En el anterior ejercicio vimos que para llegar al mensaje “**you win!”** tuvimos que pisar el ret para saltar a ese bloque. [04 Stack Nro 4](https://docs.google.com/document/d/1IYzQ4YxoO966ibRTXtV5QjJvigBgDsXdMRigDkAY6-A/edit?usp=sharing)

Bien, en este otro ejercicio en binario está compilado en 64 bits, por lo que tendremos que usar el **IDA x64**

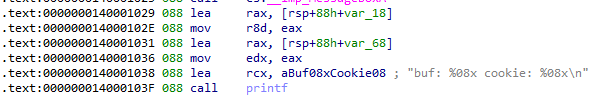
En **x64** los **primeros 4 argumentos** se pasan en los registros **RCX, RDX, R8, R9**, mientras que el resto, a través de la pila.

Sin embargo, se usa la instrucción **MOV** en lugar del **PUSH** para preparar la pila.

Vemos que esto es así, y se reservan **0x88 (136) bytes**

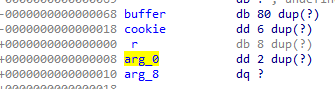


Tenemos el MessageBox y luego se imprimen las direcciones de las variables **buffer** y **cookie.**

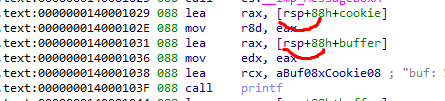


Renombramos las variables, de **var\_68** a **buffer,** y de **var\_18** a **cookie.**

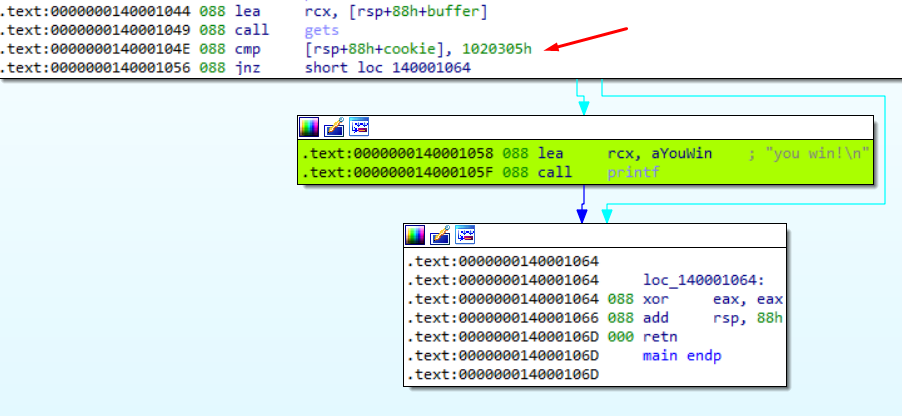
Esta es la pila, lo primero que me llama la atención es que no tenemos el **s (el base pointer)**



Para acceder a las **variables** vemos que se utiliza el registro **rsp**.



Vemos que la función **gets** recibe un puntero al inicio de la variable **buffer,** a esta misma la podemos desbordar y pisar el contenido de la **cookie** o del **return.**

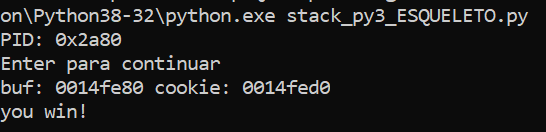
****

Se compara la **cookie** contra **0x1020305**

En **little endian** esto seria **05 03 02 01 00 00 00 00**

Mi Script de Python quedaria asi:

| import sys from subprocess import Popen, PIPE  buffer = b"A" \* 80 cookie = b"\x05\x03\x02\x01\x00\x00\x00\x00"  payload = buffer + cookie  p1 = Popen("STACK\_2\_64.exe", stdin=PIPE) print ("PID: %s" % hex(p1.pid)) print ("Enter para continuar")  p1.communicate(payload) p1.wait() input() |
| --- |

****

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

Referencia:

* [IDA64](https://book.huihoo.com/reverse-engineering-for-beginners/reverse-engineering-for-beginners-en-a5.pdf)
* [Register\_and\_Instruction\_Quick\_Start](https://wiki.cdot.senecacollege.ca/wiki/X86_64_Register_and_Instruction_Quick_Start)