

Descodificación bomba Alejandro Menor Molinero

Noelia Escalera Mejías. Grupo A3

Al abrir el programa con gdb, vemos que tiene guardadas dos variables, una llamada password y otra passcode, vamos a ver su interior.

```
Register group: general
rax      0x4007f9 4196345      rbx      0x0      0
rcx      0x400980 4196736      rdx      0x7fffffffdd08 140737488346376
rsi      0x7fffffffddcf8 140737488346360      rdi      0x1      1
rbp      0x400980 0x400980 <_libc_csu_init>      rsp      0x7fffffffddc18 0x7fffffffddc18
r8       0x7ffff7dd0d80 140737351847296      r9       0x7ffff7dd0d80 140737351847296
r10      0x2      2      r11      0x7      7
r12      0x4006b0 4196016      r13      0x7fffffffddcf0 140737488346352
r14      0x0      0      r15      0x0      0
rip      0x4007f9 0x4007f9 <main>      eflags   0x246 [ PF ZF IF ]

B> 0x4007f9 <main>      push    %rbx
0x4007fa <main+1>      sub     $0xa0,%rsp
0x400801 <main+8>      mov     %fs:0x28,%rax
0x40080a <main+17>     mov     %rax,0x98(%rsp)
0x400812 <main+25>     xor     %eax,%eax
0x400814 <main+27>     lea     0x200855(%rip),%rdi      # 0x601070 <password>
0x40081b <main+34>     callq   0x4007d5 <codificacion_pw>      # 0x601068 <passcode>
0x400820 <main+39>     mov     0x200842(%rip),%edi      # 0x601068 <passcode>
0x400826 <main+45>     callq   0x4007ec <codificacion_pin>
0x40082b <main+50>     mov     %eax,0x200837(%rip)      # 0x601068 <passcode>

native process 4130 In: main      L??      PC: 0x4007f9
(gdb) br main
Punto de interrupci┐n 1 at 0x4007f9
(gdb) run
Starting program: /home/noelia/Escritorio/bomba_alejandro_menor

Breakpoint 1, 0x00000000004007f9 in main ()
(gdb) x/1sb 0x601070
0x601070 <password>:  "metamorfosis\n"
(gdb) x/ldw 0x601068
0x601068 <passcode>:  4444
(gdb) ]
```

Vemos que la contraseña vale “metamorfosis\n” y el pin 4444. Vamos a ver si estas son las claves verdaderas:

```
noelia@noelia-HP-ENVY-17-Notebook-PC:~/Escritorio$ ./bomba_alejandro_menor

Introduce la contraseña: metamorfosis

Introduce el pin: 4444

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

Efectivamente, son estos los credenciales y hemos desactivado la bomba. No obstante, el programa llama posteriormente a una función codificar, es decir, codifica las contraseñas pero las guarda sin codificar y se pueden averiguar sin descifrar la codificación, sin embargo, a mí me gustaría descifrarla.

Hemos introducido como contraseña “hola\n”. Si avanzamos hasta que se comparan las contraseñas y accedemos a los parámetros de strcmp. Como se ve en la captura de la siguiente página, compara “hola\n” con “metamorfosis\n”, es decir, la codificación consiste en cambiar la primera letra de la contraseña por una n.

```

Register group: general
rax      0xa      10      rbx      0xffffffffdb0  140737488346016
rcx      0xa616c6f  174156911  rdx      0xb5      181
rsi      0x601070  6295664    rdi      0xffffffffdb0  140737488346016
rbp      0x400980  0x400980 <_libc_csu_init>  rsp      0x7fffffffdb70  0x7fffffffdb70
r8       0x602675  6301301    r9       0x7ffff7fda500  140737353983232
r10      0x602010  6299664    r11      0x246      582
r12      0x4006b0  4196016    r13      0x7fffffffdf0  140737488346352
r14      0x0      0           r15      0x0      0
rip      0x400888  0x400888 <main+143>  eflags   0x246      [ PF ZF IF ]

0x400876 <main+125>  mov     %rbx,%rdi
0x400879 <main+128>  callq  0x4007d5 <codificacion_pw>
0x40087e <main+133>  lea     0x2007eb(%rip),%rsi  # 0x601070 <password>
0x400885 <main+140>  mov     %rbx,%rdi
> 0x400888 <main+143>  callq  0x400670 <strcmp@plt>
0x40088d <main+148>  mov     %eax,%edi
0x40088f <main+150>  mov     $0xffffffffffff,%rax
0x400896 <main+157>  test    %edi,%edi
0x400898 <main+159>  mov     $0x0,%rdi
0x40089f <main+166>  cmovne %rax,%rdi

native process 4130 In: main  L?? PC: 0x400888
(gdb) nt
0x00000000040087e in main ()
(gdb) nt
0x000000000400885 in main ()
(gdb) nt
0x000000000400888 in main ()
(gdb) x/1sb %rdi
0x7fffffffdb0: "\nola\n"
(gdb) x/1sb %rsi
0x601070 <password>: "\netamorfofis\n"
(gdb)

```

Como pin hemos introducido 1234. Si accedemos a passcode, ahora el pin vale 8888. En main+283 ha codificado nuestro pin también, ha guardado el resultado en 0xc(%rsp), y ahora nuestro pin vale 2468. De aquí podemos deducir que la codificación consiste en multiplicar el pin por 2, es decir, el pin vale 4444.

```

Register group: general
rax      0x9a4      2468      rbx      0x1      1
rcx      0x10      16         rdx      0x9a4      2468
rsi      0x1      1         rdi      0x4d2      1234
rbp      0x400980  0x400980 <_libc_csu_init>  rsp      0x7fffffffdb70  0x7fffffffdb70
r8       0x0      0         r9       0x0      0
r10      0x7ffff7b82cc0  140737349430464  r11      0x400afa  4197114
r12      0x4006b0  4196016    r13      0x7fffffffdf0  140737488346352
r14      0x0      0           r15      0x0      0
rip      0x40091d  0x40091d <main+292>  eflags   0x202      [ IF ]

0x40090b <main+274>  cmp     $0x1,%ebx
0x40090e <main+277>  jne     0x4008c8 <main+207>
0x400910 <main+279>  mov     0xc(%rsp),%edi
0x400914 <main+283>  callq  0x4007ec <codificacion_pin>
0x400919 <main+288>  mov     %eax,0xc(%rsp)
> 0x40091d <main+292>  cmp     0x200745(%rip),%eax  # 0x601068 <passcode>
0x400923 <main+298>  je      0x40092f <main+310>
0x400925 <main+300>  mov     $0xffffffff,%edi
0x40092a <main+305>  callq  0x400797 <momento_de_la_verdad>
0x40092f <main+310>  lea     0x10(%rsp),%rdi

native process 4989 In: main  L?? PC: 0x40091d
(gdb) x/ldw 0x601068
0x601068 <passcode>: 8888
(gdb) x/ldw 0xc+%rsp
0x7fffffffdb7c: 2468
(gdb)

```

Ahora vamos a cambiarle la contraseña, vamos a permitir la escritura en el ejecutable y vamos a volver a comprobar en qué direcciones están el password y el passcode

```

[ Register Values Unavailable ]

0x4007f9 <main>      push  %rbx
0x4007fa <main+1>    sub   $0xa0,%rsp
0x400801 <main+8>    mov   %fs:0x28,%rax
0x40080a <main+17>   mov   %rax,0x98(%rsp)
0x400812 <main+25>   xor   %eax,%eax
0x400814 <main+27>   lea   0x200855(%rip),%rdi    # 0x601070 <password>
0x40081b <main+34>   callq 0x4007d5 <codificacion_pw>
0x400820 <main+39>   mov   0x200842(%rip),%edi    # 0x601068 <passcode>
0x400826 <main+45>   callq 0x4007ec <codificacion_pin>
0x40082b <main+50>   mov   %eax,0x200837(%rip)    # 0x601068 <passcode>

exec No process in:
(gdb) layout regs
(gdb) set write on
(gdb) file bomba
bomba: No existe el archivo o el directorio.
(gdb) file bomba_alejandro_menor
Leyendo símbolos desde bomba_alejandro_menor...(no se encontraron símbolos de depuración)hecho.
(gdb) x/1sb 0x601070
0x601070 <password>:  "metamorfosis\n"
(gdb) x/ldw 0x601068
0x601068 <passcode>:  4444
(gdb)

```

Ahora cambiamos las contraseñas: el password a “cambiada\n” y el passcode a 2018.

```

[ Register Values Unavailable ]

0x4007f9 <main>      push  %rbx
0x4007fa <main+1>    sub   $0xa0,%rsp
0x400801 <main+8>    mov   %fs:0x28,%rax
0x40080a <main+17>   mov   %rax,0x98(%rsp)
0x400812 <main+25>   xor   %eax,%eax
0x400814 <main+27>   lea   0x200855(%rip),%rdi    # 0x601070 <password>
0x40081b <main+34>   callq 0x4007d5 <codificacion_pw>
0x400820 <main+39>   mov   0x200842(%rip),%edi    # 0x601068 <passcode>
0x400826 <main+45>   callq 0x4007ec <codificacion_pin>
0x40082b <main+50>   mov   %eax,0x200837(%rip)    # 0x601068 <passcode>

exec No process in:
(gdb) x/1sb 0x601070
0x601070 <password>:  "metamorfosis\n"
(gdb) x/ldw 0x601068
0x601068 <passcode>:  4444
(gdb) set{char[14]}0x601070="cambiada\n"
(gdb) set{int}0x601068=2018
(gdb) x/1sb 0x601070
0x601070 <password>:  "cambiada\n"
(gdb) x/ldw 0x601068
0x601068 <passcode>:  2018
(gdb)

```

Vamos a ver que los cambios se hayan resuelto correctamente:

```

noelia@noelia-HP-ENVY-17-Notebook-PC:~/Escritorio$ ./bomba_alejandro_menor
Introduce la contraseña: cambiada
Introduce el pin: 2018
.....
... bomba desactivada ...
.....

```

Luego se han hallado los resultados esperados.