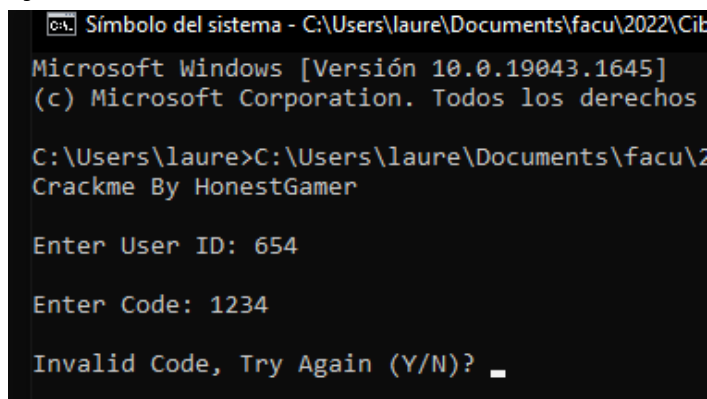


Ciberseguridad

Alumno: Laureano Enrique

Profesor: Matias Mevied

Ejercicio 1:



```
Símbolo del sistema - C:\Users\laure\Documents\facu\2022\Cib
Microsoft Windows [Versión 10.0.19043.1645]
(c) Microsoft Corporation. Todos los derechos reservados

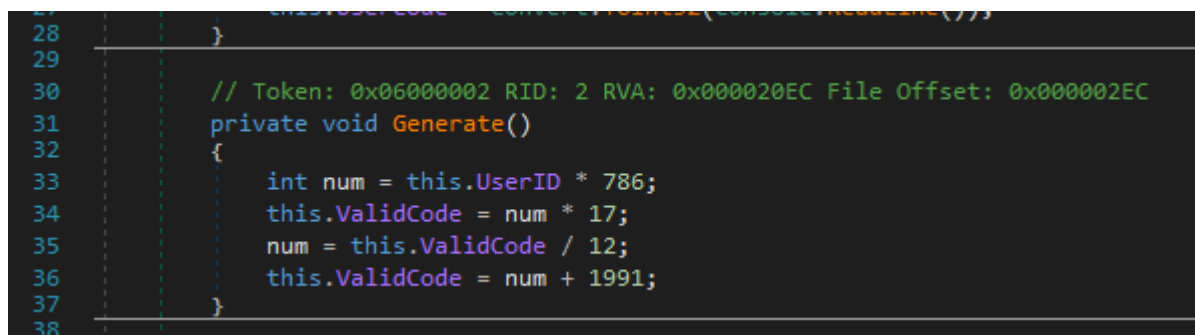
C:\Users\laure>C:\Users\laure\Documents\facu\2022\Cib
Crackme By HonestGamer

Enter User ID: 654

Enter Code: 1234

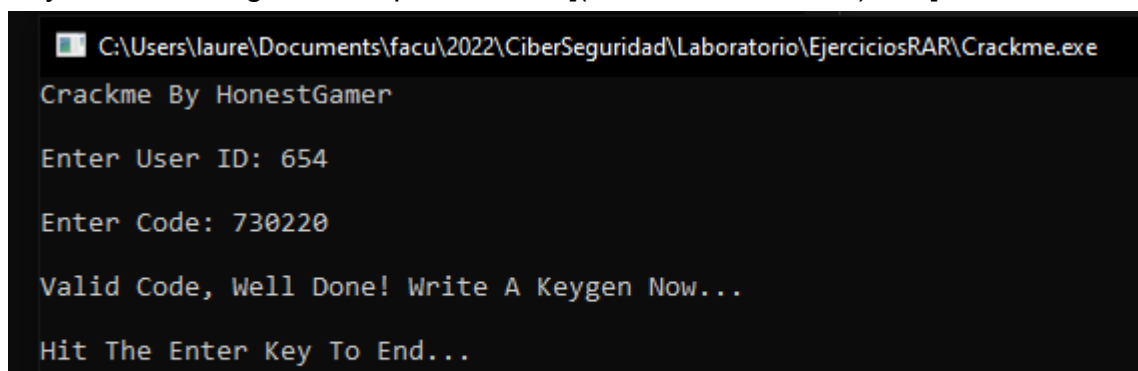
Invalid Code, Try Again (Y/N)? _
```

Es corrido el ejecutable crackme.exe y se intenta ingresar el código, pero este mismo es inválido.



```
28     }
29
30     // Token: 0x06000002 RID: 2 RVA: 0x000020EC File Offset: 0x000002EC
31     private void Generate()
32     {
33         int num = this.UserID * 786;
34         this.ValidCode = num * 17;
35         num = this.ValidCode / 12;
36         this.ValidCode = num + 1991;
37     }
38
```

Se abre el programa con el dnSpy que nos permite ver el código del mismo. Mirando la lógica para generar un código válido, este toma como referencia el user ID y realiza las siguientes operaciones: $[(\text{USERID} * 786 * 17) / 12] + 1991$



```
C:\Users\laure\Documents\facu\2022\CiberSeguridad\Laboratorio\EjerciciosRAR\Crackme.exe
Crackme By HonestGamer

Enter User ID: 654

Enter Code: 730220

Valid Code, Well Done! Write A Keygen Now...

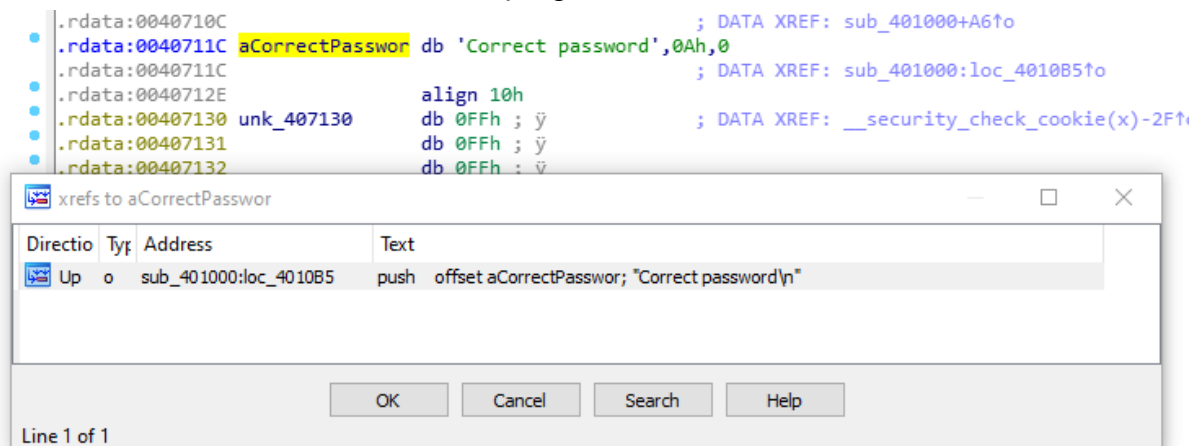
Hit The Enter Key To End...
```

Entonces ingresando el ID 654, se comprueba que el código válido corresponde a 730220.

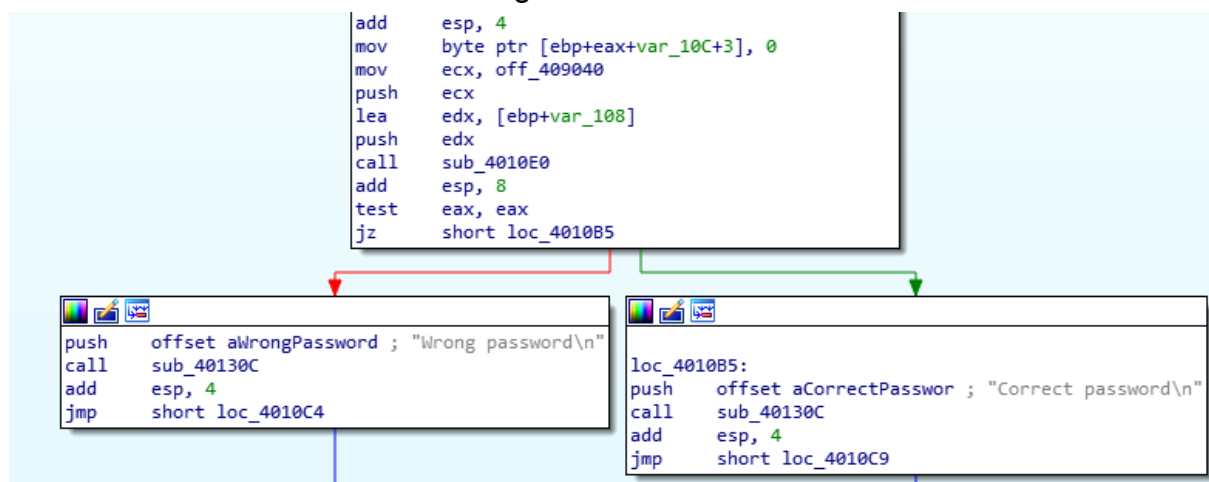
Ejercicio 2:

```
C:\Users\laure\Documents\facu\2022\CiberSeguridad\Laboratorio\Ejercicio 2
Enter password: hola
Wrong password
Enter password: chau
Wrong password
Enter password: _
```

Intento de adivinar la contraseña del programa “encuentreLaContraseña.exe”



Con el programa IDA se buscan las strings del programa y se busca en que momento se hace referencia al string “Correct Password”



Observando la logica del programa, sabemos que ingresa por el correct o wrong password segun el JZ (jump si la flag de zero esta activada).

```
.text:0040109A call sub_4010E0
.text:0040109F add esp, 8
.text:004010A2 test eax, eax
.text:004010A4 jmp short loc_4010B5
```

Cambiamos el JZ por JMP, entonces siempre realizaria el jump a la locacion 4010B5, que seria el camino de la password correcta, esto nos permite que cualquier string que ingresemos sea la password correcta como se ve a continuación:

```

C:\Users\laure>"C:\Users\laure\Documents\facu\2022\CiberSeguri
Enter password: hola
Correct password

C:\Users\laure>"C:\Users\laure\Documents\facu\2022\CiberSeguri
Enter password: otrapalabra
Correct password

C:\Users\laure>"C:\Users\laure\Documents\facu\2022\CiberSeguri
Enter password: asd123
Correct password

C:\Users\laure>_

```

Ejercicio 3:

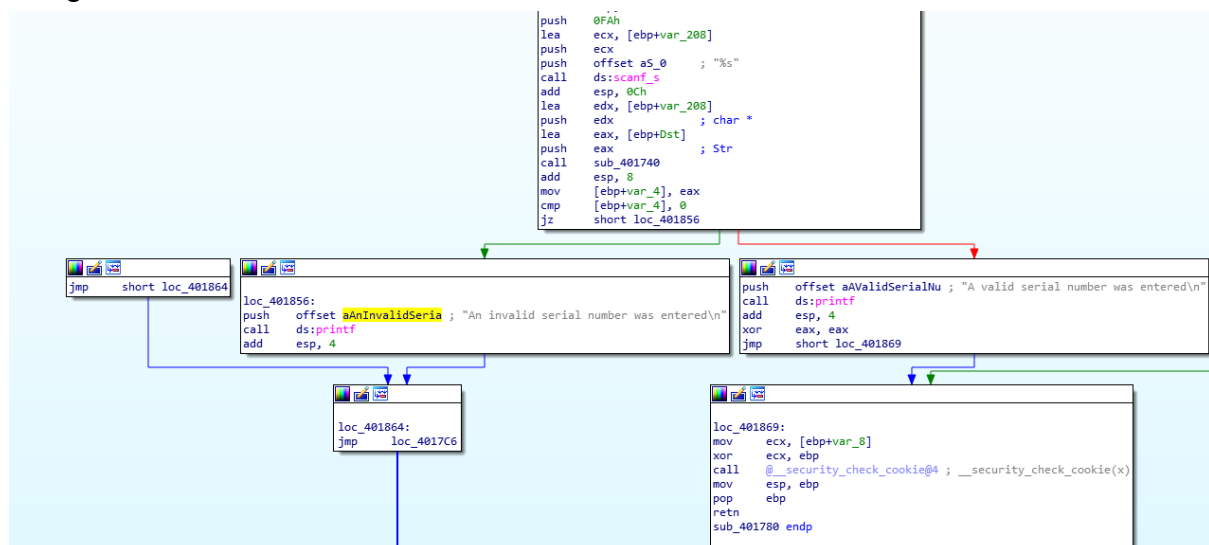
```

C:\Users\laure>C:\Users\laure\Documents\facu\2022\CiberSeguri
removeCheck.exe
Enter organization name:
asdd
Enter serial number:
ffs
An invalid serial number was entered
Enter organization name:
asdd
Enter serial number:
asdf
A valid serial number was entered

```

Como primera instancia corremos el programa y observamos que el serial number es correcto cuando la longitud de los strings coinciden.

Luego abrimos el IDA y ingresamos a todos los lugares donde se hace referencia al string valid Serial



Como vemos se hace una comprobación de si la longitud del nombre de la org coincide con la longitud del serialName.

```
.text:00401836      add     esp, 8
.text:00401839      mov     [ebp+var_4], eax
.text:0040183C      cmp     [ebp+var_4], 0
.text:00401840      jz      short loc_401856
.text:00401842      push    offset aAValidSerialNu ;
.text:00401847      call    ds:printf
.text:0040184D      add     esp, 4
```

Entonces podríamos omitir esta comprobación de JZ colocando una instrucción de NO_OP (90, en hexadecimal)

```
00401830  50 E8 0A FF FF FF 83 C4 0
00401840  90 90 68 54 30 40 00 FF 1
00401850  33 C0 EB 15 EB 0E 68 78 3
```

```
C:\Users\laure>C:\Users\laure\Documents\facu
removeCheck.exe
Enter organization name:
asd
Enter serial number:
asddd
A valid serial number was entered
C:\Users\laure>
```

Corremos el programa y ahora observamos que no importa si la longitud de los strings coinciden, el serial number es correcto de todas formas.

Parte 2:

No conseguí hacerla