Práctica 3: Bomba Digital
Juan Antonio Villegas Recio

DESACTIVANDO LA BOMBA DE: MAXIMILIANO SUAREZ

Lo primero que hice fue mirar el código en ensamblador para posteriormente cargarlo en **ddd**. Con run y nexti, me situé en

```
804868b:  89 c2                    mov    %eax,%edx
```

sospechando que en EAX podría estar la clave, ya que a este registro previamente se le asigna una dirección de memoria que podría ser la contraseña, mediante un volcado de memoria obtuve:

```
0x804a030 <password>:    "holaquetal\n"
```

Y tras una rápida ejecución concluí que **holaquetal** es la contraseña.
Seguidamente transformé mediante el editor hexadecimal **ghex** los saltos condicionales en incondicionales, para que así el programa aceptara como correctos cualquier contraseña y código independientemente del tiempo tardado en introducirse.

Observando el desensamblado en busca del código, me fijé en las siguientes líneas:

```
...
 804870b: e8 c0 fd ff ff           call   80484d0
<__isoc99_scanf@plt>
 8048710: 8b 54 24 34              mov    0x34(%esp),%edx
 8048714: a1 3c a0 04 08           mov    0x804a03c,%eax
 8048719: 39 c2                    cmp    %eax,%edx
 804871b: eb 05                    jmp    8048722 <main+0x112>
 804871d: e8 82 fe ff ff           call   80485a4 <boom>
…
```

Es decir, el código introducido en scanf() será almacenado en EDX y el auténtico código estará en EAX, que anteriormente estaba en la dirección 0x804a03c, llevando hasta la dirección 0x8048719 el puntero del depurador y mirando el valor de EAX obtenemos el código de la bomba: **1998**.

Además de las variaciones en los saltos he cambiado la contraseña a “holaquepasa”

DESACTIVANDO LA BOMBA DE: VÍCTOR CASTRO

Los primeros pasos fueron idénticos a los anteriores.

```
…
 8048739: 68 00 a1 04 08           push   $0x804a100
 804873e: e8 2d fd ff ff           call   8048470 <fgets@plt>
 8048743: 83 c4 10                 add    $0x10,%esp
 8048746: e8 c0 fe ff ff           call   804860b <setpassword>
 804874b: 83 ec 0c                 sub    $0xc,%esp
 804874e: 68 60 a0 04 08           push   $0x804a060
 8048753: e8 68 fd ff ff           call   80484c0 <strlen@plt>
 8048758: 83 c4 10                 add    $0x10,%esp
 804875b: 83 ec 04                 sub    $0x4,%esp
 804875e: 50                       push   %eax
 804875f: 68 60 a0 04 08           push   $0x804a060
 8048764: 68 00 a1 04 08           push   $0x804a100
 8048769: e8 82 fd ff ff           call   80484f0 <strncmp@plt>
 804876e: 83 c4 10                 add    $0x10,%esp
 8048771: 85 c0                    test   %eax,%eax
 8048773: 74 05                    je     804877a <main+0x89>
```

```
 8048775: e8 f7 fe ff ff          call   8048671 <boom>
…
```

En este caso, me fijé que a strncmp() se le pasan 3 argumentos, el primero almacenado en 0x804a100, que es la cadena introducida por el usuario en fgets(), el segundo almacenado en la dirección 0x804a060, que también es la dirección que se apila justo antes de llamar a strlen(), y el último es la longitud de la cadena que procesa strlen, almacenado en EAX. Sabiendo todo esto, podemos concluir que la cadena que representa la contraseña está almacenada en 0x804a060, haciendo un volcado de memoria, obtenemos la contraseña:

```
0x804a060 <password>:    "Granada\nAlhambra"
```

Luego la contraseña buscada es **Granada\nAlhambra**, a la hora de introducirlo en el programa se introducirá tan solo "Granada".

De nuevo procediendo de forma similar a la anterior, el desensamblado nos da una pista de donde buscar el código.

```
 80487c4: e8 71 fe ff ff          call   804863a <setcode>
 80487c9: 8b 15 64 a1 04 08       mov    0x804a164,%edx
 80487cf: a1 c4 a0 04 08          mov    0x804a0c4,%eax
 80487d4: 39 c2                   cmp    %eax,%edx
 80487d6: 74 05                   je     80487dd <main+0xec>
 80487d8: e8 94 fe ff ff          call   8048671 <boom>
```

Compara los valores almacenados en EAX y EDX, lo lógico es pensar que uno contendrá el valor introducido en scanf() y otro contendrá el auténtico código. La función setcode() almacena en EAX el valor introducido, por lo que EDX contendrá el auténtico código. Colocando el puntero del depurador en la dirección 0x80487d4, observamos el valor de ambos registros, pudiendo comprobar que EAX contiene el que nosotros le hemos introducido y EDX el valor **5**, que estaba en la dirección 0x804a164, siendo este el código de la bomba.

Mediante el editor hexadecimal **ghex** he cambiado la contraseña: la nueva contraseña es "Pacopepe.Alhambra"

CONCLUSIÓN

	Maximiliano		**Víctor**	
	Dirección	**Dato**	**Dirección**	**Dato**
Contraseña	0x804a030	"holaquetal\n"	0x804a060	"Granada\nAlhambra"
Código	0x804a03c	1998	0x804a164	5