1. Alice รันโปรแกรม Set-UID ที่มี owner คือ Bob และโปรแกรมนี้ต้องการ read ไฟล์ /tmp/x ซึ่ง Alice สามารถ read ไฟล์นี้ได้แต่เพียงผู้เดียว โปรแกรมนี้จะ read ไฟล์ /tmp/x นี้ได้หรือไม่ เพราะเหตุใด

   คำตอบ ไม่ได้ เนื่องจากว่า Alice รันโปรแกรมที่มีการ Set-UID ที่มี Owner คือ Bob ซึ่ง Bob ไม่มีสิทธิ์ที่จะ Read ไฟล์ /tmp/x ได้เลยทำให้อ่านไม่ได้

2. Process ต้องการจะ open ไฟล์เพื่อจะ read โดยค่า EUID ของ process คือ 1000 และ RUID คือ 2000 ถ้าไฟล์นี้ตั้ง permission ให้ RUID 2000 สามารถ read ได้ แต่ RUID 1000 ไม่สามารถ read ได้ process นี้จะยังสามารถ open ไฟล์นี้ได้หรือไม่ เพราะเหตุใด

   คำตอบ ไม่ได้ เนื่องจากว่าระหว่างรัน Process นี้ถือสิทธิ์ของ EUID 1000 อยู่

3. โปรแกรม Set-UID ที่ owner คือ root มีกลไกป้องกันไม่ให้ user ทั่วไปที่มารันโปรแกรมนี้ และได้สิทธิพิเศษของ root (root privilege) ณ ขณะที่รัน ทำอันตรายต่อระบบได้อย่างไร

   คำตอบ การรันโปรแกรมดังกล่าวจะทำให้ User ทั่วไปได้สิทธิ์เทียบเท่า root แต่ทว่าโปรแกรมไม่ได้ให้สิทธิ์แก่ User ทั้งหมด เพียงแต่ให้แค่เท่าที่ในโปรแกรมมีเท่านั้น

4. เราต้องการเปลี่ยนโปรแกรมที่ owner คือ seed ให้เป็นโปรแกรม Set-UID ที่ owner คือ root เราสามารถทำได้โดยการใช้คำสั่ง 2 คำสั่งตามลำดับนี้ได้หรือไม่ เพราะเหตุใด

```
$ sudo chmod 4755 prog
$ sudo chown root prog
```

   คำตอบ ไม่ได้ เนื่องจากว่า chown จะ disable บิท Set-UID โดยอัตโนมัติ เมื่อเปลี่ยน owner ของไฟล์

5. คำสั่ง `chown` จะทำการ disable บิท Set-UID โดยอัตโนมัติ นิสิตคิดว่าทำไม่ `chown` จึงมีพฤติกรรมเช่นนั้น

   คำตอบ เพื่อป้องกันการได้รับสิทธิ์อย่างไม่ถูกต้องด้วยวิธีการ Set-UID

6. เมื่อเราต้องการ debug โปรแกรม เราสามารถเปลี่ยนค่าตัวแปรภายในของโปรแกรมในระหว่าง การประมวลผลได้ และการทำเช่นนี้อาจทำให้พฤติกรรมการทำงานของโปรแกรมเปลี่ยนไป ถ้าเราจะ ใช้เทคนิคนี้ debug โปรแกรม Set-UID เราสามารถทำให้พฤติกรรมของโปรแกรมเปลี่ยนไปได้หรือ ไม่ เช่น ถ้าโปรแกรมมีการ open ไฟล์ /tmp/xyz เราสามารถไปเปลี่ยนชื่อไฟล์นี้ให้กลายเป็น /etc/passwd เพื่อให้โปรแกรม Set-UID นี้ไป open ไฟล์ /etc/passwd แทนได้หรือไม่

   คำตอบ ได้ ตราบใดที่เราสามารถเปลี่ยนค่าใน Debugger

7. ฟังก์ชั่น system() และ execve() สามารถนำมาใช้ในการเรียกโปรแกรมภายนอกมาประมวลผลได้ อธิบายว่าทำไม system() จึงมีความปลอดภัยน้อยกว่า execve()

   คำตอบ เนื่องจากว่า system() ทำการรันด้วยการใช้ /bin/sh ซึ่งเมื่อ Set-UID เป็น Root แล้วจะทำให้เราสามารถเรียก shell ของ root ผ่าน system() ได้ แต่ execve() จะแปลง Process ที่รันอยู่ในปัจจุบันอย่างเดียวจึงทำให้ ปลอดภัยกว่า

8. เมื่อโปรแกรมรับ input จาก user เราสามารถทำ input redirection เพื่อให้โปรแกรมรับ input จากไฟล์แทนการรับจาก keyboard ได้ เช่น prog < myfile จะทำให้ข้อมูลจาก myfile ถูกส่งเข้าไปที่ prog เพื่อประมวลผล ถ้าให้ว่า prog เป็นโปรแกรม Set-UID ที่ owner คือ root เราสามารถใช้ เทคนิค input redirection เพื่อทำการ read จาก /etc/shadow ได้หรือไม่

```
$ prog < /etc/shadow
```

   คำตอบ ไม่ได้ เนื่องจากไฟล์ /etc/shadow เป็นไฟล์ที่ Normal user ไม่สามารถเข้าถึงได้อยู่แล้ว ดังนั้นจึงไม่สามารถทำ input redirection ได้เนื่องจาก permission denied error

9. ถ้า parent process เป็น Set-UID process ที่มี EUID เป็น root และ RUID เป็น bob และ parent process นี้ fork() child process ออกมา และตัว child process ได้ drop privilege จาก root ลงมา จงอธิบายว่า process นี้จะยังสามารถทำให้เกิด capability leak ได้หรือไม่ เพราะเหตุใด และ child process นี้จะทำอันตรายต่อระบบได้หรือไม่ อย่างไร

   คำตอบ ได้เนื่องจากตอนที่รันอยู่ตัว child ได้สิทธิของ root อยู่แล้ว ซึ่งตราบใดที่ child process นี้ยังไม่ถูกปิดตัวลง Privilege ที่ child process ได้จะยังเหมือนเดิม

10. Superuser ในระบบต้องการให้สิทธิ Alice ในการ read ไฟล์ทุกตัวในระบบโดยใช้คำสั่ง `more` โดย superuser ได้ใช้ 3 คำสั่งต่อไปนี้สร้างโปรแกรม Set-UID ชื่อ `mymore` ขึ้นมา

```
$ cp /bin/more /tmp/mymore
$ sudo chown root /tmp/mymore
$ sudo chmod 4700 /tmp/mymore
```

เนื่องจาก permission บิทเป็น 4700 ตัว superuser ต้องใช้คำสั่งพิเศษเพื่อให้ Alice เป็นผู้ใช้งาน `mymore` ได้แต่เพียงผู้เดียว (คำสั่งพิเศษนี้ไม่ได้แสดงไว้) อธิบายว่า Alice สามารถยกระดับสิทธิตัวเองให้มากกว่าการ read ไฟล์ได้หรือไม่ ขอให้นิสิตศึกษา manual ของคำสั่ง `more` แล้วอธิบายแนวทางที่เป็นไปได้

คำตอบ ได้ เนื่องจากภายใต้คำสั่ง more สั่ง command ที่ทำให้ user สามารถเปิด Editor ได้ซึ่งการเปิด Editor นี้จะเสมือนการใช้ superuser ในการเปิดเลยทีเดียว

11. ถ้าเราต้องการให้ไฟล์สามารถ read ได้เฉพาะ user ที่มี user ID ต่ำกว่า 1000 อธิบายแนวทางในการสร้างกลไก access control ตาม policy นี้

คำตอบ สร้าง Group ใหม่ขึ้นมาสำหรับ user ที่มี user ID ต่ำกว่า 1000 โดยสามารถหา user ที่มี user ID ต่ำกว่า 1000 ได้ด้วยคำสั่ง `$ cat /etc/passwd | cut -d : -f1,3`

12. Sam เขียนโปรแกรมเพื่อ download บทความหลายๆอันจากเว็บเพจหน้าหนึ่ง เพื่อเขาจะได้ไม่ต้อง ไปเสียเวลาคลิกทีละลิงค์ โดยโครงสร้างหลักของโปรแกรมเป็นไปดังต่อไปนี้

```
char command[100];
char* line, url;
line = getNextLine(file);// Read in one line from the HTML file.
while (line != NULL) {
  // Parse the line to get a URL string.
  url = parseURL (line);
  if (url != NULL){
    // construct a command, and execute it
    sprintf(command, "%s %s", "wget", url);
    system(command);
```

```
  }
  line = GetNextLine(file);
}
```

โปรแกรม wget เป็นคำสั่งในระบบปฏิบัติการ UNIX สำหรับดาวน์โหลดไฟล์จาก URL ที่ใส่เป็น input ให้ wget ส่วน `getNextLine()` และ `parseURL()` เป็นฟังก์ชั่นที่ Sam เขียนขึ้นมาเอง

Admin ของเว็บเพจหน้านี้ต้องการจะ attack โปรแกรมของ Sam โดย admin รู้ว่า Sam พยายามทำอะไรกับเพจของเขา และ admin รู้โครงสร้างโปรแกรมด้านบน แต่ไม่รู้ code ภายในของ `getNextLine()` และ `parseURL()` อย่างไรก็ตาม admin เชื่อว่าน่าจะมีบั๊กในทั้งสองฟังก์ชั่นนี้ อธิบาย แนวทางที่เป็นไปได้ในการ attack โปรแกรมของ Sam และนิสิตจะแนะนำ Sam ให้แก้ไขโปรแกรมของ เขาอย่างไรเพื่อให้มันปลอดภัยมากขึ้น

<u>คำตอบ</u> แนวทางการการโจมตีคือการแก้ให้ url ที่จะเข้าไปใน function กลายเป็น command อื่น เช่น `url = "localhost:8000/ ; rm -rf *"` ซึ่งเมื่อรัน `system(command);` แล้วจะทำให้ `rm -rf` ทำงานเป็นคำสั่งที่สอง ซึ่งวิธีป้องกันคือให้ใช้ `execve()` แทน `System()`