**บทที่ 2**

**หลักการและทฤษฏีที่เกี่ยวข้อง**

การปรับปรุงโมดูลยืนยันตัวตนเพื่อเข้าใช้ระบบจัดการการส่งการบ้านปฏิบัติการ ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยเชียงใหม่ โดยการปรับปรุงใหม่นั้นจำเป็นต้องใช้บัญชีผู้ใช้ไอทีของทางมหาวิทยาลัย ทำให้ต้องปรับปรุงทางด้านความปลอดภัยต่อการใช้งานของนักศึกษา อาจารย์ และ ผู้ดูแลระบบโดยการใช้เอสเอสแอล (SSL) ในการแลกเปลี่ยนข้อมูล เพื่อความปลอดภัยสูง ตรวจความถูกต้องของบัญชีผ่านเครื่องแม่ข่ายเรเดียส (RADIUS) และรับข้อมูลรายละเอียดนักศึกษาได้รับจากเอพีไอ (API) ของทางสำนักบริการเทคโนโลยีสารสนเทศมหาวิทยาลัยเชียงใหม่

**2.1 ทีแอลเอส (TLS) หรือชื่อเดิม เอสเอสแอล**

เอสเอสแอลโพรโทคอลรักษาความปลอดภัย เพื่อให้โปรแกรมบนเครื่องแม่ข่ายและเครื่องลูกข่ายเชื่อมต่อส่งผ่านข้อมูลระหว่างเครือข่ายอย่างปลอดภัย โดยถูกออกแบบมาเพื่อป้องกันการถูกดักจับข้อมูล   
 การใช้งานจริง ทีแอลเอส หรือ เอสเอสแอล ถูกนำมาประยุกต์กับโพรโทคอลในชั้นสื่อสารทรานสปอร์ต (Transport Layer Protocols) ห่อหุ้มโพรโทคอลที่ใช้งาน เช่น เฮสทีทีพี (HTTP) เอฟทีพี (FTP) เอสเอมทีพี (SMTP) โดยปกติมักถูกใช้บนเว็บที่ต้องการความปลอดภัยเข้าใช้งานของผู้ใช้งาน และเว็บซื้อขายบนอินเทอร์เน็ต  
 เครื่องแม่ข่ายและลูกข่ายจะสามารถใช้เอสเอสแอล ได้เมื่อมีการตกลงที่จะใช้เหมือนกัน คือ วิธีการจับมือกัน (Handshake) โดยระหว่างการจับมือทั้งเครื่องแม่ข่ายและลูกข่ายต้องตกลงกันในแต่ละเงื่อนไขเพื่อความปลอดภัยของการเชื่อมต่อโดยมีขั้นตอนดังนี้

1. เครื่องลูกข่ายส่ง หมายเลขรุ่นของเอสเอสแอล รายละเอียดการตั้งค่าเข้ารหัส (Cipher settings) ข้อมูล   
   ของช่องสื่อสาร (Session) ที่ได้รับ และข้อมูลจำเป็นอื่นที่แม่ข่ายต้องใช้ติดต่อกับเครื่องลูกข่ายผ่าน  
   เอสเอสแอล
2. เครื่องแม่ข่ายส่ง หมายเลขรุ่นของเอสเอสแอล รายละเอียดการตั้งค่าเข้ารหัส ข้อมูลของช่องสื่อสาร ที่ส่งไป ข้อมูลอื่นที่ลูกข่ายต้องใช้ติดต่อกับเครื่องแม่ข่ายผ่านเอสเอสแอล ทั้งส่งหนังสือรับรอง (Certificate) ของแม่ข่าย และถ้าหากเครื่องลูกข่ายร้องขอทรัพยากรของแม่ข่ายเพื่อการยืนยันตัวตนบนระบบ เครื่องแม่ข่ายจะร้องขอหนังสือรับรอง จากเครื่องลูกข่ายด้วย
3. เครื่องลูกข่ายใช้ข้อมูลที่ได้มาจากแม่ข่ายเพื่อยืนยันตัวตนเครื่องแม่ข่าย ยกตัวอย่างเช่น เว็บเบราว์เซอร์   
   (Web Browser) เชื่อมต่อกับเครื่องแม่ข่าย ตรวจสอบหนังสือรับรองที่ได้รับว่าชื่อแม่ข่ายตรงกับที่ได้รับ   
   เชื่อถือหนังสือรับรองได้หรือไม่ หนังสือรับรองหมดอายุแล้วหรือไม่ ถ้าไม่สามารถยืนยันเครื่องแม่ข่ายได้  
   จึงเตือนผู้ใช้ว่าเชื่อมต่อไม่ได้ หรือ สามารถยืนยันแม่ข่ายได้ถูกต้องก็ดำเนินการขั้นต่อไป
4. จากข้อมูลทั้งหมดในกระบวนการจับมือ เครื่องลูกข่ายต้องสร้างรหัสลับพื้นฐาน (Pre-master secret)   
   สำหรับช่องสื่อสารปัจจุบัน เข้ารหัสด้วยกุญแจสาธารณะ (Public Key) ที่ได้จากเครื่องแม่ข่ายในขั้นตอน  
   ที่ 2 จากนั้นจึงส่งรหัสลับพื้นฐานที่ถูกเข้ารหัสให้กับแม่ข่าย
5. (กรณีเครื่องแม่ข่ายร้องขอการยืนยันตัวตนจากลูกข่าย) เครื่องลูกค่ายต้องสร้างข้อมูลอื่นซึ่งจะเป็นที่เด่นชัดว่าเป็นเครื่องที่ควรส่งข้อมูลหากันอยู่จริง ซึ่งกรณีนี้เครื่องลูกข่ายต้องส่งเอกสารใหม่ พร้อมใบรับรองเครื่อง และส่งรหัสลับพื้นฐานที่ถูกเข้ารหัสให้กับแม่ข่าย
6. (กรณีเครื่องแม่ข่ายร้องขอการยืนยันตัวตนจากลูกข่าย) เครื่องแม่ข่ายยืนยันลูกข่ายว่าเข้าใช้งานได้
   1. ถ้าไม่สามารถยืนยันตัวตนของลูกข่ายให้ทำการปิดช่องสื่อสารกับลูกข่าย
   2. ถ้ายืนยันลูกข่ายได้สำเร็จ เครื่องแม่ข่ายจึงทำการถอดรหัสลับพื้นฐานที่ได้โดยใช้กุญแจลับ (Private Key) ที่มีเฉพาะเครื่องแม่ข่าย และทำงานพร้อมเครื่องลูกข่ายเพื่อสร้าง รหัสลับหลัก   
      (Master Secret)
7. ทั้งเครื่องแม่ข่ายและลูกข่ายใช้รหัสลับหลักเพื่อสร้าง กุญแจสื่อสาร (Session Key) ซึ่งเป็นกุญแจใช้ร่วมกันของแม่ข่ายและลูกข่าย เพื่อการเข้ารหัสและถอดรหัสข้อมูลสื่อสารแลกเปลี่ยนระหว่างเปิดเอสเอสแอล และเพื่อยืนยันความเป็นบูรณภาพของข้อมูล
8. เครื่องลูกข่ายส่งข้อความให้แก่เครื่องแม่ข่ายเพื่อบอกว่าในอนาคตข้อมูลจะถูกส่งโดยถูกเข้ารหัสด้วยกุญแจสื่อสาร จากนั้นจึงบอกว่าการจับมือด้านเครื่องลูกข่ายได้เสร็จสิ้น
9. เครื่องแม่ข่ายส่งข้อความให้แก่เครื่องลูกข่ายเพื่อบอกว่าในอนาคตข้อมูลจะถูกส่งโดยถูกเข้ารหัสด้วยกุญแจสื่อสาร จากนั้นจึงบอกว่าการจับมือด้านเครื่องแม่ข่ายได้เสร็จสิ้น



**รูปที่ 2.1 แสดงการส่งข้อมูลบนทีแอลเอสระหว่างลูกข่ายและแม่ข่าย**

หมายเหตุ

* ถึงขั้นตอนสุดท้ายถือเป็นการเสร็จสิ้นของการจับมือของเอสเอสแอล และเริ่มการเปิดช่องการสื่อสาร ทั้งสองด้านใช้กุญแจสื่อสารเข้ารหัสและแกะรหัสข้อมูล
* ถ้ามีขั้นตอนด้านบนขั้นตอนใดล้มเหลว จะไม่มีการเชื่อมต่อระหว่างสองฝั่ง

**2.2 เรเดียส (RADIUS)**

เรเดียส ย่อมาจาก Remote Access Dial In User Service เป็นโพรโทคอลเครือข่ายที่ให้การตรวจสอบ, อนุมัติ และการจัดการการบัญชี (AAA) จากส่วนกลาง สำหรับคอมพิวเตอร์ที่เชื่อมต่อและใช้บริการเครือข่าย   
 เรเดียสเป็นโพรโทคอลแบบไคลเอ็นต์/เซิร์ฟเวอร์ที่วิ่งในชั้นแอพพลิเคชัน ใช้ยูดีพีเป็นช่องทางขนส่ง  
หลักการของเอเอเอ

เอเอเอ ย่อมาจาก การตรวจสอบ (Authentication),การอนุมัติ (Authorization) และการจัดการการบัญชี (Accounting) เป็นโครงสร้างความปลอดภัยสำหรับระบบที่กระจายการบริการใช้งานเพื่อควบคุมผู้ใช้ที่จะเข้าใช้งาน  
 การตรวจสอบ คือ การที่ผู้ใช้ที่เข้าใช้ระบบนั้นเป็นผู้ใช้จริงหรือไม่

การอนุมัติ คือ การอนุญาตว่าผู้ใช้แต่ละคนสามารถใช้ทรัพยากรได้อย่างไร ใช้ระบบได้นานเท่าไหร่ ฯลฯ

การจัดการการบัญชี คือ การเก็บข้อมูลการใช้งานของผู้ใช้ เวลาเริ่มใช้งาน เวลาที่เลิกใช้งาน จำนวนทรัพยากร ระยะเวลา ระหว่างที่ใช้งานระบบ

**2.3 คำสั่งบนเทอร์มินอลที่จำเป็นสำหรับลินุกซ์**

[ ] = ไม่จำเป็นต้องมีก็ได้ เป็นส่วนเสริมเพิ่มกับคำสั่ง

**2.3.1 คำสั่งจัดการกับไฟล์**

1. เปลี่ยนที่อยู่โฟลเดอร์กำลังทำงาน
2. **cd [*options(s)*] [*directory*]**

เช่น cd /etc เข้าไปยังโฟลเดอร์หลักชื่อ etc

1. แสดงไฟล์ในไดเรคทอรี่ (Directory) ปัจจุบัน

**ls [*option(s)*] [*file(s)*]**

เช่น ls –a แสดงไฟล์ทั้งหมดในไดเรคทอรี่ปัจจุบันรวมทั้งไฟล์ที่ซ่อนไว้

1. ทำการลบไฟล์ในระบบ

**rm [*option(s)*] *file(s)***

เช่น rm text.txt ทำการลบไฟล์ที่ชื่อว่า text.txt

1. คัดลอกไฟล์บนระบบ

**cp [*option(s)*] *sourcefile targetfile***

เช่น cp ./myfolder/thisfile /etc/ คัดลอกไฟล์จาก myfolder ชื่อไฟล์ thisfile ไปยังโฟลเดอร์ etc

1. ย้ายไฟล์บนระบบ

**mv [*option(s)*] *sourcefile targetfile***

1. สร้างโฟล์เดอร์บนระบบ

**mkdir [*option(s)*] *directoryname***

เช่น mkdir newfolder สร้างโฟลเดอร์ใหม่ชื่อ newfolder

1. ลบโฟล์เดอร์บนระบบ

**rmdir [*option(s)*] *directoryname***

เช่น rmdir newfolder ลบโฟลเดอร์ชื่อ newfolder

1. โอนย้ายความเป็นเจ้าของไฟล์ให้ผู้ใช้

**chown [*option(s)*] *username.group file(s)***

เช่น chown wwwdata config.conf โอนย้ายให้ผู้ใช้ wwwdata สามารถใช้ไฟล์ config.conf ได้

1. เปลี่ยนการอนุญาตเข้าใช้งานไฟล์

**chmod [*options*] *mode file(s)***เช่น chmod u=rw,g=r,o= internalPlan.txt ผู้ใช้เจ้าของไฟล์แก้ไขและอ่านไฟล์ได้ กลุ่มสามารถอ่านได้นอกเหนือจากนั้นไม่สามารถเข้าถึงไฟล์ได้

**2.3.2 คำสั่งทำงานกับข้อมูลอักขระภายในไฟล์**

1. แสดงรายละเอียดภายในไฟล์

**cat [*option(s)*] *file(s)***

1. โปรแกรมเพื่อใช้เปิดดูไฟล์

**less [*option(s)*] *file(s)***

PgUp - PgDown เลื่อนดูข้อความที่ละครึ่งจอ ขึ้น-ลง

Space เพื่อเลื่อนลงเต็มจอ

Home – End เลื่อนไปที่บนสุด – ท้ายสุดของไฟล์

Q เพื่อออกโปรแกรม

1. การค้นหาคำเฉพาะในไฟล์

**grep [*option(s)*] *searchstring filenames***

ผลลัพธ์ที่ได้คือข้อความทั้งบรรทัดที่มีคำเฉพาะนั้นอยู่ สามารถค้นหาโดยใช้นิพจน์ปกติ (Regular Expression) สามารถใช้ค้นหาหลังจากได้ผลลัพธ์จากคำสั่งอื่น เช่น cat studentname.txt | grep suchat โดยระบบจะใช้ข้อความนำเข้าแทนไฟล์นำเข้า

1. เปรียบเทียบข้อมูลอักขระภายในไฟล์

**diff [*option(s)*] *file1 file2***

ผลลัพธ์ที่ได้คือการแสดงอักขระในบรรทัดที่ไม่เหมือนกัน

1. โปรแกรมสำหรับแก้ไขไฟล์

**vi *filename***

โปรแกรม vi ทุกชื่อโหมดการทำงานไม่สามารถใช้พิมพ์ใหญ่แทนพิมพ์เล็ก หรือเล็กแทนใหญ่ก็ไม่ได้ โดยแสดงที่กำลังทำงานอยู่ในโหมดทางด้านล่างซ้าย การบันทึกและออกจากไฟล์ให้ใช้ **:** (semi-colon) นำหน้าคำสั่งนั้น

|  |  |
| --- | --- |
| ESC | ออกจากทุกโหมดที่กำลังทำงาน |
| i | เข้าสู่โหมดแก้ไขข้อความในไฟล์ |
| u | ย้อนกลับการแก้ไขล่าสุด |
| dd | ลบทั้งบรรทัดที่ตำแหน่งตัวชี้อยู่ |
| :q! | ออกจากไฟล์โดยไม่สนใจการเปลี่ยนแปลงใดๆ |
| :w [*filename*] | บันทึกไฟล์ |
| :x | บันทึกการแก้ไขล่าสุดและออกจากโปรแกรม |

**ตารางที่ 2.1 ตารางแสดงคำสั่งบนโปรแกรม vi**

**2.3.3 คำสั่งทำงานกับระบบปฏิบัติการ**

1. แสดงข้อมูล รายละเอียดของหน่วยความจำสำรอง

**df [*option(s)*] [*directory*]**

เมื่อใช้โดยไม่เติม พารามิเตอร์จะเป็นการแสดงพื้นที่ทั้งหมด ส่วนที่ถูกใช้ และส่วนที่ว่างต่อการใช้งาน

1. แสดงข้อมูลโปรแกรมที่กำลังทำงาน หน่วยประมวลผลทำงานที่อยู่

**top [*options(s)*]**

1. แสดงข้อมูลโปรแกรมที่กำลังทำงาน หน่วยประมวลผลทำงานที่อยู่ อย่างละเอียด

**ps [*option(s)*] [*process ID*]**

1. การปิดโปรแกรมที่ไม่สามารถปิดด้วยวิธีธรรมดา

**kill [*option(s*)] *process ID***

**killall [*option(s)*] *processname***

โดย process ID หาได้จากการเรียก ps ส่วนการเรียกโดย killall จะใช้ชื่อโปรแกรมที่ต้องการหยุดโดยทุกหน่วยที่มีชื่อตาม processname จะถูกหยุดทำงานทั้งหมด

1. การทดสอบการเชื่อมต่อของเครือข่าย

**ping [*option(s)*] *host name*** *|* ***IP address***

โดยใช้ชื่อเครื่องเป้าหมาย หรือหมายเลขไอพี เพื่อทดสอบการเชื่อมต่อ ถ้าไม่ใส่ค่าจำนวนครั้งจะไม่หยุดการทำงานจนกว่าจะกด Ctrl+C

**2.3.4 คำสั่งเสริมที่จำเป็น**

1. การเปลี่ยนรหัสเข้าใช้งาน

**passwd [*option(s)*] [*username*]**

ผู้ใช้งานที่เรียกคำสั่งสามารถเปลี่ยนรหัสเฉพาะของตัวเองเท่านั้น แต่สิทธิผู้ดูแลระบบ root สามารถเปลี่ยนรหัสให้กับผู้ใช้ทุกคน

1. การเปลี่ยนการใช้งานเป็นผู้ใช้งานอื่น

**su [*option(s)*] [*username*]**

โดยสามารถเรียกได้จากผู้ใช้ทุกคน ถ้าไม่มีการใส่ username จะเป็นการเรียกใช้ผู้ดูแลระบบ root โดยจะต้องมีรหัสเข้าใช้งาน root เท่านั้น

1. คำสั่งหยุดการใช้งานระบบ

**halt [*option(s)*]**

เพื่อป้องกันความเสียหายต่อข้อมูล halt จะช่วยปิดโปรแกรมและหยุดระบบเพื่อปิดใช้งานเครื่องด้วยให้เติม –p เพื่อให้ปิดกระแสไฟในเครื่องอัตโนมัติ

1. คำสั่งให้เครื่องเริ่มต้นทำงานใหม่

**reboot [*option(s)*]**

1. การลบทุกอักขระบนหน้าจอผู้ใช้

**clear**

1. การเรียกคำสั่งโดยผู้ใช้ที่ไม่ใช่ผู้ดูแลระบบ แต่ต้องใช้สิทธิผู้ดูแลระบบ

**sudo [*commands*]**

ผู้ใช้ที่สามารถเรียก sudo ต้องมีการบันทึกชื่อไว้ภายในไฟล์ /usr/local/etc/sudoers (บางครั้ง /etc/sudoers) โดยปกติจะใช้รหัสของผู้ใช้ที่เรียก แต่สามารถเปลี่ยนให้ใช้รหัสผู้ดูแลระบบเท่านั้นได้โดยเติมคำสั่ง Defaults rootpw

1. แสดงตำแหน่งที่อยู่เต็มของโปรแกรม

**which [ *options* ] *programname***

โดย programname เป็นชื่อโปรแกรมที่ได้ติดตั้งในระบบแล้ว

**2.4 ซิงเกิลไซน์ออน (Single Sign-on)**

**2.4.1 รูปแบบของซิงเกิลไซน์ออน**

1. **ซิงเกิลไซน์ออนแบบดั้งเดิม**

การพิสูจน์ตัวจริงแบบดั้งเดิมที่ใช้ในระบบเก่ามีดังนี้

1. ผู้ใช้ใส่ข้อมูลผู้ใช้งาน
2. ระบบยืนยันความถูกต้อง ถ้าผ่านข้อมูลถูกจัดเก็บไว้ในคุกกี้
3. เมื่อผู้ใช้ลงบันทึกเข้า ข้อมูลผู้ใช้ที่เก็บไว้ในคุกกี้ถูกตรวจสอบ

อย่างไรก็ตามซิงเกิลไซน์ออน ในรูปแบบนี้ต้องทำให้มั่นใจว่าข้อมูลผู้ใช้เป็นเอกลักษณ์กับแต่ละระบบ ทำให้เป็นการสิ้นเปลืองทรัพยากรระบบ และเกิดช่องโหว่ความปลอดภัยเมื่อข้อมูลภายในคุกกี้ถูกอ่านได้

1. **ซิงเกิลไซน์ออนบนพื้นฐานเคอร์เบอร์รอส (Kerberos-based SSO)**

เคอร์เบอร์รอส เป็นมาตรฐานส่งผ่านกุญแจ นิยามโดยคณะทำงานเฉพาะกิจด้านวิศวกรรมอินเทอร์เน็ต (IETF) โดยซิงเกิลไซน์ออนบนพื้นฐานเคอร์เบอร์รอส ถูกนำมาใช้งานจริงโดยมี ศูนย์กลางกระจายกุญแจ (KDC) ที่ทำหน้าที่กระจาย ตั๋วและ พิสูจน์ตัวจริงผู้ใช้ โดยแกนหลักการทำงานคือ ตั๋ว และหลักการทำงาน คือ

1. ผู้ใช้พิสูจน์ตัวจริงโดยส่งข้อมูลผู้ใช้ให้ ศูนย์กลางกระจายกุญแจ ถ้าผู้ใช้ลงทะเบียนเป็นครั้งแรก บัตรผ่านแรก (Ticket-granting Ticket, TGT) ถูกส่งไปให้ผู้ใช้
2. เมื่อศูนย์กลางกระจายกุญแจ ได้รับการร้องขอ บัตรผ่านขอใช้บริการ (Service Ticket , ST) พร้อมรับ บัตรผ่านแรกจากผู้ใช้จึงส่ง บัตรผ่านขอใช้บริการให้ผู้ใช้
3. สุดท้ายผู้ใช้ขอรับการบริการจากเครื่องแม่ข่ายโปรแกรมประยุกต์โดยเสนอบัตรผ่านขอใช้บริการ ให้แก่เครื่องแม่ข่าย

****

**รูปที่ 2.2 การส่ง-ผ่านข้อมูลของเครื่องแม่ข่ายโปรแกรม ผู้ใช้ และ ศูนย์กลางกระจายกุญแจ**

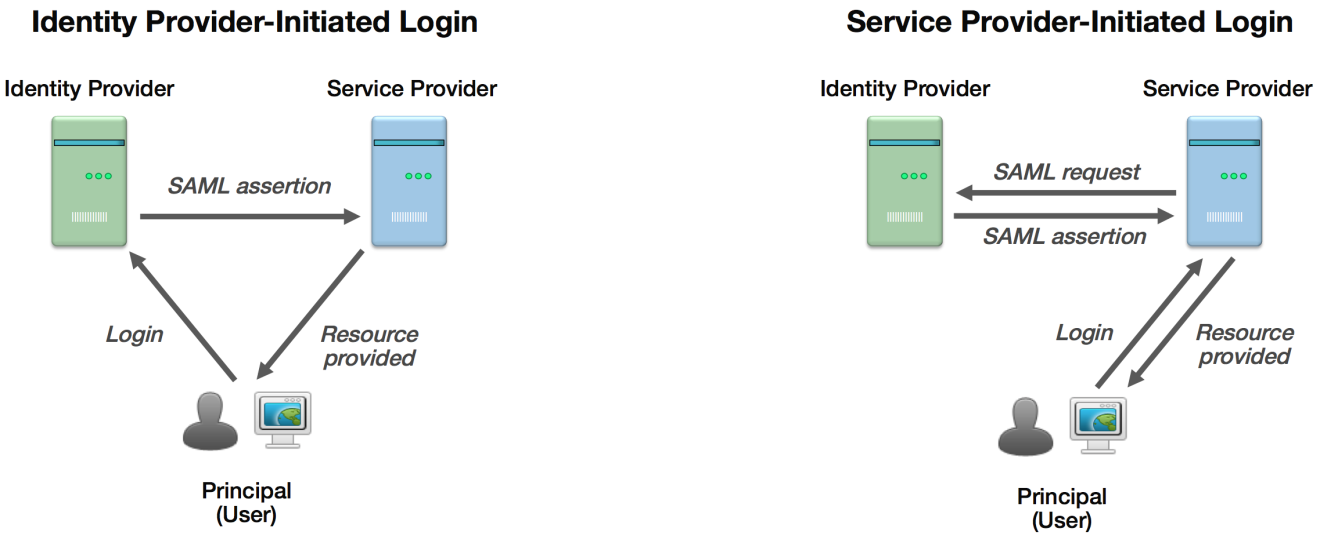
1. **ซิงเกิลไซน์ออนบนพื้นฐานแซมแอล (SAML-based SSO)**

โอเอซิส (OASIS) ได้ยอมรับการพัฒนาของแซมแอล ซึ่งเป็นภาษาที่ต่อยอดจากเอกซ์เอ็มแอล (XML)

ซึ่งแซมแอล มีคุณสมบัติที่ไม่ขึ้นกับ โพรโทคอล (Protocol) และ แพลตฟอร์ม (Platform) เพื่อควบคุมการเข้าถึงทรัพยากรสำหรับการพิสูจน์ตัวตน

ข้อความยืนยัน เป็นหัวใจหลักของแซมแอล ซึ่งประกอบด้วย การพิสูจน์ตัวจริง คุณลักษณะ และ การอนุญาต ผู้ให้บริการ (Services Provider) ส่งข้อมูลผู้ใช้สำหรับการพิสูจน์เอกลักษณ์ เมื่อผู้ใช้เข้าถึงส่วนบริการซึ่งถูกสร้างขึ้นด้วยผู้ให้บริการผ่านเบราว์เซอร์

ผู้ให้บริการตรวจสอบยืนยันตัวตน (Identity Provider) และ ผู้ให้บริการทำการแลกเปลี่ยนข้อมูล เมื่อผู้ใช้ร้องขอการบริการ ผู้ให้บริการตรวจสอบยืนยันตัวตนจึงสร้างข้อความแซมแอล และส่งข้อความไปยังผู้ให้บริการ จากนั้นผู้ให้บริการยืนยันความถูกต้องของแซมแอล เพื่อให้การบริการ



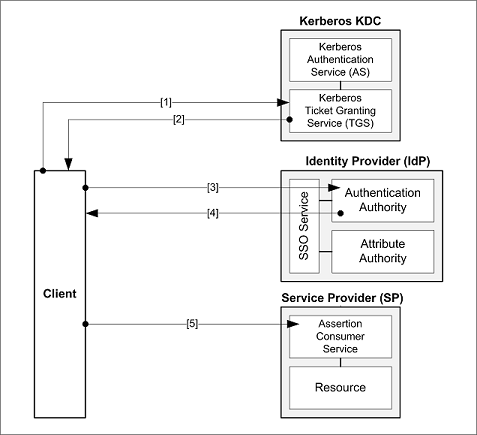
**รูปที่ 2.3 การส่งผ่านข้อมูลของ ผู้ใช้ ผู้ให้บริการ และ ผู้ให้บริการตรวจสอบยืนยันตัวตน**

1. **ซิงเกิลไซน์ออนบนพื้นฐานรวมของแซมแอล และเคอร์เบอร์รอส**

การจัดการพิสูจน์ตัวจริงแบบรวม และความสามารถในการพิสูจน์ตัวจริงสองทิศทางเป็นความสามารถของ ซิงเกิลไซน์ออนบนพื้นฐานของเคอร์เบอร์รอส ซึ่งช่วยลดปัญหาการบำรุงรักษาระบบ และเพิ่มความปลอดภัยอย่างมาก แต่ว่ายังมีข้อเสียกับการส่งผ่านของข้อมูลตั๋ว การเข้ารหัสและถอดรหัสของตั๋วยังมีความยุ่งยากอยู่มาก นอกจากนั้นข้อมูลผู้ใช้ยังเล็ดลอดได้ง่ายถ้ามีการส่งผ่านระหว่างเบราว์เซอร์และเครื่องแม่ข่ายรุ่นเก่า

แซมแอล เป็นภาษาที่ดั้งเดิมมาจากเอกซ์เอ็มแอล มีคุณลักษณะไม่ขึ้นกับแพลตฟอร์มและภาษาใด ทำงานได้กับระบบที่ไม่เข้ากัน เพื่อให้ง่ายต่อการส่งข้อมูล ผู้ใช้พิสูจน์ตัวจริงกับระบบในรูปแบบข้อความยืนยัน สร้างความปลอดภัยเพราะข้อความถูกใช้แค่ครั้งเดียวจึงถูกลบ แต่มีข้อเสียที่ระบบถูกโจมตีได้ง่ายเพราะขาดการพิสูจน์ตัวจริงแบบสองด้าน (Mutual Authentication)

แผนการสร้างระบบซิงเกิลไซน์ออนโดยการรวมจุดเด่นของเคอร์เบอร์รอส และแซมแอล ช่วยเพิ่มประสิทธิภาพ โดยการรวมความสามารถของแต่ละรูปแบบ เช่น การพิสูจน์แบบสองทางของเคอร์เบอร์รอส ช่วยเพิ่มความปลอดภัยแก่ซิงเกิลไซน์ออน ที่ทำงานกับแซมแอล และแซมแอล ช่วยเป็นตัวกลางสำหรับการส่งตั๋ว นอกจากนั้นการหลุดลอดของข้อมูลยังสามารถถูกแก้ไขโดยการใช้ข้อความยืนยันของแซมแอล



**รูปที่ 2.4 การส่งข้อมูลของซิงเกิลไซน์ออนบนพื้นฐานรวมของแซมแอล และเคอร์เบอร์รอส**

**2.5 เทคโนโลยีการยืนยันตัวตนนักศึกษา ที่มีใช้งานในมหาวิทยาลัยเชียงใหม่**

1. **เรเดียส ( RADIUS )**
2. **การเชื่อมต่อกับการบริการเรเดียส**

การเชื่อมต่อกับส่วนบริการเรเดียส ของมหาวิทยาลัยเชียงใหม่มีความต้องการ คือ เครื่องแม่ข่ายใช้ไอพีสาธารณะ การส่งข้อมูลยืนยันใช้อีเมล์ รหัสผ่าน ที่อยู่ของผู้ให้บริการเรเดียส และรหัสลับเข้าใช้งาน โดยสามารถเชื่อมต่อผ่านพีเฮชพี โดยการใช้คลังโปรแกรมของ พีอีซีแอลเรเดียส ( PECL Radius )

โดย 2 ค่าที่ได้กลับมาคือ Access-Accept ระบบยืนยันว่ามีผู้ใช้ และ Access-Reject คือการยืนยันตัวตนไม่สำเร็จ



**รูปที่ 2.5 แผนภาพการส่งข้อมูลระหว่างผู้ขอบริการและบริการเรเดียสของสำนักบริการเทคโนโลยี**

1. **เอพีไอ ( API )**
2. **คำสั่งเพื่อทำงานกับเอพีไอ**
3. การยืนยันตัวตนนักศึกษา

https://account.cmu.ac.th/v1/api/validateUser?appId={AppID}&appSecret={AppSecret}&user={UserName}&pw={Password}

การยืนยันตัวตนแบบเกท ( GET ) ตามยูอาร์แอล ( URL ) มี พารามิเตอร์( Parameter ) ดังนี้

1. เลขโปรแกรม ( AppId ) คือ หมายเลขเอพีไอที่กำลังเรียกฟังก์ชัน validateUser
2. เลขรหัสโปรแกรม ( AppSecret ) คือ รหัสลับเพื่อเข้าใช้งานเอพีไอ
3. ชื่อผู้ใช้ ( Userame ) ที่ใช้เป็นชื่อผู้ใช้จากอีเมล์ของมหาลัยเชียงใหม่ โดยไม่ใส่ @cmu.ac.th
4. รหัสผ่าน ( Password ) ใช้รหัสผ่านเดียวกับของมหาวิทยาลัยเชียงใหม่
5. การร้องขอข้อมูลนักศึกษา

https://account.cmu.ac.th/v1/api/Students/{userName}?appId={AppID}&appSecret={AppSecret}&accees\_token={access\_token}

การขอข้อมูลนักศึกษาแบบเกท ตามยูอาร์แอลมี พารามิเตอร์ ดังนี้

1. ชื่อผู้ใช้ ( Username ) เป็นของนักศึกษาที่ยืนยันตัวตนสำเร็จ
2. เลขโปรแกรม
3. เลขรหัสโปรแกรม
4. โทเค็นผ่าน ( Access Token ) เป็นสายอักขระที่ได้รับหลังยืนยันตัวตนนักศึกษาสำเร็จ
5. **ตัวอย่างรูปแบบข้อมูล เจสัน (JSON) ที่ได้จากเอพีไอ**
6. ข้อมูลเจสันที่ได้มาจากการยืนยันตัวนักศึกษาสำเร็จ

{

"success": true,

"ticket": {

"access\_token": "zMbjvFfRCD",

"expires\_in": 1800,

"userName": "siwaphol\_boonpan",

"issued": "19/11/2557 16:18:12",

"expires": "19/11/2557 16:48:12"

}

}

1. ข้อมูลเจสันที่ได้มาจากการร้องขอข้อมูลนักศึกษา

{

"success": true,

"ticket": {

"success": true,

"ticket": {

"access\_token": "26ggzJGG1f",

"expires\_in": 1800,

"userName": "siwaphol\_boonpan",

"issued": "19/11/2557 16:20:41",

"expires": "19/11/2557 16:50:41"

}

},

"student": {

"id": "540510828",

"personType": {

"th\_TH": "นักศึกษาปัจจุบัน",

"en\_US": "Present Student"

},

"prefix": {

"th\_TH": "ศิวพล",

"en\_US": "siwaphol"

},

"firstName": {

"th\_TH": "ศิวพล",

"en\_US": "SIWAPHOL"

},

"lastName": {

"th\_TH": "บุญปั๋น",

"en\_US": "BOONPAN"

},

"level": null,

"citizen\_id": "1579900428149",

"faculty": {

"code": "05",

"th\_TH": "คณะวิทยาศาสตร์",

"en\_US": "Faculty of Science"

},

"image": null

}

}

โดยค่าของ ticket จะได้ก็ต่อเมื่อการร้องขอสำเร็จคือ ค่าของ success เป็นจริง



**รูปที่ 2.6 แผนภาพการส่งข้อมูลระหว่างผู้ขอบริการและเอพีไอของสำนักบริการเทคโนโลยี**