1. **业务安全漏洞挖掘**
   1. **身份认证安全**
2. **暴力破解**

在没有验证码限制或者一次验证码可以多次使用的地方，使用已知用户对密码进行暴力破解或者用一个通用密码对用户进行暴力破解。 简单的验证码爆破。

案列：某药app暴力破解

前一段时间某药app在地铁站里做广告,于是就下了看了下,抓包一看请求全是明文的,感觉渗透有戏,于是,开始渗透

先从登陆界面开始:



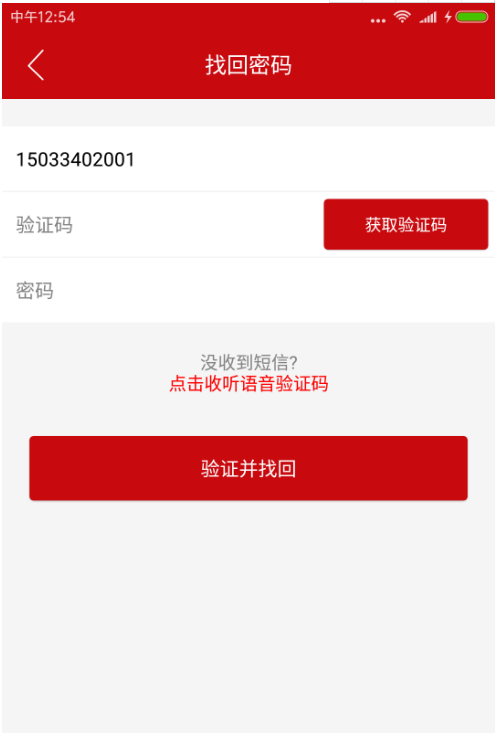
从登陆界面上看,使用手机号登陆的,随便输入几个手机号和密码,发现返回内容不一样,手机号没有注册过的,返回手机号不存在,手机号注册过的,返回用户名密码错误,这就好办了,自己构造个手机号字典去刷,如图：



ok,拿到手机号了,下面就开始**暴力破解**验证码了,用自己手机试了下,验证码是4位的,而且请求也是明文的,接下来**暴力破解**验证码,刚开始成功了,但是用这个验证码却登陆不上去,于是就关了那个爆破结果,后来我静下来想想,想到人家验证码可能是一次有效的,后来我又刷了一遍,但是可能被发现了,再刷结果返回302了,失败



于是我发现有个找回密码功能,也是手机验证码,填上自己自定义的密码,然后获取验证码,抓包



继续爆破,这次找到了,其实**暴力破解**登陆验证码的时候跟这个一样,只不过我没保存下来,如图：



用我修改的密码登陆,如图：



登陆成功

总结,这次爆破的跟之前不太一样,之前的验证码都是用过之后还有效的,这次验证码是一次性的,可能我经验太少了,大牛估计直接就想到了。

一些工具及脚本

Burpsuite

htpwdScan 撞库爆破必备 URL: <https://github.com/lijiejie/htpwdScan>

hydra 源码安装xhydra支持更多的协议去爆破 (可破WEB，其他协议不属于业务安全的范畴)

1. **Cookie&session**
2. 会话固定攻击

利用服务器的session不变机制，借他人之手获得认证和授权，冒充他人。

案列：新浪广东美食后台验证逻辑漏洞，直接登录后台，566764名用户资料暴露

网站源码可直接下载本地，分析源码可直接登陆后台。暴露所有用户个人资料，联系方式等，目测用户566764名。

用户资料暴露



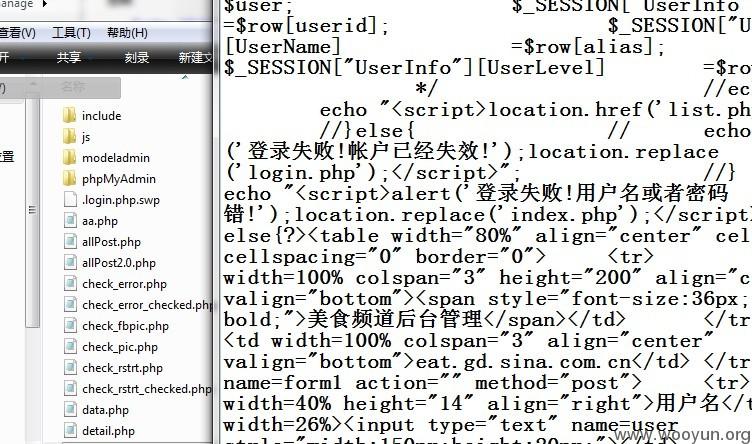
566764名用户资料



直接进入后台



源码直接下载



566764名用户资料直接跨后台浏览



修复方案：

漏洞较多，不知怎么说起，只列举一项phpsessid会话固定，通过利用此漏洞，攻击者可以进行会话固定攻击。在一个会话固定攻击，攻击者将用户的会话ID之前，用户甚至登录到目标服务器，从而消除了需要获得用户的会话ID之后。从php.ini设置session.use\_only\_cookies = 1。该选项使管理员能够使他们的用户不会受到攻击涉及在URL传递的会话ID，缺省值为0。

1. Cookie仿冒：

修改cookie中的某个参数可以登录其他用户。

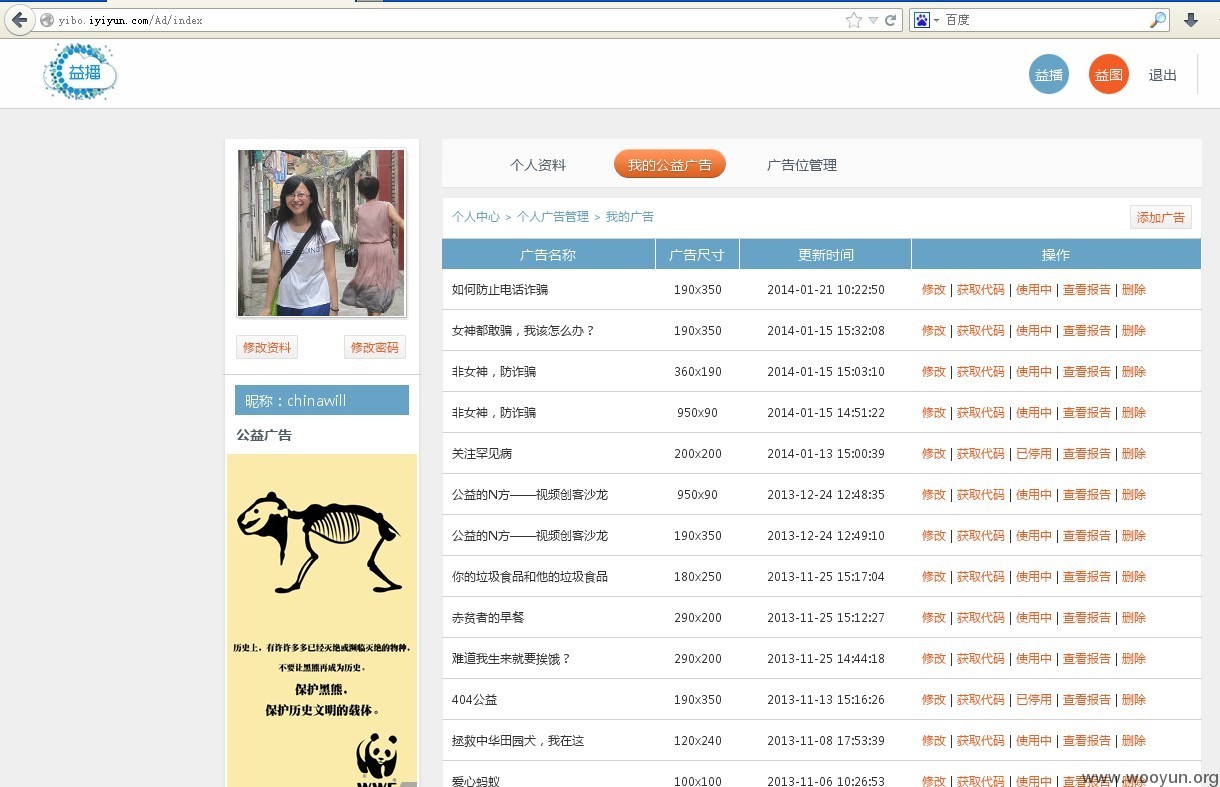
案例：益云广告平台任意帐号登录

只需要添加cookie

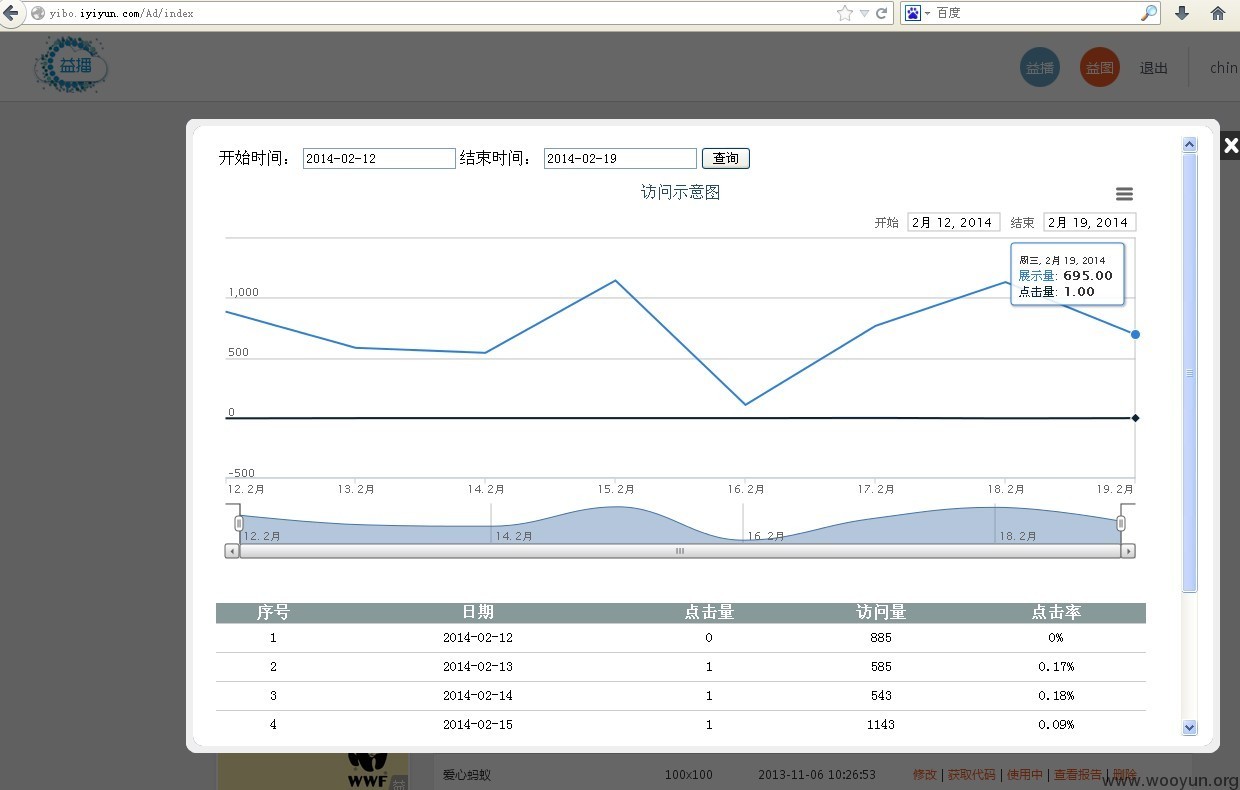
yibouid=数字 即可登录任意用户帐号！

通过遍历 找到一个官方管理的ID 291

登录



看看浏览 还是不错嘛



修复方案：

增强对cookie的验证机制！

1. **加密测试**

未使用https，是功能测试点，不好利用。

前端加密，用密文去后台校验，并利用smart decode可解。

* 1. **业务一致性安全**

1. **手机号篡改**

抓包修改手机号码参数为其他号码尝试，例如在办理查询页面，输入自己的号码然后抓包，修改手机号码参数为其他人号码，查看是否能查询其他人的业务。

1. **邮箱和用户名更改**

抓包修改用户或者邮箱参数为其他用户或者邮箱

案例：绿盟RSAS安全系统全版本通杀权限管理员绕过漏洞，包括最新 RSAS V5.0.13.2

RSAS默认的审计员

账号是：reporter，auditor

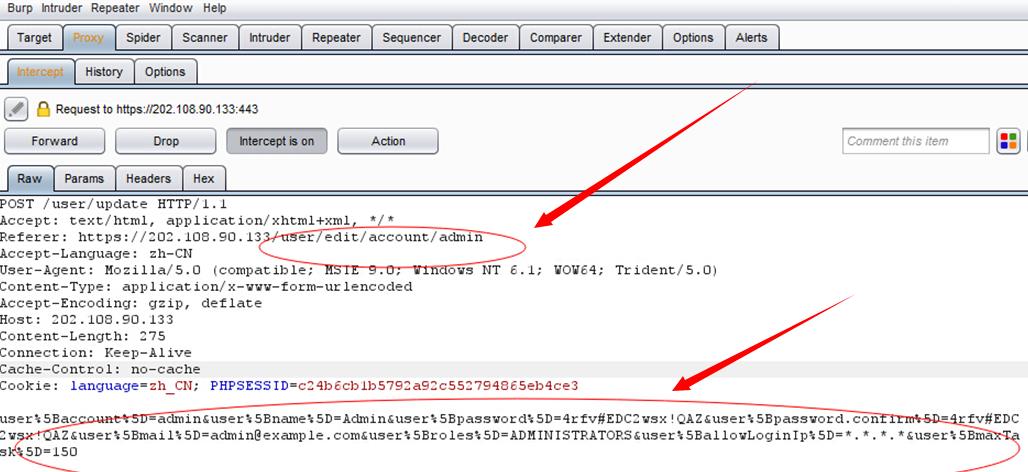
密码是：nsfocus

普通账户登陆后

查看版本，为最新V\*\*.\*\*.\*\*.\*\* 版本



然后修改审计员密码，抓包，将referer处的auditor和post的数据里面的auditor一律修改为admin，也就是管理员账号，2处修改完后的数据包如下图：



提交数据后，直接返回给我们超级管理员的密码修改页面，利用逻辑错误直接得到超级权限，如图：

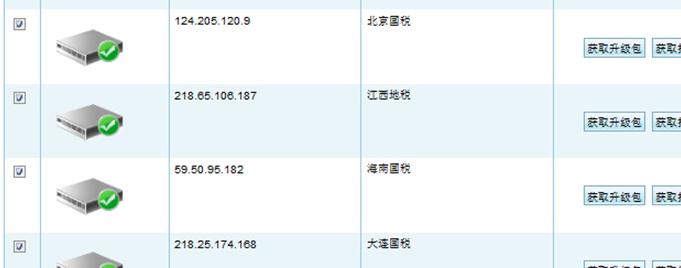


我们直接在这里修改admin的密码，然后提交即可：



超级管理员登陆





1. **订单ID更改**

查看自己的订单id，然后修改id（加减一）查看是否能查看其它订单信息。

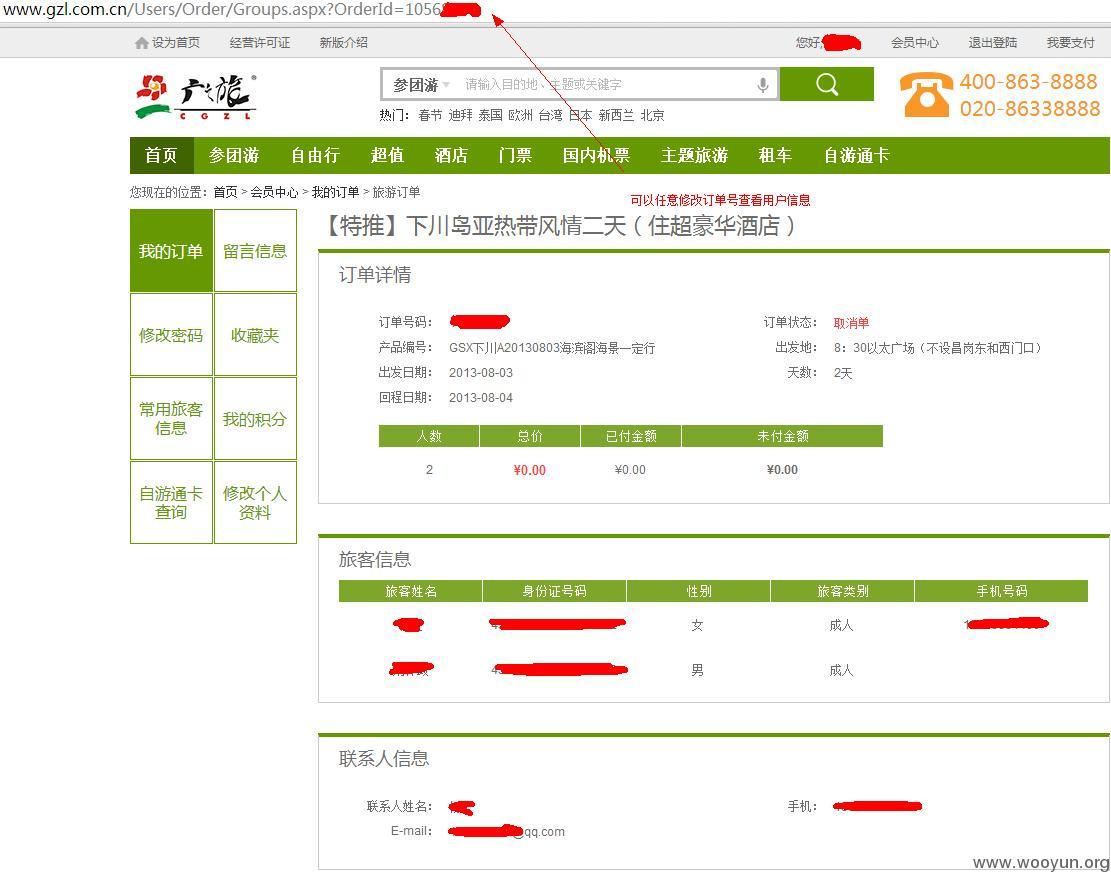
案例：广之旅旅行社任意访问用户订单

可以任意访问旅客的订单，泄露旅客的敏感信息！

用户登陆广之旅官方网站注册登陆

http://www.gzl.com.cn/

只要随便假订一张订单，在我的订单里面获得订单号，就能穷举其它订单信息



http://www.gzl.com.cn/Users/Order/Groups.aspx?OrderId=订单号

该页没过滤权限，相信还有更大的漏洞。

修复方案：

对

http://www.gzl.com.cn/Users/Order/Groups.aspx

订单页增加过滤

1. **商品编号更改**

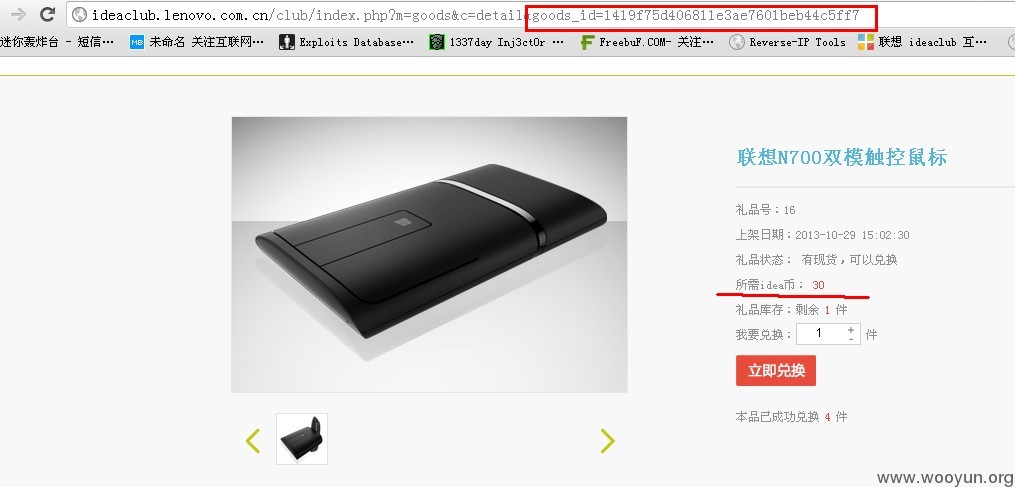
例如积分兑换处，100个积分只能换商品编号为001,1000个积分只能换商品编号005，在100积分换商品的时候抓包把换商品的编号修改为005，用低积分换区高积分商品。

案例：联想某积分商城支付漏洞再绕过

http://ideaclub.lenovo.com.cn/club/index.php?m=goods&c=lists

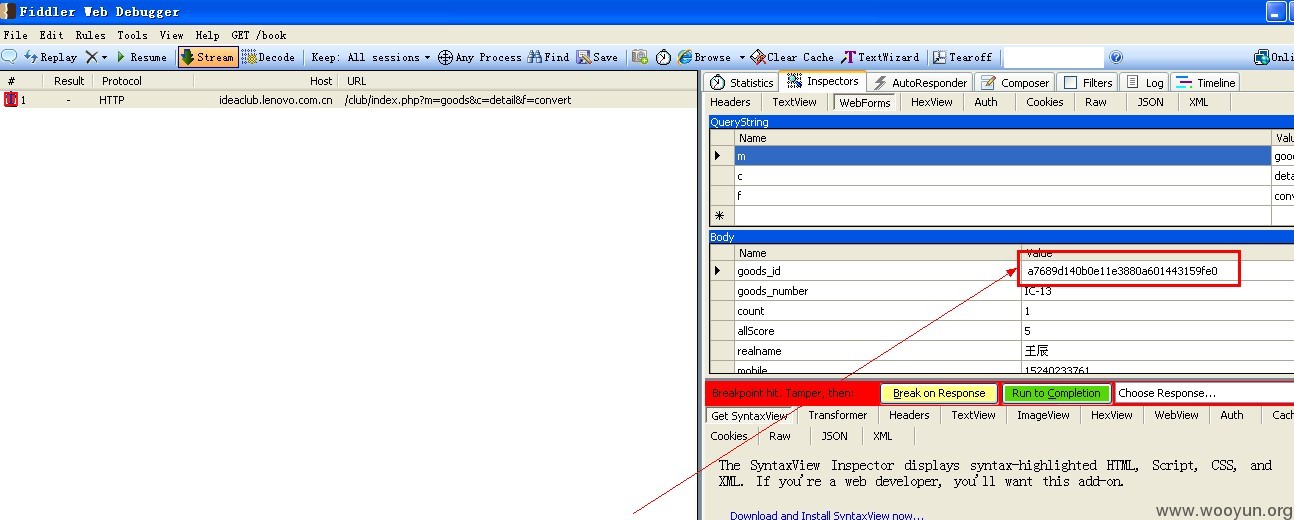
还是这个积分商城、 看我怎么用最低的积分换最高积分的礼物的~

1.我先挑选出我最喜欢的礼物，并复制下goods\_id=1419f75d406811e3ae7601beb44c5ff7



2.选择积分最低的礼物兑换（5积分的杯子），并填好相关信息，抓包修改goods\_id

替换为1419f75d406811e3ae7601beb44c5ff7



3.这里显示兑换成功，虽然显示的是被子兑换成功，但是兑换记录里，就不相同了





到这我们心仪的礼物要30积分，我只花5积分就兑换来了，是不是很划算？

修复方案：

积分参数还是放后台来操作，有goods\_id,在后台计算的时候取出并计算，这样会安全些。

1. **用户ID篡改**

抓包查看自己的用户id，然后修改id（加减1）查看是否能查看其它用户id信息。

案列：拉勾网百万简历泄漏风险(包括手机、邮件、应聘职位等信息、还可冒充企业身份筛选简历、发面试通知等)

注册一个企业账户，发布一个职位，然后看到有２个人投了简历：



打开简历，点转发：



抓包：



改简历id，话说这是get：



去邮箱看到有２封邮件：



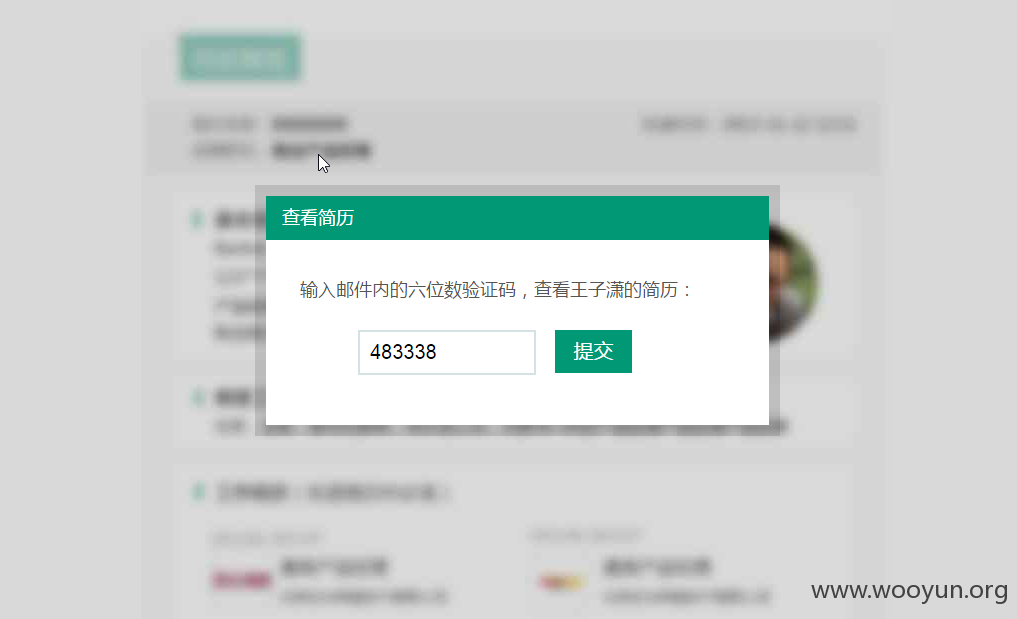
第一封是我发布职位里的没错：



第二封跪了：



看看能打开吗：





换个人：



查看联系方式：



手机，邮箱出来了：



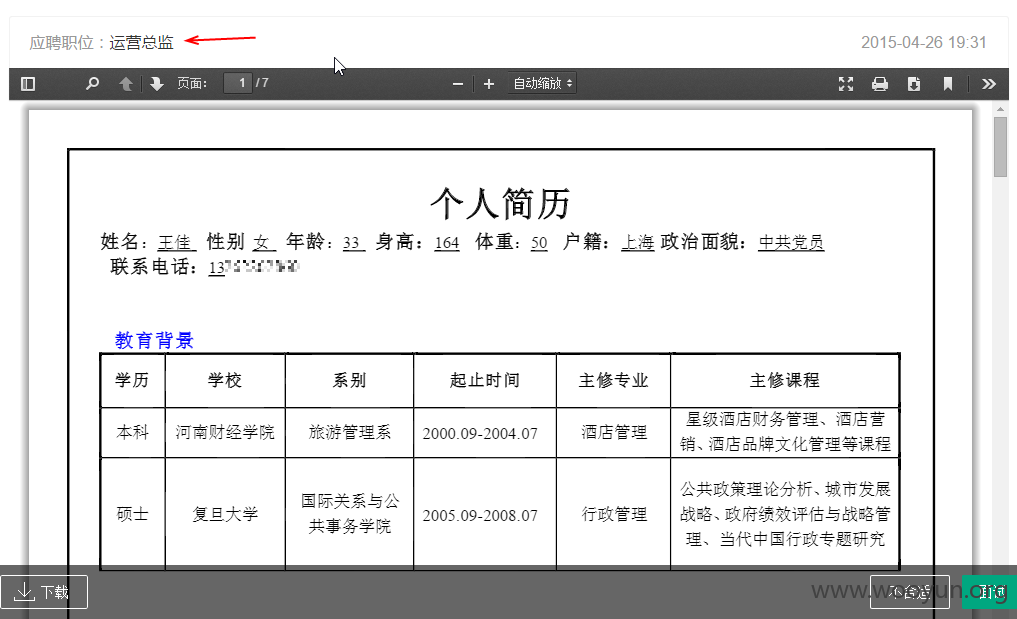
进一步操作，可以看到应聘职位：

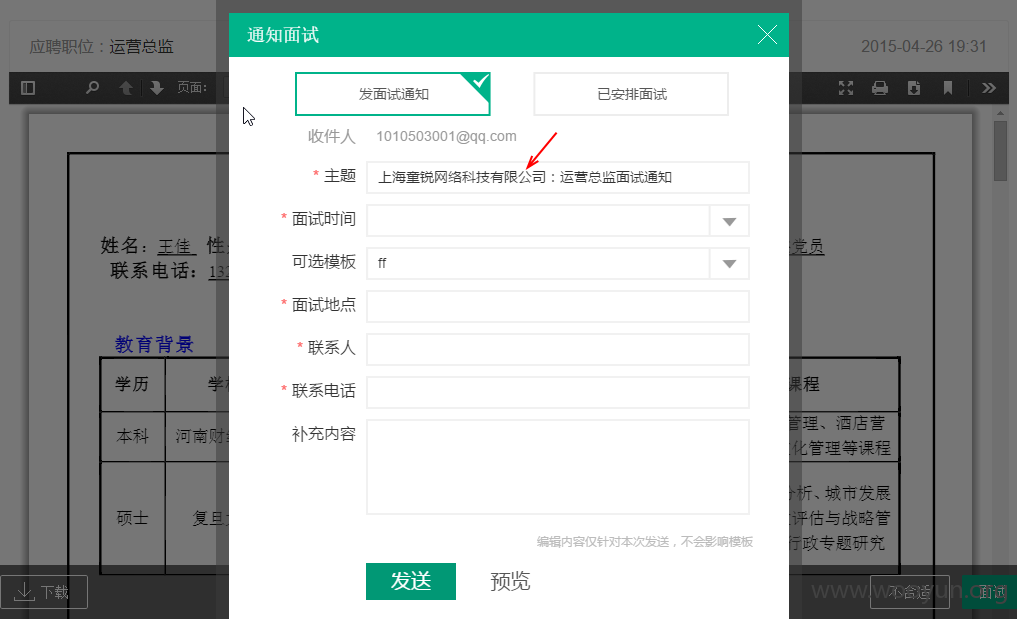


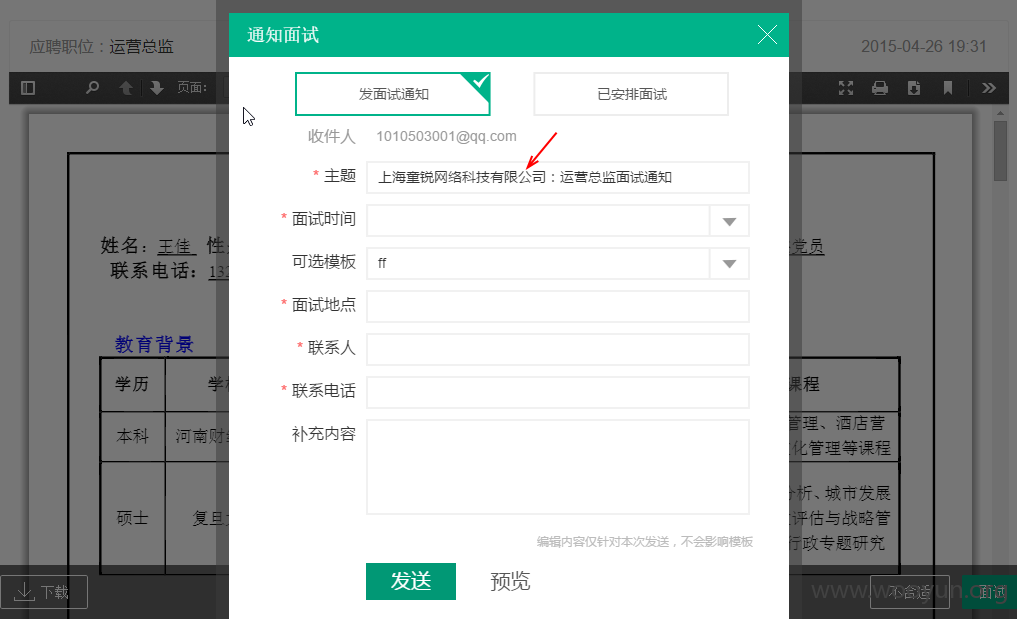
经过自己的账号测试，简历操作都会真实改动。

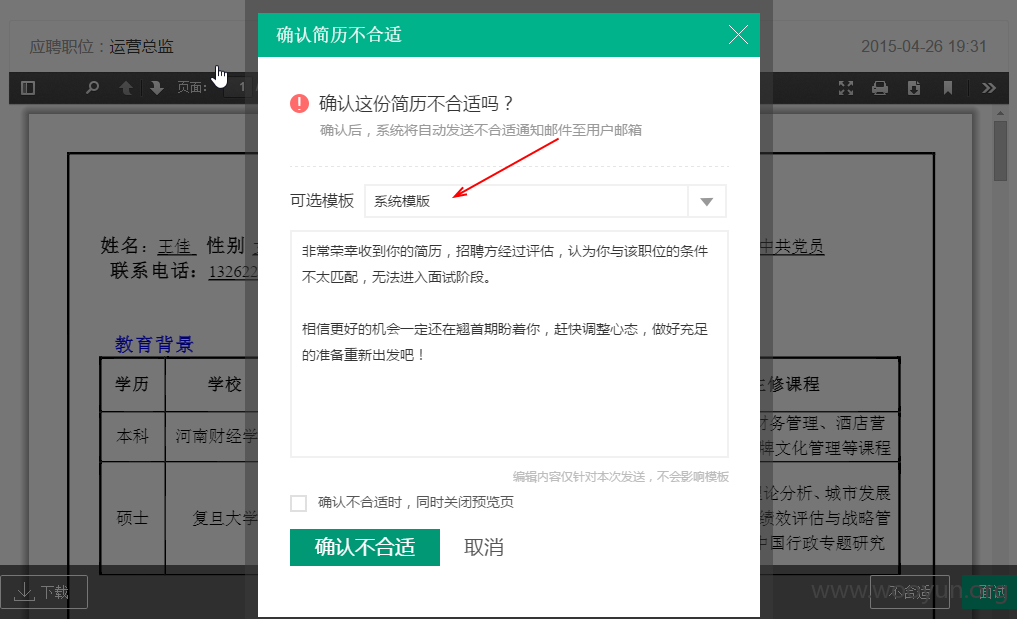
随机抽查：





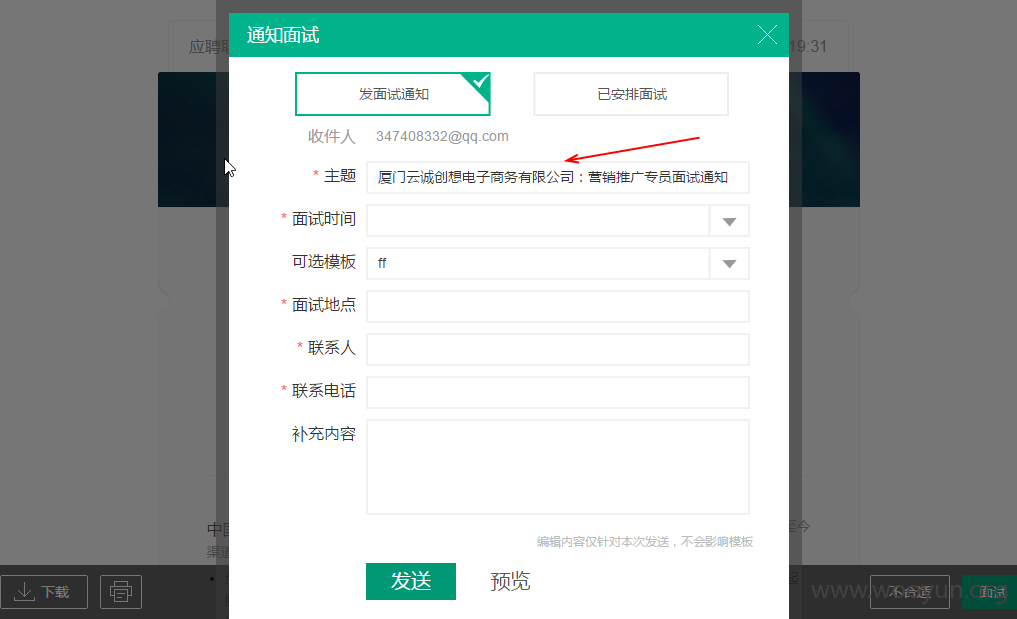
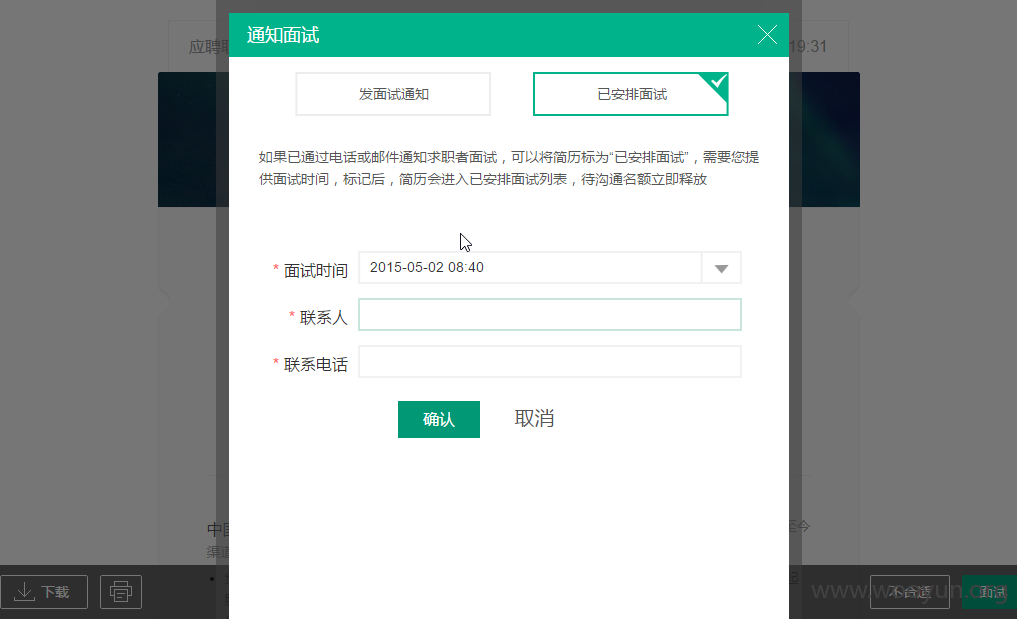










使用pydbg这种工具可瞬间建立离线简历库，还好我是良民- -，有没有信息贩子路过你们好好查查吧。

* 1. **业务数据篡改**

1. **金额数据篡改**

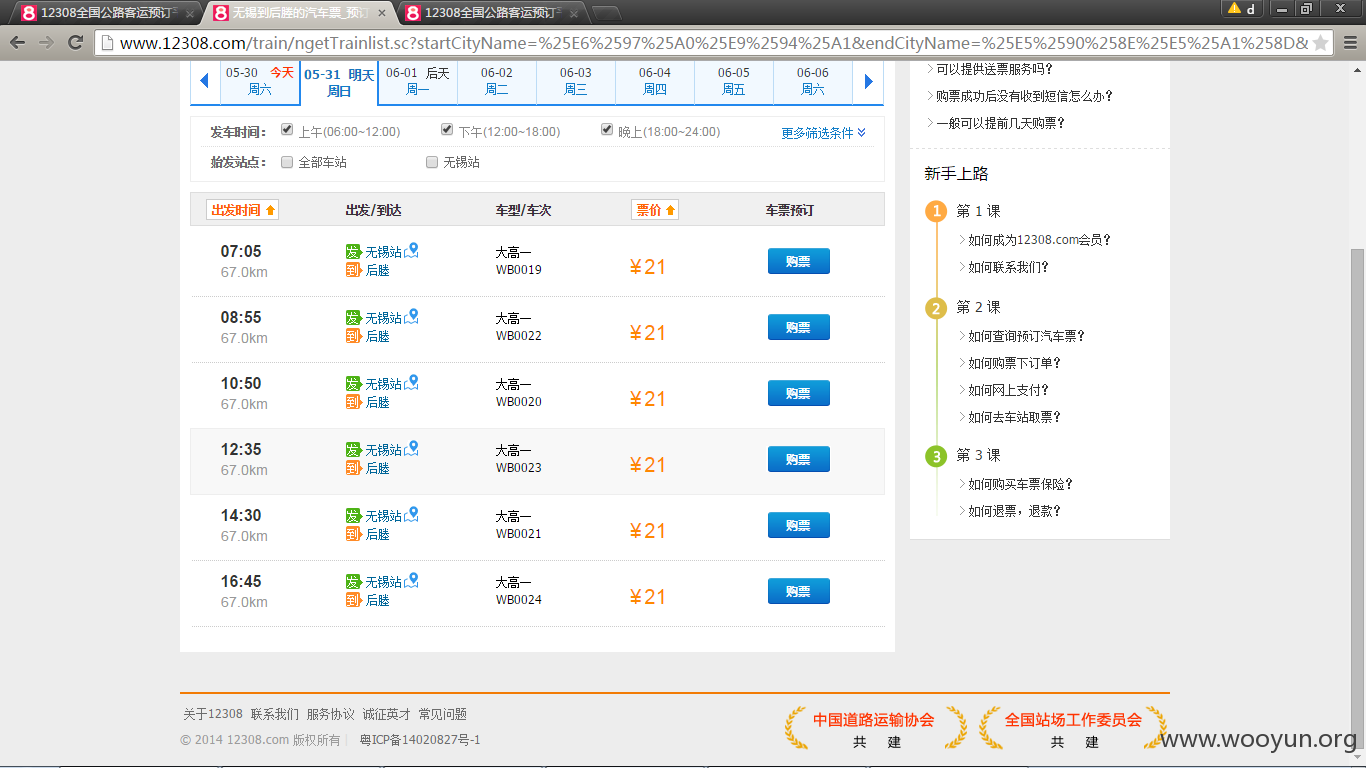
抓包修改金额等字段，例如在支付页面抓取请求中商品的金额字段，修改成任意数额的金额并提交，查看能否以修改后的金额数据完成业务流程。

案例：12308订单支付时的总价未验证漏洞(支付逻辑漏洞)

在支付时可修改订单总价

在支付时可以修改订单总价,为了避免不必要的麻烦,未完成支付的最后一步.但是跳转到支付宝价格已被修改。未完全确认，希望厂商自行测试。

1.下单







2.修改





修复方案：

加强验证

1. **商品数量篡改**

抓包修改商品数量等字段，将请求中的商品数量修改成任意数额，如负数并提交，查看能否以修改后的数量完成业务流程。

案例：蔚蓝团支付逻辑漏洞(可负数支付)

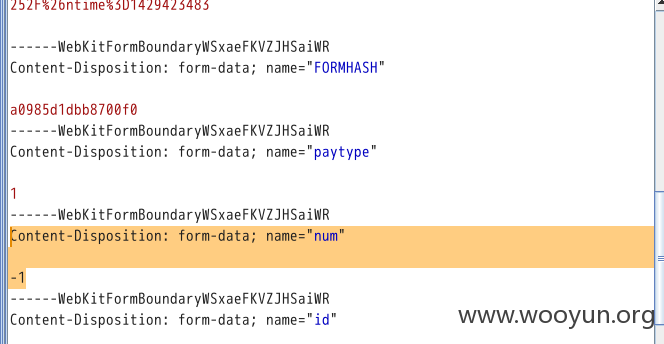
某团购网站通过修改数量为负，成功获取账户余额，该余额可成功支付其他订单！

蔚蓝团团购网站：http://tuan.wl.cn/

选好团购商品，然后点击最下面的支付宝支付：



抓包，并修改数量为-1：



网站里出现未支付订单：



直接点击付款，马上就跳转到显示付款成功的页面：



账户余额增加了：



使用该账户余额正常购买别的商品：



利用余额支付成功：



1. **最大数限制突破**

很多商品限制用户购买数量时，服务器仅在页面通过js脚本限制，未在服务器端校验用户提交的数量，通过抓包修改商品最大数限制，将请求中的商品数量改为大于最大数限制的值，查看能否以修改后的数量完成业务流程。

1. **本地js参数修改**

部分应用程序通过Javascript处理用户提交的请求，通过修改Javascript脚本，测试修改后的数据是否影响到用户。