

Desactivar Bomba Digital

David Martínez Díaz

Bomba RLF 2020:

- 1.- En primer lugar es ejecutarlo, donde vemos que tenemos que averiguar su contraseña y pin correspondiente.

```
dmartinez01@dmartinez01-VirtualBox:~/Escritorio/FalIn/bomba_RLF_2020$ ./bomba_RLF_2020

Introduce la contraseña: hola

      _ _ _ _ _  
    _(_(_(_(_(_(_(  
   (_(_(_(_(_(_(_(  
  (_(_(_(_(_(_(_(_(  
     ~~~~|_|_|~~~~  
         | |  
         ; ;  
-----| |-----  
*****  
***;||BOOM!!***  
*****  
dmartinez01@dmartinez01-VirtualBox:~/Escritorio/FalIn/bomba_RLF_2020$
```

- 2.- Para ello debemos introducirnos en dicho ejecutable a través del debug e ir mirando los registros para comprobar y obtener las contraseñas.

→ gdb bomba

→ layout regs

Una vez estamos debugueando, hay que buscar el lugar exacto donde se comparan las contraseñas para poder continuar con el ejercicio, donde si nos damos cuenta, esta se realiza en el string `compare`, en la línea (`*main+111`).

```

Register group: general
rax    0x64    100      rbx
rdx    0x9     9        rsi
rbp    0x4008a0 0x4008a0 <_libc_csu_init>    rsp
r9     0x7ffff7fe2500 140737354016000    r10
r12    0x400640 4195904   r13
r15    0x0     0        rip
cs     0x33    51       ss
es     0x0     0        fs

0x4007b4 <main+89>    je     0x400785 <main+42>
0x4007b6 <main+91>    mov    $0x0,%eax
0x4007bb <main+96>    cmp    $0x63,%eax
0x4007be <main+99>    jle    0x4007df <main+132>
0x4007c0 <main+101>   lea    0x30(%rsp),%rdi
0x4007c5 <main+106>   mov    $0x9,%edx
0x4007ca <main+111>   lea    0x200897(%rip),%rsi    # 0x001068 <password>
0x4007d1 <main+118>   callq  0x4005d0 <strncmp@plt>
0x4007d6 <main+123>   test   %eax,%eax
0x4007d8 <main+125>   je     0x4007f3 <main+152>
0x4007da <main+127>   callq  0x400727 <boom>
0x4007df <main+132>   movslq %eax,%rcx
0x4007e2 <main+135>   movzbl 0x30(%rsp,%rcx,1),%ebx
0x4007e7 <main+140>   lea    0x2(%rbx),%edx

native process 7145 In: main
(gdb) b *main+111
Punto de interrupción 1 at 0x4007ca
(gdb) run
Starting program: /home/dmartinez01/Escritorio/Falin/bomba_RLF_2020/bomba_RLF_20
20

Breakpoint 1, 0x00000000004007ca in main ()
(gdb)

```

Por tanto, para sacar su contraseña debo ver los valores que hay en el registro %rsi, viendo así su contraseña correspondiente:

```

0x4007b4 <main+89>      je      0x400785 <main+42>
0x4007b6 <main+91>      mov     $0x0,%eax
0x4007bb <main+96>      cmp     $0x63,%eax
0x4007be <main+99>      jle     0x4007df <main+132>
0x4007c0 <main+101>     lea     0x30(%rsp),%rdi
0x4007c5 <main+106>     mov     $0x9,%edx
0x4007ca <main+111>     lea     0x200897(%rip),%rsi      # 0x601068 <password>
0x4007d1 <main+118>     callq   0x4005d0 <strncmp@plt>
0x4007d6 <main+123>     test    %eax,%eax
> 0x4007d8 <main+125>     je      0x4007f3 <main+152>
0x4007da <main+127>     callq   0x400727 <boom>
0x4007df <main+132>     movslq  %eax,%rcx
0x4007e2 <main+135>     movzbl  0x30(%rsp,%rcx,1),%ebx
0x4007e7 <main+140>     lea     0x2(%rbx),%edx

Active process 7145 In: main
db) b *main+111
nto de interrupci n 1 at 0x4007ca
db) run
Starting program: /home/dmartinez01/Escritorio/Fal n/bomba_RLF_2020/bomba_RLF_20

Breakpoint 1, 0x0000000004007ca in main ()
db) ni
0000000004007d1 in main ()
db) ni
0000000004007d6 in main ()
db) ni
0000000004007d8 in main ()
db) x/s $rsi
601068 <password>:  "tchcfkg|\f"
db)

```

Esto quiere decir que la contraseña esta encriptada, y no vamos a poder poner directamente el resultado de %rsi, para ello podemos comprobar con nuestra contraseña anterior y mirar en el registro %rdi para ver dicha comparación y calcular esa razón correspondiente para saber cuánto hay que sumarlo o restarle a esta.

Por ello si introducimos la contraseña “hola” y nos introducimos en el registro %rdi obtenemos lo siguiente:

[illegible]

Donde nuestra nueva contraseña encriptada seria “jqnc”, para saber la razón de encriptación simplemente basta saber qué valor llega desde la primera letra de nuestra contraseña hasta la primera letra de la contraseña encriptada:

“hola” ---> “jqnc”;

“h” ---> “i”

Para sacarlo nos vamos a la tabla ASCII y vemos sus correspondientes valores:

“h” = 104 // “j” = 106

Con esto llegamos a la conclusión de que la razón es 2, y simplemente para saber su contraseña hay que restarle 2 a cada letra.

Consiguiendo así la contraseña encriptada:
"tchcfkg|" ---> "rafadiez";

Vamos a comprobarlo:

```
dmartinez01@dmartinez01-VirtualBox: ~/Escritorio/Falin/bomba_RLF_2020
Archivo Editar Ver Buscar Terminal Ayuda
dmartinez01@dmartinez01-VirtualBox:~/Escritorio/Falin/bomba_RLF_2020$ ./bomba_RLF_2020
Introduce la contraseña: rafadiez
Introduce el pin:
```

Vemos que la contraseña era correcta pero se nos presenta otra problema, ahora nos pide un pin, veamos entonces de nuevo en el gdb, donde se encuentra este.

Como podemos ver se realiza otro string compare el cual vamos a analizar para saber si ahí se realiza la comparación entre pines. Analizando las sentencias llegamos hasta el main+260, donde se realiza una comparación entre los registros con la orden "jne":

```
--Register group: general
rax      0x1      1      rbx      0x0
rdx      0x7ffff7dcf8d0  140737351842000  rsi      0x0
rbp      0x4008a0  0x4008a0 <__libc_csu_init>  rsp      0x0
r9        0x0      0      r10     0x0
r12       0x400640  4195904  r13     0x0
r15       0x0      0      r1p     0x0
cs        0x33     51      ss      0x0
es        0x0      0      fs      0x0

0x400847 <main+236>  jne  0x40085a <main+255>
0x400849 <main+238>  lea  0x2bb(%rip),%rdi      # 0x400b0b
0x400850 <main+245>  mov  $0x0,%eax
0x400855 <main+250>  callq 0x400620 <__isoc99_scanf@plt>
0x40085a <main+255>  cmp  $0x1,%ebx
0x40085d <main+258>  jne  0x400817 <main+188>
+> 0x40085f <main+260>  mov  0x2007fb(%rip),%eax  # 0x601060 <passcode>
0x400865 <main+266>  cmp  %eax,0xc(%rsp)
0x400869 <main+270>  je   0x400870 <main+277>
0x40086b <main+272>  callq 0x400727 <boom>
0x400870 <main+277>  lea  0x10(%rsp),%rdi
0x400875 <main+282>  mov  $0x0,%esi
0x40087a <main+287>  callq 0x4005f0 <gettimeofday@plt>
0x40087f <main+292>  mov  0x10(%rsp),%rax

active process 7192 In: main
gdb) b *main+260
unto de interrupci n 1 at 0x40085f
gdb) run
tarting program: /home/dmartinez01/Escritorio/Falin/bomba_RLF_2020/bomba_RLF_2020
reakpoint 1, 0x00000000040085f in main ()
gdb)
ntroduce el pin: 1000
```

Donde si miramos bien las sentencias y hacemos un par de next instructions vemos como sacan un dato de la pila en la direcci n de (%rip+0x2d10), y lo mete en el registro %eax, si lo mostramos vemos que:

```
Breakpoint 1, 0x00000000040085f in main ()
(gdb) ni
0x000000000400865 in main ()
(gdb) ni
0x000000000400869 in main ()
(gdb) print $eax
$1 = 1202
(gdb)
```

Vamos a comprobar si este pin es el correcto:

```
dmartinez01@dmartinez01-VirtualBox:~/Escritorio/Falín/bomba_RLF_2020$ ./bomba_RLF_2020
Introduce la contraseña: rafadiez
Introduce el pin: 1202
(●_●)
... bomba desactivada ...
dmartinez01@dmartinez01-VirtualBox:~/Escritorio/Falín/bomba_RLF_2020$
```

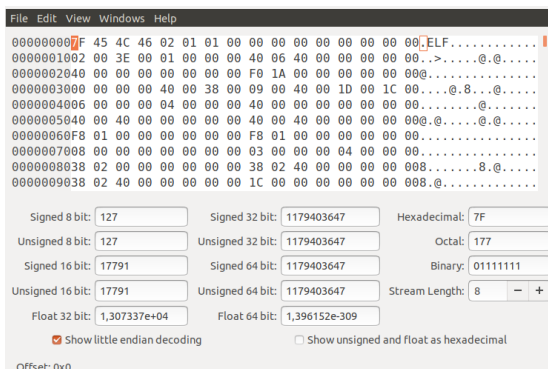
Como hemos podido comprobar el pin que habíamos obtenido era correcto y no se encontraba encriptado por lo que hemos podido desactivar la bomba. Por lo que esta sería la manera de desactivar la bomba.

- **Modificar la contraseña y pin de la bomba:**

Para modificar la contraseña vamos a utilizar el comando ghex que permite ver el código de manera hexadecimal:

“ghex bomba_MRG_2020”

Donde nos aparecerá lo siguiente:



Una vez estamos dentro, tenemos que buscar dicha contraseña, que en mi caso se encuentra encriptada:

```
.....@.....
..@.....@.....@.....@.....&.....6.....
.....tchcfkg|.GCC:
(Ubuntu 7.5.0-3ubuntu1~18.04) 7.5.0.....
.....8.....
```

Una vez que la hemos sacado vemos su Offset, simplemente si la modificamos debería cambiar (teniendo en cuenta la razón de encriptación).
Por ejemplo, si yo cambio la contraseña de esta manera:

“Vacaroja” → “xceptqlc”

```
dmartinez01@dmartinez01-VirtualBox:~/Escritorio/Falin/bomba_RLF_2020$ ./bomba_RLF_2020
Introduce la contraseña: vacaroja
Introduce el pin: █
```

Vemos que si me acepta la contraseña y la hemos podido modificar con éxito.
Para modificar el pin, vamos a emplear otra vez el comando ghex:

Signed 8 bit:	<input type="text" value="-78"/>
Unsigned 8 bit:	<input type="text" value="178"/>
Signed 16 bit:	<input type="text" value="1202"/>
Unsigned 16 bit:	<input type="text" value="1202"/>
Float 32 bit:	<input type="text" value="1,684361e-42"/>
<input checked="" type="checkbox"/> Show little endian decoding	

Una vez encontramos nuestro pin, probamos a cambiarlo poniendo F4 por ejemplo, quedando:

Signed 8 bit:	<input type="text" value="-14"/>
Unsigned 8 bit:	<input type="text" value="242"/>
Signed 16 bit:	<input type="text" value="1266"/>
Unsigned 16 bit:	<input type="text" value="1266"/>
Float 32 bit:	<input type="text" value="1,774044e-42"/>
<input checked="" type="checkbox"/> Show little endian decoding	

Por ultimo vamos a comprobar si funciona:

```
dmartinez01@dmartinez01-VirtualBox:~/Escritorio/Falin/bomba_RLF_2020$ ./bomba_RLF_2020
Introduce la contraseña: vacaroja
Introduce el pin: 1266
      (●_●)
.....
... bomba desactivada ...
.....
dmartinez01@dmartinez01-VirtualBox:~/Escritorio/Falin/bomba_RLF_2020$
```

Y confirmamos que se ha modificado con éxito.