flag售货机wp

flag售货机 POINT: 400
题目ID： 68
题目描述： http://115.28.78.16:13333/b5062b7c25276283ed5f8a5e9026b762/
欢迎来到flag售货机
Hint: 1、源码泄露

这里提示了题目有源码，这里说几种常见的源码泄露：
1、编辑器的自动备份，一般有index.php.bak,index.php~,.index.php.swp,甚至还有比较少见的.index.php.swo
2、还有之前用过的lfi的方式可以读源码
3、代码仓库导致的源码泄露，一般是.git/.hg/.svn，这里只推荐一个工具是https://github.com/kost/dvcs-ripper

这份代码是perl写的，建议linux上使用，linux中自带perl环境，+x就可以了

拿到源码后分析源码

这里漏洞其实我留的比较刻意，看的懂代码的很容易找到问题所在，先分析逻辑。

1、注册登陆后赠送了一张充值卡，可以充值2000000块，但是flag需要$2333333
2、充值卡是绑定用户的，也就是我们不能通过使用别人的充值卡来多次充值
3、因为前面出过一道竞争的题目，你可能会想到这里使用竞争来解决问题，这里涉及到了一个session锁的问题

具体看这个http://konrness.com/php5/how-to-prevent-blocking-php-requests/

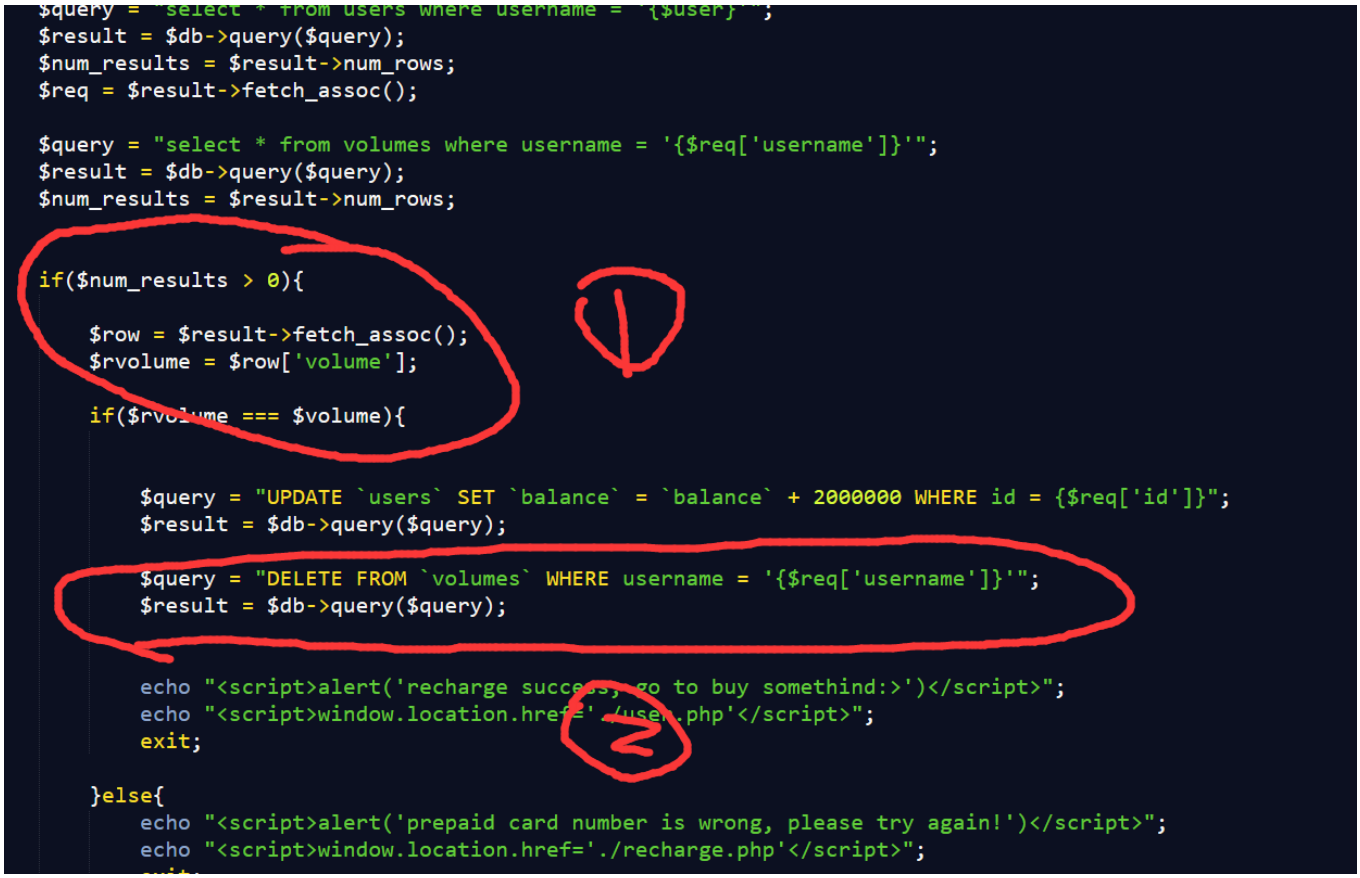
那么上面的办法都没有用，我们只能从recharge.php找问题，这里其实漏洞留得很刻意了，有心肯定能发现。

```php
$user = $_SESSION['user'];

$query = "select * from users where username = '{$user}'";
$result = $db->query($query);
$num_results = $result->num_rows;
$req = $result->fetch_assoc();

$query = "select * from volumes where username = '{$req['username']}'";
$result = $db->query($query);
$num_results = $result->num_rows;


if($num_results > 0){

    $row = $result->fetch_assoc();
    $rvolume = $row['volume'];

    if($rvolume === $volume){


        $query = "UPDATE `users` SET `balance` = `balance` + 2000000 WHERE id = {$req['id']}";
        $result = $db->query($query);

        $query = "DELETE FROM `volumes` WHERE username = '{$req['username']}'";
        $result = $db->query($query);


        echo "<script>alert('recharge success, go to buy somethind:>')</script>";
        echo "<script>window.location.href='./user.php'</script>";
        exit;

    }else{
```

下面的语句都使用了从第一个里select出来的数据，这样会使前面注册时候的过滤无效，形成二次注入。

但是这里的select只用来判断充值卡号是否相等，所以除非使用二次盲注，否则我们无法获得更多数据，但是由于flag是写死在代码里的，而mysqli的连接方式不允许多语句执行，我们无法执行一个update语句，那么这里我们只能使用别的办法。

这里我们应该整理一下我们的要求



1、我们构造的注入语句，首先需要使输入的充值卡号和select出来的相等，进入判断
2、其次我们需要让下面的delete无效，这样才能反复多次的充值

事实上这里有很多有趣的解决方案，所以我才说这是一个非常有趣的题目

这里我使用的方法是一个小trick，熟悉sql语法的人会知道union，你可能不知道的是，联合查询只存在于select中，而对于update等语法，如果想要执行多条，只能使用分号来多语句执行，如果使用union就会直接报错退出。

根据上面的分析，你可能一下子任督二脉贯通，写下了下面的payload。

```
' union select 1,2 #
```

无情的事情发生了，提示充值卡号错误，不过这里可以把前面的语句无效化处理，但是其实你可以看看语句完整是什么样的。

```
select * from volumes where username = '' union select 1,2 #
```

很好，这里语句select出来的是username为空的账户的充值卡号，所以你可以注册一个test用户，记录下充值卡号，然后构造用户名

```
test' union select 1,2 #
```

提交判断成功。