

## Enumeration and Exploitation

### Binary 2 (Hard)

This challenge evaluates the contestant's ability to use a debugger to exploit a compiled binary. One possible tool to use is the "GDB" Linux program. A Linux binary is provided and the contestant is tasked with extracting the secret flag. This can be solved by attaching GDB to the provided binary to help search for any clues in the program. The GDB command, "info functions" will return a list of all the functions that are in scope. This list contains an interesting function called "getflagbytid" and can be called by breaking on line 1 of the main function and using the "call" command in GDB.

```
Reading symbols from NCL-2015-RE2_64bit...(no debugging symbols found)...done.
(gdb) info functions
All defined functions:

Non-debugging symbols:
0x0000000000400640 _init
0x0000000000400670 putchar@plt
0x0000000000400680 puts@plt
0x0000000000400690 strlen@plt
0x00000000004006a0 printf@plt
0x00000000004006b0 memset@plt
0x00000000004006c0 __libc_start_main@plt
0x00000000004006d0 __gmon_start__@plt
0x00000000004006e0 strtol@plt
0x00000000004006f0 fflush@plt
0x0000000000400700 __isoc99_scanf@plt
0x0000000000400710 exit@plt
0x0000000000400720 sleep@plt
0x0000000000400730 _start
0x0000000000400760 deregister_tm_clones
0x0000000000400790 register_tm_clones
0x00000000004007d0 __do_global_ctors_aux
0x00000000004007f0 frame_dummy
0x000000000040081d getflagbytid
0x0000000000400990 main
0x0000000000400b00 __libc_csu_init
0x0000000000400b70 __libc_csu_fini
0x0000000000400b74 _fini
```

```
(gdb) break main
Breakpoint 1 at 0x400994
(gdb) r
Starting program: /root/NCL-2015-RE2_64bit

Breakpoint 1, 0x0000000000400994 in main ()
(gdb) call getflagbytid(1234)
NCL-FY0F-4U44
$1 = 14
```

Question	Answer
What is the flag hidden in the program?	NCL-FY0F-4U44