Week3 writeup

Web

Ordinal Scale

> 这仍然是我感觉最有情怀的一部作品 ——一名刀剑老粉的心声

进去登录页，F12发现有source.zip，下载下来发现是网站源码，所以又是一道代码审计，先点进去玩了一下，死着死着玩到了第二名，然后发现无法再前进了，查看源码，fight函数的逻辑明确说明没办法通过对战到达第一名，那只有其他的办法了，看了一下主逻辑，首先$encryptKey不知道，假设知道的话，通过$encryptKey可以推算出sign签名，从而知道Monster类的$encryptKey，然后就可以推出cookie的签名逻辑，从而通过修改cookie来影响服务器，看到109行的unserialize函数的时候更加坚定了我的想法，但是$encryptKey真的不知道该怎么弄出来，最后E99给的hint中提到由于sprintf的逻辑，如果用户名输入%s，是可以把$encryptKey弄出来的,最后真的弄出来了，接下来顺着程序逻辑，就可以很容易算出sign，然后陷入了怎么利用unserialize函数的问题，由于78行的if判断，以为无法改动session['rank']，但思路一直放在序列化Rank类这个方向上，想到半夜3点多实在想不出来就睡觉了，第二天再看一遍源码，发现Game类的Rank属性是公有属性，而其他的都是私有属性，猜想如果反序列化出来一个Game类，通过修改构造时Rank的排名，是不是可以影响到原有的Game类的公有属性，于是序列化一个新的Game类，把Rank属性的rank值设为1，然后反序列化，签名，链接，base64_encode，转成cookie，扔到服务器上，成功.....

Cosmos的二手市场

> 这是我第一次因为python被卡住.....python的异步有那么不堪吗.....

点进去，界面非常古朴，F12没有啥提示信息，唯一给出的hint是赚到一个亿，但是服务器价格不变动，每次卖还有收手续费，傻子都知道会亏，所以纠结了好久，没有其他办法的情况下，猜想是交易系统中的高并发竞争的漏洞，简单来说就是同时卖多次，于是我用python的asyncio+aiohttp异步请求，一次买5个，然后发40个post请求，一次卖1个，结果服务器稳如老狗，尝试增加卖的请求数，还是一样，最后被逼无奈，既然是并发问题只能找go，于是用go协程重写了一下程序，由于一开始资金有限，buyer一次买10个，暂停1秒，solver一共40个，一次卖一个，死循环跑，没过多久就赚了10W，然后继续加大solver的卖出数量，加大buyer的买入数量，减小暂停时间，没过多久成功达成小目标，拿到flag