DESACTIVACIÓN DE LA BOMBA

CONTRASEÑA: abrelabomba
PIN: 1997
CONTRASEÑA MODIFICADA: acrelabomba
PIN MODIFICADO: 2815
COMPILADO CON: gcc -Og bomba.c -o bomba -no-pie

La bomba que he creado cuenta con una contraseña y un pin ya cifrado en el código. Para comprobar que la contraseña que nos introducen es correcta lo que he implementado es un método compruebaPass(char pass[]) que lo que realiza es la codificación de la contraseña introducida para luego compararla con la contraseña que tengo en el código, la que desbloquea la bomba, que está codificada, este método devuelve true o false dependiendo si es o no correcta. La codificación que he realizado es intercambiar pares de caracteres, produciendo por ejemplo si introducimos la palabra “hola” la codificación de esta será “ohal”. En cuanto al pin lo que he hecho es codificar el pin en su variable como el pin original con los pares cambiados (pares cambiados en el caso de la contraseña que yo le he puesto que es 1997 y la tengo guardada como 9179) y lo único que realiza para comprobar si la contraseña introducida es la correcta es restarle un valor que produzca la contraseña, en el caso que he realizado es restándole 7182 al valor 9179.

PASOS PARA AVERIGUAR LA CONTRASEÑA Y EL PIN:

1. Lo primero es poner un breakpoint (b) en el main, para ello realizamos “b main” seguidamente realizamos run y se nos cargan todos los registros.

2. Ahora ponemos los siguientes breakpoints en el main:

```
b *main+96
b *main+220
b *main+226
b *main+231
```

3.Ahora ejecutamos next (n) introducimos una contraseña cualquiera, tenemos que recordarla en mi caso va a ser “holaquehace”, y seguidamente se nos quedara en el breakpoint “main+96”, realizamos setepi (si) una vez y entramos en la subrutina de “compruebaPass”.

4. Ahora ponemos los siguientes breakpoints en compruebaPass:

```
b *compruebaPass+34
b *compruebaPass+49
```

Los ponemos aquí por que se realizan las subrutinas de “LongitudCadena” dos veces, queremos ver que es lo que realiza, pero entendemos que nos calculará el tamaño de las cadenas.

5. Una vez puesto estos breakpoints realizamos una vez next y comprobamos que en la operación “mov %eax, %r12d” se encuentra el tamaño de la cadena que nosotros hemos introducido en este caso en %eax el valor es de 11, que este luego va a ser trasladado a %r12d. Realizamos ahora una vez stepi (si) y vemos que estamos en “compruebaPass+37”, ahora vamos a ver el valor que hay en password, para ello realizamos:

```
p (char*)0x601070
```

(Igual el valor de 0x601070 puede variar)

Al ver el valor que hay dentro nos aparece una cadena de caracteres, concretamente la cadena "baeralobbma\n", pero esta no es la contraseña de la bomba. Después de comprobar esto realizamos next una sola vez y llegamos al otro breakpoint que pusimos el de “compruebaPass+49”, el cual va a calcular la longitud de la cadena almacenada en password la cual tiene 11 caracteres (sin contar el /n) y lo guarda en %eax.

6. Ahora ponemos el siguiente breakpoint en compruebaPass:

```
b *compruebaPass+68
```

después realizamos un next y vamos a ver como se va a realizar la comprobación de si ambas cadenas la original y la que hemos introducido es de la misma longitud, comprueba valor de %eax (longitud de la cadena password) y %r12d (longitud de la cadena que hemos introducido), aquí lo que hacemos es si la cadena que hemos introducido tiene mismo valor que la de password no hacemos nada en cambio si hemos introducido una cadena mas pequeña o más grande lo que hacemos es cambiar el valor de %eax o %r12d para que ambos sean iguales, para ellos podemos realizar lo siguiente:

```
"set $r12d=11" o "set $eax=11"
```

Siendo 11 el valor que queremos introducir. Con esto nos saltamos la comprobación de que ambas cadenas sean iguales. Destacar que para comprobar los valores que hay en %eax y en %r12d se puede realizar:

```
p (int)$r12d  o p (int)$eax
```

7. Ahora ponemos el siguiente breakpoint:

```
b *compruebaPass+152
```

Seguidamente realizamos next y comprobamos el valor de %rdi, para ello realizamos:

```
p (char*)$rdi
```

Con el que nos damos cuenta que en %rdi tenemos guardada el valor que habiía en password que no es la contraseña final si no una contraseña codificada. Ahora si realizamos 3-4 stepi nos damos cuenta que vamos a entrar en un bucle for, ya que si vemos en “compruebaPass+109” se suma a %eax (inicialmente puesto a 0) un valor de 2 y luego se comprueba que %eax no sea igual a %r12d que es el tamaño de la contraseña que estamos codificando, anteriormente calculado. Destacar también que si realizamos:

```
p (char*)$rbx
```

Vamos a encontrarnos con la contraseña que hemos introducido lo que deducimos de esto es que en %rdi tenemos la contraseña del “password” y en %rbx la cadena que hemos introducido.

8. Ponemos los siguientes breakpoints:

```
b *compruebaPass+120
b *compruebaPass+125
```

Ejecutamos seguidamente con next y vemos que en compruebaPass+120 lo que se hace es acceder a la posicion i+1 de la cadena que hemos introducido y luego en compruebaPass+125 lo que se compara es el carácter de la posición i+1 con “0xa=/n” en hexadecimal.

9. Ponemos los siguientes breakpoints:

```
b *compruebaPass+133
```

b *compruebaPass+167

Al poner este breakpoint si imprimimos estando en este el valor de %rdi con:

p (char*)$rdi

Y luego lo imprimimos despues de realizar un stepi, vamos a ver que el valor de %rdi ha cambiado, y lo que ha ocurrido es que los pares de elementos se estan cambiando. Si ahora realizamos todos los pasos con next hasta que se realicen todos los 11 caracteres vemos que nuestra palabra que era “holaquehace” se ha convertido en “ohaluqhecae”. Una vez que hayamos llegado a compruebaPass+167 lo que vemos es que en compruebaPass+174 se realiza un strncmp de password (la que esta guardada en el codigo) y la que se ha generado intercambiando valores, es decir, en nuestro caso el de “ohaluqhecae”, con esto deducimos que lo que se ha realizado es el intercambio de los pares de caracteres de nuestra contraseña introducida y la ha comparado con la contraseña guardada en password lo que nos da a entender es que en password lo que existe es una contraseña con los pares cambiados, luego si la contraseña que esta guardada en password anteriormente hemos visto que era "baeralobbma\n" si le damos la vuelta a los pares sin contar el “\ n” podemos deducir que la contraseña de la bomba es “abrelabomba”.

10. Como ya sabemos la contraseña para hacerlo más rapido lo que he realizado es con el gdb abierto sin cerrarlo ponemos run de nuevo, pulsamos en yes ponemos el siguiente breakpoint:

b *main+125

y ejecutamos con next ponemos la contraseña y ejecutamos next hasta llegar a main+125 y ejecutamos stepi hasta llegar a main+135 en el que se compara los 5 segundo del gettimeofday, si vemos que %rax no tiene valor menor o igual a 5 lo que hacemos es poner este valor a 5 con:

set $rax=5

Si es menor o iguala 5 no realizamos esto. Realizamos seguidamente next e introducimos por ejemplo 1234, seguidamente hacemos stepi y nos encontramos en main+226 en el cual si imprimimos %eax con:

p (int) $eax

Vemos que tiene un valor de 9179, luego si vemos lo que pasa en main+226, nos damos cuenta que se produce una resta del valor 0x1c0e= 7182 a %eax, es decir, que en %eax después de realizar la intrucción de main+226 con stepi el valor que nos queda es el de “1997” ya que si lo imprimimos con :

p (int)$eax

Nos aparece que el valor es “1997”, si luego observamos en main+231 se compara 0xc(%rsp) con %eax, si imprimimos 0xc(%rsp) con:

p* (int*)(0xc+$rsp)

Podemos observar que es el valor que hemos introducido y que lo compara con “1997” lo que nos hace ver que el pin de la bomba es finalmente “1997”. Tan solo nos quedaría probarlo.

CAMBIO CONTRASEÑA DE LA BOMBA

Entramos en gdb con la bomba, ponemos el archivo para poder escribir con “set write on”, seguidamente como sabemos que en password estaba la contraseña codificada y en passcode el pin codificado simplemente modificamos ambos con:

```
set*(char*)0x601070='c'
set*(int*)0x601068=9997
```

Y con esto conseguimos que la contraseña de la bomba sea “acrelabomba” y pin “2815”. El pin es este por que como dijimos a 9997-7182=2815.