Práctica 4 segunda parte

Por: Arturo Cortés Sánchez

Bomba de Alberto Cano Jaen

Contraseña=segmentation
Pin=1124

Contraseña modificada=Segmentation
Pin modificado=2124

Observando el código ensamblador de esta bomba con gdb podemos ver que no hay ninguna función de cifrado, por lo que la modificación que le hace a la contraseña y al pin debe estar en el main. También vemos que ha mantenido los nombres <passcode> y <password>. Si los consultamos podemos ver sus valores.

(gdb) x/1bs 0x404068
0x404068 <password>: "oegmentatisn\n"

(gdb) x/1wd 0x404060
0x404060 <passcode>: 1111

Si en la ejecucion del programa introducimos el string abcdefghijk como contraseña y consultamos el valor justo antes del strncmp, obtenemos esto
(gdb) x/bs $r8
0x7fffffffde00: "jbcdefghiak\n"

Ha cambiado la primera letra por la penúltima

Repetimos con una contraseña mas corta: abcde

(gdb) x/1bs $r8
0x7fffffffde00: "dbcae\n"

Vemos que sigue cambiando la primera letra por la penúltima. Así que deshacemos el proceso sobre <password> y nos da segmentation

Para descubrir el pin avanzamos hasta donde se hace el scanf y vemos el siguiente grupo de instrucciones

```
0x4012d9 <main+291>     mov    0x2d81(%rip),%eax        # 0x404060 <passcode>
0x4012df <main+297>     add    $0xd,%eax
0x4012e2 <main+300>     mov    %eax,0x2d78(%rip)        # 0x404060 <passcode>
0x4012e8 <main+306>     cmp    0xc(%rsp),%eax
```

Vemos que guarda el passcode en eax, le suma 0xd y lo devuelve a su posición original. Así que el pin debe ser el valor de <passcode> mas 13, lo que nos da 1124

Para modificar la contraseña he usado ghex. He buscado oegmentatisn y he cambiado la s por una S.
Para editar el pin he abierto el ejecutable con el gdb en modo escritura y he escrito el siguiente comando: set{int}0x404060=2111

Bomba de Ángeles Caballero Floro

Contraseña= Contraseña:
Pin=8038

Contraseña modificada= Controseña:
pin modificado = 8888

Observando el código ensamblador de esta bomba podemos ver que tiene una función llamada encriptar, pero si extraemos los valores de <password> y <passcode> vemos que contienen la contraseña y el pin sin ningún tipo de cifrado. Si seguimos la ejecución de esta bomba podemos ver que lo que hace es pasar por la función de cifrado tanto la contraseña que introduzcamos como <password> y los compara. No aplica ningún cifrado al pin.

(gdb) x/1bs 0x601068
0x601068 <password>:   "Contraseña: \n"

(gdb) x/1wd 0x601060
0x601060 <passcode>:   8038

Para modificar la contraseña he usado ghex. He buscado Contraseña: y he cambiado la primera “a” por una “o”.
Para editar el pin he abierto el ejecutable con el gdb en modo escritura y he escrito el siguiente comando: set{int}0x601060=8888

Bomba de Juan Manuel Consigliere Picco

Contraseña=esdificil
pin=2101

Contraseña modificada = Esdificil
pin modificado = 1000

Observando el código ensamblador de esta bomba podemos ver que tiene una función llamada aCesar, por lo que podemos suponer que estamos ante un cifrado cesar. Primero vemos los valores de <password> y <passcode>

x/1bs 0x601068
0x601068 <password>:   "kyjoloior\n"

x/1wd 0x601060
0x601060 <passcode>:   4202

Si ejecutamos la bomba en gdb e introducimos abcdef de contraseña, vemos que tras la ejecución de aCesar se ha convertido en ghijkl. Se le ha aplicado un cifrado cesar el cual ha sumado 6 a cada carácter.

(gdb) x/1bs $r8
0x7fffffffdde0: "ghijkl\n"

Por lo tanto para descifrar la contraseña restamos 6 a cada carácter de <password> y nos queda: esdificil.

Para descifrar el pin nos vamos inmediatamente después de scanf. Vemos que hay una instrucción add %eax,%eax y poco después hace la comparación. Por lo tanto para obtener el pin debemos dividir 4202 entre 2 y nos queda 2101

Para modificar la contraseña he usado ghex. He buscado kyjoloior y he cambiado la “k” por una “K” Para editar el pin he abierto el ejectutable con el gdb en modo escritura y he escrito el siguiente comando: set{int}0x601060=2000.

Bomba de Daniel Ballesteros Fernandes

Contraseña=onepiece
pin=2525

Contraseña modificada= Onepiece
pin modificado =900

Observando el código ensamblador de esta bomba podemos ver que tiene dos funciones, una llamada modificaPin y otra llamada modificaPassword. Primero vemos el valor de <passcode>

(gdb) x/1wd 0x601068
0x601068 <passcode>:   4625

Esta bomba tiene el password un poco escondido. Suponemos que en algún momento tiene que hacer uso de la contraseña encriptada, y el lugar mas probable es cerca del strncmp. Dos instrucciones antes de strncmp encontramos la siguiente instrucción: mov $0x601070,%esi . Si comprobamos el contenido de dicha dirección de memoria nos encontramos con la contraseña cifrada

(gdb) x/1bs 0x601070
0x601070 <password>:   "oogsmjil\n"

Para averiguar la contraseña avanzamos en la ejecución hasta que la bomba nos pida una contraseña. Introducimos “aaaaaaaa” y avanzamos hasta después de la llamada a la función modificaPassword. Comprobamos el resultado tras la función de cifrado.

(gdb) x/1bs $r8
0x7fffffffde00: "abcdefgh\n"

Vemos que a cada carácter le ha sumado su posición en el string. Por lo tanto si a cada letra de oogsmjil le restamos su posición en el string nos quedará la contraseña original: onepiece

Para averiguar el pin avanzamos hasta scanf, introducimos un numero cualquiera, por ejemplo 1000 y avanzamos hasta despues de modificaPin. Vemos que %rax contiene 3100

rax 0xc1c 3100

Volvemos a ejecutar la bomba y probamos con otro numero, en este caso 2000, al pasar de modificaPin vemos que %rax contiene 4100

rax 0x1004 4100

Así podemos asumir que el cifrado del pin es sumarle 2100. Entonces le restamos 2100 al passcode y nos queda 2525.

Para modificar la contraseña he abierto el ejecutable con ghex, he buscado oogsmjil y he cambiado la primera o por una O.

Para editar el pin he abierto el ejecutable con el gdb en modo escritura y he ejecutado el siguiente comando set{int}0x601068 =3000

Bomba de Daniel Ruiz Medina

Contraseña=botella
pin=100

Contraseña modificada = Botella
pin modificado = 10000

Observando el código ensamblador de esta bomba podemos ver que tiene dos funciones, una llamada p1 y otra llamada p2_ft. De primeras no están <passcode> ni <password>, pero en p1 hay una variable llamada <pa> y en p2 una llamada <p3>

(gdb) x/1bs 0x601058
0x601058 <pa>: "ly~ovvk\n"

(gdb) x/1wd 0x601064
0x601064 <p3>: 500

Ya tenemos la contraseña y el pin cifrado, pero esta vez la técnica de introducir una contraseña y ver en que ha resultado no funciona, así que pasamos a analizar el código ensamblador de <p1>

```
0x40071b <p1>           mov    $0x0,%edx
0x400720 <p1+5>         mov    $0x0,%eax
0x400725 <p1+10>        cmp    $0x6,%edx
0x400728 <p1+13>        setbe  %sil
0x40072c <p1+17>        test   %eax,%eax
0x40072e <p1+19>        sete   %cl
0x400731 <p1+22>        test   %cl,%sil
0x400734 <p1+25>        jne    0x400738 <p1+29>
0x400736 <p1+27>        repz retq
0x400738 <p1+29>        movslq %edx,%rcx
0x40073b <p1+32>        movsbl (%rdi,%rcx,1),%eax
0x40073f <p1+36>        add    $0xa,%eax
0x400742 <p1+39>        lea    0x20090f(%rip),%rsi        # 0x601058 <pa>
0x400749 <p1+46>        movsbl (%rsi,%rcx,1),%ecx
0x40074d <p1+50>        sub    %ecx,%eax
0x40074f <p1+52>        add    $0x1,%edx
0x400752 <p1+55>        jmp    0x400725 <p1+10>
```

Vemos que parece un bucle en el que %edx es la “i”. Comienza realizando dos comprobaciones: la del final del bucle y que %eax sea 0. Luego hace un salto y guarda el carácter “i” de la contraseña introducida en %eax y le suma 10, después extrae el carácter “i” de <pa> y lo guarda en %ecx. Acto seguido le resta %ecx a %eax y para acabar la función, le suma 1 a %edx.

Al haber analizado la funcion vemos que necesitamos que %eax valga 0, y eso se consigue haciendo que la “sub %ecx,%eax” de 0 de resultado. Para ello necesitamos introducir una contraseña que sea cada carácter de <pa> -10, es decir botella.

Para averiguar el pin analizamos <p2_ft>

```
0x400754 <p2_ft>        lea    (%rdi,%rdi,4),%eax
0x400757 <p2_ft+3>      cmp    %eax,0x200907(%rip)        # 0x601064 <p3>
0x40075d <p2_ft+9>      jne    0x400765 <p2_ft+17>
0x40075f <p2_ft+11>     mov    $0x0,%eax
0x400764 <p2_ft+16>     retq
0x400765 <p2_ft+17>     mov    $0x1,%eax
0x40076a <p2_ft+22>     retq
```

Vemos que en “lea (%rdi,%rdi,4),%eax” multiplica el pin introducido por 5 y lo compara con <p3>. Por lo tanto el pin es <p3> entre 5, es decir 100.

Para modificar la contraseña he abierto el ejecutable con ghex, he buscado ly~ovvk y he cambiado la l por una L.

Para editar el pin he abierto el ejecutable con el gdb en modo escritura y he ejecutado el siguiente comando set{int}0x601064=5000

Bomba de Juan Ignacio Villegas Llano

contraseña= melocoton
pin = 5000

Contraseña modificada = Melocoton
pin modificado = 1024

Observando el código ensamblador de esta bomba podemos ver que tiene dos funciones, una llamada modificar_passwd y otra llamada modifica_passcd. Primero buscamos la contraseña y el pin cifrados

(gdb) x/1bs 0x601068
0x601068 <passwordcambiada>: "nfmpdpupo\n"

(gdb) x/1wd 0x601060
0x601060 <passcodecambiado>: 5000

A continuación analizamos modificar_passwd.

```
0x400727 <modificar_passwd>     mov    %rdi,%r8
0x40072a <modificar_passwd+3>   mov    $0x0,%esi
0x40072f <modificar_passwd+8>   movslq %esi,%rdx
0x400732 <modificar_passwd+11>  mov    $0xffffffffffffffff,%rcx
0x400739 <modificar_passwd+18>  mov    $0x0,%eax
0x40073e <modificar_passwd+23>  mov    %r8,%rdi
0x400741 <modificar_passwd+26>  repnz scas %es:(%rdi),%al
0x400743 <modificar_passwd+28>  mov    %rcx,%rax
0x400746 <modificar_passwd+31>  not    %rax
0x400749 <modificar_passwd+34>  sub    $0x2,%rax
0x40074d <modificar_passwd+38>  cmp    %rax,%rdx
0x400750 <modificar_passwd+41>  jb     0x400754 <modificar_passwd+45>
0x400752 <modificar_passwd+43>  repz retq
0x400754 <modificar_passwd+45>  add    %r8,%rdx
0x400757 <modificar_passwd+48>  movzbl (%rdx),%eax
0x40075a <modificar_passwd+51>  add    $0x1,%eax
0x40075d <modificar_passwd+54>  mov    %al,(%rdx)
0x40075f <modificar_passwd+56>  add    $0x1,%esi
0x400762 <modificar_passwd+59>  jmp    0x40072f <modificar_passwd+8>
```

Podemos apreciar un bucle que va desde 0 a %rax donde %rax es el tamaño del string pasado como parámetro menos 2 y donde %esi es la i.

En estas 3 lineas podemos ver lo que hace el bucle

```
0x400757 <modificar_passwd+48>  movzbl (%rdx),%eax
0x40075a <modificar_passwd+51>  add    $0x1,%eax
0x40075d <modificar_passwd+54>  mov    %al,(%rdx)
```

Suma 1 a cada carácter del string, por lo tanto si restamos 1 a cada carácter de nfmpdpupo nos queda melocotón

En el caso del pin vemos que hay una función llamada modifica_passcd, pero esta nunca es llamada así que el pin está sin cifrar y es 5000

Para modificar la contraseña he abierto el ejecutable con ghex, he buscado nfmpdpupo y he cambiado la n por una N.

Para editar el pin he abierto el ejecutable con el gdb en modo escritura y he ejecutado el siguiente comando set{int}0x601060=1024

Bomba de Elvira Castillo Fernandez

Contraseña=SwadMola
pin=2018

Contraseña modificada=swadMola
pin modificado=1518

Observando el código ensamblador de esta bomba podemos ver que tiene dos funciones, una llamada explota y otra llamada explota_ . Primero buscamos la contraseña y el pin cifrados

(gdb) x/1wd 0x404060
0x404060 <passcode>:   4035

(gdb) x/1bs 0x404068
0x404068 <password>:   "Sv_aIjfZ\n"

A continuación analizamos la función explota

```
0x4011e6 <explota>      mov    %rdi,%rsi
0x4011e9 <explota+3>    mov    $0xffffffffffffffff,%rcx
0x4011f0 <explota+10>   mov    $0x0,%eax
0x4011f5 <explota+15>   repnz scas %es:(%rdi),%al
0x4011f7 <explota+17>   not    %rcx
0x4011fa <explota+20>   sub    $0x2,%ecx
0x4011fd <explota+23>   mov    $0x0,%eax
0x401202 <explota+28>   cmp    %ecx,%eax
0x401204 <explota+30>   jge    0x401211 <explota+43>
0x401206 <explota+32>   movslq %eax,%rdx
0x401209 <explota+35>   sub    %al,(%rsi,%rdx,1)
0x40120c <explota+38>   add    $0x1,%eax
0x40120f <explota+41>   jmp    0x401202 <explota+28>
0x401211 <explota+43>   retq
```

Vemos que las primeras lineas son un strlen. Después hay un bucle que va desde 0 hasta el valor obtenido del strlen. La “i” del bucle es %eax.

En esta instrucción podemos ver como al carácter “i” del string introducido le resta “i”

```
0x401209 <explota+35>   sub    %al,(%rsi,%rdx,1)
```

Por lo que podemos asumir que el algoritmo de cifrado consiste en restarle a cada carácter su posición en el string. Sabiendo esto para obtener la contraseña solo tenemos que sumar a cada carácter de “Sv_aIjfZ” su posición en el string, lo que nos da como resultado SwadMola.

Para obtener el pin analizamos la funcion explota_

```
0x401212 <explota_>     lea    -0x1(%rdi,%rdi,1),%eax
0x401216 <explota_+4>   retq
```

Vemos que multiplica el pin introducido por dos y le resta 1. Por lo tanto tenemos que sumarle 1 a 4035 y dividirlo entre 2, lo que nos da 2018.

Para modificar la contraseña he abierto el ejecutable con ghex, he buscado Sv_aIjfZ y he cambiado la S por una s.

Para editar el pin he abierto el ejecutable con el gdb en modo escritura y he ejecutado el siguiente comando set{int}0x404060=3035

Bomba de Fernando Roldan Zafra

Contraseña=HOLIHOLI
pin=2015

Contraseña modificada=IOLIHOLI
pin modificado = 2018

Observando el código ensamblador de esta bomba podemos ver que tiene dos funciones, una llamada encriptacion y otra llamada encriptacion2 . Primero buscamos la contraseña y el pin cifrados

(gdb) x/1bs 0x601068
0x601068 <password>:   "cjgdcjgd%"

(gdb) x/1wd 0x601060
0x601060 <passcode>:   4030

Para averiguar la contraseña sin cifrar ejecutamos la bomba en gdb y avanzamos hasta que nos pida la contraseña, introducimos “aaaaa” y continuamos hasta después de la llamada a la función encriptacion

Comprobamos el resultado:

(gdb) x/1bs $r8
0x7fffffffddf0: "|||||%"

Vemos que a cada carácter le ha sumado 27. Por lo que si a cada carácter de <password> le restamos 27 nos queda HOLIHOLI,

Para averiguar el pin analizamos encriptacion2

```
0x40077e <encriptacion2>        lea    (%rdi,%rdi,1),%eax
```

```
│0x400781 <encriptacion2+3>      retq
```

Vemos que unicamente multiplica por 2 el valor introducido, asi que dividimos entre 2 <passcode> y nos queda 2015.

Para modificar la contraseña he abierto el ejecutable con ghex, he buscado cjgdcjgd% y he cambiado la primera c por una d.

Para editar el pin he abierto el ejecutable con el gdb en modo escritura y he ejecutado el siguiente comando set{int}0x601060 =4036

Bomba de Francisco Jimenez Rodriguez

Contraseña=ecmola!
Pin=666

Contraseña modificada =ECmola!
Pin modificado=466

Observando el código ensamblador de esta bomba podemos ver que tiene dos funciones, una llamada printfp y otra llamada scanfp. Si buscamos el pin y la contraseña cifrados vemos que no hay ningún <password> o <passcode>, pero dentro de las funciones hay dos variables llamadas <error> y <notfound>. Consultamos sus valores:

(gdb) x/1bs 0x601068
0x601068 <error>: "!alomce\n"

(gdb) x/1wd 0x601060
0x601060 <notfound>: 346

En el caso de <error> podemos ver a simple vista que es “ecmola!” del revés.

Para averiguar el pin analizamos la función scanfp:

```
│0x40087e <scanfp>       mov    0x2007dc(%rip),%eax        # 0x601060 <notfound>
│0x400884 <scanfp+6>     lea    -0x1a(%rax,%rax,1),%eax
│0x400888 <scanfp+10>    cmp    %edi,%eax
│0x40088a <scanfp+12>    je     0x400895 <scanfp+23>
│0x40088c <scanfp+14>    sub    $0x8,%rsp
│0x400890 <scanfp+18>    callq  0x400747 <boom>
│0x400895 <scanfp+23>    repz retq
```

En esta instrucción vemos como coge 346, lo multiplica por 2 y le resta 26, lo que nos da de resultado 666

```
│0x400884 <scanfp+6>     lea    -0x1a(%rax,%rax,1),%eax
```

Para modificar la contraseña he abierto el ejecutable con ghex, he buscado !alomce% y he cambiado la primera c y la e por una C y una E.

Para editar el pin he abierto el ejecutable con el gdb en modo escritura y he ejecutado el siguiente comando set{int}0x601060 =246

Bomba de Guillermo Lupiañez Tapia

Contraseña=picapiedra
Pin=112233

Contraseña modificada=Picapiedra
Pin modificado= 2233

Observando el código ensamblador de esta bomba podemos ver que tiene dos funciones, una llamada codificarPw y otra llamada codificarPc . Primero buscamos la contraseña y el pin cifrados

(gdb) x/1s 0x601068
0x601068 <password_codificada>: "rkecrkgftc\n"

(gdb) x/1wd 0x601060
0x601060 <passcode_codificado>: 112235

Para averiguar la contraseña sin cifrar ejecutamos la bomba en gdb y avanzamos hasta que nos pida la contraseña, introducimos “aaaa” y continuamos hasta después de la llamada a la función codificarPw

Comprobamos el resultado:

(gdb) x/1s $rdi
0x7fffffffdde0: "cccc\n"

Por lo tanto asumimos que es un cifrado cesar que le suma 2 a cada carácter. Sabiendo esto la contraseña seria “picapedra”

Para averiguar el pin analizamos la función codificarPc:

```
0x400779 <codificarPc>          addl   $0x2,(%rdi)
0x40077c <codificarPc+3>        retq
```

Vemos que simplemente le suma 2 a lo que introducimos, por lo tanto debemos restarle 2 a <passcode_codificado> y nos queda 112233

Para modificar la contraseña he abierto el ejecutable con ghex, he buscado rkecrkgftc y he cambiado la primera r por una R.

Para editar el pin he abierto el ejecutable con el gdb en modo escritura y he ejecutado el siguiente comando set{int}0x601060 =2235

Bomba de Jose Antonio Ramírez Diez

Contraseña=holaqtal
Pin=1115
Contraseña modificada= Holaqtal
Pin modificado=2155

Observando el código ensamblador de esta bomba podemos ver que tiene una función llamada desencripta. Primero buscamos la contraseña y el pin cifrados

(gdb) x/s 0x601070
0x601070 <password>: "gnkapsak\n

(gdb) x/1wd 0x601068
0x601068 <passcode>: 1115

Para averiguar la contraseña sin cifrar ejecutamos la bomba en gdb y avanzamos hasta que nos pida la contraseña, introducimos “abcdefgh” y continuamos hasta después de la llamada a la función densencripta.

Consultamos el valor del resultado

(gdb) x/1s $r8
0x7fffffffdde0: "aabcdefg\n"

Obtenemos el mismo string pero con una a al principio y una letra menos al final

Probamos con otra contraseña, esta vez “bbbbb”

(gdb) x/1s $r8
0x7fffffffdde0: "ccccc\n"

Tras pensar un poco vemos que a cada letra le suma 1 menos a la “a” que la deja intacta. Por lo tanto si a cada letra de “gnkapsak” excepto a la “a” le restamos 1 nos queda “holaqtal”

En el caso del pin vemos que <passcode> se usa debajo en la comparación directamente, por lo que no hay cifrado de pin.

Para modificar la contraseña he abierto el ejecutable con ghex, he buscado gnkapsak y he cambiado la primera g por una G.

Para editar el pin he abierto el ejecutable con el gdb en modo escritura y he ejecutado el siguiente comando set{int}0x601068 =2115

Bomba de Jose Antonio Martín Alarcón

Contraseña=jose
pin=1996

Contraseña modificada=kose
pin modificado=1740

Observando el código ensamblador de esta bomba podemos ver que tiene una función llamada cod. Primero buscamos la contraseña y el pin cifrados

(gdb) x/s 0x601064
0x601064 <password>: "kqvi\n"

No hay ninguna variable que contenga el pin, pero algo debajo del scanf nos encontramos la siguiente instrucción

```
0x400879 <main+254>     cmp    $0x7c6,%eax
```

Si pasamos 0x7c6 a decimal nos da 1990

Para averiguar la contraseña sin cifrar ejecutamos la bomba en gdb y avanzamos hasta que nos pida la contraseña, introducimos “aaaaa” y continuamos hasta después de la llamada a la función cod.

Consultamos el string resultante:

(gdb) x/s $rdi
0x7fffffffdde0: "bcdef\n"

Vemos que a cada letra le ha sumado su posición + 1, así que si a cada letra de kqvi le restamos posición en el string +1 nos dará la contraseña, es decir jose.

Para el pin ya que no hay función dedicada, volvemos a mirar debajo del scanf. Vemos que a lo que hemos introducido le resta 6 y lo compara con 1990, así que el pin es 1996.

```
0x40086e <main+243>     mov    0xc(%rsp),%eax
0x400872 <main+247>     sub    $0x6,%eax
0x400875 <main+250>     mov    %eax,0xc(%rsp)
0x400879 <main+254>     cmp    $0x7c6,%eax
```

Para modificar la contraseña he abierto el ejecutable con ghex, he buscado kqvi y he cambiado la l por una k.

Para modificar el pin he usado el ghex, he buscado 7c6 en little endian, es decir c607 y reemplazo el 7 con un 6. Con este cambio el pin pasa a ser 1740.

Bomba de Jose Maria Gomez

Contraseña=Tamagotchi_01
Pin=1996

Contraseña modificada=tamagotchi_01
Pin modificado=1994

Observando el código ensamblador de esta bomba podemos ver que tiene dos funciones, una llamada codifypassword y otra llamada codifypasscode . Primero buscamos la contraseña y el pin cifrados

```
(gdb) x/s 0x601070
0x601070 <password>:    "Tbodktzjpri;=\n"

(gdb) x/1wd 0x601068
0x601068 <passcode>:    3994
```

Para averiguar la contraseña sin cifrar ejecutamos la bomba en gdb y avanzamos hasta que nos pida la contraseña, introducimos “aaaaa” y continuamos hasta después de la llamada a la función codifypassword.

(gdb) x/s $rdi
0x7fffffffddf0: "abcde\n"

Vemos que a cada carácter del string le ha sumado su posición. Por lo que si a Tbodktzjpri;= le restamos su posición nos queda Tamagotchi_01

Para averiguar el pin analizamos el código de codifypasscode

```
0x40078b <codifypasscode>       mov    (%rdi),%eax
0x40078d <codifypasscode+2>     lea    0x2(%rax,%rax,1),%eax
0x400791 <codifypasscode+6>     mov    %eax,(%rdi)
0x400793 <codifypasscode+8>     retq
```

Vemos que multiplica por 2 y suma 2 al pin introducido. Sabiendo esto tenemos que restar 2 al <passcode> y dividirlo entre 2. Nos queda 1996.

Para modificar la contraseña he abierto el ejecutable con ghex, he buscado Tbodktzjpri;= y he cambiado la T por una t.

Para editar el pin he abierto el ejecutable con el gdb en modo escritura y he ejecutado el siguiente comando set{int}`0x601068` =3990.

Bomba de Jose Saldaña Mercado

Contraseña=estaesmibomba
pin=97531

Contraseña modificada= Estaesmibomba
pin modificado = 98531

Observando el código ensamblador de esta bomba podemos ver que tiene dos funciones, una llamada encript y otra llamada encode . Primero buscamos la contraseña y el pin cifrados

(gdb) x/s 0x601070
0x601070 <password>:   "abmobimseatse\n"

(gdb) x/wd 0x601068
0x601068 <passcode>: 13579

En el caso de <password> vemos a simple vista que es “estaesmibomba” del revés

Para averiguar el pin analizamos la función encode

```
0x400813 <encode>       mov    $0x1b206,%eax
0x400818 <encode+5>     sub    %edi,%eax
0x40081a <encode+7>     retq
```

Vemos que al pin introducido le resta 0x1b206 (111110) y en el main compara el resultado de la resta con el passcode. Por lo tanto debemos de restarle a 111110 el <passcode> y nos da como resultado el pin: 97531

Para modificar la contraseña he abierto el ejecutable con ghex, he buscado abmobimseatse y he cambiado la última e por una E.

Para editar el pin he abierto el ejecutable con el gdb en modo escritura y he ejecutado el siguiente comando `set{int}0x601068=12579`

Bomba de Juan Carlos Pineda Muñoz

Contraseña=GRaNaDa97
Pin=3230

```
Contraseña modificada=GRaNaDa96
Pin modificado= 1000
```

Observando el código ensamblador de esta bomba podemos ver que tiene tres funciones, una llamada encriptarInicial, otra llamada encriptarcode y otra llamada encriptarLogin. Comenzamos como siempre buscando la contraseña y el pin cifrados, pero solo vemos una variable <password> en la función encriptarcode:

(gdb) x/s 0x601070
0x601070 <password>: "\236\f"

Vemos que no parece ser un string, así que probamos con un numero

(gdb) x/wd 0x601070
0x601070 <password>: 16150

Como la contraseña cifrada no aparece paso a mirar en el volcado objdump completo (objdump -D) y cerca del final encontramos esto:

0000000000601078 <login>:
 601078: 47 52   rex.RXB push %r10
 60107a: 61   (bad)
 60107b: 4e 61   rex.WRX (bad)

```
60107d:       44 61                   rex.R (bad)
60107f:       39 37                   cmp    %esi,(%rdi)
```

Así que consultamos esa dirección de memoria en gdb

```
(gdb) x/s  0x601078
0x601078 <login>:       "GRaNaDa97"<error: Cannot access memory at address 0x601081>
```

Parece un string sin cifrar, así que introducimos como contraseña y en efecto funciona. Si <login> es la contraseña podemos asumir que <password> contiene el pin cifrado, así que analizamos la función en la que se encuentra <password>:

```
0x4008c7 <encriptarcode>        lea    (%rdi,%rdi,4),%eax
0x4008ca <encriptarcode+3>      cmp    0x2007a0(%rip),%eax        # 0x601070 <password>
0x4008d0 <encriptarcode+9>      je     0x4008e0 <encriptarcode+25>
0x4008d2 <encriptarcode+11>     sub    $0x8,%rsp
0x4008d6 <encriptarcode+15>     mov    $0x0,%eax
0x4008db <encriptarcode+20>     callq  0x4007c6 <boom>
0x4008e0 <encriptarcode+25>     repz retq
```

En la primera linea vemos que simplemente multiplica por 5 el pin introducido, así que si dividimos entre 5 el pin cifrado obtendremos el pin correcto: 3230

Para modificar la contraseña he abierto el ejecutable con ghex, he buscado GRaNaDa97 y he cambiado el 7 por un 6.

Para editar el pin he abierto el ejecutable con el gdb en modo escritura y he ejecutado el siguiente comando `set{int}` 0x601070=5000

Bomba de Luis Escobar Reche

Contraseña=dabironico
Pin= 3263

Contraseña modificada=Dabironico
pin modificado=2152

Observando el código ensamblador de esta bomba podemos ver que tiene dos funciones, una llamada cifrado_c y otra llamada cifrado_p . Primero buscamos la contraseña y el pin cifrados

```
x/wd 0x601068
0x601068 <passcode>:    6666

(gdb) x/s 0x601070
0x601070 <password>:    "efcnstondt\n"
```

A continuación ejecutamos el programa en gdb, introducimos aaaaaaaaaa como contraseña y avanzamos hasta después de la llamada a cifrado_c. Comprobamos el string resultante:

```
(gdb) x/s $rdi
0x7fffffffddf0: "bfbfbfbfbfb\n"
```

Vemos que a las letras impares les ha sumado 1 y a las pares 5. Así que si realizamos el proceso inverso sobre “efcnstondt” nos da dabironico

Para averiguar el pin comprobamos la función cifrado_p

```
0x400930 <cifrado_p>            lea    0x8c(%rdi,%rdi,1),%eax
0x400937 <cifrado_p+7>          retq
```

Vemos que multiplica por dos el pin introducido y le suma 0x8c (140) así que si al <passcode> le restamos 140 y lo dividimos entre 2 nos queda el pin: 3263

Para modificar la contraseña he abierto el ejecutable con ghex, he buscado `efcnstondt` y he cambiado la e por una E.

Para editar el pin he abierto el ejecutable con el gdb en modo escritura y he ejecutado el siguiente comando `set{int} 0x601068=4444`

Bomba de Manuel Carmona Pérez

Contraseña=HoLaMuNdO
Pin= 2018
Contraseña modificada =MoLaMuNdO
Pin modificado = 1018

Observando el código ensamblador de esta bomba podemos ver que no tiene ninguna función, así que el proceso de cifrado debe realizarse en el main. Cambien vemos que no aparece ninguna variable, así que miramos su volcado completo con objdump (objdump -D) y sobre el final vemos estas dos variables:

```
0000000000601068 <passcode>:
  601068:       da 07                   fiaddl (%rdi)
  60106a:       00 00                   add    %al,(%rax)
  60106c:       00 00                   add    %al,(%rax)
        ...

0000000000601070 <password>:
  601070:       68 6f 6c 61 6d          pushq  $0x6d616c6f
  601075:       75 6e                   jne    6010e5 <_end+0x55>
  601077:       64 6f                   outsl  %fs:(%rsi),(%dx)
  601079:       0a 00                   or     (%rax),%al
```

Así que comprobamos dichas direcciones en gdb:

```
(gdb) x/s 0x601070
0x601070 <password>:    "holamundo\n"

(gdb) x/wd 0x601068
0x601068 <passcode>:    2010
```

A continuación ejecutamos el programa en gdb, introducimos AAAAA como contraseña y avanzamos hasta antes de la llamada a strncmp. Comprobamos el string resultante:

(gdb) x/s $rdi
0x7fffffffddf1: "AaAaA*"

Vemos que ponen en minúscula las letras pares, así que si ponemos en mayúscula las letras impares <password> obtenemos la contraseña: HoLaMuNdO

Para obtener el pin analizamos el código entre el scanf y la comparación con <passcode>

```
0x4008f7 <main+353>     callq  0x400640 <__isoc99_scanf@plt>
0x4008fc <main+358>     cmp    $0x1,%ebx
0x4008ff <main+361>     jne    0x4008bf <main+297>
0x400901 <main+363>     mov    0xc(%rsp),%eax
0x400905 <main+367>     sub    $0x8,%eax
0x400908 <main+370>     mov    %eax,0xc(%rsp)
0x40090c <main+374>     cmp    0x200756(%rip),%eax        # 0x601068 <passcode>
```

Vemos que simplemente se le resta 8 al valor introducido. Así que si sumamos 8 al <passcode> obtenemos el pin: 2018

Al introducir el pin vemos que explota pase lo que pase. Esto es debido a que esta bomba pide el pin dos veces, la primera con un fgets, y la segunda con scanf. Para que no explote después del fgets hay que pulsar intro sin poner ningún pin, y después, cuando lo pide por segunda vez, introducimos 2018.

Para modificar la contraseña he abierto el ejecutable con ghex, he buscado holamundo y he cambiado la h por una m.

Para editar el pin he abierto el ejecutable con el gdb en modo escritura y he ejecutado el siguiente comando `set{int}` 0x601068`=1010`

Bomba de Manuel Jesús Núñez Ruiz

Contraseña=rootroot
Pin=935

Contraseña modificada=rOotroot
Pin modificado=500

Observando el código ensamblador de esta bomba podemos ver que tiene dos funciones, una llamada cifrador y otra llamada cifrador2 . Primero buscamos la contraseña y el pin cifrados

(gdb) x/s 0x601068
0x601068 <password>:   "|yy~|yy~\n"

(gdb) x/wd 0x601060
0x601060 <passcode>:   1870

A continuación ejecutamos el programa en gdb, introducimos aaaaa como contraseña y avanzamos hasta después de la llamada a cifrador. Comprobamos el string resultante:

(gdb) x/s $rdi
0x7fffffffdde0: "kkkkk\024\n\n", <incomplete sequence \360>

Vemos que parece que ha sumado 10 a cada carácter y ha añadido algo al final
Probamos introduciendo otro string: abcdf

(gdb) x/s $rdi
0x7fffffffdde0: "klmnop\024\n", <incomplete sequence \360>

Vemos que vuelve a sumar 10 a cada carácter y a añadir algo al final, probablemente como resultado de sumar 10 al carácter de fin de linea y al de fin de string.

Para obtener la contraseña sin cifrar restamos 10 a cada carácter de “|yy~|yy~” y nos da “rootroot”

Para averiguar el pin analizamos la función cifrador2

```
0x400781 <cifrador2>    lea    0x0(,%rdi,4),%eax
0x400788 <cifrador2+7>  sar    %eax
0x40078a <cifrador2+9>  retq
```

Vemos que multiplica por 4 el valor introducido y le realiza un desplazamiento 1 bit a la derecha. Para deshacer dicho cifrado he hecho esta función:

```
void descifrar(){
      int a=1870;
      a<<=1;
      a/=4;
      printf("%d\n", a);
}
```

La cual da como resultado 935

Para modificar la contraseña he abierto el ejecutable con ghex, he buscado |yy~|yy~ y he cambiado la primera y por una Y.

Para editar el pin he abierto el ejecutable con el gdb en modo escritura y he ejecutado el siguiente comando `set{int}` 0x601060=1000

Bomba de María Isabel Abellan

Contraseña teórica=listo
Contraseña real= cualquier string que en la quinta posición tenga una o
pin=1997
Contraseña modificada=cualquier string que en la quinta posición tenga una c
pin modificado=1741

Observando el código ensamblador de esta bomba podemos ver que tiene una función llamada comprobar y no vemos ninguna variable con nombre, así que toca mirar con mas detalle. Vemos que no hay ninguna llamada a strncmp, así que que la comparación entre la contraseña introducida y la cifrada se debe realizar de forma manual en algún lado. Mirando el main no vemos nada fuera de lo habitual así que asumimos que se debe realizar en <comprobar>

```
0x400742 <comprobar>    mov    $0x0,%edx
0x400747 <comprobar+5>  jmp    0x40076d <comprobar+43>
0x400749 <comprobar+7>  movslq %edx,%rcx
0x40074c <comprobar+10> movsbl 0x601060(%rcx),%eax
0x400753 <comprobar+17> add    $0x2,%eax
0x400756 <comprobar+20> movsbl (%rdi,%rcx,1),%ecx
0x40075a <comprobar+24> cmp    %ecx,%eax
0x40075c <comprobar+26> je     0x400765 <comprobar+35>
0x40075e <comprobar+28> mov    $0x0,%eax
0x400763 <comprobar+33> jmp    0x40076a <comprobar+40>
0x400765 <comprobar+35> mov    $0x1,%eax
0x40076a <comprobar+40> add    $0x1,%edx
0x40076d <comprobar+43> cmp    $0x4,%edx
0x400770 <comprobar+46> jle    0x400749 <comprobar+7>
0x400772 <comprobar+48> repz retq
```

Vemos que en “`0x40074c <comprobar+10>” se hace uso de una dirección de memoria desconocida asi que la consultamos.`

(gdb) x/1bs 0x601060
0x601060 <password>: "jgqrm"<error: Cannot access memory at address 0x601065>

Nos da la contraseña cifrada y un error que ignoramos, pues ya tenemos lo que inicialmente buscábamos. Podemos intuir un bucle donde %edx es la “i”

```
0x400749 <comprobar+7>  movslq %edx,%rcx
0x40074c <comprobar+10> movsbl 0x601060(%rcx),%eax
0x400753 <comprobar+17> add    $0x2,%eax
0x400756 <comprobar+20> movsbl (%rdi,%rcx,1),%ecx
0x40075a <comprobar+24> cmp    %ecx,%eax
```

En esta parte del bucle podemos ver como en %eax se guarda el carácter “i” de la contraseña cifrada y como se le suma 2 a %eax. A continuación vemos como guarda en %ecx el carácter “i” de la contraseña introducida y como compara %ecx y %eax.

Dependiendo del resultado de la comparación guarda 0 o 1 en %eax. El problema es que dicho fragmento de código está en un bucle, por lo tanto solo se guarda el resultado del la comparación de la ultima iteración. Por esto, aunque el código parece indicar que la contraseña es el resultado de sumar 2 a cada carácter de “jgqrm”, es decir listo, la contraseña real es cualquier string cuya quinta letra sea una “o”

Para averiguar el pin basta con mirar instrucción inmediatamente después de la llamada a scanf:
0x40085c <main+188> cmpl $0x7cd,0xc(%rsp)
Si pasamos 0x7cd a decimal nos da 1997. Al no haber ningún tipo de cifrado, 1997 es el pin.
Para modificar la contraseña he abierto el ejecutable con ghex, he buscado jgqrm y he cambiado la m por una a. Ahora la bomba admite cualquier contraseña que tenga en la quinta posición una c.

Para modificar la contraseña esta vez no vale el truco de las veces anteriores ya que el pin no está almacenado en una dirección. Así que usando el ghex busco 7cd en little endian, es decir cd07 y reemplazo el 7 con un 6. Con este cambio el pin pasa a ser 1741.

Bomba de María Sanz Sánchez

Contraseña=holi
Pin=2912

Contraseña modificada=Holi
Pin modificado=1912

Observando el código ensamblador de esta bomba podemos ver que tiene dos funciones, una llamada encriptar y otra llamada encriptar_pin . Primero buscamos la contraseña y el pin cifrados

```
(gdb) x/wd 0x601068
0x601068 <passcode_encriptado>: 2787
```

Solo encontramos el passcode encriptado, así que buscamos con objudump -D :

```
0000000000601068 <passcode_encriptado>:
  601068:       e3 0a                   jrcxz  601074 <passcode>
        ...

000000000060106c <password_encriptado>:
  60106c:       71 78                   jno    6010e6 <_end+0x56>
  60106e:       75 72                   jne    6010e2 <_end+0x52>
  601070:       00 00                   add    %al,(%rax)
        ...

0000000000601074 <passcode>:
  601074:       60              (bad)
  601075:       0b 00                   or     (%rax),%eax
        ...

0000000000601078 <password>:
  601078:       68 6f 6c 69 00          pushq  $0x696c6f
```

Curiosamente aparecen versiones cifradas y sin cifrar de la contraseña y el pin, así que consultamos sus direcciones de memoria:

```
(gdb) x/s 0x60106c
0x60106c <password_encriptado>: "qxur"

(gdb) x/wd 0x601074
0x601074 <passcode>:    2912
```

(gdb) x/s 0x601078
0x601078 <password>: "holi"

Introducimos la contraseña y el pin sin cifrar y en efecto funcionan.

Para modificar la contraseña he abierto el ejecutable con ghex, he buscado holi y qxur. Les he cambiado su primera letra por su versión mayúscula.

Para editar el pin he abierto el ejecutable con el gdb en modo escritura y he ejecutado los siguientes comandos: `set{int}` 0x601074=1912 `y set{int}` 0x601068=1787.

Bomba de Martín José García Muñoz

Contraseña=laberinto
Pin=1403

Contraseña modificada=Laberinto
Pin modificada=2403

Observando el código ensamblador de esta bomba podemos ver que no tiene ninguna función extra añadida por el programador, así que el cifrado debe realizarse en el main. Primero buscamos la contraseña y el pin cifrados.

(gdb) x/s 0x601068
0x601068 <password>: "mbcfsjoup\n"

(gdb) x/wd 0x601074
0x601074 <passcode>: 1400

A continuación ejecutamos el programa en gdb, introducimos “aaaaaaaaaaaaaaaaaaaa” como contraseña y avanzamos hasta antes de la llamada a strncmp. Comprobamos el string resultante:

(gdb) x/s $rdi
0x7fffffffde01: "bbbbbbbb", 'a' <repeats 12 times>, "\n"

Vemos que a los primeros 8 caracteres se les ha sumado 1, así que si restamos 1 a cada carácter de mbcfsjoup obtenemos la contraseña: laberinto

Para obtener el pin analizamos el código entre el scanf y la comparación con <passcode>

```
0x4008d6 <main+299>     callq  0x400670 <__isoc99_scanf@plt>
0x4008db <main+304>     cmpl   $0x1,-0x94(%rbp)
0x4008e2 <main+311>     jne    0x40088f <main+228>
0x4008e4 <main+313>     mov    -0x9c(%rbp),%eax
0x4008ea <main+319>     lea    -0x3(%rax),%edx
0x4008ed <main+322>     mov    0x200781(%rip),%eax        # 0x601074 <passcode>
```

Vemos que simplemente se le resta 3 al pin introducido, por lo que el pin real es <passcode> + 3, es decir 1403

Para modificar la contraseña he abierto el ejecutable con ghex, he buscado mbcfsjoup y he cambiado la m por una M.

Para editar el pin he abierto el ejecutable con el gdb en modo escritura y he ejecutado el siguiente comando `set{int}` 0x601074=2400

Bomba de Pablo Martinez Garcia

Contraseña=estaes
Pin=1998

Contraseña modificada= Estaes
Pin modificado=1996

Observando el código ensamblador de esta bomba podemos ver que tiene dos funciones, una llamada encriptar_pass y otra llamada encriptar_pin . Primero buscamos la contraseña y el pin cifrados

(gdb) x/s 0x601068
0x601068 <password>: "etvdix\n"

(gdb) x/wd 0x601060
0x601060 <passcode>: 3998

Para averiguar la contraseña ejecutamos el programa en gdb, introducimos “aaaaaa” como contraseña y avanzamos hasta después de la llamada a encriptar_pass. Comprobamos el string resultante:

(gdb) x/s $rsi
0x7fffffffde00: "abcdef\n"

Vemos que simplemente a cada letra le suma su posición, así que si a cada letra de etvdix le restamos su posición, obtenemos la contraseña original: estaes.

Para averiguar el pin analizamos la función encriptar_pin:

```
0x40078b <encriptar_pin>        lea    0x7d0(%rdi),%eax
0x400791 <encriptar_pin+6>      retq
```

Vemos que simplemente le suma 2000 a lo introducido, así que si al passcode le restamos 2000 obtenemos el pin original: 1998

Para modificar la contraseña he abierto el ejecutable con ghex, he buscado etvdix y he cambiado la primera e por una E.

Para editar el pin he abierto el ejecutable con el gdb en modo escritura y he ejecutado el siguiente comando `set{int}` 0x601060=3996

Bomba de Rubén Mogica Garrido

Contraseña=c0n7r453n14
Pin=2046

Contraseña modificada=C0n7r453n14
Pin modificado = 100

Observando el código ensamblador de esta bomba podemos ver que tiene dos funciones, una llamada cifrado1 y otra llamada cifrado2. Primero buscamos la contraseña y el pin cifrados, no hay variables llamadas <password> o <passcode>, pero hay dos variables llamadas <incognita1> e <incognita2>

```
(gdb) x/s 0x601068
0x601068 <incognita1>:  "c1p:v9;:v:>\n"

(gdb) x/wd 0x601060
0x601060 <incognita2>:  4186116
```

Para averiguar la contraseña ejecutamos el programa en gdb, introducimos “aaaaa” como contraseña y avanzamos hasta después de la llamada a cifrado. Comprobamos el string resultante:

```
(gdb) x/s $rdi
0x7fffffffddf0: "abcde\017\006\a\345\273", <incomplete sequence \372>
```

Vemos que a cada letra le ha sumado su posición, además ha seguido sumando después de que el string terminase. Por tanto si a cada carácter de c1p:v9;:v:> le restamos su posición, nos queda c0n7r453n14

Para averiguar el pin analizamos la función cifrado2:

```
0x400772 <cifrado2>     imul   %edi,%edi
0x400775 <cifrado2+3>   mov    %edi,%eax
0x400777 <cifrado2+5>   retq
```

Vemos que simplemente eleva al cuadrado el pin introducido, así que si le hacemos la raiz cuadrada a <incognita2> obtenemos el pin: 2046

Para modificar la contraseña he abierto el ejecutable con ghex, he buscado c1p:v9;:v:> y he cambiado la primera c por una C.

Para editar el pin he abierto el ejecutable con el gdb en modo escritura y he ejecutado el siguiente comando set{int} 0x601060=10000