บทที่ 3

โครงสร้างความมั่นคงของระบบยูนิกซ์

3.1 นิยามที่เกี่ยวข้อง

มีผู้ให้นิยามของความมั่นคงของคอมพิวเตอร์ (Computer Security) ไว้ดังนี้

นิยามที่ 1 ความมั่นคงของคอมพิวเตอร์เกี่ยวข้องกับการป้องกันข้อมูลในคอมพิวเตอร์และการเข้าถึงทรัพยากรของคอมพิวเตอร์ (Farrow, 1991)

นิยามที่ 2 ความมั่นคงของคอมพิวเตอร์ คือ การที่ผู้ใช้สามารถไว้วางใจในระบบคอมพิวเตอร์นั้นๆ และโปรแกรมทำงานตามที่ผู้ใช้ต้องการ (Garfinkel, Spafford, 1991)

การดูแลความมั่นคงของระบบคอมพิวเตอร์ ประกอบด้วย

3.1.1 ความลับ (Secrecy) หมายถึงส่วนสำคัญของคอมพิวเตอร์เข้าถึงได้เฉพาะบุคคลหรือกลุ่มบุคคลที่มีสิทธิเท่านั้น

3.1.2 ความเชื่อถือได้ (Integrity) หมายถึงส่วนสำคัญของคอมพิวเตอร์ถูกเปลี่ยนแปลงได้เฉพาะบุคคลหรือกลุ่มบุคคลที่มีสิทธิเท่านั้น

3.1.3 การเข้าถึงได้ (Availability) หมายถึงส่วนสำคัญของระบบคอมพิวเตอร์ ต้องสามารถเข้าถึงได้โดยบุคคลหรือกลุ่มบุคคลที่มีสิทธิ (Pfleeger, 1989).

การคุกคาม (threat) คือการกระทำใดๆก็ตามที่อันจะก่อให้เกิดความสูญเสีย หรือความเสียหายต่อข้อมูลหรือระบบคอมพิวเตอร์ ไม่ว่าการกระทำนั้นจะเกิดจากธรรมชาติ ,มนุษย์ ,ฮาร์ดแวร์ หรือ โปรแกรม

3.2 Trusted Computing Base

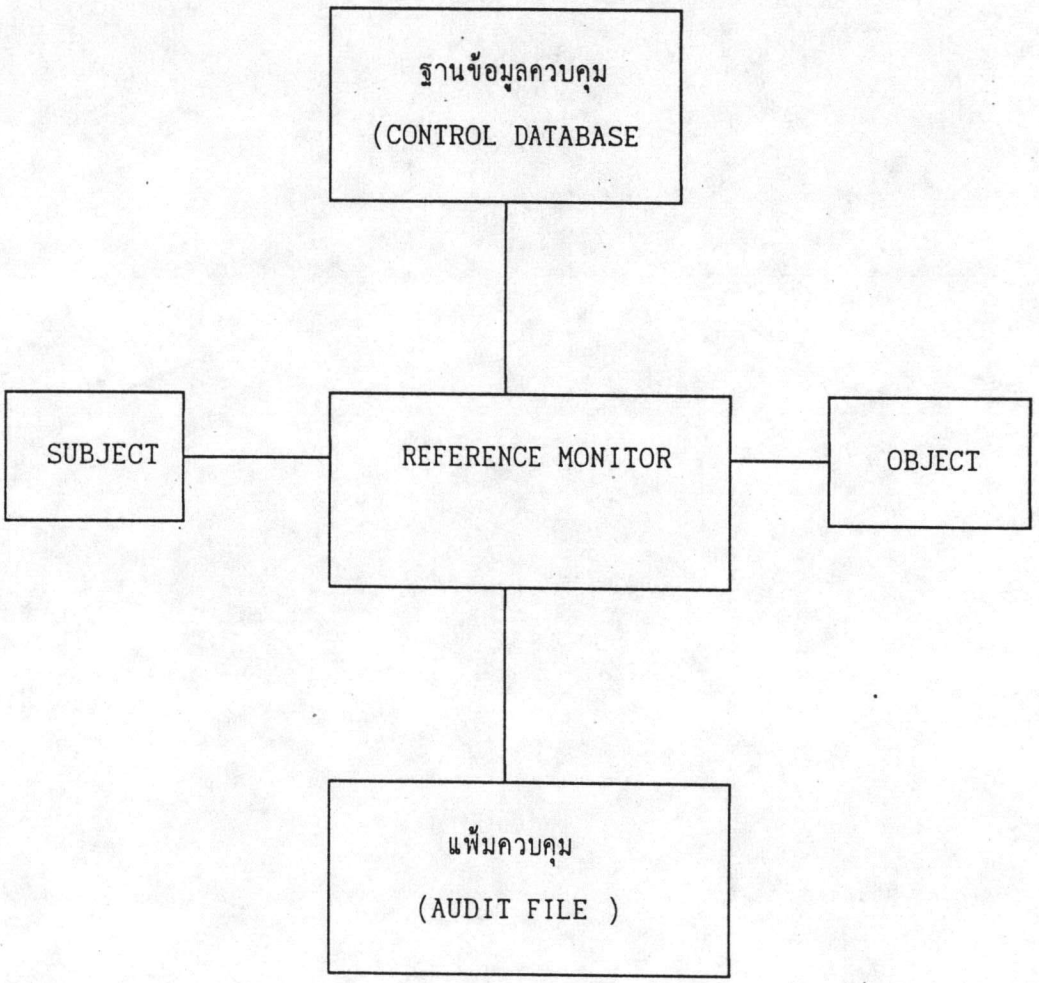
ในการกำหนดผู้ใช้ , บันทึกการทำงานของผู้ใช้ และควบคุมการเข้าถึง (access) ของผู้ใช้ เป็นหน้าที่ของ ฮาร์ดแวร์ (hardware) และซอฟท์แวร์ (software) ของคอมพิวเตอร์ในส่วน trusted portion ของฮาร์ดแวร์ รวมถึง กระบวนการแบ่งเนื้อที่ความจำและการป้องกันไม่ให้การทำงานของโปรแกรม เข้ารบกวนการทำงานของโปรแกรมที่ตำแหน่งความจำอื่น ส่วนระบบปฏิบัติการนั้น ถือว่าเป็นส่วน trusted portion ของซอฟท์แวร์ ดังนั้น trusted computing base เป็นการประสานการทำงานระหว่าง ฮาร์ดแวร์และระบบปฏิบัติการ ที่รับผิดชอบการดูแลระบบความมั่นคง

subject เป็นเอนติที (entity) เช่น บุคคลหรือโปรแกรมที่ทำงานตามความต้องการของผู้ใช้ ในระบบยูนิกซ์ โปรแกรมที่กำลังทำงานอยู่ เรียกว่า โปรเซส (process) ดังนั้นโปรเซสถือว่าเป็น subject

object เป็นเอนติที ที่ถูกกระทำ ซึ่งเก็บข้อมูลไว้ ซึ่งอาจจะเป็นแฟ้ม , อุปกรณ์ หรือ ระบบ เช่น ถ้าผู้ใช้ต้องการเรียกบรรณาธิกร (editor) vi เพื่ออ่านแฟ้ม โปรเซส vi ถือว่าเป็น subject ซึ่งพยายามเข้าเรียกแฟ้ม ซึ่งเป็น object

reference monitor เป็นส่วนหนึ่งของโปรแกรมที่รับผิดชอบการควบคุมการเข้าถึง โดยทั่วไป reference monitor มักจะมีขนาดเล็กเพื่อให้ง่ายต่อการวิเคราะห์ ถ้า reference monitor มีขนาดใหญ่ และสลับซับซ้อน จะวิเคราะห์ได้ยากว่า reference monitor ทำงานได้ดีเพียงไร

reference monitor จะใช้ทรัพยากร (resource) จาก 2 แหล่งด้วยกัน คือ ฐานข้อมูลควบคุม (control database) และแฟ้มตรวจสอบ (audit file) ฐานข้อมูลควบคุม จะเก็บข้อมูลการเข้าถึงของ object , subject ที่เข้าถึง object ตลอดจนประเภทของการเข้าถึง object ส่วนแฟ้มตรวจสอบนั้น reference monitor ใช้สำหรับบันทึกการทำงาน ข้อมูลที่เก็บไว้ในแฟ้มตรวจสอบนั้น อาจเริ่มจากไม่มีอะไรเลย , เวลาที่ object ถูกเข้าถึง , การเก็บบันทึกว่า subject ไหนที่เข้าถึง object ประเภทอะไร และเวลาที่ subject เข้าถึง object (Farrow, 1991)



รูปที่ 3.1 แสดงความสัมพันธ์ระหว่าง subject, reference monitor , ฐานข้อมูลควบคุม และแฟ้มตรวจสอบ

ในแกนของยูนิกซ์ (UNIX kernel) มีบางส่วนที่ทำหน้าที่ reference monitor ตัวอย่างเช่น ถ้าผู้ใช้ thong มีความต้องการดูแฟ้ม /etc/passwd โดยใช้คำสั่งว่า "more /etc/passwd" จะเกิดขั้นตอนดังนี้

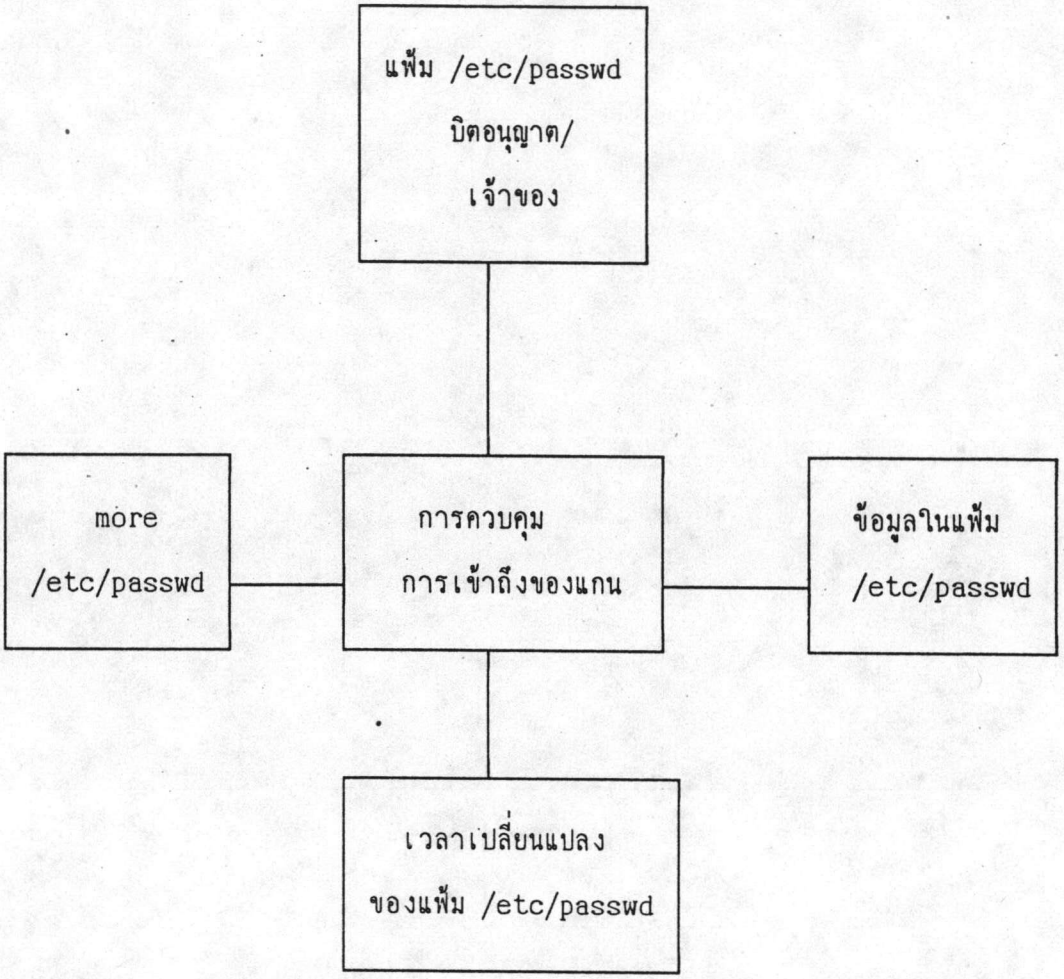
3.2.1 จะเกิดโปรเซส more และ /etc/passwd ซึ่งเป็น object ก็จะเป็น argument

3.2.2 โปรเซส more ซึ่งเป็น subject จะใช้คำสั่งระบบ (system call) เพื่อเปิดแฟ้ม /etc/passwd เพื่อทำการอ่าน

3.2.3 reference monitor ซึ่งเป็นโปรแกรมที่อยู่ในแกนยูนิกซ์ จะทำการตรวจสอบบิตอนุญาต (permission mode) ของแฟ้ม /etc/passwd และพบว่า โปรเซส more สามารถอ่านได้

3.2.4 reference monitor จะทำการเปิดแฟ้ม /etc/passwd เพื่อให้ โปรเซส more อ่าน

3.2.5 reference monitor จะทำการแก้แฟ้มตรวจสอบของแฟ้ม /etc/passwd โดยการเปลี่ยนแปลงเวลาที่แฟ้ม /etc/passwd ถูกอ่าน



รูปที่ 3.2 ส่วนที่ควบคุมการเข้าถึงของแกนยูนิกซ์ ทำหน้าที่ reference monitor เมื่อโปรเซส more /etc/passwd พยายามที่เปิดแฟ้ม /etc/passwd เพื่ออ่าน โดยที่ฐานข้อมูลควบคุมคือ บิตอนุญาตของobject และแฟ้มควบคุมประกอบด้วย เวลาเปลี่ยนแปลง

3.3 โครงสร้างระบบความมั่นคงของยูนิกซ์ (UNIX Security Structure)

3.3.1 การตรวจสอบผู้ใช้ (User Authentication)

ระบบยูนิกซ์ใช้รหัสผ่าน (password) ในการตรวจสอบผู้ใช้แต่ละคน หลังจากที่ผู้ใช้เลือกรหัสผ่านของตนเองแล้ว ระบบยูนิกซ์จะให้ตัวเลขสุ่ม (random number) 12 บิตที่เรียกว่า salt ซึ่งจะนำมาต่อกับรหัสผ่านที่ผู้ใช้เลือก จากนั้นยูนิกซ์จะทำการเข้ารหัสลับ (encrypt) ซึ่งผลลัพธ์ของการเข้ารหัส จะได้ตัวเลขสุ่ม 64 บิตเก็บไว้ที่แฟ้มข้อมูล /etc/passwd

เมื่อผู้ใช้เข้าใช้เครื่อง (login) และใส่รหัสผ่าน ยูนิกซ์จะนำ salt 12 บิต จากแฟ้ม /etc/passwd ต่อเข้ากับรหัสผ่านที่ผู้ใช้ใส่ แล้วทำการเข้ารหัสลับ ถ้าผลลัพธ์ของการเข้ารหัส ตรงกับรหัสที่มีอยู่แล้วในแฟ้ม /etc/passwd ผู้ใช้คนนั้นก็สามารถเข้าไปในระบบได้ แต่ถ้ารหัสไม่ตรงกัน ผู้ใช้คนนั้นก็ไม่สามารถเข้าไปในระบบได้

อัลกอริทึม (algorithm) ที่ใช้ในการเข้ารหัสลับคือ DES (Data Encryption Standard) ของ NIST (National Institute of Standards and Technology) ปัจจุบันยังไม่มีวิธีการใดๆ ที่จะสามารถแปลงรหัสผ่านที่ผ่านการเข้ารหัสแล้ว (ciphertext) กลับเป็นข้อความปกติ (clear text) (Garfinkel, Spafford, 1991)

3.3.2 แฟ้มข้อมูล /etc/passwd และ /etc/shadow

ระบบยูนิกซ์จะเก็บข้อมูลเกี่ยวกับผู้ใช้ทุกคน ในแฟ้ม /etc/passwd ซึ่งแฟ้ม /etc/passwd จะมีโครงสร้างดังนี้

```
name:encrypt password:user id:group id:user info:home dir:shell
```

ในแฟ้ม /etc/passwd แต่ละเขต (field) จะแยกกันด้วยเครื่องหมาย colon(:) ส่วนเขตสุดท้ายซึ่งเป็น login shell นั้นถ้าไม่ใส่ หมายถึงว่า ให้ใช้เบอร์นเชลล์ (Bourne shell) เป็น login shell

เนื่องจากแฟ้ม /etc/passwd นั้นทุกคนสามารถเรียกดูได้ ดังนั้นในยูนิกซ์รุ่นใหม่ เช่น ยูนิกซ์ ซิสเตมไฟว์ (Unix System V) ได้นำเขตที่เป็นรหัสลับไปเก็บไว้ในแฟ้ม /etc/shadow ซึ่งในแฟ้มนี้เฉพาะผู้จัดการระบบเท่านั้นที่สามารถดูได้

เนื่องจากแฟ้ม /etc/passwd และ แฟ้ม /etc/shadow เป็นแฟ้มที่เก็บรายละเอียดเกี่ยวกับผู้ใช้ทุกคนที่อยู่ในระบบ จึงถือว่าเป็นแฟ้มข้อมูลที่มีความสำคัญมากที่สุดในระบบยูนิกซ์

3.3.3 ประเภทของผู้ใช้ในระบบยูนิกซ์

3.3.3.1 ผู้จัดการระบบ (super user) ในระบบยูนิกซ์ทุกเครื่องจะมีผู้ใช้พิเศษซึ่งจะปรากฏในแฟ้ม /etc/passwd โดยมีชื่อว่า root ซึ่งจะมีหมายเลขประจำตัว (User Identification) เป็น 0 ในการที่จะเป็น root นี้ ผู้ใช้จะต้องใส่รหัสผ่านเหมือนผู้ใช้ประเภทอื่นๆ ซึ่งมักเรียกกันว่า root password เนื่องจากว่าผู้จัดการระบบมีอำนาจสิทธิสูงสุดในระบบ สามารถที่จะล้มเลิกสิทธิใดๆ (override permission) ได้ ดังนั้นรหัสผ่านสำหรับการเป็นผู้จัดการระบบนี้จึงมีความสำคัญที่สุดสำหรับระบบยูนิกซ์ทุกเครื่อง

3.3.3.2 ผู้ใช้ธรรมดา (ordinary user) ผู้ใช้ประเภทนี้จะมีสิทธิเท่าที่ผู้จัดการระบบอนุญาตเท่านั้น

3.3.3.3 ผู้ใช้พิเศษ (special user) เพื่อลดอันตรายอันเกิดจากการที่ผู้จัดการระบบมีอำนาจสูงสุด ระบบยูนิกซ์ส่วนใหญ่จะมีชื่อผู้ใช้ที่มีอำนาจพิเศษบางอย่าง เช่นสามารถเข้าถึงแฟ้มหรือไดเรคทอรีบางส่วนของระบบ ส่วนใหญ่ผู้ใช้พิเศษเหล่านี้จะเกี่ยว

ข้องกับการทำงานของระบบมากกว่าผู้ใช้ที่มีตัวตนจริง (Garfinkel, Spafford, 1991)

3.3.4 ประเภทของแฟ้มข้อมูลในระบบยูนิกซ์

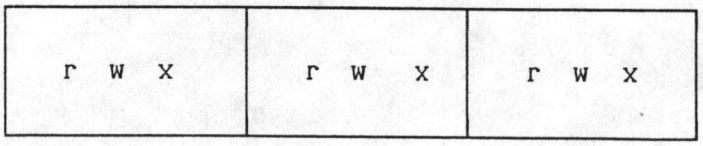
3.3.4.1 แฟ้มข้อมูลแบบธรรมดา (ordinary file) คือแฟ้มข้อมูลที่เก็บข้อมูลในลักษณะข้อความ (text) หรืออาจจะอยู่ในรูปที่สามารถกระทำการได้ (execute program)

3.3.4.2 แฟ้มข้อมูลแบบไดเรคทอรี (directory file) คือแฟ้มข้อมูลที่เก็บรายการของแฟ้มข้อมูลต่างๆ

3.3.4.3 แฟ้มข้อมูลแบบพิเศษ (special file) คือแฟ้มข้อมูลที่อ้างอิงกับอุปกรณ์ต่างๆ ผู้ใช้สามารถที่จะเขียนหรืออ่านข้อมูลจากแฟ้มแบบนี้ได้ เหมือนกับแฟ้มข้อมูลธรรมดา แต่จะมีการกระทำกับอุปกรณ์ทางกายภาพจริงๆ

3.3.5 กลไกการอารักขาแฟ้มข้อมูล (File Protection Mechanism)

บิตอนุญาตของแฟ้มข้อมูล (file permission bit) ของระบบยูนิกซ์เป็นดังนี้



owner group other

owner คือ เจ้าของแฟ้มข้อมูลนั้น

group คือ บุคคลอื่นที่อยู่ในกลุ่มเดียวกับเจ้าของแฟ้ม ซึ่งกำหนดไว้ในแฟ้ม /etc/group

other คือ บุคคลอื่นๆ ในระบบที่ไม่ได้อยู่ในกลุ่มเดียวกับเจ้าของแฟ้ม

ในกรณีที่แฟ้มข้อมูลนั้นเป็นแฟ้มข้อมูลธรรมดา (ordinary file)

r คือ สามารถอ่านแฟ้มข้อมูลนั้นได้

w คือ สามารถแก้ไขหรือลบแฟ้มข้อมูลนั้นได้

x คือ สามารถกระทำการ (execute) แฟ้มข้อมูลนั้นได้

ในกรณีที่แฟ้มข้อมูลนั้นเป็นแฟ้ม ไดเรคทอรี (directory)

r คือ สามารถดูว่าในไดเรคทอรีนั้นเป็นแฟ้มข้อมูลอะไรบ้าง

w คือ สามารถสร้างหรือลบแฟ้มข้อมูลในไดเรคทอรีนั้นได้

x คือ สามารถเข้าไปในไดเรคทอรีนั้นได้

3.3.6 sticky bit

ในบางกรณีแฟ้มข้อมูลหรือไดเรคทอรีอาจมีบิตพิเศษ ที่เรียกว่า sticky bit ซึ่งเมื่อดูบิตอนุญาตแล้ว ตัวอักษร x จะแทนที่ด้วย ตัวอักษร t

ถ้าแฟ้มข้อมูลมี sticky bit มีความหมายว่า โปรแกรมนั้นจะยังอยู่ใน swap space ถึงแม้ว่าโปรแกรมนั้นจะเสร็จสิ้นการทำงานแล้วก็ตาม โดยทั่วไปโปรแกรมที่มี sticky bit มักจะใช้กับโปรแกรมที่ทำงานบ่อยๆ และมีผู้ใช้หลายๆคน เช่น โปรแกรมบรรณาธิกร (editor)

ถ้าแฟ้มข้อมูลนั้นเป็น ไดเรคทอรี มีความหมายว่า แฟ้มข้อมูลที่อยู่ในไดเรคทอรีนั้นจะถูกเปลี่ยนชื่อ หรือลบได้โดยเจ้าของแฟ้ม ,เจ้าของไดเรคทอรี และผู้จัดการระบบเท่านั้น

3.3.7 <u>การได้สิทธิชั่วคราว</u> (Temporary Permission)

เมื่อโปรแกรมใดๆ ถูกกระทำการ จะเกิดโปรเซส (process) ขึ้น ซึ่งโปรเซสนั้นจะมีตัวเลข 4 ตัวที่เกี่ยวข้อง คือ real UID, effective UID, real GID และ effective GID

โดยทั่วไป real UID ซึ่งคือหมายเลขประจำตัวของผู้ใช้ (User Identification) ที่สร้างโปรเซส จะมีค่าเท่ากับ effective UID ในทำนองเดียวกัน real GID(Group Identification) ซึ่งคือหมายเลขประจำกลุ่มของผู้ใช้ที่สร้างโปรเซสก็จะเท่ากับ effective GID เช่นกัน

เมื่อใดก็ตามที่มีการกระทำการ กับโปรแกรมที่มีบิตกำหนดผู้ใช้ (set-UID) โปรเซสที่เกิดจากโปรแกรมนั้น จะมี effective UID เท่ากับ UID ของเจ้าของโปรแกรมนั้น ดังนั้นโปรเซสที่เกิดจากโปรแกรมที่มีบิตกำหนดผู้ใช้ จะมีสิทธิ (permission) เท่ากับเจ้าของโปรแกรมนั้น ไม่ว่าผู้ใดจะเป็นผู้กระทำการโปรแกรมนั้นก็ตาม ผู้ที่สามารถกำหนดบิตกำหนดผู้ใช้คือ เจ้าของโปรแกรมและผู้จัดการระบบเท่านั้น (Ritchie, 1984)

ในทำนองเดียวกัน ถ้ามีการกระทำการกับโปรแกรมที่มีบิตกำหนดกลุ่มผู้ใช้ (set-GID) โปรเซสที่เกิดจากโปรแกรมนั้นจะมี effective GID เท่ากับ GID ของเจ้าของโปรแกรม ซึ่งถ้าโปรเซสนี้มีการสร้างแฟ้มข้อมูล แฟ้มข้อมูลนี้อาจจะมีหมายเลขกลุ่มเท่ากับ effective GID ของโปรเซสนี้ (Garfinkel, Spafford, 1991)

โปรแกรมที่มีบิตกำหนดผู้ใช้ จะมีรูปแบบบิตอนุญาตดังนี้

```
-rwsr-xr-x
```

โปรแกรมที่มีบิตกำหนดกลุ่มผู้ใช้ จะมีรูปแบบบิตอนุญาตดังนี้

```
-rwxr-sr-x
```

โปรแกรมสามารถที่จะมีทั้งบิตกำหนดผู้ใช้ และบิตกำหนดกลุ่มในเวลาเดียวกันได้