ชุดโปรแกรมเน็ตเวิร์กไฟร์วอล

NETWORK FIREWALL PROGRAM SUITE

นายจุมพล ทุมมาวัด
นายประภาส ผ่องสนาม
นายสุรศักดิ์ ทับเคลียว

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2547

ชุดโปรแกรมเน็ตเวิร์กไฟร์วอล

NETWORK FIREWALL PROGRAM SUITE

โดย

นายจุมพล	ทุมมาวัด	45015359
นายประภาส	ผ่องสนาม	45015373
นายสุรศักดิ์	ทับเคลียว	45015387

อาจารย์ที่ปรึกษา

อาจารย์อัครเดช วัชระภูพงษ์

อาจารย์ธนา หงษ์สุวรรณ

อาจารย์ธนัญชัย ตรีภาค

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

ปริญญานิพนธ์ปีการศึกษา 2547

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ชุดโปรแกรมเน็ตเวิร์กไฟร์วอล

NETWORK FIREWALL PROGRAM SUITE

ผู้จัดทำ

1. นายจุมพล ทุมมาวัด รหัสประจำตัว 45015359
2. นายประภาส ผ่องสนาม รหัสประจำตัว 45015373
3. นายสุรศักดิ์ ทับเคลียว รหัสประจำตัว 45015387

_____ อาจารย์ที่ปรึกษา
(อาจารย์อัครเดช วัชระภูพงษ์)

_____ อาจารย์ที่ปรึกษา
(อาจารย์ธนา หงษ์สุวรรณ)

_____ อาจารย์ที่ปรึกษา
(อาจารย์ธนัญชัย ตรีภาค)

ชุดโปรแกรมเน็ตเวิร์กไฟร์วอล

นายจุมพล ทุมมาวัด 45015359
นายประภาส ผ่องสนาม 45015373
นายสุรศักดิ์ ทับเคลียว 45015387
อาจารย์อัครเดช วัชระภูพงษ์ อาจารย์ที่ปรึกษา
อาจารย์ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษา
อาจารย์ธนัญชัย ตรีภาค อาจารย์ที่ปรึกษา
ปีการศึกษา 2547

บทคัดย่อ

ในปัจจุบันความปลอดภัยของเครือข่ายเป็นเรื่องที่ผู้คนให้ความสนใจกันมากขึ้น เนื่องจากการเจริญเติบโตของอินเทอร์เน็ต ทำให้มีความเสี่ยงต่อเรื่องของความปลอดภัยจึงสูงขึ้น ไฟร์วอลเป็นสิ่งหนึ่งซึ่งช่วยในการรักษาความปลอดภัย แต่ไฟร์วอลที่มีใช้อยู่ในปัจจุบันยังมีข้อจำกัดในการนำมาใช้ในงานในระบบเครือข่ายซึ่งจะทำให้เกิดความยุ่งยากและ ซับซ้อนในการจัดการดูแล แก้ไขหากมีเครื่องในเครือข่ายจำนวนมาก

โครงงานนี้ เป็นการพัฒนาไฟร์วอลโดยใช้ความสามารถของระบบแอ็คทีฟไดเรกทอรี บนระบบปฏิบัติการวินโดว์ 2000 ซึ่งเป็นเทคโนโลยีที่ใช้ในการจัดการ ควบคุมดูแลระบบเครือข่าย โดยจะนำมาใช้เพื่อควบคุม ดูแล และจัดการกฎการทำงานต่าง ๆ ของไฟร์วอล ได้จากศูนย์กลาง โดยแบ่งระบบการทำงานออกเป็น 3 ส่วน คือ เพอร์ซันนอลไฟร์วอล ไฟร์วอลแอดมินิสเทรเตอร์ และล็อกมอนิเตอร์ โดยเมื่อเริ่มการทำงานเพอร์ซันนอลไฟร์วอลจะมีการอ่านกฎของไฟร์วอลที่เก็บไว้ที่แอ็คทีฟไดเรกทรอรี เพื่อนำมาเริ่มการทำงานของระบบ ส่วนของเพอร์ซันนอลไฟร์วอลจะทำงานควบคู่กันกับระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ เพื่อทำการตรวจสอบ และ รายงานผล สิ่งผิดปกติ รวมทั้งพฤติกรรมผิดปกติที่เกิดขึ้นกับเครื่องลูกข่าย ไปเก็บไว้ยังฐานข้อมูลล็อกที่เซิร์ฟเวอร์

NETWORK FIREWALL PROGRAM SUITE

Mr. Jumpon	Tummawat	
Mr. Prapas	Phongsanam	
Mr. Surasak	Tubcleio	
Mr. Akkradach	Watcharapupong	Advisor
Mr. Thana	Hongsuwan	Advisor
Mr. Thanunchai	Threepak	Advisor

ABSTRACT

Nowadays, the security of network is more concerned because internet has started showing up in all fact of every day life. That will make the risk of network increase. Firewall is a method used for securing data and resource on a network by limiting network traffic between user and external threats or the Internet. However, current firewall architecture has some limiting of using in the network. It so hard to manage many firewalls when have many hosts in network.

This project is concerned with the design and development of new firewall architecture by using active directory in Microsoft windows 2000 server. It can control and manage rules of personal firewalls at central. This project has 3 parts are personal firewall, firewall administrator and log monitor. At first personal firewall will get rules from active directory for start working with that rules. Personal firewall will working with intrusion detection system for detect who try to attack and report abnormal action to keep at server for analyze by administrator.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้จะไม่สามารถสำเร็จสมบูรณ์ได้ถ้าไม่ได้รับความช่วยเหลือ และความร่วมมือของบุคคลหลาย ๆ ฝ่ายด้วยกันโดยเฉพาะอย่างยิ่งบุคคลผู้ซึ่งเป็นผู้จุดประกายความคิดให้เกิดหัวข้อโครงการนี้ขึ้น นั้นคือท่าน อาจารย์อัครเดช วัชระภูพงษ์ ซึ่งเป็นอาจารย์ที่ปรึกษา รวมทั้งท่านอาจารย์ทุกท่านในภาควิชา วิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้ให้การอบรมสั่งสอน และให้วิชาความรู้ต่าง ๆ ที่ดีแก่คณะผู้จัดทำเสมอมา

ขอขอบพระคุณท่านที่สำคัญที่สุดในชีวิตนี้ ที่ได้ให้กำเนิด ให้การอบรมดูแล และเอาใจใส่ ทั้งด้านการศึกษา ด้านการดำเนินชีวิต และด้านอื่นทุกด้าน ที่คงไม่อาจมีใครอีกแล้ว ที่เสมอเหมือนท่านทั้งสองนี้ นั้นคือ บิดา และ มารดา ผู้ซึ่งเป็นที่เคารพรักอย่างยิ่ง ผู้ซึ่งคอยให้กำลังใจในยามที่ไม่มีใครเหลือ ผู้ซึ่งคอยชี้นำทางที่ถูกเสมอ ผู้ที่ให้การสนับสนุนในการทำสิ่งที่ถูก และให้คำชี้นำหากสิ่งที่เราทำไม่ใช่สิ่งที่ดี ไม่ใช่สิ่งที่ควร จึงขอกราบขอบพระคุณมา ณ ที่นี้

สุดท้ายนี้ ขอขอบคุณภาควิชาวิศวกรรมคอมพิวเตอร์ โดยเฉพาะห้องวิจัย และพัฒนาการรักษาความปลอดภัยข้อมูล (ISAG) และ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้เอื้อเฟื้อสถานที่ ให้คณะผู้จัดทำได้ทำการวิจัย และช่วยอำนวยความสะดวกต่าง ๆ ขอขอบคุณเพื่อน ๆ พี่ ๆ น้อง ๆ ชาว ISAG ที่คอยให้ความช่วยเหลือ ในการทำงาน ตลอดเวลา ทางคณะผู้จัดทำขอขอบพระคุณมา ณ ที่นี้ด้วย

นายจุมพล ทุมมาวัด
นายประภาส ผ่องสนาม
นายสุรศักดิ์ ทับเคลียว

สารบัญ

สารบัญ(ต่อ)

สารบัญ(ต่อ)

สารบัญ(ต่อ)

สารบัญตาราง

สารบัญรูป

สารบัญรูป(ต่อ)

บทที่1
บทนำ

1.1 ความสำคัญและที่มา

ในปัจจุบันระบบเครือข่ายคอมพิวเตอร์ มีการใช้งานกันอย่างแพร่หลาย และมีจำนวนเพิ่มมากขึ้น ซึ่งทำให้เกิดความยุ่งยากซับซ้อนในการที่จะจัดการดูแล โดยเฉพาะในด้านการรักษาความปลอด ภัยซึ่งอาจทำให้การรักษาความปลอดภัยทำได้ไม่เต็มประสิทธิภาพเท่าที่ควร และ อาจเกิดความเสี่ยงมากยิ่งขึ้นด้วย

ดังนั้นจึงได้มีการพัฒนาไฟร์วอล พร้อมทั้งระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ ที่มีความสามารถในการจัดการ และควบคุมการทำงาน รวมทั้งการกำหนดกฎควบคุมต่าง ๆ ของ ไฟร์วอลให้สามารถควบคุมได้ที่จุดศูนย์กลางเพียงจุดเดียว โดยใช้ความสามารถของแอ็คทีฟไดเรกทอรีเซอร์วิส (Active Directory Service) บนระบบปฏิบัติการวินโดวส์ 2000 โดยที่เมื่อผู้ใช้ทำการล็อกอินเข้าใช้งานเครื่องในเครือข่าย จะได้รับกฎของไฟร์วอล จากเครื่องเซอร์ฟเวอร์เพื่อนำมาใช้ เริ่มการทำงานของไฟร์วอลต่อไป และเมื่อเกิดการโจมตีเกิดขึ้นที่เครื่องลูกข่าย จะมีการแจ้งเตือนขึ้น และมีการส่งข้อมูลการโจมตีที่ตรวจจับได้ไปเก็บไว้ในฐานข้อมูลที่เครื่องเซอร์ฟเวอร์กลางด้วย

จากหลักการทำงานดังกล่าวทำให้ผู้ดูแลระบบสามารถที่จะกำหนดกฎการรักษาความปลอดภัยได้อย่างทั่วถึง และ คลอบคลุมอีกทั้งทราบถึงเหตุการณ์ผิดปกติที่เกิดขึ้นกับเครื่องในเครือข่ายด้วยโดยที่สามารถดูได้จาก ล็อกไฟล์ที่เก็บอยู่ในฐานข้อมูลของเครื่องเซอร์ฟเวอร์กลาง

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อศึกษาโครงสร้างและหลักการทำงานของไฟร์วอล บนระบบปฏิบัติการวินโดวส์
2. เพื่อศึกษาหลักการทำงานและวิธีการทำงานของระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์
3. เพื่อเขียนโปรแกรมติดต่อ และประยุกต์ใช้งานร่วมกับระบบแอ็คทีฟไดเรกทอรีเซอร์วิส ของวินโดวส์ 2000

1.3 ขอบเขตของโครงงาน

การพัฒนาโครงงานนี้แบ่งออกเป็น 3 ส่วนด้วยกันคือ

1. ไฟร์วอลแอดมินิสเทรเตอร์ (Firewall Administrator) ที่ทำงานอยู่บนระบบปฏิบัติการวินโดวส์ 2000 ซึ่งจะทำหน้าที่ในการ กำหนดกฎของเครื่องลูกข่ายในแต่ละกลุ่มผู้ใช้งาน
2. ล็อกมอนิเตอร์ (Log Monitor) เป็นส่วนของการจัดการเกี่ยวกับการเก็บ ล็อกไฟล์ที่รับมาจากเครื่องภายเครือ ข่ายจัดเก็บลงฐานข้อมูล และ จัดการเกี่ยวกับการแสดงผลในรูปแบบต่าง ๆ
3. เพอร์ซันนอลไฟร์วอล พร้อมทั้งระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ (Personal Firewall with Network Intrusion Detection System) ซึ่งทำงานบนระบบปฏิบัติการวินโดวส์ 2000 ทั้งสอง

ส่วนจะทำงานควบคู่กันโดยที่เมื่อมีการตรวจพบการโจมตีเกิดขึ้น จะมีการแจ้งและส่งล็อกไฟล์ไปเก็บไว้ที่เซอร์ฟเวอร์กลาง

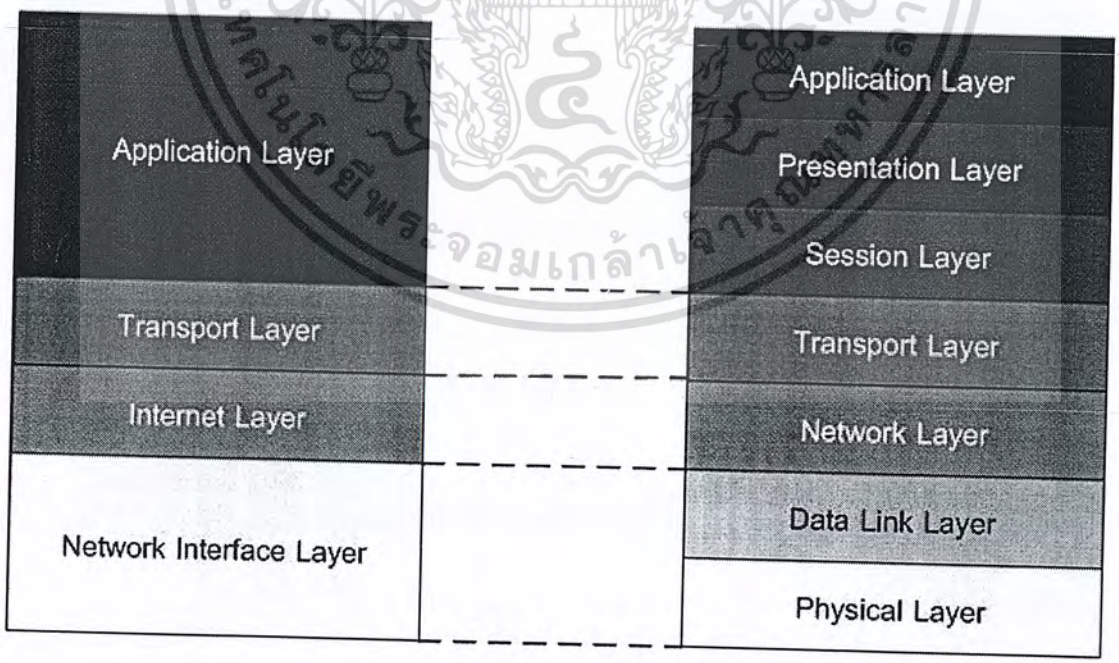
1.4 ขั้นตอนการดำเนินงาน

1. ศึกษารายละเอียดเกี่ยวกับโพรโตคอล ทีซีพี/ไอพี รวมทั้งโพรโตคอลอื่นที่เกี่ยวข้อง
2. ศึกษาเกี่ยวกับการทำงานและวิธีการทำงานของระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์
3. ศึกษาเกี่ยวกับหลักการทำงาน และวิธีการทำงานของไฟร์วอล บนระบบปฏิบัติการวินโดวส์ 2000
4. ศึกษาการทำงานของระบบแอ็คทีฟไดเรกทอรีบนระบบปฏิบัติการวินโดวส์ 2000 และการเขียนโปรแกรมเพื่อประยุกต์ใช้งานร่วมกับระบบแอ็คทีฟไดเรกทอรี
5. พัฒนาโปรแกรมป้องกันการโจมตีแบบไดเรกทอรีเบส (Directory based Firewall) พร้อมทั้งระบบตรวจจับผู้บุกรุกทางเครือข่ายที่สามารถ จัดการเกี่ยวกับกฎ และ การส่งล็อกไฟล์ที่ศูนย์กลางได้

บทที่2
โปรโตคอลทีซีพี/ไอพี

2.1 ความเป็นมาของโปรโตคอล ทีซีพี/ไอพี

ทีซีพี/ไอพี เป็นมาตรฐานการรับส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์สองระบบ ที่มีขึ้นเมื่อกระทรวงกลาโหมสหรัฐฯ หรือ Department Of Defense (DOD) ทำการทดลองในปี ค.ศ.1969 เชื่อมโยงคอมพิวเตอร์ทางทหารของแต่ละหน่วย ซึ่งเป็นคอมพิวเตอร์ต่างชนิดกันให้สามารถติดต่อรับส่งข้อมูลกันได้ โครงการนี้มีชื่อว่า Advanced Research Projects Agency Network หรือ ARPANET ซอฟต์แวร์ที่ใช้ควบคุมการรับส่งข้อมูล ARPANET ประกอบด้วยส่วนหลัก ๆ 2 ส่วนคือ ทีซีพี (Transmission Control Protocol หรือ TCP) และ ไอพี (Internet Protocol หรือ IP) ซึ่งทีซีพีมีหน้าที่ตรวจสอบการรับส่งข้อมูลระหว่างคอมพิวเตอร์ระหว่างผู้รับ และผู้ส่ง ให้ได้รับข้อมูลถูกต้องครบถ้วน ส่วนไอพีจะมีหน้าที่เลือกเส้นทางที่ใช้รับส่งข้อมูล ผ่านระบบเครือข่าย และตรวจสอบที่แอดเดรสของผู้รับ เรียกว่าไอพีแอดเดรส (IP Address) ต่อมาในปี ค.ศ. 1983 ทีซีพี/ไอพี ถูกกำหนดให้เป็นมาตรฐานการรับส่งข้อมูลของกระทรวงกลาโหมสหรัฐฯ และได้รวมเป็นส่วนหนึ่งของระบบปฏิบัติการยูนิกซ์ ส่งผลให้มีการใช้งานกันอย่างแพร่หลาย ในปันจุบันมีการใช้งานกันอยู่แทบทุกเครือข่าย ไม่ว่าจะเป็นเครือข่ายเฉพาะที่ หรือเครือข่ายในบริเวณกว้าง ทีซีพี/ไอพีเชื่อมกลุ่ม เครือข่ายย่อยเข้าด้วยกันเป็นเครือข่ายขนาดใหญ่ หรืออินเทอร์เน็ต



รูปที่ 2-1 แสดงการเปรียบเทียบเลเยอร์ของโอเอสไอกับเลเยอร์ของ ทีซีพี/ไอพี

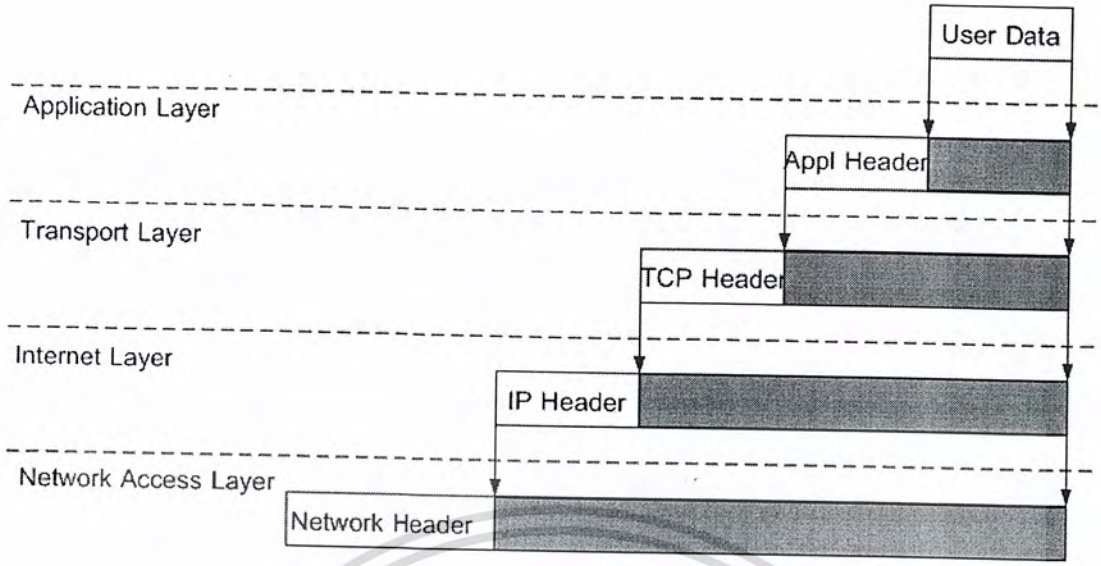
2.2 การเชื่อมต่อของโปรโตคอล ทีซีพี/ไอพี (TCP/IP Linking)

ทีซีพี/ไอพี (Transmission Control Protocol/Internet Protocol) เป็นโปรโตคอลในการสื่อสารในระบบอินเทอร์เน็ต และ อินทราเน็ต การทำงานของทีซีพี/ไอพี สามารถเปรียบเทียบกับโมเดลอ้างอิงโอเอสไอ (Open System Interconnection Reference Model: OSI) ตามมาตรฐานไอเอสโอ (International Organization for Standardization: ISO) ได้ดังรูปที่ 2-1

แต่ในระดับชั้นของ ทีซีพี/ไอพี (TCP/IP) จะมีการทำงานที่แตกต่างกัน ตั้งแต่การติดต่อกับแอพพลิเคชัน จนกระทั่งแปลงเป็นสัญญาณส่งไปตามสายสัญญาณ ซึ่งการทำงานในแต่ละระดับชั้นของ ทีซีพี/ไอพี มีดังตารางที่ 2-1

ชื่อระดับชั้น	หน้าที่
1. ชั้นแอพพลิเคชัน (Application Layer)	ชั้นนี้รองรับการทำงานของแอพพลิเคชันต่าง ๆ ที่ทำงานเป็นโปรเซสอยู่ในเครื่องต้นทาง และ ปลายทาง โดยจัดการเชื่อมต่อระหว่างโปรเซส หรือแอพพลิเคชันที่อยู่ต่างเครื่องกัน โดยการทำงานของแอพพลิเคชันต่าง ๆ มีการติดต่อกันตามแต่ละโปรโตคอล เฉพาะแล้วแต่แอพพลิเคชัน ที่ใช้งานซึ่งจะขอบริการจากชั้นทรานสพอร์ตอีกทีหนึ่ง
2. ชั้นทรานสพอร์ต (Transport Layer)	มีการสร้างการเชื่อต่อกันระหว่างแอพพลิเคชันแบบ end-to-end โดยจุดที่เชื่อมต่อกันเพื่อรับส่งข้อมูลนี้เรียกว่า พอร์ต(Port) หรือ ซ็อกเก็ต (Socket) ในชั้นนี้มีการบริการหลักอยู่ 2 แบบคือ Connection Oriented โดยเรียกผ่านโปรโตคอล ทีซีพี (TCP: Transmission Control Protocol) และ Connectionless ซึ่งเรียกผ่านโปรโตคอลยูดีพี (UDP: User Datagram Protocol) ซึ่งจะกล่าวถึงในหัวข้อถัดไป
3. ชั้นอินเทอร์เน็ต (Internet Layer)	ชั้นนี้มีหน้าที่ส่งผ่านข้อมูลระหว่างเครือข่าย โดยมีโปรโตคอลที่ทำงานเป็นกลไกสำคัญในการส่งผ่านข้อมูลไปยังเครือข่ายใด ๆ ในอินเทอร์เน็ต คือ ไอพี(IP) ซึ่งกล่าวถึงในหัวข้อถัดไป นอกจากนี้ในชั้นนี้ยังมีโปรโตคอลทำงานอยู่ด้วยอีก 2 ชนิดคือ ไอซีเอ็มพี (ICMP) และ เออาร์พี (ARP)
4. ชั้นเน็ตเวิร์กอินเตอร์เฟส (Network Interface Layer)	ทำหน้าที่ในการแปลงข้อมูลให้อยู่ในรูปที่เหมาะสมกับเครือข่ายแต่ละแบบ ซึ่งแตกต่างกันออกไป และแปลงเป็นสัญญาณ ไฟฟ้าส่งไปยังเครือข่าย

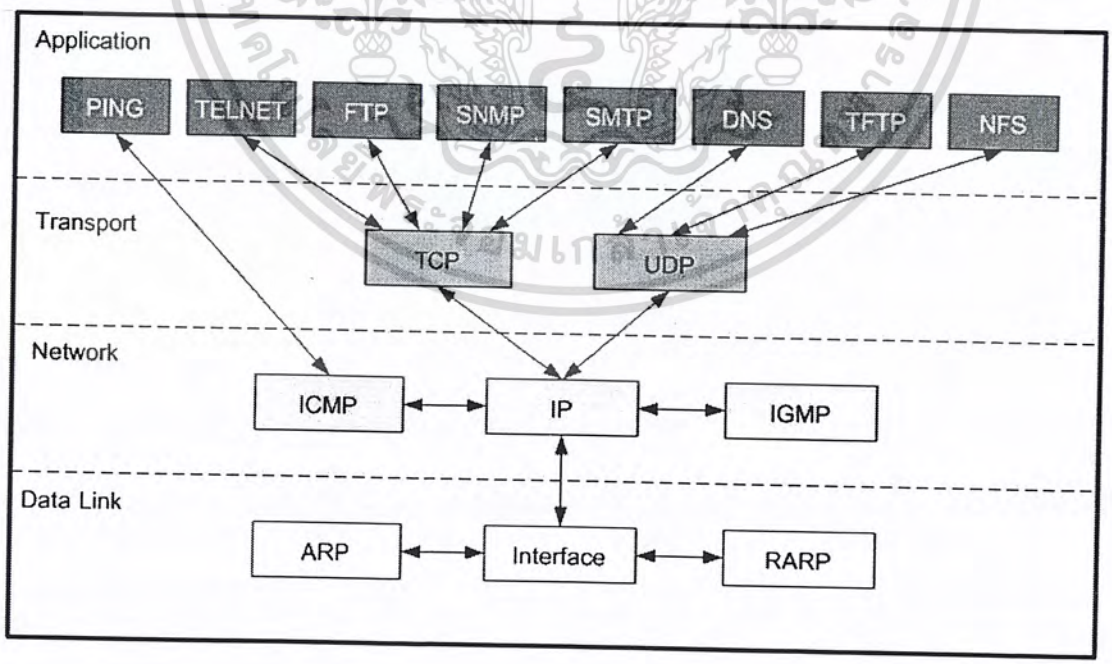
ตารางที่ 2-1 การทำงานของแต่ละระดับชั้นของ ทีซีพี/ไอพี



รูปที่ 2-2 แสดงการห่อหุ้มข้อมูลตามลำดับโปรโตคอลแสตก

2.3 โปรโตคอลแสตก

การทำงานตามโปรแกรมประยุกต์หนึ่ง ๆ ไม่ได้ใช้โปรโตคอลพร้อมกันทั้งชุด หากแต่ใช้เพียงโปรโตคอลที่สัมพันธ์กันไปในแต่ละระดับชั้นของแบบอ้างอิงตัวอย่างเช่นการใช้งาน เทลเน็ต (Telnet) ที่จะอาศัย ทีซีพี/ไอพี (TCP/IP) ตามลำดับ การซ้อนทับของโปรโตคอลจากระดับชั้นบนไปชั้นล่าง เรียกว่าโปรโตคอลแสตก (Protocol Stack) แสดงดังรูปที่ 2-3



รูปที่ 2-3 โปรโตคอลแสตกของ ทีซีพี/ไอพี

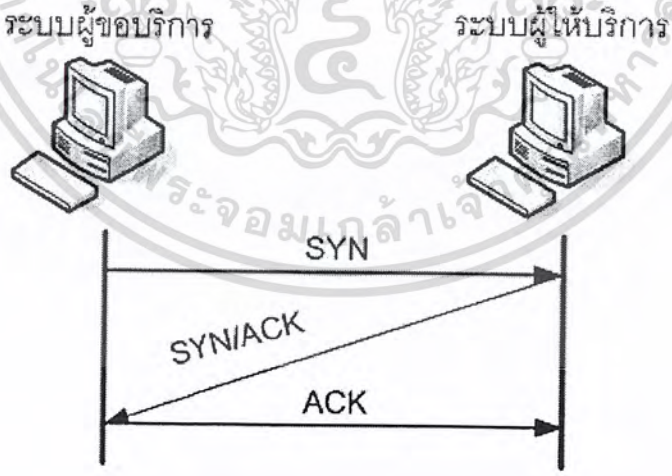
ไอพีซึ่งอยู่ในระดับชั้นเน็ตเวิร์คตามรูปที่ 2-3เป็นแกนสำคัญของโปรโตคอลแสตกเนื่องจากทั้งทีซีพี(TCP) และ ยูดีพี (UDP) ต้องใช้ไอพีเลือกเส้นทางส่งแพ็กเก็ต ในระดับชั้นเน็ตเวิร์ค ยังมีไอซีเอ็มพี (ICMP) สนับสนุนการทำงานของไอพีเพื่อรายงานข้อผิดปกติที่เกิดขึ้น เนื่องจากการส่งแพ็กเก็ต และมีไอจีเอ็มพี (IGMP) ดูแลการจัดกลุ่มโฮสต์ในเครือข่ายมัสติคาสต์

ระดับชั้นทรานสพอร์ตมีสองโปรโตคอลสำคัญคือ ทีซีพีและยูดีพี แอพลิเคชันจะเลือกใช้ทีซีพีหรือยูดีพีตามลักษณะงาน

โปรโตคอลระดับล่างถัดจากไอพีได้แก่ โปรโตคอลระดับเดทาลิงค์ซึ่งกำหนดการทำงานตามเทคโนโลยีเครือข่ายที่ใช้งาน เช่น โปรโตคอลระดับเดทาลิงค์ ซึ่งกำหนดการทำงานตามมาตรฐานอีเทอร์เน็ต ในระดับชั้นนี้มีโปรโตคอลในชุดของทีซีพี/ไอพี ทำหน้าที่สนับสนุนการทำงานอยู่สองโปรโตคอล คือ เออาร์พีและอาร์เออาร์พี ทั้งสองโปรโตคอลจะทำหน้าที่แปลงค่าระหว่างไอพีแอดเดรสกับฮาร์ดแวร์แอดเดรส

2.4 โปรโตคอลทีซีพี (TCP: Transmission Control Protocol)

การทำงานที่สำคัญอย่างหนึ่งที่สำคัญของโปรโตคอลทีซีพีคือการทำ"3-way Handshake" ซึ่งเป็นกระบวนการเริ่มต้นในการสร้างการเชื่อมต่อในชั้นทรานสพอร์ต กล่าวคือ ในการติดต่อกันระหว่างระบบในเครือข่ายต้องมีการสร้างการเชื่อมต่อ ไปยังระบบที่ ให้บริการก่อนโดยผู้ขอบริการส่งสัญญาณ SYN เพื่อขอบริการ จากนั้นผู้ให้บริการจะส่งสัญญาณ ACK เพื่อตอบรับการเชื่อมต่อที่ร้องขอมาจึงสามารถรับส่งข้อมูลได้ แสดงดังรูปที่ 2-4



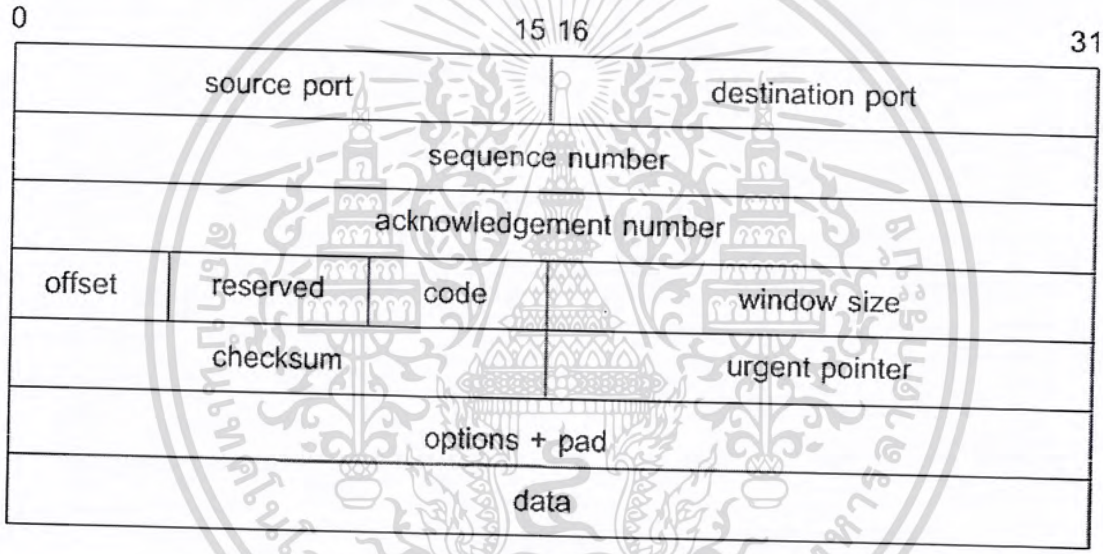
รูปที่ 2-4 แสดงการทำ 3-way Handshake

การเชื่อมต่อแบบ 3-way Handshake นี้เป็นการตรวจสอบความพร้อมของทั้งฝ่ายส่ง และผ่ายรับ และเป็นการกำหนดค่าเริ่มต้นของพารามิเตอร์ต่าง ๆ ของทั้งสองฝ่ายให้ตรงกัน หลังจากกระบวนการทำ 3-way Handshake สิ้นสุด ทั้งสองฝ่ายจึงสามารถรับและ ส่งข้อมูลซึ่งกันและกันได้

ดังนั้นโปรโตคอลทีซีพีจึงเป็นโปรโตคอลที่มีการรับส่งข้อมูลแบบ “Connection Oriented” ทำให้การทำงานของทีซีพีมีความน่าเชื่อถือมากขึ้น หน้าที่การทำงานของทีซีพี ในการรับส่งข้อมูลมีหน้าที่หลัก 6 ข้อคือ

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

ส่วนประกอบของทีซีพีเฮดเดอร์



รูปที่ 2-5 ทีซีพีเฮดเดอร์

- source port ขนาด 16 บิต: เป็นหมายเลขพอร์ตของบริการที่เครื่องต้นทาง
- destination port ขนาด 16 บิต: เป็นหมายเลขพอร์ตของบริการเครื่องปลายทาง
- sequence number ขนาด 32 บิต: เป็นหมายเลขที่บอกลำดับของการรับส่งข้อมูลของเครื่องที่ต้องการขอส่งข้อมูล
- acknowledgement Number ขนาด 32 บิต: เป็นหมายเลขที่บอกลำดับของการรับส่งข้อมูลที่ฝั่งรับข้อมูลปกติค่าของ Acknowledgement Number จะมีค่าเท่ากับ Sequence Number (ของอีกฝั่ง) + 1 เสมอ
- data offset ขนาด 4 บิต: เป็นตัวบอกค่าออฟเซตของข้อมูล เพราะทีซีพีนั้นไม่มีการกำหนดความยาวที่แน่นอนของข้อมูล จึงต้องมีออฟเซตเป็นตัวบอก
- code ประกอบด้วย 6 ฟิลด์ย่อยแต่ละฟิลด์ย่อยมีขนาด 1 บิต ได้แก่

URG	ACK	PSH	RST	SYN	FIN

- URGent ถ้าบิตนี้เป็น "1" หมายความว่า Urgent pointer บรรจุตำแหน่งข้อมูลที่ต้องรีบดำเนินการเร่งด่วนก่อน
- ACKnowledgement ถ้าบิตนี้เป็น "1" หมายถึงเป็นเซกเมนต์ตอบรับ โดยตอบอ้างอิงเลขลำดับตามที่กำหนดในฟิลด์ Acknowledgement number
- PuSH ถ้าบิตนี้เป็น "1" หมายความว่าทันทีที่สถานีปลายทางได้รับเซกเมนต์ ต้องรีบส่งข้อมูลไปยังโปรโตคอลประยุกต์ทันทีโดยไม่ต้องรอให้บัฟเฟอร์เต็ม
- ReSeT ถ้าบิตนี้เป็น "1" หมายถึงให้ยกเลิกการเชื่อมต่อนี้ เนื่องจากอาจมีความผิดปกติเกิดขึ้นระหว่างคู่สถานีที่กำลังติดต่อกันอยู่ หากจำเป็นต้องส่งข้อมูลระหว่างกันอีกก็ต้องเริ่มสถาปนาการเชื่อมต่อใหม่
- SYNchronize ถ้าบิตนี้เป็น "1" หมายถึงขอเริ่มต้นสถาปนาการเชื่อมต่อและเมื่อการสถาปนาเสร็จสิ้น บิตนี้จะถูกกำหนดให้เป็น "0" หลังจากนั้นจึงสามารถส่งผ่านข้อมูลระหว่างกันได้
- FINish ถ้าบิตนี้เป็น "1" หมายถึงขอจบการเชื่อมต่อ

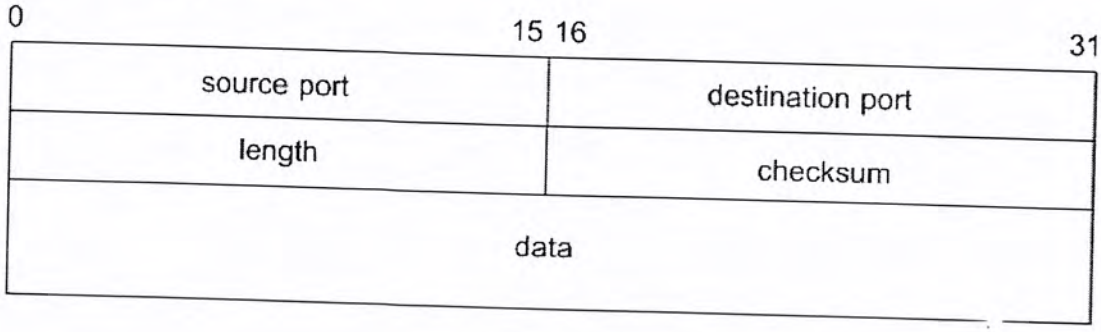
- windows size ขนาด 16 บิต: สถานีปลายทางใช้ฟิลด์นี้แจ้งขนาดบัฟเฟอร์ที่มีอยู่ (หน่วยเป็นไบต์) สถานีที่ติดต่อด้วยต้องไม่ส่งข้อมูลเกินค่านี้
- checksum ขนาด 16 บิต: ผลรวมตรวจสอบความถูกต้องของเซกเมนต์โดยคำนวณทั้งเฮดเดอร์และ ข้อมูล
- urgent pointer ขนาด 16 บิต: พอยเตอร์ชี้ตำแหน่งไบต์ข้อมูลที่ต้องดำเนินการเร่งด่วนที่ต้องการให้โปรแกรมประยุกต์ดำเนินการทันที ค่าที่บรรจุในฟิลด์นี้จะมีความหมายก็ต่อเมื่อแฟล็ก URG ถูกเซตเป็น "1"
- options ขนาดแปรเปลี่ยนได้: ใช้กำหนดงานเพิ่มเติมให้กับทีซีพีซึ่งจะมีหรือไม่ก็ได้ หากฟิลด์ offset หากมีค่าเป็น 5 แสดงว่ามีเฮดเดอร์ขนาด 20 ไบต์ซึ่งหมายถึงไม่ใช่ออพชัน
- pad ขนาด 0 ถึง 24: ใช้เป็นส่วนที่ทำให้ขนาดของออพชันเป็นจำนวนเท่าของ 32 บิต

2.5 โปรโตคอลยูดีพี (UDP: User Datagram Protocol)

โปรโตคอลยูดีพีเป็นโปรโตคอลในการติดต่อสื่อสารในชั้นทรานสปอร์ต (Transport Layer) การทำงานคล้ายกับทีซีพีมาก คือ จัดการเกี่ยวกับการสื่อสารระหว่างเครื่อง แต่เป็นแบบ Connectionless คือ ทั้งฝ่ายส่งและฝ่ายรับไม่จำเป็นต้องอาศัยการสร้างช่องทางเชื่อมต่อกัน โดยไม่ต้องมีการแจ้งให้ฝ่ายรับข้อมูลเตรียมรับข้อมูลเหมือนโปรโตคอลทีซีพี และ ไม่มีการส่งสัญญาณตรวจสอบว่าข้อมูลถึงปลายทางอย่าง

ถูกต้องครบถ้วนสมบูรณ์ ในการส่งข้อมูลแต่ละครั้งจึงไม่มี การส่งข้อมูลใหม่อีกในกรณีที่เกิดความผิดพลาดของการส่งข้อมูล

ส่วนประกอบของยูดีพีเดทาแกรม



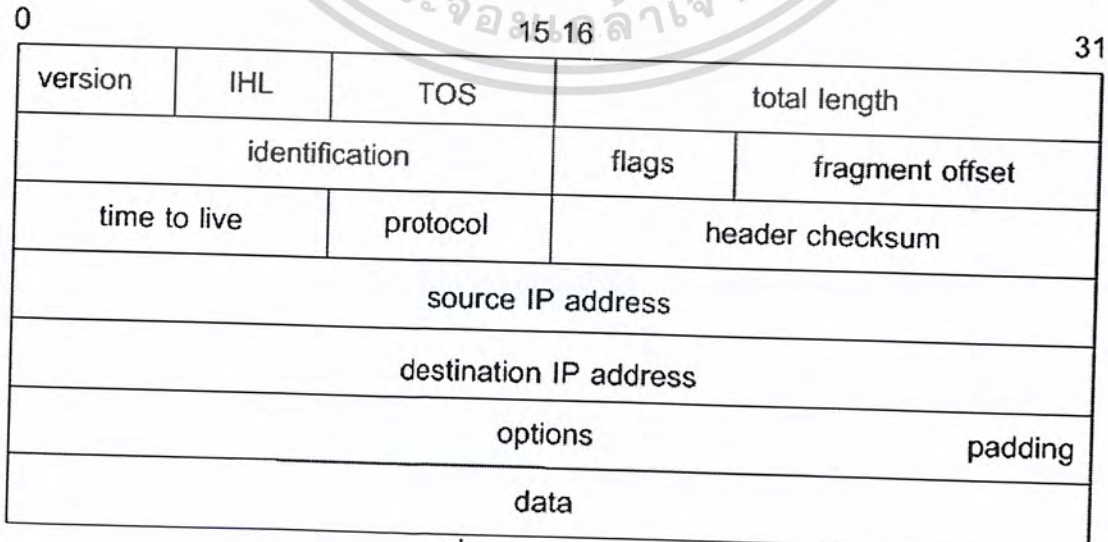
รูปที่ 2-6 ยูดีพีเดทาแกรม

- source port ขนาด 16 บิต: บอกพอร์ตของเครื่องต้นทาง
- destination port ขนาด 16 บิต: บอกพอร์ตของเครื่องปลายทาง
- length ขนาด 16 บิต: บอกความยาวของเดทาแกรม (ทั้งเฮดเดอร์และข้อมูล) เป็นจำนวนไบต์
- checksum ขนาด 16 บิต: ผลรวมตรวจสอบ คำนวณจากผลรวมของเฮดเดอร์และข้อมูล

2.6 โปรโตคอลไอพี (IP: Internet Protocol)

โปรโตคอลไอพีเป็นโปรโตคอลที่จัดการเกี่ยวกับแอดเดรสของแต่และแพ็กเก็ต เพื่อให้ส่งแพ็กเก็ตต่าง ๆ ไปยังเป้าหมายได้อย่างถูกต้อง การทำงานของไอพีเป็นเพียงการส่งข้อมูลไปยังเครื่องเป้าหมายเท่านั้นไม่มีการส่งสัญญาณขอบริการ หรือสัญญาณให้บริการระหว่างกันเหมือนทีซีพี เรียกว่าการเชื่อมต่อแบบ Connectionless ซึ่งระบบทั้งสองตั้งสมมุติฐานว่าการเชื่อมต่อระหว่างกันไม่มีความผิดพลาดเกิดขึ้น

ส่วนประกอบของไอพีเดทาแกรม

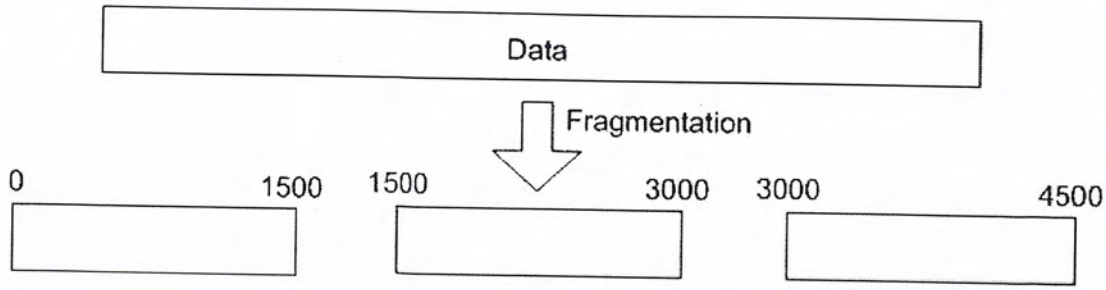


รูปที่ 2-7 ไอพีเดทาแกรม

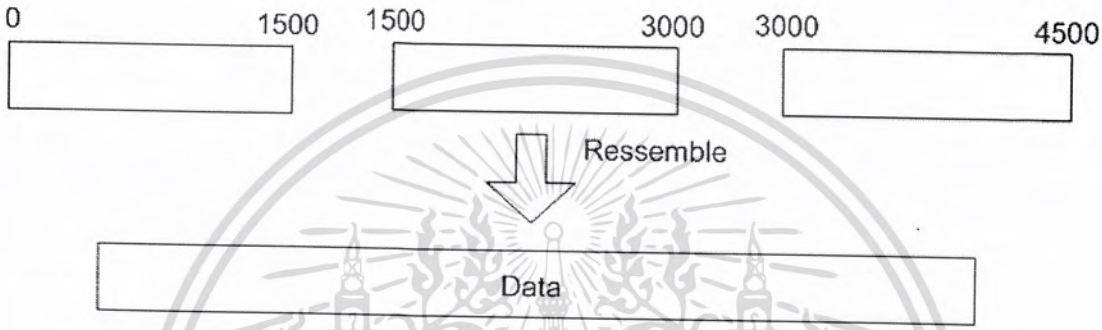
- version ขนาด 4 บิต: แสดงรุ่นของโปรโตคอล
- Internet Header Length (IHL) ขนาด 4 บิต: บอกความยาวเฉพาะเฮดเดอร์ ของเดทา แกรมโดยนับจาก version จนถึงไบต์สุดท้ายก่อนจะถึงข้อมูล
- Type Of Service (TOS) ขนาด 8 บิต: ฟิลด์นี้ใช้กำหนดรูปแบบการให้บริการตามลักษณะโปรโตคอลแอพพลิเคชัน
- total length มีขนาด 16 บิต: บอกถึงความยาวทั้งหมดของเดทาแกรม
- identification ขนาด 16 บิต
- flags ขนาด 3 บิต
- Time To Live (TTL) ขนาด 8 บิต: ฟิลด์นี้ใช้กำหนดจำนวนเราเตอร์ที่เดทาแกรมจะเดินทางผ่านไปได้อีกนัยหนึ่งคือ กำหนดอายุของเดทาแกรมซึ่งมีค่าสูงสุดตามขนาดฟิลด์คือ 2^8-1= 255
- Protocol ขนาด 8 บิต:ฟิลด์บอกชนิดของโปรโตคอลในระดับบนที่เอ็นแคปซูเลตในเดทาแกรม เพื่อให้สถานีปลายทางสามารถส่งข้อมูลไปยังโปรโตคอลระดับบนได้อย่างถูกต้อง
- header checksum ขนาด 16 บิต: ใช้สำหรับตรวจสอบความผิดพลาดของเฮดเดอร์โดยไม่รวมส่วนของข้อมูล
- source IP address ขนาด 32 บิต: กำหนดไอพีแอดเดรสต้นทาง
- destination IP address ขนาด 32 บิต: กำหนดไอพีแอดเดรสปลายทาง
- option ขนาดไม่คงที่: ใช้สำหรับกำหนดข่าวสารเพิ่มเติมสำหรับเดทาแกรม ค่าที่ใช้ในปัจจุบันเกี่ยวข้องกับการรักษาความปลอดภัย
- padding ขนาด 0 ถึง 3 ไบต์: ใช้สำหรับผนวกเพิ่มเพื่อให้จำนวนไบต์ของ option รวมกับ padding เป็นจะนวนเท่าของ 32 บิต
- data ขนาดไม่คงที่: ข้อมูลของโปรโตคอลระดับบน

เนื่องจากมาตรฐานในเครือข่ายมีหลากหลาย ขนาดของแพ็กเก็ตในแต่ละมาตรฐานจึงมีความแตกต่างกันออกไป ทำให้การส่งข้อมูลระหว่างอุปกรณ์ในเครือข่ายนั้นอาจมีการแบ่งข้อมูลออกเป็นแพ็กเก็ตย่อย ๆ ในระหว่างการส่ง เรียกว่าการทำแฟล็กเมนเตชัน (Fragmentation) เช่นแพ็กเก็ต FDDI มีขนาด 4,500 ไบต์ หากเครื่องปลายทางอยู่ในเครือข่าย Ethernet ซึ่งมีขนาดของแพ็กเก็ตสูงสุด 1,500 ไบต์ดังนั้นการส่งแพ็กเก็ตไปยังเครื่องปลายทางจึงต้องมีการ แบ่งเป็นเพ็กเก็ตย่อย ๆ และเมื่อแพ็กเก็ตย่อยมาถึงเครื่องเป้าหมายก็จะรวมกันเป็นแพ็กเก็ตเดิมที่มีขนาด 4,500 ไบต์ อีกครั้งหนึ่ง เรียกการรวมกันนี้ว่าการรีแอสเซม-เบิ้ล (Reassemble) ซึ่งทำให้ได้ข้อมูลเหมือนกับที่ส่งมาจากเครื่องต้นทาง

เดทาแกรมที่ถูกแฟล็กเมนจะมีเฮดเดอร์ประจำตัวเอง โดยมีข้อมูลเพิ่มเติมกำหนดลักษณะของแฟล็กเมนประกอบไปด้วย แต่ละชิ้นของแฟล็กเมนจะเป็นเดทาแกรมที่สมบูรณ์ในตัวเอง เราเตอร์ระหว่างทางที่รับเดทาแกรมจะไม่รวมเดทาแกรม (Reassembly) แต่จะปล่อยให้เป็นหน้าที่ของเราเตอร์ปลายทาง



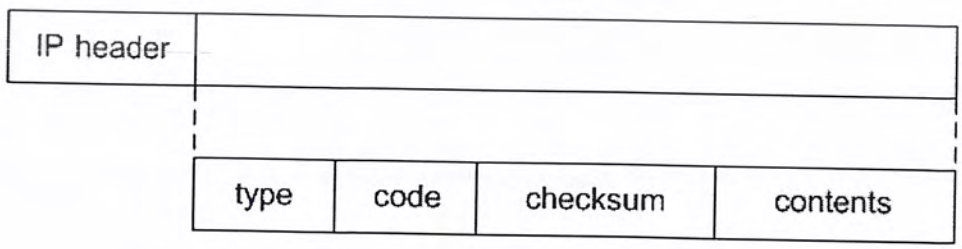
รูปที่ 2-8 แสดงการทำแฟล็กเมนเตชั่น



รูปที่ 2-9 แสดงการรีแอสเซมเบิ้ล

2.7 โปรโตคอลไอซีเอ็มพี (ICMP: Internet Control Message Protocol)

หน้าที่หลักของโปรโตคอล ไอซีเอ็มพี คือการแจ้งหรือแสดงข้อความจากระบบ เพื่อบอกให้ผู้ใช้ทราบว่าเกิดอะไรขึ้นในการส่งผ่านข้อมูลนั้น ซึ่งปัญหาส่วนมากที่พบคือส่งไปไม่ได้ หรือปลายทางรับข้อมูลไม่ได้ เป็นต้น นอกจากนี้โปรโตคอล ไอซีเอ็มพียังถูกเรียกใช้งานจากเครื่องเซอร์ฟเวอร์ และ เราเตอร์อีกด้วยเพื่อแลกเปลี่ยนข้อมูลที่ใช้ควบคุม ส่วนรูปแบบการทำงานของโปรโตคอลไอซีเอ็มพีนั้น จะทำงานควบคู่กับโปรโตคอลไอพี (IP) ในระดับเดียวกัน และข้อความต่าง ๆ ที่แจ้งให้ทราบจะถูกรวมอยู่ภายในข้อมูลของไอพีเดทาแกรม อีกทีหนึ่ง



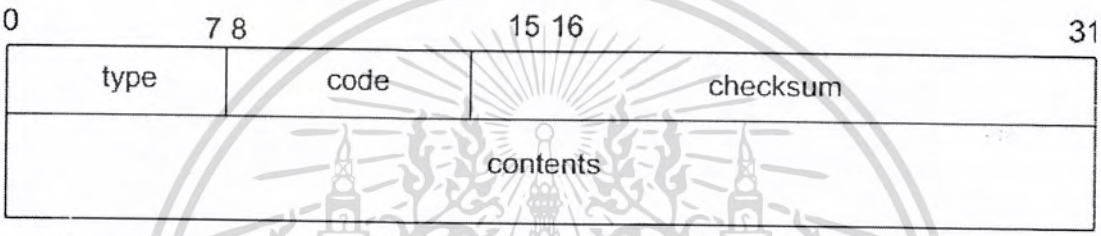
รูปที่ 2-10 ข่าวสารของไอซีเอ็มพีบรรจุในไอพี

ข้อความที่โปรโตคอลไอซีเอ็มพีส่งนั้นแบบออกได้เป็น 2 แบบคือ ICMP error message คือ ข้อความแจ้งข้อผิดพลาด และ ICMP query คือ ข้อความเรียกขอข้อมูลเพิ่ม ตัวอย่างกลไกการทำงานของโปรโตคอ

ไอซีเอ็มพีเช่น เมื่อมีการส่งผ่านข้อมูลจากผู้ใช้ไปยังปลายทางที่ไม่ถูกต้อง หรือขณะนั้นเครื่อปลายทางเกิดปัญหา จะไม่สามารถรับข้อมูลได้ ที่เราท์เตอร์จะส่งข้อความ แจ้งเป็น ไอซีเอ็มพี ที่ชื่อ destination unreachable ให้กับผู้ส่งข้อมูล นอกจากนี้ตัวข้อมูลที่แจ้งข้อความก็จะมีส่วนของข้อมูลไอพีเดทาแกรมที่เกิดปัญหาด้วย ดังนั้นเมื่อผู้ส่งข้อมูลได้รับข้อความแจ้งแล้วก็จะได้ทราบว่าจุดที่เกิดปัญหาอยู่ที่ใด

ดังนั้นโปรโตคอล ไอซีเอ็มพี จึงกลายเป็นเครื่องมืออย่างหนึ่งในการช่วยทดสอบเครือข่าย เช่น คำสั่ง ping ที่เรามักใช้ทดสอบว่าเครื่อเซิร์ฟเวอร์ที่ให้บริการหรืออุปกรณ์ที่ต่อยู่ในเครือข่าย อินเทอร์เนตนั้นยังทำงานเป็นปกติหรือ ไม่ แล้วคำสั่ง ping มีการเรียกใช้งานโปรโตคอล ไอซีเอ็มพี แจ้งเป็นข่าวสารให้ทราบอีกต่อไป

โครงสร้างของไอซีเอ็มพี



รูปที่ 2-11 โครงสร้างของไอซีเอ็มพี

- type ขนาด 8 บิต: กำหนดทั้งค่าความผิดพลาดและการรายงานสถานะ การใช้งานในปัจจุบันมีทั้งหมด 15 ประเภท
- code ขนาด 8 บิต: รหัสความผิดพลาด
- checksum ขนาด 16 บิต: ค่าผลรวมตรวจสอบแบบ 1's complement สำหรับตรวจสอบความผิดพลาด โดยคำนวณผลรวมของ type, code และ contents
- contents ขนาดไม่คงที่ : ฟิลด์นี้ใช้บรรจุข้อมูลข่าวสารเพิ่มเติมเพื่อแจ้งกลับซึ่งจะขึ้นอยู่กับค่า type และ code

บทที่ 3
ไฟร์วอล

ไฟร์วอล คือ เครื่องมือที่ใช้ป้องกัน และ ควบคุมการทำงานของแพ็กเก็ตข้อมูลที่ผ่านเข้าออกภายในเครือข่าย

3.1 ประเภทของไฟร์วอล

ชนิดของไฟร์วอลแบ่งตามเทคโนโลยีที่ใช้ในการตรวจสอบ และควบคุมแบ่งได้เป็น 3 ประ เภท ดังนี้คือ

1. แพ็กเก็ตฟิลเตอริง (Packet Filtering)
2. พร็อกซีเซอร์วิซ (Proxy Service)
3. สเตตฟูลอินสเปกชัน (Stateful Inspection)

3.1.1 แพ็กเก็ตฟิลเตอริง (Packet Filtering)

แพ็กเก็ตฟิลเตอริง (Packet Filtering) คือ เราเตอร์ หรือ ซอร์ฟแวร์ ที่ทำหน้าที่หาเส้นทางส่งต่อ (route) อย่างมีเงื่อนไข โดยจะพิจารณาจากข้อมูลส่วนที่อยู่ในเฮดเดอร์ (header) ของแพ็กเก็ตที่ผ่านเข้ามา เทียบกับกฎ (rules) ที่กำหนดไว้ และตัดสินว่าควรจะทิ้ง (drop) แพ็กเก็ตนั้นไปหรือว่าจะยอม (accept) ให้แพ็กเก็ตนั้นผ่านไปได้ ในการพิจารณาเฮดเดอร์ แพ็กเก็ตฟิลเตอริง จะตรวจสอบในระดับของอินเตอร์เน็ตเลยเยอร์ (Internet Layer) และทรานสพอร์ตเลเยอร์ (Transport Layer) ในอินเตอร์เน็ตโมเดล ซึ่งในอินเตอร์เน็ตเลเยอร์จะมีแอตทริบิวที่สำคัญ ต่อแพ็กเก็ตฟิลเตอริง ดังนี้

- ไอพีต้นทาง
- ไอพีปลายทาง
- ชนิดของโปรโตคอล (TCP UDP และ ICMP)

และในระดับทรานสพอร์ตเลเยอร์มีแอตทริบิวต์ที่สำคัญคือ

- พอร์ตต้นทาง
- พอร์ตปลายทาง
- แฟล็ก (Flag ซึ่งจะมีเฉพาะในเฮดเดอร์ของแพ็กเก็ต TCP)
- ชนิดของ ICMP message (ในแพ็กเก็ต ICMP)

ซึ่งพอร์ตของทรานสพอร์ตเลเยอร์ คือทั้ง TCP และ UDP นั้นจะเป็นสิ่งที่บอกถึงแอพพลิเคชันที่แพ็กเก็ตนั้นต้องการติดต่อด้วยเช่น พอร์ต 80 หมายถึง HTTP, พอร์ต 21 หมายถึง FTP เป็นต้น ดังนั้นเมื่อแพ็กเก็ตฟิลเตอริง (Packet Filter) พิจารณาเฮดเดอร์ จึงทำให้สามารถควบคุมแพ็กเก็ตที่มาจากที่ต่าง ๆ และ มีลักษณะต่าง ๆ ได้

แพ็กเก็ตฟิลเตอริง (Packet Filtering) สามารถอิมพลีเมนต์ได้ 2 แพลตฟอร์ม คือ

- เราเตอร์ที่มีความสามารถในการทำแพ็กเก็ตฟิลเตอริง
- คอมพิวเตอร์ที่ทำหน้าที่เป็นเราเตอร์

ซึ่งจะมีข้อได้เปรียบเสียเปรียบกันดังนี้

	ข้อดี	ข้อเสีย
เราเตอร์	ประสิทธิภาพสูงมีจำนวนอินเตอร์เฟสมาก	เพิ่มเติมฟังก์ชันการทำงานได้ยาก อาจต้องการหน่วยความจำมาก
คอมพิวเตอร์ที่ทำหน้าที่เป็นเราเตอร์	เพิ่มฟังก์ชันการทำงานได้ไม่จำกัด	ประสิทธิภาพปานกลาง, จำนวนอินเตอร์เฟสน้อย , อาจมีความเสี่ยงจากระบบปฏิบัติการที่ใช้

ตารางที่ 3-1 เปรียบเทียบข้อดีข้อเสียในการเลือกอุปกรณ์มาทำหน้าที่แพ็กเก็ตฟิลเตอร์ริ่ง

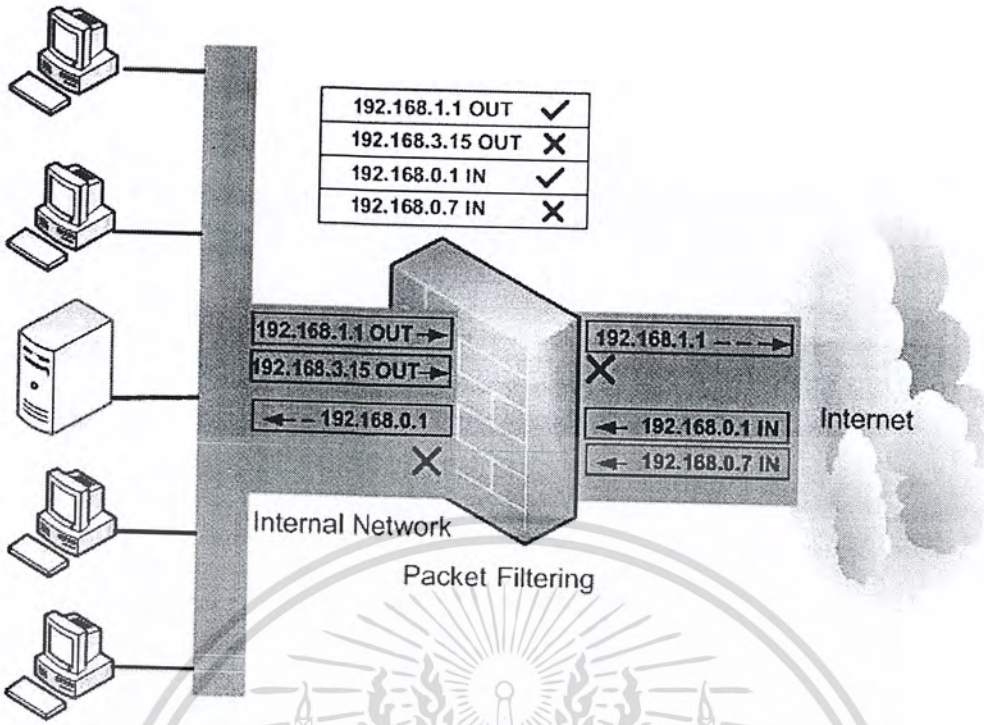
ข้อดี-ข้อเสียของแพ็กเก็ตฟิลเตอริง

ข้อดี

- ไม่ขึ้นกับแอพพลิเคชัน
- มีความเร็วสูง
- รองรับการขยายตัวได้ดี

ข้อเสีย

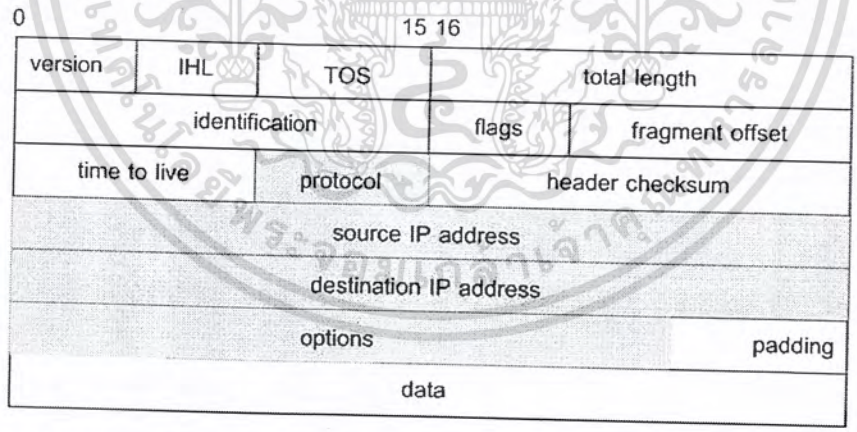
- บางโปรโตคอลไม่เหมาะสมกับการใช้ แพ็กเก็ตฟิลเตอริง (Packet Filtering) เช่น FTP, ICQ

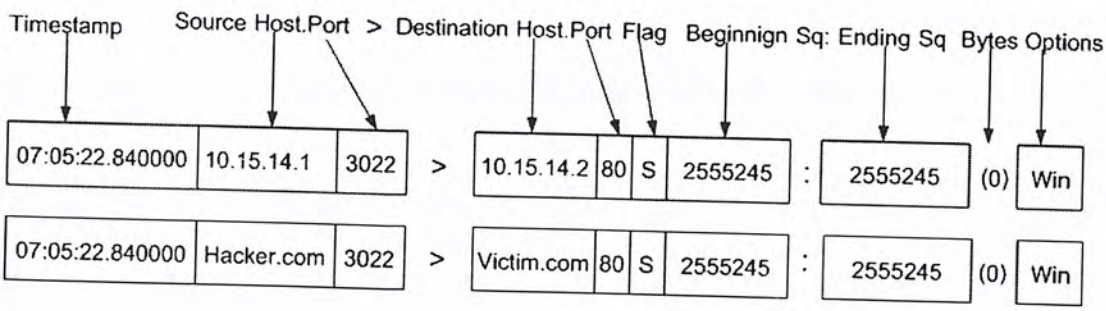


รูปที่ 3-1 แสดงการทำหน้าที่ของแพ็กเก็ตฟิลเตอริง (Packet Filtering)

3.1.1.1 วิธีการอ่านแพ็กเก็ตเพื่อนำมาใช้ฟิลเตอร์

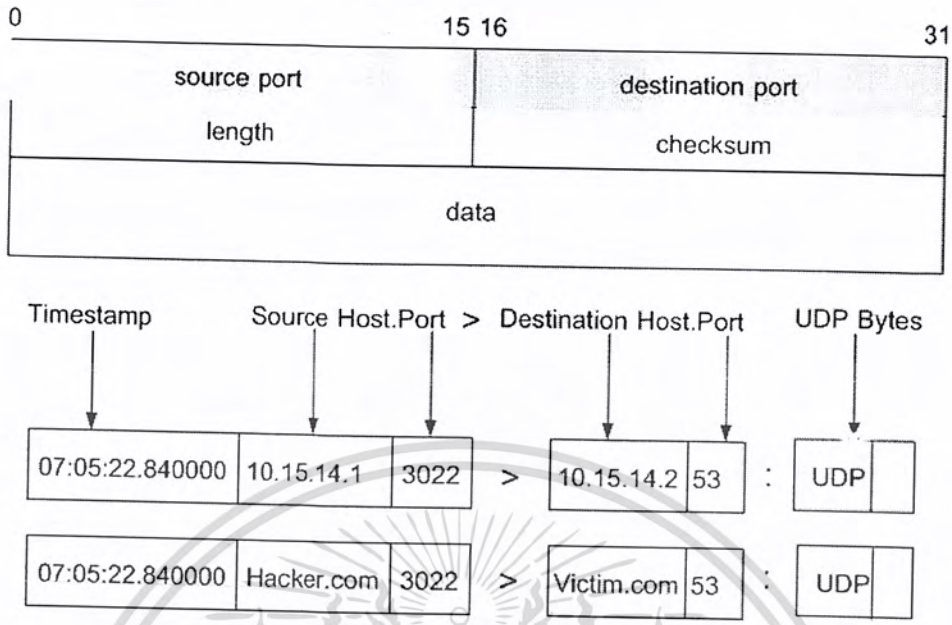
1. โปรโตคอล ทีซีพี/ไอพี (TCP/IP)





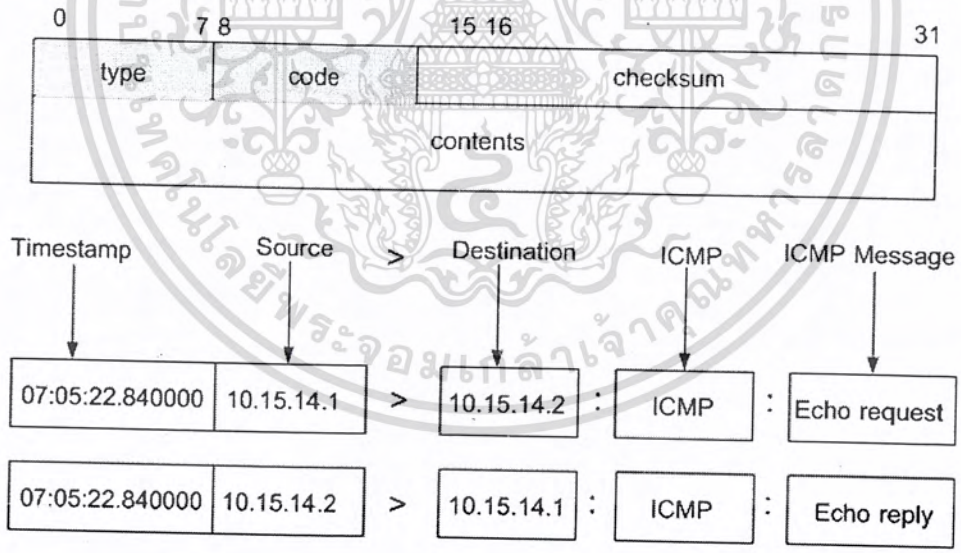
รูปที่ 3-2 แสดงการวิเคราะห์ฟิลด์ต่าง ๆ ของโปรโตคอล ทีซีพี (TCP)

2. โปรโตคอล ยูดีพี (UDP)



รูปที่ 3-3 แสดงการวิเคราะห์ฟิลด์ต่าง ๆ ของโปรโตคอล ยูดีพี (UDP)

3. โปรโตคอล ไอซีเอ็มพี (ICMP)



รูปที่ 3-4 แสดงการวิเคราะห์ฟิลด์ต่าง ๆ ของโปรโตคอล ไอซีเอ็มพี (ICMP)

3.1.2 พร็อกซี่ (Proxy)

พร็อกซี่ (Proxy) หรือ แอพพลิเคชั่นเกตเวย์ (Application Gateway) เป็นแอพพลิเคชันโปรแกรมที่ทำงานอยู่บนไฟร์วอล ที่ตั้งอยู่ระหว่างเน็ตเวิร์ค 2 เน็ตเวิร์ค ทำหน้าที่เพิ่มความปลอดภัยของระบบเน็ตเวิร์คโดยการควบคุมการเชื่อต่อระหว่างเน็ตเวิร์คภายใน กับเน็ตเวิร์คภายนอก พร็อกซี่ จะช่วย

เพิ่มความปลอดภัยได้มากเนื่องจากมีการตรวจสอบข้อมูลถึงในระดับของแอพพลิเคชันเลเยอร์ (Application Layer) เมื่อไคลเอนต์ (Client) ต้องการใช้เซอร์วิสภายนอก ไคลเอนต์จะทำการติดต่อไปยังพร็อกซี่ก่อน ไคลเอนต์จะเจรจา (negotiate) กับ พร็อกซี่ ติดต่อไปยังเครื่องปลายทางให้แล้วจะมีการเชื่อมต่อ (connection) 2 การเชื่อมต่อ คือ ไคลเอนต์ กับ พร็อกซี่ และ พร็อกซี่ กับเครื่องปลายทาง โดยที่พร็อกซี่ จะทำหน้าที่รับข้อมูล และ ส่งข้อมูลให้ใน 2 ทิศทาง ทั้งนี้ พร็อกซี่ จะทำหน้าที่ในการตัดสินใจว่าจะให้มีการเชื่อมต่อกันหรือไม่ จะส่งต่อแพ็กเก็ตให้หรือไม่

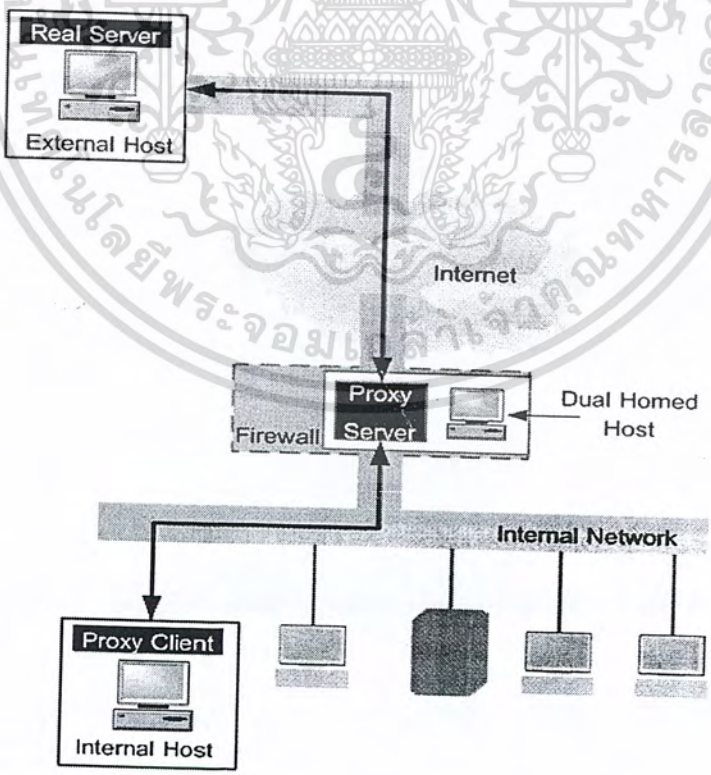
ข้อดี-ข้อเสียของพร็อกซี่

ข้อดี

- มีความปลอดภัยสูง
- รู้จักข้อมูลในระดับแอพพลิเคชัน

ข้อเสีย

- ประสิทธิภาพต่ำ
- แต่ละบริการมักจะต้องการโปรเซสของตนเอง
- สามารถขยายตัวได้ยาก



รูปที่ 3-5 แสดงการใช้ Dual-homed Host เป็น พร็อกซี่เซอร์ฟเวอร์ (Proxy Server)

3.1.3 สเตทฟูอินสเป็คชั่น (Stateful Inspection)

โดยปกติแล้ว แพ็กเก็ตฟิลเตอริง (Packet Filtering) แบบธรรมดา จะควบคุมการเข้าออกของแพ็กเก็ตโดยพิจารณาข้อมูลจากเฮดเดอร์ของแต่ละแพ็กเก็ต นำมาเปรียบเทียบกับกฎที่มีอยู่ ซึ่งกฎที่มีอยู่ก็จะเป็นกฎที่สร้างจากข้อมูลส่วนที่อยู่ในเฮดเดอร์เท่านั้น ดังนั้นแพ็กเก็ตฟิลเตอริง แบบธรรมดาจึงไม่สามารถทราบได้ว่า แพ็กเก็ตนี้อยู่ส่วนไดของการเชื่อมต่อ เป็นแพ็กเก็ตที่เข้ามาติดต่อใหม่หรือเปล่า หรือว่าเป็นแพ็กเก็ตที่เป็นส่วนของการเชื่อมต่อที่เกิดขึ้นแล้ว เป็นต้น

สเตตฟูลอินสเปกชัน (Stateful Inspection) เป็นเทคโนโลยีที่เพิ่มเข้าไปใน แพ็กเก็ตฟิลเตอริง โดยในการพิจารณาว่าจะยอมให้แพ็กเก็ตผ่านไปนั้น แทนที่จะดูข้อมูลจากเฮ็ดเดอร์เพียงอย่างเดียว สเตตฟูลอินสเปกชัน ยังจะนำเอาส่วนข้อมูลของแพ็กเก็ต (message content) และข้อมูลที่ได้จากแพ็กเก็ตก่อนหน้านี้ที่ได้ทำการบันทึกเอาไว้ นำมาพิจารณาด้วย จึงทำให้สามารถระบุได้ว่าแพ็กเก็ตใดเป็นแพ็กเก็ตที่ติดต่อเข้ามาใหม่ หรือว่าเป็นส่วนหนึ่งของการเชื่อมต่อที่มีอยู่แล้ว

	Packet Filtering	Stateful Inspection	Proxy
ข้อดี	• ประสิทธิภาพดี • ง่ายในการ implement • ไม่ขึ้นกับแอพพลิเคชั่น	• ประสิทธิภาพดี • เปิดพอร์ตเฉพาะเมื่อมีการติดต่อ • สนับสนุนเกือบทุกบริการ	• ไม่เปิดเผยหมายเลขไอพีภายใน • พิจารณาเนื้อหาของข้อมูลด้วย • มี User Authentication • เก็บรายละเอียด log
ข้อเสีย	• เปิดเผยหมายเลขไอพีภายใน • มีการเปิดช่องว่างทิ้งไว้ถาวร • No User Authentication • ใช้การเชื่อมต่อโดยตรงกับภายนอก	• No User Authentication • ใช้การเชื่อมต่อโดยตรงกับภายนอก • เปิดเผยหมาเลขไอพีภายใน	• ประสิทธิภาพต่ำกว่า • ต้องมีพร็อกซี่สำหรับทุก ๆ แอพพลิเคชั่นที่ใช้ • ไม่มีการป้องกันที่ต่ำกว่าชั้นแอพพลิเคชั่น • เปิดเผยระบบปฏิบัติการ

ตารางที่ 3-2 เปรียบเทียบการทำงานของไฟร์วอลทั้ง 3 ประเภท

3.2 ภัยจากการโจมตี และ ความสามารถของไฟร์วอล

3.2.1 SYN Flooding

เครื่องผู้บุกรุกจะส่ง SYN Flag มายังเครื่องเป้าหมาย เพื่อทำ 3way-handshaking เมื่อเครื่องเป้าหมายได้รับ SYN Flag จะส่ง ACK และ SYN กลับไปยังเครื่องผู้บุกรุก ในขณะเดียวกันเครื่องเป้าหมายจะจองพื้นที่ในหน่วยความจำเพื่อรอรับ ACK ซึ่งหากผู้บุกรุกปลอมหมายเลขไอพี และ ส่ง SYN

มาเป็นจำนวนมาก หน่วยความจำของเครื่องเป้าหมายก็จะถูกใช้จนหมดไป จนเครื่องเป้าหมายไม่สามารถให้บริการได้หรือ ใช้งานไม่ได้ชั่วขณะ

SYN flooding เป็นการโจมตีในเชิงปริมาณ และความเร็ว ซึ่งกฎของไฟร์วอลไม่ครอบคลุมถึงปริมาณ และ ความเร็วของข้อมูล การมีไฟล์วอลเป็นด่านหน้าจะช่วยบรรเทาความรุนแรงเนื่องจากการ SYN flood ได้ เพราะไฟร์วอลจะเป็นผู้รักษาการติดต่อไว้ทั้งหมดตั้งแต่เริ่ม SYN จนสิ้นสุดเท่านั้น

3.2.2 PING Attack

ไอซีเอ็มพีมีหน้าที่ส่งข่าวสารและคำสั่งควบคุมของไอพี โดยเฉพาะการรายงานข้อผิดพลาดในการรับส่ง PING เป็นข้อความไอซีเอ็มพีประเภทหนึ่ง ใช้ในการหาข้อมูลเกี่ยวกับเครือข่าย

ซึ่งในการป้องกันโดยใช้ไฟร์วอลโดยไม่ควรให้เครื่องภายนอก หาข้อมูลเครื่องภายในเครือข่ายได้ โดยกำหนดกฎให้ เหมาะสม เช่นอาจจะไม่ให้มีการตอบสนอง ICMP packet เป็นต้น หรืออาจจะใช้โปรแกรมอื่นช่วยในการตรวจสอบการทำงานที่ผิดปกติ เช่น ระบบตรวจจับผู้บุกรุก (Intrusion Detection)

3.2.3 Tiny Fragment

หากแพ็กเก็ตมีขนาดใหญ่เกินกว่าที่ชั้นดาต้าลิงค์อนุญาต โปรโตคอลไอพี มีความสามารถในการแบ่งแพ็กเก็ตออกเป็นแพ็กเก็ตย่อย ๆ แล้วประกอบกลับให้เหมือนเดิมได้เมื่อแพ็กเก็ตถึงปลายทาง เนื่องจากโปรโตคอล ทีซีพี จะมีเฮดเดอร์แพ็กเก็ตของตัวเอง และถูกเติมเข้ากับเฮดเดอร์ของโปรโตคอลไอพีก่อนส่งไปยังปลายทาง หากแพ็กเก็ตมีขนาดใหญ่ แพ็กเก็ตก็จะถูกแบ่งออกเป็นแพ็กเก็ตย่อย ซึ่งจะมีเพียงแพ็กเก็ตแรกเท่านั้นที่บอกว่าแพ็กเก็ตนี้ถูกส่งมาจากหมายเลขไอพีใด หากเป็นหมายเลขไอพีที่ไม่ได้รับอนุญาตให้ผ่าน ไฟร์วอลก็จะมีเพียงแพ็กเก็ตแรกแพ็กเก็ตเดียวเท่านั้นที่ถูกดร็อป หรืออาจจะไม่มีการดร็อปแพ็กเก็ตใด ๆ เลย หาก Sequence Number มีการแก้ไขเป็น 1 (ซึ่งปกติเป็น 0) เหมือนว่าไม่ใช่แพ็กเก็ตแรก

3.2.4 Port Scanning

การสแกนพอร์ตทุกครั้ง ผลลัพธ์ที่ได้คือรายละเอียดที่จะบอกได้ว่า โฮสต์ที่ถูกสแกนใช้ระบบปฏิบัติการอะไร มีแอพพลิเคชั่นใดทำงานอยู่บ้าง เปรียเสมือนบอกผู้บุกรุกว่ามีช่องทางใดที่จะเจาะระบบเข้ามาได้

การแสกนพอร์ตจะทำให้ผู้บุกรุกทราบได้ว่ามีแอพพลิเคชั่นใดทำงานอยู่ หลังจากนั้นจะทำการสแกนแบบเจาะลึก ก็จะทราบว่าแอพพลิเคชั่นนั้นเป็นโปรแกรมอะไร เวอร์ชันใด มีข้อบกพร่องอย่างไร และควรนำเครื่องมือ หรือ เทคนิคใดในการเจาะ

ซึ่งการป้องกันโดยใช้ไฟร์วอลอาจจะต้องใช้โปรแกรมอื่นใช้งานร่วมเพื่อให้เกิดประสิทธิภาพการทำงานได้อย่าง ถูกต้องมาก

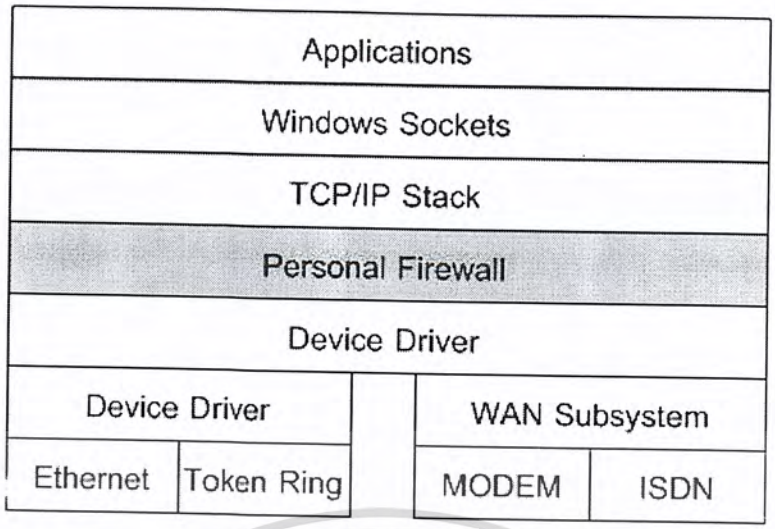
3.3 เพอร์ซันนอลไฟร์วอล (Personal Firewall)

เพอซันนอลไฟร์วอลจะมีความแตกต่างกับไฟร์วอลทั่วไป เพราะไฟร์วอลทั่วไปนั้นมีเป้าหมายหลักอยู่ที่ความปลอดภัยของเน็ตเวิร์คและ การควบคุมทราฟฟิคที่ผ่านเข้าออกระหว่าง เน็ตเวอร์คภายในกับเน็ตเวอร์คภายนอก ทำงานอยู่ระหว่างโปรโตคอลในชั้น ทรานสพอร์ตเลเยอร์ (Transport Layer) กับแอพพลิเคชั่นเลเยอร์ (Application Layer) แต่สำหรับเพอร์ซันนอลไฟร์วอลแล้ว การทำหน้าที่เพียงควบคุมทราฟฟิคเข้าออกนั้นไม่เพียงพอที่จะคุ้มครองป้องกันผู้ใช้ให้ได้อย่างปลอดภัย เพราะภัยของเครื่องคอมพิวเตอร์ส่วนบุคคลนั้นได้ครอบคลุมไปในหลายเรื่อง และส่วนใหญ่ก็มีความจำเป็นก็มีความจำเป็นที่จะต้องไปยุ่งเกี่ยวกับแอพพลิเคชั่นด้วย เดิมนั้นเพอร์ซันนอลไฟร์วอลมีรากฐานมาจากไฟร์วอลปกติเพียงแต่เป็นการนำมารวมเข้าไว้ในเครื่องของผู้ใช้ เสียเลยแทนที่ จะต้องแยกเป็นไฟร์วอลต่างหากอีกเครื่องหนึ่งซึ่งจะเป็นการสิ้นเปลืองโดยใช่เหตุ แต่ในเมื่อมภัยหลายประการที่จะส่งผลกระทบต่อผู้ใช้ ผู ผลิตจึงได้นำโปรแกรมป้องกันหลาย ๆ ชนิดมารวมกันเป็นชุดโปรแกรมรักษาความปลอดภัยสำหรับเครื่องคอมพิวเตอร์ส่วนบุคคลโดยประกอบ ด้วย

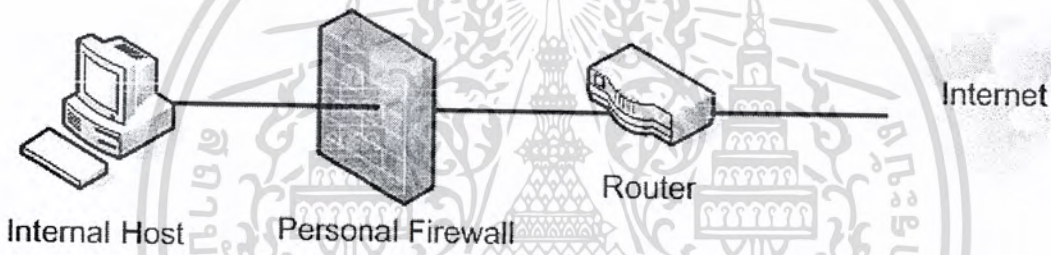
- ไฟร์วอล (Firewall) สำหรับควบคุมทราฟฟิค เป็นการป้องกันการเจาะระบบของแฮ็กเกอร์
- โปรแกรมป้องกันการทำงานของแอคทีฟคอนเทนต์
- โปรแกรมป้องกันเว็บไซต์ที่ไม่เหมาะสม (Parental Control)
- โปรแกรมบล็อกโฆษณาทางอินเทอร์เน็ต (Advertising Blocking)
- โปรแกรมป้องกันไวรัส, ตรวจสอบหาโปรแกรมม้าโทรจัน และ โปรแกรมประเภทแบ็คดอร์
- โปรแกรมตรวจจับ และ ป้องกันการบุกรุก (Intrusion Detection & Protection)

ซึ่งขึ้นอยู่กับบริษัทผู้ผลิตว่าจะรวบรวมโปรแกรมเหล่านี้เข้าด้วยกันมากน้อยแค่ไหน อย่างไร ดังนั้นเพอร์ซันนอลไฟร์วอลที่มีจำหน่ายอยู่ในปัจจุบันจึงกลายเป็นชุดรักษาความปลอดภัยแบบครบวงจร

ในหน้าที่ส่วนที่เป็นไฟร์วอลนั้นนอกจากจะมีขอบเขตหน้าที่การทำงานที่ต่างจาก ไฟร์วอล ทั่วไปแล้ว โครงสร้างในทางเทคนิคของเพอร์ซันนอลไฟร์วอลก็มีความแตกต่างจากไฟร์วอลธรรมดาอยู่มากพอสมควร กล่าวคือ ไฟร์วอลทั่วไปมักจะมีโครงสร้างที่เป็นองค์ประกอบของฮาร์ดแวร์ และ ซอฟต์แวร์ทำงานร่วมกัน มีเน็ตเวอร์คอะแดปเตอร์ คอยรับข้อมูลที่เข้ามาและส่งข้อมูลที่ผ่านการตรวจสอบแล้วออกไป แต่สำหรับเพอซันนอลไฟร์วอลนั้นเป็นเพียงซอร์ฟแวร์เท่านั้น ไม่มีฮาร์ดแวร์ใด ๆ มาเกี่ยวข้องแม้แต่น้อย เป็นซอฟแวร์ที่ทำหน้าที่ตรวจตราและควบคุมการเรียกใช้เน็ตเวิร์คดีไวซ์บนเครื่องคอมพิวเตอร์นั้นเอง



รูปที่ 3-6 เพอร์ซันนอลไฟร์วอลเมื่อร่วมเข้ากับซอฟต์แวร์บนโฮสต์



รูปที่ 3-7 แสดงการนำไฟร์วอลมาให้บริการในลักษณะของเพอร์ซันนอลไฟร์วอล

3.4 Firewall API

ในการเขียนโปรแกรมบนวินโดวส์ เพื่อติดต่อและควบคุมลงไปถึง NIC (Network Interface Card) แล้วจะมีความยุ่งยากมาก เพราะ NIC แต่ละอันก็มีความแตกต่างกันไป โดยทั่ว ๆ ไปจะเป็นการนำ API (Application Program Interface) ที่เตรียมไว้ใน Microsoft SDK (Software Development Kit) มาใช้ในการพัฒนาโปรแกรม จะทำให้การพัฒนาโปรแกรมบนวินโดวส์ทำได้ง่ายขึ้น

3.4.1 Firewall-Hook Driver

การพัฒนาไฟร์วอลบนระบบปฏิบัติการวินโดวส์ 2000 จะใช้วิธีฮูค (Hook) ไดร์เวอร์ของไฟร์วอลเข้ากับระบบของระบบปฏิบัติการ โดยวิธีนี้ไมโครซอฟท์ ไม่ได้ให้ข้อมูลเกี่ยวกับเรื่องนี้มากนัก ซึ่งมีทางเป็นไปได้ที่จะศึกษาคือ ต้องศึกษา DDK header file (Microsoft Driver Development Kit) ซึ่งก็คือไฟล์ ipFirewall.h ซึ่งเป็นวิธีการที่น่าสนใจมากเพราะเป็นการพัฒนาโปรแกรมประเภทแพ็คเก็ตฟิลเตอริง (Packet Filtering) ได้อย่างมีประสิทธิภาพ

3.4.2 การฟิลเตอร์แพ็กเกตบนวินโดวส์

สำหรับสถาปัตยกรรมของ NDIS นั้นได้อนุญาตให้สามารถที่จะเขียนโปรแกรมเพื่อเข้าไปฟิลเตอร์แพ็กเกตได้หลายวิธีโดยการเรียกใช้ฟังก์ชันการทำงานจากชั้นต่าง ๆ ของ NDIS ซึ่งแต่ละวิธีก็มีวิธีการ รูปแบบการใช้งาน และความสามารถในการทำงานที่แตกต่างกันไป มีทั้งในแบบ User-Mode และ Kernel-Mode

- User-Mode Network Data Filtering
  - Winsock Layered Service Provider (LSP) เป็นไปได้ที่เราสามารถที่จะเรียก Kernel-Mode TCP/IP Driver ผ่าน Transport Data Interface (TDI) โดยไม่จำเป็นต้องผ่าน Winsock แต่ในงานที่จำเป็นจะต้อง ตรวจสอบหรือกระทำการกับทุก ๆ แพ็กเกตนั้นไม่ควรที่จะอิงการทำงานของ Winsock LSP แต่ควรจะประยุกต์ใช้วิธีการ Kernel-mode มากกว่า
  - Windows 2000 Packet Filtering Interface สำหรับระบบปฏิบัติการวินโดวส์ 2000 ได้มี API (Application Program Interface) ที่อนุญาตให้แอปพลิเคชัน หรือ เซอร์วิสที่ทำงานใน User-mode สามารถที่จะกำหนดชุดของ Filter Descriptor ซึ่งจะถูกนำไปใช้โดยคอมโพเนนท์ (Component) ทีซีพี/ไอพี (TCP/IP) ในชั้นล่าง การฟิลเตอร์นี้จะดูที่ หมายเลขไอพีต้นทาง หมายเลขไอพีปลายทาง และ หมายเลขพอร์ต โดยสามารถกำหนดได้เพียงแค่อนุญาตให้ผ่าน หรือ ไม่ให้ผ่าน เท่านั้น
  - Winsock Replacement DLL ก่อนที่จะมี LSP วิธีการเดียวที่จะเพิ่มความสามารถ ของฟังก์ชันในการทำงานของ Winsock คือการสร้าง DLL ชุดใหม่มาทับ DLL ของ Winsock ซึ่งก็สามารถประยุกต์ใช้ในการฟิลเตอร์ แพ็กเกตได้ แต่ไม่เป็นที่นิยมเนื่องจากความยากในการทำ อีกทั้งยังต้องยุ่งเกี่ยวกับฟังก์ชันการทำงานที่ไม่มีคู่มืออธิบายอีกด้วย
- Kernel-Mode Network Data Filtering
  - Transport Data Interface (TDI) Filter Driver ทำงานในชั้นของ TCP/IP Driver
  - NDIS Intermediate Driver ทำงานอยู่ระหว่าง Miniport Driver กับ Transport Driver ซึ่งสามารถที่จะประยุกต์ฟิลเตอร์แพ็คเกตในชั้นนี้ได้เช่นกัน
  - Windows 2000 Filter-Hook Driver เป็นตัวที่ ขยายขอบข่ายความสามารถออกมาจาก IP Filter Driver ซึ่งมากับระบบปฏิบัติการ
  - NDIS Hooking Filter Driver

3.4.3 การทำงานของไฟร์วอลแบบ Hook Driver ที่สร้างขึ้น

ไดร์เวอร์ของไฟร์วอลแบบ ฮูค นี้ไม่ใช่ไดรเวอร์ของเน็ทเวิร์ค ดังนั้นมันจะทำงานในโหมดของเคอร์เนล (Kernel Mode) โดยการติดตั้งฟังก์ชันคอลแบ็ค (Callback Function) โดยเริ่มที่ติดตั้งไดรเวอร์ไฟร์วอลแบบ ฮูค นี้โดยไฟล์ ipFirewall.h จะมีดังนี้

```
typedef struct _IP_SET_FIREWALL_HOOK_INFO
{
        // Packet filter callout.
        IPPacketFirewallPtr FirewallPtr;
        // Priority of the hook
        UINT Priority;
        // if TRUE then ADD else DELETE
        BOOLEAN Add;
} IP_SET_FIREWALL_HOOK_INFO, *PIP_SET_FIREWALL_HOOK_INFO;

#define DD_IP_DEVICE_NAME L \\Device\\Ip
#define _IP_CTL_CODE (function, method, access) \
    CTL_CODE (FSCTL_IP_BASE, function, method, access)
#define IOCTL_IP_SET_FIREWALL_HOOK \
    _IP_CTL_CODE (12, METHOD_BUFFERED, FILE_WRITE_ACCESS)
```

ซึ่งเราสามารถกำหนดค่าต่างๆ ให้กับโครงสร้าง โดยในฟิลด์ Priority เป็นส่วนที่กำหนดลำดับของฟังก์ชันฟิลเตอร์ และเราสามารถนำไปใช้ดังนี้

```
PDEVICE_OBJECT ipDeviceObject=NULL;
IP_SET_FIREWALL_HOOK_INFO filterData;

//.....

// Init structure filterData.
FirewallPtr = filterFunction;
filterData.Priority = 1;
filterData.Add = TRUE;

//.....

// Send the commando to ip driver
IoCallDriver(ipDeviceObject, irp);
```

ถ้าต้องการเลิกติดตั้งฟังก์ชันฟิลเตอร์ สามารถทำได้โดยใช้โค้ดเดิม แต่ใส่ค่าที่ filterData.Add เป็น FALSE

ฟังก์ชันฟิลเตอร์ (The Filter Function)

ฟังก์ชันฟิลเตอร์สำหรับไฟร์วอลแบบสูดนี้ จะมีความซับซ้อนมาก เพราะไม่มีข้อมูลเกี่ยวกับฟังก์ชันและพารามิเตอร์ โดยฟังก์ชันฟิลเตอร์มีดังนี้

```
FORWARD_ACTION cbFilterFunction(VOID**pData,
    UINT RecvInterfaceIndex,
    UINT*pSendInterfaceIndex,
    UCHAR* pDestinationType,
    VOID*pContext,
    UINT ContextLength,
    Struct IPRcvBuF**pRcvBut):
```

พารามิเตอร์ต่างๆ มีดังนี้

pData * pData เป็นพ้อยเตอร์ (pointer) ที่ชี้ไปยังโครงสร้างที่เก็บข้อมูลแพ็กเกต (struct IPRcvBuf*)

RecvInterfaceIndex เป็นส่วนที่รับข้อมูลเข้ามา

pSendInterfaceIndex เป็นพ้อยเตอร์ (pointer) ชนิด unsigned int เก็บค่าที่ชี้ข้อมูลที่ส่ง

pDestinationType เป็นพ้อยเตอร์ (pointer) ชนิด unsigned int เก็บค่าของเป้าหมาย เช่น network, remote, broadcast, multicast, etc.

pContext เป็นพ้อยเตอร์แบบโครงสร้าง FIREWALL_CONTEXT_T เพื่อหาข้อมูลของแพ็กเกตที่เข้าหรือออก

pRcvBuf*pRcvBuf ถูกเซตให้เป็น NULL เสมอ

ข้อมูลทั้งหมดนี้อาจมีการเปลี่ยนแปลงได้ในวินโดวส์เวอร์ชันต่อไปในอนาคต เพราะไม่มีข้อมูลอย่างเป็นทางการ แต่ข้อมูลเหล่านี้ได้ถูกทดสอบในวินโดวส์ 2000 และวินโดวส์ XP

แต่ละแพ็กเกต เมื่อฟังก์ชันถูกเรียกใช้ การทำงานจะขึ้นอยู่กับค่าที่ส่งกลับมา ว่าจะให้แพ็กเกตผ่านหรือว่าไม่ให้ผ่าน ซึ่งค่าที่ส่งกลับมีดังนี้

- FORWARD แพ็กเกตจะถูกอนุญาตให้ผ่าน
- DROP แพ็กเกตจะถูกปฏิเสธไม่ให้ผ่าน
- ICMP_ON_DROP แพ็กเกตจะถูกปฏิเสธไม่ให้ผ่านและแพ็กเกต ICMP จะถูกส่งไปที่เครื่องรีโมทโดยในส่วนนี้ซึ่งสามารถกำหนดได้ทั้งการฟิลเตอร์ทั้งขาเข้าและขาออก พารามิเตอร์หลักสำหรับฟังก์ชันนี้ คือ PF_FILTER_DESCRIPTOR เพื่อเป็นข้อกำหนดกฎต่างๆ ในฟังก์ชันการฟิลเตอร์ ที่จะส่งเข้าไปยังไดร์เวอร์ไฟร์วอลแบบสูดให้ทำการฟิลเตอร์นั่นเอง

ฟังก์ชัน PF_FILTER_DESCRIPTOR

Filter Descriptor นั้นเป็นรูปแบบโครงสร้างของกฎการฟิลเตอร์ ในการที่เราจะกำหนดกฎแต่ละข้อซึ่งในขั้นตอนแรกนั้น เราจะต้องสร้าง Filter Descriptor ขึ้นมาและกำหนดค่าต่างๆ โดยโครงสร้างของ PF_FILTER_DESCRIPTOR นั้นเป็นดังต่อไปนี้

```
Typedef struct PF_FILTER_DESCRIPTOR {
        DWORD dwFilterFlage;
        DWORD dwRule;
        PFADDRESSTYPE pfatType;
        PBYTE SrcAddr;
        PBYTE SrcMadk;
        PBYTE DstSddr;
        PBYTE DdtMask;
        DWORD dwProtocol;
        DWORD fLateBound;
        WORD wSrcPort;
        WORD wDstpor;
        WORD wSrcportHighRange;
        WORD wDstPortHighRange;
} PF_FILTER_DESCRIPTOR, *PPF_FILTER_DESCRIPTOR;
```

โดยมีรายละเอียดของแต่ละฟิลด์ดังนี้

dwFilterFlags ณ ขณะนี้มีเพียงแค่ค่าเดียวที่ถูกกำหนดขึ้นสำหรับฟิลด์นี้ คือ FD_FLAGS_NOSYN โดยเมื่อแอปพลิเคชันต้องการที่จะสร้างการเชื่อมต่อแบบ 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 ถ้าต้องการกำหนดหมายเลขไอพีหมายเลขเดียวแอดเดรสจะต้องสมบูรณ์และค่ามาสก์จะต้องเป็น FF.FF.FF.FF. ในทางกลับกัน ถ้าต้องการจะกำหนดฟิลเตอร์สำหรับช่วงของหมายเลขไอพี ก็ต้องกำหนดส่วนที่เป็นช่วงของแอดเดรส ให้เป็นศูนย์เช่นเดียวกันกับส่วนเดียวกันสำหรับมาสก์ (แอดเดรส a.b.c.0 และมาสก์ FF.FF.FF.00 จะฟิลเตอร์หมายเลขไอพีทั้งหมดที่ขึ้นต้นด้วย a.b.c) ในการกำหนดฟิลเตอร์ขาออก จะต้องกำหนดให้แอดเดรสของตัวมันเองเป็นแอดเดรสต้นทาง ส่วนการฟิลเตอร์ขาเข้าจะต้องกำหนดแอดเดรสของฝั่งรีโมท (Remote)เป็นต้นทาง dwProtocol กำหนดโพรโตคอลไอพีที่จะต้องฟิลเตอร์ ซึ่งมี ICMP,TCP,UDP หรือทั้งหมด fLateBound ใช้ในการกำหนดข้อมูลเกี่ยวกับแอดเดรสที่ต้องการถูกปรับปรุงเมื่อมีการ rebound ฟิลเตอร์ (เช่นเมื่อมีการเชื่อมต่อใหม่ของ dial-up adapter) wSrcPort หมายเลขพอร์ตต้นทาง wSrcPortHighRange ขอบบนของหมายเลขพอร์ตต้นทาง wSrcport หมายเลขพอร์ตปลายทาง wDstportHighRange ขอบบนของหมายเลขพอร์ตปลายทางการกำหนดพอร์ตในการฟิลเตอร์นั้น ถ้ากำหนดหมายเลขพอร์ตเป็นศูนย์จะหมายถึงทุกพอร์ต แต่ถ้ากำหนดหมายเลขพอร์ตที่ไม่ใช่ศูนย์แล้วจะต้องกำหนด PortHighRange ด้วยโดยถ้าต้องการระบุพอร์ตใดพอร์ตหนึ่งก็ให้กำหนดทั้งสองฟิลด์ให้เป็นค่าเดียวกัน แต่ถ้าต้องการกำหนดเป็นช่วงพอร์ตก็ให้กำหนด PortHighRange ให้เป็นค่าช่วงของหมายเลขพอร์ต กล่าวคือ กำหนดค่าพอร์ตให้เป็นหมายเลขพอร์ตต่ำสุดที่ต้องการฟิลเตอร์ และ PortHighRange เป็นค่าสูงสุดของพอร์ตที่ต้องการฟิลเตอร์ เช่น ถ้าต้องการให้มีการฟิลเตอร์ตั้งแต่ พอร์ต 21 ถึงพอร์ต 48 ก็ให้กำหนด Port เป็น 21 และ PortHighRange เป็น 48 เป็นต้น

บัฟเฟอร์ของแพ็กเกตในไดร์เวอร์ของไฟร์วอลแบบฮูคนั้น จะไม่ได้รับส่วนหัวของแพ็กเกตและส่วนที่บรรจุในแพ็กเกตนั้นของบัฟเฟอร์โดยทันที แต่บัฟเฟอร์นั้นจะมีโครงสร้างการทำงาน ทั้งส่งและรับแพ็กเกตที่ผ่านในพารามิเตอร์ pData โดย *pData จะชี้ที่ IPRcvBuf ซึ่งมีโครงสร้างดังนี้

```
Struct IPRcvBuf
{
        //Point to the next buffer in the chain
        struct IPRcvBuf*ipr_next;
        //Always 0
        UINT ipr_owner;
        //Buffer data
        UCHAR *ipr_buffer;
```

```
//Buffer data size
UINT ipr_size;

//In my tests always a pointer to NULL.
//Maybe the system could use MDLs instead of IPRcvBuf structures
//(but i never have seen it).
PMDL_ipr_pMdI;

//Always a pointer to NULL.
UINT*ipr_pClientCnt;

//Always a pointer to NULL.
UCHAR*ipr_RcvContect;

//Always 0. I suppose this field is an offset into buffer data
//but because I haven"t a value different from 0, I can affirm it.
UINT ipr_RcvOffset;

//In windows 2003 DDk the name of this field have changed to flags.
//In my tests I always get 0 value for local traffic and  0 for
//remote. It"s the only thing I can tell  you about this field.
ULONG ipr_ promiscuous;
};
```

จากโครงสร้างนี้ ฟิลด์ ipr_buffer จะเก็บ ipr_size ซึ่งเป็นขนาดของแพ็กเกตมีหน่วยเป็นไบต์ ส่วนฟิลด์ ipr_next จะใช้เก็บโครงสร้างต่อไป เพราะการเก็บข้อมูลของแพ็กเกตนี้จะเก็บเป็นโครงสร้างของข้อมูลต่อๆ กันไปจนโครงสร้างสุดท้ายจะชี้ไปที่ NULL โดยในแพ็กเกต ซึ่งมีลักษณะการใช้งานดังนี้

```
Char *pPacket=NULL:
Int iBufferSize;
Struct IPRcvBuf*pBuffer=(struct IPRcvBuf*)*pData;

//First, I calculate the total size of the packet
```

```
iBuffersize=buffer->ipr_size;
while(pBuffer->ipr_next!=NULL)
{
        pBuffer=pBuffer->ipr_next;
        iBufferSize+=pBuffer->ipr_size;
}
//Reserve memory to the lineal buffer.
pPacket=(char*)ExAllocatePool(NonPagedPool, iBufferSize);
if(pPacket!=NULL)
{
        unsigned int iOffset = 0;
        pBuffer = (struct IPRcvBuf*)*pData;

        //we are going to copy each buffer of the chain in the lineal buffer.
        Memcpy(pPacket, pBuffer->ipr_buffer,pBuffer->ipr_size);
        While(pBuffer ->ipr_next!= NULL)
        {
            iOffset = pBuffer ->ipr_size;
            pBuffer = pBbuffer->ipr->next;
            memcpy(pPacket + iOffset,pBuffer->ipr_buffer,
            pBbuffer ->ipr_size);
        }
}
```

บทที่ 4
ระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

4.1 ความหมายของการตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

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

โดยระบบนี้ต้องเก็บข้อมูลของแพ็กเก็ตต่างๆ ที่เข้ามาสู่ระบบ แล้วนำมาวิเคราะห์เปรียบเทียบกับกฎต่างๆ ที่ตั้งไว้ รวมถึงนโยบายขององค์กรก็นำมาพิจารณาด้วย เพื่อตรวจสอบว่ามีสิ่งผิดปกติเกิดขึ้นกับระบบหรือไม่ หากเกิดสิ่งผิดปกติก็จะแจ้งเตือนไปยังผู้ดูแลระบบหรือเก็บไว้ใน ล็อกไฟล์ต่อไป

การตรวจจับผู้บุกรุกทางคอมพิวเตอร์สามารถแบ่งตามลักษณะของการโจมตีได้ 5ประเภท

1. การพยายามเจาะเข้าไปทำลายเครือข่าย (Attempted break-ins)
2. การปลอมแปลงเพื่อเข้ามาโจมตีเครือข่าย (Masqucrade attacks)
3. การอาศัยจุดบกพร่องของระบบรักษาความปลอดภัยเพื่อเจาะเข้าสู่เครือข่าย (Pcnctration of the sccurity control system)
4. การโจมตีเพื่อปิดบริการ (Denial of service)
5. การสำรวจระบบ (System survey)

4.2 ขอบเขตของระบบต้นแบบการตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ที่สร้างขึ้น

ระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ที่สร้างขึ้น มุ่งเน้นการศึกษา ออกแบบ และพัฒนาระบบการตรวจจับการสำรวจระบบและการโจมตีเพื่อให้ปิดบริการ โดยเป็นหนึ่งในประเภทของการตรวจจับผู้บุกรุกตามที่ได้มาแล้ว ซึ่งมีแนวโน้มเพิ่มขึ้นทุกวัน

4.3 วิธีการตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

ระบบตรวจจับผู้บุกรุกที่สร้างขึ้นนี้มีวิธีการตรวจจับผู้บุกรุกที่สามารถแบ่งออกตามประเภทของการบุกรุกเป็น 2 กรณี ได้แก่

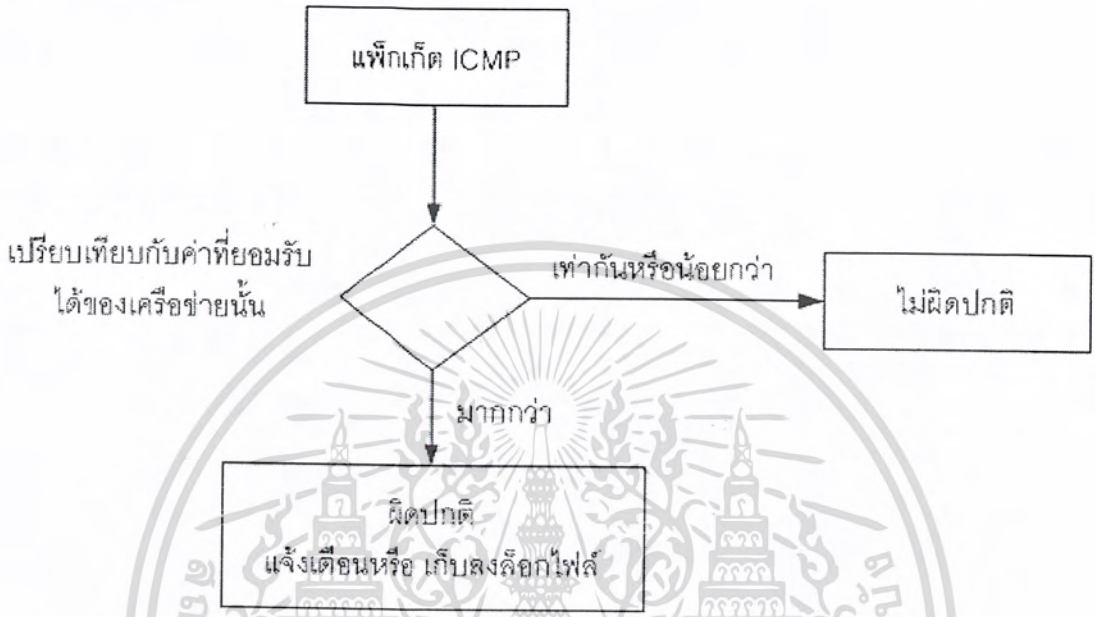
4.3.1 การบุกรุกเพื่อสำรวจระบบ

การบุกรุกเพื่อทำการสำรวจระบบเป็นการกระทำเพื่อเก็บข้อมูลของระบบ เพื่อใช้ในการโจมตีโดยข้อมูลที่ผู้โจมตีมักต้องการได้แก่ หมายเลขไอพีหรือชื่อเครื่อง. โครงสร้างทางเครือข่ายของระบบเป้าหมาย. ชื่อของบริการที่เปิด และระบบปฏิบัติการรวมทั้งเวอร์ชันที่ติดตั้งบนเครื่องเป้าหมาย

สำหรับการสำรวจระบบที่ระบบตรวจจับสามารถทำการตรวจจับได้มีอยู่ 3วิธีการสำรวจคือ

4.3.1.1 ปิงสวีป (Ping sweep)

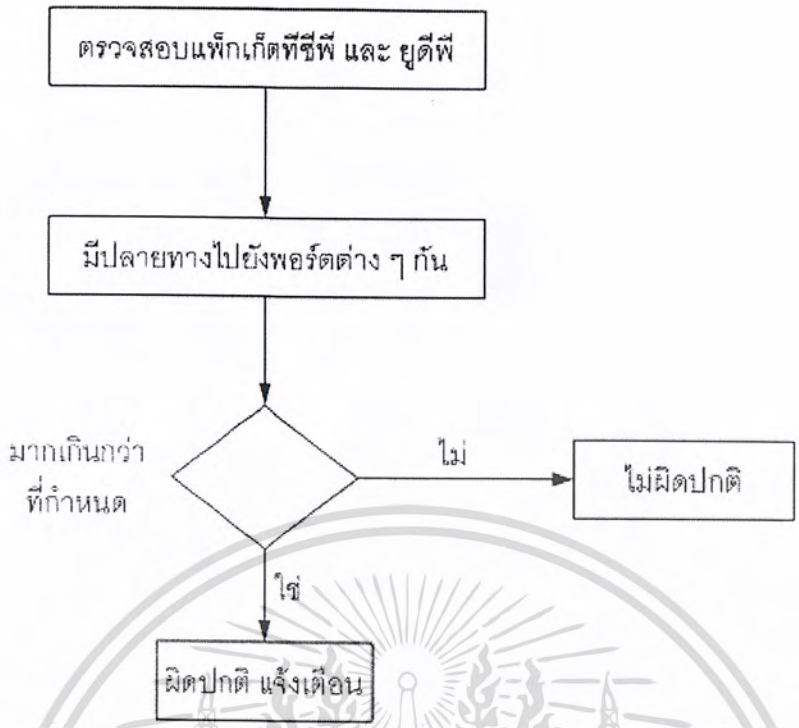
การตรวจจับการปิงสวีป (Ping sweep) สามารถทำได้โดยการตรวจสอบดูแพ็กเก็ต ICMP Request (Type8) ที่เข้ามายังระบบ หากมีแพ็กเก็ตลักษณะนี้จำนวนมากและมีปลายทางแตกต่างกันจะสามารถสรุปได้ว่าในเครือข่ายกำลังถูกสำรวจโดยการปิงสวีป



รูปที่ 4-1 แสดงการตรวจสอบการปิงสวีป

4.3.1.2 การแสกนพอร์ต

การตรวจสอบการสแกนพอร์ตทำได้โดยการตรวจดูแพ็กเก็ตที่มีหมายเลขพอร์ตปลายทางในลักษณะกระจาย คือแพ็กเก็ตมีการส่งไปยังเครื่องๆ เดียวแต่มีการส่งไปยังพอร์ตต่างๆ กันเป็นจำนวนมาก

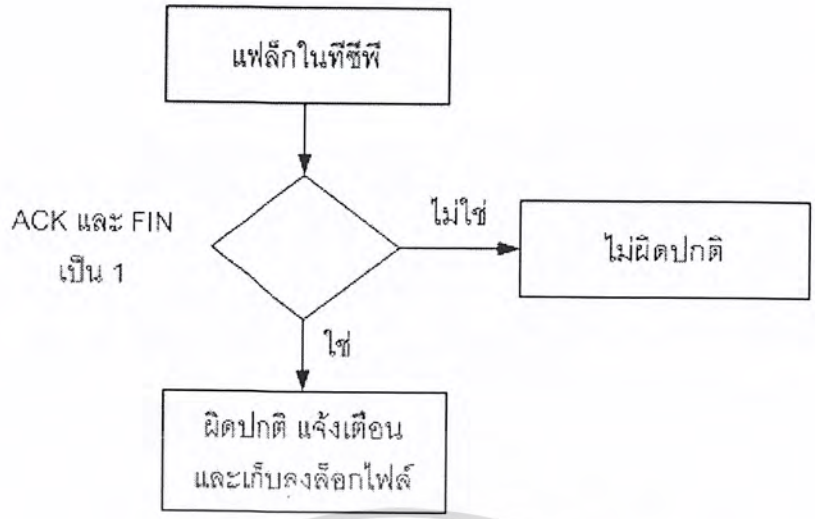


รูปที่ 4-2 แสดงการตรวจสอบการแสกนพอร์ต

4.3.1.3 การตรวจสอบระบบปฏิบัติการ

การตรวจจับการตรวจสอบปฏิบัติการสามารถตรวจสอบได้จากการพิจารณาแฟล็กที่ถูกส่งไปในชั้นทีซีพีของทุกๆ พอร์ตว่าเป็นแพ็กเก็ตแบบผิดปกติหรือไม่ กล่าวคือในแต่ละ สเตทของทีซีพีโพรโตคอลนั้นจะมีรูปแบบแฟล็กตายตัวอยู่ ตามสถานะปัจจุบันของสเตทของทีซีพี เช่น หากต้องการเริ่มต้นการเชื่อมต่อโปรโตคอลทีซีพี จะต้องกำหนดให้แฟล็ก ACK เป็น 1 ส่วนแฟล็กอื่นต้องเป็น 0 หรือหากต้องการยกเลิกการเชื่อมต่อโพรโตคอลทีซีพี จะต้อกำหนดให้แฟล็ก FIN เป็น 1 ส่วนแฟล็กอื่นเป็น 0 เป็นต้น

วิธีการตรวจจับที่ใช้ในโปรแกรมนั้นได้ทำการตรวจจับโดยหากแพ็กเก็ตมีแฟล็ก ACK และ FIN เป็น 1 พร้อมกันในแพ็กเก็ตตัวเดียวกัน จะระบุว่าเป็นการบุกรุกโดยการสำรวจเพื่อระบุระบบปฏิบัติการ เนื่องจากการตรวจจับวิธีนี้เป็นกรณีมาตรฐานที่โปรแกรมที่ทำการระบุระบบปฏิบัติการทุกโปรแกรมจะนำมาตรวจสอบ และเป็นแฟล็กที่ไม่สามารถเกิดได้จริงเมื่อมีการใช้งานโพรโตคอลทีซีพี



รูปที่ 4-3 แสดงการตรวจจับการตรวจสอบระบบปฏิบัติการ

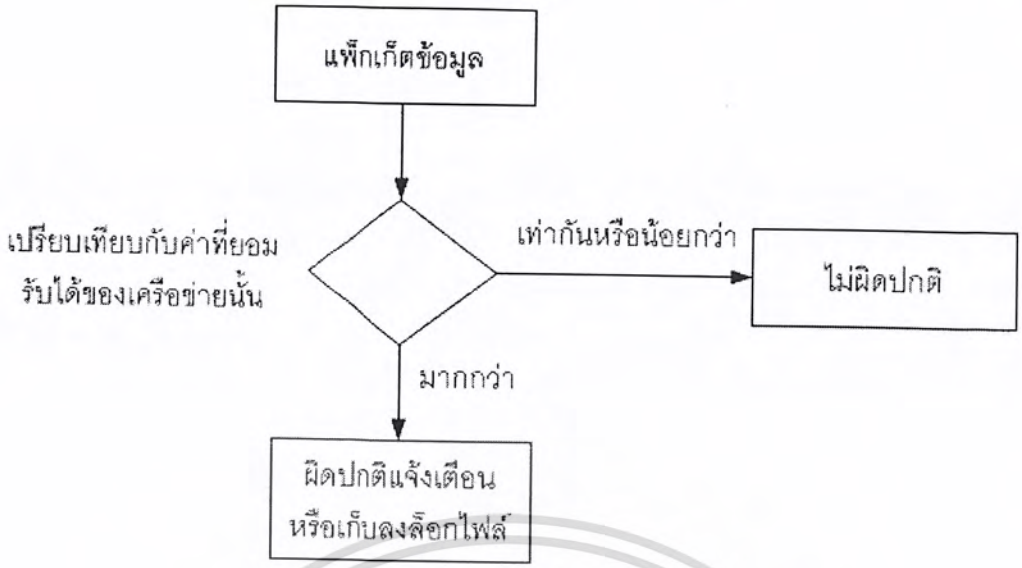
4.3.2 การโจมตีเพื่อให้ปิดบริการ

คือการกระทำใดๆ ที่ทำให้ระบบเป้าหมายไม่สามารถให้บริการบางอย่างได้ หรือไม่สามารถให้บริการต่อไปได้อีกการโจมตีระบบเครือข่ายคือการสร้างภาระให้กับเครือข่าย แบ่งได้ 3 ประเภทดังนี้

4.3.2.1 การส่งแพ็กเก็ตปริมาณมาก

การตรวจจับ แพ็กเก็ตที่เข้ามาในลักษณะนี้ทำได้โดยใช้การนับจำนวนแพ็กเก็ตที่เข้ามาสู่ระบบโดยพิจารณาจากแอดเดรสปลายทาง (Destination Address) ในแพ็กเก็ตเฮดเดอร์ของไอพี หากเป็นค่าเดียวกันให้นับจำนวนแพ็กเก็ตที่เข้ามาในช่วงเวลาหนึ่ง แล้วนำค่าที่ได้มาเปรียบเทียบกับค่าที่ยอมรับได้ หากค่าที่นับได้มากกว่าค่าที่ยอมรับได้ ก็ให้แจ้งเตือนแก่ผู้ดูแลระบบ หรือเก็บไว้ในล็อกไฟล์ ซึ่งการทำงานดังกล่าวนี้เป็นไปดังรูปที่ 4-3ความยากของการวิเคราะห์แบบนี้อยู่ที่การหาค่าที่ระบบยอมรับได้ เพราะขึ้นอยู่กับปัจจัยหลายประการ เช่น ความเร็วของเครือข่าย ความเร็วของหน่วยประมวลผลเครื่อง ปริมาณหน่วยความจำในเครื่องเป็นต้น

การหาค่าที่ระบบยอมรับได้นี้ สามารถทำได้โดยการเชื่อมต่อกับระบบที่วิเคราะห์ จากนั้นหาจำนวนแพ็กเก็ตที่เข้ามาในระบบในลักษณะการใช้งานปกติของแต่ละช่วงเวลา จากนั้นนำค่าสูงสุดที่ได้มาเป็นค่าที่ระบบยอมรับได้ ค่าที่ผ่านการวิเคราะห์และยอมรับได้โดยปกติมีค่าประมาณประมาณ 20, 000 - 30,000 แพ็กเก็ตต่อนาที



รูปที่ 4-4 แสดงการตรวจสอบการส่งแพ็กเก็ตจำนวนมาก

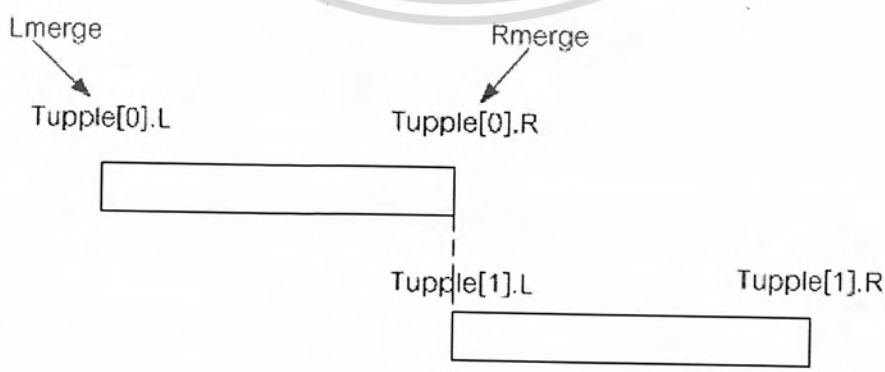
4.3.2.2 ความผิดปกติของแฟร็กเมนต์

การตรวจสอบความผิดปกติ ของแฟร็กเมนต์มีขั้นตอนค่อนข้างซับซ้อน ซึ่งแยกอธิบายตามประเภทของความผิดปกติได้ ดังต่อไปนี้

1. การส่งแพ็กเก็ตที่มีลำดับผิดปกติ และ แพ็กเก็ตที่มีขนาดเหลื่อมล้ำกัน

การวิเคราะห์ความผิดปกติของแพ็กเก็ตในลักษณะนี้ต้องวิเคราะห์หลังจากกระบวนการริแอสเซมเบิ้ลไปแล้ว ดังนั้นจึงนำบัฟเฟอร์เข้ามาช่วยในการเก็บข้อมูล เพื่อนำมาวิเคราะห์ ดังนี้

- Fragment Buffer คือ บัฟเฟอร์ที่เก็บข้อมูลในการวิเคราะห์ ซึ่งเก็บข้อมูลของแพ็กเก็ตไอพี และข้อมูลที่จำเป็นอื่นๆ ไว้ ได้แก่ หมายเลขไอพีของผู้ส่ง (IP_Src) หมายเลขไอพีของผู้รับ (IP_Dst), Idcntification, Protocol, scc, PointArray, Array_Fragment การเก็บข้อมูลดังกล่าวจะเก็บในลักษณะของโครงสร้างข้อมูลแบบลิงส์ลิสต์



รูปที่ 4-5 แสดงการเก็บข้อมูลของตัวแปร Tupple

การเก็บข้อมูลใน Fragment Buffer มีตัวแปรต่างๆ ที่จัดเก็บดังตารางที่ 4-1

IP_Src	IP_Dst	Identification	Protocal	Sec	Pointarray	Arrau_Fragment

ตารางที่ 4-1 แสดงโครงสร้างการเก็บข้อมูลของ Fragment Buffer

การเก็บข้อมูลส่วน Fragment โดยจะเก็บเป็น Array มีข้อมูลตามตารางที่ 4-2

Flag_U	Flag_D	Flag_M	Offset	Size_Data

ตารางที่ 4-2 แสดงโครงสร้างการเก็บข้อมูลของ Fragment

- Overlap Buffer เป็นบัฟเฟอร์ที่เก็บข้อมูลเมื่อตรวจพบว่าเกิดการเหลื่อมล้ำของแพ็กเก็ต มีการเก็บในลักษณะของลิงส์ลิสต์

- Gap Frame Buffer คือ บัฟเฟอร์ที่เก็บข้อมูล เมื่อตรวจพบว่าเกิดการประกอบเฟรมไม่ได้ในลักษณะมีช่องว่างระหว่างแพ็กเก็ต มีการเก็บในลักษณะของลิงค์ลิสต์

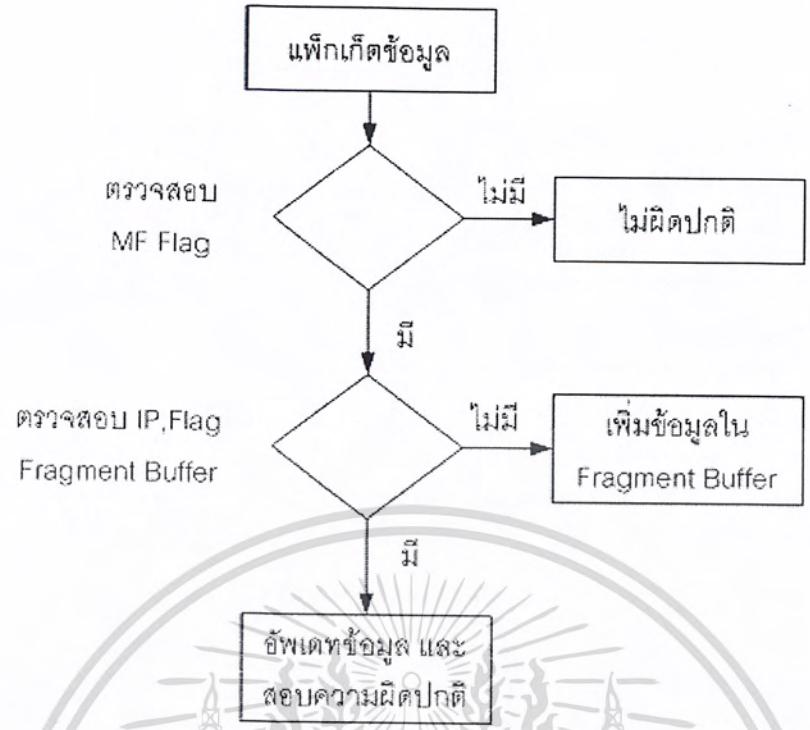
ในการวิเคราะห์ใช้บัฟเฟอร์นี้ร่วมกันโดยเก็บข้อมูลเพ็กเก็ตที่เข้ามาทั้งหมดลงในFragment Buffer และหากแพ็กเก็ต ที่ส่งมาสามารถรวมกันได้ก็รวมกันเป็นแพ็กเก็ตเดียวที่ต่อเนื่องกัน โดยดูจากขอบซ้ายและขอบขวา

แต่หากรวมกันแล้วเกิดความผิดปกติ ให้แจ้งมายัง Overlap Buffer หรือ Gap Frame Buffer แล้วแต่ความผิดปกติที่เกิดขึ้น

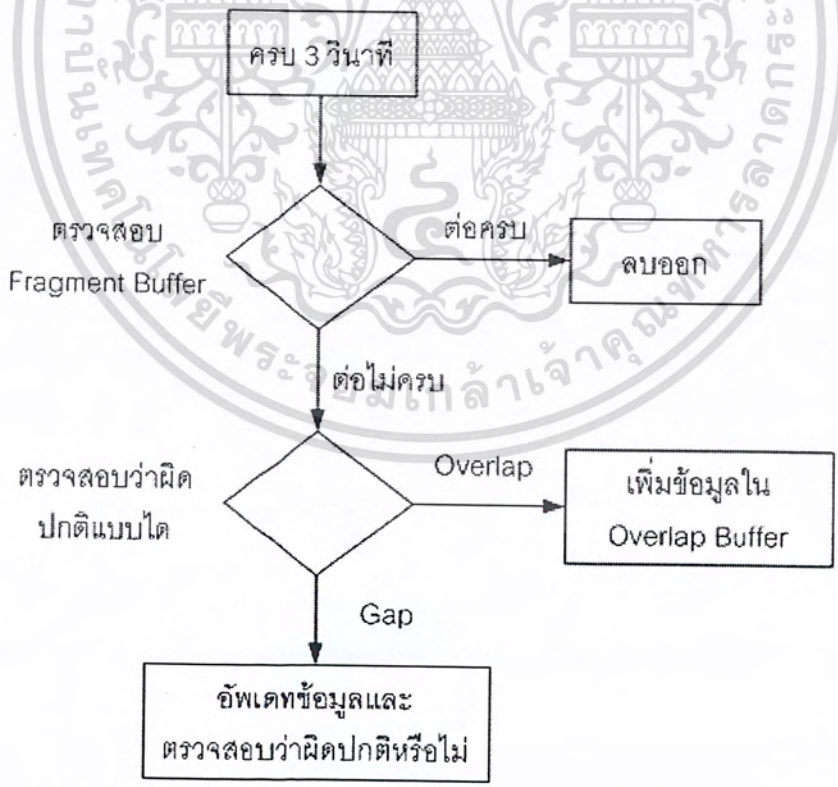
หากไม่มีความผิดปกติเกิดขึ้น เมื่อครบ 3 วินาที โปรแกรมตรวจสอบจะ Fragment Buffer ว่าหากมีแพ็กเก็ตใดยังไม่ได้ประกอบ หรือ ประกอบไม่ครบ ให้เก็บไว้ใน Overlap Buffer หรือ Gap Frame Buffer เช่นเดียวกัน

เมื่อครบ 3 วินาที ข้อมูลใน Overlap Buffer และ Gap Buffer นี้ จะออกมาที่หน้าจอ เพื่อแจ้งให้ผู้ดูแลระบบทราบ หรือ เก็บไว้ใน ล็อกไฟล์ เพื่อบันทึกความผิดปกติที่เกิดขึ้นไว้

หากไม่มีความผิดปกติใด ๆ เกิดขึ้นเลย และ แพ็กเก็ตเหล่านั้นสามารถประกอบเป็นเฟรมได้ อย่างถูกต้อง ให้ลบเฟรมเหล่านั้นออกจากบัฟเฟอร์ทันที เพื่อไม่ให้สิ้นเปลืองเนื้อที่ในการจัดเก็บ



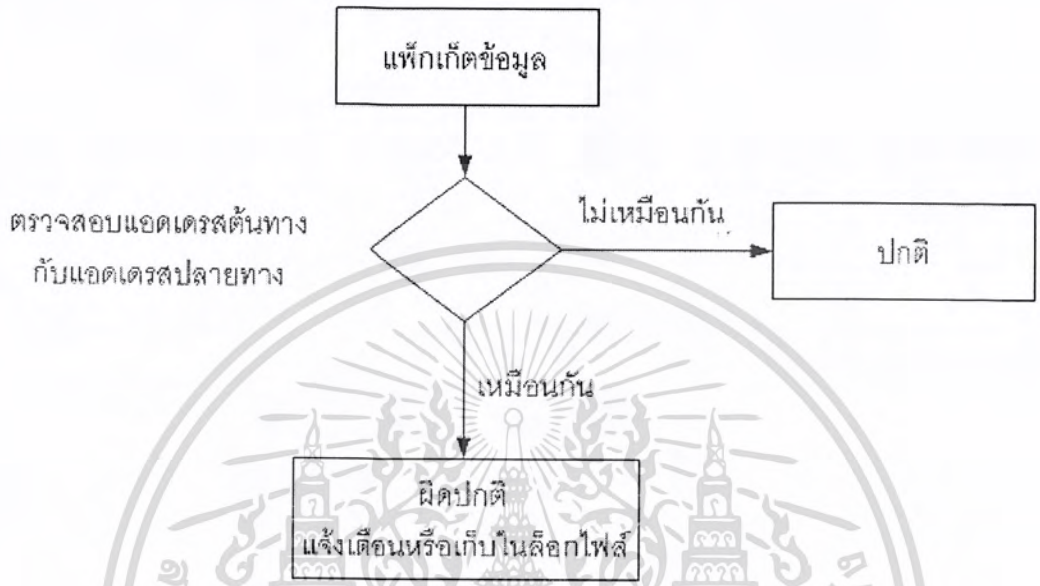
รูปที่ 4-6 แสดงการเก็บข้อมูลลง Fragment Buffer



รูปที่ 4-7 แสดงการตรวจสอบความผิดปกติในการทำแฟร็กเมนเตชัน

2. การส่งแพ็กเก็ตแบบวนลูป

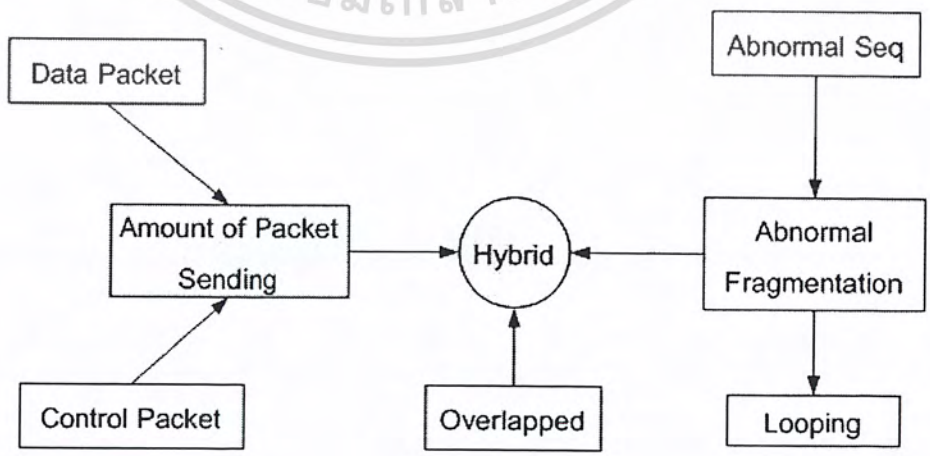
สามารถทำได้โดยการเปรียบเทียบค่าแอดเดรสต้นทาง และ แอดเดรสปลายทางของแพ็กเก็ต ไอพี หากค่า ไอพี ต้นทาง และ ไอพีปลายทาง เป็นค่าเดียวกัน แสดงว่ามีความผิดปกติเกิดขึ้น เพราะทำให้เกิดการส่งในลักษณะวนลูป ซึ่งขั้นตอนการตรวจสอบเป็นไปตามรูปที่ 4-8



รูปที่ 4-8 แสดงการตรวจสอบแพ็กเก็ตที่ส่งแบบวนลูป

4.3.2.3 การโจมตีแบบผสม (Hybrid)

การวิเคราะห์แพ็กเก็ตประเภทนี้ให้นำวิธีการวิเคราะห์ที่กล่าวข้างต้นมาใช้ร่วมกัน เนื่องจากเกิดวิธีการที่ผสมผสานกันระหว่างวิธีต่าง ๆ ที่ได้กล่าวมาแล้ว ซึ่งสามารถแยกวิเคราะห์ออกเป็นแต่ละแบบหรือ วิเคราะห์รวมกันก็ได้

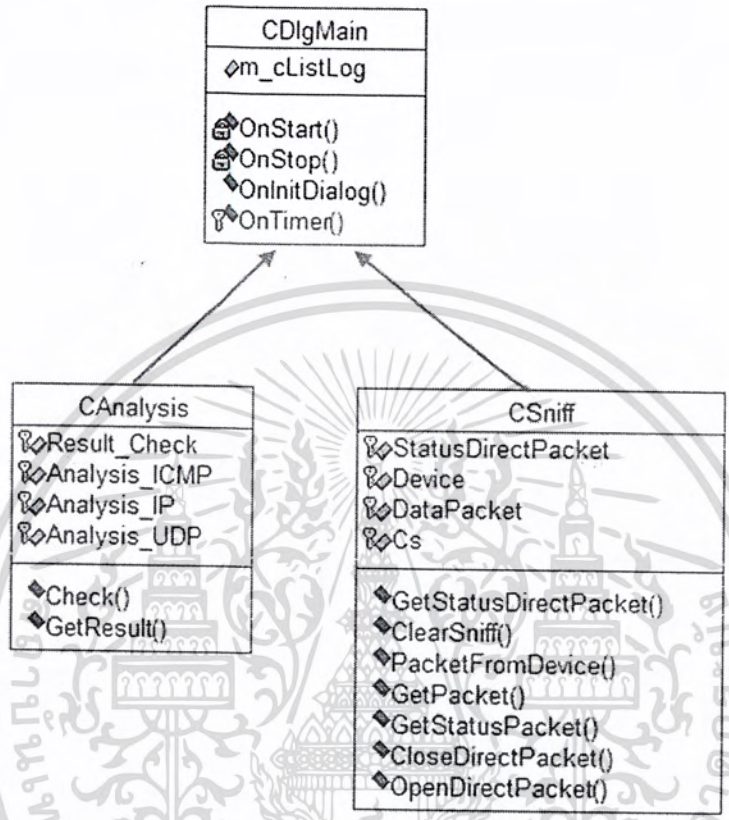


รูปที่ 4-9 แสดงแผนภูมิแสดงประเภทของการโจมตีเพื่อให้ปิดบริการแบบผสม

4.4 โครงสร้าง และการทำงานของโปรแกรม

สามารถแบ่งส่วนของการทำงานของโปรแกรมออกเป็น 2 ส่วนคือ

1. คลาสหลักของระบบ

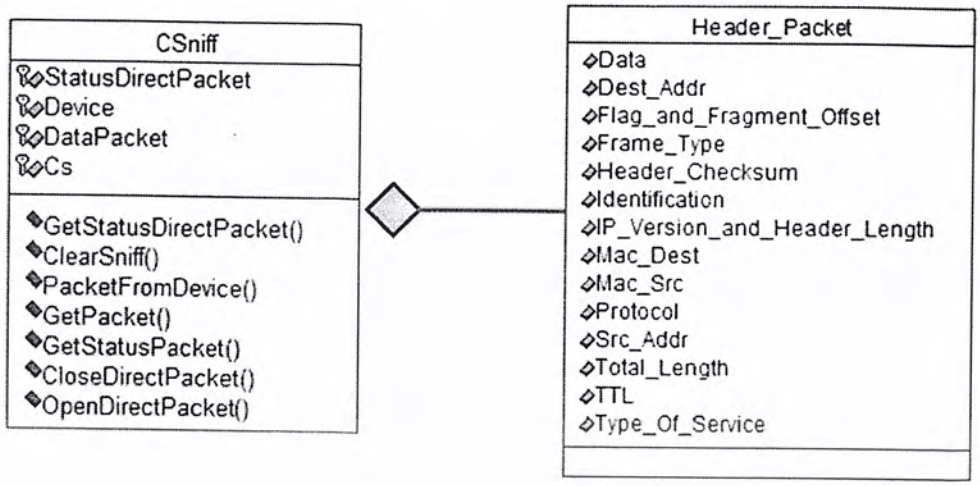


รูปที่ 4-10 คลาสไดอะแกรมหลักของระบบ

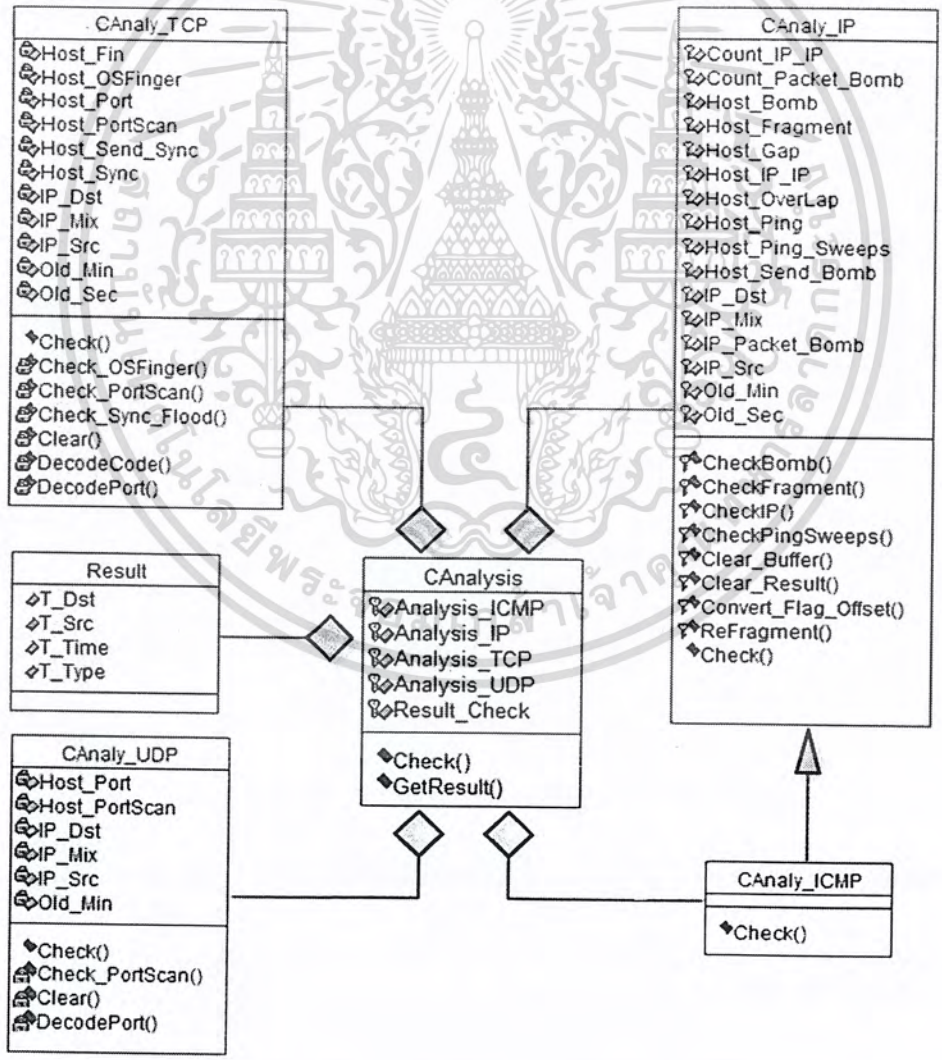
หน้าที่ของส่วนต่างๆ มีดังนี้

1. CDlgMain ทำหน้าที่ควบคุมการทำงานของระบบตรวจจับผู้บุกรุก
2. CSniff ทำหน้าที่ดักจับแพ็กเก็ตข้อมูลในเครือข่ายเก็บไว้ในบัฟเฟอร์
3. CAnalysis ทำหน้าที่วิเคราะห์ข้อมูลในบัฟเฟอร์ที่ดักจับมาได้

2. คลาสไดอะแกรมส่วนตรวจจับแพ็กเก็ต และ วิเคราะห์การบุกรุก



รูปที่ 4-11 คลาสไดอะแกรมส่วนที่ทำการดักจับข้อมูล



รูปที่ 4-12 คลาสไดอะแกรมส่วนวิเคราะห์การบุกรุก

4.4.1 รายละเอียดการทำงานของโปรแกรมแต่ละส่วน

ฟังก์ชันการทำงานแต่ละส่วนสามารถแบ่งตามคลาสได้ดังนี้

1. คลาส CDlgMain ควบคุมการทำงานทั้งหมดของระบบตรวจจับผู้บุกรุก

ทำหน้าที่กำหนด ค่าต่างๆ และควบคุมส่วนต่างๆ ให้ทำงานร่วมกัน ในส่วนของการควบคุมจะใช้คลาส CDlgMain เพื่อเริ่มการทำงานดังต่อไปนี้

ขั้นตอนการทำงาน

1. ตรวจสอบว่ามีการสั่งหยุดการทำงานหรือยัง
2. นำข้อมูลออกมาจากบัฟเฟอร์ของคลาส CSniff
3. เคลียร์ลิงค์ลิสต์ของแพ็กเก็ตในกรณีที่มีการโจมตี โดยวิธี รีแฟร็กเมนต์แบบ Gap โดยฟังก์ชัน ClearSniff() ของคลาส CSniff

- void CDlgMain::OnStart()

จุดมุ่งหมาย

เป็นส่วนที่ใช้ในการเริ่มการทำงานของเธรดต่างๆ และสั่งให้เธรดเริ่มการทำงาน

ขั้นตอนการทำงาน

1. ตรวจว่ามีการเปิดโหมดการทำงานเป็นไดเร็คของการ์ดแลนหรือยัง
2. ทำการเปิดโหมดการทำงานเป็นไดเร็คของการ์ดแลน
3. เริ่มการทำงานของ Thread_Sniff()
4. เริ่มการทำงานของ Thread_Analysis()

- void CDlgMain::OnStop()

จุดมุ่งหมาย

เป็นส่วนที่ใช้ในการหยุดการทำงานของเธรด และ ลบข้อมูลออกจากบัฟเฟอร์

ขั้นตอนการทำงาน

1. ตรวจสอบว่ามีการเปิดโหมดไดเร็คของการ์ดแลน หรือ ไม่ถ้ามีการเปิดจะทำตามขั้นตอนดังนี้คือ
2. สั่งให้ Thread_Sniff() หยุดการทำงาน
3. สั่งให้ Thread_Analysis() หยุดการทำงาน
4. ทำการเคลียข้อมูลในบัฟเฟอร์ของการ์ดแลน
5. ทำการปิดโหมดการทำงานของการ์ดแลน

- UINT Thread_Sniff()

จุดมุ่งหมาย

เป็นส่วนที่ใช้เก็บข้อมูลที่เป็นของเครื่องตนเก็บไว้ในบัฟเฟอร์โดยการทำงานในส่วนนี้จะทำงานเป็นเธรดแยกออกมาจากฟังก์ชันหลัก

ขั้นตอนการทำงาน

1. เช็คว่ามีการสั่งหยุดการทำงานหรือยัง
2. สั่งให้ทำการเก็บข้อมูลจากการ์ดแลน ด้วยฟังก์ชัน PacketFromDevice() ของคลาส CSniff

- UINT Thread_Analysis()

จุดมุ่งหมาย

เป็นส่วนที่ใช้ในการวิเคราะห์แพ็กเก็ตข้อมูลที่อยู่ในบัฟเฟอร์ที่รับเข้ามาจากการ์ดเน็ตเวิร์ค โดยการทำงานในส่วนนี้จะทำงานเป็นเธรดแยกออกมาจากฟังก์ชันหลัก

4.4.2 การเก็บข้อมูล

ในการเก็บข้อมูลจากเน็ตเวิร์คมาทำการวิเคราะห์ โปรแกรมจะเก็บข้อมูลส่วนหัว (Header) ของแต่ละแพ็กเก็ต ในชั้นต่างๆ เช่นเก็บเฉพาะข้อมูลส่วนหัวของแพ็กเก็ตในชั้นไอพี และทีซีพี ในการเก็บข้อมูลจะใช้คลาส Csniff เป็นตัวเก็บข้อมูลโดยมีฟังก์ชันการทำงานดังต่อไปนี้

- BOOL OpenDirectPacket(int NumberDevice)

จุดมุ่งหมาย

เพื่อทำการเปิดไดเร็คแพ็กเก็ตโหมดของการ์ดแลน

ขั้นตอนการทำงาน

1. เช็คว่ามีการเปิดไดเร็คแพ็กเก็ตโหมดของการ์ดแลนอยู่หรือไม่
2. ทำการเปิดไดเร็คแพ็กเก็ตโหมดของการ์ดแลนตามค่า NumberDevice ที่รับเข้ามา

- BOOL CloseDirectPacket()

จุดมุ่งหมาย

เพื่อทำการปิดไดเร็คแพ็กเก็ตโหมดของการ์ดแลน

ขั้นตอนการทำงาน

1. เช็คว่ามีการเปิดไดเร็คแพ็กเก็ตโหมดของการ์ดแลนอยู่หรือไม่
2. ทำการปิดไดเร็คแพ็กเก็ตโหมดของการ์ดแลนที่เปิดอยู่

- BOOL GetStatusPacket()

จุดมุ่งหมาย

เพื่อตรวจสอบว่ามีการรับแพ็กเก็ตเข้ามาเท่าไหร่ แล้วมีการสูญหายไปเท่าไหร่

ขั้นตอนการทำงาน

1. เช็คว่ามีการเปิดไดเร็คแพ็กเก็ตโหมดของการ์ดแลนอยู่หรือไม่
2. ทำการดึงค่าขึ้นมาจาก Winpcap

- BOOL GetPacket()

จุดมุ่งหมาย

เพื่อนำค่าเฮดเดอร์ของแพ็กเก็ตที่เก็บไว้ส่งออกไปให้ฟังก์ชั่นอื่นทีละ 1 แพ็กเก็ต

ขั้นตอนการทำงาน

1. เช็คว่ามีแพ็กเก็ตอยู่ในลิงค์ลิสต์หรือไม่
2. อ่านค่าใน head ของลิงค์ลิสต์ออกมา
3. ลบส่วนหัวของลิงค์ลิสต์ทิ้ง

- BOOL PacketFromDevice()

จุดมุ่งหมาย

เพื่อนำข้อมูลเฮดเดอร์แพ็กเก็ตในการ์ดแลนออกมา แล้วนำมาเข้าลิงค์ลิสต์

ขั้นตอนการทำงาน

1. เช็คว่ามีการเปิดไดเร็คแพ็กเก็ตโหมดของการ์ดแลนอยู่หรือไม่
2. จัดการนำข้อมูลในบัฟเฟอร์ของการ์ดแลนออกมา
3. ทำการแยกออกเป็นแพ็กเก็ตโดยจะแยกเอาเฉพาะส่วนหัวของแพ็กเก็ต
4. นำเข้าไปเก็บในลิงค์ลิสต์โดยที่ 1 โหนด จะเก็บ 1 แพ็กเก็ต

- BOOL ClearSniff()

จุดมุ่งหมาย

เพื่อลบข้อมูล ในลิงค์ลิสต์ทั้งหมด

ขั้นตอนการทำงาน

1. จะทำการลบข้อมูลในลิงค์ลิสต์

- BOOL GetStatusPromiscuous()

จุดมุ่งหมาย

เพื่อเช็คดูว่ามีการเปิดไดเร็คแพ็กเก็ตโหมดของการ์ดแลนหรือไม่

ขั้นตอนการทำงาน

1. เช็คว่ามีการเปิดไดเร็คแพ็กเก็ตโหมดของการ์ดแลนอยู่หรือไม่
2. ส่งผลลัพธ์กลับสู่ฟังก์ชันหลักว่าเปิดหรือไม่

4.4.3 การวิเคราะห์ข้อมูล

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

- BOOL CAnaly_IP::Check()

  จุดมุ่งหมาย

  เพื่อเช็คแพ็กเก็ตที่ได้รับมา แล้วรายงาน

  ขั้นตอนการทำงาน

  1. ส่งแพ็กเก็ตที่ได้รับมาไปวิเคราะห์ เพื่อแจ้งผล
  2. ตรวจสอบว่าแพ็กเก็ตไหนที่ยังไม่สามารถรีแฟร็กเมนต์ได้ในระยะเวลาที่กำหนด เพื่อแจ้งผล

- BOOL CAnaly_ICMP::Check()

  จุดมุ่งหมาย

  เพื่อเช็คแพ็กเก็ตที่ได้รับมา แล้วรายงาน

  ขั้นตอนการทำงาน

  1. ส่งแพ็กเก็ตที่ได้รับมาไปวิเคราะห์ เพื่อแจ้งผล

- BOOL CAnalysis::Check()

  จุดมุ่งหมาย

  เพื่อแยกชนิดของแพ็กเก็ตว่าเป็นชนิดไหนเพื่อจะได้นำไปวิเคราะห์ ต่อไป

  ขั้นตอนการทำงาน

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

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

จุดมุ่งหมาย

เพื่อตรวจสอบว่ามีแพ็กเก็ตจากหมายเลขไอพีหนึ่ง ไปยังอีกหมายเลขไอพีหนึ่งมากเกินไปหรือไม่ ในเวลาที่เหมาะสม

ขั้นตอนการทำงาน

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

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

จุดมุ่งหมาย

เพื่อตรวจสอบว่ามีแพ็กเก็ตจากหมายเลขไอพีต้นทางไปยังหมายเลขไอพีปลายทางเป็นหมายเลขไอพีเดียวกันหรือไม่

ขั้นตอนการทำงาน

1. นำแพ็กเก็ตที่ได้มาดูว่ามาจากหมายเลขไอพีไหน แล้วส่งไปยังหมายเลขไอพีไหน
2. ดูว่าหมายเลขไอพีตรงกันหรือไม่
3. ถ้าตรงกันจะมีการเปรียบเทียบกับข้อมูลของการโจมตีแบบนี้สำหรับหมายเลขไอพีนี้ ว่ามีการโจมตีแบบนี้ไปแล้วหรือยัง เพื่อจะได้ไม่มีการแจ้ง บ่อยเกินไป

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

จุดมุ่งหมาย

เพื่อแยกแพ็กเก็ตตามหมายเลขไอพีต้นทาง, หมายเลขไอพีปลายทาง, โพรโตคอลและฟิลด์ Identifier ตรวจสอบว่ามีแพ็กเก็ตจากหมายเลขไอพีต้นทางไปยังหมายเลขไอพีปลายทางต้องการทำแฟร็กเมนต์หรือไม่

ขั้นตอนการทำงาน

1. เช็คว่าแพ็กเก็ตที่ได้รับมาเป็นแพ็กเก็ตที่จะต้องทำรีแฟร็กเมนต์หรือไม่
2. จะดูว่ามีข้อมูลเดิมอยู่หรือไม่โดยจะดูที่หมายเลขไอพีต้นทาง ,หมายเลขไอพีปลายทาง , โพรโตคอลและ Identification
3. นำเข้าสู่ลิงค์ลิสต์

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

จุดมุ่งหมาย

เพื่อตรวจสอบว่าแพ็กเก็ตมีปัญหาเรื่องการแฟร็กเมนต์หรือไม่

ขั้นตอนการทำงาน

1. เช็คว่าในแพ็กเก็ตที่ได้รับมาแล้วนั้นมี ส่วนหัวของแพ็กเก็ตและส่วนท้ายของแพ็กเก็ตหรือยัง
2. จะลองประกอบดูว่าสามารถประกอบได้หรือยัง
3. ถ้าประกอบได้จะลบออกจากลิงค์ลิสต์

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

จุดมุ่งหมาย

เพื่อเคลียร์ข้อมูลการโจมตี

ขั้นตอนการทำงาน

1. ทำการเคลียร์ข้อมูลการโจมตี

4.4.4 การรายงานผล

ในส่วนของการแสดงผลในส่วนของ Client จะมีการแสดง ผลการโจมตีที่สามารถตรวจจับได้ โดยที่แสดงผลผ่านตัวแปร m_cListLog ซึ่งเป็นตัวแปรชนิด CListBox โดยแสดงออกที่หน้าหลัก และจะมีการส่ง ข้อมูลของการโจมตีที่ได้ไปยังเครื่องเซิร์ฟเวอร์ (Server) กลางด้วย

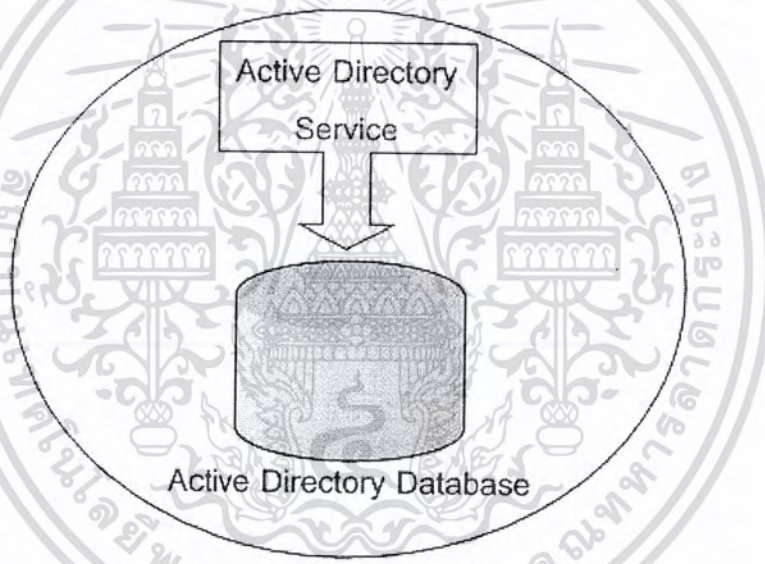
บทที่ 5

Windows 2000 Active Directory

เป็นบริการหนึ่งของ Windows 2000 Server เกี่ยวกับการจัดเก็บข้อมูลของทรัพยากรที่มีอยู่ในเครือข่าย ในไดเรกทรอรี โดยทรัพยากรมีความหมายครอบคลุม รายชื่อผู้ใช้ รายชื่อเครื่องที่อยู่ในเครือข่าย รายชื่อแชร์โฟลเดอร์บนไฟล์เซิร์ฟเวอร์ และรายชื่อเครื่องพิมพ์ที่ให้บริการ โดยแอ็คทีฟไดเรกทรอรี จะมีฐานข้อมูลในตัวเองสำหรับจัดเก็บรายละเอียดของทรัพยากร

5.1 ส่วนประกอบของวินโดวส์ 2000 แอ็คทีฟไดเรกทอรี

การทำงานแบ่งออกเป็นสองส่วนคือดังรูปที่ 5-1



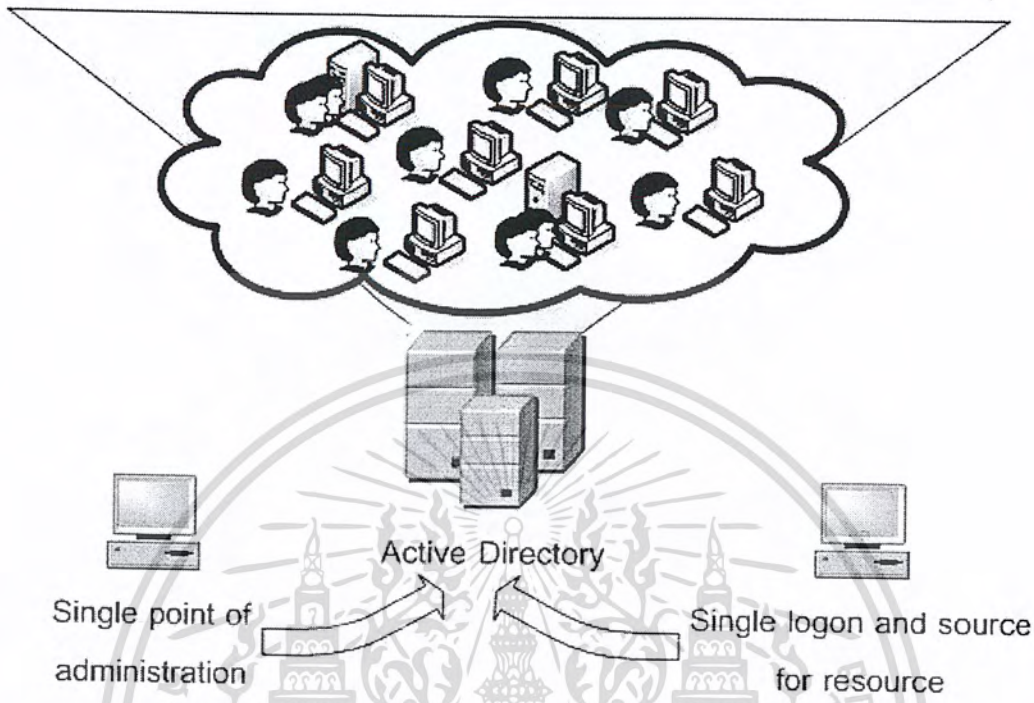
รูปที่ 5-1 แสดงส่วนการทำงานของแอ็คทีฟไดเรกทอรี (Active Directory)

5.1.1 แอ็คทีฟไดเรกทอรีเซอร์วิส (Active Directory Service)

- ให้บริการแก่ผู้บริหารระบบเครือข่ายในการเรียกค้นและจัดการกับบัญชีรายชื่อผู้ใช้ (User Account) และรายชื่อกลุ่ม (Group) ที่จัดเก็บไว้ในแอ็คทีฟไดเรกทอรีดาต้าเบส เช่น การสร้างบัญชีรายชื่อผู้ใช้ใหม่ให้กับผู้ใช้ในองค์กร การตั้งรหัสผ่าน และอื่นๆ รวมถึงให้บริการในการเซตค่า คอนฟิกูเรชัน(configuration) ต่างๆ ให้กับผู้ใช้ ไม่ว่าจะเป็น Domain Policy หรือ Group Policy ต่างๆ ดังนั้น เราจึงสามารถกล่าวได้ว่าแอ็คทีฟเป็นจุดศูนย์กลางในการบริหารระบบเน็ตเวิร์กของ Windows 2000

- อำนวยความสะดวกในการค้นหารายชื่อทรัพยากร ในระบบเครือข่ายที่มีคุณสมบัติ (Attributes) ตามที่ผู้ใช้หรือผู้บริหารระบบต้องการ เช่น ค้นหาว่าเครื่องพิมพ์ใดบ้างที่มีคุณสมบัติ "สามารถพิมพ์สีได้" เป็นต้น แอ็คทีฟไดเรกทอรีเซอร์วิสจะเข้าไปค้นหาจากแอ็คทีฟไดเรกทอรีดาต้าเบส แล้วแสดง

รายชื่อเครื่องพิมพ์ทั้งหมดที่มีคุณสมบัติดังกล่าว มาให้ผู้ใช้เลือก พร้อมทั้งบอกรายละเอียดอื่นๆ ให้ทราบด้วย เช่น เครื่องพิมพ์นี้ติดตั้งไว้ที่เซิร์ฟเวอร์ชื่ออะไร



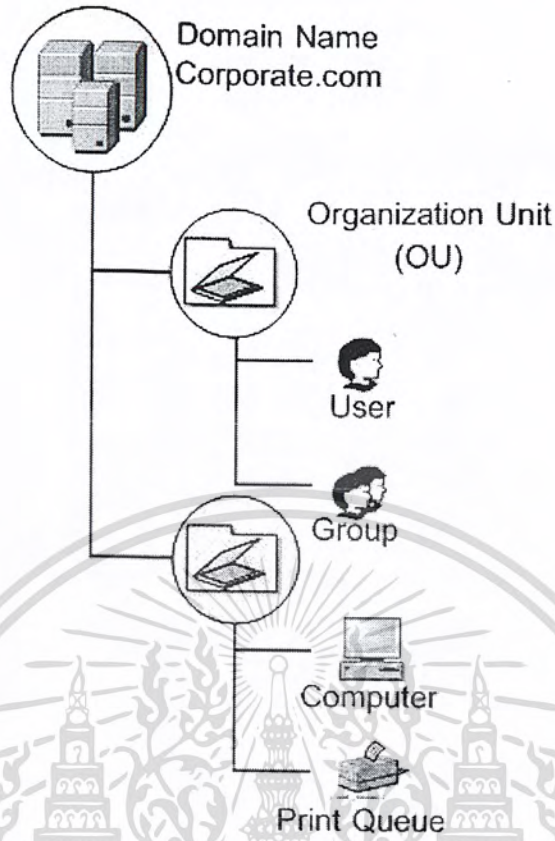
รูปที่ 5-2 การให้บริการของแอ็คทีฟไดเรกทอรีเซอร์วิส (Active Directory Service)

5.1.2 แอ็คทีฟไดเรกทอรีดาต้าเบส (Active Directory Database)

ทำหน้าที่ในการจัดเก็บข้อมูลในระบบไดเรกทอรี โดยการเข้าถึงดาต้าเบสจะต้องผ่านการ ล็อกอิน (Login) ถึงจะมีสิทธิ

5.2 ข้อมูลที่ถูกจัดเก็บอยู่ในแอ็คทีฟไดเรกทอรีดาต้าเบส

- บัญชีรายชื่อผู้ใช้และบัญชีรายชื่อกลุ่ม (User account / Group Account) ของผู้ใช้ในระบบเครือข่ายและรายชื่อเครื่องคอมพิวเตอร์ในเครือข่าย บัญชีรายชื่อเหล่านี้ถือว่าเป็นออบเจ็กต์ประเภทหนึ่งเหมือนกัน



รูปที่ 5-3 โครงสร้างโดยรวมของแอ็คทีฟไดเรกทอรีดาต้าเบส (Active Directory Database)

- ออบเจ็กต์ต่างๆที่ใช้เป็นตัวแทนของทรัพยากรในระบบเครือข่าย เช่น ออบเจ็กต์ที่เป็นตัวแทนของแชร์โฟลเดอร์(Shared Folder Object) ออบเจ็กต์ที่เป็นตัวแทนของเครื่องพิมพ์

- คอนเทนเนอร์ต่างๆที่เป็นของระบบตั้งแต่เริ่มต้นเช่นคอนเทนเนอร์ที่ชื่อUsers, Computers และ Built-in

- คอนเทนเนอร์พิเศษที่เรียกว่า Organizational Unit (OU)

- System Configuration ต่างๆ ของโครงสร้าง Active Directory ทั้งหมด เช่น Schema , Global Catalog

- เป็นที่เก็บ Group Policy Object (GPO) เพื่อใช้ในการควบคุม และจัดการเครื่องคอมพิวเตอร์ ของผู้ใช้

- อื่นๆอีกมากมายแล้วแต่จะขยายเพิ่มเติมในอนาคตเช่นออบเจ็กต์ที่เป็นตัวแทนของ อุปกรณ์สื่อสารในระบบเน็ตเวิร์ก

5.2.1 คอนเทนเนอร์ (Container) คลาส (Class) และ แอตตริบิวต์ (Attributes)

5.2.1.1 คอนเทนเนอร์ (Container)

ตู้บรรจุออบเจ็กต์อื่นเพื่อแยกหมวดหมู่ในการออกแบบ ในโครงสร้างของแอ็คทีฟไดเรกทอรีดาต้าเบส ออบเจ็กต์ รายชื่อทรัพยากรทุกอย่างที่อยู่แอ็คทีฟไดเรกทอรีดาต้าเบสจะถูกมองเสมือนว่าเป็น ออบเจ็กต์หนึ่ง ดังนั้น แต่ละออบเจ็กต์ที่เก็บอยู่ในแอ็คทีฟไดเรกทอรีดาต้าเบสจึงถือได้ว่าเป็นตัวแทนของทรัพยากรที่มีอยู่ในระบบเครือข่ายและเป็นที่จัดเก็บรายชื่อผู้ใช้ รายชื่อกลุ่ม รายชื่อเครื่องคอมพิวเตอร์ของผู้ใช้และเครื่องคอมพิวเตอร์ในเน็ตเวิร์ก

5.2.1.2 คลาส (Class)

องออบเจ็กต์จะเป็นตัวแบ่งประเภทของออบเจ็กต์ ตัวอย่างเช่น รายชื่อผู้ใช้ ทุกรายชื่อที่สร้างขึ้นมาให้กับผู้ใช้ในระบบเครือข่ายจะถูกมองว่าเป็นออบเจ็กต์ที่มีอยู่ในคลาส Users และรายชื่อเครื่องคอมพิวเตอร์ (Computer Account) ในเน็ตเวิร์กก็ถูกมองว่าเป็นออบเจ็กต์ที่จัดอยู่ในคลาส Computer นอกจากนั้น ทุกออบเจ็กต์ที่อยู่ในคลาสเดียวกันจะต้องมีแอตตริบิวต์ และข้อมูลกำหนด (definition) ต่างๆ เหมือนกัน

5.2.1.3 แอตตริบิวต์ (Attribute)

ใช้ในการบ่งบอกคุณลักษณะ คุณสมบัติ และพฤติกรรมต่างๆ ของออบเจ็กต์ โดยออบเจ็กต์ทุกตัวที่อยู่ในคลาสเดียวกันจะต้องมีแอตตริบิวต์ทุกอย่างเหมือนกัน เช่น ออบเจ็กต์ในคลาส User (รายชื่อผู้ใช้) จะมีแอตตริบิวต์ First Name ,Last Name, Logon Name (ชื่อที่ใช้ในการล็อกออนเข้าสู่เน็ตเวิร์ก), Password (รหัสผ่าน) และอื่นๆ เหมือนกัน แต่สิ่งที่ทำให้ออบเจ็กต์แต่ละตัวแตกต่างกันไปก็คือ ค่า (value) ของแอตตริบิวต์

5.2.2 สกีมา (Schema)

เป็นฐานข้อมูลส่วนย่อยที่เก็บอยู่ในแอ็คทีฟไดเรกทอรีดาต้าเบส ซึ่งทำหน้าที่เก็บรวบรวมข้อกำหนดเกี่ยวกับคลาสของออบเจ็กต์ทุกคลาส เช่น ออบเจ็กต์ที่อยู่ในแต่ละคลาสจะต้องมีแอตตริบิวต์อะไร ค่าของแต่ละแอตตริบิวต์มีคุณสมบัติเป็นอะไร Schemaในแอ็คทีฟไดเรกทอรีดาต้าเบสของวินโดวส์ 2000 สามารถเพิ่มขยายได้โดยผู้บริหารระบบเครือข่าย ได้แก่ การเพิ่มคลาสประเภทใหม่ๆ เข้าไป หรือการเพิ่ม แอตตริบิวต์ใหม่ให้กับคลาสเดิมที่มีอยู่แล้ว

5.2.3 ออกาไนเซชันยูนิต (Organization Unit)

จะทำหน้าที่เสมือนคอนเทนเนอร์ ถูกใช้เพื่อจัดแบ่งออบเจ็กต์ต่างๆ ให้เป็นกลุ่ม สอดคล้องกับโครงสร้างตามลำดับชั้นของการบริหารหน่วยงานหรือแผนกต่างๆ ในองค์กร หรือให้สอดคล้องกับที่ตั้งของหน่วยงาน/สาขาต่างๆ ภายในองค์กร

ประโยชน์ที่ได้รับเมื่อมีการสร้าง OU ขึ้นมาเพื่อจัดแบ่งกลุ่มของออบเจ็กต์

- ผู้บริหารระบบโดเมนสามารถจัดระเบียบและมองเห็นรายชื่อผู้ใช้ได้ง่ายว่า

- ลดภาระในการจักการทรัพยากร กล่าวคือ สามารถจัดการกับออบเจ็กต์เป็นกลุ่มได้ทีเดียว เช่น การกำหนดนโยบายการรักษาความปลอดภัย (Security Policy) และนโยบายในการติดตั้งซอฟต์แวร์ (Software Group Policy) ให้กับออบเจ็กต์ประเภท User หรือ Computer ที่อยู่ภายใต้ OU นั้นๆ

- ผู้บริหารระบบสามารถกำหนดสิทธิ (Permission) ต่างๆ ของผู้ใช้ในการเข้าถึงออบเจ็กต์ผ่านทาง OU ได้

- ผู้ดูแลโดเมนในระดับบนสุดสามารถกระจายอำนาจการจัดการให้ผู้ดูแลแผนกหรือ ภูมิภาคนั้นๆ เป็นผู้มีอำนาจเด็ดขาดในการจัดการกับรายชื่อผู้ใช้ และทรัพยากรต่างๆ ภายใต้แผนกของตนได้อย่างเต็มที่ ลักษณะนี้เรียกว่า "Delegate Administrative Control" ซึ่งทำได้โดยการกำหนดให้ผู้รับผิดชอบแผนกนั้นมีสิทธิสมบูรณ์บน OU ที่ตนต้องรับผิดชอบ จากนั้นเขาสามารถจัดการกับออบเจ็กต์ที่อยู่ภายใต้ OU นั้นอย่างเต็มที่

5.2.4 โดเมน (Domain)

โดเมนเป็นการรวมกลุ่มทรัพยากรต่างๆ ในระบบเครือข่ายต่างๆ ตั้งแต่ เครื่องพิมพ์ แชร์โฟลเดอร์ เครื่องคอมพิวเตอร์ เครื่องเซิร์ฟเวอร์ รายชื่อผู้ใช้เข้าไว้ด้วยกันให้มีความหมายในเชิงบริหาร และการจัดการ โดยทรัพยากรต่างๆ ข้างต้นจะต้องอยู่ภายใต้ชื่อโดเมนเดียวกัน และชื่อโดเมนจะบ่งบอกให้เห็นถึงชื่อของหน่วยงานหรือชื่อองค์กรนั้นๆ ซึ่งผู้บริหารระบบโดเมนจะมีความสามารถเต็มที่ในการจัดทรัพยากรต่างๆ ที่อยู่ภายใต้โดเมน

โดเมนยังถูกมองได้ว่าเป็นเสมือนขอบเขตของระบบรักษาความปลอดภัย(Security Boundary) และขอบเขตของการบริหาร (Administrative Boundary) ด้วย ดังจะเห็นได้จากข้อเท็จจริงต่อไปนี้

- ผู้ใช้ เมื่อผู้ใช้เข้ามาในโดเมนใดโดเมนหนึ่งเรียบร้อยแล้ว ผู้ใช้สามารถใช้ทรัพยากรต่างๆ ทั้งหมดที่อยู่ภายใต้โดเมนนั้นได้ตามสิทธิที่ตัวเองมี แต่ผู้ใช้จะไม่สามารถเข้าไปใช้งานทรัพยากรที่ติดตั้งอยู่ภายใต้โดเมนอื่นได้ จนกว่าผู้บริหารระบบของโดเมนอื่นจะกำหนดสิทธิอนุญาตไว้ให้

- ผู้บริหารระบบ มีสิทธิเต็มที่ในการจัดการบัญชีรายชื่อผู้ใช้และทรัพยากรต่างๆ ได้เฉพาะภายในโดเมนของตนเท่านั้น ไม่มีสิทธิเข้าไปวุ่นวายหรือจัดการกับโดเมนอื่น (ยกเว้นว่าผู้บริหารโดเมนอื่นจะได้อนุญาตเป็นกรณีพิเศษ)

ภายในโดเมนหนึ่งๆจะมีฐานข้อมูลส่วนกลางที่ทำหน้าที่จัดเก็บรวบรวมรายละเอียดของทรัพยากรและรายชื่อต่างๆข้างต้น เพื่อให้เครื่องที่ทำหน้าที่เป็นเซิร์ฟเวอร์ทุกตัวในระบบสามารถอ้างอิง และใช้งานฐานข้อมูลส่วนกลางนี้ได้ ซึ่งถูกจัดเก็บไว้ที่เครื่อง Windows 2000 Server ที่ทำหน้าที่เป็น Domain Controller (โดเมนคอนโทรลเลอร์)

โหมดของโดเมน (Domain Modes) แบ่งออกได้เป็น 2 โหมดคือ

1. Mixed Mode หลังจากติดตั้งแอ็คทีฟไดเรกทอรีเพื่อสร้างโดเมนของ Windows 2000 ขึ้นมาแล้ว โหมดของโดเมนจะถูกกำหนดให้เป็น Mixed Mode เพื่อให้โดเมนคอนโทรลเลอร์ของ Windows 2000 สามารถซิงโครไนซ์บัญชีรายชื่อผู้ใช้และสื่อสารกับเครื่อง Backup Domain Controller (BDC) ของ Windows NT 4.0 ที่ทำงานอยู่ในโดเมนเดียวกันได้ จุดประสงค์อันนี้ก็เพื่อรองรับการอัปเกรดจากเน็ตเวิร์ก

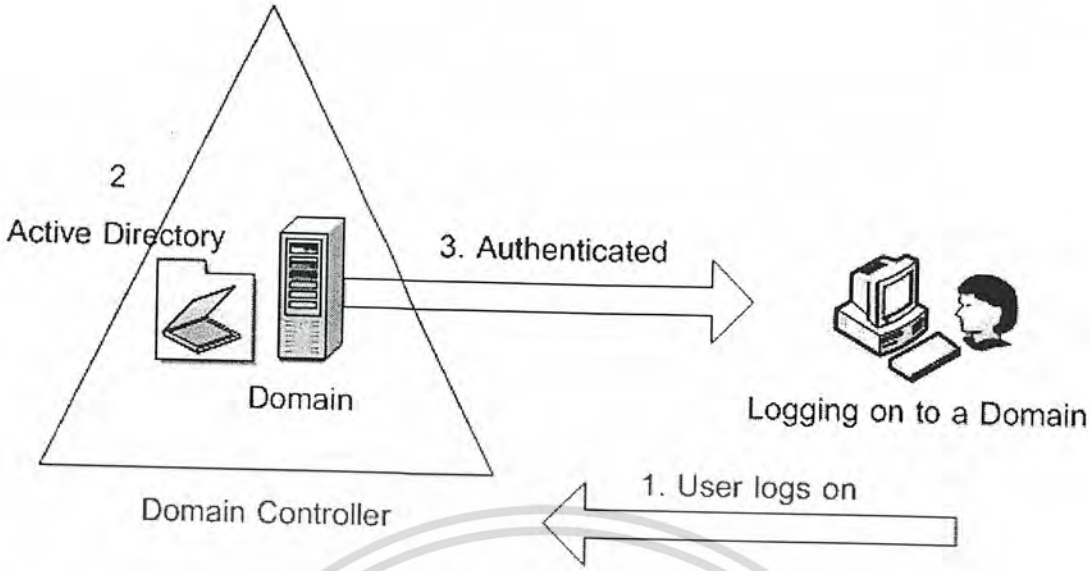
ของ NT 4.0 มาเป็น Windows 2000 นั่นเอง เพราะในการอัปเกรด เราจะเริ่มอัปเกรดที่เครื่อง PDC ของ Windows NT 4.0มาเป็นโดเมนคอนโทรลเลอร์ของ Windows 2000 ก่อน โดยที่เครื่อง BDC ยังคงทำงานอยู่บน Windows NT Server 4.0

2. Native Mode หลังจากที่เครื่องโดเมนคอนโทรลเลอร์ทุกตัวได้รับการอัปเกรดจาก NT 4.0 มาเป็นโดเมนคอนโทรลเลอร์ที่ทำงานบน Windows 2000 Server แล้ว เราสามารถเปลี่ยนจาก Mixed Mode ให้เป็น Native Modeได้ ซึ่งการทำงานใน Native Mode มีข้อได้เปรียบกว่า Mixed Mode หลายเรื่องของการจัดรายชื่อกลุ่มซ้อนกลุ่ม หรือที่เรียกว่าการทำ Group Nesting และการประยุกต์ใช้รายชื่อกลุ่มพิเศษที่ชื่อ Universal Group Nesting

5.2.5 โดเมนคอนโทรลเลอร์ (Domain Controller)

โดเมนคอนโทรลเลอร์จะมีหน้าที่ตรวจสอบรายชื่อผู้ใช้ที่ต้องการใช้งานระบบตรงกับบัญชีรายชื่อผู้ใช้ในแอ็คทีฟไดเรกทอรีดาต้าเบสหรือไม่ ก่อนจะอนุญาตให้ ผู้ใช้เข้ามาใช้งานทรัพยากรต่างๆในระบบได้ และเมื่อมีเข้าใช้งานระบบเสร็จเรียบร้อยแล้ว ผู้ใช้สามารถติดต่อเข้าไปใช้ทรัพยากร (แชร์โฟลเดอร์ เครื่องพิมพ์) จากเซิร์ฟเวอร์ใดก็ได้ที่อยู่ภายใต้โดเมนนั้นๆ ส่งผลให้ผู้ใช้สามารถใช้รายชื่อเดียวเพื่อการเข้าถึงทรัพยากรและเซิร์ฟเวอร์ต่างๆ ได้ทั้งหมด ลักษณะนี้เรียกว่า "Single Network Logon" อีกทั้งผู้บริหารระบบเครือข่ายก็ได้รับความสะดวกด้วย เพราะเสียเวลาในการสร้างรายชื่อผู้ใช้ของผู้ใช้ในโดเมนขึ้นมาเพียงครั้งเดียวแล้วจัดเก็บในแอ็คทีฟไดเรกทอรีดาต้าเบส จากนั้นไม่ว่าในอนาคตจะมีเซิร์ฟเวอร์เกิดขึ้นมากี่ตัว ผู้บริหารเครือข่ายก็ไม่จำเป็นต้องไปเสียเวลาสร้างรายชื่อผู้ใช้ขึ้นมาใหม่ที่เซิร์ฟเวอร์เหล่านั้นอีก เพียงแค่ติดตั้งให้เซิร์ฟเวอร์เหล่านี้เข้ามาเป็นสมาชิกของโดเมนที่มีอยู่แล้วเท่านั้นเอง เซิร์ฟเวอร์เหล่านี้จะรู้จักรายชื่อผู้ใช้ทั้งหมดในโดเมนทันที

ในรูปที่ 5-4 เมื่อผู้ใช้ขอเข้าใช้งานระบบ (ขั้นที่ 1) โดเมนคอนโทรลเลอร์จะตรวจสอบชื่อผู้ใช้ที่ระบุเข้ามามีอยู่ในแอ็คทีฟไดเรกทอรีดาต้าเบสหรือไม่ และรหัสผ่านที่ระบุถูกต้องหรือไม่ (ขั้นที่ 2) ถ้าถูกต้อง โดเมนคอนโทรลเลอร์ยอมรับผู้ใช้ และอนุญาตให้ใช้ทรัพยากรต่างๆ ภายใต้โดเมนได้ ในขั้นนี้ถือว่าผู้ใช้ผ่านตรวจสอบจากโดเมนคอนโทรลเลอร์แล้ว (ขั้นที่ 3) ผ่านการพิสูจน์ตน (Authenticate) โดเมนคอนโทรลเลอร์มีหน้าที่ควบคุมการให้บริการแอ็คทีฟไดเรกทอรี เก็บรักษาและดูแลฐานข้อมูลสำคัญของโดเมนในแอ็คทีฟไดเรกทอรีดาต้าเบส



รูปที่ 5-4 แสดงการทำงานของการเข้าใช้งานระบบบนโดเมน

5.2.6 Replication (เรพลิเคชัน)

แอ็คทีฟไดเรกทอรีดาต้าเบสที่จัดเก็บอยู่บนเครื่องโดเมนคอนโทรลเลอร์เป็นฐานข้อมูลที่มีความสำคัญมากต่อการทำงานของระบบโดเมน ไมโครซอฟต์จึงได้ออกแบบให้ใน 1 โดเมนสามารถมีเครื่องโดเมนคอนโทรลเลอร์ได้มากกว่า 1 ตัว โดยที่ทุกๆตัวเก็บแอ็คทีฟไดเรกทอรีดาต้าเบสชุดเดียวกัน

ในยามปกติทุกๆ โดเมนคอนโทรลเลอร์จะช่วยกันทำหน้าที่ตรวจสอบการเข้าใช้งานระบบของผู้ใช้ และให้บริการ แอ็คทีฟไดเรกทอรีดังที่ได้กล่าวไปข้างต้น แต่ในกรณีที่ตัวใดตัวหนึ่งเกิดปัญหา ตัวอื่นที่เหลืออยู่ก็สามารถให้บริการแก่ผู้ใช้ในโดเมนต่อไปได้ เพื่อให้แอ็คทีฟไดเรกทอรีดาต้าเบสที่เก็บไว้บนแต่ละโดเมนคอนโทรลเลอร์ในโดเมนมีความสอดคล้องกันอยู่ตลอดเวลา โดเมนคอนโทรลเลอร์จะมีระยะอัพเดตข้อมูลในแอ็คทีฟไดเรกทอรีดาต้าเบส เรียกว่า การเรพลิเคชัน ระหว่างโดเมนคอนโทรลเลอร์

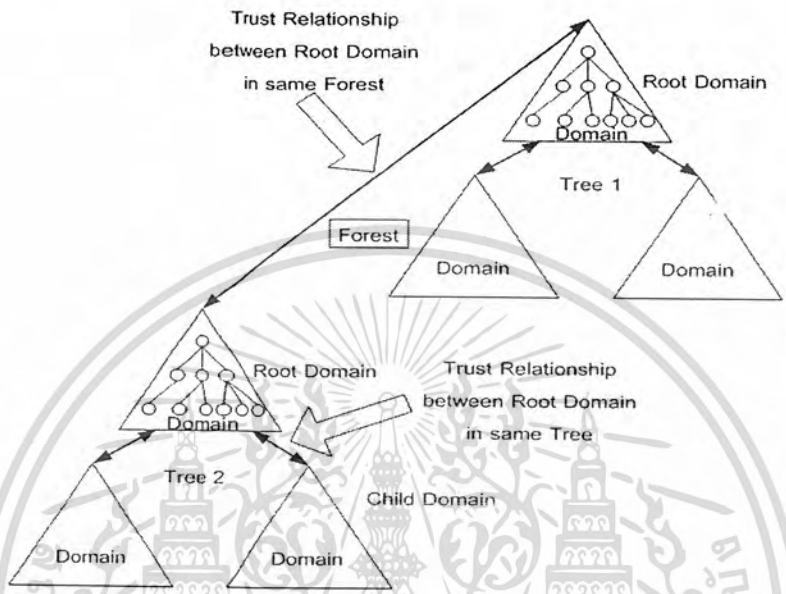
5.2.7 Trust Relationship

Trust Relationship เป็นเสมือน "Communication Link" เชื่อมโยงระหว่าง 2 โดเมน โดยการที่โดเมนหนึ่งเชื่อใจอนุญาตให้ผู้ใช้ของโดเมนอื่นสามารถเข้ามาใช้งานทรัพยากรที่อยู่ในโดเมนของตนได้ผ่านเน็ตเวิร์กที่หน้าจอเซิร์ฟเวอร์ในโดเมนของตน โดยอาศัยการพิสูทธ์ตนจากแอ็คทีฟไดเรกทอรีดาต้าเบสบนโดเมนคอนโทรลเลอร์ในโดเมนเดิมของผู้ใช้คนนั้นๆ Trust Relationship จึงเป็นสิ่งที่ใช้ในการแลกเปลี่ยนทรัพยากรระหว่างโดเมน หรือเราสามารถกล่าวได้อีกอย่างหนึ่งว่า Trust Relationship เป็นเครื่องมือที่ใช้สร้างความสัมพันธ์ระหว่างโดเมนกับโดเมน

5.2.8 ทรี (Tree) และ ฟอร์เรสต์ (Forest)

ทรี (tree) เป็นการรวมกลุ่มโดเมนเข้าด้วยกันผ่านทาง Trust Relationship เพื่อให้เกิดการแลกเปลี่ยนการใช้งานทรัพยากรข้ามระหว่างโดเมนในองค์กรเดียวกัน โดเมนที่อยู่ในระดับบนสุดของทรี

เรียกว่า Parent Domain และโดเมนที่อยู่ข้างใต้ Parent Domain จะถูกเรียกว่า Child Domain ฟอร์เรสต์เป็นการรวมกลุ่มทรีหลายๆ ทรีเข้าด้วยกันดังแสดงในรูป เพื่อการแลกเปลี่ยนทรัพยากรระหว่างทรีโดยทั่วไป การสร้างฟอร์เรสต์จะถูกใช้เพื่อการสื่อสารระหว่างสององค์กรมากกว่า อย่างเช่น บริษัทหนึ่งก็จะนำเอาทรีของตนมาเชื่อมโยง (Join) เป็นฟอร์เรสต์



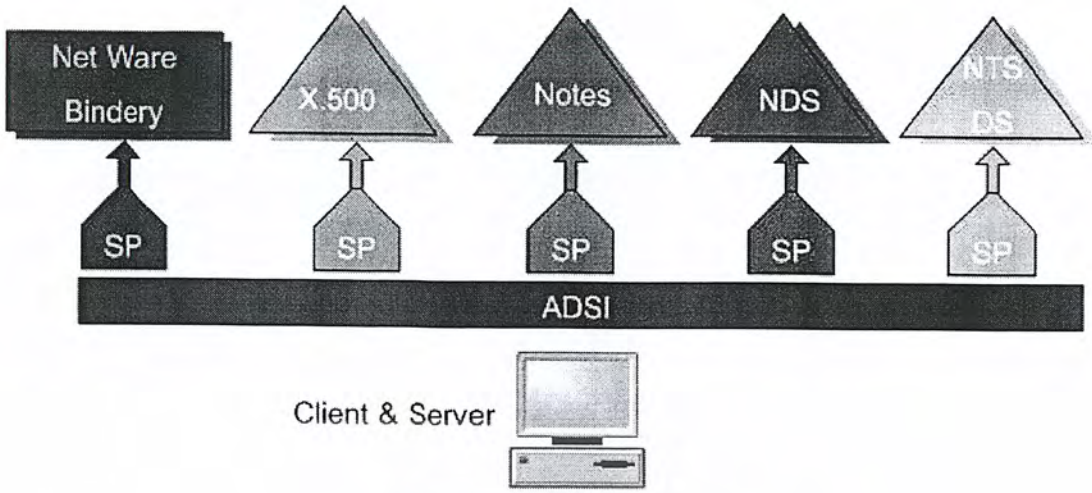
รูปที่ 5-5 แสดงโครงสร้างของ ทรี (Tree) และ ฟอร์เรสต์ (Forest)

5.2.9 ไซต์ (Site)

การจัดรวม IP Subnet เข้าไว้ด้วยกัน โดยที่ Subnet ทั้งหลายเหล่านั้นเชื่อมต่อกันด้วยเน็ตเวิร์กความเร็วสูง การกำหนดไซต์จะช่วยให้ผู้บริหารระบบสามารถกำหนด โทโปโลยี (Topology) ของการเรพลิเคชันระหว่าง เครื่องโดเมนคอนโทรลเลอร์ได้ และกำหนดกลุ่มของเครื่องโดเมนคอนโทรลเลอร์ที่ทำหน้าที่ตรวจสอบการเข้าใช้ระบบของผู้ใช้ที่อยู่ในพื้นที่เดียวกัน

5.3 Active directory Service Interfaces (ADSI)

เป็น API ใช้เข้าถึงข้อมูลในไดเรกทอรีเซอร์วิส สำหรับระบบเน็ตเวิร์คที่แตกต่างกัน ดังรูปที่ 5-6 ดังนั้นเราใช้ ADSI อย่างเดียวก็สามารถเข้าไปบริหารทรัพยากรในระบบเน็ตเวิร์คที่ต่างระบบการจัดการไม่ว่าเป็น NetWare Bindery, X.500, Notes, NDS และ NTS DS ผู้ดูแล หรือผู้พัฒนา สามารถใช้ ADSI ค้นหาหรือจัดการทรัพยากรในไดเรกทอรีเซอร์วิส และง่ายต่อการใช้นำไปใช้งาน



รูปที่ 5-6 การใช้ ADSI เข้าถึงข้อมูล

5.3.1 Active Directory Service Interfaces Architecture

ADSI ประกอบด้วย ADSI object และ Dependent object ผู้ใช้สามารถกระทำงานต่างๆ ผ่านทาง Interfaces ADSI สร้าง ADSI Object และหลาย Interface

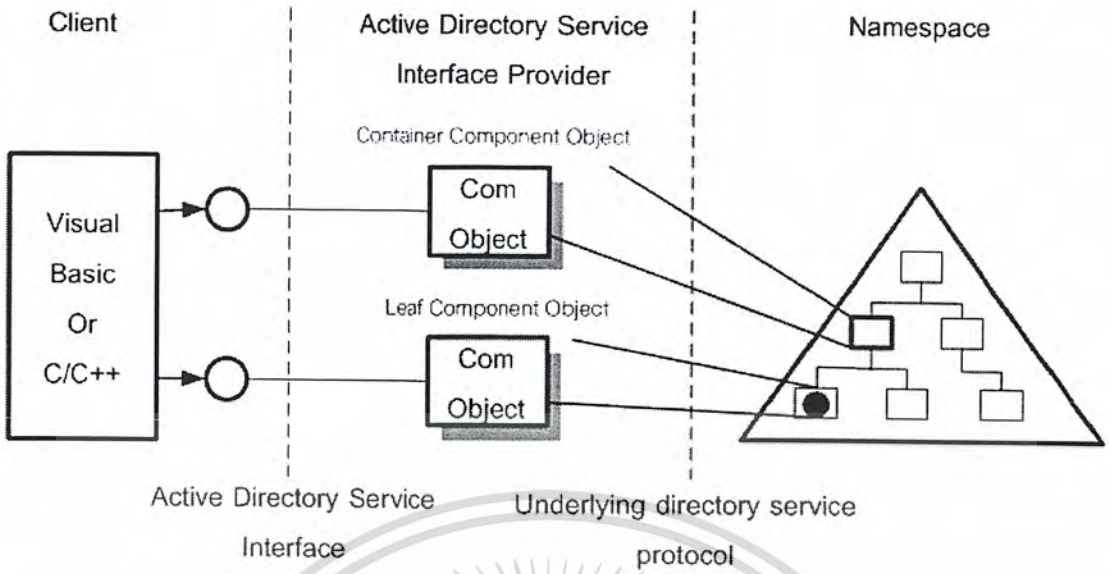
Active Directory Service Interfaces Object

เป็น COM ที่ทำงานอยู่ภายใต้ไดเรกทอรีเซอร์วิสสามารถใช้งาน ADSI ผ่านทาง COM ADSI object แบ่งออกเป็น 2 กลุ่มดังนี้

1. directory service leaf objects สามารถบรรจุ ADSI object อื่นได้
2. directory service container objects ไม่สามารถบรรจุ ADSI object อื่นได้

5.3.2 Active Directory Service Interfaces Provider

ADSI สร้างมาให้บรรจุ ADSI object และ dependent objects ทำหรับ Namespace โดยเฉพาะ จากรูปที่ 5-7 แสดงถึงผู้ใช้เกี่ยวกับการค้นหาหรือใช้ข้อมูลผ่านทาง Interfaces เท่านั้น โดยไม่ต้องรู้รายละเอียดภายใน



รูปที่ 5-7 โครงสร้าง Provider

5.3.3 Active Directory Service Interfaces Schema Management

ADSI สร้างให้สามารถเข้าถึงnamespaces ที่แตกต่างกันได้ด้วยการเขียนที่เหมือนกัน อย่างไรก็ตามในการเข้าถึงไดเรกทอรีอาจจะใช้หลายฟังค์ชั่นขึ้นอยู่กันรายละเอียดเฉพาะของADSI ได้เรกทรอรีอาจจะไม่ได้ประกาศวัตถุที่บรรจุเหมือน ADSI อื่นๆ ในการเพิ่ม ต่อเติมข้อมูลในไดเรกทอรีเซอร์วิสอนุญาตให้แก้ไขได้เฉพาะ Schema พื้นฐาน และสามารถเพิ่มเติมได้ตามอำเภอใจของผู้ดูแลและอิสระจากผู้ขายเราสามารถขยายวัตถุด้วย Schema Management Active Directory Service Interfaces objects ใช้ได้ดังนี้

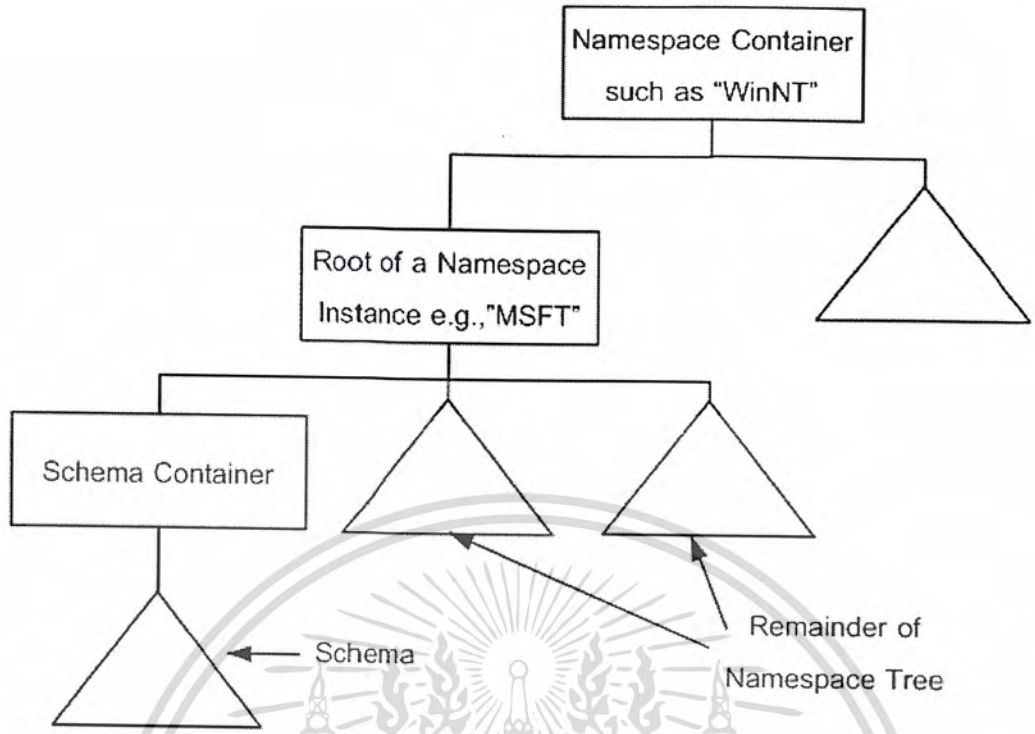
- เรียกดูข้อมูลวัตถุที่ประกาศไว้แล้ว
- ขยายข้อมูลวัตถุที่ประกาศไว้

5.3.4 Schema Management Active Directory Service Interfaces Objects

สามารถใช้ค้นหาและแก้ไขข้อมูลของ Schema ของ Namespace ประกอบด้วยรายละเอียดดังนี้

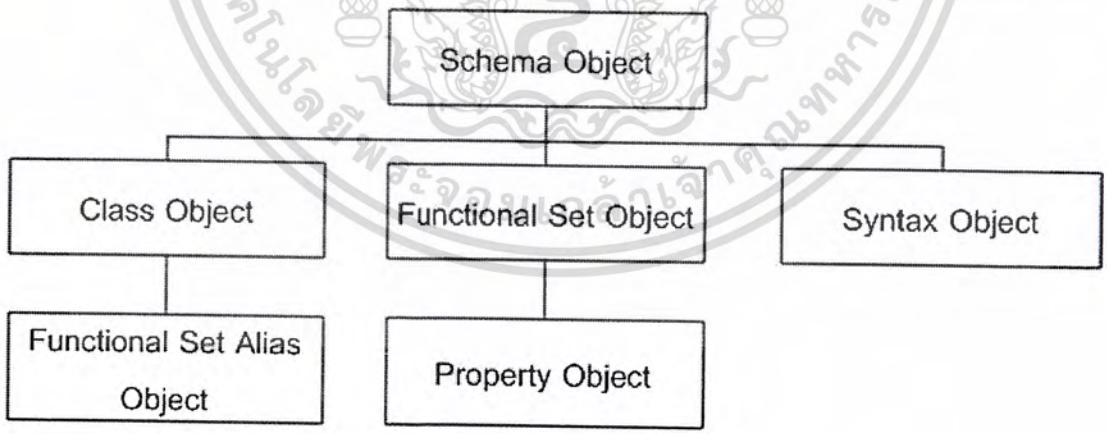
- Schema container object

ใช้รวมวัตถุที่ประกาศไว้ในส่วนหนึ่งของ ไดเรกทอรี โดยทั่วไปทุกวัตถุจะต้องมี schema เป็นของตัวเอง ADSI จะแสดงโดยที่การวาง schema container บนลูกของ directory root



รูปที่ 5-8 Schema containers

รูปที่ 5-8 แสดงถึงที่อยู่โดยทั่วไป อย่างไรก็ตาม ADSI ไม่ได้จำกัดว่าจะอยู่ตรงไหนของไดเรกทรอรี ความซับซ้อนของไดเรกทอรีอาจะอนุญาตให้มี schema ออกเป็นหลายส่วนดังรูปที่ 4

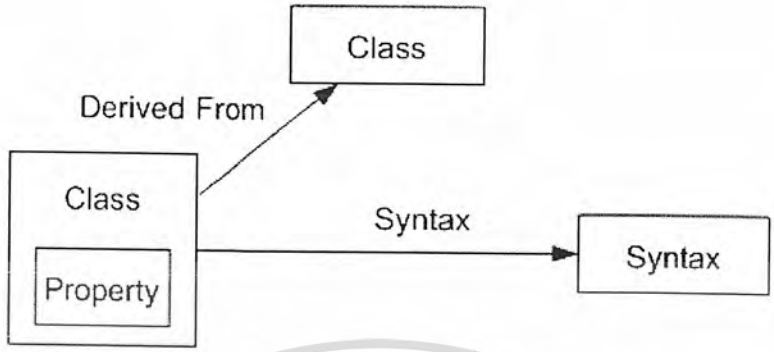


รูปที่ 5-9 Schema hierarchy

schema container ตัวมันเองจะประกอบด้วย class, functional set, property, และ syntax ดังรูปที่ 5-9

- Class container object

ใช้ในการประกาศ Class ที่สร้างใน ไดเรกทอรี Classใหม่สามารถถอดคุณสมบัติจากClass อื่นที่ใช้อยู่ในไดเรกทอรี



รูปที่ 5-10 Creating a class

รูปที่ 5-10 แสดงให้เห็นถึงความสัมพันธ์ของ คลาสอื่น คุณสมบัติ และการสร้างเป็นคลาสใหม่

5.3.5 Active Directory Service Interfaces Caching

ทุก วัตถุ ADSI จะมีสอง methods คือ GetInfo และ SetInfo เพื่อให้ง่ายต่อการเข้าถึงข้อมูลคุณสมบัติ การทำการอ่าน หรือกำหนดค่าคุณสมบัติจะทำงานอยู่ใน cache

เราสามารถเรียกใช้คุณสมบัติจาก วัตถุ ADSI ได้ทุกเวลาหลังจากที่มีการเข้าถึงวัตถุแล้ว ในการเรียกเราต้องเรียก GetInfo ก่อน ถ้าคุณสมบัติไม่ได้มีการถามมาก่อนจะอ่านข้อมูลมาให้และเก็บไว้ใน Cached และเมื่อมีการถามอีกจะอ่านจาก Cached

- GetInfo จะเป็นการเรียกให้มีการอ่านข้อมูลที่แน่นอนออกมา
- SetInfo จะเป็นการเขียนข้อมูลลงไดเรกทรอรี

5.3.6 Active Directory Service Interfaces Names

ชื่อที่ใช้เรียกวัตถุในไดเรกทอรีจะมีเพียงชื่อเดียว ซึ่งจริงๆแล้วจะมีการจัดลำดับตามไดเรกทอรีเพื่อให้หาได้ง่าย ดังตัวอย่าง

CN=JSmith, OU=Sales, DC=ArcadiaBay, DC=Com

จุดหมายในการลดขนาดของความรู้ในโค้ดที่ต้องมีสำหรับที่อยู่ของวัตถุ ซึ่งเราจะเรียกที่อยู่ของวัตถุงว่า ADsPath ทำให้เกิดการเรียกที่อยู่ของวัตถุได้หลายแบบ ดังนี้

WinNT://REDMOND/jsmith

WinNT://REDMOND/comp1, computer

WinNT://REDMOND/comp1/alice

LDAP://OU=Sales, DC=ArcadiaBay, DC=COM

LDAP://exch01/O=Microsoft

5.3.7 Active Directory Service Interfaces Navigation

ADSI ถูกสร้างให้ง่ายโดนใช้ผ่าน IADsContainer การใช้interface สามารถเรียกใช้เฉพาะวัตถุที่สนใจ

ตัวอย่างการเข้าถึง organizational unit

```
Set ou = GetObject("LDAP://host1/OU=Sales, DC=ArcadiayBay,DC=COM")
For each obj in ou
  Debug.Print  obj.Name
Next
```

ตัวอย่าง แสดงรายชื่อผู้ใช้ใน ArcadiaBay โดเมน

```
Set dom = GetObject("WinNT://ArcadiaBay")
dom.Filter = Array("user")
For each usr in dom
  Debug.Print usr.Name
Next
```

5.3.8 Active Directory Service Interfaces Searching

การค้นหาข้อมูลเป็นอย่างหนึ่งที่มีการใช้งานจากผู้ใช้บ่อยในหลายไดเรกทอรีใน ADSI สามารถค้นหาโดยใช้ผ่านทาง OLE DB interfaces หรือ ADSI ทำหรับการควบคุมที่ดีในการค้นหาจะต้องมีการกำหนดในส่วนของรายละเอียดอื่นเช่น search page size, sort, size limit, search level, search scope และ other options

ตัวอย่าง การอ่านค่าจาก Active Directory โดนใช้ ADO

```
Dim con As New Connection, rs As New Recordset
Dim Com As New Command
con.Provider = "ADsDSOObject"
con.Open "Active Directory Provider"
Set Com.ActiveConnection = con
Com.CommandText = "select name from 'LDAP://DC=ArcadiayBay,DC=COM' where
objectClass='*' ORDER BY NAME"
Com.Properties("Page Size") = 1000
```

```
Com.Properties("Timeout") = 30 'seconds
Com.Properties("searchscope") = ADS_SCOPE_SUBTREE
Set rs = Com.Execute
While Not rs.EOF
  Debug.Print rs.Fields("Name").Value
  rs.MoveNext
Wend
```

5.3.9 Active Directory Service Interfaces Security

ADSI ถูกสร้างมาให้สามารถใช้ได้ทั้งการพิสูทธ์ตนและพิสุทะสิทธิของที่ผู้ใช้ที่เข้ามาใช้ ถ้าผู้ใช้เข้ามาใช้งาน Active Directory ผู้ใช้อาจให้การพิสูทด้วย Kerberos หรือ NTLM ขึ้นอยู่กับโครงสร้างที่เหมาะสม

ตัวอย่าง JSmith เข้ามาใช้งานระบบ

```
Dim dso As IADsOpenDSObject
Dim domain As IADsDomain
Set dso = GetObject("WinNT:")
Set domain = dso.OpenDSObject("WinNT://ArcadiaBay", "JSmith", "secret",
ADS_SECURE_AUTHENTICATION)
```

ตัวอย่าง การเพิ่ม ACE ด้วยสิทธิของวัตถุ

```
Dim Ace1 as new IADsAccessControlEntry
Dim Ace2 As new IADsAccessControlEntry
Dim Dacl as new IADsAccessControlList
Dacl.AclRevision = 4 'DS ACL Revision
Ace1.AccessMask = -1 'Full Permission (Allowed)
Ace1.AceType = ADS_ACETYPE_ACCESS_ALLOWED
Ace1.AceFlags = ADS_ACEFLAG_INHERIT_ACE
Ace1.Trustee = "ACTIVED\Administrator"
Dacl.AddAce Ace1
Dacl.AddAce Ace2
```

บทที่ 6
การออกแบบ โครงสร้าง และการทำงานของระบบ

6.1 แนวคิด

ไฟร์วอลในปัจจุบันมีหลากหลายชนิด และความสามารถแตกต่างกัน ร่วมถึงการติดตั้งไฟร์วอลในหลายที ทำให้การควบคุมทำได้อยาก เพราะต้องตามไปกำหนดกฎในทุกที่ติดตั้งไฟล์ยิ่งมีการติดตั้งมากหนึ่งจุดก็จะยิ่งมีความยุ่งยากมากยิ่งขึ้น ยิ่งในปัจจุบันมีการนิยมใช้งานเพอร์ซันนอลไฟร์วอลมากยิ่งขึ้น จึงมีความคิดที่จะมีการเพิ่มความสามารถของการกำหนดกฎการป้องกันจากส่วนกลางเพียงจุดเดียวทำให้ง่ายต่อการควบคุม และ จัดการ

ทั้งในปัจจุบันมีการนำไดเรกทอรีเซอร์วิสมาใช้ในองค์กรมากยิ่งขึ้นเพราะมีความง่ายในการควบคุม จัดการ และเก็บข้อมูลของผู้ใช้ อีกทั้งยังมีการพิสูจน์ตน (Authentication) ก่อนที่จะเข้าใช้งานทรัพยากรของระบบ และมีมาตรการ การรักษาความปลอดภัยข้อมูลของระบบที่มีประสิทธิภาพ จึงได้นำคุณสมบัติของ แอ็คทีฟไดเร็คทอรี่ในวินโดวส์ 2000 มาเพิ่มความสามารถเพื่อใช้ในการควบคุมกฎของไฟร์วอล และ ใช้ในการเก็บล็อกลงในฐานข้อมูล จากเครื่องลูกข่ายที่อยู่ในระบบปฏิบัติการวินโดว์2000 ทำให้สามารถควบคุมการทำงานของไฟร์วอลได้ที่จุดศูนย์กลางเพียงจุดเดียว

ซึ่งหากมีการโจมตีเกิดขึ้นนั้นสามารถเก็บผลจากการโจมตีได้ เพื่อรวบรวมเข้าด้วยกันให้สามารถนำมาวิเคราะห์กฎที่ จะเพิ่มเข้าไปได้อย่างเหมาะสมทำให้ระบบ มีความปลอดภัยยิ่งขึ้น

จากที่ได้กล่าวมาข้างต้นทำให้เกิดแนวความคิดสร้างไฟร์วอลที่มีความสามารถดังนี้

- ไฟร์วอลสามารถป้องกันการโจมตีได้ตามกฎ พร้อมทั้งมีระบบตรวจจับผู้บุกรุกทำงานร่วมด้วย
- สามารถกำหนดกฎของไฟร์วอลได้จากส่วนกลาง
- สามารถจัดเก็บการรายละเอียดการโจมตีไว้ที่ฐานข้อมูลกลางได้

สามารถแบ่งโปรแกรมการทำงานออกเป็น 3 ส่วนดังนี้คือ

- Personal Firewall

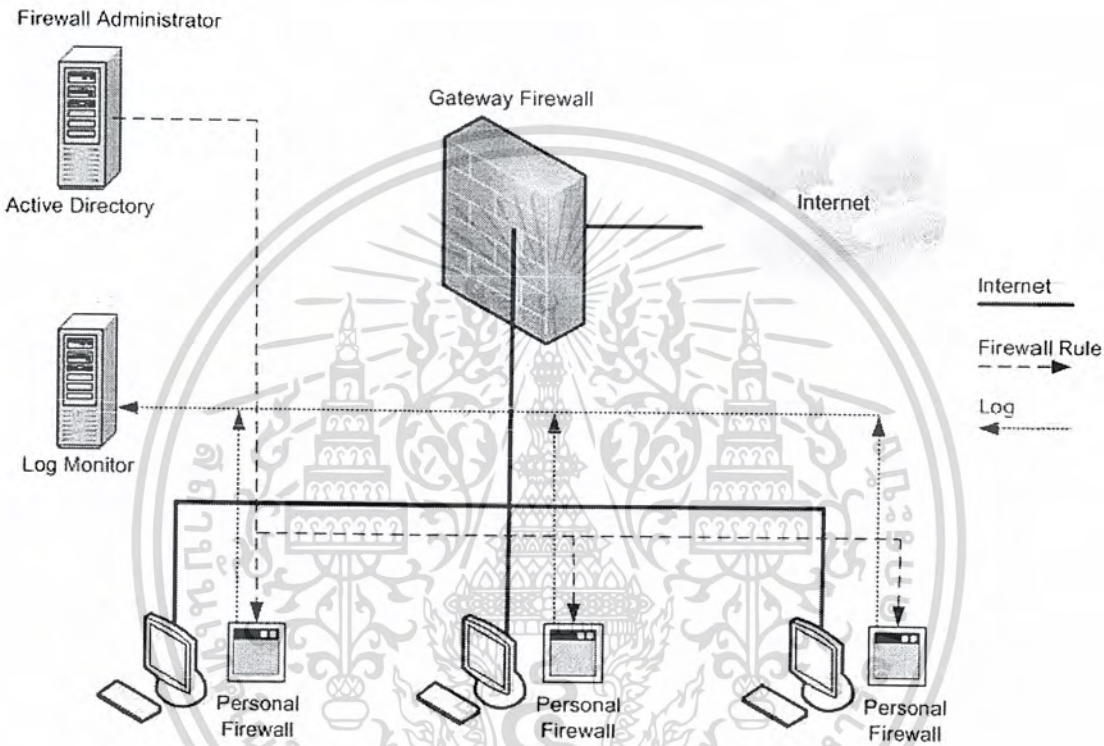
  ออกแบบให้ทำหน้าที่ป้องกันการโจมตีได้ตามกฎที่ได้รับจากเครื่องแม่ข่ายได้อย่างถูกต้อง และเมื่อมีการบุกรุกจะส่งข้อมูลการบุกรุกให้เครื่องล็อกมอนิเตอร์
- Firewall Administrator

  ออกแบบให้ทำหน้าที่กำหนดกฎการป้องกันในแต่ละกลุ่มผู้ใช้ เพื่อให้ไฟร์วอลในระบบได้รับกฎป้องกันการบุกรุก
- Log Monitor

  ออกแบบให้รับรายละเอียดการโจมตีจากเครื่องลูกข่ายที่อยู่ในระบบ และจัดการเกี่ยวกับการนำล็อกไฟล์จากฐานข้อมูลขึ้นมา แสดงผลได้

6.2 โครงสร้างระบบ

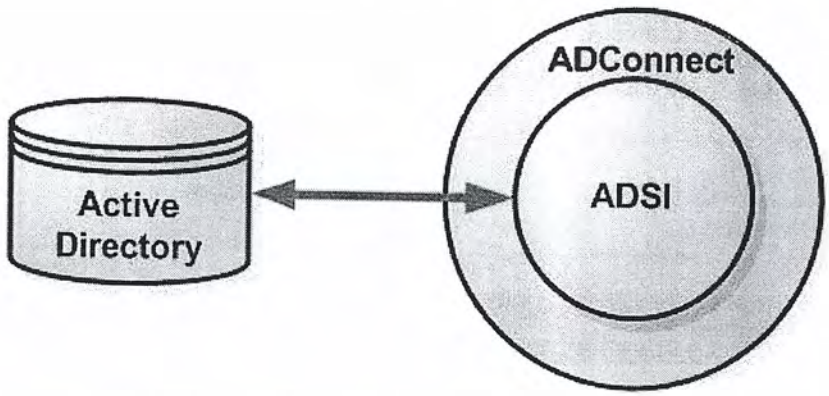
การออกแบบการป้องกันการโจมตีต้องมีการออกแบบการวางระบบให้มีจุดที่เหมาะสม และมีโครงสร้างระบบดังรูปที่ 6-1 โดยนำไฟร์วอลที่ได้ออกแบบมาทำการวางอยู่ในโครงสร้างโดยที่ Firewall Administration กับ Log Monitor จะอยู่ในส่วนของ Server หรือขอบเขตที่ต้องการกันไว้เป็นส่วนที่เข้าถึงได้อยากให้เข้าถึงได้เฉพาะภายในระบบ และส่วนของ Personal Firewall จะติดตั้งในส่วนต่างๆ ของระบบ ส่วน Gateway Firewall จะใช้ความสามารถเหมือนกัน Personal Firewall



รูปที่ 6-1 โครงสร้างและส่วนประกอบหลักของระบบ

6.3 ส่วนประกอบ และ หลักการทำงาน

การติดต่อทุกอย่างในระบบไฟร์วอลจะทำงานอยู่บนความสามารถ Active directory ของ Windows 2000 โดยข้อมูลจะเก็บอยู่ใน Active Directory Database และเข้าถึงข้อมูลผ่าน ADSI ไปยัง Active Directory Service เพื่อเข้าถึงข้อมูลที่จัดเก็บ แสดงดังรูปที่ 6-2



รูปที่ 6-2 แสดงรูปแบบการเข้าถึง Active Directory Service

6.3.1 Schema

โครงสร้างของ Active Directory Database ประกอบไปด้วย คอนเทนเนอร์ คลาส และ แอตตริบิวต์ โดยการเข้าถึงข้อมูลเหล่านั้นจะมีการกำหนดหน้าที่การทำงานมาแล้ว ดังนั้นถ้าเราต้องการนำข้อมูลที่เราต้องการจัดเก็บหรือเข้าถึงรูปแบบใหม่เข้าไปจึงต้องมีการสร้างขึ้นมาใหม่ เพื่อให้เหมาะสมกับการทำงานของระบบ ไดเรกทรอรีเบสไฟร์วอล์

ข้อมูลที่จัดเก็บประกอบด้วย ข้อมูลของกฎและข้อมูลรายละเอียดการโจมตีโดยเพิ่ม Schema ที่เหมาะสมลงไปใน Active Directory Database ดังนี้

1. แอตตริบิวต์ รายละเอียดการเพิ่มจะเป็นดังตาราง ที่ 6-1

Common Name	OID	Syntax
firewallRule	1.2.840.113556.1.4.7000.142	Case Insensitive String
firewallLog	1.2.840.113556.1.4.7000.144	Case Insensitive String

ตารางที่ 6-1 รายละเอียด Attribute ที่เพิ่มใน Active Directory Database

2. คลาส รายละเอียดการเพิ่มจะเป็นดังตาราง ที่ 6-2

Common Name	OID	Syntax
ISAGFW	1.2.840.113556.1.4.7000.143	Auxiliary

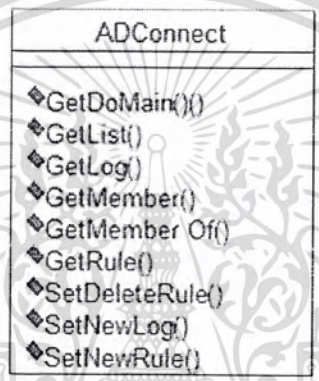
ตารางที่ 6-2 รายละเอียด Class ที่เพิ่มใน Active Directory Database

โดยกำหนดให้ Attribute firewallRule เป็น Mandatory และ FirewallLog จะเป็น Auxiliary ใน class ISAGFW ดังรูปที่ 6-3

3. ผนวก Schema ใหม่ เข้ากับ คลาส groups รายละเอียดดังรูปที่ 6-4

6.3.2 ADConnect Class

ในการเข้าถึงข้อมูลที่อยู่ใน Active Directory Service นั้นต้องใช้ ADSI ดังนั้นจึงพัฒนา function ที่ใช้เข้าถึงและกำหนดการทำงานได้อย่างง่ายได้ขึ้นมา โดยจะร่วมอยู่ใน Class ADConnect จะมีรายละเอียด Function ดังรูปที่ 6-3 โดยClass ADConnect จะใช้ในทุกโปรแกรมที่พัฒนาเพื่อติอต่อ Active Directory Service



รูปที่ 6-3 รายละเอียดของคลาส ADConnect

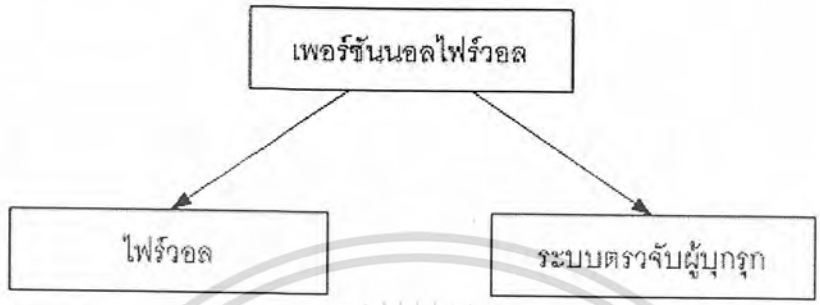
Function	การทำงาน
GetDomain	หาชื่อเครื่องที่ผู้ใช้สังกัดอยู่
GetList	อ่านรายละเอียดที่เป็นจุดอ้างอิง
GetLog	อ่านรายละเอียด Log
GetMember	อ่านรายชื่อ User จาก Group
GetMemberOf	อ่านรายชื่อ Group จาก User
GetRule	อ่าน กฎ ที่มีอยู่
SetDeleteRule	ลบ กฎ ที่ต้องการลบ
SetNewLog	สร้าง Log
SetNewRule	สร้างกฎ

ตารางที่ 6-3 รายละเอียดการทำงานของฟังค์ชั่นใน ADConnect

6.3.3 เพอร์ซันนอลไฟร์วอล (Personal Firewall)

เพอร์ซันนอลไฟร์วอลจะแบ่งการทำงานออกเป็น 2 ส่วน คือ

1. ไฟร์วอล
2. ระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

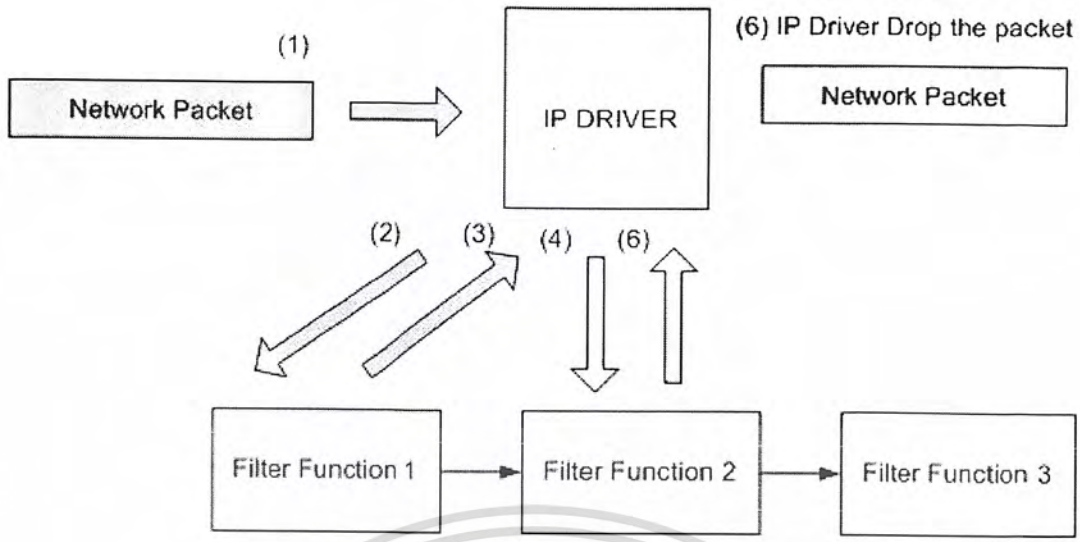


รูปที่ 6-4 แสดงส่วนประกอบเพอร์ซันนอลไฟร์วอล

6.3.3.1 แพ็กเก็ตฟิลเตอร์ริ่งไฟร์วอลบนวินโดวส์ 2000

ไฟร์วอลที่พัฒนาเป็นแบบ แพ็กเก็ตฟิลเตอร์ริง (Packet Filtering) ซึ่งจะสามารถใช้ฟังก์ชันฟิลเตอร์เพียงทีละฟังก์ชันเท่านั้น ดังนั้นหากโปรแกรมหนึ่งเรียกใช้ไดร์เวอร์ไฟร์วอลอยู่โดยส่งฟังก์ชันฟิลเตอร์เข้าไป ไดร์เวอร์ไฟร์วอลจะไม่สามารถถูกใช้ได้อีกในขณะนั้น แต่เราสามารถแก้ปัญหานี้ได้ โดยการกำหนดลำดับของฟิลเตอร์ฟังก์ชัน แล้วให้ระบบเรียกฟังก์ชันฟิลเตอร์เข้าไปทำงานทีละฟังก์ชันตามลำดับ จนกว่าจะมีฟังก์ชันใดฟังก์ชันหนึ่งส่งค่ากลับเป็น "DROP PACKET" ถ้าฟังก์ชันทั้งหมดส่งค่ากลับเป็น "ALLOW PACKET" แพ็คเกจเหล่านั้น จะถูกอนุญาติให้ผ่านไปได้

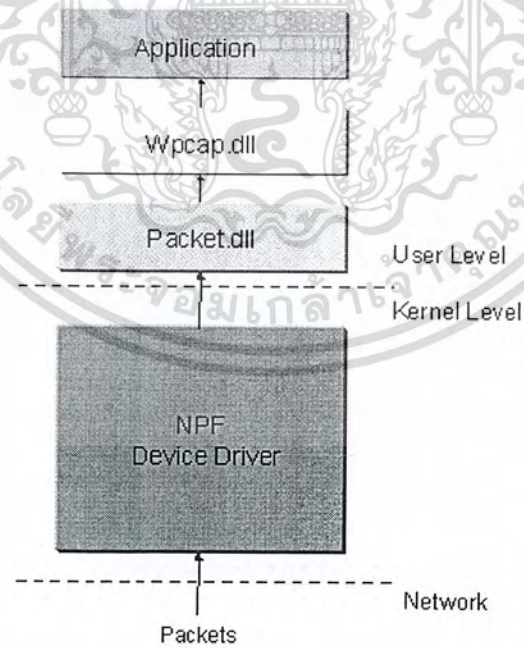
เมื่อเครื่องได้รับแพ็กเกตเข้ามา โดยที่ IP Driver มีฟังก์ชันฟิลเตอร์อยู่ตามที่กำหนดไว้ IP Driver จะส่งแพ็กเกจนั้นเข้าไปโดย ผ่านเข้าไปยังฟังก์ชันฟิลเตอร์ตามลำดับไปเรื่อยๆ โดยรอค่าที่ส่งกลับออกมาสมมุติฟังก์ชันแรกส่งค่ากลับเป็น "ALLOW PACKET" เมื่อ IP DRIVER ได้รับค่าส่งกลับจากฟังก์ชันแรกเป็น "ALLOW PACKET" ดังนั้น IP DRIVER จะส่งแพ็กเกจนั้นไปที่ฟังก์ชันที่สองต่อไป ในกรณีนี้สมมุติให้ฟังก์ชันที่สองนี้ส่งค่ากลับเป็น "DROP PACKET" เมื่อ IP DRIVER ได้รับค่าส่งกลับจากฟังก์ชันที่สองเป็น "DROP PACKET" ดังนั้น IP DRIVER จะไม่ส่งแพ็กเกจนี้ต่อไปยังระบบ และจะไม่ส่งไปยังฟังก์ชันต่อไปอีก



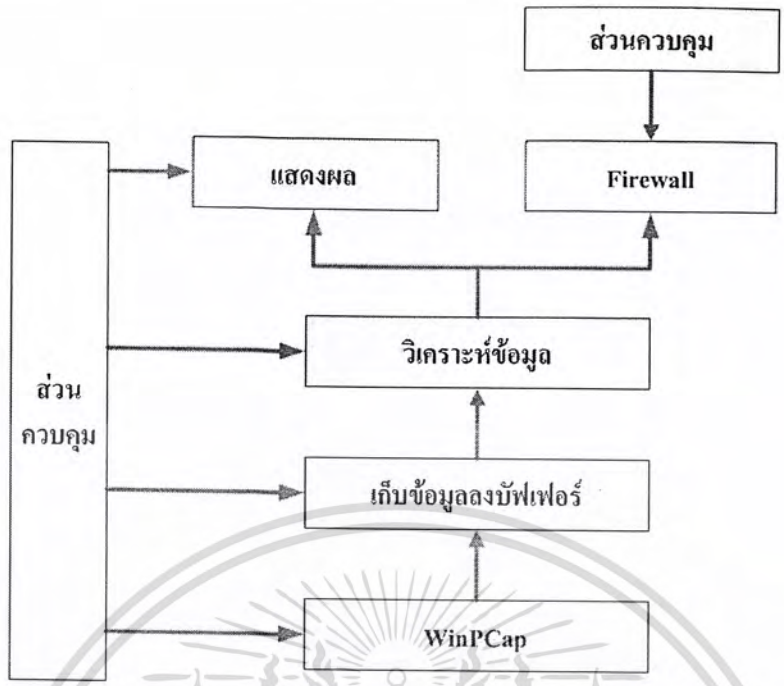
รูปที่ 6-5 แสดงขั้นตอนการทำงานของไฟร์วอล

6.3.3.2 ระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์

ระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ จะอาศัยคุณสมบัติ และ ความสามารถของ WinPCap ซึ่งเป็นไลบรารีที่ทำการติดต่อกับการ์ดแลน เพื่อทำการควบคุมการทำงานของ การ์ดแลนและตรวจจับแพ็กเก็ต เพื่อนำวิเคราะห์

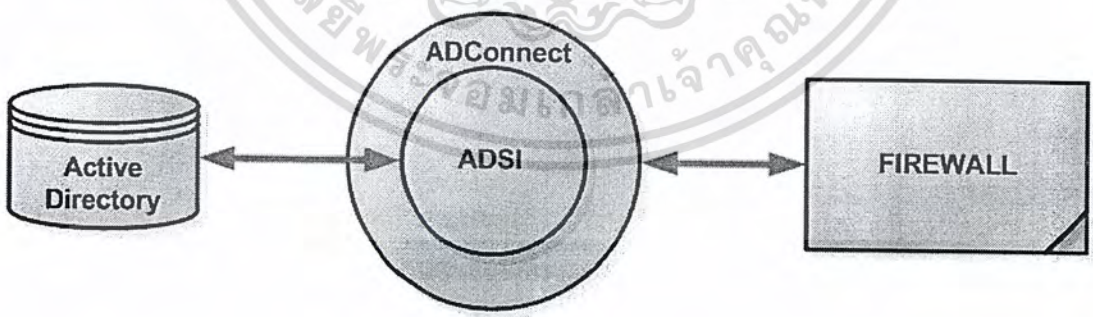


รูปที่ 6-6 แสดงระดับชั้นการทำงานของ WinPCap



รูปที่ 6-7 โครงสร้างของระบบตรวจจับผู้บุกรุกทำงานร่วมกับไฟร์วอล

เนื่องจากเพอร์ซันนอลไฟร์วอล (Personal Firewall) จะมีความสามารถในการที่จะรับกฎ เพื่อเริ่มการทำงานของไฟร์วอล จากเครื่องเซิร์ฟเวอร์ และ จะมีการส่งล็อกไฟร์ไปเก็บไว้ยัง ศูนย์กลางเมื่อมีการตรวจพบการโจมตี หรือ มีสิ่งผิดปกติเกิดขึ้น ซึ่งโครงสร้างการทำงานของส่วนที่ใช้ในการติดต่อกับเครื่องเซิร์ฟเวอร์ แสดงดังรูปที่ 6-6 โดย เพอร์ซันนอลไฟร์วอลจะมีการติดต่อ ไปยังเซิร์ฟเวอร์โดยผ่านคลาส (Class) ADConnect ที่สร้างขึ้น



รูปที่ 6-8 แสดงโครงสร้างส่วนติดต่อกับเซิร์ฟเวอร์

เพอร์ซันนอลไฟร์วอลจะติดตั้งอยู่ที่เครื่องลูกข่ายทุกเครื่องในระบบโดยมีหน้าที่การทำงานดังนี้

- เป็นไฟร์วอลชนิดแพ็กเก็ตฟิลเตอร์ริ่ง (Packet Filtering Firewall)

- ตรวจจับผู้บุกรุกทางเครือข่ายที่เข้ามาที่เครื่องลูกข่าย รวมทั้งสิ่งผิดปกติที่ไม่เป็นไปตามมาตรฐานของ โปรโตคอลทีซีพี (TCP) , ยูดีพี(UDP), ไอพี(IP), ไอซีเอ็มพี(ICMP)
- ส่งการแจ้งเตือนกลับไปยังล็อกมอนิเตอร์เมื่อตรวจจับได้ว่ามีการบุกรุกเกิดขึ้น

ขอบเขตและความสามารถ

ในระบบจะมีเครื่องลูกข่ายหลายเครื่องโดยทุกเครื่องจะมีการติดตั้ง เพอร์ซันนอลไฟร์วอล ซึ่งก็คือเอเจนต์ที่ฝังตัวทำงานอยู่แบบอัตโนมัติ โดยจะมีการควบคุมดูแลจากส่วนกลางโดยผู้ใช้จะไม่รับรู้ถึงการทำงานของเอเจนต์ รวมไปถึงผู้ดูแลระบบสามารถกำหนดกฎการป้องการบุกรุกให้ผู้ใช้ในแต่ละกลุ่มตามความเหมาะสม

อีกทั้งยังมีส่วนของการทำงานที่เป็นระบบตรวจจับผู้บุกรุกที่คอยทำหน้าที่ตรวจสอบว่ามีการบุกรุกเข้ามาที่เครื่องลูกข่ายนั้นๆหรือไม่ ถ้ามีจะมีการแจ้งเตือนไปยังเครื่องเซิร์ฟเวอร์กลางเพื่อนำผลที่ได้ไปวิเคราะห์หาแนวทางป้องกันต่อไป

6.3.4 ไฟร์วอลแอดมินนิสเทรเตอร์ (Firewall Administrator)

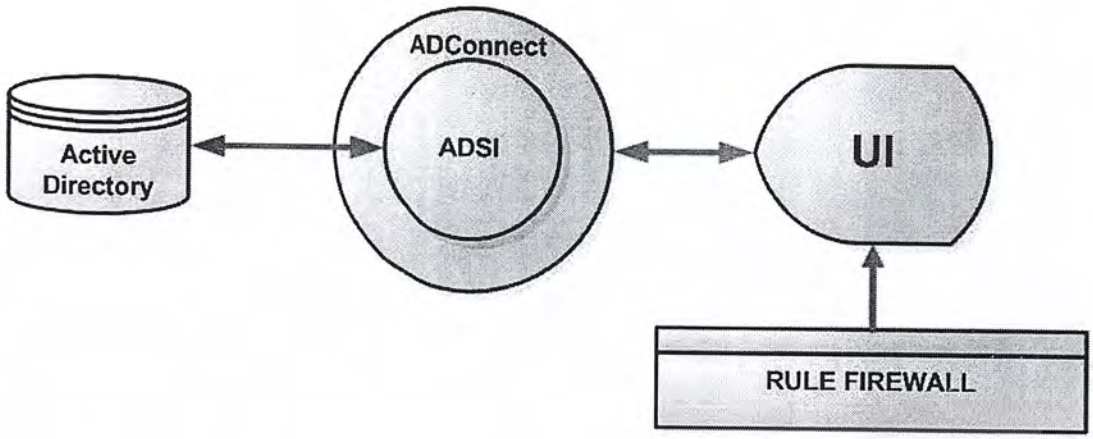
เป็นส่วนที่มีความสำคัญในระบบเพื่อกำหนดกฎของไฟร์วอล และทำงานกับ Active Directory Service เพื่อให้มีการเข้าถึงได้จากเครื่องลูกข่ายเพื่อนำกฎไปใช้ในการป้องกันการบุกรุกโดยข้อมูลกฎจะเก็บอยู่ใน Active Directory Database

ขอบเขตและความสามารถ

- สามารถพิสูจน์ผู้ใช้ที่มาขอใช้ทรัพยากรได้อย่างถูกต้องโดยใช้ความสามารถของแอ็คทีฟไดเร็คทอรี่
- สามารถกระจายกฎและจัดเก็บได้อย่างมีระเบียบ
- สามารถตรวจสอบได้ว่าเครื่องลูกข่ายยังติดต่ออยู่ในระบบหรือไม่

การทำงาน

เมื่อเปิดโปรแกรม Administration firewall จะทำการอ่านข้อมูลผ่าน ADConnect เพื่อเข้าถึงข้อมูลที่อยู่ใน Active Directory Service โดยจะเลือกแสดงตามกลุ่มผู้ใช้ที่ถูกเลือก และเมื่อมีการ เพิ่ม ลด และแก้ไข ข้อมูลกฎที่อยู่ในดาต้าเบส ดังรูปที่ 6-9 จะมีส่วนแสดงการทำงานหลักของโปรแกรมไฟร์วอลแอดมินิสเทรชั่น (Firewall Administration)



รูปที่ 6-9 แสดงโครงสร้างการทำงานของไฟร์วอลแอดมินิสเทรชั่น

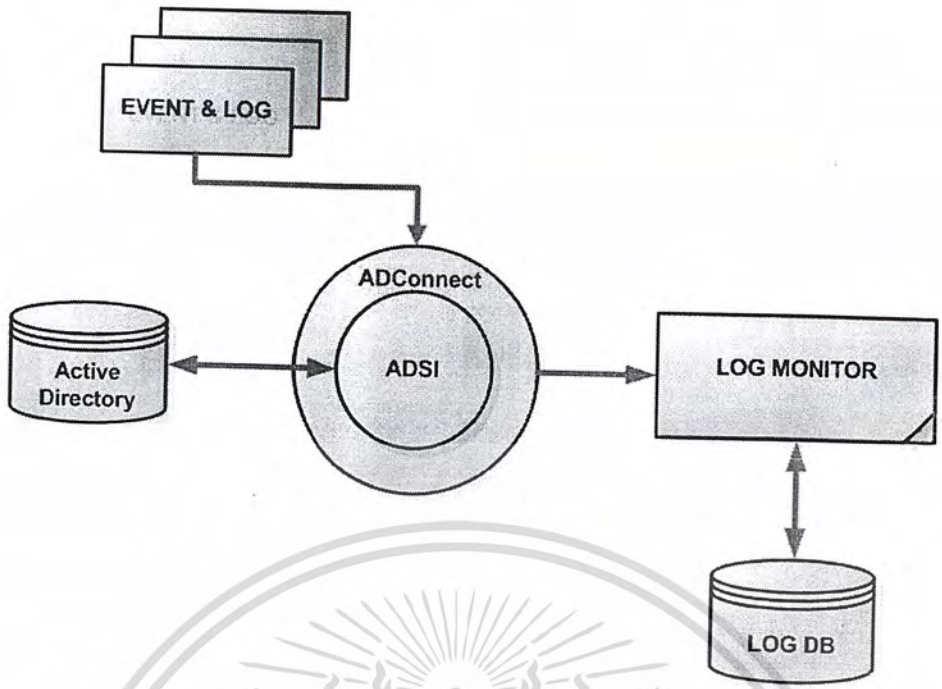
6.3.5 ล็อกมอนิเตอร์ (Log Monitor)

ในระบบการใช้จริงจะมีเครื่องลูกข่ายในระบบจำนวนมาก ซึ่งอาจจะมีการบุกรุกที่มากมายตามมาด้วย เมื่อเอเจนต์ที่อยู่ในเครื่องลูกข่ายตรวจพบการบุกรุก รวมทั้งการโจมตีจะแสดงรายละเอียด รวมทั้งจะส่งข้อมูลมาที่เครื่องล็อกมอนิเตอร์ เพื่อแสดงรายละเอียดและเก็บลงฐานข้อมูล เพื่อให้ผู้ดูแลระบบสามารถตรวจสอบและวิเคราะห์หาแนวทางการป้องกันที่เหมาะสมได้ต่อไป

ขอบเขตและความสามารถ

ล็อกมอนิเตอร์จะติดตั้งอยู่บนเครื่องที่เป็นโดเมนอื่นที่ไม่ใช้โดเมนหลัก หรือ เครื่องโดเมนหลักที่ติดตั้งไฟร์วอล Administration ทำหน้าที่รอรับรายละเอียดการบุกรุกจากเครื่องลูกข่าย
และแสดงผลรายละเอียดที่มีอยู่ในฐานข้อมูล

การทำงานติดต่อทุกอย่างในระบบไฟร์วอลจะทำงานอยู่บนความสามารถของบริการ Active directory ของ windows 2000 โดยข้อมูลจะเก็บอยู่ใน Active Directory Database และเข้าถึงข้อมูลผ่าน ADSI ไปยัง Active Directory Service เพื่อเข้าถึงข้อมูลที่จัดเก็บ ซึ่งการทำงานของล็อกมอนิเตอร์ที่ทำงานร่วมกันกับ แอ็คทีฟไดเร็คทอรี่ (Active Directory) แสดงโครงสร้างการทำงาน ดังรูปที่ 6-7



รูปที่ 6-10 แสดงโครงสร้างการทำงานระหว่างล็อกมอนิเตอร์กับแอ็คทีฟไดเรคทอรี่

โดยที่โครงสร้างของตารางที่ใช้ในการเก็บข้อมูลการโจมตีต่าง ๆ แสดงดังรูปที่ 6-8 ซึ่งมีรายละเอียดให้การเก็บข้อมูลดังนี้

- LogNo เป็นข้อมูลชนิด Number และ เป็นคีย์หลัก แสดงลำดับที่ของ ล็อก
- AttackDst เป็นข้อมูลชนิด Text ใช้เพื่อเก็บหมายเลข ไอพีปลายทาง
- AttackSrc เป็นข้อมูลชนิด Text ใช้เพื่อเก็บหมายเลขไอพีต้นทาง
- AttackType เป็นข้อมูลชนิด Text ใช้เพื่อเก็บชนิดหรือ ประเภทการโจมตี
- AttackDate เป็นข้อมูลชนิด Text ใช้เก็บวันที่มีการเก็บล็อกการโจมตีนั้น ๆ
- AttackTime เป็นข้อมูลชนิด Text ใช้เก็บเวลาที่มีการเก็บล็อกลงฐานข้อมูล
- User เป็นข้อมูลชนิด Text ใช้เพื่อเก็บชื่อ User ที่แจ้งล็อกมายังเซิร์ฟเวอร์
- Group เป็นข้อมูลชนิด Text ใช้เพื่อเก็บ Group ของผู้ใช้ที่แจ้งล็อกมายังเซิร์ฟเวอร์

ชื่อเขตข้อมูล	ชนิดข้อมูล	คำอธิบาย
LogNo (คีย์หลัก)	Number	หมายเลขลำดับที่ของล็อก
AttackDst	Text	หมายเลขไอพีปลายทาง
AttackSrc	Text	หมายเลขไอพีต้นทาง
AttackType	Text	ชนิดของการโจมตี
AttackDate	Text	วันที่
AttackTime	Text	เวลา
User	Text	ชื่อผู้ใช้
Group	Text	กลุ่มผู้ใช้

ตารางที่ 6-4 แสดงโครงสร้างตารางที่ใช้ในการเก็บล็อก

บทที่ 7
การทดสอบการทำงาน

7.1 ระบบที่ใช้ทดสอบ

ขั้นตอนในการทดสอบการทำงานของระบบชุดโปรแกรมไฟร์วอลพร้อมมีรายละเอียดของเครื่องที่ทำการทดสอบดังต่อไปนี้

เครื่องที่ติดตั้งโปรแกรม Isag Personal Firewall

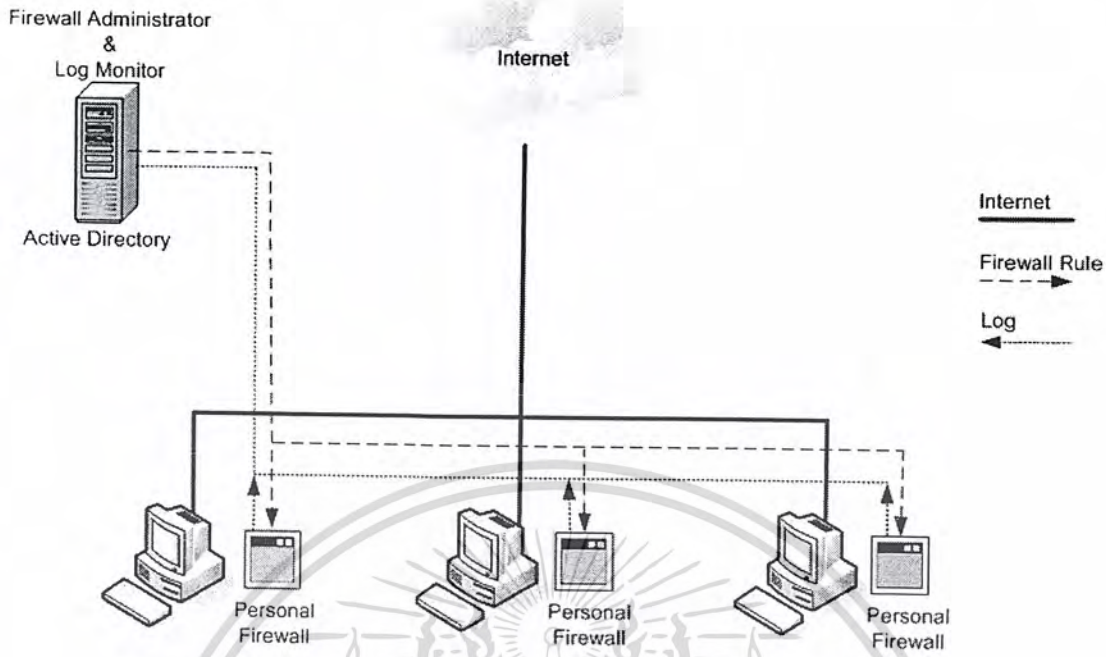
- หน่วยประมวณผลความเร็ว 1.8 GHz
- หน่วยความจำหลัก 512 MB
- ระบบปฏิบัติการไมโครซอฟท์วินโดวส์ 2000 Professional (SP4)
- การ์ดแลนความเร็ว 100 Mb

เครื่องที่ติดตั้งโปรแกรม Isag Firewall Administrator และ Isag Log Monitor

- หน่วยประมวณผลความเร็ว 1.5 GHz
- หน่วยความจำหลัก 512 MB
- ระบบปฏิบัติการไมโครซอฟท์วินโดวส์ 2000 Server (SP4)
- การ์ดแลนความเร็ว 100Mb

7.2 โครงสร้างของระบบที่ใช้ในการทดสอบ

โครงสร้างของระบบที่สามารถทำงานได้เต็มประสิทธิภาพนั้น ควรจะติดตั้งให้ครบทั้งส่วนไฟร์วอลแอดมินิสเตชัน, ส่วนเพอร์ซันนอลไฟร์วอลพร้อมระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ ในการทดสอบระบบนี้ ได้ติดตั้งโปรแกรมครบตามโครงสร้างทุกส่วนที่กล่าวมาแล้ว

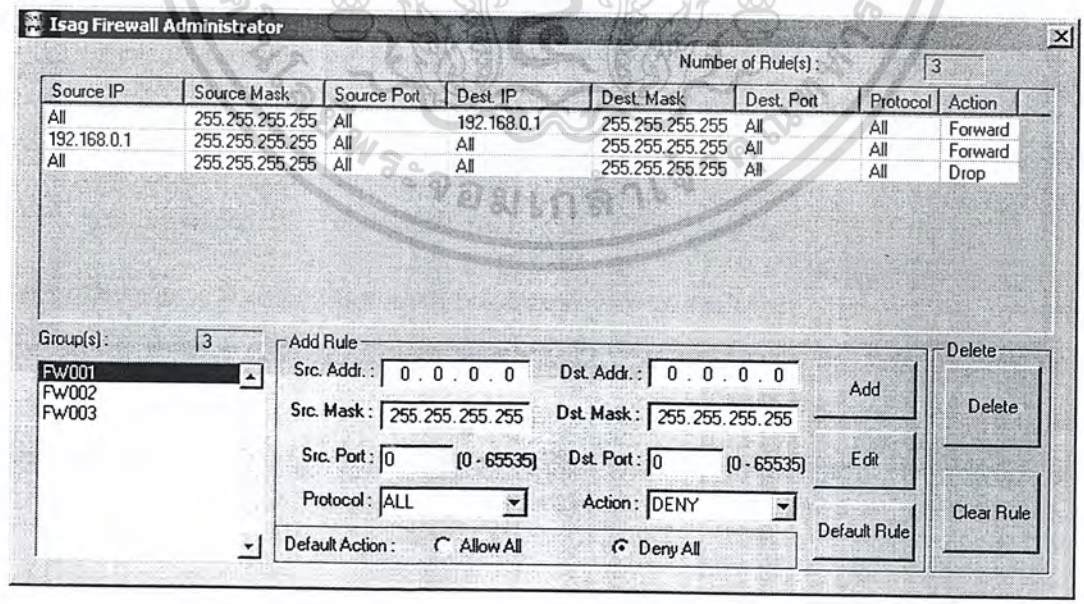


รูปที่ 7-1 โครงสร้างทางเครือข่ายของระบบที่ใช้ในการทดสอบ

7.2.1 ทดสอบการทำงานของไฟร์วอลแอดมินิสเทรเตอร์ (Firewall Administrator)

- การเริ่มต้นการทำงานของไฟร์วอลแอดมินิสเทรเตอร์ (Firewall Administrator)

เมื่อโปรแกรมไฟร์วอลแอดมินิสเทรเตอร์เริ่มต้นการทำงานแล้ว จะมีลักษณะดังต่อไปนี้



รูปที่ 7-2 โปรแกรมไฟร์วอลแอดมินิสเทรเตอร์

โปรแกรมจะแบ่งออกเป็นส่วนๆ ดังนี้

- ส่วนเลือกกลุ่มเพื่อกำหนดกฎการฟิลเตอร์ให้กับกลุ่มที่เลือก
- ส่วนแสดงกฎการฟิลเตอร์ของแต่ละกลุ่ม
- ส่วนเพิ่ม ลบ และแก้ไขกฎการฟิลเตอร์

การทำงานจะเริ่มที่กำหนดกฎการฟิลเตอร์ที่ต้องการให้กับกลุ่มนั้นๆ ที่ต้องการ เช่น สมมุติว่าต้องการให้ กลุ่ม FW001 ต้องการให้ใช้งานเพียงแค่ ไปยังเครื่องเซิร์ฟเวอร์ เราก็กำหนดกฎการฟิลเตอร์ดังรูป โดยกำหนดกฎการฟิลเตอร์พื้นฐานไว้ที่ Deny All คือไม่อนุญาตให้การติดต่ออื่นๆ ผ่านได้

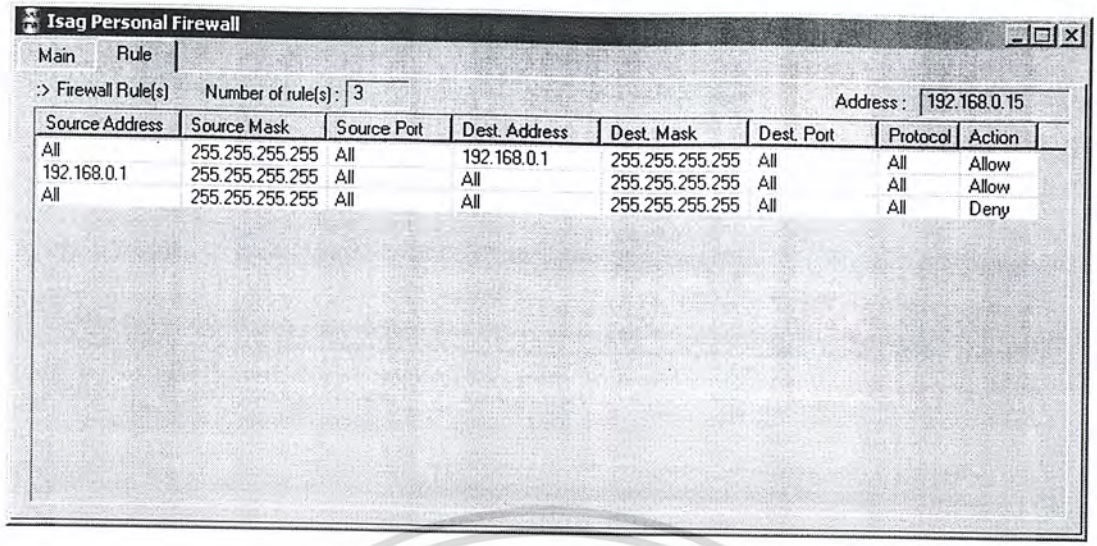
7.2.2 ทดสอบการทำงานของเพอร์ซันนอลไฟร์วอล (Personal Firewall)

- เริ่มต้นการทำงานของเพอร์ซันนอลไฟร์วอล

เมื่อโปรแกรมเพอร์ซันนอลไฟร์วอลเริ่มทำงานแล้ว จะมีลักษณะดังต่อไปนี้

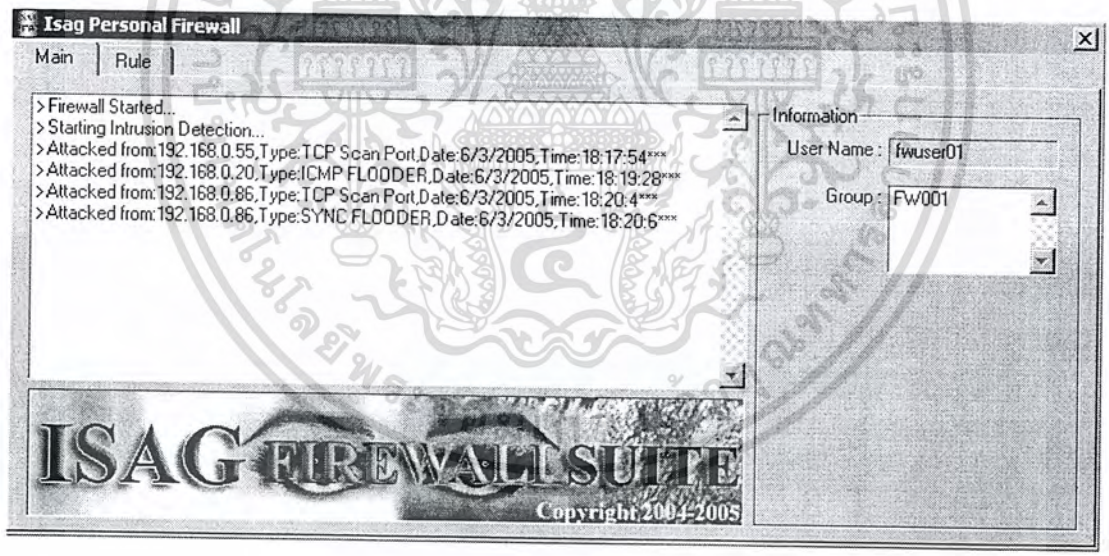
รูปที่ 7-3 โปรแกรมเพอร์ซันนอลไฟร์วอล

เมื่อโปรแกรมเริ่มทำงาน จะทำการรับเอากฎการฟิลเตอร์ของกลุ่มของผู้ล็อคอิน (Log in) ที่กำหนดไว้ที่ส่วนโปรแกรมไฟร์วอลแอดมินิสเทรเตอร์มาเป็นกฎการฟิลเตอร์ของตัวโปรแกรมเพอร์ซันนอลไฟร์วอลก่อนจะเริ่มทำงานด้วยการฟิลเตอร์ต่างๆ ตามกฎการฟิลเตอร์ที่รับมาจากส่วนกลาง และจะเริ่มการทำงานของส่วนตรวจจับผู้บุกรุกด้วยทันที โดยกฎที่รับมาจะแสดงดังรูป



รูปที่ 7-4 แสดงกฎการฟิลเตอร์ที่รับมาจากเซิร์ฟเวอร์

และเมื่อมีการโจมตีหรือมีการบุกรุกทางเครือข่ายคอมพิวเตอร์เกิดขึ้น โปรแกรมก็จะทำการส่งรายละเอียดต่างๆ ของการโจมตีหรือการบุกรุกนั้นๆ ไปยังเก็บไว้ที่เซิร์ฟเวอร์ ดังรูป

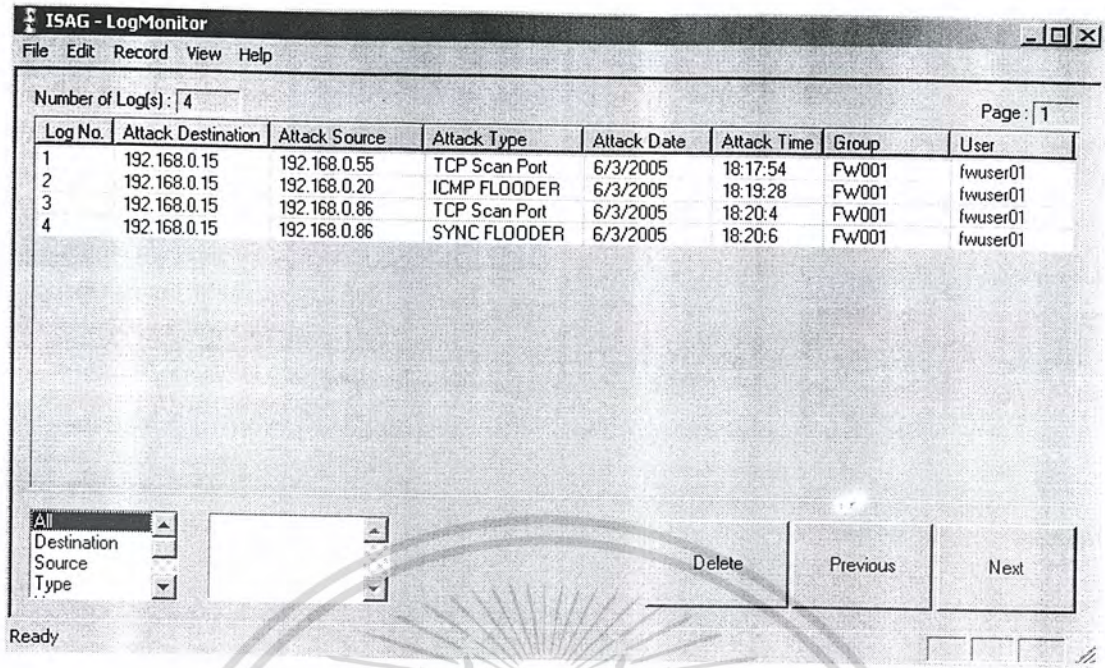


รูปที่ 7-5 การแจ้งเตือนการบุกรุกหรือถูกโจมตี

7.2.3 ทดสอบการทำงานของล็อกมอนิเตอร์ (Log Mornitor)

- เริ่มการทำงานของโปรแกรมล็อกมอนิเตอร์

เมื่อโปรแกรมล็อกมอนิเตอร์เริ่มทำงานแล้ว จะมีลักษณะดังต่อไปนี้



รูปที่ 7-6 โปรแกรมล็อกมอนิเตอร์

จะสามารถเลือกดูบันทึกข้อมูลจากเครื่องไคลเอนท์ (Client) ที่ส่งมาทั้งหมดได้ ทั้งแบบเลือกดูแต่ละประเภทได้ สามารถลบได้ เลื่อนหน้า ไปดูหน้าอื่นๆ ได้

บทที่ 8
สรุปผลและวิจารณ์

8.1 สรุปผลการทดสอบ

จากการพัฒนาโปรแกรมป้องกันการบุกรุกแบบไดเรกทรอรีเบส ทำให้สามารถ เข้าใช้งานแอ็คทีฟไดเร็คทอรี (Active Directory) เพื่อทำการ เพิ่ม ลด แก้ไข เปลี่ยนแปลง กฎการป้องกันการบุกรุกของไฟร์วอลที่อยู่ใน ไดเรกทอรีดาต้าเบสได้ ทำให้สามารถป้องกันการโจมตีได้ตามการกำหนดกฎการป้องกัน ที่มีอยู่ในไดเรกทอรีดาต้าเบส และสามารถนำข้อมูลการโจมตีที่ตรวจจับได้จากเครื่องลูกข่ายมาเก็บลงฐานข้อมูลโดยล็อกมินิเตอร์ โดยสามารถนำข้อมูลที่ได้มาแสดงผลและวิเคราะห์เพื่อกำหนดกฎการป้องกันได้ตามการโจมตีที่เกิดขึ้นได้จริง

แต่การทำงานที่อยู่ในระบบวินโดว์2000 จะมีปัญหาพัฒนาเนื่องการโปรแกรมป้องกันที่พัฒนาขึ้นมาทดลองนั้นต้องทำงานอยู่ในระดับเคอร์เนลโหมด (Kernel mode) จึงต้องมีการเขียนโปรแกรมแบบ System Service เพื่อให้สามารถทำงานได้กับผู้ใช้ทุกกลุ่มโดยไม่มีปัญหา

การพัฒนาในขณะนี้ จะมุ่งเน้นไปที่การเข้าถึงการทำงานของไดเรกทอรีเซอร์วิส เพื่อนำมาใช้งาน และการจัดการกฎการป้องกันการโจมตีที่เครื่องเพอร์ซันนอลไฟร์วอลเพื่อใช้ป้องกันการโจมตีในเครือข่าย

8.2 ปัญหาและอุปสรรคในการพัฒนา

- การศึกษาโครงงานเดิมที่มีอยู่แล้ว ซึ่งจำเป็นที่จะต้องศึกษาโดยละเอียดเพื่อที่จะให้สามารถนำมาพัฒนาต่อได้อย่างถูกต้อง และมีประสิทธิภาพ แต่เนื่องจากโครงงานเดิมนั้นมีข้อความอธิบายโปรแกรม (Comment) น้อยมากทำให้การศึกษาโค้ดโปรแกรม และ โครงสร้างการทำงานของโปรแกรมทำได้ยาก และ ช้าซึ่งจะต้องเสียเวลานาน
- ปัญหาการรวมโครงงานแต่ละส่วนเข้าด้วยกันเพื่อที่จะให้ทำงานร่วมกันได้อย่างถูกต้องนั้นจะต้องมีการทำความเข้าใจหลักการทำงาน ของโปรแกรมแต่ละส่วนเป็นอย่างดีเพื่อที่จะให้สามารถนำมารวมเข้าด้วยกันแล้ว มีประสิทธิภาพมากที่สุด
- ในการคอมไพล์ (Compile) โค้ดโปรแกรมในบางครั้งถ้าหาก คอมไพเลอร์ (Complier) ที่ใช้หรือ รุ่นของเซอร์วิสแพ็ค (Service Pack) ไม่ตรงกันอาจจะคอมไพล์โค้ดโปรแกรมแล้วเกิดข้อผิดพลาดเกิดขึ้น ดังนั้นจึงควรตรวจสอบรุ่น และ เซอร์วิสแพ็คของโค้ดโปรแกรมที่เขียนขึ้นว่าพัฒนามาจากเวอร์ชันใด

8.3 การพัฒนาต่อ

การพัฒนาต่อในขั้นต่อไปจะแบ่งออกเป็น 2 ส่วนดังนี้

8.2.1 การพัฒนาเพิ่มความสามารถพื้นฐาน

1. เพิ่มความสามารถในการทำงานโปรแกรมภายใต้สิทธิของผู้ใช้ได้ทุกกลุ่ม
2. การจัดการกฎการป้องกัน ให้มีความถูกต้องมากขึ้นเนื่องจากการเก็บข้อมูลของ ไดเรกทอรีจะไม่ค่อยแน่นอน

8.2.2 การพัฒนาเพิ่มความสามารถให้ใช้งานได้หลากหลายมากยิ่งขึ้น

1. การนำไดเรกทอรีมาใช้ในการเก็บข้อมูลได้มากขึ้น เช่น อ็อฟชันในการป้องกันการโจมตีที่มีการโจมตีบ่อยๆ เป็นต้น
2. พัฒนาโปรแกรมป้องกันการบุกรุกในรูปแบบอื่นๆ เช่น เกตเวย์ไฟร์วอล หรือโปรแกรมป้องกันเฉพาะด้าน เป็นต้น
3. ในการพัฒนาระบบตรวจจับผู้บุกรุกทางเครือข่ายคอมพิวเตอร์ควรใช้ระบบตรวจจับผู้บุกรุกที่เป็นแบบเปิดเผยโค้ด (Open source) ที่มีการปรับปรุงและพัฒนารูปแบบการในการตรวจจับผู้บุกรุกตลอดเวลา เพื่อที่จะให้สามารถตรวจจับการโจมตีรูปแบบใหม่ ๆ ได้อย่างมีประสิทธิภาพ

บรรณานุกรม

[1] ยุทธนา ลีลาศวัฒนกุล 2544. **คู่มือการเขียนโปรแกรมและใช้งาน Visual C++ 6.0.** พิมพ์ครั้งที่ 1 กรุงเทพฯ : บริษัท อินโฟเพรส จำกัด

[2] ยุทธนา ลีลาศวัฒนกุล. **Advance Visual C++ Version 6.0 ฉบับโปรแกรมเมอร์.** กรุงเทพฯ : บริษัท ซัคเซส มีเดีย จำกัด

[3] เรืองไกร รังสิพล 2544. **เจาะระบบ TCP/IP จุดอ่อนโปรโตคอลและวิธีป้องกัน .** พิมพ์ครั้งที่ 1 กรุงเทพฯ : บริษัท โปรวิชั่น จำกัด

[4] เรืองไกร รังสิพล 2545. **เปิดโลก Firewall และ Internet Security .** พิมพ์ครั้งที่ 1 กรุงเทพฯ : บริษัท โปรวิชั่น จำกัด

[5] สุรวัฒน์ ปุณณชัยยะ 2543. **เปิดโลกของ TCP/IP และ โปรโตคอลของอินเตอร์เน็ต.** ครั้งที่ 1 กรุงเทพฯ : บริษัท โปรวิชั่น จำกัด

[6] สุรศักดิ์ สงวนพงษ์ 2545. **สถาปัตยกรรมและโปรโตคอลทีซีพี/ไอพี .** พิมพ์ครั้งที่ 2 กรุงเทพฯ : บริษัท ซีเอ็ดยุเคชั่น จำกัด

[1] Anthonh Jones. 2002. **Network Programming for Microsoft Windows Second Edition.** Canada : Microsoft Press.

[2] Charlie Kaufman. 2002. **Network Security Private Communication in a PUBLIC World.** New Jersey : Prentice Hall.

[3] David J. Kruglinski. **Programming Microsoft Visual C++ Fifth Edition .**

[4] Douglas E. Comer. 2000. **Internetworking With TCP/IP Principles, Protocols, and Architecture Forth Edition.** New Jersey : Prentice Hall.

[5] Ian Sommerville. **Software Engineering 6th Edition**. Addision-wesley,2001

[6] Joel Scambray, Sturat McClure. 2001. **Hacking Exposed: Network Security Secrets & Solution Second Edition.** New York : McGraw-Hill Companies.

[7] Microsoft MSDN Library, Microsoft Corporation, August 1999