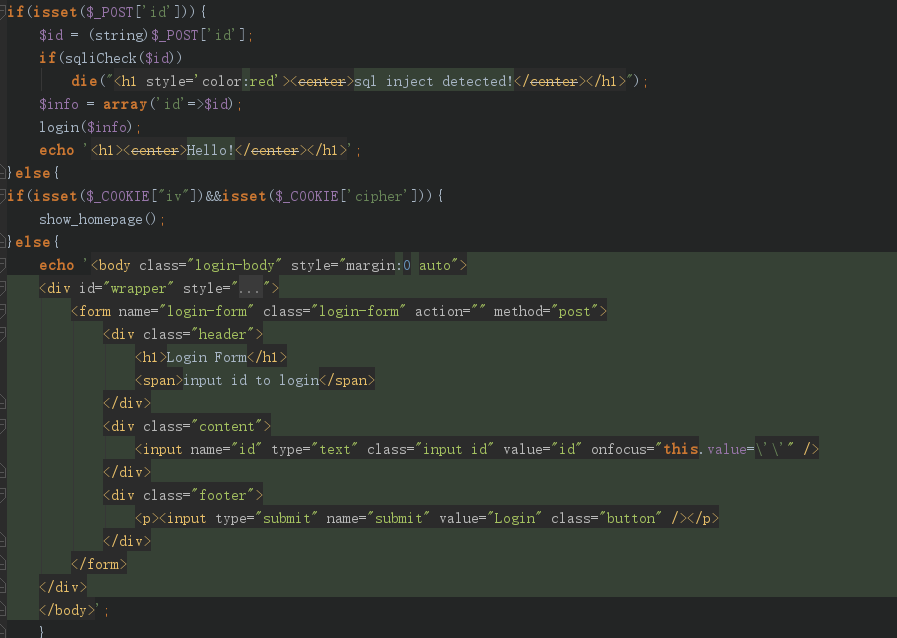
cbc翻转攻击

题目链接

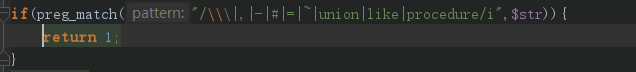
[http://ctf5.shiyanbar.com/web/jiandan/index.php](http://shiyanbar.com/ctf/javascript:;)

有源码在test.php页面

分析代码过程



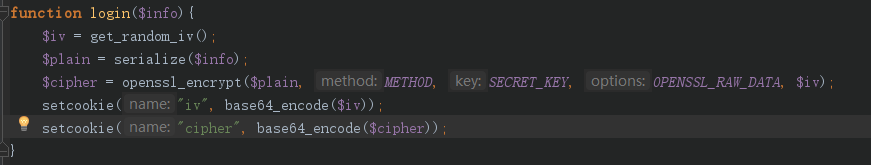
如果post id，将id转字符串，然后进入sqliCheck函数，



如果函数返回true，就停了，

否则，将id转成array

然后到login函数

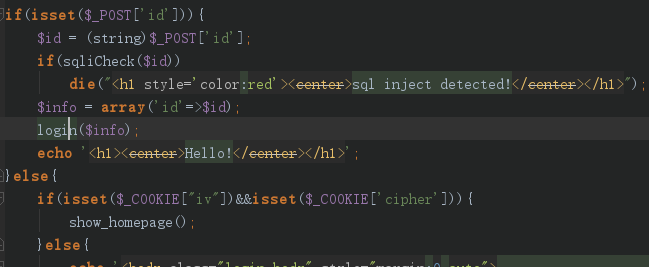


这里将$info序列化，然后aes加密，进行base64后，进行setcookie

这里也就得到了，抓包发现的那个



这里要注意的是，它的逻辑是，如果你输入有id，那是不会检查你的cookie的，也就是说，有id的情况下，是不会执行sql语句的，也就是会一直显示那个Hello



如果没有post id，那就检测cookie，

如果存在iv，和cipher 那就进入show\_homepage();



将两个参数base64解密，然后用openssl\_decrypt()将密文解密，再进行反序列化，然后进行sql查询

进过上面的分析得知，过滤了很多函数，其实最重要的是过滤了#号和-号，这样子就无法

将那段sql语句后面的,0注析掉了

所以只能在cookie处入手

这里用到CBC反转攻击

参考文章http://sec.chinabyte.com/228/13511728.shtml

文章的例子



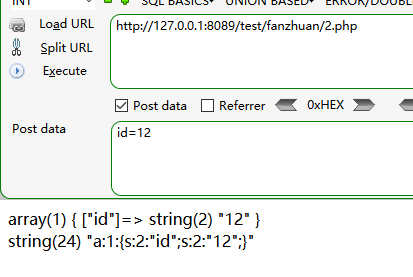
第一块的密文用来生成后一块明文

也就是说要修改第一块的密文

6对应的第一块的明文是2，偏移量为2

在题目中，我们知道的是密文，还有偏移量，第一段的明文，那就可以修改明文的数据

参考writeup中的，输入id = 12



得到的$info和$cipher

得到偏移量为4

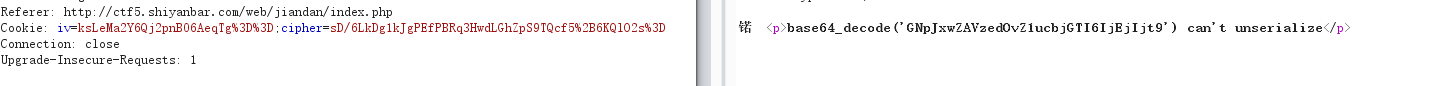


相对应的第一块为}

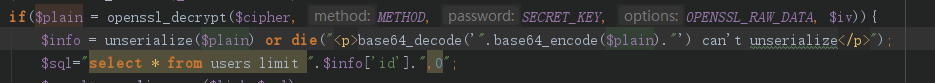
我们尝试将2换成#，起到注析后面语句的作用

得到新的cipher





返回源码看



是无法进行序列化

并得到了一个新明文

无法进行序列化是因为iv不正确

那根据 新iv = 原iv^原明文^新明文 可以得到新的iv



也就是成功将2修改成#号了

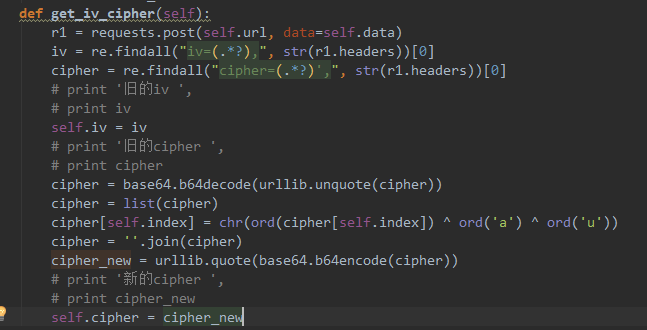
但是这里有一个问题就是，虽然#号可以注析后面的语句，但是一旦我们的语句很长，那我们要修改的密文就可能不是第一块了，那就会导致翻转很麻烦，那就是想办法用其他方式注析掉后面的语句，这里可以用%00截断（和改包上传的原理一样），这里不可以直接将%00写到payload里面去，因为会被解码的，所以可以当是一个不可见字符，用python的chr(0)写入。

将上面的逻辑整理一下

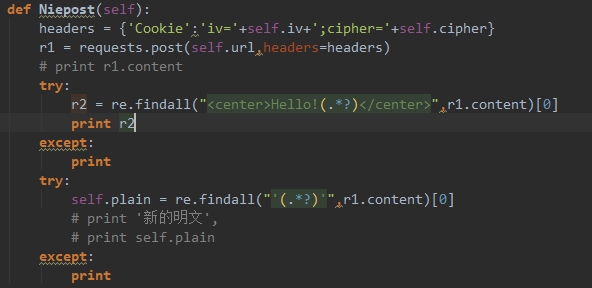
1. 输入id，访问得到iv和cipher
2. 将cipher内容改变，然后去掉id，将iv和新cipher加到cookie里面去访问，显示无法序 列化，得到明文，然后要获取新iv 新iv = 新明文^原明文^原iv
3. 将新cipher和新iv添加到cookie里面，再次访问，成功反序列化，进行sql查询

按照上面的思路，写脚本

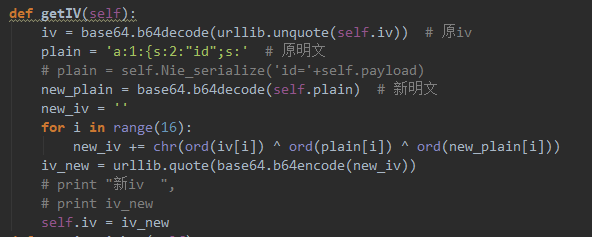
第一次获取iv和cipher



访问请求得到新明文

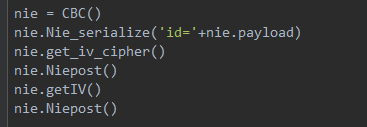


获取新iv



再次访问Niepost执行SQL语句

所以程序运行顺序就是



（这是本地搭建的）



