BOMBA DE ANTONIO DAVID LOPEZ MACHADO

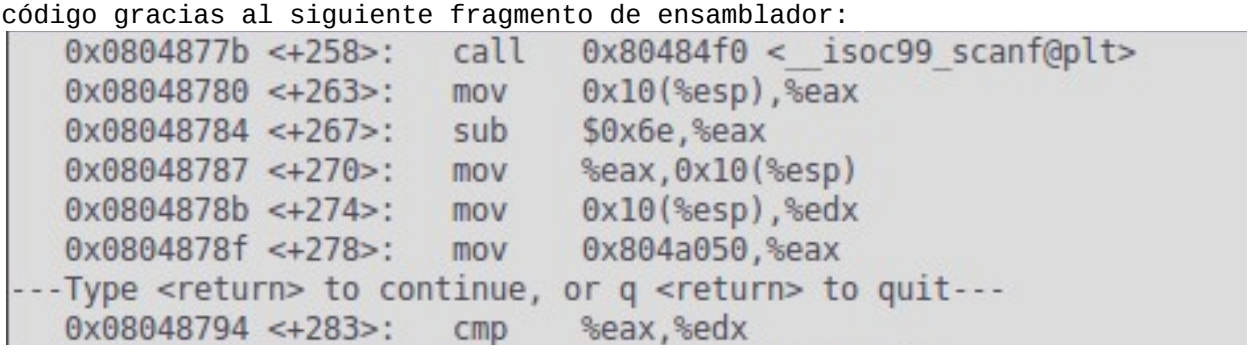
CONTRASEÑA: soyunabomba
CÓDIGO: 80987

En primer lugar, para averiguar la contraseña de esta bomba he utilizado el comando ltrace -i -S ./bomba_AntonioDavidLopezMachado, mediante el cual he obtenido lo siguiente, tras introducir una contraseña aleatoria, en este caso, aaaaaaaaaaaa:

Donde podemos ver que aparecen las funciones strlen(“sp{xrfhvukk\n”) y strncmp(“abcdefghijka\n”, sp{xrfhvukk\n”, 12) , lo que nos indica que la contraseña, en este caso codificada es sp{xrfhvukk\n, y también aparece la contraseña que nosotros hemos introducido, pero en esta ocasión, también se ha codificado obteniendo la siguiente: abcdefghijka\n, finalmente vemos el 12, que nos indica la longitud de la contraseña original.

Tras ello podemos ver como al primer carácter de nuestra contraseña no se le ha hecho nada, pero al resto si, y es, en este caso, aplicarle un ciclo for, que sume desde 0 (al primer carácter) hasta 10 (el penúltimo carácter) dejando el último igual pero esto ha sucedido dado que la contraseña que hemos introducido tiene una longitud de 12 sin contar el \n, por tanto debería de ser de 11 y además el \n para dar como resultado 12. Por tanto estudiaremos nuestra contraseña desde “a” hasta “k”. A partir de esto, es sencillo apreciar que si aplicamos la misma codificación pero al revés a la contraseña que hemos encontrado, podremos obtener la contraseña original que, gracias a ello y a la tabla ASCII, ha resultado ser: soyunabomba.

Por otro lado, para averiguar el código de esta bomba, he recurrido al gdb para ver el código en ensamblador y tras estudiarlo detenidamente, he obtenido el código gracias al siguiente fragmento de ensamblador:



Aquí, en primer lugar he colocado un punto de ruptura justo después de la función scanf, ya que esta lee se utiliza para leer el código que nosotros introducimos. Posteriormente si nos fijamos en el código, vemos que mueve el dato introducido al registro %eax, al que posteriormente le resta 0x6e, que en decimal es 110.
Posteriormente, mueve el dato que hemos introducido, que en mi ocasión ha sido 8888 + 110 = 8998 a la pila, y tras ello, lo mueve al registro %edx. Aquí es donde debemos fijarnos puesto que mueve lo que hay en la dirección 0x804a050 a %eax y posteriormente compara %eax con %edx, por lo que si en %edx se encontraba nuestra contraseña, 8998, en %eax se encuentra la contraseña original menos 110, por lo que para saber la contraseña final debemos sumarle a lo que encontremos en ese registro, 110.

Por tanto, si hacemos un print $eax vemos que aparece el valor

```
(gdb) print $eax
$3 = 80877
```

que si le sumamos 110, obtenemos 80987 que resulta ser el código que buscábamos.

Aquí podemos ver como se desactiva:

```
ixjosemi@ixjosemi-GE70-2PE:~/Downloads$ ./bomba_AntonioDavidLopezMachado
Introduce la contraseña: soyunabomba
Introduce el código: 80987
************************
*** bomba desactivada ***
************************
ixjosemi@ixjosemi-GE70-2PE:~/Downloads$
```