

Practica 4: Bombas resueltas

- Bomba Raúl Soria González**

Contraseña:xsr900 Pincodex:1998

En primer lugar hice un br *main y run hasta llegar a la llamada de strcmp, llamé a p (char*) 0x601080 que tiene la etiqute de password y me mostró la contraseña que es: xsr900.

```
Actividades - Terminal - vie 12:05
juse@juse-OMEN-by-HP-Laptop: ~/Escritorio/Universidad_ubuntu/EC/Practica4/Bombas_resueltas/bomba_Raul_Soria_Gonzalez

Archivo Editar Ver Buscar Terminal Ayuda

Register group: general
rax 0x7fffffffcd90 140737488346256 rdx 0x0 0 rcx 0xa6f746ef547961 751947680457193825
rbp 0x400a30 0x400a30 <_libc_csu_init> rsi 0x7fffffffcd90 140737488346256 rdi 0x7fffffffcd90 140737488346256
r9 0x7fffffffdf500 140737354003712 rsp 0x7fffffffcd40 0x7fffffffcd40 r8 0x602679 6301305
r12 0x400720 4196128 r10 0x602010 6299664 r11 0x246 582
r15 0x0 0 r13 0x7fffffffdd00 140737488346592 r14 0x0 0
cs 0x33 51 rip 0x4008a0 0x4008a0 <main+101> eflags 0x206 [ PF IF ]
es 0x0 0 fs 0x0 0 ds 0x0 0 gs 0x0 0

0x400860 <main+37> callq 0x4006c0 <gettimeofday@plt>
0x400865 <main+42> lea 0x30c(%rip),%rsi # 0x400b78
0x40086c <main+49> mov $0x1,%edi
0x400871 <main+54> mov $0x0,%eax
0x400876 <main+59> callq 0x4006f0 <_printf_chk@plt>
0x40087b <main+64> lea 0x50(%rsp),%rdi
0x400880 <main+69> mov 0x200809(%rip),%rdx # 0x601090 <stdin@GLIBC_2.2.5>
0x400887 <main+76> mov $0x64,%esi
0x40088c <main+83> callq 0x4006d0 <fgets@plt>
0x400891 <main+86> test %rax,%rax
0x400894 <main+89> je 0x4008a5 <main+42>
0x400896 <main+91> lea 0x50(%rsp),%rdi
0x40089b <main+96> mov $0x8,%edx
> 0x4008a0 <main+101> lea 0x2007d2(%rip),%rsi # 0x601080 <password>
0x4008a7 <main+108> callq 0x400690 <strcmp@plt>
0x4008ac <main+113> test %eax,%eax

Native process 5334 In: main
(gdb) nt
0x00000000040088c in main ()
0x000000000400891 in main ()
(gdb) nt
0x000000000400894 in main ()
0x000000000400896 in main ()
(gdb) p (char*) ($rdi+0x50)
$1 = 0x7fffffffddc1 "\n"
(gdb) p (char*) ($rsp+0x50)
$2 = 0x7fffffffddc90 "raytonto\n"
(gdb) nt
0x00000000040089b in main ()
0x0000000004008a0 in main ()
(gdb) p (char*) (0x2007d9+$rip)
$3 = 0x601079 <password+1> "\a"
(gdb) p (char*) 0x601080
$4 = 0x601080 <password> "xsr900\n"
(gdb) 
```

Tras probar la contraseña te pide que le des la hora, el minuto y segundo en que la introdujiste, usando gdb y con el br *main+300 ves que p (char*) (0x47+\$rsp) y te devuelve 12:23:13 que es lo que he puesto yo p (char*) (0x3f+\$rsp) te devuelve 12:20:37 12:23:13 y entonces vemos cual es el que hay que poner, que es 12:20:37 el momento exacto.

```

Register group: general
rax 0x7fffffffcd87 140737488346247 rbx 0x7fffffffcd4b 140737488346101 rcx 0x33313a33 858864179 rdx 0x7fffffffdd18d0 140737351850192
rsi 0x7fffffffcd7f 140737488346239 rdi 0x7fffffffcd37 140737488346247 rbp 0x400a30 0x400a30 <_libc_csu_init> rsp 0x7fffffffcd40 0x7fffffffcd40
rbx 0 0 r9 0x7fffffffdf500 140737354003712 r10 0x7fffffffdf500 140737354003712 r11 0x246 582
r12 0x400720 4190128 r13 0x7fffffffdd0 140737488346592 r14 0x0 0 r15 0x0 0
rip 0x400973 0x400973 <main+312> eflags 0x206 [ PF IF ] cs 0x33 51 ss 0x2b 43
ds 0x0 0 es 0x0 0 fs 0x0 0 gs 0x0 0

0x40093f <main+260> mov $0x1,%edi
0x400944 <main+265> mov $0x9,%eax
0x400949 <main+270> callq 0x400e0f <__printf_chk@plt>
0x40094e <main+275> lea 0x47(%rsp),%rdi
0x400953 <main+280> mov 0x200736(%rip),%rdx # 0x4001090 <stdinc@GLIBC_2.2.5>
0x40095a <main+287> mov $0x9,%esi
0x40095f <main+292> callq 0x400e0d <fgetc@plt>
0x400964 <main+297> test %rax,%rax
0x400967 <main+300> je 0x400938 <main+253>
0x400969 <main+302> lea 0x3f(%rsp),%rsi
0x40096c <main+307> lea 0x47(%rsp),%rdi
0x400973 <main+312> mov $0x9,%edi
0x400978 <main+317> callq 0x400e09 <strncmp@plt>
0x40097d <main+322> test %eax,%eax
0x40097f <main+324> je 0x400968 <main+331>
0x400981 <main+326> callq 0x400807 <boom>

native process 8009 In: main
(gdb) nl
0x0000000000400953 in main ()
0x000000000040095a in main ()
0x000000000040095f in main ()
0x0000000000400964 in main ()
(gdb) nl
0x0000000000400967 in main ()
(gdb) nl
0x0000000000400969 in main ()
0x000000000040096e in main ()
0x0000000000400973 in main ()
(gdb) p (int) (0x3f+%rsp)
$3 = 9809
(gdb) p (char*) (0x3f+%rsp)
$4 = 0x7fffffffcd7f "12:20:3712:23:13"
(gdb) p (char*) (0x47+%rsp)
$5 = 0x7fffffffcd07 "12:23:13"
(gdb)

```

Saltado ya el pasado de la hora, vemos como compara el pin introducido con `p*(int*)(0xc+$rsp)` y vemos que es 1998.

- Bomba Alberto Villanueva Copado**

Contraseña:contrasenia Pincod:1221

Seguimos el mismo procedimiento de la anterior bomba, ponemos un `br *main` y run. Con ni llegamos hasta la dirección `0x400962` y observamos que el valor del registro `%rsi` es contrasenia.

Tras obtener la cotraseña seguimos hasta la dirección `0x400a5e` y ahí se compara el pin, hacemos `p (int) $eax` y obtenemos 1221 que es el pincod

- Bomba María Alcázar Marcos**

Contraseña:Rorrecta Pincod:

```

Actividades  Terminal
juse@juse-OMEN-by-HP-Laptop: ~/Escritorio/Universidad_ubuntu/EC/Practica4/Bombas_resueltas/Maria Alcazar Marcos

--Register group: general
rax 0x00000000 140737488346160 3rbx 0x0 0 rcx 0xa616c6f 174156911 rdx 0x0 8
rsi 0x001068 6295656 rdi 0x7fffffffdc30 140737488346160 rbp 0x400890 0x400890 <__libc_csu_init> rsp 0x7fffffffdc00 0x7fffffffdc00
r8 0x002675 6301301 r9 0x7fffffffdf500 140737354003712 r10 0x002010 6299664 r11 0x240 582
r12 0x000640 4195904 r13 0x7fffffffdd80 140737488346496 r14 0x0 0 r15 0x0 0
rip 0x4007ce 0x4007ce <main+115> eflags 0x206 [ PF IF ] cs 0x33 51 ss 0x2b 43
ds 0x0 0 es 0x0 0 fs 0x0 0 gs 0x0 0

0x4007a7 <main+76> mov $0x64,%esi
0x4007ac <main+81> callq 0x400600 <fgets@plt>
0x4007b1 <main+86> test %rax,%rax
0x4007b4 <main+89> je 0x4007b5 <main+42>
0x4007b6 <main+91> movb $0x52,0x2008ab(%rip) # 0x601068 <password>
0x4007bd <main+98> lea 0x30(%rsp),%rdi
0x4007c2 <main+103> mov $0x3,%edx
0x4007c7 <main+108> lea 0x20089a(%rip),%rsi # 0x601068 <password>
0x4007ce <main+115> callq 0x4005d0 <strncmp@plt>
0x4007f3 <main+120> test %eax,%eax
0x4007d5 <main+122> je 0x4007fc <main+129>
0x4007d7 <main+124> callq 0x400727 <boom>
0x4007dc <main+129> lea 0x20(%rsp),%rdi
0x4007e1 <main+134> mov $0x0,%esi
0x4007e6 <main+139> callq 0x4005f0 <gettimeofday@plt>
0x4007eb <main+144> mov 0x20(%rsp),%rax

native process 6980 In: main
0x0000000000400796 in main ()
0x000000000040079b in main ()
(gdb) nl
0x00000000004007a8 in main ()
0x00000000004007a7 in main ()
0x00000000004007ac in main ()
0x00000000004007b1 in main ()
(gdb) nl
0x00000000004007b4 in main ()
0x00000000004007b6 in main ()
(gdb) nl
0x00000000004007bd in main ()
0x00000000004007c2 in main ()
0x00000000004007c7 in main ()
0x00000000004007ce in main ()
(gdb) p (char*) 0x601068
$10 = 0x601068 <password> "Rorrecta"
(gdb)

```

Ejecutamos un run con un br *main y tras pedirnos la contraseña justo antes de la comparación de cadenas por strncmp se hace un mov y un lea, usamos p(char*) 0x601068 y observamos que la contraseña es Rorrecta.

```

Actividades  Terminal
juse@juse-OMEN-by-HP-Laptop: ~/Escritorio/Universidad_ubuntu/EC/Practica4/Bombas_resueltas/Maria Alcazar Marcos

--Register group: general
rax 0x1 1 rbx 0x1 1 rcx 0x10 16 rdx 0x7fffffffdd80 140737351850192
rsi 0x1 1 rdi 0x0 0 rbp 0x400890 0x400890 <__libc_csu_init> rsp 0x7fffffffdc00 0x7fffffffdc00
r8 0x0 0 r9 0x0 0 r10 0x7ffff7b82cc0 140737349430464 r11 0x400a0a 4196874
r12 0x000640 4195904 r13 0x7fffffffdd80 140737488346496 r14 0x0 0 r15 0x0 0
rip 0x40084a 0x40084a <main+239> eflags 0x246 [ PF Zf If ] cs 0x33 51 ss 0x2b 43
ds 0x0 0 es 0x0 0 fs 0x0 0 gs 0x0 0

0x40084a <main+239> imul $0x3,0xc(%rsp),%eax
0x40084f <main+244> cmp 0x20080b(%rip),%eax # 0x601060 <passcode>
0x400855 <main+250> je 0x40085c <main+257>
0x400857 <main+252> callq 0x400727 <boom>
0x40085c <main+257> lea 0x10(%rsp),%rdi
0x400861 <main+262> mov $0x0,%esi
0x400866 <main+267> callq 0x4005f0 <gettimeofday@plt>
0x40086b <main+272> mov 0x10(%rsp),%rax
0x400870 <main+277> sub 0x20(%rsp),%rax
0x400875 <main+282> cmp $0x5,%rax
0x400879 <main+286> jle 0x400880 <main+293>
0x40087b <main+288> callq 0x400727 <boom>
0x400880 <main+293> callq 0x400741 <defused>
0x400885 nopw %cs:0x0(%rax,%rax,1)
0x40088f nop
0x400890 <__libc_csu_init> push %r15

native process 12559 In: main
(gdb) layout regs
(gdb) p (int) 0x601060
$1 = 6295648
(gdb) p (int) 0x601060
$2 = 23331
(gdb)

```

Vamos a br *main+239 y se ve que multiplica por 3 el valor, si hacemos p {int} 0x601060 nos da 23331 que pone como etiqueta passcode, sin embargo sabemos que ha sido multiplicado por 3, por lo tanto el verdadero pincode es p {int} 0x601060 = 23331 dividido entre 3 que sale 7777.