Desactivación bombas de compañeros

Alejandro Rubio Martínez

JAMM_bomba2020.

Lo primero que veo en la bomba al leer el código en ensamblador es una función sospechosa llamada `__fcheck_plt_` en `<main+96>` así que voy a meterme dentro. Dentro de la función vemos un strcmp en `<__fcheck_plt_+46>` así que vamos hasta el haciendo stepi. Una vez encima probamos a ver el contenido del registro %rsi haciendo `p(char*)$rsi` y vemos como la contraseña es "defuse". Ahora para que no nos explote la cambiamos a la que hemos introducido ("hola") haciendo `set $rsi="hola\n"`.

Seguimos avanzando en el main teniendo cuidado de que no nos explote por el tiempo, para ello haciendo en `<main+126>` la instrucción `set $rax=0x3c`.

Ahora buscamos la siguiente instrucción sospechosa, y esta sería un `cmp $eax,0xc(%rsp)` en `<main+215>`. Avancemos hasta él y probemos a hacer `p*(int*)(0xc+$rsp)` y obtenemos el pin que nosotros hemos introducido, así que comprobemos el %eax haciendo `p(int)$eax` y obtenemos el 4525 que es sospechoso de ser el pin. Para continuar con el programa hagamos `set $eax=4525` para que no nos explote. Ahora ya solo seguimos la bomba teniendo cuidado en `<main+251>`de cambiar el tiempo para que no nos explote. Efectivamente la bomba se ha desactivado.

```
#Esto sería un archivo .gdb
file JAMM_bomba2020
#Establecemos el punto de ruptura
br *main+96
run
#La contraseña
enchufe
#Nos metemos en la funcion
si
si
si
si
si
si
si
si
si
si
si
#Veamos la contraseña
p(char*)$rsi
#Cambiemos la contraseña suponiendo que hemos introducido enchufe
set $rsi="enchufe\n"
#Pongamos un breakpoint en el pin
br *main+215
#Continuemos hasta el
c
#El pin
10
#ahora veamos el valor del pin
```

```
p*(int*)$eax
#Por último cambiemos su valor suponiendo que hemos introducido 10
set $eax=10
```

MRL_bomba

En esta bomba lo que vemos primero es muchas funciones implementadas por el compañero, entre la que destacan dos de nombres sospechosos en `<main+108>` y `<main+197>` de nombre acierto1 y acierto2. Avancemos hasta la primera y a ver que contiene. Efectivamente vemos un strncmp en `<acierto1+85>` así que avancemos hasta él.

Haciendo un `p(char*)$rsi` vemos la contraseña que sería "nocagaste".

Vamos ahora a acierto2. No hay ninguna instrucción `cmp` así que metámonos en la función veamos en `<acierto2+9>`. En esto vemos una instrucción `cmp` en `<veamos+13>` por lo que volvemos a avanzar hasta ella. Viendo que justo antes hay una función llamada código vamos a comprobar %eax. Usando `p(int)$eax` obtenemos 242 que es el código.

bomba_SBR_2020

Esta bomba la he resuelto de dos simples pasos debido a la similitud con la mía. Primero he visto una llamada a strncmp en `<main+134>` y me he ido a por ella y al hacer `p(char*)$rsi` he obtenido "camellotactico" que es la contraseña.

Ahora me he ido a por la instrucción `cmp` de la línea `<main+268>` y al hacer `p(int)$eax` he obtenido el código que es 851.

Muy bonito lo de los colores por parte del compañero.