Práctica 4: Desactivación de Bombas

Desactivación Bomba de Alejandro Rubio

Pablo Olivares Martínez

`contraseña = "movimientohelicoidal"`

`pin = 1001`

A continuación, vamos a tratar de desactivar la bomba de Alejandro Rubio. Primero, usemos la fuunción ltrace para facilitar la detección de cadenas y sus funciones. Para probarla, escribiremos **"aaaaaaaaa"** y ver si nuestra cadena o la suya sufre alguna modificación:

```
pablo@laptop:~/Bombas$ ltrace -i -S ./bomba_ARM_2020
                                                    ...
[0x7fb0c61b71e7] SYS_write(1, "Introduce la contrase\303\261a: ", 26Introduce
la contraseña: ) = 26
[0x7fb0c61b7142] SYS_read(0aaaaaaaaaaa
, "aaaaaaaaaaa\n", 1024) = 12
[0x40121b] <... fgets resumed> "aaaaaaaaaaa\n", 100, 0x7fb0c6291980) =
0x7ffe740e6fa0
[0x40122c] puts("\n Contrase\303\261a correcta " <unfinished ...>
[0x7fb0c61b71e7] SYS_write(1, "\n", 1
)           = 1
[0x7fb0c61b71e7] SYS_write(1, " Contrase\303\261a correcta \n", 23 Contraseña
correcta
) = 23
[0x40122c] <... puts resumed> )                   = 24
[0x40123b] gettimeofday(0x7ffe740e6f90, 0)        = 0
[0x40125c] strlen("movimientohelicoidal\n")       = 21
[0x401279] printf("\nIntroduce el pin: " <unfinished ...>
[0x7fb0c61b71e7] SYS_write(1, "\n", 1
)           = 1
[0x401279] <... printf resumed> )                 = 19
[0x40128f] __isoc99_scanf(0x402126, 0x7ffe740e6f7c, 0, 0 <unfinished ...>
[0x7fb0c61b71e7] SYS_write(1, "Introduce el pin: ", 18Introduce el pin: ) = 18
[0x7fb0c61b7142] SYS_read(01111
, "1111\n", 1024)      = 5
[0x40128f] <... __isoc99_scanf resumed> )         = 1
[0x4011a6] puts("\n*****************\n*** KABOOM!!!"... <unfinished ...>
[0x7fb0c61b71e7] SYS_write(1, "\n*****************\n*** KABOOM!!!"..., 55
*****************
*** KABOOM!!! ***
*****************
) = 55
                                                    ...
```

Esta bomba es muy curiosa, ya que, como vemos, nos dice que la contraseña es correcta sin que siquiera compare las cadenas con **strcmp**, mientras que sería muy sospechoso que hayamos acertado a la primera de casualidad. Sin embargo, podemos comprobar que el programa hace una llamada a la función **strlen**, lo cual podría implicar que primero tenga que comprobar la

longitud y luego, si coinciden, tal vez compararlas en sí. Por ello, probemos añadiendo el mismo número de carácteres a ver que sucede.

```
                                                ...
[0x7fdfa3c79142] SYS_read(0aaaaaaaaaaaaaaaaaaa
, "aaaaaaaaaaaaaaaaaaa\n", 1024) = 21
[0x40121b] <... fgets resumed> "aaaaaaaaaaaaaaaaaaa\n", 100, 0x7fdfa3d53980) =
0x7ffd2a54f840
[0x40122c] puts("\n Contrase\303\261a correcta " <unfinished ...>
[0x7fdfa3c791e7] SYS_write(1, "\n", 1
)           = 1
[0x7fdfa3c791e7] SYS_write(1, " Contrase\303\261a correcta \n", 23 Contraseña
correcta
) = 23
[0x40122c] <... puts resumed> )                   = 24
[0x40123b] gettimeofday(0x7ffd2a54f830, 0)        = 0
[0x40125c] strlen("movimientohelicoidal\n")       = 21
[0x401279] printf("\nIntroduce el pin: " <unfinished ...>
[0x7fdfa3c791e7] SYS_write(1, "\n", 1
)           = 1
[0x401279] <... printf resumed> )                 = 19
[0x40128f] __isoc99_scanf(0x402126, 0x7ffd2a54f81c, 0, 0 <unfinished ...>
[0x7fdfa3c791e7] SYS_write(1, "Introduce el pin: ", 18Introduce el pin: ) = 18
[0x7fdfa3c79142] SYS_read(01111
, "1111\n", 1024)      = 5
[0x40128f] <... __isoc99_scanf resumed> )         = 1
[0x4011a6] puts("\n*****************\n*** KABOOM!!!"... <unfinished ...>
[0x7fdfa3c791e7] SYS_write(1, "\n*****************\n*** KABOOM!!!"..., 55
*****************
*** KABOOM!!! ***
*****************
) = 55
                                                ...
```

Como vemos, nos aporta la misma información, por lo que he llegado a la conclusión de que tal vez compare las cadenas tras comprobar el pin. Por ello, vamos a **gdb** y comprobemos que sucede. Antes de nada, destacar que hay una cadena, "**movimientohelicoidal**", que posiblemente nos sea de utilidad:

```
0x4012a1 <main+215>     callq  0x401090 <__isoc99_scanf@plt>

0x4012a6 <main+220>     cmp    $0x1,%ebx

0x4012a9 <main+223>     jne    0x401268 <main+158>

0x4012ab <main+225>     mov    0x2deb(%rip),%eax        # 0x40409c <passsword>

0x4012b1 <main+231>     cmp    %eax,0xc(%rsp)
```

Depurando, nos damos cuenta de ciertas cosas. Cuando introducimos la contraseña, ésta es comparada tras introducir después de pedir el código, lo cual confirma una de nuestras sospechas. Lo primero es que se comprueba primero el pin. Al imprimir eax nos damos cuenta que $eax = 1001. Además, vemos que el pin no sufre transformaciones, por tanto **password = 1001 es el código**. Ahora volvamos a **ltrace** para facilitar la búsqueda de la cadena clave. Igualmente, si nos fijamos en gdb veremos que:

```
0x4012c1 <main+247>     lea    0x2da8(%rip),%rsi        # 0x404070 <nomemires>
0x4012c8 <main+254>     callq  0x401030 <strncmp@plt>
```

Donde **<nomemires>** vemos que es **"movimientohelicoidal\n"**, obtenido mediante `x/s 0x404070`. De momento todas las pistas apuntan a que la contraseña es esa, así que comprobémoslo:

```
[0x40128f] <... __isoc99_scanf resumed> )                              = 1
[0x4012cd] strncmp("aaaaaaaaaa\n", "movimientohelicoidal\n", 22)       = -12
```

Efectivamente, comprobamos que **strcmp** compara con movimientohelicoidal, mientras que no modifica la entrada. Por tanto, ya tenemos la solución:

```
pablo@laptop:~/Bombas$ ./bomba_ARM_2020

Introduce la contraseña: movimientohelicoidal

 Contraseña correcta

Introduce el pin: 1001

**************************************************
*** Esta vez no exploto, pero habra una proxima ***
**************************************************
```