น ดีเทคชัน คือ การตรวจจับการบุกรุกเมื่อผู้บุกรุกผ่านเข้ามาภายในเครือข่ายได้แล้ว และวางไอดีเอสไว้หลังไฟร์วอลล์ จะเป็นการตรวจจับการบุกรุกที่เข้าถึงเครือข่ายภายใน ซึ่งบอกถึงการรั่วของไฟร์วอลล์ได้, ตรวจจับการบุกรุกผ่านแบ็คดอร์ และตรวจจับการบุกรุกที่เกิดขึ้นจากภายในเครือข่ายเอง



รูปที่ 2.2 การวางระบบไอดีเอสแบบอินทรูชั่น ดีเทดชั่น

2.4.1.6 ส่วนประกอบของไอดีเอส

ไอดีเอสประกอบด้วยองค์ประกอบต่างๆ ได้แก่



รูปที่ 2.3 แสดงส่วนประกอบของไอดีเอส

1) โฮสต์ ซิสเท็ม/เน็ตเวิร์ก ซนิฟเฟอร์ (Host system/Network sniffer)

ทำหน้าที่เป็นตัวตรวจจับเหตุการณ์ต่างๆที่เกิดขึ้นในโฮสต์หรือเครือข่าย ข้อมูลจากส่วนนี้เป็นเหมือนอินพุท (Input) ที่เข้าไปประมวลผลในไอดีเอส

2) พรี-โพรเซสซิ่ง (Pre-processing)

จัดรูปแบบของอินพุทที่รับเข้ามาเพื่อให้นำไปประมวลผลได้สะดวกต่อไป

3) สเตติสติคอล อนาไลซิส (Statistical analysis)

ส่วนวิเคราะห์ผลทางสถิติ

4) ซิกเนเจอร์ แมทชิ่ง (Signature matching)

ส่วนวิเคราะห์จากพฤติกรรมที่มีแบบแผน แนวความคิดตรงนี้นำมาจากที่ว่า การบุกรุกมักจะมีรูปแบบที่ค่อนข้างแน่นอน ส่วนนี้จะทำงานได้ก็ต่อเมื่อมีข้อมูลมากพอที่จะวิเคราะห์ได้ว่าพฤติกรรมที่ปรากฏในระบบเป็นรูปแบบของการบุกรุกหรือไม่

5) โนวเลจ เบส (Knowledge Base)

เป็นตัวเก็บข้อมูลเกี่ยวกับพฤติกรรมของการบุกรุก ข้อมูลนี้ถูกใช้โดยส่วนของซิกเนเจอร์ แมทชิ่ง โดยข้อมูลส่วนนี้มีความสำคัญมากกับระบบ เป็นตัวตัดสินว่าระบบฉลาดพอที่จะตรวจจับการบุกรุกได้หรือไม่

6) อะเลิท เมเนเจอร์ (Alert Manager)

ทำหน้าที่เป็นตัวตัดสินใจว่าเหตุการณ์ที่เกิดขึ้นในระบบควรจะต้องแจ้งเตือนหรือไม่ ระบบจะมีความเซนซีทีฟ (sensitive) มากหรือน้อยแค่ไหนก็ขึ้นอยู่กับส่วนนี้เช่นกัน

7) ยูสเซอร์ อินเตอร์เฟส (User Interface)

เป็นส่วนที่โต้ตอบกับผู้ใช้ อาจจะเป็นการแสดงผลที่หน้าจอ ส่งเสียงเตือนถึงการบุกรุก สั่งพิมพ์เป็นฮาร์ดก๊อปปี้ (hardcopy) หรือแม้แต่เชื่อมกับเพจเจอร์ (pager) หรือโทรศัพท์ นอกจากนี้ยูสเซอร์ อินเตอร์เฟสยังเป็นส่วนโต้ตอบระหว่างผู้ใช้กับระบบเพื่อเปลี่ยนแปลงหรืออัพเดทข้อมูลในโนวเลจเบส

8) เรสพอนส์ เมเนเจอร์ (Response Manager)

รับข้อมูลจากอะเลิทเมเนเจอร์เพื่อนำมาตัดสินใจว่าจะโต้ตอบกับการบุกรุกอย่างไร

2.4.1.7 รูปแบบของไอดีเอส

แบ่งรูปแบบการทำงานของไอดีเอสออกได้เป็น 5 อย่างตามลักษณะวิธีการตรวจจับ ได้แก่

1) อนอมาลิ ดีเทคชัน (Anomaly Detection)

อะนอมาลิ ดีเทคชัน คือ การตรวจหาสิ่งผิดปกติ โดยมีกลไกก็คือต้องวิเคราะห์ระบบ หรือเครือข่าย (ขึ้นกับแหล่งข้อมูล) ให้ได้คำตอบก่อนว่าอะไรคือการทำงาน "ปกติ" การตรวจจับจะวัดตามค่าทางสถิติ หรือ ฮิวริสติก (heuristic) เพื่อดูว่าเหตุการณ์ที่เกิดอยู่ในขอบเขตของคำว่า "ปกติ" หรือไม่ ถ้าไม่แสดงว่าเหตุการณ์ที่เกิดเป็นสิ่งที่ไม่ปกติและหมายถึงการบุกรุก ปกติจะใช้การวิเคราะห์โดย

นิวรอล เน็ต (neural nets),สเตติสติคอล อนาไลซิส (statistical analysis)หรือสเตท-เชนจ์อนาไลซิส (state-changed analysis)

- ข้อดี คือสามารถปรับให้ตรวจจับการบุกรุกได้ทุกประเภท รวมทั้งการบุกรุกที่ไม่เคยเกิดขึ้นมาก่อน
- ข้อเสีย คือเกิดความผิดพลาดในการตรวจจับบ่อย
- ตัวอย่างของระบบ ได้แก่ ไอดีอีเอส/เอ็นไอดีอีเอส (IDES/NIDES (Statistical + Rule-based detection)), จีอาร์ไอดีเอส (GrIDS (Graph-based IDS)) และเอ็มเมอรัลด์ (Emerald (Multi-layer IDS))

2) โฮสต์ยูซ ดีเทคชัน (Misuse Detection)

คือหาว่าอะไรคือองค์ประกอบของการบุกรุกแล้วพยายามตรวจจับจุดนั้น วิธีการอาจจะใช้เน็ตเวิร์ก เกรป(Network grep) หาสตริงในเน็ตเวิร์ก คอนเน็คชั่นที่แสดงถึงการบุกรุก หรือแพทเทิร์นแมทชิ่ง (pattern matching) ตรวจการเปลี่ยนแปลงของสเตท (state) เช่น โอนเนอร์ (owner) ของ /etc/passwd ถูกเปลี่ยน ตามด้วย /etc/passwd ถูกเปิดแก้ไขและบันทึกลงไปใหม่

- ข้อดี คือ ใช้งานได้ง่ายและเร็ว โอกาสผิดพลาดมีน้อย
- ข้อเสีย คือ ต้องรู้จักวิธีการบุกรุกถึงจะตรวจจับได้ และมีโอกาสที่ระบบจะถูกลวงได้ง่าย
- ตัวอย่างของระบบ ได้แก่ ไอเอสเอส เรียลซีเคียว (ISS RealSecure),ซิสโก้ เน็ตเรนเจอร์ (Cisco NetRanger),เอ็นเอไอ ไซเบอร์ค็อพ (NAI CyberCop) และเอ็นเอฟอาร์ (NFR (Network Flight Recorder))

3) เบอร์กละ อะลาม (Burglar Alarm)

คือ มิสยูซ ดีเทคชันที่เจาะจงเป้าหมายที่แน่นอน การทำงานจะขึ้นอยู่กับการตั้งโพลิซี (policy) ของไซท์ (site) นั้นๆ โดยจะตรวจจับการฝ่าฝืนโพลิซีที่กำหนดไว้เป็นหลัก โดยเบอร์กละอะลามอย่างง่ายอาจจะทำได้จากทีซีพีดั๊มพ์ + เพิร์ล (tcpdump + perl) หรือที่เป็นซอฟต์แวร์ก็มีเน็ตล็อก (NetLog) และเน็ตเวิร์ก ไฟล์ท เรคคอร์ดเดอร์ (Network Flight Recorder)

- ข้อดี คือมีความน่าเชื่อถือ ใช้งานได้ง่าย และอาจจะตรวจจับการบุกรุกที่ไม่รู้จัก
- ข้อเสีย คือเป็นโพลิซี-เบส (policy-based) ซึ่งต้องอาศัยความรู้และประสบการณ์ในการตั้งโพลิซี

4) ฮันนี่ พอทส์ (Honey pots)

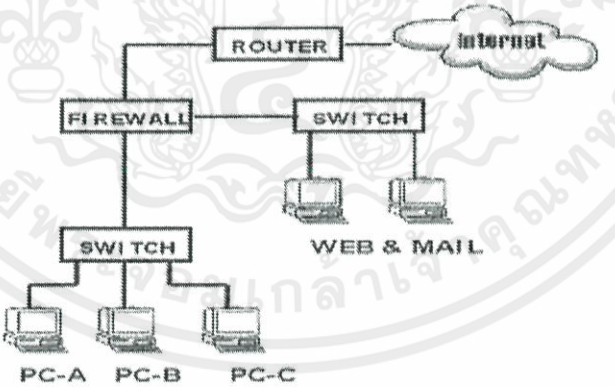
คือระบบลวง ที่ตั้งเพื่อล่อให้ถูกบุกรุก ส่วนการตรวจจับสามารถใช้ทูลส์ (tools) ง่ายๆเช่น ทีซีพีแรพเพอร์ (tcpwrapper), เบอร์กละ อะลาม (burglar alarm) หรือแม้แต่ซิสเท็มล็อก (system logs)

- ข้อดี คือใช้งานง่าย และไม่ลดประสิทธิภาพของระบบ
- ข้อเสีย คือ ไม่สามารถป้องกันการบุกรุกจากแฮกเกอร์ (Hacker) ที่มีความสามารถสูงๆได้

5) ไฮบริด ไอดีเอส (Hybrid IDS)

คือระบบรวมหลายๆระบบมาผสมกัน ระบบที่มีในปัจจุบันส่วนใหญ่จะเป็นไฮบริด ไอดีเอส เช่น มิสยูซ ดีเทคชัน (misuse detection) + เอ็กซ์เพิร์ท ซิสเท็ม (expert system) + สเตติสติคอล อนอมาลิ อนาไลซิส (statistical anomaly analysis) ไฮบริดไอดีเอสระบบนี้เป็นระบบที่ดีที่สุด เพราะใช้การทำงานหลายๆอย่างเพื่อตรวจจับและกำจัดข้อเสียที่มีจากระบบดังกล่าวข้างต้น แต่มีข้อเสียคือ เกิดฟอลซโพสซิทีฟ (false positive) มากเกินไป

2.4.1.8 ความเสี่ยงที่จะเกิดขึ้นในระบบงานที่ไม่มีการติดตั้งไอดีเอส



รูปที่ 2.4 ทิปปิคอล เน็ตเวิร์ก วิท ไฟร์วอลล์ (Typical Network with Firewall)

1) ความเสี่ยงเนื่องจากการเปิดบริการใช้งานบางประเภทที่ไฟร์วอลล์ (Firewall) ให้กับบุคคลภายนอกที่มาจากอินเทอร์เน็ต

ความเสี่ยงเนื่องจากการเปิดบริการใช้งานบางประเภทที่ไฟร์วอลล์ให้กับบุคคลภายนอกที่มาจากอินเทอร์เน็ตเช่น การบริการเว็บแอปพลิเคชัน, การบริการรับส่งเมล์ และการบริการถ่ายโอนข้อมูล (เอฟทีพี) เป็นต้น เนื่องจากผู้ไม่ประสงค์ดีจะนิยมเข้ามาจู่โจมและก่อให้เกิดความเสียหายแก่

ระบบงานภายในองค์กร โดยเข้าทางที่ไฟร์วอลล์อนุญาตเป็นส่วนมาก ตัวอย่างของความเสี่ยงที่พบเห็นได้บ่อยครั้ง เช่น

- เอสเอ็มทีพี โอเวอร์โฟลว (SMTP overflow)

เป็นความเสี่ยงที่จะก่อปัญหาให้กับระบบเมล์เซิร์ฟเวอร์ทำงานช้าลง เพราะแฮกเกอร์จะพยายามส่งเมล์จากภายนอกที่มีขนาดใหญ่มากๆเข้ามาที่เมล์เซิร์ฟเวอร์ภายในองค์กร

- เว็บ แอทแทค (Web Attacks)

เป็นความเสี่ยงที่เกิดจากการเข้ามาขโมยข้อมูลรหัสผ่านที่เมล์เซิร์ฟเวอร์โดยใช้ช่องทางพอร์ท 80 มาติดตั้งระบบแฮนด์-คราฟท์ ยูอาร์แอล (hand-crafted URL) หลังจากนั้นก็จะดึงข้อมูลจากไฟล์ "/etc/shadow" ที่เก็บรหัสผ่านทั้งหมดไว้ไปใช้งาน

- ไวรัส "โค้ดเรด" (Virus "CodeRed")

เป็นความเสี่ยงที่ทำให้การใช้งานอินเทอร์เน็ตช้าลงเพราะไวรัสโค้ดเรดจะส่งการร้องขอเรียกดูเว็บเพจที่อยู่ในเครือข่ายอินเทอร์เน็ตที่ไม่มีระบบป้องกันไวรัสโค้ดเรดอยู่ โดยที่จะมีการร้องขอจำนวนมากจนแบนด์วิดท์ (Bandwidth) ที่ใช้งานอยู่เต็ม

2) ความเสี่ยงจากบุคคลภายใน

เป็นความเสี่ยงที่เป็นอันตรายมากกว่าบุคคลภายนอกเพราะว่าเครือข่ายภายในจะไม่มีระบบรักษาความปลอดภัย ทำให้คนภายในด้วยกันสามารถที่จะทำการแฮกกันได้ง่ายกว่าบุคคลภายนอก ตัวอย่างที่พบเห็นได้ชัดเจน เช่น

- แบ็คดอร์ (Backdoor)

เป็นความเสี่ยงที่เกิดจากการแอบนำเอาซอฟต์แวร์ประเภทแบ็คดอร์เช่น เน็ทบัสโปร (Net bus Pro) ไปติดตั้งที่เครื่องที่ต้องการขโมยข้อมูล เมื่อติดตั้งเสร็จก็สามารถที่จะดึงข้อมูลที่อยู่บนเครื่องนั้นๆได้ และยังเฝ้าดูการใช้งานของเจ้าของเครื่องที่ถูกติดตั้งซอฟต์แวร์ดังกล่าวได้ เช่น การคีย์รหัสผ่านสำหรับแอปพลิเคชันที่เป็นความลับได้

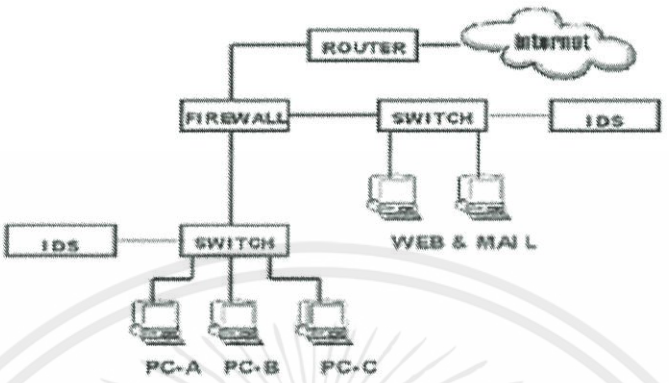
- ความพยายามเข้าไปใช้งานในระบบงานที่ไม่ได้อนุญาตไว้

เช่น ความพยายามเทลเน็ต, เอฟทีพีหรือพยายามติดต่อกับฐานข้อมูลของบริษัท

- ไวรัส (Virus)

ถึงแม้ว่าจะมีการติดตั้งระบบป้องกันไวรัสแต่ในบางครั้งอาจจะมีการแพร่กระจายไวรัสโดยไม่ตั้งใจเช่น การนำเอาอุปกรณ์คอมพิวเตอร์เข้ามาทดสอบหรือนำเอาเครื่องคอมพิวเตอร์จากภายนอกมาใช้ในองค์กร ซึ่งอาจก่อให้เกิดความเสียหายแก่ระบบงานได้

2.4.1.9 สิ่งที่ได้รับหลังการติดตั้งระบบตรวจจับผู้บุกรุก (ไอดีเอส (IDS))



รูปที่ 2.9 ทิปปิคอล เน็ตเวิร์ก วิท ไฟร์วอลล์ (Typical Network with Firewall)

เมื่อติดตั้งระบบไอดีเอส ระบบจะสามารถหยุดผู้บุกรุกที่มีพฤติกรรมที่เป็นอันตรายและสร้างความเสียหายแก่ระบบงานดังต่อไปนี้

1) ป้องกันการบุกรุกจากแฮกเกอร์ที่เข้ามาทางอินเทอร์เน็ตและผ่านการตรวจเช็คจากไฟร์วอลล์เช่น การจู่โจมและปลอมแปลงที่เข้าทางพอร์ท 80 (เว็บ),พอร์ท 25 (เมล์), พอร์ท 21 (เอฟทีพี) เมื่อไอดีเอสป้องกันการบุกรุกแล้วยังเสนอแนวทางการแก้ไขว่าองค์กรควรที่จะปรับปรุงอะไรบ้าง เช่น ซีเคียวริตี้ แพทช์ (Security Patch) ของ O/S หรือแก้ไขคอนฟิกไฟล์ (Configure File) ต่างๆเพื่อไม่ให้แฮกเกอร์ใช้วิธีการเดิมเข้ามาทำลายระบบได้

2) เฝ้าดูพฤติกรรมของบุคคลภายในว่ามีการใช้งานอะไรบ้างที่จะเป็นความเสี่ยงต่อระบบงานที่มีอยู่ โดยจะแจ้งเตือนและหยุดการทำงานของบุคคลนั้นๆ พร้อมทั้งยังแจ้งว่าใครเป็นผู้กระทำ เมื่อไร อย่างไร และใครเป็นผู้ถูกกระทำ ความเสี่ยงที่สามารถตรวจจับและป้องกันได้เช่น แบ็คดอร์, ไวรัสการใช้คำสั่งเทลเน็ต และเอฟทีพีโดยไม่ได้รับอนุญาต เป็นต้น

3) ความเสี่ยงที่เกิดขึ้นกับองค์กรนั้น บางส่วนสามารถที่จะรู้ที่มาและหาวิธีการล่วงหน้าในการป้องกันได้ แต่ถ้าไม่สามารถที่จะรู้ที่มาของความเสี่ยงตลอดจนวิธีการในการป้องกันได้ จำเป็นต้องใช้ระบบไอดีเอสเพื่อช่วยให้เราสามารถทำการตรวจจับและดึงรายละเอียดของแพ็คเกตมาทำการวิเคราะห์เพื่อดูพฤติกรรมว่าน่าจะเป็นความเสี่ยงต่อระบบงานหรือไม่เช่น แพ็คเกตที่ทำงานแบบเดิมๆและบ่อยมากๆ และมีผลต่อเครือข่ายหรือระบบงาน เราก็นำเอาวิธีการของการกระทำนั้นๆมากำหนดเป็นเงื่อนไขในการตรวจเช็คและป้องกันได้เอง

2.4.2 มัลแวร์ (Malware)

เมื่อเราพูดถึงไวรัส (virus), เวิร์ม (worm), โทรจัน (Trojan) เรามักจะหมายถึงซอฟต์แวร์ที่ส่งผลเสียต่างๆ โดยคำที่จะคลอบคลุมทั้งหมดคือคำว่า มัลแวร์ (malware) โดยมัลแวร์นั้นจะคลอบคลุมถึงทุกๆโปรแกรมที่ส่งผลเสียต่างๆเช่น ทำให้ไม่สามารถเข้าถึง, เปลี่ยนแปลงแก้ไข, ลบหรือสร้างช่องทางให้ผู้ไม่หวังดีเข้าถึงข้อมูลอิเล็กทรอนิคส์ได้

มัลแวร์จะมีฟังก์ชั่นในการทำลายเพย์โหลดที่แต่ละฟังก์ชั่นจะส่งผลเสียแตกต่างกัน บางมัลแวร์ก็เพียงแค่ก่อกวนเราด้วยวิธีต่างๆ จนไปถึงการขโมยข้อมูลที่สำคัญหรือแม้กระทั่งการลบ ฮาร์ดไดร์ว (hard drive) มัลแวร์นั้นแบ่งได้เป็น 3 กลุ่มใหญ่ๆ ได้แก่ โทรจัน ฮอร์ส (Trojan horses), เวิร์ม และไวรัส นอกจากนั้นมัลแวร์ยังอาจจะรวมไปถึงฮอก (hoax) เช่นกัน

2.4.2.1 โทรจัน ฮอร์ส (Trojan Horses)

โทรจัน ฮอร์ส เป็นมัลแวร์ประเภทหนึ่งซึ่งแตกต่างกับเวิร์มและไวรัสที่มันจะไม่คัดลองตัวเอง โดยโทรจันนั้นมีจุดหมายหลักอยู่ที่การส่งและรับข้อมูล โดยเฉพาะอย่างยิ่งคำสั่งต่างๆที่ส่งมาจากระบบอื่นๆผ่านพอร์ท โดยพอร์ทที่ใช้อาจจะเป็นพอร์ทที่รู้จักกันดีเช่น เอชทีทีพี พอร์ท80 (http port 80) หรือเป็นพอร์ทหมายเลขใดก็ได้อย่างเช่น พอร์ท7777

โทรจันนั้นมักจะถูกแปลงโฉมให้ยูสเซอร์คิดว่าโปรแกรมโทรจันนั้นเป็นแอปพลิเคชันธรรมดาที่ไม่มีพิษภัยและทำให้ยูสเซอร์สั่งรันโปรแกรมนั้น โดยโทรจันนั้นอาจแฝงตัวมากับสกรีนเซฟเวอร์หรือเกม (game) ต่างๆที่รับมาจากทางอีเมล

ไอคอน (Icon) ของโทรจันนั้นจะถูกแฮกเกอร์แก้ไขไอคอนทำให้ดูเหมือนโปรแกรมธรรมดาหรืออาจจะปลอมไอคอนให้ดูเหมือน ไมโครซอฟต์ เวิร์ด ด็อคคิวเมนท์ (Microsoft word document) หรือ เท็กซ์ไฟล์ (text file)

จุดเด่นของโทรจันนั้นในขั้นตอนแรกมันจะต้องถูกเอ็กซีคิวต์โดยยูสเซอร์และต่อมาก็จะรอส่งหรือรับข้อมูลที่อาจจะเป็นคำสั่งต่างๆที่มาจากระบบของผู้โจมตี

บางครั้งโทรจันก็เข้าไปแฝงตัวกับแอปพลิเคชันอื่นๆ โดยแอปพลิเคชันในที่นี้อาจจะเป็นแฟลช เกม (flash game),แพทช์ (Patch) ของระบบปฏิบัติการหรือแม้กระทั่ง แอนติไวรัส (Antivirus) โดยไฟล์ของแอปพลิเคชันเหล่านั้นส่วนหนึ่งเป็นไฟล์ธรรมดาอีกส่วนหนึ่งก็จะเป็นไฟล์โทรจัน ซึ่งจะถูกสั่งให้ทำงานโดยมีสิทธิเดียวกับยูสเซอร์ที่สั่งรันมัน โดยการส่งข้อมูลของโทรจันนั้นไม่จำเป็นต้องทำตอนที่ออนไลน์ (online) เสมอไป แต่ยังสามารถสั่งการขณะที่ออฟไลน์ (offline) ได้เช่นกัน ตัวอย่างเช่น สั่งการผ่านทางอีเมล, เอชทีทีพี ยูอาร์แอล (HTTP URL)

วิธีการปกปิดอำพรางตัวเองของโทรจันนั้นมีได้มากมาย โทรจันสามารถซ่อนตัวในคอมมานด์ ไลน์ (Command line) ที่เป็นคำสั่งแอดมิน (admin) ได้ โดยในระบบปฏิบัติการยูนิกซ์มักใช้ เช่น passwd,ps หรือ เน็ตสแตท (netstat)

พฤติกรรมโดยทั่วไปอย่างหนึ่งของโทรจันคือการทำ ออโต สตาร์ท (Auto start) ทุกครั้งที่ระบบเกิดการรีบูท (Reboot) ซึ่งวิธีออโตสตาร์ทก็มีได้หลายวิธี โดยด้านล่างเป็นรีจิสติคีย์ที่จะถูกแก้ไขเพื่อให้โทรจันสามารถออโตสตาร์ทได้

1) HKLM\Software\Microsoft\Windows\Current Version\Run
2) HKLM\Software\Microsoft\Windows\Current Version\Runonce
3) HKLM\Software\Microsoft\Windows\Current Version\RunServices
4) HKLM\Software\Microsoft\Windows\Current Version\RunServicesOnce
5) HKLU\Software\Microsoft\Windows\Current Version\Run
6) HKLU\Software\Microsoft\Windows\Current Version\RunOnce

โดยโทรจันสามารถแบ่งออกได้เป็น 6 ประเภท อันได้แก่

1) รีโมท แอคเซส โทรจัน (Remote Access Trojan)

โทรจันประเภทนี้จะทำให้แฮกเกอร์สามารถเข้ามาควบคุมระบบของเป้าหมายได้อย่างเต็มรูปแบบ โดยขั้นตอนก็คือ โปรแกรมที่เป็นเซิร์ฟเวอร์จะส่งไปยังเครื่องเป้าหมายและตัวลิสเทนเนอร์ (listener) ที่เป็นไคลเอนต์ (client) นั้นจะอยู่ที่ระบบของ แฮกเกอร์ และหลังจากเซิร์ฟเวอร์ทำงานแล้วก็จะสร้างคอนเนคชันกลับมายังไคลเอนต์ ผ่านทางพอร์ทที่ได้ตั้งค่าไว้ ซึ่งโดยส่วนมากแล้ว โทรจันทั่วไปจะเป็นประเภทดังกล่าว

2) ดาต้า เซนด์ดิง โทรจัน (Data Sending Trojans)

โทรจันประเภทนี้จะส่งข้อมูลต่างๆกลับมายังระบบของแฮกเกอร์ โดยตัวอย่างของข้อมูลที่ส่งมาเช่น พาสเวิร์ด (Password) , คุกกี้ (Cookies) หรือ คีย์ สโตรค (Key Stroke)

3) เดสตรัคทีฟ โทรจัน (Destructive Trojans)

โทรจันประเภทนี้มีเป้าหมายอยู่ที่การทำลาย เช่น การลบไฟล์ ทำให้ระบบปฏิบัติการทำงานผิดพลาด ทำให้ระบบแครช (crash) หรือแม้กระทั่งทำลายระบบเกี่ยวกับซีเคียวริตี้ (security) อย่างเช่น แอนตีไวรัสและไฟร์วอลล์

4) ดีดีโอเอส แอทแทค โทรจัน (DDos Attack Trojans)

โทรจันประเภทนี้จะทำให้เครื่องเป้าหมายกลายเป็นซอมบี้ (Zombie) ที่จะรอรับ คำสั่งจากดีดีโอเอส เซิร์ฟเวอร์ (DDos Server) ในอินเทอร์เน็ต โดยการโจมตีจะเริ่มจากการทำให้มีเครื่องที่ติดเชื้อและกลายเป็นซอมบี้มากมายหลายเครื่อง หลังจากนั้นดีดีโอเอส เซิร์ฟเวอร์ก็จะส่งคำสั่งมายังระบบที่ติดเชื่ออยู่ทุกระบบให้ฟลัดแพ็คเกต (flood packet) ไปยังเซิร์ฟเวอร์เป้าหมายทำให้เซิร์ฟเวอร์เป้าหมายไม่สามารถตอบสนองและทำงานต่อได้

5) ซีเคียวริตี้ ซอฟต์แวร์ ดิสเอเบลอร์ โทรจัน (Security Software Disabler Trojan)

โทรจันประเภทนี้มีเป้าหมายหลักอยู่ที่การทำลายระบบเกี่ยวกับความปลอดภัยอย่างเช่น แอนติไวรัสและไฟร์วอลล์ หรือทำให้ระบบทำงานอย่างผิดพลาด เพื่อปูพรมให้กับการโจมตีอื่นๆที่จะเกิดขึ้น

6) พร็อกซี โทรจัน (Proxy Trojans)

ในกรณีที่ต้องการจะปกปิดร่องรอยของแฮกเกอร์นั้น แฮกเกอร์จะส่งคำสั่งผ่าน ตัวกลางที่เป็นระบบๆหนึ่ง และเมื่อโทรจันนั้นถูกจับได้ เรคคอร์ด (Record) จะแสดงว่าคนที่ทำคือตัวของระบบเองที่ไม่ใช่ของแฮกเกอร์

2.4.2.2 ไวรัส (Viruses)

ไวรัสมีเป้าหมายคือ คัดลอกตัวเองและแพร่พันธ์ไปยังเครื่องอื่นๆ และเพื่อที่จะทำแบบนั้นได้มันจำเป็นที่จะต้องเกาะติดไปกับไฟล์หรือฝังไปกับบูท เซคเตอร์ (boot sector) หรือว่าพวกดาต้าแคร์เรียร์ (data carrier) เช่น รีมูฟเอเบิลสโตรเรจ (Removable Storage) ต่างๆ บ่อยครั้งที่มันจะแฝงตัวไปกับแผ่นดิสก์ (Disk) ที่เก็บซอฟต์แวร์ที่ผิดกฎหมาย หรือทางระบบเน็ตเวิร์กและทางอีเมล ไวรัสสามารถฝังตัวเองเข้าสู่ส่วนต่างๆของระบบปฏิบัติการและสามารถทำงานได้หลากหลายฟังก์ชั่นหลากหลายแชนแนล (channel)

โดยไวรัสสามารถแบ่งออกได้เป็น 4 ประเภท อันได้แก่

1) บูท เซคเตอร์ ไวรัส (Boot sector viruses)

บูท เซคเตอร์ (Boot sector) หรือ เอ็มบีอาร์ ไวรัส (MBR viruses (master boot record viruses)) หรือเรียกกันในชื่อ เมมโมรี่-เรสซิเดนท์ (memory-resident) นั้นจะไปฝังตัวอยู่ที่ด้านหน้าของบูทเซคเตอร์เพื่อที่จะสามาถอ่านโค้ดไวรัสก่อนที่จะเข้ามาอ่านที่บูทเซคเตอร์ เมื่อคอมพิวเตอร์ใช้การบูทจากบูทเซคเตอร์นั้น ดังนั้นไวรัสประเภทนี้จะฝังตัวในระบบโดยที่ไม่สามารถดักจับได้หลังจากนั้นมันก็จะถูกรันเมื่อฮาร์สดิสก์บูทขึ้นมา

บ่อยครั้งที่โค้ดของไวรัสนั้นจะยังคงฝังอยู่ในเมมโมรี่ หลังจากที่ทำให้ระบบติดเชื้อแล้ว โดยไวรัสนี้จะมีวิธีส่งต่อไปยังเครื่องอื่นๆผ่านทางดิสก์เกตต์ (diskette) โดยจะไปฝังตัวอยู่ในดิสก์เกตต์จังหวะที่สั่งฟอร์แมท (format) อย่างไรก็ตามบูทเซคเตอร์ไวรัสนั้นไม่ได้แพร่กระจายตนเองด้วยวิธีการฟอร์แมทเท่านั้น ไวรัสสามารถแพร่กระจายด้วยคำสั่งดีไออาร์ (DIR) ของดอส (DOS) เช่นกัน การทำลายล้างของบูทเซคเตอร์ไวรัสนั้นขึ้นอยู่กับฟังก์ชั่นของมันว่าจะแค่ก่อกวนหรือว่าเป็นภัยใหญ่หลวง บูทเซคเตอร์ไวรัสที่เก่าแก่และรู้จักกันดีที่สุดก็คือ ฟอร์ม (Form)

2) File viruses

ไวรัสหลายตัวที่มักจะซ่อนอยู่ในไฟล์ต่างๆ ซึ่งทำได้โดยการลบหรือเขียนไฟล์ทับเข้าไป หรือแม้แต่เกาะไปกับไฟล์นั้นโดยไฟล์นั้นต้องทำงานได้อยู่ และเมื่อโปรแกรมที่โดนไวรัสฝังตัวถูกเรียกให้ทำงานแล้วโค้ดของไวรัส (ส่วนมากเป็นภาษาแอสเซมบลี (assembly)) ก็จะทำงานก่อน หลังจากนั้นโปรแกรมหลักจึงจะทำงาน

3) Macro Viruses

มาโคร ไวรัสจะฝังตัวอยู่ในไฟล์แต่ไม่ได้ฝังอยู่ใน .exe ไฟล์ นอกจากนั้นมาโครไวรัสไม่ได้ถูกเขียนด้วยแอสเซมบลี แต่ถูกเขียนในภาษามาโคร ตัวอย่างเช่น ภาษาวิชชวล เบสิก (visual basic) โดยไวรัสนี้จะใช้มาโคร อินเตอร์พรีทเตอร์ (macro interpreter) ที่จะมีใน ไมโครซอฟต์เวิร์ด (Microsoft Word), เอ็กซ์เซล (Excel), แอกเซส (Access), พาวเวอร์พอยท์ (PowerPoint) ที่จะสั่งให้ทำงาน

4) สเตรท์ช ไวรัส (Stealth viruses) และ รูทคิทส์ (rootkits)

สเตรท์ช (Stealth) หรือแคมมะฟลาจ (camouflage) ไวรัส จะมีความสามารถในการพรางตัวเพื่อหนีการตรวจจับของแอนติไวรัส โดยมันจำให้แอนติไวรัส หรือยูสเซอร์ไม่สามารถมองเห็นไฟล์นั้นได้

2.4.2.3 เวิร์ม (Worms)

เวิร์ม มีการเรียกเป็นภาษาไทยว่า "หนอนอินเทอร์เน็ต" โดยมีความแตกต่างจากไวรัสตรงที่ไวรัสจะแพร่กระจายตัวโดยจำเป็นต้องฝังไปกับโปรแกรมอื่นๆ แต่เวิร์มสามารถแพร่กระจายได้ด้วยตัวเอง ทำให้คอมพิวเตอร์ส่วนตัวและในระบบเครือข่ายเกิดความเสียหายขึ้น เวิร์มในปัจจุบันนี้มีหลากหลายมาก มีการแพร่กระจายตัวได้อย่างรวดเร็ว ทั้งนี้เนื่องจากเวิร์มจะสามารถแพร่กระจายผ่าน

ทางอีเมลได้ ไม่ว่าจะเป็นเอาท์ลุค เอ็กเพรส(Outlook Express) หรือไมโครซอฟต์ เอาท์ลุค (Microsoft Outlook)

การป้องกันอย่างหนึ่งสำหรับมัลแวร์ประเภทนี้ คือการอัพเดทโปรแกรมให้ทันสมัยอยู่เสมอ ดังรายละเอียดด้านล่างนี้

1) ความเสียหายที่อาจเกิดขึ้น
   - เครื่องคอมพิวเตอร์ทำงานช้าลง
   - เครื่องคอมพิวเตอร์ไม่สามารถทำงานได้
   - ไม่สามารถติดต่อระบบเครือข่ายได้
   - ไม่สามารถทำงานในระบบอินเทอร์เน็ตได้

2) การอัพเดทโปรแกรมสำหรับการใช้งานอินเทอร์เน็ตเอ็กซ์โพลเรอร์
   - อัพเดทไปยัง Internet Explorer 5.01 SP2
   - อัพเดทไปยัง IE 5.5 SP2
   - อัพเดทไปยัง IE 6.0

3) ตัวอย่างรายชื่อของวอร์ม
   - WORM_KLEZ.H
   - WORM_YAHA.K
   - WORM_OPASERV.E
   - WORM_KWBOT.C
   - WORM_FRETHEM.M

นอกจากนี้เรายังมีวิธีการป้องกันเบื้องต้นก็คือ การติดตั้งโปรแกรมตรวจสอบไวรัส แบบเรียลไทม์ (Realtime) หมายถึงตรวจสอบอีเมลทุกครั้งที่เข้ามา ตรวจสอบเว็บไซท์ที่มีการแวะเวียนเข้าไปแบบอัตโนมัติ เป็นต้น และที่สำคัญควรหลีกเลี่ยงการเปิดเมล์ที่เราไม่รู้จักหรือไม่แน่ใจ

2.4.3 โปรแกรมแอนติไวรัส (Antivirus) และวิธีการดักจับ

โปรแกรมป้องกันไวรัส หรือ แอนติไวรัส เป็นโปรแกรมที่สร้างขึ้นเพื่อคอยตรวจจับ ป้องกันและกำจัดโปรแกรมคุกคามทางคอมพิวเตอร์หรือมัลแวร์

โปรแกรมป้องกันไวรัสมี 2 แบบหลักๆ อันได้แก่

1) แอนติไวรัส (Anti-Virus) เป็นโปรแกรมป้องกันไวรัสทั่วๆไป จะค้นหาและทำลายไวรัสในคอมพิวเตอร์

2) แอนติสปายแวร์ (Anti-Spyware) เป็นโปรแกรมป้องกันการโจรกรรมข้อมูล จากไวรัส สปายแวร์ และจากแฮ็กเกอร์ รวมถึงการกำจัดแอดซ์แวร์ (Adsware) ซึ่งเป็นป๊อปอัพ (Pop up) โฆษณาอีกด้วย

โปรแกรมป้องกันไวรัสจะค้นหาและทำลายไวรัสที่ไฟล์โดยตรง แต่ในทุกๆวันจะมีไวรัสชนิดใหม่เกิดขึ้นมาเสมอ ทำให้เราต้องอัปเดทโปรแกรมป้องกันไวรัสตลอดเวลาเพื่อให้คอมพิวเตอร์ของเราปลอดภัย โดยแอนติไวรัสจะมีหลายรูปแบบตามแต่แต่ละบริษัท และแต่ละบริษัทจะมีการอัปเดทและการป้องกันไม่เหมือนกัน แต่ในคอมพิวเตอร์เครื่องเดียวไม่ควรมีแอนติไวรัส 2 โปรแกรม เพราะจะทำให้โปรแกรมขัดแย้งกันเองจนไม่สามารถใช้งานได้

2.4.3.1 วิธีที่ใช้ในการตรวจหาไวรัส

โปรแกรมป้องกันไวรัสมีวิธีค้นหาไวรัสอยู่หลายวิธีดังนี้

1) ตรวจสอบไวรัส ซิกเนเจอร์ (Virus signature)

ไวรัสซิกเนเจอร์ คือ สัญลักษณ์ของไวรัส ซึ่งไวรัสแต่ละตัวจะมีสัญลักษณ์ที่แตกต่างกันออกไป เปรียบเหมือนลายเซ็นของคนทั่วไปที่ล้วนแตกต่างกันออกไป โดยหลักการทำงานนั้น โปรแกรมป้องกันไวรัสจะมีการตรวจสอบไฟล์ว่ามีรหัสเหมือนกับไวรัสซิกเนเจอร์หรือไม่ ซึ่งหากใช่นั้น หมายถึงว่าไฟล์ตัวนั้นคือไวรัส

โปรแกรมป้องกันไวรัส จึงควรต้องหมั่นอัปเดทอยู่เป็นประจำ เพื่อให้การป้องกันไวรัสเป็นไปได้อย่างทั่วถึง และวิธีนี้เป็นวิธีที่ใช้กันแพร่หลายในปัจจุบัน

2) ตรวจสอบ คอมเปลี่ยนแปลงของข้อมูล

เป็นการตรวจหาค่าพิเศษที่เรียกว่า เช็คซัม (Checksum) ของไฟล์ ซึ่งถ้าหากเกิดการเปลี่ยนแปลงในตัวไฟล์ ซึ่งอาจเกิดจากไวรัส ค่านี้ก็จะเปลี่ยนแปลง ซึ่งข้อดีก็คือจะตรวจจับไวรัสชนิดใหม่ๆได้ แต่ปัญหาคือต้องแน่ใจว่าตัวเครื่องนั้นไม่มีการติดเชื้อ

3) ตรวจสอบการกระทำแปลกปลอม

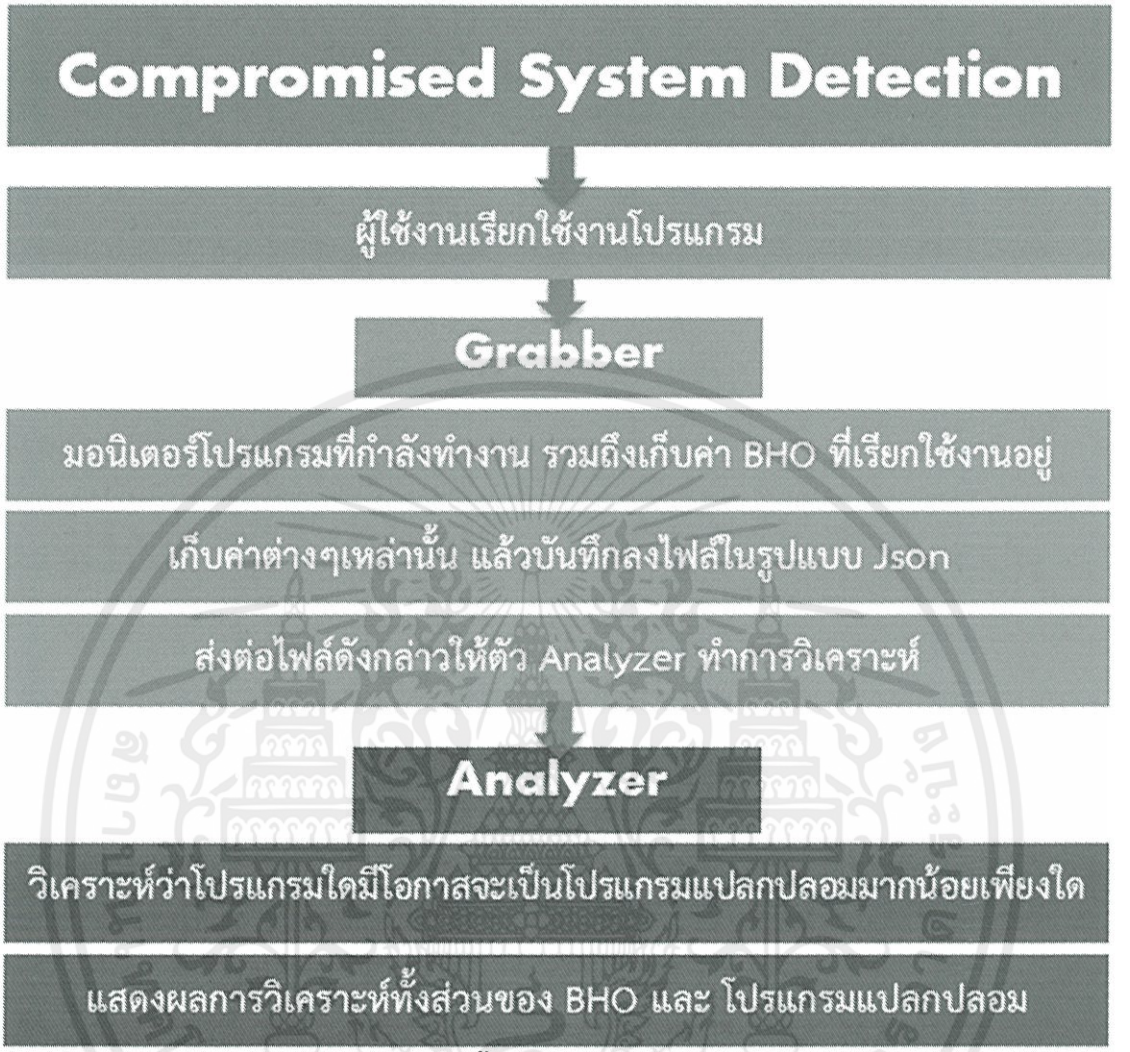
คอยตรวจสอบการกระทำที่แปลกปลอมจากไวรัสต่างๆอาทิเช่น พยายามทำลาย เปลี่ยนแปลง หรือแก้ไขข้อมูลระบบโดยไม่ได้รับอนุญาต พยายามจดจำตัวเองในระบบรูท (root) พยายามดาวน์โหลด และอัปโหลดข้อมูลและไฟล์ต่างๆ

บทที่ 3
การออกแบบและพัฒนาซอฟต์แวร์

ในการพัฒนาซอฟต์แวร์จำเป็นต้องมีการวางแผนและออกแบบซอฟต์แวร์ไว้ล่วงหน้า ซึ่งในบทนี้จะอธิบายถึงแนวความคิดในการพัฒนาซอฟต์แวร์ เครื่องมือและภาษาที่ใช้ในการพัฒนา รายละเอียดของซอฟต์แวร์เชิงเทคนิค (Software Specification) และโครงสร้างหลักของซอฟต์แวร์

3.1 แนวคิดในการพัฒนา

โครงงานชิ้นนี้จะใช้แนวคิดคล้ายๆกับโปรแกรม Hijack this ซึ่งมีโปรแกรมทั้งหมด 2 โปรแกรมคือ แกรบเบอร์ (Grabber) (ตัวดึงข้อมูลจากHost) และ อนาไลเซอร์ (Analyzer) (ตัววิเคราะห์ข้อมูล) แต่โครงงานนี้จะเพิ่มโปรแกรมมาอีกส่วนหนึ่งคือ มอนิเตอร์ (Monitor) ซึ่งเป็นตัวจับตาเฝ้าดูการทำงานของโปรแกรม โดยเครื่องมือที่ใช้ในการพัฒนาคือ Visual Studio 2012 ซึ่งภาษาที่ใช้ในการพัฒนาในครั้งนี้คือ C++



รูปที่ 3.1 ภาพรมขั้นตอนการทำงานของโปรแกรม

3.1.1 แนวคิดในการพัฒนาแกรบเบอร์ (Grabber)

สำรับแนวคิดในการพัฒนาตัว Grabber นี้จะพัฒนาให้ตัวโปรแกรมสามารถทำหน้าที่ในการดึงข้อมูลต่างๆจาก Host โดยข้อมูลที่ดึงได้จะมีดังนี้

1) รายละเอียดของโพรเซสที่ทำงานอยู่ในเครื่อง (ชื่อ, Path, Process Id)
2) การสร้างการเชื่อมต่อ (Connection) ของโพรเซส
3) โพรเซสที่มีคุณสมบัติออโตสตาร์ท (Autostart)
4) พาทซ์ (Path) ของตัว Uninstall ทั้งหมดที่มีอยู่ในเครื่อง
5) Browser helper object
6) พฤติกรรมแปลกปลอมของแต่ละโปรแกรม

โดยนอกจากจะทำหน้าที่ดึงข้อมูลจากโฮสต์แล้วมันยังทำหน้าที่ในการบันทึกข้อมูลที่เก็บได้ลงในไฟล์ซึ่งโครงสร้างของข้อมูลนั้นจะเป็นโครงสร้างแบบ JSON โดยต่อไปจะเป็นการอธิบายลำดับการทำงานของตัวแกรบเบอร์โดยจะอิงตาม Source Code

3.1.2 ลำดับการทำงานของ Grabber

มีลำดับขั้นตอนการทำงาน ดังนี้

1) Class EntryGrabber เรียกฟังก์ชั่น Grab Autostart เพื่อดึงรายชื่อและPathของ Program ที่มีคุณสมบัติ Auto Start เก็บลงเป็น รูปแบบ JSON ซึ่งมีตัวแทนคือตัวแปร autoStartGrabbed .jsonformat ซึ่งเป็นออบเจกต์ของคลาส JSON::value โดย jsonformat นั้นมีโครงสร้างดังนี้

```
{
    "type" : "autoStart",
    //เป็นListของโปรแกรมที่เป็นAutostart
    "autoStartList" : [
            {
              "autoStartupName" : "IgfxTray",
              "autoStartupPath" : "C:\\Windows\\system32\\igfxtray.exe",
              "baseKey" : "LOCAL_MACHINE_64",
              "subKey" : "Software\\Microsoft\\Windows\\CurrentVersion\\Run"
            }
    ]
}
```

โดยวิธีดึงชื่อโพรเซสที่เป็น Autostart นั้นจะใช้วิธี lookup จากรีจิสทรีคีย์ที่เกี่ยวข้อง

2) Class EntryGrabber เรียกฟังก์ชั่น Grab Uninstall เพื่อดึงรายชื่อและพาทช์ของตัว Uninstall ทั้งหมดที่มีอยู่ในระบบ และเก็บลงใน รูปแบบ JSON ซึ่งมีตัวแทนคือตัวแปร uninstallGrabbed.jsonformat ซึ่งเป็นออบเจกต์ของคลาส JSON::value โดย jsonformat นั้นมีโครงสร้างดังนี้

```
{
        "type" : "uninstall",
          //เป็นListของตัวUninstall(ถอนการติดตั้ง)ที่มีอยู่ในเครื่อง
        "uninstallList" : [
          {
            "baseKey" : "LOCAL_MACHINE_64",
            "displayIcon" : "",
            "displayName" : "CDisplayEx 1.9.16",
            "installLocation" : "C:\\Program Files\\CDisplayEx",
            "publisher" : "cdisplayex.com",
            "uninstallKeyName" : "CDisplayEx_is1",
            "uninstallString" : "\"C:\\Program Files\\CDisplayEx\\unins000.exe\""
          }
        ]
}
```

โดยวิธีดึงตัวUninstallนั้นจะใช้วิธี lookup จากรีจิสทรีคีย์ที่เกี่ยวข้อง

3) Class EntryGrabber เรียกฟังก์ชั่น Grab Bho เพื่อดึงรายชื่อและ CLSID ของ BHO ที่มีอยู่ในเครื่องพร้อมทั้งนำ CLSID ของ BHO แต่ละตัวเข้าไปค้นหาใน www.systemlookup.com และบันทึก HTML ผลลัพธ์ ซึ่ง HTML นี้จะเป็นตัวบอกว่า BHO นี้เป็น BHOที่ปลอดภัยหรือไม่ และเราก็จะเอาCLSID และ HTML ที่ได้จากการ Search นี้บันทึกลงในรูปแบบ ของ JSON ซึ่งมีตัวแทนคือตัวแปร bhoGrabbed.jsonformat ซึ่งเป็นออบเจกต์ของคลาส JSON::value โดยมีโครงสร้างดังนี้

```
{
        //เป็นListของBHOที่มีอยู่ในเครื่อง
    "bhoList" : [
      {
        "Clsid" : "{0055C089-8582-441B-A0BF-17B458C2A3A8}",
        "Description" : "IDM integration (IDMIEHlprObj Class)",
        "Path" : "C:\\Program Files (x86)\\Internet Download Manager\\IDMIECC64.dll",
        "lookupDescription" :
      }
}
```


โดย LookupDescription เป็น HTML ที่ได้ไปดึงมาจากการเอา CLSID ไปค้นหาใน www.systemlookup.com

4) ClassEntryGrabber เรียกฟังก์ชั่น processDetailGrabbing, connectionGrabbing, monitorProcess เพื่อดึงค่ารายละเอียดของ process ที่ทำงานอยู่ในเครื่อง อย่างเช่น Process Id, Path ของ Process และดึงสถานะการมี Connection ของ Process หลังจากนั้นก็ทำการเฝ้าดูพฤติกรรมของโพรเซสชั่วขณะและบันทึงสิ่งที่เก็บค่าได้ลงในรูปแบบ JSON ซึ่งมีตัวแทนคือตัวแปร processDetailGrabbed.jsonformat ซึ่งเป็นออบเจกต์ของคลาส JSON::value โดยมีโครงสร้างดังนี้


```
{
    //เป็นListของProcessที่รันอยู่ในเครื่อง
  "processDetailList" : [
        {
          "behavior" : [],
          "bit" : 64,
          "canMonitor" : 6,
          "copyright" : "Copyright (C) 2006 - 2012 Sublime HQ Pty Ltd",
          "filedescription" : "Sublime Text 2",
        //เป็นListของConnectionที่processนี้สร้างขึ้น
```

```
"processConnection" : [
  {
    "localAddress" : "192.168.32.131",
    "localPort" : "63617",
    "remoteAddress" : "50.116.34.243",
    "remotePort" : "443",
    "state" : "8"
  }
],
"processId" : 1820,
"processName" : "sublime_text.exe",
"processPath" : "C:\\Program Files\\Sublime Text 2\\sublime_text.exe",
"productversion" : "2178",
"version" : "2178"
}]}
```


5) หลังจากที่ดึงค่าต่างๆได้ครบแล้วก็จะทำการนำ ตัวแปร jsonformat จากทุกๆ Class ที่ได้ผ่านการบันทึกจาก 3 ขั้นตอนที่ผ่านมา (UninstallGrabbed,bhoGrabbed,ProcessDetailGrabbed) ประกอบกันเป็น JSON ก้อนใหญ่ แล้วบันทึงลงในไฟล์ โดยใช้ฟังก์ชัน summary() ของคลาส EntryGrabber

3.1.3 แนวคิดในการพัฒนามอนิเตอร์ (Monitor)

เนื่องจากมีความต้องการที่จะเฝ้าดูการทำงานของแต่ละโพรเซส จึงทำการฝังตัวมอนิเตอร์ (Monitor) เข้าไปยังแต่ละโพรเซส และให้ตัวอินเจคเตอร์นั้นทำการส่งข้อความพฤติกรรมที่แปลกปลอมของโพรเซสกลับมายังตัวแกรบเบอร์ โดยเครื่องมือที่ใช้ในการพัฒนาคือ Visual Studio 2012 ซึ่งภาษาที่ใช้ในการพัฒนาในครั้งนี้คือ C++

ซึ่งตัวมอนิเตอร์นั้น ได้ออกแบบให้มีการมอนิเตอร์ฟังก์ชันต่างๆได้ ดังต่อไปนี้

- OpenEventLog
- ClearEventLog
- LookupPrivilegeValue
- AdjustTokenPrivilegesHooking
- ExitWindowsExHooking
- FindFirstFile
- FindNextFile
- FindCloseHooking
- GetCurrentDirectory
- SetCurrentDirectory
- CreateDirectory
- RemoveDirectory
- DeleteFile
- MoveFile
- OpenProcessHooking
- GetCurrentProcessHooking
- CloseHandleHooking
- CreateProcess
- TerminateProcessHooking
- CreateToolhelp32SnapshotHooking
- Process32First
- Process32Next
- EnumProcessesHooking
- GetModuleFileNameEx
- QueryFullProcessImageName
- NtQueryInformationProcessHooking
- EnumProcessModulesHooking
- GetModuleBaseName
- RegSetValueEx
- GetUserObjectInformation
- ProcessIdToSessionIdHooking

- GetCurrentThreadIdHooking
- GetThreadDesktopHooking
- CreateCompatibleBitmapHooking
- GetAsyncKeyStateHooking
- EnumWindowStations
- GetLastInputInfoHooking
- OpenWindowStation
- GetProcessWindowStationHooking
- SetProcessWindowStationHooking
- OpenDesktop
- SetThreadDesktopHooking
- SwitchDesktopHooking
- GetDCHooking
- GetDesktopWindowHooking

3.1.4 แนวคิดในการพัฒนาอนาไลเซอร์ (Analyzer)

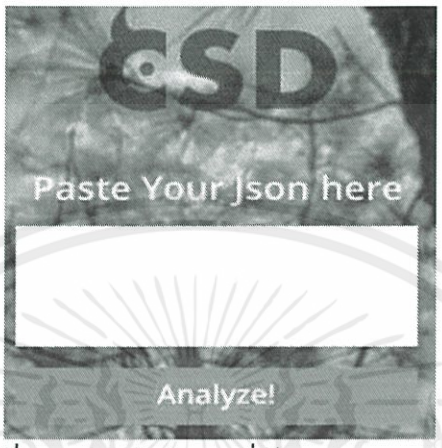
สำหรับตัวอนาไลเซอร์นั้นจะทำหน้าที่รับ JSON ที่ตัวแกรบเบอร์ได้สร้างขึ้นและนำมาวิเคราะห์และแสดงผลโดยตัวอนาไลเซอร์นี้ โดยเครื่องมือที่ใช้ในการพัฒนาคือ Sublime Text2, NodeJS, GruntJS, Less, Chrome Debugger ส่วนภาษาที่ใช้ในการพัฒนาได้แก่ CSS, HTM และ JavaScript ในการสร้างกราฟและข้อมูลของแต่ละโพรเซส โดย Userinter face ที่คิดไว้นั้นมีหน้าต่างดังนี้

3.2 อธิบายรายละเอียดของยูสเซอร์อินเตอร์เฟส (User Interface)

3.2.1 ส่วนพื้นที่รับอินพุท (Input Area)

ส่วนนี้จะมีแค่ Input Box ให้ใส่ JSON ที่แกรบ (Grab) มา ซึ่งสามารถใช้งานได้ทั้งในรูปแบบที่ตัวแกรบเบอร์ส่งไฟล์ JSON ให้กับอนาไลเซอร์เองอัตโนมัติ หรือผู้ใช้งานคัดลอกเนื้อหาในไฟล์ JSON ที่ต้องการ ซึ่งอาจเป็นการบันทึกค่ามาจากคอมพิวเตอร์เครื่องอื่น หรือกล่าวคือสามารถใช้ส่วนของตัวแกรบเบอร์และอนาไลเซอร์แยกกันได้ ถ้าหากผู้ใช้งานเลือกใช้ในรูปแบบที่ต้องการคัดลอกเนื้อหาในไฟล์ JSON เอง เมื่อคัดลอกและวางลงบนส่วนของพื้นที่รับอินพุทแล้ว ผู้ใช้จะต้องกดปุ่ม Analyze ที่

นำ Json นั้นไปวิเคราะห์และสร้างเป็นกราฟของโพรเซสและรายละเอียดของ BHO โดยในการกด Analyze นั้นได้มีการวิเคราะห์ด้วยว่ารูปแบบของ Grab_log.json นั้นถูกต้องหรือไม่



รูปที่ 3.2 ส่วนรับอินพุทที่เป็นรูปแบบ JSON

นอกจากนั้นจะมีลิงค์ให้กดดาวน์โหลดตัวแกรบเบอร์ในหน้าเวปไซต์ด้วย

3.2.2 ส่วนของรายละเอียดของBHO

ส่วนนี้จะเป็นข้อมูลของ BHO โดยจะแสดงรายละเอียดดังนี้

รูปที่ 3.3 ส่วนแสดงผลในส่วนของการวิเคราะห์ BHO

ตารางที่ 3.1 รายละเอียดของส่วนแสดงผลของยูสเซอร์อินเตอร์เฟส

ชื่อองค์ประกอบ	รายละเอียด
ชื่อ BHO	แสดงชื่อของแต่ละ BHO ที่ตรวจพบ
สถานะของ BHO	แสดงสถานะของBHOว่าปลอดภัยหรือไม่โดยจะใช้วิธีนำ CLSID ไปค้นหาใน Website www.systemlookup.com
CLSID	แสดงค่า CLSID ของ BHO นั้น
Filename	แสดงชื่อไฟล์ของ BHO นั้น
Path	แสดงที่อยู่ของไฟล์ BHO นั้น
Description	แสดงรายละเอียดที่เกี่ยวข้องกับ BHO นั้น

โดยในแต่ละรายละเอียดที่ได้แสดงนั้น เมื่อกดเข้าไปแล้วจะเปลี่ยนหน้าไปยังหน้าต่างของ www.systemlookup.com

รูปที่ 3.4 ส่วนแสดงผลที่เชื่อมโยงจากโปรแกรมไปยังหน้าเวปไซต์

3.2.3 ส่วนกราฟแสดงรายละเอียดของแต่ละโพรเซส (Process Graph)

รูปที่ 3.5 ส่วนแสดงผลกราฟรายละเอียดของแต่ละโพรเซส

โดยส่วนนี้จะเป็นกราฟความน่าสงสัยของโพรเซส ซึ่งหากโพรเซสนั้นต้องสงสัยหรือไม่น่าเชื่อถือ จะมีกราฟที่ยาวและสีในแต่ละสแต็ก (Stack) ของกราฟนั้น มีความหมายแตกต่างกันออกไป ดังนี้

- สีเขียว : บ่งบอกว่าโพรเซสนี้เป็นโพรเซสที่ไม่มีตัว Uninstall ทำให้โพรเซสนี้ดูน่าสงสัย
- สีแดง : บ่งบอกว่าโพรเซสนี้ทำพฤติกรรมแปลกปลอม อย่างเช่นในรูปที่ 3.5 โพรเซส keylogger.exe ได้ทำพฤติกรรม keylogger ทำให้โพรเซสนี้ดูน่าสงสัย
- สีม่วง : บ่งอบอกว่าโพรเซสนี้มีรายละเอียดบางอย่างหายไปอย่างเช่น Copyright , Version ทำให้โพรเซสนี้ดูน่าสงสัย
- สีน้ำเงิน : บ่งบอกว่าโพรเซสนี้มีการสร้างการเชื่อมต่อ (Connection) ทำให้โพรเซสนี้ดูน่าสงสัยว่าจะแอบส่งข้อมูลให้กับแฮกเกอร์ (Hacker)
- สีเหลือง : บ่งบอกว่าโพรเซสนี้เป็น Autostart คือทำงานทุกครั้งที่เปิดเครื่องทำให้โพรเซสนี้ดูน่าสงสัย

โดยเราสามารถคลิกที่ตัวบาร์ (Bar) ของกราฟได้ และจะมีรายละเอียดของโพรเซส เลื่อนออกมาจากด้านขวา

รูปที่ 3.6 ส่วนแสดงผลรายละเอียดของโพรเซส

บทที่ 4
ตัวอย่างและการทดสอบการทำงานของโปรแกรม

4.1 ทดสอบโปรแกรมบนระบบปฏิบัติการไมโครซอฟต์วินโดว์ 7

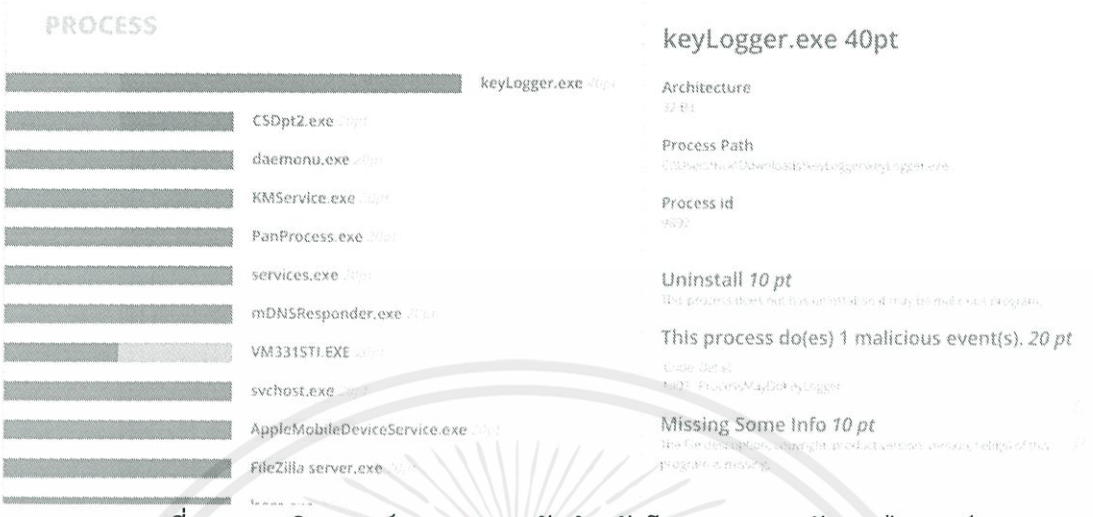
ได้ทดลองนำโปรแกรมแกรบเบอร์ (Grabber) ไปทำงานบนระบบปฏิบัติการไมโครซอฟต์วินโดว์ 7 ทั้งแบบ 32 บิต แล 64 บิต โดยได้ทดลองให้เครื่องทำการเรียกใช้โปรแกรม Key logger ซึ่งหลังจากนำสิ่งที่แกรบเบอร์อ่านค่าได้นั้น พบว่าสามารถตรวจจับโปรแกรม Key logger ได้ หรืออาจกล่าวได้ว่าสามารถแยกแยะระหว่างโปรแกรมธรรมดาและโปรแกรมแปลกปลอมออกจากกันได้ โดยตัวแกรบเบอร์นี้ยังสามารถเก็บค่าพาทช์ (Path) และตัวแปรอื่นๆที่ใช้ในการวิเคราะห์ได้อย่างครบถ้วนตามที่ได้ตั้งใจ และ นอกจากนั้น ยังสามารถตรวจจับ BHO และ Internet explorer add on ได้อีกด้วย

รูปที่ 4.1 โปรแกรมแกรบเบอร์ทำการดึงค่าต่างๆ และพฤติกรรมของ Keylogger ที่ตรวจพบในโพรเซสไอดี (Process Id) หมายเลข 9892

```
{
   "behavior" : [
      {
         "behaviorCode" : "UI01",
         "behaviorText" : "ProcessMayDoKeyLogger"
      }
   ],
   "bit" : 32,
   "canMonitor" : 1,
   "canReadInfo" : 1,
   "copyright" : "",
   "filedescription" : "",
   "hasConnection" : 0,
   "highPermission" : 0,
   "processConnection" : [],
   "processId" : 9892,
   "processName" : "keyLogger.exe",
   "processPath" : "C:\\Users\\rick\\Downloads\\keyLogger\\keyLogger.exe",
   "productversion" : "",
   "version" : ""
},
```


รูปที่ 4.2 ค่าในไฟล์ grab_log.json ที่ระบุว่าโพรเซส ทำพฤติกรรม Keylogger

รูปที่ 4.3 การวิเคราะห์ความปลอดภัยสำหรับ BHO ของตัวอนาไลเซอร์



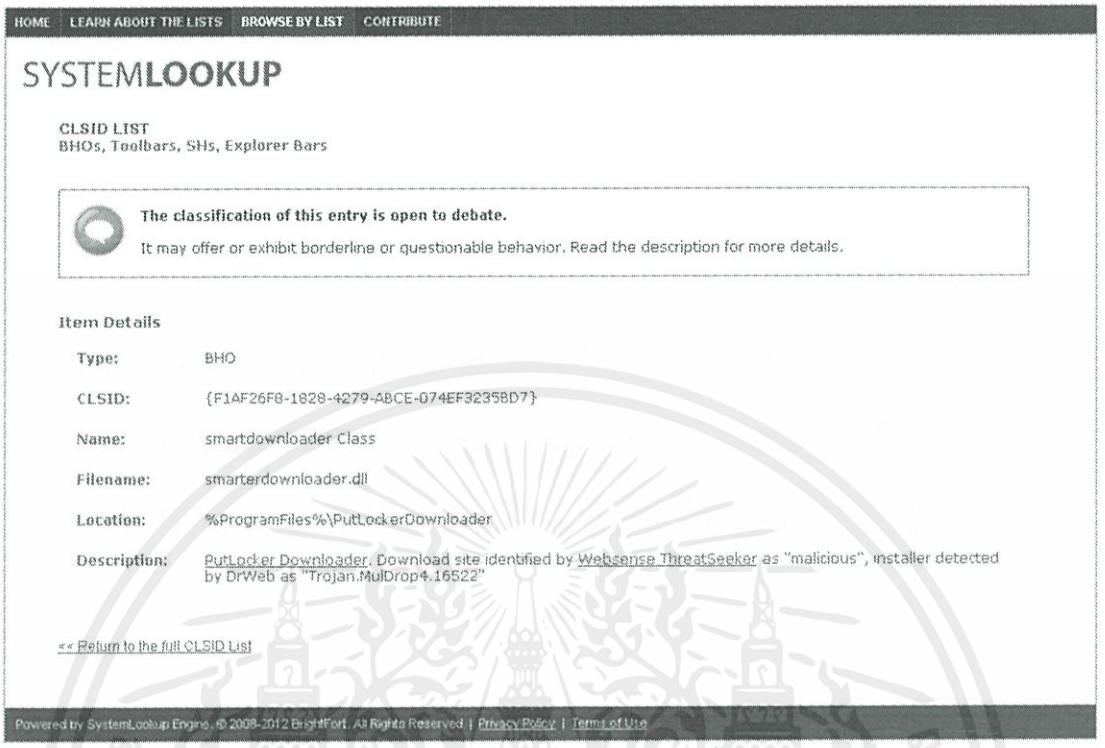
รูปที่ 4.4 การวิเคราะห์ความปลอดภัยสำหรับโปรแกรมของตัวอนาไลเซอร์

ตัวอนาไลเซอร์สามารถวิเคราะห์สิ่งที่ตัวแกรบเบอร์ได้ดึงค่ามาอย่างถูกต้อง พร้อมทั้งแสดงรายละเอียดต่างๆของแต่ละโปรแกรมอย่างครบถ้วน โดยภาพด้านบนนั้นเป็นภาพของโพรเซสชาร์ท (Process Chart) ที่ได้แสดงผลว่าโปรแกรม keylogger.exe นั้นทำพฤติกรรมแปลกปลอมประเภท Keylogger อีกทั้งโปรแกรม Keylogger นั้นยังไม่มีตัว uninstall รวมถึงมีรายละอียดบางอย่างเช่น Copyright, Product Version, Version) ขาดหายไป

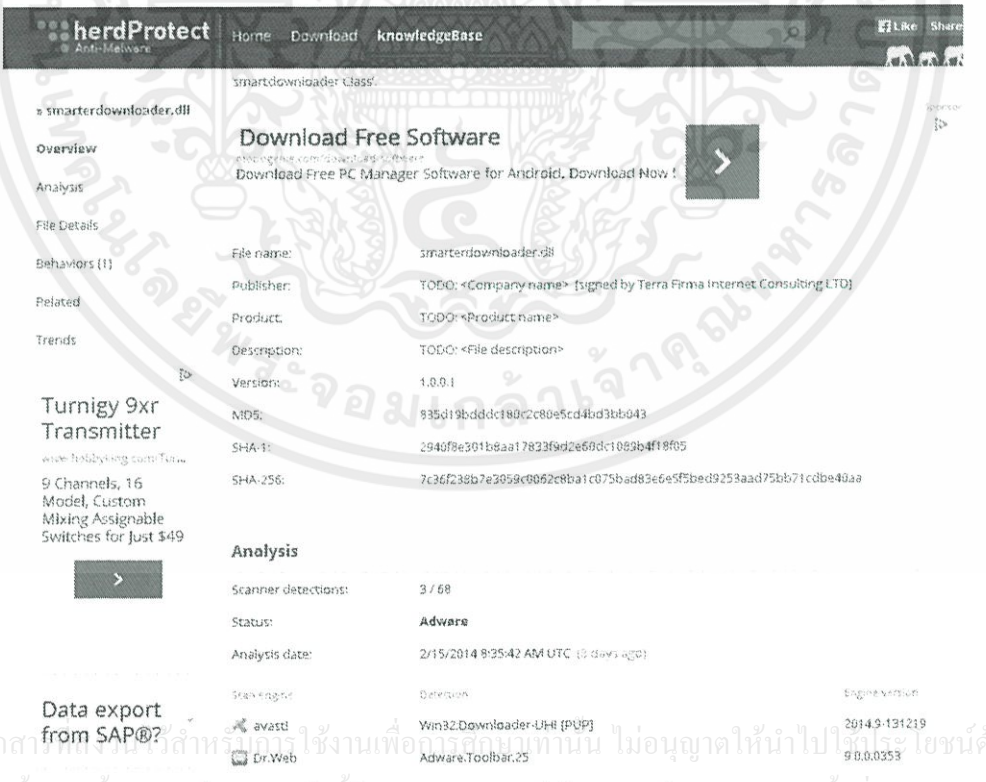
4.2 ทดสอบโปรแกรมบนระบบปฏิบัติการไมโครซอฟต์วินโดว์ 8

ทดลองนำโปรแกรมแกรบเบอร์ไปทำงานบนระบบปฏิบัติการไมโครซอฟต์วินโดว์ 8 ทั้งแบบ 32 บิต และ 64 บิต ที่สุ่มเสี่ยงว่าจะมีโปรแกรมแปลกปลอมอยู่ ซึ่งหลังจากทดลองเรียกใช้งานแล้วพบว่า ตัวอนาไลเซอร์นั้นตรวจพบ BHO แปลกปลอมอยู่จำนวน 1 ตัว โดยหลังจากผู้พัฒนาได้นำชื่อของ BHO ตัวนี้ไปค้นคว้าถึงความไม่ชอบมาพากลของมันในไซต์หลายไซต์พบว่า ในหลายไซต์ได้ระบุว่า BHO ตัวนี้เป็น BHO ที่แปลกปลอมจริงๆ กล่าวโดยสรุปแล้วผลลัพธ์ของโปรแกรมก็ทำงานได้อย่างถูกต้อง คือสามารถตรวจจับได้โดยไม่ต้องพึ่งพาฐานข้อมูล (Database) เข้ามาเกี่ยวข้อง

รูปที่ 4.5 BHOแปลกปลอมที่ค้นพบในระบบที่ทำการตรวจสอบ



รูปที่ 4.6 เวปไซต์ System Lookup ได้ระบุว่า BHO ตัวนี้เป็น BHO ที่น่าสงสัยเช่นกัน



รูปที่ 4.7 เวปไซต์ Herdprotect ได้ระบุว่าเป็น Adware

บทที่ 5
สรุปและวิจารณ์

5.1 ขอบเขตและข้อจำกัดของโปรแกรม

- ทำงานได้ในระบบปฏิบัติการไมโครซอฟต์วินโดว์ 7 และไมโครซอฟต์วินโดว์ 8 ทั้งแบบ 32 บิต และ 64 บิต
- ต้องติดตั้ง Visual C++ Redistributable for Visual Studio 2012 Update 4
- ผู้ใช้งานต้องมีการเชื่อมต่ออินเทอร์เน็ต (Internet)
- ตั้งค่าให้เรียกใช้ Google Chrome เป็นเบราวเซอร์เริ่มต้น เพื่อการแสดงผลที่สมบูรณ์แบบ

5.2 กลุ่มผู้ใช้งานโปรแกรม

บุคคลทั่วไปที่สงสัยว่าระบบของตนเองนั้นถูกละเมิดความปลอดภัย หรือว่าบุคคลที่ต้องการตรวจสอบว่าระบบของตนเองนั้นมีโปรแกรมอะไรทำงานอยู่บ้าง

5.3 ปัญหาและอุปสรรค

ในการพัฒนาตัวมอนิเตอร์ (Monitor) นั้นทำได้หลายวิธีมาก แต่วิธีที่เหมาะสมที่สุดนั้นมีเพียงวิธีเดียว ผู้พัฒนาจึงพบอุปสรรคในการศึกษาค้นคว้าว่าวิธีไหนดีกว่ากัน และนอกจากนั้นยังต้องใช้เวลาในการลองผิดลองถูกอีกด้วย

ในการสร้างโปรแกรมด้วย Visual Studio นั้นต้องทำการตั้งค่า (Configuration) ไฟล์โปรเจกต์ (Project) ให้เหมาะสม ซึ่งตัวผู้พัฒนาในตอนเริ่มต้นนั้นไม่มีความรู้ในเรื่องนี้จึงเสียเวลาในการลองผิดลองถูกอีกเช่นกัน

ในการสร้างไลบรารี (Library) ภายนอกที่ไม่สามารถสร้างได้ด้วย Visual Studio นั้นจำเป็นต้องใช้ คอมมานด์ไลน์ (Command Line) ในการสร้าง ซึ่งตัวผู้พัฒนาตอนเริ่มต้นไม่มีความรู้ในเรื่องนี้จึงเสียเวลาในการลองผิดลองถูกอีกเช่นกัน

บางครั้งผู้พัฒนาต้องการเลือกวิธีการเขียนโปรแกรมอย่างมีรูปแบบที่สวยงามทำให้เสียเวลา

ทฤษฎีเรื่อง API hooking นั้น การหาแหล่งข้อมูลในการศึกษาและอิมพลีเมนต์ (Implement) ยากมากทำให้เสียเวลาค้นคว้าอยู่นานพอสมควร

ผู้พัฒนายังไม่มีความชำนาญในภาษา C++ ทำให้เสียเวลาเรียนรู้และลองผิดลองถูกในการเขียนโปรแกรม

5.4 แนวทางการพัฒนา

สำหรับผู้ที่จะพัฒนาตัวแกรบเบอร์ต่อนั้น สามารถพัฒนาได้อย่างง่ายดายเพราะในโครงสร้างนั้นได้แบ่งโปรแกรมออกเป็นโมดูลย่อยๆโดยดูตามฟังก์ชันการใช้งาน และสำหรับผู้ที่จะพัฒนาตัวอนาไลเซอร์ก็ทำได้ง่ายเช่นกัน เพราะผู้พัฒนานั้นก็ได้พัฒนาโดยใช้หลักการ MVC โดยแนวทางการพัฒนานั้นอาจจะพัฒนาให้ตัวแกรบเบอร์สามารตรวจจับรูทคิท (Rootkit) ได้ พัฒนาให้ตัวแกรบเบอร์นั้นทำงานแบบ Always On พัฒนาให้สามารถดึงค่าแบนด์วิทช์ (Bandwidth) และการใช้งานซีพียู (CPU usage) และมีฟังก์ชันในการเลือกโพรเซสที่ตัวเองมั่นใจว่าปลอดภัย ส่วนตัวอนาไลเซอร์นั้นอาจจะพัฒนาให้ทำงานได้ในโทรศัพท์มือถือ หรือออกแบบให้เป็นโมบาย แอปพลิเคชัน (Mobile application) ได้อีกด้วย

บรรณานุกรม

[1] 32-64 Bit Registry key

http://msdn.microsoft.com/en-us/library/windows/desktop/aa384129(v=vs.85).aspx

http://msdn.microsoft.com/en-us/library/windows/desktop/ms724072(v=vs.85).aspx

[2] BHO

http://en.wikipedia.org/wiki/Browser_Helper_Object

[3] BHO Registry key

http://www.spywareinfoforum.com/topic/11738-bho-registry-location/

[4] CLSID Registry key

http://msdn.microsoft.com/en-us/library/ee488140.aspx

http://pcsupport.about.com/od/termshm/g/hkey_classes_root.htm

[5] Startup Registry key

http://www.bleepingcomputer.com/tutorials/windows-program-automatic-startup-locations/

[6] Uninstall Registry Key

http://msdn.microsoft.com/en-us/library/aa372105(v=vs.85).aspx

[7] Registry key API

http://msdn.microsoft.com/en-us/library/ctb3kd86(v=vs.110).aspx

http://msdn.microsoft.com/en-us/library/microsoft.win32.registrykey(v=vs.110).aspx

[8] System String Convert to string by using marshall

http://stackoverflow.com/questions/775708/converting-string-to-stdstring-basic-string-error-how-can-i-fix-this

[9] List

http://www.cplusplus.com/reference/list/list/list/

[10] JSON Cpp

http://jsoncpp.sourceforge.net/

http://stackoverflow.com/questions/4289986/jsoncpp-writing-to-files

[11] Convert Byte between Big eindian and Little eindian

http://stackoverflow.com/questions/105252/how-do-i-convert-between-big-endian-and-little-endian-values-in-c

[12] Handle Type Cast Warning
http://stackoverflow.com/questions/290699/warning-c4244-argument-conversion-from-size-t-to-dword-possible-loss-o

[13] C++Regular Expression
http://www.cplusplus.com/reference/regex/
http://www.cplusplus.com/reference/regex/regex_search/
http://www.developerz.com/cplusplus_escapesequences.htm

[14] C++ LibCurl Use to get html of BHO
http://curl.haxx.se/libcurl/c/libcurl-tutorial.html
http://stackoverflow.com/questions/9786150/save-curl-content-result-into-a-string-in-c

[15] Get Full Process Path
http://social.msdn.microsoft.com/Forums/vstudio/en-US/c48bcfb3-5326-479b-8c95-81dc742292ab/windows-api-to-get-a-full-process-path?forum=vcgeneral

[16] Process Status API
http://msdn.microsoft.com/en-us/library/windows/desktop/ms684884(v=vs.85).aspx
http://msdn.microsoft.com/en-us/library/windows/desktop/ms683217(v=vs.85).aspx
http://msdn.microsoft.com/en-us/library/windows/desktop/ms683198(v=vs.85).aspx
http://winprogger.com/getmodulefilenameex-enumprocessmodulesex-failures-in-wow64/

[17] File Version info Class
http://msdn.microsoft.com/en-us/library/system.diagnostics.fileversioninfo(v=vs.110).aspx

[18] IPC – mail Slot
http://www.codeproject.com/Articles/34073/Inter-Process-Communication-IPC-Introduction-and-S

[18] DLL Injection
http://resources.infosecinstitute.com/using-createremotethread-for-dll-injection-on-windows/

[19] API hooking
http://www.codeproject.com/Articles/30140/API-Hooking-with-MS-Detours

[20] Common registry where malware reside
http://www.wilderssecurity.com/showthread.php?t=142620

[21] Get Connection of process
http://www.codeproject.com/Articles/14423/Getting-the-active-TCP-UDP-connections-using-the-G

[22] Html agility pack library for get html from url
http://htmlagilitypack.codeplex.com/downloads/get/437942

[23] Xpath
http://www.w3schools.com/xpath/default.asp

[24] Serialize Json
http://json.codeplex.com/

[25] MultiThread
http://stackoverflow.com/questions/9732709/the-calling-thread-cannot-access-this-object-because-a-different-thread-owns-it

[26] WPF
http://www.codeproject.com/Articles/140611/WPF-Tutorial-Beginning

[27] WPF – Data Binding
http://www.codeproject.com/Articles/29054/WPF-Data-Binding-Part-1

[28] CSS
http://www.w3schools.com/css/

[29] HTML
http://www.w3schools.com/html/

[30] javascript
http://www.w3schools.com/js/js_howto.asp

[31] Backbone js
http://adrianmejia.com/blog/2012/09/11/backbone-dot-js-for-absolute-beginners-getting-started/

[32] Underscore js
http://underscorejs.org/

[33] Grunt js
http://gruntjs.com/getting-started

[34] Less
http://www.lesscss.org/

[35] BigVideo js
http://dfcb.github.io/BigVideo.js/
http://www.webdesignermag.co.uk/tutorials/integrate-full-screen-video-backgrounds-using-bigvideo-js/