加群

扫码入群，群满加公开课小助手微信进群：fbgkk2017

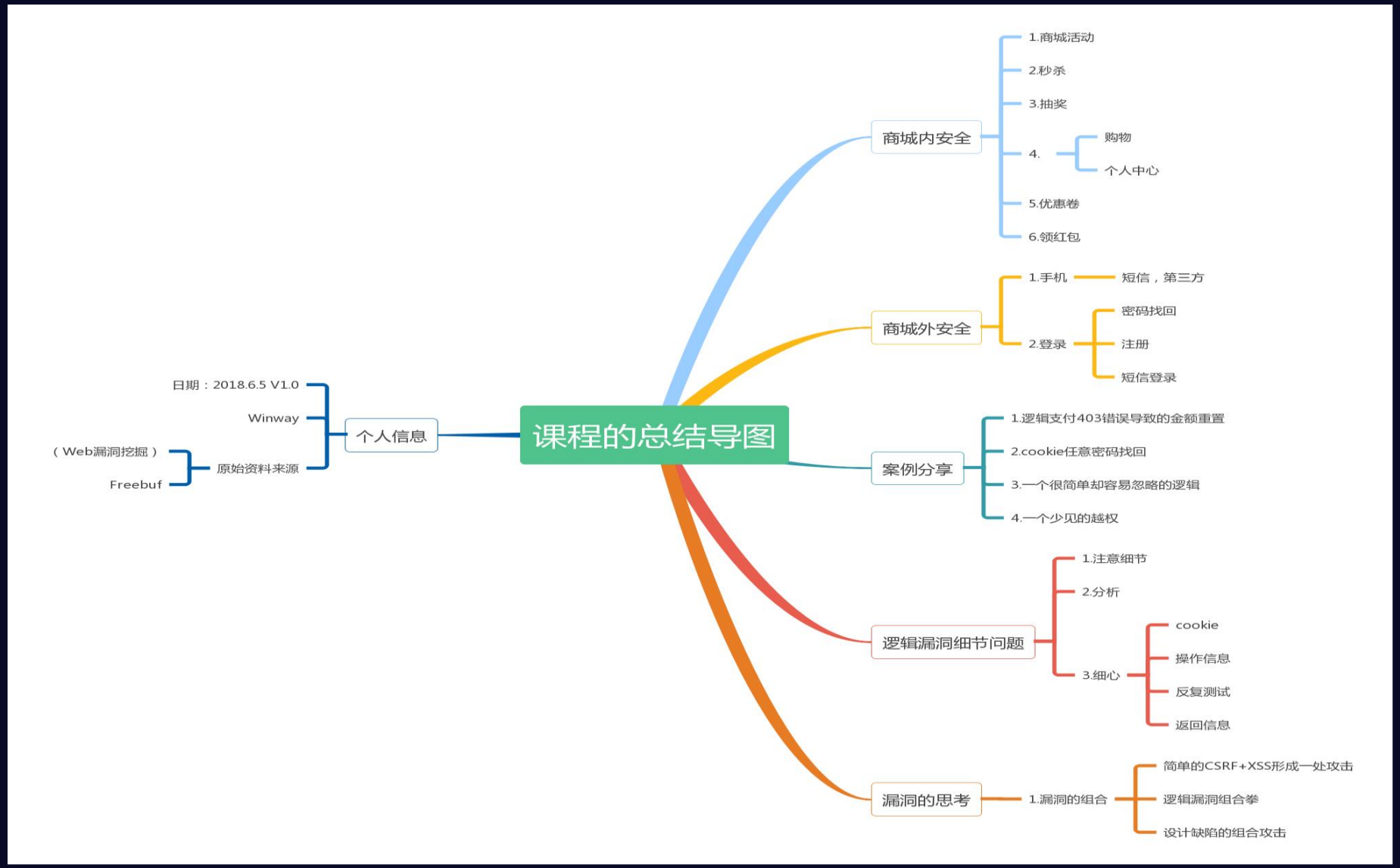
FREEBUF

Web漏洞挖掘"奇淫技巧"总结

美团年度Top1白帽 WINWAY

1 目录

2 总结过程引导图


课程的总结导图
个人信息
日期：2018.6.5 V1.0
Winway
原始资料来源
（Web漏洞挖掘）
Freebuf
商城内安全
1.商城活动
2.秒杀
3.抽奖
4.
购物
个人中心
5.优惠卷
6.领红包
商城外安全
1.手机
短信，第三方
2.登录
密码找回
注册
短信登录
案例分享
1.逻辑支付403错误导致的金额重置
2.cookie任意密码找回
3.一个很简单却容易忽略的逻辑
4.一个少见的越权
逻辑漏洞细节问题
1.注意细节
2.分析
3.细心
cookie
操作信息
反复测试
返回信息
漏洞的思考
1.漏洞的组合
简单的CSRF+XSS形成一处攻击
逻辑漏洞组合拳
设计缺陷的组合攻击

3.1 商城网站

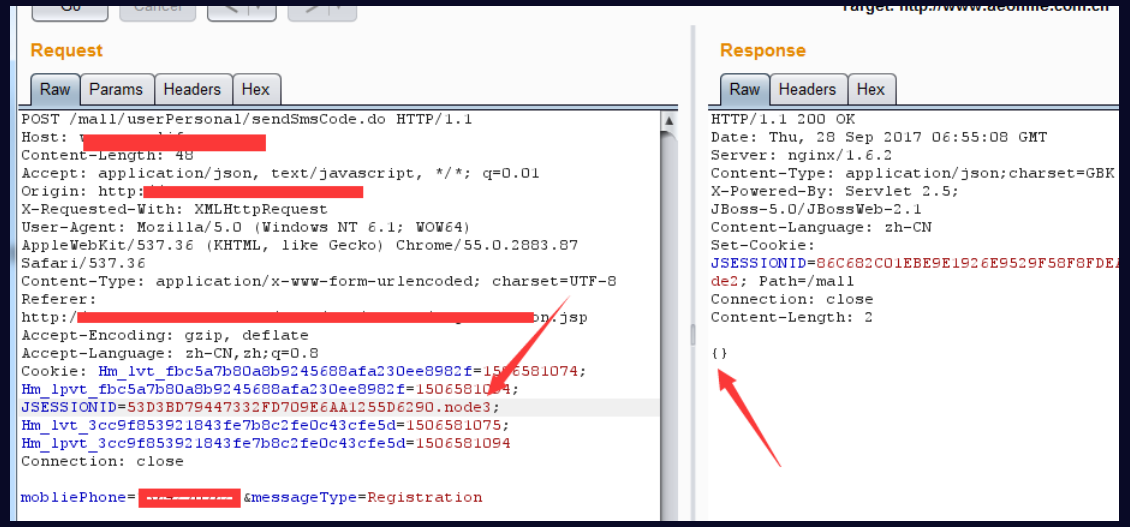
登录，注册，找回密码，第三方登录

3.2 注册

短信轰炸
任意用户注册
绕过验证
批量注册
手机号枚举遍历

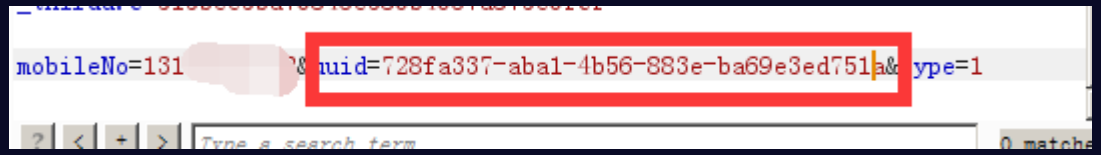
3.2.1 短信轰炸

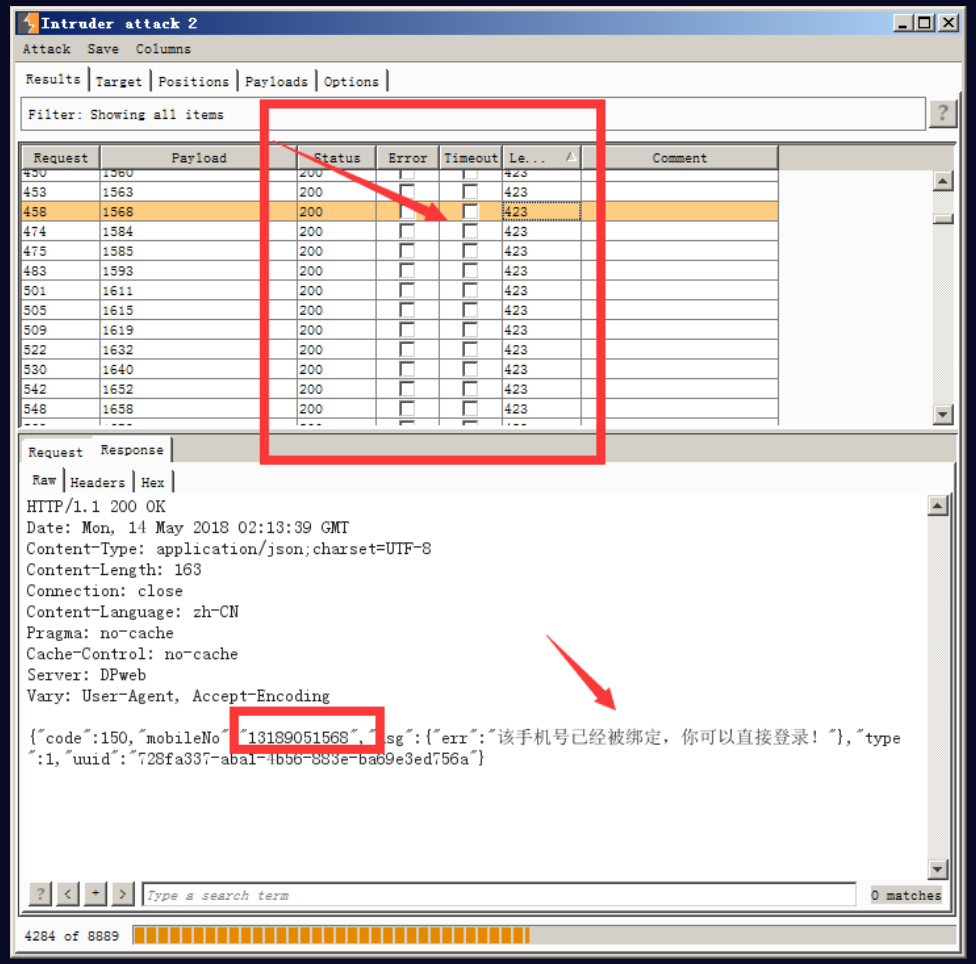
1.参数修改短信轰炸
2.Cookie短信轰炸
3.根据业务的判断



3.2.2 手机号遍历枚举

1.绕过风控继续遍历
2.简单遍历接口





3.2.3 绕过验证

1.删除参数
2.修改参数将验证码修改成0000
3.构造验证码

1.删除验证码参数导致验证码被绕过
2.修改参数中的验证码导致逻辑错误绕过
3.构造验证码key值原理

3.2.3 任意用户注册

1.验证码爆破
2.绕过验证码
3.返回包中的验证码

1.使用burpsiute 对验证码没有任何频繁拦截时简单的爆破出4-6位数的验证码

2.验证码绕过，修改验证码参数，删除参数

3.返回包直接返回出验证码

3.2.3 批量注册用户

1.注册时数据包中存在参数userid 代表一个新用户的诞生，将userid：123456 中对应的值遍历批量的注册小号

4 密码找回

短信验证码
邮箱验证码
找回链接url参数
修改参数
绕过找回密码步骤

4.1 短信验证码以及邮箱验证码

1.删除参数
2.修改参数
3爆破验证码

1.简单的删除掉验证码的参数导致错误而被绕过，其次很少这种案例

2.修改参数，改为空格等等 而被绕

3.爆破验证码，对于验证码的4-6位数进行简单暴力破解

4.2 url参数修改以及绕过步骤找回密码

1.链接中的参数修改
2.绕过步骤

1.链接中存在key=xxxx 以及code=xxxx 分为有加密跟无加密的有些直接显示用户名修改直接就重置，有些显示为加密的值将值尝试是否可以组成而达成密码重置

2.url中可能会存在每一个步骤的过程比如 find1 find2 find3 find4 等等，有些步骤可能会出现在post数据里面绕过，比如在find1的时候直接将find1 改为find4 直接到最后一步修改密码达成了密码重置

4.3 修改参数

1.在对应业务中存在的每一个可疑参数进行修改，对比，比如
post提交的参数值 get的参数值
Cookie的参数值 对应而感到可疑的参数进行修改加以测试

5 登录-第三方登录

1.任意用户登录

2.登录凭证劫持

3.短信验证码

1.修改参数 userid 或者其他可疑的参数登录了其他用户的账号

2.登录凭证的劫持，可能在某些app中把用户的token丢在了url中

如果这条链接被利用跟第三方缺陷可能导致用户的token被劫持

3.短信验证码

6 业务层逻辑

个人中心

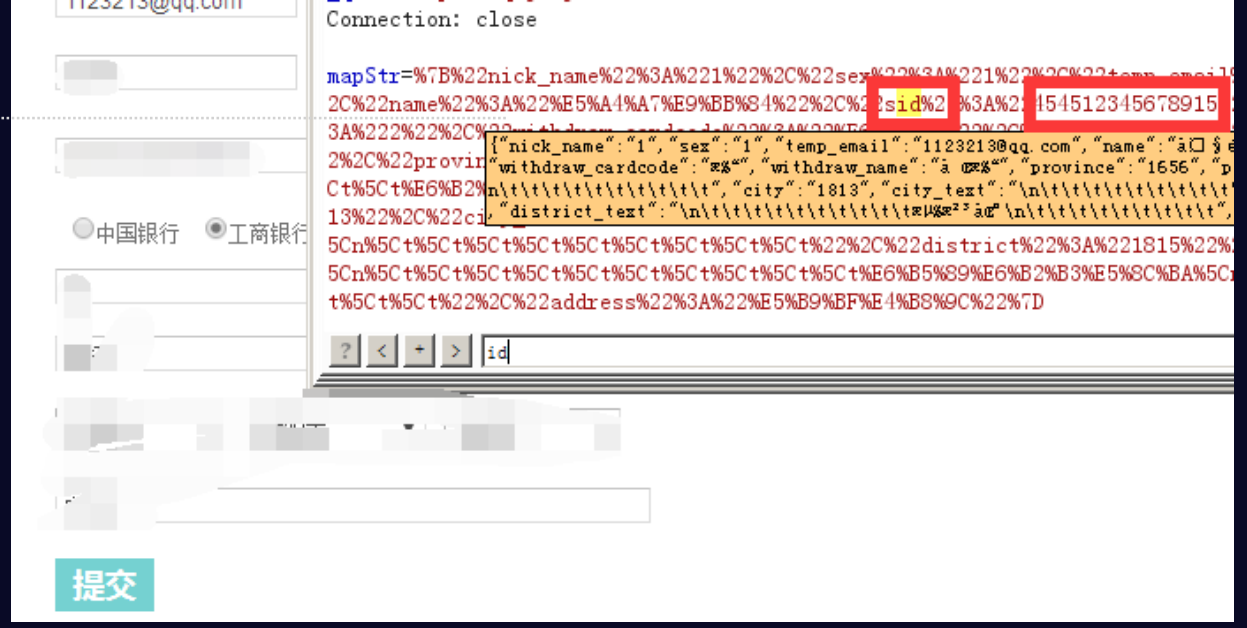
购买产品

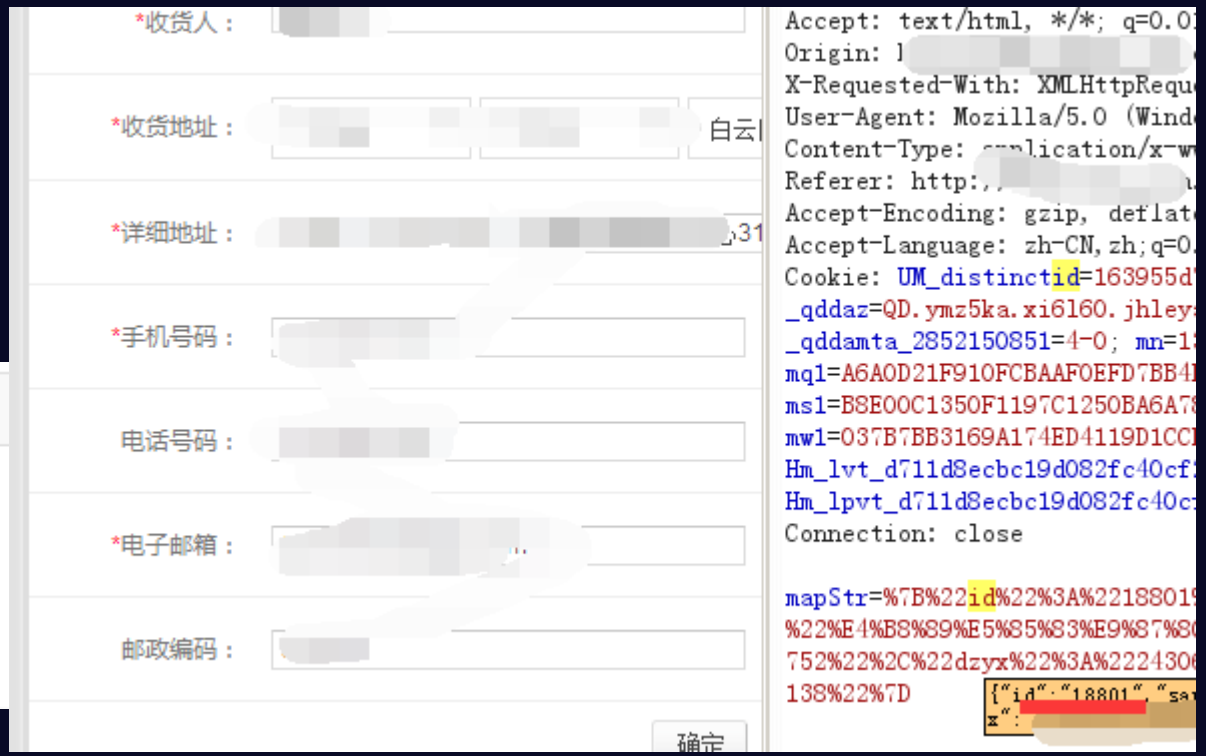
商城活动

6.1 个人中心

url中的参数越权

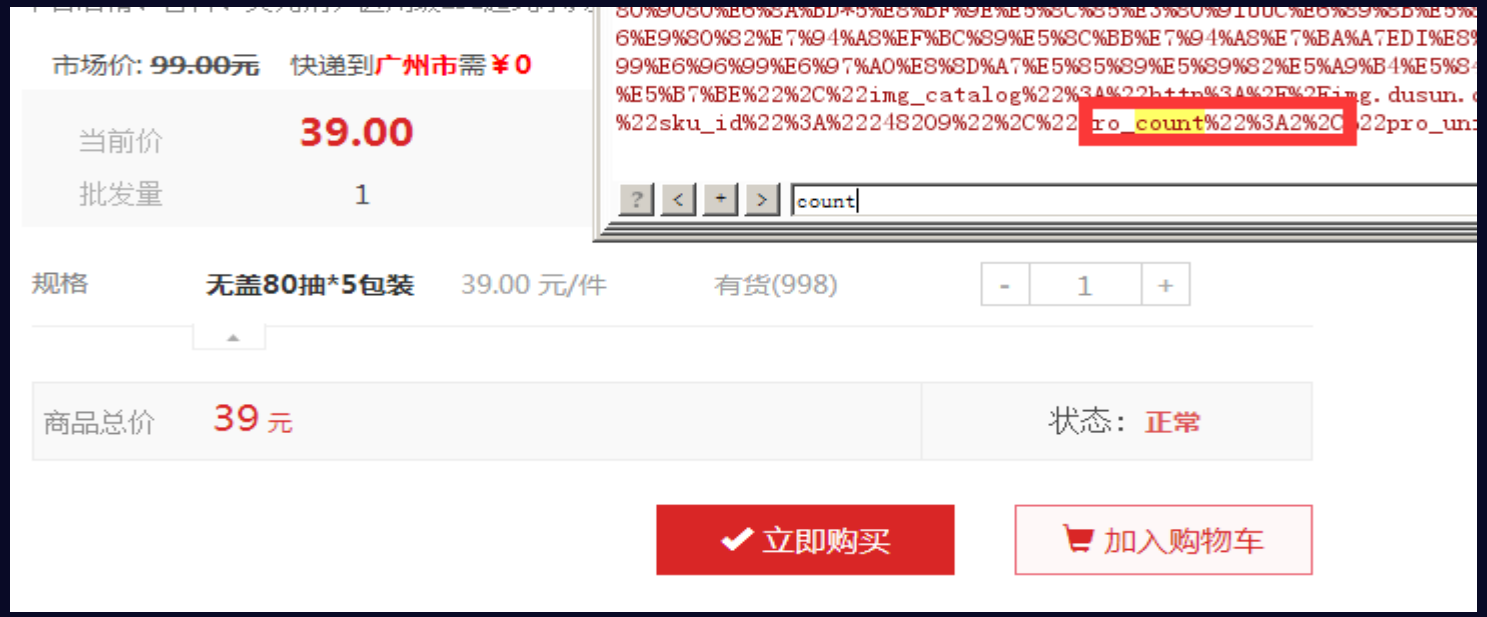
Post数据包中的id修改越权





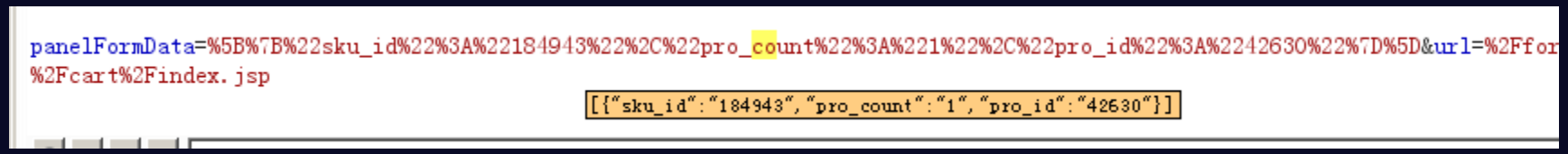
6.2 购物与活动功能

找到喜欢的商品，添加购车，提交订单，支付订单，基本的流程是这样的，但是在这三个物过程中可能都会存在某个缺陷根据业务的设计来判断，其次可以修改产品的数量，产品的时间，优惠卷，产品的id，sku值，等等



6.3 支付漏洞

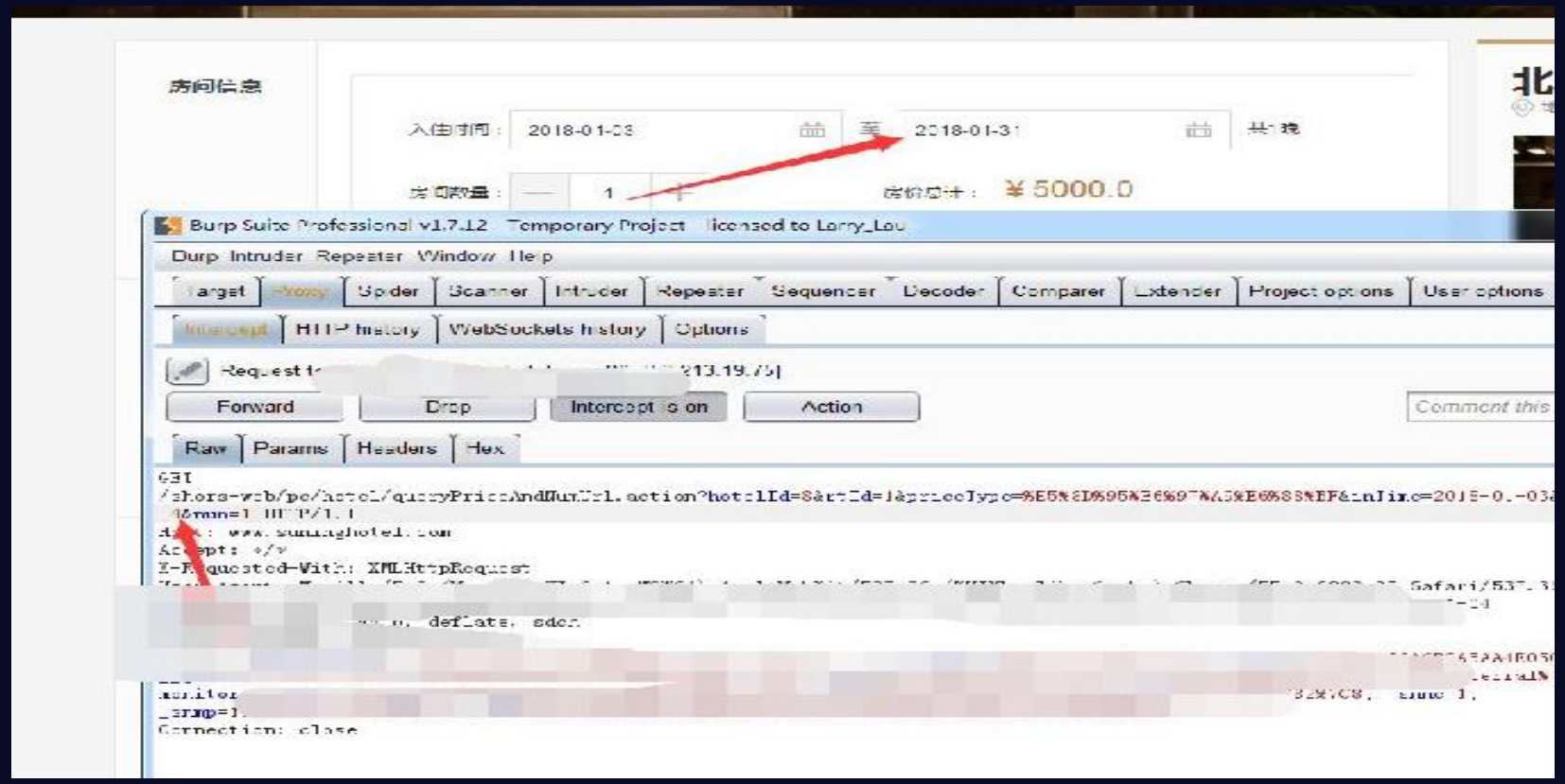
无论哪一种支付的方式，积分支付



无论是怎么样的一种支付过程都是要非常细心的分析每一个参数来组成一个逻辑支付

6.4 支付漏洞 某src案例

找到一个酒店，在支付的时候选择，进去里面选择时间，在选择时间的时候抓包，将数据包的日期修改，选择入住时间 1.3-1.30 一天是5000 x 30 是150000 元 但是在数据中的时间改为 1.3-1.4 金额变成 5000 而日期是1.3-1.30 支付成功



6.5 支付漏洞

修改金额


1
贝贝 春秋款儿童帆布鞋高帮童鞋男女童学生单鞋|尺码:36码|颜色:深蓝色
- /双
1
-
商品应付总价: 0 元
订单总额: 元
快递
物流
EMS
当前运费: 0.0 元
买家留言
0 件商品
总商品金额： 0元
优惠券总额： 0元
(如对运费有异议，可在订单提交后联系客服修改运费，按实际运费进行订单支付)
运费： 0元
使用红包抵扣： 1.0元
使用余额抵扣： 0.01元
应付总额： 0.01元
提 交

7 商城活动

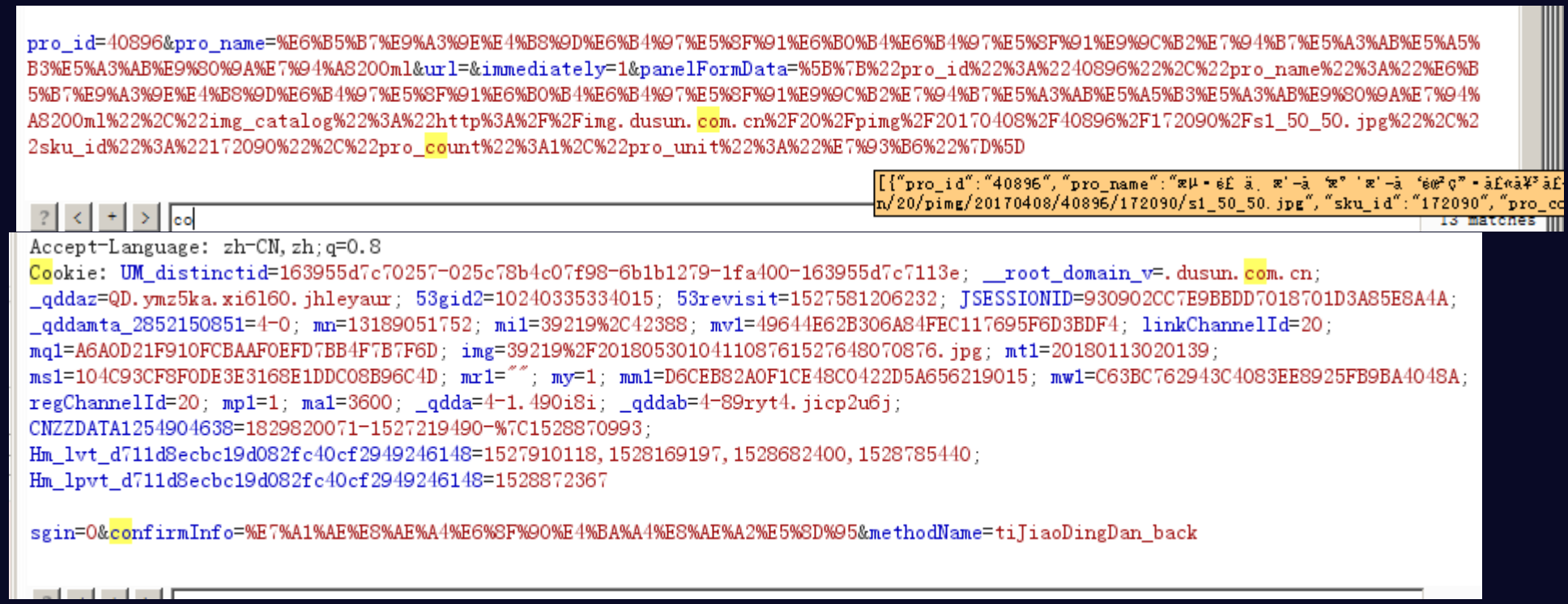
抽奖
领取优惠卷

商城里面基本都有一个轮盘抽奖 中奖率非常的底可能存在着漏洞而修改后可以抽中想要的产品

7.1 刷优惠卷

一些领取优惠卷，送优惠卷的渠道，一个账户领取多张优惠卷

8 细心+细节



1细心.Post参数 get参数 cookie 参数 js参数
对这些参数都需要细心的测试，重复组合测试
2细节.在测试的过程中对每一个可能存在漏洞的按钮进行测试，重复利用，
相同的功能可能一个存在着漏洞一个不存在这都是可能有这些问题。

9 案例分享

1.逻辑支付

2.垂直 越权

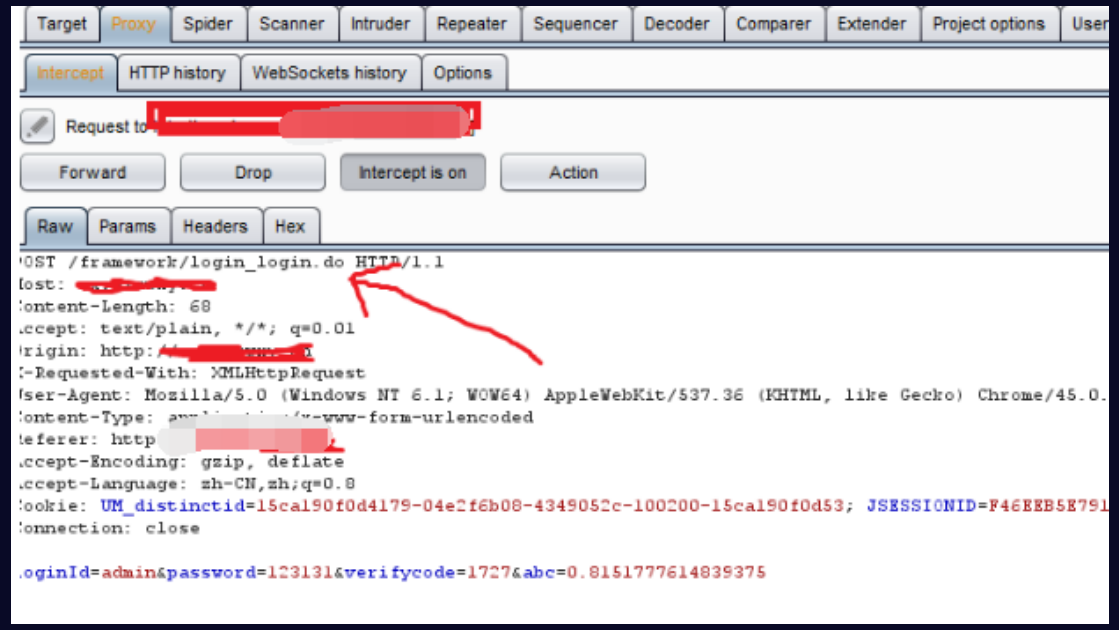
3.密码重置

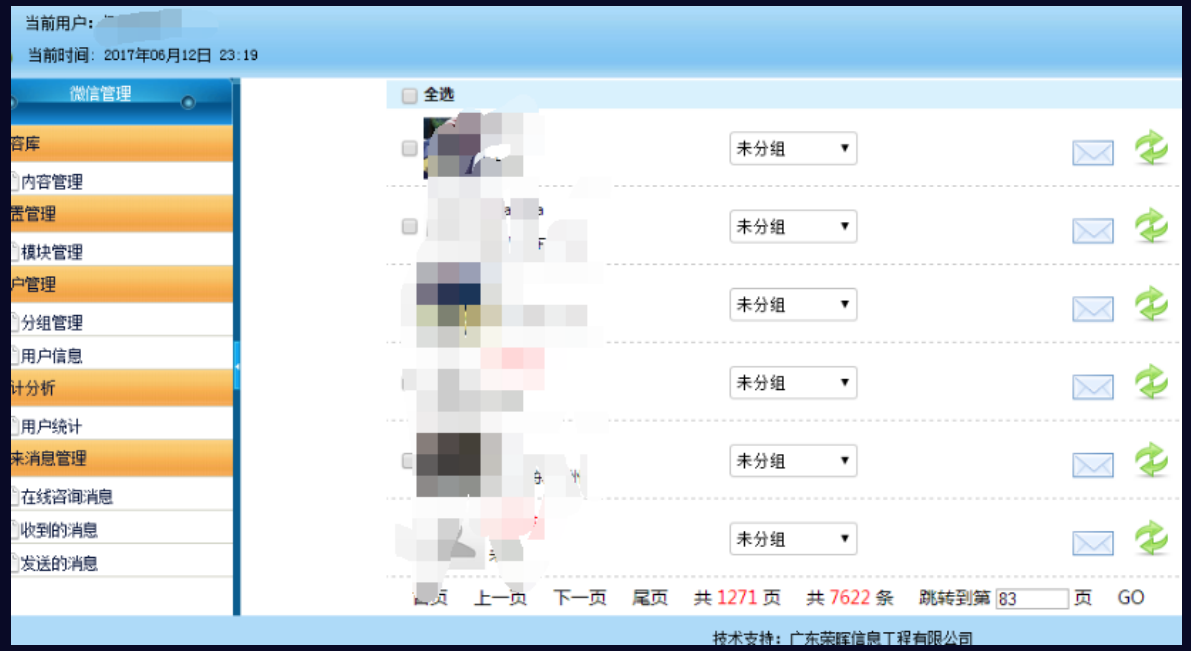
9.1 （案例分享）逻辑支付

第一步，打开xx你会发现有一个骑手商城，点击进去，可以看到有些很贵的产品，我是用APP抓包的，所以截图是分开的，点击一个最贵的产品，然后在点击支付，支付的时候抓包，将包保存下来，丢掉第二步，找到最便宜的产品，点击购买，将刚才那个数据包，全部内容，记住是全部数据内容，替换过去，ok，会跳转到提交订单处，点击一下提交会系统错误，重新支付，但是你多点几次，ok订单价格替换掉，支付价格也替换了

9.2 （案例分享）越权

一个奇葩的越权

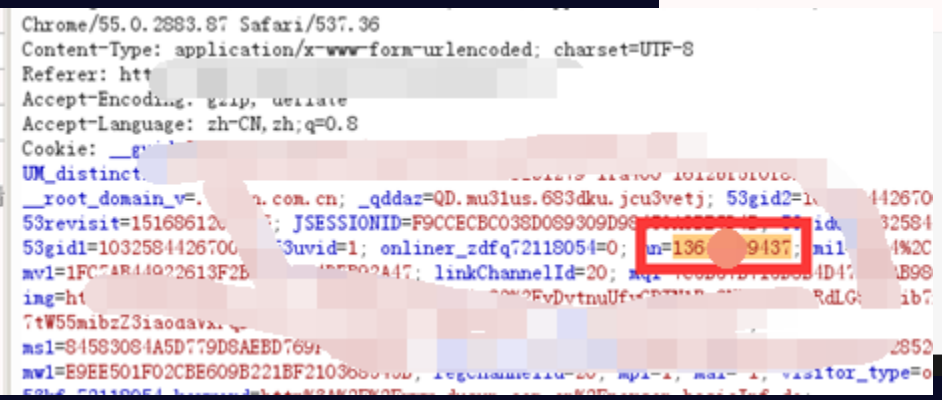


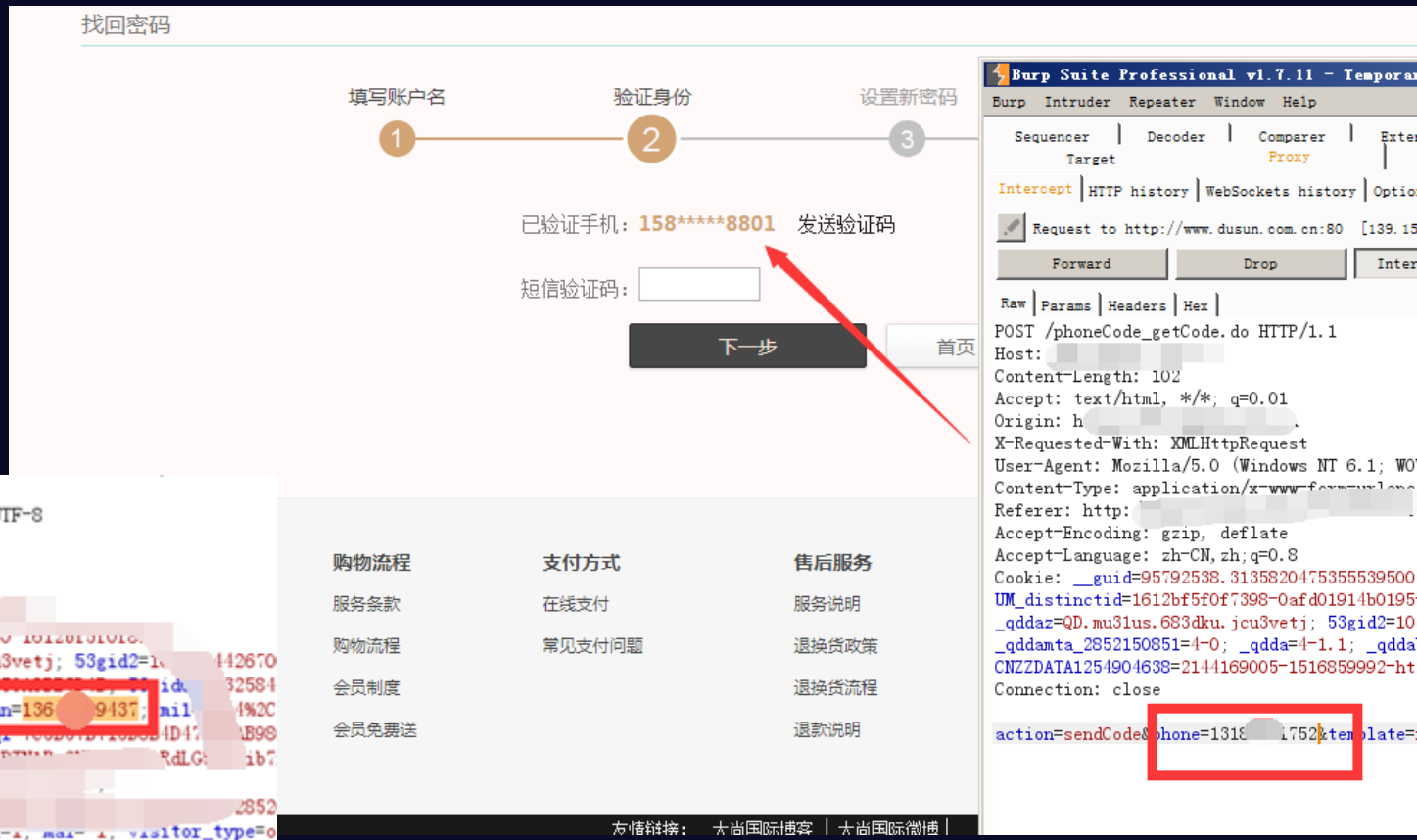


9.3 （案例分享 密码重置）


找回密码
填写账户名
验证身份
设置新密码
完成
1
2
3
4
已验证手机：131*****1752
发送验证码
短信验证码：
下一步
首页


新 密 码：
确认密码：
验 证 码：
882969
882969看不清
下一步


Chrome/55.0.2883.87 Safari/537.36
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Accept-Language: zh-CN,zh;q=0.8
Cookie:


找回密码
填写账户名
验证身份
设置新密码
1
2
3
已验证手机：158*****8801
发送验证码
短信验证码：
下一步
首页
购物流程
服务条款
购物流程
会员制度
会员免费送
支付方式
在线支付
常见支付问题
售后服务
服务说明
退换货政策
退换货流程
退款说明
友情链接：
大尚国际博客
大尚国际微博

Burp Suite Professional v1.7.11 - Temporar
Burp Intruder Repeater Window Help
Sequencer Decoder Comparer
Target Proxy
Intercept HTTP history WebSockets history
Request to http://www.dusun.com.cn:80
Forward Drop
Raw Params Headers Hex
POST /phoneCode_getCode.do HTTP/1.1
Host:
Content-Length: 102
Accept: text/html, */*; q=0.01
Origin:
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 6.1;
Content-Type: application/x-www-form-urlencoded
Referer:
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.8
Cookie:
Connection: close
action=sendCode&phone=1318 1752

9.3 （案例分享 密码重置）

输入一个正常要重置的手机号，然后在抓包将数据包中的重置手机号替换成自己的手机号来接收短信验证码，在获取验证码直接进入设置新密码，第三步

然后在填写一个新的密码之后抓包，mn 处有显示一个重置的手机号，直接填写 成要重置的手机号直接更换

这样会一次性重置掉了两个账号的密码，反正也不知道什么情况就是奇葩

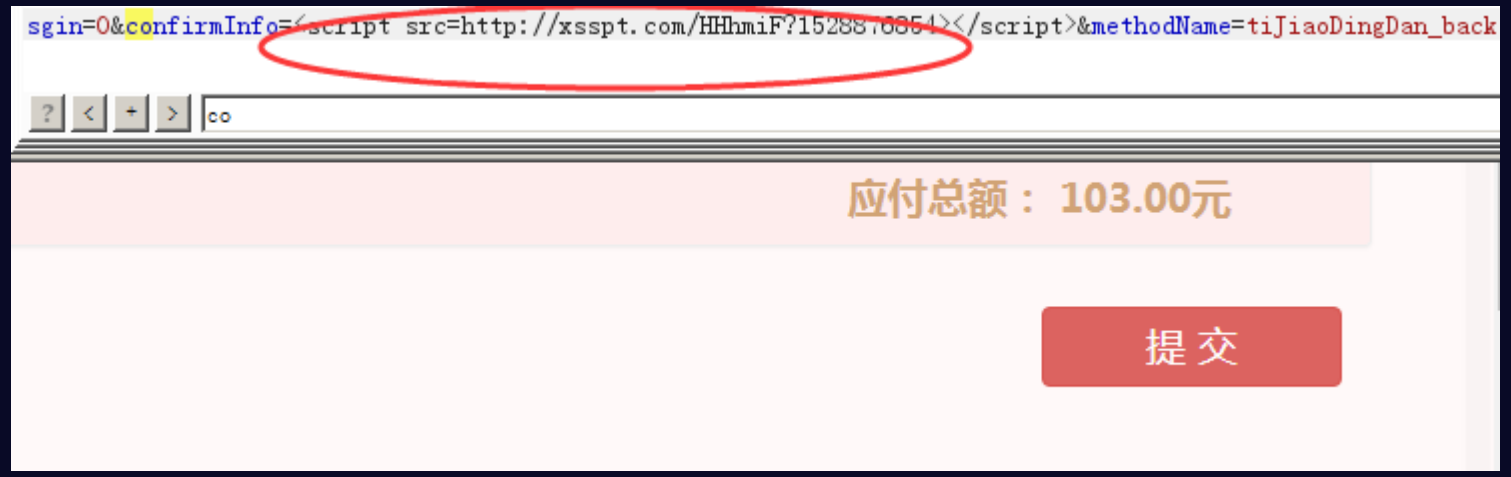
10 漏洞思考组合拳

1.Self-xss+Csrf组合拳

2.逻辑漏洞组合拳

3.设计缺陷组合拳

10.1 xss+csrf组合拳



在商城中我提交订单的同事会弹出一个信息框，这个信息框可以在提交的同时做出修改，将信息框要弹出的信息修改成一句JavaScript，存在着一个self-xss 那么这个xss并不是毫无意义

10.1 xss+csrf组合拳

简单的用burpsiute 生成一个poc ps(最好的情况是自己构造，不要用生成的，案例演示就简单化一点)

```
<html>
  <!-- CSRF PoC - generated by Burp Suite Professional -->
  <body>
    <form action="https:/[redacted]do" method="POST">
      <input type="hidden" name="sgin" value="0" />
      <input type="hidden" name="confirmInfo"
value="&amp;quot;&amp;quot;&amp;gt;&amp;lt;script&amp;gt;alert&amp;#40;&amp;quot;1&amp;quot;&amp;#41;&amp;lt;&amp;#47;script&amp;gt;" />
      <input type="hidden" name="methodName" value="tiJiaoDingDan&amp;#95;back" />
      <input type="submit" value="Submit request" />
    </form>
  </body>
</html>
```

10.1 xss+csrf组合拳

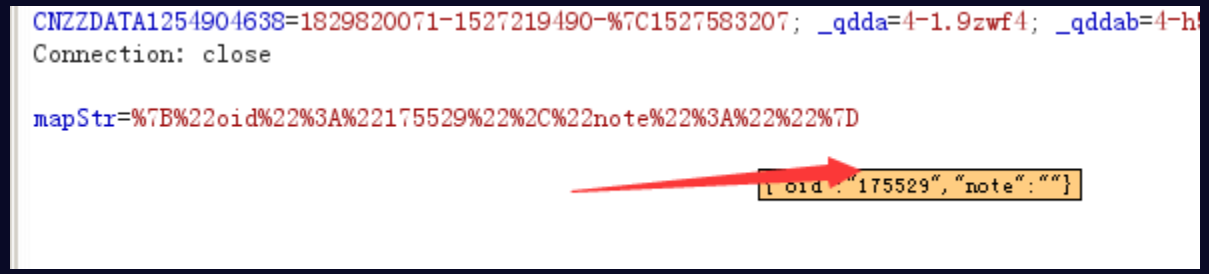
受害者打开连接后

10 逻辑组合拳

购买了一件最便宜的商品，在创建一个高价值的订单，在将订单退货抓取数据包，将数据包中的oid修改为高价值的产品订单，那么在订单中就会把一个还未支付的订单给退款了，很多时候漏洞需要去思考，当你挖掘 滴滴app的漏洞时你可能会选择去做一个滴滴司机才能挖掘到更多，挖掘美团外卖的时候你要成为一个骑手，一个商家 这样才能完整的对每一个业务的测试

10 逻辑组合拳


提示操作
是否申请退款？
原因：
取消
提交
支付状态
操作
等待卖家发货
已付款
订单查看
申请退款
已申请退订
待退款
已提交
未付款
付款
取消


CNZZDATA1254904638=1829820071-1527219490-%7C1527583207; _qdda=4-1.9zwf4; _qddab=4-h
Connection: close
mapStr=%7B%22oid%22%3A%22175529%22%2C%22note%22%3A%22%22%7D
{"oid":"175529","note":""}

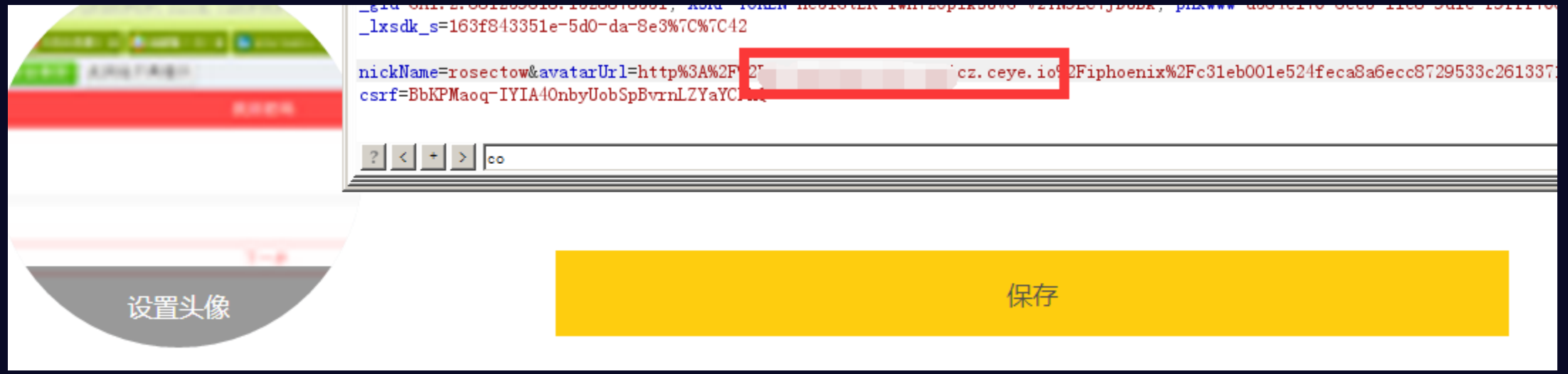

订单类型/订单号
订单商品
收货人
订单金额
---全部时间---
支付状态
操作
20180529044730591
¥0.01
在线支付
2018-05-29
16:47:30
等待卖家发货
已付款
订单查看
申请退款
20180529043352605
¥297.00
在线支付
2018-05-29
16:33:52
已申请退订
待退款
订单查看

10 逻辑组合拳

可能这个逻辑漏洞看起来很简单，但是其次的过程在于思考，思考与哪个部位组合起来达成一个高危害的攻击

11 设计缺陷组合拳

因为系统的设计缺陷，可以修改imgurl参数中的链接，但是需要绕过，链接中必须存在白名单url



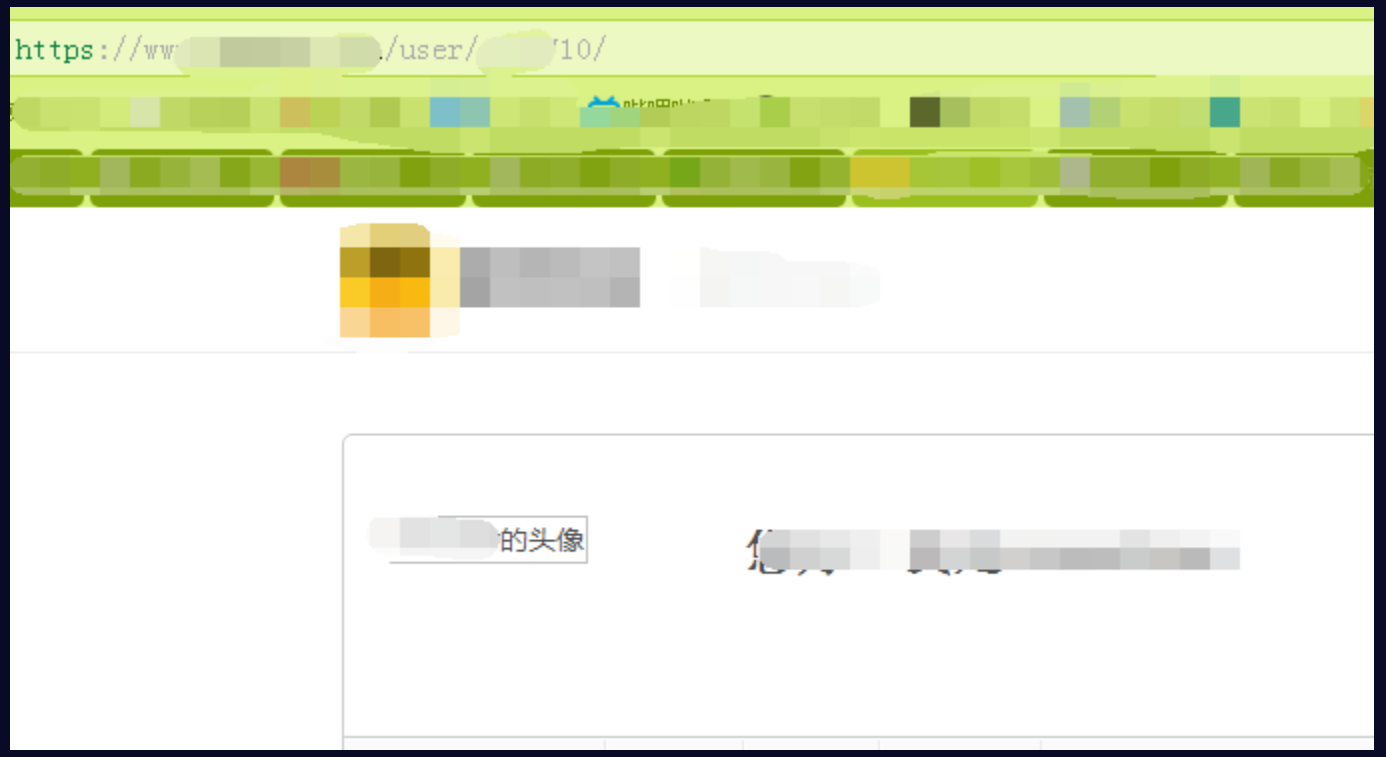
11 设计缺陷组合拳

保存后被访问的情况，以及记录

ID	Name	Remote Addr
127325	http://p0[redacted]e.io/iphoenix/5d6545bb3ac2a07a9e3d135b6388 7a75[redacted]	[redacted]
127324	http://p[redacted]t.vdqlcz.ceye.io/iphoenix/c31eb001e524feca8a6 ecc8729533c261[redacted]	[redacted]

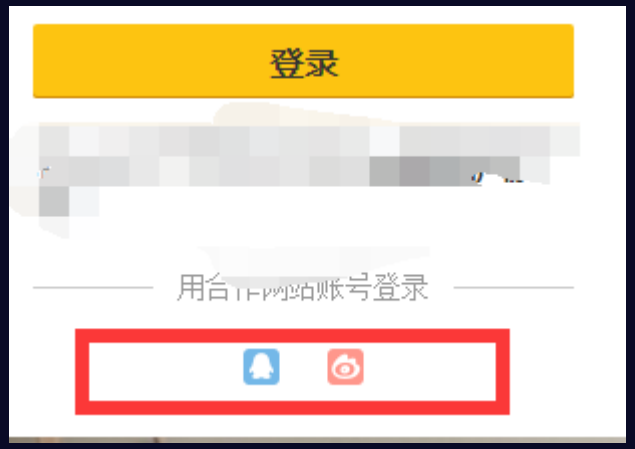
11 设计缺陷组合拳

第三方登录，存在缺陷的头像url https://wwwxxx.com/user/xxxx710/



11 设计缺陷组合拳

第三方登录



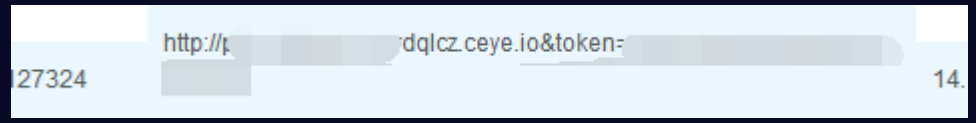
11 设计缺陷组合拳

修改uri地址，构造payload，组合拳，

11 设计缺陷组合拳

受害者访问构造的payload

登录了网站，获取到登录链接带token



完结！