ชุดโปรแกรมจำแนกและบันทึกพฤติกรรมผู้บุกรุก

Honeypot Program Suite

นาย ดิเรก ยิ้มละมัย

นาย สมพล พูนภัสสร

นางสาว สันต์อาวี กรรณล้วน

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

ชุดโปรแกรมจำแนกและบันทึกพฤติกรรมผู้บุกรุก

Honeypot Program Suite

โดย

นาย ดิเรก ยิ้มละมัย

นาย สมพล พูนภัสสร

นางสาว สันต์อาวี กรรณล้วน

อาจารย์ที่ปรึกษา

อ. ธนา หงษ์สุวรรณ

อ. อัครเดช วัชระภูพงษ์

อ.ธนัญชัย ตรีภาค

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

ปริญญานิพนธ์ ปีการศึกษา 2547

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ชุดโปรแกรมจำแนกและบันทึกพฤติกรรมผู้บุกรุก

Honeypot Program Suite

คณะผู้จัดทำ นาย ดิเรก ยิ้มละมัย รหัสประจำตัว 45015365

นาย สมพล พูนภัสสร รหัสประจำตัว 44010507

นางสาว สันต์อาวี กรรณล้วน รหัสประจำตัว 44010517

_____ อาจารย์ที่ปรึกษา

(อ. ธนา หงษ์สุวรรณ)

_____ อาจารย์ที่ปรึกษา

(อ. อัครเดช วัชระภูพงษ์)

_____ อาจารย์ที่ปรึกษา

(อ.ธนัญชัย ตรีภาค)

ชุดโปรแกรมจำแนกและบันทึกพฤติกรรมผู้บุกรุก

นาย สมพล พูนภัสสร 44010507
นางสาว สันต์อาวี กรรณล้วน 44010517
นาย ดิเรก ยิ้มละมัย 45015365
อ. ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษา
อ. อัครเดช วัชระภูพงษ์ อาจารย์ที่ปรึกษา
อ.ธนัญชัย ตรีภาค อาจารย์ที่ปรึกษา
ปีการศึกษา 2547

บทคัดย่อ

การรักษาความปลอดภัย ในระบบเครือข่ายคอมพิวเตอร์ในปัจจุบันนั้นองค์กรมีความซับซ้อนมาก อันเนื่องมาจากมีผู้ร่วมใช้ระบบมากขึ้นทำให้ความเสี่ยงที่ระบบขององค์กรจะถูกโจมตีมีเพิ่มขึ้นด้วย หากว่าภายในองค์กรมีการใช้เครื่องมือในการรักษาความปลอดภัยเช่นไฟร์วอลหรือตัวโปรแกรมตรวจจับพฤติกรรมระบบเครือข่าย ซึ่งการรักษาความปลอดภัยโดยใช้เครื่องมือดังกล่าวยังไม่สามารถให้รายละเอียดของพฤติกรรมของผู้ไม่ประสงค์ดีได้ เนื่องจากว่าเครื่องมือเหล่านั้น ใช้เพื่อป้องกันผู้ไม่ประสงค์ดีเข้ามาในระบบเครือข่ายโดยใช้กฎที่ได้กำหนดไว้ก่อน หากการโจมตีนั้นไม่ตรงกับกฎที่กำหนดไว้ก่อนหน้าเครื่องมือนั้นก็ไม่สามารถรักษาความปลอดภัยได้และจะไม่สามารถเก็บข้อมูลการบุกรุกครั้งนั้นได้ ซึ่งแตกต่างกับตัวฮันนี่พ็อตที่ใช้หลักการรอรับการโจมตี เฝ้าดูและบันทึกพฤติกรรมการบุกรุก โดยกำหนดเขตของการโจมตีไม่ให้ผู้ไม่ประสงค์ดีเข้าโจมตีระบบจริงได้ เพื่อให้สามารถนำมาวิเคราะห์รูปแบบการโจมตีในแบบต่างๆ ที่อาจเกิดขึ้นใหม่แม้ไม่ตรงกับกฎที่ได้กำหนดไว้ก่อนหน้าเนื่องจากตัวฮันนี่พ็อตใช้หลักการล่อหลอกผู้บุกรุก ให้หลงกลเข้ามายังตัวฮันนี่พ็อตเพื่อให้เครื่องที่เปิดบริการจริงได้รับความปลอดภัยเพราะผู้บุกรุกได้ตกเข้าไปอยู่ในกับดักแล้ว

Honeypot Program Suite

Mr. Sompol Poonpussorn 44010507

Miss Sanarwee Kanluan 44010517

Mr. Direk Yimlamai 45015365

Mr. Tana Hongsuwan

Advisor

Mr. Akkradach Watcharapupong

Advisor

Mr. Tanunchai Tripak

Advisor

Academic Year 2004

Abstract

In the present network security and host-based security becomes more of an interest and concern for organizations, and most security tools cannot protect and prevent all problems of security of organization. There is no tool which can learn behaviors of hackers, so Honeypot was introduced to solve these problems. Honeypot ,a new generation tool , is a trap set to detect or deflect attempts at unauthorized use of information systems. Honeypots can protect organizations in one of three ways; prevention, detection, and response.

กิตติกรรมประกาศ

เอกสารฉบับนี้และตัวชิ้นงานชุดโปรแกรมจำแนกและบันทึกพฤติกรรมผู้บุกรุกสำเร็จลุล่วงได้ด้วยดี ก็เนื่องมาจากการให้โอกาส การดูแล ให้คำแนะนำต่างๆ การสนับสนุน การให้คำสั่งสอนและให้คำปรึกษาเป็นอย่างดีเสมอมาจาก จากอาจารย์ ธนา หงษ์สุวรรณ อาจารย์ อัครเดช วัชระภูพงษ์ และอาจารย์ ธนัญชัย ตรีภาค ซึ่งต้องขอขอบพระคุณอาจารย์ทั้ง 3 ท่านเป็นอย่างสูง ขอขอบคุณภาควิชาวิศวกรรมคอมพิวเตอร์ และสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่ได้จัดเตรียมสิ่งอำนวยความสะดวก เพื่อให้งานวิจัยดำเนินไปได้อย่างสะดวกและรวดเร็ว ขอบคุณห้องวิจัยไอแซก (ISAG) ที่เป็นแหล่งประสิทธิ์ประสาทวิชาให้ความรู้ความเข้าใจ เป็นสถานที่ ที่มีความอบอุ่น ท่านอาจารย์ พี่ๆ ที่เป็นผู้ให้แนวทางแก้ไขและที่ปรึกษาของชิ้นงานจนลุลวงด้วยดี และท้ายที่สุดต้องขอขอบพระคุณบุคคลที่สำคัญที่สุดในชีวิตของข้าพเจ้าที่ทำให้ข้าพเจ้ามีทุกวันนี้คือ บิดา มารดา และบุคคลทุกคนในครอบครัวของข้าพเจ้า อันเป็นที่เคารพรัก คอยอุ้มชูเลี้ยงดู อบรมสั่งสอนข้าพเจ้ามาเป็นอย่างดี ทั้งยังให้ความรักความห่วงใย และกำลังใจที่ดีเสมอมา ข้าพเจ้าต้องขอขอบพระคุณมา ณ ที่นี้ด้วย

นาย ดิเรก ยิ้มละมัย
นาย สมพล พูนภัสสร
น.ส. สันต์อาวี กรรณล้วน

สารบัญ

สารบัญ (ต่อ)

สารบัญ (ต่อ)

สารบัญ (ต่อ)

สารบัญภาพประกอบ

สารบัญภาพประกอบ(ต่อ)

หน้า

สารบัญตาราง

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญ

ในระบบเครือข่ายปัจจุบัน มีผู้ไม่ประสงค์ดีและประสงค์ดีอยู่ในระบบเครือข่ายเป็นจำนวนมากในการบริการของระบบทั่วๆ ไปหากมีการโจมตีของกลุ่มผู้ไม่ประสงค์ดีแล้วกว่าที่ผู้ดูแลระบบจะทราบได้ว่ามีการโจมตีก็ล่าช้า และ ไม่สามารถที่จะเรียนรู้พฤติกรรมของผู้โจมตีได้ในทันที ต้องมาอ่านและวิเคราะห์เนื้อหาของระบบการดักจับพฤติกรรม (IDS) และอาจจะไม่ทราบทั้งหมดว่าผู้โจมตีได้ทำอะไรลงไปเพราะว่าเราได้เพียงแต่จับในส่วนของ packet ที่ส่งผ่านเข้า และ ออก จากระบบเท่านั้น ทั้งนั้นไม่มีผู้ดูแลระบบคนไหนที่ต้องการเรียนรู้พฤติกรรมการโจมตีด้วยระบบจริงของตนเอง ด้วยเหตุเหล่านี้ดังนั้น ผู้ดูแลระบบจึงต้องมีระบบฮันนี่พ็อต เพื่อให้เป็นแหล่งของการเฝ้าดูพฤติกรรมผู้โจมตี และ แยกแยะได้ระหว่างผู้ประสงค์ดีและผู้โจมตี

ระบบฮันนี่พ็อตมีสองประเภทหลักๆ คือ

1. การโต้ตอบในระดับขั้นพื้นฐาน (Low-interaction)
2. การโต้ตอบในระดับขั้นสูง(High-interaction)

ระบบฮันนี่พ็อตเองนั้นไม่ได้มีเพียงแค่ตัวฮันนี่พ็อตเท่านั้นเนื่องจากว่าระบบฮันนี่พ็อตเองไม่ได้มีความสามารถในการแยกแยะ ระหว่างผู้ประสงค์ดีกับผู้โจมตีจึงมีหลากหลายวิธีที่ช่วยให้การแยกแยะนั้นมีประสิทธิภาพ ทั้งในการจัดระบบเครือข่ายเองและการใช้อุปกรณ์อื่นเข้ามาช่วยในการแยกแยะ

ในปัจจุบันการดูแลระบบเครือข่าย มีความยากมากขึ้นเนื่องมาจากว่ามีผู้ร่วมใช้ระบบมากขึ้นและมีความเสี่ยงที่ระบบขององค์กรณ์นั้นๆ มีอัตราเสี่ยงในการถูกเข้าโจมตีมากขึ้น ดังนั้นระบบฮันนี่พ็อตจึงได้เข้ามามีบทบาท ในการช่วยลดการเสี่ยงและเพิ่มโอกาสในการศึกษาพฤติกรรมและวิธีการเข้ามาในระบบของเราขากผู้ไม่ประสงค์ดี แต่ระบบเครือข่ายขององค์กรณ์ที่มีความซับซ้อน นั้นจะเป็นการยากที่เราจะวางตัวระบบฮันนี่พ็อตให้ได้ในตำแหน่งที่ดีที่สุดเนื่องจากว่า เราต้องวิเคราะห์และศึกษาในรายละเอียดของระบบเครือข่ายขององค์กรณ์นั้นๆ ก่อน

1.2 วัตถุประสงค์

1. เพื่อศึกษาวิธีการจำแนกผู้บุกรุกจากผู้ใช้งานปกติแล้วบันทึกพฤติกรรม
2. เพื่อสร้างโปรแกรมต้นแบบสำหรับจำแนกและบันทึกพฤติกรรมผู้บุกรุก

1.3 ขอบเขตของโครงงาน

1. ศึกษาความเป็นมาและรูปแบบของตัวฮันนี่พ็อต
2. ศึกษาตัวการจัดตั้งตัวฮันนี่พ็อตและตำแหน่งเหมาะสมในการวางในระบบเครือข่าย
3. เพื่อศึกษาวิธีการจำแนกผู้บุกรุกจากผู้ใช้งานปกติแล้วบันทึกพฤติกรรม
4. เพื่อสร้างโปรแกรมต้นแบบสำหรับจำแนกและบันทึกพฤติกรรมผู้บุกรุก

1.4 ขั้นตอนการดำเนินงาน

1. ศึกษาความรู้พื้นฐานเพื่อให้เข้าใจระบบฮันนี่พ็อต
2. ศึกษาส่วนที่เพิ่มเติมเพื่อให้ระบบฮันนี่พ็อตมีประสิทธิภาพมากขึ้น
3. วิเคราะห์ความเหมาะสมในการวางตัวระบบฮันนี่พ็อต
4. หาแนวทางการจำแนกผู้บุกรุกจากผู้ใช้งานปกติ
5. สร้างโปรแกรมต้นแบบเพื่อจำแนกและบันทึกพฤติกรรมผู้บุกรุก
6. ทดลองการใช้ระบบฮันนี่พ็อตที่ได้สร้างขึ้นมา
7. ทำรายงานและสรุปผล

บทที่ 2
หลักการพื้นฐานของระบบฮันนี่พ็อต

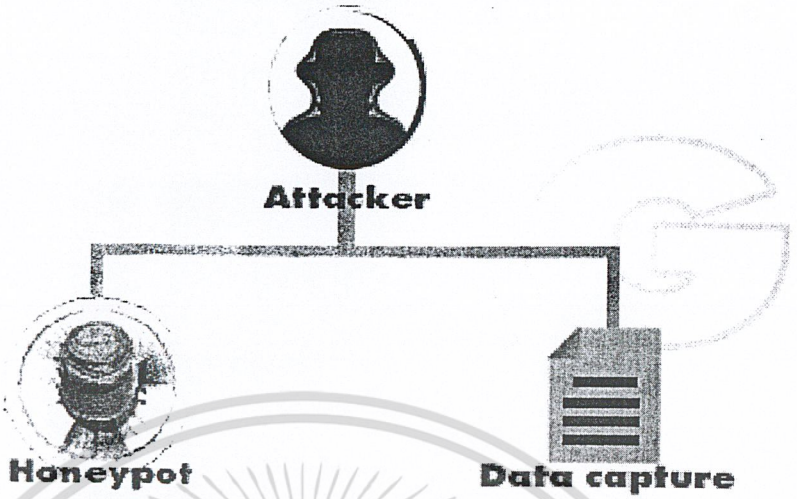
ในส่วนนี้เป็นความรู้เบื้องต้นที่จะทำให้ผู้ศึกษาได้เข้าใจในระบบฮันนี่พ็อตเองเพื่อให้ได้มาซึ่งความเข้าใจที่จะก้าวต่อไปในบทต่อๆ ไปที่เราได้จัดทำไว้ให้แล้ว จะกล่าวถึงนิยาม แนวทางเบื้องต้น

2.1 เกี่ยวกับ Honeypot

ในความเข้าใจพื้นฐาน honeypot จะไม่แสดงตนว่าตนเองคือ honeypot จะคอยแอบอยู่เพื่อไม่ให้ผู้ประสงค์ไม่ดีเข้าไปแล้วควบคุมการทำงานของบุคคลที่หลุดเข้าไป เพราะอาจเป็นการกระทำที่เลวร้ายแล้วแต่การกำหนดของ honeypot นั้นๆ แต่หลักของการกำหนด honeypot คือ " **An information system resource whose value lies in unauthorized or illicit use of that resource** "

ประโยชน์ของตัว honeypot ที่ได้รวมเข้าไว้ด้วยกัน

- Small data sets : honeypot จะเลือกข้อมูลที่รายงานได้สองแบบ Collect attack หรือ Unauthorized activity บางองค์กรณ์มีรายงาน 1,000 alert ต่อวันซึ่งเป็น log ที่มาจาก honeypot ดังนั้นการเลือก alert ที่ต้องการจะช่วยลดจำนวนลงได้ทำให้การจัดเก็บ log file ได้ง่ายขึ้น
- Reduced false positive : honeypot เองมีการลดการรายงานอย่างรวดเร็วซึ่งอาจจะเกิดข้อผิดพลาด ซึ่งโดยปกติแล้วจะเก็บเฉพาะที่มีการพยายามเข้าอย่างผิดกฎ
- Catching False Negative : honeypot สามารถพิสูจน์และจับการโจมตีแบบไม่เหมาะสมที่เกิดขึ้นมาก่อนหน้านี้ได้
- Minimal Resources : ตัว honeypot เองร้องขอ resource น้อย และ สม่ำเสมอบนระบบเครือข่ายที่ทำอย่างนี้ค่าใช้จ่ายจะสูงและเห็นผลอย่างแท้จริงในการแก้ปัญหา
- Encryption : honeypot สามารถตรวจจับ Packet ที่โจมตีมาโดยการ Encryption ได้
- In-depth Information : honeypot สามารถตรวจจับ data ที่ไม่รู้จักกับเทคโนโลยีได้มีการรวบรวมการโจมตีจากผู้โจมตีต่างๆ และการที่น่าจะเป็นการโจมตีได้
- IPv6 : มีการรองรับ IPv6 ในส่วนของเทคโนโลยีรุ่นใหม่บางตัวไม่สามารถตรวจจับได้ แต่ตัว honeypot เองสามารถที่จะตรวจจับได้



รูป 2.1 รูปเพื่อความเข้าใจเบื้องต้น

เมื่อ Attacker เข้ามาเอาข้อมูล Attacker นั้นนึกว่าที่เข้ามานั้นเป็นระบบจริง (Honeypot) จากนั้นระบบจะส่งข้อมูลไปยัง **Data capture** เพื่อเก็บการกระทำแล้วกักขังผู้ประสงค์ร้ายในระบบ

(แนวคิดของภาพมาจาก Video file ของ HoneynetWeb)

2.2 คำนิยาม Honeypot

คำนิยาม และ ค่าของ Honeypot (โดย Lance Spitzner) honeypot เป็นเรื่องที่น่าตื่นตาตื่นใจ เนื่องจากเพิ่งจะเกิดเป็นเทคโนโลยีใหม่เป็นขนิดที่ถือหลักเกณฑ์ของ Cliff Stoll ในหนังสือชื่อ The Cuckoo's Egg และ Bill Cheswick มีเอกสารชื่อ An Evening with Berferd โดยได้กล่าวไว้อย่างละเอียดว่า honeypot นั้นใช่เพื่อเป็นประโยชน์ในการตรวจสอบและตรวจจับแต่ไม่มีความสามารถในการป้องกันให้ข้อมูลปลอดภัย

คำนิยาม

ในแรกเริ่มเดิมทีไม่มีความเข้าใจว่า honeypot คืออะไร ในสมันนั้นยังไม่สามารถหาคำอธิบายได้ หลังจากนั้นเราจึงได้รู้ได้ยินและเริ่มรู้จักมันดีขึ้น มันไม่เหมือนกันกับโปรแกรมหรือระบบรักษาความปลอดภัยจำพวก firewall ชนิดต่างๆ หรือ IDS ตัว honeypot จะไม่แก้ปัญหาเจาะจงการนำมาวางในที่ที่เหมาะสมมาแทนที่เครื่องมือบางอย่างนั้นจะทำให้ honeypot มีศักยภาพที่โดดเด่นไม่น้อยเลยทีเดียว

พวกเราสามารถทำทุกสิ่งที่เป็นการตรวจสอบการโจมตีแบบ Encryption ในรูปแบบของ IPv6 ได้ระบบเครือข่ายนั้นอาจจะยังมีจุดอ่อนที่มองไม่เห็นแต่จะตรวจจับมันได้ตัวระบบเครือข่ายที่ดีและเหมาะสมนั้นจะส่งผลให้ honeypot มีพลังความสามารถที่จะรองรับการ

ตรวจสอบได้อย่างเหมาะสม มันมีการปรับปรุงให้สามารถท้าทายเหล่า Attacker ที่เข้ามาและได้รับความไว้วางใจโดยกระทำตามหลักของการวาง honeypot ดังที่ได้กล่าวมา
ตัวข้อมูลของระบบที่มีข้อมูลข้อมูลไม่ยอมให้มีสิทธิ์ที่จะขโมยใช้ข้อมูล

ในการอธิบายขั้นพื้นฐานเป็นแบบมุมมองโดยรวมต่างกันที่การแสดงตัวตนของ honeypot เราจะพูดในบทนี้ในส่วนของตัวอย่าง honeypot ซึ่งจะได้กล่าวต่อไป
โดยทั้งหมดจะอยู่ภายใต้การกำหนดของพวกเรา โดยมีค่าที่ไว้ใช้ลวงเหล่า Attacker เพื่อโต้ตอบสนองเหล่าผู้โจมตี

ในแนวคิดทั่วไปส่วนใหญ่ Honeypot จะทำงานเหมือนกันมี resources ที่ไม่ให้สิทธิ์ในการเข้าถึง แต่เราก็ไม่สามารถที่จะมีทุกๆผลิตภัณฑ์ตั้งสมมุติว่าตัว honeypot ไม่ควรที่จะเห็นอยู่ในระบบเพราะว่ามันไม่ต้องการให้ถูกรับรู้ว่ามีตัว honeypot อยู่ภายในระบบเครือข่ายของเราในความหมายโดยรวมของการโต้ตอบกับ honeypot จะคล้ายกับการไม่ให้สิทธิ์กับผู้ประสงค์ร้าย

ในการทุกๆการติดต่อจะพยายามนำเข้าสู่ Honeypot ซึ่งโดยมากจะทำเพื่อตรวจสอบการโจมตี หรือเข้ามาเพื่อหาผลประโยชน์อื่นๆ จาก honeypot จนกระทั่งมี concept ที่ง่ายมากๆ คือมันง่ายมากที่จะให้ตัว honeypot เองให้ผลประโยชน์ หรือ ไม่ให้มีก็สามารถทำได้ จะอธิบายขยายความจากที่กล่าวว่าผลที่ได้และไม่ได้คืออะไร

2.3 ข้อเด่นข้อด้อย

Advantages: honeypot ทั้งหลายนั้นมี concept พื้นฐานมากมายกับการให้บุคคลบางคนนั้นได้ใช้ประโยชน์ได้สูงสุด

Small data sets ของข้อมูลที่มีค่าสูง : Honeypot จะรวบรวมข้อมูลที่ถูกต้องไว้ในจำนวนไม่มากแทนที่จำต้องมี log files ขนาดเป็น Gigabyte ต่อวันเราสามารถลดขนาดให้เป็น Megabyte ต่อวัน ก็พอ ในหารเข้ามาของข้อมูลทั่วๆ ไปเป็น 10,000 alert ต่อวันนั้นเราสามารถที่จะลดให้เหลือ 10 alert ต่อวันได้ จงจำไว้ว่า honeypot เองนั้นจะจับแต่การกระทำที่ไม่ถูกต้องทุกๆ การกระทำที่ honeypot ดูเหมือนว่าจะมีการพยายามละเมิดสิทธิ์ หรือ พยายามที่จะเอาผลประโยชน์ เช่น honeypot จะลดการกระจายตัวของ log files นั้นโดยการรวมกลุ่มแต่ข้อมูลที่มีมากๆ จากการกระทำของผู้ประสงค์ร้ายนั้นหมายความหมายว่าจะมีมากทีเดียวที่จะส่งผลให้การแยกแยะ Data ของ honeypot เป็นกลุ่มๆ และได้มาซึ่งค่าข้อมูลที่ต้องการ

New tools and tactics :

- เครื่องมือใหม่ๆ และ ยุทธวิธี : honeypot เองมีการออกแบบให้จับตาดูทุกอย่างที่กระทำต่อมัน จึงได้รวบรวมเอาเครื่องมือหรือยุทธวิธีที่ไม่เคยพบมาก่อน

- Minimal resources : honeypot ร้องขอ resources เพียงเล็กน้อยจะจับตาเฉพาะการกระทำที่ไม่ถูกต้อง
- Encryption or IPv6 : ไม่เหมือนกันกับระบบรักษาความปลอดภัยโดยทั่วๆ ไป เช่น IDS ตัว honeypot จะทำงานได้ในส่วนที่เข้ารหัส หรือ IPv6 มันจะไม่เข้ากันกับเนื้อข้อมูลของผู้ประสงค์ร้ายที่เข้ามาใน honeypot นั้นตัว honeypot เองจะสามารถตรวจจับ และ บันทึกไว้ได้
- Information : honeypot สามารถจัดกลุ่มในรายละเอียดส่วนลึกได้ถ้าตัวเทคโนโลยีอื่นๆสามารถรองรับได้
- Simplicity : สุดท้าย honeypot เองก็มีแนวคิดที่เป็นพื้นฐานและเรียบง่ายมากๆ ไม่จำเป็นต้องมี algorithm ที่หรูหราวุ่นวายแต่อย่างใดในการทำมาสร้างมีตารางกำหนดการจัดการ หรือ การปรับปรุง Signature ในเทคโนโลยี เริ่มแรกจะมีข้อผิดพลาดหรือแปลความผิดพลาดได้

Disadvantages : คล้ายกับเทคโนโลยีต่างๆ ตัว honeypot เองก็มีส่วนที่เป็นจุดอ่อนอยู่มาก เพราะว่ามันไม่สามารถนำมาวางแทนเทคโนโลยี ปัจจุบันได้ แต่มันก็ทำงานกับเทคโนโลยีในปัจจุบันได้

- Limited view : honeypot เองสามารถมีแนวทางและจัดการกับการกระทำที่เกิดขึ้นโดยตรงกับ เหล่า honeypot เองตัว honeypot จะไม่สามารถจับการโจมตีในส่วนของระบบอื่นได้ เว้นแต่ว่าการโจมตีนั้นเกิดขึ้นกับตัว honeypot เองโดยตรง
- Risk : ในทุกๆ ความปลอดภัยย่อมมีความเสี่ยงตัว firewall เองมีความเสี่ยงของการถูกเจาะในการเข้ารหัสทำให้เกิดความเสี่ยงที่เริ่มเกิดบ้างแล้ว IDS จะคอยรับรู้ว่ามีอัตราเสี่ยงของการผิดพลาดของการโจมตี honeypot ก็ไม่ต่างกันกับสิ่งเหล่านี้ที่มีอัตราเสี่ยงเช่นกันโดยในแต่ละชนิด honeypot มีอัตราเสี่ยงของการเริ่มถือโอกาสเข้าโดยบุคคลที่ไม่ประสงค์ดี และการทำอันตรายระบบ ในอัตราเสี่ยงต่างๆ สำหรับความต่างของ honeypot โดยจะขึ้นอยู่กับชนิดของ honeypot มันจะมีอัตราเสี่ยงน้อยลงถ้าใช้ IDS Sensor ขณะเดียวกันบาง Honeypot มีแนวทางจัดการอัตราเสี่ยง พวกเราถือว่าตัว honeypot มีอัตราเสี่ยงมากน้อยเท่าใดนั้นจะกล่าวในภายหลัง

ที่ได้กล่าวมาทั้งหมดนี้เป็นการช่วยตัดสินใจว่าคุณกำลังทำให้ Advantages และ Disadvantages โดยการกำหนดค่า honeypot ของคุณ

ชนิดของระบบฮันนี่พ็อต

ระบบฮันนี่พ็อตมีรูปแบบของการกระทำการโต้ตอบอยู่สองรูปแบบที่สำคัญคือ Low-interaction และ High-interaction เป็นสองรูปแบบหลักที่ระบบฮันนี่พ็อตเองได้สร้างไว้เพื่อให้ตอบสนองต่อความต้องการที่ผู้ดูแลระบบต้องการ ในส่วนนี้ได้อธิบายถึงข้อแตกต่าง ข้อดี และส่วนที่เกี่ยวข้องพร้อมทั้งรูปแบบระบบเครือข่ายที่ได้ยกตัวอย่างไว้ให้

2.4 ชนิดของฮันนี่พ็อต

Honeypot ต่างๆ มาในหลายรูปแบบ และ หลายขนาดการทำพวกมันนั้นยากที่จะเข้าใจ ในการช่วยให้เรานั้นเข้าใจ honeypot และ ความต่างของชนิดต่างๆ พวกเราจะนำแบ่งมันออกเป็นลักษณะทั่วๆ ไป 2 ลักษณะ คือ Low-interaction และ High-interaction การแบ่งแยกนี้ช่วยให้เราเข้าใจว่า honeypot ชนิดอะไรที่คุณทำ มันมีทั้งจุดเด่น และ ด้อย ในการทำให้มีผลที่ส่งถึงกัน การกำหนดระดับของการโต้ตอบแก่ตัว honeypot เพื่อตอบรับผู้โจมตีส่วนของ Low-interaction นั้นจะจำกัดในส่วนของการตอบสนอง พวกมันจะมีการกระทำแบบเสมือนโดยการ emulator ตัว server นั้นๆ และ ระบบปฏิบัติการ ผู้โจมตีจะได้รับการตอบกลับที่มีข้อจำกัดโดยขึ้นอยู่กับระดับของตัว emulator ตัวอย่างเช่น ทำ emulation FTP service ให้ listen บน port 21 บางทีอาจจะให้ emulate ตัว FTP login หรือบางทีจะทำให้รองรับหลายๆ อย่างบน command line
ในความได้เปรียบของ honeypot แบบ Low-interaction คือมันไม่ยุ่งยากตัว honeypot เองจะง่ายต่อการเปลี่ยนแปลง และ บำรุงรักษากับการเกิดอัตราเสี่ยงเล็กน้อยโดยปกติแล้วพวกเขาจะร่วมกับการติดตั้ง software การเลือกระบบปฏิบัติการ และ service ที่ต้องการจะ emulator และ monitor และจัดการทำ honeypot ไปจากที่นั่น

มีระบบ Plug & Play ซึ่งจะทำให้การเปลี่ยนแปลงเป็นไปได้โดยสะดวกสำหรับองค์กรณ์ส่วนใหญ่เช่นกัน การ emulate ตัว service จะช่วยลดความเสี่ยงโดยจะบรรจุสิ่งที่จะใช้ตอบโต้ผู้บุกรุก ผู้บุกรุกจะไม่สามารถเข้ายึดครองระบบปฏิบัติการนั้น (honeypot) เพื่อใช้ในการโจมตีผู้อื่นต่อไป

ในส่วนของ Disadvantages ของตัว honeypot แบบ Low-interaction คือการมี log ที่รายงานออกมาจำกัด และ ต้องออกแบบเฉพาะการตรวจจับเฉพาะการกระทำที่รู้เท่านั้นการ emulate ตัว service สามารถให้ผลได้ดีดังนั้นมันจะง่านในการที่จะตรวจสอบพฤติกรรมของผู้โจมตีไม่มีผลเสียแต่อย่างใดกับการ emulate ที่เหมือนจริงสามารถจัดการโจมตีในขั้นสุดท้ายในปัจจุบันได้ ตัวอย่างhoneypot ที่เป็นแบบ Low-interaction เช่น Specter, Honeyd และ KFSensor

Honeypot แบบ High-interaction มีความยุ่งยากซับซ้อนกับปัญหาที่นำไปสู่ระบบปฏิบัติการที่ใช้อยู่ และ application เมื่อไม่มีตัว emulate ก็จะทำให้เกิดการโจมตีจริง ถ้าคุณต้องการให้ Linux

honeypot จำลอง FTP server คุณจะต้อง run ตัว FTP server จริงๆ ในระบบของคุณข้อดีของ High-interaction คือครอบคลุมเพราะว่ามีทุกอย่างเหมือนจริง

เริ่มแรกคุณสามารถตรวจจับสรุปผลจากข้อมูลได้จากการโจมตีจริงระบบจะตอบสนองกับผู้โจมตีคุณจะได้เรียนรู้ขอบเขตของพฤติกรรมของเหล่าผู้โจมดีได้ดียิ่งขึ้น ทุกๆ อย่างจะรองรับอยู่ในตัว rootkits ตัวใหม่ถึง session IRC ข้อดีข้อที่สองของ honeypot แบบ High-interaction คือไม่ได้จำลองว่าควรจะตอบกลับแบบใดเพราะว่าที่ใช้เป็นตัวระบบจริงที่ติดตั้งอยู่บน honeypot ซึ่งแทนที่พวกเขาจะจัดหาตัวที่คอยดูการกระทำต่างๆ นี้คือตัว honeypot แบบ High-interaction ซึ่งเป็นตัวแก้ปัญหาและศึกษาพฤติกรรมบางอย่างที่เราไม่ได้คาดคิดไว้ก่อนมีตัวอย่างที่ดีของ honeypot แบบ High-interaction คือ honeypot จะจับตาดูเข้ารหัส backdoor command บน IP Protocol (ไม่เป็นมาตรฐาน) เป็น Protocol 11 , Network Voice Protocol อย่างไรก็ตามตัว honeypot แบบ High-interaction ก็เพิ่มอัตราเสี่ยงเพราะว่า hacker นั้นสามารถที่จะใช้ระบบ honeypot ซึ่งเป็นระบบปฏิบัติการจริงโจมตีระบบที่ไม่ใช่ honeypot ก็อาจจะเป็นได้เช่นกัน ดังนั้นตามผลลัพธ์นี้เราจึงต้องเพิ่มขั้นตอนที่รองรับการโจมตีจากการโจมตีระบบอื่น ที่ไม่ใช่ honeypot จะได้รับในกรณีทั่วๆ ไป honeypot แบบ High-interaction สามารถทำได้หลายอย่างได้มากกว่า honeypot แบบ Low-interaction อย่างไรก็ตามเราสามารถแก้ไขความยุ่งยากและประปรุงได้ตัวอย่างของตัว honeypot แบบ High-interaction ก็เช่น Symantec Decoy Server และ Honeynets จะขยายความให้เข้าใจถึง honeypot แบบ Low-interaction และ High-interaction ดังตัวอย่างต่อไปนี้

2.5 รูปแบบของระบบที่โต้ตอบผู้โจมตีระดับต่ำ

Honeyd : Low-interaction honeypot

Honeyd เป็น Low-interaction honeypot สร้างโดย Niels Provg honeyd เป็น OpenSource และออกแบบมาเพื่อทำงานบนระบบ UNIX (แม้ว่าจะมี port ของระบบ Windows) honeyd ทำงานบนหลักการของการ monitor โดยไม่ใช้ IP space บ่อยครั้งที่มันจะมองดูการ connect ที่พยายามเข้ามาโดยไม่ใช่ IP ที่มีในกรณีปกติ
มันมักมีปฏิกริยาต่อการติดต่อมาของผู้โจมตีเหยื่อได้เข้ามาติดกับเราแล้ว

โดยพื้นฐานแล้ว honeyd จะตรวจสอบและเก็บ log ทุกๆ การติดต่อทั้ง UDP หรือ TCP ในการเพิ่มเติมคุณสามารถ configure ตัวจำลองการบริการได้ไม่ใช่แต่ว่า honeypot จะคอยตรวจสอบและเก็บ log เท่านั้นแต่มันยังจับตาดูทุกๆ การโจมตีที่ทำกับการจำลองบริการนั้น แต่ มันยังจับตาดูทุกๆ การโจมตีที่ทำกับการจำลองบริการนั้นๆ ในกรณีของการจำลอง FTP server พวกเราสามารถใช้ขีดความสามารถของการจับตาดูการ login และ Password ของผู้โจมตีจากการออกคำสั่งที่พวกเขาทำและบางเหตุการที่เขากำลังค้นหา หรือ หลักฐานที่เหล่าผู้โจมตีได้ทิ้งเอาไว้ทั้งหมดนี้ขึ้นกับระดับของตัวจำลองที่ honeypot ทำให้ส่วนใหญ่ตัวจำลองการบริการจะใช้กันมากพวกเขานั้นคาดว่าชนิดของพฤติกรรมและเหล่า program ที่โต้ตอบก็มีข้อจำกัดของมันหากว่าผู้โจมตี โจมตีรูปแบบ A ให้ทำดังนี้ ถ้าโจมตีแบบ B ให้ทำดังนี้เป็นหนทางที่ใช้เพื่อการโต้ตอบก็มีข้อจำกัดของมันอยู่หากว่าผู้โจมตีทำบางอย่างที่เกินขอบเขตของตัวจำลองที่ได้คาดเดาไว้มันจะไม่สามารถโต้ตอบกลับไปได้ ส่วนใหญ่แล้ว Low-interaction จะตอบ Error messages ง่ายๆ กลับไปคุณเห็นว่าอะไรเป็นคำสั่งใน FTP server สำหรับ honeyd แล้วรองรับเท่าใดนั้นสามารถดูได้จาก source code

Honeypot บางตัว ยกตัวอย่างเช่น Honeyd ไม่สามารถจำลองตัวบริการต่างๆ ได้แต่จำลองระบบปฏิบัติการจริงๆ ในคำกล่าวอื่น ตัว honeyd ไม่สามารถทำตัวเหมือนเพื่อตอบการโจมตีจากผู้โจมตี เช่น Cisco Router, WinXP webserver หรือ Linux DNS server มีประโยชน์มากทีเดียวหากเราจำลองระบบปฏิบัติการที่ต่างๆ กันออกไป

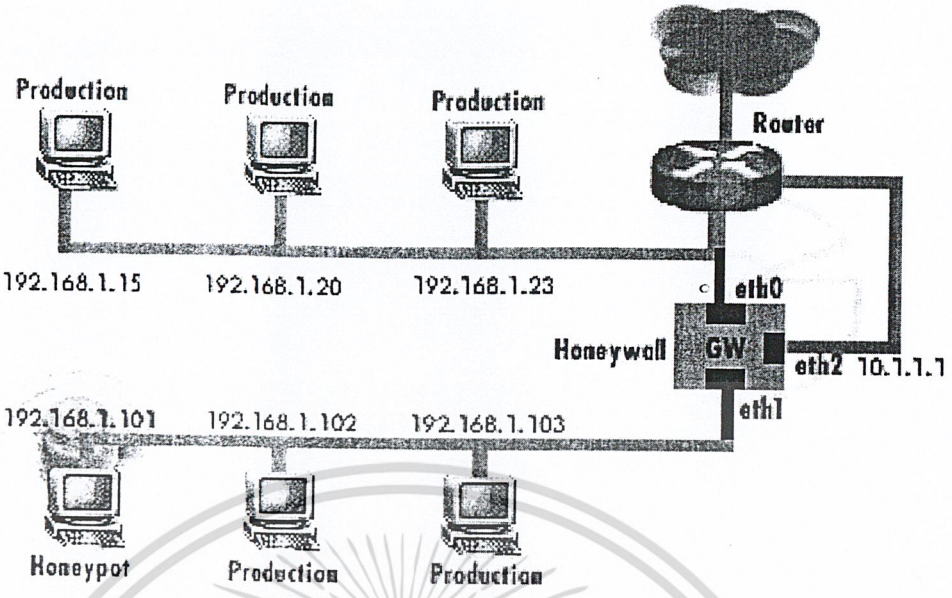
เริ่มแรก ระบบ honeypot นั้นสามารถประกอบร่วมเข้ากันกับระบบเครือข่ายต่างๆ ได้เป็นอย่างดีถ้า honeypot มีประโยชน์และพฤติกรรมคล้ายกันกับผลิตภัณฑ์ที่มีอยู่ในระบบ
ขั้นที่สอง คุณสามารถกำหนดเป้าหมายต่อผู้โจมตีโดยเตรียมระบบและบริการที่พวกเขาใช้เป็นเป้าหมายบ่อยๆ หรือ ระบบปฏิบัติการของคุณเองต้องเรียนรู้เอาเอง มีอยู่สองส่วนในเรื่องของการจำลองระบบปฏิบัติการส่วนแรกเกี่ยวกับการจำลองการบริการ เมื่อเหล่าผู้โจมตีติดต่อเข้ามายังบริการที่จำลองเอาไว้ คุณสามารถมีตัวบริการที่มีพฤติกรรมคล้ายและดูเหมือนกับรูปแบบในตัวระบบปฏิบัติการ สำหรับตัวอย่างถ้าคุณมีตัวบริการที่จำลองอยู่เป็น webserver และคุณต้องการให้ honeypot ของคุณดูเหมือนระบบปฏิบัติการ Windows 2000 server ดังนั้นคุต้องจำลองพฤติกรรมของ

ตัว IIS webserver สำหรับ Linux คุณควรจำลองเป็นตัว Apache webserver โดยส่วนใหญ่แล้ว honeypot เองจะจำลอง manner ของระบบปฏิบัติการ บาง honeypot ที่ทันสมันกว่าจะนำมาจำลองต่อไปเพิ่มขึ้นอีก (ซึ่ง Honeyd ก็ได้ทำ แต่ไม่ได้ทำแค่เพียงเท่านี้ พวกเขายังเพิ่มการจำลองต่างๆ) ที่ IP stack level ถ้าบางคนตรวจสอบจาก active fingerprinting เพื่อตัดสินใจว่าเป็นระบบปฏิบัติการใดของตัว Honeypot ของคุณ โดยมากแล้ว honeypot เองจะตอบสนองกับ IP stack จะเลียนแบบพฤติกรรมการโต้ตอบให้เหมือนกันกับ IP Stack ของระบบปฏิบัติการที่ใช้ลงบน honeypot ตัว honeyd จะตอบกลับแบบลวงไป โดยการทำไม่ได้แต่เพียงการจำลองการบริการเท่านั้น แต่ ตัวจำลอง IP stack จะเลียนแบบพฤติกรรมให้เหมือนกับระบบปฏิบัติการที่ต้องการระดับของการจำลองและใช้การตบตาโดขึ้นกับว่า honeypot ที่คุณเลือกนั้นทันสมัยเพียงใด

2.6 รูปแบบของระบบที่โต้ตอบผู้โจมตีระดับสูง

Honeynets : High-interaction honeypot

ตัว honeynet เป็นตัวที่ดีที่สุดที่จะมาเป็นตัวอย่างของ honeypot แบบ high-interaction ตัว honeynet ไม่ใช่ผลิตภัณฑ์ แต่เป็นเพียง sorfware มันจะติดตั้งลงบนเครื่องคอมพิวเตอร์ของคุณแทนที่ ซึ่ง honeynet คือ สถาปัตยกรรม ที่ครบถ้วนในเรื่องระบบเครือข่ายคอมพิวเตอร์ที่ถูกออกแบบมาเพื่อผู้โจมตีในแนวคิดคือมีสถาปัตยกรรมที่สร้างให้มีการควบคุมระบบเครือข่ายอย่างเต็มประสิทธิภาพ หนึ่งในการกระทำทั้งหมดคือการควบคุมและจับตาในระบบเครือข่ายที่เราวางเพื่อเจตนาให้เป็นเหยื่อในระบบคอมพิวเตอร์จริงที่รัน application จริงบุคคลที่ไม่ประสงค์ดีโดจมตีและเข้าสู้ระบบที่พวกเราจัดเตรียมไว้ เมื่อพวกเขาทำอะไรก็จะไม่ทราบว่าได้เข้าสู้ Honeynet แล้ว ทุกๆ การกระทำของผู้โจมตีเหล่านั้นผู้ดูแลระบบจะได้รับข้อมูลผ่านมาทาง SSH session หรือ เข้า สู่ email หรือ file uploads เหล่าผู้โจมตีจะถูกจับตาดูจนกระทั่งออกไปตัว honeynet ก็จะรู้ว่าออกไปแล้วโดยการทำงานจะใส่เข้าไปใน kernel module บนระบบที่เราทำเป็นเหยื่อเพื่อจับตาดูทุกการกระทำของผู้โจมตีในขณะเดียวกัน Honeynet จะควบคุมการทำงานของผู้โจมตีเช่นกัน ตัว Honeynet จะจะทำหน้าที่บอกได้จาก Honeywall gateway นี้จะเป็น gateway ที่เป็นระบบปิดเป็นภายในระบบของเหยื่อแต่ก็ควบคุมภายนอกเช่นกันโดยการใช้ intrusion prevention ต่างๆ นี้จะทำให้เกิดความยืดหยุ่นกับผู้โจมตีที่เข้าอยู่ในระบบเหยื่อ แต่ การขัดขวางผู้โจมตีจากอันตรายต่างๆ จากการโจมตีเครื่องที่ไม่ใช่ Honeynet



รูปที่ 2.2 การเชื่อมต่อพื้นฐาน

2.7 คุณค่าของระบบฮันนี่พ็อต

ตอนนี้เราได้ทำความเข้าใจในสองแบบคร่าวๆ แล้ว พวกเราสามารถจับประเด็นบนค่าต่างๆ ของมันโดยแต่ละชนิดนั้นอย่างไรเราก็สามารถใช้ Honeypot อื่นๆ ได้จากในนั้นเราเลือกมาสองแบบ ตัว honeypot เองสามารถใช้ร่วมกันกับผลิตภัณฑ์ หรือ งานวิจัยได้เมื่อใช้ตัวผลิตภัณฑ์ประกอบเข้ากับประกอบเข้ากับ honeypot จะป้องกันองค์กรณ์ได้นี้ควยใสเครื่องป้องกัน, เครื่องตรวจจับ หรือช่วยเหลือองค์กรณ์ให้โต้ตอบกับผู้โจมตีได้

เมื่อเราใช้ร่วมกับการวิจัย honeypot จะเริ่มใช้เป็นการเก็บข้อมูลข้อมูลที่ว่านี้จะต่างกันกับข้อมูลที่องค์กรณ์มี มีบางอย่างที่อาจจะต้องการเพื่อเป็นแนวในการเรียนรู้ที่จะดูการโจมตีของผู้โจมตีของผู้โจมตีจนถึงสิ่งที่สนใจต่างๆ ที่คิดว่าเข้าใกล้อันตรายและเดาว่าน่าจะเกิด หรือ ถูกบังคับในลักษณะทั่วๆ ไป Honeypot แบบ Low-interaction บ่อยครั้งที่ใช้เพื่อเป็นผลิตภัณฑ์ร่วมในขณะที่ Honeypot แบบ High-interaction จะใช้เพื่อร่วมกับการวิจัย อย่างไรก็ตามในแต่ละชนิดของ Honeypot สามารถใช้ได้ตามแต่วัตถุประสงค์ เมื่อใช้ Product ที่ตรงตามที่ต้องการแล้วตัว Honeypot จะสามารถป้องกันองค์กรณ์คุณได้หนึ่งในสามเส้นทางนี้

1. Prevention
2. Detection
3. Response

เราจะแสดงให้เห็นในรายละเอียดว่า Honeypot ทำงานอย่างไรตัว honeypot สามารถช่วยป้องกันการโจมตีได้ โดยทางแรกจะมีการโจมตีแบบอัตโนมัติ เช่น หนอนไวรัส หรือ การ auto-rooters การโจมตีส่วนมากจะอยู่บนพื้นฐานของการ Scan โดย Tool ต่างๆ การ Scan แบบ random ในส่วนต่างๆ ของ Network เพื่อค้นหาส่วนที่อ่อนแอ ถ้าจุดที่อ่อนแอบน Network ถูกค้นพบจะถูกเครื่องมือต่างๆ โจมตีทั้ง tools อัตโนมัติต่างๆ และ หนอนไวรัสเข้ายึดครองระบบในหนทางหนึ่งที่ honeypot ช่วยป้องกันการกระทำ scan ที่นานก็จะ drop ทิ้งเรียกกันว่า Sticky honeypot เหล่านี้จะมีคอยดูแลโดยไม่ต้องใช้ IP และทำให้การโจมตีหยุดลงโดยพวกเค้าจะใช้ความหลากหลายของ TCP จับ เช่น ตัว windows size มีขนาด 0 โดยเกิดจากการที่ Attacker จัดการ Pattern มาจะจัดการให้การ scan ช้าลง หรือ ป้องกันการแพร่ของหนอนไวรัสภายในองค์กรณ์ของคุณ หนึ่งในตัวอย่างของ Sticky honeypots เช่น LaBrea Tarpit ตัว Sticky honeypot ส่วนมากแล้วจะเป็น Low-interaction (คุณสามารถทำเป็น no-interaction ได้)จะทำให้ attacker ทำงานได้ช้าลงตัว honeypot จะมีทั้งแบบล่อหลอกและ ป้องกันเป็นแนวคิดที่ทำให้ Attacker สับสนเพื่อให้เขาใช้เวลาอยู่ใน honeypot ในระหว่างนั้นเองคุณก็พบว่า Attacker กำลังทำอะไรและใช้เวลากับ Attacker ได้ทั้งโต้ตอบเขา หรือ จะหยุดเขา เพื่อนำไปใช้ในขั้นตอนต่อไปถ้าผู้โจมตีรู้ว่าองค์กรณ์ของคุณคือเครื่องไหน ระบบ จริงคือเครื่องไหน พวกเขาบางทีแล้วในเรื่องของการฉวยโอกาสจับโดยการใช้ Honeypot และ ที่แน่ๆ คือ Attacker โจมตีคุณไม่ได้แน่นอนเชื่อได้เลยว่า Honeypot จะเป็นอุปสรรคต่อ Attacker ในตัวอย่างของ Honeypot ที่ออกแบบมาโดยทำเป็น Deception toolkit เป็นชุดของ Low-interaction Honeypot

ในหนทางที่สอง honeypot สามารถช่วยป้องกันองค์กรณ์ของคุณโดยการตรวจสอบทั้งหมด การตรวจสอบเป็น Critical จุดมุ่งหมายของมันคือ ID ที่ผิดพลาด หรือ ถูกหยุดยั้งจากการป้องกันการไร้ความระมัดระวังของการป้องกันองค์กรณ์ จะมีข้อผิดพลาดเกิดขึ้นโดยตลอดถ้าสำหรับไม่คิดถึงเหตุผลอื่นๆ โดยบุคคลแล้วจะมันแต่สนใน Process โดยการตรวจสอบผู้โจมตีคุณสามารถตอบโต้ได้อย่างรวดเร็วจะหยุดหรือทำให้การโจมตีเบาลงจากการกระทำของ Attacker ในแต่ก่อนนั้นการทดลองกับส่วนที่ยากต่อการกระทำที่สุดเทคโนโลยีเช่น IDS และระบบการรายงานมีการทดลองแล้วไม่ได้สิ่งที่ดีสำหรับการอยู่รอด พวกเขาได้สร้าง DATA จำนวนมากทำให้มี file ขนาดใหญ่ยากต่อการตรวจสอบ ไม่สามารถตรวจสอบการโจมตีใหม่ๆ ได้และไม่สามารถทำงานบนการ Encrypt บน IPv6 ได้ในการสร้าง Low-interaction ที่ดีในการตรวจจับพวกมันติดตั้งและปรับปรุงง่ายกว่า High-interaction และลดอัตราเสี่ยง

สุดท้ายคือทางที่ สาม honeypot สามารถป้องกันองค์กรณ์โดยการตอบโต้นี้เป็นหนทางหนึ่งที่บ่อยครั้งจะเป็นการตอบโต้ออย่างดีมันมีจำนวนข้อมูลไม่มากนักที่บอกว่าผู้โจมตีคือใครเข้ามาอย่างไรหรือโจมตีอย่างไรจึงจะสำเร็จในสถานะการณ์ที่มีข้อมูลรายละเอียดบนการโจมตีของผู้โจมตีอย่างไรจึงจะสำเร็จในสถานะการณ์ที่มีข้อมู,รายละเอียดบนการโจมตีของผู้โจมตีแบบ Critical มีสองปัญหาที่รวมกันในการตกสู่การโต้ตอบเริ่มต้นบ่อยครั้งที่ระบบส่วนใหญ่ไม่ยอมให้วิเคราะห์แบบ

offline เช่นผลิตภัณฑ์ของ Mail Server การโจมตีขั้น Critical เช่นการ hack หน้าที่ของฝ่ายความปลอดภัยอาจจะไม่สามารถหยุดยั้งการ Down ของระบบและจำแนกการกระทำได้ไม่ถูกต้องแทนที่พวกเขาจำกัดการวิเคราะห์ระบบตั้งแต่ที่ยังอยู่รอด ขณะที่หยุดยั้งการบริการต่างๆ นี้จะทำให้ระบบหยุดการบริการ และ สามารถวิเคราะห์ได้ว่าเกิดอะไรขึ้น มรการโจมตีมากเท่าไหร่ที่สำเร็จ และ การกระทำถ้าถูกผู้โจมตีมีการทำลายระบบอื่นๆ ในผลกระทบอื่นๆ ระบบจะถูกดึงให้เข้าเป็น Offline จะเกิดข้อมูลที่เป็นขยะจำนวนมาก ที่ยากต่อการที่จะค้นหาบุคคลที่น่าสงสัยว่าได้ทำอะไรอย่างไรลงไปบ้าง โดยข้อมูลขยะเหล่านี้แสดงให้เห็นว่ามีการโจมตีเป็นจำนวนมาก มันเป็นไปได้ยากที่จะมานั่งตรวจดูว่าทำอะไรต่อวัน และ ใครเป็นผู้โจมตี Honeypot นี้จะช่วยแก้ปัญหาทั้งสองนี้ได้โดย Honeypot ทำการแยกเป็นแต่ละส่วน พวกมันสามารถทำได้อย่างรวดเร็ว และ ง่ายโดยเอาแต่ส่วน Offline สำหรับการมสิทธิ์ในการวิเคราะห์ระบบบอย่างเต็มที่จะรู้การโจมตีจากภายนอกได้ เช่นเดียวกันในส่วนของการโจมตี Honeypot จะจัดการเก็บ พวกการเข้าแบบไม่ได้รับอนุญาตการโจมตีแบบประสงค์ร้ายโดยการทำให้ตัว Honeypot นั้นง่ายต่อการ Hack ระบบมากที่สุดเพื่อใช้เป็นตัววิเคราะห์การ Hack ระบบก็เช่นกันใน Data ที่รับเข้ามาทาง Honeypot จะคลายกันกับที่ Attacker กระทำ ค่าของ Honeypot จัดหาอย่างรวดเร็วในการส่งไปเพื่อจัดการในส่วนของข้อมูลที่ต้องการอย่างรวดเร็ว และ โต้ตอบในทันทีที่เกิดข้อผิดพลาดในกรณีทั่วๆ ไป Honeypot แบบ High-interaction สร้างการโต้ตอบได้เป็นอย่างดีในการโต้ตอบการบุกรุกคุณจะสามารถรู้ได้ว่าเขากำลังจะทำอะไร และ อย่างไรที่พวกเขาทำลายเข้ามา รวมทั้งเครื่องมือที่เขาใช้ สำหรับ Data ที่คุณต้องการตัว Honeypot แบบ High-interaction

เข้าสู่ประเด็นที่ดูว่าสามารถทำอย่างไรจึงจะทำให้ Honeypot สามารถป้องกันองค์กรของคุณได้ พวกเขาจะพูดเกี่ยวกับความต่างของความต่างในการใช้ Honeypot ที่ต่างกัน Research Honeypots เองซึ่งปกป้องค์กรณ์ของคุณได้แต่พวกเขาก็สามารถใช้ข้อมูลที่ต้องการและกระทำการบุกรุกที่น่ากลัวได้ ข้อมูลที่น้อยใน technology ที่ต่างกัน แต่ Honeypot สามารถนำมารวมกันได้หนึ่งในปัญหาที่ดีที่สุดของความปลอดภัย โฉมหน้าของผู้ทำหน้าที่ซึ่งยังขาดข้อมูล หรือ ไหวพริบที่ดีบนสังคม Cyber สามารถป้องกันได้อย่างไรต่อผู้ที่เป็นศัตรู เมื่อพวกเขาไม่รู้การกระทำจึงทำให้ไม่รู้ว่าใครคือศัตรู และ ควรจะต้องป้องกันอย่างไรในการกระทำของ Attacker ทำไมถึงเลือกข้อมูลของความปลอดภัยที่จะต้องป้องกันต่างๆ ได้การวิจัย Honeypots การวางตัวโดยเข้ากับตำแหน่งที่เหมาะสมข้อมูลต่างๆ สามารถบอกได้ในส่วนของการวิเคราะห์ใน tools ที่ต่างกันจะรองรับผู้โจมตีที่ต่างกันมีการเตือนและคาดการณ์ล่วงหน้าได้ หรือสามารถรับรู้ถึงสาเหตุได้ ตัวอย่างหนึ่งที่ดีที่สุดของการใช้ Honeypot การวิจัยเป็นงานที่ต้องทำโดยผู้อาสาร่วมช่วยวิจัยไม่มีการหาผลประโยชน์จากการวิจัยด้าน Security ข้อมูลทุกอย่างบน Honeypot มากมายทั่วโลกนำมาเพื่อใช้ประโยชน์ในการวิจัย

2.8 ยุคที่สองของระบบที่ได้พัฒนา

ใน GenII (2nd. Generation) ตัว Honeynet เป็นขั้นต่อไปที่ได้พัฒนาต่อมาของตัว Honeynet technology โดยเป็นพื้นฐานของการรวมระหว่างวิธีการทั้งเก่าและใหม่ที่หลากหลาย ตัว GenII ของ Honeynet สามารถเกิดความยืดหยุ่นในการจัดการ และ ในส่วนของความปลอดภัยในตัวของ Honeynet ในส่วนของเอกสารนี้จะเป็นการเริ่มต้นของตัว technology และสามารถใช้เป็นแนวทางอย่างเป็นขั้นเป็นตอนว่าในการสร้างตัว Honeynet ในยุค 2nd. Generation เป็นอย่างไร อย่างไรก็ตามก่อนหน้าที่คุณจะอ่านต่อไป ได้ตั้งสมมุติฐานขึ้นก่อนว่าคุณได้อ่านและทำความเข้าใจ Concept มาเป็นอย่างดีแล้วว่าความเสี่ยง และ ส่งที่เป็นข้อเสนอของตัว Honeynet สามารถอ่านได้จากด้านบนมันมีผลกระทบรุนแรงทีต้องเข้าใจเป็นพื้นฐานก่อนจะอ่านในลายละเอียดต่อไป

ในเอกสารนี้จากต้นจนจบจะแนะนำถึงส่วนประกอบของ GenII ของ Honeynet ในส่วนแรกจะแนะนำตัว Concept ของ Honeywall ก่อนมันเป็นหนทางที่ผู้ดูแลระบบจะคอยจับตาเพื่อควบคุมตัวระบบเครือข่ายได้ ในส่วนของ Data Control นั้นเราได้ตัดมาเพื่อให้เห็นได้บางส่วนที่ผู้โจมตีได้กระทำบนระบบเครือข่ายของคุณ ในส่วนที่สามตัว Data Capture เป็นรายละเอียดของตัว method ที่ใช้ดูการกระทำของ Attacker ได้ครบถ้วนทั้งสองส่วนคือ Host Level และ Network Level ในส่วนของการแจ้งเตือนให้โดยอัตโนมัตินั้นมี method ที่คอยแจ้งให้ผู้ดูแลระบบได้ทราบถึงปัญหาได้เป็นอย่างดีและสุดท้ายคือส่วนของการทดสอบในปัจจุบันเส้นทางในการตรวจสอบการ configure ของขั้นตอนที่ผ่านมาในการอธิบายนั้นจะเน้นที่ Honeywall ที่ทำงานบน kernel 2.4.X และ บนสถาปัตยกรรม X86 หรือคุณอาจจะใช้ระบบอื่นที่ง่าย และ รองรับต่อการทำงาน

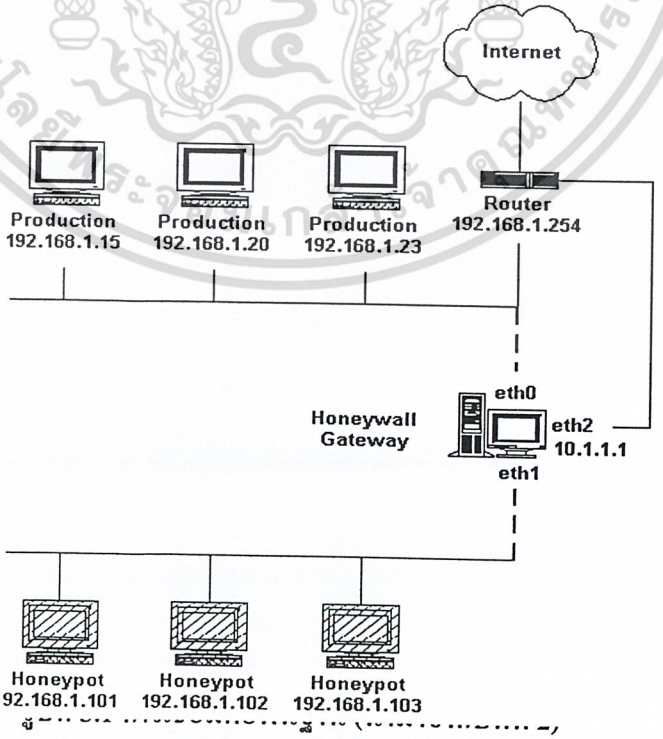
บทที่ 3
วิธีการแยกแยะผู้โจมตี กับ ผู้ใช้งานทั่วไป

ในส่วนนี้เป็นการหาวิธีการแยกแยะระหว่างผู้โจมตีระบบและผู้ที่เข้าใช้ระบบแบบปกติทั่วๆ ไป ในส่วนนี้จะยังกล่าวถึงการ ส่วนการควบคุมการกระทำในขณะที่ผู้โจมตีเข้ามาในระบบ (Data Control) จัดเก็บข้อมูลที่ได้ดักจับไว้ (Data capture) ในส่วนของการแจ้งเตือนให้ผู้ดูแลระบบรับทราบการเปลี่ยนแปลงของตัวระบบฮันนี่พ็อต (Data Alerting)

3.1 The Architecture

ตัว Honeypot ไม่ใช่ผลิตภัณฑ์ไม่ใช่ง่ายที่คุณจะติดตั้งจาก CD-ROM แล้วใช้งานมัน การควบคุมระบบ network ให้ได้ผลสูงสุดต้องใช้การรวมการวิเคราะห์การกระทำของผู้โจมตีที่เก่งกาจ แต่อย่างไรเสียตัวระบบก็ขึ้นอยู่กับคุณเองที่จะเลือก

ในส่วนของตัว Honeypot คือ Gateway เพื่อเป็นตัวแยกแยะว่าใครควรเป็นเหยื่อของตัว Honeypot สำหรับคนบนโลก ตัว Gateway จะกระทำตัวเหมือนกำแพง , ในความเป็นจริงควรเรียกว่า Honeywall ในส่วนนี้จะพูดถึงการส่ง และ ควบคุมศูนย์กลางของ Honeynet ทุกเหตุการณ์ที่อัศจรรย์เกิดขึ้นที่นี่ คุณสามารถเห็นได้จากตัวอย่างของพวกเราที่ได้นำเสนอตัว Honeynet ในรูป honeypot2



ตัว gateway ของเราเป็น layer 2 bridge หรือ layer 3 bridge การ routing gateway สามารถทำได้หาก bridge เอื้ออำนวย มันจะยากในขณะที่เราจะ detected ในขั้นตอนของพวกเรา Honeynet คือตัวที่เป็น internal network 192.168.1.0/24 ใน pass นี้มี Honeypot จำนวนมาก ซึ่งมีความสืบเนื่องออกมาจากส่วนภายนอก หรือขอบเขตของ networkกับการใช้ layer 2 gateway ตัว Honeynet สามารถจะรวมกันกับ internal network เข้ากันได้ ได้ยอมรับการติดต่อและเรียนรู้ไม่เพียงแต่ external ส่วนเดียว แต่ยังนำส่วน internal อีกด้วย

ตัว honeywall (เป็น Bridge Gateway) เป็นการแยกตั้งแต่เริ่มต้นตัวระบบจากตัว honeynet network ซึ่งจะบรรจุโดยเหยื่อของพวกเราโดย external interface คือ eth0 เป็นตัวคอยต่อตัว Honeynet เข้ากับ Network โดยภายในมี interface eth1 ซึ่งต่อกับตัว Honeynet เมื่อเราได้ bridge ตามนี้ คือ ทั้ง internet และ external ต่างมี IP เดียวกัน พวกเรามีอีก interface คือ eth2 จุดประสงค์ของ interface นี้เพื่อ remote admin ของตัว GW ทำการเพิ่มหรือย้าย logs หรือ จะจับตาดู data ส่งไปยังศูนย์กลางก็ได้ ในส่วน internal และ external interface จะต่อในลักษณะ bridge mode โดย ip address ที่เรากำหนดให้พวกมัน จะไม่มีกำหนดให้แต่อย่างไรแล้ว interface ที่ 3 จะมีการกำหนด ip ให้ ในตัวอย่างคือ 10.1.1.1 นี้เป็นตัว network ที่แยกออกมา ความปลอดภัยของ Network นี้เฉพาะผู้ดูแลระบบเท่านั้นประโยชน์ของรูปแบบ GW นี้คือ ยากที่จะตรวจจับ ไม่มี routing hop ไม่มีการลดค่า TTL จึงทำให้จับไม่ได้ว่ามี GW อยู่ ไม่มี MAC Address รวมเข้าใน GW ตัวนี้ดังนั้นเราสามารถเอาพื้นฐานของ honeynet มารวมเข้าด้วยกันโดยมี Data Control และ Data Capture บน Single gateway ในลำดับต่อไปคือการสร้างให้ GW รองรับกับรูปแบบที่เราได้ออกแบบไว้นี้สำหรับ GW ของพวกเรามีความปลอดภัยต่ำบนระบบ Linux มันจะถูก critical ได้เพราะเป็นระบบที่มีความน่าเชื่อถือได้สูง จะไม่มี Attacker คนไดเข้าสู่ GW ได้ ต่อไปพวกเราจะสามารถเข้าใจได้ว่า GW เรารองรับ ระบบ bridge มี Linux หลาย Distributions ที่รองรับการทำ bridge เป็น default หากว่าระบบของคุณไม่รองรับสามารถหา bridging rpm's หรือ source code จาก http://bridge.net

Gateway # rpm –q bridge –utils

Bridge –utils -0.9.3-4

เป็นโชคไม่ดีในขณะที่หลาก distributions รองรับ bridging แต่ก็มีมากมายที่ไม่รองรับ IPtable ในแบบ bridge mode IPtable คือ ตัวสำคัญไม่เท่านั้นมันยังช่วยป้องกันในด้านความปลอดภัยให้ GW ของพวกเราด้วย พวกเราจะใช้มันสำหรับ Data control คุณสามารถเลือก IPtables ขณะที่ bridge ของคุณทำงานคุณจะต้อง patch หรือ compile kernel ให้รองรับสิ่งเหล่านี้ คุณสามารถจะเลือก kernel version สุดท้าย และคุณก็เลือกให้รองรับ bridge ด้วยและ kernel จะ patch เพื่อให้รองรับกับ IPtables ใน bridge mode คุณสามารถ configure ตัว GW ได้ซึ่งตอนนี้จะทำได้อย่างที่ได้กล่าวมาแล้ว ตัว Honeypot project เองได้มี file ที่สร้างไว้ชื่อ rc.firewall ใน script นี้ได้สนับสนุนสิ่งจำเป็น

ในหลายส่วนมากมายของ GW ของคุณ พวกเราจะเปรียบเทียบตลอดทั้งเอกสารนี้ สำหรับ GW ของพวกเราตัว script จะสนับสนุนการทำ bridge เป็น firewall gateway, และ config เพื่อจัดการกับ interface ควบคุมโดยผู้ซึ่งสามารถยอมให้สามารถเข้ามาใน GW ได้เท่านั้น(ผู้ควบคุมระบบ) โดยเขาได้อย่างไร,จากที่ไหน มี log ของการกระทำของ NW และสนับสนุน Data control คุณจะเห็นได้ว่า script นี้สำคัญเพียงใด มัน configure ตัว GW ของคุณสำหรับความต้องการอันมากมายของ honeypot ไปสู่การสนับสนุน script นี้(และก็เช่นกันกับการ configure Honeywall ของคุณ) คุณจะจ่ายเมื่อจะ configure ค่า variable .ใน script และ run

- เตือน การ run script นี้จะไม่สามรถ run ได้หากคุณอ่านบทความนี้ไม่จบ การแทนค่าที่บ่งบอกอะไรต่างๆในรายละเอียดและอย่างไรมันถึงจะทำงาน(ตัว script ทำสิ่งนั้นสำหรับคุณ) เข้าอ่านได้ที่หัวข้อ complete script เพื่อใช้กับสถาปัตยกรรมของระบบเราได้บรรยายไว้ด้านบนอย่างแรกคุณมี file configuration script แล้วสำหรับองค์กรของคุณคุณก็พร้อมที่จะไม่อ่านส่วนของ Data control

3.2 Data control

Data control จะป้องกันเหล่าผู้โจมตีจะใช้ตัว honeypot เพื่อโจมตีหรือ ทำอันตรายในเครื่องอื่นๆที่ไม่ใช่ระบบ honeynet ตัว Data control จะลดอัตราเสี่ยงมันทำการกำจัดออกกับตัว Data control หนึ่งในคำถามที่คุณจะได้รับคำตอบว่ามันมากเท่าไหร่ทั้งการกระทำจากภายนอกที่มากระทำกับ Data control ของคุณ คุณรับเหล่า attacker เข้ามาทำสิ่งต่างๆทำให้คุณได้เรียนรู้มากทีเดียว แต่จำนวนมาเท่าไหร่และกับ attacker ที่จะเข้าทำในระบบคุณ แต่คุณไม่สามารถบรรจะข้อมูลที่มีมากมายในอะไรสักอย่างที่เล็กๆกะทัดรัดให้คุณเรียนรู้ได้มากเท่าไหร่ที่คุณรับและอนุญาตให้ attacker เจ้าทำในสิ่งที่ขึ้นอยู่บนความเสี่ยงที่คุณตั้งสมมุติฐานได้ ทำให้เหมือนกับว่าคุณต้องแข่งขันกับมัน พวกเราได้รวม attacker ที่เรารู้ไว้ ที่ความสำเร็จจำต้องทำ พวกเราได้ทำให้มันสนับสนุน 2 technology คือการต่อเข้ากับตัวนับและ NIPS(Network Instrusion Prevension System) ต่อเข้ากับตัวนับเพื่อนับว่าหากถึง limit ที่เข้ามาจำนวนมาก honeypot สามารถกำหนดได้ NIPSจะสามารถ block รูปแบบการกระทำได้,รวมทั้ง อย่างนี้เข้าไว้ด้วยกันจะทำให้มันช่วยกันทำงาน และเกิดความยืดหยุ่นที่ data control พวกเราจะรวมทั้ง 2 technology ไว้ที่ layer 2 GW ที่พวกเราทำให้ data control ไปจัดไว้ที่ GW เพราะว่าเป็นที่ซึ่งทั้งขาเข้าและขาของเดิน packet มากที่สุด เพื่อจัดการให้โยกย้าย Attacker ได้อย่างง่ายดายที่สุดสิ่งแรกที่คุณต้องมีคือการ config GW ในขั้นแรกโดยจัดให้รองรับการติดต่อที่มี limit ก่อนพวกเราจะกำหนดว่าสามารถ connect มาได้เท่าไหร่ และ attacker สามารถมาได้เท่าไหร่วัตถุประสงค์ที่ให้มีการนับการ connect ที่มาจากภายนอกและเมื่อจะติดต่อให้ limit เท่าใดเพื่อป้องกันการ connect ที่จะมีมากมายเป็นการลดอัตราเสี่ยงของการ scan การโจมตีหรือการ denial of service มันยากที่จะป้องกันไม่ให้เกิดโจมตีจำนวนมาก เมื่อคุณได้ limit ค่าจำนวนของการเชื่อมต่อ

คุณสามารถตั้งค่าเริ่มต้น outbound อย่างไรก็ตามมันก็ทำให้รู้สึกถึงการที่สามารถมาจากลักษณะเฉพาะได้อีกด้วยในการโจมตีบางทีอาจจะสามารถตรวจจับได้จาก honeynet ของคุณได้ โดยง่ายโดยการกำหนดค่าเริ่มต้นและจะเห็นว่าถ้าหากพวกเขาถูก block หลังจากกำหนดหมายเลขแล้วไม่มีพลาด ค่าพื้นฐานสำหรับการเชื่อมต่อจะกำหนดได้ใน re.firewall ซึ่งทำตามดังนี้ เตือน ตัวค่า variable อื่นๆที่เป็น IP Protocal มีNOT,TCP,UDP หรือ ICMP (เช่น IPsec,IPv6,tunneling,Network Voice Protocal และอื่นๆ)

Set the connection outbound limits for different protocols

SCALE="day"

TCPRATE="15"

UDPRATE="20"

ICMPRATE="50"

OTHERRATE="15"

นี้เป็นตัวบอกให้ว่า IPtable กำหนดดูค่า limit อย่างไรเมื่อผู้โจมตีทำงายและเข้าสู่ honeypot การกำหนดค่าการเชื่อมต่อขาออกจาก network จากบุคคลที่หลากหลาย (ทั้ง download toolkits,setup automated bots,IRC chat,send email, etc) ทุกๆๆเวลาใน 1 connection การกำหนดค่า outbound , ตัว firewall จะ counts พวกเขาเหล่านั้นเมื่อถึงlimit , ตัว IPtable จะ block ทุกการเชื่อมต่อที่ติดมา honeypot ฉะนั้น IPtables จะ return ทุกๆๆอันด้วยตัวมันเอง อนุญาตทุกการเชื่อมต่อ ในช่วงเวลาที่อนุญาตสำหรับตัวอย่างนี้เราได้อนุญาตให้ TCP มีขาออกที่ 25 การเชื่อมต่อ เมื่อผู้โจมตีเข้าสู่ตัว honeypot พวกเขาจะได้รับอนุญาตแค่ 25 การ connect เท่านั้น เมื่อถึง 25 TCP connect แล้ว พวกเขาไม่สามารถให้ IPtables เริ่มการนับให้อนุญาตอีกได้ จะได้รับอนุญาตอีกครั้งก็ต้องเป็น 24 ชั่วโมงข้างหน้า limit การเชื่อมต่อ ใน 1 ชั่วโมง ถ้าพวกเรา set ให้เป็นต่อ 1 ชั่วโมงจะกำหนดlimit การพบได้ มันจะอนุญาต 25 การเชื่อมต่อ อย่างต่อเนื่องใน 1 ชั่วโมง เฉลี่ยแล้วในทุก2.4 นาทีในโลกแห่งความเป็นจริงตัวอย่างของพฤติกรรมนี้จะเปรียบเทียบได้กับ IPtables logs ของ win2K ที่ honeypot ติดเชื้อ code red II worm และพยายาม scan ออกภายนอก

คุณมีการสนับสนุนโดยจาก limit กับ rc.firewall ขั้นต่อไปคือการทำให้สนับสนุน NIPS ซึ่งเป็น function จำได้ว่าเป้าหมายของ NIPSคือการจำแนก และ block เมื่อรู้ว่านี้คือการโจมตี มันทำหน้าที่นี้โดยจะสอดส่องดู Packet อื่นๆที่ผ่านเข้ามายัง GW ถ้าตัว packet นั้น match กับ IDS rules มันจะไม่เพียงแค่สร้างสัญญาณเตือน(เหมือน NIDS) แต่ packet จะถูก drop (ป้องกันการโจมตี) หรือเปลี่ยนแปลง(หยุดการโจมตี) จะมีประโยชน์ที่น่าทึ่งมาลดอัตราเสี่ยงของการโจมตีขาออกได้ ข้อเสียที่มีคือเมื่อมีผู้โจมตีที่รู้ระบบ ในกรณีของอัตรา limit เรากำหนดอนุญาตโดยปกติคือการเชื่อมต่อ TCP 15 การเชื่อมต่อ/วัน อะไรที่เกิดขึ้นถ้า honeynet ของคุณใช้นั้น ติดเชื้อตัวหนอนไวรัสและในสิ่งที่

กำหนดไว้ตอนแรกคือ การเชื่อมต่อ 15 การเชื่อมต่อขาออก มันเป็นการทดลองให้ระบบติดเชื้อหรือ? ในขณะที่อัตราการจำกัดมีการลดขนาดลงของระบบที่มันสามารถติดเชื้อคุณจะยังเสี่ยงต่ออันตรายต่ออยู่ดี แนวคิดของNIPS มันจะ block หรือ disable ที่รู้ใน 15 การเชื่อมต่อแรก สำหรับสิ่งนี้ตัว honeynet project ใช้ snort_inline การดัดแปลง พัฒนา version ของ snort มันจะสามารถดัดแปลง packets ได้

สำหรับ snort_inline ที่ทำงานเหมือนกับ NIPS(ใน mode GW) มันมีบางการ route packet สำหรับตัวมัน,snort_inline ไม่สามารถรู้ว่าจะ route อย่างไร(ip_forward)ด้วยเหตุนี้เราจึงมีการ route บาง packet สำหรับ,snort_inline และคอยจัดการ route ตัว process packet ส่งให้ snort_inline เพื่อจัดการวิเคราะห์ หนึ่งในสิ่งที่ snort_inlineทำกับ packet ,มันมีมือคอยจัดการบอกเส้นทางของ process มันเป็น process ของIPtables พวกเรา configure ตัว IPtables เพื่อจัดการกับ packets มันจะ forward ไปให้วางมันเหล่านั้นลงใน user space สำหรับ snort_inline เพื่อวิเคราะห์,ดังนั้น IPtables จะทำการ route packet อย่างต่อเนื่องมันเป็นส่วนเพิ่มที่เป็นของ IPtables คือ จะเรียกกันว่า user_space queuing ,มันจะทำหน้าที่ร้องขอตัว ip_queue module เพื่อ load เข้าสู่ kernel เพื่อให้ทำงานในส่วนนี้ได้ พวกเราจะเริ่มให้ทำการ enable QUEUE ในส่วนของพวกเราคือ rc.firewall script(ด้วยเช่นกันมันจะ enable ตัว kernel module ที่ชื่อว่า ip_queue หนึ่งในสิ่งที่จะทำให้รู้สึกเมื่อรวม snort_inline และ IPtables จะนับจำนวน, IPtables จะ count ใน ขาออกของการเชื่อมต่อ ถ้าตัว snort_inline อนุญาตให้ตัว packet ผ่านหรือจะ block packet เพราะทุกๆ packet ผ่านไปผ่านมาผ่านทาง internal interface เริ่มก่อนอื่นคือจะวิเคราะห์โดยตัว snort_inline การเชื่อมต่อมีการนับก่อนตัว snort_inline เคยพบมา

IPtables script can be used with the snort_inline filter

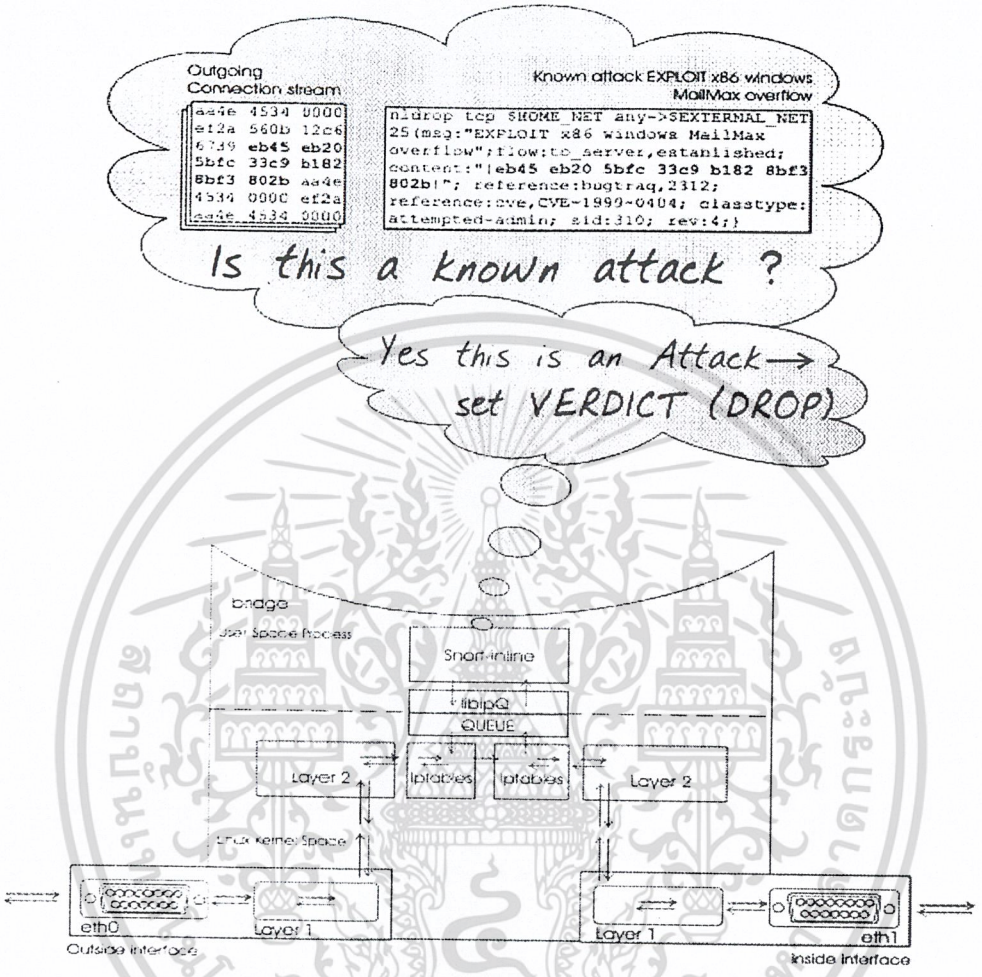
#QUEUE="no" **#Do not use experimental QUEUE support**

QUEUE="yes" **#Use experimental QUEUE support**

ถ้าคุณ enable ตัว snort_inline ที่ส่วนความจุ ดังนั้นคุณควรที่จะมี snort_inline ทำงานอยู่ถ้าคุณไม่มี snort_inline ไม่ทำงานตามสมควรจะไม่มี packets ใดๆจะผ่าน IPtables (นี้เป็นการป้องกันไม่ใช่เพราะเป็น bug) เช่น พวกเราทำขั้นต่อไปคือ configure ตัว snort_inline ในความเหมือนโดยทั่วไปของ snort คาดหวังว่ามันใช้ กฎที่ต่างออกไปในความรู้สึกพวกเรามีเป้าหมายคือไม่ drop ที่ขาออกของการผ่านของ packet ทั้งหมดของการโจมตีเดียวดังนั้น พวกเราต้องการใช้กฎที่เรากำหนดนั้นมีเพียงการโจมตีเดียวดังนั้น พวกเราต้องการใช้กฎที่เรากำหนดนั้นมีเพียงการโจมตีแบบหวังผล "exploits" คุณไม่ต้องการ block ข้อมูลขาออกเช่นICMP ping, finger, หรือโดยทั่วๆไปเช่น HTTP GET command ถ้าพวกเราใช้ ALL ของกฎ snort เมื่อนั้นผู้โจมตีจะโจมตีอีกครั้งไม่สามารถทำอะไรออกไปได้เลย เช่น พวกเราใช้กฎของ snort เพื่อให้เป็นการโจมตีซึ่งเป็นปัจจุบัน ในองค์กรณ์ต่างๆ

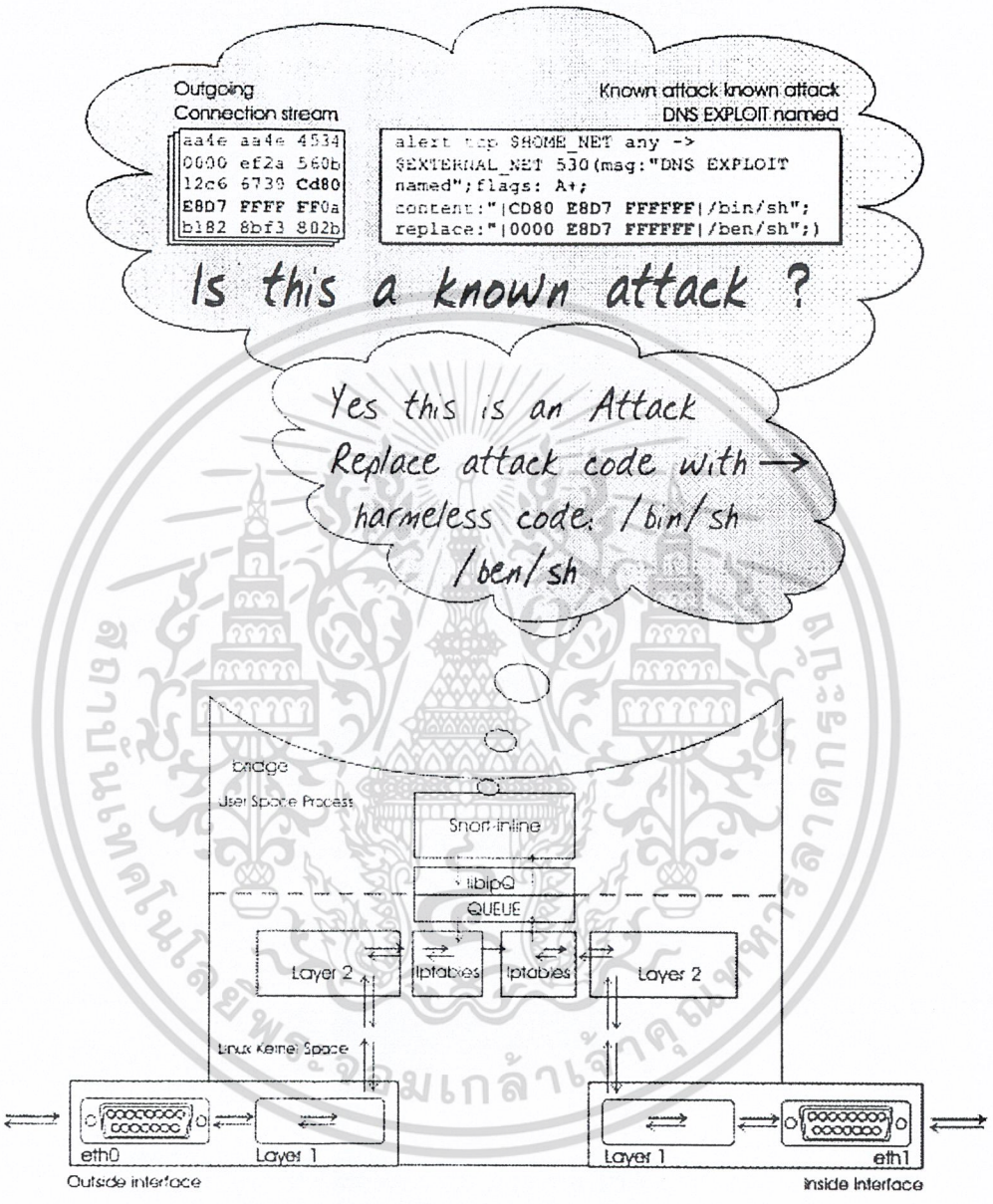
อาจจะมีความต่างของการกำหนดว่าอะไรคือการโจมตี ดังนั้นพวกเราจะแนะนำให้คุณพิจารณาดูดัดแปลงตัวกฎของ snort_inline ก่อนที่จะใช้พวกมัน ดังนั้นกฎของพวกเรามีส่วนที่กลับกันจากรูปแบบที่เป็นกฎทั่วๆ ไปของตัว snort_inline ที่เป็นกฎพื้นฐานโดยที่จะจับตาที่การโจมตีขาเข้ากัวตัว snort_inline นั้นพวกเราจับตาดูที่การโจมตีขาออก เป้าหมายของการป้องกันจากโลกภายของจากตัว honeynet สุดท้ายตัวกฎที่ต่างกันพวกเราจะไม่แจ้งเตือน บน การกระทำ แต่ในส่วนที่เป็นปัจจุบันนี้จะใช้การ drop หรือ การแก้ไขตัวรูปแบบการโจมตีเอา การ drop

DROP MODE OPERATION



รูปที่ 3.2 รูปแบบการดร็อบของระบบตรวจจับผู้บุกรุก

จะตั้งกฎเพื่อให้วัตถุประสงค์ที่เป็นส่วนของการวิเคราะห์ตัว packet และ ถ้าหากมันเห็นการโจมตีจากภายนอกการ block หรือ drop ตัว packet จะบรรจุการโจมตีไว้ ทำการ block หรือ drop ตัว packet ดูรูป honeypot3 ที่บรรจุการโจมตีพวกเราสามารถเห็นรูปแบบนี้ได้จากตัวอย่างของการ drop ตัว Code Red II ซึ่งเป็นการโจมตีที่เปรียบเทียบให้เห็นง่ายๆ โดยการใช้วิธีการวางทับ replace ruleset จะไม่ทำการ block ดังรูปแบบการดร็อบของระบบตรวจจับผู้บุกรุก



รูปที่ 3.3 รูปแบบการวางทับข้อมูลในแพ็กเก็ตของระบบตรวจจับผู้บุกรุก

การแทนที่มันคือการดัดแปลงตัว Contents ต่างๆ ที่เป็นการโจมตีในปัจจุบัน หยุดยังการเข้ายึดระบบ exploit นี้เป็นส่วนแยกของความต่างที่เป็นการควบคุมการกระทำสำหรับตรวจจับ attacker พวกเขาจะเห็น Attacker ต่างๆ นั้นไปถึงในเป้าหมายที่ได้เจตนาที่จะไปไว้แต่ก็ไม่สามารถแสดงออกมาได้ว่าทำไม Attacker จึงพลาดในการกระทำนั้น ตัวเครื่องมือ snortconfig เป็นตัว script ที่ช่วย

ทำให้เกิดความยืดหยุ่นนี้กลับกันกับจากกฎพื้นฐานของตัว snort_inline ที่สามารถทำได้ เช่น การ drop การ sdrop หรือ การ replace โดยที่เครื่องมือนี้มีความสำคัญเป็นอันมากรักษากฎของกฎที่ได้จัดเตรียมไว้ของ snort_inline

สำหรับวัตถุประสงค์ของ แปลงของพวกเรา พวกเราจะติดตั้งเป็นแบบ Drop ที่การจัดตั้งกฎ และ ในส่วน snort_inline.conf โดยการเข้าไป configure file ที่ /etc/snort_inline พกวเราใช้การแบ่งตัวแฟ้มข้อมูลจาก /etc/snort ดังนั้นจะไม่ทำให้พวกเขาสับสน ในการเริ่มต้นกับตัว snort_inline พวกเราจะใช้ snort_inline startup script ตัว script นี้เป็นตัวเริ่มต้นของการจัดค่าตัวแปรให้เหมาะสมกับ snort_inline ที่จะสามารถทำงานได้ มันทำให้ชวิตง่ายขึ้นเยอะเลยหล่ะ ทั้งหมดของตัว snort_inline config file, rules, startup script และ เริ่มทำตามนี้เหล่านี้ได้ถูกบรรจุเข้าไว้ใน Honeynet Snort_inline Toolkit ใช้เครื่องมือนี้สร้างส่วนย่อยของตัว snort_inline ที่ง่ายขึ้นทำตามที่ต้องการ รวมความทันสมัยเหล่านี้เพื่อลดความเสี่ยงแต่มันเหล่านั้นก็ไม่สามารถตัดความเสี่ยงออกได้ทั้งหมด ทุกเวลาที่คุณพบกัยผู้โจมตีในบ้านของคุณ สามารถเกิดในสิ่งที่เลวร้ายได้เสมอ

3.3 DataCapture

สิ่งหนึ่งที่เรามีการดัดแปลงตัว Data Control พวกเราก็สามารถที่จะทำกับ Data Capture ได้เช่นกัน วัตถุประสงค์ของตัว Data Capture คือ คือจะเก็บทุกๆ logs ของการกระทำที่เหล่าผู้โจมตีได้กระทำเอาไว้ เป็นกลอุบายของตัว Honeynet ที่จะเก็บข้อมูลเอาไว้ กับการเอา Data Capture ออก ตัว Honeynet เองจะไม่มีข้อมูลใดๆ เลย ตัว Data Capture เป็นส่วนสำคัญที่จะเก็บข้อมูลต่างๆ ในชั้นต่างๆ เท่าที่จะเป็นไปได้ เพราะว่าไม่ใช่เพียงชั้นเดียวที่จะสามารถมีข้อมูลให้ทุกอย่าง ตัวอย่างเช่น มีคนมากมายที่เดียวที่คิดว่าข้อมูลที่เค้าต้องการจาก Attacker คือการกด key แต่อย่างไรก็ตามนั่นยังไม่ใช่ความถูกต้องเสียทั้งหมด อะไรจะเกิดขึ้นเมื่อเหล่าผู้โจมตีได้ใช้เครื่องมือ อย่างไรที่จะรู้ว่าเครื่องมือนั้นทำงานอะไรถ้าคุณไม่จับตามองไปที่เครื่องมือนั้นๆ หรือ การจับ Network Traffic และระบบที่กำลังทำงานอยู่ พวกเราจะอธิบายการทำให้สนับสนุนของทั้งสามวิธีจากนี้

ตัว file logs ของ Filrewall จะมีทุกๆ ข้อมูลพื้นฐานพวกเราพร้อมแล้วที่จะทำในส่วนนี้โดยการนำ rc.filrewall script เข้ามาใช้พวกเราพร้อมที่จะจับตาการเชื่อมต่อ ทั้งขาเข้าและขาออก ไปสู่ file /var/log/message นี้เองจะเป็นข้อมูลที่สำคัญ เพราะว่ามันจะเป็นเครื่องบ่งชี้ว่า Attacker ทำอะไรอยู่ มันจะเตือนครั้งแรกเลยเมื่อเริ่มมีการเชื่อมต่อขาออก หรือ การโจมตีขาออกโดยพื้นฐานการทำลองงานนนี้ ตัว firewall จะตรวจสอบความล่อแหลมอย่างรวดเร็วๆในความต่างที่เป็นพฤติกรรมใหม่หรือ สพฤติกรรมที่ไม่รู้นั่นเอง ตัว script เองจะมี สี่ ความแตกต่างของการเดิน packet ดังนี้ TCP, UDP , ICMP และ OTHER จะมีตัว Data Control ตัว OTHER จะเข้ามาแทนทุกๆ อันที่เป็น non-IP proto 1, 6 หรือ 17 ชนิดเหล่านี้ มันจะมีความสนใจต่อเมื่อมีบางคนใช้ non-standard IP traffic เหล่านี้เหมือนกับเป็การพยายามโจมตีใหม่ๆ หรือ วิธีการที่ไม่เคยพบมาก่อน อาจจะเป็นช่องทางของ

Backdoor ที่เคยมีการค้นพบ อ่านได้ในบทความ Scan of the month 22 (http://www.honeynet.org/scans/scan22/)

ในส่วนที่สองเป็นการตรวจจับทุกๆ Packet และ มันมีประโยชน์มากในการจับทั้งเมื่อเข้ามาและ เมื่อออกไปจากตัว Honeynet ในขณะที่ตัว snort_inline กำลังดำเนินการกระทำอยู่นั้นอาจเป็นไปได้ว่ามันมีสิ่งนี้ "พวกเราจะไม่วางไข่ทั้งหมดลงในตระกล้าเดียวกัน" คือ แทนที่พวกเราจะ configure และ run ไปทั้งสองอย่าง เพื่อตรวจจับการกระทำของผู้โจมตี พวกเราทำได้โดยใช้มาตรฐานของ snort.conf configuration file นี้จะเป็นการ configure file เพื่อตรวจจับทุกๆ IP ที่สัญจรผ่านในเครือข่าย ไปสู้ tcpdump log file สำหรับนำมาวิเคราห์ต่อในภายหลัง ดังนั้น พวกเราต้องการที่จะเวียนใช้ log เป็นตัวเก็บการตรวจจับทุกๆ วันพวกเราสามารถทำได้กับ snort.sh start script จะเป็นตัวเริ่มการทำงานเพื่อให้ไม่มี cron ในทุกวัน แจ้งให้ทราบว่า จะทำอย่าไรในการใช้ startup script พวกเราจะสร้างการตรวจจับไว้ที่ส่วน interface ในคือ eth1 นี้เองจะเป็นส่วนที่จงใจทำให้ล่อแหลมจะผิดถนัดเลยหล่ะหากว่าคุณวางตัวการตรวจจับไว้ที่ interface ด้านนอก คือ eth0 มันจะได้ข้อมูลที่ไม่ใช่เพียงแต่ในส่วนของ Honeynet แต่คุณจะได้การสัญจรของ packet ที่อยู่ด้านนอกที่เป็นระบบเครือข่ายด้านนอกทั้งหมดอีกด้วย นี้เองจะทำให้เกิดเป็นข้อมูลขยะจำนวนมากที่คุณตรวจจับมาได้ โดยในหลักการที่เราใช้คือจะตรวจจับแต่เฉพาะส่วนในเท่านั้นคุณจะสามารถตรวจจับได้แค่ ด้านใน หรือ ด้านนอก จองการส่งผ่านข้อมูลจากตัว Honeynet เท่านั้นคุณจะได้ข้อมูลที่คุณต้องการแน่นอนเป็นข้อมูลที่ถูกต้องประโยชน์ต่างๆ ที่มีประโยชน์ของตัว startup script คือความเป็นมาตรฐานเดียวกันที่ซึ่งมีการตรวจจับข้อมูลที่เป็น log file สิ่งที่สำคัญที่สุดคือมี Honeynet จำนวนมากเข้าสู่ศูนย์ใหญ่ (จะอธิบายในส่วนหลัง)

ในส่วนที่สามนี้เป็นการท้าทายการดักจับเหล่าผู้โจมตีบนตัว Honeypot เองมันเป็นแบบง่ายมันมีการใช้กันมานานแล้วกับระบบที่เป็นแบบ cleartext protocol เช่น FTP, HTTP และ Telnet คุณแค่เพียงมีตัว sniff ที่ดักจับการกด keystroke อย่างไรก็ตาม ผู้โจมตี จะเหมือนกันกับคุณคือมีการเข้ารหัสของข้อมูลในปัจจุบันพวกเขาใช้ SSH หรือ 3DES เป็นช่องทางในการเชื่อมต่อการสื่อสารกันกับเครื่อง computer พวกเราจะไม่สามารถจับการกดคีย์(keystrokes) ผ่านทางสายได้อีกต่อไปแทนการที่จะจับพวกเขาจากระบบที่เขาใช้หนึ่งในประโยชน์ของระบบที่มีการเข้ารหัสนั้นคือเครื่องปลายทางในที่นี้คือเครื่อง honeypot ของคุณ ถ้าหากว่ามีการจับข้อมูลบนเครื่อง honeypot ของคุณ ซึ่งข้อมูลเป็นการ decryption เราสามารถส่งข้อมูลผ่านช่องทางที่ encryption ได้ ตัว Sebek เป็นเครื่องมือที่กระทำการนี้โดยตรงตัว Sebek จะซ่อนตัวอยู่ใน kernel ซึ่งเป็น module (หรือบางทีเรียกว่า patch) ความสามารถในการจับตาดูความเคลื่อนไหวของเหล่า Attacker หนึ่งในการติดตั้งตัว honeypot ตัว Sebek client จะทำงานอยู่ใน kernel ตัวข้อมูลจะถูกจับกลุ่มรวมกันโดย Sebek client โดยจะไม่เก็บไว้ในส่วนของตัว Honeypot เองเพราะว่ามันเป็นที่ซึ่งเหล่าผู้โจมตีจะสามารถเข้าไปค้นพบได้ แทนที่โดย ตัว Sebek client จะส่งข้อมูลผ่านทาง UDP เป็นการตรวจจับตัวเครื่องจักร เช่น ตัว Honetwall

Gateway หรือ การ Remote loggin ไปยังตัวระบบต่างๆ ผู้โจมตีจะไม่สามารถเห็น และ ก็ไม่เห็น packet ของตัว sniff เช่นกัน ราวกับว่าตัว Sebek Client นั้นซ่อนตัวอยู่บน Honeypot และหลบซ่อนไม่ให้เขาเหล่านั้นเห็น เหตุการณ์หากว่าผู้โจมตี download หรือ เขาใช้เครื่องมือในการดักจับ Sniffing Sebek เองจะทำงานโดยการซ่อนตัวจากพวกเขาเหล่านั้นนี้เป็นการทำโดยปรับปรุงตัว honeypot ให้มันไม่เห็นการดักจับ packet กับการออกแบบเริ่มต้นเป็นคล้ายกับเวทมนต์ที่ได้ทำไว้ไม่เห็นทั้งจำนวน และ port ของ UDP ที่เราเปิด ตัว Sebek เองได้ทำการส่งข้อมูลของเหล่าผู้โจมตีไปบนสายกับการตรวจจับโดยตัว Gateway ตั้งแต่นั้นมาตัว Honeypot เองก็ถูกควบคุมโดย Sebek ไม่มีผู้ใดที่จะสามารจับการกดคีย์ (keystroke) บนสายได้ (หมายเหตุ: ถ้าคุณมีตัว Honeypot แต่คุณเองไม่มี Sebek ติดตั้งอยู่ หรือ ตัว Sebek คุณติดตั้งแล้วทำการ Configure ไม่ถูกต้อง) และเหล่าผู้โจมตีจะเข้ามาควบคุมระบบของคุณได้ดังนั้นเมื่อเขาสามารถที่จะดักฟัง packet ที่เข้ามาจากระบบอื่นๆ เหล่า Packet จะไม่ถูกซ่อนตัวอีกต่อไป

ตัว Sebek นี้ทำงานใน Kernel โดยมีการ Compile มาสำหรับแต่ละชนิดของ OS และ Kernel version นั้นๆ ของ Honeypot ของคุณในขณะที่ Client รุ่นอื่นๆ ที่ต่างกัน และ ระบบปฏิบัติการที่ต่างกัน พวกมันจะมีการ Configure ระบบในลักษณะที่คล้ายกัน file ข้างล่างเป็นตัวอย่างวัตถุประสงค์ของการConfigure ตัวระบบเพื่อกำหนดข้อมูลที่ถูกรวบรวม และอย่างไรก็ตามข้อมูลจะถูกส่งไปตามสาย โดยปกติแล้ว Sebek เองจะตรวจจับทุกๆ การกระทำบนระบบอย่างไรก้ตามคุณก็ยังมีทางเลือกที่จะเลือกตรวจจับการกดคีย์เพียงอย่างเดียวก็ได้ การรวบรวมระหว่างหมายเลขอันน่าอัศจรรย์ (แล้วแต่ชนิดของ Sebek) และ ปลายทางของหมายเลข UDP port ที่ตัดสินใจว่า packet ไหนบ้างที่จะซ่อนตัวทุกๆ Honeypot ในกลุ่มเดียวกันจะมีการแบ่งการทำงานที่ร่วมกันได้เพื่อให้ได้มาซึ่งผลประโยชน์ที่แท้จริง

```
#----- INTERFACE:
INTERFACE="eth0"

#----- DESTINATION_IP:
DESTINATION_IP="10.0.0.1"

#----- DESTINATION_MAC:
DESTINATION_MAC="FF:FF:FF:FF:FF:FF"

#----- SOURCE_PORT:
```

SOURCE_PORT=1101

#----- DESTINATION_PORT:

DESTINATION_PORT=0

#----- MAGIC_VAL

MAGIC_VAL=0

#----- KEYSTROKE_ONLY:

KEYSTROKE_ONLY=0

#----- TESTING:

TESTING=0

หนึ่งในการ Configure Sebek เองจะนำข้อมูลทั้งหมดของระบบเข้าสู่ระบบเครือข่ายมี packet ที่พวกเขาเคยใช้สร้างขึ้นมาโดยเหล่าผู้โจมตี และ ทำให้สะดุดตาเมื่อมีผู้โจมตี โจมตีระบบของคุณมีการเรียนรู้เกี่ยวกับ Sebek สามารถอ่านได้จาก หัวข้อ Sebek

3.4 Alerting

มีอยู่อีกหนึ่งส่วนสำคัญที่เราจะต้องทำก่อนที่จะเสร็จสิ้นการทำ Honeynet ของคุณคือส่วนของการแจ้งเตือนมีบางคนจ้องที่จะเข้ามาในตัว Honeynet ของคุณเป็นการดีที่จะได้เรียนรู้ศึกษาในส่วนต่างๆ เว้นเสียแต่ว่าเราไม่ทราบว่าเขาเข้ามาข้างในระบบเราแล้ว ทำให้แน่ใจไปเลยว่าเราได้มีการแจ้งเตือน (และการตอบสนองของตัวระบบ) กับการกระทำที่ร้ายแรงและสำเร็จที่ตัว Honeynet แนวความคิดที่เหมาะที่สุดก็คือมี round-the-clock คอยสอดส่อง โดยอาศัยความชำชองของผู้ดูแลระบบอย่างไรก็ตามสำหรับองกรณ์ที่ไม่สามารถรองรับ 24/7 ของ staff ได้ หนึ่งในทางเลือกของการแจ้งเตือนที่เหมาะสมโดยหนึ่งในการเลือกใช้ระบบการเฝ้าระวังแบบอัตโนมัติคือ Swatch, the Simple Watcher โดยตัว Swatch คือเครื่องมือที่เป็นตัวเฝ้าระบบวังแบบอัตโนมัติมีความสามารถในการเตือนให้ผู้ดูแลระบบทราบถึงปัญหาที่เกิดขึ้นจากเผล่าผู้โจมตีได้อย่างสมบูรณ์แบบบนตัว Honeynet ตัว Swatch นั้นมีการเก็บ log file ไว้อย่างเป็นรูปแบบที่มาตรฐานขึ้นกับการ configure file ที่เราได้จัดการไว้ เมื่อมีรูปแบบอย่างที่เรากำหนดขึ้นและพบเข้าจะมีการแจ้งเตือนในหลายรูปแบบเช่น Email , ระบบเสียง , โทรศัพท์ และ สามารถเพิ่มเติมโดยการใช้ คำสั่ง หรือ โปรแกรม

ในกฎพื้นฐานของ Swatch มีมาตรฐานของรูปแบบการเฝ้าระวัง โดยกำหนดใน list ทำการ Configure ตามตัวอย่างโดยตัวอย่างใช้การกำหนด Configure บนระบบ Swatch 3.0

```
watchfor /Firewall:  OUTBOUND CONNECTION/
   echo normal
   mail=admin@honeynet.org,subject=------ ALERT! OUTBOUND CONN --------
   throttle 10:0:0
```

คำว่า mail= นี่เป็นตัวกำหนดการกระทำที่มีการบรรจุไว้แล้วในคำสั่งการ configure โดยจะมีการใน e-mail address list ไว้ในการส่งนั้นจะมีการกำหนดให้ส่งในทุกๆ Hour:Minute:Second เป็นรูปแบบของการกำหนด นี้เป็นการใช้งานในส่วนของการหลีกเลี่ยงการทำให้ mail box เต็มเมื่อเราใช้กฎนี้ จากด้านบนที่ได้กล่าวไปนั้น เป็นการ configure file การนำการแสดงผลอยู่ที่ /var/log/messages ที่ซึ่ง IPTables ที่เป็น log file ที่จัดเก็บทั้งการติดต่อ ทั้งขาเข้า และ ขาออก โดยพวกเราได้กำหนดการ configure file ให้เฝ้าระวังแค่เพียงเฉพาะการติดต่อ ขาออก เท่านั้น เป็นการ เฝ้าดูอย่างดี ที่เป็นการลดขนาดของการเฝ้าระวังของตัว Honeynet เมื่อมีรูปแบบที่ตรงกันกับที่ได้กำหนด จะมีการส่ง e-mail ไปให้ยังผู้ดูแลระบบ ในการ configure นี้มีการเตือนผ่านทาง e-mail ไม่มากไปกว่า 10 ฉบับใน 1 ชั่วโมง รูปแบบของการเฝ้าระวัง และ การกระทำจะเปลี่ยนแปลงระหว่างตัว Honeynet ที่ได้แยกตัวออกมา มีความสำคัญของการกรองเจ้าตัวข้อมูลที่สำคัญเก็บไว้ ของการ Configure ตัว Swatch

จะต้องมีความเข้าใจในเหตุการณ์ที่จะเกิดได้บนตัว honeynet โดยที่ผู้ดูแลระบบจะต้องทราบโดยการกำหนดการเฝ้าดูโดยการเก็บข้อมูลที่ควรค่าแก่การจัดเก็บเพื่อนำมาแจ้งเตือน ในจุดมุ่งหมายแท้จริงของการแจ้งเตือนโดบระบบอัตโนมัตินั้นจะจัดหาให้แก่ผู้ดูแลระบบ โดยจะมีข้อมูลปริมาณมาก แต่มีเฉพาะข้อมูลที่เป็นปัญหา ดังนั้น จะทำให้เราสามารถโต้ตอบกลับได้อย่างมีประสิทธิภาพ ทุกๆ การแจ้งเตือนที่ส่งมาโดย Swatch นั้นจะค่อนข้างมีขนาดเล็ก โดยมีเพียงแค่ source , destination IP และ source , destination PORT ที่เป็นการรู้มาจากตัว packet ที่ผ่านมา วันเวลา ที่เป็นการกระทำที่เกิดขึ้นขณะนั้น และ เป็นข้อมูลที่เพียงพอแล้วที่ผู้ดูแลระบบต้องการ โดยปกติแล้ว Swatch จะมีการเพิ่มบรรทัดของการบอกว่าเป็นการ match กับกฎใด เราได้มีตัวอย่างของ E-mail ที่ได้มีการส่งให้แก้ผู้ดูแลระบบดังนี้

To: admin@honeynet.org

From: yourdatacontrol@yourdomain.org

Subject: ------ ALERT!: OUTBOUND CONN --------

Apr 6 17:19:05 honeywall FIREWALL:OUTBOUND CONN UDP:IN=br0
PHYSIN=eth1 OUT=br0 PHYSOUT=eth2 SRC=192.168.1.101
DST=63.107.222.112 LEN=123 TOS=0x00 PREC=0x00 TTL=255 ID=43147
PROTO=UDP SPT=5353 DPT=79 LEN=103

เป็นเหตุการณ์ที่เกิดโดยการส่งมาจากเครื่องที่เป็น Data control เป็นผลให้ตัว Honeynet เองต้องการค่าที่ได้กำหนดมาจากผู้ดูแลระบบ การกำหนดค่าคุณสมบัติต่างๆ ตัว Swatch เองสามารถแจ้งให้ผู้ดูแลระบบทราบถึงเหตุการณ์ที่เกิดขึ้นกับระบบเครือข่ายของตนได้อย่างรวดเร็ว อย่างไรก็ตามมันไม่ได้ขึ้นอยู่กับการที่เรากำหนดค่าการติดต่อจากภายนอกที่เป็นส่วนของการแจ้งเตือนสำหรับตัวอย่างของผู้โจมตีที่บางทีอาจจะถูกผูกมัดกับตัวระบบ แต่ไม่ได้พยายามที่จะติดต่อออกไปภายนอกเพื่อความแน่นอนของการเฝ้าระวังค่าของข้อมูลต่างๆ เช่นการกดของ keyboard โดยการใช้ตัว Sebek client

3.5 วิธีการจำแนกแยกแยะผู้โจมตี กับ ผู้ใช้งานทั่วไปตามทฤษฎีของกลุ่ม

3.5.1 แนวคิดพื้นฐานเดิม

ในส่วนของการจำแนกแยกแยะนั้นความคิดตั้งต้น ตัวฮันนี่พ็อตเองได้เพียงแต่เสมือนกับว่าเป็นระบบที่ดูอ่อนแอกว่าวางร่วมอยู่กับระบบจริงเท่านั้นเมื่อผู้บุกรุกเข้ามาในระบบเครือข่ายส่วนนั้น ผู้บุกรุกเองจะสะดุดตากับระบบที่ดูอ่อนแอนั้น ระบบที่ว่านี้เรียกว่ากรงดัก (Cage) เป็นระบบที่มีการวางส่วนการรักษาความปลอดภัยในมุมมองของความเป็นจริงเสมือนกับว่าผู้บุกรุกได้เข้ามายังองค์กรหนึ่งซึ่งในองค์กรเองเปรียบเสมือนระบบเครือข่ายและห้องต่างๆ ที่อยู่ภายในองค์กรเปรียบเสมือนเครื่องที่อยู่ภายในระบบเครือข่าย ในเมื่อทุกห้องได้มีการรักษาความปลอดภัยเป็นอย่างดีมีการรูดบัตรผ่านทุกครั้งเพื่อแสดงตนก่อนเข้าไปยังห้องนั้นๆ แต่แล้วก็มีอยู่ห้องหนึ่งที่ไม่มีเครื่องอ่านบัตรเป็นเพียงประตูลูกบิดธรรมดาๆ บานหนึ่ง ผู้บุกรุกเองจึงคิดว่าห้องที่เป็นเพียงลูกบิดนี้ น่าจะเข้าได้ง่ายกว่าห้องอื่นใดในระบบเดียวกันนี้ ผู้บุกรุกจึงงัดรูกุญแจที่ลูกบิดแล้วก็ไขเข้าไปได้ ผู้บุกรุกก็คิดว่าตนได้เข้ามาภายในห้องที่ไร้การดูแลรักษาจึงเริ่มค้นหาสิ่งที่เป็นประโยชน์แก่ตน แต่แท้จริงแล้วห้องที่ผู้บุกรุกได้เข้ามานั้นเป็นห้องที่มีการเฝ้าระวังเป็นอย่างดี ไม่ว่าจะเป็นกล้องวงจรปิด ที่คอยสอดส่องดูว่าผู้บุกรุกกระทำพฤติกรรมใดอยู่ มีเครื่องตรวจจับการเคลื่อนไหวของสิ่งของภายในห้อง เพื่อดูว่าสิ่งของใดที่ถูกจับต้อง และ มีเครื่องตรวจจับเสียงเพื่อให้ทราบว่าผู้บุกรุกได้เอ่ยคำใดบ้างขณะที่อยู่ในกรงดักของเรา ผู้บุกรุกเอง ก็จะออกไปแล้วผู้ดูแลระบบเองก็จะนำการบันทึกเหล่านั้นมาศึกษาว่าผู้บุกรุกได้ทำอะไรไปบ้าง สิ่งใดเป็นพฤติกรรมที่น่าสนใจ แต่ทั้งนี้ทั้งนั้นสิ่งที่เกิดขึ้นนี้ต้องเป็นการเลือกที่จะเข้ามาในกับดักเองของผู้บุกรุก หากว่าผู้บุกรุกเองไม่ตัดสินใจที่จะเข้ามายังห้องที่ได้จัดไว้ทุกสิ่งที่ได้กล่าวมานั้นจะไม่มีทางเกิดขึ้นได้อย่างที่เราคาดหวังไว้ ระบบฮันนี่พ็อตเองก็เสมือนกับว่าไร้ค่า ดังนั้นแนวคิดเบื้องต้นจึงเป็นทิศทางที่ปูทางให้เป็นจุดกำเนิดแนวคิดที่ดีขึ้นไป ในส่วนของแนวคิดเบื้องต้นนั้นมีดังนี้

- เครื่องที่ทำตัวเป็นกับดักนั้นจะมีการเตรียมพร้อมที่จะดักและจัดเก็บข้อมูลทั้งหมดที่เกิดขึ้นภายในกับดักนั้นและส่งค่าไปยังเครื่องจัดเก็บข้อมูลหลักอีกทีหนึ่ง
- เครื่องที่เป็นตัวป้องกันไม่ให้ผู้บุกรุกใช้เครื่องกับดักเป็นฐานกำลัง นี้เป็นอีกส่วนหนึ่งที่ใช้ความสามารถของไฟร์วอลและระบบตรวจจับพฤติกรรมผู้บุกรุกจะกล่าวทั้งสองเครื่องมือนี้ในบทที่ 8

ข้อดีของระบบที่เป็นแนวคิดเบื้องต้นมีดังนี้

1. เมื่อผู้บุกรุกได้เข้ามายังเครื่องที่ได้จัดเตรียมไว้ ซึ่งได้เรียกว่าเครื่องกับดัก จะทำให้สามารถเก็บพฤติกรรมผู้บุกรุก

2. เมื่อผู้บุกรุกได้พยายามใช้เครื่องกับดักเป็นเครื่องเพื่อโจมตีเครื่องอื่นในระบบเครือข่ายหรือนอกเครือข่าย จะถูกกักกันการกระทำนั้นไม่ให้ชิ้นข้อมูลที่ส่งผลร้ายออกไปสู่ภายนอกระบบ

3. เป็นระบบหนึ่งที่ได้ช่วยให้เกิดความปลอดภัยมากขึ้นให้แก่ระบบเครือข่ายเนื่องจากว่าผู้บุกรุกไม่ได้เข้าโจมตีเครื่องที่ให้บริการจริงแต่อย่างไร

ข้อเสียของระบบที่เป็นแนวคิดเบื้องต้นมีดังนี้

1. เมื่อผู้บุกรุกไม่ได้เข้ามายังเครื่องที่ได้จัดเตรียมไว้ จะทำให้ระบบฮันนีพ็อต กลายเป็นเพียงเครื่องเครื่องหนึ่งที่ไม่ได้ใช้เพื่อประโยชน์ใดทั้งสิ้นในเรื่องของการรักษาความปลอดภัย

3.5.2 แนวคิดที่พัฒนาต่อ

เมื่อได้ทราบแนวคิดเบื้องต้นแล้ว จึงทำให้กลุ่มผมสามารถคิดค้นการทำงานที่สร้างความมีประสิทธิภาพสูงสุดให้แก่ระบบฮันนี่พ็อต เพื่อตัดจุดอ่อนที่ได้กล่าวไปแล้วข้างต้นว่า เมื่อผู้บุกรุกเองไม่ได้เข้ามายังเครื่องที่ได้จัดเตรียมไว้ ก็เป็นอันว่าระบบฮันนีพ็อตเองไม่มีความหมายใดๆ แต่เมื่อเราเชื่อมต่อความสามารถทั้งสองเครื่องมือที่ชื่อว่าเป็นไฟร์วอลและระบบตรวจสอบพฤติกรรมผู้บุกรุกซึ่งแต่เดิมไม่สามารถที่จะควบคุมระหว่างกันได้ เนื่องจากว่าแต่ละโปรแกรมก็ทำงานแยกจากกันอย่างอิสระ ดังนั้นกลุ่มผมเองจึงได้สร้างผู้เชื่อมต่อขึ้นมาซึ่งเรียกว่าเป็นผู้สื่อสารกลางให้ทำหน้าที่สื่อสารแปลความระหว่างไฟร์วอลและระบบตรวจสอบพฤติกรรมผู้บุกรุก ให้เป็นดังนี้

- เมื่อมีผู้บุกรุกเข้ามาในระบบซึ่งเรียกว่าประตูกลอัจฉริยะ honeywall เป็นเกตเวย์เพื่อคัดแยกระหว่างผู้บุกรุกกับผู้ใช้งานทั่วไป ให้ไปยังส่วนที่เหมาะสมตัวไฟร์วอลจะส่งชิ้นข้อมูลไปยังส่วนที่เรียกว่า ip_queue (QUEUE) เพื่อให้ระบบตรวจสอบพฤติกรรมผู้บุกรุกสามารถอ่านได้ต่อไป
- เมื่อชิ้นข้อมูลมาถึงในส่วนของ QUEUE ตัวโปรแกรมระบบตรวจสอบพฤติกรรมผู้บุกรุกเองจะดูว่าเข้าข่ายการโจมตีตามกฎที่มีหรือไม่ หากว่ามีให้แจ้งเตือน
- เมื่อแจ้งเตือนจะแจ้งมายังโปรแกรมผู้สื่อสารกลางซึ่งจะนำเอาการแจ้งเตือนของโปรแกรมระบบตรวจสอบพฤติกรรมผู้บุกรุกว่าควรบอกไปยังโปรแกรมไฟร์วอลอย่างไรให้เป็นไปอย่างเหมาะสม

- เมื่อตัวโรแกรมผู้สื่อสารกลางได้แปลงการแจ้งเตือนออกไปเป็นคำสั่งจะทำให้ไฟร์วอลเองได้เข้าใจว่าตนเองควรส่งให้ผู้ที่เข้ามานี้ไปยังเครื่องกับดักที่ได้จัดเตรียมไว้อย่างถูกต้องเหมาะสม

ข้อดีของระบบที่เป็นแนวคิดพัฒนาต่อมีดังนี้

1. เมื่อผู้ที่เข้ามาเป็นผู้บุกรุกจะถูกส่งเข้ามายังเครื่องที่เป็นกับดักที่ได้จัดเตรียมไว้อย่างแน่นอนและผู้บุกรุกเองก็ไม่มีทางที่จะรู้ตัวเลยด้วยซ้ำว่าตนได้เข้ามายังเครื่องที่เป็นกับดัก
2. เมื่อผู้บุกรุกต้องการใช้เครื่องที่เป็นกับดักโจมตีเครื่องที่อยู่ภายในเครือข่ายเดียวกันหรือภายนอกเครือข่ายจะถูกเครื่องที่เป็นประตูกลอัจฉริยะป้องกันไม่ให้กระทำได้
3. ระบบเครือข่ายที่ติดตั้งระบบฮันนีพ็อตจะมีความปลอดภัยมากขึ้นและมีส่วนดีที่จะได้ประโยชน์ในการได้รู้พฤติกรรมของผู้บุกรุกว่าได้กระทำอะไรเมื่อได้เข้ามายังกับดัก
4. ได้พัฒนาส่วนของการดูแลกับดักเพื่อไม่ให้ระบบที่เป็นกับดักถูกทำลายจะยากที่จะควบคุมให้เป็นไปตามความต้องการที่ว่าไม่ให้เป็นฐานกำลังแก่ผู้บุกรุก
5. สะดวกในการติดตั้งเพราะได้พัฒนาบนหลักการที่ว่า วางในส่วนใดของระบบเครือข่ายได้ และมีโปรแกรมการติดตั้งพร้อมการทำฮาร์ดเดนนิงระดับแรกไว้ให้เรียบร้อยแล้ว

ข้อเสียของระบบที่เป็นแนวคิดพัฒนาต่อมีดังนี้

1. เครื่องที่ใช้เป็นประตูกลอัจฉริยะนั้นจะต้องมีประสิทธิภาพสูงพอควรเนื่องจากว่าต้องคอยจัดสรรเส้นทางที่เหมาะสมให้แก่ผู้ที่ได้เข้ามาในส่วนนี้
2. เครื่องที่เป็นกับดักต้องวางตัวอยู่บนเครื่องเสมือนหรือที่เรียกว่าเวอร์ชวลแมชชีนซึ่งเวอร์ชวลแมชชีนเองต้องวางตัวอยู่บนลีนุกซ์ ซึ่งเรียกได้ว่าเป็นเวอร์ชวลแมชชีนเซิร์ฟเวอร์

3.5.3 แนวคิดที่จะพัฒนาต่อในอนาคต

เนื่องจากเวลาที่มีจำกัด ฮันนี่พ็อตในปีนี้จึงมีความก้าวหน้าได้เพียงแต่สามารถจำแนกแยกแยะผู้บุกรุกได้เท่านั้น แต่ก็ได้เสริมในส่วนของการดูแลกับดักไว้บ้างเพื่อใช้เวลาอย่างคุ้มค่า แต่แล้วก็ยังขาดอยู่หลายส่วนที่มีความจำเป็นมีดังนี้

- ในส่วนของเครื่องที่เป็นกับดักนั้นต้องการให้มีความแนบเนียนเพื่อไม่ให้ผู้บุกรุกรู้ได้ว่าถูกหลอกให้เข้ามาในเครื่องที่ไม่ใช่เป้าหมายของตน จึงต้องทำให้แนบเนียนดังนี้
  - เมื่อผู้บุกรุกใช้คำสั่งที่เป็นการแสดงตัวตนของเครื่อง จะต้องไม่สามารถรู้ได้ว่าตนเองมาอยู่บนเครื่องที่ไม่ใช่ แนะนำให้แก้ไขที่ kernel source หรือ สร้าง script หรือแนวคิดที่ดีกว่า
- ในส่วนของการส่งเข้าไปยังกับดักนั้นให้มีได้มากกว่าหนึ่งกับดักเนื่องจากว่าที่ได้ทำเป็นเพียงส่วนเริ่มหากต้องการให้เป็นหลายเครื่องก็ทำซ้ำโดยการวางคนละห้องและตั้งค่าหมายเลขเครื่องและกฎของระบบตรวจจับพฤติกรรมผู้บุกรุกตามต้องการ แต่หากจะทำแล้วมีวิธีการดังนี้
  - เข้าแก้ไข source code ของโปรแกรมผู้แปลงสารและให้มีส่วนที่เป็นตัวกำหนดว่ามีกี่กับดักมีหมายเลขประจำเครื่องหรือไอพีอะไรบ้าง ห้ามทำเป็นอาร์เรย์ธรรมดา ให้ทำเป็นโครงสร้างอาร์เรย์หรือลิงค์ลิสท์ได้ เพื่อให้รวดเร็วในการทำงาน มีความปลอดภัย และดีบักเพื่อป้องกันการเกิดเซกเมนเตชั่นฟอลท์ได้
- ในส่วนการดูแลเครื่องกับดักว่าใกล้ถึงจุดที่ยากต่อการควบคุมแล้วหรือยังให้มีการทำงานที่ควรมีอย่างน้อยดังนี้
  - เมื่อกับดักเองถึงจุดต้องจัดเก็บให้จัดเก็บตัวอิมเมจของตัวกับดักและนำอิมเมจที่สมบูรณ์มาวางแทนอย่างอัตโนมัติ หรือเลือกได้ ฉะนั้นต้องมีไฟล์ที่เป็นตัวกำหนดหรือคอนฟิกกูเรชันไฟล์
  - จัดเก็บเนื้อความที่สามารถดักจับได้ที่อยู่ในฐานข้อมูลมาจัดเก็บไว้ให้เหมาะสม
  - เมื่อต้องการนำอิมเมจที่บอบช้ำนั้นมาศึกษาต้องสามารถทำได้โดยง่ายเช่นมี script ที่ทำงานให้อย่างอัตโนมัติ
  - เมื่อนำกับดักที่บอบช้ำนั้นมาทำงานต้องควบคุมการกระทำที่เป็นลูกโซ่อยู่ได้

บทที่ 4
ไฟร์วอล

4.1 คุณสมบัติทั่วไปของไฟร์วอล

ไฟร์วอลเป็นเครื่องมือรักษาความปลอดภัยที่ทำงานในเชิงป้องกัน ซึ่งทำหน้าที่ในการควบคุมการเข้าถึงระบบเครือข่าย โดยอาศัยกฎเป็นพื้นฐาน (Rule based) สำหรับคุณสมบัติแต่ละอย่างของไฟร์วอลมีรายละเอียดดังนี้

1. การป้องกัน (Protect)
   ไฟร์วอลเป็นเครื่องมือที่ใช้งานในเชิงป้องกัน โดยชิ้นข้อมูลที่ผ่านได้นั้น จะต้องเป็นชิ้นข้อมูลที่ไฟร์วอลเห็นว่ามีความปลอดภัย ชิ้นข้อมูลที่ไฟร์วอลเห็นว่าไม่ปลอดภัยหรืออาจจะนำมาซึ่งความไม่ปลอดภัยก็จะถูกกำจัด คือไม่ส่งต่อ โดยการที่ไฟร์วอลจะตัดสินใจว่าชิ้นข้อมูลใดปลอดภัยและชิ้นข้อมูลใดไม่ปลอดภัยนั้นจะขึ้นอยู่กับกฎของผู้ดูแลไฟร์วอล เป็นผู้กำหนดไว้ล่วงหน้า ซึ่งเงื่อนไขของกฎเหล่านี้เองทำให้ไฟร์วอลสามารถป้องกันชิ้นข้อมูลที่อาจจะส่งผลร้ายไม่ให้ผ่านเข้าไปถึงเครือข่ายคอมพิวเตอร์ได้

2. ควบคุมการเข้าถึง (Access Control)
   การเข้าถึง หมายถึงการที่เครื่องใดเครื่องหนึ่งนั้นสามารถสื่อสารข้อมูลที่ต้องการไปยังเครื่องปลายทางได้สำเร็จ การเข้าถึงในแต่ละระดับจะมีวิธีการแตกต่างกันออกไป ทำให้การควบคุมการเข้าถึงสำหรับแต่ละระดับแตกต่างกันออกไปด้วย ไฟร์วอลเองจึงต้องมีการทำงานหลายลักษณะตามวิธีที่ไฟวอลใช้ควบคุมการเข้าถึง

3. กฎพื้นฐาน (Rule Based)
   ไฟร์วอลจะควบคุมการเข้าถึงโดยอาศัยหลักการเปรียบเทียบคุณสมบัติของชิ้นข้อมูลที่จะผ่านไฟรวอลกับกฎของการเข้าถึงที่ได้กำหนดไว้ หากพบว่าไม่มีกฎที่ห้ามไว้ก็จะอนุญาตให้ชิ้นข้อมูลนั้นผ่านได้ หากมีกฎที่ห้ามไว้ชิ้นข้อมูลนั้นก็จะถูกสกัดกั้นไว้ด้วยวิธีใดวิธีหนึ่ง

4.2 ประเภทของไฟร์วอล

ไฟร์วอลสามารถจำแนกประเภทจากลักษณะการทำงานได้ดังนี้

4.2.1 แพ็กเก็ตฟิลเตอริงไฟร์วอล / สกรีนิ่งเราเตอร์ (Packet Filtering Firewall/Screen Router)

เป็นไฟร์วอลพื้นฐานที่มีความสามารถ ในการควบคุมแทรฟฟิกโดยอาศัยการตรวจสอบข้อมูลที่ปรากฏอยู่ในแพ็กเก็ต ไฟร์วอลประเภทนี้อาจจะเป็นความสามารถที่เพิ่มเติมมาในเราเตอร์ โดยอาศัยโครงสร้างพื้นฐานที่เราเตอร์มีอยู่ให้ทำหน้าที่มากกว่าการจัดเส้นทาง ให้ชิ้นข้อมูลไปตามทิศทางที่เหมาะสมเพียงอย่างเดียว แต่จะทำการตรวจสอบเปรียบเทียบกับเงื่อนไขที่กำหนดไว้ก่อนจึงจะทำการจัดการเส้นทางชิ้นข้อมูลออกไป

ก่อนที่จะรู้จักการทำงานของแพ็กเก็ตฟิลเตอริงไฟร์วอลนั้น จะขอกล่าวถึงองค์ประกอบของแพ็กเก็ตเสียก่อน

แพ็กเก็ตเป็นหน่วยพื้นฐานของการรับส่งข้อมูลชั้นเน็ตเวิร์คเลเยอร์ การรับส่งข้อมูลแต่ละครั้งของเน็ตเวิร์คเลเยอร์จะส่งข้อมูลออกไปชุดหนึ่ง โดยที่ความยาวของข้อมูลนี้จะมีค่าเท่าใดนั้นจะเป็นไปตามคุณสมบัติของเน็ตเวิร์คเลเยอร์นั้นๆ ข้อมูลแต่ละชุดนั้นเรียกว่าแพ็คเก็ต สำหรับโปรโตคอล ทีซีพี/ไอพี นั้นจะใช้ ไอพีเป็นโปรโตคอลหลักในการขนส่งข้อมูลระหว่างเครื่องโดยใช้ไอพีซึ่งอยู่ในอินเตอร์เน็ตเลเยอร์จะส่งข้อมูลลงไปยังเน็ตเวิร์คเลเยอร์ตามลำดับ โดยที่หากขนาดของดาต้าแกรม (ไอพีดาต้าแกรม) ที่จะส่งนั้นสามารถส่งไปได้โดยใช้แพ็กเก็ตเดียว ไอพีก็จะส่งดาต้าแกรมนั้นไปทันที และแพ็กเก็ตนั้นคือข้อมูลของไอพี 1 ดาต้าแกรม แต่หากขนาดของไอพีดาต้าแกรมใหญ่กว่าขนาดของเน็ตเวิร์คเลเยอร์แล้วไอพีก็จะต้องทำการแบ่งส่วนหรือ การแฟร็กเม้นเตชั้น คือการกระจายดาต้าแกรมออกเป็นส่วนย่อยเสียก่อนแล้วจึงค่อยส่งลงไปที่เน็ตเวิร์คเลเยอร์ ซึ่งในกรณีนี้ข้อมูล 1 แพ็กเก็ตจะเป็นเพียงส่วนย่อยหรือแฟร็กเม้นต์หนึ่งของดาต้าแกรมเท่านั้น ดังนั้นข้อมูล 1 แพ็กเก็ตจึงไม่จำเป็นต้องเป็นข้อมูล 1 ดาต้าแกรมเสมอไป แต่อย่างไรก็ตามแพ็กเก็ตทุกชิ้นที่ส่งมาจากไอพีจะมีข้อมูลอย่างน้อยที่สุดคือ ไอพีแอดเดรสต้นทางและปลายทางเสมอ ซึ่งจำเป็นสำหรับให้แพ็กเก็ตนั้นวิ่งออกไปจนถึงที่หมายปลายทางได้

หากข้อมูล 1 แพ็กเก็ตนั้นบรรจุครบถ้วนทั้ง ไอพีด้าต้าแกรมแล้ว ก็จะทำให้สามารถทราบถึงข้อมูลของโปรโตคอลเลเยอร์ที่สูงขึ้นไปด้วยว่าเป็น ไอซีเอ็มพี, ไอซีเอ็มพี, ทีซีพี, ยูดีพี หรือโปรโตคอลอื่นใดที่อาศัยอยู่ใน ไอพีดาต้าแกรม นั้นแต่หากแพ็กเก็ตนั้นไม่สามารถบรรจุข้อมูลได้ครบทั้งดาต้าแกรมแล้วก็จะทำให้เพียงแต่ทราบว่าแพ็กเก็ตนั้นเป็นไอพีแพ็กเก็ตเท่านั้น

ข้อมูลที่สำคัญของแพ็กเก็ต

ภายในแพ็กเก็ตแต่ละแพ็กเก็ตนั้นจะประกอบด้วยข้อมูลที่สำคัญซึ่งสามารถนำมาใช้เพื่อเป็นเงื่อนไขสำหรับการควบคุมแทรฟฟิก โดยไฟร์วอลดังนี้

1. **ไอพีแอดเดรส ต้นทาง**
   ไอพีแอดเดรสของต้นทาง เพื่อใช้ในการพิจารณาต้นทางของข้อมูลว่าอยู่ในเงื่อนไขที่อนุญาตหรือไม่
2. ไอพีแอดเดรสปลายทาง
   ไอพีแอดเดรสของปลายทาง เพื่อใช้ในการพิจารณาปลายทางของข้อมูลว่าอยู่ในเงื่อนไขที่อนุญาตหรือไม่
3. โปรโตคอล
   ระบุโปรโตคอลที่อาศัยอยู่ในไอพีดาต้าแกรม ที่กำลังพิจารณานี้
4. พอร์ตต้นทาง
   ระบุพอร์ตต้นทางสำหรับโปรโตคอลที่ใช้พอร์ตคือ ทีซีพี และ ยูดีพี ซึ่งข้อมูลพอร์ตต้นทางนี้ส่วนใหญ่มักจะมีความสำคัญในลำดับรองลงไป และ ไม่ถูกนำมาใช้ควบคุมแทรฟฟิกมากนัก
5. พอร์ตปลายทาง
   ระบุพอร์ตปลายทางที่แพ็กเก็ตนี้ต้องการติดต่อด้วยสำหรับโปรโตคอลที่ใช้พอร์ตเช่น ทีซีพี และ ยูดีพี
6. ข้อมูลสำคัญอื่นๆ ตามลักษณะของโปรโตคอลเช่น ทีซีพี แฟลก, ไอซีเอ็มพีเมสเซสจ์ เป็นต้น

ข้อมูลทั้ง 6 ส่วนนี้จะมีได้อย่างครบถ้วนสมบูรณ์ก็ต่อเมื่อแพ็กเก็ตนั้นมีข้อมูลครบถ้วนทั้งหมดของ ไอพีดาต้าแกรมหากข้อมูลแพ็กเก็ตนั้นเป็นแฟรกเมนต์ อาจจะทำให้ข้อมูลในส่วนที่ 3 เป็นต้นไปซึ่งอยู่ในโปรโตคอลที่เลเยอร์สูงกว่าไอพีไม่สมบูรณ์ อย่างไรก็ตามไฟร์วอลส่วนใหญ่ทำการติดตั้งใช้งานในเครือข่ายซึ่งมีขนาดของแพ็กเก็ตที่ใหญ่พอสำหรับรองรับไอพีดาต้าแกรมได้ทั้งหมดจึงมักไม่ค่อยมีปัญหาแต่อย่างใดยกเว้นมีการแฟร็กเมนต์โดยความต้องการของไอพีเอง

จากข้อมูลที่สำคัญของแพ็กเก็ตข้างต้นนี้ จะสามารถนำมาใช้เป็นเงื่อนไขสำหรับควบคุมการผ่านเข้าออกของข้อมูลได้ โดยการพิจารณาข้อมูลทั้งหมดให้เป็นไปตามกฎที่ระบุไว้ซึ่งเรียกว่า แอคเซสรูลหรือกฎของการควบคุมการผ่านเออกของชิ้นข้อมูล โดยทั่วไปรูปแบบของการแอคเซสรูลเบื้องต้นจะเป็นดังนี้

Source Address	Destination Address	Protocol	Service (Dest.Port)	Action

รูปที่ 4.1 รูปแบบของการสร้างกฎของไอพีเทเบิลส์

โดยที่ข้อมูลทั้งหมดจะเป็นเสมือนตัวแปรที่จะนำมาเปรียบเทียบกับค่าที่ได้ระบุไว้ในแอคเซสรูลทีละค่า เงื่อนไขในการเปรียบเทียบของแต่ละตัวแปรจะเป็นตรรกะ "และ" ส่วนข้อมูลฟิลด์สุดท้ายหมายถึงสิ่งที่ไฟร์วอลกระทำเมื่อค่าในแพ็กเก็ตนั้นตรงกับเงื่อนไข

ตัวอย่างเช่น

Source Address	Destination Address	Protocol	Service (Dest.Port)	Action
192.168.5.19	ANY	TCP	80	Accept

รูปที่ 4.2 รูปแบบตัวอย่างของการสร้างกฎของไอพีเทเบิลส์

นั่นหมายถึงแอคเซสรูลนี้ได้ระบุไว้ว่าจะอนุญาตให้แพ็กเก็ตที่มีต้นทาง ไอพีแอดเดรส 192.168.5.19 และ ปลายทางใดๆ และใช้โปรโตคอล ทีซีพี และหมายถึงพอร์ตปลายทางเท่ากับ 80 ผ่านไฟร์วอลไปได้ หากไฟร์วอลมีแอคเซสรูลนี้เพียงข้อเดียว ก็เท่ากับอนุญาตให้โฮสต์เพียงโฮสต์เดียวคือ โฮสต์ที่มีไอพีแอดเดรส 192.168.5.19 เท่านั้นที่สามารถใช้บริการเอชทีทีพี (เอชทีทีพี พอร์ต 80) ไปยังโฮสต์อื่นที่อยู่อีกฟากหนึ่งของไฟร์วอลได้

นี่เป็นเพียงหลักการพื้นฐานในการควบคุมแทรฟฟิกของแพ็กเก็ตฟิลเตอร์ริงไฟร์วอลและไฟร์วอลชนิดนี้โดยทั่วไปจะเรียกว่าสกรีนิ่งเราเตอร์ เพราะว่าเป็นการทำเอาเราเตอร์ทั่วไปที่มีความสามารถกำหนดกฎการเข้าถึงมาดัดแปลงใช้ในการควบคุมแทรฟฟิก ซึ่งการกำหนดแอคเซสรูลของแทรฟฟิกทำได้โดยพิจารณาจากข้อมูลของแต่ละชิ้นข้อมูล แต่เรื่องจากเราเตอร์เป็นอุปกรณ์ที่มีพื้นฐานจากการทำงานในอินเตอร์เน็ตเลเยอร์ ทำหน้าที่เราต์แพ็กเก็ต เป็นหลักโดยพิจารณาจากไอพีแอดเดรส ทั้งต้นทางและปลายทางเท่านั้น สำหรับข้อมูลในส่วนของโปรโตคอลในเลเยอร์สูงขึ้นไป เช่น ทีซีพี, ยูดีพี, ไอซีเอ็มพี นั้นเนื่องจากเราเตอร์มีขีดจำกัดในการรับรู้ข้อมูลในเลเยอร์ถัดขึ้นไป คือ ทรานสปอร์ตเลเยอร์ จึงทำให้สามารถควบคุมแทรฟฟิกโดยระบุเงื่อนไขของโปรโตคอลในทรานสปอร์ตได้อย่างจำกัด คือจะสามารถควบคุมแทรฟฟิกได้เฉพาะเมื่อข้อมูลในทรานสปอร์ตเลเยอร์นั้นจะ

สามารถบรรจุได้ในแพ็กเก็ตเดียว หากมีการแฟร็กเมนต์และต้องเชื่อมโยงกันระหว่างหลายแพ็กเก็ต แล้วเราเตอร์จะไม่สามารถรับรู้การเชื่อมต่อนั้นได้

ข้อดีของแพ็กเก็ตฟิลเตอร์ริง

1. ราคาถูกเพราะเป็นคุณสมบัติที่มักมีในเราเตอร์อยู่แล้ว อาศัยเพียงการกำหนดแอคเซสรูลที่เหมาะสมเท่านั้น หากยังไม่มีไฟร์วอลอยู่เลย ก็สามารถใช้เพื่อช่วยป้องกันเน็ตเวิร์กภายในได้ดีพอสมควรในระดับหนึ่ง
2. หากเน็ตเวิร์คภายในใหญ่มาก และมีการใช้งานอินเตอร์เน็ตอย่างจำกัด ก็สามารถใช้ทดแทนไฟร์วอลได้ทันที
3. การใช้สกรีนิ่งเราเตอร์ควบคู่กับไฟร์วอลจะเป็นการแบ่งเบาภาระของไฟร์วอลได้มาก หากทำการกำหนดแอคเซสรูลได้อย่างสอดคล้องกันแล้ว จะทำให้มีการป้องกันที่เข้มแข็ง
4. การป้องกันบางประเภทไม่สามารถป้องกันได้โดยไฟร์วอล จะต้องทำโดยการกำหนดที่เราเตอร์เท่านั้น

ข้อเสียของแพ็กเก็ตฟิลเตอร์ริง

1. การกำหนดแอคเซสรูลทำได้ยาก ไม่มีระบบยูสเซอร์อินเตอร์เฟสช่วยในการทำงาน ส่วนใหญ่ก็จะใช้วิธีเทลเน็ตเข้าไปยังเราเตอร์ แล้วป้อนคำสั่งในลักษณะของคอมมารด์ไลน์เข้าไปโดยตรงที่เราเตอร์ ทำให้โอกาสที่จะกำหนดผิดพลาดเนื่องจากการป้อนข้อมูลผิดรูปแบบเป็นไปได้สูง
2. คำสั่งในการทำงานจะผูกติดกับยี่ห้อของเราเตอร์ ไม่มีมาตรฐานของคำสั่ง หากเปลี่ยนยี่ห้อของเราเตอร์ก็ต้องศึกษารูปแบบของคำสั่งใหม่
3. ไม่สามารถกำหนดกฎที่ซับซ้อนได้เนื่องจากขีดจำกัดของเราเตอร์ที่ทำงานโดยพิจารณาครั้งละแพ็กเก็ตเท่านั้น
4. มีความสามารถจำกัด เช่นไม่สามารถ บันทึกข้อมูลที่สำคัญ (Log) ของแพ็กเก็ตที่ต้องสงสัยไว้ตรวจสอบภายหลังได้
5. เราเตอร์มีกำลังในการประมวลผลจำกัด หากระบบเครือข่ายมีขนาดใหญ่ และมีการสื่อสารข้อมูลหนาแน่น เราเตอร์จะทำงานหนักอยู่แล้ว เมื่อต้องมาทำการประมวลผลแอคเซสรูลด้วยก็อาจจะทำให้ประสิทธิภาพในการเราต์ ชิ้นข้อมูลต่ำลงไปมาก และการสื่อสารข้อมูลก็จะติดขัดเป็นคอขวดที่เราเตอร์

4.2.2 เซอร์กิตเลเวลไฟร์วอล / สเตตฟูลอินสเป็กชั่นไฟร์วอล (Circuit-Level Firewall / Stateful Inspection Firewall)

การสื่อสารโดยทั่วไปจะเป็นการสื่อสารแบบต่อเนื่อง โต้ตอบกันไปมาระหว่างผู้รับและผู้ส่งอยู่เสมอ โปรโตคอลที่อยู่ในชั้นที่สูงกว่าอินเตอร์เน็ตเลเยอร์ ไม่ว่าจะเป็นทรานสปอร์ตอย่างเช่น ทีซีพี , ยูดีพี หรือเลยไปถึงแอพพลิเคชั่นเลเยอร์ เช่น เอฟทีพี, เอชทีทีพี, เอสเอ็มทีพี ล้วนแล้วแต่จะต้องมีสถานะของการสื่อสาร (State) เสมอ สถานะนี้จะทำให้ทั้งสองฝั่งสามารถสื่อสารกันได้อย่างต่อเนื่องคือทราบว่าตอนนี้กำลังอยู่ ณ จุดใดและจะต้องส่งหรือรับข้อมูลใดเป็นลำดับต่อไป

ความแตกต่างของการพิจารณาข้อมูลแบบแพ็กเก็ตฟิลเตอร์ริงกับสเตตฟูล (Stateful)

อันที่จริงสองเรื่องนี้มิได้ขัดแย้งกันแต่ประการใด แพ็กเก็ตนั้นเป็นการสื่อสารที่เป็นส่วนย่อยของการสื่อสารทั้งหมด ผลของการสื่อสารข้อมูลก็คือผลรวมของการสื่อสารข้อมูลหลายๆ แพ็กเก็ตนั่นเอง แต่อย่างไรก็ตามการฟิลเตอร์หรือกรอง โดยพิจารณาทีละแพ็กเก็ตของทุกแพ็กเก็ตที่ผ่านเข้าออกนั้นอาจจะมีผลลัพธ์แตกต่างจากการฟิลเตอร์ของในแบบที่สองสถานะและภาพรวมหรือที่เรียกว่าสเตตฟูล (Stateful) หากเปรียบเทียบการพิจารณาข้อมูลครั้งละแพ็กเก็ตกับการพิจารณาแบบสเตตฟูลแล้ว ตัวอย่างที่น่าจะช่วยให้เข้าใจได้ง่ายขึ้นคือ

เปรียบเทียบการสื่อสารข้อมูลทั้งหมดเสมือนภาพยนต์ แพ็กเก็ตก็จะหมายถึงภาพนิ่งแต่ละภาพที่นำมาต่อรวมกันแล้วเปิดดูอย่างรวดเร็ว ภาพนิ่งเหล่านั้นจะกลายเป็นภาพเคลื่อนไหว ดังนั้นการพิจารณาแพ็กเก็ตก็เป็นเสมือนการดูภาพนิ่งทีละภาพ แต่จะไม่สามารถเซ็นเซอร์เนื้อเรื่องซึ่งเป็นสิ่งที่เกิดขึ้นจากความสัมพันธ์ของภาพหลายๆ ภาพได้มีโอกาสเป็นไปได้ว่ากิจกรรมบางชนิดที่หากดูเป็นภาพนิ่งแล้วจะไม่รู้สึกว่าสิ่งใดที่ไม่เหมาะสม แต่หากนำภาพนิ่งมาดูอย่างต่อเนื่องเป็นภาพเคลื่อนไหวแล้วก็อาจจะเป็นสิ่งที่ไม่พึงปรารถนาที่จะให้ปรากฏบนภาพยนตร์ก็เป็นได้

เซอร์กิตเลเวลไฟร์วอลเป็นไฟร์วอลที่ทำงานโดยที่สามารถเข้าใจสถานะของการสื่อสารทั้งกระบวนการ เพราะถือว่าการสื่อสารข้อมูลจะสมบูรณ์ได้นั้นต้องมีทั้งการส่งแล้การรับอย่างสอดคล้องสัมพันธ์กันนั่นเอง หมายถึงหากไฟร์วอลจะสามารถควบคุมการสื่อสารได้จริงก็จะต้องสามารถเข้าใจกรำบวนการของการสื่อสารตั้งแต่ต้นจนจบ โดยทั่วไปเราจะเรียกไฟร์วอลแบบนี้ว่า “สเตตฟูลอินสเป็กชั่นไฟร์วอล” (หรือเรียกย่อๆ ว่าสเตตฟูลไฟร์วอล) เป็นไฟร์วอลที่ทำการควบคุมแทรฟฟิกโดยใช้หลักการของแพ็กเก็จฟิลเตอร์ริง และการกำหนดแอคเซสรูลเช่นเดียวกับสกรีนิงเราเตอร์แต่สเตตฟูลไฟร์วอลจะมีความสามารถในการวิเคราะห์และรับรู้ความต่อเนื่องของแพ็กเก็ตใน

โปรโตคอลในระดับที่สูงขึ้นไปมากกว่า ไม่ว่าจะเป็น ทีซีพี, เอฟทีพี, เอชทีทีพี หรือแม้กระทั่งโปรโตคอลในระดับแอพพลิเคชั่น ที่จะมีวิธีการกำหนดสเตตของตนเอง

สเตตฟูลไฟร์วอลเป็นเครื่องมือที่ถูกออกแบบมาเพื่อทำหน้าที่ในการควบคุมแทรฟฟิกโดยเฉพาะไม่ได้เป็นการดัดแปลงการทำงานมาจากเราเตอร์จึงมีความสามารถในการควบคุมแทรฟฟิกการกำหนดแอคเซสรูล การบริการ รวมไปถึงความยืดหยุ่นจองการควบคุมแทรฟฟิก และประสิทธิภาพในการทำงานที่สูงกว่าสกรีนิ่งเราเตอร์เป็นอย่างมาก โดยทั่วไปหากพูดถึงไฟร์วอลจะหมายถึงไฟร์วอลประเภทนี้เอง

ตามที่ได้กล่าวไว้ข้างต้นว่า ความแตกต่างที่สำคัญของไฟร์วอลทั้งสองชนิดนี้ในแง่ของการตรวจสอบแทรฟฟิกคือ สเตตฟูลไฟร์วอล มีความสามารถในการวิเคราะห์แทรฟฟิกที่ผ่านไปมาในโปรโตคอลที่เลเยอร์สูงขึ้นไปไม่ว่าจะเป็น ทีซีพี, ยูดีพี, ไอซีเอ็มพี ได้อย่างสมบูรณ์ต่างจาก สกรีนิงเราเตอร์ที่สามารถวิเคราะห์ได้เฉพาะเท่าที่จะมีข้อมูลใน 1 แพ็กเก็ตเท่านั้นเพราะบางครั้งแทรฟฟิกที่อ่านไปมานั้นเชื่อมโยงกันหลายแพ็กเก็ต โดยเฉพาะ ทีซีพี ซึ่งจะมีลำดับของการติดต่อวื่อวารที่สัมพันธ์กันในแต่ลำแพ็กเก็ต การพิจารณาแพ็กเก็ตใดแพ็กเก็ตหนึ่งโดยไม่พิจารณาแพ็กเก็ตอื่นที่เกี่ยวข้องก็อาจจะไม่สามารถควบคุมแทรฟฟิกของ ทีซีพี ได้นอกจากนี้ยังรวมไปถึงการที่สเตตฟูลไฟร์วอลมีความสามารุในการประกอบรวมแฟร็กเมนต์เข้าด้วยกันให้เป็นดาต้าแกรมที่สมบูรณ์ก่อนหลังจากนั้นจึงนำดาต้าแกรมนั้นมาทำการตรวจสอบเปรียบเทียบกับแอคเซสรูล

นอกจากการเชื่อมโยงกันของหลายแพ็กเก็ตสำหรับแพ็กเก็ตโปรโตคอล ทีซีพี ในทรานสปอร์ตเลเยอร์แล้ว ในแอพพลิเคชั่นเลเยอร์ก็มีแอพพลิเคชั่นบางชนิดที่จะต้องอาศัยการพิจารณาแทรฟฟิกอย่างต่อเนื่องเพื่อที่จะนำมากำหนดเป็นแอคเซสรูล ยกตัวอย่างเช่น การทำงานของเอฟทีพี ซึ่งในระหว่างการทำงานของแอพพลิเคชั่นนั้นโฮสที่เป็นไคลเอนต์จะสามารถกำหนดพอร์ตชั่วคราวขึ้นมาเป็นเซิร์ฟเวอร์พอร์ตใช้สำหรับรับ-ส่งไฟล์ได้ โดยพอร์ตเหล่านี้จะปิดลงเมื่อการรับ-ส่งข้อมูลเสร็จสิ้นสมบูรณ์ ซึ่งในกรณีนี้หากไม่มีการพิจารณาแทรฟฟิกที่มีมาก่อนหน้าแล้วไฟร์วอลอาจจะถือได้ว่าการเปิดให้บริการใหม่ขึ้นมาก็ได้ดังนั้นสเตตฟูลไฟร์วอลจึงมีการทำงานที่ค่อยข้างใกล้ชิดกับแอพพลิเคชั่นได้ค่อนข้างดี เพราะแอพพลิเคชั่นที่ใช้งานอยู่ในเน็ตเวิร์กไม่ได้มีเฉพาะแอพพลิเคชั่นพื้นฐานเท่านั้น มีแอพพลิเคชั่นอื่นๆ อีกมาก แต่หากแอพพลิเคชั่นใดมีการใช้งานอย่างแพร่หลาย และเป็นที่นิยมของผู้ใช้ โดยส่วนใหญ่ผู้ผลิตจะใส่บิวต์อิน การควบคุมแทรฟฟิกสำเร็จรูปมาให้อยู่ในไฟร์วอลเลย

ข้อดีของสเตตฟูลไฟร์วอล

1. ใช้งานง่ายเพราะถูกออกแบบมาทำหน้าที่ของไฟร์วอลโดยเฉพาะ ตรวจสอบแก้ไขรูลได้ง่าย ทำให้ผู้ใช้ไม่ต้องคอยกังวลถึงคำสั่ง และรูปแบบของคำสั่งถึงแม้ว่าจะต่างยี่ห้อกันก็สามารถเรียนรู้ใหม่ได้อย่างรวดเร็ว
2. ประสิทธิภาพในการทำงานสูง เนื่องจากออกแบบมาทำหน้าที่ไฟร์วอลโดยเฉพาะ สารมารถรองรับแอคเซสรูลที่ซับซ้อนได้ โดยที่ความสามารถในการทำงานไม่ลดลง
3. มีคุณสมบัติเพิ่มเติมให้ใช้ได้มากนอกเหนือจากการควบคุมแทรฟฟิก เช่นสามารถนำไปใช้ร่วมกับระบบการตรวจจับการบุกรุกหรือ IDS (Intrusion Detection System) เพื่อป้องกันการโจมตีได้อัตโนมัติ, สามารถบันทึกข้อมูลเอาไว้กลับมาดูภายหลังได้, สามารถใช้งานร่วมกับระบบป้องกันไวรัสได้เป็นต้น
4. การกำหนดแอคเซสรูลทำได้ง่ายเพราะไฟร์วอลมีความเข้าใจในโปรโตคอลระดับสูง ดังนั้นผู้ใช้อาจจะไม่จำเป็นต้องมีความเชี่ยวชาญในเรื่องระบบเครือข่ายมากนัก ก็พอจะใช้งานไฟร์วอลได้โดยกำหนดกฎพื้นฐานของแอพพลิเคชั่นที่ผู้ใช้รู้จัก มากกว่าการกำหนดกฎโดยใช้ข้อมูลบนแพ็กเก็ตโดยตรง เช่นแทนที่จะต้องกำหนดแอคเซสรูลให้อนุญาต ICMP Time exceed in Transit ให้ผ่านได้ เพื่อใช้คำสั่ง Teaceroute (ซึ่งโดยทั่วไปแล้วผู้ใช้ไม่ทราบว่าโปรแกรมใดใช้โปรโตคอลอะไร แต่จะรู้ว่าตนเองต้องการใช้โปรแกรมหรือแอพพลิเคชั่นอะไรบ้าง) ก็ระบุในไฟร์วอลว่าอนุญาตให้คำสั่ง Traceroute ทำงานได้หลังจากนั้นไฟร์วอลจึงกำหนดเป็นแอคเซสรูลที่ระบุโปรโตคอลนั่นเอง
5. สามารถเพิ่มเติมความปลอดภัยโดยระบบการตรวจสอบผู้ใช้ (Authenticate) ได้
6. การสื่อสารระหว่างไฟร์วอลกับแอดมินคอลโซล (Administration Console : เครื่องที่ทำหน้าที่ในการบริหารไฟร์วอล) จะมีความปลอดภัยสูง มีการตรวจสอบสิทธิ์ของผู้ที่เป็นแอดมินรวมทั้งการสื่อสารระหว่างไฟร์วอลกับคอลโซลจะมีการรักษาความปลอดภัยที่เข้มงวด มีการเข้ารหัสเพื่อนป้องกันการดักอ่านข้อมูล

ข้อเสียของสเตตฟูลไฟร์วอล

1. มีราคาแพง ถึงแม้ว่าปัจจุบันจะลดลงไปมากแล้วแต่ก็ยังแพงอยู่
2. ในกรณีที่เป็นไฟร์วอลแบบซอฟต์แวร์ที่ทำงานบนระบบปฏิบัติการทั่วไปต่างก็มีความเสี่ยงที่จะถูกเจาะได้เนื่องจากปัญหาของแต่ละระบบปฏิบัติการเอง ซึ่งจะสามารถเจาะได้ง่ายกว่าการเจาะเราเตอร์ เพราะรูรั่วของระบบปฏิบัติการมีมากกว่าของเราเตอร์

3. ในกรณีไฟร์วอลเป็นประเภทเน็ตเวิร์คแอพพลิแอนซ์ (Network Appliance) คือ ออกแบบทั้งซอฟต์แวร์และฮาร์ดแวร์เป็นเครื่องเดียวกันเพื่อทำหน้าที่เป็นไฟร์วอล โดยเฉพาะผู้ใช้จะเป็นต้องพึ่งพาผู้ผลิตค่อนข้างมาก หากมีปัญหาอาจจะไม่สามารถแก้ไขโดยการใช้อะไหล่ทดแทนจากที่อื่นได้

4.2.3 แอพพลิเคชั่นไฟร์วอล (พร็อกซี่) (Application Level Firewall (Proxy))

พร็อกซี่เป็นเครื่องมือในการควบคุมแทรฟฟิกชนิดหนึ่ง ซึ่งทำงานที่ระดับของแอพพลิเคชั่นในลักษณะที่เป็นตัวกลางในการสื่อสารระหว่างไคลเอนต์กับเซิร์ฟเวอร์ โดยทำหน้าที่ป้องกันไม่ให้มีการสื่อสารโดยตรงระหว่างไคลเอนต์กับเซอร์เวอร์ แต่ยังคงให้ไคลเอนต์สามารถใช้งานแอพพลิเคชั่นบนเซิร์ฟเวอร์ได้ตามปกติ และผู้ใช้ซึ่งใช้งานแอพพลิเคชั่นนั้นๆ จะไม่ได้รับผมกระทบแต่อย่างใด

ลักษณะการทำงานของพร็อกซี่

โดยทั่วไปแล้วการสื่อสารระหว่างไคลเอนต์กับเซิร์ฟเวอร์นั้นจะต้องมรการเชื่อมต่อ เกิดขึ้นระหว่างไคลเอนต์กับเซิร์ฟเวอร์ตลอดเวลาที่สื่อสารกันอยู่ จุกสำคัญอยู่ตรงที่การเชื่อมต่อโดยตรงนั้นจะมีความเสี่ยงหลายประการจึงมีการทำพร็อกซี่เข้ามาใช้งาน

หน้าที่ในการทำงานของพร็อกซี่ คือ เป็นตัวกลางรับข้อมูลจากไคลเอนต์มาแล้วทำงานส่งต่อไปยังเซิร์ฟเวอร์ และรับข้อมูลที่ตอบกลับจากเซิร์ฟเวอร์กลับมายังไคลเอนต์ที่ทำการร้องขอและจะทำหน้าที่นี่อยู่ตลอดเวลาที่ไคลเอนต์และเซิร์ฟเวอร์นั้นติดต่อกันซึ่งการที่มีพร็อกซี่มาเป็นตัวกลางระหว่างไคลเอนต์กับเซิร์ฟเวอร์นั้นทำให้โฮสต์ทั้งคู่ไม่จำเป็นต้องติดต่อกันโดยตรง เพียงแค่ติดต่อกับตัวกลางคือพร็อกซี่เท่านั้น และการทำงานของแอพพลิเคชั่นทั้งสองฝั่งยังคงทำงานได้เช่นเดิม

ขั้นตอนการนำพร็อกซี่เข้ามาใช้งาน

1. การเริ่มต้นการทำงานของแอพพลิเคชั่นโดยทั่วไป เริ่มจากการที่แอพพลิเคชั่นบนไคลเอนต์ขอรับข้อมูลจากเซิร์ฟเวอร์ตามโปรโตคอลในแอพพลิเคชั่นเลเยอร์ที่กำหนดไว้เช่น เว็บบราวเซอร์กับเว็บเซิร์ฟเวอร์ จะใช้โปรโตคอล เอชทีทีพี ในการสื่อสารระหว่างกัน
2. เมื่อเว็บเซิร์ฟเวอร์ได้รับการขอข้อมูลจากบราวเซอร์ก็จะทำการติดต่อสื่อสารกันและทั้งฝั่งไคลเอนต์และเซิร์ฟเวอร์ก็จะทำการติดต่อสื่อสารกันตามที่โปรโตคอล เอชทีทีพี

กำหนดจนจบการสื่อสารเซสชั่นนั้น อย่างไรก็ตามโปรโตคอล เอชทีทีพี นั้นจะต้องอาศัย ทีซีพี ในการรับส่งข้อมูลระหว่างไคลเอนต์กับเซิร์ฟเวอร์ นั่นหมายถึงไคลเอนต์จะต้องสามารถติดต่อกับเซิร์ฟเวอร์ได้ด้วย ทีซีพี เสียก่อนเนื่องจาก ทีซีพีเป็นโปรโตคอลเลเยอร์ที่อยู่ภายใต้ เอชทีทีพี อีกเลเยอร์หนึ่งดังนั้นสภาวะการทำงานปกติของเอชทีทีพี บราวเซอร์จะต้องสามารถติดต่อกับเซิร์ฟเวอร์โดยตรงเสมอนั่นคือในสภาวะการทำงานปกตินั่นแพ็กเก็ตของทีซีพี/ไอพี จะต้องสามารถส่งถึงกันระหว่างโฮสต์ทั้งคู่ได้

3. เมื่อนำพร็อกซี่มาใช้งานจะต้องติดตั้งตรงจุดที่คั่นระหว่างไคลเอนต์กับเซิร์ฟเวอร์เพื่อเป็นตัวกลาง โดยที่พร็อกซี่จะต้องมี 2 อินเตอร์เฟส โดยอินเตอร์เฟสหนึ่งต่ออยู่กับระบบเครือข่ายของเครื่องลูกข่ายและอีกอินเตอร์เฟสหนึ่งต่ออยู่กับเซิร์ฟเวอร์ ซึ่งหากพิจารณาที่พร็อกซี่แล้วจะเห็นว่าสามารถติดต่อได้โดยตรงกับทั้งไคลเอนต์และเซิร์ฟเวอร์แต่สำหรับไคลเอนต์และเซิร์ฟเวอร์จะติดต่อได้แต่เพียงกับพร็อกซี่เท่านั้น

ในลักษณะที่มีพร็อกซี่มาคั่นกลางระหว่างระบบเครือข่ายทั้งสอง การสื่อสารระหว่างไคลเอนต์และเซิร์ฟเวอร์ด้วยวิธีการเดิมโดยใช้ เอชทีทีพี เช่นเดิมเหมือนกับมีการสื่อสารกันโดยตรงนั้นย่อมไม่สามารถจะกระทำได้เพราะการสื่อสารในเลเยอร์ล่างของ ทีซีพี/ไอพี นั้นไม่สามารถทำได้สำเร็จดังนั้นจึงจำเป็นต้องมีการปรับปรุงแก้ไขโปรโตคอลให้สามารถรองรับการสื่อสารที่มีตัวกลางมาถ่ายทอดข้อมูลได้โดยให้ในระดับ ทีซีพี/ไอพี นั้นกำหนดให้เพียงโฮสต์แต่ละฝั่งสามารถติดต่อกับพร็อกซี่เท่านั้น ส่วนในระดับเอชทีทีพี นั้นพร็อกซี่จะทำการส่งต่อระหว่างทั้งสองฝั่งให้ดูประหนึ่งว่าสามารถติดต่อกันได้โดยตรงซึ่งจุดสำคัญของพร็อกซี่ก็จะอยู่ตรงนี้เอง อาจจะกล่าวโดยสรุปคือพร็อกซี่จะทำให้โฮสต์ติดต่อกันได้โดยโปรโตคอล ทีซีพี/ไอพี แต่จะสามารถติดต่อกันได้ด้วยโปรโตคอลในระดับแอพพลิเคชั่นเลเยอร์

อย่างที่กล่าวข้างต้นคือแอพพลิเคชั่นที่ใช้งานพร็อกซี่นั้นจะต้องมีการแก้ไขในระดับแอพพลิเคชั่นในบางส่วนเพื่อให้สามารถสื่อสารผ่านพร็อกซี่ได้ ดังเช่นเว็บบราวเซอร์ หากจะทำการสื่อสาร โดยผ่านพร็อกซี่นั้นจะต้องทำการปรับแต่งเพื่อให้บราวเซอร์ทราบว่าจะให้ติดต่อกับเว็บไซท์โดยผ่านพร็อกซี่หรือจะติดต่อไปยังเซิร์ฟเวอร์ใดก็เพียงแต่ส่งคำขอไปยังพร็อกซี่เท่านั้นหลังจากนั้นก็เป็นภาระของ พร็อกซี่ในการติดต่อกับเว็บเซิร์ฟเวอร์ตัวจริง แล้วจึงนำผลที่ได้จากเว็บเซิร์ฟเวอร์ตอบกลับมายังบราวเซอร์

เมื่อปรับแต่งให้บราวเซอร์ ทำการสื่อสารผ่านพร็อกซี่ การทำงานจะมีการเปลี่ยนแปลง ไปคือจากเดิมเมื่อเว็บบราวเซอร์ต้องการติดต่อกับเว็บเซิร์ฟเวอร์ก็จะส่งคำขอในระดับแอพพลิเคชั่น ซึ่งในกรณีนี้คือ เอชทีทีพี ไปยังเซิร์ฟเวอร์ปลายทางคือพร็อกซี่แค่นั้น ไม่ว่าเว็บเซิร์ฟเวอร์จะติดต่อไปยังเว็บเซิร์ฟเวอร์ซึ่งอยู่ที่ใดก็ตามแพ็กเก็ตจริงๆ ก็จะเดินทางไปแต่พร็อกซี่

เท่านั้น ในขณะเดียวกันพร็อกซี่ก็คอยโต้ตอบในระดับของเอชทีทีพี กลับไปยังไคลเอนต์ประหนึ่งว่าตนเองเป็นเว็บเซิร์ฟเวอร์ปลายทางจริง

ข้อดีของการใช้งานพร็อกซี่

1. สามารถควบคุมการติดต่อสื่อสารระหว่างอินเตอร์เน็ตกับเน็ตเวิร์กภายในให้อยู่ในระดับแอพพลิเคชั่นเท่านั้น ตัดขาดการติดต่อโดยตรงในระดับเน็ตเวิร์กเลเยอร์ระหว่างอินเตอร์เน็ตกับเน็ตเวิร์กภายในออกจากกันอย่างเด็ดขาด ทำให้ลดความเสี่ยงต่อการถูกควบคุมจากการสแกน การถูกจาะระบบ การก่อกวนโดยใช้เทคนิคในระดับเน็ตเวิร์กเลเยอร์ที่จะเข้ามายังเน็ตเวิร์กภายในได้อย่างเด็ดขาด
2. สามารถเพิ่มเติมหน้าที่การทำงานอย่างอื่นเข้าไปในพร็อกซี่ได้ เช่นสำหรับเว็บพร็อกซี่นอกจากจะเป็นตัวกลางในกาติดต่อแล้ว ยังสามารถควบคุมไม่ให้เว็บบราวเซอร์ติดต่อกับเว็บไซท์ที่ไม่ต้องการได้อีกด้วย โดยการกำหนดรายชื่อเว็บไซท์เหล่านั้นไว้ในพร็อกซี่
3. สามารถทำการแคชข้อมูลเก็บไว้ในตัวพร็อกซี่ สำหรับข้อมูลใดที่มีการเรียกใช้บ่อยๆ ก็ไม่จำเป็นต้องไปอ่านจากเซิร์ฟเวอร์ใหม่ทุกครั้ง แต่ส่วนนี้จะใช้กับข้อมูลที่เป็นสแตติกเท่านั้นข้อมูลที่มีการเปลี่ยนแปลงตลอดเวลาเป็นไดนามิคอาจจะไม่สามารถแคชไว้ได้
4. ทำให้ผู้ใช้บริการใช้แบนด์วิดธ์ร่วมกันอย่างมีประสิทธิภาพ โดยเฉพาะเมื่อใช้ร่วมกับการแคชที่มีอยู่ในพร็อกซี่ทำให้ช่วยประหยัดการใช้งานแบนด์วิดธ์ไปได้มาก
5. สามารถเพิ่มเติมส่วนการตรวจสอบผู้ใช้ (Authenticate) เข้าไปในหน้าที่หนึ่งของพร้อกซี่ได้โดยการอนุญาตให้สามารถใช้งานพร็อกซี่นั้นจะขึ้นอยู่กับสิทธิ์การใช้งานที่ผู้ใช้มีอยู่ทำให้สามารถควบคุมการใช่งานได้ใกล้ชิดมากขึ้นมากกว่าการควบคุมแทรฟฟิคโดยพิจารณาจาก ไอพีแอดเดรส ของโฮสต์เพียงอย่างเดียว
6. สามารถทำการกลั่นกรองเนื้อหาข้อมูลได้ (Content Filtering) ทำให้สามารถนำมาเป็นเงื่อนไขในการอนุญาตให้ข้อมูลเหล่านั้นผ่านเข่านออกได้ เช่นเว็บพร็อกซี่สามารถตรวจสอบเนื้อหาของเว็บไซท์ที่ผู้ใช้เข้าไปดูหากปรากฏว่ามีข้อความที่ไม่เหมาะสมพร็อกซี่ก็จะสามารถดรอปเซสชั่นที่ผู้ใช้ขอมาได้ หรือในกรณีที่เป็นอีเมล์ พร็อกซี่ก็จะสามารถตรวจสอบเนื้อหาให้อีเมล์ได้ว่ามีข้อความที่ไม่เหมาะสมหรือไม่ และอาจจะครอบคลุมถึงการตรวจสอบเนื้อหาที่แนบมากับจดหมายด้วย

ข้อเสียของการใช้งานพร็อกซี่

1. ขึ้นอยู่กับแอพพลิเคชั่น หากแอพพลิเคชั่นไม่รองรับการสื่อสารโดยผ่านพร็อกซี่ก็ไม่สามารถใช้งานได้
2. ไม่สามารถใช้งานกับแอพพลิเคชั่นที่ต้องการ การสื่อสารโดยตรงแบบ end-to-end ซึ่งแพ็กเก็ตจะต้องมากจากโฮสต์ปลายทางทั้งคู่เท่านั้นผ่านตัวกลางไม่ได้
3. เสี่ยงต่อการละเมิดความเป็นส่วนบุคคล (Privacy) เนื่องจากข้อมูลทั้งหมดที่สื่อสารจะต้องผ่านพร็อกซี่ก่อนเสมอ และพร็อกซี่ก็มีความสามารถที่จะเก็บข้อมูลเหล่านั้นไว้ตรวจสอบได้หากมีผู้นำข้อมูลเหล่านั้นไปวิเคราะห์จะสามารถทราบการใช้งานหรืออาจจะทราบข้อมูลทั้งหมดของผู้ใช้ได้
4. เนื่องจากลักษณะของแต่ละแอพพลิเคชั่นจะแตกต่างกันออกไป ดังนั้นพร็อกซี่แต่ละแอพพลิเคชั่นจึงทำหน้าที่เฉพาะแอพพลิเคชั่นนั้นๆ ไม่สามารถใช้ราวมกันหด้หากโฮสต์ที่อยู่หลังพร็อกซี่มีการใช่งานหลายแอพพลิเคชั่นก็จะต้องมีพร็อกซี่จำนวนมากเปิดให้บริการตามจำนวนแอพพลิเคชั่นนั้นๆ
5. ความสามารถในการประมวลผลของโฮสต์ที่ทำหน้าที่พร็อกซี่อาจจะเป็นคอขวดของระบบได้เพราะการสื่อสารทั้งหมดของไคลเอนต์และเซิร์ฟเวอร์จะถูกรวมศูนย์ พร็อกซี่ก่อนเสมอ แทนที่จะกระจายไปยังไคลเอนต์และเซิร์ฟเวอร์ ปัญหาลักษณะนี้จะสามารถพบได้ชัดเมื่อไคลเอนต์จำนวนมาก
6. เนื่องจากพร็อกซี่เป็นแอพพลิเคชั่นชนิดหนึ่งเช่นกัน การติดต่อกันในระบบจะอาศัยระบบปฏิบัติการเป็นหลักจึงมีความสามารถในการป้องกันตัวเองต่ำกว่าไฟร์วอลทั่วไป ตัวพร็อกซี่เองจึงมีความเสี่ยงต่อการถูกโจมตีได้มากและเปราะบางต่อการโจมตีให้ปิดบริการด้วยเทคนิคในระดับเน็ตเวิร์ก ซึ่งอาจส่งผลให้พร็อกซี่หยุดทำงานได้โดยง่าย โดยเฉพาะเมื่อพร็อกซี่นั้นเป็นโฮสต์ที่ต่อโดยตรงกับอินเตอร์เน็ตจึงเป็นเสมือนด่านหน้าของระบบเครือข่ายที่จะต้องถูกสแกน ถูกเจาะอย่างแน่นอน แต่ในระดับความต้านทานของพร็อกซี่นั้นต่ำกว่าไฟร์วอลทั่วไป จึงมีแนวโน้มหากว่าใช้พร็อกซี่โดยปราศจากไฟร์วอลร่วมด้วยโอกาสที่พร็อกซี่จะโดนเจาะนั้นมีอยู่สูงมาก

บทที่ 5
ระบบตรวจจับผู้บุกรุก

5.1 ระบบตรวจจับผู้บุกรุก (Intrusion Detection System)

ระบบตรวจจับผู้บุกรุกระบบคอมพิวเตอร์เป็นกระบวนการตรวจสอบเครือข่ายและระบบคอมพิวเตอร์เมื่อมีการละเมิดกฎการรักษาความปลอดภัย โดยระบบตรวจจับผู้บุกรุกประกอบด้วยหน้าที่หลัก 3 ประการ คือ รวบรวมข้อมูลเหตุการณ์ที่เกิดขึ้นไว้ในเรคอร์ด มีตัววิเคราะห์ในการตรวจจับผู้บุกรุก และส่วนตอบโต้ผู้บุกรุก

คำว่า ผู้บุกรุก หมายถึง ผู้ที่ชอบหาช่องโหว่ของโปรแกรมและเจาะเข้าไปในระบบคอมพิวเตอร์รวมทั้งหมายถึง แฮกเกอร์ คือ ผู้ที่ชอบเข้าไปศึกษาบางสิ่งบางอย่างในระบบ

5.1.1 ความจำเป็นที่ต้องมีระบบตรวจจับผู้บุกรุกระบบคอมพิวเตอร์

เป็นที่รู้กันดีว่าระบบคอมพิวเตอร์และเครือข่ายมีการออกแบบที่ไม่ค่อยปลอดภัยมากนักทำให้ผู้บุกรุกมีโอกาสที่จะบุกรุกระบบได้ง่ายแม้ว่ามีไฟร์วอล อยู่แล้วก็ตามก็ไม่สามารถป้องกันการบุกรุก ได้อย่าง 100 เปอร์เซ็นต์ เนื่องจากไฟร์วอลเป็นการป้องกันระหว่างระบบคอมพิวเตอร์ภายในเครือข่าย (Internal Network) กับระบบภายนอก (Internet) หากผู้บุกรุกเป็นคนภายในไฟร์วอลเอง ก็ไม่สามารถป้องกันได้ ดังนั้นการมีไฟร์วอลเปรียบเสมือนการมีรั้วกั้นรอบบ้าน และระบบตรวจจับผู้บุกรุกระบบคอมพิวเตอร์เปรียบเสมือนการติดตั้งกล้องวีดีโอคอยตรวจสอบสิ่งผิดปกติทำให้การป้องกันระบบทำได้ดียิ่งขึ้น

5.2 ข้อดีและข้อเสียของระบบตรวจจับผู้บุกรุก

5.2.1 ข้อดีของระบบตรวจจับผู้บุกรุก

1. การตอบสนองทันทีทันใด

จริงๆ แล้วการวิเคราะห์การบุกรุกนั้น หากเป็นผู้เชี่ยวชาญที่มีความรู้ความเข้าใจด้านระบบเครือข่ายและโปรโตคอลเป็นอย่างดีก็จะวิเคราะห์ได้โดยอาศัยเครื่องมือเพียงเล็กน้อยเท่านั้น คือ ใช้เครื่องมือทำการจัดเก็บบันทึกข้อมูลทั้งหมดที่มีการสื่อสารกันบนเน็ตเวิร์ค แล้วนำข้อมูลที่ได้เหล่านั้นมาวิเคราะห์โดยพฤติกรรมและความสัมพันธ์ ก็จะสามารถหาสิ่งผิดปกติที่เกิดขึ้นได้ แต่การวิเคราะห์ในลักษณะดังกล่าวจะกระทำได้ก็ต่อเมื่อได้เกิดเหตุการณ์ไปแล้ว เนื่องจากการวิเคราะห์จะเป็นไปในลักษณะการวิเคราะห์ข้อมูลย้อนหลัง ไม่สามารถจะกระทำได้ในทันที ซึ่ง IDS จะช่วยแก้ไขข้อบกพร่องในส่วนนี้ เพราะ IDS สามารถตรวจจับได้ทันทีที่มีความผิดปกติเกิดขึ้น และช่วยให้

ทำการแก้ไขได้ทันท่วงที การทำงานพื้นฐานของ IDS จะเหมือนกันบการท่ทำโดยคน เพียงแต่ IDS นั้นทำงานโดยอัตโนมัติ และการทำงานอยู่ตลอดเวลาไม่มีหยุดจึ่งสามารถตอบสนองต่อสิ่งผิดปกติได้รวดเร็วกว่า ซึ่งถ้าให้คนมานั่งตรวจจับก็ไม่สามารถทำได้ตลอดเวลา

2. การมีฐานความรู้ของการวิเคราะห์

จากที่ได้กล่าวมาข้างต้น การที่จะตรวจจับสิ่งผิดปกติและแยกแยะกิจกรรมเหล่านั้นออกจากการสื่อสารข้อมูลตามปกติได้นั้นจะต้องอาศัยความชำนาญและเข้าใจใจรูปแบบของการสื่อสารข้อมูลและการบุกรุกเป็นอย่างดี นั้นคือทักษะที่จำเป็นของนักวิเคราะห์การบุกรุก (Intrusion Analyst) ซึ่งผู้เชี่ยวชาญในระดับที่จะทำงานเช่นนี้ได้มีไม่มากนัก ประกอบกับเทคนิคและกลวิธีในการบุกรุกหรือก่อกวนนั้นได้พัฒนาขึ้นทุกวัน วิธีการตรวจจับและวิเคราะห์จำเป็นต้องพัฒนาตามให้สอดคล้องกันจึงจะตรวจจับได้อย่างมีประสิทธิผล ซึ่งในส่วนนี้ผู้เชี่ยวชาญเองก็อาจจะทำได้ไม่ดีเท่า

IDS สามารถช่วยแบ่งเบาภาระของนักวิเคราะห์ลงได้มาก โดยหากรู้รูปแบบพฤติกรรมแน่ชัดว่าเป็นการมุ่งร้ายก็ให้จัดเก็บข้อมูลรูปแบบเหล่านี้ใน IDS เสีย เมื่อมีกิจกรรมดังกล่าวเกิดขึ้นในระบบเครือข่าย IDS ก็สามารถตรวจพบได้ทันที และเมื่อค้นพบรูปแบบใหม่ก็จัดเก็บลงใน IDS อีกทำให้ IDS เสมือนมีฐานความรู้ในการวิเคราะห์การบุกรุกได้ดีในระดับหนึ่ง และขีดความสามารถก็จะเพิ่มขึ้นเรื่อยๆ ตามปริมาณของรูปแบบที่เก็บอยู่ในฐานความรู้นั่นเอง หากมีการบำรุงรักษาฐานความรู้ใน IDS ได้ดีและนำ IDS ไปใช้ในส่วนที่เหมาะสมแล้ว การบุกรุกที่ไม่ใช่เทคนิคใหม่ล่าสุดจริงๆ ก็แทบจะไม่สามารถเล็ดรอดสายตา IDS ไปได้ถึงขั้นนี้แล้วแม่ว่าจะเป็น IDS แบบธรรมดาๆ ก็มีความสามารถมกกว่าผู้บริหารระบบทั่วไปเสียอีก

สำหรับนักวิเคราะห์แล้วเมื่อมี IDS จะทำให้ไม่ต้องห่วงหน้าพะวงหลัง เพราะการบุกรุกที่สามารถตรวจจับได้ง่ายๆ ก็สามารถตรวจพบได้โดย IDS อย่างน้อย IDS ก็ช่วยกลั่นกรองข้อมูลเบื้องต้นได้ในระดับหนึ่งและแบ่งเบาภาระได้พอสมควร

3. การช่วยตรวจสอบข้อบกพร่องของระบบป้องกันอื่นๆ

ระบบเครือข่ายของผู้ใช้อาจจะมีการป้องกันการบุกรุกอยู่แล้วโดยใช้ไฟร์วอล อย่างไรก็ตามไฟร์วอลมิใช่เครื่องมือที่จะป้องกันการบุกรุกได้โดยอัตโนมัติ จะต้องอาศัยผู้ที่บริหารระบบกำหนดกฎให้เหมาะสมกับการใช้งาน อีกประการหนึ่ง ถึงแม้จะมีการตั้งกฎที่เหมาะสมแล้วก็ตาม แต่กฎเหล่านั้นอาจไม่สามารถป้องกันการบุกรุกได้ การบริหารไฟร์วอลที่ดีก็ควรจะมีการตรวจสอบย้อนหลัง (Audit) และการทดสอบการเจาะระบบ (Penetration Test) เพื่อเป็นการตรวจทานระบบอีกครั้งหนึ่ง

IDS สามารถช่วยได้มากโดยติดตั้ง IDS ไว้หลังไฟร์วอล และทำการทดสอบเจาะระบบด้วยวิธีต่างๆ เพื่อดูว่าจะมีสีเทคนิคใดที่สามารเจาะผ่านไฟร์วอลได้บ้าง และหากมีแพ็กเก็ตใดผ่านเข้าไปได้ IDS ก็จะตรวจพบ ทำให้ผู้บริหารระบบสามารถปรับปรุงกฎให้รัดกุมมากขึ้น

5.2.2 ข้อเสียของระบบตรวจจับผู้บุกรุก

ถึงแม้ IDS จะมีประโยชน์ค่อนข้างมากในการช่วยรักษาความปลอดภัยและการเตือนภัยล่วงหน้าแต่ก็มีข้อเสียอยู่หลายประการซึ่งผู้ที่จะนำไปใช้จะต้องตระหนักไว้

1. การละเมิดความเป็นส่วนบุคคล

เนื่องจากว่า IDS มีพื้นฐานจากการนำข้อมูลทั้งหมดที่สื่อสารกันมาทำการวิเคราะห์ซึ่งข้อมูลเหล่านั้นจะต้องครอบคลุมถึงข้อมูลทั่วไปที่มีการสื่อการกันตามปกติ และการที่จะทราบว่ามีความผิดปกติหรือไม่นั้นก็จะต้องอ่านข้อมูลทั้งหมดด้วย การแชทคุยกัน ไอซีคิว, อีเมล์ และกิจกรรมอื่นๆ ที่สื่อสารข้อมูลผ่านระบบเครือข่าย ก็จะสามารถถูกเปิดอ่านได้จาก IDS นั่นหมายความว่า IDS สามารถนำไปใช้ทางที่ผิดเพื่อละเมิดสิทธิส่วนบุคคลได้ การทำงานของ IDS เปรียบเสมือนการที่ตำรวจต้องการตรวจสอบและดักจับผู้ไม่หวังดีที่คอยโทรศัพท์ก่อกวนชาวบ้านในหมู่บ้าน และเพื่อการนี้ตำรวจเองจึงต้องดักฟังโทรศัพท์ของทุกคนที่อยู่ในหมู่บ้านนั้น ซึ่งอาจจะมีเพียงหนึ่งในพันที่เป็นผู้ร้าย แต่ตำรวจผู้ทำหน้าที่ดักฟังก็จะรู้ความลับของคนทุกคน บางทีการที่มีคนดักฟังความลับของทุกคนอาจเป็นอันตรายกว่าการโดนผู้ร้ายก่อกวนก็เป็นได้

ดังนั้นการนำ IDS มาติดตั้งในระบบเครือข่ายจะต้องได้รับการอนุมัติจากหน่วยงานอย่างถูกต้องแล้วเท่านั้น และผู้ทำหน้าที่ในด้านนี้จะต้องเป็นผู้ที่ได้รับความไว้วางใจและมีความรับผิดชอบสูงในอันที่จะไม่ละเมิดสิทธิส่วนบุคคลของผู้อื่น และหากเห็นข้อมูลใดๆ ก็จะต้องไม่เปิดเผยข้อมูลเหล่านั้นแก่บุคคลอื่น โดยทั่วไปแล้วการติดตั้งอุปกรณ์ที่สามารถอ่านข้อมูลของผู้อื่นบนระบบเครือข่ายได้นั้นจะเป็นข้อห้ามอันดับต้นๆ ในนโยบายการรักษาความปลอดภัยเลยทีเดียว สิ่งที่เป็นข้อสังเกต คือ การกระทำในลักษณะนี้ยากต่อการป้องกันในทางวิธีการ ดังนั้นหน่วยงานโดยทั่วไปจึงต้องกำหนดเป็นข้อห้ามในนโยบายความปลอดภัย และมีบทลงโทษสำหรับผู้ที่ละเมิดในขั้นรุนแรง

2. การตอบโต้อัตโนมัติ

IDS ที่มีจำหน่ายอยู่ในท้องตลาดจะมีส่วนหนึ่งที่ให้ผู้ใช้สามารถกำหนดการดำเนินการอย่างหนึ่งอย่างใดเมื่อตรวจพบการบุกรุกเกิดขึ้น เช่น ส่งจดหมายเตือนผู้ดูแลระบบ เรียกวิทยุติดตามตัว ส่งคำสั่งไปยังไฟร์วอลเพื่อจำกัดการเข้าออกของข้อมูล และสิ่งที่สำคัญที่สุดซึ่งอาจจะส่งผลเสียหายใหญ่หลวงต่อเจ้าของได้คือการโจมตีกลับไปยังต้นกำเนิดของการบุกรุก (counter

attack) โดยที่ IDS เองก็จะรู้จักวิธีการโจมตีแบบต่างๆ ดีอยู่แล้ว จึงไม่ใช่เรื่องยากเย็นแต่อย่างใดที่จะทำการโจมตีผู้อื่น ผู้ผลิตจึงมักเพิ่มเติมส่วนนี้ให้แก่ IDS เสมือนหนึ่งการติดอาวุธไว้ให้ต่อกรกับแฮคเกอร์ เลยทีเดียว

ผู้ดูแลระบบบางส่วนอาจจะรู้สึกสะใจและคิดว่าเหมาะสมแล้วกับการโจมตีกลับไปยังผู้บุกรุกเหล่านั้นให้หลาบจำ จะได้ไม่พยายามข้องแวะอีก เป็นนโยบายการรักษาความปลอดภัยแบบตาต่อตาฟันต่อฟัน และเชื่อว่าหากกำหนดให้การโจมตีกลับเป็นไปอย่างอัตโนมัติแล้วน่าจะทำให้ปลอดภัยมากขึ้น ในค่ำคืนที่เงียบสงบใครจะรู้ว่า IDS อาจจะกำลังต่อกรอยู่กับผู้บุกรุก ที่พยายามแอบเข้ามาในระบบอย่างสุดกำลัง และสู้ยิบตาเพื่อรักษามิให้ผู้บุกรุก นั้นบุกรุกเข้ามาในระบบเครือข่ายได้ในโลกแก่งความเป็นจริงแล้วการตัดสินใจว่าผู้ใดเป็นผู้บุกรุก อย่างชัดเจนมิได้ทำได้โดยง่ายและในเวลาอันรวดเร็ว การที่กำหนดให้ IDS ทำการตอบโต้กลับไปในทันทีโดยมีข้อมูลเพียงแค่ผิวเผินนั้น นิดจากจะไม่ช่วยให้ระบบเครือข่ายของตนเองปลอดภัยแล้ว ยังจะทำให้เรากลายเป็นผู้บุกรุกไปด้วย เนื่อกจากคอยโจมตีผู้อื่นเสียเอง ยกตัวอย่างความเสียหายเช่น

- การวิเคราะห์ผิดพลาดเข้าใจว่ากิจกรรมที่เกิดขึ้นเป็นการบุกรุกและ IDS ก็ดำเนินการโจมตีกลับไปในทันที กรณีนี้ผู้บริสุทธิ์ก็จะถูกโจมตีจาก IDS ของเราโดยไม่รู้เรื่องใดๆ
- การวิเคราะห์ถูกต้องแต่หมายเลขประจำเครื่องต้นทางเป็นหมายเลขปลอม กรณีนี้หาก IDS ไม่มีกลไกในการตรวจสอบหมายเลขประจำเครื่องที่มีประสิทธิภาพอาจจะไม่สามารถแยกแยะได้ว่าต้นทางของการโจมตีที่แท้จริงแล้วเป็นที่ไหน และเมื่อทำการโจมตีกลับไปก็อาจจะมิใช่ตัวการที่แท้จริง และเหตุการณ์จะเลวร้ายยิ่งขึ้นหากหมายเลขเครื่องที่ปลอมมานั้นเป็นของหน่วยงานทางความมั่นคง หรือหน่วยงานทางการทหาร แต่ที่เลวร้ายที่สุดคือ หน่วยงานของเราเอง และเมื่อนั้นผู้ดูแลระบบอาจจะตระหนักได้ว่า IDS ตัวเดียวอาจจะทำให้เขาต้องเข้าไปนอนในคุกหลายคืน เทคนิคการปลอมหมายเลขเครื่องเช่นนี้อาจจะเป็นการยืมมือ IDS ของเราไปโจมตีผู้อื่นอีกทอดหนึ่งได้เป็นอย่างดี
- การวิเคราะห์หมายเลขเครื่องที่ถูกต้อง และการโจมตีกลับไปก็ตรงไปยังผู้บุกรุกอย่างถูกต้องตามที่ต้องการ แต่ผลที่ได้ก็เพียงอาจจะทำให้ผู้บุกรุก หยุดความพยายามไปชั่วขณะเท่านั้น อีกไม่นานก็จะหาวิธีกลับมาใหม่ และไม่เกิดผลใดๆ เลยนอกจากจะเป็นการยั่วยุให้มีความรุนแรงมากขึ้นเท่านั้นเอง

สิ่งสำคัญที่ผู้ทำหน้าที่ด้านความปลอดภัยและผู้บริหารระบบควรจะตระหนักไว้ให้จงหนักคือ ท่านไม่มีสิทธิพิเศษที่จะไปตอบโต้ผู้บุกรุกโดยการโจมตีกลับไม่ว่ากรณีใดๆ สิ่งที่ท่านจะทำได้ดีที่สุด คือ ทำให้ระบบแข็งแรงมั่นคงและปลอดภัยที่สุดเท่านั้น นั่นคือ ปิดประตูบ้านให้แน่นหนา

ตรวจตราอย่างรัดกุมและใช้งานเท่าที่จำเป็น ส่วนผู้ที่กระทำผิดเหล่านั้นควรจะปล่อยให้เป็นไปตามกฎหมายและกระบวนการยุติธรรมจะดีที่สุด เพราะว่าการโต้ตอบการกระทำที่ผิดกฎหมายจะทำให้เรากลายเป็นจำเลยไปด้วยในที่สุด

3. การเตือนภัยที่ผิดพลาด

ข้อมูลอาจไม่ใช่ข้อเสียที่สำคัญของการใช้ IDS หากผู้ใช้มีความรู้ในการใช้งานที่ดีพอและเข้าใจหลักการวิเคราะห์การบุกรุกของ IDS ได้ดี อย่างที่ได้กล่าวมาแล้วข้างต้น ก็คืออาจจะมีกิจกรรมปกติหลายอย่างที่มีลักษณะใกล้เคียงหรือบางครั้งเหมือนกับการพยายามบุกรุก ซึ่งแน่นอนว่าหาก IDS ได้ถูกกำหนดให้ตรวจจับกิจกรรมประเภทดังกล่าวแล้วก็จะมีการเตือนในทันทีที่ตรวจพบและเป็นหน้าที่ของนักวิเคราะห์ที่จะทำการสืบค้นข้อมูลด้านอื่นๆ มาประกอบการวินิจฉัยอีกครั้งหนึ่งว่าพฤติกรรมดังกล่าวที่ตรวจพบนั้นเป็นการบุกรุกหรือไม่ อย่างไร IDS ที่ถูกกำหนดให้มีความไวเป็นพิเศษมักจะสามารถตรวจจับพฤติกรรมที่ก้ำกึ่งนั้นได้มากเป็นพิเศษ ตัวอย่างเช่น IDS ถูกกำหนดไว้ว่า เมื่อได้รับ Ping Packet จากหมายเลขประจำเครื่องเดิมติดต่อกัน 10 แพ็กเก็ต ถายใจเวลา 30 วินาที ให้เตือนว่าเป็นการพยายามโจมตีโดยเทคนิค Ping Flood เป็นต้น หากระบบเครือข่ายดังกล่าวเป็นระบบเครือข่ายโดยวิสวกรระบบ และมีการทดสอบการ Ping บ่อยๆ ก็อาจจะทำให้ IDS เตือนอยู่แทบตลอดเวลาโดยไม่ได้มีการบุกรุกที่แท้จริง

การเตือนโดยมิได้มีการบุกรุกจริงนั้น อาจจะดูเหมือนว่าไม่ส่งผลเสียหายประการใดและน่าจะเกิดประโยชน์เสียด้วยซ้ำ เพราะจะทำให้ผู้ดูแลระบบมีความตื่นตัวตลอดเวลา แต่ในความเป็นจริงแล้วธรรมชาติของมนุษย์มีแนวโน้มที่จะละเลยต่อสิ่งเหล่านี้หากมีการเตือนแล้วไม่มีการบุกรุกจริงบ่อยครั้งเข้าความน่าเชื่อถือของ IDS ก็จะลดลงตามลำดับ และเมื่อมีความพยายามที่จะบุกรุกจริงก็จะไม่ได้ให้ความสนใจเท่าที่ควรและไม่ได้หาทางป้องกันอย่างเหมาะสม นั่นคือ IDS จะกลายเป็นเด็กเลี้ยงแกะที่เวลาหมาป่าเข้ามาจริงๆ ก็ไม่มีผู้ใดรับฟัง ดูเผินๆ อาจจะเหมือนกันว่ายังดีกว่าการไม่มี IDS เสียเลย แต่การมี IDS อยู่ในระบบโดยไม่ได้นำมาปรับแต่งอย่างเหมาะสม และเชื่อมั่นว่า IDS สามารถจะคอยระแวดระวังและเก็บหลักฐาน ต่างๆ ไว้ให้นั้นจะทำให้ผู้บริหารระบบนิ่งนอนใจและคลายความเคร่งครัดในการปฏิบัติงานลง อาจจะถึงขั้นหย่อนยานกว่าการป้องกันในระดับปกติที่ไม่มี IDS ได้ นอกจากนี้การปล่อยให้ IDS มีการเตือนอย่างไม่เหมาะสมจะทำให้เกิดข้อมูลในลักษณะที่เป็นการบุกรุกจริงและการเตือนผิดพลาดผสมกันอยู่ อาจทำให้การเตือนที่เป็นของจริงถูกกลบไปและยากต่อการสังเกต อย่าลืมว่าผู้บุกรุก ทีมีความสามารถจะทิ้งร่องรอยของการบุกรุกไว้เพียงเล็กน้อยอาจจะมีเพียง 2-3 รอยเท่านั้นที่ IDS สามารถตรวจพบได้ หากร่องรอยเหล่านี้ถูกนำไปผสมปนเปกับการตรวจจับอื่นๆ อีกนับพัน ย่อมมีโอกาสสูงที่จะถูกมองเลยไปโดยไม่มีผู้ใดให้ความสนใจ

5.3 พฤติกรรมโดยทั่วไปของผู้บุกรุก

โดยประกอบด้วยขั้นตอนสำคัญ 3 ขั้นซึ่งจำเป็นและเป็นพื้นฐานทั่วไปของการบุกรุก

5.3.1 การแกะรอย (Footprinting)

เป็นการรวบรวมข้อมูลของเครื่องเป้าหมายที่ต้องการให้ได้มากที่สุดโดยเฉพาะช่องโหว่ที่มีอยู่บนระบบเครือข่าย ทำให้ทราบโพรไฟร์ (Profile) ของเครื่องเป้าหมายที่เชื่อมต่ออยู่กับอินเตอร์เน็ต (Internet) ทั้งในส่วนของระบบเครือข่ายภายในหรืออินทราเน็ต (Intranet) และ เอ็กซ์ทราเน็ต (Extranet) รวมทั้งการให้บริการการเชื่อมต่อจากระยะไกล (remote access) ประกอบด้วย 3 ขั้นตอนย่อยดังนี้

1. กำหนดขอบเขตของการแกะรอย เป็นการพิจารณาว่าต้องการแกะรอยระบบเครือข่ายทั้งองค์กร หรือสนใจเฉพาะบางส่วนโดยค้นหาข้อมูลจากแหล่งข้อมูลที่เปิดเผยได้ (Open Source Search) ทำให้ทราบข้อมูลบางอย่างที่น่าสนใจได้ เช่น นโยบายด้านการรักษาความปลอดภัยซึ่งบ่งบอกให้ทราบถึงกลไกการรักษาความปลอดภัยที่ใช้งานอยู่ในปัจจุบัน ชื่อผู้ติดต่อและอี-เมล์แอดเดรส หรือตำแหน่งที่ตั้ง
2. การรวบรวมรายละเอียดต่างๆ ของระบบเครือข่ายเป้าหมาย โดยหาชื่อโดเมนและระบบเครือข่ายที่เกี่ยวโยงกับเครื่องเป้าหมาย แล้วเรียกใช้โปรแกรมเช่น whois (เป็นโปรแกรมที่ใช้หาข้อมูลว่ามีใครใช้งานอยู่ในระบบบ้าง) ดูโดเมนเนมของระบบ และค้นหาค่ากำหนดของเครื่องที่ทำงานอยู่ระบบเครือข่ายเป้าหมาย เช่น ชื่อเครื่อง รุ่นของระบบปฏิบัติการ ชื่อผู้ใช้ทั้งหมดของระบบ
3. การสำรวจระบบเครือข่าย โดยพยายามสำรวจเส้นทางที่แพ็กเก็ตไอพี เริ่มส่งจากเครื่องต้นทางไปถึงเครื่องปลายทาง

5.3.2 การสแกนเพื่อตรวจสอบ

เปรียบเสมือนการเคาะกำแพงเพื่อสำรวจหาประตูบ้านและหน้าต่าง (ช่องโหว่) โดยการแกะรอยจะทำให้ได้หมายเลขประจำเครื่อง และข้อมูลเกี่ยวกับระบบเครือข่ายของเครื่องเป้าหมายผ่านทางการสอบถามจากฐานข้อมูล whois ส่วนต่อมาคือ ทำการตรวจสอบว่าเครื่องคอมพิวเตอร์ปลายทางใดบ้างที่เปิดอยู่และสามารถเข้าถึงได้โดยจตรงผ่านทางอินเตอร์เน็ตและถ้าเป็นไปได้ควรทราบด้วยว่า มีหมายเลขพอร์ตใดเปิดอยู่บ้าง โดยการใช้เครื่องมือและเทคนิคต่างๆ เช่น ping sweeps, port scans และ automated discovery tools

เทคโนโลยี	ข้อมูลสำคัญเกี่ยวกับเทคโนโลยีนั้นๆ
อินเทอร์เน็ต (Internet)	เน็ตเวิร์คเซอร์วิสที่ทำงานบนโพโตคอล TCP และ UDP ประเภทของโพรเซสเซอร์ (เช่น SPARC, X86)
	Access Control List (ACL) เป็นกลไกควบคุมการเข้าถึงทรัพยากรต่างๆ บนเครื่องระบบป้องกันผู้บุกรุก (Intrusion Detection Systems, IDSes)
	การระบุรายละเอียดเกี่ยวกับระบบในแง่ต่างๆ (บัญชีรายชื่อผู้ใช้และกลุ่มต่างๆ , ตารางเราท์ติงของเราเตอร์, ข้อมูลของ SNMP)
อินทราเน็ต (Intranet)	เน็ตเวิร์คโพรโตคอลที่ใช้งานอยู่ภายใน (เช่น โพรโตคอล IP, IPX, และอื่นๆ)
	ชื่อโดเมนภายใน
	เน็ตเวิร์คบล็อค
	ไอพีแอดเดราของเครื่องคอมพิวเตอร์ที่เชื่อมต่อโดยตรงกับอินเทอร์เน็ต
	เน็ตเวิร์คเซอร์วิสที่ทำงานบนโพรโตคอล TCP และ UDP ประเภทของโพรเซสเซอร์ (เช่น SPARC, X86)
	Access Control List (ACL) เป็นกลไกควบคุมการเข้าถึงทรัพยากรต่างๆ บนเครื่อง
	ระบบป้องกันผู้บุกรุก (Intrusion Detection Systems, IDSes)
	การระบุรายละเอียดเกี่ยวกับระบบในแง่ต่างๆ (บัญชีรายชื่อผู้ใช้และกลุ่มต่างๆ , ตารางเราท์ติงของเราเตอร์, ข้อมูล SNMP)
การเชื่อมต่อจากระยะไกล (Remote Access)	เบอร์โทรศัพท์ในระบบอะนาล็อก/ดิจิตอล
	ประเภทของเครื่องให้บริการ
	กลไกการตรวจสอบผู้ใช้ (Authentication Mechanism)
เอ็กทราเน็ต (Extranet)	คอนเน็กชันต้นทางและปลายทาง
	ประเภทของคอนเน็กชัน
	กลไกควบคุมการเข้าถึงทรัพยากร (Access Control Mechanism)

ตารางที่ 5.1 เทคโนโลยีและข้อมูลสำคัญที่ผู้บุกรุกต้องการค้นหา

5.3.3 การค้นหาและรวบรวมรายละเอียด (Enumeration)

เป็นการค้นหาบัญชีผู้ใช้หรือค้นหาทรัพยากรที่แชร์ไว้ ซึ่งข้อแตกต่างสำคัญระหว่างเทคนิคการรวบรวมข้อมูลและเบาะแสกับการค้นหาและรวบรวมรายละเอียดต่างๆ คือ ระดับหรือความร้ายแรงของการบุกรุก หมายความว่า ทราบชื่อ แอคเคานต์ของผู้ใช้ที่ถูกต้องหรือทราบชื่อ แชร์โฟลเดอร์ ผู้บุกรุกจะพยายามทำการคาดเดารหัสผ่านหรือค้นหาจุดบกพร่องของสิทธิที่ได้ตั้งไว้ที่แชร์โฟลเดอร์นั้นๆ เมื่อผู้บุกรุกสามารถเข้าไปเป็นผู้ใช้ทั่วไปแล้วก็สามารถหาช่องทางขยายสิทธิ์ให้ได้เป็นผู้ดูแลระบบ

ประเภทของข้อมูลที่ผู้บุกรุก ต้องการรวบรวม สามารถบางออกเป็น 3 กลุ่มใหญ่คือ

- รายชื่อทรัพยากรนระบบเครือข่าย เช่น ชื่อแชร์โฟลเดอร์ เป็นต้น
- รายชื่อแอคเคานต์ของผู้ใช้และรายชื่อกลุ่ม
- รายชื่อแอพพลิเคชั่น

5.4 รูปแบบของระบบตรวจจับผู้บุกรุกระบบคอมพิวเตอร์ มีวิธีหลักๆ ในการป้องกันผู้บุกรุก 2 วิธีคือ

5.4.1 วิธีเปรียบเทียบพฤติกรรมผู้ใช้กับรูปแบบการบุกรุกที่รู้จัก (Misuse Intrusion Detection)

วิธีเปรียบเทียบพฤติกรรมผู้ใช้กับรูปแบบการบุกรุกที่รู้จักประกอบด้วยการเก็บบันทึกและการระบุรูปแบบของการบุกรุกซึ่งอาจบุกรุกซึ่งอาจบุกรุกจากจุดอ่อนของระบบหรือการละเมิดกฎรักษาความปลอดภัย โดยมีตัวตรวจจับคอยดูแลกิจกรรมต่างๆ ที่กระทำในปัจจุบันว่าเหมือนกันกับพฤติกรรมการบุกรุกที่เคยเกิดขึ้นมาก่อนหรือได้รับรายงายว่าเป็นการบุกรุกหรือไม่ ในบางระบบมีการใช้กฎ (Rule-based expert system) โดยตั้งกฎขึ้นจากพฤติกรรมที่น่าสงสัย เช่น การ login ที่ล้มเหลวเกินกว่า 3 ครั้งต่อเนื่องกันในเวลา 5 นาที ถือว่าพยายามบุกรุก และข้อมูลที่ใช้ตรวจสอบจะถูกนำมาเปรียบเทียบกับกฎ ที่มีอยู่สังเกตว่ามีการนำข้อมูลทางเวลาเข้ามาพิจารณาด้วย การตรวจจับการบุกรุกโดยวิธีนี้สามารถมีการแก้ไขกฎหรือเพิ่มกฎได้ ในระบบที่เป็นปัญญาประดิษฐ์ (Artificial Intrlligence) ระบบอาจทำการแก้ไขกฎหรือเพิ่มกฎได้ด้วยตนเอง

ข้อเสียของวิธีเปรียบเทียบพฤติกรรมผู้ใช้กับรูปแบบการบุกรุกที่รู้จักคือ

1. ประสิทธิภาพของระบบการตรวจจับชนิดนี้ขึ้นอยู่กับรูปแบบการบุกรุกที่ระบบรู้จัก

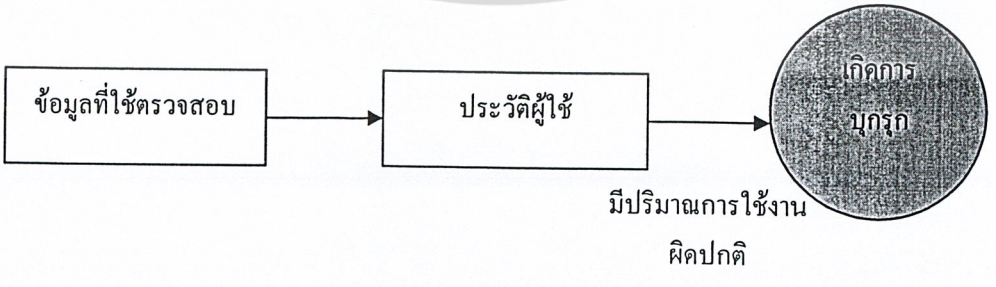
2. มีข้อจำกัดในเรื่องจำนวนของรูปแบบในการบุกรุก ซึ่งหากเป็นการบุกรุกที่ระบบไม่รู้จักมาก่อนจะทำให้ไม่สามารถตรวจจับการบุกรุกได้

5.4.2 วิธีตรวจสอบการใช้งานระบบที่ผิดปกติ (Abnomaly Intrusion Detection)

วิธีตรวจสอบการใช้งานทรัพยากรระบบที่ผิดปกติตั้งอยู่บนสมมติฐานว่าการกระทำใดๆ ที่เป็นการบุกรุกจะต้องมีการใช้งานระบบอย่างผิดปรกติ โดยมีกระบวนการเก็บประวัติพฤติกรรมการใช้งานของผู้ใช้ และสังเกตการทำงานที่เกิดขึ้นของผู้ใช้ว่าเมื่อเข้ามาในระบบได้กระทำสิ่งใดบ้าง แล้วรายงานเก็บเป็นประวัติซึ่งสามารถใช้เป็นข้อมูลตรวจสอบในการนำมาเปรียบเทียบว่าพฤติกรรมในปัจจุบันมีการใช้งานระบบที่ผิดปกติว่าประวัติพฤติกรรมของผู้ใช้เดิมมากน้อยเพียงใด หากมีปริมาณการใช้ในปริมาณมากผิดปกติจึงถือว่าเกิดการบุกรุก

5.4.3 ปัญหาของการตรวจจับโดยวิธีตรวจสอบการใช้งานระบบที่ผิดปกติ คือ

1. อาจมีพฤติกรรมการใช้งานทรัพยากรระบบของผู้ใช้ที่ผิดปกติเกิดขึ้นแต่ไม่ได้เป็นการบุกรุกระบบทำให้ระบบสถานะผิดพลาด
2. ตรวจไม่พบการบุกรุกระบบเนื่องจากการบุกรุกนั้นไม่ได้ใช้ทรัพยากรระบบอย่างผิดปกติ
3. หากผู้บุกรุกค่อยๆ เปลี่ยนพฤติกรรมการใช้งานไปทีละเล็กทีละน้อยระบบจะไม่สามารถตรวจจับความผิดปกติได้



รูปที่ 5.1 การทำงานขอการตรวจจับผู้บุกรุกโดยวิธีตรวจสอบการใช้งานระบบที่ผิดปกติ

5.5 แนวทางการปฏิบัติเมื่อมีการบุกรุกระบบ

1. จัดตั้งกลุ่มที่รับผิดชอบในการจัดการได้ทันทีที่ทราบว่าถูกบุกรุก

2. กำหนดแนวทางในการรับมือ เช่น ควรให้ความสำคัญกับการจัดการให้ระบบเครือข่ายทำงานได้เป็นปกติในสภาพเดิมเร็วที่สุด หรือให้ความสำคัญในการหาตัว ผู้บุกรุก ผู้ดูแลระบบสามารถถอดสายเน็ตเวิร์คโดยทันทีหรือไม่หากพบว่ามีผู้บุกรุกกำลังทำความเสียหมายให้กับระบบอยู่เมื่อพบผู้บุกรุกแล้วจะคอยดูพฤติกรรมต่อไปหรือจัดการทันที ให้ลองคิดปัญหาที่เกิดขึ้นแล้วหาทางการดำเนินการก่อนที่จะเกิดขึ้นจริง เพราะเมื่อเกิดเหตุการณ์ขึ้นแล้วจะไม่มีเวลาคิด

3.ให้กำหนดแนวทางในการแจ้งปัญหา เช่น หากเกิดปัญหาแล้วจะแจ้งผู้บังคับบัญชาในลำดับสูงขึ้นไปก่อน หรือ แจ้งหน่วยงานที่เกี่ยวข้องเลย ต้องแจ้งเหตุการณ์การบุกรุกที่เกิดขึ้นกับหน่วยงานที่คอยให้คำปรึกษาและช่วยเหลือใด (ในอเมริกามีหลายหน่วยงาน เช่น FIRST หรือ CERT) ต้องแจ้งตำรวจหรือไม่ จะแจ้งเหตุการณ์นี้กับหุ้นส่วนทางธุรกิจหรือไม่ จะปิดข่าวนี้หรือไม่

4. ให้จัดระบบและขั้นตอนในการดำเนินงานในการเก็บบันทึก วิเคราะห์ และเฝ้าติดตามข้อมูลทันทีประเด็นสำคัญในการหาร่องรอยของผู้บุกรุก คือ ต้องมีการเก็บบันทึกไว้ได้อย่างเพียงพอสำหรับการวิเคราะห์

5. ให้คำแนะนำกับผู้ใช้ทุกคนให้ทราบเกี่ยวกับการป้องกันการบุกรุก

5.6 ใช้สถิติในการวิเคราะห์หาความผิดปกติ (Statistical Anomaly Detection)

จะใช้การเก็บเป็นสถิติจากพฤติกรรมใช้งานปกติ โดยมีตัวแปรที่ต้องสนใจคือ ผู้ใช้, กลุ่มของผู้ใช้, เครื่องที่ใช้, เครื่องที่ให้บริการ, ไฟล์, อุปกรณ์เชื่อมต่อเครือข่าย และอื่นๆ พื้นฐานคือ ถ้าหากว่ามีการใช่งานผิดปกติไปจากเดิมนั่นเอง อาทิ เช่น เวลาที่เข้าใช้งาน โปรแกรมที่ใช้ เป็นต้น

ข้อดีคือ

- เข้าใจได้ง่ายอาศัยสถิติเข้าช่วย
- จำนวนของตัวแปรในรูปแบบที่ใช้ไม่มากนักทำให้ใช้หน่วยความจำในการจัดเก็บน้อย
- สถิติที่ได้ขึ้นอยู่กับเวลาโดยมีการหาค่าเฉลี่ย ค่าน้ำหนัก และตัวแปรภายใน
- พฤติกรรมโดยรวมอย่างง่ายคือ การเข้าใช้สิทธิ์ผิดพลาด ทำให้ผู้ใช้งานเข้าใจได้ง่าย

ข้อเสียคือ

- ยังรวมไปถึงข้อมูลที่อาจจะไม่ต้องทำเป็นสถิติด้วย

- รวมค่าจากหลายตัวแปรอาจจะทำให้ได้สถิติที่ไม่ถูกต้อง
- ไม่มีมาตรฐานที่แน่นอนรับรอง
- พฤติกรรมของผู้ใช้งานแต่ละคนไม่เหมือนกัน
- แฮกเกอร์ผู้ที่รู้สถิติจะเข้าใจและพยายามหลีกเลี่ยงการตรวจสอบและเปลี่ยนไปใช้วิธีการอื่นแทน
- ผู้โจมตีอาจจะให้หลายสิทธิ์ในการแสดงพฤติกรรมที่แตกต่างหลีกเลี่ยงการตรวจจับได้
- ค่าเฉลี่ยเวลาที่ใช้ในการแสดงพฤติกรรมเพื่อโจมตีแตกต่างกัน

5.7 การตรวจสอบจากรูปแบบ (Pattern Matching Detection)

เป็นการกำหนดรูปแบบของการโจมตีไว้โดยดูจากเหตุการณ์ที่ใช้โดยเฉพาะรูปแบบที่ใช้คือ เหตุการณ์, ลำดับของเหตุการณ์, ภาพรวมของเหตุการณ์ หรือ ใช้ เรกูลาร์ เอคเพลสชั่น ซึ่งรวม AND, OR แต่ละอย่างเพื่ออนุญาตให้ใช้งานโดยใช้ ไฟไนท์-สเตตแมชชีน, ระบบ รูล-เบสด์ หรือ ทรีช่วยในการตัดสินใจ (Decision Tree), ระบบผู้เชี่ยวชาญ (Expert System), เครือข่ายนิวรอน (Neural Network), ระบบฟัซซี่ (Fuzzy Classification System) หรือโมเดลความน่าจะเป็นโดยใช้เหตุผล (Probabilistic Reasoning Model)

ข้อดี คือ จำนวนและชนิดของเหตุการณ์ที่พบจำเป็นต้องมีรูปแบบเหมือนกันและพยายามจะเพิ่มการคำนวณให้เป็นเลขทศนิยมด้วย

ข้อเสียคือ

- การขยายตัวหรือเพิ่มประสิทธิภาพเป็นหน้าที่ของขนาดและโครงสร้างของฐานข้อมูลรูปแบบหรือกฎที่ใช้
- การเพิ่มรูปแบบให้ฐานข้อมูล, ไม่มีการเรียนรู้ได้เอง

5.8 Realtime หรือ Internet Based

เป็น Realtime IDS มีเหตุการณ์ที่ต้องการจะตรวจสอบอยู่สามส่วนประกอบดังนี้

1. **Event Generator**

   ส่วนการจัดการข้อมูลเกี่ยวกับกิจกรรมของระบบ อาทิเช่น audit trails จาก network traffic หรือจากระบบย่อย และ โปรแกรมเป็นไฟร์วอล

2. **Activity Profile**

ดูแลสถานะของระบบ หรือ เครือข่ายที่ตรวจสอบถ้ามีเหตุการณ์ที่เกิดมีอยู่จาก data source จะมีการปรับปรุงในส่วน Activity Profile หรือจัวกำหนดสร้างตัวแปรขึ้นอยู่ดับกฎที่ใช้

การตอบกลับมีความจำเป็นในแบบจำลองเหตุการณ์ที่เกิด rule-based มีการเรียนรู้และเพิ่มกฎใหม่ถ้า Rule set ตรวจพบการเปลี่ยนแปลงใน Activity Profile อย่างเช่น เปลี่ยนแปลงชนิด, ความถี่การใช้งานหรือรายละเอียดของเหตุการณ์ที่มาจาก Event Generator

3. **Rule set**

เป็นตัวกำหนดว่าจะให้แสดงการโต้ตอบอย่างไรเมื่อมีการโจมตีเกิดขึ้น

5.9 ประเภทของการโจมตี (Class of Attacks)

จุดเริ่มต้น (Point of Origin)	ผู้ใช้งานภายใน (Internal User)	ผู้ใช้งานภายนอก (Extern User)
การปิดการให้บริการ (Denial of Service)	การรบกวน (Annoying)	การรบกวน (Annoying)
การเพิ่มระดับของสิทธิ์ (Increased Privilege)	ปัญหาระดับกลาง (Moderately Serious)	ปัญหาระดับความเสี่ยงสูง (Serious Risk)
ระดับผู้ดูแลระบบ (Superuser Privilege)	ปัญหาระดับความเสี่ยงสูงมาก (Very Serious)	ปัญหาความเสียหายต่อระบบ (Disaster)

ตารางที่ 7.2 ประเภทของการโจมตี

แบ่งการโจมตีออกเป็น 2 ประเภท

1. ผู้ใช้งานภายใน (Internal Users)
2. ผู้ใช้งานภายนอก (External Users)

ซึ่งระดับความปลอดภัยยังขึ้นอยู่กับกลุ่มของผู้ใช้งาน มีสิทธิ์ในแต่ละระดับไม่เท่ากันจากตารางจะเห็นได้ว่าเป็นระดับแสดงให้เห็นถึงรูปแบบโปรแกรมที่ใช้งานอีกด้วยจะผูกติดกับความน่าสนใจของปัญหาที่ควรจะใส่ใจในการโจมตีไว้ด้วย

5.9.1 ผู้ใช้งานภายใน (Internal Users)

การโจมตีการปิดบริการให้บริการภายในเครือข่าย (Internal Denial of Service Attack) เป็นการปฏิเสธการให้บริการจากภายในเครือข่ายปัญหานี้เกิดจาก

1. การใช้ทรัพยากรของระบบมาก
2. เขียนโปรแกรมประเภทใช้หน่วยความจำในบัฟเฟอร์เกินกว่าจะรองรับได้
3. เพิ่มภาระคิวงานให้กับพริ้นเตอร์
4. สร้างการติดต่อกับอุปกรณ์ input/output แบบขนาด เช่น ดิสก์

การเพิ่มระดับการใช้งานภายใน (Internal Privilege Escalation)

1. โปรแกรมไม่ได้ตรวจสอบขนาดของบัฟเฟอร์ทำให้เกิด buffer overflow attack
2. โปรแกรมไม่ได้ตรวจสอบตัวแปรอินพุตต่างๆ

3. โปรแกรมที่ใช้ทรัพยากรของระบบมาก
4. โปรแกรมมีกลยุทธ์ที่ใช้งานมาก

การเพิ่มสิทธิ์เป็นผู้ดูแลระบบภายใน (Internal Superuser Privilege)

ในระดับผู้ดูแลระบบหรือรูท (root) มีสิทธ์ที่ใช้งานโปรแกรมต่างจากผู้ใช้งานทั่วไปมาก

5.9.2 ผู้ใช้งานภายนอก (External Users)

มักเป็นบุคคลที่มีสิทธิ์ในระบบซึ่งความเป็นจริงปัญหาส่วนใหญ่มักเกิดจากบุคคลภายนอกเข้ามาโจมตีระบบเพื่อก่อให้เกิดความเสียหายกับระบบ

การทำความเสียหายโดยการปิดการให้บริการจากภายนอก (External Denial Of Service Threats)

การโจมตีเพื่อการปิดบริการจากภายนอกเป็นการยากที่จะป้องกันและแก้ไข แม้ว่าจะมี ไฟร์วอลหรือ สกรีนิงเราเตอร์ (จากที่ได้กล่าวไว้ในบทที่ 6) แต่ก็ไม่ได้ช่วยมากนัก

การเพิ่มสิทธิ์ในการใช้งานของผู้ใช้งานภายนอก (External Privilege Escalations)

มี 2 ทางที่ใช้ในการเพิ่มระดับของผู้ใช้งานจากระยะไกล

1. โปรแกรมไม่ยอมให้เข้าใช้สิทธิ์แต่ยอมรับการเชื่อมต่อเครือข่าย
2. ผู้ใช้งานระยะไกล พยายามเข้าใช้สิทธิ์มายังระบบ โดยใช้โปรแกรมดักฟังการเชื่อมต่อจากภายนอก

บทที่ 6
การจำแนกและบันทึกพฤติกรรมผู้บุกรุก

6.1 หลักการ

ในการทำงานของส่วนการจำแนกและคัดแยก ระหว่างผู้ใช้งานทั่วไปกับผู้บุกรุกนั้นเป็นส่วนการทำงานของตัวเกตเวย์ เรียกกันว่าฮันนี่วอล (Honeywall) ซึ่งเป็นส่วนที่ใช้เพื่อเป็นทางผ่านของชิ้นข้อมูลทุกชิ้นที่จะเข้ามาในระบบเครือข่าย ในลักษณะการติดตั้งตัวเกตเวย์ไว้ที่ส่วนที่เสมือนประตูทางเข้าออกนี้ เป็นส่วนที่สามารถใช้ประโยชน์ของระบบตรวจจับผู้บุกรุก (snort) เข้าไปเพื่อให้การวิเคราะห์และจำแนกผู้บุกรุกนั้นมีประสิทธิภาพสูงสุด แต่ในส่วนการทำงานเพียงลำพังของระบบตรวจจับผู้บุกรุก นั้นไม่เพียงพอที่จะจัดการให้ผู้บุกรุกเข้าสู่ระบบกับดักที่จัดเตรียมไว้ได้ เนื่องจากระบบตรวจจับผู้บุกรุกนั้นไม่มีความสามารถเพียงพอในการจัดการด้านช่องทางการเชื่อมต่อ ซึ่งความสามารถนี้เป็นความสามารถของไฟร์วอล ที่มีชื่อเรียกว่า ไอพีเทเบิล เพื่อเป็นตัวที่เพิ่มขึ้นมาเพื่อเป็นตัวจัดการ การเชื่อมต่อสื่อสารเพื่อให้เป็นไปตามความต้องการของผู้พัฒนา แต่ในการทำงานจริงนั้น ทั้งสองโปรแกรมนี้ไม่สามารถพูดคุยเพื่อควบคุมกันได้ เพราะความสามารถที่มีนั้นเป็นแค่เพียงไฟร์วอลนั้นส่งค่าของชิ้นข้อมูลไปยังส่วนที่โปรแกรมในส่วนยูเซอร์โหมด (user mode) ก็คือตัวระบบตรวจจับผู้บุกรุก (snort) เนื่องจากว่าไฟร์วอลเองเป็นโปรแกรมที่ทำงานในส่วนของเคอร์เนลโหมด (kernel mode) ดังนั้นทางผู้พัฒนาจึงได้คิดแนวทางใหม่ออกมาได้ว่า หากความสามารถของทั้งสองโปรแกรมนี้สามารถเชื่อมต่อกันได้ จะทำให้ประสิทธิภาพการทำงานของระบบจำแนกแยกแยะผู้บุกรุกนั้น มีประสิทธิภาพที่ดีมากกว่าที่เป็นอยู่ ด้วยเหตุนี้เองผู้พัฒนาจึงได้คิดหาแนวทาง โดยการสร้างโปรแกรมที่ใช้เป็นตัวกลาง เพื่อสื่อสารระหว่างระบบตรวจจับผู้บุกรุกกับไฟร์วอลเป็นตัวที่ใช้การอ่านค่าจากการแจ้งเตือนจากระบบตรวจจับผู้บุกรุก และอ่านค่าที่ได้มานั้นแปลงเป็นคำสั่งของไอพีเทเบิล อย่างเหมาะสมเพื่อสั่งให้จัดส่งชิ้นข้อมูลไปยังปลายทางที่ถูกต้อง ที่กล่าวมานี้เป็นแนวทางในการสร้างส่วนจำแนกระหว่างผู้บุกรุกกับผู้ใช้งานทั่วไป

6.2 การทำงาน

ในส่วนการทำงานของระบบฮันนี่วอลนั้น จะอ่านข้อมูลที่ผ่านเข้าออกตลอดเวลา ในส่วนการติดต่อเข้ามาใจระบบเครือข่ายนั้นจะพิจารณาว่าชิ้นข้อมูลที่กำลังผ่านเข้ามานั้น ควรจะถูกส่งไปยังที่ใดในระบบเครือข่าย ผู้ใช้ที่พิจารณามีเพียงสองประเภท ดังนี้

ผู้ใช้งานทั่วไป เป็นผู้ที่ติดต่อมายังระบบเครือข่ายที่มีรูปแบบการติดต่อเข้ามาอย่างปกติ โดยไม่มีรูปแบบการส่งชิ้นข้อมูลที่น่าสงสัย หรือกระทำพฤติกรรมที่น่าสงสัย ดังนั้นตัวฮันนี่วอลเองจะส่ง

ชิ้นข้อมูลที่ติดต่อเข้ามาให้ไปยังเครื่องปลายทางที่ผู้ใช้งานปกติต้องการ โดยจะกล่าววิธีจำแนกในหัวข้อต่อไป

ผู้บุกรุกหรือผู้ที่มีพฤติกรรมน่าสงสัย เป็นส่วนการติดต่อมายังระบบเครือข่ายที่มีรูปแบบการติดต่อที่น่าสงสัยเข้าข่ายการโจมตี เมื่อผู้บุกรุกต้องการโจมตีไปยังเครื่องที่ให้บริการจริง ระบบในส่วนของฮันนี่วอลเองจะตรวจสอบได้ว่าเป็นผู้บุกรุก จึงจัดการส่งให้การเชื่อมต่อนั้นเปลี่ยนแปลงให้ไปติดต่อในส่วนของกับดัก แต่การเชื่อมต่อนี้ผู้บุกรุกเอง เมื่อตรวจสอบการเชื่อมต่อที่เครื่องของผู้บุกรุกเอง จะพบว่าได้ติดต่อไปยังเครื่องให้บริการจริง ในส่วนการสร้างนี้ยังไม่รวมความรับผิดชอบในเรื่องของความแนบเนียนในกับดัก เช่นเมื่อใช้คำสั่งที่แสดงตัวตนของเครื่องที่ได้เชื่อมต่อไป ก็จะพบว่าไม่ใช่เครื่องที่ให้บริการจริง แต่จะได้รับการพัฒนาในปีต่อไป

6.3 วิธีจำแนก

6.3.1 จำแนกว่าเป็นผู้บุกรุก

ในการจำแนกผู้บุกรุกนั้นผู้พัฒนาได้ปรับเปลี่ยนกฎของระบบตรวจจับผู้บุกรุก (snort) เพื่อให้มีความสามารถใจการพิจารณาและแยกแยะได้ว่าการเชื่อมต่อเข้ามาเช่นไรที่เป็นรูปแบบของการบุกรุก เพียงแต่ว่าหากใช้กฎที่ผู้พัฒนาได้เปลี่ยนแปลงนี้กับระบบที่มีผู้ใช้จำนวนมากหรือเป็นแหล่งให้บริการทางระบบเครือข่าย เช่น ไอเอสพี เมื่อมีการเชื่อมต่อจำนวนมากความเสี่ยงในการที่ผู้ใช้งานทั่วไป จะถูกส่งไปยังเครื่องกับดักและจะทำให้เกิดความเสียหายทางธุรกิจได้ แต่หากว่าเป็นองค์กรที่ต้องการศึกษาวิจัยการโจมตี เพื่อค้นหาการโจมตีรูปแบบใหม่ๆ ก็อาจจะดัดแปลงกฎของระบบตรวจจับผู้บุกรุกเองหรือใช้ กฎที่ผู้พัฒนาได้ออกแบบไว้แล้ว จะยกตัวอย่างกฎบางส่วนที่ได้แก้ไขแล้วดังนี้

6.3.1.1 Mysql cazz exploit

กฎของการตรวจสอบการบุกรุกของเครื่องมือที่ใช้เข้าโจมตีโปรแกรมฐานข้อมูล ที่ชื่อว่ามายเอสคิวแอล (mysql) โดยชื่อโปรแกรมที่เรียกว่า cazz เพื่อเข้าโจมตีแบบยึดครองระบบ กฎเดิมมีดังนี้

```
alert tcp $EXTERNAL_NET any -> $SQL_SERVERS 3306
(msg:"MYSQL root login attempt";
flow:to_server,established;
content:"|0A 00 00 01 85 04 00 00 80|root|00|";
classtype:protocol-command-decode; sid:1775; rev:2;)
```

6.3.1.2 อธิบายกฎ Mysql cazz exploit

alert tcp $EXTERNAL_NET any -> $SQL_SERVERS 3306

เป็นการบอกแจ้งเตือนว่าเมื่อมีการเชื่อมต่อในรูปแบบทีซีพี ที่เป็นหมายเลขประจำเครื่องภายนอกเครือข่าย โดยมีช่องทางการเชื่อมต่อจากไคล์เอนต์ พอร์ตใดๆ มายังเครื่องที่ให้บริการฐานข้อมูลที่อยู่ในระบบเครือข่าย ทางช่องทางหมายเลข สามสามศูนย์หก

(msg:"MYSQL root login attempt";

โดยให้มีข้อความในการแจ้งเตือนว่า “MYSQL root login attempt”

flow:to_server,established;

โดยมีการเชื่อมต่อมาโดยรูปแบบขอการสถาปนาการเชื่อมต่อ

content:"|0A 00 00 01 85 04 00 00 80|root|00|";

มีหัวข้อในชิ้นข้อมูลดังนี้ ให้ ส่งค่ากลับของระบบไปยังพอยน์เตอร์ที่ได้รอรับไว้หากว่าเป็น root ดังนี้

XXXX เป็นเซ็กเมนต์

เริ่มที่ออฟเซ็ตหนึ่งร้อย

Segment:Offset	Binary code	Assembly code
XXXX:0100	0A00	OR AL,[BX+SI]
XXXX:0102	0000	ADD [BX+DI], AL
XXXX:0104	8504	TEST AX,[SI]
XXXX:0106	0000	ADD [BX+SI],AL
XXXX:0108	807C726	CMP ว่าเป็น root หรือไม่
XXXX:010C	6F	DB เป็นการ Define byte เพื่อเปรียบเทียบ คำว่า root
XXXX:010E	747C	JZ หากว่าเป็นศูนย์แสดงว่าใช่ root ให้return
XXXX:0111	0000	ADD [BX+DI], AL
XXXX:0113	7CXX	JL จะตรวจ Carry Flag ดังนั้นจึงเป็น loop เสมอเพราะ Carry Flagติดหมายถึงว่าน้อยกว่า

classtype:protocol-command-decode; sid:1775; rev:2;)

จะใช้ส่งในรูปแบบคำสั่ง

6.3.1.3 เปลี่ยนแปลงกฎเดิมเป็นดังนี้

```
alert tcp $EXTERNAL_NET any -> $SQL_SERVERS 3306
(msg:"MYSQL root login attempt";
flow:to_server,established; )
```

เพื่อตั้งให้กฎเองมีความยืดหยุ่นที่ว่าหากเข้ามาเชื่อมต่อเข้ามายังช่องทางหมายเลข สามสาม ศูนย์หก ลักษณะการขอการสถาปนา ก็ให้ถือว่าเป็นการโจมตีจากผู้บุกรุก

6.3.2 การจำแนกว่าเป็นผู้ใช้ปกติ

ในส่วนการควบคุมหากว่าไม่ต้องการให้ตรวจสอบเครื่องบางหมายเลขจะต้องกำหนดกฎการอนุญาตให้เข้าได้โดยไม่ตรวจสอบได้โดยการกำหนดกฎที่ ไฟร์วอล เนื่องจากว่าผู้ที่ควบคุมการเชื่อมต่อจริงแล้วคือ ไฟร์วอล และในส่วนของระบบตรวจจับผู้บุกรุกนั้น หากว่าจะกำหนดให้ผู้ใช้งานปกติสามารถเข้ามาได้อย่างไม่มีปัญหาในการตรวจสอบผิดพลาดจำเป็นต้องตั้งกฎของระบบตรวจจับผู้บุกรุก ให้มีความรัดกุมและบ่งบอกชัดเจนว่าผู้ใดเป็นผู้บุกรุก เพื่อให้กระบวนการตรวจสอบนั้นไม่เกิดผลการเข้าใจผิดว่า ผู้ใช้งานปกตินั้นเป็นผู้บุกรุก ดังนั้นควรที่จะใช้กฎของระบบตรวจจับผู้บุกรุกที่ได้มากจากของเดิมโดยไม่แก้ไขใดๆ หรือแก้ไขเพียงบางส่วนแต่ยังคงพฤติกรรมที่ชัดเจนในการแสดงว่าเป็นผู้บุกรุก ผู้ใช้ปกตินั้นระบบนี้จะถือว่า เมื่อไม่ตรงกับการตรวจสอบจากระบบตรวจจับผู้บุกรุก ให้ถือว่าเป็นผู้ใช้งานปกติทั้งหมด และไม่ควรตั้งให้ตัวกฎของระบบตรวจจับผู้บุกรุกนั้นส่งผ่านข้อมูลโดยไม่สนใจผ่านคำสั่งพาสส์ (pass) ซึ่งผู้บุกรุกเองอาจจะใช้ช่องทางนี้เพื่อทำการหลีกเลี่ยงการตรวจสอบได้

6.4 วิธีการล่อหลอก และ ดักเฝ้าดูพฤติกรรมผู้บุกรุก

ในส่วนการล่อหลอกผู้บุกรุกนั้นระบบฮันนี่พ็อตที่ใช้ทฤษฎีตั้งต้นเดิม (Original comcept) นั้นจะใช้การเปิดเครื่องที่อ่อนแอที่สุดในระบบเพื่อใช้ให้เป็นเครื่องกับดัก จากนั้นรอให้ผู้บุกรุกสนใจและเข้ามาในกับดักเอง ในกับดักนั้นได้มีการติดตั้งเครื่องมือ ที่ใช้ดักจับเฝ้าดูพฤติกรรม แต่หากว่าผู้บุกรุกเองไม่ได้เข้ายังเครื่องกับดักด้วยเหตุที่ว่าทราบหมายเลขประจำเครื่องของเครื่องให้บริการจริงแล้วหรือไม่สนใจเครื่องกับดักก็ตามระบบฮันนี่พ็อตตามทฤษฎีเดิมก็จะหมดความหมายไปในทันที

แต่ในส่วนของฮันนี่พ็อตตามทฤษฎีที่กลุ่มผู้พัฒนาได้คิดนั้นจะไม่รอให้ผู้บุกรุกสนใจกับตัวกับดักเอง แต่ได้ออกแบบให้มีการบังคับให้ผู้บุกรุกเข้ามายังเครื่องกับดักเองโดยไม่ทันได้รู้ตัวว่าได้เข้ามายังเครื่องกับดักแล้ว ดังนั้นจะยังไงก็ตามระบบฮันนี่พ็อตเองก็ยังคงใช้ได้ซึ่งประโยชน์ตามวัตถุประสงค์ และที่ตัวกับดักเองก็ได้ติดตั้งระบบจัดเก็บพฤติกรรม ผู้บุกรุกที่อยู่ในระบบได้ และส่วนที่ได้เพิ่มขึ้นมาคือ เมื่อกับดักเองบอบช้ำเกินกว่าที่ผู้ดูแลเห็นสมควร ระบบจะมีรูปแบบการป้องกันกับดัก ซึ่งจะได้อธิบายในบทถัดไป

6.4.1 เครื่องมือจากทฤษฎีตั้งต้น

เครื่องมือที่มีเพื่อการจัดทำเป็นระบบฮันนี่พ็อตนั้น มีชื่อเรียกและคุณประโยชน์ดังนี้

- **iptables** เป็นโปรแกรมทางฝั่งเคอร์เนลโหมด (Kernel mode) ซึ่งเรียกว่าเป็นไฟร์วอล ใช้เพื่อจัดการกับส่วนการเชื่อมต่อ การอนุญาตให้ชิ้นข้อมูลทางระบบเครือข่ายผ่านเข้าออก ในส่วนการใช้งานเพื่อให้ส่งค่าข้อมูลทางระบบเครือข่ายเข้าไปยังส่วนที่โปรแกรมทางยูสเซอร์โหมด (User mode) สามารถนำข้อมูลเหล่านั้นมาใช้ได้โดยผ่านโมดูลที่ชื่อว่า ip_queue
- **snort** เป็นโปรแกรมทางฝั่งยูสเซอร์โหมด (User mode) ซึ่งเรียกว่าเป็นไอดีเอส ซึ่งเป็นระบบตรวจจับผู้บุกรุก โดยในส่วนการใช้งานเพื่ออ่านค่าชิ้นข้อมูลแล้วนำมาทำแพตเทิลแมชชิง ว่าเข้าข่ายการโจมตีหรือไม่เพื่อกำหนดตามคำสั่งคอนโทรลดังนี้
  - **alert** เป็นการแจ้งเตือนว่าตรงกับกฎที่ได้ตั้งไว้
  - **drop** เพื่อตัดสินหยุดการส่งผ่านข้อมูลชิ้นเมื่อตรงกับกฎที่ได้ตั้งไว้
  - **pass** เพื่อไม่สนใจในชิ้นข้อมูลเมื่อตรงกับกฎที่ได้ตั้งไว้
- **sebek** เป็นโปรแกรมที่ใช้เพื่อรับผิดชอบการดูพฤติกรรมและจัดเก็บพฤติกรรมของผู้บุกรุก เพื่อใช้เป็นแหล่งข้อมูลในการ นำมาวิเคราะห์ว่าผู้บุกรุกได้กระทำการใดบ้างเมื่อเข้ามาติดอยู่ในกับดัก โดยตัวเซเบค เองก็มีส่วนการรับผิดชอบดังนี้
  - sebek-client เรียกว่า เซเบค ไคล์เอน จะซ่อนตัวอยู่ในกับดักเพื่อรอดูพฤติกรรมของผู้บุกรุก และคอยส่งข้อมูลนั้นไปยังเครื่องที่ใช้เก็บพฤติกรรมลงฐานข้อมูล โดยผ่านช่องทางที่มีความปลอดภัย มีการเข้ารหัสในรูปแบบของ เซเบคเอง และใช้การส่งเป็นแบบยูดีพี
  - sebek-server เรียกว่า เซเบค เซิร์ฟเวอร์ จะทำหน้าที่รับฟังเพื่อรับค่าข้อมูลที่เซเบค ไคล์เอนดักจับข้อมูลจากกับดัก และระบบการทำงานจะส่งค่าไปเก็บยังฐานข้อมูลที่ได้เตรียมไว้

6.4.2 เครื่องมือจากทฤษฎีใหม่

เครื่องมือที่ได้พัฒนาขึ้นมาใหม่นั้นเพื่อใช้เป็นผู้สื่อสารกลางระหว่าง ไฟร์วอล (ไอพีเทเบิล) กับ ระบบตรวจจับผู้บุกรุก (สนอร์ท) เพื่อให้สามารถสื่อสารระหว่างกันได้ และควบคุมชิ้นข้อมูลที่ผ่านเข้าออก จากระบบเครือข่ายได้อย่างมีประสิทธิภาพมาก โปรแกรมที่ได้สร้างขึ้นมีดังนี้

- **s2i** เรียกว่า เอสทูไอ (**S2I**: Snort command to iptables)เป็นโปรแกรมที่สร้างจากภาษาคอมไพล์เลอร์ เพื่อให้สามารถแปลงการแจ้งเตือนของตัวระบบตรวจจับผู้บุกรุกที่สามารถตรวจเจอผู้บุกรุก ให้แปลงเป็นคำสั่งของไฟร์วอล (ไอพีเทเบิล) ได้อย่างเหมาะสมเพื่อให้ชิ้นข้อมูลที่มาจากผู้ใช้งานปกติไปยังส่วนที่ถูกต้องตามต้องการ และในส่วนของผู้บุกรุกเองนั้นก็ได้ส่งไปยังเครื่องกับดัก ตัวเอสทูไอสามารถกำหนดเปลี่ยนแปลงค่า ต่างๆ ได้เพื่อให้เกิดความเหมาะสมในการใช้งานกับระบบเครือข่ายนั้นๆ ซึ่งมีโปรแกรมเสริมดังนี้
  - **configMM** เป็นส่วนการทำงานเพื่อกำหนดค่าจากไฟล์ที่ชื่อว่า s2i.conf เป็นส่วนการกำหนดค่า ของหมายเลขประจำเครื่องกับดัก และส่วนการบอกชื่ออินเตอร์เฟส การเชื่อมต่อของตัวยันนี่วอล ว่ามีการส่งชิ้นข้อมูลเข้าออกด้วยชื่ออินเตอร์เฟสใดบ้าง และการกำหนดหมายเลขประจำเครื่องได้นั้นเพื่ออำนวยความสะดวกในการกำหนดกับดักหลายกับดักและหลายหมายเลขได้โดยกำหนดผ่านทางไฟล์ ดังกล่าว
  - **addremoveMM** จะติดต่อกับไฟล์ที่ชื่อว่า iplists.s เป็นส่วนการกำหนดค่าโดยการเพิ่ม หรือ ลบ และสามารถดู ได้ว่าหมายเลขประจำเครื่องผู้บุกรุกเองนั้น มีหมายเลขอะไรบ้างแล้วที่เข้ามาโจมตีระบบเครือข่ายของเรา และในส่วนการทำงานนี้เองเพื่อลดการสั่งคำสั่งลงไปยังไฟร์วอล เพื่อลดรันไทม์ (run time) ของระบบให้ประหยัดทรัพยากรณ์ในระบบลงได้บ้างและเพื่อตรวจสอบดูว่าผู้ใดที่เข้ามาในระบบบ้างแล้ว
- ส่วนของ Cage Management ซึ่งจะอธิบายในบทถัดไป

6.4.3 การปรับใช้และแก้ไขเครื่องมือจากทฤษฎีตั้งต้น

จากเครื่องมือดังที่ได้กล่าวมาในส่วนของหัวข้อ 6.4.1 นั้นเป็นเครื่องมือที่ได้มีการจัดเตรียมไว้ก่อนหน้าที่ผู้พัฒนาจะมาพัฒนาต่อ ดังนั้นเมื่อของเดิมมีอยู่แล้วและมีประสิทธิภาพ ดังนั้นจึงนำมาประยุกต์ใช้งานในระบบ ดังนี้

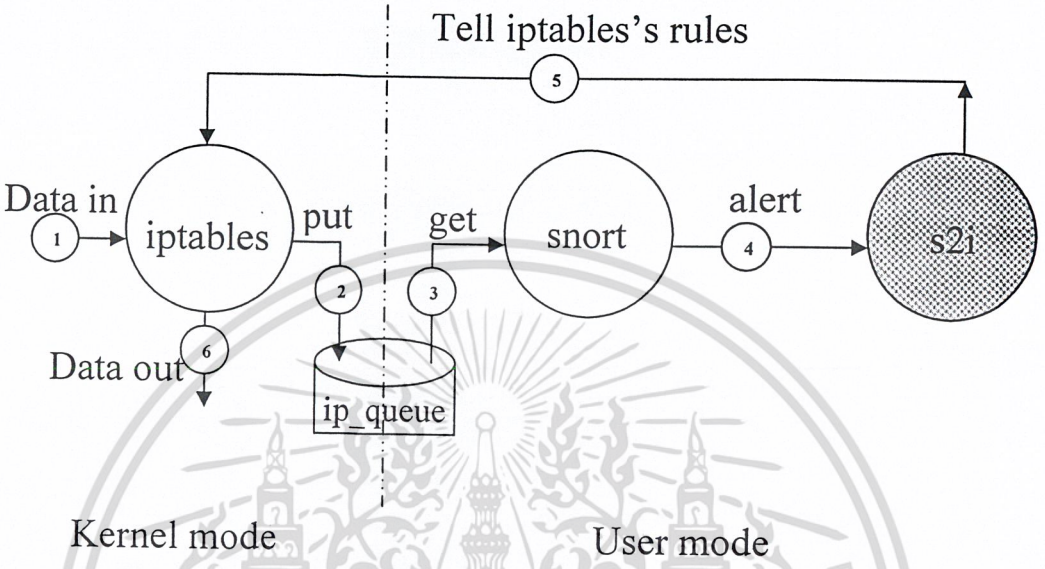
- **iptables** ตัวไฟร์วอล (ไอพีเทเบิล) นี้ศึกษาเพิ่มในส่วนของการทำ PREROUTING (DNAT: Destination Network Address Translation) เพื่อทำในส่วนการเปลี่ยนการเชื่อมต่อจากผู้บุกรุกไปยังเครื่องกับดักโดยไม่ให้ผู้บุกรุกรู้ตัวว่า ติดต่อกับเครื่องที่ไม่ใช่ที่ผู้บุกรุกต้องการ ในส่วนการทำ DNAT นั้น ต้องมีการปรับเปรียนเคอร์เนลให้เหมาะสม เพื่อใช้เป็น Bridge (IEEE 802.1d) เพื่อให้สามารถทำตัวเองเป็นเกตเวย์ได้ และใช้ความสามารถของการทำ NAT ได้ ในส่วนการทำงานของ ไฟร์วอลได้อธิบายไว้ในส่วนของบทที่ 4 เรื่องไฟร์วอล ในส่วนของการทำ DNAT นี้ตามทฤษฎีเก่าที่กล่าวมานั้นยังไม่มีการคิดค้นเพื่อใช้งาน เพียงแต่ว่าใช้ในส่วนการส่งค่าชิ้นข้อมูลทางระบบเครือข่ายไปยังส่วนที่ทำให้ยูสเซอร์โหมด อ่านค่าได้เท่านั้น แต่ในส่วนการพัฒนาตามทฤษฎีใหม่นี้ ได้ดึงความสามารถของ ไฟร์วอล (ไอพีเทเบิล) มาใช้อย่างเต็มประสิทธิภาพ และให้ผลลัพธ์ที่น่าพอใจอย่างยิ่งคือ ทางผู้บุกรุกเอง เมื่อตรวจสอบการเชื่อมต่อที่เครื่องของผู้บุกรุกเองนั้นจะพบว่าได้ติดต่อไปยังเครื่องที่ผู้บุกรุกต้องการแต่แท้จริงแล้วได้ตกมาอยู่ในกับดักที่เราได้เตรียมไว้
- **snort** ตัวระบบตรวจจับผู้บุกรุกในส่วนของตัวโปรแกรมเองทางผู้พัฒนาไม่ได้เข้าไปแก้ไขแต่ประการใด แต่ได้เข้าไปเปลี่ยนแปลงแก้ไขในส่วนของกฎที่ใช้เพื่อทำแพทเทิลแมชชิงกับข้อมูลที่อ่านขึ้นมาได้จาก ip_queue ดังที่ได้กล่าวในหัวข้อ 6.3
- **sebek** ในส่วนของเซเบคนั้นได้จัดการแก้ไขเนื้อโปรแกรมของเซเบคไคล์เอนบางส่วนและคอมไพล์เคอร์เนลให้เหมาะสมโดยการไม่เลือกในส่วนของไฟร์ซีสเตมของระบบที่เป็นตัวเพิ่มความรู้ให้กับเคอร์เนล จะทำให้ตัวเคอร์เนลเองไม่รอบรับการซ่อนตัวของเซเบค และได้แก้ไขส่วนของแฟล์กในเนื้อโปรแกรมให้เหมาะสมกับเคอร์เนลเวอร์ชัน 2.4.x และเมื่อคอมไพล์เคอร์เนลได้ตามที่แนะนำแล้วนั้น ก็จำเป็นต้องเปลี่ยนแปลงไลบรารี ของระบบให้เป็นของตัวเคอร์เนล 2.4.26 เท่านั้น เนื่องจากว่าภายในเนื้อโปรแกรมของเซเบคไคล์เอน ได้เรียกใช้ไลบรารี่ ของเคอร์เนลเวอร์ชั่นนี้โดยแท้จริง แล้วสามารถเปลี่ยนไลบรารี่ของคอมไพล์เลอร์ก็ได้เช่นกัน

6.5 สถาปัตยกรรม

6.5.1 แนวคิด

ในส่วนของแนวคิดนั้นได้เพิ่มส่วนความสามารถของระบบฮันนี่พ็อต ให้มีความสามารถสูงสุดในส่วนของการ ล่อหลอกผู้บุกรุก เนื่องจากเมื่อก่อนนั้น หากว่าผู้บุกรุกเอง ไม่เข้าไปยังเครื่องกับดัก ระบบฮันนี่พ็อตเองก็หมดซึ่งประโยชน์ในการป้องกันระบบเครือข่าย แต่ผู้พัฒนาได้เพิ่มความสามารถในการล่อหลอกผู้บุกรุก เพื่อกำจัดปัญหาความหมดประโยชน์ของฮันนี่พ็อตรุ่นเก่าออกไป โดยการใช้ความสามารถของ ไฟร์วอล และ ระบบตรวจจับผู้บุกรุกให้มีศักยภาพในการทำงานร่วมกัน เนื่องจากเมื่อก่อนนั้นสองโปรแกรมนี้ไม่สามารถเชื่อมต่อและทำงานร่วมกันได้อย่าง

เต็มที่ แต่ในเวลานี้ผู้พัฒนา ได้ออกแบบให้สามารถทำงานร่วมกันได้เพิ่มขึ้นไปอีกระดับหนึ่ง โดยการใช้ตัวสื่อสารกลางเป็นตัวแปลงคำสั่งให้เหมาะสม ดังรูปด้านล่าง



รูปที่ 6.1 ไดอะแกรมการทำงานร่วมกันได้ของไฟร์วอลและระบบตรวจจับผู้บุกรุก

จากรูปไดอะแกรมสามารถอธิบายหลักการทำงานของรูปแบบการจำแนกพฤติกรรมผู้บุกรุกได้ดังนี้

1. เมื่อมีชิ้นข้อมูลเข้ามายังตัวฮันนี่วอลจะถูกส่งมาให้ยังไฟร์วอลเพื่อรับหน้าที่แรกคือ ส่งค่าข้อมูลที่ได้รับเข้ามานั้นไปให้ยังระบบตรวจจับผู้บุกรุก ผ่านทางโมดูล ip_queue
2. ส่งค่าข้อมูลจากที่ได้รับเข้ามาไปยังส่วนของ ip_queue เพื่อให้โปรแกรมทางฝั่งยูสเซอร์โหมด สามารถอ่านไปใช้งานได้
3. ตัวระบบตรวจจับผู้บุกรุกดึงค่าชิ้นข้อมูลที่อยู่ในบ่อน้ำ ip_queue มาเพื่อตรวจสอบพฤติกรรมโดยการทำแพทเทิลแมชชิง กับกฎที่ได้กำหนดไว้
4. ในส่วนของการตรวจสอบชิ้นข้อมูลนั้นมีสองกรณีดังนี้
    4.1 เมื่อตรวจพบว่าเป็นชิ้นข้อมูลที่มีความเสี่ยงที่จะเป็นผู้บุกรุก ก็จะแจ้งเตือนออกมา
    4.2 ไม่พบว่าชิ้นข้อมูลนี้เป็นการบุกรุก ก็จะไม่แจ้งเตือนออกมา
5. ในส่วนของการทำงานตัวสื่อสารกลางนั้นมีสองกรณีดังนี้
    5.1 เมื่อมีการแจ้งเตือนออกมาจากระบบตรวจจับผู้บุกรุก ตัวสื่อสารกลางหรือที่เรียกว่าเอสทูไอ (s2i) ก็จะแปลงข้อความเพื่อบอกไฟร์วอลว่า ชิ้นข้อมูลที่ได้รับมานั้นต้องส่งไปยังเครื่องกับดัก

5.2 หากไม่มีการแจ้งเตือนตัวสื่อสารกลางจะไม่แจ้งให้ไฟร์วอลส่งชิ้นข้อมูลนั้นไปยังเครื่องกับดัก แต่จะปล่อยให้ไปติดต่อกับเครื่องที่ผู้ติดต่อเข้ามาต้องการ

6. ชิ้นข้อมูลที่ไฟร์วอลได้บอกปลายทางแล้วจะไปยังส่วนที่ถูกกำหนดไว้

6.5.2 การดูพฤติกรรม

การดูพฤติกรรมของข้อมูลที่เข้ามานั้น อาศัยหลักการส่งชิ้นข้อมูลมายังระบบตรวจจับผู้บุกรุกผ่านทางโมดูลที่ชื่อว่า ip_queue โดยหน้าที่เป็นของ ไฟร์วอลจะจัดส่งมาให้ และหลักการดูพฤติกรรมนั้นจะมีสองลักษณะดังนี้

- **Allow all Deny some** เป็นการสนใจว่า ให้ทุกคนที่อยู่ในระบบเครือข่ายนั้นเป็นบุคคลที่เชื่อถือได้ อนุญาต ให้เข้ามาได้ทั้งหมดก่อน แล้วเมื่อมีการกระทำผิดจะค่อยเพิ่มกฎการป้องกัน เพื่อให้การบริการให้แก่บุคคลทั่วไปนั้นทำได้อย่างมีประสิทธิภาพ มักใช้กับระบบงานที่ต้องให้บริการแก่บุคคลภายนอก เช่นบริษัทที่เป็นผู้ดูแลให้บริการเช่าคู่สายเชื่อมต่อระบบอินเตอร์เน็ต ในส่วนนี้จะดีต่อการติดตั้งให้แก่บริษัท หรือ ระบบเครือข่ายเพื่อธุรกิจ ดังนั้นผู้ที่อยู่ใน deny lists จะถูกส่งเข้าไปยังกับดักทั้งหมด
- **Deny all Allow some** เป็นการสนใจว่า ให้ทุกคนที่อยู่ในระบบเครือข่ายนั้นเป็นบุคคลที่เชื่อถือไม่ได้ ไม่อนุญาตให้เข้ามาในระบบเครือข่ายได้เลย หากเมื่อต้องการอนุญาตให้บุคคลใดเข้ามาในระบบได้นั้น จะต้องเพิ่มเข้ามาใน allow lists เพื่อให้หมายเลขประจำเครื่องนั้นๆ สามารถเข้ามาได้ในระบบ ซึ่งในรูปแบบเช่นนี้จะเหมาะสมในการทำวิจัยรูปแบบการโจมตีใหม่ๆ เพื่อให้ได้ทราบถึงรูปแบบการโจมตีให้ได้มากที่สุด เนื่องจากว่าการโจมตีแบบใหม่ๆ นั้นจะไม่เข้าข่ายการเข้าขอเชื่อมต่อแบบปกติ ดังนั้นบุคคลที่กระทำพฤติกรรมที่ไม่อยู่ใน allow lists จะถูกส่งเข้าไปยังกับดักทั้งหมด

6.5.3 การจำแนกระหว่างผู้ใช้งานปกติกับผู้บุกรุก

ในการจำแนกระหว่างผู้ใช้งานปกติกับผู้บุกรุกนั้น ขึ้นอยู่กับสองปัจจัยดังนี้

- ในส่วนของความต้องการขององค์กร โดยต้องเลือกใช้ว่าจะให้เป็น Allow All Deny Some หรือว่า Deny All Allow Some ขึ้นอยู่กับความเหมาะสมกับองค์กรนั้นๆ ว่าต้องการให้ระบบฮันนี่พ็อตเพื่อวัตถุประสงค์ใด หากต้องการใช้เพื่อปกป้องระบบขององค์กรในกรณีที่องค์กรเองไม่ได้เจาะจงให้บริการผู้ใช้ภายนอกทั่วๆ ไป ควรที่จะใช้รูปแบบการตั้งกฎเกณฑ์ ให้เป็นแบบ Deny All Allow Some แต่หากว่าต้องการนำระบบฮันนี่พ็อตไปใช้เพื่อป้องกันระบบเครือข่ายขององค์กรที่ต้องให้บริการแก่

บุคคลภายนอก ควรที่จะใช้รูปแบบการตั้งกฎเกณฑ์ ให้เป็นแบบ Allow All Deny Some ในส่วนนี้ต้องขึ้นอยู่กับความต้องการของผู้ที่จะนำระบบฮันนี่พ็อตไปใช้

- ในส่วนที่สองเป็นการเขียนกฎของตัวระบบตรวจจับผู้บุกรุก
  - หากว่าใช้กฎของเดิมก็จะตรวจสอบรูปแบบอย่างละเอียดซึ่งตัวระบบตรวจจับผู้บุกรุกหากใช้กฎเดิมจะตรวจพบแต่เฉพาะ รูปแบบที่เป็นการโจมตีเฉพาะเท่านั้น ดังนั้นผู้ที่จะถูกส่งไปยังกับดักนั้น จะต้องกระทำด้วยพฤติกรรมที่ถูกต้องตรงตามรูปแบบการบุกรุก ไม่ผิดเพี้ยน จะมี
    - ข้อดีที่ว่าหากผู้บุกรุกจริงจะเข้าไปยังกับดักแต่หากว่าเป็นผู้ใช้งานทั่วไปก็จะไม่หลงเข้าไปในกับดัก
    - ข้อเสีย หากว่าผู้บุกรุกใช้รูปแบบการโจมตีที่ไม่เหมือนทั้งหมด จะไม่สามารถตรวจจับได้
  - หากว่าใช้กฎที่ดัดแปลงจากของเดิม จะช่วยให้ตรวจสอบแนวการโจมตีที่คล้ายคลึงกัน หรือใกล้เคียงกันได้ตามแต่การเขียนกฎว่าตรวจสอบละเอียดเพียงใด
    - ข้อดี หากว่าผู้บุกรุกเปลี่ยนรูปแบบการโจมตีไปแต่ยังเข้าข่ายการโจมตี ก็จะถูกส่งไปยังกับดัก
    - ข้อเสีย หากว่าผู้ใช้งานทั่วไปเกิดพลาดพลั้งกระทำโดยคล้ายคลึงกับการบุกรุกเพียงเล็กน้อยก็จะถูกส่งไปยังกับดักเช่นกัน

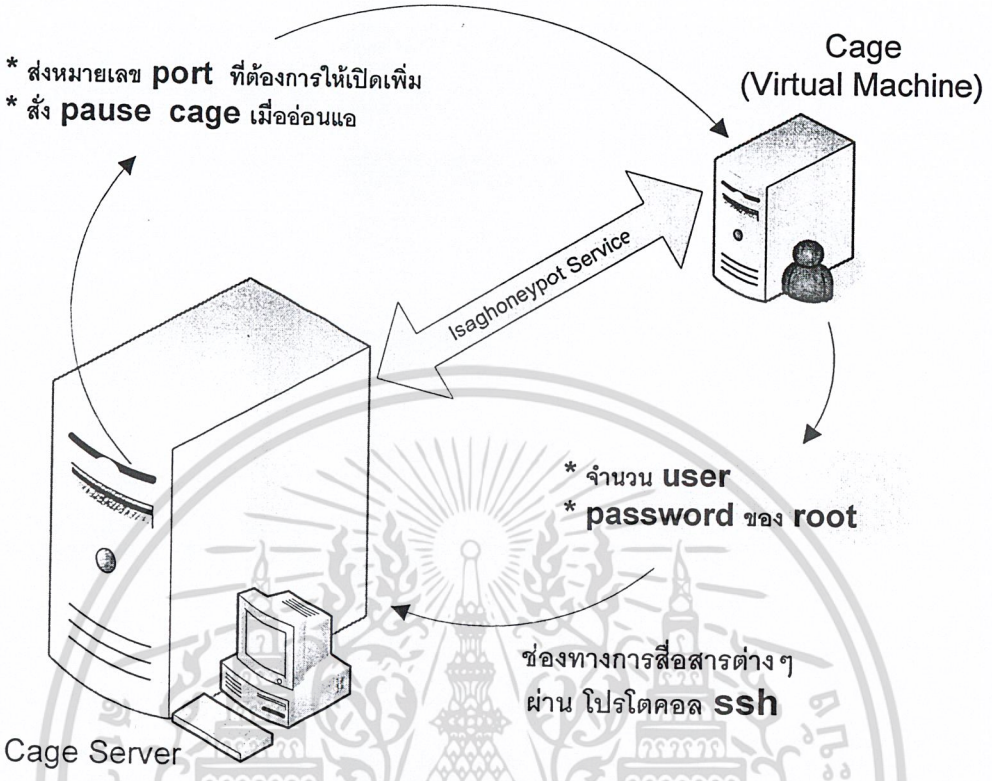
ในส่วนของการจำแนกแยกแยะจะพบว่า ไม่ใช่ว่าขึ้นกับการตั้งโครงสร้างของการอนุญาตเท่านั้นแต่ยังคงต้องอาศัยความเชี่ยวชาญในการออกกฎของระบบตรวจจับผู้บุกรุกเองด้วยเช่นกัน ดังนั้นผู้ดูแลระบบเองจำต้องมีความสามารถในการออกแบบกฎให้เหมาะสมกับการใช้งานร่วมกันกับ ความต้องการขององค์กร และระบบฮันนี่พ็อตเองด้วย หากขาดซึ่งความเชี่ยวชาญแล้วระบบให้ว่าออกแบบมาดีเพียงใด ก็เป็นเสมือนกับ ของเล่นที่ประกอบไปด้วยช่องโหว่ซะเอง อย่างนี้แล้วคงป้องกันระบบใดๆ ไม่ได้อย่างแน่นอน

บทที่ 7
การจัดการและเฝ้าระวังกับดัก

7.1 หลักการ

หลังจากที่ระบบฮันนี่พ็อตส่วนของฮันนี่วอลได้จัดเส้นทางใหม่ให้กับผู้ไม่ประสงค์ดีโดยเข้ามายังกับดักแล้ว เครื่องที่จำลองขึ้นมาเป็นกับดักก็จะทำหน้าที่ให้ผู้ไม่ประสงค์ดีเข้ามาใช้งานโดยที่ข้อมูลพฤติกรรมต่างๆที่ผู้ไม่ประสงค์ดีนั้นได้กระทำจะถูกส่งไปยังล็อคเซิร์ฟเวอร์โดยผ่านช่องทางการสื่อสารบนเซเบคโปรโตคอล (Sebek Protocol) ซึ่งเป็นช่องทางการสื่อสารที่ปลอดภัย และเครื่องที่จำลองขึ้นมานั้นจะต้องถูกตรวจสอบจากเครื่องที่ทำหน้าที่ดูแลกรงขังหรือเคจเซิร์ฟเวอร์ (Cage Server) โดยผ่านช่องทางการสื่อสารบนซีเคียวร์เชลล์โปรโตคอล (Secure Shell Protocol) ซึ่งการตรวจสอบนี้มีความจำเป็นมากเมื่อต้องการควบคุมไม่ให้กรงขังมีความอ่อนแอเกินไปเช่นมีการบุกรุกและใช้กับดักนี้เป็นเครื่องมือในการโจมตีหรือ มีการส่งข้อมูลไปยังระบบเครือข่ายมากจนเกินไป ดังนั้นกับดักจึงต้องถูกเฝ้าระวังและควบคุมอยู่เสมอเมื่อไรที่มีปัญหาเกิดขึ้นระบบจะต้องทำการปิดกับดักหรือเพิกถอนกับดักนั้นออกไปจากระบบทันทีโดยเครื่องที่ทำหน้าที่เป็นเคจเซิร์ฟเวอร์ (Cage Server) ในกรณีของระบบไอเซคฮันนี่พ็อต ได้จัดทำวิธีการการจัดการและเฝ้าระวังกับดักขึ้นโดยทำเป็นต้นแบบดังนี้

- **การจัดเตรียมช่องทางการสื่อสาร** คือการที่เครื่องที่เป็น เคจเซิร์ฟเวอร์ (Cage Server) จะติดต่อกับเครื่องที่เป็นกรงขังนั้นจะกระทำโดยผ่านโปรแกรมซีเคียวร์เชลล์ซึ่งการกระทำดังกล่าวสามารถทำได้โดยวิธิการพิสูจน์ตนแบบคู่คีย์
- **การจัดเก็บค่าเริ่มต้นของกรงขัง** คือการที่เครื่องที่เป็น เคจเซิร์ฟเวอร์ (Cage Server) เก็บค่าต่างๆที่สำคัญจากเครื่องที่จำลองเป็นกรงขังเช่นค่าต่างในไฟล์พาสเวิร์ด
- **การตรวจสอบข้อมูลที่สำคัญ (Integrity Checking)** คือการรับรองว่าข้อมูลจะไม่ถูกเปลี่ยนแปลงหรือทำลายไม่ว่าจะเป็นโดย อุบัติเหตุหรือโดยเจตนาแต่หากมีการเปลี่ยนแปลงเกิดขึ้นโปรแกรมฮันนี่พ็อตสามารถจะทำการปิดกรงขังนั้นชั่วคราวได้เพื่อให้ผู้ดูแลระบบมาตรวจสอบแก้ไขหรือเปลี่ยนกรงขังใหม่
- **การจัดการการเปิดช่องทางการสื่อสารเพิ่มเติมจากปกติ** คือการที่ผู้ดูแลระบบป้อนค่าหมายเลขพอร์ตที่ต้องการให้เครื่องที่เป็นกรงขังมีการเปิดช่องทางล่อหลอกเพิ่มเติมโดยความสามารถนี้จะต้องทำให้สมบูรณ์โดยต้องสามารถเปิดช่องทางและมีการโต้ตอบในลักษณะคอมมานด์เพื่อเก็บพฤติกรรมด้วย



รูปที่ 7.1 ลักษณะการจัดการและเฝ้าระวังกรงขัง

7.2 การทำงาน

ในส่วนของการทำงานนั้น ตัวระบบที่ใช้เพื่อตรวจสอบดูแลเครื่องกับดักนั้นจะทำการเข้าตรวจสอบความเป็นไปของกับดักโดย ส่งคำสั่งไปดึงค่า ที่ต้องการตรวจสอบซึ่งได้แก่ จำนวนผู้มีสิทธิ์ในการเข้าใช้กับดัก และ ค่าแฮชของรหัสผ่านผู้ดูแลระบบ ในส่วนการตรวจสอบนี้สามารถลดหรือเพิ่มเติมได้โดยการกำหนดค่าไว้ที่ไฟล์ cage.conf เพื่อให้ผู้ดูแลกำหนดความต้องการในการตรวจสอบดูแลกับดักไม่ให้บอบช้ำเกินกว่าความต้องการของผู้ดูแลระบบ ในส่วนการทำงานนี้สร้างขึ้นมาเพื่อให้เป็นส่วนหนึ่งในการ ปกป้องระบบภายในองค์กรที่ได้ติดตั้งตัวระบบฮันนี่พ็อตเองไม่ให้ตัวกับดักเองถูกใช้เป็นเครื่องมือในการเข้าครอบครองและใช้เป็นฐานในการโจมตีระบบเครือข่ายภายในองค์กร หากว่าไม่มีการปกป้องดูแลจะให้เกิดผลลัพธ์ที่ไม่พึงประสงค์ ในส่วนการถูกโจมตีจากภายใน ในส่วนการทำงานหลักๆ นั้นจากรูปมีการทำงานดังนี้

- ตัวเครื่องที่เป็นเคจเซิร์ฟเวอร์ (Cage server) จะติดตั้งเวอร์ชวลแมชชีนและควบคุมด้วยคำสั่งทางคอมมานด์ไลน์ (Command line) เพื่อให้สามารถควบคุมจากระยะไกลและใช้ระยะเวลาอันสั้น เนื่องจากว่า หากผู้บุกรุกเองพบเจอการเข้า

ตรวจสอบตัวกับดักจากผู้ดูแลระบบ ผู้บุกรุกจะทราบทันทีว่าตนเองเข้ามาอยู่ในกับดัก ที่ได้ถูกเตรียมไว้แล้ว ดังนั้นการไม่แสดงตัวตนของระบบฮันนี่พ็อตเองก็จะล้มเหลว

- ส่วนการควบคุมจะออกมาอยู่ในส่วนใดๆ ก็ได้ ในที่นี้ได้ออกแบบให้ระบบควบคุมเองมาอยู่ในส่วนของฮันนี่วอล เพื่อให้เป็นสัดส่วนการทำงานที่ควบคุมเป็นศูนย์กลางได้ ใช้หลักการส่งคำสั่งผ่านทางช่องทางซีเคียวเชลล์ และทำงานเป็นช่วงเวลาด้วยข้อมูลอันสั้น เพื่อให้ผู้บุกรุกเองไม่รู้ตัวว่ามีการเข้ามาตรวจสอบเครื่องที่ตนเองใช้งานอยู่ในส่วนคำสั่งเพื่อการตรวจสอบหรือสั่งงานระยะไกลนั้นใช้คำสั่งจากเครื่องฮันนี่วอลดังนี้
  - ssh root@cage ตามด้วยคำสั่งที่ต้องการ

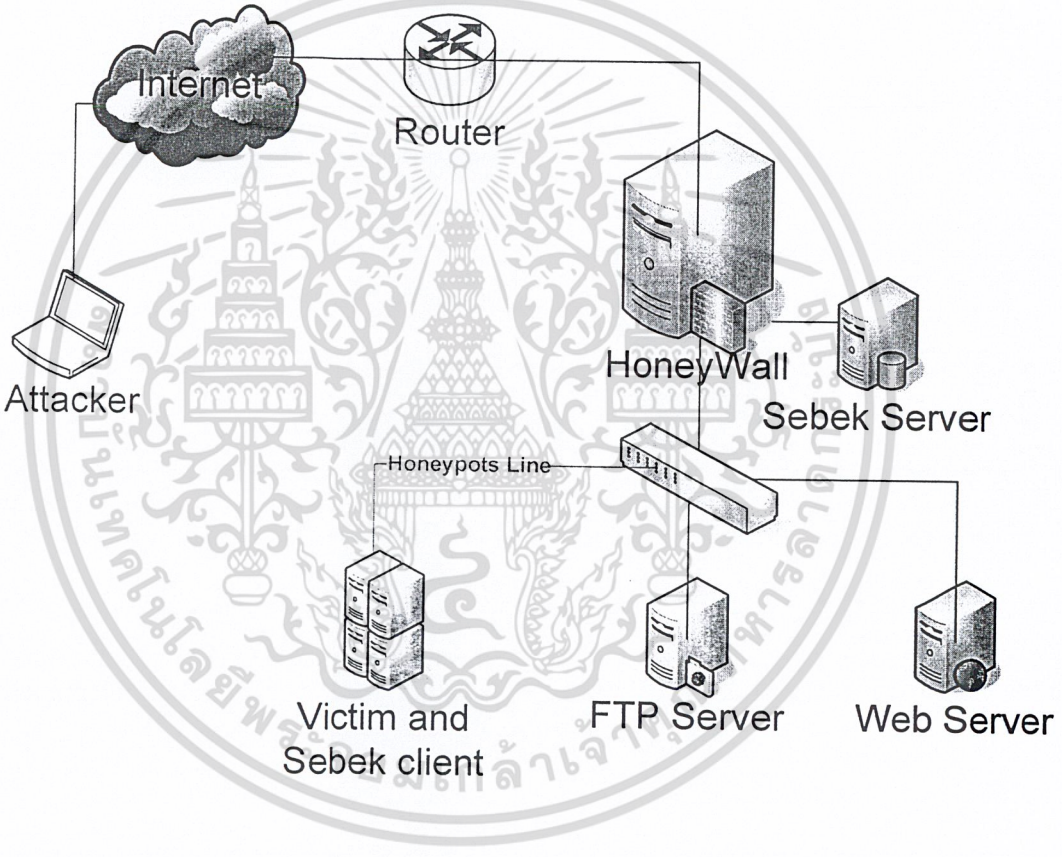
  ในการกระทำเช่นนี้เพื่อให้เกิดส่วนการทำงานที่ผู้บุกรุกเองไม่ทันเห็นว่าเครื่องที่ตนเองได้ใช้งานอยู่นั้น มีการเชื่อมต่อเข้ามาจากภายนอก แต่อย่างใดเนื่องจากว่าการเชื่อมต่อและส่งคำสั่งนี้มีระยะเวลาสั้นมาก เนื่องจากว่าได้ทำส่วนของคีย์แพร์ระหว่างเครื่องที่ควบคุมและเครื่องที่ถูกควบคุมไว้แล้วดังนั้นขั้นตอนในการ แสดงตนเพื่อเข้าขอใช้ระบบ (Authentication) จึงลดลงไป
- ที่เครื่องควบคุมเองก็มีส่วนของเซอร์วิสที่ใช้เพื่อตรวจสอบค่าที่ได้ดึงมาจากเครื่องกับดักว่าได้มีการเปลี่ยนแปลงต่างไปอย่างไรบ้าง เพื่อให้เป็นการบ่งบอกได้ว่าเครื่องกับดักบอบช้ำถึงส่วนที่ได้กำหนดไว้แล้วหรือยัง ในส่วนการทำงานนี้จะทำงานเป็นช่วงเวลาเพื่อลดการใช้ทรัพยากรของระบบและลดการเชื่อมต่อระหว่างเครื่องควบคุมกับเครื่องกับดักทำให้ผู้บุกรุกเองไม่รู้ตัว

ในส่วนการออกแบบตัวดูแลกับดักนั้นเป็นส่วนเพิ่มของการทำชุดโปรแกรมนี้เนื่องจากว่าได้เห็นว่าหากระบบที่ใช้เพื่อเป็นกับดักบอบช้ำจนเกินกว่าการควบคุมจะทำให้เกิดผลร้ายต่อระบบภายในที่ตัวฮันนี่พ็อตได้ดูแลอยู่ หากว่าการทำงานของตัวฮันนี่พ็อตนั้นดีเพียงใดแต่หากว่า เมื่อใดที่กับดักที่เราได้เก็บผู้บุกรุกไว้นั้น มีรอยรั่วทำให้ผู้บุกรุกเองออกมาทำลายระบบเครือข่ายภายใต้การดูแลของระบบฮันนี่พ็อตได้

บทที่ 8

การทดลองและผลการทดลอง

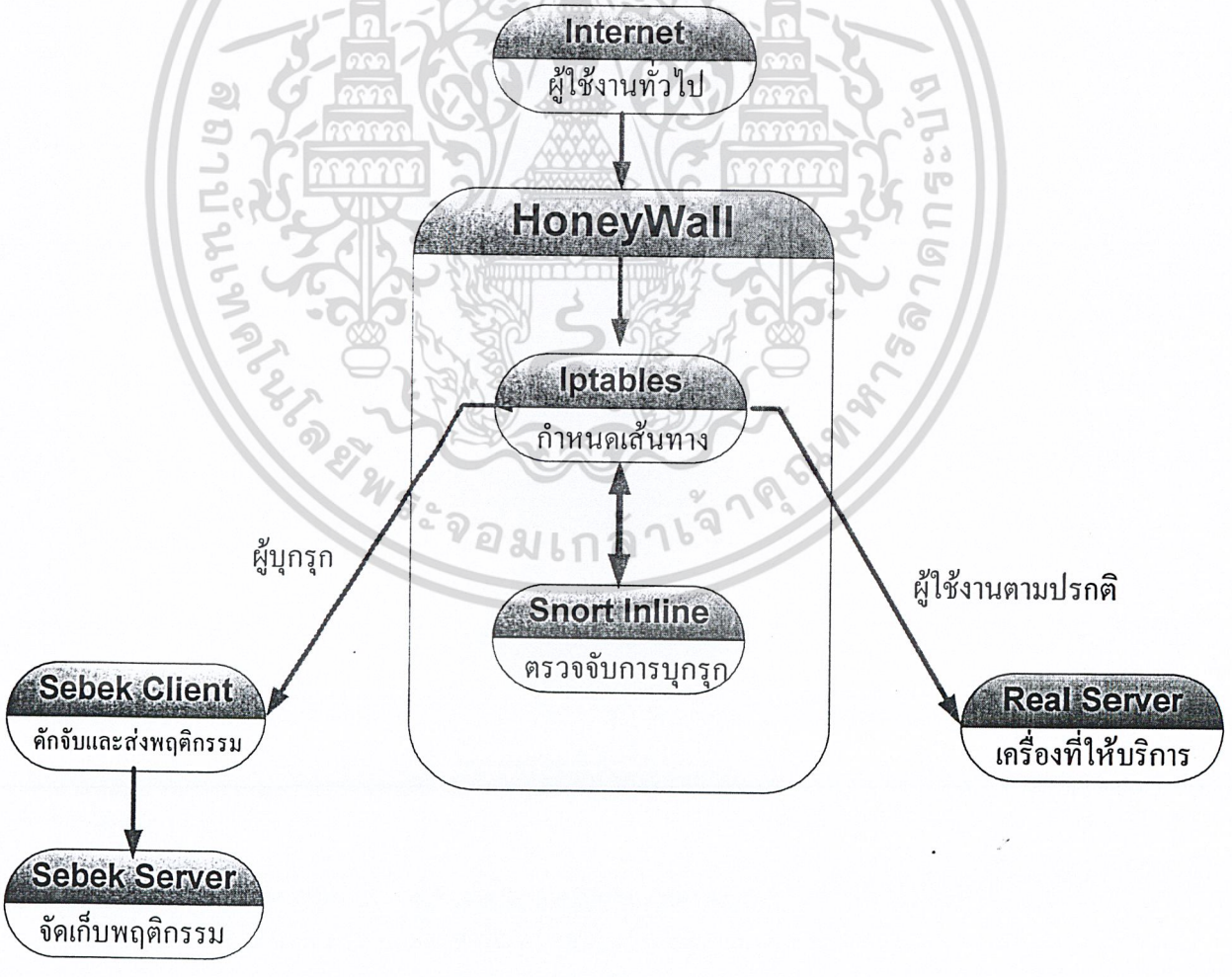
8.1 รูปแบบการเชื่อมต่อ



รูปที่ 8.1 ภาพจำลองสภาพแวดล้อมในเครือข่าย

รูปตัวอย่างด้านบนแสดงการจัดวางฮันนี่พ็อต โดยฮันนี่วอลเป็นระบบจำแนกและจัดเส้นทางการเชื่อมต่อที่เหมาะสมโดยใช้แนวคิดของระบบตรวจจับการบุกรุกทางเครือข่ายและใช้แนวคิดของ Network Address Translation (NAT) ตามลำดับ

โปรแกรมต้นแบบที่สร้างขึ้นนี้เป็นระบบรักษาความปลอดภัยตามแนวคิดของ Honeypot โดยอาศัยหลักการของไฟร์วอลและระบบตรวจจับการบุกรุกผ่านเครือข่ายเข้าร่วมจำแนก เพื่อให้ตรวจจับ บันทึกและวิเคราะห์พฤติกรรมของผู้บุกรุกได้ง่าย ระบบนี้จะล่อหลอกผู้บุกรุกให้เข้าใจผิดว่าเป็นระบบที่อ่อนแอแต่กลับเป็นระบบที่มีการเฝ้าดูพฤติกรรมทุกระยะเมื่อผู้บุกรุกเข้าสู่ระบบ ระบบจะจัดวิถีทางให้ผู้บุกรุกดำเนินไปตามความเหมาะสม ตลอดช่วงดังกล่าวก็บันทึกเหตุการณ์ ที่ผู้บุกรุกกระทำ และเมื่อผู้บุกรุกออกจากระบบ
ทุกอย่างก็จะกลับคืนสู่ภาวะเดิมเสมือนไม่มีการบุกรุกเกิดขึ้น ซึ่งผู้บุกรุกไม่อาจทราบได้เลยว่าตนเข้าไปบุกรุกระบบที่ถูกจัดไว้เฉพาะและเก็บบันทึกหลักฐานเหตุการณ์ต่างๆ เรียบร้อยแล้ว



รูปที่ 8.2 การเชื่อมต่อระหว่างส่วนต่างๆของโปรแกรมเพื่อทำการจำแนกผู้ใช้งานปรกติ กับผู้บุรุก

จากรูป Snort จะตรวจสอบแยกแยะระหว่างผู้บุกรุกกับผู้ใช้งานทั่วไปจากนั้นจะติดต่อไปยัง Iptables เพื่อให้จัดเส้นทางของการใช้งานระบบเครือข่ายอย่างเหมาะสมตามการจำแนกที่ Snort ได้ตรวจสอบมาหากเป็นผู้ใช้งานตามปรกติจะส่งไปยัง Real Server และหากเป็นผู้บุกรุกจะส่งไปยังระบบที่ได้จัดเตรียมไว้ซึ่งจะมี Sebek client ซ่อนตัวเพื่อจัดเก็บส่งการกระทำของผู้บุกรุกไปยัง Sebek server (Log Server)

8.2 ขั้นตอนการทดสอบชุดโปรแกรม

1. ผู้ใช้เข้ามาด้วยการบุกรุก
    - 1.1 snort แสดงเนื้อความที่ตรวจจับได้ตามกฎที่ตั้งไว้
    - 1.2 ส่งเนื้อความผ่านโปรแกรมสื่อสารกลาง เพื่อตั้งกฎให้ iptables
    - 1.3 โปรแกรมสื่อสารกลางตั้งกฎ iptables ตามเนื้อความที่ snort ตรวจจับได้
2. ผู้บุกรุกเข้ามายังกับดัก
    - 2.1 กรณีที่เข้ามายังช่องทางที่มีบริการจริง(service) ให้ติดต่อกับบริการจริงโดยตรง
    - 2.2 sebek client ตรวจจับพฤติกรรมและส่งไปยัง sebek server เพื่อบันทึกพฤติกรรมลงฐานข้อมูล
    - 2.3 หากช่องทางที่เข้ามาไม่มีบริการจริงรองรับ ให้portsentry รับหน้าที่ในการติดต่อกับช่องทางนั้นและบันทึกพฤติกรรมลง log โดยผ่านโปรแกรม netcat
3. ผู้ใช้เข้ามาอย่างถูกต้อง
    - 3.1 ส่งผู้ใช้เข้าติดต่อกับบริการจริงโดยตรง

8.3 ผลการทดลองมีดังต่อไปนี้

8.3.1 จากรูปเครือข่ายด้านบนให้มีการกำหนดหมายเลขประจำเครื่องดังต่อไปนี้

บทบาท	ชื่อเครื่อง	IP ที่ eth0	IP ที่ eth1	IP ที่ eth2
Attacker	Attacker	10.0.6.13	-	-
Cage	doughty	10.0.5.11	-	-
Honeywall	Honeywall	10.0.6.254	10.0.5.254	161.246.5.48
Server (ftp)	doughty	10.0.5.10	-	-
Log Server	Client1	10.0.5.12	-	-
Viewer	ISAG28	10.0.5.100	-	-

ตารางที่ 8.1 ตารางกำหนดหมายเลขประจำเครื่องในระบบเครือข่ายฮันนีพ็อต

8.3.2 ดูสถานะของการเชื่อมต่อ

ที่เครื่อง FTP Server

```
Every 1s: netstat -nat                          Wed Sep  1 23:05:57 2004

Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:21              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN
```

รูปที่ 8.3 ที่เครื่อง FTP Server

ที่เครื่อง Cage

```
Every 1s: netstat -nat                                  Tue Jan 25 12:22:09 2005

Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:139             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:21              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN
```

รูปที่ 8.4 ที่เครื่อง Cage

เพื่อให้เห็นว่ายังไม่มีการเชื่อมต่อใดๆ มาที่เครื่องเหล่านี้

8.3.3 ที่เครื่อง Honeywall

ดูตาราง NAT tables

```
honeywall:/usr/src/script# iptables -L -t nat
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
honeywall:/usr/src/script# _
```

รูปที่ 8.5 รูปการใช้คำสั่งเพื่อดูค่าตารางแนท

ดูตาราง iptables

```
honeywall:/usr/src/script# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
QUEUE      all  --  anywhere             anywhere
QUEUE      all  --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
honeywall:/usr/src/script# _
```

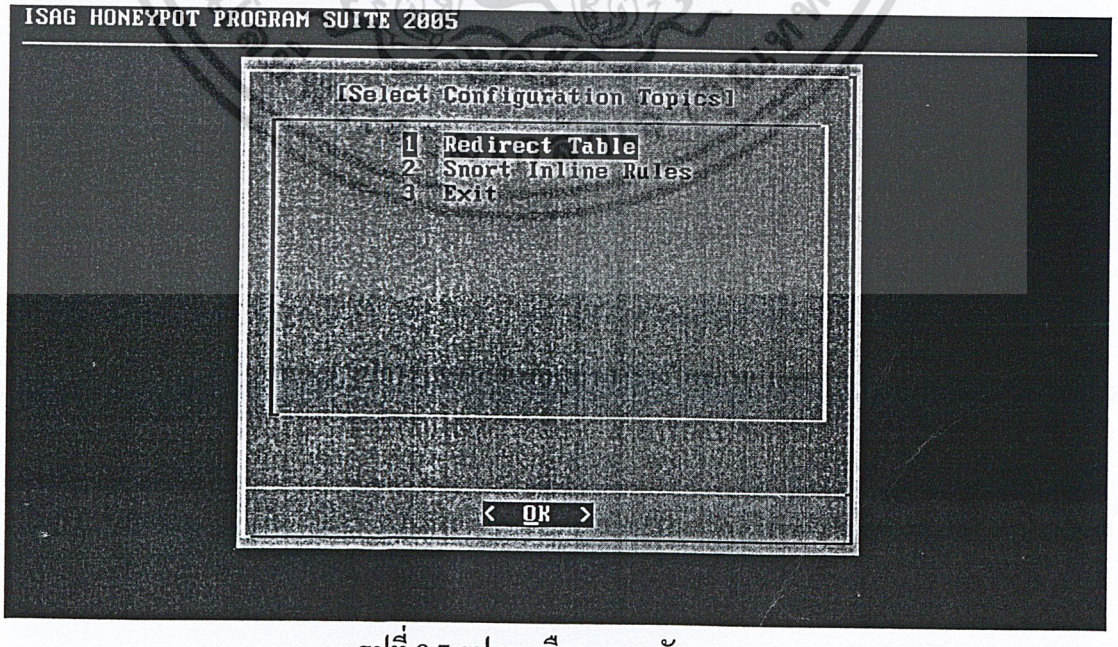
รูปที่ 8.6 รูปการใช้คำสั่งเพื่อดูค่าตารางไอพีเทเบิลส์

มีการ FORWARD ค่าไปยัง ip_queue เพื่อให้ snort_inline สามารถอ่านข้อมูลแพ็กเก็ตที่ผ่านเข้ามาในระบบเครือข่ายได้

ตั้งค่าให้ snort_inline ทำงาน

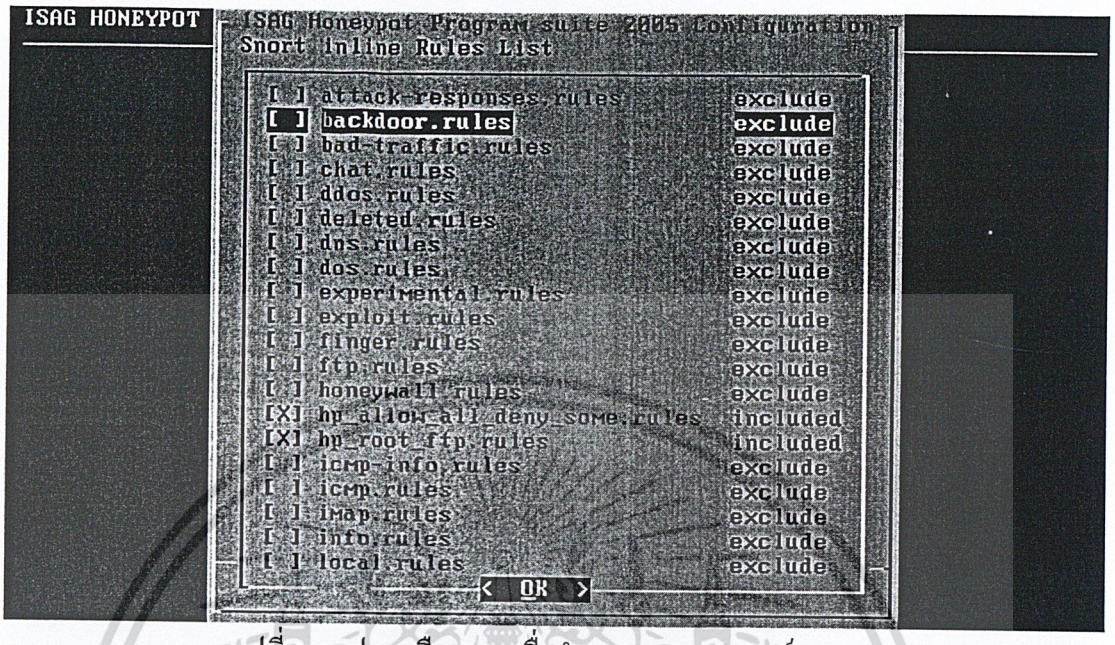
โดยเริ่มจากการตั้งค่ากฎ (Rules) เพื่อให้วิเคราะห์พฤติกรรมตามที่ต้องการ

หน้าเมนูแรก



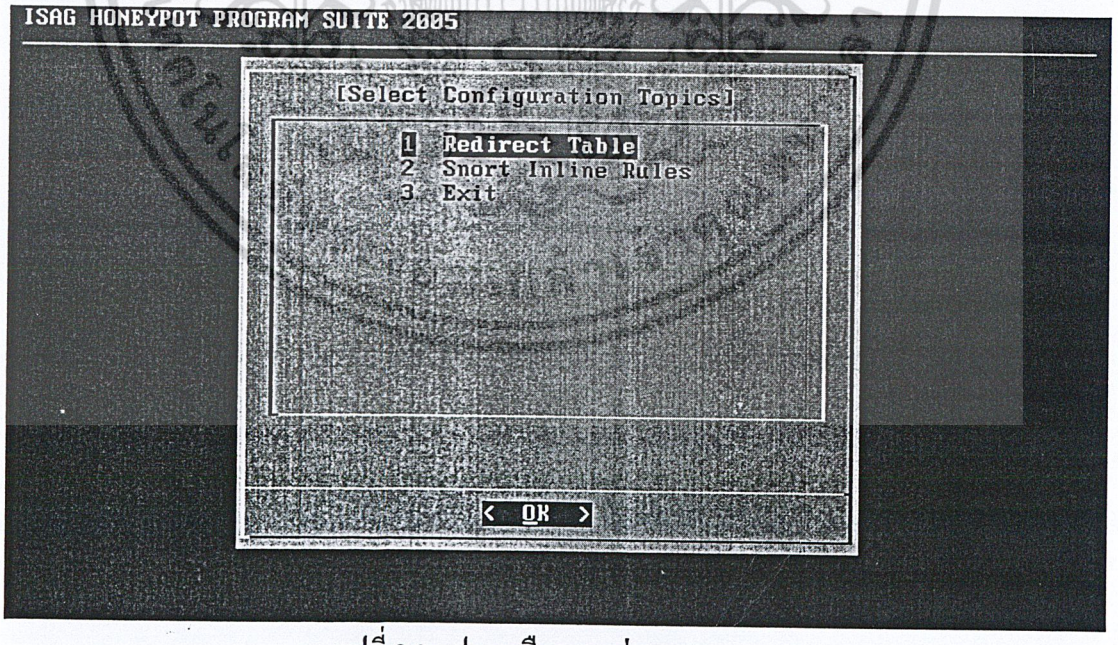
รูปที่ 8.7 รูปการเลือกเมนูหลัก

จากเมนูแรกให้เลือกหัวข้อ Snort Inline Rules เพื่อเลือกกฎตามที่ต้องการ



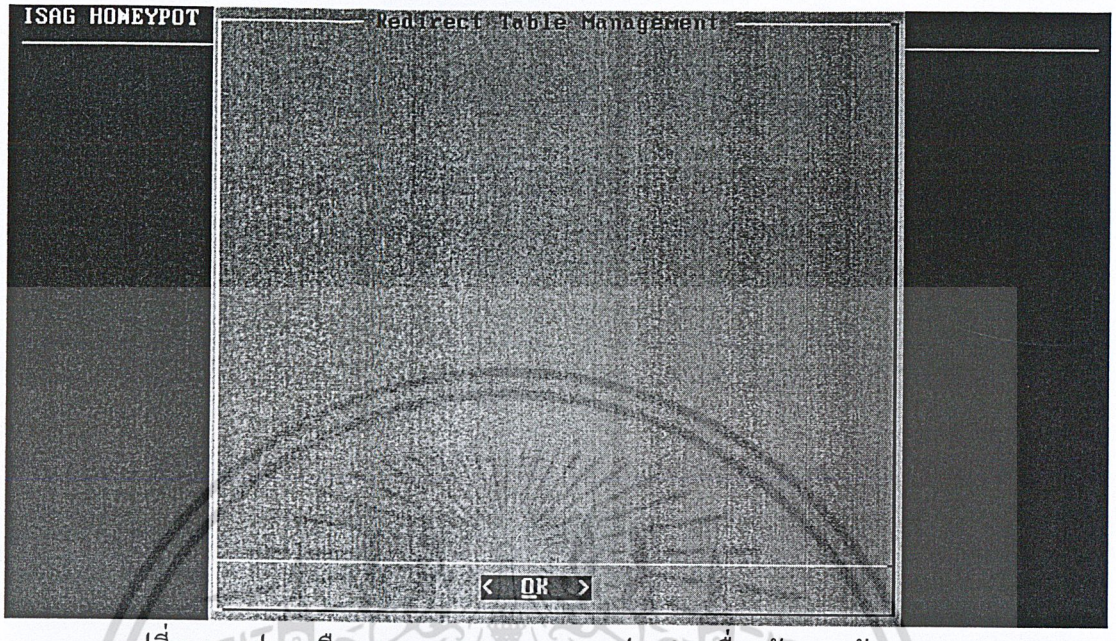
รูปที่ 8.8 รูปการเลือกเมนูเพื่อกำหนดกฎของสนอร์ท

เมื่อเลือกจากหน้าเมนูแรกหัวข้อ Redirect Table เพื่อตรวจสอบว่ามีผู้บุกรุกอยู่ใน List ของระบบแล้วหรือยังดังนี้



รูปที่ 8.9 รูปการเลือกเมนูย่อยแรก

จะไม่มีหมายเลขประจำเครื่องใดอยู่เนื่องจากว่ายังไม่มีผู้บุกรุก



รูปที่ 8.10 รูปการเลือกเมนูแสดงผลหมายเลขประจำเครื่องผู้บุกรุกผู้บุกรุก

เปิดให้ snort_inline ทำงาน

```
+----------------------[thresholding-config]----------------------------------
| memory-cap : 1048576 bytes
+----------------------[thresholding-global]----------------------------------
| none
+----------------------[thresholding-local]-----------------------------------
| none
+----------------------[suppression]------------------------------------------
| none
------------------------------------------------------------------------------
Rule application order: ->activation->dynamic->drop->sdrop->reject->alert->pass-
>log

        --== Initialization Complete ==--

********************
snort_inline-2.1.2
********************
a modification of ...

-*> Snort! <*-
Version 2.1.3 (Build 27)
By Martin Roesch (roesch@sourcefire.com, www.snort.org)
_
```

รูปที่ 8.11 รูปสั่งการให้ระบบตรวจจับผู้บุกรุกทำงาน

เปิดการทำงานของตัวผู้สื่อสารกลาง

```
honeywall:/var/log/snort/25_January_2005# ./hooksnort.sh
_
```

รูปที่ 8.12 รูปสั่งการให้ระบบสื่อสารกลางทำงาน

8.3.4 ที่เครื่อง Logserver

เปิดการรอรับฟังค่าที่ใช้เพื่อติดต่อรอการส่งข้อมูลที่เครื่องกับดักสามารถดักจับได้ เพื่อส่งเข้าดาต้าเบส

```
client1:~/sebek-server-2.1.6# ./sbk_server_start.sh
eth0: Promiscuous mode enabled.
device eth0 entered promiscuous mode
 monitoring eth0: looking for UDP dst port 2547
```

รูปที่ 8.13 รูปสั่งการให้ระบบรับค่าผลการดักจับพฤติกรรมลงฐานข้อมูล

8.3.5 ที่เครื่องกับดัก (Cage)

ใส่ตัว module ที่ใช้เพื่อแอบดักจับและส่งข้อมูลไปให้ยังเครื่องLogserver

```
doughty:/usr/src/sebek-linux-2.1.7# ./sbk_install.sh
Installing Sebek:
  honeypot.o installed successfully
  cleaner.o installed successfully
  cleaner.o removed successfully
doughty:/usr/src/sebek-linux-2.1.7# _
```

รูปที่ 8.14 รูปการใส่โมดูลและซ่อนเพื่อการจับพฤติกรรม

8.3.6 ที่เครื่องผู้บุกรุก (Attacker)

ผู้บุกรุกพยายามเข้ามาด้วยช่องทาง ftp โดยผู้ใช้เป็น root ซึ่งจากกฎที่ได้กำหนดไว้ไม่ได้อนุญาต จึงถือว่าเป็นผู้บุกรุก

```
attacker:~# ftp 10.0.5.10
Connected to 10.0.5.10.
220 doughty FTP server (Version 6.4/OpenBSD/Linux-ftpd-0.17) ready.
Name (10.0.5.10:root): root
331 Password required for root.
Password:
530 Login incorrect.
Login failed.
ftp> quit
221 Goodbye.
attacker:~#
```

รูปที่ 8.15 รูปการเข้าโจมตีจากผู้บุกรุก

8.3.7 ที่เครื่อง Honeywall (ต่อ) การส่งข้อมูลเข้าสู่ ip_queue ทำงาน

```
nf_hook: Verdict = QUEUE.
nf_hook: Verdict = QUEUE.
nf_hook: Verdict = QUEUE.
nf_hook: Verdict = QUEUE.
nf_hook: Verdict = QUEUE.
nf_hook: Verdict = QUEUE.
nf_hook: Verdict = QUEUE.
nf_hook: Verdict = QUEUE.
nf_hook: Verdict = QUEUE.
nf_hook: Verdict = QUEUE.
nf_hook: Verdict = QUEUE.
nf_hook: Verdict = QUEUE.
nf_hook: Verdict = QUEUE.
nf_hook: Verdict = QUEUE.
nf_hook: Verdict = QUEUE.
nf_hook: Verdict = QUEUE.
nf_hook: Verdict = QUEUE.
nf_hook: Verdict = QUEUE.
nf_hook: Verdict = QUEUE.
nf_hook: Verdict = QUEUE.
nf_hook: Verdict = QUEUE.
nf_hook: Verdict = QUEUE.

honeywall:/usr/src/s2i/MM#
honeywall:/usr/src/s2i/MM# _
```

รูปที่ 8.16 รูปการแสดงผลข้อมูลส่งเข้าในส่วนของ ip_queue

สามารถตรวจจับพฤติกรรมการบุกรุกได้จึงทำงาน

```
] ]    ]
 ]
MATCH TCP UDPLOOKUPIP



] ]    ]
 ]
MATCH TCP UDPLOOKUPIP



] ]    ]
 ]
MATCH TCP UDPLOOKUPIP
```

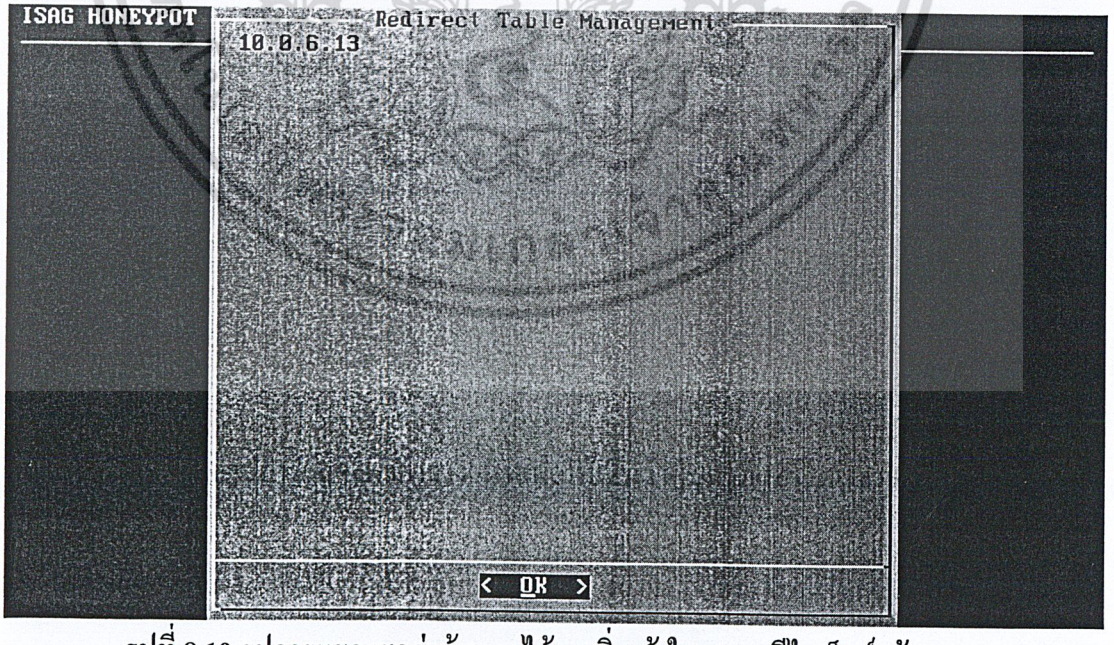
รูปที่ 8.17 รูปการแสดงผลว่าโปรแกรมผู้สื่อสารกลางกำลังแปลคำสั่งจากผลการดักจับ

เมื่อดูตาราง NAT จะพบว่าผู้บุกรุกได้ถูกจัดให้ redirect ไปยังเครื่องที่มีหมายเลขประจำเครื่องที่ 10.0.5.11 ซึ่งเป็นเครื่องของกับดัก (Cage)

```
honeywall:/usr/src/s2i/MM# iptables -L -t nat
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
DNAT       all  --  10.0.6.13           anywhere            to:10.0.5.11

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
honeywall:/usr/src/s2i/MM# _
```

รูปที่ 8.18 รูปการแสดงผลตารางแนทเพื่อให้เห็นว่าผู้บุกรุกถูกรีไดเร็กท์แล้ว

เมื่อมาดูที่ Redirect tables จะเห็นว่าหมายเลขประจำเครื่อง 10.0.6.13 ซึ่งเป็นหมายเลขประจำเครื่องของผู้บุกรุกได้เข้ามาอยู่ในตารางแล้ว



รูปที่ 8.19 รูปการแสดงผลว่าผู้บุกรุกได้ถูกเพิ่มเข้าในตารางรีไดเร็กท์แล้ว

8.3.8 ที่เครื่องผู้บุกรุก (Attacker) (ต่อ)

ผู้บุกรุกได้พยายามเข้ามาอีกครั้งในช่องทาง Secure Shell (SSH)

```
attacker:~# ssh root@10.0.5.10
root@10.0.5.10's password:
```

รูปที่ 8.20 รูปการแสดงการเข้ามาของผู้บุกรุก

เมื่อผู้บุกรุกลองตรวจสอบการติดต่อด้วยการใช้คำสั่ง netstat –nat

```
attacker:~# netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN
tcp        0      0 10.0.6.13:2050          10.0.5.10:22            ESTABLISHED
attacker:~#
```

รูปที่ 8.21 รูปการแสดงการใช้คำสั่งเพื่อตรวจสอบการเชื่อมต่อ

ผู้บุกรุกได้เข้าสู่เครื่องปลายทางที่ตนคิดว่าเป็นเครื่อง FTP server แต่จริงแล้วเข้ามายังเครื่องกับดัก (Cage)

```
attacker:~# ssh root@10.0.5.10
root@10.0.5.10's password:
Last login: Tue Jan 25 12:54:28 2005 on tty2
Linux doughty 2.4.28 #1 SMP Thu Aug 26 11:07:03 ICT 2004 i686 unknown

Most of the programs included with the Debian GNU/Linux system are
freely redistributable; the exact distribution terms for each program
are described in the individual files in /usr/share/doc/*/copyright

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Jan 25 12:54:28 2005
doughty:~#
```

รูปที่ 8.22 รูปการแสดงการเข้าสู่กับดักขณะที่ผู้บุกรุกยังไม่รู้ตัว

8.3.9 ที่เครื่องกับดัก (Cage) (ต่อ)

ที่เครื่องกับดักได้ตรวจสอบการเชื่อมต่อด้วยคำสั่ง netstat –nat

```
Every 1s: netstat -nat                          Tue Jan 25 13:03:03 2005

Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:139             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:21              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN
tcp        0      0 10.0.5.11:22            10.0.6.13:2050          ESTABLISHED
```

รูปที่ 8.23 รูปการแสดงการตรวจสอบการเชื่อมต่อทางฝั่งกับดัก

8.3.10 ที่เครื่องให้บริการจริง (Ftp Server) (ต่อ)

ที่เครื่องให้บริการจริงตรวจสอบการเชื่อมต่อโดยใช้คำสั่ง netstat –nat จะเห็นว่าตนไม่ได้เชื่อมต่อเข้ากับเครื่องของผู้บุกรุกแต่อย่างใด

```
Every 1s: netstat -nat                                  Thu Sep  2 00:36:14 2004

Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:21              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN
```

รูปที่ 8.24 รูปการแสดงการตรวจสอบการเชื่อมต่อทางฝั่งเครื่องให้บริการจริง

8.3.11 ที่เครื่องจัดเก็บข้อมูล Logserver (ต่อ)

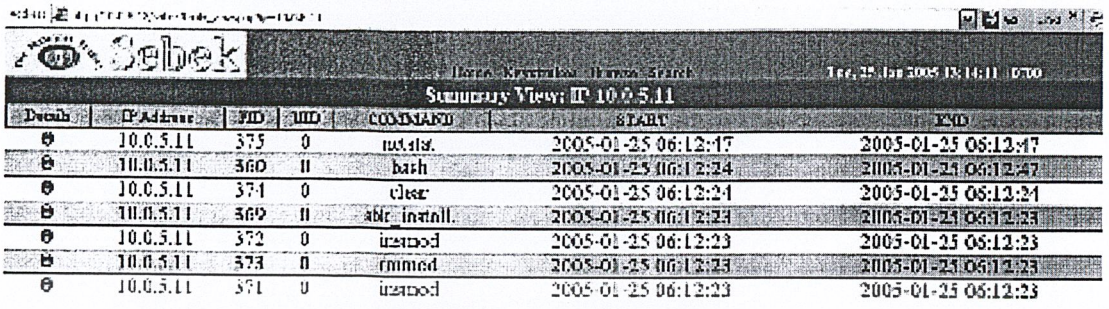
จะได้รับการกระทำที่ผู้บุกรุกได้เข้ากระทำในกับดัก

```
 monitoring eth0: looking for UDP dst port 2547

warning RX 0   Lost 0


warning RX 0   Lost 0

  Lost 15  RXed 38 frames
  Lost 16  RXed 40 frames
  Lost 20  RXed 41 frames
  Lost 87  RXed 42 frames

warning RX 0   Lost 0

  Lost 90  RXed 92 frames

warning RX 0   Lost 0

  Lost 91  RXed 2759 frames
  Lost 94  RXed 2760 frames
  Lost 97  RXed 2761 frames
  Lost 190  RXed 2762 frames

warning RX 0   Lost 0

  Lost 191  RXed 2999 frames
  Lost 194  RXed 3000 frames
  Lost 197  RXed 3001 frames
  Lost 258  RXed 3002 frames
```

รูปที่ 8.25 รูปการแสดงการรับค่าที่กับดักตรวจจับได้ลงสู่ฐานข้อมูล

8.3.12 ที่เครื่อง Viewer

เปิดดูผ่านทางเวบอินเตอร์เฟส



รูปที่ 8.26 รูปการแสดงผลข้อมูลที่ดักจับได้จากฐานข้อมูล

จะเห็นว่าตัวกับดักเองได้มีการกระทำดังนี้

- insmod เป็นการ insert module ที่ใช้เพื่อการเฝ้าดูพฤติกรรมลงไป คำสั่งที่เป็นการ insert module มีไปจนถึง sbk_install
- ผู้บุกรุกได้ใช้คำสั่ง clear และใช้คำสั่ง netstat

บทที่ 9
สรุปผลและวิจารณ์

9.1 ปัญหาและอุปสรรคในการพัฒนา

ในส่วนของการพัฒนาระบบจำแนกและบันทึกพฤติกรรมผู้บุกรุกนั้นได้พบปัญหาและอุปสรรคต่างๆ หลายประการดังนี้

- การจับความสามารถต่างๆ ของเครื่องมือที่ต้องใช้เพื่อนำมาประกอบกันเป็นแนวคิดใหม่ เนื่องจากความที่ไม่เคยมีใครได้ทำมาก่อนจึงไม่สามารถหาแนวคิดร่วมหรือตัวอย่างได้จากที่ใดๆ จึงใช้เวลาในการศึกษาส่วนของแนวคิดใหม่นี้อยู่ค่อนข้างนาน
- เมื่อจับใจความและสรุปออกมาเป็นแนวคิดใหม่ได้แล้วแต่ทว่าต้องใช้การเขียนภาษาโปรแกรมที่เป็นภาษาของคอมไพล์เลอร์ที่ชื่อว่า เฟลคซ์ (FLEX) เพื่อใช้เป็นผู้สื่อสารกลางให้ทำหน้าที่รับค่าจากตัวระบบตรวจสอบพฤติกรรมผู้บุกรุก (Snort) และจัดการบอกว่าไฟร์วอล (iptables)ควรทำหน้าที่หรือคำสั่งใดที่ควรจะบอกลงไปในกฎของไฟร์วอล
- ในส่วนของการดึงค่าที่ตัวระบบตรวจสอบพฤติกรรมผู้บุกรุกนั้นไม่สามารถส่งค่าที่ตัวมันได้เวอร์โบส (Verbose) ออกมาได้เนื่องจากว่ายังเป็นการทำงานของตัวระบบตรวจสอบพฤติกรรมผู้บุกรุกจึงยังไม่สามารถให้คำสั่งใดทำงานด้วยกันได้
- ด้วยเวลาที่จำกัดจึงยังไม่สามารถทำส่วนที่สร้างให้กับดักนั้นมีความแนบเนียนได้

9.2 แนวทางการพัฒนาต่อในอนาคต

เนื่องจากเวลาที่มีจำกัด ยันนี้พ็อตในปีนี้จึงมีความก้าวหน้าได้เพียงแต่สามารถจำแนกแยกแยะผู้บุกรุกได้เท่านั้น แต่ก็ได้เสริมในส่วนของการดูแลกับดักไว้บ้างเพื่อใช้เวลาอย่างคุ้มค่า แต่แล้วก็ยังขาดอยู่หลายส่วนที่มีความจำเป็นมีดังนี้

- ในส่วนของเครื่องที่เป็นกับดักนั้นต้องการให้มีความแนบเนียนเพื่อไม่ให้ผู้บุกรุกรู้ได้ว่าถูกหลอกให้เข้ามาในเครื่องที่ไม่ใช่เป้าหมายของตน จึงต้องทำให้แนบเนียนดังนี้
    - เมื่อผู้บุกรุกใช้คำสั่งที่เป็นการแสดงตัวตนของเครื่อง จะต้องไม่สามารถรู้ได้ว่าตนเองมาอยู่บนเครื่องที่ไม่ใช่ แนะนำให้แก้ไขที่ kernel source หรือ สร้าง script หรือแนวคิดที่ดีกว่า

- ในส่วนของการส่งเข้าไปยังกับดักนั้นให้มีได้มากกว่าหนึ่งกับดักเนื่องจากว่าที่ได้ทำเป็นเพียงส่วนเริ่มหากต้องการให้เป็นหลายเครื่องก็ทำซ้ำโดยการวางคนละห้องและตั้ง

ค่าหมายเลขเครื่องและกฎของระบบตรวจจับพฤติกรรมผู้บุกรุกตามต้องการ แต่หากจะทำแล้วมีวิธีการดังนี้

- เข้าแก้ไข source code ของโปรแกรมผู้แปลงสารและให้มีส่วนที่เป็นตัวกำหนดว่ามีกี่กับดักมีหมายเลขประจำเครื่องหรือไอพีอะไรบ้าง ห้ามทำเป็นอาร์เรย์ธรรมดา ให้ทำเป็นโครงสร้างอาร์เรย์หรือลิงค์ลิสท์ได้ เพื่อให้รวดเร็วในการทำงาน มีความปลอดภัย และดีบักเพื่อป้องกันการเกิดเซกเมนเตชั่นฟอลท์ได้

- ในส่วนการดูแลเครื่องกับดักว่าใกล้ถึงจุดที่ยากต่อการควบคุมแล้วหรือยังให้มีการทำงานที่ควรมีอย่างน้อยดังนี้
  - เมื่อกับดักเองถึงจุดต้องจัดเก็บให้จัดเก็บตัวอิมเมจของตัวกับดักและนำอิมเมจที่สมบูรณ์มาวางแทนอย่างอัตโนมัติ หรือเลือกได้ ฉะนั้นต้องมีไฟล์ที่เป็นตัวกำหนดหรือคอนฟิกกูเรชันไฟล์
  - จัดเก็บเนื้อความที่สามารถดักจับได้ที่อยู่ในฐานข้อมูลมาจัดเก็บไว้ให้เหมาะสม
  - เมื่อต้องการนำอิมเมจที่บอบช้ำนั้นมาศึกษาต้องสามารถทำได้โดยง่ายเช่นมี script ที่ทำงานให้อย่างอัตโนมัติ
  - เมื่อนำกับดักที่บอบช้ำนั้นมาทำงานต้องควบคุมการกระทำที่เป็นลูกโซ่อยู่ได้

- หากค้นหาแนวคิดที่ดีกว่าได้ ควรทำต่อไป

บรรณานุกรม

แหล่งอ้างอิง

1. หนังสือ Honeypots Tracking Hackers ผู้แต่ง Lance Spitzner แปลโดย Marcus J. Ranum ISBN: 0-321-10895-7
2. WHITE PAPER ของบริษัท Symantec™ ดังนี้
   - 2.1 Intrusion Detection System : Symantec™ ManHunt™ (Reducind the risk of Compromise)
   - 2.2 Vulnerability Assessment Guide
   - 2.3 Blended Threats: Case Study and Countermeasures
   - 2.4 Top Management's Perspective on Security
3. ข้อมูลที่หาได้จากเวปไซด์
   - 3.1 http://www.tracking-hackers.com/papers/honeypots.html
   - 3.2 http://www.honeypots.com/about.html
   - 3.3 http://www.honeynet.org/alliance/requirements.html
   - 3.4 http://www.honeynet.org/papers/gen2/
   - 3.5 http://www.all.net/dtk/index.html
   - 3.6 http://www.citi.umich.edu/u/provos/honeyd/
   - 3.7http://www.winnetmag.com/Article/ArticleID/39428/39428.html
   - 3.8 http://www.honeynet.org/papers/honeynet/
   - 3.9 http://project.honeynet.org/papers/vmware/
   - 3.10 http://www.honeynet.org/papers/stats/
   - 3.11 http://www.honeynet.org/papers/virtual/
   - 3.12 http://labrea.sourceforge.net/labrea-info.html
   - 3.13 http://www.securityprofiling.com/honeyd/honeyd.shtml
   - 3.14 http://project.honeynet.org/book/
   - 3.15 http://www.honeywall.org/honeywall-describe.html
4. ข้อมูลที่อ่านจากปริญญานิพนธ์
   - 4.1 การป้องกันเวบโดยใช้ไอพีเทเบิลส์ (Web prevention using iptables)
   - 4.2 ระบบตรวจจับผู้บุกรุกทางคอมพิวเตอร์ (Intrusion Detection System)