Nombre: José Arcos Aneas
D.N.I : 74740565-H

Practica4 EC

Mi bomba es la siguiente:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/time.h>

char password[]="califrajilistico\n";
int  passcode  = 9128;

void boom(){
        printf("***************\n");
        printf("*** BOOM!!! ***\n");
        printf("***************\n");
        exit(-1);
}

void defused(){
        printf("·····························\n");
        printf("··· bomba desactivada ···\n");
        printf("·····························\n");
        exit(0);
}

int main(){
#define SIZE 100
        char pass[SIZE];
        int  pasv;
#define TLIM 5
        struct timeval tiempo1,tiempo2;

        gettimeofday(&tiempo1,NULL);

        printf("Introduce la contraseña: ");
        fgets(pass,SIZE,stdin);
        if (pass != password)
                boom();

        gettimeofday(&tiempo2,NULL);
        if (tiempo2.tv_sec - tiempo1.tv_sec > TLIM)
                boom();

        printf("Introduce el código: ");
        scanf("%i",&pasv);
        if (pasv!=passcode)
                boom();

        gettimeofday(&tiempo1,NULL);
        if (tiempo1.tv_sec - tiempo2.tv_sec > TLIM)
                boom();

        defused();

        return 0;
}
```

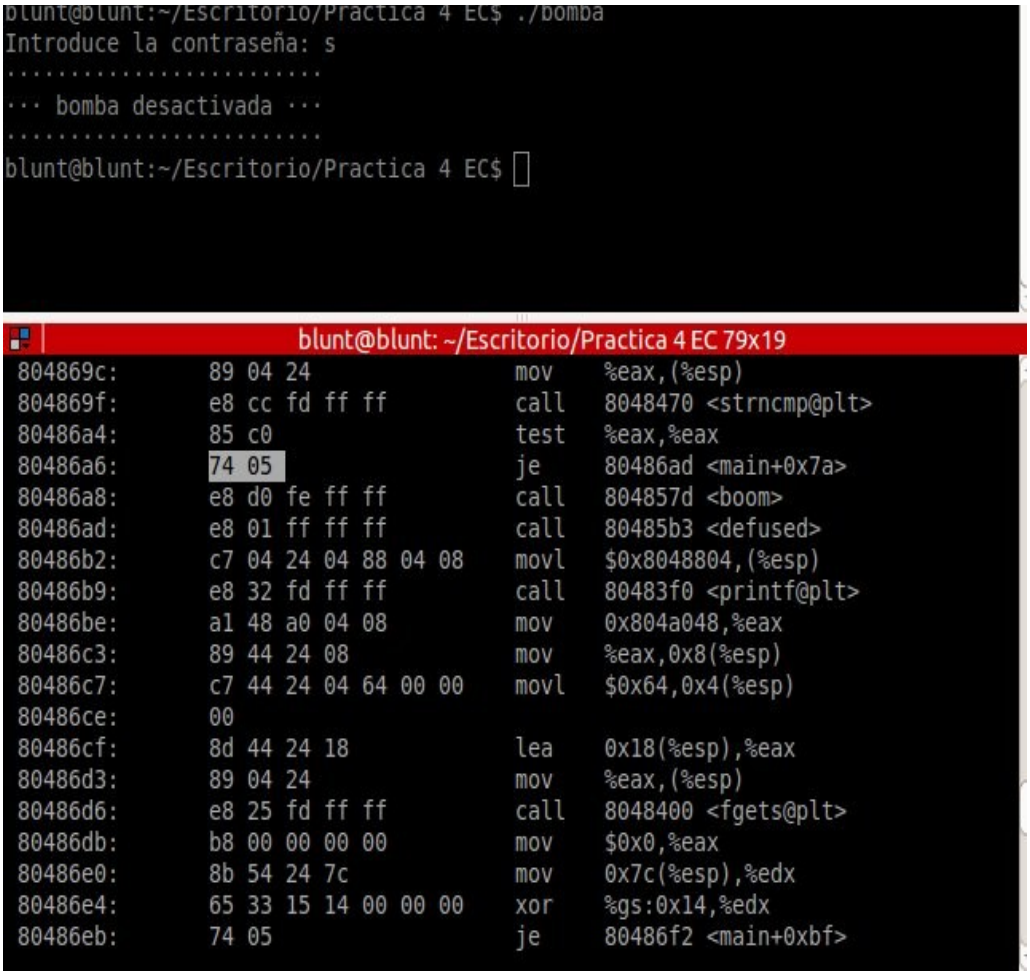
Para romper mi bomba solo hay que seguir los pasos de la guia de la ya que es muy similar, aunque en mi caso no uso strcmp.

Buscamos los saltos condicionales que usan la clave en su comprobación y los cambiar por saltos jmp cuya codificación es “eb” en hexadecimal. Usamos ghex para modificar el codigo hexadecimal de nuestra bomba con lo que la ejecución de esta nos vuelve a pedir una clave pero en este caso sea cual sea la clave la bomba se desactivará.

Para evitar las comprobaciones y desactivar las bombas he usado el editor hexadecimal "ghex". Para instalarlo he usado la versión de los repositorios y la he instalado con "sudo apt-get install ghex"

La bomba a desactivar es la siguiente:
Bomba de Miguel Porcel Jimenez .

Con “objdump -d nombre_bomba“ podemos ver el código ensamblador de la aplicación. En ambos casos podemos resaltar el siguiente codigo.



Como podemos ver se realizan varios saltos después de comprobaciones.

La codificación de estos saltos son 7e y 74 en hexadecimal.

Con ctr+h podemos abrir el buscador de cadenas de ghex y buscar las cadenas a las que pertenecen los saltos. Dado que no he visto el código de mis compañeros he preferido arriesgarme y cambiar en ambos casos los 4 saltos (o bifurcaciones condicionales) que existían en ambos programas a sabiendas de que dos de ellos probablemente pertenezcan a las comprobaciones del limite de tiempo para la escritura de las claves.
En la imagen anterior se puede ver, en el terminal de la parte superior, como al ejecutar la bomba e introduciendo cualquier código esta se desactiva.

Con esta bomba se ha procedido a determinar sus claves.

El proceso hubiese sido mas fácil si se le hubiese añadido información de depuración en la compilación.

Para sacar las claves de esta bomba he intentado usar el gdb pero ha sido frustrante puesto que no se han compilado con información de depuración y no ha encontrado símbolos de depuración.

Aun asi he intentado encotrar las claves incluyendo sentencia que me lo mostraran por pantalla aunque esto ha sido inutil.

Para encontrar(intentar) las claves he ido buscando por el código asm.

Por ejemplo me llamo la atención esta secuencia con la que estuve trabajando intentando obtener las claves de la bomba.

```
804879f:    8d 44 24 38     lea   0x38(%esp),%eax
80487a3:    89 04 24        mov   %eax,(%esp)
80487a6:    e8 55 fd ff ff  call  8048500 <strncmp@plt>
80487ab:    85 c0           test  %eax,%eax
80487ad:    eb 05           jmp   80487b4 <main+0xe0>
80487af:    e8 59 fe ff ff  call  804860d <boom>
```

Se puede ver como después de la llamada a stmcmp se realiza un test que condiciona un salto o no hacia la función que activa la bomba.

```
80487f9:    e8 f2 fc ff ff      call  80484f0 <__isoc99_scanf@plt>
80487fe:    8b 54 24 1c             mov   0x1c(%esp),%edx
8048802:    a1 4c a0 04 08          mov   0x804a04c,%eax
8048807:    39 c2           cmp   %eax,%edx
8048809:    eb 05           jmp   8048810 <main+0x13c>
804880b:    e8 fd fd ff ff  call  804860d <boom>
```

en esta ultimo trozo se observa como después de la llamada a scanf se produce una comparación entre eax y edx con lo que podemos estar seguros de que la clave del sistema esta almacenada hay.

El valor del salto “eb” corresponde con la modificación hecha a los saltos que habia anteriormente (7e y 74).

A continuación se adjunta el código ASM de la función “main” de la bomba seleccionada para desactivar.

```
080486d4 <main>:
80486d4:    55                  push   %ebp
80486d5:    89 e5               mov    %esp,%ebp
80486d7:    53                  push   %ebx
80486d8:    83 e4 f0            and    $0xfffffff0,%esp
80486db:    81 ec a0 00 00 00       sub    $0xa0,%esp
80486e1:    65 a1 14 00 00 00       mov    %gs:0x14,%eax
80486e7:    89 84 24 9c 00 00 00    mov    %eax,0x9c(%esp)
80486ee:    31 c0               xor    %eax,%eax
80486f0:    c7 44 24 04 00 00 00    movl   $0x0,0x4(%esp)
80486f7:    00
80486f8:    8d 44 24 28             lea    0x28(%esp),%eax
80486fc:    89 04 24            mov    %eax,(%esp)
80486ff:    e8 7c fd ff ff      call   8048480 <gettimeofday@plt>
8048704:    c7 04 24 64 89 04 08    movl   $0x8048964,(%esp)
804870b:    e8 50 fd ff ff      call   8048460 <printf@plt>
8048710:    a1 50 a0 04 08          mov    0x804a050,%eax
8048715:    89 44 24 08             mov    %eax,0x8(%esp)
8048719:    c7 44 24 04 64 00 00    movl   $0x64,0x4(%esp)
8048720:    00
8048721:    8d 44 24 38             lea    0x38(%esp),%eax
8048725:    89 04 24            mov    %eax,(%esp)
8048728:    e8 43 fd ff ff      call   8048470 <fgets@plt>
804872d:    c7 44 24 20 00 00 00    movl   $0x0,0x20(%esp)
8048734:    00
8048735:    c7 44 24 24 01 00 00    movl   $0x1,0x24(%esp)
804873c:    00
804873d:    eb 0a               jmp    8048749 <main+0x75>
804873f:    83 44 24 20 01          addl   $0x1,0x20(%esp)
8048744:    83 44 24 24 01          addl   $0x1,0x24(%esp)
8048749:    8d 54 24 38             lea    0x38(%esp),%edx
804874d:    8b 44 24 20             mov    0x20(%esp),%eax
8048751:    01 d0               add    %edx,%eax
8048753:    0f b6 00            movzbl (%eax),%eax
8048756:    3c 0a               cmp    $0xa,%al
8048758:    75 e5               jne    804873f <main+0x6b>
804875a:    8b 5c 24 24             mov    0x24(%esp),%ebx
804875e:    c7 04 24 40 a0 04 08    movl   $0x804a040,(%esp)
8048765:    e8 66 fd ff ff      call   80484d0 <strlen@plt>
804876a:    39 c3               cmp    %eax,%ebx
804876c:    eb 05               jmp    8048773 <main+0x9f>
804876e:    e8 9a fe ff ff      call   804860d <boom>
8048773:    8b 44 24 24             mov    0x24(%esp),%eax
8048777:    89 44 24 04             mov    %eax,0x4(%esp)
804877b:    8d 44 24 38             lea    0x38(%esp),%eax
804877f:    89 04 24            mov    %eax,(%esp)
8048782:    e8 f2 fe ff ff      call   8048679 <encriptar>
8048787:    c7 04 24 40 a0 04 08    movl   $0x804a040,(%esp)
804878e:    e8 3d fd ff ff      call   80484d0 <strlen@plt>
8048793:    89 44 24 08             mov    %eax,0x8(%esp)
8048797:    c7 44 24 04 40 a0 04    movl   $0x804a040,0x4(%esp)
804879e:    08
804879f:    8d 44 24 38             lea    0x38(%esp),%eax
80487a3:    89 04 24            mov    %eax,(%esp)
80487a6:    e8 55 fd ff ff      call   8048500 <strncmp@plt>
80487ab:    85 c0               test   %eax,%eax
80487ad:    eb 05               jmp    80487b4 <main+0xe0>
```

```
80487af:    e8 59 fe ff ff      call   804860d <boom>
80487b4:    c7 44 24 04 00 00 00        movl   $0x0,0x4(%esp)
80487bb:    00
80487bc:    8d 44 24 30                 lea    0x30(%esp),%eax
80487c0:    89 04 24            mov    %eax,(%esp)
80487c3:    e8 b8 fc ff ff      call   8048480 <gettimeofday@plt>
80487c8:    8b 54 24 30                 mov    0x30(%esp),%edx
80487cc:    8b 44 24 28                 mov    0x28(%esp),%eax
80487d0:    29 c2               sub    %eax,%edx
80487d2:    89 d0               mov    %edx,%eax
80487d4:    83 f8 05            cmp    $0x5,%eax
80487d7:    eb 05               jmp    80487de <main+0x10a>
80487d9:    e8 2f fe ff ff      call   804860d <boom>
80487de:    c7 04 24 7f 89 04 08        movl   $0x804897f,(%esp)
80487e5:    e8 76 fc ff ff      call   8048460 <printf@plt>
80487ea:    8d 44 24 1c                 lea    0x1c(%esp),%eax
80487ee:    89 44 24 04                 mov    %eax,0x4(%esp)
80487f2:    c7 04 24 96 89 04 08        movl   $0x8048996,(%esp)
80487f9:    e8 f2 fc ff ff      call   80484f0 <__isoc99_scanf@plt>
80487fe:    8b 54 24 1c                 mov    0x1c(%esp),%edx
8048802:    a1 4c a0 04 08              mov    0x804a04c,%eax
8048807:    39 c2               cmp    %eax,%edx
8048809:    eb 05               jmp    8048810 <main+0x13c>
804880b:    e8 fd fd ff ff      call   804860d <boom>
8048810:    c7 44 24 04 00 00 00        movl   $0x0,0x4(%esp)
8048817:    00
8048818:    8d 44 24 28                 lea    0x28(%esp),%eax
804881c:    89 04 24            mov    %eax,(%esp)
804881f:    e8 5c fc ff ff      call   8048480 <gettimeofday@plt>
8048824:    8b 54 24 28                 mov    0x28(%esp),%edx
8048828:    8b 44 24 30                 mov    0x30(%esp),%eax
804882c:    29 c2               sub    %eax,%edx
804882e:    89 d0               mov    %edx,%eax
8048830:    83 f8 05            cmp    $0x5,%eax
8048833:    7e 05               jle    804883a <main+0x166>
8048835:    e8 d3 fd ff ff      call   804860d <boom>
804883a:    e8 04 fe ff ff      call   8048643 <defused>
804883f:    b8 00 00 00 00              mov    $0x0,%eax
8048844:    8b 8c 24 9c 00 00 00        mov    0x9c(%esp),%ecx
804884b:    65 33 0d 14 00 00 00        xor    %gs:0x14,%ecx
8048852:    eb 05               jmp    8048859 <main+0x185>
8048854:    e8 37 fc ff ff      call   8048490 <__stack_chk_fail@plt>
8048859:    8b 5d fc            mov    -0x4(%ebp),%ebx
804885c:    c9                  leave
804885d:    c3                  ret
804885e:    66 90               xchg   %ax,%ax
```