

0x01 前言

前期通过灯塔 + ffuf + oneforall 等工具组合进行子域名收集，得到目标站点，漏洞挖掘中多次踩坑成功获取某大厂后台权限

0x02 渗透日常——单点登录

目标 URL: <https://xxxx.xxxxxxx.com/#/user/login>

(tmd, 这里有个坑就是该站点只能使用 firefox 访问, 别的浏览器试了谷歌, 联想自带浏览器, edge 访问都不行, 人给我整麻了。。。。)

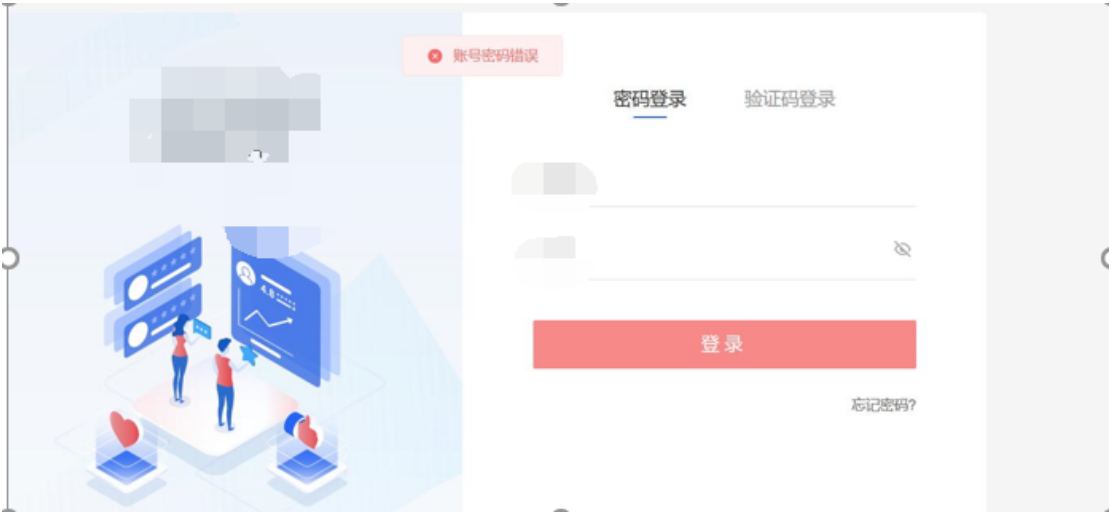


单点登录, 想必各位大佬上手就会日, 我就不再赘述。

常规操作, 爆破

将密码设置为 top100 字典 账号设置为 top10000, 用 burpsuite 进行鱼叉式爆破

经过测试，密码错误时的访问时这样的



通过响应包回显时这样的

Request	Payload	Status	Error	Timeout	Length	Comment
1288		200	<input type="checkbox"/>	<input type="checkbox"/>	1575	
1583		200	<input type="checkbox"/>	<input type="checkbox"/>	1575	
2864		200	<input type="checkbox"/>	<input type="checkbox"/>	1575	
4555	da	200	<input type="checkbox"/>	<input type="checkbox"/>	1575	
2230	ad	403	<input type="checkbox"/>	<input type="checkbox"/>	1551	
2231	admin.	403	<input type="checkbox"/>	<input type="checkbox"/>	1551	
2232	'or 1=1--	403	<input type="checkbox"/>	<input type="checkbox"/>	1551	
2235	admin'or"='	403	<input type="checkbox"/>	<input type="checkbox"/>	1551	
2245	'or"='	403	<input type="checkbox"/>	<input type="checkbox"/>	1551	
2201		400	<input type="checkbox"/>	<input type="checkbox"/>	354	
1	admin	200	<input type="checkbox"/>	<input type="checkbox"/>	332	
10	administrator	200	<input type="checkbox"/>	<input type="checkbox"/>	296	
0		200	<input type="checkbox"/>	<input type="checkbox"/>	296	

Request

Response

Pretty

Raw

Hex

Render

⌵

⌵

```
1 HTTP/2 200 OK
2 Server: nginx
3 Date: Wed, 12 Jan 2022
4 Content-Type: applicati utf-8
5 Expires: Wed, 12 Jan 2022 10:10:30 GMT
6 Cache-Control: max-age=0
7
8 {
9   "sysMessage": "成功",
10  "sysCode": 1000,
11  "isSuccess": true,
12  "businessMessage": "账号密码错误",
13  "businessCode": 1002
14 }
```

运气不错。经过漫长的爆破也是出了几个账号，对爆破出的账号再

进行密码爆破



你以为这就结束了？

用测试出来的账号 amanxxxx 密码 xxxx 登录，无任何反应，登录界面仍然是登录界面，账号也是那个账号，只留下懵逼的我

后来在测试中发现用爆破出来的，账号密码登录后，并不会跳转到后台，还是跳转到登录界面，但是根据多次尝试，这些账号密码确实是存在并且正确的，并不是误报

0x02 Fuzz 目录，出现转机

放弃是不可能放弃的，继续 fuzz 目录，发现某处存在 /js/ 目录，于是对 js 目录进行爆破，最终提取出可以访问的，发现如下图

```

wir  domainConfig = {
  homeUrl: 'https://www.31huiyi.com',
  // API 网关
  gatewayServer: 'https://[redacted]tencent-[redacted]',
  // c 网关
  cGatewayServer: 'https://[redacted]tencent-[redacted]',
  // 文件服务
  fileServer: 'https://[redacted]tencent-[redacted]',
  // 大屏站点
  evosSite: 'https://[redacted]tencent-[redacted]',
  // 展示站点
  expoSite: 'https://[redacted]31huiyi.com',
  // sso 站点
  ssoSite: 'https://[redacted]tencent-[redacted]',
  // c sso
  cSsoSite: 'https://[redacted]tencent-[redacted]',
  // 移动端站点
  microSite: 'https://[redacted]tencent-[redacted]',
  // 微信小程序设计端站点
  microClientSite: 'https://[redacted]tencent-[redacted]',
  // pc 端设计端站点
  pcSite: 'https://[redacted]tencent-[redacted]'
}

```

通过浏览器 unicode 编码后，发现大概如下

```

homeUrl: 'https://www
// 网关
gateServer: 'https
// c 网关
cGatewayServer: 'http
// 文件服务
fileServer: 'https://
// 大屏站点
evosSite: 'https://cc
// 展示站点
expoSite: 'https://te
// sso 站点
ssoSite: 'https://aut
// c sso
cSsoSite: 'https://ca
// 移动端站点
microSite: 'https://c
// 微信小程序设计端站点
microClientSite: 'ht
// pc 端设计端站点

```

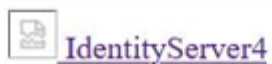
通过 jsfinder + burpsuite ，遍历出还存活的链接

访问其中某站点，发现如下



又是登录。。。。。 我累了。鲨了我吧

仔细一看，有个 logout?问题是我没有登录过啊，想起刚才那登录无反应的账号，发现这里显示的账号和那里的一样，是同一个，于是进行测试，删除浏览器 cookie 重新访问



LoginAsync

Choose how to login

Error

- Invalid username or password

Local Account

Username

Password

☒ Remember My LoginAsync

卧槽，和刚才那个似乎不太一样，少了个 logout。

整理了一下思路，发现大致流程如下

(在 <https://xxxx.tencent.com/#/user/login> 登录

例如使用 amxxx 123456 登录，他跳转回登录界面，但是在此处能显示用此账号登录)

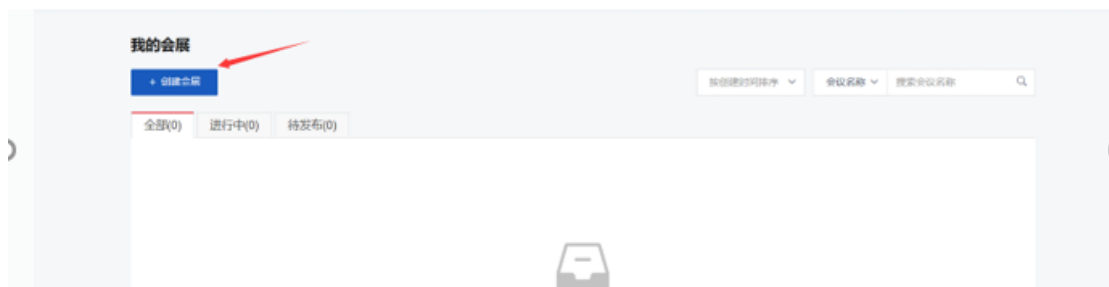
经过反复的测试，发现的这里也就比之前的那个多显示了一个，表明该账号已登录，也没有返回什么数据（当时感觉整个人裂开来了）



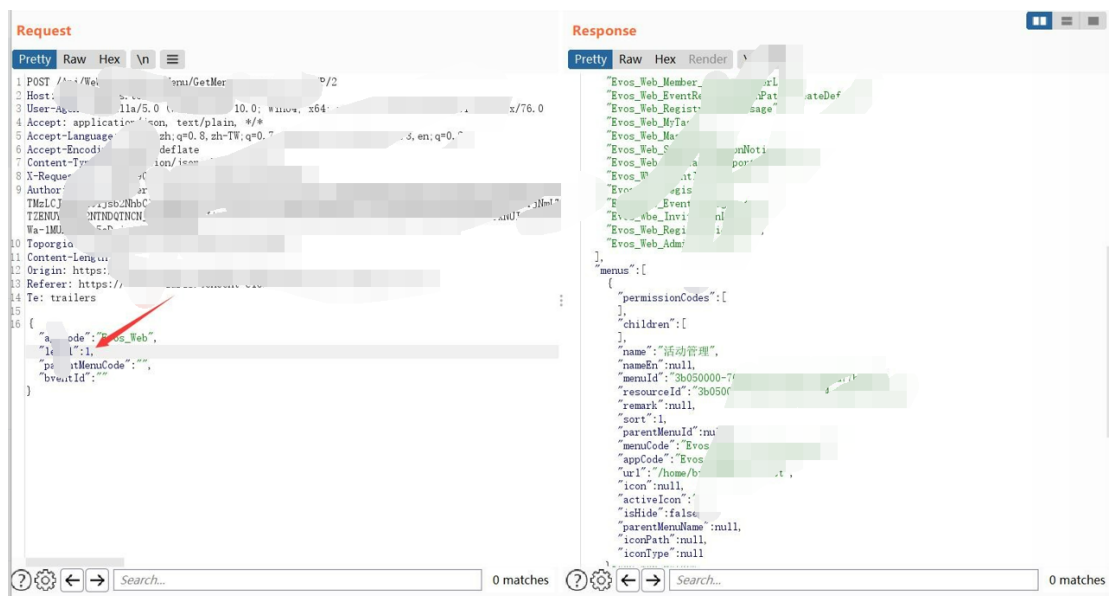
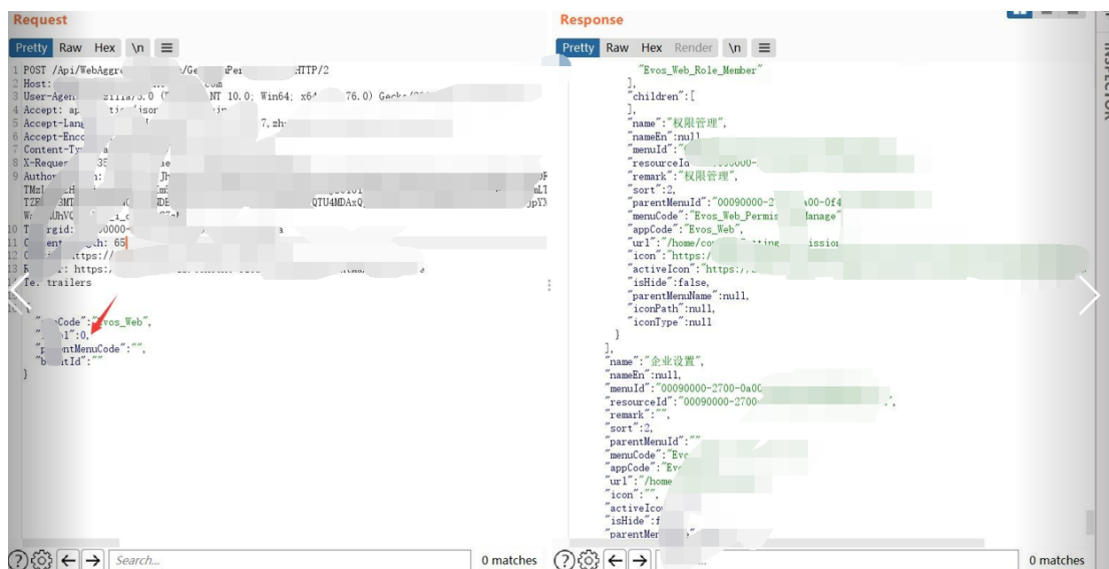
重新回去看这几个站点，生怕遗漏了什么，功夫不负有心人，发现之前遍历 js 文件里的接口，是有着某种规律的，如图

```
homeUrl: 'https://www. ....com',
// API 网关
gatewayServer: 'https://api. ....tencent. ....com',
// 内容分发网络
cdnGatewayServer: 'https://cdn. ....tencent. ....com',
// 文件存储
fileServer: 'https://file. ....tencent. ....com',
// 大会直播
evosServer: 'https://evos. ....tencent. ....com',
// 展览
expoServer: 'https://expo. ....tencent. ....com',
// 社交
ssoServer: 'https://sso. ....tencent. ....com',
// 客服
cssoServer: 'https://csso. ....tencent. ....com',
// 微服务
microServer: 'https://micro. ....tencent. ....com',
// 支付
payServer: 'https://pay. ....tencent. ....com',
```

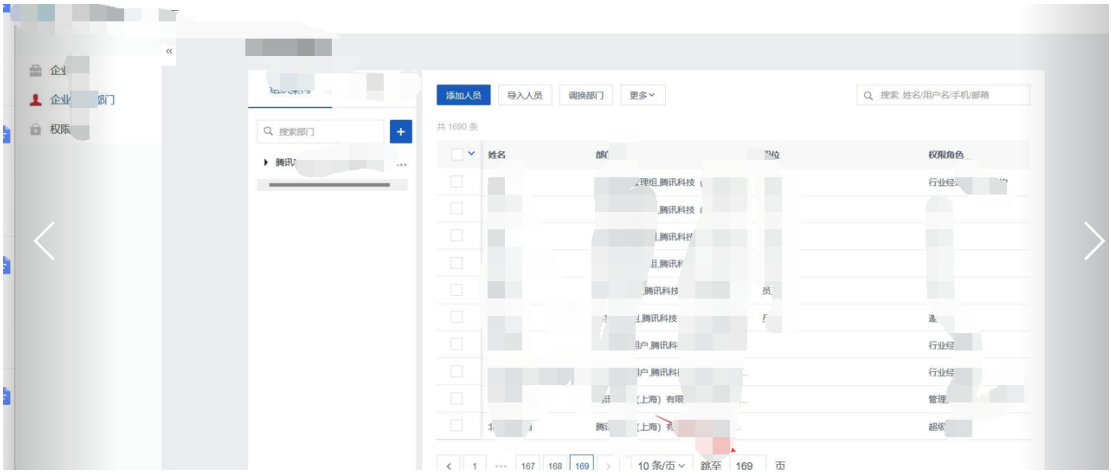
这就好办了，设置域名前缀为 polaxxxx.xxxxxx.com 再次进行子域名爆破，结合之前得到的信息，通过 ffuf 发现某处已经没有在用的 api 接口，通过 f12 调试器得到如图



接下来就很好办了，通过 burpsuite 抓包，修改 JSON 中的 level 值，一发入魂



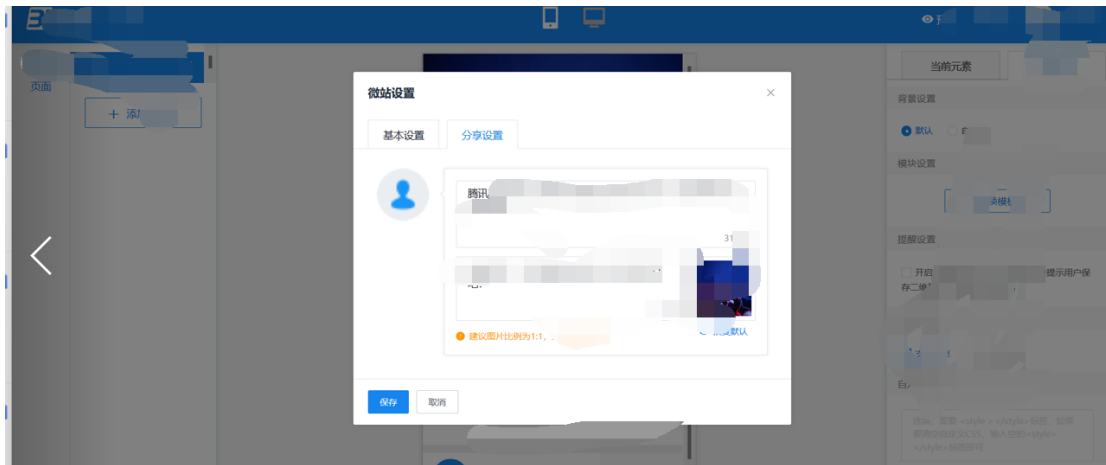
访问站点



支付宝查询行业经理信息



最后，通过该域名站点 cookie 共用的功能，直接无登录再次拿下某后台



0x03 总结

从单点登录无法利用 -> ffuf 接口 -> fuzz 可用子域 -> get 后台