浅谈关于二维码的钓鱼思路

原创先锋情报站 酒仙桥六号部队
2020-09-18原文

这是 **酒仙桥六号部队** 的第 **82** 篇文章。
全文共计1737个字，预计阅读时长7分钟。

前段时间看到了关于二维码劫持的几篇文，认真研究了一下，发现大家的观点不太一致的，今天和各位师傅分享一下，一起来认识一下二维码登录认证机制，看看二维码登录劫持到底是怎么回事，如何轮询劫持二维码进行钓鱼操作。（如有不足，欢迎补充）

登陆场景

看了某文后，经验证发现，市面上基本分为下面三种登录场景模式，在我理解下具体区别为关于登陆认证是否严谨。

- 扫描二维码登录pc系统

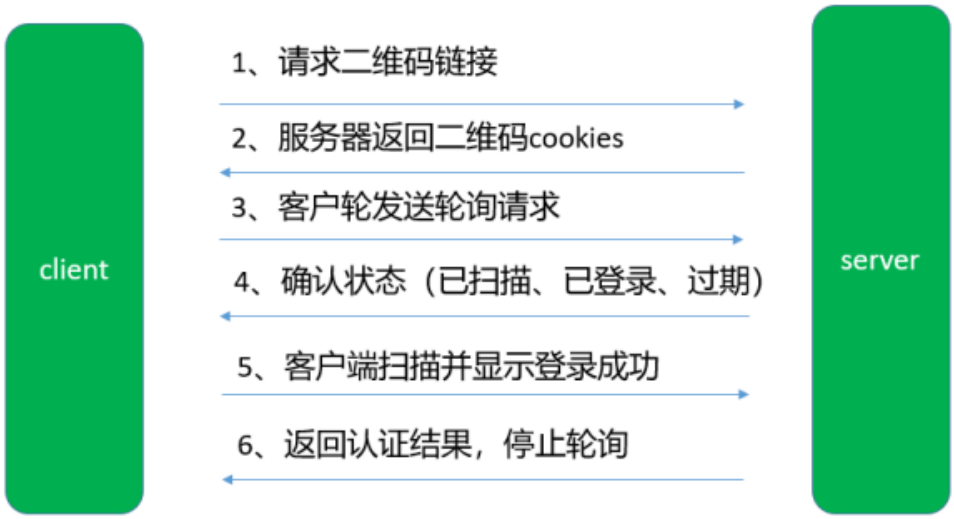
手机端已登录的前提下，扫描网页二维码，自动登录网页版，根据服务端自有认证体系与账户绑定登录，如微信app登录扫描登录网页版app，利用oauth体系，实现PC端自动登录，无需点击登录确认等操作。

- 二维码双因素认证

如微信公众号平台，在账户密码登录PC端的情况下，且手机端微信登录前提下，扫描二维码进行确认，登录网页版。

- Secure or login (sqrl)

直接使用扫描二维码登录，无需账户密码登录。登录步骤：



1. 打开pc端获取二维码图像，请求服务端登录，服务端生成二维码，并生成pc端唯一标识，比如sessionid，uuid等。
2. pc端开始轮询，获取二维码后，为保证二维码保持有效状态，持续请求。状态为:new,scaned,confirmed,refused,expired
   注意：轮询是为了保证其有效性，不断发送请求二维码，根据返

回状态判断是否能用，说白了就是保证二维码可以扫描，不失效。

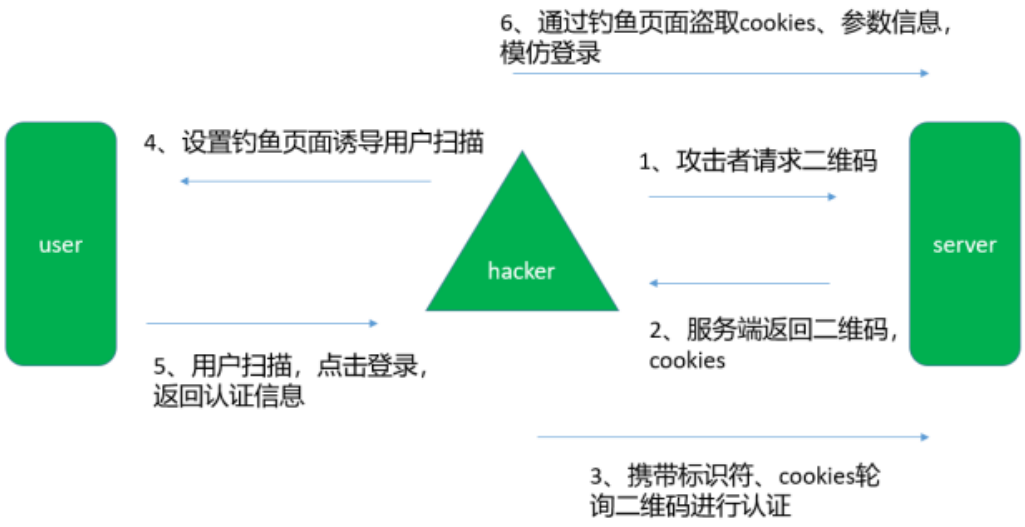
3. 手机端扫描二维码，在手机端已登录情况下，扫描网页二维码，二维码状态变为已扫描，并提示手机端点击确认登录。

4. 在确认点击登录后，二维码状态变为确认。

5. 此时完成与pc端登录连接，不再询二维码。

流程说明	二维码状态
获取二维码状态	此时状态未失效(new)
拒绝此次请求	状态为被拒绝(expired)
长时间未扫描	为二维码过期状态（expired）
扫描后等待确认登录	状态为（scaned）
登录认证完成后，二维码不在轮询	状态为已确认登录(confirmed)

劫持原理



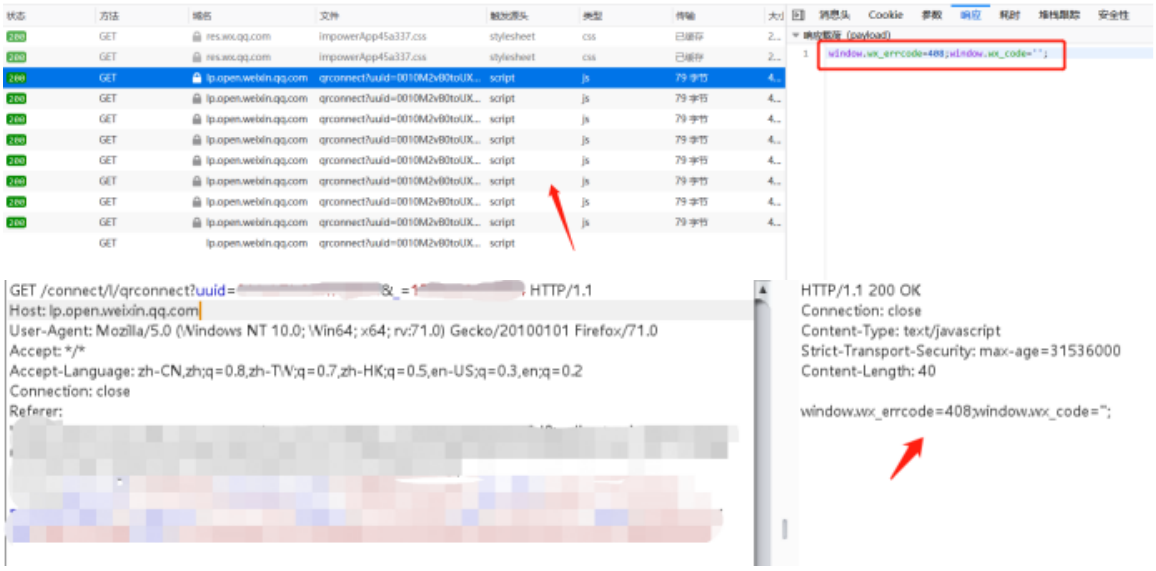
此时大家已经知道了基本二维码的认证流程，并熟悉了三种登录场景，那怎么才能劫持登陆呢，结合看过的文，举例一种试用场景。
前提：用户扫描后需在app端点击确认登录按钮才可登陆成功。

1. 攻击者将web登录二维码伪装成公众号二维码。
2. 攻击者轮询二维码保证其有效性，设置钓鱼页面。
3. 此时伪造钓鱼页面中的时候，可设置触发事件直接代替用户点击登录（此处可自由发挥）。
4. 当用户扫描后，攻击者获取用户的登录凭证。
5. 已经成功构造拼接链接，攻击者获取相关服务进行会话交互，获取敏感信息。

认证流程

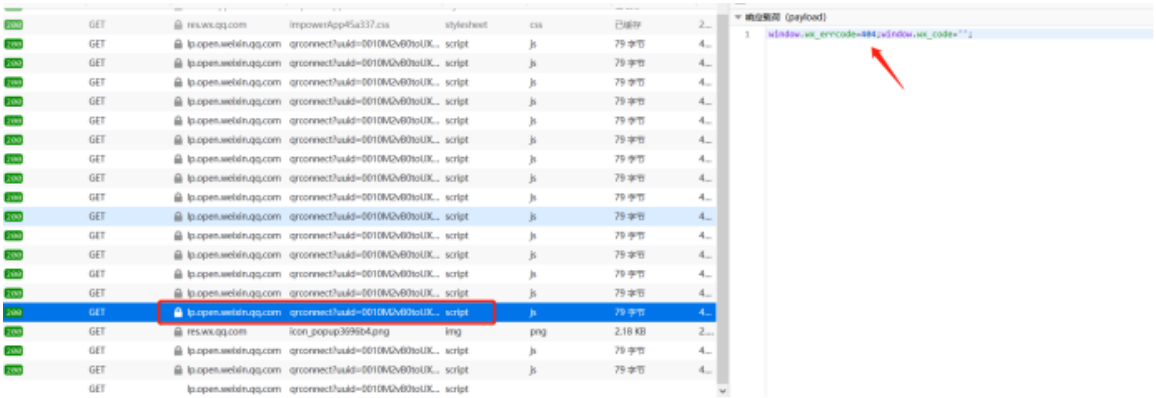
看千万遍不如亲自走一遍，fofa大法搜索微信扫码登录，找到目标，先走一遍流程分析，首先扫描二维码。

此时发起轮询，请求二维码状态，请求发现返回为408时为轮询状态，此时未登录。

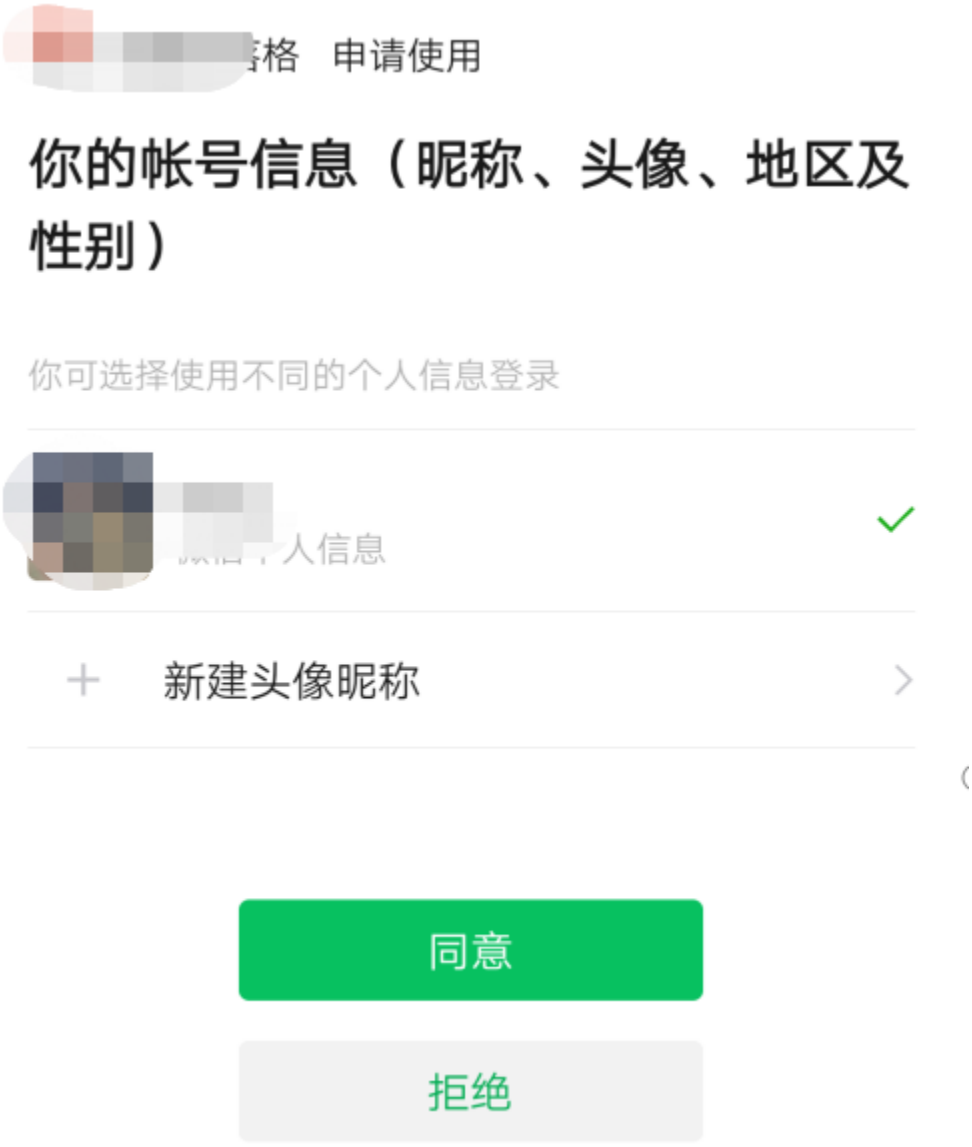


扫描该二维码，pc端显示扫描成功。

此时显示状态为404，表明为已扫描状态。



此时执行点击登录。



点击登录后，pc端已显示登陆成功。

此时完整流程已全部走完，我们回过头总结一下。

1. 首先请求该网站二维码登录链接，获取二维码与appid。

https://open.weixin.qq.com/connect/qrconnect?appid=wxef0e36842be56d2d&redirect_uri=https://www.xxx.com/e/memberconnect/weixin/loginend.php&response_type=code&scope=snsapi_login&state=xxx

2. 网页开始轮询，保证二维码时效性，且返回状态为408，轮询链接如下，uuid为pc端唯一标识，最后数字代表轮询次数的标识符。

https://lp.open.weixin.qq.com/connect/l/qrconnect?uuid=xxxx&_=1596604727840

3.扫描该二维码后，状态更改为已扫描，状态显示为404，扫描后链接如下（未点击登录）。

https://lp.open.weixin.qq.com/connect/l/qrconnect?uuid=xxxx&last=404&_=1596602065382

4.接下来需在app端点击登录，并且成功获取内容，链接如下：

http:/www.xxx.com/e/memberconnect/weixin/loginend.php?code=xxx&state=xxxx

5.重新发送轮询链接后，发现返回了code字段的value值。

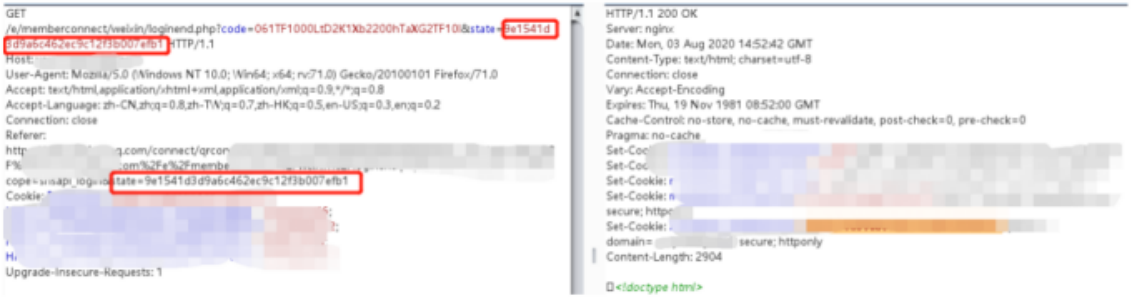
window.wx_errcode=405;window.wx_code='xxxxxxxxx';

我们分析一下登陆的url

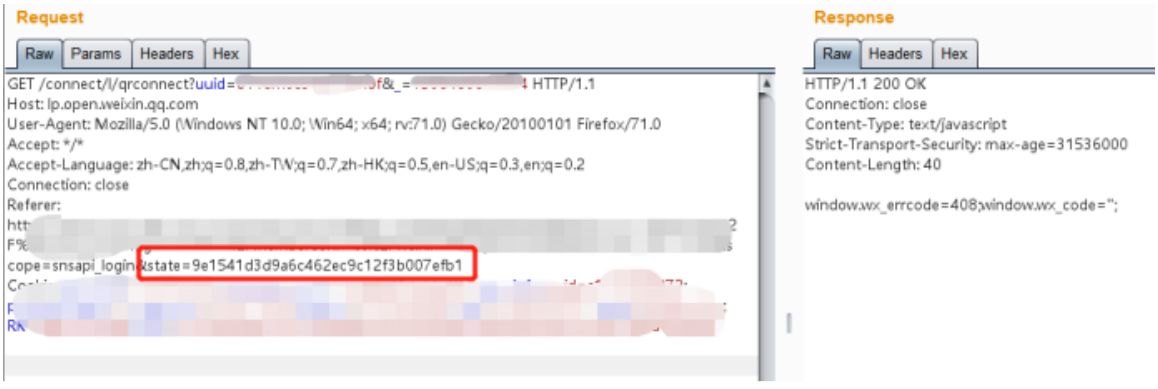
www.xxx.com/e/memberconnect/weixin/loginend.php?code=xxxx&state=xxx

发现两个参数，code、state,分析数据后发现state该参数值存在于referter字段，且在轮询时数据包也同样存在该值，并且得到登录成功cookie,猜想只要获取code字段value以及cookie即可伪造登录。

（已登录报文）



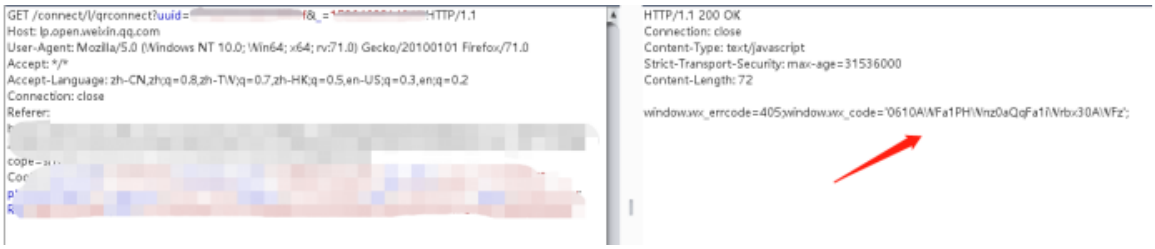
（轮询报文）



Cookie

```
GET
/e/memberconnect/weixin/loginend.php?code[redacted]&state=7a842955
831da1428190be219815e8b8 HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Connection: close
Referer:
http[redacted]https%3A%2
F%[redacted]ype=code&s
cope=snsapi_login&state=d4a1b9907[redacted]
Cookie: Hm_lvt_49c4b0ea69fee92a4371c846dbe445b[redacted];
Hm_lvt_ac325f03b1062f6ff332a9994497eb01
Hm_lpvt_49c4b0ea69fee92a4371c846dbe445b
Hm_lpvt_ac325f03b1062f6ff332a9994497[redacted]
moyublog_v2020returnu[redacted]
[redacted]ublog_v2020checkfeed[redacted]
[redacted]yublog_v2020checkplke[redacted]
PHPSESSID=gg6lt2spqj773nq[redacted]
Upgrade-Insecure-Requests: 1
```

接着继续发送轮询数据包，发现在登陆成功后，状态为405，并且返回code字段的value。

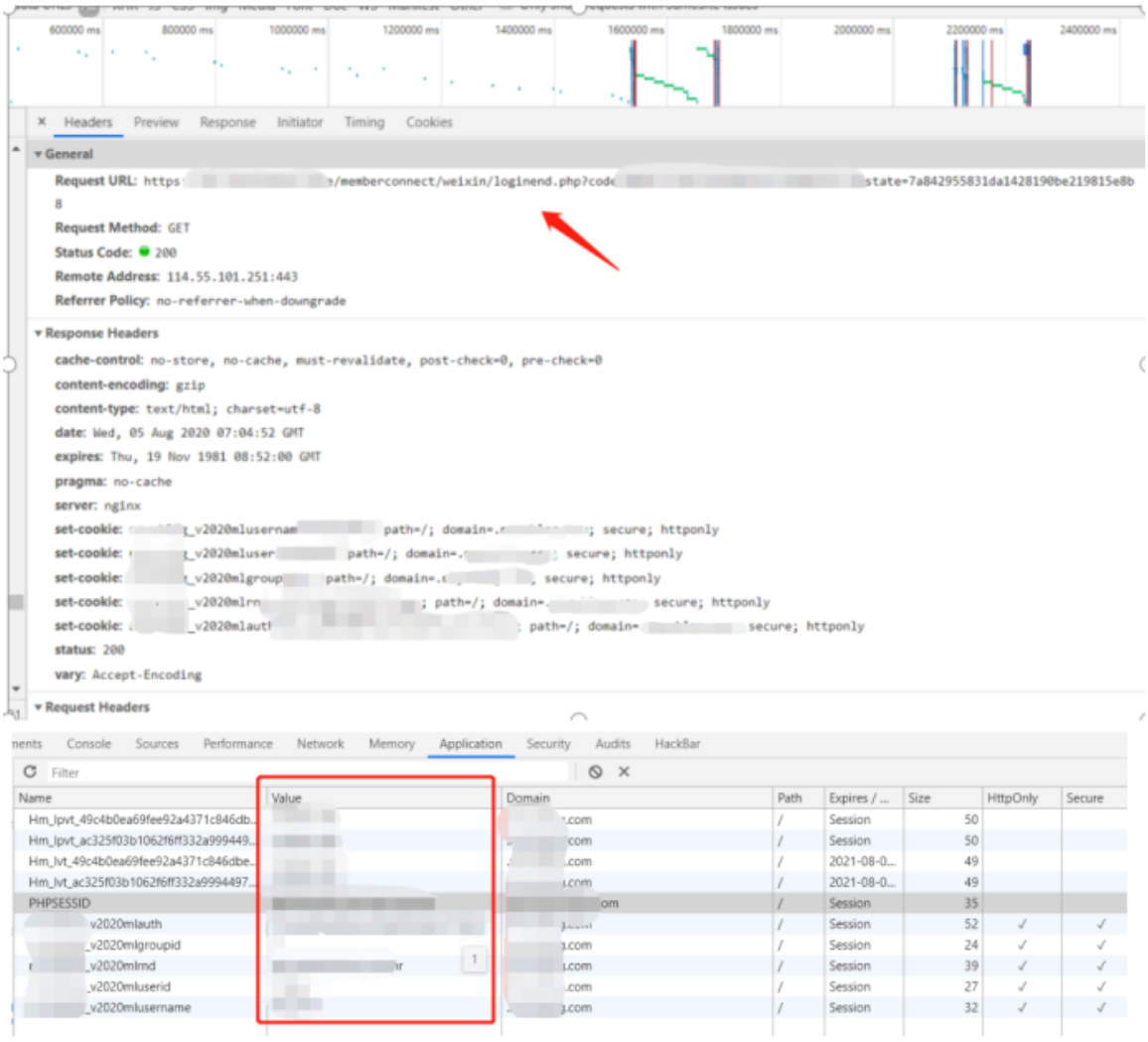


尝试攻击

我们的目标是获取用户登录成功时的cookie，再次提交轮询获取code字段value，并且通过轮询返回状态重复发送认证请求，保证二维码有效期。（此处我直接使用小号尝试）

发送二维码，用户扫描（受害者）。

获取cookie以及code。



利用已获取cookie、code尝试登录成功（攻击者）。

反思

之前一直以为可以劫持二维码登录，类似请求伪造，后来发现并不是，又找了几个网页二维码测试后，直接盗取cookie劫持登录操作存在于很多扫码登录站点这种问题，这更像是设计缺陷，只是的确是存在钓鱼风险，好吧，分析了个寂寞。

精选留言

用户设置不下载评论