สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ไฟร์วอลแบบอิงบริการไดเรกทอรี

DIRECTORY BASED FIREWALL

โดย

นายรัฐพล ปฏิการมณฑล

นายสมศักดิ์ วงศ์สินสกุล

อาจารย์ที่ปรึกษา

อาจารย์ธนา หงษ์สุวรรณ

อาจารย์อัครเดช วัชระภูพงษ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

.b.....
.i.....

เลขหมู่.....
เลขทะเบียน 49941
วัน,เดือน,ปี 2 เม.ย. 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2545

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ไฟล์วอลแบบอิงบริการไดเรกทอรี

DIRECTORY BASED FIREWALL

ผู้จัดทำ 1. นายรัฐพล ปฏิการมณฑล รหัสประจำตัว 42010294

2. นายสมศักดิ์ วงศ์สินสกุล รหัสประจำตัว 42010365

_____ อาจารย์ที่ปรึกษา

(อาจารย์ธนา หงษ์สุวรรณ)

_____ อาจารย์ที่ปรึกษา

(อาจารย์อัครเดช วัชระภูพงษ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟร์วอลแบบอิงบริการไดเรกทอรี

นายรัฐพล ปฏิการมณฑล 42010294
นายสมศักดิ์ วงศ์สินสกุล 42010365
อาจารย์ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษา
อาจารย์อัครเดช วัชระภูพงษ์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2545

บทคัดย่อ

ในปัจจุบันนี้ระบบรักษาความปลอดภัยของคอมพิวเตอร์นั้น มีส่วนสำคัญในองค์กรธุรกิจต่างๆ จึงมีระบบรักษาความปลอดภัยมากมายเกิดขึ้นเพื่อใช้ในองค์กรธุรกิจ สำหรับระบบรักษาความปลอดภัยของคอมพิวเตอร์ที่ได้รับความนิยมนั้นคงจะหนีไม่พ้นไฟร์วอล ซึ่งไฟร์วอลในแบบอิงผู้ใช้ที่ใช้ในองค์กรธุรกิจนั้นมีราคาสูง ไม่เหมาะกับองค์กรธุรกิจขนาดเล็กและขนาดกลาง เช่น เครือข่ายในการทำงานให้ธุรกิจแบบ SMEs ซึ่งธุรกิจในลักษณะนี้ จำเป็นต้องใช้เงินทุนในการพัฒนาด้านอื่นด้วย

โครงงานจึงได้มีการพัฒนาไฟร์วอลแบบอิงผู้ใช้ในลักษณะของโค้ดเปิดขึ้นมา เพื่อเป็นการลดค่าใช้จ่ายให้กับองค์กรธุรกิจประเภทนี้ โดยใช้ลีนุกซ์ ซึ่งมีโมดูลในการทำไฟร์วอลอยู่แล้ว และทำการสร้างโมดูลขึ้นมาเพื่อทำการตรวจสอบการเปลี่ยนแปลงผู้ใช้ แล้วทำการนำเอากฎที่เหมาะสมกับผู้ใช้มาใส่ให้กับไฟร์วอลเพื่อให้การทำงานของไฟร์วอลเป็นการทำงานแบบอิงผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DIRECTORY BASED FIREWALL

Mr.Rattapon Patekarnmonthon
Mr.Somsak Wongsinsakul
Mr.Thana Hongsuwan Advisor
Mr.Akkaradach Watcharapupong Advisor

ABSTRACT

Nowadays, The computer security system had become increasingly important. More security system had been put in place within the organizations. The most popular security system is Firewall. And the popular user-based firewalls used within in most organizations are very costly and not suitable for small and medium size businesses that need to spend their limited capital to develop in other parts of the business as well.

From the reason stated above, the project has developed Firewall in the form of Open Source in order to reduce the organizational expenses by using which already has Module to do Firewall and developed new module for check currently logged-on users, then use suitable rule for each user in user based firewall system.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้กว่าจะสำเร็จได้ ถ้าไม่ได้รับความช่วยเหลือ เอาใจใส่ และให้คำแนะนำต่างๆ ที่มีประโยชน์อย่างยิ่งเสมอมา จนทำให้สามารถทำให้ปริญญานิพนธ์ฉบับนี้สำเร็จไปได้ ซึ่งต้องขอขอบพระคุณอย่างมาก ต่ออาจารย์ทั้ง 2 ท่าน คือ อาจารย์ธนา หงษ์สุวรรณ อาจารย์อัครเดช วัชระภูพงษ์

และบุคคลที่สำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ได้คือ บิดา มารดา ที่เคารพยิ่ง และเป็นผู้ที่มีพระคุณสูงสุด ที่ได้ให้โอกาสในการศึกษา เลี้ยงดู เอาใจใส่ และให้กำลังใจเสมอมา ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอกราบขอบพระคุณมา ณ. ที่นี้

รัฐพล ปฏิการมณฑล
สมศักดิ์ วงศ์สินสกุล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

โครงงานนี้เป็นโครงงานพัฒนาไฟร์วอล(Firewall)ขึ้นมาภายใต้โค้ดเปิด(Open Source) ซึ่งจะเป็นการลดค่าใช้จ่ายให้กับองค์กรธุรกิจขนาดย่อม ในการที่จะต้องเสียค่าใช้จ่ายสำหรับระบบรักษาความปลอดภัย ซึ่งมีค่าใช้จ่ายสูงในการจัดหา

โครงงานนี้ เป็นโครงงานการสร้างไฟร์วอลเพื่อใช้สำหรับกำหนดกฎ(Rules)ให้กับผู้ใช้(User)ที่พิสูจน์ตน (Authenticated) เข้าสู่ระบบเครือข่าย (Network) ภายในผ่านทางไดเรกทอรีเซอร์วิส (Active Directory) บนวินโดวส์2000 (Window2000) เมื่อผู้ใช้ สามารถเข้าสู่เครือข่ายแล้วโปรแกรมบนเครื่องไดเรกทอรีเซอร์วิส ก็จะทำการตรวจสอบผู้ใช้ และส่งชื่อผู้ใช้และไอพี(IP) ของเครื่องที่ผู้ใช้นั้นๆใช้ ให้กับไฟร์วอลกำหนดกฎตามสิทธิของผู้ใช้เพื่อให้ผู้ใช้สามารถทำงานได้ตามสิทธิของตน

1.2 วัตถุประสงค์

เพื่อพัฒนาไฟร์วอลเพื่อใช้สำหรับกำหนดกฎให้กับผู้ใช้ที่พิสูจน์เข้าเครือข่าย โดยที่ผู้ใช้สามารถใช้งานที่เครื่องใดก็ได้ ในระบบเครือข่าย

1.3 ขอบเขต

โปรแกรมจะเป็นโปรแกรมที่ทำงานในเครือข่ายขนาดเล็ก จึงอาจมีข้อจำกัดในการนำไปใช้กับเครือข่ายขนาดใหญ่ที่มีการใช้งานสูง หรือมีความซับซ้อนของระบบมากหรืออาจทำให้การตอบสนองช้าหากมีการใช้งานในปริมาณที่มาก

1.4 วิธีการดำเนินงาน

1.4.1 ศึกษาไอพีเทเบิล (IP Table) บนระบบปฏิบัติการลีนุกซ์ (Linux) เพื่อพัฒนาไฟร์วอล

1.4.2 ศึกษาไดเรกทอรีเซอร์วิสบนระบบปฏิบัติการวินโดวส์ 2000

1.4.3 พัฒนาแอพพลิเคชั่นบนระบบปฏิบัติการลีนุกซ์เพื่อพัฒนาเป็นไฟร์วอล

1.4.4 พัฒนาแอพพลิเคชั่นบนระบบปฏิบัติการวินโดวส์ 2000 เพื่อทำงานร่วมกับไฟร์วอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ไฟร์วอลล์

ไฟร์วอลนั้นถือเป็นเครื่องมือแรกที่ถือกำเนิดขึ้นมาในโลกของการรักษาความปลอดภัย (Security) เป็นเครื่องมือที่พัฒนาไปมากที่สุด เป็นที่รู้จักกันมากที่สุด และเป็นเครื่องมือตัวแรกที่มักจะใส่เข้าไปในระบบเพื่อป้องกัน

2.1 ชนิดต่างๆของไฟร์วอลตามการทำงาน

ไฟร์วอลล์ สามารถแบ่งออกได้เป็นหลายประเภท หากทำการจำแนกตามวิธีการทำงาน ก็จะสามารถแยกได้เป็น 3 แบบ คือ

2.1.1 แพ็กเกจฟิลเตอร์ริ่ง (Package Filtering)

เป็นไฟร์วอลที่มีรูปแบบการทำงานที่ง่ายที่สุด และเป็นไฟร์วอลแบบแรกที่เกิดขึ้น ซึ่งในขณะนี้มักจะเป็นฟังก์ชั่นการทำงานหลักขั้นหนึ่งของเราท์เตอร์ (Router) โดยนอกจากจะมีการทำงานในการเลือกเส้นทางแล้ว ยังจะทำการกรองแพ็กเกจ (Packet) ที่ผ่านด้วย คือจะทำการอนุญาตหรือไม่อนุญาตให้แพ็กเกจนั้นๆ ผ่านเข้าออกไฟร์วอลได้หรือไม่ โดยจะมีการกำหนดรูปแบบของกฎขึ้นมา โดยอาจจะมีการกำหนดให้ปกติเป็นอนุญาตให้ผ่านทุกแพ็กเกจ ยกเว้นแพ็กเกจที่มีรูปแบบตามกฎที่กำหนดไว้ หรือจะเป็นแบบที่ปกติไม่ยอมให้แพ็กเกจใดๆผ่าน ไฟร์วอล ยกเว้นแพ็กเกจที่มีรูปแบบที่เป็นไปตามกฎที่กำหนดไว้เท่านั้นที่ไฟร์วอลยอมให้ผ่านไปได้ โดยความสามารถของไฟร์วอลประเภทนี้นั้นจะสามารถตรวจสอบได้แต่เฉพาะเฮดเดอร์ (Header) ของแพ็กเกจเท่านั้น โดยจะไม่สามารถเข้าไปตรวจสอบเนื้อหาและการทำแฟรกเมนเตชั่น (Fragmentation) ของแพ็กเกจได้ และในการทำงานนั้นจำเป็นต้องเปิดพอร์ท (Port) ที่มากกว่า 1024 ไว้ตลอดเวลา จึงอาจเป็นเป้าหมายของการโจมตีได้ ทำให้การทำงานของไฟร์วอลประเภทนี้จะสามารถแค่กลั่นกรองแพ็กเกจที่เป็นไปตามรูปแบบเท่านั้น แต่ไม่สามารถป้องกันการโจมตีที่เข้ามายังพอร์ทที่เป็นพอร์ทปกติที่ใช้งาน เช่นพอร์ท 80 ซึ่งจะใช้ในการทำงานปกติ ไฟร์วอลส่วนใหญ่จะยอมให้แพ็กเกจที่เข้ามายัง พอร์ท 80 ผ่านได้ แต่หากผู้บุกรุกทำการเทลเน็ท (Telnet) เข้ามายัง พอร์ท 80 เพื่อทำการรันโปรแกรมแบ็คดอร์ (backdoor) ไว้ไฟร์วอลก็จะไม่สามารถตรวจสอบความผิดปกติได้เลย

2.1.2 สเตทฟูลอินสเปคชั่น (Stateful Inspection)

ในไฟร์วอลแบบแพ็คเก็จฟิลเตอริงที่ได้กล่าวมาแล้วนั้น จะเห็นได้ว่าเราสามารถตั้งค่าเพื่อกรองแพ็กเกจที่ไม่ต้องการออกไป แต่ไฟร์วอลดังกล่าว ก็ไม่ปลอดภัยมากพอ เพราะแม้ว่าเราสามารถจำกัดการเชื่อมต่อให้เหลือแต่พอร์ต 80 เท่านั้นที่ติดต่อเข้ามาได้ แต่ไฟร์วอลข้างต้นก็ไม่สามารถตรวจสอบว่าการเชื่อมต่อนั้นเป็นการเชื่อมต่อปกติหรือไม่ ซึ่งหากเป็นพอร์ต 80 ก็จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปล่อยให้ผ่านหมด ทั้งๆที่แพ็กเกจนั้นอาจจะเป็นแพ็กเกจสำหรับโจมตีก็ได้ และหากเป็นการโจมตีโดยการกำหนดแพ็กเกจให้ผิดปกตินั้น ไฟร์วอลจะไม่สามารถรับรู้ได้เลย และไม่สามารถตรวจสอบได้เลย

ข้อบกพร่องทั้งหมดนี้ ไฟร์วอลแบบสเตทฟูลอินสเปคชั่นสามารถป้องกันได้ ซึ่งไฟร์วอลแบบสเตทฟูลอินสเปคชั่นนั้นจะเป็นไฟร์วอลที่มีการตั้งกฎต่างๆเช่นเดียวกับแบบ แพ็กเกจฟิลเตอริ่ง แต่ยังเพิ่มการติดตามสเตทในการเชื่อมต่อเข้าไปด้วย ทำให้สามารถทำนายถึงความผิดปกติที่เกิดขึ้นกับการเชื่อมต่อต่างๆ ได้ว่ามีความปกติ หรือมีความผิดปกติที่อาจสงสัยได้ว่าเป็นการโจมตีได้ ทำให้มีความสามารถทางด้านความปลอดภัยสูงกว่าแบบแรก ในส่วนของความสามารถในการดูแลแพ็กเกจที่ทำการแฟรกเมนเตชั่น ไฟร์วอลแบบนี้จะทำการรอให้ข้อมูลเข้ามาจนครบ คาตาแกรม(Datagram) แล้วจึงทำการรวบรวม(Reassemble) เพื่อทำการตรวจสอบว่าเป็นแพ็กเกจที่ถูกต้องหรือไม่ ทำให้การโจมตีในรูปแบบนี้จะทำให้ยากขึ้น และไฟร์วอลแบบสเตทฟูลอินสเปคชั่น จะไม่ต้องทำการเปิดพอร์ทที่มากกว่า 1024 เนื่องจากในการติดตามสเตทของ แพ็กเกจที่เข้าออกทำให้สามารถรับรู้ว่าการเชื่อมต่อนั้นๆเกิดขึ้นที่พอร์ทใดในฝั่งไคลเอนต์ ไฟรวอลจะทำการเปิดเฉพาะพอร์ทที่มีการใช้งานจริงเท่านั้น พอร์ทอื่นๆที่ไม่ได้ใช้งานก็ไม่มีความจำเป็นต้องเปิด และเมื่อการเชื่อมต่อสิ้นสุดลง ก็จะทำการผิดพอร์ทนั้นๆ ทำให้ระบบมีความปลอดภัยมากขึ้น

อย่างไรก็ตาม ไฟร์วอลสเตทฟูลนี้จะยังไม่สามารถป้องกันการโจมตีที่แทรกซึมมากับการเชื่อมต่อตามปกติได้ เช่นโพรโตคอล FTP นั้น แม้ว่าไฟร์วอลแบบสเตทฟูลจะมีการติดตามให้มีการเปิดพอร์ท 20 และ 21 อย่างถูกต้อง แต่หากมีการส่งข้อมูลอื่นๆ ที่ไม่ใช่ข้อมูล FTP แทรกมาในระหว่างการเชื่อมต่อ ก็ไม่สามารถรับรู้ได้เลย

2.1.3 แอปพลิเคชั่นพรอกซี่ (Application Proxy)

ไฟร์วอลแบบนี้จะทำงานในระดับแอปพลิเคชั่นเป็นสำคัญ โดยไฟร์วอลจะทำหน้าที่เป็นตัวแทน(Proxy) ในการเชื่อมต่อ ไปยังเซิร์ฟเวอร์ (Server) โดยในการทำงานนั้น ไฟร์วอล จะทำการรับแพ็กเกจจากไคลเอนต์ (Client) แล้วทำการถอดแพ็กเกจเดิมออกแล้วสร้างแพ็กเกจใหม่ให้มีลักษณะการร้องขอไปยังเซิร์ฟเวอร์ เช่นเดิม เสมือนกับว่าไฟร์วอลเป็นผู้ร้องขอไปยังฝั่งเซิร์ฟเวอร์เองทำให้บุคคลภายนอกจะไม่สามารถเห็นหมายเลขไอพีของไคลเอนต์ โดยจะเป็นแค่หมายเลขไอพีของไฟร์วอลเท่านั้น และในการตอบกลับก็จะทำการตอบมายังไฟร์วอล รวมถึงสามารถเข้าถึงเนื้อหาของแพ็กเกจได้ ไฟร์วอลประเภทนี้จึงสามารถที่จะกลั่นกรองแพ็กเกจที่ไม่เหมาะสมลงไปได้ในระดับข้อมูลได้ แต่เนื่องจากมีการตรวจสอบที่มากกว่า และยังต้องสร้างแพ็กเกจใหม่ขึ้นมาทุกครั้ง จึงทำให้การทำงานของไฟร์วอลประเภทนี้จะทำงานได้ช้ากว่าแบบอื่น และยังจำเป็นต้องรู้จักการทำงานของโพรโตคอล (Protocol) ที่มากขึ้นเนื่องจากต้องทำการส่งต่อแพ็กเกจโดยการสร้างขึ้นมาใหม่ ดังนั้น การใช้งานโพรโตคอลที่ไฟร์วอลไม่รู้จักก็จะทำให้ไฟร์วอลไม่สามารถทำงานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ชนิดของไฟร์วอลตามการอ้างอิงกฎ

แต่หากทำการแยกแยะตามลักษณะการอ้างอิงกฎของ ไฟร์วอล สามารถแบ่งออกได้เป็น 2 ประเภท คือ

2.2.1 แอดเดรสเบสไฟร์วอล (Address-based Firewall)

จะทำงานโดยมีการอ้างอิงกฎกับ แอดเดรส เช่น กฎกำหนดว่า แอดเดรสเครื่อง ก. สามารถทำงานได้แบบหนึ่ง ทำให้เมื่อมีการใช้งานผู้ใช้งานที่ใช้เครื่องที่มีแอดเดรสเป็นเครื่อง ก. จะได้รับสิทธิในการใช้งานเครือข่ายที่เป็นไปตามสิทธิที่ถูกกำหนดไว้ให้กับเครื่องที่ผู้ใช้งาน ได้ใช้อยู่ แต่หากมีการเปลี่ยนแปลงเครื่อง หรือเปลี่ยนแปลงแอดเดรส สิทธิของผู้ใช้ก็จะต้องถูกเปลี่ยนแปลงไปด้วย ทำให้การใช้งานของผู้ใช้ไม่มีความสะดวกเมื่อมีการเปลี่ยนแปลงเครื่อง หรือต้องทำการใช้เครื่องมากกว่าหนึ่งเครื่อง เนื่องจากต้องทำการตั้งกฎในทุกเครื่องที่ผู้ใช้ ใช้งานอยู่ และในการใช้งานเครื่องที่มีผู้ใช้งานหลายๆคน

2.2.2 ยูสเซอร์เบสไฟร์วอล (User-based Firewall)

จะทำงานโดยการอ้างอิงกฎกับผู้ใช้ เช่น ผู้ใช้ ก. สามารถทำงานได้แบบหนึ่ง ซึ่งการใช้งานของแต่ละผู้ใช้จะแตกต่างกันไป ดังนั้นหากมีการเปลี่ยนแปลงเครื่อง ผู้ใช้ก็ยังสามารถทำงานภายใต้สิทธิที่ตนเองมีได้เนื่องจากหากมีการเปลี่ยนแปลงผู้ใช้ขึ้น ตัว ไฟร์วอลก็จะทำการปรับเปลี่ยนกฎที่มีอยู่ให้เหมาะสมกับผู้ใช้ที่ทำการใช้งานอยู่ ทำให้เกิดความสะดวกสบายให้กับผู้ใช้ คือสามารถทำงานที่เครื่องใดก็ได้ โดยที่ยังได้สิทธิในการใช้งานเครือข่ายได้เหมือนเดิม และเป็นการอำนวยความสะดวกกับผู้ดูแลระบบด้วยเนื่องจากการทำการจัดการนโยบายเกี่ยวกับความปลอดภัยนั้นจะสามารถทำขึ้นได้โดยไม่ต้องทำการกำหนดลงไปถึงระดับแอดเดรส ทำให้การบริหารจัดการกับนโยบายในระบบที่มีความซับซ้อนสูงทำได้ง่ายขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ไดเรกทอรีเซอร์วิส

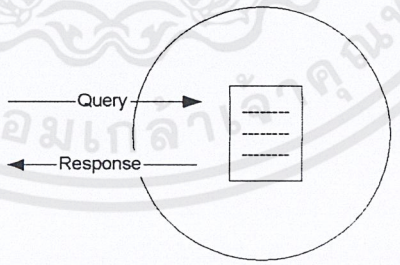
3.1 ความหมายของไดเรกทอรีเซอร์วิส

ไดเรกทอรีเซอร์วิสคือคอมพิวเตอร์สำหรับบริการซึ่งทำให้ผู้ใช้สามารถเก็บและเข้าถึงข้อมูลที่เก็บอยู่ภายในคอมพิวเตอร์บริการได้ ตัวอย่างของไดเรกทอรีเซอร์วิสที่คุ้นเคย เช่น โดเมนเนมเซอร์วิส (DNS) ซึ่งใช้ในการติดต่อกับอินเทอร์เนต

ไดเรกทอรีเซอร์วิสนั้นไม่จำเป็นจะต้องมีทุกสิ่งเหมือนกัน บางไดเรกทอรีเซอร์วิสนั้นอาจมีบริการเพียงหนึ่งบริการก็ได้ รวมทั้งฐานข้อมูลภายในของไดเรกทอรีเซอร์วิสก็ไม่เหมือนกันเช่นกัน บางองค์กรอาจมีเพียงชื่อ ที่อยู่ผู้ใช้ บางองค์กรอาจมีข้อมูลมากกว่านี้ก็ได้ หรืออาจจะมีความซับซ้อนมาก ซึ่งรวมไปถึงความปลอดภัย ความสามารถทางเน็ตเวิร์กด้วย

3.1.1ไดเรกทอรีเซอร์วิสแบบง่าย

สำหรับตัวอย่างของไดเรกทอรีเซอร์วิสอย่างง่ายนั้น คือ โดเมนเนมเซอร์วิส (DNS) ในการติดต่ออินเทอร์เนตนั้น โดเมนเนมเซอร์วิสนั้นมีความจำเป็น ซึ่งจะทำหน้าที่ในการแปลงชื่อของเว็บไซต์ต่างๆ ให้เป็นไอพี เช่น http://www.microsoft.com จะแปลงเป็น 207.46.130.14 ซึ่งข้อมูลในการแปลงนี้ จะเก็บไว้ในไดเรกทอรีเซอร์วิส โดยเมื่อผู้ใช้ติดต่อผ่านอินเทอร์เนตโดยเรียกเว็บไซต์หนึ่งๆ เครื่องที่ผู้ใช้ใช้ในการติดต่อจะทำการติดต่อไปยังโดเมนเนมเซอร์เวอร์ โดยจะไปตรวจสอบชื่อของไซต์นั้น จากนั้นโดเมนเนมเซอร์เวอร์ก็จะส่งไอพีที่เป็นของเว็บไซต์นั้นทำให้สามารถติดต่อกับเว็บไซต์นั้นได้



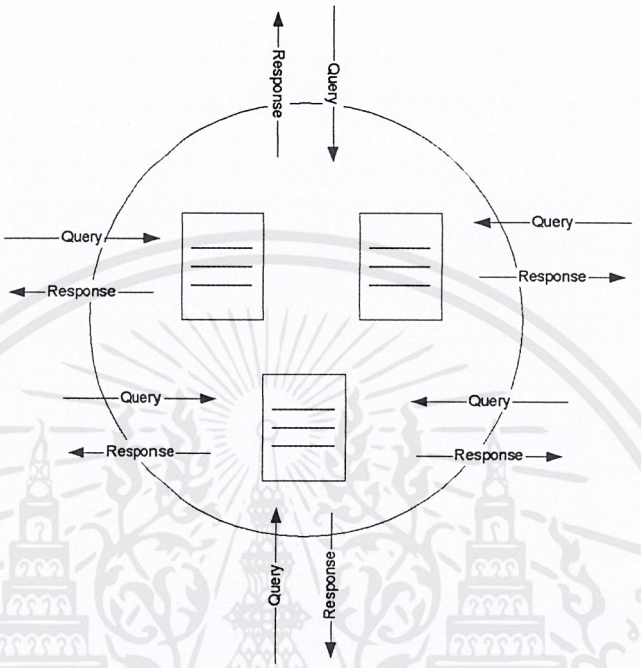
รูปที่ 3-1 แสดงไดเรกทอรีเซอร์วิสแบบง่าย

3.1.2 ไดเรกทอรีเซอร์วิสแบบซับซ้อน

เหตุผลในการใช้ไดเรกทอรีเซอร์วิสนั้นมีมากมาย ซึ่งจะแตกต่างกันไปตามสภาพแวดล้อม ซึ่งหากสภาพแวดล้อมภายในองค์กรต้องการความปลอดภัยในการรักษาข้อมูลสูงก็จำเป็นต้องใช้ไดเรกทอรีเซอร์วิสที่ซับซ้อนมากขึ้น เช่น อาจจะมีการรักษาความปลอดภัยของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูล การรักษาความปลอดภัยจากผู้ใช้งาน เป็นต้น แต่พื้นฐานของไดเรกทอรีเซอร์วิสก็ยังไม่เปลี่ยนความหมายไป กล่าวคือ ไดเรกทอรีเซอร์วิสก็ยังคงเป็นศูนย์กลางข้อมูลเหมือนไดเรกทอรีเซอร์วิสแบบทั่วไป



รูปที่ 3-2 รูปแสดงไดเรกทอรีเซอร์วิสแบบซับซ้อน

3.2 แอคทีฟไดเรกทอรี

สำหรับองค์กรธุรกิจนั้น ไดเรกทอรีเซอร์วิสนั้นจะไม่เป็นเพียงศูนย์กลางข้อมูลที่ไว้เก็บข้อมูลเพียงอย่างเดียว แต่ยังมีคุณสมบัติอื่นๆ ที่มีมากกว่าไดเรกทอรีเซอร์วิสธรรมดา สามารถตอบสนองความต้องการให้กับองค์กรได้ ซึ่งไดเรกทอรีเซอร์วิสชนิดนี้ ก็คือแอคทีฟไดเรกทอรีเซอร์วิส

3.2.1 แอคทีฟไดเรกทอรีเป็นศูนย์กลางข้อมูล

แอคทีฟไดเรกทอรีนั้น สามารถบริการข้อมูลให้กับผู้ใช้ทุกประเภท ไม่ว่าผู้ใช้นั้นจะเป็นผู้ดูแลระบบ ผู้ใช้ทั่วไป หรือ นักพัฒนา

3.2.2 แอคทีฟไดเรกทอรีทำให้เครือข่ายเป็นเรื่องง่ายสำหรับผู้ดูแลระบบ

ในปัจจุบัน แอคทีฟไดเรกทอรีมีความสำคัญกับเครือข่ายขององค์กรในแต่ละองค์กร จึงทำให้มีการพัฒนาแอคทีฟไดเรกทอรีเซอร์วิสขึ้นมามากมาย ทำให้เกิดเครื่องมือสำหรับดูแลจัดการ แอคทีฟไดเรกทอรี เพื่อให้ง่ายในการใช้งาน มากกว่าการใช้คอมมานต์โหมด ในระบบปฏิบัติการดอสและยูนิกส์ ซึ่งในโครงงานนี้ได้กล่าวถึง แอคทีฟไดเรกทอรีที่นิยมใช้กัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อย่างแพร่หลาย ก็คือ แอคทีฟไดเรกทอรีที่มีมากับ ไมโครซอฟท์วินโดวส์2000 แอดวานซ์เซิร์ฟเวอร์นั่นเอง

3.2.3 แอคทีฟไดเรกทอรีสร้างขึ้นภายใต้การรักษาความปลอดภัยของวินโดวส์2000

ดังที่ได้กล่าวมาแล้ว แอคทีฟไดเรกทอรีนั้นจะมีมาพร้อมกับ วินโดวส์ 2000 ในการใช้งานจะมีความปลอดภัยโดยการรักษาความปลอดภัยนั้น จะอยู่ภายใต้การรักษาความปลอดภัยของวินโดวส์ 2000 เช่น เมื่อมีผู้ใช้มาติดต่อขอใช้ข้อมูลในแอคทีฟไดเรกทอรี แอคทีฟไดเรกทอรีจะทำการตรวจสอบสิทธิในการเข้าถึงข้อมูลนั้นๆ ของผู้ใช้ก่อนเสมอ ซึ่งจะทำให้มีความปลอดภัยกับข้อมูลมากขึ้น

3.3 แอคทีฟไดเรกทอรีกับรีเลชั่นแนลดาต้าเบส (Relational Database)

เนื่องจากไดเรกทอรีเซอร์วิสนั้นคือศูนย์กลางข้อมูล ซึ่งข้อมูลต่างๆที่ไดเรกทอรีเซอร์วิสเก็บนั้นจัดเป็นฐานข้อมูลชนิดหนึ่ง หรือกล่าวอีกนัยหนึ่งก็คือ ไดเรกทอรีเซอร์วิสนั้นก็คือฐานข้อมูลชนิดหนึ่งนั่นเอง ซึ่งแอคทีฟไดเรกทอรีเซอร์วิสนั้นจะมีการจัดเก็บฐานข้อมูลต่างกับฐานข้อมูลทั่วไป

3.4 การรักษาความปลอดภัยของแอคทีฟไดเรกทอรี

ดังที่ได้กล่าวมาแล้วว่า แอคทีฟไดเรกทอรีนั้น มีการรักษาความปลอดภัยภายใต้พื้นฐานการรักษาความปลอดภัยของวินโดวส์ 2000 ซึ่งในหัวข้อนี้จะได้กล่าวถึงการรักษาความปลอดภัยซึ่งเป็นการรักษาความปลอดภัยทั่วๆไปของแอคทีฟไดเรกทอรี

3.4.1 การรักษาความปลอดภัยออฟเจกต์ (Object) และแอททริบิวต์ (Attribute)

ในการจัดเก็บข้อมูลภายในแอคทีฟไดเรกทอรีนั้น จะทำการจัดเก็บในรูปแบบของออฟเจกต์และแอททริบิวต์ ซึ่งในแต่ละออฟเจกต์นั้นจะมี DACL ป้องกันสิทธิการเข้าถึงของผู้ใช้กับออฟเจกต์นั้นอยู่ ซึ่งในการรักษาความปลอดภัยนี้ จะสามารถติดตั้งค่าต่างๆได้ดังนี้

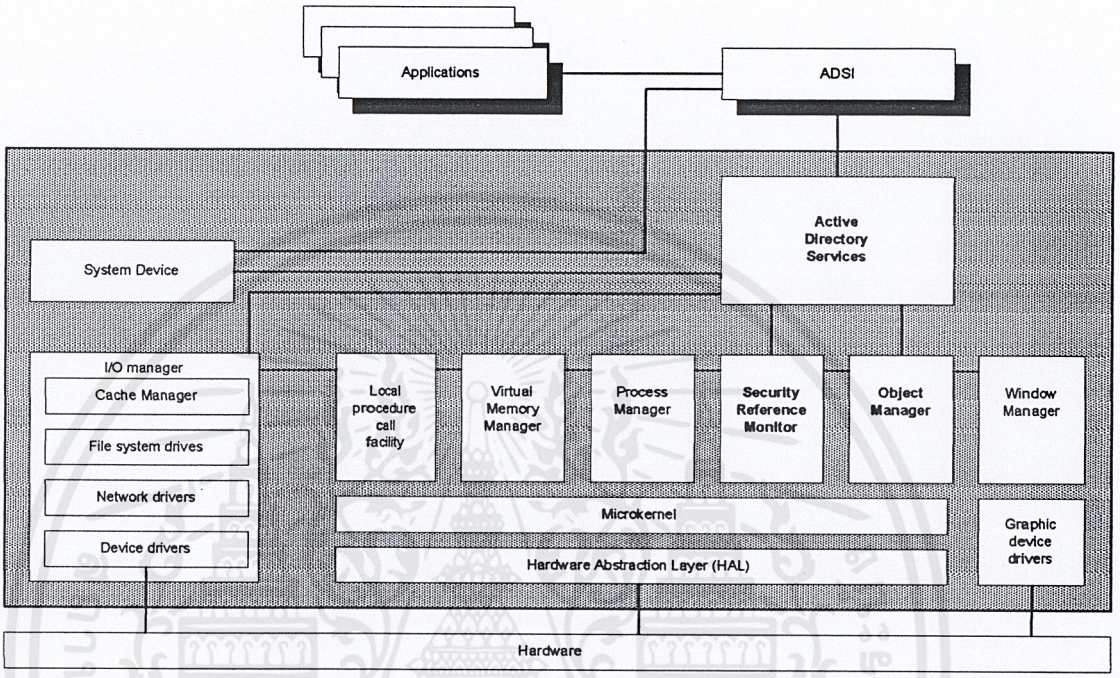
1. สามารถเข้าถึงทุกแอททริบิวต์เมื่อเข้าถึงออฟเจกต์ของแอททริบิวต์นั้น
2. สามารถแบ่งกลุ่มของแอททริบิวต์ภายในออฟเจกต์เดียวกัน

3.4.2 การรักษาความปลอดภัยของรหัสผ่าน (Password)

ถ้าจะกล่าวถึงความสำคัญของแอททริบิวต์ รหัสผ่านของผู้ใช้นั้นจะมีความสำคัญของข้อมูลมากที่สุด ดังนั้น แอคทีฟไดเรกทอรีจึงได้มีการจัดการรักษาความปลอดภัยของรหัสผ่านไว้ ซึ่งรหัสผ่านของผู้ใช้นั้น จะไม่ได้เก็บโดยตรงในแอททริบิวต์ แต่จะทำการเข้ารหัสจากรหัสผ่านจริง มาก่อน ซึ่งผู้ดูแลระบบสามารถเลือกฟังก์ชั่นในการเข้ารหัสได้ จากกนั้นจึงนำไปเก็บในแอคทีฟไดเรกทอรี หรืออาจจะจัดเก็บไว้ในสื่อต่างๆที่เคลื่อนย้ายได้ก็ได้ ในการเข้าถึงแอททริบิวต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รหัสผ่านนั้น ไม่ว่าผู้ใช้คนใดก็ไม่สามารถเข้าถึงได้ แม้แต่ผู้ดูแลระบบก็ตาม นอกเหนือจากว่าเป็นเจ้าของรหัสผ่านนั้นๆ



รูปที่ 3-3 แสดงสถานที่เก็บแอคทีฟไดเรกทอรีบนวินโดวส์ 2000

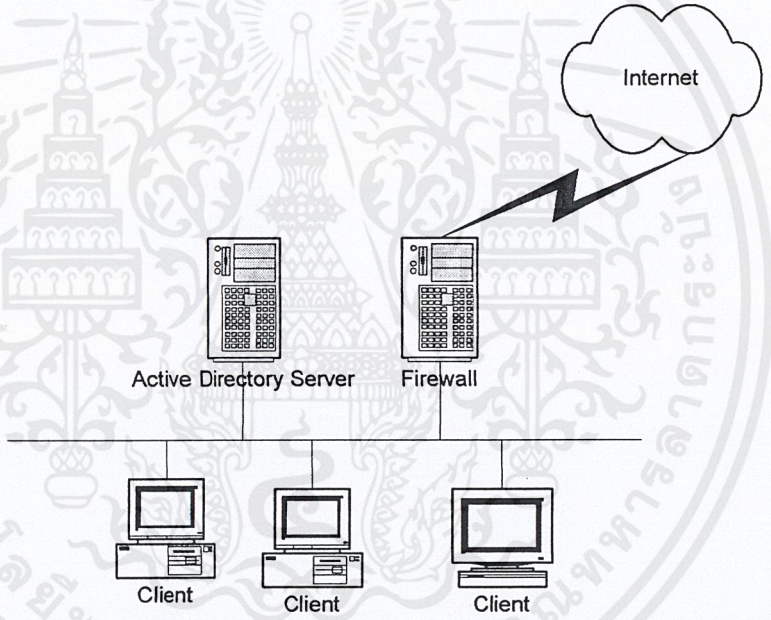
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การนำไฟร์วอลมาใช้กับไดเรกทอรีเซอร์วิส

เนื่องจากในปัจจุบัน ได้มีไฟร์วอลต่างๆออกมามากมาย ซึ่งในโครงงานนี้ก็ได้นำไฟร์วอลที่มีอยู่ในระบบปฏิบัติการลีนุกซ์อยู่แล้ว มาพัฒนาให้เป็นไฟร์วอล โดยจะใช้งานควบคู่กับไดเรกทอรีเซอร์วิส กล่าวคือ ผู้ใช้จะต้องล็อคอินผ่านเข้าสู่เครือข่ายผ่านทางแอคทีฟไดเรอทอรี่ก่อนนั่นเอง

4.1 รูปแบบเครือข่ายที่ใช้

สำหรับโครงงานนี้ ได้ออกแบบเครือข่ายโดยมีแอคทีฟไดเรกทอรีเซอร์เวอร์ รวมทั้งไฟร์วอล ซึ่งทำงานอยู่บนระบบปฏิบัติการลีนุกซ์ ดังรูปที่ 4-1



รูปที่ 4-1 แสดงแนวคิดในการออกแบบไฟร์วอลแบบอิงบริการไดเรกทอรี

จากรูป จะเห็นว่า แอคทีฟไดเรกทอรี กับ ไฟร์วอลนั้นแยกออกจากกันคนละเซิร์ฟเวอร์ โดยที่ผู้ใช้จะต้องล็อคอินเข้าสู่เครือข่าย จากเครื่องไคลเอนท์เครื่องใดเครื่องหนึ่งภายในเครือข่าย จากนั้นแอพพลิเคชั่นที่ทำงานอยู่บนแอคทีฟไดเรกทอรีเซิร์ฟเวอร์ จะทำการตรวจสอบผู้ใช้ แล้วจึงส่งชื่อผู้ใช้และไอพีแอดเดรสที่เครื่องไคลเอนท์ที่ผู้ใช้ใช้งานอยู่ไปยังไฟร์วอลเซิร์ฟเวอร์ ซึ่งทำงานอยู่บนระบบปฏิบัติการลีนุกซ์ ไฟร์วอลจะทำการตรวจสอบสิทธิของผู้ใช้ แล้วทำการกำหนดสิทธิของผู้ใช้เข้าสู่ไฟร์วอล ซึ่งสิทธิต่างของผู้ใช้จะถูกเก็บอยู่บนไฟร์วอล จะไม่ได้เก็บบนแอคทีฟไดเรกทอรีเซิร์ฟเวอร์ เมื่อไฟร์วอลได้กำหนดสิทธิของผู้ใช้เรียบร้อยแล้ว ผู้ใช้ก็สามารถใช้งานตามสิทธิของผู้ใช้นั้นได้ โดยที่ไม่จำเป็นต้องอิงกับไอพีแอดเดรสบนเครื่องไคลเอนท์นั้นๆ กล่าวคือ ผู้ใช้สามารถล็อคอินเข้าสู่ระบบได้ ไม่ว่าจะใช้เครื่อง

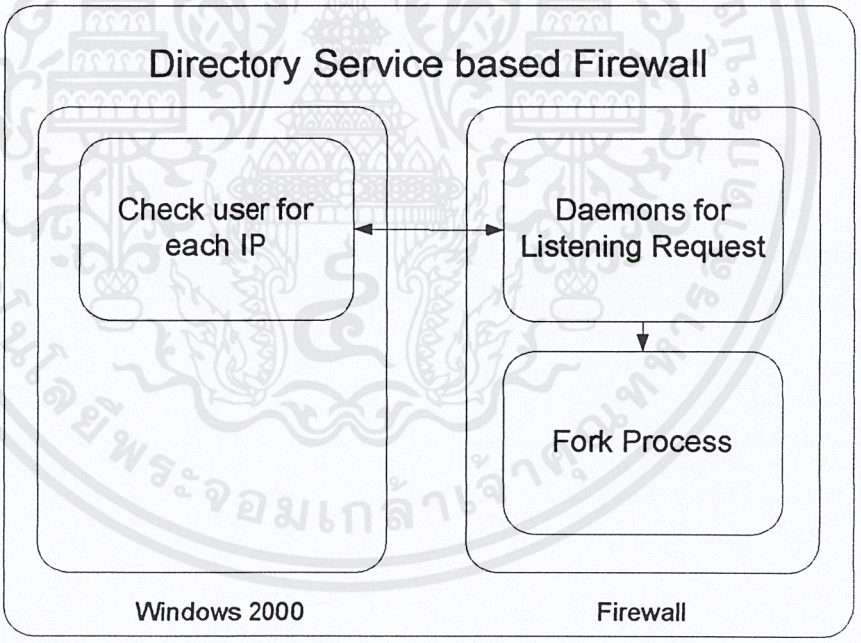
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใดก็ตาม ตามระดับของผู้ใช้ ซึ่งไฟร์วอลส่วนใหญ่ใน จะต้องอิงกับไอพีแอดเดรสที่ไคลเอนท์ที่ผู้ใช้ใช้งาน จึงจะสามารถล็อคอินเข้าสู่ระบบได้ ตามระดับของผู้ใช้

สำหรับแอพพลิเคชั่นทั้งบนฝั่งไฟร์วอลและฝั่งแอคทีฟไดเรกทอรีนั้น จะมีแอพพลิเคชั่นทำงานตลอดเวลา เพื่อทำหน้าที่ต่างๆ ซึ่งแอพพลิเคชั่นทั้งสองนี้ไม่ขึ้นต่อกัน หากแต่ทำงานร่วมกัน โดยแอพพลิเคชั่นบนแอคทีฟไดเรกทอรีจะทำการตรวจสอบผู้ใช้ที่เข้าสู่ระบบ ในขณะที่แอพพลิเคชั่นบนฝั่งไฟร์วอลจะทำหน้าที่ในการกำหนดกฎของผู้ใช้ให้กับไฟร์วอล ซึ่งทั้งสองแอพพลิเคชั่นจะทำงานร่วมกัน โดยแอพพลิเคชั่นบนฝั่งแอคทีฟไดเรกทอรีนั้นจะส่งข้อมูลของผู้ใช้ (ชื่อผู้ใช้และไอพีแอดเดรสที่ไคลเอนท์ที่ผู้ใช้ใช้งาน) ให้กับไฟร์วอล ซึ่งจะเป็นการทำงานในระดับซ็อคเก็ต (Socket) โดยทั้งสองแอพพลิเคชั่นนั้นจะมีโมดูลดังนี้

4.2 หน้าที่การทำงานในแต่ละโมดูล

สำหรับการทำงานนั้น จะแบ่งออกเป็น 2 ส่วน คือส่วนที่เป็นโมดูลบนฝั่งวินโดวส์2000 และ ส่วนที่เป็นโมดูล ซึ่งเป็นเดมอน (Daemon) บนไฟร์วอลที่ติดตั้งลีนุกซ์ ดังรูป 4-2

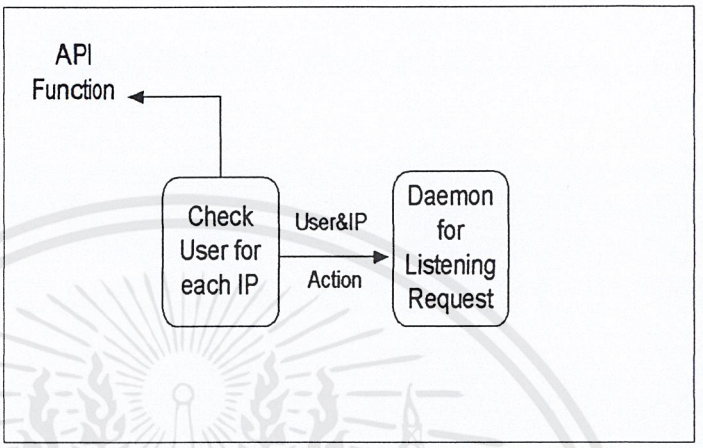


รูปที่ 4-2 แสดงโมดูลทั้ง 2 ส่วน คือบนวินโดวส์ 2000 และไฟร์วอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1 โมดูลบนฝั่งวินโดวส์ 2000

ในฝั่งวินโดวส์2000 นั้น จะสามารถทำงานได้ทั้งเครื่องบนแอคทีฟไดเรกทอรี หรือเครื่องเซิร์ฟเวอร์อื่นที่มีระบบปฏิบัติการวินโดวส์ 2000 อยู่ โดยจะมีโมดูลเพียงโมดูลเดียวเท่านั้น ดังรูป 4-3



รูปที่ 4-3 แสดงการทำงานของโมดูลบนฝั่งวินโดวส์ 2000

4.2.1.1 การตรวจสอบผู้ใช้

ในโมดูลนี้จะมีหน้าที่ในการตรวจสอบว่ามีผู้ใช้ใดได้ล็อคอินเข้าสู่ระบบ โดยการที่ผู้ใช้จะเข้าสู่ระบบนั้น ไฟร์วอลจะไม่สามารถทราบได้ว่า ผู้ใช้ใดได้เข้าสู่ระบบ เนื่องจากระบบเครือข่ายจะทำการตรวจสอบผู้ใช้ผ่านทางแอคทีฟไดเรกทอรีว่าผู้ใช้นั้นมีอยู่จริงหรือไม่ ทำให้จำเป็นต้องมีแอพพลิเคชั่นนี้เพื่อคอยบอกไฟร์วอลว่า ผู้ใช้ใดได้เข้าใช้งานในระบบ

ในส่วนของแอพพลิเคชั่นนี้ จะตรวจสอบผู้ใช้ตลอดเวลาว่ามีผู้ใช้เข้าใช้งานในระบบจริง เมื่อผู้ใช้ออกจากระบบ โมดูลส่วนนี้ก็จะตรวจสอบผู้ใช้ว่าผู้ใช้ได้ทำการล็อคเอาท์ออกจากระบบไปแล้ว จากนั้นก็จะส่งไปยังไฟร์วอลเพื่อแจ้งเว่า ผู้ใช้นั้นๆบนไอพีแอดเดรสนั้นๆ ได้ออกจากระบบไปแล้ว ทั้งนี้ก็เพื่อป้องกันไม่ให้มิให้บุคคลภายนอกข้าสู่ระบบได้ รวมทั้งเพื่อป้องกันผลประโยชน์ของผู้ใช้ด้วยกันเอง โดยภายในโมดูล จะมีฟังก์ชั่นที่ใช้ในการตรวจสอบการเข้าออกของผู้ใช้และ ฟังก์ชันในการส่งข้อมูลของผู้ใช้ไปยังไฟร์วอลในระดับซ็อคเก็ต ซึ่งจะได้กล่าวถึงในบทต่อไป

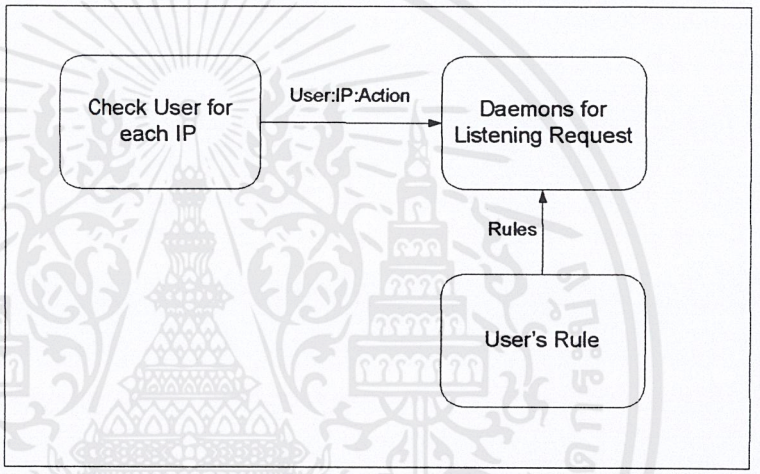
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 โมดูลบนฝั่งไฟร์วอล

สำหรับโมดูลบนฝั่งไฟร์วอลนั้น จะแบ่งออกเป็น 2 โมดูลหลักๆ ดังนี้

4.2.2.1 Daemons for Licensing Request

ดังที่ได้กล่าวมาแล้ว แอพพลิเคชั่นบนฝั่งวินโดวส์ 2000 นั้นจะทำการตรวจสอบผู้ใช้แล้วส่งข้อมูลการเข้าออกของผู้ใช้นั้น ให้กับไฟร์วอล โมดูลจึงทำหน้าที่ในการรับคำร้องจากแอพพลิเคชั่นบนฝั่งวินโดวส์2000 นั่นเอง โดยจะทำการรับข้อมูลในระดับซ็อคเก็ต ซึ่งโมดูลนี้เมื่อรับข้อมูลมาจะทำการตรวจสอบกฎต่างๆ ของผู้ใช้ที่อยู่ในไฟร์วอล และจะส่งกฎของผู้ใช้นั้นๆ ไปยังโมดูลอีกโมดูลหนึ่งบนฝั่งไฟร์วอลเซิร์ฟเวอร์ดังรูป 4-4

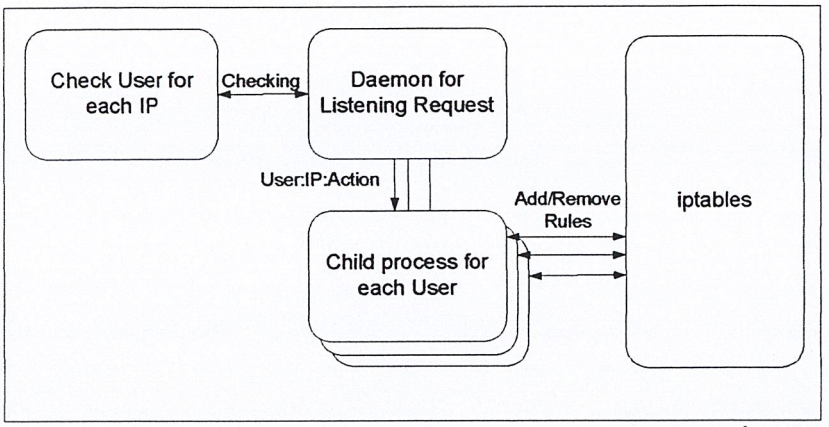


รูปที่ 4-4 แสดงการทำงานของโมดูล Daemons for Licensing Request บนฝั่งไฟร์วอล

4.2.2.2 Child Process for Each User

สำหรับโมดูลนี้ จะทำหน้าที่ในการกำหนดสิทธิต่างๆของผู้ใช้ ซึ่งสิทธิของผู้ใช้นี้จะถูกกำหนดเป็นโพรเซส (Process) โดยจะทำงานอยู่ตลอดเวลา จนกว่าจะมีการแจ้งเตือนจากฝั่งวินโดวส์2000 เซิร์ฟเวอร์ว่าผู้ใช้ออกจากระบบไป ก็จะทำการลบโพรเซสนั้นๆออกจากการทำงานทิ้งไปจากแอพพลิเคชั่น ดังรูป 4-5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-5 แสดงการทำงานของโมดูล Child Process for Each ผู้ใช้ บนฝั่งไฟร์วอล

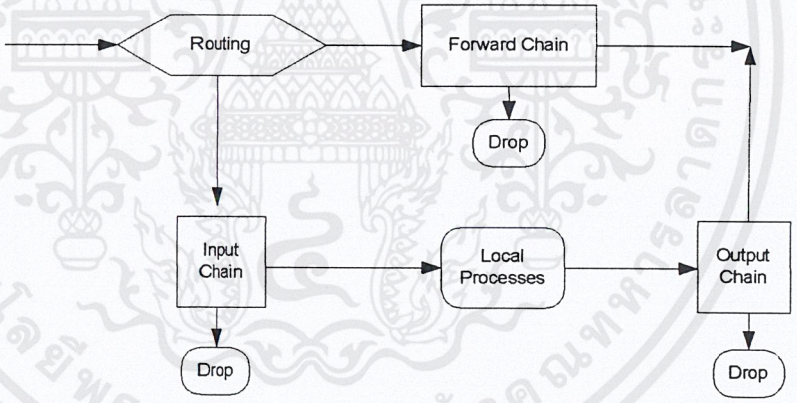
จะเห็นได้ว่า การทำงานของทั้งสองแอพพลิเคชั่นในระบบ ไม่มีแอพพลิเคชั่นใดขึ้นต่อกันโดยตรง หากแต่ทั้งสองแอพพลิเคชั่นนั้น จะต้องทำงานร่วมกัน โดยจะมีการติดต่อกันระหว่างแอพพลิเคชั่นตลอดเวลา ซึ่งการติดต่อระหว่างทั้งสองแอพพลิเคชั่นนี้จะติดต่อผ่านทางซ็อกเก็ต ซึ่งการติดต่อผ่านทางซ็อกเก็ตนี้จะต้องมีการรักษาความปลอดภัย แต่ในโครงงานนี้จะไม่กล่าวถึงความปลอดภัยของการติดต่อระหว่างซ็อกเก็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การพัฒนาไฟร์วอลโดยใช้ไอพีเทเบิล

ไอพีเทเบิล (iptables) และ เนทฟิลเตอร์(netfilter) เป็นเฟรมเวิร์คที่อยู่ใน เคอร์เนล(kernel) ลีนุกซ์ 2.4 ซึ่งจะสามารถทำการ แพ็กเกจฟิลเตอร์ริ่ง (Package filtering) การทำเน็ตเวิร์คแอดเดรสทรานสเลชั่น (Network Address Translation NAT) การตรวจสอบสเตทของการติดต่อ (State Inspection) และการทำกลั่นกรองแพ็กเกจในแบบอื่นๆ โดยจะมีการทำงานที่เข้าถึง เน็ตเวิร์คสแตค ภายในเคอร์เนล ลีนุกส์ 2.4(Linux 2.4.x kernel's network stack) ซึ่งจะทำให้เคอร์เนลอนุญาตให้ทำ register callback functions ทุกครั้งที่แพ็กเกจผ่านเข้ามา โดยจะมีการทำงานโดยการอิงชุดของกฎ(Rulesets) เข้ากับหมายเลขไอพี นอกจากนั้นยังทำการรวมเอาส่วนของแนท (NAT) เข้าไปกับเฟรมเวิร์คของไอพีเทเบิลไว้ด้วย

การทำงานของไอพีเทเบิล นั้นจะประกอบด้วยอินพุทเชน(INPUT chain) เอาท์พุทเชน (OUTPUT chain) และ ฟอร์เวิร์ดเชน (FORWARD chain) โดยแพ็กเกจขาเข้า (incoming Packet) จะผ่านเราท์ติ้งฟังก์ชัน (routing function) เพื่อวิเคราะห์ว่า แพ็กเกจ นั้นๆ จะผ่านเข้าไปตรวจสอบกับฟอร์เวิร์ดเชนหรืออินพุทเชน (Input chain) ดังรูป 5-1



รูปที่ 5-1 Netfilter packet traversal (จาก "Linux 2.4 Packet Filtering HOWTO, by Rusty Russel, v.1.0.1)

หากเป็นแพ็กเกจที่มาจากโลคอลแอดเดรส (Local Address) และผ่านอินพุทเชน แพ็กเกจนั้นก็จะถูกส่งไปยังโลคอลแอดเดรสที่กำหนดในแพ็กเกจ แต่หากเป็นแพ็กเกจจากภายนอก ที่ต้องการผ่านเข้ามาในไฟร์วอล ก็จะถูกเทียบกับฟอร์เวิร์ดเชน หากผ่านตามกฎแพ็กเกจนั้น ก็จะถูกส่งไปยังอินเทอร์เฟสที่เหมาะสมตามที่กำหนดไว้

สำหรับแพ็กเกจขาออก (Outgoing packets) ที่เป็นโลคอลแอดเดรส และผ่านเอาท์พุทเชน แพ็กเกจ นั้นก็จะถูกส่งไปยังอินเทอร์เฟส ที่เหมาะสมตามที่กำหนดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1 ส่วนประกอบของ ไอพีเทเบิล

5.1.1 ฟิลเตอร์เทเบิล (Filter table)

เป็น ส่วนที่จะดูแลการกรองแพ็กเกจต่างๆ ตามกฎที่กำหนดไว้ซึ่งจะเป็นส่วนหลักในการทำงานซึ่งจะมีความสามารถในการทำงานที่พิเศษดังนี้

- ตรวจสอบสถานะของการติดต่อซึ่งสามารถที่จะเข้าถึง ทีซีพีสเตทแฟลก (TCP state flags)และ ทีซีพีออพชั่นฟิลล์ (TCP option field) ได้
- กำหนดพอร์ตในการทำงานแบบหลายพอร์ต
- ทำการตรวจสอบการทำงานของอีเทอร์เน็ทแอดเดรสของอุปกรณ์ (MAC Address)
- เข้าถึงฟิลข้อมูลรูปแบบของการบริการในทีซีพี (TOS) โดยทำงานร่วมกับแมนเกิลเทเบิล
- ทำการควบคุมอัตราของข้อมูล
- ทำการใส่ค่าเข้าไปใน มาร์กฟิลล์ (packet's mark field)
- ทำการตรวจสอบการใช้งานของผู้ใช้โดยการอิงไอดีของผู้ใช้(User ID), ไอดีของกลุ่มของผู้ใช้(Group ID) หรือ โพรเซสไอดี(Process ID)

5.1.2 แนทเทเบิล (NAT Table)

เป็นส่วนที่ดูแลการทำแนท เช่น การทำ SNAT (Source NAT), DNAT (Destination NAT), MASQUERADE และการทำ REDIRECT โดยการทำงานนั้นจะแบ่งออกเป็น 3 แบบคือ

- การทำแนทแบบเดิม หรือ การทำแนทแบบทางเดียวออกภายนอก(Traditional, unidirectional outbound NAT) จะเป็นการทำงานที่ใช้งานบน เครือข่ายที่มีการใช้ ไพรเสทไอพี โดยจะมีสองแบบ คือ
  - แนทแบบธรรมดา จะทำการทำแผนผังเทียบ แอดเดรสที่เป็นไพรเวทในเครือข่ายภายใน ออกเป็นแอดเดรสที่เป็นเครือข่ายสาธารณะ
  - แนทที่ทำการเปลี่ยนพอร์ตของเครือข่าย(Network Address Port Translation) จะทำการเปลี่ยนจากไพรเวทแอดเดรสภายในเป็นแอดเดรสสาธารณะเพียงแอดเดรสเดียว
- การทำแนทแบบสองทาง(Bidirectional NAT) จะทำการเปลี่ยนแอดเดรสทั้งการส่งข้อมูลออกจากระบบ และการรับข้อมูลจากภายนอกเข้ามาใช้งานเครื่องที่อยู่ภายใน ซึ่งสามารถทำงานได้ทั้งไอพีเวอร์ชั่นสี่(IPv4) และไอพีเวอร์ชั่นหก (IPv6) ด้วย
- การทำแนทสองครั้ง(Twice NAT) จะเป็นการทำการแนทเมื่อมีการเปลี่ยนแปลงแอดเดรสสาธารณะ เนื่องจากหากมีการกำหนดแอดเดรสให้กับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องที่เป็นแอดเดรสที่จะทำการเนทไปนั้น จะต้องมีการเปลี่ยนแปลงการแนทใหม่ แต่การทำแนทสองครั้งนั้นจะทำการติดตามการเปลี่ยนแปลงแอดเดรสสาธารณะ และทำการปรับเปลี่ยนการแนทให้เหมาะสมกับแอดเดรสที่ได้รับใหม่ทำให้ผู้บริหารระบบไม่จำเป็นต้องติดตามการเปลี่ยนแปลงและทำการเปลี่ยนแปลงค่าที่ตั้งไว้ใหม่

5.1.3 แมนเกิลเทเบิล (Mangle table)

เป็นส่วนที่จะดูแลแพ็กเกจ แบบพิเศษกว่าฟิลเตอร์เทเบิล โดยจะทำการเปลี่ยนแปลงที่ TOS ฟิลด์ เพื่อการตัดสินใจในการทำโลคอลเราท์ติ้งหรือฟอร์เวิร์ดดิ้งซึ่งจะแบ่งเป็น

- พรีเราท์ติ้ง (PREROUTING) จะทำการเปลี่ยนแปลงแพ็กเกจขาเข้า (Incoming Packet) ก่อนที่จะทำการเราท์ติ้ง
- เอาท์พุท จะทำการเปลี่ยนแปลงแพ็กเกจ ที่จะส่งออกไปจากโลคอลแพ็กเกจ

5.2 การใช้งานไอพีเทเบิล

ในแต่ละเทเบิลนั้นจะมีการสร้างเชนที่มากับเทเบิลอยู่ดังนี้

- **ฟิลเตอร์เทเบิล** จะประกอบด้วยเชนหลักอยู่ 3 เชน คือ
  - อินพุตเชน
  - เอาท์พุตเชน
  - ฟอร์เวิร์ดเชน
- **แนทเทเบิล**
  - พรีเราท์ติ้งเชน (สำหรับการทำ DNAT กับ REDIRECT)
  - เอาท์พุตเชน (สำหรับการทำ DNAT กับ REDIRECT)
  - โฟสเราท์ติ้งเชน (สำหรับการทำ SNAT กับ REDIRECT)
- **แมนเกิลเทเบิล**
  - พรีเราท์ติ้งเชน (สำหรับการทำการเราท์แพ็กเกท)
  - เอาท์พุตเชน (สำหรับการสร้างแพ็กเกทใหม่ขึ้นภายใน)

<u>การใช้งานไอพีเทเบิล</u>

- **คำสั่งที่ใช้กับเชน**

<u>การสร้างเชนใหม่</u>

```
iptables -N name_of_chain
```

<u>การลบเชนที่สร้างขึ้นมาเอง</u>

```
iptables -X name_of_chain*
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

*ถ้า name_of_chain ไม่มี จะทำการลบ เชนที่สร้างขึ้นเองทั้งหมด

การเปลี่ยนนโยบายของเชนที่อยู่ภายใน

iptables -P name_of_chain name_of_policy*

*DROP หรือ ACCEPT

การแสดงกฎทั้งหมดที่อยู่ในเชน

iptables -L name_of_chain

การลบกฎในเชน

iptables -F name_of_chain*

*ถ้า name_of_chain ไม่มี จะทำการลบกฎทั้งหมดในเชน

การตั้งตัวนับแพ็กเกท และตัวนับข้อมูลของกฎในเชน

iptables -Z name_of_chain

- **คำสั่งที่เกี่ยวกับกฎ**

การเปลี่ยนเทเบิล (-t table_name). หาก table_name ไม่มีจะหมายถึงฟิลเตอร์เทเบิล

การกระทำกับเชน จะมี การเพิ่มไปที่กฎแรก (-A) ,การลบ (-D), การแทนที่ (-R) และการเพิ่มที่กฎสุดท้าย (-I) จากนั้นตามด้วยชื่อเชนในเทเบิลนั้นๆ

การตรวจสอบต่างๆ

-p (โปรโตคอล: TCP, UDP, ICMP หรืออื่นๆ)

-s (แอดเดรสต้นทาง)

-d (แอดเดรสปลายทาง)

-i(อินเทอร์เฟสที่เข้า)

-o (อินเทอร์เฟสออก)

--fragment (ตรวจสอบการแฟรกเมนท์)

--dport (พอร์ตปลายทาง)

--sport (พอร์ตต้นทาง)

--port (พอร์ตทั้งต้นทางและปลายทางที่เท่ากับที่กำหนด)

--mark (ค่าของ nfmark)

--tcp-flags (ตรวจสอบแฟรกต่างๆ)

--syn (การทำ SYNC Attack)

--tcp-option (ค่าของทีซีพีออพชั่น)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

49941

--state (NEW, ESTABLISHED, RELATED, INVALID)

--mac-source (source MAC address)

--tos (ค่า TOS)

--ttl (ค่า ttl)

- **การควบคุมอัตราของข้อมูล**

--limit (อัตราของข้อมูล)

--limit-burst (ค่ามากที่สุดในการ burst ข้อมูล)

- **การควมคุมอัตราของข้อมูลตามไอพี**

--iplimit-above จำกัดจำนวนการติดต่อของ ทีซีพี กับไอพี่ที่กำหนดไว้

- **การควบคุมตามผู้ส่ง**

--uid-owner userid (ควบคุมตามไอดีของผู้ใช้)

--gid-owner groupid (ควบคุมตามไอดีของกลุ่มผู้ใช้)

--pid-owner processid (ควบคุมตามไอดีของโพรเซส)

--sid-owner sessionid (ควบคุมตามไอดีของเซสชั่น)

- **เป้าหมายในการทำกับแพ็กเกจที่ตรงกับกฎนั้นๆ:**

ACCEPT – รับแพ็กเกจ

DROP – ทิ้งแพ็กเกจนั้น

REJECT – ทิ้งแพ็กเกจนั้นและแจ้งไปยังผู้ส่ง

LOG – บันทึกลงล๊อก แล้วทำงานต่อ

MIRROR – สลับแอดเดรสผู้รับและผู้ส่งและทำงานต่อ

QUEUE – ทำการส่งเข้าไปใน ยูสเซอร์สเปซโพรเซสคิว(userspace process queue)

RETURN – กลับไปยังเชนที่ทำการเรียกมา

ผลลัพท์ที่ได้จากการใช้คำสั่ง 'iptables –h'

```
iptables v1.2.2

Usage: iptables -[ADC] chain rule-specification [options]
       iptables -[RI] chain rulenum rule-specification [options]
       iptables -D chain rulenum [options]
       iptables -[LFZ] [chain] [options]
       iptables -[NX] chain
       iptables -E old-chain-name new-chain-name
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        iptables -P chain target [options]
        iptables -h (print this help information)

Commands:
Either long or short options are allowed.
  --append  -A chain            Append to chain
  --delete  -D chain            Delete matching rule from chain
  --delete  -D chain rulenum
                                Delete rule rulenum (1 = first) from
chain
  --insert  -I chain [rulenum]
                                Insert in chain as rulenum (default
1=first)
  --replace -R chain rulenum
                                Replace rule rulenum (1 = first) in
chain
  --list    -L [chain]          List the rules in a chain or all
chains
  --flush   -F [chain]          Delete all rules in  chain or all
chains
  --zero    -Z [chain]          Zero counters in chain or all chains
  --check   -C chain            Test this packet on chain
  --new     -N chain            Create a new user-defined chain
  --delete-chain
            -X [chain]          Delete a user-defined chain
  --policy  -P chain target
                                Change policy on chain to target
  --rename-chain
            -E old-chain new-chain
                                Change chain name, (moving any
references)
Options:
  --proto       -p [!] proto    protocol: by number or name, eg.
`tcp'
  --source      -s [!] address[/mask]
                                source specification
  --destination -d [!] address[/mask]
                                destination specification
  --in-interface -i [!] input name[+]
                                network interface name ([+] for
wildcard)
  --jump        -j target
                                target for rule (may load target
extension)
  --match       -m match
                                extended match (may load extension)
  --numeric     -n              numeric output of addresses and ports
  --out-interface -o [!] output name[+]
                                network interface name ([+] for
wildcard)
  --table       -t table        table to manipulate (default:
`filter')
  --verbose     -v              verbose mode
  --line-numbers                print line numbers when listing
  --exact       -x              expand numbers (display exact values)
[!] --fragment  -f              match second or further fragments
only
  --modprobe=          try to insert modules using this command
  --set-counters PKTS BYTES     set the counter during insert/append
[!] --version   -V              print package version.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 การพัฒนาไฟร์วอลโดยใช้ไอพีเทเบิล

เนื่องจากที่ ไอพีเทเบิล ทำงานในลักษณะอิงแอดเดรส (Address-Based) แต่ต้องการพัฒนาให้อิงผู้ใช้ (User-based) แนวคิดของการพัฒนาคือการให้อิงแอดเดรส ในขณะเวลาหนึ่งๆที่ผู้ใช้ คนนั้นล็อคอิน (login) อยู่ โดยต้องมีส่วนที่คอยตรวจสอบว่ามี ผู้ใช้ ทำงานอยู่ที่แอดเดรสใด และมีส่วนที่คอยรับผลจากโปรแกรมข้างต้น นำมาทำการสร้างกฎที่เหมาะสม คือ เป็นกฎสำหรับผู้ใช้ที่ล็อคอิน อยู่ที่แอดเดรสนั้นๆ แล้วทำการเพิ่ม เข้าไปและคอยตรวจสอบว่า ผู้ใช้ที่อยู่ที่แอดเดรสนั้นเปลี่ยนแปลงหรือไม่ และคอยทำการปรับเปลี่ยนกฎให้เหมาะสมกับผู้ใช้ปัจจุบันที่ล็อคอิน อยู่ที่แอดเดรสนั้นๆ โดยในโครงงานนี้ ส่วนที่คอยติดตามรับผลที่ได้จากการตรวจสอบของโปรแกรมที่อยู่ในส่วนของวินโดวส์2000 ที่จะคอยติดตามการเปลี่ยนแปลงของ ผู้ใช้ ที่ทำการล็อคอินผ่านแอคทีฟไดเรกทอรี (Active Directory) แล้วทำการส่งข้อมูลในเกี่ยวกับการเปลี่ยนแปลง ผู้ใช้ มายังเดมอน (Daemon) ที่อยู่บน ลีนุกซ์ ผ่าน โปรโตคอลที่กำหนดขึ้นมาเอง เมื่อ โปรแกรมที่อยู่บน ลีนุกซ์ ได้รับข้อมูลของการเปลี่ยนแปลง ซึ่งจะประกอบไปด้วย ชื่อผู้ใช้ กับหมายเลขไอพี (IP Address) ก็จะนำชื่อผู้ใช้ ไปค้นหากฎที่ทำการเก็บอยู่ใน แฟ้มข้อมูล แล้วนำมาใส่ในส่วนของหมายเลขไอพี แล้วทำการ เพิ่ม หรือ ลบ กฎ ลงไปในไอพีเทเบิล ซึ่งเดมอน ที่อยู่บนลีนุกซ์ จะทำการคอยรับที่พอร์ท ที่ทำการตั้งไว้ตลอดเวลา เมื่อมีข้อมูลส่งมาโปรแกรมจะรับข้อมูลมาทำงานทันที และในการทำงานจะมีการทำงานอยู่ สองแบบ คือ ทำงานเป็นเบื้องหลัง(background mode) และเมื่อมีการกระทำใดๆ ก็จะทำการ เก็บข้อมูลลงในล็อคไฟล์ (log file) ซึ่งข้อมูลที่จะเก็บจะประกอบไปด้วย

- หมายเลขไอพี
- เวลา
- ผู้ใช้ ที่ล็อคอินอยู่ที่หมายเลขไอพีนั้นๆ
- กฎที่ตั้งไว้สำหรับผู้ใช้นั้นๆ
- การกระทำที่ทำกับไอพีเทเบิล (เพิ่ม หรือ ลบ กฎ)

ส่วนอีกแบบหนึ่งคือการทำงานเป็น โพรเซส (Process) ทั่วไป และมีการแสดงผลที่ได้ออกมายังคอนโซล (Console) เหมือนโปรแกรมทั่วไป

ในส่วนของการติดต่อกับไอพีเทเบิลนั้น เนื่องจากทางผู้พัฒนาโปรแกรมไม่ได้ทำการสร้างเอพีไอ (API) เพื่อให้ โปรแกรมเมอร์ ได้ติดต่อกับ ตัวโปรแกรม ดังนั้นจึงจำเป็นต้องใช้การฟอร์คโพรเซส (fork process) มาเพื่อทำการเอ็กซ์คิว (Execute) โปรแกรม ไอพีเทเบิล ในโหมดที่ทำการ เพิ่มหรือลบกฎต่าง เมื่อไอพีเทเบิล ทำงานเสร็จเรียบร้อยแล้วก็จะทำการส่งค่ากลับออกมาโพรเซส ที่ฟอร์คขึ้นมาก็จะตายไป แล้วโปรแกรมก็จะกลับมาทำงานต่อตามปกติดังตัวอย่างโปรแกรม

```
#include <stdio.h>
#include <sys/types.h>

void parse(char *line, char **argv)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

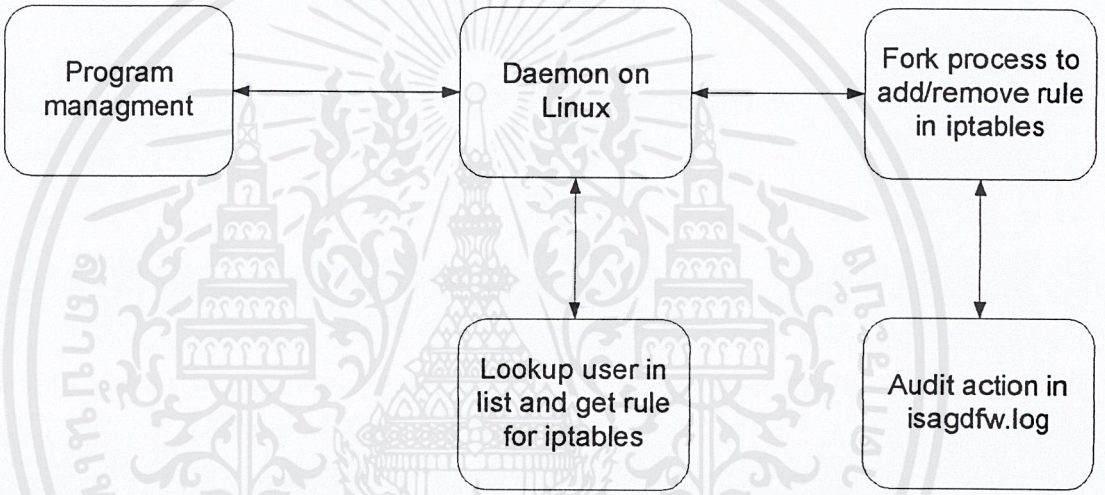
```
        while (*line != '\0') {       /* if not the end of line ..... */
    while (*line == ' ' || *line == '\t' || *line == '\n')
        *line++ = '\0';     /* replace white spaces with 0    */
    *argv++ = line;          /* save the argument position     */
    while (*line != '\0' && *line != ' ' &&
          *line != '\t' && *line != '\n')
        line++;             /* skip the argument until ...    */
        }
        *argv = '\0';                /* mark the end of argument list  */
}
void  execute(char **argv)
{
    pid_t  pid;
    int    status;

    if ((pid = fork()) < 0) {     /* fork a child process           */
        printf("*** ERROR: forking child process failed\n");
        exit(1);
    }
    else if (pid == 0) {          /* for the child process:         */
        if (execvp(*argv, argv) < 0) {     /* execute the command  */
            printf("*** ERROR: exec failed\n");
            exit(1);
        }
    }
    else {                                /* for the parent:      */
        while (wait(&status) != pid);       /* wait for completion  */
        }
}
```

รูปที่ 5-2 source code file exec.c

จากรูปที่ 5-2 นั้นไฟล์ exec.c นั้นจะมีฟังก์ชั่น อยู่ 2 ฟังก์ชั่น คือ parse() และ execute() โดยการทำงานนั้นรูทีน execvp() นั้นจะต้องการอาร์กิวเมนต์ (argument) อยู่ สองตัว โดยที่ตัวแรกจะเป็นชื่อของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม ที่ต้องการที่จะทำการเอ็กซ์คิว ส่วนอาร์กิวเมนต์ ที่สองจะเป็นอาร์เรย์ (Array) ของตัวชี้ (Pointer) ไปยังตัวอักษร (character) โดยจะเป็นอาร์กิวเมนต์ ที่จะใส่เข้าไปในโปรแกรมที่ต้องการเอ็กซ์คิว ดังนั้นจึงมีฟังก์ชั่น parse() ขึ้นมาเพื่อทำการตัดสตริง (String) ที่ได้จากการอ่านไฟล์ ที่เก็บ กฎเอาไว้ มาใส่ไว้ในอาร์เรย์ เพื่อเตรียมให้ฟังก์ชั่น execvp() ทำการใช้งาน โดยเมื่อทำการตัดสตริงออกมาเป็นอาร์เรย์ แล้วก็ส่งค่าไปยังฟังก์ชัน execute() ฟังก์ชั่นก็จะทำการฟอล์คโพรเซสขึ้นมา แล้วทำการเอ็กซ์คิวโปรแกรมแล้วรอจนกว่าโพรเซสที่ทำการเอ็กซ์คิวโปรแกรม เสร็จเรียบร้อย แล้วก็จะทำการส่งค่าฟังก์ชั่นกลับ (return function) เพื่อกลับไปทำงานในโปรแกรมหลัก (main program) ต่อไป



รูปที่ 5-3 รูปแสดงการทำงานของ โปรแกรมฝั่งลีนุกซ์

จากรูปที่ 5-3 ซึ่งจะแสดงการทำงานของโปรแกรมที่อยู่บนฝั่งลีนุกซ์ โดยการทำงานนั้นจะเป็นลักษณะของ เดมอน (Daemon process) คือจะทำการรอรับการติดต่อจากโปรแกรมที่อยู่บนฝั่ง วินโดวส์ 2000 ซึ่งจะทำการตรวจสอบการเปลี่ยนแปลงของผู้ใช้งานที่ทำการล็อกอินอยู่ที่ไคลเอนท์ แล้วจะทำการส่งข้อมูลมายังส่วนที่อยู่บนลีนุกซ์เมื่อมีการเปลี่ยนแปลง โดยจะส่งมาในรูปแบบดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<User>:<ip address><command>

โดยในการส่งข้อมูลนั้นคำสั่ง จะเป็นการสั่งให้ทำการ เพิ่ม หรือ ลบกฎออกจากไอพีเทเบิลนั้นเอง โดยเมื่อโปรแกรมที่อยู่บนฝั่งลินุกซ์ได้รับข้อความจากฝั่งวินโดวส์ 2000 นั้น ก็จะทำการแยกข้อมูลที่ได้รับมาออกเป็นสามส่วน แล้วทำการนำเอาชื่อผู้ใช้ไปค้นหาจากแฟ้มข้อมูลที่เก็บกฎของผู้ใช้ไว้ จากนั้นก็ทำการสร้างโพรเซสใหม่ขึ้นมาทำการเพิ่มหรือลบกฎออกจากไอพีเทเบิล และทำการบันทึกการทำงานลงในแฟ้มข้อมูลชื่อ isagdfw.log เพื่อเก็บข้อมูลการทำงานเพื่อการตรวจสอบต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
การพัฒนาแอพพลิเคชั่นบนฝั่งวินโดวส์ 2000

จากบทที่แล้ว ได้กล่าวถึงการทำงานของแอพพลิเคชั่นทั้งสองเซิร์ฟเวอร์ ในบทนี้จะขอกล่าวถึงการพัฒนาแอพพลิเคชั่นบนฝั่งวินโดวส์ 2000 ซึ่งแอพพลิเคชั่นนี้ไม่จำเป็นต้องทำงานบนแอคทีฟไดเรกทอรี แต่สามารถทำงานที่เครื่องที่เป็นเซิร์ฟเวอร์เครื่องใดก็ได้ ที่มีระบบปฏิบัติการวินโดวส์ 2000 ติดตั้งอยู่ในเครื่องนั้น

6.1 แนวคิดในการออกแบบโปรแกรม

สำหรับแอพพลิเคชั่นบนฝั่งวินโดวส์ 2000 นี้ จะมีหน้าที่ในการตรวจสอบการเข้าออกของผู้ใช้ จึงมีแนวคิดในการที่จะตรวจสอบผู้ใช้ภายในแต่ละเครื่องว่าเป็นผู้ใช้ใด ซึ่งในการตรวจสอบว่าผู้ใช้ใดใช้งานอยู่บนเครื่องนั้น เป็นเรื่องที่ไม่ยากเกินไปนัก แต่การที่จะทราบว่าผู้ใช้ได้เข้าสู่ระบบนั้น เป็นเรื่องที่สำคัญอย่างยิ่ง เนื่องจากเมื่อผู้ใช้ล็อคอินเข้าสู่ระบบ ผู้ใช้ก็ต้องการที่จะใช้งานเครื่องนั้นๆทันที สำหรับการเขียนโปรแกรมเพื่อตรวจสอบผู้ใช้ว่าผู้ใช้เข้าสู่ระบบเมื่อใดนั้น จำเป็นต้องมีแอพพลิเคชั่นที่ใช้สำหรับล็อคอินเอง กล่าวคือ ผู้ใช้จะต้องล็อคอินในแอพพลิเคชั่นนั่นเอง แต่โครงงานนี้ ต้องการที่จะใช้ส่วนประกอบของวินโดวส์ 2000 ซึ่งมีระบบล็อคอินอยู่แล้ว จึงทำให้ไม่สามารถตรวจสอบผู้ใช้ว่าผู้ใช้เข้าสู่ระบบเมื่อใดตามเวลาจริง โปรแกรมจะทำการตรวจสอบผู้ใช้ได้ก็ต่อเมื่อผู้ใช้เข้าสู่ระบบในระยะเวลาหนึ่ง ซึ่งอาจจะล่าช้าไปบ้าง แต่ก็ไม่ถึงกับเสียเวลามากมายนัก

ในส่วนของไอพีแอดเดรสนั้น วินโดวส์ 2000 ไม่มีฟังก์ชันที่จะสามารถรับค่าไอพีแอดเดรสจากเครื่องที่ผู้ใช้ใช้งาน เนื่องจากบนวินโดวส์ 2000 นั้นส่วนใหญ่จะใช้ชื่อผู้ใช้เป็นตัวอ้างอิงกับระบบ ดังนั้นจึงทำการตรวจสอบไอพีทุกไอพีภายในระบบ เพื่อให้ทราบว่าผู้ใช้ใดใช้เครื่องในไอพีใด หรือพูดอีกทางหนึ่งก็คือ โปรแกรมจะทำการตรวจสอบไอพีทุกไอพี โดยจะตรวจสอบว่าไอพีนั้นๆผู้ใช้ใดใช้งานอยู่บนระบบ แล้วจึงแจ้งให้กับทางไฟร์วอล ซึ่งการจะตรวจสอบว่าผู้ใช้ใดใช้งานอยู่บนไอพีนั้นๆ ก็จะทำได้โดยการใช้เน็ตเอพีไอ (NetAPI Function) ซึ่งเป็นฟังก์ชันบริการ (Service Function) สำหรับนักพัฒนาที่วินโดวส์ 2000 ได้จัดเตรียมไว้

6.2 การเรียกใช้เน็ตเอพีไอฟังก์ชั่น

เน็ตเอพีไอ (Network Management Function) ซึ่งเป็นฟังก์ชันที่ใช้สำหรับจัดการกับบัญชีผู้ใช้และทรัพยากรสำหรับเน็ตเวิร์กสำหรับแอพพลิเคชั่นที่พัฒนาด้วยภาษา C และ C++ ซึ่งจะมีข้อดี ดังนี้

- เน็ตเอพีไอ จะมีฟังก์ชันที่เอพีไอ อื่นๆไม่สามารถทำได้
- เน็ตเอพีไอใช้ได้ แม้ไม่มีแอคทีฟไดเรกทอรี

ในส่วนของโปรแกรมได้ใช้เน็ตเอพีไอเพื่อตรวจสอบว่าผู้ใช้ใดใช้งานอยู่บนไอพีแอดเดรสที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำลังตรวจสอบอยู่ สำหรับเน็ตเอพีไอ ที่ใช้นั้น ก็คือ NetWkstaUserEnum Function ซึ่งเป็นฟังก์ชันที่ใช้ตรวจสอบผู้ใช้ที่ใช้เครื่องนั้นๆอยู่ ซึ่งรายละเอียดของฟังก์ชันจะมี ดังนี้

```
NET_API_STATUS NetWkstaUserEnum(
  LPWSTR servername,
  DWORD level,
  LPBYTE *bufptr,
  DWORD prefmaxlen,
  LPDWORD entriesread,
  LPDWORD totalentries,
  LPDWORD resumehandle
);
```

servername

[in] Pointer to a string that specifies the DNS or NetBIOS name of the remote server on which the function is to execute. If this parameter is NULL, the local computer is used.

Windows NT 4.0 and earlier: This string must begin with \\.

level

[in] Specifies the information level of the data. This parameter can be one of the following values.

Value	Meaning
0	Return the names of users currently logged on to the workstation. The bufptr parameter points to an array of WKSTA_user_INFO_0 structures.
1	Return the names of the current users and the domains accessed by the workstation. The bufptr parameter points to an array of WKSTA_user_INFO_1 structures.

bufptr

[out] Pointer to the buffer that receives the data. The format of this data depends on the value of the level parameter. This buffer is allocated by the system and must be freed using the NetApiBufferFree function. Note that you must free the buffer even if the function fails with ERROR_MORE_DATA.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

prefmaxlen

[in] Specifies the preferred maximum length of returned data, in bytes. If you specify MAX_PREFERRED_LENGTH, the function allocates the amount of memory required for the data. If you specify another value in this parameter, it can restrict the number of bytes that the function returns. If the buffer size is insufficient to hold all entries, the function returns ERROR_MORE_DATA. For more information, see Network Management Function Buffers and Network Management Function Buffer Lengths.

entriesread

[out] Pointer to a value that receives the count of elements actually enumerated.

totalentries

[out] Pointer to a value that receives the total number of entries that could have been enumerated from the current resume position.

resumehandle

[in/out] Pointer to a value that contains a resume handle which is used to continue an existing search. The handle should be zero on the first call and left unchanged for subsequent calls. If resumehandle is NULL, no resume handle is stored.

Return Values

If the function succeeds, the return value is NERR_Success.

If the function fails, the return value can be one of the following error codes.

Value	Meaning
ERROR_ACCESS_DENIED	The user does not have access to the requested information.
ERROR_MORE_DATA	More entries are available. Specify a large enough buffer to receive all entries.
ERROR_INVALID_LEVEL	The level parameter is invalid.

รูปที่ 6-1 แสดงรายละเอียดของเน็ตเอพีไอ ที่เรียกใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป 6-1 จะแสดงรายละเอียดของฟังก์ชันเอพีไอที่เรียกใช้งานเพื่อตรวจสอบผู้ใช้ในแต่ละเครื่อง ซึ่งฟังก์ชันนี้ จะอยู่ในไฟล์ที่มีชื่อว่า getuserc.exe โดยจะทำการเอ็กซ์คิวไฟล์จากโปรแกรมหลักของแอพพลิเคชัน ด้วยคำสั่ง _spawnl ซึ่งจะเป็นคำสั่งที่ใช้ในการเอ็กซ์คิวไฟล์ โดยในส่วนของโปรแกรม getuserc.exe นั้น เมื่อทำการตรวจสอบผู้ใช้ของแต่ละเครื่องแล้ว จะใส่ชื่อผู้ใช้ลงในเท็กซ์ไฟล์ (Text File) ที่มีชื่อว่า user.txt

เมื่อโปรแกรมหลักทำการเอ็กซ์คิวด้วยคำสั่ง _spawnl แล้ว ก็จะทำการอ่านค่าจากไฟล์ user.txt แล้วจึงนำชื่อผู้ใช้มาประมวลผลต่อไป ดังแสดง

```
#ifndef UNICODE
#define UNICODE
#endif
#include <stdio.h>
#include <assert.h>
#include<fstream.h>
#include <windows.h>
#include <lm.h>
#include <process.h>
#pragma comment(lib,"netapi32")
#include<wchar.h>
#include <stdarg.h>

int wmain(int argc, wchar_t *argv[])
{
  LPWKSTA_USER_INFO_0 pBuf = NULL;
  LPWKSTA_USER_INFO_0 pTmpBuf=NULL;
  DWORD dwLevel = 0;
  DWORD dwPrefMaxLen = -1;
  DWORD dwEntriesRead = 0;
  DWORD dwTotalEntries = 0;
  DWORD dwResumeHandle = 0;
  DWORD i;
  DWORD dwTotalCount = 0;
  NET_API_STATUS nStatus;
  LPTSTR pszServerName = NULL;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 FILE* stream;
if (argc > 2)
{
  fwprintf(stderr, L"Usage: %s [\\\\ServerName]\n", argv[0]);
  exit(1);
}
if (argc == 2)
  pszServerName = argv[1];
do // begin do
{
  nStatus = NetWkstaUserEnum(pszServerName,
                    dwLevel,
                    (LPBYTE*)&pBuf,
                    dwPrefMaxLen,
                    &dwEntriesRead,
                    &dwTotalEntries,
                    &dwResumeHandle);

       if ((nStatus == NERR_Success) || (nStatus == ERROR_MORE_DATA))
  {
    if ((pTmpBuf = pBuf) != NULL)
    {
      for (i = 0; (i < dwEntriesRead); i++)
        {
        assert(pTmpBuf != NULL);

        if (pTmpBuf == NULL)
        {

          fprintf(stderr, "An access violation has occurred\n");
          break;
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                pTmpBuf++;
                dwTotalCount++;
              }

              pTmpBuf--;
              stream = fopen( "vfwprintf.txt", "w" );
              fprintf( stream,"%S",pTmpBuf->wkui0_username);
              fclose( stream );

        }
        else
        {
              stream = fopen( "vfwprintf.txt", "w" );
              fprintf( stream,"%S","");
              fclose( stream );
              exit(1);
        }
}
 else
     {
              stream = fopen( "vfwprintf.txt", "w" );
              fprintf( stream,"%S","");
              fclose( stream );
              exit(1);
     }

if (pBuf != NULL)
{
  NetApiBufferFree(pBuf);
  pBuf = NULL;
}

}while (nStatus == ERROR_MORE_DATA); // end do
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

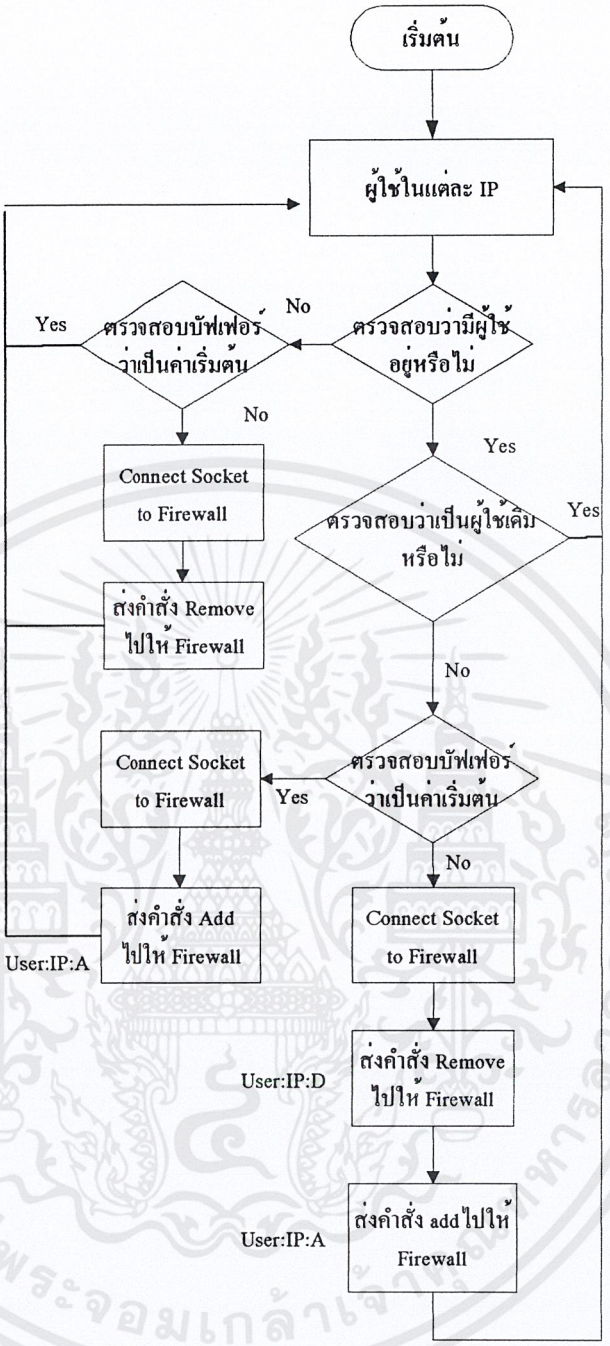
```
  if (pBuf != NULL)
    NetApiBufferFree(pBuf);
  return 0;
}
```

รูป 6-2 แสดงโค้ดของไฟล์ userc.exe

6.3 การตรวจสอบการเข้าออกของผู้ใช้

สำหรับการตรวจสอบการเข้าออกของผู้ใช้นั้น เราได้ใช้ฟังก์ชั่น NetAPI ดังที่ได้กล่าวมาในหัวข้อที่ผ่านมา โดยจะทำการตรวจสอบว่าผู้ใช้ใดใช้งานอยู่บนเครื่องใด แต่ในการตรวจสอบว่าผู้ใช้ใดออกจากระบบนั้น จะใช้ฟังก์ชันนี้เช่นกันในการตรวจสอบ โดยโปรแกรมเมื่อทำการตรวจสอบถึงไอพีใด ก็จะตรวจสอบว่าผู้ใช้ที่ใช้งานอยู่ยังเป็นผู้ใช้เดิมหรือไม่ หากเป็นผู้ใช้เดิมก็จะข้ามไปตรวจสอบไอพีอื่น แต่หากผู้ใช้เกิดการเปลี่ยนแปลงขึ้น ก็จะทำการส่งข้อมูลของผู้ใช้เก่า ซึ่งถูกเก็บในบัฟเฟอร์ให้กับไฟร์วอลว่า ผู้ใช้เก่าได้ออกจากระบบไปแล้ว จากนั้นจะนำผู้ใช้ใหม่ส่งให้กับไฟร์วอลทันทีและเก็บผู้ใช้ใหม่ในบัฟเฟอร์แทนที่ผู้ใช้เก่านั้น ดังรูป 6-3 จะเป็นโฟล์วชาร์ต (Flow Chart) ของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-3 แสดงโฟล์วชาร์ต (Flow Chart) ของการทำงานในส่วนการตรวจสอบการเข้าออกของผู้ใช้

6.4 การติดต่อกับไฟร์วอลผ่านซ็อกเก็ต (Socket)

ในการที่จะทำให้ไฟร์วอลทราบว่าผู้ใช้ใดเข้าสู่ระบบและได้ใช้เครื่องใดนั้น แอพพลิเคชั่นฝั่งวินโดวส์ 2000 จะต้องทำการส่งข้อมูลไปให้ ซึ่งบนฝั่งไฟร์วอลเองก็จะมีโมดูลสำหรับรับข้อมูลนั้นอยู่แล้ว โดยการจะส่งไปได้นั้น ก็จะใช้การส่งข้อมูลผ่านทางซ็อกเก็ต ซึ่งการเขียนโปรแกรมผ่านซ็อกเก็ตนั้น เป็นการเขียนโปรแกรมทางเน็ตเวิร์ก ที่ใช้กันอย่างแพร่หลาย ดังนั้นโครงงานนี้จึงได้เลือกการเขียนโปรแกรมด้วยวิธีนี้มาพัฒนาแอพพลิเคชั่นนี้ขึ้น ซึ่งในการเขียนโปรแกรมผ่านทางซ็อกเก็ตนั้นจะต้องอ้างอิงกับพอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้วยว่าใช้พอร์ตใดในการติดต่อ ซึ่งในโปรแกรมนี้ก็จะใช้งานพอร์ท 1170 ซึ่งเป็นพอร์ตที่ไม่ได้ถูกกำหนดให้ใช้งานอย่างตายตัว มาใช้ในการติดต่อกันระหว่างไฟร์วอลและแอพพลิเคชั่นบนวินโดวส์ 2000

6.4.1 การเขียนโปรแกรมผ่านทางซ็อกเก็ต

ในการเขียนโปรแกรมทางเน็ตเวิร์ก การเขียนโปรแกรมโดยใช้ซ็อกเก็ตนั้น เป็นที่นิยมอย่างแพร่หลาย และถือเป็นพื้นฐานของการเขียนโปรแกรมทางเน็ตเวิร์ก

สำหรับการเขียนโปรแกรมโดยใช้ซ็อกเก็ตบนวินโดวส์ 2000 นั้น จะคล้ายๆกับการเขียนโปรแกรมผ่านทางซ็อกเก็ตบนลีนุกซ์ โดยจะแตกต่างกันเพียง ในการเขียนโปรแกรมบนวินโดวส์นั้นจะต้องเรียกใช้คำสั่ง WSAStartup เพื่อเป็นการเริ่มต้นการเรียกใช้งาน DLL บนวินโดวส์ด้วย โดยมีรูปแบบดังนี้

```
Int WSAStartup(
        Word            wVersionRequired,
        LPWSADATA   lpWSAData);
```

- wVersionRequired จะเป็นค่าของเวอร์ชัน DLL ที่เรียกใช้
- LPWSADATA จะเป็นพอยน์เตอร์ไปยัง WSADATA Structure ซึ่งจะเป็นส่วนที่แสดงรายละเอียดของ DLL

เมื่อทำการกำหนดค่าเริ่มต้นให้กับ DLL แล้ว ก็จะต้อง เปิดซ็อกเก็ต (Open Socket) สำหรับใช้ในการติดต่อ โดยจะมีรูปแบบคำสั่งดังนี้

```
Typedef unsigned int SOCKET;
SOCKET PASCAL FAR socket(int af,int type, int protocol);
```

- af แสดงค่าของ แอดเดรสแฟมิลี่
- type ชนิดการส่งข้อมูลเป็น Datagram หรือ Stream
- protocol ชนิดของโพรโตคอล เป็น TCP หรือ UDP

เมื่อทำการเปิดซ็อกเก็ตแล้ว ก็จะทำการ listen หรือ connect ไป แล้วแต่การติดต่อ หากเป็นฝั่งเซิร์ฟเวอร์ ก็จะทำการ listen และเมื่อมีการ connect จากฝั่งไคลเอ้นท์ก็จะ accept หากเป็น ฝั่งไคลเอ้นท์ร้องขอการติดต่อก็จะ connect

รูปแบบคำสั่ง listen

```
Int PASCAL listen (SOCKET s,int backlog);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบคำสั่ง accept

SOCKET PASCAL accept(SOCKET s, LPSOCKADDR lpAddr , LPINT lpLength);

รูปแบบคำสั่ง connect

Int PASCAL connect(SOCKET s,const struct sockaddr *name, int namelen);

เมื่อทำการติดต่อกันระหว่างเครื่องไคลเอ้นท์กับเซิร์ฟเวอร์แล้ว ก็จะทำการส่งข้อมูลกันไปมาระหว่างไคลเอ้นท์กับเซิร์ฟเวอร์โดยผ่านทางซ็อกเก็ตที่ได้กำหนดไว้

รูปแบบคำสั่ง send

Int PASCAL send(SOCKET s, LPCSTR buf, int len, int flags);

รูปแบบคำสั่ง receive

Int PASCAL recv(SOCKET s, LPSTR buf , int len , int flags);

เมื่อทำการติดต่อรับส่งข้อมูลกันเสร็จเรียบร้อย ก็ควรจะปิดซ็อกเก็ตนั้นทิ้งไป เพื่อป้องกันการจู่โจมจากผู้ไม่ประสงค์ดี ด้วยคำสั่ง ดังนี้

รูปแบบคำสั่ง close

Int PASCAL closesocket(SOCKET s);

ดังที่กล่าวมาแล้วว่าในการเขียนโปรแกรมบนวินโดวส์นั้น จะต้องใช้ DLL และในตอนเริ่มต้นได้มีการกำหนดค่า DLL ที่ใช้งาน ดังนั้นเมื่อเลิกใช้งานซ็อกเก็ตก็จะต้องปิด DLL ที่ใช้ด้วย ด้วยคำสั่ง WSACleanUp();

ตัวอย่างการเขียนโปรแกรม ทางเน็ตเวิร์กโดยผ่านทางซ็อกเก็ต

```
WSADATA wsData;
  WORD    wVer = MAKEWORD(2,2);
        SOCKET minix;
        char    *szName=MYHOST;
        int status;
        struct hostent *he;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
WCHAR   wszRecvBuf[100];
struct sockaddr_in their_addr;
memset(wszRecvBuf,0,sizeof(wszRecvBuf));
status = WSAStartup(wVer,&wsData);
minix=socket(AF_INET,SOCK_STREAM,0);
if (minix == INVALID_SOCKET)
{
        printf("Failed to create socket: %d\n",WSAGetLastError());
        WSACleanup();
        return -1;
}
bind(minix,(const sockaddr *)&their_addr,sizeof(their_addr));
their_addr.sin_family=AF_INET;
their_addr.sin_port=htons(MYPORT);
int len=0;
gethostname(szName,len);
he = gethostbyname(szName);
their_addr.sin_addr.S_un.S_addr = *((long *)(he->h_addr));
memset(&(their_addr.sin_zero),'\0',8);
status= connect(minix,(struct sockaddr*) &their_addr,sizeof(struct sockaddr)) ;
if(status==-1)
{
        printf("\nfail to connect");
        WSAGetLastError();
        WSACleanup();
        return -1;
}
char* msg2 = (char*) malloc (sizeof(char*));

status = send(minix,(char*)msg,100,0);
printf("\nsend %s:",msg);
if(status<0)
{
        printf("\nfail to send");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        WSAGetLastError();
        WSACleanup();
    }
```

6.4.2 รูปแบบข้อมูลที่ใช้ในการติดต่อระหว่างแอพพลิเคชันกับไฟร์วอล

ในการส่งข้อมูลนั้น เมื่อโปรแกรมตรวจสอบพบว่าผู้ใช้เข้าสู่ระบบบนไอพีใด ก็จะนำ ชื่อผู้ใช้มารวมกับไอพีแอดเดรสโดยขั้นด้วยเครื่องหมาย ':' จากนั้นก็จะส่งให้กับไฟร์วอล แต่การที่จะบอกไฟร์วอลว่าข้อมูลที่ส่งไปนั้นเป็นการเข้าหรือออกของผู้ใช้นั้น โปรแกรมก็จะทำการรวมตัวอักษรไว้ต่อท้ายเพื่อให้ไฟร์วอลทราบว่าเป็นการเข้าหรือการออกของผู้ใช้ ซึ่งหากต่อท้ายด้วย A ก็จะหมายถึงผู้ใช้เข้าสู่ระบบ หากต่อท้ายด้วยตัว D ก็จะหมายถึงผู้ใช้ออกจากระบบนั่นเอง เช่น

userA:192.168.0.5D	หมายถึง	ผู้ใช้ชื่อ userA ออกจากระบบบนไอพี 192.168.0.5
userB:192.168.0.6A	หมายถึง	ผู้ใช้ชื่อ userB เข้าสู่ระบบบนไอพี 192.168.0.6

ซึ่งข้อมูลที่ถูกส่งไปนี้ จะอยู่รูปแบบสตริง ซึ่งแอพพลิเคชั่นบนฝั่งไฟร์วอลจะทำการตัดคำออกมาโดยใช้เครื่องหมาย ':' นี้เพื่อแยกระหว่างชื่อผู้ใช้กับไอพีแอดเดรส และสุดท้าย ตัว D หรือ A ออกมาเพื่อทำการเพิ่มโพรเซสหรือลบโพรเซสนั่นเอง

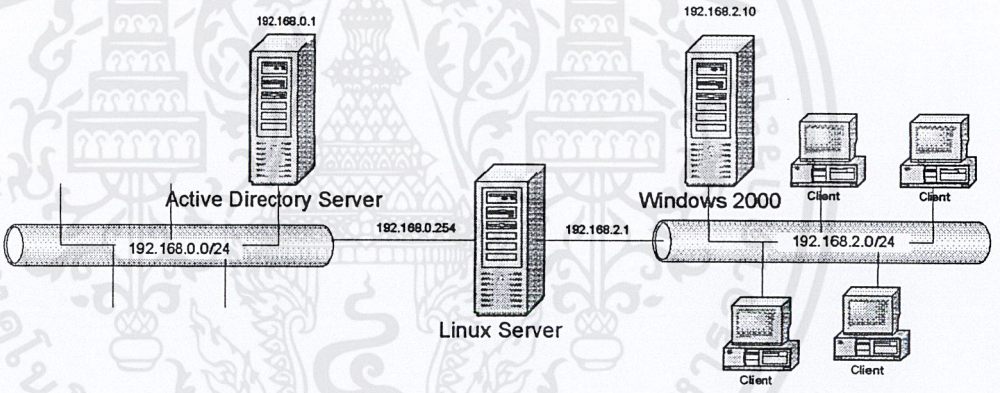
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
การทดสอบไฟร์วอลแบบอิงบริการไดเรกทอรี

สำหรับการทดสอบไฟร์วอลแบบอิงบริการไดเรกทอรีนั้น เนื่องจากว่าทางผู้พัฒนาโครงงานไม่มีงบในการติดตั้งระบบเครือข่ายจริงขึ้น จึงได้ใช้โปรแกรมเครื่องเสมือน (Virtual Machine- VMWare) มาช่วยในการทดสอบระบบ

ระบบเครือข่ายที่ใช้ในการทดสอบ จะจัดให้ เครือข่ายไคลเอนท์มีซับเน็ต (Subnet) เป็น 192.168.2.0 แอคทีฟไดเรกทอรีมีหมายเลขไอพีเป็น 192.168.0.1 วินโดวส์ 2000 เซิร์ฟเวอร์จะทำการติดตั้งโปรแกรมที่ได้พัฒนาบนส่วนของวินโดวส์ 2000 ลงไปโดยมีหมายเลขไอพีเป็น 192.168.2.10 และ ไฟร์วอล จะมีอินเทอร์เฟส 2 อินเทอร์เฟส คือ

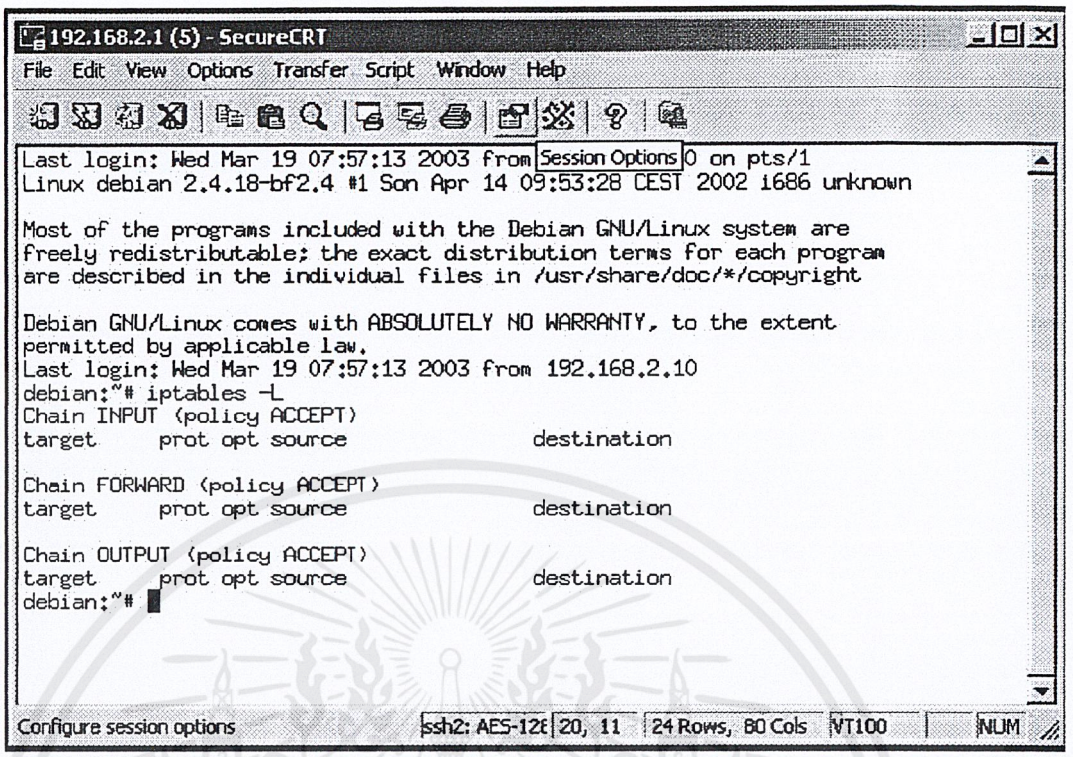
1. อินเทอร์เฟสฝั่งไคลเอนท์ มีไอพีเป็น 192.168.2.1
2. อินเทอร์เฟสฝั่งแอคทีฟไดเรกทอรีมีไอพีเป็น 192.168.0.254



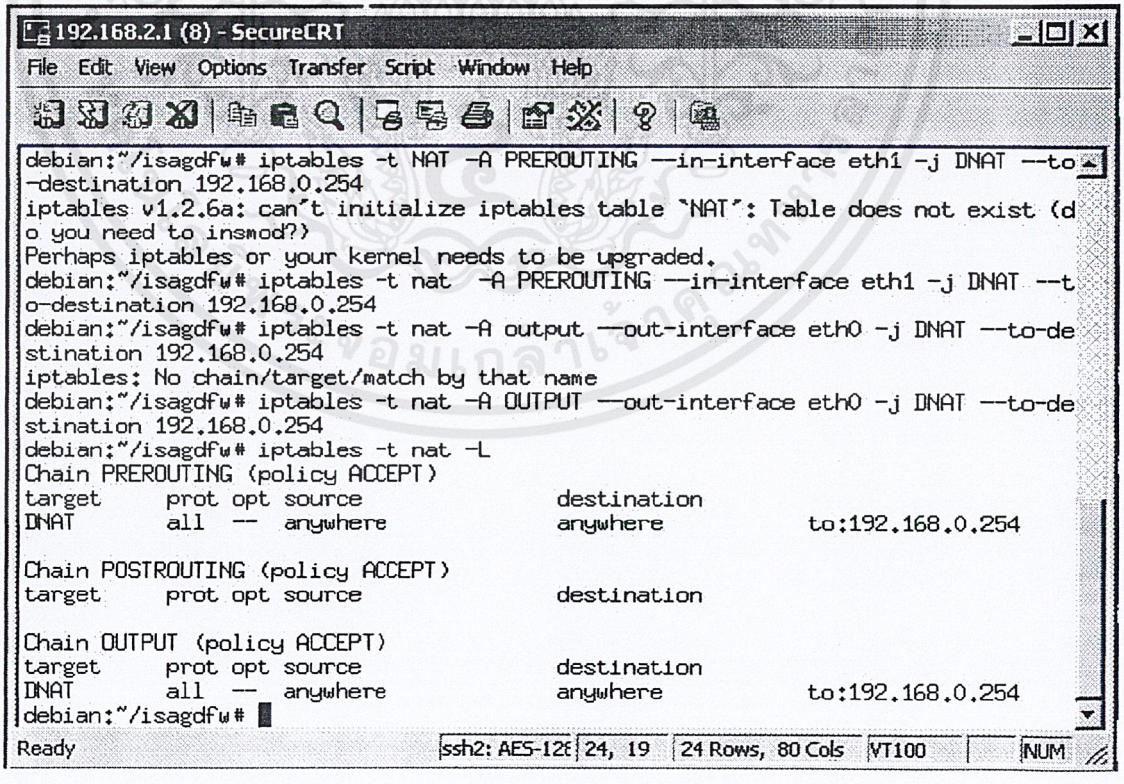
รูปที่ 7-1 ระบบเครือข่ายที่ใช้ในการทดสอบ

เริ่มต้นเราจะทำการเซ็ทค่าไอพีเทเบิลบนฝั่งลีนุกซ์ก่อน โดยจะได้ไอพีเทเบิลที่ยังไม่ได้เพิ่มกฎใดๆ ลงไปเลย ดังรูป 7-2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



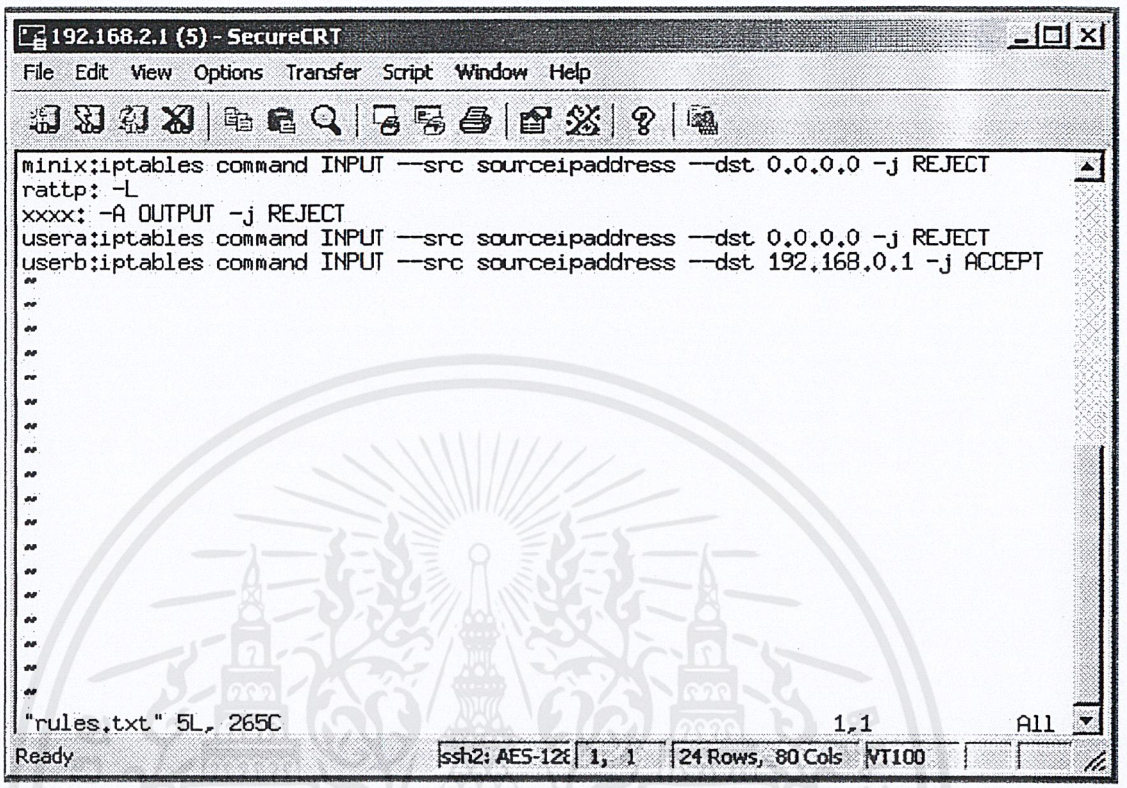
รูปที่ 7-2 ไอพีเทเบิลส่วนอินพุท เอาท์พุท และฟอร์เวิร์ด เริ่มต้น



รูปที่ 7-3 ไอพีเทเบิลส่วน พรีเราท์ติ้ง โพสต์เราท์ติ้งและเอาท์พุทเริ่มต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

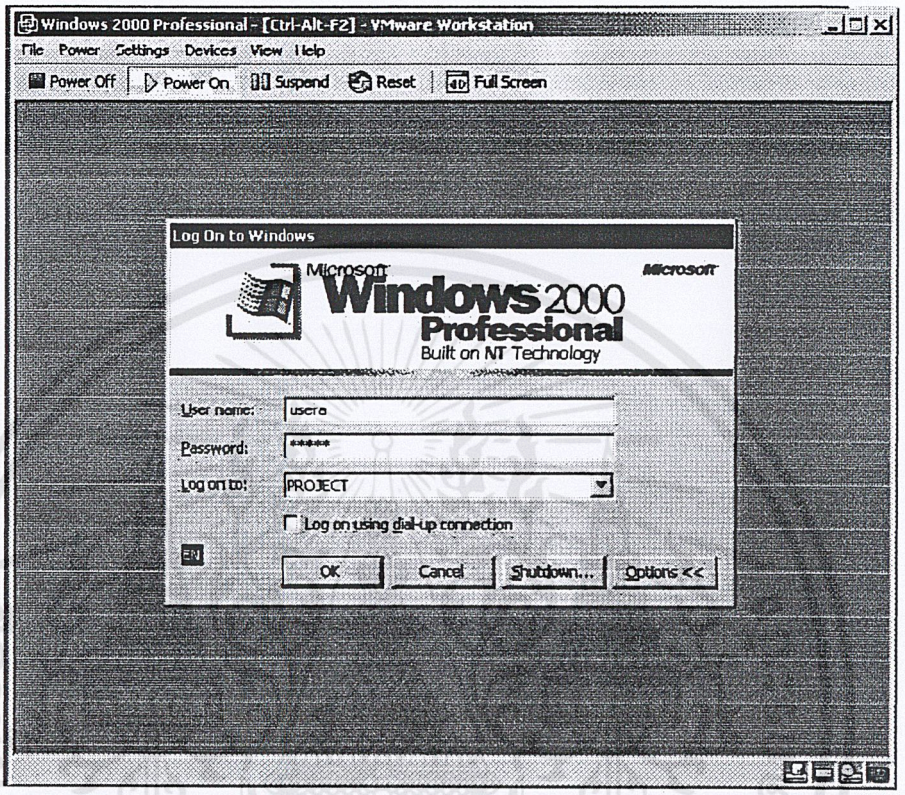
โดยกฎของผู้ใช้ จะอยู่ใน rules.txt ดังรูป 7-4



รูปที่ 7-4 กฎของผู้ใช้บนลีนุกซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

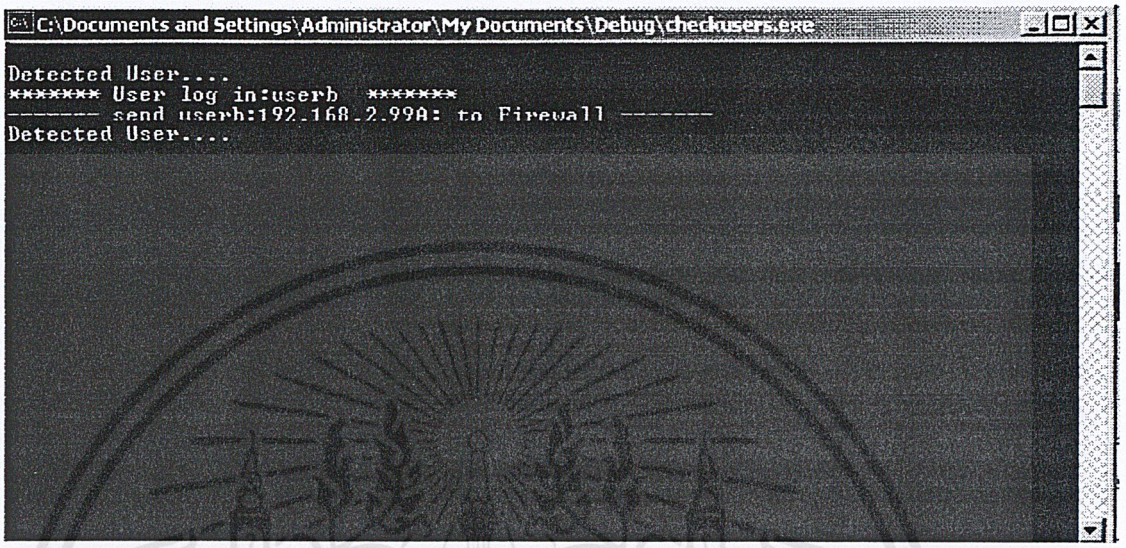
จากนั้นทำการบูต (Boot) เครื่องจาก VMWare ขึ้นมาเพื่อใช้เป็นไคลเอนท์ แล้วทำการรันโปรแกรมทั้งบนฝั่งวินโดวส์ 2000 และบนฝั่งลีนุกซ์ จากนั้นทำการล็อคอินผู้ใช้ โดยจะล็อคอินเป็น Usera ดังรูป 7-5



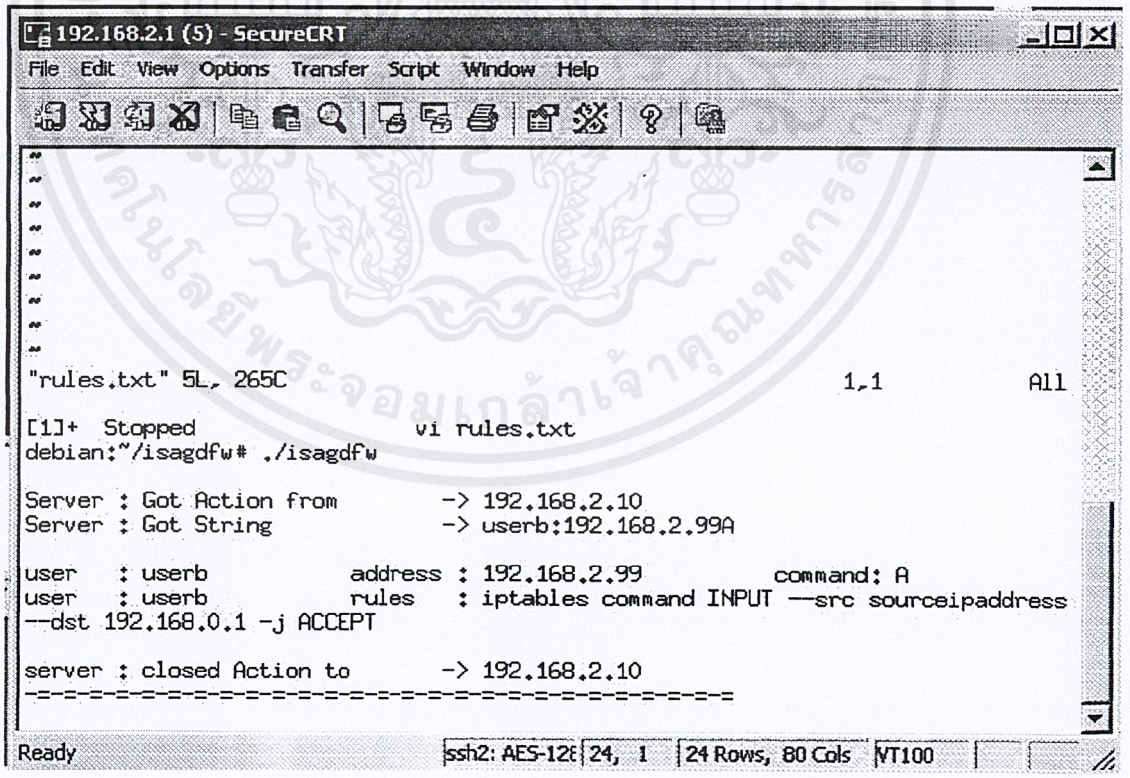
รูปที่ 7-5 ล็อคอินผู้ใช้เป็น usera

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อล็อคอินผู้ใช้แล้ว โปรแกรมบนฝั่งวินโดวส์ 2000 จะพบผู้ใช้ usera จากนั้นทำการส่งชื่อผู้ใช้ หมายเลขไอพี และแอคชั่น A เพื่อทำการเพิ่มกฎให้กับลีนุกซ์ ดังรูป 7-6 ส่วนเดมอนบนฝั่งลีนุกซ์จะรับข้อมูลที่ส่งมาจากวินโดวส์ 2000 เซิร์ฟเวอร์ แล้วทำการเพิ่มกฎของ usera ลงในไอพีเทเบิล ดังรูป 7-7



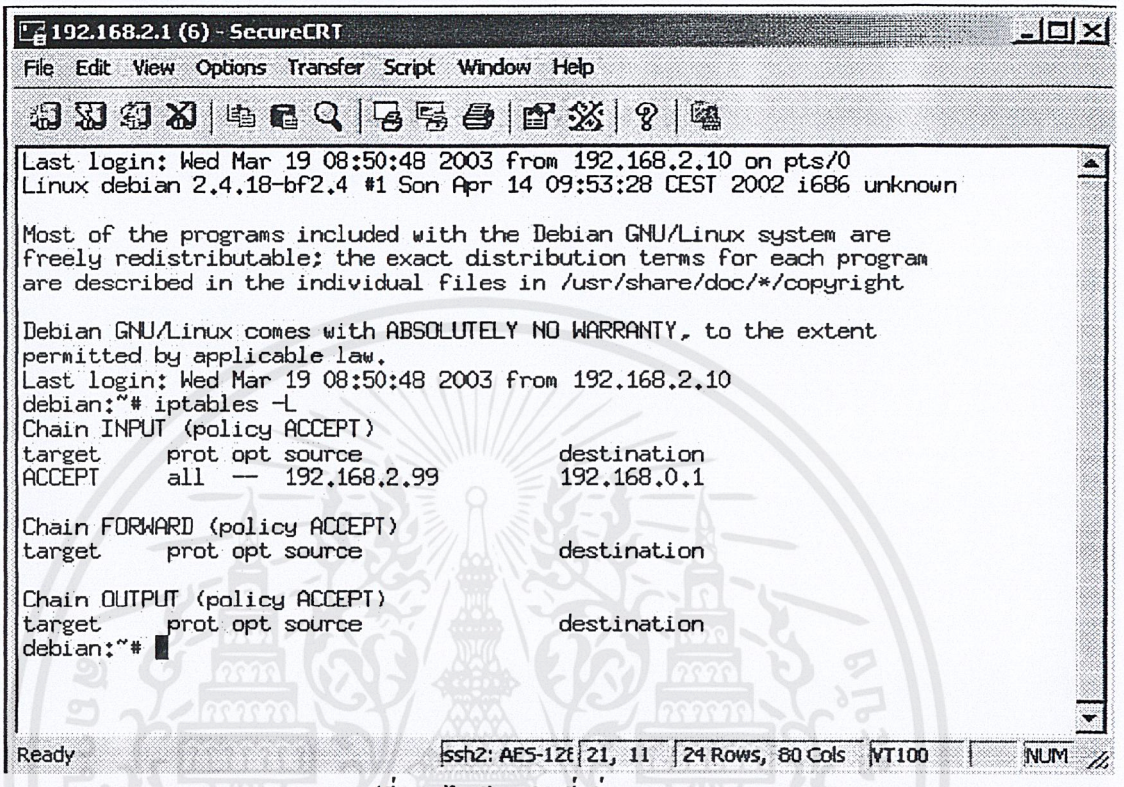
รูปที่ 7-6 โปรแกรมบนฝั่งวินโดวส์ 2000 พบผู้ใช้ usera และส่งข้อมูลไปให้ลีนุกซ์



รูปที่ 7-7 เดมอนบนลีนุกซ์รับข้อมูลจากวินโดวส์ 2000 เซิร์ฟเวอร์ และเพิ่มกฎลงในไอพีเทเบิล

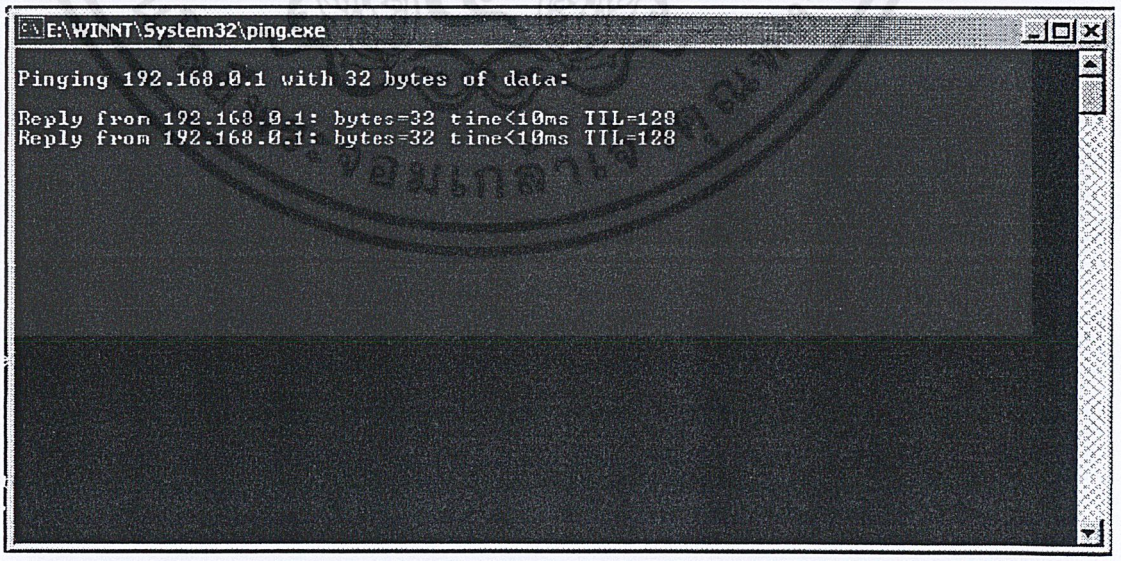
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเดมอนบนลีนุกซ์ทำการเพิ่มกฎลงในไอพีเทเบิลแล้ว ทำการตรวจไอพีเทเบิลดู จะได้ ไอพีเทเบิลที่มีกฎของ usera เพิ่มขึ้นมา ดังรูป 7-8



รูปที่ 7-8 ไอพีเทเบิลที่เพิ่มกฎของ usera

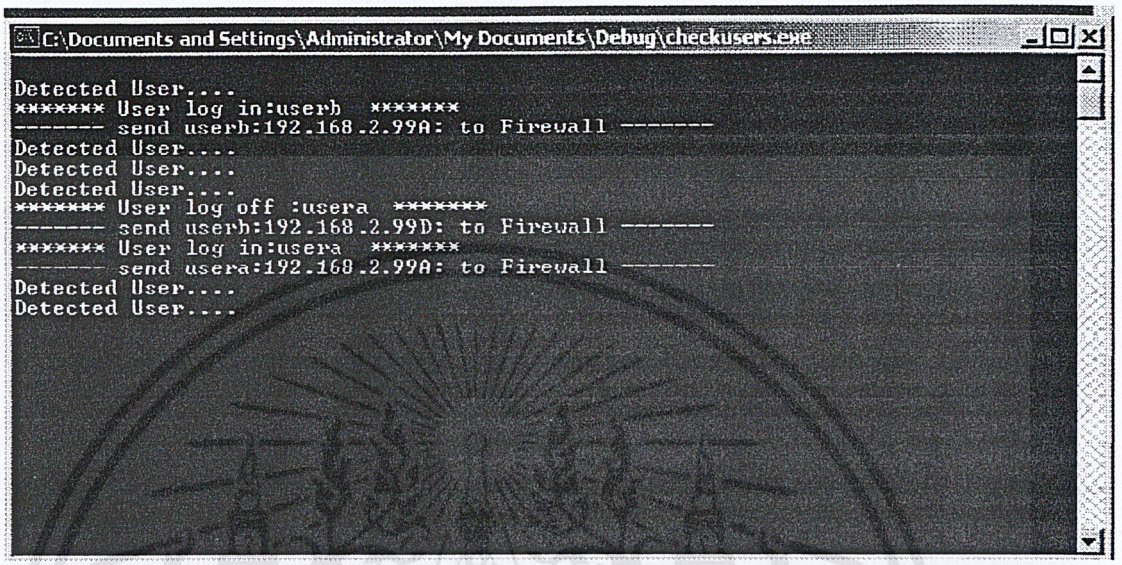
จากรูปจะเห็นว่า userb นั้นจะมีกฎให้ผ่านแพ็กเกจไปยัง 192.168.0.1 ได้ ลองทดสอบด้วยการสั่งปิง (Ping) ไปที่ 192.168.0.1 จะได้ผลดังรูป 7-9



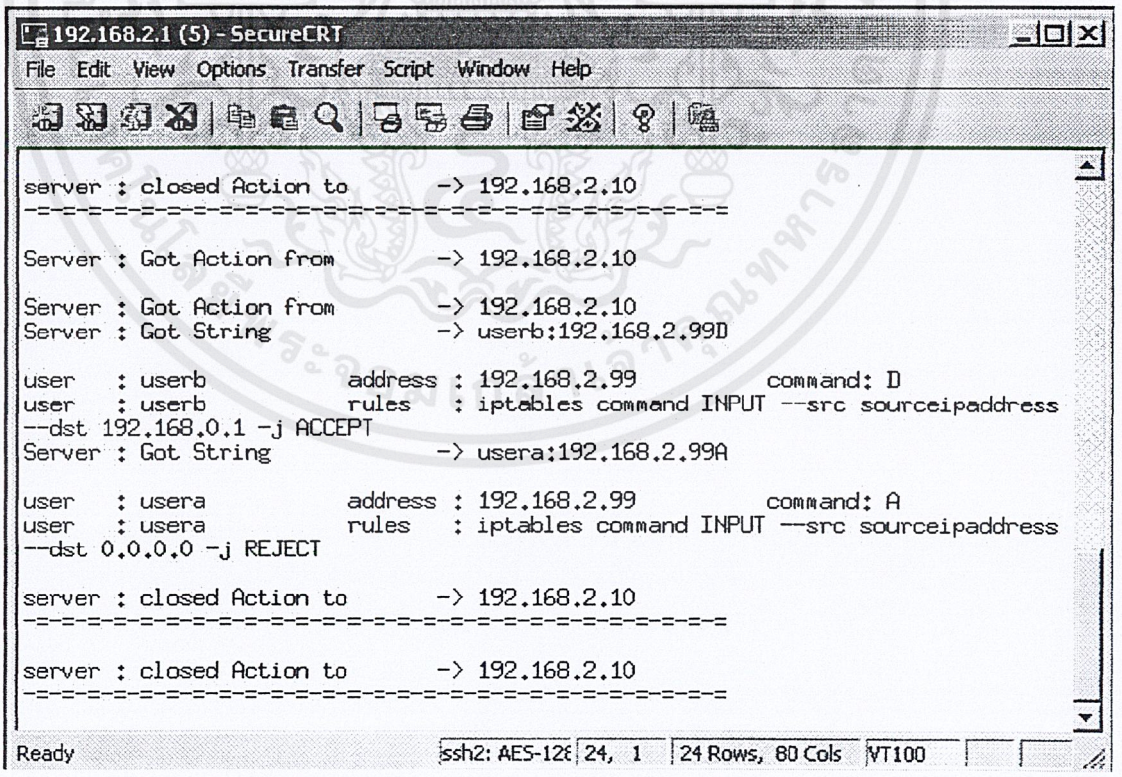
รูปที่ 7-9 ผลการปิงไปยัง 192.168.0.1 ของผู้ใช้ userb

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการเปลี่ยนผู้ใช้ โดยให้ผู้ใช้ใหม่ เป็น usera โปรแกรมบนฝั่งวินโดวส์ 2000 จะทำการส่งข้อมูลของผู้ใช้ userb ไปยังลีนุกซ์ เพื่อทำการลบกฎของไอพีเทเบิล และส่งข้อมูลของผู้ใช้ usera ไปยังลีนุกซ์เพื่อทำการเพิ่มกฎของผู้ใช้ ดังรูป 7-10 ส่วนเคมอนบนลีนุกซ์จะได้รับข้อมูลเช่นกัน ดังรูป 7-11



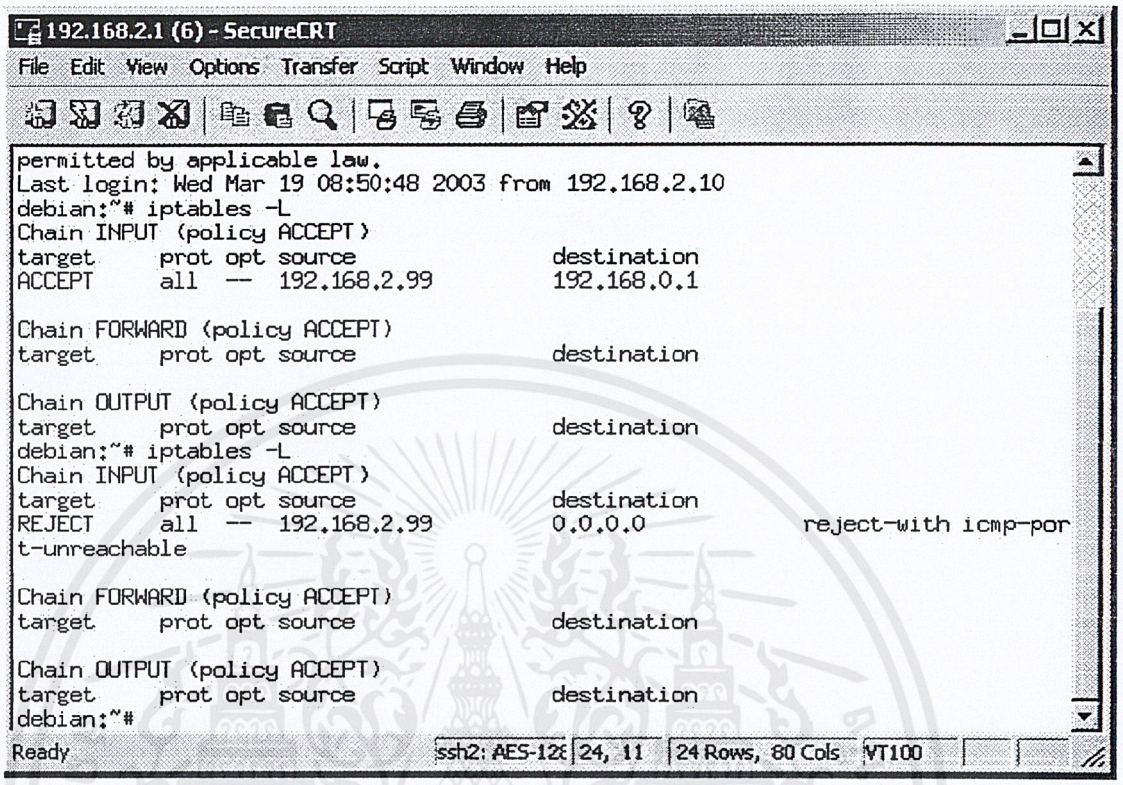
รูปที่ 7-10 โปรแกรมบนฝั่งวินโดวส์ 2000 เมื่อมีผู้ใช้คนใหม่เข้ามาแทน



รูปที่ 7-11 เคมอนบนฝั่งลีนุกซ์ได้รับข้อมูลจากวินโดวส์ 2000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

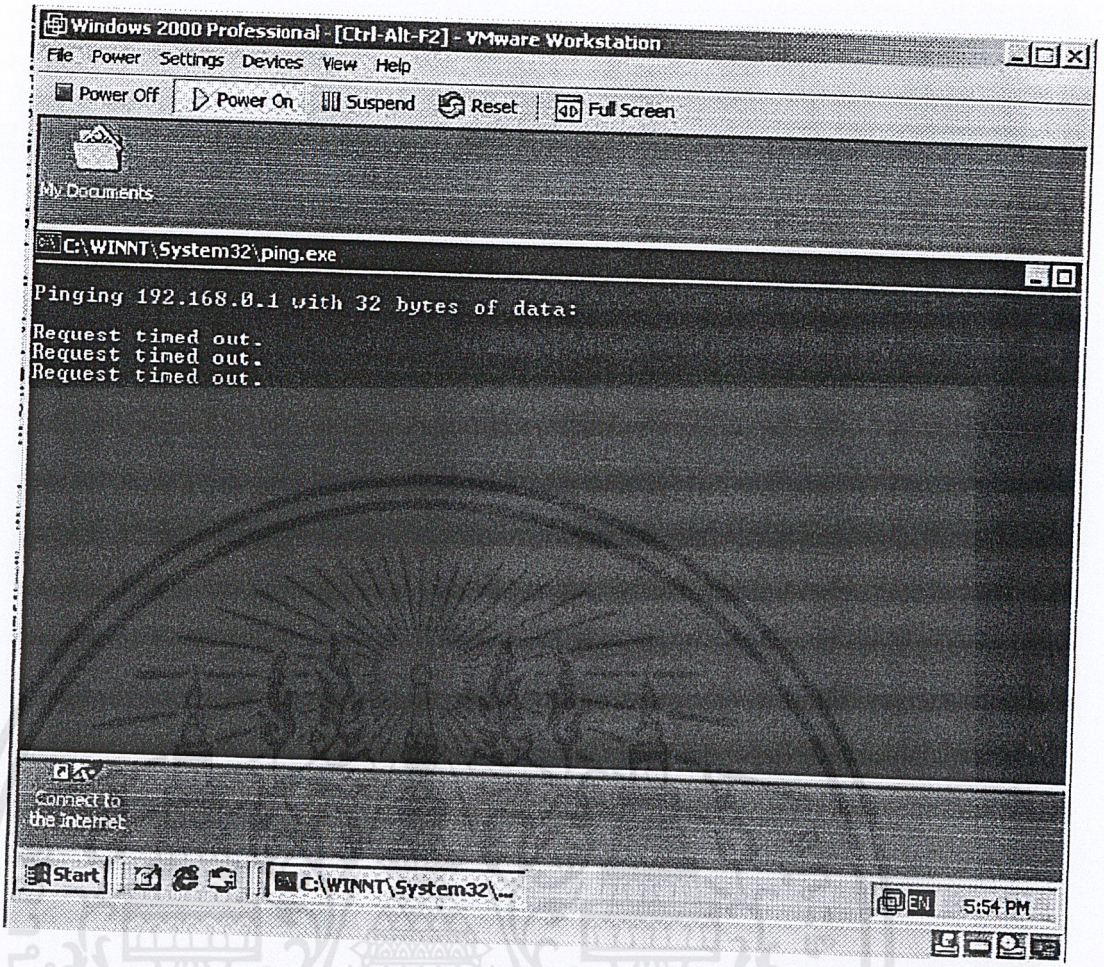
ทำการตรวจสอบไอพีเทเบิล จะได้ไอพีเทเบิลดังรูป 7-12



รูปที่ 7-12 ไอพีเทเบิลเมื่อทำการเพิ่มกฎของผู้ใช้ usera

จะเห็นว่ากฎของ usera นั้น จะถูกปฏิเสธ ลองทดสอบด้วยการปิงไปยัง 192.168.0.1 จะได้ผลดังรูป 7-13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7-13 ผลของการปิงไปยัง 192.168.0.1 ของ usera

จากผลการทดสอบ จะเห็นว่า ผู้ใช้นั้นจะถูกเพิ่มกฎ และการทำงานของผู้ใช้จะเป็นไปตามกฎที่ได้ถูกกำหนดไว้แต่แรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8
บทวิจารณ์และสรุปผล

8.1 สรุปผลงาน

จากการพัฒนาโครงงานไฟร์วอลแบบอิงบริการไคเรกทอรี สรุปผลงานได้ดังนี้

8.1.1 ได้มีการใช้ไฟร์วอลโดยใช้ไอพีเทเบิลซึ่งเป็นโมดูลบนระบบปฏิบัติการลีนุกซ์มาปรับปรุงให้เป็นไฟร์วอลที่อิงกับผู้ใช้

8.1.2 ได้มีการนำการทำงานของระบบล็อคอินบนระบบปฏิบัติการวินโดวส์ 2000 มาใช้งานเพื่อทำการล็อคอินผู้ใช้เข้าสู่ระบบโดยที่ผู้พัฒนาไม่จำเป็นต้องพัฒนาส่วนล็อคอินขึ้นมาใหม่ รวมทั้ง แอคทีฟไดเรกทอรี ซึ่งเป็นเซอร์วิสคอมโพเนนต์ (Service Component) ของวินโดวส์ 2000 มาใช้งานโดยไม่จำเป็นต้องปรับปรุงเปลี่ยนแปลงใดๆทั้งสิ้น

8.1.3 ได้มีการนำเน็ตเอพีไอมาใช้งานโดยทำการเขียนโปรแกรมด้วยภาษา C++

8.1.4 พัฒนาแอพพลิเคชั่นบนฝั่งไฟร์วอลซึ่งใช้ระบบปฏิบัติการลีนุกซ์ขึ้นมาใหม่ ด้วยภาษา C เพื่อใช้งานแอพพลิเคชั่นนี้ร่วมกับไอพีเทเบิลบนระบบปฏิบัติการลีนุกซ์ ขึ้นมาเป็นไฟร์วอล

8.1.5 พัฒนาแอพพลิเคชั่นบนระบบปฏิบัติการวินโดวส์ 2000 เพื่อใช้งานร่วมกับไฟร์วอลที่พัฒนาขึ้น โดยทำหน้าที่ตรวจสอบการเข้าออกของผู้ใช้เป็นหลัก

8.2 ประโยชน์จากโครงงาน

8.2.1 ทำให้เกิดความรู้ทางด้านไฟร์วอล ซึ่งเป็นสิ่งสำคัญสำหรับการรักษาความปลอดภัยของระบบเน็ตเวิร์ก

8.2.2 ทำให้ทราบถึงการพัฒนาไฟร์วอลโดยใช้ไอพีเทเบิลบนระบบปฏิบัติการลีนุกซ์

8.2.3 ทำให้ทราบถึงแอคทีฟไดเรกทอรีอย่างคร่าวๆ

8.2.4 ได้รับไฟร์วอลที่อิงกับผู้ใช้ โดยใช้งานร่วมกับแอคทีฟไดเรกทอรี บนระบบปฏิบัติการวินโดวส์ 2000 ซึ่งผู้ใช้สามารถเข้าสู่ระบบได้ด้วยเครื่องใดก็ได้

8.3 ปัญหาที่พบจากโครงงาน

ในส่วนของลีนุกซ์การพัฒนาช่วงแรกยังมีปัญหาในการ Execute External command เนื่องจากในการเอ็กซ์คิวคอมมานด์ (execute command) นั้น เมื่อเอ็กซ์คิว เสร็จโปรแกรมจะทำการเทอร์มิเนต (Terminate) ตัวเองทันที ทำให้ไม่สามารถทำงานต่อได้ แต่ก็สามารถแก้ปัญหาด้วยการฟอล์คโพรเซสขึ้นมาใหม่เพื่อรองรับการเอ็กซ์คิวคอมมานด์ทำให้ปัญหาหมดไป

ในส่วนของวินโดวส์ 2000 นั้นจะพบปัญหาในการเปลี่ยนผู้ใช้ ซึ่งฟังก์ชันเอพีไอที่ใช้ในการตรวจสอบผู้ใช้นั้น จะจดจำผู้ใช้เก่า และส่งค่ามาให้เสมอ แต่แก้ปัญหาด้วยการรับผู้ใช้สุดท้ายที่ส่งเข้ามา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนั้น จากการที่ต้องให้ไคลเอนท์ทำการล็อคอินเข้ามาเพื่อตรวจสอบผู้ใช้ นั้น ทำให้ต้องใช้ไคลเอนท์ หลายๆเครื่องในการทดสอบ ซึ่งผู้พัฒนามีคอมพิวเตอร์ที่ใช้ในการพัฒนาอยู่เครื่องเดียว จึงต้องลง VMWare ซึ่งเป็นโปรแกรมเครื่องเสมือนโดยจะเปรียบเสมือนมีคอมพิวเตอร์อีกเครื่องในคอมพิวเตอร์ตัวเดียวกัน แต่การใช้ VMWare จะกินระบบของคอมพิวเตอร์มาก ทำให้จำกัดการทดสอบได้เพียงไคลเอนท์เพียงเครื่องเดียวเท่านั้น

8.4 แนวทางการพัฒนาโครงงานต่อ

ในการพัฒนาประยุกต์กับงานอื่นๆ นั้น อาจมีแนวทางในการเพิ่มความสามารถในการอ้างอิงไดเรกทอรีเซอร์วิสอื่นๆ นอกจากไมโครซอฟท์แอคทีฟไดเรกทอรี โดยอาจทำการสร้างโปรแกรมที่ฝังตัวในไคลเอนท์ขึ้นมาเพื่อเปิดโปรแกรมบนเครื่องที่ต้องการใช้งานเพื่อทำการเข้าสู่ระบบ, ติดต่อกับไฟร์วอล และติดตามการใช้งานเพื่อทำการจบการติดต่อ โดยอาจเพิ่มความสามารถทางด้านการเข้ารหัส (เช่น SSL) เพื่อความปลอดภัยในการติดต่อสื่อสารระหว่าง ตัวโปรแกรมบนเครื่องเซิร์ฟเวอร์ กับไฟร์วอล ในการแลกเปลี่ยนข้อมูลการซิงโครไนส์ (Synchronize) ผู้ใช้ ของไคลเอนท์ ณ.เวลาต่างๆ และปรับปรุงการเก็บกฎของผู้ใช้ในไฟร์วอล ให้มีความสะดวกมากขึ้น และยังอาจเพิ่มความสามารถในส่วนของสเตตฟูลอินสเปกชั่นไฟร์วอลได้ หากมีการเผยแพร่ไฟร์วอลที่เป็นแบบสเตตฟูลอินสเปกชั่นโดยนำส่วนของการทำยูสเซอร์เบส มาดัดแปลงเพื่อให้ทำงานกับไฟร์วอลนั้นๆ ได้

8.5 สรุปภาพรวมของโครงงาน

โดยภาพรวมของโครงงานนี้จะเป็นการศึกษาการพัฒนาไฟร์วอลโดยใช้ไอพีเทเบิล ซึ่งทำงานร่วมกับไดเรกทอรีเซอร์วิส ซึ่งโครงงานได้เลือกระบบปฏิบัติการลีนุกซ์สำหรับพัฒนาไฟร์วอลและระบบปฏิบัติการวินโดวส์ 2000 สำหรับผู้ใช้ที่ใช้งานเป็นไคลเอนท์ และได้เลือกแอคทีฟไดเรกทอรีบนระบบปฏิบัติการวินโดวส์ 2000 มาใช้งาน แต่อันเนื่องมาจากการใช้งานองค์ประกอบของวินโดวส์ 2000 มากเกินไป ทำให้ไม่สามารถกำหนดกฎต่างๆของผู้ใช้ตามเวลาจริง ซึ่งหากเป็นเครือข่ายขนาดเล็กก็คงจะไม่พบปัญหามากนัก แต่หากเป็นเครือข่ายขนาดใหญ่จะพบว่า ผู้ใช้จะใช้งานได้ก็ต่อเมื่อไฟร์วอลกำหนดสิทธิของผู้ใช้เสร็จแล้ว ถึงแม้ว่าผู้ใช้จะล็อคอินเข้าสู่เครือข่ายแล้วโดยผ่านทางแอคทีฟไดเรกทอรีก็ตาม แต่ก็ยังสามารถทำงานได้ตามจุดประสงค์ของโครงงานคือ สามารถพัฒนาไฟร์วอลที่อิงกับผู้ใช้ โดยไม่อิงกับไอพี ทำให้ผู้ใช้ในเครือข่ายไม่จำเป็นต้องใช้งานแต่เครื่องตนเองเสมอไป แต่จะสามารถใช้งานเครื่องใดก็ได้ในระบบ ไม่ว่าผู้ใช้จะมีบัญชีผู้ใช้ในระดับใดก็ตาม จึงนับว่าโครงงานนี้พัฒนาประสบผลสำเร็จในระดับหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] H. M. Deitel, P. J. Deitel : "C++ How to Program", Second Edition ,1998

[2] W. Richard Stevens : "UNIX® Network Programming" ,1994

[3] Mark Mitchell, Jeffrey Oldham,and Alex Samuel :"Advanced Linux Programming", 2001

[4] Robert L. Ziegler : "Linux Firewall", Second Edition ,2001

[5] Brian "Beej" Hall : "Beej's Guide to Network Programming Using Internet Sockets" ,2001

[6] David Iseminger : "Active Directory Service for Microsoft Window 2000" , Technical Reference

[7] Ralph Davis : "Window NT Network Programming", How to Survive in a 32-Bit Networking World , 1994

[8] http://www.ecst.csuchico.edu/~beej/guide/net/

[9] http://www.netfilter.org

[10] http://msdn.microsoft.com

[11] http://www.knowplace.org/netfilter/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้