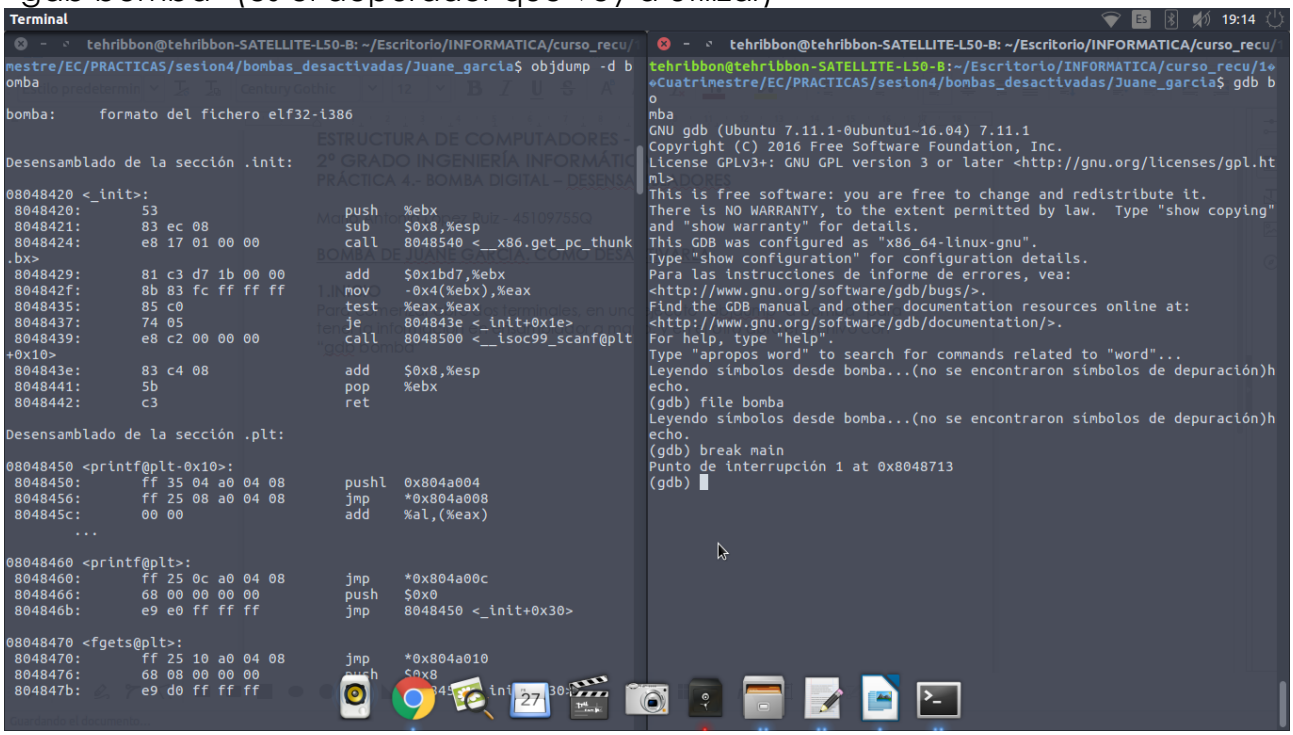
ESTRUCTURA DE COMPUTADORES -
2º GRADO INGENIERÍA INFORMÁTICA
PRÁCTICA 4.- BOMBA DIGITAL – DESENSAMBLADORES

Mario Antonio López Ruiz - 45109755Q

BOMBA DE JUAN EMILIO GARCIA MARTÍNEZ. COMO DESACTIVARLA

1.INICIO

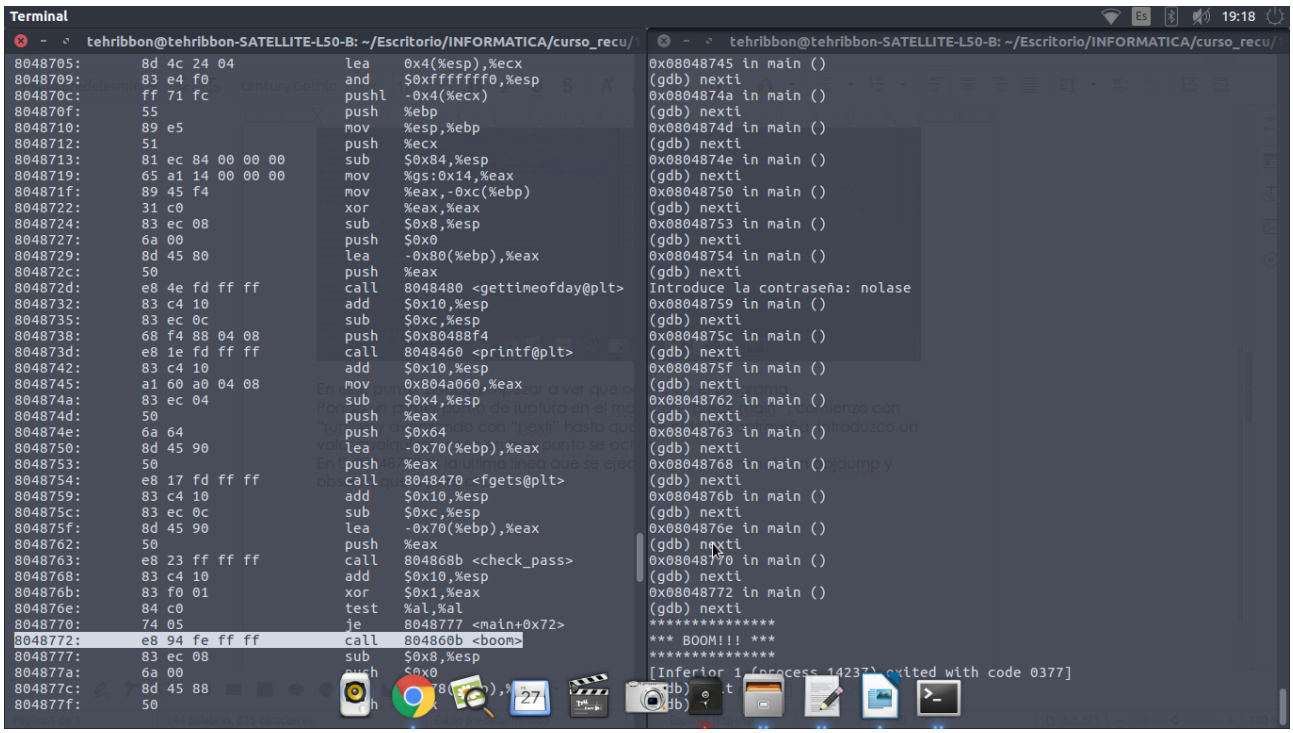
Primero tengo que darle permisos al archivo para poder ejecutarlo → "chmod a+x bomba".
Para comenzar abro dos terminales, en uno ejecuto "objdump -d bomba" para tener la información en ensamblador a mano, y en el otro abro el archivo con "gdb bomba" (es el depurador que voy a utilizar)



En este punto, puedo empezar a ver que ocurre en el programa.

2.CONTRASEÑA

Pongo un primer punto de ruptura en el main con "break main" , comienzo con "run" voy avanzando con "nexti" hasta que me pide la contraseña. Introduzco un valor cualquiera y veo en que punto se activa.
En 0x08048772 es la ultima linea que se ejecuta, voy al terminal con objdump y observo que ocurre ahí:



De aquí puedo deducir que la bomba se activa con una función <boom>. Dos líneas por encimas podemos ver una orden test %al, %al , y en la siguiente línea se comprueba si los valores guardados en esos registros son iguales. Por lo tanto, lo único que hay que hacer es poner un breakpoint en la linea donde ocurre el test y cambiar el valor de %al para evitar ese salto:

"break *0x0804876e"

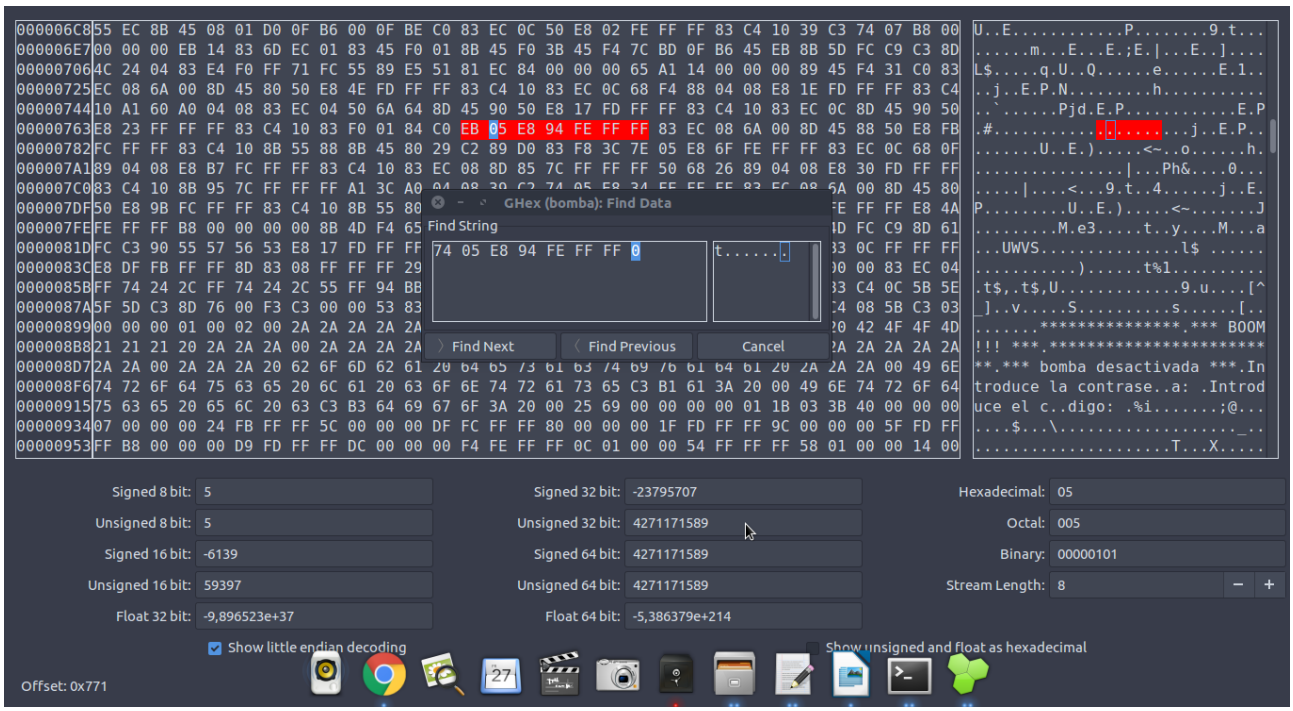
"set $al=0"

Avanzo el programa con "cont" y veo que me pide el código, por lo tanto he conseguido evitar la contraseña.

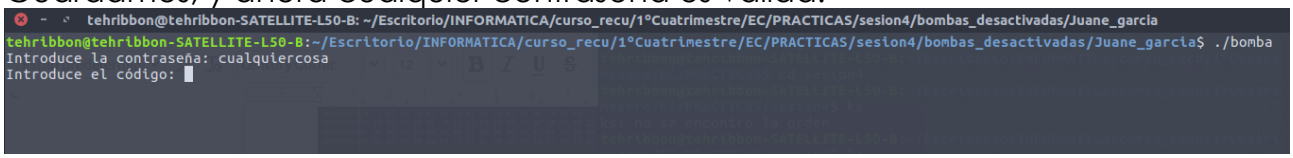
```
[Inferior 1 (process 14335) exited with code 0377]
(gdb) run
Starting program: /home/tehribbon/Escritorio/INFORMATICA/curso_recu/1ºCuat
rimestre/EC/PRACTICAS/sesion4/bombas_desactivadas/Juane_garcia/bomba
Introduce la contraseña: asfgdasf

Breakpoint 2, 0x0804876e in main ()
(gdb) set $al=0
(gdb) cont
Continuando.
Introduce el código:
```

Como el salto que se produce es del tipo JE voy a cambiarlo a JMP con un editor hexadecimal, buscando la posicion 74 05 e8 94 fe ff ff y cambiando 74→eb



Guardamos, y ahora cualquier contraseña es válida.



Sin embargo, también puedo encontrar cual es la contraseña, procedo a explicar el proceso.
Volvemos al inicio, al punto en el que me pide la contraseña y vuelvo a observar que ocurre con la contraseña:

```
8048754:       e8 17 fd ff ff          call   8048470 <fgets@plt>
8048759:       83 c4 10                add    $0x10,%esp
804875c:       83 ec 0c                sub    $0xc,%esp
804875f:       8d 45 90                lea    -0x70(%ebp),%eax
8048762:       50                      push   %eax
8048763:       e8 23 ff ff ff          call   804868b <check_pass>
8048768:       83 c4 10                add    $0x10,%esp
804876b:       83 f0 01                xor    $0x1,%eax
804876e:       84 c0                   test   %al,%al
8048770:       74 05                   je     8048777 <main+0x72>
8048772:       e8 94 fe ff ff          call   804860b <boom>
```

Veo que se llama a una función check_pass en la que probablemente se trabaje con la contraseña.
Antes de nada voy a ver las variables que hay definidas, para observar si puedo obtener algo de información:

```
0x0804a03c  passcode
0x0804a040  password
```

He encontrado estas dos variables, cuyos nombres son sospechosos, (el código lo dejo para mas adelante), asique compruebo que hay en ellas:

```
(gdb) x /1sb 0x0804a040
0x804a040 <password>:   "\nLIVOM"
(gdb)
```

Dentro de la variable password hay una cadena bastante sospechosa "\nLIVOM". Digo sospechosa porque normalemente las cadenas o string tiene el \n al final y no al principio. Quizá la contraseña está invertida, es decir, se le ha dado la vuelta para corregirlo en la función check_pass. Si le doy la vuelta a esa cadena obtengo "MOVIL\N", lo cual parece una cadena válida para una contraseña. Voy a comprobar si desactiva la bomba:

```
tehribbon@tehribbon-SATELLITE-L50-B:~/Escritorio/INFORMATICA/curso_recu/1◆
◆Cuatrimestre/EC/PRACTICAS/sesion4/bombas_desactivadas/Juane_garcia$ ./bom
b
a
Introduce la contraseña: MOVIL
Introduce el código:
```

Efectivamente, la contraseña es correcta. No he necesitado entrar en qué hacia la funcion check_pass porque ver el \n al comienzo me daba bastantes pistas, pero probablemente lo que hace esa función es darle la vuelta a la cadena.

3.CÓDIGO

```
(gdb) nexti
0x080487bb in main ()
(gdb) nexti
Introduce el código:
```

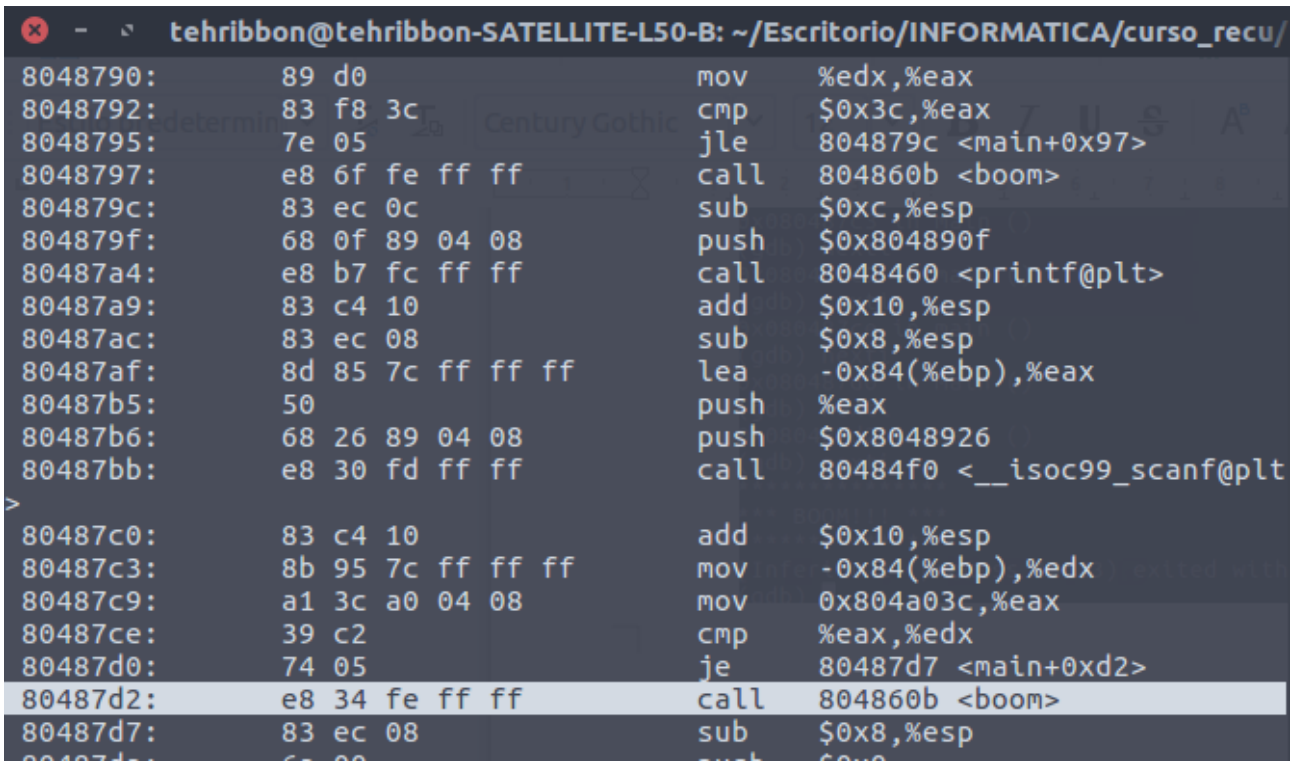
Retomo el programa desde el último breakpoint y ejecuto "nexti" para ver en que posición del se solicita la entrada del código:

```
(gdb) nexti
0x080487bb in main ()
(gdb) nexti
Introduce el código:
```

Continúo ejecutando con "nexti" para ver cuando se activa la bomba:

```
0x080487bb in main ()
(gdb) nexti
Introduce el código: asdf
0x080487c0 in main ()
(gdb) nexti
0x080487c3 in main ()
(gdb) nexti
0x080487c9 in main ()
(gdb) nexti
0x080487ce in main ()
(gdb) nexti
0x080487d0 in main ()
(gdb) nexti
0x080487d2 in main ()
(gdb) nexti
***************
*** BOOM!!! ***
***************
[Inferior 1 (process 14623) exited with code 0377]
(gdb)
```

En la posicion 0x080487d2 ocurre probablemente la llamada a boom, lo miro en el terminal con las instrucciones en ensamblador:

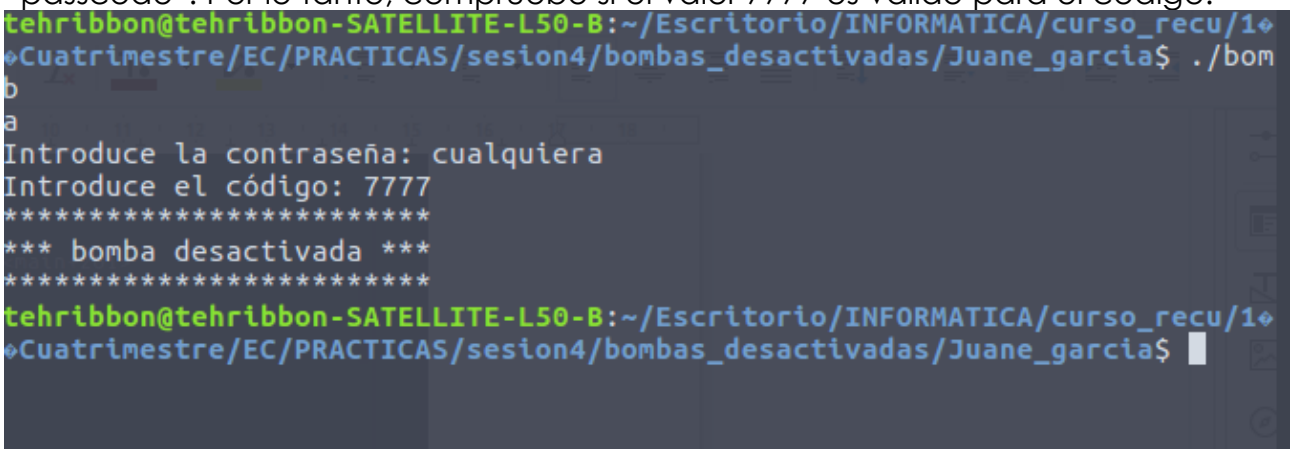


Efectivamente en esa línea se activa la bomba. Mirando las dos líneas anteriores veo que se hace un salto del tipo JE (jump equal) comparando %eax y %edx. Ahora tengo dos opciones, realizar el mismo proceso que para la contraseña (evitar ese salto), o ver que hay en esos registros para saber cual es el código correcto. Voy a optar por esta segunda opción.
Pongo un punto de ruptura en la posicion en la que se compara y veo que hay en esos registros:

```
(gdb) break *0x080487ce
Punto de interrupción 3 at 0x80487ce
(gdb) run
Starting program: /home/tehribbon/Escritorio/INFORMATICA/curso_recu/1ºCuat
rimestre/EC/PRACTICAS/sesion4/bombas_desactivadas/Juane_garcia/bomba
Introduce la contraseña: cualquiera
Introduce el código: 978

Breakpoint 3, 0x080487ce in main ()
(gdb) info registers
eax            0x1e61   7777
ecx            0x1      1
edx            0x3d2    978
ebx            0x0      0
esp            0xffffcda0       0xffffcda0
ebp            0xffffce28       0xffffce28
esi            0xf7fac000       -134561792
edi            0xf7fac000       -134561792
eip            0x80487ce        0x80487ce <main+201>
eflags         0x286    [ PF SF IF ]
cs             0x23     35
ss             0x2b     43
ds             0x2b     43
es             0x2b     43
fs             0x0      0
gs             0x63     99
(gdb) x/s 0x804a03c
0x804a03c <passcode>:    "a\036"
(gdb)
```

En %eax se introduce el valor del registro 0x804a03c. Con "x/s 0x804a03c" compruebo lo que hay en ese registro, que es el correspondiente a la variable "passcode". Por lo tanto, compruebo si el valor 7777 es válido para el código:



Por lo tanto, ya tenemos la bomba desactivada.