

BOMBA PRÁCTICA 4

ÍNDICE

1. [Código de la bomba.](#)
2. [Desactivar.](#)
3. [Cambio de claves con ghex.](#)

CÓDIGO DE LA BOMBA

Para compilar el código de la bomba, tan solo necesitamos ejecutar la orden que tenemos al comienzo del archivo "bombaJoseMaria.c"

```
gcc -Og bombaJoseMaria.c -o bombaJoseMaria -no-pie -fno-guess-branch-probability
```

La contraseña original es "AverSiAdivina".

El pin original es 1314.

Una vez cambiadas las claves, la contraseña será "hola,adios123" y el PIN será 1315.

A continuación se muestra el código de la bomba:

```
// gcc -Og bombaJoseMaria.c -o bombaJoseMaria -no-pie -fno-guess-branch-probability

#include <stdio.h> // para printf(), fgets(), scanf()
#include <stdlib.h> // para exit()
#include <string.h> // para strcmp()
#include <sys/time.h> // para gettimeofday(), struct timeval

#define SIZE 100
#define TLIM 7

char password[]="AverSiAdivina\n"; // contraseña
int passcode = 1314; // pin

void boom(void){ //Imprime la explosión
    printf( "\n"
            "*****\n"
            "*** BOOM!!! ***\n"
            "*****\n"
            "\n");
    exit(-1);
}

void defused(void){ //Imprime bomba desactivada
    printf( "\n"
            "-----\n"
            "----- bomba desactivada -----\n"
            "-----\n"
            "\n");
    exit(0);
}
```

```

int main(){
    char pw[SIZE];
    int pc, n;

    struct timeval tv1,tv2; // gettimeofday() secs-usecs

    gettimeofday(&tv1,NULL);      //Tomamos el tiempo

    do printf("\nIntroduce la contraseña, tienes 7 segundos.\nContraseña: ");
    while ( fgets(pw, SIZE, stdin) == NULL ); //Obtenemos un array de char desde stdin

    if ( strcmp(pw,password,sizeof(password)) )    //Comparamos las claves
        boom();

    gettimeofday(&tv2,NULL);      //Tomamos el tiempo

    if ( tv2.tv_sec - tv1.tv_sec > TLIM )    //Si tarda mas de TLIM (7s), explota
        boom();

    do { printf("\nIntroduce el pin, tienes 7 segundos.\nPIN: ");    //Leemos el pin
        if ((n=scanf("%i",&pc))==0)
            scanf("%*s") == 1;
    } while ( n!=1 );

    if ( pc != passcode )    //Si no coinciden, explota
        boom();

    gettimeofday(&tv1,NULL);

    if ( tv1.tv_sec - tv2.tv_sec > TLIM )    //Si hemos tardado más de lo permitido, explota
        boom();

    defused();    //Se desactiva la bomba

}

```

DESACTIVAR

Para desactivar la bomba, primero compilamos con gcc usando el comando indicado al principio del código.

Luego abrimos gdb ejecutando la siguiente orden

```
gdb <nombre de la bomba>
```

Una vez abierto gdb, mostramos el código ensamblador y los registros de memoria

```
layout asm  
layout reg
```

Ahora mismo deberíamos ver algo similar a lo siguiente

This GDB was configured as "x86_64-linux-gnu".
-.Type <RET> for more, q to quit, c to continue without paging.-c
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from BombaJoseMaria...
(No debugging symbols found in BombaJoseMaria)
(gdb) layout asm
(gdb) layout reg
(gdb) [

Una vez aquí, creamos un *breakpoint* en el main y comenzamos la ejecución del programa

```
br main  
run
```

Ahora si navegamos con las flechas del teclado, encontraremos la variable *password*, que es la contraseña que tenemos que descifrar.

Podemos ejecutar la siguiente orden y nos mostrará la clave

```
p(char[0xd])password
```

Y podemos hacer lo mismo con la variable *passcode*, el PIN

```
p(int)passcode
```

Si bajando al buscar la variable se nos descuadra la página, podemos volver a cuadrarla pulsando *Ctrl+L*.

Ahora mismo la ejecución se vería de la siguiente forma

```
Archivo Editar Ver Buscar Terminal Ayuda
jramirez@jramirez-HP-Pavilion-Notebook: /mnt/c3096feb-ceba-4048-b201-af08720979e4/Documentos/Universidad/Programacion/Segundo/EC/BombaJoseMaria

--Register group: general--
rax      0x401236      4198966      rbx      0x401360      4199264      rcx      0x401360      4199264      rdx      0x7fffffffdba8      140737488346024
rsi      0x7fffffffdb98      140737488346008      rdi      0x1      1      rbp      0x0      0      rsp      0x7fffffffdaa8      0x7fffffffdaa8
r8       0x0      0      r9       0x7fffffffdbd0      140737354009936      r10      0x0      11      r11      0x2      2
r12      0x40111b      4198672      r13      0x7fffffffdb90      140737488346000      r14      0x0      0      r15      0x0      0
rip      0x401236      0x401236 <main>      eflags   0x246      [ PF ZF IF ]      cs       0x33      51      ss       0x2b      43
ds       0x0      0      es       0x0      0      fs       0x0      0      gs       0x0      0

0x401220 <main+179> callq 0x4010e0 < printf_0x4010e0>
0x401220 <main+184> lea 0xc(%rsp),%rsi
0x401220 <main+189> lea 0xe6(%rip),%rdi # 0x402108
0x401220 <main+196> mov $0x0,%eax
0x401220 <main+201> callq 0x4010e0 < _isoc99_scanf@plt>
0x401304 <main+206> mov %eax,%ebx
0x401304 <main+208> test %eax,%eax
0x401304 <main+210> jne 0x401310 <main+229>
0x401304 <main+212> lea 0x5a(%rip),%rdi # 0x40210b
0x401310 <main+219> mov $0x0,%eax
0x401310 <main+224> callq 0x4010e0 < _isoc99_scanf@plt>
0x401310 <main+229> cmp $0x1,%ebx
0x401310 <main+232> jne 0x401310 <main+162>
0x401320 <main+234> mov 0x2da(%rip),%eax # 0x404000 <passcode>
0x401320 <main+240> cmp %eax,0xc(%rsp)

Native process 8720: In: main
Type <show copying> and <show warranty> for details.
This GDB was configured as "x86_64-linux-gnu".
--Type <RET> for more, q to quit, c to continue without paging.--
Type <show configuration> for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type 'help'.
Type <apropos word> to search for commands related to 'word'...
Reading symbols from bombaJoseMaria...
(No debugging symbols found in bombaJoseMaria)
(gdb) layout asm
(gdb) layout reg
(gdb) br main
Punto de interrupci3n 1 at 0x401230
(gdb) run
Starting program: /mnt/c3096feb-ceba-4048-b201-af08720979e4/Documentos/Universidad/Programacion/Segundo/EC/BombaJoseMaria/bombaJoseMaria

Breakpoint 1, 0x401230 in main ()
(gdb) p(char[0x1])password
$1 = "AversiAdivina"
(gdb) p(int)passcode
$2 = 1334
(gdb) []
```

Una vez realizado esto, tenemos 2 opciones, o bien nos aseguramos de meter la contraseña y el PIN rápido, ya que aún no ha empezado a contar el tiempo, o bien vamos paso a paso desactivando la comprobación del tiempo y de las claves.

Vamos a optar por la 2ª, ya que es la que incluye más complicaciones.

Si nos fijamos, en la línea *main+112* llama a *strncmp*, una función para comparar cadenas de caracteres. Vamos a crear un *breakpoint* en el *test* que hay después de esa línea para saltarnos la comprobación de la contraseña

```
br *main+117
cont
```

Cuando nos pida la contraseña, realmente podemos escribir la que nos de la gana, ya que vamos a saltarnos esa comprobación, en mi caso voy a escribir “hola”.

Ahora para saltarnos esa comprobación simplemente ponemos %eax a 0, ya que es el registro donde se realiza la comprobación

```
set $eax=0
```

```

jmmiravez@jmmiravez-HP-Pavilion-Notebook: /mnt/c3099fe8-ceba-4048-b201-af08720979e4/Documents/Universidad/Programacion/Segundo/EC/BombaJoseMaria
Archivo Editor Ver Buscar Ayuda
Registers: general
rax 0x0 0 rbx 0x401360 4199264 rcx 0xffffffff 4294967295 rdx 0x41 65
rsi 0x404068 4210792 rdi 0x7fffffffda30 140737488345648 rbp 0x0 0 rsp 0x7fffffffda00 0x7fffffffda00
r8 0x7fffffffda30 140737488345648 r9 0xc7c 124 r10 0x4004ed 4195565 r11 0xf 15
r12 0x401116 4189022 r13 0x7fffffffdb90 140737488346000 r14 0x0 0 r15 0x0 0
rip 0x4012b4 0x4012b4 <main+126> eflags 0x240 [ PF_Z IF ] cs 51 ss 43
ds 0x0 0 es 0x0 0 fs 0x0 0 gs 0x0 0

0x401270 <main+73> mov 0x2dfa(rip),rdx # 0x404060 <string@GLIBC.2.2.5>
0x401280 <main+80> mov $0x4,%esi
0x401290 <main+85> callq 0x401290 <__getx@plt>
0x401290 <main+90> test %rax,%rax
0x401290 <main+93> je 0x401294 <main+46>
0x401290 <main+95> lea 0x20(%rsp),rdi
0x401290 <main+100> mov $0xf,%edx
0x401290 <main+105> lea 0x2dc2(rip),rsi # 0x404068 <password>
0x4012a0 <main+112> callq 0x4012a0 <__strncmp@plt>
0x4012a0 <main+117> test %eax,%eax
0x4012a0 <main+119> je 0x4012b4 <main+126>
0x4012b0 <main+121> callq 0x4012b0 <__close@plt>
0x4012b4 <main+126> lea 0x20(%rsp),rdi
0x4012c0 <main+131> mov $0x0,%esi
0x4012c0 <main+136> callq 0x4012c0 <__timeofday@plt>
0x4012c0 <main+141> mov 0x20(%rsp),%rax
0x4012d0 <main+146> sub 0x10(%rsp),%rax
0x4012d0 <main+151> cmp %r8,%rax

Active process 10065 In: main
(gdb) run
Starting program: /mnt/c3099fe8-ceba-4048-b201-af08720979e4/Documents/Universidad/Programacion/Segundo/EC/BombaJoseMaria/bombaJoseMaria

Breakpoint 1, 0x0000000000401230 in main ()
(gdb) br *main+117
Punto de interrupci3n 3 at 0x4012ab
(gdb) cont
Continuando.

Breakpoint 2, 0x00000000004012a0 in main ()
(gdb) cont
Continuando.

Breakpoint 3, 0x00000000004012ab in main ()
(gdb) set $eax=0
Breakpoint 3, 0x00000000004012ab in main ()
(gdb) ni
0x00000000004012ab in main ()
(gdb) ni
0x00000000004012b4 in main ()
(gdb)

```

```
br *main+151
cont
```

```
set $eax=0
```

```

jramirez@jramirez-HP-Pavilion-Notebook /mnt/c309fe8-eba-4048-b201-af08720979e4/Documentos/Universidad/Programacion/Segundo/EC/BombaJoseMaria
Archivo  Editar  Ver  Buscar  Terminal  Ayuda

rax 0x0 0 0 rbx 0x401360 4199264 rcx 0x18 24 rdx 0x2260d0 2253088
rsi 0x2260d0 2253088 rdi 0x7ffffffcb080 140737353920640 rbp 0x0 0x0 rsp 0x7ffffffda0 7ffffffda00
r8 0x7ffffffcb080 140737353920640 r9 0x7ffffffd9d8 140737488345568 r10 0x7fffffffd9d8 140737488345552 r11 0x5fcb64e 1607165158
r12 0x401200 4180872 r13 0x7ffffffdb09 140737488346000 r14 0x0 0 r15 0x0 0
rip 0x401200 0x401200 <main+162> eflags 0x293 [ CF AF SF IF ] cs 0x33 51 ss 0x2b 43
ds 0x0 0 es 0x0 0 fs 0x0 0 gs 0x0 0

0x401200 <main+119> je 0x401204 <main+126>
0x401201 <main+121> callq 0x4011f0 <boom>
0x401202 <main+126> lea 0x20(%rsp),%rdi
0x401203 <main+131> mov %$0x, %esi
0x401204 <main+136> callq 0x4010c0 <gettimeofday@plt>
0x401205 <main+141> mov 0x20(%rsp), %rax
0x401206 <main+146> sub 0x10(%rsp), %rax
0x401207 <main+151> cmp %$0x, %rax
0x401208 <main+155> jle 0x401208 <main+162>
0x401209 <main+157> callq 0x4011f0 <boom>
0x40120a <main+162> lea 0x75(%rip), %rsi # 0x40125a
0x40120b <main+169> mov %$0x1, %edi
0x40120c <main+174> mov %$0x, %eax
0x40120d <main+179> callq 0x4011e0 <printf_chk@plt>
0x40120e <main+184> lea 0xc(%rsp), %rsi
0x40120f <main+189> lea 0xe6e(%rip), %rdi # 0x401208
0x401210 <main+196> mov %$0x, %eax
0x401211 <main+201> callq 0x4011d0 <_Isoc99_scanf@plt>

Native process 10065 In: main
Continuando.

Breakpoint 3, 0x0000000000004012ab in main ()
(gdb) set $eax=0
(gdb) ni
0x0000000000004012ad in main ()
(gdb) ni
0x0000000000004012b1 in main ()
(gdb) br *main+151
Punto de interrupci3n 4 at 0x4012cd
(gdb) cont
Continuando.

Breakpoint 4, 0x0000000000004012cd in main ()
(gdb) set $eax=0
(gdb) ni
0x0000000000004012d1 in main ()
(gdb) ni
0x0000000000004012d8 in main ()
(gdb)

```

Ahora si nos fijamos en la línea *main+240*, vemos que se realiza una comparación justo después de mover la variable *passcode*, esa es la comprobación del pin introducido.

Aquí podemos o bien meter el pin (que ya sabemos que es *1314*), o bien introducir cualquier pin, el de nuestra elección, y modificar el registro sobre el que se realiza la comparación para que no nos de error.

Vamos a optar por lo segundo.

Creamos entonces un *breakpoint* en la línea *main+240*

```
br *main+240
cont
```

Cuando nos pida el PIN, vamos a introducir, por ejemplo, “111”.

Ahora, si nos fijamos en los registros veremos que *%rax* tiene el valor de la *passcode* (1314).

rsi	0x2260d0	2253008
rax	0x522	1314
rsi	0x0	0
r8	0xa	10
ds	0x0	0
	326	326 240>

Tenemos que cambiar el valor del registro para que la comparación del valor salga bien. En nuestro caso, como hemos introducido “111” como PIN, tendremos que poner que *%rax* valga “111”

```
set $eax=111
```

Y si ejecutamos *ni*, veremos que nos saltamos el *boom*.

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
jramirez@jramirez-HP-Pavilion-Notebook: /mnt/c399fe8-ceba-4048-b201-a08720979e4/Documentos/Universidad/Programacion/Segundo/EC/BombaJoseMaria

Register group: General
rax 0x5f 111 rbx 0x1 140737488344256 rcx 0x0 0 rdx 0x0 0
rsi 0x0 0 rdi 0x7fffffff04c0 140737488344256 rbp 0x0 0 rsp 0x7fffffffda00 0x7fffffffda00
r8 0xa 10 r9 0x0 0 r10 0x7ffff7f50ac0 140737353419456 r11 0x0 0
r12 0x401110 4198672 r13 0x7ffff7f50ac0 140737488346000 r14 0x0 0 r15 0x0 0
r16 0x401331 0x401331 <main+251> r17 0x246 [ PF ZF IF ] cs 0x33 51 ss 0x2b 43
ds 0x0 0 es 0x0 0 fs 0x0 0 gs 0x0 0

0x401300 <main+210> jne 0x401310 <main+229>
0x401304 <main+212> lea 0xe5a(%rip),%rdi # 0x40210b
0x401311 <main+219> mov $0x0,%eax
0x401316 <main+224> callq 0x401010 <_isoc99_scanfgplt>
0x40131b <main+229> cmp %eax,%ebx
0x40131e <main+232> jne 0x401320 <main+162>
0x401320 <main+234> mov 0x2d3a(%rip),%eax # 0x404050 <passcode>
B+ 0x401320 <main+240> cmp %eax,0xc(%rsp)
0x401324 <main+244> je 0x401331 <main+251>
0x401328 <main+246> callq 0x401110 <boom>
0x401331 <main+251> lea 0x10(%rsp),%rdi
0x401336 <main+256> mov $0x0,%esi
0x40133b <main+261> callq 0x401010 <_gettimeofday@plt>
0x401340 <main+266> mov 0x10(%rsp),%rax
0x401345 <main+271> sub 0x20(%rsp),%rax
0x401348 <main+276> cmp %eax,%rax
0x40134b <main+280> jle 0x401352 <main+287>
0x401350 <main+282> callq 0x401110 <boom>

native process 10065 in: main
Continuando.
L77 PC: 0x401331

Breakpoint 4, 0x0000000000401331 in main ()
(gdb) set $eax=0
(gdb) ni
0x0000000000401331 in main ()
(gdb) ni
0x0000000000401330 in main ()
(gdb) br *main+240
Punto de interrupción 5 at 0x401320
(gdb) cont
Continuando.

Breakpoint 5, 0x0000000000401320 in main ()
(gdb) set $eax=111
(gdb) ni
0x0000000000401320 in main ()
(gdb) ni
0x0000000000401331 in main ()
(gdb)
```


Ahora tendremos que desactivar el contador del tiempo, que se realiza exactamente igual que como lo hicimos anteriormente.
Vamos a *main+276*, donde se realiza el *cmp*, y cambiamos el valor de *%rax* a 0

```
br *main+276
cont
set $eax=0
```

Ahora al ejecutar ni hasta llegar al final del programa, veremos que a bomba ha quedado desactivada.

```
Archivo Editar Ver Buscar Terminal Ayuda
jramirez@jramirez-HP-Pavilion-Notebook: /mnt/C3096fe8-ceba-4048-b201-af08720979e4/Documentos/Universidad/Programacion/Segundo/EC/BombaJoseMaria

Register group: general
rax 0x0 0 rdx 0x2bf42a 2880554
rsi 0x2bf42a 2880554 rdi 0x7ffffff7c080 140737353920640 rbp 0x0 0 rsp 0x7ffffff7da00 0x7ffffff7da00
r8 0x7ffffff7c080 140737353920640 r9 0x7ffffff7d9d8 140737488345560 r10 0x7ffffff7d9d8 140737488345552 r11 0x5fcb6b56 1607166806
r12 0x401110 4198072 r13 0x7ffffff7d9d8 140737488346000 r14 0x0 0 r15 0x0 0
rip 0x401355 0x401355 <main+287> eflags 0x203 [ CF AF SF IF ] cs 0x33 51 ss 0x20 43
ds 0x0 0 es 0x0 0 fs 0x0 0 gs 0x0 0

0x401311 <main+219> mov $0x0,%eax
0x401316 <main+224> callq 0x40107f <_isoc99_scanf@plt>
0x40131b <main+229> cmp $0x1,%ebx
0x40131e <main+232> jne 0x401208 <main+162>
0x401320 <main+234> mov 0x2d3a(%rip),%eax # 0x404060 <passcode>
0x401323 <main+237> cmp %eax,0xc(%rsp)
0x401326 <main+240> je 0x401331 <main+251>
0x40132c <main+246> callq 0x4011f0 <boom>
0x401331 <main+251> lea 0x10(%rsp),%rdi
0x401336 <main+256> mov $0x0,%esi
0x40133b <main+261> callq 0x4010c0 <gettimeofday@plt>
0x401340 <main+266> mov 0x10(%rsp),%rax
0x401343 <main+271> sub 0x20(%rsp),%rax
0x401346 <main+276> cmp $0x0,%rax
0x40134e <main+280> jle 0x401355 <main+287>
0x401350 <main+284> callq 0x4011f0 <boom>
0x401355 <main+287> callq 0x401210 <defusebomb>
0x40135a <main+292> nopw 0x0(%rax,%rax,1)

Native process 10065 in: main
Continuando.
L7? PC: 0x401355

Breakpoint 5, 0x0000000000401320 in main ()
(gdb) set $eax=111
(gdb) ni
0x0000000000401320 in main ()
(gdb) ni
0x0000000000401331 in main ()
(gdb) br *main+276
Punto de interrupci3n 6 at 0x40134e
(gdb) cont
Continuando.
Breakpoint 6, 0x000000000040134e in main ()
(gdb) set $eax=0
(gdb) ni
0x000000000040134e in main ()
(gdb) ni
0x0000000000401355 in main ()
(gdb) □
```

CAMBIO DE CLAVES CON GHEX

Para realizar el cambio de claves, usaremos *ghex*.

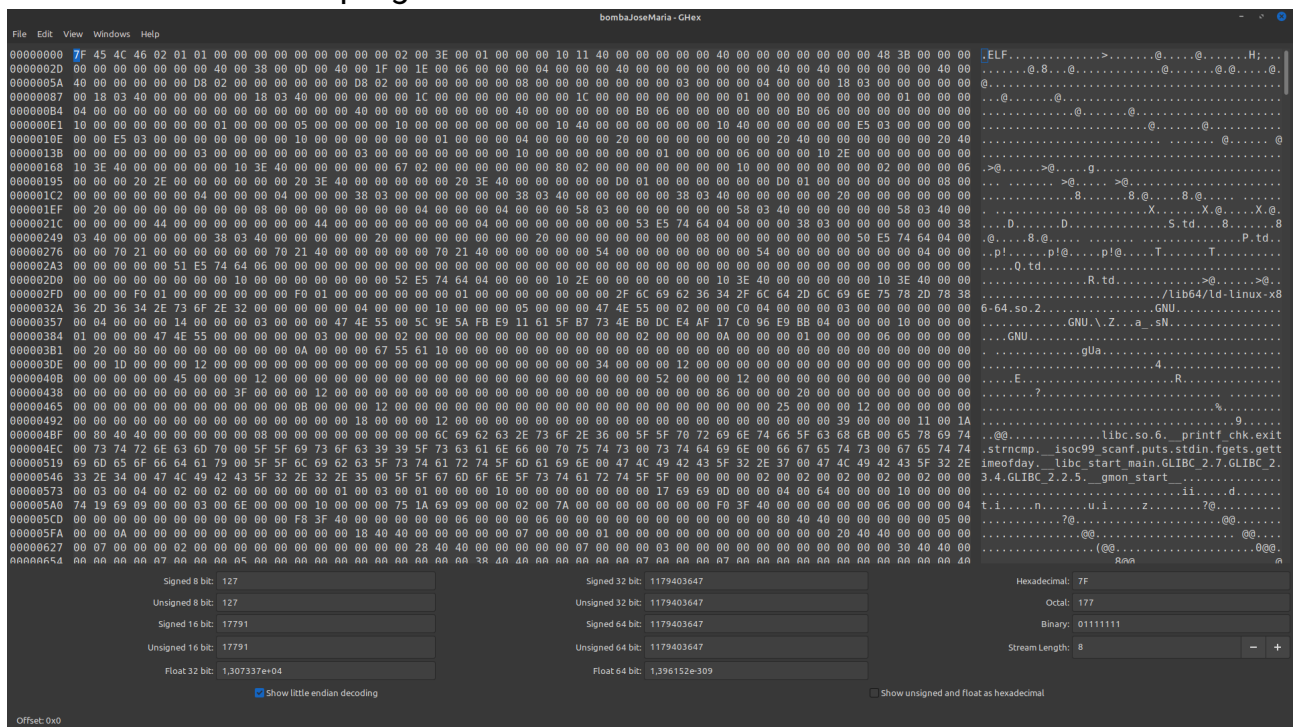
Si no tenemos este programa instalado, podemos instalarlo con la siguiente orden

```
sudo apt install ghex
```

Una vez instalado, vamos a abrir *ghex* con la siguiente orden

ghex <nombre de la bomba>

Y nos debería abrir el programa con una interfaz similar a esta

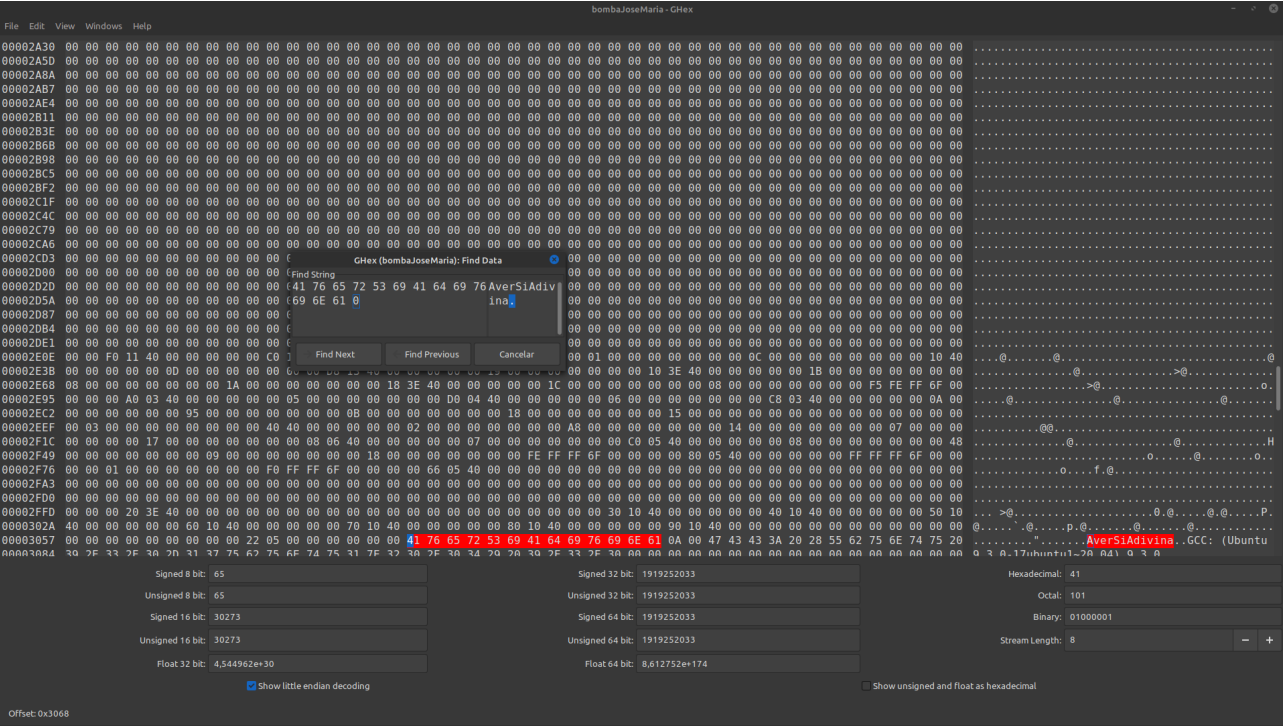


Recordemos que ya sabemos las claves por el apartado anterior ("AverSiAdivina", "1314").

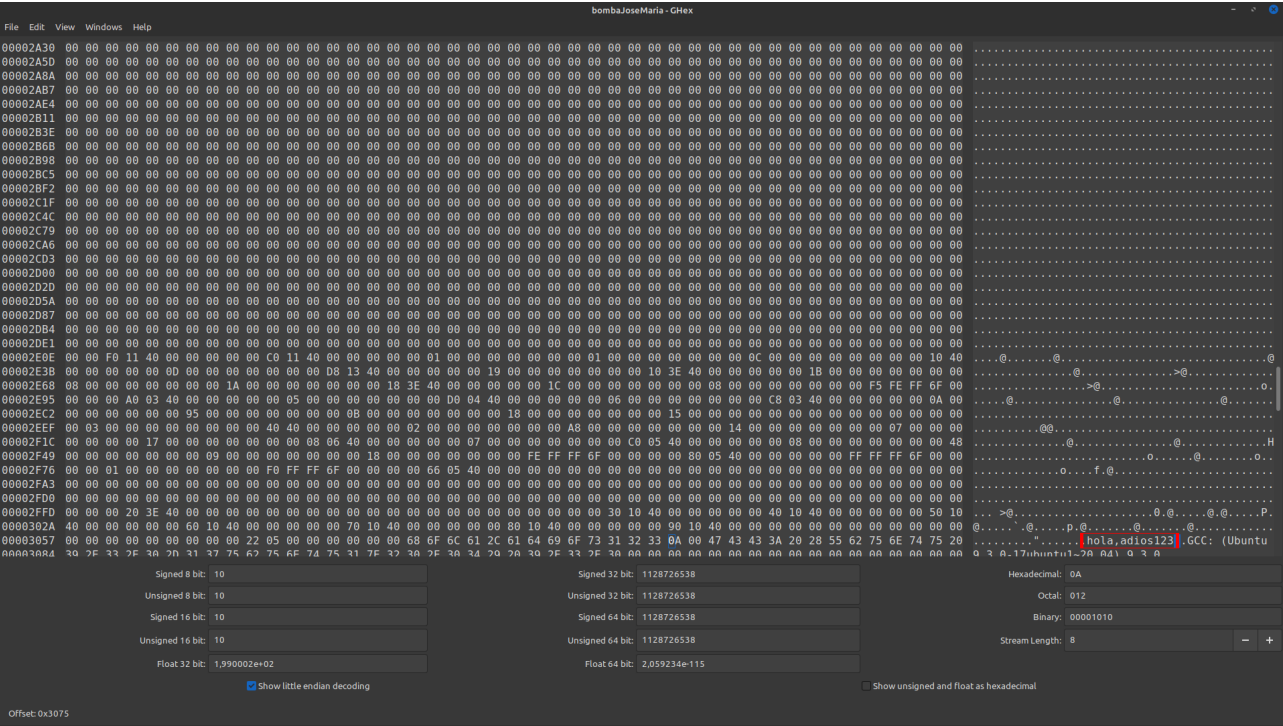
Para encontrar estas claves entre tanto hexadecimal, podemos usar la función de búsqueda de *ghex*. Si pulsamos Ctrl+F se nos abre el menú de búsqueda.

Vamos a empezar con la contraseña.

Escribimos “AverSiAdivina” en el menú de búsqueda y vemos que solo tenemos una coincidencia.

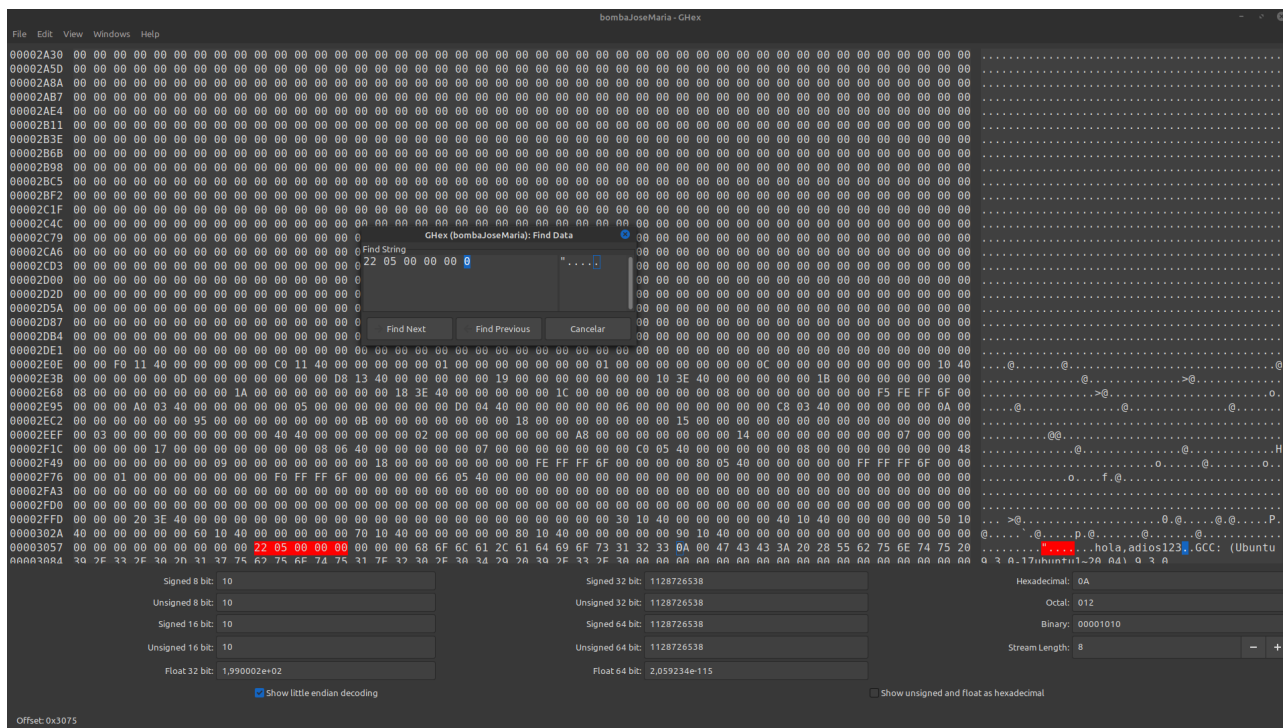


Para cambiar la contraseña, vamos a lo señalado en rojo a la derecha, clickamos en la primera letra y vamos escribiendo. Por ejemplo podemos escribir “hola,adios123”. Después de cambiarla, guardamos con **Ctrl+S**.



Para cambiar el PIN necesitaremos la calculadora de Ubuntu.
Pasaremos el PIN a hexadecimal con la calculadora en modo programación.
 $1314_{10} \rightarrow 522_{16}$
Una vez que tenemos el valor en hexadecimal, lo buscamos en el cuadro de búsquedas de *ghex*, pero en little-endian.

Es decir `00 00 00 05 22` \rightarrow `22 05 00 00 00`.



Como vemos solo tenemos 1 coincidencia. Si cambiamos los dígitos (es decir, pasamos, por ejemplo, de `22 05 00 00 00` a `23 05 00 00 00`) el PIN pasará a ser el nuevo valor de esa cadena (**Recordemos, en little-endian**).

Así pues, yo voy a cambiar el PIN a 1315.

The screenshot shows a hex editor window with a memory dump. The dump is organized into columns of hexadecimal values and their corresponding ASCII characters. The ASCII column shows a password field containing the text "hola,adios123" and a PIN field containing the text "1315". The hex values are in pairs, and the ASCII values are in single characters. The editor also displays various conversion options (Signed/Unsigned 8, 16, 32, 64 bits, Float 32/64 bit) and a checkbox for "Show little endian decoding".

Después de cambiar el PIN, guardamos con *Ctrl+S* y una vez que volvamos a ejecutar el programa, la nueva contraseña será “hola,adios123” y el nuevo PIN será 1315