**郭又嘉**

**AosoraKuo**

**B06705020**

**Casino**

在看完程式之後

顯然read\_int()不能進行buffer overflow

但是read()可以

所以可以在讀name的地方放上shellcode

然後再看到casino裡面其中一行

guess[idx] = read\_int();

沒有顧到他可能會超過guess原本預設範圍的可能性

然後發現這個程式有用got table

所以使用got table hijacking

因為read\_int()的回傳值是4bytes

而地址是8bytes

所以要尋找能夠改兩次的目標

也就是改第一次沒有影響改第二次之後還會呼叫到的function

發現puts是一個好的人選

因為他在猜中lottery號碼之後會run

至於hijack的地址就是看如何塞name

在塞name的過程中發現name的堆疊下面還有seed和age

所以我先塞0x24個 ’/x00’ 進去name再塞shellcode

如此一來shellcode的開頭會比原本name的位置多了0x24

也就是到了0x602114

然後因為是little-endian

所以兩次要改的值分別是0x602114, 0x0

也就是6299924跟0

然後就是puts在got table的存放位置了

可以查到是在0x602020

所以idx的兩次的值就是 decimal(- (0x6020d0-0x602020)/4) + 1 = -43和

decimal(- (0x6020d0-0x602024)/4) + 1 = -42

再來再修改一下原始碼讓他執行時可以吐出lottery的值就好(casino1.c)

因為seed已經被我們塞滿了0而不會改變

最後就是將pwntools寫好讀出lottery的值

再把他轉成讀網站上題目的值就可以在/home/casino/cat flag拿到flag了(casinopwntl.py)