# ระบบตรวจจับผู้บุกรุกเครือข่ายบน Win32

# Network Intrusion Detection System for Win32



นายชาวดี บารมี

นายเมธี ชุมภูชา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

# ระบบตรวจจับผู้บุกรุกเครือข่ายบน Win32

# Network Intrusion Detection System for Win32


โดย

นายชาวดี บารมี

นายเมธี ชุมภูชา

อาจารย์ที่ปรึกษา

อาจารย์ธนา หงษ์สุวรรณ

อาจารย์อัครเดช วัชระภูพงษ์


ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

ปริญญานิพนธ์ปีการศึกษา 2544

ภาควิชา วิศวกรรมศาสตร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบตรวจจับผู้บุกรุกเครือข่ายบน Win32

Network Intrusion Detection System for Win3


ผู้จัดทำ

1. นายชาวดี บารมี รหัสประจำตัว 42015298
2. นายเมธี ชุมภูชา รหัสประจำตัว 42015314


_____ อาจารย์ที่ปรึกษา

(อาจารย์ธนา หงษ์สุวรรณ)

_____ อาจารย์ที่ปรึกษา

(อาจารย์อัครเดช วัชระภูพงษ์)

# ระบบตรวจจับผู้บุกรุกเครือข่ายบน Win32

นายชาวดี บารมี รหัสประจำตัว 42015298
นายเมธี ชุมภูชา รหัสประจำตัว 42015314
อาจารย์ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษา
อาจารย์อัครเดช วัชระภูพงษ์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2544

**บทคัดย่อ**

ความปลอดภัยนับเป็นปัจจัยสำคัญอย่างหนึ่งในการใช้งานคอมพิวเตอร์ที่มีการเชื่อมต่อเป็นเครือข่ายอย่างในปัจจุบัน เนื่องจากมีผู้ต้องการบุกรุกเครื่องคอมพิวเตอร์ผ่านทางเครือข่ายมากขึ้น โดยลักษณะของการบุกรุกสามารถแบ่งได้เป็น 2 รูปแบบคือ การบุกรุกเพื่อค้นหาข้อมูลของเครือข่าย และการบุกรุกโดยการโจมตีเครือข่ายเพื่อให้ปิดบริการแบบ DoS (Denial of Services) การติดตั้งระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ ช่วยให้ผู้ดูแลระบบสามารถวิเคราะห์สาเหตุเมื่อมีปัญหาการบุกรุกในเครือข่ายที่ตนเองรับผิดชอบได้ ซึ่งในปัจจุบันระบบตรวจจับผู้บุกรุกทางเครือข่ายได้รับการพัฒนามาอย่างต่อเนื่อง แต่โดยทั่วไปเน้นไปในการพัฒนาบนแพลตฟอร์มยูนิกซ์ เนื่องจากเหตุผลด้านความแข็งแกร่งของตัวระบบปฏิบัติการเอง ที่ระบบปฏิบัติการยูนิกซ์มักมีผลกระทบจากการโจมตีในชั้นทีซีพีไอพีน้อยกว่าระบบปฏิบัติการวินโดว์ส

การพัฒนาระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์บน Win32 เป็นการพัฒนาระบบตรวจจับผู้บุกรุกที่เน้นความสามารถของการรองรับผลจากการโจมตีที่อาจเกิดขึ้นบนคอมพิวเตอร์ที่ติดตั้งระบบตรวจจับเอง โดยในขั้นตอนการออกแบบโครงสร้างของระบบทั้งหมด ได้มีการศึกษาถึงรูปแบบการโจมตีที่มีผลต่อการทำงานของชั้นทีซีพีไอพีในระบบปฏิบัติการวินโดว์ส ผลที่ได้จากการศึกษานำไปสู่การควบคุมปริมาณแพ็กเก็ตและเวลาที่เหมาะสมในการวิเคราะห์การโจมตี เพื่อป้องกันไม่ให้เกิดผลกระทบจากการโจมตีบนเครื่องที่ติดตั้งระบบตรวจจับผู้บุกรุกในกรณีที่ได้รับแพ็กเก็ตที่เป็นการโจมตีมากเกินไป

# Network Intrusion Detection System (NIDS) for Win32

Mr. Chawdee Baramee
Mr Matee Chumpucha
Mr. Thana Hongsuwan Advisor
Mr. Akkradach Watcharapupong Advisor

**ABSTACT**

Security is one of the most important topics when use computers on network because there are many attacks via computer network nowadays. This kind of network intrusion can be categorized into 2 types, first is Reconnaissance system and second is Denial of Services (DoS) attack. NIDS can help Administrator to analyze any problem when their network is attacked. Today, NIDS is developed in UNIX platform because UNIX operating system is stronger in TCP/IP stack than Windows operating system.

NIDS for Win32 is developed in ability to reduce attacking effect in own host. NIDS for Win32 was designed by knowledge in attack's effect. This Knowledge bring forth to control attacking effect by restrict amount of packet and opportunity time for attack analysis.

# กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จได้ด้วยดี เนื่องจากได้รับการแนะนำ สนับสนุน และให้คำปรึกษาเป็นอย่างดีจาก อาจารย์ธนา หงษ์สุวรรณ และอาจารย์อัครเดช วัชระภูพงษ์ อาจารย์ที่ปรึกษาปริญญานิพนธ์ ซึ่งต้องขอขอบพระคุณเป็นอย่างสูง รวมทั้งอาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่าน ที่ให้การอบรมสั่งสอนวิชาความรู้แก่คณะผู้จัดทำมาโดยตลอด

และขอขอบพระคุณเป็นอย่างสูงสำหรับบุคคลที่สำคัญที่สุดที่ทำให้คณะผู้จัดทำมีวันนี้ คือ บิดา มารดา ผู้เป็นที่เคารพรักยิ่งของคณะผู้จัดทำ ซึ่งท่านให้การอบรมสั่งสอน เลี้ยงดู และให้โอกาสในการศึกษาอย่างเต็มที่ จึงขอกราบขอบพระคุณมา ณ ที่นี้

สุดท้ายนี้ขอขอบพระคุณผู้ดูแลระบบคอมพิวเตอร์ภาควิชาวิศวกรรมศาสตร์ และสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่อำนวยความสะดวกในการใช้งานเครือข่าย และขอขอบคุณเพื่อนๆ ที่ให้ข้อคิดเป็น และเป็นกำลังใจให้เสมอมา

ชาวดี บารมี

เมธี ชุมภูชา

# สารบัญ

# สารบัญ (ต่อ)

# สารบัญ (ต่อ)

# สารบัญภาพประกอบ

หน้าที่

# สารบัญภาพประกอบ (ต่อ)

หน้าที่

# สารบัญตาราง

หน้าที่



# บทที่ 1
# บทนำ

## 1.1 ความสำคัญและที่มา

ปัจจุบันได้มีการนำคอมพิวเตอร์มาให้บริการเพื่อความสะดวกในชีวิตประจำวันและเพื่อสร้างโอกาสทางธุรกิจมากขึ้น เช่นการบริการติดต่อสื่อสารผ่านระบบอิเล็กทรอนิกส์เมล แต่เดิมใช้เพียงติดต่อหากันในกลุ่มคน แต่ปัจจุบันเป็นกลับเป็นส่วนสำคัญในการติดต่อสื่อสารระหว่างองค์กรต่างๆ รวมถึงเป็นส่วนสำคัญในการทำธุรกรรมบนอินเตอร์เน็ต หรือการให้บริการในเวปไซต์ต่างๆ แต่เดิมอาจใช้ในลักษณะการเผยแพร่ข้อมูลความรู้ต่างๆ หรือเป็นการแสดงข่าวสารขององค์กรต่างๆ ซึ่งยังไม่มีการแสวงหากำไรในบริการนั้นอย่างชัดเจน แต่ปัจจุบันเวปไซต์จะพบว่าเวปไซต์ที่จัดทำเพื่อดำเนินธุรกิจบนอินเตอร์เนตมีมากขึ้น จะเห็นว่าการให้บริการต่างๆ บนอินเตอร์เน็ตล้วนแล้วแต่มีความสำคัญต่อโลกปัจจุบันทั้งสิ้น

การให้บริการต่างๆ เหล่านี้ล้วนทำงานในลักษณะไคลเอนต์-เซิร์ฟเวอร์ กล่าวคือ เครื่องที่เปิดให้บริการได้รอรับการเชื่อมต่อจากไคลเอนต์ผ่านจุดเชื่อมต่อหรือพอร์ต โดยแต่ละบริการจะมีพอร์ตเฉพาะของตนเอง เมื่อไคลเอนต์ต้องการใช้บริการใดๆ บนเครื่องเซิร์ฟเวอร์ ไคลเอนต์จะสร้างการเชื่อมต่อ (คอนเนคชั่น) ผ่านพอร์ตนั้นๆ จากนั้นจะเริ่มใช้บริการผ่านเส้นทางการสื่อสารดังกล่าว จนกระทั่งเมื่อไคลเอนต์ต้องการยกเลิกการเชื่อมต่อจึงสัญญาณยกเลิกการเชื่อมต่อไปยังเซิร์ฟเวอร์ เมื่อทั้งสองฝ่ายยกเลิกการเชื่อมต่อแล้วจึงถือว่าการให้บริการนั้นเสร็จสิ้นโดยสมบูรณ์

ปกติแล้วการสร้างการเชื่อมต่อแต่ละครั้งล้วนต้องใช้ทรัพยากรของระบบทั้งสิ้น เช่นหน่วยความจำที่ทำหน้าที่เป็นบัฟเฟอร์ หรือความสามารถของซีพียู รวมทั้งแบนด์วิดของเครือข่าย โดยในสภาวะปกติแล้วระบบสามารถจัดหาทรัพยากรเหล่านี้ได้อย่างพอเพียง แต่หากอยู่ในสภาวะที่ไม่ปกติ เช่นมีการโจมตีจากเครือข่าย จะส่งผลให้ทรัพยากรที่มีอยู่ถูกใช้ไปอย่างรวดเร็ว และส่งผลระบบไม่สามารถให้บริการต่อไปได้

เนื่องจากผลของการโจมตีดังกล่าวมีสูงมาก ทำให้ต้องมีการตรวจสอบข้อมูลในเครือข่ายอยู่ตลอดเวลาว่ามีการโจมตีหรือไม่ ทำให้เกิดแนวคิดการทำระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ (Network Intrusion Detection System : NIDS) ขึ้นมา เพื่อตรวจสอบการบุกรุกทางเครือข่ายและทำการแจ้งเตือนไปยังผู้ดูแลระบบ และเก็บข้อมูลการโจมตีดังกล่าวไว้ในล็อกไฟล์ เพื่อใช้ตรวจสอบได้ในภายหลัง

ปริญญานิพนธ์นี้จึงมุ่งเน้นการศึกษาประเภทต่างๆ ของการโจมตีแบบ DoS โดยจัดแบ่งประเภทของการโจมตี รวมทั้งศึกษาแนวทางการตรวจสอบการโจมตีในแต่ละประเภท เพื่อพัฒนาระบบตรวจจับบนระบบปฏิบัติการวินโดว์ให้สามารถตรวจจับการโจมตีดังกล่าวได้อย่างมีประสิทธิภาพ

**1.2 วัตถุประสงค์ของปริญญานิพนธ์**

ปริญญานิพนธ์ที่จัดทำขึ้นนี้ จัดทำภายใต้วัตถุประสงค์หลัก 4 ประการ ได้แก่

(1) เพื่อศึกษารายละเอียดและการทำงานของการบุกรุกทางเครือข่ายคอมพิวเตอร์

(2) เพื่อแบ่งประเภทของการบุกรุกทางเครือข่ายคอมพิวเตอร์

(3) เพื่อศึกษาแนวทางการตรวจสอบการบุกรุกทางเครือข่ายคอมพิวเตอร์

(4) เพื่อสร้างระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

**1.3 ขอบเขตของปริญญานิพนธ์**

ขอบเขตการทำงานของปริญญานิพนธ์นี้ ได้แก่

(1) จัดแบ่งประเภทของการบุกรุกให้เพื่อให้ปิดบริการสำหรับระบบเครือข่าย

(2) ออกแบบ และพัฒนาระบบการตรวจจับการบุกรุกแต่ละประเภทบนระบบปฏิบัติการวินโดว์

(3) ระบบที่สร้างขึ้นต้องสามารถตรวจจับ แจ้งเตือน และเก็บข้อมูลที่เกิดการบุกรุกได้อย่างถูกต้อง รวมทั้งยังสามารถพัฒนาต่อไปในอนาคตเพื่อรองรับการโจมตีแบบใหม่ๆ ได้

**1.4 ขั้นตอนการดำเนินงาน**

(1) ศึกษารายละเอียดเกี่ยวกับทีซีพี/ไอพีเบื้องต้น

(2) ศึกษาเกี่ยวกับระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

(3) ศึกษารายละเอียดและการทำงานของการการสำรวจเครือข่ายและโจมตีเพื่อให้ปิดบริการสำหรับสแต็กทีซีพี/ไอพี

(4) จัดแบ่งประเภทของการโจมตีเพื่อให้ปิดบริการสำหรับสแต็กทีซีพี/ไอพี

(5) กำหนดวิธีการตรวจจับการบุกรุกแต่ละประเภท

(6) ออกแบบโครงสร้างของระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

(7) ศึกษาการเขียนโปรแกรมผ่านเครือข่าย

(8) ออกแบบขั้นตอนการทำงานของระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

(9) พัฒนาระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

(10) ทดสอบและปรับปรุงระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

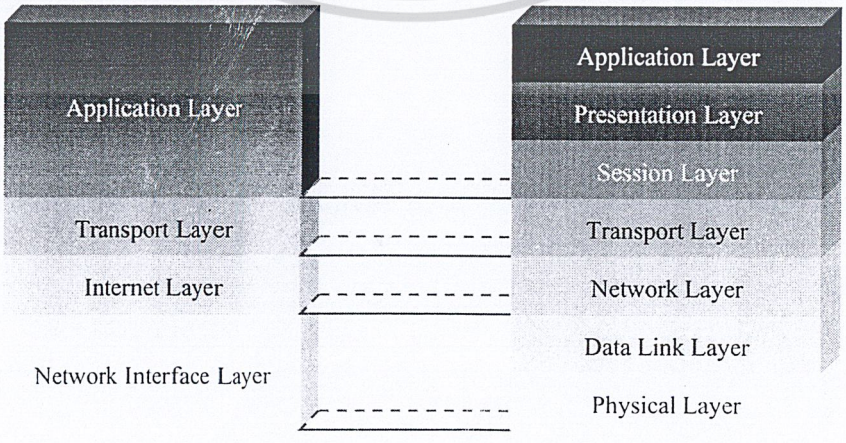
# บทที่ 2
# โพรโตคอลทีซีพี/ไอพี

## 2.1 ความเป็นมาของโพรโตคอลทีซีพี/ไอพี

ทีซีพี/ไอพี เป็นมาตรฐานการรับส่งข้อมูลระหว่างคอมพิวเตอร์สองระบบที่มีขึ้นเมื่อกระทรวงกลาโหมสหรัฐฯ หรือ Department Of Defense (DOD) ทำการทดลองในปี ค.ศ.1969 เชื่อมโยงคอมพิวเตอร์ทางทหารของแต่ละหน่วย ซึ่งเป็นคอมพิวเตอร์ต่างชนิดกันให้สามารถติดต่อรับส่งข้อมูลกันได้ โครงการนี้มีชื่อว่า Advanced Research Projects Agency Network หรือ ARPANET ซอฟต์แวร์ที่ใช้ควบคุมการรับส่งข้อมูลของ ARPANET ประกอบด้วยส่วนหลักๆ 2 ส่วน คือ ทีซีพี ( Transmission Control Protocol หรือ TCP ) และ ไอพี ( Internet Protocol หรือ IP ) ซึ่ง ทีซีพี มีหน้าที่ตรวจสอบการรับส่งข้อมูลระหว่างคอมพิวเตอร์ผู้รับและผู้ส่ง ให้ได้รับข้อมูลถูกต้องครบถ้วน ส่วนไอพีจะมีหน้าที่เลือกเส้นทางที่ใช้รับส่งข้อมูลผ่านระบบเครือข่าย และตรวจสอบที่แอดเดรสของผู้รับ เรียกว่าไอพีแอดเดรส (IP Address) ต่อมาในปี ค.ศ.1983 ทีซีพี/ไอพีถูกกำหนดให้เป็นมาตรฐานการรับส่งข้อมูลของกระทรวงกลาโหมสหรัฐฯ และได้รวมเป็นส่วนหนึ่งของระบบปฏิบัติการยูนิกซ์ ส่งผลให้มีการใช้งานกันอย่างกว้างขวาง ในปัจจุบันใช้งานอยู่ในแทบทุกเครือข่าย ไม่ว่าจะเป็นเครือข่ายเฉพาะที่หรือเครือข่ายในบริเวณกว้าง ทีซีพี/ไอพีเชื่อมกลุ่มเครือข่ายย่อยเข้าด้วยกันเป็นเครือข่ายขนาดใหญ่ หรือ อินเทอร์เน็ต (Internet)

## 2.2 การเชื่อมต่อของโพรโตคอลทีซีพี/ไอพี (TCP/IP Linking)

ทีซีพี/ไอพี (TCP/IP หรือ Transmission Control Protocol/Internet Protocol) เป็นโพรโตคอลในการสื่อสารในระบบอินเทอร์เน็ตและอินทราเน็ต การทำงานของทีซีพี/ไอพีสามารถเปรียบเทียบกับโมเดลอ้างอิงโอเอสไอ (Open System Interconnection Reference Model: OSI) ตามมาตรฐานไอเอสโอ (International Organization for Standardization: ISO) ได้ดังรูปที่ 2-1

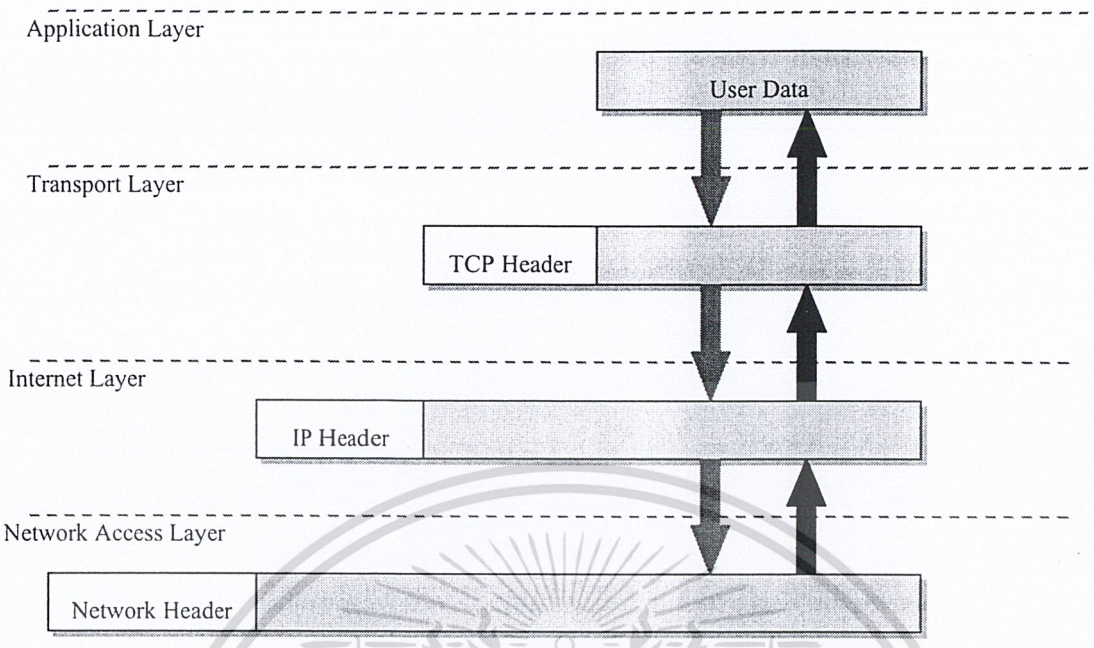



*รูปที่ 2-1 แสดงการเปรียบเทียบเลเยอร์ของโอเอสไอกับเลเยอร์ของทีซีพี/ไอพี*

ในแต่ละระดับชั้นของทีซีพี/ไอพีมีการทำงานที่แตกต่างกัน ตั้งแต่การติดต่อกับแอพพลิเคชันจนกระทั่งแปลงเป็นสัญญาณส่งไปตามสายสัญญาณ ซึ่งการทำงานในแต่ละระดับชั้นของทีซีพี/ไอพี มีดังตารางที่ 2-1

| ชื่อระดับชั้น | หน้าที่ |
|---|---|
| 1. ชั้นแอพพลิเคชัน (Application Layer) | ชั้นนี้รองรับการทำงานของแอพพลิเคชันต่างๆ ที่ทำงานเป็นโพรเซสอยู่ในเครื่องต้นทางและปลายทาง โดยจัดการเชื่อมต่อระหว่างโพรเซส หรือแอพพลิเคชันที่อยู่ต่างเครื่องกัน โดยการทำงานของแอพพลิเคชันต่างๆมีการติดต่อกันตามแต่ละโพรโตคอลเฉพาะแล้วแต่แอพพลิเคชันที่ใช้งาน ซึ่งจะขอบริการจากชั้นทรานสปอร์ตอีกทีหนึ่ง |
| 2. ชั้นทรานสปอร์ต (Transport Layer) | มีการสร้างการเชื่อมต่อกันระหว่างแอพพลิเคชันแบบ end-to-end โดยจุดที่เชื่อมต่อกันเพื่อรับส่งข้อมูลนี้เรียกว่า พอร์ต (port) หรือซ็อกเก็ต (Socket) ในชั้นนี้มีบริการหลักอยู่ 2 แบบ คือ Connection Oriented โดยเรียกผ่านโพรโตคอลทีซีพี (TCP: Transmission Control Protocol) และ Connectionless ซึ่งเรียกผ่านโพรโตคอลยูดีพี (UDP: User Datagram Protocol) ซึ่งกล่าวถึงในหัวข้อถัดไป |
| 3. ชั้นอินเทอร์เน็ต (Internet Layer) | ชั้นนี้มีหน้านี้ส่งผ่านข้อมูลระหว่างเครือข่าย โดยมีโพรโตคอลที่ทำงานเป็นกลไกสำคัญในการส่งผ่านข้อมูลไปยังเครือข่ายใดๆ ในอินเทอร์เน็ต คือ ไอพี (Internet Protocol: IP) ซึ่งกล่าวถึงในหัวข้อถัดไป นอกจากนี้ในชั้นนี้ยังมีโพรโตคอลทำงานอยู่ด้วยอีก 2 ชนิด คือ ไอซีเอ็มพี (Internet Control Message Protocol: ICMP) และเออาร์พี (Address Resolution Protocol: ARP) |
| 4. ชั้นเน็ตเวิร์กอินเตอร์เฟส (Network Interface Layer) | ทำหน้าที่ในการแปลงข้อมูลให้อยู่ในรูปที่เหมาะสมกับเครือข่ายแต่ละแบบ ซึ่งแตกต่างกันออกไป และแปลงเป็นสัญญาณไฟฟ้าส่งไปยังเครือข่าย |

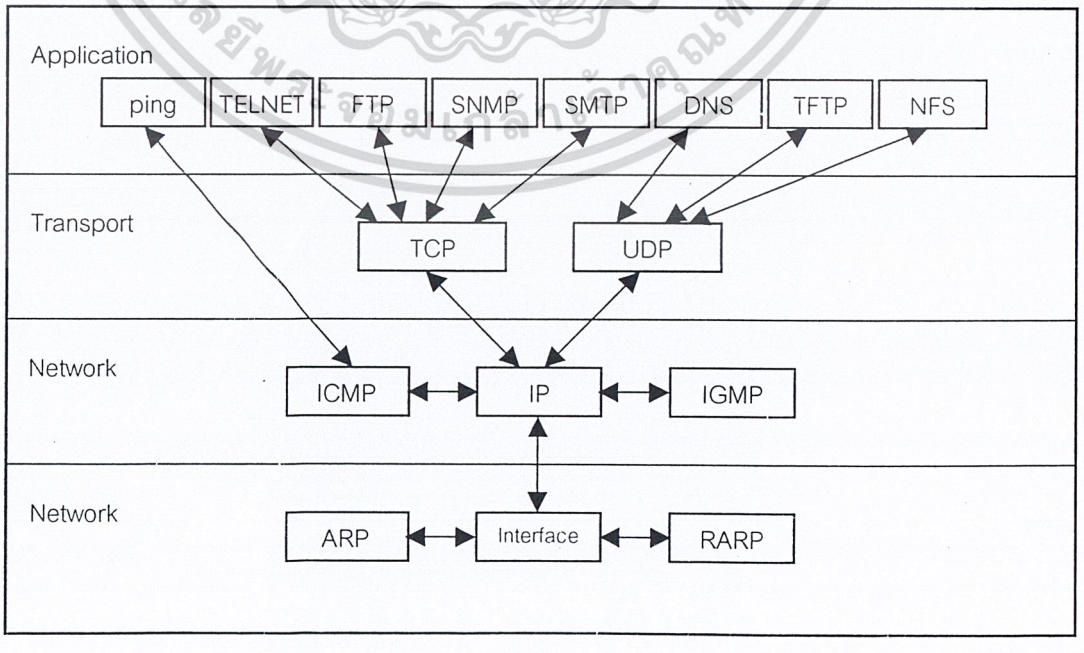
*ตารางที่ 2-1 การทำงานของแต่ละระดับชั้นของทีซีพี/ไอพี*




*รูปที่ 2-2 แสดงการข้อมูลที่ส่งผ่านในโมเดลของทีซีพี/ไอพี*

**2.3 โพรโตคอลสแต็ก**

การทำงานตามโปรแกรมประยุกต์หนึ่งๆไม่ได้ใช้โพรโตคอลพร้อมกันทั้งชุด หากแต่ใช้เพียงโพรโตคอลที่สัมพันธ์กันไปในแต่ละระดับชั้นของแบบอ้างอิง ตัวอย่างเช่นการใช้งานเทลเน็ต (Telnet) จะอาศัยทีซีพีและไอพี ตามลำดับ การซ้อนทับของโพรโตคอลจากระดับชั้นบนไปชั้นล่างเรียกว่าโพรโตคอลแสตค (Protocol Stack) ดังรูปที่ 2-3



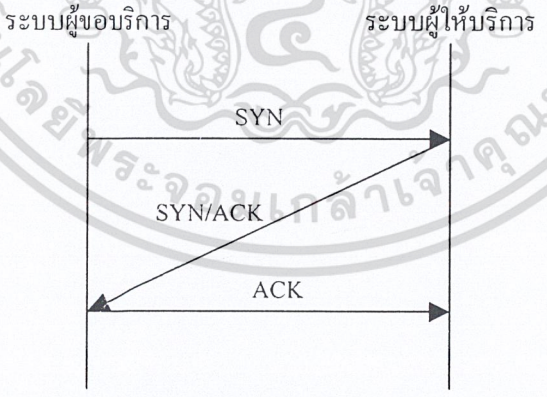

*รูปที่ 2-3 โพรโตคอลสสต็กของทีซีพี/ไอพี*

ไอพีซึ่งอยู่ในระดับชั้นเน็ตเวอร์คตามรูป เป็นแกนสำคัญของ โพรโตคอลแสตค เนื่องจากทั้ง ทีซีพี และ ยูดีพี ต้องใช้ไอพีเพื่อเลือกเส้นทางส่งแพ็กเก็ต ในระดับชั้นเน็ตเวอร์คยังมีไอซีเอ็มพีสนับสนุนการทำงานของไอพีเพื่อรายงานข้อผิดปกติที่เกิดขึ้นเนื่องจากการส่งแพ็กเก็ต และมีไอจีเอ็มพีดูแลการจัดกลุ่มโฮสต์ในเครือข่ายมัลติคาสต์ ระดับชั้นทรานสปอร์ตมี 2 โพรโตคอล ที่สำคัญ คือ ทีซีพีและยูดีพี แอพลิเคชันจะเลือกใช้ทีซีพีหรือยูดีพีตามลักษณะงาน โพรโตคอลระดับล่างถัดจากไอพีได้แก่ โพรโตคอล ระดับเน็ตเวอร์คอินเตอร์เฟสซึ่งกำหนดการทำงานตามเทคโนโลยีเครือข่ายที่ใช้งาน ในระดับชั้นนี้มี โพรโตคอลในชุดของ ทีซีพี/ไอพี ทำหน้าที่สนับสนุนการทำงานอยู่สอง โพรโตคอล คือ เออาร์พี และ อาร์เออาร์พี ทั้งสองโพรโตคอลทำหน้าที่แปลงค่าระหว่างแอดเดรสไอพี กับ ฮาร์ดแวร์แอดเดรส

ในชุดโพรโตคอลทีซีพี/ไอพีนี้ มีโพรโตคอลหลักที่ขอกล่าวถึง 5 โพรโตคอล ได้แก่ โพรโตคอลทีซีพี โพรโตคอลยูดีพี ซึ่งทำงานในชั้นทรานสปอร์ต และโพรโตคอลไอพี โพรโตคอลเออาร์พี โพรโตคอลไอซีเอ็มพี ซึ่งทำงานในชั้นอินเทอร์เน็ต โดยมีรายละเอียดดังต่อไปนี้

**2.4 โพรโตคอลทีซีพี (TCP: Transmission Control Protocol)**

การทำงานที่สำคัญอย่างหนึ่งของโพรโตคอลทีซีพี คือ การทำ "3-way Handshake" ซึ่งเป็นกระบวนการเริ่มต้นในการสร้างการเชื่อมต่อในชั้นทรานสปอร์ต กล่าวคือ ในการติดต่อกันระหว่างระบบในเครือข่ายต้องมีการสร้างการเชื่อมต่อไปยังระบบที่ให้บริการก่อน โดยผู้ขอบริการส่งสัญญาณ SYN เพื่อขอบริการ จากนั้นผู้ให้บริการจะส่งสัญญาณ ACK เพื่อตอบรับการเชื่อมต่อที่ร้องขอมาจึงสามารถรับส่งข้อมูลกันได้ ดังรูปที่ 2-4




*รูปที่ 2-4 แสดงการทำ 3-way Handshake*

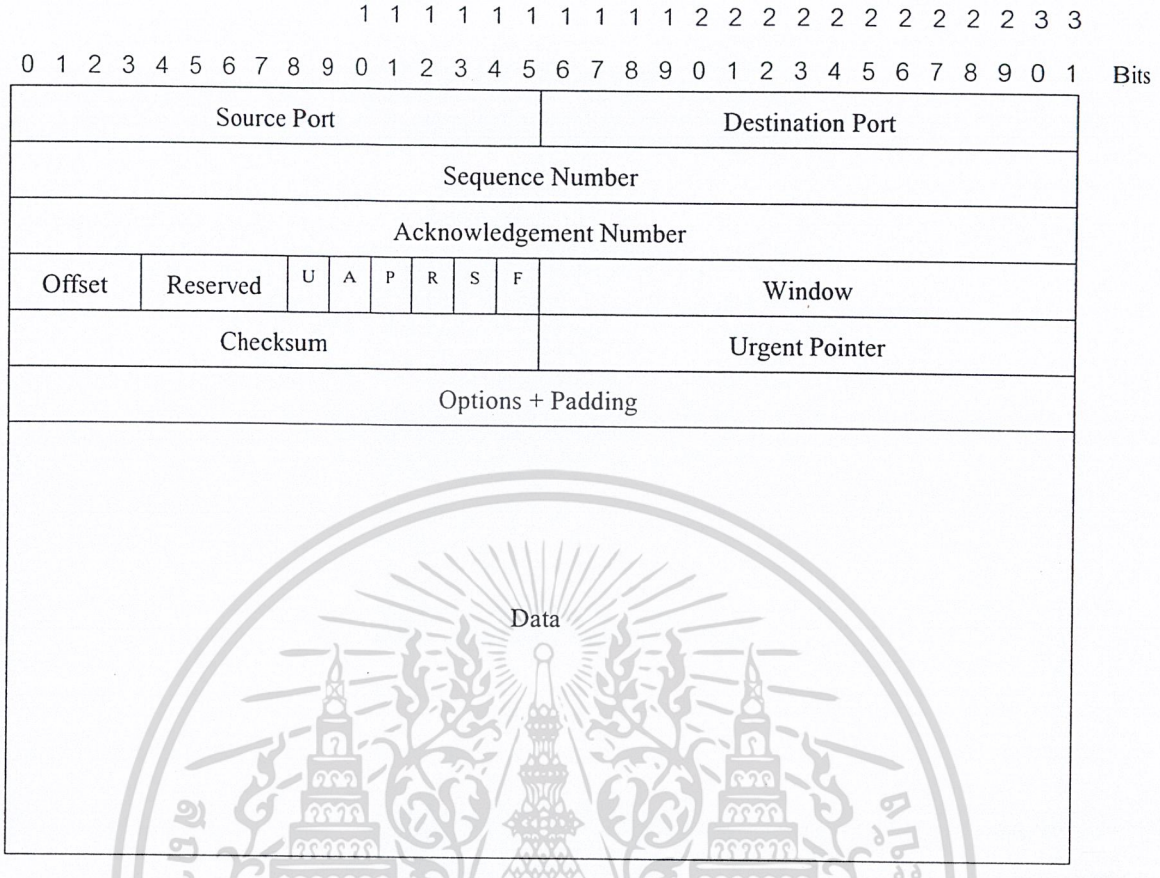
การเชื่อมต่อแบบ 3-way handshake นี้ เป็นการตรวจสอบความพร้อมของทั้งฝ่ายส่งและฝ่ายรับ และการกำหนดค่าเริ่มต้นของพารามิเตอร์ต่างๆ ของทั้งสองฝ่ายให้ตรงกัน หลังจากกระบวนการทำ 3-way handshake สิ้นสุด ทั้งสองฝ่ายจึงสามารถรับและส่งข้อมูลซึ่งกันและกันได้

ดังนั้นโพรโตคอลทีซีพีจึงเป็นโพรโตคอลที่มีการรับส่งข้อมูลแบบ “Connection Oriented” ทำให้การทำงานของทีซีพีมีความน่าเชื่อถือมากขึ้น หน้าที่การทำงานของทีซีพีในการรับส่งข้อมูลมีหน้าที่หลัก 6 ข้อคือ

1. ควบคุมการรับส่งข้อมูล (Basic Data Transfer)
2. ความน่าเชื่อถือในการรับส่งข้อมูล (Reliability)
3. ควบคุมการไหลของข้อมูล (Flow Control)
4. การทำมัลติเพล็ก (Multiplexing)
5. ควบคุมการเชื่อมต่อ (Connection)
6. ความปลอดภัยในการรับส่งข้อมูล (Security)

<u>ส่วนประกอบของทีซีพีเฮดเดอร์</u>

1. *Source Port* : เป็นหมายเลขพอร์ตของบริการที่เครื่องต้นทาง
2. *Destination Port* : เป็นหมายเลขพอร์ตของบริการเครื่องปลายทาง
3. *Sequence Number* : เป็นหมายเลขที่บอกลำดับของการรับส่งข้อมูลของเครื่องที่ต้องการขอส่งข้อมูล
4. *Acknowledgement Number* : เป็นหมายเลขที่บอกลำดับของการรับส่งข้อมูลที่ฝั่งรับข้อมูลปกติ ค่าของ Acknowledgement Number มีค่าเท่ากับ Sequence Number (ของอีกฝั่งหนึ่ง) + 1 เสมอ
5. *Data Offset* : เป็นตัวบอกค่าออฟเซตของข้อมูล เพราะทีซีพีนั้นไม่มีการกำหนดความยาวที่แน่นอนของข้อมูล จึงต้องมีออฟเซตเป็นตัวบอก
6. *Flag* : เป็นบิตที่บอกชนิดของข้อมูล ได้แก่
    - URG : Urgent Pointer Field Significant - แสดง Urgent Pointer
    - ACK : Acknowledgement Field Significant – แสดงการ Acknowledgement
    - PSH : Push Function
    - RST : Reset The Connection - แสดงเมื่อรีเซ็ทการเชื่อมต่อ
    - SYN : Synchronize Sequence Number - หมายเลขแพ็กเก็ตที่ส่งแบบซิงโครนัส
    - FIN : No more data from sender - แสดงว่าไม่มีข้อมูลที่ส่งจากผู้ส่งแล้ว
7. *Window* : เป็นเลขบอกจำนวนของอ็อกเต็ต (octet) ของข้อมูล จัดการในส่วนของ end-to-end flow control
8. *Checksum* : เป็นส่วนที่ตรวจสอบความถูกต้องของข้อมูล
9. *Urgent Pointer* : เป็นตัวชี้ตำแหน่งของ Urgent Data
10. *Option and Padding* : เป็นตัวบอกออปชันของโพรเซสที่ใช้ทีซีพี
11. *Data* : เนื้อข้อมูลที่ต้องการสื่อสาร มีขนาดได้ไม่ต่ำกว่า 5 32-บิตเวิร์ด (6 บิตแรกสงวนไว้และกำหนดให้เป็นศูนย์)




*รูปที่ 2-5 แสดงแพ็กเก็ตทีซีพี*

**2.5 โพรโตคอลยูดีพี (UDP: User Datagram Protocol)**

โพรโตคอลยูดีพีเป็นโพรโตคอลในการติดต่อสื่อสารในชั้นทรานสปอร์ต (Transport Layer) การทำงานคล้ายกับทีซีพีมาก คือ จัดการเกี่ยวกับการสื่อสารระหว่างเครื่อง แต่เป็นแบบ Connectionless คือ ทั้งฝ่ายส่งและฝ่ายรับไม่จำเป็นต้องอาศัยการสร้างช่องทางเชื่อมต่อกัน โดยไม่ต้องมีการแจ้งให้ฝ่ายรับข้อมูลเตรียมรับข้อมูลเหมือนโพรโตคอลทีซีพี และไม่มีการส่งสัญญาณตรวจสอบว่าข้อมูลถึงเครื่องปลายทางอย่างถูกต้องครบถ้วนในการส่งข้อมูลแต่ละครั้ง จึงไม่มีการส่งข้อมูลใหม่อีกในกรณีที่เกิดความผิดพลาดของการส่งข้อมูล

<u>ส่วนประกอบของ UDP Frame</u>

1. *Source Port* : เป็นค่าตัวเลข 16 บิต บอกพอร์ตของบริการที่เครื่องต้นทาง
2. *Destination Port* : เป็นค่าตัวเลข 16 บิต บอกพอร์ตของบริการที่เครื่องปลายทาง
3. *Length* : เป็นค่าตัวเลข 16 บิต บอกความยาวของข้อมูล
4. *Checksum* : เป็นค่าตัวเลข 16 บิต ตรวจสอบความถูกต้องของข้อมูลที่ส่ง

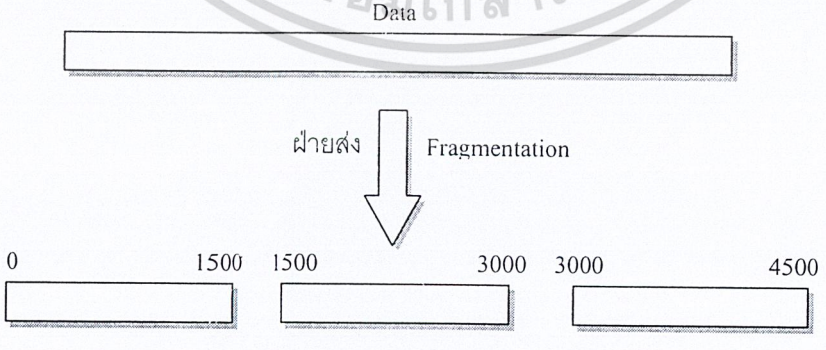
| 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 Bits | |
|---|---|
| 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 | |
| Source Port | Destination Port |
| Length | Checksum |
| Data | |

*รูปที่ 2-6 แสดงแพ็กเก็ตยูดีพี*

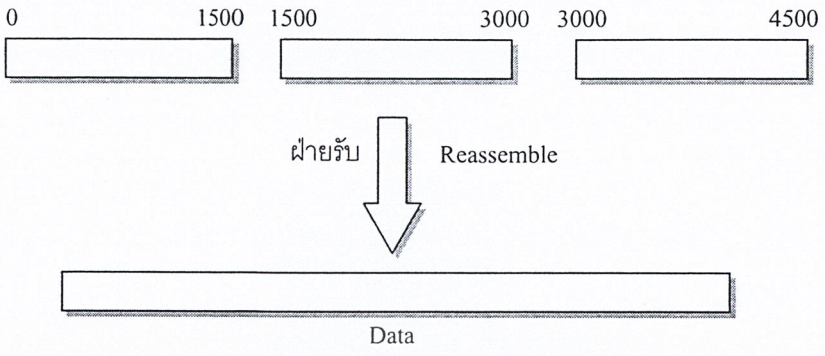
**2.6 โพรโตคอลไอพี (IP: Internet Protocol)**

โพรโตคอลไอพีเป็นโพรโตคอลที่จัดการเกี่ยวกับแอดเดรสของแต่ละแพ็กเก็ต เพื่อให้ส่งแพ็กเก็ตต่างๆ ไปยังเป้าหมายได้ถูกต้อง การทำงานของไอพีเป็นเพียงการส่งข้อมูลไปยังเครื่องเป้าหมายเท่านั้น ไม่มีการส่งสัญญาณขอบริการ หรือสัญญาณให้บริการระหว่างกันเหมือนทีซีพี เรียกว่าการเชื่อมต่อแบบ Connectionless ซึ่งระบบทั้งสองตั้งสมมติฐานว่าการเชื่อมต่อระหว่างกันไม่มีความผิดพลาดเกิดขึ้นแน่

เนื่องจากมาตรฐานในเครือข่ายมีหลากหลาย ขนาดของแพ็กเก็ตในแต่ละมาตรฐานจึงมีความแตกต่างกันออกไป ทำให้การส่งข้อมูลระหว่างอุปกรณ์ในเครือข่ายนั้นอาจมีการแบ่งข้อมูลออกเป็นแพ็กเก็ตย่อยๆ ในระหว่างการส่ง เรียกว่า การทำแฟร็กเมนเตชัน (Fragmentation) เช่น แพ็กเก็ตของ FDDI มีขนาด 4,500 ไบต์ หากเครื่องปลายอยู่ในเครือข่าย Ethernet ซึ่งมีขนาดของแพ็กเก็ตสูงสุดเพียง 1,500 ไบต์ ดังนั้นการส่งแพ็กเก็ตไปยังเครื่องปลายทางจึงต้องมีการแบ่งเป็นแพ็กเก็ตย่อย และเมื่อแพ็กเก็ตย่อยมาถึงเครื่องเป้าหมายก็จะมารวมกันเป็นแพ็กเก็ตเดิมที่มีขนาด 4,500 ไบต์อีกครั้ง เรียกการรวมกันนี้ว่า การรีแอสเซม-เบิ้ล (Reassemble) ซึ่งทำให้ได้ข้อมูลเหมือนที่ส่งมาจากเครื่องต้นทาง




*รูปที่ 2-7 แสดงการทำแฟร็กเมนเตชัน*




*รูปที่ 2-8 แสดงการรีแอสเซมเบิ้ล*

ส่วนประกอบของแพ็กเก็ตไอพี

1. *version* : เป็นค่าตัวเลข 4 บิต บอกเวอร์ชันของมาตรฐานไอพีที่ใช้ โดยปกติมีค่าเป็น 4 ซึ่งหมายถึง IPv4
2. *Internet Header Length (IHL)* : เป็นตัวบอกความยาวเฮดเดอร์ของไอพี
3. *Type of Service* : เป็นส่วนที่บอกการทำงานของแพ็กเก็ตที่ส่งว่าทำหน้าที่อะไร มีทั้งหมด 8 บิต โดย

   Bit 0-2 : บอกรายละเอียดการทำงานของแพ็กเก็ตนั้นๆ

   111 - Network Control

   110 - Internetwork Control

   101 - CRITIC / ECP

   100 - Flash Override

   011 - Flash

   010 - Immediate

   001 - Priority

   000 - Routine

   Bit 3 : บอกถึงลักษณะของดีเลย์

   0 = Normal Delay - มีดีเลย์ปกติ

   1 = Low Delay - มีดีเลย์ต่ำ

   Bit 4 : บอกถึงประเภทของทรูพุต

   0 = Normal Throughput – มีทรูพุตปกติ

   1= High Throughput – มีทรูพุตสูง

Bit 5 : บอกถึงประเภทของความน่าเชื่อถือ

0 = Normal Reliability - มีความน่าเชื่อถือพอประมาณ

1 = High Reliability - มีความน่าเชื่อถือสูง

Bit 6-7 : กันไว้ใช้ในอนาคต

4. *Total Length* : มีขนาด 16 บิต บอกถึงความยาวในดาต้าแกรมของไอพี
5. *Identification field* : เป็นตัวเลข 16 บิต เป็นค่าประจำตัวของไอพีนั้น โดยโฮสต์ที่ส่งเป็นผู้กำหนด และเพิ่มค่าขึ้นหนึ่งเมื่อมีการส่งดาต้าแกรมของไอพีใหม่ ซึ่งใช้ในการประกอบกลับ
6. *Flag* : เป็นตัวเลข 3 bit บอกลักษณะของแพ็กเก็ตว่ามีการแฟร็กเมนต์หรือไม่

Bit 0 : สงวนไว้ ปกติเป็น 0

Bit 1 : 0 = บอกว่าแพ็กเก็ตมีการแตกแพ็กเก็ตย่อย

1 = บอกว่าแพ็กเก็ตไม่มีการแตกแพ็กเก็ตย่อย

Bit 2 : 0 = บอกว่าแพ็กเก็ตนั้นเป็นแพ็กเก็ตสุดท้ายที่ได้จากการแตกแพ็กเก็ตย่อย

1 = บอกว่าแพ็กเก็ตนั้นยังไม่ใช่แพ็กเก็ตสุดท้ายที่ได้จากการแตกแพ็กเก็ตย่อย

7. *Fragment Offset* : เป็นค่าตัวเลข 13 บิต บอกออปเซตของแฟร็กเมนต์เมื่อเทียบในดาต้าแกรม
8. *Time To Live (TTL)* : เป็นตัวเลข 8 บิต บอกช่วงเวลาของแพ็กเก็ตที่ยังอยู่ในเครือข่ายได้ โดยกำหนดค่าเป็นจำนวนเราท์เตอร์สูงสุดที่ดาต้าแกรมผ่านได้ ซึ่งโดยทั่วไปทีค่าระหว่าง 32 ถึง 64 และลดค่าลงเรื่อยๆ เมื่อผ่านเราเตอร์ เพื่อเป็นการป้องกันแพ็กเก็ตล้นเครือข่าย
9. *Protocol* : เป็นตัวเลข 8 bit บอกถึงโพรโตคอลที่อยู่เหนือขึ้นไป ว่าเป็นโพรโตคอลระดับสูงกว่าประเภทใด
10. *Header Checksum* : เป็นค่าตัวเลข 32 บิต ใช้ตรวจสอบความถูกต้องของเฮดเดอร์
11. *Source Address* : เป็นค่าตัวเลข 32 บิต บอกถึงไอพีแอดเดรสของเครื่องต้นทาง
12. *Destination Address* : เป็นค่าตัวเลข 32 บิต บอกถึงไอพีแอดเดรสของเครื่องปลายทาง

| 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 | |
|---|---|
| 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 | Bits |

| Ver | IHL | Type of Service | Total Length | |
|---|---|---|---|---|
| Identifier | | | Flags | Fragment |
| Time to Live | | Protocol | Header Checksum | |
| Source Address | | | | |
| Destination Address | | | | |
| Options + Padding | | | | |
| Data | | | | |

*รูปที่ 2-9 แสดงแพ็กเก็ตไอพี*

**2.7 โพรโตคอลเออาร์พี( ARP:Address Resolution Protocol )**

โพรโตคอลเออาร์พีเป็นโพรโตคอลที่ออกแบบมาเพื่อใช้ในเครือข่ายที่สนับสนุนการบอร์ดคาสต์ ถูกเรียกใช้งานโดยโพรโตคอลไอพีเพื่อช่วยแปลงหมายเลขไอพี ไปเป็นหมายเลขฮาร์ดแวร์ปลายทาง ตัวอย่างเช่น เว็ปเซิร์ฟเวอร์เครื่องหนึ่งเชื่อมต่ออยู่ในเครือข่ายอินเทอร์เน็ต และในการเชื่อมต่อนี้ต้องอาศัยการ์ดแลน(LAN card) ติดตั้งอยู่ ที่แลนการ์ดนี้จะมีหมายเลขเฉพาะประจำฮาร์ดแวร์ที่ไม่ซ้ำกับใคร เพื่อใช้อ้างอิงการส่งข้อมูลในเครือข่าย แต่เมื่อมาใช้งานใน โพรโตคอล ทีซีพี/ไอพี ก็จะต้องมีการกำหนดหมายเลขแอดเดรสไอพี ประจำตัวเพื่อใช้อ้างอิงกัน และ โพรโตคอลเออาร์พี จะทำหน้าที่แปลงค่าหมายเลขไอพีให้เป็นหมายเลขฮาร์ดแวร์จริงในระดับการทำงานที่ชั้นอินเทอร์เน็ตนี้ ซึ่งกลไกการแปลงนี้เรียกว่า address resolution

| hardware | | protocol |
|---|---|---|
| HLEN | PLEN | operation |
| Sender HA ( octets 0-3 ) | | |
| Sender HA ( octets 4-5 ) | | Sender IA ( octets 0-1 ) |
| Sender IA ( octets 2-3 ) | | Target HA ( octets 0-1 ) |
| Target HA ( octets 2-5 ) | | |
| Target IA ( octets 0-3 ) | | |

*รูปที่ 2-10 เออาร์พีดาทาแกรม*

ส่วนประกอบของเออาร์พีดาทาแกรม

1. *Hardware 16 บิต* : กำหนดชนิดของฮาร์ดแวร์เครือข่ายที่เออาร์พีทำงานอยู่ ค่าใช้งานมีตัวอย่างดังต่อไปนี้

1 อีเทอร์เน็ต

4 โทเค็นริง

5 เคออส( chaos )

6 เครือข่าย IEEE 802

7 อาร์คเน็ต

12 โลคัลทอล์ค

2. *protocol 16 บิต* : ชนิดของโพรโตคอลที่ร้องขอใช้เออาร์พี

3. *HLEN 8 บิต* : ขนาดของฮาร์ดแวร์แอดเดรสเป็นจำนวนไบต์ ค่าปกติที่ใช้งาน คือ 6 ซึ่งเท่ากับขนาด 6 ไบต์ของอีเทอร์เน็ตฮาร์ดแวร์แอดเดรส

4. *PLEN 8 บิต* : ขนาดของแอดเดรสระดับเน็ตเวอร์คเป็นจำนวนไบต์ ค่าปกติที่ใช้ คือ 4 ซึ่งเท่ากับขนาด 4 ไบต์ของไอพีแอดเดรส

5. *Operation 16 บิต* : กำหนดรูปแบบการใช้ดาทาแกรม ค่าในฟิลด์นี้ใช้กำหนดการทำงานของทั้งเออาร์พีและอาร์เออาร์พี ซึ่งมี 4 ค่า คือ

ARP request ( ค่าเท่ากับ 1 )

ARP reply ( ค่าเท่ากับ 2)

RARP request ( ค่าเท่ากับ 3)

RARP reply ( ค่าเท่ากับ 4 )

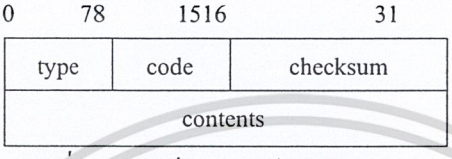
6. *Address* : ฟิลด์แอดเดรสเรียงลำดับจากฮาร์ดแวร์และเน็ตเวอร์คแอดเดรสของสถานีที่ร้องขอ ตามด้วยฮาร์ดแวร์และเน็ตเวอร์คแอดเดรสของสถานีที่ตอบรับ

**2.8 โพรโตคอล ไอซีเอ็มพี (ICMP : Internet Control Message Protocol )**

หน้าที่หลักของ โพรโตคอล ไอซีเอ็มพี คือการแจ้งหรือแสดงข้อความจากระบบ เพื่อบอกให้ผู้ใช้ทราบว่าเกิดอะไรขึ้นในการส่งผ่านข้อมูลนั้น ซึ่งปัญหาส่วนมากที่พบ คือส่งไปไม่ได้ หรือปลายทางรับข้อมูลไม่ได้ เป็นต้น นอกจากนี้ โพรโตคอล ไอซีเอ็มพี ยังถูกเรียกใช้งานจากเครื่องเซิร์ฟเวอร์ และ เราท์เตอร์ อีกด้วย เพื่อแลกเปลี่ยนข้อมูลที่ใช้ควบคุม ส่วนรูปแบบการทำงานของโพรโตคอลไอซีเอ็มพีนั้นจะทำงานควบคู่กับโพรโตคอลไอพีในระดับเดียวกัน และข้อความต่างๆที่แจ้งให้ทราบจะถูกผนึกอยู่ภายในข้อมูลของไอพี( ไอพีดาทาแกรม ) อีกทีหนึ่ง ข้อความที่โพรโตคอลไอซีเอ็มพีส่งนั้น แบ่งออกได้ 2 แบบ คือ ICMP error message หรือข้อความแจ้งข้อผิดพลาด และ ICMP query หรือข้อความเรียกขอข้อมูลเพิ่ม ตัวอย่างกลไกการทำงานของ โพรโตคอล ไอซีเอ็มพี เช่น เมื่อมีการส่งผ่านข้อมูลจากผู้ใช้ไปยังปลายทางที่ไม่ถูกต้อง หรือขณะนั้นเครื่องปลายทางเกิดปัญหาจนไม่สามารถรับข้อมูลได้ ที่เราท์เตอร์จะส่งข้อความแจ้งเป็น ไอซีเอ็มพี message ที่ชื่อ destination unreachable ให้กับผู้ส่งข้อมูล นอกจากนี้ตัวข้อมูลที่แจ้งข้อ

ความก็จะมีส่วนของข้อมูลไอพีดาทาแกรมที่เกิดปัญหาด้วย ดังนั้นเมื่อผู้ส่งข้อมูลได้รับข้อความแจ้งแล้วก็จะได้ทราบว่าจุดที่เกิดปัญหานั้นอยู่ที่ใด

ดังนั้นโพรโตคอล ไอซีเอ็มพี จึงกลายมาเป็นเครื่องมืออย่างหนึ่งในการช่วยทดสอบเครือข่าย เช่น คำสั่ง ping ที่เรามักใช้ทดสอบว่าเครื่องเซิร์ฟเวอร์ที่ให้บริการหรืออุปกรณ์ที่ต่ออยู่ในเครือข่ายอินเทอร์เน็ตนั้นยังทำงานเป็นปกติหรือไม่ แล้วคำสั่ง ping มีการเรียกใช้งานโพรโตคอล ไอซีเอ็มพี แจ้งเป็นข้อความให้ทราบอีกต่อหนึ่ง




*รูปที่ 2-11 ฟอร์แมตของ ไอซีเอ็มพี*

1. *Type ขนาด 8 บิต* : กำหนดค่าความผิดพลาดและการรายงานสถานะ การใช้งานในปัจจุบันมีทั้งหมด 15 ประเภท
2. *code ขนาด 8 บิต* : รหัสความผิดพลาดย่อย
3. *Checksum ขนาด 16 บิต* : ค่าผลรวมตรวจสอบแบบ 1's complement สำหรับใช้ตรวจสอบความผิดพลาด โดยคำนวณผลรวมของ type,code และ contents
4. *Contents ขนาดไม่คงที่* : ฟิลด์นี้ใช้บรรจุข้อมูลข่าวสารเพิ่มเติมเพื่อแจ้งกลับซึ่งจะขึ้นอยู่กับค่า typeและ code

# บทที่ 3
# การสำรวจระบบ

## 3.1 ความหมายของการสำรวจระบบ

การสำรวจระบบคือการเก็บรวบรวมข้อมูลของระบบเป้าหมาย เพื่อหาจุดอ่อนสำหรับการโจมตี ตัวอย่างข้อมูลที่เป็นประโยชน์สำหรับการโจมตี เช่น หมายเลขไอพี, พอร์ตที่เปิดให้บริการ, ลักษณะของระบบปฏิบัติการ, เวอร์ชั่นของแอพพลิเคชั่นที่ถูกติดตั้งอยู่ เป็นต้น

ประโยชน์ของการเก็บรวบรวมข้อมูลสำหรับใช้ในการโจมตีคือ ผู้โจมตีสามารถเลือกใช้งานเครื่องมือที่ใช้โจมตีได้อย่างเหมาะสม เช่น หากทราบว่าเครื่องเป้าหมายเป็นระบบปฏิบัติการวินโดว์สที่เปิดให้บริการ Netbios (พอร์ต 139) ผู้โจมตีอาจเลือกใช้เครื่องมือโจมตีที่เหมาะสมกับระบบปฏิบัติการวินโดว์สโดยกำหนดให้โจมตีที่พอร์ต 139 เป็นต้น ผลจากการโจมตีโดยมีข้อมูลของระบบและใช้เครื่องมือที่เหมาะสอ จะก่อให้เกิดผลกระทบมากกว่าการโจมตีโดยไม่ทราบข้อมูลของระบบและทำการโจมตีโดยใช้เครื่องมือที่ไม่เหมาะสม ข้อมูลที่ผู้โจมตีมักต้องการทราบได้แก่ หมายเลขไอพีคอมพิวเตอร์ที่ต่อเชื่อมอยู่กับเครือข่าย, โครงสร้างและการจัดการเครือข่าย, ชื่อของบริการที่เปิดอยู่ หรือแม้กระทั่งชื่อของระบบปฏิบัติการ เป็นต้น โดยการสำรวจแต่ละอย่างจำเป็นจะต้องใช้เครื่องมือที่แตกต่างกัน

## 3.2 เครื่องมือที่ใช้ในการสำรวจระบบ

ในขั้นตอนการสำรวจเพื่อเก็บข้อมูลนั้น จำเป็นจะต้องใช้เครื่องมือหรือโปรแกรมที่มีประสิทธิภาพ เพื่อให้ได้ข้อมูลที่ถูกต้องและรวดเร็ว ในบทนี้จะนำเสนอประเภทของเครื่องมือที่มีการใช้ในการสำรวจระบบ

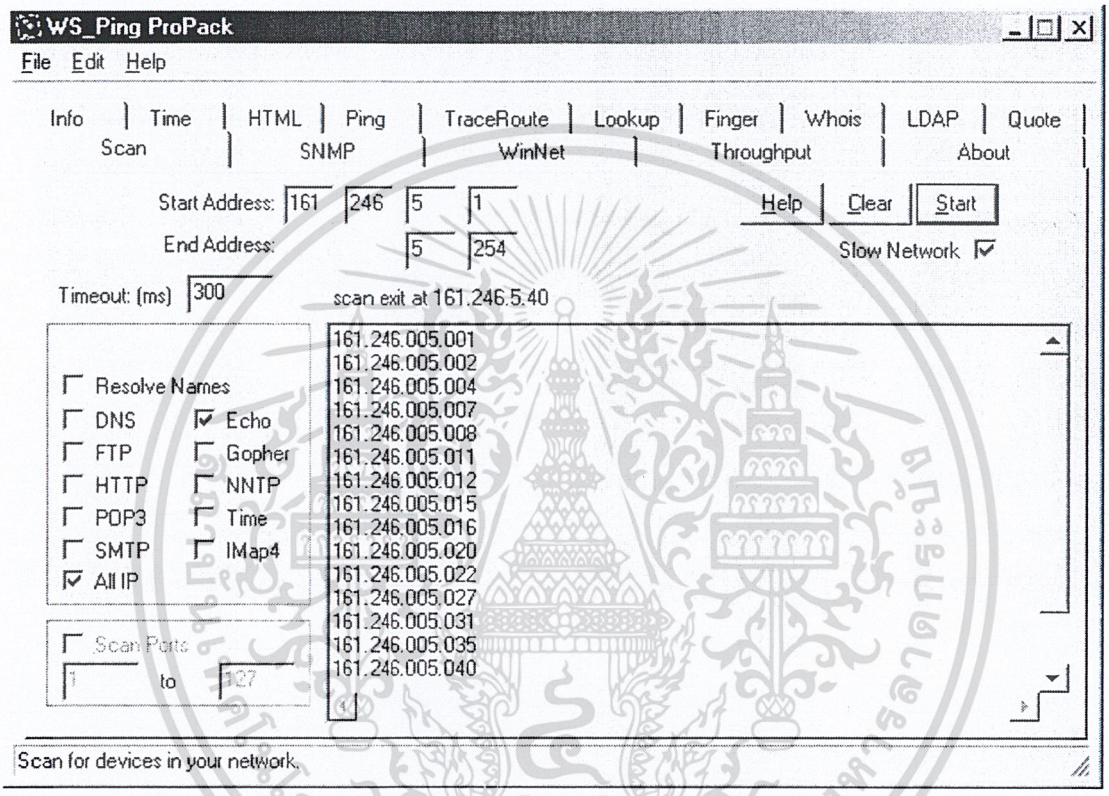
### 3.2.1 ปิงสวีป

คำสั่งปิงเป็นเครื่องมือพื้นฐานที่นิยมใช้สำหรับตรวจสอบว่าคอมพิวเตอร์เครื่องนั้นๆ ทำงานอยู่ในเครือข่ายหรือไม่ ลักษณะของการปิงคือการส่งแพ็กเก็ต ICMP Echo (Type 8) ไปยังเครื่องเป้าหมายและรอ ICMP Echo Reply (Type 0) ที่จะถูกส่งกลับมาจากเครื่องเป้าหมาย แม้ว่าการปิงในลักษณะนี้จะสามารถตรวจสอบคอมพิวเตอร์ในเครือข่ายเป้าหมายได้ แต่หากเป็นเครือข่ายขนาดใหญ่จะพบว่าประสิทธิภาพที่ได้นั้นต่ำมาก และไม่สะดวกในการใช้งาน เนื่องจากโดยทั่วไปแล้วคำสั่งปิงไม่สามารถกำหนดให้ตรวจสอบเป้าหมายแบบกลุ่มได้

การปิงสวีปเป็นการสำรวจเป้าหมายโดยการปิงไปยังเป้าหมายจำนวนมากพร้อมกัน เพื่อตรวจสอบว่าเครื่องใดบ้างในเครือข่ายเป้าหมายที่เชื่อมต่ออยู่กับคอมพิวเตอร์ และยังใช้งานได้ง่ายอีกด้วย เนื่องจากเครื่องมือที่ทำหน้าที่ปิงสวีปมักจะมีความสามารถกำหนดกลุ่มเป้าหมายในลักษณะของกลุ่มเครือข่ายย่อย (Sub Network) ได้ หรือแม้แต่สามารถสร้างรายการ (List) ของเครื่องเป้าหมายในรูปแบบแฟ้มข้อมูล

เพื่อเป็นข้อมูลสำหรับการปิงสวีปได้ ตัวอย่างของโปรแกรมปิงสวีปที่ได้รับความนิยม เช่น Gping, Nmap, WS_Ping เป็นต้น

ข้อมูลที่ได้จากการปิงสวีปก็คือ ผู้โจมตีจะได้หมายเลขไอพี หรือแม้แต่ชื่อของคอมพิวเตอร์ที่อยู่ในเครือข่ายเป้าหมาย และหากการตั้งชื่อเครื่องสอดคล้องกับหน้าที่ที่คอมพิวเตอร์ทำงานอยู่ ผู้โจมตีก็สามารถใช้การสำรวจเส้นทางเดินของข้อมูลโดยใช้คำสั่ง Tracroute เพื่อให้ทราบโครงสร้างของเครือข่าย (Network Topology) โดยสังเขปได้




*รูปที่ 3-1 โปรแกรม WS_Ping ทำการปิงสวีปไปทุกเครื่องในเครือข่าย*

### 3.2.2 พอร์ตสแกน

เป็นการตรวจสอบว่าเครื่องที่เป้าหมายนั้นเปิดให้บริการอะไรบ้าง เนื่องจากบริการต่างๆ จะรอรับอยู่ที่พอร์ต โดยปกติจะมีค่าเป็นมาตรฐาน เช่น บริการรับส่งไฟล์ (เอฟทีพี) จะให้บริการที่พอร์ต 21 หรือบริการเทอร์มินอล (เทลเน็ต) จะให้บริการที่พอร์ต 23 เป็นต้น ดังนั้นหากทราบหมายเลขพอร์ตที่เปิดรอการเชื่อมต่ออยู่ ก็จะทราบชื่อบริการที่มีอยู่ในเครื่องนั้นๆ ได้ วิธีการสแกนพอร์ตสามารถแบ่งได้ 2 วิธี คือ

#### 3.2.2.1 สแกนแบบพาสซีฟ

เป็นการสแกนโดยผู้สแกนไม่จำเป็นต้องติดต่อกับเครื่องเป้าหมายโดยตรง เช่นการแอบมองแพ็กเก็ตที่ส่งออกมาจากเครื่องเป้าหมาย หรืออาจทำได้โดยใช้เทคนิคการปลอมหมายเลขไอพี (IP Spoofing) ร่วมกับการหาความสัมพันธ์ของซีเควนซ์นัมเบอร์ เป็นต้น สำหรับการสแกนประเภทนี้จะไม่ขออธิบายใน

รายละเอียด เนื่องจากได้รับความนิยมน้อย และในโครงงานไม่ได้ออกแบบสำหรับให้ทำการตรวจสอบการสแกนพอร์ตจากวิธีการแบบพาสซีฟ

#### 3.2.2.1 สแกนแบบพาสซีฟ

เป็นการสแกนโดยผู้สแกนติดต่อกับเครื่องเป้าหมายโดยตรงผ่านโพรโตรคอลไอพี สามารถแบ่งย่อยได้อีกหลายเทคนิค ดังนี้

- **TCP SYN Scan** วิธีนี้ผู้สแกนจะทำการส่ง SYN แพ็กเก็ต (เซตค่าแฟล็ก SYN ไว้เป็น 1) เพื่อทำการติดต่อโดยตรงกับเป้าหมายโดยไม่ผ่านระบบปฏิบัติการ และรอผลการตอบรับของเป้าหมายกลับมา ซึ่งหากเป้าหมายทำงานอยู่ก็จะตอบกลับมาด้วย SYN ACK (เป็นแพ็กเก็ตที่ เซตค่าแฟล็ก SYN และ ACK ไว้เป็น1 ) หรือหากไม่มีแอพพลิเคชันทำงานอยู่จะตอบกลับมาด้วย RST การสแกนแบบนี้หากตรวจสอบบนโฮสต์เป้าหมายจะพบว่ามีการขอเชื่อมต่อเข้ามา แต่ไม่สามารถเปิดการติดต่อได้สำเร็จ เทคนิคนี้บางครั้งถูกเรียกว่า half-open scanning คือไม่สามารถทำ 3-way handshake ได้ จึงไม่มีการเชื่อมต่อใดๆเกิดขึ้นระหว่างเครื่องผู้สแกน กับเครื่องที่ถูกสแกน
- **FIN Scan** เป็นการส่ง FIN แพ็กเก็ตไปยังเป้าหมาย โดยที่เครื่องเป้าหมายก็จะยังตอบแพ็กเก็ตนั้นกลับไป แม้จะไม่มีการสื่อสารใดๆมาก่อนก็ตาม ซึ่งโดยปกติแล้วแพ็กเก็ตที่เซตค่า FIN เป็น1 จะเป็นแพ็กเก็ตที่ใช้ในการตอบกลับ และการตอบกลับของเครื่องเป้าหมายสำหรับพอร์ตที่เปิดไว้ และพอร์ตที่ไม่ได้เปิดให้บริการก็ไม่เหมือนกัน หากเป็นพอร์ตที่เปิดอยู่ก็จะตอบด้วย FIN ACK กลับไป และหากเป็นพอร์ตที่ไม่ได้เปิดก็จะตอบด้วย RST ACK
- **SYN/FIN Scan** วิธีนี้จะใช้ TCP Flag ทั้ง SYN และ FIN พร้อมกัน ซึ่งปกติเป็นแฟล็กที่ไม่มีกำหนดไว้ในโปรโตคอล และจะไม่พบแฟล็กเช่นนี้ในการสื่อสารตามปกติเป็นอันขาด เพราะโดยปกติแล้ว SYN Flag จะใช้เมื่อเริ่มการติดต่อ ส่วน FIN จะใช้เมื่อต้องการยุติการติดต่อ การตอบรับของโฮสต์แต่ละประเภทในกรณีที่ทำงานอยู่นั้นอาจจะแตกต่างกันไป เช่น เป็น SYN ACK หรือ FIN ACK อย่างใดอย่างหนึ่ง ส่วนการตอบรับในกรณีที่พอร์ตปิดจะตอบเหมือนกันคือ RST
- **Null Scan** วิธีนี้จะไม่ใช้แฟล็กใดๆในการสแกนเลย โดยส่งแพ็กเก็ตที่ไม่มีแฟล็กใดที่ถูกเซตไว้เลยไปยังเป้าหมาย เป็นการเซ็ตแฟล็กทุกค่าให้เป็น 0 หมด ซึ่งแพ็กเก็ตลักษณะนี้จะไม่มีอยู่ในโปรโตคอล โดยทั่วไปการตอบสนองแพ็กเก็ตที่ไม่ได้อยู่ในโปรโตคอล จะมีการตอบรับที่ต่างกันออกไปตามแต่ประเภทของระบบปฏิบัติการ ดังนั้นนอกจากการใช้แพ็กเก็ตเหล่านี้เพื่อการสแกนพอร์ตแล้วยังสามารถนำแพ็กเก็ตเหล่านี้ไปใช้ในการตรวจสอบระบบปฏิบัติการของเป้าหมายได้อีกด้วย โดยการส่งแพ็กเก็ตที่มีแฟล็กซึ่งไม่อยู่ในข้อกำหนด การส่งแพ็กเก็ตลักษณะนี้ หากพอร์ตของเครื่องเป้าหมายปิดอยู่ การตอบรับจะเป็นการส่ง RST กลับไป

- **X'mas Scan** จะเป็นการส่งแพ็กเก็ต TCP ที่เซ็ตแฟล็ก FIN, Push, URGENTไปยังพอร์ตเป้าหมายที่เครื่องปลายทาง ซึ่งมักไม่เป็นที่สนใจในการตรวจสอบเท่ากับ SYN-ACK-RST เครื่องปลายทางจะส่งแพ็กเก็ต TCP RST ของพอร์ตที่ปิดอยู่กลับมาให้

- **UDP Scan** จะส่งแพ็กเก็ตของโปรโตคอล UDP ไปยังพอร์ตเป้าหมาย แต่เนื่องจาก UDPมีการจัดการที่แตกต่างจาก TCPโดยโปรโตคอล UDP เป็นโปรโตคอลลักษณะคอนเนกชันเลส (connectionless) ดังนั้นผลลัพธ์ของการสแกนเมื่อพอร์ตเปิดอยู่นั้นจะไม่สามารถคาดการณ์ได้ ขึ้นอยู่กับแต่ละแอพพลิเคชัน และไม่มีมาตรฐานที่เหมือนกันแต่อย่างใด ดังนั้นการสแกน UDP จึงต้องดูผลลัพธ์จาก ICMP เป็นหลัก หากพอร์ตไม่เปิดให้บริการ จะมี ICMP Message ว่า UDP Port Unreachable กลับมา และหากพอร์ตเปิดให้บริการ อาจมีการตอบรับหรือไม่และอย่างไร จะขึ้นอยู่กับการทำงานของแอพพลิเคชันที่เปิดพอร์ตนั้น แต่ที่แน่นอนคือจะไม่มี ICMP Message กลับมา

### 3.2.3 การตรวจสอบระบบปฏิบัติการ

เป็นเครื่องมือที่ใช้สำหรับการตรวจสอบว่าเครื่องเป้าหมายใช้ระบบปฏิบัติการใด เนื่องจากจุดอ่อนของระบบปฏิบัติการแต่ละตัวมีไม่เหมือนกัน หากผู้โจมตีทราบว่าระบบปฏิบัติการของเครื่องเป้าหมายเป็นระบบปฏิบัติการใด ย่อมส่งผลให้สามารถเลือกใช้เครื่องมือในโจมตีที่ตรงกับระบบปฏิบัติการนั้นได้ ทำให้ผลจากการโจมตีเกิดได้มากกว่าการใช้เครื่องมือที่ทำงานได้ไม่ตรงกับระบบปฏิบัติการ

การตรวจสอบระบบปฏิบัติการมีเทคนิคที่ใช้หลายวิธี ตั้งแต่วิธีที่ไม่ต้องใช้เครื่องมือใดๆ เพียงแค่มีโปรแกรมเท็ลเน็ต ตัวอย่างเช่นการตรวจสอบจากแบนเนอร์ (Banner) เนื่องจากแบนเนอร์จะเป็นข้อความที่ใช้แสดงการตอบรับเมื่อทำการเชื่อมต่อสำเร็จเพื่อบอกให้ไคลเอนต์ทราบว่าเครื่องเซิร์ฟเวอร์ใช้ระบบปฏิบัติการใดอยู่ ดังนั้นในแบนเนอร์จึงมีการแสดงชื่อระบบปฏิบัติการด้วย ทำให้สามารถตรวจสอบได้ว่าเครื่องดังกล่าวใช้ระบบปฏิบัติการใด นอกจากนี้การสแกนพอร์ตยังสามารถตรวจสอบคร่าวๆ เช่นกันว่าเป้าหมายใช้ระบบปฏิบัติการตระกูลใด เช่นเครื่องที่เปิดบริการพอร์ต 139 (NetBios) โดยมากจะเป็นระบบปฏิบัติการตระกูลไมโครซอฟท์วินโดว์ส เป็นต้น

เทคนิคที่ได้กล่าวมาเป็นเทคนิคขั้นพื้นฐาน ไม่จำเป็นต้องใช้เครื่องมือพิเศษ แต่ก็มีข้อจำกัดเนื่องจากข้อมูลในแบนเนอร์หรือบริการต่างๆ สามารถถูกควบคุมได้จากผู้ดูแลระบบ ส่งผลให้บ้างครั้งข้อมูลที่ได้จากการสำรวจอาจผิดพลาด ทำให้เกิดวิธีการตรวจสอบระบบปฏิบัติการที่เรียกกว่า ทีซีพีสแต็กฟิงเกอร์ปรินซ์ (TCP Stack Fingerprint) โดยเป็นวิธีที่ใช้หลักความเป็นจริงที่ว่าชั้นทีซีพีสแต็กของระบบปฏิบัติการแต่ละระบบ จะมีการสร้างแพ็กเก็ตในชั้นทีซีพีที่แตกต่างกันในส่วนของรายละเอียดปลีกย่อยหรือมีการตอบสนองแตกต่างกันในกรณีที่ได้รับแพ็กเก็ตที่ผิดปกติ หากนำความแตกต่างของแพ็กเก็ตมาเปรียบเทียบกับข้อมูลในฐานข้อมูล จะทำให้สามารถตรวจสอบได้ว่าระบบปฏิบัติการนั้นเป็นระบบใด

การทำงานของโปรแกรมตรวจสอบระบบปฏิบัติการที่ใช้หลักการของทีซีพีสแต็กฟิงเกอร์ปรินซ์คือโปรแกรมจะสร้างข้อมูลในชั้นทีซีพีที่มีแฟล็กผิดปกติแล้วส่งไปยังเครื่องเป้าหมาย เมื่อเป้าหมายได้รับ

ข้อมูลดังกล่าวจะไม่สามารถตอบสนองด้วยรูปแบบที่กำหนดได้ เนื่องจากชั้นทีซีพีสแต็กของระบบไม่ได้ออกแบบมาสำหรับกรณีที่ได้รับแพ็กเก็ตดังกล่าว เครื่องเป้าหมายจะตอบสนองต่อแพ็กเก็ตดังกล่าวแตกต่างกันตามการโปรแกรมในชั้นทีซีพีสแต็กของแต่ละระบบปฏิบัติการ จากความแตกต่างดังกล่าวนี้เองทำให้สามารถตรวจสอบระบบปฏิบัติการว่าเป็นระบบปฏิบัติการใด ตัวอย่างของรูปแบบแพ็กเก็ตที่ผิดปกติและไม่มีทางเกิดขึ้นได้ในการสื่อสารตามปกติ และมักถูกนำมาใช้ในการตรวจสอบหาระบบปฏิบัติการมีหลายรูปแบบดังนี้

- **FIN probe** ใช้การส่งแพ็กเก็ตไปยังโพรโตคอลทีซีพี โดยกำหนดแฟล็ก FIN เป็น 1 ซึ่งมาตรฐาน อาร์เอฟซี 793 ได้กำหนดไว้ว่ารูปแบบนี้จะต้องไม่มีแพ็กเก็ตใดส่งตอบกลับไป แต่ในไดรเวอร์ของวินโดวส์เอ็นทีจะตอบสนองกลับมาด้วยแพ็กเก็ตทีซีพีที่กำหนดแฟล็ก FIN และ ACK เป็น 1
- **Bogus Flag probe** ใช้การส่งแพ็กเก็ตที่กำหนดให้ SYN เป็น 1 และกำหนดบิตที่ไม่ได้ใช้งานบางบิตให้เป็น 1 ด้วย ในระบบปฏิบัติการลีนุกซ์จะมีการตอบสนองกับแพ็กเก็ตรูปแบบนี้ด้วยการกำหนดแฟลกบางตัวในแพ็กเก็ตตอบรับ
- **Initial Sequence Number (ISN) Sampling** ใช้การตรวจสอบรูปแบบของซีเควนส์นัมเบอร์ในส่วนหัวของแพ็กเก็ตว่ามักเป็นค่าใด เนื่องจากความสัมพันธ์ของหมายเลขดังกล่าวจะแตกต่างกันจากรูปแบบการสุ่มที่แต่ละระบบปฏิบัติการเลือกใช้
- **TCP initial window size** เป็นการตรวจดูขนาดของ TCP window เพราะบางระบบปฏิบัติการจะมีการกำหนดค่านี้ไว้ตายตัว โดยมักเป็นค่าเฉพาะตัวของระบบปฏิบัติการ
- **ICMP message quoting** เป็นการตรวจดูรายละเอียดของข้อมูลจากแพ็กเก็ตไอซีเอ็มพี เนื่องจากเมื่อเกิดการผิดพลาดในการนำส่งข้อมูล จะทำให้มีการใช้โพรโตคอลไอซีเอ็มพีในการวิเคราะห์ข้อผิดพลาดเหล่านั้น ในระบบปฏิบัติการที่แตกต่างกันจะมีการให้รายละเอียดของข้อมูลในโพรโตคอลไอซีเอ็มพีที่แตกต่างกันด้วย

ตัวอย่างของเครื่องมือตรวจสอบระบบปฏิบัติการเช่น queso, nmap เป็นต้น โดยใน queso จะมีลำดับและรูปแบบแพ็กเก็ตที่ส่งไปเพื่อตรวจสอบปฏิบัติการดังนี้

1. ส่งแพ็กเก็ตไปขอเริ่มการเชื่อมต่อที่พอร์ต 80 (หรือพอร์ตใดก็ได้ที่เปิดอยู่)
2. รอการตอบรับแพ็กเก็ต SYN ACK จากเป้าหมาย เพื่อยืนยันว่าเครื่องเป้าหมายเปิดอยู่
3. ส่งแพ็กเก็ตที่เป็นไปไม่ได้ไปยังเป้าหมายดังนี้
   - ❑ SYN – ACK และ ACK Number เป็น 0
   - ❑ FIN
   - ❑ FIN – ACK และ ACK Number เป็น 0
   - ❑ PUSH
   - ❑ SYN – XXX - YYY โดย X และ Y คือค่าในฟิลด์ที่สงวนไว้ มีขนาด 6 บิต

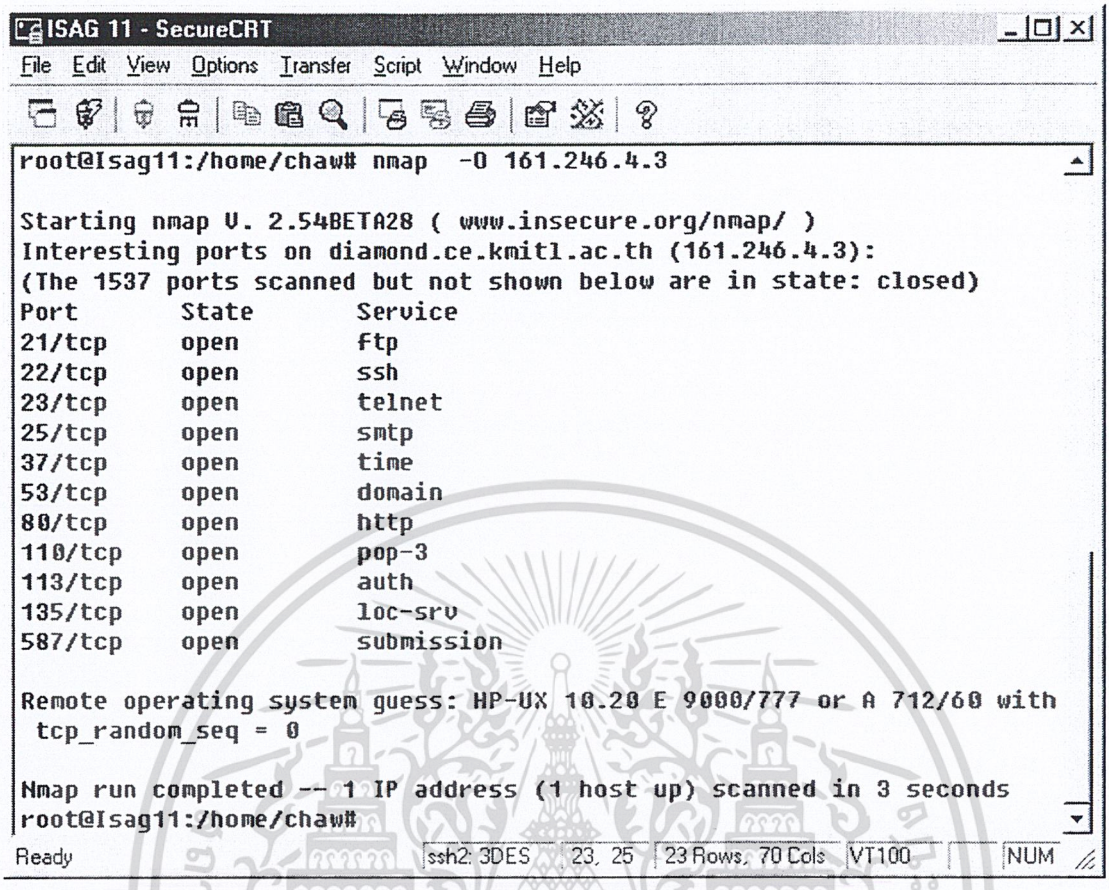
สำหรับโปรแกรม nmap (Network Mapper) ซึ่งเป็นโปรแกรมที่ได้รับการพัฒนาหลังจากโปรแกรม queso โดยได้มีการบรรจุรูปแบบการสำรวจของโปรแกรม queso บางส่วนลงในโปรแกรมด้วย สำหรับโปรแกรม nmap มีรูปแบบการสำรวจดังนี้

1. ทำพอร์ตสแกนเพื่อหารายชื่อพอร์ตที่เปิดให้บริการอยู่
2. ส่งแพ็กเก็ตไปยังเป้าหมายตามลำดับ ดังนี้
   - ❑ SYN, TCP OPTION ไปยังพอร์ตทีซีพีที่เปิดอยู่
   - ❑ NONE Flag, TCP OPTION ไปยังพอร์ตทีซีพีที่เปิดอยู่
   - ❑ SYN FIN URGENT PUSH OPTION ไปยังพอร์ตทีซีพีที่เปิดอยู่
   - ❑ ACK, TCP OPTION ไปยังพอร์ตทีซีพีที่เปิดอยู่
   - ❑ SYN, TCP OPTION ไปยังพอร์ตทีซีพีที่ปิด
   - ❑ ACK, TCP OPTION ไปยังพอร์ตทีซีพีที่ปิด
   - ❑ FIN URGENT, PUSH ไปยังพอร์ตทีซีพีที่ปิด
3. ส่งแพ็กเก็ตยูดีพีไปยังพอร์ตที่ปิด ด้วยข้อมูลจำนวนหนึ่งเพื่อดูการตอบรับจากยูดีพีสแต็ก

จะเห็นว่าโปรแกรม nmap มีขั้นตอนในการสำรวจละเอียดกว่าในโปรแกรม queso เช่นว่ามีการตรวจสอบว่าเป้าหมายเปิดให้บริการอะไรบ้างโดยใช้การสแกนพอร์ต จากนั้นจึงสำรวจเข้าไปยังพอร์ตต่างๆ เหล่านั้น ทำให้โปรแกรม nmap มีปริมาณแพ็กเก็ตที่ถูกส่งออกไปมากกว่าโปรแกรม queso ( ปกติแล้วโปรแกรม queso ส่งแพ็กเก็ตออกไปเพียง 7-10 แพ็กเก็ตเพื่อตรวจสอบระบบปฏิบัติการหนึ่งครั้ง แต่ nmap ส่งแพ็กเก็ตออกไปถึง 1000-1500 แพ็กเก็ต ขึ้นอยู่) บางครั้งการสำรวจจากโปรแกรม namp จึงใช้เวลานานกว่าโปรแกรม queso

เนื่องจากโปรแกรม nmap จะพยามยามทำการสำรวจทุกจุดของระบบที่สามารถเข้าถึงได้ เพื่อให้ได้ข้อมูลมากสุด ผลที่ได้คือสามารถ “เดา” ชื่อและเวอร์ชันของระบบปฏิบัติการของเครื่องเป้าหมายได้อย่างถูกต้อง

แต่ในความเป็นจริงแล้วในปัจจุบันได้มีการพัฒนาระบบปฏิบัติการใหม่ๆ ขึ้นมาตลอดเวลา อีกทั้งในการติดตั้งแพตช์ (Patch ) หรือฮอตฟิกซ์ (Hotfix) ต่างๆ ให้กับระบบปฏิบัติการ จะทำให้การทำงานของชั้นทีซีพีสแต็กมีการเปลี่ยนแปลงอยู่ตลอดเวลา บางครั้งอาจส่งผลให้การตรวจสอบระบบปฏิบัติการทำได้ผิดพลาด การปรับปรุงฐานข้อมูลของโปรแกรมที่ทำการตรวจสอบระบบปฏิบัติการจึงมีความสำคัญมาก โปรแกรม nmap ได้ชื่อว่าเป็นโปรแกรมที่มีการปรับปรุงฐานข้อมูลดังกล่าวอยู่ตลอดเวลา จากเหตุผลดังกล่าวจึงทำให้โปรแกรม nmap ได้คือว่าเป็นโปรแกรมตรวจสอบระบบปฏิบัติการที่ทำงานได้ถูกต้องมากที่สุดโปรแกรมหนึ่งในปัจจุบัน




*รูปที่ 3-2 โปรแกรม nmap ทำการตรวจสอบปฏิบัติการได้เป็น HP-UX*

# บทที่ 4
# การโจมตีเพื่อให้ปิดบริการสำหรับโพรโตคอลสแต็กทีซีพี/ไอพี

## 4.1 ความหมายของการโจมตีเพื่อให้ปิดบริการ

การโจมตีเพื่อให้ปิดบริการ (Denial of Services : DoS) หมายถึง การกระทำใดๆ ที่ทำให้ระบบเป้าหมายไม่สามารถให้บริการบางอย่างได้ หรือไม่สามารถให้บริการต่อไปได้อีก โดยทั่วไปโจมตีที่พอร์ตของทีซีพี/ไอพี ซึ่งเชื่อมต่อกับบริการ (Services) ที่รองรับพอร์ตนั้นๆ ดังนั้นการโจมตีพอร์ตจึงเท่ากับการโจมตีบริการของระบบนั่นเอง และอาจมีผลทำให้ระบบนั้นไม่สามารถให้บริการบางอย่างได้ หรือไม่สามารถให้บริการใดๆ ได้เลย

## 4.2 ประเภทของการโจมตีเพื่อให้ปิดบริการ

ในที่นี้ประเภทของการโจมตีสามารถแบ่งได้ดังต่อไปนี้

### 4.2.1 ประเภทอยู่ในชั้นทรานสปอร์ต หรือชั้นอินเทอร์เน็ต

การโจมตีในระดับชั้นนี้สามารถแบ่งได้เป็น 2 แบบหลักๆ ได้แก่

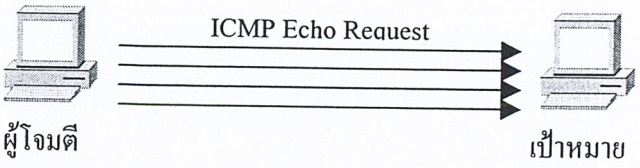
#### 4.2.1.1 การส่งแพ็กเก็ตจำนวนมาก (Amount of Packets Sending)

การโจมตีแบบนี้เป็นการส่งแพ็กเก็ตปริมาณมากเข้าไปยังระบบเป้าหมาย อาจทำให้ระบบเป้าหมายไม่สามารถให้บริการบางอย่าง หรือไม่สามารถทำงานต่อไปได้ ซึ่งแพ็กเก็ตที่ส่งออกไปนี้สามารถแบ่งออกได้เป็น

(1) แพ็กเก็ตข้อมูล (Data Packets)

การโจมตีวิธีนี้ทำได้โดยการส่งแพ็กเก็ตข้อมูลปริมาณมาก เมื่อข้อมูลเข้ามาสู่เครื่องเป้าหมายก็เก็บไว้ในบัฟเฟอร์ก่อนนำมาประมวลผลอีกครั้ง ดังนั้นหากส่งแพ็กเก็ตเข้ามาเป็นปริมาณมาก อาจทำให้บัฟเฟอร์ของเครื่องเป้าหมายไม่เพียงพอที่จะสามารถรองรับแพ็กเก็ตเหล่านั้นได้ทั้งหมด ซึ่งอาจทำให้เครื่องเป้าหมายให้บริการได้ช้าลง หรือต้องหยุดการให้บริการไปเลย ตัวอย่างการโจมตีประเภทนี้เช่น Ping Flood Attack เป็นต้น

Ping Flood เป็นการโจมตีในยุคแรกๆ ของ DoS หลักการคือส่ง ICMP Echo Request (รูปแบบเดียวกับคำสั่ง Ping) ไปยังเป้าหมายมากๆ ในระยะเวลาติดต่อกัน ทำให้เป้าหมายต้องคอยตอบ ICMP Echo Reply ตลอดเวลาจนไม่สามารถให้บริการอย่างอื่นได้ ความรุนแรงของการโจมตีขึ้นอยู่กับปริมาณแพ็กเก็ตที่โจมตีไปยังเครื่องเป้าหมาย หากเครื่องที่ทำการโจมตีมีประสิทธิภาพสูงและเครือข่ายมีแบนด์วิดธ์มาก อาจส่งผลทำให้เครื่องเป้าหมายหยุดการทำงานลงได้




*รูปที่ 4-1 การโจมตีด้วย Ping Flood Attack*

ข้อสังเกตสำหรับการโจมตีประเภทนี้คือ จะปรากฏแพ็กเก็ต ICMP Echo Request และ ICMP Echo Reply ปริมาณมหาศาล โดยมีการรับส่งกันระหว่างเครื่องเป้าหมายที่ถูกโจมตีกับเครื่องอื่นๆ ที่อาจมีหรือไม่มีตัวตนในอินเตอร์เน็ตก็ได้ เนื่องจากกระบวนการสำคัญอย่างหนึ่งของการโจมตีลักษณะนี้คือ ผู้โจมตีต้องปลอมหมายเลขไอพี (IP Spoofing) เสมอ เพื่อป้องกันไม่ให้แพ็กเก็ต ICMP Echo Reply ถูกส่งกลับมายังเครื่องตัวเอง ซึ่งจะทำให้ผู้โจมตีได้รับผลกระทบจากการโจมตีด้วย และการปลอมไอพียังเป็นหลักประกันได้ว่าจะไม่สามารถติดตามได้ว่าผู้ใดเป็นผู้โจมตี รูปแบบแพ็กเก็ตที่เกิดขึ้นจากการโจมตีมีลักษณะดังนี้

**14:49:43.217137 62.51.12.23 > 10.1.1.10 : icmp: echo request**
14:49:43.217175 10.1.1.10 > 62.51.12.23 : icmp: echo reply
**14:49:43.217195 62.51.12.23 > 10.1.1.10 : icmp: echo request**
14:49:43.217219 10.1.1.10 > 62.51.12.23 : icmp: echo reply
**14:49:43.217245 96.141.106.124 > 10.1.1.10 : icmp: echo request**
14:49:43.217279 10.1.1.10 > 96.141.10.124 : icmp: echo reply
14:49:43.219017 172.19.251.18 > 10.1.1.10 : icmp: net 162.75.127.79 unreachable
**14:49:43.237136 75.126.62.65 > 10.1.1.10 : icmp: echo request**
14:49:43.237169 10.1.1.10 > 75.126.62.65 : icmp: echo reply
**14:49:43.237193 75.126.62.65 > 10.1.1.10 : icmp: echo request**
14:49:43.237216 10.1.1.10 > 75.126.62.65 : icmp: echo reply
**14:49:43.237240 218.155.179.58 > 10.1.1.10 : icmp: echo request**
14:49:43.237272 10.1.1.10 > 218.155.17.58 : icmp: echo reply

*รูปที่ 4-2 รูปแบบแพ็กเก็ตที่เกิดขึ้นจากการโจมตีจาก Ping Flood Attack*

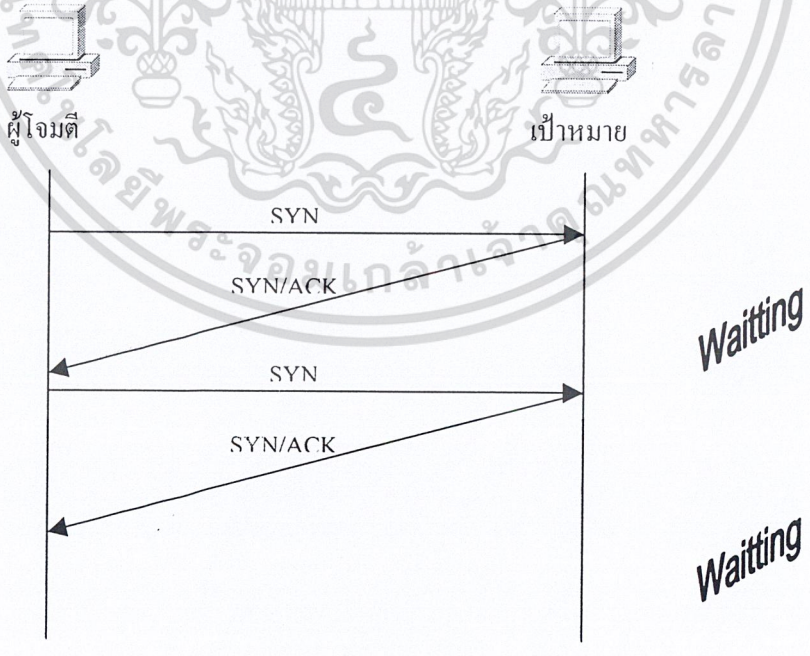
นอกจาก Ping floodign Attack จะสร้างความเสียหายแก่เครื่องเป้าหมายแล้วยังสร้างความเสียหายแก่ระบบเครือข่ายของเครื่องเป้าหมายด้วย เพราะการโจมตีวิธีนี้จะสร้างแพ็กเก็ตเป็นจำนวนมากขึ้นในเครือข่ายที่เครื่องเป้าหมายตั้งอยู่ ทำให้ระบบเครือข่ายเกิดความคับคั่งของข้อมูล(Congestion ) อาจส่งผลให้เครือข่ายเป็นอัมพาตได้

การป้องกันการการโจมตีลักษณะนี้ทำได้โดยการกำหนดที่อุปกรณ์เราเตอร์หรือไฟร์วอลล์ โดยกำหนดไม่ให้แพ็กเก็ต ICMP Echo เข้ามายังเซิร์ฟเวอร์ แต่อย่างไรก็ตามแพ็กเก็ต ICMP Echo ได้ถูกใช้ในโปรแกรม Ping หากมีการปิดกั้นแพ็กเก็ต ICMP Echo จะทำให้ไม่สามารถตรวจสอบสถานะของเซิร์ฟเวอร์ได้ แนวทางที่ควรปฏิบัติคือกำหนดแบนด์วิดธ์ของแพ็กเก็ต ICMP Echo ให้เหมาะสมในอุปกรณ์เราเตอร์ โดยให้เพียงพอต่อการใช้งานสำหรับการตรวจสอบสถานะของระบบ แต่ไม่มากจนทำให้เกิดการโจมตีได้

(2) แพ็กเก็ตสำหรับการควบคุม (Control Packets)

นอกจากแพ็กเก็ตที่เป็นตัวข้อมูลแล้ว ยังมีแพ็กเก็ตอีกรูปแบบหนึ่งที่สำคัญมากสำหรับการติดต่อสื่อสารบนโพรโตคอลทีซีพี/ไอพี นั่นคือแพ็กเก็ตส่วนการควบคุม ตัวอย่างแพ็กเก็ตประเภทนี้คือสัญญาณ SYN หรือ ACK สำหรับการสถาปนาการเชื่อมต่อ หรือสัญญาณ FIN สำหรับการยกเลิกการเชื่อมต่อเป็นต้น

ตัวอย่างของการโจมตีแบบนี้ ได้แก่ การทำ SYN flood เนื่องจากปกติการเชื่อมต่อแบบ 3-way handshake เป็นไปตามลักษณะที่ได้อธิบายในหัวข้อ 2.4 แต่ในการโจมตีลักษณะนี้ใช้วิธีทำให้การทำ 3-way handshake ไม่สมบูรณ์ กล่าวคือ เครื่องที่ขอบริการส่งสัญญาณ SYN ไป แต่เมื่อได้รับสัญญาณ ACK จากเครื่องที่ให้บริการแล้ว ไม่ส่งสัญญาณ ACK ตอบกลับไป ทำให้เครื่องที่ให้บริการต้องเปิดการเชื่อมต่อรอการตอบกลับ ดังรูปที่ 4-1 ซึ่งการเปิดการเชื่อมต่อรอเอาไว้นี้ต้องใช้ทรัพยากรของระบบส่วนหนึ่ง และหากมีการส่งสัญญาณในลักษณะนี้มากๆ และทรัพยากรของระบบมีไม่เพียงพอ อาจทำให้ระบบไม่สามารถให้บริการอย่างอื่น หรือให้บริการกับผู้ร้องขอรายอื่นได้




*รูปที่ 4-3 แสดงการส่งแพ็กเก็ตแบบ SYN Flood*

สำหรับการโจมตีแบบ SYN Flood จะทำให้เกิดแพ็กเก็ตในระบบเครือข่ายในลักษณะดังรูป

10:09:43.137 10.0.0.1 > **isag11.ce.kmitl.ac.th.80: S** 399259715:399259715(0)

10:09:43.139 10.0.0.2 > **isag11.ce.kmitl.ac.th.80: S** 399259715:399259715(0)

10:09:43.143 10.0.0.3 > **isag11.ce.kmitl.ac.th.80: S** 399259715:399259715(0)

10:09:43.149 10.0.0.4 > **isag11.ce.kmitl.ac.th.80: S** 399259715:399259715(0)

10:09:43.158 10.0.0.5 > **isag11.ce.kmitl.ac.th.80: S** 399259715:399259715(0)

10:09:43.165 10.0.0.6 > **isag11.ce.kmitl.ac.th.80: S** 399259715:399259715(0)

10:09:43.177 10.0.0.7 > **isag11.ce.kmitl.ac.th.80: S** 399259715:399259715(0)

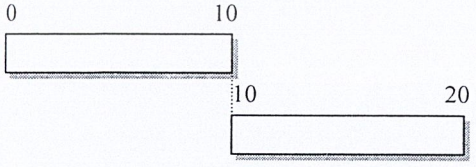
10:09:43.185 10.0.0.8 > **isag11.ce.kmitl.ac.th.80: S** 399259715:399259715(0)

*รูปที่ 4-4 แพ็กเก็ตที่เกิดขึ้นจากการโจมตีแบบ SYN flood*

ในปัจจุบันนี้การโจมตีแบบ SYN Flood ถือได้ว่าเป็นการโจมตีที่ได้ผลและหาทางป้องกันได้ยาก เนื่องจากยากที่จะแยกลักษณะของแพ็กเก็ตที่ใช้ในการโจมตีกับแพ็กเก็ตที่ขอเริ่มต้นเชื่อมต่อทั่วไป นอกจากนี้ไฟร์วอลล์หรือเราเตอร์ทั่วไปยังไม่สามารถป้องกันการโจมตีประเภทนี้ได้อย่างสมบูรณ์ หนทางที่เป็นไปได้คือการใช้ระบบตรวจจับผู้บุกรุกทางระบบเครือข่ายทำการตรวจจับการโจมตี เพื่อนำข้อมูลจากการโจมตีกลับไปตั้งค่าอุปกรณ์เราเตอร์หรือไฟร์วอลล์เพื่อป้องกันการโจมตีมายังเซิร์ฟเวอร์

#### 4.2.1.2 ความผิดปกติของแฟร็กเมนต์ (Abnormal Fragmentation)

การโจมตีวิธีนี้อาศัยหลักการแฟร็กเมนเตชันและรีแอสเซมเบิลที่กล่าวไว้ข้างต้น โดยทำให้แพ็กเก็ตนั้นต้องมีการรีแอสเซมเบิล (กำหนดค่า MF flag = 0) ซึ่งปกติการรีแอสเซมเบิลแพ็กเก็ตทั้งหมดต้องสามารถเชื่อมต่อกันได้สนิท ดังรูปที่ 4-2 แต่แพ็กเกจที่ผู้บุกรุกส่งไปมีการแก้ไขข้อมูลในบางฟิลด์ ทำให้เกิดความผิดปกติในกระบวนการรีแอสเซมเบิล ซึ่งการโจมตีในลักษณะนี้ แบ่งได้ดังต่อไปนี้

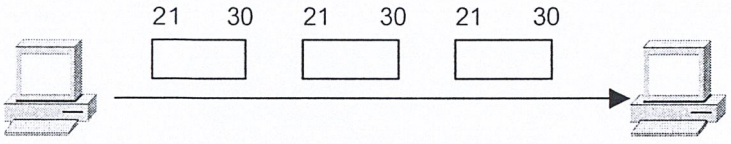



*รูปที่ 4-5 แสดงการรีแอสเซมบลีแบบปกติ*

(1) การส่งแพ็กเก็ตที่มีลำดับผิดปกติ (Abnormal Sequences of Packets Sending)

ปกติการส่งแพ็กเก็ตมักเรียงตามลำดับกันไป หากไม่เรียงลำดับก็ต้องรอจนกว่าแพ็กเก็ตก่อนหน้านี้มาถึง เพื่อเรียงลำดับแพ็กเก็ตที่เครื่องรับ แต่การโจมตีแบบนี้กลับส่งเฉพาะแพ็กเก็ตสุดท้าย เพื่อให้

ระบบเป้าหมายรอแพ็กเก็ตก่อนหน้า และส่งไปเป็นปริมาณมากๆ ซึ่งจะส่งผลให้ระบบต้องจองทรัพยากรส่วนหนึ่งเพื่อรองรับแพ็กเก็ตที่ต้องรอเหล่านั้นในปริมาณมาก จนกระทั่งระบบไม่สามารถจัดหาทรัพยากรได้เพียงพอ ส่งผลให้ระบบไม่สามารถให้บริการอย่างอื่นได้




*รูปที่ 4-6 แสดงการส่งเฉพาะแพ็กเก็ตสุดท้ายไปยังเป้าหมาย*

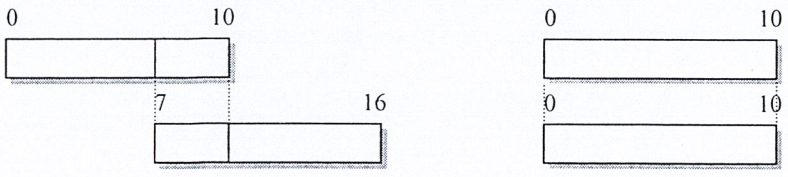
โดยปกติแล้วการโจมตีในรูปแบบนี้ผู้โจมตีจะแก้ไขข้อมูลในฟิลด์แสดงลำดับของแพ็กเก็ต (Fragment Offset) ของแพ็กเก็ตไอพี ซึ่งเป็นส่วนที่แสดงลำดับของข้อมูลหลังจากกระบวนการแฟร็กเมนต์เตชัน โดยแก้ให้ส่งแพ็กเก็ตสุดท้ายหรือแพ็กเก็ตหลังๆ เพียงแพ็กเก็ตเดียวเลย ทำให้ระบบเป้าหมายต้องรอแพ็กเก็ตก่อนหน้านี้

(2) การส่งแพ็กเก็ตที่มีขนาดเหลื่อมกัน (Overlapped Packets' Size Sending)

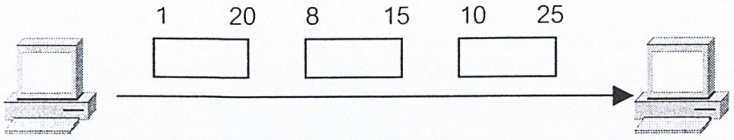
ปกติแพ็กเก็ตที่ส่งมาต้องนำมาต่อกันที่ระบบเป้าหมายได้พอดี แต่การโจมตีแบบนี้เป็นการส่งแพ็กเก็ตที่มีขนาดเหลื่อมกัน หรือซ้อนทับกัน ทำให้ข้อมูลเมื่อมาต่อกันแล้วเกิดความผิดพลาด หรือไม่สามารถเชื่อมต่อกันได้

โดยปกติแล้วการโจมตีแบบนี้ ผู้บุกรุกสามารถแก้ไขข้อมูลได้ 2 แห่งใหญ่ๆ ได้แก่

- การแก้ไขข้อมูลที่ฟิลด์แสดงลำดับของแพ็กเก็ต (Fragment Offset) ของแพ็กเก็ตไอพีหลังจากกระบวนการรีแอสเซมเบิล ซึ่งทำให้ลำดับในการส่งมีความผิดพลาด และอาจเกิดการเหลื่อมล้ำของแพ็กเก็ต กระบวนการรีแอสเซมเบิลอาจเกิดปัญหาได้
- การแก้ไขฟิลด์แสดงความยาวของ (Total Length) ของแพ็กเก็ตไอพี หลังจากกระบวนการรีแอสเซมเบิล ขนาดของแพ็กเก็ตที่มาต่อไม่พอดีกัน ทำให้ไม่สามารถรวมแพ็กเก็ตได้ หรือหากรวมได้ ข้อมูลที่ได้ก็ไม่ถูกต้อง



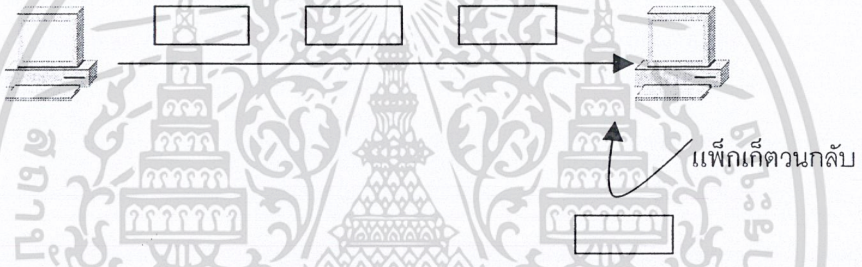

*รูปที่ 4-7 แสดงการรีแอสเซมบลีแบบแพ็กเก็ตมีขนาดเหลื่อมกัน*




*รูปที่ 4-8 แสดงการโจมตีโดยส่งแพ็กเก็ตที่ไม่สามารถรีแอสเซมเบิลได้*

**4.2.1.3 การส่งแพ็กเก็ตแบบวนลูป (Looping)**

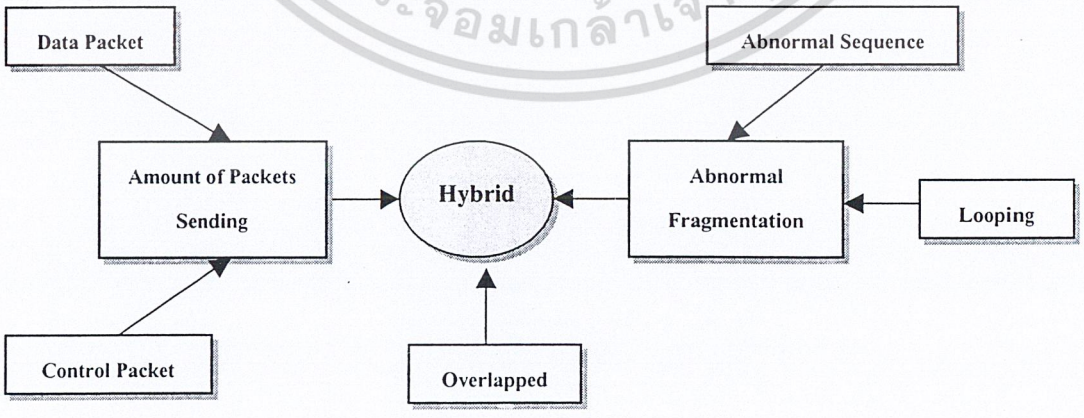
เป็นการโจมตีโดยการส่งโดยกำหนดค่าแอดเดรสต้นทาง (Source Address) และแอดเดรสปลายทาง (Destination Address) ให้เหมือนกันทำให้เกิดการรับส่งวนไปวนมาอยู่ที่เครื่องเป้าหมายเอง เช่น LAND ซึ่งเป็นโปรแกรมโจมตีที่มีการกำหนดแอดเดรสต้นทาง และแอดเดรสปลายทางเป็นค่าเดียวกัน คือ เป็นแอดเดรสของเครื่องเป้าหมายนั่นเอง ทำให้เกิดการส่งวนไปวนมาอยู่ที่เครื่องเป้าหมาย




*รูปที่ 4-9 แสดงการโจมตีโดยส่งแพ็กเก็ตแบบวนลูป*

**4.2.1.4 แบบผสม (Hybrid)**

คือ การโจมตีที่อาศัยวิธีการผสมกันระหว่างสามแบบแรกที่ได้กล่าวมาแล้ว ดังรูปที่ 4-5




*รูปที่ 4-10 แสดงแผนภูมิแสดงประเภทของการโจมตีเพื่อให้ปิดบริการสำหรับสแต็กทีซีพี/ไอพี*

**ประเภทอยู่ในชั้นแอพพลิเคชัน**

การโจมตีประเภทอื่นนอกจากที่ได้กล่าวมาแล้วข้างต้น ส่วนใหญ่เกิดจากการใช้จุดอ่อนหรือข้อผิดพลาดของแอพพลิเคชันที่เครื่องเป้าหมายใช้อยู่ในการโจมตีเครื่องเป้าหมายเอง ไม่ว่าจะเป็นจุดอ่อนของระบบปฏิบัติการ หรือข้อผิดพลาดของซอฟต์แวร์ก็ตาม

ในกรณีเช่นนี้เจ้าของเครื่องหรือผู้ดูแลระบบสามารถแก้ไขได้เอง โดยการนำโปรแกรมแพ็ตช (Patch), ฮอตฟิกซ์ (Hotfix) หรือเซอร์วิสแพ็ค (Service Pack) มาติดตั้งเพื่อแก้ข้อผิดพลาดเหล่านี้ หรือหลีกเลี่ยงไปใช้โปรแกรมอื่นที่ไม่เกิดปัญหา ซึ่งการโจมตีในลักษณะนี้ไม่อยู่ในขอบเขตที่ศึกษา

# บทที่ 5
# ระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

## 5.1 ความหมายของการตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

ระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ (Network Intrusion Detection System หรือ NIDS) เป็นแขนงหนึ่งของระบบตรวจจับผู้บุกรุก (Intrusion Detection System หรือ IDS) โดยเน้นไปทางการตรวจจับทางเครือข่ายคอมพิวเตอร์เป็นหลัก

โดยระบบนี้ต้องเก็บข้อมูลของแพ็กเก็ตต่างๆ ที่เข้ามาสู่ระบบ แล้วนำมาวิเคราะห์เปรียบเทียบกับกฎต่างๆ ที่ตั้งไว้ รวมถึงนโยบายขององค์กรก็นำมาพิจารณาด้วย เพื่อตรวจสอบว่ามีสิ่งผิดปกติเกิดขึ้นกับระบบหรือไม่ หากเกิดสิ่งผิดปกติ ก็แจ้งเตือนไปยังผู้ดูแลระบบ หรือเก็บไว้ในล็อกไฟล์ต่อไป

การตรวจจับผู้บุกรุกทางคอมพิวเตอร์สามารถแบ่งตามลักษณะของการโจมตีได้ 5 ประเภท ได้แก่

(1) การพยายามเจาะเข้าไปทำลายเครือข่าย (Attempted break-ins)

(2) การปลอมแปลงเพื่อเข้ามาโจมตีเครือข่าย (Masquerade attacks)

(3) การอาศัยจุดบกพร่องของระบบรักษาความปลอดภัยเพื่อเจาะเข้าสู่เครือข่าย (Penetration of the security control system)

(4) การโจมตีเพื่อให้ปิดบริการ (Denial of service)

(5) การสำรวจระบบ (System survey)

## 5.2 ขอบเขตของระบบต้นแบบการตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ที่สร้างขึ้น

ระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ที่สร้างขึ้น มุ่งเน้นการศึกษา ออกแบบ และพัฒนาระบบการตรวจจับการสำรวจระบบและการโจมตีเพื่อให้ปิดบริการ โดยเป็นหนึ่งในประเภทของการตรวจจับผู้บุกรุกตามที่ได้กล่าวมาแล้ว ซึ่งมีแนวโน้มเพิ่มขึ้นทุกวัน

## 5.3 วิธีการตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

ระบบตรวจจับผู้บุกรุกที่สร้างขึ้นนี้มีวิธีการตรวจจับผู้บุกรุกที่สามารถแบ่งออกตามประเภทของการบุกรุกเป็น 2 กรณี ได้แก่

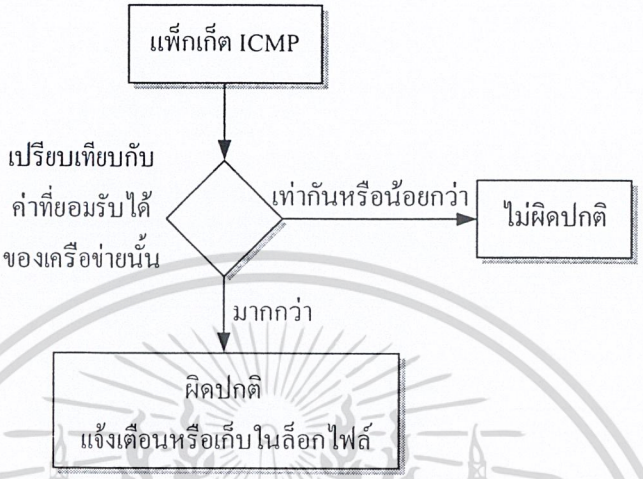
### 5.3.1 การบุกรุกเพื่อสำรวจระบบ

การบุกรุกเพื่อทำการสำรวจระบบเป็นการกระทำเพื่อเก็บข้อมูลของระบบ เพื่อใช้ในการโจมตี โดยข้อมูลที่ผู้โจมตีมักต้องการได้แก่ หมายเลขไอพีหรือชื่อเครื่อง, โครงสร้างทางเครือข่ายของระบบเป้าหมาย, ชื่อของบริการที่เปิด และระบบปฏิบัติการรวมทั้งเวอร์ชั่นที่ติดตั้งบนเครื่องเป้าหมาย

สำหรับการสำรวจระบบที่ระบบตรวจจับสามารถทำการตรวจจับได้ มีอยู่ 3 วิธีการสำรวจคือ

### 5.3.1.1 ปิงสวีป

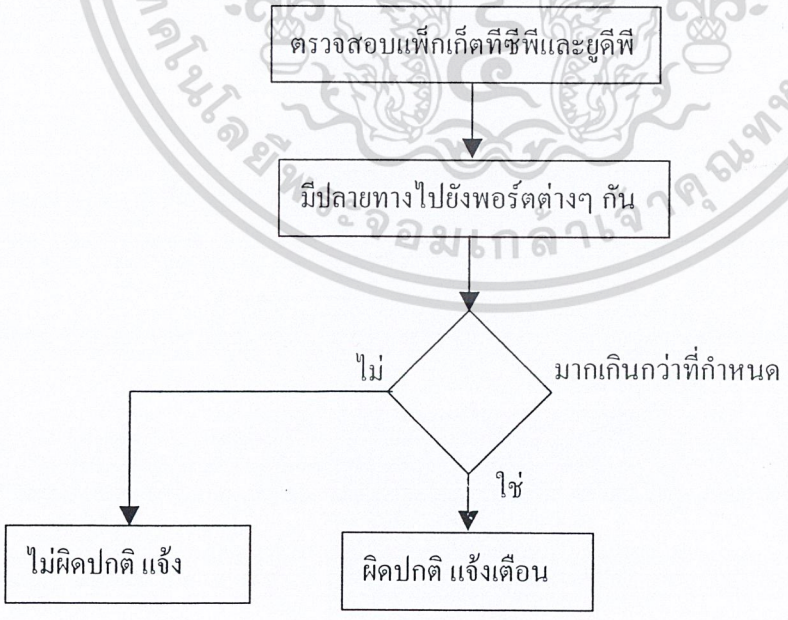
การตรวจจับการปิงสวีสามารถทำได้โดยการตรวจสอบดูแพ็กเก็ต ICMP Request (Type 8) ที่เข้ามายังระบบ หากมีแพ็กเก็ตลักษณะนี้จำนวนมากและมีปลายทางแตกต่างกัน จะสามารถสรุปได้ว่าในเครือข่ายกำลังถูกสำรวจโดยการปิงสวีป




*รูปที่ 5-1 แสดงการตรวจสอบการปิงสวีป*

### 5.3.1.2 การสแกนพอร์ต

การตรวจสอบการสแกนพอร์ตทำได้โดยการตรวจดูแพ็กเก็ตที่มีหมายเลขพอร์ตปลายทางในลักษณะกระจาย คือแพ็กเก็ตมีการส่งไปยังเครื่องๆ เดียวแต่มีการส่งไปยังพอร์ตต่างๆ กันเป็นจำนวนมาก

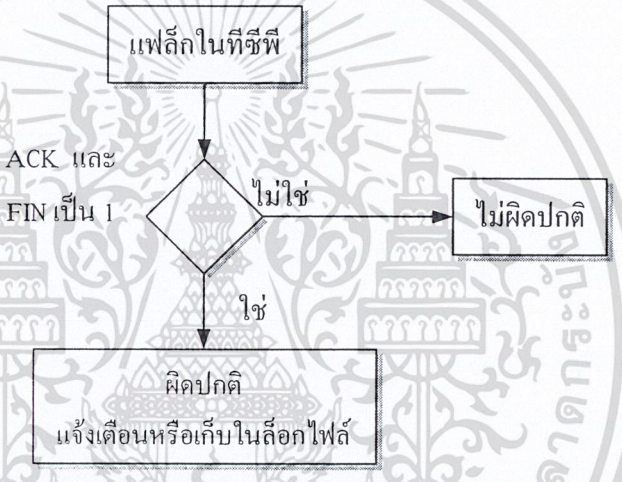



*รูปที่ 5-2 แสดงการตรวจสอบการปิงสวีป*

**5.3.1.3 การตรวจสอบระบบปฏิบัติการ**

การตรวจจับการตรวจสอบปฏิบัติการสามารถตรวจสอบได้จากการพิจารณาแฟล็กที่ถูกส่งไปในชั้นทีซีพีของทุกๆ พอร์ตว่าเป็นแพ็กเก็ตแบบผิดปกติหรือไม่ กล่าวคือในแต่ละสเตทของทีซีพีโพรโตคอลนั้นจะมีรูปแบบแฟล็กตายตัวอยู่ ตามสถานะปัจจุบันของสเตทของทีซีพี เช่นหากต้องการเริ่มต้นการเชื่อมต่อโพรโตคอลทีซีพี จะต้องกำหนดให้แฟล็ก ACK เป็น 1 ส่วนแฟล็กอื่นต้องเป็น 0 หรือหากต้องการยกเลิกการเชื่อมต่อโพรโตคอลทีซีพี จะต้องกำหนดให้แฟล็ก FIN เป็น 1 ส่วนแฟล็กอื่นเป็น 0 เป็นต้น

วิธีการตรวจจับที่ใช้ในโปรแกรมนั้นได้ทำการตรวจจับโดยหากแพ็กเก็ตมีแฟล็ก ACK และ FIN เป็น 1 พร้อมกันในแพ็กเก็ตตัวเดียวกัน จะระบุว่าเป็นการบุกรุกโดยการสำรวจเพื่อระบุระบบปฏิบัติการ เนื่องจากการตรวจจับวิธีนี้เป็นกรณีมาตรฐานที่โปรแกรมที่ทำการระบุระบบปฏิบัติการทุกโปรแกรมจะนำมาตรวจสอบ และเป็นแฟล็กที่ไม่สามารถเกิดได้จริงเมื่อมีการใช้งานโพรโตคอลทีซีพี



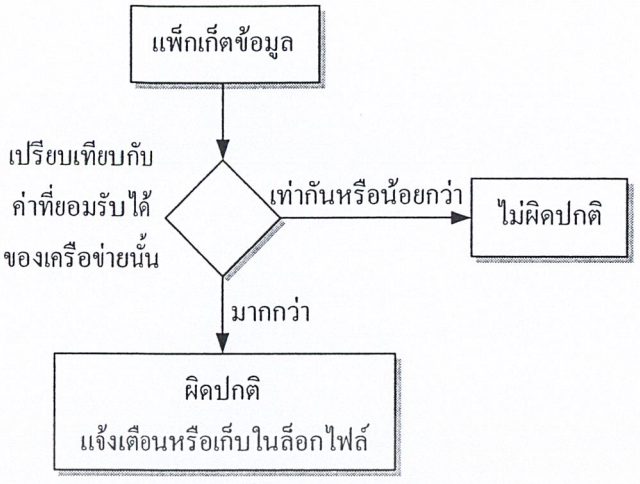

*รูปที่ 5-3 แสดงการตรวจสอบการระบุระบบปฏิบัติการ*

**5.3.2 การโจมตีเพื่อให้ปิดบริการ**

คือการกระทำใดๆ ที่ทำให้ระบบเป้าหมายไม่สามารถให้บริการบางอย่างได้ หรือไม่สามารถให้บริการต่อไปได้อีกการโจมตีระบบเครือข่ายคือการสร้างภาระให้กับเครือข่าย แบ่งได้ 3 ประเภทดังนี้

**5.3.2.1 การส่งแพ็กเก็ตปริมาณมาก**

การตรวจจับแพ็กเก็ตที่เข้ามาในลักษณะนี้ทำได้โดยใช้การนับจำนวนแพ็กเก็ตที่เข้ามาสู่ระบบโดยพิจารณาจากแอดเดรสปลายทาง (Destination Address) ในแพ็กเก็ตเฮดเดอร์ของไอพี หากเป็นค่าเดียวกันให้นับจำนวนแพ็กเก็ตที่เข้ามาในช่วงเวลาหนึ่ง แล้วนำค่าที่ได้มาเปรียบเทียบกับค่าที่ยอมรับได้ หากค่าที่นับได้มากกว่าค่าที่ยอมรับได้ ก็ให้แจ้งเตือนแก่ผู้ดูแลระบบ หรือเก็บไว้ในล็อกไฟล์ ซึ่งการทำงานดังกล่าวมานี้ เป็นไปดังรูปที่ 5-4




*รูปที่ 5-4 แสดงการตรวจสอบการส่งแพ็กเก็ตปริมาณมาก*

ความยากของการวิเคราะห์แบบนี้อยู่ที่การหาค่าที่ระบบยอมรับได้ เพราะขึ้นอยู่กับปัจจัยหลายประการ เช่น ความเร็วของเครือข่าย ความเร็วของหน่วยประมวลผลเครื่อง ปริมาณหน่วยความจำในเครื่อง เป็นต้น

การหาค่าที่ระบบยอมรับได้นี้ สามารถทำได้โดยการเปิดการเชื่อมต่อกับระบบที่วิเคราะห์ จากนั้นหาจำนวนแพ็กเก็ตที่เข้ามาในระบบในลักษณะการใช้งานปกติของแต่ละช่วงเวลา จากนั้นนำค่าสูงสุดที่ได้มาเป็นค่าที่ระบบยอมรับได้ ค่าที่ผ่านการวิเคราะห์และยอมรับได้โดยปกติมีค่าประมาณประมาณ 20,000 – 30,000 แพ็กเก็ตต่อนาที

**5.3.2.2 ความผิดปกติของแฟร็กเมนต์**

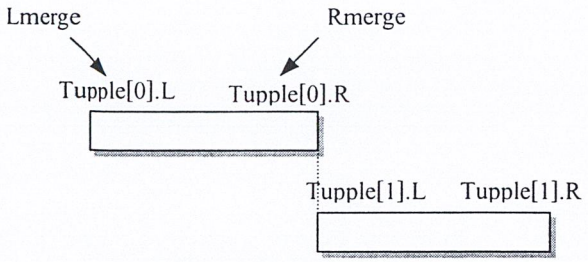
การตรวจสอบความผิดปกติของแฟร็กเมนต์มีขั้นตอนค่อนข้างซับซ้อน ซึ่งแยกอธิบายตามประเภทของความผิดปกติได้ ดังต่อไปนี้

(1) การส่งแพ็กเก็ตที่มีลำดับผิดปกติ และแพ็กเก็ตที่มีขนาดเหลื่อมล้ำกัน

การวิเคราะห์ความผิดปกติของแพ็กเก็ตในลักษณะนี้ ต้องวิเคราะห์หลังกระบวนการรีแอสเซมเบิ้ลไปแล้ว ดังนั้นจึงนำบัฟเฟอร์เข้ามาช่วยในการเก็บข้อมูล เพื่อนำมาวิเคราะห์ ดังนี้

- Fragment Buffer

คือ บัฟเฟอร์ที่เก็บข้อมูลในการวิเคราะห์ ซึ่งเก็บข้อมูลของแพ็กเก็ตไอพี และข้อมูลที่จำเป็นอื่นๆ ไว้ ได้แก่ หมายเลขไอพีของผู้ส่ง (IP_Src), หมายเลขไอพีของผู้รับ (IP_Dst), Identification, Protocol, Sec, PointArray, Array_Fragment การเก็บข้อมูลดังกล่าวจะเก็บในลักษณะของโครงสร้างข้อมูลแบบลิงส์ลิสต์




*รูปที่ 5-5 แสดงการเก็บข้อมูลของตัวแปร tupple*

การเก็บข้อมูลใน Fragment Buffer มีตัวแปรต่างๆ ที่จัดเก็บดังตารางที่ 5-1

| IP_Src | IP_Dst | Identification | Protocal | Sec | PointArray | Array_Fragment |
|---|---|---|---|---|---|---|
| | | | | | | |

*ตารางที่ 5-1 แสดงโครงสร้างการเก็บข้อมูลของ Fragment Buffer*

การเก็บข้อมูลส่วน Fragment โดยจะเก็บเป็น Array มีข้อมูลตามตารางที่ 5-2

| Flag_U | Flag_D | Flag_M | Offset | Size_Data |
|---|---|---|---|---|
| | | | | |

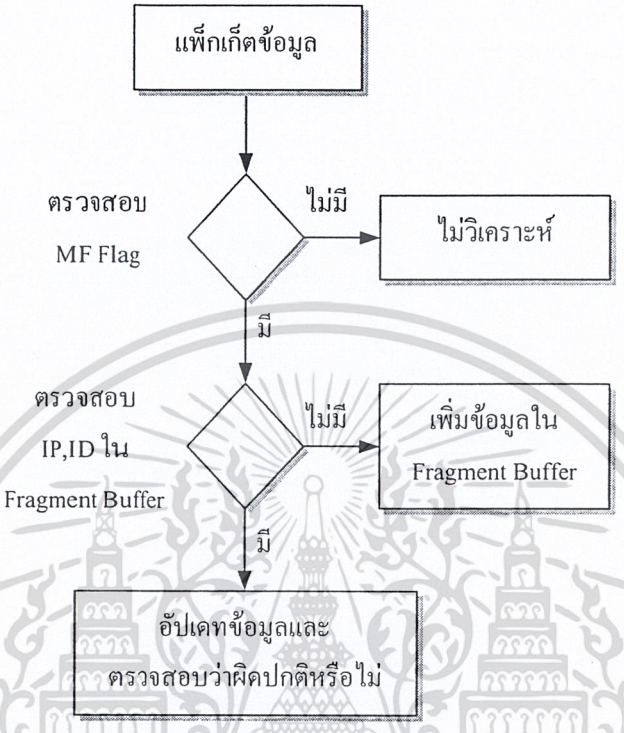
*ตารางที่ 5-2 แสดงโครงสร้างการเก็บข้อมูลของ Fragment*

- Overlap Buffer
  เป็นบัฟเฟอร์ที่เก็บข้อมูลเมื่อตรวจพบว่าเกิดการเหลื่อมล้ำของแพ็กเก็ต มีการเก็บในลักษณะของลิงส์ลิสต์

- Gap Frame Buffer
  คือ บัฟเฟอร์ที่เก็บข้อมูล เมื่อตรวจพบว่าเกิดการประกอบเฟรมไม่ได้ในลักษณะมีช่องว่างระหว่างแพ็กเก็ต มีการเก็บในลักษณะของลิงส์ลิสต์

ในการวิเคราะห์ใช้บัฟเฟอร์นี้ร่วมกัน โดยเก็บข้อมูลแพ็กเก็ตที่เข้ามาทั้งหมดลงใน Fragment Buffer และหากแพ็กเก็ตที่ส่งมาสามารถรวมกันได้ก็รวมกันเป็นแพ็กเก็ตเดียวที่ต่อเนื่องกัน โดยดูจากขอบซ้ายและขอบขวา

แต่หากรวมกันแล้วเกิดความผิดปกติ ให้แจ้งมายัง Overlap Buffer หรือ Gap Frame Buffer แล้วแต่ความผิดปกติที่เกิดขึ้น

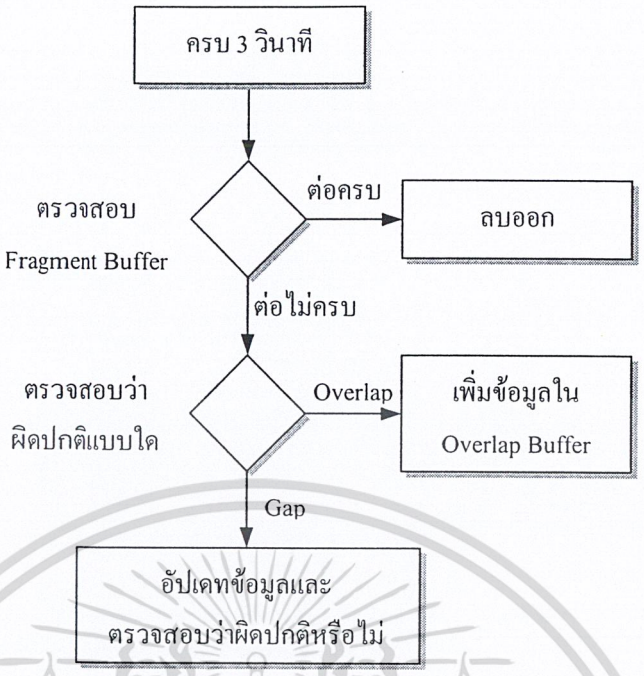



***รูปที่ 5-6 แสดงการเก็บข้อมูลลง Fragment Buffer***

หากไม่มีความผิดปกติขึ้น เมื่อครบ 3 วินาที โปรแกรมตรวจสอบจาก Fragment Buffer ว่าหากมีแพ็กเก็ตใดยังไม่ได้ประกอบ หรือประกอบไม่ครบ ให้เก็บไว้ใน Overlap Buffer หรือ Gap Frame Buffer เช่นเดียวกัน

เมื่อครบ 3 วินาที ข้อมูลใน Overlap Buffer และ Gap Buffer นี้ จะออกมาที่หน้าจอ เพื่อแจ้งให้ผู้ดูแลระบบทราบ หรือเก็บไว้ในล็อกไฟล์ เพื่อบันทึกความผิดปกติที่เกิดขึ้นไว้

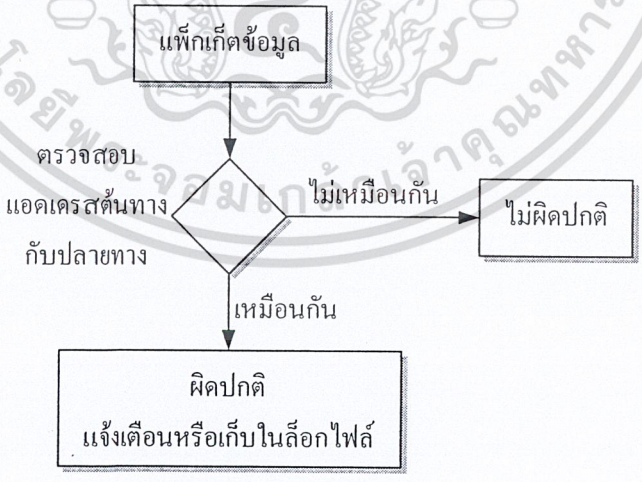
หากไม่มีความผิดปกติใดๆ เกิดขึ้นเลย และแพ็กเก็ตเหล่านั้นสามารถประกอบเป็นเฟรมได้อย่างถูกต้อง ให้ลบเฟรมเหล่านั้นออกจากบัฟเฟอร์ทันที เพื่อไม่ให้สิ้นเปลืองเนื้อที่ในการจัดเก็บ




*รูปที่ 5-7 แสดงการตรวจสอบความผิดปกติในการทำแฟร็กเมนเตชัน*

(2) การส่งแพ็กเก็ตแบบวนลูป

สามารถทำได้โดยการเปรียบเทียบค่าแอดเดรสต้นทาง และแอดเดรสปลายทางของแพ็กเก็ตไอพี หากเป็นค่าเดียวกันแสดงว่ามีความผิดปกติเกิดขึ้น เพราะทำให้เกิดการส่งในลักษณะวนลูป ซึ่งขั้นตอนการตรวจสอบเป็นไปตามรูปที่ 5-8




*รูปที่ 5-8 แสดงการตรวจสอบแพ็กเก็ตที่ส่งแบบวนลูป*

**5.3.2.3 แบบผสม**

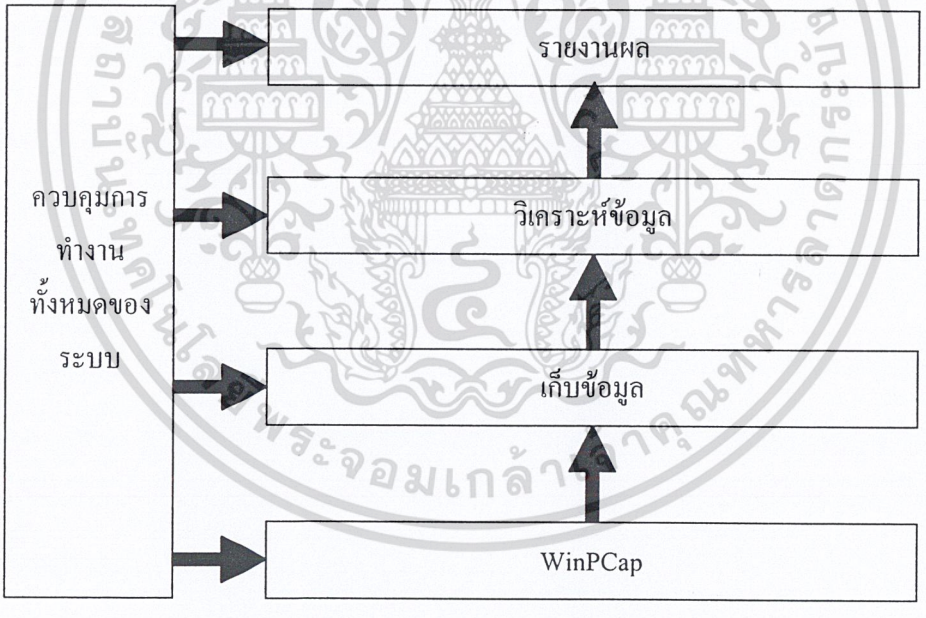
การวิเคราะห์แพ็กเก็ตประเภทนี้ให้นำวิธีการวิเคราะห์ที่กล่าวข้างต้นมาใช้ร่วมกัน เนื่องจากเกิดจากวิธีการที่ผสมผสานกันระหว่างวิธีต่างๆ ที่ได้กล่าวมาแล้ว ซึ่งสามารถแยกวิเคราะห์ออกเป็นแต่ละแบบหรือวิเคราะห์รวมกันก็ได้

# บทที่ 6
# การทำงานของระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

## 6.1 การทำงานของระบบ

ระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ พัฒนาขึ้นจาก Visual C++ เวอร์ชัน 6 ร่วมกับไลบรารี่ WinPCap โดยไลบรารีดังกล่าวจะทำงานอยู่ในชั้นล่างสุดของโปรแกรม การทำงานของไลบรารีจะทำหน้าที่ควบคุมการทำงานของการ์ดอีเธอร์เน็ต

การทำงานในส่วนของการเก็บข้อมูลที่ได้จากไลบรารี WinPcap และการวิเคราะห์ข้อมูลจะแยกกันทำงานกันอย่างอิสระในรูปแบบมัลติเธรด โดยส่วนของการเก็บข้อมูลจะนำข้อมูลจากไลบรารีมาเก็บในบัฟเฟอร์ของโปรแกรมที่พัฒนาจากโครงสร้างการเก็บข้อมลแบบลิงส์ลิสต์ เมื่อระบบต้องการวิเคราะห์จะทำการงานข้อมูลจากบัฟเฟอร์ดังกล่าวมาประมวลผล เสมือนว่าบัฟเฟอร์เป็นตัวกลางเชื่อมการทำงานระหว่างไลบรารีกับการวิเคราะห์ข้อมูล ดังโครงสร้างในรูป 6-1

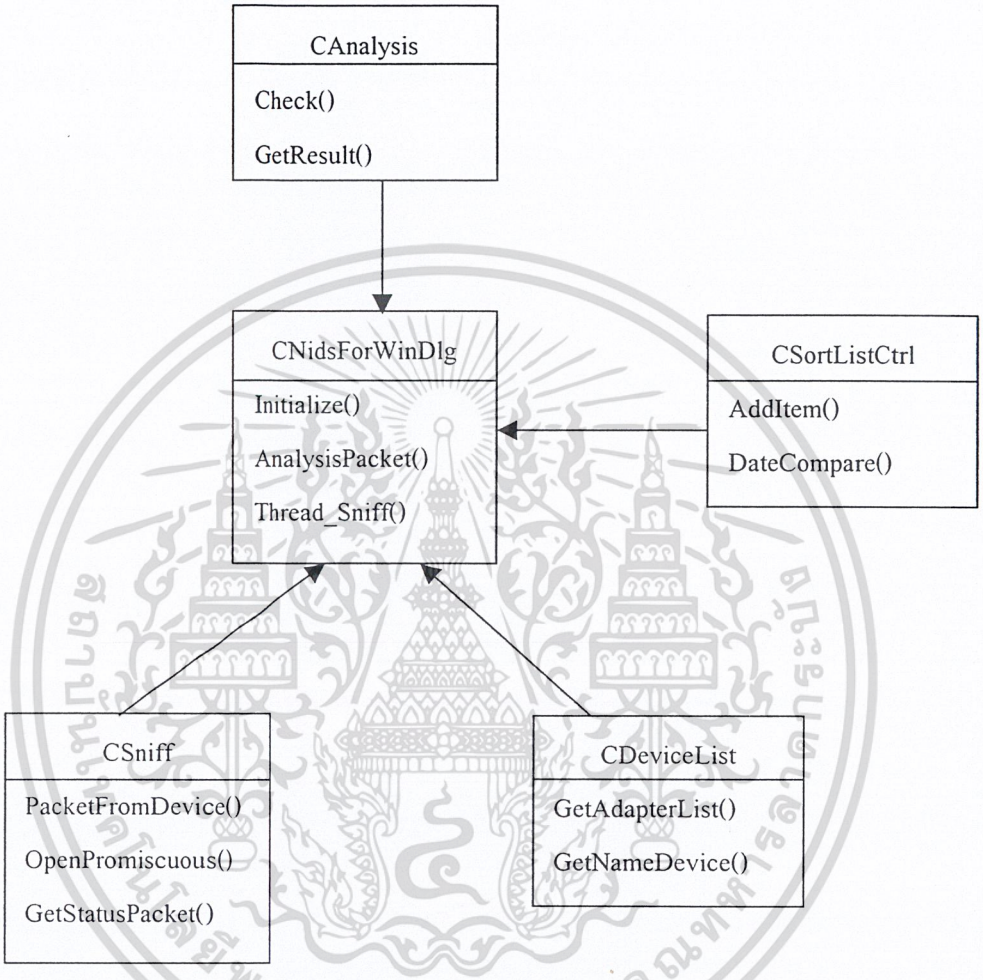



*รูปที่ 6-1 โครงสร้างของระบบ*

## 6.2 คลาสไดอะแกรมของระบบ

คลาสไดอะแกรมแบ่งเป็น 2 ส่วนคือ

### 6.2.1 คลาสไดอะแกรมหลักของระบบ

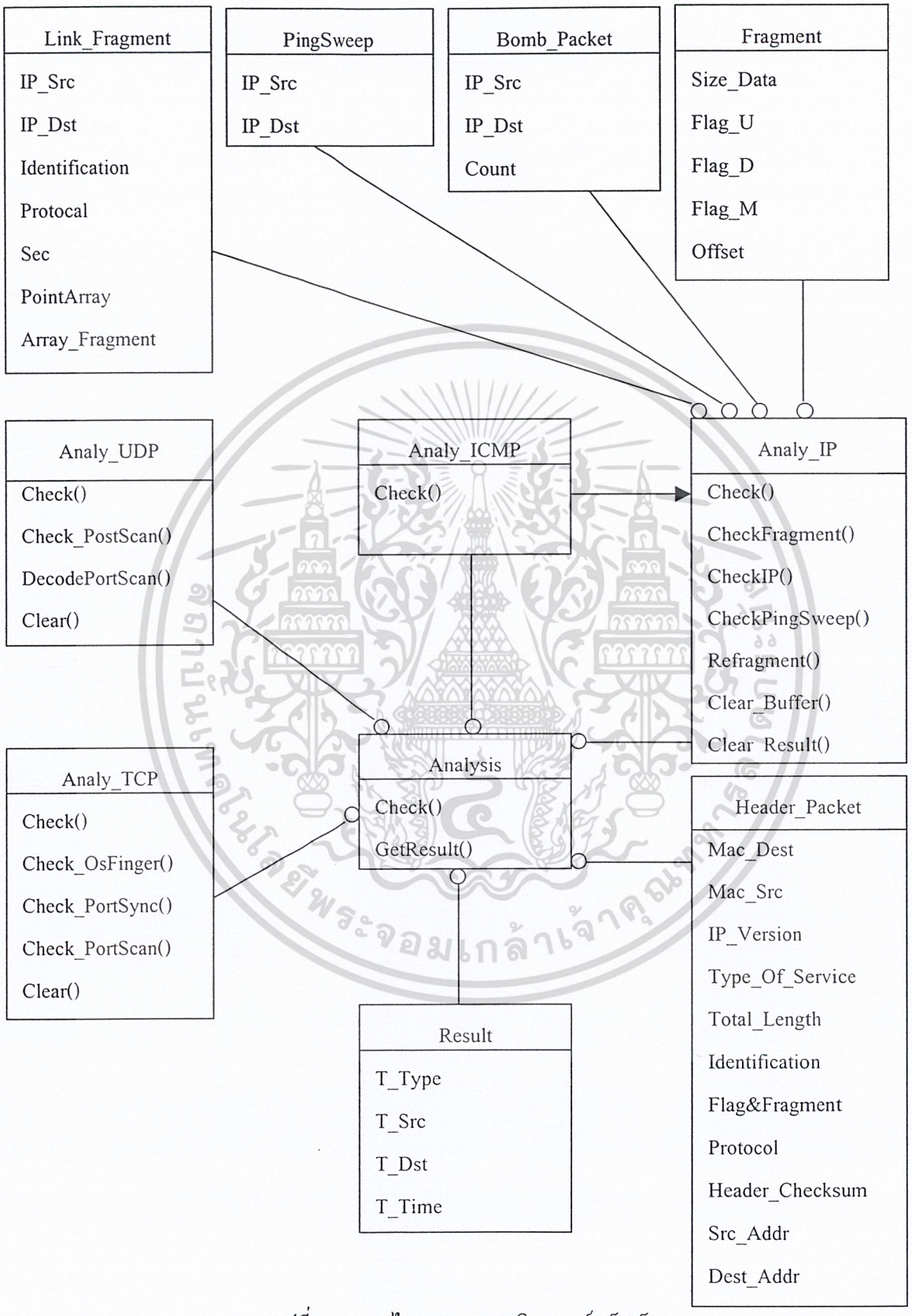



*รูปที่ 6-2 คลาสไดอะแกรมหลักของระบบ*

หน้าที่ของส่วนต่างๆ มีดังนี้

1. CNidsForWinDlg ทำหน้าที่ควบคุมการทำงานทั้งหมดของระบบ
2. CSorttListCtrl ทำหน้าที่แสดงผลการโจมตี
3. CDeviceList ทำหน้าที่เลือกการ์ดแลนที่ต้องการใช้งาน
4. CSniff ทำหน้าที่ดักจับแพ็กเก็ตข้อมูลในระบบผ่านทางการ์ดแลน
5. CAnalysis ทำหน้าที่วิเคราะห์ข้อมูลที่ดักจับได้

### 6.2.2 คลาสไดอะแกรมส่วนการวิเคราะห์การบุกรุก




*รูปที่ 6-3 คลาสไดอะแกรมการวิเคราะห์แพ็กเก็ต*

### 6.3 รายละเอียดการทำงานของโปรแกรมแต่ละส่วน

เนื่องจากแต่ละส่วนทำงานแยกจากกันอย่างอิสระจากความลักษณะการทำงานแบบคลาส ฟังก์ชันการทำงานแต่ละส่วนสามารถแบ่งตามคลาสได้ดังนี้

#### 6.3.1 ควบคุมการทำงานทั้งหมดของระบบ

ทำหน้าที่กำหนดค่าต่างๆ และควบคุมส่วนต่างๆ ให้ทำงานร่วมกัน ในส่วนของการควบคุมจะใช้คลาส CNidsForWinDlg และโกลบอลฟังก์ชั่น โดยจะมีฟังก์ชั่นการทำงานดังต่อไปนี้

- UINT Thread_Analysis()

จุดมุ่งหมาย

เป็นส่วนที่ใช้ในการวิเคราะห์ข้อมูล โดยการทำงานในส่วนนี้จะทำงานเป็นเธรดแยกออกมาจากฟังก์ชันหลัก

ขั้นตอนการทำงาน

1. เช็คว่ามีการสั่งหยุดการทำงานหรือยัง
2. สั่งให้ทำการนำข้อมูลออกมาจากลิงค์ลิสต์ของ CSniff ด้วย ฟังก์ชั่น GetPacket (Header_Packet &Packet) ของคลาส CSniff
3. เคลียร์ลิงค์ลิสต์ของแพ็กเก็ตในกรณีที่มีการโจมตี โดยวิธีรีแฟร็กเมนต์แบบ Gap โดยฟังก์ชั่น ClearSniff() ของคลาส CSniff

- UINT Thread_Sniff()

จุดมุ่งหมาย

เป็นส่วนที่ใช้ในการเก็บข้อมูล โดยการทำงานในส่วนนี้จะทำงานเป็นเธรดแยกออกมาจากฟังก์ชันหลัก

ขั้นตอนการทำงาน

1. เช็คว่ามีการสั่งหยุดการทำงานหรือยัง
2. สั่งให้ทำการเก็บข้อมูลจากการ์ดแลนด้วย ฟังก์ชั่น PacketFromDevice() ของคลาส Csniff

- void CNidsForWinDlg::OnOK()

จุดมุ่งหมาย

เป็นส่วนที่ใช้ในการเริ่มการทำงานของเธรดตางๆ และสั่งให้เธรดเริ่มการทำงาน

ขั้นตอนการทำงาน

1. ตรวจว่ามีการเปิดโพรมิสคูอัสของการ์ดแลนหรือยัง
2. ทำการเปิดโพรมิสคูอัสของการ์ดแลน
3. สั่งงานให้ Thread_Sniff() ทำงาน

4. สั่งให้ Thread_Analysis() ทำงาน

- void CNidsForWinDlg::OnCancel()

จุดมุ่งหมาย

เป็นส่วนที่ใช้ในการหยุดการทำงานของเธรดและสั่งปิดโปรแกรม

ขั้นตอนการทำงาน

1. สั่งให้ Thread_Sniff() หยุดการทำงาน
2. สั่งให้ Thread_Analysis() หยุดการทำงาน
3. ตรวจว่ามีการเปิดโพรมิสคูอัสของการ์ดแลนหรือยัง
4. ทำการปิดโพรมิสคูอัสของการ์ดแลน
5. ปิดโปรแกรม

- void void CNidsForWinDlg::OnStop()

จุดมุ่งหมาย

เป็นส่วนที่ใช้ในการหยุดการทำงานของเธรดและสั่งหยุดการทำงานของโปรแกรม

ขั้นตอนการทำงาน

1. สั่งงานให้ Thread_Sniff() หยุดการทำงาน
2. สั่งงานให้ Thread_Analysis() หยุดการทำงาน
3. ตรวจว่ามีการเปิดโพรมิสคูอัสของการ์ดแลนหรือยัง
4. ทำการปิดโพรมิสคูอัสของการ์ดแลน

- void CNidsForWinDlg::OnSelectAdap()

จุดมุ่งหมาย

เป็นส่วนที่ใช้ในการเลือก การ์ดอีเทอร์เน็ต

ขั้นตอนการทำงาน

1. สั่งให้ Thread_Sniff() หยุดการทำงาน
2. สั่งให้ Thread_Analysis() หยุดการทำงาน
3. ตรวจว่ามีการเปิดโพรมิสคูอัสของการ์ดแลนหรือยัง
4. ทำการปิดโพรมิสคูอัสของการ์ดแลน
5. ทำการเลือก การ์ดแลน

### 6.3.2 WinPcap

WinPcap เป็นไลบรารีที่ทำการติดต่อกับการ์ดแลน เพื่อทำการควบคุมการทำงานของการ์ดแลน ฟังก์ชันที่สำคัญในไลบรารีนี้มีดังนี้

- ULONG PacketGetAdapterNames()

  เป็นฟังก์ชั่นแรกที่ใช้ในการติดต่อกับไดร์เวอร์ โดยที่ฟังก์ชั่นนี้จะส่งชื่อของการ์ดแลนที่ถูกติดตั้งอยู่ภายในระบบออกมา

- LPADAPTER PacketOpenAdapter()

  เป็นฟังก์ชั่นที่รับชื่อของการ์ดแลนและจะทำการรีเทิร์นพอยต์เตอร์เกี่ยวกับการ initialized การ์ดแลนโดยจะได้มาจากฟังก์ชั่น PacketGetAdapterNames

- BOOLEAN PacketSetHwFilter()

  เป็นฟังก์ชั่นที่ใช้ในการกำหนดการทำงานของการ์ดแลนให้รับแพ็กเก็ตรูปแบบใด มีรายละเอียดดังนี้ Hardware Filter โดยมี parameter ดังนี้

  - NDIS_PACKET_TYPE_PROMISCUOUS
  - NDIS_PACKET_TYPE_DIRECTED
  - NDIS_PACKET_TYPE_BROADCAST
  - NDIS_PACKET_TYPE_MULTICAST
  - NDIS_PACKET_TYPE_ALL_MULTICASTNDIS_PACKET_TYPE_ALL_LOCAL

- BOOLEAN PacketSetBuff()

  เป็นฟังก์ชั่นที่กำหนดขนาดของบัฟเฟอร์

- BOOLEAN PacketSetReadTimeout ()

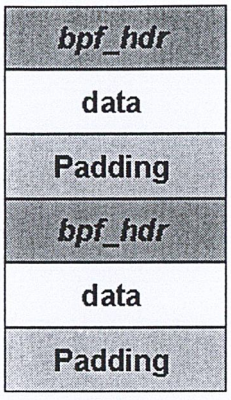
  เป็นฟังก์ชั่นที่กำหนดค่าไทมเอาท์ ซึ่งมีหน่วยเป็นมิลลิวินที หากกำหนดป็น 0 แสดงว่าไม่มีค่าไทม์เอาท์

- LPPACKET PacketAllocatePacket()

  กำหนดโครงสร้างแพ็กเก็ต และส่งค่ากลับเป็นพอยต์เตอร์

- BOOLEAN PacketReceivePacket()

  เป็นฟังก์ชั่นที่ใช้ในการเก็บข้อมูลขึ้นมาจากการ์ดแลน สำหรับโครงสร้างข้อมูลทำการเก็บในบัฟเฟอร์มีลักษณะดังรูปที่ 6-4




*รูปที่ 6-4 แสดงการเก็บข้อมูลในบัฟเฟอร์ของไลบราลี*

- BOOLEAN PacketGetStats()

  เป็นฟังก์ชั่นที่ใช้ในการแสดงสถานะของการทำงาน เช่นปริมาณแพ็กเก็ตที่รับได้และเกิดการสูญหาย

- VOID PacketFreePacket()

  เป็นฟังก์ชั่นที่ใช้ในการเคลียร์ข้อมูลแพ็กเก็ตในบัฟเฟอร์

- VOID PacketCloseAdapter()

  เป็นฟังก์ชั่นที่ใช้สำหรับหยุดการติดต่อกับการ์ดแลน

### 6.3.3 การเก็บข้อมูล

ในการเก็บข้อมูลมาทำการวิเคราะห์ โปรแกรมจะเก็บข้อมูลส่วนหัว (Header) ของแต่ละแพ็กเก็ตในชั้นต่างๆ เช่นเก็บเฉพาะข้อมูลส่วนหัวของแพ็กเก็ตในชั้นไอพีและทีซีพี ในการเก็บข้อมูลจะใช้คลาส Csniff เป็นตัวเก็บข้อมูล โดยมีฟังก์ชั่นการทำงานดังต่อไปนี้

- BOOL OpenPromiscuous()

  จุดมุ่งหมาย

  เพื่อทำการเปิดโพรมิสคูอัสของการ์ดแลน

  ขั้นตอนการทำงาน

  1. เช็คว่ามีการเปิดโพรมิสคูอัสของการ์ดแลนอยู่หรือไม่
  2. ทำการเปิดโพรมิสคูอัสของการ์ดแลนตามค่า NumberDevice

- BOOL ClosePromiscuous()

จุดมุ่งหมาย

เพื่อทำการปิดโพรมิสคูอัสของการ์ดแลน

ขั้นตอนการทำงาน

1. เช็คว่ามีการเปิดโพรมิสคูอัสของการ์ดแลนอยู่หรือไม่
2. ทำการปิดโพรมิสคูอัสของการ์ดแลนที่เปิดอยู่

- BOOL GetStatusPacket()

จุดมุ่งหมาย

เพื่อตรวจสอบว่ามีการรับแพ็กเก็ตเข้ามาเท่าไหร่ แล้วมีการสูญหายไปเท่าไหร่

ขั้นตอนการทำงาน

1. เช็คว่ามีการเปิดโพรมิสคูอัสของการ์ดแลนอยู่หรือไม่
2. ทำการดึงค่าขึ้นมาจาก Winpcap

- BOOL GetPacket()

จุดมุ่งหมาย

เพื่อนำค่าเฮดเดอร์ของแพ็กเก็ตที่เก็บไว้ส่งออกไปให้ฟังก์ชั่นอื่นทีละ 1 แพ็กเก็ต

ขั้นตอนการทำงาน

1. เช็คว่ามีแพ็กเก็ตอยู่ในลิงค์ลิสต์หรือไม่
2. อ่านค่าใน head ของลิงค์ลิสต์ออกมา
3. ลบส่วนหัวของลิงค์ลิสต์ทิ้ง

- BOOL PacketFromDevice()

จุดมุ่งหมาย

เพื่อนำข้อมูลเฮดเดอร์แพ็กเก็ตในการ์ดแลนออกมา แล้วนำมาเข้าลิงค์ลิสต์

ขั้นตอนการทำงาน

1. เช็คว่ามีการเปิดโพรมิสคูอัสของการ์ดแลนอยู่หรือไม่
2. จัดการนำข้อมูลใน buffer ของการ์ดแลนออกมา
3. ทำการแยกออกเป็นแพ็กเก็ตโดยจะแยกเอาเฉพาะส่วนหัวของแพ็กเก็ต
4. นำเข้าไปเก็บในลิงค์ลิสต์โดยที่ 1 โหนด จะเก็บ 1 แพ็กเก็ต

- BOOL ClearSniff()

จุดมุ่งหมาย

เพื่อลบข้อมูล ในลิงค์ลิสต์ทั้งหมด

ขั้นตอนการทำงาน

1. จะทำการลบข้อมูลในลิงค์ลิสต์

- BOOL GetStatusPromiscuous()

จุดมุ่งหมาย

เพื่อเช็คดูว่ามีการเปิดโพรมิสคูอัสของการ์ดแลนหรือไม่

ขั้นตอนการทำงาน

1. เช็คว่ามีการเปิดโพรมิสคูอัสของการ์ดแลนอยู่หรือไม่
2. ส่งผลลัพธ์กลับสู่ฟังก์ชันหลักว่าเปิดหรือไม่

### 6.3.4 การวิเคราะห์ข้อมูล

ในส่วนของการวิเคราะห์ข้อมูล จะใช้คลาสที่ทำหน้าที่วิเคราะห์ข้อมูลดังนี้ คลาส CAnalysis, คลาส CAnaly_IP, คลาส CAnaly_ICMP, คลาส CAnaly_TCP, และคลาส CAnaly_UDP โดยมีฟังก์ชันการทำงานดังต่อไปนี้

- BOOL CAnaly_IP::Check()

จุดมุ่งหมาย

เพื่อเช็คแพ็กเก็ตที่ได้รับมา แล้วรายงาน

ขั้นตอนการทำงาน

1. ส่งแพ็กเก็ตที่ได้รับมาไปวิเคราะห์ เพื่อแจ้งผล
2. ตรวจสอบว่าแพ็กเก็ตไหนที่ยังไม่สามารถรีแฟร็กเมนต์ได้ในระยะเวลาที่กำหนด เพื่อแจ้งผล

- BOOL CAnaly_ICMP::Check()

จุดมุ่งหมาย

เพื่อเช็คแพ็กเก็ตที่ได้รับมา แล้วรายงาน

ขั้นตอนการทำงาน

1. ส่งแพ็กเก็ตที่ได้รับมาไปวิเคราะห์ เพื่อแจ้งผล

- BOOL CAnalysis::Check()

จุดมุ่งหมาย

เพื่อแยกชนิดของแพ็กเก็ตว่าเป็นชนิดไหนเพื่อจะได้นำไปวิเคราะห์ ต่อไป

ขั้นตอนการทำงาน

1. นำแพ็กเก็ตที่ได้มาตรวจสอบว่าเป็นประเภทใด
2. นำไปทำการวิเคราะห์ ตามประเภทของแพ็กเก็ต
3. ทุกประเภทจะต้องผ่านการวิเคราะห์ จากหมายเลขไอพีก่อน ยกเว้นไอซีเอ็มพี ซึ่งจะมีการวิเคราะห์ ที่แยกออกไป แต่เป็นการสืบทอดมาจากวิเคราะห์แบบหมายเลขไอพี

- BOOL CAnaly_IP::CheckBomb() and BOOL CAnaly_ICMP::CheckBomb()

จุดมุ่งหมาย

เพื่อตรวจสอบว่ามีแพ็กเก็ตจากหมายเลขไอพีหนึ่ง ไปยังอีกหมายเลขไอพีหนึ่งมากเกินไปหรือไม่ในเวลาที่เหมาะสม

ขั้นตอนการทำงาน

1. นำแพ็กเก็ตที่ได้มาดูว่ามาจากหมายเลขไอพีไหน แล้วส่งไปยังหมายเลขไอพีไหน
2. แล้วทำการนับว่าเกินกว่าที่กำหนดหรือไม่ ( 2000 แพ็กเก็ต / วินาที )
3. ถ้าครบจะมีการเปรียบเทียบกับข้อมูลของการโจมตีแบบนี้สำหรับหมายเลขไอพีนี้ ว่ามีการโจมตีแบบนี้ไปแล้วหรือยัง เพื่อจะได้ไม่มีการแจ้งบ่อยเกินไป

- BOOL CAnaly_IP::CheckIP() และ BOOL CAnaly_ICMP::CheckIP()

จุดมุ่งหมาย

เพื่อตรวจสอบว่ามีแพ็กเก็ตจากหมายเลขไอพีต้นทางไปยังหมายเลขไอพีปลายทางเป็นหมายเลขไอพีเดียวกันหรือไม่

ขั้นตอนการทำงาน

1. นำแพ็กเก็ตที่ได้มาดูว่ามาจากหมายเลขไอพีไหน แล้วส่งไปยังหมายเลขไอพีไหน
2. ดูว่าหมายเลขไอพีตรงกันหรือไม่
3. ถ้าตรงกันจะมีการเปรียบเทียบกับข้อมูลของการโจมตีแบบนี้สำหรับหมายเลขไอพีนี้ ว่ามีการโจมตีแบบนี้ไปแล้วหรือยัง เพื่อจะได้ไม่มีการแจ้งบ่อยเกินไป

- BOOL CAnaly_IP::CheckFragment() และ BOOL CAnaly_ICMP::CheckFragment()

จุดมุ่งหมาย

เพื่อแยกแพ็กเก็ตตามหมายเลขไอพีต้นทาง, หมายเลขไอพีปลายทาง, โพรโตคอลและฟิลด์ Identifier ตรวจสอบว่ามีแพ็กเก็ตจากหมายเลขไอพีต้นทางไปยังหมายเลขไอพีปลายทางต้องการทำแฟร็กเมนต์หรือไม่

ขั้นตอนการทำงาน

1. เช็คว่าแพ็กเก็ตที่ได้รับมาเป็นแพ็กเก็ตที่จะต้องทำรีแฟร็กเมนต์หรือไม่

2. จะดูว่ามีข้อมูลเดิมอยู่หรือไม่ โดยจะดูที่หมายเลขไอพีต้นทาง ,หมายเลขไอพีปลายทาง , โพรโตคอลและ Identification
3. นำเข้าสู่ลิงค์ลิสต์

- BOOL CAnaly_IP::ReFragment() and CAnaly_ICMP::ReFragment()

จุดมุ่งหมาย

เพื่อตรวจสอบว่าแพ็กเก็ตมีปัญหาเรื่องการแฟร็กเมนต์หรือไม่

ขั้นตอนการทำงาน

1. เช็คว่าในแพ็กเก็ตที่ได้รับมาแล้วนั้นมี ส่วนหัวของแพ็กเก็ตและส่วนท้ายของแพ็กเก็ตหรือยัง
2. จะลองประกอบดูว่าสามารถประกอบได้หรือยัง
3. ถ้าประกอบได้จะลบออกจากลิงค์ลิสต์

- void CAnaly_IP::Clear() and void CAnaly_ICMP::Clear()

จุดมุ่งหมาย

เพื่อเคลียร์ข้อมูลการโจมตี

ขั้นตอนการทำงาน

1. ทำการเคลียร์ข้อมูลการโจมตี

### 6.3.5 รายงานผล

ในส่วนของการควบคุมการแสดงผลจะใช้ คลาส CSortLstCtrl โดยเป็นคลาสที่ช่วยในการแสดงผลโดยเฉพาะ

- void CNidsForWinDlg::AddItem()

จุดมุ่งหมาย เป็นส่วนที่ใช้ในการวิเคราะห์ข้อมูล โดยจะทำงานเป็นเธรด

ขั้นตอนการทำงาน

1. นำข้อมูลผลลัพธ์จากการโจมตีเพิ่มเข้ายังส่วนแสดงผล
2. รอรับการคลิกส่วนหัวของหัวข้อที่แสดง เพื่อทำการเรียงข้อมูลให้อ่านได้ง่าย

### 6.4 การติดตั้งระบบตรวจจับ

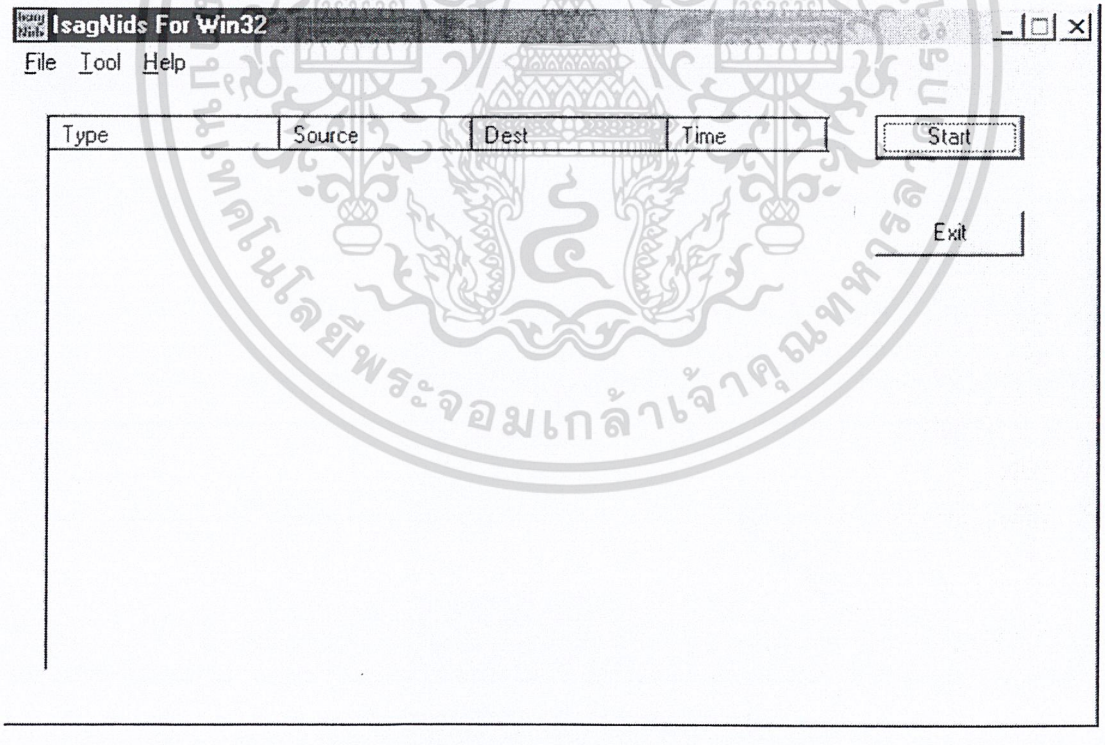
เนื่องจากการทำงานของระบบตรวจจับการบุกรุกผ่านเครือขายคอมพิวเตอร์จะต้องทำงานร่วมกับไลบราลี WinPcap ทำให้การติดตั้งเพื่อใช้งานโปรแกรมต้องแบ่งเป็น 2 ขั้นตอนดังนี้

(1) **การติดตั้งไลบราลี WinpCap** การติดตั้งไลบราลี่ WinpCap ทำได้โดยการเอ็กซิคิวต์ไฟล์ Setup.exe ในซับไดเร็กทอรี่ WinpCap จากนั้นให้ทำการคลิก Next โปรแกรมติดตั้งจะเริ่มทำการติดตั้งตัวไลบราลี่ลงสู่ระบบ จากนั้นให้ผู้ใช้ทำการรีสตาร์ทเครื่องใหม่

(2) **การติดตั้งระบบโปรแกรม IsagNids** การติดตั้งโปรแกรม IsagNids ทำได้โดยการเอ็กซิคิวต์ไฟล์ Setup.exe ในซับไดเร็กทอรี่ IsagNids จากนั้นให้ทำการเลือกซับไดเร็กทอรี่ที่ต้องการติดตั้ง โปรแกรมติดตั้งจะเริ่มทำสำเนาโปรแกรมระบบตรวจจับลงสู่ระบบ โดยไม่จำเป็นต้องรีสตาร์ทเครื่องใหม่

### 6.5 การใช้งานระบบตรวจจับ

ระบบตรวจจับการบุกรุกถูกพัฒนาให้สามารถทำงานในระบบจียูไอ (GUI : Graphics User Interface) เพื่อให้ผู้ใช้สามารถใช้งานได้ง่าย เมื่อผู้ใช้ทำการติดตั้งโปรแกรมเสร็จเรียบร้อยแล้ว จะเห็นลักษณะโปรแกรมดังนี้



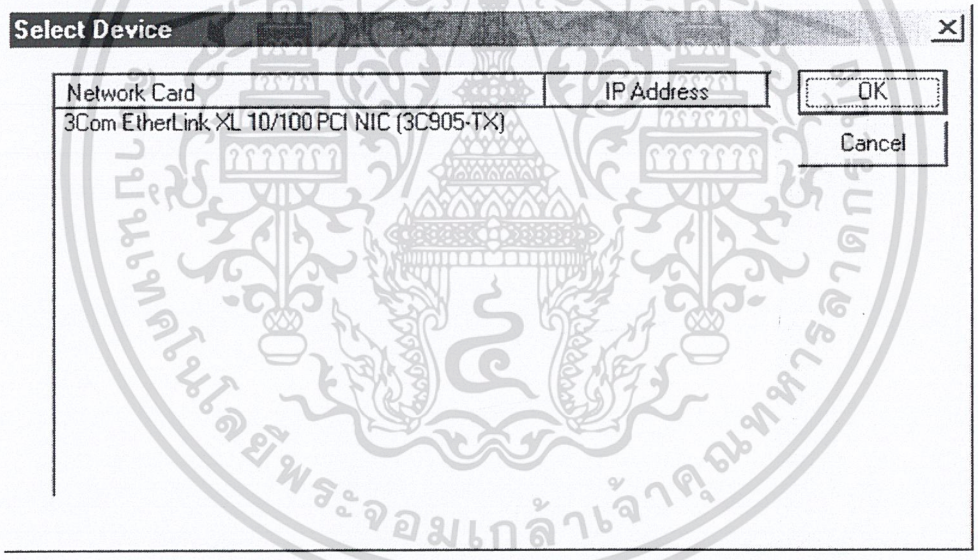

*รูปที่ 6-5 หน้าจอโปรแกรม IsagNids for Win32*

การใช้งานระบบตรวจจับผู้บุกรุกมีขั้นตอนการใช้งานดังนี้

### 6.5.1 การกำหนดการ์ดแลนสำหรับตรวจจับ

เมื่อเริ่มต้นใช้งานระบบตรวจจับ จะต้องทำการกำหนดการ์แลนสำหรับทำการตรวจจับ โดยการ์ดดังกล่าวจะต้องเชื่อมต่ออยู่กับสื่อที่ได้รับการกระจายสัญญาณเดียวกันกับเครือข่ายที่ต้องการตรวจจับการบุกรุก

เนื่องจากระบบตรวจจับถูกออกแบบมาให้สามารถทำงานกับการ์ดแลนที่ติดตั้งเพียงไดรเวอร์ของการ์ดได้ ทำให้ไม่จำเป็นต้องมีการกำหนดหมายเลขไอพีให้กับการ์ดแลน (เนื่องจากอาจเป็นช่องทางในการถูกโจมตีจากผู้โจมตีได้) การติดตั้งการ์ดแลนเข้ากับระบบปฏิบัติการจึงต้องทำอย่างสมบูรณ์ กล่าวคือมีการติดตั้งและถอนการติดตั้งอย่างสมบูรณ์ผ่านทางบริการที่ระบบมีไว้บริการ จึงจะทำให้ขั้นตอนการกำหนดการ์ดแลนทำได้อย่างมีประสิทธิภาพ มิฉะนั้นอาจทำให้ไม่สามารถกำหนดการ์ดแลนได้ ส่งผลให้ไม่สามารถทำการตรวจจับแพ็กเก็ตข้อมูลได้ การกำหนดการ์ดดีเธอร์เน็ตจะทำผ่านไดอะล็อกการกำหนดการ์แลน ผ่านทางเมนู Tool -> Select Adapter ดังรูปที่ 6-6




*รูปที่ 6-6 การกำหนดการ์ดอีเธอร์เน็ตผ่านทางไดอะล็อก*

### 6.5.2 การดูข้อมูลการบุกรุก

หลังจากเลือกการ์ดอินเทอร์เน็ตที่ต้องการแล้ว ผู้ใช้สามารถคลิกที่ปุ่ม Start เพื่อสั่งให้โปรแกรมทำงานได้ทันที จากนั้นระบบจะเริ่มทำการวิเคราะห์ทุกแพ็กเก็ตที่อยู่ในเครือข่าย หากมีการบุกรุกทั้งจากการสำรวจระบบหรือการโจมตี ระบบจะมีการแสดงผลการวิเคราะห์แยกเป็นส่วนต่างๆ ดังนี้

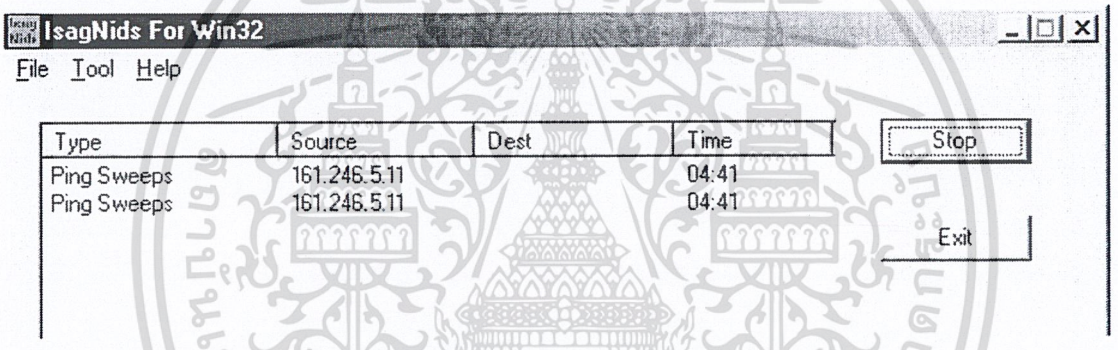
- Type แสดงรูปแบบการโจมตี แบ่งเป็นรูปแบบการโจมตีดังนี้
  - ❑ Ping sweeps
  - ❑ Scan port

- ❑ OS Finger print
- ❑ Synflood
- ❑ Bomb
- ❑ Overlap
- ❑ Gap

- Source หมายเลขไอพีของผู้โจมตี
- Dest หมายเลขไอพีของผู้ถูกโจมตี
- Time เวลาที่มีการโจมตี

การวิเคราะห์จะแยกทางรูปแบบการบุกรุกที่เกิดขึ้น ในการบุกรุกแต่ละครั้งอาจมีการแสดงผลเพียงรูปแบบเดียว หรือหลายรูปแบบก็ได้ ตามลักษณะของการบุกรุกที่เกิดขึ้น ดังตัวอย่างต่อไปนี้

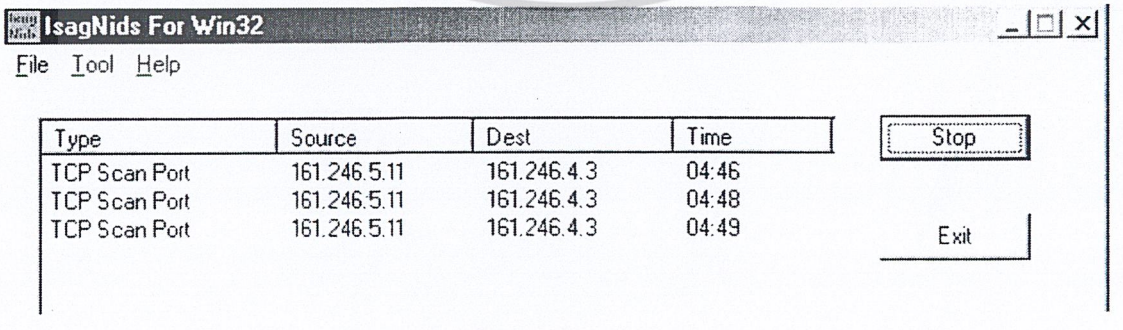
(1) การปิงสวีป




*รูปที่ 6-7 แสดงการแจ้งเตือนเมื่อเจอการบุกรุกแบบปิงสวีป*

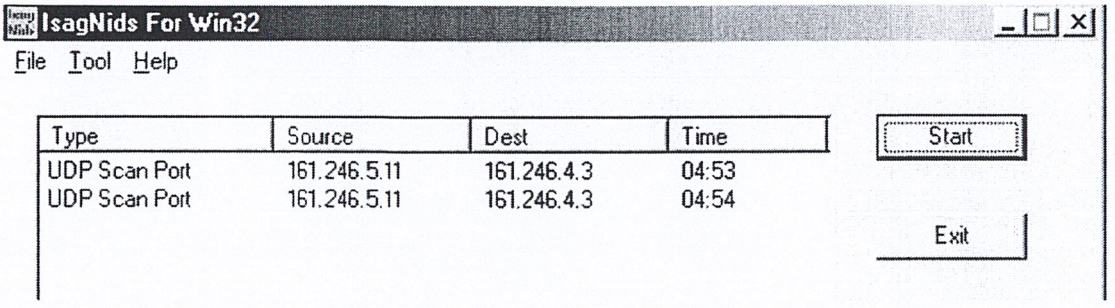
(2) การตรวจสอบการสแกนพอร์ต แบ่งเป็น 2 ประเภทคือ

- การสตรวจสอบการสแกนพอร์ตทีซีพี



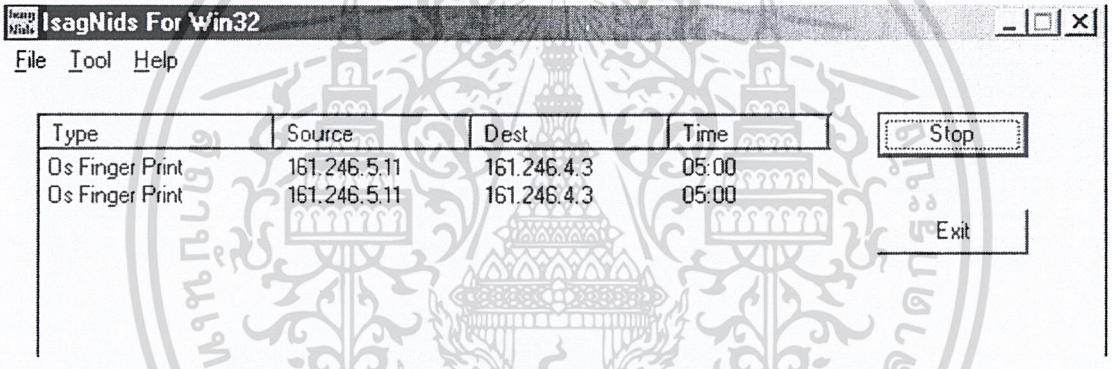

*รูปที่ 6-8 แสดงการแจ้งเตือนเมื่อเจอการบุกรุกโดยการสแกนพอร์ตทีซีพี*

- การสตรวจสอบการสแกนพอร์ตยูดีพี



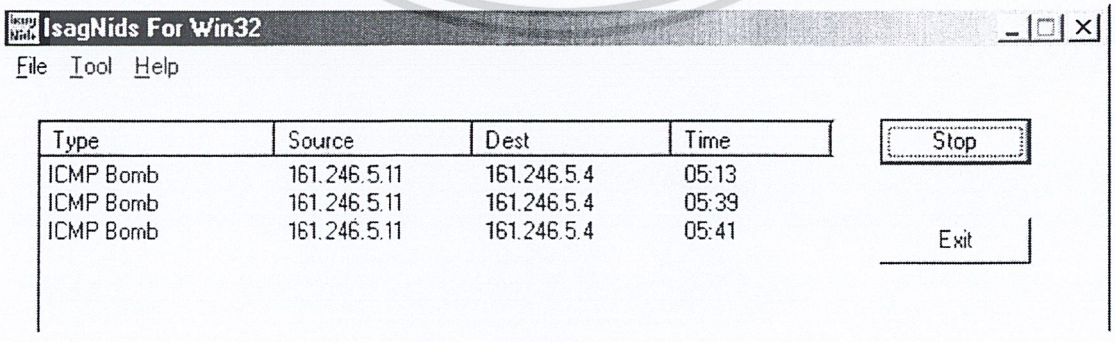

*รูปที่ 6-9 แสดงการแจ้งเตือนเมื่อเจอการบุกรุกโดยการสแกนพอร์ตยูดีพี*

(3) การตรวจสอบการตรวจสอบระบบปฏิบัติการ



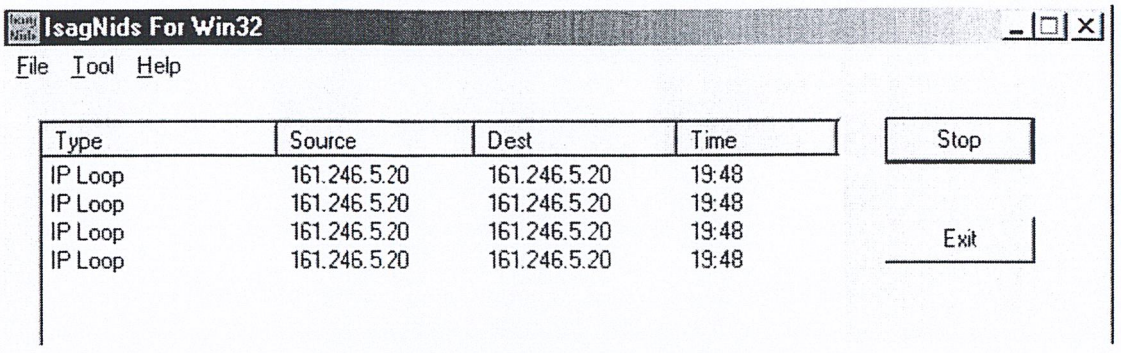

*รูปที่ 6-10 แสดงการแจ้งเตือนเมื่อเจอการบุกรุกตรวจสอบระบบปฏิบัติการ*

(4) การโจมตีแบบส่งแพ็กเก็ตจำนวนมาก



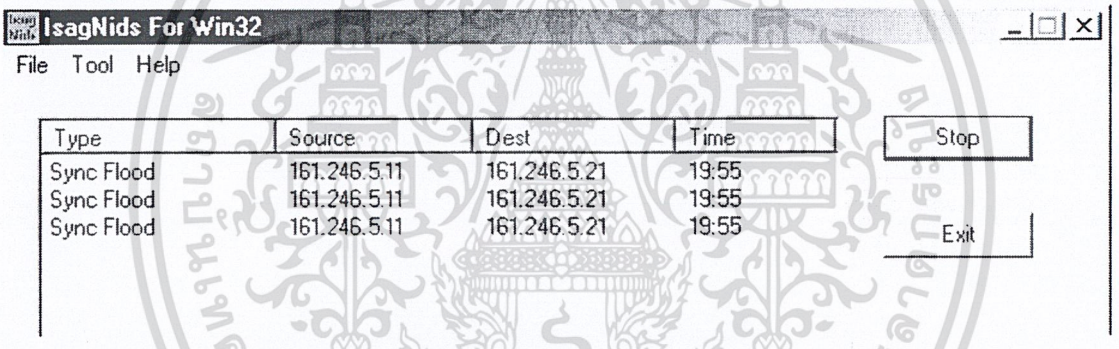

*รูปที่ 6-11 แสดงการแจ้งเตือนมีการโจมตีแบบส่งแพ็กเก็ตจำนวนมาก*

(5) การโจมตีแบบส่งแพ็กเก็ตที่มีไอพีต้นทางและปลายทางตรงกัน




*รูปที่ 6-12 แสดงการแจ้งเตือนมีการโจมตีแบบแพ็กเก็ตที่มีไอพีต้นทางและปลายทางตรงกัน*

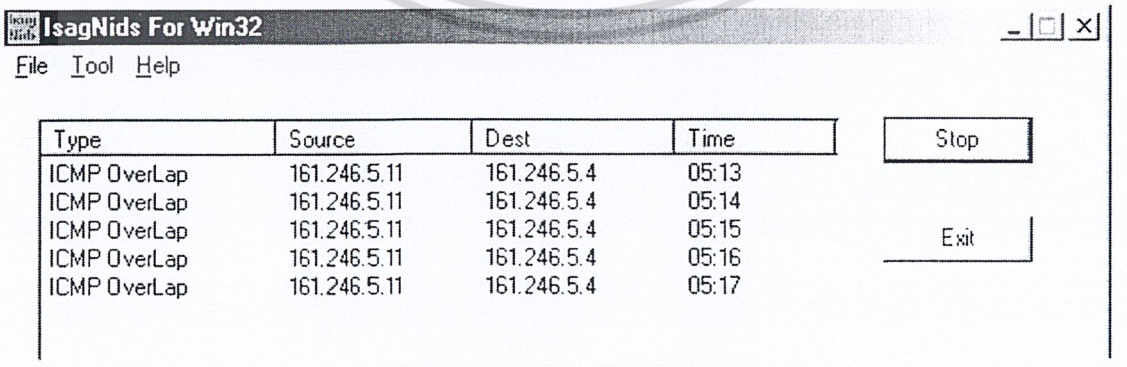
(6) การโจมตีแบบส่ง SYN flood




*รูปที่ 6-13 แสดงการแจ้งเตือนมีการโจมตีแบบส่ง SYN flood*

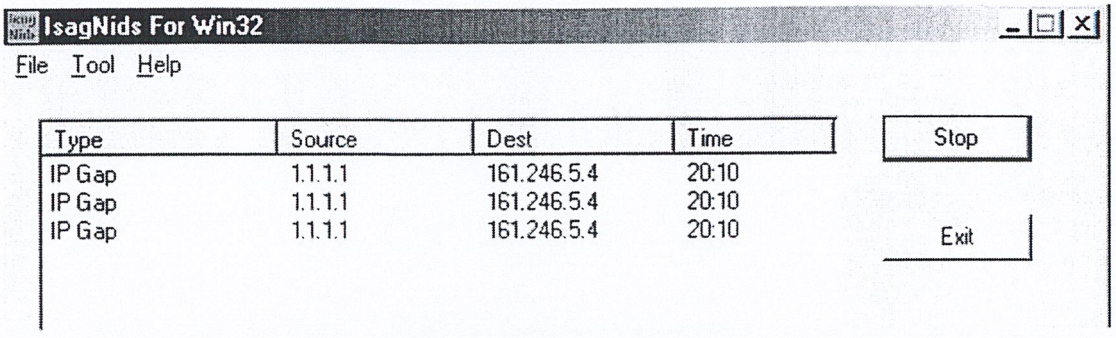
(7) การโจมตีแบบแฟรกเมนเตชัน แบ่งเป็น 2 ประเภท

- ขอบเขตของเพ็กเก็ตเหลื่อมล้ำกัน




*รูปที่ 6-14 แสดงการแจ้งเตือนมีการโจมตีแบบแฟรกเมนเตชันชนิดทับซ้อนกัน*

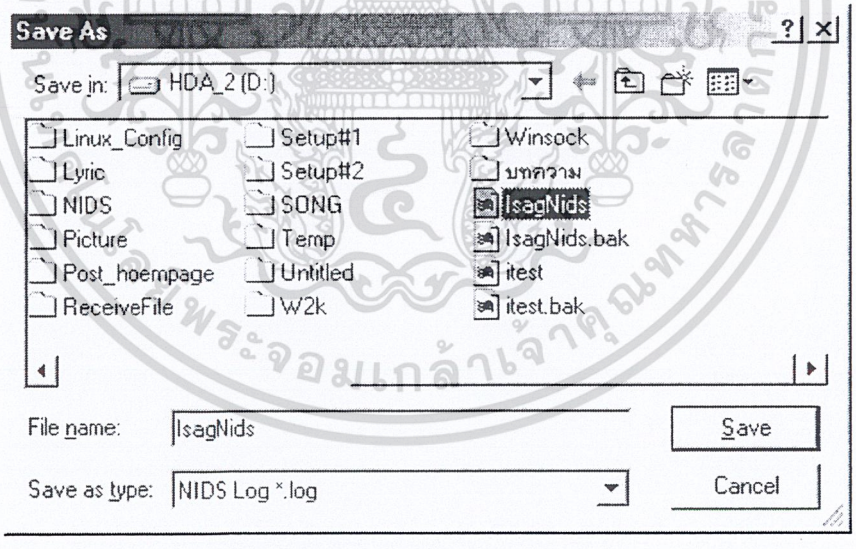
- ขอบเขตของเพ็กเก็ตเกิดช่องว่าง




*รูปที่ 6-15 แสดงการแจ้งเตือนมีการโจมตีแบบแฟรกเมนเตชันชนิดเกิดช่องว่าง*

### 6.5.3 การบันทึกข้อมูลการบุกรุก

ระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์สามารถทำการบันทึกข้อมูลการบุกรุกได้ โดยการเลือกเมนู File -> Save as จากนั้นให้ทำการกำหนดชื่อแฟ้มข้อมูลที่ต้องการบันทึกผ่านไดอะล็อกการบันทึกข้อมูลลงแฟ้มข้อมูล ดังรูป

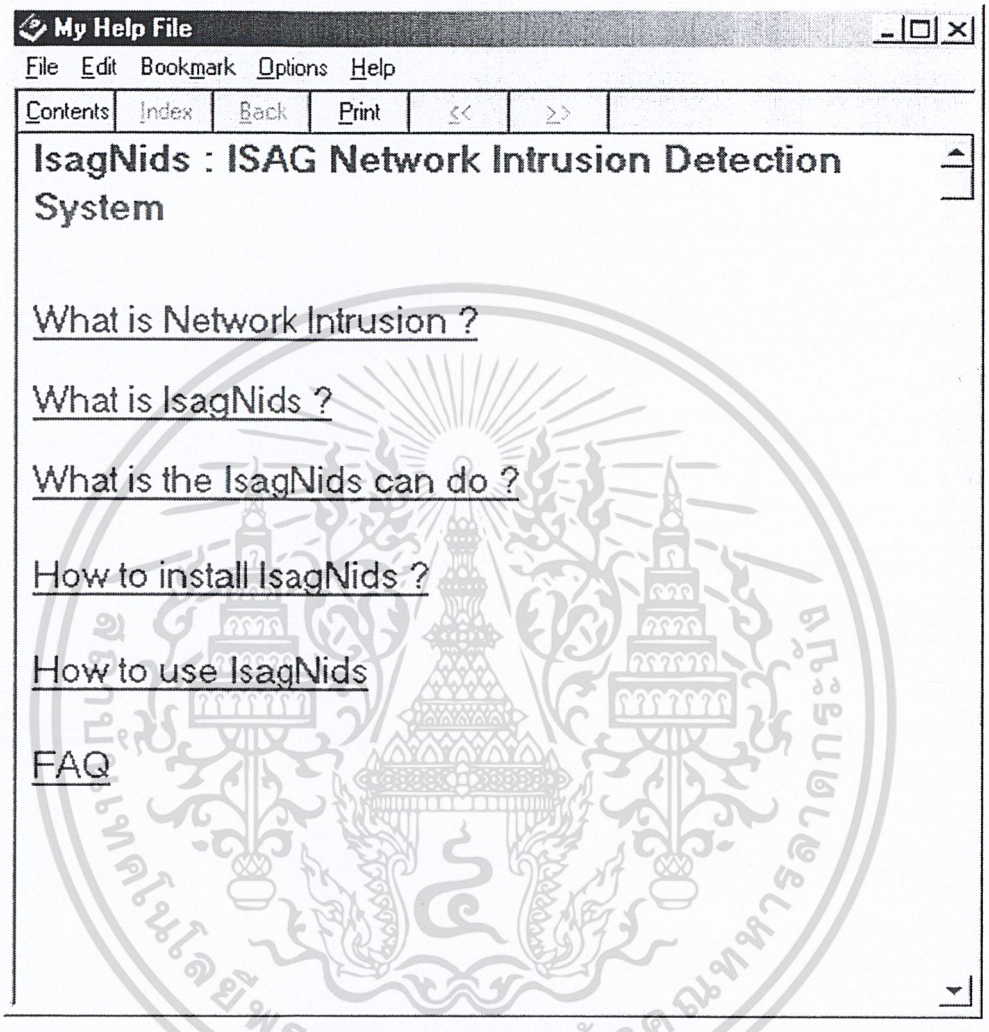



*รูปที่ 6-16 แสดงไดอะล็อกการบันทึกข้อมูลการโจมตีลงแฟ้มข้อมูล*

เมื่อต้องการนำข้อมูลที่บันทึกไว้มาแสดงผล สามารถทำได้โดยการเลือกเมนู File -> Open จากนั้นให้กำหนดชื่อแฟ้มข้อมูลที่ต้องการนำมาแสดงผล

### 6.5.4 คู่มือการใช้

ผู้ใช้สามารถเรียกดูคู่มือการเมื่อเกิดปัญหาจากการใช้งาน หรือเพื่อศึกษาหาข้อมูลเกี่ยวกับรูปแบบการบุกรุกได้โดยเรียกเมนู Help -> Content ระบบจะแสดงคู่มือการใช้ดังตัวอย่าง




*รูปที่ 6-17 แสดงตัวอย่างของคู่มือการใช้*

# บทที่ 7

# การทดสอบการทำงาน

## 7.1 การทดสอบประสิทธิภาพของระบบ

ขั้นตอนการทดสอบระบบตรวจจับผู้บุกรุกทางเครือข่ายได้ทำการทดสอบโดยมีรายละเอียดของคอมพิวเตอร์ที่ใช้ทดสอบดังนี้

เครื่องที่ติดตั้งโปรแกรม IsagNids for Win32

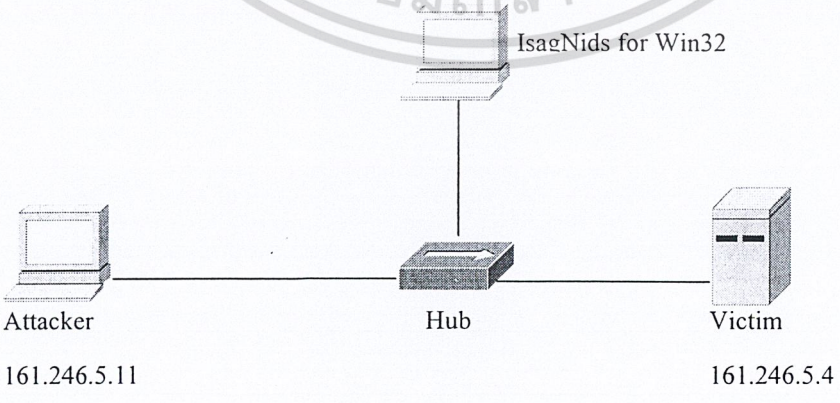
- หน่วยประมวลผลความเร็ว 400 Mhz
- หน่วยความจำหลัก 256 MB
- ระบบปฏิบัติการไมโครซอฟท์วินโดว์ 2000 ที่ติดตั้งเซอร์วิสแพ็ก 1
- การ์ดแลนความเร็ว 100 Mb

เครื่องที่ใช้ทำการโจมตี

- หน่วยประมวลผลความเร็ว 150 Mhz
- หน่วยความจำหลัก 64 MB
- ระบบปฏิบัติการลีนุกซ์สแลคแวร์ 7.0 เคอร์เนลเวอร์ชัน 2.4.10
- การ์ดแลนความเร็ว 100 Mb

## 7.2 โครงสร้างทางเครือข่ายของระบบทดสอบ

ระบบเครือข่ายที่ใช้ทดสอบในบทนี้ เป็นระบบที่ทำงานบนการ์ดแลนและอุปกรณ์ที่มีความเร็วทั้งระบบ 100 เมกกะบิต มีลักษณะดังรูป




*รูปที่ 7-1 แสดงโครงสร้างเครือข่ายในการทดสอบ*

**7.3 เครื่องมือที่ใช้ทดสอบการโจมตี**

เนื่องจากในปัจจุบันได้มีผู้พัฒนาเครื่องมือที่ใช้ในการโจมตีเพื่อให้ปฏิเสธการให้บริการกันอย่างมากมาย ส่วนใหญ่สามารถดาวน์โหลดได้ฟรีจากอินเทอร์เน็ต ความแตกต่างกันของเครื่องมือแต่ละตัวคือรูปแบบการโจมตีและระดับความรุนแรงในการโจมตี

การทดสอบการทำงานในบทนี้ได้เลือกเครื่องมือที่ได้รับความนิยมใช้กันอย่างแพร่หลาย และมีระดับความรุนแรงสูงมาใช้ทำการทดสอบ โดยมีรายชื่อเครื่องมือที่เลือกใช้พารามิเตอร์ที่สั่งให้เครื่องมือดังกล่าวทำงาน ดังนี้

| ประเภทของการโจมตี | เครื่องมือที่ใช้และพารามิเตอร์ที่ระบุ |
|---|---|
| ปิงสวีป | Nmap -sP 161.246.5.0/24 |
| สแกนพอร์ต | Nmap -sT –sU 161.246.5.4 |
| ตรวจสอบระบบปฏิบัติการ | Nmap –O 161.246.5.4 |
| ส่งแพ็กเก็ตจำนวนมาก | Killwin 161.246.5.4 -t 1000000 |
| แพ็กเก็ตแบบวนลูป | Fawx2 161.246.5.4 161.246.5.4 1000 |
| แพ็กเก็ต SYN flood | Synflood 161.246.5.11 161.246.5.4 1000000 |
| แฟรกเมนต์ที่เหลื่อมล้ำกัน | Jolt 161.246.5.4 |
| แฟรกเมนต์ที่เกิดช่องว่าง | Opentear 161.246.5.4 |

*ตารางที่ 7-1 แสดงชื่อโปรแกรมบางส่วนที่ใช้ทดสอบการบุกรุกแบบต่างๆ*

จากการทดสอบได้ผลลัพธ์ดังนี้

| ประเภทการโจมตี | ผลการทดสอบ | อัตราส่วนแพ็กเก็ตสูญหาย |
|---|---|---|
| ปิงสวีป | ถูกต้อง | 0 เปอร์เ.ซนต์ |
| สแกนพอร์ต | ถูกต้อง | 0 เปอร์เ.ซนต์ |
| ตรวจสอบระบบปฏิบัติการ | ถูกต้อง | 0 เปอร์เ.ซนต์ |
| ส่งแพ็กเก็ตจำนวนมาก | ถูกต้อง | ประมาณ 30 เปอร์เ.ซนต์ |
| แพ็กเก็ตแบบวนลูป | ถูกต้องแต่คลาดเคลื่อน | 0 เปอร์เ.ซนต์ |
| แพ็กเก็ต SYN flood | ถูกต้องแต่คลาดเคลื่อน | ประมาณ 20 เปอร์เ.ซนต์ |
| แฟรกเมนต์ที่เหลื่อมล้ำกัน | ถูกต้อง | ประมาณ 50 เปอร์เ.ซนต์ |
| แฟรกเมนต์ที่เกิดช่องว่าง | ถูกต้อง | ประมาณ 50 เปอร์เ.ซนต์ |

*ตารางที่ 7-2 แสดงผลที่ได้จากการทดสอบการบุกรุกแบบต่างๆ*

**หมายเหตุ**

- สำหรับผลการทดสอบที่ระบุว่า "ถูกต้องแต่คลาดเคลื่อน" คือระบบสามารถตรวจจับประเภทของการโจมตีดังกล่าวได้อย่างถูกต้อง แต่ในการแสดงผลตรวจจับยังมีการแสดงการโจมตีประเภทอื่นออกมาด้วย
- อัตราส่วนแพ็กเก็ตสูญหาย คือปริมาณแพ็กเก็ตที่สูญหายไปเมื่อระบบตรวจจับผู้บุกรุกกำลังทำการประมวลผลการโจมตี เทียบกับปริมาณแพ็กเก็ตที่ระบบรับเข้ามาจากเครือข่าย สาเหตุเนื่องจากปริมาณแพ็กเก็ตในระบบมีมากเกินไป ส่งผลให้ตรวจจับผู้บุกรุกไม่สามารถประมวลผลได้ทัน

จากนั้นทำการทดสอบใหม่โดยเปลี่ยนเครื่องทดสอบที่ติดตั้งระบบตรวจจับผู้บุกรุก โดยมีรายละเอียดของคอมพิวเตอร์ที่ใช้ทดสอบดังนี้

- หน่วยประมวลผลความเร็ว 700 Mhz
- หน่วยความจำหลัก 384 MB
- ระบบปฏิบัติการไมโครซอฟท์วินโดว์ 2000 ที่ติดตั้งเซอร์วิสแพ็ก 1
- การ์ดแลนความเร็ว 100 Mb

ผลที่ได้คือ ความถูกต้องในการตรวจจับการบุกรุกมีค่าเท่าเดิม แต่ปริมาณของแพ็กเก็ตที่สูญหายลดลง โดยมีค่าลดลงประมาณ 50 เปอร์เซ็นต์ของค่าเดิม

**7.4 ปัญหาที่เกิดขึ้นขณะทดสอบ**

การทดสอบได้กระทำภายในเครือข่ายเดียวกัน ทำให้ปริมาณแพ็กเก็ตที่เกิดขึ้นมีมากในระยะเวลาอันสั้น ส่งผลให้เกิดความผิดพลาดและคลาดเคลื่อนในทดสอบ ปัญหาที่พบคือในขณะทำการโจมตีรูปแบบต่างๆ มักจะมีการแจ้งเตือนว่าเกิดการโจมตีแบบการส่งแพ็กเก็ตจำนวนมากด้วย ต้นเหตุเกิดจากในขณะที่ทำการทดสอบนั้น ได้เกิดแพ็กเก็ตที่มีปลายทางไปยังเป้าหมายจำนวนมาก ส่งผลให้ระบบตรวจจับทำการแจ้งว่ามีการโจมตีแบบการส่งแพ็กเก็ตจำนวนมากด้วย

**7.5 สรุปผลการทดสอบความสามารถในการตรวจจับ**

จากการทดลองปรากฏว่าโปรแกรมสามารถตรวจจับการบุกรุกทางเครือข่าย ทั้งจากการสำรวจและการโจมตีได้ตามขอบเขตที่ระบุไว้ คือ

- ตรวจสอบการปิงสวีปได้
- ตรวจสอบการสแกนพอร์ตได้ทั้งพอร์ตยูดีพีและทีซีพี
- ตรวจสอบการระบุระบบปฏิบัติการได้ในการตรวจสอบแบบแอคตีฟ
- ตรวจสอบการโจมตีแบบส่งแพ็กเก็ตจำนวนมากได้
- ตรวจสอบการโจมรแบบ SYN flood ได้

- ตรวจสอบการโจมตีแบบส่งแพ็กเก็ตที่มีหมายเลขไอพีต้นทางและปลายทางเหมือนกันได้
- ตรวจสอบการโจมตีจากการีแอสเซมเบิลไม่สมบูรณ์เมื่อแพ็กเก็ตเกิดช่องว่างขณะประกอบได้
- ตรวจสอบการโจมตีจากการีแอสเซมเบิลไม่สมบูรณ์เมื่อแพ็กเก็ตทับซ้อนกันขณะประกอบได้

# บทที่ 8
# สรุปผลและวิจารณ์

## 8.1 ปัญหาและอุปสรรค

ในการพัฒนาระบบตรวจจับผู้บุกรุกทางระบบเครือข่ายคอมพิวเตอร์นี้ มีปัญหาและอุปสรรคในการพัฒนาหลายประการ ได้แก่

(1) โปรแกรมที่ใช้ในการบุกรุกมีอยู่มากมาย การศึกษาการทำงานของโปรแกรมเหล่านั้นต้องใช้เวลานาน ทำให้ยังไม่สามารถตรวจจับการบุกรุกได้ครอบคลุมทุกรูปแบบการบุกรุก

(2) ยังไม่สามารถตรวจจับการโจมตีในชั้นแอพพลิเคชันได้ เนื่องจากการเข้าถึงข้อมูลในชั้นดังกล่าวจะทำให้ประสิทธิภาพต่ำลงมาก และการโจมตีในชั้นนี้ยังมีอีกมากมาย ขึ้นอยู่กับประเภทของแอพพลิเคชันที่ใช้งาน

(3) โปรแกรมยังทำงานได้ช้าอยู่ เนื่องจากการพัฒนาที่ได้ใช้ไมโครซอฟท์ฟลาวเดชันคลาส ซึ่งมีความเร็วในการทำงานต่ำกว่าการเขียนโปรแกรมโดยใช้ Win32 API แต่มีความง่ายและพัฒนาได้เร็วกว่า

(4) หากเป็นการโจมตีภายในเครือข่ายเดียวกัน จะทำให้เกิดแพ็กเก็ตในเครือข่ายเป็นจำนวนมากในเวลาอันรวดเร็ว ระบบตรวจจับการบุกรุกอาจทำการวิเคราะห์ได้เร็วไม่พอ ส่งผลให้ข้อมูลการโจมตีอาจมีไม่ครบ

(5) ระบบยังต้องทำงานผ่านไลบรารี WinpCap อยู่ หากไลบรารีดังกล่าวหยุดการพัฒนา อาจส่งผลให้ระบบตรวจจับไปไม่สามารถรองรับการสื่อสารใหม่ๆ ในอนาคตได้ แต่อุปสรรคข้อนี้กลับกลายเป็นข้อดีประการหนึ่งคือ หากไลบรารีดังกล่าวมีการพัฒนาอย่างต่อเนื่อง ย่อมส่งผลให้ความสามารถในการตรวจจับของโปรแกรมดีขึ้น

## 8.2 แนวทางการพัฒนาต่อในอนาคต

(1) เนื่องจากระบบตรวจจับการโจมตีสามารถทำงานได้เฉพาะบรอดแคสโดเมนเขตของตัวเองเท่านั้น ในเครือข่ายขนาดใหญ่จึงต้องใช้ระบบตรวจจับผู้บุกรุกหลายตัว ทำให้การดูแลและสังเกตการณ์ทำได้ลำบาง จึงควรพัฒนาการตรวจจับให้เป็นแบบกระจาย (Distrubute) โดยตัวตรวจจับจะกลายเป็นเอเจนต์เล็กๆ ที่นำไปติดตั้งยังตำแหน่งที่สำคัญของเครือข่าย การควบคุมและแสดงผลจะทำผ่านตัวควบคุมหลักที่ติดต่ออยู่กับเอเจนต์ เมื่อมีการบุกรุก เอเจนต์จะส่งข้อมูลการบุกรุกมายังส่วนควบคุม

(2) พัฒนาให้สามาถรติดต่อกับเราเตอร์เพื่อให้สามารถป้องกันการโจมตีจากภายนอกได้ โดยระบบจะทำการแก้ไขค่าคอนฟิกกูเรชันของเราเตอร์โดยอัตโนมัติ เพื่อทำการควบคุมแพ็กก็ตที่ต้องการโจมตีที่ส่งมาจากภายนอก

(3) พัฒนาส่วนควบคุมการ์ดอีเธอร์เน็ตขึ้นเอง เพื่อให้สามารถควบคุมแพ็กเก็ตที่ต้องการได้อย่างถูกต้อง และควบคุมการใช้งานบัฟเฟอร์ของแพ็กเก็ตได้อย่างมีประสิทธิภาพ จะทำให้ลดปริมาณแพ็กเก็ตที่สูญหายได้

**8.3 ข้อเสนอแนะการใช้งาน**

(1) เนื่องจากในขณะทำงานระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ ระบบต้องการจะใช้ทรัพยากรในการเก็บข้อมูลและประมวลผลค่อนข้างมาก เครื่องที่ติดตั้งระบบตรวจจับดังกล่าวจึงควรมีประสิทธิภาพสูงพอสมควร

(2) เพื่อเสถียรภาพในด้านความปลอดภัยสูงสุด เครื่องที่ใช้งานเป็นระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ ควรติดตั้งการ์ดอีเธอร์เน็ตแต่ไม่มีการระบุหมายเลขไอพีแอดเดรสให้กับการ์ดดังกล่าว เนื่องจากระบบตรวจจับผู้บุกรุกสามารถทำงานได้กับการ์ดอีเธอร์เน็ตที่ไม่มีการระบุหมายเลขไอพี จึงไม่ควรให้เครื่องดังกล่าวสามารถติดต่อผ่านทางเครือข่ายได้ เป็นการป้องกันไม่ให้ผู้โจมตีทำการโจมตีระบบตรวจจับผู้บุกรุกโดยตรง

# บรรณานุกรม

## หนังสืออ้างอิง

[1] Joel Scambray, Stuart McClure, George Kurthz, *"Hacking Exposed"*, Mc Graw Hill, 2001

[2] David White, Kenn Scribner, Eugene Olafsen, *"MFC Programming with Visual C++6.0 Unleashed"*, Sams, 1999

[3] Stephen Northcutt, Judy Novak, *"Network Intrusion Detection An Analyst's Handbook"*, New Riders, 2000

[4] สุวัฒน์ ปุณณชัยยะ, ต้น ตันฑ์สุทธิวงศ์, สุพจน์ ปุณณชัยชนะ, *"เปิดโลกของ TCP/IP และโปรโตคอลของอินเตอร์เน็ต"*, โปรวิชั่น, 2543

[5] ยุทธนา ลีลาสวัฒนกุล, *"คู่มือการเขียนโปรแกรมและใช้งาน Visual C++6.0"*, อินโฟเพรส, 2544

## เว็บไซต์อ้างอิง

[6] http://www.cert.org

[7] http://www.codeguru.com

[8] http://www.nmap.org

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

[10] http://www.thaidev.com