

Explicación de las bombas desactivadas

David Infante Casas

75165296P

davidinfante@correo.ugr.es

Grupo de prácticas: D1

Se ha usado la herramienta gdb.

Las bombas desactivadas son de:

- Francisco Javier Bolívar
- Enrique Moreno Carmona

Bomba de Francisco Javier Bolívar

La contraseña y pin originales que se han de escribir son: “aaabbbbaaa\n” y 1234

La contraseña y pin modificados que lee el programa son: “bbbcccbbb\n” y 1238

Para descubrir las contraseñas y pines, lo primero que he comprobado ha sido si se encontraban las direcciones de memoria donde se guardan dichas variables en el código asm que ofrece gdb, y así ha sido, siendo 0x404068 para bbbcccbbb y 0x404074 para 1238.

```
0x40129a <main+108>    mov     $0xb,%edx
0x40129f <main+113>    lea     0x2dc2(%rip),%rsi        # 0x404068 <password>
0x4012a6 <main+120>    mov     %rax,%rdi
0x4012a9 <main+123>    callq  0x401030 <strncmp@plt>
0x4012ae <main+128>    test    %eax,%eax
0x4012b0 <main+130>    je      0x4012b7 <main+137>
0x4012b2 <main+132>    callq  0x401196 <boom>
0x4012b7 <main+137>    lea     -0x80(%rbp),%rax
0x4012bb <main+141>    mov     $0x0,%esi
0x4012c0 <main+146>    mov     %rax,%rdi
0x4012c3 <main+149>    callq  0x401070 <gettimeofday@plt>
0x4012c8 <main+154>    mov     -0x80(%rbp),%rdx
0x4012cc <main+158>    mov     -0x90(%rbp),%rax
0x4012d3 <main+165>    sub     %rax,%rdx
0x4012d6 <main+168>    mov     %rdx,%rax
0x4012d9 <main+171>    cmp     $0x3c,%rax
```

```
0x401342 <main+276>    mov     %eax,-0x98(%rbp)
0x401348 <main+282>    mov     -0x98(%rbp),%edx
0x40134e <main+288>    mov     0x2d20(%rip),%eax      # 0x404074 <passcode>
0x401354 <main+294>    cmp     %eax,%edx
0x401356 <main+296>    je      0x40135d <main+303>
0x401358 <main+298>    callq  0x401196 <boom>
0x40135d <main+303>    lea     -0x90(%rbp),%rax
0x401364 <main+310>    mov     $0x0,%esi
0x401369 <main+315>    mov     %rax,%rdi
0x40136c <main+318>    callq  0x401070 <gettimeofday@plt>
0x401371 <main+323>    mov     -0x90(%rbp),%rdx
0x401378 <main+330>    mov     -0x80(%rbp),%rax
0x40137c <main+334>    sub     %rax,%rdx
0x40137f <main+337>    mov     %rdx,%rax
0x401382 <main+340>    cmp     $0x3c,%rax
0x401386 <main+344>    jle     0x40138d <main+351>
```

A continuación he iniciado una ejecución del programa estableciendo un breakpoint en el main y probando como contraseña “hola”.

```

rax 0x7fffffffddc0 140737488346560 rdi 0x402dc2 4202950 rcs 0x0 0 rdx 0xffffffffbcb00 140737488346560
rsi 0x7fffffffddc0 140737488346560 r9 0x0 0 r10 0x405010 4214800 r11 0x246 582
r8 0x405075 4216437 r9 0x7fffffffdb80 140737353988992 cs 0x33 51 ss 0x2b 43
fs 0x0 0 gs 0x0 0 eflags 0x246 { PF ZF IF }

0x40122e <main>      push    %rbp
0x401254 <main+38>    mov     %rax,%rdi
0x401257 <main+41>    callq  0x401070 <gettimeofday@plt>
0x40125c <main+46>    lea     0xe65(%rip),%rdi    # 0x402dc8
0x401263 <main+53>    mov     $0x0,%eax
0x401268 <main+58>    callq  0x401060 <printf@plt>    # 0x404080 <stdin@GLIBC_2.2.5>
0x40126d <main+63>    mov     0x2dc(%rip),%rdx    # 0x404080 <stdin@GLIBC_2.2.5>
0x401274 <main+70>    lea     -0x70(%rbp),%rax
0x401278 <main+74>    mov     $0x64,%esi
0x40127d <main+79>    mov     %rax,%rdi
0x401280 <main+82>    callq  0x401080 <fgets@plt>
0x401285 <main+87>    test    %rax,%rax<main+46>
0x401288 <main+90>    je      0x40125c <main+46>
0x40128a <main+92>    lea     -0x70(%rbp),%rax
0x40128e <main+96>    mov     %rax,%rdi<caesarROT1>
0x401291 <main+99>    callq  0x4011ca <caesarROT1>

mainthread thread 0x7fffffffdb in: main
0x0000000000401246 in main ()
(gdb) nt
0x0000000000401268 in main ()
(gdb) nt
0x000000000040126d in main ()
(gdb) nt
0x0000000000401274 in main ()
(gdb) nt
0x0000000000401278 in main ()
(gdb) nt
0x000000000040127d in main ()
(gdb) nt
0x0000000000401280 in main ()
(gdb) nt
Introduce la contraseña: hola
0x0000000000401285 in main ()
(gdb) nt
0x0000000000401288 in main ()
(gdb) nt
0x000000000040128a in main ()
(gdb) nt
0x000000000040128e in main ()
(gdb) nt
0x0000000000401291 in main ()
(gdb) nt
0x0000000000401296 in main ()
(gdb) x/lsb $rdi
0x7fffffffddc0: "ipmb\n"
(gdb)

```

Avanzamos en la ejecución hasta que modifica la contraseña y comprobamos en el registro %rdi que nos devuelve “ipmb\n”. Pensando un poco, me doy cuenta que tiene el mismo número de caracteres que “hola” y además, cada carácter es “uno más” de su posición lógica en el abecedario, por tanto, si la clave es “bbbcccbbbb”, la que debemos introducir es **“aaabbbbaaa”**.

```

0x40122e <main>      push    %rbp
0x401254 <main+38>    mov     %rax,%rdi
0x401278 <main+74>    mov     $0x64,%esi
0x40127d <main+79>    mov     %rax,%rdi
> 0x401280 <main+82>    callq  0x401080 <fgets@plt>
0x401280 <main+82>    callq  0x401080 <fgets@plt>
0x401285 <main+87>    test    %rax,%rax<main+46>
0x401288 <main+90>    je      0x40125c <main+46>
0x40128a <main+92>    lea     -0x70(%rbp),%rax
0x40128e <main+96>    mov     %rax,%rdi<caesarROT1>
0x401291 <main+99>    callq  0x4011ca <caesarROT1>
> 0x401296 <main+104>   lea     -0x70(%rbp),%rax
0x40129a <main+108>    mov     $0xb,%edx
0x40129f <main+113>    lea     0x2dc2(%rip),%rsi    # 0x404068 <password>
0x4012a6 <main+120>    mov     %rax,%rdi
0x4012a9 <main+123>    callq  0x401030 <strncmp@plt>
0x4012ae <main+128>    test    %eax,%eax
0x4012b0 <main+130>    je      0x4012b7 <main+137>

0x0000000000401246 in main ()
(gdb) ni
0x0000000000401280 in main ()
(gdb) ni
Introduce la contraseña: hola
0x0000000000401285 in main ()
(gdb) ni
0x0000000000401288 in main ()
(gdb) ni
0x000000000040128a in main ()
(gdb) ni
0x000000000040128e in main ()
(gdb) ni
0x0000000000401291 in main ()
(gdb) ni
0x0000000000401296 in main ()
(gdb) x/lsb $rdi
0x7fffffffddc0: "ipmb\n"
(gdb)

```

```

rax    0x7fffffffdd0b 140737408346500 rdi    0x4020c8 4202096 rdx    0x0 0 rsi    0x0 0 r10    0xa61616162626261 7479860840 r11    0x7ffffffbca80 140737349724368
rsi    0x7fffffffdd0b 140737408346500 r12    0x7fffffffdd0b 140737408346500 r13    0x0 0 r15    0x405010 4214800
r8     0x40567a 4214442 r9     0x7fffffffdbb0 14073735398992 r10    0x405010 4214800 r11    0x240 582
r12    0x0 0 r13    0x0 0 r15    0x0 0
eflags 0x246 [ PF ZF IF ]

0x40122e <main>      push    %rbp
0x401254 <main>+38<  mov     %rax,%rdi
0x40122e <main>      push    %rbp
0x401254 <main>+38<  mov     %rax,%rdi
0x401257 <main>+41<  callq  0x40107b <gettimeofday@plt>
0x40125c <main>+46<  lea     0xe5(%rip),%rdi # 0x4020c8
0x401263 <main>+53<  mov     %eax,%eax
0x401268 <main>+58<  callq  0x401060 <printf@plt>
0x401268 <main>+58<  callq  0x401060 <printf@plt> # 0x404080 <stdout@GLIBC_2.2.5>
0x40126d <main>+63<  mov     %eax,%rdi # 0x404080 <stdout@GLIBC_2.2.5>
0x401274 <main>+70<  lea     -0x70(%rbp),%rax
0x401278 <main>+74<  mov     %eax,%eax
0x40127d <main>+79<  mov     %rax,%rdi
0x401280 <main>+82<  callq  0x401080 <fgets@plt>
0x401280 <main>+82<  callq  0x401080 <fgets@plt>
0x401285 <main>+87<  test    %rax,%rax <main>+46<
0x40128a <main>+92<  lea     -0x70(%rbp),%rax
0x40128e <main>+96<  mov     %rax,%rax
0x401291 <main>+99<  callq  0x40101e <caesar ROT13>
multi_thread 0x7fffffffdbb in: main
0x000000000401246 in: main ()
(gdb) nt
(gdb) nt
0x000000000401268 in: main ()
(gdb) nt
0x00000000040126d in: main ()
(gdb) nt
0x000000000401274 in: main ()
(gdb) nt
0x000000000401278 in: main ()
(gdb) nt
0x00000000040127d in: main ()
(gdb) nt
0x000000000401280 in: main ()
(gdb) nt
Introduce la contraseña: aaabbaaa
0x000000000401285 in: main ()
(gdb)

```

```
0x0000000000000037401240 toSmath (jne 0x0000000000000037401240)
(gdb) ni
0x000000000000004012cc in main ()
(gdb) ni
(gdb) ni
0x000000000000004012f0 in main ()
(gdb) ni
0x000000000000004012f5 in main ()
(gdb) ni
0x000000000000004012fc in main ()
(gdb) ni
0x000000000000004012ff in main ()
(gdb) ni
0x00000000000000401306 in main ()
(gdb) ni
0x0000000000000040130b in main ()
(gdb) ni
Introduce el pin: 1000
0x00000000000000401310 in main ()
(gdb)
```

Comprobamos el pin modificado en el registro %rbp-0x98 y nos devuelve 1004, podemos suponer rápidamente que el programa suma 4 a nuestro pin por tanto, si el pin con el que se compara es 1238, el pin debe ser 4 menos, es decir, 1234.

```
0x401337 <main+265>    jne     0x4012f5 <main+199>
0x401339 <main+267>    mov     -0x98(%rbp),%eax58>
0x40133f <main+273>    add     $0x4,%eax          # 0x4020fb
0x401342 <main+276>    mov     %eax,-0x98(%rbp)
> 0x401348 <main+282>    mov     -0x98(%rbp),%edx99_scanf@plt>
0x40134e <main+288>    mov     0x2d20(%rip),%eax  # 0x404074 <passcode>
0x0000000054401246 i94main (jne %eax,%edx 99>
(gdb) ni 56          96      e          35d      303
0x00000000004012cc in main ()
(gdb) ni
(gdb) ni
0x000000000040131d in main ()
(gdb) ni
0x0000000000401330 in main ()
(gdb) ni
0x0000000000401337 in main ()
(gdb) ni
0x0000000000401339 in main ()
(gdb) ni
0x000000000040133f in main ()
(gdb) ni
0x0000000000401342 in main ()
(gdb) ni
0x0000000000401348 in main ()
(gdb) x/lwu $rbp-0x98
0x7fffffffdd98: 1004
(gdb)
```

Como ya tenemos ambos, contraseña y pin, los probamos en una ejecución de la bomba.

La bomba es desactivada con éxito.

```
david@david-VirtualBox:~/Escritorio/EC/P4/2/bomba_FranciscoJavierBolivar$ ./bomba
Introduce la contraseña: aaabbbaaa
Introduce el pin: 1234
.....
... bomba desactivada ...
.....
david@david-VirtualBox:~/Escritorio/EC/P4/2/bomba_FranciscoJavierBolivar$
```


Para modificar la bomba y que la contraseña y pin introducidos den igual, vamos a modificar ambos saltos en el código ensamblador usando gdb y la función set write on. Modificamos el código como se muestra en la imagen.

```
0x4012e4 <main+182>    lea    0xdf9(%rip),%rdi    # 0x4020e4
0x4012eb <main+189>    mov    $0x0,%eax
0x4012f0 <main+194>    callq 0x401060 <printf@plt>
0x4012f5 <main+199>    lea    -0x98(%rbp),%rax
0x4012fc <main+206>    mov    %rax,%rsi
0x4012ff <main+209>    lea    0xdf2(%rip),%rdi    # 0x4020f8
0x401306 <main+216>    mov    $0x0,%eax
0x40130b <main+221>    callq 0x401090 <__isoc99_scanf@plt>
0x401310 <main+226>    mov    %eax,-0x94(%rbp)
0x401316 <main+232>    cmpl   $0x0,-0x94(%rbp)
0x40131d <main+239>    jne    0x401330 <main+258>
0x40131f <main+241>    lea    0xdd5(%rip),%rdi    # 0x4020fb
0x401326 <main+248>    mov    $0x0,%eax
0x40132b <main+253>    callq 0x401090 <__isoc99_scanf@plt>
0x401330 <main+258>    cmpl   $0x1,-0x94(%rbp)
0x401337 <main+265>    jne    0x4012f5 <main+199>
0x401339 <main+267>    mov    -0x98(%rbp),%eax
0x40133f <main+273>    add    $0x4,%eax
0x401342 <main+276>    mov    %eax,-0x98(%rbp)
0x401348 <main+282>    mov    -0x98(%rbp),%edx
0x40134e <main+288>    mov    0x2d20(%rip),%eax    # 0x404074 <passcode>
0x401354 <main+294>    cmp    %eax,%edx
0x401356 <main+296>    je     0x40135d <main+303>
0x401358 <main+298>    callq 0x401196 <boom>
0x40135d <main+303>    lea    -0x90(%rbp),%rax
0x401364 <main+310>    mov    $0x0,%esi
0x401369 <main+315>    mov    %rax,%rdi
0x40136c <main+318>    callq 0x401070 <gettimeofday@plt>
0x401371 <main+323>    mov    -0x90(%rbp),%rdx
0x401378 <main+330>    mov    -0x80(%rbp),%rax
0x40137c <main+334>    sub    %rax,%rdx
0x40137f <main+337>    mov    %rdx,%rax
```

exec No process In:

Leyendo símbolos desde bomba...(no se encontraron símbolos de depuración)hecho.

(gdb) layout asm

(gdb) set write on

(gdb) file bomba

Leyendo símbolos desde bomba...(no se encontraron símbolos de depuración)hecho.

(gdb) x/i 0x4012b0

0x4012b0 <main+130>: je 0x4012b7 <main+137>

(gdb) set *(char *) 0x4012b0=0xeb

(gdb) x/i 0x4012b0

0x4012b0 <main+130>: jmp 0x4012b7 <main+137>

(gdb) x/i 0x401356

0x401356 <main+296>: je 0x40135d <main+303>

(gdb) set *(char *) 0x401356=0xeb

(gdb) x/i 0x401356

0x401356 <main+296>: jmp 0x40135d <main+303>

(gdb) set write off

(gdb) quit

Probamos la bomba nueva.
Funciona.

```
david@david-VirtualBox:~/Escritorio/EC/P4/2$ ./bomba
Introduce la contraseña: qweq
Introduce el pin: 12312
.....
... bomba desactivada ...
.....
david@david-VirtualBox:~/Escritorio/EC/P4/2$
```

Bomba de Enrique Moreno Carmona

La contraseña y pin originales que se han de escribir son: “rdmcmnnckdr\n” y 1800

La contraseña y pin modificados que lee el programa son: “sendnoodles\n” y 1800

Al igual que antes, compruebo si están la password y el passcode siendo la dirección 0x601080 para la contraseña sendnoodles y 0x601060 para el pin 1800.

```
0x4007f1 <main+91>    lea    0x30(%rsp),%rbx
0x4007f6 <main+96>    mov    %rbx,%rdi
0x4007f9 <main+99>    callq 0x40075b <transform>
0x4007fe <main+104>   mov    $0xd,%edx
0x400803 <main+109>   lea    0x20085e(%rip),%rsi    # 0x601068 <password>
0x40080a <main+116>   mov    %rbx,%rdi
0x40080d <main+119>   callq 0x4005d0 <strncmp@plt>
0x400812 <main+124>   test   %eax,%eax
0x400814 <main+126>   je     0x40081b <main+133>
0x400816 <main+128>   callq 0x400727 <boom>
0x40081b <main+133>   lea    0x20(%rsp),%rdi
0x400820 <main+138>   mov    $0x0,%esi
0x400825 <main+143>   callq 0x4005f0 <gettimeofday@plt>
0x40082a <main+148>   mov    0x20(%rsp),%rax
0x40082f <main+153>   sub    0x10(%rsp),%rax
0x400834 <main+158>   cmp    $0x3c,%rax
```

```
0x400878 <main+226>   mov    $0x0,%eax
0x40087d <main+231>   callq 0x400620 <__isoc99_scanf@plt>
0x400882 <main+236>   cmp    $0x1,%ebx
0x400885 <main+239>   jne    0x40083f <main+169>
0x400887 <main+241>   mov    0x2007d3(%rip),%eax    # 0x601060 <passcode>
0x40088d <main+247>   cmp    %eax,0xc(%rsp)
0x400891 <main+251>   je     0x400898 <main+258>
0x400893 <main+253>   callq 0x400727 <boom>
0x400898 <main+258>   lea    0x10(%rsp),%rdi
0x40089d <main+263>   mov    $0x0,%esi
0x4008a2 <main+268>   callq 0x4005f0 <gettimeofday@plt>
0x4008a7 <main+273>   mov    0x10(%rsp),%rax
0x4008ac <main+278>   sub    0x20(%rsp),%rax
0x4008b1 <main+283>   cmp    $0x3c,%rax
0x4008b5 <main+287>   jle    0x4008bc <main+294>
0x4008b7 <main+289>   callq 0x400727 <boom>
```


Una vez hecho esto, comenzamos la ejecución igual que antes y en la contraseña probamos de nuevo “hola”. Comprobando el registro %rbx nos devuelve la cadena modificada como “ipmb\n”. Casualmente es exactamente lo mismo que nos devolvió la bomba anterior así que podemos asumir que lo que hace la bomba es lo mismo, dar el valor siguiente a cada carácter en el abecedario, por tanto, si la contraseña es sendnoodles, y pensamos un poco, obtenemos la cadena a introducir:

“rdmcmnnckdr”.

```

rax 0x1b 27 rsi 0x400a18 4196888 rcx 0x0 0
rax 0x4 4 rbx 0x7fffffffdb0 140737488346544 rcx 0x02 98
rdx 0x4 4 rsi 0x7fffffffdb0 140737488346544 rdi 0x5 5
r9 0x0 0 r10 0x002010 6299664 r8 0x602675 6391391
r9 0x7fffffffdb0 140737353988992 rip 0x4007d1 0x4007d1 <main+59> eflags 0x202 [ IF ]
cs 0x33 51 ss 0x2b 43 r11 0x11 17 r12 0x200 [ PF IF ]
es 0x0 0 fs 0x0 0 r13 0x10 16 r14 0x246 [ PF ZF IF ]

0: 0x400796 <main> push %rbx
0x4007c0 <main+42> lea 0x251(%rip),%rsi # 0x400a18
> 0x4007e7 <main+81> call 0x400600 <fgets@plt>
0x4007e7 <main+81> call 0x400600 <fgets@plt>
0x4007ec <main+86> test %rax,%rax<main+42>
0x4007ef <main+89> je 0x4007c0 <main+42>
0x4007f1 <main+93> lea 0x30(%rsp),%rbx
0x4007f6 <main+96> mov %rbx,%rdi<transform>
0x4007f9 <main+99> call 0x40075b <transform>
0x4007fe <main+104> mov 50x(%rdi,%rip),%rsi # 0x601068 <password>
multi-thre Thread 0x7fffffffdbb In: main L?? PC: 0x4007d1
0x40080a <main+110> mov %rbx,%rdi
0x40080d <main+113> call 0x4005d0 <strncmp@plt>
0x400812 <main+124> test %eax,%eax
0x400814 <main+126> je 0x40081b <main+133>
0x400816 <main+128> call 0x400727 <boom>
0x40081b <main+133> lea 0x20(%rsp),%rdi
0x400820 <main+138> mov 50x0,%esi
(gdb) nt
(gdb) nt
Introduce la contraseña: hola
0x0000000004007ec in main ()
(gdb) nt
0x0000000004007ef in main ()
(gdb) nt
0x0000000004007f1 in main ()
(gdb) nt
0x0000000004007f6 in main ()
(gdb) nt
0x0000000004007f9 in main ()
(gdb) x/15b $rbx
0x7fffffffdb0: "hola\n"
(gdb) nt
0x0000000004007fe in main ()
(gdb) x/15b $rbx
0x7fffffffdb0: "ipmb\n"
(gdb)

```

Ahora vemos el pin, probando como antes con 1000. Sin embargo, leyendo el código, vemos que no se pasa ninguna función que modifique al pin, esto quiere decir, que el pin es el mismo que se leyó en el passcode **1800**.

```

0x4007ec <main+86> test %rax,%rax),%rbx
0x40082f <main+153> sub 0x10(%rsp),%rax
0x400834 <main+158> cmp $0x3c,%rax
0x400838 <main+162> jle 0x40083f <main+169>
0x40083a <main+164> callq 0x400727 <boom>
0x40083f <main+169> lea 0x1ee(%rip),%rsi # 0x400a34
0x400846 <main+176> mov $0x1,%edi
0x40084b <main+181> mov $0x0,%eax
> 0x400850 <main+186> callq 0x400610 < printf_chk@plt>
0x400850 <main+186> callq 0x400610 <__printf_chk@plt>
0x400855 <main+191> lea 0xc(%rsp),%rsidi # 0x400a48
0x40085a <main+196> lea 0x1e7(%rip),%rdi # 0x400a48
0x400861 <main+203> mov $0x0,%eax< isoc99_scanf@plt>
> 0x400866 <main+208> callq 0x400620 < isoc99_scanf@plt>
0x400866 <main+208> callq 0x400620 <__isoc99_scanf@plt>
0x40086b <main+213> mov %eax,%ebx
0x40086d <main+215> test %eax,%eax
0x40086f <main+217> jne 0x400882 <main+236>
0x400871 <main+219> lea 0x1d3(%rip),%rdi # 0x400a4b
0x400878 <main+226> mov $0x0,%eax
0x40087d <main+231> callq 0x400620 <__isoc99_scanf@plt>
0x400882 <main+236> cmp $0x1,%ebx
0x400885 <main+239> jne 0x40083f <main+169>
> 0x400887 <main+241> mov 0x2007d3(%rip),%eax # 0x601060 <passcode>
0x40088d <main+247> cmp %eax,0xc(%rsp)
0x400891 <main+251> je 0x400898 <main+258>
0x400893 <main+253> callq 0x400727 <boom>
0x400898 <main+258> lea 0x10(%rsp),%rdi
0x40089d <main+263> mov $0x0,%esi
0x4008a2 <main+268> callq 0x4005f0 <gettimeofday@plt>
0x4008a7 <main+273> mov 0x10(%rsp),%rax236>
0x00000000ac4007af i78main (sub 0x20(%rsp),%rax
(gdb) ni b1 83 cmp $0x3c,%rax
0x000000000040082a in main ()
(gdb) ni
0x0000000000400861 in main ()
(gdb) ni
0x0000000000400866 in main ()
(gdb) ni
Introduce el pin: 1000
0x000000000040086b in main ()
(gdb) ni
0x000000000040086d in main ()
(gdb) ni
0x000000000040086f in main ()
(gdb) ni
0x0000000000400882 in main ()
(gdb) ni
0x0000000000400885 in main ()
(gdb) ni
0x0000000000400887 in main ()
(gdb)

```

Probamos a ejecutar la bomba con esta contraseña y pin.
Y se desactiva correctamente.

```

david@david-VirtualBox:~/Escritorio/EC/P4/2/bomba_Enrique_Moreno_Carmona$ ./bomba

Introduce la contraseña: rdmcmnckdr

Introduce el pin: 1800

.....
... bomba desactivada ...
.....

david@david-VirtualBox:~/Escritorio/EC/P4/2/bomba_Enrique_Moreno_Carmona$ █

```

Modificamos esta bomba de la misma forma que la anterior, tal y como se ve en la imagen.

```
0x400825 <main+143>    callq 0x4005f0 <gettimeofday@plt>
0x40082a <main+148>    mov    0x20(%rsp),%rax
0x40082f <main+153>    sub    0x10(%rsp),%rax
0x400834 <main+158>    cmp    $0x3c,%rax
0x400838 <main+162>    jle    0x40083f <main+169>
0x40083a <main+164>    callq 0x400727 <boom>
0x40083f <main+169>    lea    0x1ee(%rip),%rsi        # 0x400a34
0x400846 <main+176>    mov    $0x1,%edi
0x40084b <main+181>    mov    $0x0,%eax
0x400850 <main+186>    callq 0x400610 <__printf_chk@plt>
0x400855 <main+191>    lea    0xc(%rsp),%rsi
0x40085a <main+196>    lea    0x1e7(%rip),%rdi        # 0x400a48
0x400861 <main+203>    mov    $0x0,%eax
0x400866 <main+208>    callq 0x400620 <__isoc99_scanf@plt>
0x40086b <main+213>    mov    %eax,%ebx
0x40086d <main+215>    test   %eax,%eax
0x40086f <main+217>    jne    0x400882 <main+236>
0x400871 <main+219>    lea    0x1d3(%rip),%rdi        # 0x400a4b
0x400878 <main+226>    mov    $0x0,%eax
0x40087d <main+231>    callq 0x400620 <__isoc99_scanf@plt>
0x400882 <main+236>    cmp    $0x1,%ebx
0x400885 <main+239>    jne    0x40083f <main+169>
0x400887 <main+241>    mov    0x2007d3(%rip),%eax    # 0x601060 <passcode>
0x40088d <main+247>    cmp    %eax,0xc(%rsp)
0x400891 <main+251>    je     0x400898 <main+258>
0x400893 <main+253>    callq 0x400727 <boom>
0x400898 <main+258>    lea    0x10(%rsp),%rdi
0x40089d <main+263>    mov    $0x0,%esi
0x4008a2 <main+268>    callq 0x4005f0 <gettimeofday@plt>
0x4008a7 <main+273>    mov    0x10(%rsp),%rax
0x4008ac <main+278>    sub    0x20(%rsp),%rax
0x4008b1 <main+283>    cmp    $0x3c,%rax
```

exec No process in:

Leyendo símbolos desde bomba...(no se encontraron símbolos de depuración)hecho.

(gdb) layout asm

(gdb) set write on

(gdb) file bomba

Leyendo símbolos desde bomba...(no se encontraron símbolos de depuración)hecho.

(gdb) x/i 0x400814

0x400814 <main+126>: je 0x40081b <main+133>

(gdb) set *(char *) 0x400814=0xeb

(gdb) x/i 0x400814

0x400814 <main+126>: jmp 0x40081b <main+133>

(gdb) x/i 0x400891

0x400891 <main+251>: je 0x400898 <main+258>

(gdb) set *(char *) 0x400891=0xeb

(gdb) x/i 0x400891

0x400891 <main+251>: jmp 0x400898 <main+258>

(gdb) set write off

(gdb) quit

La probamos y funciona.

```
david@david-VirtualBox:~/Escritorio/EC/P4/2$ ./bomba
```

```
Introduce la contraseña: a
```

```
Introduce el pin: 1
```

```
.....
... bomba desactivada ...
.....
```

```
david@david-VirtualBox:~/Escritorio/EC/P4/2$
```