ระบบพิสูจน์ตน

AUTHENTICATION SYSTEMS

นาย ภรายุทธ วังวรัญญู

นางสาว สุจิตรา ไพบูลย์วาณิชย์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

ระบบพิสูจน์ตน

AUTHENTICATION SYSTEMS

โดย

นาย ภรายุทธ วังวรัญญู

นางสาว สุจิตรา ไพบูลย์วาณิชย์

อาจารย์ที่ปรึกษา

อ. ธนา หงษ์สุวรรณ

อ. อัครเดช วัชระภูพงษ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

ปริญญานิพนธ์ปีการศึกษา 2543

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบพิสูจน์ตน

AUTHENTICATION SYSTEMS

ผู้จัดทำ

1. นาย ภรายุทธ วังวรัญญู รหัสประจำตัว 40010543
2. นางสาว สุจิตรา ไพบูลย์วาณิชย์ รหัสประจำตัว 40010867

_____ อาจารย์ที่ปรึกษา

(อ. ธนา หงษ์สุวรรณ)

_____ อาจารย์ที่ปรึกษา

(อ.อัครเดช วัชระภูพงษ์)

ระบบพิสูจน์ตน

นายภรายุทธ วังวรัญญู รหัส 40010543
นางสาว สุจิตรา ไพบูลย์วาณิชย์ รหัส 40010867
อ. ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษา
อ. อัครเดช วัชระภูพงษ์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2543

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ ได้กล่าวถึงระบบพิสูจน์ตน (Authentication Systems) ระบบพิสูจน์ตนเป็นสิ่งที่มีความจำเป็นมากในปัจจุบัน ทั้งนี้เนื่องจากในปัจจุบัน เทคโนโลยีทางด้านคอมพิวเตอร์ได้มีการพัฒนาอย่างรวดเร็ว และได้มีการนำเอาคอมพิวเตอร์มาเชื่อต่อกันเป็นระบบเครือข่าย เพื่อให้สามารถใช้ทรัพยากรของระบบร่วมกัน และติดต่อสื่อสารกันได้ ทำให้ต้องมีการจัดการเกี่ยวกับผู้ใช้ว่าใครที่สามารถเข้าใช้ระบบได้บ้าง และสามารถเข้าใช้ส่วนไหนของระบบได้ ดังนั้นจึงต้องมีวิธีการที่พิสูจน์ว่าผู้ที่ขอเข้าใช้ระบบเป็นผู้ใช้ระบบที่ถูกต้องหรือไม่ ซึ่งคือการพิสูจน์ตนนั่นเอง การพิสูจน์ตนมีหลายวิธี และแต่ละวิธีก็มีขั้นตอน กลไกการทำงานที่แตกต่างกันไป

นอกจากนี้ ภายในปริญญานิพนธ์ฉบับนี้ ยังได้กล่าวถึงวิธีการในการทำให้ระบบปฏิบัติการสามารถทำงานร่วมกันได้ รวมถึงข้อดีข้อเสียของแต่ละวิธีด้วย ทั้งนี้เนื่องจากในองค์กรขนาดใหญ่ มักมีเซิร์ฟเวอร์หลายเซิร์ฟเวอร์ ซึ่งมีระบบปฏิบัติการแตกต่างกันไป การทำให้แต่ละระบบปฏิบัติการเหล่านั้นสามารถทำงานร่วมกันได้ จะทำให้เกิดผลดีต่อผู้ใช้และผู้ดูแลระบบเป็นอย่างมาก การทำให้ระบบสามารถทำงานร่วมกันได้มีหลายแนวทาง แต่ละแนวทางมีข้อดีข้อเสียแตกต่างกันไป แนวทางหนึ่งอาจเหมาะสมกับองค์กรหนึ่ง แต่อาจไม่เหมาะสมกับองค์กรอื่นก็ได้ การเลือกใช้แนวทางใดนั้น ต้องพิจารณาถึงข้อดีข้อเสีย และความเหมาะสมกับสภาพแวดล้อมขององค์กรนั้น

AUTHENTICATION SYSTEMS

Parayuth Wangvarunyoo

Sujitra Paiboolvanish

Thana Hongsuwan Advisor

Akkaradach Watcharapupong Advisor

ABSTRACT

This thesis is about the Authentication Systems which become very interesting and important topic nowadays. Because computer technology has been developed quickly and the used of computer on the network for communicate to each other, that lead to find the way how to manage the networking system on users and system access. There are many ways to protect the access to computer network, authentication is one way to prove that the user have authorized to access the system. There are many methods of authentication and each methods has different working steps.

In additional on this thesis, It also describe the methods how can we manage the different operating systems and make them work together, advantage and disadvantage of each methods, because in the big organization, there are many servers which have the different operating system. There are many methods to make the systems work together. Each methods has its own advantage and disadvantage. One method may suite for one organization but may not suite for the other organization. The way to select the right method for one organization have to consider advantage, disadvantage and the environment of that organization.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้คงไม่อาจเสร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือและร่วมมือจากหลายๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึง เพราะเป็นส่วนสำคัญที่ทำให้ปริญญานิพนธ์เสร็จลงได้ก็คือ อาจารย์ธนา หงษ์สุวรรณ และอาจารย์อัครเดช วัชระภูพงษ์ อาจารย์ที่ปรึกษาปริญญานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำ และช่วยเหลือเสมอมา ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก และนายปิยะ ผลิเจริญผล ที่ได้ให้คำปรึกษาและช่วยเหลือเสมอมา

นอกเหนือจากนี้ต้องขอขอบพระคุณอาจารย์ทุกท่านในภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เป็นอย่างยิ่งที่ได้ช่วยประสิทธิ์ประสาทวิชาความรู้ให้แก่คณะผู้จัดทำ รวมทั้งต้องขอบคุณห้อง ISAG ที่เอื้อเฟื้ออุปกรณ์ ทรัพยากรและสถานที่ให้ทำปริญญานิพนธ์นี้

สุดท้ายขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันที่เป็นที่เคารพรักยิ่ง ซึ่งเป็นผู้ที่ให้กำเนิด เลี้ยงดูมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ เอาใจใส่เสมอมา ในทุกๆ ด้านอันหาที่เปรียบมิได้ คณะผู้จัดทำขอระลึกในพระคุณอันยิ่งใหญ่สุดประมาณนี้ไว้กว่าชีวิตจะหาไม่ และขอกราบขอบพระคุณไว้ ณ ที่นี้ด้วย

ภรายุทธ วังวรัญญู
สุจิตรา ไพบูลย์วาณิชย์

สารบัญ

สารบัญรูปภาพ

สารบัญตาราง

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญ

ในระบบปฏิบัติการแบบมัลติยูสเซอร์นั้น ต้องมีการป้องกันผู้ใช้จากบุคคลอื่นๆ ซึ่งอาจเป็นบุคคลที่ไม่มีสิทธิ์ในการใช้ระบบ ดังนั้น ผู้ใช้ระบบจำเป็นต้องพิสูจน์ตนก่อนเข้าใช้ระบบ เพื่อพิสูจน์ว่าเป็นผู้ใช้ระบบที่ถูกต้องตามที่อ้างถึง

การพิสูจน์ตนของยูสเซอร์นั้นมี 3 ประเภทหลักคือ

1. การใช้ยูสเซอร์เนมและรหัสผ่าน (password)
2. การใช้วิธีไบโอเมทริค (biometric) เช่น การตรวจลายนิ้วมือ การตรวจม่านตา การตรวจคลื่นเสียง เป็นต้น
3. การใช้โทเคน (token) เช่นการใช้สมาร์ตการ์ดในการพิสูจน์ตน

ถึงแม้การพิสูจน์ตนมีหลายวิธี แต่ทุกวิธีอยู่บนพื้นฐานเดียวกัน โดยประกอบด้วย 2 ส่วนคือ ส่วนที่บ่งบอกว่าเป็นยูสเซอร์คนใดในระบบ และส่วนที่พิสูจน์ว่าเป็นยูสเซอร์คนนั้นจริง เช่น ยูสเซอร์เนมเป็นส่วนที่บ่งบอกว่าเป็นยูสเซอร์คนใด และรหัสผ่านเป็นส่วนที่ใช้พิสูจน์

ในปัจจุบันการติดต่อผ่านเครือข่ายมีความปลอดภัยน้อยลง ดังนั้นจึงต้องมีระบบพิสูจน์ตนที่มีความสามารถในการป้องกันการลักลอบดักฟังหรือแก้ไขข้อมูลจากบุคคลอื่นที่ไม่ประสงค์ดี และนอกจากนี้ ระบบในองค์กรมีความซับซ้อนและขยายขนาดใหญ่มากยิ่งขึ้น ทำให้มีจำนวนเซิร์ฟเวอร์มากยิ่งขึ้น และยังทำให้ในระบบเครือข่ายมีระบบปฏิบัติการหลายระบบปฏิบัติการ ซึ่งส่งผลให้การทำงานยุ่งยากขึ้น เนื่องจากระบบปฏิบัติการต่างๆ ไม่สามารถทำงานร่วมกันได้ แม้ว่าอยู่ในระบบเครือข่ายเดียวกันก็ตาม นอกจากนี้ในการที่ระบบมีหลายเซิร์ฟเวอร์ยังทำให้ผู้ใช้มีรหัสผ่านหลายรหัสผ่านด้วย ผู้ใช้ระบบจึงลำบากที่จะต้องจำรหัสผ่านของแต่ละเซิร์ฟเวอร์ และผู้ดูแลระบบก็ลำบากในการจัดการระบบ

1.2 วัตถุประสงค์

1. เพื่อศึกษาและเข้าใจการทำงานของระบบพิสูจน์ตนแบบต่างๆ
2. เพื่อศึกษาวิธีที่ทำให้แต่ละระบบปฏิบัติการสามารถทำงานร่วมกัน
3. เพื่อวิเคราะห์เปรียบเทียบข้อดีข้อเสียของแต่ละวิธีที่ทำให้แต่ละระบบปฏิบัติการสามารถทำงานร่วมกัน
4. เพื่อนำวิธีที่เหมาะสมที่ได้วิเคราะห์มาทดลองใช้

1.3 ขอบเขตของโครงงาน

1. ศึกษาการทำงานของระบบพิสูจน์ตนที่สำคัญบางระบบ
2. ศึกษาวิธีที่ทำให้แต่ละระบบปฏิบัติการสามารถทำงานร่วมกัน

3. วิเคราะห์ข้อดีข้อเสียของแต่ละวิธีที่ทำให้แต่ละระบบปฏิบัติการสามารถทำงานร่วมกัน
4. ทดลองใช้ระบบพิสูจน์ตนบางระบบ
5. ทดลองวิธีที่เหมาะสมที่ได้จากการวิเคราะห์

1.4 ขั้นตอนการดำเนินงาน

1. ศึกษาศัพท์ที่เกี่ยวข้องกับระบบพิสูจน์ตน
2. ศึกษาการทำงานของระบบพิสูจน์ตนแบบต่างๆ
3. ศึกษาวิธีในการทำให้ระบบปฏิบัติการต่างๆ สามารถทำงานร่วมกันได้
4. ทดลองใช้ระบบที่เหมาะสม
5. ทำรายงานและปริญญานิพนธ์

บทที่ 2
หลักการพื้นฐานเกี่ยวกับระบบพิสูจน์ตน

ความรู้เกี่ยวกับระบบรักษาความปลอดภัยบนคอมพิวเตอร์นับเป็นสิ่งที่มีความสำคัญมากในปัจจุบัน เนื่องจากมีการใช้คอมพิวเตอร์อย่างแพร่หลาย โดยได้เข้ามามีบทบาทในการจัดการข้อมูลต่างๆ ที่มีอยู่ในองค์กร ดังนั้นจึงถือเป็นเรื่องจำเป็นอย่างยิ่งที่ผู้มีส่วนเกี่ยวข้องจำเป็นต้องเรียนรู้วิธีการจัดการข้อมูล วิธีควบคุมสิทธิต่างๆ ให้เป็นระเบียบแบบแผน และปลอดภัยต่อผู้ที่เกี่ยวข้องกับข้อมูลทุกคน

2.1 ระบบรักษาความปลอดภัยบนคอมพิวเตอร์

ระบบรักษาความปลอดภัยบนคอมพิวเตอร์คือ สิ่งที่คอยปกป้องคุ้มครองคอมพิวเตอร์และสิ่งที่เกี่ยวข้องให้พ้นอันตรายและการสูญหาย ทุกสิ่งที่เกี่ยวข้องกับคอมพิวเตอร์จะได้รับการคุ้มครองจากระบบรักษาความปลอดภัย ตามทฤษฎีระบบรักษาความปลอดภัยมีสิ่งที่ต้องคำนึงถึงดังต่อไปนี้

1. ความมั่นคงและถูกต้อง (Integrity & Accuracy) ข้อมูลที่อยู่บนคอมพิวเตอร์ต้องปลอดภัย ไม่สูญหาย ไม่เสียหาย ไม่ถูกเปลี่ยนแปลง โดยอุบัติเหตุหรือเจตนาจากผู้ที่ไม่ได้รับอนุญาต ในการส่งผ่านข้อมูลต้องมีการรับรองอย่างถูกต้อง มีบันทึกเกี่ยวกับการรับส่ง
2. ความมั่นใจ (Confidentiality) คอมพิวเตอร์ต้องเก็บรักษาความลับได้ จำแนกได้ว่าใครเป็นผู้มีสิทธิ์ และใครคือผู้ไม่มีสิทธิ์จัดการข้อมูล
3. ความสามารถเข้าถึงข้อมูลได้ (Availability) ข้อมูลที่ควรเข้าถึงได้ ต้องเข้าถึงได้ง่าย สะดวกต่อการนำมาใช้ อยู่ในที่ๆ สามารถนำมาใช้ได้ตลอดเวลา หากเกิดอุบัติเหตุขึ้นต้องสามารถซ่อมคืนได้

2.2 ส่วนประกอบของระบบความปลอดภัย

ส่วนที่ประกอบขึ้นมาเป็นระบบรักษาความปลอดภัยมีด้วยกัน 3 ข้อ

1. การออกแบบระบบ การออกแบบระบบที่ดีทำให้สามารถรักษาความปลอดภัยได้อย่างมีประสิทธิภาพ การใช้สถาปัตยกรรมเข้ามาจัดการระบบเป็นตัวอย่างหนึ่งของการออกแบบระบบที่ดี ทำให้สามารถจัดแบ่งหน่วยความจำ และแยกอภิสิทธิ์ออกจากสิทธิ์ทั่วไปได้
2. การควบคุมการจัดข้อมูล การควบคุมข้อมูลหมายถึง การกำหนดว่าให้ใครสามารถจัดการข้อมูลได้บ้าง และต้องกำหนดด้วยว่าจัดการข้อมูลไปเพื่อจุดประสงค์ใด
3. การควบคุมการจัดการในระบบ การควบคุมการจัดการในระบบทำให้สามารถกำหนดได้ว่าใครมีสิทธิ์ใช้ข้อมูลได้ถึงระดับไหน นอกจากนี้ยังทำให้แน่ใจด้วยผู้ที่ไม่ได้รับอนุญาตจะไม่มีสิทธิ์จัดการข้อมูล

2.3 การควบคุมการจัดการ

เป้าหมายหลักของการรัษาความปลอดภัยอยู่ที่ต้องสามารถจำกัดได้ว่าให้ใครเข้าถึงข้อมูลได้มากขนาดไหน ซึ่งเรียกว่าการควบคุมการจัดการ (access control) เหตุผลที่ต้องควบคุมการจัดการคือ

1. เพื่อสนับสนุนให้การเข้าถึงข้อมูลของผู้ที่ได้รับอนุญาตเป็นไปอย่างถูกต้องและง่ายดาย
2. เพื่อส่งเสริมให้เกิดความมั่นคงของข้อมูล
3. เพื่อปกป้องความเป็นส่วนตัวในข้อมูลส่วนบุคคล

นอกจากนี้ การควบคุมการจัดการยังมีขอบเขตไปถึงการจำกัดการใช้โปรแกรมต่างๆ เพื่อลบเขียนทับ และทำสำเนาด้วย ในการควบคุมการจัดการเราต้องคำนึงถึงสิ่งต่อไปนี้

1. ใครที่สามารถใช้ได้บ้าง (Authentication)
2. ผู้ที่ได้รับอนุญาต มีสิทธ์ใช้ส่วนใดและระดับใดได้บ้าง (Authorization)
3. ต้องบันทึกการกระทำต่างๆ ของผู้ที่ได้รับอนุญาต (Accounting)

เมื่อยูสเซอร์ต้องการใช้บริการจากระบบ ยูสเซอร์ต้องบอกระบบว่าเป็นใคร และระบบจะตรวจสอบว่ายูสเซอร์เป็นคนนั้นจริงหรือไม่ ทั้งสองขั้นตอนเรียกว่าการแสดงตน (Identification) และการพิสูจน์ตน (Authentication)

2.4 การแสดงตนและการพิสูจน์ตน

การแสดงตนคือ วิธีที่ยูสเซอร์บอกให้ระบบทราบว่ายูสเซอร์เป็นใคร ส่วนการพิสูจน์ตนคือ วิธีที่ระบบใช้เพื่อตรวจสอบว่ายูสเซอร์เป็นคนนั้นตามที่ยูสเซอร์อ้าง วิธีที่ยูสเซอร์ใช้พิสูจน์ตนนั้นมี 3 วิธีคือ

1. ใช้สิ่งที่ยูสเซอร์**ทราบ** เช่น ทราบชื่อล็อกอินและรหัสผ่าน วิธีนี้มีหลักการคือ หากยูสเซอร์ทราบรหัสผ่าน ระบบถือว่ายูสเซอร์คือยูสเซอร์ที่ถูกต้อง วิธีนี้ปัญหาหลายอย่าง เพราะรหัสผ่านที่ได้มานั้น อาจได้มาจากการขโมย ขอยืม ถอดรหัส หรือแม้กระทั่งเดา วิธีนี้มีความปลอดภัยไม่มาก แต่ยังเป็นที่นิยม เนื่องจากใช้ง่าย สะดวกสบาย ไม่สิ้นเปลือง ในอนาคตคาดการณ์ว่าจะยังคงใช้วิธีนี้กันต่อไป
2. ใช้สิ่งที่ยูสเซอร์**มี** เช่น สมาร์ตการ์ด หลักการของวิธีนี้คือ หากยูสเซอร์มีสิ่งที่ถูกต้อง ระบบจะถือว่ายูสเซอร์เป็นยูสเซอร์ที่ถูกต้อง วิธีนี้นี้มีปัญหาคือ สิ่งของที่มีอยู่นั้น อาจถูกขโมย ทำหาย หรือถูกทำสำเนาไปใช้ได้
3. ใช้สิ่งที่ยูสเซอร์**เป็น** เช่น รอยนิ้วมือ เครื่องมือที่ใช้ในการพิสูจน์ตนแบบนี้เรียกว่า ไบโอเมทริค (biometric) โดยวิธีนี้นำข้อมูลที่ได้เปรียบเทียบกับข้อมูลของแต่ละบุคคลที่บันทึกอยู่ก่อนแล้ว วิธีนี้มีปัญหาคือ ระบบตรวจสอบทำงานผิดพลาด

2.5 แฟกเตอร์ (factor)

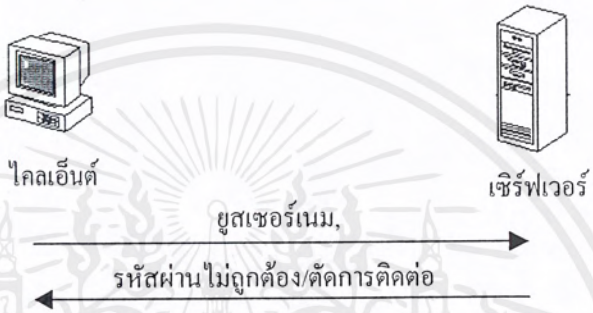
แฟกเตอร์หมายถึง สิ่งที่ใช้ในการพิสูจน์ตน เช่น การใช้ยูสเซอร์เนมและรหัสผ่าน เรียกว่าเป็นระบบรักษาความปลอดภัยแบบแฟกเตอร์เดี่ยว เนื่องจากใช้สิ่งที่ยูสเซอร์ทราบเพียงอย่างเดียว หรือการเบิก

เงินตามตู้เอทีเอ็มโดยใช้บัตรและรหัสผ่าน เป็นระบบรักษาความปลอดภัยแบบ 2 แฟกเตอร์ เนื่องจากใช้สิ่งที่ยูสเซอร์ทราบและสิ่งที่ยูสเซอร์มี

2.6 ประเภทของระบบพิสูจน์ตน

จากที่กล่าวมานี้เป็นภาพรวมของระบบรักษาความปลอดภัย จะเห็นได้ว่าการพิสูจน์ตนเป็นส่วนหนึ่งของระบบรักษาความปลอดภัย โดยการพิสูจน์ตนแบ่งออกเป็น 2 ประเภทได้ดังนี้

1. ระบบพิสูจน์ตนแบบที่ไม่มีตัวกลางในการพิสูจน์



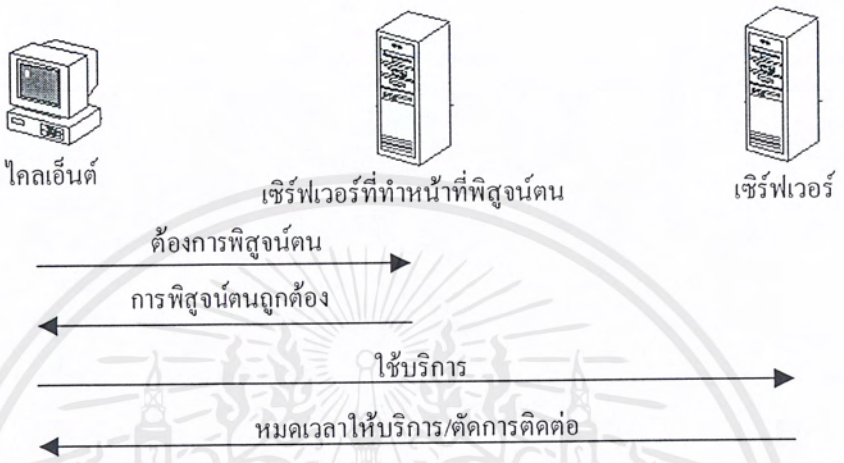
รูปที่ 2-1 แสดงขั้นตอนการพิสูจน์ตนอย่างง่าย เมื่อรหัสผ่านไม่ถูกต้อง

รูปที่ 1-2 แสดงขั้นตอนการพิสูจน์ตนอย่างง่าย โดยรหัสผ่านถูกต้อง

โดยข้อมูลที่ส่งผ่านระหว่างไคลเอ็นต์และเซิร์ฟเวอร์อาจมีการเข้ารหัสหรือไม่เข้ารหัสก็ได้

2. ระบบพิสูจน์ตนแบบที่มีตัวกลางในการพิสูจน์

การใช้ตัวกลางที่ทำหน้าที่พิสูจน์ตนก่อนที่ไคลเอ็นต์จะใช้บริการจากเซิร์ฟเวอร์ได้ ทำให้การพิสูจน์ตนมีความปลอดภัยมากขึ้น เนื่องจากไม่จำเป็นต้องเก็บข้อมูลที่เป็นความลับไว้หลายที่ถ้ามีหลายเซิร์ฟเวอร์ โดยการเก็บข้อมูลที่เป็นความลับไว้ที่เซิร์ฟเวอร์ที่ทำหน้าที่พิสูจน์ตนเพียงที่เดียว



รูปที่ 2-3 การพิสูจน์ตนโดยมีตัวกลางทำหน้าที่ตรวจสอบการพิสูจน์ตน

บทที่ 3
ระบบพิสูจน์ตนแบบที่ไม่มีตัวกลางในการพิสูจน์

3.1 ไฟล์ passwd

3.1.1 ไฟล์ /etc/passwd

3.1.1.1 การเก็บข้อมูลของไฟล์ /etc/passwd

ข้อมูลพื้นฐานที่เกี่ยวกับผู้ใช้ระบบยูนิกซ์ คือ มีผู้ใช้ระบบคนใดบ้างที่อยู่บนระบบนั้น หรือต้องการรู้ชื่อจริงของผู้ใช้ระบบ ข้อมูลเหล่านี้มักเก็บอยู่ในไฟล์ /etc/passwd

/etc/passwd เก็บข้อมูล 1 แถว (หรือ 1 เรคอร์ด) ต่อ 1 ยูสเซอร์แอ็กเคาต์ ในไฟล์ไม่สามารถมีข้อความแปลกปลอมใดๆ แต่ละแถวจะประกอบด้วยฟิลด์ต่างๆ แต่ละฟิลด์จะถูกแบ่งด้วยตัวอักษร colon (':') ตัวอย่างข้อมูลในไฟล์ passwd เช่น

root:UqstWvHV002G1:0:0:root:/root:/bin/tcsh

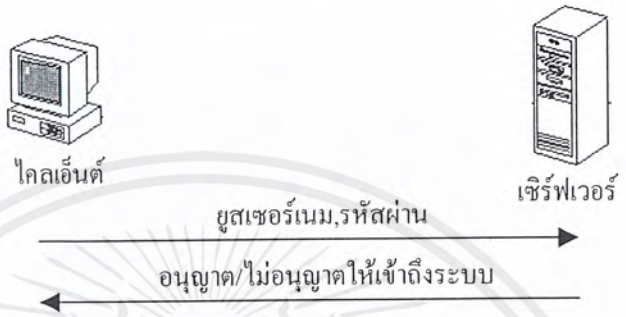
ตัวอย่างนี้แสดงถึงยูสเซอร์แอ็กเคาต์ของ root ฟิลด์ส่วนใหญ่ใช้เหมือนกันในระบบปฏิบัติการยูนิกซ์ คือ

- Username ชื่อที่บ่งบอกถึงยูสเซอร์แอ็กเคาต์ใช้เมื่อถูกล็อกอินเข้าสู่ระบบ
- Password เป็นรหัสผ่านที่ได้รับการเข้ารหัสแล้วของผู้ใช้ระบบ ในระบบส่วนใหญ่ ฟิลด์นี้จะเก็บอยู่อีกไฟล์
- User ID ID ที่เป็นตัวเลข ซึ่งให้แก่ผู้ใช้ระบบทุกคน ตัวเลขนี้จะใช้แทนยูสเซอร์เนมเมื่อมีการ ติดต่อกันภายในระบบ ส่วนใหญ่เป็นเพราะการใช้ตัวเลขที่กำหนดขนาดแน่นอน (ซึ่งส่วนใหญ่ใช้ 2 – 4 ไบต์) เป็นคีย์ในการค้นหาในตาราง และตัวเลขนี้สามารถจัดการได้ง่าย
- User's Group ID ID ที่เป็นตัวเลขของกลุ่มที่ผู้ใช้ระบบอยู่ ผู้ใช้ระบบทุกคนต้องเป็นสมาชิกกลุ่มอย่างน้อย 1 กลุ่มในระบบกลุ่มอื่นๆ อาจถูกตั้งขึ้นมาโดยใช้ไฟล์ /etc/group
- GECOS field ฟิลด์นี้ แล้วแต่ระบบต้องการใส่ซึ่งเป็นคำอธิบายของผู้ใช้ระบบ มักเป็น ชื่อจริงของผู้ใช้แอ็กเคาต์
- Home Directory บอกตำแหน่งของโฮมไดเรกทอรี ซึ่งเป็นตำแหน่งแรกและตำแหน่งประจำเมื่อผู้ใช้ระบบล็อกอินเข้าสู่ระบบ
- Shell บอกเชลล์เริ่มต้นเมื่อผู้ใช้ระบบล็อกอินเข้าสู่ระบบ

ผู้ใช้ระบบทุกคนสามารถอ่านไฟล์ /etc/passwd ได้ แต่มีผู้ดูแลระบบเท่านั้นที่สามารถเปลี่ยนแปลงข้อมูลได้ นั่นหมายความว่าผู้ใช้ระบบไม่สามารถเปลี่ยนแปลงไฟล์ได้โดยตรง แต่มีโปรแกรมพิเศษที่อนุญาตให้ผู้ใช้ระบบเปลี่ยนรหัสผ่านของตน เชลล์เริ่มต้น และ ฟิลด์ GECOS

3.1.1.2 ขั้นตอนการพิสูจน์ตน

1. ไคลเอ็นต์รับยูสเซอร์เนมและรหัสผ่านจากผู้ใช้ระบบ
2. ไคลเอ็นต์นำข้อมูลที่ได้รับจากผู้ใช้ระบบเข้ารหัสและส่งไปยังเซิร์ฟเวอร์
3. เซิร์ฟเวอร์นำข้อมูลที่ได้รับการเข้ารหัสแล้วเปรียบเทียบกับข้อมูลที่เก็บในไฟล์
4. ถ้าตรงกันก็อนุญาตให้ผู้ใช้ระบบคนนั้นสามารถเข้าถึงระบบได้ แต่ถ้าไม่ตรงกันก็ไม่อนุญาต



รูปที่ 3-1 แสดงขั้นตอนการทำงานของไฟล์ /etc/passwd

3.1.1.3 ข้อเสียของไฟล์ /etc/passwd

การเก็บรหัสผ่านที่ได้รับการเข้ารหัสของผู้ใช้ระบบในไฟล์ /etc/passwd นั้นไม่มีความปลอดภัย ถึงแม้มีอัลกอริทึมในการเข้ารหัสที่ไม่สามารถแปลงกลับได้ก็ตาม ผู้ใช้ระบบสามารถทำสำเนาไฟล์ /etc/passwd จากระบบสู่เครื่องของตน และใช้โปรแกรมแกะรหัสผ่าน เพื่อเดารหัสผ่านของผู้ใช้ระบบแต่ละคน ซึ่งมักจะได้รหัสผ่านของผู้ใช้ระบบแทบทุกระบบ

3.1.2 ไฟล์ etc/shadow

ไฟล์ /etc/shadow เก็บรหัสผ่านของผู้ใช้ระบบต่างๆ ในไฟล์อื่นซึ่งมีแต่ผู้ดูแลระบบเท่านั้นที่สามารถอ่านได้ ส่วนในฟิลด์รหัสผ่านของไฟล์ /etc/passwd เป็นตัวอักษรพิเศษที่บ่งบอกว่าแท้ที่จริงแล้วรหัสผ่านเก็บอยู่ในไฟล์ /etc/shadow ตัวอักษรนั้นอาจต่างกันไปในแต่ละระบบ และที่เก็บของไฟล์ shadow นั้นก็แตกต่างกันไปในแต่ละระบบ เนื่องจาก shadow ไม่ถือว่าเป็นมาตรฐาน

ไฟล์ etc/shadow ประกอบด้วย 9 ฟิลด์ ดังนี้

login-id	ชื่อล็อกอินของผู้ใช้ระบบ
password	รหัสผ่านที่ได้รับการเข้ารหัสแล้ว มี 13 ตัวอักษร หรือมากกว่า ขึ้นอยู่กับอัลกอริทึม
lastchg	จำนวนวันที่เปลี่ยนแปลงรหัสผ่านครั้งสุดท้าย นับจากวันที่ 1 มกราคม 2970
min	จำนวนวันที่น้อยที่สุดที่ต้องการในการเปลี่ยนรหัสผ่าน
max	จำนวนวันที่มากที่สุดที่รหัสผ่านยังใช้งานได้
warn	จำนวนวันก่อนที่รหัสผ่านจะใช้งานไม่ได้ หลังจากได้รับการเตือนให้เปลี่ยนรหัสผ่าน
inactive	จำนวนวันที่อนุญาตให้ผู้ใช้ระบบไม่ต้องทำการเปลี่ยนแปลงใดๆ
expire	จำนวนวันที่แท้จริงที่ล็อกอินนี้ไม่สามารถใช้ได้อีกต่อไป

flag ปัจจุบันนี้ยังไม่ได้ใช้

3.1.3 ตัวแกะรหัสผ่าน (Password Cracker)

3.1.3.1 ตัวแกะรหัสผ่านคืออะไร

โปแกรมแกะรหัสผ่าน (Password cracker) เป็นโปรแกรมที่สามารถถอดรหัสของรหัสผ่านได้ หรือทำการป้องกันรหัสผ่านใช้การไม่ได้ โปรแกรมแกะรหัสผ่านนั้นไม่จำเป็นต้องถอดรหัสอะไรทั้งสิ้น เพราะในความเป็นจริงแล้ว การนำรหัสผ่านมาเข้ารหัสแล้วไม่สามารถถอดรหัสกลับได้

การนำรหัสผ่านมาเข้ารหัสนั้นไม่สามารถถอดรหัสกลับได้ วิธีการในปัจจุบันส่วนใหญ่ใช้ฟังก์ชั่นแบบทางเดียว (one-way) นั่นคือ ไม่มีการทำงานใดๆ สามารถแปลงรหัสผ่านนั้นกลับให้เป็นข้อมูลที่เป็นตัวอักษรเดิมได้อีก

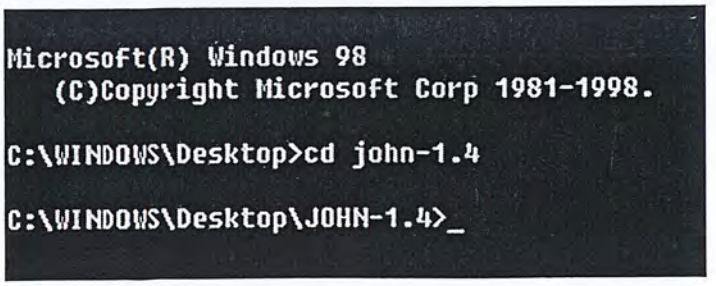
ดังนั้นจึงมีการใช้เครื่องมือเพื่อให้ได้รหัสผ่านดั้งเดิม การทำงานของโปรแกรมแกะรหัสผ่านใช้วิธีบรูตฟอร์ซ (brute-force) คือโปรแกรมจะเปรียบเทียบทีละคำ โดยการทำงานที่ความเร็วสูง ซึ่งความเร็วในการทำงานขึ้นกับอัลกอริทึมที่ใช้ทฤษฎีนี้ได้พิสูจน์ว่าใช้ได้ดี กับผู้ใช้ระบบที่ไม่ใส่ใจในการตั้งรหัสผ่านที่ดี

ผู้ใช้ระบบส่วนใหญ่ไม่ได้รับการอบรมในการตั้งรหัสผ่านที่ดี ถ้ารหัสผ่านที่ตั้งมีอยู่ในพจนานุกรม ทำให้แคร็กได้ง่ายมาก ส่วนผู้ที่ได้รับการอบรมแล้ว รหัสผ่านของพวกเขามักไม่อยู่ในพจนานุกรม ซึ่งมีความปลอดภัย ผู้ใช้ระบบหลายคนคิดว่า พวกเขาไม่มีข้อมูลที่เป็นส่วนตัวเก็บไว้ จึงไม่มีความจำเป็นที่ต้องยุ่งเกี่ยวกับการรักษาความปลอดภัย จุดเล็กๆ จุดนี้ ทำให้ระบบทั้งระบบอาจถูกทำลายโดยผู้บุกรุกได้

ในส่วนการแคร็กคีย์ซีดี/คีย์ซีเรียลนั้นไม่ถือว่าเป็นการแกะรหัสผ่าน การหาคีย์ซีดีนั้นส่วนใหญ่ได้มาจากการที่บริษัทตั้งอัลกอริทึมง่ายเกินไป ทำให้ผู้อื่นสามารถหาอัลกอริทึมได้ หรือไม่ก็มาจากบุคคลในบริษัทบอกหรือการที่มีคนลงทะเบียนไว้กับทางบริษัทแล้วบอกคีย์ซีดีกับผู้อื่นทางเว็บต่างๆ

3.1.3.2 ตัวอย่างโปรแกรมแกะรหัสผ่าน

โปรแกรมตัวอย่างเป็นโปรแกรมชื่อว่า john เวอร์ชั่น 1.4 สามารถใช้ได้กับระบบปฏิบัติการวินโดวส์ 95 ขึ้นไป ซึ่งสามารถดาวน์โหลดเป็นไฟล์นามสกุล .zip และทำการขยายขนาดไฟล์กลับ เราจะต้องนำไฟล์ passwd เก็บไว้ในไดเรกทอรี john-1.4 และเรียก MS-DOS prompt ดังรูป



รูปที่ 3-2 แสดงการใช้ MS-DOS prompt เพื่อเรียกโปรแกรม

ถ้าพิมพ์คำสั่ง john แล้วเคาะปุ่มเอ็นเตอร์โปรแกรมจะแสดงคำสั่งต่างๆ ของโปรแกรมบนหน้าจอ ซึ่งโปรแกรม john นั้น มีหลายโหมดการทำงาน

โหมดการทำงานโหมดแรกซึ่งเป็นโหมดการทำงานที่ง่ายที่สุดคือซิงเกิลโหมด (single-mode) โปรแกรมจะพยายามแคร็กรหัสผ่านโดยดูจากยูสเซอร์เนมและหารหัสผ่านที่มีความคล้ายคลึงกับยูสเซอร์เนม แต่ว่ามีประสิทธิภาพต่ำสุด ซึ่งลองใช้กับ passwd จำลองได้ผลดังนี้

```
C:\WINDOWS\Desktop\JOHN-1.4>john -single passwd.txt
John the Ripper  Version 1.4  Copyright (c) 1996,97 by Solar Designer
Loaded 8 passwords with 8 different salts (DES)
r00t     (root)
v: 1  c: 16840  s: 1 99%  c/s: 16860  w: *****DONE*****
```

รูปที่ 3-3 การทำงานแบบซิงเกิลโหมด

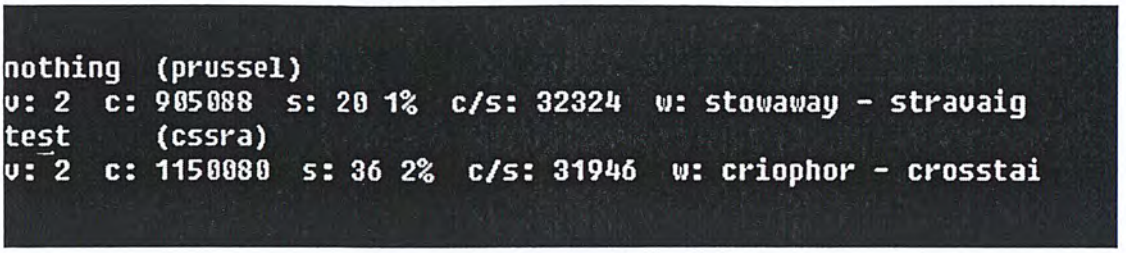
หลังจากที่ให้โปรแกรมทำงาน ปรากฏว่าได้รหัสผ่านของผู้ดูแลระบบ (root) ซึ่งเค้าได้ตั้งรหัสผ่านไว้คล้ายกับยูสเซอร์เนมของเค้า แต่ระบบส่วนใหญ่จะไม่ให้มีการล็อกอินเป็นผู้ดูแลระบบโดยตรง จึงต้องหายูสเซอร์เนมของผู้ใช้ระบบคนอื่น

ในโหมดอื่น คือ เวิร์ดลิสต์โหมด (word-list mode) ซึ่งจะสามารถกำหนดไฟล์พจนานุกรมที่อื่นได้ ถ้าไฟล์พจนานุกรมที่ใหญ่กว่า นั่นหมายความว่าเรามีโอกาสที่จะแคร็กได้มากขึ้น ซึ่งเราสามารถดาวน์โหลดไฟล์พจนานุกรมจากที่อื่นได้ แล้วนำไปใส่ในไดเรกทอรีเดียวกัน แล้วกำหนดชื่อของไฟล์พจนานุกรมดังนี้

```
C:\WINDOWS\Desktop\JOHN-1.4>john -wordfile:english.dic passwd.txt
John the Ripper  Version 1.4  Copyright (c) 1996,97 by Solar Designer
Loaded 7 passwords with 7 different salts (DES)
```

รูปที่ 3-4 การกำหนดชื่อไฟล์พจนานุกรมอื่น

เราสามารถดูการทำงานของโปรแกรมได้โดยกดสเปซบาร์ แล้วโปรแกรมจะแสดงว่าขณะนี้โปรแกรมได้ทำงานไปถึงไหนแล้ว

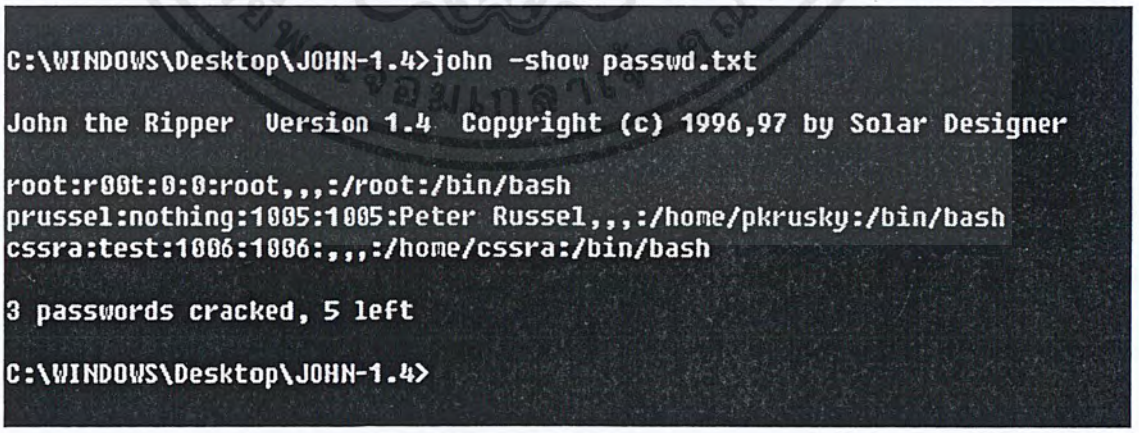


รูปที่ 3-5 การกดสเปซบาร์เพื่อดูการทำงานของโปรแกรม

ขณะนี้ได้รหัสผ่านของผู้ใช้ระบบมา 2 รหัสผ่านแล้ว แต่โปรแกรมนี้สามารถทำงานได้อย่างมีประสิทธิภาพมากกว่านี้ โปรแกรมนี้สามารถสลับตัวอักษร นำคำแต่ละคำทดลองแบบย้อนกลับ เช่น abc ก็จะทดลองเป็น cba ทดลองโดยให้เป็นตัวอักษรใหญ่ และยังมีอีกมากมาย ถ้าเราใช้โหมด rule cracking คำสั่งของโหมดนี้คือ john-rule-wordfile:english.doc การทำงานของโปรแกรมในโหมดนี้ จะใช้เวลานานกว่าแต่สามารถได้รหัสผ่านมากกว่า โปรแกรมนี้สามารถหยุดได้ทุกเมื่อที่ต้องการโดยการกด Ctrl + C และพิมพ์ – restore ภายหลังเพื่อให้ทำงานต่อจากคราวที่แล้ว

โหมดต่อไปคือ อินครีเมนทอลโหมด (incremental mode) ในโหมดนี้โปรแกรมจะพยายามค้นหารหัสผ่านทุกวิถีทางจนกว่าจะได้รหัสผ่าน ส่วนใหญ่จะใช้โหมดนี้เมื่อโหมดอื่นไม่สามารถหารหัสผ่านได้แล้ว ในอินครีเมนทอลโหมดนั้น จะมีหลายวิธีการ เช่น มีการนำตัวเลขมาใช้ การใช้ตัวอักษรตัวใหญ่และตัวเล็กผสมกัน การทดลองใช้รหัสผ่านทุกอย่างเท่าที่มี ดูในไฟล์ john.ini เพื่อดูการทำงานในอินครีเมนทอลโหมด คำสั่งคือ john –incremental:all passwd ถ้าต้องการใช้โหมดการทดลองใช้รหัสผ่านทุกอย่างเท่าที่มี

ถ้าใช้คำสั่ง john –show โปรแกรมจะแสดงชื่อและรหัสผ่านที่แคร็กมาได้ดังรูป



รูปที่ 3-6 การใช้คำสั่ง john –show เพื่อแสดงยูสเซอร์เนมและรหัสผ่านที่แคร็กได้

3.2 ระบบ S/KEY ระบบแบบใช้รหัสผ่านครั้งเดียว

S/KEY เป็นระบบที่ใช้รหัสผ่านแบบใช้ครั้งเดียวจะคอยจัดการเกี่ยวกับการแสดงตนบนเครือข่าย เพื่อป้องกันระบบจากผู้ลักลอบดักฟัง ระบบนี้จะมีความได้เปรียบเมื่อเทียบกับระบบพิสูจน์ตนด้วยการใช้รหัสผ่านซ้ำๆ ไม่มีการเก็บข้อมูลที่เป็นความลับไว้ในที่ใดๆ เครื่องฝั่งไคลเอ็นต์สามารถใช้โปรแกรมนี้ที่ใดก็ได้ และฝั่งโฮสต์สามารถนำไปรวมกับโปรแกรมประยุกต์ที่ต้องการใช้การพิสูจน์ตน

3.2.1 คุณสมบัติของ S/KEY

ระบบพิสูจน์ตน S/KEY มีวิธีการที่ธรรมดา สามารถป้องกันรหัสผ่านของผู้ใช้ระบบจากการดักฟัง (passive attack) ระบบนี้ไม่มีความสามารถเท่ากับ Kerberos หรือ SDASS และไม่สามารถป้องกันรหัสผ่านจากการดักเปลี่ยน (active attack) ได้ แต่ระบบนี้สามารถใช้ได้กับระบบปฏิบัติการยูนิกซ์ได้อย่างรวดเร็วและง่ายดายโดยไม่จำเป็นต้องให้ระบบเก็บข้อมูลเพิ่มเติม (เช่นไฟล์ที่เก็บรหัสผ่านที่เป็นตัวอักษร) ซึ่งมักจะมีผลมากกว่ารหัสผ่านที่ได้รับการเข้ารหัสซึ่งเก็บอยู่แล้วในระบบ ระบบ S/KEY สามารถใช้กับเครื่องปลายทางทั่วไปหรือพีซีโดยจะต้องมีโปรแกรมสื่อสารที่เป็นมาตรฐาน

ลักษณะบางอย่างของระบบ S/KEY :

- ป้องกันการลักลอบดักฟัง
- มีแนวคิดที่ธรรมดาและสามารถนำไปใช้ได้ง่าย
- ไม่ต้องการอุปกรณ์พิเศษและสามารถประยุกต์ได้ง่าย
- สามารถพิสูจน์ตนได้อย่างอัตโนมัติจากระบบที่เชื่อถือได้
- ไม่มีอัลกอริทึมที่เป็นความลับ
- ไม่มีการเก็บข้อมูลที่เป็นความลับบนโฮสต์

3.2.2 ระบบโดยรวมของ S/KEY

ระบบนี้จะแบ่งได้เป็น 2 ส่วนคือ ฝั่งผู้ใช้ระบบ (ไคลเอ็นต์) จะต้องสร้างรหัสผ่านแบบครั้งเดียว ส่วนฝั่งระบบ (เซิร์ฟเวอร์) จะต้องพิสูจน์รหัสผ่านนั้น รหัสผ่านแบบใช้ครั้งเดียวนั้นถูกสร้างและพิสูจน์โดยใช้ฟังก์ชั่นแบบทางเดียวบนพื้นฐานของ MD4 (หรืออาจเป็น MD5)

มีการกำหนดฟังก์ชั่นแบบทางเดียว คือ ใช้ข้อมูล 8 ไบต์เป็นอินพุต และสร้างข้อมูล 8 ไบต์เป็นเอาต์พุต โดยการนำข้อมูล 8 ไบต์เป็นอินพุตเข้า MD4 จะได้เอาต์พุตเป็น 16 ไบต์ และนำ 16 ไบต์ตัดให้เหลือ 8 ไบต์โดยการนำ 8 ไบต์แรกมา exclusive-OR กับ 8 ไบต์สุดท้าย ก็จะได้ฟังก์ชั่นแบบทางเดียว

3.2.3 การสร้างรหัสผ่านแบบทางเดียว

ลำดับของรหัสผ่านแบบใช้ครั้งเดียวถูกสร้างโดยการประยุกต์จากการใช้รหัสผ่านที่เข้ารหัสแบบทางเดียวหลายๆ ครั้ง ครั้งแรก รหัสผ่านแบบเข้ารหัสทางเดียวจะถูกสร้างโดยการเข้ารหัสของผู้ใช้ระบบ (s) แบบทางเดียวหลายๆ ครั้ง (n) สมมติว่าเป็น n = 4

$$p(1) = f(f(f(f(s))))$$

รหัสผ่านที่ใช้ในครั้งต่อไป จะถูกสร้างโดยการนำรหัสผ่านของผู้ใช้ระบบเข้ารหัสผ่านแบบทางเดียว จำนวน n-1 ครั้ง

$$p(2) = f(f(f(s)))$$

ผู้ลักลอบดักฟังที่คอยจะใช้รหัสผ่านของผู้ใช้ระบบ ที่ p(i) จะไม่สามารถสร้างรหัสผ่านที่ p(i+1) ได้เลย เนื่องจากต้องการแปลงฟังก์ชั่นกลับ แต่ไม่รู้คีย์ที่เป็นความลับ (secret key) ซึ่งเป็นจุดเริ่มต้นของการทำฟังก์ชั่นแบบซ้ำๆ

3.2.4 การตรวจสอบรหัสผ่านของระบบ

ขั้นแรกคอมพิวเตอร์ฝั่งโฮสต์จะเก็บสำเนาของรหัสผ่านแบบทางเดียวที่ได้รับมา แล้วทำการเข้ารหัส ถ้าผลลัพธ์ไม่ตรงกับสำเนาที่ได้เก็บไว้ในไฟล์เก็บรหัสผ่านของระบบ การร้องขอจะล้มเหลว ถ้าผลลัพธ์ตรง ไฟล์เก็บรหัสผ่านของระบบจะถูกปรับปรุงด้วยสำเนาของรหัสผ่านที่ได้เก็บก่อนที่จะมีการเข้ารหัสครั้งสุดท้าย (โดยเซิร์ฟเวอร์)

เนื่องจากจำนวนการเข้ารหัสแบบทางเดียวนั้น จะลดลงทีละ 1 ต่อการส่งรหัสผ่าน 1 ครั้ง ณ จุดใดจุดหนึ่งผู้ใช้ระบบจะต้องเริ่มต้นใหม่ ถ้าไม่เช่นนั้นก็จะไม่สามารถเข้าสู่ระบบได้อีก การเริ่มต้นใหม่นั้นทำได้โดยประมวลผลคำสั่ง passwd ที่พิเศษเพื่อเริ่มลำดับของรหัสผ่านใหม่การทำงานแบบนี้มีความสำคัญต่อการพิสูจน์ตนแบบธรรมดา ยกเว้นเมื่อได้รับรหัสผ่านแบบทางเดียวผ่านทางระบบเครือข่ายที่ไม่ได้ทำการตรวจสอบแถวที่มีอยู่ในไฟล์เก็บรหัสผ่านก่อนที่จะเก็บทับลงไป ด้วยวิธีนี้ การเลือกรหัสผ่านใหม่สามารถทำได้อย่างปลอดภัย

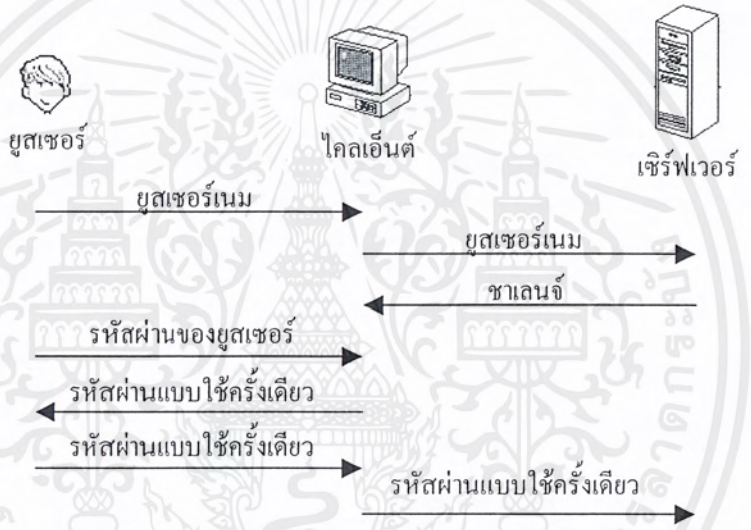
3.2.5 การทำงานของ S/KEY

จะแสดงตัวอย่างการล็อกอินเข้าสู่ระบบยูนิกซ์โดยใช้ระบบ S/KEY สมมติใช้แฮนด์เฮลด์พีซี (hand-held PC) คอมพิวเตอร์

- ผู้ใช้ระบบบอกชื่อตัวเองต่อระบบด้วยการใส่ล็อกอินเนม
- ระบบจะส่งข้อมูลมาซึ่งประกอบด้วยลำดับของการเข้ารหัสของรหัสผ่าน และซีด (seed) ซึ่งจะไม่มีการซ้ำกันในแต่ละระบบซีดจะให้ผู้ใช้ระบบมีความปลอดภัยโดยใช้ความลับที่รู้กันระหว่างโฮลต์และผู้ใช้ระบบ ในที่นี้ซีดคือ “unix3” และลำดับการเข้ารหัสคือ 54
- ผู้ใช้ระบบพิมพ์ข้อมูล 54 และ unix3 ลงไปที่แฮนด์เฮลด์พีซีคอมพิวเตอร์ ระบบจะถามถึงรหัสผ่านของผู้ใช้ระบบ
- ผู้ใช้ระบบพิมพ์รหัสผ่านซึ่งมีความยาวเท่าไรก็ได้แฮนด์เฮลด์พีซีคอมพิวเตอร์จะคำนวณรหัสผ่านที่เข้ารหัส 54 ครั้ง และแสดงผลรหัสผ่านที่เข้ารหัสครั้งที่ 54 ทางหน้าจอ
- ผู้ใช้ระบบส่งรหัสผ่านก็จะพิสูจน์ตนสำเร็จ
- เมื่อผู้ใช้ระบบต้องการติดต่อครั้งต่อไป เขาก็จะถูกถามถึงรหัสผ่านแบบใช้ครั้งเดียวลำดับที่

3.2.6 ขั้นตอนการพิสูจน์ตน

1. ยูสเซอร์ใส่ล็อกอินเนม
2. เซิร์ฟเวอร์ส่งชาเลนจ์ (Challenge) โดยประกอบด้วยซีเควนซ์นัมเบอร์ (Sequence Number) และซีด
3. ยูสเซอร์ใส่รหัสผ่านเพื่อให้เครื่องเทอร์มินอลคำนวณรหัสผ่านแบบใช้ครั้งเดียว (one-time password) จาก ซีเควนซ์นัมเบอร์และซีดโดยนำ ซีดเข้ารหัสเป็นจำนวนครั้งเท่ากับซีเควนซ์นัมเบอร์
4. ยูสเซอร์ใส่รหัสผ่านแบบใช้ครั้งเดียวที่เครื่องเทอร์มินอลคำนวณได้
5. เมื่อต้องการเข้าถึงระบบครั้งต่อไป จะใช้รหัสผ่านแบบใช้ครั้งเดียวที่ได้จากการเข้ารหัสซีดเป็นจำนวน ซีเควนซ์นัมเบอร์ – 1 ครั้ง



รูปที่ 3-7 ขั้นตอนการพิสูจน์ตนโดยใช้ระบบ S/Key

บทที่ 4
ระบบพิสูจน์ตนแบบที่มีตัวกลางในการพิสูจน์

4.1 ระบบ NIS และ NIS+

4.1.1 ระบบ NIS

ระบบ NIS (Network Information Service) มุ่งเน้นไปที่การจัดการระบบ โดยมีศูนย์กลางในการควบคุมระบบ NIS เก็บข้อมูลต่างๆ เกี่ยวกับชื่อเครื่อง (workstation name) แอดเดรส (address) ข้อมูลเกี่ยวกับผู้ใช้ระบบ (user) ข้อมูลเกี่ยวกับเครือข่าย (network) และการให้บริการในระบบเครือข่าย (network service) ข้อมูลเหล่านี้รวมเรียกว่า เนมสเปซ (namespace)

4.1.1.1 สถาปัตยกรรมของ NIS

NIS เป็นสถาปัตยกรรมแบบไคลเอ็นต์-เซิร์ฟเวอร์ (client-server) โดย NIS เซิร์ฟเวอร์จัดการให้บริการแก่ NIS ไคลเอ็นต์ NIS เซิร์ฟเวอร์หลักเรียกว่า มาสเตอร์เซิร์ฟเวอร์ (master server) และมีสลาฟเซิร์ฟเวอร์ (slave server) เก็บข้อมูลสำรองเพื่อความน่าเชื่อถือของระบบ

NIS ใช้โดเมน (domain) ในการจัดการ เครื่อง ผู้ใช้ระบบ และเครือข่ายในเนมสเปซ แต่โดเมนของ NIS นั้นไม่มีการจัดเป็นลำดับชั้น (hierachy) ดังนั้นเครือข่ายทางกายภาพที่มีการจัดเป็นลำดับขั้น จึงถูก NIS จัดการรวมให้เป็น 1 โดเมน

4.1.1.2 NIS maps

NIS เก็บข้อมูลในกลุ่มของเอกสาร (file) เรียกว่าแม็ป (map) ซึ่งมีการออกแบบมาเพื่อแทน /etc ของระบบปฏิบัติการยูนิกซ์ NIS เก็บข้อมูลมากมาย ซึ่งทำให้ NIS เนมสเปซประกอบด้วยแม็ปมากมาย

NIS แม็ปเป็นฐานข้อมูลแบบ distubuted ซึ่งเป็นตารางที่ประกอบด้วย 2 สดมภ์ โดยสดมภ์แรกเป็นคีย์ (key) NIS หาข้อมูลโดยการค้นหาผ่านคีย์ ดังนั้นข้อมูลบางอย่างจึงต้องเก็บอยู่ในหลายแม็ป เนื่องจากเป็นแม็ปที่ใช้คนละคีย์ในการค้นหา เช่น ชื่อ และ แอดเดรส ของเครื่อง เก็บอยู่ใน 2 แม็ป คือ host.byname และ host.byaddr โดยเมื่อเซิร์ฟเวอร์รู้ชื่อของเครื่อง และต้องการรู้แอดเดรส ต้องไปค้นหาใน host.byname และเมื่อเซิร์ฟเวอร์รู้แอดเดรสของเครื่อง และต้องการรู้ชื่อเครื่อง จะไปค้นหาใน host.addr

4.1.1.3 เซิร์ฟเวอร์

เซิร์ฟเวอร์แบ่งออกเป็น 2 ชนิด คือ มาสเตอร์เซิร์ฟเวอร์ และสลาฟเซิร์ฟเวอร์

มาสเตอร์เซิร์ฟเวอร์

ใน 1 โดยเมน มีมาสเตอร์เซิร์ฟเวอร์อยู่ 1 ตัว และให้คำสั่ง ypupdated ทำงาน เพื่อบอกสลาฟเซิร์ฟเวอร์ให้มีการปรับปรุงให้เหมือนตัวมาสเตอร์ทั้งหมดหรือแค่บางตาราง มาสเตอร์เซิร์ฟเวอร์ควรมีคุณสมบัติดังนี้

- สามารถให้ผู้ดูแลระบบเข้าถึงได้ง่าย
- มีเสถียรภาพสูง สามารถทำงานได้อย่างมีประสิทธิภาพแม้มีผู้ใช้ในระบบมากก็ตาม
- สามารถเข้าถึงได้ง่ายจากหลายเครือข่าย แม้มีผู้ติดต่อจากระยะไกลโดยผ่านเกตเวย์ (gateway) และบริดจ์ (bridge) มากมาย

ในระบบขนาดใหญ่ อาจมีผู้ติดต่อกับมาสเตอร์เซิร์ฟเวอร์มากจนอาจทำให้มาสเตอร์เซิร์ฟเวอร์ไม่สามารถรับคำร้องขอได้ทัน และถ้ามีผู้ส่งคำร้องขอบริการในระบบมากเกินไป อาจทำให้มาสเตอร์เซิร์ฟเวอร์ไม่ทำงานได้ จึงต้องมีเซิร์ฟเวอร์อื่นคอยช่วยเหลือในระบบ นั่นคือ สลาฟเซิร์ฟเวอร์

สลาฟเซิร์ฟเวอร์

สลาฟเซิร์ฟเวอร์เป็นตัวกลางระหว่างมาสเตอร์เซิร์ฟเวอร์กับไคลเอ็นต์ โดยการเก็บข้อมูลต่างๆ ที่มาสเตอร์เซิร์ฟเวอร์เก็บ ทุกอย่างที่เกิดความเปลี่ยนแปลงที่มาสเตอร์เซิร์ฟเวอร์ จะมีการกระจายไปให้สลาฟเซิร์ฟเวอร์ทุกตัว สลาฟเซิร์ฟเวอร์สามารถตอบรับการร้องขอจากผู้ใช้ระบบได้เหมือนกับมาสเตอร์เซิร์ฟเวอร์ทุกอย่าง จึงสามารถช่วยลดภาระของมาสเตอร์เซิร์ฟเวอร์ และสามารถทำงานแทนมาสเตอร์เซิร์ฟเวอร์ได้ในกรณีที่มาสเตอร์เซิร์ฟเวอร์ไม่สามารถใช้งานได้

โดยปกติแล้ว ทุกโดเมนมีสลาฟเซิร์ฟเวอร์อยู่อย่างน้อย 1 ตัว จำนวนของสลาฟเซิร์ฟเวอร์พิจารณาจากเวลาจากการตอบรับหลังจากที่ติดต่อไป (response time) ถ้าจำนวนสลาฟเซิร์ฟเวอร์มาก ทำให้เวลาในการตอบรับน้อยลง แต่ทำให้ต้องมีการสำเนาข้อมูลต่างๆ ทุกสลาฟเซิร์ฟเวอร์ ซึ่งถ้ามีการสำเนามาก ในการปรับปรุงให้เหมือนกันมาสเตอร์เซิร์ฟเวอร์จะยุ่งยากขึ้น

4.1.1.4 ข้อเสียของระบบ NIS

1. ระบบ NIS ไม่สามารถรองรับไคลเอ็นต์และเซิร์ฟเวอร์จำนวนมากได้
2. การติดต่อไม่มีความปลอดภัย
3. ผู้ดูแลระบบต้องเปลี่ยนแปลงข้อมูลเอง

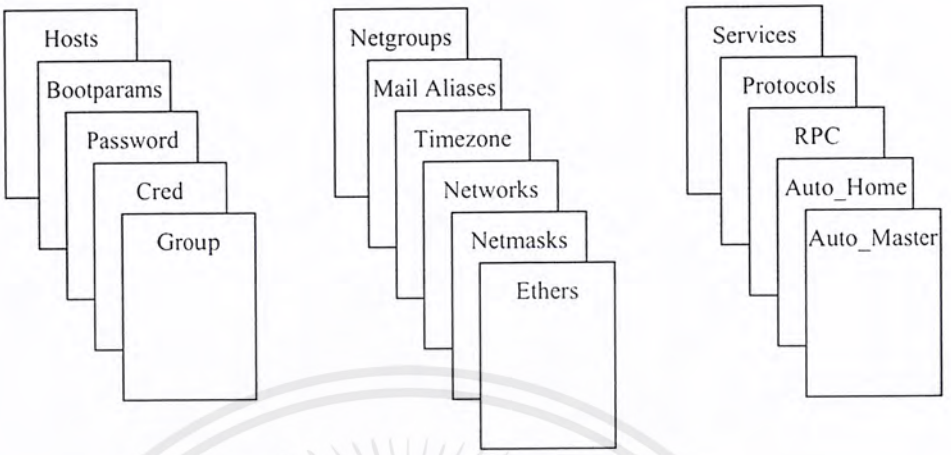
4.1.2 ระบบ NIS+

4.1.2.1 การเก็บข้อมูลของ NIS+

NIS+ ได้ถูกออกแบบมาเพื่อทดแทน NIS NIS+ เนมสเปซออกแบบให้โดเมนมีการแบ่งเป็นลำดับชั้น ดังนั้นจึงมีการจัดการได้ง่าย เช่น บริษัทต้องการแบ่งออกเป็น 2 ส่วน NIS+ เนมสเปซก็สามารถแบ่งได้เป็น 2 โดเมน ซึ่งสามารถจัดการได้อย่างอัตโนมัติ

มีสิ่งหนึ่งที่ NIS+ แตกต่างกับ NIS โดยสิ้นเชิง เนื่องจากข้อมูลที่ใช้โดย NIS ไม่ค่อยมีการเปลี่ยนแปลงดังนั้น การปรับปรุงให้ข้อมูลของสลาฟเซิร์ฟเวอร์เหมือนกับมาสเตอร์เซิร์ฟเวอร์นั้น จึงให้คนเป็นผู้ทำ ต่อมาระบบเครือข่ายมีการขยายตัว ข้อมูลต้องมีการปรับปรุงบ่อยขึ้น NIS+ มีการปรับปรุงข้อมูลให้สลาฟเซิร์ฟเวอร์โดยอัตโนมัติ ทำให้สามารถมั่นใจได้ว่าข้อมูลที่สลาฟเซิร์ฟเวอร์เหมือนกับมาสเตอร์เซิร์ฟเวอร์

NIS+ เก็บข้อมูลในตารางแบ่งได้เป็น 16 ตาราง



รูปที่ 4-1 แสดงตารางข้อมูลต่างๆ ใน NIS+

โดยในตารางต่างๆ เก็บข้อมูลดังนี้

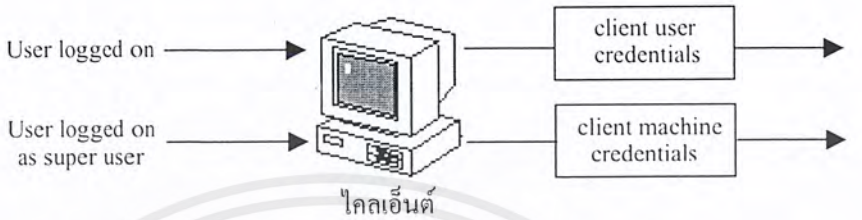
Hosts	แอดเดรสในเครือข่ายและชื่อทุกเครื่องในโดเมน
Bootparams	ที่เก็บของรูต สวอพ และไคลเอ็นต์ทุกตัวที่ไม่มีดิสก์ ในโดเมน
Password	ข้อมูลเกี่ยวกับรหัสผ่านของผู้ใช้ระบบทุกคนในโดเมน
Cred	ตัวของแต่ละพรินซิพอล (principal) ซึ่งเป็นเจ้าของโดเมน
Group	รหัสผ่านของกลุ่ม, หมายเลขประจำตัวกลุ่ม และสมาชิกกลุ่ม UNIX Group ในโดเมน
Netgroup	Netgroup ของเครื่องและผู้ใช้ระบบ
Aliases	ข้อมูลเกี่ยวกับชื่อปลอมของเมล์ของผู้ใช้ระบบในโดเมน
Timezone	ไทม์โซนของทุกๆ เครื่องในโดเมน
Networks	เครือข่ายต่างๆ ในโดเมนและชื่อที่ถูกต้องของเครือข่าย
Netmasks	เครือข่ายในโดเมน และเน็ตมาส์ก (Netmasks) ที่เกี่ยวข้อง
Ethers	อีเธอร์เน็ทแอดเดรสของทุกๆ เครื่องในโดเมน
Protocols	แสดงถึงโพรโตคอล IP ทุกโพรโตคอลที่ใช้ในโดเมน
RPC	หมายเลข RPC โปรแกรมสำหรับบริการต่างๆ ของ RPC ที่มีในโดเมน
Auto_Home	ที่อยู่ของโฮมไดเรกทอรีของผู้ใช้ระบบในโดเมน
Auto_Master	ข้อมูลที่เกี่ยวกับ Auto Master แม็ป

แต่ละตารางเก็บข้อมูลที่แตกต่างกัน เช่น ตาราง Password เก็บข้อมูลเกี่ยวกับผู้ใช้ระบบในเครือข่าย

ตาราง NIS+ นั้นเป็นฐานข้อมูลแบบ relational สามารถเข้าถึงได้ทุกสดมภ์ ไม่ใช่แค่สดมภ์แรก (ที่เป็นคีย์) สามารถกำจัดไฟล์ที่เก็บซ้ำซ้อนได้ เช่น host.byname และ host.byaddr ที่ใช้โดย NIS

4.1.2.2 พรินซิพอลและไคลเอ็นต์ของ NIS+

NIS พรินซิพอลคือ ผู้ที่ล็อกออนเข้าสู่ไคลเอ็นต์ หรือผู้ที่ล็อกเป็นซูเปอร์ยูสเซอร์ (superuser) และส่งคำร้องขอบริการจาก NIS+ โดยผ่านทาง NIS+ ไคลเอ็นต์ หรือ NIS+ แมชีน (NIS+ machine) โดยผู้ใช้ธรรมดาล็อกผ่านไคลเอ็นต์ยูสเซอร์ (client user) และซูเปอร์ยูสเซอร์ล็อกผ่านไคลเอ็นต์แมชีน



รูปที่ 4-2 แสดงตัวที่ใช้ของผู้ใช้ระบบและซูเปอร์ยูสเซอร์

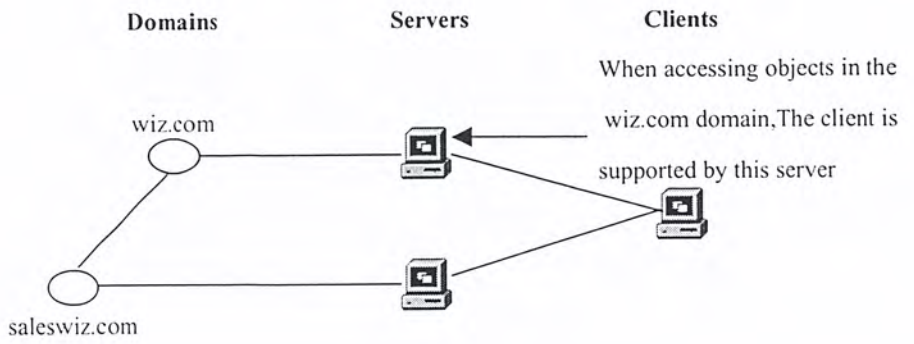
NIS+ ไคลเอ็นต์ คือ เครื่องที่ได้รับการติดตั้งให้สามารถรับการบริการจาก NIS+ เซิร์ฟเวอร์ได้ ในการติดตั้ง NIS+ ไคลเอ็นต์ต้องสร้างตัว จัดให้เป็นสมาชิกอยู่ในกลุ่มของ NIS+ ที่ถูกต้อง และสุดท้ายต้องให้สคริปเริ่มต้น (initial script) ทำงาน

แม้ NIS+ ไคลเอ็นต์สามารถเข้าถึงทุกส่วนของเนมสเปซ แต่ 1 ไคลเอ็นต์อยู่ได้แค่ 1 โดเมน ซึ่งนั่นคือโฮมโดเมน (Home Domain) โฮมโดเมนของไคลเอ็นต์มักถูกกำหนดระหว่างการติดตั้ง แต่สามารถเปลี่ยนภายหลังได้ ข้อมูลทุกอย่างที่เกี่ยวกับไคลเอ็นต์ เช่น IP แอดเดรสและตัวของไคลเอ็นต์ เก็บอยู่ในตารางของ NIS+ ที่โฮมโดเมน

การแยกระหว่าง NIS+ ไคลเอ็นต์กับการที่ได้เก็บข้อมูลอยู่ในตารางของ NIS+ นั้นเป็นการยาก สามารถอธิบายได้คร่าวๆ คือ การที่เก็บข้อมูลของเวิร์กสเตชั่นลงในตารางของ NIS+ ไม่ได้ทำให้เวิร์กสเตชั่นนั้นเป็น NIS+ ไคลเอ็นต์โดยอัตโนมัติ แต่ข้อมูลเหล่านั้นช่วยให้เวิร์กสเตชั่นนั้นเป็น ไคลเอ็นต์ได้ เวิร์กสเตชั่นนั้นไม่ได้รับบริการจาก NIS+ เซิร์ฟเวอร์ถ้าหากว่าเวิร์กสเตชั่นนั้ไม่ได้รับการติดตั้งให้เป็นไคลเอ็นต์

ในทางกลับกัน การทำไคลเอ็นต์ให้เป็นเวิร์กสเตชั่น NIS+ ไคลเอ็นต์ไม่จำเป็นต้องใส่ข้อมูลใดๆ เกี่ยวกับเวิร์กสเตชั่นลงในตารางของ NIS+ เวิร์กสเตชั่นนั้นสามารถรับบริการของ NIS+ เซิร์ฟเวอร์ได้เลย ถ้าข้อมูลของเวิร์กสเตชั่นนั้นไม่ชัดเจนเนื่องจากการใส่ข้อมูลในตารางโดยผู้ดูแลระบบ ไคลเอ็นต์อื่นไม่สามารถตรวจสอบข้อมูลเวิร์กสเตชั่นนั้นได้

เมื่อไคลเอ็นต์ต้องการเข้าถึงข้อมูลในเนมสเปซ มันจะส่งคำร้องขอไปที่โดเมนที่เจาะจงไว้ในเนมสเปซ ดังนั้นไคลเอ็นต์จะส่งคำร้องขอไปยังเซิร์ฟเวอร์ที่สนับสนุนโดเมนนั้น การทำงานสามารถอธิบายได้คร่าวๆ ดังรูป



รูปที่ 4-3 แสดงไคลเอ็นส่งคำร้องขอเพื่อเข้าถึงเนมสเปซ

NIS+ ไคลเอ็นต์รู้ว่าควรติดต่อไปยังเซิร์ฟเวอร์ใดด้วยการลองผิดทดลองถูก โดยเริ่มต้นจากโฮมเซิร์ฟเวอร์ของไคลเอ็นต์ ต่อจากนั้นก็ลองไปเรื่อยๆ จนกว่าจะเจอเซิร์ฟเวอร์ที่ถูกต้อง เมื่อติดต่อผิดเซิร์ฟเวอร์แล้ว ไคลเอ็นต์มีแคชเพื่อเก็บข้อมูลต่างๆ เพื่อให้การค้นหามีประสิทธิภาพมากขึ้น

4.1.2.3 กลไกในการรักษาความปลอดภัยของระบบ NIS+

NIS+ มีระบบรักษาความปลอดภัยที่ซับซ้อน เพื่อป้องกันข้อมูลต่างๆ ที่เก็บรักษา โดยใช้การพิสูจน์ตนและการพิสูจน์สิทธิ์ โดยการพิสูจน์ตนเป็นการบ่งบอกว่าเป็นผู้ใช้ในระบบเครือข่ายหรือไม่ และการพิสูจน์สิทธิ์บ่งบอกว่าผู้ใช้ระบบนั้นได้รับอนุญาตให้สามารถเข้าถึงข้อมูลได้หรือไม่ และระดับใด

ระดับรักษาความปลอดภัยสามารถแบ่งได้ดังนี้

ระดับ 0 ถูกออกแบบเพื่อใช้ในการทดสอบ และติดตั้งระบบ ถ้า NIS+ วิ่งที่ระดับ 0 ไคลเอ็นต์ต่างๆ จะได้รับอนุญาตให้สามารถทำทุกอย่างได้ในโดเมน

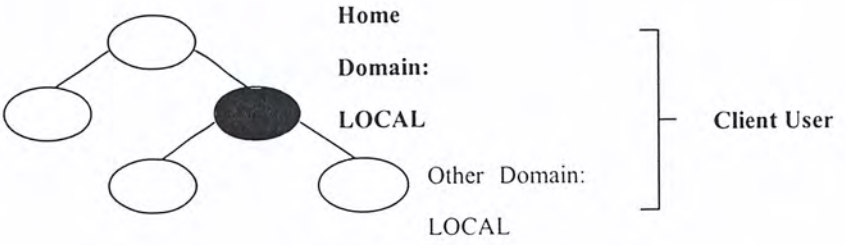
ระดับ 1 ใช้โลคอล (LOCAL) หรือ DES ที่ไม่สลับซับซ้อน เป็นตั๋วในการแสดงตน ถ้าไม่มีตั๋วจะจัดอยู่ในคลาส nobody ไม่ควรใช้ในเครือข่ายที่ไม่น่าไว้วางใจ

ระดับ 2 เป็นระดับการรักษาความปลอดภัยที่ใช้ทั่วไปในระบบ NIS+ ใช้ตั๋วแบบ DES ถ้าใช้ตั๋วแบบโลคอลหรือไม่มีตั๋ว ให้จัดอยู่ในคลาส nobody

4.1.2.4 การพิสูจน์ตนและตั๋วของ NIS+

ชนิดของตั๋ว มีอยู่ 2 ชนิด คือโลคอลและ DES ผู้ใช้ระบบสามารถใช้ได้ทั้ง 2 ชนิด แต่เครื่องสามารถใช้ได้เฉพาะ DES

ตั๋วแบบ DES ข้อมูลสามารถเก็บอยู่ในตาราง Cred ที่โฮมโดเมนเท่านั้น ไม่ว่าเป็นไคลเอ็นต์ยูสเซอร์หรือไคลเอ็นต์แมชีน ส่วนตั๋วแบบโลคอลเก็บไว้ที่ใดก็ได้ในโดเมน ในความเป็นจริงแล้ว ในการล็อกเข้าสู่รีโมตโดเมน ไคลเอ็นต์ต้องเก็บตั๋วแบบโลคอลไว้ในตาราง Cred ของรีโมตโดเมน



รูปที่ 4-4 แสดงการเก็บข้อมูลของ ตั๋วแบบ DES และตั๋วแบบโลคอล

4.1.2.5 การพิสูจน์สิทธิ์และการเข้าถึงของระบบ NIS+

Authorization Class

ในความเป็นจริงแล้ว NIS+ ไม่ได้อนุญาตที่ตัวไคลเอ็นต์โดยตรง แต่ว่าอนุญาตเป็นคลาสต่างๆ ดังนี้

Owner Class

ผู้ที่เข้าถึงข้อมูลใน NIS+ ต้องมีการแสดงตนก่อน จึงได้รับอนุญาตให้อยู่ในคลาสนี้

โดยปกติแล้วผู้ที่สร้างวัตถุ (วัตถุ คือ ข้อมูลที่ต้องการเก็บลงฐานข้อมูล อาจเป็นระดับ เรคคอร์ด สคมภ์ หรือ ตาราง) นั้นขึ้นมา ได้รับอนุญาตให้อยู่ในคลาสนี้ แต่ผู้ที่สร้างสามารถยกความเป็นเจ้าของให้ผู้อื่นได้ 2 วิธี

- ผู้สร้างวัตถุนั้นเจาะจงความเป็นเจ้าของซึ่งเป็นผู้อื่นในขณะสร้างวัตถุนั้นขึ้นมา
- ผู้สร้างสามารถเปลี่ยนความเป็นเจ้าของให้ผู้อื่นได้ หลังจากที่สร้างวัตถุนั้นแล้ว

Group Class

ผู้ที่เข้าถึงข้อมูลใน NIS+ ต้องมีการแสดงตนก่อน และเป็นเจ้าของคลาสนี้ จึงได้รับอนุญาตให้อยู่ในคลาสนี้ได้

group ใน NIS+ เป็นการรวมกันของเครื่องต่างๆ มีการรวมกลุ่มกันเพื่อความสะดวกในการจัดการการเข้าถึงข้อมูลในเนมสเปซ การอนุญาตให้เข้าถึงวัตถุในคลาสนี้นั้นจะเป็นกับทุกๆ เครื่องที่เป็นสมาชิกในคลาสนี้ (class owner นั้น ก็สามารถเข้าถึงข้อมูลได้โดยไม่จำเป็นต้องอยู่ในคลาสนี้)

เมื่อวัตถุหนึ่งถูกสร้างขึ้น อาจมีการกำหนดให้เป็นกลุ่มใดกลุ่มหนึ่งที่กำหนดไว้เป็นมาตรฐาน แต่สามารถกำหนดกลุ่มใหม่ได้ตลอดเวลา

World Class

ในคลาสนี้ประกอบด้วยกลุ่มผู้ใช้ระบบทุกคนที่มีการแสดงตนกับ NIS+ นั่นหมายความว่า ทุกคนในกลุ่ม owner และ group class และประกอบกับทุกคนซึ่งมีการแสดงตนโดยใช้ตั๋วแบบ DES ที่ถูกต้อง

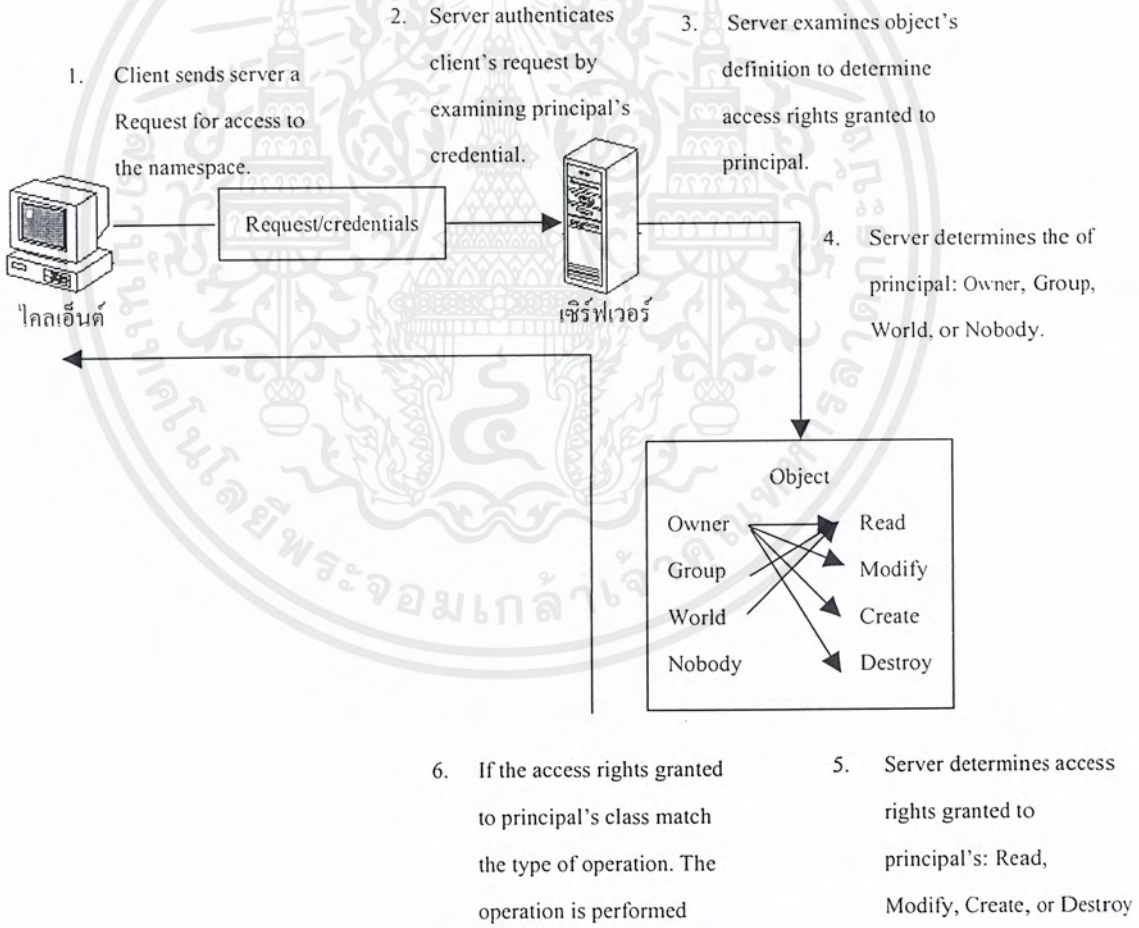
Nobody Class

ในคลาสนี้ประกอบด้วยทุกๆ คนที่ไม่ได้มีการแสดงตน นั่นหมายความว่า ทุกคนที่ไม่ได้แสดงตั๋วแบบ DES ที่ถูกต้อง

4.1.2.6 Access Right

NIS+ มีรูปแบบคล้ายกับระบบ UNIX ใช้สำหรับอนุญาตการกระทำต่างๆ ของผู้ใช้ระบบ มีอยู่ 4 ชนิด

1. อ่าน (Read) ผู้ที่ได้รับการอนุญาตให้อ่าน สามารถดูข้อมูลนี้ได้
2. แก้ไข (Modify) ผู้ที่ได้รับการอนุญาตให้แก้ไข สามารถเปลี่ยนแปลงข้อมูลนั้นได้
3. ทำลาย (Destroy) ผู้ที่ได้รับการอนุญาตให้ทำลาย สามารถทำลายข้อมูลนั้นทิ้งไปได้
4. สร้าง (Create) ผู้ที่ได้รับการอนุญาตให้สร้าง สามารถสร้างวัตถุใหม่ได้ โดยแบ่งเป็น 3 ระดับ คือระดับ directory ระดับตาราง และระดับสคมภ์ ตามลำดับ ถ้าได้รับอนุญาตให้สามารถสร้างในระดับที่สูง ก็สามารถสร้างลำดับที่ต่ำลงมาได้ด้วย เช่นได้รับอนุญาตให้สามารถสร้างไดเรกทอรีก็สามารถสร้างตารางได้



รูปที่ 4-5 แสดงขั้นตอนการล็อกอินและการจัดคลาสให้แก่ผู้ใช้ระบบ

4.1.3 ข้อแตกต่างระหว่าง NIS และ NIS+

NIS	NIS+
1. ชื่อเครื่องและชื่อผู้ใช้ระบบสามารถเหมือนกันได้	1. ชื่อเครื่องและชื่อผู้ใช้ระบบไม่สามารถเหมือนกันได้และห้ามมีจุด (.)
2. โดเมนไม่มีการจัดเป็นลำดับชั้น	2. โดเมนมีการจัดเป็นลำดับชั้น ข้อมูลเก็บอยู่ในระดับที่ต่างกันในเนมสเปซ
3. ชื่อและคำสั่งเป็นแบบ case sensitive	3. ชื่อและคำสั่งไม่เป็นแบบ case sensitive
4. ข้อมูลเก็บอยู่ในตาราง 2 สดมภ์	4. ข้อมูลเก็บอยู่ในตารางหลายสดมภ์
5. ไม่มีการแสดงตน	5. ใช้การแสดงตนแบบ DES
6. ใน 1 เรคคอร์ดห้ามเกิน 1024 byte	6. ไม่จำกัด
7. ปรับปรุงข้อมูลของสลาฟเซิร์ฟเวอร์ให้ตรงกับมาสเตอร์เซิร์ฟเวอร์โดยใช้คนจัดการ	7. ปรับปรุงข้อมูลของสลาฟเซิร์ฟเวอร์ให้ตรงกับมาสเตอร์เซิร์ฟเวอร์โดยอัตโนมัติ

ตารางที่ 4-1 แสดงข้อแตกต่างระหว่าง NIS และ NIS+

4.2 Kerberos

โพรโตคอล Kerberos เป็นกระบวนการในการพิสูจน์ตนระหว่าง 2 ฝ่าย ก่อนที่จะมีการเชื่อมต่อกันเกิดขึ้น เช่นการพิสูจน์ตนระหว่างไคลเอ็นต์กับเซิร์ฟเวอร์ หรือระหว่างเซิร์ฟเวอร์ตัวหนึ่งกับเซิร์ฟเวอร์อีกตัวหนึ่ง โพรโตคอลนี้จะมีข้อสมมติว่า การติดต่อกันที่เกิดขึ้นระหว่างไคลเอ็นต์กับเซิร์ฟเวอร์นั้น เกิดขึ้นบนระบบเปิดที่มีความปลอดภัยไม่เพียงพอ และข้อมูลที่ถูกส่งไป ตามเครือข่ายนั้น สามารถที่ถูกแอบดูหรือถูกแก้ไขได้โดยผู้ที่ไม่เกี่ยวข้อง

4.2.1 คำศัพท์ต่างๆ ที่เกี่ยวข้องกับขั้นตอนในการพิสูจน์ตนของ Kerberos

Authenticator

เมื่อมีคนต้องการจะเข้ามาในระบบ คนที่ต้องการเข้ามานั้นจะต้องแสดงออเธนติเคเตอร์ (Authenticator) เพื่อแสดงว่าตนเองเป็นยูสเซอร์ที่ถูกต้อง โดยออเธนติเคเตอร์จะถูกเข้ารหัสด้วยคีย์ที่เป็นความลับและข้อมูลในออเธนติเคเตอร์นั้น จะเปลี่ยนแปลงทุกครั้งที่ใช้งาน เพื่อป้องกันการแอบดูข้อมูลแล้วนำข้อมูลนี้กลับมาใช้ใหม่ ทางผู้ดูแลการขอเข้าระบบก็จะถอดรหัสข้อมูลเหล่านี้ ถ้าสามารถถอดได้ถูกต้องก็แสดงว่า คนที่ส่งออเธนติเคเตอร์นั้นมามีคีย์ที่ถูกต้อง และเป็นยูสเซอร์ที่ถูกต้องด้วย

ถ้าผู้ที่อยู่นอกระบบต้องการให้ผู้ตรวจสอบการขอเข้าระบบพิสูจน์ตนเช่นกัน ก็จะสามารถทำได้โดย โพรโตคอลเดียวกันนี้ แต่ทำแบบตรงกันข้าม คือ ผู้ตรวจสอบการขอเข้าระบบจะตัดเอาข้อมูลบางส่วนจากออเธนติเคเตอร์มา แล้วส่งกลับไปให้ยูสเซอร์ ยูสเซอร์ก็จะตรวจสอบข้อมูลนี้ว่าตรงกับข้อมูลเดิมที่ส่งไปให้หรือไม่ ถ้าตรงกันก็แสดงว่า ผู้ดูแลการขอเข้าระบบได้รับออเธนติเคเตอร์เรียบร้อยแล้ว

Key Distribution

KDC (Key Distribution Center) จะมีหน้าที่เป็นสื่อกลางระหว่างไคลเอ็นต์กับเซิร์ฟเวอร์ ซึ่ง KDC จะเป็นเซอร์วิซที่ทำงานอยู่บนเซิร์ฟเวอร์ที่มีความปลอดภัย มันจะเก็บข้อมูลของพรินซิพอล (กลุ่มของบุคคลที่ถูกตรวจสอบคุณสมบัติ) ในขอบเขตของมัน นอกจากเก็บข้อมูลแล้ว KDC ก็จะเก็บคีย์ที่ใช้ในการเข้ารหัสไว้ด้วย โดยคีย์นี้จะรู้เฉพาะพรินซิพอลและ KDC เท่านั้นคีย์นี้จะใช้ในการแลกเปลี่ยนข้อมูลระหว่างพรินซิพอลและ KDC เราเรียกคีย์นี้ว่าลองเทอมคีย์ (Long Term Key) ซึ่งลองเทอมคีย์นี้จะได้จากพาสเวิร์ดของยูสเซอร์

Session Tickets

เมื่อไคลเอ็นต์ต้องการที่จะติดต่อกับเซิร์ฟเวอร์ไคลเอ็นต์จะต้องร้องขอไปยัง KDC ก่อน และ KDC ก็จะให้เซสชันคีย์ (Session Key) กลับไปยังไคลเอ็นต์เพื่อใช้เซสชันคีย์นี้ติดต่อกับเซิร์ฟเวอร์ โดยเซสชันคีย์ของไคลเอ็นต์จะถูก เข้ารหัสด้วยคีย์ที่ไคลเอ็นต์กับ KDC เท่านั้นที่รู้ (Long Term Key) ส่วนเซสชันคีย์ของเซิร์ฟเวอร์จะถูกฝังไว้ในข้อมูลที่ KDC จะส่งกลับไปให้ไคลเอ็นต์ด้วยเรียกว่าเซสชันทิคเกต (Session Ticket) โดยข้อมูลในเซสชันทิคเกตจะถูกเข้ารหัสด้วยคีย์ที่เซิร์ฟเวอร์กับ KDC เท่านั้นที่รู้ ซึ่งเซสชันทิคเกตนี้ไคลเอ็นต์จะใช้ในการขอบริการจากเซิร์ฟเวอร์

Credentials

เป็นข้อความที่ไคลเอ็นต์ส่งให้กับเซิร์ฟเวอร์เมื่อต้องการขอใช้บริการจากเซิร์ฟเวอร์ โดย Credential จะประกอบด้วยเซสชันทิคเกตที่ถูกเข้ารหัสด้วยลองเทอมคีย์ของเซิร์ฟเวอร์ และออเธนติเคเตอร์ที่ถูกเข้ารหัสด้วยเซสชันคีย์

Ticket – Granting Ticket

ลองเทอมคีย์ของยูสเซอร์นั้นจะได้มาจากพาสเวิร์ด คือเมื่อผู้ใช้ล็อกอินตัว Kerberos ไคลเอ็นต์ที่อยู่บนเวิร์กสเตชันจะรับพาสเวิร์ดนั้นมา แล้วเปลี่ยนเป็นคีย์ที่ใช้ในการเข้ารหัส โดยเอาพาสเวิร์ดที่อยู่ในรูป Text นี้ผ่านไป Hashing Function ผลที่ได้ออกมาก็จะเป็นลองเทอมคีย์ของยูสเซอร์

ตัว KDC จะเก็บลองเทอมคีย์ของผู้ใช้ไว้ในเรคอร์ดข้อมูลของผู้ใช้ในฐานข้อมูลของมัน เมื่อ KDC ได้รับการติดต่อจาก Kerberos ไคลเอ็นต์ KDC ก็จะหาลองเทอมคีย์ของยูสเซอร์ในฐานข้อมูลของมันและเอาลองเทอมคีย์นี้ออกมา

กระบวนการที่กล่าวมานี้จะเกิดขึ้นเมื่อยูสเซอร์ล็อกอินครั้งแรกเพียงครั้งเดียวเท่านั้นทันทีหลังจากที่ ได้รับพาสเวิร์ดของยูสเซอร์และเปลี่ยนมาเป็นลองเทอมคีย์ตัว Kerberos ไคลเอ็นต์บนเวิร์กสเตชันจะขอเซสชันทิคเกตและเซสชันคีย์ซึ่งมันจะใช้ในการติดต่อกับ KDC และ KDC ก็จะตอบสนองการร้องขอนี้โดยการส่งเซสชันทิคเกตกลับไปเซสชันคีย์นี้ จะเรียกว่า "Ticket – Granting Ticket" (TGT) TGT จะมีลักษณะเหมือนกับเซสชันทิคเกตธรรมดา ซึ่งใน TGT จะประกอบด้วยเซสชันคีย์ที่ KDC จะใช้ในการติดต่อกับไคลเอ็นต์ซึ่งถูกเข้ารหัสด้วยลองเทอมคีย์ของ KDC นอกจาก TGT แล้ว ในข้อมูลที่ KDC ส่งกลับไปให้ไคลเอ็นต์จะประกอบด้วยเซสชันคีย์ที่ไคลเอ็นต์ใช้ติดต่อกับ KDC อีกด้วย ซึ่งเซสชันคีย์นี้จะถูกเข้ารหัสด้วยลองเทอมคีย์ของไคลเอ็นต์

เมื่อไคลเอ็นต์ได้รับการตอบสนองกลับมาจาก KDC ไคลเอ็นต์จะใช้ลองเทอมคีย์มาถอดรหัสข้อความที่ได้รับมา ดังนั้นไคลเอ็นต์ก็จะมีเซสชันคีย์และมันก็จะใช้เซสชันคีย์ตัวนี้ติดต่อกับ KDC ในครั้งต่อๆ ไปโดยที่ไม่ต้องใช้ลองเทอมคีย์เลย แต่ตัว TGT นี้ก็เป็นเพียงแค่ตั๋วชั่วคราวเหมือนกับตัวอื่น มันจะใช้ไม่ได้เมื่อมันหมดอายุ หรือยูสเซอร์ล็อกออฟ

จากมุมมองของไคลเอ็นต์ TGT ก็จะเป็นตั๋วอันหนึ่ง ก่อนที่ไคลเอ็นต์จะทำการติดต่อกับเซิร์ฟเวอร์ใดๆ ก็ตาม ตัวไคลเอ็นต์จะเช็คในครีเดนเทียลแคช (Credential Cache) ก่อนว่ามีเซสชันทิคเกตที่จะต้องใช้ในการติดต่อกับเซอร์วิซนั้นอยู่หรือไม่ ถ้าไม่มี มันก็จะไปเช็คใน Cache อีกทีว่ามี TGT หรือไม่ ถ้าพบว่ามี TGT อยู่ไคลเอ็นต์จะเอาเซสชันคีย์ที่ใช้ติดต่อกับ KDC มาเข้ารหัสออเธนติเคเตอร์แล้วส่งออเธนติเคเตอร์พร้อมกับ TGT ไปให้ KDC เพื่อขอเซสชันคีย์สำหรับบริการนั้นๆ

ทางด้านฝ่าย KDC นั้น ก็จะทำเหมือนกับเซิร์ฟเวอร์ทั่วไป คือจะใช้ลองเทอมคีย์ของตัวเองถอดรหัส TGT จากนั้นก็จะเอาเซสชันคีย์ที่ได้จากการถอดรหัสด้วยลองเทอมคีย์ไปถอดรหัสข้อความที่ได้รับมาจากไคลเอ็นต์

4.2.2 หลักการทำงานของ Kerberos

Kerberos ประกอบด้วยโพรโตคอลย่อย 3 โพรโตคอล คือ

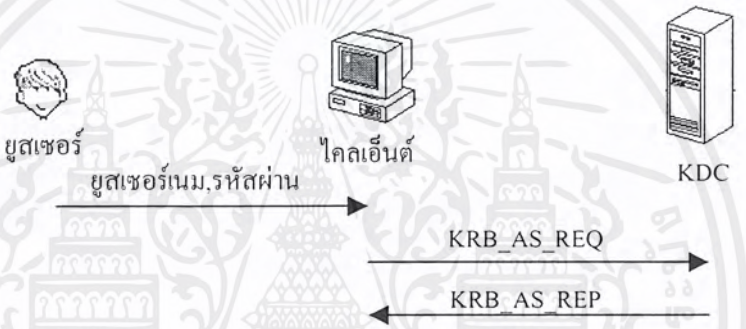
1. Authenticator Service (AS) Exchange
2. Ticket – Granting Service (TGS) Exchange
3. Client / Server (CS) Exchange

ซึ่งแต่ละโพรโตคอลย่อยมีหน้าที่ต่างๆ ดังนี้

4.2.2.1 AS Exchange เป็นโพรโตคอลย่อยที่ทำหน้าที่ในช่วงที่ KDC (Key Distribution Center) ให้เซสชันคีย์และ TGT (Ticket – Granting Ticket) แก่ไคลเอ็นต์โดยมีขั้นตอนการทำงานคือ

1. ยูสเซอร์ที่ต้องการพิสูจน์ตนต่อเซิร์ฟเวอร์ ใส่ยูสเซอร์เนมและพาสเวิร์ดลงในเครื่องไคลเอ็นต์
2. Kerberos ไคลเอ็นต์ที่เวิร์กสเตชันของยูสเซอร์จะแปลงพาสเวิร์ดนี้ ให้เป็นคีย์ที่ใช้ในการเข้ารหัส และเก็บคีย์นี้ไว้ในครีเดนเทียลแคช
3. ไคลเอ็นต์จะส่ง Kerberos Authentication Service Request (KRB_AS_REQ) ไปที่ KDC โดยส่วนแรกของข้อความที่ส่งไปจะแสดงถึงยูสเซอร์เนม และชื่อของบริการที่ยูสเซอร์ต้องการใช้ และในส่วนที่สองจะประกอบด้วยพรีออเธนติเคเตอร์ดาตา (Preauthentication Data) เพื่อแสดงให้ KDC รู้ว่า ยูสเซอร์รู้พาสเวิร์ด ซึ่ง ส่วนมากมักจะเป็นไทม์สแตมป์ (Timestamp) ที่ถูกเข้ารหัสด้วยคีย์ของยูสเซอร์
4. เมื่อ KDC ได้รับ KRB_AS_REQ KDC ก็จะหาข้อมูลของยูสเซอร์คนนี้ในฐานข้อมูลของมัน เพื่อเอาคีย์ของยูสเซอร์คนนั้นมาถอดรหัสพรีออเธอติเคเตอร์ดาตาที่ส่งมาจากไคลเอ็นต์ถ้าสามารถถอดรหัสได้ ก็จะเอาไทม์สแตมป์นั้นมาตรวจดู ถ้าถูกต้องตามเวลาก็แสดงว่า ยูสเซอร์ที่ติดต่อมานั้นเป็นยูสเซอร์ที่ถูกต้องจริงๆ

5. KDC จะสร้างครีเดนเทียลไว้ให้ Kerberosไคลเอ็นต์ที่ตัวเวิร์กสเตชันใช้เพื่อขอ Ticket – Granting Service โดยในขั้นแรก KDC จะสร้างเซสชันคีย์ที่ให้เมื่อไคลเอ็นต์ติดต่อกับ KDC แล้วเข้ารหัสเซสชันคีย์นี้ด้วยลองเทอมคีย์ของยูสเซอร์ ต่อมา KDC จะเอาเซสชันคีย์ที่เหมือนกันนี้อีกชุดหนึ่ง ใส่ใน TGT พร้อมกับข้อมูลอื่นๆ ที่เกี่ยวกับยูสเซอร์ แล้วเข้ารหัส TGT ด้วยลองเทอมคีย์ของ KDC เอง
6. KDC จะส่ง Kerberos Authentication Service Reply (KRB_AS_REP) ซึ่งประกอบด้วยเซสชันคีย์ที่ถูกเข้ารหัส และ TGT กลับไปให้ไคลเอ็นต์
7. เมื่อไคลเอ็นต์ได้รับ KRB_AS_REP มันจะใช้ลองเทอมคีย์ของยูสเซอร์ ถอดรหัสเซสชันคีย์และเก็บเซสชันคีย์นี้ไว้ในครีเดนเทียลแคชและจากนั้นก็เอา TGT เก็บไว้ในครีเดนเทียลแคชเช่นกัน



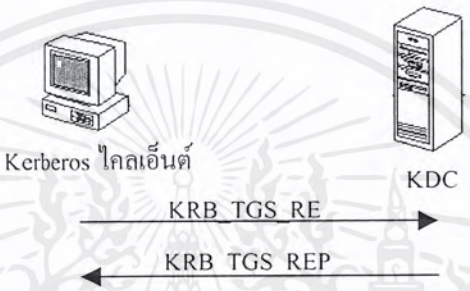
รูปที่ 4-6 แสดงขั้นตอน AS Exchange

4.2.2.2 TGT Exchange เป็นโพรโตคอลย่อยที่ทำหน้าที่ในช่วงที่ยูสเซอร์ ขอเซสชันทิคเกต (เพื่อใช้ขอบริการจากเซิร์ฟเวอร์) จาก KDC มีขั้นตอนการทำงานดังนี้

1. Kerberosไคลเอ็นต์ที่เวิร์กสเตชันของยูสเซอร์ ขอเซสชันทิคเกตที่จะใช้ขอบริการจากเซิร์ฟเวอร์ จาก KDC โดยจะส่ง Kerberos Ticket – Granting Service Request (KRB_TGS_REQ) ไปให้ KDC ซึ่งในข้อความนี้จะประกอบด้วย ชื่อ ยูสเซอร์ ออเธนติเคเตอร์ซึ่งถูกเข้ารหัสด้วยเซสชันคีย์ของยูสเซอร์ (TGT ที่ได้มาตอนโพรโตคอล AS Exchange) และชื่อของ Service ที่ยูสเซอร์ต้องการขอบริการ
2. เมื่อ KDC ได้รับ KRB_TGS_REQ มันจะถอดรหัส TGT ด้วยลองเทอมคีย์ของ KDC เอง แล้วเอาเซสชันคีย์ออกมา (จากโพรโตคอล AS Exchange : TGT จะถูกเข้ารหัสด้วยลองเทอมคีย์ของ KDC และใน TGT จะมีเซสชันคีย์อยู่) จากนั้นก็จะเอาเซสชันคีย์นี้ไปถอดรหัสออเธนติเคเตอร์แล้วก็ตรวจสอบ
3. ถ้าออเธนติเคเตอร์นั้นถูกต้อง KDC ก็จะสร้างเซสชันคีย์ที่ให้ไคลเอ็นต์ใช้ติดต่อกับเซิร์ฟเวอร์ ซึ่ง KDC จะเอาเซสชันคีย์นี้เข้ารหัสด้วยเซสชันคีย์ของยูสเซอร์ แล้วเอาเซสชันคีย์ที่ สร้างขึ้น

ใหม่นี้อีกชุดหนึ่ง ใส่ลงในเซสชันทิคเกตพร้อมกับข้อมูลของยูสเซอร์แล้วเข้ารหัสด้วยลองเทอมคีย์ของเซิร์ฟเวอร์

4. KDC จะส่ง Kerberos Ticket – Granting Service Request (KRB_TGS_REQ) ซึ่งประกอบด้วย Session ที่ยูสเซอร์จะใช้ติดต่อกับ Service ซึ่งถูกเข้ารหัสด้วยเซสชันคีย์ของยูสเซอร์และเซสชันทิคเกตไปให้แก่ไคลเอ็นต์
5. เมื่อไคลเอ็นต์ได้รับ KRB_TGS_REPไคลเอ็นต์ก็จะใช้เซสชันคีย์ของยูสเซอร์ ถอดรหัสเอาเซสชันคีย์ที่จะใช้กับ Service ออกมาเก็บไว้ในครีเดนเทียลแคชจากนั้นก็จะเอาเซสชันทิคเกตออกมาเก็บไว้ใน Cache ด้วยเช่นกัน

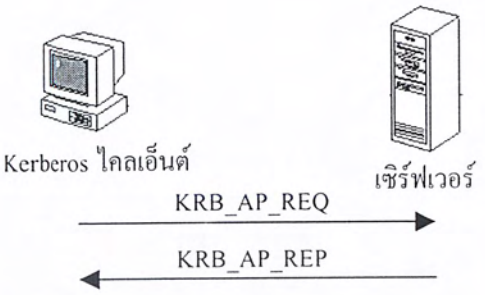


รูปที่ 4-7 แสดงขั้นตอน TGS Exchange

4.2.2.3 CS Exchange เป็นโพรโตคอลย่อยที่ทำหน้าที่ในช่วงที่ไคลเอ็นต์แสดงเซสชันทิคเกตเพื่อขอบริการจาก Service โดยมีขั้นตอนการทำงานดังนี้

1. Kerberosไคลเอ็นต์ที่อยู่บนเวิร์กสเตชันของยูสเซอร์ ส่ง Kerberos Application Request (KRB_AP_REQ) ไปให้เซิร์ฟเวอร์ โดยในข้อความนี้จะประกอบด้วยออเธนติเคเตอร์ที่ถูกเข้ารหัสด้วยเซสชันคีย์ที่ไคลเอ็นต์ใช้ติดต่อกับเซิร์ฟเวอร์ (Session Ticket นี้ได้ตอน TGS Exchange) และแฟล็ก (Flag) ที่เป็นตัวบอกว่าไคลเอ็นต์ต้องการการพิสูจน์ตนจากเซิร์ฟเวอร์ด้วยเช่นกัน
2. เมื่อเซิร์ฟเวอร์ได้รับ KRB_AP_REQ ก็จะถอดรหัสทิคเกตที่ได้มานั้น แล้วดึงเอาข้อมูลของยูสเซอร์และเซสชันคีย์ออกมา จากนั้นก็ใช้เซสชันคีย์นี้ถอดรหัสออเธนติเคเตอร์ของยูสเซอร์และตรวจสอบไทม์สแตมป์ข้างใน
3. ถ้าตรวจสอบออเธนติเคเตอร์ผ่านแล้วเซิร์ฟเวอร์ก็จะดูแฟล็กที่แสดงการพิสูจน์ตนว่า ต้องการการพิสูจน์ตนจากเซิร์ฟเวอร์เช่นกันหรือไม่ ถ้าต้องการ (แฟล็ค มีค่าเป็น 1) เซิร์ฟเวอร์ก็จะใช้เซสชันคีย์เข้ารหัสไทม์สแตมป์ที่ได้จากออเธนติเคเตอร์ของยูสเซอร์
4. เซิร์ฟเวอร์จะส่ง Kerberos Application Reply (KRB_AP_REP) กลับไปให้ยูสเซอร์ โดยจะมีออเธนติเคเตอร์ของเซิร์ฟเวอร์อยู่ด้วย
5. เมื่อไคลเอ็นต์ได้รับ KRB_AP_REPไคลเอ็นต์ก็จะใช้เซสชันคีย์ที่ใช้ในการติดต่อกับเซิร์ฟเวอร์ถอดรหัสออเธนติเคเตอร์ของเซิร์ฟเวอร์แล้วตรวจสอบไทม์สแตมป์ว่าเหมือนกับ

ตอนที่ส่งไป หรือไม่ ถ้าเวลาตรงกันไคลเอ็นต์ก็จะรู้ว่าเซิร์ฟเวอร์ได้รับออเธนติเคเตอร์แล้ว จากนั้นก็จะเริ่มการติดต่อกัน



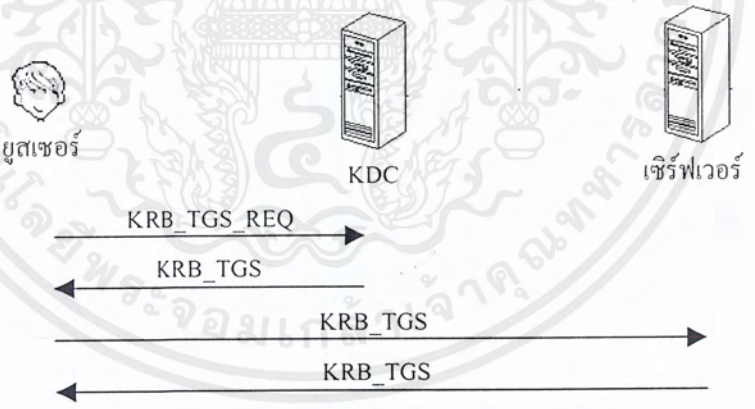
รูปที่ 4-8 แสดงขั้นตอน CS Exchange

4.2.3 สรุปขั้นตอนในการติดต่อขอบริการของยูสเซอร์

การติดต่อแบ่งได้เป็น 2 แบบคือ

1. ยูสเซอร์มี TGT อยู่แล้ว
2. ยูสเซอร์ยังไม่มี TGT

4.2.3.1 ยูสเซอร์มี TGT อยู่แล้ว



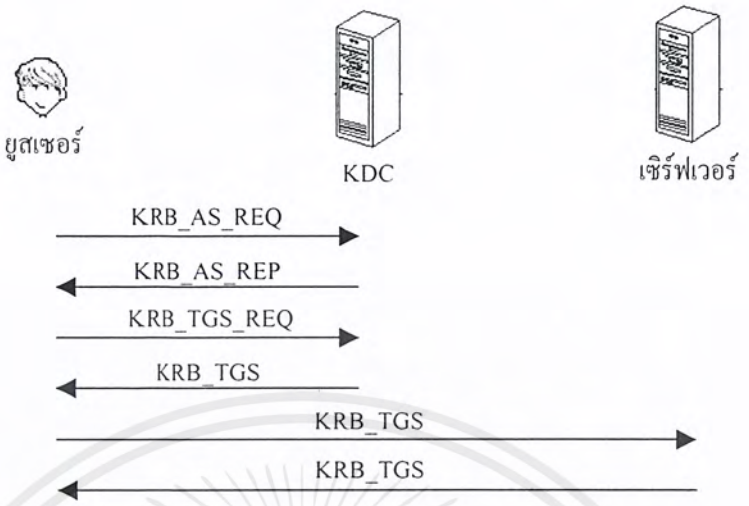
รูปที่ 4-9 ขั้นตอนการพิสูจน์ตนในขณะที่มี TGT แล้ว

1. ยูสเซอร์ส่ง KRB_TGS_REQ ไปให้ KDC เพื่อขอเซสชันทิคเกตจาก KDC
2. KDC ส่ง KRB_TGS_REP กลับมาให้ไคลเอ็นต์
3. ไคลเอ็นต์ส่ง KRB_AP_REQ ไปให้เซิร์ฟเวอร์เพื่อขอบริการจากเซิร์ฟเวอร์
4. เซิร์ฟเวอร์ส่ง KRB_AP_REP กลับมายังไคลเอ็นต์

1 – 2 เป็น TGS Exchange

3 – 4 เป็น CS Exchange

4.2.3.2 ยูสเซอร์ยังไม่มี TGT



รูปที่ 4-10 แสดงขั้นตอนการพิสูจน์ตนในขณะที่ยังไม่มี TGT

1. ยูสเซอร์ส่ง KRB_AS_REQ ไปให้ KDC เพื่อ TGT
2. KDC ส่ง KRB_AS_REP กลับมาให้ไคลเอ็นต์
3. ยูสเซอร์ส่ง KRB_TGS_REQ เพื่อขอเซสชันทิคเกตจาก KDC
4. KDC ส่ง KRB_TGS_REP กลับมาให้ไคลเอ็นต์
5. ยูสเซอร์ส่ง KRB_AP_REQ เพื่อขอบริการจากเซิร์ฟเวอร์
6. เซิร์ฟเวอร์ส่ง KRB_AP_REP กลับมาให้ไคลเอ็นต์

1 – 2 เป็น AS Exchange

3 – 4 เป็น TGS Exchange

5 – 6 เป็น CS Exchange

4.2.4 การพิสูจน์ตนข้ามโดเมน

หน้าที่ของ KDC สามารถแบ่งได้เป็น 2 หน้าที่ คือ การออก TGT และการออกเซสชันทิคเกตการที่มีหน้าที่ 2 อย่างนี้ทำให้โพรโตคอล Kerberos ต้องมีการทำงานข้ามขอบเขตของโดเมนโดยไคลเอ็นต์สามารถที่จะขอ TGT จาก KDC ในโดเมนหนึ่ง และใช้ TGT นี้ไปขอเซสชันทิคเกตจาก KDC ของโดเมนอื่นได้

เพื่อที่จะเห็นว่าการพิสูจน์ตนข้ามโดเมนทำงานอย่างไร ตอนแรกเราต้องพิจารณากรณีที่ง่ายที่สุดก่อนคือ ภายในระบบเครือข่ายนั้นมี 2โดเมน (East และ West) เราสามารถทำการพิสูจน์ตนข้ามโดเมนได้โดยการใช้ Inter – Domain Key ที่ KDC ใน 2โดเมนรู้เหมือนกัน ด้วยวิธีการนี้ สมมติว่า ยูสเซอร์ที่อยู่ในโดเมน East ต้องการที่จะขอบริการจากเซิร์ฟเวอร์ ซึ่งอยู่ในโดเมน West Kerberosไคลเอ็นต์ที่อยู่บนเวิร์กสเตชันของยูสเซอร์ จะร้องขอเซสชันทิคเกตจาก KDC ของโดเมน East KDC ในโดเมน East พบว่า

การร้องขอนั้นไม่ได้ขอบริการที่อยู่ในโดเมนของตน มันก็จะตอบไคลเอ็นต์โดยการส่งรีเฟอร์รอลทิคเกต (Referral Ticket) ไปให้รีเฟอร์รอลทิคเกตนี้ก็ คือ TGT ธรรมดาที่ถูกเข้ารหัสด้วย Inter – Domain Key ซึ่ง KDC ของโดเมน East และ KDC ของโดเมน West แชร์กันไคลเอ็นต์ก็จะใช้รีเฟอร์รอลทิคเกตนี้เพื่อขอตั๋วจาก KDC ของโดเมน West เมื่อ KDC ของโดเมน West ได้รับการร้องขอ มันก็จะใช้ Inter – Domain Key ถอดรหัสรีเฟอร์รอลทิคเกต ถ้าถอดรหัสได้สำเร็จมันก็จะส่งเซสชันทิคเกตที่ใช้ในการขอบริการเซิร์ฟเวอร์กลับไปให้ไคลเอ็นต์

การใช้วิธีนี้จะมีความซับซ้อนมากบนเครือข่ายที่มีมากกว่า 2 โดเมนในทางทฤษฎี KDC ในแต่ละโดเมนจะต้องมีการเชื่อมต่อโดยตรงกับ KDC ในโดเมนอื่นทุกๆ โดเมนบนระบบเครือข่าย และก็ Share Inter – Domain Key กัน โดยที่ Inter – Domain Key จะต้องไม่เหมือนกันด้วย แต่ในทางปฏิบัติ วิธีนี้ไม่สามารถ ทำได้ โพรโตคอล Kerberos จึงแก้ปัญหานี้โดยไม่ใช้การเชื่อมต่อถึงกันโดยตรงระหว่าง KDC แต่ไคลเอ็นต์ในโดเมนหนึ่ง สามารถที่จะขอตั๋วเพื่อขอบริการในโดเมนอื่นๆ โดยการขอผ่านโดเมนที่เป็นตัวกลาง (Intermediate Domain) ไปจนกว่าจะถึงโดเมนที่บริการนั้นอยู่

ยกตัวอย่างเช่น สมมติว่ามี 3โดเมนคือ East West และ CorpHQ.KDC ใน East ไม่ได้ Share Inter – Domain Key ไว้กับ KDC ใน West และ KDC ใน East และ West Share Domain Key กับ CorpHQ. ถ้ายูสเซอร์ใน Eastโดเมนต้องการเข้าถึงเซิร์ฟเวอร์ ใน Westโดเมนการขอจะเริ่มต้นที่ Eastโดเมนผ่านไปยังโดเมนที่เป็นตัวกลาง คือ CorpHQ.โดเมนจากนั้นจึงค่อยผ่านไปยัง Westโดเมนขั้นตอนที่เกิดขึ้นจะมีดังนี้

1. ไคลเอ็นต์ขอทิคเกต เพื่อขอใช้บริการในโดเมน West จาก KDC ในโดเมน East KDC ในโดเมน East ก็จะส่งรีเฟอร์รอลทิคเกตให้ไคลเอ็นต์เพื่อให้ไคลเอ็นต์ส่งรีเฟอร์รอลทิคเกตนี้ให้กับ CorpHQ. รีเฟอร์รอลทิคเกตนี้ ถูกเข้ารหัสด้วย Inter – Domain Key ที่ Share ไว้กันระหว่างโดเมนEast และ CorpHQ. ไปให้กับไคลเอ็นต์
2. ไคลเอ็นต์ขอเซสชันทิคเกตเพื่อขอใช้บริการในโดเมน West จาก KDC ในโดเมน CorpHQ. KDC ในโดเมน CorpHQ. จะส่งรีเฟอร์รอลทิคเกตที่ใช้ติดต่อกับ KDC ใน West ให้กับไคลเอ็นต์ โดยที่ทิคเกตนี้จะถูกเข้ารหัสด้วย Inter – Domain Key ที่ CorpHQ. Domain Share ไว้กับโดเมน West
3. ไคลเอ็นต์ขอเซสชันทิคเกตจาก KDC ในโดเมน West KDC ในโดเมน West ก็จะส่งเซสชันทิคเกตให้กับไคลเอ็นต์เพื่อใช้ติดต่อกับเซิร์ฟเวอร์ต่อไป

4.2.5 ตั๋ว (Ticket)

ชื่อฟิลด์	รายละเอียด
tkt-vno	บอก Version ของรูปแบบของตั๋ว (ใน Kerberos เป็น V.5)
realm	ชื่อโดเมนที่ออกตั๋ว
Iname	ชื่อเซิร์ฟเวอร์
3 ฟิลด์แรกนี้จะเป็นข้อมูลธรรมดาที่ไม่ถูกเข้ารหัส	
ฟิลด์ส่วนที่เหลือต่อไปนี้จะถูกเข้ารหัสด้วยคียของเซิร์ฟเวอร์	
flag	บอกออปชัน (Option) ของตั๋ว
key	เซสชันคีย์
Crealm	ชื่อโดเมนของไคลเอ็นต์
Cname	ชื่อไคลเอ็นต์
transite	ชื่อโดเมนที่ผ่านมา ในการพิสูจน์ตนข้ามโดเมนจน ถึงโดเมนที่เซิร์ฟเวอร์อยู่
Authtime	เวลาที่ไคลเอ็นต์ทำการพิสูจน์ตน
Starttime	เวลาที่ทิคเกตนี้เริ่มใช้งานได้
Endtimer	เวลาหมดอายุของตั๋ว
renew-till	เวลามากที่สุดที่จะรีนิวเอเบิล (Renewable) ตั๋ว

ตารางที่ 4-2 แสดงฟิลด์ต่างๆ ภายในตั๋ว

4.2.6 KDC กำหนดเวลาของตั๋วได้อย่างไร

KDC จะคิดเอนด์ไทม์ (Endtime) โดยการเอาช่วงเวลาที่อนุญาติให้ไคลเอ็นต์ใช้งานได้ (กำหนดโดย Kerberos) มาบวกกับเวลาในฟิลด์ Starttime แล้วเอาเวลาที่ได้นี้ไปเปรียบเทียบกับเวลาเลิกใช้งานที่ยูสเซอร์บอกมา ถ้าเวลาไหนมีค่าน้อยกว่ากันก็จะใช้เวลานั้นเป็นเอนด์ไทม์

4.2.7 อะไรจะเกิดขึ้นเมื่อตั๋วหมดอายุ

ถ้าไคลเอ็นต์ส่งตั๋วที่หมดอายุไปแล้วไปขอบริการจากเซิร์ฟเวอร์ เซิร์ฟเวอร์จะส่ง Error Message กลับมาบอกไคลเอ็นต์ดังนั้นไคลเอ็นต์จะต้องขอทิคเกตใหม่จาก KDC ถ้ายูสเซอร์ติดต่อกับเซิร์ฟเวอร์ได้ก่อนที่ตั๋วจะหมดอายุ เมื่อตั๋วหมดอายุ การติดต่อนั้นก็จะยังใช้งานได้อยู่ จะไม่มีส่วนเกี่ยวข้องอะไรด้วย

4.2.8 Renewable TGTS

การป้องกันการแอบดูเซสชันคีย์สามารถทำได้โดยกำหนดให้ อายุของตั๋วสั้นๆ เพื่อที่คีย์จะได้เปลี่ยนบ่อยๆ หรืออีกทางหนึ่งก็คือ ใช้การรีนิวเอเบิลด้วยวิธีนี้เซสชันคีย์จะเปลี่ยนทุกช่วงเวลาที่กำหนดไว้

โดยที่ไม่ต้องออกตั๋วใบใหม่ ดังนั้นการใช้วิธีนี้จะมีเวลาหมดอายุอยู่ 2 เวลา คือ เวลาที่เป็นเวลาหมดอายุของตั๋วที่ใช้เซสชันคีย์ปัจจุบันอยู่ (เวลาในการเปลี่ยนคีย์) และเวลาที่เป็นเวลาที่หมดอายุของตั๋วใบนั้นเลย

4.2.9 ขั้นตอนในการใช้ Kerberos ของยูสเซอร์

ขั้นแรกต้องสร้าง Kerberos พรินซิพอลก่อน ซึ่งมีลักษณะเหมือนกับแอ็คเคาต์บนเครื่อง โดยปกติชื่อของยูสเซอร์จะเป็น your_name@YOUR.REALM ส่วนที่อยู่หน้า @ เป็นส่วนที่ยูสเซอร์เลือกเอง ซึ่งโดยมากจะเหมือนกับชื่อแอ็คเคาต์ส่วนที่อยู่หลัง @ เป็นชื่อของ realm

พรินซิพอลจะประกอบด้วย ชื่อ พาสเวิร์ด และข้อมูลอื่นๆ ข้อมูลเหล่านี้จะถูกเก็บในฐานข้อมูลของ Kerberos ซึ่งถูกเข้ารหัสด้วย Kerberos มาสเตอร์คีย์ดังนั้นคนอื่นจึงไม่สามารถตรวจสอบได้

Kerberos สำหรับผู้ใช้แบบปกติจะไม่ค่อยยุ่งยากนัก โดยจะมีเซิร์ฟเวอร์เพียงจำนวนหนึ่งเท่านั้นที่จะถูกรักษาความปลอดภัยโดย Kerberos เช่น rlogin การที่จะใช้เซิร์ฟเวอร์นี้ คุณจำเป็นที่จะต้องได้รับ TGT ก่อน โดยใช้คำสั่ง kinit

```
%kinit
Password for your_name@YOUR.REALM:
```

เมื่อคุณใส่พาสเวิร์ดของคุณ โปรแกรม kinit จะติดต่อเซิร์ฟเวอร์ที่ทำหน้าพิสูจน์ตนเพื่อขอ TGT โดย พาสเวิร์ดของคุณจะถูกใช้ในการหาคีย์ซึ่งมันจะใช้ในการถอดรหัสของข้อมูลที่ถูกตอบสนองมาจากเซิร์ฟเวอร์ที่ทำหน้าที่พิสูจน์ตน (ส่วนนี้คือ ส่วนบรรจุการตอบรับของการร้องขอ เช่น เซสชันคีย์) ถ้าใส่พาสเวิร์ดถูกต้อง จะได้รับ TGT ซึ่งสามารถที่จะตรวจสอบโดยใช้คำสั่ง klist

```
%klist
Ticket cache : /var/tmp/krb5cc_1234
Default principal : your_name@YOUR.REALM

Valid starting          Expires              Service pr.
24 – Jul – 95 12:58:02  24 – Jul – 95 20:58:15  krbtgt/YOU
```

ทิคเกตแคชจะบอกตำแหน่งที่เก็บ Credential Cache Default Principal คือพรินซิพอลของ TGT Output ที่ได้รับจากตั๋วในขณะนี้จะยังคงค่าต่อไปเรื่อยๆ ยกเว้นส่วนของ Service Principal (krbtgt, etc) ซึ่งแสดงตั๋ว (TGT) จะสามารถใช้ได้ในช่วงเวลาสั้นๆ ในกรณีนี้คือ 8 ชั่วโมง (แต่อาจมากกว่าหรือน้อยกว่าก็ได้)

ถ้าขณะนี้ใช้โปรแกรม rlogin ซึ่งเป็น Version ที่มี Kerberos ควบคุมอยู่ โปรแกรมจะทำการใช้ TGT ใน Credentials Cache ในการขอตั๋วเพื่อที่จะทำการ Run โปรแกรม rlogin บนเครื่องซึ่งกำลังใช้งานอยู่ ซึ่งส่วนนี้มันจะทำแบบอัตโนมัติ และจะได้เห็นสิ่งต่อไปนี้

```
%rlogin newhost.domain
Last login : Fri Jul 21 12:04:40 from etc etc
```

อีกทางหนึ่ง ถ้าต้องการข้อมูลที่ต่างออกไปโดยสามารถเอามาจาก Cache คือ

```
%klist
Ticket cache : /var/tmp/krb5cc_1234
Default principal : your_name@YOUR.REALM

Valid starting          Expire              Service pr
24 – Jul – 95 12:58:02  24 – Jul – 95 20:58:15  krbtgt/YOU
24 – Jul – 95 12:58:02  24 – Jul – 95 20:58:15  host/newhost
```

เมื่อเสร็จการทำงานแล้ว rlogin จะยังคงปล่อยตั๋วที่คุณได้รับทิ้งเอาไว้ในแคชสิ่งนี้ไม่มีปัญหาต่อการรักษาความปลอดภัย นอกจากว่ามีบางคนสามารถยึดเครื่องเทอร์มินัลซึ่งกำลังล็อกอินอยู่ไปใช้แต่มันก็เป็นไปได้ยาก

เราสามารถลบครีเดนเทียลออกจากแคชได้โดยใช้คำสั่ง kdestroy

```
%kdestroy
%klist
klist : No credentials cache file found while setting
(ticket cache /var/tmp/krb5cc_1234)
```

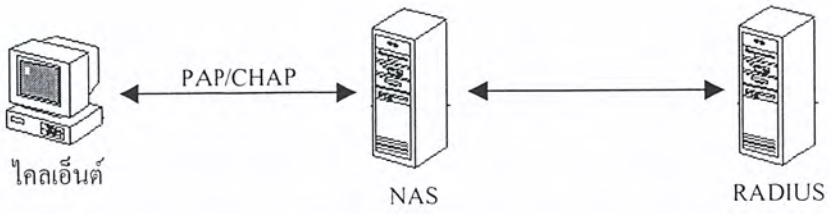
kdestroy จะทำการกำจัดตั๋วทั้งหมด (รวมทั้ง TGT) ออกจากแคชซึ่งเราจะไม่สามารถที่จะใช้ Kerberized ได้อีก (โปรแกรม Kerberized คือ โปรแกรมที่ได้รับการรักษาความปลอดภัยจาก Kerberos)

4.3 RADIUS (Remote Authentication Dial-In User Service)

RADIUS เป็นระบบรักษาความปลอดภัยจากการติดต่อจากภายนอกโดยใช้โมเด็ม มีสถาปัตยกรรมแบบไคลเอ็นต์/เซิร์ฟเวอร์ ซึ่งจะคอยรักษาความปลอดภัยของระบบและการบริการของระบบจากผู้ที่ไม่ได้ทำการพิสูจน์ตนจากระบบ RADIUS ประกอบด้วย 2 ส่วน คือ เซิร์ฟเวอร์ที่ทำการพิสูจน์ตน และ ไคลเอ็นต์โพรโตคอล โดยเซิร์ฟเวอร์ที่ทำการพิสูจน์ตน เก็บข้อมูลที่เป็นความลับทั้งหมดไว้ที่เดียว และพิสูจน์ข้อมูลของผู้ที่ต้องการใช้บริการของระบบว่าถูกต้องหรือไม่

4.3.1 การทำงานของ RADIUS

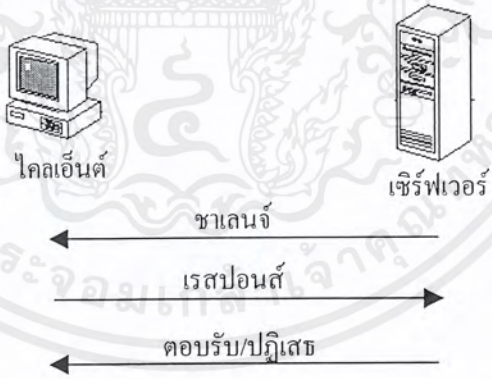
- การพิสูจน์ตนเองของ RADIUS จะผ่านระบบการสื่อสารมากมายระหว่างไคลเอ็นต์กับเซิร์ฟเวอร์ เมื่อผู้ใช้ระบบทำการพิสูจน์ตน ไคลเอ็นต์จะจัดหาทางที่ดีที่สุดในการติดต่อ นี่คือการพิสูจน์ตนโดยผ่าน NAS (Network Access Server) เซิร์ฟเวอร์ และ RADIUS
- ผู้ใช้ระบบจะทำการหมุนโมเด็มเพื่อติดต่อไปที่ NAS เซิร์ฟเวอร์ เมื่อติดต่อสำเร็จ NAS จะถามผู้ใช้ถึงยูสเซอร์เนมและรหัสผ่าน
- NAS จะสร้างแพ็กเก็ต (packet) ของข้อมูลจากข้อมูลต่างๆ ที่ได้ เรียกว่า authentication request แพ็กเก็ตจะมีข้อมูลที่บ่งบอกว่า NAS ใดเป็นผู้ส่ง authentication request NAS ซึ่งทำตัวเหมือน RADIUS ไคลเอ็นต์จะนำรหัสผ่านไปเข้ารหัสก่อนที่จะส่งให้ RADIUS เซิร์ฟเวอร์ เพื่อกันการลักลอบดักฟัง
- Authentication request จะถูกส่งผ่านระบบเครือข่ายจากไคลเอ็นต์ไปยังเซิร์ฟเวอร์ โดยการติดต่อผ่าน LAN หรือ WAN ซึ่งจะมีการจัดการเพื่อให้ RADIUS ไคลเอ็นต์สามารถหาเส้นทางเพื่อส่งคำร้องขอไปยังเซิร์ฟเวอร์อื่น
- เมื่อเซิร์ฟเวอร์ได้รับ authentication request เซิร์ฟเวอร์ที่ทำการพิสูจน์ตนจะตรวจสอบความถูกต้องของคำร้องขอ และทำการถอดรหัสข้อมูล เพื่อให้ได้ยูสเซอร์เนมและรหัสผ่าน ข้อมูลเหล่านี้จะถูกส่งไปยังระบบที่มีการรักษาความปลอดภัยที่ดี อาจจะเป็นไฟล์ passwd ในระบบปฏิบัติการยูนิกซ์ หรือ kerberos
- ถ้ายูสเซอร์เนมและรหัสผ่านที่ส่งมาถูกต้อง เซิร์ฟเวอร์จะส่ง authentication acknowledgment ซึ่งประกอบด้วย ข้อมูลของผู้ใช้ในระบบเครือข่ายและการบริการที่ต้องการ เช่น RADIUS เซิร์ฟเวอร์จะถาม NAS ว่าผู้ใช้ระบบต้องการ TCP/IP หรือต้องการ SLIP (Serial Line Internet Protocol) เพื่อติดต่อเข้าสู่ระบบเครือข่าย Authentication Acknowledgment สามารถมีข้อมูลที่จำกัดการเข้าถึงข้อมูลบางอย่างในระบบเครือข่าย
- ถ้ามีจุดใดในการล็อกอินเกิดความผิดพลาด RADIUS เซิร์ฟเวอร์จะส่ง authentication reject NAS และผู้ใช้ระบบก็จะถูกปฏิเสธในการเข้าถึงระบบ
- เพื่อให้แน่ใจว่าการตอบรับกลับมานั้นไม่ใช่ผู้เจาะระบบในระบบเครือข่าย RADIUS เซิร์ฟเวอร์จะส่ง authentication key หรือลายเซ็นต์เพื่อพิสูจน์ตนต่อ RADIUS ไคลเอ็นต์ เมื่อ NAS ได้รับข้อมูลเหล่านี้ NAS จะให้อำนาจต่างๆ ที่จำเป็นไปยังผู้ใช้ระบบ



รูปที่ 4-11 แสดงส่วนประกอบหลักของระบบพิสูจน์ตน RADIUS

4.3.2 การทำงานโดยใช้ Challenge/Response

การทำงานโดยใช้ชาเลนจ์/เรสปอนส์ (Challenge/Response) มีเพื่อป้องกันการลักลอบดักฟังข้อมูลจากผู้ไม่ประสงค์ดี รหัสผ่านของผู้ใช้ระบบถูกเข้ารหัสก่อนที่จะส่งไปที่เซิร์ฟเวอร์ โดยการเข้ารหัสนิยมใช้การเข้ารหัสทางเดียว เช่น MD5 หรือสูงกว่านั้น การทำงานเริ่มต้น โดยเซิร์ฟเวอร์ส่งชาเลนจ์ให้ไคลเอ็นต์ โดยชาเลนจ์เป็นเลขจำนวนหนึ่งซึ่งสุ่มขึ้นมา ไม่สามารถคาดเดาได้ เมื่อไคลเอ็นต์ได้รับชาเลนจ์จะนำชาเลนจ์ไปรวมกับรหัสผ่านแล้วนำไปเข้ารหัสแบบทางเดียว หลังจากนั้นจึงส่งผลลัพธ์กลับไปที่เซิร์ฟเวอร์ ซึ่งผลลัพธ์เรียกว่าเรสปอนส์เมื่อเซิร์ฟเวอร์ได้รับเรสปอนส์กลับมาจากไคลเอ็นต์ เซิร์ฟเวอร์จะนำรหัสผ่านของผู้ใช้ระบบที่เก็บในฐานข้อมูลของเซิร์ฟเวอร์มารวมกับชาเลนจ์ที่เหมือนกับที่ส่งไปให้ไคลเอ็นต์แล้วเข้ารหัส แล้วนำมาเปรียบเทียบกับข้อมูลที่ไคลเอ็นต์ส่งมา ถ้าเหมือนกัน แสดงว่าไคลเอ็นต์ได้ใส่รหัสผ่านถูกต้อง สามารถใช้ทรัพยากรของระบบได้



รูปที่ 4-12 แสดงลำดับการทำงานโดยใช้งานชาเลนจ์/เรสปอนส์

4.3.3 ตัวอย่างการทำงาน

NAS ส่ง Access-Request ไปยัง RADIUS เซิร์ฟเวอร์ ซึ่งประกอบด้วย NIS-Identifier NIS-Port ยูสเซอร์เนม และ รหัสผ่านของผู้ใช้ระบบ หลังจากนั้นเซิร์ฟเวอร์ส่ง Access-Challenge ซึ่งประกอบด้วยสเตท และข้อความตอบรับว่า "Challenge 12345678, enter your response at the prompt" ข้อความนี้จะขึ้นที่หน้าจอของ NAS เมื่อ NAS ได้รับการตอบรับจากไคลเอ็นต์ จะส่ง Access-Request แพ็กเก็ตใหม่ไปยังเซิร์ฟเวอร์ด้วย NAS-Identifier (ด้วย ID ใหม่) NAS-Port ยูสเซอร์เนม รหัสผ่านของผู้ใช้ระบบที่เข้ารหัส

และแอตทริบิวต์ของสเตทเดียวกันกับ Access-Challenge หลังจากนั้น เซิร์ฟเวอร์จะส่ง Access-Accept หรือ Access-Reject ขึ้นกับว่าเรสปอนส์นั้นตรงกับที่เซิร์ฟเวอร์คำนวนไว้หรือไม่ หรือเซิร์ฟเวอร์อาจส่ง Access-Challenge อื่นอีก เพื่อถามข้อมูลเพิ่มเติมของไคลเอ็นต์ได้

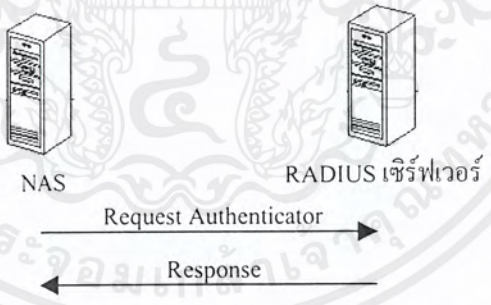
4.3.4 การใช้งาน RADIUS กับโพรโตคอล PAP และ CHAP

RADIUS สามารถใช้ได้กับโพรโตคอล Password Authentication Protocol (PAP) และ Challenge Handshake Authentication Protocol (CHAP) ซึ่งทั้ง 2 โพรโตคอลนี้ เป็นโพรโตคอลที่ใช้ในการติดต่อระหว่างไคลเอ็นต์และ NAS เซิร์ฟเวอร์

4.3.5 การให้เซิร์ฟเวอร์พิสูจน์ตนต่อไคลเอ็นต์

เพื่อเป็นการตรวจสอบว่าคำสั่งนั้นมาจากเซิร์ฟเวอร์จริงหรือไม่ มีขั้นตอนดังนี้

1. NAS ส่งข้อมูลไปยัง RADIUS เซิร์ฟเวอร์ โดยส่ง Request Authenticator ไปด้วย ซึ่งประกอบด้วยเลขสุ่ม
2. เมื่อ RADIUS เซิร์ฟเวอร์ส่งเรสปอนส์คือส่งคำสั่งไปยัง RADIUS ไคลเอ็นต์ เช่น Access Accept จะส่ง Response Authenticator ไปด้วย
3. เมื่อ RADIUS ไคเอ็นต์ได้รับ Response Authenticator ก็จะสร้างขึ้นมาอีก 1 ชุด เพื่อนำมาเปรียบเทียบ ถ้าไม่ถูกต้องจะไม่สนใจต่อคำสั่งนั้น



รูปที่ 4-13 แสดงการพิสูจน์ตนของเซิร์ฟเวอร์ต่อไคลเอ็นต์

4.3.6 ประโยชน์ของการรักษาความปลอดภัยของระบบโดยใช้ RADIUS

1. มีสถาปัตยกรรมที่ดี

สถาปัตยกรรมของ RADIUS เป็นแบบไคลเอ็นต์/เซิร์ฟเวอร์ โดยไคลเอ็นต์มีหน้าที่ร้องขอบริการจากผู้ใช้บริการไปยังเซิร์ฟเวอร์ และเซิร์ฟเวอร์มีหน้าที่ทำการพิสูจน์ตนว่าผู้ใช้ระบบนั้นเป็นผู้ที่ได้รับอนุญาตให้สามารถใช้ทรัพยากรของระบบได้หรือไม่ และเป็นผู้ใช้ระบบตัวจริงหรือไม่ และมีหน้าที่พิสูจน์สิทธิ์ว่าผู้ใช้ระบบนั้นได้รับอนุญาตให้เข้าถึงข้อมูลได้หรือไม่ ระดับใด และมีหน้าที่ให้บริการต่างๆ แก่ผู้ใช้ระบบที่ร้องขอบริการจากเซิร์ฟเวอร์

2. มีการรักษาความปลอดภัยที่ดีกว่า

   RADIUS สร้างฐานข้อมูลเก็บข้อมูลเกี่ยวกับผู้ใช้ระบบและบริการต่างๆ ที่มีในระบบ โดยระบบประกอบด้วยโมเด็มต่างๆ ที่ติดต่อเข้ามามากมายและรีโมตเซิร์ฟเวอร์มากกว่า 1 เซิร์ฟเวอร์ ฐานข้อมูลนี้มีการเก็บข้อมูลที่ต้องรักษาความปลอดภัยไว้ที่เดียว (central database) คือที่ RADIUS เซิร์ฟเวอร์ ซึ่งเป็นเซิร์ฟเวอร์ที่ทำกรพิสูจน์ตน แทนที่การเก็บข้อมูลแบบกระจายบนระบบเครือข่ายซึ่งเก็บบนอุปกรณ์ที่ต่างกัน การเก็บข้อมูลนี้จะเป็นการเพิ่มความปลอดภัยของระบบ เนื่องจากสามารถรักษาความปลอดภัยง่ายกว่าต้องรักษาความปลอดภัยหลายๆ เซิร์ฟเวอร์ โดยต้องสื่อสารผ่านระบบขนาดใหญ่และไม่มีความปลอดภัย

3. มีการเปิดโพรโตคอลให้ผู้อื่นรับรู้

   RADIUS มีการเปิดซอร์สโค้ดอย่างเปิดเผย และ RADIUS สามารถนำไปประยุกต์ใช้งานให้เข้ากับระบบและโพรโตคอลได้ ซึ่งช่วยประหยัดเวลาอย่างมาก โดยอนุญาตให้ผู้ใช้ระบบทำการแก้ไข RADIUS เซิร์ฟเวอร์ เพื่อให้เหมาะสมกับระบบเครือข่ายที่ใช้ แทนที่จะต้องเปลี่ยนแปลงระบบเครือข่ายทั้งระบบ

RADIUS สามารถเปลี่ยนแปลงให้สามารถใช้กับระบบรักษาความปลอดภัยแบบใดก็ได้ และสามารถทำงานได้กับอุปกรณ์การสื่อสารทุกอย่างที่สนับสนุน RADIUS โพรโตคอล

4. มีความยืดหยุ่นในการพิสูจน์ตน

   RADIUS เซิร์ฟเวอร์สามารถรับรองการพิสูจน์ตนได้หลายแบบ เมื่อผู้ใช้ระบบได้ใส่ยูสเซอร์เนม และรหัสผ่านของผู้ใช้ระบบต่อเซิร์ฟเวอร์ โดยสามารถใช้วิธีล็อกอินผ่านระบบปฏิบัติการยูนิกซ์ และมีโพรโตคอล PAP และ CHAP เป็นต้น

5. สามารถจัดการฐานข้อมูลได้ง่าย

   เนื่องจากระบบ RADIUS เก็บฐานข้อมูลที่ RADIUS เซิร์ฟเวอร์เพียงที่เดียวเท่านั้น ดังนั้นเมื่อมีการแก้ไขหรือต้องการเปลี่ยนแปลงฐานข้อมูลจึงสามารถทำได้ง่าย ไม่ต้องยุ่งยากเรื่องการปรับเปลี่ยนข้อมูลตามเซิร์ฟเวอร์ต่างๆ เพื่อให้เหมือนกับเซิร์ฟเวอร์ที่เก็บข้อมูลจริง

บทที่ 5
โพรโตคอลที่เกี่ยวข้องกับการพิสูจน์ตน

5.1 โพรโตคอล PAP และ CHAP ที่ใช้ในการติดต่อแบบ PPP

Point-to-Point Protocol (PPP) เป็นโพรโตคอลมาตรฐานในระดับเน็ตเวิร์คเลเยอร์ (Network Layer) โพรโตคอล PPP ได้กำหนดโพรโตคอลในการพิสูจน์ตน เพื่อให้เพียร์พิสูจน์ตนก่อนได้รับอนุญาตให้สามารถใช้เน็ตเวิร์คเลเยอร์เพื่อส่งข้อมูลผ่านระบบเครือข่าย

โพรโตคอลที่สำคัญในการพิสูจน์คือ โพรโตคอล (Password Authentication Protocol) PAP และ (Challenge Handshake Authentication Protocol) CHAP

5.1.1 ศัพท์เบื้องต้น

Remote Router เราเตอร์ที่ต้องการติดต่อ

Authenticator เราเตอร์ที่เริ่มการพิสูจน์ตนโดยส่งแพ็คเก็ต PAP หรือ CHAP Request ทั้งเราเตอร์ธรรมดาหรือรีโมตเราเตอร์สามารถเป็นออเธนติเคเตอร์ได้

Peer เราเตอร์ที่ตอบรับการ Authentication Request โดยการพิสูจน์ตนเองต่อเราเตอร์อื่น ทั้งเราเตอร์ธรรมดาหรือรีโมตเราเตอร์สามารถเป็นเพียร์ได้

5.1.2 การทำงานโดยใช้โพรโตคอล PAP

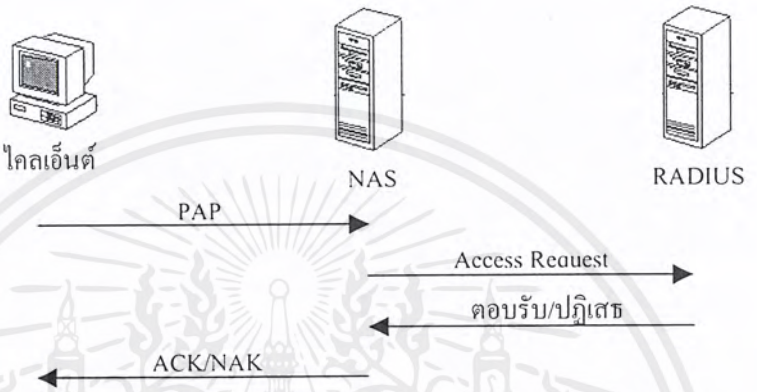
PAP เป็นโพรโตคอลที่ง่าย เพราะใช้ทูเวย์แฮนด์เชค (two-way handshake) และรหัสผ่านซึ่งเป็นตัวพิสูจน์เพียร์ต่อออเธนติเคเตอร์ ไม่ได้รับการเข้ารหัส การทำแฮนด์เชคจะทำเมื่อเริ่มการติดต่อเท่านั้น การทำงานของ PAP เป็นดังนี้

1. เมื่อเริ่มต้นการพิสูจน์ตน เพียร์จะส่งแพ็กเก็ต PAP ซึ่งประกอบด้วยยูสเซอร์เนมและรหัสผ่านถ้าเพียร์
   - ไม่ได้รับการตอบรับกลับมา มันจะรอชั่วระยะเวลาหนึ่ง และส่งแพ็คเก็ต PAP ใหม่อีกจนกระทั่งได้รับการตอบรับกลับมา ซึ่งอาจเป็น Positive (ACK) หรือเป็น Negative (ACK)
   - ได้รับการตอบรับแบบ ACK มันจะนำ Network Control Protocols (NCPs) มาใช้ เพื่อใช้โพรโตคอลอื่นต่อไป เช่น IPX หรือ IP
   - ได้รับการตอบรับแบบ NAK การพิสูจน์ตนล้มเหลว ตัวออเธนติเคเตอร์ตัดการติดต่อ
2. ตัวออเธนติเคเตอร์รอแพ็คเก็ต PAP ระยะเวลาหนึ่ง
   - ถ้าไม่ได้รับ มันจะรอแพ็คเก็ตในระยะเวลาที่กำหนด ถ้ายังไม่ได้รับแพ็คเก็ต PAP มันจะตัดการติดต่อ

- ถ้าได้รับ มันจะเปรียบเทียบรหัสผ่านในฐานข้อมูลกับรหัสผ่านที่ได้จากแพ็คเก็ต PAP ถ้าตรงกัน มันจะส่ง ACK กลับไปที่เพียร์ ถ้ารหัสผ่านไม่ตรงกัน มันจะส่ง NAK กลับไปที่เพียร์และตัดการติดต่อ

ข้อดีของการติดต่อแบบ PAP คือ สามารถใช้งานง่าย

ข้อเสียของการติดต่อแบบ PAP คือ รหัสผ่านไม่ได้รับการเข้ารหัสและส่งผ่านระบบเครือข่าย ถ้ามีผู้ดักจับข้อมูลสามารถนำรหัสผ่านนั้นไปใช้ได้



รูปที่ 5-1 แสดงตัวอย่างการทำงานโดยใช้โพรโตคอล PAP ของ RADIUS เซิร์ฟเวอร์

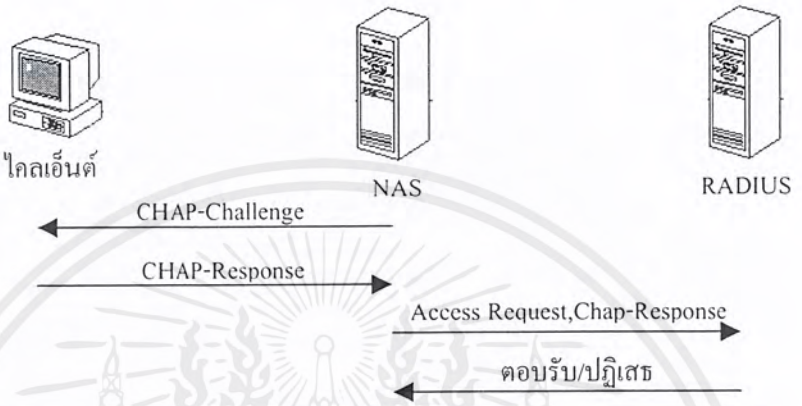
5.1.3 การทำงานโดยใช้โพรโตกอล CHAP

CHAP ใช้ทรีเวย์แฮนด์เชค (three-way handshake) ในการพิสูจน์ตนของเพียร์ CHAP จะมีการกระทำแบบนี้เมื่อเริ่มติดต่อ และจะสามารถทำแฮนด์เชคได้อีกหลังจากการติดต่อสำเร็จแล้ว การทำงานของ CHAP เป็นดังนี้

1. หลังจากการติดต่อสำเร็จออเธนติเคเตอร์ส่งข้อความชาเลนจ์ไปให้เพียร์และรอการตอบกลับ ถ้าออเธนติเคเตอร์ไม่ได้รับการตอบกลับในเวลาที่กำหนดออเธนติเคเตอร์จะส่งข้อความชาเลนจ์ไปอีกออเธนติเคเตอร์จะทำเช่นนี้ไปเรื่อยๆ จนกว่าจะถึงเวลาที่กำหนดไว้ ถ้าถึงเวลาที่กำหนดไว้ออเธนติเคเตอร์จะเลิกส่งและตัดการติดต่อ
2. เมื่อเพียร์ได้รับข้อความชาเลนจ์เพียร์จะนำข้อความเข้ารหัส ซึ่งในข้อความประกอบด้วยความลับที่รู้กันระหว่างเพียร์และออเธนติเคเตอร์ และส่งผลลัพธ์ที่ได้เข้ารหัสกลับไปให้ออเธนติเคเตอร์ในแพ็คเก็ตเรสปอนส์
3. ออเธนติเคเตอร์ใช้ความลับที่ได้รับมาตรวจสอบเรสปอนส์ว่าเป็นข้อความที่เข้ารหัสที่ได้ส่งไปหรือเปล่า ถ้าใช่ออเธนติเคเตอร์จะส่ง ACK และนำ NCPs มาใช้ แต่ถ้าไม่ใช่ออเธนติเคเตอร์จะส่ง NAK และตัดการติดต่อ
4. ถ้าเพียร์ไม่ได้รับข้อความตอบกลับที่แสดงว่าการพิสูจน์สำเร็จหรือไม่ มันจะรอจนถึงเวลาที่กำหนด เมื่อถึงเวลาที่กำหนด มันจะส่งเรสปอนส์ครั้งสุดท้ายที่ได้ส่งไป เมื่อเพียร์ได้รับ ACK มันจะนำ NCPs มาใช้ แต่ถ้ามันได้รับ NAK มันจะหยุดการติดต่อ

เพื่อป้องกันการดักจับข้อมูล การติดต่อแบบ CHAP จะไม่มีการส่งความลับผ่านระบบเครือข่าย การคำนวณเพื่อหาความลับนั้นจะเกิดที่ตัวเซิร์ฟเวอร์หลักเพียงที่เดียวเท่านั้นถ้ามีการส่งความลับผ่านระบบเครือข่ายจะต้องเข้ารหัสทุกครั้ง

ข้อดีของการติดต่อแบบ CHAP คือ รหัสผ่านได้รับการเข้ารหัสทุกครั้งก่อนส่งผ่านระบบเครือข่าย จึงสามารถแน่ใจได้ว่าผู้ดักจับข้อมูลไม่สามารถนำรหัสผ่านไปใช้ได้



รูปที่ 5-2 แสดงตัวอย่างการทำงานโดยใช้โพรโตคอล CHAP ของ RADIUS เซิร์ฟเวอร์

5.2 โพรโตคอล Secure Socket Layer (SSL)

5.2.1 ความรู้เบื้องต้นเกี่ยวกับการเข้ารหัสคีย์

การเข้ารหัสแบบพับลิกคีย์ (Public Key) เป็นวิธีการที่ใช้คีย์ 1 คู่ซึ่งไม่สมมาตรกันในการเข้ารหัสและถอดรหัส แต่ละคู่ประกอบด้วยพับลิกคีย์และไพรเวตคีย์ (Private Key) พับลิกคีย์เป็นคีย์ที่สร้างขึ้นมาและแจกจ่ายออกไป ส่วนไพรเวตคีย์ จะเก็บเป็นความลับ ไม่แจกจ่ายออกไป

ข้อมูลที่ถูกเข้ารหัสด้วยพับลิกคีย์ สามารถถอดรหัสได้โดยใช้ไพรเวตคีย์เท่านั้น ในทางกลับกันข้อมูลที่ถูกเข้ารหัสด้วยไพรเวตคีย์ จะสามารถถอดรหัสได้โดยใช้พับลิกคีย์เท่านั้น ความไม่สมมาตรแบบนี้เป็นคุณสมบัติที่ทำให้การเข้ารหัสแบบพับลิค คีย์สามารถนำไปใช้งานได้ดี

5.2.2 การนำการเข้ารหัสแบบพับลิกคีย์ใช้ในการพิสูจน์ตน

จะใช้การยกตัวอย่างการติดต่อระหว่าง A กับ B โดยใช้การเข้ารหัสแบบพับลิกคีย์ ซึ่ง {something}key หมายความว่า something ได้ถูกเข้ารหัสหรือถอดรหัสโดยใช้ Key

สมมติว่า A ต้องการให้ B ทำการพิสูจน์ตน B มีคีย์อยู่คู่หนึ่ง คือ พับลิกคีย์ และ ไพรเวตคีย์ B เปิดเผยพับลิกคีย์ต่อ A หลังจากนั้น A สร้างข้อความสุ่มและส่งไปให้ B

A -> B random-message

B จะใช้ไพรเวตคีย์เข้ารหัสข้อมูลที่ได้รับและส่งกลับไปให้ A

B -> A {random-message}B private key

เมื่อ A ได้รับข้อความจะถอดรหัสโดยใช้พับลิกคีย์ของ B ที่ได้รับมาก่อนหน้านั้น A จะเปรียบเทียบข้อความที่ได้ถอดรหัสแล้วกับข้อความที่ได้ส่งไปให้ B ถ้าตรงกัน แสดงว่าเข้าได้ติดต่อกับ B อยู่ ผู้ดักจับข้อมูลไม่สามารถรู้ไพรเวตคีย์ของ B ได้ ดังนั้นเขาจึงไม่สามารถเข้ารหัสข้อความที่ A ส่งไปหา B ได้

เว้นแต่ข้อความที่ส่งมาให้เข้ารหัสนั้นไม่ได้เป็นข้อความที่เกิดจากการสุ่มขึ้นมา ดังนั้นจึงไม่ควรเข้ารหัสอะไรบางอย่างด้วย ไพรเวตคีย์และส่งไปให้ผู้อื่น ดังนั้นแทนที่ B จะเข้ารหัสข้อความที่ A ส่งมาให้นั้น B จะสร้างแมสเซจไดเจสต์ (message digest) และนำไปเข้ารหัสแมสเซจไดเจสต์ได้มาจากข้อความที่สุ่มขึ้นมาแมสเซจไดเจสต์ที่ดีควรที่คุณสมบัติดังนี้

- ไดเจสต์สามารถแปลงกลับได้ยาก ผู้ที่ดักจับข้อมูลของ B ไม่สามารถแปลงจากไดเจสต์ไปสู่ข้อความเดิมได้
- ผู้ที่ดักจับข้อมูลไม่สามารถหาข้อความที่ต่างออกไปแต่ว่าคำนวณแล้วได้ไดเจสต์เดียวกัน

B สามารถป้องกันตนเองโดยการใช้ไดเจสต์เมื่อ B คำนวณไดเจสต์จากข้อความสุ่มที่ได้รับจาก A ได้แล้ว จะนำไดเจสต์ไปเข้ารหัส และส่งกลับไปให้ A A สามารถคำนวณค่าไดเจสต์เดียวกันได้ และพิสูจน์ B โดยถอดรหัสข้อความที่ B ส่งมา แล้วนำไปเปรียบเทียบกับค่าที่คำนวณได้

วิธีการที่ได้อธิบายนี้ได้เรียกว่าดิจิตอลซิกเนเจอร์ (digital signature) B ได้แสดงเครื่องหมายจากข้อความที่ A ได้สร้างขึ้นมา หลังจากนั้น แทนที่ B จะส่งข้อความสุ่มที่เข้ารหัสกลับไปให้ A ซึ่งอันตรายนั้น จะใช้โพรโตคอลในการพิสูจน์ตนโดยข้อมูลบางส่วนหรือทั้งหมดถูกสร้างโดย B

A -> B hello, are you B?

B -> A A, this is B

{digest[A, this is B]}B private key

เมื่อใช้โพรโตคอลนี้ B ส่งข้อความที่ไม่ได้เข้ารหัสไปก่อน หลังจากนั้นจะส่งข้อความที่เข้ารหัสและใช้ไดเจสต์ให้ A

การใช้พับลิกคีย์

จากโพรโตคอลนี้

A -> B hello

B -> A hi, I'm B, B public key

A -> B prove it

B -> A A, this is B

{digest[A, this is B]}B private key

จากโพรโตคอลนี้ ใครก็สามารถเป็น B ได้ เพราะต้องการเพียงแต่ไพรเวตคีย์และพับลิกคีย์ที่เข้าคู่กัน ผู้อื่นที่ไม่ใช่ B สามารถส่งพับลิกคีย์ของเขาแทนพับลิกคีย์ของ B หลังจากนั้นส่งข้อความที่เข้ารหัสด้วยไพรเวตคีย์ของเขาเอง A ก็ไม่สามารถรู้ได้ว่าผู้ที่ติดต่ออยู่ไม่ใช่ B

เพื่อแก้ปัญหานี้ จึงมีมาตรฐานในการติดต่อ เรียกว่าเซอร์ติฟิเคต (certificate) โดยเซอร์ติฟิเคตเป็นตัวรับรองไพรเวตคีย์ที่ใช้ เมื่อใช้เซอร์ติฟิเคตทุกคนสามารถตรวจสอบเซอร์ติฟิเคตได้ว่าเป็นผู้ที่ต้องการติดต่อจริงหรือไม่ จึงได้โพรโตคอลใหม่ดังนี้

A -> B hello

B -> A hi, I'm B, B-certificate

A -> B prove it

B -> A A, this is B

{digest[A, this is B]}B private key

เมื่อ A ได้รับข้อความแรกจาก B A สามารถตรวจสอบเซอร์ติฟิเคตตรวจสอบซิกเนเจอร์ (โดยใช้ไดเจสต์และพับลิกคีย์) และตรวจสอบข้อมูลต่างๆ เช่น ชื่อของ B ว่าผู้ที่ติดต่อด้วยเป็น B จริงๆ ขณะนี้ A สามารถมั่นใจได้ว่าพับลิกคีย์ที่ได้รับเป็นพับลิกคีย์ของ B จริงๆ หลังจากนั้น การทำงานก็เหมือนเดิม ถ้ามีผู้ที่ต้องการปลอมตัวเป็น B จะเกิดปัญหา (ให้ M แทนผู้ที่ปลอมตัว)

A -> M hello

M -> A hi, I'm B, B-certificate

A -> M prove it

M -> A ????

ผู้ที่ต้องการปลอมตัวเป็น B ไม่สามารถส่งข้อความที่ A ต้องการได้ เนื่องจากผู้ที่ต้องการปลอมตัวเป็น B ไม่มีพับลิกคีย์ ของ B

การแลกเปลี่ยนความลับ

เมื่อ B พิสูจน์ต่อ A สำเร็จแล้ว A จะมีการกระทำอื่นๆ เช่น ส่งข้อความไปหา B ซึ่งมี B เท่านั้นที่สามารถถอดรหัสได้

A -> B {secret}B public key

หนทางเดียวที่สามารถดูความลับได้คือ ต้องถอดรหัสข้อความด้วยไพรเวตคีย์ของ B การกระทำเช่นนี้มีความปลอดภัยมาก ถ้ามีผู้ดักจับข้อมูลในขณะที่ A และ B ติดต่อกัน ผู้ดักจับข้อมูลไม่สามารถอ่านความลับที่ A และ B ส่งถึงกันได้

แต่ผู้ดักจับข้อมูลยังมีวิธีการอื่นอีก แม้ว่าผู้ดักจับข้อมูลไม่สามารถอ่านความลับที่ A และ B ส่งถึงกันได้ แต่เขาสามารถข้อความนั้นได้ ยกตัวอย่างเช่น

A -> M hello

M -> B hello

B -> M hi, I'm B, B-certificate

M -> A hi, I'm B, B-certificate

A -> M prove it

M -> B prove it

B -> M A, This is B

{digest[A, This is B]}B private key

M -> A A, This is B

{digest[A, This is B]}B private key

A -> M ok B, here a secret {secret}B public key

M -> B ok B, here a secret {secret}B public key

B -> M {some message} secret key

M -> A Garble[{some message} secret key]

ผู้ดักจับข้อมูลปล่อยให้ข้อมูลผ่านไปโดยไม่มีการเปลี่ยนแปลง จนกระทั่ง A และ B แชร์ความลับ ผู้ดักจับข้อมูลจะทำลายข้อความที่ B ส่งไปหา A ในขณะนี้ A ไว้วางใจว่าผู้ที่ติดต่อเป็น B ดังนั้น A จึงเชื่อว่าข้อมูลที่ถูกทำลายนั้นเป็นข้อมูลจาก B จริงๆ ในความเป็นจริง ผู้ดักจับข้อมูลไม่รู้ความลับระหว่าง A และ B ที่เขาสามารถทำได้คือทำลายข้อมูลที่ถูกเข้ารหัสแล้วเท่านั้น ดังนั้น การติดต่อระหว่าง A กับ B จึงถูกบิดเบือนข้อมูล ทำให้ไม่สามารถติดต่อกันได้

เพื่อป้องกันเหตุการณ์แบบนี้ A และ B จึงใช้ Message Authentication Code (MAC) ในโพรโตคอล MAC เป็นข้อมูลที่มาจากการคำนวณ โดยใช้ความลับและบางส่วนของข้อมูลที่ส่ง อัลกอริทึมแบบไดเจสต์เป็นวิธีที่ดีในการสร้างฟังก์ชั่น MAC

MAC := Digest[some message, secret]

เนื่องจากผู้ดักจับข้อมูลไม่รู้ความลับจึงไม่สามารถคำนวนค่าไดเจสต์ที่ถูกต้อง ถ้าผู้ดักจับข้อมูลบิดเบือนข้อมูลที่ส่งระหว่าง A กับ B โอกาสที่จะประสบความสำเร็จจะยิ่งน้อยลงถ้าขนาดของไดเจสต์ใหญ่มากขึ้น แต่ในความเป็นจริงแล้วไม่สามารถเป็นไปได้เลย ดังนั้นโพรโตคอลใหม่จึงเป็นดังนี้

A -> B hello

B -> A Hi, I'm B, B-certificate

A -> B prove it

B -> A A, This is B

{digest[A, This is B]}B private key

A -> B ok B, here a secret {secret}B public key

B -> A {some message, MAC}secret key

ขณะนี้ผู้ดักจับข้อมูลไม่สามารถทำลายข้อมูลที่ส่งระหว่าง A กับ B ได้แล้ว เนื่องจากถ้า A หรือ B ถอดรหัสโดยใช้คีย์ความลับแล้วในข้อความที่ส่งมานั้น ไม่มี MAC A และ B จะรู้ว่าข้อความที่ส่งถึงกันได้ถูกบิดเบือน A และ B จะหยุดส่งข้อความถึงกัน ผู้ดักจับข้อมูลก็จะไม่สามารถบิดเบือนข้อมูลได้อีกต่อไป

5.3 LDAP (Lightweight Directory Access Protocol)

5.3.1 ไดเรกทอรีเซอร์วิซ

ไดเรกทอรีเซอร์วิซนั้นคล้ายกับฐานข้อมูล แต่ต่างกับฐานข้อมูลทั่วไปตรงที่รูปแบบการใช้งาน คือ ไดเรกทอรีประกอบด้วยข้อมูลที่มักถูกอ่านบ่อยๆ และแทบจะไม่ถูกแก้ไขเลย ในขณะที่ฐานข้อมูลทั่วไปเน้นไปที่การจัดการและการดูแลข้อมูล ซึ่งมักมีการเปลี่ยนแปลงเสมอ ผลที่ตามมาคือ ไดเรกทอรีนั้นไม่ต้องการการจัดการที่ซับซ้อนในการทำทรานส์แอ็กชั่นเหมือนกับฐานข้อมูลโดยทั่วไป แต่ไดเรกทอรีนั้นต้องมีความสามารถค้นหาข้อมูลได้ โดยใช้เวลาตอบสนองน้อย การที่จะทำให้ใช้เวลาตอบสนองในการค้น

หาข้อมูลน้อยนั้น สามารถทำได้โดยสร้างข้อมูลที่เหมือนกันหลายๆ ชุด ซึ่งทำให้สามารถหาข้อมูลไปพร้อมกันได้ และเวลาที่ใช้ในการค้นหาข้อมูลก็จะลดลง

5.3.2 การทำงานของไดเรกทอรีเซอร์วิซ

ไดเรกทอรีเซอร์วิซประกอบด้วยเอนทรี (Entry) ซึ่งภายในเอนทรีประกอบด้วยข้อมูล เช่น ในไดเรกทอรีมีเอนทรีที่อธิบายถึงอุปกรณ์ในระบบเครือข่าย ได้แก่ เครื่องพิมพ์ เป็นต้น ข้อมูลที่อยู่ภายในเอนทรีถูกเก็บในแอตทริบิวต์ของเอนทรีนั้น และแอตทริบิวต์จะเป็นตัวกำหนดชนิดของข้อมูลที่เก็บด้วย ยกตัวอย่างเช่น แอตทริบิวต์ที่อธิบายถึงบุคคล อาจประกอบด้วย ชื่อ, เบอร์โทรศัพท์ และอีเมล์แอดเดรส

Entry ของ Barbara Jensen อาจมีแอตทริบิวต์ดังนี้

cn : Barbara Jensen

mail : baba@ace.com

telephone number: 555-1212

roam number : 3995

นอกจากนี้ใน 1 แอตทริบิวต์สามารถมีค่าได้หลายค่าเช่น

cn : Jennifer Jensen

mail : jen@ace.com

telephone number: 555-1213

telephone number: 555-2029

roam number : 3996

ไดเรกทอรีเซอร์วิซออกแบบมาให้จัดการกับเอนทรีและแอตทริบิวต์ในไดเรกทอรี ซึ่งจะทำให้ผู้ใช้และโปรแกรมประยุกต์อื่นๆ สามารถใช้เอนทรีและแอตทริบิวต์นั้นได้ ยกตัวอย่างเช่น ผู้ใช้อาจใช้ไดเรกทอรีเซอร์วิซเพื่อค้นหาเบอร์โทรศัพท์ของบุคคลหนึ่ง เป็นต้น

รูปแบบของไดเรกทอรีเซอร์วิซขึ้นอยู่กับเนมสเปซใน X.500 (X.500 เป็นมาตรฐานของ CCITT ซึ่งกำหนดโพรโตคอลและรูปแบบของข้อมูลสำหรับโกลบอลไดเรกทอรีเซอร์วิซ (Global Directory service) ที่ไม่ขึ้นกับโปรแกรมประยุกต์ และแพลตฟอร์มของระบบเครือข่าย) เนมสเปซจะเป็นสเตทอย่างชัดเจน และเป็นลำดับขั้นด้วย ทุกเอนทรีใน X.500 Directory Information Tree (DIT) จะเป็นกลุ่มของแอตทริบิวต์ ในแต่ละแอตทริบิวต์ประกอบด้วยข้อมูลซึ่งอาจมีมากกว่า 1 ข้อมูลได้

ในมาตรฐาน X.500 ได้กำหนด ออบเจ็กต์คลาสพื้นฐานสำหรับไดเรกทอรีไว้ 17 ออบเจ็กต์คลาส ดังต่อไปนี้

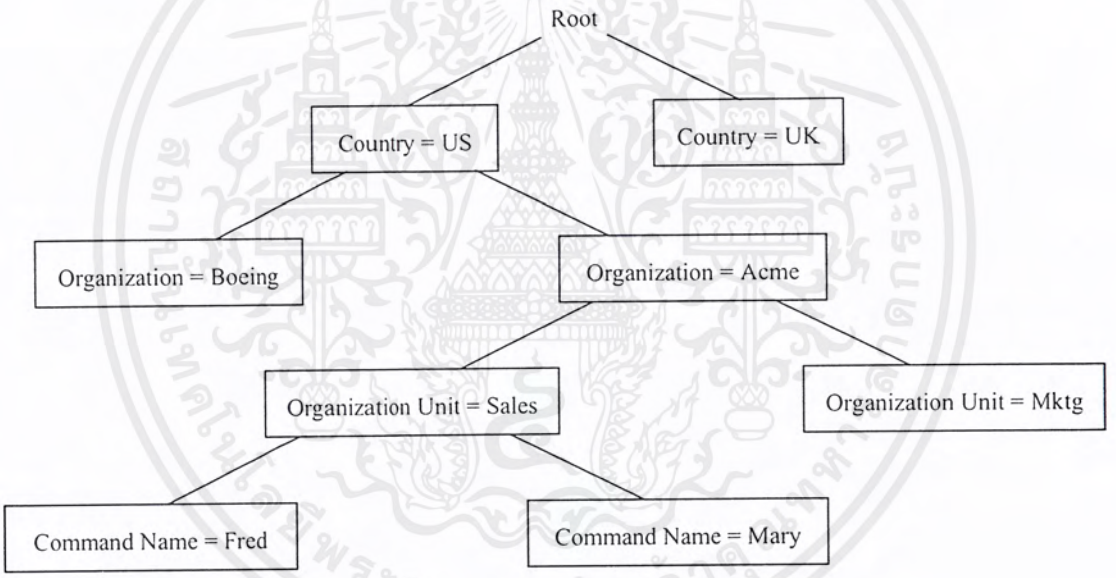
- Atlas
- Country
- Locality
- Organization
- Organization Unit

- Person

ออบเจ็กต์ในออบเจ็กต์คลาสเหล่านี้ถูกอธิบายโดยแอตทริบิวต์ของมัน แอตทริบิวต์พื้นฐานใน X.500 มี 40 แอตทริบิวต์ บางส่วนของแอตทริบิวต์เหล่านี้ได้แก่

- Common Name (CU)
- Organization Name (O)
- Organization Unit Name (OU)
- Locality Name (L)
- Street Address (SA)
- State or Province Name (S)
- Country ©

แต่ละเอนทรีใน X.500 ถูกอ้างอิงโดยใช้ "Distinguished Name" ยกตัวอย่างเช่น



รูปที่ 5-3 แสดงตัวอย่างเอ็นทรีใน X.500

จากรูป เมื่อต้องการจะอ้างถึงเอนทรีที่มี Common Name = Fred สามารถอ้างอิงได้โดย {C = US, O = Acme, CU = Sales, CN = Fred}

ในการอ้างอิงเอนทรีนั้นเริ่มต้นจากระดับบนสุดในลำดับชั้น ซึ่งคือรูต (Root) แล้วไล่ลงมายังชั้นที่ต่ำกว่า

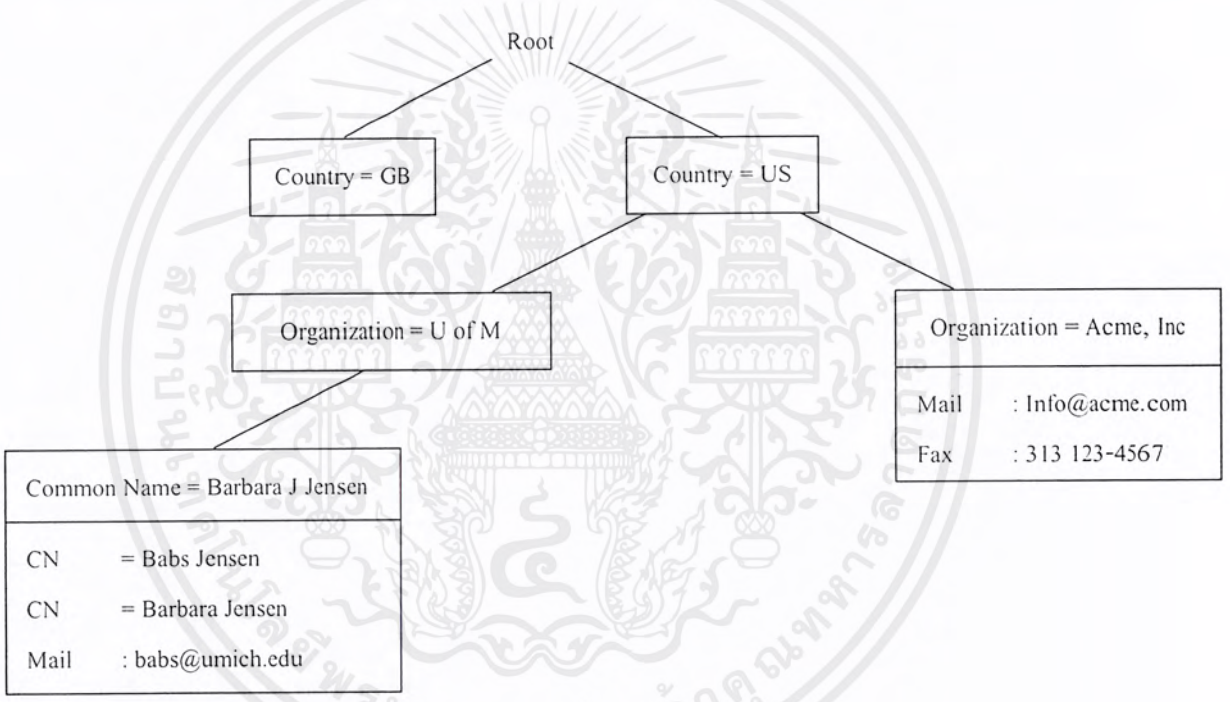
โปรแกรมประยุกต์และยูสเซอร์ สามารถเข้าถึงไดเรกทอรีได้โดยผ่านทาง Directory User Agent (DUA) ซึ่ง DUA จะส่งคำร้องขอเข้าถึงไดเรกทอรีไปให้ Directory System Agent (DSA) โดยใช้ Directory Access Protocol

5.3.3 LDAP

LDAP ย่อมาจากคำว่า Lightweight Directory Access Protocol ถูกสร้างขึ้นมาเพื่อใช้เป็นโพรโตคอลในการที่ไคลเอ็นต์จะขอเข้าถึงไดเรกทอรีที่ใช้มาตรฐาน X.500 ได้

5.3.3.1 ชนิดของข้อมูลที่เก็บในไดเรกทอรี

LDAP มีรูปแบบไดเรกทอรี คล้ายกับ X.500 คือใช้ลำดับชั้นของเอนทรีแบบออบเจ็กต์ออเรียนเท็ดข้อมูลถูกจัดให้เป็นแบบลำดับชั้น เริ่มต้นจากรูตและขยายลงมาจนถึงแต่ละเอนทรีในระดับบนสุดของลำดับชั้นจะเป็นองค์กรที่ใหญ่ และต่ำลงมาจะเป็นองค์การที่เล็กลงเรื่อยๆ ส่วนในระดับต่ำสุดของระดับชั้นจะเป็นแต่ละเอนทรีดังตัวอย่างต่อไปนี้



รูปที่ 5-4 ตัวอย่างการเก็บข้อมูลใน LDAP

5.3.3.2 การอ้างอิงข้อมูลในไดเรกทอรีของ LDAP

แต่ละเอนทรีถูกอ้างอิงโดยใช้ Distinguished Name (DN) ซึ่ง DN เกิดจากการเอาชื่อของเอนทรีซึ่งเรียกว่า Relative Distinguished Name (RDN) ต่อด้วยชื่อของเอนทรีที่อยู่เหนือมันขึ้นไป ยกตัวอย่างเช่น

Entry Barbara Jensen ในตัวอย่างข้างบนมี RDN คือ “CN = Barbara J Jensen” และ DN คือ “CN = Barbara J Jensen, O = U of M, C = US”

ข้อมูลที่เก็บในไดเรกทอรีสามารถถูกเก็บไว้ที่ LDAP เซิร์ฟเวอร์หลายตัวได้ ทำให้มี LDAP เซิร์ฟเวอร์บางตัวที่กำหนดให้สามารถอ้างไปถึง LDAP เซิร์ฟเวอร์ตัวอื่นได้

5.3.3.3 การทำงานของ LDAP เซิร์ฟเวอร์ และ LDAP ไคลเอ็นต์

LDAP ไดเรกทอรีเซอร์วิซอยู่บนพื้นฐานของรูปแบบแบบไคลเอ็นต์-เซิร์ฟเวอร์ LDAP เซิร์ฟเวอร์จะเก็บข้อมูลที่ถูกสร้างเป็น LDAP ไดเรกทอรี

LDAP ไคลเอ็นต์จะสามารถเข้าถึงข้อมูลเหล่านี้ได้ และ LDAP โพรโตคอลได้กำหนดการทำงานที่ไคลเอ็นต์สามารถทำได้ไว้ดังนี้

1. ค้นหาและดึงข้อมูลของเอนทรีจากไดเรกทอรี
2. เพิ่มเอนทรีใหม่เข้าไปในไดเรกทอรี
3. ปรับปรุง (Update) เอนทรีในไดเรกทอรี
4. ลบเอนทรีในไดเรกทอรี
5. เปลี่ยนชื่อเอนทรีในไดเรกทอรี

LDAP มีการป้องกันข้อมูล เพื่อไม่ให้ผู้อื่นสามารถดูข้อมูลได้ โดยใช้การพิสูจน์ตนของไคลเอ็นต์ที่ต้องการเข้าถึงข้อมูล

บทที่ 6

Single Sign-On

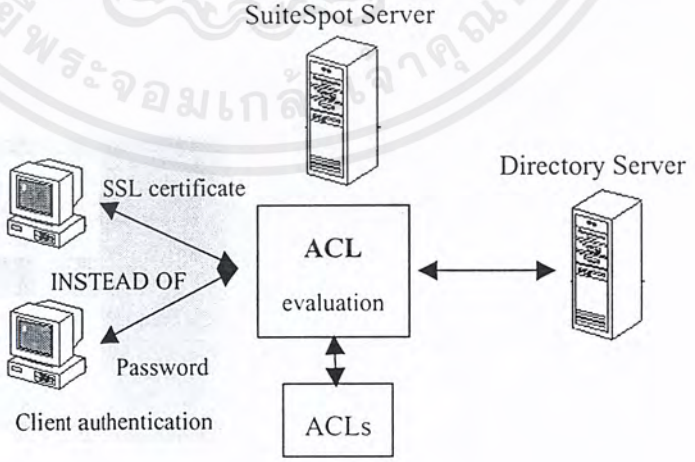
ยูสเซอร์โดยทั่วไปจะมีรหัสผ่านโดยเฉพาะสำหรับแต่ละเซิร์ฟเวอร์ที่ยูสเซอร์ต้องการขอใช้บริการ การมีรหัสผ่านหลายๆ รหัสผ่านอย่างนี้ จะทำให้ยูสเซอร์เกิดความลำบากที่ต้องจดจำรหัสทั้งหมด และยังทำให้ผู้ดูแลระบบเกิดความลำบากด้วย นอกจากนี้ยังทำให้ยูสเซอร์ตั้งรหัสผ่านที่ไม่ดี หรือจดจำรหัสผ่านนี้ไว้ ทำให้ความปลอดภัยของรหัสผ่านน้อยลง

แนวทางในการแก้ปัญหาที่กล่าวมาข้างต้น สามารถทำได้โดยให้ยูสเซอร์ล็อกอินครั้งแรกเพียงครั้งเดียวเท่านั้น แต่สามารถใช้บริการจากเซิร์ฟเวอร์ทั้งหมดที่ยูสเซอร์มีสิทธิ์ที่จะใช้ได้ โดยที่ไม่ต้องส่งรหัสผ่านนี้ไปบนระบบเครือข่าย วิธีการที่กล่าวมานี้เรียกว่า Single Sign-On

6.1 ความรู้เบื้องต้นเกี่ยวกับ Single Sign-On

เมื่อยูสเซอร์ทำการร้องขอทรัพยากรจากเซิร์ฟเวอร์ เซิร์ฟเวอร์จะตรวจสอบ Access Control Lists (ACLs) ซึ่งเก็บข้อมูลเกี่ยวกับทรัพยากรนั้น ถ้าเซิร์ฟเวอร์ที่เก็บทรัพยากรที่ยูสเซอร์ต้องการใช้ ต้องการพิสูจน์ยูสเซอร์ เซิร์ฟเวอร์จะร้องขอการพิสูจน์ตนของไคลเอ็นต์จากยูสเซอร์ โดยอาจจะอยู่ในรูปแบบต่างๆ เช่น ชื่อและรหัสผ่าน หรือ ดิจิตอลเซอร์ติฟิเคตที่แสดงในรูปของโพรโตคอล SSL

หลังจากที่เซิร์ฟเวอร์ตรวจสอบข้อมูลของยูสเซอร์ (โดยส่วนมากจะประกอบด้วย ข้อมูลของยูสเซอร์ และกลุ่มของยูสเซอร์ที่เก็บใน Lightweight Directory Access Protocol (LDAP) ไดเรกทอรี) เสร็จแล้ว เซิร์ฟเวอร์ก็จะไปตรวจสอบใน ACL และอนุญาตหรือไม่อนุญาตให้ยูสเซอร์ที่ร้องขอมานั้นใช้ทรัพยากรตามสิทธิ์ที่มีของแต่ละยูสเซอร์



รูปที่ 6-1 แสดงถึงขั้นตอนพื้นฐานในการตรวจสอบ ACL โดยใช้การพิสูจน์ตนแบบ ส่งชื่อและรหัสผ่านไปบนระบบเครือข่าย หรือใช้เซอร์ติฟิเคตบน SSL

ประโยชน์ของการใช้วิธีนี้

1. สามารถใช้งานได้ง่าย ยูสเซอร์สามารถล็อกอินเพียงครั้งเดียว และได้รับอนุญาตให้ใช้ทรัพยากรจากเซิร์ฟเวอร์ทั้งหมดที่ยูสเซอร์มีสิทธิ์ที่จะใช้
2. รหัสผ่านมีความปลอดภัย เนื่องจากจะไม่มีการส่งรหัสผ่านไปบนระบบเครือข่าย
3. สามารถจัดการได้ง่าย เนื่องจากผู้ดูแลระบบเก็บข้อมูลว่ายูสเซอร์สามารถขอใช้บริการได้จากเซิร์ฟเวอร์ใดบ้าง ซึ่งจัดการโดยซอฟต์แวร์ และการเก็บในลักษณะนี้จะง่ายกว่าการเก็บยูสเซอร์เนมและรหัสผ่าน นอกจากนี้ยังเกิดการเปลี่ยนแปลงน้อยกว่าอีกด้วย
4. การควบคุมการเข้าถึงไม่ได้รับผลกระทบ เนื่องจาก SSO จะเกี่ยวข้องแค่วิธีการพิสูจน์ตนของยูสเซอร์ ไม่ได้เกี่ยวข้องกับวิธีการในการควบคุมการเข้าถึง

การพิสูจน์ตนของไคลเอ็นต์ที่ใช้ชื่อและรหัสผ่าน เรียกว่า การพิสูจน์ตนแบบง่าย (Basic Authentication) ส่วน SSO ที่ ใช้เซอร์ติฟิเคตในการพิสูจน์ตน บางครั้งเรียกว่า การพิสูจน์ตนแบบปลอดภัย (Strong Authentication)

เซอร์ติฟิเคตเป็นเอกสารอิเล็กทรอนิกส์ที่ใช้พิสูจน์ตนของแต่ละบุคคล บริษัท หรืออื่นๆ Certificate Authentication (CAs) เป็นส่วนที่ทำหน้าที่ตรวจสอบหลักฐานและออกเซอร์ติฟิเคต CAs อาจเป็นหน่วยงานที่ 3 ที่เป็นอิสระจากองค์กร หรืออาจเป็นหน่วยงานภายในองค์กรที่ใช้ซอร์ฟแวร์ในการออกเซอร์ติฟิเคตก็ได้

6.2 การพิสูจน์ตนของผู้ใช้และ Single Sign-On

ข้อมูลถูกส่งจากคอมพิวเตอร์เครื่องหนึ่งไปยังเครื่องอื่นๆ โดยผ่านระบบเครือข่าย TCP/IP จะต้องผ่านส่วนอื่นๆ อีกมากมายกว่าจะถึงจุดหมายปลายทาง สมมติว่าในการส่งข้อมูลนี้ไม่มีการดักจับข้อมูลจากบุคคลอื่นหรือถูกแก้ไขจากบุคคลอื่น แต่ก็ยังมีปัญหาที่เกิดขึ้นอีก คือ ยูสเซอร์ไม่สามารถรู้ได้ว่าเว็บไซต์ที่พวกเขาเข้าไปนั้น มีจุดมุ่งหมายอย่างไร และผู้ดูแลระบบก็ไม่สามารถรู้ได้ว่าใครเข้าเว็บไซต์นี้

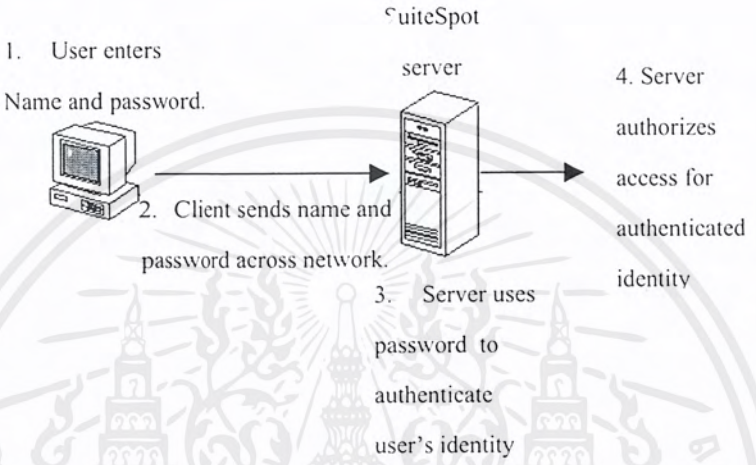
ความเสี่ยงในด้านความปลอดภัยนี้ สามารถแก้ไขได้ในบางส่วน โดยใช้ซอร์ฟแวร์ที่ไคลเอ็นต์และยูสเซอร์ที่รองรับการพิสูจน์ตน อย่างเช่นในการขอใช้บริการจากเซิร์ฟเวอร์ ยูสเซอร์จะต้องพิมพ์ชื่อและรหัสผ่านก่อนที่จะได้รับอนุญาต และเซิร์ฟเวอร์ใช้ข้อมูลเหล่านี้และข้อมูลจากฐานข้อมูลของเซิร์ฟเวอร์เองในการพิสูจน์ยูสเซอร์ เพื่อพิสูจน์ว่ายูสเซอร์เป็นยูสเซอร์ที่ถูกต้องตามที่ยูสเซอร์อ้างถึง

ในส่วนของการพิสูจน์ตน โดยมากจะไม่ได้รวมถึงการป้องกันด้านความเป็นส่วนตัวของข้อมูลหรือความเป็นส่วนเดียวกันของข้อมูล SSL เป็นมาตรฐานที่ใช้ในการพิสูจน์ตน และยังรวมถึงการป้องกันความเป็นส่วนตัว และความเป็นส่วนเดียวกันของข้อมูลด้วย SSL เป็นโพรโตคอลที่ทำงานบน TCP/IP และอยู่ภายใต้ HTTP LDAP IMAP NNTP SSL อนุญาตให้เซิร์ฟเวอร์สามารถพิสูจน์ตนต่อไคลเอ็นต์ได้ และอนุญาตให้ไคลเอ็นต์พิสูจน์ตนต่อยูสเซอร์ได้ นอกจากนี้ การทำงานทั้ง 2 ส่วนนี้ยังใช้การเข้ารหัสอีกด้วย

การพิสูจน์ตนของไคลเอ็นต์ เป็นขั้นตอนที่จำเป็นในการรักษาความปลอดภัยบนระบบเครือข่าย ในที่นี้จะแบ่งการพิสูจน์ตนออกเป็น 2 ประเภท คือ

1. การพิสูจน์ตนแบบง่าย
2. การพิสูจน์ตนแบบปลอดภัย

6.2.1 การพิสูจน์ตนแบบง่าย



รูปที่ 6-2 แสดงขั้นตอนพื้นฐานในการพิสูจน์ตนของยูสเซอร์โดยใช้ชื่อและรหัสผ่าน

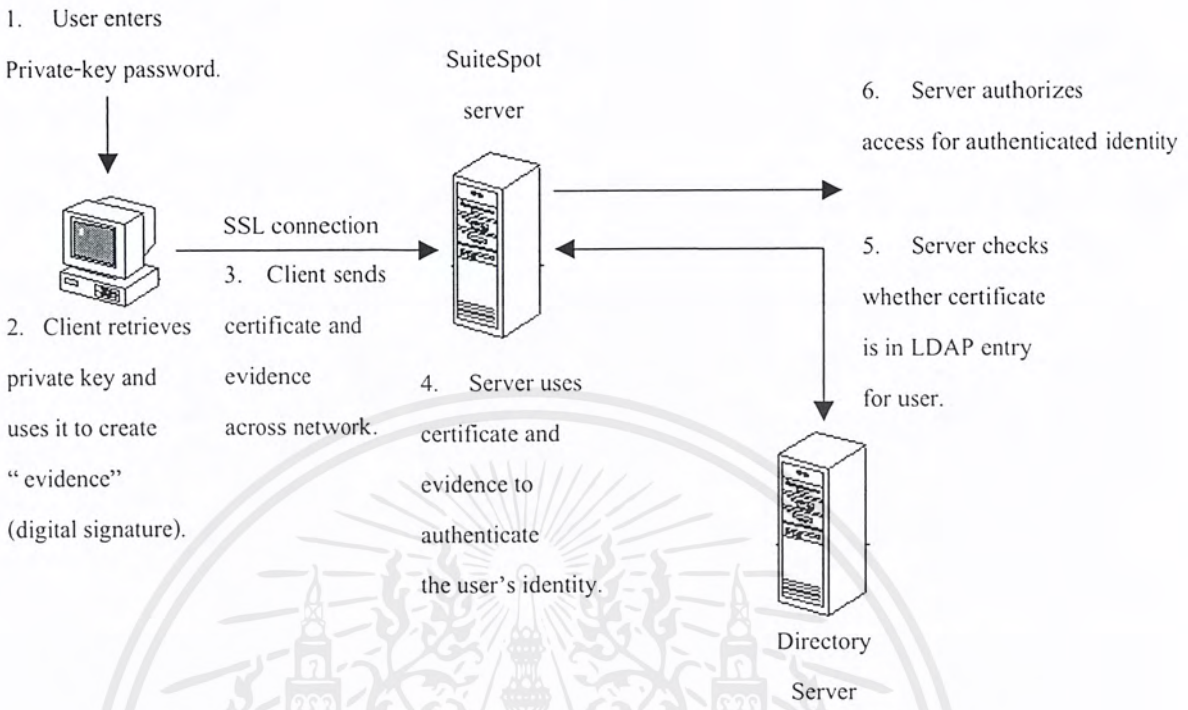
ขั้นตอนต่างๆ มีดังนี้

1. ยูสเซอร์ต้องพิมพ์ชื่อและรหัสผ่าน สำหรับเซิร์ฟเวอร์ที่ยูสเซอร์ต้องการจะติดต่อ
2. ไคลเอ็นจะรับชื่อและรหัสผ่านนี้ส่งไปให้เซิร์ฟเวอร์ผ่านระบบเครือข่าย
3. เซิร์ฟเวอร์จะค้นหาชื่อและรหัสผ่านในฐานข้อมูลของมัน และตรวจสอบกับที่ได้รับมา ถ้าตรงกันก็แสดงว่ายูสเซอร์นั้นถูกต้อง
4. เซิร์ฟเวอร์ตรวจสอบ ACL ของมัน และกำหนดว่ายูสเซอร์คนนี้สามารถใช้ทรัพยากรจากเซิร์ฟเวอร์ได้ และจากนั้นก็อนุญาตให้ยูสเซอร์สามารถเข้าถึงเซิร์ฟเวอร์ได้

6.2.2 การพิสูจน์ตนแบบปลอดภัย

Single Sign-On ใช้เซอร์ติฟิเคตอันเดียว แทนการใช้รหัสผ่านหลายๆ รหัสผ่านในการพิสูจน์ตนต่อหลายๆเซิร์ฟเวอร์

ในการพิสูจน์ตนของยูสเซอร์ต่อเซิร์ฟเวอร์ ไคลเอ็นต์จะสร้างข้อมูลที่สุ่มขึ้นมาชุดหนึ่ง และส่งทั้งเซอร์ติฟิเคตของข้อมูลชุดที่สร้างขึ้นมานั้นไปให้แก่เซิร์ฟเวอร์ผ่านระบบเครือข่าย ขั้นตอนการพิสูจน์ตนสามารถแสดงได้ดังรูป



รูปที่ 6-3 แสดงขั้นตอนการพิสูจน์ตน

มีการทำงานดังนี้

1. ไคลเอ็นต์จะให้ยูสเซอร์ใส่รหัสผ่านในครั้งแรกที่ยูสเซอร์ติดต่อครั้งเดียวเท่านั้น จากนั้นไคลเอ็นต์จะดูในฐานข้อมูลของมันว่ามีไพรเวตคีย์เป็นอะไร
2. ไคลเอ็นต์จะเอาไพรเวตคีย์ที่ได้มานี้ไปเข้ารหัสข้อมูลที่สุ่มขึ้นมาชุดหนึ่ง ข้อมูลที่เข้ารหัสแล้วนี้เรียกว่าดิจิตอลซิกเนเจอร์ (digital signature)
3. ไคลเอ็นต์ส่งเซอร์ติฟิเคตและดิจิตอลซิกเนเจอร์ไปให้เซิร์ฟเวอร์
4. เซิร์ฟเวอร์ใช้ดิจิตอลซิกเนเจอร์และเซอร์ติฟิเคตที่ได้รับมาพิสูจน์ยูสเซอร์
5. เซิร์ฟเวอร์เอาเซอร์ติฟิเคตที่ได้รับมาไปเปรียบเทียบกับเซอร์ติฟิเคตที่เก็บในเอนทรีใน LDAP ไดเรกทอรี
6. ถ้าการเปรียบเทียบถูกต้อง เซิร์ฟเวอร์จะตรวจสอบใน ALC เพื่อกำหนดว่ายูสเซอร์สามารถใช้ทรัพยากรอะไรได้บ้าง และจากนั้นก็อนุญาตให้ยูสเซอร์เข้าถึงเซิร์ฟเวอร์ได้

6.3 การเตรียม LDAP ไดเรกทอรี

จากรูปที่ 6-3 เห็นได้ว่า เซิร์ฟเวอร์ต้องหาเซอร์ติฟิเคตของยูสเซอร์ใน LDAP ไดเรกทอรี และเพื่อที่สนับสนุน SSO จะต้องทำ 3 ข้อต่อไปนี้ เวลาสร้างไดเรกทอรี

- ต้องทำให้สามารถแน่ใจได้ว่า distinguished name (DN) ที่อยู่ในเซอร์ติฟิเคตสามารถเปลี่ยนเป็นไดเรกทอรีทรีได้
- แอตทริบิวต์ที่อยู่ในเซอร์ติฟิเคตของยูสเซอร์ต้องมีในเอนทรีของยูสเซอร์
- ต้องจำกัดการเข้าถึงเอนทรีโดยให้เข้าถึงเฉพาะส่วนที่เก็บเซอร์ติฟิเคตของยูสเซอร์

6.4 Certificate, DN และการค้นหาของ LDAP

เซอร์ติฟิเคตแสดงความเกี่ยวพันธ์ระหว่าง DN กับพับลิกคีย์

DN เป็นชื่อของเอนทรีเช่น

uid = doe, e=doe@netscape.com , cn=John Doe,

O=Netscape Communications Corps, c=US.

เซิร์ฟเวอร์จะใช้ไฟล์ที่ชื่อ certmap.conf เพื่อเป็นตัวบอกว่าส่วนไหนใน DN ที่นำมาใช้ในการค้นหาของยูสเซอร์ภายใน LDAP ไดเรกทอรี

การค้นหาเอนทรีในไดเรกทอรีไม่จำเป็นต้องใช้ข้อมูลทั้งหมดใน DN ก็ได้ ใช้แค่มั่นใจได้ว่าเมื่อเปลี่ยนจาก DN ไปเป็นเอนทรีแล้ว จะได้เอนทรีเดียวเท่านั้น

บทที่ 7
Windows 2000

7.1 Active Directory

แอ็กทีฟไดเรกทอรี (Active Directory) เป็นส่วนสำคัญในวินโดวส์ 2000 ซึ่งเป็นตัวจัดการไดเรกทอรีเซอร์วิซ ซึ่งออกแบบมาเพื่อใช้กับระบบเครือข่ายแบบกระจายแอ็กทีฟไดเรกทอรีช่วยในการใช้ข้อมูลร่วมกันและจัดการข้อมูลเกี่ยวกับทรัพยากรและผู้ใช้ระบบ นอกจากนี้แอ็กทีฟไดเรกทอรียังทำหน้าที่เป็นศูนย์กลางจัดการความปลอดภัย ทำให้ระบบปฏิบัติการสามารถพิสูจน์ผู้ใช้ระบบ และควบคุมการเข้าถึงทรัพยากรระบบแอ็กทีฟไดเรกทอรียังเป็นตัวรวบรวม เพื่อให้ระบบเป็นหนึ่งเดียว และมีการจัดการที่ดี ทำให้ไม่จำเป็นต้องมีการจัดการระบบแยกในแต่ละไดเรกทอรี

การใช้ไดเรกทอรีเซอร์วิซมีผลดีคือ

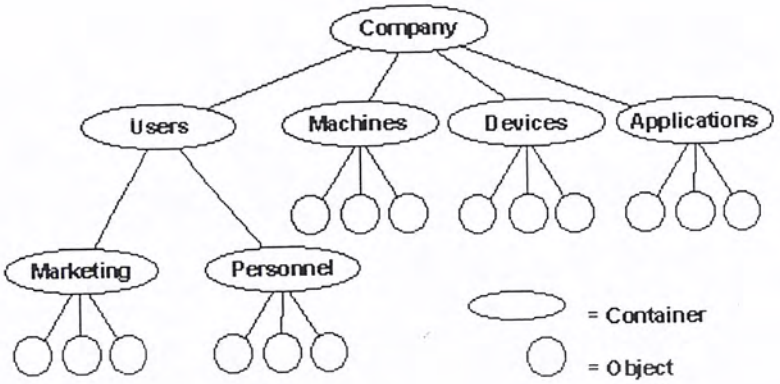
- การจัดการที่ง่าย เนื่องจากมีการจัดการข้อมูลเกี่ยวกับผู้ใช้ระบบ โปรแกรมประยุกต์ อุปกรณ์ต่างๆ ในระบบเครือข่าย ที่ศูนย์กลางเพียงที่เดียว
- มีความปลอดภัยสูง จัดการการใช้ทรัพยากรระบบของผู้ใช้ระบบ และมีการจัดการระบบด้วยเครื่องมือที่มีประสิทธิภาพสูง หรือความปลอดภัยในการบริการ
- การทำงานร่วมกันของหลายแพลตฟอร์มสามารถติดต่อกับไดเรกทอรีอื่นที่มีมาตรฐานได้

7.1.1 การทำงานของ Active Directory

แอ็กทีฟไดเรกทอรีเก็บข้อมูลขององค์กรเป็นแบบขั้นลำดับ ด้วยวิธีออบเจ็กต์โอเรียนเท็ด และรองรับการติดต่อแบบมัลติยูสเซอร์และระบบเครือข่ายแบบกระจาย

7.1.1.1 การรวบรวมแบบขั้นลำดับ

แอ็กทีฟไดเรกทอรีใช้ออบเจ็กต์ในการแสดงทรัพยากรของระบบเครือข่าย เช่น ผู้ใช้ระบบ กลุ่ม เครื่อง อุปกรณ์ต่างๆ และโปรแกรมประยุกต์แอ็กทีฟไดเรกทอรีใช้คอนเทนเนอร์ในการแสดงกลุ่มที่จัดตั้งขึ้นมา เช่น แผนกการตลาด หรือ การรวมตัวของออบเจ็กต์ที่เกี่ยวข้องกัน เช่น ปริ๊นเตอร์ แอ็กทีฟไดเรกทอรีจะรวบรวมข้อมูลให้มีโครงสร้างแบบทรีซึ่งประกอบด้วยคอนเทนเนอร์และออบเจ็กต์



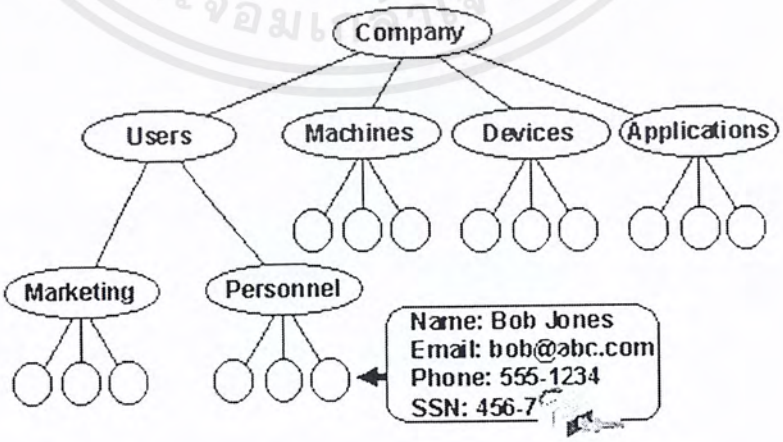
รูปที่ 7-1 แอ็กทีฟไดเรกทอรีรวบรวมข้อมูลแบบขั้นลำดับเพื่อให้จัดการข้อมูลในระบบได้ง่าย

นอกจากนี้แอ็กทีฟไดเรกทอรียังจัดการเกี่ยวกับความสัมพันธ์ระหว่างคอนเทนเนอร์และออบเจ็กต์เพื่อการดูระบบอย่างเป็นศูนย์กลาง เพื่อการหา จัดการ และใช้ระบบเครือข่ายได้ง่ายขึ้น

จากรูปที่ 7-1 คอนเทนเนอร์ใช้แสดงกลุ่มของผู้ใช้ระบบ เครื่อง อุปกรณ์ และโปรแกรมประยุกต์คอนเทนเนอร์สามารถซ้อนกันได้ (สร้างคอนเทนเนอร์ในคอนเทนเนอร์) เพื่อให้เหมือนกับโครงสร้างขององค์กรอย่างแน่นอน การรวมกลุ่มของออบเจ็กต์ทำให้ผู้ดูแลระบบสามารถจัดการได้อย่างมีประสิทธิภาพ และมีความแน่นอน

7.1.1.1.2 Object Storage

แอ็กทีฟไดเรกทอรีเก็บข้อมูลที่เกี่ยวกับระบบเครือข่ายเป็นแบบออบเจ็กต์ ออบเจ็กต์เหล่านี้สามารถกำหนดแอตทริบิวต์ซึ่งช่วยอธิบายลักษณะเฉพาะของแต่ละออบเจ็กต์ซึ่งช่วยให้องค์กรเก็บข้อมูลที่หลากหลายในไดเรกทอรีและสามารถควบคุมได้อย่างมีประสิทธิภาพ

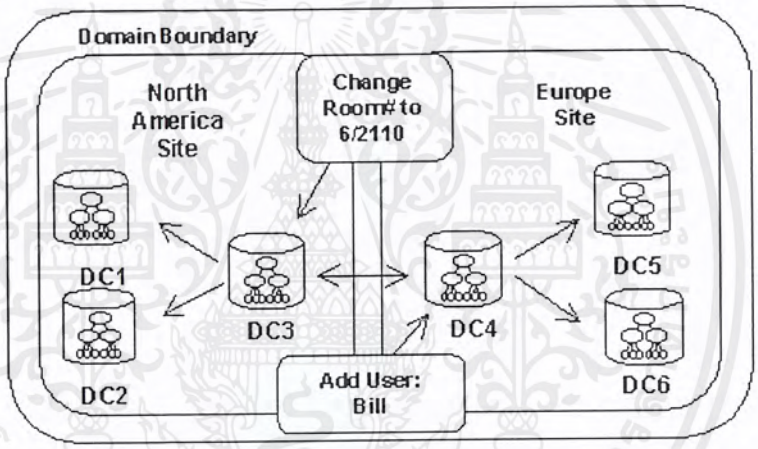


รูปที่ 7-2 ออบเจ็กต์และแอตทริบิวต์ในแอ็กทีฟไดเรกทอรีป้องกันโดย ACL

จากรูปที่ 7-2 การรักษาความปลอดภัยในระดับออบเจ็กต์และแอตทริบิวต์ ทำให้ผู้ดูแลระบบสามารถควบคุมการเข้าถึงข้อมูลในไดเรกทอรีของผู้ใช้ระบบ เช่น ข้อมูลของออบเจ็กต์ชื่อ Bob Jones ซึ่งประกอบด้วยแอตทริบริวต์ อีเมล์แอดเดรส เบอร์โทรศัพท์ และ SSN ข้อมูลทุกอย่างทุกคนสามารถดูได้ ยกเว้น Social Security Number ซึ่งผู้ใช้ระบบคนอื่นๆ ไม่ได้สามารถดูได้

7.1.1.3 Multi-Master Replication

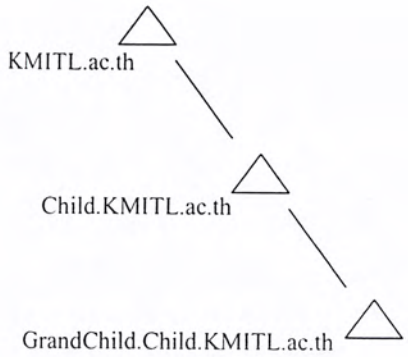
แอ็กทีฟไดเรกทอรีใช้มัลติมาสเตอร์ เพื่อให้ระบบมีประสิทธิภาพสูง สามารถเชื่อได้ และมีความยืดหยุ่น ในรูปที่ 7-3 องค์กรได้สร้างสำเนาของไดเรกทอรีเรียกว่าไดเรกทอรีเรพลิคา (Directory Replica) ไว้อยู่คนละระบบเครือข่าย เมื่อมีการเปลี่ยนแปลงบนระบบเครือข่าย จะมีการกระจายข้อมูลไปอย่างอัตโนมัติออกไปนอกระบบ



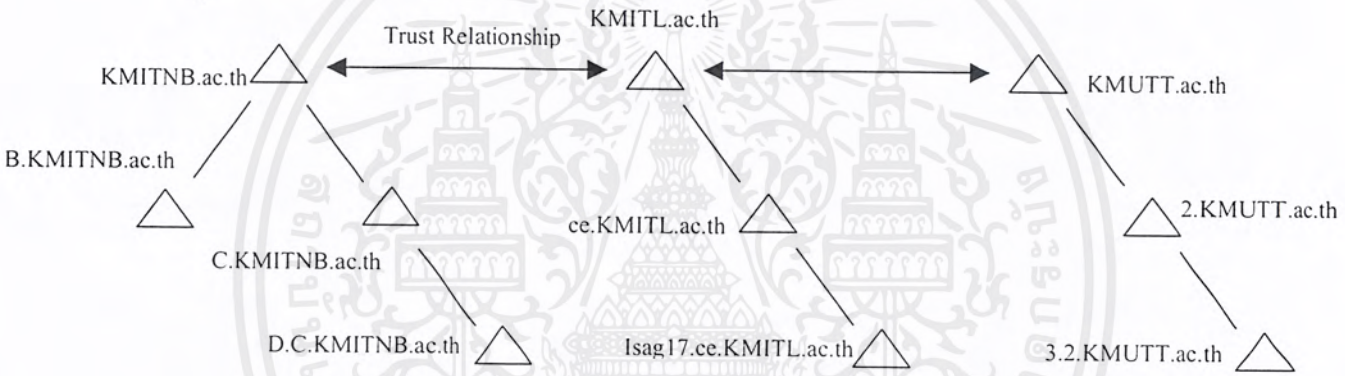
รูปที่ 7-3 แอ็กทีฟไดเรกทอรีสนับสนุนมัลติมาสเตอร์เพื่อความยืดหยุ่น สามารถเชื่อถือได้ และประสิทธิภาพสูง

7.1.2 คำศัพท์พื้นฐานเกี่ยวกับแอ็กทีฟไดเรกทอรี

- Domain กลุ่มของคอมพิวเตอร์ที่เป็นส่วนหนึ่งในระบบเครือข่าย ซึ่งมีการใช้ฐานข้อมูลที่เป็นไดเรกทอรีร่วมกันโดเมนจะถูกจัดเป็นขั้นลำดับ แต่ละโดเมนจะมีชื่อไม่ซ้ำกัน บนอินเทอร์เน็ต โดเมนจะถูกกำหนดด้วย IP แอดเดรส ทางด้านอินเทอร์เน็ต โดเมนมักเป็นตัวบอกชนิดของแอดเดรสนั้น เช่น .com .org
- Domain Trees and Forest ใน 1 โดเมนนั้นจะมีอย่างน้อย 1 โดเมนคอนโทรลเลอร์ ถ้าในระบบเครือข่ายที่มีหลายโดเมนจำเป็นต้องสร้างหลายโดเมนโดยฟอเรส (Forest) คือโดเมนตั้งแต่ 1 โดเมนขึ้นไปที่มีการใช้ Schema และ Global Catalog เดียวกัน ถ้ามีหลายโดเมนในฟอเรสมี DNS โดเมนเนม ที่ต่อเนื่องกัน โครงสร้างแบบนี้เรียกว่าโดเมนทรี



รูปที่ 7-4 โดเมนทรี ในโดเมนฟอเรส



รูปที่ 7-5 การติดต่อระหว่างโดเมนทรี

จากรูป 12.5 ถ้ามีหลายโดเมนซึ่งมี DNS โดเมนเนม ไม่ต่อเนื่องกัน ถือว่าเป็นการแยกทรี ในฟอเรส ในฟอเรสอาจมีตั้งแต่ 1 โดเมนทรี ขึ้นไป โดเมนแรกในฟอเรสถือว่าเป็นรูตโดเมน (Root Domain)

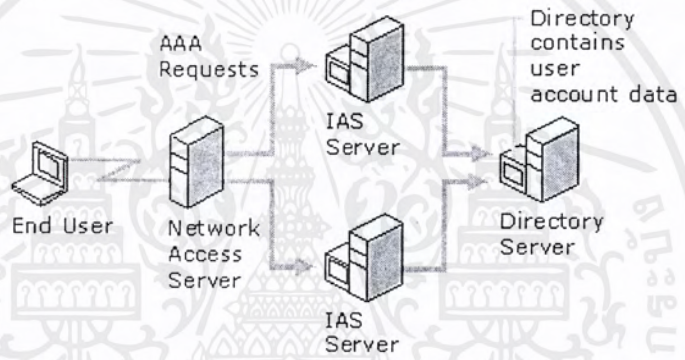
- Domain Controller คือ คอมพิวเตอร์ที่มีระบบปฏิบัติการคือ วินโดวส์ 2000 เซิร์ฟเวอร์ และใช้แอ็กทีฟไดเรกทอรีเป็นส่วนประกอบที่ช่วยจัดการทางด้านยูสเซอร์และคอมพิวเตอร์ ในแบบเครือข่ายโดเมนคอนโทรลเลอร์เก็บข้อมูลเป็นไดเรกทอรี รวมทั้งเก็บข้อมูลเกี่ยวกับการติดต่อระหว่าง User_Domain เช่น ล็อกอิน ค้นหาไดเรกทอรีในโดเมนมีได้มากกว่า 1 โดเมนคอนโทรลเลอร์ โดยปกติมักใช้มากกว่า 1 โดเมนคอนโทรลเลอร์ที่เป็นวินโดวส์ 2000 เซิร์ฟเวอร์สามารถซิงโครไนซ์ข้อมูลที่เป็นไดเรกทอรีกัน เพื่อให้แน่ใจว่าข้อมูลมีความถูกต้องตลอดเวลา โดยมี Back Up Domain Controller และ Primary Domain Controller ที่มีการอ่านและเขียนไดเรกทอรี

7.2 Internet Authentication Service (IAS)

IAS เป็นส่วนประกอบหนึ่งในวินโดวส์ 2000 เซิร์ฟเวอร์ ซึ่งรวมการทำงาน การพิสูจน์ตน การพิสูจน์สิทธิ์ การบันทึกการใช้งานของผู้ใช้ระบบ ในการติดต่อจากผู้ใช้ระบบผ่านทาง VPN หรือ การติดต่อผ่านทางสายโทรศัพท์ และ IAS ใช้โพรโตคอลที่เป็นมาตรฐาน นั่นคือ RADIUS โพรโตคอล

7.2.1 การพิสูจน์ตนและการพิสูจน์สิทธิ์ของ IAS

ในขั้นตอนของการพิสูจน์ผู้ใช้ระบบและขั้นตอนการให้สิทธิ์แก่ผู้ใช้ระบบ ในการเข้าใช้ระบบ มีNAS ซึ่งทำหน้าที่เป็นไคลเอ็นต์ของ IAS เซิร์ฟเวอร์ โดย NAS จะส่งข้อมูลของผู้ใช้ระบบไปให้แก่ IAS เซิร์ฟเวอร์ และรับการตอบสนองกลับจากเซิร์ฟเวอร์



รูปที่ 7-6 แสดงกระบวนการในการพิสูจน์ตนและพิสูจน์สิทธิ์ของ IAS

IAS เซิร์ฟเวอร์รับการขอติดต่อของผู้ใช้ระบบ ตรวจสอบการพิสูจน์ตนของผู้ใช้ระบบ พิสูจน์สิทธิ์ของผู้ใช้ระบบ และจากนั้นก็จะส่งข้อมูลที่จำเป็นต่อการให้บริการแก่ผู้ใช้ระบบไปให้ RADIUS ไคลเอ็นต์

7.2.2 ลักษณะเด่นของ IAS

1. รวมการพิสูจน์ตนไว้ที่ศูนย์กลาง สามารถรองรับการพิสูจน์ตนได้หลายโพรโตคอล
   - การติดต่อโดยโพรโตคอล PPP เช่น PAP CHAP MS-CHAP EAP
   - EAP รองรับการพิสูจน์ตนแบบอื่นได้ เช่น การใช้สมาร์ทการ์ดเซอร์ติฟิเคต รหัสผ่านแบบใช้ครั้งเดียว
   - Dialed Number Identification Service (DNIS) ซึ่งทำงานบนพื้นฐานของหมายเลขที่ผู้ใช้ติดต่อเข้ามา
   - Automatic Number Identification Service (ANI) ซึ่งทำงานบนพื้นฐานของหมายเลขของผู้ใช้ที่ติดต่อเข้ามา ANI มีอีกชื่อหนึ่งคือ Caller ID
   - Guest Authentication

2. รวมการพิสูจน์สิทธิ์

การพิสูจน์สิทธิ์ของ IAS ใช้รีโมตแอ็กเซสโพลิซี (Remote Access Polocies) ในการการหนดสิทธิ์ต่างๆ รีโมตแอ็กเซสโพลิซีเป็นข้อกำหนดในการติดต่อระบบเครือข่ายของผู้ใช้ระบบ ซึ่งทำให้ผู้ดูแลระบบกำหนดสิทธิ์ได้ง่ายขึ้น

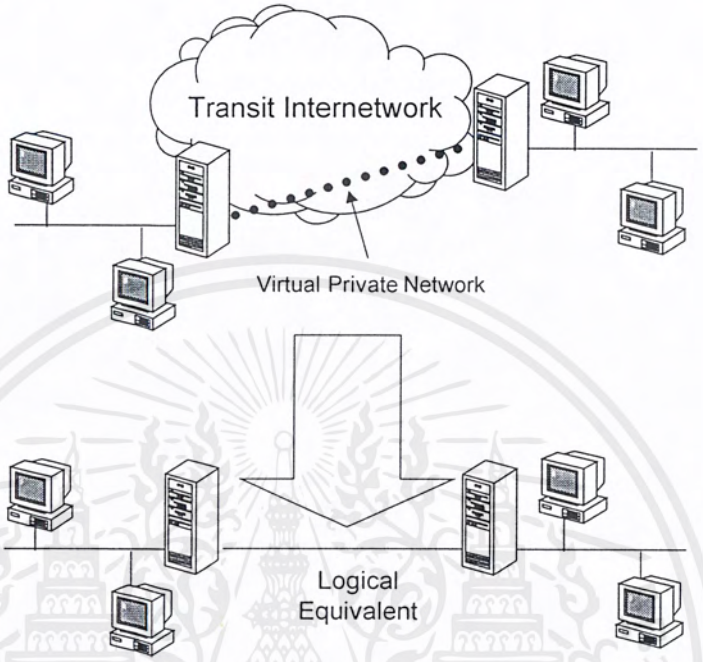
รีโมตแอ็กเซสโพลิซีเป็นตัวกำหนดว่าผู้ใดอนุญาตให้เข้าสู่ระบบได้ ซึ่งเป็นการทำงานพื้นฐานของการกำหนดสิทธิ์ แต่ยังมีการทำงานอื่นเพิ่มเติมอีก เพื่อความยืดหยุ่นในการอนุญาต โดยมีเงื่อนไขให้เลือกหลายอย่าง โดย

แอตทริบิวต์	ความหมาย
NAS IP Address	IP แอดเดรสของ NAS
Service Type	ชนิดของบริการที่ร้องขอเข้ามา เช่น ล็อกอิน (ตัวอย่าง เช่น การเชื่อมต่อโดยผ่าน telnet)
Framed Protocol	ชนิดของเฟรมที่มากกับแพคเก็จ เช่น PPP, SLIP
Called Station ID	เบอร์โทรศัพท์ของ NAS
Calling Station ID	เบอร์โทรศัพท์ของผู้ที่ติดต่อเข้ามา
NAS Port Type	ชนิดของตัวกลางการสื่อสารที่ใช้ของผู้ที่ติดต่อเข้ามา เช่น สายโทรศัพท์, ISDN, VPN
Day and Time Restriction	ช่วงเวลาที่อนุญาตให้ผู้ติดต่อสามารถติดต่อเข้ามาได้
Client IP Address	IP แอดเดรสของ NAS
NAS Manufacturer	ผู้ขาย NAS โดยของวินโดวส์ 2000 คือ Microsoft RAS
Client Friendly Name	ชื่อของ RADIUS ไคลเอ็นต์ที่ร้องของการพิสูจน์ตน
Windows Groups	ชื่อของกลุ่มยูสเซอร์ในวินโดวส์
Tunnel Type	ชนิดของ tunnel ที่สร้างโดยไคลเอ็นต์ โดย tunnel type ประกอบด้วย Peer-to-Peer Tunneling Protocol (PPTP) และ Layer Two Tunneling Protocol (L2TP)

ตารางที่ 7-1 แสดงข้อจำกัดในการติดต่อจากผู้ใช้

7.3 Virtual Private Network (VPN)

VPN (Virtual Private Network) เป็นการทำให้สามารถเชื่อมต่ออุปกรณ์ของระบบเครือข่ายหนึ่งโดยผ่านระบบเครือข่ายอื่นได้



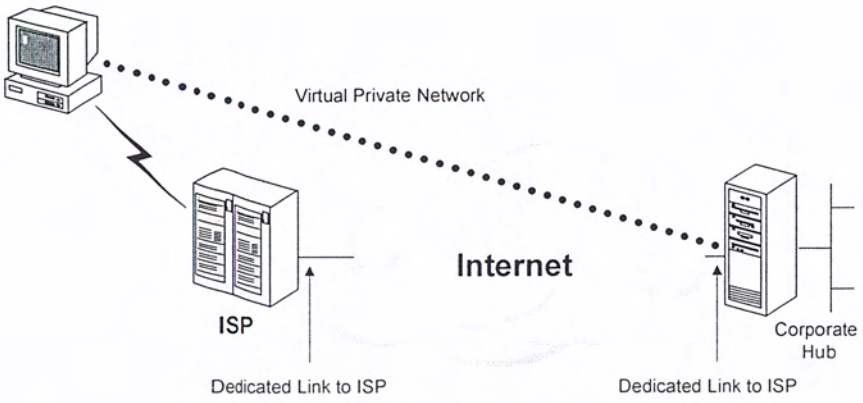
รูปที่ 7-7 แสดง VPN

VPN ทำให้ผู้ใช้ระบบที่ทำงานอยู่ที่บ้านหรือที่อื่นๆ สามารถติดต่อไปยังเซิร์ฟเวอร์ของบริษัทซึ่งอยู่ไกลกัน โดยใช้ระบบเครือข่ายสาธารณะ เช่น อินเทอร์เน็ต ในสายตาของผู้ใช้นั้น VPN จะเป็นการเชื่อมกันโดยตรงระหว่างคอมพิวเตอร์ของเขากับเซิร์ฟเวอร์ โดยการส่งข้อมูลก็จะเหมือนกับการส่งข้อมูลผ่านการเชื่อมต่อที่เป็นส่วนตัว

ลักษณะการใช้งานทั่วไปของ VPN

- รีโมตแอ็กเซสบนอินเทอร์เน็ต

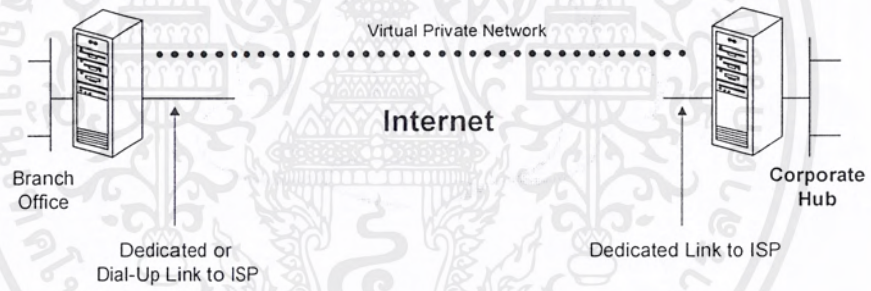
VPNs ทำให้สามารถรีโมตแอ็กเซสเข้าไปใช้ทรัพยากรของบริษัทได้โดยผ่านเครือข่ายอินเทอร์เน็ต ซึ่งข้อมูลที่ถูกส่งก็จะมีความปลอดภัยด้วย



รูปที่ 7-8 แสดง VPN ที่เชื่อมต่อรีโมตไคลเอ็นต์กับระบบภายในของบริษัท

- ติดต่อระบบเครือข่ายผ่านอินเทอร์เน็ต

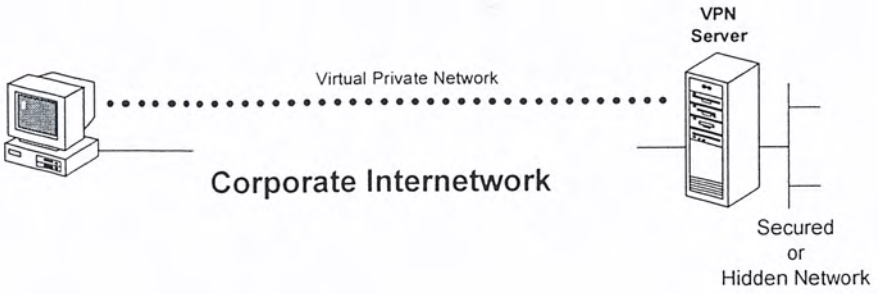
เป็นการใช้ VPN ในการเชื่อมต่อแลนของไซต์ที่อยู่ไกลเกิน



รูปที่ 7-9 แสดงการเชื่อมต่อระหว่าง 2 ไซต์

- ติดต่อคอมพิวเตอร์ผ่านอินทราเน็ต

บางครั้งในบางแผนกของบริษัท มีการเก็บข้อมูลที่เป็นความลับ ดังนั้นจึงต้องแยกเครือข่ายของแผนกนี้ออกจากแผนกอื่นๆ ที่เหลือ ถ้าแยกเครือข่ายโดยแยกทางกายภาพ จะทำให้เกิดปัญหาในการเข้าถึงข้อมูลของแผนก ดังนั้น สามารถที่จะใช้ VPN ได้ดังแสดงในรูป



รูปที่ 7-10 แสดงการเชื่อมต่อระหว่าง 2 เครื่องในแลนเดียวกัน

VPN จะทำให้ระบบเครือข่ายของแผนกยังเชื่อต่อทางด้านกายภาพกับแผนกอื่นๆ อยู่ แต่จะมีเฉพาะผู้ใช้ระบบที่ได้รับอนุญาตที่สามารถสร้าง VPN ให้สามารถเข้าถึงข้อมูลในแผนกได้

7.4 Service For Netware (SFN)

SFN เป็นโปรแกรมอรรถประโยชน์ สำหรับวินโดวส์เอ็นทีเซิร์ฟเวอร์และวินโดวส์ 2000 เซิร์ฟเวอร์ ทำให้ผู้ใช้ในระบบเครือข่ายของเน็ตแวร์สามารถกสามารถเพิ่มความสามารถให้กับระบบเครือข่ายด้วยวินโดวส์เอ็นทีเซิร์ฟเวอร์ SFN ประกอบด้วย File and Print Service for Netware (FPNW) และส่วนสำคัญของ SFN คือ Microsoft Directory Synchronization Service (MSDSS) ซึ่งเป็นตัวซิงโครไนซ์ข้อมูลต่างๆ ที่เก็บในแอ็กทีฟไดเรกทอรีและข้อมูลใน NDS เพื่อให้สามารถรวมและให้วินโดวส์กับเน็ตแวร์ทำงานร่วมกันได้

7.4.1 MSDSS

MSDSS สนับสนุนการซิงโครไนซ์ 2 ทางบน NDS และสนับสนุนการซิงโครไนซ์ทางเดียวบนเน็ตแวร์ 3.x MSDSS ยังสนับสนุนการซิงโครไนซ์รหัสผ่านและการโยกย้าย (migration) อีกด้วย

MSDSS ให้ผู้ใช้เน็ตแวร์จัดการแอ็กทีฟไดเรกทอรีโดยไม่ต้องนำมาแทนไดเรกทอรีเดิมที่มีอยู่ หรือแยกใช้เป็น 2 ไดเรกทอรี ทำให้ผู้ใช้มีความคล่องตัวในการ

- รวมการจัดการไดเรกทอรี เมื่อในระบบมีหลายไดเรกทอรี
- จัดการแอ็คเคาต์จากทั้ง 2 ไดเรกทอรี
- ใช้โปรแกรมประยุกต์ อุปกรณ์ และบริการต่างๆ ที่สนับสนุนการทำงานบนไดเรกทอรีบนวินโดวส์ 2000 เซิร์ฟเวอร์แอ็กทีฟไดเรกทอรี

MSDSS สนับสนุนเน็ตแวร์แพลตฟอร์มทั้งหมด และสนับสนุนไดเรกทอรีส่วนใหญ่ และยังสนับสนุนโพรโตคอล IPX/SPX และ TCP/IP

MSDSS ทำงานอย่างไร

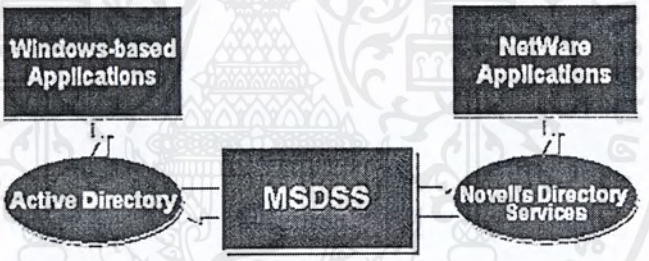
การซิงโครไนซ์คือ ความสามารถที่ทำให้อีกไดเรกทอรีหนึ่งเปลี่ยนแปลงด้วย เมื่อมีไดเรกทอรีหนึ่งเปลี่ยนแปลง ดังนั้นข้อมูลทั้ง 2 ไดเรกทอรีจะเหมือนกัน ซึ่ง MSDSS ทำระหว่างแอ็กทีฟไดเรกทอรีและ NDS

ลักษณะ 3 อย่างที่สำคัญในการซิงโครไนซ์คือ

- Session
- Object-Level Synchronize
- Directional Synchronize

Session

แอ็กทีฟไดเรกทอรีและ NDS ซิงโครไนซ์เป็นเซสชัน เซสชันเป็นการซิงโครไนซ์ที่ไม่ต่อเนื่องระหว่าง NDS หรือ Netware 3.x และแอ็กทีฟไดเรกทอรีในการซิงโครไนซ์จะเริ่มต้นเซสชันด้วยการสั่งจากผู้ดูแลระบบหรือทำเป็นระยะ ซึ่งกำหนดโดยผู้ดูแลระบบ ในแต่ละเซสชัน MSDSS จะตรวจสอบความเปลี่ยนแปลงของแต่ละไดเรกทอรี รวบรวมการเปลี่ยนแปลง และกระจายการเปลี่ยนแปลงนั้นให้กับอีกไดเรกทอรี ดังนั้นข้อมูลทั้ง 2 ไดเรกทอรีจะเหมือนกัน



รูปที่ 7-11 แสดงการทำงานของ MSDSS

จากรูปที่ 7-11 การทำงานของ MSDD เหมือนกับตัวเชื่อมระหว่างข้อมูลที่เก็บในแอ็กทีฟไดเรกทอรีและ NDS และข้อมูลสามารถไหลได้ทั้ง 2 ทางคือ จากแอ็กทีฟไดเรกทอรีไป NDS และจาก NDS ไป Active Directory

แต่ละคู่ของ Organization Unit (OU) ที่ซิงโครไนซ์ในแอ็กทีฟไดเรกทอรีและ NDS จะมีเซสชันของตนเอง MSDSS สนับสนุนการซิงโครไนซ์ 50 เซสชันในเวลาเดียวกัน องค์กรใดที่ต้องการมากกว่า 50 ครั้งในเวลาเดียวกัน จำเป็นต้องมี MSDSS เซิร์ฟเวอร์อีกตัว

เพื่อความแน่ใจว่าการเปลี่ยนแปลงของ OU ในไดเรกทอรีมีผลต่ออีก OU ในอีกไดเรกทอรี MSDSS มีแผนผังที่เชื่อมโยงออบเจ็กต์และพรอเพอตี้ (property) ของออบเจ็กต์จากไดเรกทอรีหนึ่งให้ตรงกับไดเรกทอรีหนึ่ง

ขั้นตอนในการซิงโครไนซ์

1. เซสชันซิงโครไนซ์ถูกกระตุ้นให้ทำงาน (ทั้งในตารางกำหนดการหรือ สั่งโดยผู้ดูแลระบบ)
2. MSDSS สอบถามความเปลี่ยนแปลงของทั้งแอ็กทีฟไดเรกทอรีและ NDS (จะสอบถามความเปลี่ยนแปลงบน NDS ก็ต่อเมื่อเป็นการซิงโครไนซ์แบบสองทาง)
3. MSDSS ดูความเปลี่ยนแปลงของแอตทริบิวต์ในแอ็กทีฟไดเรกทอรีและดูความเปลี่ยนแปลงของออบเจ็กต์ใน NDS
4. ความเปลี่ยนแปลงในแอ็กทีฟไดเรกทอรีจะฟอร์เวิร์ดซิงโครไนซ์ (Forward Synchronize) ไปสู่ NDS (เรียกว่า Publish Synchronization)
5. ความเปลี่ยนแปลงใน NDS จะรีเวอร์สซิงโครไนซ์ (Reverse Synchronize) ไปสู่ แอ็กทีฟไดเรกทอรี(เรียกว่า Subscriber Synchronization)
6. เมื่อจบแต่ละเซสชันจะมีการเก็บล็อกไว้เพื่อบอกรายละเอียดของความผิดพลาดและสัญญาณเตือน

Object-Level Synchronization

ในแต่ละ session จะมีการเปลี่ยนแปลงเฉพาะแต่ละออบเจ็กต์ ในไดเรกทอรีที่มีการซิงโครไนซ์ session สุดท้ายเท่านั้น การซิงโครไนซ์ในระดับออบเจ็กต์ช่วยลดภาระในการส่งข้อมูลผ่านระบบเครือข่าย มี 2 วิธี คือ

Object-Specific Synchronization

การซิงโครไนซ์จะไม่สนใจที่อยู่ของออบเจ็กต์ในโครงสร้างของทรีจึงอนุญาตให้ออบเจ็กต์เดียวกันอยู่ในที่ที่แตกต่างกันบนแต่ละไดเรกทอรีได้

ตัวอย่างเช่น บนแอ็กทีฟไดเรกทอรีผู้ใช้ระบบ ก เป็นยูสเซอร์ออบเจ็กต์ใน Accounting คอนเทนเนอร์ใน user OU แต่ผู้ใช้ระบบ ก สามารถอยู่ใน NDS โดยเป็นยูสเซอร์ออบเจ็กต์ใน Chicago คอนเทนเนอร์ใน user OU ในการซิงโครไนซ์ไม่จำเป็นต้องอยู่ใน OU เดียวกัน

การซิงโครไนซ์แบบสองทาง ผู้ดูแลระบบของ NDS ใน Chicago สามารถเปลี่ยนแปลงข้อมูลส่วนตัว เช่น เบอร์โทรศัพท์ ในขณะที่ผู้ดูแลระบบของแอ็กทีฟไดเรกทอรีใน Accounting สามารถกำหนดสิทธิได้ว่าสามารถให้ผู้ใช้ระบบ ก สามารถใช้โปรแกรมใดได้บ้าง ในขณะซิงโครไนซ์แอตทริบิวต์ของผู้ใช้ระบบ ก ที่ถูกเปลี่ยนแปลงทั้งในแอ็กทีฟไดเรกทอรีและ NDS จะส่งไปให้อีกไดเรกทอรีหนึ่ง

Location-Specific Synchronization

การซิงโครไนซ์จะดูที่โครงสร้างและออบเจ็กต์ในนั้นเป็นหลัก ถ้าออบเจ็กต์ใดอยู่ในคอนเทนเนอร์ที่เจาะจง MSDSS จะสร้างคอนเทนเนอร์ที่ตรงกันในอีกไดเรกทอรี ถ้ามีการย้ายออบเจ็กต์ Dirsync จะย้ายออบเจ็กต์ให้เหมือนกันในอีกไดเรกทอรีหนึ่ง จุดประสงค์ของการซิงโครไนซ์แบบนี้คือ เพื่อซิงโครไนซ์ออบเจ็กต์ที่เปลี่ยนแปลง และรักษาโครงสร้างของทรีที่ซิงโครไนซ์ให้เหมือนกัน

การซิงโครไนซ์แบบทางเดียว ผู้ดูแลระบบของแอ็กทีฟไดเรกทอรีได้รับสิทธิในการเปลี่ยนแปลง Attribute ของผู้ใช้ระบบ ก และการเปลี่ยนแปลงจะส่งไปยัง NDS เพื่อการดูแลระบบที่เป็นศูนย์กลาง และไดเรกทอรีทั้ง 2 จะต้องมีการแบ่งส่วนและโครงสร้างเหมือนกัน

ทิศทางในการซิงโครไนซ์ (Directional Synchronization)

MSDSS สนับสนุนการซิงโครไนซ์ 2 ชนิดคือ ซิงโครไนซ์ทางเดียว และ ซิงโครไนซ์สองทาง

ซิงโครไนซ์แบบทางเดียว (One-Way Synchronization) เป็นการซิงโครไนซ์ที่ส่งความเปลี่ยนแปลงจากแอ็กทีฟไดเรกทอรีให้กับ NDS แต่ไม่สามารถดูความเปลี่ยนแปลงบน NDS ได้ การซิงโครไนซ์จากแอ็กทีฟไดเรกทอรีไปสู่ NDS เรียกว่าฟอร์เวิร์ดซิงโครไนซ์ การฟอร์เวิร์ดซิงโครไนซ์ทำให้เป็นระดับแอตทริบิวต์มากขึ้น (อ่านออบเจ็กต์ที่เปลี่ยนแปลงบนแอ็กทีฟไดเรกทอรีและเขียนแอตทริบิวต์ที่มีการเปลี่ยนแปลงบน NDS) การซิงโครไนซ์แบบทางเดียวใช้เมื่อต้องการดูแลระบบเป็นศูนย์กลาง

ซิงโครไนซ์แบบสองทาง (Two-Way Synchronization) เป็นการซิงโครไนซ์ที่ดูการเปลี่ยนแปลงของทั้งแอ็กทีฟไดเรกทอรีและ NDS และซิงโครไนซ์การเปลี่ยนแปลงนั้นไปให้อีกไดเรกทอรี ดังนั้นจึงเป็นทั้งฟอร์เวิร์ด และรีเวอร์สซิงโครไนซ์ การรีเวอร์สซิงโครไนซ์จะทำในระบบออบเจ็กต์ (ดูออบเจ็กต์ใน NDS ทั้งหมด คัดออบเจ็กต์ที่ไม่มีการเปลี่ยนแปลงออกไป และเขียนเฉพาะออบเจ็กต์ที่เปลี่ยนแปลงลงแอ็กทีฟไดเรกทอรี) การซิงโครไนซ์แบบสองทางใช้กับระบบที่ต้องมีการจัดการกับทั้ง 2 ไดเรกทอรี โดยแยกกัน และข้อมูลที่เปลี่ยนไปที่ใดที่หนึ่งแล้วจะมีผลกับอีกทีหนึ่งด้วย

7.5 Service For UNIX (SFU)

ใน SFU จะประกอบด้วยการซิงโครไนซ์รหัสผ่าน ซึ่งอนุญาตให้มีการซิงโครไนซ์รหัสผ่านของยูสเซอร์บนวินโดวส์กับรหัสผ่าน ของยูสเซอร์บนยูนิกซ์ได้

ในหลายๆ บริษัทจะมีเครือข่ายคอมพิวเตอร์ที่ประกอบด้วยระบบที่แตกต่างกัน ซึ่งมีทั้งวินโดวส์และยูนิกซ์โดยที่แต่ละระบบจะถูกจัดการ โดยไม่เกี่ยวข้องกัน และถูกดูแลโดยใช้วิธีการเข้าถึงที่แตกต่างกัน นอกจากนี้สิ่งที่สำคัญอีกอย่างหนึ่ง คือ ทั้ง 2 ระบบนี้ใช้กลไกในการพิสูจน์ตนของยูสเซอร์ต่างกัน ผลที่ตามมาคือยูสเซอร์ต้องมียูสเซอร์แอ็คเคาต์ และรหัสผ่าน ที่แตกต่างกันของแต่ละระบบ

การต้องเก็บรักษา 2 รหัสผ่าน ทำให้เกิดความยุ่งยากแก่ยูสเซอร์และเกิดความยุ่งยากแก่ผู้ดูแลระบบอีกด้วย SFU จึงแก้ปัญหาเหล่านี้โดยการซิงโครไนซ์รหัสผ่าน ซึ่งจะทำให้ยูสเซอร์ใช้รหัสผ่านเดียวกันทั้งบนระบบยูนิกซ์และระบบวินโดวส์

7.5.1 เป้าหมายของการ Synchronize Password

1. รหัสผ่านของยูสเซอร์บนวินโดวส์และบนยูนิกซ์สามารถซิงโครไนซ์กันได้ทำให้ ยูสเซอร์จำเพียงแค่รหัสผ่านเดียวแล้วสามารถใช้ได้ทั้งวินโดวส์และยูนิกซ์

2. ผู้ดูแลระบบสามารถดูแลการเปลี่ยนแปลงของ รหัสผ่าน ได้ง่ายขึ้น
3. กลไกในการซิงโครไนซ์นั้นต้องมีความปลอดภัย

หมายเหตุ การซิงโครไนซ์รหัสผ่านไม่ได้ทำให้เกิดสิ่งต่อไปนี้

1. SSO ระหว่างยูนิกซ์และวินโดวส์ซึ่งยูสเซอร์ยังคงต้องล็อกอินเข้าสู่ระบบก่อนที่จะเข้าถึงมัน
2. Synchronize Application Password การซิงโครไนซ์รหัสผ่านจะซิงโครไนซ์เฉพาะรหัสผ่าน ที่ใช้ล็อกอินเข้าสู่ระบบเท่านั้น แต่จะไม่ซิงโครไนซ์รหัสผ่านของโปรแกรมประยุกต์อื่นๆ

7.5.2 ข้อจำกัดของการซิงโครไนซ์รหัสผ่านโดยใช้ SFU

จะสามารถซิงโครไนซ์รหัสผ่านได้ในกรณีที่ยูสเซอร์เนม ของผู้ใช้บนยูนิกซ์และวินโดวส์เหมือนกันเท่านั้น ถ้ายูสเซอร์เนมของผู้ใช้ไม่เหมือนกันจะไม่สามารถใช้ได้

โครงสร้างการซิงโครไนซ์รหัสผ่าน

การซิงโครไนซ์รหัสผ่านประกอบด้วยองค์ประกอบ 2 ส่วนคือ

1. การซิงโครไนซ์รหัสผ่านจากวินโดวส์ไปยูนิกซ์
2. การซิงโครไนซ์รหัสผ่านจากยูนิกซ์ไปวินโดวส์

ในกรณีที่ใช้การซิงโครไนซ์รหัสผ่านจากยูนิกซ์ไปวินโดวส์จะทำให้ใช้ซิงโครไนซ์รหัสผ่านจากยูนิกซ์ไปวินโดวส์จะทำให้ใช้ซิงโครไนซ์รหัสผ่านจากวินโดวส์ไปยูนิกซ์ได้ด้วย

เมื่อยูสเซอร์เปลี่ยนรหัสผ่านบนเครื่องคอมพิวเตอร์ Password Synchronize Software ที่อยู่ที่ไคลเอ็นต์จะเก็บรหัสผ่านใหม่ไว้ จากนั้นก็ส่งคำร้องขอเปลี่ยนรหัสผ่านพร้อมกับรหัสผ่านใหม่ไปให้ Password Synchronize Software ที่เซิร์ฟเวอร์ในการส่งข้อมูลนี้จะใช้ TCP/IP Socket ในการส่ง และใช้ Triple DES ในการเข้ารหัส และถอดรหัสข้อมูล

7.5.3 การซิงโครไนซ์รหัสผ่านจากวินโดวส์ไปยูนิกซ์

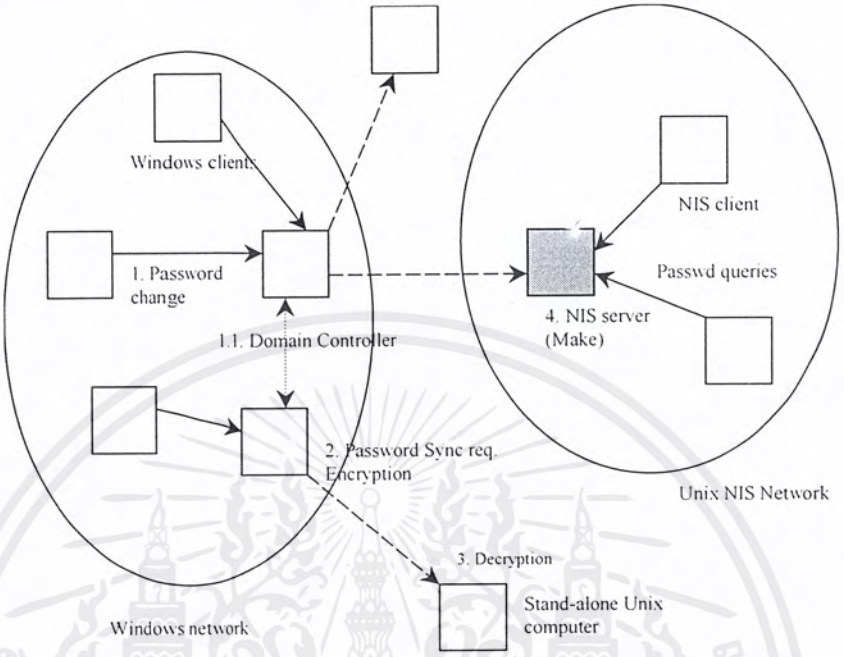
Password Synchronize for Domain User

ในการซิงโครไนซ์รหัสผ่านของยูสเซอร์ที่อยู่ในโดเมนจะต้องลง Password Synchronize Component ที่ทุกๆโดเมนคอนโทรลเลอร์ (PCS) ซึ่งอยู่ภายในโดเมนเนื่องจากเมื่อยูสเซอร์ทำการเปลี่ยน รหัสผ่าน รหัสผ่านของยูสเซอร์จะต้องถูกเปลี่ยนในทุกๆโดเมนคอนโทรลเลอร์

ขั้นตอนในการ Setup การซิงโครไนซ์รหัสผ่านจากวินโดวส์ไปยูนิกซ์

1. Install SSOD (Password Change Service) บนเครื่องที่เป็นยูนิกซ์ทุกเครื่องที่ต้องการให้เกิดการซิงโครไนซ์รหัสผ่านหรือถ้าต้องการให้ซิงโครไนซ์กับ NIS หรือ NIS+ ก็ต้อง Install SSOD ที่ NIS หรือ NIS+ Master Server
2. คอนฟิกูเรชันไฟล์ sso.conf ในเครื่องที่เป็นยูนิกซ์

3. Install Password Synchronize ที่ทุกๆ เครื่องที่เป็นวินโดวส์



รูป 7-12 แสดงการซิงโครไนซ์รหัสผ่านจากวินโดวส์โดเมนไปยังยูนิกซ์

ลำดับเหตุการณ์ที่เกิดขึ้นเมื่อมีการเปลี่ยนรหัสผ่าน

1. ยูสเซอร์เปลี่ยนรหัสผ่านที่เครื่องไคลเอ็นต์ที่ใช้ระบบปฏิบัติการวินโดวส์ซึ่งอยู่ภายในโดเมน จากนั้นไคลเอ็นต์ก็จะส่งการร้องขอเปลี่ยน รหัสผ่านไปยังโดเมนคอนโทรลเลอร์
2. เมื่อโดเมนคอนโทรลเลอร์ได้รับการร้องขอ ก็จะทำการเข้ารหัส รหัสผ่านที่ยูสเซอร์เปลี่ยนจากนั้นก็จะส่งไปให้เครื่องที่มีระบบปฏิบัติการเป็นยูนิกซ์ที่เกี่ยวข้องกับการซิงโครไนซ์รหัสผ่าน
3. SSOD ที่ทำงานอยู่บนเครื่องที่เป็นยูนิกซ์เมื่อได้รับการร้องขอการเปลี่ยน รหัสผ่านก็จะถอดรหัสข้อความที่ได้รับมา และจากนั้นก็เปลี่ยน รหัสผ่านของยูสเซอร์บนเครื่องที่เป็นยูนิกซ์
4. ถ้าเครื่องที่เป็นยูนิกซ์นั้นเป็น NIS เซิร์ฟเวอร์ที่ถูกคอนฟิกูเรชันแล้วมันจะเปลี่ยนรหัสผ่านNIS ของยูสเซอร์ทำให้ NISไคลเอ็นต์ที่ติดต่อกับ NIS เซิร์ฟเวอร์ได้รับ รหัสผ่านใหม่ของยูสเซอร์

การซิงโครไนซ์รหัสผ่านสำหรับผู้ใช้ระบบแบบโลคอล

มีหลักการทำงานคล้ายกับการซิงโครไนซ์สำหรับโดเมนยูสเซอร์ต่างกันตรงที่จะติดตั้ง Password Synchronize Component ที่เครื่องคอมพิวเตอร์ที่ต้องการให้เกิดการซิงโครไนซ์รหัสผ่านแทนที่การติดตั้งที่โดเมนคอนโทรลเลอร์

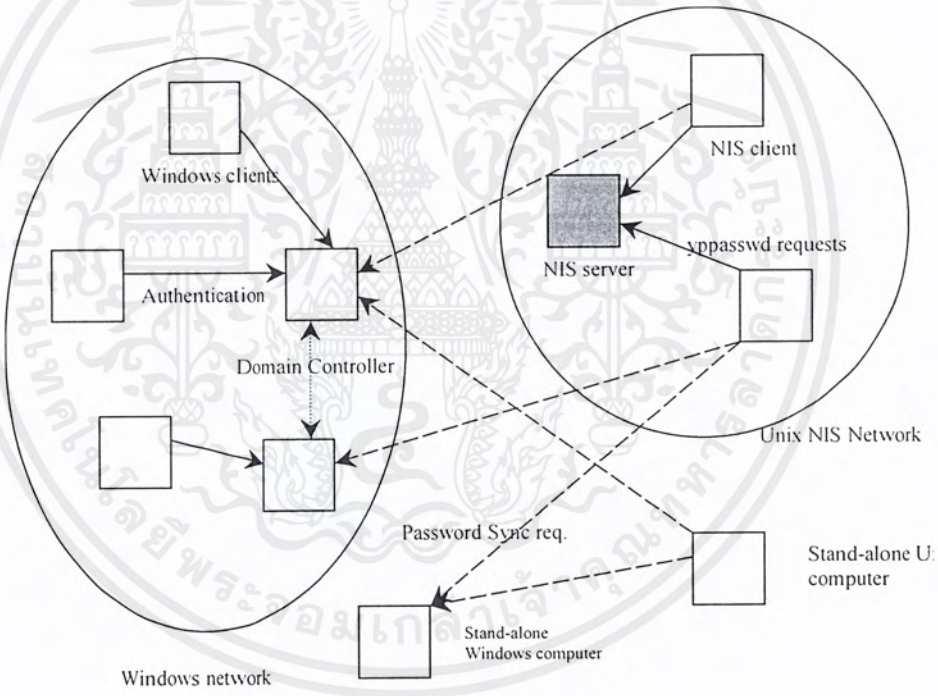
ส่วนทางด้านคอมพิวเตอร์ที่เป็นยูนิกซ์จะมีหลักการเหมือนการซิงโครไนซ์รหัสผ่านของโดเมนยูสเซอร์

7.5.4 การ ซิงโครไนซ์ รหัสผ่าน จากยูนิกซ์ไปวินโดวส์

การคอนฟิกูเรชันให้เป็นการซิงโครไนซ์รหัสผ่านจากยูนิกซ์ไปวินโดวส์จะทำให้สามารถ ซิงโครไนซ์รหัสผ่านจากวินโดวส์เป็นยูนิกซ์ได้ด้วย

ขั้นตอนในการ Setup การซิงโครไนซ์รหัสผ่านจากยูนิกซ์ไปวินโดวส์

1. Install Password Synchronization Pluggable Module (pam_sso.so) บนเครื่องที่เป็นยูนิกซ์ทุกเครื่องที่ยูสเซอร์ใช้ในการเปลี่ยนรหัสผ่าน
2. Install Password Synchronize ที่ทุกๆ เครื่องที่เป็นวินโดวส์



รูป 7-13 แสดงการซิงโครไนซ์รหัสผ่านจากวินโดวส์ไปยูนิกซ์

ลำดับเหตุการณ์ที่เกิดขึ้นเมื่อมีการเปลี่ยนรหัสผ่าน

1. ยูสเซอร์เปลี่ยนรหัสผ่านที่เครื่องไคลเอ็นต์ที่ใช้ระบบปฏิบัติการยูนิกซ์โดยใช้คำสั่ง passwd
2. pam_sso.so จะเข้ารหัส รหัสผ่านแล้วส่งไปให้กับคอมพิวเตอร์ที่เป็นวินโดวส์ที่เกี่ยวข้อง

3. Password Synchronization Service ที่ทำงานอยู่บนวินโดวส์จะถอดรหัส รหัสผ่านที่ได้รับและทำการเปลี่ยนรหัสผ่านของยูสเซอร์

การซิงโครไนซ์รหัสผ่าน สามารถทำงานได้บนแพลตฟอร์มดังแสดงในตารางต่อไปนี้

	Windows to UNIX Synchronization Module (SSOD)	UNIX to Windows Synchronization Module (pam_sso.so)
Solaris 2.6 and above	✓	✓
HP-UX 10.3 and above	✓	✓
IBM AIX 4.2 and above	✓	✗
Digital Tru64	✓	✓
Linux (Redhat 5.2 and above)	✓	✓

ตารางที่ 7-1 แสดงแพลตฟอร์มที่ SFU สามารถทำงานได้

บทที่ 8
Windows NT Server

8.1 Microsoft Windows NT Service for UNIX

Microsoft Windows NT Service For UNIX ช่วยให้สามารถรวมระบบปฏิบัติการที่เป็นวินโดวส์เอ็นทีเวิร์กสเตชัน 4.0 และวินโดวส์เอ็นทีเซิร์ฟเวอร์ 4.0 เข้ากับระบบที่มีระบบปฏิบัติการที่เป็นยูนิกซ์ที่มีอยู่แล้วภายในองค์กรได้อย่างง่ายขึ้น

8.1.1 ลักษณะเด่นของ SFU

ลักษณะเด่นของ SFU สามารถแบ่งได้เป็น 3 กลุ่มคือ

1. บริการเกี่ยวกับไฟล์ (file service) NTSFU ประกอบด้วยส่วนสนับสนุน Network File System (NFS) ไคลเอ็นต์ และ NFS เซิร์ฟเวอร์ ซึ่งสามารถทำให้ใช้ไฟล์ที่เป็นทรัพยากรของยูนิกซ์ และยังทำให้ยูนิกซ์ไคลเอ็นต์สามารถใช้ไฟล์ที่เป็นทรัพยากรของวินโดวส์เอ็นทีได้อีกด้วย
2. บริการเกี่ยวกับการเชื่อต่อกัน (Connectivity Service) ประกอบด้วยบริการ Telnet เซิร์ฟเวอร์ และ text-based Telnet ไคลเอ็นต์
3. Usability Service ประกอบด้วย UNIX Utilities และ Learn Shell ซึ่งทำให้ผู้ใช้และผู้ดูแลระบบที่มีความคุ้นเคยกับการใช้ระบบ UNIX สามารถใช้งานได้ง่าย

นอกจากนี้ยังมีการซิงโครไนซ์รหัสผ่านอีกด้วย ซึ่งจะกล่าวถึงเฉพาะการซิงโครไนซ์รหัสผ่านเท่านั้น

8.1.2 การซิงโครไนซ์รหัสผ่าน

การซิงโครไนซ์รหัสผ่านใน NT SFU นี้เป็นการซิงโครไนซ์แบบทางเดียว คือ อนุญาตให้จัดการรหัสผ่านของระบบที่เป็นยูนิกซ์และระบบที่เป็นวินโดวส์ผ่านวินโดวส์เอ็นทีเท่านั้น (หมายความว่าเมื่อผู้ใช้เปลี่ยนรหัสผ่านบนระบบที่เป็นวินโดวส์เอ็นที รหัสผ่านของผู้ใช้ที่ระบบยูนิกซ์จะเปลี่ยนไปด้วย แต่ถ้าผู้ใช้เปลี่ยนรหัสผ่านที่ระบบเป็นยูนิกซ์ รหัสผ่านของผู้ใช้ที่ระบบวินโดวส์เอ็นทีจะไม่เปลี่ยนไปด้วย) SFU อยู่รวมมากับ Password Synchronization Daemons (SSOD) ที่คอมไพล์มาแล้วสำหรับระบบปฏิบัติการที่เป็น HP-UX Sun-OS และ Digital UNIX หรืออาจเป็นซอร์สโค้ดให้มาคอมไพล์เองสำหรับระบบปฏิบัติการอื่นๆ

การซิงโครไนซ์รหัสผ่านมี 2 แบบคือ

1. การซิงโครไนซ์รหัสผ่านแบบปลอดภัยโดยใช้ Secure Daemon (SSOD) ซึ่งรับรหัสผ่านที่ถูกเข้ารหัสจากวินโดวส์เอ็นทีและถอดรหัสรหัสผ่านนั้น

2. การซิงโครไนซ์รหัสผ่านแบบไม่ปลอดภัย โดยใช้ rlogin ส่งรหัสผ่านโดยไม่มีการเข้ารหัส ถึงแม้การใช้วิธีนี้จะไม่มีความปลอดภัยมากนัก แต่ว่าวิธีนี้ง่ายต่อการติดตั้ง วิธีนี้เหมาะกับระบบเครือข่ายที่มีขนาดเล็ก และถูกป้องกันจากเครือข่ายภายนอกที่เพียงพอ

Creating and Configuring a UNIX Pod

ยูนิกซ์โฮสต์จะถูกจัดรวมกันไว้เป็น pod การซิงโครไนซ์รหัสผ่านจะจัดการต่อ pod pod หนึ่งเท่านั้น เราสามารถสร้าง pod ใหม่ เพิ่มยูนิกซ์โฮสต์ลงใน pod หรือเปลี่ยนวิธีซิงโครไนซ์ของ pod นั้นได้โดยใช้ Password Synchronize Administrator (PSADMIN.EXE)

8.1.3 การ Synchronize Password โดยใช้ rlogin

โดยส่วนมาก ระบบที่เป็นยูนิกซ์มักจะไม่อนุญาตให้ใช้ rlogin เนื่องจากไม่ค่อยมีความปลอดภัย แต่ถ้าระบบเครือข่ายนั้นแยกเป็นอิสระจากเครือข่ายภายนอก เช่น ใช้ไฟร์วอลล์ และระบบเครือข่ายนั้นเป็นระบบเครือข่ายที่มีขนาดเล็ก การใช้ rlogin ในการซิงโครไนซ์รหัสผ่านจะมีความเหมาะสมมาก

ก่อนที่จะใช้ rlogin ในการซิงโครไนซ์รหัสผ่าน จะต้องทำให้รีโมตยูนิกซ์โฮสต์นั้นสนับสนุน rlogin ก่อน ซึ่งทำได้โดยคอนฟิกไฟล์ remote.rhosts ที่อยู่บนเซิร์ฟเวอร์ ไฟล์ rhosts จะต้องอนุญาตในสิ่งต่างๆ เช่น อนุญาตให้ผู้ดูแลระบบสามารถเขียนไฟล์ได้เท่านั้น ผู้ดูแลระบบต้องเป็นเจ้าของไฟล์นี้ และไฟล์นี้จะต้องอยู่ในโฮมไดเรกทอรีของรูต นอกจากนี้ในไฟล์นี้จะต้องประกอบด้วยชื่อเครื่องที่รหัสผ่านจะต้องเปลี่ยน (โดยส่วนมากจะเป็นไพรมารีโดเมนคอนโทรลเลอร์)

ข้อควรระวังข้อหนึ่งคือ แอ็กเคาต์บนยูนิกซ์เป็นเคสเซนซิทีฟ เราต้องสร้างแอ็กเคาต์บนวินโดวส์เอ็นทีให้เหมือนกับยูนิกซ์แอ็กเคาต์ (โดยปกติใช้เฉพาะตัวอักษาตัวเล็กเท่านั้น) การที่ยูสเซอร์แอ็กเคาต์ของวินโดวส์เอ็นทีกับยูนิกซ์ไม่เหมือนกันนั้นจะทำให้การซิงโครไนซ์รหัสผ่านล้มเหลว

8.1.4 การซิงโครไนซ์รหัสผ่านโดยใช้การซิงโครไนซ์รหัสผ่านแบบปลอดภัย

SFU สามารถซิงโครไนซ์รหัสผ่านได้โดยใช้การซิงโครไนซ์รหัสผ่านแบบปลอดภัย โดย daemons นี้จะทำงานอยู่บนยูนิกซ์เซิร์ฟเวอร์เพื่อรับรหัสผ่านที่ถูกเข้ารหัสที่ส่งมาจากวินโดวส์เอ็นที จากนั้นก็จะถอดรหัสรหัสผ่านที่ได้รับ แล้วนำรหัสผ่านที่ได้รับนี้ไปเปลี่ยนแอ็กเคาต์ที่เหมือนกันบนแอ็กเคาต์บนวินโดวส์เอ็นที

การติดตั้ง Password Synchronize Deamon บนยูนิกซ์

ถ้าได้ติดตั้ง NFS Client บนเครื่องที่เป็นวินโดวส์เอ็นทีแล้วให้เอ็กพอร์ทไฟล์ซิสเต็มจากเครื่องที่เป็นยูนิกซ์ แล้ว mount ไฟล์ซิสเต็มนี้ลงบนเครื่องที่เป็นวินโดวส์เอ็นที จากนั้นใช้ explorer ทำสำเนาไฟล์ SSOD และไฟล์ SSOD.confige ไปที่เครื่องที่เป็นยูนิกซ์ โดยปกติจะทำสำเนาไว้ที่ /usr/local/etc แต่อาจเปลี่ยนแปลงได้ แล้วแต่ระบบ

เมื่อทำสำเนาไปแล้ว ให้ติดตั้งไฟล์นี้ลงบนเครื่องที่เป็นยูนิกซ์ จากนั้นต้องแก้ไขไฟล์ SSOD.config โดยไฟล์นี้จะสนับสนุนกลไกการพิสูจน์ตนแบบ NIS และ /etc/passwd และถ้าเราเลือกใช้ /etc/passwd เราจะสามารถที่จะใช้ /etc/shadow ได้ เราสามารถ start deamon ได้โดยทำเองหรือสามารถใช้ startup file ก็ได้

ข้อสังเกต ทุกโฮสต์ในยูนิกซ์ pod ต้องมีคีย์ที่ใช้ในการถอดรหัสคีย์เดียวกัน แต่ถ้าเป็น pod ต่างกัน เราสามารถใช้คีย์ต่างกันได้

8.2 DSMN (Directory Service Manager for NetWare)

ในขณะที่ระบบเครือข่ายที่เป็นเน็ตแวร์ขยายขึ้น ผู้ใช้เน็ตแวร์ 2.x 3.x และ 4.x ที่เป็นผู้ดูแลระบบโดยใช้ NetWare Bidery จะมีภาระหนักมาก DSMN ช่วยทำให้การจัดการระบบง่ายขึ้น DSMN ช่วยจัดการระบบที่มีหลายแพลตฟอร์ม ช่วยให้ยูสเซอร์ล็อกอินเพียงครั้งเดียวเข้าถึงได้ทุกระบบ และช่วยให้ผู้ดูแลระบบมีการจัดการที่ศูนย์กลางเพียงจุดเดียว

ในระบบที่เป็นเน็ตแวร์หลายเซิร์ฟเวอร์ ยูสเซอร์ต้องจำชื่อเซิร์ฟเวอร์ที่ตนต้องล็อกอิน แอ็กเคาต์และรหัสผ่านของแต่ละเซิร์ฟเวอร์ ในการดูและยูสเซอร์แอ็กเคาต์และรหัสผ่าน ผู้ดูแลระบบต้องดูแลแต่ละ bindery ของเซิร์ฟเวอร์ (ฐานข้อมูลเก็บแอ็กเคาต์ของโนเวลล์) DSMN ช่วยทำสำเนาเน็ตแวร์แอ็กเคาต์ไปยังวินโดวส์เอ็นทีไดเรกทอรีเซอร์วิซ และกระจายความเปลี่ยนแปลงไปยังเน็ตแวร์เซิร์ฟเวอร์โดยไม่ต้องใช้โปรแกรมใดๆ บนเน็ตแวร์เซิร์ฟเวอร์

นอกจากนี้ DSMN สามารถซิงโครไนซ์แอ็กเคาต์เน็ตแวร์เซิร์ฟเวอร์ทั้งหมด เพื่อรองรับการล็อกอินครั้งเดียวในระบบเครือข่าย ยูสเซอร์จะใช้แอ็กเคาต์เดียว เพื่อเข้าถึงเน็ตแวร์และวินโดวส์เอ็นที DSMN สามารถทำสำเนายูสเซอร์แอ็กเคาต์ของเน็ตแวร์ไปยังวินโดวส์เอ็นทีเซิร์ฟเวอร์ไดเรกทอรี ดังนั้นผู้ดูแลระบบสามารถจัดการแอ็กเคาต์ทั้งหมดได้

DSMN ช่วยในการดูแลระบบให้ง่ายโดย

- จัดการยูสเซอร์แอ็กเคาต์ในวินโดวส์เอ็นทีและเน็ตแวร์อย่างเป็นศูนย์กลาง โดยใช้วินโดวส์เอ็นทีไดเรกทอรีเซอร์วิซ เมื่อมีการเปลี่ยนแปลง จะส่งผลกลับไปให้เน็ตแวร์เซิร์ฟเวอร์
- เป็นกราฟิก ทำให้ใช้งานง่าย
- มีทางเลือกในการตั้งค่าเริ่มต้นของรหัสผ่านมาก
- สามารถจัดการไดเรกทอรีเซอร์วิซจากที่ใดก็ได้ในระบบเครือข่าย รวมทั้งการไดแอลอัพ
- แบ็กอัพฐานข้อมูลของแอ็กเคาต์ไปยังทุกที่ในระบบเครือข่าย
- สามารถกลับไปเป็นแบบก่อนเกิดความเสียหายได้อย่างรวดเร็วและง่ายดาย เมื่อใช้วินโดวส์เอ็นทีเซิร์ฟเวอร์ตัวอื่นแบ็กอัพข้อมูล

DSMN ช่วยให้ผู้ใช้ระบบเข้าถึงระบบได้ง่ายโดย

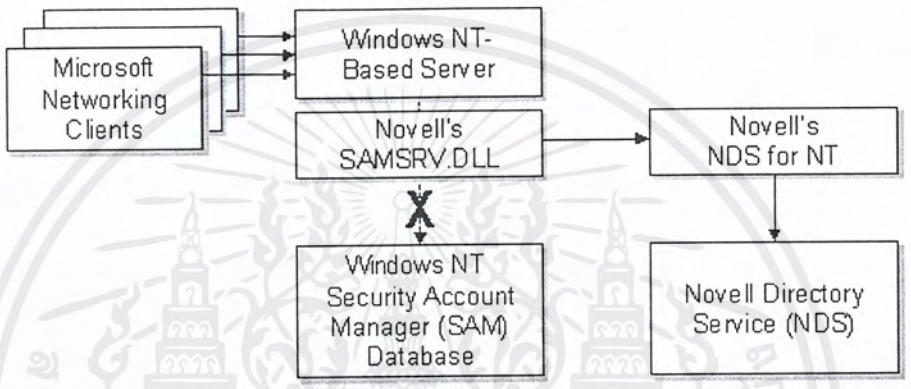
- ต้องการเพียง 1 ยูสเซอร์เนมและรหัสผ่านในวินโดวส์เอ็นทีไดเรกทอรีเซอร์วิซ ยูสเซอร์แอ็กเคาต์จะกระจายไปยังเน็ตแวร์เซิร์ฟเวอร์มีสิทธิ์เข้าถึง
- จัดการเกี่ยวกับความเหมือนกันของยูสเซอร์เนมและรหัสผ่านบนวินโดวส์เอ็นทีเซิร์ฟเวอร์และเน็ตแวร์ ดังนั้นยูสเซอร์มีเพียงยูสเซอร์เนมและรหัสผ่านเดียว ไม่ว่าจะล็อกอินที่ใด
- สามารถให้ผู้ใช้ล็อกอินโดยใช้ RAS ของวินโดวส์เอ็นที ยูสเซอร์สามารถล็อกอินโดยการไดแอลอัพโดยใชยูสเซอร์เนมและรหัสผ่านเดียวกัน (เครื่องที่ติดต่อโดยการใช้ RAS นั้น ต้องได้รับอนุญาตจาก RAS ด้วย)
- จัดการเกี่ยวกับการเข้าถึงโปรแกรมประยุกต์ต่างๆ ซึ่งรันบนวินโดวส์เอ็นทีเซิร์ฟเวอร์ของยูสเซอร์ ยูสเซอร์สามารถเข้าถึงโปรแกรมประยุกต์ต่างๆ และบริการเกี่ยวกับไฟล์และการปริ๊นต์บนวินโดวส์เอ็นทีได้โดยการล็อกอินเพียงครั้งเดียว
- ยูสเซอร์สามารถเปลี่ยนรหัสผ่านโดยใช้ CHGPASS และการเปลี่ยนแปลงนั้นจะส่งผลไปยังเน็ตแวร์และวินโดวส์เอ็นทีเซิร์ฟเวอร์ทั้งหมด

บทที่ 9

Novell NetWare

9.1 NDS for NT

9.1.1 การทำงานของ NDS for NT



รูปที่ 9-1 แสดงการทำงานของ NDS for NT

จากรูปที่ 9-1 ในขณะที่อยู่ในสภาพแวดล้อมที่เป็นวินโดวส์เอ็นทีเซิร์ฟเวอร์ ฐานข้อมูล Security Account Manager (SAM) จะเก็บข้อมูลเกี่ยวกับผู้ใช้ระบบและสิทธิของผู้นั้น และวินโดวส์เอ็นทีเซิร์ฟเวอร์จะตรวจสอบข้อมูลโดยการใช้ Dynamic Link Library (DLL) ชื่อ SAMSRV.DLL เมื่อผู้ดูแลระบบติดตั้ง NDS for NT ในวินโดวส์เอ็นทีเซิร์ฟเวอร์ การติดตั้งจะนำ SAMSRV.DLL ของโนเวลล์แทนที่ของไมโครซอฟท์ ซึ่งช่วยให้เกิดการย้ายไปร้องขอที่ NDS เซิร์ฟเวอร์แทนที่ฐานข้อมูล SAM หลังจากนั้น NDS เซิร์ฟเวอร์สามารถตัดสินใจในด้านการพิสูจน์ตนและการพิสูจน์สิทธิ์ได้โดยใช้ข้อมูลบนฐานข้อมูลของ NDS

9.1.2 การพิสูจน์ตนของผู้ใช้ระบบ และ SSO

NDS for NT สามารถรวมวินโดวส์เอ็นทีและ NDS ให้เป็นระบบเดียวกัน โดยการจัดการผู้ใช้ระบบด้วย SSO ซึ่งหมายถึง เมื่อผู้ใช้ระบบต้องการเข้าถึงทรัพยากรในระบบของวินโดวส์เอ็นทีหรือ NDS ผู้ใช้ระบบจะใส่รหัสผ่านเพียงครั้งเดียว

รหัสผ่านของผู้ใช้ระบบ

วินโดวส์เอ็นทีใช้ MD4 เป็นวิธีในการเข้ารหัสรหัสผ่าน ในขณะที่ NDS ใช้ RSA ซึ่งมีความปลอดภัยมากกว่า เมื่อสร้างรหัสผ่านของผู้ใช้ระบบแล้ว รหัสผ่านนั้นจะเข้ารหัสแล้วเก็บไว้บนเซิร์ฟเวอร์ เมื่อผู้ใช้ระบบล็อกอิน รหัสผ่านจะถูกเข้ารหัสด้วย RSA ที่เวิร์กสเตชัน และส่งไปให้ NDS ตรวจสอบความถูกต้อง ถ้าตรงกับที่เก็บใน NDS ก็จะอนุญาตให้เข้าใช้ระบบ NDS ได้ ในขณะเดียวกัน รหัสผ่านที่เข้ารหัสด้วย MD4 และส่งไปให้วินโดวส์เอ็นทีเซิร์ฟเวอร์ ถ้าตรวจสอบข้อมูลที่อยู่ในยูสเซอร์ออบเจ็กต์ของโดเมน ก็จะอนุญาตให้เข้าใช้ระบบของวินโดวส์เอ็นทีได้

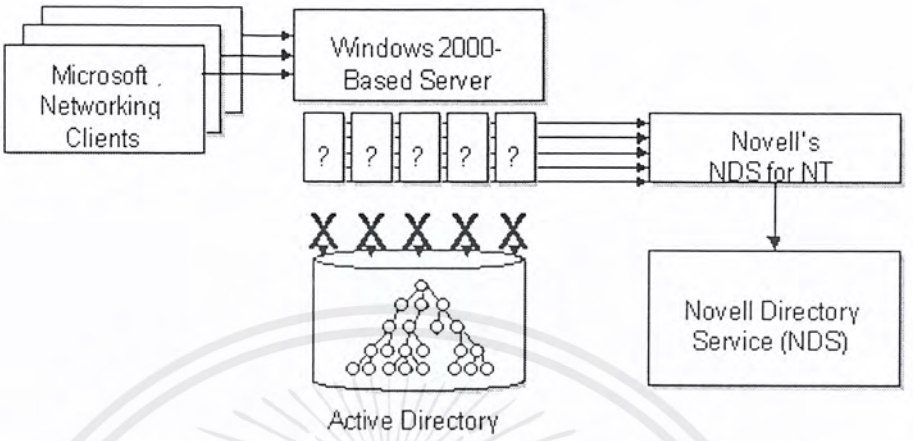
เมื่อโดเมนถูกย้ายไปรวมกัน NDS ดังนั้น รหัสผ่านที่ถูกเข้ารหัสก็จะถูกย้ายไปรวมด้วย เป็นการอนุญาตให้ผู้ใช้ระบบได้ใช้รหัสผ่านเดิมกับเซิร์ฟเวอร์ที่ถูกย้าย NDS จะเก็บรหัสผ่านทั้งที่เป็น MD4 และ RSA การล็อกอินไปยัง NDS และวินโดวส์เอ็นทีเซิร์ฟเวอร์จะไม่เกิดการเปลี่ยนแปลง เนื่องจากไม่มีการเปลี่ยนแปลงที่เวิร์กสเตชัน เมื่อล็อกอิน รหัสผ่านที่เข้ารหัสด้วย RSA จะส่งไปยัง NDS และรหัสผ่านที่เข้ารหัสด้วย MD4 จะส่งไปยังวินโดวส์เอ็นทีเซิร์ฟเวอร์

การซิงโครไนซ์รหัสผ่าน

เนื่องจากโนเวลล์ดูแลความปลอดภัยของไมโครซอฟท์ จึงจำเป็นต้องให้รหัสผ่านใน NDS และวินโดวส์เอ็นทีซิงโครไนซ์กัน ทางที่ดีที่สุดที่จะดูแลเรื่องซิงโครไนซ์รหัสผ่าน จะเป็นด้านการเซตอัพที่ไคลเอ็นต์เวิร์กสเตชัน เพื่อพิสูจน์ตนต่อ NDS และวินโดวส์เอ็นทีในเวลาเดียวกัน วิธีที่ช่วยในการจัดการการซิงโครไนซ์รหัสมีดังนี้

1. ใช้ intraNetWare ไคลเอ็นต์กับวินโดวส์เอ็นทีและตรวจสอบการซิงโครไนซ์รหัสผ่าน โดยดูที่การล็อกอิน โดยให้ซิงโครไนซ์ถ้ารหัสผ่านที่ใช้ปัจจุบันยังไม่ได้ซิงโครไนซ์ โดย intraNetWare Client เปลี่ยนรหัสผ่านโดยใช้ NWGINA
2. ใช้ NWAdmin โดยเลือกให้ snap-in ซึ่งทำให้เกิดการเปลี่ยนแปลงทั้ง NDS และวินโดวส์เอ็นที โดยใช้ NWAdmin แทน UserManager NETADMIN และ SETPASS
3. กำหนดอายุของรหัสผ่านใน NDS และไม่ต้องกำหนดในวินโดวส์เอ็นที เพื่อให้แน่ใจว่ารหัสผ่านในวินโดวส์เอ็นทีจะไม่เปลี่ยนแปลง นอกจากการซิงโครไนซ์มาจาก NDS

9.1.3 การใช้ NDS for NT บนวินโดวส์2000 และ Active Directory



รูปที่ 9-2 แสดงการทำงานของ NDS for NT บนวินโดวส์ 2000 ที่ใช้แอ็กทีฟไดเรกทอรี

ในวินโดวส์ 2000 จะใช้แอ็กทีฟไดเรกทอรีแทน SAM ซึ่งทำให้มีความปลอดภัยมากกว่าวินโดวส์เอ็นที 4.0 ดังนั้น SAMSRV.DLL จึงเป็นเพียงส่วนหนึ่งในระบบรักษาความปลอดภัย และรุ่นของ SAMSRV.DLL ในวินโดวส์ 2000 ต่างจากวินโดวส์เอ็นทีเซิร์ฟเวอร์ 4.0

ในการนำ NDS มาใช้ในโครงสร้างของวินโดวส์ 2000 โนเวลล์ต้องแทนที่ส่วนรักษาความปลอดภัยและส่วนการทำงานของไดเรกทอรีที่แอ็กทีฟไดเรกทอรีให้บริการ ซึ่งมีการทำงานใกล้เคียงกับ NDS ซึ่ง Active Directory มีการทำงานที่เฉพาะมากกว่า SAM ในวินโดวส์เอ็นทีเซิร์ฟเวอร์ 4.0 เช่นมีการใช้ Kerberos สนับสนุน ซึ่งใน NDS ไม่มี ดังนั้น NDS for NT จึงไม่เหมาะกับวินโดวส์ 2000

9.2 Novell Account Management for Windows 2000 (AM)

Novell AM for Windows 2000 เป็นการรวมยูสเซอร์ กรุ๊ป OU ของแอ็กทีฟไดเรกทอรีและ NDS ด้วยการซิงโครไนซ์โดยไม่ได้ใช้วิธีรีไดเร็ก โปรแกรมนี้ประกอบด้วย 6 ส่วนคือ ส่วนที่สนใจคือ Password Synchronization Service และ Password Filter

ในการนำไปใช้กับวินโดวส์ 2000 สิ่งที่ต้องเข้าใจคือ

- จะจัดการเฉพาะยูสเซอร์ กรุ๊ป และ OU เท่านั้น ไม่รวมออบเจ็กต์อื่นๆ เช่น เวิร์กสเตชัน ปริ๊นเตอร์
- ซิงโครไนซ์ได้เฉพาะยูสเซอร์และ OU รวมไปถึงแอตทริบิวต์ แต่ไม่ซิงโครไนซ์กรุ๊ป แต่สามารถจัดการความสามารถในการจัดการเป็นกลุ่มได้โดยใช้ ConsoleOne
- ไม่มีการซิงโครไนซ์ส่วนควบคุมการเข้าถึงและการอนุญาตการเข้าถึง ซึ่งมีความยุ่งยาก
- โปรแกรมไม่ได้กำจัดส่วนประกอบใดๆ ของแอ็กทีฟไดเรกทอรี

การใช้ Novell AM for Windows 2000 จัดการ eDirectory และวินโดวส์ 2000 เซิร์ฟเวอร์ ช่วยลดการจัดการแบบวันต่อวัน ผู้ดูแลระบบสามารถจัดการยูสเซอร์ กรุ๊ป และ OU จากโปรแกรมเดียว และส่งผลการเปลี่ยนแปลงไปให้ทั้งระบบ ยูสเซอร์มีรหัสผ่านเดียวทั้ง 2 ระบบ และเมื่อมีการเปลี่ยนรหัสผ่าน โนเวลล์ AM จะกระจายความเปลี่ยนแปลงดังกล่าวให้ทั้งแอ็กทีฟไดเรกทอรีและ eDirectory

9.2.1 การซิงโครไนซ์รหัสผ่าน

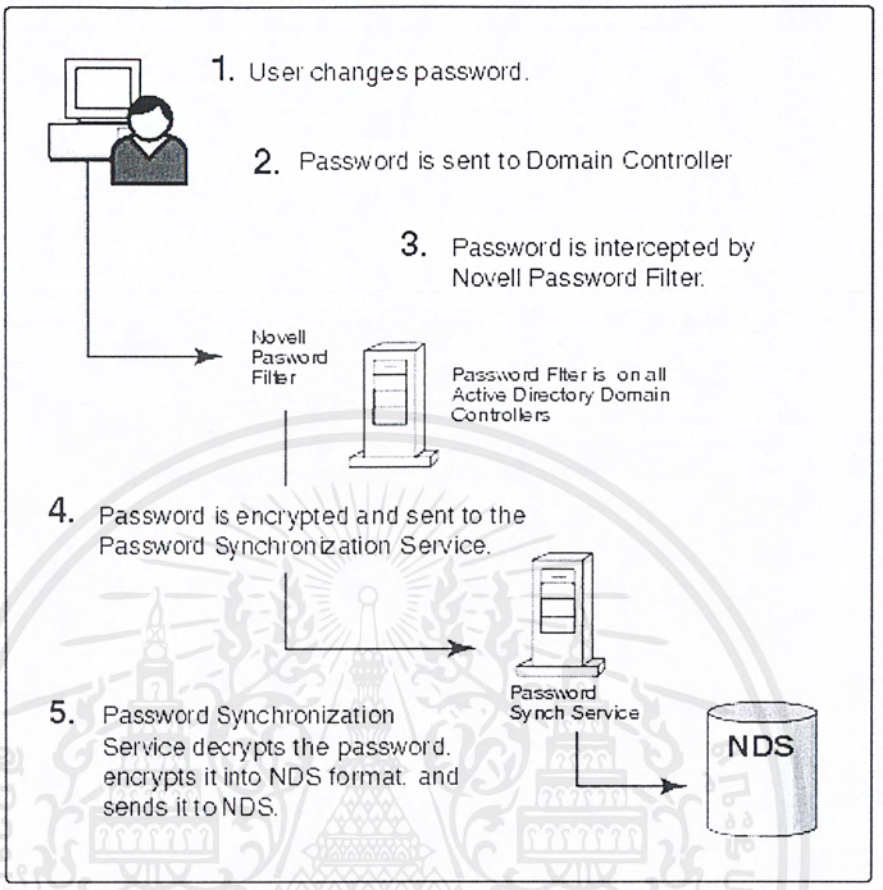
การซิงโครไนซ์รหัสผ่านแบบ 2 ทาง เป็นข้อดีอีกหนึ่งอย่างของ Novell AM for Windows 2000 ที่ดีกว่าวิธีการซิงโครไนซ์ไดเรกทอรีแบบอื่น การซิงโครไนซ์แบบอื่นมีความลำบากและยุ่งยาก เช่น ต้องให้ยูสเซอร์ใช้โปรแกรมพิเศษหรือเว็บเพจในการซิงโครไนซ์รหัสผ่านในระบบ ข้อเสียของการเปลี่ยนรหัสผ่านจากเวิร์กสเตชันแทนการใช้โปรแกรมพิเศษคือ อาจทำให้เกิดความไม่ถูกต้องของรหัสผ่านระหว่าง 2 ไดเรกทอรี

เฉพาะ Novell AM for Windows 2000 ที่ซิงโครไนซ์รหัสผ่านระหว่าง eDirectory และแอ็กทีฟไดเรกทอรี โดยไม่ต้องมีผู้อื่นเข้ามาเกี่ยวข้อง

การซิงโครไนซ์รหัสผ่านประกอบด้วย 3 ส่วนคือ 1. NDS Password Synchronization Container 2. Win32 Service "Novell Password Synchronize Service" 3.Windows 2000 Password Filter ส่วนประกอบเหล่านี้ช่วยทำให้รหัสผ่านซิงโครไนซ์กันระหว่าง eDirectory และ แอ็กทีฟไดเรกทอรี ไม่ว่าจะเปลี่ยนรหัสผ่านจากหน้าจอวินโดวส์ หรือโปรแกรมการจัดการแอ็กทีฟไดเรกทอรี

9.2.2 การทำงานของ Novell AM for Windows 2000

เมื่อยูสเซอร์เปลี่ยนรหัสผ่านในแอ็กทีฟไดเรกทอรี รหัสผ่านจะถูกส่งไปยังโดเมนคอนโทรลเลอร์ในรูปแบบที่สามารถแปลงกลับได้ แอ็กทีฟไดเรกทอรีโดเมนคอนโทรลเลอร์ถอดรหัสรหัสผ่านของยูสเซอร์ใหม่และส่งรหัสผ่านผ่าน Password Filter ซึ่งติดตั้งบนโดเมนคอนโทรลเลอร์ทุกตัวที่อยู่ในแอ็กทีฟไดเรกทอรี โดเมนจะนำรหัสผ่านของยูสเซอร์ใหม่ เข้ารหัสด้วย RSA พลับลิคคีย์ซึ่งเก็บอยู่ใน NDS Password Synchronization Container และส่งไปให้ NDS Password Synchronization Service ซึ่งรันอยู่บนวินโดวส์ 2000 เซิร์ฟเวอร์ตัวเดิม NDS Password Synchronization Service ถอดรหัสของรหัสผ่านโดยใช้ไพรเวทคีย์ เข้ารหัสให้อยู่ในรูปแบบของ NDS และอัพเดตออบเจ็กต์ที่สัมพันธ์กันใน NDS Password จะไม่ถูกส่งผ่านระบบเครือข่ายในรูปของไฟล์เท็กซ์หรืออยู่ในรูปที่ไม่ปลอดภัย



รูปที่ 9-3 แสดงการทำงานของ Novell Account Manager for Windows 2000

บทที่ 10
การทำงานร่วมกันของระบบปฏิบัติการ

การทำงานร่วมของระบบปฏิบัติการต่างๆ สามารถทำได้หลายวิธีซึ่งแต่ละวิธีมีกลไกการทำงานที่แตกต่างกัน นอกจากนี้แต่ละวิธีมีข้อดีข้อเสียของมัน เราสามารถสรุปวิธีการต่างๆได้เป็นดังนี้

10.1 การทำงานร่วมกันระหว่างระบบปฏิบัติการวินโดวส์ 2000 และยูนิกซ์

10.1.1 Kerberos

เนื่องจาก Kerberos เป็นวิธีพิสูจน์ตนที่ใช้ได้ในระบบปฏิบัติการทั้งวินโดวส์ 2000 เซิร์ฟเวอร์และยูนิกซ์ ซึ่งมีการเก็บข้อมูลและกลไกในการพิสูจน์ตนเหมือนกัน ดังนั้นจึงสามารถใช้ Kerberos เป็นตัวกลางในการติดต่อระหว่างวินโดวส์ 2000 เซิร์ฟเวอร์และยูนิกซ์ให้สามารถทำงานร่วมกันได้

ข้อดีของการใช้ Kerberos

- ไม่เสียค่าใช้จ่ายใดๆ เนื่องจากเป็นส่วนประกอบหนึ่งที่มีให้ในระบบปฏิบัติการ
- เป็นวิธีการที่เป็นมาตรฐาน ซึ่งระบบปฏิบัติการส่วนใหญ่สนับสนุน ซึ่งสามารถขยายระบบได้ง่าย
- เป็นระบบที่มีผู้พัฒนาอย่างต่อเนื่อง เห็นได้จากมีเวอร์ชันออกมาเรื่อยๆ ทำให้เป็นระบบที่มีความน่าเชื่อถือ และมีการพัฒนาให้ทันต่อเทคโนโลยี

ข้อเสียของการใช้ Kerberos

- มีหลายเวอร์ชัน ซึ่งแต่ละระบบปฏิบัติการใช้ Kerberos คนละเวอร์ชันกัน ทำให้ไม่สามารถใช้งานร่วมกันได้ ต้องเลือกที่ใช้เวอร์ชันเดียวกัน
- สามารถทำได้แค่การทำงานร่วมกันของระบบปฏิบัติการ แต่ไม่สามารถซิงโครไนซ์รหัสผ่านได้

10.1.2 Service For UNIX (SFU)

เป็นโปรแกรมที่ใช้ในการซิงโครไนซ์รหัสผ่านระหว่างวินโดวส์ 2000 และยูนิกซ์ ต้องมีการติดตั้งโปรแกรมทั้งเครื่องที่มีระบบปฏิบัติการวินโดวส์และยูนิกซ์

ข้อดีของการซิงโครไนซ์รหัสผ่าน

- ติดตั้งและใช้งานง่าย
- สามารถซิงโครไนซ์รหัสผ่าน สำหรับคอมพิวเตอร์ที่อยู่ในโดเมนของวินโดวส์หรือ รหัสผ่าน ที่เป็นโลคอล สำหรับสแตนด์อโลนคอมพิวเตอร์ (Standalone Computer) นอกจากนี้ยังสามารถใช้ได้ทั้งวินโดวส์เอ็นทีและวินโดวส์ 2000
- สามารถซิงโครไนซ์รหัสผ่านของระบบยูนิกซ์ได้ทั้งสแตนด์อโลนคอมพิวเตอร์หรือคอมพิวเตอร์ที่อยู่ใน NIS หรือ NIS+

- มีการเข้ารหัส รหัสผ่าน ก่อนส่ง โดยใช้การเข้ารหัสแบบไพรเวตคีย์
- สามารถซิงโครไนซ์รหัสผ่านจากวินโดวส์ไปยูนิกซ์หรือจากยูนิกซ์ไปวินโดวส์ก็ได้
- เมื่อเกิดการเปลี่ยนแปลงรหัสผ่านขึ้น จะบอกให้เครื่องอื่นๆ รู้ทันทีที่เกิดการเปลี่ยนแปลง ซึ่งต่างจากวิธีอื่นที่จะเก็บการเปลี่ยนแปลงที่เกิดขึ้นไว้ก่อน แล้วค่อยบอกทีเดียวภายหลัง (Batch)

ข้อเสียของการใช้ SFU

- ต้องเสียค่าใช้จ่าย
- จะสามารถซิงโครไนซ์รหัสผ่านได้ในกรณีที่ยูสเซอร์เนมของผู้ใช้บนยูนิกซ์และวินโดวส์เหมือนกันเท่านั้น
- เป็นระบบปิด และสนับสนุนเฉพาะระหว่างวินโดวส์และยูนิกซ์ และไม่มีการเปิดเผยซอร์สโค้ด ทำให้ผู้อื่นไม่สามารถนำไปพัฒนาได้

10.2 การทำงานร่วมกันระหว่างระบบปฏิบัติการวินโดวส์ 2000 และโนเวลล์เน็ตแวร์

10.2.1 Service For Netware (SFN)

เป็นโปรแกรมที่ใช้ในการซิงโครไนซ์รหัสผ่านระหว่างวินโดวส์ 2000 และโนเวลล์เน็ตแวร์ โดยสามารถเลือกได้ว่าต้องการซิงโครไนซ์ทางเดียวหรือ 2 ทาง โดยถ้าเป็นการซิงโครไนซ์แบบทางเดียวจะเป็นการซิงโครไนซ์จากวินโดวส์ 2000 ไปยังโนเวลล์เน็ตแวร์

ข้อดีของการใช้ SFN

- สามารถซิงโครไนซ์รหัสผ่านได้
- ติดตั้งและใช้งานง่าย
- ลดการจัดการไดเรกทอรีผ่านการซิงโครไนซ์ 2 ทาง
- สามารถย้ายข้อมูลจาก NDS ไปสู่วินโดวส์ 2000 เซิร์ฟเวอร์ได้

ข้อเสียของการใช้ SFN

- ต้องเสียค่าใช้จ่าย
- เป็นระบบปิด และสนับสนุนเฉพาะระหว่างวินโดวส์และโนเวลล์เน็ตแวร์ และไม่มีการเปิดเผยซอร์สโค้ด ทำให้ผู้อื่นไม่สามารถนำไปพัฒนาได้

10.2.2 Novell Account Management for Windows 2000 (AM)

เป็นโปรแกรมของโนเวลล์ซึ่งช่วยซิงโครไนซ์รหัสผ่านระหว่างวินโดวส์ 2000 และโนเวลล์เน็ตแวร์ โดยสามารถซิงโครไนซ์ 2 ทางได้ ทำให้สามารถใช้เซิร์ฟเวอร์ตัวใดเป็นศูนย์กลางก็ได้

ข้อดีของการใช้ Novell AM for Windows 2000

- สามารถซิงโครไนซ์รหัสผ่านแบบ 2 ทางได้

- ไม่ต้องมีผู้ดูแลในการซิงโครไนซ์
- ใช้โปรแกรมพิเศษในการเปลี่ยนแปลงรหัสผ่าน ทำให้ความถูกต้องระหว่าง 2 ไดเรกทอรีมีความถูกต้องสูง
- ไม่มีการส่งยูสเซอร์เนมและรหัสผ่านผ่านระบบเครือข่ายในรูปของเท็กซ์ไฟล์หรืออยู่ในรูปแบบที่ไม่ปลอดภัย

ข้อเสียของการใช้ Novell AM for Windows 2000

- ไม่สามารถจัดการออบเจ็กต์อื่นๆ นอกจาก ยูสเซอร์ กรุ๊ป และ OU ได้
- ซิงโครไนซ์ได้เฉพาะยูสเซอร์และ OU โดยไม่สามารถซิงโครไนซ์กรุ๊ปได้
- ไม่มีการซิงโครไนซ์ส่วนการควบคุมการเข้าถึงและส่วนอนุญาตเข้าถึงระบบ
- เสียค่าใช้จ่ายเพิ่มเติม

10.3 การทำงานร่วมกันระหว่างระบบปฏิบัติการวินโดวส์เอ็นทีเซิร์ฟเวอร์และโนเวลล์เน็ตแวร์

10.3.1 NDS for NT

เป็นโปรแกรมที่ใช้ในการซิงโครไนซ์รหัสผ่านระหว่างวินโดวส์เอ็นทีเซิร์ฟเวอร์และโนเวลล์เน็ตแวร์ โดยการใช้ NDS บนวินโดวส์เอ็นทีเซิร์ฟเวอร์

ข้อดีของการใช้ NDS for NT

- สามารถซิงโครไนซ์รหัสผ่านได้
- ติดตั้งและใช้งานง่าย

ข้อเสียของการใช้ NDS for NT

- ไม่เหมาะสมกับวินโดวส์ 2000 เนื่องจากวินโดวส์ 2000 ใช้แอ็กทีฟไดเรกทอรี ทำให้โปรแกรมนี้ไม่สามารถจัดการได้ทั้งหมด

10.3.2 Directory Service Manager for NetWare (DSMN)

เป็นโปรแกรมที่ซิงโครไนซ์ระหว่างวินโดวส์เอ็นทีและโนเวลล์เน็ตแวร์ โดยเป็นการซิงโครไนซ์แบบทางเดียว คือจากวินโดวส์เอ็นทีไปยัง NDS ดังนั้นต้องให้ตัววินโดวส์เอ็นทีเซิร์ฟเวอร์เป็นศูนย์กลาง

ข้อดีของการใช้ DSMN

- มีออบชันให้เลือกมาก
- สามารถจัดการไดเรกทอรีเซอร์วิซจากที่ใดก็ได้ในระบบเครือข่าย
- มีการแบ็กอัพฐานข้อมูลของแอ็กเคาต์ไปยังทุกที่ในระบบเครือข่าย
- สามารถรีโคเวอรีได้ในกรณีที่ระบบล่ม

ข้อเสียของการใช้ DSMN

- เสียค่าใช้จ่ายเพิ่มเติม

- สามารถซิงโครไนซ์ได้แค่ทางเดียว

10.3.3 โปรแกรมของบริษัทอื่น

นอกจาทางไมโครซอฟต์และโนเวลล์แล้ว ยังมีบริษัทอื่นๆ คิดวิธีที่สามารถทำให้ซิงโครไนซ์รหัสผ่านแบบสองทางได้ เช่น โปรแกรมของบริษัท SecurePass เป็นต้น

ข้อดีของการใช้โปรแกรมของบริษัทอื่น

- สามารถซิงโครไนซ์รหัสผ่านแบบ 2 ทางได้
- ไม่ขึ้นกับบริษัทผู้ผลิตมากจนเกินไป

ข้อเสียของการใช้โปรแกรมของบริษัทอื่น

- ไม่มีการเปิดเผยการทำงาน
- เสียค่าใช้จ่ายเพิ่มเติม
- การพัฒนาเป็นไปโดยยากลำบาก

10.4 การทำงานร่วมกันระหว่างระบบปฏิบัติการวินโดวส์เอ็นทีเซิร์ฟเวอร์และยูนิกซ์

10.4.1 Microsoft Windows NT Service For UNIX

เป็นโปรแกรมที่ช่วยให้สามารถการซิงโครไนซ์รหัสผ่านของผู้ใช้บนระบบที่เป็นวินโดวส์เอ็นที และระบบที่เป็นยูนิกซ์ได้ โดยจะเป็นการซิงโครไนซ์แบบทางเดียว

ข้อดีของการใช้ Microsoft Windows NT Service For UNIX

- ติดตั้งและใช้งานง่าย
- สามารถรวมระบบที่อยู่บนวินโดวส์เอ็นทีได้ทั้งในปัจจุบัน และรองรับถึงอนาคต
- สามารถรองรับกับสภาพแวดล้อมที่มีหลายระบบปฏิบัติการได้
- สามารถเลือกรูบแบบการซิงโครไนซ์ที่เหมาะสมกับองค์กรได้ (แบบที่มีการเข้ารหัสหรือแบบที่ไม่มีการเข้ารหัส)

ข้อเสียของการใช้ Microsoft Windows NT Service For UNIX

- ต้องเสียค่าใช้จ่ายเพิ่ม
- สามารถซิงโครไนซ์รหัสผ่านได้แบบทางเดียวเท่านั้นทำให้เกิดการจำกัดการขยายตัวของระบบในอนาคต

10.4.2 โปรแกรมของบริษัทอื่น

เนื่องจากบริษัทไมโครซอฟต์สร้างแต่การซิงโครไนซ์รหัสผ่านแบบทางเดียว ดังนั้นจึงทำให้มีบริษัทอื่นๆ คิดวิธีที่สามารถทำให้ซิงโครไนซ์รหัสผ่านแบบสองทางได้ เช่น โปรแกรมของบริษัท COSuser หรือโปรแกรมของบริษัท SecurePass เป็นต้น

ข้อดีของการใช้โปรแกรมของบริษัทอื่น

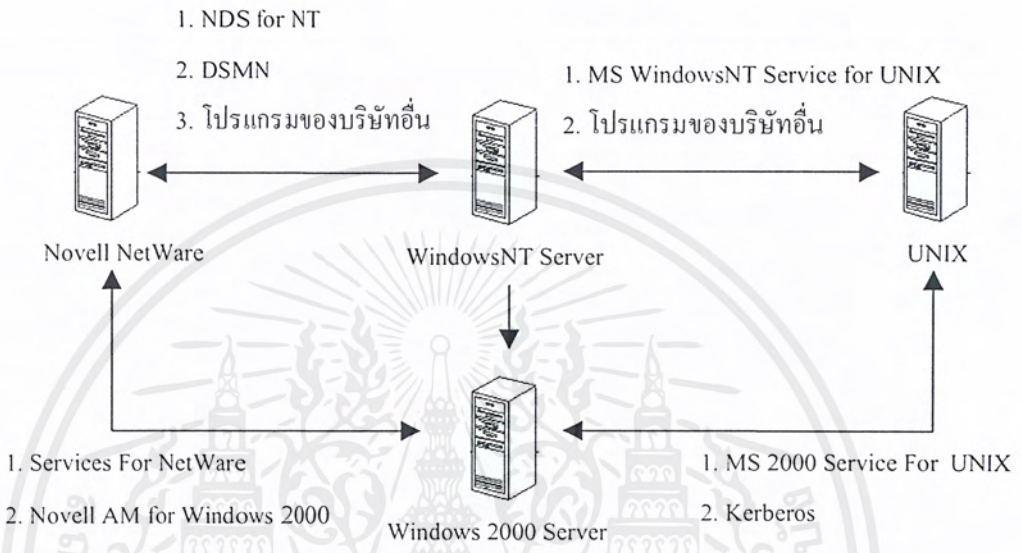
- สามารถซิงโครไนซ์รหัสผ่านแบบ 2 ทางได้
- ไม่ขึ้นกับบริษัทผู้ผลิตมากจนเกินไป

ข้อเสียของการใช้โปรแกรมของบริษัทอื่น

- ไม่มีการเปิดเผยการทำงาน
- เสียค่าใช้จ่ายเพิ่มเติม
- การพัฒนาเป็นไปโดยยากลำบาก

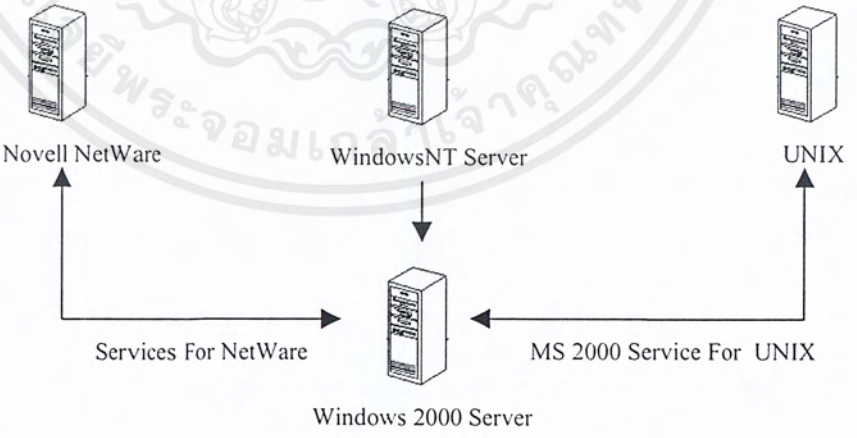
บทที่ 11
ระบบที่มีระบบปฏิบัติการต่าง ๆ

11.1 แนวทางในการทำงานร่วมกันระหว่างเซิร์ฟเวอร์ต่างๆ

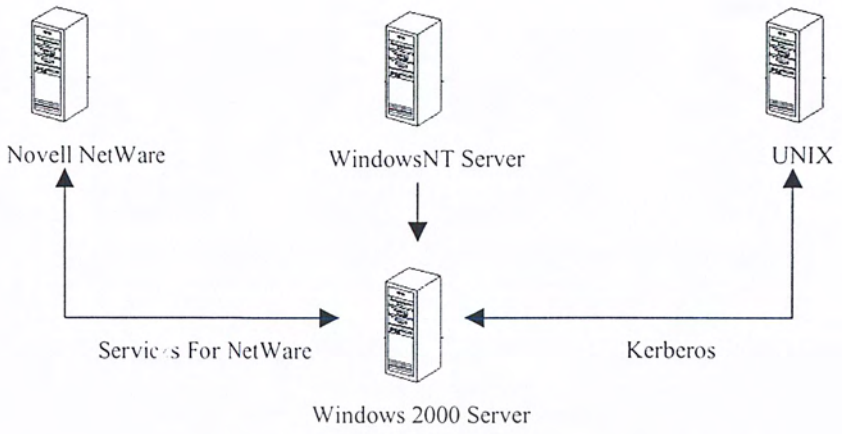


รูปที่ 11-1 แนวทางทั้งหมดที่ทำให้ระบบสามารถทำการซิงโครไนซ์กันได้

11.2 ระบบที่มีศูนย์กลาง

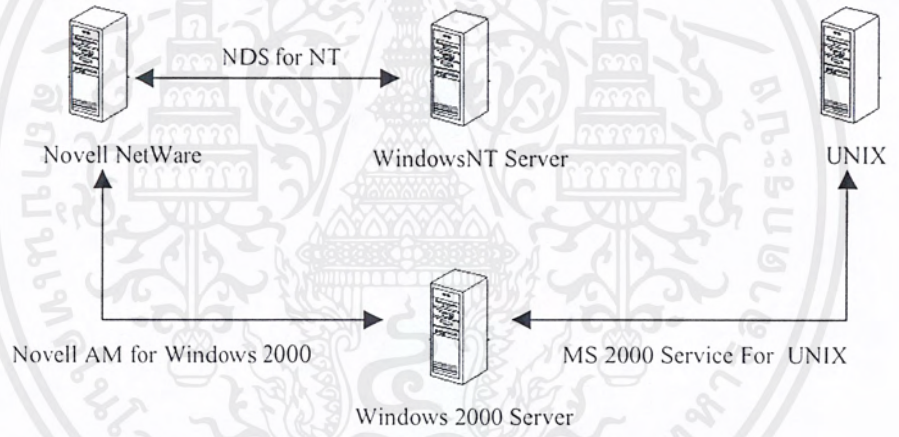


รูปที่ 11-2 ระบบที่มีศูนย์กลางแบบที่ 1

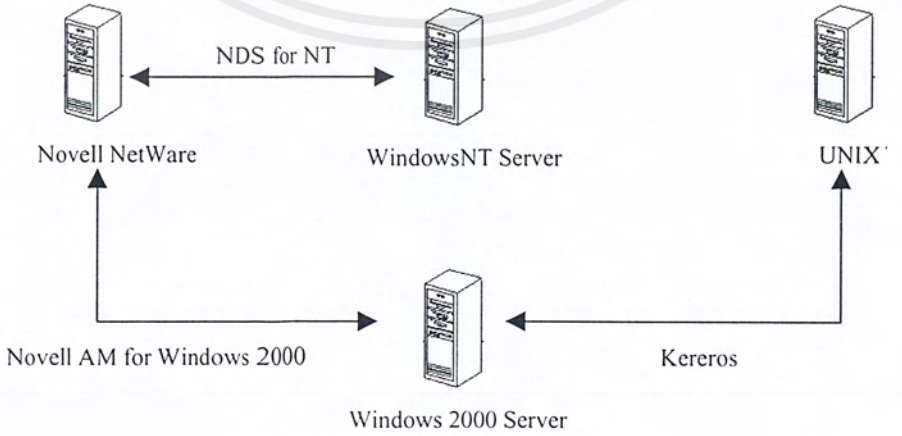


รูปที่ 11-3 ระบบที่มีศูนย์กลางแบบที่ 2

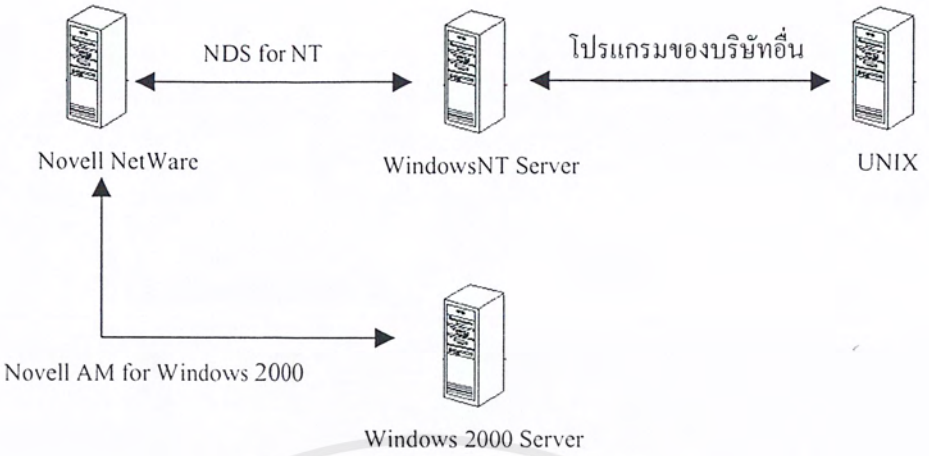
11.3 ระบบที่ไม่มีศูนย์กลาง



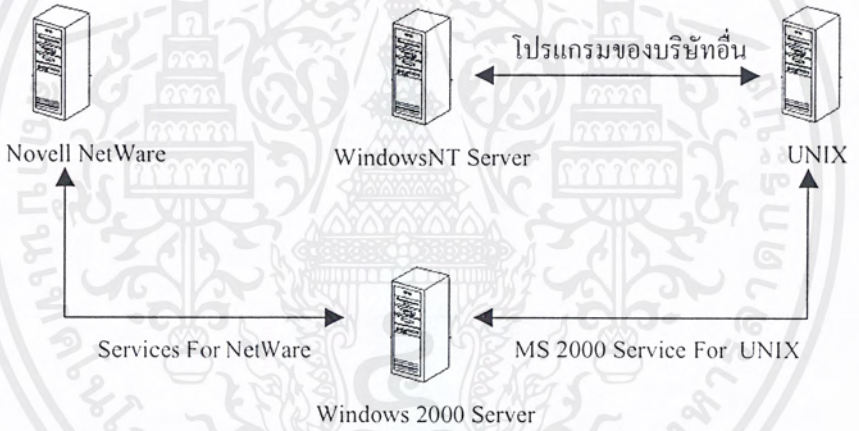
รูปที่ 11-4 ระบบที่ไม่มีศูนย์กลางแบบที่ 1



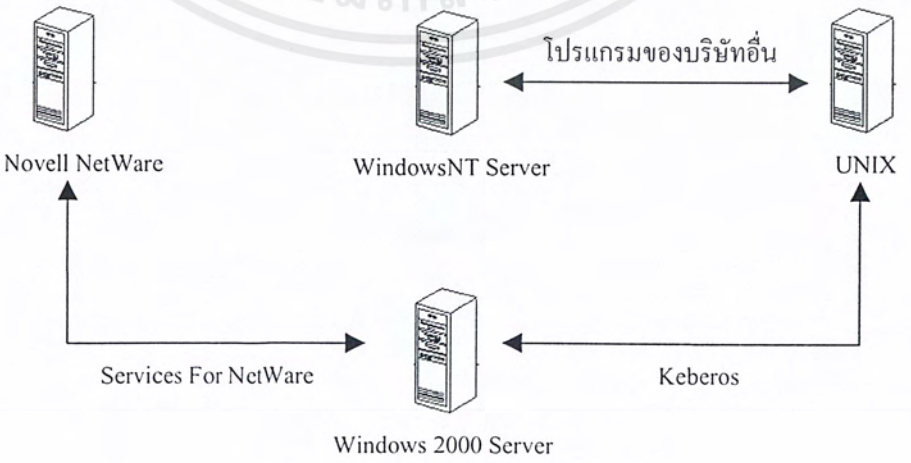
รูปที่ 11-5 ระบบที่ไม่มีศูนย์กลางแบบที่ 2



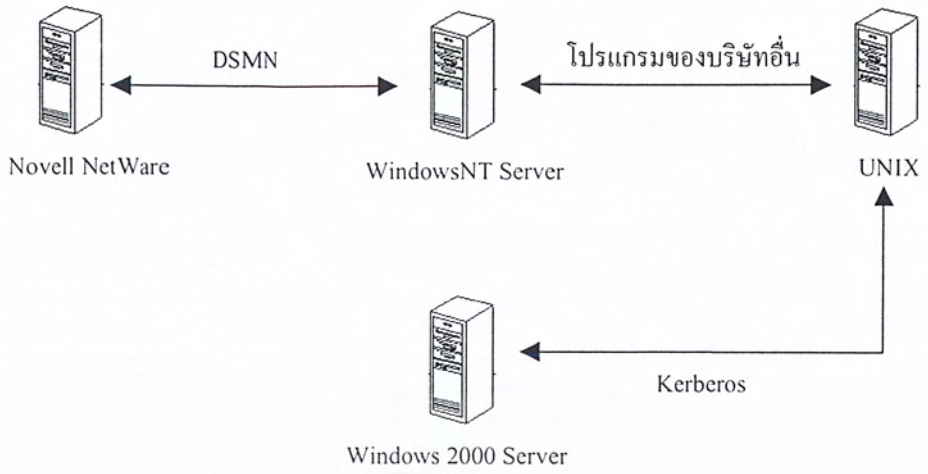
รูปที่ 11-6 ระบบที่ไม่มีศูนย์กลางแบบที่ 3



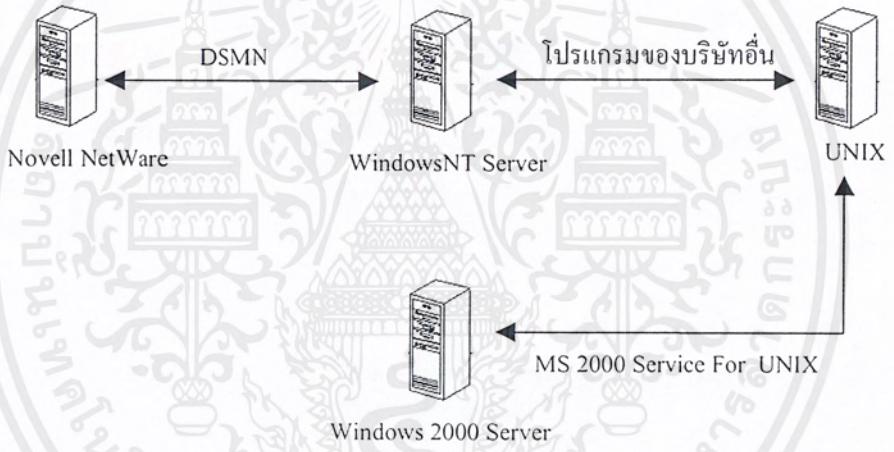
รูปที่ 11-7 ระบบที่ไม่มีศูนย์กลางแบบที่ 4



รูปที่ 11-8 ระบบที่ไม่มีศูนย์กลางแบบที่ 5



รูปที่ 11-9 ระบบที่ไม่มีศูนย์กลางแบบที่ 6



รูปที่ 11-10 ระบบที่ไม่มีศูนย์กลางแบบที่ 7

บทที่ 12
ระบบเครือข่ายในภาควิชา

12.1 เซิร์ฟเวอร์ในภาควิชา

ในภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มีเซิร์ฟเวอร์ที่สนใจดังนี้

12.1.1 Diamond

เป็นระบบปฏิบัติการยูนิกซ์ HP-UX ซึ่งให้บริการต่างๆ ดังนี้

- บริการเก็บแฟ้มข้อมูล
- บริการโฮมเพจ
- บริการรับส่งอิเล็กทรอนิกส์เมล์ (SMTP POP3)
- บริการโปรแกรมแอพพลิเคชันต่างๆ
- บริการโดเมนเนม (DNS) อินเทอร์แอ็กทีฟเชล
- บริการฐานเวลา (NTP)
-

12.1.2 Compnet

เป็นระบบปฏิบัติการโนเวลล์เน็ตแวร์ ซึ่งให้บริการต่างๆ ดังนี้

- บริการเซิร์ฟเวอร์ NDP
- บริการจัดเก็บแฟ้มข้อมูล
- บริการโปรแกรมแอพพลิเคชันต่างๆ
- บริการคิวการพิมพ์ไปยังเครื่องพิมพ์เลเซอร์

12.1.3 Garnet

เป็นระบบปฏิบัติการวินโดวส์เอ็นที 4.0 ซึ่งให้บริการต่างๆ ดังนี้

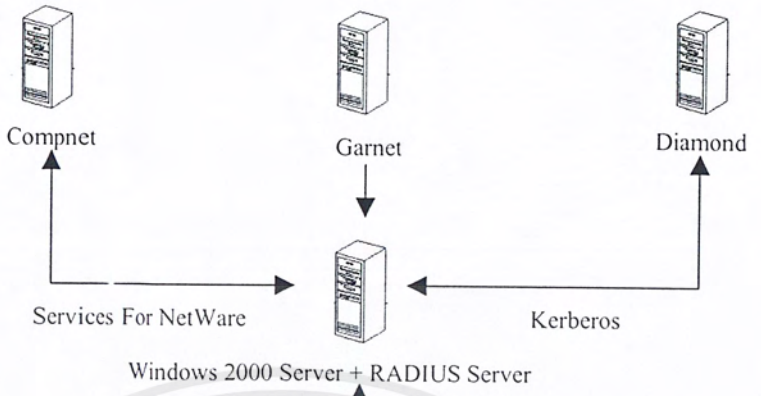
- บริการไพรมารีโดเมนคอนโทรลเลอร์ "CE-NET"
- บริการเรพลิเคชันระหว่าง CE-NET กับ CE-NT ผ่าน NDS for NT
- บริการเซิร์ฟเวอร์ DHCP
- บริการเซิร์ฟเวอร์ Wins
- บริการโปรแกรมแอพพลิเคชันต่างๆ

12.1..4 Amethyst

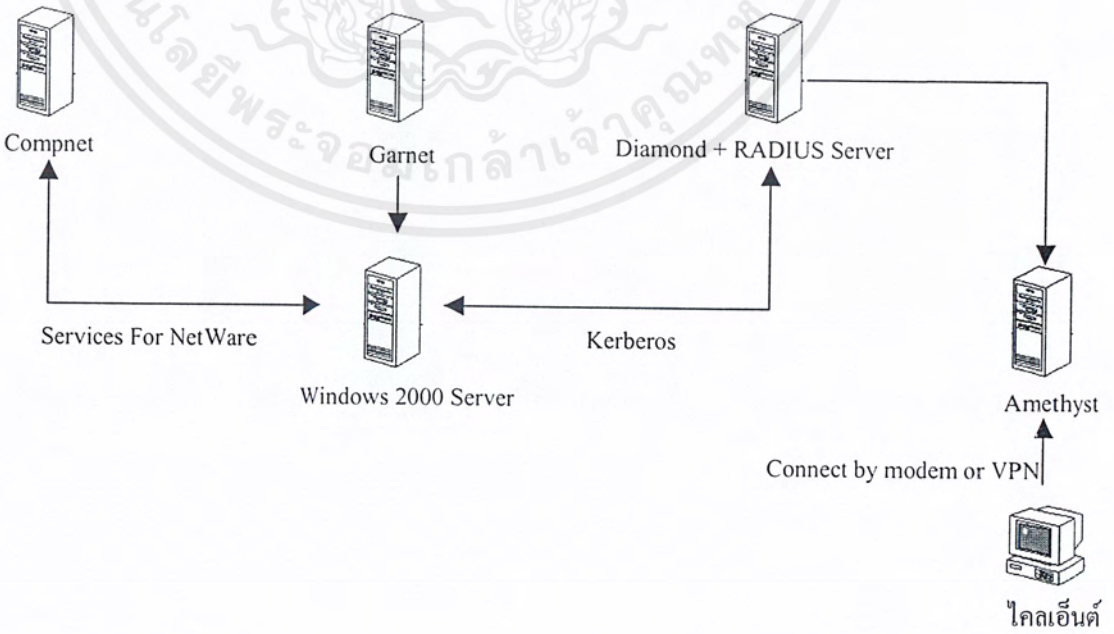
เป็นระบบปฏิบัติการลีนุกซ์ (Linux) ซึ่งให้บริการต่างๆ ดังนี้

- บริการไดอัลอับเซิร์ฟเวอร์ (Slip/PPP)

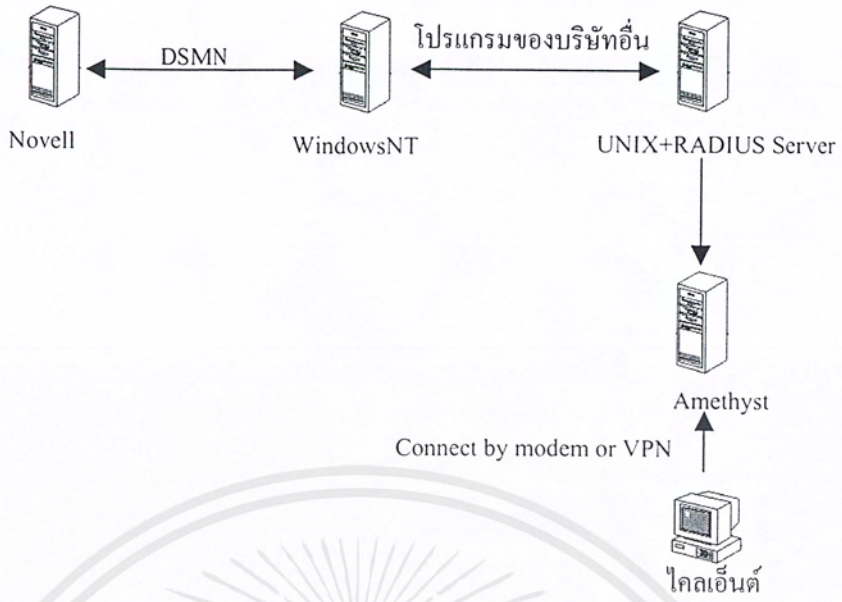
12.2 การออกแบบระบบภาควิชา



รูปที่ 12-1 แสดงการทำงานร่วมกันของเซิร์ฟเวอร์ในภาควิชาแบบที่ 1



รูปที่ 12-2 แสดงการทำงานร่วมกันของเซิร์ฟเวอร์ในภาควิชาแบบที่ 2

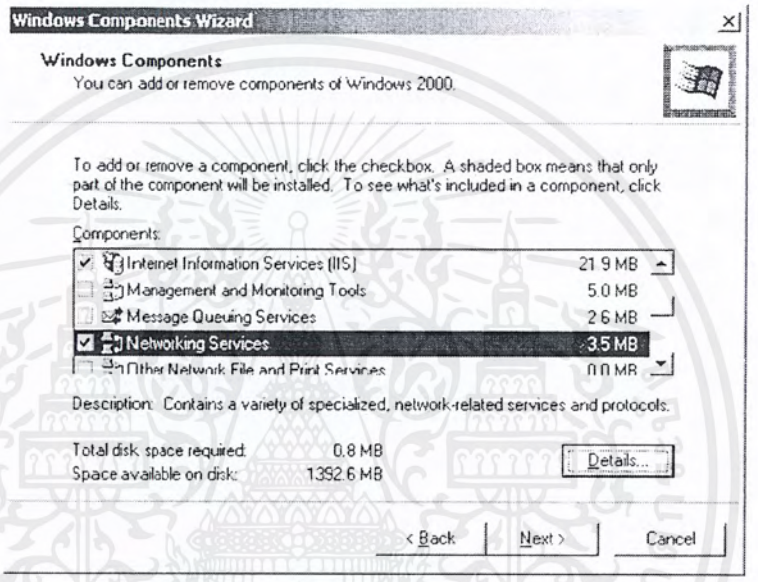


รูปที่ 12-3 แสดงการทำงานร่วมกันของเซิร์ฟเวอร์ในภาควิชาแบบที่ 3

บทที่ 13
การคอนฟิกูเรชัน

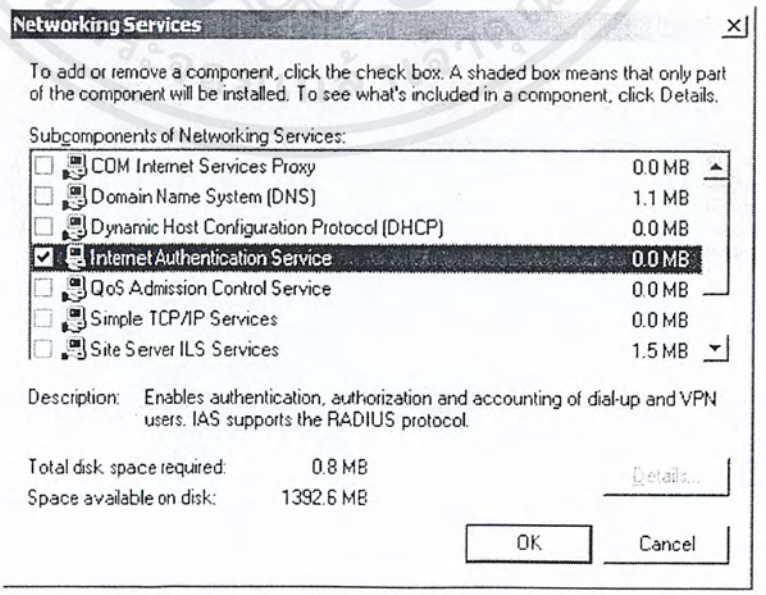
10.1 การติดตั้ง IAS

เริ่มจากไปที่ Start Menu ของวินโดวส์ จากนั้นเลือก Settings และจากนั้นเลือกคอนโทรลพาแนล (Control Panel) ในหน้าต่างของคอนโทรลพาแนล เลือก Add/Remove Programs และเลือก Add/Remove Windows Components ในหน้าต่างนี้ ให้เลือกที่ Networking Services ดังรูป



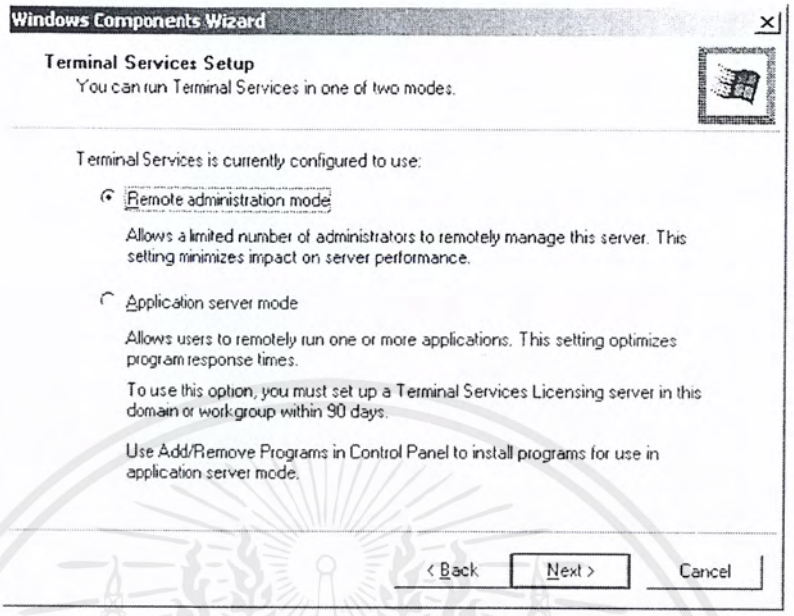
รูปที่ 13-1 แสดงหน้าต่าง Windows Components

ในหน้าต่างของ Networking Services นั้น เลือก Internet Authentication Services ดังรูป



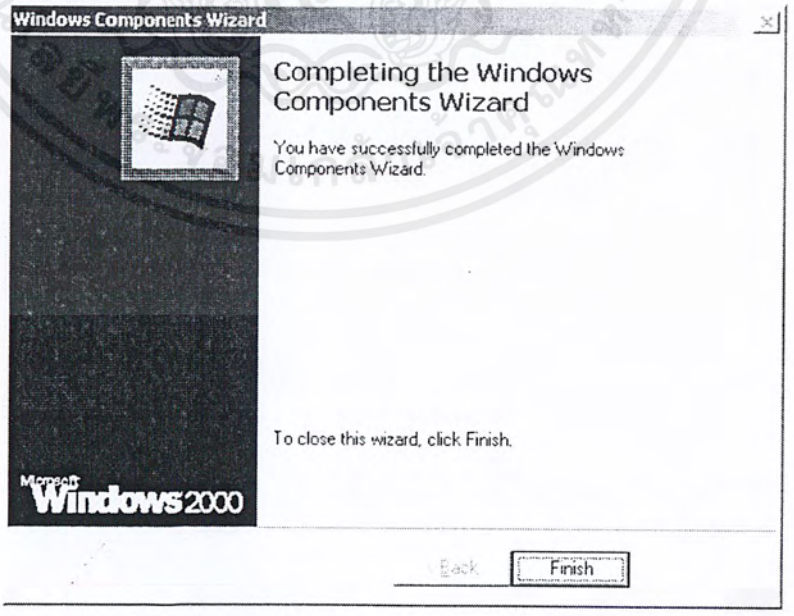
รูปที่ 13-2 แสดงหน้าต่าง Networking Services

หลังจากนั้นกด OK แล้วจะมีหน้าต่างใหม่ขึ้นมา ดังรูป



รูปที่ 13-3 แสดงหน้าต่าง Windows Components Wizard

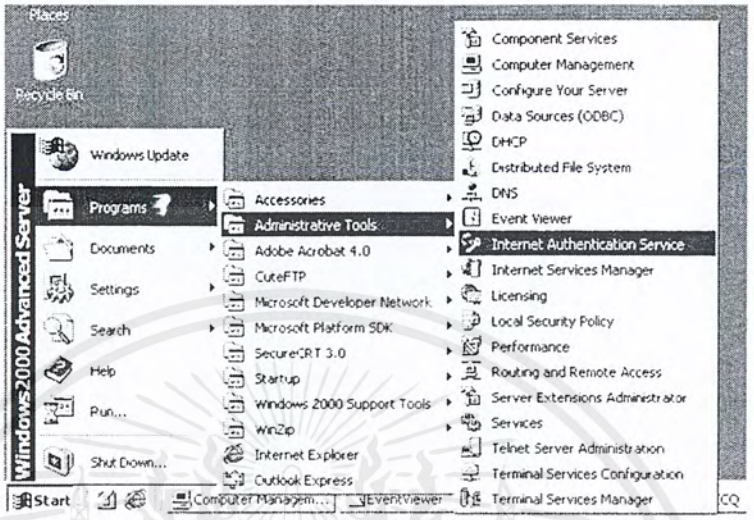
โดย Remote administration mode นั้น เป็นการจำกัดจำนวนของผู้ดูแลระบบให้มีน้อย เป็นการทำให้มีการกระทำบนเซิร์ฟเวอร์น้อยที่สุด ส่วน Application server mode นั้น เป็นแอพพลิเคชันเซิร์ฟเวอร์ โดยผู้ใช้ระบบสามารถใช้โปรแกรมที่รันอยู่บนเซิร์ฟเวอร์ได้ ในที่นี้ได้เลือก Remote administration mode หลังจากนั้นเลือก finish ดังรูป



รูปที่ 13-4 แสดงหน้าจอการติดตั้ง IAS สำเร็จ

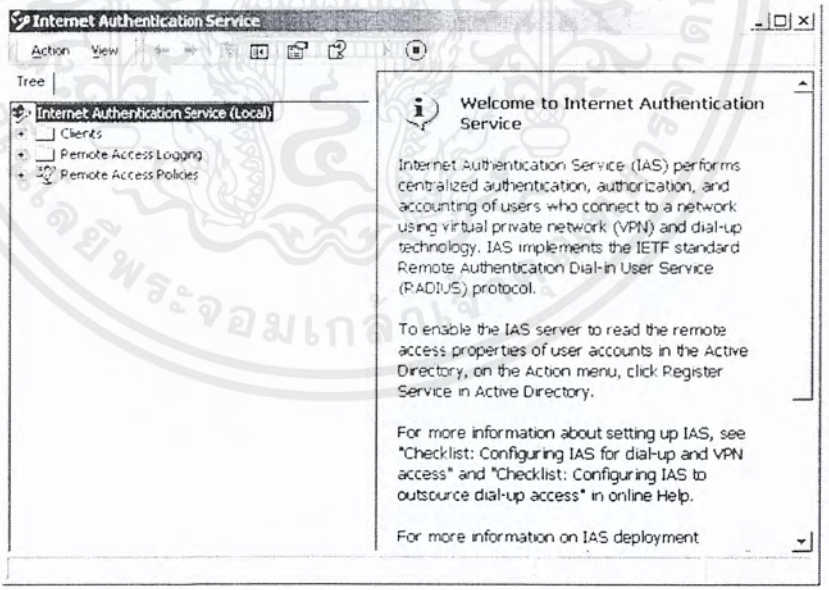
10.2 การคอนฟิกูเรชัน IAS

สามารถคอนฟิกูเรชัน IAS ได้โดยกด Start Menu ในวินโดวส์ หลังจากนั้นเลือก Programs และ Administrative tools และ Internet Authentication Service ดังรูป



รูปที่ 13-5 แสดงการเรียก IAS เพื่อการคอนฟิกูเรชัน

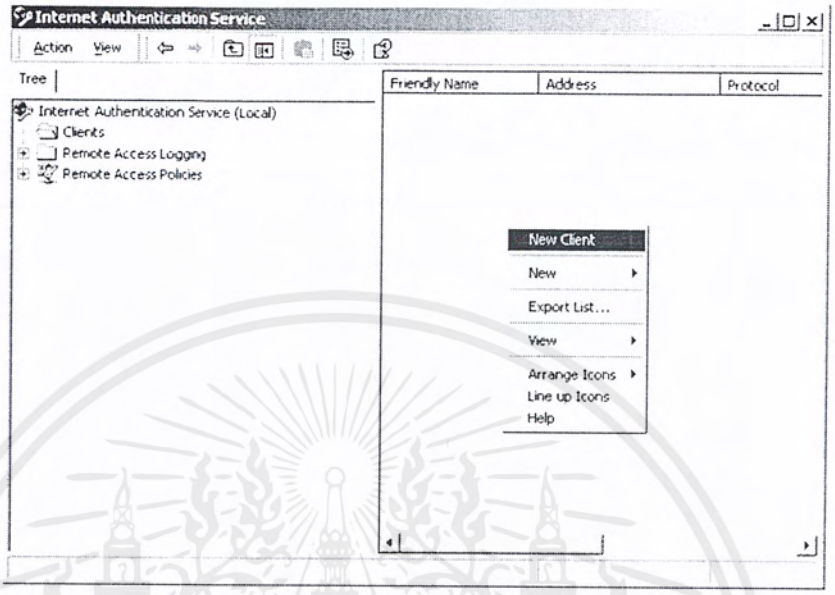
หลังจากนั้นจะมีหน้าต่างของ IAS ดังรูป



รูปที่ 13-6 แสดงหน้าต่างของ IAS

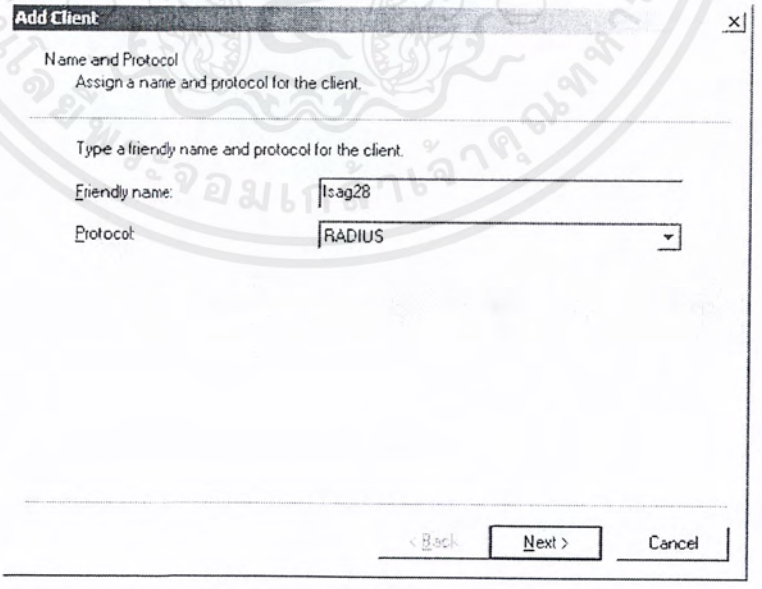
หน้าต่างนี้ประกอบด้วย 3 ส่วน คือ ส่วน Clients ส่วน Remote Access Logging และ Remote Access Policies โดยส่วน Clients นั้นคือส่วนที่บอกว่ามี RADIUS ไคลเอ็นต์ตัวใดบ้าง และบอกรายละเอียดของ RADIUS ไคลเอ็นต์ตัวต่างๆ และส่วน Remote Access Logging เป็นส่วนที่บอกว่าเมื่อมีเหตุ

การณ์ต่างๆ นั้น มีการเก็บล็อกไฟล์ไว้ที่ใด และส่วนสุดท้ายคือส่วน Remote Access Policies คือส่วนที่เพิ่ม เติม ลบ และแก้ไขข้อจำกัดในการติดต่อของผู้ใช้ระบบต่างๆ โดยที่ไคลเอ็นต์นั้น สามารถเพิ่ม RADIUS ไคลเอ็นต์ได้โดยเลือก Clients และคลิ้กขวาที่หน้าจอทางด้านขวา ดังรูป



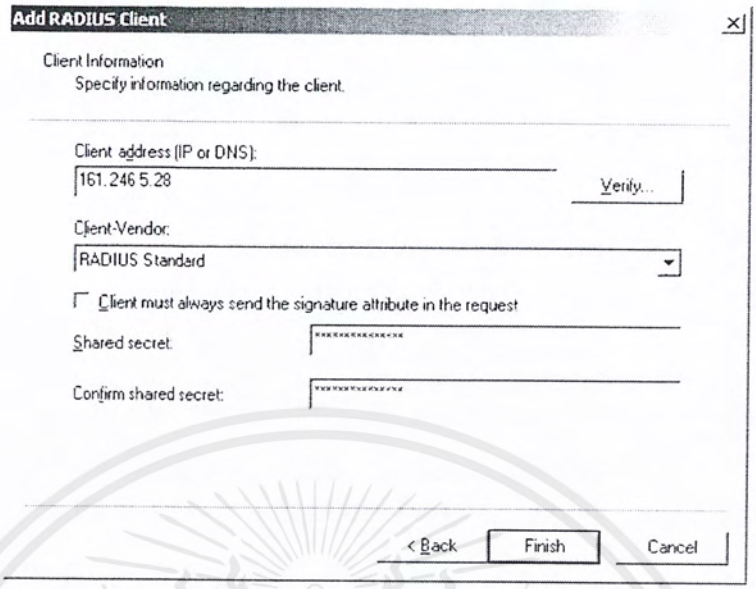
รูปที่ 13-7 แสดงการเพิ่ม RADIUS ไคลเอ็นต์ขั้นตอนแรก

เมื่อเลือก Net Client แล้ว จะมีหน้าต่างใหม่ขึ้น โดยให้ใส่ชื่อของ RADIUS ไคลเอ็นต์ โดยใส่ในช่อง Friendly Name และเลือก Protocol เป็น RADIUS ดังรูป



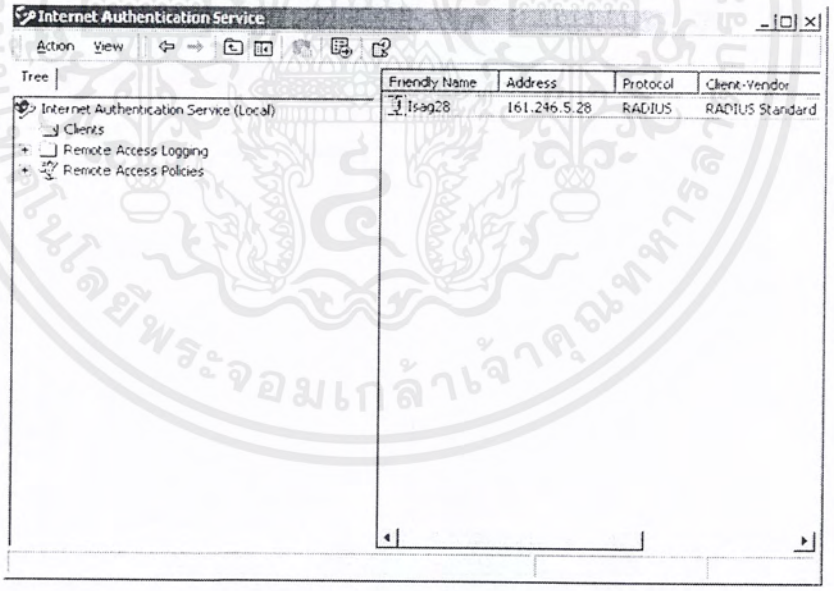
รูปที่ 13-8 แสดงหน้าจอใส่ชื่อของ RADIUS ไคลเอ็นต์และโพรโตคอล

หลังจากกด Next แล้วจะมีหน้าต่างต่อไปให้ใส่เลขหมายไอพีแอดเดรส และความลับร่วม ดังรูป



รูปที่ 13-9 แสดงหน้าต่างใส่ไอพีแอดเดรสและความลับร่วม

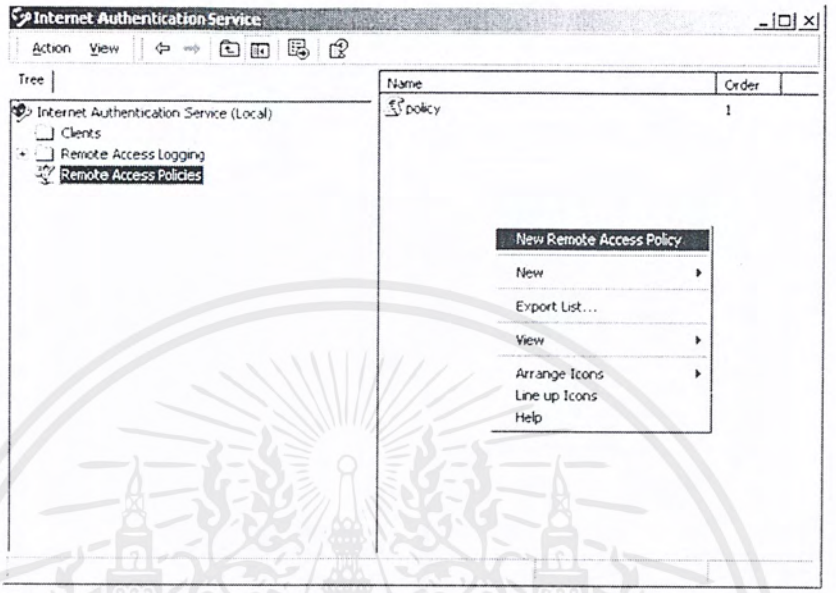
เมื่อสำเร็จทุกขั้นตอนแล้ว ก็จะได้ RADIUS ไคลเอ็นต์ ดังรูป



รูปที่ 13-10 แสดงหน้าจอ RADIUS ไคลเอ็นต์

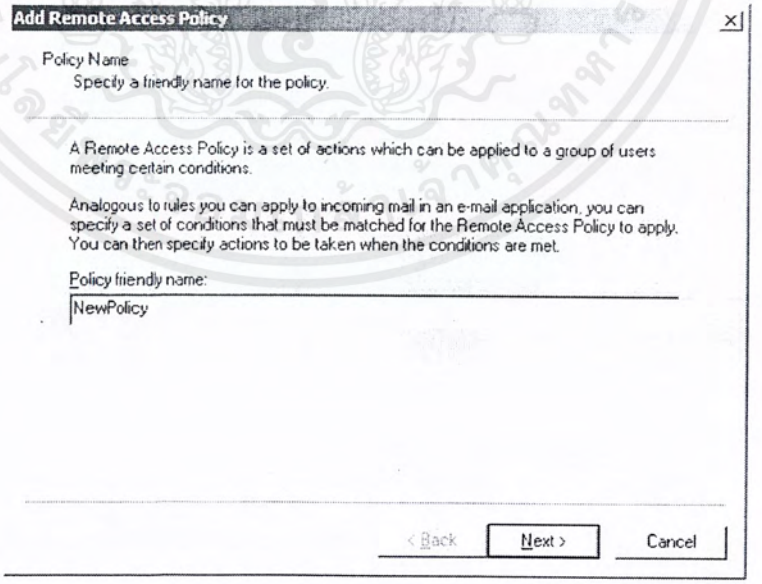
โดยในรูปนั้น แสดง RADIUS ไคลเอ็นต์ที่มีชื่อว่า Isag28 โดยมีไอพีแอดเดรสคือ 161.246.5.28 และใช้ RADIUS โพรโตคอลในการติดต่อระหว่าง RADIUS เซิร์ฟเวอร์ และ RADIUS ไคลเอ็นต์

ต่อไปเป็นการกำหนดข้อจำกัดในการติดต่อของผู้ใช้ระบบจากภายนอกโดยการแก้ไข Policies โดยสามารถเพิ่ม Policies ได้ด้วยการเลือกที่ Remote Access Policies และที่หน้าต่างด้านขวาคลิ้กขวาและเลือก New Remote Access Policies ดังรูป



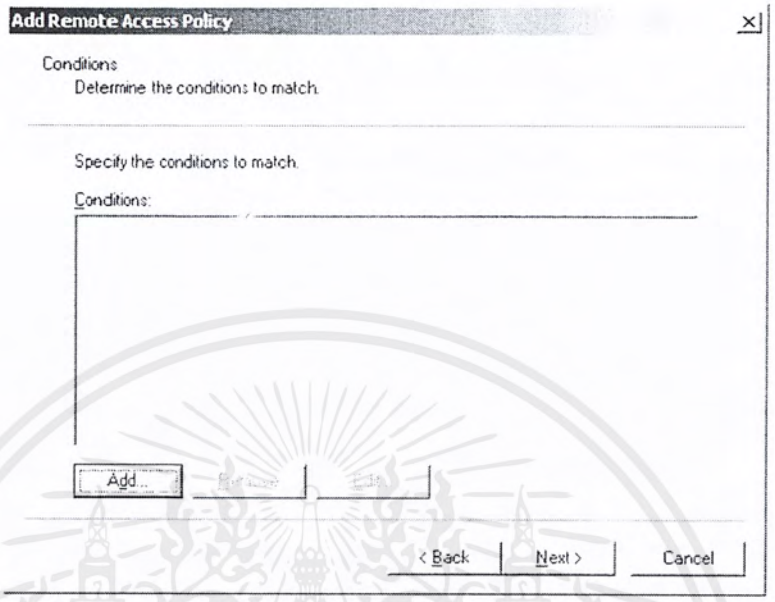
รูปที่ 13-11 แสดงการเพิ่ม Policy

หลังจากนั้นจะเกิดหน้าต่างใหม่ เพื่อให้ใส่ชื่อของ Policy นั้น โดยในที่นี้ใส่ชื่อว่า NewPolicy ดังรูป



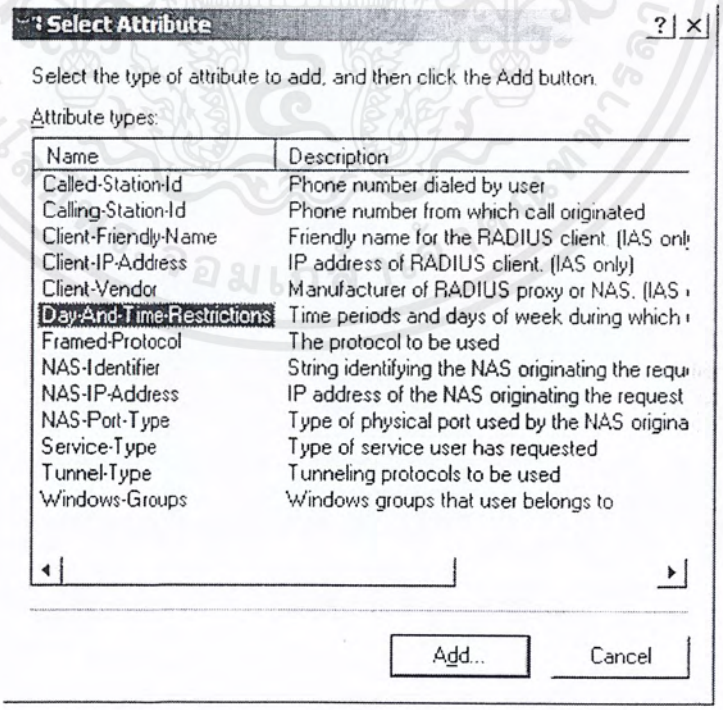
รูปที่ 13-12 แสดงหน้าจอใส่ชื่อของ Policy

หลังจากใส่ชื่อเรียบร้อยแล้ว จะเป็นการใส่ข้อจำกัดในการติดต่อจากภายนอกว่าต้องมีเงื่อนไขใดบ้าง ดังรูป



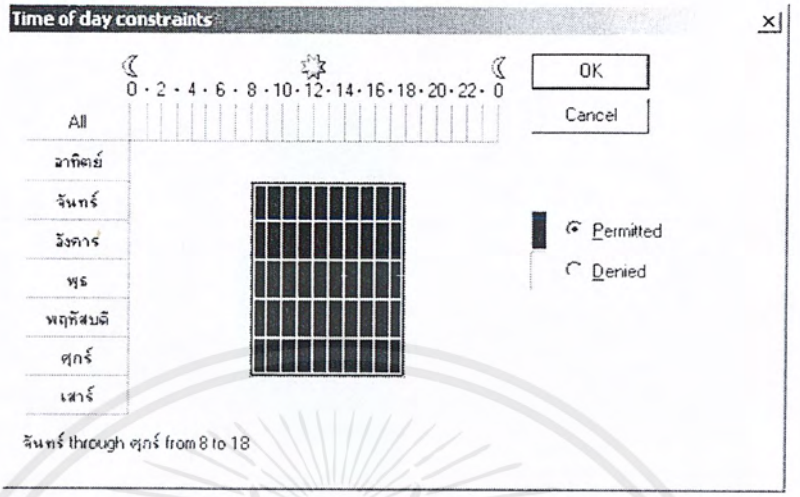
รูปที่ 13-13 แสดงหน้าจอการเพิ่ม ลบ หรือแก้ไข เงื่อนไขต่างๆ

ในที่นี้จะเพิ่ม Policy โดยเลือกเงื่อนไขเกี่ยวกับวันและเวลาในการติดต่อจากผู้ใช้ระบบ



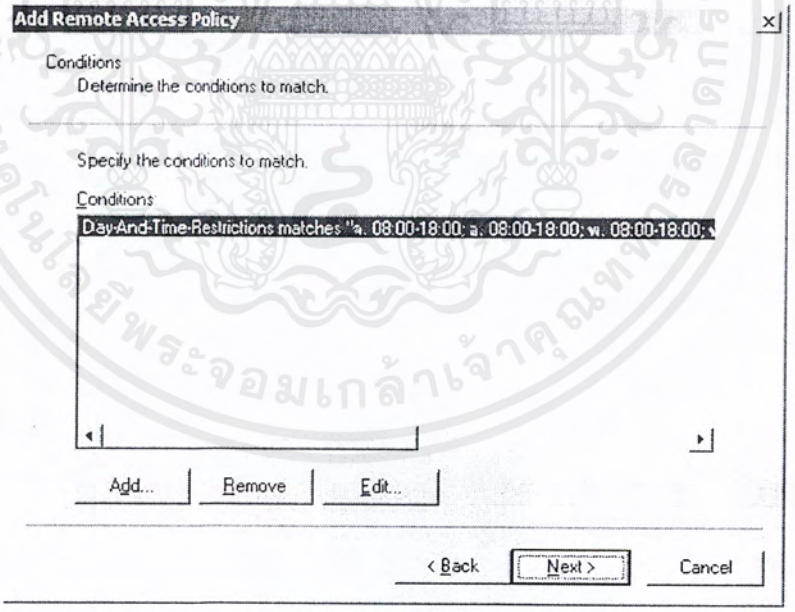
รูปที่ 13-14 การเลือกเงื่อนไขต่างๆ เพื่ออนุญาตการติดต่อของผู้ใช้ระบบ

เมื่อเลือกเงื่อนไขที่เกี่ยวกับวันและเวลา จะมีหน้าต่างใหม่ขึ้นมาถามถึงวันและเวลาที่ต้องการอนุญาตหรือจำกัดในการติดต่อจากผู้ใช้ระบบ ในที่นี้ได้เลือก วันจันทร์ถึงศุกร์ เวลา 8 นาฬิกาถึง 18 นาฬิกา



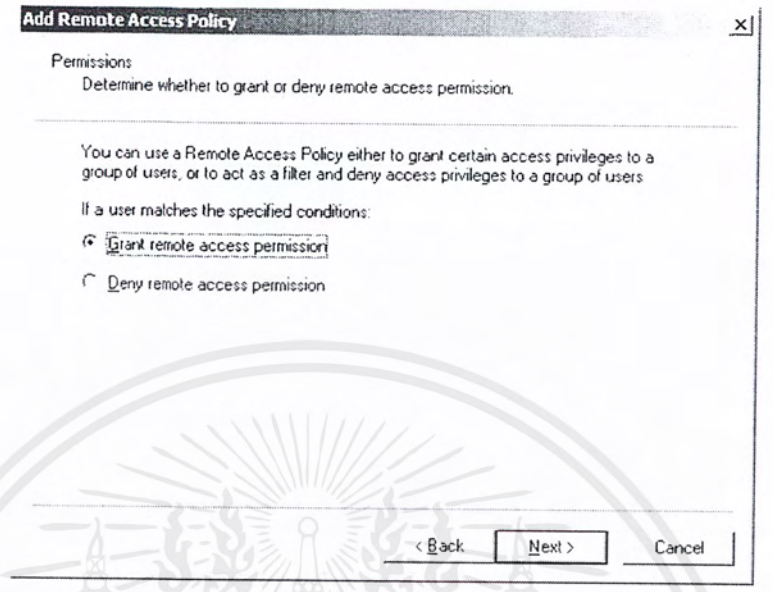
รูปที่ 13-15 แสดงหน้าจอเลือกวันและเวลาในการอนุญาตหรือจำกัดการติดต่อ

หลังจากเลือกเงื่อนไขเรียบร้อยแล้ว จะมีหน้าจอใหม่ขึ้นมาแสดงเงื่อนไขที่ได้กำหนดไว้ ดังรูป



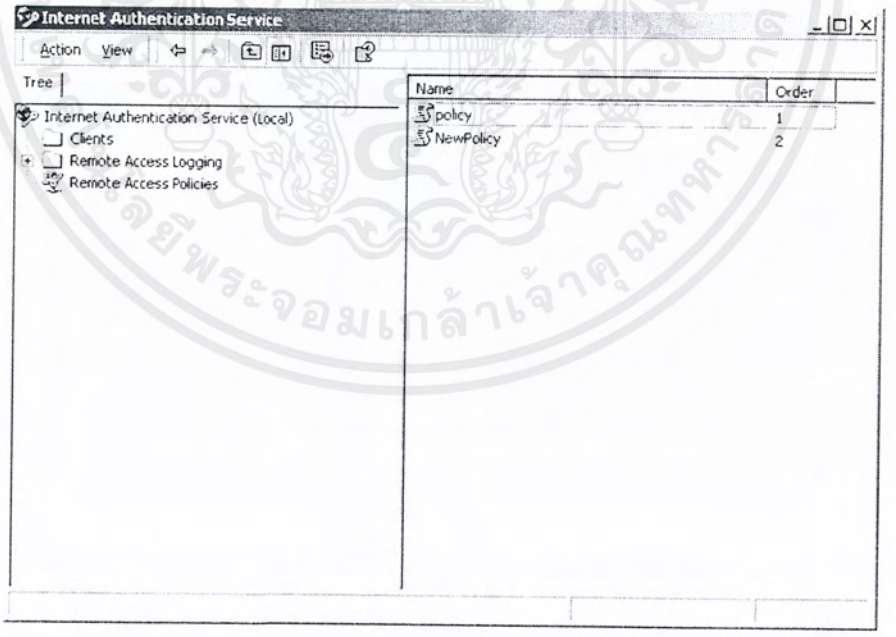
รูปที่ 13-16 แสดงหน้าจอแสดงเงื่อนไขที่ได้กำหนดไว้

หลังจากนั้นจะมีหน้าต่างใหม่ขึ้นมาถามว่าจากเงื่อนไขที่ได้เลือกนั้น ถ้าตรงตามเงื่อนไข จะอนุญาตให้ติดต่อหรือปฏิเสธการติดต่อ



รูปที่ 13-17 แสดงหน้าจอถามอนุญาตหรือไม่ ถ้าตรงตามเงื่อนไขที่กำหนด

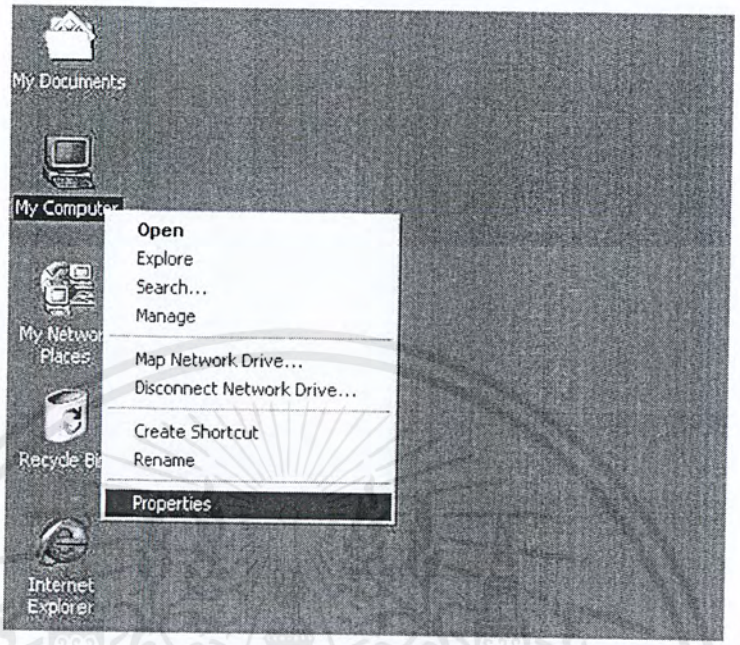
เมื่อสำเร็จทุกขั้นตอน จะได้ policy ใหม่ดังรูป



รูปที่ 13-18 แสดงผลลัพธ์จากการสร้าง policy

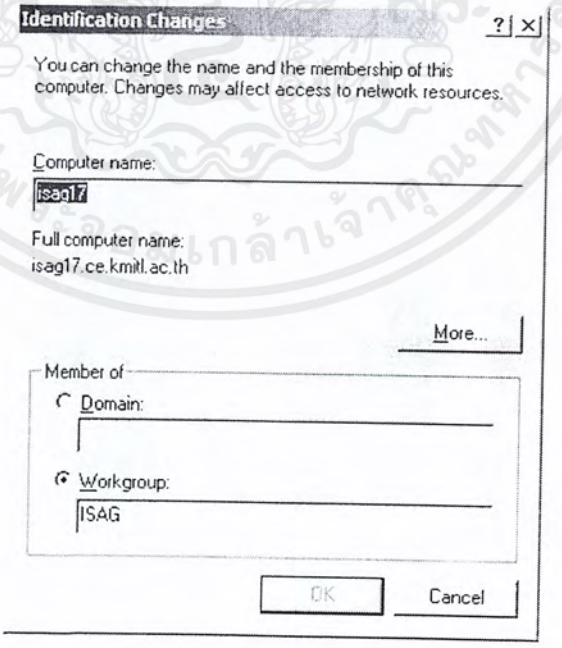
10.3 การ Join Domain

เริ่มต้นจากที่เดสก์ทอปของวินโดวส์ โดยคลิ้กขวาที่ไอคอน My Computer แล้วเลือก Properties ดังรูป



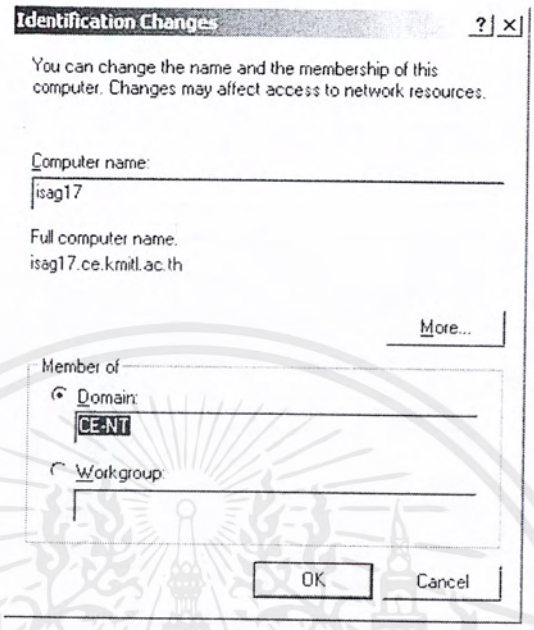
รูปที่ 13-19 แสดงหน้าจอเดสก์ทอปของวินโดวส์

หลังจากนั้นเลือกแท็ป Network Identification แล้วกดปุ่ม Propereties จะมีหน้าจอแสดงดังรูป



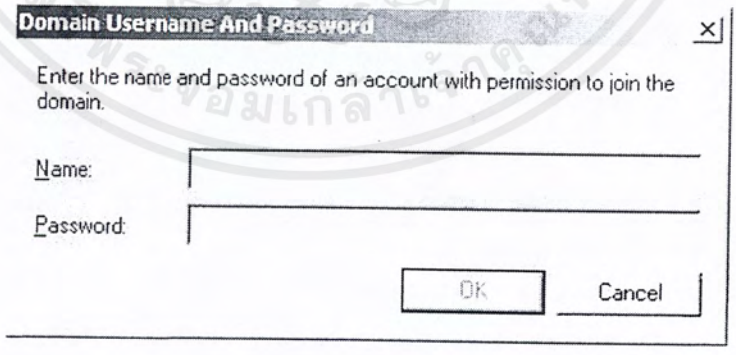
รูปที่ 13-20 แสดงหน้าต่างเปลี่ยนแปลง Identification ของคอมพิวเตอร์

จากรูปเป็นกรณีที่ปัจจุบัน เครื่องนี้ชื่อ isag17 อยู่ในเวิร์กกรุ๊ป ISAG ดังนั้นจึงไม่ได้อยู่ในโดเมนใดเลย แต่เมื่อต้องการ Join Domain จึงเปลี่ยนจากเวิร์กกรุ๊ปมาอยู่ในโดเมนแทน ซึ่งในที่นี้ชื่อ CE-NT



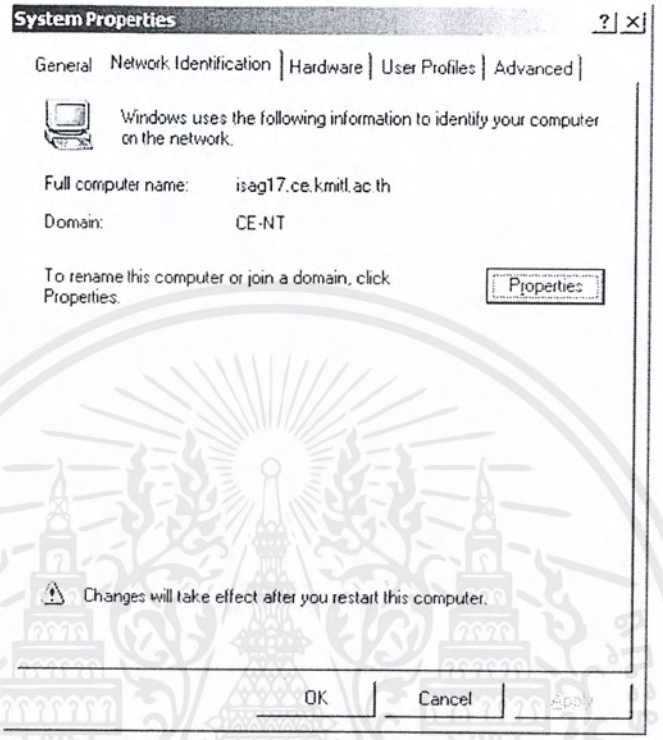
รูปที่ 13-21 แสดงการเปลี่ยนจากอยู่ในเวิร์กกรุ๊ปเป็นอยู่ในโดเมน

เมื่อเปลี่ยนไปอยู่ในโดเมนเรียบร้อยแล้ว วินโดวส์จะถามถึงรหัสผ่านของผู้ดูแลระบบ ของโดเมน (ซึ่งในที่นี้คือ CE-NT) เพื่อเป็นการยืนยันจากผู้ดูแลระบบของโดเมนนั้นว่าอนุญาตให้เครื่องนี้สามารถเป็นส่วนหนึ่งของโดเมนนั้นได้



รูปที่ 13-22 แสดงหน้าจอให้ใส่ชื่อและรหัสผ่านของผู้ดูแลระบบของโดเมน

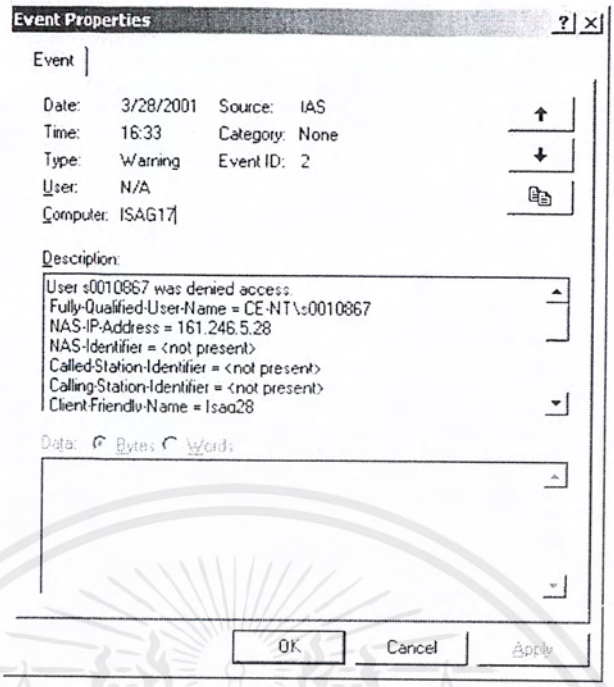
หลังจากที่ผู้ดูแลระบบของโดเมนได้ใส่ชื่อและรหัสผ่านเป็นที่เรียบร้อยแล้ว ที่แท็ป Network Identification ก็จะแสดงข้อความใหม่ที่ได้เปลี่ยนแปลงไป และมีคำเตือนให้รีสตาร์ทเครื่องใหม่ หลังจากที่รีสตาร์ทเครื่องใหม่ การ Join Domain ก็เสร็จสมบูรณ์



รูปที่ 13-23 แสดงหน้าจอที่แท็ป Network Identification หลังจากที่เปลี่ยนแปลงแล้ว

13.4 ข้อความที่แสดงเมื่อมีการติดต่อมายัง RADIUS เซิร์ฟเวอร์

เมื่อมีการติดต่อจากไคลเอ็นต์เข้ามายัง RADIUS ไคลเอ็นต์แล้ว RADIUS ไคลเอ็นต์จะติดต่อไปยัง RADIUS เซิร์ฟเวอร์ เพื่อให้ RADIUS เซิร์ฟเวอร์ตอบรับหรือปฏิเสธการติดต่อ โดยสามารถดูข้อความเหล่านี้ได้จาก Event Viewer โดยอยู่ใน Administrator tools โดยเมื่อมีการติดต่อเข้ามาจากไคลเอ็นต์และไม่สำเร็จเนื่องจากผู้ที่ติดต่อเข้ามาใส่ชื่อหรือรหัสผ่านผิด จะได้ข้อความดังรูป



รูปที่ 13-24 แสดงหน้าจอข้อความบอกถึงมีผู้ติดต่อเข้ามาแต่ปฏิเสธเนื่องจากชื่อหรือรหัสผ่านผิด

โดยข้อความทั้งหมดเป็นดังนี้

Event Type: Warning

Event Source: IAS

Event Category: None

Event ID: 2

Date: 3/28/2001

Time: 4:33:14 PM

User: N/A

Computer: ISAG17

Description:

User s0010867 was denied access.

Fully-Qualified-User-Name = CE-NT\s0010867

NAS-IP-Address = 161.246.5.28

NAS-Identifier = <not present>

Called-Station-Identifier = <not present>

Calling-Station-Identifier = <not present>

Client-Friendly-Name = Isag28

Client-IP-Address = 161.246.5.28

NAS-Port-Type = Async

NAS-Port = 0

Policy-Name = <undetermined>

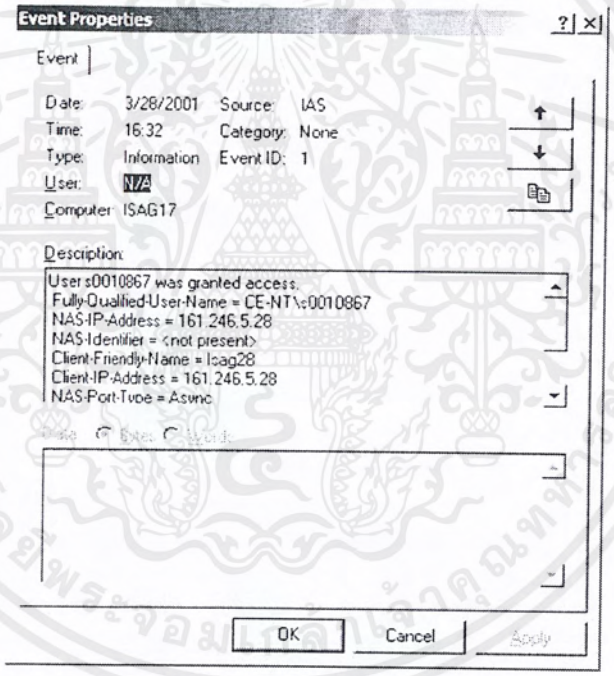
Authentication-Type = <undetermined>

EAP-Type = <undetermined>

Reason-Code = 16

Reason = There was an authentication failure because of an unknown user name or a bad password.

โดยรูปแบบของข้อความจะมีชนิดเป็น Warning หรือคำเตือน แต่ถ้ามีผู้ติดต่อเข้ามาใส่ชื่อและรหัสผ่านถูกต้องและ RADIUS เซิร์ฟเวอร์ให้บริการจะมีข้อความดังรูป



รูปที่ 13-25 แสดงหน้าจอข้อความบอกถึงมีผู้ติดต่อเข้ามาและ RADIUS เซิร์ฟเวอร์ให้บริการ

โดยข้อความทั้งหมดเป็นดังนี้

Event Type: Information

Event Source: IAS

Event Category: None

Event ID: 1

Date: 3/28/2001

Time: 4:32:28 PM

User: N/A

Computer: ISAG17

Description:

User s0010867 was granted access.

Fully-Qualified-User-Name = CE-NT\s0010867

NAS-IP-Address = 161.246.5.28

NAS-Identifier = <not present>

Client-Friendly-Name = Isag28

Client-IP-Address = 161.246.5.28

NAS-Port-Type = Async

NAS-Port = 0

Policy-Name = policy

Authentication-Type = PAP

EAP-Type = <undetermined>

โดยข้อความที่เกิดขึ้นเป็นข้อความชนิด Information หรือข้อมูล

บรรณานุกรม

[1]. "Kerberos", *MIT,* rfc1510

[2]. "Passwd and Shadow file", http://www.cs.ust.hk/~wongk/solaris/node40.html

[3]. "Crack Password", http://www.hackersclub.com/km/newbies

[4]. "UNIX Authentication tools", http://ciac.llnl.gov/ciac/ToolsUnixAuth.html

[5]. "Network Information Services (NIS and NIS+) Guide", http://sp-www.iu.edu/doc_link/en_US/a_doc_lib/aixbman/nisplus/frame10_toc.htm

[6]. "RADIUS Whitepaper",http://www.livingston.com/marketing/whitepapers/radius_paper.html

[7] "RADIUS", rfc2138

[8] "LDAP",rfc1777

[9] "An Introduction to LDAP", http://ldapman.org/articles/intro_to_ldap.html

[10] "LDAPman RFC page", http://www.ldapman.org/ldap_rfcs.html

[11] "Outline: Introduction to LDAP" , http://ldap-project.berkeley.edu/reports/introduction/LDAPintro.html

[12] "ciscopress.com" , http://www.knowcisco.com/matter/tut1000006/

[13] "LDAP Documentation", http://www.umich.edu/~dirsvcs/ldap/doc/

[14] "LDAP RFC", http://umich.edu/~dirsvcs/ldap/doc/rfc/rfc1777.txt

[15] "How SSL Works", http://developer.netscape.com/tech/security/ssl/howitworks.html

[16] "Single Sing-On Deployment Guide", http://developer.netscape.com/docs/manuals/security/SSO/contents.htm