การออกแบบและพัฒนาระบบป้องกันการบุกรุกเครือข่ายโดยอัตโนมัติโดยใช้เครือข่ายแอ็กทีฟ เพื่อรับมือกับการโจมตีจากเครือข่ายภายในที่ถูกผู้บุกรุกยึดครอง

นาย มหคม อร่ามเสรีวงศ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2548
ISBN 974-53-2529-5

A DESIGN AND DEVELOPMENT OF AN AUTOMATIC NETWORK-INTRUSION DEFENSE SYSTEM USING ACTIVE NETWORK FOR HANDLING ATTACKS FROM COMPROMISED INTERNAL NETWORKS

Mr. Mahacom Aramsereewong

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science Program in Computer Science

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2005

ISBN 974-53-2529-5

หัวข้อวิทยานิพนธ์	การออกแบบและพัฒนาระบบป้องกันการบุกรุกเครือข่ายโดยอัตโนมัติโดยใช้เครือข่ายแอ็กทีฟ เพื่อรับมือกับการโจมตีจากเครือข่ายภายในที่ถูกผู้บุกรุกยึดครอง
โดย	นาย มหคม อร่ามเสรีวงศ์
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษา	อาจารย์ ดร.เฉลิมเอก อินทนากรวิวัฒน์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.ดิเรก ลาวัณย์ศิริ)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(อาจารย์ จารุมาตร ปิ่นทอง)

..... อาจารย์ที่ปรึกษา
(อาจารย์ ดร.เฉลิมเอก อินทนากรวิวัฒน์)

..... กรรมการ
(อาจารย์ ดร.อรรถวิทย์ สุดแสง)

..... กรรมการ
(อาจารย์ ธงชัย โรจน์กังสดาล)

มหคม อร่ามเสรีวงศ์ : การออกแบบและพัฒนาระบบป้องกันการบุกรุกเครือข่ายโดยอัตโนมัติโดยใช้เครือข่ายแอ็กทีฟ เพื่อรับมือกับการโจมตีจากเครือข่ายภายในที่ถูกผู้บุกรุกยึดครอง. (A DESIGN AND DEVELOPMENT OF AN AUTOMATIC NETWORK-INTRUSION DEFENSE SYSTEM USING ACTIVE NETWORK FOR HANDLING ATTACKS FROM COMPROMISED INTERNAL NETWORKS) อ. ที่ปรึกษา : อ.ดร.เฉลิมเอก อินทนากรวิวัฒน์, 151 หน้า. ISBN 974-53-2529-5.

วิทยานิพนธ์ฉบับนี้ได้พัฒนาระบบป้องกันการบุกรุกเครือข่าย ที่มีอุปกรณ์ป้องกันการบุกรุกเครือข่ายหลายตัวทำงานร่วมกันโดยใช้เครือข่ายแอ็กทีฟในการพัฒนา ทำให้สามารถปรับเปลี่ยนโปรแกรมควบคุมการทำงานของอุปกรณ์ได้ตลอดเวลา ระบบนี้มีความสามารถในการตรวจสอบ ค้นหาที่มา และหยุดการบุกรุกให้อยู่เฉพาะในเครือข่ายย่อยได้ งานวิจัยนี้ยังได้นำเสนอวิธีตรวจสอบการบุกรุกแบบกระจาย โดยที่อุปกรณ์หลายตัวจะช่วยกันตรวจสอบการบุกรุก เป็นการกระจายภาระในการตรวจสอบ อีกทั้งอุปกรณ์แต่ละตัวไม่ต้องตรวจสอบการบุกรุกทุกรูปแบบที่มีอยู่ แต่ตรวจสอบเฉพาะการบุกรุกที่มุ่งโจมตีเครื่องที่อยู่ในเครือข่ายย่อยที่ดูแลเท่านั้น จากผลการทดสอบพบว่าวิธีการนี้ให้อัตราการส่งผ่านข้อมูลดีขึ้น เมื่อเทียบกับการใช้วิธีตรวจสอบแบบทั่วไป

ภาควิชา..... วิศวกรรมคอมพิวเตอร์.....ลายมือชื่อนิสิต.....

สาขาวิชา.....วิทยาศาสตร์คอมพิวเตอร์....ลายมือชื่ออาจารย์ที่ปรึกษา.....

ปีการศึกษา ...2548.....

45714465 : MAJOR COMPUTER SCIENCE
KEY WORD: ACTIVE NETWORKS / INTRUSION PREVENTION SYSTEM / DISTRIBUTED DETECTION

MAHACOM ARAMSEREEWONG : A DESIGN AND DEVELOPMENT OF AN AUTOMATIC NETWORK-INTRUSION DEFENSE SYSTEM USING ACTIVE NETWORK FOR HANDLING ATTACKS FROM COMPROMISED INTERNAL NETWORKS. THESIS ADVISOR : CHALERMEK INTANAGONWIWAT, Ph.D., 151 pp. ISBN 974-53-2529-5.

This thesis presents a network-intrusion prevention system that is a collaboration among multiple intrusion prevention devices. By using Active Networks, a control program can be dynamically loaded into any intrusion prevention devices. This system can detect, traceback, and stop intruders at theirs sub-networks. In addition, we propose “distributed detection” for multiple detectors to co-operate in detecting intrusion and to share the detection load. Each device does not have to detect all intrusion signatures, but only the signatures that are known to be feasible attacks on the host platform in its sub-network. Our experimental results indicate that the throughput of this new detection method is more than that of the general approach.

Department..... Computer Engineering.....Student's signature.....
Field of study.....Computer Science.....Advisor's signature.....
Academic year ...2005.....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยความช่วยเหลืออย่างดียิ่งของอาจารย์ที่ปรึกษาวิทยานิพนธ์ อาจารย์ ดร.เฉลิมเอก อินทนากรวิวัฒน์ ซึ่งท่านได้แนะนำ และให้ข้อคิดเห็นต่าง ๆ ในการวิจัยด้วยดีมาตลอด รวมทั้งตรวจแก้วิทยานิพนธ์ฉบับนี้อย่างละเอียด ผู้วิจัยขอกราบขอบพระคุณในความกรุณาจากอาจารย์เป็นอย่างสูง รวมถึงอาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ทุกท่าน ที่ประสิทธิประสาทวิชาความรู้ให้ผู้วิจัย

ขอขอบคุณ อาจารย์ จารุมาตร ปิ่นทอง ประธานกรรมการสอบวิทยานิพนธ์ รวมถึงกรรมการสอบอีกสองท่านได้แก่ อาจารย์ ดร.อรรถวิทย์ สุดแสง และ อาจารย์ ธงชัย โรจน์กังสดาล ที่ได้ช่วยกรุณาสละเวลามาช่วยตรวจสอบ ดำเนินการสอบ และแก้ไขวิทยานิพนธ์ฉบับนี้ให้สมบูรณ์

ขอขอบคุณเหล่าเพื่อนร่วมรุ่น และเพื่อนร่วมงานทุกคน ที่คอยถามไถ่ถึงความคืบหน้าในการทำงานวิจัย และให้ความช่วยเหลือในด้านต่าง ๆ เป็นอย่างดี

ท้ายนี้ผู้วิจัยขอกราบขอบพระคุณบิดา มารดา และทุกคนในครอบครัวที่คอยสนับสนุนในด้านการเรียน และให้กำลังใจแก่ผู้วิจัยเสมอมาจนสำเร็จการศึกษา

สารบัญ

หน้า

สารบัญภาพ

หน้า

สารบัญตาราง

หน้า

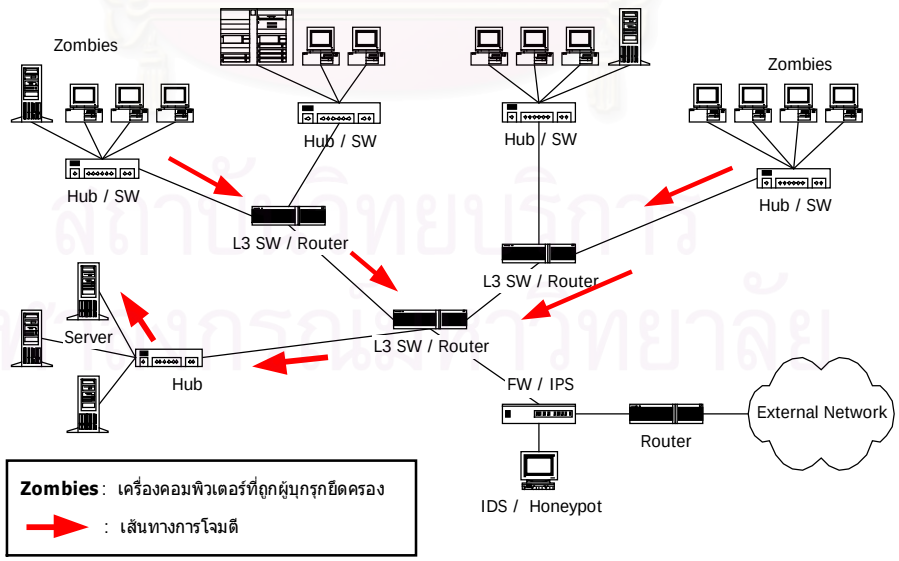
บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัญหาการบุกรุกและโจมตีเครือข่ายคอมพิวเตอร์นั้นเป็นปัญหาสำคัญที่ต้องการการแก้ไขอย่างเร่งด่วน เนื่องจากรูปแบบการดำเนินชีวิตในปัจจุบันนั้น จำเป็นต้องพึ่งพาอาศัยเครือข่ายคอมพิวเตอร์เป็นอย่างมาก เพื่ออำนวยความสะดวกในการติดต่อสื่อสาร ทั้งในด้านการดำเนินธุรกิจ การทำงาน การใช้ชีวิตประจำวัน หรือเพื่อความบันเทิงในรูปแบบต่าง ๆ และมีแนวโน้มว่าจะยิ่งมีความจำเป็นเพิ่มมากขึ้นในอนาคต ซึ่งดูได้จากจำนวนผู้ใช้งานเครือข่ายที่เพิ่มมากขึ้นตลอดเวลา การพัฒนารูปแบบการให้บริการและโปรแกรมประยุกต์ประเภทใหม่ ๆ เกิดขึ้นอย่างสม่ำเสมอ เทคโนโลยีทางด้านเครือข่ายได้รับการพัฒนาให้มีประสิทธิภาพและความเร็วสูงขึ้นเรื่อย ๆ รวมทั้งการใช้งานเครือข่ายหรือการเข้าถึงเครือข่ายก็ได้รับการพัฒนาให้มีรูปแบบที่หลายหลายขึ้น แต่นั่นก็เท่ากับเปิดโอกาสให้ผู้บุกรุกมีช่องทางในการโจมตีเครือข่ายได้มากขึ้น การโจมตีสามารถทำได้เร็วขึ้น และความเสียหายที่เกิดจากการโจมตีก็รุนแรงขึ้น อีกทั้งยังทำให้การตรวจสอบและป้องกันการบุกรุกโจมตีทำได้ยากขึ้นอีกด้วย เพราะเมื่อปริมาณการใช้งานเครือข่ายสูงขึ้น ทำให้ข้อมูลที่จะต้องตรวจสอบก็มีเพิ่มมากขึ้นด้วย และที่สำคัญก็คือ ไม่ใช่แค่เพียงระบบเครือข่ายเท่านั้นที่ได้รับการพัฒนา แต่วิธีการหรือรูปแบบของการโจมตีเครือข่ายก็ได้รับการพัฒนาอยู่เสมอด้วยเช่นกัน ตัวอย่างเช่นในปัจจุบันรูปแบบการโจมตีเครือข่ายจะอาศัยวิธีการส่งโปรแกรมที่ผู้บุกรุกสร้างขึ้นมา เช่น ไวรัส (Virus) หรือ เวิร์ม (Worm) ไปติดตั้งไว้ที่เครื่องคอมพิวเตอร์จำนวนมาก โดยใช้วิธีการแพร่กระจายโปรแกรมดังกล่าวผ่านจดหมายอิเล็กทรอนิกส์เป็นต้น ซึ่งเมื่อเครื่องคอมพิวเตอร์ถูกติดตั้งโปรแกรมดังกล่าวแล้ว ผู้บุกรุกก็จะสามารถควบคุมเครื่องเหล่านั้นเพื่อใช้เป็นเครื่องมือในการร่วมกันโจมตีเครื่องเป้าหมายในภายหลังได้ นอกจากนี้การโจมตีในปัจจุบันก็ไม่ได้ยุ่งยากเหมือนก่อน เพราะมีการสร้างโปรแกรมขึ้นมาเพื่อใช้เป็นเครื่องมือสำหรับโจมตีเครือข่ายได้โดยอัตโนมัติและมีเผยแพร่อยู่ทั่วไปในอินเตอร์เน็ต ทำให้แม้แต่ผู้ใช้ทั่วไปที่ไม่ใช่ผู้เชี่ยวชาญ ก็สามารถนำเครื่องมือเหล่านี้มาใช้ในการโจมตี สร้างความเสียหายให้กับผู้อื่นได้ไม่ยาก

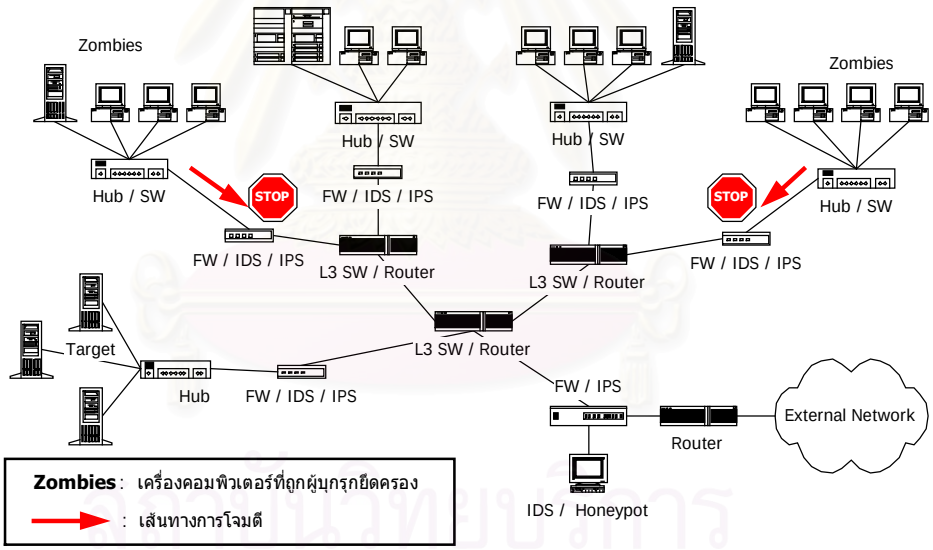
ในเครือข่ายคอมพิวเตอร์ทั่ว ๆ ไป จะใช้วิธีติดตั้งอุปกรณ์รักษาความปลอดภัยไว้ในเครือข่าย ซึ่งก็จะมีให้เลือกใช้งานอยู่หลายประเภท เช่น Firewall, Intrusion Detection System (IDS) หรือ Intrusion Prevention System (IPS) เป็นต้น ซึ่งถ้ามีการเลือกติดตั้งอุปกรณ์ได้ในตำแหน่งที่เหมาะสม และปรับค่าการทำงานต่าง ๆ ได้อย่างถูกต้อง รวมทั้งมีการบำรุงรักษาและปรับปรุงความสามารถของอุปกรณ์เหล่านี้อยู่อย่างสม่ำเสมอแล้ว ก็พอจะมั่นใจได้ว่าอุปกรณ์เหล่านี้จะสามารถป้องกันการบุกรุกโจมตีเครือข่ายได้ดีในระดับที่น่าพอใจ ซึ่งโดยทั่วไปแล้วการติดตั้งโดย

ส่วนใหญ่ก็จะติดตั้งไว้ที่ทางเข้าออกระหว่างเครือข่ายภายในกับเครือข่ายภายนอก แต่สำหรับเครือข่ายคอมพิวเตอร์ขนาดใหญ่ เช่น ในมหาวิทยาลัย ในหน่วยงานราชการ หรือในบริษัทขนาดใหญ่ เป็นต้น ซึ่งมักจะมีเครื่องที่คอยให้บริการต่าง ๆ จำนวนมาก มีเครือข่ายหลัก (Core-Network, Backbone) ที่ถูกเชื่อมต่อด้วยเครือข่ายย่อย (Sub-Network) จำนวนมาก รวมทั้งมีจำนวนเครื่องผู้ใช้งานจำนวนมาก และด้วยปริมาณการใช้งานเครือข่ายสูงทำให้ข้อมูลที่วิ่งอยู่ภายในเครือข่ายและวิ่งเข้าออกระหว่างเครือข่ายมีปริมาณมาก ซึ่งหากนำอุปกรณ์ตรวจสอบป้องกันนี้ไปติดตั้งไว้ที่ทางเข้าออกระหว่างเครือข่ายภายในกับภายนอกแล้ว การที่จะทำให้อุปกรณ์เหล่านี้มาใช้ป้องกันให้ได้ประสิทธิภาพทำได้ค่อนข้างยาก เพราะอุปกรณ์ตรวจสอบป้องกันนี้จะต้องทำงานกับข้อมูลจำนวนมาก และโดยเฉพาะอย่างยิ่งถ้าเครื่องที่ใช้งานในเครือข่ายมีการใช้ระบบปฏิบัติการหลายชนิด มีเครื่องที่เปิดบริการให้ใช้งานหลายแบบแล้ว อุปกรณ์ตรวจสอบก็ต้องทำการตรวจสอบรูปแบบของการบุกรุกด้วยทุกรูปแบบที่มีอยู่ เพื่อให้ครอบคลุมเครื่องเหล่านั้นทั้งหมด ทำให้ความเสี่ยงที่อุปกรณ์ตรวจสอบจะทำงานผิดพลาดมีค่อนข้างสูง อีกทั้งการตรวจสอบก็จะไม่สามารถทำได้ครอบคลุมทุกจุดในเครือข่าย เพราะถ้าติดตั้งอุปกรณ์ป้องกันและตรวจสอบไว้ที่ทางเข้าออก ก็จะสามารถป้องกันได้เฉพาะการบุกรุกและโจมตีระหว่างเครือข่ายภายในกับภายนอกเท่านั้น ในขณะที่วิธีการโจมตีที่พบในปัจจุบันสามารถใช้เครื่องที่อยู่ภายในเครือข่ายเป็นฐานในการโจมตีได้ โดยเฉพาะในเครือข่ายขนาดใหญ่ที่มีเครื่องคอมพิวเตอร์จำนวนมากนั้น ผู้บุกรุกสามารถอาศัยเครื่องที่อยู่ในเครือข่ายดังกล่าวเป็นเครื่องมือหรือเป็นฐานในการโจมตีได้เป็นอย่างดี ดังรูปที่ 1



รูปที่ 1 เครือข่ายคอมพิวเตอร์ที่ติดตั้งอุปกรณ์ป้องกันผู้บุกรุกไว้ที่จุดเชื่อมต่อระหว่างเครือข่ายภายในกับเครือข่ายภายนอก ซึ่งไม่สามารถป้องกันการโจมตีที่มาจากเครื่องภายในเครือข่ายได้

ปัญหาดังกล่าวสามารถแก้ไขได้ด้วยการเปลี่ยนรูปแบบการติดตั้งอุปกรณ์ตรวจสอบป้องกันการบุกรุก จากเดิมที่ติดตั้งไว้เพียงจุดเดียวที่ทางเข้าออกระหว่างเครือข่ายภายในกับเครือข่ายภายนอก เปลี่ยนไปเป็นการติดตั้งไว้หลาย ๆ จุดในเครือข่าย เช่น ติดตั้งไว้ในทุก ๆ จุดที่เป็น "บริเวณขอบของเครือข่ายหลัก" (บริเวณที่เป็นจุดเชื่อมต่อระหว่างเครือข่ายหลักกับเครือข่ายย่อย) ซึ่งการติดตั้งในรูปแบบนี้จะมีข้อดีตรงที่อุปกรณ์ตรวจสอบป้องกัน จะรับหน้าที่ตรวจสอบเฉพาะข้อมูลที่วิ่งผ่านระหว่างเครือข่ายย่อยกับเครือข่ายหลักเท่านั้น ซึ่งนอกจากจะช่วยกันแบ่งเบาภาระของอุปกรณ์ตรวจสอบแต่ละตัวแล้ว ยังสามารถปรับค่าการตรวจสอบของอุปกรณ์ให้เหมาะสมกับเฉพาะเครือข่ายย่อยนั้น ๆ ได้อีกด้วย เช่นในเครือข่ายย่อยที่มีแต่เครื่องที่ติดตั้งระบบปฏิบัติการยูนิกซ์ ก็จะปรับค่าการทำงานของอุปกรณ์ให้ตรวจสอบเฉพาะการบุกรุกที่โจมตีเฉพาะระบบปฏิบัติการยูนิกซ์เท่านั้น และเมื่อได้มีการติดตั้งอุปกรณ์ตรวจสอบป้องกันไว้ที่บริเวณขอบของเครือข่ายหลักแล้ว หากอุปกรณ์ตรวจสอบที่ติดตั้งไว้ตรวจพบการบุกรุก ก็จะสามารถจัดการกับการบุกรุกนั้น เพื่อสกัดกั้นไม่ให้ผ่านไปได้ ดังรูปที่ 2



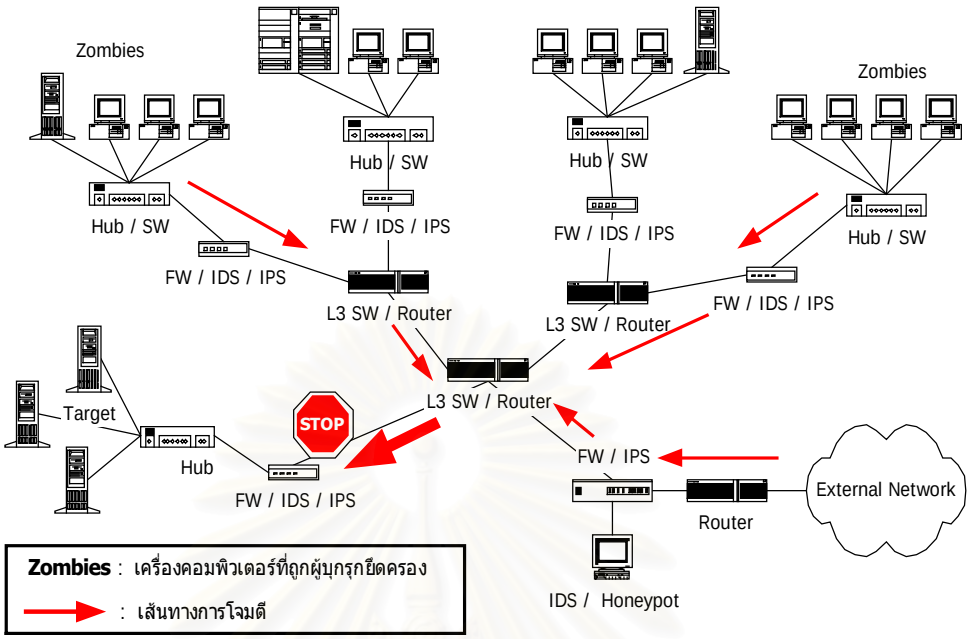
รูปที่ 2 การติดตั้งอุปกรณ์ป้องกันการบุกรุกในเครือข่ายย่อย จะช่วยป้องกันการโจมตีจากเครือข่ายภายในได้

แต่การติดตั้งอุปกรณ์ตรวจสอบป้องกันจำนวนมาก ๆ ในเครือข่ายนั้น นอกจากค่าใช้จ่ายที่สูงขึ้นแล้ว ยังมีปัญหาที่ตามมาอีกก็คือ การบำรุงรักษา การปรับปรุงอุปกรณ์ให้ทันสมัย และการปรับค่าการทำงานจะทำได้ลำบาก เนื่องจากอุปกรณ์ตรวจสอบมีจำนวนมาก อีกทั้งยังถูกติดตั้งกระจายไปในส่วนต่าง ๆ ของเครือข่าย ซึ่งอาจแก้ปัญหาได้โดยการกำหนดผู้ดูแลเครือข่ายย่อยขึ้นเพื่อรับผิดชอบ แต่ก็จะก่อให้เกิดปัญหาตามมาอีกเช่นกัน เช่น อุปกรณ์ไม่ได้รับการปรับปรุงให้ทัน

สมัยหากผู้ดูแลเครือข่ายไม่ได้ติดตามข่าวสาร หรือการปรับค่าอุปกรณ์ไม่เหมาะสมเพราะขาดความรู้ความเข้าใจที่ถูกต้อง และความเสี่ยงในเรื่องการรักษาความปลอดภัยเป็นต้น

จากความไม่สะดวกดังกล่าว ผู้ผลิตอุปกรณ์ป้องกันการบุกรุกส่วนใหญ่จึงได้ปรับเปลี่ยนรูปแบบของอุปกรณ์ตรวจสอบให้มีลักษณะเป็นอุปกรณ์เซ็นเซอร์ ติดตั้งไว้ตามจุดต่าง ๆ ในเครือข่าย ซึ่งมีความสามารถในการตรวจสอบหลากหลายรูปแบบ เช่น การตรวจสอบสัญลักษณ์ของแพ็กเก็ต (Packet Signature Detection) การตรวจสอบความผิดปกติของการใช้งานเครือข่าย (Anomaly Detection) การตรวจสอบปริมาณการใช้งานเครือข่าย (Bandwidth Detection) เป็นต้น ทั้งนี้ก็เพื่อที่จะให้การตรวจสอบทำได้อย่างมีประสิทธิภาพสูงสุด โดยจะมีเครื่องที่เป็นศูนย์กลางที่ติดตั้งซอฟท์แวร์สำหรับทำหน้าที่จัดการอุปกรณ์เซ็นเซอร์ทั้งหมดในเครือข่าย ซึ่งจะทำให้กระบวนการในการปรับปรุงความสามารถของอุปกรณ์เซ็นเซอร์ทุกตัวในเครือข่ายสามารถทำได้สะดวกขึ้น อีกทั้งยังสามารถเรียกค้นข้อมูลจากอุปกรณ์เซ็นเซอร์ เพื่อนำมาวิเคราะห์ถึงลักษณะของการบุกรุกโจมตีได้ แต่วิธีการของผู้ผลิตอุปกรณ์ป้องกันการบุกรุกเหล่านี้ก็ยังมีข้อจำกัดอยู่ตรงที่การปรับปรุงความสามารถของการตรวจสอบจากศูนย์กลางนั้น จะทำได้เพียงแค่การปรับเปลี่ยนข้อมูลสัญลักษณ์ของแพ็กเก็ตที่จะใช้ในการตรวจสอบเท่านั้น ซึ่งถ้าหากมีการบุกรุกในรูปแบบใหม่เกิดขึ้น ที่จำเป็นจะต้องเพิ่มความสามารถหรือเปลี่ยนแปลงวิธีการในการตรวจสอบแล้ว ระบบดังกล่าวจะไม่สามารถทำได้จากศูนย์กลาง เช่นการปรับเปลี่ยนหรือเพิ่มเติม “ขั้นตอนวิธี” ที่ใช้ในการตรวจสอบ เพื่อจัดการกับการบุกรุกโจมตีรูปแบบใหม่ ๆ เป็นต้น

อย่างไรก็ตาม วิธีการดังกล่าวข้างต้นนั้นอาจจะไม่สามารถป้องกันการบุกรุกได้ในกรณีที่การตรวจสอบพบการโจมตีนั้นถูกพบโดยอุปกรณ์ป้องกันที่เครือข่ายย่อยของเครื่องเป้าหมาย แต่อุปกรณ์ป้องกันที่เครือข่ายย่อยของเครื่องที่โจมตีตรวจสอบไม่พบ ซึ่งกรณีนี้อาจเกิดขึ้นได้เนื่องจากหลายสาเหตุ เช่น หากผู้บุกรุกใช้วิธีการโจมตีที่ส่งข้อมูลในรูปแบบปกติ แต่ว่าส่งในปริมาณมากเพื่อให้เครื่องเป้าหมายรองรับไม่ไหวจนกระทั่งเครื่องเป้าหมายหยุดทำงานเป็นต้น ซึ่งวิธีการเช่นนี้อุปกรณ์ตรวจสอบที่อยู่ที่เครือข่ายของผู้โจมตีจะไม่สามารถตรวจสอบพบความผิดปกติได้ หากผู้บุกรุกส่งข้อมูลออกมาจำนวนไม่มากนัก แต่จะอาศัยการร่วมมือกันโจมตีจากเครือข่ายย่อยหลาย ๆ เครือข่ายพร้อม ๆ กัน ซึ่งเมื่อข้อมูลที่ถูกส่งมาโจมตีจากหลาย ๆ เครือข่ายมารวมกันที่เครือข่ายเป้าหมาย ก็จะมีปริมาณมากพอที่จะทำให้เครื่องเป้าหมายหยุดทำงานได้ และเมื่ออุปกรณ์ป้องกันที่เครือข่ายเครื่องเป้าหมายตรวจสอบพบความผิดปกติดังกล่าว ก็จะแก้ปัญหาโดยการสกัดกั้นข้อมูลที่จะส่งเข้ามา ดังรูปที่ 3



รูปที่ 3 การใช้หลายเครือข่ายย่อยร่วมกันโจมตี เพื่อหลบหลีกการตรวจสอบจากอุปกรณ์ตรวจสอบที่ติดตั้งไว้ที่บริเวณเครือข่ายย่อยที่ผู้บุกรุกอาศัยเป็นฐานโจมตี

และเนื่องจากข้อมูลที่ส่งมาโจมตีมีรูปแบบที่ปกติ ทำให้อุปกรณ์ตรวจสอบไม่สามารถแยกแยะความแตกต่างของข้อมูลที่ส่งมาโจมตีกับข้อมูลปกติได้ อุปกรณ์ป้องกันก็อาจจะสกัดกั้นข้อมูลทั้งหมด ซึ่งจะไม่ได้กันเฉพาะข้อมูลที่ส่งมาโจมตีเท่านั้น แต่จะกันข้อมูลของผู้ใช้งานปกติไปด้วย ทำให้ผู้ใช้งานปกติไม่สามารถที่จะเข้าใช้งานเครื่องให้บริการที่อยู่ในเครือข่ายนั้นได้ และเมื่อดูปริมาณข้อมูลที่วิ่งเข้ามาในเครือข่ายโดยรวมแล้ว ก็จะพบว่าประสิทธิภาพของเครือข่ายลดลง

ดังนั้นเพื่อที่จะลดระดับความเสียหายจากการถูกโจมตีให้น้อยที่สุด การแก้ปัญหาจึงควรจะป้องกันไม่ให้ผู้บุกรุกส่งข้อมูลออกมาจากเครือข่ายย่อยได้ ซึ่งการที่จะทำอย่างนี้ได้นั้น อุปกรณ์ที่ตรวจสอบพบว่าถูกโจมตี จะต้องตรวจสอบดูจากข้อมูลที่ส่งมาโจมตีนั้นว่ามีต้นกำเนิดมาจากเครื่องที่อยู่ในเครือข่ายใด จากนั้นอุปกรณ์ป้องกันก็จะสามารถสกัดกั้นข้อมูลจากเครือข่ายนั้นไม่ให้ส่งออกมาได้ อย่างไรก็ตามการตรวจสอบเพื่อค้นหาว่าข้อมูลที่ส่งมาโจมตีนั้นถูกส่งออกมาจากเครือข่ายย่อยใด โดยดูจากหมายเลขประจำเครื่องที่ระบุไว้ในข้อมูลจะไม่สามารถทำได้ เพราะการโจมตีวิธีนี้โดยส่วนมากแล้วจะใช้การปลอมแปลงตัวตนที่แท้จริง เพื่อไม่ให้เครื่องเป้าหมายทราบถึงแหล่งที่มาที่แท้จริงของข้อมูลเหล่านั้นได้ ดังนั้นการค้นหาต้นกำเนิดของข้อมูลที่ถูกผู้บุกรุกสร้างปลอมแปลงขึ้นมาจึงไม่ใช่เรื่องง่ายนัก แต่ก็ได้มีผู้คิดค้นและเสนอวิธีการขึ้นมาเพื่อใช้สำหรับการค้นหาต้นกำเนิดของการโจมตี (IP Traceback) ซึ่งก็จะมีอยู่หลายวิธีด้วยกัน เช่น Link test [1], ICMP message traceback [2], Probabilistic Packet Marking [3], Packet Logging [4]

เป็นต้น แต่การที่จะนำเอาวิธีการต่าง ๆ ในการค้นหาต้นกำเนิดของการโจมตีเหล่านั้นมาใช้ในเครือข่าย ก็ต้องมีการปรับเปลี่ยนโครงสร้างของเครือข่ายพอสมควร เช่น ถ้าจะใช้วิธี Packet Logging ก็จะต้องทำการติดตั้งอุปกรณ์เซ็นเซอร์ไว้ในจุดต่าง ๆ ของเครือข่าย หรือหากจะใช้วิธี Probabilistic Packet Marking ก็จะต้องปรับเปลี่ยนเราเตอร์ในเครือข่ายทั้งหมดให้สามารถแก้ไขข้อมูลที่ส่งผ่านได้เป็นต้น ซึ่งในทางปฏิบัติแล้วนั้นจะทำได้ค่อนข้างลำบาก

จากที่กล่าวมาแล้วข้างต้นนั้น การจะป้องกันการบุกรุกโจมตีเครือข่ายได้อย่างมีประสิทธิภาพ ควรจะต้องติดตั้งอุปกรณ์ตรวจสอบและป้องกันการบุกรุกโจมตีเครือข่ายแบบต่าง ๆ ไว้ในเครือข่ายในทุก ๆ จุดที่ผู้บุกรุกจะสามารถอาศัยเป็นช่องทางในการโจมตีได้ ซึ่งจุดที่ผู้วิจัยคิดว่าเหมาะสมที่สุดนั้นก็คือให้ติดตั้งไว้ที่บริเวณขอบของเครือข่ายทั้งหมด คือบริเวณที่เครือข่ายย่อยเชื่อมต่อเข้ากับเครือข่ายหลัก การติดตั้งในลักษณะเช่นนี้ถ้าหากว่าเป็นองค์กรขนาดเล็กก็ไม่น่าจะมีปัญหาอะไร แต่สำหรับในองค์กรขนาดใหญ่ที่มีเครือข่ายย่อยจำนวนมาก ๆ การติดตั้งในลักษณะดังกล่าวจะก่อให้เกิดความยุ่งยากในการบำรุงรักษา การตรวจสอบติดตามสถานะของอุปกรณ์ การปรับปรุงประสิทธิภาพการทำงานและการปรับปรุงความสามารถของอุปกรณ์ เนื่องจากอุปกรณ์ถูกติดตั้งกระจายอยู่ทั่วทั้งเครือข่าย ซึ่งในวิทยานิพนธ์ฉบับนี้จะนำเทคโนโลยี "เครือข่ายแอ็กทีฟ" (Active Network) เข้ามาช่วยแก้ปัญหาดังกล่าว เนื่องจากเครือข่ายแอ็กทีฟซึ่งจะประกอบด้วยอุปกรณ์เครือข่าย "แอ็กทีฟโหนด" (แอ็กทีฟโหนด) ทำหน้าที่เป็นเสมือนเราเตอร์ (Router) ที่มีความสามารถพิเศษ สามารถปรับเปลี่ยนรูปแบบการทำงานได้แบบพลวัต โดยการเปลี่ยนแปลงโปรแกรมที่ใช้ควบคุมตัวมัน ดังนั้นหากเรานำเทคโนโลยีเครือข่ายแอ็กทีฟมาพัฒนาให้เป็นระบบตรวจสอบและป้องกันการบุกรุกโจมตีเครือข่าย เพื่อใช้แทนอุปกรณ์ตรวจสอบป้องกันแบบเดิม ก็จะช่วยให้การปรับเปลี่ยนรูปแบบการทำงาน ที่ใช้สำหรับตรวจสอบการบุกรุกโจมตีสามารถทำได้สะดวกรวดเร็วขึ้น อีกทั้งยังสามารถเขียนโปรแกรมบนแอ็กทีฟโหนดเพื่อใช้แทนอุปกรณ์ตรวจสอบป้องกันการโจมตีหลาย ๆ อย่าง แล้วรวบรวมเข้าไว้ด้วยกันให้ทำงานบนแอ็กทีฟโหนดเพียงตัวเดียวได้ตามความต้องการอีกด้วย ซึ่งจะมีข้อดีกว่าการนำอุปกรณ์ป้องกันหลายประเภทมาติดตั้งให้ทำงานร่วมกันโดยที่อุปกรณ์แต่ละตัวทำงานแยกจากกัน เพราะการใช้แอ็กทีฟโหนดที่มีความสามารถของอุปกรณ์หลายอย่างรวมกัน จะช่วยลดปริมาณงานบางอย่างที่ซ้ำซ้อนกันได้ ดังนั้นผู้วิจัยจึงเห็นว่า การนำเทคโนโลยีเครือข่ายแอ็กทีฟมาพัฒนาเป็นระบบป้องกันการบุกรุกเครือข่ายเพื่อใช้กับเครือข่ายขนาดใหญ่ที่มีจำนวนเครือข่ายย่อยมาก ๆ แล้ว จะช่วยเพิ่มประสิทธิภาพในการตรวจสอบ ป้องกัน และจัดการกับการบุกรุกโจมตีเครือข่ายให้ดีขึ้นกว่าวิธีการป้องกันที่ใช้กันอยู่ในปัจจุบันได้เป็นอย่างดี

1.2 วัตถุประสงค์ของการวิจัย

เพื่อออกแบบและพัฒนาระบบป้องกันการบุกรุกเครือข่ายที่สามารถทำงานได้โดยอัตโนมัติ มีความยืดหยุ่นสูง และสะดวกในการบำรุงรักษาหรือการปรับปรุงความสามารถของระบบ เพื่อใช้ในการป้องกันการบุกรุกโจมตีเครือข่ายคอมพิวเตอร์ขององค์กร จากเครื่องคอมพิวเตอร์ภายในเครือข่ายที่ถูกผู้บุกรุกยึดครอง

1.3 ขอบเขตของการวิจัย

พัฒนาโปรแกรมควบคุมเพื่อใช้กับแอ็กทีฟโหนด ซึ่งโปรแกรมที่จะพัฒนาประกอบไปด้วย

1. โปรแกรมสำหรับค้นหาเครื่องคอมพิวเตอร์ในเครือข่ายย่อย (Computer Discoverer)
2. โปรแกรมค้นหาบริการที่เปิดของเครื่องคอมพิวเตอร์ในเครือข่ายย่อย (Port Scanner)
3. โปรแกรมตรวจสอบชนิดของระบบปฏิบัติการ (OS fingerprint)
4. โปรแกรมตรวจสอบสัญลักษณ์ของแพ็กเก็ต (Packet Signature Detection)
5. โปรแกรมบันทึกข้อมูลแพ็กเก็ตที่วิ่งผ่านแอ็กทีฟโหนด (Packet Logger)
6. โปรแกรมที่ใช้ค้นหาเครือข่ายที่เป็นต้นกำเนิดของแพ็กเก็ต (Packet Traceback)
7. โปรแกรมสำหรับติดต่อสื่อสารกับมาสเตอร์โหนดและแอ็กทีฟโหนดตัวอื่น ๆ
8. โปรแกรมสำหรับควบคุมการทำงานของโปรแกรมต่าง ๆ ในแอ็กทีฟโหนด

พัฒนาซอฟท์แวร์สำหรับมาสเตอร์โหนดเพื่อทำหน้าที่ดังนี้

1. ใช้ติดต่อสื่อสารกับแอ็กทีฟโหนดตัวอื่น ๆ
2. จัดเก็บรายชื่อและกำหนดรหัสประจำตัวของแอ็กทีฟโหนดทั้งหมด
3. ใช้กำหนดโปรแกรมที่จะติดตั้งในแอ็กทีฟโหนดตามนโยบายความปลอดภัย
4. ส่วนติดต่อกับผู้ใช้งาน เพื่อใช้สำหรับให้ผู้ดูแลเครือข่ายใช้ในการดูแลจัดการระบบ

1.4 คำจำกัดความที่ใช้ในการวิจัย

เครือข่ายแบบแอ็กทีฟ แอ็กทีฟโหนด มาสเตอร์โหนด วิธีตรวจสอบการบุกรุกแบบกระจาย

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. งานวิจัยชิ้นนี้จะสามารถนำไปประยุกต์ใช้ในการป้องกันการบุกรุกโจมตีได้ โดยเฉพาะในเครือข่ายขนาดใหญ่ เพื่อเพิ่มประสิทธิภาพในการป้องกันการบุกรุกเครือข่ายจากระบบที่ใช้อยู่ในปัจจุบันได้เป็นอย่างดี

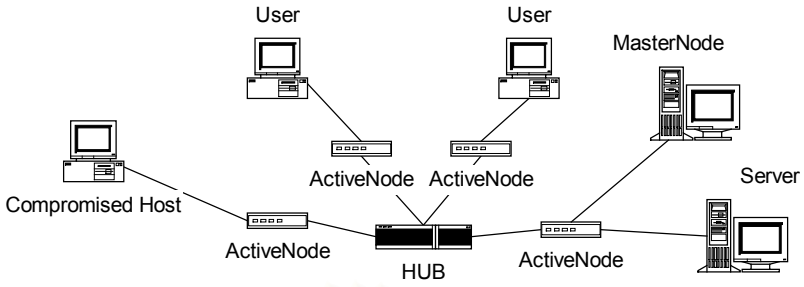
2. ระบบนี้ช่วยทำให้สามารถรวบรวมความสามารถของกระบวนการในการจัดการกับผู้บุกรุกที่มีอยู่หลากหลายรูปแบบ มารวมไว้ในระบบเดียว เป็นผลให้ประสิทธิภาพของการป้องกันดีกว่า และยังมีความยืดหยุ่นสามารถปรับเปลี่ยน กลไกต่าง ๆ ได้ตลอดเวลา
3. ช่วยให้การดูแลรักษา การปรับปรุงระบบให้ทันสมัย ทำได้สะดวกรวดเร็วขึ้น เป็นการลดโอกาสที่ผู้บุกรุกจะสามารถใช้ เพื่อที่จะเข้ามาโจมตีเครือข่ายได้
4. เมื่อถูกผู้บุกรุกโจมตีเครือข่าย ระบบนี้ก็จะสามารถบรรเทาความรุนแรงของการโจมตี โดยการค้นหาและจำกัดขอบเขตของการโจมตี เพื่อให้ผู้ใช้งานอื่น ๆ สามารถใช้งานเครือข่ายต่อไปได้

1.6 วิธีดำเนินการวิจัย

1. ศึกษาถึงวิธีการพัฒนาโปรแกรมบนเครือข่ายแอ็กทีฟ และวิธีการบุกรุกโจมตีแบบต่าง ๆ
2. พัฒนาโปรแกรมควบคุมต่าง ๆ ที่ใช้กับแอ็กทีฟโหนด
3. พัฒนาซอฟท์แวร์ต่าง ๆ บนเครื่องคอมพิวเตอร์ที่จะทำหน้าที่เป็นมาสเตอร์โหนด
4. สร้างเครือข่ายต้นแบบเพื่อใช้สำหรับทดสอบการทำงานและประสิทธิภาพของระบบ
5. ทดสอบความสามารถพื้นฐานต่าง ๆ ของระบบในเครือข่ายต้นแบบที่สร้างขึ้น
6. ทดสอบความสามารถของระบบในการป้องกันการโจมตีเครือข่าย
7. ปรับปรุงความสามารถของระบบและแก้ไขปัญหาที่พบระหว่างการทดสอบ
8. สรุปผลการทดสอบพร้อมข้อเสนอแนะ
9. จัดทำรายงานวิทยานิพนธ์

1.7 ลำดับขั้นตอนในการเสนอผลการวิจัย

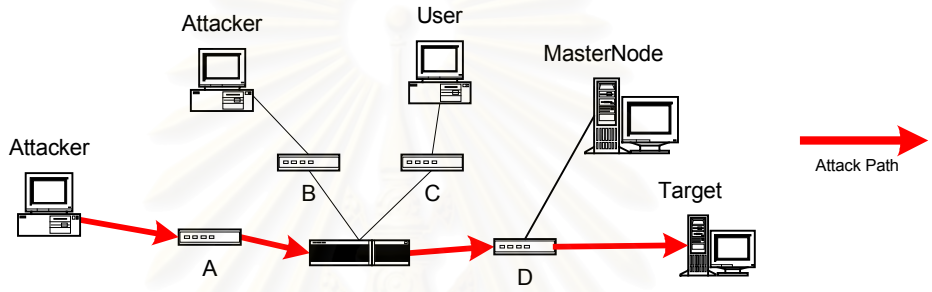
1. สร้างเครือข่ายต้นแบบที่ใช้เทคโนโลยีเครือข่ายแอ็กทีฟ เพื่อใช้ทดสอบการทำงานของระบบ ซึ่งในเครือข่ายต้นแบบจะประกอบไปด้วยแอ็กทีฟโหนด และมาสเตอร์โหนด ที่ติดตั้งโปรแกรมควบคุมต่าง ๆ ที่พัฒนาขึ้นมา และเครื่องคอมพิวเตอร์สำหรับทดสอบที่ติดตั้งอยู่ในเครือข่ายย่อย เช่น เครื่องผู้บุกรุก เครื่องที่ถูกยึดครอง เครื่องผู้ใช้งานทั่วไป เครื่องเป้าหมายที่จะถูกโจมตี เป็นต้น ซึ่งเครือข่ายต้นแบบจะมีลักษณะดังตัวอย่างในรูปที่ 4



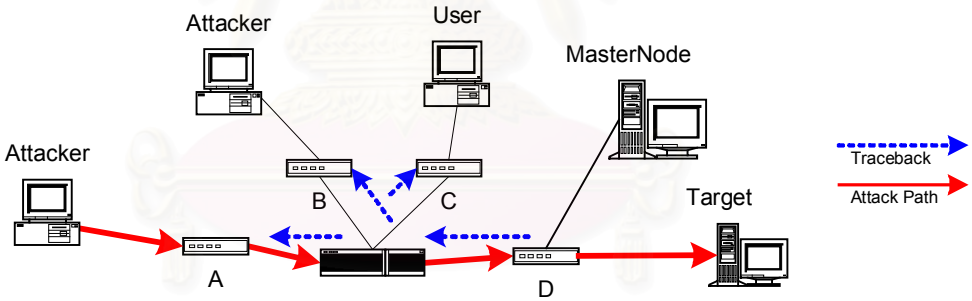
รูปที่ 4 เครือข่ายต้นแบบที่จะสร้างขึ้นเพื่อทดสอบความสามารถในการป้องกันการบุกรุก

2. ทดสอบการทำงานของระบบ ระบบป้องกันการบุกรุกเครือข่ายโดยอัตโนมัติที่จะพัฒนาขึ้นในวิทยานิพนธ์นี้ เมื่อพัฒนาเสร็จแล้วจะมีความสามารถดังต่อไปนี้ ซึ่งสามารถทดสอบความสามารถต่าง ๆ เหล่านี้ได้โดยใช้เครือข่ายต้นแบบที่สร้างขึ้นมาดังกล่าวข้างต้น
   - แอ็กทีฟโหนดสามารถดาวน์โหลดโปรแกรมควบคุมจากมาสเตอร์โหนดมาติดตั้งได้โดยอัตโนมัติ
   - แอ็กทีฟโหนดสามารถค้นหาเครื่องคอมพิวเตอร์ที่อยู่ในเครือข่ายย่อยได้โดยอัตโนมัติ
   - แอ็กทีฟโหนดระบุชนิดของระบบปฏิบัติการและพอร์ตที่เปิดให้บริการของเครื่องคอมพิวเตอร์ที่อยู่ในเครือข่ายย่อยได้
   - แอ็กทีฟโหนดจะตรวจพบการบุกรุกได้ถ้ามีโปรแกรมตรวจสอบสัญลักษณ์แพ็กเก็ตของการบุกรุกนั้นอยู่
   - แอ็กทีฟโหนดสามารถส่งคำขอพร้อมกับข้อมูลแพ็กเก็ตใด ๆ ไปยังแอ็กทีฟโหนดตัวอื่นในเครือข่ายเพื่อค้นหาว่าแอ็กทีฟโหนดตัวใดที่ส่งแพ็กเก็ตดังกล่าวนั้นออกมา
   - แอ็กทีฟโหนดจะสามารถส่งคำขอไปยังแอ็กทีฟโหนดตัวใด ๆ ในเครือข่ายเพื่อให้แอ็กทีฟโหนดตัวนั้นทำการสกัดกั้นการส่งแพ็กเก็ตออกมาจากเครือข่ายย่อยได้
   - ผู้ดูแลเครือข่ายจะสามารถใช้งานมาสเตอร์โหนด เพื่อส่งคำสั่งไปยังแอ็กทีฟโหนดใด ๆ เพื่อให้แอ็กทีฟโหนดนั้นทำการสกัดกั้นการส่งแพ็กเก็ตออกมาจากเครือข่ายย่อยได้
   - ผู้ดูแลเครือข่ายจะสามารถใช้งานมาสเตอร์โหนด เพื่อส่งคำสั่งไปยังแอ็กทีฟโหนดใดเพื่อให้แอ็กทีฟโหนดนั้นทำการดาวน์โหลดโปรแกรมควบคุมจากมาสเตอร์โหนดได้
3. ทดสอบความสามารถในการป้องกันการโจมตีเครือข่าย ระบบที่พัฒนาเสร็จแล้วจะมีความสามารถในการค้นหาและจัดการกับการโจมตีเครือข่ายได้ที่เครือข่ายต้นกำเนิดการโจมตี ซึ่งสามารถทดสอบระบบได้ด้วยการทดลองโจมตีเครือข่ายต้นแบบที่สร้างขึ้นมา ซึ่งในเครือข่ายต้นแบบจะประกอบไปด้วย เครื่องที่เป็นผู้โจมตี เครื่องให้บริการที่เป็นเป้าหมายการโจมตี เครื่องที่เป็นผู้ใช้งานปกติ และแอ็กทีฟโหนด A, B, C, D ดังรูปที่ 5 โดยที่

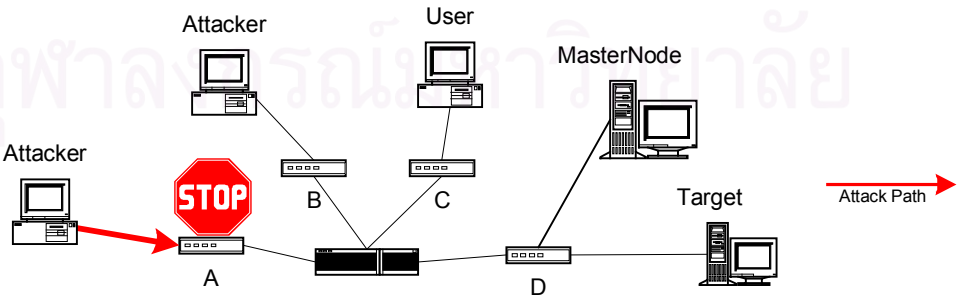
การทดลองโจมตีอาจจะใช้เครื่องมือสำหรับโจมตีเครือข่ายที่มีอยู่ทั่วไป และตั้งสมมติฐานให้เครื่องผู้โจมตีอยู่ในเครือข่ายย่อยของแอ็กทีฟโหนด A และเครื่องที่ถูกโจมตีอยู่ในเครือข่ายย่อยของแอ็กทีฟโหนด D ซึ่งเมื่อแอ็กทีฟโหนด D ตรวจสอบพบการโจมตี ดังรูปที่ 6 ก็จะส่งคำขอให้แอ็กทีฟโหนดตัวอื่นในเครือข่ายค้นหาข้อมูลแพ็กเก็ตที่เก็บไว้เพื่อหาว่าแอ็กทีฟโหนดตัวใดที่ส่งแพ็กเก็ตที่ผิดปกตินั้นออกมาได้ จากนั้นก็จะสามารถส่งคำขอให้แอ็กทีฟโหนดตัวดังกล่าวจัดการสกัดกั้นการส่งข้อมูลออกมาได้ ดังรูปที่ 7 ทำให้เครื่องผู้ใช้งานปกติสามารถเข้าใช้งานเครื่องให้บริการได้ตามปกติ



รูปที่ 5 การทดลองโจมตีเครือข่ายเพื่อทดสอบความสามารถของระบบ



รูปที่ 6 แอ็กทีฟโหนด D ตรวจสอบพบการโจมตีและทำการค้นหาต้นกำเนิดการโจมตี



รูปที่ 7 แอ็กทีฟโหนด A สกัดกั้นการส่งแพ็กเก็ตออกมาจากเครือข่ายย่อยได้

บทที่ 2

เอกสารและงานวิจัยที่เกี่ยวข้อง

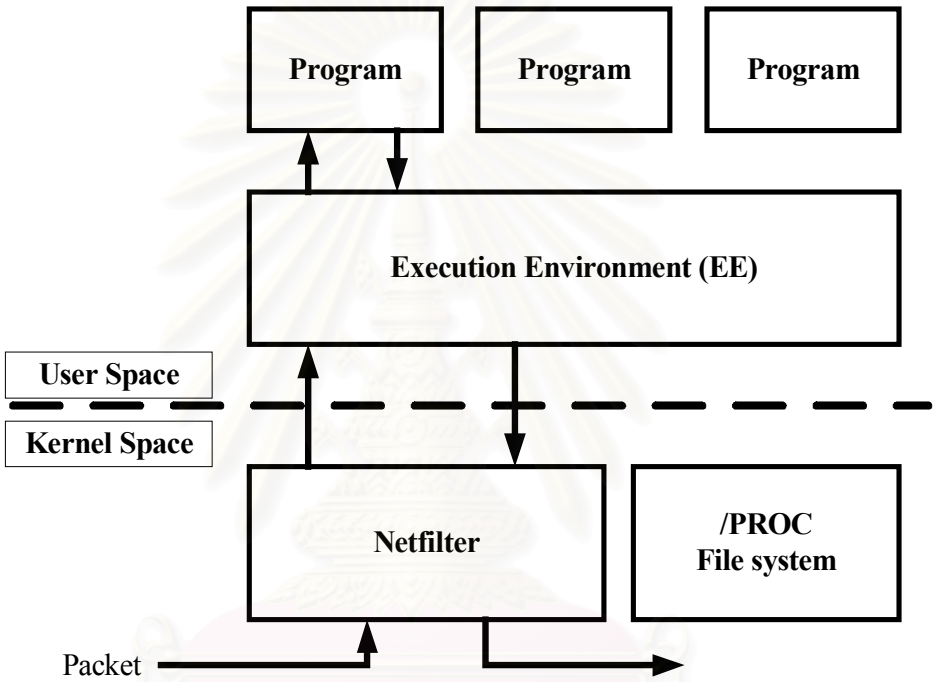
2.1 แนวคิดและทฤษฎีที่เกี่ยวข้อง

เครือข่ายแบบแอ็กทีฟ ประกอบด้วยอุปกรณ์เครือข่ายแอ็กทีฟโหนด (แอ็กทีฟโหนด) หลาย ๆ ตัวต่อร่วมกันเป็นเครือข่าย ทำหน้าที่เสมือนเราเตอร์ (Router) ชนิดพิเศษที่สามารถเขียนโปรแกรมเพื่อควบคุมการทำงานได้ โดยโปรแกรมที่เขียนขึ้นมาเพื่อใช้ควบคุมการทำงานของแอ็กทีฟโหนดนั้น จะถูกส่งเข้าไปในเครือข่ายให้ไปทำงานบนแอ็กทีฟโหนดตัวที่ต้องการได้ และแอ็กทีฟโหนดก็จะจัดการกับข้อมูลที่วิ่งผ่านตามคำสั่งที่ถูกกำหนดไว้ในโปรแกรมควบคุม ซึ่งเทคโนโลยีทางด้านเครือข่ายแอ็กทีฟที่มีอยู่ในปัจจุบันจะสามารถแบ่งได้เป็น 2 ประเภทตามลักษณะของวิธีการส่งโปรแกรมควบคุมเข้าไปในแอ็กทีฟโหนด นั่นคือ

1. วิธีส่งโปรแกรมพร้อมกับข้อมูล (Integrated approach) หรือประเภทแคปซูล (Capsule) ในรูปแบบนี้โปรแกรมควบคุมจะถูกส่งไปพร้อมกับข้อมูล และแอ็กทีฟโหนดจะจัดการกับข้อมูลที่อยู่ในแพ็กเก็ต (Packet) นั้นตามคำสั่งที่อยู่ในโปรแกรมควบคุมที่แนบมาด้วย ซึ่งรูปแบบนี้จะเหมาะกับการประยุกต์ใช้ในงานที่ใช้โปรแกรมควบคุมขนาดเล็ก ๆ ตัวอย่างของเครือข่ายแอ็กทีฟประเภทนี้ได้แก่ ANTS [5], PLAN [6]
2. วิธีส่งโปรแกรมแยกกันกับข้อมูล (Discrete approach) หรือประเภทเครือข่ายที่โปรแกรมได้ (Programmable Network) ในรูปแบบนี้จะเหมาะสำหรับการประยุกต์ใช้งานที่ต้องใช้โปรแกรมควบคุมขนาดใหญ่ และต้องการให้โปรแกรมควบคุมทำงานตลอดเวลา โดยโปรแกรมควบคุมที่ส่งเข้าไปจะมีลักษณะเป็นโปรแกรมเสริม ที่สามารถติดตั้งเพิ่มเติมเข้าไปในโปรแกรมหลักของตัวแอ็กทีฟโหนด เครือข่ายแอ็กทีฟในรูปแบบนี้จะมีวิธีการส่งโปรแกรมควบคุมเข้าไปในเครือข่ายที่แตกต่างจากรูปแบบแรกตรงที่ ตัวโปรแกรมควบคุมกับข้อมูลจะไม่ได้ถูกส่งเข้าไปพร้อมกัน โดยโปรแกรมควบคุมจะถูกส่งเข้าไปติดตั้งไว้ในแอ็กทีฟโหนดก่อน และจะทำหน้าที่จัดการกับข้อมูลที่วิ่งผ่านตัวมันในภายหลัง ซึ่งตัวอย่างของเครือข่ายแอ็กทีฟประเภทนี้ได้แก่ AMnet [7], ASP EE [8]

เครือข่ายแอ็กทีฟที่จะนำมาใช้ในงานวิจัยนี้คือ AMnet ซึ่งเหมาะที่จะนำมาพัฒนาเป็นระบบป้องกันการบุกรุกเครือข่าย เนื่องจากมีลักษณะเป็นเครือข่ายที่โปรแกรมได้ เพราะการทำงานของระบบนี้ จะต้องติดตั้งโปรแกรมตรวจสอบการบุกรุกไว้ในแอ็กทีฟโหนดก่อน แล้วโปรแกรมดังกล่าวจะทำหน้าที่ตรวจสอบแพ็กเก็ตที่วิ่งผ่านในภายหลัง นอกจากนั้น AMnet ยังได้ใช้โครงสร้างพื้นฐานเป็น Netfilter ที่มีมาพร้อมกับระบบปฏิบัติการลินุกซ์ ซึ่งมีคนทดสอบและใช้งานกันทั่วโลก ทำให้มั่นใจได้ในความเสถียรในโครงสร้างพื้นฐานของแอ็กทีฟโหนด อีกทั้งการเขียนโปรแกรมควบคุม

ใน AMnet จะใช้ภาษาซี ที่มีข้อดีในด้านความเร็วในการทำงาน ซึ่งเป็นสิ่งที่จำเป็นอย่างยิ่งกับการนำมาใช้เป็นอุปกรณ์ตรวจสอบการบุกรุก ที่ต้องการการตรวจสอบการบุกรุกและการส่งผ่านข้อมูลด้วยความเร็วสูง โครงสร้างของแอ็กทีฟโหนดชนิดนี้จะเป็นดังรูปที่ 8 ประกอบด้วย Netfilter ทำหน้าที่ตรวจสอบและรับส่งข้อมูลจากเครือข่ายเพื่อส่งต่อให้กับ Execution Environment (EE) ซึ่งทำหน้าที่เชื่อมต่อระหว่าง Netfilter กับโปรแกรมควบคุม และดูแลการทำงานของโปรแกรมควบคุมให้ถูกต้อง



รูปที่ 8. โครงสร้างของ AMnet แอ็กทีฟโหนด

2.2 เอกสารและงานวิจัยที่เกี่ยวข้อง

1. **AMnet 2.0: An improved architecture for programmable networks** [7] เป็นเทคโนโลยีเครือข่ายแอ็กทีฟประเภท Programmable Network ที่อาศัยคุณสมบัติของ Netfilter ที่มีมากับระบบปฏิบัติการลินุกซ์ รุ่น 2.4 ซึ่งผู้วิจัยคาดว่าจะนำมาใช้ในการพัฒนาระบบป้องกันผู้บุกรุกเครือข่ายในวิทยานิพนธ์นี้ โดยที่เครือข่ายแอ็กทีฟนี้จะประกอบไปด้วยแอ็กทีฟโหนด และ โปรแกรมควบคุม Repository ที่ใช้สำหรับเก็บโปรแกรมควบคุมเพื่อห้แอ็กทีฟโหนดมาดาวน์โหลดไปใช้งาน ซึ่งการใช้วิธีนี้จะมีข้อดีในเรื่องการรักษาความปลอดภัย ทำให้มั่นใจได้ว่าโปรแกรมควบคุมที่ดาวน์โหลด มาจะใช้งานได้ดี ไม่ใช่โปรแกรมควบคุมที่ถูกผู้บุกรุกสร้างขึ้นมา เพราะในการดาวน์โหลดโปรแกรมควบคุมไปติดตั้งในแอ็กทีฟโหนดนั้น ก่อนติดตั้งจะต้องตรวจสอบลายมือชื่ออิเล็กทรอนิกส์ของมาสเตอร์โหนดที่แนบมากับโปรแกรมควบคุมด้วยว่าถูกต้องหรือไม่ ซึ่งถ้าไม่ถูกต้องก็สงสัยว่าจะเป็นโปรแกรมควบคุมที่ถูกปลอมแปลงมา
2. **The ASP EE: An Active Network Execution Environment** [8] เป็นเทคโนโลยีเครือข่ายแอ็กทีฟประเภท Programmable Network อีกตัวหนึ่ง โดยตัวแอ็กทีฟโหนดของ ASP จะแตกต่างจาก AMnet ตรงที่จะใช้ Janos [9] แทน Netfilter และตัว EE ถูกพัฒนาขึ้นมาด้วยภาษาจาวา ซึ่งจะมีข้อดีกว่าตรงที่ Janos จะสามารถทำงานได้บนระบบปฏิบัติการอื่น ๆ นอกเหนือจากลินุกซ์ แต่ในด้านประสิทธิภาพนั้นอาจจะน้อยกว่า เนื่องจากเป็นการพัฒนาที่แยกจากตัวระบบปฏิบัติการ ในขณะที่ Netfilter ถูกพัฒนามาพร้อมกับระบบปฏิบัติการ
3. **Active Network Based DDoS Defense** [1] เป็นการจัดการกับการโจมตีโดยใช้แอ็กทีฟโหนดที่ทำหน้าที่เป็นเราเตอร์เชื่อมต่อระหว่างโครงข่ายในการค้นหาต้นกำเนิดของการโจมตี ซึ่งเมื่อเกิดการบุกรุกที่เครื่องเป้าหมายแล้ว จะมีเครื่องควบคุมที่ทำหน้าที่ส่งโปรแกรมสำหรับจัดการกับผู้บุกรุกไปยังตัวแอ็กทีฟโหนด ที่ต่ออยู่กับเครื่องเป้าหมายโดยโปรแกรมดังกล่าวนี้จะทำหน้าที่ควบคุมปริมาณการส่งผ่านข้อมูลไม่ให้เกินค่าที่กำหนดไว้ จากนั้นตัวแอ็กทีฟโหนดดังกล่าวจะดูว่า ข้อมูลที่ส่งมาโจมตีมาจากเครือข่ายไหนที่เชื่อมต่ออยู่ เมื่อพบก็จะคัดลอกโปรแกรมดังกล่าวแล้วส่งต่อไปให้แอ็กทีฟโหนดตัวถัดไป ทำอย่างนี้ไปจนกระทั้งถึงต้นกำเนิดของผู้บุกรุก ซึ่งวิธีการนี้จะใช้การจัดการกับผู้บุกรุกแบบย้อนกลับจากเครื่องที่ถูกโจมตีไปยังผู้บุกรุกครั้งละฮอป (Hop) ซึ่งจะต้องใช้เวลาในการตรวจสอบคัดลอกโปรแกรม และส่งต่อโปรแกรมหลายครั้งกว่าจะไปถึงเครือข่ายที่เป็นผู้โจมตี ซึ่งถ้า

หากว่าข้อมูลที่ส่งมาโจมตีมีปริมาณมาก หรือการตรวจสอบทำได้ช้าแล้ว การส่งโปรแกรมย้อนกลับไปยังเครือข่ายผู้บุกรุกก็อาจจะใช้เวลานานหรืออาจไม่สามารถทำได้เลยก็เป็นได้ ซึ่งจะต่างจากวิธีที่จะใช้ในหัวข้อวิทยานิพนธ์นี้ที่จะใช้แอ็กทีฟโหนดของเครือข่ายที่ถูกโจมตีในการส่งคำขอค้นหาผู้บุกรุกไปยังแอ็กทีฟโหนดอื่น ๆ โดยจะทำการส่งคำขอเพียงแค่แพ็กเก็ตเดียวก็จะสามารถส่งถึงแอ็กทีฟโหนดทุกตัวที่ต่ออยู่บริเวณขอบของเครือข่ายหลักได้ จากนั้นแอ็กทีฟโหนดที่ได้รับคำขอก็จะตรวจสอบในหน่วยความจำของตนเองเพื่อดูว่ามีแพ็กเก็ตที่ตรงกับในคำขอหรือไม่

4. Flexible Intrusion Detection and Response Framework for Active Networks [10] เป็นวิธีการป้องกันและตรวจสอบผู้บุกรุกอีกวิธีหนึ่งที่มีความยืดหยุ่นสูง โดยใช้เครือข่ายแอ็กทีฟของ AMnet ซึ่งแอ็กทีฟโหนดจะทำหน้าที่คล้ายเป็นประตูเข้าออกของแต่ละเครือข่ายย่อย ซึ่งจะทำการโหลดโปรแกรมโมดูลที่อยู่ใน โปรแกรมควบคุม Repository มาเพื่อใช้งานเป็นระบบตรวจสอบและป้องกันการบุกรุกของแต่ละเครือข่ายแยกจากกัน ขึ้นอยู่กับนโยบายรักษาความปลอดภัยที่ถูกกำหนดโดยผู้ดูแลระบบของแต่ละเครือข่ายย่อยไว้ในแอ็กทีฟโหนดนั้นๆ ซึ่งจะแตกต่างกับวิทยานิพนธ์ฉบับนี้ที่ไม่ต้องใช้ผู้ดูและระบบในการกำหนดนโยบายความปลอดภัย แต่ได้เพิ่มกระบวนการในการค้นหาช่องโหว่ของเครือข่ายย่อยโดยอัตโนมัติไว้ และยังมีระบบตรวจสอบเพื่อค้นหาต้นกำเนิดของการโจมตีเพิ่มเข้าไปอีกด้วย รวมทั้งยังเพิ่มความสามารถของมาสเตอร์โหนดเพื่อใช้จัดการกับแอ็กทีฟโหนดอื่น ๆ ได้ และยังมีการติดต่อสื่อสารกันระหว่างแอ็กทีฟโหนดแต่ละตัวได้อีกด้วย
5. Tracing Network Attacks to Their Sources [4] เป็นกลไกสำคัญที่จะนำมาใช้ในการค้นหาต้นกำเนิดของการโจมตีในวิทยานิพนธ์ฉบับนี้ ในงานวิจัยนี้จะใช้วิธีการติดตั้งตัวค้นหาไว้ในทุกจุดที่ต้องการ โดยที่ตัวค้นหาจะทำการเก็บข้อมูลของแพ็กเก็ตที่วิ่งผ่านไปมา ซึ่งจะเก็บเฉพาะข้อมูลบางส่วนของแพ็กเก็ตไว้เท่านั้นเพื่อประหยัดเนื้อที่ในหน่วยความจำซึ่งจะถูกเขียนทับถ้าหากหน่วยความจำเต็ม และเมื่อต้องการค้นหา ก็จะมีเครื่องคอมพิวเตอร์ที่ทำหน้าที่ในการควบคุมการค้นหา ซึ่งจะแตกต่างจากวิธีการที่จะใช้ในวิทยานิพนธ์นี้ที่ใช้แอ็กทีฟโหนดติดตั้งไว้ที่บริเวณขอบของเครือข่ายหลัก ซึ่งนอกจากจะทำหน้าที่ตรวจสอบการบุกรุกแล้ว ยังทำหน้าที่เป็นตัวเก็บแพ็กเก็ตอีกด้วยทำให้ไม่ต้องติดตั้งตัวค้นหาเพิ่มเติมและไม่ต้องมีเครื่องคอมพิวเตอร์ที่จะใช้ควบคุมการค้นหา เพราะจะใช้ความสามารถของแอ็กทีฟโหนดที่ทำงานแทนได้ด้วยตัวเอง อีกทั้งตัว แอ็กทีฟโหนดนั้นยังมีความสามารถใน

การที่จะปรับเปลี่ยนวิธีการในการเก็บข้อมูลแพ็กเก็ตได้สะดวกกว่าด้วย เนื่องจากสามารถปรับเปลี่ยนโปรแกรมควบคุมได้ตลอดเวลา

6. **IBAN: Intrusion Blocker Based on Active Network** [11] เป็นการนำเอาเครือข่ายแอ็กทีฟมาทำเป็นระบบป้องกันการบุกรุกเครือข่าย ซึ่งประกอบไปด้วยแอ็กทีฟโหนด และ “สถานีจัดการ” ที่เป็นโปรแกรมติดต่อผู้ใช้แบบกราฟฟิก ไว้สำหรับให้ผู้ดูแลเครือข่ายใช้ควบคุมการทำงานของ IBAN ซึ่งใน IBAN จะมีโปรแกรมอยู่ 2 ชนิดที่ใช้ในการจัดการผู้บุกรุกคือ Vulnerabilities Scanner และ Intrusion Blockers จุดประสงค์หลักของ IBAN ก็คือใช้สำหรับป้องกันการบุกรุกโจมตีเครือข่ายที่อาศัยช่องโหว่ใหม่ ๆ ที่เพิ่งค้นพบ โดยผู้ดูแลระบบจะต้องติดตามข่าวสารตามแหล่งข้อมูลต่าง ๆ และเมื่อได้รับทราบข่าวว่ามีการค้นพบช่องโหว่ใหม่เกิดขึ้น ผู้ดูแลระบบก็จะใช้ ”สถานีจัดการ” ในการส่ง Scanner เข้าไปยังแอ็กทีฟโหนดต่าง ๆ ในเครือข่าย เพื่อตรวจสอบดูว่ามีเครื่องใดในเครือข่ายที่มีช่องโหว่ดังกล่าว เมื่อพบก็จะส่ง Intrusion Blocker เข้าไปติดตั้งในแอ็กทีฟโหนดนั้น เพื่อทำหน้าที่เป็น IPS คอยตรวจสอบการบุกรุกที่อาศัยช่องโหว่ดังกล่าว ซึ่งถ้าพบก็จะสกัดกั้นไม่ให้ผ่านไปได้ แต่เนื่องจาก IBAN จะใช้เครือข่ายแอ็กทีฟที่เป็น Capsule-Based ซึ่งไม่รองรับการเขียนโปรแกรมในระดับที่จะจัดการกับแพ็กเก็ตได้โดยตรง อีกทั้งเครือข่ายแอ็กทีฟแบบนี้จะให้โปรแกรมทำงานในลักษณะชั่วคราว จึงไม่สามารถที่จะทำเป็น Intrusion Blocker ได้ จึงต้องสร้างซอฟท์แวร์ขึ้นมาต่างหากเพื่อแก้ปัญหา โดยพัฒนาโปรแกรม Intrusion Blocker ด้วยภาษา C++ และต้องทำการติดตั้งโปรแกรมไว้ก่อนในแอ็กทีฟโหนด ซึ่งเวลาจะใช้งาน Intrusion Blocker ผู้ดูแลระบบก็จะใช้ Management Station ส่งโปรแกรมมาสั่งการให้ซอฟท์แวร์ดังกล่าวทำงาน นอกจากนั้นแอ็กทีฟโหนดแบบนี้จะไม่สามารถติดตั้งโปรแกรมเพื่อใช้ในการตรวจสอบแบบถาวรได้ ในขณะที่วิทยานิพนธ์นี้จะใช้เครือข่ายแอ็กทีฟที่เป็น Programmable Network [7] ที่สามารถจัดการกับแพ็กเก็ตได้โดยตรงและรูปแบบของโปรแกรมที่ส่งเข้าไปในแอ็กทีฟโหนด ก็จะเป็นแบบที่สามารถติดตั้งให้ทำงานได้แบบถาวร ซึ่งจะเหมาะสมกว่าที่จะนำมาพัฒนาเป็นระบบป้องกันการบุกรุกเครือข่าย นอกจากนั้นระบบที่จะพัฒนาในวิทยานิพนธ์นี้ยังมีความสามารถหลายอย่างที่ IBAN ไม่มี นั่นคือ ความสามารถในการค้นหาเครือข่ายที่ถูกผู้บุกรุกอาศัยเป็นฐานในการโจมตี ทำให้สามารถจัดการกับการโจมตีในจุดที่เป็นต้นกำเนิดของปัญหาได้ และนอกจากนั้นระบบนี้จะใช้วิธีการค้นหาช่องโหว่ของเครื่องในเครือข่าย 2 ขั้นตอนคือ ขั้นตอนแรกจะทำการค้นหาช่องโหว่โดยอัตโนมัติ เมื่อพบช่องโหว่ก็จะไปดาวน์โหลดโปรแกรมตรวจสอบที่เหมาะสม

มาติดตั้ง ในขณะที่ IBAN นั้นจะต้องใช้ผู้ดูแลระบบเป็นคนทำ ขั้นตอนต่อมาก็จะดูจากแพ็กเก็ตที่วิ่งผ่านไปมา เพื่อดูว่าเครื่องในเครือข่ายมีอะไรเปลี่ยนแปลงไปบ้างเช่น มีการเปลี่ยนแปลงระบบปฏิบัติการใหม่ หรือมีการเปิดใช้งานบริการใหม่ ๆ เป็นต้น ซึ่งเมื่อพบการเปลี่ยนแปลง ก็จะไปดาวน์โหลดโปรแกรมตรวจสอบมาใหม่ให้เหมาะสม ซึ่งใน IBAN จะไม่มีคุณสมบัติเหล่านี้

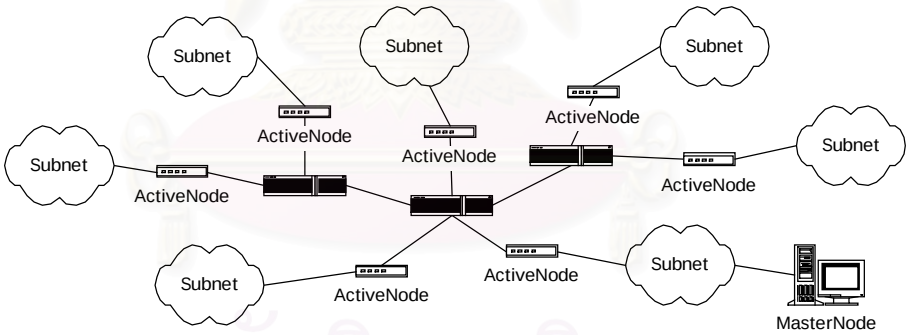
บทที่ 3

วิธีดำเนินการวิจัย

ทำการพัฒนาซอฟท์แวร์บนเครือข่ายแอ็กทีฟ ให้เป็นระบบสำหรับทำหน้าที่ตรวจสอบ ป้องกัน และจัดการกับการบุกรุกโจมตีเครือข่าย แทนที่การใช้งานอุปกรณ์ตรวจสอบป้องกันและจัดการการบุกรุกที่ทำงานแยกจากกัน ซึ่งจะทำให้กลไกที่ใช้ในการป้องกันต่าง ๆ ดังกล่าวสามารถทำงานร่วมกันได้อย่างดี มีความยืดหยุ่นสูง ในการปรับปรุงความสามารถ หรือการเปลี่ยนแปลงคุณสมบัติของระบบ อีกทั้งยังสามารถควบคุมจัดการและดูแลรักษาได้อย่างสะดวกอีกด้วย

3.1 สร้างเครือข่ายเพื่อใช้สำหรับทดสอบการทำงานของระบบที่พัฒนาขึ้น

ส่วนประกอบที่สำคัญของระบบนี้ก็คือแอ็กทีฟโหนด ซึ่งจะถูกติดตั้งไว้ในทุก ๆ จุดที่เป็นบริเวณขอบของเครือข่าย คือทุกจุดที่เครือข่ายย่อยเชื่อมต่อกับเครือข่ายหลัก และจะมีเครื่องที่เป็นศูนย์กลางที่จะทำหน้าที่ดูแลจัดการ การทำงานของแอ็กทีฟโหนดทั้งหมด รวมทั้งเป็นที่เก็บโปรแกรมควบคุมเพื่อให้แอ็กทีฟโหนดมาดาวน์โหลดไปใช้งานได้ ซึ่งต่อไปนี้จะเรียกเครื่องดังกล่าวนี้ว่า “มาสเตอร์โหนด” (มาสเตอร์โหนด) จากรูปที่ 9 จะแสดงให้เห็นถึงการนำแอ็กทีฟโหนดมาติดตั้งในเครือข่ายในลักษณะเป็น ประตูเข้าออก (Gateway) ที่เชื่อมเครือข่ายย่อยเข้าสู่เครือข่ายหลัก ซึ่งเครือข่ายหลักโดยมากก็จะเป็นอุปกรณ์เราเตอร์ที่ทำหน้าที่ในการส่งผ่านแพ็กเก็ต



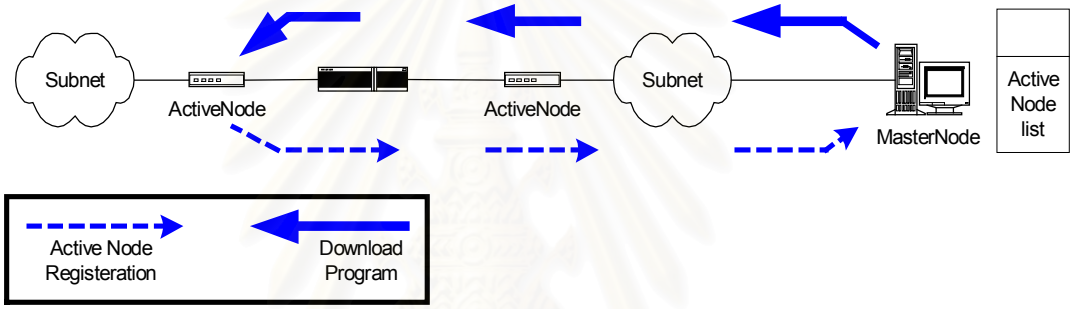
รูปที่ 9 การติดตั้งแอ็กทีฟโหนดและมาสเตอร์โหนดในเครือข่าย

3.2 การทำงานของระบบ

หลักการทำงานของระบบก็คือ อันดับแรกเมื่อแอ็กทีฟโหนดเริ่มทำงานจะตรวจสอบดูว่าตัวเองมีรหัสประจำเครื่องหรือยังถ้ายังไม่มี ก็จะส่งคำขอลงทะเบียนไปยังมาสเตอร์โหนดโดยอัตโนมัติ เมื่อมาสเตอร์โหนดได้รับคำขอลงทะเบียนก็จะกำหนดรหัสประจำเครื่องสำหรับแอ็กทีฟโหนดตัวนั้นแล้วบันทึกลงในตารางรายชื่อแอ็กทีฟโหนด จากนั้นจะส่งรหัสประจำเครื่องกลับไปให้แอ็กทีฟโหนดที่ขอลงทะเบียนมา เมื่อลงทะเบียนเสร็จแล้วแอ็กทีฟโหนดก็จะส่งรายชื่อและรุ่นของโปรแกรมควบคุมทั้งหมดที่แอ็กทีฟโหนดตัวนั้นมีอยู่ไปให้กับมาสเตอร์โหนดเพื่อตรวจสอบว่าแอ็กทีฟโหนดมี

โปรแกรมควบคุมที่จำเป็นต้องใช้งานครบหรือไม่ และเป็นโปรแกรมรุ่นล่าสุดหรือไม่ จากนั้นมาสเตอร์โหนดก็จะตอบกลับเพื่อให้แอ็กทีฟโหนดทราบว่าต้องทำการดาวน์โหลดโปรแกรมจากมาสเตอร์โหนดหรือไม่ ถ้ามีแอ็กทีฟโหนดก็จะทำการดาวน์โหลดโปรแกรมตามที่มาสเตอร์โหนดระบุ ดังรูปที่ 10 ซึ่งตัวอย่างโปรแกรมที่แอ็กทีฟโหนดต้องดาวน์โหลดมาใช้งานจะมีดังเช่น

1. โปรแกรมสำหรับค้นหาเครื่องคอมพิวเตอร์ในเครือข่ายย่อย (Device Discoverer)
2. โปรแกรมสำหรับค้นหาบริการของเครื่องคอมพิวเตอร์ในเครือข่ายย่อย (Port Scanner)
3. โปรแกรมตรวจสอบชนิดของระบบปฏิบัติการในเครือข่ายย่อย (OS fingerprinter)
4. โปรแกรมสำหรับบันทึกข้อมูลแพ็กเก็ตที่วิ่งผ่าน (Packet Logger)
5. โปรแกรมที่ใช้ค้นหาเครือข่ายต้นกำเนิดการโจมตี (Packet Traceback)



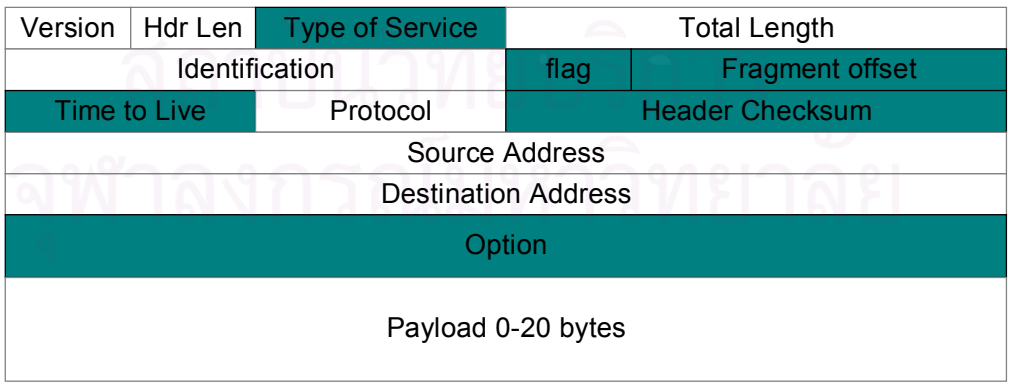
รูปที่ 10 การลงทะเบียนและดาวน์โหลดโปรแกรมจากมาสเตอร์โหนดมาติดตั้ง

เมื่อแอ็กทีฟโหนดดาวน์โหลดโปรแกรมเรียบร้อยแล้วก็จะเริ่มต้นทำการตรวจสอบเครือข่ายย่อยของตนเองทันที ซึ่งสิ่งแรกที่จะทำก็คือจะใช้โปรแกรมค้นหาเครื่องคอมพิวเตอร์ ทำการตรวจดูว่าในเครือข่ายย่อยมีเครื่องคอมพิวเตอร์อะไรอยู่บ้าง เมื่อตรวจสอบเสร็จก็จะใช้โปรแกรมค้นหาช่องโหว่ของเครื่องคอมพิวเตอร์ในเครือข่ายย่อย ทำการตรวจสอบดูว่าเครื่องแต่ละเครื่องในเครือข่ายย่อยมีช่องโหว่อะไรบ้าง เช่น ใช้ระบบปฏิบัติการอะไร รุ่นไหน มีการเปิดให้บริการอะไรบ้าง เป็นต้น จากนั้นเมื่อตรวจสอบเรียบร้อยแล้วก็จะส่งข้อมูลไปให้มาสเตอร์โหนด เพื่อขอดาวน์โหลดโปรแกรมที่ใช้ตรวจสอบสัญลักษณ์ของแพ็กเก็ต เมื่อมาสเตอร์โหนดได้รับข้อมูลก็จะตรวจดูในตารางว่าเครื่องคอมพิวเตอร์ที่มีช่องโหว่ตามที่แอ็กทีฟโหนดส่งมาให้นั้นจะต้องใช้โปรแกรมตรวจสอบสัญลักษณ์ของแพ็กเก็ตตัวไหนบ้าง จากนั้นก็จะส่งโปรแกรมเหล่านั้นกลับไปให้กับแอ็กทีฟโหนดตัวที่ส่งคำขอมาเพื่อนำไปติดตั้งต่อไป จากนั้นแอ็กทีฟโหนดก็จะสั่งให้โปรแกรมตรวจสอบความเปลี่ยนแปลงของระบบปฏิบัติการเริ่มทำงาน ที่ทำหน้าที่คอยอ่านดูข้อมูลที่วิ่งผ่านไปมาเพื่อดูว่าเครื่องต่าง ๆ ในเครือข่ายมีการเปลี่ยนแปลงระบบปฏิบัติการไปหรือไม่ ซึ่งการทำงานของโปรแกรมนี้จะให้ผลในลักษณะคล้ายกันกับโปรแกรมค้นหาช่องโหว่ของเครื่องคอมพิวเตอร์ แต่จะต่างกันตรงที่โปรแกรมตรวจสอบความเปลี่ยนแปลงจะตรวจสอบโดยการดึงข้อมูลขึ้นมาอ่านโดยที่จะไม่ส่งข้อ

มูลออกมารบกวนในเครือข่าย และถ้าหากตรวจสอบพบว่าเครื่องในเครือข่ายมีการเปลี่ยนแปลงแล้ว แอ็กทีฟโหนดก็จะส่งคำขอไปให้กับมาสเตอร์โหนดเพื่อดาวน์โหลดโปรแกรมมาใหม่ให้ตรงกับคุณสมบัติของเครื่องที่เปลี่ยนแปลงไป จากนั้นแอ็กทีฟโหนดก็จะสั่งให้โปรแกรมบันทึกแพ็กเก็ตที่วิ่งผ่านเริ่มทำงาน ซึ่งโปรแกรมนี้จะทำการบันทึกข้อมูลแพ็กเก็ตทั้งหมดที่วิ่งผ่านตัวแอ็กทีฟโหนดเพื่อเอาไว้ใช้ค้นหาต้นกำเนิดของแพ็กเก็ตในภายหลัง

3.3 การบันทึกข้อมูลแพ็กเก็ตและการค้นหาเครือข่ายย่อยที่ถูกผู้บุกรุกใช้เป็นฐานโจมตี

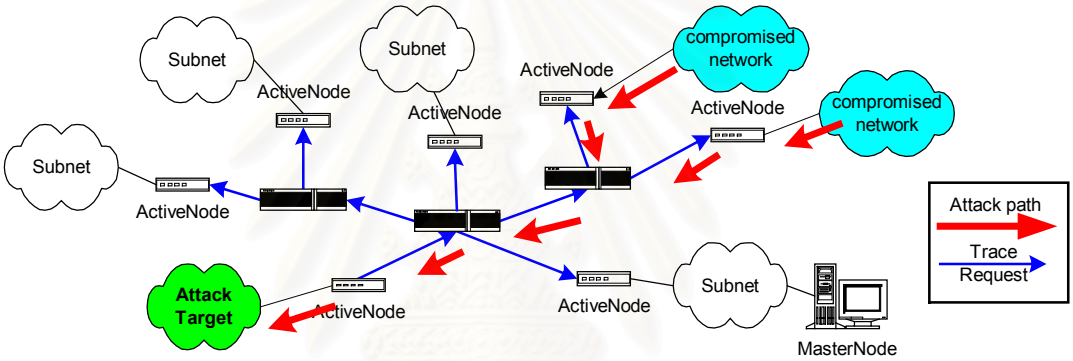
การค้นหาต้นตอของการโจมตีนั้น จะใช้วิธีเก็บข้อมูลของแพ็กเก็ตที่วิ่งผ่านแอ็กทีฟโหนดเพื่อใช้ค้นหาที่มาของแพ็กเก็ตดังกล่าวในภายหลัง ซึ่งแอ็กทีฟโหนดทุกตัวจะทำการเก็บข้อมูลแพ็กเก็ตไว้ในหน่วยความจำ โดยจะเก็บข้อมูลเฉพาะบางส่วนของแพ็กเก็ตที่ไม่มีการเปลี่ยนแปลงระหว่างการเดินทางเท่านั้น ส่วนข้อมูลอื่น ๆ ในแพ็กเก็ตที่อาจถูกเปลี่ยนแปลงระหว่างทาง เช่น TTL, Checksum จะไม่เก็บไว้ เพราะจะไม่สามารถนำมาใช้ในการระบุที่มาของแพ็กเก็ตได้ ซึ่งข้อมูลที่จะเก็บนั้นประกอบไปด้วย Version, Header Length, Total Length, Identification, Protocol, Source Address, Destination Address รวมทั้งหมด 16 Bytes และ Payload อีก 20 Bytes รวมเป็น 34 Bytes ดังรูปที่ 11 ซึ่งเพียงพอที่จะใช้ในการแยกแยะความแตกต่างของแต่ละแพ็กเก็ตที่วิ่งในเครือข่าย [4] การเก็บข้อมูลไว้ในหน่วยความจำนั้นก็เพื่อเพื่อความรวดเร็วในการเก็บและการค้นหา ดังนั้นต้องคำนึงถึงด้วยว่าจะเก็บข้อมูลนี้ไว้นานเท่าใด เนื่องจากหน่วยความจำในแอ็กทีฟโหนดนั้นมีจำกัด ซึ่งโดยเฉลี่ยแล้วแพ็กเก็ตที่วิ่งในเครือข่ายทั่ว ๆ ไปจะมีขนาดประมาณ 400 Bytes [12] ซึ่งเมื่อนำวิธีการนี้มาใช้งานในเครือข่ายที่ความเร็ว 100 Mbps และต้องการเก็บข้อมูลแพ็กเก็ตเป็นเวลา 1 นาทีแล้วจะต้องใช้เนื้อที่ในหน่วยความจำประมาณ 50 Mbytes



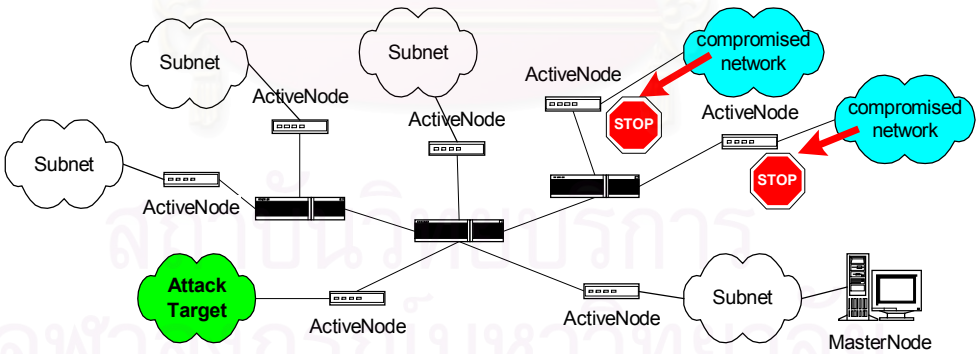
รูปที่ 11 ข้อมูลแพ็กเก็ตที่จะเก็บไว้ (ช่องสีขาว) เพื่อใช้ค้นหาที่มาของแพ็กเก็ตในภายหลัง

จากนั้นหากโปรแกรมตรวจสอบสัญลักษณ์ของแพ็กเก็ตตรวจพบว่าเกิดการโจมตี แอ็กทีฟโหนดก็จะสั่งให้โปรแกรมค้นหาเครือข่ายต้นกำเนิดของการโจมตีเริ่มทำงาน โดยโปรแกรมค้นหาดัง

กล่าวจะรับข้อมูลแพ็กเก็ตของการบุกรุกมาจากโปรแกรมตรวจสอบสัญลักษณ์ของแพ็กเก็ต แล้วเริ่มทำการค้นหาโดยการส่งคำขอพร้อมกับข้อมูลแพ็กเก็ตของการบุกรุกที่ต้องการค้นหา ไปยังแอ็กทีฟโหนดทั้งหมดที่เชื่อมต่ออยู่ในเครือข่าย โดยใช้การส่งแบบมัลติคาสต์ ดังรูปที่ 12 ซึ่งแอ็กทีฟโหนดทุกตัวจะใช้มัลติคาสต์กลุ่มเดียวกัน เมื่อแอ็กทีฟโหนดตัวอื่น ๆ ได้รับคำขอ ก็จะสั่งให้โปรแกรมค้นหาเครือข่ายต้นกำเนิดการโจมตีเริ่มทำงาน โดยการค้นหาข้อมูลแพ็กเก็ตที่ตนเองเก็บไว้ดูว่ามีแพ็กเก็ตที่ตรงกันกับที่ในคำขอส่งมาหรือไม่ ถ้ามีก็จะตอบกลับไปยังแอ็กทีฟโหนดที่ส่งคำขอมา แต่ถ้าไม่พบก็จะไม่ตอบอะไรกลับไป ดังนั้นเมื่อพบแล้วว่าเครือข่ายย่อยใดเป็นต้นกำเนิดของการโจมตี ก็จะสั่งให้แอ็กทีฟโหนดของเครือข่ายย่อยนั้นทำการสกัดกั้นข้อมูลของเครือข่ายนั้นทิ้งไปไม่ให้ส่งออกมาได้อีก ดังรูปที่ 13



รูปที่ 12 แอ็กทีฟโหนดที่ถูกโจมตี ทำการ Traceback ไปยังเครือข่ายของผู้บุกรุก



รูปที่ 13 แอ็กทีฟโหนดของเครือข่ายที่ถูกผู้บุกรุกยึดครองสกัดไม่ให้ผู้บุกรุกส่งข้อมูลออกมา

จะเห็นได้ว่ากระบวนการต่าง ๆ ทั้งหมดนี้ ระบบสามารถทำได้เองโดยอัตโนมัติ ตั้งเริ่มต้นจนถึงการค้นพบผู้บุกรุกเครือข่าย ซึ่งทำให้สามารถป้องกันและแก้ไขปัญหาการโจมตีที่ทำให้เครื่องให้บริการหยุดทำงาน สามารถกลับมาใช้งานได้อย่างรวดเร็ว นอกจากความสามารถในการป้องกันการบุกรุกที่ทำได้รวดเร็วแล้ว การดูแลรักษาระบบ และการปรับปรุงระบบให้ทันสมัยก็เป็นสิ่งสำคัญที่ต้องคำนึงถึง ซึ่งด้วยความสามารถของเครือข่ายแอ็กทีฟทำให้เราสามารถที่จะทำการปรับ

เปลี่ยนหรือแก้ไขการทำงานของตัวแอ็กทีฟโหนดทั้งหมดที่อยู่ในเครือข่ายได้อย่างสะดวกง่ายดาย โดยการเปลี่ยนโปรแกรมควบคุมของแอ็กทีฟโหนดต่าง ๆ ที่อยู่ในมาสเตอร์โหนด แล้วสั่งให้แอ็กทีฟโหนดมาดาวน์โหลดไปใหม่ แต่เนื่องจากว่าในเครือข่ายแอ็กทีฟนี้ไม่ได้มีกระบวนการในการจัดการในส่วนนี้ไว้ให้ จึงต้องพัฒนากลไกในส่วนที่จะใช้จัดการแอ็กทีฟโหนดขึ้นมาเอง โดยใช้ทุกโหนดจะต้องมีโปรแกรมควบคุมสำหรับรับคำสั่งจากมาสเตอร์โหนดเพื่อใช้สำหรับการดูแลแอ็กทีฟโหนดทั้งหมดได้จากจุดเดียว เช่น การปรับเปลี่ยนรูปแบบของการค้นหาเครื่องคอมพิวเตอร์ การปรับปรุงขั้นตอนวิธีในการตรวจสอบผู้บุกรุก การดาวน์โหลดโปรแกรมควบคุมประเภทใหม่ ๆ หรือการเรียกดูสถานะและข้อมูลต่าง ๆ ในตัวแอ็กทีฟโหนด เป็นต้น นอกจากนั้นในกรณีที่ระบบไม่สามารถตรวจสอบพบการบุกรุกได้เองโดยอัตโนมัตินั้น ผู้ดูแลระบบก็สามารถจะใช้มาสเตอร์โหนดสั่งการเองได้โดยตรง เพื่อควบคุมแอ็กทีฟโหนดที่ต้องการได้

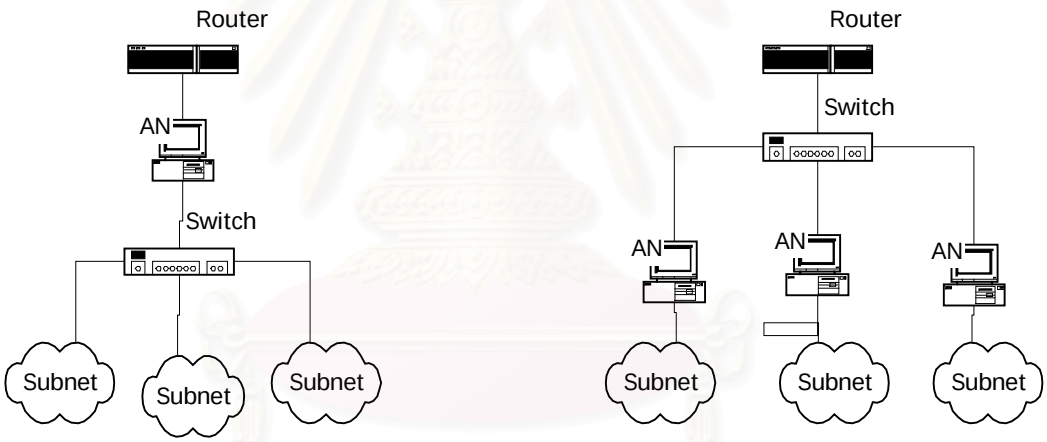
3.4 ความปลอดภัยของระบบ

ในด้านการรักษาความปลอดภัยของตัวระบบเองก็เป็นเรื่องสำคัญที่ควรจะพิจารณาด้วย ซึ่งในระบบป้องกันผู้บุกรุกที่พัฒนานี้ก็อาจจะถูกผู้บุกรุกอาศัยเป็นเครื่องมือในการโจมตีเครือข่ายได้เช่นกัน เช่นผู้บุกรุกอาจจะส่งคำสั่งที่ปลอมขึ้นมาเพื่อหลอกให้แอ็กทีฟโหนดทำการสกัดกั้นข้อมูลของผู้ใช้งานปกติได้ หรืออาจจะส่งโปรแกรมควบคุมปลอมเพื่อไปติดตั้งในแอ็กทีฟโหนดได้ ซึ่งการป้องกันปัญหาการปลอมแปลงคำสั่งหรือโปรแกรมควบคุม ก็จะอาศัยวิธีลงลายมือชื่ออิเล็กทรอนิกส์ไว้ในทุกคำสั่งและทุกโปรแกรมควบคุมที่จะไปติดตั้งในแอ็กทีฟโหนด โดยที่ตัวมาสเตอร์โหนดจะมีกุญแจสาธารณะ (Public Key) และกุญแจส่วนตัว (Private Key) อยู่ 1 คู่ ซึ่งในแอ็กทีฟโหนดทุกตัวจะได้รับกุญแจสาธารณะของมาสเตอร์โหนด เพื่อไว้ใช้ในการตรวจสอบคำสั่งที่ได้รับว่ามาจากมาสเตอร์โหนดจริงหรือไม่ ส่วนกุญแจส่วนตัวจะถูกเก็บเป็นความลับไว้ที่มาสเตอร์โหนด และเพื่อป้องกันผู้บุกรุกใช้วิธีการคัดลอกคำสั่งเก็บไว้แล้วนำมาใช้ในภายหลัง เนื่องจากคำสั่งที่ถูกลงลายมือชื่อ จะเหมือนเดิมทุกครั้งถ้าคำสั่งไม่เปลี่ยน จึงต้องป้องกันโดยการระบุวันที่และเวลาที่จะส่งคำสั่งลงไปในการลงลายมือชื่อด้วย เพื่อที่จะให้คำสั่งเปลี่ยนไปทุกครั้งแม้ว่าข้อมูลที่สั่งจะยังเหมือนเดิม จากนั้นเมื่อแอ็กทีฟโหนดได้รับคำสั่งแล้ว ก็จะตรวจสอบโดยใช้กุญแจสาธารณะของมาสเตอร์โหนด แล้วดูวันที่และเวลาในคำสั่งเทียบกับเวลา ณ.ปัจจุบัน ถ้าหากเวลาตรงกันหรือต่างกันเล็กน้อย ก็จะสามารถมั่นใจได้ว่าเป็นคำสั่งที่ส่งมาจากมาสเตอร์โหนดจริง ส่วนในเรื่องการป้องกันการปลอมแปลงคำสั่งที่ส่งโดยอัตโนมัติจากแอ็กทีฟโหนดนั้น สามารถป้องกันได้เนื่องจากการติดตั้งแอ็กทีฟโหนดจะติดตั้งไว้ที่ทุก ๆ บริเวณที่เป็นขอบของเครือข่ายหลัก ซึ่งในระบบนี้จะกำหนดให้แอ็กทีฟโหนดยอมรับคำสั่งที่ส่งมาจากทางเครือข่ายหลักเท่านั้น ซึ่งก็หมายถึงคำสั่งจะ

ถูกส่งมาจากแอ็กทีฟโหนดตัวอื่น ๆ เท่านั้น ดังนั้นถ้าหากมีการส่งคำสั่งปลอมมาจากเครื่องใด ๆ ในเครือข่ายย่อย คำสั่งนั้นก็จะต้องส่งผ่านตัวแอ็กทีฟโหนดทางด้านที่ต่อกับเครือข่ายย่อยนั้น ซึ่งแอ็กทีฟโหนดก็จะทราบได้ทันทีว่าเป็นคำสั่งปลอมและไม่สนใจคำสั่งนั้น

3.5 การจัดเตรียมแอ็กทีฟโหนด

การจัดเตรียมระบบเพื่อใช้ในการทดสอบนั้น ขั้นแรกต้องนำเครื่องที่จะทำเป็นแอ็กทีฟโหนดมาทำให้มีลักษณะเป็น Transparent Gateway เสียก่อน เพื่อที่จะสามารถนำแอ็กทีฟโหนดไปติดตั้งไว้ระหว่างเครือข่ายหลักกับเครือข่ายย่อยได้สะดวก และไม่กระทบกับระบบเครือข่ายเดิมที่มีอยู่แล้ว โดยจะใช้วิธีทำให้เครือข่ายทั้งสองแยกออกจากกันเฉพาะใน Layer 2 เท่านั้น แต่ใน layer 3 จะไม่ทราบว่าเครือข่ายหลักกับเครือข่ายย่อยแยกออกจากกัน ดังนั้นเมื่อเราได้ทำการติดตั้งแอ็กทีฟโหนดเพิ่มเข้าไปในเครือข่าย ดังรูปที่ 14 ผู้ใช้งานแทบจะไม่ทราบเลยว่าได้มีการติดตั้งแอ็กทีฟโหนดเพิ่มเข้ามา



รูปที่ 14. การติดตั้งแอ็กทีฟโหนดเข้าไปในเครือข่ายโดยไม่มีผลกระทบกับเครือข่ายเดิม

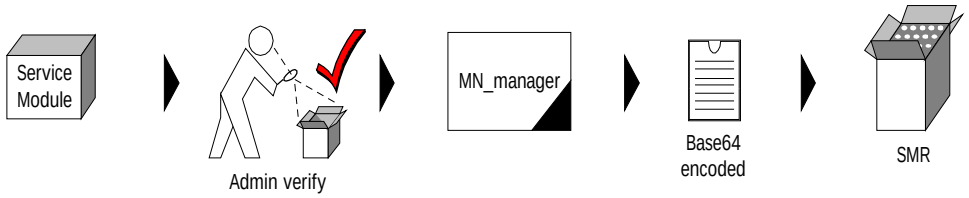
ในการจัดเตรียมแอ็กทีฟโหนดนั้น จะต้องใช้เครื่องคอมพิวเตอร์ที่ติดตั้ง Network Interface Card จำนวน 2 ชุด และติดตั้งระบบปฏิบัติการลินุกซ์ที่มีเคอร์เนลตั้งแต่รุ่น 2.4 ขึ้นไป เนื่องจากว่าซอฟท์แวร์ Active Network ของ AMnet จำเป็นจะต้องใช้ฟังก์ชั่น Netfilter และ Iptables ในการทำงานด้วย ซึ่งฟังก์ชั่นทั้งสองนี้ เริ่มมีอยู่ใน Linux kernel ตั้งแต่รุ่น 2.4 ขึ้นไป นอกจากนั้นก็ยังต้องติดตั้ง Source code ของเคอร์เนลเข้าไปด้วย เพราะสาเหตุแรกคือ AMnet จะทำการแก้ไข library “libipq” ที่ใช้ใน iptables มาปรับปรุงให้เป็น library “libxipq” เพื่อให้ packet queue สามารถรองรับได้หลาย Process ซึ่งรายละเอียดสามารถดูได้จาก www.flexinet.de และสาเหตุที่สองคือเราจะต้องทำการแก้ไขและสร้างเคอร์เนลขึ้นใหม่ ซึ่งสาเหตุที่ต้องสร้างเคอร์เนลขึ้นมาใหม่ ก็เนื่องมาจากว่า การที่เราต้องการทำให้เครื่องคอมพิวเตอร์นี้มีลักษณะเป็น Transparent

Gateway ดังนั้นเราจะต้องเปิด function Bridge ในเคอร์เนลเพื่อให้แพ็กเก็ตถูกส่งผ่านไปมาระหว่างการ์ดเครือข่ายทั้งสอง แต่ว่าเมื่อเราทำการเปิด function Bridge แล้วจะมีผลทำให้ Netfilter ซึ่งทำงานใน Layer 3 จะไม่ได้รับแพ็กเก็ตที่ถูกส่งผ่านไปมาระหว่างการ์ดเครือข่ายทั้งสอง ดังนั้นจึงต้องทำการแก้ไขและสร้างเคอร์เนลขึ้นมาใหม่ เพื่อให้เคอร์เนลสามารถส่งแพ็กเก็ตดังกล่าวขึ้นไปยัง Netfilter ได้ ซึ่งการแก้ไขเคอร์เนลนี้จะทำได้โดยใช้ Patch “ipmode” ซึ่งสามารถหาได้จาก http://www.ssi.bg/~ja และหลังจากที่ทำการแก้ไขและสร้างเคอร์เนลใหม่เสร็จแล้ว ก็เริ่มทำการติดตั้งซอฟท์แวร์ Active Network (AMnet) ซึ่งสามารถดาวน์โหลดซอฟท์แวร์นี้ได้จาก www.flexinet.de

3.6 มาสเตอร์โหนด และโปรแกรมควบคุมมาสเตอร์โหนด MN_manager

ในส่วนของมาสเตอร์โหนด จะใช้เครื่องคอมพิวเตอร์พีซีที่ติดตั้งระบบปฏิบัติการลินุกซ์ เช่นเดียวกันกับที่ใช้ในแอ็กทีฟโหนด ซึ่งก็นับเป็นข้อดีอีกอย่างของระบบนี้ที่ใช้เครื่องพีซี และระบบปฏิบัติการลินุกซ์ เพราะว่าเครื่องพีซีสามารถหาได้ง่าย ราคาไม่แพงมาก อีกทั้งเรายังสามารถนำเครื่องที่ไม่ใช้งานแล้วมาทำเป็นแอ็กทีฟโหนดหรือมาสเตอร์โหนดได้ เนื่องจากระบบนี้ไม่ต้องการทรัพยากรที่สูงมากนัก และระบบปฏิบัติการลินุกซ์ก็เป็นแบบ Open source ซึ่งสามารถนำมาใช้งานได้ฟรี ในวิทยานิพนธ์นี้ได้พัฒนาระบบขึ้นมาโดยเลือกใช้ระบบปฏิบัติการลินุกซ์ทะเล รุ่น 5.0 ซึ่งเป็นระบบปฏิบัติการที่พัฒนาโดย ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC) สามารถดาวน์โหลดได้จาก http://www.opentle.org ซึ่งการจัดเตรียมเครื่องที่จะทำเป็นมาสเตอร์โหนด จะมีอยู่ 2 ขั้นตอน ขั้นตอนแรกคือ การติดตั้ง Service Module Repository (SMR) ที่ทำหน้าที่เก็บโปรแกรมควบคุมต่าง ๆ และให้บริการการดาวน์โหลดโปรแกรมควบคุมกับแอ็กทีฟโหนดทั้งหมดในเครือข่าย และส่วนขั้นตอนที่ 2 คือ การติดตั้งโปรแกรมควบคุม MN_manager และโปรแกรมติดต่อกับผู้ใช้งาน mn.tcl สำหรับให้ผู้ดูแลเครือข่ายใช้ในการสั่งการแอ็กทีฟโหนดทั้งหมดในเครือข่าย

ในส่วนของมาสเตอร์โหนดที่ทำหน้าที่เป็น SMR นั้น จะใช้วิธีการเก็บโปรแกรมควบคุมไว้ใน Directory Server ซึ่งจะใช้ LDAP server ที่มีอยู่แล้วในระบบปฏิบัติการลินุกซ์ทะเล โดยที่การจัดเก็บโปรแกรมควบคุมลงใน SMR นั้น จะต้องนำแฟ้มโปรแกรมที่ผ่านการทดสอบโดยผู้ดูแลเครือข่ายแล้ว ซึ่งจะมีรูปแบบของแฟ้มเป็น Shared Object library มาทำการแปลงให้เป็นแฟ้มประเภท base64 ASCII เสียก่อน จึงจะจัดเก็บลงใน SMR ได้ ซึ่งกระบวนการเหล่านี้จะเป็นหน้าที่ของ MN_manager ที่ผู้ดูแลระบบสามารถเรียกใช้งานได้ ซึ่งขั้นตอนจะเป็นดังรูปที่ 15



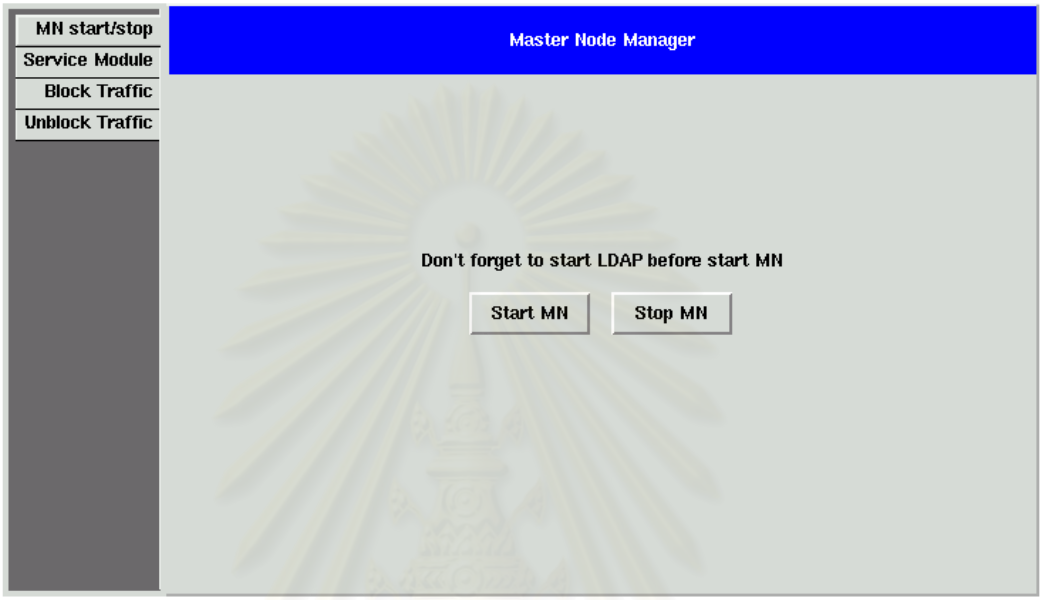
รูปที่ 15. ขั้นตอนการเก็บโปรแกรมควบคุมไว้ในมาสเตอร์โหนด

หน้าที่อีกอย่างของมาสเตอร์โหนดก็คือ การควบคุมแอ็กทีฟโหนดทั้งหมดในเครือข่าย ซึ่งแอ็กทีฟโหนดทุกตัวที่ติดตั้งเพิ่มเข้ามาในเครือข่าย จะต้องลงทะเบียนกับมาสเตอร์โหนดก่อน เพื่อให้มาสเตอร์โหนดตรวจสอบว่าเป็นแอ็กทีฟโหนดที่ถูกต้องหรือไม่ โดยที่แอ็กทีฟโหนดจะต้องส่งคำร้องขอลงทะเบียนไปยังมาสเตอร์โหนด ซึ่งคำร้องดังกล่าวจะต้องผ่านการลงลายมือชื่ออิเล็กทรอนิกส์และจะถูกส่งไปพร้อมกัน เพื่อเป็นการยืนยันว่าเป็นคำร้องดังกล่าวถูกส่งมาจากแอ็กทีฟโหนดที่ถูกต้อง โดยก่อนที่จะมีการติดตั้งแอ็กทีฟโหนดนั้น ผู้ดูแลระบบจะต้องสร้างกุญแจสาธารณะขึ้นมา 1 ชุดประกอบไปด้วย Public key และ Private key ไว้ให้เป็นของแอ็กทีฟโหนดแต่ละตัว ซึ่งระบบจะเก็บ Public key ของแอ็กทีฟโหนดนี้ไว้ในมาสเตอร์โหนด เพื่อใช้ตรวจสอบตัวตนที่แท้จริงของแอ็กทีฟโหนดในภายหลัง และเมื่อแอ็กทีฟโหนดได้ลงทะเบียนเสร็จแล้ว มาสเตอร์โหนดก็จะส่งรายชื่อโปรแกรมควบคุมพื้นฐานที่จำเป็นต้องใช้ไปให้กับแอ็กทีฟโหนด เพื่อให้แอ็กทีฟโหนดเข้ามาดาวน์โหลดโปรแกรมควบคุมเหล่านั้นไปติดตั้ง ซึ่งโปรแกรมควบคุมพื้นฐานนี้จะเป็นโปรแกรมควบคุมที่แอ็กทีฟโหนดทุกตัวจำเป็นจะต้องติดตั้งเพื่อให้มีความสามารถที่จะทำงานเบื้องต้นได้ เช่น โปรแกรมบันทึกข้อมูลแพ็กเก็ตที่วิ่งผ่านแอ็กทีฟโหนด (Packet Logger) เป็นต้น

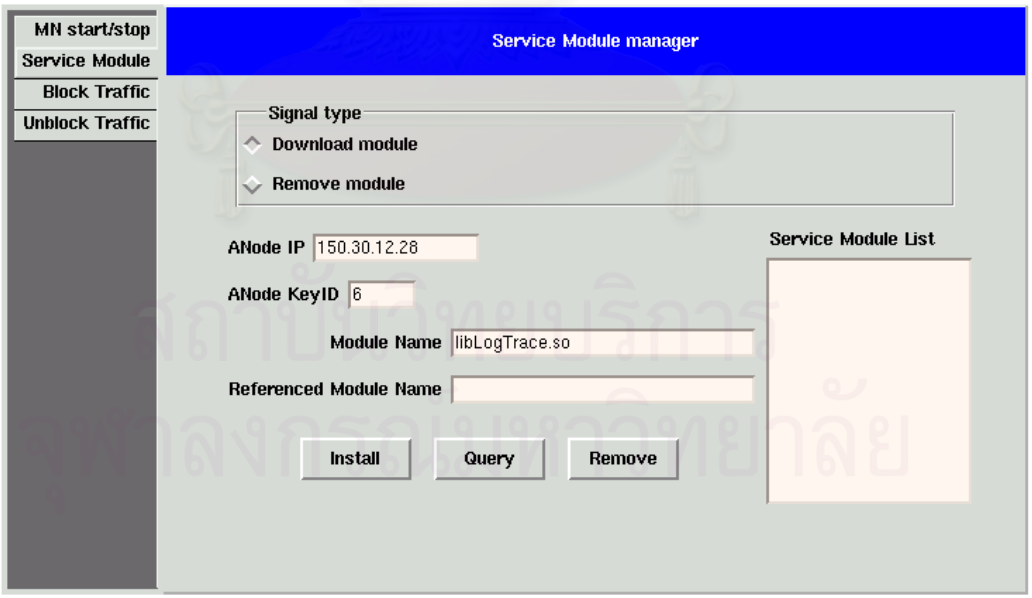
นอกจากนั้นผู้ดูแลเครือข่ายยังสามารถใช้มาสเตอร์โหนดเพื่อสั่งการให้แอ็กทีฟโหนดใด ๆ ทำงานตามต้องการได้ เช่นสั่งให้แอ็กทีฟโหนดสกัดกั้นการส่งผ่านข้อมูลของพอร์ตใด ๆ หรือสั่งให้แอ็กทีฟโหนดทำการดาวน์โหลดโปรแกรมควบคุมใด ๆ ไปติดตั้งเป็นต้น โดยการสั่งงานจะใช้โปรแกรมที่เป็นส่วนติดต่อกับผู้ใช้งานเพื่อให้ผู้ดูแลระบบทำงานได้สะดวก ซึ่งโปรแกรมจะมี 4 ส่วน คือ

1. ส่วนแรกจะใช้สำหรับการเริ่มต้นหรือสิ้นสุดการทำงานของมาสเตอร์โหนด ดังรูปที่ 16
2. ส่วนที่สองจะใช้สำหรับการสั่งการให้แอ็กทีฟโหนดทำการดาวน์โหลดโปรแกรมควบคุมหรือทำการยกเลิกการทำงานของโปรแกรมควบคุมใด ๆ ในตัวแอ็กทีฟโหนด ดังรูปที่ 17
3. ส่วนที่สามจะใช้สำหรับการสั่งให้แอ็กทีฟโหนดทำการสกัดกั้นการส่งแพ็กเก็ตออกมานอกเครือข่ายย่อย โดยที่ผู้ดูแลเครือข่ายสามารถที่จะระบุโปรโตคอล หรือหมายเลขพอร์ตที่ต้องการได้ ดังรูปที่ 18

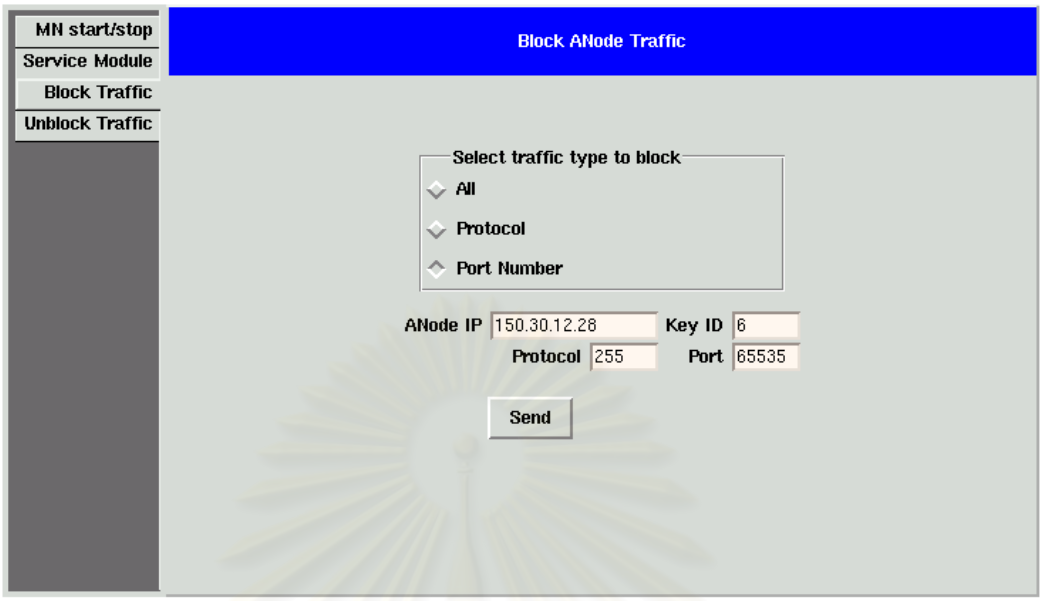
4. ส่วนสุดท้ายก็คือส่วนที่ใช้สำหรับผู้ดูแลเครือข่ายเพื่อสอบถามข้อมูลของแอ็กทีฟโหนดใด ๆ ว่ามีการสกัดกั้นการส่งผ่านข้อมูลใด ๆ อยู่บ้าง และยังใช้สำหรับการยกเลิกคำสั่งการสกัดกั้นการส่งผ่านข้อมูลได้อีกด้วย ดังรูปที่ 19



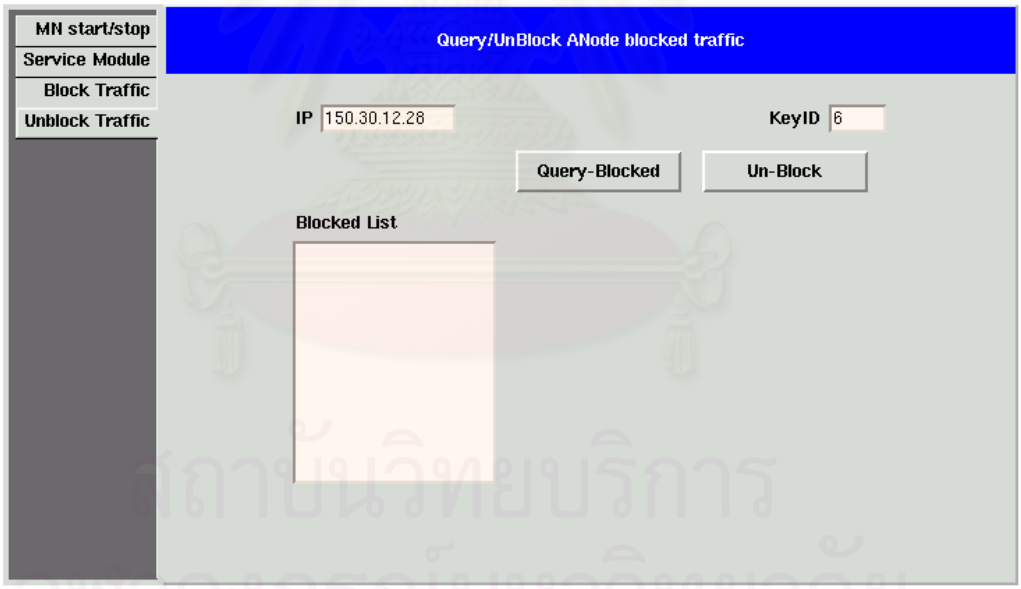
รูปที่ 16 หน้าจอสำหรับสั่งเริ่มต้นหรือสิ้นสุดการทำงานของโปรแกรม MN_manager



รูปที่ 17 หน้าจอสำหรับควบคุมการติดตั้งหรือถอดถอนโปรแกรมควบคุมบนแอ็กทีฟโหนด



รูปที่ 18 หน้าจอสำหรับสั่งให้แอ็กทีฟโหนดสกัดกั้นการส่งข้อมูล



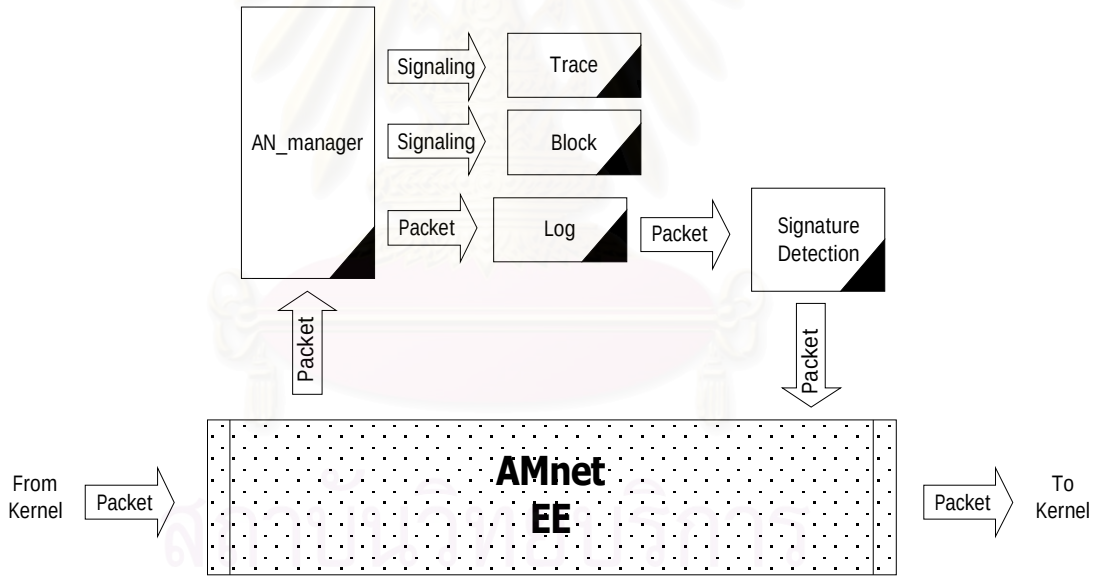
รูปที่ 19 หน้าจอสำหรับสอบถามหรือสั่งให้แอ็กทีฟโหนดยกเลิกการสกัดกั้นการส่งข้อมูล

3.7 โปรแกรม AN_manager สำหรับควบคุมการทำงานของแอ็กทีฟโหนด

เป็นโปรแกรมที่ใช้ควบคุมการทำงานทั้งหมดของแอ็กทีฟโหนด โดยจะถูกติดตั้งเข้าไปใน EE เป็นโปรแกรมแรก และเมื่อ AN_manager เริ่มทำงาน ในขั้นแรกก็จะทำการขอลงทะเบียนกับมาสเตอร์โหนดก่อนเป็นอันดับแรก โดยการส่งคำขอลงทะเบียนไปให้มาสเตอร์โหนดและเมื่อได้รับการตอบกลับมาจากมาสเตอร์โหนดด้วยข้อมูลที่ระบุรายชื่อโปรแกรมควบคุมที่แอ็กทีฟโหนดจำ

เป็นต้องใช้งาน หลังจากนั้นแอ็กทีฟโหนดก็จะเชื่อมต่อไปยัง LDAP server บนมาสเตอร์โหนดเพื่อขอดาวน์โหลด ในการดาวน์โหลดจะใช้วิธีการเข้าไปอ่านข้อมูลใน SMR แล้วเก็บข้อมูลไว้เป็นแฟ้ม ซึ่งแฟ้มที่ได้จะมีรูปแบบเป็น base64 ASCII ซึ่ง AN_manager ก็จะแปลงกลับให้เป็นแฟ้มในรูปแบบ Shared Object Library หลังจากนั้นก็จะทำการติดตั้งโปรแกรมควบคุมเหล่านั้นเข้าไปใน EE เพื่อทำงานต่อไป

หลังจากติดตั้งโปรแกรมควบคุมเรียบร้อยแล้ว แอ็กทีฟโหนดก็จะคอยตรวจสอบดูว่าแพ็กเก็ตที่วิ่งผ่านเข้ามานั้นเป็น คำร้องขอที่ใช้ในระบบนี้หรือไม่ ซึ่งถ้าตรวจสอบแล้วพบว่าไม่ใช่ก็จะส่งผ่านแพ็กเก็ตให้ โปรแกรมควบคุมตัวถัดไปทำหน้าที่ต่อไป แต่ถ้าหากว่าแพ็กเก็ตดังกล่าวเป็นคำร้องขอของระบบ ก็จะดำเนินการตามคำสั่งที่ระบุไว้ในแพ็กเก็ตนั้น ๆ เช่น คำร้องขอค้นหาที่มาของแพ็กเก็ตที่ส่งมาจากแอ็กทีฟโหนดตัวอื่น ๆ หรือการสั่ง Block เครือข่ายย่อยที่ผู้ดูแลเครือข่ายสั่งการโดยใช้มาสเตอร์โหนด เป็นต้น โดยการส่งผ่านแพ็กเก็ตบนตัวแอ็กทีฟโหนดจะเป็นดังรูปที่ 20

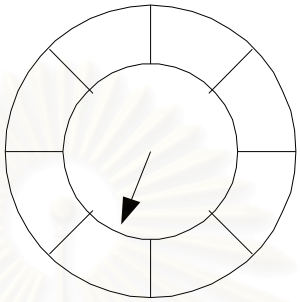


รูปที่ 20. แสดงลักษณะการส่งผ่านข้อมูลบนตัวแอ็กทีฟโหนดผ่านโปรแกรมควบคุมต่าง ๆ

3.8 โปรแกรมควบคุม Log&Trace ใช้บันทึกแพ็กเก็ตและค้นหาที่มาของแพ็กเก็ต

Log&Trace เป็นโปรแกรมควบคุมพื้นฐานที่แอ็กทีฟโหนดทุกตัวจะต้องติดตั้ง ทำหน้าที่ในการเก็บบันทึกข้อมูลบางส่วนของแพ็กเก็ตไว้ในหน่วยความจำ เพื่อที่จะใช้ค้นหาที่มาของแพ็กเก็ตในภายหลัง ซึ่งในการเก็บบันทึกข้อมูลแพ็กเก็ตนี้จะไม่ได้เก็บทั้งแพ็กเก็ต แต่จะเก็บเพียงแค่บางส่วนของแพ็กเก็ตเท่านั้น โดยที่โปรแกรมควบคุมนี้จะจองพื้นที่ในหน่วยความจำไว้เพื่อทำการเก็บ

ข้อมูล และจะใช้วิธีการเก็บในลักษณะเป็น Circular Log โดยข้อมูลแพ็กเก็ตแรกจะถูกเก็บไว้ที่จุดเริ่มต้นของหน่วยความจำที่จองไว้ และจะเก็บต่อเนื่องไปเรื่อย ๆ สำหรับทุกแพ็กเก็ตที่วิ่งผ่าน และเมื่อใดที่หน่วยความจำที่จองไว้เต็ม ก็จะวนกลับมาบันทึกที่จุดเริ่มต้นของหน่วยความจำที่จองไว้อีกครั้ง เป็นอย่างนี้ไปตลอดดังรูปที่ 21



รูปที่ 21. แสดงให้เห็นถึงลักษณะการเก็บบันทึกข้อมูลของแพ็กเก็ต

ในเรื่องของการค้นหาที่มาของแพ็กเก็ตนั้น หากว่า AN_manager ได้รับคำร้องมาจากแอ็กทีฟโหนดตัวอื่น ก็จะทำการค้นหาโดยจะนำข้อมูลแพ็กเก็ตที่ต้องการค้นหา ที่ได้รับมาพร้อมกับคำร้องนั้น นำมาเปรียบเทียบกับแพ็กเก็ตที่เก็บไว้ในหน่วยความจำ ซึ่งถ้าพบว่ามีแพ็กเก็ตที่ตรงกัน ก็จะส่งคำร้องตอบกลับไปยังแอ็กทีฟโหนดตัวที่ส่งคำร้องขอมา

3.9 สัญญาณควบคุมที่ใช้สื่อสารในระบบ

3.9.1 รูปแบบของสัญญาณควบคุม

การส่งสัญญาณควบคุมเพื่อใช้สื่อสารกันระหว่างเครื่องในระบบนี้ จะใช้วิธีการส่งข้อความโดยใช้โปรโตคอล UDP บนพอร์ตหมายเลข 19555 ซึ่งในข้อความที่ส่งจะมีรูปแบบดังรูปที่ 22

Request type	Message length	Source Key ID	Destination Key ID	Digital Signature	Sending Time	Command

รูปที่ 22. รูปแบบของสัญญาณควบคุมที่ใช้สื่อสารกันในระบบ

โดยมีความหมายของแต่ละส่วนดังต่อไปนี้

1. Request type ใช้ระบุประเภทของคำร้อง
2. Message length ใช้ระบุความยาวของคำร้อง
3. Source Key ID เป็นหมายเลขประจำตัวกุญแจสาธารณะของโหนดผู้ส่ง

4. Destination Key ID เป็นหมายเลขประจำตัวกุญแจสาธารณะของโหนดผู้รับ
5. Digital Signature เป็นลายมือชื่ออิเล็กทรอนิกส์เพื่อยืนยันความถูกต้องของคำร้อง
6. Sending Time คือเวลาขณะที่ส่งคำร้อง
7. Command คือข้อมูลที่ระบุคำร้อง ซึ่งจะแตกต่างกันไปในคำร้องแต่ละประเภท

ซึ่งในระบบที่พัฒนาขึ้นมานี้ จะประกอบไปด้วยคำร้องหลายประเภท ยกตัวอย่างเช่น

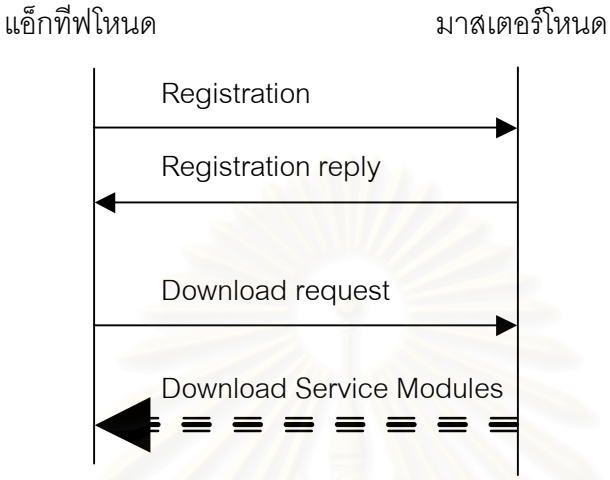
1. Registration แอ็กทีฟโหนดส่งไปมาสเตอร์โหนดเพื่อขอลงทะเบียนตอนเริ่มต้นการทำงาน
2. Registration Reply ส่งจากมาสเตอร์โหนดไปยังแอ็กทีฟโหนด เพื่อตอบรับการลงทะเบียน พร้อมระบุ รายชื่อโปรแกรมควบคุมพื้นฐานที่กำหนดให้แอ็กทีฟโหนดทุกตัวต้องติดตั้ง
3. Trace ส่งจากแอ็กทีฟโหนดใด ๆ ไปยังแอ็กทีฟโหนดตัวอื่น เพื่อค้นหาที่มาของแพ็กเก็ตที่
4. Trace Reply แอ็กทีฟโหนดส่งตอบกลับเมื่อค้นพบแพ็กเก็ตในหน่วยความจำที่บันทึกไว้
5. Block ส่งจากแอ็กทีฟโหนดหรือมาสเตอร์โหนดไปยังแอ็กทีฟโหนดใด ๆ เพื่อขอให้สกัดกั้นการส่งผ่านข้อมูลออกมาจากเครือข่ายย่อย
6. Un-Block ส่งจากแอ็กทีฟโหนดหรือมาสเตอร์โหนดไปยังแอ็กทีฟโหนดใด ๆ เพื่อขอให้ยกเลิกการสกัดกั้นการส่งผ่านข้อมูลออกมาจากเครือข่ายย่อย
7. Download มาสเตอร์โหนดส่งไปแอ็กทีฟโหนดเพื่อสั่งให้ติดตั้งโปรแกรมควบคุมตามที่ระบุ
8. Remove Module มาสเตอร์โหนดส่งไปแอ็กทีฟโหนดเพื่อสั่งให้ถอดถอนโปรแกรมควบคุม

กระบวนการที่ใช้เพื่อรับส่งสัญญาณควบคุมนี้ ในแอ็กทีฟโหนดกับมาสเตอร์โหนดจะใช้วิธีที่แตกต่างกัน โดยที่ในมาสเตอร์โหนดจะเปิดพอร์ต UDP หมายเลข 19555 ไว้เพื่อคอยรับสัญญาณควบคุมที่ส่งเข้ามา แต่ว่าในแอ็กทีฟโหนดจะไม่ได้ใช้วิธีการเปิดพอร์ตทิ้งไว้ เนื่องจากในแอ็กทีฟโหนดจะถูกกำหนดให้ Netfilter ทำการให้ดึงแพ็กเก็ตทุกแพ็กเก็ตที่ถูกส่งผ่านขึ้นมาเพื่อรอให้โปรแกรมควบคุม AN_manager ตรวจสอบอยู่แล้ว ดังนั้นการตรวจสอบสัญญาณควบคุมจะเป็นหน้าที่ของ AN_manager ซึ่งถ้าตรวจพบว่าแพ็กเก็ตที่ได้รับเป็นโปรโตคอล UDP และหมายเลขพอร์ต 19555 ก็แสดงว่าเป็นสัญญาณควบคุมของระบบ ซึ่ง AN_manager ก็จะดำเนินการต่อไปตามคำร้องขอที่กำหนดไว้ในแพ็กเก็ตนั้น ๆ แต่ถ้าหากพบว่าไม่ใช่สัญญาณควบคุม แพ็กเก็ตดังกล่าวก็จะถูกส่งต่อให้กับโปรแกรมควบคุมตัวที่อยู่ถัดไป

3.9.2 การส่งสัญญาณควบคุมของระบบในกรณีต่าง ๆ

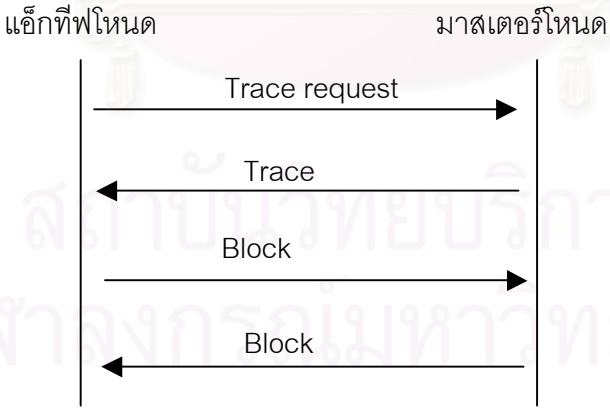
กรณีเริ่มต้นการทำงานของแอ็กทีฟโหนดในครั้งแรก แอ็กทีฟโหนดจะทำการส่งคำร้องขอลงทะเบียนไปยังมาสเตอร์โหนด ซึ่งมาสเตอร์โหนดจะตอบกลับพร้อมรายชื่อโปรแกรมควบคุมพื้น

ฐาน จากนั้น แอ็กทีฟโหนดจะเชื่อมต่อไปยังมาสเตอร์โหนดอีกครั้งเพื่อทำการดาวน์โหลดโปรแกรมควบคุม ดังรูปที่ 23



รูปที่ 23. ตัวอย่างการส่งสัญญาณควบคุมเพื่อสื่อสารกันในการลงทะเบียนกับมาสเตอร์โหนด

กรณีที่แอ็กทีฟโหนดใด ๆ ตรวจพบการบุกรุกเกิดขึ้น และโปรแกรมที่ตรวจสอบพบการบุกรุกได้กำหนดให้ทำการค้นหาที่มาของแพ็กเก็ต ดังนั้นแอ็กทีฟโหนดก็จะส่งคำขอค้นหาที่มาของแพ็กเก็ตไปยังแอ็กทีฟโหนดทุกตัว และรอการตอบกลับ ซึ่งหากว่าโปรแกรมได้กำหนดให้ทำการสกัดกั้นการส่งผ่านข้อมูลด้วย ตัวแอ็กทีฟโหนดก็จะทำการสกัดกั้นการส่งผ่านข้อมูลตามที่ระบุไว้



รูปที่ 24. ตัวอย่างการส่งสัญญาณควบคุมเพื่อสื่อสารกันในการค้นหาที่มาของข้อมูลการบุกรุก

3.9.3 การลงลายมือชื่ออิเล็กทรอนิกส์ในสัญญาณควบคุม

การติดต่อสื่อสารกันของแอ็กทีฟโหนดและมาสเตอร์โหนดที่อาศัยการส่งแพ็กเก็ตสัญญาณควบคุมนั้น จำเป็นจะต้องมีการตรวจสอบที่มาของที่แท้จริงของแพ็กเก็ตด้วย เพราะถ้า

หากไม่มีการตรวจสอบแล้ว ผู้บุกรุกก็จะสามารถสร้างแพ็กเก็ตขึ้นมาเองเพื่อใช้สั่งการให้ แอ็กทีฟโหนดทำงานตามต้องการได้ ซึ่งกระบวนการที่จะนำมาใช้ในการตรวจสอบที่มาของแพ็กเก็ตว่าถูกส่งมาจากแอ็กทีฟโหนดจริงหรือไม่นั้น จะอาศัยการตรวจสอบลายมือชื่ออิเล็กทรอนิกส์เข้ามาช่วยเพื่อพิสูจน์ว่าแพ็กเก็ตที่ได้รับนั้น ได้ถูกส่งมาจากแอ็กทีฟโหนดหรือมาสเตอร์โหนดจริง ๆ หรือว่าจะเป็นแพ็กเก็ตที่ถูกสร้างขึ้นมาโดยผู้บุกรุก

การที่จะพิสูจน์ว่าแพ็กเก็ตที่ได้รับมานั้นถูกต้องแท้จริงหรือไม่ จะใช้วิธีการลงลายมือชื่ออิเล็กทรอนิกส์ของแอ็กทีฟโหนดหรือมาสเตอร์โหนดที่เป็นผู้ส่งแพ็กเก็ต ลงไปในทุก ๆ แพ็กเก็ตของสัญญาณควบคุม และเมื่อผู้รับปลายทางซึ่งอาจจะเป็นมาสเตอร์โหนดหรือแอ็กทีฟโหนดตัวอื่น ๆ เมื่อได้รับแพ็กเก็ตคำสั่งแล้ว ก็จะทำการตรวจสอบลายมือชื่ออิเล็กทรอนิกส์ก่อนว่าเป็นแพ็กเก็ตที่ถูกต้องหรือไม่ ซึ่งถ้าตรวจสอบแล้วพบว่าถูกต้อง ก็จะดำเนินการไปตามคำสั่งที่ระบุมาในแพ็กเก็ตนั้น ๆ แต่ถ้าหากพบว่าเป็นแพ็กเก็ตที่ไม่ถูกต้อง ก็จะทิ้งแพ็กเก็ตนั้นไปโดยไม่ดำเนินการใด ๆ ตามคำสั่งที่ระบุในแพ็กเก็ตดังกล่าว

3.9.4 กุญแจสาธารณะ (Public key, Private key)

กระบวนการในการลงลายมือชื่ออิเล็กทรอนิกส์นั้น จะต้องอาศัยกุญแจ (Key) ในการเข้ารหัสข้อมูล ซึ่งจะต้องใช้ Public key และ Private key ร่วมกับ One-way Hash function โดยจะใช้ Hash function เพื่อทำการ Digest ข้อมูลก่อน ซึ่งจะได้ผลลัพธ์ออกมาเป็น Message Digest จากนั้นก็จะใช้ Private key ในการเข้ารหัสข้อมูล Message Digest นั้น ซึ่งจะได้ผลลัพธ์ออกมาเป็นลายมือชื่ออิเล็กทรอนิกส์ จากนั้นข้อมูลคำสั่งพร้อมกับลายมือชื่ออิเล็กทรอนิกส์ก็จะถูกส่งออกไปด้วยกันเป็นสัญญาณควบคุม ซึ่งเมื่อทางฝั่งผู้รับได้รับสัญญาณควบคุมก็จะนำ Public key ของแอ็กทีฟโหนดหรือมาสเตอร์โหนดที่เป็นผู้ส่ง มาใช้ในการถอดรหัสลายมือชื่ออิเล็กทรอนิกส์ ซึ่งก็จะได้ผลลัพธ์เป็น Message Digest จากนั้นก็ทำการ Hash ข้อมูลที่ได้รับมา และนำมาเปรียบเทียบกัน เพื่อตรวจสอบว่าตรงกันหรือไม่ ซึ่งถ้าตรงกันก็จะมั่นใจได้ว่าเป็นข้อมูลที่ส่งมาจากเจ้าของ Private Key จริง

ในระบบนี้จะใช้กุญแจคู่สาธารณะที่สร้างขึ้นโดยใช้ Digital Signature Algorithm (DSA) ร่วมกับ Secure Hashing Algorithm (SHA-1) ซึ่งในปัจจุบันการ Authentication โดยใช้ Public key algorithm นั้นจะนิยมใช้กันอยู่ 2 แบบคือ RSA และ DSA ซึ่งความแตกต่างของทั้งสองคือ DSA จะใช้เวลาในการ Sign น้อยกว่า RSA แต่ในการ Verify นั้น RSA จะสามารถทำได้เร็วกว่า DSA มาก ๆ แต่ถ้าหากมองโดยรวม ทั้งการ Sign และการ Verify แล้ว DSA จะทำได้เร็วกว่า RSA พอสมควร [13] [14] ดังตารางที่ 1

ตารางที่ 1. เปรียบเทียบเวลาที่ใช้ระหว่าง DSA กับ RSA (หน่วยเป็น 1 ในพันวินาที)

	RSA-1024	DSA-1024
Sign	43	7
Verify	0.6	27
Key Gen	1100	7
Param Gen	0	6,500

สิ่งที่แตกต่างอีกอย่างหนึ่งก็คือขนาดของลายมือชื่อ โดย DSA จะมีขนาดของลายมือชื่อคงที่ไม่ขึ้นกับขนาดของกุญแจที่ใช้ คือประมาณ 46 ไบต์ ส่วนขนาดลายมือชื่อของ RSA นั้น จะมีขนาดเท่ากับขนาดของกุญแจที่เลือกใช้ เช่น ถ้ากุญแจมีขนาด 1024 บิต ลายมือชื่อที่ได้ก็จะมีขนาด 1024 บิต (128 ไบต์) เช่นกัน จะเห็นได้ว่าจากขนาดของลายมือชื่อที่ได้นั้นแบบ DSA จะมีขนาดเล็กกว่าแบบ RSA อยู่มากทีเดียว

ระบบที่พัฒนาขึ้นในวิทยานิพนธ์นี้จะเลือกใช้ DSA ในการทำสร้างลายมือชื่ออิเล็กทรอนิกส์ ซึ่งสาเหตุที่เลือก DSA มาใช้ก็เพราะคุณสมบัติของ DSA ดังที่กล่าวมาแล้วข้างต้น คือมีขนาดของ ลายมือชื่อ ที่เล็กกว่า RSA ซึ่งเป็นสิ่งที่จำเป็นอย่างยิ่ง เพราะการส่งสัญญาณควบคุมในระบบนั้นถ้าจะให้ได้ประสิทธิภาพที่ดีแล้ว ก็ควรจะต้องส่งคำสั่งออกไปด้วยข้อมูลเพียงแพ็กเก็ตเดียว และในแพ็กเก็ตนั้นก็ต้องบรรจุลายมือชื่อเข้าไปด้วย ดังนั้น DSA จึงเหมาะสมกว่า RSA ในแง่ขนาดของลายมือชื่อ ส่วนอีกสาเหตุที่ทำให้เลือกใช้ DSA ก็เพราะความเร็วโดยรวมในการสร้างลายมือชื่อบวกกับการตรวจสอบลายมือชื่อนั้น DSA จะทำได้เร็วกว่า RSA ส่วนในเรื่องของการสร้าง Message Digest นั้นจะใช้ Secure Hashing Algorithm (SHA-1) เนื่องจาก DSA ต้องการ Message Digest ที่มีขนาด 160 บิต (20 ไบต์) และ SHA-1 ก็ถูกออกแบบมาให้ใช้กับ DSA โดยมีขนาด Message Digest เท่ากับ 160 บิต ซึ่งปัจจุบันนี้ในทางปฏิบัติยังไม่พบว่า SHA-1 มีจุดอ่อนที่สามารถโจมตีได้แต่อย่างใด [15]

3.9.5 การสร้างและการจัดการกุญแจคู่สาธารณะ

มาสเตอร์โหนดและแอ็กทีฟโหนดทุกตัวในเครือข่าย จะต้องมีกุญแจคู่สาธารณะประจำตัวของตนเองอยู่ 1 คู่ เพื่อใช้ในการพิสูจน์ตัวตนที่แท้จริงของสัญญาณควบคุมที่ส่งเพื่อสื่อสารกันในเครือข่าย ซึ่งในการสร้างกุญแจคู่สาธารณะจะต้องทำที่ตัวมาสเตอร์โหนดเท่านั้น โดยมีขั้นตอนดังนี้ คือ เมื่อต้องการติดตั้งแอ็กทีฟโหนดขึ้นใหม่ ผู้ดูแลเครือข่ายก็จะต้องสร้างกุญแจคู่สาธารณะ 1 คู่

จากมาสเตอร์โหนด ซึ่งจะได้กุญแจมา 2 ตัวคือ Public key กับ Private key โดยที่มาสเตอร์โหนดจะเก็บ Public key ที่สร้างขึ้นมาใหม่ไว้ในแฟ้มที่ใช้สำหรับเก็บ Public key ของแอ็กทีฟโหนดทุกตัว โดยแฟ้มนี้จะชื่อว่า “All_Publickeys” จากนั้นมาสเตอร์โหนดจะทำการสร้างแฟ้มที่ใช้เก็บกุญแจ Private key ขึ้นเพื่อให้ผู้ดูแลเครือข่ายนำแฟ้มนี้ไปเก็บไว้ในแอ็กทีฟโหนดตัวที่จะสร้างขึ้นใหม่ต่อไป

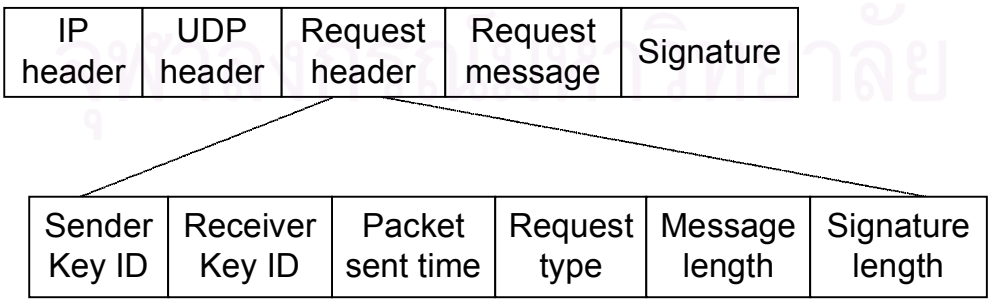
กุญแจสาธารณะที่ถูกสร้างขึ้นมาใหม่โดยมาสเตอร์โหนดนั้น นอกจากจะมี Public key และ Private key แล้ว ยังมีข้อมูลอย่างอื่นที่ถูกกำหนดขึ้นมาพร้อม ๆ กับการสร้างกุญแจ ซึ่งจะมีรายละเอียดดังต่อไปนี้

1. Key ID: เป็นเลขจำนวนเต็ม 1,2,3… เพื่อใช้ระบุ key แต่ละตัวที่ถูกสร้างขึ้นมาทั้งหมด
2. Key generated time: ระบุวันที่สร้าง Key ขึ้นมา เพื่อใช้สำหรับกำหนดอายุการใช้งาน
3. Key status: เป็นสถานะของ Key ซึ่งมีได้หลายสถานะ เช่น NEWKEY, ACTIVE, EXPIRE
4. Key owner’s IP: ใช้ระบุหมายเลขไอพีของแอ็กทีฟโหนดที่เป็นเจ้าของ key

ข้อมูลเหล่านี้จะถูกเก็บไว้ในแฟ้ม All_Publickeys บนตัวมาสเตอร์โหนด คู่กับ Public Key ที่สร้างขึ้น ซึ่งข้อมูลต่าง ๆ เหล่านี้จะถูกนำมาใช้ในกระบวนการติดต่อสื่อสารกันระหว่างแอ็กทีฟโหนดและมาสเตอร์โหนดต่อไปในภายหลัง

3.9.6 รูปแบบของสัญญาณควบคุมแบบต่าง ๆ

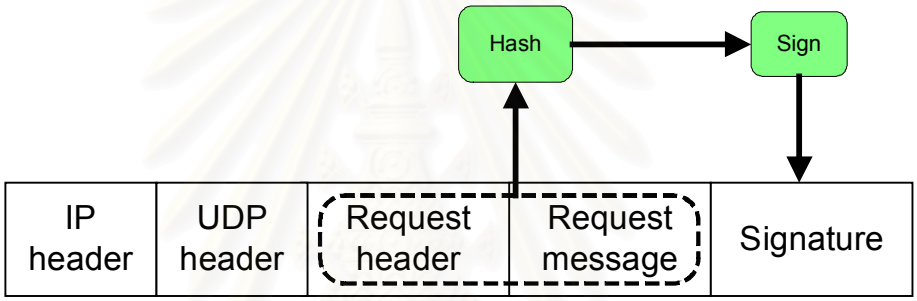
สัญญาณควบคุมที่ถูกส่งออกมาไม่ว่าจะมาจากแอ็กทีฟโหนดหรือมาสเตอร์โหนด จะมีรูปแบบที่เหมือนกัน ดังรูปที่ 25 ซึ่งจะประกอบไปด้วย 3 ส่วนหลัก ๆ โดยส่วนแรกจะเป็น Request Header ซึ่งใช้ระบุข้อมูลต่าง ๆ ที่เกี่ยวกับสัญญาณควบคุมนั้น เช่น Key ID ของผู้ส่ง, เวลาที่ส่ง, ประเภทของคำสั่ง, ความยาวของคำสั่ง, ความยาวของลายมือชื่อ ส่วนที่สองคือข้อมูลคำสั่ง และส่วนสุดท้ายคือลายมือชื่ออิเล็กทรอนิกส์ของคำสั่งนั้น ๆ



รูปที่ 25. แสดงรายละเอียดของข้อมูลที่อยู่ในสัญญาณควบคุม

3.9.7 การลงทะเบียนของแอ็กทีฟโหนด

เมื่อผู้ดูแลเครือข่ายได้ทำการสร้างกุญแจคู่สาธารณะแล้วก็จะได้แฟ้ม Private key มาจาก มาสเตอร์โหนด ซึ่งจะต้องนำแฟ้มดังกล่าวไปเก็บไว้ในเครื่องที่จะใช้เป็นแอ็กทีฟโหนด โดยที่เครื่องดังกล่าวจะต้องติดตั้งซอฟท์แวร์ Active Network (AMnet) พร้อมทั้งโปรแกรมควบคุม AN_manager ไว้ก่อนแล้ว จากนั้นก็จะสามารถเริ่มการทำงานของแอ็กทีฟโหนดได้ทันที โดยในขั้นแรกนั้นแอ็กทีฟโหนดจะส่งสัญญาณควบคุมเพื่อขอลงทะเบียนไปยังมาสเตอร์โหนด ซึ่งก่อนที่จะทำการส่งสัญญาณควบคุมทุกครั้ง แอ็กทีฟโหนดจะทำการลงลายมือชื่อข้อมูลคำร้องขอที่จะส่งออกไปด้วย Private key ของตนเองก่อน ซึ่งการลงลายมือชื่อนี้จะทำกับข้อมูลทั้งในส่วน Request header รวมกับส่วนที่เป็นตัวคำสั่งคือ Request message ดังรูปที่ 26



รูปที่ 26 การลงลายมือชื่อข้อมูลในคำขอ ซึ่งจะทำทั้ง Request header และ Request message

เมื่อมาสเตอร์โหนดได้รับคำขอลงทะเบียนมาแล้ว ก็จะทำการบันทึกการเริ่มใช้งานกุญแจดังกล่าวลงในแฟ้ม All_Publickeys ในกุญแจตัวที่มี Key ID ตรงกับ Key ID ที่ระบุไว้ในคำขอลงทะเบียนนั้น ซึ่งก่อนที่จะบันทึกก็จะต้องตรวจสอบดูด้วยว่าเป็นคำร้องขอที่ส่งมาจากแอ็กทีฟโหนดจริงหรือไม่ ด้วยการตรวจสอบลายมือชื่อ ซึ่งขั้นตอนต่าง ๆ จะมีดังต่อไปนี้

1. ตรวจสอบ Key ID ที่ส่งมาพร้อมกับแพ็กเก็ตเพื่อนำไปค้นหาข้อมูลต่าง ๆ ของกุญแจตัวนั้น ในแฟ้ม All_Publickeys
2. เมื่อได้ข้อมูลของกุญแจตัวนั้นแล้วก็จะอ่านดู Key generated time ว่า Key ID ดังกล่าวมีสถานะเป็นอะไร ซึ่งในกรณีเริ่มต้น Key status จะมีสถานะเป็น “NEWKEY” ซึ่งจะหมายถึงเป็นกุญแจตัวใหม่ที่เพิ่งสร้างขึ้น
3. เมื่อพบว่าเป็นกุญแจตัวใหม่ที่ยังไม่ได้ลงทะเบียน ก็จะดูว่ากุญแจนี้ได้ถูกสร้างขึ้นมานานเกินระยะเวลาที่กำหนดไว้หรือไม่ ซึ่งค่านี้จะสามารถกำหนดได้ใน Configuration file ของมาสเตอร์โหนด เช่น กำหนดไว้ว่าหลังจากที่สร้างกุญแจไปแล้ว แอ็กทีฟโหนดจะต้องมาลง

ทะเบียนกับมาสเตอร์โหนดภายใน 24 ชม. เป็นต้น ซึ่งถ้าเกินเวลาที่กำหนด กุญแจดังกล่าวก็จะไม่สามารถใช้งานได้อีกต่อไป

4. จากนั้นก็จะทำการตรวจสอบลายมือชื่อที่ถูกส่งมาพร้อมกับคำร้องขอ โดยใช้ Public key ของ Key ID ดังกล่าว ซึ่งถ้าตรวจสอบแล้วพบว่าถูกต้อง ก็จะเริ่มดำเนินการตามคำสั่งที่ระบุมาในแพ็กเก็ตนั้นต่อไป

ในขั้นตอนแรกที่เป็นการลงทะเบียน มาสเตอร์โหนดจะทำการเปลี่ยน Key status ในแฟ้ม All_Publickeys ให้มีสถานะเป็น ACTIVE จากนั้นก็จะอ่านหมายเลขไอพีต้นทางของแพ็กเก็ตดังกล่าวเพื่อนำมาบันทึกไว้ในแฟ้ม All_Publickeys เพื่อผูกหมายเลขไอพีไว้กับ Key ID นั้น เมื่อกระบวนการลงทะเบียนเสร็จเรียบร้อย กุญแจดังกล่าวก็พร้อมที่จะใช้งานได้ทันที และมาสเตอร์โหนดก็จะเริ่มทำงานตามขั้นตอนที่เหลือต่อไป ก็คือส่ง Registration Reply ไปให้กับแอ็กทีฟโหนดเพื่อบอกให้แอ็กทีฟโหนดเข้ามาดาวน์โหลดโปรแกรมควบคุมพื้นฐานไปติดตั้ง ซึ่งหลังจากที่แอ็กทีฟโหนดได้ติดตั้งโปรแกรมควบคุมพื้นฐานเสร็จแล้ว ก็จะเป็นอันเสร็จสิ้นกระบวนการลงทะเบียน จากนั้นแอ็กทีฟโหนดก็จะสามารถเริ่มทำงานได้ทันที

3.9.8 การป้องกันการโจมตีโดยใช้วิธี Replay Attack

ถึงแม้ว่าในการส่งข้อมูลคำสั่งไปยังแอ็กทีฟโหนดใด ๆ นั้น จะใช้การลงลายมือชื่ออิเล็กทรอนิกส์เข้ามาช่วย เพื่อป้องกันการส่งคำสั่งปลอมที่ไม่ได้มาจากแอ็กทีฟโหนดที่แท้จริงแล้วก็ตาม แต่ก็อาจจะถูกผู้บุกรุกใช้วิธีการโจมตีแบบ Replay Attack เพื่อสั่งการแอ็กทีฟโหนดได้ โดยที่ผู้บุกรุกจะใช้วิธีการดักจับแพ็กเก็ตที่แอ็กทีฟโหนดส่งเข้าไปในเครือข่ายมาเก็บไว้ก่อน และหลังจากนั้นเมื่อต้องการจะโจมตี ผู้บุกรุกก็จะนำแพ็กเก็ตดังกล่าวที่เก็บไว้ออกมาส่งเข้าไปในเครือข่ายเพื่อใช้สั่งการให้แอ็กทีฟโหนดทำงานได้ ที่เป็นเช่นนี้ก็เนื่องมาจากว่าข้อมูลที่ถูกส่งไปยังแอ็กทีฟโหนดใด ๆ ถ้าคำสั่งในนั้นเป็นคำสั่งเดิมที่เคยส่งไปแล้ว เมื่อส่งใหม่อีกครั้งก็จะมีข้อมูลเหมือนกันทุกครั้ง ซึ่งนั่นก็จะทำให้ลายมือชื่อที่ได้เหมือนกันทุกครั้งด้วย ดังนั้นหากผู้บุกรุกใช้วิธีการโจมตีดังกล่าวแล้ว แอ็กทีฟโหนดที่ได้รับคำสั่งก็จะไม่สามารถรู้ได้เลยว่าคำสั่งนี้ถูกส่งมาจากผู้บุกรุก ดังนั้นเพื่อเป็นการป้องกันการโจมตีในลักษณะนี้ จึงจำเป็นต้องเพิ่มข้อมูลบางอย่างเข้าไปในแพ็กเก็ต เพื่อป้องกันไม่ให้การส่งคำสั่งแบบเดียวกันออกไป 2 ครั้งมีลายมือชื่อเหมือนกัน ซึ่งข้อมูลที่เพิ่มเข้าไปนี้ก็คือ เวลาที่ใช้ในการส่งแต่ละครั้งนั่นเอง ผลที่ได้ก็คือจะได้ลายมือชื่อที่ได้ในแต่ละครั้งไม่ซ้ำกัน ดังนั้นผู้บุกรุกจะไม่สามารถใช้วิธีการโจมตีในลักษณะนี้ได้

นอกจากนั้นผู้บุกรุกยังสามารถใช้วิธีการโจมตีแบบ Replay Attack ได้อีกรูปแบบหนึ่ง คือ การเปลี่ยนหมายเลขไอพีปลายทาง เนื่องจากในการลงลายมือชื่อสัญญาณควบคุมนั้นจะทำกับข้อ

มูลที่ระดับ Application leyer เท่านั้น ดังนั้นหากผู้บุกรุกทำการดักจับแพ็กเก็ตคำสั่งขึ้นมา แล้วทำการเปลี่ยนหมายเลขไอพี ซึ่งเป็นข้อมูลในระดับ Network leyer ไปเป็นหมายเลขไอพีของแอ็กทีฟโหนดตัวอื่น แล้วส่งเข้าไปในเครือข่ายก็จะทำให้ผู้บุกรุกสามารถโจมตีแอ็กทีฟโหนดตัวอื่น ๆ ได้ตามต้องการ ดังนั้นเพื่อป้องกันการโจมตีแบบนี้จึงจำเป็นต้องระบุหมายเลขผู้รับ (Receiver Key ID) เข้าไปในสัญญาณควบคุมด้วย ดังนั้นถึงแม้ว่าผู้บุกรุกจะทำการเปลี่ยนหมายเลขไอพีไปเป็นหมายเลขไอพีของแอ็กทีฟโหนดตัวอื่นก็ตาม แต่เมื่อแอ็กทีฟโหนดปลายทางได้รับแล้วตรวจสอบพบว่า Receiver Key ID ไม่ตรงกับของตัวเอง ก็จะทิ้งแพ็กเก็ตนั้นไป

3.10 การตรวจสอบเครื่องคอมพิวเตอร์ในเครือข่ายย่อยด้วยโปรแกรมควบคุม NetScan

การบุกรุกหรือโจมตีเครือข่ายหรือเครื่องคอมพิวเตอร์ที่อยู่ในเครือข่ายนั้น มักจะอาศัยช่องโหว่ของเครื่องคอมพิวเตอร์ที่อยู่ในเครือข่ายนั้น ๆ ซึ่งโดยมากแล้วก็มักจะขึ้นอยู่กับบริการที่เครื่องดังกล่าวนั้นเปิดให้บริการอยู่ หรือขึ้นอยู่กับระบบปฏิบัติการที่เครื่องนั้นใช้ ยกตัวอย่างเช่น การระบาดของหนอนต่างๆในเครือข่าย เช่น slammer worm [16] จะอาศัยช่องโหว่ของบริการ MS-SQL server หรือ Sasser worm [17] อาศัยช่องโหว่ของระบบปฏิบัติการ windows 2000, windows XP เป็นต้น ดังนั้นหากเราสามารถตรวจสอบได้ว่าในแต่ละเครือข่ายย่อยนั้นมี เครื่องคอมพิวเตอร์ที่ใช้ระบบปฏิบัติการอะไรอยู่บ้าง และเปิดให้บริการอะไรไว้บ้าง ก็จะช่วยให้ผู้ดูแลเครือข่ายสามารถที่จะเตรียมการป้องกันเครือข่ายเหล่านั้นไว้เสียแต่เนิ่น ๆ และด้วยเหตุนี้จึงเป็นที่มาของโปรแกรมควบคุมที่ชื่อว่า NetScan ซึ่งจะทำหน้าที่ในการตรวจสอบเครื่องคอมพิวเตอร์ที่อยู่ในเครือข่ายย่อย ว่าใช้ระบบปฏิบัติการอะไรอยู่บ้าง และมีการเปิดให้บริการพอร์ตหมายเลขใดไว้บ้าง และเมื่อตรวจสอบเสร็จแล้ว ก็จะส่งข้อมูลดังกล่าวให้กับมาสเตอร์โหนด และรอรับคำตอบจากมาสเตอร์โหนดว่าต้องการให้แอ็กทีฟโหนดติดตั้งโปรแกรมตรวจสอบการบุกรุกตัวใดบ้าง เพื่อให้เหมาะสมกับเครื่องคอมพิวเตอร์ที่อยู่ในเครือข่ายย่อยนั้น ๆ ซึ่งวิธีการนี้จะช่วยให้แอ็กทีฟโหนดไม่จำเป็นต้อง ดาวน์โหลดโปรแกรมตรวจสอบการบุกรุกจำนวนมากมาติดตั้งไว้โดยไม่จำเป็น ซึ่งจะทำให้แอ็กทีฟโหนดทำงานได้อย่างมีประสิทธิภาพที่ดีที่สุด ซึ่งโปรแกรมควบคุมที่จะถูกติดตั้งในแอ็กทีฟโหนดจะมีอยู่ 3 ประเภทคือ

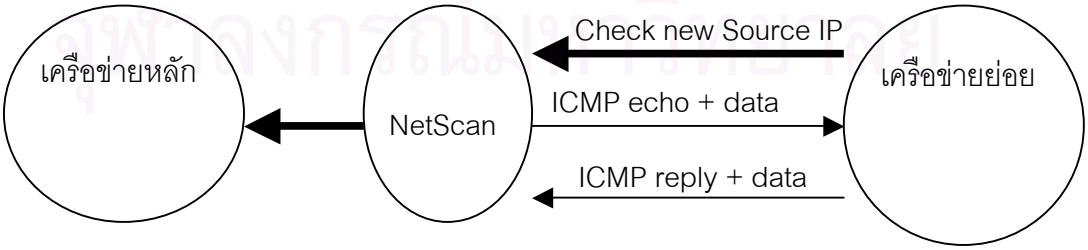
- AN_manager : เป็นโปรแกรมควบคุมหลัก ที่ถูกติดตั้งมาอยู่ก่อนแล้วบนแอ็กทีฟโหนด ทำหน้าที่เป็นตัวควบคุมการทำงานทั้งหมดของแอ็กทีฟโหนด
- โปรแกรมควบคุมพื้นฐาน: เป็นโปรแกรมควบคุมที่จำเป็นจะต้องติดตั้งไว้ที่แอ็กทีฟโหนดทุกตัว ซึ่งโปรแกรมเหล่านี้จะถูกดาวน์โหลดไปติดตั้งไว้ในแอ็กทีฟโหนดทันทีหลังจากเริ่มทำงาน เช่น โปรแกรมควบคุม Log&Trace ที่ใช้เก็บบันทึกแพ็กเก็ตที่วิ่งผ่านตัวแอ็กทีฟโหนด

- โปรแกรมตรวจสอบการบุกรุก: เป็นโปรแกรมควบคุมสำหรับใช้ในการตรวจสอบการบุกรุกหรือโจมตีเครือข่าย ซึ่งในแต่ละเครือข่ายย่อยจะเลือกใช้งานโปรแกรมเหล่านี้แตกต่างกันไป ขึ้นอยู่กับว่าในเครือข่ายย่อยนั้นมีเครื่องคอมพิวเตอร์ที่ใช้ระบบปฏิบัติการอะไร และเปิดให้บริการพอร์ตหมายเลขอะไรบ้าง

โปรแกรมควบคุม NetScan เป็นหนึ่งในโปรแกรมควบคุมพื้นฐานที่แอ็กทีฟโหนดทุกตัวต้องติดตั้ง โดยจะถูกดาวน์โหลดจากมาสเตอร์โหนดในขณะที่เริ่มต้นทำงาน ซึ่งขั้นตอนการทำงานของ NetScan มีดังนี้

1. ทำการตรวจสอบดูว่าในเครือข่ายย่อยนั้นมีเครื่องหมายเลขไอพีใดอยู่บ้าง

โดยในขั้นแรกโปรแกรม NetScan จะเตรียมหน่วยความจำไว้สำหรับเก็บข้อมูลของเครื่องแต่ละตัวที่อยู่ในเครือข่ายย่อย จากนั้นแอ็กทีฟโหนดจะตรวจสอบหมายเลขไอพีต้นทางของทุกแพ็กเก็ตที่วิ่งผ่านเข้ามาจากทางด้านเครือข่ายย่อย และบันทึกหมายเลขไอพีดังกล่าวไว้ ซึ่งถ้าหากพบว่ามีหมายเลขไอพีต้นทางตัวใหม่เพิ่มขึ้นมา ก็จะทำการบันทึกหมายเลขไอพีไว้ แต่ถ้าเป็นหมายเลขไอพีเดิมที่มีอยู่แล้ว ก็จะส่งผ่านแพ็กเก็ตนั้นไปตามปกติ และเพื่อเป็นการป้องกันการโจมตีแอ็กทีฟโหนดจากเครื่องที่อยู่ในเครือข่ายย่อยด้วยการสร้างแพ็กเก็ตที่ปลอมแปลงหมายเลขไอพีต้นทางขึ้นมาจำนวนมาก ๆ เพื่อให้แอ็กทีฟโหนดต้องเก็บหมายเลขไอพีที่ไม่มีอยู่จริงเป็นจำนวนมาก จึงต้องตรวจสอบการมีตัวตนอยู่จริงของหมายเลขไอพีเหล่านั้น โดยก่อนที่แอ็กทีฟโหนดจะเพิ่มข้อมูลหมายเลขไอพีเข้าไปในหน่วยความจำ จะทำการส่ง ICMP echo request พร้อมด้วยข้อมูลไปยังหมายเลขไอพีเหล่านั้น และรอรับ ICMP echo reply ว่าได้ถูกส่งกลับมาหรือไม่ และมีข้อมูลตรงกับที่ส่งไปหรือไม่ ซึ่งหากไม่มีการตอบกลับหรือมีการตอบกลับแต่ข้อมูลไม่ตรง ก็จะไม่บันทึกหมายเลขไอพีนั้น ในทางกลับกัน หากได้รับการตอบกลับและมีข้อมูลตรงกับที่ส่งไป ก็จะบันทึกหมายเลขไอพีนั้นไว้เพื่อใช้ในกระบวนการขั้นต่อไป



รูปที่ 27. แสดงกระบวนการตรวจสอบเครื่องที่ถูกติดตั้งเข้ามาใหม่

อย่างไรก็ตามแอ็กทีฟโหนดก็จะมีการจำกัดจำนวนเครื่องสูงสุดที่รองรับได้ ซึ่งหากมีการพยายามที่จะดักฟัง ICMP echo request แล้วสร้าง ICMP echo reply ที่มีข้อมูลถูกต้องขึ้นมาเพื่อ

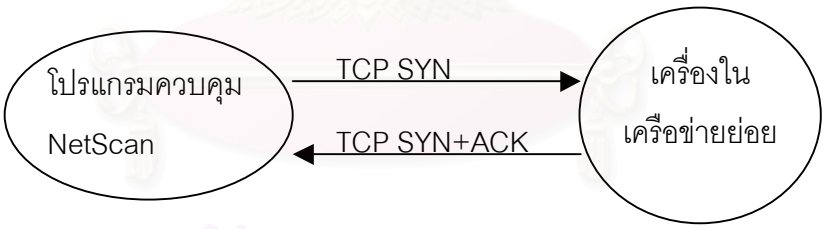
โจมตีแอ็กทีฟโหนดแล้ว จึงต้องป้องกันโดยการตั้งค่าจำนวนเครื่องสูงสุดไว้ ซึ่งเมื่อหมายเลขไอพีที่บันทึกไว้มีจำนวนสูงเกินค่าที่กำหนดแล้ว แอ็กทีฟโหนดก็จะส่งข้อความ เพื่อแจ้งไปยังมาสเตอร์โหนดให้ทราบถึงความผิดปกติที่เกิดขึ้นในเครือข่ายย่อยนั้น ๆ

2. ทำการตรวจสอบเครื่องเหล่านั้นว่ามีการเปิดบริการพอร์ตหมายเลขอะไรไว้บ้าง

หลังจากที่ได้หมายเลขไอพีของเครื่องคอมพิวเตอร์ที่อยู่ในเครือข่ายย่อยแล้ว ขั้นตอนต่อไปของ NetScan ก็คือการตรวจสอบดูว่าเครื่อง ดังกล่าวมีการเปิดให้บริการอะไรไว้บ้าง โดยทำการตรวจสอบพอร์ตที่เปิดไว้ ทั้งประเภท TCP และ UDP ซึ่งผู้ดูแลเครือข่ายสามารถกำหนดหมายเลขพอร์ตที่ต้องการตรวจสอบไว้ในโปรแกรมควบคุม NetScan ได้

2.1 การค้นหาพอร์ตแบบ TCP ที่เปิดให้บริการในเครือข่ายย่อย

โปรแกรมควบคุม NetScan จะทำการตรวจสอบพอร์ต TCP ของเครื่องในเครือข่ายย่อย โดยการส่ง TCP SYN ไปยังเครื่องดังกล่าวแล้วรอรับ TCP SYN+ACK จากเครื่องนั้น ซึ่งถ้าได้รับ SYN+ACK ก็แสดงว่าพอร์ตหมายเลขดังกล่าวเปิดอยู่ ก็จะบันทึกหมายเลขพอร์ตนั้นเก็บไว้ พร้อมกับส่ง TCP Reset ออกไปเพื่อจบการติดต่อ แต่ถ้าหากว่าไม่ได้รับ SYN+ACK หรือว่าได้รับ Reset ก็แสดงว่าพอร์ตหมายเลขดังกล่าวไม่ได้เปิดให้บริการ ก็จะข้ามไปตรวจสอบพอร์ตหมายเลขถัดไป

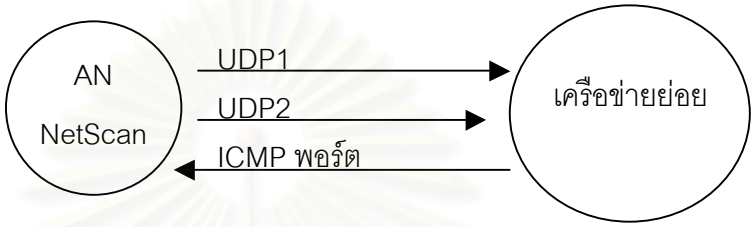


รูปที่ 28. การค้นหาบริการแบบ TCP ที่เปิดอยู่ของเครื่องในเครือข่ายย่อย

2.2 การค้นหาพอร์ตแบบ UDP ที่เปิดให้บริการในเครือข่ายย่อย

จะใช้วิธีการคล้าย ๆ กับแบบ TCP ก็คือ ส่งแพ็กเก็ตออกไปเพื่อทำการทดสอบพอร์ตแล้วรอการตอบกลับ ซึ่งในกรณีของ UDP จะทำการส่งแพ็กเก็ตที่ระบุหมายเลขพอร์ตปลายทางที่ต้องการตรวจสอบ แต่ว่าการตรวจสอบพอร์ตประเภท UDP จะแตกต่างจาก TCP ตรงที่ TCP จะมีการตอบกลับมาหากว่าพอร์ตถูกเปิดอยู่ แต่ในกรณีของ UDP นั้นจะไม่มีการตอบกลับหากว่าพอร์ตเปิดอยู่ แต่ในทางกลับกันถ้าหากว่าพอร์ตถูกปิด จะมีการตอบกลับมาด้วย ICMP พอร์ต unreachable ดังนั้นจึงเป็นไปได้ว่าหากพอร์ตถูกปิดและ

เครื่องได้ส่ง ICMP พอร์ต unreachable ออกมา แต่ส่งมาไม่ถึงตัวแอ็กทีฟโหนดเนื่องด้วยสาเหตุใดก็ตาม ก็อาจทำให้เข้าใจผิดว่าพอร์ตนั้นถูกเปิดอยู่ ดังนั้นเพื่อให้แน่ใจว่าพอร์ตเปิดอยู่จริง จึงต้องตรวจสอบ 2 ครั้งต่อหนึ่งพอร์ต เพื่อดูว่ามี ICMP พอร์ต unreachable ตอบกลับมาหรือไม่ ถ้าไม่มีก็จะถือว่าพอร์ตดังกล่าวเปิดอยู่ ดังนั้นโปรแกรมควบคุม NetScan ก็จะบันทึกหมายเลขพอร์ตนั้นไว้ เพื่อส่งข้อมูลให้มาสเตอร์โหนดต่อไป



รูปที่ 29. การค้นหาบริการแบบ UDP ที่เปิดอยู่ของเครื่องในเครือข่ายย่อย

และเนื่องจากในข้อกำหนดของ RFC-1812 (section 4.3.2.8) [18] ได้มีการกำหนดไว้ว่า ให้มีการควบคุมปริมาณการส่ง ICMP error message ไว้ด้วยโดยจะกำหนดว่าไม่ให้มีการส่ง ICMP error message ออกมาเร็วเกินค่าใดค่าหนึ่ง ซึ่งค่าดังกล่าวก็จะแตกต่างกันไปตามผู้ผลิตระบบปฏิบัติการแต่ละราย ดังนั้นการส่งแพ็กเก็ตเพื่อตรวจสอบไปยังพอร์ต UDP ของเครื่องปลายทาง หากว่าเครื่องปลายทางไม่ได้เปิดพอร์ตนั้นอยู่ก็จะส่ง ICMP พอร์ต unreachable ออกมา ซึ่งถ้าในการตรวจสอบได้มีการส่งแพ็กเก็ตออกมาเร็วเกินไปก็จะทำให้เครื่องปลายทางระงับการส่ง ICMP พอร์ต unreachable ตอบกลับมา ซึ่งจะมีผลให้การตรวจสอบผิดพลาดได้ ดังนั้นในการตรวจสอบพอร์ต UDP จึงจำเป็นต้องมีการหน่วงเวลาในการส่งแพ็กเก็ตไว้ด้วย ซึ่งในที่นี้ได้กำหนดไว้ที่ 1 พอร์ตต่อ 1 วินาที และผลจากการที่ต้องมีการหน่วงเวลานี้เองทำให้การตรวจสอบพอร์ต UDP ต้องใช้เวลามากพอสมควร ดังนั้นจำนวนพอร์ตที่จะตรวจสอบจึงไม่ควรมากจนเกินไป เช่นอาจกำหนดไว้ว่าให้ตรวจสอบไม่เกิน 60 พอร์ต ก็จะใช้เวลาในการตรวจสอบเท่ากับ 1 เครื่องต่อนาที ซึ่งโดยปกติแล้วเครื่องที่ใช้งานทั่ว ๆ ไปก็จะเปิดไว้เพียงไม่กี่พอร์ตเท่านั้น ดังนั้นการตรวจสอบก็ให้ระบุเฉพาะหมายเลขพอร์ตที่สนใจจะตรวจสอบก็เพียงพอแล้ว

3. ทำการตรวจสอบไปที่เครื่องต่าง ๆ เหล่านั้นว่าใช้ระบบปฏิบัติการอะไร

หลังจากที่ได้ทำการตรวจสอบพอร์ตเสร็จสิ้นแล้ว NetScan ก็จะเริ่มทำการตรวจสอบระบบปฏิบัติการของเครื่องนั้น ซึ่งจะทำโดยใช้วิธีการส่งแพ็กเก็ต ICMP รูปแบบต่าง ๆ ไปยังเครื่องที่ต้องการตรวจสอบแล้วรอรับการตอบกลับ จากนั้นก็ตรวจสอบรายละเอียดของแพ็กเก็ตที่ได้รับ

การตอบกลับมา เพื่อวิเคราะห์ว่าเครื่องดังกล่าวใช้ระบบปฏิบัติการชนิดใด เนื่องจากในระบบปฏิบัติการที่ต่างชนิดกัน แพ็กเก็ตที่ตอบกลับมาจะมีรายละเอียดที่แตกต่างกัน [19] ทำให้สามารถระบุได้ว่าเครื่องดังกล่าวเป็นระบบปฏิบัติการชนิดใด ซึ่งแพ็กเก็ตที่ถูกส่งออกไปจะประกอบไปด้วย ICMP จำนวน 4 แพ็กเก็ต และ UDP อีก 1 แพ็กเก็ตดังนี้

- ICMP echo request
- ICMP TimeStamp request
- ICMP AddressMask request
- ICMP Information request
- UDP พอร์ต 0

โดยแพ็กเก็ตที่คาดว่าจะตอบกลับมาจะมีดังนี้

- ICMP echo reply
- ICMP TimeStamp reply
- ICMP AddressMask request
- ICMP Information request
- ICMP พอร์ต unreachable

เมื่อได้รับการตอบกลับทั้งหมดแล้ว หรือว่ายังไม่ได้รับการตอบกลับทั้งหมดแต่ว่าเกินเวลาที่กำหนดไว้แล้ว NetScan ก็จะนำข้อมูลที่ได้รับมาพิจารณาเทียบกับฐานข้อมูล OS Signature เพื่อตรวจสอบว่าเป็นระบบปฏิบัติการชนิดใด ซึ่งตัวอย่างของ OS signature จะเป็นดังภาคผนวก ก.

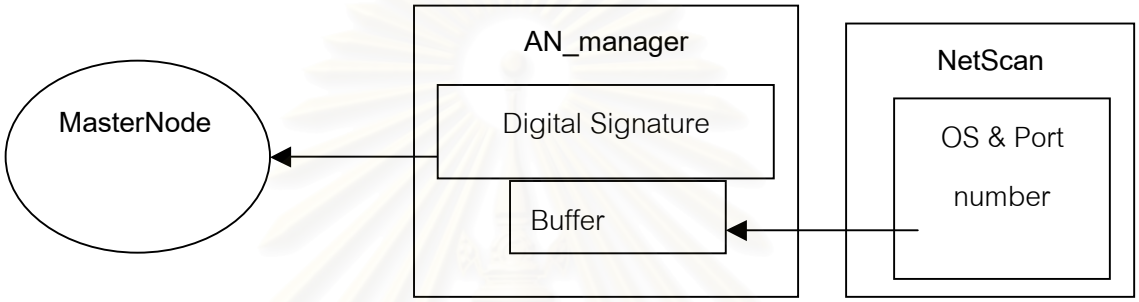
4. เมื่อตรวจสอบเสร็จแล้วก็จะส่งข้อมูลดังกล่าวไปยังมาสเตอร์โหนด

เมื่อตรวจสอบเสร็จแล้วว่าเครื่องใช้ระบบปฏิบัติการชนิดใด NetScan ก็จะเตรียมข้อมูลดังกล่าวเพื่อส่งให้กับมาสเตอร์โหนด เพื่อสอบถามว่าต้องการให้ทำอะไรต่อไป โดยข้อมูลดังกล่าวจะถูกสร้างขึ้นเป็นสัญญาณควบคุมเพื่อส่งไปยัง มาสเตอร์โหนด โดยมีโครงสร้างของข้อมูลดังรูปที่ 30

IP address	OS-ID	% match	จำนวนพอร์ต TCP	จำนวนพอร์ต UDP	TCP-พอร์ต	UDP-พอร์ต
32 bits	16 bits	16 bits	16 bits	16 bits	N bits	N bits

รูปที่ 30. ข้อมูลที่ส่งให้มาสเตอร์โหนดเพื่อแจ้งรายละเอียดของเครื่องที่อยู่ในเครือข่ายย่อย

เมื่อได้เตรียมข้อมูลสำหรับส่งไปยังมาสเตอร์โหนดเรียบร้อยแล้วโปรแกรมควบคุม NetScan จะไม่ได้เป็นผู้ส่งแพ็กเก็ตออกไปโดยตรง แต่ว่าจะนำข้อมูลที่ต้องการส่งไปวางไว้ใน Buffer ของโปรแกรมควบคุม AN_manager ดังรูปที่ 31 ซึ่งหลังจากนั้น AN_manager จะทำการตรวจสอบข้อมูลใน Buffer แล้วนำข้อมูลดังกล่าวมาเข้ารหัสเพื่อสร้าง ลายมือชื่ออิเล็กทรอนิกส์ โดยใช้ Private key ของตัวแอ็กทีฟโหนดเอง จากนั้นจึงจะส่งออกไปยังมาสเตอร์โหนด ซึ่งมาสเตอร์โหนดก็จะสามารถตรวจสอบได้ว่าสัญญาณควบคุมนี้ได้ถูกส่งออกมาจากแอ็กทีฟโหนดจริงๆ



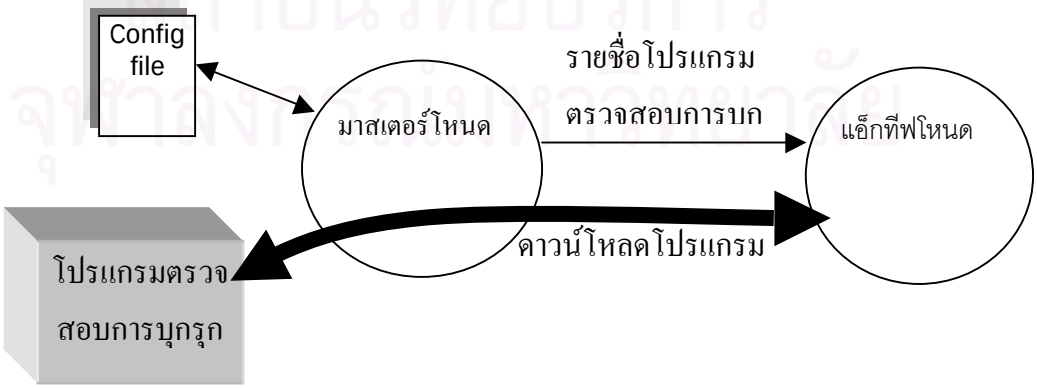
รูปที่ 31. การส่งสัญญาณควบคุมโดยการส่งผ่านโปรแกรม AN_manager

เมื่อมาสเตอร์โหนดได้รับข้อมูลจากโปรแกรมควบคุม NetScan ของแอ็กทีฟโหนดแล้ว ก็จะทำการตรวจสอบข้อมูลที่อยู่ใน configuration file บนมาสเตอร์โหนด เพื่อดูว่าระบบปฏิบัติการและบริการที่เปิดอยู่ตามข้อมูลที่ได้รับมาจากแอ็กทีฟโหนดนั้น จะต้องใช้โปรแกรมตรวจสอบการบุกรุกตัวใดบ้าง ซึ่งข้อมูลนี้จะสามารถกำหนดได้โดยผู้ดูแลเครือข่ายว่าต้องการให้ระบบปฏิบัติการหรือพอร์ตที่เปิดให้บริการใดใช้โปรแกรมตรวจสอบการบุกรุกอะไรบ้างดังรูปที่ 32

```
#
# configuration file
#
service_module [
        name “DetectWorm”
        osid    “*”
        tcp     “21,22,23,53”
        udp     “137,138,139”
]
service_module [
        name “DetectSlammer”
        osid    “91,92,93,94”
        tcp     “”
        udp     “1434”
]
```

รูปที่ 32. ตัวอย่างการกำหนดค่าใน Configuration file เพื่อกำหนดการเลือกใช้โปรแกรมควบคุม

เมื่อมาสเตอร์โหนดตรวจสอบเสร็จแล้วก็จะเตรียมการส่งสัญญาณควบคุม NetScan Reply ที่ประกอบไปด้วยข้อมูลชื่อของโปรแกรมตรวจสอบการบุกรุกที่อยู่ใน configuration file ที่ตรงกับข้อมูลของเครื่องที่ได้รับจากแอ็กทีฟโหนด และทำการสร้าง ลายมือชื่ออิเล็กทรอนิกส์ของสัญญาณควบคุมนั้นพร้อมระบุ key ID ของแอ็กทีฟโหนดปลายทาง จากนั้นจึงส่งออกไปยังแอ็กทีฟโหนด เพื่อบอกให้แอ็กทีฟโหนดทำการดาวน์โหลดโปรแกรมเหล่านั้นไปติดตั้ง ดังรูปที่ 33



รูปที่ 33. ขั้นตอนการติดตั้งโปรแกรมตรวจสอบการบุกรุกตามที่กำหนดไว้โดยผู้ดูแลเครือข่าย

เมื่อแอ็กทีฟโหนดทำการดาวน์โหลดโปรแกรมเสร็จ ก็เป็นอันเสร็จสิ้นกระบวนการทำงานทั้งหมดของ NetScan สำหรับเครื่อง 1 เครื่อง จากนั้น NetScan ก็จะคอยตรวจสอบต่อไปว่ามีเครื่องใหม่ติดตั้งเข้ามาในเครือข่ายย่อยอีกหรือไม่ ซึ่งถ้าพบ NetScan ก็จะเริ่มทำงานตามกระบวนการดังที่กล่าวมาแล้วทั้งหมดอีกครั้ง ดังนั้นโปรแกรมตรวจสอบการบุกรุกที่ถูกติดตั้งในแอ็กทีฟโหนดจะได้รับการปรับปรุงอยู่เสมอแม้ว่าเครื่องที่อยู่ในเครือข่ายย่อยมีการเปลี่ยนแปลงไป

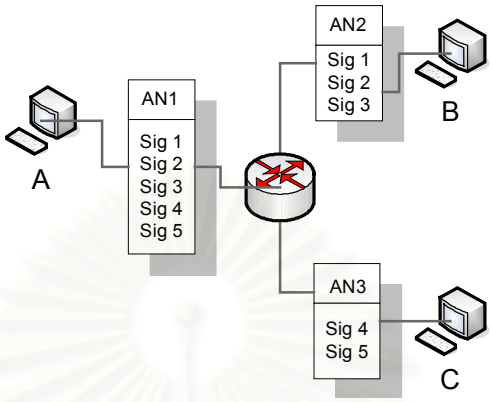
3.11 การตรวจสอบการบุกรุกแบบกระจาย

จากแนวคิดของวิทยานิพนธ์นี้ที่จะป้องกันการบุกรุกเครือข่ายโดยใช้วิธีการสกัดกั้นการบุกรุกไว้ในจุดที่เป็นเครือข่ายย่อย เพื่อจำกัดขอบเขตของการโจมตีให้เล็กที่สุด ทำให้ประสิทธิภาพโดยรวมของเครือข่ายได้รับผลกระทบน้อยที่สุดหากมีการบุกรุกเกิดขึ้น ดังนั้นโครงสร้างของระบบที่พัฒนาขึ้นจึงได้ใช้การติดตั้งแอ็กทีฟโหนดไว้ที่เครือข่ายย่อยทั้งหมด เพื่อทำหน้าที่เป็นตัวตรวจสอบและป้องกันการบุกรุกที่อาจจะเกิดขึ้น แอ็กทีฟโหนดแต่ละตัวจะทำหน้าที่ป้องกันเครือข่ายย่อยของตนเองด้วยการตรวจสอบข้อมูลที่วิ่งผ่านเข้าออกระหว่างเครือข่ายย่อยกับเครือข่ายหลัก ซึ่งข้อมูลเหล่านี้จะถูกตรวจสอบโดยโปรแกรมตรวจสอบการบุกรุกทั้งหมดที่ถูกติดตั้งไว้ในตัวแอ็กทีฟโหนด

ปัญหาที่สำคัญของการตรวจสอบและป้องกันการบุกรุกก็คือต้องใช้เวลาในการอ่านข้อมูลที่วิ่งผ่านทั้งหมด เพื่อค้นหาว่าข้อมูลนั้นจะเป็นการบุกรุกหรือไม่ ดังนั้นการตรวจสอบการบุกรุกจะส่งผลกระทบโดยตรงกับอัตราการส่งผ่านข้อมูลของตัวแอ็กทีฟโหนด ยิ่งถ้ามีจำนวนโปรแกรมที่ใช้ในการตรวจสอบการบุกรุกมากเท่าใด อัตราการส่งผ่านข้อมูลก็จะลดลงมากเท่านั้น แต่เนื่องจากว่าการตรวจสอบการบุกรุกของแอ็กทีฟโหนดแต่ละตัว มีจุดประสงค์เพื่อทำหน้าที่ป้องกันเครื่องที่อยู่ในเครือข่ายย่อยของตนเองเท่านั้น ดังนั้นการติดตั้งโปรแกรมตรวจสอบการบุกรุกไว้ในแอ็กทีฟโหนดจึงไม่จำเป็นต้องติดตั้งทั้งหมดทุกโปรแกรม แต่จะเลือกติดตั้งโปรแกรมตรวจสอบแตกต่างกันไปขึ้นอยู่กับว่าในเครือข่ายย่อยนั้นมีการใช้งานเครื่องประเภทใดอยู่บ้าง ซึ่งวิธีการนี้เป็นผลดีกับประสิทธิภาพในการส่งผ่านข้อมูลของแอ็กทีฟโหนด เนื่องจากการตรวจสอบการบุกรุกจะตรวจสอบเฉพาะการบุกรุกที่คาดว่าจะมีผลกระทบกับเครื่องที่อยู่ในเครือข่ายย่อยเท่านั้น

นอกจากนั้นการตรวจสอบข้อมูลที่วิ่งผ่านตัวแอ็กทีฟโหนดไม่จำเป็นที่จะต้องตรวจสอบข้อมูลที่วิ่งออกจากเครือข่ายย่อย เนื่องจากว่าหน้าที่ของแอ็กทีฟโหนดคือการตรวจสอบการบุกรุกที่มุ่งจะโจมตีเครื่องที่อยู่ในเครือข่ายย่อย ดังนั้นการตรวจสอบเฉพาะข้อมูลที่วิ่งเข้าสู่เครือข่ายย่อยก็จะเพียงพอในการที่จะป้องกันเครื่องในเครือข่ายย่อยได้ ซึ่งวิธีนี้จะส่งผลดีให้กับอัตราการส่งผ่านข้อมูลของแอ็กทีฟโหนดเป็นอย่างมาก อย่างไรก็ตามการตรวจสอบข้อมูลเพียงด้านเดียวเช่นนี้ ไม่ได้หมายความว่าข้อมูลที่วิ่งออกจากเครือข่ายย่อยจะไม่ถูกตรวจสอบ เพราะในที่สุดแล้วเมื่อข้อมูลที่

วิ่งออกจากแอ็กทีฟโหนดต้นทาง ก็จะวิ่งเข้าสู่เครื่องปลายทางซึ่งก็จะต้องวิ่งผ่านแอ็กทีฟโหนดที่ดูแลเครื่องปลายทางอยู่ดี ดังนั้นข้อมูลก็จะถูกตรวจสอบโดยแอ็กทีฟโหนดที่ดูแลเครื่องปลายทางอยู่



รูปที่ 34. ตัวอย่างการตรวจสอบการบุกรุกแบบกระจาย

ตัวอย่างการตรวจสอบการบุกรุกดังรูปที่ 34 ประกอบไปด้วยแอ็กทีฟโหนด 3 ตัว คือ AN1 AN2 และ AN3 ซึ่ง AN1 จะติดตั้งโปรแกรมตรวจสอบการบุกรุกจำนวน 5 ตัว ส่วน AN2 และ AN3 จะติดตั้งโปรแกรมตรวจสอบจำนวน 3 และ 2 ตัวตามลำดับ ในกรณีที่เครื่อง A ส่งแพ็กเก็ตไปให้เครื่อง B แพ็กเก็ตจะถูกตรวจสอบด้วย AN2 เพราะเป็นแพ็กเก็ตขาเข้าของ AN2 แต่จะไม่ถูกตรวจสอบด้วย AN1 เพราะเป็นแพ็กเก็ตขาออกของ AN1 ดังนั้นแพ็กเก็ตดังกล่าวจะถูกตรวจสอบด้วยโปรแกรมตรวจสอบการบุกรุกจำนวน 3 โปรแกรมคือ Sig1 Sig2 และ Sig3 ในขณะที่หากเครื่อง B ส่งแพ็กเก็ตกลับไปยังเครื่อง A ก็จะถูกตรวจสอบที่ AN1 ด้วยโปรแกรมตรวจสอบจำนวน 5 ตัว เท่ากับว่าแต่ละแพ็กเก็ตจะถูกตรวจสอบด้วยแอ็กทีฟโหนดเพียงแค่ 1 ตัวเท่านั้น ซึ่งวิธีการนี้นอกจากจะเป็นการกระจายภาระในการตรวจสอบของตัวแอ็กทีฟโหนดด้วยการแบ่งกันตรวจสอบเฉพาะแพ็กเก็ตขาเข้าของตนเองแล้ว ยังทำให้การตรวจสอบเป็นไปอย่างเหมาะสมไม่ได้ตรวจสอบการบุกรุกทุกรูปแบบที่มีอยู่ทั้งหมด แต่จะตรวจสอบเพียงการบุกรุกที่มุ่งโจมตีเครื่องที่อยู่ในเครือข่ายย่อยที่แอ็กทีฟโหนดต้องป้องกันเท่านั้น ซึ่งจะส่งผลดีกับอัตราการส่งผ่านข้อมูลโดยรวมภายในเครือข่าย

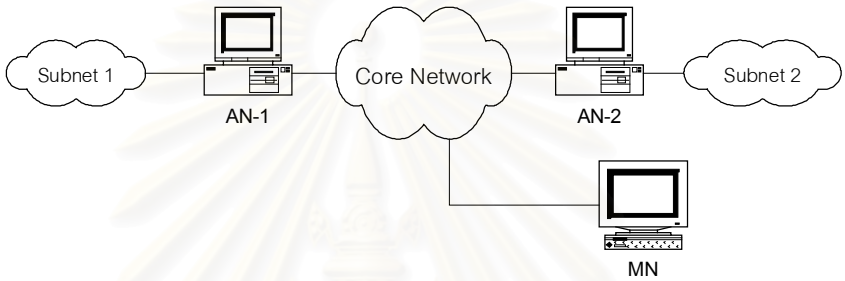
อย่างไรก็ตาม ในบางกรณีผู้ดูแลเครือข่ายก็อาจต้องการที่จะกำหนดให้แอ็กทีฟโหนดทำการตรวจสอบข้อมูลที่วิ่งออกจากเครือข่ายย่อยได้ เพื่อป้องกันการใช้งานเครือข่ายที่ไม่เหมาะสม ซึ่งอาจใช้โปรแกรมตรวจสอบข้อมูลเพื่อป้องกันการส่งข้อมูลที่ละเมิดนโยบายความปลอดภัยขององค์กรไม่ให้ส่งผ่านแอ็กทีฟโหนดออกมาได้ เพื่อป้องกันข้อมูลบางประเภทที่อาจส่งผลกระทบกับเครือข่ายหลักถ้าหากปล่อยให้ข้อมูลดังกล่าวหลุดออกจากเครือข่ายย่อย ยกตัวอย่างเช่นการตรวจสอบการส่งข้อมูลในปริมาณที่มากผิดปกติ ซึ่งอาจจะเกิดจากการที่เครื่องคอมพิวเตอร์ในเครือข่ายติดไวรัส ทำให้เครื่องดังกล่าวส่งแพ็กเก็ตออกมาเป็นจำนวนมากเพื่อทำการแพร่เชื้อไวรัส เป็นต้น

บทที่ 4

การทดสอบการทำงานของระบบ

4.1 ทดสอบการทำงานเบื้องต้นของระบบ

การทดสอบการทำงานเบื้องต้นของระบบ โดยติดตั้งแอ็กทีฟโหนดจำนวน 2 ตัวเข้าไปในเครือข่าย พร้อมทั้งเตรียมโปรแกรมควบคุม Log&Trace ไว้ในมาสเตอร์โหนด โดยเครือข่ายจะประกอบด้วยแอ็กทีฟโหนด AN-1 และ AN-2 ซึ่งทำหน้าที่เชื่อมระหว่างเครือข่ายหลักกับเครือข่ายย่อย เครือข่ายย่อย 1 และ เครือข่ายย่อย 2 ดังรูปที่ 35



รูปที่ 35. เครือข่ายที่สร้างขึ้นเพื่อใช้ในการทดลอง

เริ่มต้นการทำงานโดยแอ็กทีฟโหนดจะทำตามขั้นตอนก็คือไปขอลงทะเบียนกับมาสเตอร์โหนด จากนั้นมาสเตอร์โหนดก็จะตอบกลับมาพร้อมกับส่งรายชื่อโปรแกรมควบคุมพื้นฐานมาให้ ซึ่งในขณะนี้ก็จะมีเพียงโปรแกรม Log&Trace ซึ่งเมื่อแอ็กทีฟโหนดได้รับการตอบกลับ ก็จะเริ่มดาวน์โหลดโปรแกรมควบคุมตามรายชื่อที่ได้รับมา และติดตั้งเข้าไว้ในแอ็กทีฟโหนดทันที หลังจากติดตั้งเสร็จเรียบร้อยโปรแกรม Log&Trace ก็จะเริ่มบันทึกแพ็กเก็ตไปตามปกติ ซึ่งขนาดของหน่วยความจำที่จองไว้สำหรับบันทึกจะเท่ากับ 10,000 แพ็กเก็ต

จากนั้นก็เริ่มทดสอบการค้นหาที่มาของแพ็กเก็ต โดยการส่งแพ็กเก็ตตัวอย่าง ออกจากเครื่องที่อยู่ใน เครือข่ายย่อย 1 ไปยังเครื่องที่อยู่ใน เครือข่ายย่อย 2 หลังจากนั้นจึงทำการส่งคำร้องขอค้นหาที่มาของแพ็กเก็ต พร้อมระบุแพ็กเก็ตตัวอย่างที่ต้องการค้นหาลงไปด้วย โดยส่งคำร้องขอจาก AN-2 ไปยัง AN-1 ซึ่งในการทดสอบนี้ได้ทำการทดลองกับเครือข่ายที่ใช้งานจริง ซึ่งทำงานอยู่ในสภาวะการใช้งานเครือข่ายปกติ และจากผลการทดสอบพบว่า หลังจากที่ AN-1 ได้รับคำร้องขอค้นหาที่มาของแพ็กเก็ตจาก AN-2 แล้ว โปรแกรมควบคุม Log&Trace ของ AN-1 ก็สามารถค้นหาแพ็กเก็ตตามที่ระบุไว้ในคำขอได้พบในทันที

4.2 ทดสอบความสามารถของแอ็กทีฟโหนดในการติดตั้งโปรแกรมควบคุมโดยอัตโนมัติ

คุณสมบัติพิเศษของแอ็กทีฟโหนดคือสามารถเปลี่ยนแปลงโปรแกรมควบคุมการทำงานของตัวมันเองได้ ซึ่งด้วยคุณสมบัตินี้ทำให้เราสามารถเปลี่ยนแปลงโปรแกรมตรวจสอบการบุกรุกได้

อย่างสะดวกและสามารถทำได้โดยอัตโนมัติ ซึ่งในระบบนี้จะเก็บโปรแกรมตรวจสอบการบุกรุกไว้ในเครื่องมาสเตอร์โหนด เพื่อให้แอ็กทีฟโหนดเข้ามาดาวน์โหลดไปติดตั้งใช้งาน โดยที่ผู้ดูแลเครือข่ายจะสามารถกำหนดในมาสเตอร์โหนดได้ว่า จะให้แอ็กทีฟโหนดตัวใดติดตั้งโปรแกรมตรวจสอบอะไรบ้าง ซึ่งจะสามารถทำได้ 2 ลักษณะ คือ

- กำหนดให้เป็นค่าเริ่มต้น โดยที่แอ็กทีฟโหนดทุกตัวจะต้องติดตั้งโปรแกรมตรวจสอบนี้
- กำหนดเป็นค่าเฉพาะสำหรับระบบปฏิบัติการแต่ละชนิดหรือเฉพาะแต่ละพอร์ตที่เปิดให้บริการ

ในแบบแรกนั้นจะใช้สำหรับกำหนดให้แอ็กทีฟโหนดติดตั้งโปรแกรมที่จำเป็นต้องใช้เพื่อให้สามารถทำงานขั้นพื้นฐานได้อย่างถูกต้อง และใช้สำหรับผู้ดูแลเครือข่ายที่จะสามารถกำหนดรายชื่อโปรแกรมตรวจสอบพื้นฐาน ที่แอ็กทีฟโหนดทุกตัวต้องติดตั้งตามนโยบายความปลอดภัยขององค์กรได้ โดยการกำหนดในส่วนนี้จะให้ระบุชื่อโปรแกรมที่ต้องติดตั้งไว้ในแฟ้ม “DefaultMod” ส่วนในแบบที่สองนั้นจะใช้สำหรับกำหนดให้แอ็กทีฟโหนดแต่ละตัวติดตั้งโปรแกรมเฉพาะสำหรับแต่ละเครือข่ายย่อยได้ ซึ่งผู้ดูแลเครือข่ายสามารถกำหนดได้ว่า ระบบปฏิบัติการใดต้องใช้โปรแกรมตรวจสอบอะไร หรือหากมีการเปิดพอร์ตหมายเลขอะไรไว้ก็ให้ติดตั้งโปรแกรมตรวจสอบเฉพาะพอร์ตนั้น ซึ่งแอ็กทีฟโหนดจะสามารถรู้ได้โดยอัตโนมัติว่าในเครือข่ายย่อยที่ดูแลอยู่นั้นมีเครื่องที่ใช้ระบบปฏิบัติการชนิดใดและเปิดพอร์ตอะไรไว้ โดยใช้โปรแกรมควบคุม NetScan ในการตรวจสอบ และผู้ดูแลเครือข่ายจะต้องกำหนดรายชื่อโปรแกรมที่ต้องติดตั้งไว้ในแฟ้ม “service_module.conf” เพื่อให้แอ็กทีฟโหนดติดตั้งโปรแกรมตรวจสอบการบุกรุกให้สอดคล้องกับนโยบายความปลอดภัยสำหรับแต่ละเครือข่ายย่อยได้ ตัวอย่างการระบุรายชื่อโปรแกรมในแฟ้ม “DefaultMod” เป็นดังรูปที่ 36 ซึ่งได้กำหนดให้ติดตั้งโปรแกรม LogTrace และ NetScan ไว้ในแอ็กทีฟโหนดทุกตัว


```
###############
# DefaultMod file
###############
libLogTrace.so
libNetScan.so
```

รูปที่ 36. การระบุรายชื่อโปรแกรมควบคุมในแฟ้ม DefaultMod

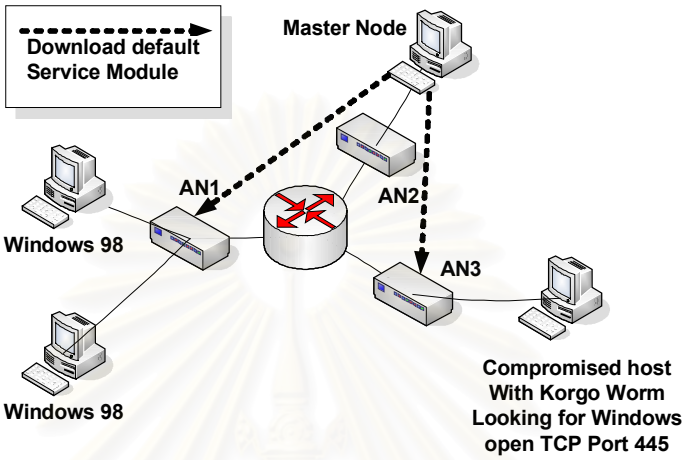
ตัวอย่างของแฟ้ม “service_module.conf” ในมาสเตอร์โหนด ซึ่งแอ็กทีฟโหนดจะติดตั้งโปรแกรมตามข้อมูลที่ได้จาก NetScan ซึ่งจากตัวอย่างในรูปที่ 37 โปรแกรมควบคุมตัวแรกได้กำหนดให้แอ็กทีฟโหนดที่ดูแลเครือข่ายย่อยที่มีระบบปฏิบัติการหมายเลข 95,96,97 ซึ่งก็คือ Windows XP SP1a, Windows XP SP1 และ Windows XP (ดูรายละเอียดได้จากภาคผนวก ก.) และเปิดพอร์ตหมายเลข 445 ไว้บนโปรโตคอล TCP ซึ่งก็คือบริการ LSASS ของ Windows ที่มีช่องโหว่อยู่ ให้ทำการติดตั้งโปรแกรมตรวจสอบ “LsassExploit” เพื่อป้องกันการโจมตีเครื่องในเครือข่ายย่อยโดยอาศัยช่องโหว่ดังกล่าว ส่วนในโปรแกรมควบคุมตัวที่สองนั้นให้ติดตั้งโปรแกรม “DetectWorm” สำหรับแอ็กทีฟโหนดทุกตัวโดยการระบุเครื่องหมาย “*” ไว้ ซึ่งจะหมายถึงให้ติดตั้งโปรแกรมควบคุมนี้ไม่ว่าจะเป็นระบบปฏิบัติการอะไร

```
##########################
#  Configuration file
##########################
service_module [
        name “LsassExploit”
        osid    “95,96,97”
        tcp     “”
        tcp     “445”
]
service_module [
        name “DetectWorm”
        osid    “*”
        tcp     “*”
        tcp     “*”
]
```

รูปที่ 37. ตัวอย่างการระบุรายชื่อโปรแกรมควบคุมในแฟ้ม service_module.conf

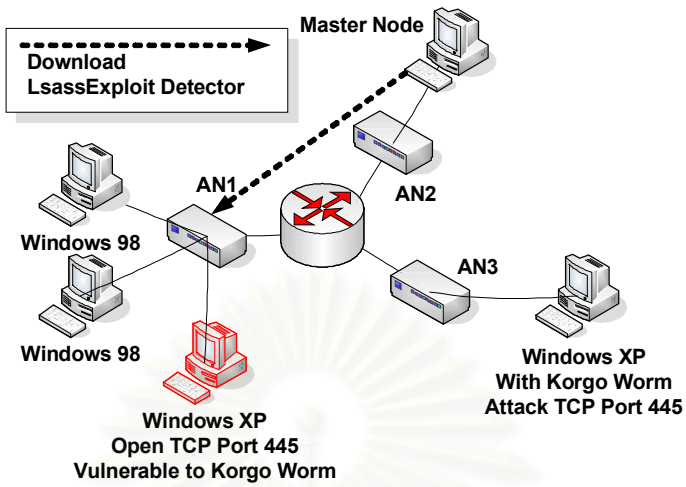
เครือข่ายที่ใช้ทดสอบประกอบไปด้วยแอ็กทีฟโหนด 3 ตัว AN1, AN2, AN3 และมาสเตอร์โหนดดังรูปที่ 38 การทดสอบเริ่มจากให้แอ็กทีฟโหนดเริ่มทำงาน ซึ่งแอ็กทีฟโหนดทุกตัวได้ทำการติดตั้งโปรแกรมควบคุมที่กำหนดไว้ในแฟ้ม DefaultMod และแอ็กทีฟโหนดแต่ละตัวได้ติดตั้งโปรแกรมตรวจสอบเฉพาะสำหรับแต่ละเครือข่ายย่อย เช่น AN1 ได้ทำการติดตั้งโปรแกรมตรวจสอบสำหรับ Windows 98 เป็นต้น โดยในตัวอย่างนี้จะมีเครื่องที่ถูกผู้บุกรุกยึดครองแล้วคือเครื่องที่อยู่ในเครือข่ายย่อยของ AN3 ซึ่งเครื่องดังกล่าวได้ทำการค้นหาเครื่องอื่นๆในเครือข่ายที่มีช่องโหว่

LSASS อยู่ตลอดเวลา โดยส่ง TCP SYN ออกไปยังหมายเลขไอพีปลายทางที่สุ่มขึ้นมา ทำให้มีปริมาณทราฟฟิคในเครือข่ายค่อนข้างสูง ซึ่งช่องโหว่นี้เป็นบริการที่ Windows จำเป็นต้องใช้งานซึ่งทำให้ไม่สามารถที่จะปิดพอร์ตนี้ทิ้งไปได้



รูปที่ 38. เครือข่ายที่ใช้ทดสอบการติดตั้งโปรแกรมควบคุมโดยอัตโนมัติ

จากนั้นได้ทำการติดตั้งเครื่องที่มีช่องโหว่ LSASS เพิ่มเข้าไปในเครือข่ายย่อยของ AN1 ดังรูป และทันทีที่เครื่องดังกล่าวเริ่มมีการส่งแพ็กเก็ตออกไปนอกเครือข่ายย่อย AN1 ก็ตรวจพบว่ามีเครื่องใหม่ถูกติดตั้งเข้ามาในเครือข่าย ก็สั่งให้โปรแกรม NetScan ใน AN1 เริ่มทำการตรวจสอบเครื่องดังกล่าวทันที โดยจะทำการตรวจสอบว่าเครื่องดังกล่าวใช้ระบบปฏิบัติการอะไร และมีการเปิดพอร์ตอะไรไว้บ้าง ซึ่งได้ตรวจพบว่าเป็น Windows XP และได้มีการเปิดพอร์ต 445 ไว้ จากนั้น AN1 ได้ส่งสัญญาณควบคุม NetScan ไปยังมาสเตอร์โหนดเพื่อขอรายชื่อโปรแกรมตรวจสอบที่ต้องติดตั้งเพิ่มเติม ซึ่งในการทดลองนี้ได้กำหนดไว้ในแฟ้ม service_module.conf แล้วว่าหากพบว่ามี Windows XP ที่เปิด พอร์ต TCP 445 ให้ทำการติดตั้งโปรแกรมตรวจสอบ LsassExploit จากนั้น มาสเตอร์โหนด ก็ได้ตอบกลับมาที่ AN1 พร้อมกับชื่อโปรแกรมตรวจสอบที่ต้องใช้ และเมื่อ AN1 ได้รับคำตอบแล้วก็เริ่มทำการเชื่อมต่อไปยังมาสเตอร์โหนดเพื่อขอดาวน์โหลดโปรแกรมตรวจสอบ LsassExploit มาและทำการติดตั้งและเริ่มใช้งานทันที



รูปที่ 39. แอ็กทีฟโหนดติดตั้งโปรแกรมโดยอัตโนมัติเมื่อตรวจพบระบบปฏิบัติการชนิดใหม่

ในการตรวจสอบว่าเครื่องมีการเปิดพอร์ตอะไรไว้บ้างนั้น จะไม่ได้ตรวจสอบหมดทุกพอร์ต เนื่องจากจำนวนพอร์ตที่เป็นไปได้มีถึง 65536 หมายเลข ซึ่งถ้าทำการตรวจสอบทั้งหมดจะต้องใช้เวลานานโดยเฉพาะการตรวจสอบ UDP พอร์ต ซึ่งต้องใช้เวลาในการตรวจสอบค่อนข้างนาน อีกทั้งยังเป็นการเพิ่มปริมาณทราฟฟิคในเครือข่ายโดยไม่จำเป็นด้วย ดังนั้นจึงทำการตรวจสอบเฉพาะหมายเลขพอร์ตที่สนใจเท่านั้น โดยผู้ดูแลเครือข่ายต้องระบุไว้ในโปรแกรม NetScan ว่าจะให้ตรวจสอบพอร์ตอะไรบ้าง และจากการทดสอบโดยใช้จำนวน UDP พอร์ต ที่ตรวจสอบจำนวนต่าง ๆ กัน ซึ่งได้ผลการทดสอบดังตารางที่ 2

ตารางที่ 2. แสดงเวลาที่ใช้ในการตรวจสอบบริการแบบ UDP ที่จำนวนพอร์ตต่าง ๆ

จำนวน UDP พอร์ต	10	20	30	40	50
เวลาที่ใช้ (วินาที)	23	43	63	83	103

หลังจากนั้นเครื่องที่ติด Korgo Worm ได้ทำการสุ่มหมายเลขไอพีมาจนพบเครื่องใหม่ที่เพิ่งติดตั้งเข้าไปใน AN1 และทำการโจมตีทันที แต่โปรแกรมตรวจสอบ LsassExploit ใน AN1 ได้ตรวจสอบพบและสามารถยับยั้งการโจมตีดังกล่าว พร้อมกับค้นหาและสั่งการให้ AN3 หยุดส่งข้อมูลที่มีเป้าหมายเป็นพอร์ต 445 ไม่ให้ออกมาอีก ซึ่งผลการทดลองพบว่า เครื่องในเครือข่ายย่อยที่ AN3 ดูแลอยู่สามารถส่งข้อมูลไปยังพอร์ตอื่น ๆ ได้ตามปกติยกเว้น TCP พอร์ต 445 ส่วนในเครือข่ายย่อยอื่น ๆ สามารถส่งข้อมูลได้ตามปกติทั้งหมดรวมทั้ง TCP พอร์ต 445 ด้วย

จากผลการทดสอบจะเห็นได้ว่าคุณสมบัติของเครือข่ายแบบแอ็กทีฟ ที่สามารถเปลี่ยนแปลงโปรแกรมควบคุมได้อย่างอิสระ ได้ช่วยให้แอ็กทีฟโหนดทำการเพิ่มเติมโปรแกรมตรวจสอบการบุกรุกโจมตีเครือข่ายได้อย่างสะดวกรวดเร็ว และเป็นไปโดยอัตโนมัติเมื่อมีการเปลี่ยนแปลงของเครื่องในเครือข่ายย่อย ซึ่งเป็นการเพิ่มขีดความสามารถในการตรวจสอบการบุกรุกของ แอ็กทีฟโหนด โดยอัตโนมัติโดยที่ผู้ดูแลเครือข่ายไม่ต้องทำอะไรเลย ซึ่งหากเป็นวิธีการทั่ว ๆไปที่ไม่ได้ใช้เครือข่ายแบบแอ็กทีฟจะไม่สามารถทำเช่นนี้ได้ โดยเฉพาะในเครือข่ายที่มีการขยายเพิ่มเติมหรือปรับเปลี่ยนอยู่ตลอดเวลา

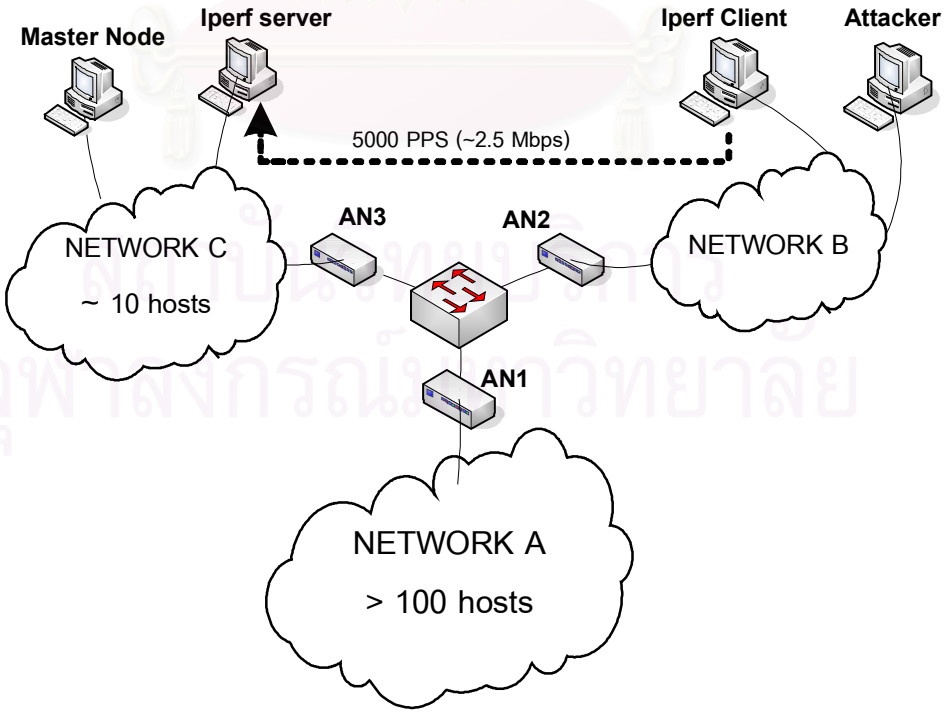
ผลการทดสอบประสิทธิภาพของแอ็กทีฟโหนดในด้านอัตราการส่งผ่านข้อมูล เมื่อติดตั้งโปรแกรมควบคุมในจำนวนต่าง ๆ กัน โดยที่โปรแกรมควบคุมที่ติดตั้งเข้าไปเพื่อทดสอบนี้จะเป็นโปรแกรมควบคุมที่ไม่ได้กระทำการใด ๆ กับข้อมูลที่ส่งผ่าน เพียงแค่อ่านข้อมูลเข้ามาแล้วส่งต่อไปยังโปรแกรมควบคุมตัวถัดไป ซึ่งผลการทดลองที่ได้เป็นไปดังตารางที่ 3 ซึ่งจากผลการทดลองที่ได้แสดงให้เห็นว่า หากติดตั้งโปรแกรมควบคุมในจำนวนที่มากกว่า 2000 โปรแกรมแล้ว จะทำให้อัตราการส่งผ่านข้อมูลของแอ็กทีฟโหนดลดลงมากกว่า 20 % และมีอัตราการลดลงอย่างรวดเร็ว

ตารางที่ 3. ผลการทดลองติดตั้งโปรแกรมควบคุมจำนวนต่าง ๆ เพื่อวัดอัตราการส่งผ่านข้อมูล

จำนวนโปรแกรมควบคุมที่ทดสอบ	อัตราการส่งผ่านข้อมูลของแอ็กทีฟโหนด (Mbits/sec)	อัตราการส่งผ่านข้อมูลคิดเป็นร้อยละ
0	7.9	100 %
100	7.9	100 %
500	7.85	99 %
1000	7.8	98 %
1500	7.1	89 %
2000	6.9	87 %
2500	4.8	60 %
3000	3.7	46 %
4000	2.5	31 %
5000	2.0	25 %

4.3 ทดสอบการทำงานของระบบในการป้องกันการบุกรุกเครือข่าย

การทดลองนี้จะเป็นการทดสอบความสามารถในการป้องกันการโจมตีเครือข่ายของระบบ โดยจะใช้หนอนเครือข่ายในการทดสอบการโจมตี ได้แก่หนอน Korgo [20], Sasser และ MS-Blaster ซึ่งจะโจมตีเครื่องที่ใช้ระบบปฏิบัติการวินโดวส์โดยอาศัยช่องโหว่แตกต่างกันสำหรับหนอนแต่ละตัว โดยเครือข่ายสำหรับทดลองจะเป็นดังรูปที่ 40 ซึ่งจะประกอบไปด้วยเครือข่ายจำนวน 3 เครือข่าย จึงต้องใช้แอ็กทีฟโหนดจำนวน 3 ตัวและมาสเตอร์โหนด 1 ตัว โดยในเครือข่าย A จะเป็นเครือข่ายภายในขององค์กรแห่งหนึ่งซึ่งจะมีขนาดใหญ่ที่สุดในจำนวน 3 เครือข่าย โดยจะมีเครื่องติดตั้งอยู่มากกว่า 100 เครื่องขึ้นไป จึงจำเป็นต้องมองเครือข่าย A เป็นเสมือนกับเครือข่ายภายนอก ดังนั้นจึงจะไม่ติดตั้งโปรแกรม NetScan ไว้บนตัว AN1 เพราะถ้าติดตั้งจะทำให้ใช้เวลาในการสำรวจทรัพยากรในเครือข่ายนานมาก ส่วนในเครือข่าย B จะเป็นเครือข่ายย่อยที่ติดตั้งเครื่องที่ใช้โจมตีคือ Attacker ซึ่งได้ติดตั้งหนอนทั้ง 3 ชนิดไว้ และมีเครื่อง Iperf Client ซึ่งจะทำหน้าที่สร้าง Traffic เพื่อทำให้เครือข่าย B และ C มีการใช้งานเครือข่ายในปริมาณมาก โดยจะส่ง Traffic ไปยังเครื่อง Iperf Server ซึ่งติดตั้งอยู่ในเครือข่าย C ด้วยแพ็กเก็ตขนาด 512 ไบต์ ด้วยอัตราเร็ว 5,000 แพ็กเก็ตต่อวินาที โดยจะมีมาสเตอร์โหนดติดตั้งอยู่บน Virtual Machine ของ AN3 จึงเสมือนกับอยู่ในเครือข่าย C ด้วย โดยเครือข่าย C จะเป็นเครือข่ายย่อยของหน่วยงานหนึ่งซึ่งมีเครื่องของผู้ใช้งานทั่วไปอยู่ประมาณ 10 เครื่องที่ใช้งานติดต่อระหว่างเครือข่าย C กับเครือข่าย A



รูปที่ 40. เครือข่ายสำหรับใช้ทดสอบการป้องกันการบุกรุก

คุณลักษณะของแอ็กทีฟโหนดทั้ง 3 ตัวจะเป็นดังนี้คือ AN1 ได้ใช้เครื่องที่มีหน่วยประมวลผล Pentium III 500 MHz หน่วยความจำ 128 เมกกะไบต์ ส่วน AN2 และ AN3 ได้ใช้เครื่องที่มีหน่วยประมวลผล Intel Celeron 2.4 GHz หน่วยความจำ 256 เมกกะไบต์ โดยที่แอ็กทีฟโหนดทั้ง 3 จะติดตั้งการ์ดเครือข่าย (Network Interface Card) ชนิด 10/100 Mbps จำนวนเครื่องละ 2 ชุด และติดตั้งระบบปฏิบัติการลินุกซ์ทะเลรุ่น 5.0 ใช้เคอร์เนล (Kernel) รุ่น 2.4.24

การทดลองเริ่มจากเมื่อแอ็กทีฟโหนดเริ่มทำงานก็จะดาวน์โหลดโปรแกรมตรวจสอบมาติดตั้งไว้ ซึ่งในการทดลองนี้ได้ใช้โปรแกรมตรวจสอบการแพร่กระจายตัวของหนอนเครือข่าย Korgo, MS-Blaster และ Sasser ซึ่งโปรแกรมตรวจสอบนี้จะถูกกำหนดในมาสเตอร์โหนดว่าให้ติดตั้งในเครือข่ายย่อยที่มีเครื่องที่ใช้ระบบปฏิบัติการวินโดวส์ 2000 และระบบปฏิบัติการวินโดวส์ XP ซึ่งในเครือข่ายทั้ง 3 จะมีเครื่องที่ใช้วินโดวส์ 2000 และ XP อยู่ ดังนั้นแอ็กทีฟโหนดทุกตัวจึงถูกติดตั้งด้วยโปรแกรมตรวจสอบดังกล่าว จากนั้นได้ทดลองสั่งให้หนอนเครือข่ายที่ติดตั้งบนเครื่อง Attacker เริ่มทำงาน โดยจะสั่งให้หนอนทำงานทีละตัวในการทดลองแต่ละครั้ง ซึ่งเมื่อหนอนเริ่มทำงานจะทำให้มีการส่งข้อมูลออกมาตลอดเวลาด้วยหมายเลขไอพีปลายทางที่ตัวหนอนสุ่มขึ้นมา โดยมีพอร์ตปลายทางหมายเลข 445 ซึ่งเป็นบริการที่หนอน Korgo และ Sasser จะใช้ในการบุกรุกเครื่องเป้าหมาย ส่วนหนอน MS-Blaster จะใช้พอร์ตหมายเลข 135 ซึ่งเครื่องที่ใช้ระบบปฏิบัติการวินโดวส์ทุกเครื่องจะเปิดบริการนี้ไว้อยู่แล้ว

ตารางที่ 4. ผลการทดลองการหยุดการแพร่กระจายตัวของหนอน

หนอนที่ใช้ทดลอง	ลำดับที่ทดลอง	เวลาที่ใช้สื่อสารกันระหว่าง AN3 และ AN2 (Second)	เวลาในการค้นหาแพ็กเก็ตและสกัดการบุกรุก (Microsecond)
Korgo	1	1.424	45
	2	1.523	409
	3	0.974	1092
Sasser	1	1.488	270
	2	0.510	232
	3	1.289	500
MS-Blaster	1	0.790	458
	2	1.058	532
	3	0.555	502

จากรูปที่ 40 จะเห็นว่า AN2 ซึ่งไม่ได้ตรวจสอบแพ็กเก็ตขาออกก็จะส่งผ่านแพ็กเก็ตที่หนอนสร้างขึ้นมาทั้งหมดออกไป และเมื่อหนอนสุ่มมาเจอหมายเลขไอพีของเครื่องที่อยู่ในเครือข่าย A แพ็กเก็ตก็จะถูกส่งผ่านมายัง AN1 ซึ่ง AN1 ได้ตรวจพบว่าแพ็กเก็ตนั้นเป็นการโจมตี จึงทิ้งแพ็กเก็ตนั้นไป ทำให้ไม่มีแพ็กเก็ตที่ใช้โจมตีของหนอนหลุดลอดเข้าไปในเครือข่าย A ได้ หลังจากนั้น AN1 จะส่งสัญญาณควบคุมไปยังแอ็กทีฟโหนดตัวอื่น ๆ พร้อมกับข้อมูลบางส่วนของแพ็กเก็ตที่หนอนใช้โจมตีดังกล่าว เพื่อขอให้แอ็กทีฟโหนดทุกตัวตรวจสอบดูว่าเป็นแพ็กเก็ตที่ได้ส่งออกมาหรือไม่ ซึ่ง AN2 ก็ได้ตรวจพบว่าตนเองเป็นผู้ส่งผ่านแพ็กเก็ตนี้ออกไป ดังนั้น AN2 ก็จะดำเนินการตามคำสั่งที่ระบุมาในสัญญาณควบคุม คือจะทำการสกัดกั้นการส่งผ่านข้อมูลที่มีหมายเลขพอร์ตปลายทางที่หนอนใช้ในการโจมตีโดยเพิ่มหมายเลขพอร์ตเข้าไปใน Block List ของ AN2 ซึ่งหลังจากนั้นหาก AN2 ตรวจพบแพ็กเก็ตที่มีหมายเลขพอร์ตปลายทางตรงกับใน Block List ก็จะทิ้งแพ็กเก็ตนั้นทันที ทำให้หนอนเครือข่ายไม่สามารถแพร่กระจายตัวไปยังเครือข่ายอื่นได้อีกต่อไป โดยผู้ดูแลเครือข่ายสามารถใช้มาสเตอร์โหนดเพื่อขอดูข้อมูลใน Block List ของ AN แต่ละตัวได้

จากผลการทดลองที่ได้ดังตารางที่ 4 ซึ่งเป็นการทดลองโจมตีจากเครือข่าย B ไปยังเครือข่ายอื่น โดยใช้หนอน 3 ตัว ทดลองตัวละ 3 ครั้ง ปรากฏว่า AN1 สามารถตรวจพบการบุกรุกได้ทุกครั้งที่มีการโจมตี โดยเวลาที่ใช้ตั้งแต่เริ่มส่งคำสั่งไปยังแอ็กทีฟโหนดจนกระทั่งแอ็กทีฟโหนดจัดการ Block เสร็จและตอบกลับมา จะใช้เวลาอยู่ในช่วง 0.5 ถึง 1.5 วินาที ในการทดลองทั้งหมด 9 ครั้ง อย่างไรก็ตามในช่วงระยะเวลาดังกล่าวจะไม่มีแพ็กเก็ตที่หนอนใช้โจมตีหลุดลอดเข้าไปในเครือข่ายอื่นได้เพราะจะถูกตรวจพบและสกัดกั้นทุกครั้ง เพียงแต่ในช่วงเวลาดังกล่าวอาจจะยังมีแพ็กเก็ตของหนอนออกมาจาก AN2 อยู่บ้างในช่วงก่อนที่ AN2 จะเพิ่มข้อมูลเข้าไปใน Block List

สำหรับเวลาที่ใช้ในการค้นหาแพ็กเก็ตใน Log ในการทดลองทั้งหมด 9 ครั้ง ซึ่งได้จับเวลาตั้งแต่ AN2 ได้รับคำสั่งจาก AN1 ให้ค้นหาแพ็กเก็ตจนกระทั่ง AN2 หาแพ็กเก็ตพบ จะใช้เวลาในช่วงประมาณ 45 ถึง 1000 Microsecond ของการทดลองทั้ง 9 ครั้ง ซึ่งเวลาที่แตกต่างกันนี้ก็จะขึ้นอยู่กับปริมาณข้อมูลที่บันทึกอยู่ใน Log Buffer เช่นหากแพ็กเก็ตถูกบันทึกไว้ที่ช่วงต้น ๆ ของ Log ก็จะค้นหาได้เร็ว แต่ถ้าถูกบันทึกไว้ที่ปลาย Log ก็จะใช้เวลาค้นหานานกว่า ซึ่งในการทดลองนี้ได้มีการสร้าง Traffic ขึ้นมาเพื่อให้เหมือนกับมีการใช้งานเครือข่ายปริมาณมาก ซึ่งจะทำให้ Log เต็มเร็วขึ้น เพราะถ้าหาก Log เต็มก็จะวนมาบันทึกทับข้อมูลเดิมซึ่งอาจทำให้เกิดการค้นหาแพ็กเก็ตไม่พบได้ โดยใช้การส่งแพ็กเก็ตในอัตราเร็ว 5000 แพ็กเก็ตต่อวินาทีออกไปยัง AN3 ซึ่งจากการทดลองแสดงให้เห็นว่า AN2 ยังคงสามารถค้นหาแพ็กเก็ตได้พบทุกครั้ง

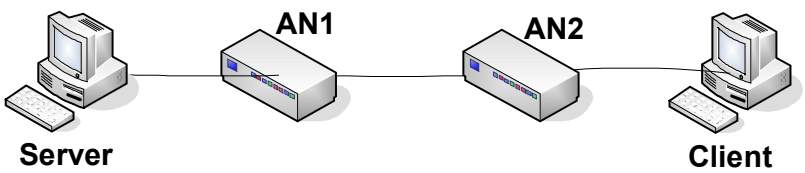
หลังจากที่ AN2 ได้ทำการหยุดส่งผ่านแพ็กเก็ตที่มีหมายเลขพอร์ตปลายทางที่ตรงกับข้อมูลใน Block List แล้ว จากการตรวจสอบก็พบว่าไม่มีแพ็กเก็ตดังกล่าวออกมาจาก AN2 อีกเลย ในขณะที่ข้อมูลอื่น ๆ ยังสามารถส่งผ่าน AN2 ได้ตามปกติ ทำให้ผู้ใช้งานที่อยู่ในเครือข่าย B สามารถติดต่อกับเครือข่ายอื่นได้ตามปกติยกเว้นพอร์ตที่ถูกระงับไว้เท่านั้น

4.4 การทดสอบอัตราการส่งผ่านข้อมูลของวิธีตรวจสอบการบุกรุกแบบกระจาย

โดยวัดอัตราความเร็วในการส่งผ่านข้อมูลระหว่างเครือข่ายย่อย 2 เครือข่าย ซึ่งข้อมูลจะถูกส่งผ่านแอ็กทีฟโหนดจำนวน 2 ตัวดังรูปที่ 41 โดย AN1 ใช้ Pentium III 500 MHz หน่วยความจำ 128 เมกกะไบต์ ส่วน AN2 ใช้ Intel Celeron 2.4 GHz หน่วยความจำ 256 เมกกะไบต์ โดยที่แอ็กทีฟโหนดทั้ง 2 ตัวได้ติดตั้งการ์ดเครือข่าย 10/100 Mbps ตัวละ 2 ชุด ใช้ระบบปฏิบัติการลินุกซ์ทะเลรุ่น 5.0 เคอร์เนลรุ่น 2.4.24 โดยแอ็กทีฟโหนดทั้ง 2 ตัวจะถูกติดตั้งโปรแกรมตรวจสอบสัญลักษณ์การบุกรุกสำหรับใช้ในการทดลอง โดยติดตั้งโปรแกรมตรวจสอบต่างกันในการทดลองแต่ละครั้ง โดยครั้งที่ 1 จะติดตั้งโปรแกรมตรวจสอบที่ AN1จำนวน 10 โปรแกรม (1 โปรแกรมจะมี 10 สัญลักษณ์) โดยจะตรวจสอบทั้งแพ็กเก็ตขาเข้าและขาออก ส่วน AN2 ไม่ติดตั้งโปรแกรมตรวจสอบ ในการทดลองครั้งที่ 2 ติดตั้งโปรแกรมตรวจสอบที่ AN1 จำนวน 10 โปรแกรมและที่ AN2 อีก 10 โปรแกรม โดยจะทำการตรวจสอบเฉพาะแพ็กเก็ตขาเข้า และการทดลองครั้งที่ 3 ติดตั้งโปรแกรมตรวจสอบที่ AN1 จำนวน 10 โปรแกรมและที่ AN2 จำนวน 5 โปรแกรม โดยโปรแกรมจะทำการตรวจสอบเฉพาะแพ็กเก็ตขาเข้า

ตารางที่ 5. ผลทดสอบอัตราการส่งผ่านข้อมูลของการตรวจสอบการบุกรุกแบบกระจาย

ลำดับ	จำนวนสัญลักษณ์ของ AN1	จำนวนสัญลักษณ์ของ AN2	อัตราการดาวน์โหลด (KBytes/S)	อัตราการอัพโหลด (KBytes/S)
1	100 (in/out)	0	322.6	322.0
2	100 (in)	100 (in)	583.8	402.0
3	100 (in)	50 (in)	707.2	393.6

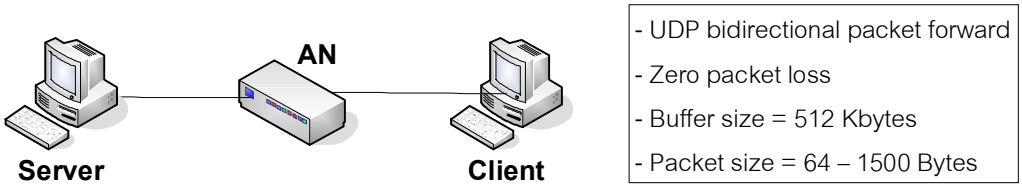


รูปที่ 41. เครือข่ายที่ใช้ทดลองอัตราการส่งผ่านข้อมูลของการตรวจสอบการบุกรุกแบบกระจาย

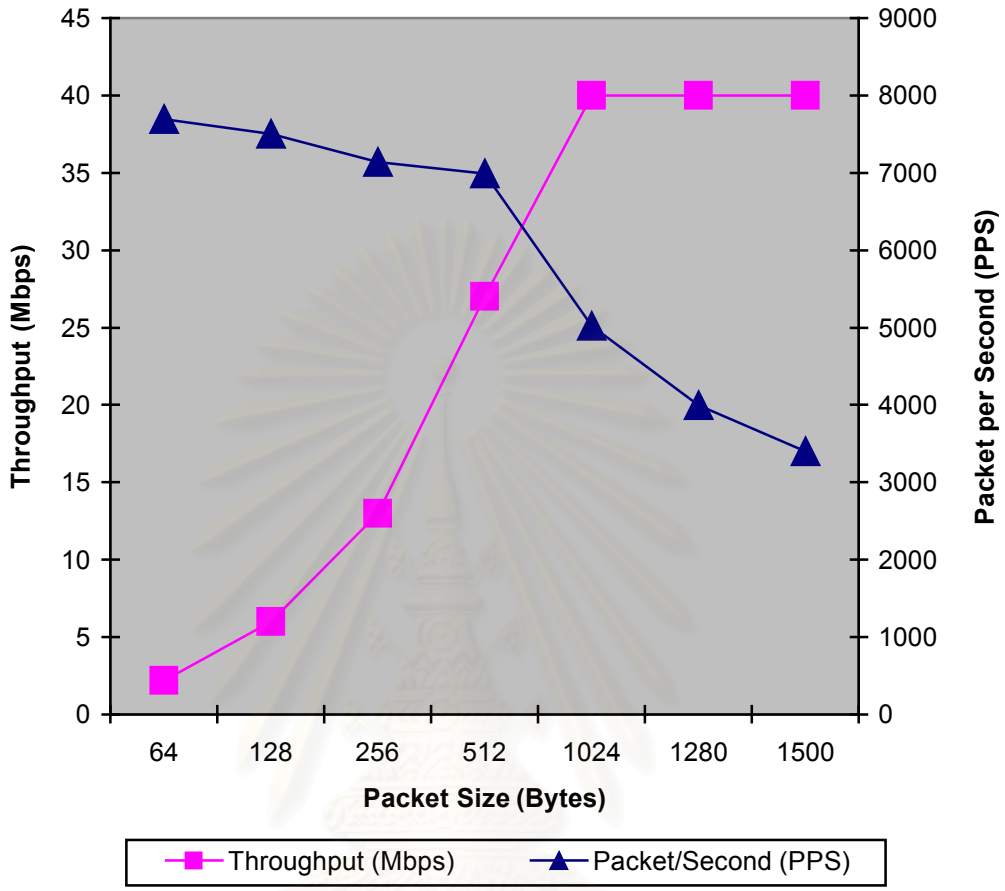
จากผลการทดลองที่ได้ในตารางที่ 5 แสดงให้เห็นว่า การทดลองครั้งที่ 1 ซึ่งเป็นวิธีการตรวจสอบสัญลักษณ์การบุกรุกแบบที่ใช้กันทั่ว ๆ ไป คือการตรวจสอบแพ็กเก็ตทั้งขาเข้าและขาออก จะใช้เวลามากกว่าการทดลองครั้งที่ 2 ซึ่งเป็นการตรวจสอบการบุกรุกแบบกระจาย ที่ใช้วิธีแบ่งหน้าที่กันตรวจสอบของอุปกรณ์ 2 ตัวทำให้ใช้เวลาในการตรวจสอบน้อยลง ส่งผลให้อัตราการส่งผ่านข้อมูลสูงขึ้นถึงร้อยละ 25 (คิดจากการอัพโหลดครั้งที่ 1 และ 2) ด้วยจำนวนโปรแกรมการตรวจสอบที่เท่ากัน (จำนวนสัญลักษณ์ในการตรวจสอบเท่ากัน) ส่วนในการทดลองครั้งที่สามนั้นแสดงให้เห็นถึงข้อดีของระบบนี้ ในการที่อุปกรณ์ตรวจสอบบางตัวไม่จำเป็นต้องติดตั้งโปรแกรมตรวจสอบทั้งหมด แต่จะติดตั้งเพียงบางโปรแกรมที่เหมาะสมกับเครื่องที่อยู่ในเครือข่ายย่อยเท่านั้น ซึ่งจะเห็นได้ว่าจะทำให้อัตราการส่งข้อมูล (ดาวน์โหลด) เร็วกว่าการทดลองครั้งแรกมากกว่า 2 เท่า

4.5 การทดสอบอัตราการส่งผ่านแพ็กเก็ตของแอ็กทีฟโหนด

การทดลองวัดอัตราส่งผ่านแพ็กเก็ตของแอ็กทีฟโหนดจะทำการเชื่อมต่อเครือข่ายเป็นดังรูปที่ 42 โดยแอ็กทีฟโหนดจะใช้เครื่องคอมพิวเตอร์ที่มีหน่วยประมวลผล Intel Celeron 2.4 GHz หน่วยความจำ 256 เมกะไบต์ เน็ตเวิร์คการ์ด (Network Interface Card) 10/100 Mbps 2 ชุด และติดตั้งระบบปฏิบัติการลินุกซ์ทะเลรุ่น 5.0 ใช้เคอร์เนล (Kernel) รุ่น 2.4.24 และทดสอบโดยได้ทำการส่งแพ็กเก็ต UDP ผ่านตัวแอ็กทีฟโหนดแบบสองทิศทาง โดยส่งจาก Client ไปยัง Server และจาก Server ไปยัง Client พร้อม ๆ กัน และการทดลองในแต่ละครั้งจะส่งแพ็กเก็ต UDP ที่ขนาดต่าง ๆ กัน ตั้งแต่แพ็กเก็ตขนาดเล็กที่สุดไปจนถึงแพ็กเก็ตขนาดใหญ่ที่สุดที่เครือข่ายประเภทอีเทอร์เน็ตจะส่งผ่านได้ คือตั้งแต่ 64 ไบต์ไปจนถึง 1500 ไบต์ และทำการวัดอัตราการส่งผ่านแพ็กเก็ตสูงสุดที่จะไม่ทำให้เกิดการสูญหายของแพ็กเก็ตเกิดขึ้น (Zero Loss) โดยใช้ Buffer ขนาด 512 กิโลไบต์ ซึ่งจากผลการทดลองที่ได้ดังรูปที่ 43 แสดงให้เห็นว่า แอ็กทีฟโหนดจะมีอัตราส่งผ่านแพ็กเก็ตสูงสุดที่ 7692 แพ็กเก็ตต่อวินาทีเมื่อใช้แพ็กเก็ตที่มีขนาดเล็กที่สุดคือ 64 ไบต์ ซึ่งจะได้อัตราส่งผ่านข้อมูลเท่ากับ 2.2 เมกกะบิตต่อวินาที และมีอัตราส่งผ่านแพ็กเก็ตต่ำที่สุดที่ 3396 แพ็กเก็ตต่อวินาทีเมื่อใช้แพ็กเก็ตที่มีขนาดใหญ่ที่สุดคือ 1500 ไบต์ ซึ่งจะได้อัตราส่งผ่านข้อมูลเท่ากับ 40 เมกกะบิตต่อวินาที ซึ่งจะเป็นอัตราการส่งผ่านข้อมูลที่สูงที่สุดแอ็กทีฟโหนดจะสามารถส่งผ่านได้



รูปที่ 42. เครือข่ายสำหรับทดลองวัดอัตราการส่งผ่านแพ็กเก็ตของแอ็กทีฟโหนด



รูปที่ 43. ผลการทดสอบอัตราการส่งผ่านแพ็กเก็ตของแอ็กทีฟโหนด

บทที่ 5

สรุปการวิจัย

5.1 สรุปแนวคิดและวิธีการในการออกแบบระบบและประโยชน์ที่ได้รับ

การพัฒนาระบบป้องกันการบุกรุกเครือข่ายในวิทยานิพนธ์นี้ ได้เริ่มต้นจากแนวคิดที่ต้องการจะป้องกันปัญหาที่เกิดขึ้นในเครือข่ายขนาดใหญ่ อันเนื่องมาจากการบุกรุกโจมตีที่เกิดขึ้นจากเครื่องที่อยู่ภายในเครือข่ายขององค์กรเอง เพราะในปัจจุบันการโจมตีในลักษณะนี้เกิดขึ้นมาก เนื่องจากการโจมตีที่มาจากเครือข่ายภายนอกนั้นจะถูกป้องกันไว้อย่างแน่นหนา แต่ภายในเครือข่ายขององค์กรเองกลับไม่ได้มีการป้องกันที่ดีเพียงพอ จึงได้พัฒนาระบบที่ใช้วิธีการจำกัดขอบเขตของการบุกรุก เพื่อกักกันผู้บุกรุกให้อยู่ในบริเวณที่จำกัดไม่ให้เข้าไปทำความเสียหายกับส่วนอื่น ๆ ของเครือข่ายได้ ซึ่งวิธีนี้ถึงแม้จะไม่ได้จัดการกับผู้บุกรุกโดยตรง แต่หากมองในภาพรวมของเครือข่ายทั้งหมดแล้วจะส่งผลดีกับผู้ใช้งานส่วนใหญ่ เพราะจะทำให้เครือข่ายส่วนใหญ่ยังคงสามารถใช้งานได้อย่างปกติ จะมีเพียงเครือข่ายย่อยที่ผู้บุกรุกอาศัยอยู่เท่านั้นที่จะไม่สามารถใช้งานได้

ระบบที่ได้พัฒนาขึ้นมานี้จะประกอบด้วยอุปกรณ์ป้องกันการบุกรุก ซึ่งจะถูกติดตั้งไว้ที่บริเวณขอบของเครือข่ายทั้งหมด และจะทำหน้าที่ในการตรวจสอบและสกัดกั้นการบุกรุก ซึ่งได้นำเทคโนโลยีเครือข่ายแอ็กทีฟมาใช้ในการพัฒนาเป็นอุปกรณ์ป้องกันการบุกรุก ซึ่งจะช่วยให้การดูแลและปรับปรุงอุปกรณ์ที่ถูกติดตั้งกระจายไปยังที่ต่าง ๆ ในเครือข่ายสามารถทำได้อย่างสะดวก และนอกจากนั้นยังได้นำเสนอแนวคิดในการตรวจสอบการบุกรุกแบบกระจาย ซึ่งเป็นการตรวจสอบแบบทิศทางเดียวโดยจะตรวจสอบเฉพาะข้อมูลที่วิ่งเข้าสู่เครือข่ายย่อยที่ต้องการป้องกัน และจะทำงานร่วมกับวิธีการค้นหาที่มาของแพ็กเก็ต โดยใช้การบันทึกข้อมูลบางส่วนของแพ็กเก็ตที่สามารถระบุลักษณะเฉพาะของแต่ละแพ็กเก็ตเก็บไว้ เพื่อใช้ในการค้นหาเครือข่ายย่อยต้นทางที่เป็นผู้ส่งแพ็กเก็ตออกมาในภายหลัง ซึ่งวิธีการตรวจสอบการบุกรุกแบบกระจายนี้จะเน้นในการตรวจสอบเฉพาะการโจมตีที่มุ่งไปยังเครื่องที่อยู่ในแต่ละเครือข่ายย่อยเท่านั้น ซึ่งทำได้โดยอาศัยการสำรวจทรัพยากรในเครือข่ายย่อยและเลือกติดตั้งโปรแกรมตรวจสอบการบุกรุกให้เหมาะสมกับทรัพยากรที่มีอยู่ ทำให้ลดภาระในการตรวจสอบที่ไม่จำเป็นลงได้

ในการพัฒนาระบบนี้ได้คำนึงถึงการนำไปใช้งานจริงในทางปฏิบัติ จึงได้พัฒนาให้อุปกรณ์ป้องกันการบุกรุกสามารถนำไปติดตั้งในบริเวณใดของเครือข่ายก็ได้โดยที่ไม่จำเป็นต้องทำการแก้ไขหรือเปลี่ยนแปลงใด ๆ กับเครือข่ายที่มีอยู่แล้ว และนอกจากนั้นยังคำนึงถึงว่าระบบนี้จะต้องเป็นระบบที่ใช้งบประมาณในการลงทุนไม่สูงมาก ซึ่งสามารถนำเครื่องคอมพิวเตอร์ที่มีประสิทธิภาพไม่

สูงมากมาสร้างเป็นอุปกรณ์ป้องกันการบุกรุกได้ นอกจากนั้นระบบนี้ได้ยังได้ใช้ระบบปฏิบัติการและซอฟท์แวร์สนับสนุนต่างที่เป็นแบบ Open Source ทั้งหมด ดังนั้นจะช่วยให้การนำระบบนี้ไปติดตั้งใช้งานจริงจะสามารถทำได้โดยใช้งบประมาณที่ไม่สูงมาก

จากที่ได้กล่าวมาแล้วทั้งหมดนั้น ได้แสดงให้เห็นถึงความสามารถและประสิทธิภาพในด้านต่าง ๆ รวมถึงข้อดีที่สำคัญของระบบนี้ ซึ่งในมุมมองของผู้ดูแลเครือข่ายจะเห็นได้ว่า ระบบนี้จะช่วยเพิ่มประสิทธิภาพในการตรวจสอบการบุกรุก และช่วยลดภาระในการดูแลรักษาความปลอดภัยของเครือข่ายได้เป็นอย่างมาก เนื่องจากผู้ดูแลเครือข่ายจะสามารถปรับเปลี่ยนโปรแกรมตรวจสอบการบุกรุกหรือโปรแกรมควบคุมการทำงานของอุปกรณ์ป้องกันการบุกรุกจำนวนมากได้อย่างสะดวก ซึ่งจะไม่ใช่การปรับเปลี่ยนที่เป็นเพียงแค่การเพิ่มเติมสัญลักษณ์การบุกรุกเพิ่มเข้าไปในอุปกรณ์เท่านั้น แต่จะสามารถปรับเปลี่ยนโปรแกรมหรือขั้นตอนวิธีที่ใช้ในการตรวจสอบได้ โดยการสั่งให้อุปกรณ์ป้องกันการบุกรุกทำการติดตั้งโปรแกรมตรวจสอบตัวใหม่เข้าไปได้ตามต้องการ โดยสั่งการผ่านตัวมาสเตอร์โหนด ซึ่งความสามารถในการปรับเปลี่ยนโปรแกรมของอุปกรณ์จำนวนมากได้อย่างสะดวกรวดเร็วจะมีความจำเป็นอย่างยิ่งในแง่ของการดูแลรักษาความปลอดภัยของระบบเครือข่าย เพราะการบุกรุกเครือข่ายมักจะมีการพัฒนาและเปลี่ยนแปลงรูปแบบหรือวิธีการที่ใช้โจมตีอยู่ตลอดเวลา ดังนั้นหากผู้ดูแลเครือข่ายพบว่าโปรแกรมตรวจสอบการบุกรุกมีข้อบกพร่องหรือจำเป็นต้องใช้โปรแกรมตรวจสอบการบุกรุกรูปแบบใหม่ ๆ ที่เพิ่งจะมีการคิดค้นขึ้นมา ก็จะสามารถทำได้อย่างสะดวกและรวดเร็วเป็นอย่างมาก

5.2 **สรุปผลการทดลอง**

ในการทดสอบการทำงานของระบบในลักษณะต่าง ๆ ได้ทำการทดลองกับอุปกรณ์และเครือข่ายจริง ซึ่งแสดงให้เห็นว่าระบบนี้จะสามารถนำไปใช้งานได้จริง ถึงแม้ว่าจะมีบางส่วนที่อาจจะต้องได้รับการปรับปรุงเพิ่มเติมให้ดีขึ้นเพื่อเพิ่มความสะดวกในการใช้งานในทางปฏิบัติ โดยสามารถสรุปผลการทดลองหลัก ๆ ได้ดังต่อไปนี้

1. จากผลการทดลองที่ได้ทดสอบระบบในเรื่องการติดตั้งโปรแกรมโดยอัตโนมัติ จะแสดงให้เห็นว่าอุปกรณ์ป้องกันการบุกรุกจะสามารถติดตั้งโปรแกรมป้องกันการบุกรุกต่าง ๆ ได้เองโดยอัตโนมัติตามทรัพยากรที่มีอยู่ในเครือข่ายย่อย ซึ่งจะช่วยให้สามารถติดตั้งโปรแกรมตรวจสอบการบุกรุกได้อย่างเหมาะสมสำหรับแต่ละเครือข่ายย่อย ส่งผลให้การตรวจสอบการบุกรุกเครือข่ายทำได้อย่างมีประสิทธิภาพ หรือหากแม้ว่าจะมีการติดตั้งเครื่องเพิ่มเข้ามาในเครือข่ายย่อย

แอ็กทีฟโหนดก็จะตรวจพบและจะทำการติดตั้งโปรแกรมตรวจสอบเพิ่มเองได้โดยอัตโนมัติ ซึ่งจะเป็นการช่วยลดภาระให้กับผู้ดูแลเครือข่ายได้เป็นอย่างมาก

2. การทดลองเรื่องการวัดอัตราการส่งผ่านข้อมูลของการตรวจสอบการบุกรุกแบบกระจายได้แสดงให้เห็นว่าการตรวจสอบสัญลักษณ์การบุกรุกด้วยวิธีนี้จะทำให้การส่งข้อมูลในแต่ละครั้งนั้น แพ็กเก็ตจะถูกตรวจสอบการบุกรุกด้วยสัญลักษณ์การบุกรุกได้อย่างเหมาะสมและเท่าที่จำเป็นเท่านั้น และนอกจากนั้นยังเป็นการตรวจสอบที่จะกระจายภาระในการตรวจสอบไปยังอุปกรณ์ป้องกันการบุกรุกที่อยู่ในแต่ละเครือข่ายย่อย ซึ่งจะทำให้อัตราการส่งผ่านข้อมูลของระบบเครือข่ายดีกว่าการตรวจสอบด้วยวิธีอื่น

3. การทดลองในเรื่องการทดสอบการป้องกันการบุกรุกเครือข่าย เป็นการทดสอบความสามารถของระบบในการป้องกันการบุกรุกโจมตีของหนอนเครือข่าย ซึ่งได้นำหนอนเครือข่ายที่มีการแพร่ระบาดอยู่ทั่วไปในอินเตอร์เนตมาใช้ในการทดลอง จากผลการทดลองที่ได้ทั้ง 9 ครั้ง ซึ่งได้จากการใช้หนอนเครือข่าย 3 ชนิดในการโจมตี ได้แสดงให้เห็นว่าระบบนี้สามารถสกัดกั้นการแพร่การจายตัวของหนอนเครือข่ายได้เป็นอย่างดี ซึ่งจากการทดลองทั้ง 9 ครั้งนั้น ระบบได้ใช้เวลาเพียงประมาณ 0.5 ถึง 1.5 วินาทีในการจำกัดขอบเขตของการแพร่กระจายตัวของหนอนหลังจากตรวจสอบพบการโจมตี ส่งผลให้เครือข่ายอื่น ๆ ไม่ได้รับผลกระทบและยังคงสามารถใช้งานเครือข่ายได้อย่างปกติ อย่างไรก็ตาม เวลาที่ใช้ไปในช่วง 0.5 ถึง 1.5 วินาทีนั้นไม่ได้หมายความว่าหนอนหรือผู้บุกรุกจะสามารถอาศัยช่วงเวลาดังกล่าวในการโจมตีเป้าหมายได้ เนื่องจากแพ็กเก็ตของการบุกรุกโจมตีจะถูกทิ้งไปตั้งแต่ตอนแรกที่ตรวจพบแล้ว แต่ช่วงเวลาดังกล่าวจะเป็นเวลาที่ใช้ในการติดต่อสื่อสารกันระหว่างแอ็กทีฟโหนด เพื่อทำการจำกัดขอบเขตของการบุกรุกที่ต้นกำเนิดของการบุกรุกเท่านั้น

4. การทดลองที่เกี่ยวกับประสิทธิภาพของระบบในด้านต่าง ๆ นั้น เช่นการทดสอบอัตราการส่งผ่านแพ็กเก็ตของแอ็กทีฟโหนด ซึ่งได้แสดงให้เห็นถึงอัตราการส่งผ่านแพ็กเก็ตสูงสุดของแพ็กเก็ตขนาดต่าง ๆ โดยใช้วิธีการทดสอบตามมาตรฐาน RFC 2544 หรือการทดสอบการติดตั้งโปรแกรมเข้าไปยังอุปกรณ์ป้องกันการบุกรุกในจำนวนมาก ๆ เพื่อวัดอัตราการส่งผ่านข้อมูล ซึ่งผลการทดลองเหล่านี้จะช่วยให้การนำระบบไปประยุกต์ใช้งานจริงสามารถนำค่าที่ได้จากการทดลองดังกล่าว ไปใช้คำนวณเพื่อออกแบบ และติดตั้งระบบได้อย่างมีประสิทธิภาพสูงสุด

5.3 สรุปปัญหาและอุปสรรคที่พบ

จากการพัฒนาและทดสอบการทำงานของระบบในลักษณะต่าง ๆ ทำให้ผู้วิจัยได้พบกับปัญหาและอุปสรรคบางอย่างที่ต้องคำนึงถึงดังนี้

1. อัตราการส่งผ่านข้อมูลของอุปกรณ์ที่ไม่สูงนัก หากเปรียบเทียบกับอุปกรณ์ตรวจสอบการบุกรุกที่มีขายในปัจจุบันเพราะระบบนี้จะทำงานด้วยซอฟท์แวร์ ดังนั้นจึงควรติดตั้งอุปกรณ์ของระบบนี้ไว้ในบริเวณที่เป็นขอบของเครือข่ายภายในเท่านั้น แต่ไม่เหมาะที่จะติดตั้งในจุดที่เชื่อมต่อเครือข่ายภายในกับภายนอกหรือในจุดที่มีการส่งผ่านข้อมูลในปริมาณมาก
2. ในการสำรวจทรัพยากรในเครือข่ายที่ได้นำวิธีการตรวจสอบระบบปฏิบัติการของ Xprobe2 มาใช้ ดังนั้นหาก Xprobe2 มีการปรับปรุงสัญลักษณ์ระบบปฏิบัติการเพื่อให้สามารถตรวจสอบระบบปฏิบัติการรุ่นใหม่ ๆ ได้แล้ว ผู้ดูแลเครือข่ายจะต้องนำสัญลักษณ์ดังกล่าวมาบรรจุลงไปในโปรแกรม NetScan ด้วยตนเอง
3. การสำรวจทรัพยากรในเครือข่ายในเรื่องการสำรวจพอร์ตที่เปิดไว้โดยเฉพาะประเภท UDP จะใช้เวลาในการสำรวจค่อนข้างนาน ดังนั้นจึงควรกำหนดพอร์ตที่ต้องการสำรวจเท่าที่จำเป็น โดยเน้นสำรวจหมายเลขพอร์ตที่คาดว่ามีการเปิดให้บริการ หรือพอร์ตที่เป็นบริการที่สำคัญที่ต้องได้รับการปกป้อง หรือพอร์ตที่เป็นบริการที่มีช่องโหว่ที่ผู้บุกรุกมักจะใช้ในการโจมตี เช่น บริการที่ติดอันดับ Top 20 ที่เปิดเผยโดย SANS ซึ่งหาดูได้จาก http://www.sans.org/top20
4. การจัดการกับกุญแจสาธารณะที่จำเป็นต้องติดตั้งไว้ในแอ็กทีฟโหนดทุกตัว ซึ่งจะต้องบรรจุกุญแจไว้ในแอ็กทีฟโหนดตั้งแต่ขั้นตอนเริ่มแรกของการติดตั้ง ดังนั้นหากมีความจำเป็นที่จะต้องเปลี่ยนหรือยกเลิกการใช้งานกุญแจดังกล่าว จะทำได้ไม่สะดวกนักเพราะผู้ดูแลเครือข่ายต้องนำกุญแจไปติดตั้งในแอ็กทีฟโหนดด้วยตนเอง
5. การติดตั้งโปรแกรมควบคุมการทำงานหลักของแอ็กทีฟโหนด “AN_manager” จะต้องติดตั้งไว้ในแอ็กทีฟโหนดตั้งแต่เริ่มแรกเช่นเดียวกันกับการบรรจุกุญแจสาธารณะ และไม่สามารถเปลี่ยนได้จากมาสเตอร์โหนด
6. ขั้นตอนและวิธีการในการติดตั้งระบบ โดยเฉพาะการสร้างแอ็กทีฟโหนดและมาสเตอร์โหนดค่อนข้างยุ่งยาก ซึ่งอาจเกิดความผิดพลาดและเกิดปัญหาขึ้นได้หากผู้ดูแลเครือข่ายไม่มีความชำนาญและคุ้นเคยกับระบบเพียงพอ

5.4 ข้อเสนอแนะ

จากปัญหาและอุปสรรคที่ได้กล่าวถึงข้างต้นนั้น แสดงให้เห็นว่าระบบนี้ยังมีจุดที่ควรจะต้องพัฒนาและปรับปรุงต่อไปอีกหากต้องการนำไปใช้งานจริงได้อย่างมีประสิทธิภาพ ซึ่งสามารถสรุปได้เป็นหัวข้อดังต่อไปนี้

1. อาจทำการพัฒนาแอ็กทีฟโหนดขึ้นโดยใช้ฮาร์ดแวร์ ซึ่งจะทำให้ระบบมีประสิทธิภาพที่ดีขึ้นและมีขนาดเล็กลงทำให้สามารถติดตั้งได้สะดวกยิ่งขึ้น
2. ทำการปรับปรุงหรือพัฒนาโปรแกรมที่ช่วยให้สามารถนำสัญลักษณ์ของระบบปฏิบัติการรุ่นใหม่มาติดตั้งในโปรแกรม NetScan ของแอ็กทีฟโหนดได้อย่างสะดวกยิ่งขึ้น
3. ปรับปรุงวิธีการจัดการกับกุญแจสาธารณะให้สามารถติดตั้งปรับเปลี่ยนกุญแจได้อย่างสะดวกโดยสามารถสั่งการได้ผ่านทางมาสเตอร์โหนด
4. ปรับปรุงให้แอ็กทีฟโหนดสามารถปรับเปลี่ยนโปรแกรมควบคุมหลัก “AN_manager” ได้จากมาสเตอร์โหนด
5. ปรับปรุงให้การสร้างแอ็กทีฟโหนดและมาสเตอร์โหนดสามารถทำได้ง่ายและสะดวกขึ้น

รายการอ้างอิง

1. Dan Sterne; et al. Active Network Based DDoS Defense. In Proceedings of the DARPA Active Networks Conference and Exposition (DANCE'02), pp. 193-203. San Francisco, CA, USA, May 2002.
2. Steve Bellovin; Marcus Leech; and Tom Taylor. ICMP traceback messages[Online]. 2001. Available from: http://www3.ietf.org/proceedings/01dec/I-D/draft-ietf-itrace-01.txt [2006,February 10]
3. Stefan Savage; David Wetherall; Anna R.Karlin; and Tom Anderson. Practical Network Support for IP Traceback. In Proceedings of ACM SIGCOMM Conference, pp. 295-306. Stockholm, Sweden, August 2000.
4. Tatsuya Baba; and Shigeyuki Matsuda. Tracing Network Attacks to Their Sources. IEEE Internet Computing 6(2) (March 2002): 20-26.
5. Wetherall, D. J.; Guttag, J.; and Tennenhouse, D. L. ANTS: A Toolkit for Building and Dynamically Deploying Network Protocols. In Proceedings of the IEEE Open Architectures and Network Programming (OPENARCH'98), pp. 117-129. San Francisco, CA, USA, April 1998.
6. Hicks M.; et al. PLANnet: An Active Internetwork. In Conference on Computer Communications (INFOCOM'99), pp. 1124–1133. New York, USA, March 1999.
7. Fuhrmann, T.; Harbaum, T.; Scholler, M.; and Zitterbart, M. AMnet 2.0: An Improved Architecture for Programmable Networks. In Proceedings of the 4th International Working Conference (IWAN2002), pp. 162-176. Switzerland, December 2002.
8. Robert Braden; Bob Lindell; Steven Berson; and Ted Faber. The ASP EE: An Active Network Execution Environment. In Proceedings of the DARPA Active Networks Conference and Exposition (DANCE'02), pp. 238. San Francisco, USA, May 2002.
9. Patrick Tullmann; Mike Hibler; and Jay Lepreau. Janos: A Java-Oriented OS for Active Network. IEEE Journal on Selected Areas of Communication 19(3) (March 2001): 501-510.
10. Hess, A.; Jung, M.; and Schaefer, G. FIDRAN: A Flexible Intrusion Detection and Response Framework for Active Networks. Proceedings of 8th IEEE Symposium on Computers and Communications (ISCC'2003), (July 2003): pp. 1219-1224.

11. William La Cholter; et al. IBAN: Intrusion Blocker Based on Active Networks. In Proceedings of the DARPA Active Networks Conference and Exposition (DANCE'02), pp. 182. San Francisco, CA, USA, May 2002.

12. Agilent Technologies. Journal of Internet Test Methodologies (JTC003: Mixed Packet Size Throughput)[Online]. 2005. Available from: http://advanced.comms.agilent.com/n2x/docs/journal/JTC_003.html, [2006,February 2004]

13. Sam Simpson. PGP DH vs. RSA FAQ[Online]. 1999. Available from: http://www.scramdisk.clara.net/pgpfaq.html, [2006,February 10]

14. Wiener, M.J. Performance Comparison of Public-Key Cryptosystems. RSA CryptoBytes 4(1) (Summer 1998): 1-5.

15. Wagner, R.; Kreizman, G.; and Pescatore J. Encryption Flaws Present No Immediate Security Risk[Online]. Gartner Research, 2004. http://www4.gartner.com/resources/122400/122460/encryption_flaw.pdf, [2006 February 10]

16. CERT Coordination Center. CERT Advisory CA-2003-04, MS-SQL Server Worm[Online]. 2003. Available from: http://www.cert.org/advisories/CA-2003-04.html, [2006,February 10]

17. Microsoft. Malicious Software Encyclopedia: Win32/Sasser. 2005. Available from: http://www.microsoft.com/security/encyclopedia/details.aspx?name=Win32%2fSasser, [2006,February 10]

18. Baker, F. RFC-1812: Requirements for IP Version 4 Routers[Online]. Cisco System. 1995. Available from: http://www.ietf.org/rfc/rfc1812.txt?number=1812, [2006,February 10]

19. Ofir arkin. A remote active OS fingerprinting tool using ICMP[Online]. 2002. Available from: http://www.sys-security.com/archive/articles/login.pdf. [2006,February 10]

20. John H. Sawyer. An In-Depth Analysis of the Korgo.P Worm[Online]. GIAC Certified Incident Handler. 2004. Available from: http://www.giac.org/certified_professionals/practicals/gcih/0631.php, [2006,February 10]

ภาคผนวก

ภาคผนวก ก
ตัวอย่างของ OS Signature

OS_ID	icmp_echo_code	icmp_echo_ip_id	icmp_echo_tos_bits	icmp_echo_df_bit	icmp_echo_reply_ttl	icmp_timestamp_reply	icmp_timestamp_reply_ttl	icmp_timestamp_reply_ip_id	icmp_addrmask_reply	icmp_addrmask_reply_ttl	icmp_addrmask_reply_ip_id	icmp_info_reply	icmp_info_reply_ttl	icmp_info_reply_ip_id	icmp_unreach_echoed_dtsize	icmp_unreach_reply_ttl	icmp_unreach_precedence_bits	icmp_unreach_df_bit	icmp_unreach_ip_id	icmp_unreach_echoed_udp_cksum	icmp_unreach_echoed_ip_cksum	icmp_unreach_echoed_ip_id	icmp_unreach_echoed_total_len	icmp_unreach_echoed_3bit_flags
AIX 5.1	3	3	3	1	255	1	255	3	0	255	3	1	255	3	8	255	0	1	3	0	2	5	20	5
AIX 4.3.3	3	3	3	1	255	1	255	3	0	255	3	1	255	3	8	255	0	1	3	0	2	5	20	5
Cisco IOS 12.2	3	4	3	1	255	1	255	4	0	255	4	1	255	4	8	255	192	0	3	5	5	5	5	5
Cisco IOS 12.0	3	4	3	1	255	1	255	4	0	255	4	1	255	4	8	255	192	0	3	5	5	5	5	5
Cisco IOS 11.3	3	4	3	1	255	1	255	4	0	255	4	1	255	4	8	255	192	0	3	5	5	5	5	5
Cisco IOS 11.2	3	4	3	1	255	1	255	4	0	255	4	1	255	4	8	255	0	0	3	5	5	5	5	5
Cisco IOS 11.1	3	4	3	1	255	1	255	4	0	255	4	1	255	4	8	255	0	0	3	5	5	5	5	5
Foundry Networks Device SW V.07.5.04T53	3	3	3	0	64	0	64	3	1	64	3	0	64	3	8	64	0	0	3	5	5	5	5	5
Foundry Networks Device SW V.07.5.05KT53	3	3	3	0	64	0	64	3	1	64	3	0	64	3	8	64	0	0	3	5	5	5	5	5
Foundry Networks Device SW V.07.6.01BT51	3	3	3	0	64	0	64	3	1	64	3	0	64	3	8	64	0	0	3	5	5	5	5	5
FreeBSD 5.1	3	3	3	1	64	1	64	3	0	64	3	0	64	3	8	64	0	1	3	0	5	5	5	5
FreeBSD 5.0	3	3	3	1	64	1	64	3	0	64	3	0	64	3	8	64	0	1	3	0	5	5	5	5
FreeBSD 4.8	3	3	3	1	64	1	64	3	0	64	3	0	64	3	8	64	0	1	3	0	5	5	5	5
FreeBSD 4.7	3	3	3	1	64	1	64	3	0	64	3	0	64	3	8	64	0	1	3	0	5	5	5	5
FreeBSD 4.6.2	3	3	3	1	64	1	64	3	0	64	3	0	64	3	8	64	0	1	3	0	5	5	5	5
FreeBSD 4.6	3	3	3	1	64	1	64	3	0	64	3	0	64	3	8	64	0	1	3	0	5	5	5	5
FreeBSD 4.5	3	3	3	1	64	1	64	3	0	64	3	0	64	3	8	64	0	1	3	0	5	5	5	5
FreeBSD 4.4	3	3	3	1	64	1	64	3	0	64	3	0	64	3	8	64	0	1	3	0	5	5	5	5
FreeBSD 4.3	3	3	3	1	255	1	255	3	0	255	3	0	255	3	8	255	0	1	3	0	5	5	5	5
FreeBSD 4.2	3	3	3	1	255	1	255	3	0	255	3	0	255	3	8	255	0	1	3	0	5	5	5	5
FreeBSD 4.1.1	3	3	3	1	255	1	255	3	0	255	3	0	255	3	8	255	0	1	3	0	5	5	5	5
FreeBSD 4.0	3	3	3	1	255	1	255	3	0	255	3	0	255	3	8	255	0	1	3	0	2	6	5	6
FreeBSD 3.5.1	3	3	3	1	255	1	255	3	0	255	3	0	255	3	8	255	0	1	3	0	2	6	5	6
FreeBSD 3.4	3	3	3	1	255	1	255	3	0	255	3	0	255	3	8	255	0	1	3	0	2	6	5	6
FreeBSD 3.3	3	3	3	1	255	1	255	3	0	255	3	0	255	3	8	255	0	1	3	0	2	6	5	6
FreeBSD 3.2	3	3	3	1	255	1	255	3	0	255	3	0	255	3	8	255	0	1	3	0	2	6	5	6
FreeBSD 3.1	3	3	3	1	255	1	255	3	0	255	3	0	255	3	8	255	0	1	3	0	2	6	5	6
FreeBSD 2.2.8	3	3	3	1	255	1	255	3	0	255	3	0	255	3	8	255	0	1	3	0	2	6	5	6
FreeBSD 2.2.7	3	3	3	1	255	1	255	3	0	255	3	0	255	3	8	255	0	1	3	0	2	6	5	6
HP UX 11.0i	3	3	3	1	255	0	255	3	0	255	3	0	255	3	64	255	0	1	3	5	5	5	5	5
HP UX 11.0	3	3	3	1	255	0	255	3	1	255	3	0	255	3	64	255	0	1	3	5	5	5	5	5
HP JetDirect ROM A.03.17 EEPROM A.04.09	0	3	0	0	60	0	60	3	0	60	3	0	60	3	8	60	0	0	3	5	0	5	5	5
HP JetDirect ROM A.05.03 EEPROM A.05.05	0	3	0	0	60	0	60	3	0	60	3	0	60	3	8	60	0	0	3	5	0	5	5	5
HP JetDirect ROM G.05.34 EEPROM G.05.35	0	3	0	0	60	0	60	3	0	60	3	0	60	3	8	60	0	0	3	5	0	5	5	5
HP JetDirect ROM G.07.02 EEPROM G.07.20	0	3	0	0	60	0	60	3	0	60	3	0	60	3	8	60	0	0	3	5	5	5	5	5

OS_ID	icmp_echo_code	icmp_echo_ip_id	icmp_echo_tos_bits	icmp_echo_df_bit	icmp_echo_reply_ttl	icmp_timestamp_reply	icmp_timestamp_reply_ttl	icmp_timestamp_reply_ip_id	icmp_addrmask_reply	icmp_addrmask_reply_ttl	icmp_addrmask_reply_ip_id	icmp_info_reply	icmp_info_reply_ttl	icmp_info_reply_ip_id	icmp_unreach_echoed_dtsize	icmp_unreach_reply_ttl	icmp_unreach_precedence_bits	icmp_unreach_df_bit	icmp_unreach_ip_id	icmp_unreach_echoed_udp_cksum	icmp_unreach_echoed_ip_cksum	icmp_unreach_echoed_ip_id	icmp_unreach_echoed_total_len	icmp_unreach_echoed_3bit_flags
HP JetDirect ROM G.07.19 EEPROM G.07.20	0	3	0	0	60	0	60	3	0	60	3	0	60	3	8	60	0	0	3	5	5	5	5	5
HP JetDirect ROM G.07.19 EEPROM G.08.03	0	3	0	0	60	0	60	3	0	60	3	0	60	3	8	60	0	0	3	5	5	5	5	5
HP JetDirect ROM G.08.08 EEPROM G.08.04	0	3	0	0	60	0	60	3	0	60	3	0	60	3	8	60	0	0	3	5	5	5	5	5
HP JetDirect ROM G.08.21 EEPROM G.08.21	0	3	0	0	60	0	60	3	0	60	3	0	60	3	8	60	0	0	3	5	5	5	5	5
HP JetDirect ROM H.07.15 EEPROM H.08.20	0	3	0	0	60	0	60	3	0	60	3	0	60	3	8	60	0	0	3	5	5	5	5	5
HP JetDirect ROM L.20.07 EEPROM L.20.24	3	3	3	1	64	1	64	3	0	64	3	0	64	3	8	64	0	1	3	0	0	6	5	6
Linux Kernel 2.4.21	3	3	3	0	64	1	64	3	0	64	3	0	64	3	65	64	192	0	3	5	5	5	5	5
Linux Kernel 2.4.20	3	3	3	0	64	1	64	3	0	64	3	0	64	3	65	64	192	0	3	5	5	5	5	5
Linux Kernel 2.4.19	3	3	3	0	64	1	64	3	0	64	3	0	64	3	65	64	192	0	3	5	5	5	5	5
Linux Kernel 2.4.18	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.4.17	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.4.16	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.4.15	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.4.14	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.4.13	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.4.12	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.4.11	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.4.10	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.4.9	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.4.8	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.4.7	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.4.6	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.4.5	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.4.4	3	0	3	1	255	1	255	0	0	255	0	0	255	0	65	255	192	1	0	5	5	5	5	5
Linux Kernel 2.4.3	3	0	3	1	255	1	255	0	0	255	0	0	255	0	65	255	192	1	0	5	5	5	5	5
Linux Kernel 2.4.2	3	0	3	1	255	1	255	0	0	255	0	0	255	0	65	255	192	1	0	5	5	5	5	5
Linux Kernel 2.4.1	3	0	3	1	255	1	255	0	0	255	0	0	255	0	65	255	192	1	0	5	5	5	5	5
Linux Kernel 2.4.0	3	0	3	1	255	1	255	0	0	255	0	0	255	0	65	255	192	1	0	5	5	5	5	5
Linux Kernel 2.2.25	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.2.24	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.2.23	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.2.22	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.2.21	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.2.20	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.2.19	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.2.18	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.2.17	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.2.16	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.2.15	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.2.14	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.2.13	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.2.12	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.2.11	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.2.10	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5

OS_ID	icmp_echo_code	icmp_echo_ip_id	icmp_echo_tos_bits	icmp_echo_df_bit	icmp_echo_reply_ttl	icmp_timestamp_reply	icmp_timestamp_reply_ttl	icmp_timestamp_reply_ip_id	icmp_addrmask_reply	icmp_addrmask_reply_ttl	icmp_addrmask_reply_ip_id	icmp_info_reply	icmp_info_reply_ttl	icmp_info_reply_ip_id	icmp_unreach_echoed_dtsize	icmp_unreach_reply_ttl	icmp_unreach_precedence_bits	icmp_unreach_df_bit	icmp_unreach_ip_id	icmp_unreach_echoed_udp_cksum	icmp_unreach_echoed_ip_cksum	icmp_unreach_echoed_ip_id	icmp_unreach_echoed_total_len	icmp_unreach_echoed_3bit_flags
Linux Kernel 2.2.9	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.2.8	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.2.7	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.2.6	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.2.5	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.2.4	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.2.3	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.2.2	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.2.1	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.2.0	3	3	3	0	255	1	255	3	0	255	3	0	255	3	65	255	192	0	3	5	5	5	5	5
Linux Kernel 2.0.36	3	3	3	0	64	1	64	3	0	64	3	0	64	3	65	64	192	0	3	5	5	5	5	5
Linux Kernel 2.0.34	3	3	3	0	64	1	64	3	0	64	3	0	64	3	65	64	192	0	3	5	5	5	5	5
Linux Kernel 2.0.30	3	3	3	0	64	1	64	3	0	64	3	0	64	3	65	64	192	0	3	5	5	5	5	5
Microsoft Windows 2003 Server Enterprise Edition	0	3	0	1	128	1	128	3	0	128	3	0	128	3	65	128	0	0	3	5	5	5	5	5
Microsoft Windows 2003 Server Standard Edition	0	3	0	1	128	1	128	3	0	128	3	0	128	3	65	128	0	0	3	5	5	5	5	5
Microsoft Windows XP SP1a	0	3	0	1	128	1	128	3	0	128	3	0	128	3	8	128	0	0	3	5	5	5	5	5
Microsoft Windows XP SP1	0	3	0	1	128	1	128	3	0	128	3	0	128	3	8	128	0	0	3	5	5	5	5	5
Microsoft Windows XP	0	3	0	1	128	1	128	3	0	128	3	0	128	3	8	128	0	0	3	5	5	5	5	5
Microsoft Windows 2000 Server Service Pack 4	0	3	0	1	128	1	128	3	0	128	3	0	128	3	8	128	0	0	3	5	5	5	5	5
Microsoft Windows 2000 Server Service Pack 3	0	3	0	1	128	1	128	3	0	128	3	0	128	3	8	128	0	0	3	5	5	5	5	5
Microsoft Windows 2000 Server Service Pack 2	0	3	0	1	128	1	128	3	0	128	3	0	128	3	8	128	0	0	3	5	5	5	5	5
Microsoft Windows 2000 Server Service Pack 1	0	3	0	1	128	1	128	3	0	128	3	0	128	3	8	128	0	0	3	5	5	5	5	5
Microsoft Windows 2000 Server	0	3	0	1	128	1	128	3	0	128	3	0	128	3	8	128	0	0	3	5	5	5	5	5
Microsoft Windows 2000 Workstation SP4	0	3	0	1	128	1	128	3	0	128	3	0	128	3	8	128	0	0	3	5	5	5	5	5
Microsoft Windows 2000 Workstation SP3	0	3	0	1	128	1	128	3	0	128	3	0	128	3	8	128	0	0	3	5	5	5	5	5
Microsoft Windows 2000 Workstation SP2	0	3	0	1	128	1	128	3	0	128	3	0	128	3	8	128	0	0	3	5	5	5	5	5
Microsoft Windows 2000 Workstation SP1	0	3	0	1	128	1	128	3	0	128	3	0	128	3	8	128	0	0	3	5	5	5	5	5
Microsoft Windows 2000 Workstation	0	3	0	1	128	1	128	3	0	128	3	0	128	3	8	128	0	0	3	5	5	5	5	5
Microsoft Windows Millennium Edition (ME)	0	3	3	1	128	1	128	3	0	128	3	0	128	3	8	128	0	0	3	5	5	5	5	5

OS_ID	icmp_echo_code	icmp_echo_ip_id	icmp_echo_tos_bits	icmp_echo_df_bit	icmp_echo_reply_ttl	icmp_timestamp_reply	icmp_timestamp_reply_ttl	icmp_timestamp_reply_ip_id	icmp_addrmask_reply	icmp_addrmask_reply_ttl	icmp_addrmask_reply_ip_id	icmp_info_reply	icmp_info_reply_ttl	icmp_info_reply_ip_id	icmp_unreach_echoed_dtsize	icmp_unreach_reply_ttl	icmp_unreach_precedence_bits	icmp_unreach_df_bit	icmp_unreach_ip_id	icmp_unreach_echoed_udp_cksum	icmp_unreach_echoed_ip_cksum	icmp_unreach_echoed_ip_id	icmp_unreach_echoed_total_len	icmp_unreach_echoed_3bit_flags
Microsoft Windows NT 4 Server Service Pack 6a	0	3	3	1	128	0	128	3	0	128	3	0	128	3	8	128	0	0	3	5	5	5	5	5
Microsoft Windows NT 4 Server Service Pack 5	0	3	3	1	128	0	128	3	0	128	3	0	128	3	8	128	0	0	3	5	5	5	5	5
Microsoft Windows NT 4 Server Service Pack 4	0	3	3	1	128	0	128	3	0	128	3	0	128	3	8	128	0	0	3	5	5	5	5	5
Microsoft Windows NT 4 Server Service Pack 3	0	3	3	1	128	0	128	3	1	128	3	0	128	3	8	128	0	0	3	5	5	5	5	5
Microsoft Windows NT 4 Server Service Pack 2	0	3	3	1	128	0	128	3	1	128	3	0	128	3	8	128	0	0	3	5	5	5	5	5
Microsoft Windows NT 4 Server Service Pack 1	0	3	3	1	128	0	128	3	1	128	3	0	128	3	8	128	0	0	3	5	5	5	5	5
Microsoft Windows NT 4 Server	0	3	3	1	128	0	128	3	1	128	3	0	128	3	8	128	0	0	3	5	5	5	5	5
Microsoft Windows NT 4 Workstation Service Pack 6a	0	3	3	1	128	0	128	3	0	128	3	0	128	3	8	128	0	0	3	5	5	5	5	5
Microsoft Windows NT 4 Workstation Service Pack 5	0	3	3	1	128	0	128	3	0	128	3	0	128	3	8	128	0	0	3	5	5	5	5	5
Microsoft Windows NT 4 Workstation Service Pack 4	0	3	3	1	128	0	128	3	0	128	3	0	128	3	8	128	0	0	3	5	5	5	5	5
Microsoft Windows NT 4 Workstation Service Pack 3	0	3	3	1	128	0	128	3	1	128	3	0	128	3	8	128	0	0	3	5	5	5	5	5
Microsoft Windows NT 4 Workstation Service Pack 2	0	3	3	1	128	0	128	3	1	128	3	0	128	3	8	128	0	0	3	5	5	5	5	5
Microsoft Windows NT 4 Workstation Service Pack 1	0	3	3	1	128	0	128	3	1	128	3	0	128	3	8	128	0	0	3	5	5	5	5	5
Microsoft Windows NT 4 Workstation	0	3	3	1	128	0	128	3	1	128	3	0	128	3	8	128	0	0	3	5	5	5	5	5
Microsoft Windows 98 Second Edition (SE)	0	3	3	1	128	1	128	3	1	128	3	0	128	3	8	128	0	0	3	5	5	5	5	5
Microsoft Windows 98	0	3	3	1	128	1	128	3	1	128	3	0	128	3	8	128	0	0	3	5	5	5	5	5
Microsoft Windows 95	0	3	3	1	32	0	32	3	1	32	3	0	32	3	8	32	0	0	3	5	5	5	5	5
NetBSD 1.6.1	3	3	3	0	255	1	255	3	0	255	3	0	255	3	8	255	0	0	3	5	5	5	5	5
NetBSD 1.6	3	3	3	0	255	1	255	3	0	255	3	0	255	3	8	255	0	0	3	5	5	5	5	5
NetBSD 1.5.3	3	3	3	1	255	1	255	3	0	255	3	0	255	3	8	255	0	0	3	5	5	5	5	5
NetBSD 1.5.2	3	3	3	1	255	1	255	3	0	255	3	0	255	3	8	255	0	0	3	5	5	5	5	5
NetBSD 1.5.1	3	3	3	1	255	1	255	3	0	255	3	0	255	3	8	255	0	0	3	5	5	5	5	5
NetBSD 1.5	3	3	3	1	255	1	255	3	0	255	3	0	255	3	8	255	0	0	3	5	5	5	5	5
NetBSD 1.4.3	3	3	3	1	255	1	255	3	0	255	3	0	255	3	8	255	0	0	3	5	5	5	5	5
NetBSD 1.4.2	3	3	3	1	255	1	255	3	0	255	3	0	255	3	8	255	0	0	3	5	5	5	5	5
NetBSD 1.4.1	3	3	3	1	255	1	255	3	0	255	3	0	255	3	8	255	0	0	3	5	5	5	5	5

OS_ID	icmp_echo_code	icmp_echo_ip_id	icmp_echo_tos_bits	icmp_echo_df_bit	icmp_echo_reply_ttl	icmp_timestamp_reply	icmp_timestamp_reply_ttl	icmp_timestamp_reply_ip_id	icmp_addrmask_reply	icmp_addrmask_reply_ttl	icmp_addrmask_reply_ip_id	icmp_info_reply	icmp_info_reply_ttl	icmp_info_reply_ip_id	icmp_unreach_echoed_dtsize	icmp_unreach_reply_ttl	icmp_unreach_precedence_bits	icmp_unreach_df_bit	icmp_unreach_ip_id	icmp_unreach_echoed_udp_cksum	icmp_unreach_echoed_ip_cksum	icmp_unreach_echoed_ip_id	icmp_unreach_echoed_total_len	icmp_unreach_echoed_3bit_flags
NetBSD 1.4	3	3	3	1	255	1	255	3	0	255	3	0	255	3	8	255	0	0	3	5	5	5	5	5
NetBSD 1.3.3	3	3	3	1	255	1	255	3	0	255	3	0	255	3	8	255	0	1	3	0	0	6	5	6
NetBSD 1.3.2	3	3	3	1	255	1	255	3	0	255	3	0	255	3	8	255	0	1	3	0	0	6	5	6
NetBSD 1.3.1	3	3	3	1	255	1	255	3	0	255	3	0	255	3	8	255	0	1	3	0	0	6	5	6
NetBSD 1.3	3	3	3	1	255	1	255	3	0	255	3	0	255	3	8	255	0	1	3	0	0	6	5	6
OpenBSD 3.3	3	3	3	1	255	1	255	3	0	255	3	0	255	3	8	255	0	0	3	5	2	5	-20	5
OpenBSD 3.2	3	3	3	1	255	1	255	3	0	255	3	0	255	3	8	255	0	0	3	5	2	5	-20	5
OpenBSD 3.1	3	3	3	1	255	1	255	3	0	255	3	0	255	3	8	255	0	0	3	5	2	5	-20	5
OpenBSD 3.0	3	3	3	1	255	1	255	3	0	255	3	0	255	3	8	255	0	0	3	5	2	5	-20	5
OpenBSD 2.9	3	3	3	1	255	1	255	3	0	255	3	0	255	3	8	255	0	0	3	5	5	5	-20	5
OpenBSD 2.8	3	3	3	1	255	1	255	3	0	255	3	0	255	3	8	255	0	0	3	5	5	5	-20	5
OpenBSD 2.7	3	3	3	1	255	1	255	3	0	255	3	0	255	3	8	255	0	0	3	5	5	5	-20	5
OpenBSD 2.6	3	3	3	1	255	1	255	3	0	255	3	0	255	3	8	255	0	0	3	5	5	5	-20	5
OpenBSD 2.5	3	3	3	1	255	1	255	3	0	255	3	0	255	3	8	255	0	0	3	5	5	5	5	5
OpenBSD 2.4	3	3	3	1	255	1	255	3	0	255	3	0	255	3	8	255	0	1	3	0	0	6	5	6
Sun Solaris 9 (SunOS 2.9)	3	3	3	1	255	1	255	3	1	255	3	0	255	3	64	255	0	1	3	5	5	5	5	5
Sun Solaris 8 (SunOS 2.8)	3	3	3	1	255	1	255	3	1	255	3	0	255	3	64	255	0	1	3	5	5	5	5	5
Sun Solaris 7 (SunOS 2.7)	3	3	3	1	255	1	255	3	1	255	3	0	255	3	64	255	0	1	3	5	5	5	5	5
Sun Solaris 6 (SunOS 2.6)	3	3	3	1	255	1	255	3	1	255	3	0	255	3	64	255	0	1	3	5	5	5	5	5
Sun Solaris 2.5.1	3	3	3	1	255	1	255	3	1	255	3	0	255	3	64	255	0	1	3	5	5	5	5	5

ภาคผนวก ข.
การติดตั้งและทดสอบระบบเบื้องต้น

ขั้นตอนการสร้างและติดตั้งมาสเตอร์โหนด

1. สิ่งที่ต้องใช้

- เครื่องคอมพิวเตอร์
- การ์ดเครือข่าย 1 ชุด
- ระบบปฏิบัติการลินุกซ์ (ในที่นี้จะใช้ LinuxTLE 5.0)
- ซอฟท์แวร์สำหรับมาสเตอร์โหนด {ชื่อไฟล์ mn.tar.gz}
- Scripts สำหรับจัดการ SMR {ชื่อไฟล์ bin.tar.gz}
- OpenLDAP server {ชื่อไฟล์ openldap-2.1.10.tgz}
- Tcl/Tk {ชื่อไฟล์ ActiveTcl-8.4.10.1.149958-linux-ix86.tar.gz }
- library สำหรับใช้อ่าน configuration {ชื่อไฟล์ libtc-1.1.0.tar.gz}
- ไฟล์กำหนดค่าเริ่มต้นของ SMR {ชื่อไฟล์ slapd.conf}
- ไฟล์กำหนดค่าฐานข้อมูล {ชื่อไฟล์ amnet.schema}
- Script สำหรับติดตั้ง root node ใน LDAP server {ชื่อไฟล์ anidsan.ldif}
- โปรแกรมควบคุมพื้นฐาน (.so) ที่จำเป็นต้องใช้ในระบบ และ script สำหรับการติดตั้งหรือถอดถอนโปรแกรม (.ldif)
    a. libLogTrace.so, LogTrace.ldif, LogTrace_del.ldif
    b. libNetScan.so, NetScan.ldif, NetScan_del.ldif
- โปรแกรมตรวจสอบการบุกรุก (โปรแกรมทดสอบ) และ script สำหรับการติดตั้ง
    a. libSD_LsassExploit.so, SD_LsassExploit.ldif, SD_LsassExploit_del.ldif
- โปรแกรมสำหรับสร้างกุญแจสาธารณะ {ชื่อไฟล์ genkey}

2. การสร้างคลังเก็บโปรแกรมควบคุม (service module repository)

เครือข่ายแอ็กทีฟ AMnet จะใช้ LDAP server เป็นคลังเก็บโปรแกรมควบคุม (SMR) ซึ่งในที่นี้จะใช้ซอฟท์แวร์ของ OpenLDAP โดยการติดตั้งให้เตรียมไฟล์ openldap-2.1.10.tgz ซึ่งสามารถดาวน์โหลดได้จาก http://www.openldap.org จากนั้นให้ทำตามขั้นตอนดังนี้

```
$ cd   $HOME
$ tar  -xzvf openldap-2.1.10.tgz
$ cd openldap-2.1.10
$ ./configure --enable-ldbm (หมายเหตุ: เลือกใช้ฐานข้อมูลแบบ "ldbm")
$ make depend
$ make
$ make test
$ su
# make install (หมายเหตุ: คำสั่งนี้ต้องป้อนโดย superuser)
```

หลังจากติดตั้งแล้วต้องทำการกำหนดค่าเริ่มต้นของ SMR โดยใช้ไฟล์ที่เตรียมไว้

```
$ cp  slapd.conf   /usr/local/etc/openldap/
$ cp  amnet.schema   /usr/local/etc/openldap/schema/
```

แก้ไขค่าในไฟล์ slapd-amnet.conf ให้ถูกต้องดังต่อไปนี้

1> เปลี่ยนตำแหน่ง directory ใหม่ดังนี้

```
/etc/openldap/schema ---------> /usr/local/etc/openldap/schema
/var/state/slapd.args ------------> /usr/local/var/slapd.args
/var/run/slapd/slapd.pid --------> /usr/local/var/slapd.pid
/var/openldap-ldbm -------------> /usr/local/var/openldap-data
/etc/openldap/schema/amnet-tutorial.schema -------->
/usr/local/etc/openldap/schema/amnet.schema
```

2> เปลี่ยนรหัสผ่านในบรรทัดที่มีคำว่า "rootpw YOUR_PASSWORD" ให้เป็นตามต้องการ

ทำการ extract เครื่องมือสำหรับจัดการ LDAP server

```
$ tar  –xzvf   bin.tar.gz
$ cd  bin
$ export PATH=$PATH:$HOME/bin/
```

สร้างไฟล์ชื่อ SMR ไว้เพื่อระบุหมายเลขไอพีของ LDAP server

```
$  echo  LDAPHOST=192.168.200.1  >  $HOME/bin/SMR
```

เริ่มต้นการทำงานของ LDAP server โดย

```
$ startldap
```

เพิ่ม root node ไว้ใน LDAP server โดยใช้ script ชื่อ amnet.ldif

```
$ add -f amnet.ldif
```

เสร็จสิ้นการติดตั้ง LDAP server เพื่อใช้เป็นคลังเก็บโปรแกรมควบคุม ทดสอบด้วยคำสั่ง

```
$ show c
```

3. **การติดตั้งโปรแกรมไว้ในคลังเก็บโปรแกรม**

แปลงโปรแกรมให้อยู่ในรูปแบบ base64 เพื่อให้สามารถติดตั้งใน LDAP server ได้

```
$ encdec –e –b < libPROGRAM.so > libPROGRAM.so.64
```

ติดตั้งโปรแกรมควบคุมพื้นฐานโดยใช้ script สำหรับโปรแกรมนั้น ๆ

```
$ add -f LogTrace.ldif
$ add -f NetScan.ldif
```

การติดตั้งโปรแกรมตรวจสอบการบุกรุก (โปรแกรมตัวอย่าง) โดย

```
$ add -f SD_LsassExploit.ldif
```

หากต้องการถอดถอนโปรแกรมควบคุมหรือโปรแกรมตรวจสอบการบุกรุกให้ใช้ script สำหรับถอดถอนโปรแกรม เช่น

```
$ del -f SD_LsassExploit_del.ldif
$ del -f NetScan_del.ldif
$ del -f LogTrace_del.ldif
```

หมายเหตุ หากทำการสร้างโปรแกรมขึ้นมาใหม่ต้องเขียน script สำหรับติดตั้งและถอดถอนโปรแกรมใหม่ด้วย โดยดูตัวอย่างจาก script ที่เตรียมไว้ให้แล้ว

4. **การสร้างกุญแจสาธารณะ**

การสร้างกุญแจสาธารณะสำหรับมาสเตอร์โหนดโดยใช้คำสั่ง

```
$ genkey –mn
```

จะได้ไฟล์ MN_PrivateKey ซึ่งจะเป็นไฟล์ที่เก็บ Private key ของมาสเตอร์โหนดและจะได้ไฟล์ All_Publickeys ซึ่งจะเป็นไฟล์สำหรับเก็บ Public key ของมาสเตอร์โหนดและแอ็กทีฟโหนดทั้งหมด

การสร้างกุญแจสาธารณะสำหรับแอ็กทีฟโหนดให้ใช้คำสั่ง

```
$ genkey –an
```

จะได้ไฟล์ AN_PrivateKey ซึ่งจะเป็นไฟล์ที่เก็บ Private key ของแอ็กทีฟโหนด ส่วน Public key ของแอ็กทีฟโหนดจะถูกเก็บไว้ในไฟล์ All_Publickeys โดยไฟล์ทั้ง 2 ไฟล์นี้จะต้องนำไปติดตั้งไว้ในแอ็กทีฟโหนดที่ถูกสร้างขึ้นใหม่ และหลังจากการสร้างกุญแจสาธารณะของแอ็กทีฟโหนดขึ้นใหม่ ต้องทำการ restart โปรแกรม MN_manager ด้วยทุกครั้ง

การดูข้อมูลเกี่ยวกับกุญแจที่อยู่ในไฟล์ให้ใช้คำสั่ง

```
$ genkey –rd
```

5. การเริ่มต้นการทำงานของมาสเตอร์โหนด

```
$  startldap
$  $HOME/MN/mn.tcl
```

จากนั้นให้กดปุ่ม start MN เพื่อเริ่มต้นการทำงานของมาสเตอร์โหนด

ขั้นตอนการสร้างและติดตั้งแอ็กทีฟโหนด

1. สิ่งที่ต้องใช้

- เครื่องคอมพิวเตอร์
- การ์ดเครือข่าย 2 ชุด
- ระบบปฏิบัติการลินุกซ์ทะเล รุ่น 5.0
- ลินุกซ์เคอร์เนล รุ่น 2.4.24 {ชื่อไฟล์ linux-2.4.24.tar.gz}
- ซอฟท์แวร์เครือข่ายแอ็กทีฟ “AMnet” version 2.6 {ชื่อไฟล์ flexinet-2.6.anidsan.tar.gz}
- Patch สำหรับปรับปรุงฟังก์ชั่น Bridge ของลินุกซ์ ดาวน์โหลดจาก http://www.ssi.bg/~ja/#bridging {ชื่อไฟล์: bridge-ipmode-2.4.22-2.diff}
- Bridge Utility สำหรับใช้สั่งการทำงานของ Bridge ดาวน์โหลดจาก http://bridge.sourceforge.net {ชื่อไฟล์: bridge-utils-1.0.4.tar.gz}
- Patch สำหรับปรับปรุง Bridge Utility {ชื่อไฟล์: bridge-utils-1.0.4-ipmode-1.diff}
- โปรแกรมควบคุมหลักของแอ็กทีฟโหนด AN_manager {ชื่อไฟล์ libAN_manager.so}
- Script สำหรับการเรียกใช้งานโปรแกรม AN_manager {ชื่อไฟล์ AN_manager.ee}
- ไฟล์กุญแจสาธารณะ (Public Key) ที่ได้มาจากมาสเตอร์โหนด {ชื่อไฟล์ All_Publickeys}
- ไฟล์กุญแจ (Private Key) ที่ได้มาจากมาสเตอร์โหนด {ชื่อไฟล์ AN_PrivateKey}

2. การสร้าง kernel ใหม่

ดาวน์โหลด kernel source ที่ต้องการจาก http://www.kernel.org (เช่น linux-2.4.24.tar.gz) แล้ว save ไว้ที่ /usr/src และ extract ไฟล์ด้วยคำสั่ง

```
$ tar  -xzvf   linux-2.4.24.tar.gz
$ cd  /usr/src/linux-2.4.24
```

จากนั้นแก้ไขค่า Extraversion ในไฟล์ที่ชื่อ Makefile ให้เป็นตามต้องการ เช่น -MyKernel แล้วทำการกำหนดค่า kernel ใหม่โดยใช้คำสั่ง

```
$ make   menuconfig
```

จะปรากฏ menu เพื่อให้เลือกส่วนประกอบของ kernel ใหม่ตามต้องการเมื่อกำหนดค่าที่ต้องการแล้วให้บันทึกและออกจาก menu จากนั้นให้ทำการสร้าง kernel ใหม่โดยคำสั่ง

```
$ make dep (ทำการตรวจสอบความขึ้นต่อกันของแต่ละ package)
$ make clean (ทำการลบไฟล์ทิ้งในกรณีที่เราเคยทำการ compile ไว้ก่อนหน้า)
$ make bzImage (ทำการสร้าง kernel)
$ make modules (ทำการสร้าง kernel modules)
$ su
# make modules_install (ทำการติดตั้ง kernel modules ไว้ใน /lib/modules/…
# make install
```

ซึ่งคำสั่ง make install จะทำการ copy kernel ใหม่ไปไว้ใน /boot และจะสร้าง initial ramdisk ไว้เพื่อใช้ในขณะ boot หรือสามารถสร้างเองก็ได้ด้วยคำสั่ง
mkinitrd /boot/initrd-mybridge.img 2.4.24 (โดย 2.4.24 คือชื่อ Directory ที่ใช้เก็บ modules ของ kernel ที่ถูกสร้างขึ้นด้วยคำสั่ง make modules_install โดยที่ Directory นี้จะอยู่ใน /lib/modules/...)
นอกจากนั้นคำสั่ง make install จะทำการ copy ไฟล์ System.map ไปไว้ใน /boot ให้และยังทำการเพิ่มเมนูของ kernel ใหม่ในไฟล์ /boot/grub/grub.conf ให้โดยอัตโนมัติ

3. การสร้างแอ็กทีฟโหนด

- เตรียมเครื่องคอมพิวเตอร์ที่ติดตั้งระบบปฏิบัติการลินุกซ์
- extract ไฟล์ linux-2.4.24.tar.gz ไว้ใน /usr/src
- extract ไฟล์ bridge-utils-1.0.4.tar.gz ไว้ใน $HOME/
- ทำการ patch kernel และ bridge tool (brctl) เพื่อให้ bridge สามารถส่ง packet ขึ้นไปยัง IPTABLES ได้

```
$ cd /usr/src/linux-2.4.24/
$ patch -p1 < bridge-ipmode-2.4.22-2.diff
$ cd $HOME/ bridge-utils-1.0.4/
$ patch -p1 < bridge-utils-1.0.4-ipmode-1.diff
```

- compile kernel ใหม่โดยให้เปิดใช้งาน 802.1d ethernet bridging และ netfilter ด้วย (ดูวิธีการสร้าง kernel ที่หัวข้อก่อนหน้านี้)
- จากนั้นให้เพิ่มบรรทัดเหล่านี้เข้าในไฟล์ /etc/rc.local และแก้ IP_ADDRESS และ DEFAULT_GATEWAY ให้ถูกต้อง

```
### Add Bridge
/usr/local/sbin/brctl addbr mybridge
/usr/local/sbin/brctl addif mybridge eth0
/usr/local/sbin/brctl addif mybridge eth1
### Set Bridge Port
ifconfig eth0 0.0.0.0 up
ifconfig eth1 0.0.0.0 up
### Add IPaddress to Bridge
ifconfig mybridge <IP_ADDRESS> up
### Set Default Gateway
route add default gw <DEFAULT_GATEWAY>
### Enable IP-Mode to Bridge
```

```
/usr/local/sbin/brctl ipmode mybridge on
### Enable IP forwarding
echo 1 > /proc/sys/net/ipv4/ip_forward
### Mark all packet with interface-card ID ( eth0 = 1, eth1 = 2 )
iptables -A PREROUTING -i eth0 -t mangle -j MARK --set-mark 1
iptables -A PREROUTING -i eth1 -t mangle -j MARK --set-mark 2
```

เมื่อถึงขั้นนี้ก็สามารถใช้เครื่องเป็น Transparent Bridge ได้แล้ว ซึ่งขั้นตอนต่อไปคือติดตั้งซอฟท์แวร์ AMnet Active Network

```
$  cd  $HOME
$  tar  -xzvf   flexinet-2.6.anid.tar.gz
$  cd  flexinet-2.6
$  make
$ su –
$ cd  $HOME/flexinet-2.6
$ make install
```

เมื่อถึงขั้นนี้ ซอฟท์แวร์ AMnet จะถูกติดตั้งไว้ในเครื่องแล้ว จากนั้นให้สร้าง directory ชื่อ anidsan ไว้ใน flexinet-2.6/ee/ แล้วนำไฟล์ libAN_manager.so และ AN_manager.ee ไปเก็บไว้ จากนั้นให้เข้าไปที่มาสเตอร์โหนดเพื่อทำการสร้าง Private key สำหรับแอ็กทีฟโหนดโดยใช้คำสั่ง

```
$  genkey  -an
```

ซึ่งจะได้ไฟล์ AN_PrivateKey และไฟล์ All_Publickeys ให้ copy ไฟล์ดังกล่าวไปเก็บไว้ในเครื่องแอ็กทีฟโหนดใน flexinet-2.6/ee/anidsan/
เมื่อมาถึงขั้นนี้แอ็กทีฟโหนดก็จะสามารถใช้งานได้แล้ว ซึ่งสามารถเริ่มการทำงานได้โดย

```
$  su  -
```

```
# cd   $HOME/flexinet-2.6/ee/anidsan
# ee   AN_manager.ee
```

ซึ่งแอ็กทีฟโหนดก็จะเริ่มทำงานและติดต่อไปยังมาสเตอร์โหนดเพื่อขอดาวน์โหลดโปรแกรมควบคุมมาติดตั้ง

4. การแก้ปัญหาเมื่อ AMnet หา Library ไม่พบ

หากต้องการเขียนโปรแกรมควบคุมใน AMnet ที่มีการเรียกใช้ไลบรารี่ที่ไม่ได้ระบุไว้ใน Makefile ที่มีมากับ AMnet จะเกิด error ว่า undefined symbol :ขึ้นมา ซึ่งปัญหานี้สามารถแก้ไขได้โดยการกำหนดไลบรารี่ที่โปรแกรมควบคุมนั้นเรียกใช้ โดยเข้าไปแก้ไข Makefile เพื่อกำหนดชื่อไลบรารี่ไฟล์และพาธที่เก็บไฟล์นั้น โดยใช้ออปชั่น -l เพื่อระบุชื่อไฟล์ และออปชั่น -L เพื่อระบุพาธ เช่น L และ -l

ตัวอย่างเช่น ไลบรารี่ไฟล์ชื่อ liblist.a อยู่ใน $HOME/flexinet-2.6/ee/list/ ก็ให้แก้ Makefile โดยเพิ่มบรรทัดนี้เข้าไป

```
LDFLAGS += -L$HOME/flexinet-2.6/ee/list
LIBS += -llist
```

หมายเหตุ - ชื่อไลบรารี่ไฟล์ liblist.a จะต้องตัด lib และนามสกุล .a ออกก่อน

จากนั้นก็ให้เพิ่มออปชั่นทั้งสองเข้าไปในบรรทัดคำสั่งที่ใช้ในการคอมไพล์ เช่น

```
$(CC) $(CFLAGS) $(LDFLAGS) -shared -o $@ $< $(LIBS)
```

ข้อสังเกตุ - ต้องใส่ชื่อไลบรารี่ $(LIBS)ไว้หลังออปชั่น -o

หลังจากนั้นให้คอมไพล์แล้วลองรันดู เท่านี้ error ก็จะหายไป

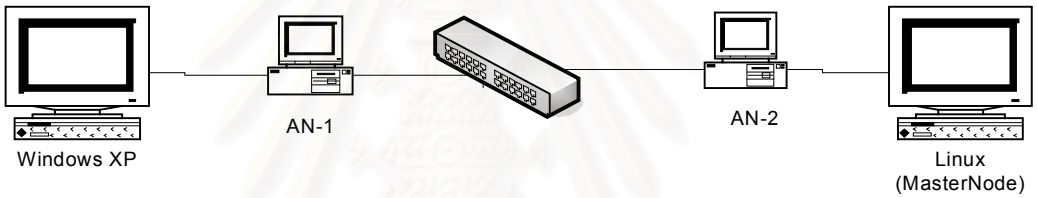
ทดสอบการทำงานเบื้องต้นของระบบ

1. สิ่งที่ต้องใช้ในการทดสอบ

- เครื่องที่ใช้เป็นแอ็กทีฟโหนด 2 เครื่อง และมาสเตอร์โหนด 1 เครื่อง
- เครื่องคอมพิวเตอร์สำหรับติดตั้งในเครือข่ายย่อย 2 เครื่อง
- โปรแกรมสำหรับบุกรุกเครื่องที่อาศัยช่องโหว่ของบริการ Lsass {ชื่อไฟล์ lsassexploit}
- โปรแกรมทดสอบการส่งผ่านข้อมูลที่พอร์ตต่าง ๆ {ชื่อไฟล์ hping}

2. ขั้นตอนการทดสอบการทำงานของระบบเบื้องต้น

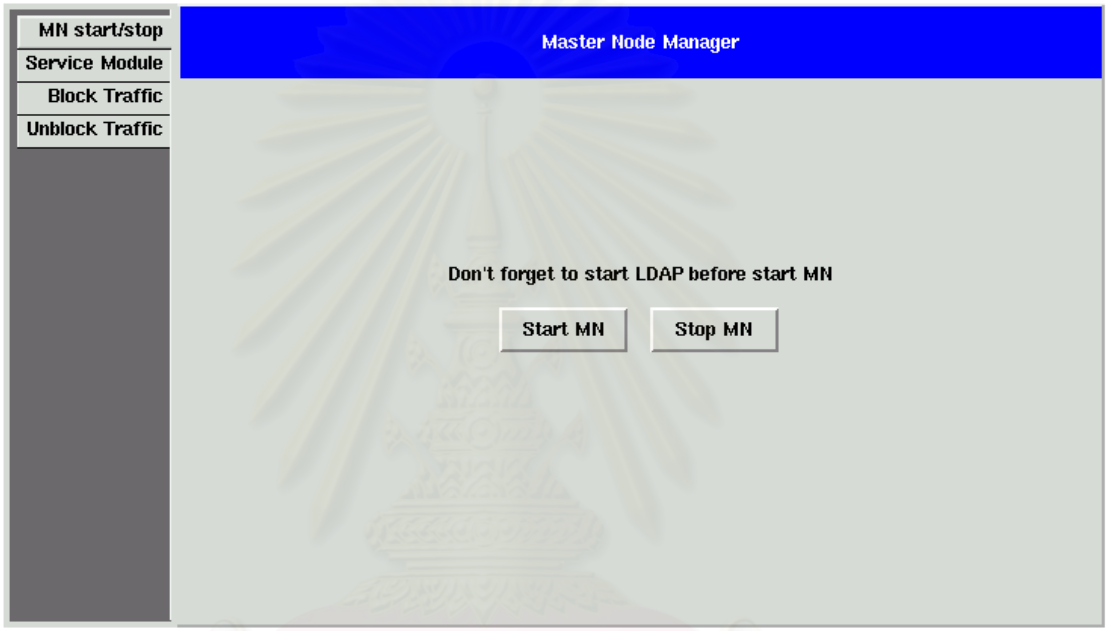
เมื่อเตรียมมาสเตอร์โหนดและแอ็กทีฟโหนดเรียบร้อยแล้ว ก็สามารถทดสอบการทำงานเบื้องต้นของระบบได้ โดยนำมาสเตอร์โหนดและแอ็กทีฟโหนดเชื่อมต่อกันให้เป็นเครือข่ายดังรูป



จากนั้นให้เข้าไปที่มาสเตอร์โหนดแล้วพิมพ์คำสั่งเพื่อเริ่มการทำงานดังนี้

```
$ startldap
$ $HOME/MN/mn.tcl
```

จะปรากฏหน้าจอดังรูปต่อไปนี้ ให้เลือกแท็บ “MN start/stop” แล้วกดปุ่ม Start MN

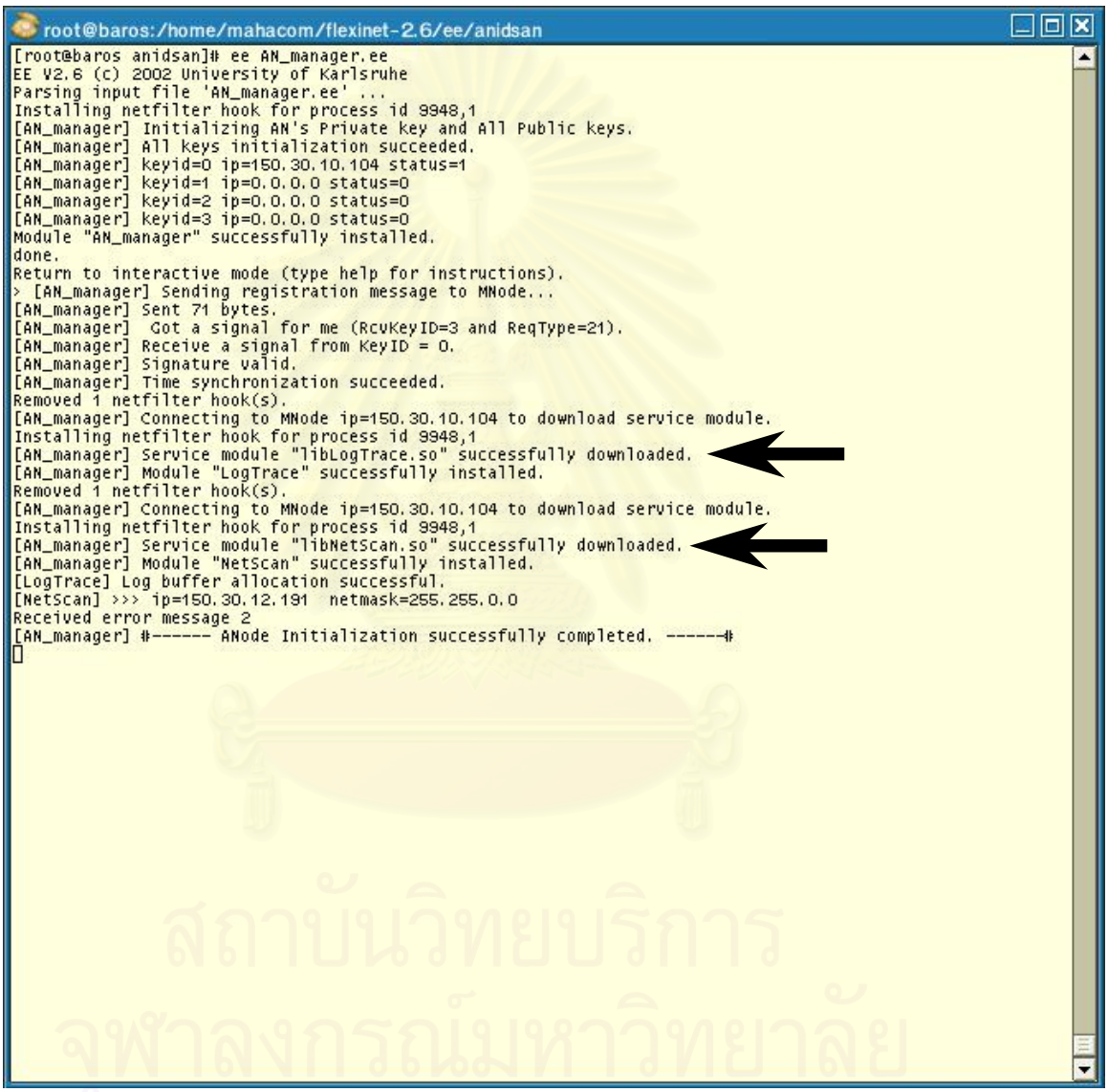


เมื่อเริ่มการทำงานมาสเตอร์โหนดเสร็จแล้ว จากนั้นให้เข้าไปที่แอ็กทีฟโหนดเพื่อเริ่มการทำงานโดย

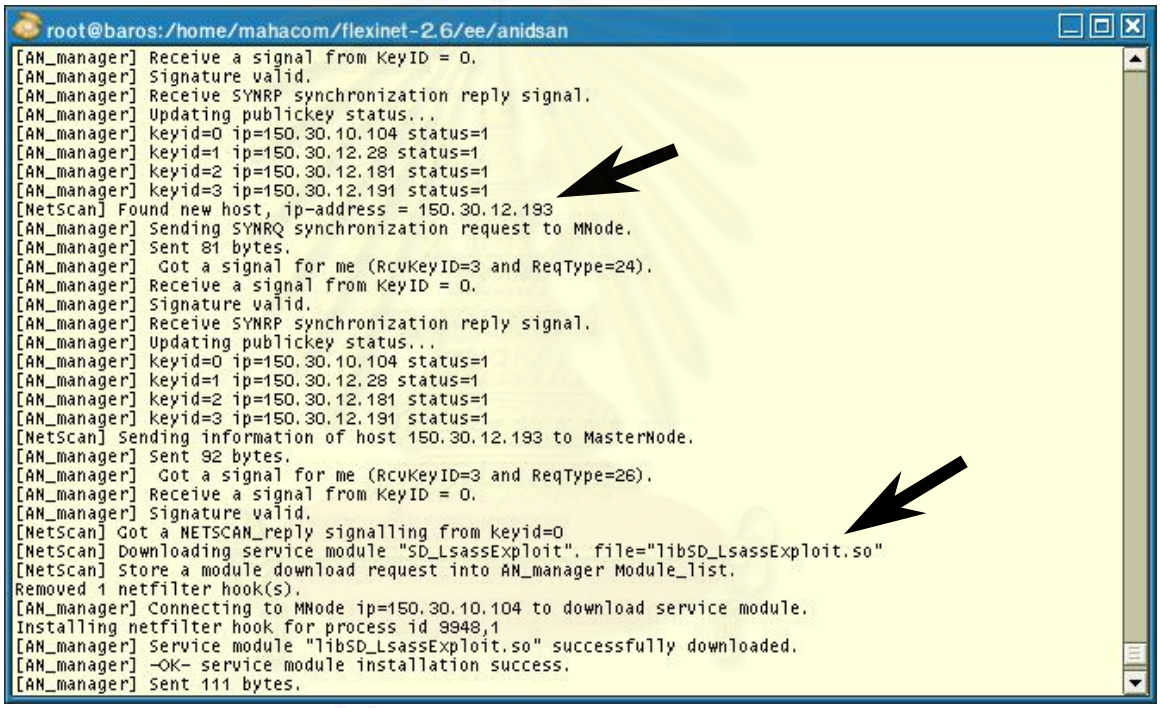
```
$ su –
# cd  $HOME/flexinet/ee/anidsan
# ee  AN_manager.ee
```

ซึ่งคำสั่งนี้จะเป็นการติดตั้งโปรแกรมควบคุม AN_manager เข้าไปใน Execution Environment ของแอ็กทีฟโหนด

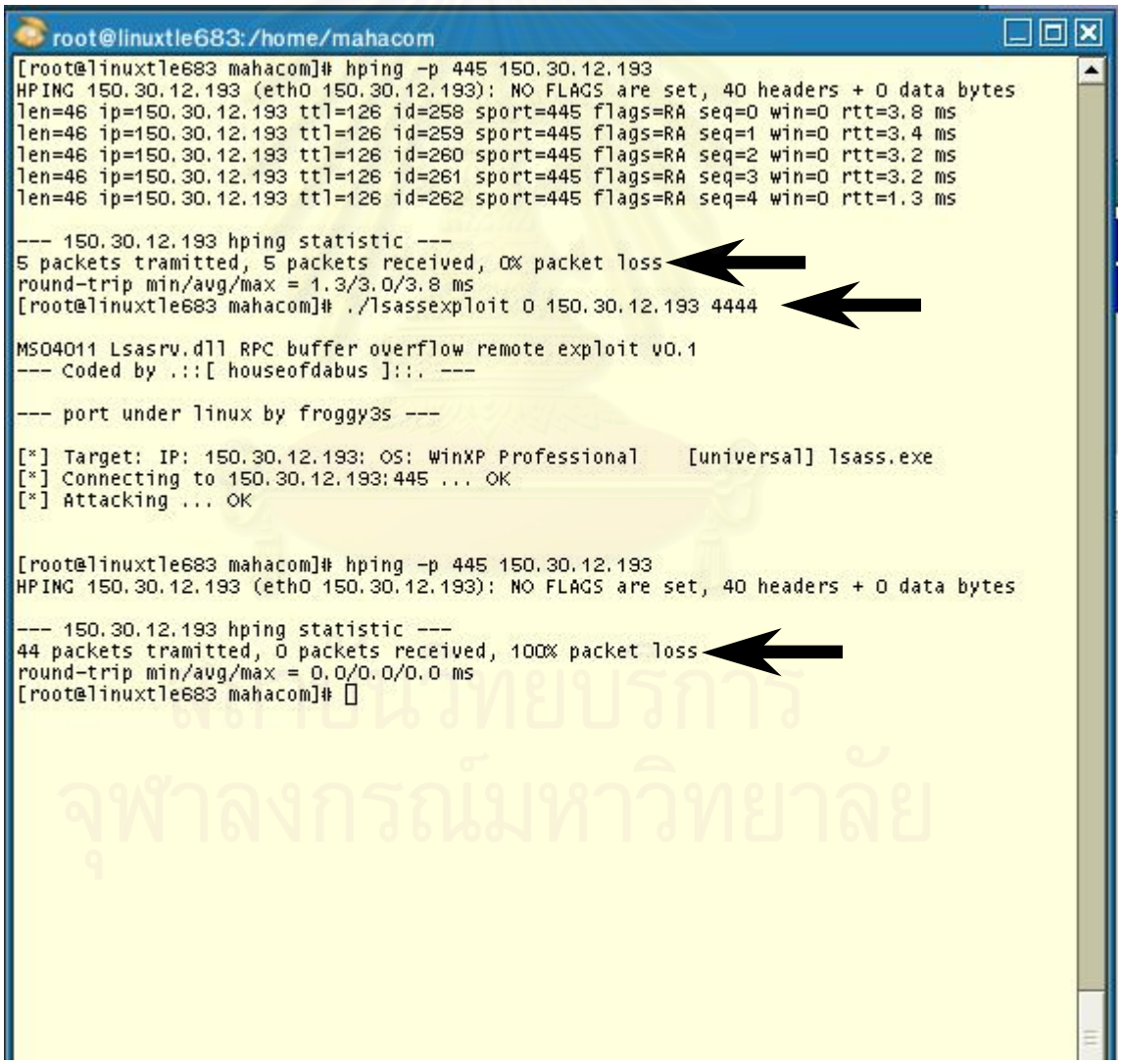
จากนั้นแอ็กทีฟโหนดก็จะเริ่มทำงานตามคำสั่งในโปรแกรมควบคุม AN_manager โดยจะเริ่มส่งสัญญาณควบคุมเพื่อทำการติดต่อกับมาสเตอร์โหนดเพื่อขอดาวน์โหลดโปรแกรมควบคุมพื้นฐานมาติดตั้ง ซึ่งจะปรากฏหน้าจอดังรูปต่อไปนี้



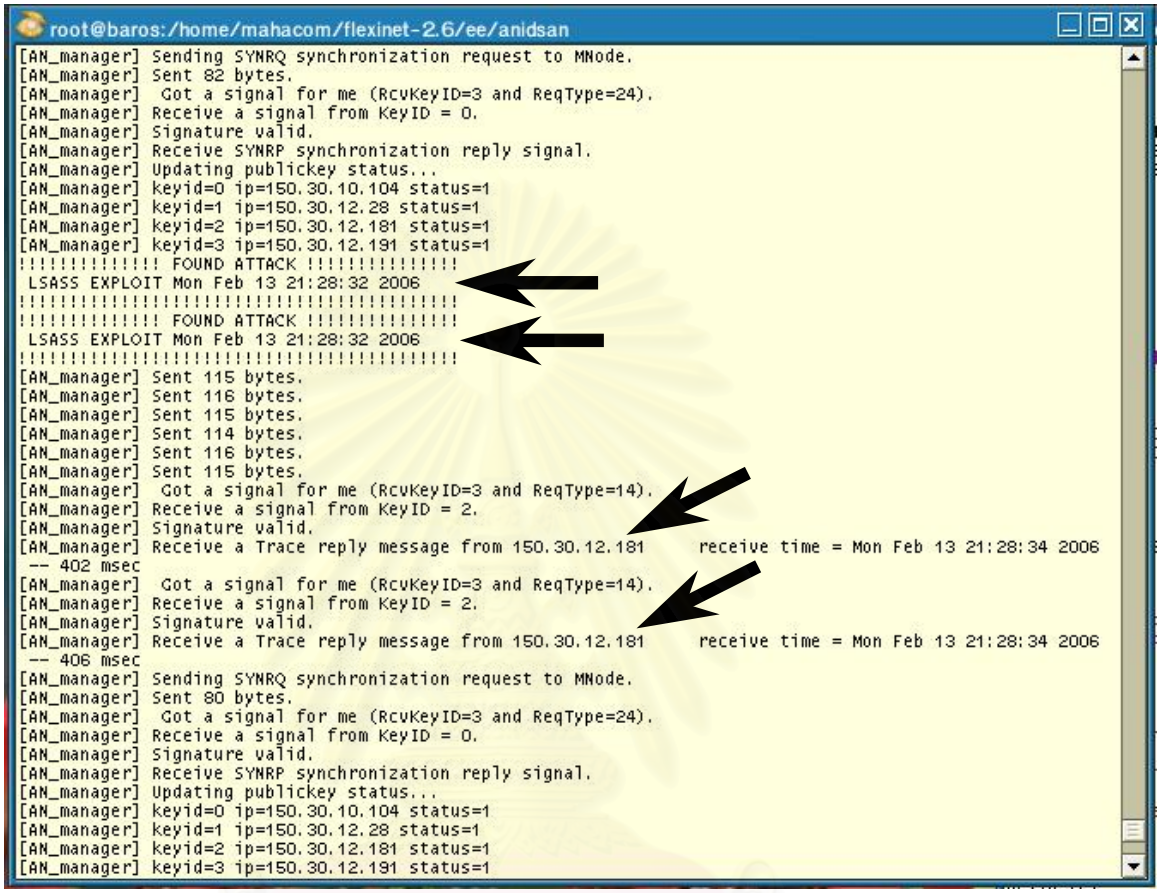
จากนั้นนำเครื่องคอมพิวเตอร์ที่ติดตั้งระบบปฏิบัติการวินโดวส์ 2000 ที่มีช่องโหว่ของบริการ Lsass ติดตั้งเข้าไปในเครือข่ายย่อยที่ 1 หรือต่อเข้ากับอีกด้านของแอ็กทีฟโหนด AN1 และเมื่อเครื่องดังกล่าวได้เริ่มส่ง packet ออกมา แอ็กทีฟโหนดก็จะตรวจพบและจะเริ่มทำการสำรวจเครื่องที่ติดตั้งใหม่นั้นทันที และเมื่อสำรวจเสร็จแล้วก็จะส่งสัญญาณติดต่อกับมาสเตอร์โหนดเพื่อสอบถามรายชื่อโปรแกรมตรวจสอบการบุกรุกที่จะต้องติดตั้งเพิ่ม และเมื่อมาสเตอร์โหนดตอบกลับมาแอ็กทีฟโหนดก็จะเริ่มดาวน์โหลดโปรแกรมตรวจสอบการบุกรุก SD_LsassExploit มาติดตั้งทันที ซึ่งจะปรากฏรายละเอียดบนหน้าจอของแอ็กทีฟโหนดดังรูปต่อไปนี้



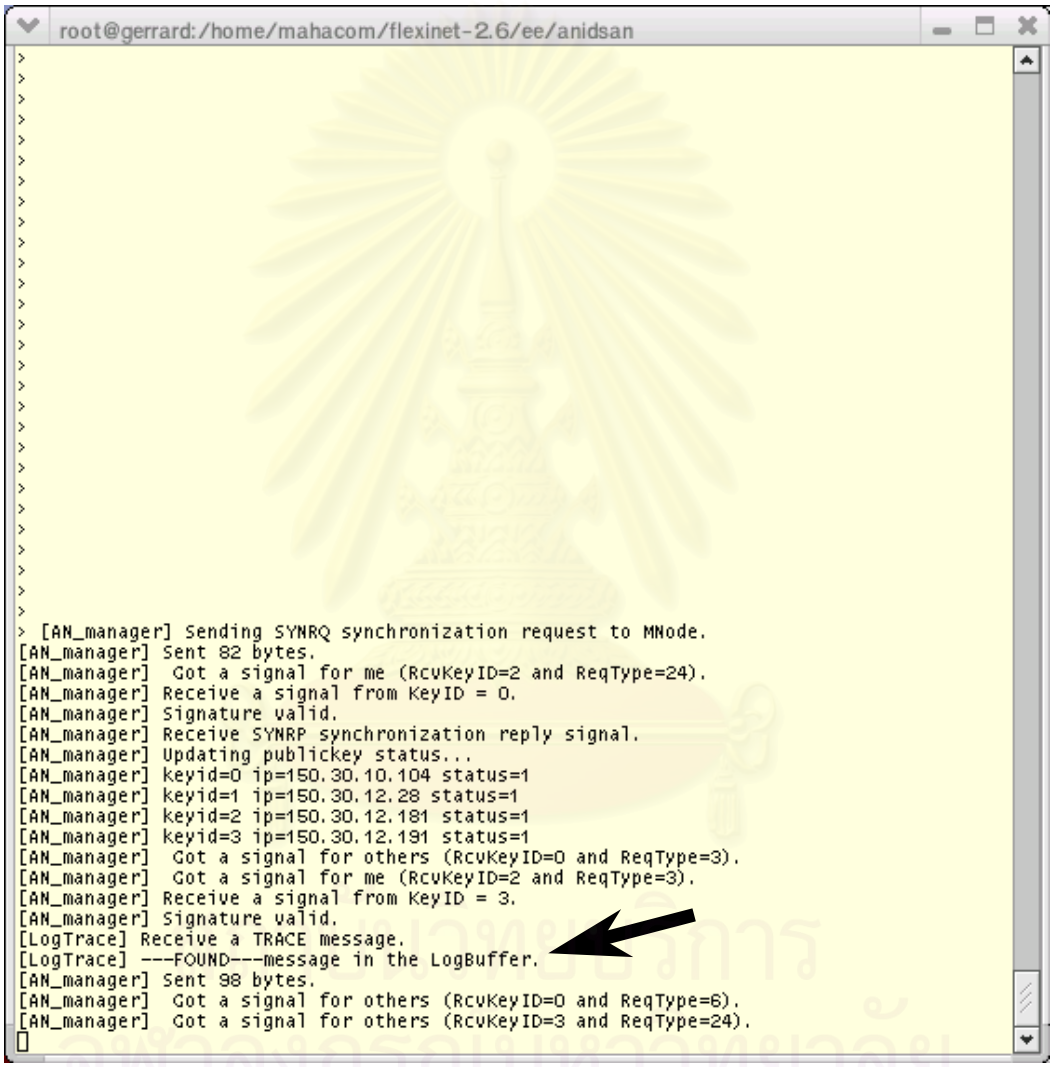
หลังจากนั้นให้เริ่มต้นการทำงานแอ็กทีฟโหนดอีกเครื่องโดยวิธีเดียวกันกับที่กล่าวมาแล้ว และเริ่มทดสอบการตรวจสอบการบุกรุกโดยใช้โปรแกรมสำหรับบุกรุกที่อาศัยช่องโหว่ของบริการ Lsass โดยให้ติดตั้งโปรแกรมสำหรับบุกรุกไว้ในเครื่องที่อยู่ในเครือข่ายย่อยที่ 2 และทดสอบการส่งผ่านข้อมูลไปที่หมายเลขพอร์ต 445 ซึ่งเป็นพอร์ตที่จะใช้บุกรุกโดยใช้เครื่องมือ hping ซึ่งในตอนแรกจะสามารถส่งผ่านข้อมูลได้ปกติ จากนั้นทำการบุกรุกเครื่องที่อยู่ในเครือข่ายย่อยที่ 1 ซึ่งแอ็กทีฟโหนด AN1 ที่ติดตั้งโปรแกรมตรวจสอบการบุกรุก SD_LsassExploit ก็จะตรวจสอบพบการบุกรุกและจะทำการค้นหาเครือข่ายที่ส่งข้อมูลเข้ามาบุกรุกทันทีเพื่อสกัดกั้นการบุกรุกทันที ทำให้หลังจากนั้น hping จะไม่สามารถส่งข้อมูลผ่าน AN2 ได้อีก ดังรูปต่อไปนี้



และที่แอ็กทีฟโหนด AN1 จะแสดงผลว่าได้ตรวจพบการบุกรุกบนหน้าจอดังรูปต่อไปนี้



ในส่วนของแอ็กทีฟโหนด AN2 ซึ่งเป็นผู้ส่งผ่านข้อมูลของผู้บุกรุกออกมาเมื่อได้รับคำขอให้ค้นหาแพ็กเก็ต ก็จะทำการค้นหา และก็พบว่าแพ็กเก็ตดังกล่าวอยู่ใน Log Buffer ดังรูปต่อไปนี้ และจะทำตามคำสั่งที่ระบุมาในคำขอ ซึ่งก็คือให้สกัดกั้นการส่งผ่านข้อมูลที่ใช้หมายเลขพอร์ต 445 ซึ่งเป็นช่องโหว่ที่ใช้บุกรุก



ประวัติผู้เขียนวิทยานิพนธ์

นายมหคม อร่ามเสรีวงศ์ เกิดเมื่อวันที่ 19 มกราคม 2516 สำเร็จการศึกษาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์ จากสถาบันเทคโนโลยีราชมงคล ในปีการศึกษา 2537 เข้าศึกษาระดับปริญญามหาบัณฑิต สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2545