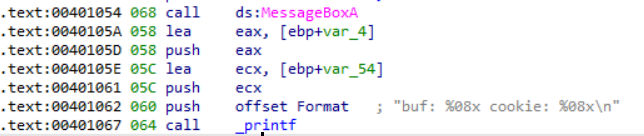
En este tercer ejercicio **STACK3\_VS\_2017** también se reservan **0x54 (84) bytes** al comienzo de la función.

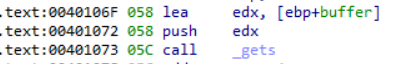


Otra vez hay un MessageBox y un Printf



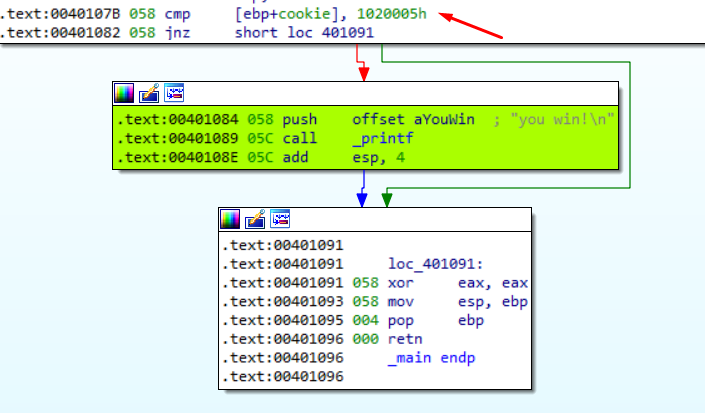
Renombramos las variables a **buffer** y **cookie**

A la función **gets** se le pasa un puntero al comienzo de la variable **buffer**

****

En fin, hasta aquí es todo igual a los ejercicios anteriores.

Lo único que veo es que el valor de la comparación ha vuelto a cambiarse.



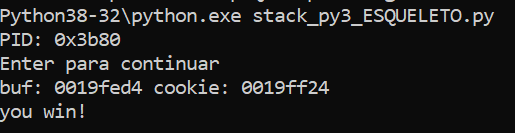
Ahora se compara la **cookie** contra **0x1020005**

En **little endian** eso seria visto asi **0x05 0x00 0x02 0x01**

En este caso el **0x00** nos puede llegar a cortar la cadena de bytes que le pasemos a la variable **buffer.** Sabemos que las cadenas terminan con un **0x00**

De hecho, si yo pruebo el script de python pasandole el **0x00** para mi sorpresa funciona bien, y nos muestra el “**you win!” **

| import sys from subprocess import Popen, PIPE  buffer = b"A" \* 80 cookie = b"\x05\x00\x02\x01"  payload = buffer + cookie  p1 = Popen("STACK3\_VS\_2017.exe", stdin=PIPE) print ("PID: %s" % hex(p1.pid)) print ("Enter para continuar")  p1.communicate(payload) p1.wait() input() |
| --- |

****

Bueno, me fijé en la descripción de la función **gets**, y al parecer usa **EOF(0x1A)** (Fin de archivo) o **newline (0x0A)** para terminar la entrada de flujo. Por lo que al pasar un **0x00** no afectará en nada.

Ejercicio anterior [02 Stack Nro 2](https://docs.google.com/document/d/1mhYTbhJ0FW-7oGWoZ5MxBMGb3u5lXNwzp00fF0KxKDM/edit?usp=sharing)

Ejercicio siguiente [04 Stack Nro 4](https://docs.google.com/document/d/1IYzQ4YxoO966ibRTXtV5QjJvigBgDsXdMRigDkAY6-A/edit?usp=sharing)