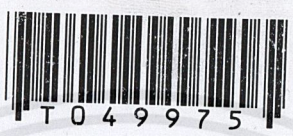
ไฟร์วอลแบบกระจายพร้อมระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

Distributed Firewall with Network Intrusion Detection System



นายวงศ์รพี แก้วญาณะ
นายวันเฉลิม สินสวัสดิ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2545

ไฟร์วอลแบบกระจายพร้อมระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

Distributed Firewall with Network Intrusion Detection System

โดย

นายวงศ์รพี แก้วญานะ 43015381

นายวันเฉลิม สินสวัสดิ์ 43015382

อาจารย์ที่ปรึกษา

อาจารย์ธนา หงษ์สุวรรณ

อาจารย์อัครเดช วัชระภูพงษ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

ปริญญานิพนธ์ปีการศึกษา 2545

ภาควิชา วิศวกรรมศาสตร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ไฟร์วอลแบบกระจายพร้อมระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

Distributed Firewall with Network Intrusion Detection System

ผู้จัดทำ

1. นาย วงศ์รพี แก้วญาณะ รหัสประจำตัว 43015381
2. นาย วันเฉลิม สินสวัสดิ์ รหัสประจำตัว 43015382

_____ อาจารย์ที่ปรึกษา

(อาจารย์ธนา หงษ์สุวรรณ)

_____ อาจารย์ที่ปรึกษา

(อาจารย์อัครเดช วัชระภูพงษ์)

ไฟร์วอลแบบกระจายพร้อมระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

นาย วงศ์รพี แก้วญานะ 43015381
นาย วันเฉลิม สินสวัสดิ์ 43015382
อาจารย์ ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษา
อาจารย์ อัครเดช วัชระภูพงษ์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2545

บทคัดย่อ

ในปัจจุบันการรักษาความปลอดภัยในระบบเครือข่ายคอมพิวเตอร์มีความซับซ้อนมากขึ้น ดังนั้นจึงเป็นการยากที่จะดูแลควบคุมระบบไฟร์วอล(Firewall) ให้มีประสิทธิภาพและทำงานได้ถูกต้องทั้งหมด ยิ่งถ้าไฟร์วอล (Firewall) ในระบบเครือข่ายมีมากกว่า 1 จุด รวมถึงเพอร์ซันนอลไฟร์วอล (Personal Firewall) ด้วยแล้ว ยิ่งทำให้ยากลำบากต่อการควบคุมทั้งหมด

ดังนั้นจึงได้พัฒนาระบบไฟร์วอลที่มีความสามารถในการจัดการเกี่ยวกับ Policy ต่างๆ และทำการวิเคราะห์เพื่อนำมาควบคุมเพอร์ซันนอลไฟร์วอลทุกตัว ที่ติดตั้งอยู่ภายในเครือข่ายนั้นๆ ได้ ด้วยการควบคุมจาก Firewall Administration เพียงที่เดียว และหากต้องการที่จะเพิ่มประสิทธิภาพของระบบความปลอดภัยในเครือข่ายให้สูงขึ้นด้วยแล้วจึงควรมีระบบที่มีความสามารถในการตรวจสอบ ลักษณะการทำงานต่างๆ อันจะเป็นสาเหตุนำมาซึ่งความเสียหายแก่เครือข่ายไม่ว่าทางใดก็ตาม ดังนั้นจึงได้ผนวกระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ เข้าไปไว้ในระบบไฟร์วอลนี้ด้วย

Distributed Firewall with Network Intrusion Detection System

Mr. Wongrapee Keawyana
Mr. Wanchalerm Sinsawasd
Mr. Thana Hongsuwan Advisor
Mr. Akkradach Watcharapupong Advisor

ABSTRACT

Nowadays, the network security is getting more complicated so it is hard to control the Firewall system to be effective and work properly even if we have more than 1 Firewall in the network system including Personal Firewall, then it is even more difficult to control.

Therefore, we have developed Firewall which can provide the management of any policy and analyzing for controlling all of the Personal Firewall that be installed in the network system by only controlling for Firewall Administrator.

Also we have added the Networks Intrusion Detection System into the Firewall system for increase the ability of the security system.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดี เนื่องจากได้รับการดูแล คำแนะนำต่างๆ การสนับสนุนและให้คำปรึกษาเป็นอย่างดี จากคำแนะนำของอาจารย์ธนา หงษ์สุวรรณ และอีกท่านหนึ่งคือ อาจารย์อัครเดช วัชระภูพงษ์ ซึ่งเป็นอาจารย์ที่ปรึกษา ซึ่งต้องขอขอบพระคุณเป็นอย่างสูง รวมทั้งอาจารย์ทุกท่านในภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้ให้การอบรมสั่งสอน และได้ให้วิชาความรู้ต่างๆ ที่ดีแก่คณะผู้จัดทำเสมอมา

และสำหรับบุคคลสองท่านที่สำคัญที่สุดในชีวิตนี้ ที่ได้ให้กำเนิด ให้การอบรมดูแลและเอาใจใส่ทั้งด้านการศึกษา ด้านการดำเนินชีวิต และด้านอื่นทุกๆ ด้าน ที่คงไม่อาจมีใครอีกแล้ว ที่เสมอเหมือนท่านทั้งสองนี้ นั่นคือ บิดา และ มารดา ผู้ซึ่งเป็นที่เคารพรักอย่างยิ่ง ผู้ซึ่งคอยให้กำลังใจในยามที่ไม่มีใครเหลือ ผู้ซึ่งคอยชี้นำทางที่ถูกเสมอ ผู้ที่ให้การสนับสนุนในการทำสิ่งที่ถูก และให้คำชี้นำหากสิ่งที่เราทำไม่ใช่สิ่งที่ดีไม่ใช่สิ่งที่ควรทำ จึงขอกราบขอบพระคุณมา ณ ที่นี้

สุดท้ายนี้ขอขอบคุณภาควิชาวิศวกรรมคอมพิวเตอร์ และ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่อำนวยความสะดวกในด้านการให้สถานที่การทำงาน และเครื่องมือกับอุปกรณ์ใช้งานต่างๆ ทุกชนิด รวมทั้ง การอำนวยความสะดวกในด้านการใช้งานเครือข่ายของภาควิชาฯ และ สิ่งสำคัญที่ลืมไม่ได้อีกอย่างคือ ขอบคุณมิตรภาพ และ ความเป็นห่วง ของเพื่อนๆ รวมทั้งพี่น้องทุกๆ คน ที่มีให้กันเสมอมา ที่คอยให้คำปรึกษา ให้กำลังใจ ให้ความรัก ให้ความดูแล เป็นอย่างดี

วงศ์รพี แก้วญานะ
วันเฉลิม สินสวัสดิ์

สารบัญ

หน้าที่

สารบัญภาพประกอบ

หน้าที่

สารบัญตาราง

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

ระบบความปลอดภัยในระบบเครือข่ายคอมพิวเตอร์ในปัจจุบันนี้ มีความจำเป็นและสำคัญอย่างยิ่งเพราะองค์กรต่างๆ ทุกองค์กรย่อมมีข้อมูลที่สำคัญ ที่จะอนุญาตให้กับเฉพาะบุคคลที่เกี่ยวข้องเท่า นั้นสามารถเข้าถึงข้อมูลนั้นได้ แต่ถ้าหากมีบุคคลอื่นที่ไม่หวังดีสามารถเข้าถึงข้อมูลนั้นๆ ได้ หรือ สามารถทำให้การให้บริการข้อมูลเหล่านั้นถูกปิดบริการไป ทำให้อาจจะเกิดความเสียหายเป็นอย่างมากต่อองค์กรนั้นๆ ได้

แต่ถ้าหากระบบความปลอดภัยในเครือข่ายนั้น เคร่งครัดมาก เช่น ติดตั้งไฟร์วอล (Firewall) เข้าไปในระบบเครือข่ายมาก และหลายๆ ชั้น ก็จะทำให้ระบบความปลอดภัยที่ว่านี้มีความซับซ้อนมากขึ้น ดังนั้นจึงเป็นการยากที่จะดูแลควบคุมระบบไฟร์วอล ให้มีประสิทธิภาพและทำงานได้ถูกต้องทั้งหมด ยิ่งถ้าไฟร์วอล (Firewall) ในระบบเครือข่ายมีมากกว่า 1 จุดดังที่กล่าวมาแล้ว อีกทั้งรวมถึงเพอร์ซันนอลไฟร์วอล (Personal Firewall) ด้วยแล้ว ยิ่งทำให้ยากลำบากต่อการควบคุมทั้งหมด

ดังนั้นจึงได้พัฒนาระบบไฟร์วอลที่มีความสามารถในการจัดการเกี่ยวกับ Policy ต่างๆ และทำการวิเคราะห์เพื่อนำมาควบคุมเพอร์ซันนอลไฟร์วอลทุกตัว ที่ติดตั้งอยู่ภายในเครือข่ายนั้นๆ ได้ ด้วยการควบคุมจาก Firewall Administration เพียงที่เดียว

และหากต้องการที่จะเพิ่มประสิทธิภาพของระบบความปลอดภัยในเครือข่ายให้สูงขึ้นด้วยแล้วจึงควรมีระบบที่มีความสามารถในการตรวจสอบ ลักษณะการทำงานต่างๆ อันจะเป็นสาเหตุนำมาซึ่งความเสียหายแก่เครือข่ายไม่ว่าทางใดก็ตาม ดังนั้นจึงได้ผนวกระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ หรือ Networks Intrusion Detection System เข้าไปไว้ในระบบไฟร์วอลนี้ด้วย

ระบบไฟร์วอลแบบกระจายพร้อมระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ ที่ได้สร้างขึ้นมานี้ จะมีความสามารถในการควบคุมไฟร์วอลได้ทุกจุดที่เชื่อมต่ออยู่ในเครือข่าย ทั้งส่วนที่เป็นเกตเวย์ไฟร์วอล และส่วนที่เป็นเพอร์ซันนอลไฟร์วอลของแต่ไคลเอนต์ และยังเพิ่มประสิทธิภาพให้ปลอดภัยยิ่งขึ้นด้วยระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ที่สามารถรายงานผลการบุกรุกกลับไปยังเซิร์ฟเวอร์ เพื่อวิเคราะห์และดำเนินการป้องกันต่อไป

1.2 วัตถุประสงค์ของโครงงาน

โครงงานที่ได้สร้างขึ้นมานี้มีวัตถุประสงค์ดังต่อไปนี้

- เพื่อศึกษาโครงสร้างและลักษณะการทำงานของไฟร์วอล
- เพื่อศึกษาหลักการทำงานของระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์
- เพื่อศึกษาการเขียนโปรแกรมในการติดต่อกันผ่านทางเครือข่าย
- เพื่อศึกษาการเขียนโปรแกรมบนระบบปฏิบัติการลีนุกซ์และไมโครซอฟท์วินโดวส์

1.3 ขอบเขตของโครงงาน

การพัฒนาโครงการนี้ระบบส่วนใหญ่หลักๆ จะมีด้วยกันสองส่วนคือ

ส่วนแรกคือส่วนที่เป็นไฟร์วอลแอดมินิสเตชัน และส่วนที่เป็น เกต์เวย์ไฟร์วอล ซึ่งทั้งสองส่วนนี้ทำงานอยู่บนระบบปฏิบัติการลีนุกซ์ ไฟร์วอลแอดมินิสเตชันจะทำหน้าที่เป็นเซิร์ฟเวอร์ ซึ่งส่วนนี้จะมีเป็นส่วนที่ทำหน้าที่หลักในการกำหนดกฎ(Rule) ให้กับไฟร์วอลทั้งหมด ส่วนเกตเวย์ไฟร์วอล ก็คือส่วนไฟร์วอลที่ติดตั้งอยู่ตรงส่วนที่เป็นเกตเวย์ของเครือข่าย จะทำงานร่วมกับ IPTABLES

ส่วนที่สองจะทำงานบนระบบปฏิบัติการวินโดวส์ คือส่วนที่เป็นเพอร์ซันนอลไฟร์วอลกับระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ ซึ่งทั้งสองส่วนนี้จะต้องทำงานทั้งการตรวจสอบวิเคราะห์การโจมตีและการป้องกันต่างๆ ร่วมกันได้ ซึ่งในส่วนนี้จะมีเอเจนต์ในการแจ้งเตือน หากมีการบุกรุกใดๆ เกิดขึ้น เอเจนต์จะส่งการแจ้งเตือนไปยังไฟร์วอลแอดมินิสเตชัน ในส่วนของเพอร์ซันนอลไฟร์วอล ก็จะรอรับกฎที่ถูกส่งมาจากเซิร์ฟเวอร์ พร้อมทั้งเพิ่มกฎที่ได้รับมานี้แล้วทำงานตามกฎได้ทันที อีกทั้งหากไม่สามารถทำการติดต่อกับเซิร์ฟเวอร์ได้ ตัวเพอร์ซันนอลไฟร์วอลนี้ก็ยังคงสามารถทำงานได้ตามปกติ

1.4 ขั้นตอนการดำเนินงาน

1) ศึกษารายละเอียดเกี่ยวกับทีซีพี/ไอพี
2) ศึกษาเกี่ยวกับระบบไฟร์วอลทั้งบนระบบปฏิบัติการลีนุกซ์และระบบปฏิบัติการวินโดวส์
3) ศึกษาเกี่ยวกับระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์
4) ศึกษาการเขียนโปรแกรมผ่านเครือข่าย
5) ออกแบบโครงสร้างและขั้นตอนการทำงานต่างๆ ของระบบไฟร์วอลแบบกระจายพร้อมระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์
6) พัฒนาระบบไฟร์วอลแบบกระจายพร้อมระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ ทดสอบและปรับปรุงระบบไฟร์วอลแบบกระจายพร้อมระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

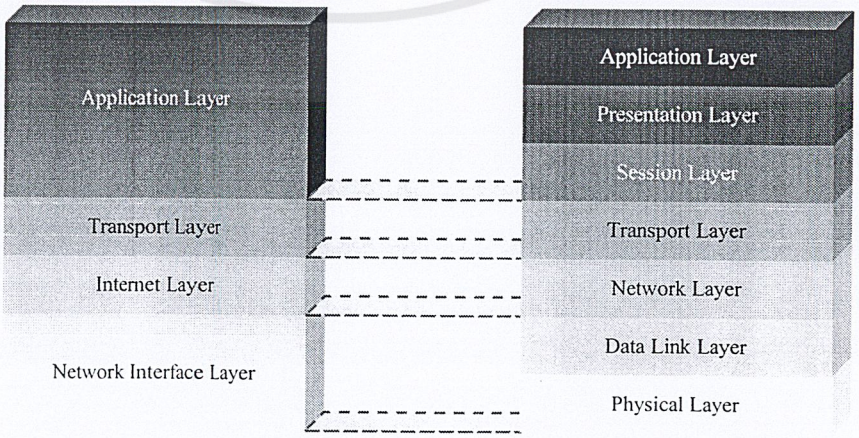
บทที่ 2
โพรโตคอลทีซีพี/ไอพี

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

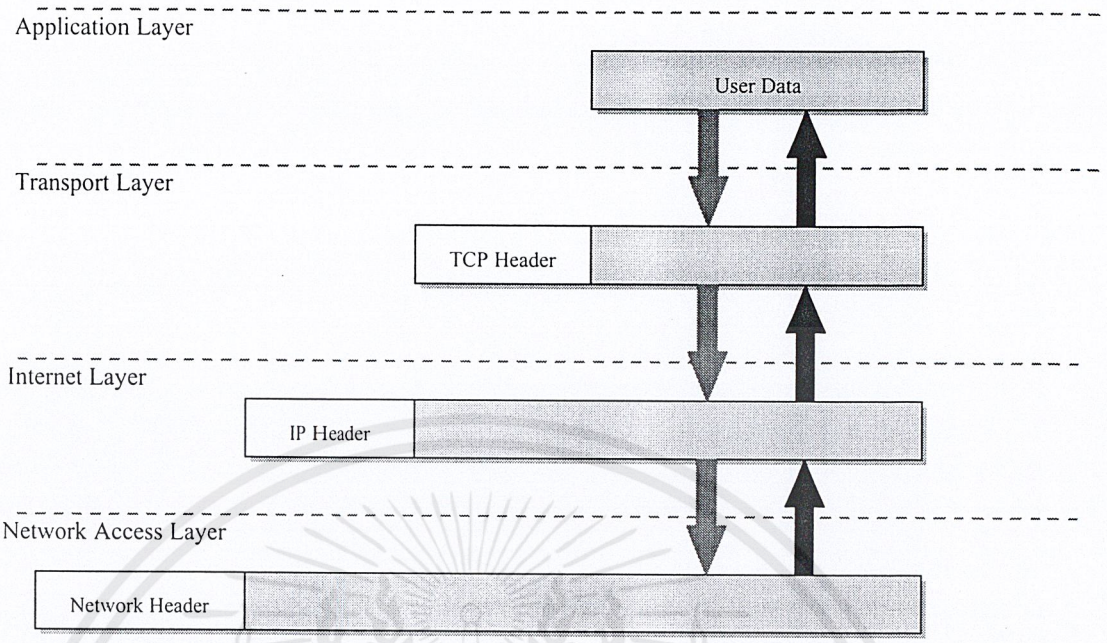


รูปที่ 2-1 แสดงการเปรียบเทียบเลเยอร์ของโอเอสไอกับเลเยอร์ของทีซีพี/ไอพี

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

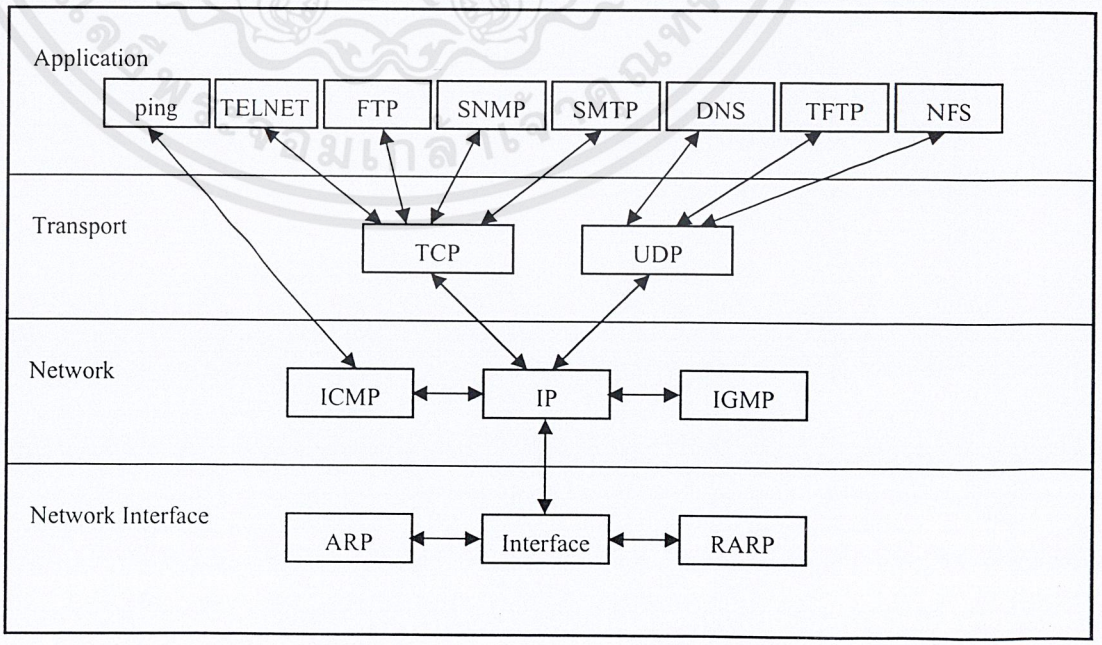
ตารางที่ 2-1 การทำงานของแต่ละระดับชั้นของทีซีพี/ไอพี



รูปที่ 2-2 แสดงการข้อมูลที่ส่งผ่านในโมเดลของทีซีพี/ไอพี

2.3 โพรโตคอลสแตก

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



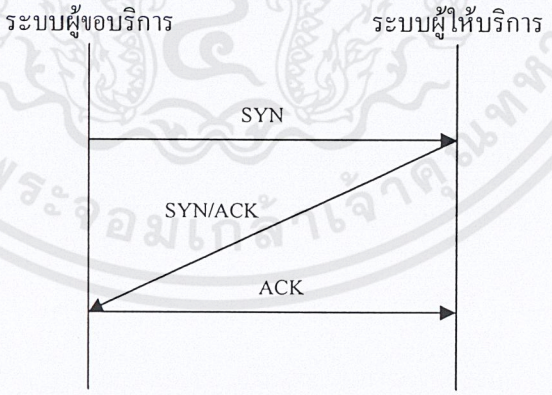
รูปที่ 2-3 โพรโตคอลสแตกของทีซีพี/ไอพี

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

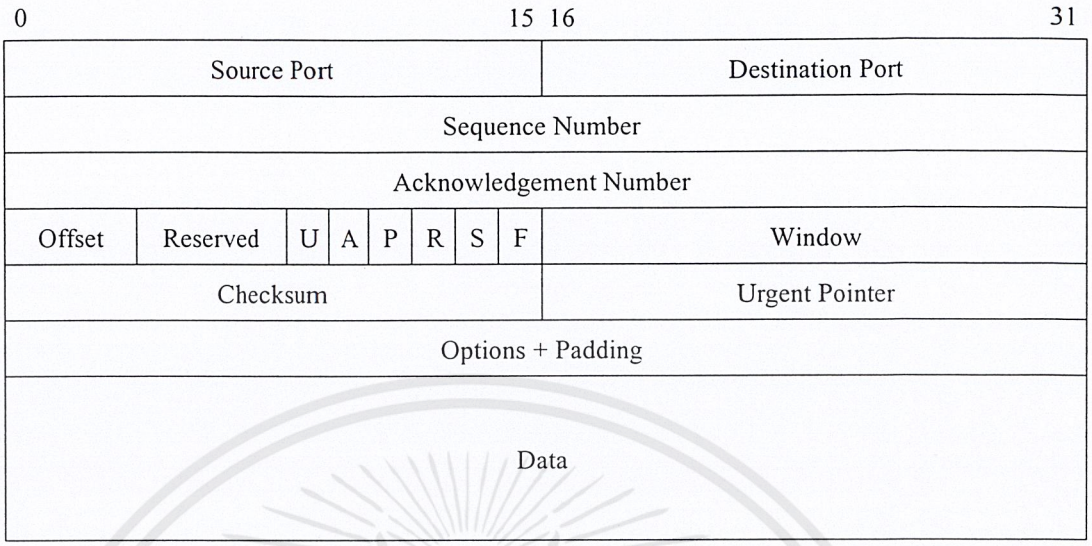
การเชื่อมต่อแบบ 3-way handshake นี้ เป็นการตรวจสอบความพร้อมของทั้งฝ่ายส่งและฝ่ายรับ และการกำหนดค่าเริ่มต้นของพารามิเตอร์ต่างๆ ของทั้งสองฝ่ายให้ตรงกัน หลังจากกระบวนการทำ 3-way handshake สิ้นสุด ทั้งสองฝ่ายจึงสามารถรับและส่งข้อมูลซึ่งกันและกันได้

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

ส่วนประกอบของ UDP Frame

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

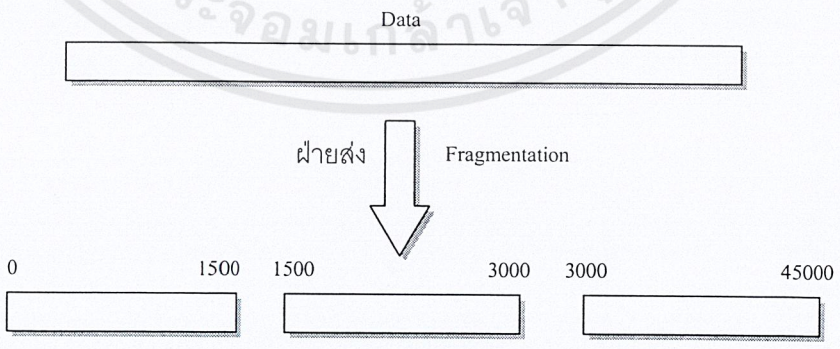
0	15 16	31
Source Port		Destination Port
Length		Checksum
Data		

รูปที่ 2-6 แสดงแพ็กเก็ตยูดีพี

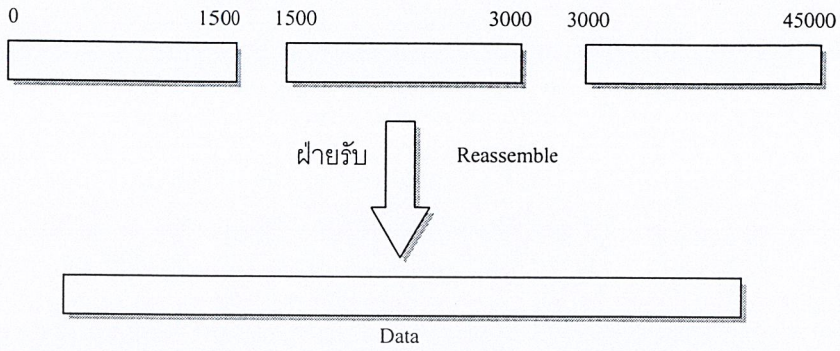
2.6 โพรโตคอลไอพี (IP: Internet Protocol)

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

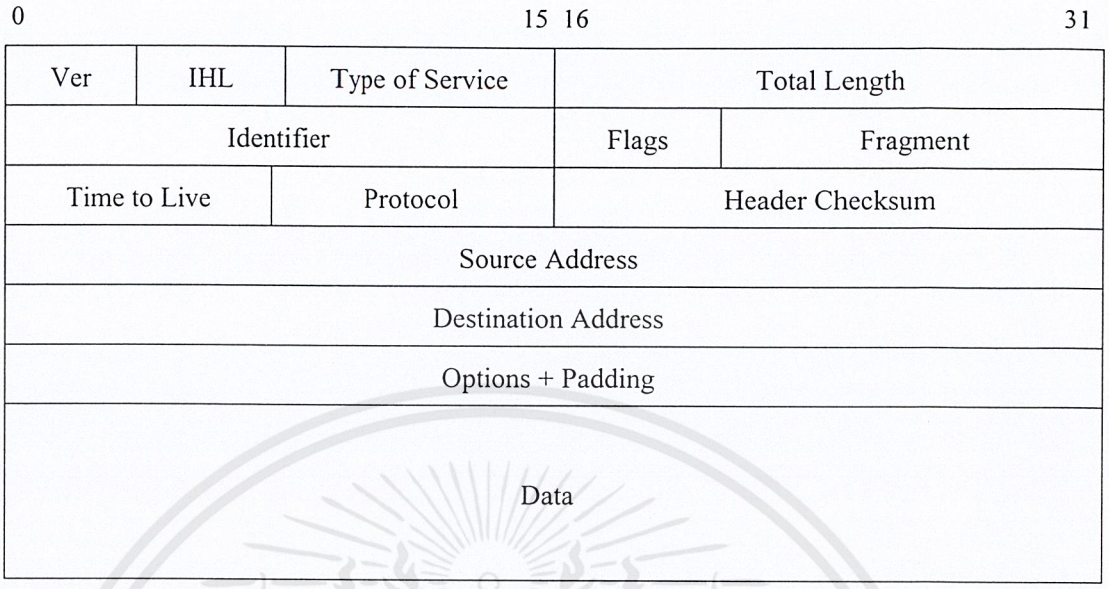
Bit 1 : 0 = บอกว่าแพ็กเก็ตมีการแตกแพ็กเก็ตย่อย

1 = บอกว่าแพ็กเก็ตไม่มีการแตกแพ็กเก็ตย่อย

Bit 2 : 0 = บอกว่าแพ็กเก็ตนั้นเป็นแพ็กเก็ตสุดท้ายที่ได้จากการแตกแพ็กเก็ตย่อย

1 = บอกว่าแพ็กเก็ตนั้นยังไม่ใช่แพ็กเก็ตสุดท้ายที่ได้จากการแตกแพ็กเก็ตย่อย

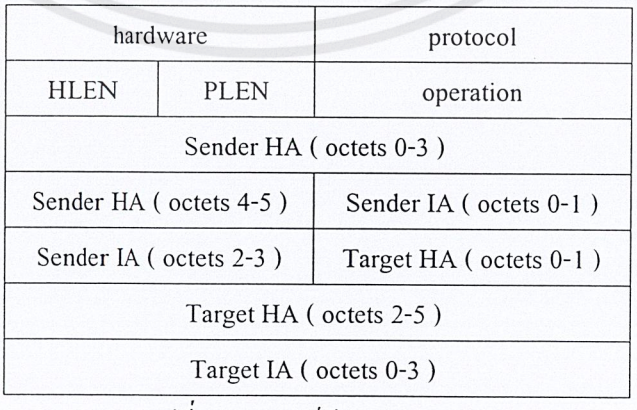
7. *Fragment Offset* : เป็นค่าตัวเลข 13 บิต บอกออปเซตของแฟร็กเมนต์เมื่อเทียบในดาต้าแกรม
8. *Time To Live (TTL)* : เป็นตัวเลข 8 บิต บอกช่วงเวลาของแพ็กเก็ตที่ยังอยู่ในเครือข่ายได้ โดยกำหนดค่าเป็นจำนวนเราท์เตอร์สูงสุดที่ดาต้าแกรมผ่านได้ ซึ่งโดยทั่วไปทีค่าระหว่าง 32 ถึง 64 และลดค่าลงเรื่อยๆ เมื่อผ่านเราเตอร์ 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



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



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)

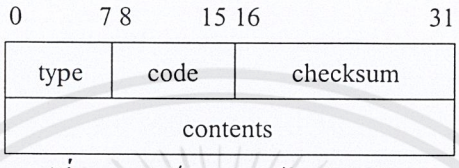
6. *Address* : ฟิลด์แอดเดรสเรียงลำดับจากฮาร์ดแวร์และเน็ตเวอร์คแอดเดรสของสถานีที่ร้องขอ ตามด้วยฮาร์ดแวร์และเน็ตเวอร์คแอดเดรสของสถานีที่ตอบรับ

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 ห รือขณะนั้นเครื่องปลายทางเกิดปัญหาจนไม่สามารถรับข้อมูลได้ ที่ เราท์เตอร์จะส่งข้อความแจ้งเป็น ไอซีเอ็มพี message ที่ชื่อ destination unreachable ให้กับผู้ส่งข้อมูล นอกจากนี้ตัวข้อมูลที่แจ้ง

ข้อความก็จะมีส่วนของข้อมูลไอพีดาทาแกรมที่เกิดปัญหาด้วย 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 type และ code

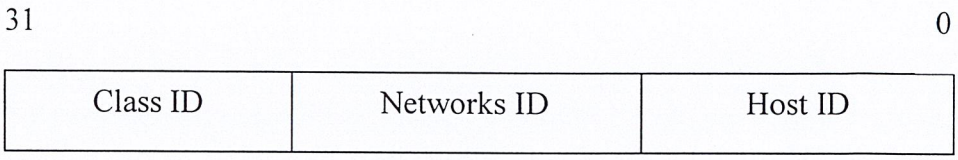
2.9 รูปแบบและการกำหนดแอดเดรสของ ทีซีพี/ไอพี

ในหัวข้อนี้จะได้อธิบายรูปแบบการกำหนดแอดเดรสของโพรโตคอลทีซีพี/ไอพี ซึ่งลักษณะแอดเดรสของโพรโตคอล นี้ค่าของแอดเดรสของเครื่องในระบบเครือข่ายจะไม่ซ้ำกันเลย โดยเรียกเลขนี้ว่าหมายเลขไอพี เป็นเลข 32 บิตซึ่งแบ่งเป็นคลาส ตามหลักในการพิจารณาที่จะได้กล่าวต่อไปนี้

การแบ่งคลาสเน็ตเวิร์ก

เนื่องจากหมายเลขแอดเดรสของคอมพิวเตอร์เครื่องใดๆนั้น จะต้องสามารถบอกถึงความแตกต่างระหว่างตัวเรื่องเอง ตลอดจนเครือข่ายที่คอมพิวเตอร์นั้นเชื่อมต่ออยู่ด้วย หมายเลขไอพีจึงแบ่งแยกออกเป็น 2 ส่วน ได้แก่ ส่วนที่แสดงหมายเลขของโฮสต์ และส่วนที่เป็นหมายเลขของเครือข่าย

การแบ่งคลาสของแอดเดรสทำได้โดยพิจารณาจำนวนบิตของ 2 ส่วนประกอบข้างต้น ซึ่งมีการแบ่งออกเป็น 5 คลาส แต่มีการใช้เพียง 3 คลาสแรก คือ คลาส A, คลาส B และ คลาส C ส่วนคลาส D



IP Address Format

และ คลาส E ถูกสงวนไว้สำหรับจุดประสงค์พิเศษ

Network Class	Networks	Hosts per Network
A	124	16,777,214
B	16,382	65,534
C	2,097,150	254

รูปที่ 2-12 แสดงคลาส, จำนวนเครือข่าย และจำนวนโฮสต์ของแต่ละคลาส

การทำซับน็ต (Subnetting)

การทำซับเน็ตเป็นการเปลี่ยนแปลงการใช้หมายเลขของเครื่องโฮสต์และหมายเลขของเครือข่ายในระดับท้องถิ่น โดยในทางตรรก คือ การเลื่อนเส้นแบ่งที่แยกหมายเลขเครื่อง และหมายเลขของเครือข่ายที่อยู่ในหมายเลขไอพีโดยที่ปริมาณของหมายเลขโฮสต์และหมายเลขเครือข่ายจะแปรผกผันกัน ยกตัวอย่าง เช่น หากมีปริมาณของเครือข่ายมาก ก็จะทำให้เครื่องใดๆที่จะต่อกับระบบเครือข่ายหนึ่งๆน้อยลง เป็นต้น ในการปฏิบัติการทำซับเน็ตทำโดยการทำซับเน็ตมาสก์ (Subnet Mask) คือตัวเลขจำนวน 32 บิต มาทำการ AND กับหมายเลขไอพีตัวอย่างเช่นกำหนดหมายเลขไอพีเป็น 161.246.5.24 และซับเน็ตมาสก์คือ 255.255.255.0 จะได้เป็น 161.246.5.0

จะเห็นว่าหากพิจารณาโดยไม่มีการทำซับเน็ตแล้วจะได้หมายเลขของเน็ตเวิร์คคือ 161.246 และหมายเลยประจำเครื่องคือ 5.24 แต่ผลที่ได้จากการทำซับเน็ตจะได้หมายเลขเน็ตเวิร์คเป็น 161.2476.5.0 และหมายเลขเครื่องคือ 24 หรืออาจกล่าวได้ว่า คอมพิวเตอร์เครื่องนี้มีหมายเลขเครื่องเท่ากับ 24 และอยู่บนเครือข่ายย่อยหมายเลข 161.246.5

2.10 หมายเลขพอร์ต

เนื่องจากในเวลาใดๆ สามารถมีโพรเซสของผู้ใช้สามารถใช้ยูดีพีหรือทีซีพีได้พร้อมๆ กันหลายๆ โพรเซส ดังนั้นจึงต้องมีวิธีแยกแยะว่าข้อมูลเป็นของโพรเซสใดซึ่งวิธีที่ทีซีพีและยูดีพีใช้ คือการใช้หมายเลขพอร์ต

เมื่อโพรเซสของเครื่องไคลเอนต์ต้องการที่จะติดต่อกับเซิร์ฟเวอร์ ไคลเอนต์จะต้องติดต่อแต่ลำพังรู้หมายเลขอินเตอร์เน็ต 32 บิต เพียงอย่างเดียวนั้นไม่เพียงพอ เพราะว่าสามารถติต่อกับโฮสต์ได้เพียง

อย่างเดียวแต่ไม่สามารถเจาะจงโพรเซสที่จะทำการติดต่อได้ ดังนั้นเพื่อแก้ปัญหานี้ทั้งทีซีพีและยูดีพี ได้มีการกำหนดหมายเลขพอร์ตมาตรฐาน (well-known ports) ซึ่งเป็นที่รู้จักกัน เช่น ทุกๆระบบทีซีพี/ไอพี จะกำหนดพอร์ต 23 เป็นพอร์ตบริการของเทลเน็ต เป็นต้น

เมื่อทีซีพีหรือยูดีพี กำหนดหมายเลขพอร์ตที่ไม่ซ้ำกันให้โพรเซสของผู้ใช้ เราเรียกหมายเลขพอร์ตนี้ว่าหมายเลขพอร์ตชั่วคราว (ephemeral port numbers) เมื่อไคลเอนต์เลิกใช้หมายเลขพอร์ตนี้แล้ว สามารถกำหนดหมายเลขพอร์ตนี้ให้ไคลเอนต์อื่นได้ โพรเซสที่ได้รับหมายเลขพอร์ตชั่วคราวนี้จะไม่สนใจว่ามีค่าเท่าไร แต่เป็นหน้าที่ของอีกโพรเซสหนึ่งที่ต่อกันที่ต้องสนใจ เพราะต้องส่งข้อมูลกลับมาที่พอร์ตนี้ในทีซีพี และ ยูดีพี นั้น หมายเลขพอร์ตตั้งแต่ 1-1023 เป็นพอร์ตที่สงวนไว้สำหรับหมายเลขพอร์ตมาตรฐาน

บทที่ 3
ไฟร์วอล

3.1 ประเภทของไฟร์วอล

3.1.1 เกตเวย์ไฟร์วอล (Gateway Firewall) หรือไฟร์วอล (Firewall)

ไฟร์วอลเป็นระบบหรือกลุ่มของระบบป้องกันที่บังคับใช้นโยบายการรักษาความปลอดภัยระหว่างเครือข่ายกับเครือข่าย หรือเครือข่ายกับอินเทอร์เน็ต ไฟร์วอลเป็นตัวกำหนดว่าบริการของเครือข่ายภายในชนิดใดบ้างที่เข้าถึงได้จากภายนอก และบริการภายนอกใดที่เข้าถึงได้จากผู้ใช้ภายใน ไฟร์วอลสามารถที่จะป้องกันการโจมตีจากภายนอกเครือข่ายได้ โดยจะกรองข้อมูล (หมายเลขไอพี สับเน็ต พอร์ต ฯลฯ) และอนุญาตให้ผู้ใช้ที่มีข้อมูลที่น่าไว้ใจเท่านั้น ผ่านเข้ามาในระบบเครือข่ายของเรา

วิธีการของไฟร์วอล จะจำกัดให้มีการผ่านเข้าออกได้ ที่จุดเดียว (Controlled point) และป้องกันผู้บุกรุกที่พยายามจะเข้ามาในเครือข่าย ดังนั้นการรับส่งข้อมูลทั้งหมดต้องผ่านไฟร์วอลซึ่งเป็นด่านสำหรับตรวจสอบแพ็กเก็ต[1] ไฟร์วอลมีหน้าที่ตัดสินใจว่าจะอนุญาตให้แพ็กเก็ตนั้นผ่านไปหรือไม่ ซึ่งการตัดสินใจนี้ ขึ้นอยู่กับกฎเกณฑ์ที่กำหนดไว้ สอดคล้องกับนโยบายขององค์การ

<u>สิ่งที่ไฟร์วอลสามารถทำได้</u>

- เป็นจุดสำคัญของการตัดสินใจเพื่อรักษาความปลอดภัย เนื่องจากเป็นจุดเดียวที่เครือข่ายติดต่อกับเครื่องภายนอกเครือข่าย
- เป็นจุดสำคัญของการตัดสินใจเพื่อรักษาความปลอดภัย เนื่องจากเป็นจุดเดียวที่เครือข่ายติดต่อกับเครื่องภายนอกเครือข่าย
- สามารถตรวจสอบ และเก็บรายละเอียดกิจกรรมต่างๆ ระหว่างเครือข่ายภายใน และเครือข่ายภายนอก เพราะในการติดต่อทุกครั้งต้องผ่านไฟร์วอล
- สามารถกำหนดกฎเกณฑ์ นโยบายในการอนุญาต หรือไม่อนุญาตในการใช้บริการต่างๆ ภายในเครือข่าย
- มีการตรวจตราบริการต่างๆ ทำงานได้อย่างถูกต้อง

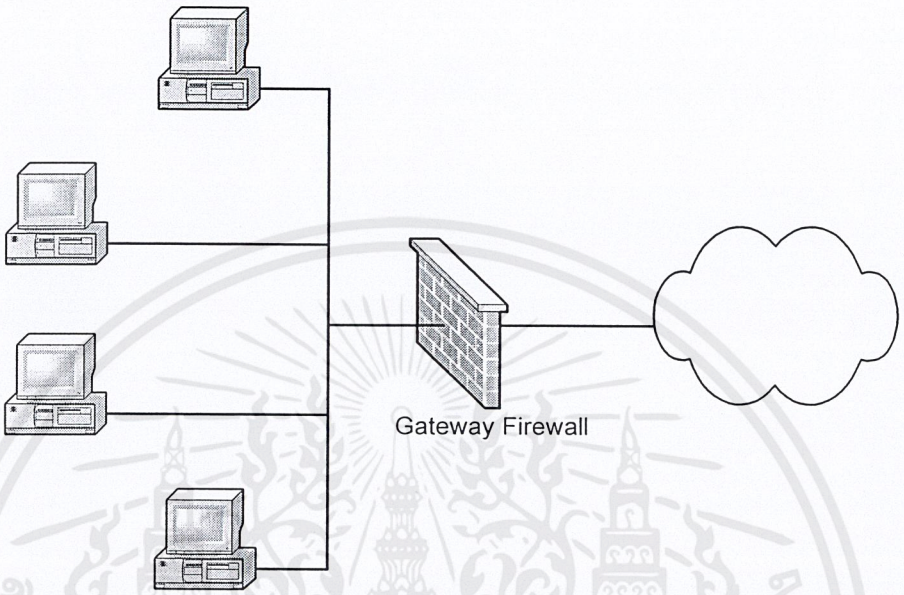
<u>สิ่งที่ไฟร์วอลไม่สามารถทำได้</u>

- ไฟร์วอลไม่สามารถป้องกันผู้บุกรุกที่อยู่ภายในเครือข่าย(Internal Network) เปรียบเสมือนการล็อคประตูบ้าน โดยที่โจรเข้ามาในบ้านแล้ว
- ไฟร์วอลไม่สามารถป้องกันการโจมตีที่ไม่ได้ผ่านไฟร์วอล เช่น ผู้ใช้ภายในเครือข่ายมีการเชื่อมต่อกับอินเทอร์เน็ตในทางอื่น ซึ่งไม่ผ่านไฟร์วอล โดยที่ผู้ดูแลระบบไม่รับทราบ เช่น การ Dial-up ไปยัง

[1] รูปแบบของข้อมูลที่ใช้ในการรับ-ส่งในเครือข่าย คือ มีส่วนของที่อยู่ผู้ส่ง-ผู้รับ พอร์ตผู้ส่ง-ผู้รับ และโพรโตคอลที่ใช้ รวมทั้งเนื้อหาของข้อมูล

อินเทอร์เน็ตจากเครื่องคอมพิวเตอร์ส่วนตัวที่อยู่ภายในเครือข่าย เปรียบได้กับเราล็อคประตูบ้านเรียบร้อยแล้ว แต่มีคนในบ้านเปิดหน้าต่างทิ้งไว้

- ไม่สามารถป้องกันไวรัสหรือ Trojan Horse[2] ได้ เพราะไฟร์วอลไม่สามารถตรวจสอบรายละเอียดข้อมูลที่อยู่ภายในแพ็กเก็ตว่ามีไวรัสหรือไม่



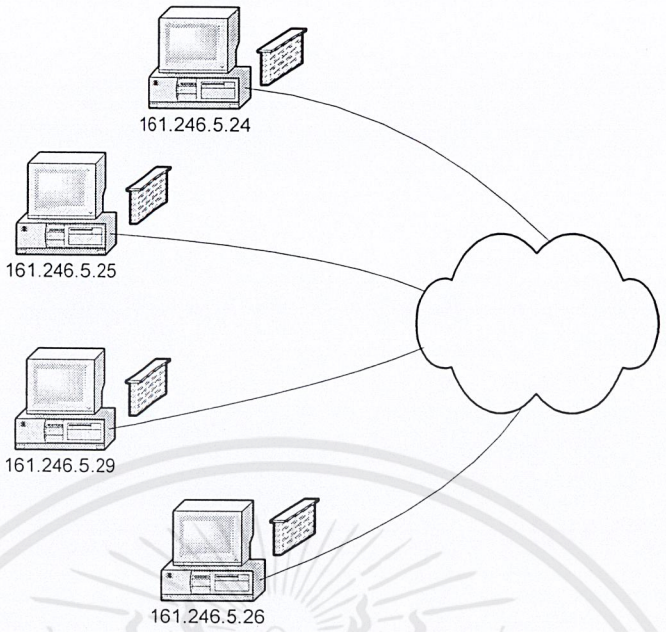
รูปที่ 3-1 เกตเวย์ไฟร์วอล

3.1.2 เพอร์ซันนอลไฟร์วอล

เพอร์ซันนอลไฟร์วอล (Personal Firewall) เป็นซอฟต์แวร์แอปพลิเคชั่นออกแบบมาเพื่อผู้ใช้ทั่วไปที่มีการเชื่อมต่อแบบ “always-on” อย่างเช่น ดีเอสแอล หรือ เคเบิลโมเด็ม โดยพื้นฐานการทำงานเป็นลักษณะเดียวกับเกตเวย์ไฟร์วอล คือ ตรวจสอบข้อมูลเข้าออก ต่างกันที่เพอร์ซันนอลไฟร์วอล จะทำงานบนเครื่องคอมพิวเตอร์เครื่องหนึ่งไม่ได้ทำงานในระดับเครือข่าย มีลักษณะการทำงานคล้ายโปรแกรมประเภทป้องกันไวรัส เพอร์ซันนอลไฟร์วอลจึงได้รับความสนใจ เนื่องจากความเสี่ยงของการถูกบุกรุกในปัจจุบันนั้นเพิ่มมากขึ้น

---

[2] เป็นโปรแกรมที่แอบแฝงมาในคราบของโปรแกรมปกติ แต่โปรแกรมประเภทนี้ยังทำหน้าที่อื่นแอบแฝงโดยที่เราไม่รู้ตัว เช่น ดักจับรหัสผ่าน เก็บข้อมูลการกดแป้นพิมพ์ เป็นต้น รวมไปถึงที่เกี่ยวข้องกับพอร์ตคือ เมื่อถึงเวลาที่กำหนดไว้ โทรจันเหล่านี้ ก็จะอาศัยพอร์ตใดพอร์ตหนึ่งในการแอบส่งข้อมูลออกไปยังจุดหมายปลายทางที่ใดที่หนึ่ง



รูปที่ 3-2 เพอร์ซันนอลไฟร์วอล

3.2 รูปแบบการทำงานของไฟร์วอล

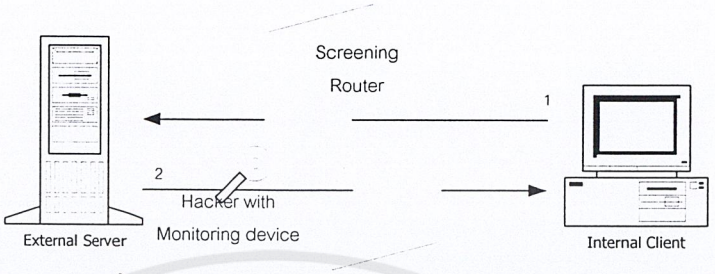
เราสามารถแบ่งรูปแบบการทำงานของไฟร์วอลออกได้เป็น 3 ประเภท ตามกลวิธีในการป้องกันเครือข่ายออกจากเครือข่ายอื่นๆ สำหรับการทำงานบนเราเตอร์ในเลเยอร์ระดับล่าง จะใช้วิธีการกรองแพ็กเก็ต โดยอาศัยข้อมูลในส่วนเฮดเดอร์ เรียกว่า "แพ็กเก็ตฟิลเตอริง" (Packet Filtering) ส่วนการทำงานในเลเยอร์ระดับสูง ซึ่งจะมีพร็อกซีเซิร์ฟเวอร์คอยตรวจสอบเนื้อหาภายในแพ็กเก็ตและแสดงผลการตรวจสอบ เรียกว่า "พร็อกซีเซิร์ฟเวอร์เกตเวย์" และประเภทสุดท้าย จะเก็บสถานะการทำงานไว้เป็นลำดับขั้น คือ "สเตทฟูลอินสเป็คชัน"

3.2.1 แพ็กเก็ตฟิลเตอริง

เป็นวิธีที่ใช้กันอย่างแพร่หลาย หลักการทำงานคือ ไฟร์วอลจะกรองแพ็กเก็ตโดยพิจารณาแพ็กเก็ตที่เข้าออกตามกฎที่ตั้งไว้ การตัดสินใจกรองแพ็กเก็ตเหล่านี้จะยึดข้อมูลที่อยู่ในเฮดเดอร์ของแพ็กเก็ตตัวนั้นๆ เช่น แอดเดรสต้นทาง แอดเดรสปลายทาง พอร์ตหรือโพรโตคอล เมื่อแพ็กเก็ตแต่ละแพ็กเก็ตเข้ามาสู่ไฟร์วอล จะนำมาเทียบกับกฎ หากเข้ากับกฎใดกฎหนึ่ง ก็จะดูว่ากฎนั้นสั่งให้ส่งแพ็กเก็ตต่อไป หรือดร็อปแพ็กเก็ตนั้นทิ้งไป และหากไม่เข้ากับกฎใดเลย ก็จะพิจารณาว่าค่าดีฟอลต์เป็นอะไรให้ส่งหรือดร็อปทิ้งไป

แพ็กเก็ตฟิลเตอริง เป็นการทำงานในชั้นเน็ตเวิร์ก วิธีการนี้เป็นวิธีที่ง่ายและทำงานได้รวดเร็วที่สุด แต่จุดอ่อนคือ ความยากในการกำหนดกฎต่างๆ ให้รัดกุมและการติดต่อกันระหว่างโฮสต์ต้นทางและโฮสต์ปลายทางได้โดยตรง อาจส่งผลให้ข้อมูลต่างๆ ของโฮสต์ปลายทาง รวมทั้งโฮสต์อื่นๆ ที่ติดต่ออยู่กับโฮสต์ปลายทางถูกโจมตีได้

ไฟร์วอลแบบนี้สามารถต่อต้านการโจมตีได้หลายชนิด เช่น IP spoofing , Source Routing Attack , Tiny Fragmentation Attack

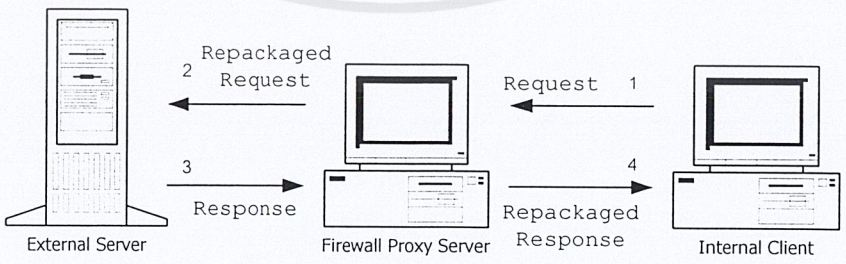


รูปที่ 3-3 รูปแบบการทำงานของแพ็กเก็ตฟิลเตอริง

3.2.2 พร็อกซีเซิร์ฟเวอร์เกตเวย์ (Proxy-Server Gateway)

พร็อกซีทำงานอยู่ในเลเยอร์ระดับสูงเป็นโปรแกรมแอปพลิเคชั่นที่ทำงานอยู่ระหว่างสองเครือข่าย โดยจะทำงานในลักษณะของการส่งข้อมูลต่อให้ เมื่อแพ็กเก็ตมาถึง พร็อกซีจะแยกข้อมูลส่วนที่เป็นเฮดเดอร์ของแพ็กเก็ตออก เหลือแต่ส่วนของข้อมูลในส่วนของชั้นแอปพลิเคชั่น เช่น หากเป็นข้อมูลเว็ป พร็อกซีจะแยกเฮดเดอร์ออกเหลือแต่ส่วนโพรโตคอล HTTP จากนั้นก็จะนำเฮดเดอร์มาพิจารณากับกฎต่างๆ ที่กำหนดเอาไว้ หากเป็นไปตามกฎที่ให้ส่งต่อ ก็จะนำข้อมูล HTTP นั้นมาประกอบเป็นแพ็กเก็ตขึ้นมาใหม่ แล้วส่งต่อตามฟังก์ชันการหาเส้นทาง

จากการที่พร็อกซี เพิ่มความสามารถ ในการติดตามและควบคุมการผ่านเข้าออก ของแพ็กเก็ตระหว่างเครือข่าย คือ เมื่อผู้ใช้ภายในต้องการส่งข้อมูลไปยังเครื่องใดเครื่องหนึ่งในอินเทอร์เน็ต จะต้องส่งผ่านมาให้กับพร็อกซี เมื่อพร็อกซีได้รับและ ตรวจสอบว่าเป็นไปตามกฎแล้ว พร็อกซีจึงจะส่งข้อมูลต่อให้อินเทอร์เน็ตอีกทีหนึ่ง เสมือนกับเป็นการส่งมาจากผู้ใช้ภายในโดยตรง การทำงานแบบนี้ ทำให้การติดต่อระหว่างผู้ใช้ภายในกับภายนอก ไม่ต้องติดต่อกันโดยตรง ผู้ดูแลระบบสามารถมองเห็นเหตุการณ์ที่เกิดขึ้น บริเวณเกตเวย์ แต่พร็อกซีจะต้องรับภาระอย่างหนัก ทำให้ประสิทธิภาพในการติดต่อระหว่างเครือข่ายลดลง ทำงานช้า แต่มีความปลอดภัยมากกว่าเนื่องจากมีการพิจารณาข้อมูลถึงระดับแอปพลิเคชั่น ทั้งนี้ไฟร์วอลจะส่งต่อได้เฉพาะโพรโตคอลที่ไฟร์วอลรู้จักเท่านั้น



รูปที่ 3-4 รูปแบบการทำงานของพร็อกซีเซิร์ฟเวอร์เกตเวย์

พร็อกซีเซิร์ฟเวอร์เกตเวย์แบ่งออกได้เป็น 2 ประเภท

- **เซอร์กิต-เลเวล เกตเวย์ (Circuit – Level Gateway)**

เป็นพร็อกซีที่ทำการควบคุมการติดต่อกันระหว่างเครือข่ายภายในและภายนอกโดยไม่มีช่องว่าง กล่าวคือ จะมีวงจรเสมือน (Virtual Circuit) อยู่ระหว่างโฮสต์ภายในเครือข่ายกับพร็อกซีเซิร์ฟเวอร์ เมื่อมีการร้องขอการติดต่อ (Request) จากเครือข่ายภายใน แพ็กเก็ตจะถูกส่งให้วงจรเสมือน ผ่านไปยังพร็อกซีเซิร์ฟเวอร์ซึ่งจะส่งการร้องขอการติตต่อไปยังอินเทอร์เน็ต หลังจากทำการแปลงหมายเลขไอพีเรียบร้อยแล้ว ในทำนองเดียวกัน การตอบรับ(Response)จากอินเทอร์เน็ต จะถูกส่งมายังพร็อกซีเซิร์ฟเวอร์ผ่านวงจรเสมือน ก่อนส่งกลับให้โฮสต์ต้นทาง โดยในการติดต่อนี้เครือข่ายภายนอกจะไม่สามารถมองเห็นโฮสต์ใดๆ ภายในเครือข่ายได้เลย การติดต่อแบบนี้จะใช้ในกรณีที่ผู้ใช้ภายในเครือข่ายกับอินเทอร์เน็ตไว้ใจได้เท่านั้น

- **แอปพลิเคชั่น-เลเวล เกตเวย์ (Application – Level Gateway)**

สำหรับแอปพลิเคชั่นเกตเวย์ นอกจากจะทำงานเช่นเดียวกับวงจรเสมือนแล้ว ยังเพิ่มความสามารถในการตรวจสอบแพ็กเก็ตด้วย ไม่ว่าจะเป็นส่วนของเฮดเดอร์หรือเนื้อหาข้อมูลภายในแพ็กเก็ต เพื่อหยุดยั้งการส่งข้อมูลจากภายนอก หากมีข้อมูลจากแฮ็กเกอร์แอบซ่อนอยู่ภายในแพ็กเก็ต นอกจากนี้แอปพลิเคชั่นเกตเวย์ยังสนับสนุนการให้บริการ สำหรับโพรโตคอลแบบต่างๆ คือ เทลเน็ต, เอฟทีพี , เอชทีทีพี และ เอสเอ็มทีพี โดยที่ผู้ดูแลระบบจะต้องทำการติดตั้งในพร็อกซีแยกสำหรับแต่ละการให้บริการ

3.2.3 สเตทฟูลอินสเป็คชั่นหรือไดนามิกแพ็กเก็ตฟิลเตอริง

เป็นรูปแบบการทำงานแบบใหม่ ซึ่งปรับปรุงมาจากแพ็กเก็ตฟิลเตอริงอย่างเดิมซึ่งจะตัดสินใจโดยดูข้อมูลในส่วนของเฮดเดอร์เท่านั้น และจะพิจารณาเฉพาะแพ็กเก็ตนั้นๆ โดยไม่ได้คำนึงถึงแพ็กเก็ตก่อนหน้านี้ ส่วนสเตทฟูลอินสเป็คชั่นจะเอาข้อจำกัดนี้ ทั้งในส่วนของข้อมูลที่ใช้ในการตัดสินใจ โดยจะดูถึงข้อมูลในส่วนของ Payload ด้วย และจะคำนึงถึงส่วนของแพ็กเก็ตอื่นๆ ที่อยู่ก่อนหน้านี้ประกอบในการตัดสินใจ ทำให้ความสามารถในการตัดสินใจมากขึ้น เพราะรู้ข้อมูลมากขึ้น โดยเพิ่มตารางเก็บสถานะ เช่น เมื่อส่งข้อมูลให้อินเทอร์เน็ต ตารางสถานะจะเก็บหมายเลขพอร์ตต้นทาง หมายเลขพอร์ตปลายทาง การเก็บนี้เรียกว่า “Saving the state” เมื่อมีแพ็กเก็ตที่เป็นการตอบรับ ก็จะนำแพ็กเก็ตที่รับมานั้น เปรียบเทียบกับ “Saved state” ที่เก็บไว้ว่ามีข้อมูลตรงกัน

การทำงานของสเตทฟูลอินสเป็คชั่น จะมีการติดตามสถานะ(State) การทำงานของการเชื่อมต่อแบบ ทีซีพี ซึ่งเป็นผลให้สามารถดูรูปแบบการทำงานได้ทั้งกระบวนการ ไม่ได้ดูเพียงข้อมูลในแต่ละแพ็กเก็ต โดยสามารถดูลักษณะการเชื่อมต่อ การโต้ตอบของแต่ละโพรโตคอลที่มีลักษณะที่แตกต่างกัน โดยสามารถแยกแยะโพรโตคอลที่ถูกต้องกับโพรโตคอลที่ไม่ถูกต้องออกจากกันได้ นอกจากนี้สเตทฟูลอินสเป็คชั่นยังมีความปลอดภัยมากกว่า เพราะสามารถจะปิดพอร์ตที่มีหมายเลขมากกว่า 1,024 ได้ เนื่องจากการเชื่อมต่อแบบ ทีซีพี อย่างเช่นเว็บนั้น แม้เมื่อเริ่มแรกจะติดต่อกันโดยผ่านพอร์ต 80 แต่หลังจากที่ติดต่อกันแล้ว จะมีการใช้หมายเลขพอร์ตแบบสุ่ม โดยมีหมายเลขมากกว่า 1,024 ซึ่งทำให้ไฟร์วอลแบบ

แพ็กเก็ตฟิลเตอริงจำเป็นต้องเปิดพอร์ตที่มีหมายเลขมากกว่า 1,024 ไว้ตลอดเวลาแต่สเตทฟูลจะเปิดพอร์ตเฉพาะเวลาที่มีการเชื่อมต่อผ่านพอร์ตนั้นๆ เท่านั้น หากแพ็กเก็ตถูกตรวจสอบแล้วว่าไม่เป็นไปตามกฎก็จะปิดพอร์ตนั้นทันที สำหรับข้อจำกัดของสเตทฟูลอินสเป็คชั่น คือ ไม่รู้จักการทำงานในระดับแอปพลิเคชั่นและยังคงเป็นการติดต่อระหว่างผู้ใช้ภายในกับภายนอกโดยตรง

ดังนั้นไฟร์วอลที่ดี จึงมักจะนำวิธีการหลายๆวิธีเข้ามาใช้ด้วยกัน เช่น ใช้พร็อกซีเซิร์ฟเวอร์และและสเตทฟูลอินสเป็คชั่นทำงานร่วมกัน

	Packet Filter	**Stateful Inspection**	**Proxy-Server Gateways**
ข้อดี	• ประสิทธิภาพดี • ง่ายในการ implement • ไม่ขึ้นกับแอปพลิเคชั่น (Application Independent)	• ประสิทธิภาพดี • เปิดพอร์ตเฉพาะเมื่อมีการติดต่อ • สนับสนุนเกือบทุกบริการ	• ไม่เปิดเผยหมายเลขไอพีภายใน • พิจารณาเนื้อหาของข้อมูลด้วย • มี User Authentication • เก็บรายละเอียด log ได้มาก
ข้อเสีย	• เปิดเผยหมายเลขไอพีภายใน • มีการเปิดช่องว่างทิ้งไว้ถาวร • No User Authentication • ใช้การเชื่อมต่อโดยตรงกับภายนอก	• No User Authentication • ใช้การเชื่อต่อโดยตรงกับภายนอก • เปิดเผยหมายเลขไอพีภายใน	• ประสิทธิภาพต่ำกว่า • ต้องมีพร็อกซี สำหรับทุกๆแอปพลิเคชั่นที่ใช้ • ไม่มีการป้องกันในระดับชั้นที่ต่ำกว่าชั้นแอปพลิเคชั่น • เปิดเผยระบบปฏิบัติการ

ตารางที่ 3-1 เปรียบเทียบรูปแบบการทำงานของไฟร์วอลทั้ง 3 ประเภท

3.3 การกำหนดกฎของไฟร์วอล

สำหรับการกำหนดกฎโดยทั่วไป จะมี 2 แบบ คือ

- Default Deny Stance : อะไรที่ไม่ได้ระบุว่าอนุญาตถือว่าไม่อนุญาต
- Default Permit Stance : อะไรที่ไม่ได้ระบุว่าไม่อนุญาตถือว่าอนุญาต

โดยส่วนใหญ่แล้วจะเลือกวิธีการแบบที่ 1 เพราะในแง่ของความปลอดภัยถือว่าวิธีการที่ 1 มีความปลอดภัยสูงกว่า เนื่องจากวิธีการแบบที่ 2 เป็นการยากที่จะสามารถระบุกฎให้ครอบคลุมจุดอ่อนทั้งหมดได้ ฟิลด์ในเฮดเดอร์ที่สามารถนำมาใช้พิจารณาได้ มีดังนี้

- แอดเดรสต้นทางและแอดเดรสปลายทาง
- ส่วนขยายของไอพี
- โพรโตคอล เช่น ทีซีพี , ยูดีพีหรือไอซีเอ็มพี
- หมายเลขพอร์ตต้นทางและหมายเลขพอร์ตปลายทาง
- ชนิดของข้อความแจ้งเตือน
- ACK bit ใน ทีซีพีเซกเมนต์

ในที่นี้จะขอยกตัวอย่างในการกำหนดกฎเกณฑ์เพื่อง่ายต่อการเข้าใจ

Service Destination	Packet Direction	Source Address	Dest. Address	Packet Type	Source Port	Dest. Port	ACK Set
Outbound	Outgoing	Internal	External	TCP	Y	23	
Outbound	Incoming	External	Internal	TCP	23	Y	Yes
Inbound	Incoming	External	Internal	TCP	Z	23	
Inbound	Outgoing	Internal	External	TCP	23	Z	Yes

ตารางที่ 3-2 ประเภทของแพ็กเก็ตที่ให้บริการเทลเน็ต

Y, Z เป็นพอร์ตที่ได้มาจากการสุ่ม ซึ่งมีค่ามากกว่า 1023

หากไม่ต้องการให้อนุญาตทุกแพ็กเก็ต ยกเว้น outgoing เทลเน็ตก็จะสามารถกำหนดกฎเกณฑ์ได้ดังนี้

Rule	Direction	Source Address	Dest. Address	Protocol	Source Port	Dest. Port	ACK Set	Action
A	Out	Internal	Any	TCP	>1023	23	Either	Permit
B	In	Any	Internal	TCP	23	>1023	Yes	Permit
C	Either	Any	Any	Any	Any	Any	Either	Deny

ตารางที่ 3-3 ตัวอย่างการกำหนดกฎที่อนุญาตเฉพาะการให้บริการเทลเน็ต

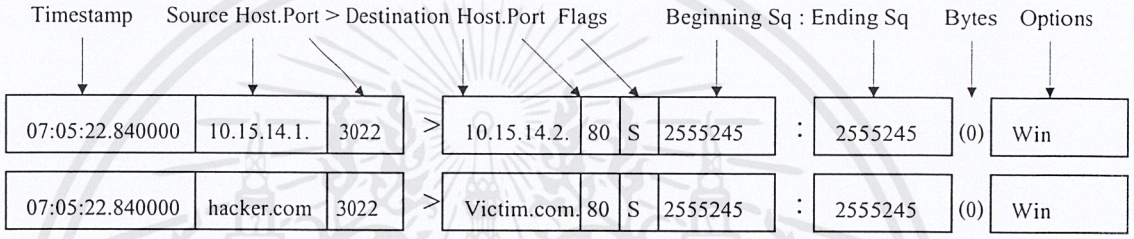
รายละเอียดของตัวอย่างในตารางที่ 3-3 มีดังนี้

- A อนุญาตทุกแพ็กเก็ตเพื่อขอใช้บริการเทลเน็ต
- B อนุญาตแพ็กเก็ตตอบรับ เนื่องจากสามารถพิสูจน์ได้ว่า ACK บิต ถูกเซ็ท
- C เป็นดีฟอลต์ ถ้าข้อมูลของเฮดเดอร์ ไม่ตรงกับทั้ง A และ B แพ็กเก็ตจะไม่ได้รับอนุญาตให้ผ่านเข้ามาตามกฎ C

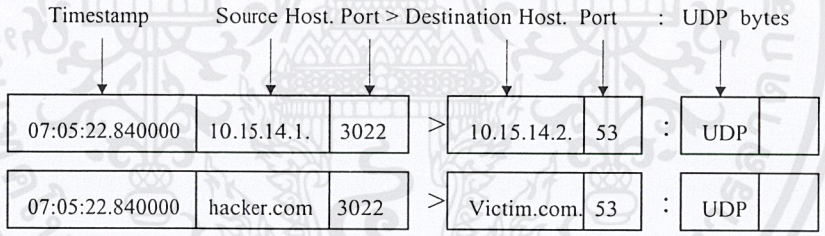
บางครั้งการกำหนดกฎที่คลุมเครือ อาจทำให้มีความขัดแย้งกันของกฎได้

3.4 วิธีการอ่านแพ็กเก็ตเพื่อนำมาใช้ฟิลเตอร์

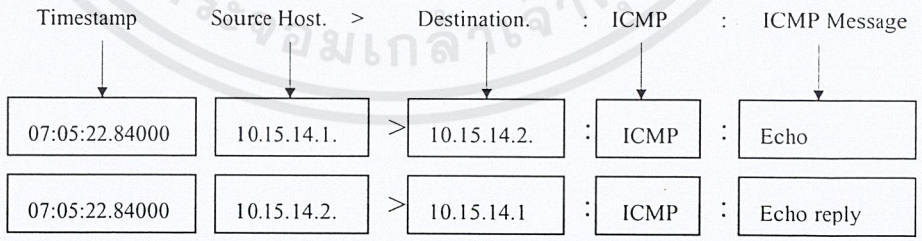
1. โพรโตคอล ทีซีพี/ไอพี



2. โพรโตคอล UDP



2. โพรโตคอล ICMP



3.5 ฟิลด์ที่นำมาใช้ในการฟิลเตอร์แพ็กเก็ต

Version	IHL	Type of Service	Total Length	
Identification			Flags	Fragment Offset
TTL		**Protocol**	Header Checksum	
Source IP Address				
Destination IP Address				
Option				
Data				

โดยหากโพรโตคอลคือ ทีซีพี ก็จะนำส่วนเฮดเดอร์ของทีซีพีมาพิจารณาด้วย

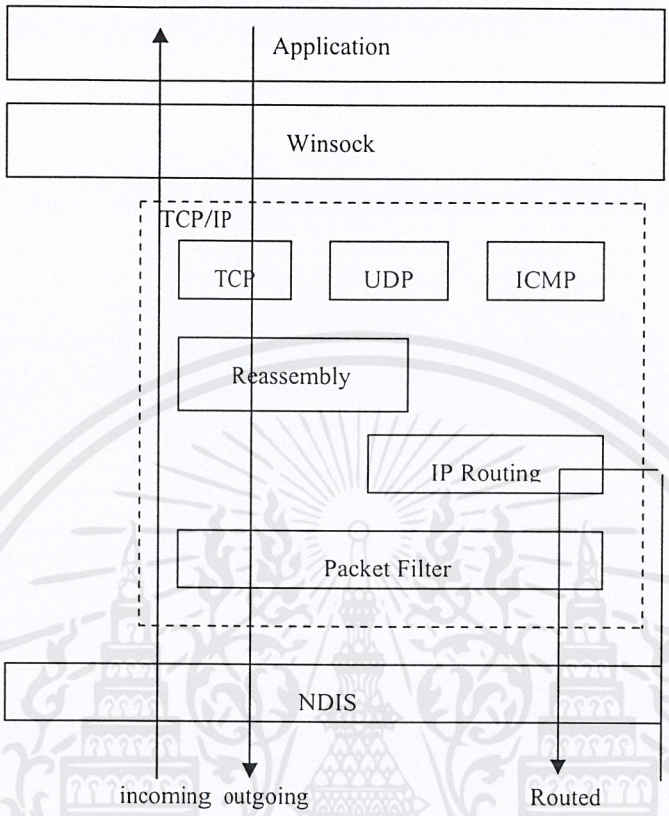
Source Port							**Destination Port**
Sequence No.							
Acknowledgement No.							
			ε				Window
Checksum							Urgent Pointer
Options and Padding							

โพรโตคอลยูดีพี

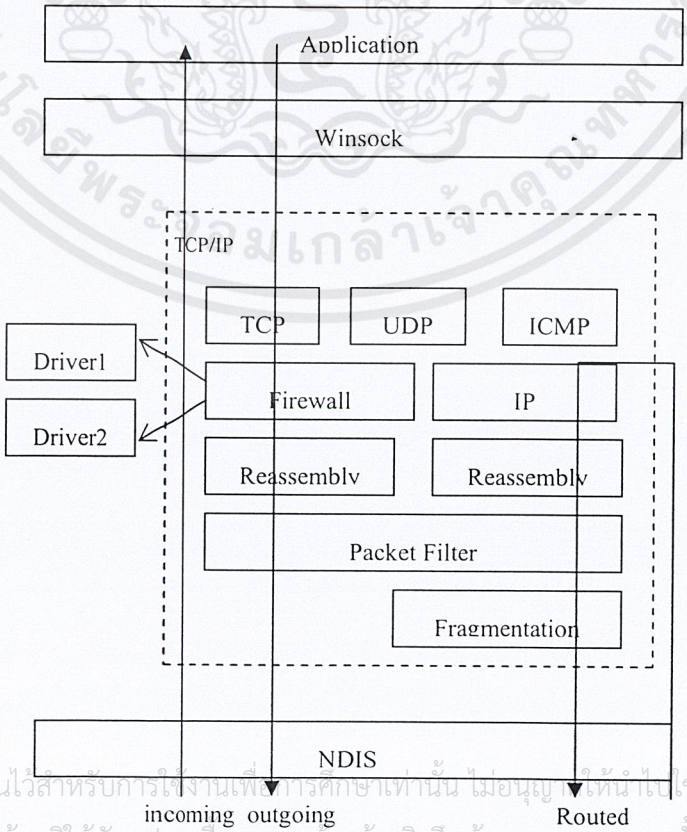
Source Port	**Destination Port**
Length	Checksum

3.6 เส้นทางเดินของแพ็กเก็ต

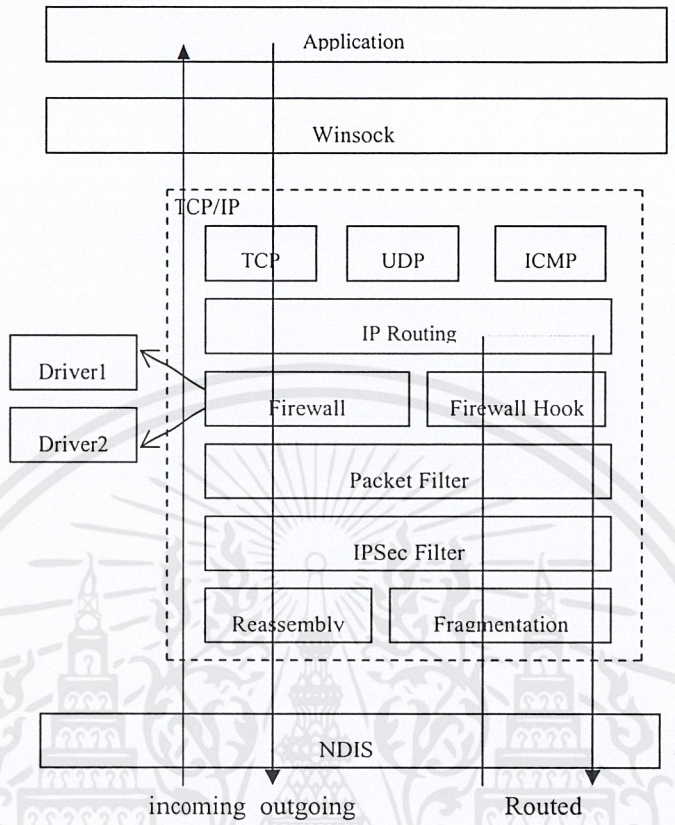
3.6.1 เส้นทางเดินของแพ็กเก็ตเมื่อไม่มีไฟร์วอลฮุกและ IPSec



3.6.2 เส้นทางเดินของแพ็กเก็ตที่ผ่านไฟร์วอลฮุก แต่ไม่ผ่าน IPSec



3.6.3 เส้นทางของแพ็กเก็ตที่ผ่านไฟร์วอลฮุกและ IPSec



3.7 ภัยจากการโจมตีและความสามารถของไฟร์วอล

3.7.1 SYN flooding

เครื่องผู้บุกรุก ส่ง SYN Flag มายังเครื่องเป้าหมาย เพื่อทำ 3way-handshaking เมื่อเครื่องเป้าหมายได้รับ SYN Flag จะส่ง ACK และ SYN กลับไปยังเครื่องผู้บุกรุก ในขณะเดียวกันเครื่องเป้าหมายจะจองพื้นที่ในหน่วยความจำเพื่อรอรับ ACK ซึ่งหากผู้บุกรุกปลอมหมายเลขไอพีและส่ง SYN จำนวนมาก หน่วยความจำของเครื่องเป้าหมายจะถูกใช้จนหมด จนเครื่องเป้าหมายไม่สามารถให้บริการได้หรือใช้งานได้ชั่วขณะ

การป้องกันโดยไฟร์วอล

SYN flooding เป็นการโจมตีในเชิงปริมาณและความเร็ว ซึ่งกฎของไฟร์วอลไม่ครอบคลุมถึงปริมาณและความเร็วของข้อมูล การมีไฟร์วอลเป็นด่านหน้าจะช่วยบรรเทาความรุนแรงเนื่องจากการ SYN flood ได้ เพราะไฟร์วอลจะเป็นผู้รักษาการติดต่อไว้ทั้งหมดตั้งแต่เริ่ม SYN จนสิ้นสุด เท่านั้น

3.7.2 PING Attack

ไอซีเอ็มพีมีหน้าที่ส่งข่าวสารและคำสั่งควบคุมของไอพี โดยเฉพาะการรายงานข้อผิดพลาดในการรับส่ง PING เป็นข้อความไอซีเอ็มพีประเภทหนึ่งใช้ในการหาข้อมูลเกี่ยวกับเครือข่าย

การป้องกันโดยไฟร์วอล

ไม่ควรให้เครื่องภายนอกไฟร์วอลหาข้อมูลของเครื่องภายในเครือข่ายได้ โดยการกำหนดกฎ

Rule	Src Addr	Dst Addr	Protocol	Action
A	External	Internal	echo reply	permit
B	External	Internal	echo request	deny

3.7.3 Tiny Fragmentation

หากแพ็กเก็ตมีขนาดใหญ่เกินกว่าที่ชั้นดาต้าลิงก์อนุญาต โพรโตคอลไอพี มีความสามารถในการแบ่งแพ็กเก็ตออกเป็นแพ็กเก็ตย่อยๆ แล้วรวมกลับให้เหมือนเดิมเมื่อแพ็กเก็ตถึงปลายทาง เนื่องจากโพรโตคอล ทีซีพี จะมีเฮดเดอร์แพ็กเก็ตของตัวเอง และถูกเติมเข้ากับเฮดเดอร์ของโพรโตคอลไอพีก่อนส่งไปยังปลายทาง หากแพ็กเก็ตมีขนาดใหญ่ แพ็กเก็ตก็จะถูกแบ่งออกเป็นแพ็กเก็ตย่อย ซึ่งเพียงแพ็กเก็ตแรกเท่านั้นที่บอกว่าแพ็กเก็ตนี้ถูกส่งมาจากหมายเลขไอพีใด หากเป็นหมายเลขไอพีที่ไม่ได้รับอนุญาตให้ผ่านไฟร์วอล ก็จะมีเพียงแพ็กเก็ตแรกแพ็กเก็ตเดียวเท่านั้นที่ถูกดร็อป หรืออาจจะไม่มีการดร็อปแพ็กเก็ตใดๆเลย หาก Sequence No มีการแก้ไขเป็น1 (ซึ่งปกติเป็น 0) เสมือนว่าไม่ใช่แพ็กเก็ตแรก

3.7.4 Port Scanning

การสแกนพอร์ตทุกครั้ง ผลลัพธ์ที่ได้คือรายละเอียดที่จะบอกได้ว่า โฮสต์ที่ถูกสแกนใช้ระบบปฏิบัติการอะไร มีแอปพลิเคชั่นใดทำงานอยู่บ้าง เปรียบเสมือนบอกผู้บุกรุกว่ามีช่องทางใดที่จะเจาะระบบเข้ามาได้

ผู้บุกรุกสามารถเจาะเข้าสู่ระบบได้ โดยอาศัยข้อบกพร่องของแอปพลิเคชั่น โดยข้อมูลเหล่านี้สามารถหาได้ตามเว็บไซท์ที่เป็นแหล่งชุมชนของผู้บุกรุก

การสแกนพอร์ตจะทำให้ผู้บุกรุกทราบได้ว่ามีแอปพลิเคชั่นใดทำงานอยู่ หลังจากนั้นจะทำการสแกนแบบเจาะลึก ก็จะทราบว่าแอปพลิเคชั่นนั้นเป็นโปรแกรมอะไร เวอร์ชั่นใด มีข้อบกพร่องอย่างไร และควรนำเครื่องมือ หรือเทคนิคใดในการเจาะระบบ

พอร์ตที่เปิดให้บริการ จึงเป็นสิ่งที่ดึงดูดใจผู้บุกรุกเสมอ เพราะบอกว่าได้เป้าหมายมีส่วนประกอบอะไร ทำหน้าที่อะไร มีความสำคัญแค่ไหน และมีจุดอ่อนจุดแข็งอย่างไร

การเปิดพอร์ต

เราสามารถกำหนดได้เพียงส่วนที่อยู่ในระดับไอพี คือ หมายเลขไอพี สับเน็ตมาสก์ และ เกตเวย์เท่านั้น จะไม่สามารถกำหนดได้ว่าจะเปิดปิดพอร์ตใดบ้าง การที่พอร์ตใดจะเปิดให้บริการเป็นเซิร์ฟเวอร์พอร์ตนั้นจะต้องมีแอปพลิเคชั่นทำงานอยู่บนพอร์ตนั้นเสมอ คือมีโปรแกรมที่รับหน้าที่โต้ตอบและจัดการการสื่อสารที่มายังพอร์ตนั้น จึงอาจเปรียบได้ว่าพอร์ตก็คือแอปพลิเคชั่น การที่มีพอร์ตเปิดอยู่ก็หมายถึงการมีแอปพลิเคชั่นทำงานอยู่

นอกจากแอปพลิเคชั่นจะเปิดพอร์ตเพื่อใช้งานแล้ว ระบบปฏิบัติการที่อาศัย ทีซีพี/ไอพี ก็จะต้องเปิดพอร์ตเพื่อใช้ในกิจการของระบบปฏิบัติการด้วย โดยที่ผู้ใช้ไม่รู้ตัว เพราะเป็นการใช้งานภานในของระบบปฏิบัติการและผู้ผลิตคิดว่าผู้ใช้ไม่จำเป็นต้องรู้ จึงทำให้ผู้ใช้ถูกบุกรุกจากพอร์ตเหล่านี้ด้วย ซึ่งเมื่อเริ่มใช้แอปพลิเคชั่นมาก เครื่องคอมพิวเตอร์ของเราก็จะเริ่มเปิดพอร์ตมากขึ้น ซึ่งเป็นการเปิดช่องทางให้ผู้อื่นติดต่อเข้ามาได้มากขึ้นตามไปด้วย

การปิดพอร์ต

การปิดพอร์ต คือ การไม่ยอมรับการติดต่อเข้ามายังพอร์ตนั้นๆ เช่นเดียวกับการเปิดพอร์ต เราไม่สามารถปิดพอร์ตนั้นโดยตรงได้ด้วยโฮสต์ทั่วไป หากจะปิดพอร์ต จะต้องหยุดการทำงานของแอปพลิเคชั่นก่อนแล้วพอร์ตจะถูกปิดไปเอง หรือสามารถทำได้โดยผ่านไฟร์วอล เราเตอร์ หรืออุปกรณ์ Layer 4 Switch

การปิดพอร์ตไม่ใช่เรื่องยาก ปัญหาที่เกิดขึ้นเนื่องมาจากพอร์ตไม่ได้ปิด ด้วยสาเหตุต่างๆ

1. พอร์ตที่เปิดไว้โดยไม่ได้ตั้งใจ

การเปิดพอร์ตเป็นการเปิดแบบลอจิคัลและมองไม่เห็น ดังนั้นหากไม่ทำการตรวจสอบโฮสต์ของเราให้ดี จะไม่ทราบว่ามีพอร์ตใดเปิดอยู่ ส่วนใหญ่เกิดจากแอปพลิเคชั่นอื่นๆ มาเปิดพอร์ตบนโฮสต์เราโดยที่เราไม่เคยติดตั้งเข้าไปด้วยเลย โดยแอปพลิเคชั่นเหล่านี้ได้มาตั้งแต่ขั้นตอนการติดตั้งระบบปฏิบัติการซึ่งผู้ผลิตคาดว่าผู้ใช้ต้องการใช้แอปพลิเคชั่นเหล่านั้น ผู้ใช้สามารถตรวจสอบว่ามีแอปพลิเคชั่นใดบ้างที่ทำงานอยู่โดยที่เราไม่ต้องการ ให้หยุดการทำงานของแอปพลิเคชั่นเหล่านั้น พอร์ตก็จะถูกปิดไปเอง

2. พอร์ตของระบบปฏิบัติการ

เป็นพอร์ตที่จำเป็นสำหรับระบบปฏิบัติการนั้นๆ หากไม่เปิดพอร์ตเปล่านี้ ระบบปฏิบัติการก็จะไม่สามารถทำงานได้อย่างสมบูรณ์ เช่น ไมโครซอฟท์ วินโดวส์ เอ็นที จะต้องใช้พอร์ต 135-139 ของ ทีซีพี ในการทำงาน พอร์ตประเภทนี้จะไม่สามารถปิดลงได้ เนื่องจากแอปพลิเคชั่นที่ใช้งานพอร์ตนั้นเป็นส่วนหนึ่งของระบบปฏิบัติการ ข้อเสียอย่างมากก็คือ นอกจากผู้ใช้จะไม่สามารถปิดพอร์ตเหล่านี้ได้ ยังเป็นการบอกผู้บุกรุกอีกด้วยว่าใช้ระบบปฏิบัติการอะไร และทำให้ผู้บุกรุกสามารถโจมตีได้ง่ายขึ้น

3. พอร์ตที่เปิดแบบสุ่ม

เกิดจากแอปพลิเคชั่นบางประเภทที่มีการใช้งานพอร์ตมากกว่า 1 พอร์ต โดยมีหมายเลขพอร์ตที่คงที่ไว้เป็นหลัก 1 พอร์ต ส่วนพอร์ตที่จะเปิดเป็นการชั่วคราวนี้ ไคลเอนต์และเซิร์ฟเวอร์ จะมีการตกลงกันเพื่อเปลี่ยนไปสื่อสารกันที่พอร์ตนั้นๆ ซึ่งการเปิดพอร์ตประเภทนี้มีปัญหาคือ

- พอร์ตจะปิดลงเมื่อการใช้งานเสร็จสิ้น แต่หากแอปพลิเคชั่นทำงานผิดพลาดหรือหยุดลงกลางคัน พอร์ตก็อาจจะถูกเปิดค้างทิ้งไว้
- การไม่มีหมายเลขพอร์ตแน่นอน ทำให้ควบคุมและตรวจสอบได้ยาก หากพอร์ตที่ใช้บังเอิญตรงกับพอร์ตที่อันตรายซึ่งใช้โดยโปรแกรมประเภทโทรจัน
- หากมีการนำไฟร์วอลมาใช้งาน การกำหนดกฎสำหรับไฟร์วอลจะทำได้ยาก เพราะกฎของไฟร์วอลจะตั้งอยู่บนพื้นฐานของการใช้พอร์ตเป็นหลัก

การป้องกันโดยไฟร์วอล

สเตทฟูลอินสเปคชั่นจะเปิดพอร์ตเฉพาะเวลาที่มีการเชื่อมต่อผ่านพอร์ตนั้นๆเท่านั้น หากแพ็กเก็ตถูกตรวจสอบแล้วว่าไม่เป็นไปตามกฎก็จะปิดพอร์ตนั้นทันที

บทที่ 4
การโจมตีเพื่อให้ปิดบริการสำหรับโพรโตคอลสแตกทีซีพี/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

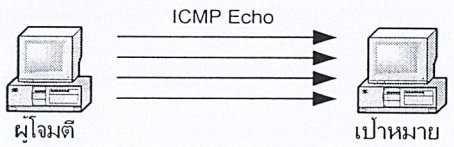
4.2.1.1 การส่งแพ็กเก็ตจำนวนมาก (Amount of Packets Sending)

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 และการปลอมไอพียังเป็นหลักประกันได้ว่าจะไม่สามารถติดตามได้ว่าผู้ได้เป็นผู้โจมตี 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

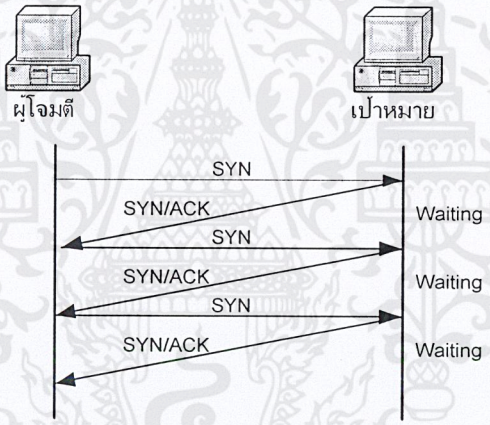
นอกจาก Ping floodign Attack จะสร้างความเสียหายแก่เครื่องเป้าหมายแล้วยังสร้างความเสียหายแก่ระบบเครือข่ายของเครื่องเป้าหมายด้วย เพราะการโจมตีวิธีนี้จะสร้างแพ็กเก็ตเป็นจำนวนมากขึ้นในเครือข่ายที่เครื่องเป้าหมายตั้งอยู่ ทำให้ระบบเครือข่ายเกิดความคับคั่งของข้อมูล(Congestion) อาจส่งผลให้เครือข่ายเป็นอัมพาตได้

การป้องกันการการโจมตีลักษณะนี้ทำได้โดยการกำหนดที่อุปกรณ์เราเตอร์หรือไฟร์วอล 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 ตัวอย่างแพ็กเก็ตประเภทนี้คือ สัญญาณ SYN หรือ ACK สำหรับการสถาปนาการเชื่อมต่อ หรือสัญญาณ FIN สำหรับการยกเลิกการเชื่อมต่อ เป็นต้น

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 > **isag24.ce.kmitl.ac.th.80:** S 399259715:399259715(0)

10:09:43.139 10.0.0.2 > **isag24.ce.kmitl.ac.th.80:** S 399259715:399259715(0)

10:09:43.143 10.0.0.3 > **isag24.ce.kmitl.ac.th.80:** S 399259715:399259715(0)

10:09:43.149 10.0.0.4 > **isag24.ce.kmitl.ac.th.80:** S 399259715:399259715(0)

10:09:43.158 10.0.0.5 > **isag24.ce.kmitl.ac.th.80:** S 399259715:399259715(0)

10:09:43.165 10.0.0.6 > **isag24.ce.kmitl.ac.th.80:** S 399259715:399259715(0)

10:09:43.177 10.0.0.7 > **isag24.ce.kmitl.ac.th.80:** S 399259715:399259715(0)

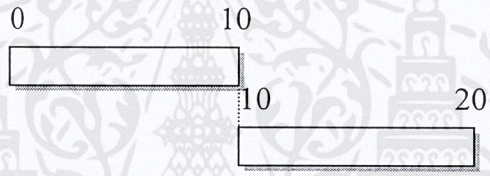
10:09:43.185 10.0.0.8 > **isag24.ce.kmitl.ac.th.80:** S 399259715:399259715(0)

รูปที่ 4-4 แพ็กเก็ตที่เกิดขึ้นจากการโจมตีแบบ SYN Flood

ในปัจจุบันนี้การโจมตีแบบ SYN Flood ถือได้ว่าเป็นการโจมตีที่ได้ผลและหาทางป้องกันได้ยาก เนื่องจากยากที่จะแยกลักษณะของแพ็กเก็ตที่ใช้ในการโจมตีกับแพ็กเก็ตที่ขอเริ่มต้นเชื่อมต่อทั่วไป นอกจากนี้ไฟร์วอลหรือเราเตอร์ทั่วไปยังไม่สามารถป้องกันการโจมตีประเภทนี้ได้อย่างสมบูรณ์ หนทางที่เป็นไปได้คือการใช้ระบบตรวจจับผู้บุกรุกทางระบบเครือข่ายทำการตรวจจับการโจมตี เพื่อนำข้อมูลจากการโจมตีกลับไปตั้งค่าอุปกรณ์เราเตอร์หรือไฟร์วอลเพื่อป้องกันการโจมตีมายังเซิร์ฟเวอร์

4.2.1.2 ความผิดปกติของแฟร็กเมนต์ (Abnormal Fragmentation)

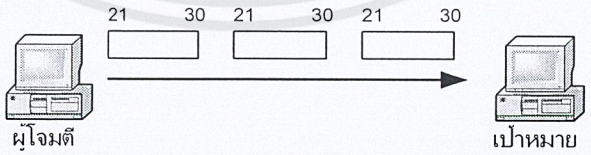
การโจมตีวิธีนี้อาศัยหลักการแฟร็กเมนเตชันและรีแอสเซมเบิลที่กล่าวไว้ข้างต้น โดยทำให้แพ็กเก็ตนั้นต้องมีการรีแอสเซมเบิล (กำหนดค่า MF flag = 0) ซึ่งปกติการรีแอสเซมเบิลแพ็กเก็ตทั้งหมดต้องสามารถเชื่อมต่อกันได้สนิท ดังรูปที่ 4-2 แต่แพ็กเกจที่ผู้บุกรุกส่งไปมีการแก้ไขข้อมูลในบางฟิลด์ ทำให้เกิดความผิดปกติในกระบวนการรีแอสเซมเบิล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



รูปที่ 4-5 แสดงการรีแอสเซมบลีแบบปกติ

(1) การส่งแพ็กเก็ตที่มีลำดับผิดปกติ (Abnormal Sequences of Packets Sending)

ปกติการส่งแพ็กเก็ตมักเรียงตามลำดับกันไป หากไม่เรียงลำดับก็ต้องรอจนกว่าแพ็กเก็ตก่อนหน้านี้มาถึง เพื่อเรียงลำดับแพ็กเก็ตที่เครื่องรับ แต่การโจมตีแบบนี้กลับส่งเฉพาะแพ็กเก็ตสุดท้าย เพื่อให้ระบบเป้าหมายรอแพ็กเก็ตก่อนหน้า และส่งไปเป็นปริมาณมากๆ ซึ่งจะส่งผลให้ระบบต้องจองทรัพยากรส่วนหนึ่งเพื่อรองรับแพ็กเก็ตที่ต้องรอเหล่านั้นในปริมาณมาก จนกระทั่งระบบไม่สามารถจัดหาทรัพยากรได้เพียงพอ ส่งผลให้ระบบไม่สามารถให้บริการอย่างอื่นได้



รูปที่ 4-6 แสดงการส่งเฉพาะแพ็กเก็ตสุดท้ายไปยังเป้าหมาย

โดยปกติแล้วการโจมตีในรูปแบบนี้ผู้โจมตีจะแก้ไขข้อมูลในฟิลด์แสดงลำดับของแพ็กเก็ต (Fragment Offset) ของแพ็กเก็ตไอพี ซึ่งเป็นส่วนที่แสดงลำดับของข้อมูลหลังจากกระบวนการแฟร็กเมนต์เตชัน โดยแก้ให้ส่งแพ็กเก็ตสุดท้ายหรือแพ็กเก็ตหลังๆ เพียงแพ็กเก็ตเดียวเลย ทำให้ระบบเป้าหมายต้องรอแพ็กเก็ตก่อนหน้านี้

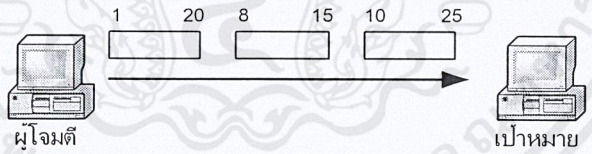
(2) การส่งแพ็กเก็ตที่มีขนาดเหลื่อมกัน (Overlapped Packets' Size Sending)

ปกติแพ็กเก็ตที่ส่งมาต้องนำมาต่อกันที่ระบบเป้าหมายได้พอดี แต่การโจมตีแบบนี้เป็นการส่งแพ็กเก็ตที่มีขนาดเหลื่อมกัน หรือซ้อนทับกัน ทำให้ข้อมูลเมื่อมาต่อกันแล้วเกิดความผิดพลาด หรือไม่สามารถเชื่อมต่อกันได้

โดยปกติแล้วการโจมตีแบบนี้ ผู้บุกรุกสามารถแก้ไขข้อมูลได้ 2 แห่งใหญ่ๆ ได้แก่

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

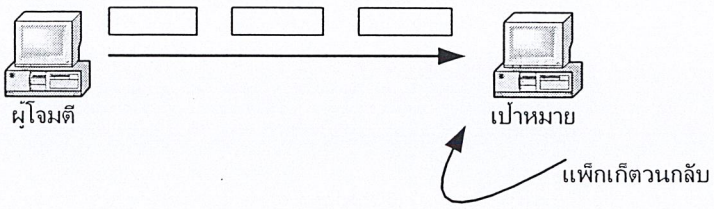
รูปที่ 4-7 แสดงการรีแอสเซมบลีแบบแพ็กเก็ตมีขนาดเหลื่อมกัน



รูปที่ 4-8 แสดงการโจมตีโดยส่งแพ็กเก็ตที่ไม่สามารถรีแอสเซมเบิ้ลได้

4.2.1.3 การส่งแพ็กเก็ตแบบวนลูป (Looping)

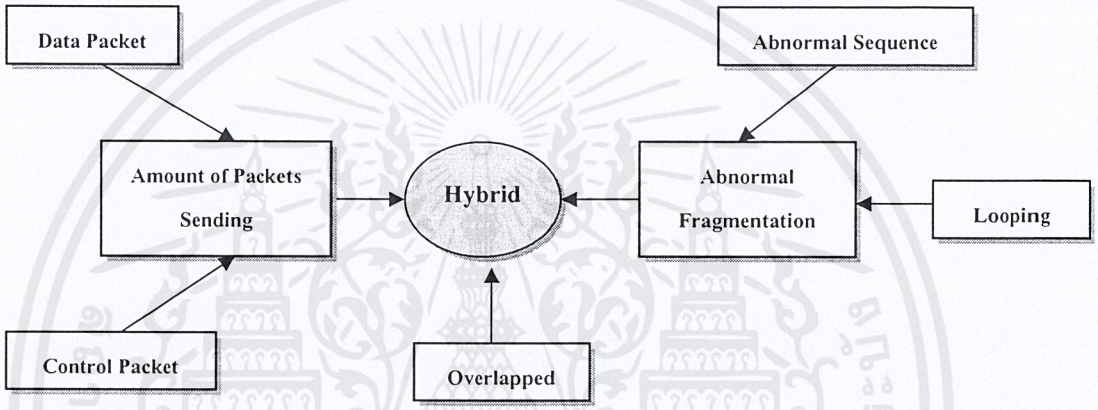
เป็นการโจมตีโดยการส่งโดยกำหนดค่าแอดเดรสต้นทาง (Source Address) และแอดเดรสปลายทาง (Destination Address) ให้เหมือนกันทำให้เกิดการรับส่งวนไปวนมาอยู่ที่เครื่องเป้าหมายเอง เช่น LAND ซึ่งเป็นโปรแกรมโจมตีที่มีการกำหนดแอดเดรสต้นทาง และแอดเดรสปลายทางเป็นค่าเดียวกัน คือเป็นแอดเดรสของเครื่องเป้าหมายนั่นเอง ทำให้เกิดการส่งวนไปวนมาอยู่ที่เครื่องเป้าหมาย



รูปที่ 4-9 แสดงการโจมตีโดยส่งแพ็กเก็ตแบบวนลูป

4.2.1.4 การโจมตีแบบผสม (Hybrid)

คือ การโจมตีที่อาศัยวิธีการผสมกันระหว่างสามแบบแรกที่ได้กล่าวมาแล้ว ดังรูปที่ 4-10



รูปที่ 4-10 แสดงแผนภูมิแสดงประเภทของการโจมตีเพื่อให้ปิดบริการสำหรับสแตกทีซีพี/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

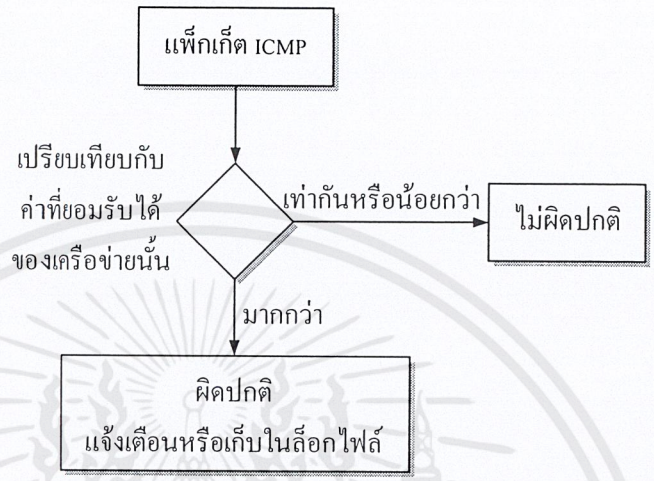
5.3.1 การบุกรุกเพื่อสำรวจระบบ

การบุกรุกเพื่อทำการสำรวจระบบเป็นการกระทำเพื่อเก็บข้อมูลของระบบ เพื่อใช้ในการโจมตี โดยข้อมูลที่ผู้โจมตีมักต้องการได้แก่ หมายเลขไอพีหรือชื่อเครื่อง, โครงสร้างทางเครือข่ายของระบบเป้าหมาย, ชื่อของบริการที่เปิด และระบบปฏิบัติการรวมทั้งเวอร์ชั่นที่ติดตั้งบนเครื่องเป้าหมาย

สำหรับการสำรวจระบบที่ระบบตรวจจับสามารถทำการตรวจจับได้ มีอยู่ 3 วิธีการสำรวจคือ

5.3.1.1 การปิงสวีป

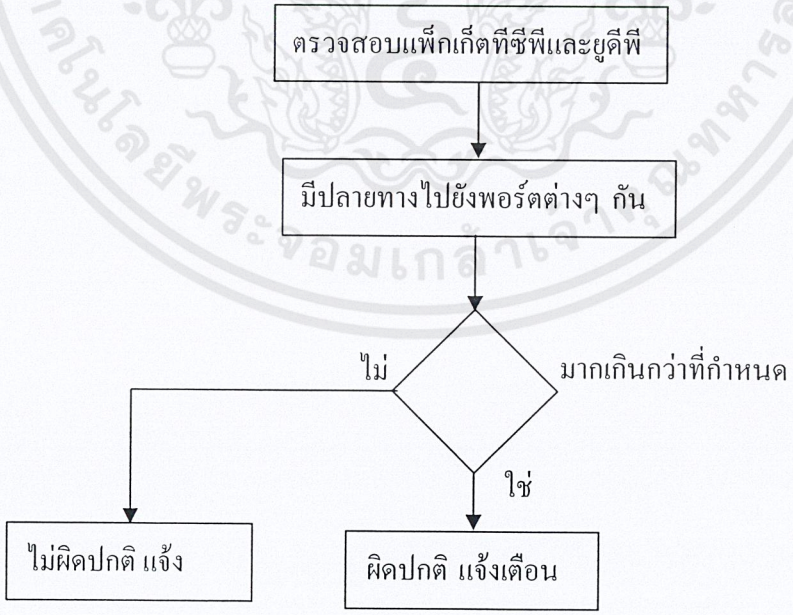
การตรวจจับการปิงสวีสามารถทำได้โดยการตรวจสอบดูแพ็กเก็ต ICMP Request (Type 8) ที่เข้ามายังระบบ หากมีแพ็กเก็ตลักษณะนี้จำนวนมากและมีปลายทางแตกต่างกัน จะสามารถสรุปได้ว่าในเครือข่ายกำลังถูกสำรวจโดยการปิงสวีป



รูปที่ 5-1 แสดงการตรวจสอบการปิงสวีป

5.3.1.2 การสแกนพอร์ต

การตรวจสอบการสแกนพอร์ตทำได้โดยการตรวจดูแพ็กเก็ตที่มีหมายเลขพอร์ตปลายทางในลักษณะกระจาย คือแพ็กเก็ตมีการส่งไปยังเครื่องๆ เดียวแต่มีการส่งไปยังพอร์ตต่างๆ กันเป็นจำนวนมาก

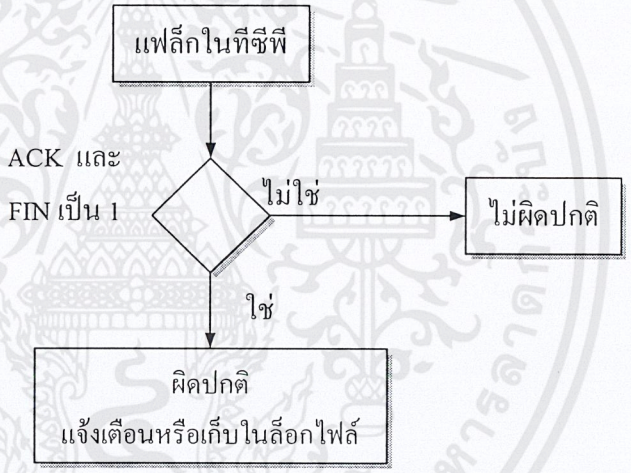


รูปที่ 5-2 แสดงการตรวจสอบการสแกนพอร์ต

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



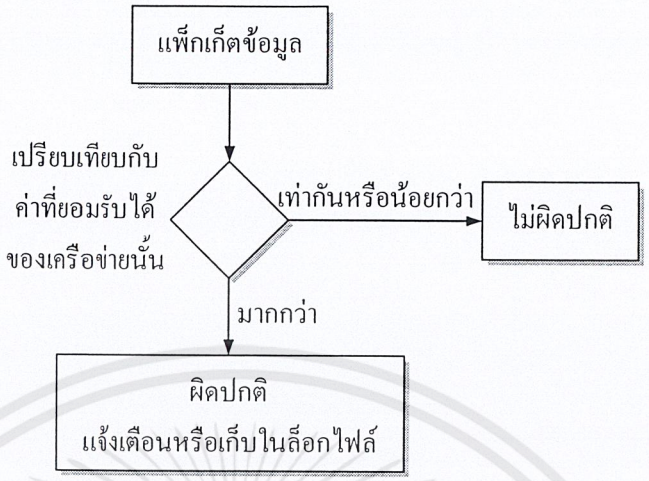
รูปที่ 5-3 แสดงการตรวจสอบการระบุระบบปฏิบัติการ

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

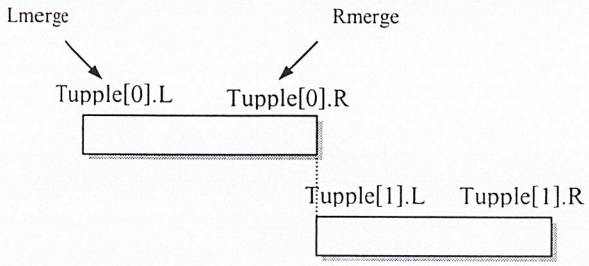
การตรวจสอบความผิดปกติของแฟร็กเมนต์มีขั้นตอนค่อนข้างซับซ้อน ซึ่งแยกอธิบายตามประเภทของความผิดปกติได้ ดังต่อไปนี้

(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

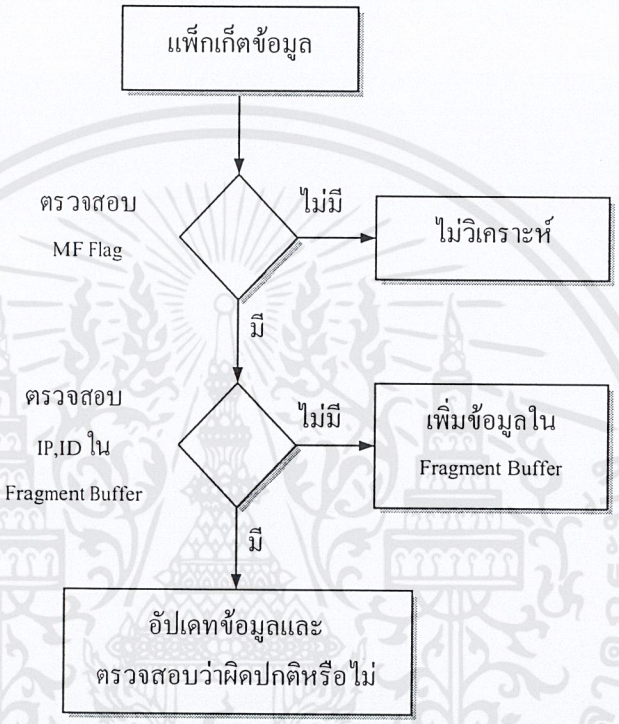
  เป็นบัฟเฟอร์ที่เก็บข้อมูลเมื่อตรวจพบว่าเกิดการเหลื่อมล้ำของแพ็กเก็ต มีการเก็บในลักษณะของลิงส์ลิสต์

- Gap Frame Buffer

  คือ บัฟเฟอร์ที่เก็บข้อมูล เมื่อตรวจพบว่าเกิดการประกอบเฟรมไม่ได้ในลักษณะมีช่องว่างระหว่างแพ็กเก็ต มีการเก็บในลักษณะของลิงส์ลิสต์

ในการวิเคราะห์ใช้บัฟเฟอร์นี้ร่วมกัน โดยเก็บข้อมูลแพ็กเก็ตที่เข้ามาทั้งหมดลงใน Fragment Buffer และหากแพ็กเก็ตที่ส่งมาสามารถรวมกันได้ก็รวมกันเป็นแพ็กเก็ตเดียวที่ต่อเนื่องกัน โดยดูจากขอบซ้ายและขอบขวา

แต่หากรวมกันแล้วเกิดความผิดปกติ ให้แจ้งมายัง Overlap Buffer หรือ Gap Frame Buffer แล้วแต่ความผิดปกติที่เกิดขึ้น

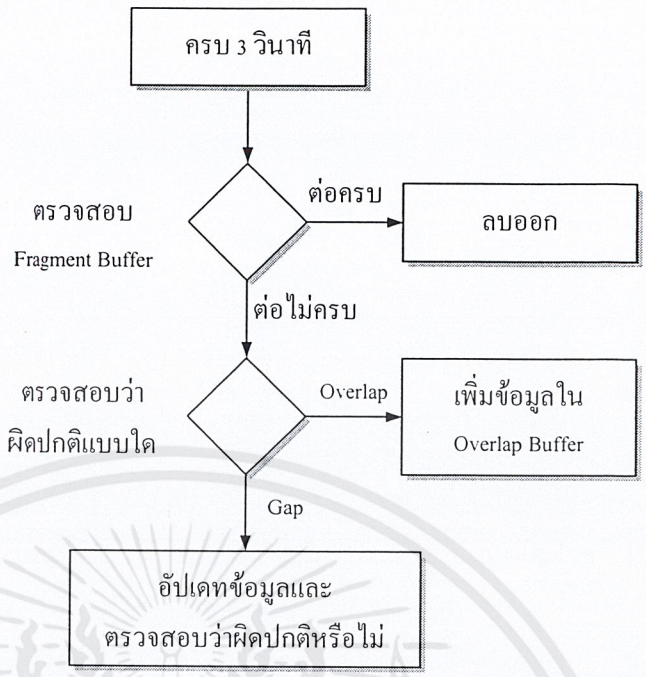


รูปที่ 5-6 แสดงการเก็บข้อมูลลง Fragment Buffer

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

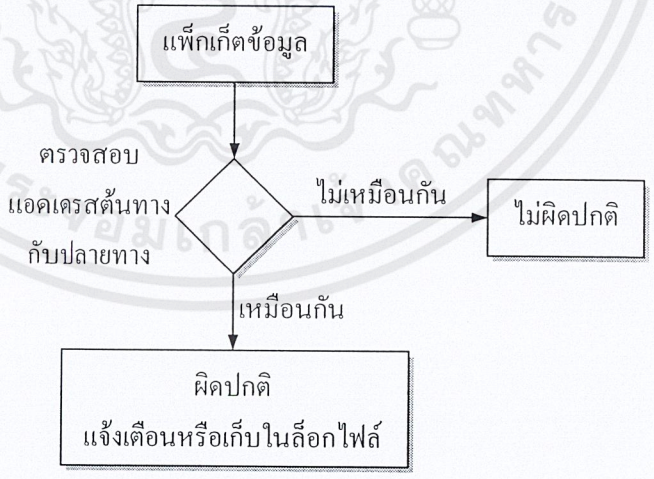
หากไม่มีความผิดปกติใดๆ เกิดขึ้นเลย และแพ็กเก็ตเหล่านั้นสามารถประกอบเป็นเฟรมได้อย่างถูกต้อง ให้ลบเฟรมเหล่านั้นออกจากบัฟเฟอร์ทันที เพื่อไม่ให้สิ้นเปลืองเนื้อที่ในการจัดเก็บ



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

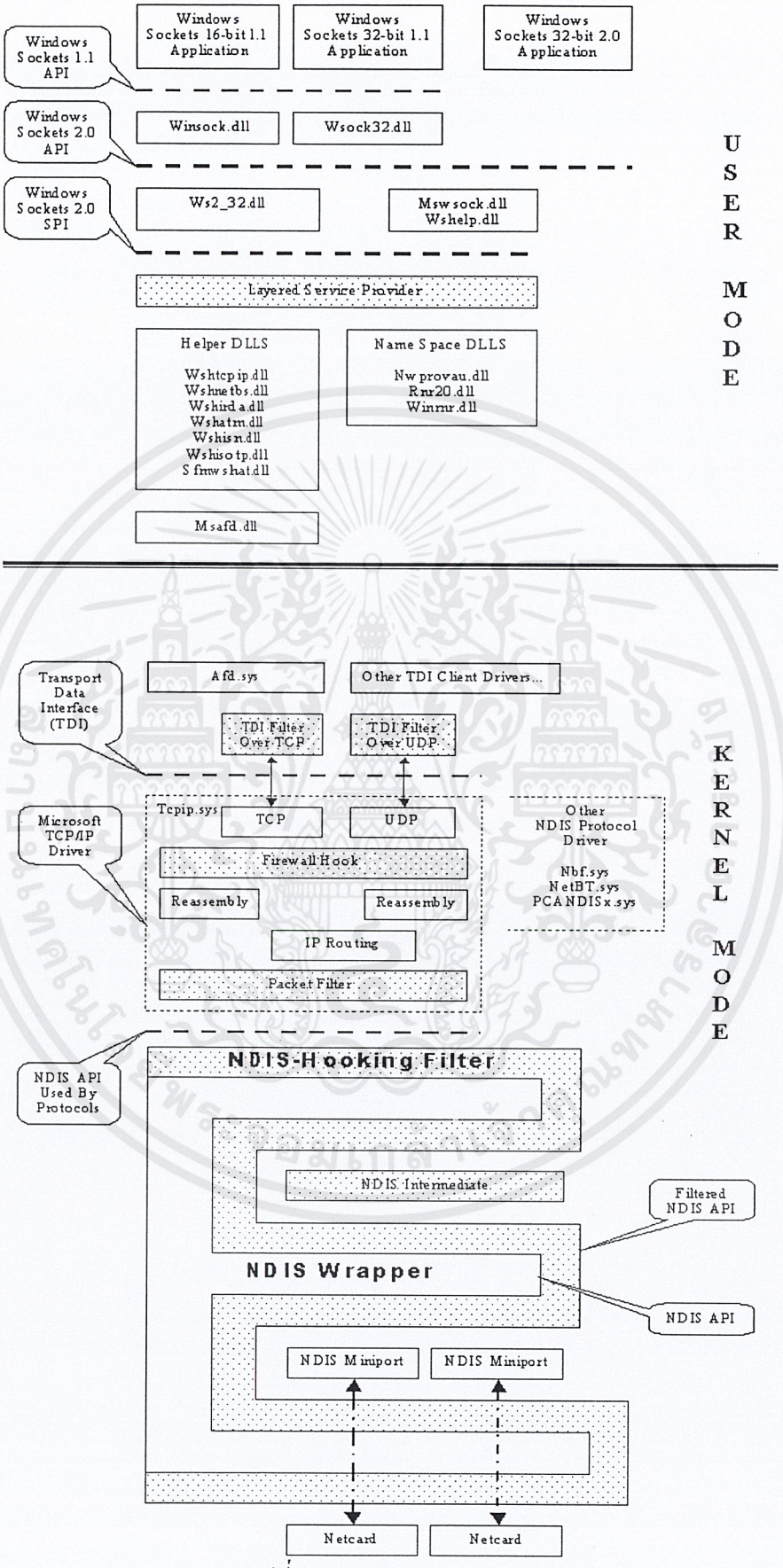
การเขียนโปรแกรมเน็ตเวิร์คบนแพลตฟอร์มวินโดวส์

ก่อนนี้การเขียนโปรแกรมที่เกี่ยวกับเน็ตเวิร์คบนแพลตฟอร์มของวินโดวส์เป็นเรื่องที่ไม่ง่ายนัก เนื่องจากจำเป็นที่จะต้องเขียนไดรเวอร์ เพื่อติดต่อและควบคุมลงไปถึง NIC (Network Interface Card) ซึ่ง NIC แต่ละอันก็จะแตกต่างกันไป ทำให้เป็นการยุ่งยากมากในการเขียนโปรแกรม

ไมโครซอฟต์ได้สร้างมาตรฐาน NDIS (Network Driver Interface Specification) ขึ้นมาซึ่งช่วยทำให้การเขียนโปรแกรมที่เกี่ยวกับเน็ตเวิร์คทำได้อย่างง่ายขึ้น

6.1 Network Driver Interface Specification

NDIS เป็นตัวกำหนดลักษณะการเชื่อมต่อของ NIC Driver , Protocol Driver ซึ่งอาจมีจำนวนมากกว่าหนึ่งไดรเวอร์ต่อหนึ่งระบบปฏิบัติการ โดย NDIS จะทำให้การพัฒนา Network driver เป็นไปได้โดยไม่จำเป็นต้องสนใจในฮาร์ดแวร์ข้างล่างอย่าง NIC ว่าจะเป็นอะไร เปรียบเสมือนเป็นการกำหนด API (Application Programming Interface) มาตรฐานสำหรับการเขียนโปรแกรมติดต่อกับ NIC เนื่องจาก NDIS จะเป็นผู้ที่เป็นตัวเชื่อมระหว่าง Network Driver กับ NIC เอง



รูปที่ 6-1 สถาปัตยกรรมของ NDIS

สำหรับส่วนประกอบหลักๆของ NDIS ก็มีดังต่อไปนี้

6.1.1 NIC Driver คือไดรเวอร์ที่ทำหน้าที่ติดต่อควบคุม NIC ซึ่งเป็นการติดต่อโดยตรงกับฮาร์ดแวร์ ในขณะเดียวกันก็ทำหน้าที่ให้บริการไดรเวอร์ที่อยู่ระดับสูงขึ้นไปในการรับ-ส่งแพ็กเก็ตหรือกระทำการต่างๆ กับ NIC โดย NICไดรเวอร์ มีอยู่ 2 ประเภท

- **Full NIC Driver** จะจัดการส่วนของงานหรือฟังก์ชันที่เจาะจงฮาร์ดแวร์และระบบปฏิบัติการซึ่งปกติเป็นหน้าที่ของ NDIS ซึ่งทำให้ Legacy Driver สามารถเรียกใช้ฟังก์ชันที่เป็นเฉพาะของฮาร์ดแวร์หรือระบบปฏิบัติการนั้นๆได้
- **Miniport Driver** ใช้ฟังก์ชันซึ่งไม่ขึ้นกับแพลตฟอร์มในการจัดการ จะไม่มีการเรียกใช้บริการจากระบบปฏิบัติการโดยตรง

6.1.2 Intermediate Protocol Driver จะอยู่ระหว่าง Legacy Protocol Driver กับ Miniport Driver โดยถ้ามองจาก Transport Driver ด้านบนลงมาจะดูเหมือนเป็น Miniport Driver แต่ถ้ามองจาก Miniport Driver จะดูเหมือน Protocol Driver จุดประสงค์ที่มี Intermediate Protocol Driver นี้สำหรับการแปลง Media Type ระหว่าง Transport Driver กับ Miniport Driver ซึ่งจัดการ Media type แบบใหม่ซึ่งไม่เป็นที่รู้จักของ Transport Driver

6.1.3 Upper Level Protocol Driver ประยุกต์ TDI Interface หรือ Interface เฉพาะอื่นๆเพื่อให้บริการแก่ผู้ใช้ อย่างเช่น การสร้างแพ็คเก็ต การเอาข้อมูลใส่เข้าไปในแพ็คเก็ตและส่งแพ็คเก็ตลงไปสู่ Driver ระดับล่างกว่าโดยเรียกผ่าน NDIS

6.2 การฟิลเตอร์แพ็กเก็ตบนวินโดวส์

สำหรับสถาปัตยกรรมของ NDIS นั้น ก็ได้อนุญาตให้เราสามารถที่จะเขียนโปรแกรมเพื่อเข้าไปฟิลเตอร์แพ็กเก็ตได้หลายวิธีโดยการเรียกใช้ฟังก์ชันการทำงานจากชั้นต่างๆของ NDIS ซึ่งแต่ละวิธีก็มีวิธีการ รูปแบบการใช้งาน และความสามารถในการทำงานที่แตกต่างกันไป มีทั้งในแบบ User-Mode และ Kernel-Mode

6.2.1 User-Mode Network Data Filtering

- Winsock Layered Service Provider (LSP) เป็นไปได้ที่เราสามารถที่จะเรียก Kernel-Mode TCP/IP Driver ผ่าน Transport Data Interface (TDI) โดยไม่จำเป็นต้องผ่าน Winsock แต่ในงานที่จำเป็นจะต้องตรวจสอบหรือกระทำการกับทุกๆแพ็กเก็ตนั้นไม่ควรที่จะอิงการทำงานของ Winsock LSP แต่ควรจะประยุกต์ใช้วิธีการ Kernel-mode มากกว่า

- Windows 2000 Packet Filtering Interface สำหรับวินโดว์ 2000 ได้มี API ที่อนุญาตให้แอปพลิเคชั่นหรือเซอร์วิสที่ทำงานใน User-mode สามารถที่จะกำหนดชุดของ Filter Descriptor ซึ่งจะถูกนำไปใช้โดยคอมโพเนนท์ทีซีพี/ไอพีในชั้นล่าง การฟิลเตอร์นี้จะดูที่หมายเลขไปพีต้นทาง ปลายทาง และหมายเลขพอร์ต โดยสามารถกำหนดได้เพียงแค่อนุญาตให้ผ่านหรือไม่ เท่านั้น

- Winsock Replacement DLL ก่อนที่จะมี LSP วิธีการเดียวที่จะเพิ่มความฟังก์ชันในการทำงานของ Winsock คือการสร้าง DLL ชุดใหม่มาทับ DLL ของ Winsock ซึ่งก็สามารถประยุกต์ใช้ในการฟิลเตอร์แพ็คเก็ตได้ แต่ไม่เป็นที่นิยมเนื่องจากความยากในการทำ อีกทั้งยังต้องยุ่งเกี่ยวกับฟังก์ชันการทำงานที่ไม่มีคู่มืออธิบายอีกด้วย

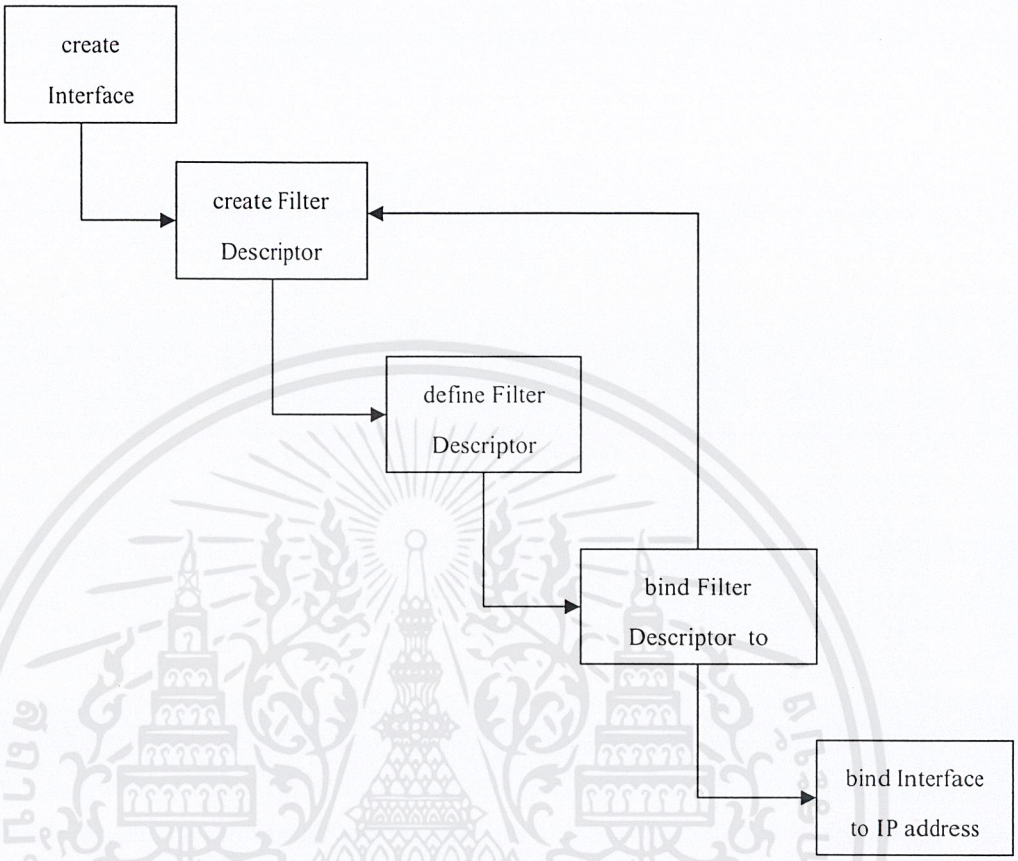
6.2.2 Kernel-Mode Network Data Filtering

- **Transport Data Interface (TDI) Filter Driver** ทำงานในชั้นของ TCP/IP Driver
- **NDIS Intermediate Driver** ทำงานอยู่ระหว่าง Miniport Driver กับ Transport Driver ซึ่งสามารถที่จะประยุกต์ฟิลเตอร์แพ็คเก็ตในชั้นนี้ได้เช่นกัน
- **Windows 2000 Filter-Hook Driver** เป็นตัวที่ขยายขอบข่ายความสามารถออกมาจาก IP Filter Driver ซึ่งมากับระบบปฏิบัติการ
- **NDIS Hooking Filter Driver**

6.3 Windows 2000 Packet Filter API

จากการที่มีไลบรารีชุด iphlpapi.dll เข้ามาในวินโดวส์ 98 และวินโดวส์ 2000 นั้นได้ช่วยการทำงานที่เกี่ยวกับเน็ตเวิร์คให้ง่ายขึ้นหลายอย่าง ในอดีต แม้แต่การเรียกค่าหมายเลขไอพีของการ์ดเน็ตเวิร์คของเครื่องยังเป็นเรื่องที่ยุ่งยาก iphlpapi.dll (IP Helper API) จึงถูกสร้างขึ้นเพื่อแก้ไขข้อจำกัดเหล่านี้ เราสามารถใช้ API ตัวนี้ในการเรียกดูหมายเลขไอพี, routing table, ARP table entries, และสถิติทางเน็ตเวิร์ค โดยที่ในวินโดวส์ 98 จะสนับสนุนเพียงแค่บางส่วนของ iphlpapi.dll ส่วนวินโดวส์ 2000 นั้นสนับสนุนทั้งหมด โดย iphlpapi.dll นั้นเป็นส่วนหนึ่งของ Microsoft Platform SDK

หนึ่งใน interface ของ iphlpapi.dll นี้คือ Packet Filtering interface ซึ่งอยู่ภายใต้ Routing and Remote Access services ซึ่งมีในทุกเวอร์ชันของวินโดวส์ 2000 ซึ่ง interface นี้มีความสามารถในการฟิลเตอร์แพ็คเก็ต TCP/IP ซึ่งทำให้เราสามารถที่จะระงับการติดต่อกับคอมพิวเตอร์บางเครื่องหรือบริการบางอย่างได้ ซึ่งก่อนหน้านี้ การจะทำเช่นนี้ได้อย่างน้อยจะต้องเขียนไดรเวอร์ในชั้นของ NDIS ขึ้นมา แต่เมื่อมี API ตัวนี้แล้ว สามารถทำได้เพียงแค่เรียกฟังก์ชันเท่านั้น



รูปที่ 6-2 ขั้นตอนการทำงานของ Packet Filter API

ในการกำหนดกฎการฟิลเตอร์และผูกเข้ากับการ์ดเน็ตเวิร์ค จะต้องมี INTERFACE_HANDLE ก่อน ซึ่งสามารถได้จากฟังก์ชัน PfCreateInterface() ซึ่งฟังก์ชันนี้ยังใช้ในการกำหนดค่าดีฟอลต์ในการฟิลเตอร์แพ็คเก็ตสำหรับแพ็คเก็ต TCP/IP ขาเข้าและขาออกอีกด้วย เราสามารถกำหนดให้ interface ที่เราสร้างขึ้นมาบล็อคแพ็คเก็ตทั้งหมดยกเว้นเฉพาะแพ็คเก็ตที่ตรงกับข้อกำหนดของกฎการฟิลเตอร์ (PF_ACTION_DROP) หรือกำหนดให้แพ็คเก็ตทั้งหมดผ่านไปได้ยกเว้นเฉพาะแพ็คเก็ตที่ตรงกับข้อกำหนดของกฎการฟิลเตอร์ (PF_ACTION_FORWARD) เรายังสามารถกำหนดได้ว่า interface ที่สร้างขึ้นนั้นจะสามารถแชร์ได้หรือไม่ ซึ่งการแชร์ได้หมายถึงการที่อนุญาตให้โพรเซสอื่นๆสามารถเพิ่มหรือลบกฎการฟิลเตอร์ใน interface นี้ได้

หลังจากที่เราได้ handle จากฟังก์ชัน PfCreateInterface() แล้ว เราสามารถที่จะเพิ่มกฎการฟิลเตอร์เข้าไปได้โดยผ่านฟังก์ชัน PfAddFiltersToInterface() ซึ่งสามารถกำหนดได้ทั้งการฟิลเตอร์ทั้งขาเข้าและขาออก พารามิเตอร์หลักสำหรับฟังก์ชันนี้คือ PF_FILTER_DESCRIPTOR ซึ่งจะได้กล่าวถึงต่อไป หลังจากที่ได้กำหนดกฎการฟิลเตอร์ครบทั้งหมดแล้ว ขั้นตอนต่อไปคือการผูก interface handle เข้ากับการ์ดเน็ตเวิร์ค โดยใช้หมายเลขไอพี การฟิลเตอร์จะทำงานและมีผลจนกว่าจะมีการเรียกฟังก์ชัน PfDeleteInterface()

PF_FILTER_DESCRIPTOR

Filter Descriptor นั้นเป็นรูปแบบโครงสร้างของกฎการฟิลเตอร์ ในการที่เราจะกำหนดกฎแต่ละข้อ ซึ่งในขั้นตอนแรกนั้น เราจะต้องสร้าง Filter Descriptor ขึ้นมาและกำหนดค่าต่างๆ โดยโครงสร้างของ PF_FILTER_DESCRIPTOR นั้นเป็นดังต่อไปนี้

```
typedef struct PF_FILTER_DESCRIPTOR {
  DWORD           dwFilterFlags;
  DWORD           dwRule;
  PFADDRESSTYPE   pfatType;
  PBYTE           SrcAddr;
  PBYTE           SrcMask;
  PBYTE           DstAddr;
  PBYTE           DstMask;
  DWORD           dwProtocol;
  DWORD           fLateBound;
  WORD            wSrcPort;
  WORD            wDstPort;
  WORD            wSrcPortHighRange;
  WORD            wDstPortHighRange;
} PF_FILTER_DESCRIPTOR, *PPF_FILTER_DESCIPTOR;
```

โดยมีรายละเอียดของแต่ละฟิลด์ดังนี้

dwFilterFlags ณ ขณะนี้มีเพียงแค่ค่าเดียวที่ถูกกำหนดขึ้นสำหรับฟิลด์นี้ คือ FD_FLAGS_NOSYN โดยเมื่อแอปพลิเคชั่นต้องการที่จะสร้างการเชื่อมต่อแบบ TCP จะต้องมีการส่ง synchronization request ซึ่ง flag นี้เป็นการห้ามไม่ให้ทำเช่นนั้น แต่จากการทดลองพบว่าไม่มีความแตกต่างในการกำหนดหรือไม่กำหนดค่าให้สำหรับ flag นี้

dwRule เป็นค่าที่ผู้ใช้กำหนดขึ้นซึ่งจะถูกสำเนาไปสู่ log (ถ้ามีการใช้งาน log) เพื่อแสดงว่ากฎการฟิลเตอร์ข้อไหนที่รับผิดชอบสำหรับแต่ละข้อของ log

pfatType กำหนดเวอร์ชันของไอพี ซึ่งสามารถกำหนดได้เป็น PF_IPV4 (IP เวอร์ชัน 4) หรือ PF_IPV6 (IP เวอร์ชัน 6)

SrcAddr หมายเลขไอพีต้นทาง

SrcMask เน็ตมาสก์ต้นทาง

DstAddr หมายเลขไอพีปลายทาง

DstMask เน็ตมาสก์ปลายทาง

หมายเลขไอพีต้นทาง ปลายทาง รวมถึงมาสก์ เป็นการกำหนดช่วงของหมายเลขไอพีที่กฎการฟิลเตอร์มีผล โดยมาสก์เป็นตัวกำหนดขอบเขตของแอดเดรสที่จะถูกทดสอบ และเมื่อแอดเดรสถูกกำหนดเป็นช่วง (โดยการกำหนดส่วนของมาสก์เป็นศูนย์) ส่วนของแอดเดรสที่สัมพันธ์กันก็ต้องถูกกำหนดให้เป็นศูนย์ด้วยเช่นกัน ตัวอย่างเช่น ถ้าต้องการกำหนดหมายเลขไอพีหมายเลขเดียว แอดเดรสจะต้องสมบูรณ์และค่ามาสก์ก็จะต้องเป็น FF.FF.FF.FF ในทางกลับกัน ถ้าต้องการจะกำหนดฟิลเตอร์สำหรับช่วงของ

หมายเลขไอพี ก็ต้องกำหนดส่วนที่เป็นช่วงของแอดเดรสให้เป็นศูนย์เช่นเดียวกันกับส่วนเดียวกันสำหรับมาสก์ (แอดเดรส a.b.c.0 และมาสก์ FF.FF.FF.00 จะฟิลเตอร์หมายเลขไอพีทั้งหมดที่ขึ้นต้นด้วย a.b.c) ในการกำหนดฟิลเตอร์ขาออก จะต้องกำหนดให้แอดเดรสของตัวเองเป็นแอดเดรสต้นทาง ส่วนการฟิลเตอร์ขาเข้าจะต้องกำหนดแอดเดรสของฝั่งรีโมทเป็นต้นทาง

dwProtocol กำหนดโพรโตคอลไอพีที่จะต้องฟิลเตอร์ ซึ่งมี ICMP, TCP, UDP หรือทั้งหมด

fLateBound ใช้ในการกำหนดข้อมูลเกี่ยวกับแอดเดรสที่ต้องการถูกปรับปรุงเมื่อมีการ rebound ฟิลเตอร์ (เช่นเมื่อมีการเชื่อมต่อใหม่ของ dial-up adapter) สำหรับการใช้งานของฟิลด์นี้ยังเป็นที่สงสัยอยู่

wSrcPort หมายเลขพอร์ตต้นทาง

wSrcPortHighRange ขอบบนของหมายเลขพอร์ตต้นทาง

wDstPort หมายเลขพอร์ตปลายทาง

wDstPortHighRange ขอบบนของหมายเลขพอร์ตปลายทาง

การกำหนดพอร์ตในการฟิลเตอร์นั้น ถ้ากำหนดหมายเลขพอร์ตเป็นศูนย์จะหมายถึงทุกพอร์ต แต่ถ้ากำหนดหมายเลขพอร์ตไม่เลขที่ไม่ใช่ศูนย์แล้วจะต้องกำหนด PortHighRange ด้วย โดยถ้าต้องการระบุพอร์ตใดพอร์ตหนึ่งก็ให้กำหนดทั้งสองฟิลด์ให้เป็นค่าเดียวกัน แต่ถ้าต้องการกำหนดเป็นช่วงของพอร์ตก็ให้กำหนด PortHighRange ให้เป็นค่าช่วงของหมายเลขพอร์ต กล่าวคือ กำหนดค่าพอร์ตให้เป็นหมายเลขพอร์ตต่ำสุดที่ต้องการฟิลเตอร์ และ PortHighRange เป็นค่าสูงสุดของพอร์ตที่ต้องการฟิลเตอร์ เช่น ถ้าต้องการให้มีการฟิลเตอร์ตั้งแต่พอร์ต 21 ถึงพอร์ต 48 ก็ให้กำหนด Port เป็น 21 และ PortHighRange เป็น 48 เป็นต้น

6.4 การเขียนโปรแกรมเน็ตเวิร์กโดยใช้ Winsock

ในการเขียนโปรแกรมเน็ตเวิร์กระหว่างเซิร์ฟเวอร์กับไคลเอนต์ด้วยการใช้โปรโตคอลทีซีพี/ไอพี จะนิยมใช้ Windows Socket API หรือเรียกอีกอย่างว่า Winsock ซึ่ง Winsock จะอยู่ในไลบรารี่ของ MFC ซึ่งมีอยู่ 2 คลาส คือ CAsyncSocket และ CSocket แต่ในที่นี้จะใช้คลาส CAsyncSocket เป็นหลัก คลาสนี้ห้อมล้อมด้วย Windows Socket API ที่ระดับต่ำมาก (very low level)

- **ซ็อคเก็ต**

ซ็อคเก็ตถูกกำหนดหรือนิยามไว้ว่า เป็นคู่ของการสื่อสาร หรือคู่ของโพรเซส (หรือเธรด) โดยที่การสื่อสารบนเน็ตเวิร์คใช้คู่ของซ็อกเก็ตสำหรับแต่ละโพรเซส

สำหรับซ็อคเก็ตประกอบไปด้วย IP Address กับหมายเลข Port (Port Number) โดยทั่วๆ ไปแล้วซ็อกเก็ตใช้สถาปัตยกรรมไคลเอนต์เซิร์ฟเวอร์ เซิร์ฟเวอร์จะรอการเข้ามาตามการขอร้องของไคลเอนต์โดยการฟังที่พอร์ตเฉพาะ เมื่อการขอร้องได้รับ เซิร์ฟเวอร์ก็จะยอมรับการเชื่อมต่อจากซ็อกเก็ตไคลเอ็นต์เพื่อให้สมบูรณ์ในการเชื่อมต่อ

ตัวอย่างการสื่อสารด้วยซ็อกเก็ต เมื่อเธรดไคลเอนต์เริ่มต้นการขอร้องสำหรับการเชื่อมต่อ จะถูกกำหนดพอร์ตโดยโฮสต์คอมพิวเตอร์ พอร์ตนี้เป็นหมายเลขใดก็ได้ที่มากกว่า 1024 ตัวอย่างเช่น ถ้าไคลเอนต์บนโฮสต์ A มี IP Address 161.246.5.24 ต้องการที่จะสร้างการเชื่อมต่อกับเซิร์ฟเวอร์ http ที่มี IP

Address 161.246.5.25 โฮสต์ A จะถูกกำหนดพอร์ตเป็น 2545 และที่ฝั่งเซิร์ฟเวอร์จะเป็นพอร์ต 80 แพ็กเก็ตที่ไปมาระหว่างโฮสต์ทั้งสองจะถูกส่งไปยังเธรดที่เหมาะสม ซึ่งขึ้นอยู่กับหมายเลขพอร์ตปลายทางด้วย

การเชื่อมต่อทั้งหมดเป็นคุณสมบัติเฉพาะ ดังนั้นถ้าโพรเซสอื่นๆ บนโฮสต์ A ต้องการสร้างการเชื่อมต่ออื่นๆ กับเซิร์ฟเวอร์ http เดียวกัน เซิร์ฟเวอร์จะกำหนดหมายเลขพอร์ตที่มากกว่า 1024 การทำอย่างนี้เพื่อให้แน่ใจว่าการเชื่อมต่อทั้งหมดประกอบด้วยคู่ที่เป็นยูนิค (Unique) ของซ็อกเก็ตหรือเป็นสิ่งที่ไม่ซ้ำกับการเชื่อมต่ออื่นๆ ของซ็อกเก็ต

โดยปกติแล้วเซิร์ฟเวอร์จะมีหลายๆ การขอร้องที่พร้อมกัน จะต้องใช้ระยะเวลาหนึ่งที่ไคลเอ็นต์ต้องรอคอยเพื่อที่จะถูกบริการโดยเซิร์ฟเวอร์เธรดเดียว ซึ่งจะไม่สามารถรับได้

เพื่อแก้ไขสถานการณ์นี้เซิร์ฟเวอร์ต้องจัดการการขอร้องที่พร้อมๆ กัน โดยการกำหนดเธรดแยกออกมาเพื่อบริการแต่ละการขอร้องที่เข้ามา ตัวอย่างเช่น เซิร์ฟเวอร์ http ที่ไม่ว่างจะกำหนดเธรดแยกออกมาเพื่อบริการแต่ละการขอร้องสำหรับเว็บเพจ

ชนิดของซ็อกเก็ต

ชนิดของซ็อกเก็ตมีอยู่สามชนิด คือ

Connection-Orianted Socket เป็นซ็อกเก็ตการเชื่อมต่อแบบต่อเนื่องที่อนุญาตให้โพรเซสเชื่อมต่อกับโพรเซสระยะไกล (Remote) ซึ่งใช้โปรโตคอล TCP (Transmission Control Protocol) ดังนั้นด้วยวิธีการนี้ทำให้ข้อมูลเชื่อถือได้ เมื่อการเชื่อมต่อได้เกิดขึ้นโพรเซสก็จะมีการส่งข้อมูลกลับไปจนกระทั่งฝั่งใดฝั่งหนึ่งหรืออื่นๆ มีการปิดการเชื่อมต่อ ชนิดของซ็อกเก็ตนี้บางครั้งเรียกว่า สตรีมซ็อกเก็ต (Stream Socket) ทั้ง ftp และ http ใช้ซ็อกเก็ตแบบนี้ในการสื่อสาร

Connectionless Socket หรือเรียกอีกอย่างว่า ดาต้าแกรมเป็นซ็อกเก็ตแบบไม่ต่อเนื่อง และนำมาใช้เป็นประโยชน์ในการส่งเมสเสจสั้นๆ ซึ่งไม่สามารถสนับสนุนส่วนหัว ดังนั้นจึงพิจารณาการเชื่อมต่อประเภทนี้เป็นแบบเชื่อถือไม่ได้ซึ่งก็คือ การไม่รับประกันข้อมูลที่ถูกส่งออกไป ไม่เหมือนกับซ็อกเก็ตการเชื่อมต่อแบบต่อเนื่องที่ซ็อกเก็ตปลายทางถูกตรวจสอบเมื่อแพ็กเก็ตถูกส่งออกไป

ซ็อกเก็ตแบบไม่ต่อเนื่อง เปรียบเสมือนกับการบริการของไปรษณีย์ที่ผู้ส่งจดหมายไปจามที่อยู่แล้วใส่ในกล่องรับจดหมาย ผู้ส่งจะไม่ทราบว่าผู้รับได้รับจดหมายหรือไม่ ซ็อกเก็ตแบบนี้นิยมใช้กันในเซิร์ฟเวอร์ DNS ที่ใช้ซ็อกเก็ตดาต้าแกรมในการตอบสนองต่อการขอร้องที่เข้ามามากๆ

Raw Socket เป็นซ็อกเก็ตที่อนุญาตให้เข้าถึงโปรโตคอล Transport Raw Socket ยังสามารถนำมาใช้เพื่อจัดการข้อมูลส่วนหัว IP (IP Header) นอกจากนี้แล้วการใช้ซ็อกเก็ตชนิดนี้ ต้องการความรู้อย่างมากของโครงสร้างโปรโตคอลพื้นฐาน

- **Winsock Programming**

วินซ็อก (WinSock) เป็นมาตรฐานเปิดเน็ตเวิร์ค API โดยที่วินซ็อกถูกออกแบบมาครั้งแรก เพื่อสร้างการโปรแกรมอินเตอร์เฟส ที่เป็นมาตรฐานสำหรับ TCP/IP ในทุกเวอร์ชันของระบบปฏิบัติการ

วินโดวส์รวมทั้ง Windows 2000, Windows NT, Windows 98 ซึ่งจะเป็นวินซ็อกเวอร์ชัน 2.2 แต่ถ้าเป็นระบบปฏิบัติการวินโดวส์รุ่นดั่งเดิม เช่น Windows 95 และ Windows CE นั้นจะใช้วินซ็อกเวอร์ชัน 1.1

มีสองเหตุผลในการใช้วินซ็อก คือ การคอนโทรล และ ความมีประสิทธิภาพ

วินซ็อกเป็นเน็ตเวิร์คแอพพลิเคชันโปรแกรมมิ่งอินเตอร์เฟส ไม่ใช่โปรโตคอล ซึ่งวินซ็อกนั้นมีรากฐานเดียวกับซ็อกเก็ตของยูนิกซ์ตระกูล BSD (Berkeley Software Distribution) เวอร์ชัน 4.3 รายละเอียดได้รวมทั้งรูทีนซ็อกเก็ตสไตล์ BSD และการขยายสเปคมาใช้กับวินโดวส์

การใช้วินซ็อกอนุญาตให้แอพพลิเคชันของเรา ทำการติดต่อสื่อสารข้ามเน็ตเวิร์คใดๆ ก็ได้ที่กระทำกับ WinSock API แพลตฟอร์ม Win 32 ซึ่งวินซ็อกได้ให้เธรดที่ปลอดภัย (thread safety)

คลาส MFC สนับสนุนการโปรแกรม WinSock API โดยการใช้คลาส CAsyncSocket และ CSocket คลาส CAsyncSocket ห้อมล้อมด้วยวินโดวส์ซ็อกเก็ต API ที่ระดับล่าง ซึ่งมีความรู้เกี่ยวกับการสื่อสารข้อมูลผ่านเน็ตเวิร์ค แต่ถ้าเราต้องการอินเตอร์เฟสที่ง่ายกว่าคลาสนี้แล้วควรจะใช้คลาส CSocket

คลาส CSocket สืบทอดจากคลาส CAsysncSocket และรวมทั้งสืบทอดการห้อมล้อมของวินซ็อก API ออบเจ็กต์C Socket แสดงในระดับสูง กว่าออบเจ็กต์ CAsyncSocket ออบเจ็กต์ CSocket ทำงานร่วมกับคลาส CSocketFile และคลาส CArchive เพื่อจัดการการส่งและการรับข้อมูล

บทที่ 7
การเขียนโปรแกรมเน็ตเวิร์คบนแพลตฟอร์มลีนุกซ์

ซ็อกเก็ตเป็นกลไกที่ให้โปรแกรมเมอร์ สามารถแอ็คเซสเน็ตเวิร์คโพรโตคอล เพื่อให้โปรเซสสามารถติดต่อกันแบบโลคัลหรือผ่านเน็ทเวิร์คได้ (Network Application Programming Interface) ยูทิลิตี้ของยูนิกซ์ที่ทำงานผ่านเน็ตเวิร์ค ปกติก็จะใช้ซ็อกเก็ตเช่น telnet หรือ ftp

แนวทางการสร้างและใช้งานซ็อกเก็ตจะแตกต่างกับไปป์ เนื่องจากมีการระบุโปรเซสไคลเอนต์และเซิร์ฟเวอร์อย่างชัดเจน นอกจากนี้สำหรับโปรเซสเซิร์ฟเวอร์เพียงโปรเซสเดียว ปกติแล้วสามารถ ที่จะทำงานกับโปรเซสไคลเอ็นต์หลายๆ โปรเซสได้

7.1 หลักการเบื้องต้นการติดต่อโดยใช้ซ็อกเก็ต(Socket Connection Oriented)

7.1.1 การสร้างซ็อกเก็ตให้แก่ฝั่งเซิร์ฟเวอร์

1. สิ่งแรกคือเซิร์ฟเวอร์จะต้องสร้างซ็อกเก็ตซึ่งก็ถือเป็นรีซอร์สของระบบอย่างหนึ่งที่ควบคุมโดยโปรเซสเซิร์ฟเวอร์ โดยใช้ฟังชั่น socket()
2. ลำดับต่อไปคือการกำหนดชื่อให้แก่ซ็อกเก็ตโดยปกติโลคัลซ็อกเก็ตจะเป็นชื่อไฟล์กำหนดไว้ใน /tmp แต่ถ้าเป็นเน็ตเวิร์คซ็อกเก็ตจะประกอบด้วยหมายเลขพอร์ตและจุดเชื่อมต่อของโปรเซส จะเป็นค่าเฉพาะสำหรับเน็ตเวิร์คนั้น การกำหนดชื่อให้แก่โปรเซสทำได้โดยใช้ฟังชั่น bind()
3. จากนั้นโปรเซสเซิร์ฟเวอร์จะรอรับการเชื่อมต่อจากโปรเซสไคลเอนต์โดยใช้ซ็อกเก็ตที่กำหนดชื่อไปข้างต้น โดยใช้ฟังชั่น listen() เพื่อสร้างคิวรอรับข้อมูลการเชื่อมต่อ
4. หลังจากนั้นถ้ามีไคลเอนต์โปรเซสติดต่อเข้ามาโปรเซสเซิร์ฟเวอร์จะตอบรับการเชื่อมต่อด้วยฟังชั่น accept()

7.1.2 การสร้างซ็อกเก็ตให้แก่ฝั่งไคลเอนต์

1. สำหรับ บนฝั่งไคลเอนต์จะสร้างได้ง่ายกว่านั้นคือใช้ฟังชั่น socket() เพื่อจองรีซอร์สของระบบ
2. จากนั้นเรียกฟังชั่น connect() เพื่อเชื่อมต่อกับเซิร์ฟเวอร์โปรเซส

หลังจากสร้างช่องทางการเชื่อม ต่อได้แล้ว เราสามารถใช้หมายเลขแทนช่องทางการเชื่อมต่อนั้นๆ (คล้ายๆกับ File descriptor) โดยสามารถใช้ช่องทางการเชื่อมต่อข้อมูลระหว่างโปรเซสได้ทั้ง 2 ทิศทาง (Full duplex) องค์ ประกอบของซ็อกเก็ต

ต่อไปจะมาดูองค์ประกอบของเน็ทเวิร์คบนระบบยูนิกซ์ ซึ่งก็เป็นส่วนประกอบที่กำหนดไว้อยู่ในซ็อกเก็ตนั่นเอง จะประกอบด้วย 3 ส่วนด้วยกันคือ รูปแบบการกำหนดแอดเดรสของคอมพิวเตอร์บนเน็ตเวิร์ค (Domain) ชนิดของซ็อกเก็ต (Socket type) และโพรโตคอลที่ใช้งาน

- **Socket Domain**

เป็นตระกูลของโพรโตคอลที่ใช้งาน ที่ใช้งานในที่นี้คือ AF_INET หมายถึงโพรโตคอลใช้งานกับอินเตอร์เน็ต TCP/IP ระบบจะใช้โพรโตคอล IP กำหนดแอดเดรสในการเชื่อมต่อ โดยมีรูปแบบของแอดเดรสโดยเฉพาะประกอบด้วยตัวเลข 4 จำนวน คั่นด้วยจุด และค่าจะน้อยกว่า 256 ซึ่งจะเป็นแอดเดรส

จริงของคอมพิวเตอร์บนอินเทอร์เน็ต เมื่อไคลเอนต์ ติดต่อกับเน็ตเวิร์คเซิร์ฟเวอร์ผ่านซ็อกเก็ตจะต้องทราบ IP Address ของเซิร์ฟเวอร์นั้นจึงจะสามารถติดต่อได้ สำหรับการกำหนดชนิดของการบริการจากเซิร์ฟเวอร์ของอินเตอร์เน็ทจะกำหนดโดยใช้หมายเลขพอร์ต (IP Port) ซึ่งเป็นหมายเลขขนาด 16 บิต (ซ็อกเก็ต TCP/UDP จะต้องได้รับการกำหนดพอร์ตก่อนจึงจะสามารถใช้ช่องทางการสื่อสารได้) โปรเซสเซิร์ฟเวอร์จะรอการติดต่อที่เฉพาะพอร์ต บนระบบยูนิกซ์จะกำหนดให้มีค่าตั้งแต่ 1-1023 ซึ่งจะเป็ยพอร์ตที่ใช้งานสำหรับโปรเซสระบบเท่านั้นปกติต้องมีสิทธิ์เป็นผู้ดูแลระบบเท่านั้นจึงจะสามารถจองพอร์ตเหล่านี้ไว้ใช้งานได้

สำหรับ Unix file system domain (AF_UNIX) ใช้สำหรับซ็อกเก็ตในการติดต่อกับโพรเซสที่อยู่บนเครื่องเดียวกันเท่านั้น โพรโตคอลที่ใช้ในการติดต่อกันก็คือชื่อของไฟล์นั่นเอง ส่วนแอดเดรสที่ใช้ในการติดต่อคือ ชื่อของไฟล์แบบ Absolute filename

โดเมนอื่นที่อาจจะใช้งานคือ AF_ISO สำหรับเน็ตเวิร์คที่ใช้โพรโตคอลของ ISO หรือ AF_NS เป็นโพรโตคอลของ Xerox network system

- **Socket Type**

แต่ละโดเมนอาจจะมีซ็อกเก็ตหลายรูปแบบ ซึ่งหมายถึงวิธีการส่งข้อมูลของแต่ละโดเมนไม่เพียงแต่ AF_UNIX ซึ่งเราสามารถใช้การติดต่อได้ทั้ง 2 ทางเท่านั้น โดเมนของเน็ทเวิร์ค AF_INET ก็จะมีวิธีการส่งข้อมูล 2 แบบคือ Streams (TCP) และ Datagrams (UDP)

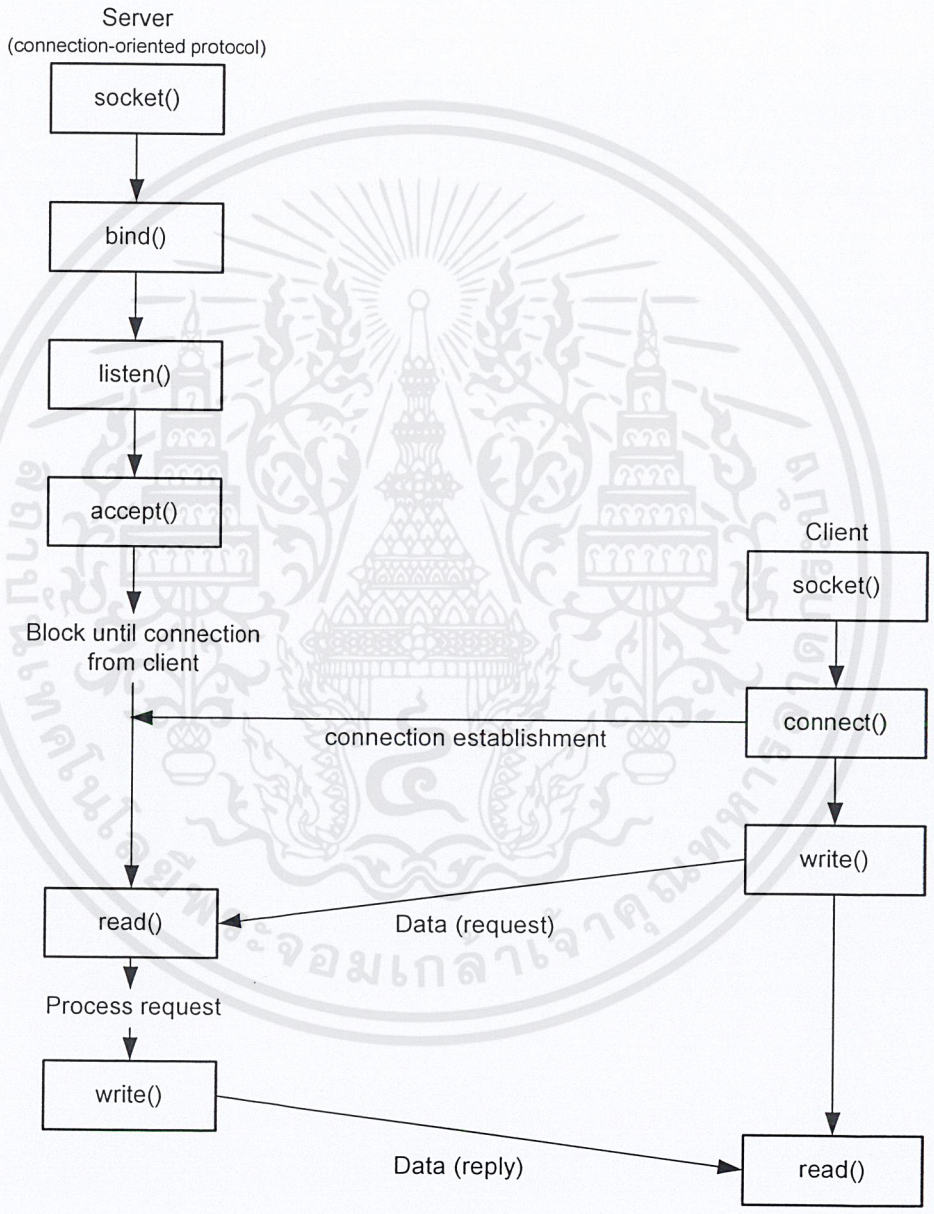
วิธีการ Streams (TCP) จะมีความน่าเชื่อถือของการส่งข้อมูลมากกว่า จะสร้างช่องทางการเชื่อมต่อจนกระทั้งการเชื่อมต่อเสร็จสิ้น (Connection-oriented) รวมทั้งสามารถจะติดต่อกันได้ทั้ง 2 ทาง ข้อมูลที่ถูกส่งไปจะได้รับการรับรองติดตามผลการส่งว่าส่งถึงหรือไม่ ถ้ามีปัญหา จะต้องส่งใหม่ การใช้จะกำหนดโดย SOCKET_STREAM ในโดเมน AF_INTET

ส่วนอีกวิธีการหนึ่งคือ Datagrams กำหนดโดยใช้ SOCK_DGRAM จะไม่สร้างช่องทางการเชื่อมต่อ(Connectionless) หรือแม้แต่การจัดการลำดับของข้อมูลที่จะได้รับ ส่งเสร็จแล้วก็ไม่มีการติดตามผลการส่งข้อมูล ไม่มีการตรวจสอบใดๆทั้งสิ้น อย่างไรก็ตามการใช้งานรีซอร์ส จะไม่เปลืองระบบมากนักเนื่องจากไม่ต้องจองช่องทางการส่งไว้ใช้งาน โดยทั่วไปจะใช้ส่งข้อมูลบางอย่างที่ต้องทำเป็นประจำ (Single-short inquiries)

- **Socket Protocols**

หลังจากที่ได้กำหนดโดเมนของเน็ตเวิร์คโพรโตคอลแล้ว ก็ต้องเลือกโพรโตคอลที่จะใช้งานจริง ซึ่งจะมีความจำเพาะต่อโดเมนโพรโตคอลและวิธีการส่งข้อมูลที่ใช้งาน ดังที่ได้อธิบายไว้แล้วตามตารางจะแสดงเฉพาะโดเมนของ AF_INET

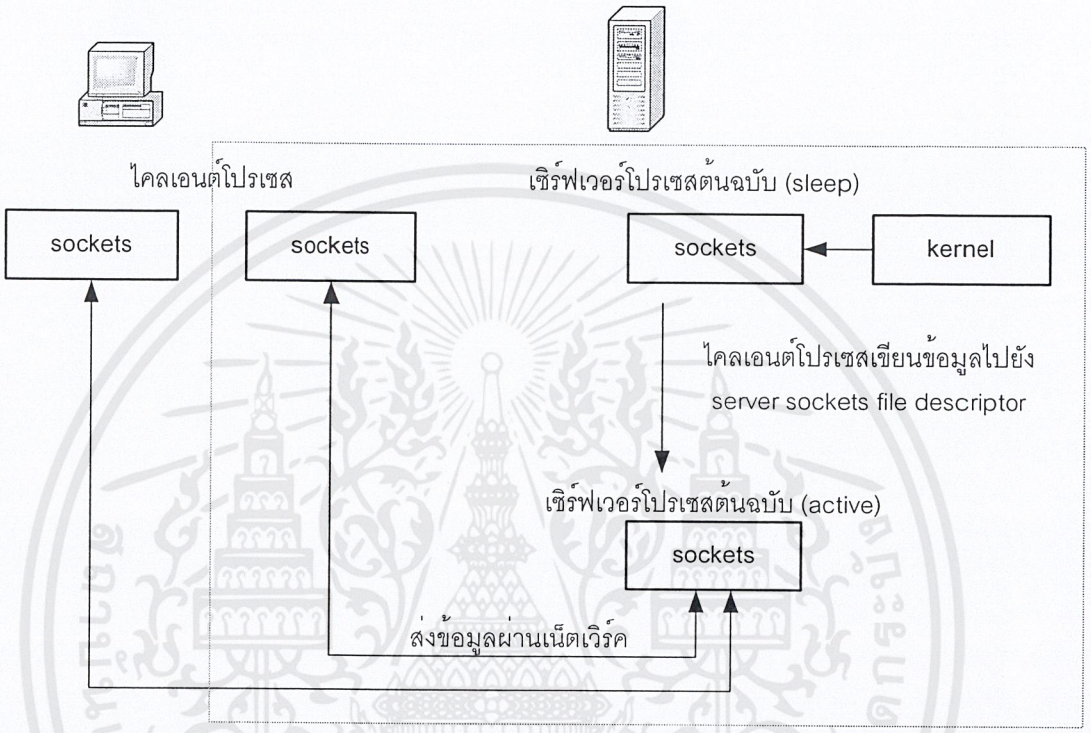
โดเมน	วิธีการส่งข้อมูล	ชื่อที่ใช้งานในซ็อกเก็ต	โพรโตคอล
AF_INET	SOCK_DGRAM	IPPTOTO_UDP	UDP
AF_INET	SOCK_STREAM	IPPTOTO_TCP	TCP
AF_INET	SOCK_RAW	IPPTOTO_ICMP	ICMP
AF_INET	SOCK_RAW	IPPTOTO_RAW	raw



รูปที่ 7-1 แสดงการเชื่อมต่อโดยใช้ซ็อกเก็ต

7.2 การรันไคลเอนต์หลายโปรเซส

ในกรณีที่มีเซิร์ฟเวอร์เพียงโปรเซสเดียวแต่ต้องจัดการกับหลายๆไคลเอนต์ในเวลาเดียวกัน เคอร์เนล รอจนมีเหตุการบางอย่างเกิดขึ้น แล้วจึงเรียกโปรเซสมาทำงาน เช่นสั่งให้รอข้อมูลจนกระทั่งมีข้อมูลใน file descriptor จึงจะอ่านเข้ามา เซิร์ฟเวอร์จะทำงานที่ละหลายๆไคลเอนต์จนกระทั่งไคลเอนต์ส่งข้อมูลเข้ามาโดยใช้ฟังก์ชั่น select()



รูปที่ 7-2 แสดงการรันไคลเอ็นต์หลายโปรเซสโดยใช้ฟังก์ชัน select()

7.3 ข้อมูลของเน็ตเวิร์ค

ไคลเอนต์โปรเซสและเซิร์ฟเวอร์โปรเซสใช้ออบเจ็ค 2 อย่างรวมกันในการอ้างอิงถึงบริการที่ต้องการคือหมายเลขพอร์ตและ IP Address กำหนดคอมพิวเตอร์ที่รันเซิร์ฟเวอร์ โปรเซสสนับสนุนการบริการนั้นเราสามารถใช้ฟังชั่น บางอย่างเพื่อหาค่าแอดเดรสหรือพอร์ตจากข้อมูลที่อยู่ในระบบ และจะต้องมีเพอร์มิสชั่นที่เพียงพอด้วย

ยกตัวอย่างใช้ชื่อโดเมนของคอมพิวเตอร์แทน IP Address ทำให้โปรแกรมค้นหา IP Address จากไฟล์ /etc/hosts หรือ DNS (Domain Name Service) หรือ NIS (Network Information Service)

หรือกำหนดชื่อของการให้บริการของเซิร์ฟเวอร์ไว้ที่ไฟล์ /etc/services ทำให้ไม่ต้องใช้หมายเลขพอร์ตแต่จะใช้ชื่อนั้นแทน

ฟังก์ชั่นแอ็กเซสโครสร้างข้อมูลเน็ตเวิร์คต่อไปนี้จะกำหนดไว้ในไฟล์ netdb.h

```
#include <netdb.h>
struct hostent *gethostbyaddr(const void *addr, size_t len,int type);
struct hostent *gethostbyname(const char *name);
```

ค่าที่ส่งกลับจากฟังชั่นข้างต้นจะมีฟิลด์ข้อมูลดังต่อไปนี้

```
struct hostent {
      char h_hame;            /* name of the host */
      char **h_aliases;       /* list of alisaes (nickname) */
      int addrtype;           /* address type */
      int h_length;           /* length in bytes of the address */
      char **a_addr_list;     /* list of address (network order) */
};
```

ถ้าไม่มีข้อมูลของ IP Address ที่กำหนด จะส่งค่ากลับเป็น null pointer สำหรับข้อมูลที่เกี่ยวข้องกับบริการต่างๆ ของเซิร์ฟเวอร์หรือพอร์ตจะหาได้โดยฟังชั่นดังต่อไปนี้

```
#include <netdb.h>
struct servent *getserverbyname(const char *name, const char *proto);
struct servent *getserverbyport(int port, const char *proto);
```

proto จะกำหนดโพรโตคอลที่ใช้งานสำหรับบริการนั้นๆ เช่น TCP สำหรับ SOCK_STREAM หรือ UDP สำหรับ SOCK_DGRAM

โครงสร้างข้อมูล servent จะประกอบด้วยฟิลด์ข้อมูลดังต่อไปนี้

```
struct servent{
      char s_name;            /* name of the services */
      char **s_aliases;       /* list of alisaes */
                              /* (alternative names) */
      int s_port;             /* The ip port number */
      char s_proto;           /* The service type */
};
```

ฟังชั่น gethostbyname ข้างต้นจะค้นหาข้อมูลของคอมพิวเตอร์ที่กำหนดโดยใช้ชื่อสำหรับรายชื่อแอดเดรสของเน็ตเวิร์ค จะต้องเปลี่ยนจากรูปแบบของเน็ตเวิร์คเป็นเกซ์ ก่อนที่จะนำไปประมวลผลในรูปของข้อความโดยการใช้ฟังชั่น inet_ntoa() ดังต่อไปนี้

```
#include <arpa/inet.h>
char *inet_ntoa(struct in_addr in);
```

โดยจะส่งค่ากลับเป็นเท็กซ์ของ IP Address หรือ -1 ถ้ามี Error เกิดขึ้น

นอกจากนี้ยังมีฟังชั่น gethostname ซึ่งจะเขียนชื่อของคอมพิวเตอร์ลงตัวแปร name และ namelength จะกำหนดความยาวของชื่อคอมพิวเตอร์ ถ้าเกินก็จะตัดออก จะส่งค่า 0 เมื่อทำสำเร็จหรือ -1 เมื่อเกิด Error ดังต่อไปนี้

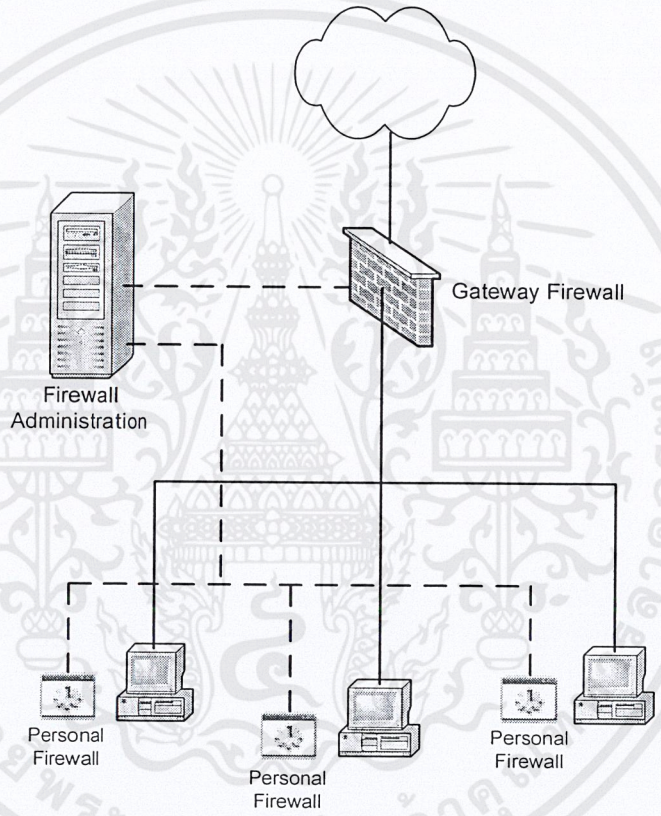
```
#include <unistd.h>
int gethostbyname (char *name, int namelength);
```

บทที่ 8

การออกแบบระบบและการทำงานของระบบ

8.1 โครงสร้างและส่วนประกอบหลักของระบบ

โครงสร้างและส่วนประกอบหลักของระบบไฟร์วอลแบบกระจายพร้อมระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์จะแสดงได้ดังรูปต่อไปนี้



รูปที่ 8-1 โครงสร้างและส่วนประกอบหลักของระบบ

โครงสร้างของระบบนั้นจะแบ่งออกเป็นส่วนประกอบหลักๆ ได้ 3 ส่วน คือ

- **Firewall Administration**

ในส่วนนี้ได้ออกแบบมาเพื่อใช้จัดการกฎต่างๆ ที่ใช้ในการควบคุมไฟร์วอลที่อยู่ในระบบให้ทำงานตามกฎได้ และหากได้รับการแจ้งเตือนการโจมตีหรือการบุกรุกทางเครือข่ายคอมพิวเตอร์ก็จะนำไปทำการป้องกันต่อไป

- **Gateway Firewall**

ส่วนของเกตเวย์ไฟร์วอลนี้จะออกแบบให้สามารถทำงานร่วมกับ IPTABLES ได้

- **Personal Firewall**

ส่วนของเพอร์ซันนอลไฟร์วอลจะออกแบบให้ทำงานร่วมกับระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ จะมีหน้าที่เป็นไฟร์วอล และ แจ้งเตือนการบุกรุกกลับไปยังเซิร์ฟเวอร์

8.2 ไฟร์วอลแอดมินิสเตชัน

8.2.1 หลักการ

ส่วนของไฟร์วอลแอดมินิสเตชัน(Firewall Administration) นี้นับว่าเป็นส่วนที่สำคัญที่สุดของระบบ เพราะว่า ส่วนนี้จะทำหน้าที่ในการดูและกฎ(Rule) ต่างๆ ของไฟร์วอลที่อยู่ในระบบทั้งหมด ไม่ว่าจะเป็นเกต์เวย์ไฟร์วอล หรือ เพอร์ซันนอลไฟร์วอล ส่วนของไฟร์วอลแอดมินิสเตชันนี้จะเป็นเซิร์ฟ เวอร์รอรับการเชื่อมต่อจาก ไฟร์วอลเกตเวย์ และ เพอร์ซันนอลไฟร์วอล ซึ่งจะมีหน้าที่หลักๆ อยู่สองหน้าที่คือ ควบคุม และ รับการแจ้งเตือน

8.2.2 ขอบเขตและความสามารถ

ไฟร์วอลแอดมินิสเตชันจะทำหน้าที่เปิดให้บริการเป็นเซิร์ฟเวอร์รอการเชื่อมต่อจากไคลเอนต์อยู่ ซึ่งในที่นี้ก็คือ เกตเวย์ไฟร์วอล และ เพอร์ซันนอลไฟร์วอล การทำงานของไฟร์วอลแอดมินิสเตชันจะมีอยู่สองส่วนคือ ส่วนที่ทำงานร่วมกับเกตเวย์ไฟร์วอล และส่วนที่ทำงานร่วมกับเพอร์ซันนอลไฟร์วอล

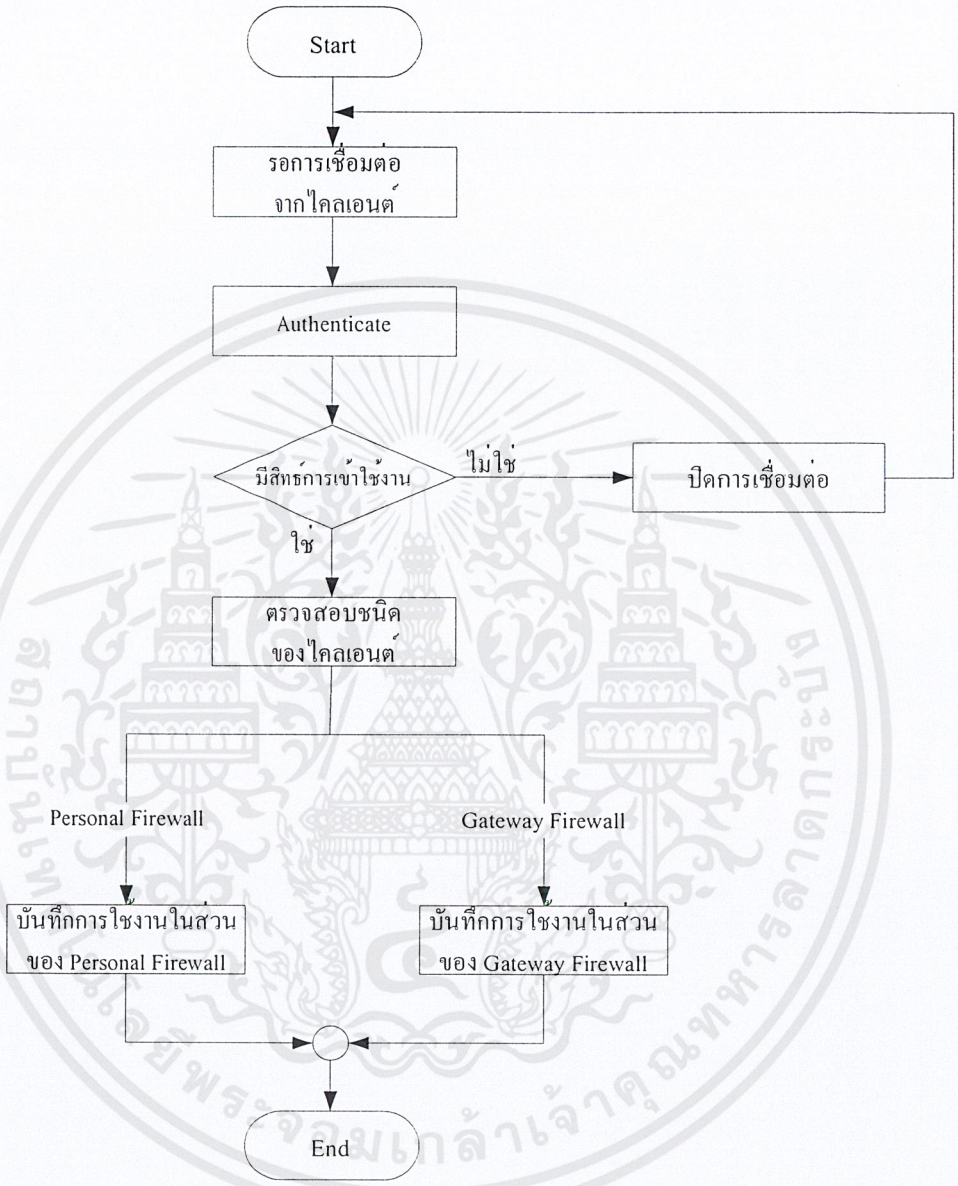
ส่วนแรกที่ทำงานร่วมกับเกตเวย์ไฟร์วอลจะมีความสามารถในการควบคุมเกตเวย์ไฟร์วอลได้หลายจุดซึ่งก็คือจะสามารถเพิ่มหรือลบเกตเวย์ไฟร์วอลที่ต้องการจะควบคุมได้ และสามารถที่จะอัพเดตกฎได้พร้อมๆ กันทั้งหมด

ส่วนที่สองคือส่วนที่ทำงานร่วมกับเพอร์ซันนอลไฟร์วอล ส่วนนี้ก็จะมีความสามารถในการควบคุมเพอร์ซันนอลไฟร์วอลได้หลายจุดเช่นกัน ซึ่งส่วนนี้จะสามารถแบ่งการควบคุมได้หลายระดับ ซึ่งจะแบ่งเป็นกลุ่ม(Group) ซึ่งสามารถกำหนดระดับความปลอดภัยที่แตกต่างกันในแต่ละกลุ่มได้

ความสามารถอีกอย่างหนึ่งของไฟร์วอลแอดมินิสเตชัน คือสามารถดูการเชื่อมต่อของเกตเวย์ไฟร์วอล หรือ เพอร์ซันนอลไฟร์วอล ที่เชื่อมต่อเข้ามาได้ สามารถดูสถานะของแต่ละไคลเอนต์ที่อยู่ในระบบนี้ได้ทั้ง เกตเวย์ไฟร์วอล และ เพอร์ซันนอลไฟร์วอล

8.2.3 หน้าที่การทำงานของไฟร์วอลแอดมินิสเตชัน

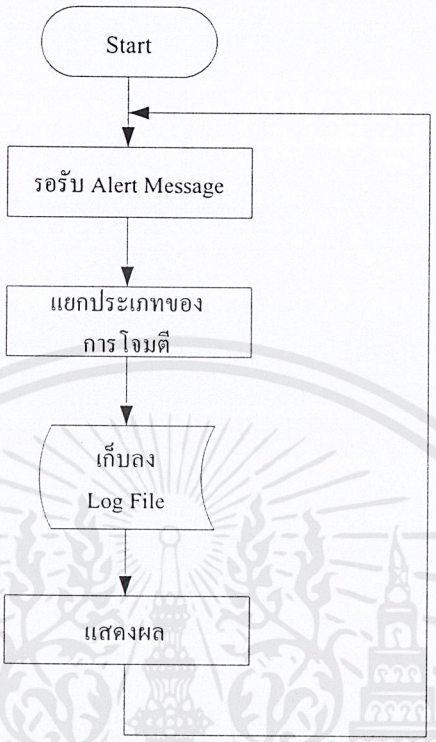
- การทำงานในส่วนของการจัดการไคลเอนต์



รูปที่ 8-2 การทำงานในส่วนของการจัดการไคลเอนต์

การทำงานในส่วนของการจัดการไคลเอนต์ของไฟร์วอลแอดมินิสเตชันจะมีลำดับการทำงานดังรูปที่แสดงนี้

- การทำงานในส่วนของการรอรับการแจ้งเตือน



รูปที่ 8-3 การทำงานในส่วนของการรอรับการแจ้งเตือน

การทำงานในส่วนของการรอรับการแจ้งเตือนของไฟร์วอลแอดมินิสเตชันจะมีลำดับการทำงานดังรูปที่แสดงนี้ ซึ่งในส่วนนี้จะรอรับการแจ้งเตือนตลอดเวลา และเมื่อมีการแจ้งเตือนเข้ามาก็จะทำงานในลำดับต่อๆ ไป

8.3 เกตเวย์ไฟร์วอล

8.3.1 หลักการ

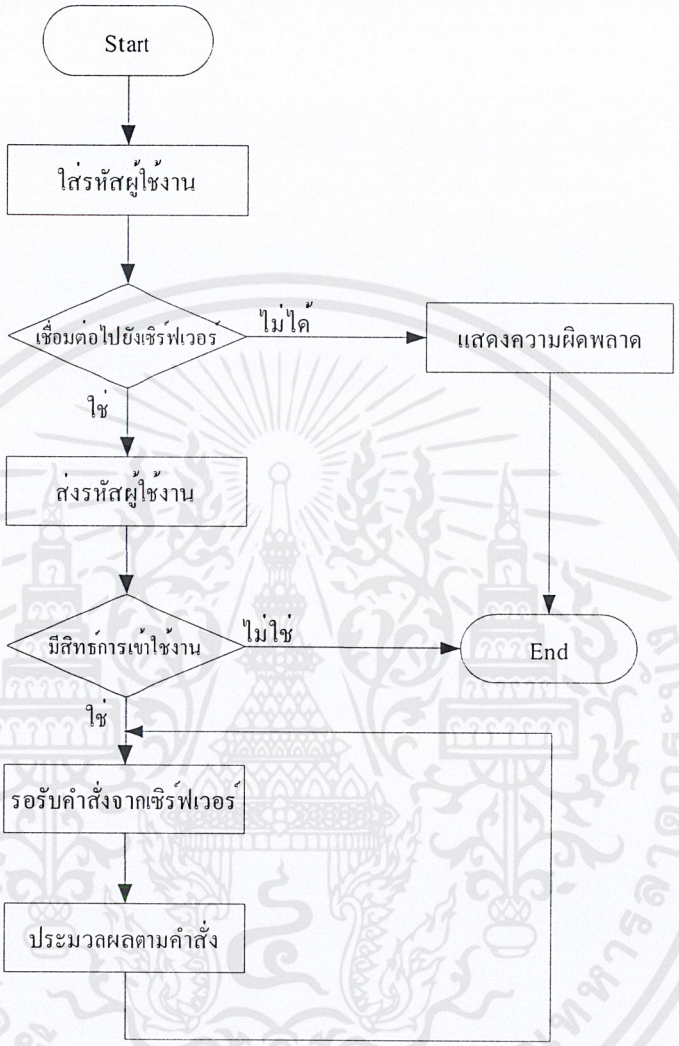
ส่วนของเกตเวย์ไฟร์วอล(Gateway Firewall) นี้จะมีหน้าที่ทำงานเป็นไฟร์วอลอยู่ตรงส่วนที่เป็นเกตเวย์ และ ทำงานร่วมกับ IPTABLES ในการเพิ่มหรือลบกฎที่ซึ่งจะถูกควบคุมจากไฟร์วอลแอดมินิสเตชันอีกทีหนึ่ง

8.3.2 ขอบเขตและความสามารถ

เกตเวย์ไฟร์วอลจะทำงานอยู่ตรงส่วนเกตเวย์ ซึ่งจะในขณะที่เชื่อมต่อกับ ไฟร์วอลแอดมินิสเตชันอยู่นั้นก็จะรอรับการควบคุมจากไฟร์วอลแอดมินิสเตชันซึ่งก็คือ เมื่อเกตเวย์ได้รับคอนโทรลเมสเซจ (Control Message) ที่ถูกส่งมาก็จะทำ การเพิ่มกฎ หรือ ลบกฎ ใน IPTABLES ตามแต่ที่ได้รับคอนโทรลเมสเซจมาจากไฟร์วอลแอดมินิสเตชัน

8.3.3 หน้าที่การทำงานของเกตเวย์ไฟร์วอล

- การทำงานในส่วนของเกตเวย์ไฟร์วอล



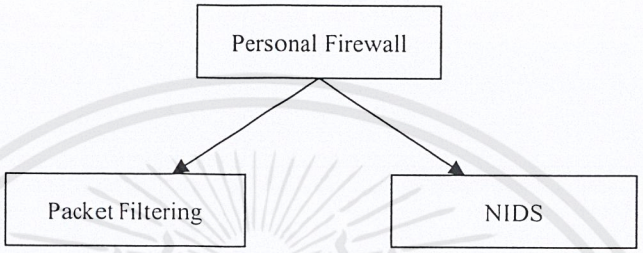
รูปที่ 8-4 การทำงานในส่วนของเกตเวย์ไฟร์วอล

เกตเวย์ไฟร์วอลจะมีหน้าที่เพียงอย่างเดียวนั่นก็คือติดต่อกับเซิร์ฟเวอร์เพื่อรอรับคำสั่งที่ จะนำมาใช้ในการควบคุม IPTABLES อีกทีหนึ่ง

8.4 เพอร์ซันนอลไฟร์วอล

8.4.1 หลักการ

เพอร์ซันนอลไฟร์วอลจะติดตั้งอยู่ที่แต่ละไคลเอนต์ในเครือข่าย มีหน้าที่อยู่สองอย่างหลักๆ ก็คือ ทำหน้าที่เป็น Packet Filtering และทำหน้าที่เป็นระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์(Network Intrusion Detection System) และมีระบบการแจ้งเตือนกลับไปยังไฟร์วอลแอดมินิสเตชันหากตรวจจับได้ว่ามีการบุกรุกเกิดขึ้น



รูปที่ 8-5 หน้าที่การทำงานของเพอร์ซันนอลไฟร์วอล

8.4.2 ขอบเขตและความสามารถ

เครื่องทุกๆ เครื่องในเครือข่ายจะต้องมีเอเจนต์ฝังตัวทำงานอยู่ ซึ่งเอเจนต์ทั้งหลายนั้นจะทำหน้าที่คล้ายกับเพอร์ซันนอลไฟร์วอล คือจะทำหน้าที่ตรวจสอบแพ็กเก็ตที่เข้า - ออกเครื่องนั้นๆ แต่จะต่างกับเพอร์ซันนอลไฟร์วอลธรรมดาตรงที่สามารถควบคุมดูแลและบริหารเอเจนต์ทั้งหมดจากส่วนกลาง โดยผู้ใช้จะไม่รับรู้ถึงการทำงานของเอเจนต์ อีกทั้งผู้ดูแลระบบ สามารถกำหนดกฎให้กับผู้ใช้แต่ละคนตามความเหมาะสมว่าบุคคลนั้น มีสิทธิ์ในการใช้บริการมากน้อยเพียงใด จากสถาปัตยกรรมการทำงานแบบนี้ทำให้สามารถที่จะรวมเอาข้อดีของไฟร์วอลทั้งสองประเภทเอาไว้เข้าด้วยกันได้

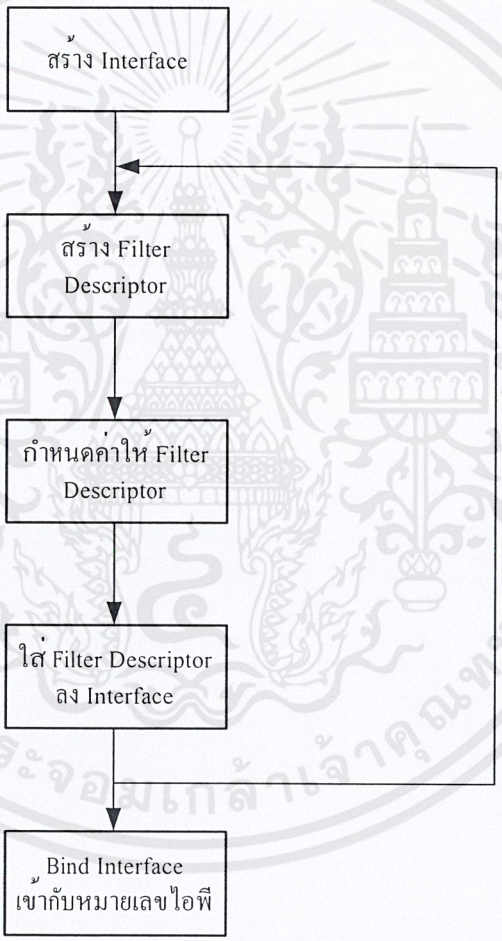
อีกทั้งยังมีส่วนของการทำงานที่เป็นระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ที่จะคอยทำหน้าที่ตรวจสอบว่ามีการบุกรุกเข้ามาหรือไม่ ถ้าหากมีการบุกรุกเกิดขึ้น ก็จะทำการแจ้งเตือนกลับไปยังไฟร์วอลแอดมินิสเตชันเพื่อดำเนินการป้องกันต่อไป

8.4.3 หน้าที่การทำงานของเพอร์ซันนอลไฟร์วอล

- **การตรวจสอบและกรองแพ็กเก็ต**

การตรวจสอบและกรองแพ็กเก็ตสามารถทำได้ผ่าน Windows 2000 Packet Filtering API ในวินโดวส์ 2000 นั้นได้มี API (Application Programming Interface) ที่ใช้สำหรับทำแพ็กเก็ตฟิลเตอร์โดยเฉพาะมาให้ โดยไลบรารีของ API นี้จะอยู่ใน Microsoft Platform SDK ในไฟล์ IphIpApi.lib โดยจะต้อง include เฮดเดอร์ไฟล์ ชื่อ Fltdefs.h เข้าไปด้วย

สำหรับขั้นตอนการทำงานของ Packet Filtering API เป็นดังนี้



รูปที่ 8-6 ขั้นตอนการทำงานของ Packet Filter API

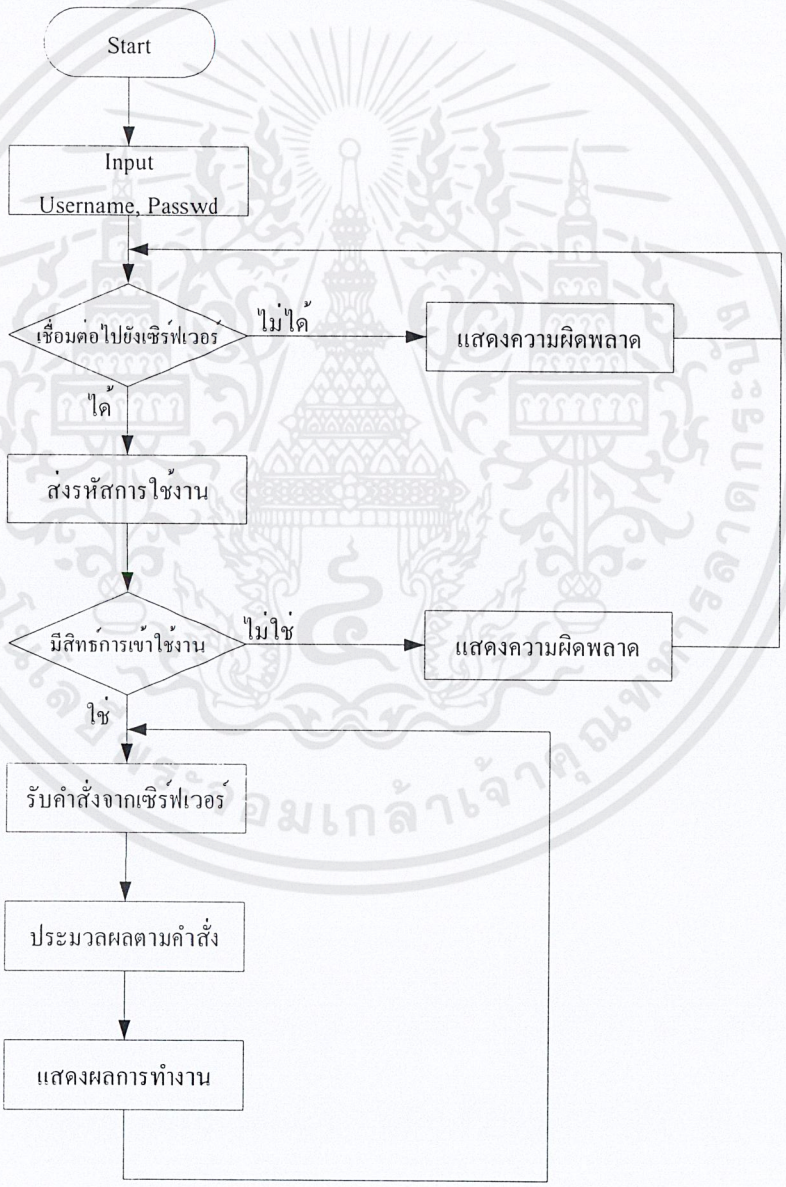
- **รูปแบบกฎของไฟร์วอล**

สำหรับโครงสร้างกฎของไฟร์วอลนั้นได้ยึดตามรูปแบบของ Windows 2000 Packet Filter API ซึ่งเป็นดังต่อไปนี้

Src Address	Src Mask	Src Port	Dst Address	Dst Mask	Dst Port	Protocol

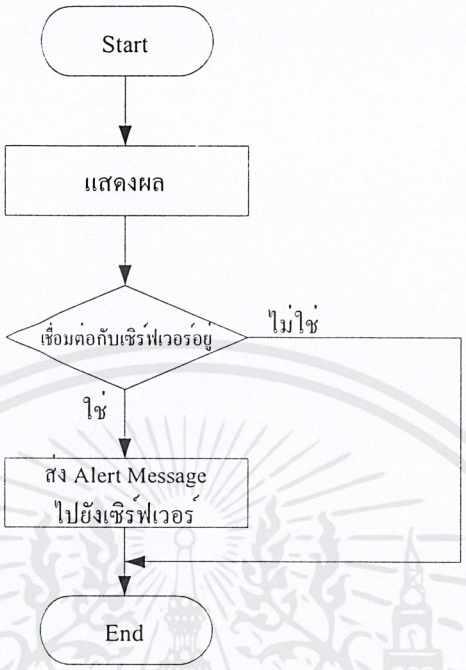
รูปที่ 8-7 รูปแบบกฎของไฟร์วอล

- **การทำงานของเพอร์ซันนอลไฟร์วอล**



รูปที่ 8-8 การทำงานของเพอร์ซันนอลไฟร์วอล

- **การแจ้งเตือนการบุกรุกของระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์**

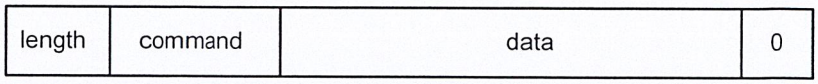


รูปที่ 8-9 การแจ้งเตือนการบุกรุกของระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

ในส่วนของการแจ้งเตือนการบุกรุก หลักการทำงานก็คือ เมื่อ ระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์สามารถตรวจจับได้ว่ามีการบุกรุกเกินขึ้น และในขณะนั้นได้ทำการเชื่อมต่ออยู่กับเซิร์ฟเวอร์ก็จะแจ้งเตือนกลับไปยังเซิร์ฟเวอร์ทันที

8.5 การติดต่อระหว่างไคลเอนต์และเซิร์ฟเวอร์

ในการส่งข้อมูลควบคุมระหว่างไคลเอนต์และเซิร์ฟเวอร์จะมีโครงสร้างของโพรโตคอลดังต่อไปนี้



รูปที่ 8-10 โพรโตคอลในการติดต่อระหว่างไคลเอนต์และเซิร์ฟเวอร์

length คือความยาวสูงสุดของของมูลทั้งหมดมีขนาด 1 ไบต์

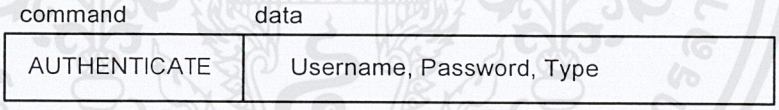
command คือรหัสคำสั่งที่ใช้ควบคุมมีขนาด 1 ไบต์

data คือข้อมูลที่จะมีหรือไม่มีก็ได้ขึ้นอยู่กับ Command ว่าต้องการมีหรือไม่มีขนาดสูงสุด 252 ไบต์ ถ้าไม่มีข้อมูลจะถูกตั้งค่าไว้เป็น 0 มีขนาด 1 ไบต์

0 คือส่วนท้ายของข้อมูลมีขนาด 1 ไบต์

8.5.1 คำสั่ง AUTHENTICATE

คำสั่งที่ใช้เมื่อไคลเอนต์เป็นผู้เริ่มขอการเชื่อมต่อไปยังไฟร์วอลแอดมินิสเตชัน ทิศทางของคำสั่งไคลเอนต์ไปยังเซิร์ฟเวอร์



Username คือ ID ของยูสเซอร์ที่มีอยู่ในระบบ

Password คือรหัสผ่านของยูสเซอร์

Type ชนิดของไคลเอนต์ที่ขอเชื่อมต่อ “1” เมื่อเป็นเกตเวย์ไฟร์วอล เป็น “2” เมื่อเป็นเพอร์ซันนอลไฟร์วอล

รูปที่ 8-11 รูปแบบของคำสั่ง AUTHENTICATE

8.5.2 คำสั่ง AUTHENTICATE_OK

command	data
AUTHENTICATE_OK	NULL

รูปที่ 8-12 รูปแบบของคำสั่ง AUTHENTICATE_OK

เป็นคำสั่งเพื่อบอกว่ายูสเซอร์มีสิทธ์ที่จะเชื่อมต่อกับไฟร์วอลแอดมินิสเตชัน ทิศทางของคำสั่งเซิร์ฟเวอร์ไปยังไคลเอนต์

8.5.3 คำสั่ง AUTHENTICATE_REJECT

command	data
AUTHENTICATE_REJECT	NULL

รูปที่ 8-13 รูปแบบของคำสั่ง AUTHENTICATE_REJECT

คำสั่งในกรณีที่ไคลเอนต์ไม่มีสิทธ์เข้าใช้งานตัวเซิร์ฟเวอร์ ทิศทางของคำสั่ง เซิร์ฟเวอร์ไปยังไคลเอนต์

8.5.4 คำสั่ง ALERT

command	data
ALERT	Attack_type, Attacker_Address

รูปที่ 8-14 รูปแบบของคำสั่ง ALERT

Attack_type ชนิดของการโจมตีมีรายละเอียดดังต่อไปนี้

<u>การโจมตีในระดับ IP</u>

- IP_BOMB
- IP_IPLOOP
- IP_OVERLAPPED
- IP_GAP

การโจมตีในโพรโตคอล ICMP

- ICMP_BOMB
- ICMP_IPLOOP
- ICMP_PINGSWEEP
- ICMP_OVERLAPPED

การโจมตีในโพรโตคอล TCP

- TCP_SCANPORT
- TCP_SYNCFLOOD
- TCP_OSFINGERPRINT

การโจมตีในโพรโตคอล UDP

- UDP_SCANPORT

Attacker_address IP Address ของผู้โจมตี

8.5.5 คำสั่ง GATEWAY_UPDATE_RULE

command	data
GATEWAY_UPDATE_RULE	IPTABLES_Command

รูปที่ 8-15 รูปแบบของคำสั่ง GATEWAY_UPDATE_RULE

IPTABLES_Command คำสั่งที่ใช้กับ IPTABLES บนระบบปฏิบัติการลีนุกซ์ เวอร์ชัน 2.4 เป็นคำสั่งเปลี่ยนแปลงกฎให้แก่เกตเวย์ไฟร์วอล ทิศทางของคำสั่ง เซิร์ฟเวอร์ไปยังไคลเอนต์

8.5.6 คำสั่ง PERSONAL_UPDATE_RULE

command	data
PERSONAL_UPDATE_RULE	Win32_Firewall_API_Command

รูปที่ 8-16 รูปแบบของคำสั่ง PERSONAL_UPDATE_RULE

Win32_Firewall_API_Command เป็นโครงสร้างข้อมูลของ Win32 Firewall API เป็นคำสั่งเปลี่ยนแปลงกฎให้แก่เพอร์ซันนอลไฟร์วอล ทิศทางของคำสั่ง เซิร์ฟเวอร์ไปยังไคลเอนต์

บทที่ 9

การทดสอบการทำงาน

9.1 การทดสอบประสิทธิภาพของระบบ

ขั้นตอนในการทดสอบการทำงานของระบบไฟร์วอลแบบกระจายพร้อมระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์นี้ มีรายละเอียดของเครื่องที่ทำการทดสอบดังต่อไปนี้

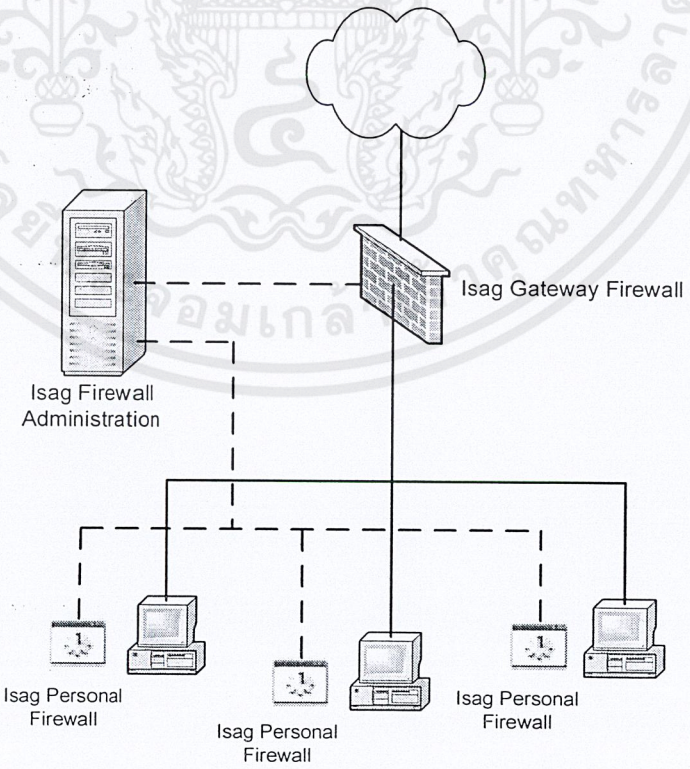
เครื่องที่ติดตั้งโปรแกรม Isag Personal Firewall

- หน่วยประมวลผลความเร็ว 950 Mhz
- หน่วยความจำหลัก 256 MB
- ระบบปฏิบัติการไมโครซอฟท์วินโดว์ 2000 ที่ติดตั้งเซอร์วิสแพ็ก 3
- การ์ดแลนความเร็ว 100 Mb

เครื่องที่ติดตั้งโปรแกรม Isag Firewall Administration และ Isag Gateway Firewall

- หน่วยประมวลผลความเร็ว 1000 Mhz
- หน่วยความจำหลัก 512 MB
- ระบบปฏิบัติการลีนุกซ์เคเบียน เคอร์เนลเวอร์ชัน 2.4.20
- การ์ดแลนความเร็ว 100 Mb

9.2 โครงสร้างของระบบที่ใช้ในการทดสอบ



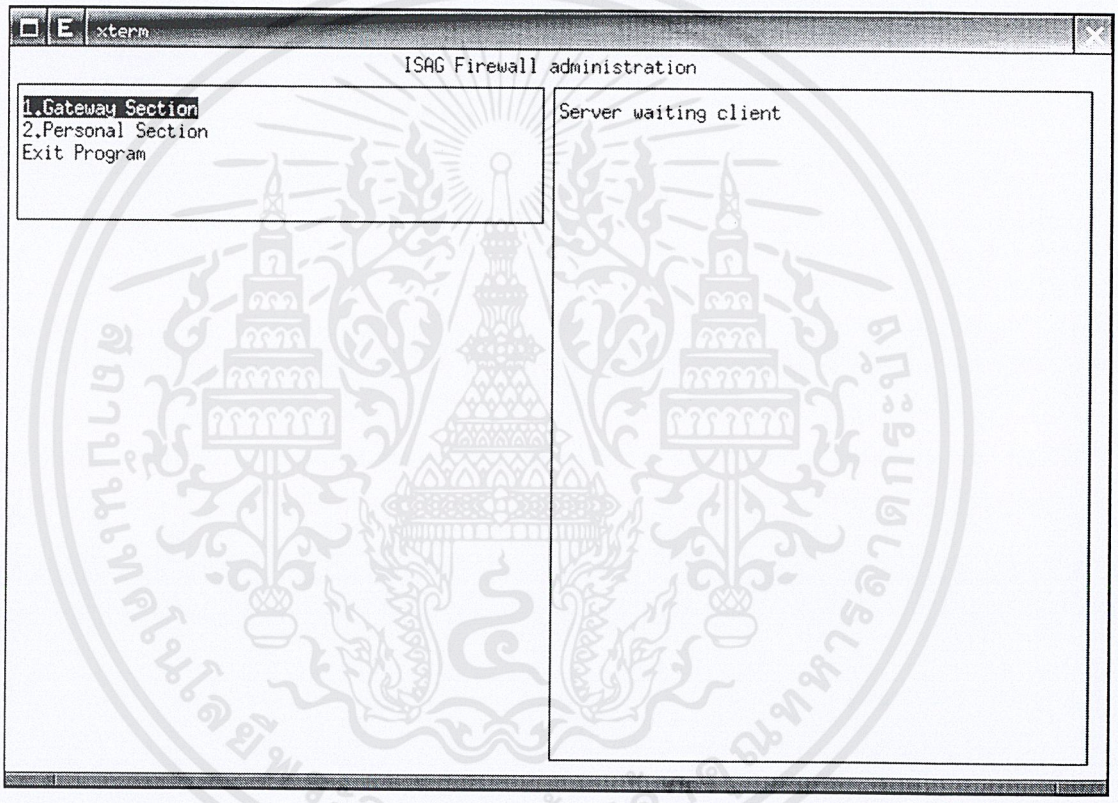
รูปที่ 9-1 โครงสร้างทางเครือข่ายของระบบที่ใช้ในการทดสอบ

โครงสร้างของระบบที่สามารถทำงานได้เต็มประสิทธิภาพนั้น ควรจะต้องติดตั้งให้ครบทุกส่วน ทั้งส่วนไฟร์วอลแอดมินิสเตชัน, ส่วนเกตเวย์ไฟร์วอล และส่วนเพอร์ซันนอลไฟร์วอลพร้อมระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ ในการทดสอบระบบนี้ ได้ติดตั้งโปรแกรมครบตามโครงสร้างทุกส่วนดังที่ได้กล่าวมาแล้ว

9.3 เริ่มต้นการทดสอบระบบ

- การเริ่มต้นการทำงานของไฟร์วอลแอดมินิสเตชัน

  เมื่อโปรแกรมไฟร์วอลแอดมินิสเตชันเริ่มต้นการทำงานแล้ว จะมีลักษณะดังต่อไปนี้



รูปที่ 9-2 โปรแกรมไฟร์วอลแอดมินิสเตชัน

โปรแกรมจะแบ่งออกเป็นสองส่วนหลักๆ ก็คือ ส่วนแรก เป็นเมนูในการควบคุมส่วนต่างๆ ของระบบ ซึ่งจะอยู่ทางด้านซ้ายของหน้าต่างโปรแกรม อีกส่วนจะเป็นส่วนที่เป็นมอนิเตอร์ ซึ่งจะเป็นตัวแสดงผลการทำงานต่างๆ เช่น เมื่อมีไคลเอนต์ทำการเชื่อมต่อเข้ามา หรือมีการแจ้งเตือนจากไคลเอนต์เข้ามา ส่วนแสดงผลส่วนนี้ก็จะแสดงให้รู้ว่ามีเชื่อมต่อเกิดขึ้น

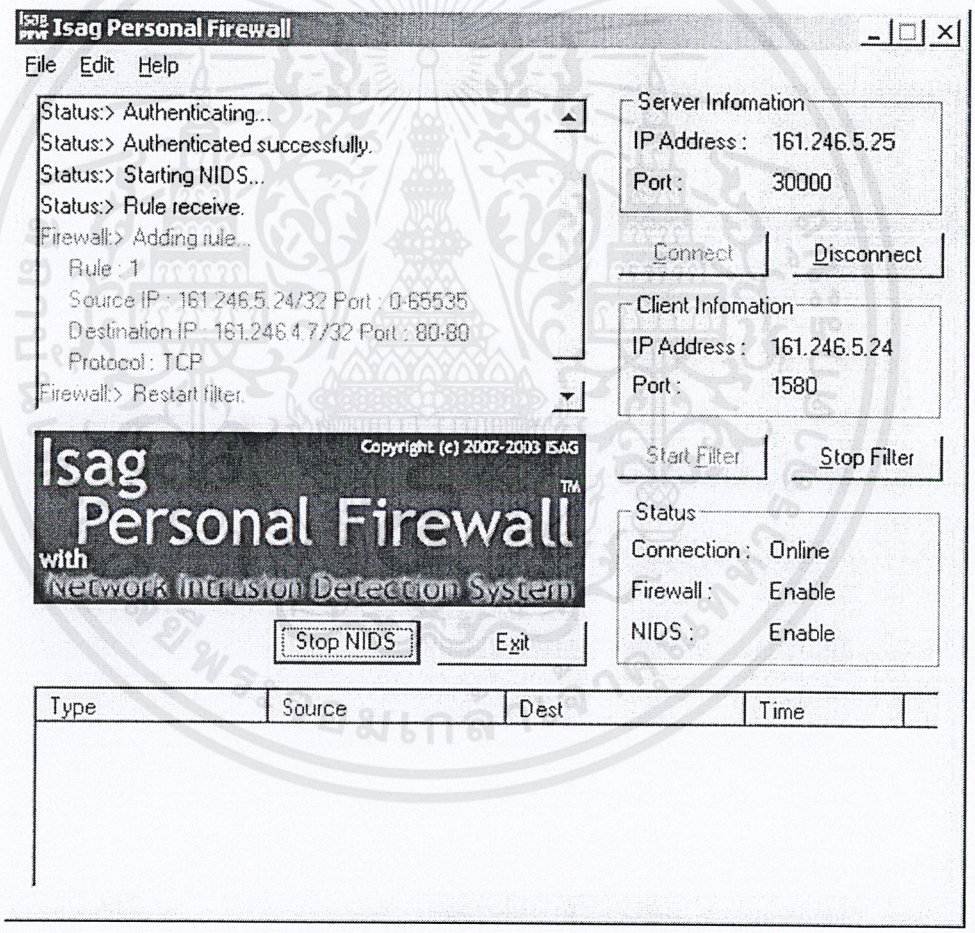
- การเริ่มต้นการทำงานของเกตเวย์ไฟร์วอล

สิ่งแรกเครื่องคอมพิวเตอร์ที่ติดตั้งเกตเวย์ไฟร์วอลนี้ จะต้องติดตั้ง IPTABLES อยู่ในเครื่อง ไม่เช่นนั้นจะไม่สามารถทำงานได้ เพราะเกตเวย์ไฟร์วอล จะทำงานควบคู่กับ IPTABLES

เมื่อเกตเวย์ไฟร์วอล เริ่มทำงานในตอนแรก สิ่งที่เกี่ยวข้องกับผู้ใช้งานก็คือ ต้องกรอกชื่อและรหัสผ่าน เพื่อทำการเชื่อมต่อไปยังเซิร์ฟเวอร์ เพื่อเข้าใช้งานหลังจากนั้น เมื่อทำการเชื่อมต่อกับเซิร์ฟเวอร์ได้แล้ว ก็จะไม่มีส่วนที่เกี่ยวข้องกับผู้ใช้เลย ตัวโปรแกรมก็จะทำงานต่อไป แต่หากการเชื่อมต่อกับเซิร์ฟเวอร์เกิดความผิดพลาดขึ้นมา เกตเวย์ไฟร์วอลก็ยังสามารถทำงานต่อไปได้ด้วย IPTABLES แต่จะไม่สามารถที่จะทำการแก้ไข หรือเปลี่ยนแปลงกฎของไฟร์วอลได้

- การเริ่มต้นการทำงานของเพอร์ซันนอลไฟร์วอล

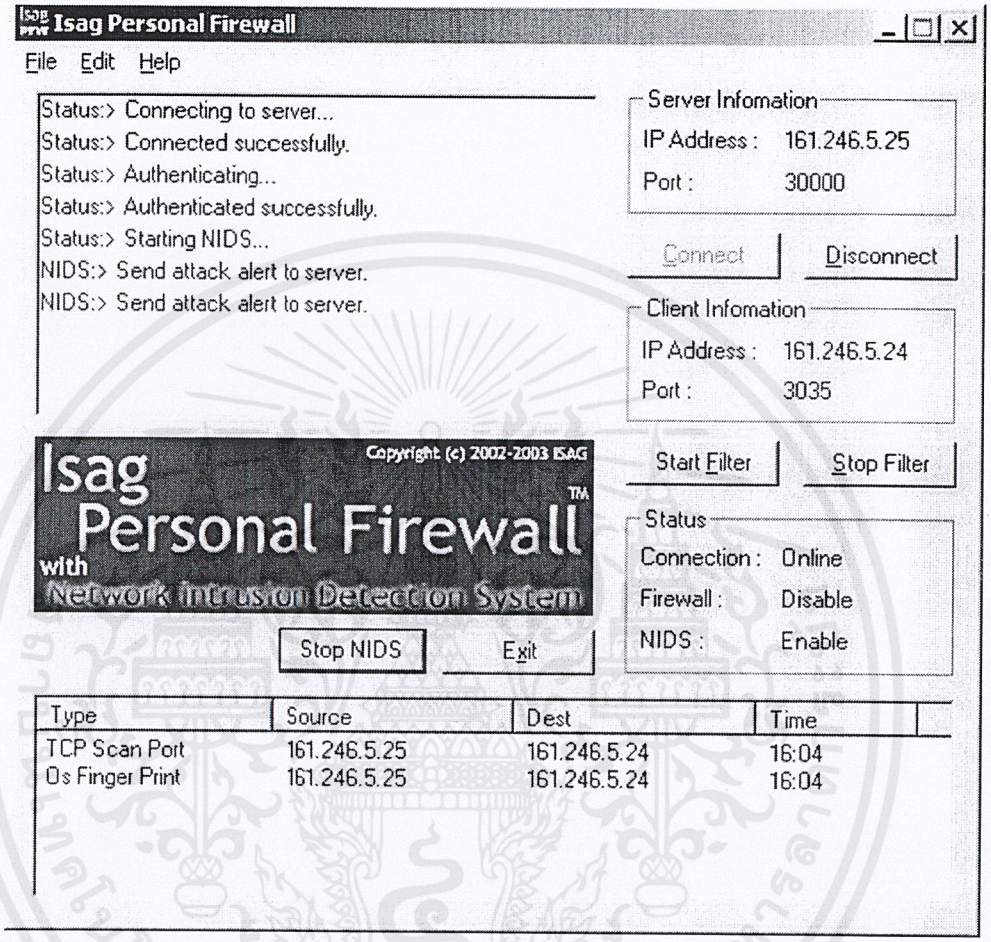
เมื่อโปรแกรมเพอร์ซันนอลไฟร์วอลเริ่มการทำงานแล้ว จะมีลักษณะดังต่อไปนี้



รูปที่ 9-3 โปรแกรมเพอร์ซันนอลไฟร์วอล

เมื่อโปรแกรมเริ่มทำงานแล้ว ต้องทำการเชื่อมต่อไปยังเซิร์ฟเวอร์ และ เริ่มต้นการทำงานของระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ เมื่อทำการเชื่อมต่อไปยังเซิร์ฟเวอร์สำเร็จ ได้จะได้รับคำสั่งในการอัพเดทกฎ ที่ทางเซิร์ฟเวอร์ส่งมา

และเมื่อมีการโจมตีหรือการบุกรุกทางเครือข่ายคอมพิวเตอร์เกิดขึ้น โปรแกรมก็จะแสดงผลรายละเอียดต่างๆ ของการโจมตีหรือการบุกรุกนั้นๆ ให้ทราบและ ส่งข้อมูลการโจมตีหรือการบุกรุกไปยังเซิร์ฟเวอร์ดังรูป



รูปที่ 9-4 การแจ้งเตือนการบุกรุก

9.4 ปัญหาที่เกิดขึ้นขณะทดสอบ

ในการทดสอบนั้น ได้พบกับปัญหาที่ไม่ได้เกิดจากการตัวโปรแกรมโดยตรงอย่างหนึ่งก็คือ ในขั้นตอนของการเชื่อมต่อเข้าหาเซิร์ฟเวอร์นั้น ในบางครั้งยังไม่สามารถเชื่อมต่อเข้าได้ทันที ต้องรอซักพัก ก็จะสามารถเชื่อมต่อเข้ากับเซิร์ฟเวอร์ได้เหมือนเดิม สาเหตุนี้เกิดขึ้นเพราะ ในขณะที่ทำการทดสอบอยู่นั้น ตัวโปรแกรมทางฝั่งเซิร์ฟเวอร์ได้ปิดให้บริการลงไป เมื่อเปิดโปรแกรมทางฝั่งเซิร์ฟเวอร์และ เชื่อมต่อเข้าไปอีกครั้งก็จะยังไม่สามารถเชื่อมต่อได้ เพราะว่าเซิร์ฟเวอร์กำลังปิดการเชื่อมต่อเดิมอยู่ และอยู่ในสถานะ TIME_WAIT ซึ่งเมื่อรอซักพักก็จะเชื่อมต่อได้เหมือนเดิม

บทที่ 10
สรุปผลและวิจารณ์

10.1 ปัญหาและอุปสรรคในการพัฒนาโปรแกรม

ในช่วงเวลาที่ได้พัฒนาโปรแกรมไฟร์วอลแบบกระจายพร้อมระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์นั้นได้พบปัญหาต่างที่เป็นอุปสรรคต่อการพัฒนาหลายประการ ได้แก่

- การศึกษาโครงงานเดิมที่มีอยู่แล้ว ซึ่งจำเป็นที่จะต้องศึกษาอย่างละเอียดและให้เข้าใจเป็นอย่างดี เพื่อที่จะนำส่วนการทำงานหลักๆ นั้นมาใช้ในการพัฒนาเป็นโครงงานชิ้นนี้ ซึ่งปัญหาก็คือ การเขียนโปรแกรมของโครงงานเดิมนั้น มีคอมเมนต์น้อยมาก และชื่อฟังก์ชั่นกับตัวแปรบางตัวนั้นไม่สื่อความหมายเท่าที่ควร เป็นเหตุผลให้การศึกษาเพื่อให้เข้าใจการทำงานของโครงงานเดิมนั้นเป็นไปได้อย่างช้ามาก
- ในการพัฒนาส่วนของระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์นั้นได้พบกับปัญหา และอุปสรรคต่างๆ ดังนี้
  - โปรแกรมที่ใช้ในการบุกรุกมีอยู่มากมาย การศึกษาการทำงานของโปรแกรมเหล่านั้นต้องใช้เวลานาน ทำให้ยังไม่สามารถตรวจจับการบุกรุกได้ครอบคลุมทุกรูปแบบการบุกรุกที่มีอยู่ในปัจจุบัน และที่กำลังจะมีเพิ่มขึ้นในอนาคต
  - ระบบยังต้องทำงานผ่านไลบรารี WinPcap ซึ่งในช่วงทดลองโครงงานเดิมนั้น ได้ลองกับไลบรารี WinPcap เวอร์ชันใหม่ ที่ไม่ใช่เวอร์ชันเดิมที่เคยใช้ ผลออกมาคือไม่สามารถใช้งานได้ เพราะในเวอร์ชันใหม่ของ WinPcap ได้แก้ไขฟังก์ชันภายในซึ่งทำให้ไม่เหมือนกับเวอร์ชันเดิม จึงทำให้ไม่สามารถเลือกเน็ตเวิร์คการ์ดที่ติดตั้งอยู่ในเครื่องได้ รวมถึงไม่สามารถที่จะเรียกใช้ฟังก์ชันการทำงานเดิมบางฟังก์ชันได้
- ปัญหาในการนำเอาไฟร์วอลกับระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์มารวมกัน และทำให้สามารถทำงานด้วยกันอย่างสอดคล้องกันนั้น ต้องรู้หลักการทำงานของทั้งสองระบบเป็นอย่างดี ด้วยเหตุนี้ทำให้ต้องใช้เวลานานพอสมควรในการศึกษาและทำความเข้าใจอย่างละเอียด

10.2 แนวทางการนำไปพัฒนาต่อในอนาคต

เนื่องจากโครงงาน ไฟร์วอลแบบกระจายพร้อมระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์นี้ ได้นำส่วนหนึ่งของโครงงานเดิมที่มีการพัฒนาไว้แล้ว นำมาพัฒนาต่อให้เป็นระบบใหม่ที่สามารถนำไปใช้งานได้จริง แต่หากจะนำโครงงานนี้ไปพัฒนาต่อแล้ว ต้องอาศัยความเข้าใจทั้งในเรื่องไฟร์วอล และในเรื่องระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์เป็นอย่างดี อีกทั้งต้องมีความรู้ในเรื่องการเขียนโปรแกรมบนระบบปฏิบัติการวินโดวส์ และระบบปฏิบัติการลีนุกซ์ด้วย เพราะโครงงานชิ้นนี้จะทำงานอยู่บนทั้งสองระบบปฏิบัติการ

แนวทางการพัฒนาต่อจึงควรนำไปทำให้อยู่ในชุดของไฟร์วอลสูท(Firewall Suite) คือควรนำ ไปพัฒนาร่วมกับโครงงานไฟร์วอลรูปแบบอื่นที่มีหน้าที่ลักษณะการทำงานที่ไม่เหมือนกับโครงงานนี้ เพื่อที่จะเพิ่มประสิทธิภาพของระบบไฟร์วอลให้แข็งแกร่งและมีประสิทธิภาพดียิ่งขึ้นต่อไป

บรรณานุกรม

หนังสืออ้างอิง

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

[2] W. Richard Stevens, *"UNIX Network Programming"*, Prentice-Hall, 1999

[3] Behrouz A. Forouzan, *"TCP/IP Protocol Suite"*, McGraw-Hill, 2003

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

[5] สันติ ศรีลาศักดิ์, วรวุฒิ เที่ยงธรรม, *"เจาะประเด็นงานเขียนโปรแกรมบนลีนุกซ์"*, ซีเอ็ดยูเคชั่น, 2542

เว็บไซต์อ้างอิง

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

[7] http://www.thaicert.nectec.or.th

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