



ugr

Universidad
de Granada



DEACTIVAR BOMBA PROFESOR NUMERO 15

Curso 2016/2017
José Antonio Padial Molina

Índice

INFORMACIÓN	2
ENCONTRAR LA CONTRASEÑA.....	2
ENCONTRAR EL PIN.....	5
CAMBIAR LA CONTRASEÑA.....	6
CAMBIAR EL PIN.....	7

INFORMACIÓN

ORIGINAL

contraseña: IMwYLtpa\n

pin: 3687

MODIFICADA

contraseña: jajajaja

pin: 3598

Nombre de la bomba: bombaProf5_15

Nombre del ejecutable de la bomba modificada: nuevo

ENCONTRAR LA CONTRASEÑA

Vamos a utilizar el GDB para resolver esta bomba, vamos a mostrar el procedimiento paso a paso en las siguientes imágenes. Hay que tener en cuenta a lo largo de la práctica que Intel™ es Little Endian, con lo cual los números en hexadecimal se lee de derecha a izquierda. Lo primero es realizar un *break main* para crear un punto de parada en el comienzo del main. Tras crear este punto realizamos un *run* para poder usar *disas* para ver el programa desensamblado para saber dónde poner el siguiente break.

```
jose@jose-AspireV3-571G:~/Escritorio/bombaProf5$ gdb bombaProf5_15
GNU gdb (Ubuntu 7.11.1-0ubuntu1~16.04) 7.11.1
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
Para las instrucciones de informe de errores, vea:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Leyendo símbolos desde bombaProf5_15...(no se encontraron símbolos de depuración)hecho.
(gdb) break main
Punto de interrupción 1 at 0x0804865f
(gdb) disas
Ningún marco seleccionado.
(gdb)
Ningún marco seleccionado.
(gdb) run
Starting program: /home/jose/Escritorio/bombaProf5/bombaProf5_15

Breakpoint 1, 0x0804865f in main ()
(gdb) disas
Dump of assembler code for function main:
   0x0804865c <+0>:  push    %ebp
   0x0804865d <+1>:  mov     %esp,%ebp
=>  0x0804865f <+3>:  and     $0xffffffff0,%esp
   0x08048662 <+6>:  push    %edi
   0x08048663 <+7>:  push    %ebx
   0x08048664 <+8>:  sub     $0x98,%esp
   0x0804866a <+14>: mov     %gs:0x14,%eax
   0x08048670 <+20>: mov     %eax,0x8c(%esp)
   0x08048677 <+27>: xor     %eax,%eax
```

```

0x08048670 <+20>: mov    %eax,0x8c(%esp)
0x08048677 <+27>: xor    %eax,%eax
0x08048679 <+29>: movl   $0x0,0x4(%esp)
0x08048681 <+37>: lea    0x1c(%esp),%eax
0x08048685 <+41>: mov    %eax,(%esp)
0x08048688 <+44>: call   0x80484ac <gettimeofday@plt>
0x0804868d <+49>: movl   $0x80488ea,0x4(%esp)
0x08048695 <+57>: movl   $0x1,(%esp)
0x0804869c <+64>: call   0x804847c <__printf_chk@plt>
0x080486a1 <+69>: mov    0x804a3a0,%eax
0x080486a6 <+74>: mov    %eax,0x8(%esp)
0x080486aa <+78>: movl   $0x64,0x4(%esp)
0x080486b2 <+86>: lea    0x28(%esp),%ebx
0x080486b6 <+90>: mov    %ebx,(%esp)
0x080486b9 <+93>: call   0x804848c <fgets@plt>
---Type <return> to continue, or q <return> to quit---
0x080486be <+98>: mov    $0x804a260,%edi
0x080486c3 <+103>: mov    $0x0,%eax
0x080486c8 <+108>: mov    $0xffffffff,%ecx
0x080486cd <+113>: repnz  scas %es:(%edi),%al
0x080486cf <+115>: not    %ecx
0x080486d1 <+117>: sub    $0x1,%ecx
0x080486d4 <+120>: mov    %ecx,0x8(%esp)
0x080486d8 <+124>: movl   $0x804a260,0x4(%esp)
0x080486e0 <+132>: mov    %ebx,(%esp)
0x080486e3 <+135>: call   0x80484dc <strncmp@plt>
0x080486e8 <+140>: test   %eax,%eax
0x080486ea <+142>: je     0x80486f1 <main+149>
0x080486ec <+144>: call   0x804860e <boom>
0x080486f1 <+149>: movl   $0x0,0x4(%esp)
0x080486f9 <+157>: lea    0x14(%esp),%eax
0x080486fd <+161>: mov    %eax,(%esp)
0x08048700 <+164>: call   0x80484ac <gettimeofday@plt>
0x08048705 <+169>: mov    0x14(%esp),%eax
0x08048709 <+173>: sub    0x1c(%esp),%eax
0x0804870d <+177>: cmp    $0x3c,%eax
0x08048710 <+180>: jle    0x8048717 <main+187>
0x08048712 <+182>: call   0x804860e <boom>
0x08048717 <+187>: movl   $0x8048905,0x4(%esp)
---Type <return> to continue, or q <return> to quit---
0x0804871f <+195>: movl   $0x1,(%esp)
0x08048726 <+202>: call   0x804847c <__printf_chk@plt>
0x0804872b <+207>: lea    0x24(%esp),%eax
0x0804872f <+211>: mov    %eax,0x4(%esp)
0x08048733 <+215>: movl   $0x804891c,(%esp)
0x0804873a <+222>: call   0x80484cc <__isoc99_scanf@plt>
0x0804873f <+227>: mov    0x24(%esp),%eax
0x08048743 <+231>: cmp    0x804a274,%eax
0x08048749 <+237>: je     0x8048750 <main+244>
0x0804874b <+239>: call   0x804860e <boom>
0x08048750 <+244>: movl   $0x0,0x4(%esp)
0x08048758 <+252>: lea    0x1c(%esp),%eax
0x0804875c <+256>: mov    %eax,(%esp)
0x0804875f <+259>: call   0x80484ac <gettimeofday@plt>
0x08048764 <+264>: mov    0x1c(%esp),%eax
0x08048768 <+268>: sub    0x14(%esp),%eax
0x0804876c <+272>: cmp    $0x3c,%eax
0x0804876f <+275>: jle    0x8048776 <main+282>
0x08048771 <+277>: call   0x804860e <boom>
0x08048776 <+282>: call   0x80485c0 <defused>
0x0804877b <+287>: mov    0x8c(%esp),%edx
0x08048782 <+294>: xor    %gs:0x14,%edx
0x08048789 <+301>: je     0x8048795 <main+313>
---Type <return> to continue, or q <return> to quit---
0x0804878b <+303>: nop
0x0804878c <+304>: lea    0x0(%esi,%eiz,1),%esi
0x08048790 <+308>: call   0x80484bc <__stack_chk_fail@plt>
0x08048795 <+313>: add    $0x98,%esp
0x0804879b <+319>: pop    %ebx
0x0804879c <+320>: pop    %edi
0x0804879d <+321>: mov    %ebp,%esp
0x0804879f <+323>: pop    %ebp
0x080487a0 <+324>: ret
End of assembler dump.

```

Busco la función strcmp (comparador de cadenas)

```
0x080486d1 <+117>: sub    $0x1,%ecx
0x080486d4 <+120>: mov    %ecx,0x8(%esp)
0x080486d8 <+124>: movl   $0x804a260,0x4(%esp)
0x080486e0 <+132>: mov    %ebx,(%esp)
0x080486e3 <+135>: call   0x80484dc <strcmp@plt>
0x080486e8 <+140>: test   %eax,%eax
0x080486ea <+142>: je     0x80486f1 <main+149>
0x080486ec <+144>: call   0x804860e <boom>
0x080486f1 <+149>: movl   $0x0,0x4(%esp)
0x080486f9 <+157>: lea    0x14(%esp),%eax
```

Después borro el primer break point(break del main). Hago un nuevo *break* en la dirección del strcmp. Ejecuto de nuevo la orden *run* y solicitará la contraseña, introduzco "prueba\n". Vuelvo a ejecutar *disas*.

```
(gdb) break *0x080486e3
Punto de interrupción 2 at 0x080486e3
(gdb) delete 1
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/jose/Escritorio/bombaProf5/bombaProf5_15
Introduce la contraseña: prueba

Breakpoint 2, 0x080486e3 in main ()
(gdb) disas

0x080486aa <+78>: movl   $0x64,0x4(%esp)
0x080486b2 <+86>: lea    0x28(%esp),%ebx
0x080486b6 <+90>: mov    %ebx,(%esp)
0x080486b9 <+93>: call   0x804848c <fgets@plt>
---Type <return> to continue, or q <return> to quit---
0x080486be <+98>: mov    $0x804a260,%edi
0x080486c3 <+103>: mov    $0x0,%eax
0x080486c8 <+108>: mov    $0xffffffff,%ecx
0x080486cd <+113>: repnz  scas %es:(%edi),%al
0x080486cf <+115>: not    %ecx
0x080486d1 <+117>: sub    $0x1,%ecx
0x080486d4 <+120>: mov    %ecx,0x8(%esp)
0x080486d8 <+124>: movl   $0x804a260,0x4(%esp)
0x080486e0 <+132>: mov    %ebx,(%esp)
0x080486e3 <+135>: call   0x80484dc <strcmp@plt>
0x080486e8 <+140>: test   %eax,%eax
0x080486ea <+142>: je     0x80486f1 <main+149>
0x080486ec <+144>: call   0x804860e <boom>
0x080486f1 <+149>: movl   $0x0,0x4(%esp)
0x080486f9 <+157>: lea    0x14(%esp),%eax
0x080486fd <+161>: mov    %eax,(%esp)
0x08048700 <+164>: call   0x80484ac <gettimeofday@plt>
```

A continuación, observo el contenido de esa dirección y encuentro la contraseña (password):
lMwYLtpa\n

```
End of assembler dump.
(gdb) x/s 0x804a260
0x804a260 <password>: "lMwYLtpa\n"
```

ENCONTRAR EL PIN

Una vez obtenida la contraseña ya se puede proceder a buscar la instrucción *cmp*, donde se van a comparar el pin que desbloquea la bomba con el introducido.

```
0x0804873a <+222>: call 0x80484cc <__isoc99_scanf@plt>
0x0804873f <+227>: mov 0x24(%esp),%eax
0x08048743 <+231>: cmp 0x804a274,%eax
0x08048749 <+237>: je 0x8048750 <main+244>
0x0804874b <+239>: call 0x804860e <boom>
0x08048750 <+244>: movl $0x0,0x4(%esp)
0x08048758 <+252>: lea 0x1c(%esp),%eax
0x0804875c <+256>: mov %eax,(%esp)
```

Ejecutamos la instrucción *info b* para ver los break point que hay actualmente, borramos el break point de *strcmp* y realizamos un nuevo break point en la dirección de la instrucción *cmp*. Volvemos a ejecutar el programa. Primero nos pedirá la contraseña (obtenida anteriormente), si es correcta nos pedirá el pin. En este ejemplo vamos a introducir 1234. Realizamos un *info r* para ver el estado de los registros. En los cuales podemos comprobar que en ellos no se ha almacenado la contraseña. Pero podemos observar como realiza *cmp*. Miramos con que compara la contraseña y obtenemos la contraseña, debido a que pone **passcode**. Como hemos mencionado anteriormente esta en hexadecimal. Convertida a decimal es 3687.

```
(gdb) info b
Num      Type          Disp Enb Address      What
2        breakpoint     keep y  0x080486e3 <main+135>
breakpoint already hit 1 time
(gdb) delete 2
(gdb) break *0x08048743
Punto de interrupción 3 at 0x08048743
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/jose/Escritorio/bombaProf5/bombaProf5_15
Introduce la contraseña: lMwYLtpa
Introduce el código: 1234

Breakpoint 3, 0x08048743 in main ()
(gdb) info r
eax          0x4d2      1234
ecx          0x1       1
edx          0xf7fb587c -134522756
ebx          0xffffcfe8 -12312
esp          0xffffcfc0 0xffffcfc0
ebp          0xffffd068 0xffffd068
esi          0xf7fb4000 -134529024
edi          0x804a26a 134521450
eip          0x08048743 0x08048743 <main+231>
eflags      0x246      [ PF ZF IF ]
cs          0x23      35
ss          0x2b      43
ds          0x2b      43
es          0x2b      43
fs          0x0       0
gs          0x63      99
(gdb) x/10x 0x804a274
0x804a274 <passcode>: 0x67 0x0e 0x00 0x00 0xd9 0x1a 0x00 0x00
0x804a27c <numrand2>: 0x40 0x26
(gdb)
```


Con lo cual ya estaría desactivada la bomba, lo comprobamos.

```
jose@jose-AspireV3-571G:~$ cd Escritorio/bombaProf5/
jose@jose-AspireV3-571G:~/Escritorio/bombaProf5$ ./bombaProf5_15
Introduce la contraseña: lMwYLtpa
Introduce el código: 3687
.....
... bomba desactivada ...
.....
jose@jose-AspireV3-571G:~/Escritorio/bombaProf5$
```

CAMBIAR LA CONTRASEÑA

Usamos GHEx para buscar donde se encuentra la contraseña. Cuando la encontramos la podemos cambiar por lo que queramos.

The screenshot shows the GHEx application interface. The top section displays a hex dump of memory, with the ASCII column on the right showing the password 'lMwYLtpa' at offset 0x125A. The bottom section contains various data type conversion fields and checkboxes.

Field	Value
Signed 8 bit:	97
Unsigned 8 bit:	97
Signed 16 bit:	2657
Unsigned 16 bit:	2657
Float 32 bit:	5,631283e+14
Signed 32 bit:	1476397665
Unsigned 32 bit:	1476397665
Signed 64 bit:	1476397665
Unsigned 64 bit:	1476397665
Float 64 bit:	1,879392e+21
Hexadecimal:	61
Octal:	141
Binary:	01100001
Stream Length:	8

☒ Show little endian decoding ☐ Show unsigned and float as hexadecimal

Offset: 0x1267; 0x8 bytes from 0x1260 to 0x1267 selected

CAMBIAR EL PIN

Para cambiar el pin realizamos la misma operación que hemos hecho con la contraseña. Pero como es un número hay que convertir el decimal a hexadecimal.

bombaProf5_15 - CHex

000010E0 29 24 2D 3C 21 3B 26 31 27 2F 35 2B 27 24 28 29 6C 47 3B 74 6C 60 3C 44 67 2C 54
000010FB 61 3E 5A 5D 7B B7 F6 43 4B C4 8A 8B 87 DC 24 20 0C 20 31 86 8C 2B 28 5D 78 C7 26
00001116 E9 C4 45 82 BE CA 67 84 DC 1C 1E 1F 00 00 06 A7 00 00 12 A9 00 00 8F C9 00 00 15
00001131 EF 00 00 9D 8E 00 00 87 B9 00 00 08 DF 00 00 BA 97 8A 9E 9B 95 85 1B B1 AE 97 E8
0000114C CA BD 07 9D 53 73 63 7B 4F 6B 70 6E 73 53 42 6F 67 65 50 5D 4F 46 5F 7B 44 63 5A
00001167 5D 76 6F 75 6D 6F 52 5D 52 F2 EC BD BC A4 95 B5 B1 A3 1A 01 96 80 1C FE E0 A3 BD
00001182 00 00 C2 05 00 00 B5 BB 00 00 E9 AD 00 00 8D C1 00 00 8F CD 00 00 94 E7 00 00 92
0000119D C8 00 00 F8 C0 45 AA A8 B6 41 61 43 21 BB E4 33 8E DC 7E 31 5F 3B 48 6B BF 20 9E
000011B8 0C 53 4E 78 2B 9E 1B 80 69 23 5F 60 2E 46 55 66 68 6D 49 3D 2A 6B 29 47 28 30 3D
000011D3 32 25 27 3D 2B 2A 29 31 20 2A 35 36 3D 37 28 26 30 38 29 34 30 36 32 36 32 23 39
000011EE 21 38 51 42 69 68 69 5C 41 39 67 3D 38 4C 7D 5C 70 7C 7D 1B 97 0E BD 08 6C 80
00001209 77 8D 64 EF BE AF F5 B4 EC A8 A9 28 1B EC 8D 81 98 17 25 2B CE 72 94 A7 8D 00 00
00001224 08 FA 00 00 F2 B8 00 00 E8 80 00 00 D7 F6 00 00 9E D8 00 00 8C C3 00 00 A8 B0 00
0000123F 00 EC EC BF D5 EE 8C C4 E6 01 AB 9E B4 C9 AB B1 03 6F 65 73 6A 5C 57 67 43 4F 70
0000125A 4C 5A 49 63 6C 63 6C 4D 77 59 4C 74 70 61 0A 00 58 73 78 59 44 42 72 70 0A 00 67
00001275 0E 00 00 D9 1A 00 00 40 26 00 00 63 41 71 59 65 78 51 69 0A 00 00 00 00 00 00
00001290 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 60 5E 6C 71 72 6D 5B 77 6E 67 6E
000012AB 45 51 42 64 53 C4 B2 98 A4 16 A5 F7 9C 14 10 92 99 90 EB E6 EC E1 94 00 00 18 05
000012C6 00 00 CA 9E 00 00 E3 16 00 00 DA B2 00 00 10 ED 00 00 86 F0 00 00 E1 A2 00 00 4B
000012E1 36 7A F7 74 37 B4 8E 9A 04 23 06 AE A0 E6 96 4E 79 3E 38 FB 00 C1 21 35 EB 10 AF
000012FC A9 16 55 B2 4F 2C 5D 4D 7E 42 42 41 22 5F 6B 52 39 44 35 75 23 30 25 23 23 22 3B
00001317 26 2B 2E 39 26 3F 2B 2C 27 3E 3F 3C 21 3C 36 37 20 27 33 3B 28 25 2F 26 2F 65 5A
00001332 28 4D 69 5E 46 24 49 7E 70 29 5C 23 71 60 E3 81 A3 C3 7C D0 07 36 26 11 2E BD C8
0000134D 0F CB 58 ED 12 28 4A 27 DE FE C7 77 4A 4F A6 DE D4 D3 1A AC C1 00 00 1A 91 00 00

Signed 8 bit: 103 Signed 32 bit: 3687 Hexadecimal: 67
Unsigned 8 bit: 103 Unsigned 32 bit: 3687 Octal: 147
Signed 16 bit: 3687 Signed 64 bit: 3687 Binary: 01100111
Unsigned 16 bit: 3687 Unsigned 64 bit: 3687 Stream Length: 8
Float 32 bit: 5,166587e-42 Float 64 bit: 1,458448e-310
☒ Show little endian decoding ☐ Show unsigned and float as hexadecimal
Offset: 0x1274