

Descodificación bomba Antonio Cuadros Lapresta

Noelia Escalera Mejías. Grupo A3

Al abrir el programa con gdb nos encontramos con lo siguiente:

```
Register group: general
rax      0x40075b 4196187      rbx      0x0      0      140737488346376
rcx      0x4008c0 4196544      rdx      0x7fffffffdd08 140737488346376
rsi      0x7fffffffdcf8 140737488346360  rdi      0x1      1      0x7fffffffddc18 140737488346376
rbp      0x4008c0 0x4008c0 <__libc_csu_init>  rsp      0x7fffffffddc18 0x7fffffffddc18
r8       0x7ffff7dd0d80 140737351847296  r9       0x7ffff7dd0d80 140737351847296
r10      0x2      2      0x7 7
r12      0x400640 4195904  r13      0x7fffffffdcf0 140737488346352
r14      0x0      0      0x0 0
rip      0x40075b 0x40075b <main>  eflags   0x246 [ PF ZF IF ]

B> 0x40075b <main> push %rbx
0x40075c <main+1> sub $0xa0,%rsp
0x400763 <main+8> mov %fs:0x28,%rax
0x40076c <main+17> mov %rax,0x98(%rsp)
0x400774 <main+25> xor %eax,%eax
0x400776 <main+27> lea 0x10(%rsp),%rdi
0x40077b <main+32> mov $0x0,%esi
0x400780 <main+37> callq 0x4005f0 <gettimeofday@plt>
0x400785 <main+42> lea 0x1cc(%rip),%rsi # 0x400958
0x40078c <main+49> mov $0x1,%edi

native process 3145 In: main L?? PC: 0x40075b
(gdb) layout regs
(gdb) br main
Punto de interrupción 1 at 0x40075b
(gdb) run
Starting program: /home/noelia/Escritorio/bombaAntonioCuadrosLapresta

Breakpoint 1, 0x000000000040075b in main ()
(gdb) █
```

Avanzamos con ni hasta que nos pida la contraseña, nosotros hemos introducido “hola\n” como contraseña:

```
Register group: general
rax      0x1b      27      rbx      0x0      0      140737351842304
rcx      0x0      0      rdx      0x7fffffffdcfa00 140737351842304
rsi      0x64      100     rdi      0x7fffffffdb70 140737488346016
rbp      0x4008c0 0x4008c0 <__libc_csu_init>  rsp      0x7fffffffdb70 0x7fffffffdb70
r8       0x0      0      r9       0x0      0
r10      0x602010 6299664  r11      0x246      582     0x246 582
r12      0x400640 4195904  r13      0x7fffffffdcf0 140737488346352
r14      0x0      0      r15      0x0      0
rip      0x4007ac 0x4007ac <main+81>  eflags   0x206 [ PF IF ]

0x400785 <main+42> lea 0x1cc(%rip),%rsi # 0x400958
0x40078c <main+49> mov $0x1,%edi
0x400791 <main+54> mov $0x0,%eax
0x400796 <main+59> callq 0x400610 <__printf_chk@plt>
0x40079b <main+64> lea 0x30(%rsp),%rdi
0x4007a0 <main+69> mov 0x2008d9(%rip),%rdx # 0x601080 <stdin@@GLIBC_2.2.5>
0x4007a7 <main+76> mov $0x64,%esi
> 0x4007ac <main+81> callq 0x400600 <fgets@plt>
0x4007b1 <main+86> test %rax,%rax
0x4007b4 <main+89> je 0x400785 <main+42>

native process 3145 In: main L?? PC: 0x4007ac
0x0000000000400796 in main ()
(gdb) ni
0x000000000040079b in main ()
(gdb) ni
0x00000000004007a0 in main ()
(gdb) ni
0x00000000004007a7 in main ()
(gdb) ni
0x00000000004007ac in main ()
(gdb) ni
Introduce la contraseña: hola█
```

Seguimos avanzando y nos encontramos una variable llamada password, vemos su contenido:

Noelia Escalera Mejías

Grupo A3

```

--Register group: general--
rax      0x6      6      rbx      0x0      0
rcx      0xfffffffffffff9  -7      rdx      0xd      13
rsi      0x68      104      rdi      0x7fffffffdba6  140737488346022
rbp      0x4008c0  0x4008c0 < _libc_csu_init>  rsp      0x7fffffffdb70  0x7fffffffdb70
r8       0x7fffffffdba0  140737488346016  r9       0x7ffff7fda500  140737353983232
r10      0x602010  6299664      r11      0x246      582
r12      0x400640  4195904      r13      0x7fffffffdcf0  140737488346352
r14      0x0      0      r15      0x0      0
rip      0x4007f8  0x4007f8 <main+157>  eflags   0x246      [ PF ZF IF ]

0x4007e0 <main+133>  mov     %rdx,%rcx
0x4007e3 <main+136>  mov     %r8,%rdi
0x4007e6 <main+139>  repnz  scas %es:(%rdi),%al
0x4007e8 <main+141>  mov     %rcx,%rax
0x4007eb <main+144>  not     %rax
0x4007ee <main+147>  mov     %sil,0x2d(%rsp,%rax,1)
0x4007f3 <main+152>  mov     $0xd,%edx
> 0x4007f8 <main+157>  lea     0x200869(%rip),%rsi      # 0x601068 <password>
0x4007ff <main+164>  mov     %r8,%rdi
0x400802 <main+167>  callq   0x4005d0 <strncmp@plt>

native process 3145 In: main                                L??  PC: 0x4007f8
(gdb) ni
0x00000000004007eb in main ()
(gdb) ni
0x00000000004007ee in main ()
(gdb) ni
0x00000000004007f3 in main ()
(gdb) ni
0x00000000004007f8 in main ()
(gdb) x/1sb 0x601068
0x601068 <password>:  "rullpointen\n"
(gdb)

```

Según esta variable, la contraseña es “rullpointen\n”, vamos a comprobar si es así:

```

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

Introduce la contraseña: rullpointen

**
** BOOM!!!
**

```

Luego no es la contraseña, está codificada, vamos a intentar descifrar la codificación. Avanzamos en el programa hasta que llegamos a una función strncmp, si comprobamos los argumentos vemos lo siguiente:

```

--Register group: general--
rax      0xfffffffef  4294967279      rbx      0x0      0
rcx      0x72      114      rdx      0x0      0
rsi      0x601068  6295656      rdi      0x7fffffffdba0  140737488346016
rbp      0x4008c0  0x4008c0 < _libc_csu_init>  rsp      0x7fffffffdb70  0x7fffffffdb70
r8       0x7fffffffdba0  140737488346016  r9       0x7ffff7fda500  140737353983232
r10      0x3      3      r11      0xd      13
r12      0x400640  4195904      r13      0x7fffffffdcf0  140737488346352
r14      0x0      0      r15      0x0      0
rip      0x400807  0x400807 <main+172>  eflags   0x293      [ CF AF SF IF ]

0x4007eb <main+144>  not     %rax
0x4007ee <main+147>  mov     %sil,0x2d(%rsp,%rax,1)
0x4007f3 <main+152>  mov     $0xd,%edx
0x4007f8 <main+157>  lea     0x200869(%rip),%rsi      # 0x601068 <password>
0x4007ff <main+164>  mov     %r8,%rdi
0x400802 <main+167>  callq   0x4005d0 <strncmp@plt>
> 0x400807 <main+172>  test    %eax,%eax
0x400809 <main+174>  je      0x400810 <main+181>
0x40080b <main+176>  callq   0x400727 <boom>
0x400810 <main+181>  lea     0x20(%rsp),%rdi

native process 3145 In: main                                L??  PC: 0x400807
(gdb) ni
0x00000000004007ff in main ()
(gdb) ni
0x0000000000400802 in main ()
(gdb) ni
0x0000000000400807 in main ()
(gdb) x/1sb %rdi
0x7fffffffdba0: "aolh\n"
(gdb) x/1sb %rsi
0x601068 <password>:  "rullpointen\n"
(gdb)

```

Uno de los argumentos es la contraseña codificada anteriormente hallada y el otro es “aolh\n”:
nuestra contraseña (“hola\n”) con el primer y el último carácter (sin contar ‘\n’) intercambiados, es decir, nuestra contraseña codificada. Si aplicamos esta codificación a la contraseña del programa, nos queda “nullpointer\n”.

Ahora vamos a intentar hallar el pin. Vamos a seguir avanzando, para ello nos debemos saltar los distintos test para que no se active el boom:

```
Register group: general
rax      0x0      0
rcx      0x72     114
rsi      0x601068 6295656
rbp      0x4008c0 0x4008c0 <_libc_csu_init>
r8       0x7fffffffdb70 140737488346016
r10      0x3      3
r12      0x400640 4195904
r14      0x0      0
rip      0x400810 0x400810 <main+181>
eflags   0x246    [ PF ZF IF ]

0x400802 <main+167> callq 0x4005d0 <strcmp@plt>
0x400807 <main+172> test %eax,%eax
0x400809 <main+174> je 0x400810 <main+181>
0x40080b <main+176> callq 0x400727 <boom>
> 0x400810 <main+181> lea 0x20(%rsp),%rdi
0x400815 <main+186> mov $0x0,%esi
0x40081a <main+191> callq 0x4005f0 <gettimeofday@plt>
0x40081f <main+196> mov 0x20(%rsp),%rax
0x400824 <main+201> sub 0x10(%rsp),%rax
0x400829 <main+206> cmp $0x5,%rax

native process 3145 In: main
0x0000000000400807 in main ()
(gdb) x/1sb $rdi
0x7fffffffdb70: "aolh\n"
(gdb) x/1sb $rsi
0x601068 <password>: "nullpointer\n"
(gdb) set $eax=0
(gdb) ni
0x0000000000400809 in main ()
(gdb) ni
0x0000000000400810 in main ()
(gdb) █
```

```
Register group: general
rax      0x5      5
rcx      0x20c49ba5e353f7cf 2361183241434822607
rsi      0x0      0
rbp      0x4008c0 0x4008c0 <_libc_csu_init>
r8       0x7fffffffdb70 140737488346016
r10      0x7fffffffdb70 140737488345968
r12      0x400640 4195904
r14      0x0      0
rip      0x400839 0x400839 <main+222>
eflags   0x246    [ PF ZF IF ]

0x40082d <main+210> jle 0x400839 <main+222>
0x40082f <main+212> callq 0x400727 <boom>
0x400834 <main+217> cmp $0x1,%ebx
0x400837 <main+220> je 0x40087e <main+291>
> 0x400839 <main+222> lea 0x134(%rip),%rsi # 0x400974
0x400840 <main+229> mov $0x1,%edi
0x400845 <main+234> mov $0x0,%eax
0x40084a <main+239> callq 0x400610 <__printf_chk@plt>
0x40084f <main+244> lea 0xc(%rsp),%rsi
0x400854 <main+249> lea 0x12d(%rip),%rdi # 0x400988

native process 3145 In: main
0x000000000040081f in main ()
(gdb) ni
0x0000000000400824 in main ()
(gdb) ni
0x0000000000400829 in main ()
(gdb) set $rax=5
(gdb) ni
0x000000000040082d in main ()
(gdb) ni
0x0000000000400839 in main ()
(gdb) █
```

Cuando nos pide el pin, introducimos 1234:

Noelia Escalera Mejías

Grupo A3

```

--Register group: general--
rax      0x0      0
rcx      0x0      0
rsi      0x7fffffffdb7c  140737488345980
rbp      0x4008c0  0x4008c0  <_libc_csu_init>
r8        0x0      0
r10      0x400974  4196724
r12      0x400640  4195904
r14      0x0      0
rip      0x400860  0x400860  <main+261>
rbx      0x0      0
rdx      0x7ffff7dd18c0  140737351850176
rdi      0x400988  4196744
rsp      0x7fffffffdb70  0x7fffffffdb70
r9        0x7ffff7fda500  140737353983232
r11      0x246     582
r13      0x7fffffffdcf0  140737488346352
r15      0x0      0
eflags   0x206     [ PF IF ]

0x400839 <main+222> lea    0x134(%rip),%rsi    # 0x400974
0x400840 <main+229> mov    $0x1,%edi
0x400845 <main+234> mov    $0x0,%eax
0x40084a <main+239> callq 0x400610 <__printf_chk@plt>
0x40084f <main+244> lea    0xc(%rsp),%rsi
0x400854 <main+249> lea    0x12d(%rip),%rdi    # 0x400988
0x40085b <main+256> mov    $0x0,%eax
> 0x400860 <main+261> callq 0x400620 <_isoc99_scanf@plt>
0x400865 <main+266> mov    %eax,%ebx
0x400867 <main+268> test   %eax,%eax

native process 3145 In: main
0x000000000040084a in main ()
(gdb) ni
0x000000000040084f in main ()
(gdb) ni
0x0000000000400854 in main ()
(gdb) ni
0x000000000040085b in main ()
(gdb) ni
0x0000000000400860 in main ()
(gdb) ni
Introduce el pin: 1234

```

Si seguimos avanzando, encontramos una variable llamada passcode, vamos a comprobar su valor:

```

--Register group: general--
rax      0x4c8     1224
rcx      0x10     16
rsi      0x1      1
rbp      0x4008c0  0x4008c0  <_libc_csu_init>
r8        0x0      0
r10      0x7ffff7b82cc0  140737349430464
r12      0x400640  4195904
r14      0x0      0
rip      0x400889  0x400889  <main+302>
rbx      0x1      1
rdx      0x7ffff7dd18d0  140737351850192
rdi      0x0      0
rsp      0x7fffffffdb70  0x7fffffffdb70
r9        0x0      0
r11      0x40098a  4196746
r13      0x7fffffffdcf0  140737488346352
r15      0x0      0
eflags   0x212     [ AF IF ]

0x40086b <main+272> lea    0x119(%rip),%rdi    # 0x40098b
0x400872 <main+279> mov    $0x0,%eax
0x400877 <main+284> callq 0x400620 <_isoc99_scanf@plt>
0x40087c <main+289> jmp    0x400834 <main+217>
0x40087e <main+291> mov    0xc(%rsp),%eax
0x400882 <main+295> sub    $0xa,%eax
0x400885 <main+298> mov    %eax,0xc(%rsp)
> 0x400889 <main+302> cmp    0x2007d1(%rip),%eax    # 0x601060 <passcode>
0x40088f <main+308> je     0x400896 <main+315>
0x400891 <main+310> callq 0x400727 <boom>

native process 25564 In: main
(gdb) ni
0x000000000040087e in main ()
(gdb) ni
0x0000000000400882 in main ()
(gdb) ni
0x0000000000400885 in main ()
(gdb) ni
0x0000000000400889 in main ()
(gdb) x/ldw 0x601060
0x601060 <passcode>: 1234
(gdb)

```

Su valor es '1234', vamos a ver si este es el verdadero pin:

```

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

Introduce la contraseña: nullpointer

Introduce el pin: 1234

**
** BOOM!!!
**

```

Luego este no es el pin correcto y está codificado, vamos a intentar averiguar la codificación. Veamos qué se le pasa a cmp:

```

Register group: general
rax      0x4c8  1224
rcx      0x10  16
rsi      0x1    1
rbp      0x4008c0 0x4008c0 <__libc_csu_init>
r8        0x0    0
r10      0x7ffff7b82cc0 140737349430464
r12      0x400640 4195904
r14      0x0    0
rip      0x400889 0x400889 <main+302>
eflags   0x212  [ AF IF ]

0x40086b <main+272> lea 0x119(%rip),%rdi # 0x40098b
0x400872 <main+279> mov $0x0,%eax
0x400877 <main+284> callq 0x400620 <__isoc99_scanf@plt>
0x40087c <main+289> jmp 0x400834 <main+217>
0x40087e <main+291> mov 0xc(%rsp),%eax
0x400882 <main+295> sub $0xa,%eax
0x400885 <main+298> mov %eax,0xc(%rsp)
> 0x400889 <main+302> cmp 0x2007d1(%rip),%eax # 0x601060 <passcode>
0x40088f <main+308> je 0x400896 <main+315>
0x400891 <main+310> callq 0x400727 <boom>

native process 18509 In: main L?? PC: 0x400889
(gdb) ni
0x000000000040087e in main ()
(gdb) ni
0x0000000000400882 in main ()
(gdb) ni
0x0000000000400885 in main ()
(gdb) ni
0x0000000000400889 in main ()
(gdb) x/ldw 0xc+$rsp
0x7fffffffdb7c: 1224
(gdb)

```

Como vemos se comparará con passcode un número ‘1224’, que es nuestro pin codificado. La codificación consiste en restarle 10 al número, luego podemos deducir que su pin original es ‘1244’, vamos a comprobar que esto sea correcto:

```

noelia@noelia-HP-ENVY-17-Notebook-PC:~/Escritorio$ ./bombaAntonioCuadrosLapresta
Introduce la contraseña: nullpointer

Introduce el pin: 1244

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

```

Luego efectivamente hemos hallado los credenciales correctos.

Ahora vamos a cambiar el password y el passcode, como en la otra bomba que he desactivado, voy a hacerlo en un archivo a parte. Tenemos que cambiar las direcciones 0x601068 (password) y 0x601060 (passcode), he cogido estas direcciones de las anteriores capturas.

```

[ Register Values Unavailable ]

0x40075b <main> push %rbx
0x40075c <main+1> sub $0xa0,%rsp
0x400763 <main+8> mov %fs:0x28,%rax
0x40076c <main+17> mov %rax,0x98(%rsp)
0x400774 <main+25> xor %eax,%eax
0x400776 <main+27> lea 0x10(%rsp),%rdi
0x40077b <main+32> mov $0x0,%esi
0x400780 <main+37> callq 0x4005f0 <gettimeofday@plt>
0x400785 <main+42> lea 0x1cc(%rip),%rsi # 0x400958
0x40078c <main+49> mov $0x1,%edi

exec No process In: L?? PC: ??
(gdb) layout regs
(gdb) x/1sb 0x601068
0x601068 <password>: "nullpointer\n"
(gdb) x/ldw 0x601060
0x601060 <passcode>: 1234
(gdb)

```

Vamos a cambiar la contraseña por “cambio\n” y el passcode por ‘123’, para ello habrá que introducir los credenciales codificados, es decir: “oambic\n” y ‘113’.

```
[ Register Values Unavailable ]

0x40075b <main>      push  %rbx
0x40075c <main+1>     sub   $0xa0,%rsp
0x400763 <main+8>     mov   %fs:0x28,%rax
0x40076c <main+17>    mov   %rax,0x98(%rsp)
0x400774 <main+25>    xor   %eax,%eax
0x400776 <main+27>    lea   0x10(%rsp),%rdi
0x40077b <main+32>    mov   $0x0,%esi
0x400780 <main+37>    callq 0x4005f0 <gettimeofday@plt>
0x400785 <main+42>    lea   0x1cc(%rip),%rsi      # 0x400958
0x40078c <main+49>    mov   $0x1,%edi

exec No process in:
(gdb) layout regs
(gdb) x/1sb 0x601068
0x601068 <password>:  "rullpointen\n"
(gdb) x/ldw 0x601060
0x601060 <passcode>:  1234
(gdb) set write on
(gdb) file bombaAntonioCuadrosLapresta_modificada
Leyendo símbolos desde bombaAntonioCuadrosLapresta_modificada...(no se encontraron símbolos de depuración)hecho.
(gdb) set {char[13]}0x601068="oambic\n"
(gdb) set {int}0x601060=113
(gdb)
```

Comprobamos que lo hemos cambiado correctamente:

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

Introduce la contraseña: cambio

Introduce el pin: 123

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

Luego está correctamente cambiado.