Jesús Miguel Jaldo Ruiz 2ºD

DESACTIVACIÓN DE LA BOMBA

CONTRASEÑA: `lbkjbjnvr`

CÓDIGO: `6948`

La bomba cuenta con una codificación de la contraseña que se introduce mediante teclado utilizando el número del Cesar pero de una forma especial, los valores pares del vector char que le pasamos como parámetro los valores pares de este se aumentaran 1 dentro del rango alfabético (ej: a → b), y los impares se reducirán 1.

Para codificar la entrada tenemos una función denominada decodificar (decodificar(char contra[])) que recorre el array que se le pasa como parámetro y sus valores se irán aumentando o disminuyendo dependiendo de si la posición del array sea par o impar.

La explicación de cómo se quedan estos arrays antes y después de decodificar es la siguiente:

1.- Antes de llamar a la función decodifica() el array está así:

```
pass = "lbkjbjnvr\n"
```

2.- Después de llamar a la función decodifica():

```
pass = "malicious\n"
```

Una vez explicado el funcionamiento de la función se explicará el "truco" o atajo para poder resolver la bomba. Si desensamblamos el ejecutable, podemos ver que hay una llamada a la función decodificar() justo después de introducir la contraseña. La funcionalidad de esta función se debe a la aplicación de una pequeña modificación a la contraseña escrita por la persona que intenta desactivar la bomba (como se ha explicado antes).

Si el alumno se introduce en el código de la función decodificar encontrará un copioso código el cual es un tanto dificil de entender, y como veremos más adelante no será necesario (a no ser que se tenga curiosidad por ver exactamente lo que hace).

El alumno puede estar mucho rato intentando comprender esta función, pero también se puede dar cuenta de que el main es exactamente igual al ejemplo visto en clase solo que cambia en que se llama a la función decodifica() . En este caso se podría optar directamente por consultar el valor de la variable %rdi, la cual contedrá el valor de la cadena introducida después de la modificación, la cual se comparara con el valor password.

PASOS PARA AVERIGUAR LAS CONTRASEÑAS

- En primer lugar nos moveremos hasta la línea donde se llama la función con un breakpopint

  br *main+94

- Tras esto si queremos investigar a fondo lo que sucede en la función podremos entrar para ver su contenido. En este caso lo omitiremos porque no es del todo necesario averiguar qué hace la función, nos bastará con ver el resultado de la misma.
- Comprobaremos como siempre los valores de las variables %rsi (password) y%rdi (introducido por nosotros).

  (gdb) p(char*)$rsi

  $1 = 0x601070 <password> "malicious\n"

  (gdb) p(char*)$rdi

  $2 = 0x7fffffffdbf0 "n`mhdhptt\n"

  Con esto sabemos que la contraseña ha cambiado, en este caso al introducir la contraseña correcta “malicious” cambia a “n`mhdhptt\n”. Si nos fijamos bien vemos que los valores pares aumentan 1 y los impares decrementan 1.

- Una vez sepamos cual es la modificación que se realiza a nuestra contraseña lo unico que habrá que averiguar es el password con la que se comprara, para ello se usará el siguiente comando:

  (gdb) p(char[10])password

  $3 = "malicious\n"

  Con esto sabemos que la contraseña con la que estamos comparando es “malicious\n” y asi podremos ordenar de manera adecuada la contraseña que introducimos para que al decodificarla esta cambie su valor por “malicious\n”.

- Una vez sabemos la contraseña, procederemos a averiguar el código. Para ello paramos la ejecución del programa, quitamos el breakpoint que tenemos y lo ponemos en la línea que compara el código que hemos introducido con otro:

  br *main+240

  Un poco mas arriba podremos ver que el valor %eax donde se almacena el codigo introducido pasa a duplicar su valor (add %eax,%eax) y por tanto ese nuevo valor será el que se comparará con passcode. Ahí vemos que eax vale el doble de lo introducido y originalmente valía el valor introducido.

- Una vez sepamos cual es la modificación que se realiza a nuestro código lo unicio que habrá que averiguar es el password con la que se comprara, para ello se usará el siguiente comando:

  (gdb) p(int)passcode

  $4 = 13896

  Con esto sabemos que el código con el que estamos comparando es “13896” y así podremos calcular de manera adecuada el código que introducimos para que al duplicar su valor este cambie a “13896”.

- Ya sabemos la contraseña y el código, ahora solo queda volver a ejecutar el programa, introducirlos y desactivar la bomba.